Agnostic Feature Selection

Guillaume Florent Doquet

université PARIS-SACLAY

NNT : 20XXSACLXXXX

Thèse de doctorat

UNIVERSITÉ PARIS SUD
Comprendre le monde, construire l'avenir

Agnostic Feature Selection

Thèse de doctorat de l'Université Paris-Saclay
préparée à Université Paris-Sud

Ecole doctorale n°580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Gif-sur-Yvette, le 29/11/19, par

GUILLAUME DOQUET

Composition du Jury :

Jérémie MARY Senior Researcher, Criteo	Rapporteur
Mohamed NADIF Professeur, Université Paris-Descartes	Rapporteur
Anne VILNAT Professeur, Université Paris-Sud	Président du jury
Gilles GASSO Professeur, INSA Rouen	Examinateur
Amaury HABRARD Professeur, Université de St-Etienne	Examinateur
Michèle SEBAG DR1, Université Paris-Sud	Directeur de thèse

Abstract

With the advent of Big Data, databases whose size far exceed the human scale are becoming increasingly common. The resulting overabundance of monitored variables (friends on a social network, movies watched, nucleotides coding the DNA, monetary transactions...) has motivated the development of Dimensionality Reduction (DR) techniques. A DR algorithm such as Principal Component Analysis (PCA) or an AutoEncoder typically combines the original variables into new features fewer in number, such that most of the information in the dataset is conveyed by the extracted feature set.

A particular subcategory of DR is formed by Feature Selection (FS) methods, which directly retain the most important initial variables. How to select the best candidates is a hot topic at the crossroad of statistics and Machine Learning. Feature importance is usually inferred in a supervised context, where variables are ranked according to their usefulness for predicting a specific target feature.

The present thesis focuses on the *unsupervised* context in FS, *i.e.* the challenging situation where no prediction goal is available to help assess feature relevance. Instead, unsupervised FS algorithms usually build an artificial classification goal and rank features based on their helpfulness for predicting this new target, thus falling back on the supervised context. Additionally, the efficiency of unsupervised FS approaches is typically also assessed in a supervised setting.

In this work, we propose an alternate model combining unsupervised FS with data compression. Our *Agnostic Feature Selection* (AGNOS) algorithm does not rely on creating an artificial target and aims to retain a feature subset sufficient to recover the whole original dataset, rather than a specific variable. As a result, AGNOS does not suffer from the selection bias inherent to clustering-based techniques.

The second contribution of this work [1] is to establish both the brittleness of the standard supervised evaluation of unsupervised FS, and the stability of the new proposed AGNOS.

1. *Agnostic Feature Selection*, G. Doquet and M. Sebag, ECML PKDD 2019

Résumé

Les bases de données dont la taille dépasse largement l'échelle humaine sont de plus en plus courantes. La surabondance de variables considérées qui en résulte (amis sur un réseau social, films regardés, nucléotides codant l'ADN, transactions monétaires...) a motivé le développement des techniques de réduction de dimensionalité (DR).

Une sous-catégorie particulière de DR est formée par les méthodes de sélection d'attributs (SA), qui conservent directement les variables initiales les plus importantes. La manière de sélectionner les meilleurs candidats est un sujet d'actualité à la croisée des chemins entre statistiques et apprentissage automatique. L'importance des attributs est généralement déduite dans un contexte supervisé, où les variables sont classées en fonction de leur utilité pour prédire une variable cible spécifique.

Cette thèse porte sur le contexte non supervisé de la SA, c'est-à-dire la situation épineuse où aucun objectif de prédiction n'est disponible pour évaluer la pertinence des attributs. Au lieu de cela, les algorithmes de SA non supervisés construisent généralement un objectif de classification artificiel et notent les attributs en fonction de leur utilité pour prédire cette nouvelle cible, se rabattant ainsi sur le contexte supervisé.

Dans ce travail, nous proposons un autre modèle combinant SA non supervisée et compression de données. Notre algorithme AGNOS (Agnostic Feature Selection) ne repose pas sur la création d'une cible artificielle, et vise à conserver un sous-ensemble d'attributs suffisant pour reconstruire l'intégralité des données d'origine, plutôt qu'une variable cible en particulier. Par conséquent, AGNOS ne souffre pas du biais de sélection inhérent aux techniques basées sur le clustering.

La seconde contribution de ce travail[2] est d'établir à la fois la fragilité du processus supervisé standard d'évaluation de la SA non supervisée ainsi que la stabilité du nouvel algorithme proposé AGNOS.

2. *Agnostic Feature Selection*, G. Doquet and M. Sebag, ECML PKDD 2019

Table des matières

Chapitre 1

Feature Selection in an unsupervised context

Data collection and processing have played an integral role in organized societies since antiquity, be it for recording taxes, managing cattle or organizing a military force. The advent of computers, however recent on the time scale of human History, has led to a paradigm shift ; data collection can now be performed *automatically* rather than manually. Additionally, dematerialized storage allows keeping record of unprecedented amounts of data.

Together, these two breakthroughs have led us to the *Information era*, in which data is gathered at an ever-growing rate. Consequently, a new challenge arose : the pace at which data is *collected* often vastly outscales the rate at which it can be *processed*. A notorious example of this issue is given by the CIA in the early 2000s, when the US federal agency accumulated large amounts of intelligence but lacked the manpower to analyze even a fraction of it. Accordingly, a new branch of research and industry dedicated to tackling this class of problems appeared in the last decade, and was named *Big Data*.

Rather than allocating evermore computational power to data processing, a cost-efficient way of handling large databases is provided by *Dimensionality Reduction (DR)*. DR is informally defined as the process of compressing the information contained in the original high-dimensional dataset into a new data representation of lower dimension. Although DR predates computer science (Pearson, 1901), it has become an increasingly prevalent tool in view of Big Data.

1.1 Context of the thesis

The presented work is concerned with *automatic* DR. The compressed data representation is discovered with Machine Learning (ML). The approach will rely on the basics of neural networks (Haykin, 1994).

1.1.1 A need for interpretability

As ML becomes more and more popular in varied application fields such as medical research, financial market prediction or weather forecast, it is increasingly important to the end user of ML to be able to *make sense of* the learning results. The motivation underlying this *interpretability* requirement is threefold.

The first desired property is *fairness*, corresponding to the absence of malicious or unwanted bias in the algorithm output (O'neil, 2016). Fairness has been a hotly debated topic in the past few years, for example in the US legal system ; the dangerousness of defendants and convicts is evaluated through risk assessment algorithms, and it was soon found that African-American citizens were on average assigned higher risk scores than for other origins. The question remained, however, to determine whether this discrepancy was the consequence of racist discrimination or the byproduct of other correlated factors such as firearm possession (Skeem et al., 2016). Interpretability is therefore needed to assess the fairness of a learning agent.

The second desired property is *transparency*, which corresponds the understandability of a model. Transparency is especially useful in domains where Artificial Intelligence outperforms human experts, such as epilepsy prediction or the game of Go (Deepmind, 2019). In order for physicians to improve their own diagnoses or Go players their own skill, they need to understand *why* the ML algorithm makes certain decisions. In other words, interpretability is required for humans to learn from the machine.

The third desired property is *accountability*, such that there is clarity regarding who holds responsibility of the decisions made by the algorithm. The need for accountability was recently highlighted, with the first fatal accident involving a self-driving car happening in early 2018. Detailed analysis of the driving model is required in order to determine if the algorithm is faulty and if the car passenger is to blame. Moving forward, interpretability will thus become even more crucial, as ML algorithms become trusted with vital decisions.

Accordingly, the ML community as a whole is increasingly concerned with *Fair, Transparent and Accountable (FTA)* learning, for both research (Doshi-Velez and Kim, 2017) and industry (FTA, 2018) purposes. The aim of the presented work is thus to achieve FTA DR.

In order to perform efficient information compression, DR techniques typically produce new variables, thereafter called *features*, that are obtained from the original features *via* an arbitrarily complex mapping. Following this unconstrained functional complexity, the resulting features are in the general case hardly interpretable by humans, regardless of their expertise.

Consequently, this thesis is concerned with a particular case of DR called *Feature Selection (FS)*. Instead of producing new composite features, FS methods filter out the least promising original variables, retaining only the best candidates. The result of FS is therefore a subset containing interpretable features (assuming the original data was interpretable to begin with), such as e.g. $\{\text{age}, \text{smoker}\}$ for a lung cancer prediction task on medical data. FS is therefore more appropriate than generic DR (often called *Feature Construction (FC)*) to achieve FTA learning.

1.1.2 The unsupervised context

Three particular different ML settings are relevant to this work : *supervised*, *semi-supervised* and *unsupervised*. In the supervised case, the end goal of learning is fully known. Consider for instance a visual recognition task where the learning agent is presented with pictures of street traffic and tasked with identifying pedestrians, given that the human expert knows all the correct answers.

In the semi-supervised setting, only a few correct answers are available. This is the case of the lung cancer prediction task example. Physicians can tell that patients with visible symptoms are ill, but are uncertain about healthy-looking cases.

In the unsupervised setting, no ground truth is available at all. Furthermore, the final goal of learning is itself unknown. Using the previous illustrating example, this means the learning agent does not know whether the medical data will ultimately be leveraged for lung cancer prediction, breast cancer prediction, or possibly something else entirely.

One may then wonder what the purpose of unsupervised learning is. If we do not even know what we want ourselves, how could ML be of any help ? An answer to this questioning is provided by the following argument, made by G. Hinton in 1996 :

"When we're learning to see, nobody's telling us what the right answers are — we just look. Every so often, your mother says "that's a dog", but that's very little information. You'd be lucky if you got a few bits of information — even one bit per second — that way. The brain's visual system has 10^{14} neural connections. And you only live for 10^9 seconds. So it's no use learning one bit per second. You need more like 10^5 bits per second. And there's only one place you can get that much information : from the input itself".

In other words, life is *at best* a semi-supervised learning experience. Furthermore, supervised learning is very limited in terms of scalability due to the scarcity of ground truth information. Yet, cognitive mammals are able to learn quickly and efficiently using only their sensory inputs. This shows that unsupervised learning is actually a powerful tool, and is well-suited for real-world applications. The power and crucial importance of the unsupervised context has been further underlined by Y. LeCun (LeCun, 2016), essentially arguing that supervised learning is merely the tip of the iceberg, while unsupervised learning forms the submerged part.

Consequently, this thesis is mainly focused on *unsupervised* FS. This setting is however particularly challenging for the sake of interpretability, given the absence of a definitive goal shedding light on the results. For instance, the previous selection subset $\{\text{age}, \text{smoker}\}$ is easily understandable for lung cancer prediction purposes, but less so without knowledge about the objective.

In order to comply with the FTA learning requirement, defining a clear unsupervised goal is therefore of paramount importance. Designing such a goal and demonstrating its soundness will in the following prove to be a cornerstone of the presented work.

1.2 Motivation

As previously mentioned, DR is a useful tool in a wide range of applications, such as banking, genomics research, online advertising, power grids or video game development. DR is essentially required for any Big Data endeavor ; as soon as one is concerned with large amounts of input information, FS should at least be considered as a pre-processing option.

Furthermore, as the computational power of CPUs and GPUs keeps increasing, private companies and governments alike pursue collection of more and more data, in the hopes that their data processing capabilities will eventually cease to be a bottleneck.

Moreover, the recent advent of cloud computing is an additional incentive for indiscriminate data harvesting, given the resulting boost of storage capacity. A telltale example of this trend is provided by Walmart (the world's biggest retail company), owner of the largest private cloud on the planet, which is able to process 2.5

petabytes of consumer data *per hour*. This means that the company is recording millions of tidbits of information (features) about each of their clients, not all features being equally interesting. There is thus little doubt that DR algorithms, and more specifically FS methods, play a large role in the data processing pipeline.

Consequently, even though our contributions were so far applied only to comparatively much smaller real-world datasets (spanning medical data, image data and text data), we expect unsupervised FS to be of potential use in a wide array of different domains moving forward.

1.3 Main contributions and organization of the work

The first contribution presented in this manuscript is algorithmic. The goal is to design a method bridging the gap between FC and FS, leveraging the constructed variables to guide the selection.

The second contribution is methodological in nature, and pertains to the evaluation procedure of unsupervised FS. We claim and empirically show that the typical performance assessment scheme is unreliable, and propose a more adequate stable efficiency criterion to rely on instead.

1.3.1 Performing unsupervised FS

The most common approach consists of i) performing spectral clustering (**?**) to equip the datapoints with pseudo-labels ; ii) falling back on supervised techniques, select the features best able to predict the pseudo-labels. Still, the reliability and robustness of the clusters is not guaranteed. This manuscript investigates an alternate approach, called *Agnostic Feature Selection* (AGNOS), that does not rely on pseudo-labels.

Inspired by regularized regression (Tibshirani, 1996; Simon et al., 2013) and feature selection based on neural networks (Verikas and Bacauskiene, 2002; Roy et al., 2015; Li et al., 2016), the proposed AGNOS combines Auto-Encoders with structural regularization, and delegates the combinatorial optimization problem at the core of feature selection to a regularized data compression scheme.

1.3.2 Assessing unsupervised FS

The efficiency and relevance of unsupervised FS are usually estimated within a supervised learning setting. Accordingly, the ranking of different selection algorithms depends on the supervised goal.

In order to address this limitation, we introduce a novel performance indicator, called FIT, that corresponds to an unsupervised learning setting. Unsupervised FS algorithms are ranked w.r.t. the informativity of the respective selection subsets to retrieve the whole initial feature set.

1.3.3 Thesis outline

Chapter 2 introduces the general background of the DR problem setting and presents the concept of intrinsic data dimension.

Chapter 3 provides an overview of the DR field and focuses on the positioning of unsupervised FS approaches with respect to other FC techniques.

Chapter 4 introduces the proposed unsupervised FS method AGNOS, including its three declinations AGNOS-S, AGNOS-W and AGNOS-G, each corresponding to a particular structural regularization scheme.

Chapter 5 discusses the prominent supervised performance indicators for assessing unsupervised FS, and introduces the proposed FIT.

The experimental validation of the three versions of AGNOS is described and discussed in chapter 6. The empirical results of AGNOS are compared to baselines both w.r.t. the typical supervised indicators and the proposed unsupervised criterion. The sensitivity of the performance indicators w.r.t. the hyperparameters of the method is also assessed.

Chapter 7 concludes the thesis with a summary of our contributions and a discussion of further perspectives.

Chapitre 2

Dimensionality Reduction : formal background

This chapter first introduces the general setting of Dimensionality Reduction (DR), the main associated challenges and a tentative taxonomy of DR methods in section 2.1. An important hyperparameter of DR techniques is the dimensionality of the resulting low-dimensional data representation, which is tied to the *intrinsic dimension (ID)* of the dataset. Consequently, section 2.2 provides an overview of the ID estimation field. Lastly, section 2.3 concludes the chapter with a discussion of the presented concepts and methods.

2.1 Setting

The following definitions and notations will be used throughout the entirety of this work.

Notations Given strictly positive integers n and D, let $X \in \mathbb{R}^{n \times D}$ denote a dataset containing n samples $(\boldsymbol{x_1}, \ldots, \boldsymbol{x_n})$, each $\boldsymbol{x_i}$ corresponding to a point in a D-dimensional space. $\forall (i,j) \in [1, \ldots, n] \times [1, \ldots, D]$, the coordinate of the i-th datapoint on the j-th dimension is denoted as $f_j(x_i)$. $\forall j \in [1, \ldots, D]$, the n-dimensional vector $\boldsymbol{f_j} = (f_j(x_1), \ldots, f_j(x_n))$ is called the j-th ***feature***. $F = \{\boldsymbol{f_1}, \ldots, \boldsymbol{f_D}\}$ is called the ***original feature set***.

Performing dimensionality reduction is in essence *finding a new feature set containing fewer elements than the original s.t. "information" is preserved*. In order to be theoretically well-grounded, all DR algorithms rely on the following underlying hypothesis, thereafter referred to as *the manifold assumption* :

The manifold assumption *The D-dimensional datapoints $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$ lie near a manifold $\mathcal{M}_{d^*}$ of dimension d^* s.t. $d^* << D$.*

Dimensionality reduction techniques can be categorized with respect to the kind of information they aim to preserve. For instance, Principal Component Analysis (PCA) attempts to capture the variance in the data, while Locally Linear Embedding (LLE) (Roweis and Saul, 2000) is concerned with preserving the pairwise distances between points (more in chapter 3).

Formally, given target reduced dimensionality $d \in \mathbb{N}^*$ s.t. $d << D$, let $\mathcal{H}_d$ denote the set of all sets containing d features computed from F, and $C : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}$ a cost function. A dimensionality reduction problem can then be formulated as finding Z_d^* s.t. :

$$Z_d^* = \underset{Z_d \in \mathcal{H}_d}{\arg\min}\, C(F, Z_d) \tag{2.1}$$

The cost function C measures the loss of information occurring due to the change of representation from F to Z_d^*.

Along this setting, three main questions are addressed in order in the following sections :

— **Should the search for Z_d^* span the entirety of $\mathcal{H}_d$, or should it be restricted to a particular subset of $\mathcal{H}_d$?**(section 2.1.1)

— **How should C be defined ?**(section 2.1.2)

— **How should d be determined ?**(section 2.2)

2.1.1 Constraining the search : Feature Construction and Feature Selection

The general case of DR where there are no restrictions put on $\mathcal{H}_d$ is called *Feature Construction (FC)*[1]. FC techniques are well-suited to perform efficient data compression. However, the results are hardly interpretable, given that the mapping between the original features (elements of F) and the new ones (elements of Z_d^*) is potentially non-linear and arbitrarily complex. In the light of the growing need for *fair*, *transparent* and *accountable (FTA)* learning (Doshi-Velez and Kim, 2017) underlined in chapter 1, recent years saw a rise of popularity of the particular case of *Feature Selection (FS)*. In this setting, the search for Z_d^* is restricted to the subsets of F containing d elements :

$$Z_d^* = \underset{\substack{Z_d \subset F \\ |Z_d| = d}}{\arg\min}\, C(F, Z_d) \tag{2.2}$$

In other words, the new features are selected among the original ones. As such, the results of FS are typically easier to interpret than in the general case of FC[2]. Furthermore, equation (2.2) defines a combinatorial problem ; the goal is to find the best candidate from the C_D^d subsets of F containing d elements (each original feature being either rejected or retained). Exhaustive exploration being too expensive computationally-wise, FS techniques rely on C to quickly orient the search towards a selection subset $\widetilde{Z_d^*}$ s.t. :

$$C(F, \widetilde{Z_d^*}) - C(F, Z_d^*) \leq \epsilon, \text{ with } \epsilon \in \mathbb{R}^{+*}$$

The main challenge of FS is thus to preserve the desired "information" originally contained in F while simultaneously tailoring the cost function C so that the approximation $\widetilde{Z_d^*}$ is as accurate as possible, under the constraint of an affordable time complexity budget.

1. This problem is also frequently referred to as *Feature Extraction (FE)*. However, we find this alternate name to be a potential source of ambiguity, due to the similarity in meaning with Feature Selection. We will therefore prefer using the term FC over FE in this work. Note that FC is also infrequently used in the litterature to describe the process of *adding extra features to* F, thus pertaining to the data augmentation setting, which is outside the scope of this thesis.

2. Consider for instance medical data which purpose is to predict lung cancer in patients. The selection subset $\{\text{age}, \text{smoker}\}$ is likely more understandable to a physician than the artificial features $\{\tanh(0.87\text{age} + 0.13\text{gender}), \sigma(0.95\text{smoker} + 0.05\text{name})\}$.

2.1.2 Defining the cost function : Supervised and unsupervised DR

In the event that the end goal of learning is known at the time of DR, this objective defines an additional *target feature*[3], which can be leveraged to design the cost function C. Consider a FS task on medical data for the purpose of predicting a certain disease among the patients. If one knows which patients are affected by the disease, then C should reflect the usefulness of the original features for discriminating between ill and healthy patients. This idea is the motivation of the early *Fisher score* (Duda et al., 2000) approach. Considering for instance the age feature. Let $\overline{age}$ denote its mean across the whole dataset, and $\overline{age_h}, \sigma_h, \overline{age_i}, \sigma_i$ its mean and variance on the respective subsets of the n_h healthy and n_i ill patients. Then the cost of the $\{age\}$ selection subset singleton w.r.t. the Fisher score is given by :

$$C(\{age\}) = \frac{n_h\sigma_h + n_i\sigma_i}{n_h(\overline{age_h} - \overline{age})^2 + n_i(\overline{age_i} - \overline{age})^2}$$

The older (or younger) the ill patients relatively to the healthy ones, the more important the age variable. Exploiting the learning goal in this way to define C corresponds to the most extensively studied problem in the literature, called *supervised* DR.

An alternate setting, lower in popularity despite its crucial importance (LeCun, 2016) as highlighted in chapter 1, corresponds to the case where **no** learning goal is available. Such a context is referred to as *unsupervised*, in which C is harder to define (more in chapter 3).

Discussion

Following the discussions of sections 2.1.1 and 2.1.2, a tentative taxonomy of DR methods is proposed in Figure 2.1.

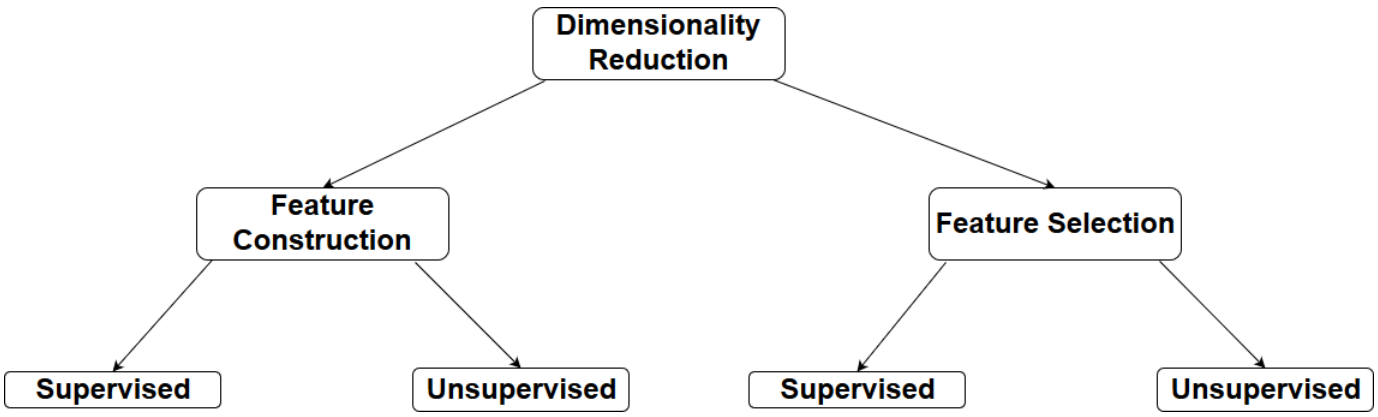

FIGURE 2.1: High-level Taxonomy of Dimensionality Reduction methods

3. There may also exist multiple learning goals simultaneously (Zhang and Zhou, 2007). This variant is called *multi-label* learning (Zhang and Zhou, 2013).

This is in line with the typical point of view in the DR literature, where FS is treated as a concept separate from FC rather than a particular case. Here, this setting is adapted for the sake of a clear and streamlined presentation. However, an important aspect of our algorithmic contribution is to *cross the gap between FC and FS* (chapter 4).

The resulting four categories of DR algorithms, namely **Supervised FC**, **Unsupervised FC**, **Supervised FS** and **Unsupervised FS** will be discussed in chapter 3, along with some of the most prominent approaches for each category.

Following the manifold assumption, FC can be interpreted as the task of projecting the datapoints onto the latent manifold. Therefore, it naturally comes that hyperparameter d should be s.t. $d \approx d^*$. Consequently, the task of tuning d is closely tied to estimating the dimension of the underlying manifold[4]. This can in turn be cast to a problem of *intrinsic dimension* estimation, which we will focus on in the following section.

2.2 Intrinsic dimension

The *intrinsic dimension (ID)* of a dataset is an informal concept commonly defined in Machine Learning[5] as :

Definition 1. *The intrinsic dimension $ID(X)$ of dataset $X \in \mathbb{R}^{n \times D}$ is the minimum number of features required to represent X with negligible loss of information w.r.t. the original D-dimensional representation.*

After this definition, the intrinsic dimension should be an integer and bounded by D. It will be seen that continuous ID is also of interest. After the manifold assumption, ID should by construction match the dimension d^* of the underlying manifold, s.t. $\forall X \in \mathbb{R}^{n \times D}, ID(X) = d^*$. Turning definition 1 into a formal definition proves to be challenging, as underlined by Pestov (2007) :

"A search for the "right" concept of intrinsic dimension of a dataset is not yet over, and most probably one will have to settle for a spectrum of various dimensions, each serving a particular purpose, complementing each other".

To the best of our knowledge, there is no unifying framework for the concept of ID as of yet, even though promising new approaches have been proposed in the last decade (Facco et al., 2017). Additionally, authors often use their own taxonomy of ID estimators ; Facco et al. (2017) differentiate between *fractal* and *neighborhood*-based methods, and we will partially rely on their categorization. Note that e.g. Campadelli et al. (2015) refer to other distinctions such as *projection* and *topological* techniques, interpreted as global vs local estimates. In order to provide an overview of the ID estimation field, we will in the following rely on the tentative taxonomy from figure 2.2.

One may argue that the boundary between fractal and neighborhood-based methods is slim, as it will be shown in sections 2.2.2 and 2.2.3 that both share the core idea of "zooming in" on the samples and providing

4. The link between d and d^* is less immediate in the case of FS : there is no guarantee that the latent manifold can be retrieved from exactly d^* original features. Nevertheless, the manifold assumption provides a lower bound for the selection subset size, that is $d^* \leq d$. Estimating d^* is thus still of interest for the purpose of FS.

5. The concept of intrinsic dimensionality is also relied upon in signal processing (Trunk, 1976), where it refers to the minimum number of variables needed to generate a near-perfect approximation of the original signal.

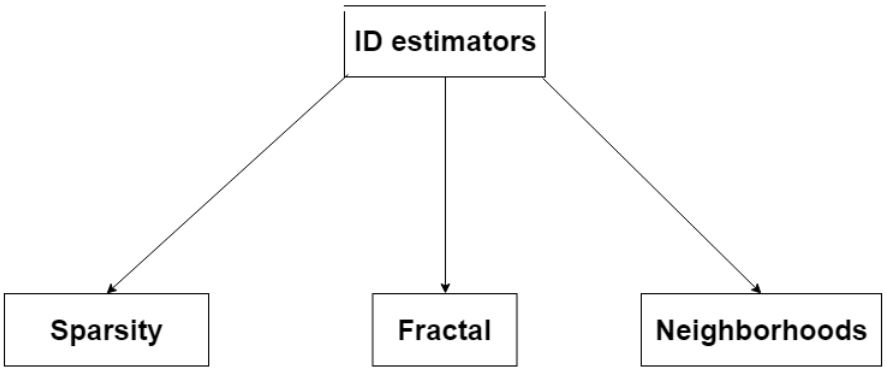

FIGURE 2.2: Tentative taxonomy of intrinsic dimension estimation techniques

local ID estimates. This argument is the main reason for the varying taxonomies of ID estimators found in the literature.

Sparsity-based methods are concerned with estimating the impact on the pairwise distances between dataset samples of the *curse of dimensionality*, which we will now discuss.

