Statistical Understanding of Adversarial Robustness

Morgane Goibert

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT052

Thèse de doctorat

TELECOM Paris

IP PARIS

Statistical Understanding of Adversarial Robustness

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°574 École doctorale de mathématiques Hadamard (EDMH)
Spécialité de doctorat: Mathématiques Appliquées

Thèse présentée et soutenue à Palaiseau, le 30 Novembre 2023, par

Morgane Goibert

Composition du Jury :

Florence d'Alché-Buc Professor, Télécom Paris (LTCI - S2A)	Présidente/Examinatrice
Seyed-Mohsen Moosavi-Dezfooli Assistant Professor, Imperial College London (DEEE)	Examinateur
Anna Korba Assistant Professor, ENSAE (CREST - Statistics)	Examinatrice
Yann Chevaleyre Professor, Université Paris-Dauphine PSL (LAMSADE)	Rapporteur
Eyke Hüllermeier Professor, LMU München (Computer Science)	Rapporteur
Stéphan Clémençon Professor, Télécom Paris (LTCI - S2A)	Directeur de thèse

Remerciements

Sur la route semée d'embûches que constitue le doctorat, mes premiers remerciements se dirigent en premier lieu à ceux qui ont su baliser mon chemin : mes superviseurs.

Stéphan, d'abord, merci pour ton soutien pendant ces quatre années, pour ton mentorat sur nos divers projets ensemble qui m'a permis non seulement d'aboutir à des publications efficaces mais aussi d'élargir mes horizons mathématiques, ainsi que pour ton calme indéfectible qui m'a parfois permis de garder le mien dans les moments stressants !

Clément, ensuite, merci pour ton dévouement lors de ta supervision qui n'était, après tout, pas prévue ! Merci de m'avoir apporté la structure et la rigueur dont j'avais besoin lors de nos travaux, et surtout merci d'avoir su trouver la meilleure dose de supervision pour m'aider à évoluer pendant ma thèse.

Elvis, merci pour avoir cru en moi depuis le départ, pour ton accompagnement si bienveillant, pour tout ce que tu m'as apporté lors de nos innombrables discussions, ainsi que les opportunités que tu m'a présentées

Ekhine, merci pour ton aide précieuse sur les rankings, tous nos échanges qui m'ont aidé à y voir plus clair sur nos projets communs, et tes propositions qui ont permis d'aller plus loin dans ma compréhension de mes sujets.

Je voudrais également remercier les membres de mon jury de thèse: Florence d'Alché-Buch, Eyke Hullermeïer, Yann Chevaleyre, Seyed-Mohsen Moosavi-Dezfooli et Anna Korba pour leur temps, leur pertinence et leur appréciation. En partciulier, merci aux deux rapporteurs pour leur long travail et leurs rapports !

J'en profite également pour remercier Criteo et l'ensemble des équipes qui m'ont permis de mener ma thèse dans un contexte on ne peut plus idéal. Je voudrais remercier tous celles et ceux qui m'ont accompagné ou aidé pendant ma thèse: Liva, l'ensemble de l'équipe EEL (Lorenzo, Louis, Maria, Hugo, Jérémie, Marc, Vianney), la team des doctorants (en particulier Thibaut, Ilana, Otmane et tous les autres) et tous les autres, en vrac Ludovic, Jean-Yves, Alain, Ugo,... Vous m'avez tous permis de réaliser ma thèse dans de meilleures conditions, ponctuées d'une ambiance agréable de travail, de petits conseils, et de partage. Merci à tous.

Enfin, ces quatre années auront été beaucoup moins évidentes également sans le soutien et l'entourage de mes proches.

Je pense d'abord à mes parents, qui étaient forcément beaucoup plus stressés que moi pendant mon parcours et qui ont su plus ou moins bien le dissumuler. Merci de m'avoir soutenue et d'avoir tenté de m'aider de votre mieux pendant cette période, pour trouver comment faire passer mes articles en conférence ou même en relisant l'intégralité de ma thèse malgré le manque de compréhension !

Je voudrais aussi remercier mon grand-père, qui a été une source d'inspiration pour me lancer dans un parcours de recherche, et dont l'incomparable curiosité m'a impregnée

depuis longtemps.

Je voudrais aussi dire un mot à mes amies de la team lycée et de la team primaire pour leur soutien pendant ces quatre années et qui m'ont approté un équilibre important. En partciulier, merci à Morgane et Laura, qui ont eu le courage de venir assister à ma soutenance !

Un grand merci également à Vincent à qui je dois une grande part de bonheur, et dont l'incroyable compréhension et bienveillance m'ont beaucoup aidé à aborder les derniers mois de ma thèse un peu plu sereinement.

Enfin, je voudrais en profiter pour mentionner mes petites boules de poils, Calix, Arya et Estë, dont la simple présence et les calins ont été si importants pour moi.

Abstract

This thesis focuses on the question of robustness in machine learning, specifically examining two types of attacks: poisoning attacks at training time and evasion attacks at inference time.

The study of poisoning attacks dates back to the sixties and has been unified under the theory of robust statistics. However, prior research was primarily focused on classical data types, mainly real-numbered data, limiting the applicability of poisoning attack studies. In this thesis, robust statistics are extended to ranking data, which lack a vector space structure and have a combinatorial nature. The work presented in this thesis initiates the study of robustness in the context of ranking data and provides a framework for future extensions. Contributions include a practical algorithm to measure the robustness of statistics for the task of *consensus ranking*, and two robust statistics to solve this task.

In contrast, since 2013, evasion attacks gained significant attention in the deep learning field, particularly for image classification. Despite the proliferation of research works on adversarial examples, the theoretical analysis of the problem remains challenging and it lacks unification. To address this matter, the thesis makes contributions to understanding and mitigating evasion attacks. These contributions involve the unification of adversarial examples' characteristics through the study of under-optimized edges and information flow within neural networks, and the establishment of theoretical bounds characterizing the success rate of modern low-dimensional attacks for a wide range of models.

Resumé

Cette thèse se concentre sur la question de la robustesse en apprentissage automatique, en examinant spécifiquement deux types d'attaques : les attaques de contamination pendant l'apprentissage et les attaques d'évasion pendant l'inférence.

L'étude des attaques de contamination remonte aux années soixante et a été unifiée sous la théorie des statistiques robustes. Cependant, les recherches antérieures se sont principalement concentrées sur des types de données classiques, comme les nombres réels. Dans cette thèse, les statistiques robustes sont étendues aux données de classement, qui ne possèdent pas de structure d'espace vectoriel et ont une nature combinatoire. Les contributions de la thèse comprennent notamment un algorithme pour mesurer la robustesse des statistiques pour la tâche qui consiste à trouver un rang consensus dans un ensemble de données de rangs, ainsi que deux statistiques robustes pour résoudre ce même problème.

En revanche, depuis 2013, les attaques d'évasion ont suscité une attention considérable dans le domaine de l'apprentissage profond, en particulier pour la classification d'images. Malgré la prolifération des travaux de recherche sur les exemples adversaires, le problème reste difficile à analyser sur le plan théorique et manque d'unification. Pour remédier à cela, cette thèse apporte des contributions à la compréhension et à l'atténuation des attaques d'évasion. Ces contributions comprennent l'unification des caractéristiques des exemples adversaires grâce à l'étude des paramètres sous-optimisés et à la circulation de l'information au travers des réseaux de neurones, ainsi que l'établissement de bornes théoriques caractérisant le taux de succès des attaques, récemment créées, de faible dimension.

Resumé détaillé en français

Motivation : Comprendre l'Importance de la Robustesse en Apprentissage Automatique

La robustesse constitue désormais un domaine essentiel de la recherche en apprentissage automatique, et elle est devenue encore plus importante avec l'avènement des applications interactives basées sur l'apprentissage automatique. En effet, les algorithmes d'apprentissage automatique sont utilisés dans une vaste gamme d'applications, notamment la reconnaissance d'image, le traitement du langage naturel, la reconnaissance vocale et les systèmes de recommandation. Ces applications basées sur de l'apprentissage automatique ont désormais envahi notre quotidien : qui n'a jamais entendu parler, vu ou utilisé des véhicules autonomes, des systèmes de recommandation de films, des modèles de langage génératif à grande échelle, etc. ? Toutes ces technologies ont rapidement été déployées au cours des dernières années grâce aux progrès exceptionnels du domaine de l'apprentissage automatique, qui a su produire des technologies très efficaces pour nous assister au quotidien. Cependant, avec le nombre croissant d'applications critiques de l'apprentissage automatique, disposer de technologies efficaces ne suffit plus. Nous avons maintenant besoin d'applications d'apprentissage automatique non seulement efficaces, mais aussi sûres et fiables, pour éviter des comportements défaillants graves résultant de nos modèles.

De nombreuses situations mettent en lumière la vulnérabilité des données et des modèles à une utilisation abusive, aux erreurs et aux biais. Par exemple, en 2016, le journal Bloomberg a réalisé une analyse montrant qu'Amazon excluait principalement des zones habitées par des personnes noires de certains de ses services de livraison. Bien que non intentionnelle, cette exclusion était influencée par des facteurs raciaux qui n'avaient pas été correctement pris en compte, entraînant un biais d'équité. Un autre exemple concerne les accidents causés par le système Autopilot des voitures autonomes de Tesla : en 2023, le Washington Post a conclu que 736 accidents (et 17 décès) s'étaient produits depuis 2019, probablement en raison de défauts du système qui ne reconnaît pas correctement certains obstacles tels que les motos ou les véhicules d'urgence stationnés sur le bord de route.

De telles situations illustrent les risques potentiels associés aux systèmes d'apprentissage automatique. Garantir la sécurité de ces systèmes dans des conditions normales et anormales constitue un défi majeur pour la communauté de l'apprentissage automatique aujourd'hui et dans un avenir prévisible. Dans le contexte plus large de la construction d'une IA digne de confiance, englobant divers domaines tels que l'équité, la confidentialité ou l'explicabilité, le domaine de la robustesse émerge comme un domaine d'intérêt

particulièrement intrigant. La robustesse aborde des scénarios dans lesquels les modèles d'apprentissage automatique rencontrent des entrées ou des données ayant été manipulées de manière malveillante pour tromper la réponse du modèle. À mesure que les utilisateurs interagissent plus fréquemment avec les systèmes d'apprentissage automatique, ces tentatives d'exploiter les points faibles des modèles deviennent de plus en plus courantes.

Considérez l'exemple des véhicules autonomes, comme illustré par le travail de Eykholt et al. (2018). Les auteurs ont démontré qu'ils pouvaient créer des patchs à coller sur des panneaux de signalisation pour empêcher des modèles de reconnaissance d'images de reconnaître correctement ces panneaux, ce qui met en lumière les risques d'accidents énormes si de tels patchs étaient utilisés.

Pour faire face à de tels problèmes, le domaine de la robustesse s'est développé de manière indépendante dans différentes zones de l'apprentissage automatique, comme le détaillera la Section 1.2. Un aspect supplémentaire intéressant des études de robustesse est leur relation avec les autres sujets de l'IA digne de confiance, qui revêtent tous une importance majeure. En particulier, la robustesse est liée à la question de l'explicabilité des modèles d'apprentissage automatique, car comprendre pourquoi certains modèles sont si vulnérables aux exemples adverses est une question prédominante dans le domaine.

Introduction à la Robustesse

L'intérêt pour la construction de méthodes statistiques robustes n'est pas nouveau. De telles notions ont d'abord émergé dans le domaine de la physique, où, selon Huber and Ronchetti (2009), de nombreux chercheurs tels que Simon Newcomb ou Arthur Eddington avaient une bonne compréhension des concepts de robustesse à la fin du XIXe siècle. Cependant, des travaux structurés autour de la robustesse ont été principalement initiés par Huber dans les années soixante, avec, par exemple, Huber (1964), puis formulés sous la forme d'un livre complet dans Huber and Ronchetti (2009). Ce que Huber appelait "robustesse" dans ses travaux englobait en réalité la robustesse contre ce que nous appelons aujourd'hui des attaques par *empoisonnement* : il abordait la robustesse des modèles ou des statistiques contre la contamination des données d'entraînement. Les notions pertinentes sur la robustesse par empoisonnement de Huber seront détaillées dans la Section 1.3.1.

En plus des travaux fondamentaux de Huber, de nouveaux types d'attaques ont également émergé contre des algorithmes spécifiques : dans le domaine de l'apprentissage profond, les problèmes de robustesse ont connu un regain d'intérêt indépendant en 2013, lorsque les auteurs de Szegedy et al. (2013) ont découvert la notion *d'exemples adverses* dans le contexte de la vision par ordinateur, comme détaillé dans la Section 1.4.1. Contrairement à celles étudiées par Huber, de telles attaques ne visent pas à modifier le résultat des algorithmes appris en ciblant les données d'entraînement, mais se concentrent plutôt sur le fait de tromper un bon algorithme préalablement entrainé en modifiant les données au moment de l'inférence, ce que l'on appelle les *attaques d'évasion*.

La première partie de cette thèse sera consacrée à la robustesse contre les attaques par empoisonnement. Bien que ce sujet ait déjà été traité dans la litérature, notamment par Huber and Ronchetti (2009); Fox and Weisberg (2002); Ben-Tal and Nemirovski (2000);

Møller et al. (2005) et bien d'autres, ces travaux se sont majoritairement concentrés sur des types de données classiques, à savoir des données réelles ou multivariées. Les concepts liés à des types de données plus complexes avec des topologies complexes ont été peu étudiés auparavant : c'est notamment le cas pour les données de classement, où seul le travail Agarwal et al. (2020) existait antérieurement à cette thèse. La Section 1.3 introduira donc les concepts pertinents pour les attaques par empoisonnement, les défis spécifiques liés aux données de classement, ainsi que les contributions de la thèse à ce sujet.

La deuxième partie sera consacrée à la robustesse contre les attaques d'évasion. Comme ce concept a émergé dans le domaine de l'apprentissage profond pour la vision par ordinateur mais reste largement obscur, la présente thèse se concentrera sur ce domaine et fournira une meilleure compréhension de ce phénomène. Ce concept a été découvert dans Szegedy et al. (2013), et a ensuite été largement étudié. De nombreux travaux ont proposé différents algorithmes d'attaque, parmi lesquels Goodfellow et al. (2014); Madry et al. (2018); Carlini and Wagner (2017); Moosavi-Dezfooli et al. (2016) sont de bons exemples. Une quantité équivalente de travail a été consacrée à la robustification des algorithmes d'apprentissage profond, avec différentes stratégies telles que celles présentées dans Papernot et al. (2016); Hendrycks and Gimpel (2016); Ma et al. (2018); Madry et al. (2018); Shafahi et al. (2019a) entre autres. Concomitamment à ces travaux visant à mettre en œuvre des attaques adverses ou des méthodes robustes en pratique, la littérature s'est également concentrée sur une meilleure compréhension du phénomène. Un premier ensemble de travaux a aboutis à des résultats théoriques sur l'existence d'exemples adverses, tels que Tsipras et al. (2019); Fawzi et al. (2018b); Dohmatob (2019). Un deuxième ensemble de travaux a examiné les caractéristiques des exemples adverses pour permettre d'expliquer leur succès, bien que les caractéristiques exactes et les raisons sous-jacentes de l'efficacité des exemples adverses restent obscures et encore débattues dans la communauté. La Section 1.4 introduira donc le concept d'attaques adverses plus en profondeur, ainsi que les découvertes récentes sur leur fonctionnement et détaillera les contributions de cette thèse sur ce sujet.

Les deux types d'attaques précédemment introduits peuvent être étudiés dans le même contexte choisis par cette thèse : l'apprentissage automatique supervisé, où les tableaux de données consistent généralement en des couples constitué des la donnée d'entrée et sa classe associée.

Données. En apprentissage automatique, les données consistent généralement en les éléments suivants:

- $X = (X_1, ..., X_m) \in \mathcal{X}^m$ qui sont les *données d'entrée*, où $\mathcal{X}$ est l'espace des données d'entrées et m est sa dimension.
- $Y \in \mathcal{Y}$ est la prédiction. Pour une tâche de classification à K classes Y est le *label* et $\mathcal{Y} = [\![1, K]\!]$
- X et Y sont des variables aléatoires distribuées selon la loi jointe $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X}, \mathcal{Y})$, où $\mathcal{M}_+^1(\mathcal{X}, \mathcal{Y})$ est l'ensemble des mesures de probabilité sur $\mathcal{X} \times \mathcal{Y}$
- Commes les variables aléatoires X et Y, et la distribution $P_{X,Y}$ ne sont pas connus

en pratique, nous nous basons sur des observations empiriques. $S_N = \{(x_i, y_i), i \in [\![1, N]\!]\} \overset{\text{i.i.d.}}{\sim} P_{X,Y} \in (\mathcal{X}, \mathcal{Y})^N$ correspond au jeu de données disponible, qui définit une distribution empirique: $\hat{P}_N = \sum_{x,y \in S_N} \delta_{x,y}$, où δ_a est une distribution de Dirac en a. Pour simplifier, nous identifions généralement $\hat{P}_N$ et S_N.

Modèle. Un modèle d'apprentissage automatique est défini avec les éléments suivants:

- $\mathcal{F} \subseteq (\mathcal{X} \to \mathcal{Y})$ correspond à la classe de modèles (supervisés).
- $F : P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X}, \mathcal{Y}) \to f \in \mathcal{F}$ est un algorithme qui apprend à partir d'une distribution des données et qui retourne un modèle spécifique. Un modèle résultant d'une algorithme d'apprentissage automatique sera généralement noté f.

Attaques Bien qu'une définition rigoureuse des attaques soit proposée en Definition 1.2.1, concentrons nous sur une explications intuitive de ce qu'est une attaque, qu'elle soit fonctionne par empoisonnement ou par évasion. Dans les deux cas, l'objectif d'une attaque est de faire en sorte que l'évaluation d'un certain modèle (un modèle entrainé sur des données corrompues dans le premier cas, ou un modèle entrainé normalement dans le second) sur certaines données (des données normales dans le premier cas, des données corrompues dans le second) soit significativement moins bonne que si le modèle normalement entrainé avait été évalué sur des données normales. Tous ces éléments (la mesure d'évaluation, le "budget" d'attaque et l'écart entre l'évaluation du modèle attaqué et du modèle normal) sont des paramètres de ces attaques.

Attaques par Empoisonnement et Données de Préférence : Notions, Difficultés et Contributions

Notions

Les attaques par empoisonnement ont été largement étudiées dans la litérature pour les données multivariées notamment. Ce n'est cependant pas le cas lorsque l'espace des données est moins pratique que l'espace des données réelles. Cette limitation s'applique particulièrement à l'espace des données de préférence, qui présente deux défis majeurs : le manque de structure d'espace vectoriel et la nature combinatoire de l'espace.

Une introduction détaillée à l'espace des données de préférence sera fournie dans le Chapter 2, mais concentrons-nous sur une desciption brève de cet espace. L'espace des données de préférence est l'espace des permutations sur n éléments, c'est-à-dire le groupe symétrique $\mathfrak{S}_n$. Une préférence est notée par $\sigma \in \mathfrak{S}_n$ et représente la préférence (d'un utilisateur) sur un ensemble de n éléments.

Dans le cadre des attaques par empoisonnement sur les données de préférence, cette thèse se concentre sur la robustification de ce qu'on appelle la tâche d'estimation du paramètre de position, comme expliqué plus en détail en Section 1.3.1.

Concrètement, l'objectif est de trouver une statistique $T : P \in \mathcal{M}_+^1(\mathcal{Y}) \mapsto \mathcal{Y}$, dont

l'élément de sorti doit correspondre au mieux au centre de la distribution P. Intuitivement, il s'agit de trouver "la moyenne" de P, sachant qu'une telle notion de moyenne n'est pas défini dans le cadre de données de préférence. Dans la litérature sur les données de préférence, cette "moyenne" prend le nom de "consensus".

Trouver un bon "consensus" est une problématique qui a été longuement traitée, et dont la solution la plus classique consiste à résoudre le problème d'optimisation suivant :

$$T(P) = \underset{\sigma \in \mathfrak{S}_n}{\operatorname{argmin}} \mathbb{E}[l(\sigma, \Sigma)],$$

où l désigne une distance adéquate sur l'espace des données de préférence. Lorsque le choix de l correspond à ce qu'on appelle la distance de Kendall, la solution obtenue s'appelle le consensus de Kemeny et correspond à la méthode la plus connue pour résoudre ce problème. Cependant, il doit être noté que plusieurs distances l ont été définies sur cet espace, et qu'en fonction de celle utilisée, le résulat, c'est-à-dire le consensus, n'est pas le même. A cela s'ajoute le fait que ce problème d'optimisation est difficile à résoudre dans un cadre général. Malgré la formulation assez simple du problème qui nous occupe dans cette partie de la thèse, trouver un bon consensus n'est donc pas aisé.

Difficultés

De ce fait, la question de trouver un consensus qui soit non seulement pertinent mais aussi robuste est d'autant plus difficile. En effet, plusieurs défis se présentent:

- L'espace des données de préférence n'est pas un espace vectoriel. De ce fait, certaines opération évidentes dans le cadre de données réelles telles que l'addition ou la multiplication ne peuvent pas être définies entre deux préférences. Cela complexifie l'étude de cette espace de deux manière. D'abord, parce que certaines procédure qui peuvent tout de même être généralisées à cet espace deviennent très couteuses en temps de calcul (comme ça peut être le cas de la résolution du problème d'optimisation du consensus). Ensuite, parce que certains concepts ne peuvent pas être facilement généralisés à cet espace (comme c'est le cas de la notion de quantile).
- L'espace des données de préférence est un espace combinatoire. C'est un espace fini à $n!$ éléments, ce qui signifie que pour appliquer certains algorithmes en pratique, il faut être particulièrement vigilant au temps de calcul des algorithmes.
- La robustesse dans un espace de données aussi particulier et complexe que celui des données de préférence n'est pas bien définie et est un sujet encore jamais abordé dans la litérature. Les bases nécessaires à la résolution du problème du consensus robuste sont donc manquantes.

Contributions

La première partie de cette thèse concerne la robustification de la tâche d'estimation du paramètre de position (consensus) d'une distribution de données sur l'espace des données de préférences.

Pour ce faire, cette thèse propose deux contributions majeures dans les Chapters 3 and 4 qui découlent de deux publications différentes. La première, appelée *Statistical Depth*

Functions for Ranking Distributions: Definitions, Statistical Learning and Applications par Morgane Goibert, Stéphan Clémençon, Ekhine Irurozki et Pavlo Mozharovskyi a été publiée à la conférence AISTATS 2022, voir Goibert et al. (2022a). La seconde, appelée *Robust Consensus in Ranking Data Analysis: Definitions, Properties and Computational Issues* par Morgane Goibert, Clément Calauzènes, Ekhine Irurozki et Stéphan Clémençon a été publiée dans la conférence ICML 2023, voir Goibert et al. (2023).

Chapitre 3. Plus spécifiquement, le Chapter 3 qui reprend majoritairement l'article Goibert et al. (2022a) s'attache à développer une procédure de robustification inspirée de celles de Huber en construisant des statistiques de rangs sur les distributions sur les données de préférence.

En effet, les statistiques basées sur les rangs sont très utiles pour définir des analogues de quantiles, qui peuvent à leur tour fournir des caractéristiques beaucoup plus informatives sur une distribution étudiée P que simplement la médiane, c'est-à-dire le consensus. Le but de ce chapitre est de définir ces analogues de quantiles, de rangs et les procédures statistiques pertinentes basées sur de telles quantités pour l'analyse des données de préférence au moyen d'une notion de fonction de profondeur basée sur une métrique sur le groupe symétrique.

En surmontant l'absence de structure d'espace vectoriel sur l'espace des données de préférence, la fonction de profondeur proposée définit une notion d'ordre (du centre vers l'extérieur) pour les préférences dans le support de P et étend la la recherche classique, sans robustesse, du consensus.

Les propriétés axiomatiques que les fonctions de profondeur de sur l'espace des données de préférence devraient idéalement posséder sont énumérées, et les problèmes computationnels et de généralisation sont étudiés en détail. Au-delà de l'analyse théorique réalisée, la pertinence des nouveaux concepts et méthodes est illustrée par la création d'une stratégie de troncage pour renforcer le consensus de Kemeny, qui s'inspire des statistiques de moyenne ou de médiane tronquées dans le contexte des données réelles.

Cette stratégie de troncage est démontrée comme étant plus performante que le consensus de Kemeny en termes de robustesse, tant sur le plan théorique qu'empirique. De plus, il est démontré que les procédures basées sur la profondeur sont pertinentes pour d'autres tâches de statistique classique, ce qui met en évidence l'utilité et la flexibilité de ce concept pour les données de préférence.

Pour résumer, les contributions de ce chapitre sont donc les suivantes :

- La profondeur statistique et les propriétés axiomatiques associées sont étendues aux données de préférence afin d'étendre les notions de quantiles et de positions pour des variables aléatoires évaluées dans l'esapce des données de préférence.
- Une analyse sur échantillon fini garantit la praticité d'utilisation de la notion de profondeur de classement que nous venons d'introduire.
- Un algorithme d'une grande simplicité qui utilise la fonction de profondeur pour construire des distributions de préférences empiriques stochastiquement transitives (sur la base desquelles des tâches statistiques cruciales telles que le retrouver le consensus de Kemeny sont simples) est proposé.

- La fonction profondeur et ses régions de quantiles associées dans l'espace des données de préférence peuvent être utilisées pour l'analyse statistique des données de préférence pour de nombreuses tâches : 1) une récupération rapide et robuste des consensus, 2) des représentations graphiques informatives des données de préférence, 3) la détection d'anomalies et de nouveautés, 4) les tests d'homogénéité.

Chapitre 4. Ensuite, le chapitre Chapter 4 qui reprend majoritairement l'article Goibert et al. (2023) s'attache à établir une évaluation complète du gain, en terme de robustesse, apporté par une statistique robuste par rapport à une statistique usuelle comme le consensus de Kemeny.

Dans ce chapitre, notre attention se porte sur l'introduction d'un algorithme d'approximation spécifiquement conçu pour évaluer la robustesse de toute statistique en se basant sur la notion de *point de rupture*, tout en abordant les défis computationnels associés. Cette méthode d'évaluation de la robustesse constitue un outil précieux pour mesurer la résilience de différentes statistiques face à des scénarios adverses.

De plus, nous présentons un plugin de statistique robuste capable d'améliorer la robustesse de toute statistique classique utilisée pour résoudre le problème du consensus. Cette méthode offre non seulement des gains significatifs en robustesse mais garantit également une perte minimale de précision. Cette caractéristique montre l'intérêt de notre approche, la positionnant comme une meilleure solution que les méthodes existantes telles que le consensus de Kemeny pour résoudre cette tâche de manière précise et robuste.

En exploitant ces avancées, nous visons à fournir un cadre complet pour évaluer et améliorer la robustesse des statistiques de consensus. À travers une analyse rigoureuse et des évaluations empiriques, nous démontrons les avantages pratiques de notre méthode et son potentiel à surpasser les approches traditionnelles.

Dans ce chapitre, nous complétons le Chapter 3 sur la question de la robustesse à la manipulation des votes en examinant comment le concept de point de rupture peut s'appliquer à la tâche du consensus.

L'une des principales difficultés dans ce contexte réside dans le fait que les consensus usuels sont souvent obtenus en résolvant un problème d'optimisation et qu'aucune forme analytique simple pour les solutions n'est généralement disponible. Par conséquent, le calcul des points de rupture des statistiques sur les données de préférence constitue généralement un défi computationnel. Notre proposition principale ici consiste à approximer ce calcul en résolvant une version assouplie du problème d'optimisation du point de rupture en utilisant une technique de lissage qui permet de calculer des gradients pertinents et éventuellement d'effectuer une descente de gradient.

De plus, nous fournissons également un plugin robuste qui peut être ajouté à n'importe quelle statistique de consensus. Au-delà du consensus de Kemeny tronquée présentée dans le Chapter 4, nous tirons parti de la structure spécifique de l'espace de des données de préférences pour fournir une méthode de robustification spécifique. L'idée est d'assouplir la contrainte stipulant que le consensus d'une distribution sur les données de préférence doit nécessairement être représenté par une préférence stricte qui ordonne tous les éléments. Au lieu de cela, nous suggérons d'autoriser le consensus à être une préférence "en seau", c'est à dire qu'autoriser la possibilité d'observer des ex-aequo entre les éléments, ce

que s'avère avoir des avantages cruciaux en matière de robustesse.

Pour résumer, les contributions de ce chapitre sont donc les suivantes :

- Nous proposons une évaluation théorique de la robustesse, mesurée par la fonction de rupture, des statistiques de consensus usuelles. Plus précisément, nous dévoilons une borne inférieure générale pour leur fonction de rupture et une borne supérieure pour le consensus de Kemeny.
- Nous fournissons un algorithme pratique qui approxime la fonction de rupture de n'importe quelle statistique de consensus. Cet algorithme peut s'adapter aux statistiques produisant une préférence stricte ou en seau.
- Nous proposons une extension des concepts pertinents (métriques et distances, fonction de rupture, etc.) pour les préférences en seau.
- Nous créons un plugin appelé le plugin de Fusion Descendante (Downward Merge) qui fournit une couche robuste après avoir calculé un consensus usuel. Le plugin de fusion descendante s'avère empiriquement très efficace pour renforcer le consensus avec une perte minimale de précision : il constitue ainsi un meilleur choix par rapport aux alternatives classiques comme le consensus de Kemeny.

Attaques par Evasion en Apprentissage Automatique : Notions, Difficultés et Contributions

Notions

Dans le cadre de l'apprentissage automatique, la robustesse prends une forme un peu différente de celle que nous avons vu pour les données de préférence. Le phénomène appelé "attaques adversaires" dans la communauté a été découvert en 2013 dans Szegedy et al. (2013), où les auteurs ont montré qu'il était très facile de tromper un réseaux de neurones dont l'objectif est de faire de la classification d'images , et ce de manière quasiment systématique.

Avant de se pencher sur ces attaques adversaires, résumons en un mot ce que sont les modèles d'apprentissages automatiques, c'est-à-dire les réseaux de neurones. Dans le cadre de la classification d'images, un réseau de neurones doit indiquer quel est l'objet représenté par une image, parmi une liste d'objets possible définis à l'avance. Intuitivement, un réseau de neurones est une fonction qui prend comme argument un vecteur, dans notre cas un vecteur de grande dimension représentant une image, et qui en sort, pour chaque objet de la liste la probabilité que l'image représente cet objet. La fonction qui définit un réseau de neurones est assez simples : il s'agit simplement d'un mélange d'opérations linéaires et d'activations non linéaires. La particularité d'un réseau de neurones réside en fait dans le fait qu'il est paramétré par un très grand nombres de paramètres. Pour trouver les "bons" paramètres pour résoudre la tâche, un réseau de neurones doit être entrainé : cette étape se fait grace à une fonction de perte, qui mesure l'eurreur que fait le réseau dans sa réponse, et un algorithme de descente de gradients, qui permet d'améliorer les paramètres (et les performances du réseau) étape par étape. De ce fait, il est très complexe de comprendre quels sont les paramètres optimaux pour un réseau de neurone

sur une tâche données, ou pour comprendre comment le réseau est arrivé à un ensemble de paramètres une fois l'entrainement fini : on dit que les réseaux de neurones sont des boîtes noires.

Dans leur article Szegedy et al. (2013), les auteurs se sont rendus compte qu'il était possible de tromper systématiquement un réseau pourtant très bien entrainé sur une tâche de classification d'images en rajoutant une perturbation imperceptible malveillante sur les données. Ainsi, un réseau qui fait très bien la différence entre des images de chats et de chiens pourra être trompé par une nouvelle image de chat à laquelle on rajoute cette perturbation adversaire, alors que la différence n'est pas visible à l'oeil nu. Ces images modifiés par une perturbation très minime et qui trompent très souvent un réseau de neurones sont appelées des exemples adversaires. Pour les obtenir, depuis 2013, les chercheurs de la communauté ont développé de nombreux algorithmes différents qui tentent de calculer la meilleure perturbation, ce qu'on appelle les attaques adversaires. En parallèle, d'autres travaux se sont concentrés sur tenter de rendre les réseaux de neurones plus robustes à ces exemples adversaires. Enfin, certains travaux se sont penchés sur l'analyse des exemples adversaires pour tenter d'expliquer leur succès en répondant à ces questions : quelles sont les caractéristiques des exemples adversaires qui les rendent si efficaces ? Peut-on analyser théoriquement la robustesse ou la vulnérabilité des réseaux de neurones contre certains types d'attaques adversaires ?

La compréhension du phénomène des exemples adversaires reste encore très parcellaire, et les découvertes faites par les chercheurs dans ce domaine entrent parfois en contradiction. Cette seconde partie de la thèse s'attache donc à éclaircir ce phénomène et à proposer une meilleure compréhension des exemples adversaires.

Difficultés

Les réseaux de neurones sont très difficiles à analyser d'un point de vue théorique, à cause notamment de la complexité de l'apprentissage d'un réseau de neurones, de sa dimension aléatoire, et de sa très grande dimensionalité. De ce dait, il est tout aussi complexe d'étudier le phénomène des exemples adversaires. Malgré les travaux entrepris depuis la découverte du phénomène en 2013, sa compréhension reste très obscure. Deux limitations principales peuvent êtres notées :

- Le manque de compréhension adéquate du phénomène des exemples adversaires. De nombreux travaux ont cherché ce qui rend les exemples adversaires efficaces, et plus généralement ont étudié les caractéristiques des exemples adversaires. Comme il est très difficile d'étudier théoriquement les réseaux neuronaux, la grande majorité des travaux reposent soit sur des méthodologies expérimentales, soit sur des travaux théoriques sur des versions simplifiées des réseaux neuronaux. Dans les deux cas, les découvertes sur ces sujets reposeent sur une accumulation de preuves, et tous les articles ne sont pas d'accord sur les mêmes conclusions. Comme le domaine de la robustesse adversaire est encore assez récent, de nombreuses hypothèses n'ont pas encore été explorées et les méta-analyses sont rarement disponibles.
- Les réseaux de neurones sont peut-être intrinsèquement vulénrables. Un ensemble de stravaux s'est concentré à étudié théoriquement la vulnérabilité des réseaux de neurones. Ces travaux fournissent en général, des bornes sur le succès (ou l'échec)

des exemples adversaires, mais sont limités par les hypothèses qu'ils doivent faire sur les réseaux, sur la distribution des données, ou sur les méthodes d'attaque adversaire poru parvenir à un résultat. Avec le développement de méthodes d'attaque adversaire de plus en plus sophistiquées, de tels travaux doivent rester à jour avec les avancées heuristiques les plus récentes des attaques. Très récemment, un changement important a modifié le développement des exemples adversaires, avec la découverte des attaques universelles et des attaques de basse dimension. Ces attaques se concentrent essentiellement sur la modification d'un petit sous-espace des images fournies en entrées, contrairement aux attaques plus classiques qui sont conditionnées uniquement sur un budget global, en modifiant par exemple un unique pixel. De tels exemples adversaires n'opèrent pas sur la totalité de la dimensionnalité des données, et donc les techniques de preuve traditionnellement utilisées dans le domaine, qui reposaient principalement sur le fléau de la dimensionnalité, ne peuvent plus être utilisées.

Contributions

La seconde partie de cette thèse concerne l'étude des exemples adversaires contre les réseaux de neurones pour la classification d'images.

Pour ce faire, cette thèse propose deux contributions majeures dans les Chapters 7 and 8 qui découlent de deux publications différentes. La première, appelée *n Adversarial Robustness Perspective on the Topology of Neural Networks* par Morgane Goibert, Thomas Ricatte et Elvis Dohmatob a été publiée dans le ML Safety Workshop de la conférence NeurIPS 2022, voir Goibert et al. (2022b). La seconde, appelée *Origins of Low-dimensional Adversarial Perturbations* par Elvis Dohmatob, Chuan Guo et Morgane Goibert a été publiée à la conférence AISTATS 2023, voir Dohmatob et al. (2023).

Chapitre 7. Le Chapter 7 reprend majoritairement l'article Goibert et al. (2022b) et fournit un cadre regroupant différentes caractéristiques des exemples adversaires exposés dans la littérature, à travers l'étude d'un objet générique émergent des réseaux neuronaux, le graphe. Plus précisément, ce chapitre étudie l'impact de la topologie du réseau neuronal sur la robustesse adversaraire. Notre objectif principal est d'explorer la structure du graphe qui émerge lorsqu'une image d'entrée traverse toutes les couches d'un réseau de neurones. Nous découvrons des différences dans ces graphes en comparant les exemples normaux aux exemples adversaires. Plus précisément, les graphes dérivés des exemples normaux présentent une distribution plus centralisée autour de ce que nous appelons les "arêtes autoroutières". D'autre part, les graphes associés aux exemples adversaires affichent un motif plus diffus, exploitant stratégiquement les "arêtes sous-optimisées".

Pour établir l'intérêt de ces résultats, nous menons des expériences approfondies couvrant divers ensembles de données et architectures. Les résultats montrent que les arêtes sous-optimisées représentent une source de vulnérabilité pour les réseaux neuronaux, nous découvrons leur utilité dans la détection des exemples adversaires. Au-delà de ces résultats expérimentaux, nous fournissons un argument théorique corroborant l'importance des arêtes sous-optimisées pour la vulnérabilité des réseaux neuronaux et suggérons que les techniques d'élagage peuvent fournir plus de robustesse.

Pour résumer, les contributions de ce chapitre sont les suivantes :

- Nous proposons et justifions une hypothèse, regroupant plusieurs caractéristiques des adversaires, sur la manière dont la structure topologique des réseaux de neurones et les paramètres sous-optimisés sont liés au phénomène des exemples adversaires.
- Nous proposons méthode principale pour extraire des caractéristiques topologiques structurelles basées sur les diagrammes de persistance et les arêtes sous-optimisées.
- Nous menons des expériences pour valider notre hypothèse en utilisant nos caractéristiques nouvellement définies. Parmi les expériences réalisées, nous mettons au point un détecteur pour les exemples adversaires qui donne de meilleurs résultats que les méthodes de l'état de l'art.

Chapitre 8. Ensuite, le Chapter 8 se concentre sur les récents progrès dans la recherche d'algorithmes d'attaques adversaraires plus pratiques. Ces nouvelles attaques, dites attaques universelles et de basse dimension, ont modifié le paradigme des algorithmes d'attaques avec la création de perturbations adversaraires pouvant être trouvées par une recherche en boîte noire en utilisant étonnamment peu de requêtes, ce qui restreint essentiellement la perturbation à un sous-espace de dimension bien plus petite que la dimension de l'espace des images.

Les constatations empiriques du succès de ces attaques de basse dimension nous conduisent à émettre l'hypothèse que des perturbations adversaraires existent avec une probabilité élevée dans des sous-espaces de basse dimension, ce qui soulève la question : la vulnérabilité aux attaques en boîte noire de basse dimension est-elle inhérente ou pouvons-nous espérer les éviter ? Plusieurs travaux ont abordé ces questions pour des types d'attaques plus génériques (des attaques en pleine dimension), de tels résultats théoriques ne peuvent s'appliquer directement aux types d'attaques de basse dimension, car le principe de la malédiction de la dimension ne peut pas être utilisé.

Dans ce chapitre, nous entreprenons une étude rigoureuse du phénomène des perturbations adversariales de basse dimension. Nos résultats caractérisent précisément les conditions suffisantes pour l'existence de ces perturbations, et nous montrons que ces conditions sont satisfaites pour les réseaux neuronaux en pratique, y compris le régime dit "paresseux" où les paramètres du réseau entraîné restent proches de leurs valeurs à l'initialisation. En plus de cette contribution théorique, nos résultats sont confirmés par des expériences sur des données synthétiques et réelles.

Notre analyse théorique des perturbations adversaraires de basse dimension repose principalement sur la régularité du classifieur et sur les propriétés géométriques du sous-espace d'attaque. Les bornes auxquelles nous aboutissons mettent en lumière le rôle de plusieurs paramètres : 1) la régularité locale de la frontière de décision du classifieur, 2) l'alignement du sous-espace d'attaque avec les vecteurs normaux unitaires à la frontière de décision du classifieur, 3) la distribution de la marge ponctuelle du classifieur, 4) le budget de l'attaquant.

Pour résumer, les contributions de ce chapitre sont donc les suivantes :

- Nous formalisons la notion de sous-espace adversaire viable, qui fournit une caractérisation des sous-espaces de basse dimension qui peuvent être pertinents pour mener des attaques adversaires. Plus précisément, cette notion établit une condition d'alignement entre le sous-espace d'attaque et le gradient du modèle pour

que le sous-espace d'attaque soit utilisable en pratique pour élaborer des attaques adversaires réussies.

- Nous présentons nos bornes théoriques pour les modèles ayant une frontière de décision Lipschitzienne. Cette caractéristique de régularité nous permet d'obtenir des résultats généraux sur l'efficacité des perturbations adversariales de basse dimension, ce qui est également illustré dans des cas où le modèle est linéaire ou hyperellipsoïdal, par exemple.
- nNus présentons également nos bornes théoriques pour les modèles ayant des frontières de décision localement presque affines. Cette caractéristique de régularité nous permet d'obtenir des résultats similaires pour des modèles pratiques de pointe, par exemple, les réseaux neuronaux avec des fonctions d'activation ReLU dans le régime de caractéristiques aléatoires ou le régime paresseux.
- Nous réalisons des expériences pour illustrer la puissance informative de nos bornes théoriques pour les réseaux de neurones génériques entraînés. Nos bornes sont démontrées comme étant applicables dans ce cas, même lorsque le réseau de neurones est grand ou entraîné de manière adversaire.

Conclusion

Cette thèse se concentre sur la question de la robustesse en apprentissage automatique. La robustesse peut principalement être subdivisée en deux parties différentes : les attaques par empoisonnement qui ciblent les modèles lors de l'entraînement, et les attaques par évasion qui ciblent les modèles lors de l'inférence.

Il est intéressant de noter que la recherche sur ces deux types d'attaques en est à des stades très différents.

Les attaques par empoisonnement ont commencé à être étudiées dans les années 1960 et ont été unifiées sous une théorie exhaustive, généralement appelée statistiques robustes. Cependant, les principales limitations des études sur les attaques par empoisonnement sont dues à la restriction de la recherche aux types de données classiques, principalement les données réelles. Dans cette thèse, les statistiques robustes cont étendues aux données de classement, surmontant le manque de structure d'espace vectoriel et la nature combinatoire de l'espace. La plupart des travaux fournis dans cette thèse consistent donc à initier l'étude de la robustesse dans cet espace particulier et à fournir un cadre permettant des extensions de ces travaux de manière structurée.

En revanche, les attaques par évasion suscitent un grand intérêt dans le contexte de l'apprentissage profond pour la classification d'images depuis 2013. Ce domaine a été largement reconnu, déclenchant une prolifération de travaux de recherche sur le sujet des exemples adversaires. Ces travaux sont principalement expérimentaux en raison de la difficulté d'analyser théoriquement le problème et du manque d'unification. Pour résumer en quelques mots, les contributions de cette thèse sur ce sujet sont une unification de certaines caractéristiques des exemples adversaires à travers l'étude des arêtes sous-optimisées et de la topologie des réseaux de neurones, ce qui permet de mieux comprendre le fonctionnement des exemples adversaires et de créer une méthode de détection efficace; de plus,

nous développons des bornes théoriques (grâce à l'utilisation de la géométrie de l'espace adversaire et la régularité du modèle au lieu d'arguments basés sur la dimensionnalité) pour caractériser le taux de succès des attaques de basse dimension pour une large classe de modèles et illustrées par des expériences.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

It is a strange fate that we should suffer so much fear and doubt over so small a thing. Such a little thing.

J.R.R Tolkien, Lord of the Rings.

Contents

1.1 Motivation: Understanding the Role of Robustness in Machine Learning

Robustness is a critical aspect of machine learning research, and it has become even more important with the rise of interactive machine learning-based applications. Indeed, machine learning algorithms are used in a wide range of applications, including image recognition, natural language processing, speech recognition, and recommender systems. Machine learning systems have now flooded our daily lives: who has never heard about, seen, or used autonomous vehicles, movie recommendation systems, generative large language models, and so on? All of these technologies have rapidly been deployed in the last few years thanks to the exceptional progress of the machine learning field, that have been able to produce very efficient technologies to assist us daily. However, with the increasing number of critical applications of machine learning, having efficient technologies is not enough. We now need efficient, but also secure and trustworthy machine learning applications to avoid critical misbehaviors stemming from our models.

Numerous instances highlight the vulnerability of data and models to misuse, errors, and biases. For example, in 2016, the journal Bloomberg conducted an analysis showing that Amazon excluded predominantly Black areas from some of its delivery services. Although unintentional, this exclusion was influenced by racial factors that were not properly accounted for, resulting in fairness bias. Another example is the accidents caused by the Autopilot system of Tesla autonomous cars: in 2023, the Wahsington Post concluded that 736 accident (and 17 fatalities) occurred since 2019, probably due to defects of the system that does not correctly recognize certain obstacles like motorcycles or parked emergency vehicles.

Such situations exemplify the potential risks associated with machine learning systems. Ensuring the security and safety of these systems under both normal and non-normal conditions stands as a major challenge for the machine learning community today and in the foreseeable future. Within the broader context of building *trustworthy AI*, which encompasses diverse areas such as fairness, privacy, or explainability, the field of *robustness* emerges as a particularly intriguing area of focus. Robustness addresses scenarios in which machine learning models encounter inputs or data that have been maliciously manipulated to deceive the model's response. As users interact more frequently with machine learning systems, these attempts to exploit blind spots in the models become increasingly common.

Consider the example of autonomous vehicles, as exemplified by the work of Eykholt et al. (2018). The authors have shown their ability to create patches to stick on traffic signs that prevent the models from recognizing these signs correctly. Figure 1.1 shows an illustration of that, where a patch has been added to a stop sign, which is now predicted to be a 45 mph speed limit sign instead. The potential dangers of such attacks in the real world are readily apparent.

To tackle such issues, the field of robustness has grown independently in different areas of machine learning, as will be detailed in Section 1.2. An interesting additional aspect of robustness studies is its relation with the other topics from trustworthy AI, which are all of major importance. In particular, robustness is linked to the question of the explainability of machine learning models, as understanding why some models are so vulnerable to adversarial examples is a predominant question in the field. Moreover, robustness is

Figure 1.1: Adversarial attack against a real stop sign using black and white patches, from Eykholt et al. (2018). The stop sign is misclassified by deep learning models as a 45 mph speed limit sign.

closely linked to anomaly detection, as well as generalization to out-of-distribution data. It can help to better understand and improve the generalization power of machine learning models to unseen data. Robustness can thus help create models that can handle a wide range of inputs and scenarios.

In conclusion, robustness is a critical aspect of machine learning research, and it has become even more important with the rise of interactive machine learning applications. By building robust models that can handle unexpected or adversarial inputs, researchers can help to ensure that their models are reliable, fair, transparent, and safe for all users.

1.2 Introduction to the Robustness Studies

1.2.1 A Brief Overview

The interest in building statistical methods that are robust is not novel. Such ideas can be traced back a long time ago, especially in the field of physics, where, according to Huber and Ronchetti (2009), many researchers such as Simon Newcomb or Arthur Eddington had a good understanding of the concepts of robustness in the late 1900s. But structured work around robustness has in fact been initiated mainly by Huber in the sixties, with e.g. Huber (1964), and then formulated in the form of a comprehensive book in Huber and Ronchetti (2009). What Huber called 'robustness' in his works actually encompasses robustness against what we now call *poisoning* attacks: it tackles the robustness of models or statistical procedures against the contamination of *training* data. The relevant notions about poisoning robustness from Huber will be detailed in Section 1.3.1.

In addition to Huber's seminal works, novel types of attacks have also emerged against specific types of algorithms: in the area of deep learning, robustness issues have known an independent renewed interest in 2013, when the authors of Szegedy et al. (2013) unveiled the notion of *adversarial examples* in the context of computer vision, as detailed in Section 1.4.1. Such attacks, contrary to the ones studied by Huber, do not focus on modifying the outcome of the learned algorithms by targeting training data but rather focus on fooling a good learned algorithm by modifying data at *inference* time, which is

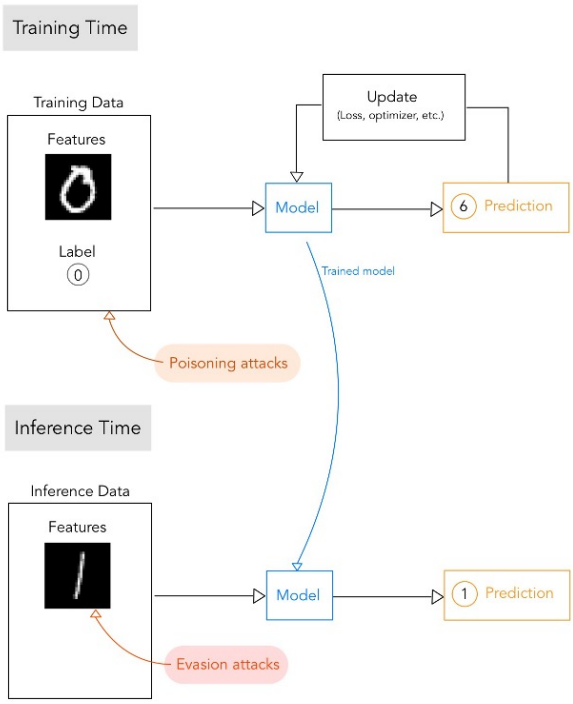


Figure 1.2: Schema of where poisoning and evasion attacks operate in machine learning. Poisoning attacks occur during or before training and focus on the training data. Evasion attacks occur at inference and target inputs submitted to an already trained model.

known as *evasion* attacks.

The functioning of both types of attacks is illustrated in Figure 1.2, and the present thesis will explore both types of attacks in two different parts.

The first part will be dedicated to robustness against *poisoning* attacks. Of course, this topic has been widely covered starting, as mentioned, from Huber in Huber (1964), for real-numbered data and methods such as regression problems Fox and Weisberg (2002); Hubert and Branden (2003), linear programming problems Ben-Tal and Nemirovski (2000), outlier detection Rousseeuw and Leroy (1987), and multivariate data analysis Møller et al. (2005); Zuo (2006), parameter estimation Diakonikolas et al. (2018, 2020), principal component analysis Hubert et al. (2005), etc. The common factor of these works is their focus on classical types of data, namely real numbered or multivariate data. Concepts with more complex types of data with challenging topologies have scarsely been studied before: this is specifically the case for ranking data, where only the work of Agarwal et al. (2020) preexisted. Section 1.3 will thus introduce the relevant concepts for *poisoning* attacks, the specific challenges related to ranking data, as well as the contributions of the thesis on this matter.

The second part will be dedicated to robustness against *evasion* attacks. As this concept has emerged in the field of deep learning for computer vision but is still largely obscure, the present thesis will focus on this area and provide a better understanding of this phenomenon. This concept was discovered in Szegedy et al. (2013), and deeply has been studied afterward. Many works have proposed different attack algorithms, among which

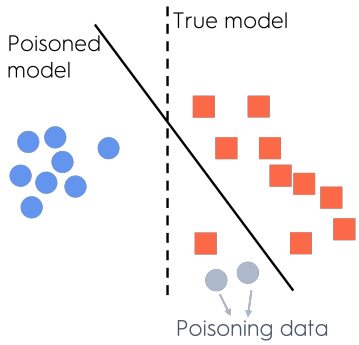
Goodfellow et al. (2014); Madry et al. (2018); Moosavi-Dezfooli et al. (2016); Carlini and Wagner (2017); Moosavi-Dezfooli et al. (2018); Chen et al. (2022); Guo et al. (2019) are good examples. An equivalent amount of work has been dedicated to robustifying deep learning algorithms, with various strategies, such as defensive distillation Papernot et al. (2016); Liang and Samavi (2023), detection methods Hendrycks and Gimpel (2016); Ma et al. (2018); Lee et al. (2018); Li et al. (2019), and the famous adversarial training strategies Madry et al. (2018); Pang et al. (2021); Zhang et al. (2019a); Shafahi et al. (2019b) among others. Concurrent with these works that aim at implementing adversarial attacks or robust methods in practice, the literature has also focused on better understanding the phenomenon. A first stream of work has derived theoretical results on the existence of adversarial examples, like Tsipras et al. (2019); Fawzi et al. (2018a); Dohmatob (2019). A second stream of work has investigated characteristics of adversarial examples to account for their success, even though the exact features and underlying reasons for adversarial examples' effectiveness remain unclear. For example, Papernot et al. (2017) demonstrated that adversarial examples can *transfer* to other neural networks, Goodfellow et al. (2014) proposed the local linearity of neural networks to justify the success of adversarial examples, but this was challenged by Tanay and Griffin (2016) with the opposite finding. Similarly, Tsipras et al. (2019) suggested that there is a fundamental tradeoff between robustness and accuracy, which is challenged by the opposite finding from Rozsa et al. (2016); Cubuk et al. (2017). These examples illustrate how debated the characteristics of adversarial examples are. In addition, Ilyas et al. (2019) showed that non-robust features exist in the data distribution, Moosavi-Dezfooli et al. (2019) showed that large curvature of the decision boundary negatively impacts robustness, Rice et al. (2020); Manoj and Blum (2021); Wu et al. (2021) suggest that overfitting of neural networks may be a source of vulnerability, etc. Section 1.4 will introduce the concept of *evasion* attacks through the lens of adversarial examples in deep learning, explain more deeply the main findings and challenges unveiled by the literature, as well as the contributions of the thesis on this matter.

1.2.2 Framework and Setup

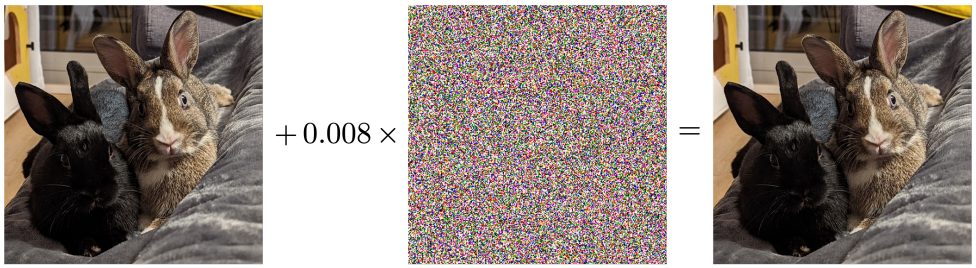
Even though the two types of attacks are different and require different notions, they both operate on the same machine-learning setup as illustrated by Figure 1.2. In this thesis, we study a general framework for *supervised* machine learning, where data usually consists in inputs *and* associated labels.

Data. Machine learning data consists of the following elements:

- $X = (X_1, ..., X_m) \in \mathcal{X}^m$ are the *input features*, where $\mathcal{X}$ is the input space and m denotes the dimensionality of the input. For example, for a MNIST image (see LeCun and Cortes (2010)), $\mathcal{X} = [0, 1]$ and $m = 784$.
- $Y \in \mathcal{Y}$ is the prediction. For a K-class classification tasks, Y is called the *label* and $\mathcal{Y} = [\![1, K]\!]$
- X and Y are random variables distributed according to an unknown joint probability distribution $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X}, \mathcal{Y})$, where $\mathcal{M}_+^1(\mathcal{X}, \mathcal{Y})$ is the set of probability measures on $\mathcal{X} \times \mathcal{Y}$



(a) Illustration of a poisoning attack



(b) Illustration of an evasion attack

Figure 1.3: Examples of a poisoning attack and an evasion attack.

- As the theoretical random variables X and Y, as well as the distribution $P_{X,Y}$, are not available in practice, we rely on empirical observations. $S_N = \{(x_i, y_i), i \in [\![1, N]\!]\} \overset{\text{i.i.d.}}{\sim} P_{X,Y} \in (\mathcal{X}, \mathcal{Y})^N$ is then the available dataset. This dataset defines an empirical distribution defined by $\hat{P}_N = \sum_{x,y \in S_N} \delta_{x,y}$, where δ_a denotes the Dirac distribution in a. To simplify the notation, we will usually identify $\hat{P}_N$ with S_N.

Model. A machine learning model can be defined using the following elements:

- $\mathcal{F} \subseteq (\mathcal{X} \to \mathcal{Y})$ denotes the (supervised) model class. It corresponds to all the possible models after choosing a type of machine learning technique: for example, choosing deep learning algorithms will result in a different model class than support vector machines.
- $F : P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X}, \mathcal{Y}) \to f \in \mathcal{F}$ denotes the algorithm that learns from the data distribution and outputs a specific model. When only the dataset S_N drawn from $P_{X,Y}$ is available, the algorithm can take as input the empirical distribution $\hat{P}_N$.

  A classical and broad type of algorithm is the *Risk Minimization* (RM) one, which can be described as follows:

$$F_{RM}^\star(P_{X,Y}) = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, \mathbb{E}_{X,Y \sim P_{X,Y}}(l(Y, f(X))), \tag{1.2.1}$$

  where $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function. Moreover, when $\hat{P}_n$ is used instead of $P_{X,Y}$ this is referred to as empirical risk minimization (ERM).

- A model, as outputted by a machine learning algorithm, will generally be denoted by $f \in \mathcal{F}$. Reusing the previous example, the RM model would be denoted by $f^\star_{RM} = F^\star_{RM}(P_{X,Y})$. Note that we will often drop the dependency in $P_{X,Y}$ (or S_N for the empirical version) in the notation whenever the context is clear.

Attacks. As mentioned in Section 1.2.1, two types of attacks can be considered: poisoning and evasion attacks, as illustrated by Figure 1.2. Both focus on different parts of a machine learning model life and if both target the data, they don't operate on the same distributions. However, they have similarities in their concepts, which can be summarized as follows: an attack is a modification of a data distribution (the train or the test distribution) aiming at creating a small distribution shift between the train and test distribution that would result in a large difference in their *evaluation*, which will be more formally defined for each type of attack in Sections 1.3 and 1.4. Two practical examples of such attacks are illustrated in Figure 1.3.

Definition 1.2.1. ATTACK. *Let $\mathcal{A}$ be a measurable space, $P \in \mathcal{M}^1_+(\mathcal{A})$ a distribution and F a supervised algorithm. Let m be a (normalized) metric over distributions, $\varepsilon \in [0,1]$ and $\delta > 0$. Finally, let $\mathcal{L} : \mathcal{M}^1_+(\mathcal{A}) \times \mathcal{F} \to \mathbb{R}$ be an evaluation metric to minimize for the output of algorithm F on distribution P.*

An attack over the distribution P and algorithm F with budget ε on m and amplitude at least δ on $\mathcal{L}$ is a distribution $Q_{m,\mathcal{L}}(F, P, \varepsilon, \delta)$ whose goal is to fool model $F(P)$ while satisfying a budget constraint depending on ε.

Thus, $Q_{m,\mathcal{L}}(F, P, \varepsilon, \delta)$ is defined as a distribution such that $m(P, Q_{m,\mathcal{L}}(F, P, \varepsilon, \delta)) \leq \varepsilon$ and

$$\textit{(Poisoning)} \quad \mathcal{L}(P, F(Q_{m,\mathcal{L}}(F, P, \varepsilon, \delta))) \geq \mathcal{L}(P, F(P)) + \delta \tag{1.2.2}$$

$$\textit{(Evasion)} \quad \mathcal{L}(Q_{m,\mathcal{L}}(F, P, \varepsilon, \delta), F(P)) \geq \mathcal{L}(P, F(P)) + \delta \tag{1.2.3}$$

Whenever the context is clear, the attack distribution will be simply denoted as Q_ε, where we drop the dependence in $\delta, m, \mathcal{L}, P$ and F in the notation.

1.3 Poisoning Attacks and Ranking Data: Notions, Challenges, and Contributions

This Section will introduce the relevant concepts to study the robustness of poisoning attacks: the attacks that target a model at *training* time, meaning that focus on changing the model learned using training data. Section 1.3.1 will introduce the task at hand in the present thesis, the poisoning attacks against the statistics that solve this kind of task, and all the related concepts to evaluate the robustness of a statistic as well as different classical methods that improve the robustness. Section 1.3.2 will detail the challenges associated with studying robustness to poisoning attacks in spaces presenting similar difficulties to those encountered in the ranking space. Finally, Section 1.3.3 will present a high-level overview of the contributions of this thesis on this field.

1.3.1 Huber's Robustness Concepts on Poisoning Attacks

Location estimation task. As previously mentioned, a broad class of learning problems can be defined as a Risk Minimization problem from Equation (1.2.1), which can be re-written the following way to simplify the notation:

$$f^\star_{RM} \in \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{X,Y\sim P_{X,Y}}(l(Y, f(X))) \tag{1.3.1}$$

In Theorem 2.8 of Steinwart (2007), it has been shown that such a problem can be equivalently mapped to a point-wise problem:

$$f^\star_{RM} : x \mapsto \operatorname*{argmin}_{y\in\mathcal{Y}} \mathbb{E}_{Y\sim P_{Y|X=x}}(l(Y, y)) \tag{1.3.2}$$

The focus can thus be moved to the inner problem, that is $\operatorname{argmin}_{y\in\mathcal{Y}} \mathbb{E}_{Y\sim P_{Y|X=x}}(l(Y, y))$, that leads to two remarks. First, in the context of poisoning attacks, where the attacker has access to the training distribution or dataset to change the learned model, it is then relevant to attack only the conditional distribution $P_{Y|X=x}$ rather than the joint distribution, as justified by Equation (1.3.2). Second, it is thus sufficient to solve the more general related problem:

$$\operatorname*{argmin}_{y\in\mathcal{Y}} \mathbb{E}_{Y\sim P}(l(Y, y)), \text{ for an arbitrary distribution } P \in \mathcal{M}^1_+(\mathcal{Y}). \tag{1.3.3}$$

Both remarks motivate our focus on the seminal work of Huber and Ronchetti (2009) and the following works on robust statistics for the *location estimation task*. Simply put, a *location* estimate is a statistic that is meant to estimate the average value of a dataset or distribution. For real numbers, the mean or the median are two types of location estimates which correspond to the formulation of Equation (1.3.3) when the metric l is the L_2-norm for the mean or the L_1-norm for the median respectively.

Definition 1.3.1. Location Estimation Task. *Solving the location estimation task consists in finding a statistic $T : P \in \mathcal{M}^1_+(\mathcal{Y}) \mapsto \mathcal{Y}$ to define the center of a given distribution.*

It is often defined as:

$$T(P) = \operatorname*{argmin}_{y\in\mathcal{Y}} \mathbb{E}_{Y\sim P}(l(Y, f(X))), \tag{1.3.4}$$

where l is a loss function.

Poisoning attacks on the location estimation task. As motivated by Definition 1.3.1, a poisoning attack on a location estimation model, or statistic, will target the predictor part of the data, meaning $Y \in \mathcal{Y}$. Inspired from Definition 1.2.1, it is more precisely defined as follows:

Definition 1.3.2. Poisoning attack on Location Estimation Statistics. *Let $\mathcal{Y}$ be the predictor set, $T : P \in \mathcal{M}^1_+(\mathcal{Y}) \mapsto \mathcal{Y}$ a statistic and $P \in \mathcal{M}^1_+(\mathcal{Y})$ an arbitrary distribution. Let ε and $\delta \in [0, 1]$, d be a metric on $\mathcal{Y}$ and m a metric on $\mathcal{M}^1_+(\mathcal{Y})$. Then, a poisoning attack of amplitude δ and of budget ε is defined as:*

$$d(T(P), T(Q_{\varepsilon,\delta})) \geq \delta \quad \textit{such that:} \quad m(P, Q_{\varepsilon,\delta}) \leq \varepsilon \tag{1.3.5}$$

The notation $Q_{\varepsilon,\delta}$ of the attack distribution does not reveal its dependence in distribution P, statistic T, and metrics d and m, as the context is clear.

Many poisoning attacks can be created for the same setup: usually, the robustness refers to the robustness against the *worst-case* poisoning attacks. Such a notion is incorporated into the definition of the different robustness measures.

Robustness measures. In the robustness literature, the main robustness measure of an estimator is called the *breakdown point*. Quoting Huber and Ronchetti (2009), "the breakdown point is the smallest fraction of bad observations that may cause an estimator to take on arbitrarily large aberrant values". The classical notion of breakdown point has usually been defined with an empirical finite sample version or an empirical asymptotic version, but the present thesis will provide a more theoretical, distribution-based definition that is more general and can be adapted easily in an empirical version.

Definition 1.3.3. BREAKDOWN POINT. *Let $\mathcal{Y}$ be a measurable space, $P \in \mathcal{M}_+^1(\mathcal{Y})$ a probability distribution, $T : \mathcal{M}_+^1(\mathcal{Y}) \to \mathcal{Y}$ a statistic, $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and $m : \mathcal{M}_+^1(\mathcal{Y}) \times \mathcal{M}_+^1(\mathcal{Y}) \to \mathbb{R}$ two metrics. The breakdown point for the statistic T on distribution P with metrics m and d is defined by:*

$$\varepsilon^\star(T, P, m, d) = \inf \left\{ \varepsilon > 0 \;\middle|\; \sup_{Q \mid m(P,Q) \leq \varepsilon} d(T(P), T(Q)) = \infty \right\} \tag{1.3.6}$$

To obtain an empirical version of the breakdown point when one prefers to study a dataset rather than a distribution, it is sufficient to replace the theoretical distribution P in the above definition with its empirical counterpart $\hat{P}_N$.

The breakdown point is a powerful tool to quantify the robustness of different statistics. In particular, the common sense observation that the median is more robust than the mean can be demonstrated by computing their respective breakdown points.

Example 1.3.4. *Let $\mathcal{Y} = \mathbb{R}$, $P \in \mathcal{M}_+^1(\mathbb{R})$ be a distribution, $T_{mean} : P \in \mathcal{M}_+^1(\mathbb{R}) \mapsto \operatorname{argmin}_{y \in \mathbb{R}} \mathbb{E}_{Y \sim P}((Y-y)^2)$ the mean statistic and, similarly, $T_{median} : P \in \mathcal{M}_+^1(\mathbb{R}) \mapsto \operatorname{argmin}_{y \in \mathbb{R}} \mathbb{E}_{Y \sim P}(|Y-y|)$ the median statistic.*

Let $m = TV$ be the total-variation distance, and $d = L_2$ the L_2-norm. Then we have the following results:

1) $\forall\ P \in \mathcal{M}_+^1(\mathbb{R})$, $\varepsilon^\star(mean, P, TV, L_2) = 0$.

2) $\forall\ P \in \mathcal{M}_+^1(\mathbb{R})$, $\varepsilon^\star(median, P, TV, L_2) = 1/2$.

The previous results can be obtained by observing that 1) a distribution $Q_a = (1-a)P + a\delta_b$, where δ_b is the Dirac measure in b with b going to $+\infty$, would have a mean of $+\infty$ even when a is infinitely small, and 2) obtaining a median equal to $+\infty$ requires allocating at least half of the probability mass to a Dirac in $+\infty$.

Intuitively, having a very small probability mass on the value $+\infty$ is enough to change the mean from a finite to an infinite value, whereas it requires changing half of a distribution to modify the median to infinity. The median is thus quantitatively much more robust than the mean, as measured by the breakdown point.

Other measures of robustness have been proposed in the literature. This is the case, for example, for the popular notion of the *influence function*.

Definition 1.3.5. Influence Function. *Let $\mathcal{Y}$ be a measurable space, $P, Q \in \mathcal{M}_+^1(\mathcal{Y})$ two probability distributions, $T : \mathcal{M}_+^1(\mathcal{Y}) \to \mathcal{Y}$ a statistic. The influence function for the statistic T on distribution P in direction Q is defined by:*

$$IF(T, P, Q) = \lim_{t \to 0^+} \frac{T((1-t)P + tQ) - T(P)}{t} \tag{1.3.7}$$

In particular, for $y \in \mathcal{Y}$, when $Q = \delta_y$ is the Dirac measure in y, we have

$$IF(T, P, y) = \lim_{t \to 0^+} \frac{T((1-t)P + t\delta_y) - T(P)}{t} \tag{1.3.8}$$

measures the influence in y.

The drawback of the influence function is that it focuses on measuring the influence of specific data points on a statistic. It is therefore not directly meant to compare different statistics between them, which is the reason why the present thesis will focus on the notion of breakdown point.

Classical robustification procedures. The robustification of the location estimation task has been treated in depth by the literature on robust statistics, in particular in the real numbers case. The studied strategies can be divided into several categories.

The first category includes all the statistics known as *M-estimators*. Simply put, an M-estimator is a statistic that generalizes the notion of maximum likelihood estimator. More formally, an M-estimator location statistic $T : S_N \in \mathcal{Y}^N \mapsto \mathcal{Y}$ is of the form: $T(S_N) = \text{argmin}_{t \in \mathcal{Y}} \sum_{y \in S_N} \rho(y, t)$, where ρ is an arbitrary and minimizable function. Such estimators have gained a lot of interest in the robust statistics field as they can combine high breakdown points and high computational efficiency. However, such robustness results highly depend on the choice of function ρ. For example, both the mean and the median are M-estimators, but as has been shown previously, the mean is not robust at all.

