Kernel-based sensitivity indices for high-dimensional optimization problems

Adrien Spagnol

N° d'ordre NNT: 2020LYSEM012

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

École des Mines de Saint-Étienne

École Doctorale N° 488
(Sciences, Ingénierie, Santé)

Spécialité de doctorat: Mathématiques appliquées
Discipline: Sciences des Données

Soutenue publiquement le 02/07/2020, par:

Adrien Spagnol

Indices de sensibilité via des méthodes à noyaux pour des problèmes d'optimisation en grande dimension

—

Kernel-based sensitivity indices for high-dimensional optimization problems

---

Devant le jury composé de:

Roustant, Olivier	Professeur, INSA Toulouse, France	Président du Jury
Gratton, Serge	Professeur, INP Toulouse, France	Rapporteur
Iooss, Bertrand	Ingénieur de Recherche Senior, HDR, EDF R&D, France	Rapporteur
Bischl, Bernd	Professeur, Ludwig-Maximilians-Universität München, Allemagne	Examinateur
Roustant, Olivier	Professeur, INSA Toulouse, France	Examinateur
Le Riche, Rodolphe	Directeur de recherche, CNRS LIMOS à EMSE, France	Directeur de thèse
Da Veiga, Sébastien	Ingénieur de recherche, Safran Tech, France	Co-encadrant de thèse

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	SCIENCES DES IMAGES ET DES FORMES	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	N. Absi, Maitre de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	Ph. Lalevée, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	MR	Génie industriel	CMP
AUGUSTO	Vincent	CR	Image, Vision, Signal	CIS
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BADEL	Pierre	MA(MDC)	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BEIGBEDER	Michel	MA(MDC)	Informatique	FAYOL
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	PR	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
CAMEIRAO	Ana	MA(MDC)	Génie des Procédés	SPIN
CHRISTIEN	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	MR	Sciences des Images et des Formes	SPIN
DEGEORGE	Jean-Michel	MA(MDC)	Génie industriel	Fayol
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
BERGER-DOUCE	Sandrine	PR1	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
DUTERTRE	Jean-Max	MA(MDC)		CMP
EL MRABET	Nadia	MA(MDC)		CMP
FAUCHEU	Jenny	MA(MDC)	Sciences et génie des matériaux	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FOREST	Valérie	MA(MDC)	Génie des Procédés	CIS
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Sciences de la Terre	SPIN
GAVET	Yann	MA(MDC)	Sciences des Images et des Formes	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GONZALEZ FELIU	Jesus	MA(MDC)	Sciences économiques	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NAVARRO	Laurent	CR		CIS
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche	Mécanique et ingénierie	CMP
NORTIER	Patrice	PR1	Génie des Procédés	SPIN
O CONNOR	Rodney Philip	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PINOLI	Jean Charles	PR0	Sciences des Images et des Formes	SPIN
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROUSSY	Agnès	MA(MDC)	Microélectronique	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
SANAUR	Sébastien	MA(MDC)	Microélectronique	CMP
SERRIS	Eric	IRD		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR0	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

Mise à jour : 30/08/2018

Supervisor
Rodolphe Le Riche, Research Director, CNRS LIMOS at EMSE, France

Co-supervisors
Sébastien Da Veiga, Research Engineer, Safran Tech, France

Jury
Serge Gratton, Professor, INP Toulouse, France
Bertrand Iooss, Senior Research Engineer, EDF, France
Bernd Bischl, Professor, Ludwig-Maximilians-Universität München, Germany,
Olivier Roustant, Professor, INSA Toulouse, France

Opponent
Adrien Spagnol

Contact information
Department of Mathematics and Industrial Engineering
Henri Fayol Institute, Mines Saint-Étienne
158 cours Fauriel, F-42023 Saint-Étienne cedex 2, France
Email address: webmaster@emse.fr
URL: https://www.mines-stetienne.fr/
Telephone: +33 (0)4 77 42 01 23
Fax: +33 (0)4 77 42 00 00

Remerciements

Me voilà enfin arrivé à l'écriture de cette section, synonyme de la fin de la rédaction de ce manuscrit mais également occasion de prendre un instant pour remercier toutes les personnes que j'ai eu la chance de croiser durant ces trois années de thèse, en tachant de n'oublier personne (auquel cas il faudra m'excuser !).

Tout d'abord, je souaite remercier mon directeur de thèse, Rodolphe Le Riche, ainsi que mon co-encadrant, Sébastien Da Veiga. Je m'estime infiniment chanceux d'avoir pu mener ce travail à vos côtés, vos qualités scientifiques ne sont plus à prouver et vos qualités humaines ont fait de ces trois ans une expérience très plaisante. Rodolphe, malgré les heures de décalage au démarrage, merci pour nos nombreux échanges à Saint-Étienne (ou ailleurs en conférence) ainsi que d'avoir accepté d'inlassablement corriger mes *due to* qui te faisaientt grincer des dents. Sébastien, merci d'avoir été là du début à la fin, quand je sais à quel point tu peux être sollicité, tu n'as jamais manqué à l'appel et ça a été une grande chance pour moi de pouvoir compter sur toi. Je sais que je t'ai promis une bière et je pense qu'il y aura des occasions pour que je puisse te l'offrir au détour d'un GdR ou autre !

Je souhaite également remercier Bertrand Iooss et Serge Gratton pour avoir accepté de relire ce manuscrit ainsi que pour leurs précieux commentaires, que ce soit dans leur rapport ou lors de l'échange après la soutenance. *I would also like to thank Pr. Bernd Bischl who accepted to be part of my jury.* Enfin, merci à Olivier Roustant d'avoir accepté d'être le président de mon jury de thèse, je sais que tu avais un regard un peu lointain sur cette thèse à ses débuts, je suis donc content que tu aies pu voir sa conclusion.

Une pensée pour tous mes collègues à Safran Tech, qui m'ont presque vu quotidiennement pendant 3 ans, surtout mes compagnons doctorants Maxence, Yannis, Florian, Clément B., Anthony, Thomas, Moubine, Perle, Camille, Clément O., Loic. Merci pour les indénombrables cafés/thés ainsi que les virées en escape game. Une pensée à ceux qui n'ont pas terminés quand j'écris ces quelques lignes. Ensuite, je souhaite également remercier tous les Stéphanois, présents ou partis depuis, je ne suis pas venu très souvent mais c'était toujours un plaisir d'échanger avec vous autour d'un thé et quelques patisseries, ou bien avec une bière en main au Lipo ou un verre de vin chez Lulu. Merci à tous ceux que j'ai croisé en conférence, tout spécialement David, pour m'avoir accompagné bien souvent, mais également Léonard, Jérôme, Quentin, Matthieu, Vincent ou Cédric, j'en garde d'excellents souvenirs, quoi qu'un peu flous parfois !

Je souhaite ensuite remercier mes amis, ceux de toujours Matthieu et Rémi, c'est toujours un plaisir de vous voir, chacun de nos voyages était une parenthèse parfaite. Viennent ensuite

Mehdi, Anthony et Ugo, en espérant que notre traditionnelle soirée bière(s) - jeu(x) - film - rhum café perdure longtemps, j'ai bien hate de la prochaine ! Vous n'êtes pas les seuls de l'IFMA que je souhaite remercier : merci aux membres du G1 pour les nombreux échanges sur Whatsapp, autour d'une bière à Paris ou en voyage, merci à Maxime pour m'avoir gentillement hébergé chez lui à Toulouse alors qu'il neigeait, merci à Amandine et Anthony (encore) pour avoir été mes compagnons de thèse, une pensée pour la première qui aura bientôt terminé et pour le second qui aura réussi l'exploit de terminer un jour avant moi ! Et merci à tous les autres que j'aurais omis ici, vous vous reconnaitrez.

Enfin, un mot pour ma famille, qui suivaient depuis Nice mon doctorat. Spécialement pour ma mère Geneviève et mon père Jean-Michel, merci d'avoir toujours dit oui lorsque je voulais expliquer ce que je faisais quand je savais que vous n'y compreniez rien. Merci de m'avoir toujours soutenu, même lorsque cela impliquait de prendre le viaduc de Magnan tous les matins pendant deux ans ou de conduire jusqu'à Vollenhove depuis Nice. Merci à ma soeur Pandora, savoir que je te rends fière est une source de motivation incroyable. Merci également à mes grand-mères Ninine et Mamou (j'utiliserai ici vos pseudonymes affectifs !). Je vous aime énormément.

Pour terminer, je souhaite remercier celle qui partage ma vie désormais. Ariane, cette thèse aura rythmée nos vies pendant pas moins de 3 ans et depuis sa fin, notre vie aura été bien chamboulée et j'ai hâte d'écrire la suite avec toi. Je t'aime infinement, merci pour tout ce que tu fais pour moi, je suis incroyablement chanceux de t'avoir rencontré.

Introduction

Contexte industriel

Au cours des 50 dernières années, les conceptions d'ingénieurs en industrie se sont de plus en plus complexifiées. Cela a été alimenté par le besoin de respecter de nouvelles contraintes de conception, résultant de standards de sûreté ou de normes environnementales plus restristives. Étant donné que ces conceptions nécessitent souvent de recourir à des expériences rares et couteuses (tel que des mannequins d'essai de choc dans l'automobile ou la cration de moteur aéronautiques), ceux-ci ont été remplacés par des modèles numériques haute fidélité. Bien souvent, ces modèles sont des approximations mathématiques des systèmes conçus puisque la physique réelle de ceux-ci ne peut être calculée analytiquement. La simulation numérique est maintenant largement utilisée dans de nombreux domaines, tels que la physique, la chimie, la biologie ou en ingénierie pluridisciplinaire. Elle permet de reproduire le comportement d'un système en considérant une large possibilité de conceptions différentes et cela dans diverses conditions. Afin de rendre ces modèles numériques plus fidèles, il est nécessaire de considérer un nombre grandissant de paramètres d'entrée. Par example, pour des applications d'ingénierie mécanique, les différents composants sont généralement représentés par Conception Assistée par Ordinateur (CAO). Afin d'obtenir une définition plus fine de ces pièces, un grand nombre de variables est requis pour pleinement décrire leur forme : longueur, hauteur ou profondeur de chacune des pièces, diamètre et profondeur des trous, caractérisation des soudures et ainsi de suite. Considérer un modèle incluant un grand nombre de paramètres est décrit comme un problème complexe en grande dimension.

Les modèles numériques offrent la possibilité d'explorer de nouvelles conceptions et de mieux comprendre les compromis sous-jacents. L'optimisation est une des méthodes d'exploration les plus importantes : elle consiste à trouver la configuration d'entrées $\mathbf{X}^*$ donnant la meilleure performance $f(\mathbf{X}^*)$ tout en respectant un certain ensemble de contraintes $f(\mathbf{X}^*)$. Par exemple, considérant l'optimisation de la forme des pales du rotor du compresseur d'un moteur d'avion, voir Figure 1 pour sa géométrie paramétrée. Le concepteur souhaitera maximiser les performance aéro-dynamique du compresseur en changeant les différents paramètres de forme des pales, tout en garantissant un espace nécessaire en bout de pale ainsi qu'en respectant les différentes contraintes mécaniques. Il est commun de considérer l'objectif f comme une fonction dite *boite-noire* : ni l'expression analytique ni ses dérivées ne sont accessibles. Il est uniquement possible de connaitre les valeurs de la fonction en évaluant différentes valeurs pour $\mathbf{X}$. Des examples typiques de boite-noires sont les modèles numériques avec des équations aux dérivées partielles

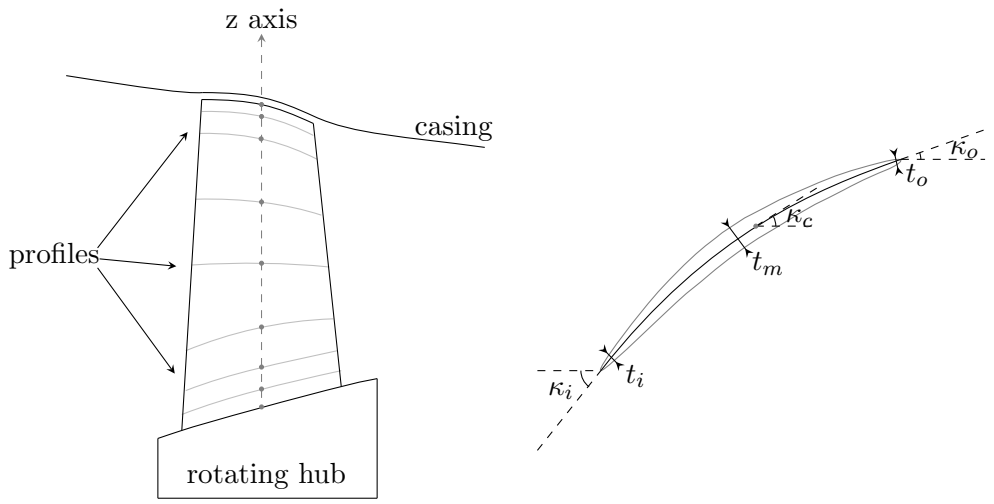

FIGURE 1 – Vue méridienne de la pale du rotor d'un compresseur de moteur aéronautique (à gauche). La pale est décrite par plusieurs profils verticalement de la base à la pointe et chaque profil est défini via plusieurs paramètres (à droite). Les propriétés matériaux de la pale pourrait également être considérées comme des entrées du modèle.

tels que l'analyse par éléments finis, et plus généralement tout simulateur numérique accessible sous forme d'un exécutable sans source exploitable. Dans certains cas, un appel au modèle numérique peut avoir un coût non négligeable en terme de temps ou de ressources numériques nécessaires. Le nombre d'appels aux fonctions f et g est alors limité pendant l'optimisation, typiquement avec un maximum d'une centaine d'évaluations possibles.

Approches standards

Plusieurs algorithmes et méthodes d'optimisation ont été proposées dans la littérature. Lorsque les dérivées sont connues, il est possible d'utiliser les méthodes à base de gradients [NW06]. Pour ces méthodes, la qualité de la solution trouvée dépend de la localisation du point de départ de l'optimisation, même si cela peut être corrigé en considérant des stratégies de *multi-start* (en répétant l'optimisation plusieurs fois, avec des points initiaux différents). Dans le formalisme boite-noire, puisque les informations de dérivées ne sont pas accessibles, des méthodes d'approximation sont envisageables avec un coût associé (pour $\mathbf{X} \in \mathbb{R}^d$, une stratégie par différences finies nécessite $d+1$ appels à la fonction dont on souhaite approcher le gradient). Cependant, ces méthodes ne sont pas envisageables lorsque la dimension du problème est grande puisque le coût résultant d'une seule itération de l'optimisation devient prohibitif. Les métaheuristiques d'optimisation tels que les algorithmes probabilistes [Zhi12] et les algorithmes évolutionnistes) (AE) [ES+03] (contenant les algorithmes génétiques [Mit98], l'optimisation par essaims particulaires [EK95] ou les stratégies d'évolution [BS02])) sont des méthodes dites d'ordre zéro puisqu'elles ne demandent pas d'avoir connaissance des dérivées, mais un grand nombre d'appels au modèle est requis avant de trouver un optimum (ou une estimation suffisamment précise). Ces méthodes ne sont donc pas adaptées lorsque la fonction boite-noire a un coût associé important, puisque le nombre d'appels à celle-ci est limitée. Une troisième classe de méthodes, appelées les méthodes d'optimisation par surfaces de réponses, repose sur des substituts peu couteux du vrai modèle [Jon01 ; WS06]. Les surfaces de réponses (ou métamodèles, tels que les Processus Gaussiens [WR06], les machines à vecteurs de support [SSB+02] ou les fonctions de base radiale [Reg16 ;

Gut01]) sont des approximations de la vraie fonction objectif et peuvent être utilisés pour la recherche de l'optimum. Ces méthodes utilisant des métamodèles sont efficientes avec peu d'évaluations de la fonction objectif et ont donc été utilisées sur une large gamme d'applications [Sha+15].

Cependant, ces méthodes reposant sur des modèles d'approximation sont limitées à des problèmes définis en faible dimension puisqu'elles ne sont plus aussi efficientes quand le nombre de variables définissant le problème devient conséquent, un problème connu sous le nom du *fléau de la dimension* [Bel15]. Afin de passer à des modèles de grande dimension, plusieurs stratégies ont été proposées, telles que les calculs parallélisés, la décomposition du problème en sous-problèmes [SW10] ou la projection de l'espace des paramètres d'entrée dans un sous-espace de taille réduite [Bou+16 ; Gau+19]. Dans cette thèse, nous nous intéressons à des méthodes réduisant le volume de l'espace de recherche via des approches qui détectent quelles sont les variables influentes du problème afin d'effectuer l'optimisation dans un espace de plus petite taille. L'élément clef de nos approches est l'analyse de sensibilité globale, qui étudie comment la variabilité de la sortie d'un modèle évolue lorsque certaines des entrées sont fixées [IL15]. Bien qu'utilisées sur de nombreuses applications [RLM09 ; FKR11 ; Cho+14], les stratégies à base de décomposition de la variance ont un certain nombre de limitations lorsqu'utilisées sur nos problèmes d'optimisation. L'analyse de sensibilité *goal-oriented* est un ensemble d'approches qui considèrent des quantités d'intérêt spécifiques de la sortie du modèle. Celles-ci prennent en compte le type d'étude étant mené (celle-ci pouvant être une étude de fiabilité, d'optimisation, etc). Dans la lignée de ces méthodes, nous répondons aux questions suivantes dans ces travaux :

- Quelle est la quantité d'intérêt à considérer dans le cadre d'un problème d'optimisation ?
- Comment sélectionner les variables détectées comme importantes pour la résolution du problème d'optimisation et que faire des variables restantes ?
- Comment implémenter une telle sélection de variables au sein d'un algorithme d'optimisation ?

Structure du manuscrit et constributions

La principale contribution de cette thèse est la proposition de nouveaux indices de sensibilité dédiés à l'optimisation sous contraintes. Nous nous intéressons aux problèmes en grande dimension puisqu'ils présentent un certain nombre de challenges précédemment mentionnés. Notre méthode peut être utilisée en amont de la procédure d'optimisation ou directement intégrée au sein de celle-ci.

Le chapitre 2 introduit les bases de l'analyse de sensibilité dans la section 2.1 avec une description rapide des méthodes locales et basées sur la variance de la sortie, qui sont les plus largement utilisées pour leur simplicité et facilité de compréhension. Puisque ces méthodes se limitent à une quantité d'intérêt spécifique de la distribution de la sortie du modèle, d'autres approches intégrant toute la distribution furent proposées. Parmi ces méthodes se trouvent les stratégies à base de noyaux dont les différents aspects théoriques ainsi que plusieurs stratégies d'estimation sont décrits dans la section 2.2. Puisque le principal objectif de ces travaux est lié à l'optimisation, la section 2.3 présente diverses stratégies *goal-oriented* qui prennent en compte le type d'étude lors de la recherche des variables pertinentes.

Le chapitre 3 décrit les indices de sensibilité dédiés à des problèmes d'optimisation et basés sur des méthodes à noyaux, et donne une stratégie d'implémentation au sein d'une procédure d'optimisation. L'analyse de sensibilité permet de réduire la dimension du problème et d'ainsi faciliter l'étape de l'obtention d'un optimum, avec pour conséquence une dégradation de la précision de celui-ci. Ce chapitre est une adaptation de [SLRDV19] avec des détails supplémentaires. Nous proposons dans la section 3.1 une modification de la sortie du modèle afin de définir une nouvelle quantité d'intérêt ad hoc pour les études d'optimisation sous contraintes. Cela permet de formuler un nouvel indice de sensibilité appelé HSIC-IT. Puis, dans la section 3.2, nous définissions une approche d'optimisation qui tire parti de la sélection de variables menées par le biais de ces nouveaux indices. Les avantages liées à une optimisation menée en dimension réduite avec notre algorithme sont montrés dans la section 3.3 sur plusieurs fonctions jouets. La convergence semble plus stable le nombre d'appels nécessaire pour obtenir une solution au problème d'optimisation est considérablement réduit. Cependant, puisque des dimensions ont été retirées du problème initial, cela résulte en une légère dégradation de la solution optimale.

Enfin, le chapitre 4 traite à la fois du problème de la grande dimension et de l'aspect couteux des évaluations des fonctions du problème d'optimisation en couplant une sélection de variables avec une stratégie d'optimisation Bayésienne séquentielle. La section 4.1 détaille le cadre de l'optimisation Bayésienne, décrivant régression par processus Gaussiens et les différentes fonctions d'acquisition. Dans la section 4.2, les limitations rencontrées par cette méthode lorsque le problème est défini en grande dimension sont présentés ainsi que diverses stratégies présentes dans la littérature. Ensuite, dans la section 4.3, nous présentons notre algorithme, intégrant la précédente sélection de variables par le biais de méthodes à noyaux au sein de la boucle d'optimisation. Des améliorations ont été obtenues grâce à la réduction de la dimension, décrits sur quelques cas-tests. Enfin, différentes extensions de l'algorithme sont présentés dans la section 4.4 ainsi que des résultats sur plusieurs exemples afin de montrer qu'une sélection réfléchie des variables permet d'obtenir une meilleure solution au problème d'optimisation pour un nombre d'évaluations donné.

En guise de conclusion, le chapitre 5 résume les différents résultats obtenus tout au long de cette thèse et discute des perspectives de ces travaux pour de futures recherches.

Contents

List of Figures

List of Tables

Publications and communications

The research contribution presented in this manuscript is based on diverse scientific contributions listed herebelow.

Publication and conference proceeding

- Spagnol, Adrien, Le Riche, Rodolphe , & Da Veiga, Sébastien (2019). Global sensitivity analysis for optimization with variable selection. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2), 417-443. DOI 10.1137/18M1167978.
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga (2019). Bayesian optimization in effective dimensions using kernel-based sensitivity indices, *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP13)*, 439-446. DOI 10.22725/ICASP13.093.

Conferences

- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga, Olivier Roustant. Global sensitivity analysis for optimization with variable selection, *PGMO Days 2017*, Nov 2017, Saclay, France.
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga, Olivier Roustant. Global sensitivity analysis for optimization with variable selection, *Journées de la Chaire OQUAIDO 2017*, Nov 2017, Orléans, France. (Poster)
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga. Global sensitivity analysis for optimization with variable selection, *MASCOT-NUM Annual Conference 2018*, Mar 2018, Nantes, France. (Poster)
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga. Global sensitivity analysis for optimization with variable selection. *12th International Conference on Learning and Intelligent Optimization (LION 12)*, June 2018, Kalamata, Greece.
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga. Bayesian optimization in effective dimensions using kernel-based sensitivity indices, *Journées de la Chaire OQUAIDO 2018*, Nov. 2018, Cadarache, France. (Poster)
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga. Bayesian optimization in effective dimensions using kernel-based sensitivity indices, *MASCOT-NUM Annual Conference 2019*, Mar. 2019, Rueil-Malmaison, France. (Award for the best PhD oral presentation)
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga. Bayesian optimization in effective dimensions using kernel-based sensitivity indices, *13th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP13)*, May 2019, Seoul, South Korea
- Adrien Spagnol, Rodolphe Le Riche, Sébastien Da Veiga. Bayesian optimization in effective dimensions using kernel-based sensitivity indices, *30th European Conference on Operational Research (EURO 2019)*, June 2019, Dublin, Ireland.

Chapter 1

Introduction

Contents

1.1 Industrial context

For the past 50 years, engineering practice in the industry has evolved towards the design of increasingly complex systems. This evolution has been fueled by the need to abide with new design constraints such as reliability and tighter environmental standards. Since such designs often require expensive tests with limited occurrences (for example crash test dummies for the automotive industry or aeronautic engine prototypes), they have logically been replaced with high-fidelity numerical models. Most often, these models are mathematical approximations of the real system as the underlying physic cannot be computed analytically. Numerical simulation is now relied upon in many fields, such as physics, chemistry, biology or multidisciplinary engineering. Computer simulations allow to reproduce the behavior of the system under a wide variety of design and working conditions. In the permanent effort made to have more realistic numerical models, a larger number of inputs has to be specified. For example, in mechanical engineering applications, sets of components are commonly represented using Computer Aided Designs (CAD). In order to obtain a refined definition of the system, a large number of parameters are necessary to fully represent its shape: length, height or width for each piece, diameter and height of holes, characterization of fillets and so forth. Working with models that encompass a large number of inputs yields to *high-dimensional* decision problems.

Numerical models provide an opportunity to explore new design configurations and better understand design trade-offs. Optimization is one of the most important exploration that can be performed: it consists in finding the variables layout $\mathbf{X}^*$ which produces the best possible performance $f(\mathbf{X}^*)$ without violating a given set of constraints $g(\mathbf{X}^*)$. As an example, consider the optimization of the shape of the rotor blades in the compressor stage of an aeronautical engine, see Figure 1.1 for its geometry. The designer often wants to maximize the aerodynamic performance of the compressor by changing the shape parameters of its rotor blades, while considering thresholds on the tip gap and the mechanical resistance of the part as constraints. It is common to consider the objective f as a *black-box* function: neither the analytic expression nor the derivatives of f are available. Evaluations of the function are only possible by querying its value for different settings of $\mathbf{X}$. Classical examples of black-boxes are numerical codes involving partial differential equations such as finite element analyses, and more generally every numerical simulator in the form of an executable without access to the sources. In some cases, a call to the numerical model has a non negligible cost in terms of the time or computing resources required. The optimization is then limited in the number of evaluations of f and g. For a typical cost of a single simulation, the maximum number of evaluations is of the order of the hundred.

1.2 Standard approaches

Different optimization approaches and algorithms were proposed in the literature. One can resort to gradient-based methods [NW06] if derivatives of the models are known. The quality of the solution found depends on the starting point, even though this can be improved by considering multi-start strategies (i.e. running several times the optimization from different starting points). In the black-box setting, as the gradient is not accessible, approximation techniques exists at the expense of additional calls to the model (for $\mathbf{X} \in \mathbb{R}^d$, a finite difference approximation to the gradient requires $d+1$ calls). However, the total cost of the optimization iterations scales poorly with the dimension, making such methods impracticable when the dimension of

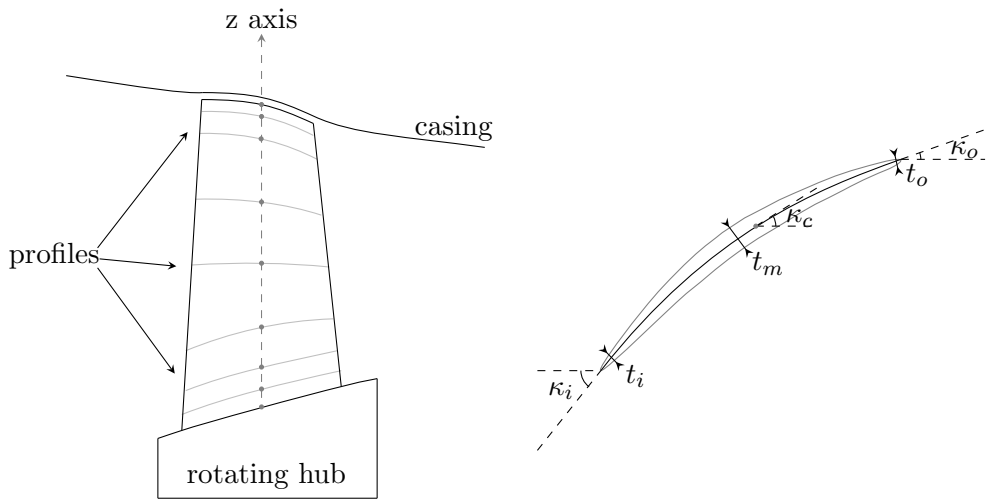

Figure 1.1 – Meridional view of a rotor blade in the compressor stage of an aeronautical engine (left). The blade is defined by stacking multiple profiles from the hub to the blade tip and each profile is described using several parameters (right). Material properties of the blade could also be considered as tunable inputs.

the problem is large. Population-based strategies such as probabilistic algorithms [Zhi12] and evolutionary algorithms (EA) [ES+03] (including genetics algorithms [Mit98], particle swarm optimization [EK95] or evolution strategies [BS02]) are a class of zero-order methods in the sense that they do not require any derivative information, but they usually necessitate numerous model evaluations before finding (a sufficiently accurate estimate of) an optimum. These methods do not fit well the *expensive* black-box setting where the budget is strongly limited. A third class of methods that rely on computationally efficient substitute to the true model are called *surrogate optimization methods* [Jon01; WS06]. The surrogate models (e.g. Gaussian processes [WR06], support vector regression [SSB+02] or radial basis functions [Reg16; Gut01]) are approximations to the objective function and can be used to search for the optimum. These surrogate optimization methods were proven to be efficient at low number of evaluations and have therefore been applied in a wide range of applications [Sha+15].

However, optimization strategies directly based on surrogates are limited to low dimensional spaces as they also scale poorly with the dimension, a phenomenon referred to as the *curse of dimensionality* [Bel15]. To tackle high-dimensionality, several approaches have been proposed, including parallel computing, decomposition of the problem into sub-problems [SW10] and projection of the input space into a lower dimensional space [Bou+16; Gau+19]. In this thesis, we focus on methods that reduce the volume of the design space using strategies to detect significant variables and then perform the optimization in the reduced space. Global sensitivity analysis is a key ingredient in our approaches. It studies how the variability of a function output changes when some of its inputs are frozen [IL15]. Despite having been used on different applications [RLM09; FKR11; Cho+14], the popular variance-based approaches suffer from different drawbacks when applied to optimization problems. *Goal-oriented sensitivity analysis* is a set of methodologies that consider specialized, ad hoc, quantities of interest of the output. The quantity of interest takes into account the type of study done (e.g. reliability, optimization, etc). Following this trend of thought, we aim in this thesis at answering the following questions:

- What is the quantity of interest to consider when conducting an optimization study?
- How to select variables that are important in order to best solve the optimization problem and what to do with the remaining dimensions?
- How to implement such selection within an optimization strategy?

1.3 Structure of the manuscript and contributions

The main contribution of this thesis is the proposition of new sensitivity indices dedicated to optimization tasks under constraints. We focus on high-dimensional problems as they raise challenging issues in the expensive scope (see above). Our method can be used prior to the optimization procedure or it can directly be integrated within it.

Chapter 2 starts with the basics of sensitivity analysis in Section 2.1, with a brief description of local and variance-based measures, which are broadly used for their simplicity and easy understanding. Since these analyses measure a specific quantity of interest of the output distribution, other approaches that use the full distribution were later proposed. Among these methods are the *kernel-based* strategies: the associated theoretical aspects and different approaches for their estimation are reported in Section 2.2. As the main goal of this thesis is optimization, Section 2.3 presents different *goal-oriented* methods which take into account the type of the problem when assessing the relevance of the different inputs.

Chapter 3 proposes kernel-based indices for optimization-oriented sensitivity analysis and gives an implementation in terms of an optimization procedure. The sensitivity analysis step leads to a dimension reduction which eases the optimization procedure, at the cost of a slightly degraded optimum. This chapter is an adaptation from [SLRDV19] with additional details. We propose in Section 3.1 a modification of the output to define a new ad hoc quantity of interest for constrained optimization studies. It results in a new sensitivity index called HSIC-IT. Then in Section 3.2, we define an optimization approach that takes advantage of the HSIC-IT based input selection. The benefits of conducting dimension reduction with our algorithm are shown in Section 3.3 on several test cases. The convergence to the optimum appears more stable and the number of calls to the objective function is considerably reduced. However, because of the removal of some dimensions and the resulting loss in fine tuning possibilities, the obtained solution can be slightly deteriorated.

Finally, Chapter 4 tackles the issue of high dimensional and expensive functions to optimize by coupling the variable selection phase within a Bayesian sequential optimization strategy. Section 4.1 recalls the framework of Bayesian optimization, explaining both Gaussian processes regression and the different acquisition functions. In Section 4.2, the limitations faced for high dimensional problems are presented along with strategies found in the literature to face such issues. Next, in Section 4.3, we introduce our algorithm which integrates the kernel-based variable reduction previously presented within the optimization loop. Improvements obtained thanks to the dimension selection phase are highlighted on a couple of test cases. Finally, different extensions of the algorithm are defined in Section 4.4 along with results on tests cases showing that a clever variable selection leads to better objective function values for a given number of calls to the costly numerical model.

To conclude, Chapter 5 summarizes the different results obtained throughout this thesis and

discusses perspectives for future work.

Bibliography

[Bel15] Richard E Bellman. *Adaptive control processes: a guided tour*. Vol. 2045. Princeton university press, 2015.

[Bou+16] Mohamed Amine Bouhlel et al. "Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction". In: *Structural and Multidisciplinary Optimization* 53.5 (2016), pp. 935–952.

[BS02] Hans-Georg Beyer and Hans-Paul Schwefel. "Evolution strategies–A comprehensive introduction". In: *Natural computing* 1.1 (2002), pp. 3–52.

[Cho+14] Hyunkyoo Cho et al. "An efficient variable screening method for effective surrogate models for reliability-based design optimization". In: *Structural and Multidisciplinary Optimization* 50.5 (2014), pp. 717–738.

[EK95] Russell Eberhart and James Kennedy. "Particle swarm optimization". In: *Proceedings of the IEEE international conference on neural networks*. Vol. 4. Citeseer. 1995, pp. 1942–1948.

[ES+03] Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*. Vol. 53. Springer, 2003.

[FKR11] Guangtao Fu, Zoran Kapelan, and Patrick Reed. "Reducing the complexity of multiobjective water distribution system optimization through global sensitivity analysis". In: *Journal of Water Resources Planning and Management* 138.3 (2011), pp. 196–207.

[Gau+19] David Gaudrie et al. "Modeling and Optimization with Gaussian Processes in Reduced Eigenbases". In: *Structural and Multidisciplinary Optimization* (2019). accepted for publication.

[Gut01] H-M Gutmann. "A radial basis function method for global optimization". In: *Journal of global optimization* 19.3 (2001), pp. 201–227.

[IL15] Bertrand Iooss and Paul Lemaître. "A review on global sensitivity analysis methods". In: *Uncertainty management in simulation-optimization of complex systems*. Springer, 2015, pp. 101–122.

[Jon01] Donald R Jones. "A taxonomy of global optimization methods based on response surfaces". In: *Journal of global optimization* 21.4 (2001), pp. 345–383.

[Mit98] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[NW06] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[Reg16] Rommel G Regis. "Multi-objective constrained black-box optimization using radial basis function surrogates". In: *Journal of computational science* 16 (2016), pp. 140–155.

[RLM09] Uwe Reuter, Martin Liebscher, and Heiner Müllerschön. "Global sensitivity analysis in structural optimization". In: *7th European LS-DYNA Conference, Salzburg*. 2009.

[Sha+15] Bobak Shahriari et al. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175.

[SLRDV19] Adrien Spagnol, Rodolphe Le Riche, and Sébastien Da Veiga. "Bayesian optimization in effective dimensions via kernel-based sensitivity indices". In: *13th International Conference on Applications of Statistics and Probability in Civil Engineering(ICASP13)*. Seoul National University. 2019.

[SSB+02] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[SW10] Songqing Shan and G Gary Wang. "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions". In: *Structural and Multidisciplinary Optimization* 41.2 (2010), pp. 219–241.

[WR06] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[WS06] G Gary Wang and Songqing Shan. "Review of metamodeling techniques in support of engineering design optimization". In: (2006).

[Zhi12] Anatoly A Zhigljavsky. *Theory of global random search*. Vol. 65. Springer Science & Business Media, 2012.

Chapter 2

Sensitivity analysis

Contents

2.1 Background

In this manuscript, we consider the following formalism for the computer function f studied:

$$\begin{aligned} f : \mathcal{X} \subseteq \mathbb{R}^d &\to \mathcal{Y} \subseteq \mathbb{R} \\ \mathbf{x} &\mapsto y = f(\mathbf{x}) \end{aligned} \tag{2.1}$$

where $\mathbf{x} \in \mathcal{X}$ (resp. $y \in \mathcal{Y}$) is called the input vector, here of dimension d (resp. the output, here presented as a scalar). The model f is considered as a *black-box* function, which means that at any point in time, one can only interact with f by querying output corresponding to a given input vector x. No further information is available for the user, such as direct closed-form or gradients of the function.

Sensitivity analysis allows to answer many different questions about the model. Finding out what are the inputs that contribute the most to the system variability is primary goal of sensitivity analysis. If the said variability is synonym of inaccuracy for the model output value, an improvement of the model response quality should be pursued. Variability of the model can be characterized at a lower cost by focusing on the variables influencing the uncertainties the most. One should note that this approach is not always viable, since the variability of an input variable can be inherent to its direct nature, and not to a lack of knowledge or measurement done with low precision.

Sensitivity analysis can also help to determine how close to the real phenomenon the model is, by displaying some variables as impactful when they are already, physically, known to be negligible. In such cases the reliability of the model and/or the a priori knowledge about the real impact of the variables might need discussion. Furthermore, using sensitivity analysis to detect variables interactions leads to a better understanding of the modeled process.

Another setting for sensitivity analysis is to capture the impact of each variable on the numerical model and characterizing the influence of an input can be done at several levels. First of all, one can rank variables based on how great is the output variance reduction when the giving input is set to its true value. [Sal+04] calls it *factor prioritization*. One might also want to identify which inputs can be set to any given value of their design set without impacting the output variance: this is denoted as *factor fixing* [Sal+04], where initially focusing on the negligible inputs leads to a possible model dimension reduction by considering those variables as deterministic ones (e.g. setting them to their mean value). Finally, the approach named *factors mapping* [Sal+04], catches which variables are most responsible for producing output. It is a key practice for constrained or goal-oriented studies. [Sal+04], [Sal+08], [IL15] and [BP16] give a broader picture on the different settings for sensitivity analysis.

Generally, methods developed for sensitivity analysis can be sorted out in two major categories: local sensitivity analysis approaches, which study the quantitative impact of a small variation of the entries around a nominal value $\mathbf{x}^0$ (which can be all inputs set to a given value, often the mean or the mode); and global sensitivity analysis approaches, which evaluate the variations of the output when the inputs vary in their entire domain of uncertainty. Global sensitivity analysis (GSA) studies variability of the output when the local sensitivity analysis (LSA) focuses more on the actual response value. Figure 2.1 gives an illustration of both methods for a unidimensional problem.

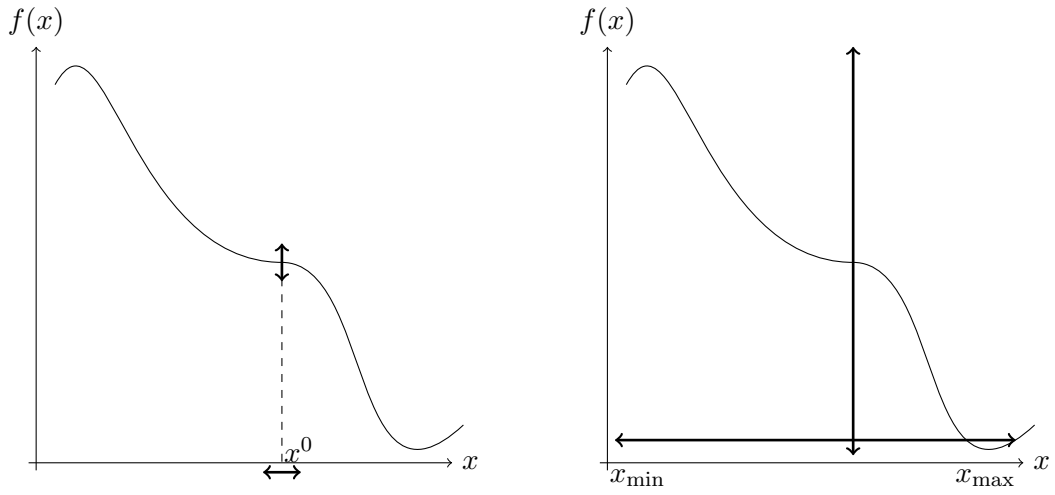

Figure 2.1 – An illustration on a one dimensional function of local sensitivity analysis around the nominal point x_0 (left) and global sensitivity analysis (right).

2.1.1 Local sensitivity analysis

As mentioned above, local methods [Sal+08] only provide information about the impact of one input around a set value $\mathbf{x}^0$, hence their names. Most of these sensitivities are estimated through gradients or partial derivatives of the output at these nominal values. As an example of such methods, assuming f is differentiable at $\mathbf{x}^0$, the partial derivatives

$$\Delta(\mathbf{x}^0)_i = \frac{\partial f}{\partial x_i}(\mathbf{x}^0) = \frac{\partial f}{\partial x_i}(x_1^0, \ldots, x_d^0) \tag{2.2}$$

for $i = 1, \ldots, d$, can be used to rank the inputs e.g., through the absolute value of the partial derivatives.

Despite being widely used, the main drawback of local methods comes from their limited range of action since they only focus on variation around nominal values while inputs usually vary over a full definition domain. If the variations are reasonable and the model linear, those methods for inputs ranking could suffice. Finally, one should note that an estimation of the derivatives is necessary: if they cannot be evaluated explicitly, approximation techniques are required, hence a non negligible computational cost ($\mathcal{O}(d)$ for one finite difference).

2.1.2 Screening methods

Screening methods close the gap between local and global methods by repeating local analyses throughout the definition domain. One of the most used method is the Morris method [Mor91] whose idea is to determine whether an input has an effect that is negligible, linear and additive, or nonlinear or interacting with other inputs. The approach simply consists in discretizing the input space $\mathcal{X}$ in levels for each variable (e.g. with a grid) then performing a given number n of one-at-a-time (OAT) designs. Assuming, the input space is discretized with a d-dimensional grid with n levels for each input, the *elementary effect* of the i-th variable at the j-th repetition is:

$$\Delta^{\mathrm{M}}(\mathbf{x}^j)_i = \frac{f(\mathbf{x}^j + \delta_{\mathrm{M}}\mathbf{e}_i) - f(\mathbf{x}^j)}{\delta_{\mathrm{M}}} \tag{2.3}$$

where δ_{M} is the perturbation (proportional to $\frac{1}{(n-1)}$) and $\mathbf{e}_i$ a vector of the canonical base. For each input, Morris then derives two sensitivity indices based on $\Delta^{\mathrm{M}}(\mathbf{x}^j)_i$, the mean of the absolute value of the elementary effects

$$\mu_i = 1/n \sum_{i=1}^{n} |\Delta^{\mathrm{M}}(\mathbf{x}^j)_i| \tag{2.4}$$

and the standard deviation of the elementary effects:

$$\sigma_i = \sqrt{1/n \sum_{i=1}^{n} \left(\Delta^{\mathrm{M}}(\mathbf{x}^j)_i - 1/n \sum_{i=1}^{n} \Delta^{\mathrm{M}}(\mathbf{x}^j)_i \right)^2}. \tag{2.5}$$

μ_i characterizes the effect of x_i on the output. The absolute value is here to avoid compensation between elementary effects [CCS07]. A larger value of μ_i implies a larger contribution of variable x_i to the dispersion of the output. σ_i is a measure of nonlinear and/or interaction effects of x_i. A small value suggests a linear relationship between the output and x_i, while a large value suggests nonlinear effects or interactions with at least one other variable. Morris' sensitivity indices are usually plotted directly in a graph (with σ_i^2 as a function of μ_i) providing a qualitative tool to assess the importance of an input as points close to the origin are negligible. Yet, for models with very large number of inputs, Morris's indices typically fail at identifying influential factors [Sal+08] and cannot differentiate nonlinear from interacting effects, which can turn out to be quite critical. Another screening method overcomes the local deficiency by averaging the square of Equation (2.2) over the parameter space. This defines a sensitivity index called the Derivative-based global sensitivity measure (DGSM) [KS09]. Assuming f depends on $\mathbf{X} = (X_1, \ldots, X_d)$, with joint probability distribution function $p_{\mathbf{X}}$ on $\mathbb{R}^d$, the DGSM ν_i is given by

$$\nu_i = \int_{\mathbb{R}^D} \left(\frac{\partial f(\mathbf{X})}{\partial x_i} \right)^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = \mathbb{E}\left[\frac{\partial f(\mathbf{X})}{\partial x_i} \right]^2. \tag{2.6}$$

ν_i is more accurate than its Morris counterpart as the elementary effects are evaluated as strict local derivatives with small increments compared to the variable uncertainty ranges. DGSM are appealing measures as their cost is lower than most other global sensitivity methods when gradients of the function are available.

2.1.3 Variance-based methods

The screening methods described above provide qualitative tools usable to rank input factors in order of importance but do not assess by how much one given factor is more important than another. These methods come in handy for a fast exploration of the input space or characterization of the model behavior and dependencies. Among quantitative global methods which provide importance measures, Sobol indices [Sob93] are one of the most widely used sensitivity analysis strategy. Assuming f is square-integrable and defined on the unit hypercube $[0,1]^d$, with $\mathbf{X} = (X_1, \ldots, X_d)$ the random vector of d mutually independent inputs, one can decompose the output $Y = f(\mathbf{X})$ as a sum of increasing dimension functions [Hoe48]

$$Y = f(\mathbf{X}) = f_0 + \sum_{i=1}^{d} f_i(X_i) + \sum_{1 \le i < j \le p} f_{i,j}(X_i, X_j) + \ldots + f_{1\ldots d}(\mathbf{X}) \tag{2.7}$$

$$= \sum_{u \subset \{1,\dots,d\}} f_u(X_u)$$

where $f_0 = \mathbb{E}[f(\mathbf{X})] = \int f(\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}$, with $p_{\mathbf{x}}$ the product of d uniform marginals over $[0,1]$, $f_i(X_i) = \mathbb{E}[f(X) \mid X_i] - f_0$ and $f_u(X_u) = \mathbb{E}[f(X) \mid X_u] - \sum_{v \subset u} f_v(X_v)$ with $X_u = (X_i)_{i \in u}$, for any $u \subset \{1, \dots, d\}$, i.e. all the possible subset combinations without repetitions. This decomposition exists and is unique under the conditions

$$\int_0^1 f_{i_1 \dots i_s}(x_{j_1}, \dots, x_{j_s}) dx_{j_k} = 0, \forall k \in \{1, \dots, s\}, \forall \{j_1, \dots, j_s\} \subseteq \{1, \dots, d\} \tag{2.8}$$

This condition of uniqueness implies that f_0 is a constant. From Equation (2.7), we can derive the functional decomposition of variance, also named functional analysis of variance (or ANOVA) [ES81]:

$$\begin{aligned} \mathbb{V}[f(\mathbf{X})] &= \sum_{i=1}^{d} \mathbb{V}[f_i(X_i)] + \sum_{1 \le i < j \le d} \mathbb{V}[f_{i,j}(X_i, X_j)] + \dots + \mathbb{V}[f_{1\dots d}(\mathbf{X})] \\ &= \sum_{u \subset \{1,\dots,d\}} \mathbb{V}[f_u(X_u)] \end{aligned} \tag{2.9}$$

The Sobol indices [Sob93; Sob01], or variance-based indices, are obtained from Equation (2.9), if $\mathbb{V}[f(\mathbf{X})] \neq 0$:

$$S_i = \frac{\mathbb{V}[\mathbb{E}[Y \mid X_i]]}{\mathbb{V}[Y]} \tag{2.10}$$

$$S_{ij} = \frac{\mathbb{V}[\mathbb{E}[Y \mid X_i, X_j]] - \mathbb{V}[\mathbb{E}[Y \mid X_i]] - \mathbb{V}[\mathbb{E}[Y \mid X_j]]}{\mathbb{V}[Y]} \tag{2.11}$$

Equation (2.10) defined the first-order Sobol indices which evaluate the share of variance of the output due to the sole effect of the input X_i. The second order Sobol index in Equation (2.11) measures the effect due to the interaction between X_i and X_j minus the main effect of each variable. The total number of indices rises to $2^d - 1$ and they sum to one. Indices of order higher than two are usually not evaluated to save computational time and because they become difficult to interpret. Instead, one prefers to compute the total Sobol index proposed by [HS96]:

$$S_{T_i} = S_i + \sum_{i<j} S_{ij} + \sum_{j \neq i, k \neq i, j<k} S_{ijk} + \dots = \sum_{\substack{v \subset \{1,\dots,d\} \\ v \supset i}} S_v \tag{2.12}$$

S_{T_i} is the sum of all the Sobol indices containing the index i. It measures the share of the output variance due to all the combined effect in which X_i is involved. Most of the time, when d becomes large, only the first and total indices are estimated as they suffice to provide enough information on the model sensitivities. One can express the total Sobol index of an input i as the difference between the sum of all sensitivity indices which must be one and all terms of any order that do not include X_i

$$S_{T_i} = 1 - \frac{\mathbb{V}[\mathbb{E}[Y \mid \mathbf{X}_{\sim i}]]}{\mathbb{V}[Y]} \tag{2.13}$$

where $\mathbf{X}_{\sim i}$ means $\mathbf{X}$ withoutX_i. Now, following [Sal+08], since by the low of total variance

$$\mathbb{E}_{X_i}[\mathbb{V}_{\mathbf{X}_{\sim i}}[Y \mid X_i]] + \mathbb{V}_{X_i}[\mathbb{E}_{\mathbf{X}_{\sim i}}[Y \mid X_i]] = \mathbb{V}[Y] \tag{2.14}$$

one can rewrite Equation (2.13) as

$$S_{T_i} = \frac{\mathbb{E}[\mathbb{V}[Y \mid \mathbf{X}_{\sim i}]]}{\mathbb{V}[Y]} \tag{2.15}$$

Several techniques have been devised to compute those sensitivity indices usually based on Monte-Carlo estimations: [Sob93] proposes an estimator for first order and interaction indices and [Sal02] introduces estimators for first and total indices. Yet, these methods are computationally expensive for a precise estimation of the indices (since their convergence is $\mathcal{O}(\sqrt{n})$ with n the sample size). They usually require ten of thousands of model calls to estimate the Sobol index of one input. Other sampling strategies have been proposed such as the Quasi Monte-Carlo sampling or the FAST method [CLS77], based on a multi-dimensional Fourier transform. Sobol indices provide a meaningful interpretation as each index characterizes the contribution of one input to the variance of the output.

To cope with the computational cost issue, another popular solution involves the use of surrogate models to compute Sobol indices. For example, [Sud08] relies on polynomial chaos expansion (PCE) to efficiently derive sensitivities. In [Mar+09], which is further developed in [LGMS16], Sobol indices are estimated using Gaussian Processes (GP) and their confidence intervals are also derived. However, these sensitivity indices give information on the influence of inputs in the full design domain and the moment they focus on is specific. This may fail to reflect the true importance of variables in cases where the variance is not sufficient to describe how an input matters, e.g. when the output distribution is highly skewed, heavy-tailed or multimodal [LCS06].

2.1.4 Dissimilarity-based methods

To overcome the aforementioned limitations, several alternatives to variance-based sensitivity indices have been proposed, in particular *distribution-based* methods. [Bor07] proposes an approach without references to any particular moment of the output Y by assessing the density shift between the distribution of the output $p_Y(y)$ and the conditional density of the output given that one of the inputs X_i is set to a given value $p_{Y|X_i=x}(y)$ (represented by the red shaded area in Figure 2.2) as

$$s(X_i) = \int |p_Y(y) - p_{Y|X_i=x}(y)| dy \tag{2.16}$$

Taking the expected value w.r.t. X_i of the previous equation defines the moment independent importance measure of Borgonovo

$$S_i^\delta = \frac{1}{2}\mathbb{E}_{X_i}(s(X_i)) = \int p_{X_i}(x_i) \left(\int |p_Y(y) - p_{Y|X_i=x}(y)| dy \right) dx_i \tag{2.17}$$

which represents the normalized expected shift of the distribution of the output Y due to the input X_i. [Bor07] also extends the definition of the previous index to any group of inputs.