2.2.1 The curse of dimensionality and sparsity-based ID estimation

The expression ”curse of dimensionality” was coined by Richard E. Bellman to describe various phenomena arising in high-dimensional spaces that hinder typical data analysis. Among these adverse effects, the *sparsity* issue is the one most prominently considered in ML, and is informally described as :

Definition 2. *As the dimensionality of a dataset grows, samples become more and more spread apart. Eventually, all datapoints are* equally far away from each other.

This property is illustrated on an toy experiment in Figure 2.3. The sparsity property from definition 2 can be rephrased from the perspective of the distribution of pairwise distances between samples :

Definition 3. *As the dimensionality D of the dataset X grows larger, the mean $\mu_D(X)$ of the Gaussian-shaped histogram of pairwise distances increases, while its standard deviation $\sigma_D(X)$ decreases.*

The mathematical characterization of the root cause of this sparsity phenomenon is beyond the scope of this thesis. For the sake of completeness, we refer the interested reader to Pestov (1999) for a more formal explanation of the curse of dimensionality through the lens of the so-called *concentration of measure*.

Given the low variance in pairwise distance, every data sample has many equally remote ”close” neighbors. Therefore, the sparsity property is typically harmful to ML algorithms relying on clustering, or more generally investigating local structure. Considering that this issue is mitigated or even absent in low-dimensional spaces, this provides an additional motivation to perform DR, besides those discussed in chapter 1 : escaping the curse of dimensionality.

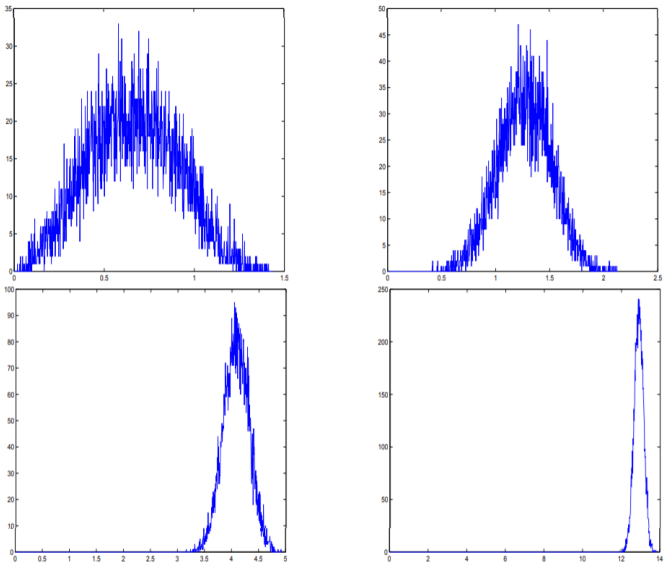

FIGURE 2.3: Distribution of Euclidean distances between 10^4 randomly chosen pairs of points sampled from the unit hypercube $\mathbb{I}^D$, for different values of dimension D. Top-left panel corresponds to $D = 3$, top-right to $D = 10$, bottom-left to $D = 100$ and bottom-right to $D = 1000$. Image taken from Pestov (2007).

The goal of a sparsity-based ID approximation is thus to quantify *how much* the dataset is plagued by the sparsity effect. Chávez et al. (2001) provided a simple estimator using the notations from definition 3 :

$$\widehat{ID}_{sparse}(X) = \frac{\mu_D(X)^2}{2\sigma_D(X)^2} \tag{2.3}$$

The stronger the sparsity effect in X, the larger $\widehat{ID}_{sparse}(X)$. Given that this ID estimation is both quite intuitive and computationally inexpensive, the $\widehat{ID}_{sparse}$ has become a popular tool in the ML community [6].

By contrast with the informal definition $ID(X) \in \mathbb{N}^*$, the above formula does not necessarily return an integer (and can return 0 in pathological cases), that is $\widehat{ID}_{sparse}(X) \in \mathbb{R}^+$. Although a non-integer dimension may seem counter-intuitive, this is in line with *fractal dimensions*, which we will now introduce and discuss.

2.2.2 Fractal-based ID estimation

Though seemingly unrelated, DR and fractal geometry (Mandelbrot, 1983) bear some similarities, that we will now exhibit. As per the manifold assumption, DR relies on the hypothesis that there is a disconnect between the *representation* dimension D of dataset X and its *"true"* dimension $ID(X)$.

6. It directly follows from equation (2.3) that the estimated ID of a dataset composed of a single sample is infinite. Although it has limited practical implications, this mathematical oddity has interestingly enough been shown (Pestov, 2007) to be a required property of any theoretically well-grounded ID estimator.

On the other hand, fractals can be *represented* in 2D or 3D, such as the notorious real-life example of a snowflake (fig. 2.4), or the artificial Julia set (fig 2.5). Given that these two entities can be drawn on a sheet of paper, their representation dimension, thereafter called *topological* dimension, is $D = 2$.

FIGURE 2.4: A snowflake presents self-similarity properties characteristic of fractal patterns. Image taken from Wikipedia.

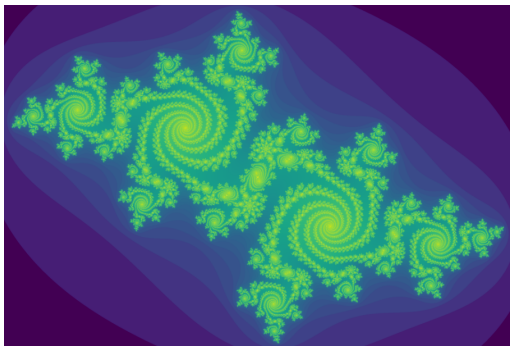

FIGURE 2.5: The Julia set, an artificially generated fractal pattern drawn in 2D. Image taken from Wikipedia.

This does not however provide us with any insight regarding whether these two objects are equally "complex" fractal patterns. Therefore, an additional concept to characterize complexity in a fractal, called *fractal dimension* (Mandelbrot, 1983) is needed, as informally defined in definition 4 :

Definition 4. *The* fractal dimension of any geometrical object is the rate at which details in the pattern change with the scale of measure.

In other words, assume a magnifying glass is at one's disposal. The fractal dimension of the snowflake corresponds to the speed at which it becomes more detailed as one zooms in on a small region, w.r.t. the optical power of the lens.

A well-studied real-world illustration of this concept is provided by the problem of measuring the length of the coastline of Great Britain (Mandelbrot, 1967), as shown in figure 2.6.

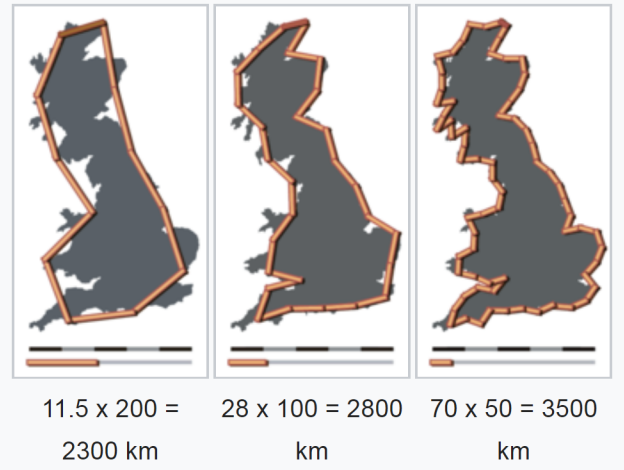

FIGURE 2.6: The measured length of the coastline of Great Britain increases as the length of the measuring stick decreases. Image taken from Wikipedia.

As the size of the segments used to approximate the coastline pattern decreases, the total measured length increases. The ratio between the two quantities is a tentative definition of the intrinsic dimension of Great Britain.

In light of definition 4, the link between fractal geometry and DR is made clearer ; the intrinsic dimension of a dataset can be thought of as the rate at which samples become simpler to depict as one closes in on the datapoints. The concept of fractal dimension can thus be transposed to the context of DR (Camastra and Vinciarelli, 2002; Kégl, 2003) in ML to approximate $ID(X)$. Similarly as for the ID, numerous formal definitions of the fractal dimension were proposed over the years.

Box-counting dimension The most well-known fractal dimension estimator is the *box-counting* dimension (also sometimes referred to as *Minkowski* dimension), which is illustrated in figure 2.7.

The process consists in covering the dataset with square-like boxes of side length ϵ, then observing how the number $N(\epsilon)$ of boxes needed to achieve full coverage increases as ϵ decreases :

$$\widehat{ID}_{box}(X) = \lim_{\epsilon \to 0} \frac{log(N(\epsilon))}{log(\frac{1}{\epsilon})} \tag{2.4}$$

Informally, this means that if X contains n samples, $\widehat{ID}_{box}(X)$ is the exponent of the power law such that $N(\frac{1}{n}) \propto n^{\widehat{ID}_{box}(X)}$.

Hausdorff dimension An alternative to the box-counting dimension is provided by the *Hausdorff* dimension, which relies on balls rather than square boxes, as illustrated in figure 2.8.

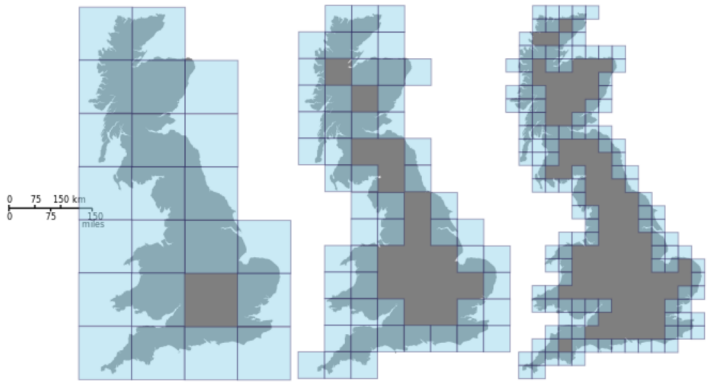

FIGURE 2.7: Illustration of the box-counting dimension concept applied to the Great Britain coastline example. Image taken from Wikipedia.

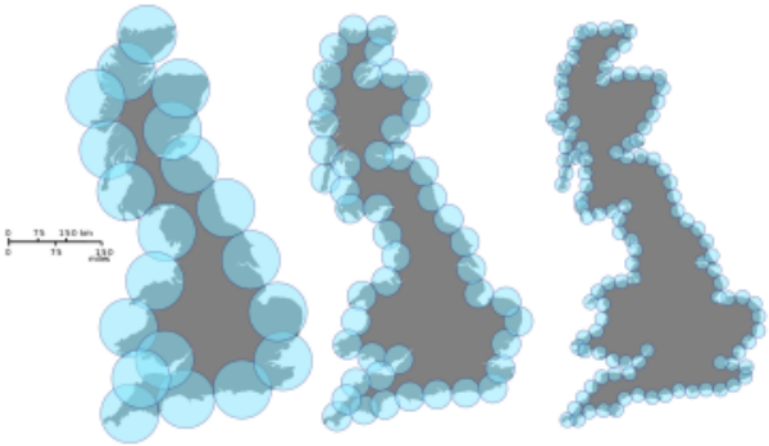

FIGURE 2.8: Illustration of the Hausdorff dimension concept applied to the Great Britain coastline example. Image taken from Wikipedia.

Defining $S(X)$ the (infinite) set of possible covers of X by balls centered on the datapoints, each ball being associated to its radius $r_i > 0$, the *Hausdorff content* of X is defined as :

$$C^d(X) = \inf_{S(X)} \sum_{r_i} r_i^d \tag{2.5}$$

The Hausdorff dimension of X is then given by :

$$\widehat{ID}_{\text{Hausdorff}}(X) = \inf_{d \geq 0} \Big(C^d(X) = 0 \Big) \tag{2.6}$$

The underlying idea is thus to cover the dataset with progressively smaller spheres, and observe how the total volume of the coverage decreases as the spheres shrink. Therefore, despite having a less intuitive

definition, the Hausdorff dimension is conceptually close to the box-counting dimension. Interestingly enough, these two quantities are linked by the following inequality :

$$\forall X \in \mathbb{R}^{n \times D}, \widehat{ID}_{\text{Hausdorff}}(X) \leq \widehat{ID}_{box}(X) \tag{2.7}$$

Even though $\widehat{ID}_{\text{Hausdorff}}(X) = \widehat{ID}_{box}(X)$ in most cases, the equality does not necessarily hold (e.g. considering $\mathbb{Q}$ the set of rational numbers, $\widehat{ID}_{\text{Hausdorff}}(\mathbb{Q}) = 0$ whereas $\widehat{ID}_{box}(\mathbb{Q}) = 1$).

Discussion

Both fractal dimensions presented above are hardly affordable in practice in terms of computational complexity. In order to address this issue, the *correlation* dimension (Camastra and Vinciarelli, 2002) and *packing* dimension (Kégl, 2003) were designed by the ML community as computationally efficient variants of respectively the box-counting and Hausdorff dimensions.

An additional benefit of the fractal geometry framework is to shed light on the counter-intuitive notion of a non-integer intrinsic dimension, as was first showcased in section 2.2.1 ; the rate at which details appear in a fractal is not necessarily a multiple of the zoom multiplier. Going back to the Koch snowflake example, the total length of the pattern contour increases by a factor 4 every time the scale is enhanced by a factor 3. Therefore, the Koch snowflake is considered of fractal dimension $log(4)/log(3) \approx 1.27$.

Global and local ID estimates There is an important contrast in terms of methodology between sparsity-based and fractal-based methods. The technique from Chávez et al. (2001) is only concerned with the histogram of pairwise distances of X, thus considers the ID on the *global* scale and assumes it is invariant across the samples. By contrast, fractal-based methods investigate the complexity of each individual sample. The final $ID(X)$ is therefore actually an aggregate of multiple estimations made on the *local* scale. This is tied to the underlying assumption that the ID dimension of a dataset is fluctuating across the different regions of the representation space. For example, it is likely that the ID of a cluster near the origin (meaning all feature values are simultaneously close to 0) differs from outliers. This idea that one must examine the ID on a local basis forms the basis of *neighborhood-based* ID estimators, which we will now introduce.

2.2.3 Neighborhood-based ID estimation

Early neighborhood-based ID estimators (Pettis et al., 1979; Verveer and Duin, 1995) directly relied on the Euclidean distance in the original representation space of the dataset to detect neighborhoods. However, following section 2.2.1, the sparsity effect caused by the curse of dimensionality prevents this strategy from being sound in high-dimensional spaces. In order to provide reliable results, one must therefore devise alternate ways of detecting neighborhoods.

To that end, (Costa and Hero, 2004) rely on a popular unsupervised Feature Construction method called *Isomap* (more in chapter 3). The idea is to aggressively prune the pairwise sample similarity graph provided by Isomap, so that all datapoints remain connected using as few edges as possible. An illustration of this process is provided in figure 2.9, on the well-known *Swiss Roll* artificial dataset. Despite being originally

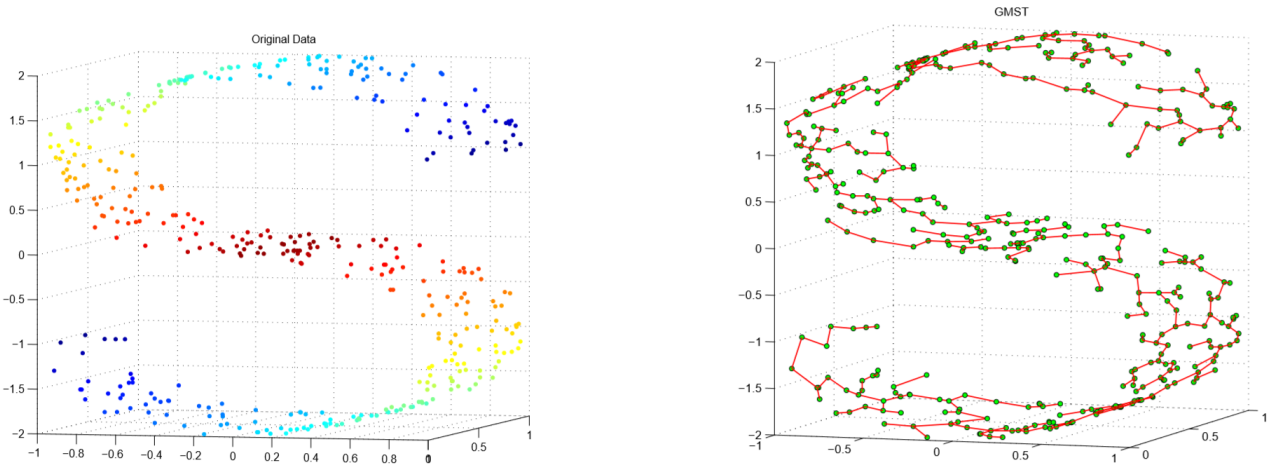

FIGURE 2.9: An example of Geodesic Minimal Spanning Tree on a Swiss Roll. Image taken from (Costa and Hero, 2004).

generated and represented in a 3D space, this manifold can be "unfolded" and mapped to a 2D space. Most ID estimators will therefore return $\widehat{ID}(\text{SwissRoll}) \approx 2$.

The pruned similarity graph is called a *Geodesic Minimal Spanning Tree (GMST)*. It has been shown (Costa and Hero, 2004) that the depth of this tree can be used to approximate the intrinsic dimension of the manifold. The main shortcoming of the GMST approach is that the quality of its ID estimation depends on the accuracy of the similarity graph provided by Isomap. Unfortunately, Isomap has been experimentally demonstrated (Balasubramanian and Schwartz, 2002) to be topologically unstable, meaning a small error in neighborhood assessment (relying still on the Euclidean distance) can lead to a large error in the final graph.

Consequently, the GMST approach has decreased in popularity to the benefit of *Maximum Likelihood Estimation (MLE)* (Levina and Bickel, 2005). This technique relies on the assumption that if one "zooms in" on a particular datapoint x, then the density $f(x)$ of samples in this region is roughly constant in a ball $S_x(r)$ centered on x with a small radius r, s.t. $f(x) \approx C$ everywhere in $S_x(r)$. The idea is then to consider the samples as an homogenous Poisson process in $S_x(r)$. The number $N(x,r)$ of datapoints inside $S_x(r)$ is then s.t. $N(x,r) \propto Cr^{d(x)}$. Here, $d(x)$ denotes the dimension of sphere $S_x(r)$ and corresponds to the local ID *at point* x. Given a fixed sphere radius r_0, the final ID approximation $\widehat{ID}_{MLE}$ is then simply obtained by averaging the local estimators :

$$\forall X \in \mathbb{R}^{n \times D}, \widehat{ID}_{MLE}(X) = \frac{1}{n} \sum_{x \in X} d(x) \tag{2.8}$$

As said, the MLE ID estimator relies on the strong assumption of locally constant sample density. In practice, the more inhomogenous the underlying Poisson process, the less accurate $\widehat{ID}_{MLE}$.

Building upon the MLE technique, Facco et al. (2017) proposed a neighborhood-based ID estimator that only relies on the *two closest* neighbors of each point, denoted $\widehat{ID}_{2NN}$. This essentially means that the

homogenous Poisson process assumption needs only to hold in smaller spheres than before. $\widehat{ID}_{2NN}$ is therefore better suited than $\widehat{ID}_{MLE}$ to deal with non-smooth manifolds.

Similarly as before, $\widehat{ID}_{2NN}(X)$ is obtained by averaging local estimates $d(x)$. However, rather than from counting the number of samples in $S(x, r_0)$, $d(x)$ is derived from the volume of the hyperspherical shell enclosed between the closest and second closest neighbors of x, as illustrated in figure 2.10.

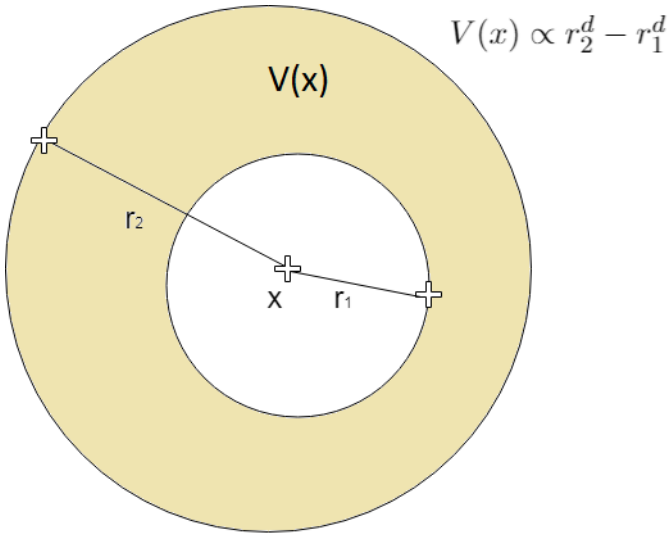

FIGURE 2.10: Two-dimensional example of local ID estimation via the 2NN technique.

Before averaging the local results, the top 10% ranked datapoints w.r.t. the ratio $\frac{r_2}{r_1}$ are discarded. This essentially means that samples with only one close neighbor are ignored, given that these are likely to be outliers and would skew the final estimator $\widehat{ID}_{2NN}$.

An additional benefit of the $2-NN$ method over traditional MLE estimation is the reduced computational cost ; Muja and Lowe (2014) have shown that dedicated algorithms were able to find the first two neighbors of n points in approximately $O\left(nlog(n)\right)$ time.

2.3 Discussion

As discussed above, the boundary between fractal and neighborhood is slim, e.g. the pioneer MLE estimator from Levina and Bickel (2005) is referred to as a fractal-based technique in Facco et al. (2017).

Criteria for ID estimation As shown, ID estimation is key to DR. In this context, the criteria for choosing an adequate ID estimator include :

— Affordable computational cost (both time and space-wise)

— Resilience w.r.t. the sparsity effect of the curse of dimensionality

— Accuracy of the results : $\widehat{ID}$ should be close to the dimension d^* of the latent manifold

The sparsity-based estimator from Chávez et al. (2001), the correlation dimension (**?**) and the packing dimension (Kégl, 2003) all meet the first criterion. However, it has since been demonstrated (Pestov, 2007) that the resulting $\widehat{ID}$ is not necessarily close to d^* ; all these techniques thus fail to meet the accuracy criterion. Additionally, correlation and packing dimensions are also sensitive to the sparsity effect (Pestov, 2007).

By contrast, the $2-NN$ method (Facco et al., 2017) is both inexpensive computationally-wise and resilient to the sparsity effect. Furthermore and to the best of our knowledge, $\widehat{ID}_{2NN}$ is empirically a close approximation of d^*, in cases where ground truth knowledge about the dimension of the underlying manifold is known. Therefore, we will in the context of this thesis rely on the $2-NN$ approach for the purpose of ID estimation. The discussion of this chapter should however not make us forget that many authors (Guyon et al., 2002; Li et al., 2016; Ye and Sun, 2018) instead proceed manually or iteratively to tune d.

Alternate visual-based tuning of d d can be tuned relying e.g. on Principal Component Analysis (PCA) using a grid search process. The idea would be to perform a PCA of the original data for different number of principal components (corresponding to the constructed features), then visually examine the smoothness of the respective covariance matrices spectra.

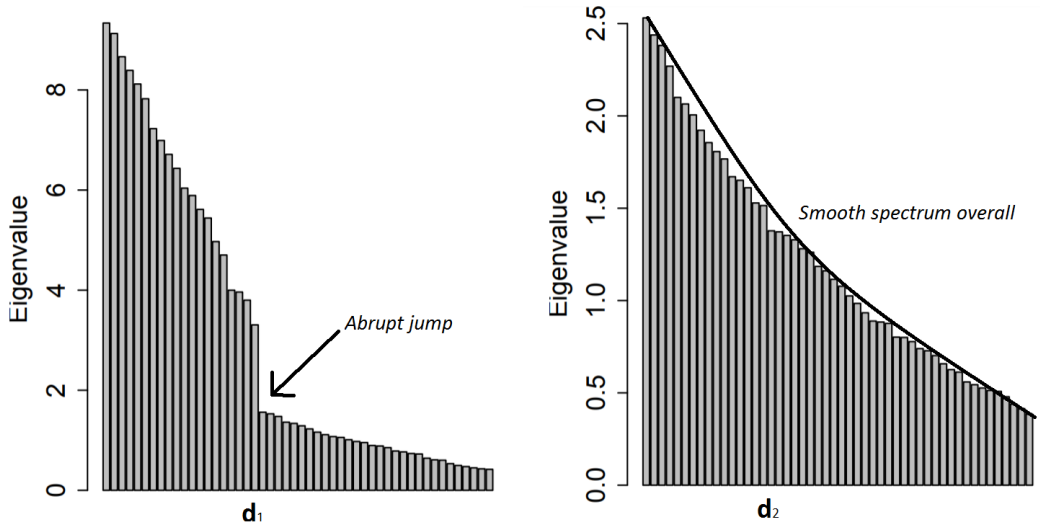

FIGURE 2.11: Example of covariance matrices spectra resulting of PCA for different dimensions d_1 and d_2

In the example provided in figure 2.11, the large gap between the leading eigenvalues and the tail of the spectrum for $d = d_1$ indicates that some of the constructed features are much less important than the others for capturing the variance in the data. Therefore, d should be lowered until the spectrum becomes smooth, as is the case for $d = d_2$.

This ”trial-and-error” technique come with the significant downside of being computationally expensive, given that the FC algorithm relied upon must be ran many times to obtain accurate results. This approach is therefore ill-advised in view of Big Data.

Online selection subset size calibration Another possibility exclusive to FS is to forego ID estimation and determine the number of features to select online, that is during the execution of the FS algorithm. This is a popular solution in *supervised* FS. In the *Drop-Out-One* approach (Ye and Sun, 2018) (more in chapter 3) for instance, features are greedily eliminated until the resulting change in predictive accuracy of the target features becomes lower than a threshold.

To the best of our knowledge, this type of technique has not yet been proposed for the *unsupervised* context, where a suitable stopping criterion is harder to define. This defines a direction for future research (chapter 7).

Chapitre 3

Dimensionality reduction : overview

Based on the formal background on Dimensionality Reduction (*chap.* 2), this chapter presents the state-of-the-art in *Feature Extraction (FC)* (section 3.1) and *Feature Selection (FS)* (section 3.2), in both supervised and unsupervised contexts (thereafter abbreviated as *sup.* and *unsup.*). A brief review of some of the most impactful and popular approaches is provided for each of the four problem settings. The chapter concludes with a discussion on the strengths and limitations of the aforementioned methods, coupled with the main lessons learned.

An important methological difference between FC and FS lies in the way *sup.* and *unsup.* learning are perceived. In the general FC context, *unsup.* historically came first, with early methods such as PCA (Pearson, 1901), and remains the main focus of the literature. *Sup.* learning is then layered on top of *unsup.*, and label information is taken advantage of as much as possible.

On the other hand, FS is most often studied in the *sup.* context. The absence of labels is seen as a handicap. *Unsup.* FS therefore aims to build pseudo-label information in order to fall back on *sup.* learning.

Accordingly, section 3.1 will introduce *unsup.* FC methods first, followed by *sup.* approaches. By contrast, section 3.2 begins with *sup.* FS and presents *unsup.* techniques afterwards.

3.1 Feature Construction

Following the general problem setting exposed in chapter 2 and using the same notations, DR aims to find a mapping $\phi^* : F \to \mathcal{H}_d$ s.t. $\phi^*(F) = Z_d^*$ minimizes cost function C. FC is not limited to "sparse filters" mappings selecting a subset of the initial features (as is FS). Instead, ϕ^* can correspond to either a *linear* combination of the initial features or an arbitrarily complex *non-linear* function thereof.