The second category includes strategies based on the rejection of outliers, more precisely trimmed and winsorized statistics. They are particular cases of L-estimators, meaning estimators based on a linear combination of order statistics (like quantiles).

Definition 1.3.6. Trimmed and winsorized statistics. *Let $\mathcal{Y}$ be a Euclidean space, $S_N \in \mathcal{Y}^N$ a dataset, $T : \mathcal{Y}^N \to \mathcal{Y}$ a statistic and $\alpha \in (0, 1)$. Let us denote by $q_\alpha(S_N)$ the α-quantile of dataset S_N.*

1) The α-trimmed statistic on S_N is defined by: $T_\alpha^{trim}(S_N) = T((S_N)_\alpha^{trim})$, where $(S_N)_\alpha^{trim} = \{y \in S_N \mid q_\alpha(S_N) \leq y \leq q_{1-\alpha}(S_N)\}$.

2) The α-winsorized statistic on S_N is defined by: $T_\alpha^{win}(S_N) = T((S_N)_\alpha^{win})$*, where* $(S_N)_\alpha^{win} = (S_N)_\alpha^{trim} \cup \{q_\alpha(S_N)\}^{\#\{y \in S_N \mid y < q_\alpha(S_N)\}} \cup \{q_{1-\alpha}(S_N)\}^{\#\{y \in S_N \mid y > q_{1-\alpha}(S_N)\}}$.

Simply put, the trimmed version of a statistic T consists in computing the same statistic on a dataset where the rightmost and leftmost data points have been removed; the winsorized version of the statistic T consists in computing the same statistic on a dataset where the rightmost and leftmost data points have been replaced by the closest acceptable value.

The last category includes the minimax approaches, and more precisely *Distributionally Robust Optimization* (DRO) problems.

Definition 1.3.7. DISTRIBUTIONALLY ROBUST OPTIMIZATION *Let $\mathcal{Y}$ be a measurable space, $P \in \mathcal{M}_+^1(\mathcal{Y})$ a probability distribution, and $m : \mathcal{M}_+^1 \times \mathcal{M}_+^1 \to \mathbb{R}$ a metric. The Distributionally Robust Optimization problem for distribution P of level ε consists in solving the following problem:*

$$T_{DRO}(P, \varepsilon) = \underset{y \in \mathcal{Y}}{\operatorname{argmin}} \max_{Q \mid m(P,Q) \leq \varepsilon} \mathbb{E}_{Y \sim Q}(l(Y, y)) \tag{1.3.9}$$

The DRO statistic thus focuses on optimizing the worst distribution in a set sufficiently close to the source distribution P, which is, optimizing an *adversarial* distribution. Notice how closely related it is to the definition of the breakdown point in Definition 1.3.3.

1.3.2 Main Challenges in Extending Robustness Techniques Against Poisoning Attacks for Ranking Data

The different concepts and results seen in Section 1.3.1 have been successfully applied to real numbered data, as illustrated by the diversity of works on this topic, mentioned in Section 1.2.1.

However, robustness has not been extensively studied whenever the data space is less convenient than the space of real-numbered data. This limitation in the number of works particularly applies to the space of ranking data, which accumulates two challenges: the lack of vector-space structure, and the combinatorial nature of the space.

A proper introduction to the ranking space will be provided in Chapter 2. This Section will only provide a high-level description of this space. The ranking space is the space of permutations over n items, *i.e.* the symmetric group $\mathfrak{S}_n$ of $\{1, ..., n\}$. A ranking is denoted by $\sigma \in \mathfrak{S}_n$ and represents the preference (of a user) over a set of n items. To illustrate this concept, let's give an example with morning drinks: a ranking over the set of items {'coffee', 'tea', 'orange juice'} would be an object σ representing the preference of the sentence 'I prefer orange juice over tea over coffee'.

The rankings, or *preference* data, are naturally used in recommender systems. With the explosion of recommender system-based applications using user preferences (advertisement, e-commerce with movies, music or books recommendation, dating applications, social media and traditional media, etc.). The study of such data has recently become of central interest. However, the nature of the ranking space is particular, and, as previously mentioned, challenging.

Lack of vector-space structure. The ranking space $\mathfrak{S}_n$ is not a vector space. As a quick reminder, a vector space is a set E equipped with two binary operations: the first one is an internal binary operation which is, among other properties, commutative; the second one is an external binary operation. The space of real numbered data $\mathbb{R}$, as well as the multivariate space $\mathbb{R}^n$ are two examples of vector spaces that are both equipped with the traditional sum $+$ and scalar multiplication $\times$ operations.

Thus, the ranking space cannot be equipped with these two operations. In fact, in addition to metrics, the ranking space can only be equipped with an internal binary operation that is usually denoted by $\circ$. The $\circ$ operation allows for the composition of rankings and is not commutative. This characteristic of the ranking space prevents the straightforward generalization of several concepts provided in Section 1.3.1, which can be divided into two categories:

- Generalization is not convenient: some concepts can, in fact, be adapted to the ranking space. It is, for example, the case for the notion of *mean* (as well as M-estimators in general). For real-numbered data, the mean can be defined as $\overline{x}_n = 1/n \sum_{i=1}^n x_i$, which necessitates the use of a (commutative) internal binary operation and an external binary operation. But, as previously mentioned, the mean can also be defined using a metric-based definition: $\overline{x}_n = \operatorname{argmin}_{x \in \mathbb{R}} \mathbb{E}_{X \sim S_N}((X - x)^2)$, which only requires the space to be equipped with a metric, which is possible for the ranking space as it is finite. The challenge here does not necessarily comes from *defining* the relevant concept for the ranking space, but rather in its *computation*, which is deeply linked to the challenge related to the combinatorial nature of the space.
- Generalization is difficult: some concepts cannot be adapted easily to the ranking space. This is, for example, the case of the notion of quantile and related statistics (the trimmed and winsorized ones in particular, and all L-estimators in general), which stems from the absence of the notion of *total order*. Such a notion is achievable for real-numbered data as it can be axiomatically defined using the two binary operations from a vector space, even though it is not the only way, and a vector space is not necessarily totally ordered (for example, the space of multivariate data $\mathbb{R}^n$ is partially ordered but not totally ordered). Thus, these ordered-based notions cannot be directly generalized to the ranking space (neither to the multivariate space), and some additional tools are needed to replicate such concepts, as is the case for *depth functions*, that will be addressed in Chapter 3.

Combinatorial nature of the space. The ranking space $\mathfrak{S}_n$ is finite, and of known cardinality: it has $n!$ elements. As it is a finite space, every defined concept can be theoretically computed in an exhaustive, brute-force manner.

For example, it is possible to compute the *mean* of a distribution over rankings P by computing $\mathbb{E}_{\Sigma \sim P}(l_2(\Sigma, \sigma))$, where $l_2(\sigma_1, \sigma_2) = \sum_{i=1}^n (\sigma_1(i) - \sigma_2(i))^2$, for all elements $\sigma \in \mathfrak{S}_n$, and find the ranking associated with the smallest loss. However, such a computation would require spanning the entire space of rankings of size $n!$, which is, in practice, unachievable even for relatively small values of n. As an example, $10! = 3628800$: let's say that the computation of the expected loss for one ranking takes 10^{-2} second (by looping over the $10!$ points of the theoretical distribution), it would take approximately 1 hours to finish

the computation (repeat the computation of the expected loss over the same 10! points), which is huge to 'just' compute a mean.

Considering that many applications using recommender systems deal with hundreds to millions of items, such a strategy is simply not feasible. It is thus necessary to develop methods that take this computational issue into account and are scalable, even if this requires additional hypotheses on the type of data distribution that can be processed.

Drawbacks of embedded representations of ranking data. A main line of works that circumvent the two limitations mentioned earlier (lack of vector space structure and combinatorial nature of the space) consists in embedding the rankings into a space that is simpler to study. This idea was prominently introduced and studied by Diaconis (1988, 1989) with the *spectral representation* of ranking data.

It provides a mathematical framework for analyzing and understanding the structure of rankings: in this representation, the rankings are transformed into a spectral space using the tools of Fourier analysis.

This representation of ranking data has significantly contributed to the understanding and analysis of rankings. However, such techniques come with limitations that we precisely want to avoid for studying robustness for rankings. In particular, the spectral representation condenses the ranking data into a lower-dimensional space, which can lead to a loss of detailed information. While it captures important global patterns, it may not fully capture the nuances and finer-grained characteristics of individual rankings. Additionally, the interpretability of the spectral components (eigenvalues and eigenvectors) may be challenging. Extracting meaningful insights and translating spectral information into actionable knowledge often requires careful analysis and domain expertise.

These limitations motivate our approach consisting of studying directly the ranking space, rather than an embedded representation. Note, however, that even though this thesis studies the robustness of ranking data by analyzing *complete* orderings over a set of items, meaning full rankings, most of our work can in fact be extended to *partial rankings*, where only some ordering between items are observed.

Discarding the 'embedding' approach, the two specific types of challenges for the ranking space explain the lack of work on the topic of robustness in this context. Part I of the present thesis will thus present my work to initiate the study of robustness for ranking data. The main contributions are developed in Section 1.3.3.

1.3.3 Main Contributions on Pioneering the Study of Robustness for Ranking Data

The contribution of the present thesis on the field of robustness against poisoning attacks focuses on overcoming the challenges mentioned in Section 1.3.2. More specifically, Part I will be dedicated to my contribution to this field, and will be organized as follows:

- Chapter 2 will introduce more formally the ranking space, as well as the task of *Consensus Ranking*, which is the equivalent of the location task using the specific terminology of the literature on rankings. This Chapter will go through the classical

approach to solving such a task without taking into account the issue of robustness, present some results focusing specifically on computational efficiency, and then introduce the vulnerability of the classical method to poisoning attacks. The first few works on the topic of robustness in ranking will also be addressed as an introduction to the problem.

- Chapter 3 will introduce an extension of the notion of total orders to the space of ranking through the scope of a center-outward ordering function. This object called a *depth function*, will be used to define analogs of quantiles, and thus a trimming strategy to robustify the classical *consensus ranking* statistics on the space of rankings. This strategy overcomes, in particular, the difficulty to define statistics based on *order* to the ranking space, as mentioned in Section 1.3.2.
- Chapter 4 will focus on the evaluation of the robustness (in the sense of the breakdown point), as well as the evaluation of the precision (in the sense of the loss), of all kinds of statistics for the consensus ranking task. This Chapter will additionally provide a very efficient plugin method to robustify any statistics, which overcomes the computational issue mentioned in Section 1.3.2. Finally, this Chapter will draw a comparison between several consensus ranking methods and show that the proposed plugin improves the robustness while not impairing the precision.

From a high-level perspective, the contribution of the present thesis is to initiate the study of robustness to poisoning attacks to the ranking space. To do so, as motivated in Section 1.3.1, the robustification of the location estimation task, namely *Consensus Ranking*, will be at the core of Part I. Consequently, we will not only provide two different robust statistics but also provide a way to evaluate and check the robustness of consensus ranking statistics. Therefore, the present thesis provides not only ready-to-use solutions to robustify the task of consensus ranking in an efficient way, but also the starting point to robustify many recommender systems-based related tasks. It also provides a way to thoroughly evaluate the robustness of any statistic, and thus facilitates the development of novel statistics and establishes a framework for comparing diverse approaches, thus serving as a cornerstone for the progression of future works.

1.4 Evasion Attacks and Adversarial Examples in Deep Learning: Notions, Challenges, and Contributions

The present Section introduces the concept of evasion attacks, which are the attacks led at *inference* time, after having trained a convenient model, when users can interact with the said model. Section 1.4.1 will provide the context and definition of evasion attacks, specifically on deep learning models, which has yielded great success and massive interest since 2013. Section 1.4.2 will present the current challenges identified by the literature on this topic, as well as some relevant results. Section 1.4.3 will introduce a high-level overview of the contribution of this thesis to this field.

1.4.1 Neural Networks under the Threat of Adversarial Examples as Evasion Attacks

Neural Network Classification Task. Before digging directly into the adversarial phenomenon, the main concepts related to neural networks for classification tasks (such as computer vision classification) will be recalled. Using the formalism introduced in Section 1.2.2, a neural network (NN) can be simply described as a parametric model: denoting $\mathcal{F}_\Theta : \mathcal{X} \to [\![1, ...K]\!]$ the parametric model class for a K-classification problem, a neural network can simply be described as a model $f_\theta \in \mathcal{F}_\Theta$. More specifically, a neural network consists of an interconnection of several *layers*, connected through linear operations and non-linear *activation functions*, to produce a vector of scores of size K, associated with each class of the problem. The predicted class is then chosen to be the one with the highest score. Importantly, a neural network can be identified with its *feature map*, which is the function outputting the aforementioned vector of scores of size K, also called the *logits vector*. A prominent example of neural networks subclasses is called *multilayer perceptron* and defined as follows:

Definition 1.4.1. MULTILAYER PERCEPTRON (MLP). *Let $\mathcal{F}_\Theta$ be a (parametric) model class. $f_\theta \in \mathcal{F}_\Theta$ is a multilayer perceptron with L layers if and only if:*

$$f_\theta(x) = \underset{k=1,...,K}{\operatorname{argmax}}\, g_\theta(x) \;\; \forall\, x \in \mathcal{X}, \;\; with \tag{1.4.1}$$

$$g_\theta(x) = W_L \sigma_{L_1}\left(W_{L-1}\sigma_{L-2}(...\sigma_1\left(W_1 x + b_1\right)) + ... + b_{L-1}\right) + b_L, \tag{1.4.2}$$

where $g_\theta : \mathcal{X} \to \mathbb{R}^K$ is the feature map, $\forall\, l \in [\![1, ..., L]\!], \sigma_l$ is the activation function (e.g. a ReLU function), and $\theta = (W_i, b_i)_{1 \leq i \leq L}$ are the parameters.

The feature map g_θ thus outputs a vector of size K. To transform this vector of scores into a probability vector, the softmax function is usually used to define $\tilde{g}_\theta(x) = softmax(g_\theta(x))$, where:

$$\forall\, z \in \mathbb{R}^K, \;\; softmax(z) = \left(\frac{\exp(z_k)}{\sum_{i=1}^K \exp(z_i)}\right)_{1 \leq k \leq K} \tag{1.4.3}$$

Similarly to the feature map g_θ, the probability vector $\tilde{g}_\theta$ can also be identified with the neural network's prediction $f_\theta(x)$, since $f_\theta(x) = \operatorname{argmax}_{k=1,...,K} g_\theta(x) = \operatorname{argmax}_{k=1,...,K} \tilde{g}_\theta(x)$.

In the context of supervised learning, the training phase of neural networks consists in optimizing its parameters θ, in order to achieve the best results possible for the task at hand. Informally, as there is only one 'correct' class for an input image, the goal is to have all the images assigned to their corresponding correct class: this is the purpose of the so-called *0-1 loss*. However, since this loss is not differentiable, it is not possible to directly optimize it, and so the training process replaces the 0-1 loss with a smoother loss to perform the training, which is usually obtained via *stochastic gradient descent (SGD)*, even though other methods exists. More specifically:

Definition 1.4.2. NEURAL NETWORKS OPTIMIZATION PROBLEM. *Let $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ be a distribution, $f_\theta \in \mathcal{F}_\Theta$ be a neural network on a K-classification problem and $\tilde{g}_\theta$ its corresponding probability vector function. Let $\phi : \Delta^{\mathcal{Y}} \times \mathcal{Y} \to \{0, 1\}$ be the 0-1 loss, where $\phi(p, y) = 0$ if $\operatorname{argmax}_{k \in [\![1,...K]\!]} p(k) = y$, and 0 else. The optimization goal of the neural network is to find $f_{\theta^\star}$ where:*

$$\theta^\star = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \Phi(\theta, P_{X,Y}), \tag{1.4.4}$$

with $\Phi(\theta, P_{X,Y}) := \mathbb{E}_{X,Y\sim P_{X,Y}}(\phi(\tilde{g}_\theta(X), Y))$.

As previously mentioned, this optimization problem is intractable in practice. Fortunately, as shown in Bartlett et al. (2006), smoother losses, called *consistent surrogate* losses, can be used to approximate efficiently the 0-1 loss. One of their results is simplified here:

Theorem 1.4.3. CONVERGENCE OF RISKS FOR CONSISTENT SURROGATE LOSSES. *Let f_θ be a neural network on a binary classification problem, associated with its feature map g_θ and probability vector $\tilde{g}_\theta$. Let $\phi : \Delta^{\{-1,+1\}} \times \pm 1 \to \{0, 1\}$ be the 0-1 loss and $l : \Delta^{\{-1,+1\}} \times \{-1, +1\} \to \mathbb{R}^+$ a loss. Suppose that l is a consistent surrogate loss, meaning it satisfies some constraint that will not be restated here. Then for every sequence $(\theta_i)_{i\geq 0}$ and probability distribution $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X} \times \{-1, +1\})$, we have*

$$\begin{aligned} &\mathbb{E}_{X,Y\sim P_{X,Y}}(l(g_{\theta_i}(X), Y)) \underset{i\to+\infty}{\longrightarrow} \inf_{\theta\in\Theta} \mathbb{E}_{X,Y\sim P_{X,Y}}(l(g_\theta(X), Y)) \Rightarrow \\ &\mathbb{E}_{X,Y\sim P_{X,Y}}(\phi(\tilde{g}_{\theta_i}(X), Y)) \underset{i\to+\infty}{\longrightarrow} \inf_{\theta\in\Theta} \mathbb{E}_{X,Y\sim P_{X,Y}}(\phi(\tilde{g}_\theta(X), Y)) \end{aligned} \tag{1.4.5}$$

Theorem 1.4.3 means that optimizing over a consistent surrogate loss l boils down to optimizing over the 0-1 loss. This is the reason why neural networks can be trained using a surrogate loss l.

Definition 1.4.4. NEURAL NETWORKS TRAINING. *Let $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ be a distribution, $f_\theta \in \mathcal{F}_\Theta$ be a neural network on a K-classification problem and g_θ its corresponding feature map. Let $l : \Delta^{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}^+$ be a surrogate loss. The optimization goal of the neural network is to find $f_{\theta^\star}$ where:*

$$\theta^\star = \underset{\theta\in\Theta}{\operatorname{argmin}}\, L(\theta, P_{X,Y}), \qquad \text{with} \quad L(\theta, P_{X,Y}) := \mathbb{E}_{X,Y\sim P_{X,Y}}(l(g_\theta(X), Y)). \tag{1.4.6}$$

This optimization goal is traditionally achieved through stochastic gradient descent (SGD), meaning that the parameters θ are optimized step by step following the opposite direction of the gradient of the loss with respect to parameters θ.

The optimization problem objective described in Definition 1.4.2 and the training procedure defined in Definition 1.4.4 thus aims at obtaining an *accurate* neural network, meaning a small expected 0-1 loss on the distribution $P_{X,Y}$, or, alternatively, a high accuracy, as defined by:

Definition 1.4.5. ACCURACY. *Let $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ be a distribution and $f_\theta \in \mathcal{F}_\Theta$ be a neural network on a K-classification problem. The accuracy of f_θ on $P_{X,Y}$ is defined as:*

$$Acc(f_\theta, P_{X,Y}) = \mathbb{E}_{X,Y\sim P_{X,Y}}(\mathbb{1}[f_\theta(X) = Y]) \tag{1.4.7}$$

A simple example of the computation of the accuracy is provided in Figure 1.4 as an illustration.

Neural networks are popular in many fields and specifically in computer vision classification because they are the class of models achieving the highest accuracy on several complex datasets (for example, the ImageNet dataset Deng et al. (2009)), sometimes even better than human classification, as shown in Geirhos et al. (2017). However, the performance of neural networks on perturbed data has not been studied until recently.

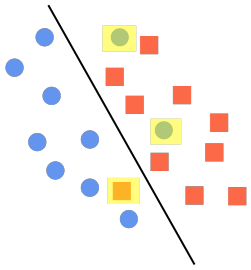


Figure 1.4: Computation of the accuracy: 3 points, highlighted in yellow, are wrongly classified by the model, out of 20 points. Therefore, the accuracy of the model on this dataset is 17/20.

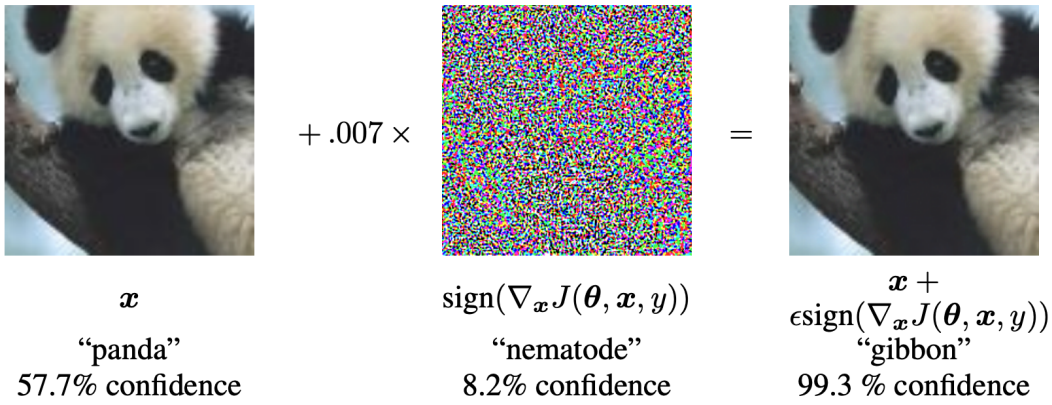


Figure 1.5: Illustration of an adversarial example creation and its (incorrect) classification. Courtesy of Goodfellow et al. (2014).

Theory of Adversarial Example and the Robust Optimization Problem. In Szegedy et al. (2013), the authors unveiled the concept of *adversarial examples* in the context of deep learning for computer vision classification. In that field, adversarial examples are clean images on which a malevolent perturbation has been added, that cannot be detected by human eyes but, surprisingly, fool state-of-the-art deep learning classification models: this coincides, in that case, to the notion of *evasion attack*.

From a high-level perspective, adversarial examples have surprised the community of deep learning researchers because 1) they are able to fool models that are very good at classification tasks and are able to generalize efficiently to unseen clean images, and 2) the magnitude of perturbation needed to fool a model is so small that the difference between a clean image and its adversarial counterpart is usually unnoticeable to the human eye.

Figure 1.5, extracted from the follow-up work Goodfellow et al. (2014), shows an instance of an adversarial example crafted from a clean image of a panda: the resulting adversarial image does not look different from the clean one by the human eye, yet is wrongly classified as a gibbon by the neural network.

Since these seminal works, the adversarial example phenomenon has gained a huge interest in the community, and the literature on the subject has become very large. This amount of work has enabled the community not only to better understand mathematical limits to the robustness of neural networks, typologies of adversarial examples, and characteristics of such examples, but also to propose more and more efficient attack algorithms, defense, and detection methods. Recent works have been developed to structure the knowledge

around adversarial examples with surveys on the topic, with for example Chakraborty et al. (2022); Akhtar and Mian (2018); Chen et al. (2020b); Han et al. (2023); Cabral Costa et al. (2023). Here, the main ideas and notions from the literature will be introduced.

In theory, a perfect adversarial attack is a function that finds the perfect adversarial counterpart of a clean input $x \in \mathcal{X}$, which is the closest input to x that is classified differently from x. Formally, this is defined as follows.

Definition 1.4.6. PERFECT ADVERSARIAL ATTACK. *Let f_θ be a neural network model and $||.||$ a norm on $\mathcal{X}$. The perfect adversarial attack, denoted by $A : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, is defined by:*

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad A(x, y) = \underset{x' \in \mathcal{X}}{\operatorname{argmin}} ||x - x'||, \quad \text{such that } f_\theta(x') \neq f_\theta(x). \tag{1.4.8}$$

The adversarial example corresponding to $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is then denoted by $x^{adv} = A(x, y)$.

Remark 1.4.7. *The perfect adversarial attack A from Definition 1.4.6 is made dependent on the class label y because practical adversarial attacks do depend on y.*

With the Definition 1.4.6, it is possible that for some inputs (x, y), the perfect adversarial examples are far away from them, meaning that $||x - x^{adv}||$ is large. To get better control of the size of the perturbation, this definition can be equivalently formulated as the following dual problem.

Definition 1.4.8. PERFECT ADVERSARIAL ATTACK - DUAL VERSION. *Let f_θ be a neural network model, $\tilde{g}_\theta$ its corresponding probability vector function, ϕ the 0-1 loss and $\Phi(\theta, P_{X,Y}) = \mathbb{E}_{X,Y \sim P_{X,Y}}(\phi(\tilde{g}_\theta, Y))$ the $0-1$ loss of the neural network on $P_{X,Y}$ and $||.||'$ a norm on $\mathcal{X} \times \mathcal{Y} \to \mathcal{X}$. Let $T_A : (x, y) \in \mathcal{X} \times \mathcal{Y} \mapsto A(x, y), y \in \mathcal{X} \times \mathcal{Y}$.*

The perfect adversarial attack A can be characterized:

$$A = \underset{A'}{\operatorname{argmax}}\, \Phi(\theta, T_{A'} \# P_{X,Y}) \quad \text{such that } \; ||A' - Id||' \leq \tilde{\epsilon}, \tag{1.4.9}$$

for some constant $\tilde{\epsilon}$, where Id is the identity function and $T_{A'} \# P_{X,Y}$ the pushforward distribution of $P_{X,Y}$ by $T_{A'}$.

These adversarial attack formulations explore the attacker's point of view on the more general two-player game whose objective is to train a robust neural network. This generic problem is at the core of the quest for the robustification of neural networks against adversarial examples and can be theoretically formulated as follows.

Definition 1.4.9. ROBUST NEURAL NETWORKS OPTIMIZATION PROBLEM. *Let $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ be a distribution, $f_\theta \in \mathcal{F}_\Theta$ be a neural network on a K-classification problem and $\tilde{g}_\theta$ its corresponding probability vector function.*

Let $\phi : \Delta^{\mathcal{Y}} \times \mathcal{Y} \to \{0, 1\}$ be the 0-1 loss, $||.||'$ a norm on $\mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, $T_A : (x, y) \in \mathcal{X} \times \mathcal{Y} \mapsto A(x, y), y \in \mathcal{X} \times \mathcal{Y}$ and $\tilde{\varepsilon} \in [0, 1]$.

The robust optimization goal of the neural network is to find $f_{\theta^\star}$ where:

$$\begin{aligned} \theta^\star &= \underset{\theta \in \Theta}{\operatorname{argmin}} \max_{A : ||A - Id||' \leq \tilde{\varepsilon}} \Phi(\theta, T_{A'} \# P_{X,Y}), \quad \text{with} \\ \Phi(\theta, T_{A'} \# P_{X,Y}) &:= \mathbb{E}_{X,Y \sim T_{A'} \# P_{X,Y}}(\phi(\tilde{g}_\theta(X), Y)). \end{aligned} \tag{1.4.10}$$

Adversarial Examples in practice. The problem raised by Definition 1.4.6 or Definition 1.4.8 incorporated in the broader problem defined by Definition 1.4.9 is a very difficult problem to study in general. To overcome this issue, the field has evolved either to modify Equation (1.4.8) in the definition to solve a simpler problem (this is the case for the adversarial attack method called L-BFGS [1], Szegedy et al. (2013)), or to propose heuristics to craft adversarial examples to get $\Phi(\theta, T_A \# P_{X,Y})$ high, similarly to Equation (1.4.9) (this is the case for the adversarial attack method called FGMS, Goodfellow et al. (2014)). In both cases, a practical adversarial example thus results to be a perturbed version of a clean input, with a controlled perturbation size, that aims to fool the neural network. The adversarial attack A may, in such cases, vary, but for simplicity, the literature has focused on additive attacks. Formally, this is defined as follows.

Definition 1.4.10. ADVERSARIAL ATTACK. *Let f_θ be a neural network, and $||.||$ a norm on $\mathcal{X}$. Let $\varepsilon \in [0,1]$ be the perturbation budget. An ε-practical adversarial attack is a function $A_\varepsilon : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, defined by:*

$$\forall\, (x,y) \in \mathcal{X} \times \mathcal{Y},\ A_\varepsilon(x,y) = x + \delta_\varepsilon(x,y) \quad \text{such that } ||\delta_\varepsilon(x,y)|| \leq \varepsilon \\ \text{with } f_\theta(A_\varepsilon(x,y)) \neq f_\theta(x) \text{ as often as possible.} \tag{1.4.11}$$

The adversarial example corresponding to $(x,y) \in \mathcal{X} \times \mathcal{Y}$ is generally denoted by $x^{adv} = A_\varepsilon(x,y)$.

Usually, $\delta(x,y)$ will be simply denoted by δ, and the norm used is in general the L_1, L_2 or L_∞ norm. This practical definition is thus at the core of the study of the adversarial phenomenon: this is the one this thesis will refer to when mentioning 'adversarial examples', without any additional specification. The most famous attack method and one of the first since it was introduced by Goodfellow et al. (2014), is called the *Fast Gradient Sign Method*, or FGSM. It is defined as follows.

Definition 1.4.11. FGSM ATTACK. *Let f_θ be a (trained) neural network, g_θ its corresponding feature map, $l : \Delta^{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}^+$ its training loss, and $\varepsilon \in [0,1]$ the perturbation budget. The FGSM attack, denoted by $FGSM_\varepsilon$, is defined as follows:*

$$\forall\, (x,y) \in \mathcal{X} \times \mathcal{Y},\ x_\varepsilon^{adv} = FGSM_\varepsilon(x,y) := x + \varepsilon\, sign\left(\nabla_x l(g_\theta(x), y)\right), \tag{1.4.12}$$

where $\nabla_x l(g_\theta(x), y)$ denotes the gradient of l in x.

The FGSM attack method has laid the groundwork and provided a foundational model for the development of the subsequent attack algorithms. Its tremendous success lies in its simplicity associated with its success rate. More precisely, the FGSM attack is very simple and approximate: the only deviation from the clean input it introduces is just the addition of subtraction of a fixed constant to all the pixel values of an image. Moreover, as explained in Goodfellow et al. (2014), when $\varepsilon = 0.25$, the *adversarial accuracy* (the accuracy computed on adversarial examples only) of a multilayer perceptron trained on the dataset ImageNet is reduced to 0.1%, which is impressive for such a simple method.

[1] The attack method is based on the optimization algorithm with the same name

1.4.2 Main Challenges in Exploring the Complexity of the Adversarial Phenomenon in Deep Learning.

The field of *adversarial robustness* in the context of deep learning for computer vision classification is quite recent, since the phenomenon was unveiled in 2013. However, such a tremendous breach in the efficiency and security of deep learning algorithms has attracted a lot of effort and attention from the research community to better understand and treat the phenomenon. To give a proper illustration of the explosion of the field, here are some statistics: one paper, Szegedy et al. (2013), was published on the subject in 2013; 4 papers in 2014; 15 in 2015; 42 in 2016; 501 in 2018; 1221 in 2020 and 1949 in 2022, thanks to the consolidation work from Nicolas Carlini to enumerate all the papers related to the field, which can be found on his blog at the following address: https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html.

Thus, many findings about this phenomenon have been unveiled by the community in recent years, and many are still largely open to debate. A proper introduction to some of these papers and to the main concepts and findings on adversarial examples will be provided in Chapter 6. However, this Section will provide a high-level overview of the two main challenges in the field.

Indeed, the open questions that remain in the field of adversarial robustness can be aggregated into two main categories of challenges. The first challenge is centered around the important question *'Why do adversarial examples exist in deep learning?'*. As neural networks provide great results in the field of computer vision classification, and since they are able to generalize efficiently to unseen images, the question remains to know what is so specific to adversarial examples that this generalization ability completely fails. The second challenge is to understand under which conditions adversarial examples are inevitable, meaning that neural networks will remain vulnerable. Such a question is obviously very important for security reasons.

Lack of proper understanding of the adversarial phenomenon. Many works have focused on trying to better understand what makes adversarial examples succeed, and more generally studying the characteristics of adversarial examples. As an illustration, a paper as seminal as Goodfellow et al. (2014) has provided the so-called 'linear or linearity' hypothesis. This hypothesis aims at providing intuition and an assumption to explain the adversarial phenomenon. The challenge here resides in the fact that it is very hard to study theoretical neural networks, so a vast majority of works rely on either experimental methodologies or theoretical works on simplified versions of neural networks. In both cases, the discoveries on these subjects can rely on an accumulation of pieces of evidence, and, of course, all papers do not agree on the same findings. As the adversarial robustness field is still quite recent, still numerous hypotheses have not yet been explored and meta-analyses are rarely available. Thus, the problem of understanding why adversarial examples are so efficient on neural networks is still largely open, even though many hypotheses have been proposed to explain it, as will be detailed in Section 6.2.

Potential intrinsic vulnerability of Neural Networks. A line of works has focused on trying to explore under which conditions neural networks are, inevitably, vulnerable. These works provide a theoretical analysis of neural networks and, in general, bounds on the

success (or failure) of adversarial examples such as in Fawzi et al. (2018b,a); Mahloujifar et al. (2019); Bubeck et al. (2019); Dohmatob (2019); Ford et al. (2019); Melamed et al. (2023). These works are inherently limited by the hypotheses on the neural networks, the data distribution, or the adversarial attack methods. With the development of more and more sophisticated adversarial attack methods, such lines of work must remain up-to-date with the most recent heuristics advances of attacks. Very recently, an important shift has modified the development of adversarial examples, with the finding of so-called *universal attacks* Moosavi-Dezfooli et al. (2017) and *low-dimensional attacks*, as in Guo et al. (2018a); Huang and Zhang (2019); Yan et al. (2019); Tu et al. (2019); Chen et al. (2020a). These attacks basically focus on modifying only a small subspace of the input features, contrary to more classical attacks that are conditioned only on an overall budget. To give a simple example of a low-dimensional adversarial attack, modifying a unique pixel for all images targeted by the attack is a relevant strategy that has been explored in Su et al. (2019). Such adversarial examples do not operate on the full dimensionality of the data, and thus the proofs' techniques traditionally used in the field, which mostly relied on the curse of dimensionality, cannot be used anymore.

These two challenges will be tackled separately in Part II, and the contributions of the thesis are detailed in Section 1.4.3.

1.4.3 Main Contributions in Understanding and Unifying Recent Advances on Adversarial Robustness

The contribution of the present thesis on the field of robustness against evasion attacks in the context of deep learning for image classification thus focuses on overcoming the challenges mentioned in Section 1.4.2. More specifically, Part II will tackle this field and will be organized as follows:

- Chapter 6 will provide an in-depth introduction to the adversarial robustness field, and present more precisely the current state of the research on the subject. The main current attack methods, defense algorithms, and detection strategies will be presented, as well as their results on traditional image classification tasks. Moreover, the theoretical findings on adversarial attacks will be summarized to provide an overview of the proof strategies and the limits of such works. Finally, the findings on the characteristics of adversarial examples will be discussed in depth.
- Chapter 7 will provide the study of a hypothesis to explain the adversarial phenomenon, which takes into account both the main unveiled characteristics of adversarial examples and the characteristics and architecture of neural networks. The interaction of these two aspects leads to the hypothesis that an important reason for the vulnerability of neural networks resides in their over-parametrization. This hypothesis will be studied experimentally thanks to topological tools and theoretically grounded, and an efficient detection method will be built upon these findings.
- Chapter 8 will provide a theoretical analysis of the very recent heuristics of universal and low-dimensional attacks, using original proof strategies to provide bounds on the success rate of such attacks under general conditions. This work enables ground the aforementioned heuristics and provides a theoretical argument to advocate for their wide sprayed adoption in the community.

Part I

Pioneering the Study of Robustness for Ranking Data

Table of Contents

List of Notations

Ranking Objects

$\mathfrak{S}_n$	The symmetric group on $\{1, ..., n\}$
Π_n	The space of bucket rankings on $\{1, ..., n\}$
$\sigma, \nu \in \mathfrak{S}_n$	Usual notations for rankings
π	Usual notation for a bucket ranking
Σ	A ranking random variable
$\tau_{i,j}$	A transposition between items i and j
$D_P : \mathfrak{S}_n \to \mathbb{R}_+$	Ranking depth functions based on distribution P

Metrics

d, l	Usual notations for metrics on rankings
m	Usual notation for a metric on ranking distributions
d_τ	Kendall Tau distance
d_1	Spearman's Footrule distance
d_2	Spearman's Rho distance

Stochastic Objects

P	Usual notation for a distribution on rankings
$\hat{P}_N$	Empirical distribution of the dataset S_N drawn from P
$p_{i,j}$	Usual notation for the pairwise probability between items i and j
$PL(w)$	Plackett-Luce distribution of parameters $w \in \mathbb{R}^n$
$M(\sigma_0, \theta)$	Mallows distribution of parameters σ_0 (center) and θ (scale)
δ_a	Dirac distribution in a

Generic Objects

$[k]$	The set $\{1, ..., k\}$ of integers from 1 to k
$\#E$	Cardinality of set E
$\mathbb{1}(\mathcal{E})$	Indicator function of event $\mathcal{E}$

Chapter 2

Introduction to Rankings

The best argument against democracy is a five-minute conversation with the average voter.

Winston Churchill.

Contents

As explained in Section 1.3, our study of poisoning attacks will focus on the location estimation task of ranking data. In this Chapter, the most important notions for such a task with such data will be clarified, and some results on the task at hand will also be addressed.

2.1 Fundamentals of Ranking Data and Distributions

The space of rankings is, as mentioned in Section 1.3.2, of peculiar nature. This specificity explains in part the different notations that exist in the literature and the different objects that can be used in this context. This Section will thus not only introduce rankings but also clarify the notation used in the rest of the thesis.

2.1.1 Basic Definitions for Rankings

The symmetric group over a set X, denoted by $\mathfrak{S}_X$, is the space of permutations over X. Mathematically, a *permutation* $\sigma \in \mathfrak{S}_X$ is a bijective function from X to X, meaning a rearrangement of this set. Using the same example as Section 1.3.2, let's suppose that $X = \{\text{'coffee'}, \text{'tea'}, \text{'orange juice'}\}$, then an example of permutation $\sigma \in \mathfrak{S}_X$ can be defined by $\sigma(\text{'coffee'}) = \text{'orange juice'}$, $\sigma(\text{'tea'}) = \text{'coffee'}$, $\sigma(\text{'orange juice'}) = \text{'tea'}$.

In the context of recommendation applications, permutations are interesting when they are *rankings*, meaning when the space X is $\{1, ..., n\}$. In that case, the set X is simply the set of n items $\{1, ..., n\}$ (and the ranking space is denoted by $\mathfrak{S}_n$), which is useful to consider the image space as *ranks*. A ranking is thus a bijective function that takes as input an *item* and outputs its *rank*. This is denoted by $\sigma(i) = r$, where $i \in [n]$ usually denotes an item and $r \in [n]$ a rank. Note that the literature is quite divided about this formalism, and some works prefer to use $\sigma(r) = i$: throughout this thesis, a ranking will always be a function that assigns a rank to an item, *i.e.* $\sigma(i) = r$.

Going back to the previous example, the morning drinks can therefore be assigned to a number. For example, 'coffee'=1, 'tea'=2, 'orange juice'=3. Then, an example of ranking $\sigma \in \mathfrak{S}_n$ can be given by $\sigma(1) = 3$, $\sigma(2) = 2$ and $\sigma(3) = 1$. This specific ranking thus considers that the rank of 'coffee' is 3, and in general that 'orange juice' is better than 'tea', which is better than 'coffee'.

To simplify this description of a ranking over a finite set, and considering that a ranking gives an *order* between the items, we can use the following, simpler notation: $\sigma = 3 \succ 2 \succ 1$, where $\succ$ denotes that an item is preferred over another.

A ranking can thus be considered at the same time as a bijective function and as a strict total order. We can thus formally define a ranking as follows:

Definition 2.1.1. RANKING. *A ranking $\sigma \in \mathfrak{S}_n$ is:*

1) A bijective function from $[n]$ to $[n]$ that takes as input an item $i \in [n]$ and outputs its rank $r \in [n]$.

2) A strict total order, meaning a sequence of elements $(\sigma^{-1}(1), ..., \sigma^{-1}(n))$ such that $i \succ_\sigma j \Leftrightarrow \sigma(i) < \sigma(j)$. Usually, $\succ_\sigma$ will be denoted as $\succ$ whenever the context is clear.

As previously mentioned in Section 1.3.2, the space of rankings $\mathfrak{S}_n$ is equipped with an internal (non-commutative) binary operation. This operation, denoted by $\circ$, allows for the composition of rankings $\sigma \circ \nu$, for any $\sigma, \nu \in \mathfrak{S}_n$. This composition maps an element $k \in [n]$ to the value $\sigma(\nu(k))$. In the present thesis, the composition of rankings will be relevant, in fact, only to swap adjacent items thanks to the composition of a ranking σ with a *transposition* $\tau \in \mathfrak{S}_n$. Informally, a transposition is a ranking that is the identity function, except on two elements.

Definition 2.1.2. TRANSPOSITION. *A transposition $\tau \in \mathfrak{S}_n$ is a ranking satisfying: $\exists\, i, j \in [n]$ with $i \neq j$ such that $\forall k \neq i, j,\ \tau(k) = k$, and $\tau(i) = j, \tau(j) = i$. In that case, the transposed items are i and j.*

Usually, a transposition of the items i and j will be simply denoted by $\tau_{i,j}$.

Then, the ranking ν created by the composition of a ranking σ with a transposition $\tau_{i,j}$, meaning $\nu = \sigma \circ \tau_{i,j}$, is the same ranking as σ except that the rank of item i is now the rank of item j and vice versa. As an example, consider the ranking $\sigma = 3 \succ 4 \succ 1 \succ 2$ and the transposition $\tau_{1,3}$. Then we have $\sigma \circ \tau_{1,3} = 1 \succ 4 \succ 3 \succ 2$.

The most important use case of the composition of a ranking with a transposition is, as previously mentioned, to swap adjacent items. Two items i and j are said to be adjacent (by ranking σ) if $\sigma(i) = \sigma(j) \pm 1$. Thus, a ranking ν that coincides with σ except on the adjacent items i and j can be defined by $\nu = \sigma \circ \tau_{i,j}$. When only the rank r of one of the items is known, it can be written $\nu = \tau_{r,r\pm1} = \sigma \circ \tau_{\sigma^{-1}(r),\sigma^{-1}(r\pm1)}$.

2.1.2 Metrics for Rankings

The ranking space $\mathfrak{S}_n$ is, as a finite space, metrizable. In the literature, several distances have been defined and used to evaluate all sorts of results in the field. The literature on this topic is quite active, going from classical metrics to much newer ones, see Järvelin and Kekäläinen (2000); Yilmaz et al. (2008); Carterette (2009); Kumar and Vassilvitskii (2010), that rely not only on rankings but also on specific features that vary depending on the specific task being addressed.

As the thesis focuses on the simple but core task of location estimation, as introduced in Section 1.3.1, the relevant metrics are mainly the most classical ones. Among them, the most relevant ones are the following:

Definition 2.1.3. KENDALL TAU DISTANCE. *The Kendall Tau distance, denoted as $d_\tau : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{N}$ is defined as:*

$$\forall \sigma_1, \sigma_2 \in \mathfrak{S}_n, \quad d_\tau(\sigma_1, \sigma_2) = \sum_{i<j} \mathbb{1}[(\sigma_1(i) - \sigma_1(j))(\sigma_2(i) - \sigma_2(j)) < 0], \tag{2.1.1}$$

Kendall Tau distance is the main distance used in various fields of rankings, including ranking data analysis and social science, thanks to its well-established properties. It counts the number of pairwise disagreements between the two rankings σ_1 and σ_2, as illustrated by Figure 2.1: for example, suppose that $\sigma_1 = 1 \succ 2 \succ 3$ and $\sigma_2 = 1 \succ 3 \succ 2$, then $d_\tau(\sigma_1, \sigma_2) = 1$ because the rankings disagree on the pairs $(2, 3)$ but not on the pairs $(1, 2), (1, 3)$.

Figure 2.1: Illustration of the computation of Kendall tau distance. σ_1 and σ_2 agree that $1 \succ 2$ and that $1 \succ 3$, but disagrees on items 2 and 3, because σ_1 orders $2 \succ 3$ and σ_2 orders $3 \succ 2$. Therefore, the Kendall Tau distance between σ_1 and σ_2 is $d_\tau(\sigma_1, \sigma_2) = 1$.

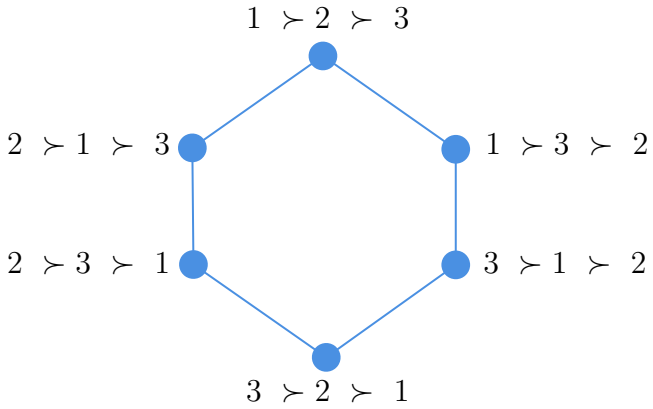


Figure 2.2: Visualization of $\mathfrak{S}_3$. Rankings linked by an edge are at distance 1 by Kendall Tau.

The distance is lower-bounded by 0 and upper-bounded by $n(n-1)/2$. An important remark to be made is that two rankings that are at distance 1 of each other according to Kendall Tau distance are *neighbors* because one can be obtained from the other by just swapping the two adjacent items on which they disagree. Equivalently, this means that if $d_\tau(\sigma, \nu) = 1$, then $\exists\, (i,j) \in [n]^2, i \neq j$ such that $\nu = \sigma \circ \tau_{i,j}$. This allows for convenient visualization of the ranking space, as illustrated by the case when $n = 3$ in Figure 2.2.

Definition 2.1.4. SPEARMAN'S FOOTRULE DISTANCE. *The Spearman's Footrule distance, denoted as* $d_1 : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{N}$ *is defined as:*

$$\forall \sigma_1, \sigma_2 \in \mathfrak{S}_n, \quad d_1(\sigma_1, \sigma_2) = \sum_{i=1}^{n} |\sigma_1(i) - \sigma_2(i)|, \tag{2.1.2}$$

Spearman's Footrule distance is the equivalent of the L_1-norm distance for the rankings. It is lower-bounded by 0 and upper-bounded by $n^2/2$ if n is even and $(n-1)(n+1)/2$ if n is odd.

Definition 2.1.5. SPEARMAN'S RHO DISTANCE. *The Spearman's Rho distance, denoted as* $d_2 : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ *is defined as:*

$$\forall \sigma_1, \sigma_2 \in \mathfrak{S}_n, \quad d_2(\sigma_1, \sigma_2) = \left(\sum_{i=1}^{n} (\sigma_1(i) - \sigma_2(i))^2 \right)^{1/2}, \tag{2.1.3}$$

Spearman's Rho distance is the equivalent of the L_2-norm distance for the rankings. It is lower-bounded by 0 and upper-bounded by $(n-1)n(n+1)/3$.

All the aforementioned distances, when used in the thesis, will be normalized to ease the comparisons. However, this short list is, of course, non-exhaustive: other distances, like

the Hamming distance, the Cayley distance, or others in Bachmaier et al. (2015), can also be considered, but the three aforementioned distances remain the most used ones. Each of the distances exhibits features that make them useful for specific tasks. For example, the Kendall Tau distance has a (naive) complexity of $\mathcal{O}(n^2)$, contrary to the two Spearman's distances which have a complexity of $\mathcal{O}(n)$. But the Kendall Tau distance considers the relative ordering of the items rather than specific values.

2.1.3 Classical Ranking Distributions

Probabilistic ranking models are an efficient tool to facilitate the development of statistical models and to analyze ranking data, which have been studied in depth by the literature, like Thurstone (1927, 1931) which introduced distributions based on ordering Gaussian vectors or Bradley and Terry (1952); Luce (1959); Plackett (1975); Mallows (1957a) which studied different variants of exponential family distributions.

The Mallows model, which is the most famous one, is a distance-based model defined with respect to a central ranking. More specifically:

Definition 2.1.6. MALLOWS MODEL. *Let $\sigma_0 \in \mathfrak{S}_n$ be a central ranking, $\theta > 0$ a dispersion parameter and d_τ Kendall Tau distance. The probability distribution $P \sim M(\sigma_0, \theta)$ defined by:*

$$\forall \sigma \in \mathfrak{S}_n,\ P(\sigma) = \frac{1}{\Psi(\theta, d_\tau)} e^{-\theta d_\tau(\sigma_0, \sigma)} \tag{2.1.4}$$

is referred to as the Mallows model (or distribution) of center σ_0 and dispersion parameter θ, where $\Psi(\theta, d_\tau)$ is the normalization constant.

Another famous model is the Plackett-Luce model:

Definition 2.1.7. PLACKETT-LUCE MODEL. *Let $w = (w_1, ..., w_n) \in \mathbb{R}_+^n$ be a vector of parameters. The probability distribution $P \sim PL(w)$ defined by:*

$$\forall \sigma \in \mathfrak{S}_n,\ P(\sigma) = \prod_{r=1}^{n} \frac{w_{\sigma^{-1}(r)}}{\sum_{p=r}^{n} w_{\sigma^{-1}(p)}} \tag{2.1.5}$$

is referred to as the Plackett-Luce model (or distribution) of parameters w.

The Plackett-Luce distribution has become quite popular in the literature thanks to two different characteristics:

Remark 2.1.8. *The Plackett-Luce distribution enables a very fast and easy pairwise comparison of items because it satisfies the following property: if $P \sim PL(w)$, we have $P(\Sigma(i) < \Sigma(j)) = w_i/(w_i + w_j)$.*

Remark 2.1.9. *The Plackett-Luce distribution is easy to simulate numerically using the Gumbel trick, as it satisfies the following property: if $G \sim Gumbel(0, 1; n)$ is a random vector of size n whose elements are independent standard Gumbel variables, then* $\text{argsort}(G + \log(w)) \sim PL(w)$.

See for example Kool et al. (2019) for more details about this trick.

2.2 Consensus Ranking

In the ranking literature, the location estimation task is usually referred to as *Ranking Aggregation*, or *Consensus Ranking*. Usually, the location estimate is called the *consensus*. The first works studying this problem trace back to social choice theory with, for example, Condorcet (1785). This Section summarizes the main methods to solve this task as explored by the literature.

2.2.1 Kemeny's Consensus and Other Classical Methods

The main approach to solving the Consensus Ranking problem is metric-based and solves a simple optimization problem. More specifically, it is defined as follows.

Definition 2.2.1. Classical Consensus Statistics. *Let $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a distance on rankings. A classical consensus statistics is a function $T_l : \mathcal{M}_+^1(\mathfrak{S}_n) \to \mathfrak{S}_n$ solving the following optimization problem: $\forall P \in \mathcal{M}_+^1(\mathfrak{S}_n)$,*

$$T_l(P) \in \underset{\sigma \in \mathfrak{S}_n}{\operatorname{argmin}} \, \mathbb{E}_{\Sigma \sim P}(l(\Sigma, \sigma)), \tag{2.2.1}$$

The output of statistics T_l is usually denoted by $\sigma_l^\star$ (where the dependence in P is dropped when the context is clear) and is simply called the consensus.

In particular, when the distance l chosen is Kendall Tau, meaning $l = d_\tau$, then the problem defined by Definition 2.2.1 is called *Kemeny's aggregation*, Kemeny's statistics is thus denoted by T_{d_τ} and the solution $\sigma_{d_\tau}^\star$ is called *Kemeny's consensus*.

Kemeny's aggregation method, based on Kendall Tau distance, is certainly the most popular choice to solve the Consensus Ranking task, even though it has the following major drawback: computing the consensus from an empirical distribution is an NP-hard problem in the general case, meaning that it cannot be solved in polynomial time, given P $\neq$ NP. This popularity is explained by the fact that Kemeny's aggregation method satisfies numerous properties that are desirable for a consensus method, contrary to methods using other distances like Spearman's Footrule or Spearman's Rho. These desirable properties are too numerous to be exhaustively developed, but from a high-level perspective, they reveal the characteristics that a good consensus should exhibit. For example, the main one that is satisfied by Kemeny's aggregation is the following.

Property 2.2.2. Condorcet Criterion. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, and $\sigma^\star(P)$ be a consensus. Suppose that $\exists \, i_0 \in [n]$ such that $\forall i \in [n], P(\Sigma(i_0) < \Sigma(i)) \geq 1/2$, then $\sigma^\star(P)$ is said to satisfy Condorcet Criterion if $\sigma^\star(P)^{-1}(1) = i_0$.*

A consensus satisfying Condorcet Criterion thus ensures that an item being preferred over all other items in every head-to-head contest, meaning in pairwise comparison, must then be the preferred item. Kemeny's consensus satisfies this fundamental property, which is not the case for the same metric-based method when using either Spearman's footrule or Spearman's rho distances. Other properties satisfied by Kemeny's consensus are ranking consistency (which ensures that if the source dataset or distribution is divided into several parts and all parts exhibit the same Kemeny's consensus, then it must be Kemeny's consensus for the full problem), Pareto efficiency, and independence of Smith-dominated alternatives, see for example Dwork et al. (2001a).

For these reasons, even though different choices of distances can be considered in the metric-based approach defined by Equation (2.2.1), Kemeny's aggregation method remains the most popular and studied choice.

However, other approaches can also be considered to solve the Consensus Ranking problem. Such choices include the Borda Count method, the Copeland method, the Minimax Condorcet method, or even Markov Chains-based methods. Among them, the Borda Count remains a popular method thanks to its simplicity and its computational efficiency, even though it does not satisfy several desirable properties like Condorcet Criterion.

Definition 2.2.3. BORDA COUNT. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. The Borda count of an item $i \in [n]$ for distribution P is defined by:*

$$B_P(i) = \sum_{\sigma \in \mathfrak{S}_n} P(\sigma)\sigma(i) \tag{2.2.2}$$

Then, the Borda statistics is given by:

$$T_{Borda}(P) \in \text{argsort}(B_P), \tag{2.2.3}$$

where $\text{argsort}(s) = \{\sigma \in \mathfrak{S}_n, \forall r \in [n-1], s_{\sigma^{-1}(r)} \leq s_{\sigma^{-1}(r+1)}\}$

An interesting property of the Borda Count is that it corresponds to a consensus statistics when the metric l used is Spearman's Rho, as shown in Calauzènes et al. (2013).

2.2.2 Practical Approaches on Solving Kemeny's Consensus

As previsouly mentioned, Kemeny's Aggregation method is NP-hard in the general case, even for a small number of items such as $n = 4$, as proved by Dwork et al. (2001b). Fortunately, Kemeny's Consensus can be either approximated using Equation (2.2.1) with the Spearman's Footrule distance or local Kemenization as shown in Dwork et al. (2001b), or, alternatively, can be efficiently computed with additional hypothesis on the distribution under study.

This latter possibility has been investigated in Korba et al. (2017). They introduced the important notion of *stochastic transitivity*, which is recalled here.

Definition 2.2.4. PAIRWISE PROBABILITIES. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. Its corresponding pairwise probability matrix, denoted by $(p_{i,j})_{1 \leq i,j \leq n}$ is the matrix composed of the pairwise probabilities as defined by:*

$$\forall (i,j) \in [n]^2, \quad p_{i,j} = P(\Sigma(i) < \Sigma(j)). \tag{2.2.4}$$

As a quick remark, it obviously holds that $\forall (i,j) \in [n]^2,\ p_{j,i} = 1 - p_{i,j}$. As the Kendall Tau distance computes the number of pairwise disagreements between two rankings, it has a clear connection with the notion of pairwise probabilities. In fact, as stated in Korba et al. (2017), the following result can be derived:

Property 2.2.5. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, and $(p_{i,j})_{1 \leq i,j \leq n}$ its pairwise probability matrix. Then, $\forall \sigma \in \mathfrak{S}_n$:*

$$\mathbb{E}_{\Sigma \sim P}(d_\tau(\Sigma, \sigma)) = \sum_{i<j} p_{i,j}\mathbb{1}(\sigma(i) > \sigma(j)) + (1 - p_{i,j})\mathbb{1}(\sigma(i) < \sigma(j)) \tag{2.2.5}$$

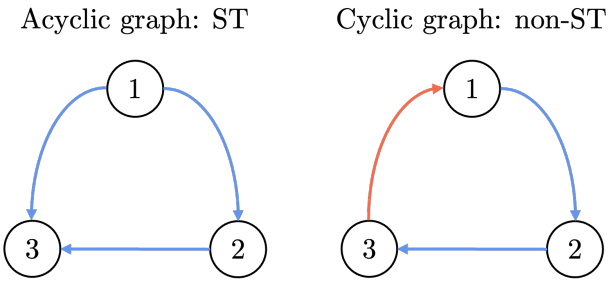


Figure 2.3: Illustration of stochastic transitivity. The two graphs represent the pairwise probabilities associated with two different distributions. An arrow from i to j indicates that $p_{i,j} > 1/2$. The leftmost distribution corresponds to a case where $p_{1,2} > 1/2, p_{2,3} > 1/2$ and $p_{1,3} > 1/2$; the corresponding graph is acyclic, and thus the distribution is ST. The rightmost distribution corresponds to a case where $p_{1,2} > 1/2, p_{2,3} > 1/2$ but $p_{1,3} < 1/2$; the corresponding graph is cyclic, and thus the distribution is non-ST.

Now, the aforementioned notion of *stochastic transitivity* is defined below, and illustrated in Figure 2.3 for more clarity.

Definition 2.2.6. STOCHASTIC TRANSITIVITY (ST). *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, and $(p_{i,j})_{1 \leq i,j \leq n}$ its pairwise probability matrix. P is said to be stochastically transitive (ST) if it satisfies:*

$$\forall (i,j,k) \in [n]^3, \quad p_{i,j} \geq 1/2 \text{ and } p_{j,k} \geq 1/2 \Rightarrow p_{i,k} \geq 1/2. \tag{2.2.6}$$

Furthermore, P is said to be strictly ST (SST) if all the comparisons in the previous Equation are strict.

The *stochastic transitivity* property, first explored in Davidson and Marschak (1959); Fishburn (1973), is fulfilled by some widely used ranking distributions, such as the Mallows distribution, and shown to facilitate various statistical tasks, see for example Shah et al. (2015); Shah and Wainwright (2018). In particular, Korba et al. (2017) demonstrated this important result:

Theorem 2.2.7. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a strictly stochastichally transitive (SST) distribution. Then,*

$$\sigma^\star = \text{argsort}(s), \quad \textit{with} \ \ \forall i \in [n], \ s(i) = 1 + \sum_{i \neq j} \mathbb{1}(p_{i,j} < 1/2) \tag{2.2.7}$$

is the unique Kemeny's consensus for distribution P.

Under the assumption that distribution P is SST, the computational cost of computing Kemeny's consensus is thus completely reduced. In an empirical case where a dataset S_N of size N is available, the computational cost of computing the pairwise probability matrix is $\mathcal{O}(n^2 N)$, and afterward the computation of Kemeny's consensus is $\mathcal{O}(nlog(n))$, which is computationally tractable.

In practice, this SST hypothesis is quite reasonable: not only most of the parametric distributions, such as the Mallows distribution or the Plackett-Luce distributions, are stochastically transitive, but 'real-world' datasets available, for example, in the *preflib*

library, at https://www.preflib.org/, are also stochastically transitive. Of course, non-stochastically transitive distributions or datasets can be constructed, for example using mixtures of different distributions, or by contaminating a dataset with additional adversarial inputs, as will be explored in Chapter 3.

2.2.3 Vulnerability of Consensus Median

Computing a consensus for any distribution P is thus not obvious. Not only several methods, as stated in Section 2.2.1, can be derived and lead to very different results (for example, Kemeny's consensus and Borda's consensus do not coincide in general, even when the distribution is SST), but the aforementioned methods are not always guaranteed to output unique results (Kemeny's consensus is not necessarily unique when the distribution is not SST). If, in addition, a probability or a dataset faces an adversarial attack, it can be sometimes very easy to modify the consensus.

Let's explore an example to better understand this limitation.

Suppose that $n = 3$ and distribution P_0 is defined as follows: $P_0(1 \succ 2 \succ 3) = P(3 \succ 2 \succ 1) = 1/2$. Then, Kemeny's consensus is the whole set $\mathfrak{S}_3$, so any ranking is a consensus. But now, if we have P_1 defined by $P_1(1 \succ 2 \succ 3) = 0.501$ and $P(3 \succ 2 \succ 1) = 0.499$; and equivalently we have distribution P_2 defined by $P_1(1 \succ 2 \succ 3) = 0.499$ and $P(3 \succ 2 \succ 1) = 0.501$, then Kemeny's consensus is unique in both cases and is $1 \succ 2 \succ 3$ for P_1 and $3 \succ 2 \succ 1$ for P_2, even though the differences in the three distributions P_0, P_1 and P_2 are very small.

This example is typically an illustration of the vulnerability of Kemeny's consensus to perturbations, and in particular adversarial ones. Prior to the research presented in this thesis, Agarwal et al. (2020) has identified and addressed the issue of robustness in a similar context. Their work focuses on a pairwise comparison-based setup, where full rankings are not available, only pairwise comparisons are. Notice that this can be equivalently mapped to a full ranking problem when studying Kemeny's consensus because, as previously shown in Property 2.2.5, Kendall Tau necessitates only pairwise probabilities to be computed. They study the problem of identifiability of the weights of a Bradley-Terry-Luce distribution P, which is a generalization of the Plackett-Luce model. By formulating the problem using graphs, they first derive a specific attack called Single Cut Corruption, which modifies some pairwise probabilities. Then, they provide conditions under which the true parameters can still be identified from a corrupted graph, and later they provide an efficient algorithm to achieve this identification.

Their work has paved the way for the study of robustness in rankings, and the work presented in Chapters 3 and 4 extends theirs by considering a non-parametric approach and non-constrained class of adversarial perturbations. To do so, two broad research directions can be identified. The first one is to derive new statistics or consensus methods, to reach a more accurate representation of the studied distribution. The work presented in Chapter 3 falls under this category by defining the concept of *Depth functions* for ranking distributions. The second direction aims at robustifying an existing consensus method like Kemeny's consensus, to account for potential adversarial perturbations. The work presented in Chapter 4 follows this idea by presenting a plugin that can be added to Kemeny's consensus to robustify it.

Summary of contributions on poisoning attacks

Chapter 3 is inspired by the following article: Morgane Goibert, Stéphan Clémençon, Ekhine Irurozki, Pavlo Mozharovskyi (2022). Statistical Depth Functions for Ranking Distributions: Definitions, Statistical Learning and Applications. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics (AISTATS 2022)* . See Goibert et al. (2022a)

It presents an adaptation of the concept of depth function for ranking data, which provides analogs of quantiles. It thus enables the computation of statistical procedures based on ranks, and specifically, it develops a trimming algorithm that aims at recovering a robust consensus for the consensus ranking task. This strategy is shown to be theoretically and experimentally effective.

Chapter 4 is inspired by the following article: Morgane Goibert, Clément Calauzènes, Ekhine Irurozki, Stéphan Clémençon (2023). Robust Consensus in Ranking Data Analysis: Definitions, Properties and Computational Issues. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*. See Goibert et al. (2023)

It presents a rigorous framework for computing and evaluating the breakdown function for any statistics devoted to the consensus ranking task. It also provides a robustification plugin that can be added to the output of any statistics, based on bucket rankings, which aims at incorporating undecidability for close items, which is shown to provide much more robustness and almost no precision loss compared to traditional statistics. These results are illustrated by experiments on synthetic and real data.

Chapter 3

Depth Functions for Ranking Distributions

Guess if you can, choose if you dare.

Pierre Corneille

Contents

3.1 High-level Overview

3.1.1 Outline of the Rationales of the Chapter

As explained at length in Sections 1.3 and 2.2, the question of finding a *consensus* ranking to solve the *location estimation* task, also called *consensus ranking* in the community, is at the core of the training of a machine learning model on ranking data. Inspired by Huber's robustification procedures explored in Section 1.3, a first approach to provide more robustness to the classical consensus statistics is to build statistics based on the *ranks* of a ranking random variable.

Indeed, rank-based statistics are very useful to define analogs of *quantiles*, which in turn can provide much more informative features about a studied distribution $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ than just the *median*, meaning the consensus. It is the purpose of this Chapter to define these analogs of quantiles, ranks, and the relevant statistical procedures based on such quantities for the analysis of ranking data by means of a metric-based notion of *depth function* on the symmetric group.

Overcoming the absence of vector space structure on $\mathfrak{S}_n$, the proposed depth function defines a center-outward ordering of the permutations in the support of P and extends the classic metric-based formulation of consensus ranking. The axiomatic properties that ranking depths functions should ideally possess will be listed, and computational and generalization issues are studied at length. Beyond the theoretical analysis carried out, the relevance of the novel concepts and methods are illustrated through the crafting of a *trimming trategy* to robustify the classical Kemeny's consensus, which is inspired by the typical trimmed mean or trimmed median statistics in the context of real-numbered data. This trimming strategy is shown to outperform Kemeny's consensus in terms of robustness both theoretically and empirically. Additionally, depth-based procedures are shown to be relevant for other classical statistical tasks, which showcase the usefulness and flexibility of this concept for ranking data.

3.1.2 Outline of the Main Contributions of the Chapter

This Chapter is devoted to defining quantities based on ranks for ranking data, as well as defining more robust statistics than classical ones such as Kemeny's statistics or Borda's statistics.

To do so, the concept of statistical depth function is first extended to the space of rankings. Some basics in statistical depth theory are briefly recalled in Section 3.2, while Section 3.3 introduces an extension of the notion of depth function tailored to ranking data. Desirable axioms for ranking depths are listed therein, and shown to hold under mild conditions, *e.g.* stochastic transitivity.

In Section 3.4, statistical guarantees are provided for the ranking depth and its by-products, in the form of non-asymptotic bounds for the deviations between the ranking depth function and its statistical counterpart in particular.

Then, in Section 3.4.2, the trimming algorithm, based on the ranking depth concept is proposed. One of its versions aims at recovering automatically a stochastically transitive version of the empirical ranking distribution so that computing Kemeny's consensus on

this trimmed dataset is ensured to produce a unique relevant solution. Other versions of the same algorithm, for example with a fixed proportion of the dataset to trim, are also explored.

Finally, beyond the theoretical and algorithmic concepts introduced previously and analyzed throughout the Chapter, the relevance of the notion of ranking depth for robustification purposes is explored experimentally in Section 3.5. Furthermore, the depth is also shown to be very interesting to solve a wide variety of statistical applications beyond robustness.

The main contributions are thus summarized below:

- Statistical depth and related axiomatic properties are extended to ranking data, in order to emulate quantiles/ranks for r.v.'s valued in $\mathfrak{S}_n$.
- A finite-sample analysis ensures the usability of the notion of ranking depth introduced.
- An algorithm of great simplicity that uses ranking depth to build stochastically transitive empirical ranking distributions (based on which, crucial statistical tasks such as consensus ranking are straightforward) is proposed.
- The ranking depth, and its related quantile regions in $\mathfrak{S}_n$, can be used for the statistical analysis of rankings: 1) fast and robust recovery of medians in consensus ranking, 2) informative graphical representations of ranking data, 3) anomaly/novelty detection, 4) homogeneity testing.

3.2 Background and Preliminaries

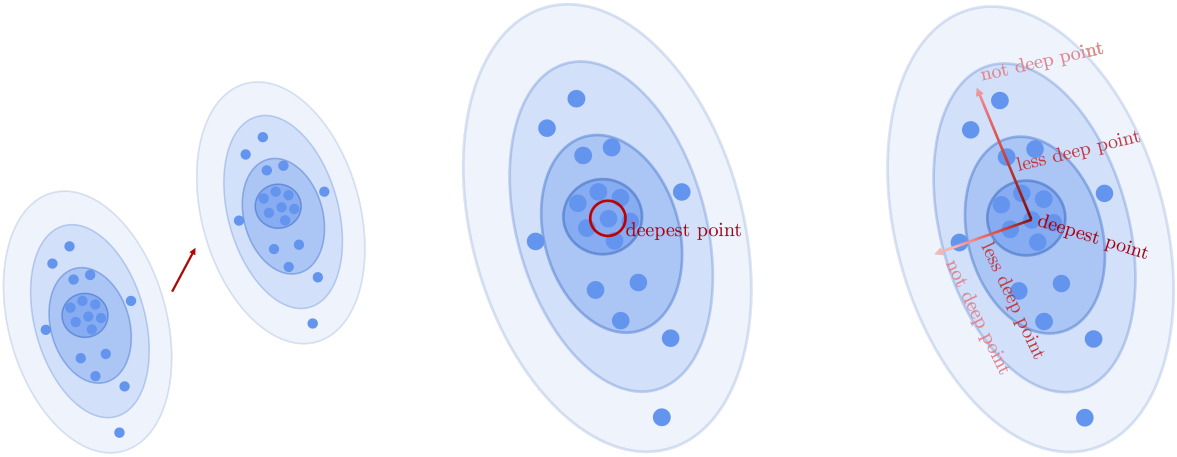
In this Section, the notion of *depth function* for multivariate data is thoroughly introduced and explained. For completeness, some results on the classical consensus aggregation techniques will also be recalled.