The moment independent importance measure defined by Borgonovo is actually a special case of a broader class of sensitivity measures called the *dissimilarity-based* measures [DV15; Rah16], which characterize the impact of X_i on Y by

$$S_i^d = \mathbb{E}_{X_i}(d(P_Y, P_{Y|X_i})) \tag{2.18}$$

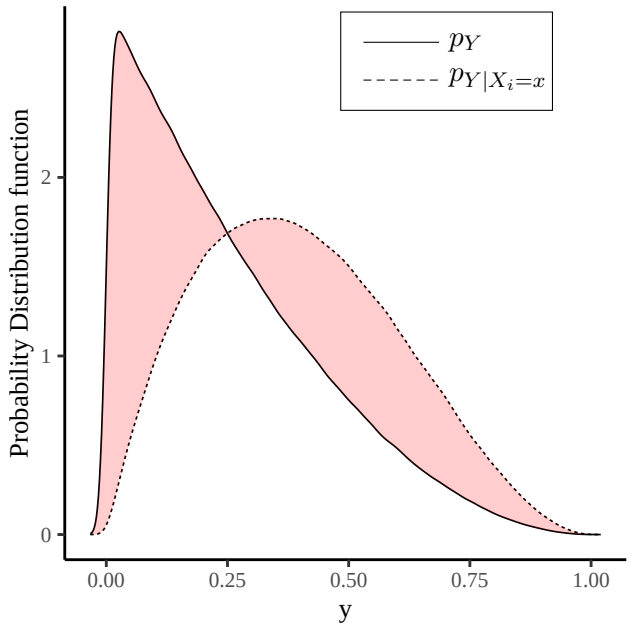

Figure 2.2 – Illustration of the Borgonovo importance measure. The shift between the two densities is measured by the red shaded area.

where $d(\cdot,\cdot)$ is a given dissimilarity measure between two probability distributions. The choice of the measure used directly impacts the associated sensitivity index. For example, considering the distance between the mean values of the two probability distributions

$$d(P_Y, P_{Y|X_i}) = (\mathbb{E}(Y) - \mathbb{E}(Y \mid X_i))^2 \tag{2.19}$$

one can obtain, after some calculations, the non-normalized main order Sobol index Equation (2.10)

$$S_i^d = \mathbb{V}(\mathbb{E}(Y \mid X_i)). \tag{2.20}$$

[DV15] defines a wide class of dissimilarity measures by the use of Csizar f-divergences, assuming that all input variables have an absolutely continuous distribution with respect to the Lebesgue measure on $\mathbb{R}$

$$d_{f_C}(P_Y, P_{Y|X_i}) = \int_{\mathbb{R}} f_C\left(\frac{p_Y(y)}{p_{Y|X_i}(y)}\right) p_{Y|X_i}(y)dy \tag{2.21}$$

where $f_C(\cdot)$ is a convex function such that $f_C(1) = 0$. The divergence function can be chosen among a wide list of functions, such as

- the Kullback-Leibler divergence: $f_C(t) = -\ln t$ or $f_C(t) = t \ln t$;
- the Pearson χ^2 divergence: $f_C(t) = (t-1)^2$ or $f_C(t) = t^2 - 1$;
- the Kolmogorov total variation distance: $f_C(t) = |t-1|$;

and so forth. Considering a given divergence function f_C and plugging it in Equation (2.18) yields the definition of the following sensitivity index:

$$S_i^{f_C} = \int_{\mathbb{R}^2} f_C\left(\frac{p_{X_i}(x)p_Y(y)}{p_{X_i,Y}(x,y)}\right) p_{X_i,Y}(x,y)dxdy \tag{2.22}$$

where $p_{X_i,Y}(x,y)$ is the joint probability distribution function of X_i and Y. [DV15] exhibits several advantages for the wide class of sensitivity indices defined this way. First of all, they measure the influence of an input X_i on the full distribution of the model output Y and not solely on a specific quantity of interest (as the variance for Sobol indices). Furthermore, the indices are non-negative and if X_i and Y are independent, $S_i^{f_C} = 0$. Last but not least, considering specific f_C functions, one retrieves well-known sensitivity indices from the literature. For example, with the Kolmogorov total variation distance, $f_C(t) = |t-1|$, one easily obtain the Borgonovo indices Equation (2.17). In addition, for the Kullback-Leibler divergence, $f_C(t) = -\ln t$, one reconstructs

$$S_i^{f_C} = \int_{\mathbb{R}^2} \ln\left(\frac{p_{X_i,Y}(x,y)}{p_{X_i}(x)p_Y(y)}\right) p_{X_i,Y}(x,y)dxdy \tag{2.23}$$

that is directly the Mutual Information $I(X_i, Y)$ between X_i and Y [Sha48], a dependence measure that relates to the entropy of the probability distribution functions. Dependence measures aim at quantifying the dependence between X and Y, with the property of being null only in the case of independence. The most familiar measure of dependence is the Pearson product-moment correlation coefficient which quantifies the degree of linear dependence between two random variables.

The bottleneck of these methods comes from the density estimation required in the computation of the indices. Typically, it is estimated from samples of the two distributions using Parzen windows or mixture of Gaussians, but this is directly affected by the curse of dimensionality. To circumvent this issue, some new methods avoid any density estimation and use representation of the distribution in Hilbert spaces. We call such methods *kernel-based* and describe them in the following section.

2.2 Kernel-based sensitivity analysis

Kernel representation is the backbone of many practical applications where the algorithm is expressed in terms of an inner product $\langle x, y\rangle$, since one can replace the inner product by a positive definite kernel. This generalizes the algorithm to nonlinear data treatment because the kernel is equivalent to an inner product in a nonlinearly mapped space, but the mapping does not have to be explicit (*kernel trick*). Among all data learning approaches that are kernelized, one can include *support vector machine* (SVM) [CV95] and *principle component analysis* (PCA) [Hot33]. In this section, concepts and notations required to understand kernel-based sensitivity analysis are introduced, starting from the basis of reproducing kernel Hilbert spaces (RKHS) and moving to probability embeddings and distances in the feature space.

2.2.1 Reproducing Kernel Hilbert Space and distribution embeddings

Let $\mathcal{X}$ be any space. For any positive definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a unique reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with $k(.,.)$ as its reproducing kernel. $\mathcal{H}$ is a Hilbert space of $\mathbb{R}$-valued functions on $\mathcal{X}$ endowed with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. By construction, the RKHS has two important properties:

1. $\forall x \in \mathcal{X}, k(x,\cdot) \in \mathcal{H}$,
2. for all $h \in \mathcal{H}$ and $x \in \mathcal{X}$, $h(x) = \langle h, k(x,\cdot)\rangle_{\mathcal{H}}$.

The function $k(x, \cdot)$ is called the Riesz representer of the evaluation functional at the point x. Following the reproducing property, one can write

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} \tag{2.24}$$

where $\phi(x)(\cdot) = k(x, \cdot) \in \mathcal{H}$. The map $x \in \mathcal{X} \to \phi(x) \in \mathcal{H}$ is called the feature map associated with $\mathcal{H}$ as any point in $\mathcal{X}$ is represented by the function $k(x, \cdot)$ in the feature space. Although we do not need to know $\mathcal{H}$ and $\langle , \rangle_{\mathcal{H}}$ explicitly, one can derive the feature map from the kernel [SSB+02]. [Aro50] states that for any positive definite function $k(\cdot, \cdot)$ on $\mathcal{X} \times \mathcal{X}$, there exists a unique RKHS with k as its reproducing kernel.

This means that the RKHS $\mathcal{H}$ is fully characterized by its associated reproducing kernel k, and it uniquely determines k, and vice versa. Many kernels can be found in the literature, depending on the nature of the data. Popular kernel functions on $\mathbb{R}^d$ include

- the polynomial kernel of order $p \in \mathbb{N}$: $k(x, x') = (\langle x, x' \rangle + c)^p$,
- the Gaussian RBF kernel, for $\theta > 0$: $k(x, x') = \exp\left(-\|x - x'\|_2^2 / \theta^2\right)$,
- the Laplace kernel, for $\theta > 0$: $k(x, x') = \exp\left(-\|x - x'\|_1 / \theta\right)$,

where θ is the lengthscale. As a generalization of the feature mapping of individual points, one can also map probability distributions into RKHS. Denote $\mathcal{M}_+^1(\mathcal{X})$ the space of probability measures over a measurable space $\mathcal{X}$. Then, we define the representer in $\mathcal{H}$ of any probability measure P by the mapping [BTA11; Smo+07]

$$\begin{aligned} \mu_P : \mathcal{M}_+^1(\mathcal{X}) &\to \mathcal{H} \\ P \mapsto \mu_P &:= \int k(x, \cdot) dP(x) = \int \phi(x) dP(x) \end{aligned} \tag{2.25}$$

denoted μ_P. This mapping is often called the *kernel mean embedding* of the probability space $\mathcal{M}_+^1(\mathcal{X})$. Following [Smo+07], the sufficient condition for the kernel mean embedding μ_P to exist and to belong to the RKHS $\mathcal{H}$ is $\mathbb{E}_X(\sqrt{k(X, X)}) < \infty$. Figure 2.3 depicts a schematic illustration of the kernel mean embeddings.

Understanding what information of the distribution is retained by the kernel mean embedding is a key aspect. In the case of the linear kernel $k(x, x') = \langle x, x' \rangle$, μ_P is simply the first statistical moment of P, whereas for the polynomial kernel $k(x, x') = (\langle x, x' \rangle + 1)^2$ μ_P contains both the first and the second moments of P. We can extend this with the polynomial kernel of order $p \in \mathbb{N}$, whose corresponding embedding incorporates moments of P up to the p-th order. In order to fully represent the distribution in the RKHS, $\mathcal{H}$ must be a characteristic RKHS, meaning that its corresponding kernel is also characteristic. This property is essential for kernel mean embedding as it ensures that no information is lost when mapping the distribution into the Hilbert space. Characteristic kernels were introduced in [FBJ04] as the kernels for which mapping Equation (2.25) is injective. [Fuk+08] showed that Gaussian and Laplace kernels are characteristic on $\mathbb{R}^d$ and properties of characteristic kernels were further studied in [Sri+08; SFL11]. When $\mu : P \mapsto \mu_P$ is injective, the Hilbert space in which the distribution is mapped should contain a sufficiently rich class of functions to fully differentiate all higher moments of neighboring distributions [Fuk+08].

One has rarely a complete knowledge of the true underlying distribution P and must rely on

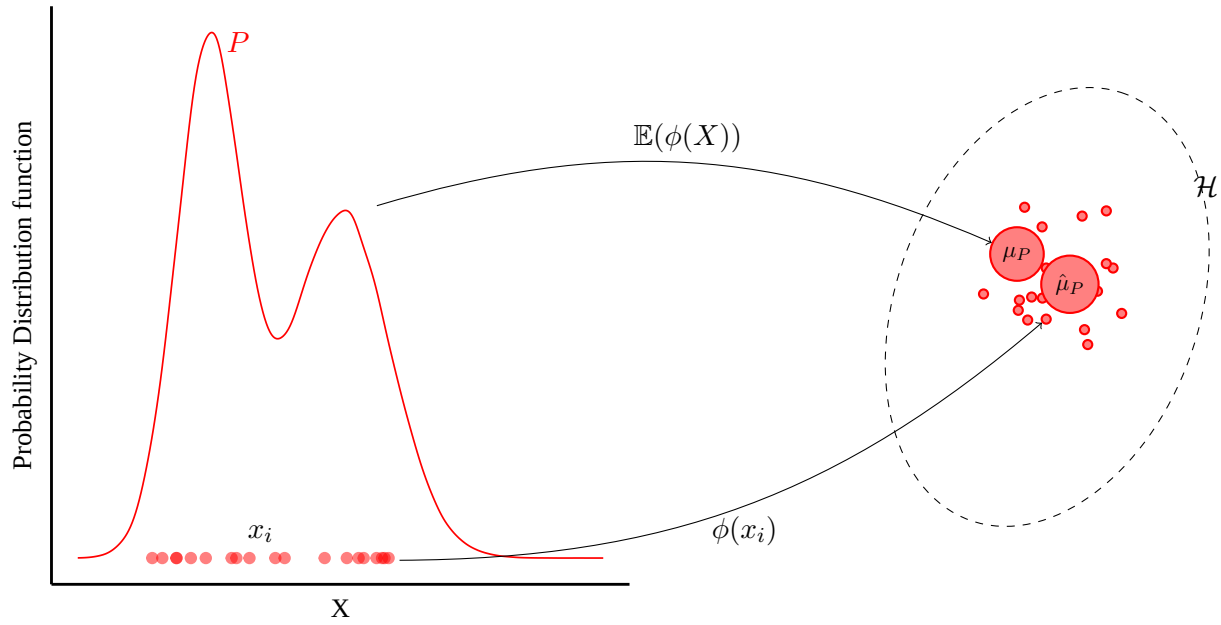

Figure 2.3 – Representation of a distribution embedding, mapped into a RKHS $\mathcal{H}$ using the expectation operation defined in Equation (2.25). The finite sample estimate follows Equation (2.26).

samples drawn from P. Assuming an independent and identically distributed sample $\mathbb{X} = (x^1, \dots, x^n) \sim P$, the standard estimator of the kernel mean embedding is an empirical average

$$\hat{\mu}_P = \frac{1}{n} \sum_{i=1}^{n} k(x^i, \cdot) \tag{2.26}$$

The estimator converges to the true kernel mean embedding almost surely as $n \to \infty$ [Sri+12]. [BTA11] shows that the convergence happens at a rate of $\mathcal{O}(n^{-1/2})$ which, interestingly, is independent of the dimension of X meaning that statistics based on kernel embeddings are less prone to the curse of dimensionality.

2.2.2 Maximum Mean Discrepancy

Kernel embeddings of probability measures provide a natural way to define distance between distributions as the distance between their embeddings in the Hilbert space. Assume X and Y are two random variables defined in $\mathcal{X}$ with probability distributions P_X and P_Y, respectively, and $\mathcal{H}$ a RKHS with kernel k. [Gre+12] defines the Maximum Mean Discrepancy (MMD) γ expressed as the distance between kernel mean embeddings in $\mathcal{H}$

$$\gamma(P_X, P_Y) = \|\mu_{P_X} - \mu_{P_Y}\|_{\mathcal{H}}. \tag{2.27}$$

We can express the MMD in terms of the associated kernel k by taking the square of Equation (2.27)

$$\begin{aligned} \gamma^2(P_X, P_Y) &= \langle \mu_{P_X} - \mu_{P_Y}, \mu_{P_X} - \mu_{P_Y} \rangle_{\mathcal{H}} \\ &= \|\mu_{P_X}\|^2_{\mathcal{H}} + \|\mu_{P_Y}\|^2_{\mathcal{H}} - 2\langle \mu_{P_X}, \mu_{P_Y} \rangle_{\mathcal{H}} \end{aligned} \tag{2.28}$$

$$= \mathbb{E}_{X,X'}(k(X,X')) + \mathbb{E}_{Y,Y'}(k(Y,Y')) - 2\mathbb{E}_{X,Y}(k(X,Y))$$

where $X, X' \sim P_X$ and $Y, Y' \sim P_Y$ are independent copies. The previous result directly comes from

$$\|\mu_{P_X}\|^2_{\mathcal{H}} = \langle \mathbb{E}_X(k(X,\cdot)), \mathbb{E}'_X(k(\cdot,X'))\rangle_{\mathcal{H}} = \mathbb{E}_{X,X'}(k(X,X')) \tag{2.29}$$

If k is characteristic, then $\gamma(P_X, P_Y) = 0$ if and only if $P_X = P_Y$.

This metric in terms of embeddings can be viewed as a particular instance of an integral probability metric (IPM) [Mül97] between P_X and P_Y on a measurable space $\mathcal{X}$

$$\gamma(P_X, P_X) = \sup_{f\in\mathcal{F}} \left(\int f(x) dP_X(x) - \int f(y) dP_Y(y) \right) \tag{2.30}$$

where $\mathcal{F}$ is a space of real-valued bounded measurable functions on $\mathcal{X}$. When the supremum is taken over functions in the unit ball in an RKHS $\mathcal{H}$, i.e. $\mathcal{F} = \{f, \|f\|_{\mathcal{H}} \leq 1\}$, using the reproducing property of $\mathcal{H}$ and the linearity of the inner product, one can show that the resulting metric is the MMD,

$$\begin{aligned}
\gamma(P_X, P_X) &= \sup_{\|f\|\leq 1} \left(\int f(x) dP_X(x) - \int f(y) dP_Y(y) \right) \\
&= \sup_{\|f\|\leq 1} \left(\langle f, \int k(x,\cdot) dP_X(x)\rangle - \langle f, \int k(y,\cdot) dP_Y(y)\rangle \right) \\
&= \sup_{\|f\|\leq 1} \left(\langle f, \mu_{P_X} - \mu_{P_Y}\rangle \right) \\
&= \|\mu_{P_X} - \mu_{P_Y}\|_{\mathcal{H}}
\end{aligned} \tag{2.31}$$

When considering other spaces $\mathcal{F}$, one can obtain other well-known distances between distributions. For example, if $\mathcal{F} = \{1_{(\infty,t]}\}$, i.e. the max norm of the difference between their cumulative distributions, we obtain the Kolmogorov distance between distribution. Setting $\mathcal{F} = \{f, \|f\|_L \leq 1\}$, where $\|f\|_L = \sup\{|f(x) - f(y)|/\rho(x,y), x \neq y \in \mathcal{X}\}$ is the Lipschitz semi-norm of a real-valued function f where ρ is some metric on a compact space $\mathcal{X}$, yields the Wasserstein distance.

Considering i.i.d. samples $\mathbb{X} = \{x^1, \ldots, x^m\} \sim P_X$ and $\mathbb{Y} = \{y^1, \ldots, y^n\} \sim P_Y$, one can write an unbiased estimator of the MMD from Equation (2.28) entirely in terms of k as a sum of two U-statistics and a sample average [Bor+06]

$$\begin{aligned}
\gamma_u^2(\mathbb{X}, \mathbb{Y}) = &\frac{1}{m(m-1)} \sum_{i=1}^{m}\sum_{j\neq i}^{m} k(x^i, x^j) + \frac{1}{n(n-1)} \sum_{i=1}^{n}\sum_{j\neq i}^{n} k(y^i, y^j) \\
&- \frac{2}{mn} \sum_{i=1}^{m}\sum_{j=1}^{n} k(x^i, y^j)
\end{aligned} \tag{2.32}$$

with a computational cost of $\mathcal{O}((m+n)^2)$. When $m = n$, a slightly simpler estimate may be used. Let $\mathbb{Z} = \{z^1, \ldots, z^m\}$ be m i.i.d. random variables, with $z = (x, y) \sim P_X \times P_Y$. The estimate goes as

$$\gamma_u^2(\mathbb{X}, \mathbb{Y}) = \frac{1}{m(m-1)} \sum_{i\neq j}^{m} h(z^i, z^j) \tag{2.33}$$

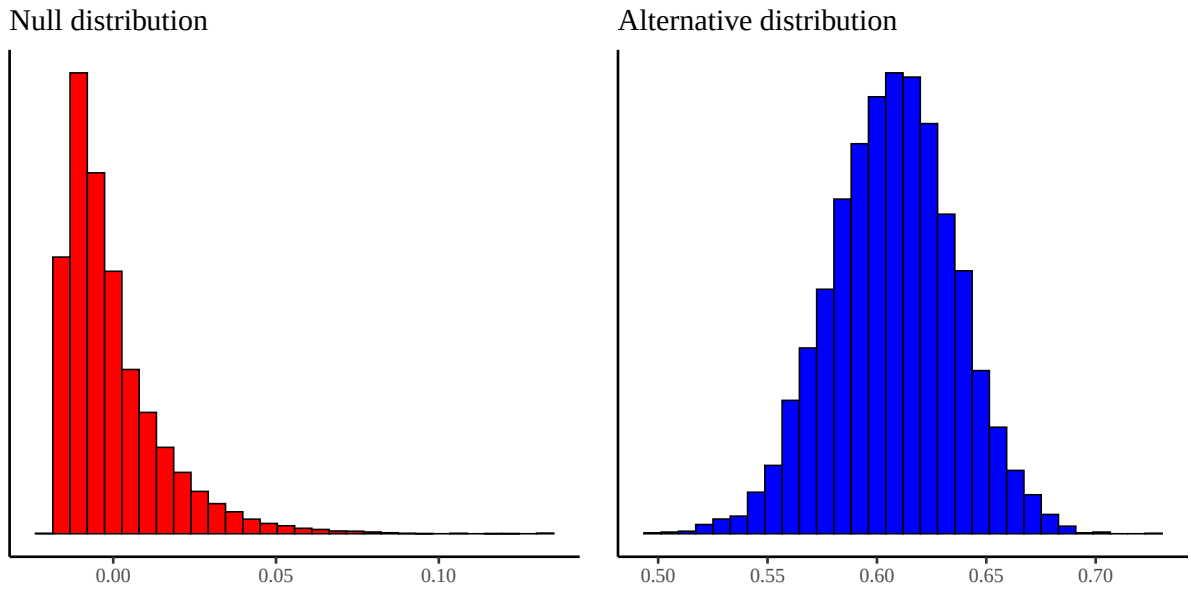

Figure 2.4 – Left: Empirical null distribution of the unbiased MMD γ_u^2 (Equation (2.32)), with P_X and P_Y both univariate Gaussians with unit standard deviation with 50 samples from each. Right: Empirical alternative distribution of the unbiased MMD γ_u^2 (Equation (2.32)), with P_X a univariate Gaussian with unit standard deviation and P_Y a univariate Gaussian with standard deviation 3 with 50 samples from each. The null distribution has a long tailed form while the alternative distribution is Gaussian. Both histograms were obtained using 10000 independent samples.

which is a one-sample U-statistic with

$$h(z^i, z^j) = k(x^i, x^j) + k(y^i, y^j) - k(x^i, y^j) - k(x^j, y^i) \tag{2.34}$$

Both estimators of $\gamma_u^2(\mathbb{X}, \mathbb{Y})$ may be negative since they are unbiased estimators of $\gamma_u^2(P_X, P_Y)$. The biased counterpart $\gamma_b^2(\mathbb{X}, \mathbb{Y})$ can be obtained using V-statistics. Following [Gre+12], the distribution of γ_u^2 when $P_X = P_Y$, called the *null distribution*, has a complicated form, expressed as an infinite sum of χ^2 variables. On the counterpart, the *alternative distribution*, namely the distribution of unbiased empirical estimator of the MMD when $P_X \neq P_Y$, converges in distribution to a Gaussian distribution

$$\sqrt{m}(\gamma_u^2(\mathbb{X}, \mathbb{Y}) - \gamma^2(P_X, P_Y)) \xrightarrow{p} \mathcal{N}(0, \sigma_u^2) \tag{2.35}$$

where

$$\sigma_u^2 = 4\left(\mathbb{E}_Z(\mathbb{E}'_Z(h(Z, Z'))^2) - (\mathbb{E}_{Z,Z'}(h(Z, Z')))^2\right). \tag{2.36}$$

Figure 2.4 shows empirical estimate of both distributions for the unbiased estimator γ_u^2. [Sri+12] shows that compared to the other distances, the MMD enjoys a rapid convergence and the rate is independent of the dimension, contrary to other metrics such as the Wasserstein distance which suffers from a rate that depends on d.

The estimator in Equation (2.32) incorporates as much information as possible from the data, which comes at a quadratic cost since all pairs of samples are considered. However, sometimes, a statistic with a faster computation can be needed without losing too much accuracy. Assuming

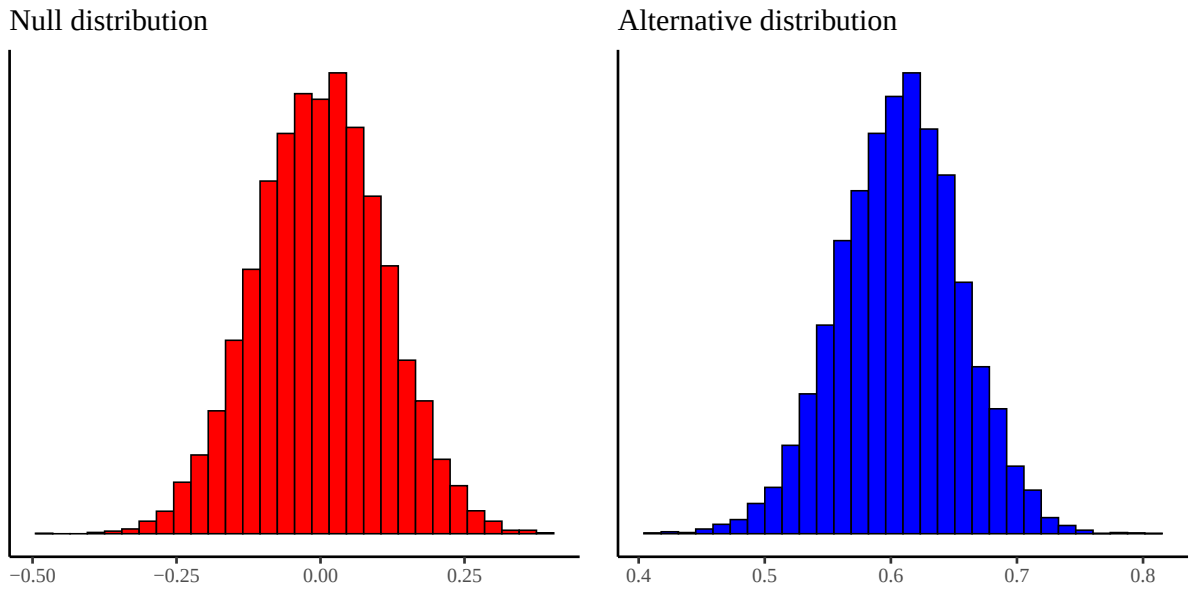

Figure 2.5 – Left: Empirical null distribution of the unbiased MMD γ_l^2 (Equation (2.37)), with P_X and P_Y both univariate Gaussians with unit standard deviation with 50 samples from each. Right: Empirical alternative distribution of the unbiased MMD γ_u^2 (Equation (2.32)), with P_X a univariate Gaussian with unit standard deviation and P_Y a univariate Gaussian with standard deviation 3 with 50 samples from each. Both distributions with this estimator are Gaussians. Both histograms were obtained using 10000 independent samples.

$m_2 = m/2$, $m = n$, and $h(z^{2i-1}, z^{2i})$ defined as in Equation (2.34), [Gre+12] proposes an unbiased estimator of the MMD that can be estimated in linear time:

$$\gamma_l^2(\mathbb{X}, \mathbb{Y}) = \frac{1}{m_2} \sum_{i=1}^{m_2} h((x^{2i-1}, y^{2i-1}), (x^{2i}, y^{2i})) \tag{2.37}$$

with a computational cost of $\mathcal{O}(m_2)$. Even though it is expected that γ_l^2 has a higher variance than γ_u^2, it is computationally more appealing and it has a useful property: both the null and the alternative distributions are Gaussian. The null distribution has a zero mean while the alternative distribution has a positive mean, see Figure 2.5.

Since the statistic is just the average of independent random variables, the central limit theorem [Ser81] allows to show that γ_l^2 converges in distribution to a Gaussian according to

$$\sqrt{m}(\gamma_l^2(\mathbb{X}, \mathbb{Y}) - \gamma^2(P_X, P_Y)) \xrightarrow{p} \mathcal{N}(0, \sigma_l^2) \tag{2.38}$$

where

$$\sigma_l^2 = 2\left(\mathbb{E}_{Z,Z'}(h^2(Z, Z')) - (\mathbb{E}_{Z,Z'}(h(Z, Z')))^2\right). \tag{2.39}$$

The null distribution is Gaussian with this estimator, therefore approximating it is quite easy since only the estimation of the variance is required. This is possible in linear time by simply computing an unbiased empirical variance estimate $\hat{\sigma}_l^2$ using the same set of samples $h(z^i, z^{i'})$,

i.e.

$$\hat{\sigma}_l^2 = \frac{1}{m_2 - 1} \sum_{i=1}^{m_2} \left(h(z^{2i-1}, z^{2i}) - \frac{1}{m_2} \sum_{j=1}^{m_2} h(z^{2j-1}, z^{2j}) \right)^2 . \tag{2.40}$$

2.2.3 Two-sample testing with the MMD

Since all estimators of the MMD are small when $P_X = P_Y$ and large otherwise, they are well-suited in the scope of *two-sample testing*: a statistical hypothesis test for equality between two samples. In practice, we consider the following:

- null hypothesis $H_0 : P_X = P_Y$,
- alternative hypothesis $H_A : P_X \neq P_Y$.

If $\gamma_u^2(\mathbb{X}, \mathbb{Y})$ is "far from zero", one can reject the null hypothesis and accept it when it is "close to zero". Hence, one need to determine whether $\gamma_u^2(\mathbb{X}, \mathbb{Y})$ shows a statistically significant difference between distributions. Given $\mathbb{X} \sim P_X$ and $\mathbb{Y} \sim P_Y$, two i.i.d. samples, the main idea is to compare a test statistic $\mathcal{T}(\mathbb{X}, \mathbb{Y})$ with a particular threshold: if the threshold is exceeded, the null hypothesis H_0 is rejected. Since we work with finite samples, returning incorrect answers is a possibility. Two types of errors exist:

- a type I error is made when $H_0 : P_X = P_Y$ is wrongly rejected. That is, the test says that the samples are from different distributions when they are not.
- a type II error is made when $H_0 : P_X = P_Y$ is wrongly accepted. That is, the null hypothesis is accepted despite the considered distributions being different.

Figure 2.6 illustrates the two errors for a given α level. A good test has often a low type II error since one is usually more interested in finding difference between samples. A test that always rejects the null hypothesis would have zero type II error but may have a large type I error. Thus, one needs to control the type I error while trying to minimize the type II error. To do so, one controls the level α of a test, defined as an upper bound on the probability of a type I error and the threshold is chosen such that $P(\mathcal{T}(X, Y) > t) \leq \alpha$. A test is said to be consistent if for a set upper bound α, it reaches zero type II error in the infinite sample limit. Both the quadratic and linear time statistics are useful in that scope but for different scenarios [Gre+12]:

- the quadratic time statistic γ_u is useful with finite samples of data, it produces results that are more accurate since it considers all the data, but faces a limitation regarding to the size of the sample, especially for large ones,
- the linear time statistic γ_l comes in handy in the "infinite" data case, when the amount of sample points is nearly unlimited, but the computational time available is not. Since data needs not to be stored, the statistic can be applied to online data.

Figure 2.7 shows how the variance of the linear time MMD is much larger than the quadratic one for fixed sample size when $P_X = P_Y$.

The null distribution is the backbone of a two-sample test. Designing the test is clear: the null distribution must be approximated in some ways, or may be analytically known and the test statistic computed on some provided data. Then, the null hypothesis is rejected when the test statistic lies above a given threshold. Since different estimators have different null distributions

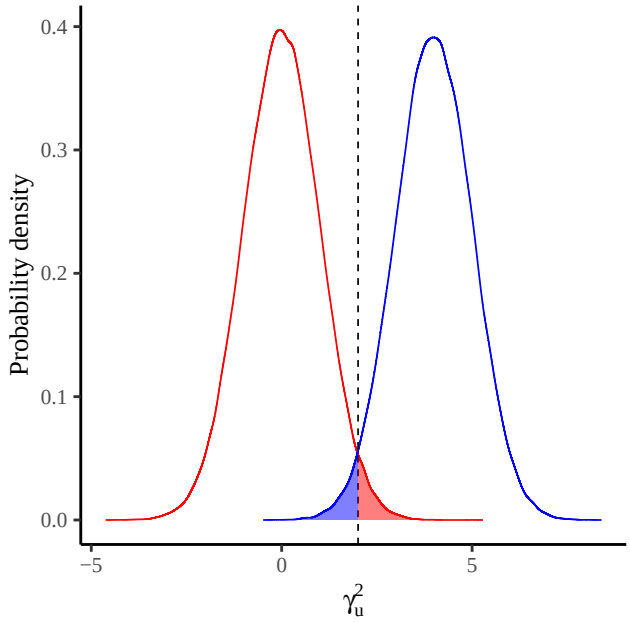

Figure 2.6 – Depiction of the two-sample testing. Statistics generated by $H_0 : P_X = P_Y$ come from the null distribution in red. Those generated by $H_A : P_X \neq P_Y$ come from the alternative distribution in blue. The dashed line at the $(1-\alpha)$-quantile of H_0 corresponds to the threshold for testing: if a statistic lies above that threshold, the probability that it was generated under H_0 is less than α. Red area represents type I error, wrongly rejecting H_0 for samples generated by the null hypothesis. Blue area represents type II error, wrongly accepting H_0 for samples from the alternative distribution that lie under the threshold.

(see Figures 2.4 and 2.5), finding an efficient general method well-suited for all estimators can be hard. In the following sections, multiple strategies are highlighted, depending on which estimator is considered between γ_u and γ_l.

Gaussian approximation of the linear MMD

As already mentioned earlier, when using the linear estimator of the MMD in a two-sample test, the null distribution is in fact a Gaussian, see Equation (2.38). It has a zero mean and its variance can be empirically estimated following Equation (2.40). Approximation of the null distribution simply relies on plugging a zero mean and the simulated variance into a normal distribution, leading to an easy computation of thresholds or p-values. Given an empirical $\hat{\sigma}_l^2$, the threshold for a test level α is given by

$$\Phi^{-1}_{\mu,\sigma_l^2}(1-\alpha) \tag{2.41}$$

with $\Phi^{-1}_{\mu,\sigma^2} : [0,1] \to \mathbb{R}$ is the inverse normal cumulative distribution for mean μ and variance σ^2, which returns the value x corresponding to the $(1-\alpha)$ quantile of the considered normal distribution. Similarly, any p-value for a statistic estimate can be computed by evaluating its position in the normal distribution using the normal cumulative distribution function Φ.

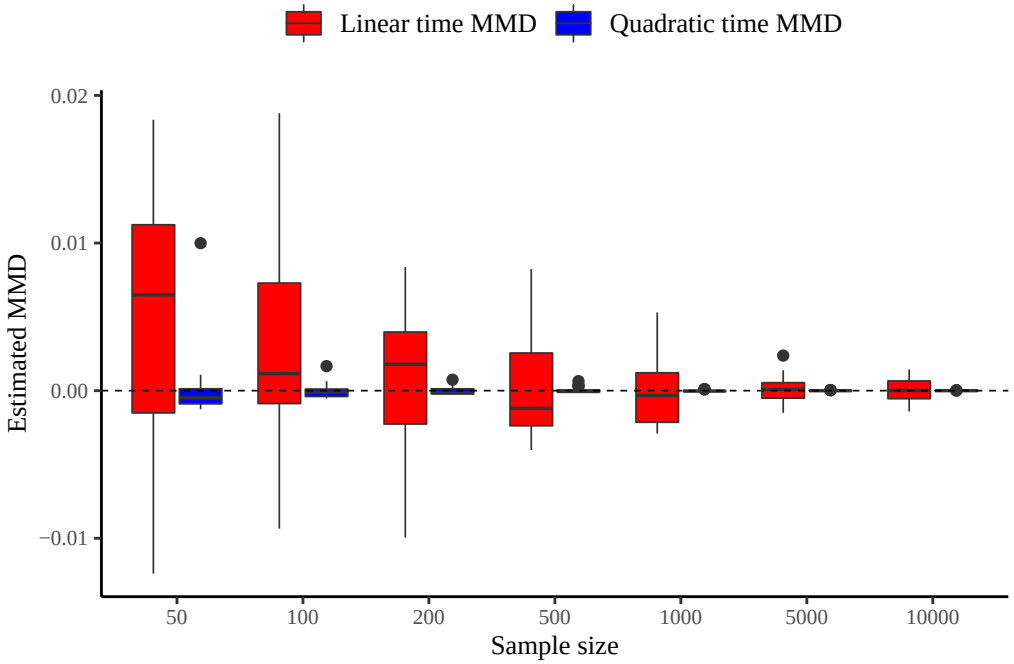

Figure 2.7 – Representation of the estimation of the MMD statistics under the null hypothesis in terms of the considered sample size. Both samples come from a standard univariate Gaussian. The computation is repeated 20 times and estimated MMD are depicted using boxplots. The linear time MMD is in red while the quadratic MMD is in blue. The dashed line corresponds to a value of 0. The estimation of the linear time MMD stabilizes for a sample size of 5000 points.

Pearson approximation of the quadratic MMD

Another method is based on the approximation of the null distribution by fitting Pearson curves to its first four moments. By taking advantage of the U-statistic, when $m = n$, we have

$$\begin{aligned}\mathbb{E}((\gamma_u^2)^2) &= \frac{2}{m(m-1)}\mathbb{E}_{Z,Z'}(h^2(Z,Z')), \\ \mathbb{E}((\gamma_u^2)^3) &= \frac{8(m-2)}{m^2(m-1)^2}\mathbb{E}_{Z,Z'}(h(Z,Z')\mathbb{E}_{Z''}(h(Z,Z'')h(Z',Z'')) + \mathcal{O}(m^{-4})\end{aligned} \tag{2.42}$$

where $h(Z, Z')$ is defined following Equation (2.34) and Z' and Z" are independent copies of Z. The fourth moment is usually not computed as it is both small and expensive to estimate. Instead, the kurtosis is replaced with a lower bound $\text{kurt}(\gamma_u^2) \geq (\text{skew}(\gamma_u^2))^2 + 1$. [Gre+12] shows that fitting Pearson curves gives a good match in the upper quantiles, where the test threshold is computed.

[Gre+09] proposes an alternative to the precedent strategy, with a two-parameter Gamma approximation of the cumulative distribution function of the biased MMD estimate. Since the null distribution of γ_u^2 approaches an infinite weighted sum of independent χ^2 random variables, the method is based on an approximation of the null distribution based on empirical eigenvalues estimates. More information on these methods can be found in [Gre+09; Gre+12].

Estimation of the null-distribution through resampling

One method can be applied for any two-sample test: sampling the null-distribution using resampling methods. Assuming that we have a given sample of observations and related output evaluations $\{\mathbb{X}, \mathbb{Y}\}$ with a corresponding statistic $\mathcal{T}(\mathbb{X}, \mathbb{Y})$, we can generate samples under the null distribution through random permutations [ET94]: a shuffled dataset is generated $\{\tilde{\mathbb{X}}, \tilde{\mathbb{Y}}\}$, where $\tilde{\mathbb{X}}$ and $\tilde{\mathbb{Y}}$ are randomly sampled from the initial population with replacement, denoted as a *bootstrapped* samples, and the statistic $\mathcal{T}(\tilde{\mathbb{X}}, \tilde{\mathbb{Y}})$ is estimated. This comes at a rather high cost since the statistic has to be recomputed for each permuted samples, hence multiplying the complexity by the number of considered permutations. Once a sufficient number of samples are obtained under the null hypothesis, these can be used to compute a threshold or a p-value. Given a test level α, the threshold is simply obtained as the $(1-\alpha)$-quantile in the samples. The statistic $\mathcal{T}(\mathbb{X}, \mathbb{Y})$ can be compared against the threshold to accept (if above the threshold) or reject (otherwise) the null hypothesis $H_0 : P_X = P_Y$. Similarly, a p-value for a given statistic can be computed by assessing the relative position of the test statistic $\mathcal{T}(\mathbb{X}, \mathbb{Y})$ compared to the samples under the null hypothesis. Assuming that we have done p bootstraps, we have

$$p_{val} = \frac{1}{p} \sum_{i=1}^{p} 1_{\mathcal{T}(\tilde{\mathbb{X}}^i, \tilde{\mathbb{Y}}^i) > \mathcal{T}(\mathbb{X}, \mathbb{Y})} \tag{2.43}$$

where $\{\tilde{\mathbb{X}}^i, \tilde{\mathbb{Y}}^i\}$ corresponds to the i-th bootstrapped sample. Given the p-value p_{val}, the null hypothesis $H_0 : P_X = P_Y$ is rejected if this value is larger than a given test level α. Comparing the statistic against the threshold and comparing the p-value against a desired test level is exactly the same thing. Note that this method is a distribution-free hypothesis test which is why we can use it for both linear and quadratic time MMD statistics.

Kernel hyperparameters

As described in the previous section, a statistical test should have a low type II error for a fixed type I error. In theory, characteristic kernels can distinguish any two distributions, but in practice their parameters have a large impact on the test's type II error. For example, the squared exponential kernel only parameter is its bandwidth σ, which basically determines the length scale at which the kernel looks at data. In order to get a low type II error, this length scale has to be set to the size where differences in the two underlying distributions P_X and P_Y appear. If being set too large or too small, the kernel is not able to detect these differences. One of the first methods proposed to choose the width of a squared exponential kernel is to use the median distance of the underlying data $\mathbb{X}$, as $\sigma^2 = \text{median}\{\|x^i - x^j\|^2\}$ with $\mathbb{X} = \{x^1, \dots, x^n\}$ and $i, j = 1, \dots, n$, see [Gre+05a]. Several empirical studies show that this heuristic works well in practice. The main advantage of this method is that it is easy to compute; all pairwise distances of data have to be computed and the median must be estimated. Since the median is a stable statistic, a low amount of samples can be sufficient. However, this method can only applied to the squared exponential kernel, which is the main downside of it.

Other strategies to define the value of the kernel hyperparameters were defined in [Fuk+09; Gre+12]. For example, the kernel can be chosen as the one that maximizes the test statistic or chosen so it maximizes the two-sample test power and it minimizes the probability of making a type II error. However, choosing a good kernel function remains an open field of research.

2.2.4 Hilbert Schmidt independence criterion

Definition

Another application of the MMD is determining whether two variables X and Y are independent, which can be interpreted as a two-sample test. Recall that X and Y are as independent if and only if their joint distribution P_{XY} factorizes as $P_{XY} = P_X \otimes P_Y$ (written $P_X P_Y$ for shorter notations from now on). The MMD for testing independence can be define as the distance between the kernel mean embeddings of P_{XY} and $P_X P_Y$, defined as [Smo+07]

$$\begin{aligned}\mu_{P_{XY}} &= \mathbb{E}_{XY}(v((X,Y),\cdot)) \\ \mu_{P_X P_Y} &= \mathbb{E}_X \mathbb{E}_Y (v((X,Y),\cdot)).\end{aligned}$$

Here we assume $\mathcal{H}$ is a RKHS of functions from $\mathcal{X}$ to $\mathbb{R}$ with kernel k. Likewise, we define a second RKHS, $\mathcal{G}$, of functions from $\mathcal{Y}$ to $\mathbb{R}$ with kernel l. Since a product of kernels is a kernel, we construct a kernel v on the product space $\mathcal{X} \times \mathcal{Y}$ with corresponding RKHS $\mathcal{V}$

$$v((x,y),(x',y')) = k(x,x')l(y,y'). \tag{2.44}$$

From this, we can define the MMD test statistic for testing independence as

$$\begin{aligned}\gamma^2(P_{XY}, P_X P_Y) =& \|\mu_{P_{XY}} - \mu_{P_X P_Y}\|^2_{\mathcal{H}\otimes\mathcal{G}} \\ =& \mathbb{E}_{X,Y}\mathbb{E}_{X',Y'}k(X,X')l(Y,Y') + \mathbb{E}_X\mathbb{E}_{X'}\mathbb{E}_Y\mathbb{E}_{Y'}k(X,X')l(Y,Y') \\ & - 2\mathbb{E}_{X,Y}\mathbb{E}_{X'}\mathbb{E}_{Y'}k(X,X')l(Y,Y')\end{aligned} \tag{2.45}$$

This in fact is directly equal to a quantity called the Hilbert Schmidt independence criterion (HSIC) [Gre+05b], defined as the squared Hilbert Schmidt norm of the cross-covariance operator associated to the joint distribution P_{XY}. The cross-covariance is a linear operator $C_{XY} : \mathcal{G} \to \mathcal{H}$ defined for every $f_h \in \mathcal{H}$ and $f_g \in \mathcal{G}$ as

$$\langle f_h, C_{XY} f_g \rangle_{\mathcal{H}} = \mathbb{E}_{XY}(f_h(X)f_g(Y)) - \mathbb{E}_X(f_h(X)\mathbb{E}_Y(f_g(Y)) = \mathrm{Cov}(f_h(X), f_g(Y)). \tag{2.46}$$

Hence, the cross-covariance operator generalizes the covariance matrix by representing higher order correlations between X and Y through nonlinear kernels. Deriving the Hilbert-Schmidt norm of the cross-covariance operator yields

$$\|C_{XY}\|^2_{HS} = \sum_{i,j} \langle u_i, C_{XY} v_j \rangle_{\mathcal{H}} \tag{2.47}$$

with $(u_i)_{i\geq 0}$ and $(v_j)_{j\geq 0}$ are orthonormal bases of $\mathcal{H}$, respectively $\mathcal{G}$. Equation (2.47) is the HSIC, which can be expressed in terms of kernel as [Gre+05b]

$$\begin{aligned}\mathrm{HSIC}(X,Y) =& \|C_{XY}\|^2_{HS} \\ =& \mathbb{E}_{X,Y}\mathbb{E}_{X',Y'}k(X,X')l(Y,Y') + \mathbb{E}_X\mathbb{E}_{X'}\mathbb{E}_Y\mathbb{E}_{Y'}k(X,X')l(Y,Y') \\ & - 2\mathbb{E}_{X,Y}\mathbb{E}_{X'}\mathbb{E}_{Y'}k(X,X')l(Y,Y')\end{aligned} \tag{2.48}$$

which is directly equivalent to Equation (2.45). The HSIC is equal to 0 if and only if X and Y are independent, when the associated RKHS $\mathcal{H}$ and $\mathcal{G}$ are characteristic. Considering an i.i.d.

sample $\{\mathbb{X}, \mathbb{Y}\} = \{(x^1, y^1), \dots, (x^n, y^n)\}$, an empirical estimator of the HSIC is

$$\text{HSIC}(\mathbb{X}, \mathbb{Y}) = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} K_{ij} L_{ij} + \frac{1}{n^4} \sum_{1 \leq i,j,p,q \leq n} K_{ij} L_{pq} - \frac{2}{n^3} \sum_{1 \leq i,j,p \leq n} K_{ij} L_{jp} \tag{2.49}$$

with $K, L \in \mathbb{R}^{n \times n}$, $K_{ij} = k(x^i, x^j)$, $L_{ij} = l(y^i, y^j)$ are Gram matrices. Equivalently, Equation (2.49) can be written under a more compact form:

$$\text{HSIC}(\mathbb{X}, \mathbb{Y}) = \frac{1}{n^2} \text{tr}(KHLH) \tag{2.50}$$

with $H \in \mathbb{R}^{n \times n}$ is a centering matrix defined $H_{ij} = \delta_{ij} - n^{-1}$, with δ_{ij} the Kronecker symbol between i and j that is equal to 1 if $i = j$ and 0 otherwise. This estimator can be computed in $\mathcal{O}(n^2)$ time.

Independence testing with the HSIC

Gamma approximation The HSIC is used as a statistical measure to test the null hypothesis $H_0 : X$ and Y are independent against its alternative $H_1 : X$ and Y are dependent. Under the assumption of independence between X and Y, the asymptotic distribution of $N \times \text{HSIC}(\mathbb{X}, \mathbb{Y})$ is an infinite sum of independent χ^2 random variables which can be approximated by a two-parameter Gamma distribution

$$n \times \text{HSIC}(\mathbb{X}, \mathbb{Y}) \sim \frac{x^{\alpha-1} \exp -x/\beta}{\beta^\alpha \Gamma(\alpha)} \tag{2.51}$$

where $\alpha = \frac{\mathbb{E}(\text{HSIC}(\mathbb{X},\mathbb{Y}))^2}{\mathbb{V}(\text{HSIC}(\mathbb{X},\mathbb{Y}))}$ and $\beta = \frac{n\mathbb{V}(\text{HSIC}(\mathbb{X},\mathbb{Y}))}{\mathbb{E}(\text{HSIC}(\mathbb{X},\mathbb{Y}))}$. More details on the computation of α and β can be found in [Gre+08]. In practice, the independent test rejects the null hypothesis H_0 when the p-value of the Gamma distribution associated to $n \times \text{HSIC}(\mathbb{X}, \mathbb{Y})$ is lower than the significance level.

Permutation-based approximation A non-parametric strategy based on permutations to test for independence can also be used, similarly to what was defined in Section 2.2.3 for the MMD. Let $\{\mathbb{X}, \mathbb{Y}\}$ be a sample of observations and evaluations, and $\mathcal{T}(\mathbb{X}, \mathbb{Y}) = \text{HSIC}(\mathbb{X}, \mathbb{Y})$ the statistical measure considered. Samples under the null distribution are generated through random permutations and for each pair of shuffled sampled $\{\tilde{\mathbb{X}}, \tilde{\mathbb{Y}}\}$ the statistic $\mathcal{T}(\tilde{\mathbb{X}}, \tilde{\mathbb{Y}})$ is estimated. Note that the random permutation eliminates the dependency between X and Y (if it exists), and therefore $\mathcal{T}(\tilde{\mathbb{X}}, \tilde{\mathbb{Y}})$ would take a value close to zero.

2.2.5 Application to global sensitivity analysis

In the global sensitivity framework, [DV15] defines a sensitivity index based on the HSIC dependence measure which characterizes how independent a given input X_i and the output Y are as

$$S_i^{\text{HSIC}} = \frac{\text{HSIC}(X, Y)}{\sqrt{\text{HSIC}(X, X)\text{HSIC}(Y, Y)}} \tag{2.52}$$

Normalizing the HSIC statistic allows to bound it in $[0, 1]$ for an easier interpretation. Usually, Equation (2.52) can be directly estimated by plugging Equation (2.50), leading to

$$\hat{S}_i^{\text{HSIC}} = \frac{\text{HSIC}(\mathbb{X}, \mathbb{Y})}{\sqrt{\text{HSIC}(\mathbb{X}, \mathbb{X})\text{HSIC}(\mathbb{Y}, \mathbb{Y})}} \tag{2.53}$$

The main advantages for this kernel-based method, among others, remain its low computational cost compared to Sobol indices and the ability to deal with large number of inputs.

[DLM16] extended this work to screening purposes and also applies it to spatial outputs [DLM17], against classical variance-based strategies. All the different strategies aforementioned are being compared on a real-life application and it highlights the consistent results obtained with the HSIC with less observations than the variance-based methods. In [MML19], the HSIC is used in a two-levels global sensitivity analysis which is useful in cases where the distribution of the inputs or the parameters of the said distributions are also unknown.

2.3 Goal-oriented sensitivity analysis

As stated before, variance-based global methods, such as Sobol indices, and moment-independent methods provide information on the influence of an input in the full design domain by characterizing which input or group of inputs cause the output to vary the most. However, in different cases or studies, it turns out that one is more interested in finding which variables are important *i)* in order to respect the constraints and have an interesting value for objective function, which is particularly true when constraints are difficult to satisfy, and *ii)* when the constraints are satisfied and the objective function produces high-performance values. In that case, these methods appear as limited since they focus on a specific quantity of interest of the output and alternative solutions were proposed to characterize the sensitivity of inputs for these specific cases.

2.3.1 Sensitivity analysis based on contrast functions

In a first place, [FKR16] coined the term *goal-oriented sensitivity analysis* (GOSA) and defined a new approach based on the goal of the study (e.g. find which variables have an importance when the output is above a given value). Their idea comes from the fact that *(...) the importance of an input variable may vary depending on what the quantity of interest is.*

Let Θ be some generic set and P_Y a probability measure defined on the space $\mathcal{Y}$, they define a *contrast function* as any function ψ

$$\begin{aligned} \psi : \Theta &\to L_1(P_Y) \\ \theta &\mapsto \psi(\cdot, \theta) : y \in \mathcal{Y} \mapsto \Psi(\theta, y) \end{aligned} \tag{2.54}$$

such that

$$\theta^* = \arg\min_{\theta \in \Theta} \Psi(\theta, Y) \tag{2.55}$$

where $\Psi : \theta \mapsto \mathbb{E}_Y[\psi(Y, \theta)]$ is the average contrast function. When $\Theta = \mathbb{R}$, the features are scalar, e.g. the mean ($\theta = \mathbb{E}(Y)$), and the corresponding mean-contrast is $\psi(\theta, y) = (y - \theta)^2$. If $\Theta = [0, 1]$, then the feature is a probability of exceeding a given threshold s, $\theta = \mathbb{P}(Y > s)$ and the corresponding contrast function is given by $\psi(\theta, y) = (1_{y>s} - \theta)^2$. Finally, considering the

following feature of the output Y conditionally to a given X_i, $\theta_i(x) = \arg\min_\theta \mathbb{E}(\psi(\theta, Y) \mid X_i = x)$, [FKR16] proposes the ψ-index of an output Y with respect to a variable X_i and a contrast function ψ as

$$S_i^\psi = \frac{\mathbb{E}[\psi(\theta^*, Y)] - \mathbb{E}_{(X_i,Y)}[\psi(\theta_i(x_i)]}{\mathbb{E}[\psi(\theta^*, Y)] - \mathbb{E}[\min_\theta \psi(\theta, Y)]}. \tag{2.56}$$

ψ-indices have several properties that the authors highlight. First of all, $S_i^\psi \in [0, 1]$, with $S_i^\psi = 0$ means independence between Y and X_i while $S_i^\psi = 1$ and $S_j^\psi = 0, j \neq i$ means we can rewrite $Y = f(X_i)$. The second important aspect is that when considering the mean-contrast $\psi : (\theta, y) \mapsto (y - \theta)^2$, the authors retrieve the global Sobol index, using $\theta^* = \mathbb{E}(Y)$ and $\theta_i(x) = \mathbb{E}(Y \mid X_i = x)$, thus yielding

$$S_i^\psi = \frac{\mathbb{V}(\mathbb{E}(Y \mid X_i))}{\mathbb{V}(Y)} \tag{2.57}$$

which is directly the first order Sobol index expressed in Equation (2.11). Their adaptability to the considered quantity of interest of the output, with variance among all of these, is a major advantage. Yet, for some specific contrast, such as the α-quantile contrast function $\psi(\theta) = \mathbb{E}(Y - \theta)(\alpha - 1_{Y \leq \theta})$, their estimation appears as troublesome and is an ongoing field of research [MDN18; Bro+17].