3.1.1 Unsupervised FC

3.1.1.1 Linear mappings

Principal Component Analysis Perhaps the most well-known DR technique overall is the *Principal Component Analysis (PCA)* algorithm (fig 3.1), an early approach dating back to the 19th century and formalized by Pearson (1901). Assuming the original features have been preemptively centered and given the data covariance matrix $X^T X$, the constructed features (referred to as *principal components*) are given by the columns

of matrix Z s.t. $Z = XW$, where W contains the eigenvectors of X^TX. In order to perform DR, only the d leading eigenvectors are considered, s.t. $Z_d = XW_d$ is now a $n \times d$ matrix [1] The underlying idea is that provided d is well-tuned (see section 2.2), the leading eigenvectors are sufficient to capture most of the variance in the data. As said in chapter 2, DR methods can be categorized by the kind of "information" they respectively aim to preserve. Accordingly, the purpose of PCA is to accurately depict *correlations* between original features.

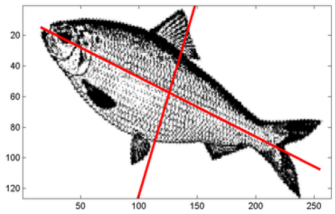

FIGURE 3.1: A 2D example of PCA applied to the picture of a fish. From Wikipedia.

Singular Value Decomposition The *Singular Value Decomposition (SVD)* of $X \in \mathbb{R}^{n \times D}$ is given by $X = U\Sigma V$, where U and V are orthogonal matrices of respective dimensions $n \times n$ and $D \times D$, and Σ a $n \times D$ diagonal matrix. The diagonal entries of Σ (the *singular values* of X) are the square roots of the non-zero eigenvalues of X^TX.

In order to perform DR (as well as greatly reduce the computational cost), only the d largest singular values and corresponding singular vectors are computed in the *truncated* SVD (Golu and Reinsch, 1971), s.t. $\hat{X} = U_d\Sigma_d V_d$. The constructed features then correspond to the rows of V_d.

Interestingly enough, SVD is equivalent to PCA if the original features have been preemptively centered to zero mean. However, centering is ill-advised in some cases, e.g. image processing tasks where features correspond to positive pixel intensity values ; mean centering transforms null values into high amplitude negative values, artificially increasing the (usually low) importance of the corresponding features. SVD is therefore usually preferred over PCA in this setting.

Multi-Dimensional Scaling *Multi-Dimensional Scaling (MDS)* (Torgerson, 1958; Borg and Groenen, 2003) follows a different line of thought to PCA and SVD ; the goal is to preserve the likeness of samples (rather than the correlations between features). Accordingly, the input of the method is a *similarity matrix* containing the pairwise Euclidean distances between datapoints, which spectral decomposition produces the constructed features. The notion of similarity matrix is also central to *unsup.* FS, and the associated theory of *spectral clustering* (Von Luxburg, 2007) will be presented in more detail in section 3.2.3.

1. Interestingly enough, the Eckart-Young theorem (Eckart and Young, 1936) guarantees that Z_d is the best approximation (in the sense of the Frobenius norm) of X by a matrix of rank d or less.

Discussion

Over the years, the three aforementioned techniques have been further refined into many computationally efficient variants such as the *k-SVD* (Aharon et al., 2006) or the *generalized MDS* (Bronsteina et al., 2006). However, a core limitation remains : the mapping ϕ^* resulting of any of PCA, SVD or MDS is linear. As such, these methods are ill-advised for performing DR on non-linear manifolds. An example of unwanted behavior is provided in figure 3.2 on an artificial Swiss Roll dataset (see section 2.2.3), which a standard PCA is unable to properly unfold.

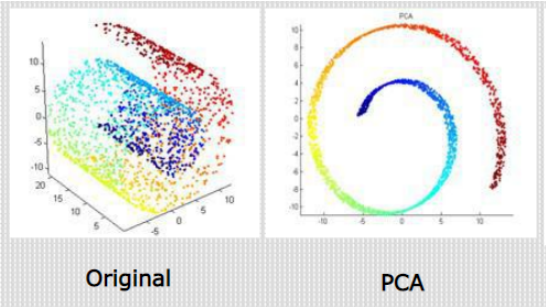

FIGURE 3.2: Far-away points are wrongly assessed as close together in the PCA projection by the Euclidean distance. From Roweis and Saul (2000).

3.1.1.2 Non-linear mappings

Isomap *Isomap* (Tenenbaum et al., 2000) is an extension of MDS designed to handle non-linear manifolds, taking note of the unreliability of the Euclidean distance (panel A of figure 3.3). The similarity matrix is first translated into graph form : each vertex represents a data point, and two vertices are connected by an edge iff one is part of the k nearest neighbors (w.r.t. the Euclidean distance) of the other. This allows computing a geodesic distance between vertices, corresponding to the length of the shortest path connecting them in the graph (panel C). Panel B shows that this new distance accurately detects neighborhoods in the Swiss roll. MDS is then performed using the geodesic similarity matrix as input. The reliance on local neighborhoods of the geodesic distance is what allows the resulting mapping ϕ^* to be non-linear.

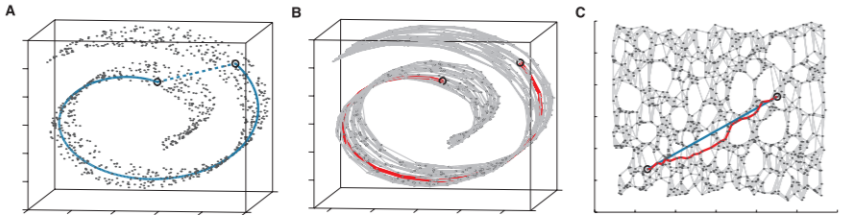

FIGURE 3.3: The geodesic distance accurately depicts pairwise similarities, and is used by Isomap to unfold the Swiss roll. From Tenenbaum et al. (2000).

As mentioned earlier (section 2.2.3), Isomap has quickly been shown to be topologically unstable (Balasubramanian and Schwartz, 2002), meaning that adding a small perturbating noise to the data is sufficient to greatly corrupt the associated similarity graph, and consequently the constructed features (figure 3.4).

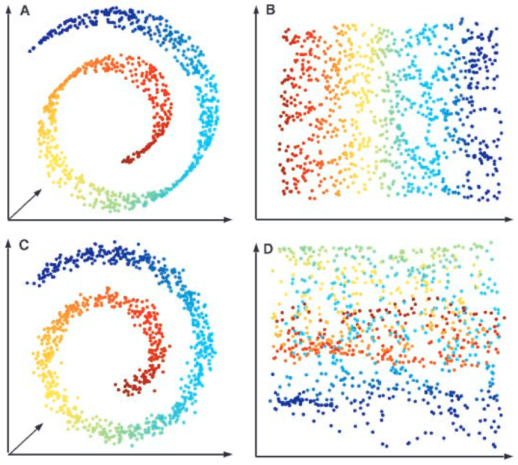

FIGURE 3.4: Adding Gaussian noise of small amplitude leads to a badly unfolded Swiss Roll. From Balasubramanian and Schwartz (2002).

Locally Linear Embedding *Locally Linear Embedding (LLE)* (Roweis and Saul, 2000; Saul and Roweis, 2003) first defines the local structure of the n data points $x_i \in \mathbb{R}^D$, through approximating each point as the barycenter of its k nearest neighbors. The goal is then to find points $y_1, \dots y_n$ in $\mathbb{R}^d$, such that the y_i satisfy the same local relationships as the x_is. Formally, let $N(i)$ denote the set of indices of the k nearest neighbors of x_i. The weights $W_{i,j}$ then minimize the Euclidean distance $\|x_i - \sum_{j \in N(i)} W_{i,j} x_j\|$ with the constraints $\sum_{j \in N(i)} W_{i,j} = 1$, $W_{i,j} \geq 0$ and $W_{i,j} = 0$ for $j \notin N(i)$. Note that W is invariant under rotation, translation or homothety on the dataset X : it captures the local structure of the samples. These local relationships are then leveraged to learn Y s.t. :

$$Y = \underset{M \in \mathbb{R}^{n \times d}}{\arg\min} \|M - WM\|_F^2$$

An illustration of the LLE DR process is provided in figure 3.5.

3.1.1.3 Non-linear mappings learned with AutoEncoders

AutoEncoders are artificial neural networks designed to perform FC. An AutoEncoder is composed of two interacting parts : an *encoder* ϕ that learns the constructed features, followed by a *decoder* ψ aims to reconstruct the original variables from the constructed ones (figure 3.6). The goal is to minimize the discrepancy between the original data and the output of the decoder, i.e. find ϕ^*, ψ^* s.t. :

$$\phi^*, \psi^* = \underset{\psi, \phi}{\arg\min}\, Loss(X, (\psi \circ \phi)X) \tag{3.1}$$

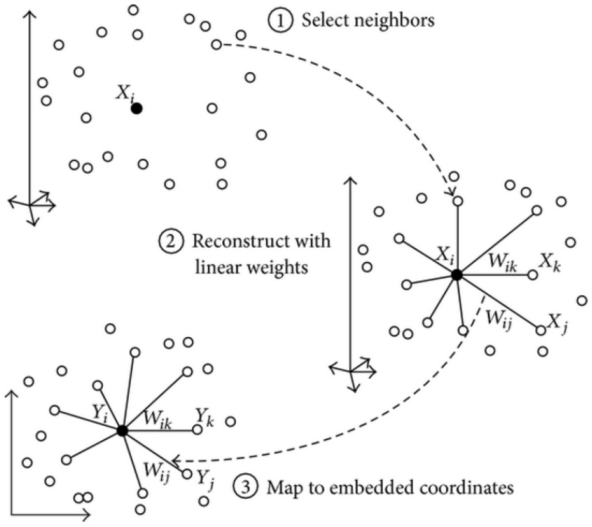

FIGURE 3.5: Illustration of the three sequential steps involved in LLE.

The most widely used loss function is the squared L_2 norm (often referred to as the *mean squared error (MSE)* loss), i.e. :

$$L = ||(\psi \circ \phi)(X) - X||_F^2 = ||\widehat{X} - X||_F^2 = \sum_{i=1}^{n} ||\widehat{\boldsymbol{x_i}} - \boldsymbol{x_i}||_2^2 \tag{3.2}$$

Alternatively, the MSE loss can be interpreted as a sum of individual errors over each feature :

$$L(F) = \sum_{i=1}^{D} ||\widehat{\boldsymbol{f_i}} - \boldsymbol{f_i}||_2^2$$

Equations (3.2) and (3.1.1.3) are clearly equivalent. However, given that our algorithmic contribution pertains to the field of unsupervised FS and relies on AutoEncoders (more in chapter 4), we will in the following prefer using the feature-based interpretation $L(F)$.

Whether ϕ and ψ are linear mappings depends on the activation functions used in the neural network. As shown by Baldi and Hornik (1989), an AutoEncoder with linear activation functions and a MSE loss essentially performs PCA. In practice, nonlinearities such as *sigmoid*, *tanh* or *ReLU* (Xu et al., 2015) are thus relied upon instead.

If the output of the encoder is of the same dimensionality as the input, then the optimization problem is trivially solved by learning an identity mapping for both the encoder and the decoder[2]. In order to enforce finding an interesting solution, the common strategy is thus to enforce $\phi : \mathbb{R}^D \to \mathbb{R}^d$ and $\psi : \mathbb{R}^d \to \mathbb{R}^D$, with

2. More precisely, it suffices that $\psi \circ \phi = Id$, which is achieved by learning any invertible $\phi : \mathbb{R}^D \to \mathbb{R}^D$ and $\psi = \phi^{-1}$. This is unlikely to produce a useful representation in the encoder. Furthermore, this can still occur even if non-linear activation functions are used (e.g. ϕ can stay in the linear regime of a sigmoid with small enough neuronal weights).

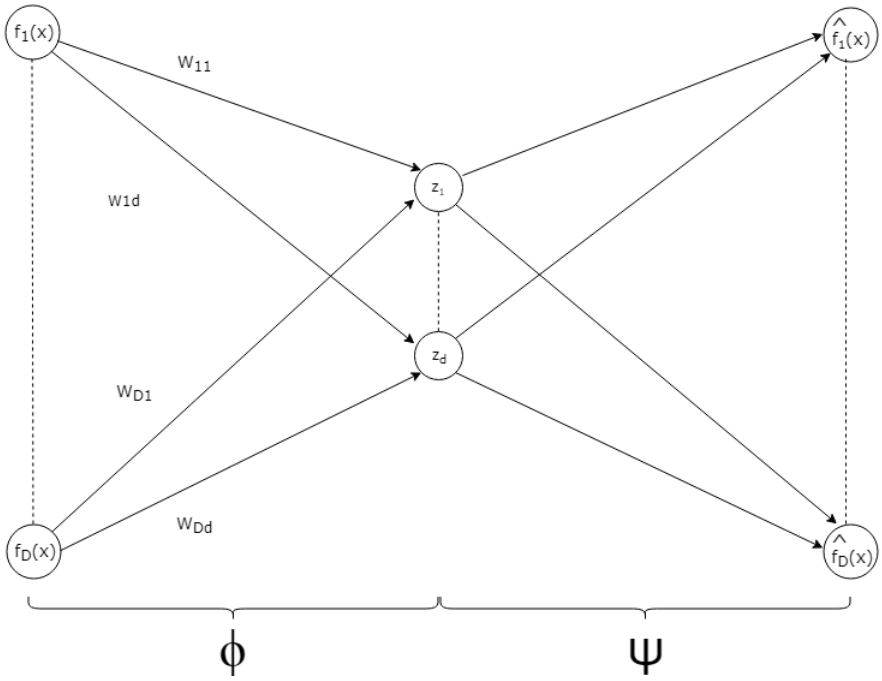

FIGURE 3.6: An AutoEncoder performs unsupervised FC by compressing the input into d latent variables.

$d << D$, thus creating an *under-complete* representation in the encoder. This under-complete representation can be viewed as data compression in the sense that ψ allows to recover the initial information.

An alternative to under-complete representations is to rely on an *over-complete* representation ($d > D$) under the constraint that ϕ is sparse. This idea of implicit compressibility rather than explicit is at the core of *sparse coding* (Olshausen and Field, 1997). Sparse encoding has been shown (Poultney et al., 2007; Boureau and LeCun, 2008) to produce expressive constructed features s.t. each of these *filters* memorizes a different piece of information related to the input. The reconstructed data is provided by the additive (linear) combination of these filters (fig. 3.7).

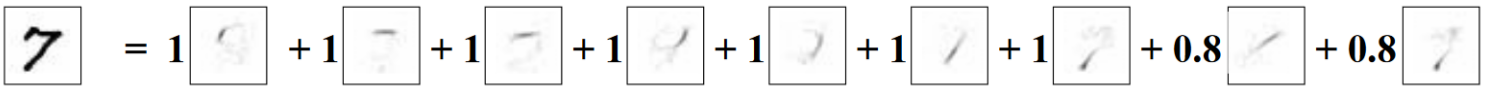

FIGURE 3.7: Example of encoder-learned filters on a handwritten digit taken from the popular MNIST database. From Boureau and LeCun (2008).

Another strategy to prevent the AutoEncoder from learning uninteresting features is to use a loss function different than the one in equation (3.2). Such a strategy is implemented by Denoising AutoEncoders (Vincent et al., 2008), which first produce a noisy version $\tilde{X}$ of X to use as input data, then task the network with

reconstructing the original "clean" data from the corrupted version (fig. 3.8) :

$$\phi^*, \psi^* = \underset{\psi,\phi}{\arg\min}\, Loss(X, (\psi \circ \phi)\tilde{X}) \tag{3.3}$$

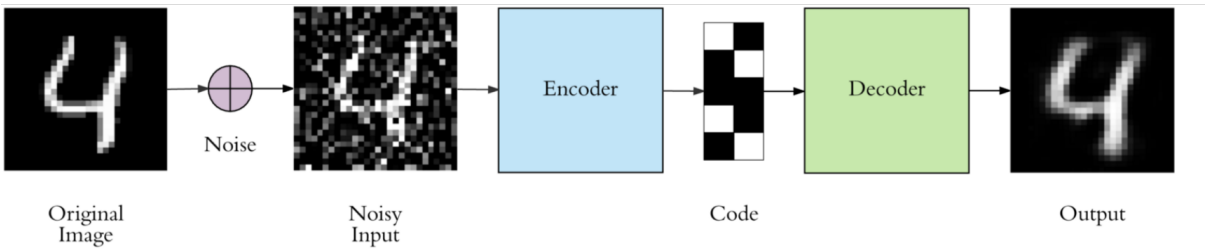

FIGURE 3.8: Illustration of the denoising AutoEncoder process on MNIST.

Following the manifold assumption (*sec.* 2.1), corrupted samples in $\tilde{X}$ will generally lie farther than the uncorrupted datapoints from the underlying low-dimensional manifold. A denoising AutoEncoder is therefore learning to project these noisy samples back onto the manifold. To guarantee the success of this *denoising* process, the features constructed by the encoder must be resilient to perturbations in the input, and should thus be more expressive than the original ones. The resulting compressed representation accordingly depends on the type and magnitude of corrupting noise applied to transform X into $\tilde{X}$. The most common corruption is the so-called *masking noise*, setting a fraction of randomly chosen feature values to zero. This method can be linked to the DropOut strategy (Hinton et al., 2012; Srivastava et al., 2014), setting a random fraction of neuronal weights to zero to avoid overfitting as well as co-adaptation of neurons.

In order to further increase the robustness of the learned compressed data representation w.r.t. input noise, Denoising AutoEncoders can be stacked together : after one AutoEncoder has been trained, the output of its decoder is corrupted, and is fed as input to the next AutoEncoder in the stack. Such an architecture is called a Stacked Denoising AutoEncoder (Vincent et al., 2010), and leads to a series of intermediate compressed representations, each being resilient to the noise applied to the previous one. The resulting embedding Z_d^* is then provided by the deepest, lowest-dimension encoder.

Along the many more AutoEncoder variants that were unmentioned so far, the most notable approach corresponds to *Variational AutoEncoders (VAE)* (Kingma and Welling, 2013) (appendix .1). Although VAEs are actually quite remote from classical Auto-Encoders, they constitute a potential extension of our algorithmic contribution (*chap.* 7).

Discussion

The unsupervised FC methods introduced in section 3.1.1 can be organized in three categories : variance-preserving mappings (PCA, SVD), similarity-preserving mappings (MDS, Isomap, LLE), and reconstruction-preserving mappings (AutoEncoders). Similarity-preserving methods rely on the Euclidean distance or related constructs such as geodesic distances to assess the likeness of datapoints. As underlined in chapter 2, this might be unreliable for high-dimensional datasets due to the curse of dimensionality.

By contrast, AutoEncoders do not need to compute any kind of pairwise sample similarity, given that these are concerned only with rebuilding the original data representation from the constructed features. This key property is at the core of the proposed unsupervised FS approach (chapter 4).

3.1.2 Supervised FC

Interestingly enough, the most well-known supervised FC approaches all correspond to similarity-preserving mappings [REFS]. Furthermore, these techniques share the same core idea that samples of the same class should be close together w.r.t. the constructed features in Z_d^*, while samples of different classes should be mapped as far away from each other as possible.

3.1.2.1 Linear mapping via the Fisher discriminant

The *Fisher Discriminant* (Fisher, 1936), often referred to as *Linear Discriminant Analysis (LDA)* is an early supervised FC approach attempting to maximize the *inter-class* scatter while simultaneously minimizing the *intra-class* scatter. Formally, consider a binary classification task composed of n_1 samples from the first class and n_2 samples from the second one, and denote by X_1 and X_2 the subsets of samples corresponding to the respective classes. $\forall i \in \{1, 2\}$, let $\boldsymbol{m_i} = \frac{1}{n_i} \sum_{\boldsymbol{x} \in X_i} \boldsymbol{x}$ denote the vector of average feature values on the i-th class. The *between-class* and *within-class* scatter matrices $S_B \in \mathbb{R}^{D \times D}$ and $S_W \in \mathbb{R}^{D \times D}$ are then defined s.t. $S_B = (\boldsymbol{m_1} - \boldsymbol{m_2})(\boldsymbol{m_1} - \boldsymbol{m_2})^T$ and $S_W = \sum_{i=1}^{2} \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{m_i})(\boldsymbol{x} - \boldsymbol{m_i})^T$.

LDA then aims to maximize the Fisher discriminant, that is find $w^* \in \mathbb{R}^{D \times 1}$ s.t. :

$$J(\boldsymbol{w^*}) = \frac{\boldsymbol{w^{*T}} S_B \boldsymbol{w^*}}{\boldsymbol{w^{*T}} S_W \boldsymbol{w^*}} = \underset{\boldsymbol{w} \in \mathbb{R}^{D \times 1}}{\arg\max}\, J(\boldsymbol{w}) \tag{3.4}$$

Intuitively, this corresponds to finding a direction maximizing the projected class means (the numerator) and minimizing the classes variance in that direction (the denominator). This direction corresponds to the vector normal to the discriminant hyperplane separating the classes (fig. 3.9).

The end result of LDA is thus a *single* constructed feature $\boldsymbol{z^*}$ s.t. $\forall i \in [1, \dots, n], \boldsymbol{z^*}(\boldsymbol{x_i}) = \boldsymbol{w}.\boldsymbol{x_i}$. This constructed feature is clearly obtained from the original variables through a linear mapping. Therefore, LDA suffers from the same issue as linear *unsup.* FC techniques, as discussed in section 3.1.1.

In order to lift this limitation, non-linear refinements of the approach were later proposed. The most prominent such refinement is the *Kernel LDA* (Mika et al., 1999), which first projects the input samples to a new space through a non-linear kernel function, then relies on the *kernel trick* (Hofmann et al., 2008) to efficiently compute the Fisher discriminant in that space. Consequently, LDA still enjoys a high popularity for modern applications, both for the purpose of FC (Ghassabeh et al., 2015) and error correction (Gorban et al., 2018).

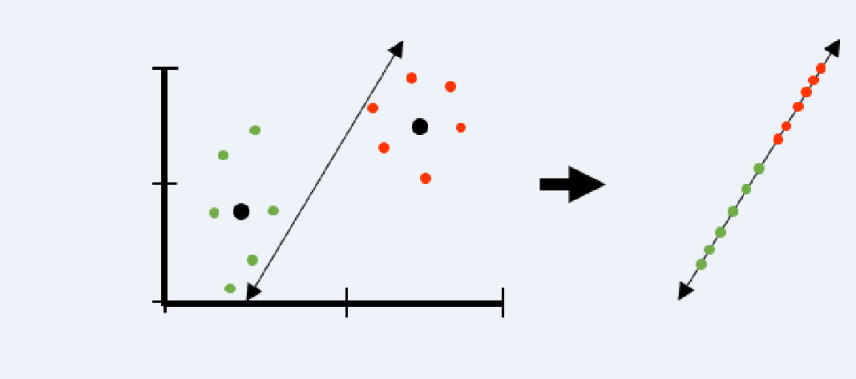

FIGURE 3.9: As a result of LDA, samples are projected on the vector normal to the discriminant hyperplane separating the first class (in green) from the second (in red). From Wikipedia.

3.1.2.2 Non-linear mappings

Non-linear supervised FC techniques can be split in two categories : those that "enrich" the *unsup.* FC approach Isomap (see section 3.1.1) with label information, and those based on neural networks.

Enriched Isomap techniques Using the same notations as for LLE in section 3.1.1, the similarity metric at the core of Isomap is in the *unsup.* context typically defined as :

$$\forall (i,j) \in [1,\dots,n]^2, S(\boldsymbol{x_i},\boldsymbol{x_j}) = \begin{cases} exp^{(-||\boldsymbol{x_i}-\boldsymbol{x_j}||^2/\beta)} \text{ if } j \in N(i) \text{ or } i \in N(j) \\ 0 \text{ otherwise} \end{cases}$$

This formalization clearly does not take advantage of the labels $(y_1,\dots,y_n)$. In order to do so, supervised approaches such as *Locally Discriminant Projection (LDP)* (Zhao et al., 2006a) and *Orthogonal Discriminant Projection (ODP)* (Li et al., 2009) propose tweaking S s.t. :

$$\forall (i,j) \in [1,\dots,n]^2, S(\boldsymbol{x_i},\boldsymbol{x_j}) = \begin{cases} exp^{(-||\boldsymbol{x_i}-\boldsymbol{x_j}||^2/\beta)}\left(1+exp^{(-||\boldsymbol{x_i}-\boldsymbol{x_j}||^2/\beta)}\right) \text{ if } j \in N(i) \text{ or } i \in N(j) \textbf{ and } y_i = y_j \\ exp^{(-||\boldsymbol{x_i}-\boldsymbol{x_j}||^2/\beta)}\left(1-exp^{(-||\boldsymbol{x_i}-\boldsymbol{x_j}||^2/\beta)}\right) \text{ if } j \in N(i) \text{ or } i \in N(j) \textbf{ and } y_i \neq y_j \\ 0 \text{ otherwise} \end{cases} \tag{3.5}$$

Equation (3.5) enforces that the similarity between neighboring samples from the same class is increased compared to regular Isomap, while the similarity between neighboring samples of different classes is decreased. Most notably, the similarity between non-neighboring samples remains zero, regardless of whether their labels are identical. This essentially means that each class can be split into multiple (potentially non-connected) components of the similarity graph, therefore allowing non-linear classification of patterns.

Neural network-based approaches Any deep neural network tasked with classification essentially performs *sup.* FC in its hidden layers, in the same way that AutoEncoders tasked with reconstruction perform *unsup.* FC. However, this will in the general case lead to constructed features which do not necessarily preserve the information of the original features ; considering for instance a supervised classifier network involving one hidden layer Z_d^*, the network may learn to encode class information in a single latent variable, ignoring the $(d-1)$ others, s.t. $\boldsymbol{z_1^*} = \boldsymbol{y}$ and $\forall i \in [2, \dots, d], \boldsymbol{z_i^*} = \mathbf{0}$.

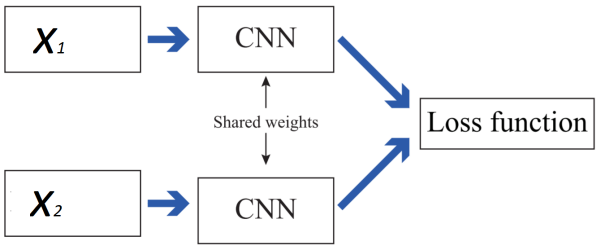

FIGURE 3.10: The structure of a Siamese Convolutional Neural Network. From Liu et al. (2018).

In order to avoid such pathological configurations, neural network-based *sup.* FC techniques explicitely encourage *all* constructed features to discriminate between classes via the objective function. Liu et al. (2018) proposed to use a *siamese* (Bromley et al., 1994) *Convolutional Neural Network (CNN)* (Krizhevsky et al., 2012) to learn the constructed features (fig. 3.10).

Defining, with a slight abuse of notation, $\delta(\boldsymbol{x_1}, \boldsymbol{x_2}) = ||\boldsymbol{Z_d^*(x_1)} - \boldsymbol{Z_d^*(x_2)}||_2$ as the Euclidean distance between the two input datapoints in the constructed feature space, $\delta(\boldsymbol{x_1}, \boldsymbol{x_2})$ is denoted δ_+ if $y_1 = y_2$ and δ_- if $y_1 \neq y_2$. Given μ a margin parameter, the final loss function of the Siamese CNN is then defined as :

$$L(\boldsymbol{x_1}, \boldsymbol{x_2}) = \max(0, \delta_+ - \delta_- + \mu) \tag{3.6}$$

The objective function from equation (3.6) essentially incentivizes minimizing the distance in Z_d^* between points of same label, while maximizing the distance between samples from different classes. This approach therefore corresponds to an non-linear adaptation, in neural network form, of the principle that similar points should be mapped close together while dissimilar points should be as remote as possible from one another in the embedding. This core idea is also prominent in FS, which we will now introduce and discuss.