3.2.1 Depth Functions for Multivariate Data

In the absence of any 'natural order' on $\mathbb{R}^d$ with $d \geq 2$, the concept of *statistical depth* provides a mean to define a center-outward ordering of points in the support of a probability distribution P on $\mathbb{R}^d$, so as to extend the notions of order and (signed) rank statistics to multivariate data, as explored in *e.g.* Mosler (2013).

A depth function $D_P : \mathbb{R}^d \to \mathbb{R}_+$ relative to P should ideally assign the highest values $D_P(x)$ to points $x \in \mathbb{R}^d$ near the 'center' of the distribution, which is one of its main interest. Furthermore, the values $D_P(x)$ should ideally decrease as one moves away from the center. Since both characteristics are desirable, they are the core components of two out of a set of four axioms that defines depth functions. This axiomatic nomenclature has been introduced in Zuo and Serfling (2000a), listing the four axioms that statistical depths should ideally satisfy, even though different formulations of a statistically equivalent set of axioms are also explored in Dyckerhoff (2004); Mosler (2013). These axioms are illustrated in Figure 3.1 and defined as follows:

(i) (Affine invariance) Denoting by P_X the distribution of any r.v. X taking its values in $\mathbb{R}^d$, it holds: $D_{P_{AX+b}}(Ax+b) = D_P(x)$ for all $x \in \mathbb{R}^d$, any r.v. X valued in $\mathbb{R}^d$, any $d \times d$ nonsingular matrix A with real entries and any vector b in $\mathbb{R}^d$.



(a) Illustration of the affine invariance axiom. When translating the distribution on the upper-right corner, the depth function does not change.

(b) Illustration of the maximality at center axiom. The deepest point corresponds to the most central point for the distribution.

(c) Illustration of the monotonicity axiom. The depth decreases along rays (shown in red) when moving away from the deepest point.

Figure 3.1: Illustration of the 3 main axioms relative to depth functions.

(*ii*) (Maximality at center) For any probability distribution P on $\mathbb{R}^d$ that possesses a symmetry center x_P (for different notions of center), the depth function D_P takes its maximum value at it, *i.e.* $D_P(x_P) = \sup_{x \in \mathbb{R}^d} D_P(x)$.

(*iii*) (Monotonicity relative to deepest point) For any probability distribution P on $\mathbb{R}^d$ with deepest point x_P, the depth at any point x in $\mathbb{R}^d$ decreases as one moves away from x_P along any ray passing through it, *i.e.* $D_P(x) \leq D_P(x_P + \alpha(x - x_P))$ for any α in $[0, 1]$.

(*iv*) (Vanishing at infinity) For any probability distribution P on $\mathbb{R}^d$, the depth function D_P vanishes at infinity, *i.e.* $D_P(x) \to 0$ as $||x||$ tends to infinity.

A depth function is thus a class of functions that satisfy the aforementioned axioms. The first depth function, originally introduced in the seminal contribution Tukey (1975), is called the *half-space depth*. Specifically, for a point $x \in \mathbb{R}^d$ relative to a distribution $P \in \mathcal{M}_+^1(\mathbb{R}^d)$, it computes the minimum of the mass $P(H)$ taken over all closed half-spaces $H \subset \mathbb{R}^d$ such that $x \in H$. Many alternatives have been proposed since then, see *e.g.* Liu (1990); Liu and Singh (1993); Koshevoy and Mosler (1997); Chaudhuri (1996); Oja (1983); Vardi and Zhang (2000); Chernozhukov et al. (2017); Zuo and Serfling (2000a).

As the distribution P of interest is generally unknown in practice, its analysis relies on the observation of $N \geq 1$ independent realizations $X_1, \ldots, X_N$ of P. A statistical version of $D_P(x)$ can be built by replacing P with its empirical counterpart $\widehat{P}_N = (1/N) \sum_{i=1}^N \delta_{X_i}$, yielding the *empirical depth function* $D_{\widehat{P}_N}(x)$. Its consistency and asymptotic normality have been studied for various notions of depth, as explored in Donoho and Gasko (1992); Zuo and Serfling (2000b), and concentration results for empirical depth and contours have been recently proved in the half-space depth case in Burr and Fabrizio (2017); Brunel (2019).

3.2.2 Reminder on Consensus Ranking

The main approach to consensus ranking, introduced in Section 2.2.1, is recalled here:

Definition 2.2.1. CLASSICAL CONSENSUS STATISTICS. *Let $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a distance on rankings. A classical consensus statistics is a function $T_l : \mathcal{M}_+^1(\mathfrak{S}_n) \to \mathfrak{S}_n$ solving the following optimization problem: $\forall P \in \mathcal{M}_+^1(\mathfrak{S}_n)$,*

$$T_l(P) \in \underset{\sigma \in \mathfrak{S}_n}{\operatorname{argmin}} \, \mathbb{E}_{\Sigma \sim P}(l(\Sigma, \sigma)), \tag{2.2.1}$$

The output of statistics T_l is usually denoted by $\sigma_l^\star$ (where the dependence in P is dropped when the context is clear) and is simply called the consensus.

This definition presents the metric approach to solving the consensus ranking problem. Intuitively, such an optimization problem finds one or several rankings $\sigma \in \mathfrak{S}_n$ that have the smallest *ranking risk* with respect to the studied distribution $P \in \mathcal{M}_+^1 \mathfrak{S}_n$. The risk of a ranking $\sigma \in \mathfrak{S}_n$ is defined as follows:

$$L_P(\sigma) = \mathbb{E}_{\Sigma \sim P}(l(\Sigma, \sigma)) \tag{3.2.1}$$

An important remark can be made here: the ranking consensus $\sigma_l^\star$ is not necessarily unique, even though it is, in all cases, an informative summary of P, and $L_P(\sigma_l^\star)$ is an informative dispersion measure.

A second important remark is that the choice of the (pseudo) distance $l(.,\ .)$ is crucial, regarding the theoretical properties of the corresponding consensus and the computational feasibility. Various distances have been considered in the literature (see *e.g.* Deza and Huang (1998)): the most popular choices, introduced in Section 2.1.2, are the Kendall Tau distance, the Spearman's Footrule and Spearman's Rho distance, which can be completed with the Hamming distance for example, defined by $\forall\ (\sigma_1, \sigma_2) \in \mathfrak{S}_n^2 d_H(\sigma_1, \sigma_2) = \sum_{i=1}^n \mathbb{1}(\sigma_1(i) \neq \sigma_2(i))$

The literature has essentially focused on solving a statistical version of the minimization problem Equation (2.2.1) in Definition 2.2.1, as in Hudry (2008); Diaconis and Graham (1977); Bartholdi III et al. (1989). Assuming that $N \geq 1$ independent copies $\Sigma_1, \ldots, \Sigma_N$ of the generic random variables Σ are observed, a natural empirical estimate of $L_P(\sigma)$ is $\widehat{L}_N(\sigma) = (1/N) \sum_{s=1}^N d(\Sigma_s, \sigma) = L_{\widehat{P}_N}(\sigma)$, where $\widehat{P}_N = (1/N) \sum_{i=1}^N \delta_{\Sigma_i}$ is the empirical measure. The set $\mathfrak{S}_n$ being of finite cardinality, an empirical ranking risk minimizer always exists, just like a solution to Equation (2.2.1), not necessarily unique however. Generalization guarantees and fast rate conditions for empirical consensus ranking have been investigated in Korba et al. (2017).

3.3 Depth Functions for Ranking Data

In order to define relevant extensions of the concept of statistical depth to ranking data, we define axiomatic properties that candidate functions on $\mathfrak{S}_n$ should satisfy. We next show that the metric-based ranking depths we propose for ranking distributions analysis satisfy these axioms under mild conditions.

3.3.1 Ranking Depth: Axioms

Just like in the multivariate setup (see Section 3.2.1), a list of key axioms that the ranking depth function D_P should ideally satisfy can be made. These axioms are essential to emulate the information provided by quantiles (respectively quantile regions) of univariate distributions (respectively multivariate distributions) in a relevant manner. Let P be a ranking distribution, d a distance on $\mathfrak{S}_n$, the axioms desirable for any ranking depth $D_P : \mathfrak{S}_n \to \mathbb{R}_+$ are listed below.

Axiom 3.3.1. INVARIANCE. *For any $\nu \in \mathfrak{S}_n$, consider the ranking distribution νP defined by: $(\nu P)(\sigma) = P(\sigma \circ \nu^{-1})$ for all $\sigma \in \mathfrak{S}_n$. It holds that: $D_P(\sigma) = D_{\nu P}(\sigma \circ \nu)$ for all $(\sigma,\ \nu) \in \mathfrak{S}_n^2$.*

Axiom 3.3.2. MAXIMALITY AT CENTER. *For any probability distribution P on $\mathfrak{S}_n$ that possesses a symmetry center σ_P (in a certain sense, e.g. w.r.t. to a given metric d on $\mathfrak{S}_n$), the depth function D_P takes its maximum value at it, i.e. $D_P(\sigma_P) = \max_{\sigma \in \mathfrak{S}_n} D_P(\sigma)$.*

Axiom 3.3.3. LOCAL MONOTONICITY RELATIVE TO DEEPEST RANKING. *Assume that the deepest ranking $\sigma^\diamond = \operatorname{argmax}_{\sigma \in \mathfrak{S}_n} D_P(\sigma)$ is unique. The quantity $D_P(\sigma)$ decreases as $d(\sigma^\diamond, \sigma)$ locally increases, i.e. for any $\sigma \in \mathfrak{S}_n$ and $(i, j) \in [n]^2$ such that $\sigma(j) = \sigma(i) + 1$, if $d(\sigma^\diamond, \sigma \circ \tau_{i,j}) > d(\sigma^\diamond, \sigma)$, then we have $D_P(\sigma) > D_P(\sigma \circ \tau_{i,j})$.*

Note that, insofar as $\mathfrak{S}_n$ is of finite cardinality, there is no relevant analog of the 'vanishing at infinity' axiom for multivariate depth. The above three axioms thus completely characterize a ranking depth function. Among them, the *local monotonicity* axiom is perhaps the most important one, as it provides exactly the ordering information we are looking for.

3.3.2 Metric-based Ranking Depth Functions: Definition

Seeking to define a ranking depth that satisfies the axioms listed above and such that the consensus $\sigma_l^\star$ of P have maximal depth, the metric approach provides natural candidates, just like for consensus ranking.

Definition 3.3.4. METRIC-BASED RANKING DEPTH. Let l be a distance and $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ a distribution on rankings. The ranking depth based on l is defined as: $D_P^{(l)}$: $\forall\ \sigma \in \mathfrak{S}_n$,

$$D_P^{(l)}(\sigma) = \mathbb{E}_{\Sigma \sim P}[||l||_\infty - l(\sigma, \Sigma)] = \frac{||l||_\infty - L_P(\sigma)}{||l||_\infty}, \tag{3.3.1}$$

with $||l||_\infty = \max_{(\sigma,\nu) \in \mathfrak{S}_n^2} l(\sigma, \nu)$.

The shift induced by $||l||_\infty \geq L^\star = \max_{\sigma \in \mathfrak{S}_n} L_P(\sigma)$ simply guarantees non-negativity, in accordance with Definition 2.1 in Zuo and Serfling (2000a), while defining the same center-outward ordering of the permutations σ in $\mathfrak{S}_n$ as $-L_P$.

Notice that metric-based ranking depths can be viewed as extensions of multivariate depth functions of type A in the nomenclature proposed in Zuo and Serfling (2000a). For simplicity, we omit the superscript (l) and rather write D_P when no confusion is possible about the distance considered. Moreover, the distances in rankings, such as Kendall Tau and Spearman's Footrule and Rho, are upper-bounded: to ease the comparison, the depth will be normalized in the rest of the Chapter, meaning divided by $||l||_\infty$.

A ranking $\sigma \in \mathfrak{S}_n$ is said to be *deeper* than another one ν relative to the ranking distribution P if and only if $D_P(\nu) \leq D_P(\sigma)$ and we write $\nu \preceq_{D_P} \sigma$. The *ranking depth ordering* $\preceq_{D_P}$ is the preorder related to the depth function D_P.

Equiped with this notion of depth on $\mathfrak{S}_N$, a straightforward remark can be made.

Remark 3.3.5. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ a distance and $D_P^{(l)}$ the depth defined as in Definition 3.3.4. Let us write the consensus the usual way: $\sigma_l^\star := \text{argmin}_{\sigma \in \mathfrak{S}_n} \mathbb{E}_{\Sigma \sim P}[l(\Sigma, \sigma)]$ and the deepest ranking: $\sigma_l^\diamond := \text{argmax}\, D_P^{(l)}$*

$$\sigma_l^\star = \sigma_l^\diamond \tag{3.3.2}$$

This remark is a natural consequence of the definition of the depth $D_P^{(l)}$, and allows for a better appreciation of the importance of the choice of Definition 3.3.4 for the depth function.

Also, if P is a Dirac mass δ_{σ_0}, the ranking depth then simply boils down to the measure of closeness defined by the distance d chosen: $D_P(\sigma) = ||d||_\infty - d(\sigma_0, \sigma)$. In contrast, if P is the uniform distribution, the ranking depth relative to a classic distance on $\mathfrak{S}_n$ is constant over $\mathfrak{S}_n$. The depth function also allows to partition the space $\mathfrak{S}_n$ into subsets of rankings with equal depth.

Definition 3.3.6. DEPTH REGIONS/CONTOURS. For any $u \in \mathbb{R}$, the region of depth u is the superlevel set $\mathcal{R}_P(u) = \{\sigma \in \mathfrak{S}_n : \ D_P(\sigma) \geq u\}$ of D_P, while the ranking contour of depth u is the set $\partial\mathcal{R}_P(u) = \{\sigma \in \mathfrak{S}_n : \ D_P(\sigma) = u\}$.

Equipped with this notation, $\partial\mathcal{R}_P(-L_P(\sigma^\star))$ is the set of medians of P w.r.t. the metric l.

Definition 3.3.7. DEPTH SURVIVOR FUNCTION. The ranking depth survivor function is $s_P : u \in \mathbb{R} \mapsto s_P(u) = \mathbb{P}\{D_P(\Sigma) \geq u\}$.

Based on the metric-based ranking depth, the quantile regions are defined as follows.

Definition 3.3.8. QUANTILE REGIONS IN $\mathfrak{S}_n$. Let $\alpha \in (0, 1)$. The depth region with probability content α is the region of depth $s_P^{-1}(\alpha) = \inf\{u \in \mathbb{R} : \ s_P(u) \leq 1 - \alpha\}$: $R_P(\alpha) = \mathcal{R}_P(s_P^{-1}(\alpha))$. The mapping $\alpha \in (0, 1) \mapsto s_P^{-1}(\alpha)$ is called the ranking quantile function.

3.3.3 Metric-based Ranking Depth Functions: Main Axioms

As the object of depth functions, as well as by-products such as depth regions, contours, survivor function, and quantile regions have been defined, we are now going to explore related propositions that can be drawn from these. More precisely, we will provide conditions under which our candidate depth function satisfies all the axioms to be considered as such.

Invariance axiom.

We now state results showing that, under mild conditions and for popular choices of l, the metric-based ranking depth introduced in Definition 3.3.4 satisfies the key axioms listed in Section 3.3.1.

Proposition 3.3.9. About Invariance. *Suppose that l is right-invariant, i.e. $l(\nu \circ \pi, \sigma \circ \pi) = d(\nu, \sigma)$ for all $(\nu, \pi, \sigma) \in \mathfrak{S}_n^3$. Then, the ranking depth $D_P^{(l)}$ satisfies the invariance axiom Axiom 3.3.1.*

We point out that Kendall tau, Spearman's Footrule and Rho, Hamming, Ulam and Cayley distances are all right-invariant. Hence, the invariance axiom is satisfied for any ranking distribution in all situations involving a classical distance, which is always the case in practice.

Proof of Proposition 3.3.9 (proposition on invariance).

Let $\nu \in \mathfrak{S}_n$ and $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, such that νP is defined by $\forall\ \sigma \in \mathfrak{S}_n, (\nu P)(\sigma) = P(\sigma \circ \nu^{-1})$. Let $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a right-invariant distance. Then we have:

$$
\begin{aligned}
D_{\nu P}(\sigma \circ \nu) &= \mathbb{E}_{\Sigma \sim \nu P}[||l||_\infty - l(\sigma \circ \nu, \Sigma)] && (3.3.3)\\
&= ||l||_\infty - \sum_{\pi \in \mathfrak{S}_n} (\nu P)(\pi) l(\sigma \circ \nu, \pi) && (3.3.4)\\
&= ||l||_\infty - \sum_{\pi \in \mathfrak{S}_n} P(\pi \circ \nu^{-1}) l(\sigma \circ \nu, \pi) && (3.3.5)\\
&= ||l||_\infty - \sum_{\pi \in \mathfrak{S}_n} P(\pi \circ \nu \circ \nu^{-1}) l(\sigma \circ \nu, \pi \circ \nu) && (3.3.6)\\
&= ||l||_\infty - \sum_{\pi \in \mathfrak{S}_n} P(\pi) l(\sigma, \pi) && (3.3.7)\\
&= D_P(\sigma) && (3.3.8)
\end{aligned}
$$

□

The two remaining axioms require more care to be satisfied. The maximality axiom is mainly related to the notion of 'center', which is not a common object for ranking data, and thus must be correctly defined. The monotonicity axiom is more complex, but also at the core of the definition of a depth function.

Maximality axiom.

To better study both axioms, we need to recall the *stochastic transitivity* axiom that characterizes smooth distributions on rankings, already introduced in Definition 2.2.6 that is restated here.

Definition 2.2.6. Stochastic Transitivity (ST). *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, and $(p_{i,j})_{1 \leq i,j \leq n}$ its pairwise probability matrix. P is said to be stochastically transitive (ST) if it satisfies:*

$$
\forall (i,j,k) \in [n]^3, \quad p_{i,j} \geq 1/2 \text{ and } p_{j,k} \geq 1/2 \Rightarrow p_{i,k} \geq 1/2. \tag{2.2.6}
$$

Furthermore, P is said to be strictly ST (SST) if all the comparisons in the previous Equation are strict.

Then, the maximality axiom relies on the critical notion of 'center', which is not properly defined for ranking distributions. We propose two notions of center in the following paragraph, called the M-center and the H-center, which outline different properties of the studied distribution. The M-center notion is inspired by the metric approach that is common to the formulation of the consensus ranking task and our depth function, thus providing a notion of center in line with our approach. The H-center is inspired by the *half-space* symmetry, a classical notion from Tukey (1975); Zuo and Serfling (2000a) used in the classical definition of *half-space depth*, which provides a clear connection between our work and the seminal works on depth functions from the aforementioned contributions.

Definition 3.3.10. M-Center *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution and $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a metric. $\sigma_0 \in \mathfrak{S}_n$ is said to be a M-center if:*

$$\begin{aligned} &\forall\ (\sigma_1, \sigma_2, \sigma_3) \text{ such that } d(\sigma_0, \sigma_1) = d(\sigma_0, \sigma_2) < d(\sigma_0, \sigma_3), \text{ we have:} \\ &\mathbb{P}(\Sigma = \sigma_1) = \mathbb{P}(\Sigma = \sigma_2) \geq \mathbb{P}(\Sigma = \sigma_3) \end{aligned} \tag{3.3.9}$$

Definition 3.3.11. H-Center *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution and $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a metric. Let us call 'hyperplane' the sets $H_{i,j} = \{\sigma : \sigma(i) < \sigma(j)\}$. $\sigma_0 \in \mathfrak{S}_n$ is said to be a H-center if:*

$$\begin{aligned} &\forall\ (i,j) \in \{(i,j) \,|\, \sigma_0(i) < \sigma_0(j)\} \text{ we have:} \\ &P(\Sigma \in H_{i,j}) > P(\Sigma \in H_{j,i}) \end{aligned} \tag{3.3.10}$$

Proposition 3.3.12. Maximality at the center. *Let $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a distance and $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. Then we have the following results:*

1) If distribution P has a M-center and l is Kendall Tau, Spearman's Footrule or Spearman's Rho distance, then the maximality axiom Axiom 3.3.2 is satisfied.

2) If distribution P has a H-center and l is Kendall Tau distance then the maximality axiom Axiom 3.3.2 is satisfied.

Proof of Proposition 3.3.12 (maximality proposition for M-center)

Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. Let $\sigma_0 \in \mathfrak{S}_n$ be a M-center for P and $(i,j) \in [n]^2$ be two items such that $\sigma_0(i) < \sigma_0(j)$. Finally, let $\nu_1 \in \mathfrak{S}_n$ be a ranking such that $\nu_1(i) < \nu_1(j) = \nu_1(i) + 1$, $\tau_{i,j}$ be the transposition of i and j, and $\nu_2 = \nu_1 \circ \tau_{i,j}$. Thus, ν_1 and ν_2 are two neighboring rankings that differ only by swapping their adjacent items i and j.

Kendall Tau distance. Let $l = d_\tau$ be Kendall Tau distance. Proving Proposition 3.3.12 will follow by proving that $D_P^{(d_\tau)}(\nu_1) > D_P^{(d_\tau)}(\nu_2) \Leftrightarrow \mathbb{E}_{\Sigma\sim P}[d_\tau(\Sigma, \nu_1)] < \mathbb{E}_{\Sigma\sim P}[d_\tau(\Sigma, \nu_2)]$.

To do so, let us make two remarks.

First, for any $\sigma \in \mathfrak{S}_n$, $d_\tau(\nu_2, \sigma) - d_\tau(\nu_1, \sigma) = 1$ if $\sigma(i) < \sigma(j)$ and $= -1$ if $\sigma(i) > \sigma(j)$.

Second, let us write $S_{\sigma_0}(d) = \{\sigma \in \mathfrak{S}_n \mid d_\tau(\sigma_0, \sigma) = d\}$ the sphere centered in σ_0 and of radius d, and $\#S_{\sigma_0}(d)$ its cardinality. As σ_0 is a M-center, we have that $\forall\ \sigma \in$

$S_{\sigma_0}(d), P(\Sigma = \sigma) := P_d$ is constant. Moreover, the following remarks hold: if $d \leq \lfloor ||d_\tau||_\infty/2 \rfloor$, then $\#S_{\sigma_0}(d) \cap \{\sigma | \sigma(i) < \sigma(j)\} > \#S_{\sigma_0}(d) \cap \{\sigma | \sigma(i) > \sigma(j)\}$. Conversely, $d \leq \lceil ||d_\tau||_\infty/2 \rceil$, then $\#S_{\sigma_0}(d) \cap \{\sigma | \sigma(i) < \sigma(j)\} < \#S_{\sigma_0}(d) \cap \{\sigma | \sigma(i) > \sigma(j)\}$. Moreover, if $||d_\tau||_\infty$ is even, then $\#S_{\sigma_0}(||d_\tau||_\infty/2) \cap \{\sigma | \sigma(i) < \sigma(j)\} > \#S_{\sigma_0}(||d_\tau||_\infty/2) \cap \{\sigma | \sigma(i) > \sigma(j)\}$.

For easiness of read, let's write $\#S_{\sigma_0}(d) \cap \{\sigma | \sigma(i) < \sigma(j)\} = \#S_{\sigma_0}(d, +)$ and $\#S_{\sigma_0}(d) \cap \{\sigma | \sigma(i) < \sigma(j)\} = \#S_{\sigma_0}(d, -)$

Then:

$$\sum_\sigma \mathbb{P}(\Sigma = \sigma) \left[d_\tau(\nu_2, \sigma) - d_\tau(\nu_1, \sigma) \right] \tag{3.3.11}$$

$$= \sum_{d=0}^{||d_\tau||_\infty} P_d \times (\#S_{\sigma_0}(d, +) - \#S_{\sigma_0}(d, -)) \tag{3.3.12}$$

$$\begin{aligned} = & \sum_{d=0}^{\lfloor ||d_\tau||_\infty/2 \rfloor} P_d \times (\#S_{\sigma_0}(d, +) - \#S_{\sigma_0}(d, -)) \\ & + \sum_{d'=\lceil ||d_\tau||_\infty/2 \rceil}^{||d_\tau||_\infty} \underbrace{P_{d'}}_{< P_{||d_\tau||_\infty - d'}} \times \underbrace{(\#S_{\sigma_0}(d, +) - \#S_{\sigma_0}(d, -))}_{<0} \end{aligned} \tag{3.3.13}$$

$$\begin{aligned} > & \sum_{d=0}^{\lfloor ||d_\tau||_\infty/2 \rfloor} P_d \times (\#S_{\sigma_0}(d, +) - \#S_{\sigma_0}(d, -)) \\ & + \sum_{d=0}^{\lfloor ||d_\tau||_\infty/2 \rfloor} P_d \times (\#S_{\sigma_0}(||d_\tau||_\infty - d, +) - \#S_{\sigma_0}(||d_\tau||_\infty - d, -)) \text{ by a change} \\ & \text{of variable } d \leftarrow ||d_\tau||_\infty - d \end{aligned} \tag{3.3.14}$$

$$\begin{aligned} > & \sum_{d=0}^{\lfloor ||d_\tau||_\infty/2 \rfloor} P_d \times \\ & \underbrace{[(\#S_{\sigma_0}(d, +) - \#S_{\sigma_0}(d, -)) + (\#S_{\sigma_0}(|||l||_\infty - d, +) - \#S_{\sigma_0}(||d_\tau||_\infty - d, -))]}_{=0} \end{aligned} \tag{3.3.15}$$

$$> 0, \text{ which ends the proof for Kendall Tau.} \tag{3.3.16}$$

Spearman's Footrule. Let $l = d_1$ be Spearman's Footrule distance. Similarly to Kendall Tau, notice the following: $\forall \sigma \in \mathfrak{S}_n$

$$d_1(\sigma, \nu_2) = \sum_{k=1}^{N} |\sigma(k) - \nu_2(k)| \tag{3.3.17}$$

$$= \sum_{k \neq i,j} |\sigma(k) - \nu_1(k)| + |\sigma(i) - \nu_1(i) - 1| + |\sigma(j) - \nu_1(j) + 1| \tag{3.3.18}$$

$$= \begin{cases} d_1(\sigma, \nu_1) \text{ if } \sigma(i) < \sigma(j) \leq \nu_1(i) < \nu_1(j) \text{ or } \nu_1(i) < \nu_1(j) \leq \sigma(i) < \sigma(j) \\ \qquad \text{or } \sigma(j) < \sigma(i) \leq \nu_1(i) < \nu_1(j) \text{ or } \nu_1(i) < \nu_1(j) \leq \sigma(j) < \sigma(i) \\ d(\sigma, \sigma_0) + 2 \text{ if } \sigma(i) \leq \nu_1(i) < \nu_1(j) \leq \sigma(j) \quad \text{(A)} \\ d(\sigma, \sigma_0) - 2 \text{ if } \sigma(j) \leq \nu_1(i) < \nu_1(j) \leq \sigma(i) \quad \text{(B)} \end{cases} \tag{3.3.19}$$

Then, we aim to compute:

$$\begin{aligned}&\sum_{\sigma} \mathbb{P}(\Sigma = \sigma)\left[d_1(\nu_2, \sigma) - d_1(\nu_1, \sigma)\right] \\ &= 2\left[\sum_{\sigma|\sigma\in(A)} \mathbb{P}(\Sigma = \sigma) - \sum_{\sigma|\sigma\in(B)} \mathbb{P}(\Sigma = \sigma)\right]\end{aligned} \tag{3.3.20}$$

Since the sets (A)=$\{\sigma|\sigma(i) \leq \nu_1(i) < \nu_1(j) \leq \sigma(j)\}$ and (B) = $\{\sigma|\sigma(j) \leq \nu_1(i) < \nu_1(j) \leq \sigma(i)\}$ are symmetric, we can pair each element of (A) with an element of (B) the following way: let $\sigma \in$ (A), then $\nu = \sigma \circ \tau_{i,j} \in$ (B). Thus, we have more broadly that (A) = (B) $\circ\, \tau_{i,j} = \{\sigma \circ \tau_{i,j} | \sigma \in (A)\}$.

Futhermore, we have for any $\sigma \in$ (A), and thus $\nu \in$ (B), that $d_1(\sigma_0, \sigma) < d_1(\sigma_0, \nu)$, which implies, because σ_0 is a M-center, that $P(\Sigma = \sigma) \geq P(\Sigma = \nu)$. Thus,

$$\sum_{\sigma} \mathbb{P}(\Sigma = \sigma)\left[d_1(\nu_2, \sigma) - d_1(\nu_1, \sigma)\right] \tag{3.3.21}$$

$$= 2\left[\sum_{\sigma|\sigma\in(A)} \mathbb{P}(\Sigma = \sigma) - \sum_{\sigma|\sigma\in(B)} \mathbb{P}(\Sigma = \sigma)\right] \tag{3.3.22}$$

$$= 2\left[\sum_{\sigma|\sigma\in(A)} \mathbb{P}(\Sigma = \sigma) - \sum_{\sigma|\sigma\in(A)} \mathbb{P}(\Sigma = \sigma \circ \tau_{i,j})\right] \tag{3.3.23}$$

$$= 2 \sum_{\sigma|\sigma\in(A)} \left[\mathbb{P}(\Sigma = \sigma) - \mathbb{P}(\Sigma = \sigma \circ \tau_{i,j})\right] \tag{3.3.24}$$

$$\geq 0, \text{ which ends the proof for Spearman's Footrule} \tag{3.3.25}$$

Spearman's Rho. Let $l = d_2$ be Spearman's Rho distance. Similarly to previous observations, since σ_0 is a M-center we have that $\forall \sigma \in \mathfrak{S}_n$ such that $\sigma(i) < \sigma(j), P(\Sigma = \sigma) \geq P(\Sigma = \sigma \circ \tau_{i,j})$. Thus,

$$\sum_{\sigma} \mathbb{P}(\Sigma = \sigma)\left[d_2(\nu_2, \sigma) - d_2(\nu_1, \sigma)\right] \tag{3.3.26}$$

$$= \sum_{\sigma|\sigma(i)<\sigma(j)} \mathbb{P}(\Sigma = \sigma)\left[d_2(\nu_2, \sigma) - d_2(\nu_1, \sigma)\right] + \sum_{\sigma|\sigma(i)>\sigma(j)} \mathbb{P}(\Sigma = \sigma)\left[d_2(\nu_2, \sigma) - d_2(\nu_1, \sigma)\right] \tag{3.3.27}$$

$$= \sum_{\sigma|\sigma(i)<\sigma(j)} \mathbb{P}(\Sigma = \sigma)\left[d_2(\nu_2, \sigma) - d_2(\nu_1, \sigma)\right] + \sum_{\sigma|\sigma(i)<\sigma(j)} \mathbb{P}(\Sigma = \sigma \circ \tau_{i,j})\left[\underbrace{d_2(\nu_2, \sigma \circ \tau_{i,j})}_{=d_2(\nu_1,\sigma) \text{ since } \nu_2=\nu_1\circ\tau_{i,j}} - \underbrace{d_2(\nu_1, \sigma \circ \tau_{i,j})}_{=d_2(\nu_2,\sigma)}\right] \tag{3.3.28}$$

$$= \sum_{\sigma|\sigma(i)<\sigma(j)} \left[P(\Sigma = \sigma) - P(\Sigma = \sigma \circ \tau_{i,j})\right]\left[d_2(\nu_2, \sigma) - d_2(\nu_1, \sigma)\right] \tag{3.3.29}$$

$$\geq 0, \text{ which ends the proof for Spearman's Rho} \tag{3.3.30}$$

□

Now, let's delve into the maximality axiom related to H-center. First, we will relate the notion of H-center to that of stochastical transitivity through the following proposition.

Proposition 3.3.13. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. We have the following equivalence:*

$$P \text{ possesses a } H\text{-center in } \sigma_0 \Leftrightarrow P \text{ is strictly stochastically transitive} \tag{3.3.31}$$

Proof First, let us suppose that P is SST. Thus, as shown in Korba et al. (2017), Kemeny's consensus can be defined by $\sigma_P^\star = \text{argsort}(s)$, where $\forall i \in [n], s(i) = 1 + \sum_{j \neq i} \mathbb{1}(p_{i,j} < 1/2)$, and is unique. Thus, let (i, j) be two items such that $\sigma_P^\star(i) < \sigma_P^\star(j)$. Let's show that $P(\Sigma \in H_{i,j}) > 1/2 \Leftrightarrow p_{i,j} > 1/2$. We have: $\sum_{k \neq j} \mathbb{1}(p_{j,k} < 1/2) - \sum_{k \neq i} \mathbb{1}(p_{i,k} < 1/2) = \sum_{k \neq i,j} \mathbb{1}(p_{k,j} > 1/2) - \mathbb{1}(p_{i,k} < 1/2) + 1 - 2\mathbb{1}(p_{i,j} < 1/2)$. As P is SST, this difference is equal to either of two solutions: A) $1 \times \#\{k | i \succ k \succ j\} + 1$, or B) $-1 \times \#\{k | j \succ k \succ i\} - 1$. Solution A) is positive, and solution B) is negative, thus only solution A) is possible since $s(i) < s(j)$, which implies that $p_{i,j} > 1/2$. Then, we indeed have that $\sigma_P^\star$ is a H-center for P.

Second, let us suppose that P possesses a H-center in σ_0. Let $(i, j, k) \in [n]^3$, $p_{i,k} > 1/2$ and $p_{k,j} > 1/2$. Let us show that $p_{i,j} > 1/2$. By definition of the H-center, $\sigma_0(i) < \sigma_0(k)$ and $\sigma_0(k) < \sigma_0(j)$ so $\sigma_0(i) < \sigma_0(j)$. This implies that $P(\Sigma \in H_{i,j}) > 1/2 \Leftrightarrow p_{i,j} > 1/2$. □

With this intermediary result, the maximality proposition relative to H-center and Kendall Tau distance follows immediately from the previous result and Korba et al. (2017).

Proof of Proposition 3.3.12 (maximality proposition for H-center and Kendall Tau)

Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution with a H-center $\sigma_0 \in \mathfrak{S}_n$. From Proposition 3.3.13, P is thus SST, and from Korba et al. (2017), σ_0 is it's unique Kemeny's consensus. Thus, $\sigma_0 = \text{argmin}_{\sigma \in \mathfrak{S}_n} \mathbb{E}_{\Sigma \sim P}[d_\tau(\Sigma, \sigma)] = \text{argmax } D_P^{(d_\tau)}$. □

Monotonicity axiom.

Finally, the monotonicity axiom is the most important axiom for depth functions and also the most restrictive. We provide here the conditions under which this axiom holds.

Proposition 3.3.14. Local Monotonicity.

Let $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a metric and $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. Then we have the following results:

1) If distribution P is SST and l is Kendall Tau distance, then Axiom 3.3.3 is satisfied.

2) If distribution P has a M-center and l is Kendall Tau, Spearman's Footrule of Spearman's Rho distance, then Axiom 3.3.3 is satisfied.

Proof of Proposition 3.3.14 (proposition on local monotonicity).

Let $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a metric and $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution

M-center version. The proof is the same as the one provided for the maximality axiom. Indeed, when l is Kendall Tau, Spearman's Footrule or Spearman's Rho distance and σ_0 is the M-center, we have shown that $D_P^{(l)}(\nu) > D_P^{(l)}(\nu \circ \tau_{i,j})$ for any ν ordering i and j the same way as σ_0: this is exactly the characterization provided by Proposition 3.3.14.

Kendall Tau/SST version. Suppose that P be is SST and let l be Kendall Tau distance. Following results from Korba et al. (2017), let's denote $\sigma^\star$ its unique Kemeny's consensus, which is also its deepest ranking using Proposition 3.3.12. Suppose, without loss of generality, that $(i, j) \in [n]^2$ are two items such that $\sigma^\star(i) < \sigma^\star(j)$. Finally, let $\nu_1 \in \mathfrak{S}_n$ be a ranking such that $\nu_1(i) < \nu_1(j) = \nu_1(i) + 1$, $\tau_{i,j}$ be the transposition of i and j, and $\nu_2 = \nu_1 \circ \tau_{i,j}$. Thus, ν_1 and ν_2 are two neighboring rankings that differ only by swapping their adjacent items i and j, meaning that $\forall k \neq i, j, \nu_2(k) = \nu_1(k)$. We have:

$$D_P^{(d_\tau}(\nu_1) \geq D_P^{(d_\tau}(\nu_2) \tag{3.3.32}$$

$$\Leftrightarrow \mathbb{E}_{\Sigma\sim P}[d_\tau(\Sigma, \nu_1)] \leq \mathbb{E}_{\Sigma\sim P}[d_\tau(\Sigma, \nu_2)] \tag{3.3.33}$$

$$\Leftrightarrow \sum_{k<k'} p_{k,k'}\mathbb{1}[\nu_2(i) > \nu_2(j)] + \sum_{k<k'} p_{k',k}\mathbb{1}[\nu_2(i) < \nu_2(j)] - \sum_{k<k'} p_{k,k'}\mathbb{1}[\nu_1(i) > \nu_1(j)] - \sum_{k<k'} p_{k,k'}\mathbb{1}[\nu_1(i) > \nu_1(j)] \geq 0 \tag{3.3.34}$$

$$\Leftrightarrow \sum_{k<k' \wedge k,k' \neq i,j} p_{k,k'}\mathbb{1}[\nu_1(i) > \nu_1(j)] + \sum_{k<k' \wedge k,k' \neq i,j} p_{k',k}\mathbb{1}[\nu_1(i) < \nu_1(j)] + p_{i,j} - \sum_{k<k' \wedge k,k' \neq i,j} p_{k,k'}\mathbb{1}[\nu_1(i) > \nu_1(j)] - \sum_{k<k' \wedge k,k' \neq i,j} p_{k,k'}\mathbb{1}[\nu_1(i) > \nu_1(j)] - p_{j,i} \geq 0 \tag{3.3.35}$$

$$\Leftrightarrow p_{i,j} > 1/2 \tag{3.3.36}$$

As P is SST and $\sigma^\star(i) < \sigma^\star(j)$, it thus holds that $p_{i,j} > 1/2$, which ends the proof. □

Notice that the M-center condition is restrictive (though satisfied by Mallows distributions, as defined in Definition 2.1.6), and in addition, a distribution having a M-center is ST. The SST condition, on the other hand, arises naturally in distributions computed on real datasets. This explains why we focus on computing the depth for the more general class of distributions being SST, rather than on those having a M-center.

3.3.4 Additional Results for Kendall's Tau Distance

In the Kendall Tau case, additional useful results can be stated. In particular, the ranking depth is then entirely determined by the *pairwise probabilities* $p_{i,j} = \mathbb{P}\{\Sigma(i) < \Sigma(j)\}$,

$1 \leq i \neq j \leq n$.

Proposition 3.3.15. *We have:* $\forall\ \sigma \in \mathfrak{S}_n$, $D_P^{(d_\tau)}(\sigma) = \binom{n}{2} - \sum_{i<j} p_{i,j}\mathbb{1}(\sigma(i) > \sigma(j)) - \sum_{i<j}(1-p_{i,j})\mathbb{1}(\sigma(i) < \sigma(j))$.

Proof The proof of Proposition 3.3.15 is a simple computation, remembering that $\forall i \neq j, p_{i,j} = \mathbb{P}(\Sigma(i) < \Sigma(j))$.

$$D_P^{(d_\tau)}(\sigma) = ||d_\tau|| - \mathbb{E}_{\Sigma\sim P}(d_\tau(\Sigma,\sigma)) \tag{3.3.37}$$

$$= \binom{n}{2} - \sum_{\nu\in\mathfrak{S}_n} P(\nu) \sum_{i<j} \mathbb{1}\left((\sigma(i)-\sigma(j))(\nu(i)-\nu(j)) < 0\right) \tag{3.3.38}$$

$$= \binom{n}{2} - \sum_{i<j} \sum_{\nu\in\mathfrak{S}_n, \nu(i)<\nu(j)} P(\nu)\mathbb{1}(\sigma(i) > \sigma(j)) - \sum_{i<j} \sum_{\nu\in\mathfrak{S}_n, \nu(i)>\nu(j)} P(\nu)\mathbb{1}(\sigma(i) < \sigma(j)) \tag{3.3.39}$$

$$= \binom{n}{2} - \sum_{i<j} p_{i,j}\mathbb{1}(\sigma(i) > \sigma(j)) - \sum_{i<j}(1-p_{i,j})\mathbb{1}(\sigma(i) < \sigma(j)) \tag{3.3.40}$$

□

This case is computationally attractive, the complexity being of order $O(n^2)$. In addition, note that the computation of D_P involves pairwise comparisons solely, which means an alternative statistical framework can be considered, where observations take the form of binary variables $\{\Sigma(\mathbf{i}) < \Sigma(\mathbf{j})\}$, $(\mathbf{i},\ \mathbf{j})$ being a random pair in $\{(i,j) :\ 1 \leq i < j \leq n\}$, independent from Σ.

Proposition 3.3.16. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a ST distribution.The following assertions hold true.*

(i) The largest ranking depth value is $D_P^\star = \sum_{i<j}\left\{\frac{1}{2} + \left|p_{i,j} - \frac{1}{2}\right|\right\}$. The deepest rankings relative to P and d_τ are the permutations $\sigma \in \mathfrak{S}_n$ such that: $\forall i < j$ s.t. $p_{i,j} \neq 1/2$, $(\sigma(j) - \sigma(i)) \cdot (p_{i,j} - 1/2) > 0$.

(ii) The smallest ranking depth value is $\underline{D}_P = \sum_{i<j}\left\{\frac{1}{2} - \left|p_{i,j} - \frac{1}{2}\right|\right\}$. The least deep rankings relative to P and d_τ are the permutations $\sigma \in \mathfrak{S}_n$ such that: $\forall i < j$ s.t. $p_{i,j} \neq 1/2$, $(\sigma(j) - \sigma(i)) \cdot (p_{i,j} - 1/2) < 0$.

(iii) If, in addition, P is SST, then we have $\partial\mathcal{R}_P(D_P^\star) = \{\sigma^\star\}$ and $\partial\mathcal{R}_P(\underline{D}_P) = \{\underline{\sigma}\}$, where $\sigma^\star(i) = 1 + \sum_{j\neq i}\mathbb{I}\{p_{i,j} < 1/2\} = n - \underline{\sigma}(i)$ for $i \in \{1,\ \ldots,\ n\}$. We also have $D_P^\star - D_P(\sigma) = 2\sum_{i<j}|p_{i,j} - 1/2| + D_P(\sigma) - \underline{D}_P = 2\sum_{i<j}|p_{i,j} - 1/2| \cdot \mathbb{I}\{(\sigma(j) - \sigma(i))(p_{i,j} - 1/2) < 0\}$.

These three results can be obtained in a straightforward manner.

Proof Observing that $n(n-1)/2 = \mathbb{E}_{\Sigma\sim P}[d_\tau(\Sigma,\sigma)] + \mathbb{E}_{\Sigma\sim P}[d_\tau(\Sigma,\sigma^R)]$ for all $\sigma \in \mathfrak{S}_n$, where σ^R is the reverse of σ, the result is essentially a reformulation of Theorem 5 in Korba et al. (2017) in terms of ranking depth, insofar as $D_P(\sigma) = n(n-1)/2 - \mathbb{E}_{\Sigma\sim P}[d_\tau(\Sigma,\sigma)]$. □

Let us recall some classical results about the Mallows distribution. Taking $l = d_\tau$, the Mallows model introduced in Mallows (1957b) and defined in Definition 2.1.6, is recalled here:

Definition 2.1.6. MALLOWS MODEL. *Let $\sigma_0 \in \mathfrak{S}_n$ be a central ranking, $\theta > 0$ a dispersion parameter and d_τ Kendall Tau distance. The probability distribution $P \sim M(\sigma_0,\theta)$ defined by:*

$$\forall \sigma \in \mathfrak{S}_n,\ P(\sigma) = \frac{1}{\Psi(\theta, d_\tau)} e^{-\theta d_\tau(\sigma_0,\sigma)} \tag{2.1.4}$$

is referred to as the Mallows model (or distribution) of center σ_0 and dispersion parameter θ, where $\Psi(\theta, d_\tau)$ is the normalization constant.

One may easily show that the normalization constant $\Psi(\theta, d_\tau)$ is independent from σ_0 and that $Z_0 = \prod_{i=1}^{n-1}\sum_{j=0}^{i} e^{-j\theta}$. When $\theta > 0$, the permutation σ_0 of reference is the mode of distribution $P \sim M(\sigma_0,\theta)$, as well as its unique Kemeny's consensus. Observe in addition that the highest the parameter θ, the spikiest the distribution P. In contrast, P is the uniform distribution on $\mathfrak{S}_n$ when $\theta = 0$.

A closed-from expression of the pairwise probabilities $p_{i,j}$ is available (see *e.g.* Theorem 2 in Busa-Fekete et al. (2014)). Setting $h(k,\theta) = k/(1-e^{-k\theta})$ for $k \geq 1$, one can then show the following:

Proposition 3.3.17. *Let $\sigma_0 \in \mathfrak{S}_n$ and $\theta \geq 0$. Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution such that $P \sim M(\sigma_0,\theta)$ Then: $\forall\ \sigma \in \mathfrak{S}_n$, $D_P^{(d_\tau)}(\sigma) = \binom{n}{2} - \sum_{\sigma(i)>\sigma(j)} H(\sigma_0(j) - \sigma_0(i),\ \theta)$, where $H(k,\theta) = h(k+1,\theta) - h(k,\theta)$ and $H(-k,\theta) = 1 - H(k,\theta)$ for $k \geq 1$.*

Proof Theorem 2 in Busa-Fekete et al. (2014) states that for the Mallows model and using our notations, $\forall i \neq j, p_{i,j} = H(\sigma_0(j) - \sigma_0(i),\theta)$. The results follow from direct application of Proposition 3.3.15 □

3.4 Statistical Issues

The ranking depth D_P is generally unknown, just like the ranking distribution P, and must be replaced by an empirical estimate based on supposedly available ranking data in practice. Here we establish nonasymptotic statistical guarantees for the empirical counterpart of the ranking depth and other related quantities. We also propose an algorithm,

based on the ranking depth, that permits to build, from any ranking dataset, an empirical ranking distribution fulfilling the crucial (strict) stochastic transitivity property, see Section 3.3.3.

3.4.1 Generalization: Learning Rates Bounds

Based on the observation of an i.i.d. sample $\Sigma_1, \ldots, \Sigma_N$ drawn from P with $N \geq 1$, statistical versions of the quantities introduced in Section 3.3.2 can be built by replacing P with the empirical distribution $\widehat{P}_N$. The empirical ranking depth is thus given by: $\forall \sigma \in \mathfrak{S}_n$, $\widehat{D}_N(\sigma) = D_{\widehat{P}_N}(\sigma) = ||d||_\infty - \mathbb{E}_{\Sigma \sim \widehat{P}_N}[l(\Sigma, \sigma)]$.

Similarly, the empirical ranking depth regions are $\widehat{\mathcal{R}}_N(u) = \{\sigma \in \mathfrak{S}_n : \widehat{D}_N(\sigma) \geq u\}$ for $u \geq 0$. In order to build an estimator of the ranking depth survivor function $S_P(u)$ with a tractable dependence structure, a *2-split* trick can be used, yielding the statistic

$$\widehat{S}_N(u) = \frac{1}{N - \lfloor N/2 \rfloor} \sum_{i=1+\lfloor N/2 \rfloor}^{N} \mathbb{I}\{\widehat{D}_{\lfloor N/2 \rfloor}(\Sigma_i) \geq u\}. \tag{3.4.1}$$

As the random variable $D_P(\Sigma)$ is discrete, the use of smoothing/interpolation procedures is required to ensure good statistical properties for the survivor function estimator and for the empirical quantiles it defines, see Sheather and Marron (1990); Ma et al. (2011). For instance, a kernel smoothed version of S_P can be computed by means of a non-negative differentiable Parzen-Rosenblatt kernel $K : \mathbb{R} \to \mathbb{R}_+$ s.t. $||K'||_\infty = \sup_{u \in \mathbb{R}} |K'(u)| < \infty$ and $\int_{\mathbb{R}} K(u)du = +1$ and a smoothing bandwidth $h > 0$, namely: $\widetilde{S}_P(u) = K_h * S_P$, which can be estimated by $\widetilde{S}_N(u) = K_h * \widehat{S}_N$, where $K_h(u) = K(u/h)/h$ for $u \in \mathbb{R}$. One may then define a smooth estimate of the ranking depth region with probability content $\alpha \in [0, 1]$ as well: $\widehat{R}_N(\alpha) = \widehat{\mathcal{R}}_N(\widetilde{S}_N^{-1}(\alpha))$. The result below provides bounds of order $O_{\mathbb{P}}(1/\sqrt{N})$ for the maximal deviations between D_P (resp. $\widetilde{S}_P$) and its empirical version.

Proposition 3.4.1. Statistical Bounds on Depth and Survivor Function. *The following assertions hold true.*

1) For any $\delta \in (0, 1)$, we have with probability at least $1 - \delta$: $\forall N \geq 1$,

$$\sup_{\sigma \in \mathfrak{S}_n} |\widehat{D}_N(\sigma) - D_P(\sigma)| \leq ||d||_\infty \sqrt{\frac{\log(2\, n!/\delta)}{2N}}. \tag{3.4.2}$$

2) For any $\delta \in (0, 1)$ and $h > 0$, we have with probability at least $1 - \delta$: $\forall N \geq 1$,

$$\sup_{u \geq 0} |\widetilde{S}_N(u) - \widetilde{S}_P(u)| \leq \sqrt{\frac{\log(4/\delta)}{2N}} + ||d||_\infty \sqrt{\frac{\log(4n!/\delta)}{2N}}. \tag{3.4.3}$$

Proof Hoeffding inequality combined with the union bound yields: $\forall t > 0$,

$$\mathbb{P}\left\{\sup_{\sigma \in \mathfrak{S}_n} \left|\widehat{D}_N(\sigma) - D_P(\sigma)\right| > t\right\} \leq \sum_{\sigma \in \mathfrak{S}_n} \mathbb{P}\left\{\frac{1}{N}\left|\sum_{i=1}^{N} \{l(\Sigma_i, \sigma) - \mathbb{E}_P[l(\Sigma, \sigma)]\}\right| > t\right\}$$

$$\leq 2n! \exp\left(-\frac{N2t^2}{||l||_\infty^2}\right), \quad (3.4.4)$$

which establishes assertion (i).

Turning to the proof of assertion (ii), we introduce

$$\bar{S}_P(u) = \mathbb{P}_\Sigma\{\widehat{D}_{\lfloor N/2 \rfloor}(\Sigma) \geq u\}, \quad u \geq 0. \quad (3.4.5)$$

By triangular inequality, we have with probability one:

$$\sup_{u\geq 0}\left|(K_h * \widehat{S}_N)(u) - (K_h * S_P)(u)\right| \leq \sup_{u\geq 0}\left|(K_h * \widehat{S}_N)(u) - (K_h * \bar{S}_P)(u)\right| + \sup_{u\geq 0}\left|(K_h * S_P)(u) - (K_h * \bar{S}_P)(u)\right|. \quad (3.4.6)$$

Observe that we almost surely have:

$$\sup_{u\geq 0}\left|(K_h * \widehat{S}_N)(u) - (K_h * \bar{S}_P)(u)\right| \leq \sup_{u\geq 0}\left|\widehat{S}_N(u) - \bar{S}_P(u)\right|. \quad (3.4.7)$$

By virtue of Dvoretsky-Kiefer-Wolfovitz inequality, we have, for all $t \geq 0$,

$$\mathbb{P}\left\{\sup_{u\geq 0}\left|\widehat{S}_N(u) - \bar{S}_P(u)\right| \geq t\right\} = \mathbb{E}\left[\mathbb{P}\left\{\sup_{u\geq 0}\left|\widehat{S}_N(u) - \bar{S}_P(u)\right| \geq t \mid \Sigma_1, \ldots, \Sigma_{\lfloor N/2 \rfloor}\right\}\right] \leq 2\exp(-2nt^2). \quad (3.4.8)$$

Let $s > 0$, we introduce the event, independent from Σ,

$$\mathcal{E}_{N,s} = \left\{\sup_{\sigma\in\mathfrak{S}_n}\left|\widehat{D}_{\lfloor N/2 \rfloor}(\sigma) - D_P(\sigma)\right| \leq s\right\}. \quad (3.4.9)$$

We almost-surely have: $\forall u \geq 0$,

$$\bar{S}_P(u) = \mathbb{P}_\Sigma\{D_P(\Sigma) \geq u + D_P(\Sigma) - \widehat{D}_{\lfloor N/2 \rfloor}(\Sigma)\}. \quad (3.4.10)$$

Consequently, on the event $\mathcal{E}_{N,s}$, it holds that: $\forall u \geq 0$,

$$(K_h{*}S_P)(u{+}s){-}(K_h{*}S_P)(u) \leq (K_h{*}\bar{S}_P)(u){-}(K_h{*}\widehat{S}_N)(u) \leq (K_h{*}S_P)(u){-}(K_h{*}S_P)(u{-}s), \quad (3.4.11)$$

as well as

$$\sup_{u\geq 0}\left|(K_h * S_P)(u) - (K_h * \bar{S}_P)(u)\right| \leq ||K'||_\infty(s/h), \quad (3.4.12)$$

since the mapping $K_h * S_P$ is differentiable, with derivative bounded by $||K'||_\infty/h$ in absolute value. Hence, using the union bound, combining Equation (3.4.6) with assertion (i) and Equation (3.4.8)-Equation (3.4.12), we get that for all $\delta \in (0,1)$, with probability larger than $1-\delta$:

$$\sup_{u\geq 0}\left|(K_h * \widehat{S}_N)(u) - (K_h * S_P)(u)\right| \leq \left(\sqrt{\log(4/\delta)} + ||l||_\infty\sqrt{\log(4n!/\delta)}\right)/\sqrt{2N}. \quad (3.4.13)$$

This proves assertion (ii). □

3.4.2 Trimming Algorithm for Consensus Ranking

As discussed in Section 3.3.3, stochastic transitivity greatly facilitates the computation of Kemeny's consensus, as shown in Proposition 3.3.16, as well as the verification of the maximality or monotonicity axioms, discussed in Propositions 3.3.12 and 3.3.14. However, although this occurs with a controlled probability (see Proposition 14 in Korba et al. (2017)), the empirical counterpart $\widehat{P}_N$ of a (strictly) stochastically transitive ranking distribution P can be of course non (S)ST. We propose below a trimming strategy based on the empirical ranking depth to recover a close (S)ST empirical ranking distribution and overcome this issue.

Algorithm 3.1: Ranking Depth Trimming

Input : Ranking dataset $\mathcal{D}_N = \{\Sigma_1, ...\Sigma_N\}$ and distribution $\widehat{P}_N = (1/N)\sum_{i=1}^N \delta_{\Sigma_i}$.
Output: Dataset $\mathcal{D} \subset \mathcal{D}_N$ of size $N_{\mathcal{D}} \leq N$ and (S)ST ranking distribution $\widehat{P}_{\mathcal{D}} = (1/N_{\mathcal{D}})\sum_{\sigma\in\mathcal{D}} \delta_\sigma$
- Initialize: $\mathcal{D} = \mathcal{D}_N$;
while $\hat{P}_{\mathcal{D}}$ *is not (S)ST* **do**
  - Determine the least deep rankings in $\mathcal{D}$: $\mathcal{O}_{\mathcal{D}} := \arg\min_{\sigma\in\mathcal{D}} D_{\hat{P}_N}(\sigma)$;
  - Update the ranking dataset $\mathcal{D} \setminus \mathcal{O}_{\mathcal{D}} \to \mathcal{D}$

Based on the ranking dataset $\mathcal{D}$ output by Algorithm 3.1, a (S)ST empirical distribution $\widehat{P}_{\mathcal{D}}$ can be computed, whose Kemeny consensus is obtained in a straightforward manner (Proposition 3.3.16) avoiding the search of solutions of an NP-hard minimization problem of type Definition 2.2.1, see Hudry (2008). As empirically supported by the experiments displayed in the next Section, this procedure allows for a fast, accurate, and robust recovery of consensus rankings. Indeed, the time complexity of Algorithm 3.1 is in $n\log(n)N^2\eta$, where n is the number of items, N is the number of samples. Indeed, $n\log(n)$ is the complexity of computing Kendall Tau distance for a pair of data using e.g. Merge Sort algorithm and N^2 to recompute the expected value of Kendall Tau to the whole dataset for every point of the dataset, and η is the (unknown) number of iterations required to obtain an SST dataset from a non-SST one.

Beyond the use of Algorithm 3.1 to recover an SST dataset from a noisy dataset, the important application of our trimming strategy arises when the said dataset is malevolently contaminated. When adding adversarial poisoning attacks to a natural, SST dataset, it is much more likely to be non-SST, as will be illustrated in Section 3.5.1.Under such kinds of attacks, the trimming algorithm described in Algorithm 3.1 becomes very handy to recover a robust consensus ranking in a tractable manner.

3.5 Applications

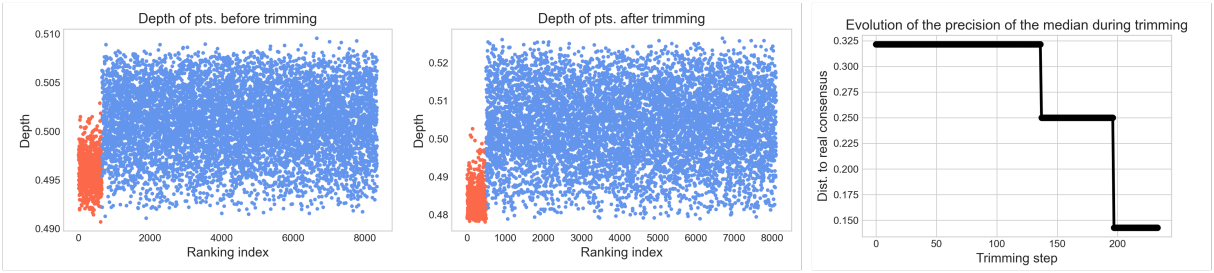
In order to illustrate the relevance of the ranking depth notion in the context of the robust consensus ranking task, we now show that our trimming strategy applied to the depth function can be used to find accurate and robust consensus, even in non-smooth settings. We provide both experiments and theoretical results demonstrating the robustness of medians based on depth.

In addition to that, this Section also illustrates the efficiency of using the depth function to perform additional tasks, including the following:

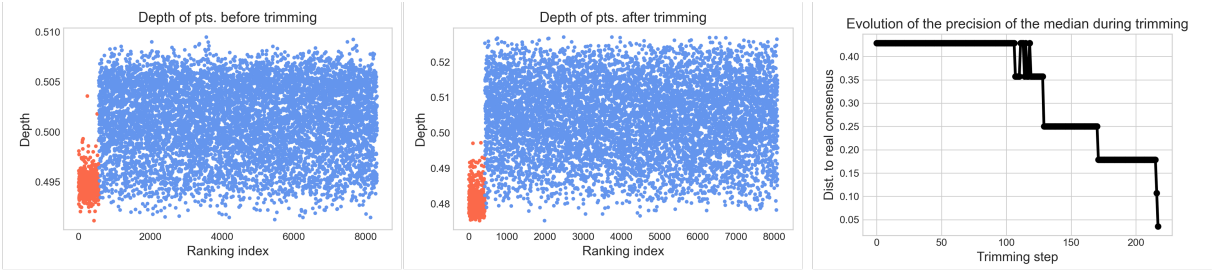
- Detection of outlying rankings: we can identify the least deep rankings and thus accurately distinguish anomalies in a dataset.
- Ranking data visualization: depth function can be used to visually make the difference between distributions or to get visual intuition e.g. on their shape.
- The two-sample (homogeneity) problem in $\mathfrak{S}_n$: depth can be used to distinguish distributions in a non-parametric way.

More generally, the depth function comes in very handy for usual applications involving rank statistics. The code for the experiments has been made publicly available here: github.com/RankingDepth/Ranking_depth_function.

3.5.1 Fast and Robust Consensus Rankings



(a) The trimming process removes 82 clean points and 430 adversarial points (out of 10000).



(b) The trimming process removes 85 clean points and 457 adversarial points (out of 10000).

Figure 3.2: Illustration of the trimming strategy: the blue points (resp. red points) correspond to the clean (resp. adversarial) points. For each row, the first plot (resp. second plot) shows the depth of points in the (contaminated) dataset before (resp. after) the trimming process. The third plot shows that the consensus computed after each trimming step gets closer and closer to the real consensus. In each case, the clean points are sampled from $M(\sigma_0, 0.1)$ and adversarial points from $M(\sigma_0^R, 1)$. The adversarial points represent 13% of the dataset (which has a total of 10000 points).

The trimming strategy proposed in Section 3.4.2 shows that we can recover smooth SST distributions from any empirical data, and solve the consensus ranking task by simply identifying the deepest ranking, which corresponds to Kemeny's consensus in the SST case: this procedure is fast, straightforward, and robust, in the sense that we can recover accurate medians even in contaminated settings. We support this claim with both experiments and a theoretical proposition below.

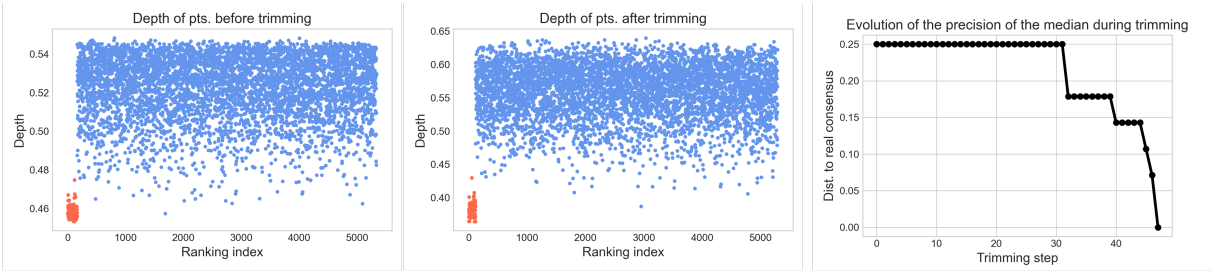
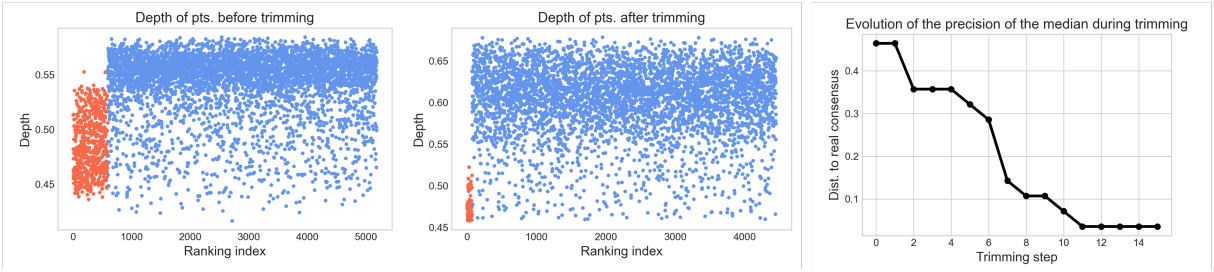
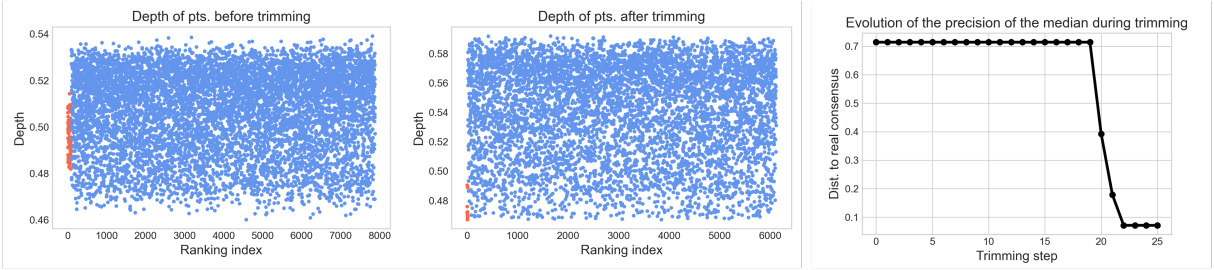


Figure 3.3: Trimming strategy when the contamination represents 25% of the dataset. The clean points are sampled from $M(\sigma_0, 0.4)$ and the adversarial ones from $M(\sigma_0^R, 2)$. The leftmost plot and the middle plot represent the depth of the points in the full dataset before and after trimming respectively. The rightmost plot shows how the consensus computed at each trimming step grows closer to σ_0 during the process. No clean point was removed during the trimming, whereas 1613 adversarial points were removed (out of 10000).

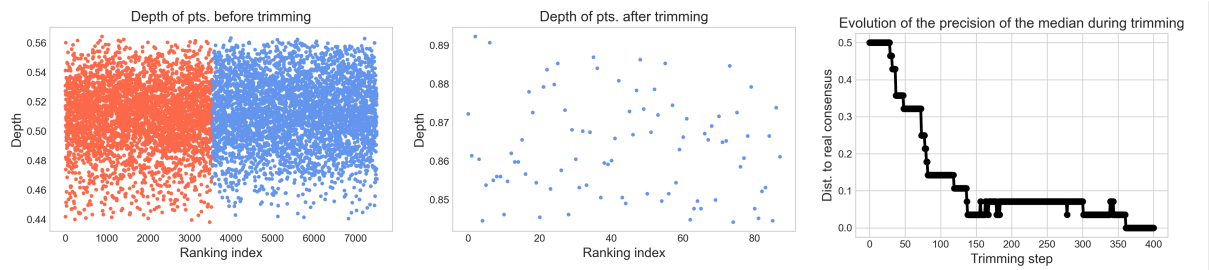


(a) The clean dataset is sampled from a Plackett-Luce (PL) distribution with random (but spread) parameters and σ_0 as consensus; the adversarial one is sampled from another PL distribution with random (but peaked) parameters and σ_0^R as consensus. The contamination represents 25% of the full dataset. 215 clean points were trimmed, and 2411 adversarial ones (out of 10000).

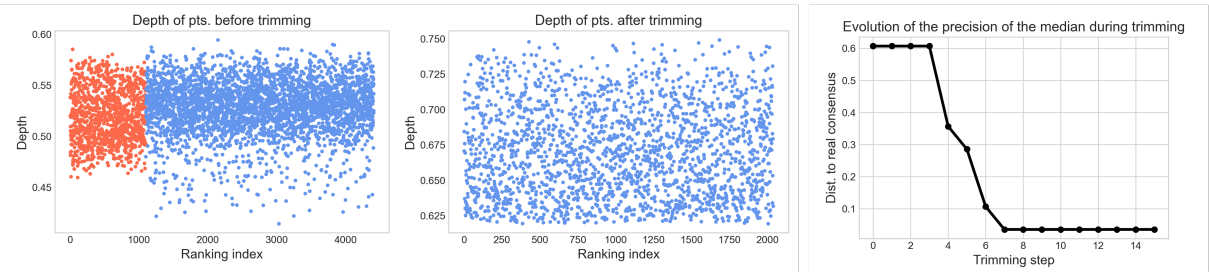


(b) The clean dataset is sampled from a Plackett-Luce (PL) distribution with random (but spread) parameters and σ_0 as consensus; the adversarial one is sampled from another PL distribution with random (but very peaked) parameters and σ_0^R as consensus. The contamination represents 10% of the full dataset. 1601 clean points were trimmed, and 965 adversarial ones (out of 10000).

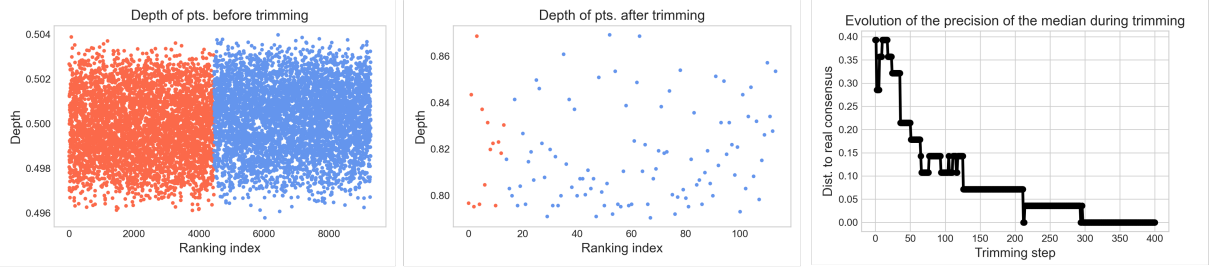
Figure 3.4: Illustration of the 'fixed' trimming strategy: the blue points (resp. red points) corresponds the clean (resp. adversarial) points. For each row, the first plot (resp. second plot) shows the depth of points in the (contaminated) dataset before (resp. after) the trimming process. The third plot shows that the consensus computed after each trimming step gets closer and closer to the real consensus. The trimming process removes 1% of the dataset at each step of the trimming, which is fixed to 15 steps for Figure 3.2a, 20 steps for Figure 3.2b.