2.3.2 Sobol on the indicator function

In the reliability analysis framework, the main problem revolves around being able to find the probability of occurrence of an undesirable event, often written as the output Y exceeding a threshold q, namely the probability of failure P_f. It can be seen as some kind of goal-oriented problem, where classical sensitivity measures give possible wrongful information on the influence of the inputs. Here what matters is finding if an input has an impact on the occurrence of undesirable events. Transposed to an optimization problem, it can be seen as assessing the relevance of certain inputs to obtain output value with good performance which comply with the constraints when they exist.

In order to do so, the focus is made on a particular quantity of interest of the output for reliability problem: the indicator function of the failure domain $1_{\mathcal{D}_f}(\cdot)$. The idea initially came from the fact that the probability of failure P_f can be expressed as the expectation of the previously defined indicator function $P_f = \mathbb{E}(1_{\mathcal{D}_f}(X))$, and [Luy+12] highlighted the relation between probability of failures and mathematical expectations:

$$P_f - P_{f|X_i} = \mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X})) - \mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X}) \mid X_i) \tag{2.58}$$

where $P_{f|X_i}$ is the conditional probability failure, leading to a new sensitivity index

$$\begin{aligned} \delta_i &= \frac{1}{2}\mathbb{E}_{X_i}((P_f - P_{f|X_i})^2) \\ &= \frac{1}{2}\mathbb{E}_{X_i}((\mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X})) - \mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X}) \mid X_i))^2) \\ &= \frac{1}{2}\mathbb{V}(\mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X}) \mid X_i)) \end{aligned} \tag{2.59}$$

The importance measure defined here reflects the effect of the variable X_i on the failure probability. Normalizing the previous expression by the total variance $\mathbb{V}(1_{\mathcal{D}_f}(X))$ retrieves similar

expressions to Sobol first order indices

$$S_i^{1_{\mathcal{D}_f}} = \frac{\mathbb{V}(\mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X}) \mid X_i))}{\mathbb{V}(1_{\mathcal{D}_f}(\mathbf{X}))} \tag{2.60}$$

and Sobol total indices

$$S_{T_i}^{1_{\mathcal{D}_f}} = 1 - \frac{\mathbb{V}(\mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X}) \mid \mathbf{X}_{\sim i}))}{\mathbb{V}(1_{\mathcal{D}_f}(\mathbf{X}))} \tag{2.61}$$

or using Equation (2.14)

$$S_{T_i}^{1_{\mathcal{D}_f}} = \frac{\mathbb{E}(\mathbb{V}(1_{\mathcal{D}_f}(\mathbf{X}) \mid \mathbf{X}_{\sim i}))}{\mathbb{V}(1_{\mathcal{D}_f}(\mathbf{X}))} \tag{2.62}$$

where $\mathbf{X}_{\sim i}$ means $\mathbf{X}$ without X_i. Usual strategies to estimate Sobol indices can be used to estimate both $S_i^{1_{\mathcal{D}_f}}$ and $S_{T_i}^{1_{\mathcal{D}_f}}$. Interestingly, Equation (2.60) can be derived, using the Bayes' theorem as the following [PD19]

$$\begin{aligned} S_i^{1_{\mathcal{D}_f}} &= \frac{\mathbb{V}(\mathbb{E}(1_{\mathcal{D}_f}(\mathbf{X}) \mid X_i))}{\mathbb{V}(1_{\mathcal{D}_f}(\mathbf{X}))} = \frac{\mathbb{V}(P_{f|X_i})}{P_f(1-P_f)} \\ &= \frac{1}{P_f(1-P_f)} \mathbb{V}\left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i) \times P_f)}{p_{X_i}(x_i)}\right) = \frac{P_f}{1-P_f} \mathbb{V}\left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)}\right) \end{aligned} \tag{2.63}$$

The same reasoning can be applied to $S_{T_i}^{1_{\mathcal{D}_f}}$. This means that the first order Sobol indices associated with the indicator function are in fact proportional to the variance of the ratios between the probability density function of X_i and their probability density function given that the output exceeds a given threshold q. [PD19] proposes to approximate the probability density functions using non-parametric approaches to estimate the ratio.

Furthermore, one can also notice that

$$\begin{aligned} \frac{P_f}{1-P_f} \mathbb{V}\left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)}\right) &= \frac{P_f}{1-P_f} \mathbb{E}_{X_i}\left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)} - \mathbb{E}_{X_i}\left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)}\right)^2\right)^2 \\ &= \frac{P_f}{1-P_f} \int \left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)} - 1\right)^2 p_{X_i}(x_i) dx_i \end{aligned} \tag{2.64}$$

since $\mathbb{E}_{X_i}\left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)}\right) = 1$. Considering the dissimilarity measures defined previously in Equation (2.21), the Sobol first order index associated with the indicator function is directly connected to a specific dissimilarity measure using the Pearson χ^2 divergence:

$$\begin{aligned} S_i^{1_{\mathcal{D}_f}} &= \frac{P_f}{1-P_f} \int \left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)} - 1\right)^2 p_{X_i}(x_i) dx_i \\ &= \frac{P_f}{1-P_f} \int \chi^2\left(\frac{p_{X_i|\mathbf{X}\in\mathcal{D}_f}(x_i)}{p_{X_i}(x_i)}\right) p_{X_i}(x_i) d(x_i) = \frac{P_f}{1-P_f} d_{\chi^2}(P_{X_i}, P_{X_i|\mathbf{X}\in\mathcal{D}_f}) \end{aligned} \tag{2.65}$$

The main issue with the previous formulation comes from the estimation of the χ^2 divergence, usually computed nonparametric approach such as kernel density estimation (KDE).

Interestingly, it is possible to exhibit a relation between the formulation of $S_i^{1_{\mathcal{D}_f}}$ obtained in Equation (2.65) and a dependence measure called the Squared-loss Mutual Information (SMI)

[Suz+09] with the following proposition:

Proposition 1. The Pearson χ^2 divergence between P_{X_i} and $P_{X_i|\mathbf{X}\in\mathcal{D}_f}$ is proportional to the Squared-loss Mutual Information between the input X_i and the indicator function $1_{\mathcal{D}_f}$ (i.e. inputs in the failure domain):

$$\frac{P_f}{1-P_f} d_{\chi^2}(P_{X_i}, P_{X_i|\mathbf{X}\in\mathcal{D}_f}) = \mathrm{SMI}(X_i, 1_{\mathcal{D}_f}) \tag{2.66}$$

Proof. Let assume $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. Let $p_{XY}(x,y)$ be the joint density probability of X and Y, and $p_X(x)$ and $p_Y(y)$ be the marginal densities of X, and Y respectively. The Squared-loss Mutual Information for X and Y is:

$$\mathrm{SMI}(X,Y)) = \frac{1}{2}\iint \left(\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} - 1\right)^2 p_X(x)p_Y(y)dxdy \tag{2.67}$$

Defining $Z = 1_{\mathcal{D}_f}$, since Z is a discrete random variable, the SMI between X_i and Z goes as

$$\begin{aligned}\mathrm{SMI}(X_i,Z)) =& \sum_{z=0}^{1}\int \left(\frac{p_{X_iZ}(x_i,z)}{p_{X_i}(x_i)P(Z=z)} - 1\right)^2 p_{X_i}(x_i)P(Z=z)dx_i \\ =& \int \left(\frac{p_{X_iZ}(x_i,0)}{p_{X_i}(x_i)P(Z=0)} - 1\right)^2 p_{X_i}(x_i)P(Z=0)dx_i \\ &+ \int \left(\frac{p_{X_iZ}(x_i,1)}{p_{X_i}(x_i)P(Z=1)} - 1\right)^2 p_{X_i}(x_i)P(Z=1)dx_i\end{aligned} \tag{2.68}$$

Deriving the first term in Equation (2.68) leads to

$$\begin{aligned}\left(\frac{p_{X_iZ}(x_i,0)}{p_{X_i}(x_i)P(Z=0)} - 1\right)^2 &= \left(\frac{p_{X_i}(x_i) - p_{X_iZ}(x_i,1)}{p_{X_i}(x_i)(1-P(Z=1))} - 1\right)^2 \\ &= \left(\frac{P(Z=1)}{1-P(Z=1)} - \frac{p_{X_iZ}(x_i,1)}{p_{X_i}(x_i)(1-P(Z=1))}\right)^2 \\ &= \frac{P(Z=1)^2}{(1-P(Z=1))^2}\left(1 - \frac{p_{X_iZ}(x_i,1)}{p_{X_i}(x_i)P(Z=1)}\right)^2\end{aligned} \tag{2.69}$$

Replacing it in Equation (2.68) yields

$$\begin{aligned}\mathrm{SMI}(X_i,Z)) =& \frac{P(Z=1)^2}{(1-P(Z=1))^2}\int \left(1 - \frac{p_{X_iZ}(x_i,1)}{p_{X_i}(x_i)P(Z=1)}\right)^2 p_{X_i}(x_i)(1-P(Z=1))dx_i \\ &+ \int \left(\frac{p_{X_iZ}(x_i,1)}{p_{X_i}(x_i)P(Z=1)} - 1\right)^2 p_{X_i}(x_i)P(Z=1)dx_i \\ =& \frac{P(Z=1)}{1-P(Z=1)}\int \left(\frac{p_{X_iZ}(x_i,1)}{p_{X_i}(x_i)P(Z=1)} - 1\right)^2 p_{X_i}(x_i)dx_i\end{aligned} \tag{2.70}$$

Finally, since $p_{X_i|Z=1}(x_i) = \frac{p_{X_iZ}(x_i,1)}{P(Z=1)}$, we obtain

$$\mathrm{SMI}(X,Z)) = \frac{P(Z=1)}{1-P(Z=1)} d_{\chi^2}(P_{X_i}, P_{X_i|Z=1}) = S_i^{1_{\mathcal{D}_f}} \tag{2.71}$$

which ends the proof. □

The SMI is a measure of independence between two variables (in our case a given input X_i and our quantity of interest $1_{\mathcal{D}_f}$) that has a direct connection to the MI with the notable advantage of being approximated from data much more efficiently and of being numerically more stable than the MI. Thus, the Sobol first order index associated with the indicator function can actually be viewed as an independence measure between the input and the quantity of interest considered here, the indicator on the failure domain.

The basic idea of the SMI is to directly estimate the density-ratio

$$r_{XY}(x, y) = \frac{p_{XY}(x, y)}{p_X(x)p_Y(y)} \tag{2.72}$$

without going through the density estimation of $p_{XY}(x, y)$, $p_X(x)$ or $p_Y(y)$. More details on the estimation of the density-ratio with possible applications are available in [Sug13].

2.3.3 Regional Sensitivity analysis

Regional Sensitivity Analysis (RSA), first introduced and investigated in [YHS78; SH80], also called Monte Carlo filtering in [Sal+04], is a family of methods aimed at identifying regions in the inputs space corresponding to particular values of the output. The input samples are partitioned into a behavioral $\mathcal{B}$ and a non behavioral $\overline{\mathcal{B}}$ group. The behavioral group corresponds to inputs sets producing a preferred model response (a behavior). The division into behavioral and non behavioral usually depends on whether the associated model simulation exhibits the expected pattern of the output or not. This distinction can also be based on a measure of performance, depending on whether the associated output is above or below a prescribed threshold. Regional Sensitivity Analysis identifies if a variable is important with respect to one group or another by comparing the probability density functions $p_m(X_i \mid Y \in \mathcal{B})$ and $p_n(X_i \mid Y \in \overline{\mathcal{B}})$. This comparison is either done through a visual inspection of the empirical cumulative distribution functions of the two sets that provides an indication on the behavioral impact due to a given input, or using standard statistical tests, such as the Kolmogorov-Smirnov statistic, to measure the divergence between the two cumulative distribution distributions. [YHS78] derives a sensitivity index based on the aforementioned statistic test as

$$S_i^{\text{RSA}} = \max_{X_i} |F_m(X_i \mid Y \in \mathcal{B}) - F_n(X_i \mid Y \in \overline{\mathcal{B}})| \tag{2.73}$$

with F the empirical cumulative distribution, where the subscript corresponds to the number of input samples lying either in the behavioral set $\mathcal{B}$ or in its complementary behavioral set $\overline{\mathcal{B}}$. The test in Equation (2.73) is performed for each input independently. The obtained value determines at which significance level α one can reject the null hypothesis $H_0 : p_m(X_i \mid Y \in \mathcal{B}) = p_n(X_i \mid Y \in \overline{\mathcal{B}})$. The smaller α, or the bigger S_i^{RSA}, the more important the input is in driving the considered behavior of the output. To perform the test, one must choose the significance level, corresponding the probability of rejecting the null hypothesis when it is true (hence flagging an input as important while it is not the case, also known as the type-I error). Then, using the critical level D_α, one can determines whether H_0 is rejected or not. This procedure is simplified in Figure 2.8 on an example displaying a significant difference between both behavioral and non behavioral subsets.

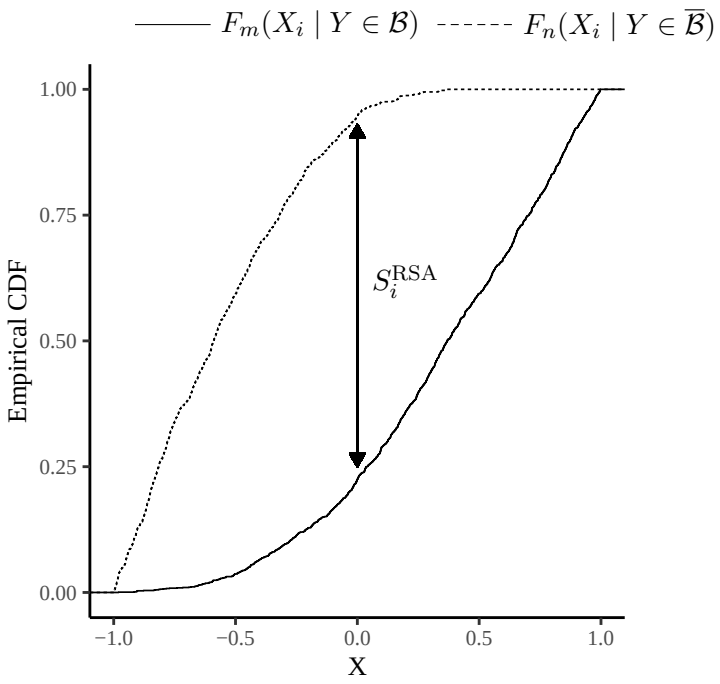

Figure 2.8 – Graphical representation of the Kolmogorov-Smirnov test computed in Equation (2.73). In this case, the cumulative distribution for $X \in \mathcal{B}$ is steepest on the left side, meaning low values of X are more likely to produce behavioral output realizations.

Despite some interesting properties, Regional sensitivity analysis has many drawbacks, highlighted in [SGS94]. The statistic used in Equation (2.73) is not sufficient for screening purposes as a value of 0 does not necessarily implies insensitivity for the considered input since the method only characterized inputs contributions related to main effects of variance based methods. Furthermore, the threshold at which the output is deemed acceptable is often a subjective choice by the designer. To overcome this previous issue, one can rank the output samples and group them in a given number of equally spaced intervals then compare the resulting cumulative distribution functions of the input variables, see [FBA96].

In a sense, the principles of Regional sensitivity analysis and Sobol indices on the indicator function are equivalent. They both characterize influence of inputs through differences between the specific distributions. Only the distance considered differs, as the Regional sensitivity analysis relies on the Kolmogorov-Smirnov distance and Sobol on the indicator function relies on the Pearson χ^2 divergence.

2.4 Conclusions

This chapter was dedicated to define what is sensitivity analysis and how it is used to measure the impact of an input or a set of inputs on the variability of the model output. Different methods were presented: the generic variance-based methods, and other distribution-based methods which try to have a broader point of view. We focus on kernel-based methods, which are at the core of our strategies. To circumvent the limitations of direct distribution-based approaches in terms of dimension, they embed the distributions in reproducing kernel Hilbert spaces. As the most used indices rely on generic quantities of interest of the output (e.g., the output variance), the need for different strategies arises when conducting specific studies such

as reliability analysis and optimization. In the light of the above, we present in the next chapter a kernel-based sensitivity strategy dedicated to high-dimensional optimization problem.

Chapter take-home messages

- Sensitivity analysis are methods which characterize the influence of the variation of an input on the variability of the output.
- Variance-based methods are the most well-known methods and focus on the second statistical moment of the output.
- Kernel-based methods are a recent approach based on distribution embeddings in Hilbert spaces, which consider the complete distribution instead of the moments.
- Depending on the goal of the study, other indices can be defined, which reflect specific quantities of interest.
- Goal-oriented sensitivity analyses mostly rely on the distance between specific distributions. A new version adapted to optimization could be proposed using kernel-based methods.

Bibliography

[Aro50] Nachman Aronszajn. "Theory of reproducing kernels". In: *Transactions of the American mathematical society* 68.3 (1950), pp. 337–404.

[Bor+06] Karsten M Borgwardt et al. "Integrating structured biological data by kernel maximum mean discrepancy". In: *Bioinformatics* 22.14 (2006), e49–e57.

[Bor07] Emanuele Borgonovo. "A new uncertainty importance measure". In: *Reliability Engineering & System Safety* 92.6 (2007), pp. 771–784.

[BP16] Emanuele Borgonovo and Elmar Plischke. "Sensitivity analysis: a review of recent advances". In: *European Journal of Operational Research* 248.3 (2016), pp. 869–887.

[Bro+17] Thomas Browne et al. "Estimate of quantile-oriented sensitivity indices". 2017. URL: https://hal.archives-ouvertes.fr/hal-01450891.

[BTA11] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[CCS07] Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. "An effective screening design for sensitivity analysis of large models". In: *Environmental modelling & software* 22.10 (2007), pp. 1509–1518.

[CLS77] Robert I Cukier, Howard B Levine, and Kurt E Shuler. "Nonlinear sensitivity analysis of multiparameter model systems". In: *The Journal of Physical Chemistry* 81.25 (1977), pp. 2365–2366.

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[DLM16] Matthias De Lozzo and Amandine Marrel. "New improvements in the use of dependence measures for sensitivity analysis and screening". In: *Journal of Statistical Computation and Simulation* 86.15 (2016), pp. 3038–3058.

[DLM17] Matthias De Lozzo and Amandine Marrel. "Sensitivity analysis with dependence and variance-based measures for spatio-temporal numerical simulators". In: *Stochastic Environmental Research and Risk Assessment* 31.6 (2017), pp. 1437–1453. ISSN: 1436-3259. DOI: 10.1007/s00477-016-1245-3. URL: https://doi.org/10.1007/s00477-016-1245-3.

[DV15] Sébastien Da Veiga. "Global sensitivity analysis with dependence measures". In: *Journal of Statistical Computation and Simulation* 85.7 (2015), pp. 1283–1305.

[ES81] Bradley Efron and Charles Stein. "The jackknife estimate of variance". In: *The Annals of Statistics* (1981), pp. 586–596.

[ET94] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[FBA96] Jim Freer, Keith Beven, and Bruno Ambroise. "Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the GLUE approach". In: *Water Resources Research* 32.7 (1996), pp. 2161–2173.

[FBJ04] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces". In: *Journal of Machine Learning Research* 5.Jan (2004), pp. 73–99.

[FKR16] Jean-Claude Fort, Thierry Klein, and Nabil Rachdi. "New sensitivity analysis subordinated to a contrast". In: *Communications in Statistics-Theory and Methods* 45.15 (2016), pp. 4349–4364.

[Fuk+08] Kenji Fukumizu et al. "Kernel measures of conditional dependence". In: *Advances in neural information processing systems*. 2008, pp. 489–496.

[Fuk+09] Kenji Fukumizu et al. "Kernel choice and classifiability for RKHS embeddings of probability distributions". In: *Advances in neural information processing systems*. 2009, pp. 1750–1758.

[Gre+05a] Arthur Gretton et al. "Kernel methods for measuring independence". In: *Journal of Machine Learning Research* 6.Dec (2005), pp. 2075–2129.

[Gre+05b] Arthur Gretton et al. "Measuring statistical dependence with Hilbert-Schmidt norms". In: *International conference on algorithmic learning theory*. Springer. 2005, pp. 63–77.

[Gre+08] Arthur Gretton et al. "A kernel statistical test of independence". In: *Advances in neural information processing systems*. 2008, pp. 585–592.

[Gre+09] Arthur Gretton et al. "A fast, consistent kernel two-sample test". In: *Advances in neural information processing systems*. 2009, pp. 673–681.

[Gre+12] Arthur Gretton et al. "A kernel two-sample test". In: *Journal of Machine Learning Research* 13.Mar (2012), pp. 723–773.

[Hoe48] Wassily Hoeffding. "A class of statistics with asymptotically normal distribution". In: *Annals of Mathematical Statistics* 19.293-325 (1948).

[Hot33] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417.

[HS96] Toshimitsu Homma and Andrea Saltelli. "Importance measures in global sensitivity analysis of nonlinear models". In: *Reliability Engineering & System Safety* 52.1 (1996), pp. 1–17.

[IL15] Bertrand Iooss and Paul Lemaître. "A review on global sensitivity analysis methods". In: *Uncertainty management in simulation-optimization of complex systems*. Springer, 2015, pp. 101–122.

[KS09] Sergei Kucherenko and Ilya M Sobol. "Derivative based global sensitivity measures and their link with global sensitivity indices". In: *Mathematics and Computers in Simulation* 79.10 (2009), pp. 3009–3017.

[LCS06] Huibin Liu, Wei Chen, and Agus Sudjianto. "Relative entropy based method for probabilistic sensitivity analysis in engineering design". In: *Journal of Mechanical Design* 128.2 (2006), pp. 326–336.

[LGMS16] Loic Le Gratiet, Stefano Marelli, and Bruno Sudret. "Metamodel-based sensitivity analysis: polynomial chaos expansions and Gaussian processes". In: *Handbook of Uncertainty Quantification - Part III: Sensitivity analysis*. 2016. URL: https://hal.archives-ouvertes.fr/hal-01428947.

[Luy+12] Li Luyi et al. “Moment-independent importance measure of basic variable and its state dependent parameter solution”. In: *Structural Safety* 38 (2012), pp. 40–47.

[Mar+09] Amandine Marrel et al. “Calculations of sobol indices for the gaussian process metamodel”. In: *Reliability Engineering & System Safety* 94.3 (2009), pp. 742–751.

[MDN18] Véronique Maume-Deschamps and Ibrahima Niang. “Estimation of quantile oriented sensitivity indices”. In: *Statistics & Probability Letters* 134 (2018), pp. 122–127.

[MML19] Anouar Meynaoui, Amandine Marrel, and Béatrice Laurent. “New statistical methodology for second level global sensitivity analysis”. In: *arXiv preprint arXiv:1902.07030* (2019).

[Mor91] Max D Morris. “Factorial sampling plans for preliminary computational experiments”. In: *Technometrics* 33.2 (1991), pp. 161–174.

[Mül97] Alfred Müller. “Integral probability metrics and their generating classes of functions”. In: *Advances in Applied Probability* 29.2 (1997), pp. 429–443.

[PD19] Guillaume Perrin and Gilles Defaux. “Efficient Evaluation of Reliability-Oriented Sensitivity Indices”. In: *Journal of Scientific Computing* 79.3 (2019), pp. 1433–1455.

[Rah16] Sharif Rahman. “The f-sensitivity index”. In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (2016), pp. 130–162.

[Sal02] Andrea Saltelli. “Making best use of model evaluations to compute sensitivity indices”. In: *Computer physics communications* 145.2 (2002), pp. 280–297.

[Sal+04] Andrea Saltelli et al. “Sensitivity analysis in practice: a guide to assessing scientific models”. In: *Chichester, England* (2004).

[Sal+08] Andrea Saltelli et al. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[Ser81] Robert J Serfling. “Approximation theorems of mathematical statistics”. In: (1981).

[SFL11] Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. “Universality, characteristic kernels and RKHS embedding of measures”. In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2389–2410.

[SGS94] Robert C Spear, Thomas M Grieb, and Nong Shang. “Parameter uncertainty and interaction in complex environmental models”. In: *Water Resources Research* 30.11 (1994), pp. 3159–3169.

[SH80] Robert C Spear and George M Hornberger. “Eutrophication in peel inlet-II. Identification of critical uncertainties via generalized sensitivity analysis”. In: *Water Research* 14.1 (1980), pp. 43–49.

[Sha48] Claude Elwood Shannon. “A mathematical theory of communication”. In: *Bell system technical journal* 27.3 (1948), pp. 379–423.

[Smo+07] Alex Smola et al. “A Hilbert space embedding for distributions”. In: *International Conference on Algorithmic Learning Theory*. Springer. 2007, pp. 13–31.

[Sob01] Ilya M Sobol. “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”. In: *Mathematics and computers in simulation* 55.1-3 (2001), pp. 271–280.

[Sob93] Ilya M Sobol. “Sensitivity estimates for nonlinear mathematical models”. In: *Mathematical modelling and computational experiments* 1.4 (1993), pp. 407–414.

[Sri+08] Bharath K Sriperumbudur et al. “Injective Hilbert space embeddings of probability measures”. In: *21st Annual Conference on Learning Theory (COLT 2008)*. Omnipress. 2008, pp. 111–122.

[Sri+12] Bharath K Sriperumbudur et al. “On the empirical estimation of integral probability metrics”. In: *Electronic Journal of Statistics* 6 (2012), pp. 1550–1599.

[SSB+02] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[Sud08] Bruno Sudret. “Global sensitivity analysis using polynomial chaos expansions”. In: *Reliability engineering & system safety* 93.7 (2008), pp. 964–979.

[Sug13] Masashi Sugiyama. “Machine learning with squared-loss mutual information”. In: *Entropy* 15.1 (2013), pp. 80–112.

[Suz+09] Taiji Suzuki et al. “Mutual information estimation reveals global associations between stimuli and biological processes”. In: *BMC bioinformatics* 10.1 (2009), S52.

[YHS78] Peter C Young, George M Hornberger, and Robert C Spear. “Modeling badly defined systems: some further thoughts”. In: *Proceedings SIMSIG Conference*. Australian National University Canberra. 1978, pp. 24–32.

Chapter 3

Offline high-dimensional optimization

This chapter is adapted from the following published article:

Spagnol, Adrien, Le Riche, Rodolphe , & Da Veiga, Sébastien (2019). Global sensitivity analysis for optimization with variable selection. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2), 417-443.

Contents

3.1 Transformation of the output

In the goal-oriented sensitivity analysis framework defined earlier, the first objective is to define a quantity of interest related to the goal of our study or being able to characterize a set of interest producing *behavioral* realizations of the output.

In the scope of our work, the main goal of the study is to optimize the objective function f, sometimes under m constraints functions, hence solving:

$$\begin{aligned} \min_{\mathbf{X}\in\mathcal{X}\subset\mathbb{R}^d} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & g_l(\mathbf{X}) \leq 0, l = 1, \dots, m. \end{aligned} \tag{3.1}$$

Traditionally, the Sobol indices of the inputs are computed and the inputs with total index close to zero are set to a fixed value, often the nominal value (when defined). This was characterized as factor fixing in Section 2.1. For optimization problems, having fewer variables considered in the problem implies a smaller search volume, thus fewer calls to the model to reach an optimum. However, fixing low-impacting inputs results in a loss of fine-tuning ability due to the simplification of the problem. Furthermore, the optimum of the modified problem might differ from the real global optimum. To illustrate this, we use the two-dimensional Dixon-Price function as an example:

$$f(\mathbf{X}) = (X_1 - 1)^2 + 2(X_2^2 - X_1)^2 \tag{3.2}$$

with $X_i \sim \mathcal{U}[-10, 10]$, for $i = \{1, 2\}$. This functions has a characteristic U-shape, cf Figure 3.1. A sensitivity analysis of the output function using Sobol indices gives a total index close to 0 for X_1 while it is close to 1 for X_2. In the light of this analysis, the first input appears as negligible and can be set to a fixed value, such as its mean value $\mu_{X_1} = 0$. But doing so makes it impossible to find the true global optimum $\mathbf{X}^* = [1, \sqrt{2}/2]$. As shown in Figure 3.1, low values of the Dixon-Price function have skewed contour lines, showing that what matters to find the global minimum is the interaction of both variables and not the sole action of one.

In the sense of the Regional sensitivity analysis, we define our set of interest as the sublevel set where the objective is below a given threshold q and the constraints $\mathbf{g}$ are respected up to a relaxation threshold $\mathbf{T}$

$$\mathcal{D}_{q,\mathbf{T}} = \{\mathbf{X} \in \mathbb{R}^d, f(\mathbf{X}) \leq q \cap \mathbf{g}(\mathbf{X}) \leq \mathbf{T}\} \tag{3.3}$$

with $\mathbf{T} \in \mathbb{R}^{m,+}$ and $q \in \mathbb{R}$. The threshold $\mathbf{T}$ relaxes the constraints when finding a feasible point is too difficult. Hereafter, all $\mathbf{T}$ values are similar and chosen in order to have a sufficient number of feasible points for the sensitivity indices computation, i.e. a few hundreds. The threshold q that contributes to the definition of $\mathcal{D}_{q,\mathbf{T}}$ is a quantile q_α of the objective function $f(\cdot)$. Low quantiles ensure that we are looking at values of the output close to the best observations.

We derive sensitivity indices adapted to optimization by three thresholding transformations of the output $f(\mathbf{X})$ and by performing sensitivity analysis on the modified output which is written Z. Each modification of the output can be seen as a new quantity of interest usable in a sensitivity analysis context, with some resembling the one considered for the reliability sensitivity analysis from the previous chapter, Section 2.3.2. We consider the following output transformations based on thresholding:

1. Zero-thresholding: $Z = f(\mathbf{X}) \times 1_{\mathcal{D}_{q,\mathbf{T}}}$,

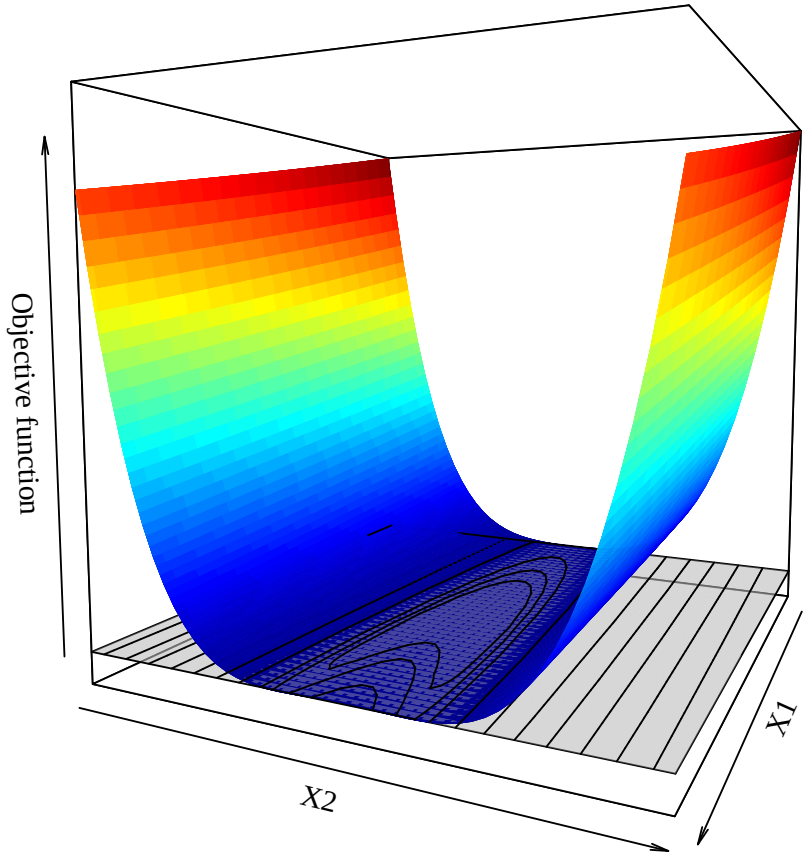

Figure 3.1 – Surface representation of the two-dimensional Dixon-Price function Equation (3.2), with its characteristic U-shape. Contour lines for low values of the objective output are also shown in black.

2. Conditional-thresholding: $Z = f(\mathbf{X})|(\mathbf{X} \in \mathcal{D}_{q,\mathbf{T}})$,

3. Indicator-thresholding: $Z = 1_{\mathcal{D}_{q,\mathbf{T}}}$,

where 1 is the indicator function, $1_{\mathcal{D}_{q,\mathbf{T}}} = 1$ if $\mathbf{X} \in \mathcal{D}_{q,\mathbf{T}}$ and 0 otherwise. We discuss in the following these different thresholdings.

3.1.1 Zero-thresholding

Recalling the previous example of the two-dimensional Dixon-Price function, we define[1] $Z = f(X) \times 1_{\mathcal{D}_{q,\infty}}$ and compute the first and total order Sobol indices of Z with respect to the value of α, see Figure 3.2 (A). In 2 dimensions, $S_1^T = S_1 + S_{12}$, resp. $S_2^T = S_2 + S_{12}$, where S_{12} is the second-order index which characterizes the effect of X_1 and X_2 varying simultaneously. In that case, when α decreases, S_2 decreases while S_2^T remains constant, meaning that S_{12} increases. Hence, for low values of f, the interaction of both inputs matters for our optimization problem and not exclusively X_2 as found before when considering the whole domain of X. The right side of Figure 3.3 shows the evolution of the contour of the Dixon-Price function which gives an insight of the results obtained previously: while most of the variance in the left plot is due to X_2 and the contour lines correspond to those of a function without interaction, the contour lines in the right plot are distorted with a stronger role of X_1 and its interaction with X_2.

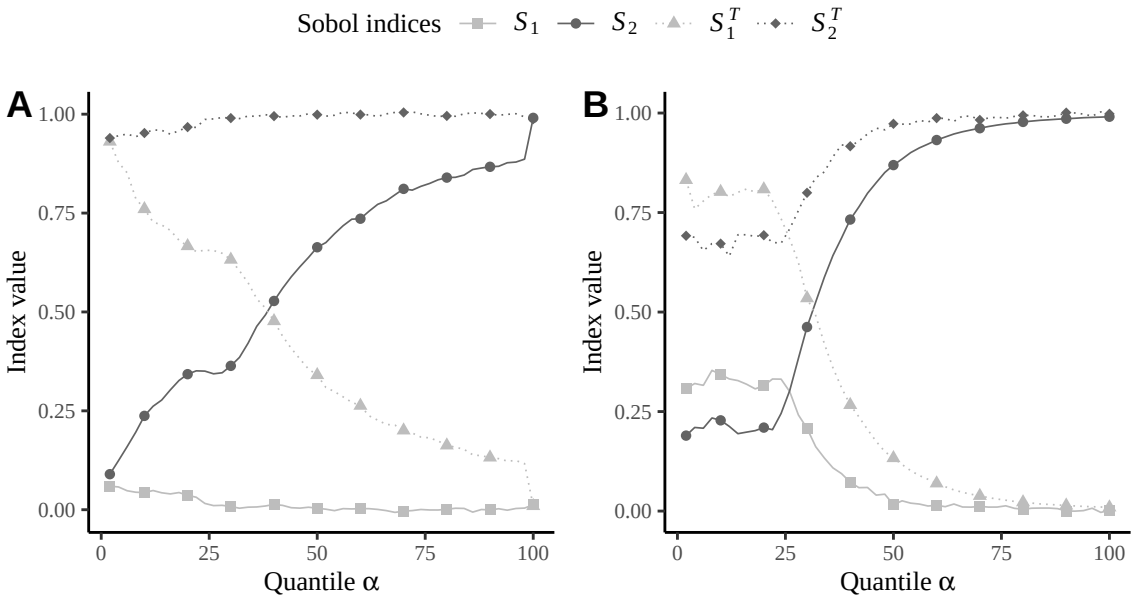

Figure 3.2 – Evolution of S_1 and S_1^T, resp. S_2 and S_2^T, with respect to the quantile α for the the Dixon-Price function Equation (3.2) using **(A)** zero- and **(B)** conditional-thresholdings.

Although in the last example the sensitivity of the variable is qualitatively well captured, the zero-thresholding is hard to interpret for two reasons. First of all, the values of Z outside of $\mathcal{D}_{q,\mathbf{T}}$ are arbitrarily fixed at zero but other value are possible[1] and this will affect the calculated sensitivities. Second, the sensitivity of Z using this thresholding characterizes both the variation of f inside $\mathcal{D}_{q,\mathbf{T}}$ and the shape of $\mathcal{D}_{q,\mathbf{T}}$. To illustrate this point let us consider the simple example

[1] As a special case for the more general C constant-thresholding, $Z = f(X) \times 1_{\mathcal{D}_{q,\infty}} + C \times 1_{\bar{\mathcal{D}}_{q,\infty}}$, with $\bar{\mathcal{D}}_{q,\infty}$ the complementary set of $\mathcal{D}_{q,\infty}$. The ∞ subscript in place of $\mathbf{T}$ expresses the lack of constraint functions, such as in the presented examples of this section.

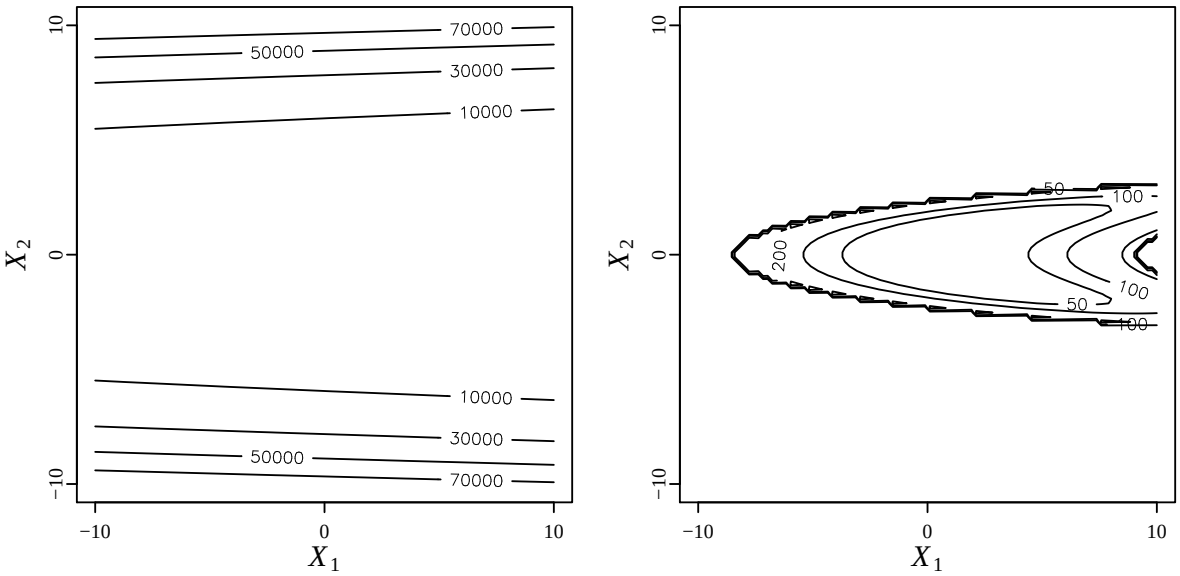

Figure 3.3 – Left: contour of the Dixon-Price function Equation (3.2), Right: contour of the thresholded Dixon-Price function for $q = q_{20\%}\}$. The contour lines on the right-hand side no longer correspond to an ellipse aligned with the reference axes, there is a change of curvature associated to a Sobol dependency between the variables.

of a linear function

$$f(\mathbf{X}) = X_1 + 2X_2, \tag{3.4}$$

defined on $[-10, 10]^2$, see Figure 3.5 for its contours. The sensitivity indices can be analytically determined on the whole domain: $S_1 = 1/5$ and $S_2 = 4/5$. Since the function is already in its decomposed form, it is obvious that there is no interaction between variables for the complete domain (i.e., $\alpha = 100\%$). Yet, interactions appear when α gets lower than 100%, as it can be seen in Figure 3.4 (A), because $\mathcal{D}_{q,\mathbf{T}}$ takes a non-rectangular shape.

3.1.2 Conditional-thresholding

Unlike the previous zero-thresholding, conditional-thresholding aims at knowing which inputs are important inside $\mathcal{D}_{q,\mathbf{T}}$. Yet, a dependency on the shape of $\mathcal{D}_{q,\mathbf{T}}$ remains, as it can be seen with the linear function of Figure 3.4 where we observe two phenomena. First, for all α below 25%, the indices reach a steady-state. Below this value of α, the shape of $\mathcal{D}_{q,\infty}$ remains a right-angled triangle of unit height and base length of two, affecting all variables in the same way, see Figure 3.5. Besides, both first order indices are equal from this point on: while X_2 is twice more sensitive than X_1 in the function definition, its interval in the sub-level $\mathcal{D}_{q,\infty}$ is twice as narrow, which makes up for the difference in terms of Sobol indices. Prior to $\alpha = 25\%$, the shape of $\mathcal{D}_{q,\infty}$ depends on α. Note that this version of thresholding does not have any arbitrary threshold value, unlike the zero-thresholding where values outside $\mathcal{D}_{q,\mathbf{T}}$ were set to zero.

3.1.3 Indicator-thresholding

This last thresholding transformation captures which variables are important in order to reach $\mathcal{D}_{q,\mathbf{T}}$ while not depending on the specific values of the objective function f inside it. This can be done through a discrete encoding using the indicator function $1_{\mathcal{D}_{q,\mathbf{T}}} : \mathcal{Y} \to [0, 1], Y = 1$ if

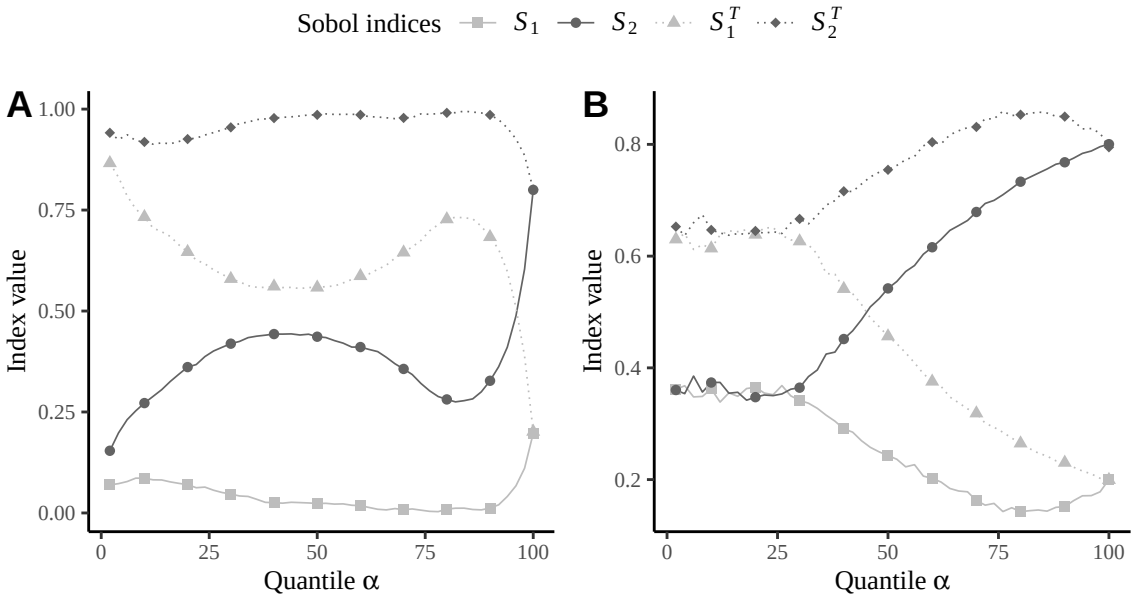

Figure 3.4 – Evolution of S_1 and S_1^T, resp. S_2 and S_2^T, with respect to the quantile α for the linear function using **(A)** zero- and **(B)** conditional-thresholdings.

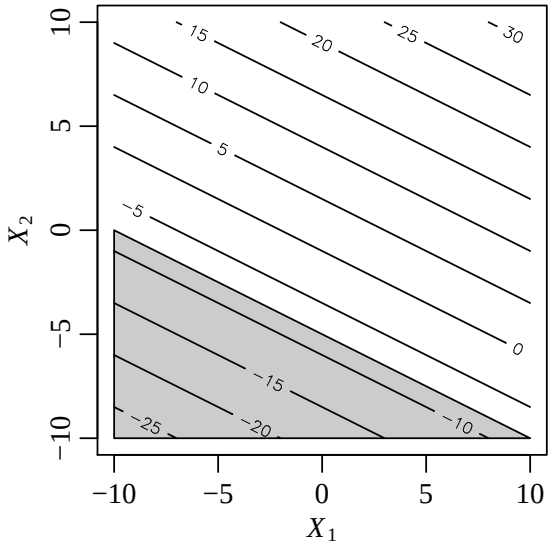

Figure 3.5 – Contour plot of the linear function. The gray area corresponds to $\mathcal{D}_{q_{25\%},\mathbf{T}}$.

$Y \in \mathcal{D}_{q,\mathbf{T}}$ and 0 otherwise. Since it transforms the output into a categorical variable, there is no need to assign a specific value to points outside $\mathcal{D}_{q,\mathbf{T}}$. Unlike previous thresholdings, the indicator-thresholding only characterizes the boundary of $\mathcal{D}_{q,\mathbf{T}}$ and keeps no information about the values inside or outside the set of interest. It is independent of any monotonous scaling of $f()$, which is a desirable invariance property in optimization [Oll+17]. From now on, we will only focus on the indicator thresholding because of its aforementioned assets.

An important aspect related to the last transformation must be considered. Since the number of observations is usually limited, when the sublevel set $\mathcal{D}_{q,\mathbf{T}}$ is rather small the binary transformation can result in a majority of zero outputs and the information conveyed by the

relative values of the output is lost. To overcome this, a possibility would be to use a smooth transformation instead of a binary one, as this would act as some kind of relaxation on the indicator function.

In the following chapter, we introduce a kernel-based sensitivity index based on the Indicator thresholding as it allows to highlight a connection between a kernel-based dependence measure and a specific distance between distribution embeddings.

3.1.4 Kernel dependence measure on categorical inputs

In Section 2.2 all the necessary theory about RKHS and the associated dependence measure, the HSIC, were detailed. In this subsection, we define a sensitivity measure using the HSIC and the indicator-thresholding modification aforementioned. The HSIC dependence measure only relies on the choice of kernel functions associated to the inputs and the outputs. This choice depends directly on the type of data: for example, for continuous data sets, it is customary to use the squared exponential kernel,

$$k(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{\sigma^2}\right) \tag{3.5}$$

with σ the bandwidth parameter. In the case of a binary transformation of the output, considering a categorical kernel for $Z = 1_{\mathcal{D}_{q,\mathbf{T}}}$ seems natural and was already suggested in [DV15]. Multiple kernels are available in that case, we can list a couple below:

- the Dirac kernel $k(x, x') = 1_{x=x'}$,
- the Linear kernel $k(x, x') = \langle x, x' \rangle$.

In order to evaluate the importance of each variable X_i separately, we define the following sensitivity measure based on the HSIC and the Indicator-Thresholding

Definition 3.1.1 (Sensitivity index from HSIC with Indicator-Thresholding, HSIC-IT)**.** Let $f() : \mathbb{R}^d \to \mathbb{R}$ and $\mathbf{g}() : \mathbb{R}^d \to \mathbb{R}^m$ be objective and constraints functions of the random variables $\mathbf{X} = (X_1, \ldots, X_d)$ and define $\mathcal{D}_{q,\mathbf{T}} = \{\mathbf{X} \in \mathbb{R}^d, f(\mathbf{X}) \leq q \cap \mathbf{g}(\mathbf{X}) \leq \mathbf{T}\}$ for any $q \in \mathbb{R}$ and $\mathbf{T} \in \mathbb{R}^{m,+}$. The sensitivity index of the variable X_i from the Hilbert-Schmidt Independence Criterion with the Indicator-Thresholding (HSIC-IT) is [2]

$$S^{\text{HSIC}}_{q,\mathbf{T}}(X_i) = \text{HSIC}(X_i, 1_{\mathcal{D}_{q,\mathbf{T}}}). \tag{3.6}$$

A variable X_i is negligible for optimization if its index $S^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$ is close to zero. The main difference between the above $S^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$ definition and the sensitivity index proposed in [DV15] lies in the use of the indicator function.

Assume we have a sample of observations $\mathbb{X}_i = (x_i^1, \ldots, x_i^n)$ and its corresponding output evaluations $\mathbb{Y} = f(\mathbb{X}_i)$. The binary transformation $1_{\mathcal{D}_{q,\mathbf{T}}}$, with q corresponding to an estimated low α quantile of the output $q = F_{\mathbb{Y}}^{-1}(\alpha)$, is applied to obtain the modified output evaluations $\mathbb{Z}$. Equation (3.6) can then be estimated by

$$\hat{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i) = \text{HSIC}(\mathbb{X}_i, \mathbb{Z}) = \frac{1}{n^2}\text{tr}(KHLH) \tag{3.7}$$

[2] When there is no constraint function considered, the index is either written as $S^{\text{HSIC}}_{q,\infty}(\cdot)$ or $S^{\text{HSIC}}_{q}(\cdot)$

with $K_{pq} = k(x_i^p, x_i^q)$ and $L_{pq} = l(z^p, z^q)$, the Gram matrices associated to the kernel k on the inputs and the kernel l on the binary output. H is a centering matrix defined as $H_{pq} = \delta_{pq} - n^{-1}$. A simple normalization of the indices is possible through

$$\tilde{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i) = \frac{\hat{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)}{\sum_{i=1}^{d} \hat{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)} \tag{3.8}$$

in order to guarantee that $\tilde{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i) \in [0,1]$. If $\tilde{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i) = 0$, the input is considered as negligible for the optimization. Alternatively, like in [DV15], $\hat{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$ could be divided by $\sqrt{\text{HSIC}(\mathbb{X}_i, \mathbb{X}_i)\text{HSIC}(\mathbb{Z}, \mathbb{Z})}$.

Interestingly, by rewriting Equation (3.6) using its kernel expression, it is possible to highlight a relation between the HSIC-IT and a specific MMD, considering the following proposition.