3.2 Feature selection

As seen in chapter 2, FS is a particular of FC where the exploration for the optimal feature set Z_d^* (w.r.t. cost function C) is limited to a small region of the search space corresponding to the C_D^d subsets of the original feature set F containing d elements. Despite this restriction, FS is not a "simpler" version of the DR problem. On the contrary, the FS setting involves multiple new challenges and concepts absent from the general case of FC. The most prominent of these specificities are discussed in section 3.2.1. Section 3.2.2 thereafter provides an overview of the most impactful methods in the field of *sup.* FS. *Unsup.* FS is

subsequently studied in section 3.2.3. The algorithmic contribution of this thesis (chapter 4) pertains to the latter setting.

Diversity of input data representations

Three implicit assumptions were made regarding the input data during the general background presentation of chapter 2 :

Assumption 1. *All original features are available from the beginning of the FS process.*

Assumption 2. *There are no structural relationships between original features*[3]*.*

Assumption 3. *All original features come from the same source.*

Most FS algorithms from the literature (both *sup.* and *unsup.*) are designed to handle tasks where all three above assumptions hold, thus correspond to the so-called *traditional FS* context. However, the rise of Big Data has led to an increased prevalence of real-life applications where at least one of these assumptions no longer holds (Li and Liu, 2017).

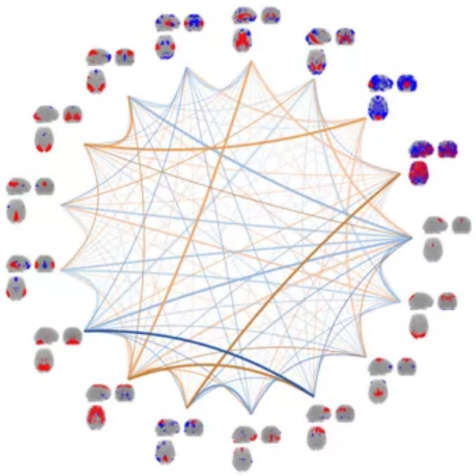

FIGURE 3.11: The spatial arrangement of the voxels is an important source of information neglected by traditional FS.

In the context of sentiment analysis using text data extracted from the Twitter social network, each feature correspond to one word of vocabulary. Given that new slang words are generated by the users every day, the size of the feature set is constantly growing, thus falsifying assumption 1. This setting corresponds to *FS for streaming data*.

3. More specifically, even though these relationships may exist, we assume no prior information or expert knowledge regarding these structures

In neuroimaging, features will typically correspond to voxels spatially arranged in a three-dimensional space so as to mirror the anatomy of the human brain (Jenatton et al., 2011) (fig. 3.11). This graph-like structure falsifies assumption 2 and is not exploited by traditional FS algorithms. This setting corresponds to *FS with structured features*.

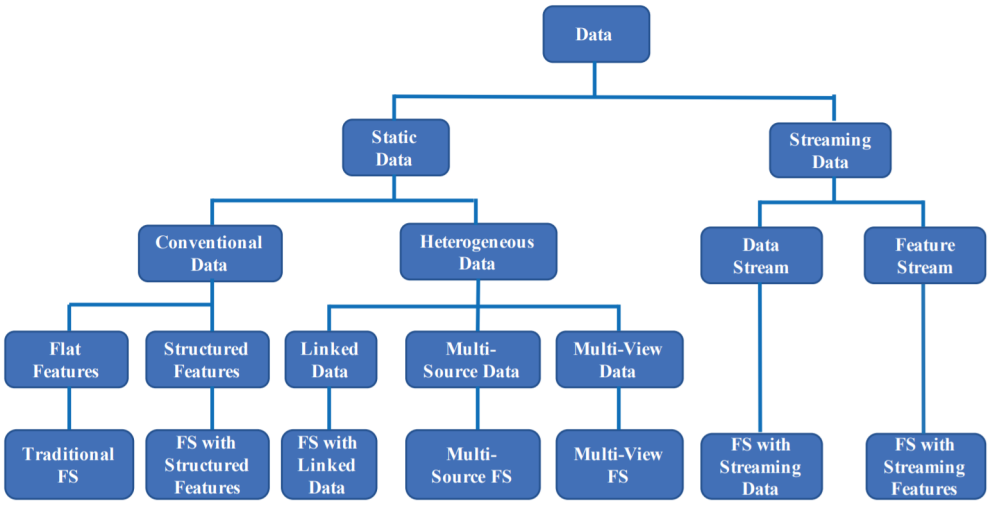

FIGURE 3.12: Taxonomy of FS from a data-driven perspective. From Li et al. (2018a)

In bioinformatical cancer research, different types of genetic material (e.g. DNA and RNA) are simultaneously exploited for predicting tumors (Zhao and Liu, 2008). Original features are therefore obtained from multiple sources different in nature [4]. This setting corresponds to *Multi-View FS*.

We refer the interested reader to Li et al. (2018a) for a thorough review of FS methods in non-traditional settings, the taxonomy of which is presented in figure 3.12.

We will in the context of this thesis focus on traditional FS. It is however important to note that some traditional FS techniques can be seamlessly adapted to other settings. For instance, *group LASSO* (Yuan and Lin, 2007) (more in section 3.2.2) is by construction well-suited for tree-like structured features (fig. 3.13), leading to its *tree-guided group LASSO* (Liu and Ye, 2010) variant.

Hierarchical group structures corresponding to figure 3.13 also provide a direction for future refinement of our algorithmic contribution (chapter 7).

4. One could also consider the previous example of social network sentiment analysis, where features are extracted from different data formats such as text, image or video

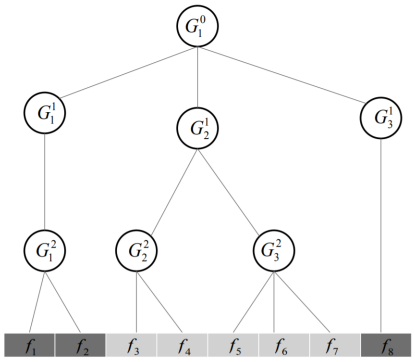

FIGURE 3.13: Example of features forming groups in a tree-like structure. From Li et al. (2018a).

3.2.1 Particularities of the Feature Selection setting

3.2.1.1 Independent, collective and semi-independent scoring

As said earlier, the aim of DR is in the general case to find $Z_d^* = \arg\min_{Z_d \in \mathcal{H}_d} C(F, Z_d)$. In the FS setting, the cost function C is responsible for estimated the quality of the selection subset. In other words, C is in this case a *scoring function* judging the "relevance" (w.r.t. a learning goal) of the original features[5]. Scoring techniques can be categorized w.r.t. the context in which relevance is assessed.

Independent scoring Kohavi and John (1997) proposed a first tentative definition of feature relevance in the *sup.* context by interpreting the features $\boldsymbol{f_1}, \ldots, \boldsymbol{f_D}$ and the learning goal $\boldsymbol{y}$ as random variables drawn from an underlying joint distribution $P(F, \boldsymbol{y})$. With a slight abuse of notation, the relevance of an original feature is defined as :

$$\forall i \in [1, \ldots, D], \text{Relevance}(f_i) = |P(\boldsymbol{f_i}, \boldsymbol{y}) - P(\boldsymbol{f_i})P(\boldsymbol{y})| \tag{3.7}$$

It follows from equation (3.7) that a feature is irrelevant *iff* it is independent from the labels. The stronger the dependency between $\boldsymbol{f_i}$ and $\boldsymbol{y}$, the more relevant $\boldsymbol{f_i}$.

Given that state-of-the-art *unsup.* FS methods typically construct an artificial learning goal $\widehat{\boldsymbol{y}}$ (more in section 3.2.3), the above definition of relevance is applicable in both *sup.* and *unsup.* contexts.

In practice, given that the underlying joint distribution is unknown, relevance can hardly be computed empirically in this way. However, this definition is an integral part of prominent scoring criteria such as the *Fisher score* (Duda et al., 2000) or the supervised version of the *Laplacian score* (He et al., 2005) (more in sections 3.2.2 and 3.2.3).

Most importantly, the relevance estimation of a particular original feature does not involve any of the other features. We will therefore in the following refer to FS approaches relying on such scoring functions as *independent scoring methods*.

5. In the *unsup.* declination of the FS problem, a common alternative (He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012) is to define *two* cost functions C_1 and C_2. C_1 is tasked with scoring the features. C_2 is responsible for assessing the quality Z_d^* after the selection process is complete. By contrast with the *unsup.* C_1, C_2 is often a *sup.* estimator (more in chapter 5)

Limitations of independent scoring methods Independent scoring methods are by construction plagued by two opposite issues : *false negatives* (features falsely considered irrelevant) and *false positives* (falsely relevant features).

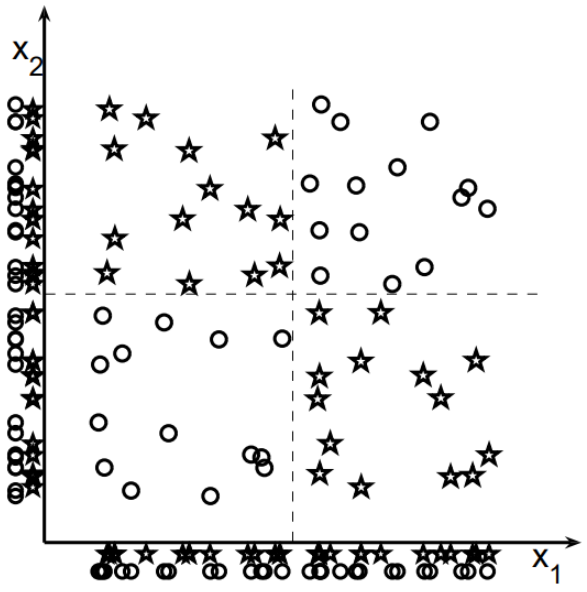

FIGURE 3.14: Independent scoring methods are unable to detect that although X_1 and X_2 are individually irrelevant, they perfectly separate the classes (stars and circles) if considered together. From Guyon and Elisseeff (2003).

Figure 3.14 provides an illustration of the false negative problem on the well-known XOR example : X_1 and X_2 are both individually irrelevant and do not separate the classes at all. However, taken jointly, they allow for a perfect non-linear separation.

Figure 3.15 illustrates the opposite issue of false positives ; X_1 and Y are dependent (left panel), but become independent conditionally to X_2, meaning X_1 is not truly relevant. This phenomenon is known as Simpson's paradox (Simpson, 1951), and is caused by feature redundancy : the information carried by X_1 is already accounted for in X_2 (X_2 however contains additional information absent from X_1). In light of these issues, the main benefit of independent scoring is its lower computational complexity relatively to both *collective* and *semi-independent* scoring, which we now introduce.

Semi-independent scoring $\forall i \in [1, \dots, D]$, let $F^{\setminus i}$ denote the subset of F containing every feature except $\boldsymbol{f_i}$. Following Kohavi and John (1997), the notion of feature relevance can be extended to tackle the

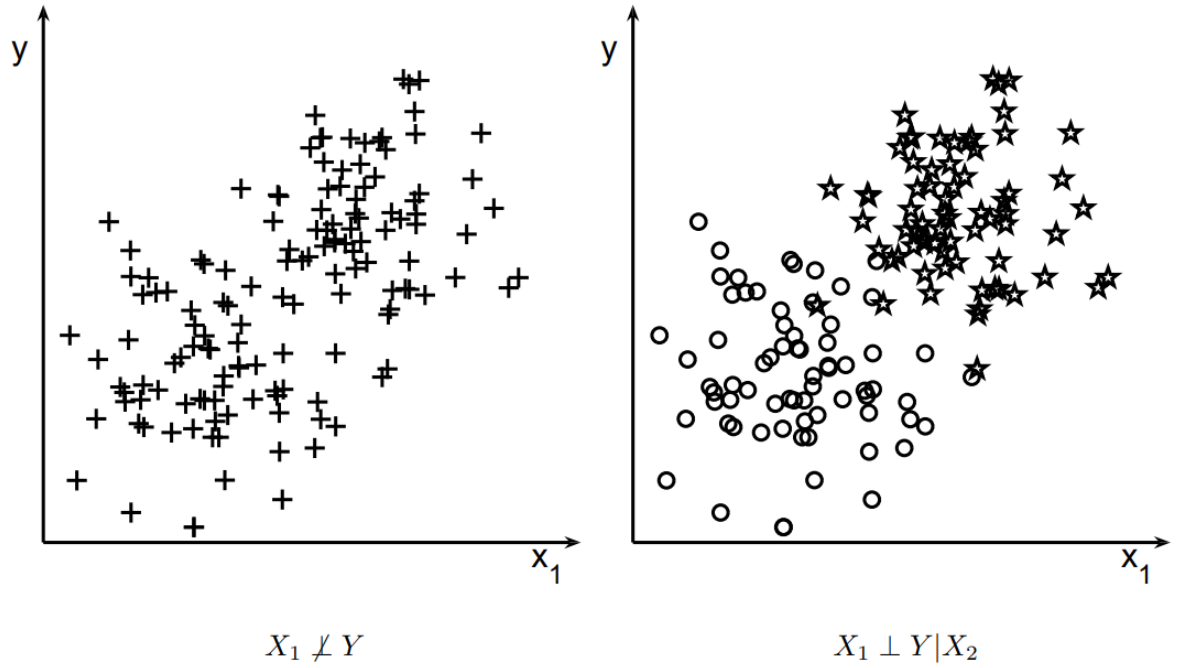

FIGURE 3.15: On the left panel, X_1 appears relevant. However, it becomes irrelevant conditionally to X_2 (denoted by circles and stars on the right panel). From Guyon and Elisseeff (2003).

aforementioned limitations :

$$\text{Relevance}(\boldsymbol{f_i}) = \sum_{S \subset F^{\setminus i}} |P(\boldsymbol{f_i}, \boldsymbol{y}|S) - P(\boldsymbol{f_i}|S)P(\boldsymbol{y}|S)| \tag{3.8}$$

Following equation (3.8), $\boldsymbol{f_i}$ is considered irrelevant *iff* it is independent from the learning goal conditionally to *any* combination of other features, indicating that the information carried by $\boldsymbol{f_i}$ (if any) is already accounted for elsewhere. This new definition solves the problem of false negatives [6].

By contrast with independent scoring, the process of estimating the relevance of an element of F involves *all* features. Nevertheless, each feature is still assigned its own individual score. Therefore, we will in the following refer to FS methods relying on this paradigm as *semi-independent scoring* techniques.

Collective scoring A third possible scoring approach consists in defining relevance on a subset level only, so that features are assessed as groups rather than individuals. This approach is intuitively well-suited for tree-like feature structures (fig. 3.13). However, the number of score estimates needed to fully explore the

6. It is however not enough to avoid false positives. In order to avoid Simpson's paradox, Kohavi and John (1997) further differentiate between *weakly* and *strongly* relevant features, with the idea of selecting only strongly relevant ones. However, the definitions of weak and strong relevance are unpractical, and bear mostly historical significance.

search space rises from D (one per original feature) to C_D^d (one per subset of F containing d elements). Exhaustive exploration is thus intractable.Gaudel and Sebag (2010) envision the search for Z_d^* as navigating through a decision tree, which is aggressively pruned to avoid considering unpromising candidate subsets. Such algorithms are in the following referred to as *collective scoring* approaches.

3.2.1.2 Forward, backward or simultaneous selection

Once scoring function C has been defined, three differing strategies are available to construct Z_d^* based on C.

Forward selection The first selection strategy consists in an iterative process, of which pseudocode is provided in Algorithm 1 [7] :

Algorithm 1 Forward selection

Input : Feature set $F = \{\boldsymbol{f_1}, ..., \boldsymbol{f_D}\}$
Parameter: Selection subset size d
Output : Selection subset Z_d
Initialize $Z_d = \emptyset$ and candidate set $S = F$
Repeat
for $f \in S$ **do**
  Compute $C(f)$
end
Determine best candidate f_{best} w.r.t. C
$Z_d \leftarrow (Z_d \cup f_{best})$
$S \leftarrow (S \backslash f_{best})$
until $|Z_d| = d$
Return Z_d.

Following alg. 1, *forward selection* (Guyon and Elisseeff, 2003) is a "bottom-up" approach, building the selection subset from the ground up.

Backward selection The second selection strategy is also an iterative process, described in algorithm 2 :

7. This algorithm is written from the perspective of semi-independent scoring. Nevertheless, its pseudocode can be slightly modified to adopt the point of view of collective scoring, without loss of generality.

Algorithm 2 Backward selection

Input : Feature set $F = \{\boldsymbol{f_1}, ..., \boldsymbol{f_D}\}$
Parameter: Selection subset size d
Output : Selection subset Z_d
Initialize $Z_d = F$
Repeat
for $f \in Z_d$ **do**
  Compute $C(f)$
end
Determine *worst* candidate f_{worst} w.r.t. C
$Z_d \leftarrow (Z_d \backslash f_{worst})$
until $|Z_d| = d$
Return Z_d.

By alg. 2, *backward selection* (Guyon and Elisseeff, 2003) (also commonly referred to as *Recursive Feature Elimination (RFE)*) is a "top-down" method, pruning the selection subset down to the desired size d.

Simultaneous selection This last strategy performs selection in a single pass rather than iteratively (alg. 3) :

Algorithm 3 Simultaneous selection

Input : Feature set $F = \{\boldsymbol{f_1}, ..., \boldsymbol{f_D}\}$
Parameter: Selection subset size d
Output : Selection subset Z_d
Initialize $Z_d = \emptyset$
for $f \in F$ **do**
  Compute $C(f)$
end
Determine the d best candidates $(f^1_{best}, \ldots, f^d_{best})$ w.r.t. C
$Z_d \leftarrow \{f^1_{best}, \ldots, f^d_{best}\}$
Return Z_d.

Discussion If C defines an independent scoring process, then the order in which features are selected/discarded is clearly irrelevant. The above three strategies are therefore equivalent in that case. Independent scoring methods therefore opt for simultaneous selection, given its lower computational cost relatively to both forward and backward approaches.

On the other hand, if C pertains to either semi-independent or collective scoring, then each strategy likely leads to different results. An illustration of the respective weaknesses is provided in figure 3.16.

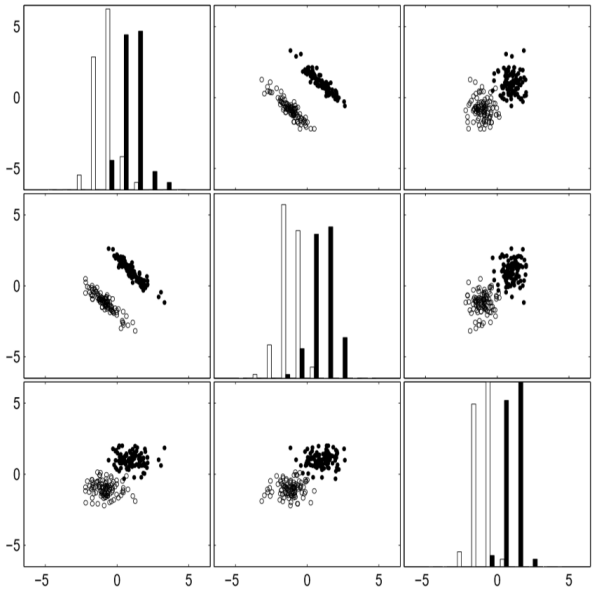

FIGURE 3.16: A binary classification task with three features. $\forall(i,j) \in [1,3]^2$, panel (i,j) depicts the separation of the two classes (resp. in black and white) achieved by selection subset $\{f_i, f_j\}$. The diagonal panels therefore resp. correspond to the singleton selection subsets $\{f_1\}, \{f_2\}, \{f_3\}$. Taken from Guyon and Elisseeff (2003).

In this *sup.* example, f_3 is the most individually relevant feature, given that it achieves the best class separation on its own (bottom-right panel). A forward selection method will therefore choose f_3 first. However, if tasked with selected a second candidate, then potential forward selections $\{f_{3,f\,1}\}$ and $\{f_3, f_2\}$ are both worse predictors than $\{f_1, f_2\}$, corresponding to the result of backward selection.

In this example, going backwards is thus better than forward for $d = 2$, and worse for $d = 1$. This is a consequence of the *greedy*[8] nature of both approaches.

Informally, it appears that the lower the selection ratio $\frac{d}{D}$, the better forward selection comparatively to RFE. In practice, both semi-independent and collective FS algorithms rely on either simultaneous (He et al., 2005; Li et al., 2012) or backward selection (Guyon et al., 2002; Ye and Sun, 2018), while forward selection is seldom implemented in recent approaches.

3.2.1.3 Filters, wrappers and embedded methods

As said, the end goal of FS (and even DR in general) is to help a learning algorithm, e.g. a supervised classifier or an unsupervised regressor. An additional way of categorizing FS approaches (complimentary to the type and ordering of feature scoring) is therefore via the relationship between the FS technique and the learning algorithm. An illustration of the three resulting groups of methods is provided in fig. 3.17.a.

8. Greedy is here used to reflect the fact that earlier selection/elimination decisions are never revisited in light of later decisions.

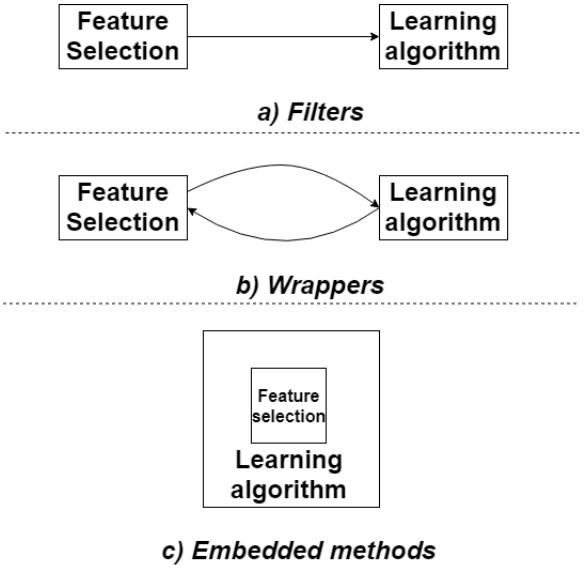

FIGURE 3.17: Schematic of filters, wrappers and embedded methods.

Filters (Duda et al., 2000; He et al., 2005)(panel a) of figure 3.17) act as a pre-processing step to the learning algorithm, the latter being uninvolved in the selection.

Wrappers (Huang et al., 2007)(panel b)) also correspond to a pre-processing step. By contrast with filters, wrappers define an iterative selection pipeline. At each step, the learning algorithm is ran using the candidate selection subset as input. The eventual performance of learning is then leveraged to guide the search for the optimal selection subset Z_d^* during the next step.

Embedded methods (Guyon et al., 2002)(panel c)) represent a paradigm shift. FS is performed *online* during the execution of the learning algorithm, rather than in pre-processing. The underlying idea is to leverage partial results of learning (such as the parameters of a neural network) to orient the search for Z_d^*.

All three aforementioned techniques are clearly applicable in the *sup.* context, where the learning algorithm typically corresponds to a supervised classifier. The main motivation for choosing an adequate selection strategy is then the computational cost. Embedded methods require few runs of the learning algorithm (possibly with a warm start), while filters require none at all. By contrast, wrappers involve running the learning algorithm multiple times. This constitutes a significant downside of wrappers in many application domains. In *sup.* image classification for instance [9], training a deep convolutional network hundred of times is typically unaffordable.

As a result, *sup.* FS most often correspond to either filters (Duda et al., 2000) or embedded methods (Guyon et al., 2002). Selection resulting of embedded FS likely yields higher learning performance than for filters, given that Z_d^* is tailor-made for the learning algorithm considered. On the flipside, filters lead to selection

9. Selecting individual pixels in a high-resolution image is hardly effective for prediction. Therefore, Computer Vision applications of FS (Chen, 2015) are concerned with identifying the best candidates among objects of greater scale, e.g. retaining the most informative convolutional filters.

subsets of higher generalization power. This means that if the learning algorithm is modified (e.g. hidden layers are added to a deep neural network), filter-based FS is invariant, whereas the learning performance resulting of embedded FS is likely degraded.

This independence property of selection w.r.t. subsequent learning is thereafter called *agnosticism*. This notion is sought for and extended in both our algorithmic and methodological contributions (chapters 4 and 5).

In the *unsup.* context, the eventual learning algorithm (if any) is typically unknown at the time of FS. Consequently, both wrappers and embedded method are ill-suited to this setting. To the best of our knowledge, all of the most impactful traditional *unsup.* FS approaches (He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012) are filters. A direction for future research would be to refine our algorithm AGNOS into an embedded *unsup.* method (chapter 7).

3.2.2 Supervised Feature Selection

This section introduces the most well-known *sup.* FS algorithms. For each approach, the scale of the scoring function (independent, semi-independent or collective), the order of assessment (simultaneous, forward selection or recursive elimination) and the link with the learning algorithm (filter, wrapper or embedded) are stated after the description of the method.

ReliefF The early *ReliefF* (Kira and Rendell, 1992; Kononenko, 1994) searches for two specific neighbors of any given point $\boldsymbol{x_i}$: the closest (w.r.t. the Euclidean distance) observation with the same label (the *nearest hit* $\boldsymbol{n_h(i)}$) and the closest observation with a different label (the *nearest miss* $\boldsymbol{n_m(i)}$). The score of feature f_j is then s.t. :

$$\forall j \in [1, \dots, D], S(f_i) = \frac{1}{n} \sum_{i=1}^{n} \left(|f_j(x_i) - f_j\left(n_m(i)\right)|^2 - |f_j(x_i) - f_j\left(n_h(i)\right)|^2 \right) \tag{3.9}$$

In order to maximize this score, an informative feature should take similar values on neighboring points of the same class and as distinct values as possible on differently labeled neighboring points. Top ranked features should thus support stark separation of the classes and lead to a high prediction accuracy. Eq. (3.9) seems to define an independent scoring method. However, it is important to note that the nearest misses and hits have been determined by considering all original features. RelieF is accordingly a semi-independent approach.

Keywords : *Filter, semi-independent scoring, simultaneous selection*

Fisher score The *Fisher score* (Duda et al., 2000) was introduced as an alternative to RelieF inspired by the Fisher discriminant (section 3.1.2). Considering a classification task with c classes, n_i denotes the number of samples in the i-th class, $\mu_j(i)$ and $\sigma_j(i)$ respectively the mean and variance of the j-th feature on the i-th

class, and μ_j the mean of the j-th feature on the whole dataset. The Fisher score of $\boldsymbol{f_j}$ is then s.t. :

$$\forall j \in [1, \dots, D], S(\boldsymbol{f_j}) = \frac{\sum_{i=1}^{c} n_i(\mu_j(i) - \mu_j)^2}{\sum_{i=1}^{c} n_i \sigma_j(i)^2} \tag{3.10}$$

Equation (3.10) implements the same core idea as RelieF, that the most informative features are those presenting a large contrast between different classes. However, contrarily to RelieF, the Fisher score is an independent scoring method and suffers from the issues presented in section 3.2.1.1. A generalized Fisher score has later been proposed in Gu et al. (2012), turning to collective scoring instead.
Keywords : *Filter, independent scoring, simultaneous selection*

RFE-SVM *Support Vector Machines (SVM)* (Cortes and Vapnik, 1995) are a type of sup. classification algorithms. The goal of a linear SVM is to find the unique hyperplane (often referred to as *decision boundary*) meeting two requirements in the separable case : i) all datapoints sharing a common label are on the same side of the border and ii) the distance between the border and the closest datapoint (thereafter called *margin*[10]) is maximal among all hyperplanes fulfilling condition i). This process is illustrated in figure 3.18, where H_1 is a poor separator. H_2 achieves perfect separation, but with only a small margin. H_3 is the optimal hyperplane maximizing inter-class margin.