(a) The clean dataset is sampled from a Plackett-Luce (PL) distribution with random parameters and σ_0 as consensus; the adversarial one is sampled from a similar PL distribution with σ_0^R as consensus. The contamination represents 48% of the full dataset. 4976 clean points were trimmed, and 4800 adversarial ones (out of 10000).



(b) The clean dataset is sampled from a Plackett-Luce (PL) distribution with random parameters and σ_0 as consensus; the adversarial one is sampled from another PL distribution with random (but quite peaked) parameters and σ_0^R as consensus. The contamination represents 40% of the full dataset. 1553 clean points were trimmed, and 4000 adversarial ones (out of 10000).



(c) The clean dataset is sampled from a Mallows distribution $M(\sigma_0, 0.1)$; the adversarial one is sampled from $M(\sigma_0, 0.1)$. The contamination represents 48% of the full dataset. 5079 clean points were trimmed, and 4785 adversarial ones (out of 10000).

Figure 3.5: Illustration of the 'fixed' trimming strategy in extreme cases: the blue points (resp. red points) corresponds the clean (resp. adversarial) points. For each row, the first plot (resp. second plot) shows the depth of points in the (contaminated) dataset before (resp. after) the trimming process. The third plot shows that the consensus computed after each trimming step gets closer and closer to the real consensus. The trimming process removes 1% at each step out of 400 steps (for Figures 3.5a and 3.5c) or 5% out of 15 (for Figure 3.5b).

Figure 3.2 illustrates the effectiveness of the trimmed-based consensus procedure in a reasonable contamination case. In this experiment, we consider a clean dataset drawn from a Mallows distribution $M(\sigma_0, 0.1)$ with $n = 8$ items, meaning that the distribution is quite spread. This clean dataset is contaminated with another Mallows distribution with opposite center, and which is less spread: $M(\sigma_0, 1)$. The clean and the adversarial datasets are merged together to form a general dataset of 10000 points, which is not SST. In the leftmost plots of Figure 3.2, the depth of each point is shown, and illustrates that if the adversarial dataset has a smaller depth in average, it is however not possible to clearly separate the two datasets using a unique depth threshold. This is where our recursive trimming procedure described in Algorithm 3.1 comes handy: it primarily removes adversarial points, and even if the number of points that are removed is small (approximatively 5% in each case), the trimmed-based consensus recovered after the trimming process is much closer to the real consensus σ_0 that the classical Kemeny's consensus, as illustrated by the rightmost plot in Figure 3.2.

Moreover, our trimming strategy can also apply to more extreme contamination setups. In Figure 3.3 for example, the adversarial dataset represents 25% of the full dataset. In that case, the clean dataset is sampled from $M(\sigma_0, 0.4)$ and the adversarial dataset from $M(\sigma_0^R, 2)$ with $n = 8$ items, meaning that the adversarial distribution is much more peaked, which thus explains the difference in depth that is clearly observable between the clean and adversarial points.

Figures 3.2 and 3.3 both illustrated the effectiveness of our trimming approach as defined by Algorithm 3.1. However, this version of the trimming procedure is restricted to cases where the full dataset is not SST, and the goal of the trimming procedure is to remove the least deep points until the recovered dataset is SST. However, we can in fact extend this procedure and trim any dataset to remove a fixed number of points. This strategy is also very effective to robustify a dataset and recover a better consensus than Kemeny's consensus. To provide an illustration of this version of our trimming procedure, we conducted experiments under various setups. In Figure 3.4, Plackett-Luce distributions were used to generate the clean and adversarial datasets. In both cases, the 'fixed' trimming strategy, which removed 1% of the dataset at each step (during 15 or 20 steps), led to the great improvement of the computed consensus.

Furthermore, this 'fixed' strategy also proves efficient in trickier, extreme cases where the contamination is very high. This situation is illustrated in Figure 3.5, where Figures 3.5a and 3.5c show the efficiency of the trimming strategy for Plackett-Luce and Mallows distribution when the contamination represents up to 48% of the dataset. In this case, when removing almost all the points when using the 'fixed' trimming procedure, the recovered consensus is once again much better than the classical Kemeny's consensus. The same conclusion can be drawn from Figure 3.5b, where 'only' 40% of the dataset is contaminated, but more points are removed at each step (5% instead of 1%) but fewer points are removed overall.

In all the setups presented in the experiments, the trimming procedure, either using the recursive SST version or the 'fixed' version, is very efficient to improve the consensus. This experimental result can be completed with a theoretical one, which provides an explanation for the efficiency of our method.

Theoretical robustness result. We derive specific robustness results when using depth-based trimming by computing the breakdown point, as defined in Definition 1.3.3 for classical versus trimmed statistics.

From a high-level perspective, we will consider the classical Borda count statistic previously defined in Definition 2.2.3 and studied in Dwork et al. (2001b); Fligner and Verducci (1988); Caragiannis et al. (2013); Collas and Irurozki (2021)) and a *depth-trimmed* Borda count statistic based on the scores $B_\mu(i) = \sum_{\sigma \in S_N} w(\sigma)\sigma(i)$, where $w(\sigma) = \mathbb{I}(D_N(\sigma) > \mu)$ (only the rankings with depth higher than μ are kept). Our goal will be to assess the robustness, via a *sample* version of the breakdown point, of these two statistics to compare them.

Here, we state that the classical Borda count statistic is less robust than the depth-trimmed one on generic distributions.

Proposition 3.5.1. *Let $\mu > 0$ be the trimming threshold and $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ a distribution such that $\mathbb{E}_{\Sigma\sim P}[D_P(\Sigma)] > \mu$. Let $\sigma^\star = \arg\max_{\sigma\in\mathfrak{S}_n} D_P(\sigma)$ be the deepest ranking and $\pi = \arg\max_{\sigma | d_\tau(\sigma^\star,\sigma)=\delta} D(\sigma)$ the ranking with highest depth among those at distance δ from the deepest ranking $\sigma^\star$. Then, the breakdown points for Borda and depth-trimmed-Borda on P are related as follows,*

$$\frac{\epsilon_\delta^B(P)}{\epsilon_\delta^{DT\text{-}B}(P)} < \frac{D_P(\pi)}{\mu} < 1. \tag{3.5.1}$$

Proposition 3.5.1 refers to the robustness of the depth-trimmed-Borda compared to the classical Borda. In the following pages, we will in fact prove some auxiliary results as well as a generalization of this proposition.

Let us first recall some definitions and results about the Borda estimators. Borda is an approximation to the barycentric ranking median (which is NP-hard for $n > 4$, see for example Dwork et al. (2001b)) for a sample of complete rankings drawn from a Mallows model, as shown in Fligner and Verducci (1988). Moreover, Borda is quasi-linear in time and outputs the correct median with high probability with a polynomial number of samples, as shown in Caragiannis et al. (2013). A robust aggregation procedure for top-k rankings in very noisy settings is proposed in Collas and Irurozki (2021).

As a reminder, the Borda count statistic is defined as follows in Section 2.2.1:

Definition 2.2.3. BORDA COUNT. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. The Borda count of an item $i \in [n]$ for distribution P is defined by:*

$$B_P(i) = \sum_{\sigma\in\mathfrak{S}_n} P(\sigma)\sigma(i) \tag{2.2.2}$$

Then, the Borda statistics is given by:

$$T_{Borda}(P) \in \text{argsort}(B_P), \tag{2.2.3}$$

where $\text{argsort}(s) = \{\sigma \in \mathfrak{S}_n, \forall r \in [n-1], s_{\sigma^{-1}(r)} \leq s_{\sigma^{-1}(r+1)}\}$

We define the depth-weighted-Borda as a generalization of the classic and depth-trimmed-Borda in which there exists a weight associated with each ranking. It generalizes Borda in

the following way: For each item i, the Borda score is computed as $B(i) = \sum_{\sigma \in X} w(\sigma)\sigma(i)$. The final estimator for the median is the ranking that orders the items by their Borda score. The depth-weighted-Borda is equivalent to replicating the rankings proportionally to their weight. This analysis generalizes to any weights that correspond to an *increasing* function of the depths. In particular, the depth-trimmed-Borda is the case of depth-weighted-Borda in which $w(\sigma) = \mathbb{I}\{D(\sigma) > \mu\}$.

We settle here the notation for the following lines. We denote by $S_N \sim P$ a sample of rankings (of size N) and A an adversarial sample.

Definition 3.5.2. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a metric and $T : P \in \mathcal{M}_+^1(\mathfrak{S}_n) \mapsto \mathfrak{S}_n$ be a statistic. Let us write $S_N \sim P$ a sample drawn from P of size N and $\sigma_{S_N}^T$ the consensus based on the estimator method T on sample S_N.*

The estimator T is said to be δ-broken on P, l and for sample size N if for any $S_N \sim P$ of size N, there exists an adversarial sample A such that $l(\sigma_{S_N}^T, \sigma_{S_N \cup A}^T) \geq \delta$.

The next result characterizes the cardinality of a sample that breaks the Borda estimator of a sample S_N distributed according to P. This is an auxiliary result for Proposition 3.5.1.

Proposition 3.5.3. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution and $S_N \sim P$. Let A^- be the adversarial sample that δ-breaks the Borda estimator on Kendall Tau distance d_τ for sample size N such that A^- is of minimal cardinality.*

Let $\bar{r}_N(i) = N^{-1} \sum_{\sigma \in S_N} \sigma(i)$ and $\bar{r}(i) = (\#A^-)^{-1} \sum_{\sigma \in A^-} \sigma(i)$ be the average ranking of item i in S_N and A^- respectively. Finally, let $\bar{R}$ be the ordered vector composed of $\frac{\bar{r}_N(j) - \bar{r}_N(i)}{\bar{r}(i) - \bar{r}(j)}$ for all (i, j) such as both the numerator and denominator are positive. Then

$$\#A^- = \left\lceil N \left[\bar{R} \right]_{(\delta)} \right\rceil \tag{3.5.2}$$

where $[x]_{(\delta)}$ denotes the δ-th quantile[1] of a vector x.

Proof By definition, A^- δ-breaks Borda if and only if the following holds.

$$d(\sigma_{S_N}^B, \sigma_{S_N \cup A^-}^B) = \delta \tag{3.5.3}$$

$$\Leftrightarrow \delta = \#\{(i < j) : \sum_{\sigma \in S_N} \sigma(i) + \sum_{\sigma \in A^-} \sigma(i) \geq \sum_{\sigma \in S_N} \sigma(j) + \sum_{\sigma \in A^-} \sigma(j)\} \tag{3.5.4}$$

$$\Leftrightarrow \delta = \#\{(i < j) : \sum_{\sigma \in S_N} \sigma(i) - \sigma(j) \geq \sum_{\sigma \in A^-} \sigma(j) - \sigma(i)\} \tag{3.5.5}$$

$$\Leftrightarrow \delta = \#\{(i < j) : \sum_{\sigma \in S_N} \sigma(j) - \sigma(i) \leq \sum_{\sigma \in A^-} \sigma(i) - \sigma(j)\} \tag{3.5.6}$$

$$\Leftrightarrow \delta = \#\{(i, j) : 0 < \sum_{\sigma \in S_N} \sigma(j) - \sigma(i) \leq \sum_{\sigma \in A^-} \sigma(i) - \sigma(j)\} \tag{3.5.7}$$

From a statistical perspective, we can bound the cardinality of A^- as follows: let (i, j) be a pair of indexes belonging to the set defined just above.

$$\sum_{\sigma \in S_N} \sigma(j) - \sigma(i) \leq \sum_{\sigma \in A^-} \sigma(i) - \sigma(j) \tag{3.5.8}$$

[1] the δ-th quantile of vector x is the smallest element of x that is larger than (or equal to) δ% of the elements of x. For example, the 0.2-th quantile of $(1, 2, 3, 4, 5, 6, 7, 9, 10)$ is 2.

$$\Leftrightarrow N\left(\bar{r}_N(j) - \bar{r}_N(i)\right) \leq \#A^-(\bar{r}(i) - \bar{r}(j)) \tag{3.5.9}$$

$$\Rightarrow \#A^- \geq \frac{N\left(\bar{r}_N(j) - \bar{r}_N(i)\right)}{\bar{r}(i) - \bar{r}(j)}, \tag{3.5.10}$$

which holds for exactly δ pairs of items (i, j). We conclude the proof by recalling that A^- is of minimal cardinality. □

The next auxiliary result shows that provided certain conditions, if a sample breaks the depth-weighted-Borda then it breaks Borda.

Proposition 3.5.4. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution and $S_N \sim P$. Let A^- (resp. A_w^-) be the adversarial sample that δ-breaks the Borda (resp. depth-weighted Borda) estimator on Kendall Tau distance d_τ for sample size N such that A^- (resp. A_w^-) is of minimal cardinality.*

Let $\bar{r}_N(i) = N^{-1}\sum_{\sigma\in S_N}\sigma(i)$ and $\bar{r}_w(i) = (\#A_w^-)^{-1}\sum_{\sigma\in A_w^-}\sigma(i)$ be the average ranking of item i in S_N and A_w^- respectively. Let $\pi_w = \arg\max_{\sigma\in A_w^-} w(\sigma)$ and $\mu = w(\pi_w)$ be the weight of maximum depth for adversarial rankings.

Finally, suppose $\hat{P}_N$ and w satisfy: $\mathbb{E}_{\hat{P}_N}(w(\Sigma)) > w(\pi_w) = \mu$ and $\forall\ (i,j)$ s.t. $\mathbb{E}_{\hat{P}_N}(\Sigma(i) < \Sigma(j))$, $\mathbb{E}_{\hat{P}_N}[w(\Sigma)(\Sigma(j) - \Sigma(i))] \geq \mathbb{E}_{\hat{P}_N}[w(\Sigma)]\mathbb{E}_{\hat{P}_N}[\Sigma(j) - \Sigma(i)]$ (these two assumptions enforce the use of a weight function that is in accordance with $\hat{P}_N$). Then, the cardinality of A^- and A_w^- are related as follows:

$$\#A_w^- \geq \frac{N^{-1}\sum_{\sigma\in S_N} w(\sigma)}{\mu}\#A^-. \tag{3.5.11}$$

Proof Since A_w^- δ-breaks the depth-weighted-Borda, we can follow the same proof outline as for Proposition 3.5.3 and bound the cardinality $\#A_w^{-1}$ as follows,

$$\sum_{\sigma\in S_N} w(\sigma)(\sigma(j) - \sigma(i)) \leq \sum_{\sigma\in A_w^-} w(\sigma)(\sigma(i) - \sigma(j)) \tag{3.5.12}$$

$$\Rightarrow N \times N^{-1}\sum_{\sigma\in S_N} w(\sigma)(\sigma(j) - \sigma(i)) \leq \#A_w^- w(\pi)(\bar{r}_w(i) - \bar{r}_w(j)) \tag{3.5.13}$$

$$\Rightarrow \#A_w^- \geq \frac{N\left(\bar{r}_n(j) - \bar{r}_N(i)\right)}{\bar{r}_w(i) - \bar{r}_w(j)}\frac{N^{-1}\sum_{\sigma\in S_N} w(\sigma)}{\mu} \tag{3.5.14}$$

Since $\frac{N^{-1}\sum_{\sigma\in S_N} w(\sigma)}{\mu}$ is independent of i, j and A_w^- also δ-breaks the Borda estimator:

$$\#A_w^- \geq \#A^-\frac{N^{-1}\sum_{\sigma\in S_N} w(\sigma)}{\mu}. \tag{3.5.15}$$

□

We are finally ready to prove a generalization of our Proposition 3.5.1. Let us first define our notion of δ-*breakdown point*, which extends the classical concept.

Definition 3.5.5. SAMPLE BREAKDOWN POINT. *$P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, let $T : P \in \mathcal{M}_+^1(\mathfrak{S}_n) \mapsto \mathfrak{S}_n$ be a statistic and $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ a metric. The δ-sample breakdown point for statistic T with respect to distribution P and metric l is defined as the smallest cardinality of an adversarial sample that δ-breaks T in the limit when $N \to \infty$ for distribution P.*

More specifically, $\epsilon_{\delta,l}^T(P) = \min \#A$ s.t. $\lim_{N\to\infty} l(\sigma_{S_N}^T, \sigma_{S_N \cup A}^T) = \delta$

In the following proposition, we write $\epsilon_\delta^B(P)$ (resp. $\epsilon_\delta^{DW-B}(P)$) the δ-breakdown point for the Borda (resp. depth-weighted Borda) estimator with respect to distribution P and $l = d_\tau$ the Kendall Tau distance.

Proposition 3.5.6. BREAKDOWN POINTS RATIO. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution and $w : P \in \mathcal{M}_+^1(\mathfrak{S}_n) \to \mathbb{R}_+$ a weight function. Suppose that $\mathbb{E}_P[w(\Sigma)] > w(\pi)$, where $\pi = \arg\max_{\sigma \mid d_\tau(\sigma^\star,\sigma)=\delta} w(\sigma)$ and $\sigma^\star = \arg\max_{\sigma\in\mathfrak{S}_n} D_P(\sigma)$.In addition, suppose that the following condition holds: $\forall(i,j)$ s.t. $\mathbb{E}_P(\Sigma(j) - \Sigma(i)) > 0$, $\mathbb{E}_P(w(\Sigma)(\Sigma(j) - \Sigma(i))) \geq \mathbb{E}(w(\Sigma))\mathbb{E}(\Sigma(j) - \Sigma(i))$. Then,*

$$\lim_{N\to\infty} \frac{\epsilon_\delta^B(P)}{\epsilon_\delta^{DW-B}(P)} < \frac{w(\pi)}{\mathbb{E}_P[w(\Sigma)]} < 1. \tag{3.5.16}$$

Proof We start by noting that for S_N to be δ-broken then the adversarial sample has to be at least at distance δ regardless of the distribution for the weights. Then, we denote $z = \mathbb{E}_P[w(\Sigma)]/w(\pi) = \lim_{N\to\infty} N^{-1} \sum_{\sigma\in S_N} w(\sigma)/w(\pi)$ (by the law of large numbers) and take Proposition 3.5.4 to write the limiting ratio of the breakdown points when the number of samples tends to infinity as follows.

$$\lim_{N\to\infty} \frac{\epsilon_\delta^B(P)}{\epsilon_\delta^{DW-B}(P)} = \lim_{N\to\infty} \frac{\frac{\#A^-}{\#A^-+N}}{\frac{\#A_w^-}{\#A_w^-+N}} < \lim_{N\to\infty} \frac{\frac{\#A^-}{\#A^-+N}}{\frac{\#A^-\cdot z}{\#A^-\cdot z+N}} < \frac{1}{z} = \frac{w(\pi)}{\mathbb{E}_P[w(\Sigma)]} < 1 \tag{3.5.17}$$

This is the main result related to the robustness of the Borda median estimator. It shows that the breakdown point of Borda is smaller than the breakdown point for the depth-trimmed-Borda provided certain conditions. We denote by μ the threshold of the depth-trimmed-Borda.

Then, our Proposition 3.5.1 is straightforward when we choose the weight function w so that $w(\sigma) = \mathbb{I}(D_P(\sigma) \geq \mu)$ in Proposition 3.5.6. □

3.5.2 Other Applications

Outlier detection in ranking data.

We now place ourselves in a situation where a single sample of rankings is observed. For simplicity, we consider the case where the underlying ranking distribution is an unbalanced

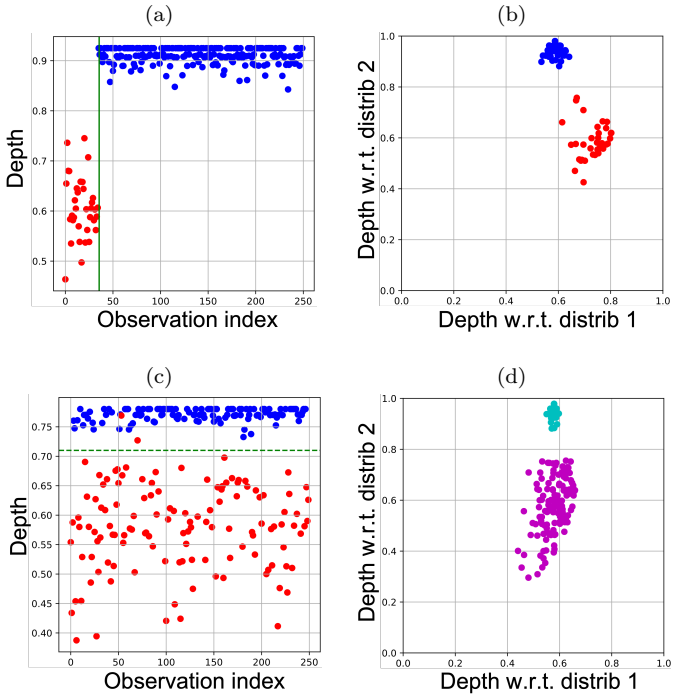


Figure 3.6: Depth plots (a,c) and DD-plots (b,d) for a mixture of Mallows-Kendall distributions. (a)-(b): distant centers and different sizes for the two components of the mixture. (c)-(d): closer centers and same size.

mixture of two Mallows distributions (for $n = 10$), strongly differing in size ($N_1 = 35$ and $N_2 = 215$), with distant centers ($d_\tau(\sigma_1^\star, \sigma_2^\star) = 15$) and parameters $\theta_1 = 0.5$ and $\theta_2 = 2.5$. Figure 3.6 (a) shows the ranking depth (relative to Kendall Tau) of each observation computed with respect to the entire sample. We observe, that despite the unavailability of labels, the ranking depth clearly distinguishes the two different components. It thus permits to perform a typical anomaly detection task in the context of ranking data, where the differing minority of permutations are viewed as abnormal rankings. The diagnostic ranking DD-plot (b) based on the identified information about the components confirms the differences.

Consider next the case of a mixture with closer centers ($d_\tau(\sigma_1^\star, \sigma_2^\star) = 11$) and equal sizes ($N_1 = N_2 = 125$), with parameters $\theta_1 = 0.25$ and $\theta_2 = 2.5$. The depth plot (c) w.r.t. to the entire sample reflects how easily we can cluster the ranking dataset into two components (we deliberately shuffle the indices and keep colors for illustrative purposes), and we suggest a separating threshold (on the level of depth $= 0.71$), which in this particular case allows for two mistaking assignments. For the diagnostic ranking DD-plot (d), we honestly include this mistake and change the colors to underline this impurity.

Graphical methods and visual inference.

The analysis of rankings suffers from the lack of graphical displays and diagrams, such as *probability plots* or *histograms*, for gaining insight into the structure of the data. Ranking depths can be readily used to design a visual diagnostic tool for ranking data, extending the Depth *vs.* Depth plot (DD-plot in abbreviated form) were originally introduced by

Position	$d_\tau(\sigma_1^\star, \sigma_2^\star)$	θ_1	θ_2	N_1	N_2
(a)	15	1	1	250	250
(b)	0	0.5	2	250	250
(c)	15	0.5	2	250	250
(d)	15	0.5	2	400	100

Table 3.1: Parameters for pairs of samples drawn from Mallows-Kendall distribution used for Figure 3.7.

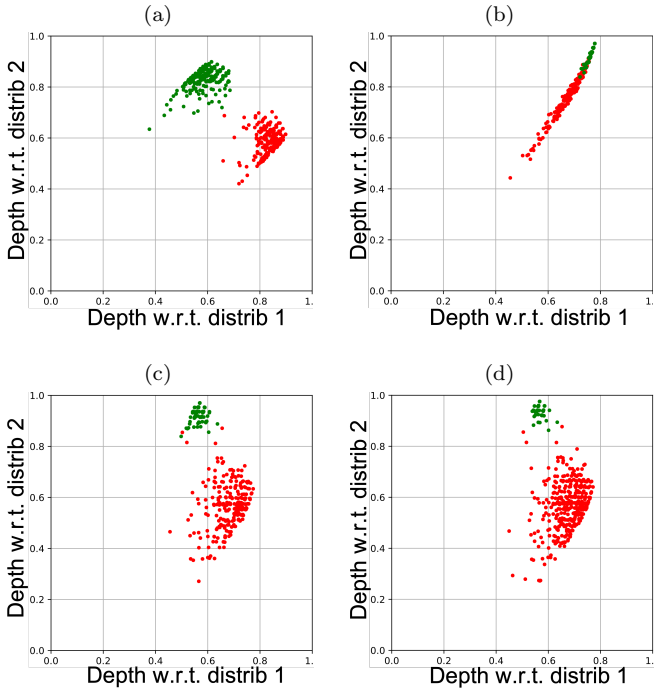


Figure 3.7: Ranking DD-plot corresponding to Mallows distributions with parameters described in Table 3.1.

Liu et al. (1999) for multivariate data. For two samples of rankings $\mathbf{\Sigma^1} = \{\sigma_1^1, \ldots, \sigma_{N_1}^1\}$ and $\mathbf{\Sigma^2} = \{\sigma_1^2, \ldots, \sigma_{N_2}^2\}$, with corresponding empirical measures $\widehat{P^1}_{N_1}$ and $\widehat{P^2}_{N_2}$, the ranking DD-plot is obtained by plotting in the Euclidean plane the points:

$$\left\{\left(D_{\widehat{P^1}_{N_1}}(\sigma), D_{\widehat{P^2}_{N_2}}(\sigma)\right) : \sigma \in \mathbf{\Sigma^1} \cup \mathbf{\Sigma^2}\right\}. \tag{3.5.18}$$

Depending on the distance d chosen, such a plot allows to reflect the location and scatter of two distributions on $\mathfrak{S}_n$, and their mutual position. To illustrate its diagnostic capacity, we plot in Figure 3.7 the ranking DD-plots relative to the Kendall Tau distance and four pairs of samples stemming from Mallows distribution with parameters defined in Table 3.1. In this and subsequent figures, the depth is re-scaled to $[0, 1]$ by diving by $\|d_\tau\|_\infty$. A few remarks can be made: For distributions differing in: 1) location only (a), the ranking DD-plot is symmetric with respect to the diagonal, 2) scatter only (b), observations from one distribution will be attributed systematically higher depth values, 3) both location and scatter (c), the distributions can be distinguished and 4) the number of the observations, it does not influence the general picture (d).

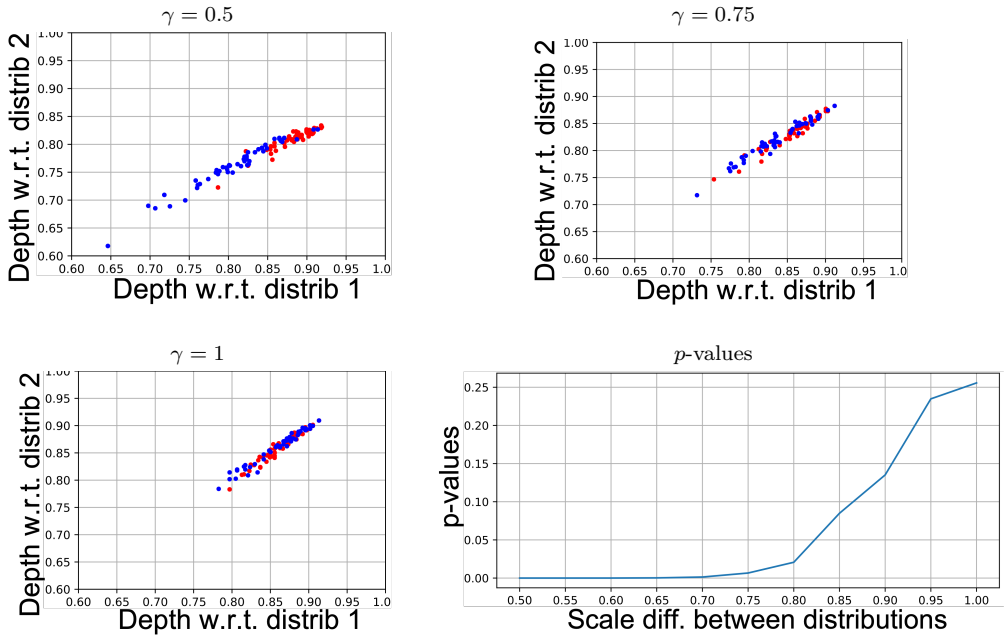


Figure 3.8: *DD*-plots of a pair of P-L distributions with gradually decreasing difference between them based on parameter γ and the corresponding average p-values for the test of homogeneity.

Rankings - Homogeneity testing.

Depth can further be used to provide a formal inference, which we exemplify as a non-parametric test of homogeneity between two Plackett-Luce distributions (Critchlow et al., 1991) with $n = 10$. The first one (red in Figure 3.8) is generated using the parameters $\boldsymbol{w}_1 = (\mathrm{e}^9, ..., \mathrm{e}^0)$, the second one represents its changed version $\boldsymbol{w}_2 = (\mathrm{e}^{\gamma 9}, ..., \mathrm{e}^{\gamma 0})$. We gradually increase γ from 0.5 (substantial difference) to 1 (equal in distribution), and provide the p-values of the Wilcoxon rank-sum test averaged over 100 repetitions in Figure 3.8. The test is performed using the reference sample (of size 500) from the first distribution, with tested sample sizes being equal (= 50) for both distributions (see Lafaye De Micheaux et al. (2020) for details on the testing procedure and Liu and Singh (1993) for more details). Figure 3.8 shows how the p-values detect very well the difference between the two distributions when it is the case, giving a formal inference to the ranking *DD*-plot visualization, whereas, remarkably, the (parametric) nature of the underlying ranking models is not used at all by the procedure. We also underline that, in a similar way, ranking depth-based *goodness-of-fit* statistics could be computed, in order to evaluate how well a specific ranking model fits a ranking dataset.

Student dataset. We now explore our homogeneity testing machinery on a real dataset (available at `https://github.com/ekhiru/students-dataset`) composed of rankings from students (with a ground truth answer) before (red) and after (blue) taking the related course. The diagnostic *DD*-plot of the two cohorts together with p-values over 1000 random repetitions and the asymptotic density under H_0 are indicated in Figure 3.9: they illustrate the improvement of the students' knowledge after the class.

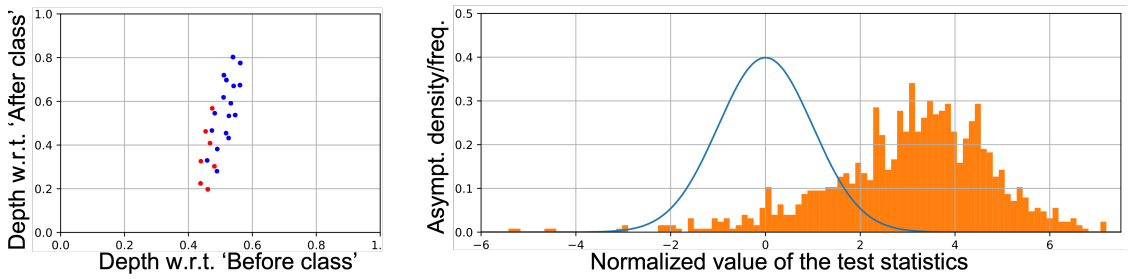


Figure 3.9: Left: DD-plot for 'before class' (red) and 'after class' (blue) students. Right: p-values of the homogeneity test.

3.6 Conclusion

In this Chapter, our focus has been on extending the concept of statistical depth to the domain of ranking data. By doing so, we aimed to overcome the inherent challenges posed by the absence of natural order and vector space structure in $\mathfrak{S}_n$, as well as the NP-hardness of solving the consensus ranking task using Kemeny's aggregation procedure in a general and adversarial setting.

We began by outlining the essential properties that a ranking depth should possess in order to effectively capture quantiles, order statistics, and ranks. Through our exploration, we discovered that using a metric-based approach, commonly used in consensus ranking, allows us to construct depth functions on $\mathfrak{S}_n$ that fulfill these properties in various scenarios. Moreover, we established theoretical results that demonstrate the accurate estimation of ranking depths and related quantities through empirical versions, with reliable guarantees.

To enhance the robustness of the consensus ranking problem in practical applications, we devised an efficient trimming strategy. This strategy enables us to recover a more robust consensus under adversarial conditions. Empirical evaluations on different datasets, as well as theoretical analyses applied to the Borda count statistic, showcased the positive impact of our trimmed statistic on enhancing robustness. Additionally, we highlighted the versatility of depth functions in various tasks, such as ranking data visualization, outlying ranking detection, and homogeneity testing.

While our findings have provided promising results in bolstering the robustness of the consensus ranking problem, there is still a need for further analysis of the robustness offered by different statistics. While we demonstrated the higher robustness of the trimmed Borda count statistic compared to the classical Borda count statistic through theoretical means, the extent of this improvement and its generalizability to other statistics remain unknown. This limitation serves as a primary focus for the next chapter, where we aim to address and explore this aspect in greater depth.

Chapter 4

Evaluating and Enhancing Robustness in Consensus Ranking

Once is happenstance. Twice is coincidence. Three times is enemy action

Ian Flemming

Contents

4.1 Introduction and High-level Overview of the Contributions

In Chapter 3, we introduced the trimmed Kemeny's aggregation statistic as the initial solution to tackle the lack of robustness in the consensus ranking task. Extensive experimentation demonstrated the effectiveness of this method in practical scenarios. However, a comprehensive evaluation is still needed to precisely quantify the robustness gained by robust statistics compared to classical statistics.

In this chapter, our focus is on introducing an approximation algorithm specifically designed to assess the robustness of any statistic based on its breakdown point, while also addressing the associated computational challenges. This robustness evaluation method provides a valuable tool for measuring the resilience of different statistics in the face of adversarial scenarios.

Furthermore, we present a robust statistic plugin that can enhance the robustness of any classical statistic employed in solving the consensus ranking problem. Importantly, our proposed method not only offers significant gains in robustness but also ensures minimal loss in precision. This characteristic sets our approach apart, positioning it as a superior alternative to existing methods such as Kemeny's aggregation to solve the consensus ranking task in both a precise and robust manner.

By leveraging these advancements, we aim to provide a comprehensive framework for evaluating and improving the robustness of consensus ranking statistics. Through rigorous analysis and empirical evaluations, we demonstrate the practical benefits of our proposed methods and their potential to outperform traditional approaches.

4.1.1 Outline of the Rationales of the Chapter

In the literature devoted to robustness for rankings, the well-known Gibbard-Satterthwaite theorem Gibbard et al. (1973); Satterthwaite (1975) states that every reasonable *voting rule* (in social choice theory, consensus medians are identified with voting rules) can be manipulated. We point out that there has been a wide body of research devoted to characterizing the complexity of computing manipulations, NP-hardness result on manipulation being considered as a guarantee for robustness Bartholdi III et al. (1989); Davies et al. (2011); Brandt et al. (2016). However, beyond-worst-case analysis shows that the problems are easy in practice Zuckerman et al. (2009), as illustrated in Sections 2.2.3 and 3.5.1.

In the Chapter, we complement Chapter 3 on the issue of robustness to vote manipulation by investigating how the concept of breakdown point may apply to consensus ranking in practice. As will be shown, one of the main difficulties faced in the considered context lies in the fact that consensus rankings are often obtained by solving an optimization problem and that no closed analytical form for the solutions is available in general. Consequently, the computation of breakdown points of ranking statistics is generally a computational challenge. Our main proposal here consists in approximating this computation by solving a relaxation of the breakdown point optimization problem by using a smoothing technique that allows for computing relevant gradients and eventually perform gradient descent.

Moreover, we also provide a robust plugin that can be added on top of any consensus ranking statistic. Beyond the trimmed Kemeny's statistic provided in Chapter 3 that stems from the classical trimmed mean or median from the literature on robustness for real-numbered data, as presented in Definition 1.3.6, we take advantage of the specific structure of the ranking space, namely the symmetric group $\mathfrak{S}_n$, to provide a specific robustification method. The idea is to relax the constraint stipulating that the summary of a ranking distribution should be necessarily represented by a single ranking (*i.e.* a strict order on the set of items indexed by $i \in \{1, \ldots, n\}$), or equivalently by a point mass on $\mathfrak{S}_n$. Instead, we suggest summarizing a ranking distribution by a *bucket ranking* (*i.e.* a weak order on the set $\{1, \ldots, n\}$), the possibility of observing ties in the considered orderings being shown to have crucial advantages regarding robustness.

4.1.2 Outline of the Main Contributions of the Chapter

In order to provide the approximation algorithm for the breakdown point and the robustification plugin based on bucket rankings, Section 4.2 will first recall the necessary concepts in consensus ranking and robustness, as well as the previous results from the literature on this topic. Section 4.3 focuses on robustness, by detailing our theoretical results on the breakdown functions for the classical consensus ranking statistics and extending this concept to bucket rankings. In Section 4.4, we provide an optimization algorithm to estimate the breakdown function in practice. Section 4.5 is dedicated to the definition of our robust plugin, called the Downward Merge statistic. Finally, experiments are conducted in Section 4.6 to highlight the usefulness of our Downward Merge plugin for solving robust consensus ranking tasks.

The main contributions are summarized below:

- A theoretical evaluation of the robustness, measured by the breakdown function, of classical consensus ranking statistics is provided. More precisely, we uncover a general lower-bound for their breakdown function, and an upper-bound for Kemeny's consensus.
- We provide a practical algorithm that approximates the breakdown function of any consensus ranking statistics. This algorithm can adapt to statistics outputting a single ranking or a bucket ranking.
- We provide an extension of the relevant concepts (metrics and distances, breakdown function, etc.) for bucket rankings. Notably, we provide two relevant Hausdorff-based extensions of the classical metrics such as Kendall Tau to the space of weak orders.
- We create a plugin called the Downward Merge plugin that provides a robust layer on top of classical consensus ranking statistics. The Downward Merge plugin is shown to be empirically very effective in robustifying consensus ranking with minimal loss in precision: it thus provides a more advantageous choice of statistics compared to classical alternatives.

4.2 Framework and Problem Statement

We start with a reminder of key concepts in ranking data analysis and robust statistics, mainly using concepts introduced in Chapter 1, which can be completed with Alvo and Yu (2014); Huber and Ronchetti (2009) for more details. Recall that a ranking over a set of $n \geq 1$ items is represented as a permutation $\sigma \in \mathfrak{S}_n$ where $\mathfrak{S}_n$ is the symmetric group. By convention, the rank r of an item $i \in [n]$ is $r = \sigma(i)$. For any measurable space $\mathcal{X}$, $\mathcal{M}_+^1(\mathcal{X})$ is the set of probability measures on $\mathcal{X}$, $\mathrm{TV}(p, q)$ the total variation distance between p and q in $\mathcal{M}_+^1(\mathcal{X})$.

4.2.1 Ranking Data and Summary Statistics

The descriptive analysis of probability distributions, or datasets for their empirical counterparts, is a fundamental problem in statistics. For distributions on Euclidean spaces such as $\mathbb{R}^d$, this problem has been widely studied and covered by the literature, with the study of statistics ranging from the simplistic sample mean to more sophisticated data functionals, such as $U/L/R/M$-statistics or depth functions, see for instance van der Vaart (1998).

Defining similar notions for probability distributions on $\mathfrak{S}_n$, the space of rankings, is challenging due to the absence of vector space structure and to the combinatorial nature of the space. However, fueled by the recent surge of applications using preference data, such as *e.g.* recommender systems, the statistical analysis of ranking data has recently regained attention and certain classic problems have been revisited, as for instance those related to consensus rankings and their generalization ability (see for example Korba et al. (2017) and the references therein), or to the extension of depth functions to ranking data as developed in Chapter 3.

Location Estimation Task. Statistics measuring centrality, such as the mean (or the median for univariate distribution), can be seen as barycenters of the sampling observations w.r.t a certain distance. Consensus ranking extends this idea to probability distributions on $\mathfrak{S}_n$, as in Deza and Deza (2009). As a reminder, this consensus ranking task is defined as follows:

Definition 2.2.1. Classical Consensus Statistics. *Let $l : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ be a distance on rankings. A classical consensus statistics is a function $T_l : \mathcal{M}_+^1(\mathfrak{S}_n) \to \mathfrak{S}_n$ solving the following optimization problem: $\forall P \in \mathcal{M}_+^1(\mathfrak{S}_n)$,*

$$T_l(P) \in \operatorname*{argmin}_{\sigma \in \mathfrak{S}_n} \mathbb{E}_{\Sigma \sim P}(l(\Sigma, \sigma)), \tag{2.2.1}$$

The output of statistics T_l is usually denoted by $\sigma_l^\star$ (where the dependence in P is dropped when the context is clear) and is simply called the consensus.

The most famous instance of this problem is Kemeny's consensus, which corresponds to the situation where l is the Kendall Tau distance:

Definition 2.1.3. Kendall Tau distance. *The Kendall Tau distance, denoted as $d_\tau : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{N}$ is defined as:*

$$\forall \sigma_1, \sigma_2 \in \mathfrak{S}_n, \quad d_\tau(\sigma_1, \sigma_2) = \sum_{i<j} \mathbb{1}[(\sigma_1(i) - \sigma_1(j))(\sigma_2(i) - \sigma_2(j)) < 0], \tag{2.1.1}$$

Another common choice is the Borda count when l is the Spearman's Rho distance, and recalled here:

Definition 2.2.3. BORDA COUNT. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution. The Borda count of an item $i \in [n]$ for distribution P is defined by:*

$$B_P(i) = \sum_{\sigma \in \mathfrak{S}_n} P(\sigma)\sigma(i) \tag{2.2.2}$$

Then, the Borda statistics is given by:

$$T_{Borda}(P) \in \text{argsort}(B_P), \tag{2.2.3}$$

where $\text{argsort}(s) = \{\sigma \in \mathfrak{S}_n, \forall r \in [n-1], s_{\sigma^{-1}(r)} \leq s_{\sigma^{-1}(r+1)}\}$

Moreover, the Borda count is a $\mathcal{O}(n \log n)$, 5-approximation of the Kemeny ranking as shown in Caragiannis et al. (2013); Jiao et al. (2016); Coppersmith et al. (2010), which is NP-hard to compute as shown in Dwork et al. (2001a). Here are recalled Spearman's Rho, as well as Spearman's Footrule distances.

Definition 2.1.4. SPEARMAN'S FOOTRULE DISTANCE. *The Spearman's Footrule distance, denoted as $d_1 : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{N}$ is defined as:*

$$\forall \sigma_1, \sigma_2 \in \mathfrak{S}_n, \quad d_1(\sigma_1, \sigma_2) = \sum_{i=1}^{n} |\sigma_1(i) - \sigma_2(i)|, \tag{2.1.2}$$

Definition 2.1.5. SPEARMAN'S RHO DISTANCE. *The Spearman's Rho distance, denoted as $d_2 : \mathfrak{S}_n \times \mathfrak{S}_n \to \mathbb{R}_+$ is defined as:*

$$\forall \sigma_1, \sigma_2 \in \mathfrak{S}_n, \quad d_2(\sigma_1, \sigma_2) = \left(\sum_{i=1}^{n} (\sigma_1(i) - \sigma_2(i))^2 \right)^{1/2}, \tag{2.1.3}$$

In this Chapter, we will focus on Kendall Tau distance as it better captures pairwise item comparisons in its formulation.

4.2.2 Robust Statistics

To evaluate the robustness of a statistic, the notion of *breakdown function* has been introduced in the seminal work of Huber (1964) and exposed in Section 1.3.1. Informally, the breakdown function for a statistic T on a distribution P measures the minimal attack budget required for an adversarial distribution to change the outcome of the statistic T by an amount at least $\delta > 0$. Here we recall the classical definition of the breakdown function provided in Definition 1.3.3.

Definition 1.3.3. BREAKDOWN POINT. *Let $\mathcal{Y}$ be a measurable space, $P \in \mathcal{M}_+^1(\mathcal{Y})$ a probability distribution, $T : \mathcal{M}_+^1(\mathcal{Y}) \to \mathcal{Y}$ a statistic, $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and $m : \mathcal{M}_+^1(\mathcal{Y}) \times \mathcal{M}_+^1(\mathcal{Y}) \to \mathbb{R}$ two metrics. The breakdown point for the statistic T on distribution P with metrics m and d is defined by:*

$$\varepsilon^\star(T, P, m, d) = \inf \left\{ \varepsilon > 0 \;\middle|\; \sup_{Q \mid m(P,Q) \leq \varepsilon} d(T(P), T(Q)) = \infty \right\} \tag{1.3.6}$$

In the context of rankings, since the symmetric group is a finite and discrete space, the distance between any rankings is finite. To address this shortfall in the definition of the breakdown point, we define formally what we call the *breakdown function*, as we informally did in Section 3.5.1.

Definition 4.2.1. BREAKDOWN FUNCTION. *Let $\mathcal{Y}$ be a measurable space, $p \in \mathcal{M}_+^1(\mathcal{Y})$ a probability distribution, $T : \mathcal{M}_+^1(\mathcal{Y}) \to \mathcal{Y}$ a statistic, $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and $m : \mathcal{M}_+^1(\mathcal{Y}) \times \mathcal{M}_+^1(\mathcal{Y}) \to \mathbb{R}$ two metrics. For any level $\delta \geq 0$, the breakdown function of the statistic T on distribution P with metrics m and d is defined by:*

$$\varepsilon(\delta, T, P, m, d) = \inf \left\{ \varepsilon > 0 \,\middle|\, \sup_{q:\mathrm{TV}(p,q)\leq\varepsilon} d(T(p), T(q)) \geq \delta \right\}. \tag{4.2.1}$$

When the context is clear, the breakdown function will be simply denoted by $\varepsilon(\delta)$.

In the extreme case, when T is the identity and $\delta = 0^+$, $\varepsilon^\star$ quantifies the budget of attack under which *identifiability* of the distribution is possible (which requires the additional knowledge that P belongs to some family).

Application to Ranking Data. In Agarwal et al. (2020) such a study on identifiability is provided for the Bradley-Terry-Luce Bradley and Terry (1952); Luce (1959) model under a budget constraint on pairwise marginals rather than the Total Variation, and Jin et al. (2018) on the Heterogeneous Thurstone Models Thurstone (1927). However, summary statistics, such as consensus statistics, are generally harder to break than the full distribution itself, so the breakdown function provides a finer quantification of robustness than the identifiability of the distribution. Since the distances on $\mathfrak{S}_n$ are bounded, in general, the full breakdown function needs to be considered and one cannot focus only on a particular level such as $\delta = 0^+$ or $\delta = +\infty$. From here and throughout, the distance d and the attack amplitude δ are normalized to lie between 0 and 1.

The robustness of the median statistic when an adversary is allowed to attack with any strategy a pairwise model has also been studied in Datar et al. (2022). They characterize the robustness of two statistics in terms of the L2 distance on distributions. We propose in Definition 4.2.1 a more general and natural measure for robustness as a function of the distance between the true and a corrupted statistic.

Bucket Rankings as a robustness candidate. In rankings, adversarial attacks often target pairs of items that are 'close' in some sense, like in Agarwal et al. (2020): consecutive ranks, a pairwise marginal probability close to $\frac{1}{2}$, ... Thus, a simple and efficient way to robustify a ranking median is to accept *ties*, rather than being restricted to a strict order.

4.2.3 More Details about Contributions

There is a wide number of median statistic studies motivated by the lack of analytical expression and the computational and statistical challenges that arise in the estimation process. However, robustness results for ranking statistics are rare and not rigorous enough for comparing different estimators.

Contribution 1. *Using Definition 4.2.1 with the Kendall tau distance provides a straightforward measure of robustness for ranking medians. In Section 4.3.1 we provide a lower-bound on the breakdown function for a ranking median (Theorem 4.3.3) and a tight upper-bound for the Kemeny consensus (Theorem 4.3.3).*

Moreover, slight perturbations in the pairwise relations of items that are similar to each other can imply breaking a median estimator, showing a lack of robustness. It is natural to propose more robust estimators by allowing pairs of items to be "equally ranked", i.e., by considering bucket ranking statistics. However, generalizations of the breakdown function for bucket rankings require the use of Kendall tau for buckets, which is computationally impractical.

Contribution 2. *In Section 4.3.2 we propose an extension of the breakdown function for bucket rankings which is built upon a Hausdorff generalization of the Kendall tau distance. We also develop an optimization algorithm to approximate this breakdown function that overcomes the computational issue of having a piece-wise constant objective function.*

We illustrate and show empirically that bucket rankings are more robust median estimators than rankings. However, finding the optimal bucket order statistic requires exhaustively searching the space of bucket rankings Π_n, which is even larger than the space of permutations, of factorial cardinality, and therefore, it is totally infeasible.

Contribution 3. *In Section 4.5 we propose a general method for robustifying medians: given a ranking median, our algorithm successively merges "similar" items together into the same bucket. We evaluate this statistic in Section 4.6, showing an improvement of robustness w.r.t. Kemeny's median without sacrificing its precision.*

4.3 Robustness for Rankings

This Section first details how to apply the notion of *breakdown function* $\varepsilon^\star$. This allows providing insights into the robustness of classical location statistics such as the Kemeny consensus. These results advocate for the introduction of a more robust type of statistics based on bucket orders that are also developed in this Section.

4.3.1 Breakdown Function for Kemeny's Consensus

For a general distribution $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$, we explore the robustness of ranking medians $\sigma_l^\star(P)$ as defined in Definition 2.2.1 for different metrics l over $\mathfrak{S}_n$. The said robustness is explored using the breakdown function with the Kendall Tau distance, namely $\varepsilon^\star(., d_\tau, P, T)$. In particular, it is possible to tightly sandwich the breakdown function for the Kemeny's consensus.

Theorem 4.3.1. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, $\sigma_P^\star = \sigma_{d_\tau}^\star(P)$ be its Kemeny's consensus and $\delta \geq 0$.*

If $\varepsilon^+(\delta) \leq 2P(\Sigma = \sigma_P^\star)$ then $\varepsilon^\star(\delta, d_\tau, P, \sigma^\star) \leq \varepsilon^+(\delta)$ with

$$\varepsilon^+(\delta) = \min_{\substack{\sigma \in \mathfrak{S}_n \\ d_\tau(\sigma, \sigma_P^\star) \geq \delta}} \max_{\substack{\nu \in \mathfrak{S}_n \\ d_\tau(\nu, \sigma_P^\star) < \delta}} \frac{\mathbb{E}_{\Sigma \sim p}\left[d_\tau(\Sigma, \sigma) - d_\tau(\Sigma, \nu)\right]}{d_\tau(\sigma_P^\star, \sigma) - d_\tau(\sigma_P^\star, \nu)}\,. \tag{4.3.1}$$

Proof Sketch. The detailed proof is provided after this high-level sketch. The proof relies on showing that, for $\varepsilon > 0$, the *attack* distribution $\bar{Q}_\varepsilon = P - \frac{\varepsilon}{2}\mathbb{1}_{[\cdot=\sigma_P^\star]} + \frac{\varepsilon}{2}\mathbb{1}_{[\cdot=\sigma_P^{\star,R}]}$, where $\sigma_P^{\star,R}$ is the reverse of $\sigma_P^\star$, is in the feasible set of the optimization problem provided by Definition 4.2.1 $\sup_{q:\mathrm{TV}(P,Q)\leq\varepsilon} d_\tau(\sigma_P^*, \sigma_{Q_\varepsilon}^*)$.

Using $\bar{Q}_\varepsilon$ provides a way to link ε and δ. The condition $\varepsilon^+(\delta) \leq 2P(\Sigma = \sigma_P^\star)$ ensures $\bar{Q}_\varepsilon$ is well-defined. □

The detailed proof is provided here, with the following remark that holds for the rest of the proofs of the Chapter. For the sake of clarity of the proofs, we switch to matrix notation as defined in the following proof.

Proof We fix an arbitrary indexation $\{\sigma^{(1)}, \ldots, \sigma^{(n!)}\}$ of $\mathfrak{S}_n$. Using this indexation, given a metric l on $\mathfrak{S}_n$, we can define the (symmetric) metric matrix $L = (l(\sigma^{(i)}, \sigma^{(j)}))_{i,j\in[n!]}$. Identifying a ranking σ with its corresponding basis vector $\mathbf{e}_i$ s.t. $\sigma = \sigma^{(i)}$, we write for any rankings $\sigma, \sigma', \nu \in \mathfrak{S}_n$,

$$\nu^\top L\sigma := l(\nu, \sigma) \qquad \text{or} \qquad \nu^\top L(\sigma - \sigma') := l(\nu, \sigma) - l(\nu, \sigma') \tag{4.3.2}$$

Further, a distribution $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ on permutation can now be seen as a $n!$-dimensional vector in $\mathbb{R}^{n!}$, which we write, for clarity reasons, $p \in \mathbb{R}^{n!}$. This allows to write, for $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$, $\sigma \in \mathfrak{S}_n$,

$$p^\top L\sigma := \mathbb{E}_{\Sigma\sim P}[l(\Sigma, \sigma)] \tag{4.3.3}$$

We re-state the theorem with the matrix notation defined above.

Theorem 4.3.2. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, $\sigma_P^\star = \sigma_{d_\tau}^\star(P)$ be its Kemeny's consensus and $S_\delta = \{\sigma \in \mathfrak{S}_n | d_\tau(\sigma, \sigma_P^\star) \geq \delta\}$.*

If $\varepsilon^+(\delta) \leq 2P(\Sigma = \sigma_P^\star)$, then $\varepsilon^\star(\delta, d_\tau, P, \sigma_P^\star) \leq \varepsilon^+(\delta)$.

$$\varepsilon^+(\delta) = \min_{\sigma\in S_\delta} \max_{\nu\in N_\delta} \frac{p^\top D_\tau(\sigma - \nu)}{{\sigma_P^\star}^\top D_\tau(\sigma - \nu)}\,, \tag{4.3.4}$$

where D_τ is the metric matrix L when the distance used $l = d_\tau$ is Kendall Tau.

Then, the proof is provided by the following.

$$\varepsilon^\star(\delta, d_\tau, P, \sigma_P^\star) = \inf\left\{\varepsilon > 0 \middle| \sup_{Q:\mathrm{TV}(P,Q)\leq\varepsilon} d_\tau(\sigma_P^\star, \sigma_Q^\star) \geq \delta\right\} \tag{4.3.5}$$

$$= \inf\left\{\varepsilon > 0 \middle| \exists Q, s.t. \mathrm{TV}(P, Q) \leq \varepsilon \text{ and } d_\tau(\sigma_P^\star, \sigma_Q^\star) \geq \delta\right\} \tag{4.3.6}$$

$$= \inf \underbrace{\left\{ \varepsilon > 0 \middle| \exists Q, s.t. \mathrm{TV}(P,Q) \leq \varepsilon \text{ and } \operatorname*{argmin}_{\sigma \in \mathfrak{S}_n} q^\top D_\tau \sigma \subseteq S_\delta \right\}}_{=:E}, \tag{4.3.7}$$

with $S_\delta = \{\sigma \in \mathfrak{S}_n | d_\tau(\sigma, \sigma_p^\star) \geq \delta\}$

Further, we define $N_\delta = \mathfrak{S}_n \setminus S_\delta$, $\sigma_P^{\star,R}$ the reverse of $\sigma_P^\star$, i.e., $\sigma_P^{\star,R}(i) = \sigma_P^\star(n-i-1)$ and the *attack* distribution $\bar{Q}_\varepsilon = P - \frac{\varepsilon}{2}\mathbb{1}_{[\cdot=\sigma_P^\star]} + \frac{\varepsilon}{2}\mathbb{1}_{[\cdot=\sigma_P^{\star,R}]}$ that removes the probability mass from the median to put it on the farthest point.

We also define the aforementioned two sets: $E = \left\{\varepsilon \middle| \operatorname{argmin}_{\sigma\in\mathfrak{S}_n} \bar{q}_\varepsilon^\top D_\tau \sigma \subseteq S_\delta\right\}$ and $\tilde{E} = \left\{0 < \varepsilon \leq 2P(\Sigma = \sigma_P^\star) \middle| \operatorname{argmin}_{\sigma\in\mathfrak{S}_n} \bar{q}_\varepsilon^\top D_\tau \sigma \subseteq S_\delta\right\} \subseteq E \cap (0, 2P(\Sigma = \sigma_P^\star)]$.

Let $\varepsilon > 0$ be such that $\varepsilon \leq 2P(\Sigma = \sigma_P^\star)$. Then

$$\begin{aligned}
\varepsilon \in \tilde{E} &\Leftrightarrow \exists \sigma \in S_\delta, \forall \nu \in N_\delta, \bar{q}_\varepsilon^\top D_\tau \sigma \leq \bar{q}_\varepsilon^\top D_\tau \nu && (4.3.8)\\
&\Leftrightarrow \exists \sigma \in S_\delta, \forall \nu \in N_\delta, p^\top D_\tau(\sigma - \nu) + \\
&\qquad \frac{\varepsilon}{2}\left(\sigma^\top D_\tau \sigma_P^{\star,R} - \sigma^\top D_\tau \sigma_P^\star + \nu^\top D_\tau \sigma_P^\star - \nu^\top D_\tau \sigma_P^{\star,R}\right) \leq 0 && (4.3.9)\\
&\Leftrightarrow \exists \sigma \in S_\delta, \forall \nu \in N_\delta, p^\top D_\tau(\sigma - \nu) \leq \frac{\varepsilon}{2}\left((\sigma_P^\star - \sigma_P^{\star,R})^\top D_\tau(\sigma - \nu)\right) && (4.3.10)\\
&\Leftrightarrow \exists \sigma \in S_\delta, \forall \nu \in N_\delta, p^\top D_\tau(\sigma - \nu) \leq \varepsilon\left(\sigma_P^{\star\top} D_\tau(\sigma - \nu)\right) \\
&\qquad \text{as } \sigma_P^{\star,R^\top} D_\tau \cdot = \|D_\tau\|_\infty - \sigma_P^{\star\top} D_\tau \cdot && (4.3.11)\\
&\Leftrightarrow \exists \sigma \in S_\delta, \forall \nu \in N_\delta, \frac{p^\top D_\tau(\sigma - \nu)}{\sigma_P^{\star\top} D_\tau(\sigma - \nu)} \leq \varepsilon && (4.3.12)\\
&\Leftrightarrow \min_{\sigma\in S_\delta} \max_{\nu \in N_\delta} \frac{p^\top D_\tau(\sigma - \nu)}{\sigma_P^{\star\top} D_\tau(\sigma - \nu)} \leq \varepsilon && (4.3.13)
\end{aligned}$$

Now, denoting $\varepsilon^+(\delta) = \min_{\sigma\in S_\delta} \max_{\nu\in N_\delta} \frac{p^\top D_\tau(\sigma-\nu)}{\sigma_P^{\star\top} D_\tau(\sigma-\nu)}$, by definition $\varepsilon^+(\delta)$ satisfies Equation (4.3.13), which means $\varepsilon^+(\delta) \in \tilde{E}$ iff $\varepsilon^+(\delta) \leq 2P(\Sigma = \sigma_P^\star)$. Thus, if $\varepsilon^+(\delta) \leq 2P(\Sigma = \sigma_P^\star)$, then

$$\varepsilon^+(\delta) = \inf \tilde{E} \geq \inf E = \varepsilon^\star(\delta, d_\tau, P, \sigma_P^\star). \tag{4.3.14}$$

□

It is also possible to provide a lower bound on the breakdown function for any generic ranking consensus, which corresponds to the ranking having the smallest average distance with respect to the studied distribution when using any distance l.

Theorem 4.3.3. *Let $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ be a distribution, d and l be two metrics on $\mathfrak{S}_n$, $\sigma_P^\star = \sigma_l^\star(P)$ be the consensus using metric l, and $\delta \geq 0$, we have $\varepsilon^\star(\delta, d, P, \sigma_P^\star) \geq \varepsilon^-(\delta)$ with*

$$\varepsilon^-(\delta) = \min_{\substack{\sigma\in\mathfrak{S}_n \\ d(\sigma,\sigma_P^\star)\geq\delta}} \max_{\substack{\nu\in\mathfrak{S}_n \\ \nu\neq\sigma}} \frac{\mathbb{E}_{\Sigma\sim P}\left[l(\Sigma,\sigma) - l(\Sigma,\nu)\right]}{\max_{\sigma'\in\mathfrak{S}_n} l(\sigma',\sigma) - l(\sigma',\nu)} \tag{4.3.15}$$

Proof We re-state the theorem with the matrix notation defined above.

Theorem 4.3.4. *For $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$, d and l two metrics on $\mathfrak{S}_n$ and $\sigma_P^\star = \sigma_l^\star(P)$, we have*

$$\varepsilon^\star_{\delta,d,P,\sigma_P^\star} \geq \min_{\sigma\in S_\delta} \max_{\nu\in\mathfrak{S}_n:\nu\neq\sigma} \frac{p^\top L(\sigma-\nu)}{\|L(\sigma-\nu)\|_\infty}\,, \tag{4.3.16}$$

where $S_\delta = \{\sigma \in \mathfrak{S}_n | d(\sigma, \sigma_p^\star) \geq \delta\}$.

Let $N_\delta = \mathfrak{S}_n \setminus S_\delta$, $E = \left\{\varepsilon \middle| \operatorname{argmin}_{\sigma\in\mathfrak{S}_n} \bar{q}_\varepsilon^\top L\sigma \subseteq S_\delta\right\}$, and $\tilde{E} = \left\{0 < \varepsilon \leq 2P(\Sigma = \sigma_P^\star) \middle| \operatorname{argmin}_{\sigma\in\mathfrak{S}_n} \bar{q}_\varepsilon^\top L\sigma \subseteq S_\delta\right\} \subseteq E \cap (0, 2P(\Sigma = \sigma_P^\star)]$.

$$\varepsilon^\star_{\delta,d,P,\sigma_P^\star} = \inf\left\{\varepsilon > 0 \middle| \sup_{Q:\mathrm{TV}(P,Q)\leq\varepsilon} d(\sigma_P^\star, \sigma_Q^\star) \geq \delta\right\} \tag{4.3.17}$$

$$= \inf\left\{\varepsilon > 0 \middle| \exists Q, s.t. \mathrm{TV}(P,Q) \leq \varepsilon \text{ and } d(\sigma_P^\star, \sigma_Q^\star) \geq \delta\right\} \tag{4.3.18}$$

$$= \underbrace{\inf\left\{\varepsilon > 0 \middle| \exists Q, s.t. \mathrm{TV}(P,Q) \leq \varepsilon \text{ and } \operatorname*{argmin}_{\sigma\in\mathfrak{S}_n} q^\top L\sigma \subseteq S_\delta\right\}}_{=:E} \tag{4.3.19}$$

$$\text{Now, } \varepsilon \in E \Leftrightarrow \exists Q, s.t. \mathrm{TV}(P,Q) \leq \varepsilon \text{ and } \operatorname*{argmin}_{\sigma\in\mathfrak{S}_n} q^\top L\sigma \subseteq S_\delta \tag{4.3.20}$$

$$\Leftrightarrow \exists Q \in \Delta^{\mathfrak{S}_n}, \mathrm{TV}(P,Q) \leq \varepsilon \text{ and } \exists \sigma \in S_\delta, \forall \nu \in \mathfrak{S}_n, q^\top L\sigma \leq q^\top L\nu \tag{4.3.21}$$

$$\Leftrightarrow \exists Q \in \Delta^{\mathfrak{S}_n}, \mathrm{TV}(P,Q) \leq \varepsilon \text{ and } \exists \sigma \in S_\delta, \forall \nu \in \mathfrak{S}_n \quad p^\top L(\sigma-\nu) \leq (q_- - q_+)^\top L(\sigma-\nu) \tag{4.3.22}$$

$$\text{where } q_+ = (q-p)_+ \text{ and } q_- = (p-q)_+$$

$$\Rightarrow \exists Q \in \Delta^{\mathfrak{S}_n}, \mathrm{TV}(p,Q) \leq \varepsilon \text{ and } \exists \sigma \in S_\delta, \forall \nu \in \mathfrak{S}_n, \quad p^\top L(\sigma-\nu) \leq \|q_+ - q_-\|_1 \|L(\sigma-\nu)\|_\infty \tag{4.3.23}$$

$$\Rightarrow \exists \sigma \in S_\delta, \forall \nu \in \mathfrak{S}_n, p^\top L(\sigma-\nu) \leq \varepsilon \|L(\sigma-\nu)\|_\infty \quad \text{as } \|q_+ - q_-\|_1 \leq \varepsilon \tag{4.3.24}$$

$$\Rightarrow \exists \sigma \in S_\delta, \forall \nu \in \mathfrak{S}_n, s.t. \sigma \neq \nu, \frac{p^\top L(\sigma-\nu)}{\|L(\sigma-\nu)\|_\infty} \leq \varepsilon \tag{4.3.25}$$

$$\Rightarrow \min_{\sigma\in S_\delta} \max_{\nu\in\mathfrak{S}_n:\nu\neq\sigma} \frac{p^\top L(\sigma-\nu)}{\|L(\sigma-\nu)\|_\infty} \leq \varepsilon. \tag{4.3.26}$$

$$\text{Finally,} \qquad \varepsilon^\star_{\delta,d,P,\sigma_P^\star} = \inf E \geq \min_{\sigma\in S_\delta} \max_{\nu\in\mathfrak{S}_n:\nu\neq\sigma} \frac{p^\top L(\sigma-\nu)}{\|L(\sigma-\nu)\|_\infty}\,. \tag{4.3.27}$$

□

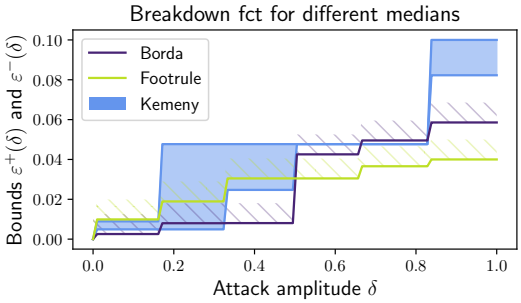


Figure 4.1: An illustration of $\varepsilon^+(\delta)$ and $\varepsilon^-(\delta)$ (from Theorem 4.3.1 and Theorem 4.3.3) for a distribution on permutations of 4 items. For Borda count and the consensus associated with Spearman's footrule, only the lower bound is displayed.

Figure 4.1 shows that no choice of metric l makes the consensus uniformly more robust than an other. Then, unfortunately, it also illustrates the fragility of consensus statistics against the corruption of the distribution. In this example, impacting the distribution P by less than 5% allows changing the Kemeny's consensus by flipping more than half item pairs ($\delta \geq 0.5$).

Sensitivity to similar items. To further illustrate the fragility of Kemeny's consensus, Figure 4.2 shows its breakdown function on specific distributions. As could be expected, if all items are almost indifferent (uniform distribution - purple curve), then a ranking consensus is very fragile: a small nudge on P is enough to change the Kemeny's consensus from one ranking to its reverse. On the contrary, when P is a point mass at a given ranking (blue curve), it requires a large attack on P to impact the consensus.

The green curve shows a weakness in the consensus: despite P being concentrated on two neighboring rankings (identical up to a pair of adjacent items), the robustness is very low for $\delta \leq 0.2$. This highlights a mechanism underlying adversarial attacks in real-world recommender systems (ex: fake reviews...): at a small cost, it is possible to be systematically ranked on top of close alternatives. This calls for using the natural alternative to (strict) rankings, which incorporates indifference between items: *bucket rankings*.

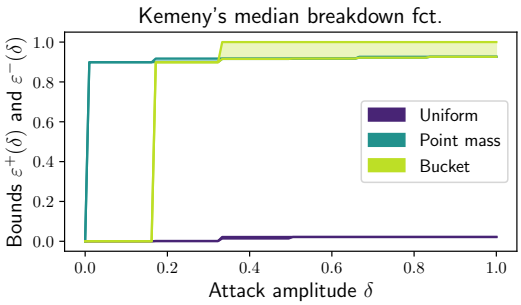


Figure 4.2: Breakdown function for Kemeny's median for different distributions P. "Uniform" denotes an almost uniform distribution; "Point mass" an almost point mass distribution, and "Bucket" an almost point mass distribution on two neighboring rankings.

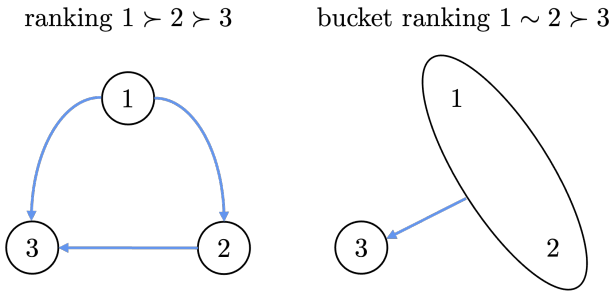


Figure 4.3: Illustration of the difference between a ranking $1 \succ 2 \succ 3$ and a bucket ranking $1 \sim 2 \succ 3$.

4.3.2 Bucket Ranking

Intuitively, bucket rankings are rankings with ties allowed. Formally, they can equivalently be defined as a total preorder – *i.e.* a homogeneous binary relation that satisfies transitivity and reflexivity (preorder) in which any two elements are comparable (total) – or as a strict weak ordering – *i.e.* a strict total order over equivalence classes of items (buckets), as illustrated in Figure 4.3.