Proposition 2. Under the assumptions that the output is discrete and the proper kernel is used for it, the HSIC with Indicator-Thresholding sensitivity index for a given input X_i is directly equal to a Maximum mean discrepancy between P_{X_i} and $P_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}$:

$$\text{HSIC}(X, Z) \propto \gamma^2(P_{X_i|Z=1}, P_{X_i}) = \gamma^2(P_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}, P_{X_i}) \tag{3.9}$$

Proof. We start from the expression of the HSIC in Equation (2.45)

$$\begin{aligned}
\text{HSIC}(X_i, Z) =& \mathbb{E}_{X_i,Z}\mathbb{E}_{X_i',Z'}k(X_i, X_i')l(Z, Z') + \mathbb{E}_{X_i}\mathbb{E}_{X_i'}\mathbb{E}_Z\mathbb{E}_{Z'}k(X_i, X_i')l(Z, Z') \\
&- 2\mathbb{E}_{X_i,Z}\mathbb{E}_{X_i'}\mathbb{E}_{Z'}k(X_i, X_i')l(Z, Z') \\
=& \iint k(x_i, x_i')l(z, z')(p_{X_iZ}(x_i, z) - p_{X_i}(x_i)p_Z(z)) \\
&\times (p_{X_iZ}(x_i', z') - p_{X_i}(x_i')p_Z(z'))dx_i dx_i' dz dz'
\end{aligned}$$

Considering that the kernel used for the output is discrete, either equal to 0 or 1, we can derive that

$$\begin{aligned}
\text{HSIC}(X_i, Z) =& \sum_{z=0}^{1} \int k(x_i, x_i')l(z, z')(p_{X_i|Z=z}(x_i) \\
&- p_{X_i}(x_i))(p_{X_i|Z=z'}(x_i)(x_i') - p_{X_i}(x_i'))P(Z = z)P(Z = z')dx_i dx_i'
\end{aligned}$$

since $p_{X_iZ}(x_i, z) = p_{X_i|Z=z}(x_i)p_Z(z)$. Considering the following two properties of $l(z, z')$: 1) $l(z, z') = 0$ if $z = z'$ and 2) $l(z, z') = 0$ if $z = 0$, we write

$$\begin{aligned}
\text{HSIC}(X_i, Z) =& \sum_{z,z'=0}^{1} \int k(x_i, x_i')l(z, z')(p_{X_i|Z=z}(x_i) - p_{X_i}(x_i)) \\
&\times (p_{X_i|Z=z'}(x_i)(x_i') - p_{X_i}(x_i'))P(Z = z)P(Z = z')dx_i dx_i' \\
=& \sum_{z=0}^{1} \int k(x_i, x_i')l(z, z')(p_{X_i|Z=z}(x_i) - p_{X_i}(x_i)) \\
&\times (p_{X_i|Z=z}(x_i)(x_i') - p_{X_i}(x_i'))P(Z = z)^2 dx_i dx_i' \\
=& \int k(x_i, x_i')(p_{X_i|Z=1}(x_i) - p_{X_i}(x_i))(p_{X_i|Z=1}(x_i)(x_i') - p_{X_i}(x_i'))P(Z = 1)^2 dx_i dx_i'
\end{aligned}$$

We directly recognize the expression of the MMD between $P_{X_i|Z=1}$ and P_{X_i} times a factor $P(Z=1)^2$:

$$\gamma^2(P_{X_i|Z=1}, P_{X_i}) = \int k(x_i, x_i')(p_{X_i|Z=1}(x_i) - p_{X_i}(x_i))(p_{X_i|Z=1}(x_i)(x_i') - p_{X_i}(x_i'))dx_i dx_i'$$

□

For other kernels $l(z, z')$, this result stands and only the factor multiplying the MMD differs. For example, for the Dirac kernel, this factor is equal to $2P(Z=1)^2$. Hence $S^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$ measures the impact of an input through how much its probability distribution changes when it is restricted by the output and constraints satisfaction. The probability distributions that are considered here are uniform distributions in $\mathcal{D}_{q,\mathbf{T}}$. This choice is implicit through the selection of samples of same weight within $\mathcal{D}_{q,\mathbf{T}}$ to calculate $S^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$. This consideration can be directly related to the Regional Sensitivity Analysis approach of Section 2.3.3, in the case of $\mathcal{B}$, namely the behavioral set, is the sublevel set of interest $\mathcal{D}_{q,\mathbf{T}}$. Indeed, considering that

$$p_{X_i}(x_i) = p_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}(x_i)P(\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}) + p_{X_i|\mathbf{X}\in\overline{\mathcal{D}_{q,\mathbf{T}}}}(x_i)(1 - P(\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}})) \tag{3.10}$$

where $\overline{\mathcal{D}_{q,\mathbf{T}}}$ is the complementary of $\mathcal{D}_{q,\mathbf{T}}$, then it comes that

$$\begin{aligned} p_{X_i}(x_i) - p_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}(x_i) &= p_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}(x_i)P(\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}) + p_{X_i|\mathbf{X}\in\overline{\mathcal{D}_{q,\mathbf{T}}}}(x_i)(1 - P(\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}})) \\ &\quad - p_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}(x_i) \\ &= p_{X_i|\mathbf{X}\in\overline{\mathcal{D}_{q,\mathbf{T}}}}(x_i)(1 - P(\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}})) - p_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}(x_i)(1 - P(\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}})) \\ &= (1 - P(\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}))(p_{X_i|\mathbf{X}\in\overline{\mathcal{D}_{q,\mathbf{T}}}}(x_i) - p_{X_i|\mathbf{X}\in\mathcal{D}_{q,\mathbf{T}}}(x_i)) \end{aligned}$$

which is similar to the quantity considered in the Regional Sensitivity Analysis framework, except they work with cumulative distribution function and the distance they use is the Kolmogorov-Smirnov distance. Identically, it is also similar to the Squared-loss Mutual Information defined earlier (and its connection with the Sobol indices on the Indicator function in place of the output), but we use a different dependence measure.

As an illustration of what the sensitivity indices measures in Equation (3.6), consider the two-dimensional Dixon-Price function already discussed previously. Figure 3.6 shows observations that lead to output evaluations $f(\mathbf{X})$ that belong to the sublevel set $\mathcal{D}_q$ with q being the empirical 20% quantile of the output. The initial marginal distributions of the inputs, $X_i \sim \mathcal{U}[-10, 10]$, for $i = \{1, 2\}$, are shown in dashed lines while the marginal distributions of the inputs given $\mathcal{D}_q$ are shown in straight lines. Computing the sensitivity indices of the inputs using the HSIC-IT yields that $\hat{S}^{\text{HSIC}}_{q,\infty}(X_1) = 0.0464$ and $\hat{S}^{\text{HSIC}}_{q,\infty}(X_2) = 0.1783$, meaning that the second variable is more influential to obtain more observations in the sublevel set of interest. Considering the particular shape of the function Figure 3.1, it is particularly clear that any change for X_2 is much more meaningful to reach the bottom of the valley. This is also noticeable on the marginal distributions since the distribution of $P_{X_2|\mathbf{X}\in\mathcal{D}_q}$ differs more from the uniform $\mathcal{U}[-10, 10]$ than the distribution of $P_{X_1|\mathbf{X}\in\mathcal{D}_q}$.

Let $\{\mathbb{X}, \mathbb{Y}\}$ be a data sample of size $n = 2000$, we can also compute the HSIC-IT sensitivity indices of both variables for varying levels of quantile: each estimation is repeated 20 times. The results are shown in Figure 3.7 and several observations are notable: first of all, as expected

the sensitivity of the second variable X_2 grows as the considered quantile diminishes, second of all, the sensitivity of the first variable X_1 rises at low quantiles since it is the interaction of both variables that allows to reach such sublevel sets $\mathcal{D}_q$ and not the sole action of X_2. Finally, the estimated error of the indices increases for lower quantiles, especially on the 10% one. This comes from the fact that the number of points that lays in $\mathcal{D}_q$ equivalently decreases: considering a quantile of 10% implies that we only a tenth of the points usable for the computation of the HSIC-IT indices, the rest corresponding to a zero value after the binary transformation using the indicator function.

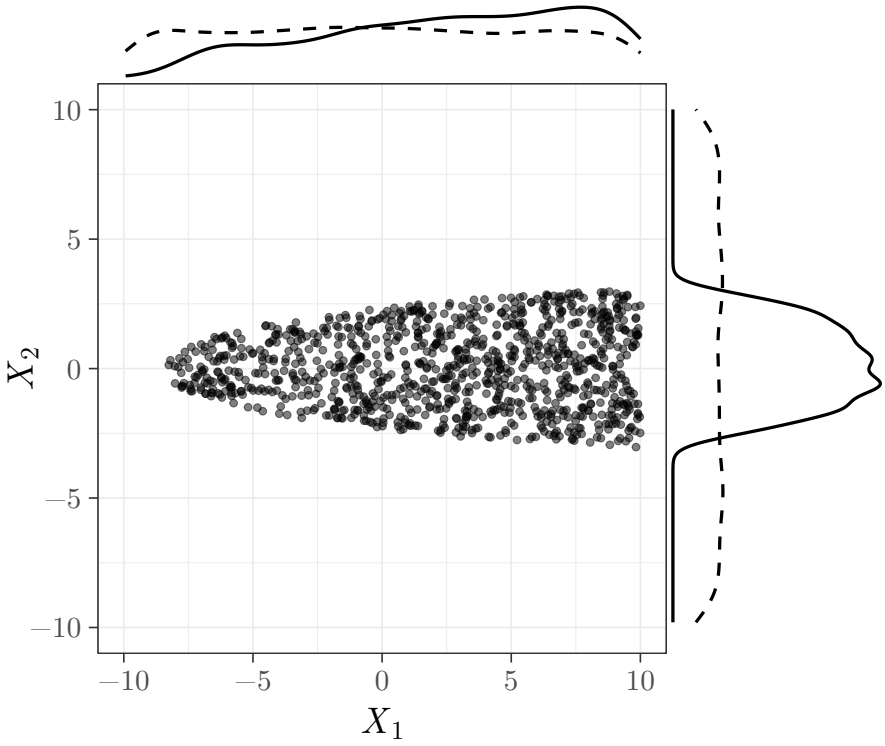

Figure 3.6 – Samples from $P_{\mathbf{X}|\mathbf{X}\in\mathcal{D}_q}$ and associated inputs marginal distributions for the Dixon-Price function. The original empirical distribution on the complete domain is also drawn in dashed lines. It is not completely uniform because of the finite size of the sample.

Another illustration is given by the following two-dimensional "Level" function whose behavior changes at a certain threshold q: above the threshold q, $f(\mathbf{X})$ only depends on X_1 but it only depends on X_2 below the threshold:

$$f(\mathbf{X}) = \begin{cases} |X_1| & \text{if} \quad |X_1| > q \\ |X_2 - 2| - 6 & \text{otherwise.} \end{cases} \tag{3.11}$$

Figure 3.8 shows the Level function for $q = 2.3$ defined for $X_i \sim \mathcal{U}[-5, 5]$, the threshold is represented to illustrate where the shift occurs. It can clearly be seen how the dependency changes from the picture. Figure 3.9 provides the HSIC-IT sensitivities $\hat{S}^{\text{HSIC}}_{q,\infty}(X_i)$, $i = 1, \ldots, 2$, for different α-quantile values. The vertical dashed line corresponds to the threshold of 2.3 (which is equal to the empirical 46% quantile of the output). It is observed that the unique dependency on X_1 is captured above that threshold where the sensitivity on X_2 is null, while both variables have a non-zero sensitivity below the threshold. Indeed, X_2 is negligible for

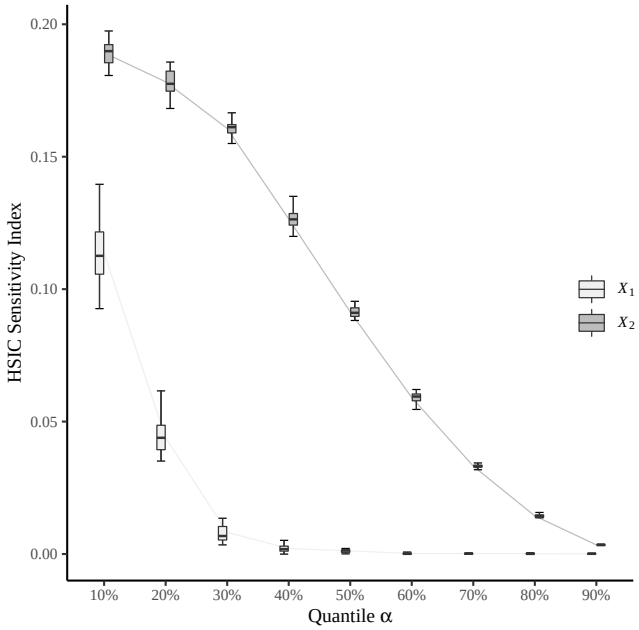

Figure 3.7 – Evolution of the HSIC-IT sensitivity indices w.r.t. the quantile α for the Dixon-Price function Equation (3.2).

reaching the set above q. Yet, below q, both inputs matter since X_1 is necessary to attain that area in a first place and X_2 matters to reach sub-areas below q. The fact that the HSIC-IT sensitivity of X_1 stays the same means that it has zero influence within the sublevel set $\mathcal{D}_q$, for all quantiles below the 46% quantile of the output.

In the following, we detail an optimization strategy including a preliminary step of sensitivity analysis with the HSIC-IT measures.

3.2 Optimization with dependence measures

The HSIC-IT measures naturally lead to an optimization strategy: the HSIC-IT are first calculated and, second, one must define a strategy to simplify the optimization problem considering the variables detected as negligible and finally carry out the optimization procedure. It aims at solving the following problem, obtaining the values for $\mathbf{X} \in \mathbb{R}^d$ inducing the best result for the objective function f, under a set of m constraints:

$$\begin{aligned} \min_{\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^d} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & g_l(\mathbf{X}) \leq 0, l = 1, \ldots, m \end{aligned} \tag{3.12}$$

.

3.2.1 Detecting important variables

As a preliminary step to the optimization, a sensitivity analysis is done in order to measure which inputs actually matter to reach certain levels of the objective function within the feasible region. We generate n points with fully-random Monte Carlo simulations $\mathbb{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^n)$ and compute the $\hat{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$, $i = 1, \ldots, d$, for multiple α values (typically $\alpha = [10\%, 40\%,$

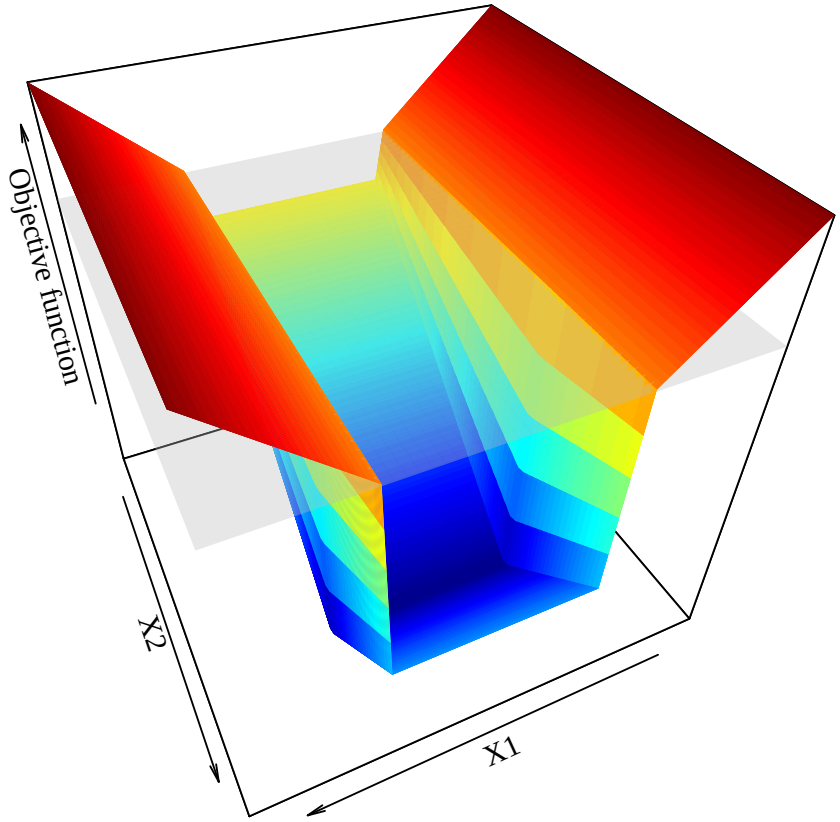

Figure 3.8 – Surface representation of the two-dimensional Level-Set function Equation (3.11). The dependence in variable shifts below the threshold $q = 2.3$ represented as a light grey surface in the drawing.

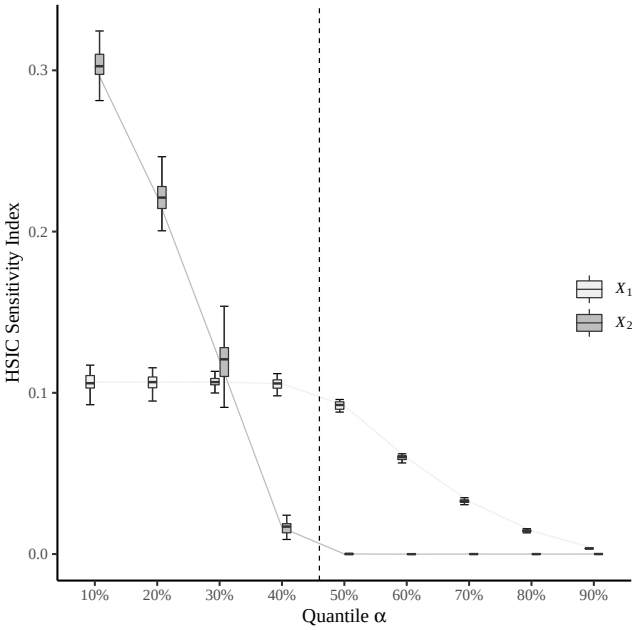

Figure 3.9 – Evolution of the HSIC-IT sensitivity indices w.r.t. the quantile α for the Level-Set function Equation (3.11).

$70\%, 100\%]$). The value of $\mathbf{T}$ is chosen to ensure a sufficient number of feasible points, typically around one hundred data samples. A Gaussian radial basis function (RBF) kernel is used for the inputs as it satisfies the characteristic property, making certain that the nullity of $\hat{S}^{\mathrm{HSIC}}_{q,\infty}(X_i)$ implies independence and that the variable X_i is negligible. The bandwidth parameter σ of the Gaussian RBF kernel is chosen as the median distance between points in the sample set $\mathbb{X}$. A linear kernel is used for the output, since it is modified into binary output using $Z = 1_{\mathcal{D}_{q_\alpha,\mathbf{T}}}$.

3.2.2 Modifying the optimization problem

After the sensitivity analysis, an input X_i is dubbed *negligible* for the optimization when its normalized $\tilde{S}^{\mathrm{HSIC}}_{q,\mathbf{T}}(X_i)$ is below a threshold $\tau = 0.1 \times \max_{i=1,\dots,d} \tilde{S}^{\mathrm{HSIC}}_{q,\mathbf{T}}(X_i)$ for a low α (here $\alpha = 10\%$). We set those inputs to a chosen value and reformulate the optimization problem: let $\mathcal{A}$ be the index set of active optimization variables whose HSIC-IT sensitivity index is above τ, and $\overline{\mathcal{A}}$ the complementary set of fixed variables, so that the initial number of variables is split into $d = \mathrm{card}(\mathcal{A}) + \mathrm{card}(\overline{\mathcal{A}})$. The modified optimization problem is

$$\begin{aligned} &\underset{X_i,\ i\in\mathcal{A}}{\text{minimize}}\ f(\mathbf{X}) \\ &\quad \text{where}\ \ X_j = x_j\ \ \text{is given},\ j \in \overline{\mathcal{A}},\ \mathbf{X} \in \mathcal{X}, \\ &\text{subject to}\ g_l(\mathbf{X}) \leq 0,\ l = 1, \dots, m. \end{aligned} \tag{3.13}$$

Two approaches for setting the non-active variables are studied:

- *Random* strategy: the negligible inputs, $x_j,\ j \in \overline{\mathcal{A}}$, are uniformly sampled from the restriction of $\mathcal{X}$ to its jth component at the beginning of the search.
- *Greedy* strategy: the negligible inputs are set to the values provided by the best feasible point of the sensitivity analysis; $x_j,\ j \in \overline{\mathcal{A}}$ is the j-th component of $\arg\min_{\substack{x^i, i=1,\dots,N \\ \mathbf{g}(x^i)\leq 0}} f(x^i)$.

Algorithm 1 summarizes the main steps of the method: the calculation of the sensitivity indices, the selection of the active variables and the patching of the inactive variables, and the final optimization. The algorithm details the computation of the indices, based on the estimator proposed in Equation (2.50). The optimization is carried out with the COBYLA algorithm [Pow94]. It is a local derivative-free optimization algorithm with nonlinear inequality and equality constraints which constructs successive linear approximations of the objective function and constraints and optimizes these approximations at each step. The implementation from the *nlopt* package [Ypm14] of the R language is used.

Including a preliminary step of sensitivity analysis allows a dimension reduction which directly reduces the cost of the optimization, since for most of the optimizers the computational cost is at least proportional to the problem dimension. One important aspect is the added cost of the sensitivity indices themselves since they require calls to the objective and constraints functions to compute q_α and then $\mathcal{D}_{q_\alpha,\mathbf{T}}$. If the feasible region is too small (α too low), a large number of points will be required to compute the HSIC-IT sensitivity indices. To overcome this issue, one can relax the problem with the coefficient $\mathbf{T}$.

Algorithm 1 Optimization with HSIC-IT sensitivity indices

Require: $\mathbb{X} = (\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n)$, $\mathbb{Y} = f(\mathbb{X})$, α, $\mathbf{T}$, τ

$q_\alpha = F_{\mathbb{Y}}^{-1}(\alpha) \leftarrow \alpha$-th empirical quantile of the output evaluations

$\mathcal{D}_{q_\alpha,\mathbf{T}} = \{\mathbf{X} \mid f(\mathbf{X}) \leq q_\alpha \cap \mathbf{g}(\mathbf{X}) \leq \mathbf{T}\}$

$\mathbb{Z} = 1_{\mathcal{D}_{q_\alpha,\mathbf{T}}}$ binary transformation of the output

$L \leftarrow l(z^p, z^r)$ assembly the transformed output Gram matrix, $p, r = 1, \ldots, n$

for $i = 1, \ldots, d$ **do**

$\mathbb{X}_i \leftarrow (X_i^1, X_i^2, \ldots, X_i^n)$

$K \leftarrow k(X_i^p, X_i^r)$ assembly the input Gram matrix, $p, r = 1, \ldots, n$

\# Estimate the i-th sensitivity index from Equation (3.7)

$\hat{S}_{q,\mathbf{T}}^{\text{HSIC}}(X_i) = \text{HSIC}(\mathbb{X}_i, \mathbb{Z}) = \frac{1}{n^2}\text{tr}(KHLH)$

end for

$\tilde{S}_{q,\mathbf{T}}^{\text{HSIC}}(X_i) = \hat{S}_{q,\mathbf{T}}^{\text{HSIC}}(X_i) / \sum_{i=1}^d \hat{S}_{q,\mathbf{T}}^{\text{HSIC}}(X_i)$ normalize indices

$\tau = 0.1 \times \max_{i=1,\ldots,d} \tilde{S}_{q_\alpha,\mathbf{T}}^{\text{HSIC}}(X_i)$

$\mathcal{A} \leftarrow \{i \mid i \in [1, d] \text{ and } \tilde{S}_{q,\mathbf{T}}^{\text{HSIC}}(X_i) \leq \tau\}$, $\overline{\mathcal{A}} \leftarrow [1, d] \setminus \mathcal{A}$

if *Random* strategy **then**

$X^{\text{fixed}} \leftarrow\sim U(\mathcal{X})$ uniform sample in search space

else if *Greedy* strategy **then**

$X^{\text{fixed}} \leftarrow \arg\min_{\substack{X^i \in \mathbf{X} \\ \mathbf{g}(X^i) \leq 0}} f(X^i)$

end if

Carry out the optimization on selected variables

$\mathbf{X}^* \leftarrow \arg\min_{\substack{X_i, i \in \mathcal{A} \\ \mathbf{g}(X_i) \leq 0}} f(\mathbf{X})$ where $X_j = X_j^{\text{fixed}}$, $j \in \overline{\mathcal{A}}$

3.3 Constrained optimization test problems

Tests will be carried out to compare the *Random* and *Greedy* problem formulations, to which we add the unmodified version of the problem, referred to as *Original*, where all d variables

are optimized. In this subsection, we purposely use a large number of points for the sensitivity analysis, with $n = 50000$, leaving for now the cost of the HSIC-IT indices aside to focus on the achievable values of the modified optimization problem.

Each estimation of $\hat{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$ is repeated 20 times to obtain a confidence interval on each index. The starting point for the optimizer are each of the 100 points of an optimized latin hypercube sampling (LHS). In order to test different settings for the *Random* version, we repeat this operation 100 times, with a different value for the negligible inputs, randomly sampled on the full domain. Using the same design of experiments for all versions would have been problematic as it cannot be guaranteed that the filling criterion is still respecter after the simplification. In the end, we obtain 10000 optimization runs for each version.

A budget of 500 calls to the objective function is given as the maximum number of calls and serves as a stopping criterion. The optimization also stops if too many consecutive steps give the same value or if the solution gap between two consecutive steps is too small. The comparisons will be based on the number of calls to the objective function at convergence and the performance of the solutions. In further results, a summary will be always presented showing the 10%, 50% and 80% quantiles of the quantity of interest here (namely the number of calls and the minimal feasible output value obtain). The best result obtained out of the 10000 runs is also an interesting baseline of comparison.

The following examples are two well-known engineering design test problem: the Gas Transmission Compressor Design problem [BP76] and the Welded Beam design problem [Deb00]. Table 3.1 summarizes characteristic features of both problems: the number of inputs d, the number of constraints m, the ratio in percent of the volume of feasible region to the volume of the complete design space, the best known feasible objective value and the corresponding $\mathbf{X}^*$.

Table 3.1 – Constrained optimization test problems.

Name	d	m	% feas. space	Best $f(\mathbf{X})$	Best known $\mathbf{X}^*$
GTCD	4	1	52.38	2964893.85	$[49.99, 1.178, 24.59, 0.389]$
WB4	4	5	$5.6 \cdot 10^{-2}$	1.7250	$[0.206, 3.473, 9.037, 0.206]$

3.3.1 Gas Transmission Compressor Design (GTCD)

The first example is a real-life problem about the design of a gas pipe line transmission system. The objective is to minimize its cost $f(X_1, X_2, X_3, X_4)$ under a nonlinear constraint. The problem objective function, constraint and search space are given below:

$$\begin{aligned} f(X) =& (8.61 \times 10^5) X_1^{1/2} X_2 X_3^{-2/3} X_4^{-1/2} + (7.72 \times 10^8) X_1^{-1} X_2^{0.219} \\ & - (765.43 \times 10^6) X_1^{-1} + (3.69 \times 10^4) X_3 \end{aligned}$$

s.t.

$$g_1(X) = X_4 X_2^{-2} + X_2^{-2} - 1 \leq 0$$

$$20 \leq X_1 \leq 50,\ 1 \leq X_2 \leq 10,\ 20 \leq X_3 \leq 50,\ 0.1 \leq X_4 \leq 60$$

Figure 3.10 shows the evolution of the conditional distributions for different quantiles α. Above each plot, the corresponding means and standard deviations, out of the 20 repetitions, of the normalized HSIC-IT sensitivities are given. X_3 is detected as negligible as its index is near

zero for the low quantile $\alpha = 10\%$. The near zero HSIC-IT of X_3 expresses the fact that its conditional probability distribution stays relatively close to the uniform distribution while other inputs see their distribution become increasingly skewed as α decreases.

The value chosen for X_3 in the *Greedy* modification of the optimization problem is 29.19 as it returned the best objective value during the sensitivity analysis, with the best point known in the literature being $\mathbf{X}^* = [49.99, 1.178, 24.59, 0.389]$.

Table 3.2 summarizes the results of the 10000 optimization runs with the 10%, 50% and 80% quantiles for the objective value after reaching a stopping criterion and the total number of calls. Complete histograms of the results are available in Figure 3.11. The *Greedy* version of the problem has a degraded optimum, $f(\mathbf{X}) = 2980651$, with respect to the *Original* one, $f(\mathbf{X}) = 2964895$. But the convergence is more robust, showing fewer runs that get trapped at local solutions and it is obvious from Table 3.2 that convergence is faster, with a median cost almost 4 times lower than the *Original* problem. The *Random* version has a cost similar to that of the *Greedy* formulation, but the cost functions at convergence vary significantly depending on the values chosen at random for the frozen inputs. In terms of the total number of feasible solutions obtained among all 10000 runs, no significant improvement can be notified between the *Original* and the *Greedy* versions, with 51.8% against 52.2% of runs leading to a feasible minimum output solution.

Both modified versions of the problem use significantly fewer calls to the objective function than the original formulation as it might be expected from problems with smaller search spaces. Although they return inferior solutions, the *Greedy* formulation is acceptable since it yields solutions close to the original optimum in a faster and more consistent manner. This difference in terms of performance might come from the difference between the value we chose for X_3 and the best value observed for X_3^* in the literature.

Table 3.2 – Quantiles 10%, 50% and 80% of minimum obtained (left) and of number of calls to the objective function at convergence, or after exceeding total budget, which explains the peak around 5000 calls, (right) in the GTCD test case. The best feasible value obtained among all the runs is also written. The optimization solver for these results is the COBYLA algorithm.

Minimum obtained				
Version	Best	10%	50%	80%
Original	2964895	2964897	2980175	3093426
Random	2964896	2968341	3040707	3203776
Greedy	2980651	2985406	2985993	3006087

Number of calls			
Version	10%	50%	80%
Original	214	502	502
Random	75	139	502
Greedy	74	131	502

3.3.2 Welded Beam (WB4)

This second example concerns a welded beam structure, constituted of a beam A and the weld required to hold it to the member B, see Figure 3.12. The objective is to minimize its fabrication cost $f(X_1, X_2, X_3, X_4)$ under 5 nonlinear inequality constraints. The optimization is summarized in the equations below:

$$f(X) = 1.10471 X_1^2 X_2 + 0.04811 X_3 X_4 (14 + X_2)$$

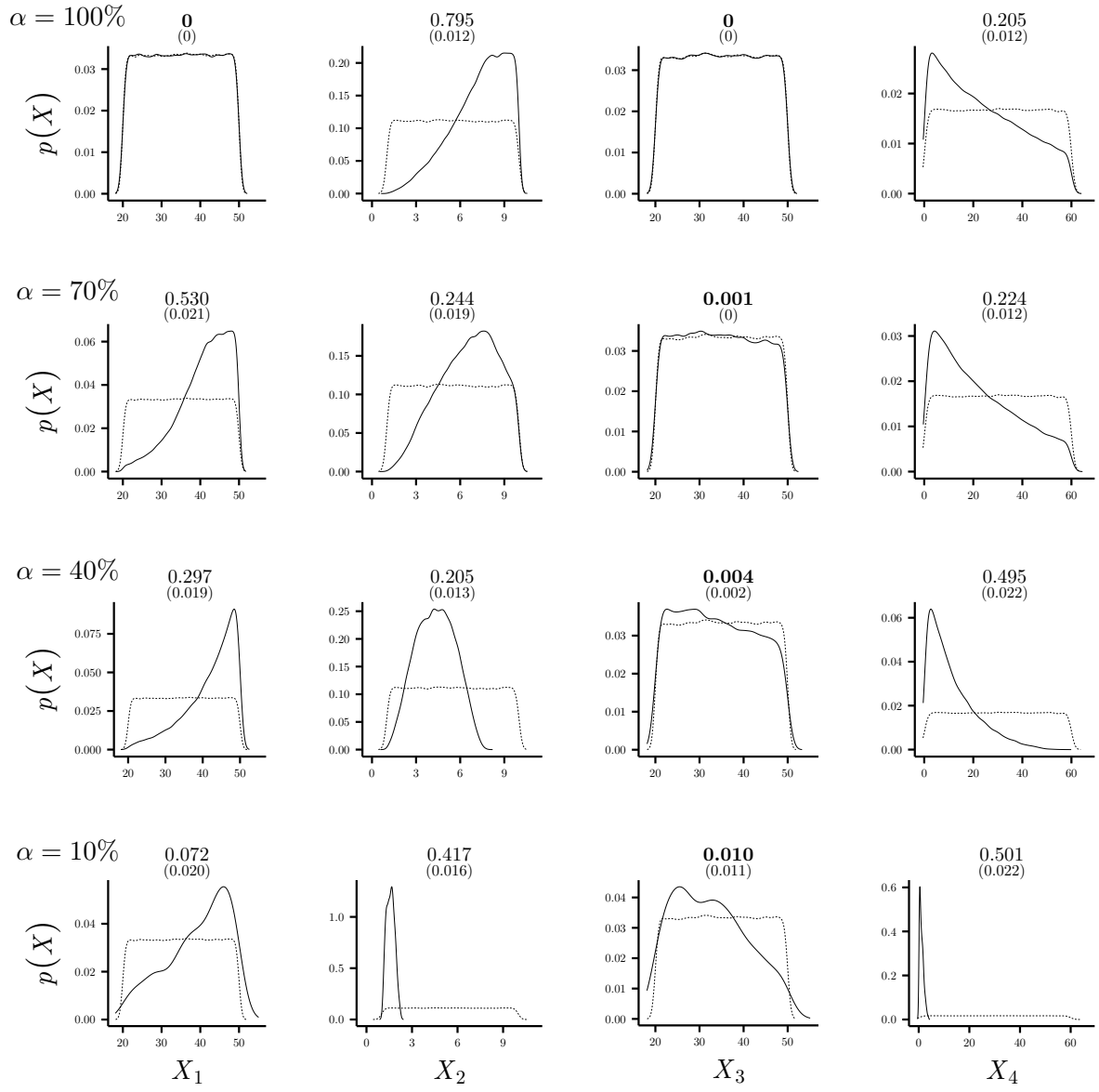

Figure 3.10 – Evolution of $p(X_i|Z = 1)(x_i)$ for different α values (continuous line) compared to the original distribution (dashed line) for the Gas Transmission Compressor Design. The continuous and dashed lines differ for $\alpha = 100\%$ because all points are not feasible. The numbers above each plot are the corresponding mean and standard deviation of $\tilde{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$. Bold numbers correspond to negligible X_i's, i.e., small $\tilde{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$'s. For better readability, the scales of the vertical axes vary.

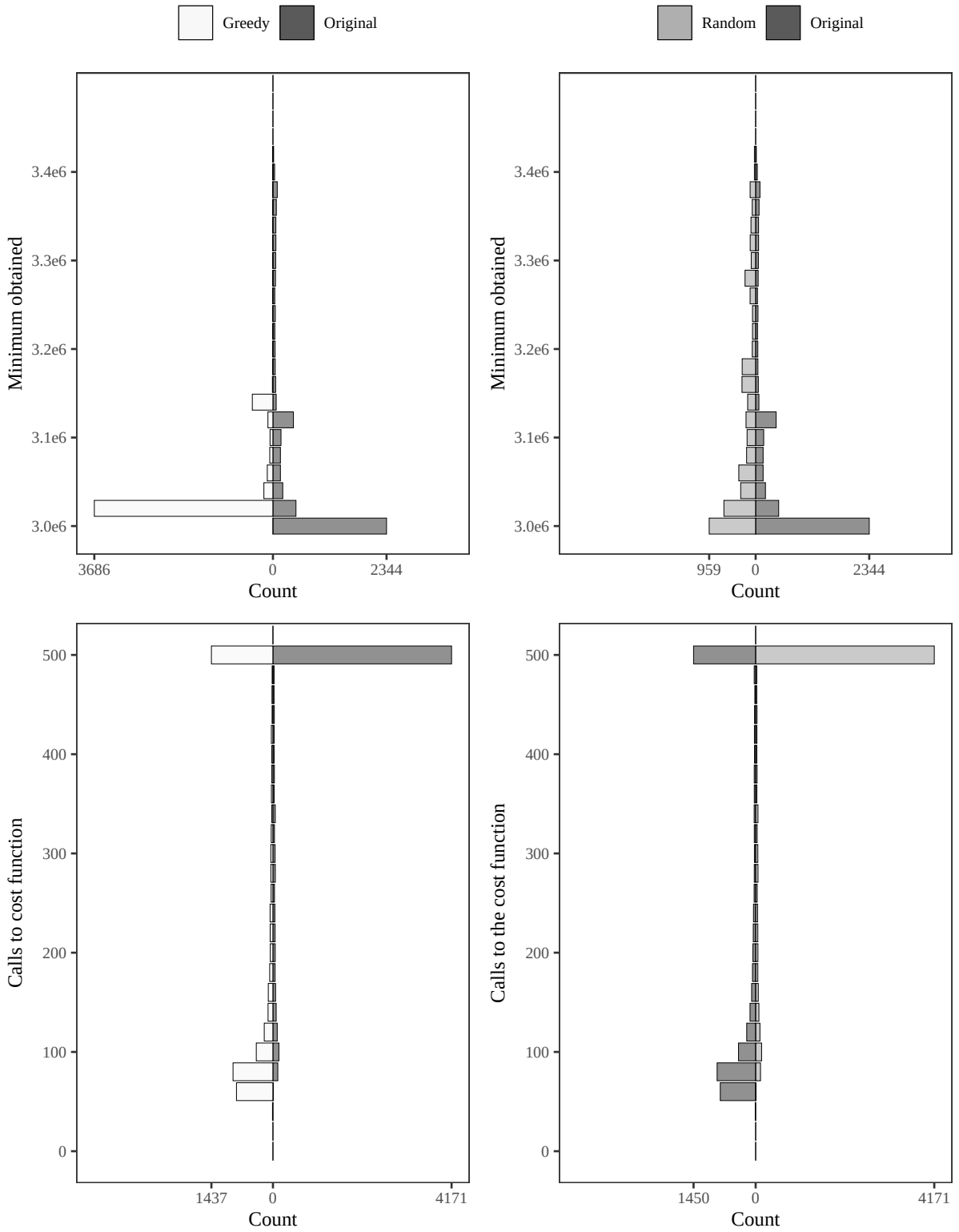

Figure 3.11 – Results of 10000 optimizations with the Original, Greedy and Random formulations for the Gas Turbine Compressor Design test case: histograms of the final objective functions (top) and number of calls to the objective function at convergence (bottom).

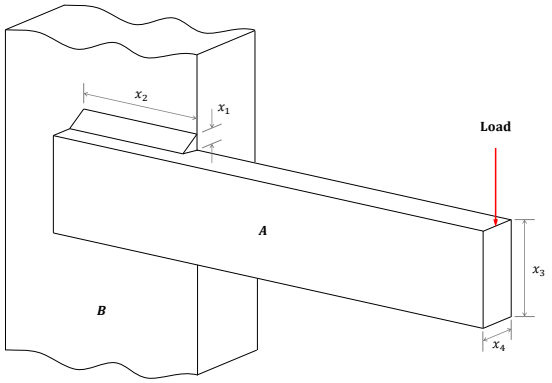

Figure 3.12 – Welded Beam.

s.t.

$$
\begin{aligned}
g_1(X) &= \tau(X) - 13600 \leq 0, \\
g_2(X) &= \sigma(X) - 30000 \leq 0, \\
g_3(X) &= X_1 - X_4 \leq 0, \\
g_4(X) &= 6000 - P_c(X) \leq 0, \\
g_5(X) &= \delta(X) - 0.25 \leq 0
\end{aligned}
$$

$$
0.125 \leq X_1 \leq 10,\ 0.1 \leq X_2 \leq 10,\ 0.1 \leq X_3 \leq 10,\ 0.1 \leq X_4 \leq 10
$$

The expression of the terms $\tau(X)$, $\sigma(X)$, $P_c(X)$ and $\delta(X)$ is:

$$
\begin{aligned}
\tau(X) &= \sqrt{\tau_1(X)^2 + \tau_2(X)^2 + X_2\tau_1(X)\tau_2(X)/\sqrt{0.25(X_2^2 + (X_1 + X_3)^2)}}, \\
\sigma(X) &= \frac{504000}{X_3^2 X_4}, \\
P_c(X) &= 102372.4(1 - 0.0282346X_3)X_3X_4^3, \\
\delta(X) &= \frac{2.1952}{X_3^3 X_4},
\end{aligned}
$$

where

$$
\begin{aligned}
\tau_1(X) &= \frac{6000}{\sqrt{2}X_1X_2}, \\
\tau_2(X) &= \frac{6000(14 + 0.5X_2)\sqrt{0.25\left(X_2^2 + (X_1 + X_3)^2\right)}}{2\left(\sqrt{2}X_1X_2(X_2^2/12 + 0.25(X_1 + X_3)^2)\right)}.
\end{aligned}
$$

Figure 3.13 shows the evolution of the conditional distributions $P_{X_i|Z=1}$ for different quantiles α. The mean and standard deviations of the HSIC-IT sensitivities associated to each α, out of 20 repetitions, can be found above each plot. X_2 and X_3 are found to be negligible as their index is near zero for $\alpha = 10\%$. Their domains are only slightly restricted by the condition on performance, $Z = 1$.

For the *Greedy* problem modification, X_2 is set to 5.36 and X_3 to 8.54 as those values gave

the lowest feasible objective function value during the sensitivity analysis. For reference, the optimal point found in the literature is $\mathbf{X}^* = [0.206, 3.473, 9.037, 0.206]$.

Results are summarized in Table 3.3 with the 10%, 50% and 80% quantiles for the function objective value after reaching a stopping criterion and the total number of calls. Complete histograms of the results are available in Figure 3.14. From the results, with the original formulation, the global optimum of performance $f(\mathbf{X}^*) = 1.72$ is reached in half of the cases, while the *Greedy* modification to the problem converges to a downgraded value of $f(X^*) = 1.97$ at a much lower cost, with a median number of 49 calls to the objective function. As can be observed in Table 3.3, the *Random* modification to the problem yields inconsistent objective function values at convergence, because many choices of "negligible" inputs lead to poor final achievable performance.

Once again, the freezing of some of the variables leads to savings in terms of calls to the objective function and to more robust convergences for the *Greedy* version. Furthermore, it seems that the modified version no longer has the local optimum around $f(X^*) = 11$ that is seen as a small mode in the top of Figure 3.14 in the original results. Furthermore, in this test problem, removing some variables leads to a decent improvement in terms of feasibility as 39.6% of the runs give a feasible solution with the *Original* while the *Greedy* version obtains a 49.1% rate of success.

Table 3.3 – Quantiles 10%, 50% and 80% of minimum obtained (left) and of number of calls to the objective function at convergence, or after exceeding total budget, which explains the peak around 5000 calls, (right) in the WB4 test case. The best feasible value obtained among all the runs is also written. The optimization solver for these results is the COBYLA algorithm.

Minimum obtained				
Version	Best	10%	50%	80%
Original	1.7244	1.7249	1.7252	2.5411
Random	1.7919	2.0828	3.9290	7.3730
Greedy	1.8618	1.8626	1.8628	1.8628

Number of calls			
Version	10%	50%	80%
Original	135	401	502
Random	34	46	63
Greedy	38	49	59

3.3.3 High dimensional versions of the test cases

The two previous examples are low dimensional problems and the cost of the sensitivity indices was not analyzed. In order to be more representative of real-world problems, we reiterate the study of higher dimensional versions of the same test cases by adding 46 dummy variables to increase the dimension from $d = 4$ to $d = 50$.

For both augmented test problems, a latin hypercube sampling of only 500 points, unlike the several thousands used in Section 3.3, is optimized with a maximin criterion before serving for the computation of the HSIC-IT indices. Figures 3.15 and 3.16 show the sensitivity indices obtained for both problems, without repetitions. The dashed line corresponds the threshold of detection still equal to $\tau = 0.1 \times \max_{i=1,\dots,d} \tilde{S}^{\text{HSIC}}_{q_\alpha,\mathbf{T}}(X_i)$.

Results are consistent with those of Sections 3.3.1 and 3.3.2 as we select the same variables as negligible among the real ones (X_3 and (X_2, X_3) for the GTCD and WB4 problems, respectively). However, because of the limited amount of points given, fake variables are sometimes above the detection threshold τ, hence deemed as important despite having no impact on the function. Even with these estimation errors, the dimension is drastically reduced and the influ-

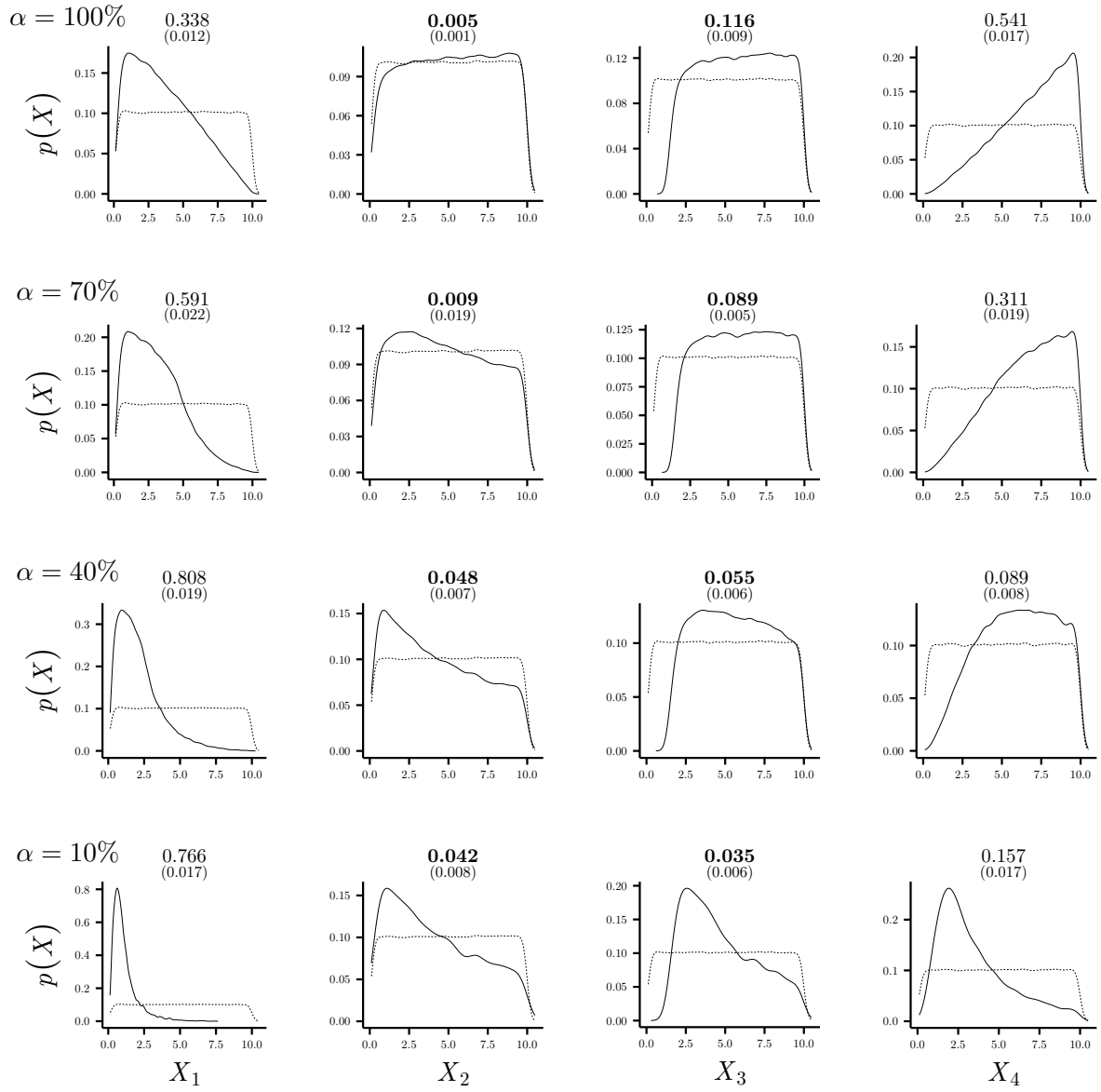

Figure 3.13 – Evolution of $p(X_i|Z=1)(x_i)$ for different α value (continuous line) compared to the original distribution (dashed line) for the Welded Beam application. The continuous and dashed lines differ for $\alpha = 100\%$ because not all points are feasible. The values above each plot are the corresponding mean and standard deviation of $\tilde{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$. Bold numbers correspond to negligible X_i's, i.e., small $\tilde{S}^{\text{HSIC}}_{q,\mathbf{T}}(X_i)$'s. For better readability, the scales of the vertical axes vary.

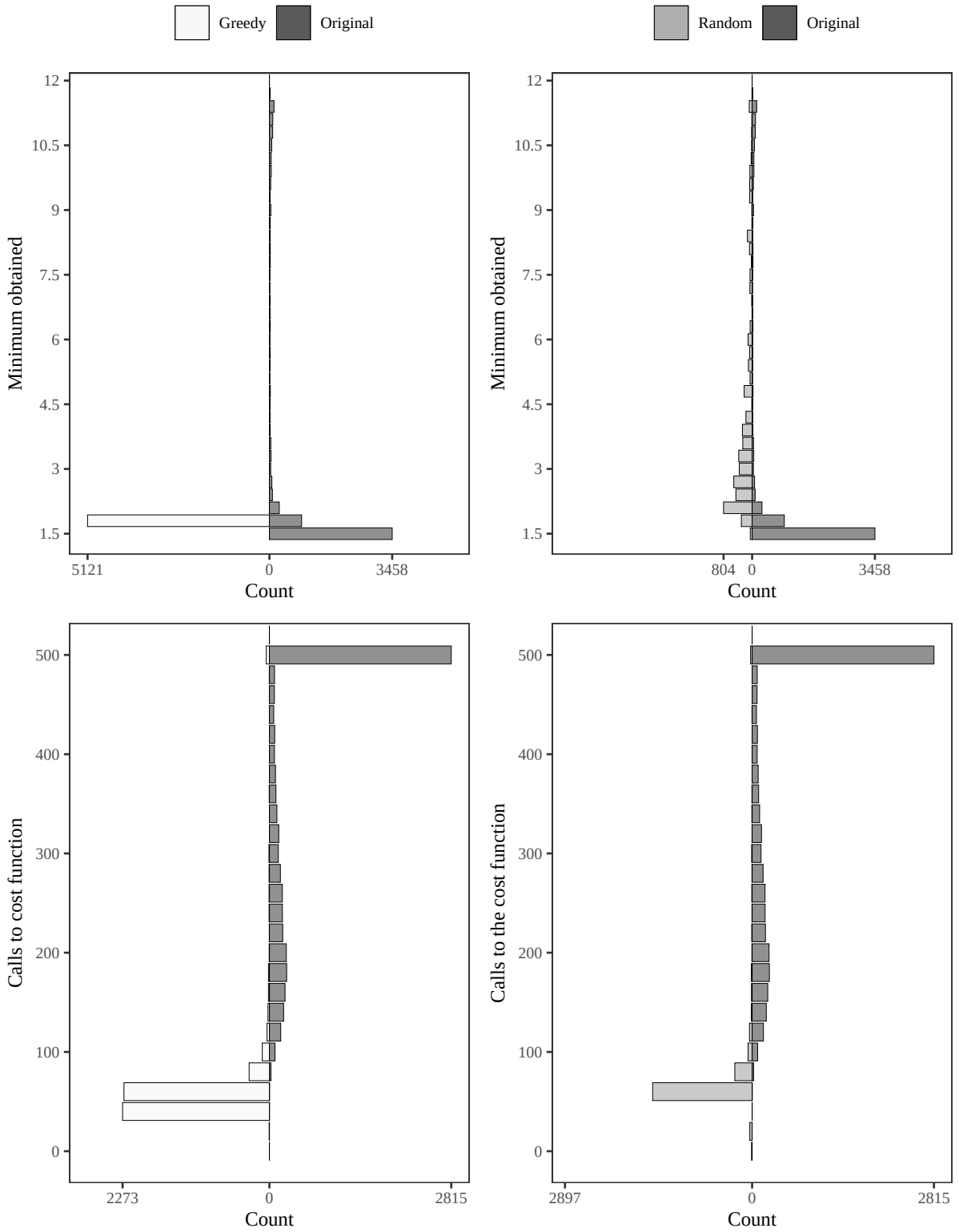

Figure 3.14 – Results of 10000 optimizations with the Original, Greedy and Random formulations for the Welded Beam test case: histograms of the final objective functions (top) and number of calls to the objective function at convergence (bottom).

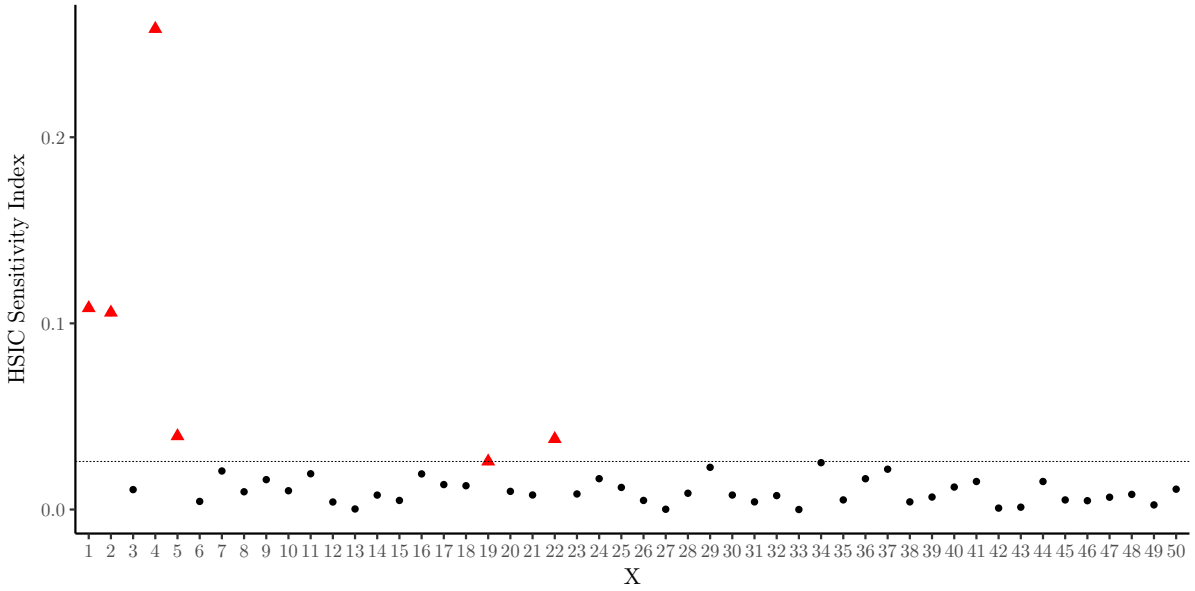

Figure 3.15 – HSIC sensitivity indices for the GTCD problem with $d = 50$. Indices are computed for $\alpha = 10\%$. Red triangles are the indices above the detection threshold.