The process of fitting the decision boundary corresponds to solving the following constrained optimization problem for $\boldsymbol{w}^*$ and b^* :

$$\begin{cases} \boldsymbol{w^*} = \arg\min_{\boldsymbol{w} \in \mathbb{R}^D} ||\boldsymbol{w}||_2 \\ \forall i \in [1, \dots, n], y_i(\boldsymbol{w^*}.\boldsymbol{x_i} - b^*) \geq 1 \end{cases} \tag{3.11}$$

By definition, the learned weight vector $\boldsymbol{w} = (w_1, \dots, w_D)$ involves one component per feature. *Recursive Feature Elimination-Support Vector Machine (RFE-SVM)* (Guyon et al., 2002) is an iterative embedded FS method leveraging this observation, of which pseudocode is provided in algorithm 4.

Algorithm 4 RFE-SVM

Input : Feature set $F = \{\boldsymbol{f_1}, ..., \boldsymbol{f_D}\}$
Parameter: Selection subset size d
Output : Selection subset Z_d
Initialize candidate set $Z_d = F$
Repeat
Train SVM from Z_d, producing $\boldsymbol{w} = (w_1, \dots, w_D)$
Find $\boldsymbol{f_{worst}}$ s.t. $w_{worst} = \arg\min_{w_j} w_j^2$
$Z_d \leftarrow Z_d \backslash \{\boldsymbol{f_{worst}}\}$
until $|Z_d| = d$
Return Z_d.

10. Admittedly, this formulation slightly diverges from the formal definition of margin, which was here simplified for the sake of brevity without loss of generality in the following discussion.

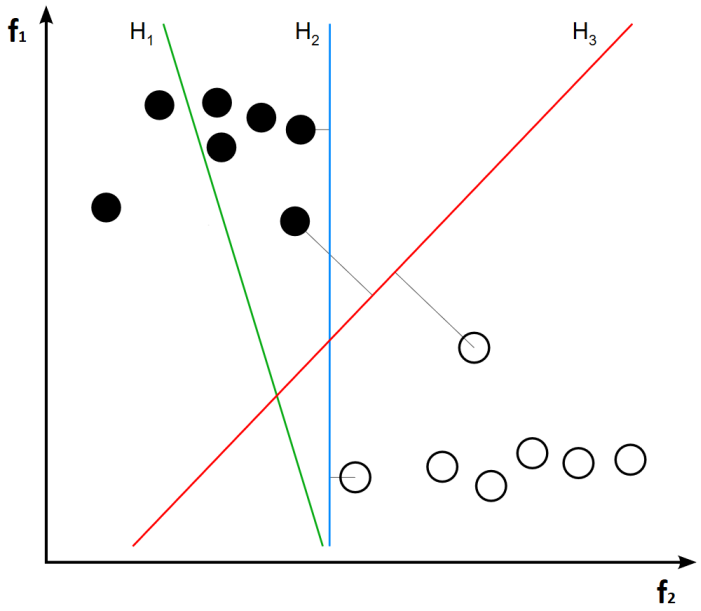

FIGURE 3.18: Illustration of a linear SVM on a binary classification task. Decision boundary H_3 maximizes the margin between the two classes (in black and white). Feature $\boldsymbol{f_1}$ should be prioritized over $\boldsymbol{f_2}$ for selection. From Wikipedia.

Even though eliminating the features with the *smallest* associated weights (alg. 4) may seem counter-intuitive (given that the goal of the optimization problem in eq. (3.11) is to minimize $||w||_2$), it is well-grounded in theory ; the following explanation is also the core motivation of many neural network-based FS approaches (Setiono and Liu, 1997; De et al., 1997; Steppe and Bauer, 1996; Zurada et al., 1997; Ye and Sun, 2018), the most impactful of which will be introduced thereafter.

Keywords : *Embedded method, semi-independent scoring, backward selection/Recursive Feature Elimination*

Motivation of RFE-SVM and neural network-based FS In the *sup.* context, label information allow considering the performance of a classifier as a collective scoring criterion. In order to avoid the expensive combinatorial problem inherent to collective scoring, a semi-independent criterion can be derived from the collective one, by analyzing the sensitivity of the classifier performance w.r.t. the removal of each feature from the input.

Formally, $\forall j \in [1, \dots, D], \forall S \subset F$, let $D_J(S, f_j) = J(S \backslash \{f_j\}) - J(S)$ denote the change in classifier performance J induced by the elimination of f_j from S. Intuitively, removing irrelevant features should hardly degrade performance. Therefore, the lower $D_J(S, f_j)$, the less relevant f_j. Note that positive values for $D_J(S, f_j)$ are possible, indicating an increase in performance due to removing harmful noise from the classifier input. Computing the exact value of $D_J(S, f_j)$ is computationally costly, as it requires training the classifier multiple times. A good approximation of it can however be obtained using the following trick : eliminating f_j can be

simulated by setting all weights associated to f_j to 0. That is, in the case of RFE-SVM :

$$\forall j \in [1, \dots, D], \forall S \subset F, D_J(S, f_j)) = J(S \text{ with } w_j \leftarrow 0) - J(S \text{ with } w_j \text{ untouched }) \tag{3.12}$$

A second order Taylor expansion of J around w_j gives :

$$D_J(S, f_j) = \frac{\partial J}{\partial w_j} w_j + \frac{1}{2} \frac{\partial^2 J}{\partial^2 w_j} w_j^2 \tag{3.13}$$

Given that J is examined after classifier training has converged, it lies in a local maximum, and the first order derivative can be neglected. Thus :

$$D_J(S, f_j) = \frac{1}{2} \frac{\partial^2 J}{\partial^2 w_j} w_j^2 \tag{3.14}$$

In the context of a linear SVM, $J(S) = \frac{1}{2} \|w\|^2$, thus $\forall j \in [1, \dots, D], \frac{\partial^2 J}{\partial^2 w_j} =$ const. . This leads to :

$$D_J(S, f_j) \propto w_j^2 \tag{3.15}$$

Equation (3.15) motivates eliminating the features with the smallest weights in linear RFE-SVM. Similar demonstrations can be obtained for non-linear SVMs, as well as for neural networks.

Neural network-based FS Over the years, the idea of eliminating the features with the smallest associated weights or decrease in classifier performance sprouted many neural network-based *sup.* FS approaches (Setiono and Liu, 1997; De et al., 1997; Steppe and Bauer, 1996; Zurada et al., 1997; Ye and Sun, 2018). These methods involve slight variations in implementation (e.g. adding a threshold value on the weights for the purpose of feature elimination). Most importantly, these algorithms differ on the type of weight regularization used.

Both the *Signal Noise Ratio (SNR)*[11] (Bauer et al., 2000) and *Feature Quality Index (FQI)* (De et al., 1997) approaches do not implement any kind of weight regularization. *Neural Network Feature Selector (NNFS)* (Setiono and Liu, 1997) adds a L_2 regularization term (also known as *weight decay*) to the loss function of the network. The Drop-Out-One (Ye and Sun, 2018) refinement instead relies on a sparse group-LASSO (Simon et al., 2013) penalty on the weights to enforce sparsity in the selection. This regularized regression technique will be further discussed below.

The *Deep Feature Selection (DFS)* (Li et al., 2016) approach is particularly relevant to the algorithmic contribution of this thesis. DFS considers an alternate neural architecture, in which a sparse one-to-one linear layer is added between the input and the first hidden layer, as illustrated in figure 3.19.

The loss function of the network is then augmented with a sparse group-LASSO penalty term on the weights of this additional layer. This architecture holds the practical benefit that the importance of input feature

11. This method consists in agumenting F artificial Gaussian noise features, and selecting only features which associated weights are significantly larger than for the noisy features, at the end of training.

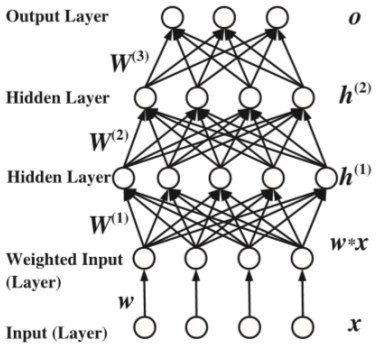

FIGURE 3.19: The Deep Feature Selection architecture. From Li et al. (2016).

f_j is now condensed in the single real-valued *slack variable* w_j, rather than inferred from an $\mathbb{R}^{D\times d}$ weight matrix.

A limitation of *sup.* neural network-based FS is that the sensitivity of the classifier performance w.r.t. the input tends to decrease as the number of hidden layers in the network increases (Pascanu et al., 2013). This corresponds to the so-called *vanishing gradient* (Hochreiter, 1998) issue. Roy et al. (2015) report that the most important initial features are hardly identifiable empirically for networks of depth ≥ 3 using SNR, FQI or NNFS.

As will be seen, our algorithmic contribution (chapter 4)is the first attempt at extending regularized neural network-based FS to the *unsup.* context.

Keywords : *Wrappers/Embedded methods, Semi-independent scoring, Backward selection/Recursive Feature Elimination*

Least Absolute Shrinkage and Selection Operator (LASSO) *Ordinary Least Squares (OLS)* (Goldberger, 1964) corresponds to the linear regression problem of finding $\boldsymbol{\beta}^* \in \mathbb{R}^D$ s.t. :

$$\boldsymbol{\beta}^* = \underset{\beta\in\mathbb{R}^D}{\arg\min} \sum_{i=1}^{n} ||y_i - \langle \boldsymbol{x_i}, \boldsymbol{\beta}\rangle||_2^2 \tag{3.16}$$

OLS is however prone to overfitting in the small n, large D regime (few datapoints, many features). In order to combat this issue, a tentative solution is to add an L_2 *regularization* penalty term to the optimization problem, parameter $\lambda \geq 0$ governing the severity of the penalization :

$$\beta^* = \underset{\beta\in\mathbb{R}^D}{\arg\min} \left(\sum_{i=1}^{n} ||y_i - \langle x_i, \beta\rangle||_2^2 \right) + \lambda||\beta||_2 \tag{3.17}$$

The vector $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_D^*)$ reflects the importance of the respective original features in the regression. However, by virtue of the L_2 geometry, $\boldsymbol{\beta}^*$ optimized via eq. (3.16) is rotationally invariant, as illustrated in the leftmost panel of figure 3.20. This means that *all* features are likely associated to non-zero coefficients. An L_2 penalty term is thus hardly discriminative.

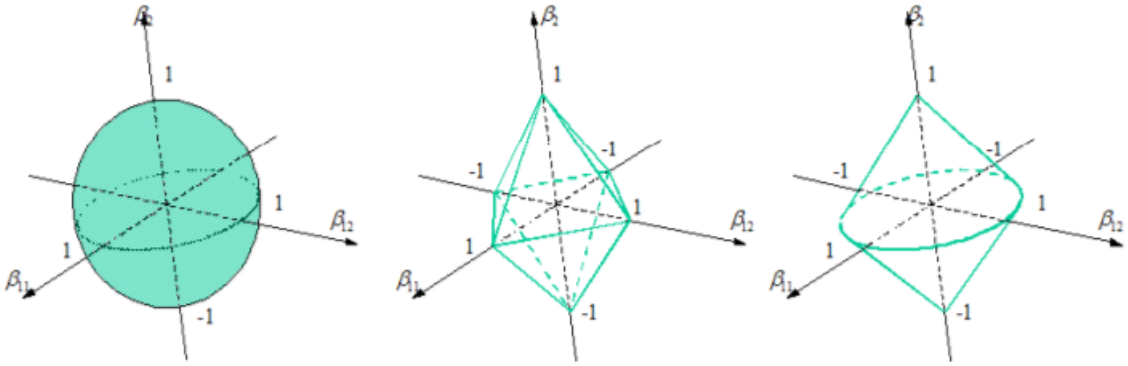

FIGURE 3.20: Outline of different penalty functions : L_2 (left), LASSO (center), group-LASSO (right).

In order to enforce sparsity in the solution, Tibshirani (1996) introduced the *Least Absolute Shrinkage and Selection Operator* (LASSO) technique, which adds a L_1 penalization term instead :

$$\boldsymbol{\beta^*} = \underset{\boldsymbol{\beta}\in\mathbb{R}^D}{\arg\min} \left(\sum_{i=1}^{n} ||y_i - \langle x_i, \boldsymbol{\beta}\rangle||_2^2 \right) + \lambda ||\boldsymbol{\beta}||_1 \tag{3.18}$$

This regularization enforces sparsity among the coefficients $\beta_j^*, j \in [1, \dots, D]$. This property stems from the nature of the L_1 geometry, in which the penalty function treats the coordinate directions differently from all other directions (center panel of figure 3.20). As a result of sparsity, the LASSO induces a supervised FS technique, discarding f_j *iff* $\beta_j^* < \epsilon$.

In order to obtain the best of both worlds of the L_1 and L_2 geometries, Yuan and Lin (2007) introduced the *group-LASSO*, in which the feature set is first partitioned in groups $G_1, \dots, G_k$. $|G_i|$ denoting the size of the i-th feature group, the $L_{2,1}$ penalized regression scheme reads :

$$\boldsymbol{\beta^*} = \underset{\boldsymbol{\beta}\in\mathbb{R}^D}{\arg\min} \left(\sum_{i=1}^{n} ||y_i - \langle \boldsymbol{x_i}, \boldsymbol{\beta}\rangle||_2^2 \right) + \lambda \sum_{i=1}^{k} \sqrt{|G_i|} \sqrt{\sum_{j\in G_i} \beta_i^2} \tag{3.19}$$

This hybrid geometry leads to sparsity at the group level (discarding as many groups of features as possible) while preserving the rotational invariance of the solution within each group (rightmost panel of figure 3.20) (Bach, 2008).

Many variants of the group LASSO have since been proposed to achieve specific sparsity and invariance properties (Meier et al., 2008; Simon et al., 2013; Ivanoff et al., 2016). Most notably, the *sparse group LASSO*, also referred to as *elastic net*, reads (Simon et al., 2013) :

$$\boldsymbol{\beta^*} = \underset{\boldsymbol{\beta}\in\mathbb{R}^D}{\arg\min} \left(\sum_{i=1}^{n} ||y_i - \langle \boldsymbol{x_i}, \boldsymbol{\beta}\rangle||_2^2 \right) + (1-\alpha)\lambda \sum_{i=1}^{k} \sqrt{|G_i|} \sqrt{\sum_{j\in G_i} \beta_i^2} + \alpha\lambda ||\boldsymbol{\beta}||_1 \tag{3.20}$$

$\alpha \in [0, 1]$ governs the convex combination of the LASSO and group LASSO. The aim of the elastic net is to achieve sparsity of the solution both at the group level and within each group, and has become a useful regularization tool for training *sup.* Neural Networks on high-dimensional data (Feng and Simon, 2017).
Keywords : *Filters, semi-independent scoring, simultaneous selection*

Discussion

During this review of *sup.* FS methods, the central idea has remained constant : "*datapoints of the same class should be neighbors, while datapoints of different classes should be strangers*". Features are selected or rejected based on how well they reflect this desired structure. Interestingly enough, this same idea is also at the core of *unsup.* FS, as will be shown in the next section.

An additional observation of particular interest to this thesis is that neural network-based FS approaches such as Ye and Sun (2018) actually combine FC and FS, in the *sup.* context : features are selected w.r.t. their importance to build the constructed features in the hidden layers of the network.

3.2.3 Unsupervised FS

By contrast with the *sup.* context, all methods presented in the following are *filters* implementing a *semi-independent* feature scoring criterion, based on *spectral clustering theory* (Von Luxburg, 2007).

Spectral clustering Let sim denote a similarity metric on the instance space, e.g. $sim(\boldsymbol{x_i}, \boldsymbol{x_j}) = exp\{-\|\boldsymbol{x_i} - \boldsymbol{x_j}\|_2^2\}$ and M the $n \times n$ matrix with $M_{i,j} = sim(\boldsymbol{x_i}, \boldsymbol{x_j})$. Let Δ be the diagonal degree matrix associated with M, i.e. $\Delta_{ii} = \sum_{k=1}^{n} M_{ik}$, and $\widetilde{L} = \Delta^{-\frac{1}{2}}(\Delta - M)\Delta^{-\frac{1}{2}}$ the normalized Laplacian matrix associated with M.

Spectral clustering relies on the diagonalization of $\widetilde{L}$, with λ_i (resp. $\boldsymbol{\xi_i}$) the eigenvalues (resp. eigenvectors) of $\widetilde{L}$, with $\lambda_i \leq \lambda_{i+1}$. Informally, the $\boldsymbol{\xi_i}$ are used to define soft cluster indicators (i.e. the degree to which x_k belongs to the i-the cluster being proportional to $\langle \boldsymbol{x_k}, \boldsymbol{\xi_j} \rangle$), with λ_k measuring the inter-cluster similarity (the smaller the better).

The general unsupervised clustering scheme proceeds by clustering the samples and falling back on supervised feature selection by considering the clusters as if they were *pseudo-labels* ; more precisely, the features are assessed depending on how well they separate clusters.

Laplacian score The *Laplacian score* (He et al., 2005) can be viewed as an extension of the Fisher score (section 3.2.2), unifying *sup.* and *unsup.* contexts. The Laplacian score of feature $\boldsymbol{f_j}$ is given by :

$$S(\boldsymbol{f_j}) = \frac{1}{\sigma_{\boldsymbol{f_j}}} \sum_{i,k=1}^{n} (f_j(x_i) - f_j(x_k)) M_{i,k} \tag{3.21}$$

$S(\boldsymbol{f_j})$ can be rewritten using the Laplacian matrix, hence the name of the approach :

$$S(\boldsymbol{f_j}) = \frac{\widetilde{\boldsymbol{f_j}}^T \widetilde{L} \widetilde{\boldsymbol{f_j}}}{\widetilde{\boldsymbol{f_j}}^T \Delta \widetilde{\boldsymbol{f_j}}}$$

with $\mathbf{1}$ the n-dimensional constant vector $[1, ..., 1]^T$ and $\widetilde{\boldsymbol{f_j}} = \boldsymbol{f_j} - \frac{\boldsymbol{f_j}^T \Delta \mathbf{1}}{\mathbf{1}^T \Delta \mathbf{1}} \mathbf{1}$. The higher $S(\boldsymbol{f_j})$, the more important $\boldsymbol{f_j}$. The Laplacian score is also remotely related to the MaxVariance approach (Kantardzic, 2003), selecting features with large variance for the sake of their higher representative power.

In the *sup.* context, a possible similarity metric is the following :

$$\forall (i,k) \in [1, \dots, n]^2, \text{supsim}(\boldsymbol{x_i}, \boldsymbol{x_k}) = \begin{cases} 1 \text{ if } y_i = y_k \\ 0 \text{ if } y_i \neq y_k \end{cases} \tag{3.22}$$

Using the similarity metric from equation (3.22), Laplacian and Fisher scores are equivalent :

$$\forall j \in [1, \dots, D], \text{Laplacian}(\boldsymbol{f_j}) = \frac{1}{1 + \text{Fisher}(\boldsymbol{f_j})} \tag{3.23}$$

In the unsupervised context using $sim(\boldsymbol{x_i}, \boldsymbol{x_j}) = exp\{-\|\boldsymbol{x_i} - \boldsymbol{x_j}\|_2^2\}$, the Laplacian score becomes a semi-independent scoring method, as are all *unsup.* methods introduced in the remainder of this section.

SPEC *SPEC*[12] (Zhao and Liu, 2007) propose three scores respectively noted ϕ_1, ϕ_2 and ϕ_3, still following the idea that relevant features should be slowly varying among samples close to each other. After Shi and Malik (1997); Ng (2001), considering eigenvectors $\boldsymbol{\xi_0}, ..., \boldsymbol{\xi_{n-1}}$ of the normalized Laplacian $\widetilde{L}$ (respectively associated with eigenvalues $\lambda_0 < \lambda_1 < ... < \lambda_{n-1}$), smooth features are aligned with the first eigenvectors, hence the score ϕ_1 :

$$\forall j \in [1, \dots, D], \phi_1(\boldsymbol{f_j}) = \widehat{\boldsymbol{f_j}}^T \widetilde{L} \widehat{\boldsymbol{f_j}} \text{ where } \widehat{\boldsymbol{f_j}} = \Delta^{\frac{1}{2}} \boldsymbol{f_j} / \left\| \Delta^{\frac{1}{2}} \boldsymbol{f_j} \right\| \tag{3.24}$$

Eigenvectors $\boldsymbol{\xi_0}, ..., \boldsymbol{\xi_{n-1}}$ of $\widetilde{L}$ define soft cluster indicators, and eigenvalues $\lambda_0 < \lambda_1 < ... < \lambda_{n-1}$ measure the separability of the clusters. The smaller $\phi_1(\boldsymbol{f_j})$, the more efficient $\boldsymbol{f_j}$ is to separate the clusters.

As the first eigenvector $\boldsymbol{\xi_0} = \Delta^{\frac{1}{2}} 1$ does not carry any information, with $\lambda_0 = 0$, one might rather consider the projection of the feature vector $\boldsymbol{f_j}$ on the orthogonal space of $\boldsymbol{\xi_0}$:

$$\phi_2(\boldsymbol{f_j}) = \frac{1}{1 - \langle \widehat{\boldsymbol{f_j}}, \boldsymbol{\xi_0} \rangle} \widehat{\boldsymbol{f_j}}^T \widetilde{L} \widehat{\boldsymbol{f_j}} \tag{3.25}$$

Finally, in the case where the target number of clusters κ is known, only the top-κ eigenvectors are considered, and score ϕ_3 is defined as :

$$\phi_{3(}\boldsymbol{f_j}) = \sum_{k=1}^{\kappa} (2 - \lambda_k) \langle \widehat{\boldsymbol{f_j}}, \boldsymbol{\xi_k} \rangle^2 \tag{3.26}$$

12. This name is not an acronym, rather a reference to the reliance on spectral clustering theory

Features are ranked in ascending order for ϕ_1 and ϕ_2, and in descending order for ϕ_3.

The above scores measure the overall capacity of a feature to separate clusters, which might prove inefficient in multi-classes/multi clusters settings : a feature most efficient to separate a pair of clusters might have a mediocre general score.

MCFS Instead of assigning one global score per feature, *Multi-Cluster Feature Selection (MCFS)* (Cai et al., 2010) address the limitations of SPEC by computing one score per feature *per cluster*, essentially attempting to capture the local informativity of features. The respective capacities of the features to separate clusters are estimated through fitting the eigenvectors (reminding that $\boldsymbol{\xi_k}$ is a soft indicator of the k-th cluster) up to a regularization term. Formally, this corresponds to defining κ independent optimization problems (one per cluster) s.t. :

$$\forall k \in [1, \dots, \kappa], \boldsymbol{a_k^*} = \min_{\boldsymbol{a_k} \in \mathcal{R}^{D \times 1}} \|\boldsymbol{\xi_k} - X\boldsymbol{a_k}\|_2^2 + \beta_k \|\boldsymbol{a_k}\|_1 \tag{3.27}$$

$\boldsymbol{a_k^*} = (a_k^*(f_1), \dots, a_k^*(f_D))$ reflects the respective abilities of the original features to identify the k-th cluster. As seen in the earlier discussion on LASSO, the L_1 regularization term enforces the sparsity of $\boldsymbol{a_k}$ (penalization strength being governed by parameter β_k), retaining only the features most relevant to this cluster. The final score of $\boldsymbol{f_j}$ then simply corresponds to the maximum value of $a_k(f_j)$ over the κ clusters, s.t. :

$$S(\boldsymbol{f_j}) = \max_{k \in [1,\dots,\kappa]} |a_k(f_j)| \tag{3.28}$$

NDFS A general limitation common to Laplacian score, SPEC and MCFS is the reliance on an Euclidean distance-based similarity metric, which leads to unstable clustering in high-dimensional spaces due to the curse of dimensionality (section 2.2.1). *Non-negative Discriminative Feature Selection (NDFS)* (Li et al., 2012) alleviates this issue by defining a joint optimization problem. The κ learning goals $(\boldsymbol{a_1^*}, \dots, \boldsymbol{a_\kappa^*})$ from MCFS are merged into a single feature importance matrix $A \in \mathbb{R}^{D \times \kappa}$, subject to a group-LASSO regularization term. Moreover, considering that the original Laplacian eigenvectors $\boldsymbol{\xi_0}, \dots, \boldsymbol{\xi_{n-1}}$ are brittle, a second objective corresponds to a cluster indicator matrix $\Xi \in \mathbb{R}^{n \times \kappa}$. The rows of Ξ are initialized with the Laplacian eigenvectors. α and β denoting two regularization weights, the goal is then to find (Ξ^*, A^*) s.t. :

$$\Xi^*, A^* = \arg\min_{\Xi, A} Tr(\Xi^T \widetilde{L}\, \Xi) + \alpha(\|\Xi - XA\|_F^2 + \beta \|A\|_{2,1}^2) \tag{3.29}$$

with the additional constraint that Ξ must be orthogonal and semi-positive definite ($\Xi^T\Xi = I_\kappa, \Xi \geq 0$).

Following Yu and Shi (2003), the first term of equation (3.29) can be rewritten as [13] :

$$Tr(\Xi^T \widetilde{L} \Xi) = \frac{1}{2} \sum_{i,j=1}^{n} sim(\boldsymbol{x_i}, \boldsymbol{x_j}) \left\| \frac{\Xi(\boldsymbol{x_i})}{\sqrt{\Delta_i}} - \frac{\Xi(\boldsymbol{x_j})}{\sqrt{\Delta_j}} \right\|_2^2 \tag{3.30}$$

Using the alternate formulation from eq. (3.30), it is apparent that minimizing the term $Tr(\Xi^T \widetilde{L} \Xi)$ provides an incentive to cluster similar points together. The orthogonality and nonnegativity constraints on Ξ further

13. Slightly abusing the notation, $\Xi(\boldsymbol{x_i})$ designates the vector of cluster affilations of the i-th sample.

push each sample to belong in exactly one cluster. Lastly, the group-LASSO penalty on the rows of A enforces sparse feature selection.

NDFS is acknowledged as a seminal work in *unsup.* FS : later approaches (Li et al., 2014; Shi et al., 2014; Qian and Zhai, 2013; Nie et al., 2016) all stem from NDFS, and provide incremental performance improvements by adding (oftentimes computationally costly) third or even fourth optimization objectives. We will now introduce these methods for the sake of completeness.

CGSSL *Clustering-Guided Sparse Structural Learning (CGSSL)* (Li et al., 2014) iterates on NDFS by making the additional assumption that the pseudo-labels associated to the samples are actually generated by an underlying d-dimensional linear model, s.t. :

$$\forall(i,j) \in [1,\dots,\kappa] \times [1,\dots,n], \widetilde{y_i(x_j)} = \boldsymbol{v_i}^T \boldsymbol{x_j} + \boldsymbol{p_i}^T Q^T \boldsymbol{x_j} \tag{3.31}$$

where $\boldsymbol{v_i} \in \mathbb{R}^D$ and $\boldsymbol{p_i} \in \mathbb{R}^d$ are weight vectors and $Q \in \mathbb{R}^{D \times d}$ is an orthogonal matrix representing the linear transformation parameterizing the shared d-dimensional subspace. The feature importance matrix A from NDFS is then redefined to $A = V + QP$, and a regularization term on V is added to the optimization problem :

$$P^*, Q^*, \Xi^*, A^* = \arg\min_{P,Q,\Xi,A} Tr(\Xi^T \widetilde{L}\, \Xi) + \alpha(\|\Xi - XA\|_F^2 + \beta \|A\|_{2,1}^2) + \gamma \|A - QP\|_F^2 \tag{3.32}$$

still subject to Ξ orthogonal and semi-positive definite ($\Xi^T\Xi = I_\kappa, \Xi \geq 0$), with α, β, γ regularization weights. The motivation for this *sparse structural learning* (Ando and Zhang, 2005a) refinement is that the selected features arguably capture the structure of the underlying d-dimensional manifold (as per the manifold assumption from chapter 2).