Definition 4.3.5. BUCKET RANKING. *A bucket order π is a strict weak order defined by an ordered partition of $[n]$,* i.e. *a sequence $(\pi^{(1)}, \dots, \pi^{(k)})$ of $k \geq 1$ pairwise disjoint non-empty subsets (buckets) of $[n]$ such that:*

$$\textit{(i)}\ i \prec_\pi j \quad \Leftrightarrow \quad \exists l < l' \in [k], (i,j) \in \pi^{(l)} \times \pi^{(l')},$$

$$\textit{(ii)}\ i \sim_\pi j \quad \Leftrightarrow \quad \exists l \in [k], (i,j) \in \pi^{(l)} \times \pi^{(l)},$$

We denote Π_n the set of bucket rankings, which is of size $\sum_{k=1}^{n} k!S(n,k)$[1] (vs $n!$ for $\mathfrak{S}_n$).

The indifference between items that bucket rankings can incorporate is an interesting feature to gain robustness, because the statistic can output alternatives between several strict orders, making it harder to attack.

As sets of permutations. A bucket ranking $\pi \in \Pi_n$ can be equivalently mapped to a subset of permutations, generated through the different ways to break ties. We say that a permutation $\sigma \in \mathfrak{S}_n$ is *compatible* with a bucket ranking $\pi \in \Pi_n$ – denoted $\sigma \in \pi$ – if for any $i, j \in [n]$, $\sigma(i) < \sigma(j) \quad \Leftrightarrow \quad i \prec_\pi j$ or $i \sim_\pi j$. For two bucket orders π_1, π_2, we say that π_1 is *stricter* that π_2, denoted $\pi_1 \subseteq \pi_2$, iff for any $\sigma \in \mathfrak{S}_n$, $\quad \sigma \in \pi_1 \Rightarrow \sigma \in \pi_2$.

As a distribution. Being a set of permutations, a bucket order $\pi \in \Pi_n$ can also be seen as a uniform distribution with restricted support. This point of view is particularly intuitive from a robustness perspective: a randomized output is generally harder to attack for an adversary.

Distances between bucket rankings. A key to applying the breakdown function from Definition 4.2.1 to bucket orders statistics is to have a metric on Π_n that extends those defined on $\mathfrak{S}_n$. To this end, we use the previous remark that weak orders are sets of

[1] $S(n,k)$ are Stirling numbers of the second kind.

rankings as well as a classical Hausdorff extension of metrics to sets. More precisely, we define:

Definition 4.3.6. NON-SYMMETRIC HAUSDORFF. *Let l be a metric on $\mathfrak{S}_n$. The non-symmetric Hausdorff pseudoquasi-metric between two bucket rankings $\pi_1, \pi_2 \in \Pi_n$ is*

$$H_l^{\text{NS}}(\pi_1, \pi_2) = \max_{\sigma_2 \in \pi_2} \min_{\sigma_1 \in \pi_1} l(\sigma_1, \sigma_2)\,. \tag{4.3.28}$$

Even though it is not a metric, H_l^{NS} is well-suited to ranking with ties. Intuitively, its lack of symmetry allows differentiating adversarial attacks whose effect is on the strict part of the bucket order (e.g. swapping two items that are strictly ordered) from those whose effect is 'only' to disambiguate a tie. More precisely, if $\pi_2 \subseteq \pi_1$, then $H_l^{\text{NS}}(\pi_1, \pi_2) = 0$. Depending on the application, one may want to focus on the first type of attacks, in which case H_l^{NS} is a suitable choice to define the breakdown function as $\varepsilon^\star(., H_l^{\text{NS}}, P, T)$.

Otherwise, it is possible (and usual) to symmetrize the Hausdorff metric.

Definition 4.3.7. 1/2-SYMMETRIC HAUSDORFF. *Let l be a metric on $\mathfrak{S}_n$. The 1/2-symmetric Hausdorff metric between two bucket rankings $\pi_1, \pi_2 \in \Pi_n$ is defined by*

$$H_l^{(1/2)}(\pi_1, \pi_2) = \frac{1}{2}\Big(H_l^{\text{NS}}(\pi_1, \pi_2) + H_l^{\text{NS}}(\pi_2, \pi_1)\Big)\,. \tag{4.3.29}$$

Usual symmetrization of the Hausdorff metric uses a maximum rather than an average, see for example Fagin et al. (2006). However, under the Kendall Tau distance, the average version is computationally simpler.

Proposition 4.3.8. *For any $\pi_1, \pi_2 \in \Pi_n$, the computation cost of $H_{d_\tau}^{\text{NS}}(\pi_1, \pi_2)$ and $H_{d_\tau}^{(1/2)}(\pi_1, \pi_2)$ is $\mathcal{O}(n^2)$.*

The average Hausdorff distance can be expressed with various expressions, necessitating the following notations (see Fagin et al. (2006)):

1. $\forall\, i \in [\![1, n]\!] \quad \bar{\pi}(i) = \sum_{\sigma \in \pi} \sigma(i)$ is the rank of item i according to weak order π.
2. $S(\pi_1, \pi_2) = \{(i < j) \mid \bar{\pi}_1(i) \neq \bar{\pi}_1(j), \bar{\pi}_2(i) \neq \bar{\pi}_2(j), [\bar{\pi}_1(i) - \bar{\pi}_1(j)][\bar{\pi}_2(i) - \bar{\pi}_2(j)] < 0\}$ is the set of item pairs $(i < j)$ that are in different buckets in both π_1 and π_2, and that are in different orders in π_1 and π_2.
3. $S(\pi_1 \setminus \pi_2) = \{(i < j) \mid \bar{\pi}_1(i) = \bar{\pi}_1(j) \text{ and } \bar{\pi}_2(i) \neq \bar{\pi}_2(j)\}$ is the set of item pairs $(i < j)$ such that both items are in the same bucket in π_1 but in different ones in π_2.
4. $\text{prof}(\pi) = (\text{prof}(\pi)_{i,j})_{i<j}$, where $\forall\ i < j, \text{prof}(\pi)_{i,j} = 1/2$ if $\bar{\pi}(i) < \bar{\pi}(j)$, $= 0$ if $\bar{\pi}(i) = \bar{\pi}(j)$ and $= -1/2$ if $\bar{\pi}(i) > \bar{\pi}(j)$. $\text{prof}(\pi)$ is called the profile vector of π.

We have the following equivalent expressions for the average Hausdorff distance:

Proposition 4.3.9. AVERAGE HAUSDORFF DISTANCE.

$$\begin{aligned} H_{d_\tau}^{(1/2)}(\pi_1, \pi_2) &:= \#S(\pi_1, \pi_2) + \frac{1}{2}\left(\#S(\pi_1 \setminus \pi_2) + \#S(\pi_2 \setminus \pi_1)\right) \\ &= \sum_{i<j} \mathbb{1}\left([\bar{\pi}_1(i) - \bar{\pi}_1(j)][\bar{\pi}_2(i) - \bar{\pi}_2(j)] < 0\right) + \end{aligned} \tag{4.3.30}$$

$$\frac{1}{2}\mathbb{1}\left([\bar{\pi}_1(i) = \bar{\pi}_1(j)]\right)\mathbb{1}\left([\bar{\pi}_2(i) \neq \bar{\pi}_2(j)]\right) + \frac{1}{2}\mathbb{1}\left([\bar{\pi}_2(i) = \bar{\pi}_2(j)]\right)\mathbb{1}\left([\bar{\pi}_1(i) \neq \bar{\pi}_1(j)]\right) \tag{4.3.31}$$

$$= \|\text{prof}(\pi_1) - \text{prof}(\pi_2)\|_1 \tag{4.3.32}$$

Proof of Average Hausdorff distance.

Let π_1, π_2 be two weak orders associated with buckets $(B_1^1, ...B_{t_1}^1)$ and $(B_1^2, ...B_{t_2}^2)$ respectively. Such buckets are sets of items i forming a partition of $[n]$ such that $i \in B_k^1$ if and only if $\bar{\pi}_1(i) = \sum_{k'<k} \#B_{k'}^1 + \frac{\#B_k^1+1}{2}$ (see Fagin et al. (2006) for a more formal definition).

Let's define as in Critchlow (2012); Fagin et al. (2006): $\forall\, i \leq t_1, \forall\, j \leq t_2, n_{i,j} = \#(B_i^1 \cap B_j^2)$.

Then, from Chapter IV of Critchlow (2012), we have the following relation: $H_{d_\tau}^{(1/2)}(\pi_1, \pi_2) = \frac{1}{2}\left(\sum_{i<i',j\geq j'} n_{i,j}n_{i',j'} + \sum_{i\leq i',j>j'} n_{i,j}n_{i',j'}\right)$.

By noting that $2\#S(\pi_1, \pi_2) = \sum_{i<i',j>j'} n_{i,j}n_{i',j'}$ and $2\#S(\pi_1 \setminus \pi_2) = \sum_{i=i',j>j'} n_{i,j}n_{i',j'}$, we derive our first equality. The second equality directly comes from re-expressing the first one. The third equality comes from Fagin et al. (2006). □

4.4 Estimation of the Breakdown Function

Definition. Putting all the pieces together, from now on, the statistic $T : \mathcal{M}_+^1(\mathfrak{S}_n) \to \Pi_n$ summarizes a distribution over $\mathfrak{S}_n$ by a bucket ranking in Π_n. Then, we use either $H_{d_\tau}^{(NS)}(\pi_1, \pi_2)$ (see Definition 4.3.6) or $H_{d_\tau}^{(1/2)}(\pi_1, \pi_2)$ on Π_n where d_τ is the Kendall Tau distance (see Definition 2.1.3).

Finally, the breakdown function $\varepsilon^\star(\delta, H_{d_\tau}^{(NS)}, P, T)$ is the result of the following optimization problem

$$\inf\left\{\varepsilon > 0 \,\middle|\, \sup_{Q:\text{TV}(P,Q)\leq\varepsilon} H_{d_\tau}^{(NS)}(T(P), T(Q)) \geq \delta\right\} \tag{4.4.1}$$

The Empirical Breakdown Function. Computing a closed-form expression for the breakdown function for any statistic T and distribution P is challenging in general. However, it can be estimated empirically: the extended expression of the breakdown function in Equation (4.4.1) can be simplified so that it is the solution to the following Lagrangian-relaxed optimization problem.

$$\inf_{q\in\Delta^{\mathfrak{S}_n}} \sup_{\lambda\geq 0} 1/2\|p - q\|_1 + \lambda(\delta - H_{d_\tau}^{(NS)}(T(P), T(Q))) \tag{4.4.2}$$

where, as before, we identified distribution $P \in \mathcal{M}_+^1(\mathfrak{S}_n)$ with $p \in \mathbb{R}^{n!}$, a $n!$-dimensional probability vector, thanks to an arbitrary indexation of $\{\sigma^{(1)}, ..., \sigma^{(n!)}\}$ of $\mathfrak{S}_n$.

Smoothing. As $H_{d_\tau}^{(NS)}(T(P), T(Q)))$ is piece-wise constant as a function of Q (with a combinatorial number of pieces), Equation (4.4.2) cannot directly be solved using standard optimization techniques. To solve this issue, we used a smoothing procedure by convolving this function with a smoothing kernel k_γ with scale γ. Thus, after the relaxation, the optimization Equation (4.4.2) becomes:

$$\inf_{q \in \Delta^{\mathfrak{S}_n}} \sup_{\lambda \geq 0} 1/2 \|p - q\|_1 + \lambda(\delta - \rho_T(p, q)), \tag{4.4.3}$$

with

$$\rho_T(p, q) = H_{d_\tau}^{(NS)}(T(p), T(q)) \star k_\gamma(q) \tag{4.4.4}$$

$$= \int_u H_{d_\tau}^{(NS)}(T(p), T(u)) \times k_\gamma(q - u) \mathrm{d}u, \tag{4.4.5}$$

On a practical note, a simple way to build a convolution kernel k_γ on a simplex like $\mathcal{M}_+^1(\mathfrak{S}_n)$, is to use a convolution kernel κ_γ on the whole Euclidean space – for instance an independent Gaussian density $\kappa_\gamma(x) = \frac{1}{\sqrt{(2\pi\gamma)^{n!}}} \exp\left(-\frac{x^{\mathrm{T}} x}{2\gamma^2}\right)$ – and set k_γ to be the density of the push-forward through a *softmax* function. We denote $\varepsilon_{p,T}^\gamma(\delta)$ the limiting value of $\|p - q\|_1/2$ at the solution of Equation (4.4.3). Note the bias induced by such a definition of k_γ fades away when γ goes to 0 in the same way as the bias induced by the convolution. This smoothing ensures ρ_T is a continuous, differentiable function with respect to q. Moreover, it can easily be estimated using a Monte-Carlo sampling, using the following remark: $\rho_T(p, q) = \mathbb{E}_{u \sim k_{(p,\gamma)}}(H_{d_\tau}^{(NS)}(T(u), T(q))$.

Optimization. When using Monte-Carlo estimation for ρ_T, Equation (4.4.3) is a stochastic saddle-point problem. To solve such problems, gradient/ascent has a rate of convergence of $\mathcal{O}(t^{1/2})$ for its ergodic average (t being the number of steps) as shown in Nemirovski and Rubinstein (2002). Our empirical optimization algorithm for computing the breakdown functions relies on stochastic gradient descent and is able to provide good approximations, as illustrated in Figure 4.5.

We denote $\hat{\varepsilon}_{p,T}^\gamma(\delta) = \|p - \bar{q}_t\|_1$, where $\bar{q}_t$ is the ergodic average of the iterates $(q_s)_{s \leq t}$ obtained during the optimization.

Let's make a couple of remarks on the empirical breakdown function $\hat{\varepsilon}_{p,T}^\gamma$. First, it is a noisy estimate of $\varepsilon_{p,T}^\gamma$ as ρ_T, and its gradients are estimated via Monte-Carlo. Thus, the choice of γ and t should trade-off the variance of $\hat{\varepsilon}_{p,T}^\gamma$ and the bias $|\varepsilon_{p,T}^\gamma - \varepsilon^\star(., d_\tau, P, T)|$. Second, as the term $\|p - q\|_1$ is minimized in Equation (4.4.3), it is expected $\hat{\varepsilon}_{p,T}^\gamma$ overestimates $\varepsilon_{p,T}^\gamma$.

4.5 Robust Consensus Ranking Statistics

As proved by Theorem 4.3.1, the classical consensus statistics as defined by Definition 2.2.1 can be easily broken, which motivates defining more robust statistics, based on bucket rankings. As illustrated by Figure 4.2, the weakness of consensus statistics comes from being 'forced' to rank all items, even those which are (almost) indistinguishable. Bucket rankings seem to be a natural solution to this problem, but *what is a good way to build a bucket order statistic?*

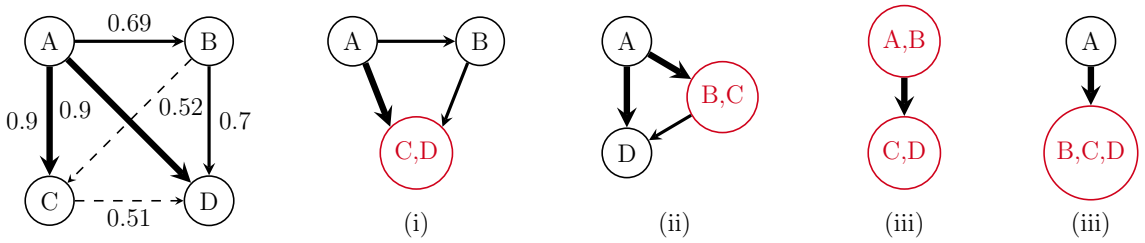


Figure 4.4: Left: Directed Graph that summarizes a pairwise marginal probability matrix. (i-iv) Graph representations of bucket orders that are compatible with merging items whose pairwise preference probability is below 0.52 (i, ii) and below 0.7 (iii,iv).

As $H^{(NS)}_{d_\tau}$ defines a (pseudoquasi-) distance on Π_n, we could adapt the idea of a consensus as in Definition 2.2.1 for bucket rankings. However, contrarily to the Borda count statistic which can be computed in a scalable way as in Caragiannis et al. (2013), Hausdorff-based consensus would require to optimize over Π_n. As its cardinality is larger than $\mathfrak{S}_n$ this problem can be more computationally challenging than Kemeny's aggregation procedure.

A more scalable approach is to start from a consensus such as the Kemeny's consensus or Borda's consensus and to robustify it using a plug-in method based on merging items that are close into buckets. Figure 4.4 illustrates this idea. The left graph describes pairwise marginal probabilities for which the Kemeny's consensus is $A \prec B \prec C \prec D$. Intuitively, merging either C and D (as $\mathbb{P}(C \prec D) = 0.51$) or B and C (as $\mathbb{P}(B \prec C) = 0.52$) leads to bucket rankings (i) and (ii), which will be harder to attack. However, this example also highlights that there is no unique way of merging items. For instance, if the constraint is to only merge items whose pairwise preference probability is in $[0.4, 0.6]$, it is possible to merge B, C or C, D, but not B, C, D as $\mathbb{P}(B \prec D) = 0.7$: *pairwise indistinguishability is not transitive.*

4.5.1 Naive Merge

In order to formalize the latter intuition and to derive a first (naive) plug-in rule, we restate the pairwise preference probability between two items, which provides a relevant notion of closeness between items.

Definition 2.2.4. Pairwise Probabilities. *Let $P \in \mathcal{M}^1_+(\mathfrak{S}_n)$ be a distribution. Its corresponding pairwise probability matrix, denoted by $(p_{i,j})_{1 \leq i,j \leq n}$ is the matrix composed of the pairwise probabilities as defined by:*

$$\forall (i,j) \in [n]^2, \quad p_{i,j} = P(\Sigma(i) < \Sigma(j)). \tag{2.2.4}$$

By convention, in the rest of the Chapter, $\forall i \in [n], p_{i,i} = 0.5$.

Then, given a bucket ranking $\pi \in \Pi_n$, we formalize the notion that two buckets can be merged, with the constraint of not changing the strict order between buckets. To this end, we define $\bar{p}_i(\pi)$, the *strongest deviation from indifference* between any two items within the i^{th} bucket $\pi^{(i)}$.

$$\bar{p}_i(\pi) = \max \left\{ |p_{l,l'} - 0.5| : (l, l') \in \pi^{(i)} \right\} \tag{4.5.1}$$

Then, one needs to quantify the value of $\bar{p}_i(\pi)$ that would result from merging bucket i to bucket j,

$$\bar{p}_{i,j}(\pi) = \max \left\{ |p_{l,l'} - 0.5| : (l, l') \in \bigcup_{\substack{l \in [n] \\ i \leq l \leq j}} \pi^{(l)} \right\} \tag{4.5.2}$$

Finally, given a threshold $\theta \in [0, 0.5]$ on the acceptable deviation from indifference, we define the set of pairs of buckets that can be merged while keeping $\bar{p}$ below θ,

$$\mathcal{G}(\pi, \theta) = \left\{ (i, j) \in [n]^2 : \bar{p}_{i,j}(\pi) \leq \theta \right\} \tag{4.5.3}$$

The first intuition is to merge buckets iteratively, starting with the most indifferent ones, as described in Algorithm 4.1. More precisely, the idea is to iteratively look for pairs of items with a pairwise probability the closest to 1/2, merge them, update the pairwise probabilities and continue until there are no items left to be merged together.

Algorithm 4.1: Naive Merge Plugin

Input : Pairwise matrix $(p_{i,j})$, ranking consensus σ, threshold $\theta \in [0, 0.5]$.

$\pi \leftarrow \sigma$ `// σ as a bucket ranking`

while $\mathcal{G}(\pi, \theta) \neq \emptyset$ **do**

$(i^*, j^*) = \text{argmin}_{(i,j) \in \mathcal{G}(\pi,\theta)} \bar{p}_{i,j}(\pi)$

update π by merging all buckets between i^* and j^*

$$\begin{cases} \pi^{(i)} & \leftarrow \pi^{(i)} \quad \text{for } i < i^* \\ \pi^{(i^*)} & \leftarrow \bigcup_{l \in [n], i^* \leq l \leq j^*} \pi^{(l)} \\ \pi^{(i - j^* + i^*)} & \leftarrow \pi^{(i)} \quad \text{for } i > j^* \end{cases}$$

Output: π

Termination of Algorithm 4.1 is guaranteed by the fact that the number of buckets in π strictly decreases at each iteration. Then, by definition of $\mathcal{G}(\pi, \theta)$, the resulting bucket ranking π is such that any of its bucket i satisfies $\bar{p}_i(\pi) \leq \theta$ – *i.e.* no two items with higher deviation than θ have been merged.

Despite being very natural, this algorithm suffers from an important limitation: when changing the threshold θ, its output only spans a limited subset of valid bucket rankings. In the example provided by Figure 4.4, the naive merge method plugged-in on the Kemeny's consensus can only output (i) and (iii). Whatever the value of θ, it can never output (ii) or (iv). This limitation is induced by its output being a monotonic (with respect to inclusion) function of θ – *i.e.* for $\theta_1 \leq \theta_2$, the resulting bucket rankings satisfy $\pi_{\theta_1} \subseteq \pi_{\theta_2}$.

4.5.2 Downward Merge

Overcoming this limitation only requires a small change to the algorithm which results in our main plug-in method named *Downward Merge*, shown in Algorithm 4.2. Downward Merge algorithm selects the two buckets (i^*, j^*) whose deviation from indifference $\bar{p}_{ij}(\pi)$

is maximal (and not minimal!) among those $\bar{p}_{ij}(\pi) \leq \theta$. Thus, intuitively, instead of taking the most similar buckets, as in the previous statistic, we take the most different pair among those that are 'similar enough'. Then, all the buckets l such that $i^* \leq l \leq j^*$ are merged. This process is repeated while there exist pairs of buckets whose deviation from indifference $\bar{p}_{ij}(\pi) \leq \theta$ and thus termination is guaranteed.

Algorithm 4.2: Downward Merge Plugin

Input : Pairwise matrix $(p_{i,j})$, ranking consensus σ, threshold $\theta \in [0, 0.5]$.

$\pi \leftarrow \sigma$ `// σ as a bucket ranking`

while $\mathcal{G}(\pi, \theta) \neq \emptyset$ **do**

$(i^*, j^*) = \text{argmax}_{(i,j)\in\mathcal{G}(\pi,\theta)} \bar{p}_{i,j}(\pi)$

update π by merging all buckets between i^* and j^*

$$\begin{cases} \pi^{(i)} & \leftarrow \pi^{(i)} \quad \text{for } i < i^* \\ \pi^{(i^*)} & \leftarrow \bigcup_{l\in[n], i^* \leq l \leq j^*} \pi^{(l)} \\ \pi^{(i-j^*+i^*)} & \leftarrow \pi^{(i)} \quad \text{for } i > j^* \end{cases}$$

Output: π

The Downward Merge method is thus able to span a larger set of bucket orders when varying θ. In the example from Figure 4.4, the Downward Merge method plugged-in on the Kemeny's consensus can generate all four bucket rankings (i-iv) for $\theta \in \{0.51, 0.52, 0.69, 0.7)\}$.

The next experimental Section illustrates the robustness improvement brought by this plug-in method over a ranking median.

4.6 Experiments

In this Section, we illustrate the relevance of the statistic outputted by our Downward Merge plug-in on Kemeny's consensus (called our *Downward Merge statistic* for short) by running several illustrative experiments for various settings and comparing with the baseline provided by the usual Kemeny's consensus. The code is available https://github.com/RobustConsensusRanking/RobustConsensusRanking.

4.6.1 Empirical Robustness

Our Downward Merge plug-in aims at providing a robustified statistic. To illustrate its usefulness, we ran experiments computing the approximate breakdown functions $\hat{\varepsilon}^{\gamma}_{p,T}(\delta)$ for the Kemeny's consensus as a baseline and our statistic when varying δ. Figure 4.5 shows the robustness as a function of attack amplitude δ and for a hand-picked distribution P that is almost a point mass on a bucket ranking.

When the threshold is set to a sensible value (here $\theta = 0.05$), the Downward Merge algorithm outputs a bucket order as a statistic: thus, the robustness increases very strongly to reach nearly optimal values even for very small values of δ, which illustrates its efficiency. When $\theta = 0.5$, the statistic is the bucket order regrouping all items. In this case, the statistic cannot be broken, and provide optimal values for the breakdown function. However, such a statistic does not provide any information about the distribution under analysis:

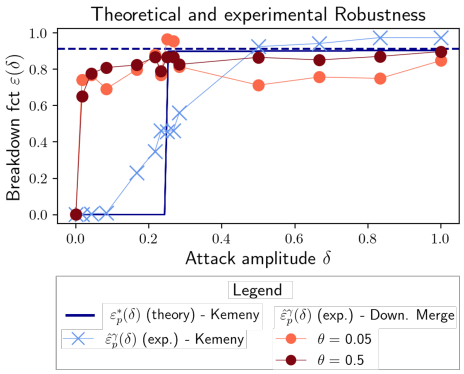


Figure 4.5: Breakdown function $\hat{\varepsilon}^\gamma_{p,T}(\delta)$ as a function of attack amplitude δ for a bucket distribution P (almost a point mass on two neighboring rankings) with $n = 4$. The plain blue line denotes the theoretical value for Kemeny's consensus $\varepsilon^\star(\delta, P)$, blue crosses (resp. red dots) the empirical approximation $\hat{\varepsilon}^\gamma_{p,T}$ for Kemeny's consensus (resp. Down. Merge statistic for different thresholds θ).

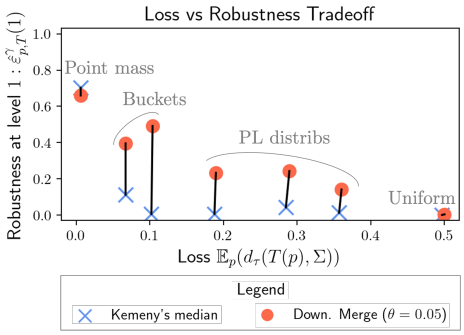


Figure 4.6: Loss/Robustness tradeoffs for different P with $\delta = 1$. Pairs of points linked by a black line denote results for Kemeny's consensus and Down. Merge statistics on the same distribution P with $n = 4$. "Buckets" are hand-picked distributions generated to be almost a point mass on a bucket order, "Uniform" (resp. "Point mass") "is an almost uniform (resp. point mass) hand-picked distribution, and "PL distribs." are random Plackett-Luce distributions.

its *precision*, or its *accuracy of location*, is very poor. Formally, the precision or accuracy of location of a statistic T is defined by its closeness (under the same metric l used in its definition) to the whole ranking distribution: $AL_{l,P}(T) := \|dl\|_\infty - \mathbb{E}_{\Sigma\sim P}(d(T(P), \Sigma))$, which is the opposite of the *loss*, as simply defined by $Loss_{l,P}(T) = \mathbb{E}_{\Sigma\sim P}(d(T(P), \Sigma))$. By definition, under metric $l = d_\tau$, Kemeny's consensus has the highest accuracy of location, *i.e.* the smallest loss. On the other hand, the Downward Merge statistic when $\theta = 0.5$ has a very high loss, which makes it irrelevant in most cases. These observations justify the analysis of the loss/robustness tradeoff of our Downward Merge statistic compared to Kemeny's median.

4.6.2 Tradeoffs between Loss and Robustness

We ran experiments for various distributions P and computed the loss and the breakdown function of Kemeny's consensus and our Downward Merge algorithm to show the loss/robustness tradeoff for each statistic. Figure 4.6 shows the results for different choices of distribution P when the number of items $n = 4$, and for $\delta = 1/6$ (normalized value of

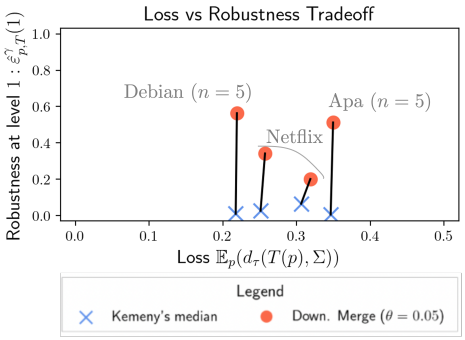


Figure 4.7: Loss/Robustness tradeoffs for different real-world datasets with $\delta = 1$. Pairs of points linked by a black line denote results for Kemeny's median and Down. Merge statistics on the same dataset.

δ that requires at least a switch between two items to break the statistic).

The point mass (resp. the uniform) distribution represents the extreme case for which Kemeny's consensus is very robust (resp. not robust at all) and for which we expect no improvement from using the Downward Merge statistic. This intuition is verified in both cases, and we can see that the Downward Merge statistic yields the same results (in loss and in robustness) as Kemeny's consensus.

The bucket distributions (for which the gap between the probabilities for two rankings in the bucket order is respectively 0.1 and 0.01) represent the settings to which our Downward Merge is best suited. As expected, the improvement in robustness when using our Downward Merge statistic is high, and the increase in loss is negligible.

Finally, the Plackett Luce distributions (for which the parameters were generated randomly) represent a random setting. The results are interestingly very similar to those for the bucket distributions: the gain in robustness is high and the increase in loss is negligible. This random setting illustrates the usefulness of our Downward Merge statistic in general cases and shows that, overall, it yields a much better compromise than Kemeny's consensus.

To corroborate these findings in more practical settings, we also ran experiments using real-world datasets from the *preflib library* that can be accessed here: https://www.preflib.org/. We used two Netflix Prize datasets (resp. with $n = 3$ and $n = 4$ items), a Debian dataset (with $n = 5$ items), and an Apa dataset (with $n = 5$ items). The results are shown in Figure 4.7, and corroborate the synthetic results: our plugin always provides much better robustness, while the increase in the loss stays minimal.

4.7 Conclusion

In this Chapter, we developed a framework to study practical robustness in rankings: not only defined breakdown functions for rankings, extended it to bucket rankings, and created an optimization algorithm to approximate its value in practice. In addition to this experimental setting, we provided theoretical bounds on the breakdown function of classical consensus rankings such as Kemeny's consensus.

Further, we developed our Downward Merge statistic as a plug-in to the classical Kemeny's consensus to provide, as confirmed by our experiments, not only improved robustness but also a better compromise between centrality and robustness. By enforcing undecidability between close items and constructing a bucket ranking as an output, our Downward Merge plugin leverage not only the structure of the symmetric group, but also the randomness as a strategy to improve robustness. Indeed, a bucket ranking can also be seen as a set of rankings: in this case, if one requires a unique ranking as consensus, a simple strategy is to sample uniformly a ranking in the bucket ranking set. This random strategy illustrates the difficulty for an adversarial attack to fool the bucket ranking, because it is harder to attack a random strategy compared to a fully deterministic one.

In addition to the robustness provided by our plugin, we ensured our Downward Merge algorithm can be used in practice as it is scalable to most practical settings. However, the evaluation of the breakdown function remains challenging because of the use of the Total-Variation distance as a metric for the budget constraint, which requires computing the L_1-norm on a vector of size $n!$, where n is the number of items. Thus, our approximation is not scalable to large values of n: the definition and study of further scalable approximations of the breakdown function remains to be done.

Chapter 5

Conclusion about Robustness in Rankings

Do you know the problem with a disguise? However hard you try, it's always a self-portrait.

Irene Adler, Sherlock

In this Part, the topic of robustness against *poisoning attacks*, specifically in the context of the consensus ranking task for ranking data, was tackled. As introduced in Section 1.3, poisoning attacks target models at *training* time. Though deeply studied for real-numbered data (or multivariate data), the robustness against poisoning attacks was not introduced for more complex data space, namely ranking data, which aggregate a lack of vector space structure and combinatorial nature.

To initiate the study of robustness for the consensus ranking task, Chapter 3 adapted the concept of depth functions to rankings. Depth functions give a way to assign a score to data points in order to provide a notion of *centrality* of a data point. This centrality score enables to construct equivalents of *ranks* to ranking data in order to build equivalents of quantiles. Thanks to the theoretical definitions and the statistical bounds provided, depth functions were used to construct a *trimming* algorithm to filter out adversarial or outlier points into a dataset. This trimming algorithm mimics the notion of *trimmed mean* to robustify traditional consensus statistics. This strategy is shown to be very relevant through experimental illustrations, but also through theoretical analysis.

In addition to this first strategy, Chapter 4 provided a clearer way to evaluate empirically the robustness of a statistic solving the consensus ranking task, via an algorithm approximating the *breakdown function*, which is a measure of robustness introduced in the classical robust statistics literature. Moreover, a plugin to improve the robustness of any statistics was proposed. The idea is to introduce *bucket rankings*, which allows a form of undecidability between items that are close to each other according to the dataset. This plugin can be added on top of any statistic and is shown to provide much more robustness (via increased breakdown function values) and almost no precision loss (via almost no decrease of the location precision).

In conclusion, these two works initiated the systematic study of robustness against poisoning attacks for rankings. By focusing on the basic task of consensus ranking, these works allow for a simple extension to the current tasks involving ranking data, which are essential in recommender systems (top-k rankings, etc.). In addition, our work focused on robustifying consensus statistics from a general, theoretical point of view, meaning that these works do not depend on specific attack algorithms. As the problem has almost not been studied before, extensions of the present works are needed to adapt to specific settings and to provide other robustification strategies. However, this thesis is essential for building a framework for more the studies on robustness in rankings.

A limitation of the present work relies in the scalability of the methods. If the robust statistics presented in Chapters 3 and 4, in particuler the Downward Merge plugin, are indeed scalable and can be computed on distributions and datasets on the symmetric group for large number of items, this is not the case for the practical evaluation of the robustness of consensus statistics, which does not fully overcome the challenge of the combinatorial nature of the ranking space. This limit is mitigated by the theoretical bounds provided in Chapter 4, but the robustness of different statistics in complex settings is not achievable via our methodology. Providing practical evaluation of robustness in a scalable way is, in our point of view, the main requirements for future perspective on the subject.

Part II

Understanding and Unifying Recent Advances on Adversarial Robustness

Table of Contents

List of Notations

Neural Networks

f_θ	A neural network
g_θ	The feature map corresponding to neural network f_θ
$\tilde{g}_\theta$	The probability vector outputted by the neural network f_θ
l	the training loss for the neural network

Data

x	An input data, usually an image
y	A label
δ	Usual notation for an adversarial perturbation
x^{adv}	The adversarial example corresponding to input x

Metrics and Norms

$\|\|x\|\|_2$	The L_2-norm of vector x
$\|\|A\|\|_{op}$	The operator norm of real matrix A

Topology and Graphs

$G(x, g_\theta, x)$	Induced graph from feature map g_θ and input x
Φ_{PD}	The PD feature extraction method
K_{PD}	The Sliced-Wasserstein Kernel

Geometric Objects

$\mathcal{S}_{d-1}$	The unit sphere of $\mathbb{R}^d$
$\mathcal{B}_d$	The unit ball of $\mathbb{R}^d$
$\Pi_A x$	The orthogonal projection of vector x on space A

Generic Objects

$[k]$	The set $\{1, ..., k\}$ of integers from 1 to k
$\#E$	Cardinality of set E
$\mathbb{1}(\mathcal{E})$	Indicator function of event $\mathcal{E}$
t_+	The max between t and 0, ie $\max(t, 0)$

Asymptotic Comparisons

$f(a) = \mathcal{O}(g(a))$	$\exists c, b > 0$ such that $\forall a \geq b, f(a) \leq cg(a)$
$f(a) \asymp g(a)$	$f(a) = \mathcal{O}(g(a))$ and $g(a) = \mathcal{O}(f(a))$

Chapter 6

Introduction to Adversarial Examples on Deep Learning Models

A prince being thus obliged to know well how to act as a beast must imitate the fox and the lion, for the lion cannot protect himself from traps, and the fox cannot defend himself from wolves. One must therefore be a fox to recognize traps, and a lion to frighten wolves.

Niccolò Machiavelli.

Contents

In this Chapter, the main concepts and notions about adversarial examples against neural networks for image classification will be introduced. Some findings on adversarial examples will be particularly highlighted.

6.1 Robustness in Deep Learning

Since the seminal work of Szegedy et al. (2013), adversarial robustness has become a sub-field of deep learning research. In particular, the field has gained some structure and good practices to ease collaboration and facilitate the comparison of different works. In this Section, the most important definitions, notions, and typologies will be introduced.

6.1.1 Definition of Adversarial Robustness

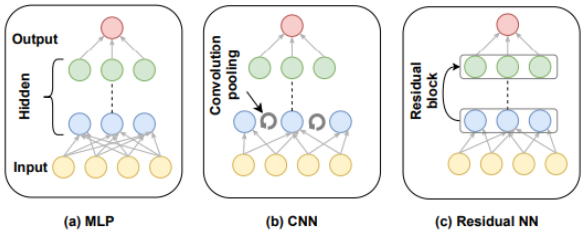


Figure 6.1: Different types of neural networks architectures. Courtesy of Baccour et al. (2022).

As illustrated with the definition of a multilayer perceptron in Definition 1.4.1, a Neural Network is a class of algorithms inspired, historically, by the functioning of the brain. It consists in a computational architecture where layers of artificial neurons are connected by weighted edges, enabling the network to apply linear transformations before using non-linear activation functions, which facilitates complex pattern recognition and information processing.

Historically, neural networks have gained a renewed interest and have become the standard type of machine learning algorithm for computer vision tasks starting from LeCun et al. (1989). In addition to enabling efficient training of neural networks through backpropagation of the gradient, these papers popularized *Convolutional neural networks*, CNNs, meaning neural networks whose some layers are *convolutional*. Since then, the success of neural networks in solving image classification tasks on more and more complex datasets has been unmatched. Concurrently, more sophisticated architecture types, or layers types, were introduced to tackle the complexification of the datasets, as illustrated by Figure 6.1, from Baccour et al. (2022).

Adversarial examples, as defined in Section 1.4.1 target all these types of architectures. To assess the robustness (or the vulnerability) of a neural network, the concept of *adversarial accuracy* is used.

Definition 6.1.1. Adversarial Accuracy against an Attack. *Let $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ be a distribution, f_θ be a neural network on a K-classification problem and $A : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ an adversarial attack. The adversarial accuracy of f_θ on distribution $P_{X,Y}$ against attack A is defined by*

$$Acc(f_\theta, P_{X,Y}, A) = \mathbb{E}_{X,Y \sim P_{X,Y}}(\mathbb{1}[f_\theta(A(X,Y)) = Y]) \tag{6.1.1}$$

The adversarial accuracy simply computes the probability that the neural network predicts the correct class for adversarial examples. It highly depends on the adversarial attack used, which explains the richness of works on creating different attacks. For example, the FGSM attack introduced in Definition 1.4.11 is very efficient on MLPs and small CNNs that do not incorporate any robustness strategy, but it is, eventually, quite simple to robustify a neural network against FGSM via different defense techniques. The goal of an attacker is, of course, to drag the adversarial accuracy towards 0. Alternatively, a neural network is robust when the adversarial accuracy is sufficiently high.

6.1.2 Adversarial Attacks in Practice: Categories of Adversarial Examples

As the development of adversarial attacks is mainly experimental, the large number of adversarial attack models crafted so far can be classified into various typologies. The common classification of adversarial examples relies on the capacity of the attack (white-box and black-box setting, subspace selection, etc.), the general type of method used (gradient-based, query-based, etc.), and the objective of the attack (targeted or untargeted). Additional categories can be discussed, for example, the computational requirements of the attacks (single-step or iterative attacks), or the scenario covered by the attacks (real-world attack versus 'laboratory' attack).

This Section will present the main typologies of adversarial examples to better explain the scope of the contribution, as well as introduce first intuitions about the phenomenon.

Targeted and Untargeted attacks. Adversarial examples aim at fooling a neural network, but *how* the network should be fooled can be different. *Targeted* attacks aim to deceive the neural network by driving it to incorrectly predict a specific class that has been predetermined in advance. They are thus more precise, and so with a lower success rate than *untargeted* attacks: such attacks just aim to deceive the neural network, whatever the prediction. More specifically, the definition of a practical attack provided in Definition 1.4.10 defines in fact an *untargeted* attack, which is recalled below.

Definition 1.4.10. Adversarial Attack. *Let f_θ be a neural network, and $||.||$ a norm on $\mathcal{X}$. Let $\varepsilon \in [0,1]$ be the perturbation budget. An ε-practical adversarial attack is a function $A_\varepsilon : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, defined by:*

$$\begin{aligned} \forall\, (x,y) \in \mathcal{X} \times \mathcal{Y},\ A_\varepsilon(x,y) = x + \delta_\varepsilon(x,y) \quad \textit{such that}\ \ ||\delta_\varepsilon(x,y)|| \leq \varepsilon \\ \textit{with}\ f_\theta(A_\varepsilon(x,y)) \neq f_\theta(x)\ \textit{as often as possible.} \end{aligned} \tag{1.4.11}$$

The adversarial example corresponding to $(x,y) \in \mathcal{X} \times \mathcal{Y}$ is generally denoted by $x^{adv} = A_\varepsilon(x,y)$.

On the other hand, a targeted one is defined as follows:

Definition 6.1.2. Targeted Attack. *Let f_θ be a neural network on a K-classification problem, and $||.||$ a norm on $\mathcal{X}$. Let $\varepsilon \in [0,1]$ be the perturbation budget, and $t \in [\![1, K]\!]$ the target class. An (ε, t) adversarial attack is a function $A_{\varepsilon,t} : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, defined by:*

$$\begin{aligned} \forall\, (x,y) \in \mathcal{X} \times \mathcal{Y},\ A_{\varepsilon,t}(x,y) = x + \delta(x,y) \quad \textit{such that}\ \ ||\delta_{\varepsilon,t}(x,y)|| \leq \varepsilon \\ \textit{with}\ f_\theta(A_{\varepsilon,t}(x,y)) = t\ \textit{as often as possible.} \end{aligned} \tag{6.1.2}$$

The adversarial example corresponding to $(x, y) \in \mathcal{X} \times \mathcal{Y}$ *is generally denoted by* $x^{adv} = A_{\varepsilon,t}(x, y)$.

In the scope of this thesis, the focus is directed towards exploring and studying the phenomenon of adversarial examples as a whole, so untargeted attacks because they correspond to the most generic form of adversarial attacks.

Capacities of the attack. Szegedy et al. (2013) started to study the adversarial phenomenon from a conceptual perspective, thus, at the early stage of the field, adversarial attacks were not necessarily meant to be used in practice. Later, the literature started to explore attacks that could be implemented on deployed available models, for example, through API. This has led the field to consider what an attacker can have access to: the main categories of adversarial attacks in that regard are *white-box* attacks and *black-box* attacks.

Simply put, a white-box attack has full access to the neural network, for example, it has access to the weights of a neural network, to the loss used, to the gradients, etc. The FGSM attack uses the gradient of the loss of the model with respect to the input in its formulation, meaning it is a white-box attack. On the contrary, black-box attacks suppose no knowledge whatsoever about the neural network and are thus meant for 'real-world' scenarios. Examples of black-box attacks are the Boundary attack Brendel et al. (2017) and the SimBA attack Guo et al. (2019), which will be used and detailed later. The capacities of the attack methods have, in fact, a deep influence on the kind of methodology used. Gradient-based methods are natural for white-box attacks, since, in this setting, an adversarial objective (similar to Equation (1.4.8)) can be directly optimized. On the other hand, query-based methods are traditionally used by black-box attacks to gather sufficient information from a neural network to perform the attack.

Interestingly, another type of category can be explored in the context of differences in capacities for adversarial attacks. If an adversarial attack has generally a magnitude or budget constraint on the perturbation size to ensure that the attack is imperceptible, how to allocate this budget is not constrained. This has led some works to define original attacks that operate only on a specific subspace of the features space, with for example Su et al. (2019) which creates an attack that changes only one pixel of the clean images, or Moosavi-Dezfooli et al. (2017) which creates an attack that is the same for all images. Such work can be regrouped under the terminology of *low-dimensional adversarial perturbations* (LDAPs) and will be thoroughly analyzed in Chapter 8.

6.1.3 Adversarial Defense in Practice: the Variety of Attributes to Robustify

Concurrently with the development of new and more efficient attacks, many works have focused on developing defense mechanisms to robustify neural networks. Since it is very difficult to provide a practical and general optimization problem for adversarial examples (which explains why so many different attack methods exist), it is also very difficult to provide such a general framework for defending neural networks that would be solvable in practice. For that reason, numerous different defense strategies also exist, which have been developed alongside the progress of adversarial attacks, similar to a cat-and-mouse

game. The defense strategies developed so far have thus focused on different parts of the data, the neural network model, or the training procedures, and can therefore be divided into several categories.

Data Modifications. A first line of work has focused on modifying the input data. For example, JPEG compression has been used in Dziugaite et al. (2016) with the idea to push back the adversarial examples near the natural data manifold. Similarly, Guo et al. (2018b) uses several data compression techniques (among which JPEG compression) at the same time before feeding the images to the classifier. Alternatively, Samangouei et al. (2018) uses a generative adversarial network (GAN) that reconstructs a similar image from an input: it is used before they are fed to the neural network classifier. All these works focus on eliminating the adversarial noise before it is exposed to the neural network, but even though these defense strategies have been shown to be efficient on some attacks, other attack methods provide perturbations that are not filtered out by these techniques.

Model Optimization. A large number of papers have been devoted to directly modifying the neural network to take into account robustness. The strategies proposed in this category can in fact be quite different. One of the first ideas was proposed in Papernot et al. (2016) and called *Defensive Distillation*: a teacher model is trained on the source distribution, and its probability vector outputs are then fed to a student model to replace the ground-truth label. Though robust to the FGSM attacks and some others, this defense has been bypassed by Carlini and Wagner (2017) and their attack called CW. Other strategies include regularization techniques like in Ma et al. (2020); Ross and Doshi-Velez (2018) or providing stochasticity in the neural network at inference time, like Gao et al. (2017); Wang et al. (2018b); Liu et al. (2018).

Part of these works relies in fact on *gradient masking*, which incorporates all techniques that hide the gradient of the loss to the attacker: this effect has been shown to be ineffective to defend against adversarial attacks in general, see Athalye et al. (2018).

Training modification. The most famous and preferred approach to robustify neural networks is *adversarial training*. It consists in modifying the training procedure to take into account both clean and adversarial inputs. Contrary to data modification strategies, adversarial training thus aims at exposing the network to a broader set of inputs and namely adversarial ones, to help it better understand adversarial examples and thus correctly classify them. Adversarial training was introduced as early as Goodfellow et al. (2014) and many works have followed afterward to improve the process, with for example Madry et al. (2018); Shafahi et al. (2019b); Zhang et al. (2019a). Adversarial training seems to be the most efficient and generic approach to robustification, even though it is not robust to every type of attack.

External Networks and Detection. A different line of work has focused on adding an extra network devoted not to the classification task, but to the detection of adversarial examples, like in Xu et al. (2017); Metzen et al. (2017); Ma et al. (2018); Lee et al. (2018). Rather than concentrating on preserving model accuracy when confronted with adversarial examples, these works have focused on detection mechanisms to identify and reject adversarial examples, irrespective of the classification made by the neural network.

These detection methods operate by studying and uncovering atypical patterns induced by adversarial perturbations, such as deviations in model behavior, anomalies in data distribution, and irregularities in learned features.

6.1.4 Current Limitations and Research Questions

Beyond the ongoing cat-and-mouse dynamics between adversarial attack and defense research, the fundamental intricacy of solving the optimization problem engendered by the adversarial phenomenon, and exposed in Definitions 1.4.6 and 1.4.8, has propelled the field to advance incrementally in both the generation and mitigation of such phenomenon. These advancements, akin to those expounded in Sections 6.1.2 and 6.1.3, have shed light on the limitations they have unveiled. These limitations are described below.

Neural Networks may be inherently vulnerable. Some works have focused on studying the robustness of neural network classifiers under a theoretical perspective, mainly to provide theoretical bounds on the robustness (or alternatively on the vulnerability) inherent to neural networks. More specifically, if Definition 6.1.1 describes the adversarial accuracy of a classifier with respect to a specific adversarial attack, the (general) adversarial accuracy of a model can be defined as follows:

Definition 6.1.3. Adversarial Accuracy. *Let $P_{X,Y} \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$ be a distribution and f_θ be a neural network on a K-classification problem, d a distance on $\mathcal{X}$ and $\mathcal{B}_{\varepsilon,d}(x) = \{x' \in \mathcal{X} \mid d(x, x') \leq \varepsilon\}$ the ε-ball around x with respect to d. The (ε, d)-adversarial accuracy of f_θ on distribution $P_{X,Y}$ is defined by*

$$Acc^{adv}(f_\theta, P_{X,Y}) = \mathbb{E}_{X,Y \sim P_{X,Y}}(\mathbb{1}[\forall\, X' \in \mathcal{B}_{\varepsilon,d}(X),\ f_\theta(X') = Y]) \tag{6.1.3}$$

The study of the adversarial accuracy of a neural network provides information on its intrinsic robustness. Several works have focused on establishing bounds on this quantity, with informal statements such as the following: either *the adversarial accuracy is upper-bounded by some function depending on the neural network and/or the data distribution*, or *the average distance between a data point and its closest adversarial counterpart is lower-bounded by some functions depending on the neural network and/or the data distribution.* Both formulations are, in fact, equivalent.

Prominent works in this field include Fawzi et al. (2018b,a); Mahloujifar et al. (2019); Bubeck et al. (2019); Dohmatob (2019); Ford et al. (2019); Melamed et al. (2023). They all provide analysis and bounds on the aforementioned adversarial accuracy with similar constraints or hypothesis on the data distribution or on the studied models. Some works suppose quite specific or small models (linear or quadratic models), but more importantly they usually rely on a form of *curse of dimensionality* to prove their result (for example, most of the aforementioned work use the Gaussian isoperimetric inequalities). The curse of dimensionality traditionally refers to the fact that the volume of a space increases exponentially fast with its dimension. In the context of adversarial robustness, this means that the volume of the adversarial space is very big, leading to inherent vulnerability.

These works thus tend to show that even when the perturbation size is small, adversarial examples are likely to exist for all types of neural networks.

Recent and practical adversarial attacks rely on heuristics. Recent advances in adversarial attacks have focused on more practical settings, like the black-box setting, to craft adversarial attacks that are at the same time usable in real-world scenarios (when interacting with neural networks through API for example), sufficiently computationally effective (not necessitating too many queries for example) and imperceptible not only to the human eye but also to concurrent defense strategies. Stemming from the seminal work of Moosavi-Dezfooli et al. (2017), different attacks have been developed that modify only a small subspace of the data space, like Guo et al. (2018a); Huang and Zhang (2019); Yan et al. (2019); Tu et al. (2019); Chen et al. (2020a). These attacks are based on effective intuitions and heuristics, for example, using an external neural network to select the relevant subspace to attack, or approximating the gradient of the loss via Monte-Carlo sampling. These attacks have been highly successful (for example, SimBA attack achieves a success rate of 98.6% using as few as 1232 queries on ImageNet). However, there is no theoretical study explaining the success of these methods: as mentioned previously, theoretical work generally relies on the curse of dimensionality to provide theoretical bounds on the adversarial robustness of neural networks, but such an argument is not possible for low dimensional adversarial perturbations.

Chapter 8 will be devoted to providing an in-depth theoretical analysis of these adversarial attacks to overcome the lack of understanding relative to their recent success.

6.2 Exploring the Complexity of Adversarial Behavior

Despite the development of adversarial attacks and defense methods, as well as the proven existence of adversarial examples, as discussed in Sections 6.1.2 to 6.1.4, a comprehensive understanding of the underlying interpretation and explanation of this phenomenon remains incomplete. This Section focuses on reviewing the current research advances on this important question, as well as exposing their limitations. Broadly speaking, understanding *how* and *why* adversarial examples succeed in fooling neural networks is a research question that can be investigated by concentrating either on the adversarial examples *per se*, meaning taking a data-centric approach, or on the vulnerable neural networks, meaning a model-centric approach. Research advances on the former approach will be presented in Section 6.2.2, and the latter in Section 6.2.1.

6.2.1 Hypothesis on the Neural Network

Based on a comprehensive review of publications by Han et al. (2023), approximately 40% of the works focus on exploring the interpretation of the adversarial phenomenon from a model-centric perspective. However, a significant variation exists in the specific aspects investigated within the models themselves, whose main results per main categories (as defined by Han et al. (2023)) are recalled below.

On properties of neural networks (linear hypothesis and architecture). As initiated very early by Goodfellow et al. (2014), the linearity hypothesis has received a lot of attention, but the conclusion remains open. Following Goodfellow et al. (2014), which have notably introduced the FGSM attack based on this hypothesis, some works have provided additional evidence supporting the fact that the local linear behavior of neural

networks may explain their vulnerability, with for example Li et al. (2021a) or Taghanaki et al. (2019). However, other works challenge this conclusion, in particular, the work of Tanay and Griffin (2016) which also introduces another hypothesis based on the behavior of the decision boundary. The linear hypothesis thus remains quite open to new evidence and is not enough to provide a clear explanation of the success of adversarial examples.

Quite recently, many works have proposed to study adversarial examples through the lens of the architecture of neural networks. More precisely, some structural elements like skip connections found in ResNet-like architectures, the width and the depth of the different layers of the networks, and the overall density of the architecture of a neural network were studied in Guo et al. (2020); Huang et al. (2021); Li et al. (2021b); Wu et al. (2020). These works aim at finding ingredients to design more robust neural networks through carefully crafting their architectural components and have paved the way for the use of architectural search for robustness purposes. Interestingly, contrary to popular belief, increased width and depth of neural networks have not been found to improve the robustness in general, and reducing width and depth specifically in the last layers has in fact been shown to be associated with better robustness.

On the training procedure (loss functions, evolutionary stalling hypothesis, and decision boundary). If adversarial training has received a lot of attention to improve the robustness of neural networks (see Goodfellow et al. (2014); Zhang et al. (2019b); Shafahi et al. (2019b); Zhang et al. (2019a); Wang et al. (2020b); Wong et al. (2020); Sitawarin et al. (2021)), other topics related to training procedures have been studied to explain the adversarial phenomenon. Among these topics, the question of the loss function is fundamental: Nar et al. (2019) shows that the *cross-entropy* loss, massively used in convolutional neural networks for image classification, can lead a trained model to output very small margins between the data points and the decision boundary. Recently, several theoretical works have shown that there is no convex surrogate loss that is calibrated for the adversarial optimization problem as formulated in Definition 1.4.9, as explored in Bao et al. (2020); Awasthi et al. (2021); Meunier et al. (2022), which open the debate for training truly robust neural networks in practice.

In addition to the study of the loss function, a phenomenon called *the evolutionary stalling hypothesis* from Rozsa et al. (2016) has conjectured to explain the vulnerability of neural networks. This hypothesis states that the gradient of correctly classified data points becomes small so that they do not participate anymore in the model update during the training phase of the neural network, and thus, the data points are likely to be very close to the decision boundary.

The aforementioned evolutionary stalling hypothesis has also paved the way for more studies on the *decision boundary* of neural networks. As previously mentioned, Tanay and Griffin (2016) explained the success of adversarial examples with the *boundary tilting* hypothesis stating that the vulnerability of neural networks may come from the position of the decision boundary: close to the sub-manifold of the data, but tilted with respect to it. Following this work, Fawzi et al. (2016, 2018c); Moosavi-Dezfooli et al. (2019) have focused on studying the curvature of the decision boundary and its link with robustness: they tend to show that less curvature is associated with higher robustness.

On the behavior of Neural Networks (identification of critical neurons and layers). Surprisingly, few works have studied the behavior of the information flow inside neural networks with a robustness perspective: according to Han et al. (2023), only approximatively 10% of the papers focusing on model-centric explanations have explored this topic. Among the few publications about it, Cantareira et al. (2021) developed a visual framework to observe the paths taken by clean and adversarial inputs into neural networks. Similarly, Qiu et al. (2019) studied the difference between effective paths taken by clean and adversarial inputs to detect adversarial examples. A simpler strategy consists in using only the distribution of the activations of specific layers to differentiate clean and adversarial examples, as in Zheng and Hong (2018); Aigrain and Detyniecki (2019).

6.2.2 Hypothesis on the Adversarial Examples

Concurrently with the analysis of neural networks to better understand their flaws, adversarial examples are also studied to understand their strengths.

On the manifold of the data. In addition to theoretical works already mentioned in Section 6.1.4 that are based on the dimensionality of the data, some works have explored the geometry of the data and, more generally, the manifold where it lies. Stutz et al. (2019) showed that most adversarial examples deviate from the data manifold in a nearly orthogonal way, whereas some other adversarial examples stay on the data manifold but are supposed to be generalization errors. Similarly, Ilyas et al. (2019) proposed two possibilities about adversarial examples: 1) they use irrelevant directions for the classification and thus do not follow the data distribution, and 2) they use relevant directions for the classification and thus follow the data distribution. Ilyas et al. (2019) showed that the second option is likely to characterize adversarial examples, but Nakkiran (2019) also showed that there are adversarial examples following the first option. Then, *on-manifold* and *off-manifold* adversarial examples coexist.

On the features extracted from the data. The analysis of the features learned by the intermediate layers of neural networks has also been leveraged to understand adversarial examples and differentiate them from clean inputs. Among other works, Ilyas et al. (2019) also make the case for the existence of robust features versus non-robust ones, and Agarwal et al. (2019); Mustafa et al. (2019) show that the difference between classes is small in the feature space, meaning that a small perturbation can change the prediction.

6.2.3 Limitations on the Current Understanding of Adversarial Examples

As illustrated in Sections 6.2.1 and 6.2.2, many works have focused on interpreting the adversarial phenomenon, taking very different paths, strategies and perspectives to do so. However, the question of *why* and *how* adversarial examples succeed in fooling neural networks remains an open problem, mainly because of the following two limitations.

Lack of theoretical understanding. Numerous studies have addressed the theoretical limits of robustness for neural networks under specific sets of constraints. These studies

tend to demonstrate that neural networks are inherently susceptible to adversarial attacks when the neural networks and the studied attacks conform to predefined conditions. However, with the emergence of new types of attacks that better meet the requirements of real-world applications, these conditions are often not satisfied. Consequently, there is still a lack of theoretical investigation into the robustness or vulnerability of general classes of models in modern and practical settings. The work presented in Chapter 8 tackles this limitation by providing a theoretical analysis of the vulnerability of a large class of models under the threat of modern low-dimensional attacks.

Absence of unification between the investigated categories explaining adversarial examples. In the wide variety of works dedicated to understanding why adversarial examples succeed, some hypotheses emerge as popular in the community, such as the *linear hypothesis*. However, even for those lines of work, gathering enough evidence to fully support a specific result is hard, leading to hypotheses that are not formally accepted as a full explanation in the community. This lack of consensus is easily explained by the evident difficulty to derive theoretical arguments when studying neural networks in general, and specific characteristics of neural networks such as adversarial examples, which is mostly experimental phenomenon, due to the difficulty in solving the robust optimization problem from Definition 1.4.9. Thus, as illustrated by the sub-division of Section 6.2 into several categories, many aspects of neural networks and data impact (or are impacted by) the robustness of neural networks. This leads to a high division of the efforts in the field, which explains that it is hard to concatenate research findings and avenues into more global and consistent systems. Still, this aggregation step is essential to get a broader and more systematic view of the phenomenon.

The work presented in Chapter 7 tries to overcome this limitation. First, using topological tools to study under-optimized edges stems precisely from an effort to aggregate several preexisting research directions from the literature. Notably, it incorporates the previous manifold inquiries (on and off-manifold adversarial examples), characteristics of features (robust and non-robust features), neural networks' architectural properties (related to over-parametrization), and the behavior of neural networks (existence of under-optimized edges after training). Furthermore, even though it's first and foremost an experimental work, it incorporates a theoretical avenue to ground the proposed hypothesis.

Summary of contributions on evasion attacks

Chapter 7 is inspired by the following article: Morgane Goibert, Thomas Ricatte, and Elvis Dohmatob (2022). An Adversarial Robustness Perspective on the Topology of Neural Networks. In *ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).* See Goibert et al. (2022b)

It presents how the topology of neural networks impacts adversarial robustness through the in-depth study of how the information flow from an adversarial example traverses specific paths, called under-optimized edges, in neural networks. It shows that the passing of the information flow from adversarial inputs is structurally different from the one of clean inputs, suggesting 1) that the topological structure of neural networks should be taken into account to improve adversarial robustness and 2) that detecting adversarial examples as they are going through a neural network is an effective strategy.

Chapter 8 is inspired by the following article: Elvis Dohmatob, Chuan Guo, and Morgane Goibert (2023). Origins of Low-dimensional Adversarial Perturbations. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023).* See Dohmatob et al. (2023)

It presents a rigorous theoretical study of the success of heuristics based on low-dimensional attacks. It provides lower bounds on the success of such attacks under specific conditions, which are shown to be satisfied by neural networks under practical settings. The tightness of the bounds is also experimentally studied with various experiments.

Chapter 7

Adversarial Robustness Perspective on the Topology of NNs

Who knows where inspiration comes from. Perhaps it arises from desperation. Perhaps it comes from the flukes of the universe, the kindness of the muses.

Amy Tan.

Contents

7.1 Introduction and High-level Overview

Following the limitation unveiled in Section 6.2.3, this Chapter is devoted to providing a framework gathering different characteristics about adversarial examples unveiled in the literature, through the study of a generic object arising neural networks, *graph*. More precisely, this Chapter delves into a comprehensive investigation of the impact of neural network topology on adversarial robustness. Our primary focus is on exploring the structure of the graph that emerges as an input traverses through all the layers of a neural network. Remarkably, we discover distinct differences in these graphs when comparing clean inputs to adversarial inputs. Specifically, we observe that graphs derived from clean inputs exhibit a more centralized distribution around what we refer to as 'highway edges'. On the other hand, graphs associated with adversarial inputs display a more diffuse pattern, strategically leveraging 'under-optimized edges'.

To establish the significance of these findings, we conduct extensive experiments encompassing various datasets and architectures. The results consistently demonstrate that these under-optimized edges represent a notable source of vulnerability within neural networks. Furthermore, we uncover their potential utility in detecting adversarial inputs, thus highlighting their multifaceted role in the realm of adversarial robustness. Beyond these experimental findings, we provide a theoretical argument corroborating the importance of under-optimized edges for the vulnerability of neural networks and suggest that pruning techniques can provide more robustness.

By unraveling the intricate relationship between neural network topology, graph structure, and vulnerability, this Chapter provides valuable insights into the underlying mechanisms driving the susceptibility of neural networks to adversarial attacks.

7.1.1 Outline of the Rationales of the Chapter

Reminders about Adversarial Examples

Adversarial examples, as previously introduces, are perturbed versions of clean inputs destined to fool neural networks.More precisely, they have been defined in Definition 1.4.10 and is recalled here:

Definition 1.4.10. Adversarial Attack. *Let f_θ be a neural network, and $||.||$ a norm on $\mathcal{X}$. Let $\varepsilon \in [0,1]$ be the perturbation budget. An ε-practical adversarial attack is a function $A_\varepsilon : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$, defined by:*

$$\forall\, (x,y) \in \mathcal{X} \times \mathcal{Y},\ A_\varepsilon(x,y) = x + \delta_\varepsilon(x,y) \quad \text{such that } ||\delta_\varepsilon(x,y)|| \leq \varepsilon \\ \text{with } f_\theta(A_\varepsilon(x,y)) \neq f_\theta(x) \text{ as often as possible.} \tag{1.4.11}$$

The adversarial example corresponding to $(x,y) \in \mathcal{X} \times \mathcal{Y}$ is generally denoted by $x^{adv} = A_\varepsilon(x,y)$.

Classical state-of-the-art (SOTA) attacks include PGD Kurakin et al. (2017), CW Carlini and Wagner (2017) for the white-box setting (the attacker has full knowledge of the neural network), or Boundary Brendel et al. (2017) for the black-box setting (the attacker has no access to the model), which will be used later in the Chapter.

Topological Data Analysis

Topological Data Analysis (TDA), initiated by Edelsbrunner et al. (2000); Zomorodian and Carlsson (2005), is a burgeoning field at the intersection of mathematics, statistics, and computer science that offers a powerful framework for analyzing complex and high-dimensional datasets. With the ever-increasing availability of data from diverse domains, traditional data analysis techniques often struggle to capture the inherent structure and relationships embedded within the data. TDA provides a novel approach to tackle this challenge by leveraging concepts from algebraic topology to extract topological features and capture the global and local geometric properties of the data.

At its core, TDA aims to uncover the underlying shape and connectivity of data by constructing topological representations, such as simplicial complexes, persistent homology diagrams, or mapper graphs. By examining the topological features of these representations, such as the presence of loops, voids, or connected components, TDA provides insights into the global structure, clusters, and patterns within the data that may not be apparent through traditional statistical analysis or dimensionality reduction techniques.

TDA is a flexible and versatile framework that can be applied to a wide range of data types, including point clouds Collins et al. (2004); Beksi and Papanikolopoulos (2019), networks Serrano and Gómez (2020); Taylor et al. (2015), shapes Li et al. (2014); Carrière et al. (2015); Turner et al. (2014), time series Umeda (2017), and even textual Gholizadeh et al. (2018) or categorical Wu and Hargreaves (2021) data. This versatility has led to its successful application in various fields, including biology Chan et al. (2013); Amézquita et al. (2020); Skaf and Laubenbacher (2022), neuroscience Xu et al. (2021); Yamanashi et al. (2021), social sciences Almgren et al. (2017), image analysis Bernstein et al. (2020); Hu and Chung (2021), and materials science Hiraoka et al. (2016), among others.

One of the key strengths of TDA lies in its ability to handle noisy and incomplete data, making it particularly useful in domains where data quality and reliability are major concerns. In the context of adversarial robustness, as adversarial perturbation is clearly different from random or noisy perturbations, this characteristic of TDA is very useful. This Chapter critically relies on *persistence diagrams*, a TDA object able to summarize the topological structure of weighted graphs that will be properly introduced in Section 7.3.

High-level overview of the idea of the Chapter. This Chapter is devoted to showing that under-optimized edges are a main source of vulnerability for neural networks. These under-optimized edges represent parameters that are not sufficiently relevant to be fully optimized by the neural network during the training and thus represent a blind spot for the neural network. We postulate in Section 7.2 that adversarial examples target the parameters and induce a very different behavior of the information flow on their edges: namely, the information flow disperses like scattered fragments, branching out in myriad directions, before adding up to create a major change in the last layer to fool the neural network. To study and confirm the relevance of this hypothesis we create a feature extraction method detailed in Section 7.4. We first select the under-optimized edges from a neural network. Then, we use topological data analysis (and more precisely, persistent diagrams, abbreviated *dgms*) to extract structural information from these edges for each input that traverses the neural network. Finally, we compare the persistent diagrams associated with clean inputs and adversarial ones to uncover their differences. In our

experiments, in Section 7.5, not only do we notice simple qualitative and quantitative differences in the persistent diagrams, but also a detector built on these features is shown to be able to outperform state-of-the-art adversarial detection methods. These experimental results confirm the relevance of our hypothesis and are also backed up by a theoretical argument showing that over-parametrization can be a source of vulnerability, presented in Section 7.5.4.

7.1.2 Related Works

Topological data analysis and neural networks. Though some works have explored the use of TDA tools to study neural networks, e.g. Naitzat et al. (2020); Zhao and Zhang (2021); Zia et al. (2023), the body of works applying topological techniques to neural networks remains limited. In particular, only the work of Gebhart et al. (2019) has explored the use of topological tools to study adversarial examples in neural networks. Our work is thus inspired by theirs and overcomes their limitations. Namely, they reconstruct subgraphs based on the main topological structure extracted from graphs computed on neural networks traversed by inputs. These subgraphs thus represent different highway edges inside the neural network for each input. Then, they compare metric-based similarities or classical summary statistics (e.g. number of edges) between subgraphs from clean and adversarial inputs. Their conclusion is that differences exist in the subgraphs between clean and adversarial inputs. The key takeaways from Gebhart et al. (2019) is that topological tools can indeed be very relevant to study adversarial examples. However, their work has the following limitations:

- 1) Uninterpretable results: they detect differences in the topology of clean vs adversarial induced graphs, but are not able to provide an explanation stating why such differences are visible. On the contrary, in our work, we first provide a hypothesis about how adversarial examples operates, and verify this hypothesis thanks to topological tools. Our work is then aligned with the objective of improving our understanding of adversarial examples. Furthermore, contrary to Gebhart et al. (2019), we study the same edges from the neural network for all inputs, which enables us to provide information on the specific behavior of adversarial examples.
- 2) Scalability: computing a persistence diagram depends on the number of edges and neurons in the graph, which is very large even for quite small neural networks like LeNets. As Gebhart et al. (2019) compute persistence diagrams for each input on the entire NN, the computation complexity is much too high to study larger networks, and indeed, the experiments focus on 4-layer convolutional neural networks. Their method does not apply to larger networks. On the contrary, by selecting only under-optimized edges in the induced graph before computing the persistence diagram, our PD method is more scalable.

Characteristics of adversarial examples. Beyond the use of topological tools to study and enhance adversarial robustness, our work is dedicated to unifying some unveiled characteristics of adversarial examples, as found by previous work in the literature. These works are detailed in Section 7.2.1, and are briefly introduced here.