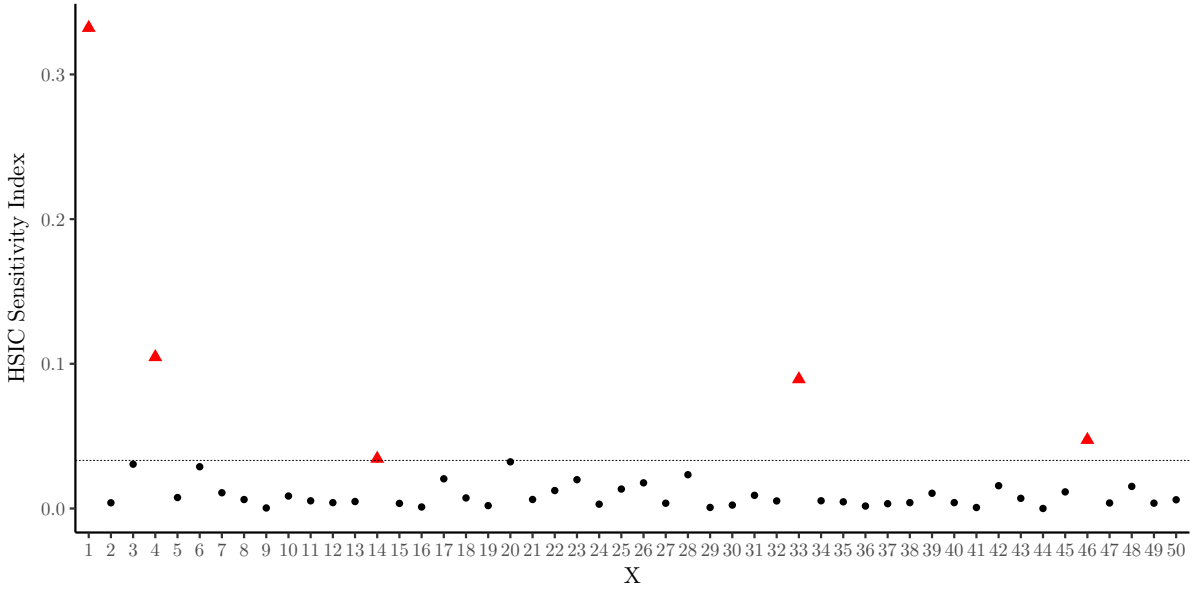

Figure 3.16 – HSIC sensitivity indices for the WB4 problem with $d = 50$. Indices are computed for $\alpha = 10\%$. Red triangles are the indices above the detection threshold.

ential variables selected. For the optimization, we only consider the *Greedy* and the *Original* modifications as the *Random* yields poor results. The maximum budget is increased to 5000 calls to the objective function to match the dimensionality augmentation but we lower the number of repetitions to only 20. The cumulative number of calls to the objective function from the sensitivity analysis (500 calls) and the optimization in reduced dimension is much lower than the optimization alone in dimension 50 for both problems, especially for the WB4 function, see Tables 3.4a and 3.4b.

The value obtained after convergence for the reduced problem is still slightly degraded as observed in Section 3.3. Overall, even with the added cost of the sensitivity indices, these results

Table 3.4 – Quantiles 10%, 50% and 80% of minimum obtained and of number of calls to the objective function at convergence (or after exceeding total budget, which explains the peak around 5000 calls) in the high dimensional test cases, with 46 additional variables. The best feasible value obtained among all the runs is also written. The optimization solver for these results is the COBYLA algorithm. The number of calls for *Greedy* results do not include the preliminary calls for computing the sensitivity indices.

(a) GCTD high dimensional problem

Minimum obtained				
Version	Best	10%	50%	80%
Original	2964894	2964948	2974736	3084906
Greedy	2985406	2985406	2985469	2991863

Number of calls			
Version	10%	50%	80%
Original	5002	5002	5002
Greedy	133	364	2069

(b) WB4 high dimensional problem

Minimum obtained				
Version	Best	10%	50%	80%
Original	1.7246	1.7249	1.7793	2.3809
Greedy	1.8628	1.8628	1.8628	1.8628

Number of calls			
Version	10%	50%	80%
Original	2813.8	5002	5002
Greedy	117	149	178

Table 3.5 – Quantiles 10%, 50% and 80% of values at convergence (or after exceeding total budget) on the left table, and of calls to the objective function on the right table, in the high dimensional test case. The optimization solver for these results is the SQP algorithm. The number of calls for *Greedy* results do not include the preliminary calls for computing the sensitivity indices but consider the additional calls required by the approximation of the gradient of the objective function.

(a) GCTD high dimensional problem

Minimum obtained				
Version	Best	10%	50%	80%
Original	2964894	2964895	2964895	2966038
Greedy	2985404	2985406	2985406	2985406

Number of calls			
Version	10%	50%	80%
Original	3673	4591	5000
Greedy	393	491	645

(b) WB4 high dimensional problem

Minimum obtained				
Version	Best	10%	50%	80%
Original	1.7220	1.7249	1.7249	1.7249
Greedy	1.8628	1.8628	1.8628	1.8628

Number of calls			
Version	10%	50%	80%
Original	1633	2143	2959
Greedy	121	181	243

show important gains in terms of calls to the objective function, especially in the case of the Welded Beam problem.

We chose to use the COBYLA algorithm as it is a good off-the-shelf option for derivative-free optimization with inequality constraints. Yet, the flowshart we followed for the optimization of the problems, simplified in Figure 3.17, is independent of the choice of the optimizer. Indeed, instead of the local derivative-free algorithm, it is possible to use a derivative-based algorithm or a global algorithm.

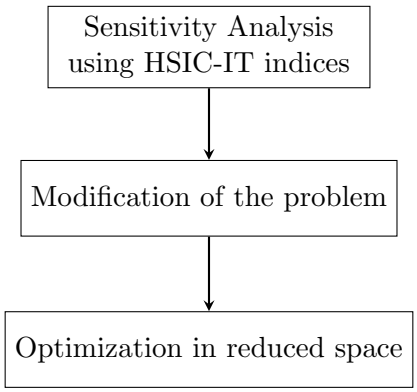

Figure 3.17 – Simplified flowchart of an optimization study including a sensitivity analysis step.

Runs of both problems (GTCD and WB4) in high-dimensional were repeated with a derivative-based algorithm: the Sequential Quadratic Programming (SQP) algorithm from the *nlopt* package in the R language. SQP optimizes successive second-order (quadratic or least-squares) approximations of the objective function, with first-order (affine) approximations of the constraints.

Since derivatives are not directly available for our problem, their approximation requires extra calls (d per iteration) to the objective function. Logically, the dimension reduction provides again a significant drop in the number of calls during the optimization. This is clearly visible in Table 3.5a where the number of objective function execution is decreased by a factor 10 for the best 10% optimization runs, without including the initial cost of the computation of the indices.

3.3.4 Further discussion

Premature convergence with local optimizers

As seen in both above examples, setting variables with small HSIC-IT indices to a fixed value chosen with the *Greedy* strategy led to significant improvements in terms of optimization cost and robustness, with an accompanying small degradation in performance at the optimum. This is due to the loss in fine-tuning ability resulting from freezing the value of the low impact inputs. This phenomenon was more visible with the *Random* strategy where the variations in values of fixed inputs led to a spread in final objective functions. This might also seems counter-intuitive as we stated that the value was non-influential, hence, one might consider that its value should not impact performance as observed.

We now argue, with the help of an illustrative example, that this impact is increased when the reduced problem is solved with a local optimization algorithm, such as in Sections 3.3.1 and 3.3.2 (using COBYLA as the local optimizer). Let us consider the following two dimensional "twisted

strip" toy function:

$$f(\mathbf{X}) = \begin{cases} 10 - (|X_1'| - A)^2 - \epsilon X_2' X_1' & \text{if} \quad |X_1| \geq A \\ 10 - \epsilon X_2' X_1' & \text{otherwise} \end{cases}$$

with $\mathbf{X}' = \mathbf{X} - \mathbf{c}$. The function is represented in Figure 3.18 below for $\mathbf{c} = (0.1, 0.1)$, $A = 0.2$ and $\epsilon = 0.1$. This function possesses a global optimum at $(-1, -1)$ (the red square) and multiples local ones (the black squares), with a significant difference in the objective value (9.069 for the global optimum against 9.289, 9.429 or 9.609 for local ones).

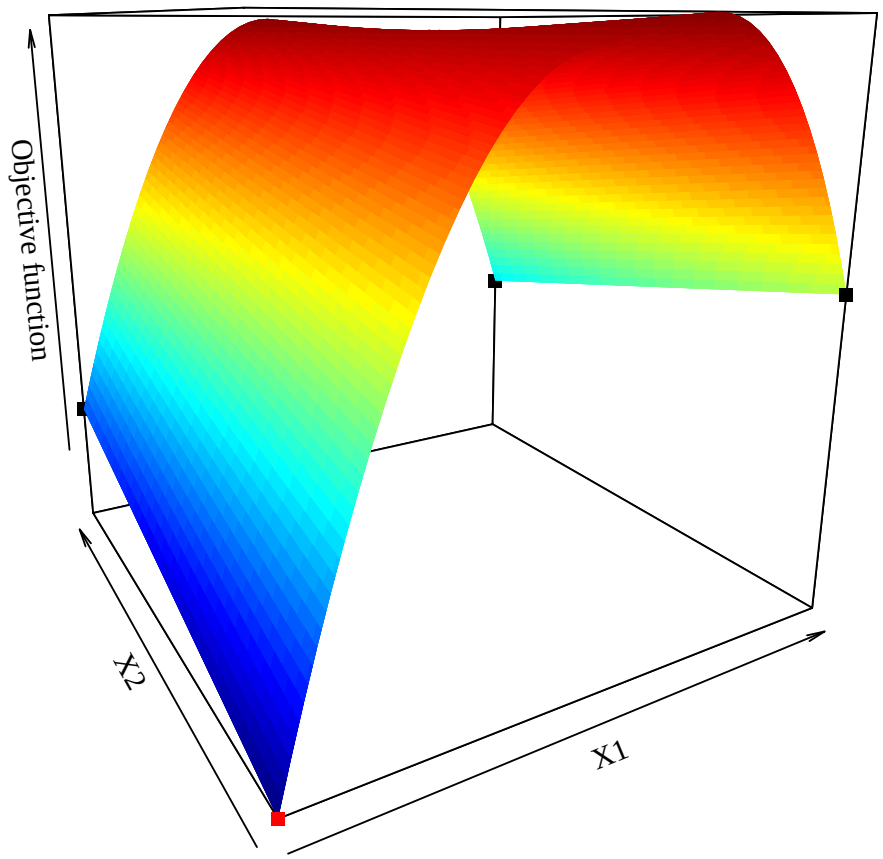

Figure 3.18 – Surface plot of the "twisted strip" function with $\mathbf{c} = (0.1, 0.1)$, $A = 0.2$ and $\epsilon = 0.1$

For this toy function, the HSIC-IT sensitivity index of the second variable is arbitrarily small, even for low quantiles. This can be seen by imagining the marginal distribution of X_2 when f is restricted to low values, which is very close to the uniform distribution. Indeed, the twisted

strip function is almost flat in the X_2 direction. Setting X_2 to a constant value simplifies the problem as it appears to be negligible. Whatever the chosen value of X_2, the reduced objective function has a global optimum at $X_1 = -1$ and a local one at $X_1 = 1$. The main difference between these 2 cases lies in the slope direction of the reduced function near $X_1 = 0 \pm A$, see the different profiles of the function in Figure 3.19. That implies that a local optimization algorithm will be sensitive to its initialization and will sometimes converge to the local optimum. Hence, depending of the choice made for the value of X_2, the frequency of convergence to the global optimum varies, increasing when X_2 is at its optimum (-1) with a success in 67% of the cases and decreasing when away from it with a success in 42% of the cases, compared to the 55% success rate for the original problem. Such behavior should be expected from functions with essential global optima, i.e., functions without "needle in the haysack", where the modified optimization problems lead to the global basin of attraction if the frozen variables are close to their optimum. In such well-behaved cases, the *Greedy* heuristic gives $X_2 \approx -1$, leading to improved results.

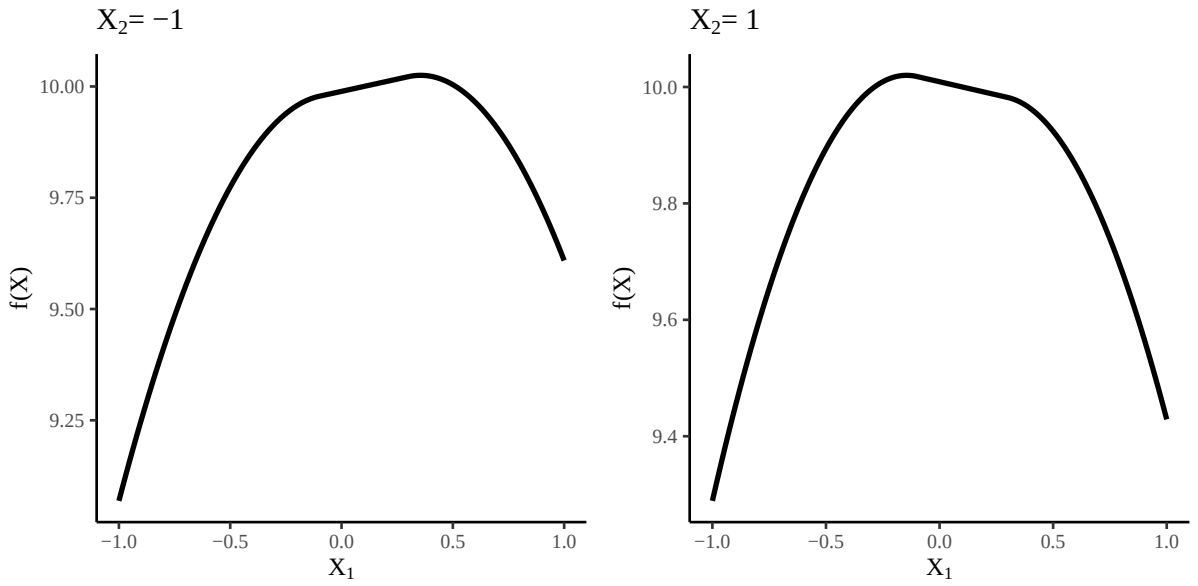

Figure 3.19 – Profile of the reduced objective function for different X_2 values.

Note that the phenomenon of convergence to a local optimum and its dependency on the frozen variables would be much lessened if a global optimizer were used: the strategy proposed in this paper of HSIC-IT sensitivity analysis followed by a greedy freezing of some variables and an optimization would further benefit from a global optimizer. Furthermore, another possibility would be to re-use an optimizer starting from the point obtained in the reduced space, but this time in the full domain in order to converge to the true optimum. For high dimensional functions, the cumulative cost of the three operations (estimation of the sensitivity indices, first optimization in reduced space and second optimization in the full space) can result in a lower number of evaluations than a single optimization in the full space.

Constraint issues

Because we restricted the data set with the indicator function, the initial size of the samples must be big enough to have sufficient points for the sensitivity analysis study. For example, if we consider an initial set of 1000 points, if one is interested by the inputs giving the 10% quantile

and below, it means that only a tenth of the 1000 points will be available. If we add constraints functions, the feasible region can become quite small, about few percents of the design space. However, because the estimation of HSIC Equation (3.7) scales at least quadratically with the sample size, large datasets will be too expensive to compute.

In order to cope with that issue, [Gre+05] employs a low rank decomposition on each Gram matrix by using an incomplete Choleski factorization [BJ02]. More recently, [Zha+18] introduced several strategies as replacements to the quadratic estimator. First of all, derived from [ZGB13], the block-based strategy estimates the HSIC on a small block of data and then averages the final estimated HSIC over all blocks. The second approach is based on a Nyström approximation [WS01], a classical low-rank kernel approximation technique. Finally, they also introduced a last estimator that uses Random Fourier Features (RFF) [RR08]. This method replaces the implicit feature map provided by the kernel by an explicit mapping to a low-dimensional Euclidean inner product space using a randomized feature map. However, this method only works with translation invariant kernels, such as the Gaussian RBF one.

3.4 Conclusions

This chapter has shown how global sensitivity analysis can be specialized for contributing to the resolution of optimization problems.

First, we have introduced three modifications of the objective function that are alternative expressions of the feasible level set idea. Each formulation is a different blend between two pieces of information, which inputs matter to reach an area close to the optima and how much each input impacts performance when being in such an area. The effect of each formulation on the Sobol indices has been observed.

Second, building on the indicator-thresholding formulation in conjunction with the Hilbert Schmidt independence criterion, we have described a new HSIC-IT sensitivity index adapted to constrained optimization problems. This sensitivity index has been interpreted as a measure of the distance between two distributions, that of the variable being analyzed and that of the same variable conditional to its objective and constraints reaching a certain performance level.

Finally, the new HSIC-IT index has served to select variables before a local optimization is carried out. Provided that the variables which are not retained are given a value in a greedy manner, we have obtained in several test cases solutions with limited performance loss, at a substantially decreased number of function evaluations, and with more stable convergence.

However, the cost of calculating the indices was practically left aside in the whole Chapter. We have only tackled one of the issues: efficiently reducing the dimension in an optimization problem. However, when the function is expensive, the number of evaluations should be reduced at all cost. It is customary to rely on a few calls to the model to build surrogate cheaper to call and faster to evaluate. In the next chapter, we introduce surrogate modeling with a focus on Gaussian Processes. We explain how optimization strategies are often derived from the surrogate and how the search can be improved with a proper selection of the design variables.

Chapter take-home messages

- A new sensitivity index was proposed. It is adapted to optimization problems. It relies on kernel-based sensitivity indices, using embeddings of distribution and distance between embeddings to assess the relevance of the different inputs.
- The dimension reduction in the different test cases allows to obtain a more stable convergence in fewer calls to the objective function, at the cost of a decrease in optimum accuracy due to the dimensions removal.

Bibliography

[BJ02] Francis R Bach and Michael I Jordan. "Kernel independent component analysis". In: *Journal of machine learning research* 3.Jul (2002), pp. 1–48.

[BP76] Charles S.. Beightler and Donald Thomas Phillips. *Applied Geometric Programming*. Wiley, 1976.

[Deb00] Kalyanmoy Deb. "An efficient constraint handling method for genetic algorithms". In: *Computer methods in applied mechanics and engineering* 186.2 (2000), pp. 311–338.

[DV15] Sébastien Da Veiga. "Global sensitivity analysis with dependence measures". In: *Journal of Statistical Computation and Simulation* 85.7 (2015), pp. 1283–1305.

[Gre+05] Arthur Gretton et al. "Measuring statistical dependence with Hilbert-Schmidt norms". In: *International conference on algorithmic learning theory*. Springer. 2005, pp. 63–77.

[Oll+17] Yann Ollivier et al. "Information-geometric optimization algorithms: A unifying picture via invariance principles". In: *Journal of Machine Learning Research* 18.18 (2017), pp. 1–65.

[Pow94] Michael JD Powell. "A direct search optimization method that models the objective and constraint functions by linear interpolation". In: *Advances in optimization and numerical analysis*. Springer, 1994, pp. 51–67.

[RR08] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems*. 2008, pp. 1177–1184.

[WS01] Christopher KI Williams and Matthias Seeger. "Using the Nyström method to speed up kernel machines". In: *Advances in neural information processing systems*. 2001, pp. 682–688.

[Ypm14] Jelmer Ypma. *Introduction to NLoptr: an R interface to NLopt*. Tech. rep. Technical report, 2014.

[ZGB13] Wojciech Zaremba, Arthur Gretton, and Matthew Blaschko. "B-test: A non-parametric, low variance kernel two-sample test". In: *Advances in neural information processing systems*. 2013, pp. 755–763.

[Zha+18] Qinyi Zhang et al. "Large-scale kernel methods for independence testing". In: *Statistics and Computing* 28.1 (2018), pp. 113–130.

Chapter 4

Online high-dimensional optimization

Contents

4.1 Surrogate modeling

Whenever the model considered in the optimization (or in any other study that requires a large number of model evaluations) is expensive, estimating the sum of the calls to the model becomes intractable. In such cases, it is common to rely on substitutes to the functions, called surrogate models, metamodels, proxies or *models of the models* in the literature [Kle87]. The surrogate $\hat{f}(\mathbf{X})$ mimics the behavior of the true function $f(\mathbf{X})$ in the following sense:

$$f(\mathbf{X}) = \hat{f}(\mathbf{X}) + \epsilon(\mathbf{X})$$

where $\epsilon(\cdot)$ is an approximation error. Usually, surrogate modeling involves the successive steps shown in Figure 4.1. In the following, we briefly detail each step with some examples for each,

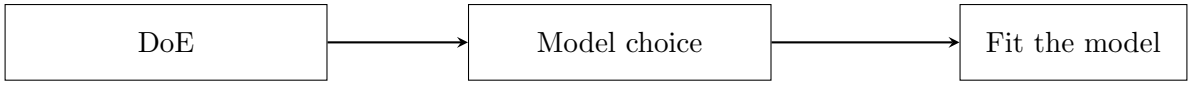

Figure 4.1 – Different steps for building a surrogate model.

but without being fully exhaustive.

The experimental design $\mathbb{X}$ where the model will be queried to obtain the corresponding simulations $\mathbb{Y}$ should be chosen so that the surrogate model encompasses as much information as possible using as few simulations as possible. Omitting the case of linear surrogates for which optimal designs are known (A/D/E-optimality [Dea+15]), the most general designs are *space-filling* as they attempt to fill the design space $\mathcal{X}$. Firstly, the full or fractional factorial designs are based on geometrical patterns to fill the design domain, whose sizes increase drastically with the dimension of the domain and the number of levels considered. Secondly, [Nie92] introduced the concept of discrepancy, defined as the deviation of a given sequence from a uniform distribution in the design domain, motivating the so-called low-discrepancy sequences. The Halton or the Sobol sequences are examples of it, showing lower discrepancies than full Monte Carlo sample and thus ensuring a better coverage of the space. Finally, the latin hypercube sampling proposed by [MBC79] generates random design of experiments but guarantees uniformity of the sample on each input marginal domain, hence resulting in a better space filling than a full Monte Carlo design. Figure 4.2 represents the three aforementioned design of experiments and a fully random one for comparison. After being generated, the LHS is usually optimized (while remaining an LHS) using different criteria, such as the maximin [Pro17], where the smallest distance between two points of the design is maximized. Alternatively, the minimax procedure [Pro17] is the minimization of the maximal distance between any point of the design and its nearest neighbor. The cost of the minimax is often considered as prohibitive as the nearest neighbor to every point must repeatedly be searched for.

After designing an appropriate design of experiments and computing the required simulations, the following step is to choose a type of model for the approximation of the true function and to fit the model to the observations. Many surrogates exist in the literature: general linear models, polynomial models, random forests, support vector machines, neural networks and Gaussian processes. Throughout this chapter, we rely on the latter which will soon be further detailed. For an illustration purpose, we give here a short explanation on the first technique as it is rather simple and remains widely used. The general linear models are a linear combination

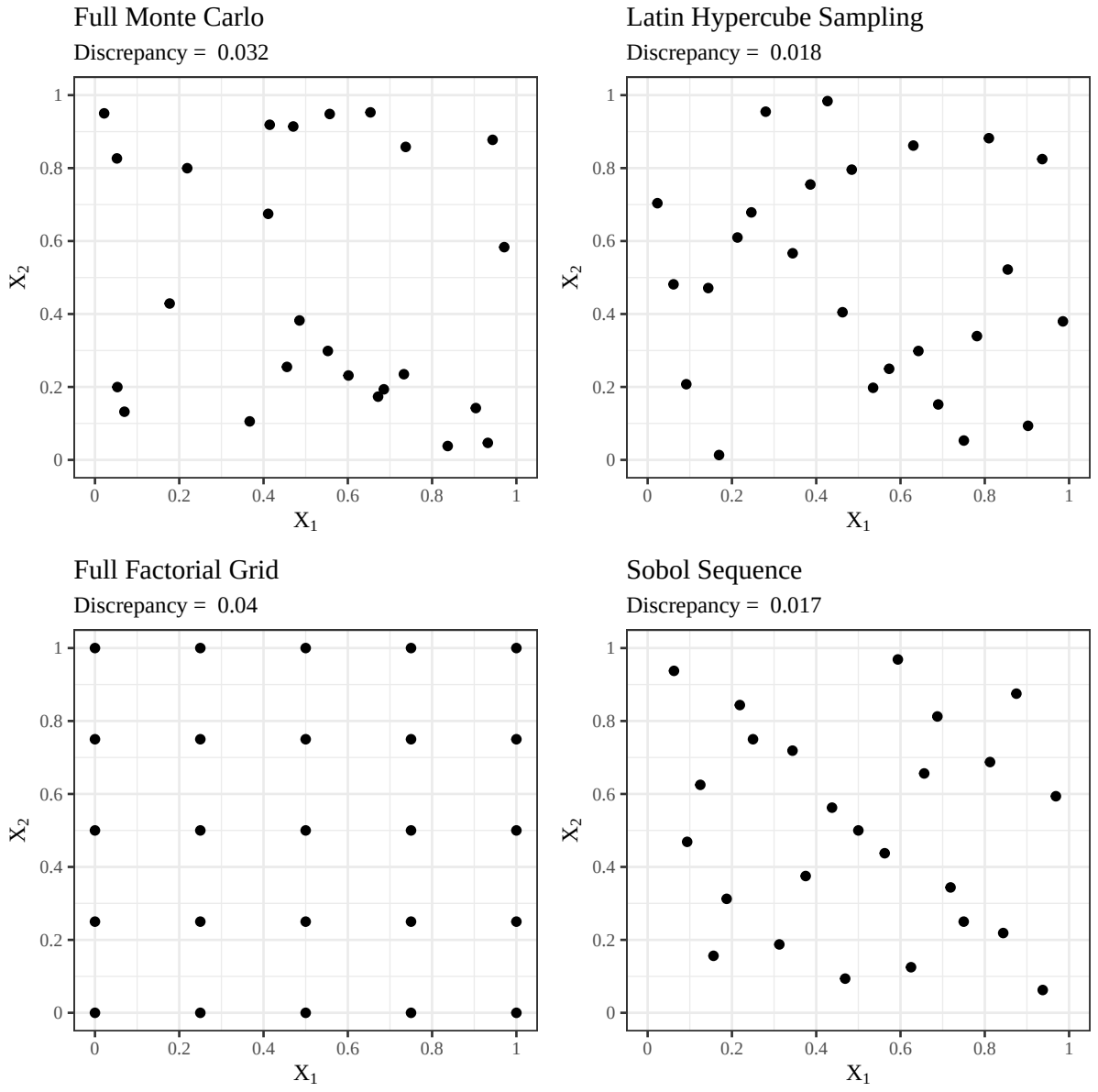

Figure 4.2 – Illustration of four different design of experiments (fully random, latin hypercube sampling, full factorial grid and a Sobol sequence) made of 25 points in a two-dimensional design domain. The corresponding discrepancy of the design of experiment is written. Note that a fully random design has a lower discrepancy than the full factorial grid.

of a finite set of p preselected functions $\mathbf{h} = \{h_i, i = 1, \dots, p\}$

$$\hat{f}(\mathbf{X}) = \sum_{i=1}^{p} \beta_i h_i(\mathbf{X}) = \beta^T \mathbf{h}(\mathbf{X}) \tag{4.1}$$

with $\beta = (\beta_i, i = 1, \dots, p) \in \mathbb{R}^p$ is a vector of coefficient that has to be determined using the design of experiments. One can sometimes add a term β_0 defined as a bias coefficient. Commonly used functions $\mathbf{h}(\cdot)$ include low-order polynomials as in Equations (4.2) and (4.3) or Fourier series.

$$\hat{f}(\mathbf{X}) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i \tag{4.2}$$

$$\hat{f}(\mathbf{X}) = \beta_0 + \sum_{i=1}^{d} \beta_i X_i + \sum_{i=1}^{d} \beta_{ii} X_i^2 + \ldots + \sum_{1 \leq i < j \leq d} \beta_{ij} X_i X_j \tag{4.3}$$

The parameters of the polynomials in Equations (4.2) and (4.3) are usually determined using least squares regression which requires that the size of $\mathbb{X}$ is greater than the number p of functionals h. However, because p is finite, it assumes that the function has a specific (say polynomial) shape, which may not be the case for real world problems. Furthermore, full polynomial expansions have a number of terms that grows rapidly with the dimension d of the problem.

Once the model is chosen and fitted, it can be used to predict simulations at any unobserved input in $\mathcal{X}$. Throughout the following, we use as surrogate models Gaussian processes (GP) for multiple reasons. First of all, Gaussian processes approximations have the appealing property of interpolating the true function at observed samples (such that $\hat{f}(\mathbf{X}) = f(\mathbf{X})$ for any $\mathbf{X} \in \mathbb{X}$). This does not hold when the output function encompasses uncertainties but this is outside the scope of this thesis. Another important property is that with any prediction of the surrogate model, an associated variance is available allowing to quantify the uncertainty of the metamodel which reflects a lack of information in the design of experiments. This is not a measure of the error coming from the approximation itself but rather a measure of the (in)accuracy of the model in areas where no observations were queried. The rest of the section gives a more detailed description of what Gaussian processes are.

4.1.1 Gaussian processes regression

Formally, a Gaussian process is *a collection of random variable, any finite number of which have a joint Gaussian distribution* [WR06]. It is completely described by its mean function $m(\mathbf{X})$ and covariance function $k(\mathbf{X}, \mathbf{X}')$, defined for a real process $Y(\mathbf{X})$ as

$$\begin{aligned} m(\mathbf{X}) &= \mathbb{E}(Y(\mathbf{X})) \\ k(\mathbf{X}, \mathbf{X}') &= \mathbb{E}((Y(\mathbf{X}) - m(\mathbf{X}))(Y(\mathbf{X}') - m(\mathbf{X}')) \end{aligned} \tag{4.4}$$

and the Gaussian process is written as

$$Y(\mathbf{X}) \sim \mathcal{GP}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X}')). \tag{4.5}$$

In an analogy with pointwise surrogates, a GP not only returns a prediction of the function, but also an estimation of its error in the form of the mean and variance of a normal dis-

tribution over the possible value of f at $\mathbf{X}$. By definition, the mean function characterizes the expected function value at $\mathbf{X}$, i.e. the average evaluation at input $\mathbf{X}$ for all functions in the distribution. Different priors on the mean can be considered, leaving the mean as an unknown function $m(\mathbf{X}) = \sum \beta_i h_i(\mathbf{X})$ with coefficients to estimate. The covariance function $k(\mathbf{X}, \mathbf{X}') = \text{Cov}(f(\mathbf{X}), f(\mathbf{X}'))$ represents the dependence between the function value at different input points $\mathbf{X}$ and $\mathbf{X}'$. Note that this covariance between the outputs is actually written as a function of the inputs, with k being called the kernel function of the Gaussian process.

Note that the kernel considered for the Gaussian process and the one used in the kernel-based sensitivity analysis Section 2.2 are different as they serve different purposes. The GP kernel used in the Gaussian process is linked to the assumptions about which class of functions f belongs to while the embedding kernel in the sensitivity analysis framework expresses which characteristics of the good X distributions are being compared.

The choice of the GP kernel is directly based on the assumptions about the function f, such as its smoothness or repeating patterns (symmetry, periodicity, spectral content). Usually, the correlation between two points decreases as a function of the distance between the points, meaning that a pair of function outputs calculated at points close to each other should be more similar than outputs at points which are further away from each other. Popular choices for k are the exponential kernel

$$k(X_i, X_j) = \sigma^2 \exp\left(-\frac{\|X_i - X_j\|}{2\theta}\right), \tag{4.6}$$

the squared exponential kernel, or Gaussian kernel,

$$k(X_i, X_j) = \sigma^2 \exp\left(-\frac{\|X_i - X_j\|^2}{2\theta^2}\right), \tag{4.7}$$

and the Matèrn kernel [Mat13]

$$k_\zeta(X_i, X_j) = \sigma^2 \frac{2^{1-\zeta}}{\Gamma(\zeta)} \left(\frac{2\sqrt{\zeta}\|X_i - X_j\|}{\theta}\right)^\zeta H_\zeta \left(\frac{\sqrt{2\zeta}\|X_i - X_j\|}{\theta}\right) \tag{4.8}$$

with $\Gamma(\cdot)$ and $H_\zeta(\cdot)$ the Gamma function and the Bessel function of order ζ. One should note that when $\zeta \longrightarrow \infty$ the Matèrn kernel is equal to the squared exponential kernel and when $\zeta = 1/2$ the Matèrn kernel is equal to the exponential kernel. For all kernels, the signal variance σ and the lengthscale θ, denoted as *hyperparameters* control the a priori correlation between points. We show how the correlation decays for all three aforementioned kernels in the left panel of Figure 4.3.

Specifying a covariance function implies a distribution over functions: one can draw samples from this distribution evaluated at any number of points. Let $\mathbb{X} = \{\mathbf{X}^1, \ldots, \mathbf{X}^n\}$ be a set of input points, the corresponding covariance matrix is typically built using Equations (4.6) and (4.7) or Equation (4.8) elementwise. Values of Y at inputs $\mathbb{X}$ from the Gaussian Process are obtained by sampling from the following multivariate normal distribution

$$Y(\mathbb{X}) \sim \mathcal{N}(H\beta, k(\mathbb{X}, \mathbb{X})) \tag{4.9}$$

with $H = (h(\mathbf{X}^1), \ldots, h(\mathbf{X}^n))^T$ the experimental matrix such that $H\beta = \sum \beta_i h_i(\mathbf{X})$ and where

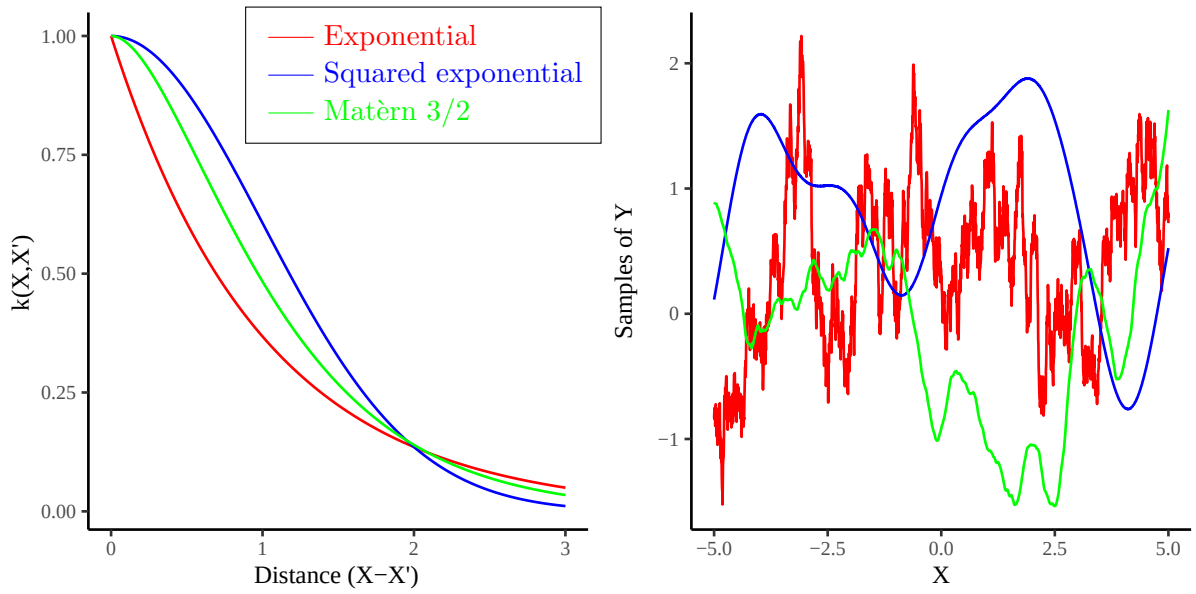

Figure 4.3 – Left: Covariance functions for the exponential, the squared exponential and the Matèrn kernels where $\zeta = 3/2$, with $\theta = \sigma = 1$. Right: random functions drawn from Gaussian processes priors with the different covariance functions. The colors match the corresponding covariance function.

the symmetric kernel matrix is given by

$$k(\mathbb{X}, \mathbb{X}) = \begin{bmatrix} k(\mathbf{X}^1, \mathbf{X}^1) & \dots & k(\mathbf{X}^1, \mathbf{X}^n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{X}^n, \mathbf{X}^1) & \dots & k(\mathbf{X}^n, \mathbf{X}^n) \end{bmatrix} \tag{4.10}$$

where the diagonal of this matrix is equal to 1 (since each point is perfectly correlated with itself). One can then plot the generated values as a discretized function of the inputs, see the right panel of Figure 4.3.

As sampling functions from the prior distribution is usually not the prime interest, we can incorporate knowledge acquired from already sampled points $\{\mathbb{X}, \mathbb{Y} = f(\mathbb{X})\}$ on the function and infer on new inputs $\mathbf{X}$ (assuming here that observations were noise-free). This is known as Gaussian process regression, also called *kriging* in the geostatics field [Mat73].

The different parameters in this regression must be calibrated: the coefficients β of the mean function m and the different kernel hyperparameters, with the correlation lengths λ and the variance σ. The usual strategy for calibration is the maximum likelihood ($\mathcal{L}$) estimation [WR06], a method that sets hyperparameters to values that maximize the likelihood to observe the model realizations. Because of the Gaussian assumption, the likelihood function is

$$\mathcal{L} = \frac{1}{(2\pi)^{n/2}|K|^{1/2}} \exp\left(-\frac{1}{2}(\mathbb{Y} - H\beta)^T K^{-1}(\mathbb{Y} - H\beta)\right) \tag{4.11}$$

with $K = k(\mathbb{X}, \mathbb{X})$ the covariance matrix on observations, $H = (h(\mathbf{X}^1), \dots, h(\mathbf{X}^n))^T$ the experimental matrix, $\mathbb{Y} = f(\mathbb{X})$ the observed realizations, and β the regressive coefficients. In the noise-free setting, the covariance matrix is factorized into $K = \sigma^2 R$ where R is the correla-

tion matrix that only depends on θ. $\hat{\beta}$ and $\hat{\sigma}^2$ can be analytically expressed as functions of θ following

$$\begin{aligned} \hat{\beta} &= (H^T R^{-1} H)^{-1} H^T R^{-1} \mathbb{Y} \\ \hat{\sigma}^2 &= \frac{1}{n} (\mathbb{Y} - H\hat{\beta})^T R^{-1} (\mathbb{Y} - H\hat{\beta}). \end{aligned} \tag{4.12}$$

The hyperparameters can be found by maximizing the likelihood function $\mathcal{L}$ Equation (4.11)

$$\theta^* = \arg\max_{\theta \in \Theta} \mathcal{L}(\theta) = \arg\min_{\theta \in \Theta} -\log \mathcal{L}(\theta) \tag{4.13}$$

with Θ the definition domain for θ. After plugging in the expressions of $\hat{\beta}$ and $\hat{\sigma}^2$, the log likelihood function reads

$$-2 \log \mathcal{L}(\theta) = n \log(2\pi) + n \log(\hat{\sigma}^2) + \log |R| + n \tag{4.14}$$

An analytic expression of the gradient of the likelihood is obtained thanks to the Gaussian prior assumption [PB01] and the objective Equation (4.13) is usually solved using either quasi-Newton local optimization (such as the L-BFGS method [LN89]) or genetic algorithms. Other approaches exist for the calibration of the hyperparameters, for example the minimization of the cross-validation error [WR06]. They are not detailed here.

We consider that our Gaussian process has zero mean, $m(\mathbf{X}) = 0$, since it simplifies the writing of the equations posterior to the observations. By definition the joint distribution between already evaluated points $Y(\mathbb{X})$ and function Y can be written as

$$\begin{bmatrix} Y(\mathbb{X}) \\ Y(\mathbf{X}) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(\mathbb{X}, \mathbb{X}) & k(\mathbb{X}, \mathbf{X}) \\ k(\mathbf{X}, \mathbb{X}) & k(\mathbf{X}, \mathbf{X}) \end{bmatrix}\right)$$

where $k(\mathbb{X}, \mathbb{X})$ is the covariance matrix between all observed inputs so far, $k(\mathbf{X}, \mathbf{X})$ is the covariance matrix between newly introduced points and $k(\mathbb{X}, \mathbf{X})$ is the covariance matrix between past and new inputs. Getting the posterior distribution over functions would consist in keeping only the functions which agree with the observed inputs. Graphically, one might generate functions from the prior, and reject the ones that do not match the observations, but this would result in an extremely inefficient method. Fortunately, using standard results [WR06], the posterior conditional distribution $p(Y \mid \mathbb{X}, \mathbb{Y}, \mathbf{X})$ is a multivariate Gaussian distribution with mean

$$\mu(\mathbf{X}) = k(\mathbf{X}, \mathbb{X}) k(\mathbb{X}, \mathbb{X})^{-1} \mathbb{Y} \tag{4.15}$$

and variance

$$s^2(\mathbf{X}) = k(\mathbf{X}, \mathbf{X}) - k(\mathbf{X}, \mathbb{X}) k(\mathbb{X}, \mathbb{X})^{-1} k(\mathbb{X}, \mathbf{X}) \tag{4.16}$$

The posterior, that we write $\hat{F}(\mathbf{X}) \sim \mathcal{N}(\mu(\mathbf{X}), s^2(\mathbf{X}))$ is also a Gaussian process and calculating its mean and variance is possible with simple operations through Equations (4.15) and (4.16). The posterior mean corresponds to a weighted average between the prior mean (0 here) and an estimate based on observations while the posterior variance is equal to the prior covariance minus a variance reduction achieved thanks to the observations. Both elements are sufficient statistics of the posterior distribution probability and efficient strategies to compute both Equations (4.15) and (4.16) are presented in [WR06]. Typically, the direct matrix inversion is replaced by a Cholesky decomposition as it is more stable and faster to compute. Additionally, a small

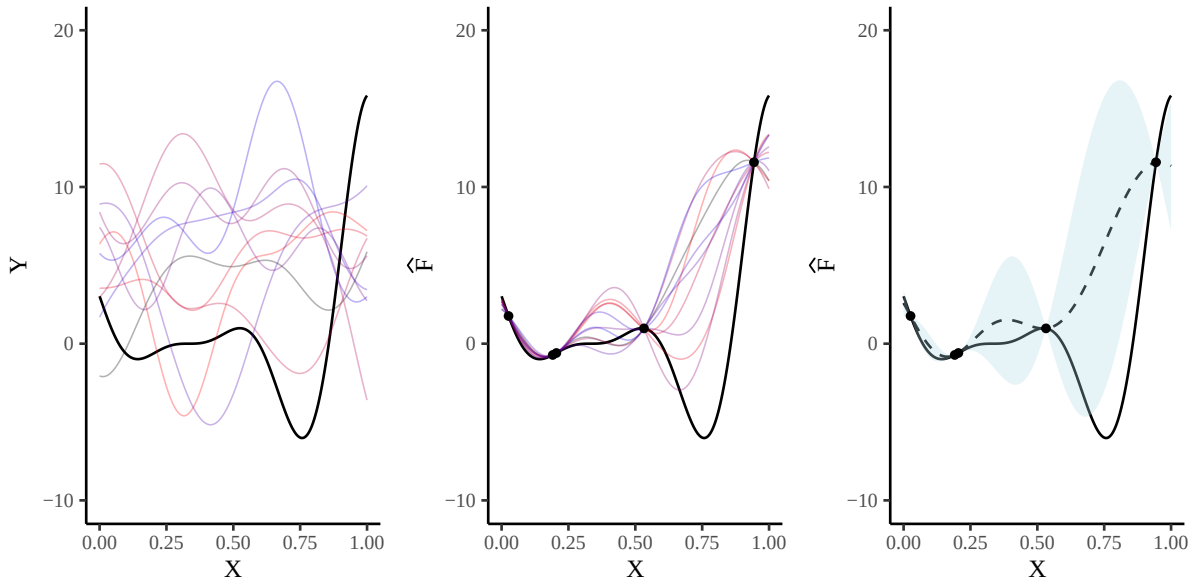

Figure 4.4 – Left: Functions drawn at random from a Gaussian process prior, the black line corresponds to the true function. Middle: Functions drawn from the posterior i.e. the prior conditioned on the five noise free observations represented by the black dots. Right: Prediction from the Gaussian process posterior with its mean in dashed line (compared to the true function drawn with the continuous line) and its confidence interval as the lightblue shaded area.

positive term is often added to the diagonal of the prior covariance to improve the numerical stability, especially when two observations are close to each other.

We can illustrate Gaussian process regression on the following Forrester function, a one-dimensional function defined for $X \in [0, 1]$ as

$$f(X) = (6X - 2)^2 \sin(12X - 4) \tag{4.17}$$

Let $(\mathbb{X}, \mathbb{Y})$ be a set of five observations randomly selected on $[0, 1]$. We choose a squared exponential kernel as written in Equation (4.7) and build the Gaussian process predictor given the observations as defined before. We obtain a mean prediction and a corresponding predictor variance as illustrated in Figure 4.4. The shaded area in the right figure is the 95% confidence intervals on the prediction at a given x.

The predictor mean interpolates exactly the observations and the variance is equal to zero at these locations. As the distance from one of the observations increases, the variance rises, an expected phenomenon as in this case[1] the covariance functions is monotone with respect to the distance to the known points. Again, the associated variance is due to a lack of knowledge in areas where observation points are missing and it is not a measure of the error of the approximation made by the surrogate model.

Hence, Gaussian process regression provides a powerful tool to model partially known functions. The key point now is to find an efficient way to explore and exploit the approximation model obtained with the Gaussian process predictor. This is often represented as the exploitation-exploration trade off in the global optimization paradigm. The overall idea is to characterize the relevance of a new candidate, through a function called the acquisition function. This

[1] covariance functions need not be monotonous, e.g. periodic covariances.

function assesses the utility of new points in terms of allowing one to learn the function as well as possible (this is exploration) or producing the best possible output (this is exploitation).

Properly calibrating the Gaussian process then exploiting it in a regression, along with the choice of the acquisition function and its optimization are crucial aspects of Bayesian optimization, described in the following.

4.1.2 Bayesian optimization

Bayesian optimization is based on Bayesian inference, using a Gaussian process prior distribution on the function f that reflects our belief about its behavior through the covariance function. Once past functions observations have been made, the GP prior becomes a Gaussian process posterior distribution $\hat{F}$ described by its mean μ and variance s^2. Bayesian optimization relies on the posterior GP to choose where to sample following points through the optimization of an acquisition function denoted $a(\cdot)$. The associated cost is low since it does not involve new calls to the model f. In a sense, the role of the acquisition function is to guide the search towards optima of the function while guaranteeing that the posterior GP improves in the regions of interest. The overall process helps in reducing the number of function evaluations, making Bayesian optimization a powerful approach for the black-box optimization of expensive functions, as considered in this thesis. Using a Gaussian prior within a Bayesian optimization process was first proposed in the late 1970s [O'H78; Žil80].

After constructing a posterior distribution $\hat{F}$ over the function, the usefulness of points candidates for a future evaluation is assessed through the acquisition function $a(\cdot)$ and the point that maximizes $a(\cdot)$ is selected. The value of the acquisition function always depends on the current posterior of the Gaussian process. After choosing a new point, the corresponding output is observed and the Gaussian process is updated accordingly and the whole process reiterates. Hence, Bayesian optimization can be viewed as an iterative procedure based on a Gaussian process regression and proceeding with the optimization of the acquisition function to know where to query new observations. This is summarized in Algorithm 2.

Algorithm 2 General Bayesian optimization

Require: Data samples $\{\mathbb{X}, \mathbb{Y} = f(\mathbb{X})\}$; Acquisition function $a(\cdot)$;
  GP prior for f with mean function m and kernel k ; budget
  for $t = 1, 2, \ldots$ budget **do**
    Choose $\mathbf{X}^t = \arg\max_{\mathbf{X}\in\mathcal{X}} a(\mathbf{X})$
    Sample $f(\mathbf{X}^t)$
    Augment data and update the GP posterior
  end for

In the following, we describe different acquisition functions among which acquisition functions based on the improvement and the confidence bounds.

Probability of improvement

This strategy, proposed in the work of [Kus64], maximizes the *probability of improvement* over the current achieved observation values denoted $f_{\min} = \arg\min_{\mathbf{X}\in\mathbb{X}} f(\mathbf{X})$ so that

$$a^{\mathrm{PI}}(\mathbf{X}) = P(\hat{F}(\mathbf{X}) \leq f_{\min}) = \Phi\left(\frac{f_{\min} - \mu(\mathbf{X})}{s(\mathbf{X})}\right) \tag{4.18}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a normal law. Maximizing the previous equation relates to a pure exploitation of the model, indeed, locations with high probability of being really close but lower than $f_{\min}$ will be drawn more often than points with larger improvements and more uncertainty. In order to circumvent this, [Kus64] introduced a coefficient $\xi \geq 0$ called the trade-off parameter:

$$a^{\mathrm{PI}}(\mathbf{X}) = P(\hat{F}(\mathbf{X}) \leq f_{\min} - \xi) = \Phi\left(\frac{f_{\min} - \mu(\mathbf{X}) - \xi}{s(\mathbf{X})}\right) \tag{4.19}$$

Following [Kus64], the value of ξ should start fairly high to allow exploration of the model and decreases to zero as the optimization is carried on. The value of the trade-off parameter was empirically studied in different works such as [Jon01; Liz08].

Expected improvement

Finding an acquisition function that not only takes into account the probability of being lower than the current best observation, but also how far is it from $f_{\min}$ is an important refinement over the previous acquisition function. [MTZ78] proposed the improvement function with respect to $f_{\min}$, which says that sampling a new point $\mathbf{X}$ brings an improvement equal to $f_{\min} - \hat{F}(\mathbf{X})$ if $f_{\min} > \hat{F}(\mathbf{X})$ and equal to 0 otherwise. More compactly, the improvement is written

$$I(\mathbf{X}) = \max(0, f_{\min} - \hat{F}(\mathbf{X})). \tag{4.20}$$

The acquisition function called the *expected improvement* is defined as

$$a^{\mathrm{EI}}(\mathbf{X}) = \mathbb{E}(I(\mathbf{X}) \mid \mathbb{X}, \mathbb{Y}) \tag{4.21}$$

where $\mathbb{E}(\cdot|\mathbb{X}, \mathbb{Y})$ means that we take the expectation under the posterior distribution given the observations $\mathbb{X}$ and the corresponding evaluations $\mathbb{Y}$. The expected improvement can be evaluated in closed form using integration by parts, [JSW98], resulting in

$$a^{\mathrm{EI}}(\mathbf{X}) = (f_{\min} - \mu(\mathbf{X}))\Phi\left(\frac{f_{\min} - \mu(\mathbf{X})}{s(\mathbf{X})}\right) + s(\mathbf{X})\phi\left(\frac{f_{\min} - \mu(\mathbf{X})}{s(\mathbf{X})}\right) \tag{4.22}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are, respectively, the cumulative and probability distribution functions of the standard Gaussian distribution. The analytic expression of the expected improvement allows to obtain analytic evaluations of its gradient and higher order derivatives, as well as fast evaluations of it.

The Bayesian optimization algorithm then evaluates the point which maximizes the acquisition function, hence the largest expected improvement

$$\mathbf{X}^t = \arg\max_{\mathbf{X} \in \mathcal{X}} a^{\mathrm{EI}}(\mathbf{X}) \tag{4.23}$$

This approach was named *Efficient Global Optimization* (EGO) by Jones in [JSW98]. It balances the two features of points that are a high expected quality (a low mean μ) and a high uncertainty (a large standard deviation s) therefore adjusting the search behavior between exploration and exploitation. To further control the exploration-exploitation trade-off, [Jon01] proposed an approach in the flavor of what was defined previously for the probability of improvement with the introduction of a trade-off parameter ξ.

Confidence bound criterion

As explained earlier, Bayesian optimization aims at finding the input that leads to the minimum output (here in an optimization problem without constraints) as efficiently as possible

$$\mathbf{X}^* = \arg\min_{\mathbf{X}\in\mathcal{X}} f(\mathbf{X}). \tag{4.24}$$

In synthetic studies, a way to quantify the quality of this search is to measure regret as the difference between the current obtained output and the best possible output, defined as the *simple* or *instantaneous* regret:

$$r(\mathbf{X}^t) = f(\mathbf{X}^*) - f(\mathbf{X}^t) \tag{4.25}$$

The cumulative regret is the sum of the regret over all iterations, and the goal of Bayesian optimization can be translated as minimizing the cumulative regret

$$r_T = \sum_{t=1}^{T} r(\mathbf{X}^t) \tag{4.26}$$

This can be also viewed as a multi-armed bandit task, a problem where there are multiple options (called *arms*) with an known probability of producing a certain reward and the objective is to maximize the overall reward (the name comes from the one armed bandit slot machines that can be found in casinos). In our case, the different inputs can be represented as the arms of the bandits and the corresponding output at these points is the unknown reward associated to each arm. The main difference with a classical multi-armed bandits problem comes from the fact that two rewards are correlated through the underlying kernel. Yet, this point of view allows to use approaches that were developed for bandits and exploit them in the Bayesian optimization framework.