RSFS *Robust Sparse Feature Selection (RSFS)* (Shi et al., 2014)) also aims at providing a more accurate cluster structure than NDFS, but does so in a different fashion than CGSSL, inspired by robust PCA (Candès et al., 2011). Instead of explicitly generating pseudo-labels, it is assumed that the learned cluster indicators may be arbitrarily corrupted, but that the corruption is sparse. This sparse noise is represented by a corruption matrix $C \in \mathbb{R}^{n \times \kappa}$ subject to a LASSO penalty. The optimization problem then becomes :

$$C^*, \Xi^*, A^* = \arg\min_{C,\Xi,A} Tr(\Xi^T \widetilde{L}\, \Xi) + \alpha(\|(\Xi - C) - XA\|_F^2 + \beta \|A\|_{2,1}^2) + \gamma |C|_1^2 \tag{3.33}$$

A parallel can be drawn between RSFS and denoising AutoEncoders (section 3.1.1) : a common idea is to increase the robustness of learning by exposition to small perturbating noise.

RUFS *Robust Unsupervised Feature Selection (RUFS)* (Qian and Zhai, 2013)) aims to improve the quality of the structure learned by NDFS by adding a cluster centroid matrix $C \in \mathbb{R}^{\kappa \times D}$ to the joint optimization problem :

$$C^*, \Xi^*, A^* = \arg\min_{C,\Xi,A} Tr(\Xi^T \widetilde{L}\, \Xi) + \alpha(\|\Xi - XA\|_{2,1}^2 + \beta \|A\|_{2,1}^2) + \gamma \|X - \Xi C\|_{2,1} \tag{3.34}$$

The authors claim that by virtue of this addition, the inaccuracies caused in the spectral clustering process by irrelevant features should mainly affect C, thus spare the more important Ξ.

SOGFS Even if clusters are dynamically updated like in the aforementioned NDFS variants, the similarity graph can still be arbitrarily corrupted by irrelevant or redundant features. *Structured Optimal Graph Feature Selection (SOGFS)* (Nie et al., 2016) aims to correct this flaw by also optimizing the similarity matrix M itself. The idea is to weigh the original features while computing pairwise similarities, according to the feature importance matrix A. The similarity metric is therefore progressively biased towards the best selection candidates.

Still following the *neighbors should look alike* idea, M^* is defined as :

$$M^* = \underset{M_i^T \mathbf{1}=1, 0 \leq m_{i,j} \leq 1}{\arg\min} \sum_{i,j} (\|\boldsymbol{x_i} - \boldsymbol{x_j}\|_2^2 \, m_{i,j} + \alpha m_{i,j}^2) \tag{3.35}$$

with α a regularization parameter to avoid the trivial solution. In order to consider only relevant features to learn the similarity matrix, this term becomes :

$$\min_{M_i^T \mathbf{1}=1, 0 \leq m_{i,j} \leq 1} \sum_{i,j} (\left\|A^T \boldsymbol{x_i} - A^T \boldsymbol{x_j}\right\|_2^2 \, m_{i,j} + \alpha m_{i,j}^2) \tag{3.36}$$

The final optimization problem of SOGFS is then :

$$M^*, \Xi^*, A^* = \underset{M,\Xi,A}{\arg\min} \, \gamma Tr(\Xi^T \widetilde{L} \, \Xi) + \beta \left\|A\right\|_{2,1}^2 + \sum_{i,j} \left(\left\|A^T \boldsymbol{x_i} - A^T \boldsymbol{x_j}\right\|_2^2 m_{i,j} + \alpha m_{i,j}^2 \right) \tag{3.37}$$

Discussion

While this review of state-of-the-art *unsup.* FS methods cannot be exhaustive [14], it supports the vision of *unsup.* FS as using spectral clustering to equip datapoints with pseudo-labels and fall back on *sup.* FS. This technique leads to two major issues :

— The already discussed reliance on a high-dimensional Euclidean distance to construct the similarity graph.

— The poor handling of redundant feature sets. Considering for instance that f_1 and f_2 are identical, then $S(f_1) \approx S(f_2)$ w.r.t. the Laplacian score, SPEC or MCFS. These twin features will consequently be either both rejected or selected, which is clearly sub-optimal no matter the cost function. Note that NDFS and its refinements address the redundancy issue.

As handling redundancy among initial features is of paramount importance, it is a cornerstone of our algorithmic contribution (chapter 4).

A third issue of state-of-the-art *unsup.* FS methods lie in their empirical validation pipeline rather than the selection itself. The efficiency of all aforementioned methods is assessed in a *sup.* environment, which goes against the *agnosticism* property of filters (more in chapter 5).

14. Additional references include for instance *TRACK* (Wang et al., 2014) and *Similarity Preserving Feature Selection (SPFS)* (Zhao et al., 2013).

Chapitre 4

Agnostic Feature Selection

This chapter presents our algorithmic contribution in the domain of *unsup.* FS, AGNOS, building upon the lessons learned from the state of the art (chapter 3). The proposed AGNOS presents an original learning criterion at the crossroad of FC and FS.

This novel combination of dimensionality reduction techniques is first discussed at a general level (*sec.* 4.1), introducing the main issues that need to be addressed (*sec.* 4.2). Section 4.3 thereafter discusses the feature scoring criteria. Lastly, three declinations of AGNOS, called AGNOS-W, AGNOS-G and AGNOS-S, are proposed in section 4.4.

4.1 AGNOS full picture

The proposed *unsup.* FS algorithm AGNOS combines two underlying motivations : data compression efficiency and generality, respectively defined and discussed in sections 4.1.1 and 4.1.2.

4.1.1 Efficient data compression (*fig.* 4.1)

State-of-the-art *unsup.* feature selection methods ((He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012; Nie et al., 2016), chapter 3) perform selection with the ultimate goal of predicting a specific target f^* not in the original feature set F, as in *sup.* learning (*chap.* 5). However, as underlined in chapter 1, *unsup.* learning constitutes the bulk of machine learning, and any element of the feature set can in principle define a learning goal (LeCun, 2016). Following this idea, AGNOS aims to select a feature subset supporting the prediction of *every* initial feature and thus tackles the *unsup.* FS problem from the angle of data compression.

Data compression is traditionally performed through FC (chapters 2 and 3). Accordingly, AGNOS is a two-step process : A compressed representation $\Phi_d \in \mathbb{R}^{n \times d}$ of the dataset is first obtained via feature construction. The original features are then ranked w.r.t. their importance for learning the d latent features $(\boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_d})$ composing Φ_d. A novelty of the approach thus is to bridge the gap between the two categories of DR techniques, using feature *construction* as a tool for feature *selection*. Interestingly enough, this essentially amounts to falling back on a *supervised* multi-labeled feature selection problem, where datapoints are assigned one continuous label per constructed feature $\boldsymbol{\phi_i}, i \in [1, \ldots, d]$.

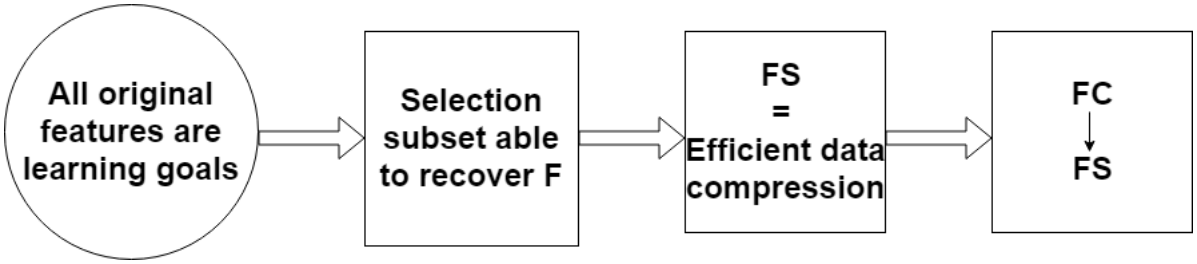

FIGURE 4.1: AGNOS combines FC and FS in order to *leave no feature behind*

4.1.2 Working hypotheses (*fig.* 4.2)

The main perk of *unsup.* learning is its increased adaptability compared to *sup.* learning (*chap.* 1) : *Unsup.* learning is applicable to *any* dataset, whether it pertains to Bioinformatics, insurance risk assessment or electrical engineering.

In order to be as general as possible, AGNOS involves minimal hypotheses. It follows the so-called Occam's razor principle (Blumer et al., 1987), formulated by Kearns and Vazirani (1994); Crowder and Carbone (2011) as *Occam learning*, specifically stipulating that the sought models involve as few contingencies as possible. The cornerstone of DR is the manifold assumption (*chap.* 2) :

The manifold assumption *The D-dimensional datapoints $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$ lie near a manifold $\mathcal{M}_{d^*}$ of dimension d^* s.t. $d^* << D$.*

In the following, the manifold assumption is the *only* assumption done in AGNOS.

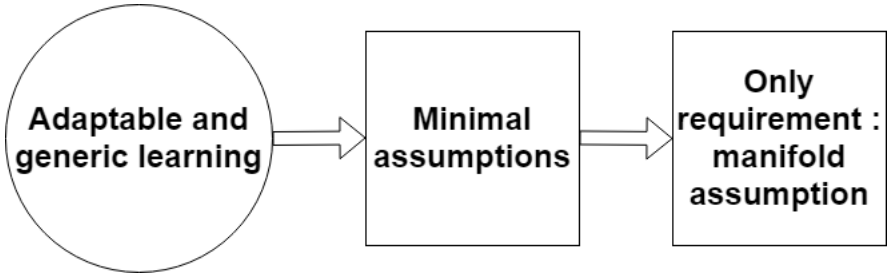

FIGURE 4.2: AGNOS operates under minimal hypotheses

4.1.3 Combining both motivations

Following section 4.1.1, AGNOS involves two interacting parts : a FC algorithm and a FS mechanism, and the question thus becomes to select the algorithms involved in each component. From the assumption

perspective and given the hybrid nature of AGNOS, it follows :

$$\text{Assumptions(AGNOS)} = \{ \text{ Manifold assumption } \} = \text{ Assumptions(FC)} \cup \text{ Assumptions(FS)}$$

where the first equality comes from *sec.* 4.1.2. Accordingly, both FC and FS components of AGNOS must not require any hypotheses beyond the manifold assumption. This requirement will rule out most DR methods (most often implicitly involving additional assumptions, e.g. PCA (Pearson, 1901) relies on the assumption that the underlying manifold be linear, or SFUS (Ma et al., 2012), assuming that all original features are boolean).

Eventually, the AGNOS DR pipeline proceeds by elimination and discards methods relying on additional hypotheses, as will now be discussed.

4.1.4 Typical DR requirements

DR methods typically make assumptions on three different components of the problem : the datapoints (*sec.* 4.1.4.1), the underlying manifold (*sec.* 4.1.4.2), or the original features (*sec.* 4.1.4.3).

4.1.4.1 Assumptions regarding the datapoints

Sup. DR algorithms (Ye and Sun, 2018; Zhao et al., 2006b), assuming that each datapoint x_i is equipped with a label y_i, are inappropriate by construction as we focus *unsup.* DR.

State-of-the-art *unsup.* FS methods (He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012; Nie et al., 2016) (*chap.* 3) rely on the assumption that the pairwise likeness of datapoints can be accurately depicted using the Euclidean distance in the original high-dimensional data representation. This assumption is unlikely to hold in the view of the curse of dimensionality (*chap.* 2). Spectral clustering-based FS is thus also out of contention.

4.1.4.2 Assumptions regarding the nature of $\mathcal{M}_d^*$

Linear FC techniques (Pearson, 1901; Golu and Reinsch, 1971) assume that the underlying manifold is linear, and are thus inappropriate in the AGNOS context [1].

Although approaches such as Isomap (Tenenbaum et al., 2000) and Locally Linear Embedding (Roweis and Saul, 2000) are able to unfold a non-linear Swiss Roll (*chap.* 3), their success hinges on *two* additional informal implicit assumptions : i) the underlying manifold $\mathcal{M}_d^*$ is *smooth* (informally, the parametric equations defining the manifold are infinitely differentiable) ; and ii) $\mathcal{M}_d^*$ does not contain "holes".

The fact that both assumptions cannot be efficiently (either computationally or statistically) be overcome is argued as follows. Consider the case of a torus (*fig.* 4.3), smooth manifold of intrinsic dimension 2 (w.r.t. any of the ID estimators from *chap.* 2). It is clear that the torus clearly cannot be flattened from 3D to 2D while preserving local neighborhoods due to the hole in the middle.

1. Note that even in the linear case, PCA and SVD require a specific configuration of the covariance matrix, s.t. the angles between its eigenvectors be sufficiently large (Martinez and Zhu, 2005).

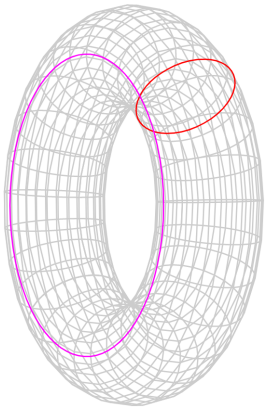

FIGURE 4.3: A torus corresponds to a smooth manifold, but contains a hole. From Wikipedia.

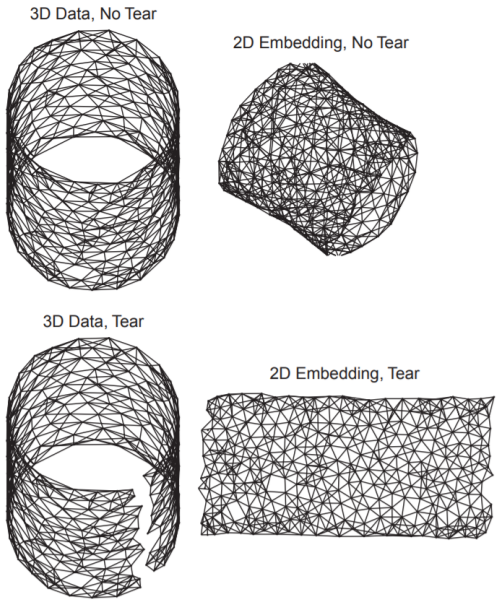

FIGURE 4.4: Like for the torus, Isomap fails to embed a 3D cylinder in a 2D space while preserving its similarity graph (upper panels). However, DR is succesful if the cylinder has been preemptively ”cut” (lower panels). From van der Maaten et al. (2008)

Indeed, an option would be to ”cut” the manifold as a form of pre-processing to DR (van der Maaten et al., 2008), so that it can thereafter be unfolded like a Swiss Roll (figure 4.4). However, such a pre-processing step involves n runs of the Dijkstra algorithm (Jianya, 1999), being thus computationally hardly affordable in most application domains. Furthermore, such a ”cutting” process effectively induces a deformation of the manifold, and would thus alter the final selection subset. Considering these two significant downsides,

similarity-preserving methods are ultimately discarded for performing FC in AGNOS.

4.1.4.3 Assumptions regarding the original features

As seen in chapter 3, independent scoring FS methods (Duda et al., 2000) assume that the original feature set F does not contain any XOR-like concept (*sec.* 3.2.1). State-of-the-art *unsup.* methods such as the Laplacian score (He et al., 2005), SPEC (Zhao and Liu, 2007) or MCFS (Cai et al., 2010) require that no elements of F are redundant in order to rank features fairly. Lastly, other approaches (Ma et al., 2012; Chang et al., 2014) assume that F contains only boolean features. Consequently, all aforementioned algorithms are not eligible to perform FS in AGNOS.

Discussion

Following the previous discussion, AGNOS should involve :

— A *non-linear unsup.* data compression scheme that does not rely on pairwise Euclidean distances between high-dimensional datapoints.

— A *semi-independent* or *collective* feature scoring criterion able to handle *redundancy*.

In view of these specifications, the natural candidate for performing FC is the *non-linear AutoEncoder* (*chap.* 3). Taking inspiration from neural network-based FS (*chap.* 3), a semi-independent feature scoring criterion is derived from the parameters of the AutoEncoder at the end of training. The following section discusses the feature redundancy problem in the context of an AutoEncoder.

4.2 The redundancy issue

In a nutshell, the general AGNOS scheme uses an AutoEncoder to produce a compression representation $\Phi_d \in \mathbb{R}^{n \times d}$ of the dataset ; the initial variables are thereafter ranked w.r.t. their importance for learning $(\boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_d})$. In the large sample limit, tuning the size of the encoder layer so that $d = ID(X)$ ensures that i) Φ_d contains all the information needed to reconstruct the original feature set ; and ii) each of $\boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_d}$ is informative to some extent.

Note however that how the information is organized and scattered among the constructed features is unknown, which might adversely affect the approach in the presence of redundant original features ; we shall come to this point in section 4.4.

Consider a basic AutoEncoder equipped only with a MSE loss :

$$L(F) = \sum_{i=1}^{D} ||\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f_i}||_2^2$$

The contributions of each feature to the total loss appear to be weighted equally. However, let us consider the case where the feature set contains κ duplicates of the first feature (for some $\kappa \in [2, \ldots, D-1]$, $\boldsymbol{f_1} = \boldsymbol{f_2} =$

$\ldots = \boldsymbol{f_\kappa}$. In an unsupervised feature selection setting, we would like the probability of selecting one copy of $\boldsymbol{f_1}$ to increase with κ. On the other hand, the probability of selecting multiple copies should clearly always be zero no matter the value of κ. However :

$$L(F) = \sum_{i=1}^{\kappa} ||\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f_1}||_2^2 + \sum_{i=\kappa+1}^{D} ||\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f_i}||_2^2$$

The contribution of $\boldsymbol{f_1}$ to the total loss is considered κ times more important than for another feature. The larger κ, the more reconstructing $\boldsymbol{f_1}$ is a priority for the AutoEncoder during training, the more influence $\boldsymbol{f_1}$ and all its duplicates ultimately hold over Φ_d. Given that AGNOS aims to score original features w.r.t. this influence, all copies of $\boldsymbol{f_1}$ will obtain the same score and be discarded or selected together.

This is a major issue, as it is very common for real world datasets to contain clusters of strongly correlated features. A sensible feature selection algorithm should select at most one representative per such cluster.

A key requirement for AGNOS to is to address the initial feature redundancy. To this end, three regularizations will be proposed (*sec.* 4.4) in the spirit of LASSO (Tibshirani, 1996) and group-LASSO (Yuan and Lin, 2007) to enforce the sparsity of the latent (*aka* constructed) features. Each regularization scheme comes with its own optimization criterion. The three criteria however rely on the same principles, disccused in section 4.3.

4.3 Feature scoring

As said, AGNOS ranks original features w.r.t. an importance score derived from the parameters of the trained AutoEncoder, reflecting their respective influence for learning Φ_d. Three scoring criteria will be examined in section 4.4, each corresponding to a declination of AGNOS.

The first criterion is based on the weights of the encoder part of the network, and is used in AGNOS-W. The second considers the gradients of the constructed features w.r.t. the input features and is relied upon by AGNOS-G. Lastly, the scoring criterion of AGNOS-S is based on an altered neural architecture.

The influence of $\boldsymbol{f_i}, i \in [1, \ldots, D]$ over Φ_d can be assessed independently for each of the constructed features $\boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_d}$. Section 4.3.1 discusses how these d influence measurements should be aggregated to obtain the final score for $\boldsymbol{f_i}$. Section 4.3.2 assesses the semi-independence nature of the resulting scoring criteria.

As AGNOS essentially amounts to falling back on a *sup.* multi-labeled FS problem, where datapoints are assigned one continuous label per constructed feature $\boldsymbol{\phi_i}, i \in [1, \ldots, d]$, one could wonder why a new scoring procedure is needed rather than simply adapting existing multi-labeled FS techniques (Ma et al., 2012; Chang et al., 2014). This interrogation is addressed by section 4.3.3.

4.3.1 From local influence to global feature score

As a result of $d = ID(X)$, in the large sample limit all constructed features are guaranteed to be relevant to the reconstruction of the initial features to some extent. $\forall (i, j) \in [1, \ldots, D] \times [1, \ldots, d]$, let $I(\boldsymbol{f_i}, \boldsymbol{\phi_j})$ denote

the influence of $\boldsymbol{f_i}$ over ϕ_j (to be formalized below). A simple definition for the ranking criterion is to consider the average influence of $\boldsymbol{f_i}$ over all constructed features :

$$\text{Score}(\boldsymbol{f_i}) = \frac{1}{d}\sum_{j=1}^{d} I(\boldsymbol{f_i}, \boldsymbol{\phi_j}) \tag{4.1}$$

However, this formulation fails to take into account the local informativity of the features. The proposed approach takes inspiration from the unsupervised feature selection algorithm *MCFS* (Cai et al., 2010) (*chap.* 3), in which a feature is considered important *iff* it is helpful to identify *at least one* cluster. Assume that for a certain $\boldsymbol{f_i}$, $\exists j \in [1, \ldots, d]$ s.t. $I(\boldsymbol{f_i}, \boldsymbol{\phi_j})$ is large and $\forall k \in [1, \ldots, d]$ s.t. $k \neq j$, $I(\boldsymbol{f_i}, \boldsymbol{\phi_k}) = 0$. Then, according to equation (4.1), Score$(\boldsymbol{f_i})$ is small, and $\boldsymbol{f_i}$ is likely to be discarded. However, if a feature has a strong influence on at least one constructed feature, it means that preserving the information it contains is very important for data compression. Therefore, such a feature $\boldsymbol{f_i}$ should be ranked highly and prioritized for selection.

Accordingly, a more suitable ranking criterion definition is to consider the maximum influence of $\boldsymbol{f_i}$ over *any* constructed feature :

$$\text{Score}(\boldsymbol{f_i}) = \max_{j \in [1,\ldots,d]} I(\boldsymbol{f_i}, \boldsymbol{\phi_j}) \tag{4.2}$$

The criteria used in AGNOS-W and AGNOS-G both follow this reasoning.

Remark. Taking the maximum value might however be inappropriate if some latent variables are significantly less important than others. If an initial feature $\boldsymbol{f_i}$ holds a strong influence over only one ϕ_j of lesser importance, then $\boldsymbol{f_i}$ should be discarded. Nevertheless, $\boldsymbol{f_i}$ will inaccurately be ranked highly w.r.t. *eq.* (4.2).

Considering this possible imbalance in constructed feature importance, a solution is to alter the structure of the AutoEncoder, so that instead of aggregating d measurements $I(\boldsymbol{f_i}, \boldsymbol{\phi_j})$, the overall influence of $\boldsymbol{f_i}$ over Φ_d is directly observed :

$$\text{Score}(\boldsymbol{f_i}) = I(\boldsymbol{f_i}, \Phi_d) \tag{4.3}$$

This change of scoring paradigm is the basis of AGNOS-S (*sec.* 4.4)

4.3.2 Is AGNOS an independent scoring method ?

As seen in chapter 2, unlike *collective* or *semi-independent* scoring methods, *independent* feature selection algorithms bear the important limitation of being unable to recognize features that are useless by themselves, but important together.

In AGNOS, although Score$(\boldsymbol{f_i})$ is ultimately assessed in isolation from the other scores, it is derived from the influence measurements $I(\boldsymbol{f_i}, \boldsymbol{\phi_j})$ (or $I(\boldsymbol{f_i}, \Phi_d)$ in the case of AGNOS-S). Given that each constructed feature is obtained by a non-linear combination of every initial variable, $I(\boldsymbol{f_i}, \boldsymbol{\phi_j})$ actually indirectly involves the whole original feature set F. AGNOS is therefore a semi-independent scoring method, and meets the requirement of being applicable to datasets containing XOR-like concepts.

4.3.3 Supervised multi-labeled feature selection via shared subspace learning

Sup. multi-labeled FS has been tackled by several approaches (Ma et al., 2012; Chang et al., 2014) in the context of image annotation. In this setting, each image is labeled with multiple concepts it is related to such as "people", "party", "entertainement".

Sup. multi-labeled FS methods then make the assumption (Ando and Zhang, 2005b) that images are likely to share *some* labels with each other (e.g. ("people","work") and ("people","party") pertain to the same topic "people"). The goal is then to learn a *shared subspace* for the original features to help predict the labels.

Formally, let $Y = [\boldsymbol{y_1}, \ldots, \boldsymbol{y_d}] \in \{0,1\}^{d \times n}$ denote the label matrix, $V \in \mathbb{R}^{D \times d}$ and $P \in \mathbb{R}^{D \times d}$ two weight matrices, and $Q \in \mathbb{R}^{D \times D}$ the shared subspace matrix. The goal is then to find V^*, P^*, Q^* s.t. :

$$(V^*, P^*, Q^*) = \underset{V,P,Q}{\arg\min}\, \text{loss}\big((V + QP)^T X^T, Y\big) + \mu\Omega(V, P) \text{ with } Q^T Q = I_D \tag{4.4}$$

where $\Omega(V, P)$ is a regularization term weighted by μ. The original features selected are then those corresponding to non-zero rows of $V^* + Q^* P^*$.

Arguably, one could adapt this framework to our unsupervised context by replacing Y with Φ_d^T. However, such an adaptation raises several theoretical and practical objections.

Firstly, the process of shared subspace learning basically amounts to optimizing a function mapping the samples to the labels. However, a specificity of our approach is that we already have access to this exact function : it is given by the encoding part of the AutoEncoder. Therefore, rather than learning a new mapping from scratch, we aim to leverage the existing one, by observing the parameters of the AutoEncoder.

Secondly, $\boldsymbol{\phi_i} \in \mathbb{R}^n$ is not a binary feature, but a continuous one. Therefore, the underlying assumption for subspace learning that samples share some labels is much less likely to hold.

Finally, a requirement for feature selection to be applicable in practice is to be less computationally expensive than learning without dimensionality reduction. Training an AutoEncoder to learn ϕ_d, then solving the optimization problem in equation (4.4) is hardly affordable in terms of time complexity.

4.4 The AGNOS algorithm

We address the issue of redundancy among the original features by adapting LASSO-inspired regularization techniques to the unsupervised context. The AutoEncoder loss function is enhanced with a penalty term enforcing that only few, non-redundant initial features are retained during learning. This section presents the three considered regularization schemes, each corresponding to a declination of AGNOS : weight-based regularization for AGNOS-W (section 4.4.1), gradient-based regularization for AGNOS-G (section 4.4.2), and *slack variable*-based regularization for AGNOS-S (section 4.4.3). The three versions of AGNOS are then discussed in section 4.4.3.

Preprocessing : normalizing the original features

Intuitively, the larger the contribution of $\boldsymbol{f_i}$ to the AutoEncoder loss $L(F)$, the more influent $\boldsymbol{f_i}$, the more likely $\boldsymbol{f_i}$ is ultimately selected.