Xu et al. (2019) shows that adversarial perturbations exploit the vulnerabilities of neural networks through various strategies called 'suppressing' or 'promoting' strategies based

on the input features it targets before cascading through the network. The input features perturbed by adversaries can also be divided into two categories depending on the nature of the adversarial example, as studied mainly by Ilyas et al. (2019); Nakkiran (2019); Stutz et al. (2019): targeting useful and non-robust features characterize *on-manifold* adversaries, and targeting non-useful features characterize *off-manifold* adversaries. Additionally, over-parametrization in neural networks, characterized by an excessive number of parameters, can exacerbate vulnerability to adversarial attacks by introducing under-optimized and non-useful parameters, as shown in Rice et al. (2020); Manoj and Blum (2021); Wu et al. (2021).

Understanding the interplay between these factors is crucial for comprehending and mitigating adversarial vulnerabilities in neural networks. Our work aims at unifying all these characteristics to provide a better understanding of adversarial examples.

Detection methods for adversarial robustness. To corroborate our findings, namely that under-optimized edges are a source of vulnerability for neural networks, we propose to build a detector of adversarial examples based on the topological features extracted from said under-optimized edges. Of course, we compare our experimental results with state-of-the-art adversarial detection methods and show that our method outperforms or matches previous detectors.

The detection of adversarial examples is distinct from robustification methods: while robustification techniques aim to improve the model's resilience against adversarial attacks, detection focuses on identifying the presence of adversarial inputs. Detecting adversarial examples offers several advantages. Firstly, it provides an additional layer of defense by identifying potential threats before they can cause any harm. Secondly, it allows for the monitoring and analysis of adversarial attacks, aiding in the understanding of attack patterns and techniques, which is exactly our purpose here. The goal of an adversarial detector is thus not to improve the adversarial accuracy of a neural network, but rather to report accurately if an input is a clean or an adversarial one. The evaluation of such methods is thus based on performance metrics for 2-class classification problems, such as the False Positive Ratio or the Area under the ROC Curve.

The sub-field of the detection of adversarial examples has evolved parallelly to robustification methods, and many works have proposed efficient detectors. In this Chapter, we chose as baseline two very popular methods. The first one, Ma et al. (2018), investigates the properties of adversarial subspaces in machine learning models. They propose a method based on *Local Intrinsic Dimensionality* (LID) to analyze the local geometry of the data space and identify regions where adversarial examples are likely to occur. They demonstrate that adversarial subspaces exhibit a lower intrinsic dimensionality compared to the overall data space, allowing for effective detection. Their methods will be called *LID* in the rest of the Chapter. The second one, Lee et al. (2018), proposes to detect both adversarial and out-of-distributions inputs that leverage the observation that such examples tend to be overly confidently classified by neural networks. They thus model that the class-conditional distribution of the neural network follows a Gaussian distribution, and then compute a confidence score between an input and its closest class-conditional Gaussian distribution using the *Mahalanobis* distance. This confidence score is then fed to a threshold-based detector to differentiate adversarial (or out-of-distribution) examples from clean ones with great success. Their method will be called *Mahalanobis* in the rest

of the Chapter.

7.1.3 Outline of the Main Contributions

The main aim of this paper is to demonstrate that the analysis of the topological structure of neural networks is highly relevant to better understand, detect, and defend against the adversarial phenomenon. We pave the way for this new line of work in this paper, which is organized as follows:

- In Section 7.2, we justify and propose a hypothesis, gathering several characteristics of adversaries, on how the topological structure of neural networks and under-optimized parameters are related to the adversarial phenomenon.
- In Section 7.4, we propose the main method to extract structural topological features based on *persistence diagrams* and under-optimized edges.
- In Section 7.5, we conduct experiments to validate our hypothesis using our newly-defined features.

7.2 Unification of Adversaries Characteristics: Our Hypothesis

7.2.1 Some Characteristics of Adversarial Examples

Adversarial perturbations are small and yet result in sufficient variation of the output to change the predicted class. What happens inside a neural network to obtain this variation? We recall here three characteristics of adversaries and link them together to suggest an answer to this question and motivate the use of graphs and topological tool to study adversaries.

Strategies used by adversaries.

Xu et al. (2019) shows that adversarial perturbations can be categorized into *suppressing* ones, meaning perturbations that focus on reducing the true label score, or *promoting* ones, meaning perturbations that focus on increasing the target label score. Adversaries can (and usually do) output a mixed behavior. Interestingly, the suppressing/promoting nature of an adversary comes from the set of input features (e.g. pixels for images) it perturbs: modification in one input neuron cascades through the whole neural network and results in a suppressing/promoting relative behavior.

What features are used by adversaries?

Using Ilyas et al. (2019); Nakkiran (2019) terminology, the features of the data distribution can be divided into 1) useful and robust, 2) useful and non-robust, 3) non-useful ones. Both of these works show the existence of two types of adversaries (see also Stutz et al. (2019)), even though one can expect that most adversaries lie on a scale between these two extremes:

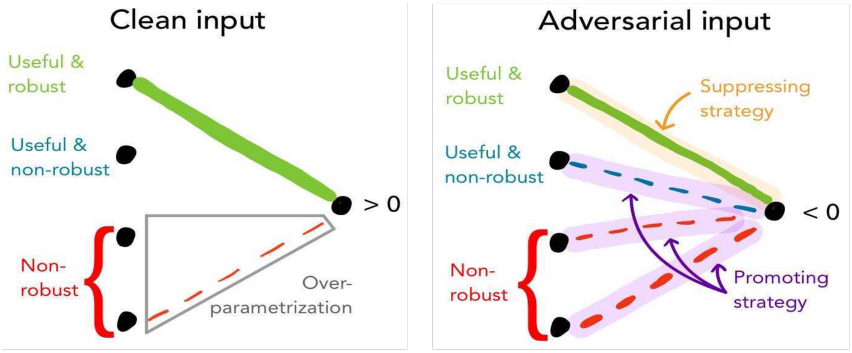


Figure 7.1: Adversarial inputs characteristics. Full (dashed) lines denote positive (negative) weights.

- Adversaries leveraging useful and non-robust directions: e.g. when an image from the class "dog" is perturbed to be classified as a "cat", the perturbation has something to do with the class "cat". Then, the adversary is on-distribution (the direction of the perturbation is parallel to the data manifold, thus the adversary does not leave the data manifold).
- Adversaries leveraging non-useful directions: e.g. the image from class "dog" is perturbed with a perturbation that has nothing to do with class "cat". Then, the adversary is off-distribution because the perturbation can occur in any arbitrary direction (the direction of the perturbation is perpendicular to the data manifold, thus the adversary leaves the manifold).

Over-parametrization.

The link between over-parametrization and robustness is still not completely understood. However, some works (e.g. Rice et al. (2020); Manoj and Blum (2021); Wu et al. (2021)) have shown that neural networks vulnerability may increase when they are over-parametrized. It occurs when a neural network has too many parameters: after training with e.g. SGD, parameters in excess still have non-zero values, and thus are used for prediction.

It enables highly curved decision boundaries Liu and Shen (2022) and can lead to over-fitting the training data. Thus, over-parametrization can translate into having a neural network with many under-optimized and non-useful parameters for the classification task at hand. These non-useful parameters can be leveraged to build adversarial attacks (e.g. via *promoting* behaviors). Such a behavior is the most expected one for standard neural networks, because they usually are over-parametrized, and most attacks (e.g. PGD) use non-useful directions to perturb clean inputs Stutz et al. (2019). In the alternative case where under-optimized and non-useful parameters are removed (by e.g. pruning), adversarial perturbations can still leverage useful but non-robust parameters to create on-distribution adversarial examples.

Figure 7.1 illustrates these characteristics, leading the neural network to classify the clean input (resp. adversarial input) as a positive (negative).

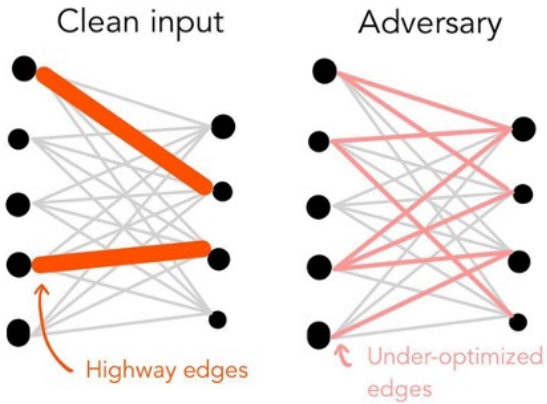


Figure 7.2: Blueprint of structural differences between graphs from clean vs adversarial inputs.

7.2.2 The Under-optimized Edges Hypothesis

Based on the observations from Section 7.2.1, and the fact that most neural networks are over-parametrized (i.e parameter count exceeds training dataset size) and that pruning away most parameters after training induces smaller models without degrading accuracy, as explored in Frankle and Carbin (2019), we hypothesize that only a small set of parameters are critically used for inference of clean inputs, while the rest of the parameters do not carry meaningful information. Considering a neural network as a graph, and parameters as edges of that graph, this means that information from clean inputs flows through highway edges, while information from adversarial inputs is more diffuse, and uses so-called under-optimized edges (i.e. useless edges not well optimized during training). This results in *structural differences* in graphs induced by clean and adversarial inputs, as simply illustrated by Figure 7.2. Using the notion of *induced graph*, which is a weighted graph representing the information flow from an input in a neural network/graph, and defined later, we can sum up our hypothesis:

Our Hypothesis. *Clean and adversarial inputs induce differences in the topological structure in their respective induced graphs, because under-optimized edges are used by adversaries, but not by clean inputs. Such edges are thus a source of adversarial vulnerability.*

7.3 Introduction to Topological Data Analysis

Here, we only provide a simple overview and some intuitions about the concepts we use, but the interested reader can find more details in Chazal and Michel (2017).

Simplicial complexes.

A simplicial complex is a topological object generalizing the notion of triangulation, composed of vertices and edges, as illustrated by Figure 7.3. Up to some constraints, it is a set of simplexes, where a n-simplex is a triangle in dimension n. We can smoothly compute their homology groups, whose elements, homology classes, represent different structural "holes" and are our relevant topological information. A graph, like our induced graphs, is of course composed of vertices and edges and thus can be seen as a simplicial complex.

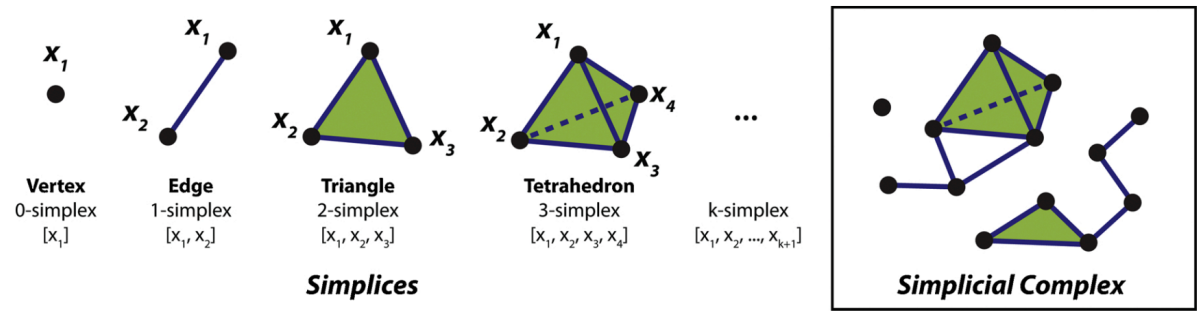


Figure 7.3: Illustration of simplices and simplicial complex. Courtesy of Zhang et al. (2020).

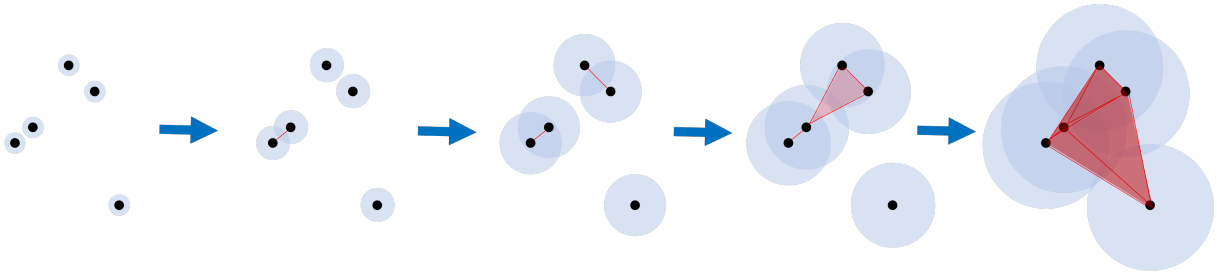


Figure 7.4: Illustration of a filtration. Each point is associated with a same-sized sphere whose diameter t is growing: this diameter is the filtration parameter. When two spheres intersect, the two vertices connect to form a new simplicial complex: this creates a nested inclusion of simplicial complexes

Persistence diagrams.

Intuitively, persistent homology aims to capture the essential topological features of a simplicial complex at multiple scales. To do so, persistent homology examines the evolution of homology groups as a parameter, typically known as the 'filtration parameter', varies. The filtration parameter encodes the notion of scale or proximity in the data set and basically enables the creation of an increasing sequence of simplicial complexes based on the inclusion order, as illustrated by Figure 7.4.

To understand how persistent homology works, let's consider a point cloud data set in a two-dimensional space, as in Figure 7.4. Initially, at a very low filtration parameter, each data point is considered as a separate component, and the homology groups are trivial. As the filtration parameter increases, the data points start to form clusters, and the homology groups detect the presence of connected components or holes. These groups are algebraic constructs that quantify the number and nature of connected components, holes, voids, and higher-dimensional voids in a simplicial complex.

The concept of persistence comes into play by tracking the birth and death of topological features as the filtration parameter increases. A feature is considered 'born' when it first appears in the data set and considered 'dead' when it merges or disappears. The persistence of a feature measures how long it exists over a range of filtration parameter values. Persistent homology captures these birth and death events and provides a way to visualize and quantify the longevity of topological features.

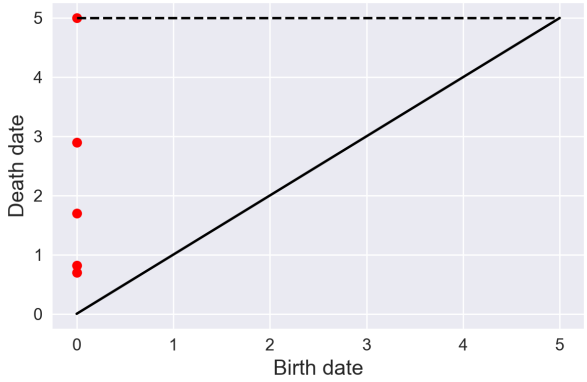


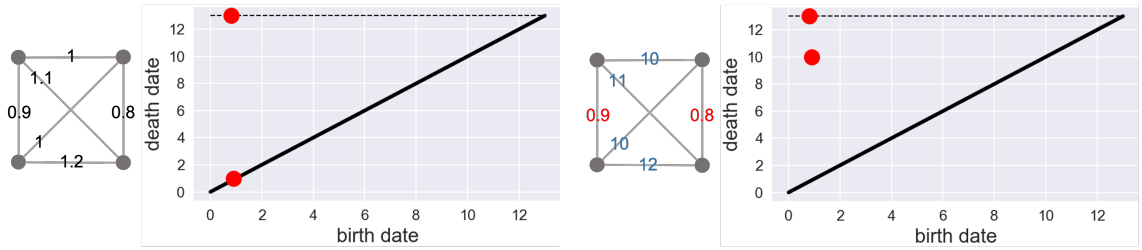
Figure 7.5: Illustration of a persistent diagram.

Persistent homology is often represented using a persistence diagram. In a persistence diagram, each topological feature, such as a connected component or a hole, is represented by a point in a two-dimensional plot. The x-coordinate represents the filtration value at which the feature is born, and the y-coordinate represents the filtration value at which it dies. For example, the persistent diagrams (of 0^{th}-dimension) of the filtration in Figure 7.4 is presented in Figure 7.5, where the dashed line corresponds to infinity. As can be seen, the points farthest from the diagonal correspond to data points that are far from the rest of the points, whereas those close to the diagonal correspond to close points. This means that in a persistent diagram, points close to the diagonal can be identified with noise, while points far from the diagonal can be identified with important features. Persistent diagrams thus offer a concise representation of the evolution of topological features in a simplicial complex.

Intuitions and illustrative example for neural networks.

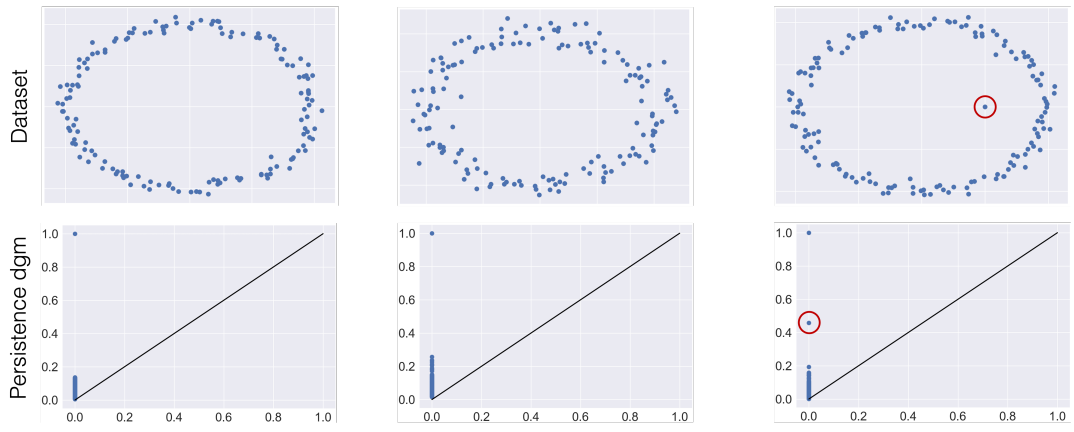
As our graphs are feedforward and do not represent 3-d objects, we focus our analysis on the 0^{th}-dimensional persistence diagrams. The sub-complex for parameter t thus is the sub-graph composed of edges with weights smaller than t (and corresponding neurons). The filtration is the collection of sub-complexes from $t = 0$ (empty graph) to $t = +\infty$ (whole graph). Intuitively, the persistence diagram then represents how the connected components of the sub-complexes evolve through different spatial scales given by the weights of the graph. Highly connected subsets of edges (with small edge weights) will form a connected component during many sub-complexes: it will create a point in the persistence diagram with a long lifetime, far from the diagonal, representing an important structural feature for the whole graph. An illustration is given in Figure 7.6. Notice that with this natural definition of sub-complexes, a small-weighted edge corresponds to an important edge, as it connects two neurons with close spatial proximity. In an induced graph $G(x, g)$, edge weight denotes information flow, not spatial proximity: a high-weighted edge thus corresponds to an important edge. To circumvent this issue, we replace the weight $w > 0$ with its opposite $-w$.

Difference between adversarial and noisy perturbations in persistent diagrams. Persistence diagrams can identify the structural properties of points clouds or graphs. In dimension 0, as previously stated, points in persistence diagrams represent the lifetime of



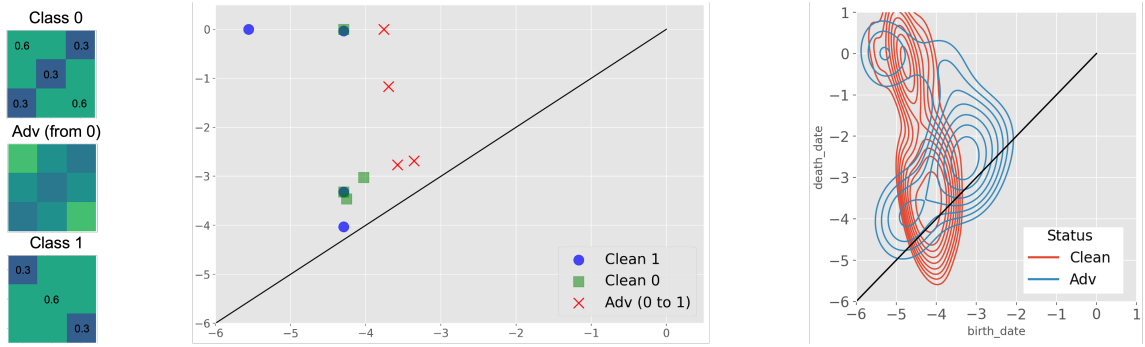
(a) A regular graph and its persistent diagram (b) A structured graph and its persistent diagram

Figure 7.6: Two graphs with different topological structures and their corresponding persistent diagrams (dashed lines correspond to infinity). In (a), the weights are similar: the only important subgraph is the whole graph, thus one point is far from the diagonal. In (b), there are two edges with much smaller values than the others (red): they form two important subgraphs, thus two points far from the diagonal.



(a) Noisy circle 1 (b) Noisy circle 2 (c) Adv. circle

Figure 7.7: Persistence diagrams are stable to random noise, not to adversarial noise.



(a) Toy dataset (b) Dgms of four inputs illustrating transition phase (c) KDE plots of clean vs adv dgms

Figure 7.8: Persistence diagrams from clean vs adv inputs are highly dissimilar.

connected components. An interesting property of persistence diagrams is that they are *robust to noise*. It means that two noisy circles (the points in the dataset were generated following a circle equation to which a Gaussian noise with mean= 0 and different standard deviations) will output very similar persistence diagrams. However, non-random noise, such as adversarial noise, can deeply modify the persistence diagram. We illustrate this feature in Figure 7.7. In the 'adversarial' circle, we see that even though there is only one adversarial point in the dataset, its position induces the presence of an abnormal point in the corresponding persistence diagram (emphasized with a red circle), whereas the two versions of the noisy circle dataset on the left output very similar diagrams.

The robustness to noise property of persistence diagrams should result in having similar clean persistent diagrams (especially for inputs from the same class), but different from adversarial persistent diagrams because adversarial perturbations are non-random. Stemming from these non-random shifts in the structure of the induced graphs, we also expect a clear transition phase from the clean regime to the adversarial one. Since persistent diagrams from classical tasks such as MNIST / LeNet have way too many points to be visually understandable, we trained a classical NN with one convolutional layer and two dense layers on a toy dataset. The dataset is a binary classification task on 3x3 images, where each pixel of an input conditionally to its class is drawn independently from a normal distribution with standard deviation= 0.05, and means as shown in Figure 7.8a. Our simple model outputs a standard accuracy of 0.99. Now, let us explore what persistent diagrams from clean vs adversarial inputs look like. We generated adversaries using PGD with $\varepsilon = 0.1$. In such a small setting, all persistent diagrams have very few points. However, even in this simple setting, we can illustrate that our hypotheses hold.

Figure 7.8b shows that persistent diagram from an adversary (created from a class 0 input, predicted as class 1) outputs a different behavior than the two clean ones: in addition to having larger birth dates, there is a particular point with a birth date and death date that do not correspond to any other point from either class 0 or class 1 diagrams. This behavior leads to a high distance between the adversarial diagrams and the clean diagrams from both classes. Figure 7.8c clearly shows, through a density estimation of points in the persistent diagrams from adversarial and clean inputs, that clean diagrams points lie in two very specific spots, whereas adversarial diagrams points are more dispersed, meaning that clean persistent diagrams (event from the two different classes) are quite similar,

contrary to adversarial persistent diagrams.

7.4 Extraction of Topological Features – Methods

As Sections 7.2 and 7.3 have introduced both the main goal of the Chapter and the tools from topological data analysis we will use, we now explore our methodology to extract the persistent diagrams from neural networks and inputs.

7.4.1 Retrieval of the Induced Graph

Definition and intuition.

Let $\mathcal{X} = \mathbb{R}^{n_0}$ be the feature space, where n_0 is the input dimension. For any input $x \in \mathcal{X}$, the induced graph (also called the activation graph) is a graph on the neurons of the network, whose edges depend both on the parameters of the network and the inner activations induced by the forward pass of x.

Formally, a neural network on a K-classification problem is a function $f_\theta : \mathcal{X} \to [\![1, K]\!]$ of the form $f_\theta(x) = \text{argmax}_{k=1,\dots,K}\, g_\theta(x)$ where $g_\theta : \mathcal{X} \to \mathbb{R}^K$ is the feature map. In the case of a multilayer perception, it can be more precisely defined as follows:

Definition 1.4.1. MULTILAYER PERCEPTRON (MLP). *Let $\mathcal{F}_\Theta$ be a (parametric) model class. $f_\theta \in \mathcal{F}_\Theta$ is a multilayer perceptron with L layers if and only if:*

$$f_\theta(x) = \underset{k=1,\dots,K}{\text{argmax}}\, g_\theta(x) \quad \forall\, x \in \mathcal{X}, \text{ with} \tag{1.4.1}$$

$$g_\theta(x) = W_L \sigma_{L_1}\left(W_{L-1}\sigma_{L-2}(\dots\sigma_1\left(W_1 x + b_1\right)) + \dots + b_{L-1}\right) + b_L, \tag{1.4.2}$$

where $g_\theta : \mathcal{X} \to \mathbb{R}^K$ is the feature map, $\forall\, l \in [\![1, \dots, L]\!], \sigma_l$ is the activation function (e.g. a ReLU function), and $\theta = (W_i, b_i)_{1 \le i \le L}$ are the parameters.

With a slight abuse of notation, we denote by $g_{\theta,l}(x) \in \mathbb{R}^{n_l}$ the output value of layer l.

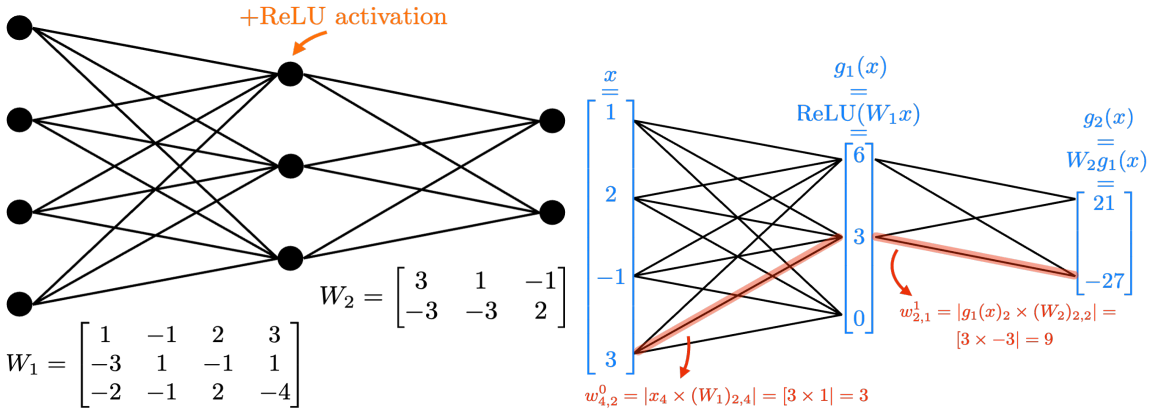
Combining information from the feature map g_θ, identified with the neural network f_θ, and an input $x \in \mathcal{X}$, we construct the so-called *induced graph*.

Definition 7.4.1. INDUCED GRAPH. *Let g_θ be the feature map of a MLP with parameters $(W_i, b_i)_{1 \le i \le L}$, $x \in \mathcal{X}$ be an input. The induced graph corresponding to g_θ and x is denoted by $G(x, g_\theta)$ and defined by:*

$$G(x, g) = (V, E), \text{ with } V = \{1, 2, \dots, n_0 + \dots + n_L\}$$
$$\text{and } E = \{(u^l, v^{l+1}, w^l_{u,v})\} \subseteq V^2 \times \mathbb{R},$$

where $w^l_{u,v} = |\, [g_{\theta,l}(x)]_u \times (W_l)_{v,u} + b_l \,|$

In this simple case, the edge weights are the value of the parameter weight of the neural network between neurons u and v multiplied by the activation of neuron u (plus the bias, which we will discard in general to simplify the notations): this definition of $w^l_{u,v}$ is meant to mimic how neural networks operate to transfer information from a layer to the next. It applies to feedforward neural networks, and can also be generalized to other structures like ResNet. Moreover, the $w_{u,v}$'s can also be obtained for convolutional layers or others as will be explained afterward.



(a) Trained neural network (b) Induced graph

Figure 7.9: A trained neural network (a) and its corresponding induced graph for an input x (b). We highlighted the *activation values* at each layer (blue), i.e. the values of the neurons. We also provided the weights for two edges (red), which denotes the information flow from input x carried by the edge.

Figure 7.9 provides a simple illustration of the way an induced graph is computed for a dense layer. Figure 7.9a shows a trained neural network, with the weights for each layer written in the matrices. For an input $x = (1, 2, -1, 3)$, Figure 7.9b shows the corresponding induced graph.

Practical computation.

We explore more into details how to compute the induced graph, for simple dense layers as well as convolutional ones.

Step 1: Get the activations by layer. As described before, the induced graph depends both on the parameters of the networks and on the inner activations induced by x. Therefore, the first step is to perform a forward pass through our network and save all the intermediate activations (note that, in practice, we only focus on a subset of the layers as detailed in Figure 7.12). For layer l, recall that by $g_{\theta,l}(x) \in \mathbb{R}^{n_l}$ denotes the the inner activation.

Step 2: Matrices per layer. To compute the induced graph, we need to weight the activations by the strength of the connection between neurons. For a linear layer parametrized by a weight matrix $W_l \in \mathbb{R}^{n_{l+1} \times n_l}$, this is straightforward and we can write:

$$w_l = W_l g_{\theta,l}(x) \quad .$$

For a convolutional layer, we need first to compute an equivalent weight matrix W_l from the kernels K_l (the 'sparse fully connected counterpart'). When padding= 0, stride= 1 and nb_channels= 1, we can notice that the equivalent matrix is simply composed of Toeplitz matrices based on each row of K_l stacked by block. Here is an example.

$g_{\theta,l}(x)$ is the stacked version of $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ so that $g_{\theta,l}(x) = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}^T$

and $K_l = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$. Then

$$W_l = \begin{bmatrix} 10 & 20 & 0 & 30 & 40 & 0 & . & . & . \\ 0 & 10 & 20 & 0 & 30 & 40 & . & . & . \\ . & . & . & 10 & 20 & 0 & 30 & 40 & 0 \\ . & . & . & 0 & 10 & 20 & 0 & 30 & 40 \end{bmatrix}$$

where the Toeplitz matrices are $T_1 = \begin{bmatrix} 10 & 20 & 0 \\ 0 & 10 & 20 \end{bmatrix}$ and $T_2 = \begin{bmatrix} 30 & 40 & 0 \\ 0 & 30 & 40 \end{bmatrix}$

The reasoning is similar in the general case where nb_channels ≥ 1, stride $\neq 1$ and padding ≥ 0. In practice, we leverage the sparseness of these matrices when we build them and use the Numba package to accelerate the computations.

Note that the weight matrices per layer are computed once at the beginning of the process so that we can simply multiply W_l and $g_{\theta,l}(x)$ to assemble the induced graph.

Step 3: Get the induced graph. The induced graph is represented by its adjacency matrix $A \in \mathbb{R}^{n_1 \dots n_L \times n_1 \dots n_L}$. For neural networks without any shortcuts (unlike ResNets for example), A can be obtained by constructing a diagonal matrix by block, where the l-th block is simply the induced matrix of layer l.

7.4.2 Selection of Under-Optimized edges

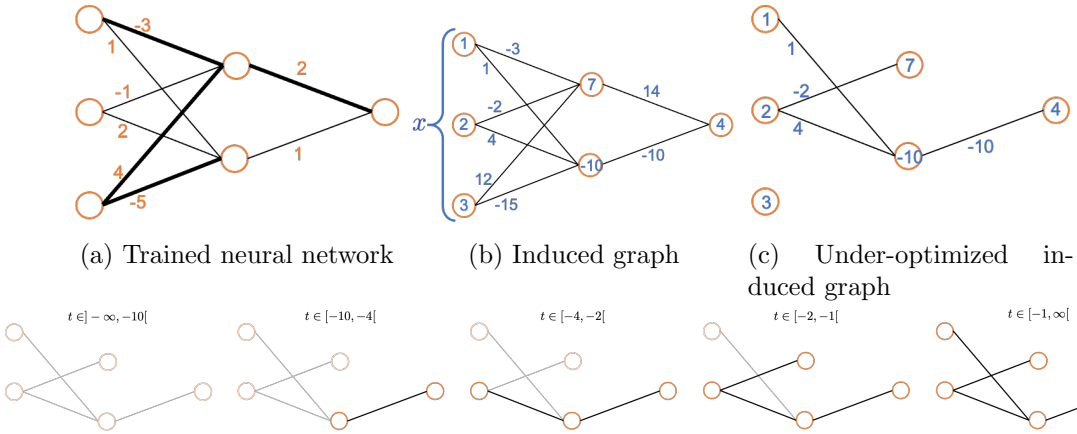
As classical neural networks have a huge number of parameters (even for small ones as LeNet), it is necessary to reduce dimensionality and select a sub-graph of the induced graph. Moreover, as we expect adversaries to leverage *under-optimized* edges, we select only these edges for our analysis. As defined and studied in Frankle and Carbin (2019); Zhou et al. (2019), an edge (u, v) is under-optimized if the *Magnitude Increase* (MI) quantity $|(W_l)_{u,v}| - |(W_l^{init})_{u,v}|$ is small, $(W_l^{init})_{u,v}$ being the parameter's initialization value. An edge (u, v) of layer l is kept in the thresholded induced graph if and only if:

$$|(W_l)_{u,v}| - |(W_l^{init})_{u,v}| < \text{quantile}(q) \quad , \tag{7.4.1}$$

where q is the target fraction of edges to keep. We denote the *thresholded induced graph* as $G^q(x, g_\theta)$. Note that no assumption is made over the initialization of the neural network and that the selection criterion of under-optimized edges does not depend on the input x, but only on the neural network g.

7.4.3 Computation of Persistent Diagrams

We use Dionysus, developed in Morozov (2017), to compute the Persistent Diagram from a custom filtration where each edge (u, v) appears at time $-|w_{u,v}^l|$ (strongest links appear first). An illustration of this process is given in Figure 7.10. The persistence diagram we obtain is just a vector of tuples, containing the birth and death dates of every point in the persistence diagram. More practically, we used the following simplified Algorithm 7.1 to compute persistent diagrams. Our feature extraction method that we just described will be often referred to as the *PD* method.



(a) Trained neural network (b) Induced graph (c) Under-optimized induced graph

(d) Filtration. Corresponding persistent diagram: $\{(-5, \infty), (-3, \infty)\}$.

Figure 7.10: Persistence Diagram illustration - If we have a simple linear neural network with its trained parameters in Figure 7.10a (for simplicity, the initial values of the parameters were set to 0) and the selection parameter $q = 0.5$, then: 1) we select only the *thin* edges, not the thick ones, in Figure 7.10a. 2) An example x flows through the graph so that we obtain the corresponding induced graph in Figure 7.10b. 3) Applying our selection parameter $q = 0.5$, we restrain ourselves to the under-optimized induced graph in Figure 7.10c. 4) The corresponding filtration is given by Figure 7.10d.

Algorithm 7.1: Persistence Diagram embedding algorithm

Input : a feature map g_θ with parameters W (after training) and W^{init} (at initialization); a dataset $\mathcal{D}$; a parameter q; the SW kernel K_{PD}.

Output: An embedding dataset $\mathcal{F} = \{\Phi_{\mathsf{PD}}(x, g_\theta) \,|\, \forall x \in \mathcal{D}\}$

for *each* $x \in \mathcal{D}$ **do**

  for *each pair of connected layers* (l, l') **do**

    `/* 1 - Adjacency matrices */`

    - Get $W_{l,l'}$ (parameter matrix) and $g_{\theta,l}(x)$ (output of layer l);

    - Compute $\forall i, j \; [A_{l,l'}(x)]_{i,j} = |[g_{\theta,l}(x)]_i * [W_{l,l'}]_{i,j}|$;

    `/* 2 - Selecting under-optimized */`

    for *each matrix indexes* (i, j) **do**

      if $|[W_{l,l'}]_{i,j}| - |[W^{init}_{l,l'}]_{i,j}| \geq quantile(q)$ **then**

        $[A_{l,l'}(x)]_{i,j} \leftarrow 0$;

  `/* 3 - Global adjacency matrix */`

  Create $A(x)$ by stacking by block the $A_{l,l'}(x)$;

  `/* 4 - Persistence Diagram */`

  - Compute $\Phi_{\mathsf{PD}}(x, g_\theta) = \mathsf{PD}(A(x))$;

  - Add $\Phi_{\mathsf{PD}}(x, g_\theta)$ to $\mathcal{F}$;

7.4.4 A Simpler Method Based on Raw Graphs

In addition to our main PD method, we also explore a much simpler one. Thus the purpose of this method, called *Raw Graph* (RG) is to use the simplest features from the induced graphs, namely just the weights of the edges of the thresholded induced graph $G^q(x, g_\theta)$. This leads to a feature mapping

$$\Phi_{\text{RG}}(x, g_\theta) = \mathsf{Vec}(W), \tag{7.4.2}$$

where W is the matrix of weights of the thresholded induced graph $G^q(x, g_\theta)$.

The goal of Raw Graph is to compare our PD method to a simpler setting where the information from the induced graph is not looked at from a *structured* or *topological* point of view. Therefore, Raw Graph will help us understand how much the structural properties of the information flow are important, compared to the raw information flow in itself. If the Raw Graph method does not match the PD method to differentiate clean and adversarial examples, it would mean that not only under-optimized edges are an important source of vulnerability, but also that adversarial examples exhibit complex behavior that does not just perturb the under-optimized edges, but also target their structural organization.

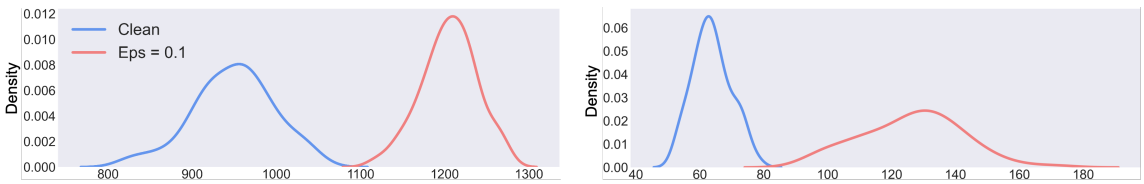
7.5 Experiments

7.5.1 Qualitative Differences in a Simple Setting

When the induced graphs are sufficiently small, differences in their persistent diagrams can be easily observable based on the number of points in the diagrams extracted from our PD method. Figure 7.11 shows this is the case for a classical MNIST / LeNet, where adversaries were computed using PGD Kurakin et al. (2017) with $\varepsilon = 0.1$. More precisely, in this simple setting, even for an attack of a small size, a perfectly accurate difference can be made between clean and adversarial inputs by just counting the number of points in their respective persistent diagrams. Thus, this can be an efficient strategy to differentiate adversarial inputs from clean ones in this simple setting, but it is not enough in more complex settings, as will be illustrated in Section 7.7.1.

7.5.2 Detecting Adversarial Examples – Method

While differences in persistent diagrams are easily observable on simple setups, it is necessary to extend our analysis to more complex, state-of-the-art setups. Even though not as easily observable in these cases, we derived a detection framework based on PDs, which



(a) Distribution of all PD points. (b) Distribution of infinitely-lived PD points.

Figure 7.11: persistent diagram points computed on MNIST / LeNet

can be used for any dataset and architecture, whose success shows that adversarial persistent diagrams (and thus adversarial inputs) are indeed different from clean ones, for a variety of SOTA attacks (PGD Kurakin et al. (2017) and CW Carlini and Wagner (2017) for the white-box setting, Boundary Brendel et al. (2017) for the black-box one) and datasets (MNIST, Fashion MNIST, SVHN, CIFAR10), using LeNets and ResNets architectures. Our code is available at: https://github.com/detecting-by-dissecting/detecting-by-dissecting.

Training details.

The usual procedure was used for training, by separating the datasets into training, validation, and test sets and using an Adam optimizer (for LeNets) and an SGD optimizer (for ResNets). The learning rate was set to 0.001 for the LeNets, and a one-cycle policy (see Smith (2017)) with varying learning rates in the range $[0.008, 0.12]$ for SVHN and CIFAR10 ResNets. The number of epochs was set to 50 for MNIST LeNet and 100 for the others.

Note that the ResNet32 model used for CIFAR100 was a pre-trained model without further training, downloadable here: https://github.com/chenyaofo/pytorch-cifar-models/releases/download/resnet

We ran all our experiments on a computer equipped with 1 GPU (Tesla V100-PCIE-16GB) and 60Gb of RAM.

Attacks details.

Recall that PGD attack (Kurakin et al., 2017) is defined by: $x_0^{adv} = x$ and $x_{t+1}^{adv} = \text{Clip}_{x,\varepsilon}\left(x_t^{adv} + \varepsilon_{\text{iter}}\,\text{sign}\left(\Delta_x l(\theta, x, y)\right)\right)$. for each $t \in [\![1, T]\!]$, where l denotes the loss. In our experiments, we set $T = 50$ and $\varepsilon_{iter} = 2 * \varepsilon/50$ and different ε values (reported in the results).

The objective of CW (Carlini and Wagner, 2017) is to find $\delta^* = \text{argmin}_\delta ||\delta||_2 + cf(x+\delta)$ with f a well-chosen function. In our experiments, we set the number of binary search steps to find c to 15; the number of iterations to optimize the objective function to 50 (Adam optimizer).

Experimental pipeline.

There are 3 steps in the detection pipeline:

- *1) Pre-processing.* We create first a (successful) adversarial dataset by running an attack on the neural network and clean inputs. For the clean dataset, we keep only examples that were not involved in the creation of the adversarial dataset.
- *2) Feature extraction.* We apply our methods (or state-of-the-art baselines) to the clean and adversarial datasets (see Algorithm 7.1 for PD).
- *3) Detector.* An SVM is trained with the features of each method, and its outputs enable us to compute any detection metric (namely the AUC).

Moreover, we ran *unsupervised* and *supervised* experiments. Supervised ones use adversarial data during training: by assuming something about the type of attack, they are

uninformative about the generalization ability of the method (they give a false sense of security). The unsupervised experiments are using a one-class SVM trained only on clean data: it is a better setting to evaluate detection methods. We only show unsupervised results in this Section (see Section 7.7.2 for supervised results, where our method still outperforms state-of-the-art methods). Note then that state-of-the-art results are not as high in this unsupervised setting compared to the results reported in other papers.

Computing the AUC.

As a reminder, when computing the AUC, the attack method (and the attack strength) and the detection parameters (like the parameter q for our method) are given. To compute this score, the SVM needs to have a kernel as input. To compute distances between different PDs extracted using our method with $\Phi_{\mathsf{PD}}(x, g_\theta) := \mathsf{PD}(G^q(x, g_\theta))$, we used the Sliced Wasserstein Kernel, defined in Carriere et al. (2017) by:

$$K_{\mathsf{PD}}(x, x') = \exp\left(-\frac{1}{2\sigma^2}\,\mathsf{SW}(\Phi_{\mathsf{PD}}(x, g), \Phi_{\mathsf{PD}}(x', g))\right),$$

where $\mathsf{SW}(\cdot, \cdot)$ is the Sliced-Wasserstein distance between persistence diagrams.

For the three other methods (RG, LID and Mahalanobis), the kernel used was just the classical *Radial Basis Function* (RBF) kernel, defined as:

$$K_\Phi(x, x') = \exp\left(-\frac{1}{2\sigma^2}\|\Phi(x) - \Phi(x')\|^2\right), \tag{7.5.1}$$

where Φ denotes the features for each method, e.g. $\Phi_{\mathrm{RG}}(x) := \Phi_{\mathrm{RG}}(x, g_\theta) = \mathrm{Vect}(W^q(x, g_\theta))$, where $W^q(x, g_\theta)$ is the matrix of weights of the under-optimized induced graph $G^q(x, g_\theta)$.

SVM outputs scores for each input: if it is above a discrimination threshold, the input is flagged as clean (otherwise, flagged as adversarial). The ROC curve is a plot representing the True Positive Rate (TPR) as a function of the False Positive Rate (FPR) when the discrimination threshold varies. The AUC is the integral of the ROC function (so that the discrimination threshold is integrated out), and represents how well the detector can separate the two classes (the higher the AUC, the better).

Confidence Interval. The main source of variability of a run comes directly from the variability of the dataset. For a fixed detector, we denote by $P_{X,Y}$ the distribution of the images. We want $[p, q]$ that satisfies (80%-confidence interval)

$$\mathbb{P}_{P_{X,Y}}\{AUC < q\} = 0.1 \text{ and } \mathbb{P}_{P_{X,Y}}\{AUC > p\} = 0.1$$

To estimate $[p, q]$, we use resampling and estimate the AUC on 100 bootstraps of size $n//2$ (where n is the total number of samples). It can be shown (see for instance Johnson (2001)) that a good approximation of $[p, q]$ is given by

$$[2\hat{AUC} - c_{90}, 2\hat{AUC} - c_{10}] \ ,$$

where $\hat{AUC}$ is the AUC estimated on the n samples, c_{10} (resp. c_{90}) is the 10-th percentile (resp. 90-th percentile) of the 100 bootstrapped AUCs.

Models	Max percentile q	List of layers
MNIST LeNet	0.025	All layers
Fashion MNIST Lenet	0.05	All layers
SVHN ResNet	0.275	Last conv. and linear layers
CIFAR10 ResNet	0.3	Last conv. and linear layers

Figure 7.12: Selection parameter used for PD and RG methods in the experiments

Models	Nearest Neigh. %	Batch size
MNIST LeNet	0.08	250
Fashion MNIST Lenet	0.02	250
SVHN ResNet	0.05	150
CIFAR10 ResNet	0.1	50

Figure 7.13: LID parameters used in the experiments

Selection of hyperparameters.

We cross-validated the parameter values for all parameters presented below, and kept only the best ones that were used afterward in our experiments.

Selection parameter for PD and RG methods. Recall that the parameter used for our PD and RG methods is denoted by q: it is the proportion of edges kept for the construction of the induced graph. We use the same value q for selected layers (uniform selection), thus we have to identify the layers kept in the analysis, and then find the parameter to use for all these layers. Note that the parameter was optimized on the PD method, and kept the same for the RG method. These parameters are shown in Figure 7.12.

Hyperparameters for the LID method. LID has two parameters that we cross-validated, and are shown in Figure 7.13.

Hyperparameters for Mahalanobis method. Mahalanobis has two parameters: the first one, $\epsilon_{\text{preprocessing}}$, controls the size of the noise added to the input, in order to make in- and out-of-distribution samples more separable. We set this parameter to 0.0. The second one is the layer selected for the analysis. When it was available (for the two setups using ResNet), we used the same layers as the one used by the authors of Mahalanobis in Lee et al. (2018). For the experiments using LeNet, we kept the last two linear layers. The parameters are thus shown in Figure 7.14.

Models	Selected leyers
MNIST LeNet	Last two linear layers
Fashion MNIST Lenet	Last two linear layers
SVHN ResNet	Last layer of each four ResNet block
CIFAR10 ResNet	Last layer of each four ResNet block

Figure 7.14: Mahalanobis parameters used in the experiments

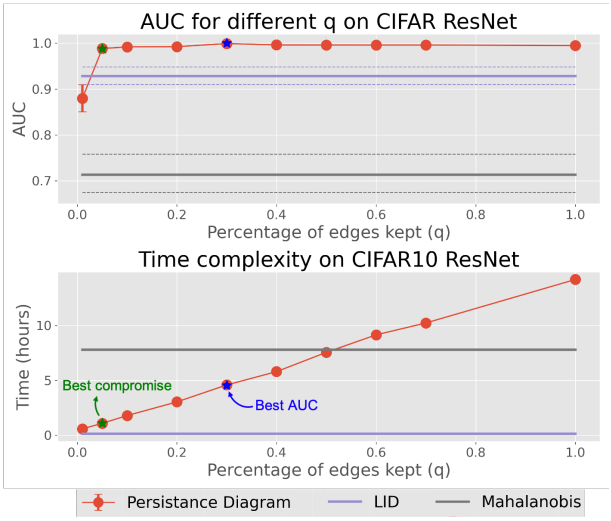


Figure 7.15: Detection AUC (up) and time (down) as a function of q (CIFAR10 ResNet vs PGD $\varepsilon = 0.05$).

In addition, note a substantial difference between our experiments and theirs when evaluating against PGD attack: the ε parameter in Lee et al. (2018)'s implementation corresponds to ε_{iter} in our paper: thus, when they run a PGD attack with strength ε, the resulting perturbation is much higher, of size $\varepsilon \times$ number of iteration for PGD. This leads to better detection results since they evaluate on much stronger attacks.

Details on time complexity.

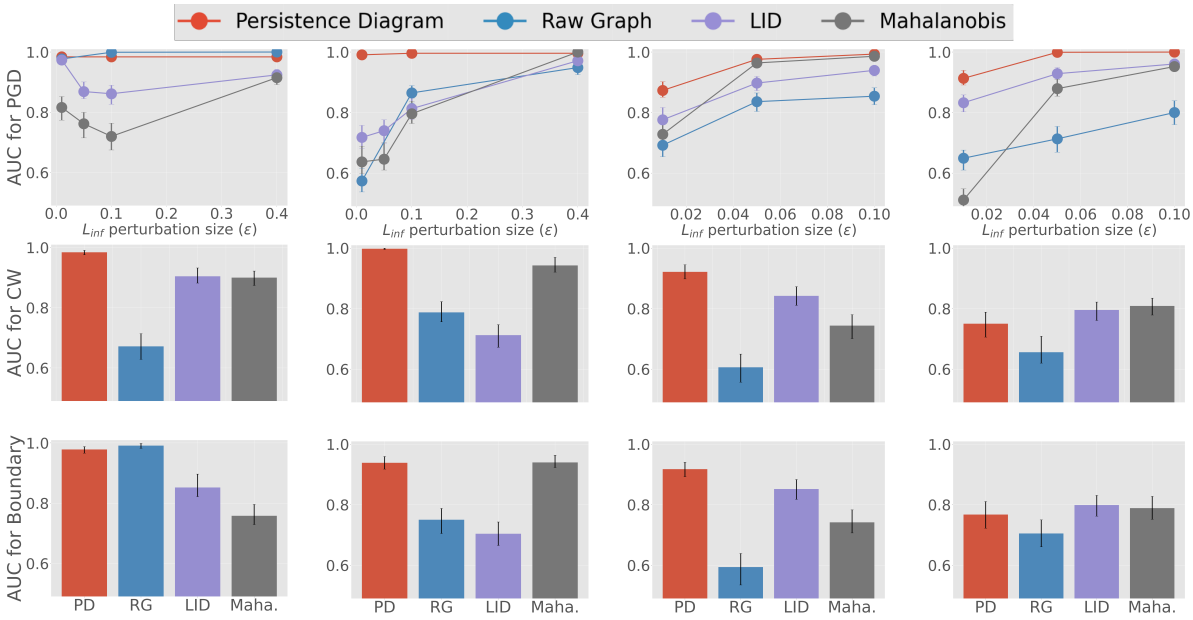
Figure 7.15 illustrates the fact that the time complexity of our PD methods grows linearly with parameter q. However, one can see that even small values of q yield great detection results, with almost no compromise on the AUC (green star). Note that Mahalanobis requires the estimation of large precision matrices (one for each considered layer, of size nb neurons x nb neurons), which makes it substantially slower than LID.

7.5.3 Detecting Adversarial Examples – Results

Based on this PD-based feature extraction method and a kernel, we can build a detector using a simple SVM. We compare our method, called PD, to state-of-the-art detection baselines: *Mahalanobis* created in Lee et al. (2018) and *Local Intrinsic Dimension (LID)* created in Ma et al. (2018). For the sake of comparison, we also compare our PD method with our very simple one called *Raw Graph (RG)*, whose features are just a vector whose elements are the weights of the thresholded induced graphs $G^q(x, g_\theta)$.

Figure 7.16 presents the AUC detection results for the different methods, against our three attacks and four setups. PD has better AUC results than state-of-the-art methods on the four datasets and architectures and on all attacks, except on CIFAR10 ResNets, where the results are similar. RG remains competitive with the two baselines on the (small) LeNet architectures. The main takeaways of these experiments are:

- RG's performances indicate that useful information can indeed be found in the thresholded induced graph, thus in the under-optimized edges. However, such a



(a) LeNeT/MNIST (b) LeNet/F-MNIST (c) ResNet/SVHN (d) ResNet/CIFAR10

Figure 7.16: Showing detection AUC for different detection methods (legend) against different kinds of adversarial attacks (rows) and model architectures and datasets (columns). We see that our proposed method based on PD outperforms the state-of-the-art methods, except for one tie.

simple method is only efficient on simple models or attacks.

- PD's performances are overall significantly better than those of previous SOTA detectors, LID, and Mahalanobis. We have succeeded in constructing a very effective detector. Additionally, structural topological information extracted from induced graphs does contain discriminative information about adversarial examples, regardless of the task complexity. Overall, the success of PD validates our main hypothesis.

The results on the Boundary black-box attack show that our methods (and also the baselines LID and Mahalanobis) do not rely on gradient masking and can generalize well. More experiments on PDs and under-optimized edges are provided in Section 7.7.

7.5.4 Relation between Pruning and Robustness

We have shown that structural information flow in under-optimized edges is different for clean vs adversarial inputs: these edges represent a vulnerability for neural networks. A natural robustification idea would stem from pruning, i.e. exactly removing these under-optimized edges during training. We present a theoretical argument showing how having less active paths, e.g. by pruning, can help robustness. For an input example $x \in \mathcal{X}$, let $\mathcal{P}(x)$ be the set of all weighted paths in the activation graph $G(x, g_\theta)$ of x as defined in Section 7.4.1. Each $\alpha \in \mathcal{P}(x)$ can be identified with a schema $u^0(\alpha) \xrightarrow{w^1(\alpha)} u^1(\alpha) \xrightarrow{w^2(\alpha)} \dots \xrightarrow{w^{L-1}(\alpha)} u^L(\alpha)$, where $u^l(\alpha) \in [\![1, n_l]\!]$ is the index of the neuron through which the path traverses the lth layer of the network, and $w^l(\alpha)$ is the weight of edge weight connecting the former neuron to the next neuron on the path. The subset $\mathcal{A}(x)$ of paths which are active for the input example x is given by $\mathcal{A}(x) := \{\alpha \in \mathcal{P}(x) \mid w^l(\alpha) \neq 0 \; \forall l \in [\![1, L]\!]\}$. Information from input to output only flows along such paths. Finally, let $W(\alpha) :=$

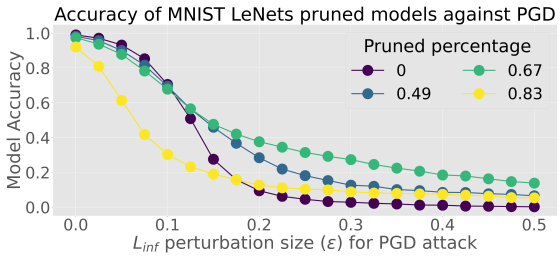


Figure 7.17: Adversarial accuracy of pruned MNIST LeNet models against PGD.

$\Pi_{l=1}^{L}(W_l)_{u^{l-1}(\alpha),u^l(\alpha)}$ be the product of all the parameters of the neural network along the path α. We have the following result:

Proposition 7.5.1. *For every class label $k \in [\![1, K]\!]$ and input feature index $j \in [\![1, n_0]\!]$, we have: $\frac{\partial[g_\theta(x)]_k}{\partial x_j} = \sum_\alpha W(\alpha)$, where the sum runs over all active paths $\alpha \in \mathcal{A}(x)$ such that $u^0(\alpha) = j$ and $u^L(\alpha) = k$, i.e., active paths which start at the j^{th} input neuron and end at the k^{th} output neuron.*

Note that it holds for the ReLU activation.

Proof Let $z_l := g_{\theta,}(x) \in \mathbb{R}^{n_l}$ be the output of the l^{th} layer of the neural network. Note that $z_l = \sigma_l(W_l z_{l-1})$. By the chain rule, we have

$$\frac{\partial[g_\theta(x)]_k}{\partial x_j} = \sum_{k'=1}^{n_{L-1}} \frac{\partial[z_L]_k}{\partial[z_{L-1}]_{k'}} \cdot \frac{\partial[z_{L-1}]_{k'}}{\partial x_j}. \tag{7.5.2}$$

On the other hand, for ReLU activation we have (still via the chain rule)

$$\frac{\partial[z_L]_k}{\partial[z_{L-1}]_{k'}} = [W_l]_{k,k'}\sigma'(W_l z_{l-1}) = [W_l]_{k,k'} \begin{cases} 1, & \text{if } [W_l]_k^\top z_{k'} > 0, \\ 0, & \text{else.} \end{cases}$$

Thus the claim follows directly from Section 7.5.4 by recurring on the depth L. □

Note that the (Frobenius) norm of the jacobian matrix $J(x) = (\frac{\partial g_\theta(x)_k}{\partial x_j})_{j,k}$ is a proxy for the robustness to perturbations on input x, as it is related to the distance to the closest adversarial example for x (see Jakubovitz and Giryes (2018) and Section 7.5.4). Thus, decreasing this sum improves robustness: we could 1) decrease/remove large $W(\alpha)$ (but it would likely hinder the standard accuracy) or 2) reduce the cardinality of $\mathcal{A}(x)$, i.e., have very few active paths: this can be achieved by pruning a neural network and suggests that under-optimized edges may be a problem for robustness because of their quantity.

Illustration.

Some works have focused on the link between adversarial robustness and sparsity (Guo et al., 2018c; Wang et al., 2018a, 2020a) but the conclusion remains unclear. We pruned a MNIST LeNet model (following Frankle and Carbin (2019)'s protocol and our definition of under-optimized edges and ran PGD attacks to measure each model's adversarial accuracy. Figure 7.17 shows that some degree of under-optimized edges pruning might be helpful for adversarial robustness (e.g. 67% seems to be desirable).

About the Jacobian matrix and its relation with robustness.

In Sokolić et al. (2017), authors have shown that the Frobenius norm of the Jacobian matrix is related to the generalization error: regularizing it induces smaller generalization errors. Following this work, Jakubovitz and Giryes (2018) have linked the Jacobian matrix to adversarial robustness. For an input x, the Froebenius norm of the Jacobian matrix at point x is related to the distance to its closest adversarial example (more precisely, their proposition 3 shows it is an upper bound for the L_2-norm of distance to the closest adversary of x): minimizing this norm thus leads to improved robustness.

7.6 Conclusion

Following an in-depth analysis of the characteristics exhibited by adversaries, we have first established a unifying hypothesis, suggesting that adversarial examples leverage under-optimized edges in neural networks in a structured manner. To verify this hypothesis, we have conducted several experiments, among which we have successfully devised a highly efficient detection method named Persistent Diagram (PD) that leverages the inherent structural properties of the under-optimized edges in neural networks. By harnessing the rich topological information that traverses the network, our approach enables the accurate identification of adversarial instances. This success confirms the solidity of our hypothesis and paves the way for a more systematic study of the topology of neural networks from a robustness perspective. Additionally, we have complemented our experimental findings with a theoretical argument that also advocates for reducing the widespread over-parametrization prevalent in neural networks. To fortify models against such vulnerabilities, a potential avenue involves imposing constraints on the network's complexity, for instance, through pruning techniques.

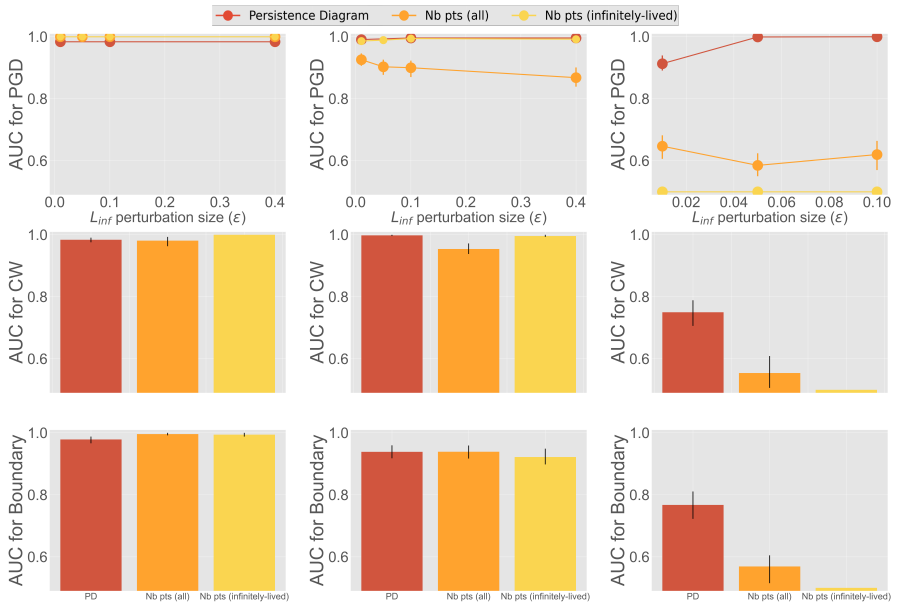
As our work remains mainly experimental, additional investigations are necessary to fully establish our hypothesis as an explanation for the adversarial vulnerability of neural networks. Importantly, even when selecting a relatively few number of under-optimized edges, our method is based on the computation of nested graphs and thus struggles to scale to very large neural network architectures and datasets. For example, additional work on the ImageNet dataset and on larger versions of ResNets, or different models such as Vision Transformers, would be appreciable. To do so, faster algorithms to compute persistent diagrams (or approximation) would be necessary.

Another interesting venue for future extensions is the study of the intricate relationship between pruning strategies, or sparse networks, and the resultant robustness of the models, which has started to be studied, as previously mentioned, but is not yet fully understood.

7.7 Additional Results

To complement the experiments presented in Section 7.5, we conducted additional experiments that shed light on specific aspects of our PD method.

7.7.1 Quantitative Differences in a Simple Setting



(a) LeNeT/MNIST (b) LeNet/F-MNIST (c) ResNet/CIFAR

Figure 7.18: Unsupervised detection results using number of points only.

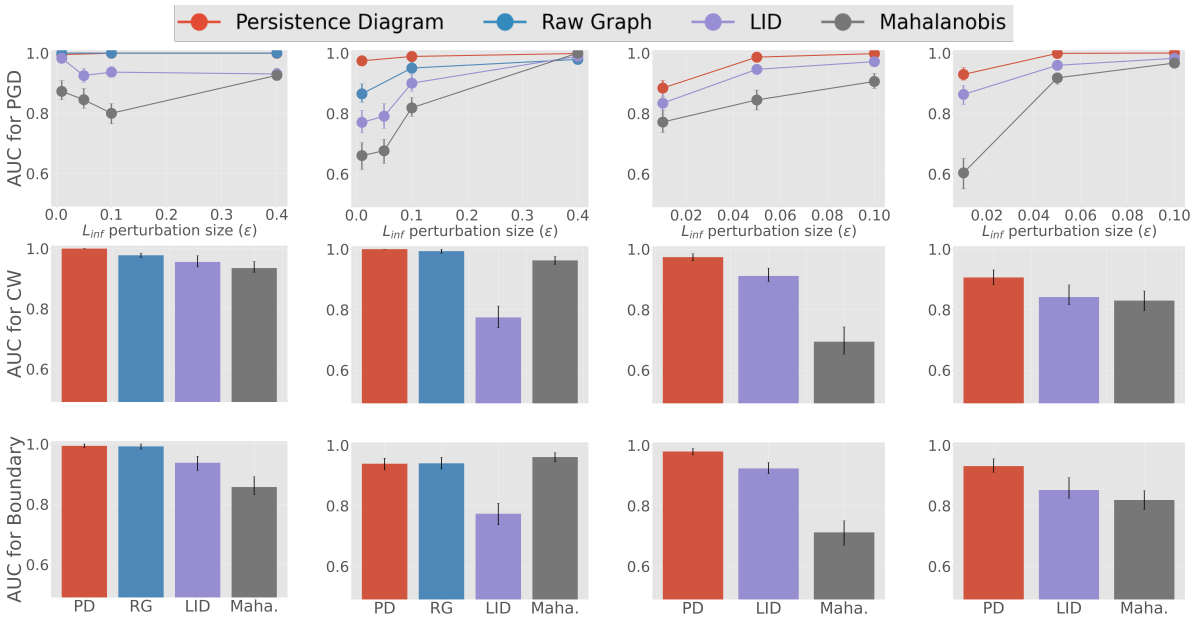
We have shown in Section 7.5.1 that counting the number of points in the persistent diagrams can be an efficient strategy to differentiate adversarial inputs from clean ones. To emphasize these results, we created two very simple detectors based on the number of points in persistence diagrams (one for all points, one for infinitely-lived points) using an SVM with an RBF kernel. The results are shown in Figure 7.18. It illustrates the fact that indeed, the number of points in diagrams provides relevant information, even enough to match our PD method in the two simplest settings. When the task is more difficult, however (in CIFAR10 / ResNet setting), it is not enough to yield as good results as when using directly all information from persistence diagrams, like in our PD method.

7.7.2 Supervised Results

As mentioned before, supervised results can give a false sense of security because, in practice, one cannot anticipate which algorithm will be used to craft an adversarial example (see Figure 7.19): for LID and Mahalanobis, the supervised AUCs are noticeably better than the unsupervised ones, with confidence intervals for these almost not overlapping; on the contrary, PD is more stable between these settings (the difference is around six times smaller). We report results from this unsupervised setting. To compare with the literature (where most of the results are reported under the supervised setting) we also provide supervised results in the Appendix. Keep in mind that great results on supervised

	Sup.	Unsup.	Diff
PD	0.884 [0.858, 0.910]	0.873 [0.851,0.902]	**0.011**
LID	0.835 [0.799, 0.870]	0.776 [0.744, 0.817]	0.059
Maha	0.772 [0.737, 0.811]	0.712 [0.664, 0.748]	0.06

Figure 7.19: Supervised vs unsupervised detection of adversarial examples. Showing AUC for ResNet / SVHN subject to PGD attacks with $\varepsilon = 0.01$. Smaller diff. is better.



(a) LeNeT/MNIST (b) LeNet/F-MNIST (c) ResNet/SVHN (d) ResNet/CIFAR10

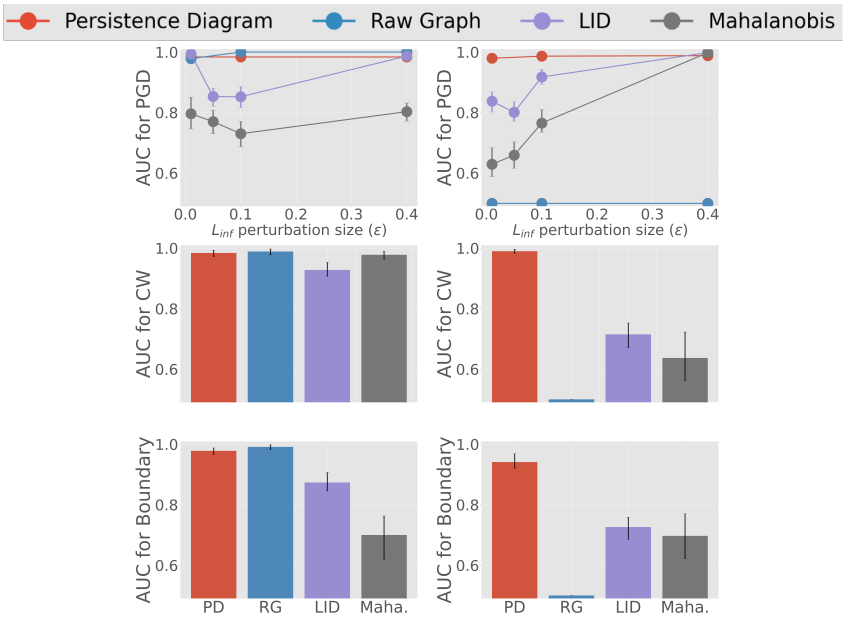
Figure 7.20: Supervised results - Showing detection AUC for different detection methods (legend) against different kinds of adversarial attacks (rows) and model architectures and datasets (columns)

experiments are easier to achieve than on unsupervised experiments because, obviously, the task is harder.

However, results using the *supervised* setting are quite similar to those obtained under the *unsupervised* setting (the AUC are overall higher, because the task is simpler): the hierarchy between the detection methods is identical, with Persistence Diagram providing the best results, followed by LID and Mahalanobis. Note that, as mentioned in the main paper, some AUC results are significantly higher in the supervised setting (Raw Graph, Mahalanobis, etc.), illustrating the false sense of security we can get by studying only supervised results.