The *Gaussian process upper confidence bound* (GP-UCB) strategy relies on the following acquisition function [Sri+10]

$$a^{\mathrm{UCB}}(\mathbf{X}) = -\mu(\mathbf{X}) + \kappa_t s(\mathbf{X}) \tag{4.27}$$

where $\kappa_t = \sqrt{\nu\tau_t}$ is the trade-off parameter.

This strategy chooses the arm for which the upper confidence bound is currently the highest. Its value depends, at a given $\mathbf{X}$ on the mean μ of the model (the higher the mean, the lower the bound) and the uncertainty at that location (the higher the uncertainty, the higher the bound). Hence, this acquisition function encompasses a natural trade-off between exploitation and exploration as the expected improvement.

Furthermore, for $\nu = 1$ and $\tau_t = 2\log(t^{d/2+2}\pi^2/3\delta)$, with $\delta \in (0, 1)$, [Sri+10] shows that this method has no regret, thus $\lim_{T\longrightarrow\infty} r_T/T = 0$. This only holds when the kernel functions is reasonably smooth, which is the case for most of the aforementioned kernels.

All acquisition functions define valid exploration-intensification trade-offs to search the domain space for the optimum, even though the probability of improvement tends to exploit the model aggressively and prematurely converge to local solutions, see Figure 4.5. Instead of relying on a single acquisition function, [HBF11] defines a portfolio with multiple acquisition functions that each provides a different candidate query input and also a criterion to select the next query point based on the different candidates.

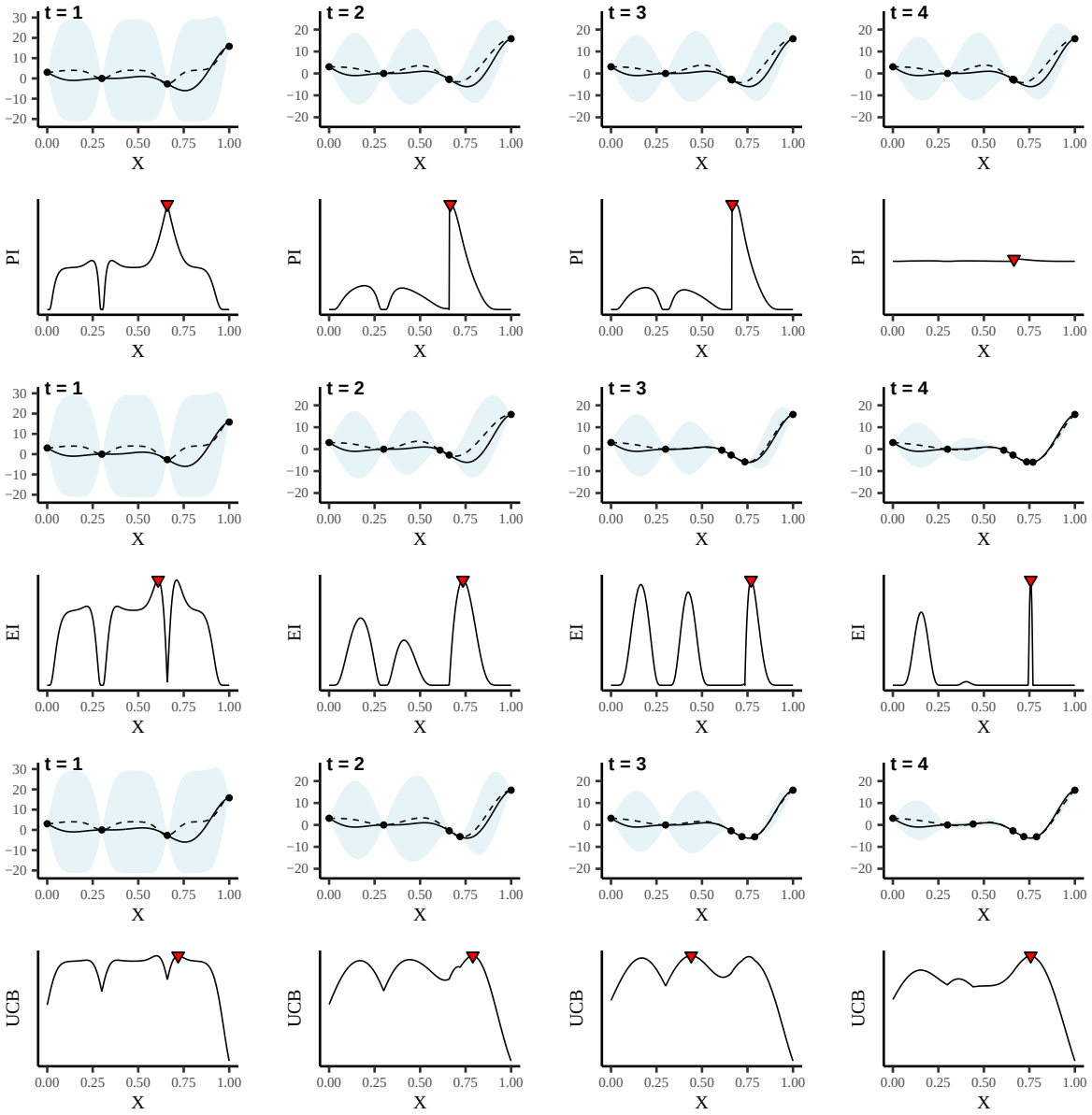

Figure 4.5 – Comparison of different acquisition functions on the Forrester function Equation (4.17): probability of improvement (top), expected improvement (middle) and upper confidence bound (bottom). Upper rows show the objective function as a black line and the Gaussian process mean as a dashed line. The lightblue areas are the Gaussian process posterior confidence intervals. Lower rows depict the respective acquisition functions with their current maximum. The optimization is initialized with the same four points but the different approaches visit different locations. In particular, the probability of improvement seems to stay stuck around the current minimum, while the behaviors with both the expected improvement and the upper confidence bound are similar for these iterations.

Optimizing the acquisition function

Bayesian optimization substitutes the optimization of an expensive function with the repeated maximization of the acquisition function, which comes at a cheaper cost. Nevertheless, the acquisition function is often multimodal, as shown in Figure 4.5, and optimizing it is a non-trivial problem.

Different strategies were proposed in the literature to optimize the acquisition functions, such as adaptive grid [BK10], direct rectangles approach (DIRECT) [JPS93], or when gradients of the acquisition functions are available, or can be approximated, one can use quasi-Newton local optimization (as the L-BFGS method [LN89]) with restarts. Finally, [Ber+11] relies on the CMA-ES algorithm [HO01], a gradient-free evolutionary algorithm for optimization on continuous domains.

As described in [Sha+15], optimizing multimodal acquisition functions can turn out to be problematic as the true optimum can be missed and assessing the quality of the solution found is difficult. This directly raises some concerns about the convergence of Bayesian optimization as theoretical bounds assume that the exact optimizer is found and chosen at each iteration.

Furthermore, the biggest issues with Bayesian optimization come from the induced limitations due to the dimension of the problem considered. We detail them in the following section.

4.2 High-dimensional issues

Bayesian optimization has been frequently and quite successfully applied in particular in engineering, but the applications were restricted to low to moderate dimensional problems (up to $d = 10$ typically). Up-scaling Bayesian optimization with dimension is a threefold problem.

The first issue relates to the exponential growth of the search volume hence the required number of evaluations to ensure a good coverage of $\mathcal{X}$ (commonly known as the *curse of dimensionality*). This is a common problem for any optimizer notably DIRECT.

Secondly, despite being continuous, the acquisition function is mostly flat with sharp local minima, and in that case, the optimizers commonly used (such as restarted BFGS) can become inefficient and the global optimum impossible to achieve.

Finally, as reminded in [KSP15], Gaussian processes, as a non-parametric regression model, have a time to convergence that grows exponentially with the dimension d.

Different methods in the literature tackled those issues separately when dealing with Bayesian optimization in high dimensions and can be sorted out depending on the assumptions made:

- approaches that assume the structure of the function or the surrogate model to facilitate optimization,
- approaches that assume a low intrinsic dimension of the problem, thus a possible model reduction leading to an easier problem to solve.

We introduce some of these approaches in the next sections.

4.2.1 Assumptions about the structure of the model

In order to deal with the dimensionality issues, a common approach is to assume that the function f can be decomposed as a sum of m functions of smaller, disjointed groups of dimensions [Has17]. By doing so, the optimization can be conducted in each group separately instead of working in the high dimension of the original problem. In an early work, [DNR11] introduced additive Gaussian processes with additive kernels and assumed a sum of functions of all combinations of lower dimensional coordinates. From their experiments, the additive structure is most often recoverable in data sets and their model approximates well the data even when the main assumption is not verified. Later on, [KSP15] proposed the group-additive Gaussian processes with an assumption of independence between the different groups. For simplicity purpose, we use *additive* in place of *group-additive* in the following.

Assume that the function f can be decomposed into an additive structure of m sub-functions of the form

$$f(\mathbf{X}) = f^{(1)}(\mathbf{X}^{(1)}) + \cdots + f^{(m)}(\mathbf{X}^{(m)}) \tag{4.28}$$

where $\mathbf{X}^{(i)} \in \mathcal{X}^{(i)}$ are disjointed lower dimensional components of $\mathbf{X}$, such that $\bigcup_{i=1}^{m} \mathcal{X}^{(i)} = \mathcal{X}$. The disjointed aspect of it means that we can write that $\mathbf{X}^{(i)} \cap \mathbf{X}^{(j)} = \emptyset$, for all i, j, with $i \neq j$.

We still consider the Bayesian paradigm and a Gaussian process prior $Y(\mathbf{X})$ over f with a zero mean and a given covariance kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Given the additive structure on f, $Y(\mathbf{X})$ also has an additive kernel $k(\mathbf{X}, \mathbf{X}') = \sum_i k(\mathbf{X}^{(i)}, \mathbf{X}^{(i)'})$. If $\mu^{(i)} = 0$ for all i, its mean can be also decomposed as $\mu(\mathbf{X}) = \mu^{(1)}(\mathbf{X}^{(1)}) + \cdots + \mu^{(m)}(\mathbf{X}^{(m)})$. It is also assumed that each group of observations $f^{(i)}$ is sampled from a Gaussian process $Y^{(i)}(\mathbf{X}^{(i)})$ with a kernel $k^{(i)} : \mathcal{X}^{(i)} \times \mathcal{X}^{(i)} \to \mathbb{R}$. A kernel $k^{(i)}$ which only involves i variables is called a i-th order kernel when k which considers all variables is a d-th order kernel.

As previously, considering past observations $(\mathbb{X}, \mathbb{Y})$, we can infer the posterior distribution at any new point $\mathbf{X}$. In the additive case, we are primarily interested in the distribution of the different sub-functions $f^{(i)}$ given the observations:

$$\begin{bmatrix} Y(\mathbb{X}) \\ Y^{(i)}(\mathbf{X}^{(i)}) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(\mathbb{X}, \mathbb{X}) & k^{(i)}(\mathbb{X}^{(i)}, \mathbf{X}^{(i)}) \\ k^{(i)}(\mathbf{X}^{(i)}, \mathbb{X}^{(i)}) & k^{(i)}(\mathbf{X}^{(i)}, \mathbf{X}^{(i)}) \end{bmatrix}\right)$$

and once again obtain the posterior distribution $\hat{F}^{(i)}$ considering the observations with mean

$$\mu^{(i)}(\mathbf{X}^{(i)}) = (k^{(i)}(\mathbf{X}^{(i)}, \mathbb{X}^{(i)})k(\mathbb{X}, \mathbb{X})^{-1}\mathbb{Y} \tag{4.29}$$

and variance

$$s^{2(i)}(\mathbf{X}^{(i)}) = k^{(i)}(\mathbf{X}^{(i)}, \mathbf{X}^{(i)}) - k^{(i)}(\mathbf{X}^{(i)}, \mathbb{X}^{(i)})k(\mathbb{X}, \mathbb{X})^{-1}k^{(i)}(\mathbb{X}, \mathbf{X}^{(i)}) \tag{4.30}$$

Using the inferred posterior distribution $\hat{F}^{(i)}$, [KSP15] defines the additive Gaussian upper confidence bound (ADD-GP-UCB), an alternative to the upper confidence bound which applies to an additive kernel, as

$$a^{\mathrm{UCB},(i)}(\mathbf{X}) = -\mu(\mathbf{X}) + \kappa_t \sum_{i=1}^{m} s^{(i)}(\mathbf{X}^{(i)}) \tag{4.31}$$

$$= \sum_{i=1}^{m} -\mu^{(i)}(\mathbf{X}^{(i)}) + \kappa_t s^{(i)}(\mathbf{X}^{(i)})$$

This acquisition function is written as a sum of functions defined on orthogonal domains, meaning that $a^{\mathrm{UCB},(i)}$ can be optimized by optimizing each function separately on $\mathcal{X}^{(i)}$. Similarly to the UCB acquisition function, the authors are able to bound the regret for specific kernels [KSP15].

Obtaining the posterior for each $Y^{(i)}$ instead of only for Y does not induce additional computations as the expensive task comes from the inversion of $K(\mathbb{X}, \mathbb{X})$ which only has to be done once for each method (and can be reused m times in the additive formulation). However, the optimization of the acquisition function is much simpler and favorable in the additive setting than with the classical UCB function.

As expected, if f is additive with a known decomposition, it can be used directly but most often it is not the case as we work in a black-box setting with no information about the function f. Hence, the decomposition can be treated as an hyperparameter of the additive kernel and maximize the likelihood with respect to the decomposition. As estimating the likelihood for all possible decompositions is too burdensome, [KSP15] proposes to randomly select a few decompositions and choose the one with the largest marginal likelihood. In order to make their algorithm more efficient, learning the decomposition is only done every N_{cyc} iterations.

More recently, [Gar+17] and [Wan+17] have relied on different sampling procedures to efficiently learn the decomposition but they still require a considerable computational effort which limits the applicability to objective functions with a high evaluation cost.

4.2.2 Assumptions about the effective dimension of the model

The second main approach with high dimensional problems considers that the function f has an effective dimension d_e, such that $d_e \ll d$ and relies on a mapping between the high dimensional space and an unknown low dimensional subspace.

REMBO

Following [Wan+17], let there be a linear effective subspace $\mathcal{T}$ of dimension d_e such that for all $\mathbf{X}_\top \in \mathcal{T} \subset \mathcal{X}$ and $\mathbf{X}_\perp \in \mathcal{T}^\perp \subset \mathcal{X}$. $\mathcal{T}^\perp$ is the orthogonal complement of $\mathcal{T}$, called the constant subspace. A function has an effective dimension d_e if it can be defined as $f(\mathbf{X}) = f(\mathbf{X}_\top + \mathbf{X}_\perp) = f(\mathbf{X}_\top)$. The name for $\mathcal{T}^\perp$ implies that the function does not change along the coordinates $\mathbf{X}_\perp$.

Following their first theorem, problems with low effective dimensionality can be solved with the use of random embeddings. Assume we have a function $f : \mathbb{R}^d \mapsto \mathbb{R}$ with an effective dimension d_e, and a random matrix $\mathbf{A} \in \mathbb{R}^{d \times \delta}$ with independent entries sampled from $\mathcal{N}(0, 1)$ and $\delta \geq d_e$. It follows that, with probability 1, for any $\mathbf{X} \in \mathbb{R}^d$, there is a $\mathbf{Z} \in \mathbb{R}^\delta$ such that $f(\mathbf{X}) = f(\mathbf{AZ})$.

It means that given any $\mathbf{X} \in \mathbb{R}^d$ and a random matrix $\mathbf{A} \in \mathbb{R}^{d \times \delta}$, with probability 1, there exists a point $\mathbf{Z} \in \mathbb{R}^\delta$ such that $f(\mathbf{X}) = f(\mathbf{AZ})$. This implies that for any optimum $\mathbf{X}_* \in \mathbb{R}^d$, there is a corresponding $\mathbf{Z}_* \in \mathbb{R}^\delta$ and that $f(\mathbf{X}_*) = f(\mathbf{AZ}_*)$. Hence, instead of conducting an optimization in the high dimensional space, we can find the optimum solution for $g(\mathbf{Z}) = f(\mathbf{AZ})$ in the lower dimensional space.

Following this observation, the authors proposed an algorithm called Bayesian optimization

with random embeddings (REMBO) which first draws a random embedding (given by $\mathbf{A}$) and then conducts the optimization in the low dimensional embedded subspace. This method has shown some good efficiency compared to other methods since no initial budget is dedicated to learning the structure of the function, even when the initial assumption of low dimensionality was not satisfied. Yet, several issues exist, mostly connected to the choice of the embedded subspace. If it is too small, the optimum $\mathbf{Z}_*$ might not belong to it whereas taking it too large might take us back to our initial problem.

Split and Doubt

A different method which does not require to specify any effective dimension value was proposed by [BS+19] under the name of *Split and Doubt*. The authors define a two-steps approach which first learns the set of influential variables through a popular heuristic, the Automatic Relevance Determination (ARD) [WR06], which classifies dimensions with large correlation lengths as non-influential. The authors state that a small correlation length corresponds to an input that has an important impact on the objective function. On the contrary, when the input has no influence, the correlation length should go to infinity. The authors provide a link between correlation lengths and variable importance based on the Derivative-based global sensitivity measures (DGSM, previously introduced in Section 2.1.2) by showing that when a correlation length goes to 0, or to infinity, the DGSM of the predictor mean of the Gaussian process for this input tends to its maximal or minimal value, respectively.

In the split and doubt scheme, a first GP is built and a first optimization is carried out in the subspace of influential variables. Then this optimization is challenged in the doubt step by working only in the subspace of *non*-influential variables: a second GP is built in this subspace with an incentive at making non-influential variables influential (decreasing their length-scale); the point where the 2 GP predictions differ the most will also be added to the DoE at the next iteration. The main motivation behind the doubt step comes from possible inaccurate estimations of the correlation length which might drastically impact the optimization results. Questioning the selection at each iteration allows to avoid the premature classification of an important variable as non-influential.

DropOut

Motivated by the dropout method in neural networks (see [Sri+14] for a brief explanation of the random deactivation of neurons in the network to avoid overfitting), [Li+17] explores the use of the dimension dropout in Bayesian optimization.

Let I_{d_e} be the indices of the randomly selected dimensions, with $\text{card}(I_{d_e}) = d_e$, and $\overline{I_{d_e}}$ the dropped out ones. By definition, $I_{d_e} \cap \overline{I_{d_e}} = \emptyset$ and $I_{d_e} \cup \overline{I_{d_e}} = \{1, \dots, d\}$. Corresponding variables are $\mathbf{X}_{I_{d_e}}$ and $\mathbf{X}_{\overline{I_{d_e}}}$ that we later write $\mathbf{X}_e$ and $\mathbf{X}_{\overline{e}}$ for convenience.

Consider a Gaussian prior distribution on the function $f(\mathbf{X}_e \mid \mathbf{X}_{\overline{e}})$, where the choice of $\mathbf{X}_{\overline{e}}$ is discussed hereafter. Using previous results, a predictive mean $\mu(\mathbf{X}_e)$ and a predictive variance $s(\mathbf{X}_e)$ can be estimated from past observations. The authors resort to the d_e-dimensional upper confidence bound acquisition function Equation (4.27):

$$a^{\text{UCB}}(\mathbf{X}_e) = -\mu(\mathbf{X}_e) + \kappa s(\mathbf{X}_e). \tag{4.32}$$

Doing so, at each iteration of the optimization, new values for $\mathbf{X}_e$ are obtained by minimizing the

acquisition function. The dimension reduction is used here only for a more efficient optimization of the function acquisition.

Different strategies are provided for the dropped out dimensions, namely:

1. *Dropout-Random*: randomly draw in the domain at each iteration, $\mathbf{X}_{\overline{e}} \sim \mathcal{U}(\mathcal{X}_{\overline{e}})$.
2. *Dropout-Copy*: use the observations giving the best function value so far $\mathbf{X}^{+,t} = \arg\min_{t' \leq t} f(\mathbf{X}^{t'})$, $\mathbf{X}_{\overline{e}} = \mathbf{X}_{\overline{e}}^{+,t}$.
3. *Dropout-Mix*: use a mixture of both above methods. For each component independently, choose a random value with probability p or copy the component of the best-so-far solution with probability $1 - p$.

The algorithm is summarized in Algorithm 3.

Algorithm 3 Dropout algorithm for high-dimensional Bayesian optimization

Require: $\mathbb{X} = (\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n)$, $\mathbb{Y} = f(\mathbb{X})$; Acquisition function $a(\cdot)$; GP prior for f with mean function μ and kernel k; budget
  for $t = 1, 2, \ldots$ budget **do**
    Randomly select d_e dimensions
    Choose $\mathbf{X}_e^t = \arg\max_{\mathbf{X}_e \in \mathcal{X}_e} a(\mathbf{X}_e)$
    Define $\mathbf{X}_{\overline{e}}^t$ using one of the three fill-in strategies (Section 4.2.2)
    $\mathbf{X}^t = \mathbf{X}_e^t \cup \mathbf{X}_{\overline{e}}^t$
    Calculate $f(\mathbf{X}^t)$
    Augment data and update the GP statistical model
  end for

Intuitively, the Dropout-Random is interesting when away from the global optimum as we do not have any information about the location of the minimum value, hence random guesses are appropriate. The Dropout-Copy should be preferred to a random choice if the best-so-far point is close to the true minimum of the function, but it is associated to a risk of premature convergence on some components. In [Li+17], the Dropout-Mix gives the best results as it allows to avoid staying in a local optimum for too long. These different methods were tested based on the cumulative regret bound they yield which depends on the choice of the number of dropped out dimensions. A judicious number for d_e will improve the bound.

A fourth strategy was suggested in [SLRDV19], denoted as *Dropout-Gauss* where the overall idea is to sample values for the dropped out dimensions based on the best λ known points $\mathbb{X}$, more precisely, along a multivariate Gaussian distribution based on the λ best observation points, $\lambda = n/2$, defined by $\mathbf{X}_{\overline{e}} \sim \mathcal{N}(\mu_\lambda, \Sigma_\lambda)$, where

$$\mu_\lambda = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \mathbb{X}_{\overline{e}}^{i:n},$$

$$\Sigma_\lambda = \frac{1}{\lambda - 1} \sum_{i=1}^{\lambda} (\mathbb{X}_{\overline{e}}^{i:n} - \mu_\lambda)(\mathbb{X}_{\overline{e}}^{i:n} - \mu_\lambda)^T.$$

$\mathbb{X}_{\overline{e}}^{i:N}$ denotes the observed points ranked from best to worst. However, this strategy fits a Gaussian distribution, which might not be adapted when multiple local optima are represented

in the λ observations.

The number of dropped out dimensions d_e is central in this method. [Li+17] investigated the influence of this value on two test functions defined for $d = 20$. First, a Gaussian mixture function defined as

$$f(\mathbf{X}) = \phi(\mathbf{X}, \mu_1, \Sigma_1) + \frac{1}{2}\phi(\mathbf{X}, \mu_2, \Sigma_2)$$

with ϕ the Gaussian probability function, $\mu_1 = [2, \dots, 2]$, $\mu_2 = [3, \dots, 3]$ and $\Sigma_1 = \Sigma_2$ is the identity matrix of dimension d. This function shows no interaction and a local maximum. The second function is the unimodal Schwefel's 1.2 function defined as

$$f(\mathbf{X}) = \sum_{i=1}^{d}(\sum_{j=1}^{i} X_j)^2.$$

Varying the number of dropped out variables led to the following conclusions. First, when there is no interacting variable, each dimension can be optimized separately and smaller values for d_e lead to good performance. However, for the second test function which has interactions, optimizing independently is not efficient. For such functions, a larger d_e leads to a faster convergence rate because it provides a higher probability of optimizing interacting variables. Their results also show that picking d_e too large makes the optimization unnecessarily expensive, whereas if d_e is too small the convergence rate is slow for functions with interacting variables. Hence, the authors suggest a compromise between these two extremes and experimentally obtain $d_e = 2$ for $d = 5$ and $d_e = 5$ for d in the order of ten.

Their experiments highlight how the Dropout-Mix strategy behaves the best for most of their test cases. It can be seen as a direct trade-off between the other two strategies since depending on the value chosen for p we can obtain the Random strategy (for $p = 1$) or the Copy strategy (for $p = 0$). Testing different configurations for p on the same examples as above, they observe that for low dimensional problems, e.g. $d = 2$ with $d_e = 1$, strategies with $p \geq 0.5$ perform better, due to the fact that the Copy strategy can get stuck in local optima. This phenomenon happens less often in higher dimensions as it has a lower probability of occurring, which leads to a good average performance of the Copy approach. Thus, in high dimensional problems, the authors suggest to rely on the Copy strategy or on the Mix approach with a small p (e.g. lower than 0.2) which can be seen as a relaxation of the Copy strategy and could avoid getting stuck in local optima.

As the authors noted in their conclusion, the main drawback of the method is the fully random aspect of the variable dropout. In order to tackle this limitation, selection of the active dimensions can be guided by sensitivity analysis, which is a classical approach. We have already shown the benefit of doing so prior to the optimization in Section 3.3.3 and this strategy can also be found in the literature, for example in [SW10] which relies on Sobol indices to reduce the dimension of the problem. In [Ulm+16], the authors weight the random selection of the dimensions using a Principal Component Analysis (PCA) and sample proportionally to the eigenvalue magnitude of the inputs. Yet, such method requires large sample size to provide a correct computation of the PCA and might be not well-suited for an optimization purpose, like variance-based sensitivity analysis strategies, as exposed previously. We therefore now investigate the use of kernel-based sensitivities for the selection of important variables.

4.3 Coupling KSA with GP-based optimization

4.3.1 Strategies

The initial problem was to minimize a high-dimensional expensive black-box function f. Relying on a surrogate-based optimization approach allows to save calls to the true objective function. The surrogate model will be a Gaussian process, thus making the optimization algorithm Bayesian. In this section, we detail our strategy to improve Bayesian optimization.

As a brief reminder, we use the kernel-based sensitivity indices defined in Section 2.2 within the Bayesian optimization algorithm presented in Section 3.2. More precisely, we rely on the Hilbert-Schmidt independence criterion to characterize the relevance of the different dimensions in order to obtain good observations of the function. The Hilbert-Schmidt independence criterion used in a goal-oriented sensitivity analysis setting quantifies the dependence between two random variables. In our context, these variables are a given component, $\mathbb{X}_i$, and the output modified by an indicator-thresholding as described in Section 3.1, $\mathbb{Z} = 1_{\mathbb{Y} \leq q_\alpha}$

$$\text{HSIC}(\mathbb{X}_i, \mathbb{Z}) = \frac{1}{n^2}\text{tr}(KHLH)$$

Such HSIC estimation only involves Gram matrices over the dimension i.

In the following, we will consider the equivalent formulation that uses the maximum mean discrepancy of Equation (3.9)

$$\text{HSIC}(X_i, Z) \propto \gamma^2(P_{X_i|Z=1}, P_{X_i}) = \gamma^2(P_{X_i|\mathbf{X}\in\mathcal{D}_{q_\alpha}}, P_{X_i}) \tag{4.33}$$

where $\mathcal{D}_{q_\alpha}$ is what we define as the α level set of interest for our objective function, corresponding to the area of the design space $\mathcal{X}$ that produces better observations than a given threshold q_α (that will be defined).

The variable selection approach is similar to that in Section 3.2 but there are differences induced by the GP when coupling the kernel-based sensitivity analysis within the now Bayesian Dropout algorithm: the definition of the sublevel set of interest is directly done using the surrogate model; new strategies are considered for the selection relying on the sensitivity indices calculated with the MMD measures; and different methods are proposed to set the left out variables. We describe these new features in the following.

First of all, the threshold q_α was previously defined as a quantile of the true function: it is now computed directly on the mean of the conditioned Gaussian process, $\hat{q}_\alpha = F^{-1}_{\mu(\mathbf{X})}(\alpha)$, where $\mu(\mathbf{X})$ is directly obtained following Equation (4.15). Thus, we obtain the level set of interest on the surrogate model as $\hat{\mathcal{D}}_{\hat{q}_\alpha} = \{\mathbf{X} \in \mathcal{X}, \mu(\mathbf{X}) \leq \hat{q}_\alpha\}$. Figure 4.6 shows the difference between $\mathcal{D}_{q_\alpha}$ and $\hat{\mathcal{D}}_{\hat{q}_\alpha}$ at the level $\alpha = 10\%$ on the Dixon-Price function Equation (3.2). For this example, a design of 16 observations was generated using a latin hypercube sampling and a squared exponential kernel was used for the Gaussian process prior.

Once the level set $\hat{\mathcal{D}}_{\hat{q}_\alpha}$ is defined, the estimation of the sensitivities for each dimension is directly given by

$$S^\gamma(X_i) = \gamma^2(P_{X_i|\mathbf{X}\in\hat{\mathcal{D}}_{\hat{q}_\alpha}}, P_{X_i}). \tag{4.34}$$

Let $\mathbb{X}_i$ be a sample of size n, we can rely on the unbiased estimator from Equation (2.32) and

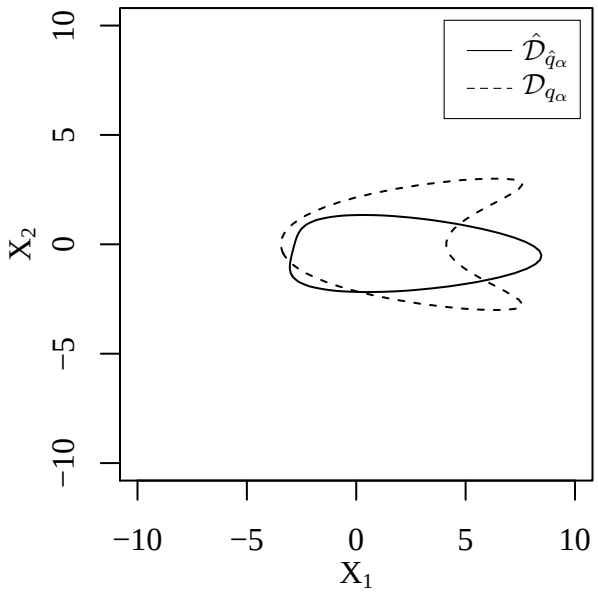

Figure 4.6 – Difference between the level sets computed on the true function and on a surrogate model for the Dixon-Price function Equation (3.2). Both are estimated at the level $\alpha = 10\%$.

compute the MMD as

$$S^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_i) = \gamma_u^2(\mathbb{X}_i, \tilde{\mathbb{X}}_i) = \frac{1}{m(m-1)} \sum_{p=1}^{m} \sum_{q \neq p}^{m} k(\mathbb{X}_i^p, \mathbb{X}_i^q) + \frac{1}{n(n-1)} \sum_{i=p}^{n} \sum_{q \neq p}^{n} k(\tilde{\mathbb{X}}_i^p, \tilde{\mathbb{X}}_i^q) - \frac{2}{mn} \sum_{p=1}^{m} \sum_{q=1}^{n} k(\mathbb{X}_i^p, \tilde{\mathbb{X}}_i^q) \tag{4.35}$$

with $\tilde{\mathbb{X}}_i$ being the sub-sample of size m (where m directly depends of n and the level α chosen for the quantile $\hat{q}_\alpha$) of $\mathbb{X}_i$ that belongs to $\hat{\mathcal{D}}_{\hat{q}_\alpha}$. The expression of Equation (4.35) involves the computation of two main terms ($k(\mathbb{X}_i^p, \mathbb{X}_i^q)$ and $k(\tilde{\mathbb{X}}_i^p, \tilde{\mathbb{X}}_i^q)$) and a cross term ($k(\mathbb{X}_i^p, \tilde{\mathbb{X}}_i^q)$). Since $\tilde{\mathbb{X}}_i$ is directly extracted from $\mathbb{X}_i$, it is possible, and computationally more efficient, to obtain the second main term and the cross term from $k(\mathbb{X}_i, \mathbb{X}_i)$. Furthermore, as the Gram matrices $k(\mathbb{X}_i, \mathbb{X}_i)$ are symmetrical, the number of required operations can also be lowered. Only half of each Gram matrix $k(\mathbb{X}_i, \mathbb{X}_i)$ is stored and the indices are computed by only extracting the proper term from it using C++ routines.

The different indices are simply normalized by

$$\hat{S}^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_i) = \frac{S^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_i)}{\sum_{j=1}^{d} S^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_j)}. \tag{4.36}$$

This allows us to be able to compare one value of index with another for varying X_i's.

The last important aspect is to select the dimensions once we have computed the associated normalized sensitivities(Equation (4.35)). We propose two strategies:

- The *Probabilistic Strategy:* $d_e < d$ dimensions are drawn at random with a probability equal to the corresponding sensitivity index $\hat{S}^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_i)$. The heuristic parameter d_e is set to

5, following the recommendation from [Li+17], since it is a compromise between a large d_e (which implies an expensive optimization) and a small d_e (which leads to a slower convergence especially when the function f has many interacting variables).

- The *Deterministic Strategy:* All dimensions whose index $\hat{S}^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_i)$ is above a given threshold τ are kept. We set $\tau = 1/d$ as it corresponds to the value we would obtain with equal normalized sensitivity indices.

Both methods favor variables with a high sensitivity index, hence those detected as important to reach locations where the predictive mean of the Gaussian Process is low, assuming the surrogate model is a good representation of the objective function.

The main difference lies in the number of variables kept as the probabilistic method activates a constant number of variables, d_e, whereas the deterministic approach activates a varying number of variables. Unlike the deterministic strategy, the probabilistic method can draw variables with almost-zero sensitivity indices.

Because all groups of variables have a non-zero probability of becoming active in the long run, the probabilistic strategy, when coupled with a global optimization algorithm, is globally convergent. On the contrary, the deterministic approach may fail to accurately converge to the optimum on functions for which some variables always have $\hat{S}^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_i)$ smaller than the selection threshold τ (e.g., a quadratic function with a high aspect ratio).

For the dropped out dimensions, we rely on the different fill-in strategies introduced by [Li+17] and recalled in Section 4.2.2. A complete overview of the method is presented in Algorithm 4.

Algorithm 4 Bayesian optimization with Dropout guided by kernel-based sensitivity indices

Require: $\mathbb{X} = (\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n)$, $\mathbb{Y} = f(\mathbb{X})$, Acquisition function $a(\cdot)$, GP prior for f with mean function m and kernel k, α, budget
  for $t = 1, 2, \ldots,$ budget **do**
    for $i = 1, 2, \ldots, d$ **do**
      $\hat{q}_\alpha = F^{-1}_{\mu(\mathbf{X})}(\alpha) \leftarrow \alpha$th quantile of the GP mean
      $\mathbb{X}_i \leftarrow (X_i^1, X_i^2, \ldots, X_i^n)$
      $K \leftarrow k(X_i^p, X_i^r)$ assembly of the Gram matrix, $p, r = 1, \ldots, n$
      Compute $\hat{S}^{\gamma}_{\hat{q}_\alpha}(\mathbb{X}_i)$ following Equation (4.36)
    end for
    Select d_e dimensions using the Probabilistic or the Deterministic strategy
    Calculate $\mathbf{X}_e^t = \arg\max_{\mathbf{X}_e \in \mathcal{X}_e} a(\mathbf{X}_e)$
    Define $\mathbf{X}_{\bar{e}}^t$ using one of the four fill-in strategies (Section 4.2.2)
    $\mathbf{X}^t = \mathbf{X}_e^t \cup \mathbf{X}_{\bar{e}}^t$
    Calculate $f(\mathbf{X}^t)$
    Augment data and update the GP statistical model
  end for

The main difference with the method presented in Chapter 3 lies in the fact that the selection is not done before the complete optimization process but within the Bayesian optimization procedure, allowing to possibly select all the variables.

4.3.2 Numerical tests

The Dropout algorithm with kernel-based sensitivities, Algorithm 4, is applied to a couple of functions. We test the different selection and fill-in strategies and compare them to a full Bayesian optimization (in all dimensions) and a classical Dropout as introduced by [Li+17].

The considered functions are the classical analytical Rosenbrock and Branin functions. We chose these functions for their diverse properties: the Rosenbrock function has an easy to find optimum valley but converging to the true minimum is difficult as the bottom of the valley is flat and curved, while the Branin function is a multimodal function with 3 known optima. Each function is defined in $\mathbb{R}^{d_{\text{eff}}}$ and $d - d_{\text{eff}}$ dummy variables are added to the problem to increase the dimensionality of the problem to d. The different characteristics of each problem are summarized in Table 4.1.

Table 4.1 – Test functions descriptions. d_{eff} is the effective dimension while d corresponds to the embedded high dimension obtained by adding dummy variables defined on $[0, 1]$.

Name	d_{eff}	d	Domain	Expression
Branin	2	25	$[-5, 10] \times [0, 15]$	$f(\mathbf{X}) = \left(X_2 - \frac{5.1}{4\pi^2}X_1^2 + \frac{5}{\pi}X_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(X_1) + 10$
Rosenbrock	5	20	$[-3, 3]^5$	$f(\mathbf{X}) = \sum_{i=1}^{d-1} 100\left(X_{i+1} - X_i^2\right)^2 + (X_i - 1)^2$

We consider the Probabilistic and Deterministic Strategies combined with the four different fill-in methods. Both methods require an hard-coded parameter: the number of variables kept by the Probabilistic Strategy is set to $d_e = 5$ for the Rosenbrock function and $d_e = 2$ for the Branin function; τ is equal to $1/d$ for the threshold of detection for the Deterministic Strategy.

The kernel-based sensitivities are computed using Equation (4.36) with $\alpha = 5\%$. The initial design of experiments is a latin hypercube sampling optimized with respect to the maximin criterion, which is a classical choice for the initialization of the Gaussian process. The DOE has size 40 for the Rosenbrock function and 30 for the Branin function. Their sizes are voluntarily small compared to the dimension of the problem, because the function f is assumed to be expensive and calls to it are limited. The rule of thumb is usually to consider a DOE of size $2 - 10$ times the dimension of the problem to ensure that the surrogate model has a sufficient accuracy. As the final results depend on the initial DOE, the runs are repeated 20 times for each configuration of the optimizer with different initial DOEs. Yet, for consistency in the comparison of the results, all versions start with the same DOE in each run. The Gaussian process is created with the package DiceKriging in the R language and we use a Matèrn 5/2 kernel. The Expected improvement (EI) Equation (4.22) is the acquisition function and it is optimized with the CMA-ES algorithm [HO01]. The optimization budget is 100 calls to the objective function f and results are compared after reaching this limit.

Before conducting any comparison between algorithms performance, we test the variable selection. Both Deterministic and Probabilistic selections are able to efficiently pick out determining variables over the iterations. Figures 4.7 and 4.9 show the cumulative selection of occurrence for each variable while Figures 4.8 and 4.10 show the average rate of selection for each variable. The dummy variables are kept at a low rate, despite having zero influence on the performance of the objective function. This is mostly due to the approximation errors of the surrogate model. Since the number of variables kept at each iteration in the Probabilistic strategy is set to 5, when a dummy variable is selected, it automatically means than a true variable was dropped

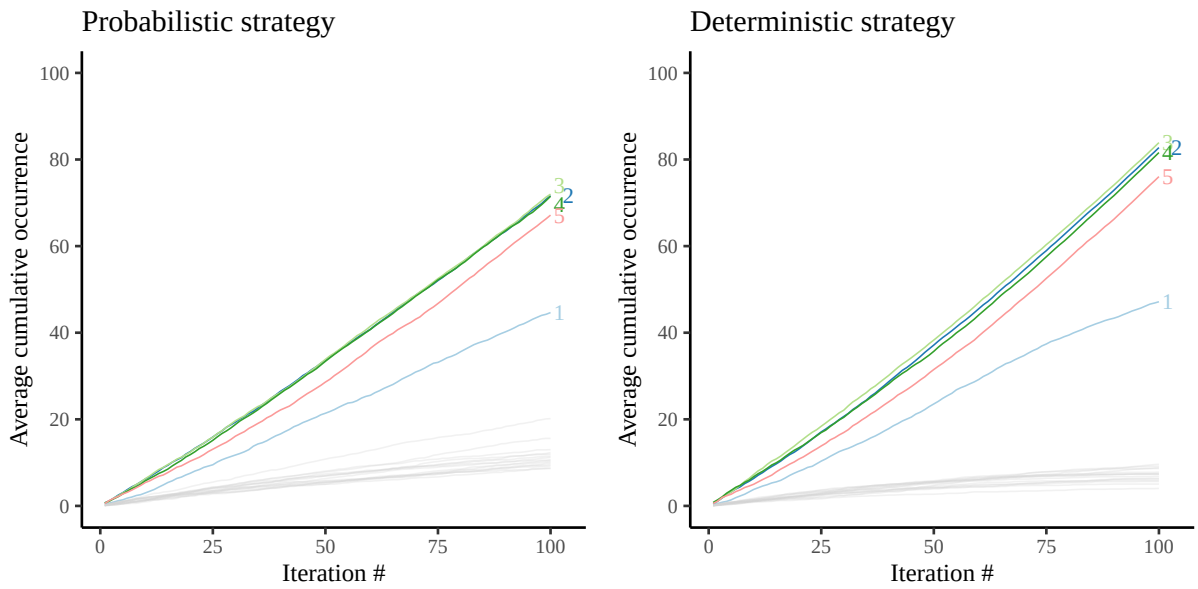

Figure 4.7 – Average cumulative selection for each variable for the Probabilistic and Deterministic strategy with a Mix fill-in approach for the Rosenbrock-20d function. The top 5 curves of each subplot correspond to the first five variables (the non-dummy ones).

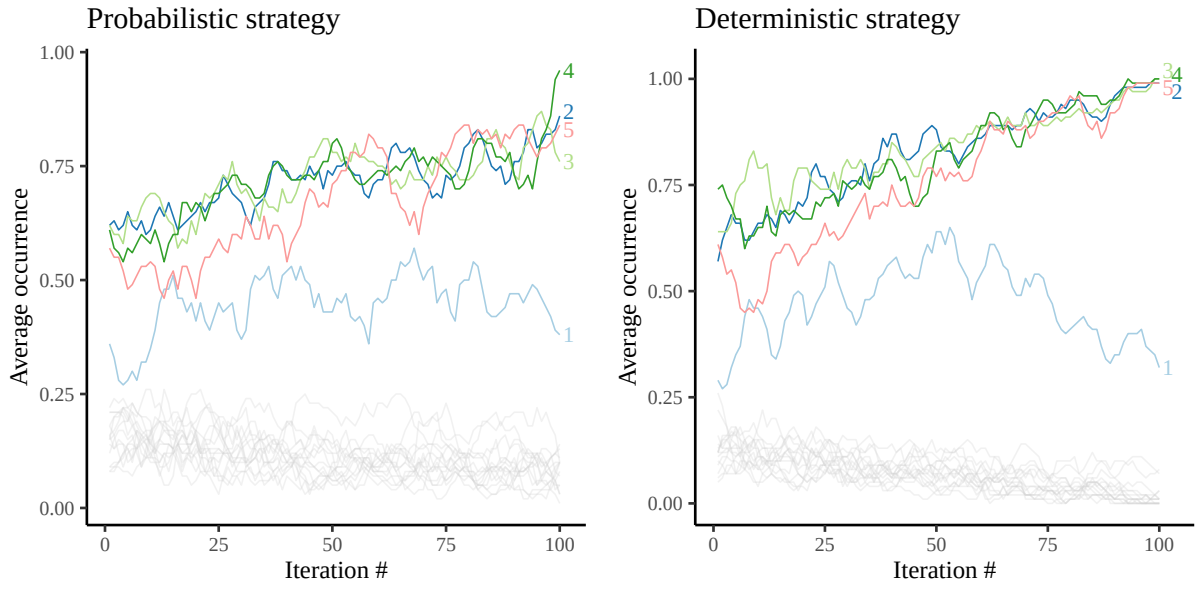

Figure 4.8 – Average selection of occurrence for each variable for the Probabilistic and Deterministic strategy with a Mix fill-in approach for the Rosenbrock-20d function. The results are smoothed using a moving average with a 5 iterations window size. The top 5 curves of each subplot correspond to the first five variables (the non-dummy ones).

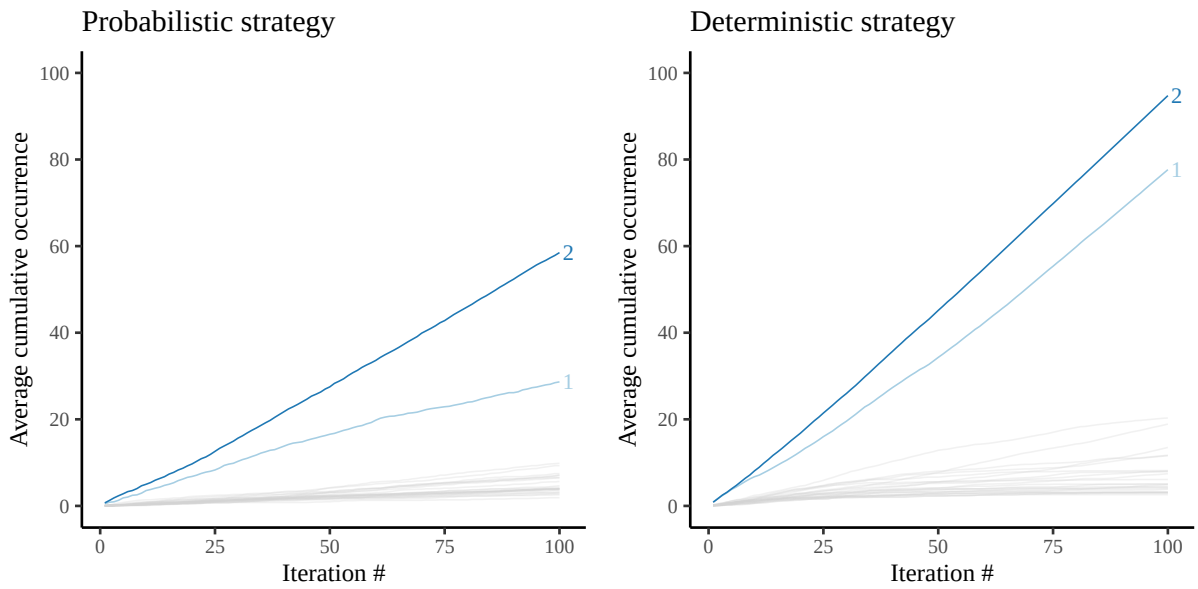

Figure 4.9 – Average cumulative selection for each variable for the Probabilistic and Deterministic strategy with a Mix fill-in approach for the Branin-25d function. The top 2 curves of each subplot correspond to the first five variables (the non-dummy ones).

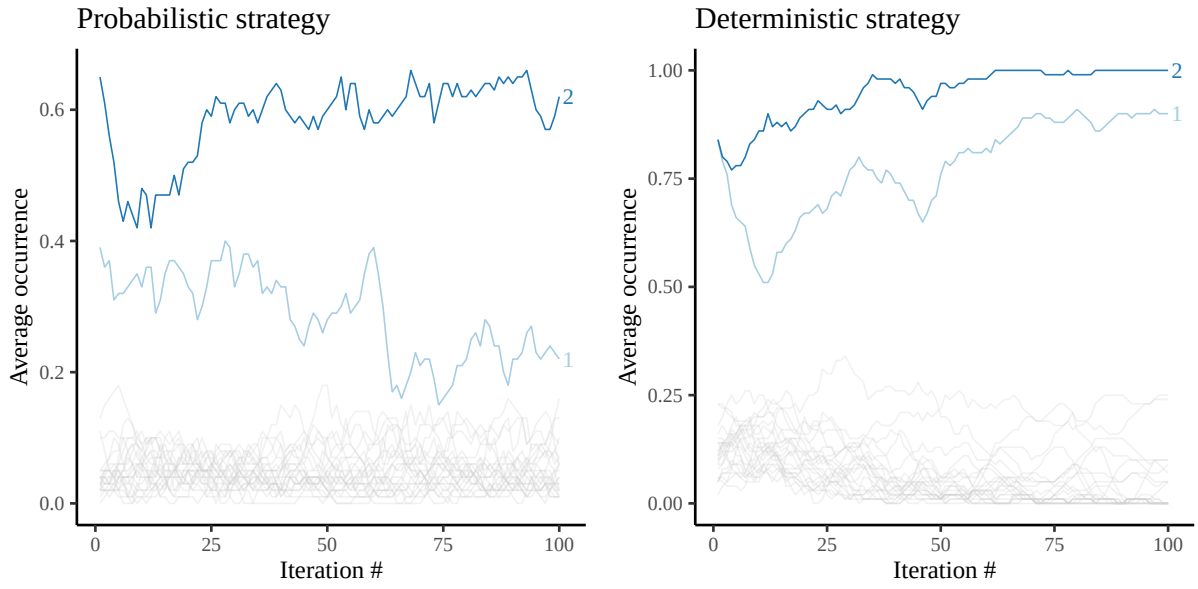

Figure 4.10 – Average selection of occurrence for each variable for the Probabilistic and Deterministic strategy with a Mix fill-in approach for the Branin-25d function. The results are smoothed using a moving average with a 5 iterations window size. The top 2 curves of each subplot correspond to the first five variables (the non-dummy ones).

out (as the effective dimension is equal to the number of variables we keep). This explains why the average cumulative occurrence is slightly lower than with the Deterministic strategy, which also exhibits less selection of the dummy variables.

Regarding performance, Figures 4.11 and 4.12 show that the Dropout version underperforms compared to a full Bayesian optimization and even more when compared to the sensitivity guided versions. It confirms that better ways to choose the variables to be optimized over exist. For the Rosenbrock-20d function, the deterministic strategy with the Copy approach for the dropped out dimensions provides more consistent results (Figure 4.11), yet its median performance is not the best until the last iterations (Figure 4.13). For the Branin-25d, all fill-in strategies with the Deterministic selection show good results (Figure 4.12), their median results are able to reach low values of the objective function quite fast, in less than 30 iterations for the Mix and the Copy ones (Figure 4.14).

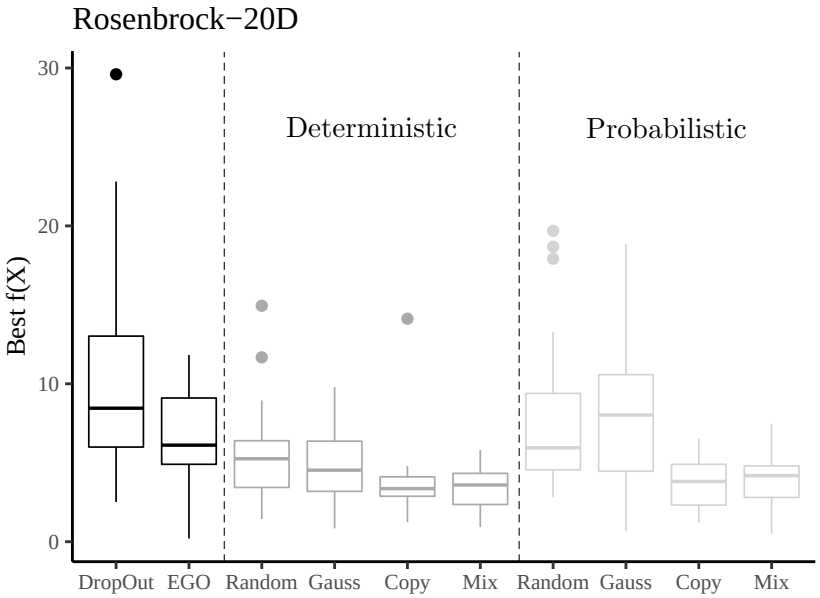

Figure 4.11 – Boxplots of the minimum obtained for the Rosenbrock-20d function over the 20 different initial DOEs with the different selection strategies (Probabilistic in light grey and Deterministic in dark grey) combined with the different fill-in approaches.

The Branin-25d example shows that when the effective dimension d_{eff} is really low compared to the high dimensional d, the Dropout struggles to converge to the optimum as it has a low probability of selecting the first two variables.

Overall, the Deterministic versions appear to be more efficient than their Probabilistic counterparts, especially on the Branin-25d function. This might directly come from the restrictive number of dimensions kept at each iteration, which slows down the convergence.

For the two test cases and both the Deterministic and the Probabilistic selections, the Mix and the Copy strategies yield the best results and consistently outperform the Dropout and the full Bayesian optimization. Yet, the performance of the two selection strategies rely on the value of the hard-coded parameters that were chosen empirically.