Assume however that for some pair of initial features $\boldsymbol{f_i}$ and $\boldsymbol{f_j}$, one has $\boldsymbol{f_j} = C * \boldsymbol{f_i}$ for some constant $C > 1$. Clearly both features carry the same information and their respective influences on Φ_d should be equal. However, like in the case of redundant features and for the same reason, the contribution of $\boldsymbol{f_j}$ to the MSE-based $L(F)$ tends to be C times larger than for $\boldsymbol{f_i}$, and $\boldsymbol{f_j}$ is prioritized over $\boldsymbol{f_i}$ for selection.

This selection bias towards features with large first and second order moments is handled by pre-processing the dataset, each initial feature being normalized and centered.

4.4.1 AGNOS with weight regularization : AGNOS-W

The first declination of AGNOS is AGNOS-W, which is inspired by supervised feature selection with neural networks techniques relying on weight-based regularization (Bauer et al., 2000; Roy et al., 2015). The AutoEncoder loss function is enhanced with a group-LASSO (Yuan and Lin, 2007) penalty term on the weights of the hidden layer. Formally, letting $W \in \mathbb{R}^{D \times d}$ denote the encoder weight matrix and $W_{i,*}$ its i-th row, the $L_{2,1}$ penalization reads :

$$L(W) = \sum_{i=1}^{D} \sqrt{\sum_{k=1}^{d} W_{i,k}^2} = \sum_{i=1}^{D} \|W_{i,*}\|_2$$

and the learning criterion of AGNOS-W is accordingly defined as :

$$L_W(F) = \sum_{i=1}^{D} ||\hat{\boldsymbol{f_i}} - \boldsymbol{f_i}||_2^2 + \lambda L(W) \tag{4.5}$$

with λ the penalization weight. This regularization leads to a *sparse input* neural network (Feng and Simon, 2017) (*chap.* 3), enforcing that only a few original features are influential for learning Φ_d. In order to simultaneously reconstruct the whole feature set and rely on as few original features as possible, the AutoEncoder is coerced to discard redundant features by setting the corresponding rows of W to 0. This learning criterion thus is meant to tackle the issue of redundant feature sets.

After training, the influence of $\boldsymbol{f_i}$ over $\boldsymbol{\phi_j}$ can be observed through coefficient $|W_{i,j}|$, as in RFE-SVM (Guyon et al., 2002). The larger this quantity, the more important $\boldsymbol{f_i}$. Naturally, considering the absolute value of $W_{i,j}$ is necessary to properly account for negative weights, which are as informative as positive ones.

As previously discussed, $\boldsymbol{f_i}$ should be considered important *iff* it is influential on at least one constructed feature. Therefore, the final score of the i-th feature is defined as the maximum influence on any hidden neuron :

$$Score_W(\boldsymbol{f_i}) = \|W_{i,*}\|_\infty \tag{4.6}$$

This leads to the following algorithm :

Algorithm 5 AGNOS-W

Input : Feature set $F = \{\boldsymbol{f_1}, ..., \boldsymbol{f_D}\}$
Parameter: λ
Output : Ranking of features in F
Normalize each feature to zero mean and unit variance.
Estimate intrinsic dimension $\widehat{ID}$ of F.
Initialize neural network with $d = \widehat{ID}$ neurons in the hidden layer.
Repeat
Backpropagate $L_W(F) = \sum_{i=1}^{D} \|\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f_i}\|_2^2 + \lambda \sum_{i=1}^{D} \|W_{i,\cdot}\|_2$
until convergence
Rank features by decreasing scores with $Score_W(\boldsymbol{f_i}) = \|W_{i,\cdot}\|_\infty$.

4.4.2 AGNOS with gradient regularization : AGNOS-G

The second proposed declination of AGNOS is AGNOS-G, inspired by studies on the benefits of gradient regularization (Rifai et al., 2011; Varga et al., 2017).

In the supervised context,Varga et al. (2017) have recently shown that L_2 regularization on the gradients of the output layer helps improve the predictive accuracy of discriminative neural networks. This improvement is explained by the fact that smaller output gradients leads to a decreased sensitivity of the learning goal w.r.t. the input, which helps combat overfitting.

In the unsupervised context, Rifai et al. (2011) introduced *contractive AutoEncoders*. This corresponds to enhancing a standard AutoEncoder with a L_2 penalty term on the gradients of the *hidden* layer w.r.t. the input dimensions. The compressed representation Φ_d produced by contractive AutoEncoders has been empirically shown to be more robust w.r.t. input noise than for traditional AutoEncoders. By contrast with the supervised setting, the output layer of an AutoEncoder is not the end goal of learning, rather a byproduct of feature construction. The actual learning goal is Φ_d, which resides in the hidden layer. This reasoning motivates penalizing the hidden layer gradients rather than the output ones.

AGNOS-G also relies on regularizing the gradients of the hidden layer. Given that the end goal of AGNOS is feature selection, this regularization should aim to simultaneously cancel *all* gradients of the constructed features w.r.t. redundant original features. The L_2 regularization used for contractive AutoEncoders is inadequate for that purpose ; as seen in chapter 3, an L_2 penalty cannot enforce sparsity.

Therefore, AGNOS-G instead employs a group-LASSO (Yuan and Lin, 2007) regularization instead, regrouping hidden layer gradients by original feature :

$$L(Z_d) = \sum_{i=1}^{D} \sqrt{\sum_{k=1}^{n} \sum_{j=1}^{d} \left(\frac{\partial \boldsymbol{z_j}}{\partial \boldsymbol{f_i}}(x_k)\right)^2} \tag{4.7}$$

The total loss function of the AutoEncoder is then :

$$L_G(F) = \sum_{i=1}^{D} ||\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f}_{\boldsymbol{i}}||_2^2 + \lambda L(Z_d) \tag{4.8}$$

Similarly as in AGNOS-W, the combination of the mean square error and the sparsity pressure incentivizes the AutoEncoder to nullify all hidden layer gradients related to superfluous initial features, therefore successfully tackling the redundancy issue.

After training, the influence of $\boldsymbol{f}_{\boldsymbol{i}}$ over $\boldsymbol{\phi}_{\boldsymbol{j}}$ can be observed through the gradient of $\boldsymbol{\phi}_{\boldsymbol{j}}$ w.r.t. $\boldsymbol{f}_{\boldsymbol{i}}$, estimated at each datapoint :

$$I(\boldsymbol{f}_{\boldsymbol{i}}, \boldsymbol{z}_{\boldsymbol{j}}) = \sum_{k=1}^{n} \left(\frac{\partial \boldsymbol{\phi}_{\boldsymbol{j}}}{\partial \boldsymbol{f}_{\boldsymbol{i}}}(x_k) \right)^2 \tag{4.9}$$

With the same reasoning as for AGNOS-W, $\boldsymbol{f}_{\boldsymbol{i}}$ should be considered important *iff* it is influential for at least one constructed feature. Therefore, the final score is defined as :

$$\text{Score}_G(\boldsymbol{f}_{\boldsymbol{i}}) = \max_{1 \leq j \leq d} \sum_{k=1}^{n} \left(\frac{\partial \boldsymbol{\phi}_{\boldsymbol{j}}}{\partial \boldsymbol{f}_{\boldsymbol{i}}}(x_k) \right)^2 \tag{4.10}$$

The larger this quantity, the more important $\boldsymbol{f}_{\boldsymbol{i}}$. This scoring criterion is similar to that of supervised *feature saliency* selection methods (Steppe and Bauer, 1996; Zurada et al., 1997), with the notable difference that we consider the hidden layer gradients rather than the output ones. This is consistent with the observation that our learning goal is not the output of the network, rather Φ_d.

Interestingly enough, this modification also holds two practical advantages over traditional feature saliency techniques. On one hand, examining the gradients halfway through the feedforward process helps reduce the probability of encountering a vanishing gradient problem (Pascanu et al., 2013). On the other hand, obtaining the pointwise gradients of the hidden layer w.r.t. the input dimensions requires $n \times d \times D$ computations for each training iteration, as opposed to $n \times D^2$ computations for the gradients of the output layer. Given that $d << D$, this is a significant reduction in time complexity.

Moreover, in the case of an encoder with a single hidden layer, this score can be computed in a simple fashion. For example, with a $tanh$ activation function, one has :

$$\text{Score}_G(\boldsymbol{f}_{\boldsymbol{i}}) = \max_{1 \leq j \leq d} \sum_{k=1}^{n} W_{i,j}^2 \left(1 - \boldsymbol{\phi}_{\boldsymbol{j}}(x_k)^2\right)^2 \tag{4.11}$$

This leads to the proposed algorithm :

Algorithm 6 AGNOS-G

Input : Feature set $F = \{\boldsymbol{f_1}, ..., \boldsymbol{f_D}\}$
Parameter: λ
Output : Ranking of features in F
Normalize each feature to zero mean and unit variance.
Estimate intrinsic dimension $\widehat{ID}$ of F.
Initialize neural network with $d = \widehat{ID}$ neurons in the hidden layer.
Repeat
Backpropagate $L_G(F) = \sum_{i=1}^{D} ||\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f_i}||_2^2 + \lambda \sum_{i=1}^{D} \sqrt{\sum_{k=1}^{n} \sum_{j=1}^{d} \left(\frac{\partial \phi_j}{\partial \boldsymbol{f_i}}(x_k)\right)^2}$
until convergence
Rank features by decreasing scores with $Score_G(\boldsymbol{f_i}) = \max_{j \in [1,...,d]} \sum_{k=1}^{n} |\frac{\partial \phi_j}{\partial \boldsymbol{f_i}}(x_k)|$.

4.4.3 AGNOS with slack variables : AGNOS-S

A third version of AGNOS is considered, called AGNOS-S and inspired from Leray and Gallinari (1999); Li et al. (2016); Goudet et al. (2018). The neural architecture is augmented with a sparse one-to-one linear layer composed of *slack variables*, inserted between the input and the first hidden layer. Formally, to each feature $\boldsymbol{f_i}$ is associated a (learned) coefficient a_i initialized to 1, and the encoder is fed with the vector $(a_i \boldsymbol{f_i})$ (fig. 4.5). The learning criterion here is the reconstruction loss augmented with an L_1 penalization on the slack variables :

$$L_S(F) = \sum_{i=1}^{D} ||\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f_i}||_2^2 + \lambda \sum_{i=1}^{D} |a_i| \tag{4.12}$$

Like in LASSO (Tibshirani, 1996), the L_1 penalization pushes the slack variables toward a sparse vector such that features unnecessary to reconstruct F are associated a null coefficient. This regularization thus efficiently tackles the issue of redundancy. In order to prevent the network from drawing slack variables toward 0 and compensating for the small slack variables by proportionally amplifying the encoder weights, the encoder weight vector W is normalized ($\|W\|_2 = 1$). In order to obtain a standardized protocol, this normalization is also applied in the AGNOS-W and AGNOS-G variants.

Similarly as in Li et al. (2016), the score of the i-th feature is eventually set to $|a_i|$: this single real-valued coefficient reflects the contribution of $\boldsymbol{f_i}$ to the latent representation, and its importance to reconstruct the whole feature set :

$$Score_S(\boldsymbol{f_i}) = |a_i| \tag{4.13}$$

The larger this quantity, the more important $\boldsymbol{f_i}$.

This corresponds to directly measuring $I(\boldsymbol{f_i}, Z_d)$ rather than $I(\boldsymbol{f_i}, \boldsymbol{z_j})$ for each $\boldsymbol{z_j}$. Therefore, a major benefit of this altered neural architecture is to provide an accurate ranking criterion even if some constructed features are more important than others.

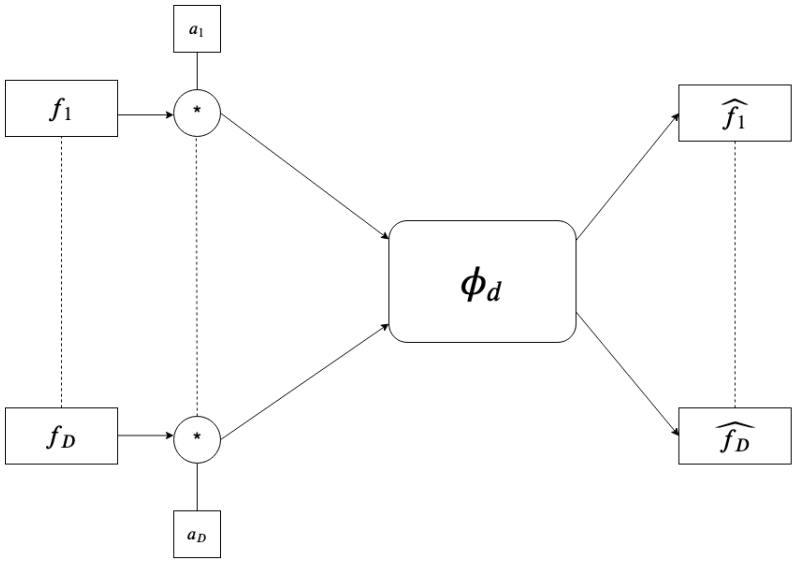

FIGURE 4.5: Structure of the neural network used in AGNOS-S

Algorithm 7 AGNOS-S

Input : Feature set $F = \{\boldsymbol{f_1}, ..., \boldsymbol{f_D}\}$
Parameter: λ
Output : Ranking of features in F
Normalize each feature to zero mean and unit variance.
Estimate intrinsic dimension $\widehat{ID}$ of F.
Initialize neural network with $(a_1, ..., a_D) = \mathbf{1_D}$ and $d = \widehat{ID}$ neurons in the hidden layer.
Repeat
Backpropagate $L_S(F) = \sum_{i=1}^{D} ||\hat{\boldsymbol{f}}_{\boldsymbol{i}} - \boldsymbol{f_i}||_2^2 + \lambda \sum_{i=1}^{D} |a_i|$
until convergence
Rank features by decreasing scores with $Score_S(\boldsymbol{f_i}) = |a_i|$.

Discussion

About tied weights While designing the architecture of an AutoEncoder, it is common practice to rely on *weight sharing* (Baldi, 2012; Bengio, 2012). This practice, also referred to as *tied weights*, consists in setting the weights of the decoder part of the network to the transpose of the encoder weights (that is, in the case of a single hidden layer, $W^{(decoder)} = W^T$). An immediate benefit lies in the reduced number of learned parameters, which both lowers the space complexity of training and helps prevent overfitting. Without weight sharing, the AutoEncoder might learn very small encoder weights (corresponding to a near linear regime in the activation functions) and compensate with large decoder weights. This essentially amounts to learning

the identity function. An additional benefit of weight sharing is therefore its potential regularizing effect, which prevents learning this degenerate solution.

However, weight sharing may be detrimental to learning when the end goal is not reconstruction but feature selection. Assume the AutoEncoder is able to reconstruct the whole feature set and that no initial feature is constant. By construction :

$$\forall j \in [1, \dots, D], \exists i \in [1, \dots, d], |W_{ij}^{(decoder)}| > 0$$

With weight sharing, $W_{ij}^{(decoder)} = W_{ji}$. Therefore :

$$\forall j \in [1, \dots, D], \exists i \in [1, \dots, d], |W_{ji}| > 0$$

In other words, there is at least one non-zero coefficient per row of W. This means that every initial feature is considered at least somewhat important w.r.t. the scoring function of AGNOS-W, hindering feature selection. This is precisely what the $L_{2,1}$ weight regularization employed in AGNOS-W aims to avoid, justifying leaving encoder and decoder weights untied.

Although the negative impact of weight sharing on AGNOS-G and AGNOS-S is less clear from a theoretical standpoint, preliminary experiments have shown that relying on tied weights led to decreased FS performance in both these versions of AGNOS.

Should the group LASSO penalty be sparse ? Recent supervised neural network-based FS methods (Roy et al., 2015; Li et al., 2016; Ye and Sun, 2018) (*chap.* 3) perform regularization via a sparse group LASSO penalty (Simon et al., 2013) on the weights of the first hidden layer.

By contrast, both AGNOS-W and AGNOS-G rely on a "vanilla" non-sparse group LASSO (Yuan and Lin, 2007) penalty instead. A first argument for this design choice is to avoid introducing the additional hyperparameter α, which would increase the complexity of the sensitivity study (more in *chap.* 6).

More importantly, the goal of the sparse group LASSO is to achieve sparsity both at the group level and inside each group, which is a desirable property in the supervised setting (Feng and Simon, 2017). However, in the context of an AutoEncoder performing unsupervised learning, a sparse group LASSO penalty would induce a compressed representation s.t. each original feature contributes to as few constructed features as possible. This corresponds to learning a *disentangled representation* (Bengio et al., 2013; Kim and Mnih, 2018). Learning a disentangled representation essentially corresponds to sacrificing some efficiency regarding data compression [2] in exchange for better model interpretability [3]. However, studying the impact of such a tradeoff is beyond the scope of this thesis.

Normalizing structural regularization strength λ The sparsity penalty terms implemented in AGNOS-S, AGNOS-W and AGNOS-G respectively involve D slack variables, $D \times d$ weights, and $D \times d \times n$ gradients.

2. This is a direct consequence of the network being prevented from spreadng the information contained by an important original feature into multiple components of the d-dimensional space.

3. Arguably, the fewer original features are involved for learning each constructed feature, the easier it is to measure their respective influence.

The number of parameters involved in the structural regularization thus depends on the considered variant of the approach. Accordingly, for the sake of consistency and homogeneity, the respective penalty strengths $(\lambda_S, \lambda_W, \lambda_G)$ are normalized s.t. $\lambda_W = \frac{\lambda_S}{d}$ and $\lambda_G = \frac{\lambda_S}{dn}$.

Chapitre 5

Performance indicators for assessing unsupervised Feature Selection

This chapter is concerned with the validation procedure of *unsup.* FS. The theoretical properties of an ideal performance indicator are first discussed in section 5.1. Section 5.2 thereafter introduces the three *sup.* criteria typically used in *unsup.* FS. Lastly, section 5.3 presents our methodological contribution, the *unsup.* FIT criterion.

How many features to select ? The results of both *sup.* and *unsup.* assessment protocols depend on hyperparameter k governing the size of the selection subset S_k. As discussed in chapter 2, k is in the *unsup.* FS context manually set by the user rather than automatically tuned like in *sup.* approaches (Ye and Sun, 2018). In the remainder of this chapter, k is considered a fixed parameter.

Empirical validation of state-of-the-art *unsup.* FS algorithms (He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012; Nie et al., 2016) typically considers multiple values for k, in order to monitor the efficiency of selection as a function of the reduced dimensionality. Accordingly, we will follow this protocol in our own empirical study (*chap.* 6).

5.1 Motivation

In *sup.* FS, the relevance of a performance indicator PI is unambiguous : PI is a good criterion *iff* it rewards selecting the best features for predicting the learning goal.

By contrast, the quality of PI is harder to define in the *unsup.* context, given the absence of ground truth. An important question then arises : “What makes a *good* performance indicator for *unsup.* FS ?”

We argue that a suitable PI should strive for the following six qualities, defined thereafter :

— Impartiality

— Expressivity

— Stability

— Interpretability

— Simplicity
— Cost-efficiency

Impartiality Impartiality denotes the absence of unwanted bias towards selecting features of a certain nature, structure or apparent purpose. Consider for instance PI_1 s.t. :

$$PI_1(S_k) = \begin{cases} 1 \text{ if } S_k \text{ contains only categorical features} \\ 0 \text{ otherwise} \end{cases}$$

PI_1 incentivizes the selection of categorical features regardless of information carried, thus fails to meet the requirement of impartiality.

Expressivity The more information is taken into account by PI, the more expressive this performance indicator. Consider e.g. the following PI_2 :

$$PI_2(S_k) = \max_{\boldsymbol{f} \in S_k} cov(\boldsymbol{f}, \boldsymbol{f_1})$$

The quality of S_k is assessed using only a small piece of information (the covariance with a single particular feature) ; the expressiveness of PI_2 is hence clearly low.

Stability The score of a particular *unsup.* FS algorithm w.r.t. PI is expected to fluctuate according to the selection subset size k and the set Θ of algorithm hyperparameters. However, the *position* of this method in the ranking of *unsup.* FS approaches should hardly depend on k and Θ.

Consequently, PI is deemed stable *iff* the associated ordering of *unsup.* selection techniques is consistent across a wide range of values for k and Θ. In other words, what matters is the *relative* hierarchy of algorithms rather than the respective absolute scores.

Consider e.g. PI_3 s.t. :

$$PI_3(S_k) = \begin{cases} PI_1(S_k) \text{ if } k \leq 10 \\ PI_2(S_k) \text{ if } k > 10 \end{cases}$$

The ordering of selection methods is likely shuffled when considering subsets of more than ten features. PI_3 is therefore unstable [1].

Interpretability In order to comply with FTA learning (*chap.* 1), a performance indicator should be easily understandable, even by a non-expert in ML. The behavior of information-theoretic measurements such as the *Variation of Information (VI)* (Meilă, 2003) is unintuitive, as claimed by Gates et al. (2018) (more in *sec.* 5.2). Such indicators thus arguably lack in interpretability.

1. Arguably, PI_3 is also *non-smooth* : the score of an algorithm abruptly changes across the boundary $k = 10$. Smoothness is also a desired property of performance indicators, although with a lower priority than stability.

Simplicity Assume FS algorithm A is ranked highly w.r.t. performance indicator PI. This could indicate that A is a better selection method than its competitors. However, it may also be that PI is *partial* towards the selection subset resulting of A. In the absence of ground truth, disentangling the experimental validation of *unsup.* FS methods from the empirical study of PI itself is therefore challenging. This issue is amplified in the presence of important hyperparameters for PI. Consider for instance PI_4 s.t. :

$$\forall(\alpha,\beta,\gamma) \in \mathbb{R}^3, PI_4(S_k,\alpha,\beta,\gamma) = \alpha PI_1(S_k) + \beta PI_2(S_k) + \gamma PI_3(S_k)$$

The goal of measuring the quality of S_k with PI_4 is mingled with the task of tuning α, β and γ. A *simple* performance indicator should therefore, unlike PI_4, include as few hyperparameters as possible, so that it can be considered a fixed component of the validation process.

Cost-efficiency In addition to all aforementioned qualities, an ideal performance indicator should also be as inexpensive as possible in terms of computational complexity (both time and space-wise). Most notably and in view of Big Data, the assessment procedure should scale well w.r.t. both the number of datapoints and the number of original features.

Discussion

As said, state-of-the-art *unsup.* FS methods (He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012; Nie et al., 2016) are typically empirically assessed with a *sup.* performance indicator. We claim that this procedure nets significant downsides for little benefits :

Claim: Typical *sup.* performance indicators for *unsup.* FS all fail to meet the *impartiality*, *expressivity* and *stability* properties.

This claim will be further discussed in section 5.2 and supported by empirical evidence in our experimental study (*chap.* 6). Our campaign of experiments will also show that, relatively to *sup.* performance indicators, the proposed *unsup.* FIT criterion is more impartial, expressive and stable. The main downside of this methodological contribution lies in its poor cost-efficiency (*chap.* 6). A direction for future research consists in improving the cost-efficiency of the FIT procedure (*chap.* 7).

5.2 Supervised performance indicators

State-of-the-art *unsup.* FS methods (He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012) typically rely on one of two techniques to quantify the quality of the retained feature subset S_k. The first technique is classifier-based (*sec.* 5.2.1), while the second one is clustering-based (*sec.* 5.2.2). Section 5.2.3 will thereafter investigate our claim that *sup.* performance indicators are ill-suited for assessing *unsup.* FS.

5.2.1 Classifier-based criterion

Given that we are here concerned with *sup.* assessment of *unsup.* FS, let $\boldsymbol{f}^*$ denote the target feature and χ a classifier. Let the datapoints be split into a training set X_{train} and a testing set X_{test}. Let $\delta : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ denote the function s.t. :

$$\forall (a, b) \in \mathbb{N}^2, \delta(a, b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ otherwise.} \end{cases}$$

The predictive power of the selection subset S_k is measured using h. h is first trained on X_{train}, **considering only the features in** S_k. The test classification error rate of ϕ is thereafter defined as :

$$ER_\phi(S_k, \boldsymbol{f}^*) = 1 - \sum_{x \in X_{test}} \delta\left(h(x), f^*(x)\right) = \frac{1}{|X_{test}|} \sum_{x \in X_{test}} \mathcal{L}\left(f^*(x), \widehat{f}(x)\right) \tag{5.1}$$

The lower $ER_\phi(S_k, \boldsymbol{f}^*)$, the more accurate χ for predicting $\boldsymbol{f}^*$ using only S_k. *Unsup.* FS algorithms are then ranked in ascending order w.r.t. the respective resulting error rates.

The resulting ranking is actually a multivariate function involving $\boldsymbol{f}^*$, the classifier structure (e.g. Decision Tree (Safavian and Landgrebe, 1991), Random Forest (Díaz-Uriarte and Andres, 2006), Gaussian SVM (Scholkopf and Smola, 2001)), the hyperparameters of h, and the random split between X_{train} and X_{test}. The dependency w.r.t. the train/test split is typically removed by relying on Leave-One-Out (LOO) cross-validation (Kohavi, 1995)[2].

The most common choice for g in the literature (Zhao and Liu, 2007; Cai et al., 2010) is the p-*Nearest-Neighbor (p-NN)* classifier. The motivation underlying this choice is fourfold.

First of all, p-NN classification is non-linear, which is required due to working hypotheses (*sec.* 4.1.2). Furthermore, the unreliability of Euclidean distance-based similarities caused by the curse of dimensionality is alleviated in the k-dimensional space defined by S_k (provided that $k << D$).

Moreover, p-NN classification is usually more resilient w.r.t. overfitting than e.g. Decision-Tree based classification such as Random Forest. Lastly, a p-NN classifier does not involve any hyperparameters besides p, typically fixed to $p = 1$ (thus predicting the class of the sample closest to the considered datapoint). This classifier therefore adheres to the notion of *simplicity* introduced earlier.

With $p = 1$ and x'_i denoting the nearest neighbor of x_i [3], the error rate from equation (5.1) can then be rewritten as (Cai et al., 2010) :

$$ER(S_k, \boldsymbol{f}^*) = 1 - \frac{1}{n} \sum_{i=1}^{n} \delta\big(f^*(x_i), f^*(x'_i)\big) \tag{5.2}$$

The empirical assessment of the three declinations of AGNOS and baseline unsupervised FS algorithms w.r.t. this performance indicator will be provided in our experimental validation (*chap.* 6) with $p = 5$ neighbors, for the sake of stability.

2. Arguably, the results still weakly depend on the order in which the n-folds are considered for training g.
3. Formally :

$$\forall (i, j) \in [1, \ldots, n]^2 \text{ s.t. } j \neq i, \sqrt{\sum_{f \in S_k} \left(f(x_i) - f(x'_i)\right)^2} \leq \sqrt{\sum_{f \in S_k} \left(f(x_i) - f(x_j)\right)^2}$$

5.2.2 Clustering-based metrics

In this procedure, samples are clustered with a standard K-means algorithm (Hartigann and Wong, 1979) **considering only the features in** S_k. Similarly as for the p-NN classifier, relying on a k-dimensional space instead of a D-dimensional one helps escape the curse of dimensionality and provide reliable clusters.

Notations As a result of K-means clustering, each sample is assigned to a unique cluster. This essentially amounts to equipping each datapoint $\boldsymbol{x_i}$ with a pseudo-label (the corresponding cluster number). Let $\widehat{\boldsymbol{f}^*}$ denote the resulting new feature vector, containing κ unique values (κ governing the number of clusters). Let c denote the number of classes in $\boldsymbol{f}^*$. $\forall (i,j) \in [1,\dots,\kappa] \times [1,\dots,c]$, let A_i and B_j respectively denote the set of samples belonging to the i-th cluster and the set of samples belonging to the j-th class.