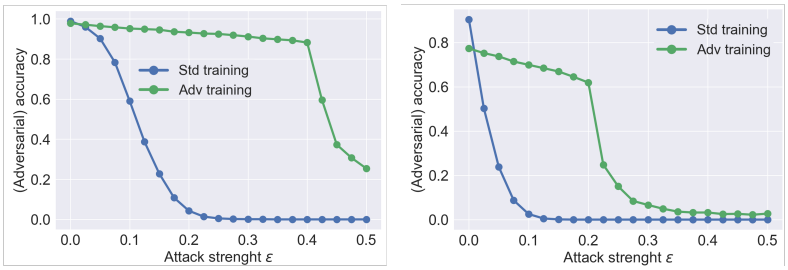
We also ran experiments using transferred attacks on MNIST and Fashion MNIST LeNets, reported in Figure 7.21. Transferred attacks were generated on control models (using the same LeNet architecture), and successful adversaries on these control models were saved. Then, these attacks were submitted to our original target models, and detection methods were launched to flag these adversaries. The results reported here correspond to a black-box setting.

The results are quite similar to those observed for the white-box setting, with our PD



(a) LeNeT/MNIST (b) LeNet/F-MNIST

Figure 7.21: Transferred attacks results - Detection AUC for different detection methods (legend) against different kinds of adversarial attacks (rows) and model architectures and datasets (columns).



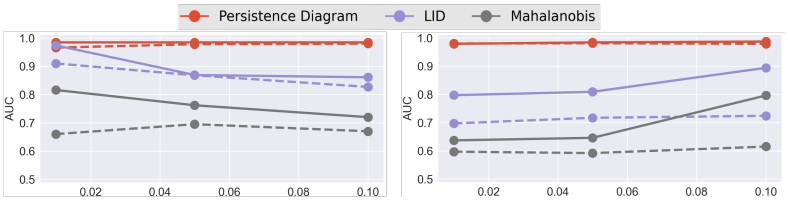
(a) LeNeT/MNIST (b) LeNet/F-MNIST

Figure 7.22: Adversarial accuracy (against PGD) of adversarially trained vs standard neural networks.

method still better than LID and Mahalanobis. As mentioned in Section 7.5.2, the three main methods (PD, LID, Mahalanobis) seem to generalize well in this black-box setting.

We illustrated in the main paper the fact that by being a structural method, PD can generalize to all sorts of adversaries. Successful adversaries on adversarially trained (AT) neural networks are unusual adversaries by nature because they can fool a robust model trained to resist the usual adversaries. As such, running our detection methods on AT neural networks is a good way to check the generalization ability of said methods: if there is no drop in performance compared to the classical setting, then the method is highly generalizable; if there is one, maybe the method was built on too strong assumptions about adversaries that are not satisfied by all of them.

Figure 7.22 shows the standard and adversarial accuracy against PGD of the AT neural networks compared to the standard ones. Figure 7.23 shows the detection results' discrepancies between standard and AT neural networks using PGD attacks, for all methods.



(a) LeNeT/MNIST (b) LeNet/F-MNIST

Figure 7.23: Unsupervised detection results (on PGD) of adversarially trained (dashed lines) vs standard neural networks (full lines)

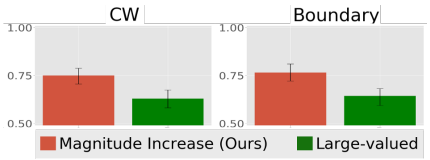


Figure 7.24: Impact of edge-selection methods on AUC (ResNet / CIFAR10).

Our PD method outputs almost no performance gap, contrary to LID and Mahalanobis, meaning that our method is more general and that all types of adversaries do leverage under-optimized edges.

7.7.3 Informative Power of Under-optimized Edges

We provide here an experimental illustration of the impact of edge-selection by comparing the use of under-optimized edges to detect adversarial inputs with our PD method, instead of "well-optimized" edges. The results shown in Figure 7.24 indicate that the detection AUC is better when using under-optimized edges vs well-optimized ones, which also supports our hypothesis stating that these edges contain more information about adversaries.

Chapter 8

Existence of Low-Dimensional Adversarial Attacks

Fool me once, fool me twice.

Billie Eilish.

Contents

8.1 Introduction and High-level Overview

In Chapter 7, we introduced a hypothesis explaining the presence of successful adversarial examples in deep learning models whatever the type of adversarial examples, meaning that on-manifold and off-manifold adversaries were both considered. Our hypothesis was backed with experiments showing that, based on our methodology, an efficient detector can be built to flag such adversarial examples.

Our experiments concentrated on classical adversarial attacks, meaning attacks that target the full-dimensional space of the input features, even though the perturbations are controlled in size by a L_2 or L_∞ norm. However, recent advances in the search for more practical adversarial attacks have changed this paradigm with the creation of adversarial perturbations that can be found by black-box search using surprisingly few queries, which essentially restricts the perturbation to a subspace of dimension k, much smaller than the dimension d of the image space. More precisely, methods such as Boundary Attack, introduced in Brendel et al. (2017), NES in Ilyas et al. (2018), SimBA in Guo et al. (2019) and HopSkipJump in Chen et al. (2020a) approximate the full gradient of the model's loss via a Monte-Carlo finite-difference estimate which sub-samples the coordinates randomly. Surprisingly, existing black-box attacks can be carried out using a very small number of queries, which suggests that adversarial examples are abundant in low-dimensional subspaces. This intuition is confirmed by subsequent works that also performed adversarial search in a *fixed* subspace such as the low-frequency subspace, as in Yin et al. (2019); Guo et al. (2018a), or by selecting the subspace in a distribution-dependent manner using an independently-trained neural network, as in Tu et al. (2019); Yan et al. (2019); Huang and Zhang (2019).

These empirical findings lead us to hypothesize that adversarial perturbations exist with high probability in low-dimensional subspaces, which raises the question: *Is the vulnerability to black-box, low-dimensional attacks inherent or can we hope to prevent them?* As previously mentioned in Section 6.2.3, even though several works have tackled this questions for more generic types of attacks (meaning full-dimensional attacks), such theoretical results cannot apply directly to low-dimensional types of attacks, as the principle of *curse of dimensionality* cannot be used.

In this Chapter, we initiate a rigorous study of the phenomenon of low-dimensional adversarial perturbations (LDAPs). Our result characterizes precisely the sufficient conditions for the existence of LDAPs, and we show that these conditions hold for neural networks under practical settings, including the so-called lazy regime wherein the parameters of the trained network remain close to their values at initialization. We thus provide rigorous explanations for the empirical success of some powerful heuristics that have appeared in the literature, such as Moosavi-Dezfooli et al. (2017); Khrulkov and Oseledets (2018); Guo et al. (2018a); Yin et al. (2019); Chen et al. (2020a). In addition to this theoretical contribution, our results are confirmed by experiments on both synthetic and real data.

8.1.1 Outline of the Rationales of the Chapter

Our theoretical analysis of the low-dimensional adversarial perturbations is mainly based on the smoothness of the classifier and on geometrical properties of the attack subspace V. More precisely, we derive bounds that reveal the role of:

- the **local smoothness** of the classifier's decision-boundary,
- the **alignment** of the subspace V with the unit-normals at the classifier's decision-boundary,
- the distribution of classifier's **pointwise margin**,
- the **attacker's budget** ε (measured in Euclidean norm).

We formalize a notion of alignment in Section 8.3.

For random subspaces of sufficiently high dimension (Guo et al., 2019) and subspaces obtained via SVD on the gradients (Moosavi-Dezfooli et al., 2017; Khrulkov and Oseledets, 2018), our results provide transparent lower-bounds on the fooling rate, which explain the empirical success of the very efficient heuristic methods that have been proposed in the literature for constructing LDAPs; see Section 8.6.1. Moreover, the lower-bounds only depend on the distributions of the predictions and the gradients of the model and so can be empirically estimated on held-out data, making them a practical predictor for the adversarial vulnerability of classifiers. Our theoretical results are confirmed by numerous experiments on real and simulated data (Section 8.6). In all cases, the bounds can be easily evaluated and are close to the actual fooling rates.

8.1.2 Literature Overview

Earlier experiments, as in Moosavi-Dezfooli et al. (2017); Khrulkov and Oseledets (2018), showed that adversarial attacks based on a single direction of feature space, called Universal Adversarial Perturbations (UAPs) can be designed to effectively fool neural networks. UAPs are often more transferable across datasets and architectures than classical attacks, making them interesting for use in practice. Their theoretical analysis has been initiated in Moosavi-Dezfooli et al. (2018), where the authors established lower bounds for the fooling rate of UAPs under certain curvature conditions on the decision boundary. The aforementioned work has two fundamental limitations. First, the notions of curvature used are stated in terms of unconstrained optimal adversarial perturbation (*i.e.* the closest point) for an arbitrary input point and thus are not easy to verify in practice. Also, the existence of the UAP is only guaranteed within a subspace which is required to satisfy a global alignment property with the gradients of the model. In contrast, we use a more flexible curvature requirement (refer to Definition 8.3.1), which is adapted to any subspace under consideration, and we prove results that are strong enough to provide a satisfactory theory of LDAPs, and UAPs in particular, under very general settings.

Guo (2020) studied LDAPs when the attacker is constrained to a uniformly random k-dimensional subspace. For classifiers whose decision regions are half-spaces and spheres in $\mathbb{R}^d$, they established the existence of low-dimensional adversarial subspaces under a Gaussian concentration assumption on the data. Our work considers more general decision regions (e.g. of certain neural networks) and more general data distributions and subspaces. Our results recover the findings of Guo (2020) as special cases.

8.1.3 Outline of the Main Contributions of the Chapter

Classical theoretical works on understanding adversarial examples, like Tsipras et al. (2019); Shafahi et al. (2019a); Mahloujifar et al. (2019); Gilmer et al. (2018); Dohmatob (2019), focus on the case of adversarial attacks on the full feature space. They use the concentration property of certain high-dimensional (*e.g.* multivariate Gaussians, distributions satisfying log-Sobolev inequalities, etc.), to establish that an imperfect classifier will admit adversarial examples. However, such techniques cannot be used directly when we add the constraint that the attacks only live in a low-dimensional subspace. Thus, new techniques are needed. Such techniques were initiated in Guo (2020) for the case of linear models, and are extended in our paper to non-linear models.

More precisely, our contributions are the following:

- In Section 8.3, we formalize the notion of *adversarially viable subspace*, which provides a characterization of the low-dimensional subspaces that can be relevant to conduct adversarial attacks. More precisely, this notion provides an alignment condition between the attack subspace and the gradient of the model for the attack subspace to be usable in practice to craft successful low-dimensional adversaries.
- In Section 8.4, we provide our theoretical bounds for models that have a Lipschitz decision boundary. This smoothness characteristic allows us to provide general results on the efficiency of LDAPs, which is also illustrated in cases when the model is linear or hyper-ellipsoidal for example.
- In Section 8.5, we also provide our theoretical bounds for models with locally almost affine decision boundaries. This smoothness characteristic allows us to provide similar results for practical, state-of-the-art models, for example, neural networks with ReLU activation functions in the random feature regime or the lazy regime.
- In Section 8.6, we conduct experiments to illustrate the informative power of our theoretical bounds for generic, trained neural networks. Our bounds are shown to hold in this case, even when the neural network is large or adversarially trained.

8.2 Preliminaries

In this Section, we clarify the context of this theoretical study, which is binary classification. We also recall some notions related to adversarial examples.

8.2.1 Binary Classification and Adversarial Examples

We consider a binary classification setup, where $X = (X_1, ..., X_d) \in \mathbb{R}^d$ denotes an input of dimension d (e.g. for the MNIST dataset, $d = 784$) drawn from a probability distribution P_X on $\mathbb{R}^d$. We will denote by $g_\theta : \mathbb{R}^d \to \mathbb{R}$ generic feature map with parameters θ, and $f_\theta = \text{sign} \circ g_\theta$ the corresponding classifier, with the arbitrary convention that $\text{sign}(0) = -1$.

For example, for neural networks, $g_\theta(x)$ would be the predicted *logit* for input x; for a closed ball of radius $r > 0$ in $\mathbb{R}_d$, $g_r(x) := (\|x\|^2 - r^2)/2$; and for a half-space (linear classifier), $g_\theta(x) := x^\top w - b$ with $\theta = (w, b)$.

The binary classifier f_θ can be unambiguously identified with a measurable subset of $\mathbb{R}^d$, formally defined as follows:

Definition 8.2.1. NEGATIVE DECISION REGION. *Let $\mathcal{X} = \mathbb{R}^d$ be the input feature space, $g_\theta : \mathbb{R}^d \to \mathbb{R}$ be a feature map, and $f_\theta = sign \circ g_\theta$ its corresponding classifier. The negative decision region of f_θ is defined by:*

$$C = \{x \in \mathbb{R}^d \mid f_\theta(x) = -1\} = \{x \in \mathbb{R}^d \mid g_\theta(x) \leq 0\} \tag{8.2.1}$$

and its complement, $C' := \mathbb{R}^d \setminus C$, is the positive decision region

Of course, the terms 'negative' or 'positive' are interchangeable, as we can always consider the classifier $-h$ instead. Therefore, without loss of generality, we shall focus our attention on adversarial attacks on the positive decision region C'.

Given an input $x \in C'$ classified by f_θ as positive, an adversarial perturbation for x is a vector $\delta \in \mathbb{R}^d$ of size $\|\delta\|_2$ such that $x + \delta \in C$. The goal of the attacker is to move points from C' to C with small perturbations. Note that we are not interested in the true labels of the inputs, just the robustness of the classifier with respect to its own predictions. However, note that this distinction is not important for classifiers which are already very accurate in the classical sense.

The notion of *margin* will be important in the sequel.

Definition 8.2.2. MARGIN AT A POINT. *Let $\mathcal{X} = \mathbb{R}^d$ be the input feature space and $g_\theta : \mathbb{R}^d \to \mathbb{R}$ be a feature map. Let $x \in \mathbb{R}^d$ be an input.*

If g_θ is differentiable at x and x is non-critical, the the margin of g_θ at x, denoted $m_{g_\theta}(x)$, is defined by:

$$m_{g_\theta}(x) := (g_\theta(x))_+ / \|\nabla g_\theta(x)\|_2 \tag{8.2.2}$$

For example, if $g_\theta(x) \equiv x^\top w - b$ for some scalar $b \in \mathbb{R}$ non-zero and $w \in \mathbb{R}^d$, as in the case where the classifier is a half-space, then $m_{g_\theta}(x) = (x^\top w - b)_+ / \|w\|_2$. In this case, $m_{g_\theta}(x)$ also corresponds to the distance of x from the negative decision region of the classifier.

8.2.2 Low Dimensional Adversarial Perturbations

In this paper, we focus on *low-dimensional* perturbations (LDAPs), as in Guo et al. (2018a, 2019); Tu et al. (2019); Yan et al. (2019); Huang and Zhang (2019); Guo (2020), meaning that the perturbations δ are limited to a k-dimensional subspace V of $\mathbb{R}^d$ whose choice is left to the attacker. Here, k can be much smaller than d. The special case where $k = 1$ corresponds to the scenario where the attacker is allowed to operate in one dimension only (e.g. modify the same pixel in all images of the same class), also famously known as *universal adversarial perturbations* (UAPs), as studied in Moosavi-Dezfooli et al. (2017); Khrulkov and Oseledets (2018).

Definition 8.2.3. ATTACKABLE REGION. *Let $\mathcal{X} = \mathbb{R}^d$ be the input feature space, C the negative decision region, $V \subseteq \mathbb{R}^d$ be a subspace, and $\varepsilon > 0$.*

The set of all points in $\mathbb{R}^d$ which can be pushed into the negative decision-region C by adding a perturbation of size ε in V is defined by:

$$C_V^\varepsilon := \{x \in \mathbb{R}^d \mid \exists v \in V \text{ with } \|v\|_2 \leq \varepsilon \text{ s.t } x + v \in C\}, \tag{8.2.3}$$

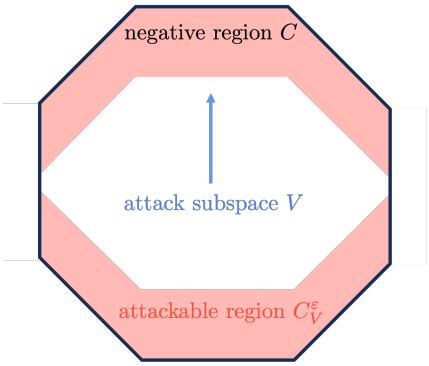


Figure 8.1: Illustration of an attackable region. Considering the negative decision region C is an octagon, and following the blue attack subspace V, the corresponding attackable region C_V^ε corresponds to the red area.

where $B_V := V \cap B_d$ *is the unit-ball in* V.

The concept of attackable region is illustrated in Figure 8.1. Note that by definition, $x \in C_V^\varepsilon$ if and only if $(x + \varepsilon B_V) \cap C \neq \emptyset$. In the particular case of full-dimensional attacks where $V = \mathbb{R}^d$, the set C_V^ε corresponds to the usual ε-expansion C^ε of C, i.e., the set of points in $\mathbb{R}^d$ which are at a distance at most ε from C. This case has been extensively studied in Shafahi et al. (2019a); Fawzi et al. (2018a); Mahloujifar et al. (2019); Dohmatob (2019). Also, note that it always holds that $C \subseteq C_V^\varepsilon \subseteq C^\varepsilon$.

Definition 8.2.4. FOOLING RATE OF A SUBSPACE. *Given an attack budget* $\varepsilon \geq 0$*, the fooling rate* $\mathrm{FR}(V;\varepsilon)$ *of a subspace* $V \subseteq \mathbb{R}^d$ *is the proportion of test data which can be moved from the positive decision-region* C' *to the negative decision-region* C *by moving a distance* ε *along* V*, that is*

$$\mathrm{FR}(V;\varepsilon) := P_X(X \in C_V^\varepsilon \mid X \in C'). \tag{8.2.4}$$

Note that by definition of C_V^ε, the fooling rate $\mathrm{FR}(V;\varepsilon)$ is a supremum over all possible attackers operating in the subspace V, and with L_2-norm budget ε. In particular, $\mathrm{FR}(\mathbb{R}^d;\varepsilon)$ is the usual optimal fooling rate of an adversarial attack with budget ε, without any subspace constraint, and already studied extensively in the literature, for example in Shafahi et al. (2019a); Fawzi et al. (2018a); Mahloujifar et al. (2019); Dohmatob (2019).

8.2.3 Illustration with a linear model

We start with the simple case of a linear binary classifier on $\mathbb{R}^d$, for which the negative decision-region C (and therefore the positive decision region too) is a half-space given by

$$H_{w,b} := \{x \in \mathbb{R}^d \mid x^\top w - b \leq 0\}, \tag{8.2.5}$$

with unit-normal vector $w \in \mathbb{R}^d$ and bias parameter $b \in \mathbb{R}$. This corresponds to taking $f(x) := x^\top w - b$ in Definition 8.2.1. The following result generalizes a result of Guo (2020) (see Lemma 2.2 therein) which was only established in the case where the marginal distribution of the features P_X is the standard Gaussian distribution on $\mathbb{R}^d$.

Proposition 8.2.5. *Let* C *be the half-space* $H_{w,b}$ *defined in Equation (8.2.5).*

For any subspace V of $\mathbb{R}^d$ and $\varepsilon \geq 0$, it holds $\mathrm{FR}(V;\varepsilon) \geq \mathbb{P}_X(X^\top w - b \leq \|\Pi_V w\|\varepsilon \mid X \in C')$.

In particular, if V is a uniformly random k-dimensional subspace of $\mathbb{R}^d$, then for any $t \in (0, \sqrt{k/d})$ it holds with probability $1 - 2e^{-t^2 d/2}$ over V that:

$$\mathrm{FR}(V;\varepsilon) \geq \mathbb{P}_X(X^\top w - b \leq (\sqrt{k/d} - t)\varepsilon \mid X \in C'). \tag{8.2.6}$$

Interpretation of Proposition 8.2.5. To understand the power of the the above proposition, consider the cause where $P_X = \mathcal{N}(0, I_d)$ and $b = 0$ so that $P_X(C) = P_X(C') = 1/2$. Note that a typical $x \sim P_X$ has a norm of order $\mathbb{E}[\|x\|_2] \asymp \sqrt{d}$. Thus a random perturbation of dimension $k = \sqrt{d} \ll d$ and of L_2-norm $\varepsilon = \sqrt{d/k} = d^{1/4} \ll \mathbb{E}[\|x\|_2]$ is sufficient to change the decision of the classifier on a proportion

$$\mathrm{FR}(V;\epsilon) \geq P_X(X^\top w \leq 1 \mid X^\top w \geq 0) = (\Phi(1) - \Phi(0))/(1/2) \approx 68\% \tag{8.2.7}$$

from negative to positive.

Proof of Proposition 8.2.5.

Indeed, one computes

$$\begin{aligned}
\mathrm{FR}(V;\varepsilon) &:= P_X(X \in C_V^\varepsilon \mid X \in C') && (8.2.8)\\
&\geq \sup_{v \in V} P_X(X \in C_v^\varepsilon \mid X \in C') && (8.2.9)\\
&= \sup_{v \in V \cap \mathcal{S}_{d-1}} P_X(X^\top w + \varepsilon v^\top w - b \leq 0 \mid X \in C') && (8.2.10)\\
&= P_X(X^\top w - b \leq \varepsilon \|\Pi_V w\|_2 \mid X \in C'), && (8.2.11)
\end{aligned}$$

which proves the first part of the claim. The second part follows from the first part combined with the fact that

$$\|\Pi_V w\|_2 \geq \sqrt{k/d} - t \text{ with proba. } 1 - 2e^{-t^2 d/2}, \tag{8.2.12}$$

by basic concentration arguments. □

Lifting the Core Ideas to the Non-Linear Setting. In the results of this Section, we will emulate the lower-bound from Proposition 8.2.5, for the case of non-linear classifiers. In this direction, first observe that, since the margin for the linear classifier is $m_{g_\theta}(x) := \max(g_\theta(x), 0)/\|\nabla f(x)\|_2 = (x^\top w + b)_+$, the lower-bound from Proposition 8.2.5 can be written in expectation-form as

$$\mathbb{E}_V[\mathrm{FR}(V;\varepsilon)] \geq P(m_{g_\theta}(x) \leq \alpha\varepsilon \mid X \in C') - \beta, \tag{8.2.13}$$

with $\alpha = \sqrt{k/d} - t$ and $\beta = 2e^{-t^2 d/2}$. The pair of scalars (α, β) capture the alignment between the random subspace V, and the gradients of the linear classifier at a random

point $X \in C'$, i.e with the normal vector $\eta(X) = \nabla g_\theta(x)/\|\nabla g_\theta(X)\|_2 = w$, in the sense that

$$P_{X,V}(\|\Pi_V \eta(X)\|_2 \geq \alpha \mid X \in C') \geq 1 - \beta. \tag{8.2.14}$$

Since $\eta(X) = w$ here, and is independent of the feature vector X, Equation (8.2.14) is just a restatement of Equation (8.2.12). In the general case of non-linear models g_θ (e.g neural nets) and arbitrary subspaces V, inequalities such as Equation (8.2.14) will be the basis of so-called adversarially viable subspaces, studied in detail in Section 8.3.

8.3 Adversarially Viable Subspaces

8.3.1 Definition of Adversarially Viable Subspace

We will formalize the notion of an *adversarially viable* subspace which is a subspace V that has a non-negligible inner product with the classifier's gradient, hence it is possible to significantly alter the value of $f(x)$ by moving strictly within V. Intriguingly, such subspaces are pivotal to the empirical success of LDAPs, and we show that popular heuristics lead to adversarially viable subspaces.

Then, we prove that when the classifier satisfies certain smoothness conditions, adversarially viable subspaces allow the attacker to follow the gradient direction within V to reach the decision boundary of C for most points $x \in C'$, hence achieving a high fooling rate.

Restricting the adversarial perturbation to a given subspace V presents a particular challenge to the attacker. If $\dim(V) < d$ and $x \in C' := \mathbb{R}^d \setminus C \neq \emptyset$, it is possible that $x \notin C_V^\varepsilon$ for all $\varepsilon > 0$. In particular, if f is convex and the subspace V is orthogonal to the gradient of f at a point $x \in \mathbb{R}^d$, then no amount of perturbation within V will make x closer to the boundary of C, in an effort to flip its predicted class label.

This intuition is illustrated by Figure 8.2. In this case, it is possible to perturb the input x such that its adversarial counterpart crosses the decision boundary when adding a perturbation from V_1, but it is not the case if the perturbation is in V_2. Thus, we can hope to establish nontrivial fooling rates only for certain subspaces.

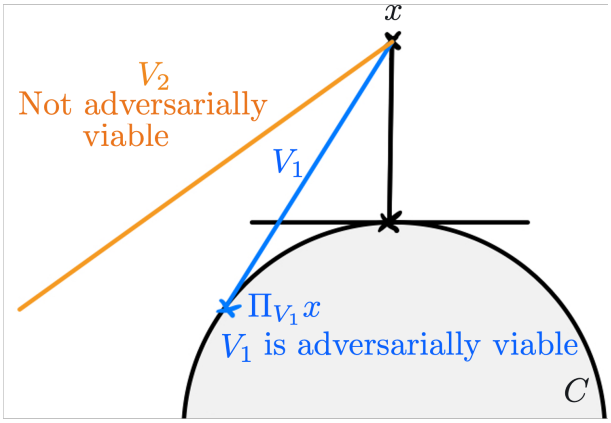


Figure 8.2: Illustration of adversarial viability: a perturbation from V_1 can push x^{adv} on the other side of the decision boundary than x, but it is not the case for V_2.

Our first contribution is a crisp characterization of subspaces for which we can hope to

achieve a nonzero fooling rate. These are so-called *adversarially viable* subspaces and are a generalization of the subspaces considered in Moosavi-Dezfooli et al. (2018); Moosavi-Dezfooli et al. (2017); Guo (2020).

Definition 8.3.1. ADVERSARIALLY VIABLE SUBSPACE. *Let $\alpha \in (0, 1]$ and $\beta \in [0, 1)$, and let $V \subseteq \mathbb{R}^d$ be a possibly random subspace. V is said to be adversarially (α, β)-viable if:*

$$P_{X,V}(\|\Pi_V \eta(X)\|_2 \geq \alpha \mid X \in C') \geq 1 - \beta, \tag{8.3.1}$$

where $\eta(x) := \nabla g_\theta(x)/\|\nabla g_\theta(x)\|_2$ is the gradient direction at x.

The above definition captures the essence of Equation (8.2.12), which was the crucial piece in the proof of Proposition 8.2.5. To see that this is a generalization of Equation (8.2.12), note that $\eta(x) \equiv w$ when C is a half-space (i.e when g_θ is a linear function $g_\theta(x) \equiv x^\top w - b$).

We now provide some important examples of adversarially viable subspaces.

8.3.2 Random Subspaces

Consider the case of a uniformly random k-dimensional subspace V of $\mathbb{R}^d$. Such subspaces have been proposed in the literature, see for example Moosavi-Dezfooli et al. (2017); Guo (2020), for constructing low-dimensional adversarial perturbations.

Lemma 8.3.2. *The random subspace as given in Proposition 8.2.5 is $(\sqrt{k/d} - t, 2e^{-t^2 d/2})$-viable for any $t \in (0, \sqrt{k/d})$.*

Indeed, this is just a restatement of Equation (8.2.12), in the language of Definition 8.3.1.

8.3.3 Eigen-subspace

Let $\Sigma_\eta \in \mathbb{R}^{d \times d}$ be the covariance matrix of the gradient direction $\eta(X)$ conditioned on $X \in C'$.

Theorem 8.3.3. *For any $k \in [d]$, let $s_k \in (0, 1]$ be the sum of first the k eigenvalues of Σ_η. Then, for any $\alpha \in (0, \sqrt{s_k})$, the (deterministic) subspace $V_{\text{eigen},k}$ of $\mathbb{R}^d$ corresponding to the top k eigendirections of Σ_η is adversarially $(\alpha, (1 - s_k)/(1 - \alpha^2))$-viable.*

Thus, if the histogram of eigenvalues of Σ_η is 'spiked' in the sense that $s_k \geq s = \Omega(1)$ for some $k = o(d)$, then $V_{\text{eigen},k}$ is a $o(d)$-dimensional adversarially $(\Omega(1), O(1 - s))$-viable subspace. Combined with the results established in the following Sections, the preceding observation provides a rigorous justification for the heuristic in Moosavi-Dezfooli et al. (2017); Khrulkov and Oseledets (2018) which proposed UAPs based on eigenvectors of the covariance matrix Σ_η. Our experiments in Section 8.6 also support this.

Proof of Theorem 8.3.3

Let $\Sigma_\eta = USU^\top$ be the singular value decomposition (SVD) of Σ_η, where S is a diagonal matrix containing the nonzero eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$ of Σ_η, $r \in [d]$ is the rank of Σ_η, and U is a $d \times r$ matrix with orthonormal columns. Then, the orthogonal projector for the subspace $V := V_{\text{eigen},k}$ is given explicitly by $\Pi_V = U_{\leq k} U_{\leq k}^\top$, where $U_{\leq k}$

is the $d \times \min(k, r)$ orthogonal matrix corresponding to the first $\min(k, r)$ columns of U. Consider the r.v $Z := \|\Pi_V \eta(X)\|_2$. By a standard formula for the expectation of a quadratic form, one computes

$$\mathbb{E}[Z^2 \mid X \in C'] = \mathbb{E}[\eta(X)^\top \Pi_V \eta(X) \mid X \in C'] \tag{8.3.2}$$

$$= \mathrm{tr}(\Pi_V \Sigma_\eta) = \mathrm{tr}(U_{\leq k} U_{\leq k}^\top \Sigma_\eta) \tag{8.3.3}$$

$$= \mathrm{tr}(U_{\leq k}^\top \Sigma_\eta U_{\leq k}) = \sum_{i=1}^{\min(k,r)} \lambda_i \tag{8.3.4}$$

$$:= s_k. \tag{8.3.5}$$

On the other hand, conditioned on $X \in C'$ we have $0 \leq Z \leq \|\eta(X)\|_2$. Thus, for any $\alpha \in (0, \sqrt{s_k})$, we have

$$X \in C' \implies \mathbb{1}(Z \geq \alpha) \geq \frac{Z^2 - \alpha^2}{1 - \alpha^2}, \tag{8.3.6}$$

with equality on the event $Z^2 \in \{\alpha^2, 1\}$. The claim then follows upon taking expectations on both sides of the above inequality conditioned on the event $X \in C'$. □

8.4 Model with Lipschitz Decision Boundary

Consider a binary classifier on $\mathbb{R}^d$ for which the negative decision-region C of the classifier is given by Definition 8.2.1, where $g_\theta : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function. Let us start by observing that, thanks to a classical result from optimization theory (see Proposition 3.2 of Azé and Corvellec (2017)), if the following condition is satisfied, then any $x \in C'$ is at a distance $d_C(x)$ at most $f(x)/\gamma$ from C.

Condition 8.4.1. Uniformly Strong Gradients. *The feature map g_θ is said to have uniformly strong gradients if there exists a constant $\gamma > 0$ such that $\|\nabla g_\theta(x)\|_2 \geq \gamma$ for all $x \in C'$ '*

Intuitively, under Condition 8.4.1, the gradient of g_θ at any point $x \in C'$ is strong enough: gradient-flow started at x then escapes the region C' after traveling a distance $O(g_\theta(x))$. This is formalized in the following result which will be extended to the case of subspace attacks in the rest of this Section.

Theorem 8.4.2. *If a feature map g_θ satisfies Condition 8.4.1, it holds for any $\varepsilon \geq 0$ that the fooling rate of full-dimensional attacks is lower-bounded as follows*

$$\mathrm{FR}(\mathbb{R}^d; \varepsilon) \geq P_X(g_\theta(X) \leq \gamma\varepsilon \mid X \in C'). \tag{8.4.1}$$

As an illustration, if we consider g_θ to be a randomly initialized [1] finite-depth ReLU neural-network, one can show, as in Daniely and Shacham (2020); Bubeck et al. (2021); Bartlett et al. (2021), that with high probability over the weights: $g_\theta(x) = \mathcal{O}(\|x\|_2/\sqrt{d})$ and $\|\nabla g_\theta(x)\|_2 = \Omega(1)$ for all $x \in \mathbb{R}^d$. The above theorem immediately predicts the existence of adversarial examples of size $\sqrt{d}$ times smaller than the typical L_2-norm of a data point.

[1] With layer widths within $\mathrm{poly}(\log d)$ factors of one another, and weights initialized in the standard way.

8.4.1 Main result on the Lower Bound

We will extend Theorem 8.4.2 to the case of subspace attacks, under the following smoothness condition.

Condition 8.4.3. LIPSCHITZ GRADIENTS. *The feature map g_θ is said to have Lipschitz gradients if there exists a constant $L \geq 0$ such that*

$$\|\nabla g_\theta(x') - \nabla g_\theta(x)\| \leq L\|x' - x\|_2, \text{ for all } x, x' \in \mathbb{R}^d. \tag{8.4.2}$$

This condition stipulates that the gradient of g_θ varies smoothly on the positive decision-region $C' = \mathbb{R}^d \backslash C$ of the classifier Definition 8.2.1. Note that when g_θ is twice-differentiable on C', Condition 8.4.3 holds with $L = \sup_{x \in C'} \|\nabla^2 g_\theta(x)\|_{op}$, where $\nabla^2 f(x) \in \mathbb{R}^{d \times d}$ is the Hessian of g_θ at x. For example, a feed-forward neural net with bounded weights and twice-differentiable activation function with bounded Hessian (e.g. sigmoid, quadratic, tanh, GELU, cos, sin, etc.) will satisfy Condition 8.4.3.

To obtain simplified and more transparent lower bounds for the fooling rate of adversarial subspaces, we will also need the following natural condition which ensures that there is a strong descent direction at a constant fraction of points in the positive decision region C', to allow for gradient-based attacks.

Condition 8.4.4. STRONG GRADIENTS. *The feature map g_θ is said to have strong gradients if there are some constants $\gamma > 0$ and $\lambda \in [0, 1)$ such that $P_X(\|\nabla g_\theta(X)\|_2 \geq \gamma \mid X \in C') \geq 1 - \lambda$.*

Note that Condition 8.4.1 is a special case of Condition 8.4.4 corresponding to $\lambda = 0$.

Now, the following theorem is one of our main results. It generalizes both Proposition 8.2.5 and Theorem 8.4.2.

Theorem 8.4.5. *Let g_θ be a feature map that satisfies the Lipschitz gradient condition from Condition 8.4.3, and let V be a possibly random adversarially (α, β)-viable subspace of $\mathbb{R}^d$. Then,*

(A) For any $\varepsilon \geq 0$, the average fooling rate of V is lower-bounded as follows:

$$\mathbb{E}_V[\mathrm{FR}(V; \varepsilon)] \geq P_X\left(m_{g_\theta}(X) \leq \min\left(\alpha\varepsilon/2, \alpha^2\|\nabla g_\theta(X)\|_2/(2L)\right) \mid X \in C'\right) - \beta. \tag{8.4.3}$$

(B) If in addition, g_θ satisfies the strong gradient condition from Condition 8.4.4, then for any $0 \leq \varepsilon \leq \alpha\gamma/L$,

$$\mathbb{E}_V[\mathrm{FR}(V; \varepsilon)] \geq P_X\left(m_{g_\theta}(X) \leq \alpha\varepsilon/2 \mid X \in C'\right) - \beta - \lambda. \tag{8.4.4}$$

Remark 8.4.6. *Note that the condition '$0 \leq \varepsilon \leq \alpha\gamma/L$' in part (B) of the theorem cannot be removed in general, as is seen in the case where $C = B_d$, and considering any subspace V with $\dim(V) < d$.*

8.4.2 Proof of the Main Result

We first give a vivid sketch of the proof before digging into it with more details.

Proof Sketch. of Equation (8.4.3).

It is an elementary fact in optimization theory that a function $g_\theta : \mathbb{R}^d \to \mathbb{R}$ which has the structure stated in Condition 8.4.3 admits the following first-order approximation: for all $x, x' \in \mathbb{R}^d$,

$$|g_\theta(x') - g_\theta(x) - \nabla g_\theta(x)^\top (x' - x)| \leq \frac{L}{2}\|x' - x\|_2^2. \tag{8.4.5}$$

Now, starting at a point $x \in C'$, let us move a distance ε in the direction $\Pi_V \nabla g_\theta(x)$ to arrive at a point $x' = x - \varepsilon \Pi_V \nabla g_\theta(x) \in \mathbb{R}^d$, the above inequality gives the quadratic approximation:

$$g_\theta(x') \leq g_\theta(x) - \varepsilon \|\Pi_V \nabla g_\theta(x)\|_2^2 + \frac{L}{2}\varepsilon^2 \|\Pi_V \nabla g_\theta(x)\|^2. \tag{8.4.6}$$

After some calculations, the right-hand side of Equation (8.4.6) can be made ≤ 0 by guaranteeing that

(1) **Alignment:** $\|\Pi_V \nabla g_\theta(x)\|_2 \geq \alpha \|\nabla g_\theta(x)\|_2$.

(2) **Small Margin:** $m_{g_\theta}(x) \leq \min(\alpha\varepsilon/2, \frac{\alpha^2 \|\nabla g_\theta(x)\|_2}{2L})$.

The requirement (1) holds because the subspace V is assumed to be (α, β)-viable (see Definition 8.3.1). (2) is obtained from (1) and a careful analysis of Equation (8.4.6). In particular, if $0 \leq \varepsilon \leq \alpha\gamma/L$, then conditioned on $\|\nabla g_\theta(x)\|_2 \geq \gamma$ the 'small margin' condition reduces to: $m_{g_\theta}(x) \leq \alpha\epsilon/2$. □

Let's now provide the full proof. To do so, we will need some auxiliary lemmas that are stated and proved below.

Lemma 8.4.7. Auxiliary lemma (1). *For any $\rho, r > 0$ and $b \in \mathbb{R}^d$, we have the identity*

$$\sup_{z \in \rho B_n} b^\top z - \frac{1}{2r}\|z\|_2^2 = \begin{cases} r\|b\|_2^2/2, & \text{if } \|b\|_2 \leq \rho/r, \\ \rho\|b\|_2 - \rho^2/(2r), & \text{otherwise.} \end{cases} \tag{8.4.7}$$

Proof of Lemma 8.4.7.

Since the quadratic function $z \mapsto (1/2)\|z\|^2$ is unchanged upon taking the *Fenchel-Legendre transform*, we have

$$\sup_{z \in \rho B_d} b^\top z - \frac{1}{2r}\|z\|_2^2 = \sup_{\|z\|_2 \leq \rho} b^\top z - \frac{1}{r}\left(\sup_{u \in \mathbb{R}^d} z^\top u - \frac{1}{2}\|u\|_2^2\right) \tag{8.4.8}$$

$$\overset{(*)}{=} \inf_{u\in\mathbb{R}^d}\left(\frac{1}{2r}\|u\|_2^2 + \sup_{\|z\|_2\le\rho} z^\top(b-u/r)\right) \tag{8.4.9}$$

$$= \inf_{u\in\mathbb{R}^d}\left(\frac{1}{2r}\|u\|_2^2 + \rho\|b-u/r\|_2\right) \tag{8.4.10}$$

$$= \inf_{v\in\mathbb{R}^d}\left(\frac{r}{2}\|v-b\|_2^2 + \rho\|v\|_2\right), \text{ by change of variable } v := b-u/r \tag{8.4.11}$$

$$= \rho\inf_{v\in\mathbb{R}^d}\left(\frac{1}{2\rho/r}\|v-b\|_2^2 + \|v\|_2\right), \text{ by factoring out } \rho \tag{8.4.12}$$

$$\overset{(**)}{=} \rho\begin{cases}\|b\|_2^2/(2\rho/r), & \text{if } \|b\|_2\le\rho/r,\\ \|b\|_2-\rho/(2r), & \text{else}\end{cases} \tag{8.4.13}$$

$$= \begin{cases} r\|b\|_2^2/2, & \text{if } \|b\|_2\le\rho/r,\\ \rho\|b\|_2-\rho^2/(2r), & \text{else,}\end{cases} \tag{8.4.14}$$

where $(*)$ uses *Sion's Minimax Theorem*, and in $(**)$ we have recognized a rescaled *Moreau envelope* of the Euclidean norm, which is the Huber function evaluated at $\|b\|_2$. □

We will also need the following auxiliary lemma.

Lemma 8.4.8. AUXILIARY LEMMA (2). *For any $r,\rho>0$ and $b\in\mathbb{R}^d$, we have the identity*

$$\sup_{z\in\rho B_n} b^\top z - \frac{1}{r}\|z\|_2 = \rho(\|b\|_2 - 1/r)_+. \tag{8.4.15}$$

Proof of Lemma 8.4.8.

By direct computation, we have

$$\sup_{\|z\|_2\le\rho} b^\top z - \frac{1}{r}\|z\|_2 = \sup_{\|z\|_2\le\rho} b^\top z - \sup_{\|u\|_2\le 1} z^\top u/r \tag{8.4.16}$$

$$= \inf_{\|u\|_2\le 1}\sup_{\|z\|_2\le\rho} z^\top(b-u/r) \tag{8.4.17}$$

$$= \rho\inf_{\|u\|_2\le 1}\|b-u/r\|_2 \tag{8.4.18}$$

$$= \rho(\|b\|_2 - 1/r)_+, \tag{8.4.19}$$

where, in the last step, we have recognized the well-known block *soft-thresholding* operator. □

Finally, we will need the following lemma.

Lemma 8.4.9. AUXILIARY LEMMA (3). *Suppose R_1, R_2, R_3 are random variables and $\phi:\mathbb{R}\to[-\infty,\infty]$ is a possibly random nondecreasing function. If $\mathbb{P}(R_2\ge R_3)\ge 1-\delta$*

$$\mathbb{P}(R_1\le\phi(R_2)) \ge \mathbb{P}(R_1\le\phi(R_3)) - \delta. \tag{8.4.20}$$

Proof of Lemma 8.4.9.

Indeed, consider the events $E_1 := \{R_1 \le \phi(R_3)\}$, $E_2 := \{R_3 \le R_2\}$, $E_3 := E_1 \cap E_2$ and $E_4 := \{R_1 \le \phi(R_2)\}$. It is clear that $E_3 \subseteq E_4$. One then easily computes

$$\begin{aligned}
\mathbb{P}(R_1 \le \phi(R_2)) = \mathbb{P}(E_4) &\ge \mathbb{P}(E_3) = \mathbb{P}(E_1 \cap E_2) && (8.4.21)\\
&= \mathbb{P}(E_1) + \mathbb{P}(E_2) - \mathbb{P}(E_1 \cup E_2) && (8.4.22)\\
&\ge \mathbb{P}(E_1) + \mathbb{P}(E_2) - 1 && (8.4.23)\\
&\ge \mathbb{P}(E_1) - \delta && (8.4.24)\\
&= \mathbb{P}(R_1 \le \phi(R_3)) - \delta, && (8.4.25)
\end{aligned}$$

as claimed. □

Proof of Theorem 8.4.5: Lipschitz decision-boundary.

We are now ready to prove Theorem 8.4.5. First, we restate it for convenience

Theorem 8.4.5. *Let g_θ be a feature map that satisfies the Lipschitz gradient condition from Condition 8.4.3, and let V be a possibly random adversarially (α, β)-viable subspace of $\mathbb{R}^d$. Then,*

(A) For any $\varepsilon \ge 0$, the average fooling rate of V is lower-bounded as follows:

$$\mathbb{E}_V[\mathrm{FR}(V;\varepsilon)] \ge P_X\left(m_{g_\theta}(X) \le \min\left(\alpha\varepsilon/2, \alpha^2\|\nabla g_\theta(X)\|_2/(2L)\right) \mid X \in C'\right) - \beta. \quad (8.4.3)$$

(B) If in addition, g_θ satisfies the strong gradient condition from Condition 8.4.4, then for any $0 \le \varepsilon \le \alpha\gamma/L$,

$$\mathbb{E}_V[\mathrm{FR}(V;\varepsilon)] \ge P_X\left(m_{g_\theta}(X) \le \alpha\varepsilon/2 \mid X \in C'\right) - \beta - \lambda. \quad (8.4.4)$$

Proof of Theorem 8.4.5.

Let $x \in C' := \mathbb{R}^d \setminus C$ and let $v(x) := \Pi_V \nabla g_\theta(x)/\|\Pi_V \nabla g_\theta(x)\|_2 \in \mathcal{S}_{d-1} \cap V$.

Define $p_V(x) := \|\Pi_V \nabla g_\theta(x)\|_2$, the L_2-norm of the orthogonal projection of the gradient vector $\nabla g_\theta(x)$ onto the subspace V. It is clear that $\nabla g_\theta(x)^\top v(x) = \|\Pi_V \nabla g_\theta(x)\|_2 = p_V(x)$. Let $d_V(x) \in (0, \infty]$ be the distance of x from C along the subspace V (see Equation (8.4.48)). By definition, $d_V(x)$ is no larger than the distance between x and the point where the line $x + \mathbb{R}v(x) := \{x + sv(x) \mid s \in \mathbb{R}\}$ first meets C (if it meets it at all!).

Thus, with the convention $\inf \emptyset = \infty$, we have

$$d_V(x) \le \inf_{s \in \mathbb{R}} |s| \text{ subject to } x + sv(x) \in C \quad (8.4.26)$$

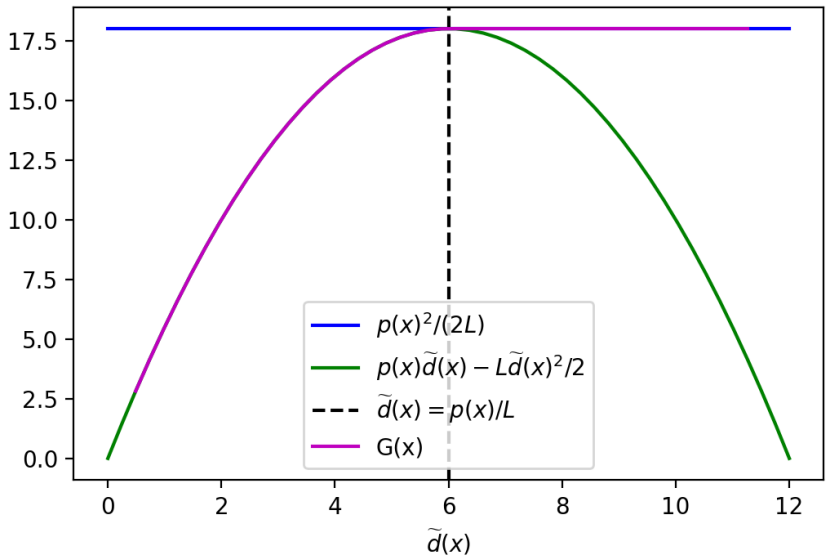


Figure 8.3: Graphical illustration of the RHS of Equation (8.4.30), denote here as $G(x)$. In this illustration, $p(x) = p_V(x)$ and L are fixed to 5 and 1 respectively. Here, $\widetilde{d}(x)$ is shorthand for $d_V(x)$, the distance of x from C along the subspace V.

$$= \inf_{s\in\mathbb{R}} |s| \text{ subject to } g_\theta(x + sv(x)) \le 0 \tag{8.4.27}$$

$$\le \inf_{s\in\mathbb{R}} |s| \text{ subject to } g_\theta(x) + s\nabla g_\theta(x)^\top v(x) + Ls^2/2 \le 0 \tag{8.4.28}$$

$$= \inf_{s\in\mathbb{R}} |s| \text{ subject to } g_\theta(x) + p_V(x)s + Ls^2/2 \le 0, \tag{8.4.29}$$

where we have invoked the right-hand side of Equation (8.4.5) with $x' = x + sv(x)$ to arrive at the third line.

$$g_\theta(x) \ge \sup_{|s|<d_V(x)} -p_V(x)s - Ls^2/2 = \begin{cases} p_V(x)^2/(2L), & \text{if } p_V(x) \le Ld_V(x), \\ p_V(x)d_V(x) - Ld_V(x)^2/2, & \text{otherwise,} \end{cases} \tag{8.4.30}$$

where the second step is an application of Lemma 8.4.7 with $n = 1$, $b = -p_V(x)$, $r = 1/L$ and $\rho = d_V(x)$.

Now, if $g_\theta(x) < p_V(x)^2/(2L)$, we deduce from Equation (8.4.30) that $d_V(x) < p_V(x)/L$ and $g_\theta(x) \ge p_V(x)d_V(x) - Ld_V(x)^2/2$ (see Figure 8.3 for geometric intuition), and so

$$d_V(x) \le p_V(x)/L - \sqrt{(p_V(x)/L)^2 - 2g_\theta(x)/L} \tag{8.4.31}$$

$$= \frac{2g_\theta(x)}{p_V(x) + \sqrt{p_V(x)^2 - 2g_\theta(x)L}} \tag{8.4.32}$$

$$\le \frac{2g_\theta(x)}{p_V(x)} = \frac{2m_{g_\theta}(x)}{\alpha_V(x)}, \tag{8.4.33}$$

where $\alpha_V(x) = p_V(x)/\|\nabla g_\theta(x)\|_2 = \|\Pi_V \nabla g_\theta(x)\|_2/\|\nabla g_\theta(x)\|_2 = \|\Pi_V \eta(x)\|_2$.

Now, because $C_V^\varepsilon = \{x \in \mathbb{R}^d \mid d_V(x) \leq \varepsilon\}$, we deduce that

$$\left\{x \in C' \mid m_{g_\theta}(x) \leq \min\left(\frac{\alpha_V(x)\varepsilon}{2}, \frac{\alpha_V(x)^2\|\nabla g_\theta(x)\|_2}{2L}\right)\right\} \subseteq C_V^\varepsilon \setminus C. \tag{8.4.34}$$

Now, define $s_V(x) := \alpha_V(x)^2\|\nabla g_\theta(x)\|_2/(2L)$ and $s(x) := \alpha^2\|\nabla g_\theta(x)\|_2/(2L)$. Since the subspace V is an adversarial (α, β)-viable by hypothesis, it follows from Definition 8.3.1 that

$$\begin{aligned} P_{X,V}(\min(\alpha_V(X)\epsilon/2, s_V(X)) \geq \min(\alpha\epsilon/2, s(X)) \mid X \in C') \\ \geq P_{X,V}(\|\Pi_V \eta(X)\|_2 \geq \alpha \mid X \in C') \geq 1 - \beta. \end{aligned} \tag{8.4.35}$$

The *Fubini-Tonelli Theorem* then gives,

$$\begin{aligned} \mathrm{FR}(V; \varepsilon) &:= \mathbb{E}_V[P_X(X \in C_V^\varepsilon \mid X \in C')] && (8.4.36)\\ &= \mathbb{E}_X[P_V(X \in C_V^\varepsilon \mid X \in C')] && (8.4.37)\\ &\geq \mathbb{E}_X[P_V(m_{g_\theta}(X) \leq \min(\alpha_V(X)\varepsilon/2, s_V(X)) \mid X \in C')] && (8.4.38)\\ &\geq \mathbb{E}_X[P_V(m_{g_\theta}(X) \leq \min(\alpha\varepsilon/2, s(X)) \mid X \in C') - \beta], && (8.4.39) \end{aligned}$$

where the last step is obtained thanks to Lemma 8.4.9 with $R_1 = m_{g_\theta}(X)$, $R_2 = \min(\alpha_V(x)\epsilon/2, s_V(X))$, $R_3 = \min(\alpha\epsilon/2, s(X))$, and $\phi = Id$, and recalling Equation (8.4.35). This proves the first part of the theorem.

For the second part, g_θ also satisfies Condition 8.4.4 so we have $P(\|\nabla g_\theta(X)\|_2 \geq \gamma \mid X \in C') \geq 1 - \lambda$. On the other hand, if $0 \leq \epsilon \leq \alpha\gamma/L$, then conditioned on the event $\|\nabla g_\theta(x)\|_2 \geq \gamma$, we have $\min(\alpha\epsilon/2, s(X)) \geq \min(\alpha\epsilon/2, \alpha^2\gamma/(2L)) = \alpha\epsilon/2$, and the result follows from the first part and Lemma 8.4.9. □

8.4.3 Some Applications

We provide a non-exhaustive list of examples to illustrate the power of Theorem 8.4.5.

Linear models.

Proposition 8.2.5 which is a generalization of Lemma 2.2 of Guo (2020) is itself a special case of part (B) of Theorem 8.4.5. Indeed the linear function $g_\theta(x) \equiv x^\top w + b$ has margin $m_{g_\theta}(x) = (x^\top w + b)_+$ and verifies Conditions 8.4.3 and 8.4.4 with $\gamma = \|w\|$, $L = 0$, and $\lambda = 0$. Also, thanks to Lemma 8.3.2, for any $k \in [d]$ and $t \in (0, \sqrt{k/d})$, a random k-dimensional subspace V of $\mathbb{R}^d$ is adversarially (α, β)-viable with $\alpha = \sqrt{k/d} - t$ and $\beta = 2e^{-t^2 d/2}$.

Hyper-ellipsoids.

We now generalize another result of Guo (2020), namely, Lemma 2.3 therein. Indeed, consider the case where $g_\theta(x) := (x^\top Bx - r^2)/2$, where B is a $d \times d$ positive semi-definite matrix and $r > 0$ is a scalar, so that the negative decision-region C of the classifier is the

hyper-ellipsoid $g_\theta \leq 0$. In particular, C is a solid sphere of radius r when $B = I_d$. One computes $\nabla g_\theta(x) = Bx$, $\nabla^2 g_\theta(x) = B$, hence Conditions 8.4.3 and 8.4.4 are satisfied with $\lambda = 0$ and

$$L = \sup_{x \in \mathbb{R}^d} \|\nabla^2 g_\theta(x)\|_{op} = \|B\|_{op}, \tag{8.4.40}$$

$$\|\nabla g_\theta(x)\|_2 = \|Bx\|_2, \text{ for all } x \in \mathbb{R}^d, \tag{8.4.41}$$

$$\gamma = \inf_{x \in C'} \|\nabla g_\theta(x)\|_2 = \inf_{x^\top Bx > r^2} \|Bx\|_2 = s_{\min}(B)^{1/2} r, \tag{8.4.42}$$

where $s_{\min}(B)$ is the smallest singular / eigenvalue of B, and $\|B\|_{op}$ is the operator norm of B, i.e, its largest eigenvalue (since B is positive semi-definite).

Moreover, the margin of g_θ at a any point $x \in \mathbb{R}^d$ is given by

$$m_{g_\theta}(x) = \frac{\max(g_\theta(x), 0)}{\|\nabla g_\theta(x)\|_2} = \frac{(x^\top Bx - r^2)_+}{2\|Bx\|_2}. \tag{8.4.43}$$

In particular, if $B = I_d$, then we deduce $L = 1$, $\gamma = r$, Moreover, for any $x \in C'$, then the distance of x from C, i.e $d(x) = \|x\|_2 - r$ and we have

$$m_{g_\theta}(x) = \frac{\|x\|_2^2 - r^2}{2\|x\|_2} = \frac{1}{2}(\|x\|_2 - r)(1 + \frac{r}{\|x\|_2}) \in (d(x)/2, d(x)), \text{ for all } x \in C' \tag{8.4.44}$$

Applying Theorem 8.4.5 with $B = I_d$ (corresponding to a solid sphere) then recovers, as is expected, exactly the bounds established in Lemma 2.3 of the work from Guo (2020) as a special case.

8.4.4 Matching Upper-Bound under Convexity Assumption

We now show that the lower-bound given in Theorem 8.4.5 is tight by establishing a corresponding upper-bound for the case where C is convex (e.g., half-spaces, balls, ellipsoids, etc.).

Theorem 8.4.10. *Suppose g_θ is convex differentiable, and let V be a subspace of $\mathbb{R}^d$ satisfying:*

$$\|\Pi_V \eta(x)\|_2 \leq \tilde{\alpha}, \text{ for some } \tilde{\alpha} \in [0, 1] \text{ and } \forall x \in C'. \tag{8.4.45}$$

Then, for any $\varepsilon \geq 0$, we have

$$\mathrm{FR}(V; \varepsilon) \leq P_X(m_{g_\theta}(X) \leq \tilde{\alpha}\varepsilon \mid X \in C'). \tag{8.4.46}$$

Proof of Theorem 8.4.10. Let $d(x) \in [0, \infty)$ be the distance of x from C and let $d_V(x) \in [0, \infty]$ be the distance of x from C along the subspace V, i.e.,

$$d(x) := \inf_{v \in \mathbb{R}^d} \|v\|_2 \text{ subject to } x + v \in C, \tag{8.4.47}$$

$$d_V(x) := \inf_{v \in V} \|v\|_2 \text{ subject to } x + v \in C, \tag{8.4.48}$$

with the convention that $\inf \emptyset = \infty$. By definition of the (ε, V)-expansion C_V^ε of C as defined in Definition 8.2.3, we have:

$$C_V^\varepsilon = \{x \in \mathbb{R}^d \mid d_V(x) \leq \varepsilon\}. \tag{8.4.49}$$

Also, it is clear that $d_V(x) \geq d(x)$, attained when $V = \mathbb{R}^d$. By definition of $d_V(x)$, it is clear that $x - d_V(x)v \in C$, where $v = \Pi_V \nabla g_\theta(x) / \|\Pi_V \nabla g_\theta(x)\|_2$. Observe that $\nabla g_\theta(x)^\top v = \|\Pi_V \nabla g_\theta(x)\|_2$. Now, thanks to the convexity of g_θ, we have

$$g_\theta(x') \geq g_\theta(x) + \nabla g_\theta(x)^\top (x' - x), \tag{8.4.50}$$

for all $x' \in \mathbb{R}^d$. Thus,

$$x - d_V(x)v \in C \implies g_\theta(x - d_V(x)v) \leq 0 \tag{8.4.51}$$

$$\implies g_\theta(x) - d_V(x) \nabla g_\theta(x)^\top v \leq 0 \text{ thanks to Equation (8.4.50)} \tag{8.4.52}$$

$$\implies m_{g_\theta}(x) \leq \frac{d_V(x) \nabla g_\theta(x)^\top v}{\|\nabla g_\theta(x)\|_2} \leq \frac{d_V(x) \|\Pi_V \nabla g_\theta(x)\|_2}{\|\nabla g_\theta(x)\|_2} \leq \tilde{\alpha} d_V(x). \tag{8.4.53}$$

We deduce that $\{x \in C' \mid m_{g_\theta}(x) \leq \tilde{\alpha}\varepsilon\} \supseteq \{x \in C' \mid d_V(x) \leq \varepsilon\} =: C_V^\varepsilon \setminus C$, and the result follows. □

8.5 Model with Locally Almost-Affine Decision Boundary

8.5.1 Main result on the Lower Bound

We now consider the following smoothness condition for the classifier Definition 8.2.1.

Condition 8.5.1. GRADIENTS VARY SMOOTHLY. *The feature map g_θ is said to have gradients that vary smoothly if there exists $0 < R \leq \infty$ and $\tau \geq 0$ such that*

$$\|\nabla g_\theta(x + \Delta x) - \nabla g_\theta(x)\|_2 \leq \tau \text{ for all } x \in \text{supp}(P_X),\ \Delta x \in \mathbb{R}^d \text{ with } \|\Delta x\|_2 \leq R. \tag{8.5.1}$$

Examples of functions that satisfy this condition include half-spaces and wide feedforward ReLU neural networks with randomly initialized intermediate weights, where $\tau = o(1)$ with high probability over the intermediate weights, as will be seen in Section 8.5.3. The following theorem is one of our main contributions.

Theorem 8.5.2. *Suppose that the feature map g_θ satisfies Conditions 8.5.1 and 8.4.4 with parameters $\gamma \in (0, \infty)$, $R \in (0, \infty]$ and $\tau \geq 0$ are in order. Let V be a possibly random adversarially (α, β)-viable subspace of $\mathbb{R}^d$ with $\alpha > \tau/\gamma$. Then, for any $0 \leq \varepsilon \leq R$, the average fooling rate of V is lower-bounded as follows:*

$$\mathbb{E}_V[\text{FR}(V; \varepsilon)] \geq P_X(m_{g_\theta}(X) \leq \overline{\alpha}\varepsilon \mid X \in C') - \beta - \lambda, \tag{8.5.2}$$

where $\overline{\alpha} := \alpha - \tau/\gamma > 0$.

Remark 8.5.3. TIGHTNESS. *Theorem 8.5.2 is tight, as can be seen by considering the case where C is a half-space in which case $g_\theta(x) = x^\top w - b$, for some unit-vector $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$; take $V = \mathbb{R}w$. **N.B.:** $\nabla g_\theta(x) \equiv w$, and so Conditions 8.5.1 and 8.4.4 hold with $\alpha = \gamma = 1$, $\tau = \lambda = 0$, and $R = \infty$.*

8.5.2 Proof of the Main Result

We first provide a quick proof sketch before providing a full proof.

Proof Sketch. of Theorem 8.5.2

The core of the proof is similar to that of Theorem 8.4.5, but with Equation (8.4.5) replaced by the following inequality which holds under Condition 8.5.1 for all $x \in \text{supp}(P_X)$ and $\Delta x \in \mathbb{R}^d$ with $\|\Delta x\|_2 \leq R$

$$-\tau\|\Delta x\|_2 \leq g_\theta(x + \Delta x) - g_\theta(x) - \nabla g_\theta(x)^\top \Delta x \leq \tau\|\Delta x\|_2. \tag{8.5.3}$$

□

Now, let's dig into the full proof.

Proof of Theorem 8.5.2

Under Condition 8.5.1, it is easy to establish the classical inequality

$$-\tau\|x'-x\|_2 \leq g_\theta(x')-g_\theta(x)-\nabla g_\theta(x)^\top(x'-x) \leq \tau\|x'-x\|_2, \text{ for all } \|x'-x\|_2 \leq R. \tag{8.5.4}$$

Now, let $x \in C' := \mathbb{R}^d \setminus C$ and let $d_V(x)$ be the distance of x from V along the subspace V. Let $v(x)$, $p_V(x)$, $\alpha_V(x)$, $s_V(x)$, and $s(x)$ be as defined in the proof of Theorem 8.4.5. By an argument analogous to the beginning of the proof of Theorem 8.4.5 but with Equation (8.5.4) used in place of Equation (8.4.5) and the restriction that $|s| \leq R$ so that Equation (8.5.4) is valid for every x' on the line $x + \mathbb{R}v(x)$, it is straightforward to establish that

$$\begin{aligned} d_V(x) &\leq \inf_{s\in\mathbb{R}} |s| \text{ subject to } x + sv(x) \in C,\ |s| \leq R \\ &\leq \inf_{s\in\mathbb{R}} |s| \text{ subject to } g_\theta(x) + p_V(x)s + \tau|s| \leq 0,\ |s| \leq R \\ &\leq \inf_{s\in\mathbb{R}} |s| \text{ subject to } g_\theta(x) + p_V(x)s + \tau|s| \leq 0,\ |s| \leq R. \end{aligned} \tag{8.5.5}$$

We deduce that

$$g_\theta(x) \geq \sup_{|s|<\min(d_V(x),R)} -p_V(x)s - \tau|s| = \min(d_V(x), R) \cdot (p_V(x) - \tau)_+, \tag{8.5.6}$$

where the equality is thanks to Lemma 8.4.8 applied with $n = 1$, $b = -p_V(x)$, $r = 1/\tau$, and $\rho = \min(d_V(x), R)$. Thus, we deduce from Equation (8.5.6) that

$$\min(d_V(x), R) \leq \frac{g_\theta(x)}{(p_V(x) - \tau)_+} = c_V(x) m_{g_\theta}(x), \tag{8.5.7}$$

with $c_V(x) := \|\nabla g_\theta(x)\|_2 / (\alpha_V(x) \|\nabla g_\theta(x)\|_2 - \tau)_+$. One the event $\alpha_V(x) \geq \alpha > \tau/\gamma$, we have $1/c_V(x) \geq \overline{\alpha} := \alpha - \tau/\gamma$. Thus, if $m_{g_\theta}(x) \leq \overline{\alpha}\varepsilon$ and $0 \leq \varepsilon < R$, then $d_V(x) \leq \varepsilon$. That is, if $0 \leq \epsilon < R$:

$$\{x \in C' \mid m_{g_\theta}(x) \leq \overline{\alpha}\varepsilon\} \subseteq C_V^\varepsilon \setminus C. \tag{8.5.8}$$

The rest of the proof is analogous to the end of the proof of the first part of Theorem 8.4.5 (starting from the set-inclusion Equation (8.4.34)), and is thus omitted. □

8.5.3 ReLU Networks in the Random Features Regime

Consider a feed-forward neural net with $M \geq 2$ layers with parameters matrices $W_1 \in \mathbb{R}^{d_0 \times d_1}, W_2 \in \mathbb{R}^{d_1 \times d_2}, \ldots, W_M = a \in \mathbb{R}^{d_{M-1} \times d_M}$, where $d_0 = d$ and $d_M := 1$. Each d_ℓ is the width of the ℓ layer, and the matrices $W_1, \ldots, W_{M-1}$ are the intermediate weights matrices, while $W_M = a$ is the output weights vector. For an input $x \in \mathbb{R}^d$, the output of the neural net is

$$\begin{aligned} g_\theta(x) = z_M := a^\top z_{M-1} \in \mathbb{R}, \text{ with } z_0 := x, \\ z_\ell := \sigma(W_\ell^\top z_{\ell-1}) \in \mathbb{R}^\ell, \forall \ell \in [M-1]. \end{aligned} \tag{8.5.9}$$

Here, σ is the *activation function* and is applied entry-wise. We will focus on the case of ReLU neural networks, where $\sigma(t) \equiv (t)_+$. The matrices $W_1, \ldots, W_M$ are randomly initialized as follows:

$$[W_\ell]_{i,j} \overset{iid}{\sim} N(0, 1/d_{\ell-1}), \text{for all } \ell \in [M], i \in [d_\ell], j \in [d_{\ell-1}]. \tag{8.5.10}$$

The output weights vector $a \in \mathbb{R}^{d_{M-1}}$ can be arbitrary, for example:

- (1) random, as in Daniely and Shacham (2020); Bartlett et al. (2021)
- (2) optimized to fit training data, as in the so-called random features (RF) regime as in Rahimi and Recht (2008, 2009), with L_2-regularization on a.

Let $d_{\min} := \min_{0 \leq \ell \leq M-1} d_\ell$ and $d_{\max} := \max_{0 \leq \ell \leq M-1} d_\ell$ be respectively, the minimum and maximum width of the layers. As in Bartlett et al. (2021), assume the following condition:

Condition 8.5.4. Genuinely Wide, Finite-Width. *The neural network is said to be genuinely wide and finite-width if its architecture satisfies the two conditions:*

(i) Bounded depth, i.e., $M = \mathcal{O}(1)$ layers.

(ii) Genuinely wide, i.e., $d_{\min} \gtrsim (\log d_{\max})^{40M}$ and $d_{\min} \to \infty$.

Then, we have the following corollary to Theorem 8.5.2.

Corollary 8.5.5. *Consider the case where the marginal distribution of the covariates X is supported on the sphere of radius $\sqrt{d}$ in $\mathbb{R}^d$, and g_θ is the ReU neural network defined in Equation (8.5.9) with random intermediate weights $W_1, \dots, W_{M-1}$ sampled according to Equation (8.5.10).*

Suppose that g_θ satisfies Condition 8.5.4. Let V be a possibly random (α, β)-viable subspace of $\mathbb{R}^d$, with $\alpha = \Omega(1)$. Then, for $0 \le \varepsilon \lesssim (\log d_{\max})^{40M}$, it holds with high probability over $W_1, \dots, W_{M-1}$ that :

$$\mathbb{E}_V[\mathrm{FR}(V;\varepsilon)] \gtrsim P_X(m_{g_\theta}(X) \le \varepsilon \mid X \in C') - \beta. \tag{8.5.11}$$

In particular, at initialization, we have $\mathbb{E}_V[\mathrm{FR}(V;\varepsilon)] \gtrsim 1 - \beta$ for all $\varepsilon \ge \varepsilon_0$, where ε_0 is an absolute constant.

The second part of the result implies that the subspace V contains adversarial perturbations of size $\sqrt{d}$ times smaller than the norm of a typical data point. Thus, it is a generalizes Daniely and Shacham (2020); Bartlett et al. (2021) to subspaces.

Proof of Corollary 8.5.5.

The first part is obtained as a consequence of Theorem 8.5.2, by combining Lemma 2.2 and Lemma 2.8 of Bartlett et al. (2021) and Lemma 8.5.6 that is provided below.