The choice of the threshold considered for the definition of $\hat{\mathcal{D}}_{\hat{q}_\alpha}$ can also greatly impact which

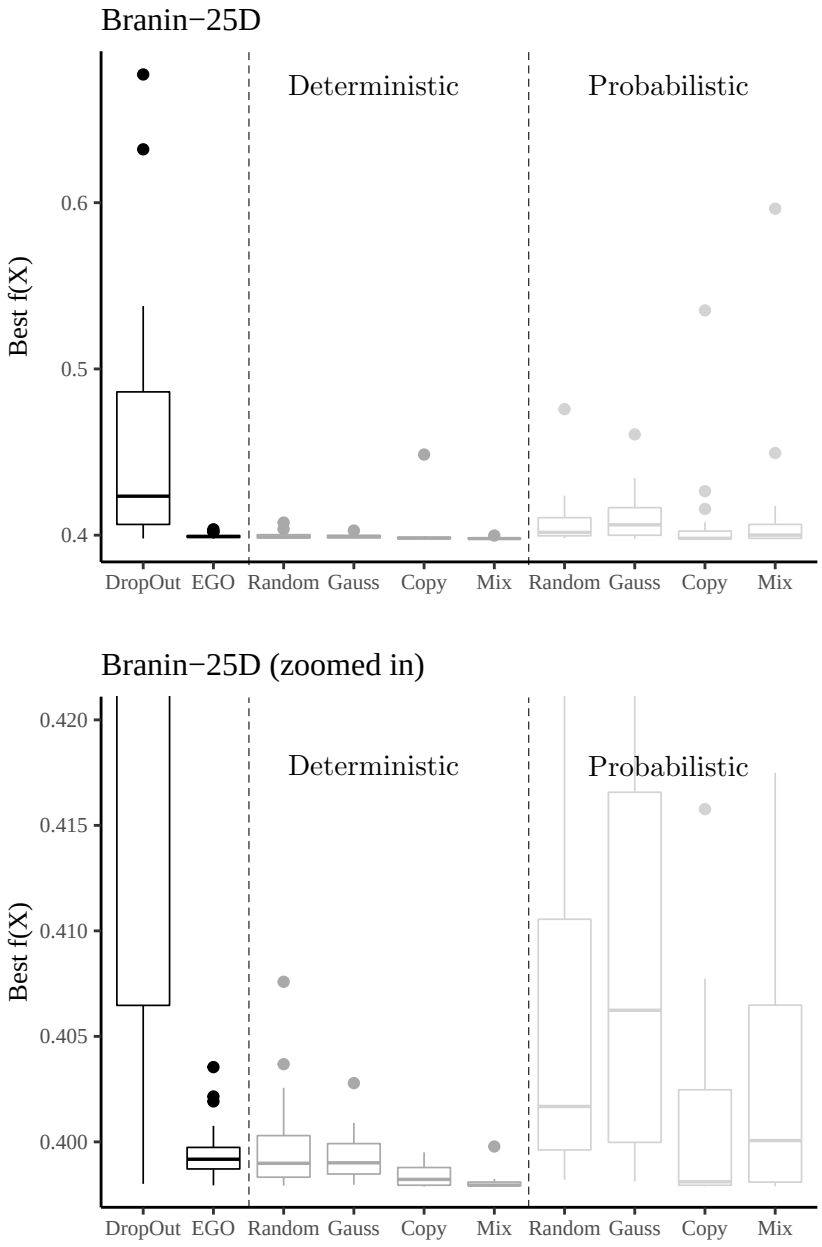

Figure 4.12 – Boxplots of the minimum obtained for the Branin-25d function over the 20 different initial DOEs with the different selection strategies (Probabilistic in light grey and Deterministic in dark grey) combined with the different fill-in approaches. The second plot is a zoom on the lowest value to show how the different methods rank.

dimensions are dropped out as the influence of the inputs differs depending on the sublevel set considered.

4.4 How to make Bayesian Optimization with KSA more robust

In this section, we present three strategies to make the previous algorithm more robust to sampling, function errors and thresholds choices.

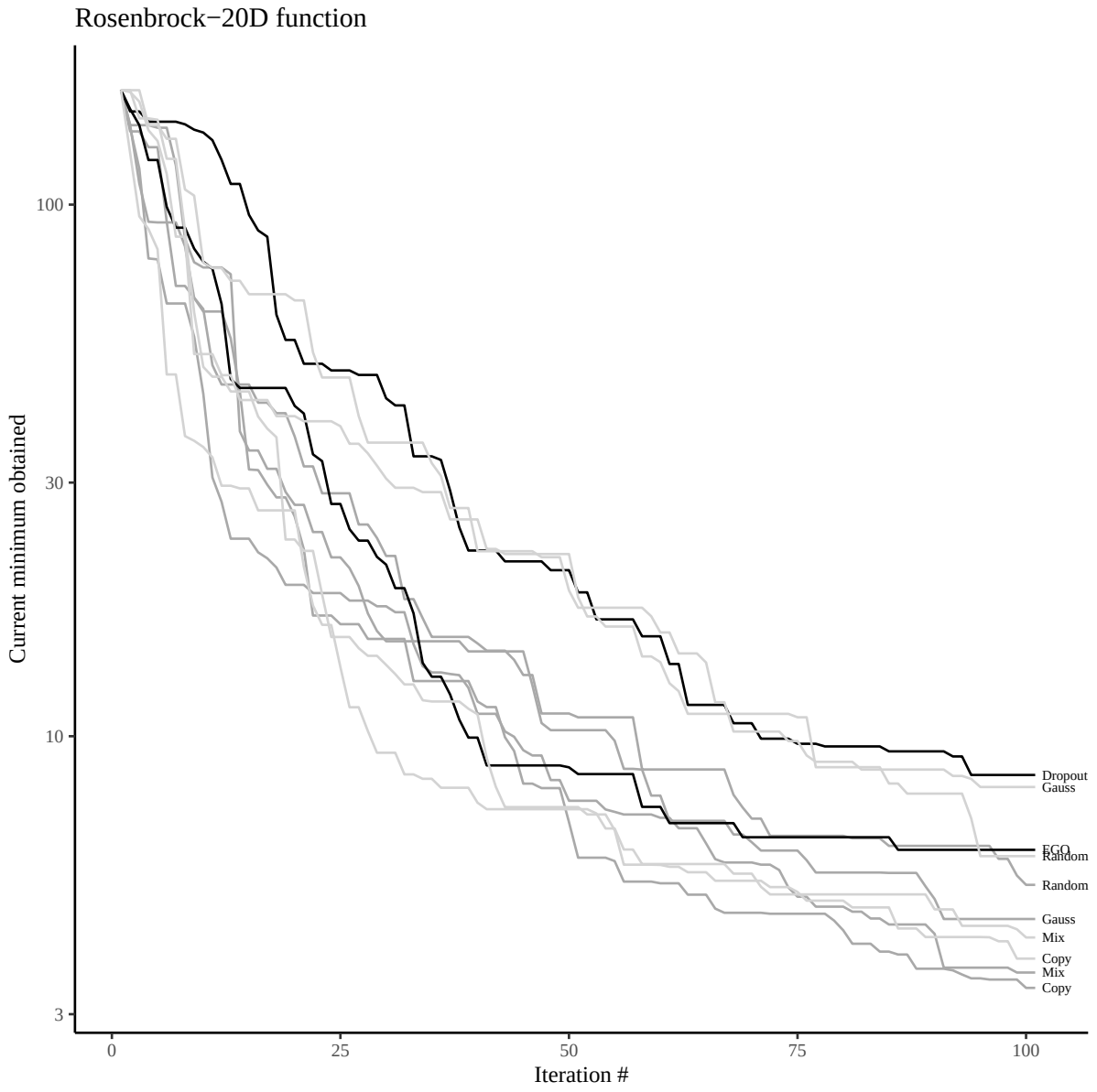

Figure 4.13 – Median current minimum over 20 repetitions obtained with each optimizer for the Rosenbrock-20d function. The dark grey lines corresponds to the Deterministic strategy while the light grey ones corresponds to the Probabilistic strategy. The name of the fill-in approaches is written next to each line.

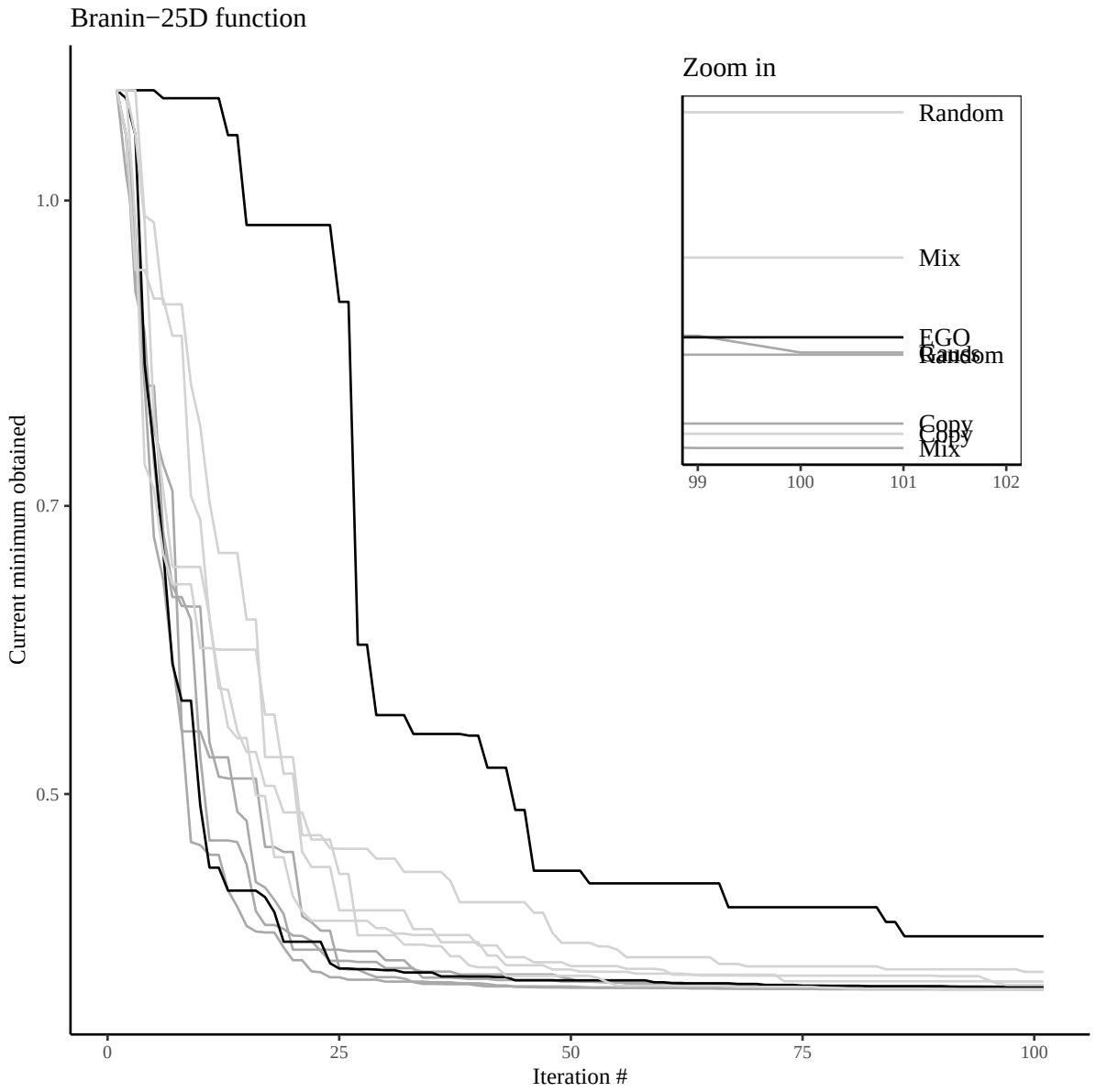

Figure 4.14 – Median current minimum over 20 repetitions obtained with each optimizer for the Branin-25d function. The dark grey lines corresponds to the Deterministic strategy while the light grey ones corresponds to the Probabilistic strategy. The subplot in the top-right corner is a zoom in on the best 6 optimizers for the last iterations, with the name of the corresponding fill-in approaches written next to each line.

4.4.1 Varying threshold levels

The previously defined sensitivity indices, either expressed as Hilbert-Schmidt independence criterion or as maximum mean discrepancy, characterize the relevance of an input to reach a given sublevel denoted $\mathcal{D}_q$ (we omit the α for conciseness). We extend this definition and propose indices that measure the influence of inputs to go from one level set $\mathcal{D}_q$ to a secondary level set $\mathcal{D}_{q'}$:

Definition 4.4.1 (Kernel-based sensitivity index between $\mathcal{D}_q$ and $\mathcal{D}_{q'}$)**.**
Let $f() : \mathbb{R}^d \to \mathbb{R}$ be the objective function of the random variables $\mathbf{X} = (X_1, \dots, X_d)$ and define $\mathcal{D}_q = \{\mathbf{X} \in \mathbb{R}^d, f(\mathbf{X}) \leq q\}$ and $\mathcal{D}_{q'} = \{\mathbf{X} \in \mathbb{R}^d, f(\mathbf{X}) \leq q'\}$ for any $q, q' \in \mathbb{R}$, such that $\mathcal{D}_{q'} \subset \mathcal{D}_q$. The kernel-based sensitivity index of the variable X_i, based on the Hilbert-Schmidt independence criterion is

$$S^{\text{HSIC}}_{\hat{q} \to \hat{q}'}(X_i) = \text{HSIC}((X_i \mid \mathbf{X} \in \mathcal{D}_q), 1_{\mathbf{X} \in \mathcal{D}_{q'}}), \quad q' < q \tag{4.37}$$

with $1_{\mathbf{X} \in \mathcal{D}_{q'}}$ the indicator function equal to 1 when $\mathbf{X} \in \mathcal{D}_{q'}$ and 0 otherwise.

An index equal to 0 means that the distribution of the input X_i when $\mathbf{X} \in \mathcal{D}_q$ is independent from the distribution of the indicator for being in $\mathcal{D}_{q'}$. It can be interpreted as the input X_i having no impact to move from $\mathcal{D}_q$ to $\mathcal{D}_{q'}$. Assume that $\mathcal{D}_q = \mathcal{X}$, we directly retrieve the indices defined in Section 3.3.3. From now on, we write $\mathcal{D}$ and $\mathcal{D}'$ in place of $\mathcal{D}_q$ and $\mathcal{D}_{q'}$ for conciseness.

Once again, we can link this independence measure to the squared maximum mean discrepancy between the kernel mean embeddings of $P_{X_i|\mathbf{X}\in\mathcal{D}}$ and $P_{X_i|\mathbf{X}\in\mathcal{D}'}$:

$$\text{HSIC}(X_i \mid \mathbf{X} \in \mathcal{D}, 1_{\mathbf{X}\in\mathcal{D}'}) \propto \gamma^2(P_{X_i|\mathbf{X}\in\mathcal{D}}, P_{X_i|\mathbf{X}\in\mathcal{D}'}) \tag{4.38}$$

Proof. As in Section 3.3.3, the only requirement to exhibit such relation comes from the choice of the kernel $l(\cdot)$ for the categorical output $Z = 1_{\mathbf{X}\in\mathcal{D}'}$. Z is a discrete variable and $l(\cdot)$ is chosen accordingly, among the different categorical kernels (e.g. the Dirac kernel or the linear kernel). From this, using the integral expression of the HSIC and exploiting the discrete nature of the output Z, we can write

$$\begin{aligned}\text{HSIC}(X_i|X \in \mathcal{D}, Z = 1_{X\in\mathcal{D}'}) = \iint_{\mathcal{X},\mathcal{X}'} \sum_{z=0}^{1}\sum_{z'=0}^{1} k(x,x')l(z,z') \left[p_{X_i|X\in\mathcal{D},Z}(x,z) - p_{X_i|X\in\mathcal{D}}(x)p_Z(z)\right] \\ \times \left[p_{X_i|X\in\mathcal{D},Z}(x',z') - p_{X_i|X\in\mathcal{D}}(x')p_Z(z')\right] dxdx'dzdz'\end{aligned} \tag{4.39}$$

Since $\mathcal{D}' \subset \mathcal{D}$ by definition, we can derive

$$p_{X_i|X\in\mathcal{D},Z}(x,z) = p_{X_i|X\in\mathcal{D}|Z}(x,z)p_Z(z) = p_{X_i|X\in\mathcal{D}'}(x)p_Z(z) \tag{4.40}$$

which leads to

$$\begin{aligned}\text{HSIC}(X_i|X \in \mathcal{D}, Z = 1_{X\in\mathcal{D}'}) = \iint_{\mathcal{X},\mathcal{X}'} \sum_{z=0}^{1}\sum_{z'=0}^{1} k(x,x')l(z,z') \left[p_{X_i|X\in\mathcal{D}'}(x) - p_{X_i|X\in\mathcal{D}}(x)\right] \\ \times \left[p_{X_i|X\in\mathcal{D}'}(x') - p_{X_i|X\in\mathcal{D}}(x')\right] p_Z(z)p_Z(z')dxdx'dzdz'\end{aligned} \tag{4.41}$$

The kernel used for the output Z is discrete and can take 0 or 1 for value, therefore we obtain

$$\begin{aligned}\mathrm{HSIC}(X_i|X \in \mathcal{D}, Z = 1_{X\in\mathcal{D}'}) = \iint_{\mathcal{X},\mathcal{X}'} k(x,x') \left[p_{X_i|X\in\mathcal{D}'}(x) - p_{X_i|X\in\mathcal{D}}(x)\right] \\ \times \left[p_{X_i|X\in\mathcal{D}'}(x') - p_{X_i|X\in\mathcal{D}}(x')\right] P_Z(z=1)P_Z(z'=1)dxdx'\end{aligned} \tag{4.42}$$

Finally, for kernels that verify $l(z,z') = 0$ if $z \neq z' \neq 1$, we then derive

$$\begin{aligned}\mathrm{HSIC}(X_i|X \in \mathcal{D}, Z = 1_{X\in\mathcal{D}'}) &= \iint_{\mathcal{X},\mathcal{X}'} k(x,x') \left[p_{X_i|X\in\mathcal{D}'}(x) - p_{X_i|X\in\mathcal{D}}(x)\right] \\ &\times \left[p_{X_i|X\in\mathcal{D}'}(x') - p_{X_i|X\in\mathcal{D}}(x')\right] P_Z(z=1)^2 dxdx' \\ &= P_Z(z=1)^2 \times \gamma^2(P_{X_i|\mathbf{X}\in\mathcal{D}}, P_{X_i|\mathbf{X}\in\mathcal{D}'})\end{aligned} \tag{4.43}$$

□

Let $\mathbb{X}$ a sample of size n and $\mathbb{Y} = f(\mathbb{X})$ the corresponding observations, we can again resort to the unbiased estimator of the maximum mean discrepancy applied to the random variables $X_i \mid \mathbf{X} \in \mathcal{D}$ and $X_i \mid \mathbf{X} \in \mathcal{D}'$. This requires to properly define $\mathcal{D}$ and $\mathcal{D}'$. The task at hand is a minimization and the level sets $\mathcal{D}$ and $\mathcal{D}'$ can be seen as an achieved and a targeted set of solutions. Then q and q' must verify $q' < q$ so that $\mathcal{D}' \subset \mathcal{D}$. To ensure sufficiently low values, we can define the two sublevel sets of interest by low quantiles of the objective function f: $q = F^{-1}_{f(\mathbf{X})}(\alpha)$ and $q' = F^{-1}_{f(\mathbf{X})}(\alpha')$. After determining the thresholds, we define for an input X_i the subsamples

$$\tilde{\mathbb{X}}_i = \mathbb{X}_i \mid \mathbb{X} \in \mathcal{D} \tag{4.44}$$

$$\hat{\mathbb{X}}_i = \mathbb{X}_i \mid \mathbb{X} \in \mathcal{D}' \tag{4.45}$$

With $\tilde{\mathbb{X}}_i$ and $\hat{\mathbb{X}}_i$, we can use Equation (4.38) and the unbiased estimator of the maximum mean discrepancy since it only involves the associated Gram matrices:

$$\begin{aligned}S^{\mathrm{HSIC}}_{q\to q'}(\mathbb{X}_i) = \gamma^2_u(\tilde{\mathbb{X}}_i, \hat{\mathbb{X}}_i) = &\frac{1}{n_1(n_1-1)} \sum_{p=1}^{n_1}\sum_{q\neq p}^{n_1} k(\tilde{\mathbb{X}}_i^p, \tilde{\mathbb{X}}_i^q) + \frac{1}{n_2(n_2-1)} \sum_{i=p}^{n_2}\sum_{q\neq p}^{n_2} k(\hat{\mathbb{X}}_i^p, \hat{\mathbb{X}}_i^q) \\ &- \frac{2}{n_1 n_2} \sum_{p=1}^{n_1}\sum_{q=1}^{n_2} k(\tilde{\mathbb{X}}_i^p, \hat{\mathbb{X}}_i^q)\end{aligned} \tag{4.46}$$

with n_1 and n_2 the size of the subsamples $\tilde{\mathbb{X}}_i$ and $\hat{\mathbb{X}}_i$, respectively. They depend on the size n of the sample $\mathbb{X}_i$ and on the values chosen for α and α'. The indices are normalized like in Equation (4.36).

The quantile levels α and α' define the sets $\mathcal{D}$ and $\mathcal{D}'$. We investigate the influence of the thresholds by testing multiple configurations (α, α'), i.e., $(\mathcal{D}, \mathcal{D}')$, on a simple ellipsoid function

$$f(\mathbf{X}) = X_1^2 + 100X_2^2. \tag{4.47}$$

For an ellipsoid, the importance of a variable for a given pair $(\mathcal{D}, \mathcal{D}')$ is easily visible: the second variable is the most important to achieve high quantile levels; the first variable is however the

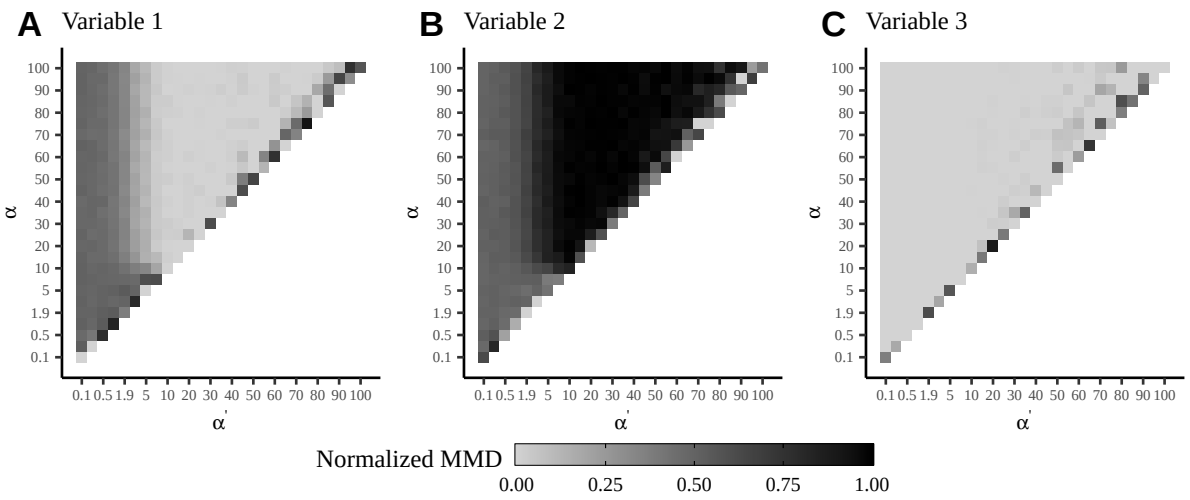

Figure 4.15 – Normalized MMD maps of the three-dimensional ellipsoid function for varying pairs (α, α'). A darker color corresponds to a high value of sensitivity. The third variable is a dummy variable and has zero influence for all pairs (α, α'). Darker areas for X_3 come from the estimation noise of the MMD because the number of points for the estimation is limited to 200.

most important to achieve low quantiles. We add a third dummy variable for comparison purposes. For varying configurations of $(\mathcal{D}, \mathcal{D}')$, the normalized indices are estimated from 10 different LHS using $n_1 = n_2 = 200$ points, i.e., n is taken sufficiently large to allow $n_1 = 200$. Figure 4.15 shows the average normalized sensitivity indices (as MMD values) for each pair (α, α').

Globally, the sensitivity maps agree with what is expected from the analytic expression of the ellipsoid: the larger sensitivities of X_2 at high α' confirm that only X_2 matters to attain high values of the ellipsoid.

At lower α' (i.e., low objectives), all active variables have substantial sensitivities. It is observed on the maps, but it is a general result: when α' decreases, the distribution of good points peaks around the best observed point. The non normalized sensitivities tend to

$$\lim_{\alpha' \to 0} \gamma^2(P_{X_i|X\in\mathcal{D}}, P_{X_i|X\in\mathcal{D}'}) \;=\; \gamma^2(P_{X_i|X\in\mathcal{D}}, \delta_{X_i^\star}) \;, \tag{4.48}$$

where $\delta_{X_i^\star}$ is the Dirac distribution centered on the i-th component of the optimum. In the ellipsoid example (Figure 4.15), the sensitivities at low α' make the left maps columns and have a similar order of magnitude for X_1 and X_2. In general, the limits of the sensitivities for challenging levels differ between variables and are given by Equation (4.48), but they differ from zero if α is sufficiently larger than α'.

It is also confirmed on the map that X_3, the dummy variable, always has near zero sensitivities, excepted close to the diagonal $\alpha = \alpha'$ for a normalization reason that we explain next.

Ultimately, the following guidelines should be followed:

1. *α should be sufficiently larger than α'.*
   Along the diagonal where $\mathcal{D} \approx \mathcal{D}'$ and before normalization, the sensitivities measured for both true variables X_1 and X_2 are similar to that of the dummy variable X_3. These sensitivities are almost zero because the closeness of α and α' results in sets $\mathcal{D}$ and $\mathcal{D}'$ similar

to each other, which in turns implies that the distance between $P_{X_i|X\in\mathcal{D}}$ and $P_{X_i|X\in\mathcal{D}'}$ measured by $S^{\text{HSIC}}_{q\to q'}(\mathbb{X}_i)$ is almost 0 like with a dummy variable, see Figure 4.16 for an illustrative example. The observed non-zero normalized sensitivities of the dummy X_3 variable on the diagonal come from the normalization by a near zero sum of sensitivities. To avoid such issue, a sufficient distance should be considered between q and q' (i.e., α and α').

2. α' *should be large enough*, typically $\alpha' > 2\%$.
   As explained earlier, setting α' below about 2% generates peaked target densities that make all true variables sensitive, therefore canceling the benefits of variable selection. In general, it should be avoided. This argument was confirmed by complementary optimization runs where small α' led to poor performance. These runs are not reported here. The special situation where both α and α' are small deserve a special attention. It is a tempting setting because of its interpretation: the level sets considered correspond to the high performance regions one is truly interested in during an optimization. However, like in the first guideline above, the densities of $X_i \mid X \in \mathcal{D}$ and $X_i \mid X \in \mathcal{D}'$ are very much alike and the estimation of their distance requires a very large number of samples n.

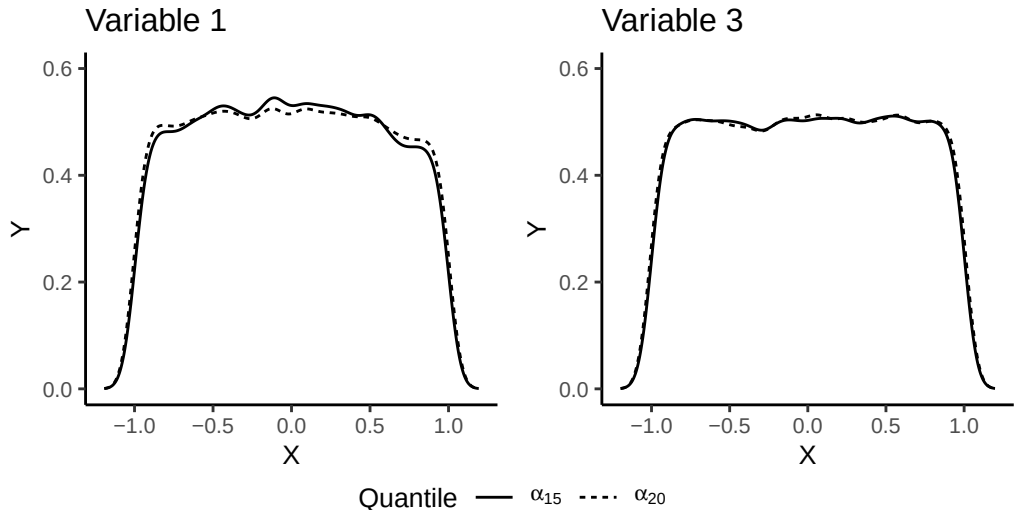

Figure 4.16 – Comparison of the distribution of $P_{\mathbb{X}_i|\mathbb{X}\in\mathcal{D}'}$ for q' equal to the 15%-quantile of the output (continuous line) and $P_{\mathbb{X}_i|\mathbb{X}\in\mathcal{D}}$ for q equal to the 20%-quantile (dashed line). The left panel shows the distributions for X_1 while the right panel shows the distributions for X_3. Variable X_3 is not active for this ellipsoid function. In both cases, there is a small difference between the distributions.

The computations of both quantiles would require numerous evaluations of the objective function, especially for quantiles in the tail of the distribution. Since we work in a Bayesian optimization setting, we can rely on the mean of the conditioned Gaussian process in place of the expensive objective function and define $\hat{q} = F^{-1}_{\mu(\mathbf{X})}(\alpha)$ and $\hat{q}' = F^{-1}_{\mu(\mathbf{X})}(\alpha')$. This allows to approach the true sublevel set $\mathcal{D}$ with $\hat{\mathcal{D}} = \{\mathbf{X} \in \mathcal{X}, \mu(\mathbf{X}) \leq \hat{q}\}$ and $\mathcal{D}'$ with $\hat{\mathcal{D}}' = \{\mathbf{X} \in \mathcal{X}, \mu(\mathbf{X}) \leq \hat{q}'\}$.

4.4.2 Accounting for model error through conditional trajectories

In Bayesian Optimization, the true function is modeled by a Gaussian Process (GP) to save evaluations. We have proposed in the current work to reduce the dimension of Bayesian Optimization through a variable selection based on kernel-based Sensitivity Analysis (KSA), yielding the KSA-BO algorithm. So far, the KSA has been made with the conditional GP mean. Instead of solely doing the KSA estimation with the predictive mean, it is possible to repeat the calculation of the indices with conditional simulations of the GP (also called conditional trajectories) and average the results over all trajectories. By doing so, the uncertainty in the model of the function (the conditional GP) is accounted for in the estimation of $S^{\text{HSIC}}_{q\to q'}(\mathbb{X}_i)$, therefore providing additional reliability in the subsequent variable selection. But this comes at the additional cost of computing the conditional trajectories and repeating the sensitivity analyses. Note that for the estimation of the kernel-based indices, the initial cost is still the same since the Gram matrix on the samples must be assembled then it only requires to extract the right element from it.

Like with the initial KSA-BO, it is first necessary to estimate level sets values as quantiles of the GP mean,$\hat{q} = F^{-1}_{\mu(\mathbf{X})}(\alpha)$ and $\hat{q}' = F^{-1}_{\mu(\mathbf{X})}(\alpha')$. Then, sublevels of interest can be defined for each conditional trajectory $\hat{f}^{(l)}(\cdot)$ as

$$\hat{\mathcal{D}}^{(l)} = \{x \in \mathcal{X} \mid \hat{f}^{(l)}(x) \leq q\}, \quad \text{idem for } \hat{\mathcal{D}}'^{(l)} \text{ with } q'.$$

A one dimensional example is given in Figure 4.17. The definition of the subsamples associated to the trajectories, $\hat{\mathbb{X}}^{(l)}_i$ and $\tilde{\mathbb{X}}^{(l)}_i$ is the same as in Equation (4.44) and Equation (4.45) using the sampled level sets $\hat{\mathcal{D}}^{(l)}$ and $\hat{\mathcal{D}}'^{(l)}$.

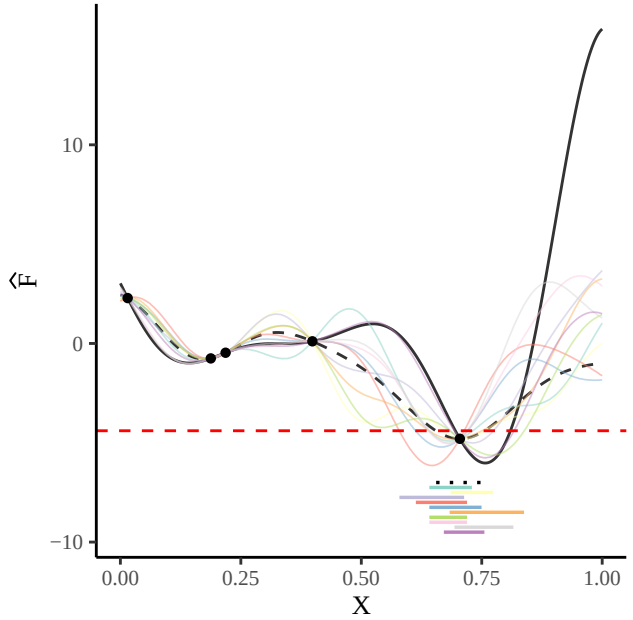

Figure 4.17 – Evolution of the sublevels of interest $\hat{\mathcal{D}}^{(l)}$ for different conditional simulations, with their range shown as horizontal bars at the bottom. The dashed horizontal line corresponds to $\hat{\mathcal{D}}$, computed on the mean of the Gaussian process, considering $\hat{\mathcal{D}} = \{\mathbf{X} \in \mathcal{X}, \mu(\mathbf{X}) \leq q\}$. The red dashed line is $q = F^{-1}_{\mu(\mathbf{X})}(10\%)$. The black line is the true function and black dots are observations.

The average estimator of the kernel-based sensitivity is then computed following

$$S^{\text{HSIC}}_{q\to q',T}(\mathbb{X}_i) = \frac{1}{T}\sum_{l=1}^{T}\gamma_u^2(\tilde{\mathbb{X}}_i^{(l)}, \hat{\mathbb{X}}_i^{(l)}) \tag{4.49}$$

for T trajectories. This operation requires to compute the sensitivity indices for each new conditional simulation, with the additional cost of simulating the conditional trajectory in a first place. Thus, having an analytic expression for $\mathbb{E}(\gamma^2(P_{X_i|\mathbf{X}\in\mathcal{D}}, P_{X_i|\mathbf{X}\in\mathcal{D}'}))$ would help. It is possible to start from Equations (4.46) and (4.49) and we have to compute :

$$\begin{aligned}\mathbb{E}\left(S^{\text{HSIC}}_{q\to q',T}(\mathbb{X}_i)\right) =& \mathbb{E}\left(\frac{1}{n_1(n_1-1)}\sum_{p=1}^{n_1}\sum_{q\neq p}^{n_1} k(\tilde{\mathbb{X}}_i^p, \tilde{\mathbb{X}}_i^q)\right) + \mathbb{E}\left(\frac{1}{n_2(n_2-1)}\sum_{i=p}^{n_2}\sum_{q\neq p}^{n_2} k(\hat{\mathbb{X}}_i^p, \hat{\mathbb{X}}_i^q)\right)\\ &- \mathbb{E}\left(\frac{2}{n_1 n_2}\sum_{p=1}^{n_1}\sum_{q=1}^{n_2} k(\tilde{\mathbb{X}}_i^p, \hat{\mathbb{X}}_i^q)\right)\\ =& A_1 + A_2 - A_3\end{aligned} \tag{4.50}$$

Let $\hat{F}$ be our posterior distribution, which is known to be Gaussian with explicit mean and variance. For the sake of readability, we use X in place of $\mathbb{X}_i$. We can work term by term and start with:

$$A_1 = \mathbb{E}\left(\frac{1}{n_1(n_1-1)}\sum_{i=1}^{n_1}\sum_{j\neq p}^{n_1} k(\tilde{X}^i, \tilde{X}^j)\right) \tag{4.51}$$

and notice that since $\tilde{X}^i$ is distributed as $X^i \mid \hat{F}(X^i) \leq q$

$$\begin{aligned}&\frac{1}{n_1(n_1-1)}\sum_{i=1}^{n_1}\sum_{j\neq i}^{n_1} k(\tilde{X}^i, \tilde{X}^j) =\\ &\qquad \frac{1}{\sum_{l=1}^{n} 1_{\hat{F}(X^l)\leq q}(\sum_{l'=1}^{n} 1_{\hat{F}(X^{l'})\leq q} - 1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(X^i, X^j) 1_{\hat{F}(X^i)\leq q} 1_{\hat{F}(X^j)\leq q}\end{aligned} \tag{4.52}$$

Computing the expectation of the previous expression leads to

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(X^i, X^j)\mathbb{E}\left(\frac{1_{\hat{F}(X^i)\leq q} 1_{\hat{F}(X^j)\leq q}}{\sum_{l=1}^{n} 1_{\hat{F}(X^l)\leq q}(\sum_{l'=1}^{n} 1_{\hat{F}(X^{l'})\leq q} - 1)}\right) =\\ &\qquad \sum_{i=1}^{n}\sum_{j\neq i}^{n} k(X^i, X^j)\mathbb{E}\left(\frac{1_{\hat{F}(X^i)\leq q\cap \hat{F}(X^j)\leq q}}{\sum_{l,l'=1}^{n} 1_{\hat{F}(X^l)\leq q\cap \hat{F}(X^{l'})\leq q} - \sum_{l=1}^{n} 1_{\hat{F}(X^l)\leq q}}\right)\end{aligned} \tag{4.53}$$

A first order approximation to the expectation of a ratio is $\mathbb{E}(X/Y) \approx \mathbb{E}(X)/\mathbb{E}(Y)$ and a second order is

$$\mathbb{E}\left(\frac{X}{Y}\right) \approx \frac{\mathbb{E}(X)}{\mathbb{E}(Y)} - \frac{\text{Cov}(X,Y)}{\mathbb{E}(Y)^2} + \frac{\mathbb{E}(X)}{\mathbb{E}(Y)^3}\mathbb{V}(Y) \tag{4.54}$$

which can be seen by Taylor expansion of $1/Y$ around $\mathbb{E}(Y)$.

We can then derive

$$
\begin{aligned}
A1 \approx \sum_{i=1}^{n}\sum_{j\neq i}^{n} k(X^i, X^j)\Bigg(& \frac{2(P(\hat{F}(X^i)\leq q, \hat{F}(X^j)\leq q)\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q, \hat{F}(X^{l'})\leq q))}{\left(\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q, \hat{F}(X^{l'})\leq q)\right)^2} \\
& - \frac{\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q, \hat{F}(X^{l'})\leq q, \hat{f}(X^i)\leq q, \hat{f}(X^j)\leq q)}{\left(\sum_{l,l'=1}^{n} P(\hat{f}(X^l)\leq q, \hat{f}(X^{l'})\leq q)\right)^2} \\
& - \frac{P(\hat{f}(X^i)\leq q, \hat{f}(X^j)\leq q)\sum_{l,l',l'',l'''} P(\hat{f}(X^l)\leq q, \hat{f}(X^{l'})\leq q, \hat{f}(X^{l''})\leq q, \hat{f}(X^{l'''})\leq q)}{\left(\sum_{l,l'=1}^{n} P(\hat{f}(X^l)\leq q, \hat{f}(X^{l'})\leq q)\right)^3} \\
& - \frac{2(P(\hat{f}(X^i)\leq q, \hat{f}(X^j)\leq q)\sum_{l=1}^{n} P(\hat{f}(X^l)\leq q))}{\left(\sum_{l=1}^{n} P(\hat{f}(X^l)\leq q)\right)^2} \\
& + \frac{\sum_{l=1}^{n} P(\hat{f}(X^l)\leq q, \hat{f}(X^i)\leq q, \hat{f}(X^j)\leq q)}{\left(\sum_{l=1}^{n} P(\hat{f}(X^l)\leq q)\right)^2} \\
& + \frac{P(\hat{f}(X^i)\leq q, \hat{f}(X^j)\leq q)(\sum_{l,l'=1}^{n} P(\hat{f}(X^l)\leq q, \hat{f}(X^{l'})\leq q))}{\left(\sum_{l=1}^{n} P(\hat{f}(X^l)\leq q)\right)^3}\Bigg)
\end{aligned}
\tag{4.55}
$$

Since $\hat{F}$ is Gaussian, we can compute directly $P(\hat{F}(X)\leq q)$ using the cumulative distribution function $\Phi(\cdot)$. Yet, Equation (4.55) requires the estimation of joint cumulative distribution functions in dimension 3 and 4. It is done with nested loops which burdens the computation of this term as n grows.

The second term leads to similar computations,

$$
\begin{aligned}
A_2 &= \mathbb{E}\left(\frac{1}{n_1(n_2-1)}\sum_{i=1}^{n_2}\sum_{j\neq p}^{n_2} k(\hat{X}^i, \hat{X}^j)\right) \\
&= \sum_{i=1}^{n}\sum_{j\neq i}^{n} k(X^i, X^j)\mathbb{E}\left(\frac{1_{\hat{F}(X^i)\leq q'\cap\hat{F}(X^j)\leq q'}}{\sum_{l,l'=1}^{n} 1_{\hat{F}(X^l)\leq q'\cap\hat{F}(X^{l'})\leq q'} - \sum_{l=1}^{n} 1_{\hat{F}(X^l)\leq q'}}\right) \\
&\approx \sum_{i=1}^{n}\sum_{j=1}^{n} k(X^i, X^j)\Bigg(\frac{2(P(\hat{F}(X^i)\leq q', \hat{F}(X^j)\leq q')\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q', \hat{F}(X^{l'})\leq q'))}{\left(\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q', \hat{F}(X^{l'})\leq q')\right)^2} \\
&\quad - \frac{\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q', \hat{F}(X^{l'})\leq q', \hat{F}(X^i)\leq q', \hat{F}(X^j)\leq q')}{\left(\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q', \hat{F}(X^{l'})\leq q')\right)^2} \\
&\quad - \frac{P(\hat{F}(X^i)\leq q', \hat{F}(X^j)\leq q')\sum_{l,l',l'',l'''} P(\hat{F}(X^l)\leq q', \hat{F}(X^{l'})\leq q', \hat{F}(X^{l''})\leq q', \hat{F}(X^{l'''})\leq q')}{\left(\sum_{l,l'=1}^{n} P(\hat{F}(X^l)\leq q', \hat{F}(X^{l'})\leq q')\right)^3} \\
&\quad - \frac{2(P(\hat{F}(X^i)\leq q', \hat{F}(X^j)\leq q')\sum_{l=1}^{n} P(\hat{F}(X^l)\leq q'))}{\left(\sum_{l=1}^{n} P(\hat{F}(X^l)\leq q')\right)^2} \\
&\quad + \frac{\sum_{l=1}^{n} P(\hat{F}(X^l)\leq q', \hat{F}(X^i)\leq q', \hat{F}(X^j)\leq q')}{\left(\sum_{l=1}^{n} P(\hat{F}(X^l)\leq q')\right)^2}
\end{aligned}
$$

$$+\frac{P(\hat{F}(X^i)\leq q',\hat{F}(X^j)\leq q')(\sum_{l,l'=1}^n P(\hat{F}(X^l)\leq q',\hat{F}(X^{l'})\leq q'))}{\left(\sum_{l=1}^n P(\hat{F}(X^l)\leq q')\right)^3}\Bigg) \tag{4.56}$$

Finally, for the cross-term, we obtain

$$\begin{aligned}
A_3 &= \mathbb{E}\left(\frac{2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} k(\tilde{X}^i,\hat{X}^j)\right)\\
&= \sum_{i=1}^n\sum_{j=1}^n k(X^i,X^j)\mathbb{E}\left(\frac{1_{\hat{F}(X^i)\leq q\cap\hat{F}(X^j)\leq q'}}{\sum_{l,l'=1}^n 1_{\hat{F}(X^l)\leq q\cap\hat{F}(X^{l'})\leq q}}\right) && (4.57)\\
&\approx \sum_{i=1}^n\sum_{j=1}^n k(X^i,X^j)\Bigg(\frac{(P(\hat{F}(X^i)\leq q,\hat{F}(X^j)\leq q')\sum_{l,l'=1}^n P(\hat{F}(X^l)\leq q,\hat{F}(X^{l'})\leq q'))}{\left(\sum_{l,l'=1}^n P(\hat{F}(X^l)\leq q,\hat{F}(X^{l'})\leq q')\right)^2}\\
&- \frac{\sum_{l,l'} P(\hat{F}(X^l)\leq q,\hat{F}(X^{l'})\leq q',\hat{F}(X^i)\leq q,\hat{F}(X^j)\leq q')}{\left(\sum_{l,l'=1}^n P(\hat{F}(X^l)\leq q,\hat{F}(X^{l'})\leq q')\right)^2} && (4.58)\\
&+ \frac{P(\hat{F}(X^i)\leq q,\hat{F}(X^j)\leq q')\sum_{l,l',l'',l'''} P(\hat{F}(X^l)\leq q,\hat{F}(X^{l'})\leq q',\hat{F}(X^{l''})\leq q,\hat{F}(X^{l'''})\leq q')}{\left(\sum_{l,l'=1}^n P(\hat{F}(X^l)\leq q,\hat{F}(X^{l'})\leq q')\right)^3}\Bigg) && (4.59)
\end{aligned}$$

The computation of all the terms quickly appears to be intractable because of the nested loops involved. We can still however consider only the first order approximation and look at its bias with respect to the empirical mean of the sensitivity. We compare it to the value of Equation (4.49) for an increasing number of trajectories for the 5 dimensional Rosenbrock function Section 4.3.2, defined in $\mathcal{X}=[-4,4]^5$, see Figure 4.18. We choose a squared exponential kernel for the Gaussian process prior. We consider $q=100\%$ quantile (i.e. $\hat{\mathcal{D}}=\mathcal{X}$) and $q'=10\%$ quantile of the mean of the surrogate model, estimated with 1000 points. The kernel used for the computation of the maximum mean discrepancy is the squared exponential kernel. For each variable, a visible bias between the empirical mean of the sensitivities and the first order approximations is visible. Yet, the order of the variable is preserved. Convergence of the empirical mean is achieved after about 1000 trajectories.

Figure 4.19 shows the different results we obtain between comparison of the trajectory-based indices using first order approximation, the empirical mean of indices (Equation (4.49), using 1000 trajectories) and the indices calculated on the GP mean, Equation (4.35). Obviously, the empirical mean, in green bullets, agrees well with the distribution of indices. The first order approximation, shown as red bullets, is usable in that its magnitude is representative of the empirical mean and the order of the variables is captured. The sensitivities calculated on the GP mean (blue bullets) are more different. In later numerical tests, both the empirical mean and the indices computed on the GP mean will be compared against other strategies.

After computing the sensitivity indices with the conditional trajectories, they can directly replace the indices used within the selection framework introduced in Section 4.3. After normalization, the indices from Equation (4.49)

$$\hat{S}^\gamma_{q\to q',T}(\mathbb{X}_i)=\frac{S^\gamma_{q\to q',T}(\mathbb{X}_i)}{\sum_{j=1}^d S^\gamma_{q\to q',T}(\mathbb{X}_j)}, \tag{4.60}$$

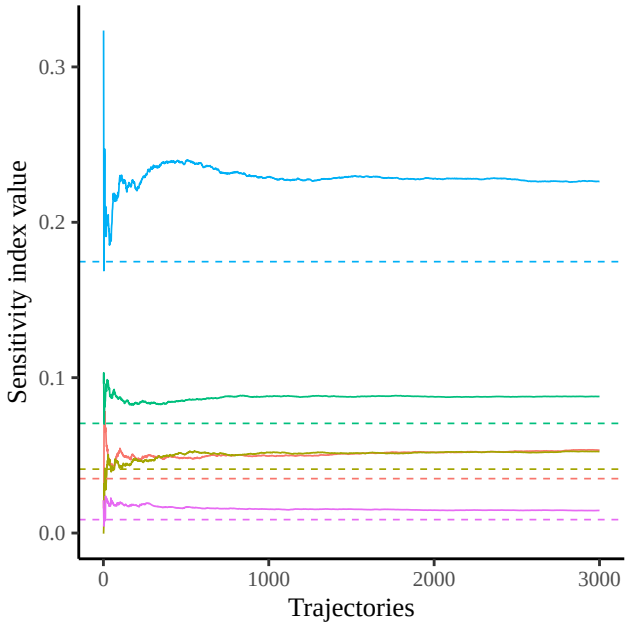

Figure 4.18 – Evolution of $S^{\text{HSIC}}_{q\to q',T}(\mathbb{X}_i)$ with the number of trajectories. Dashed lines corresponds to the first order approximation of $\mathbb{E}\left(S^{\text{HSIC}}_{q\to q',T}(\mathbb{X}_i)\right)$.

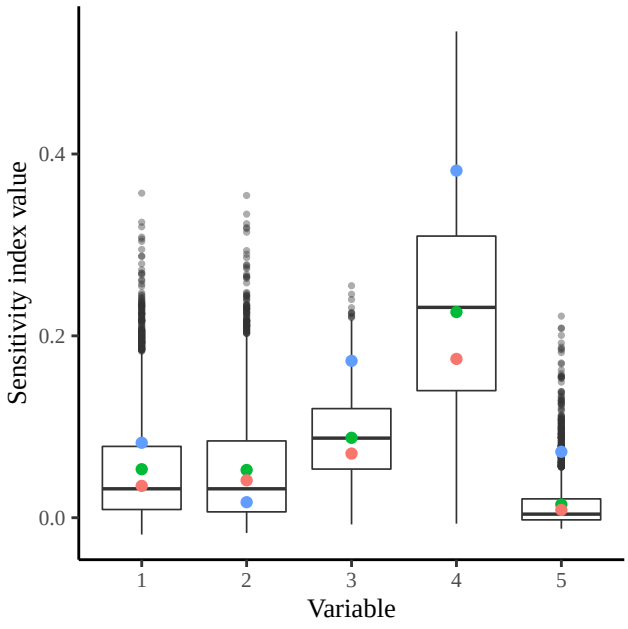

Figure 4.19 – Boxplot of $\gamma_u^2(\tilde{\mathbb{X}}_i^{(t)}, \hat{\mathbb{X}}_i^{(t)})$ for different conditional trajectories. Red dots are the first order approximation of the $\mathbb{E}\left(S^{\text{HSIC}}_{q\to q',T}(\mathbb{X}_i)\right)$, green dots correspond to the empirical average over all trajectories $S^{\text{HSIC}}_{q\to q',T}(\mathbb{X}_i)$ while the blue dots are the sensitivity indices computed solely on the predictor mean $S^{\text{HSIC}}_{q\to q'}(\mathbb{X}_i)$.

are directly integrated in the Probabilistic and Deterministic strategies. The approaches remain the same. In the Probabilistic approach, the active variables are randomly drawn with a probability given by the normalized average indices, $\hat{S}^{\gamma}_{q\to q',T}$. In the Deterministic variant, all variables for which $\hat{S}^{\gamma}_{q\to q',T}$ is above a given detection threshold are kept. The suffix "+ Traj" in the coming results section means that the indices were computed using conditional trajectories.

4.4.3 A parameter free variable selection strategy

Both selection strategies introduced in Section 4.3 require to define and tune a hyperparameter: the detection threshold of the Deterministic approach and the number of variables to keep, d_e, for the Probabilistic approach. These parameters influence the sensitivity analysis step and thus the performance of the objective function optimization.

To make the KSA-BO method less sensitive to the choice of such parameters, we propose a strategy to detect influential variables using a non-parametric statistical test. The test takes as null hypothesis that all variables do not contribute to reaching the level set of interest: $H_0 : P_{X_i|\mathbf{X}\in\mathcal{D}} = P_{X_i|\mathbf{X}\in\mathcal{D}'}$ for a given input X_i.

Since the distribution under the null hypothesis is not explicitly known, approximating it is necessary before comparing it to the test statistic. Here, the test statistic $\mathcal{T}$ is the maximum mean discrepancy computed on a given sample $\gamma^2_u(\tilde{\mathbb{X}}_i, \hat{\mathbb{X}}_i)$. Using the subsamples $(\tilde{\mathbb{X}}_i, \hat{\mathbb{X}}_i)$, an estimation of the null-distribution is obtained through permutation-based resampling. The strategy is the following: assemble both subsamples into a single set $\mathbb{X}^p$, a procedure often known as *pooling*, then randomly sample from $\mathbb{X}^p$ to obtain two new subsamples $(\tilde{\mathbb{X}}^p_i, \hat{\mathbb{X}}^p_i)$. The sensitivity index is computed for this permutation as $\gamma^2_u(\tilde{\mathbb{X}}^p_i, \hat{\mathbb{X}}^p_i)$. This process is repeated n_p times and we compute the p-value as

$$p_{\text{val}} = \frac{1}{n_p}\sum_{p=1}^{n_p} 1_{\gamma^2_u(\tilde{\mathbb{X}}^p_i,\hat{\mathbb{X}}^p_i)>\gamma^2_u(\tilde{\mathbb{X}}_i,\hat{\mathbb{X}}_i)} \tag{4.61}$$

This value is compared against a significance level s. If it is lower, we can reject the null hypothesis and classify the input as important in the optimization setting for the specified levels $(\mathcal{D}, \mathcal{D}')$. The significance level characterizes the probability of rejecting the null hypothesis when it is true. It must be chosen according to the number of permutations. Although the significance level and n_p are parameters of the method, there have a statistical meaning that allows to choose a value a priori. Therefore, they are arguably of secondary importance when compared to the parameters of the Deterministic and Probabilistic selection strategies, hence the "parameter free" denomination.