The second part follows because with high probability over intermediate weights, it holds that

$$m_{g_\theta}(x) \le |g_\theta(x)|/\|\nabla g_\theta(x)\|_2 \lesssim \|a\|_2 \|z_{L-1}(x)\|_2/\|a\|_2 = \|z_{L-1}(x)\|_2 \le \|x\|_2/\sqrt{d} = 1,$$

where the last inequality is because $z_{L-1}(x)$ is $(\|x\|_2^2/d)$-subGaussian. □

Lemma 8.5.6. *Suppose P_X is supported on the sphere $\sqrt{d}\mathcal{S}_{d-1}$ and suppose g_θ satisfies Condition 8.5.4. Then, with high probability over the random intermediate weights $W_1, \dots, W_{M-1}$, the ReLU neural network g_θ defined in Equation (8.5.9) satisfies Conditions 8.5.1 and 8.4.4 with:*

$$\lambda = 0,\ R = \frac{\sqrt{d_{min}}}{(\log d_{\max})^{80M}} = \Omega((\log d_{max})^{40M}),\ \tau = \frac{\|a\|_2}{(\log d_{\max})^{M}},\ \gamma = \|a\|_2. \tag{8.5.12}$$

8.5.4 ReLU Networks in the Lazy Regime

At the moment, we are not able to extend our theoretical results to fully-trained neural networks. An exception is when the model is in the *lazy regime*, whereby the parameters of the network stay close to their value at definition. More, precisely

Definition 8.5.7. Lazy Regime. *The neural network g_θ from Equation (8.5.9) is said to be in the lazy regime if:*

$$\sup_{j \in [d_\ell]} \frac{\|W_{\ell,j} - W^0_{\ell,j}\|_2}{\|W^0_{\ell,j}\|_2} \lesssim \frac{1}{\sqrt{d_\ell}} \text{ for all } \ell \in [M-1], \tag{8.5.13}$$

where W^0_ℓ is the initialization for the parameter matrix W_ℓ of the ℓth layer.

Note that the lazy regime as defined above subsumes both ReU neural networks at initialization and in the random features regime (studied in Section Section 8.5.3). Now, in Wang et al. (2022), it was shown that if $M = 2$ (i.e two-layer ReLU neural network), then there exist absolute positive constants c_1, c_2, c_3, and c_4 such that: if the neural network is in the lazy regime, then with high probability over the initialization, the following hold simultaneously for all $x \in \sqrt{d}\mathcal{S}_{d-1}$ and $\Delta x \in \mathbb{R}^d$ with $\|\Delta x\|_2 \leq c_1$:

$$\textbf{(Small Outputs)}\ |g_\theta(x)| \leq c_2, \tag{8.5.14}$$

$$\textbf{(Strong Gradients)}\ \|\nabla g_\theta(x)\| \geq c_3, \tag{8.5.15}$$

$$\textbf{(Bounded Gradient Oscillations)}\ \|\nabla g_\theta(x + \Delta x) - \nabla g_\theta(x)\| \leq c_4. \tag{8.5.16}$$

See Lemma B.5, Lemma B.7, and Lemma B.9 (resp.) of Wang et al. (2022). We deduce that in the lazy regime, with high probability over initialization, Conditions 8.5.1 and 8.4.4 hold with $R = c_1$ and $\gamma = c_3$, $\tau = c_4$, and with $\lambda = 0$. On the same event, we also deduce the following margin bound:

$$m_{g_\theta}(x) = \frac{(g_\theta(x))_+}{\|\nabla g_\theta(x)\|_2} \leq \frac{|g_\theta(x)|}{\|\nabla g_\theta(x)\|_2} \leq \frac{c_2}{c_3} =: c_5, \tag{8.5.17}$$

for all $x \in \sqrt{d}\mathcal{S}_{d-1}$. Combining with Theorem 8.5.2, we obtain the following important corollary.

Corollary 8.5.8. *Suppose g_θ is a two-layer neural network defined as in Equation (8.5.9) which is in the lazy regime. Also, suppose the marginal distribution of the features X is supported on the sphere $\sqrt{d}\mathcal{S}_{d-1}$. If V is an adversarially (α, β)-viable subspace of $\mathbb{R}^d$, then for any $0 \leq \varepsilon \leq R = c_1$ then with high probability over the initial weights, the average fooling rate of V is lower-bounded as in Equation (8.5.2), with $\gamma = c_3$, $\tau = c_4$, and $\lambda = 0$.*

In particular, if $c_5 \leq \varepsilon \leq c_1$, it holds that $\mathbb{E}_V[\mathrm{FR}(V; \varepsilon)] \geq 1 - \beta$.

8.6 Experimental Application to Trained Neural Networks

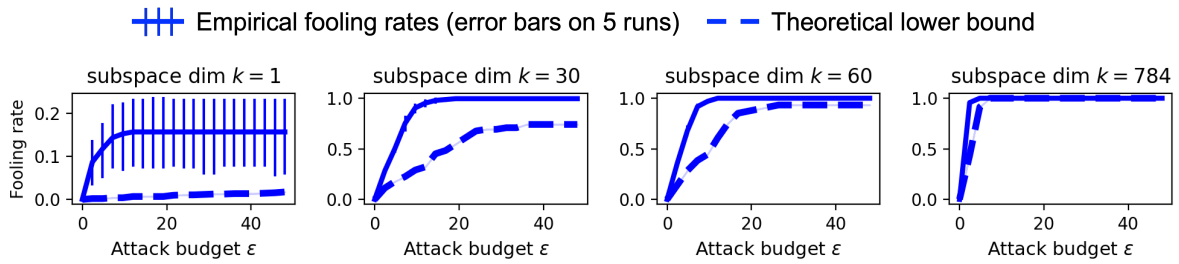
Our results are empirically verified in Figure 8.4 for random subspace attacks and Figure 8.5 for singular subspace attacks. Given a binary classifier $f_\theta : x \mapsto \mathrm{sign}(g_\theta(x))$ on $\mathbb{R}^d$, for example a neural net, with negative decision-region $C := \{x \in \mathbb{R}^d \mid f_{theta}(x) = -1\}$. For a subspace $V \subseteq \mathbb{R}^d$ and a (Euclidean) attack budget ε, refer to Definition 8.2.4 for the fooling rate of V on the classifier f_θ.

8.6.1 Consequence of Our Results

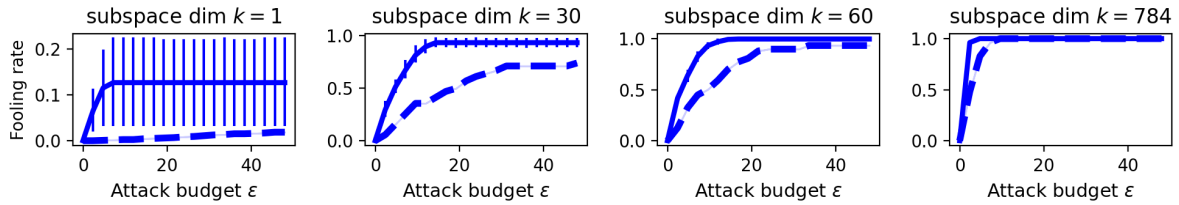
Before studying the aforementioned results, let us now outline some consequences for practical classifiers, neural networks. First, we recall the general form of our results. Given a possibly random adversarially (α, β)-viable subspace V of $\mathbb{R}^d$, we have established in Theorems 8.4.5 and 8.5.2 lower-bounds on the fooling rate of the form

$$\mathbb{E}_V[\mathrm{FR}(V; \varepsilon)] \gtrsim \mathbb{P}(m_{g_\theta}(X) \leq \overline{\alpha}\varepsilon \mid X \in C') - \beta - \lambda. \tag{8.6.1}$$

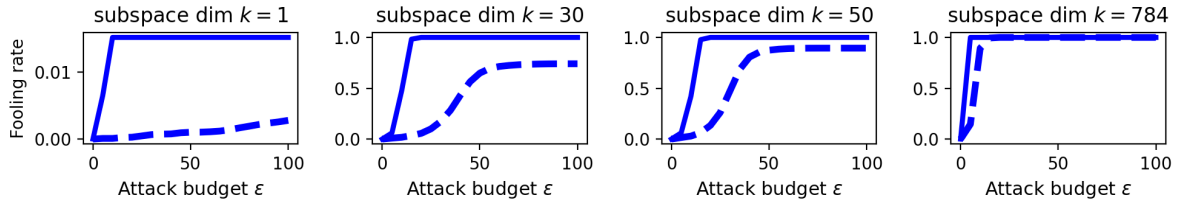
Here, the scalar $\overline{\alpha} \in (0, 1]$ depends on α, γ and the smoothness of g_θ as in Condition 8.4.4. Importantly, the generic bound Equation (8.6.1) explicitly highlights the dependence of



(a) 2-layer ReLU NN at init.: input dim. $d = 784$, width $d_1 = 100$. Simulated data.



(b) 2-layer ReLU NN in RF regime: input dim $d = 784$, width $d_1 = 100$. Simulated data.



(c) Full-trained LeNet (conv layers + dense layers + ReLU activation) on MNIST.

Figure 8.4: (Random subspace attack) Empirical confirmation of our results. Broken lines correspond to our theoretical lower-bounds Equation (8.6.1), for different neural network regimes. k is the dimension of the random subspace from which the perturbations are constructed. In the first two rows, d_1 is the width of the network. Solid curves correspond to empirically computed fooling rates, with error bars accounting for randomness in the initialization of the network, over 5 independent runs. Our theoretical lower bounds are confirmed in all cases.

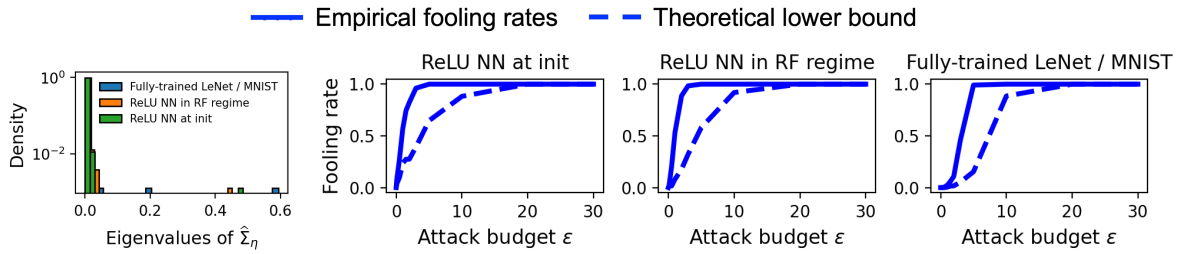


Figure 8.5: (Eigen-subspace attack). Same experimental setting in Figure 8.4. **Leftmost plot:** Showing a histogram of the eigenvalues of empirical covariance matrix $\hat{\Sigma}_\eta$ of gradient directions (computed on 1000 examples). Notice how the largest eigenvalue for each model is much larger than the other eigenvalues. **Second to fourth (rightmost) plot:** Notice how the fooling rate rises rapidly.

the fooling rate on the pointwise margin of the classifier and on the alignment of the given subspace with the gradients of g_θ.

The L_2-norm of a typical data point is of order $\sqrt{d}$, while the margin $m_{g_\theta}(X)$ is typically of order $\mathcal{O}(1)$, as (i) observed empirically observed in Jiang et al. (2019) for general trained neural networks (ii) formally proved in Daniely and Shacham (2020); Bartlett et al. (2021) for the case of ReLU networks at initialization and in Wang et al. (2022) for the case of *lazy regime* where the intermediate parameters of the network stay close to their initial values throughout training (see Equation (8.5.17)). Also, as observed in Moosavi-Dezfooli et al. (2017), the singular values of the gradient covariance matrix Σ_η are typically long-tailed.

Thus, combining with Theorem 8.3.3, our results predict that for sufficiently large $k \ll d$, the subspace spanned by the top k singular-vectors of Σ_η has a nonzero fooling rate with attack budget $\varepsilon \asymp 1/\widetilde{\alpha} = \mathcal{O}(1)$ which is $\sqrt{d}/\varepsilon = \Omega(\sqrt{d})$ times smaller than $\mathbb{E}(||X||_2)$, the L_2-norm of a typical data point, for ReLU neural networks in the lazy regime.

8.6.2 Random Subspace Attacks

In Figure 8.4 (first and second row), the distribution P_X of the features is $N(0, I_d)$, and the training labels are given from a simple linear model: $y_i = x_{ij}$. For a classical LeNet convolutional neural network trained on MNIST data (see LeCun and Cortes (2010)) (third row), we construct a binary classification dataset $n = 2 \times 10\text{K} = 20\text{K}$ by restricting it to the digits 0 and 8. As in Guo et al. (2018a), we run PGD (see Madry et al. (2018)) attacks on a randomly chosen subspace V (of different dimensions) of the feature space $\mathbb{R}^d$, and report the fooling rates (solid lines) and compare them with our proposed lower-bounds Equation (8.6.1), from Theorem 8.5.2 with $R = \infty$ and $\tau = 0$ (these extremal values work for our experiments). As we can see from the figure, in all the cases, the lower bounds (broken lines) are verified.

8.6.3 Eigen-Subspace Attacks

In Figure 8.5, we consider the same experimental setting in Figure 8.4. We use $n = 1000$ random examples $x_1, \ldots, x_n$, and compute the empirical covariance matrix of the gradient

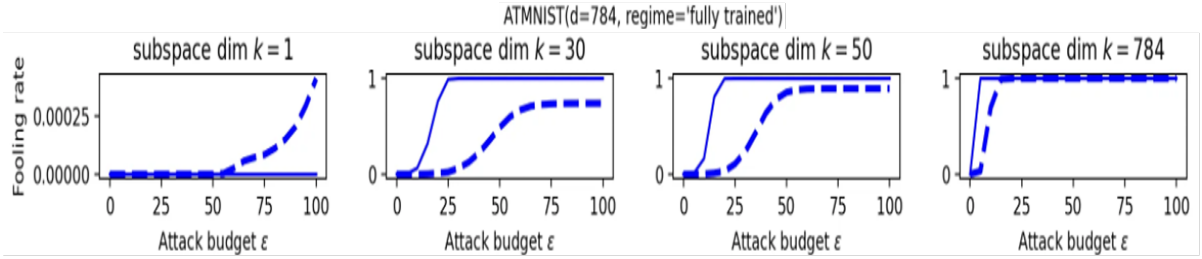


Figure 8.6: **Results for adversarially trained model.** We consider a LeNet convolutional neural network on MNIST dataset, trained with adversarial training Madry et al. (2018).

directions, i.e

$$\widehat{\Sigma}_\eta := \frac{1}{n-1}\sum_{i=1}^{n}(\eta_i - \overline{\eta})(\eta_i - \overline{\eta})^\top,$$

where $\eta_i := \eta(x_i)$, with $\overline{\eta} := (1/n)\sum_{i=1}^{n}\eta_i$. As in Khrulkov and Oseledets (2018), we extract the top eigenvector of $\widehat{\Sigma}_\eta$ and use it as a universal perturbation vector for a separate test set. In the leftmost subplot, we show a histogram of eigenvalues. Notice how the largest eigenvalue for each model is much larger than the other eigenvalues. Thanks to Theorem 8.3.3, this means that the principal eigenvector v spans an adversarially viable subspace. This is confirmed in the 2nd, 3rd, and 4th subplots where we see that the fooling rate rises rapidly as a function of the attack budget ε. We see from the figure that our predicted lower bounds (broken lines) are satisfied in all cases.

Remark 8.6.1. *The gap in Figures 8.4 and 8.5 between experiments (solid curves) and our theoretical results (broken ones) is due to the fact that our established lower bounds Equation (8.6.1), though sufficient to explain the success of low-dimensional adversarial perturbations, might be too conservative for obtaining exact quantitative estimates for the fooling rate in the case of random adversarial subspaces on neural networks, because we only use first-order information (see Conditions 8.5.1, 8.4.3 and 8.4.4) on the neural net g_θ. However, in the specific scenario where the target decision region is a half-space or hyper-ellipsoid, this gap disappears because the aforementioned first-order information is sufficient in such cases, and our estimates for the fooling rate are exact.*

8.6.4 Additional Experiments

We empirically observe in Figure 8.6 that our results remain valid both on normally and adversarially trained (AT) neural networks, and on more complex neural networks and datasets like Resnet on CIFAR10 , as shown in Figure 8.7. Compared with the last row of Figure 8.4, notice how adversarial training slightly helps to slightly decrease the fooling rate.

In this Chapter, we have conducted a rigorous analysis of the phenomenon of low-dimensional adversarial perturbations and derived tight lower bounds for the fooling rate along arbitrary adversarial subspaces based on the geometry of the target decision-region, and the alignment between the subspace and the gradients of the model, i.e., the adversarial viability of the subspace, as defined by Definition 8.3.1.

Our work provides rigorous foundations for explaining intriguing empirical observations from the literature on the subject, like (Moosavi-Dezfooli et al., 2017; Khrulkov and

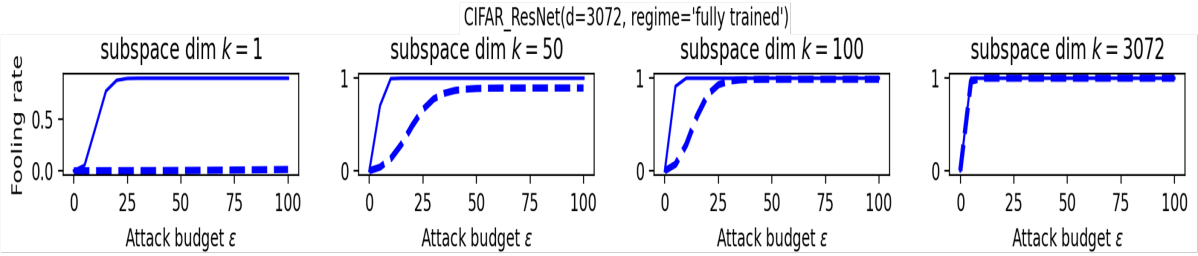


Figure 8.7: **Results on larger models.** Here we consider Resnet on CIFAR10 dataset.

Oseledets, 2018; Yin et al., 2019; Guo et al., 2018a). For the case of compact decision regions, we have shown the existence of universal adversarial perturbations. We believe our work will further generate fruitful research in this area.

In this Chapter, our analysis has focused on the case of binary classification. A non-trivial extension of our work would be the case of multi-class problems. It would also be interesting to extend our treatment of neural networks, as in Section 8.5.3, to general case, (i.e beyond the lazy regime). This would likely require the development of new theoretical tools.

Chapter 9

Conclusion about Robustness in Deep Learning

The winner plots one step ahead of the opposition and plays her trump card just after they play theirs. It's about making sure you surprise them, and they don't surprise you.

Miss Sloane.

In this Part, the topic of robustness against *evasion attacks*, specifically in the field of deep learning for image classification, was tackled. As introduced in Section 1.4, evasion attacks target already trained models at *inference* time. When considering neural networks for image classification, evasion attacks were rediscovered in 2013 and led to the explosion of the field called *adversarial robustness*, which studies evasion attacks named *adversarial attacks* by the community.

The field of adversarial robustness exploded after the seminal work of Szegedy et al. (2013). It turned out to be mainly an experimental field because of the difficulty of theoretically studying the core problem defined in Definition 1.4.9. Consequently, the field split into two distinct branches:

- Approximating the theoretical problem in Definition 1.4.9 by sequentially constructing better and better *adversarial attack algorithms* and *defense methods* in a cat-and-mouse game.
- Providing a better understanding of the adversarial phenomenon via intuitions and explanations.

In the first branch, recent breakthroughs in adversarial attack crafting methods have led to the development of *low-dimensional* adversarial perturbations. Their specificity is that they tackle only a small subspace of the image data space, contrary to most of the previously studied attacks. If theoretical works exist to justify the success of 'classical' adversarial attacks, relying on *curse of dimensionality* arguments, there wasn't any theoretical analysis explaining the success of low-dimensional attacks.

In Chapter 8, this type of attack was rigorously studied from a theoretical perspective. This thesis contributes to explaining their success by providing a lower bound on their success rate based on conditions on the model and the attack subspace. These assumptions are made on the local smoothness or linearity of the decision boundary defined by the classifier and the alignment of the attack subspace with the gradient of the model. The bounds that can be obtained based on these conditions then depend on the pointwise margin of the model, the attack budget, and the aforementioned alignment coefficient and smoothness criteria. These conditions are in fact quite general: they allow to apply the results to a large category of models, including neural networks in the random features regime and the lazy regime. Furthermore, experiments conducted on fully trained neural networks show that these bounds experimentally hold in this setting. This contribution plays a crucial role in advancing the understanding of low-dimensional attacks, which are very promising settings to apply adversarial attacks to real-world scenarios.

In the second branch, a large amount of work has been devoted to studying one aspect of neural networks or adversarial examples to better understand the phenomenon (including, for example, works on the architecture of neural networks, loss functions, decision boundary geometry, feature space, etc.).

This thesis, in Chapter 7, contributes to unifying several hypotheses that have been proposed in the literature and several characteristics of adversarial examples unveiled by previous works. More specifically, the proposed hypothesis is the following: adversarial examples leverage under-optimized edges in neural networks to transport adversarial per-

turbations to, in the end, fool the model. Indeed, neural networks are over-parameterized, meaning they have much more parameters than the number of features in the task at hand. Thus, some parameters are not properly optimized during training because the network does not need them to solve the task: this constitutes a blind spot for neural networks, that adversarial examples leverage. This hypothesis is consistent with many findings from the literature: promoting and suppressing strategies of adversarial examples, the existence of off-manifold adversarial examples, the relation between the size of neural networks and robustness, the relation between the architecture of neural networks and robustness, etc. To validate this hypothesis, Chapter 7 introduced a methodology based on *topological data analysis* to extract topological features from neural networks, and compared the topology induced by clean inputs and adversarial ones. This original methodology provides very interesting results, not only to confirm the hypothesis (through in-depth experimental results on diverse neural networks, datasets, and adversarial attacks) but also to introduce a useful tool to study neural networks and robustness.

In conclusion, this thesis has explored the two main branches of adversarial attacks studies in the context of deep learning for image classification. The two contributions have provided interesting tools and research directions to provide a better theoretical understanding of adversarial robustness. Indeed, adversarial phenomenon, current adversarial attacks, and defense methods still remain partially understood. Continuing to provide theoretical analysis or unifying unveiled characteristics that explain the phenomenon remains crucial for the quest of more robust, safe and secure applications based on neural networks.

Chapter 10

General Conclusion

Don't adventures ever have an end? I suppose not. Someone else always has to carry on the story.

J.R.R Tolkien, Lord of the Rings.

Contents

10.1 Wrap-up of the thesis

This thesis focused on the question of *robustness* in machine learning. As explained in Chapter 1, robustness can mainly be subdivided into two different parts: *poisoning* attacks which tackle models at *training* time, and *evasion* attacks which tackle models at *inference* time.

Interestingly, the research on these two types of attacks is at very different stages.

Basically, *poisoning* attacks started to be studied in the sixties and were unified under an exhaustive theory, usually called *robust statistics*. However, the main limitations of the studies about poisoning attacks are due to the restriction of the research work on classical types of data, mostly real-numbered data. In this thesis, robust statistics were extended to *ranking data*, overcoming the lack of vector space structure and the combinatorial nature of the space.

Most of the works provided in this thesis thus consisted in initiating the study of robustness in this particular space and providing a framework to allow for extensions of my works in a structural way. To summarize in a nutshell, Chapters 3 and 4 provided the following contributions.

- a practical algorithm to measure the robustness of any statistics solving the consensus ranking task.
- the definition of depth functions for ranking data to allow for defining quantile-based objects in this space.
- a trimming algorithm strategy to mimic the behavior of trimmed mean and provide more robust statistics for the consensus ranking task, that is shown to be effective experimentally and theoretically.
- a plugin based on bucket rankings to allow for undecidability between close items that can be added on top of every statistic, and is shown to be effective empirically at scale.

On the other hand, *evasion* attacks gained a large amount of interest in the context of deep learning for image classification around 2013. This field witnessed widespread recognition, triggering a proliferation of research works on the subject of adversarial examples. These works are mainly experimental, due to the difficulty of analyzing theoretically the problem, and lack of unification. To summarize in a nutshell, the contributions Chapters 7 and 8 on this topic are the following.

- a unification of some characteristics of adversarial examples unveiled by the literature through the study of under-optimized edges and information flow from the inputs passing through neural networks., which provides a better understanding of the adversarial phenomenon.
- the use of topological tools to better study neural network structure and adversarial robustness, which is new in the deep learning field and unlocks fresh avenues for

investigating neural networks.

- a very efficient detection method based on the two previously mentioned elements.
- theoretical bounds to characterize the success rate of low-dimensional attacks for a large class of models and illustrated with experiments.
- the use of the geometry of the adversarial space (and the smoothness of the model) to come up with relevant bounds instead of dimensionality-based arguments.

10.2 Extensions and perspectives

The field of robustness in machine learning holds immense promise, as numerous challenges and open problems remain to be addressed. Throughout this thesis, we have delved into some of these problems while also laying the groundwork for further exploration. Although it is impossible to cover the vast array of captivating research directions within this conclusion, there are several areas that warrant particular interest and offer intriguing possibilities for future investigations.

Firstly, our work on poisoning attacks within the realm of ranking data opens up exciting avenues for practical and prominent applications. The study of robustness can be extended to tackle typical problems encountered in the field, such as top-k ranking or information retrieval. However, it is crucial to carefully consider the computational challenges associated with real-world applications, where dealing with billions of items becomes a necessity. In particular, the evaluation of robustness (via a more efficient approximation of the breakdown function for example) seems necessary to the development of the field. While our thesis provides a preliminary exploration of this field, it is our hope that subsequent studies will delve deeper into this dimension and offer additional insights, notably on the conditions under which some statistics may be better than others. Furthermore, our approach to robustness in non-traditional data spaces, such as ranking data, holds the potential to benefit other types of extensively studied data, including graphs and time series.

Secondly, in the context of evasion attacks on deep learning models, significant efforts are still required to gain a comprehensive understanding of this phenomenon. There is a compelling opportunity to further explore the training phase of neural networks and investigate how the decision boundaries evolve during the learning process. This intricate endeavor is closely intertwined with the challenge of solving the adversarial optimization problem and deriving consistent surrogate losses. One intriguing idea worth exploring is the utilization of sequences of losses instead of a singular loss function throughout the training phase. This approach could potentially offer novel insights and help to address the complexities associated with this problem.

Furthermore, an intriguing direction for the study of robustness lies in exploring its interplay with other crucial aspects of trustworthy machine learning, such as fairness, privacy, interpretability, and domain generalization. While these fields share common goals in building reliable and ethical machine learning systems, their interaction and integration have not been extensively explored, and they become more and more strategic nowadays. For example, investigating how robustness and interpretability intersect can contribute to the development of transparent and trustworthy models. Robustness considerations

can guide the identification of influential features and decision-making factors, enabling models to provide explanations that align with their resilient behavior. This integration can enhance the interpretability of models and in return helps provide guidance to understand vulnerability issues. A nice example comes from Cantareira et al. (2021), which uses visualization techniques to identify vulnerable layers or neurons in neuraal networks.

Additionally, exploring the robustness of machine learning models in the context of multimodal data is an intriguing research direction. With the increasing prevalence of multimodal data, such as combining text and images, understanding the robustness of models to attacks across different modalities becomes crucial. Strategies such as developing robust fusion techniques, and investigating the vulnerabilities and defense mechanisms specific to such data combinations could provide insightful information to enhance the robustness of today's popular models.

This interdisciplinary approach holds great potential to advance the field of trustworthy machine learning, paving the way for the development of more robust, ethical, and reliable AI systems that address real-world challenges while preserving critical values and principles.

Bibliography

Agarwal, A., Agarwal, S., Khanna, S., and Patil, P. (2020). Rank aggregation from pairwise comparisons in the presence of adversarial corruptions. In *International Conference on Machine Learning*, pages 85–95. PMLR. (Cited on page vii, 4, 33, 70)

Agarwal, C., Nguyen, A., and Schonfeld, D. (2019). Improving robustness to adversarial examples by encouraging discriminative features. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3801–3505. IEEE. (Cited on page 100)

Aigrain, J. and Detyniecki, M. (2019). Detecting adversarial examples and other misclassifications in neural networks by introspection. *ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning*. (Cited on page 100)

Akhtar, N. and Mian, A. S. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430. (Cited on page 18)

Almgren, K., Kim, M., and Lee, J. (2017). Mining social media data using topological data analysis. In *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 144–153. (Cited on page 106)

Alvo, M. and Yu, P. L. H. (2014). *Statistical Methods for Ranking Data*. Springer-Verlag, New York. (Cited on page 68)

Amézquita, E. J., Quigley, M. Y., Ophelders, T., Munch, E., and Chitwood, D. H. (2020). The shape of things to come: Topological data analysis and biology, from molecules to organisms. *Developmental Dynamics*, 249(7):816–833. (Cited on page 106)

Athalye, A., Carlini, N., and Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR. (Cited on page 96)

Awasthi, P., Frank, N., Mao, A., Mohri, M., and Zhong, Y. (2021). Calibration and consistency of adversarial surrogate losses. *Advances in Neural Information Processing Systems*, 34:9804–9815. (Cited on page 99)

Azé, D. and Corvellec, J.-N. (2017). Nonlinear error bounds via a change of function. *Journal of Optimization Theory and Applications*, 172. (Cited on page 142)

Baccour, E., Mhaisen, N., Abdellatif, A., Erbad, A., Mohamed, A., Hamdi, M., and Guizani, M. (2022). Pervasive ai for iot applications: A survey on resource-efficient distributed artificial intelligence. *IEEE Communications Surveys and Tutorials*, 24(4):2366–2418. (Cited on page 93)

Bachmaier, C., Brandenburg, F. J., Gleißner, A., and Hofmeier, A. (2015). On the hardness of maximum rank aggregation problems. *Journal of Discrete Algorithms*, 31:2–13. 24th International Workshop on Combinatorial Algorithms (IWOCA 2013). (Cited on page 29)

Bao, H., Scott, C., and Sugiyama, M. (2020). Calibrated surrogate losses for adversarially robust classification. In *Conference on Learning Theory*, pages 408–451. PMLR. (Cited on page 99)

Bartholdi III, J. J., Tovey, C. A., and Trick, M. A. (1989). The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241. (Cited on page 39, 66)

Bartlett, P., Bubeck, S., and Cherapanamjeri, Y. (2021). Adversarial examples in multilayer random relu networks. *Advances in Neural Information Processing Systems*, 34. (Cited on page 142, 152, 153, 156)

Bartlett, P., Jordan, M., and McAuliffe, J. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156. (Cited on page 16)

Beksi, W. J. and Papanikolopoulos, N. (2019). A topology-based descriptor for 3d point cloud modeling: Theory and experiments. *Image and Vision Computing*, 88:84–95. (Cited on page 106)

Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424. (Cited on page vi, 4)

Bernstein, A., Burnaev, E., Sharaev, M., Kondrateva, E., and Kachan, O. (2020). Topological data analysis in computer vision. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, pages 673–679. SPIE. (Cited on page 106)

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345. (Cited on page 29, 70)

Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D. (2016). *Handbook of computational social choice*. Cambridge University Press. (Cited on page 66)

Brendel, W., Rauber, J., and Bethge, M. (2017). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *6th International Conference on Learning Representations (ICLR 2018)*. (Cited on page 95, 105, 121, 134)

Brunel, V. E. (2019). Concentration of the empirical level sets of tukey's halfspace depth. *Probability Theory and Relative Fields*, 173:1165–1196. (Cited on page 38)

Bubeck, S., Cherapanamjeri, Y., Gidel, G., and des Combes, R. T. (2021). A single gradient step finds adversarial examples on random two-layers neural networks. In *Advances in Neural Information Processing Systems*. (Cited on page 142)

Bubeck, S., Lee, Y. T., Price, E., and Razenshteyn, I. (2019). Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pages 831–840. PMLR. (Cited on page 21, 97)

Burr, M. A. and Fabrizio, R. J. (2017). Uniform convergence rates for halfspace depth. *Statistics and Probability Letters*, 124:33–40. (Cited on page 38)

Busa-Fekete, R., Fotakis, D., Szörényi, B., and Zampetakis, M. (2019). Optimal Learning of Mallows Block Model. In *Conference on Learning Theory (COLT)*.

Busa-Fekete, R., Hüllermeier, E., and Szörényi, B. (2014). Preference-based rank elicitation using statistical models: the case of Mallows. In *Proceedings of International Conference on Machine Learning (ICML) 2014*, pages 1071–1079. (Cited on page 49)

Cabral Costa, J., Roxo, T., Proença, H., and Inácio, P. (2023). How deep learning sees the world: A survey on adversarial attacks and defenses. *ArXiv*. (Cited on page 18)

Calauzènes, C., Usunier, N., and Gallinari, P. (2013). Calibration and regret bounds for order-preserving surrogate losses in learning to rank. *Machine Learning*, 93(2):227–260. (Cited on page 31)

Cantareira, G. D., Mello, R. F., and Paulovich, F. V. (2021). Explainable adversarial attacks in deep neural networks using activation profiles. *arXiv preprint arXiv:2103.10229*. (Cited on page 100, 165)

Caragiannis, I., Procaccia, A. D., and Shah, N. (2013). When do noisy votes reveal the truth? In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, pages 143–160, New York. ACM. (Cited on page 57, 69, 80)

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *IEEE Symposium on SP*. (Cited on page vii, 5, 96, 105, 121)

Carriere, M., Cuturi, M., and Oudot, S. (2017). Sliced wasserstein kernel for persistence diagrams. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 664–673. PMLR. (Cited on page 122)

Carrière, M., Oudot, S. Y., and Ovsjanikov, M. (2015). Stable topological signatures for points on 3d shapes. In *Computer Graphics Forum*. (Cited on page 106)

Carterette, B. (2009). On rank correlation and the distance between rankings. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 436–443. Association for Computing Machinery. (Cited on page 27)

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D. (2022). Adversarial attacks and defences: A survey. *Electronics*, 11(8). (Cited on page 18)

Chan, J. M., Carlsson, G., and Rabadan, R. (2013). Topology of viral evolution. *Proceedings of the National Academy of Sciences*, 110(46):18566–18571. (Cited on page 106)

Chaudhuri, P. (1996). On a geometric notion of quantiles for multivariate data. *Journal of the American Statistical Association*, 91(434):862–872. (Cited on page 38)

Chazal, F. and Michel, B. (2017). An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4. (Cited on page 111)

Chen, J., Jordan, M. I., and Wainwright, M. J. (2020a). Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE. (Cited on page 21, 98, 134)

Chen, K., Zhu, H., Yan, L., and Wang, J. (2020b). A survey on adversarial examples in deep learning. *Journal on Big Data*, 2:71–84. (Cited on page 18)

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26.

Chen, S., He, Z., Sun, C., Yang, J., and Huang, X. (2022). Universal adversarial attack on attention and the resulting dataset damagenet. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2188–2197. (Cited on page 5)

Chernozhukov, V., Galichon, A., Hallin, M., and Henry, M. (2017). Monge–kantorovich depth, quantiles, ranks and signs. *The Annals of Statistics*, 45(1):223–256. (Cited on page 38)

Collas, F. and Irurozki, E. (2021). Concentric mixtures of Mallows models for top-k rankings: sampling and identifiability. In *International Conference on Machine Learning (ICML)*. (Cited on page 57)

Collins, A., Zomorodian, A., Carlsson, G., and Guibas, L. J. (2004). A barcode shape descriptor for curve point cloud data. *Computers & Graphics*, 28(6):881–894. (Cited on page 106)

Condorcet, N. (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'Imprimerie Royale, Paris. (Cited on page 30)

Coppersmith, D., Fleischer, L. K., and Rurda, A. (2010). Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Trans. Algorithms*, 6:1–13. (Cited on page 69)

Critchlow, D. E. (2012). *Metric methods for analyzing partially ranked data*, volume 34. Springer Science & Business Media. (Cited on page 78)

Critchlow, D. E., Fligner, M. A., and Verducci, J. S. (1991). Probability models on rankings. *Journal of Mathematical Psychology*, 35(3):294–318. (Cited on page 63)

Cubuk, E. D., Zoph, B., Schoenholz, S. S., and Le, Q. V. (2017). Intriguing properties of adversarial examples. *Workshop of the 6th International Conference on Learning Representations (ICLR 2018)*. (Cited on page 5)

Daniely, A. and Shacham, H. (2020). Most relu networks suffer from \ell^2 adversarial perturbations. In *Advances in Neural Information Processing Systems*, volume 33, pages 6629–6636. Curran Associates, Inc. (Cited on page 142, 152, 153, 156)

Datar, A., Rajkumar, A., and Augustine, J. (2022). Byzantine spectral ranking. In *International Conference on Neural Information Processing Systems (NeurIPS)*. (Cited on page 70)

Davenport, A. and Lovell, D. (2005). Ranking pilots in aerobatic flight competitions. Technical report, IBM Research Report RC23631 (W0506-079), TJ Watson Research Center, NY.

Davidson, D. and Marschak, J. (1959). Experimental tests of a stochastic decision theory. In Churchman, C. W. and Ratoosh, P., editors, *Measurement: Definitions and Theories*, pages 233–269. John Wiley. (Cited on page 32)

Davies, J., Katsirelos, G., Narodytska, N., and Walsh, T. (2011). Complexity of and algorithms for borda manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. (Cited on page 66)

de Borda, J.-C. (1781). Mémoire sur les élections au scrutin.

De Condorcet, N. et al. (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Cambridge University Press.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee. (Cited on page 16)

Desarkar, M. S., Sarkar, S., and Mitra, P. (2016). Preference relations based unsupervised rank aggregation for metasearch. *Expert Systems with Applications*, 49:86–98.

Deza, M. and Deza, E. (2009). *Encyclopedia of Distances*. Springer. (Cited on page 68)

Deza, M. and Huang, T. (1998). Metrics on permutations, a survey. *Journal of Combinatorics, Information and System Sciences*. (Cited on page 39)

Diaconis, P. (1988). Group representations in probability and statistics. *Lecture notes-monograph series*, 11:i–192. (Cited on page 13)

Diaconis, P. (1989). A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, pages 949–979. (Cited on page 13)

Diaconis, P. and Graham, R. L. (1977). Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268. (Cited on page 39)

Diakonikolas, I., Hopkins, S. B., Kane, D., and Karmalkar, S. (2020). Robustly learning any clusterable mixture of gaussians. *arXiv preprint arXiv:2005.06417*. (Cited on page 4)

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. (2018). Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2683–2702. SIAM. (Cited on page 4)

Dohmatob, E. (2019). Generalized no free lunch theorem for adversarial robustness. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*. PMLR. (Cited on page vii, 5, 21, 97, 136, 138)

Dohmatob, E., Guo, C., and Goibert, M. (2023). Origins of low-dimensional adversarial perturbations. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, pages 9221–9237. PMLR. (Cited on page xiv, 102)

Donoho, D. L. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *The Annals of Statistics*, 20:1803–1827. (Cited on page 38)

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001a). Rank aggregation methods for the web. In *Rank aggregation methods for the Web*, pages 613–622. ACM. (Cited on page 30, 69)

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001b). Rank aggregation methods for the Web. In *International Conference on World Wide Web*, pages 613–622, New York. ACM. (Cited on page 31, 57)

Dyckerhoff, R. (2004). Data depths satisfying the projection property. *Allgemeines Statistisches Archiv*, 88(2):163–190. (Cited on page 37)

Dziugaite, G. K., Ghahramani, Z., and Roy, D. M. (2016). A study of the effect of jpg compression on adversarial images. *International Society for Bayesian Analysis (ISBA 2016) World Meeting*, abs/1608.00853. (Cited on page 96)

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2000). Topological persistence and simplification. In *IEEE FOCS*. (Cited on page 106)

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634. (Cited on page vi, 2, 3)

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., and Vee, E. (2006). Comparing partial rankings. *SIAM Journal on Discrete Mathematics*, 20(3):628–648. (Cited on page 77, 78)

Fawzi, A., Fawzi, H., and Fawzi, O. (2018a). Adversarial vulnerability for any classifier. *Advances in neural information processing systems*, 31. (Cited on page 5, 21, 97, 138)

Fawzi, A., Fawzi, O., and Frossard, P. (2018b). Analysis of classifiers' robustness to adversarial perturbations. *Machine learning*, 107(3):481–508. (Cited on page vii, 21, 97)

Fawzi, A., Moosavi-Dezfooli, S.-M., and Frossard, P. (2016). Robustness of classifiers: from adversarial to random noise. *Advances in neural information processing systems*, 29. (Cited on page 99)

Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Soatto, S. (2018c). Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770. (Cited on page 99)

Fishburn, P. C. (1973). Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical psychology*, 10(4):327–352. (Cited on page 32)

Fligner, M. A. and Verducci, J. S. (1986). Distance based ranking models. *Journal of the Royal Statistical Society*, 48(3):359–369.

Fligner, M. A. and Verducci, J. S. (1988). Multistage ranking models. *Journal of the American Statistical Association*, 83(403):892–901. (Cited on page 57)

Ford, N., Gilmer, J., Carlini, N., and Cubuk, D. (2019). Adversarial examples are a natural consequence of test error in noise. *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*. (Cited on page 21, 97)

Fox, J. and Weisberg, S. (2002). Robust regression. *An R and S-Plus companion to applied regression*, 91:6. (Cited on page vi, 4)

Frankle, J. and Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *7th International Conference on Learning Representations (ICLR 2019)*. (Cited on page 111, 118, 126)

Gao, J., Wang, B., Lin, Z., Xu, W., and Qi, Y. (2017). Deepcloak: Masking deep neural network models for robustness against adversarial samples. *Workshop at the 5th International Conference on Learning Representations (ICLR 2017)*. (Cited on page 96)

Gebhart, T. and Schrater, P. (2019). Adversarial examples target topological holes in deep networks. *arXiv preprint arXiv:1901.09496*.

Gebhart, T., Schrater, P., and Hylton, A. (2019). Characterizing the shape of activation space in deep neural networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1537–1542. IEEE. (Cited on page 107)

Geirhos, R., Janssen, D. H. J., Schütt, H. H., Rauber, J., Bethge, M., and Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. *CoRR*, abs/1706.06969. (Cited on page 16)

Gholizadeh, S., Seyeditabari, A., and Zadrozny, W. (2018). Topological signature of 19th century novelists: Persistent homology in text mining. *big data and cognitive computing*, 2(4):33. (Cited on page 106)

Gibbard, A. et al. (1973). Manipulation of voting schemes: a general result. *Econometrica*, 41:587–601. (Cited on page 66)

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. J. (2018). Adversarial spheres. *CoRR*, abs/1801.02774. (Cited on page 136)

Goibert, M., Calauzènes, C., Irurozki, E., and Clémençon, S. (2023). Robust consensus in ranking data analysis: Definitions, properties and computational issues. *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*. (Cited on page x, xi, 34)

Goibert, M., Clémençon, S., Irurozki, E., and Mozharovskyi, P. (2022a). Statistical Depth Functions for Ranking Distributions: Definitions, Statistical Learning and Applications. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*. (Cited on page x, 34)

Goibert, M., Ricatte, T., and Dohmatob, E. (2022b). An adversarial robustness perspective on the topology of neural networks. *ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. (Cited on page xiv, 102)

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations (ICLR 2015)*. (Cited on page vii, 5, 17, 19, 20, 96, 98, 99)

Guo, C. (2020). *Phd thesis: Threats and Countermeasures in Machine Learning Applications*. Cornell University. (Cited on page 135, 136, 137, 138, 141, 148, 149)

Guo, C., Frank, J. S., and Weinberger, K. Q. (2018a). Low frequency adversarial perturbation. *Conference on Uncertainty in Artificial Intelligence*. (Cited on page 21, 98, 134, 137, 156, 158)

Guo, C., Gardner, J., You, Y., Wilson, A. G., and Weinberger, K. (2019). Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493. PMLR. (Cited on page 5, 95, 134, 135, 137)

Guo, C., Rana, M., Cisse, M., and Maaten, L. (2018b). Countering adversarial images using input transformations. *6th International Conference on Learning Representations (ICLR 2018)*. (Cited on page 96)

Guo, M., Yang, Y., Xu, R., Liu, Z., and Lin, D. (2020). When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 631–640. (Cited on page 99)

Guo, Y., Zhang, C., Zhang, C., and Chen, Y. (2018c). Sparse dnns with improved adversarial robustness. In *NeurIPS 2018*, volume 31. (Cited on page 126)

Han, S., Lin, C., Shen, C., Wang, Q., and Guan, X. (2023). Interpreting adversarial examples in deep learning: A review. *ACM Comput. Surv.* Just Accepted. (Cited on page 18, 98, 100)

Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *5th International Conference on Learning Representations (ICLR 2017)*. (Cited on page vii, 5)

Hiraoka, Y., Nakamura, T., Hirata, A., Escolar, E. G., Matsue, K., and Nishiura, Y. (2016). Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040. (Cited on page 106)

Hu, C.-S. and Chung, Y.-M. (2021). A sheaf and topology approach to detecting local merging relations in digital images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4396–4405. (Cited on page 106)

Huang, H., Wang, Y., Erfani, S., Gu, Q., Bailey, J., and Ma, X. (2021). Exploring architectural ingredients of adversarially robust deep neural networks. *Advances in Neural Information Processing Systems*, 34:5545–5559. (Cited on page 99)

Huang, T.-K., Weng, R. C., and Lin, C.-J. (2006). Generalized bradley-terry models and multi-class probability estimates. *The Journal of Machine Learning Research*, 7:85–115.

Huang, Z. and Zhang, T. (2019). Black-box adversarial attack with transferable model-based embedding. *8th International Conference on Learning Representations (ICLR 2020)*. (Cited on page 21, 98, 134, 137)

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101. (Cited on page vi, 3, 4, 69)

Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics*. 2nd edition, John Wiley & Sons. (Cited on page vi, 3, 8, 9, 68)

Hubert, M. and Branden, K. V. (2003). Robust methods for partial least squares regression. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(10):537–549. (Cited on page 4)

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005). Robpca: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79. (Cited on page 4)

Hudry, O. (2008). NP-hardness results for the aggregation of linear orders into median orders. *Annals of Operations Research*, 163:63–88. (Cited on page 39, 52)

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. (2018). Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR. (Cited on page 134)

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*. (Cited on page 5, 100, 108, 109)

Irurozki, E., Calvo, B., and Lozano, J. (2019). Mallows and generalized Mallows model for matchings. *Bernoulli*, 25(2).

Jakubovitz, D. and Giryes, R. (2018). Improving dnn robustness to adversarial attacks using jacobian regularization. In *ECCV*. (Cited on page 126, 127)

Järvelin, K. and Kekäläinen, J. (2000). Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 41–48, New York, NY, USA. Association for Computing Machinery. (Cited on page 27)

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. (2019). Predicting the generalization gap in deep networks with margin distributions. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net. (Cited on page 156)

Jiao, Y., Korba, A., and Sibony, E. (2016). Controlling the distance to a kemeny consensus without computing it. In *Proceedings of the International Conference on Machine Learning (ICML)*. (Cited on page 69)

Jin, T., Xu, P., Gu, Q., and Farnoud, F. (2018). Rank aggregation via heterogeneous thurstone preference models. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. (Cited on page 70)

Johnson, R. W. (2001). An introduction to the bootstrap. *Teaching Statistics*. (Cited on page 122)

Kemeny, J. G. (1959). Mathematics without numbers. *Daedalus*, 88:571–591.

Khrulkov, V. and Oseledets, I. (2018). Art of singular vectors and universal adversarial perturbations. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8562–8570. (Cited on page 134, 135, 137, 141, 157)

Kool, W., Van Hoof, H., and Welling, M. (2019). Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pages 3499–3508. PMLR. (Cited on page 29)

Korba, A., Clémençon, S., and Sibony, E. (2017). A learning theory of ranking aggregation. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, (AISTATS)*. (Cited on page 31, 32, 39, 46, 47, 49, 52, 68)

Koshevoy, G. and Mosler, K. (1997). Zonoid trimming for multivariate distributions. *The Annals of Statistics*, 25(5):1998–2017. (Cited on page 38)

Kumar, R. and Vassilvitskii, S. (2010). Generalized distances between rankings. In *The Web Conference*. (Cited on page 27)

Kurakin, A., Goodfellow, I., and Bengio, S. (2017). Adversarial machine learning at scale. *5th International Conference on Learning Representations (ICLR 2017)*. (Cited on page 105, 120, 121)

Lafaye De Micheaux, P., Mozharovskyi, P., and Vimond, M. (2020). Depth for curve data and applications. *Journal of the American Statistical Association*. in press. (Cited on page 63)

Lebanon, G. and Lafferty, J. (2002). Cranking: Combining rankings using conditional probability models on permutations. In *Proceedings of the 19th International Conference on Machine Learning*, pages 363–370.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551. (Cited on page 93)

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database. *NA*. (Cited on page 5, 156)

Lee, K., Lee, K., Lee, H., and Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*. (Cited on page 5, 96, 108, 123, 124)

Lerasle, M., Szabo, Z., Mathieu, T., and Lecué, G. (2019). Monk – outlier-robust mean embedding estimation by median-of-means. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Li, C., Ovsjanikov, M., and Chazal, F. (2014). Persistence-based structural recognition. In *IEEE CVPR*. (Cited on page 106)

Li, H., Fan, Y., Ganz, F., Yezzi, A., and Barnaghi, P. (2021a). Verifying the causes of adversarial examples. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6750–6757. IEEE. (Cited on page 99)

Li, H., Li, G., and Yu, Y. (2019). Rosa: Robust salient object detection against adversarial attacks. *IEEE transactions on cybernetics*, 50(11):4835–4847. (Cited on page 5)

Li, Y., Yang, Z., Wang, Y., and Xu, C. (2021b). Neural architecture dilation for adversarial robustness. *Advances in Neural Information Processing Systems*, 34:29578–29589. (Cited on page 99)

Liang, Y. and Samavi, R. (2023). Advanced defensive distillation with ensemble voting and noisy logits. *Applied Intelligence*, 53(3):3069–3094. (Cited on page 5)

Liu (1990). On a notion of data depth based upon random simplices. *The Annals of Statistics*, 18(1):405–414. (Cited on page 38)

Liu, A. and Moitra, A. (2018). Efficiently learning mixtures of mallows models. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 627–638. IEEE.

Liu, A., Zhao, Z., Liao, C., Lu, P., and Xia, L. (2019). Learning Plackett-Luce Mixtures from Partial Preferences. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Liu, B. and Shen, M. (2022). Some geometrical and topological properties of dnns' decision boundaries. *Theoretical Computer Science*, 908:64–75. (Cited on page 110)

Liu, R. Y., Parelius, J. M., and Singh, K. (1999). Multivariate analysis by data depth: descriptive statistics, graphics and inference. *The Annals of Statistics*, 27(3):783–858. With discussion and a rejoinder by Liu and Singh. (Cited on page 62)

Liu, R. Y. and Singh, K. (1993). A quality index based on data depth and multivariate rank tests. *Journal of the American Statistical Association*, 88(421):252–260. (Cited on page 38, 63)

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. (2018). Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385. (Cited on page 96)

Lu, T. and Boutilier, C. (2014). Effective Sampling and Learning for Mallows Models with Pairwise-Preference Data. *Journal of Machine Learning Research*.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical analysis*. Wiley, New York, NY, USA. (Cited on page 29, 70)

Lugosi, G. and Mendelson, S. (2019). Risk minimization by median-of-means tournaments. *Journal of the European Mathematical Society*.

Ma, A., Faghri, F., Papernot, N., and Farahmand, A.-m. (2020). Soar: Second-order adversarial regularization. *arXiv preprint arXiv:2004.01832*. (Cited on page 96)

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. (2018). Characterizing adversarial subspaces using local intrinsic dimensionality. *6th International Conference on Learning Representations (ICLR 2018)*. (Cited on page vii, 5, 96, 108, 124)

Ma, Y., Genton, M. G., and Parzen, E. (2011). Asymptotic properties of sample quantiles of discrete distributions. *Annals of the Institute of Statistical Mathematics*, 63(2):227–243. (Cited on page 50)

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *6th International Conference on Learning Representations (ICLR 2018)*. (Cited on page vii, 5, 96, 156, 157)

Mahloujifar, S., Diochnos, D. I., and Mahmoody, M. (2019). The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4536–4543. (Cited on page 21, 97, 136, 138)

Mallows, C. L. (1957a). Non-null ranking models. *Biometrika*, 44:114–130. (Cited on page 29)

Mallows, C. L. (1957b). Non-null ranking models. *Biometrika*, 44(1-2):114–130. (Cited on page 49)

Manoj, N. S. and Blum, A. (2021). Excess capacity and backdoor poisoning. *NeurIPS*. (Cited on page 5, 108, 110)

Mao, A., Procaccia, A., and Chen, Y. (2013). Better human computation through principled voting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27.

Melamed, O., Yehudai, G., and Vardi, G. (2023). Adversarial examples exist in two-layer relu networks for low dimensional data manifolds. *arXiv preprint arXiv:2303.00783*. (Cited on page 21, 97)

Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. (2017). On detecting adversarial perturbations. *5th International Conference on Learning Representations (ICLR 2017)*. (Cited on page 96)

Meunier, L., Ettedgui, R., Pinot, R., Chevaleyre, Y., and Atif, J. (2022). Towards consistency in adversarial classification. *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. (Cited on page 99)

Møller, S. F., von Frese, J., and Bro, R. (2005). Robust methods for multivariate data analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 19(10):549–563. (Cited on page vii, 4)

Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., and Frossard, P. (2017). Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94. (Cited on page 21, 134, 135, 137, 141, 156, 157)

Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., Frossard, P., and Soatto, S. (2018). Analysis of universal adversarial perturbations. *CVPR 2017*, abs/1705.09554. (Cited on page 5, 135, 141)

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. (2017). Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Cited on page 95, 98)

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582. (Cited on page vii, 5)

Moosavi-Dezfooli, S.-M., Fawzi, A., Uesato, J., and Frossard, P. (2019). Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9078–9086. (Cited on page 5, 99)

Morozov, D. (2017). Dionysus. (Cited on page 118)

Mosler, K. (2013). Depth statistics. In Becker, C., Fried, R., and Kuhnt, S., editors, *Robustness and Complex Data Structures: Festschrift in Honour of Ursula Gather*, pages 17–34. Springer. (Cited on page 37)

Mustafa, A., Khan, S., Hayat, M., Goecke, R., Shen, J., and Shao, L. (2019). Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3385–3394. (Cited on page 100)

Naitzat, G., Zhitnikov, A., and Lim, L.-H. (2020). Topology of deep neural networks. *The Journal of Machine Learning Research*, 21(1):7503–7542. (Cited on page 107)

Nakkiran, P. (2019). A discussion of 'adversarial examples are not bugs, they are features': Adversarial examples are just bugs, too. *Distill*. (Cited on page 100, 108, 109)

Nar, K., Ocal, O., Sastry, S. S., and Ramchandran, K. (2019). Cross-entropy loss and low-rank features have responsibility for adversarial examples. *arXiv preprint arXiv:1901.08360*. (Cited on page 99)

Nemirovski, A. and Rubinstein, R. Y. (2002). *An Efficient Stochastic Approximation Algorithm for Stochastic Saddle Point Problems*, pages 156–184. Springer, New York, NY. (Cited on page 79)

Oja, H. (1983). Descriptive statistics for multivariate distributions. *Statistics and Probability Letters*, 1(6):327–332. (Cited on page 38)

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. (2021). Bag of tricks for adversarial training. *9th International Conference on Learning Representations (ICLR 2021)*. (Cited on page 5)

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. (Cited on page 5)

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE. (Cited on page vii, 5, 96)

Patel, T., Telesca, D., Rallo, R., George, S., Xia, T., and Nel, A. E. (2013). Hierarchical rank aggregation with applications to nnanotoxicology. *Journal of Agricultural, Biological, and Environmental Statistics*, 18(2):159–177.

Plackett, R. L. (1975). The Analysis of Permutations. *Journal of the Royal Statistical Society Series C*, 24(2):193–202. (Cited on page 29)

Procaccia, A. and Shah, N. (2016). Optimal aggregation of uncertain preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 608–614.

Pun, C. S., Xia, K., and Lee, S. X. (2018). Persistent-homology-based machine learning and its applications–a survey. *arXiv preprint arXiv:1811.00252*.

Qiu, Y., Leng, J., Guo, C., Chen, Q., Li, C., Guo, M., and Zhu, Y. (2019). Adversarial defense through network profiling based path extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4777–4786. (Cited on page 100)

Rahimi, A. and Recht, B. (2008). Uniform approximation of functions with random bases. In *IEEE, Allerton 2008*. (Cited on page 152)

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NIPS 2009*. (Cited on page 152)

Rahnama, A., Nguyen, A. T., and Raff, E. (2020). Robust design of deep neural networks against adversarial attacks based on lyapunov theory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8178–8187.

Rice, L., Wong, E., and Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In *ICML*. (Cited on page 5, 108, 110)

Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., and Borgwardt, K. (2018). Neural persistence: A complexity measure for deep neural networks using algebraic topology. *7th International Conference on Learning Representations (ICLR 2019)*.

Ross, A. and Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. (Cited on page 96)

Rousseeuw, P. and Leroy, A. (1987). *Robust regression and outlier detection*. Wiley Series in probability and mathematical statistics. Wiley, New York [u.a.]. (Cited on page 4)

Rozsa, A., Günther, M., and Boult, T. E. (2016). Are accuracy and robustness correlated. In *2016 15th IEEE international conference on machine learning and applications (ICMLA)*, pages 227–232. IEEE. (Cited on page 5, 99)

Samangouei, P., Kabkab, M., and Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. *6th International Conference on Learning Representations (ICLR 2018)*. (Cited on page 96)

Satterthwaite, M. A. (1975). Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of economic theory*, 10:187–217. (Cited on page 66)

Serrano, D. H. and Gómez, D. S. (2020). Centrality measures in simplicial complexes: Applications of topological data analysis to network science. *Applied Mathematics and Computation*, 382:125331. (Cited on page 106)

Shafahi, A., Huang, W. R., Studer, C., Feizi, S., and Goldstein, T. (2019a). Are adversarial examples inevitable? *7th International Conference on Learning Representations (ICLR 2019)*. (Cited on page vii, 136, 138)

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. (2019b). Adversarial training for free! *Advances in Neural Information Processing Systems*, 32. (Cited on page 5, 96, 99)

Shah, N. B., Balakrishnan, S., Guntuboyina, A., and Wainright, M. J. (2015). Stochastically transitive models for pairwise comparisons: statistical and computational issues. *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*. (Cited on page 32)

Shah, N. B. and Wainwright, M. J. (2018). Simple, robust and optimal ranking from pairwise comparisons. (Cited on page 32)

Sheather, S. J. and Marron, J. S. (1990). Kernel quantile estimators. *Journal of the American Statistical Association*, 85(410):410–416. (Cited on page 50)

Sitawarin, C., Chakraborty, S., and Wagner, D. (2021). Sat: Improving adversarial training via curriculum-based loss smoothing. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pages 25–36. (Cited on page 99)

Skaf, Y. and Laubenbacher, R. (2022). Topological data analysis in biomedicine: A review. *Journal of Biomedical Informatics*, page 104082. (Cited on page 106)

Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *IEEE WACV*. (Cited on page 121)

Sokolić, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. (2017). Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*. (Cited on page 127)

Steinwart, I. (2007). How to compare different loss functions and their risks. *Constructive Approximation*, 26(2):225–287. (Cited on page 8)

Stutz, D., Hein, M., and Schiele, B. (2019). Disentangling adversarial robustness and generalization. In *IEEE CVPR*. (Cited on page 100, 108, 109, 110)

Su, J., Vargas, D. V., and Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841. (Cited on page 21, 95)

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*. (Cited on page vi, vii, xii, xiii, 3, 4, 17, 19, 20, 93, 95, 160)

Taghanaki, S. A., Abhishek, K., Azizi, S., and Hamarneh, G. (2019). A kernelized manifold mapping to diminish the effect of adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11340–11349. (Cited on page 99)

Tanay, T. and Griffin, L. (2016). A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*. (Cited on page 5, 99)

Taylor, D., Klimm, F., Harrington, H. A., Kramár, M., Mischaikow, K., Porter, M. A., and Mucha, P. J. (2015). Topological data analysis of contagion maps for examining spreading processes on networks. *Nature communications*, 6(1):7723. (Cited on page 106)

Thurstone, L. L. (1927). A law of comparative judgement. *Psychological Review*, 34:278–286. (Cited on page 29, 70)

Thurstone, L. L. (1931). Rank order as a psycho-physical method. *Journal of Experimental Psychology*, 14:187–201. (Cited on page 29)

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2019). Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, volume abs/1805.12152. (Cited on page vii, 5, 136)

Tu, C.-C., Ting, P., Chen, P.-Y., Liu, S., Zhang, H., Yi, J., Hsieh, C.-J., and Cheng, S.-M. (2019). Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749. (Cited on page 21, 98, 134, 137)

Tukey, J. W. (1975). Mathematics and the picturing of data. In James, R. D., editor, *Proceedings of the International Congress of Mathematicians*, volume 2, pages 523–531. Canadian Mathematical Congress. (Cited on page 38, 43)

Turner, K., Mukherjee, S., and Boyer, D. M. (2014). Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*. (Cited on page 106)

Umeda, Y. (2017). Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239. (Cited on page 106)

van der Vaart, A. (1998). *Asymptotic Statistics*. Cambridge University Press. (Cited on page 68)

Vardi, Y. and Zhang, C.-H. (2000). The multivariate l_1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426. (Cited on page 38)

Vitelli, V., Sørensen, Ø., Crispino, M., Frigessi, A., and Arjas, E. (2018). Probabilistic preference learning with the Mallows rank model. *Journal of Machine Learning Research*, 18(1).

Wang, L., Ding, G. W., Huang, R., Cao, Y., and Lui, Y. C. (2018a). Adversarial robustness of pruned neural networks. *ArxIv preprint*. (Cited on page 126)

Wang, S., Liao, N., Xiang, L., Ye, N., and Zhang, Q. (2020a). Achieving adversarial robustness via sparsity. *arXiv preprint*. (Cited on page 126)

Wang, S., Wang, X., Zhao, P., Wen, W., Kaeli, D., Chin, P., and Lin, X. (2018b). Defensive dropout for hardening deep neural networks under adversarial attacks. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE. (Cited on page 96)

Wang, Y., Ullah, E., Mianjy, P., and Arora, R. (2022). Adversarial robustness is at odds with lazy training. In *NeurIPS 2022*. NeurIPS. (Cited on page 154, 156)

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. (2020b). Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*. (Cited on page 99)

Wong, E., Rice, L., and Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. *8th International Conference on Learning Representations (ICLR 2020)*. (Cited on page 99)

Wu, B., Chen, J., Cai, D., He, X., and Gu, Q. (2021). Do wider neural networks really help adversarial robustness? In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*. (Cited on page 5, 108, 110)

Wu, C. and Hargreaves, C. A. (2021). Topological machine learning for mixed numeric and categorical data. *International Journal on Artificial Intelligence Tools*, 30(05):2150025. (Cited on page 106)

Wu, D., Wang, Y., Xia, S.-T., Bailey, J., and Ma, X. (2020). Skip connections matter: On the transferability of adversarial examples generated with resnets. *8th International Conference on Learning Representations (ICLR 2020)*. (Cited on page 99)

Xu, K., Liu, S., Zhang, G., Sun, M., Zhao, P., Fan, Q., Gan, C., and Lin, X. (2019). Interpreting adversarial examples by activation promotion and suppression. *arXiv preprint*. (Cited on page 107, 109)

Xu, W., Evans, D., and Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. *Network and Distributed Systems Security Symposium (NDSS) 2018*. (Cited on page 96)

Xu, X., Drougard, N., and Roy, R. N. (2021). Topological data analysis as a new tool for eeg processing. *Frontiers in Neuroscience*, 15:761703. (Cited on page 106)

Yamanashi, T., Kajitani, M., Iwata, M., Crutchley, K. J., Marra, P., Malicoat, J. R., Williams, J. C., Leyden, L. R., Long, H., Lo, D., et al. (2021). Topological data analysis (tda) enhances bispectral eeg (bseeg) algorithm for detection of delirium. *Scientific reports*, 11(1):304. (Cited on page 106)

Yan, Z., Guo, Y., and Zhang, C. (2019). Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. (Cited on page 21, 98, 134, 137)

Yilmaz, E., Aslam, J. A., and Robertson, S. (2008). A new rank correlation coefficient for information retrieval. In Myaeng, S.-H., Oard, D. W., Sebastiani, F., Chua, T.-S., and Leong, M.-K., editors, *SIGIR*, pages 587–594. ACM. (Cited on page 27)

Yin, D., Lopes, R. G., Shlens, J., Cubuk, E. D., and Gilmer, J. (2019). A fourier perspective on model robustness in computer vision. *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. (Cited on page 134, 158)

Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B. (2019a). You only propagate once: Accelerating adversarial training via maximal principle. *Advances in Neural Information Processing Systems*, 32. (Cited on page 5, 96, 99)

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019b). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR. (Cited on page 99)

Zhang, M., Kalies, W. D., Kelso, J. S., and Tognoli, E. (2020). Topological portraits of multiscale coordination dynamics. *Journal of Neuroscience Methods*, 339:108672. (Cited on page 112)

Zhao, Y. and Zhang, H. (2021). Quantitative performance assessment of cnn units via topological entropy calculation. *10th International Conference on Learning Representations (ICLR 2022)*. (Cited on page 107)

Zhao, Z. and Xia, L. (2019). Learning Mixtures of Plackett-Luce Models from Structured Partial Orders. In *Advances in Neural Information Processing Systems*, pages 10143–10153.

Zheng, Z. and Hong, P. (2018). Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. *Advances in Neural Information Processing Systems*, 31. (Cited on page 100)

Zhou, H., Lan, J., Liu, R., and Yosinski, J. (2019). Deconstructing lottery tickets: Zeros, signs, and the supermask. In *NeurIPS*. (Cited on page 118)

Zia, A., Khamis, A., Nichols, J., Hayder, Z., Rolland, V., and Petersson, L. (2023). Topological deep learning: A review of an emerging paradigm. *arXiv preprint arXiv:2302.03836*. (Cited on page 107)

Zomorodian, A. and Carlsson, G. (2005). Computing persistent homology. *Discrete & Computational Geometry*. (Cited on page 106)

Zuckerman, M., Procaccia, A. D., and Rosenschein, J. S. (2009). Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412. (Cited on page 66)

Zuo, Y. (2006). Robust location and scatter estimators in multivariate analysis. *Frontiers In Statistics*, pages 467–490. (Cited on page 4)

Zuo, Y. and Serfling, R. (2000a). General notions of statistical depth function. *The Annals of Statistics*, 28(2):461–482. (Cited on page 37, 38, 40, 43)

Zuo, Y. and Serfling, R. (2000b). Structural properties and convergence results for contours of sample statistical depth functions. *The Annals of Statistics*, 28(2):483–499. (Cited on page 38)

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre: Compréhension Statistique de la Robustesse Adversaire

Mots clés: Apprentissage automatique, Robustesse, Apprentissage profond, Données de préférence, Réseaux de neurones

Résumé: Cette thèse se concentre sur la question de la robustesse en apprentissage automatique, en examinant spécifiquement deux types d'attaques : les attaques de contamination pendant l'apprentissage et les attaques d'évasion pendant l'inférence.
L'étude des attaques de contamination remonte aux années soixante et a été unifiée sous la théorie des statistiques robustes. Cependant, les recherches antérieures se sont principalement concentrées sur des types de données classiques, comme les nombres réels. Dans cette thèse, les statistiques robustes sont étendues aux données de classement, qui ne possèdent pas de structure d'espace vectoriel et ont une nature combinatoire. Les contributions de la thèse comprennent notamment un algorithme pour mesurer la robustesse des statistiques pour la tâche qui consiste à trouver un rang consensus dans un ensemble de données de rangs, ainsi que deux statistiques robustes pour résoudre ce même problème.
En revanche, depuis 2013, les attaques d'évasion ont suscité une attention considérable dans le domaine de l'apprentissage profond, en particulier pour la classification d'images. Malgré la prolifération des travaux de recherche sur les exemples adversaires, le problème reste difficile à analyser sur le plan théorique et manque d'unification. Pour remédier à cela, cette thèse apporte des contributions à la compréhension et à l'atténuation des attaques d'évasion. Ces contributions comprennent l'unification des caractéristiques des exemples adversaires grâce à l'étude des paramètres sous-optimisés et à la circulation de l'information au travers des réseaux de neurones, ainsi que l'établissement de bornes théoriques caractérisant le taux de succès des attaques, récemment créées, de faible dimension.

Title: Statistical Understanding of Adversarial Robustness

Keywords: Machine Learning, Robustness, Deep Learning, Rankings, Neural Networks

Abstract: This thesis focuses on the question of robustness in machine learning, specifically examining two types of attacks: poisoning attacks at training time and evasion attacks at inference time.
The study of poisoning attacks dates back to the sixties and has been unified under the theory of robust statistics. However, prior research was primarily focused on classical data types, mainly real-numbered data, limiting the applicability of poisoning attack studies. In this thesis, robust statistics are extended to ranking data, which lack a vector space structure and have a combinatorial nature. The work presented in this thesis initiates the study of robustness in the context of ranking data and provides a framework for future extensions. Contributions include a practical algorithm to measure the robustness of statistics for the task of *consensus ranking*, and two robust statistics to solve this task.
In contrast, since 2013, evasion attacks gained significant attention in the deep learning field, particularly for image classification. Despite the proliferation of research works on adversarial examples, the theoretical analysis of the problem remains challenging and it lacks unification. To address this matter, the thesis makes contributions to understanding and mitigating evasion attacks. These contributions involve the unification of adversarial examples' characteristics through the study of under-optimized edges and information flow within neural networks, and the establishment of theoretical bounds characterizing the success rate of modern low-dimensional attacks for a wide range of models.

Institut Polytechnique de Paris

91120 Palaiseau, France