As explained in the theoretical aspects of the maximum mean discrepancy (Section 2.2.2), when using the quadratic estimator in linear time Equation (2.37), the null distribution is Gaussian with known variance, meaning we could directly compute the threshold to compare our test statistic against. However, this estimator requires too many samples to obtain a reliable value for our sensitivity index and is not well-suited for this application. Hence, we use the unbiased estimator from Equation (2.32) in Equation (4.61).

The estimation of the permuted statistics can be fasten by precomputing the matrix $\tilde{K} = K(\tilde{\mathbb{X}}_i, \tilde{\mathbb{X}}_i)$ and by properly extracting the row and columns that are sampled for each permutation. Assembling $\tilde{K}$ has no additional cost since it is required for the computation of $\gamma^2_u(\tilde{\mathbb{X}}_i, \hat{\mathbb{X}}_i)$

in the first place.

Within the Bayesian optimization framework and for the following numerical tests, we test two configurations: $n_p = 200$ for a significance level of $s = 5\%$ and $n_p = 1000$ for a significance level of $s = 1\%$.

4.4.4 Numerical tests

Numerical tests are carried out on a set of analytic functions in order to compare the new selection strategies, those with the conditional trajectories (Section 4.4.2) and the permutations (Section 4.4.3), to the earlier approaches of Section 4.3. The different configurations that are benchmarked are listed in Table 4.2, with the corresponding selection strategy, fill-in approach and the hyperparameters values regarding each method. The name tags used in the following figures are also specified to ease the reading of the plots.

Table 4.2 – Optimization algorithms names and configurations.

Short name	Parameters	Fill-in strategy	Selection strategy
EGO	N/A	N/A	All dimensions selected
Dropout	$d_e = 5$	Mix	Full random
Det.	$\tau = 1/d$	Mix	Deterministic selection
Prob.	$d_e = 5$	Mix	Probabilistic selection
Det. + Traj	$\tau = 1/d, T = 200$	Mix	Deterministic selection
Prob. + Traj	$d_e = 5, T = 200$	Mix	Probabilistic selection
Perm. 200	$s = 5\%$	Mix	Permutation-based selection
Perm. 1000	$s = 1\%$	Mix	Permutation-based selection

Four new analytic problems are considered: the Borehole function, the Ackley function, the Schwefel function and the Stybtang function, see Table 4.3. The first two are defined in $\mathbb{R}^{d_{\text{eff}}}$ and $d - d_{\text{eff}}$ dummy variables are added to simulate high dimensionality. The Schwefel and Stybtang functions are directly defined in $\mathbb{R}^d$ in order to characterize the behavior of the optimizers when the assumption of low effective dimension is not satisfied. All variables have a decreasing influence in the Schwefel function while all variables contribute equally to the Stybtang problem.

Table 4.3 – Test functions features. d_{eff} is the effective dimension of the function while d is the embedded high dimension with additional dummy variables.

Name	d_{eff}	d	Domain	Expression
Branin	2	25	$[-5, 10] \times [0, 15]$	$f(\mathbf{X}) = \left(X_2 - \frac{5.1}{4\pi^2}X_1^2 + \frac{5}{\pi}X_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(X_1) + 10$
Rosenbrock	5	20	$[-5, 10]^5$	$f(\mathbf{X}) = \sum_{i=1}^{d-1} 100\left(X_{i+1} - X_i^2\right)^2 + (X_i - 1)^2$
Borehole	8	25	$\mathcal{X}^B$	$f(\mathbf{X}) = \frac{2\pi X_3(X_4 - X_6)}{\ln(X_2/X_1)(1 + \frac{2X_7X_3}{\ln(X_2/X_1)X_1^2X_8} + \frac{X_3}{X_5})}$
Ackley	6	20	$[-3, 3]^6$	$f(\mathbf{X}) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d X_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^d \cos(2\pi X_i)\right) + 20 + \exp(1)$
Schwefel	20	20	$[-1, 1]^{20}$	$f(\mathbf{X}) = \sum_{i=1}^d \left(\sum_{j=1}^i X_j\right)^2$
Stybtang	20	20	$[-4, 4]^{20}$	$f(\mathbf{X}) = \frac{1}{2}\sum_{i=1}^d (X_i^4 - 16X_i^2 + 5X_i)$

with $\mathcal{X}^B = [0.05, 0.15] \times [100, 50000] \times [63070, 115600] \times [990, 1110] \times [63.1, 116] \times [700, 820] \times [1120, 1680] \times [9855, 12045]$

In the test campaign, we consider 20 different initial LHS optimized with a maximin criterion and the Gaussian process prior has a Matèrn 5/2 kernel created with the R package DiceKriging. The MMD is estimated with either the normalized version of Equation (4.46) or, when applicable, the normalized version of Equation (4.49), considering a Gaussian kernel for both cases.

In light of the normalized sensitivity maps of Figure 4.15, Section 4.4.1, the values of $\hat{q}$ and $\hat{q}'$, which define $\hat{\mathcal{D}}$ and $\hat{\mathcal{D}}'$, are adapted during the run. During a first phase of the optimization, the GP model is likely to be inaccurate and high performance points are often not yet known. Caution is necessary to account for these uncertainties so that the relevant variables are selected to go from any to a mild performance level: the algorithms start with the quantiles of the GP mean $\hat{q} = F^{-1}_{\mu(\mathbf{X})}(100\%)$ and $\hat{q}' = F^{-1}_{\mu(\mathbf{X})}(30\%)$. Once a fifth of the maximum budget is reached, some good performance points should have been located and more ambitious targets are set: $\hat{q} = F^{-1}_{\mu(\mathbf{X})}(30\%)$ and $\hat{q}' = F^{-1}_{\mu(\mathbf{X})}(5\%)$.

The acquisition function is always the expected improvement and it is optimized using the CMA-ES algorithm [HO01]. The budget for the optimization is limited to 100 iterations to match realistic expensive optimization tasks.

In the same spirit as [Han+16], the performance of the optimizers is assessed by measuring the frequency at which each algorithm is successful at solving tasks of varying difficulties. Three goals are set per function (easy, medium and hard to achieve). The number of successes at reaching a goal at a given iteration are counted during the repeated trials of each version of the algorithm. The performance thresholds are defined as 90%, 50% and 10% quantiles of the final results of all algorithms for each function (see Figure 4.20 for an example on the Rosenbrock function). We do so to provide a common basis for performance comparison since the global minimum of each function is not necessarily known.

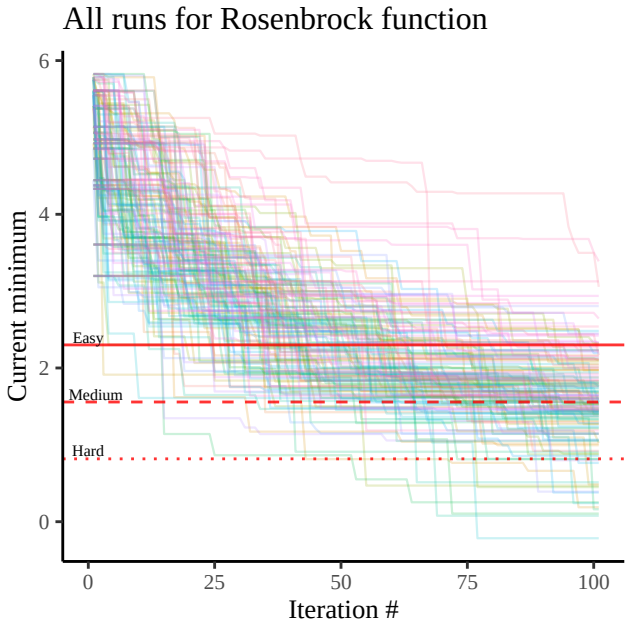

Figure 4.20 – All optimizer runs for the Rosenbrock function. The red lines (continuous, dashed and dotted) are respectively the 90%, 50% and 10% quantiles of the final results of all runs in log scale.

We first check how the different algorithms detect the presence of dummy variables. To this aim,

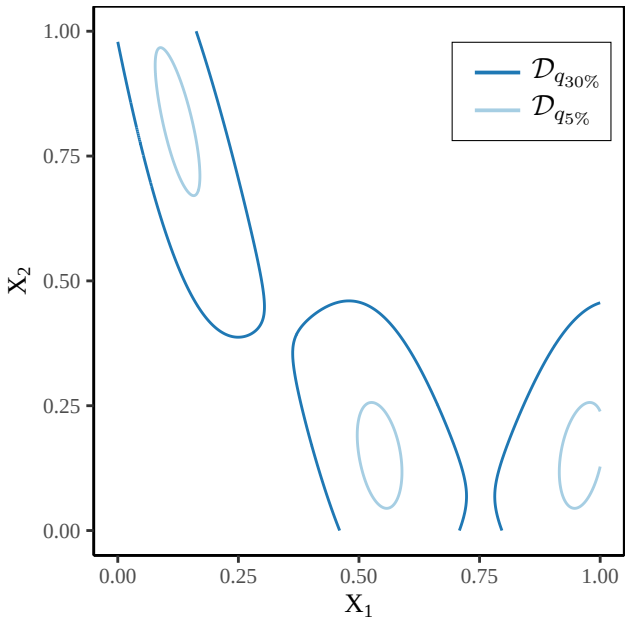

Figure 4.21 – Contours of the Branin function in $\mathbb{R}^{d_{\text{eff}}}$ for the 30% and 5% quantiles.

all the strategies are repeated 20 times on the Branin function with 23 added dummy variables. In Figure 4.22, the occurrence of selection of every variable at each iteration is averaged over the 20 runs. Two features are examined:

- The efficiency at selecting active variables: for the Branin function, Figure 4.21 shows that $\mathcal{D}_{q_{30\%}}$ contains almost all possible values for X_1 making that variable less likely to be selected by the indices while X_2 is more important due to the two minima around $X_2 = 0.2$. This phenomenon is lessened for $\mathcal{D}_{q_{5\%}}$. Both deterministic versions are able to consistently select the true inputs throughout the iterations. The average occurrence of selection is lower for other methods, with the non-parametric selection reaching between 60% and 80% and the probabilistic selection about 40% for the first variable.

- The efficiency at disregarding dummy variables: it is important not to keep dummy variables since maximizing the acquisition function gets harder as the dimension of X_d grows. By construction, the probabilistic methods keep a fixed number of variables at each iteration, a number which is set to 2 for the Branin function. The Deterministic selection, when done without simulations, keeps dummy variables much more often because of the approximation bias from the GP mean. The non-parametric strategies achieve about the same efficiency as the deterministic selection using simulations.

In Figure 4.22, there is also a visible increase in the selection rate of X_1 from the 30th iteration onward. It corresponds to the change in the targeted level sets and shows that X_1 is more important to finely optimize Branin than to reach a fair (30% quantile) level set.

Figure 4.23 shows the rate of success for KSA-BO algorithms with the various selection methods averaged over the complete test bed for the easy, medium and hard goals from left to right, respectively. It is compared to the Dropout approach and to a classical Bayesian optimization. The most noticeable result is how the Dropout underperforms even for the easy target, confirming the need for ways to choose optimization variables that are more efficient than a

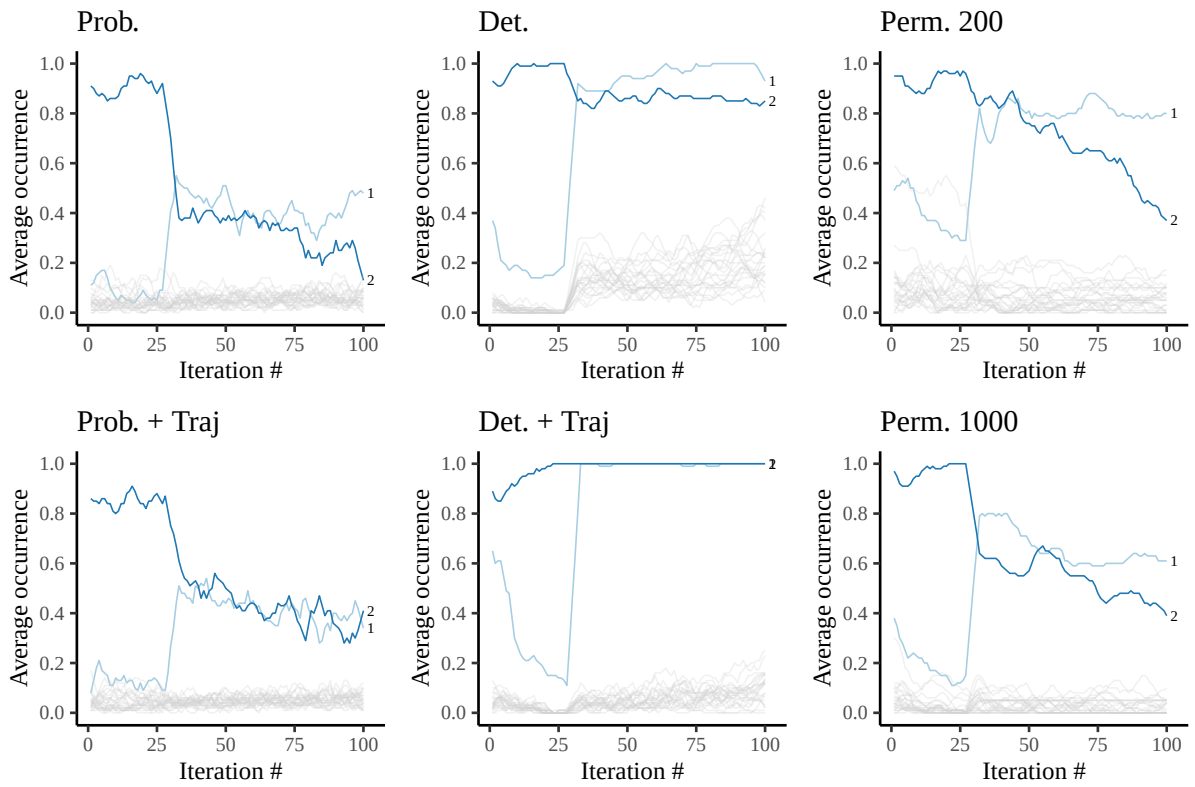

Figure 4.22 – Average selection of occurrence for each variable for the Branin-25d function. The results are smoothed using a moving average with a 5 iterations window size. Only the first two variables are annotated since they correspond to non-dummy inputs for this problem.

random pick. Additionally, the poor median performance of the EGO for medium and hard targets proves that reducing the dimension allows visible gains in term of minimum obtained for a limited budget. This corroborates what was observed earlier in the smaller benchmark of Section 4.3.

The methods using conditional simulations for computation of the sensitivity indices yield better results than the plug-in GP mean estimator: because GP model errors are accounted for, the variables are more reliably selected (which was observed in details in Figure 4.22 on the Branin function).

The strategy based on the non-parametric test outperforms other methods in the first iterations for every targets. This comes from the larger flexibility of these methods for selecting various number of variables. It can be understood in the examples of Figure 4.24 and Figure 4.25: on the Borehole function, at iteration 30 the deterministic selection only keeps X_1 when the test selection keeps X_1, and X_4 to X_8 (X_1 to X_8 are active). The deterministic selection can yield any number of variables, but in a parametric (because of the threshold τ) way. The probabilistic selections need an *a priori* number of active variables.

Overall, the deterministic selection using trajectories has the best results for both medium and hard targets at 100 iterations. The deterministic method is the most selective: the normalization will take the smaller sensitivities below the threshold $\tau = 1/d$. An example is given in Figure 4.24. When GP model errors are disregarded, such strong selection tends to induce premature convergence to false solutions. Considering GP simulations helps reducing the rate

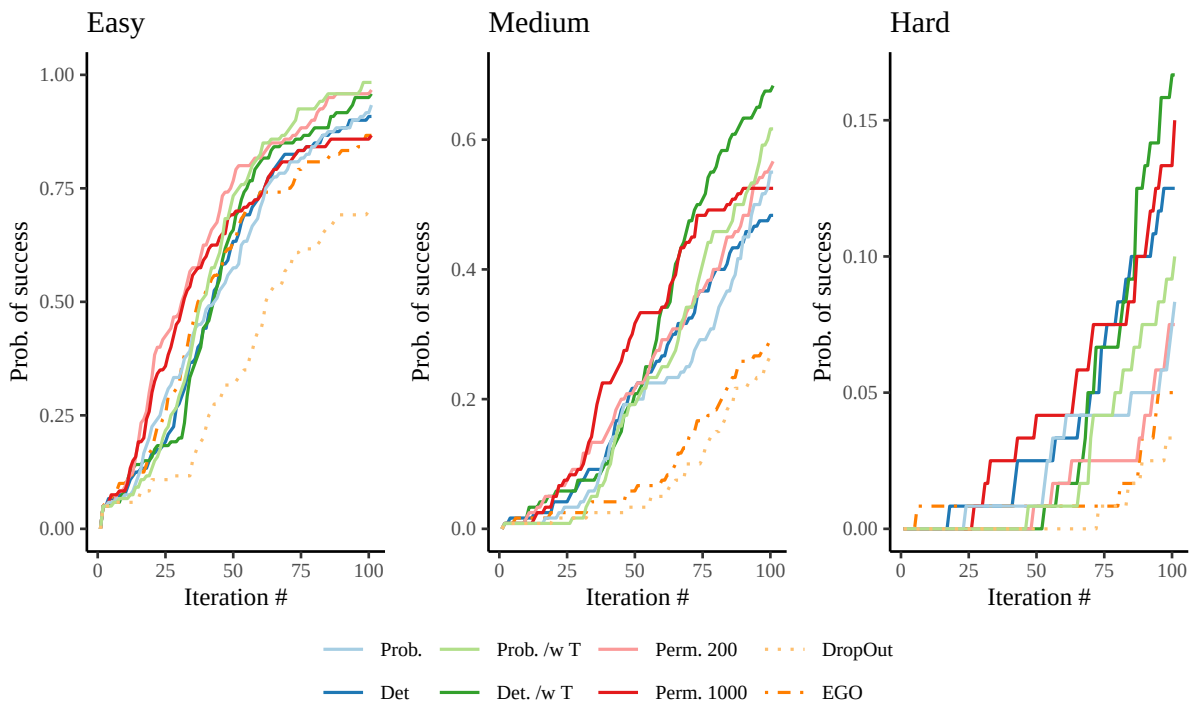

Figure 4.23 – Summary of the average rate of success on all the benchmark functions of each algorithm for the Easy, Medium and Hard goals, the higher the better.

at which these false convergences occur.

Figure 4.26 illustrates on the Borehole function how deterministic methods can get stuck in local minima much more easily than others. In this case, X_8 is missed in the first step of the search when the $q' = 30\%$ quantile level is targeted. Changing the level sets to $q = 30\%$, $q' = 5\%$ unblocks the situation and convergence to the global optimum is recovered. Note that the premature convergence of deterministic versions of KSA-BO happens despite the Mix approach to filling in inactive variables. This shows that the episodic random search on inactive variables is inefficient. This phenomenon is also visible for the easy target in Figure 4.23 with a premature convergence of the deterministic approach (especially the version using trajectories).

An interesting result is also seen in Figure 4.27, for functions devoid of dummy variables. Such problems are interesting because they violate the main assumption of the low effective dimension. In Figure 4.27, the classical Bayesian optimization and the Dropout strategies show poor performance compared to other methods. With the Stybtang function, all variables equally contribute to the output, as seen in Figure 4.28. Because of this, probabilistic methods are limited since they only pick a fixed number of variables at each iteration. For the Schwefel function, variables have a decreasing effect (meaning that X_1 is more influential than X_2 and so forth), shown in Figure 4.29. In this case, a method limited to a fixed number of variables picked at each iteration can achieve a good performance. However, once again for both functions, the best strategy is the deterministic selection with conditional simulations.

4.5 Conclusions

In this chapter, kernel based sensitivity analysis was included with in a Bayesian optimization to restrict the volume of the space searched at each iteration. Indeed, because the classical

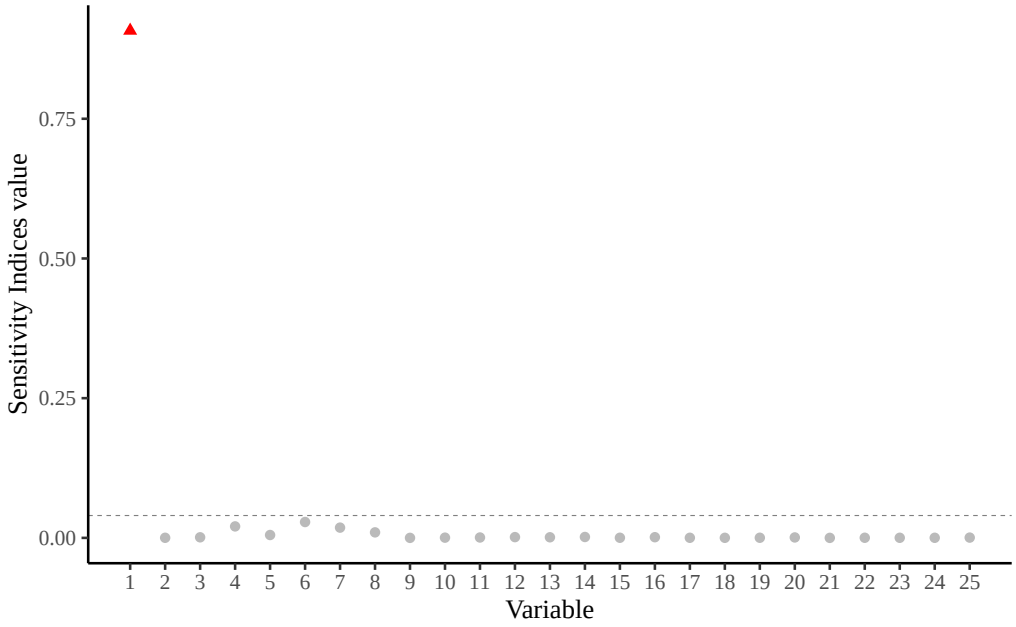

Figure 4.24 – Sensitivity indices values at the 30th iteration on the Borehole function with added dummy variables. The dashed line corresponds to $\tau = 1/d$, the threshold for detection in the *Deterministic* strategy, for which only variables X_1 would have been selected. X_1 to X_8 are active.

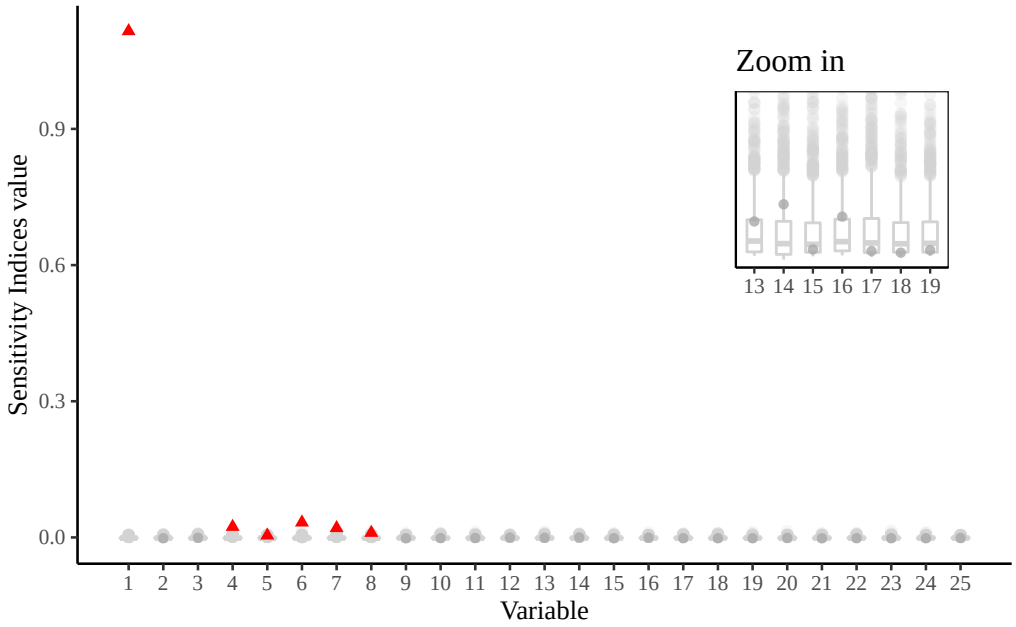

Figure 4.25 – Sensitivity indices values at the 30th iteration on the Borehole function with added dummy variables. Selected variables with the *Non-parametric* strategy are red triangles, X_1 to X_8 are active, while non-selected variables are grey dots. The permuted indices sampled under the null hypothesis are represented by boxplots, with a zoom-in view in the top-right corner for variables X_{13} to X_{19}.

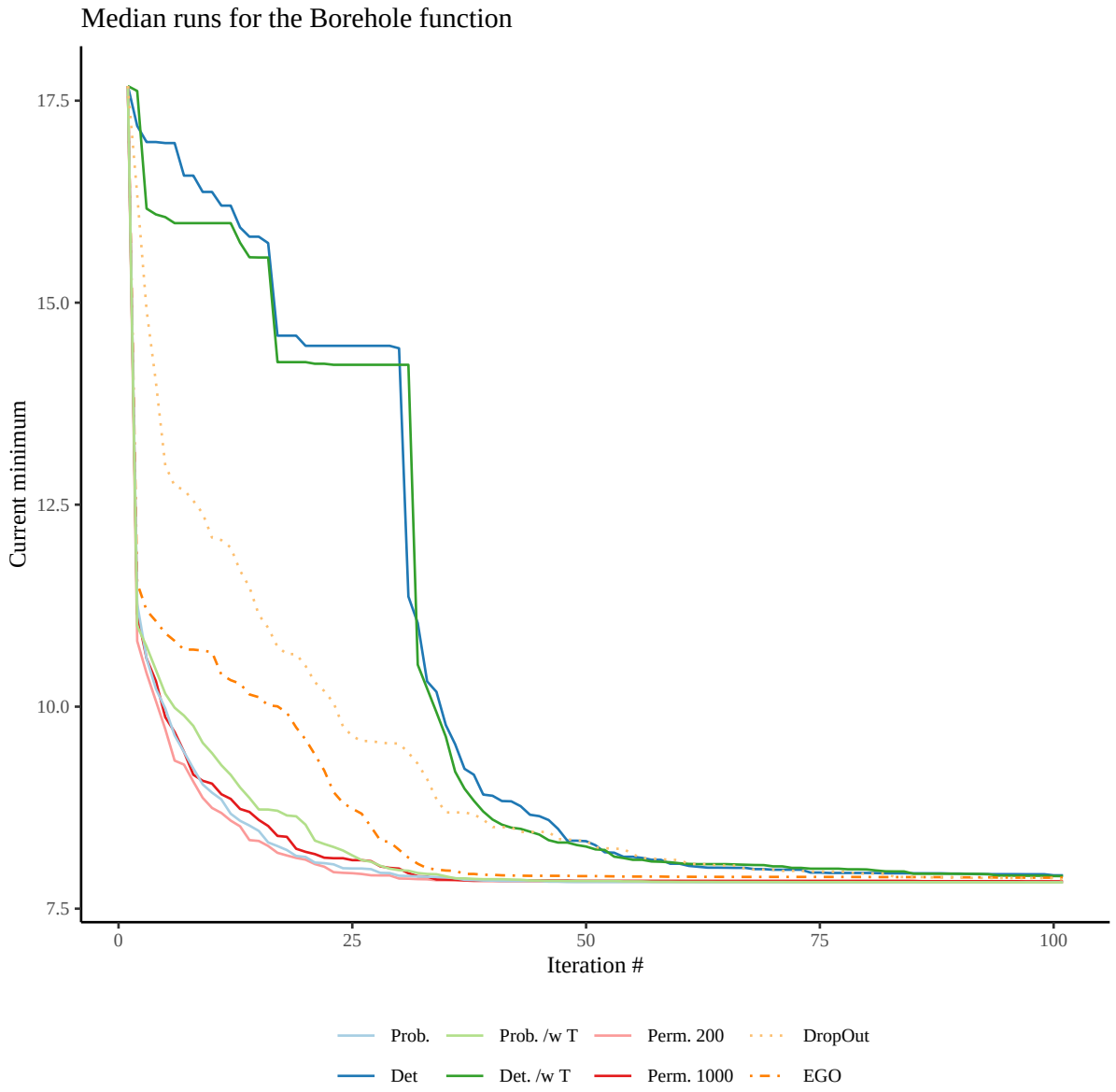

Figure 4.26 – Median best objective function for each method on the Borehole function. The deterministic strategies (with and without GP simulations) converge to a false solution in the first phase (targeting the 30% level set). Changing the thresholds allows to detect again the missed variables and to catch up with the other methods.

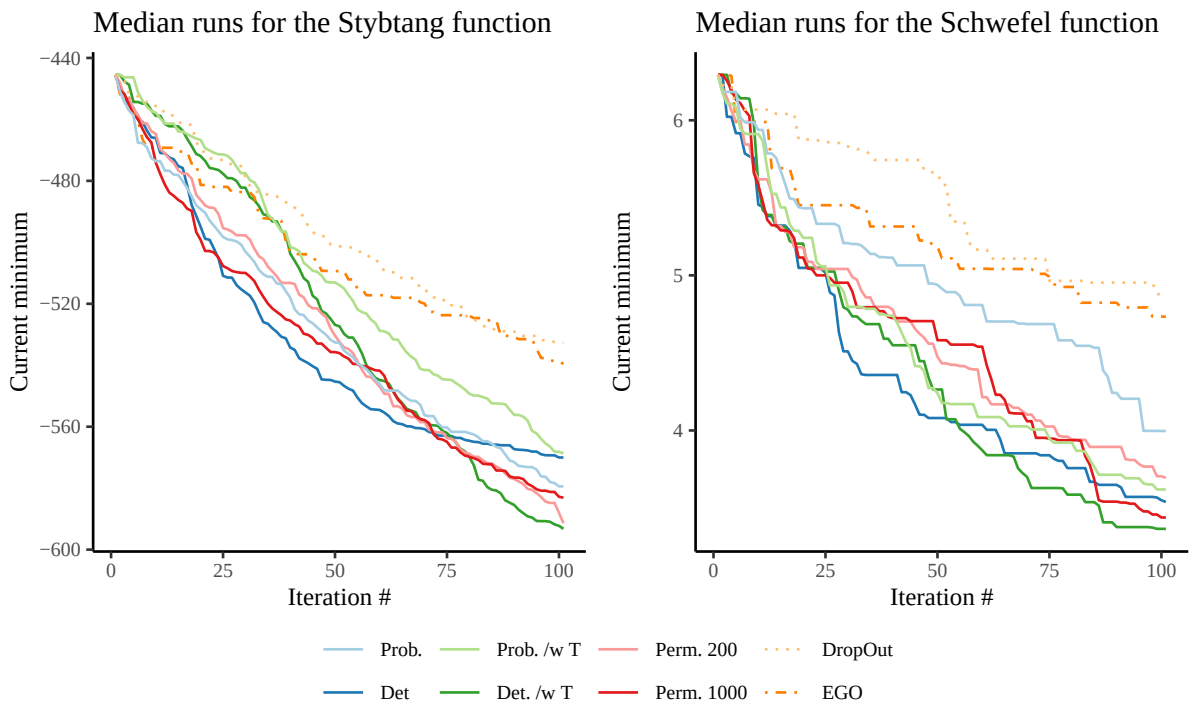

Figure 4.27 – Median objective function for each method on the Stybtang and the Schwefel functions, defined for $d = 20$ without any dummy variables.

Bayesian optimization methods can sample everywhere in the volume of the search space, they perform poorly in high-dimensional problems. Variable selection allows to restrict the volume space of search and eases the optimization.

Two strategies, one deterministic and one probabilistic, were first proposed to select the active variables, instead of choosing them randomly as is customary in the Dropout approach. The kernel-based indices adapted for optimization problems introduced in the Chapter 3 help to determine at each iteration whether a variable is fixed or optimized. Multiple approaches are considered for dropped out dimensions, with different inherent degrees of randomness.

A hyperparameter free approach was also defined. It is based on a non-parametric statistical test. There is an efficient way to compute it which relies on random selections of elements in the already assembled Gram matrix. Furthermore, since indices are estimated using Gaussian processes, indices calculated on conditional trajectories were also proposed as a way to account for model error. Finally, heuristics on the thresholds to consider in the definition of the sublevels of interest $\mathcal{D}$ and $\mathcal{D}'$, a key parameter in the kernel-based indices, were also derived.

All the approaches were tested on a benchmark of test functions. Some of the functions did not respect the low effective dimensionality assumption. Overall, a good selection of the dropped out dimensions shows a clear progress compared to random selection and compared to a generic Bayesian optimization. Deterministic approaches which select variables with GP simulations provide the best results on the complete set of test functions.

Finally, an extension of the methods proposed here to constrained optimization problems is straightforward. Indeed, it only requires to incorporate constraints in the definition of the sublevel sets of interest (like in Chapter 3). Gaussian processes can also be used to approximate the constraint functions if they are expensive to compute.

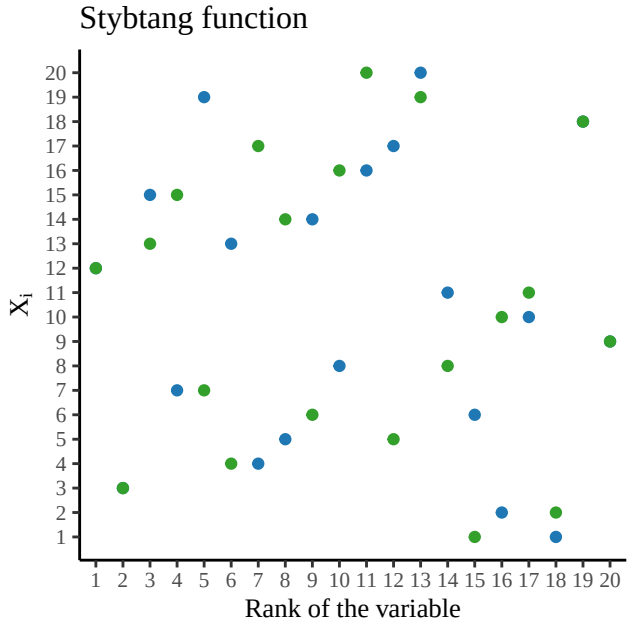

Figure 4.28 – Variable ranking at the 30th (blue dots) and last iterations (green dots) based on cumulative occurrence for the Stybtang function. If only one dot is visible, the variable ranked the same at both iterations.

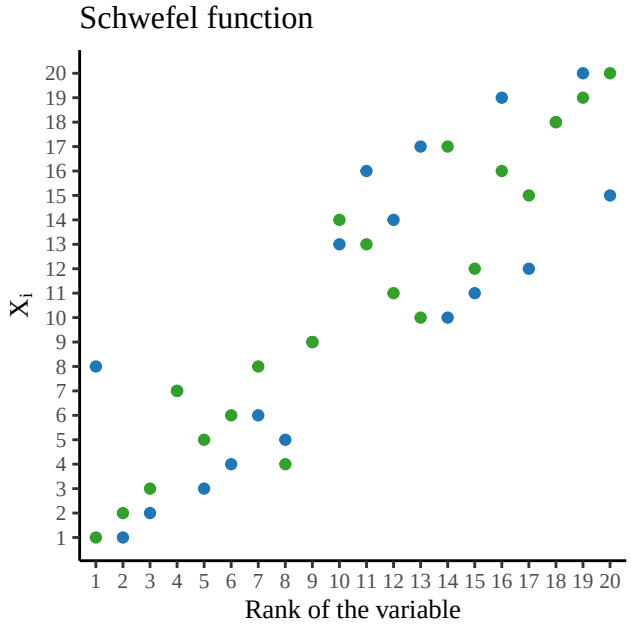

Figure 4.29 – Variable ranking at the 30th (blue dots) and last iterations (green dots) based on cumulative occurrence for the Schwefel function. If only one dot is visible, the variable ranked the same at both iterations.

Chapter take-home messages

- Sensitivity analysis helps to overcome Bayesian optimization limitations with high-dimensional problems
- Three strategies involving kernel-based indices were introduced to dropout variables, each depending on a single hyperparameter (with a proposed heuristic value)
- Sensitivity analysis leads to clear progress over random selection and classical Bayesian optimization

Bibliography

[Ber+11] James S Bergstra et al. "Algorithms for hyper-parameter optimization". In: *Advances in neural information processing systems*. 2011, pp. 2546–2554.

[BK10] Rémi Bardenet and Balázs Kégl. "Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm". In: *27th International Conference on Machine Learning (ICML 2010)*. Omnipress. 2010, pp. 55–62.

[BS+19] Malek Ben Salem et al. "Gaussian Process-Based Dimension Reduction for Goal-Oriented Sequential Design". In: *SIAM/ASA Journal on Uncertainty Quantification* 7.4 (2019), pp. 1369–1397.

[Dea+15] Angela Dean et al. *Handbook of design and analysis of experiments*. Vol. 7. CRC Press, 2015.

[DNR11] David K Duvenaud, Hannes Nickisch, and Carl E Rasmussen. "Additive gaussian processes". In: *Advances in neural information processing systems*. 2011, pp. 226–234.

[Gar+17] Jacob Gardner et al. "Discovering and exploiting additive structure for Bayesian optimization". In: *Artificial Intelligence and Statistics*. 2017, pp. 1311–1319.

[Han+16] Nikolaus Hansen et al. "COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting". ArXiv e-prints, arXiv:1603.08785. July 2016. URL: https://hal.inria.fr/hal-01294124.

[Has17] Trevor J Hastie. "Generalized additive models". In: *Statistical models in S*. Routledge, 2017, pp. 249–307.

[HBF11] Matthew D Hoffman, Eric Brochu, and Nando de Freitas. "Portfolio Allocation for Bayesian Optimization." In: *UAI*. Citeseer. 2011, pp. 327–336.

[HO01] Nikolaus Hansen and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies". In: *Evolutionary computation* 9.2 (2001), pp. 159–195.

[Jon01] Donald R Jones. "A taxonomy of global optimization methods based on response surfaces". In: *Journal of global optimization* 21.4 (2001), pp. 345–383.

[JPS93] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. "Lipschitzian optimization without the Lipschitz constant". In: *Journal of optimization Theory and Applications* 79.1 (1993), pp. 157–181.

[JSW98] Donald R Jones, Matthias Schonlau, and William J Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13.4 (1998), pp. 455–492.

[Kle87] Jack PC Kleijnen. *Statistical tools for simulation practitioners*. Marcel Dekker, 1987.

[KSP15] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. "High dimensional Bayesian optimisation and bandits via additive models". In: *International Conference on Machine Learning*. 2015, pp. 295–304.

[Kus64] Harold J Kushner. "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise". In: *Journal of Basic Engineering* 86.1 (1964), pp. 97–106.

[Li+17] Cheng Li et al. "High dimensional bayesian optimization using dropout". In: *IJCAI 2017: Proceedings of the 26th International Joint Confrerence on Artificial Intelligence*. [The Conference]. 2017, pp. 2096–2102.

[Liz08] Daniel James Lizotte. *Practical bayesian optimization*. University of Alberta, 2008.

[LN89] Dong C Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical programming* 45.1-3 (1989), pp. 503–528.

[Mat13] Bertil Matérn. *Spatial variation*. Vol. 36. Springer Science & Business Media, 2013.

[Mat73] Georges Matheron. "The intrinsic random functions and their applications". In: *Advances in applied probability* 5.3 (1973), pp. 439–468.

[MBC79] Michael D McKay, Richard J Beckman, and William J Conover. "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code". In: *Technometrics* 21.2 (1979), pp. 239–245.

[MTZ78] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. "The application of Bayesian methods for seeking the extremum". In: *Towards global optimization* 2.117-129 (1978), p. 2.

[Nie92] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Vol. 63. Siam, 1992.

[O'H78] Anthony O'Hagan. "On curve fitting and optimal design for regression". In: *J. Royal Stat. Soc. B* 40 (1978), pp. 1–32.

[PB01] Jeong-Soo Park and Jangsun Baek. "Efficient computation of maximum likelihood estimators in a spatial linear model with power exponential covariogram". In: *Computers & Geosciences* 27.1 (2001), pp. 1–7.

[Pro17] Luc Pronzato. "Minimax and maximin space-filling designs: some properties and methods for construction". In: (2017).

[Sha+15] Bobak Shahriari et al. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175.

[SLRDV19] Adrien Spagnol, Rodolphe Le Riche, and Sébastien Da Veiga. "Bayesian optimization in effective dimensions via kernel-based sensitivity indices". In: *13th International Conference on Applications of Statistics and Probability in Civil Engineering(ICASP13)*. Seoul National University. 2019.

[Sri+10] Niranjan Srinivas et al. "Gaussian process optimization in the bandit setting: no regret and experimental design". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress. 2010, pp. 1015–1022.

[Sri+14] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[SW10] Songqing Shan and G Gary Wang. "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive

black-box functions". In: *Structural and Multidisciplinary Optimization* 41.2 (2010), pp. 219–241.

[Ulm+16] Doniyor Ulmasov et al. "Bayesian optimization with dimension scheduling: Application to biological systems". In: *Computer Aided Chemical Engineering*. Vol. 38. Elsevier, 2016, pp. 1051–1056.

[Wan+17] Zi Wang et al. "Batched high-dimensional bayesian optimization via structural kernel learning". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 3656–3664.

[WR06] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[Žil80] Antanas Žilinskas. "On the use of statistical models of multimodal functions for the construction of the optimization algorithms". In: *Optimization techniques*. Springer, 1980, pp. 138–147.

Chapter 5

Conclusions and perspectives

Contents

5.1 Summary of the main contributions

This thesis focuses on variables selection for the optimization of black-box functions. It is motivated by industrial applications.

Chapter 3 provides a framework to characterize variable influence on the performance of the output. This is done using a quantity of interest that considers areas of the input space which produce valuable function results, in terms of both, objective function and constraints satisfaction. We denote such an area as the *sublevel set of interest* $\mathcal{D}$ and give guidelines for its definition. Building on the definition of $\mathcal{D}$, we introduce kernel-based sensitivity indices, which are an independence measure in a reproducing kernel Hilbert space called the Hilbert Schmidt independence criterion, HSIC. The HSIC measures the independence between the random variable $(X_i \mid \mathbf{X} \in \mathcal{D})$ and the random variable "the output reaches the target". The connection between the HSIC and a distance between the kernel mean embeddings of the random variables X_i and $(X_i \mid \mathbf{X} \in \mathcal{D})$ is clarified. The progression from the HSIC to the mean embeddings distance follows the same route as previous goal-oriented sensitivity approaches [YHS78; Luy+12] This embeddings point of view (also known as MMD for maximum mean discrepancy [Gre+05]) provides an intuitive understanding of how the sensitivity index measures the influence of an input. It is a distance between the distributions of X_i and $(X_i \mid \mathbf{X} \in \mathcal{D})$.

An algorithm is introduced, with a sensitivity analysis step prior to an optimization process. It is tested on several functions, considering different off-the-shelf optimizers suited for constrained problems, namely COBYLA and SQP. By removing non influential variables, the volume of the search space is reduced and finding an optimal solution is easier. Of course, one has to keep in mind that the uncovered solution does not correspond to the true optimum since some fine-tuning was lost when fixing some of the variables. However, with real life applications, acquiring a solution more easily and faster is beneficial even at the cost of a slight performance decrease.

Surrogate-based optimization make for an appealing set of algorithms for expensive black-box functions but these algorithms suffer from the curse of dimensionality. Multiple strategies were developed in the literature to overcome this issue, mainly through assumptions about the objective function structure (e.g., additivity) or assumptions about the effective dimension of the function. In Chapter 4, we drastically improve one of these methods, the Dropout algorithm for high-dimensional Bayesian optimization. At each iteration, a Bayesian optimizer with dropout randomly selects the subset of active variables that are searched for by maximizing the acquisition function. As this function typically has flat areas between several maxima, reducing the search space to a lower number of variables makes the optimization significantly easier. Instead of selecting variables randomly, we guide the choice with kernel-based sensitivity indices and therefore propose a new Bayesian optimization algorithm called KSA-BO for Kernel Sensitivity Analysis for Bayesian Optimization. Two options to discard or not variables based on their sensitivity are defined and investigated: one is deterministic (all variables whose sensitivity is above a given threshold are active), the other probabilistic (pick active variables proportionally to their sensitivity). Complementary approaches to set the values of variables at dropped out dimensions are also considered. The accuracy of the selection is demonstrated on multiple test cases with dummy variables, which are correctly detected by our sensitivity indices. Furthermore, both selection methods lead to better optima at a fixed budget than the Dropout and the classical Bayesian optimizations. Finally, we propose three improvements to the KSA-BO algorithm. First, a parameter free selection is defined that builds on a non-parametric statistical test. Then, taking full advantage of the Gaussian process, a sensitivity index that accounts

for GP model errors is obtained by averaging sensitivities over several conditional trajectories instead of relying solely on the GP mean. Using this mean sensitivity index leads to less biased indices and results in better optimization convergence. The best BO algorithm among those tested is made of the deterministic selection with GP simulations and the mixed random/best observed fix for the non-selected variables.

5.2 Perspectives

Several perspectives and possible improvements of the aforementioned methods can be envisioned.

Smarter level set choice In Section 4.4.1, varying the targeted level set values between iterations is proposed, allowing to detect different subsets of variables depending on the pair of levels $(\mathcal{D}, \mathcal{D}')$ considered. As a rule of thumb, only few variables have an impact on higher levels of the function, while it is often the interactions of the full set of variables that matter close to the optimum. However, the choice of the levels in this thesis is empirically based on observations on simple examples. The algorithm would benefit from an automatic selection of the thresholds at each iteration, taking into account the level of performance already achieved by the surrogate model.

Analytic expression for conditional trajectories In Section 4.4.2, conditional trajectories enable to use more information from the Gaussian process predictor than simply its mean. An analytic expression of an average over multiple conditional simulations is also provided. However, it leads to the expectation of a ratio that, when derived, involves difficult computations with several nested loops and is therefore impractical. By considering a different formulation for the quantity of interest instead of $1_{\mathbf{X}\in\mathcal{D}}$, with for example a sigmoid function to replace the indicator function, we could replace the expectation of a ratio by a single expectation with a single loop estimation. This would result in a fast computation of the indices.

Asymptotic distributions The permutation-based approach for the variable selection requires to compute a p-value in order to characterize whether an input is influential, which is attested by rejecting the null hypothesis. For high significance levels, the number of samples needed to approximate this p-value becomes large. A promising alternative would be to directly obtain and use a quantile of the asymptotic distribution as the detection threshold for the hypothesis testing. This is possible for the linear unbiased estimator of the maximum mean discrepancy since the null distribution converges to a Gaussian, see Section 2.2.3, but estimating the true kernel-based indices with such estimator would requires too many samples and is therefore not well-suited in our optimization set-up. Another estimator, called the *Block-tests* or *B-tests* [ZGB13], splits the data into multiple blocks, computes the quadratic maximum mean discrepancy on each and averages the resulting statistics. Its asymptotic distribution is Gaussian under some mild assumptions. More recently, an estimator based on an incomplete U-statistics [Yam+18] was proposed, with the property of being asymptotically Gaussian under the null hypothesis and could also be considered.

Improving REMBO Chapter 4 introduces an improvement of the Dropout method designed to deal with high-dimensional optimization problem. The same kind of variable selection could be applied to linear combinations of the original variables, $\mathbf{X}' = A\mathbf{X}$, A matrix of the coefficients

of the linear combination. Such an approach could be seen as a REMBO optimization (cf. Section 4.2.2) guided by sensitivity analysis.

Sensitivity indices for optimization under uncertainty Real-life applications often deal with a set of deterministic inputs $\mathbf{X}$ and a set of randomized inputs ξ. Optimizing models with both types of inputs is known as *optimization under uncertainty*. However, most of the time for uncertain inputs, influence is measured by looking at how a certain quantity of the output distribution, such as the mean or a quantile, changes when a given ξ is set as deterministic. Multiple approaches could be considered when using the maximum mean discrepancy, simply since it considers the full distribution of $f(\mathbf{X}, \xi)$ and does not require any choice of a statistical measure. For example, one could try to find the subset $\xi_{\mathcal{I}}$ that minimizes $\gamma^2(P_{f(\mathbf{X},\xi)}, P_{f(\mathbf{X},\xi_{\mathcal{I}})})$, under the constraints that at least a few parameters are removed since keeping all parameters is a trivial solution. The influence of the deterministic inputs on the performance of the output could possibly be assessed afterward with the methods introduced in this thesis.

Bibliography

[Gre+05] Arthur Gretton et al. "Kernel methods for measuring independence". In: *Journal of Machine Learning Research* 6.Dec (2005), pp. 2075–2129.

[Luy+12] Li Luyi et al. "Moment-independent importance measure of basic variable and its state dependent parameter solution". In: *Structural Safety* 38 (2012), pp. 40–47.

[Yam+18] Makoto Yamada et al. *Post Selection Inference with Incomplete Maximum Mean Discrepancy Estimator*. 2018. arXiv: 1802.06226 [stat.ML].

[YHS78] Peter C Young, George M Hornberger, and Robert C Spear. "Modeling badly defined systems: some further thoughts". In: *Proceedings SIMSIG Conference*. Australian National University Canberra. 1978, pp. 24–32.

[ZGB13] Wojciech Zaremba, Arthur Gretton, and Matthew Blaschko. "B-test: A non-parametric, low variance kernel two-sample test". In: *Advances in neural information processing systems*. 2013, pp. 755–763.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT: 2020LYSEM012

Author: Adrien Spagnol

Title: Kernel-based sensitivity indices for high-dimensional optimization problems

Speciality: Applied Mathematics

Keywords: Sensivitiy analysis, High dimensional, Global Optimization, Bayezian optimization, Dimension reduction

This thesis treats the optimization under constraints of high-dimensional black-box problems. Common in industrial applications, they frequently have an expensive associated cost which make most of the off-the-shelf techniques impractical. In order to come back to a tractable setup, the dimension of the problem is often reduced using different techniques such as sensitivity analysis. A novel sensitivity index is proposed in this work to distinct influential and negligible subsets of inputs in order to obtain a more tractable problem by solely working with the primer. Our index, relying on the Hilbert Schmidt independence criterion, provides an insight on the impact of a variable on the performance of the output or constraints satisfaction, key information in our study setting. Besides assessing which inputs are influential, several strategies are proposed to deal with negligible parameters. Furthermore, expensive industrial applications are often replaced by cheap surrogate models and optimized in a sequential manner. In order to circumvent the limitations due to the high number of parameters, also known as the curse of dimensionality, we introduce in this thesis an extension of the surrogated-based optimization. Thanks to the aforementioned new sensitivity indices, parameters are detected at each iteration and the optimization is conducted in a reduced space.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT: 2020LYSEM012

Auteur: Adrien Spagnol

Titre: Indices de sensibilité via des méthodes à noyaux pour des problèmes d'optimisation en grande dimension

Spécialité: Mathématiques appliquées

Mots-Clefs: Analyse de sensibilité, Grande dimension, Optimisation globale, Optimisation Bayésienne, Réduction de dimensions

Cette thèse s'intéresse à l'optimisation sous contraintes de problèmes type "boite-noire" en grande dimension. Répandus dans les applications industrielles, elles ont généralement un coût élevé ce qui empêchent d'utiliser la plupart des méthodes d'optimisation classiques. Afin de résoudre ces problèmes, la dimension de celui-ci est souvent réduite via différentes techniques telle que l'analyse de sensibilité. Un nouvel indice de sensibilité est proposé dans ces travaux afin de distinguer quelles sont les entrées du problèmes influentes et celles négligeables et d'obtenir un problème simplifié n'incluant que les premières. Notre indice, reposant sur le critère d'indépendance d'Hilbert Schmidt, fournit une connaissance de l'impact d'une variable sur la performance de la sortie ou le respect des contraintes, des aspects primordiaux dans notre cadre d'étude. Outre la caractérisation des variables influentes, plusieurs stratégies sont proposées pour traiter les paramètres négligeables. De plus, les applications industrielles coûteuses sont généralement remplacés par des modèles proxys moins coûteux qui sont optimisés de manière séquentielle. Afin de contourner les limitations dues au nombre élevé de paramètres, aussi connu sous le nom de fléau de la dimension, une extension de l'optimisation basée sur des métamodèles est proposée dans cette thèse. Grâce aux nouveaux indices de sensibilités susmentionnés, les paramètres influents sont détectés à chaque itération et l'optimisation est effectuée dans un espace de dimension inférieure.

ÉCOLE DES MINES DE SAINT-ÉTIENNE
158, cours Fauriel – CS 62362
42023 Saint-Étienne Cedex 2 – France
Tél. +33(0)4 77 42 01 23
www.mines-stetienne.fr