. Given the above framework, two performance indicators are typically designed to assess the *relevance* of the clusters. Section 5.2.2.1 introduces a first criterion measuring the homogeneity of the clusters. Section 5.2.2.2 thereafter presents an alternate definition of relevance rooted in information theory.

In order to fully uncover the learning goal, it is clear that K-means clustering should be performed with $\kappa \geq c$. However, tuning κ within the range $[c, n]$ is both challenging and crucial to the success of clustering-based performance indicators, as will be discussed in section 5.2.2.3.

5.2.2.1 Cluster purity-based performance indicator

The goal is here to measure how well aligned $\widehat{\boldsymbol{f}^*}$ is with $\boldsymbol{f}^*$, up to a reordering of the clusters. Formally, let $\Theta_{\kappa \to c}$ denote the family of functions $\theta : [1,\dots,\kappa] \to [1,\dots,c]$.

The *accuracy score (ACC)* (Cai et al., 2010; Li et al., 2012) of the selection subset S_k is then defined as :

$$\text{ACC}(S_k, \boldsymbol{f}^*) = \max_{\theta \in \Theta_{\kappa \to c}} \frac{1}{n} \sum_{i=1}^{\kappa} \sum_{j=1}^{c} |A_{\theta(i)} \cap B_j| \tag{5.3}$$

This can be rewritten from the perspective of the samples :

$$\text{ACC}(S_k, \boldsymbol{f}^*) = \max_{\theta \in \Theta_{\kappa \to c}} \frac{1}{n} \sum_{i=1}^{n} \delta\left(\theta\left(\widehat{\boldsymbol{f}^*}(x_i)\right), \boldsymbol{f}^*(x_i)\right) \tag{5.4}$$

It directly follows from *eq.* (5.4) that $\forall S_k \in F, 0 \leq \text{ACC}(S_k, \boldsymbol{f}^*) \leq 1$, tightness of the upper bound being guaranteed by $\kappa \geq c$. The higher $\text{ACC}(S_k, \boldsymbol{f}^*)$, the better the clusters allow identifying the different target concepts, the better the selection.

The ranking of the three declinations of AGNOS and baseline unsupervised FS algorithms w.r.t. the ACC score will be discussed in our empirical study (*chap.* 6).

5.2.2.2 Information theoretic performance indicator

Following information theory (Cover and Thomas, 2012), $\widehat{\boldsymbol{f}^*}$ and $\boldsymbol{f}^*$ can be interpreted as the respective realizations of two random variables. The core idea is then to measure the reduction of uncertainty (*a.k.a. entropy*) concerning the realization $\boldsymbol{f}^*$ gained from knowing the realization $\widehat{\boldsymbol{f}^*}$.

The individual entropies of $\widehat{\boldsymbol{f}^*}$ and $\boldsymbol{f}^*$ are respectively denoted $H(\widehat{\boldsymbol{f}^*})$ and $H(\boldsymbol{f}^*)$, defined as :

$$\begin{aligned} H(\widehat{\boldsymbol{f}^*}) &= -\sum_{i=1}^{\kappa} \frac{|A_i|}{n} \log\left(\frac{|A_i|}{n}\right) \\ H(\boldsymbol{f}^*) &= -\sum_{j=1}^{c} \frac{|B_j|}{n} \log\left(\frac{|B_j|}{n}\right) \end{aligned} \tag{5.5}$$

Accordingly, the joint entropy $H(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*)$ is given by :

$$H(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*) = -\sum_{i=1}^{\kappa}\sum_{j=1}^{c} \frac{|A_i \cap B_j|}{n} \log\left(\frac{|A_i \cap B_j|}{n}\right) \tag{5.6}$$

The *mutual information (MI)* (Banerjee et al., 2005) performance indicator then measures how much knowledge about learning goal $\boldsymbol{f}^*$ can be inferred from $\widehat{\boldsymbol{f}^*}$. $MI(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*)$ corresponds to the sum of individual entropies minus the joint entropy, that is :

$$MI(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*) = H(\widehat{\boldsymbol{f}^*}) + H(\boldsymbol{f}^*) - H(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*) \tag{5.7}$$

The higher the MI, the more informative the clustering for identifying the target, the better the selection.

By equations (5.5) and (5.6) :

$$\begin{aligned} 0 \leq H(\widehat{\boldsymbol{f}^*}) &\leq log(\kappa) \\ 0 \leq H(\boldsymbol{f}^*) &\leq log(c) \\ H(\widehat{\boldsymbol{f}^*})\max\big(H(\widehat{\boldsymbol{f}^*}), H(\boldsymbol{f}^*)\big) \leq H(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*) &\leq H(\widehat{\boldsymbol{f}^*}) + H(\boldsymbol{f}^*) \end{aligned}$$

Injecting these upper and lower bounds in *eq.* (5.7) yields :

$$0 \leq MI(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*) \leq \min\big(log(\kappa), log(c)\big) \tag{5.8}$$

Eq. (5.8) is source of an interpretability issue of the MI : the result $MI(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*) = 1$ can depict an arbitrarily good or bad selection subset, depending on the relative values of κ, c and n.

In order to obtain interpretable results, numerous possible normalizations of the MI have been proposed over the years, such as the *minimum Normalized Mutual Information* (NMI_{min}) (Liu et al., 2008), *joint NMI* (NMI_{joint}) (Yao, 2003), *square root NMI* (NMI_{sqrt}) (Strehl and Ghosh, 2002) or *Adjusted Mutual Information (AMI)* (Vinh et al., 2009). Each normalization variant comes with specific pros and cons. For instance, the

NMI_{min} [4] is ill-suited for continuous learning goals s.t. $c = n$, given that *any* selection subset S_k leads to a perfect score $NMI_{min} = 1$. A thorough discussion of the respective benefits and limitations of each normalization procedure can be found in Vinh et al. (2010).

For the sake of *unsup.* FS assessment, the most commonly used NMI variant (Zhao and Liu, 2007; Cai et al., 2010) is the *mean NMI* (NMI_{mean}) (Kvalseth, 1987), which we will in the rest of this work simply refer to as NMI :

$$NMI(S_k, \boldsymbol{f}^*) = 2\frac{MI(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*)}{H(\widehat{\boldsymbol{f}^*}) + H(\boldsymbol{f}^*)} \tag{5.9}$$

The higher the NMI, the better the selection. A NMI score of 0 indicates no mutual information, while the maximum score of 1 indicates perfect correlation. The empirical ranking of the three declinations of AGNOS and baseline unsupervised FS algorithms w.r.t. the NMI score will be provided in chapter 6.

5.2.2.3 Tuning the number of clusters κ

The process of normalizing the mutual information does not remove the dependency of the performance indicator on κ. Most notably, the higher κ, the higher the NMI, as illustrated in figure 5.1.

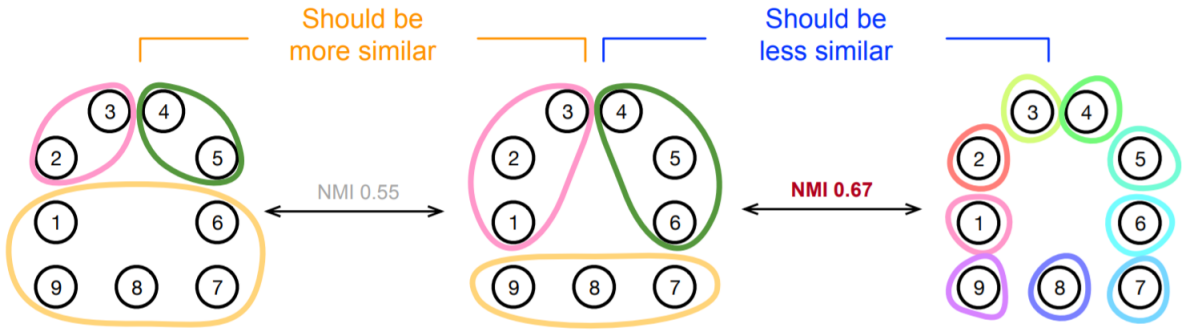

FIGURE 5.1: Despite the left clustering clearly being more similar to the central configuration than the right clustering, the NMI score is higher for the latter. From Gates et al. (2018)

The ACC score also faces the same issue, although admittedly to a lesser extent (more in *chap.* 6). In order to limit the bias in both performance indicators caused by large values of κ, we will in our experimental validation set κ to the minimal value allowing perfect scores, that is $\kappa = c$.

Other clustering-based *sup.* performance indicators such as e.g. the Variation of Information (Meilă, 2003) also depend on κ, oftentimes in a non-monotonous fashion (*fig.* 5.2).

The number of clusters is therefore a crucial hyperparameter to all aforementioned clustering-based measurements, s.t. fine-tuning κ is necessary to fairly compare *unsup.* FS algorithms. Consequently, these performance indicators do not comply with the requirement of *simplicity*.

4. Formally, $\text{NMI}_{min}(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*) = \frac{MI(\widehat{\boldsymbol{f}^*}, \boldsymbol{f}^*)}{\min\left(H(\widehat{\boldsymbol{f}^*}), H(\boldsymbol{f}^*)\right)}$.

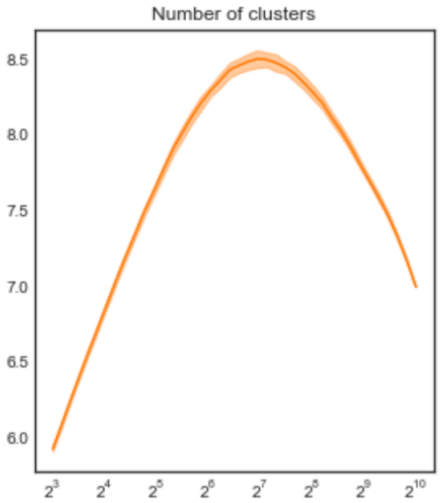

FIGURE 5.2: The Variation of Information (y-axis) performance indicator exhibits an unintuitive behavior as the number κ of clusters changes (x-axis), on a large toy dataset. From Gates et al. (2018)

In order to fulfill this requirement and lower the importance of κ, Gates et al. (2018) propose an *element-centric (EC)* clustering-based indicator. While this type of indicator has to the best of our knowledge not yet been utilized for the purpose of *unsup.* FS assessment, it is in any case ill-suited, as the validation procedure would still be **supervised**. Thus, the crux of the problem remains, as exposed in our earlier claim : *sup.* performance indicators, by design, lack *impartiality*, *expressivity* and *stability*. The following section will further discuss this claim.

5.2.3 Discussion

sup. performance indicators incentivize selecting the features most useful to predict a particular goal. More specifically and as underlined by the notations $ER(S_k, \boldsymbol{f}^*)$, $ACC(S_k, \boldsymbol{f}^*)$ and $NMI(S_k, \boldsymbol{f}^*)$, *sup.* performance criteria depend not only on the selection subset, but also on the configuration of the target variable ; $\boldsymbol{f}^*$ being by definition unaccounted for in *unsup.* FS. Therefore, **selection algorithms are ranked in an arbitrary order depending on an external unknown variable**. This *partiality* issue is illustrated in figure 5.3.

Moreover, *sup.* performance indicators only consider the link between the selection subset S_k and $\boldsymbol{f}^*$. All knowledge related to the rejected subset $F \backslash S_k$ is discarded. Given that *all* original features are potential *unsup.* learning goals (LeCun, 2016), *sup.* indicators actually leverage very little information, and therefore also lack *expressivity*.

A byproduct of the limited expressiveness of *sup.* assessment is that ignoring $D - k$ features means that *the amount of information "wasted" depends on* k. Arguably, this hinders the *stability* of the validation procedure, as will be empirically demonstrated in chapter 6.

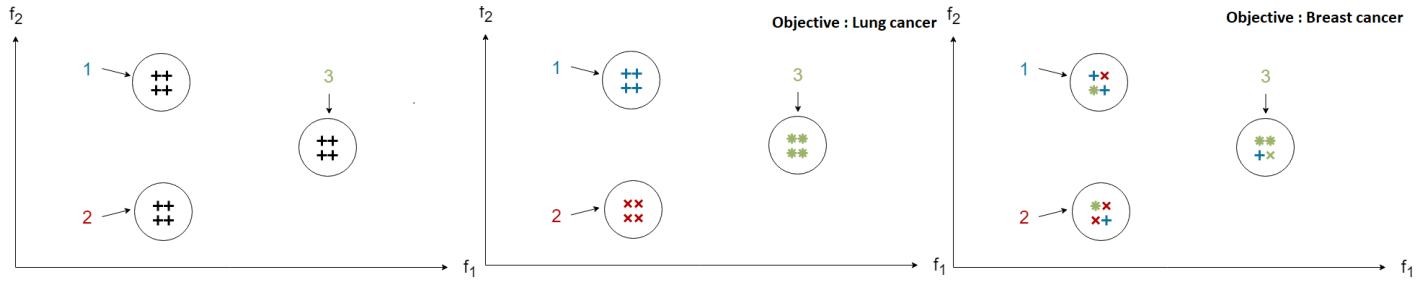

FIGURE 5.3: Clustering of medical data (left) based on the same *unsup.* selection subset leads to a perfect performance w.r.t. a certain learning goal (middle) and poor performance w.r.t. another goal (right).

5.3 *Unsup.* assessment of *unsup.* FS

5.3.1 The proposed FIT criterion

In accordance with the principle that any initial feature is potentially valuable, an *unsup.* performance indicator should evaluate the ability of S_k to recover *all* original variables simultaneously, rather than a single specific $\boldsymbol{f}^*$.

As such, we propose an intuitive adaptation of the *sup.* classifier-based criterion from section 5.2.1 to the *unsup.* context ; given S_k, D Nearest-Neighbor regressors $\chi_1, \ldots, \chi_D$ are trained from S_k to respectively recover $\boldsymbol{f_1}, \ldots, \boldsymbol{f_D}$. The motivation for relying on regressors rather than classifiers is that F may simultaneously contain categorical and continuous features.

The score of S_k is then derived from the average reconstruction error of the regressors. A reconstruction error is an unbounded quantity, and thus faces the same interpretability issue as the MI (*sec.* 5.2). We therefore rely on the R^2 score instead, which we now introduce.

The coefficient of determination R^2 Given y a regression target, $\bar{y}$ the mean of y across all samples and $\hat{y}$ the target predicted by a regressor, the R^2 score is defined as :

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5.10}$$

The R^2 score measures the proportion of the variance in y that is accounted for by the regression model. $R^2(y, \hat{y})$ is therefore an indicator of "goodness-of-fit". The maximal R^2 score of 1 indicates that the regressor predicts y perfectly. If regression fails and the model output is $\bar{y}$ for all datapoints (meaning the input variables

are disregarded), then $R^2(y, \hat{y}) = 0$. Interestingly enough, $R^2(y, \hat{y})$ can actually reach arbitrarily large negative values [5], indicating that the regression model is worse than constant $\bar{y}$ output.

If original features are normalized and centered, $\bar{y} = 0$ and $\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n} y_i^2 = \sigma(y) = 1$. Therefore, *eq.* (5.10) can be rewritten as :

$$R^2(y, \hat{y}) = 1 - MSE(y, \hat{y}) \tag{5.11}$$

Consequently, the R^2 score can in the context of this work be interpreted as a normalization of the MSE. This leads to the following algorithm :

Algorithm 8 The *unsup.* FIT criterion

Input : Dataset X, Selection subset S_k
Parameter: Number of neighbors p
Output : $FIT(S_k)$
Let $R^2_{avg}(S_k) = 0$.
for $i = 1, \ldots, n$ **do**
  Find the p nearest neighbors $\boldsymbol{x_i^1}, \ldots, \boldsymbol{x_i^p}$ of $\boldsymbol{x_i}$ w.r.t. S_k
end
for $j = 1, \ldots, D$ **do**
  Fit $\boldsymbol{f_j}$ from S_k using the p-NN regressor χ_j
  $R^2_{avg}(S_k) \leftarrow R^2_{avg}(S_k) + R^2\left(\boldsymbol{f_j}, \boldsymbol{\chi_j}(S_k)\right)$
end
Return $FIT(S_k) = \frac{R^2_{avg}(S_k)}{D}$

The time complexity of computing MSE-based scores is negligible compared to that of searching for the p nearest neighbors of each datapoint. Furthermore, the structure of the pseudocode in *alg.* 8 underlines that the neighborhood search needs only be performed once, rather than D times. Consequently, the FIT score is only marginally less *cost-efficient* than the *sup.* classifier-based criterion.

5.3.2 Discussion

The ideal selection subset S_k^* w.r.t. FIT best supports the reconstruction of the *whole* dataset. This entails three important consequences.

First of all and by contrast with *sup.* performance indicators, the FIT assessment procedure is not swayed towards features most relevant for a specific purpose, hinting at *impartiality*. Naturally, one cannot claim the approach is devoid of bias, as the retained features in S_k might be improper to the particular prediction of any considered feature (e.g. due to their distribution). This could be further alleviated by building a classifier

5. The R^2 notation is therefore slightly misleading. In order to lift this ambiguity, alternate notations such as the Nash-Sutcliffe Efficiency (NSE) (McCuen et al., 2006) are sometimes preferred instead.

based on S_k for each feature. This approach however suffers from fundamental and computational issues, see below.

By construction, FIT actually exploits the information carried by *every* variable rather than only S_k and $\boldsymbol{f}^*$, thus enjoys a higher *expressivity* relatively to *sup.* indicators.

Lastly, the underlying goal of the FIT procedure is invariant w.r.t. the size k of the selection subset : *leave no feature behind*. k only governs the "budget" available to fulfill this goal. We argue that this invariance helps the FIT criterion to provide more stable results than *sup.* assessment (ignoring $D - k$ features). This *stability* property will be studied during the sensitivity analysis portion of our empirical study (*chap.* 6).

About *unsup.* ACC/NMI

In order to support the selection of a feature subset sufficient to recover the entire dataset, one could adapt clustering-based criteria such as ACC or NMI to the *unsup.* context, and specifically use the retained features to predict each $\boldsymbol{f_i}$ ranging among the other features. As said, this process however involves two related issues.

Both ACC and NMI rely on building clusters ; the number κ_i of such clusters should depend on the considered feature $\boldsymbol{f_i}$, be it categorical or continuous. On the one hand, the S_k based clustering procedure should be repeated many times, scaling poorly in the large D regime. On the other hand, this would require fine-tuning κ_i for each f_i (*sec.* 5.2.2.3), thus failing the requirement of *simplicity*.

Summary

In this chapter, we have first discussed what constitutes a suitable performance indicator for assessing *unsup.* FS. The state-of-the-art *sup.* criteria were thereafter introduced. These criteria were claimed to admit significant limitations, most notably regarding their reliability and stability. A novel *unsup.* performance criterion, called *FIT*, was accordingly proposed to tackle these limitations, by considering the reconstruction of *all* original features simultaneously. The next chapter will empirically compare the respective merits of existing performance indicators and FIT.

Chapitre 6

Experimental validation

This chapter presents the experimental validation of AGNOS. The experimental setup is first introduced in section 6.1, along with a preliminary study regarding the intrinsic dimension of the benchmark datasets. Following chapter 5, the supervised performance of AGNOS-S, AGNOS-W and AGNOS-G is assessed and compared to baselines in section 6.2. Section 6.3 thereafter exhibits the unsupervised performance of AGNOS w.r.t. the novel proposed FIT criterion. A sensitivity study of the results is conducted in section 6.4, to assess the influence of the hyperparameters of the method. Section 6.5 concludes the chapter with a final discussion of the results.

6.1 Experimental setup and preliminary study

6.1.1 Experimental setup

6.1.1.1 The scikit-feature benchmark

Scikit-feature (skfeature for short) Li et al. (2018b); skf (2018) is an open-source feature selection repository developed at Arizona State University. It is built upon the widely used Python machine learning package scikit-learn. The skfeature databank currently contains 29 datasets commonly used in feature selection tasks and challenges. These datasets span multiple domains, including text data, face image data, genomics data, hand written text in image format, as well as artificial data specifically generated for feature selection purposes.

Skfeature also provides ready-made Python implementations for 34 popular feature selection algorithms (28 supervised and 6 unsupervised), facilitating empirical comparison of new FS approaches w.r.t. some state-of-the-art methods.

Due to its open source nature and ease of use, skfeature has risen in popularity as an international benchmark for feature selection Li et al. (2017); Chen et al. (2017). This justifies our usage of skfeature for the experimental validation of AGNOS.

6.1.1.2 Datasets

Experiments are carried on 8 datasets taken from the scikit-feature database, selected for their diversity in number of features, types (categorical and continuous) and domain (face image, sound processing and medical data). Complementary experiments on the other 21 datasets from the database have shown the representativity of the results obtained on the 8 chosen datasets. In all datasets but one (Isolet), the number of samples is small w.r.t. the number of features D. Dataset size, dimensionality, number of classes and data type are summarized in Table 6.1.

	# samples	# features	# classes	Data type
arcene	200	10000	2	Medical
Isolet	1560	617	26	Sound processing
ORL	400	1024	40	Face image
pixraw10P	100	10000	10	Face image
ProstateGE	102	5966	2	Medical
TOX171	171	5748	4	Medical
warpPie10P	130	2400	10	Face image
Yale	165	1024	15	Face image

TABLE 6.1: Summary of benchmark datasets.

6.1.1.3 Performance indicators

The three variants of AGNOS are compared to four *unsup.* baselines introduced in section 3.2.3 : the *Laplacian score* (He et al., 2005), *SPEC* (Zhao and Liu, 2007), *MCFS* (Cai et al., 2010) and *NDFS* (Li et al., 2012). The implementations of all baselines have been taken from the scikit-feature database, and all their hyperparameters have been set to their default values.

Four performance indicators have been considered, where the first three indicators correspond to the typical *sup.* assessment procedure (*chap.* 5 : the *sup.* accuracy of a p-NN classifier, the ACC score and the NMI score) and the fourth performance metric is the proposed *unsup.* FIT criterion.

6.1.1.4 Hyperparameters

In all experiments, AGNOS is ran using a single hidden layer Auto-Encoder with *tanh* activation functions for both encoder and decoder, Glorot parameter initialization (Glorot and Bengio, 2010), and the *Adam* (Ruder, 2016) gradient descent scheme, with initial learning rate of 10^{-3}. The number p of neighbors used for all p-NN regressors involved in the FIT score (*chap.* 5) is set to $p = 5$ [1]. Following section 5.2.2.3, the number of clusters to use for clustering-based performance indicators is set to $\kappa = c$ for all experiments, where c is the number of classes in the *sup.* learning goal.

1. Preliminary experiments have shown results to be more stable overall with $p = 5$ rather than $p = 1$.

The results provided in sections 6.2 and 6.3 were recorded with the following default hyperparameter values for AGNOS : hidden layer size $\boldsymbol{d = \widehat{ID}}$ (estimated intrinsic dimension of the data), sparsity penalty strength $\boldsymbol{\lambda = 1}$. The sensitivity of the results w.r.t. both d and λ will be assessed in section 6.4.

6.1.2 Intrinsic dimension and selection subset size

As seen (*chap.* 4), AGNOS includes two important preprocessing steps : i) feature normalization and ii) intrinsic dimension estimation. Therefore, we begin our experimental study by analyzing the ID estimation process. Section 6.1.2.1 presents the estimated ID for every benchmark dataset, as well as how these results are prone to change as a consequence of feature normalization. An analysis of the faithfulness of the ID estimator is thereafter conducted in section 6.1.2.2.

6.1.2.1 Intrinsic dimension estimation

Table 6.2 contains the estimated IDs for each dataset using the method from Facco et al. (2017) [2].

Dataset	Initial dimension	$\widehat{ID}$ of unnormalized data	$\widehat{ID}$ of normalized data
Arcene	10000	18.01	39.89
Isolet	617	8.29	8.53
ORL	1024	5.60	5.50
pixraw10P	10000	3.74	3.94
ProstateGE	5966	22.27	22.32
TOX171	5748	6.35	14.75
warpPIE10P	2400	2.63	2.62
Yale	1024	9.27	9.62

TABLE 6.2: Intrinsic dimensions of each dataset, using all samples.

The fact that the estimated ID is small compared to the original dimensionality for every dataset highlights the potential of feature selection for data compression. For 6 out of the 8 benchmark datasets, $\widehat{ID}$ is mostly unaffected by the rescaling of each feature to zero mean and unit variance.

However, for the remaining 2 datasets (Arcene and TOX171), the normalization process provokes a significant change in $\widehat{ID}$, which more than doubles in both cases. This suggests that the correlation between features can decrease as a result of normalization. This could limit the potential of data compression. However, feature normalization is mandatory to avoid bias in the selection (*chap.* 4), and is thus relied upon in AGNOS regardless of this drawback. We will therefore consider the $\widehat{ID}$ of *normalized* data in our experimental validation.

2. As said (*chap.* 2), the ID is not necessarily an integer. However, the number of neurons in the hidden layer of the Auto-Encoder used in AGNOS must itself be an integer. Therefore, we opt to round up the estimated intrinsic dimension : $\widehat{ID} \leftarrow \lceil \widehat{ID} \rceil$.

6.1.2.2 Assessing the quality of the ID estimation

As said, the results from table 6.2 are merely an **approximation**, *of unknown precision*, of the "true" intrinsic dimension of the data. We therefore attempt to estimate the quality of the approximation $\widehat{ID}$.

In order to simulate randomness[3], we opt to perform 20 uniform train/test splits for each dataset. $\widehat{ID}$ is thus estimated 20 times, considering for each run only the 80% of samples in the training set. The results are contained in table 6.3.

Dataset	Initial dimension	$\widehat{ID}$ of unnormalized data	$\widehat{ID}$ of normalized data
Arcene	10000	16.99 (0.74)	38.04 (7.20)
Isolet	617	9.02 (0.03)	9.25 (0.02)
ORL	1024	5.39 (0.08)	5.26 (0.07)
pixraw10P	10000	3.91 (0.19)	4.11 (0.15)
ProstateGE	5966	21.56 (2.90)	23.11 (4.36)
TOX171	5748	6.13 (0.12)	14.27 (0.94)
warpPIE10P	2400	2.58 (0.01)	2.59 (0.01)
Yale	1024	9.60 (0.98)	9.74 (0.91)

TABLE 6.3: Mean and variance (in parenthesis) of intrinsic dimension across 20 runs for every dataset, using 80% of samples drawn at random for each run.

On half of the benchmark datasets (namely Isolet, ORL, pixraw10P and warpPIE10P), the average estimated ID over the 20 splits is close to the $\widehat{ID}$ obtained with all samples (table 6.2, 10% variation or less), with a small variance (at most 5% of the expected value). This holds true for both the normalized and unnormalized versions of the data. The 2-NN method thus appears to be a trustworthy ID estimator for these datasets.

Although the mean estimated ID across splits is also consistent with table 6.2 for the remaining half of the benchmark datasets (namely Arcene, ProstateGE, TOX171 and Yale), we observe a high variance (between 7% and 20% of the expected value). This means that the results of a run are strongly dependent on which 20% of samples are omitted for ID estimation in that run. Therefore, this indicates that the ID estimator is sensitive w.r.t. outliers for these datasets, and that $\widehat{ID}$ is a brittle estimation of the true intrinsic dimension in those cases.

The mean estimated ID is twice larger in the normalized version of the data for Arcene and TOX171, which is in line with table 6.2. However, a novel observation is that feature normalization also leads to a disproportionate increase in variance in both cases (from 2% to 7% in TOX171 ; from 4% to 19% in Arcene). This suggests that rescaling does not modify the apparent intrinsic dimensionality uniformly across the dataset, increasing the sensitivity of the ID estimator.

The main takeaway of this preliminary analysis is that the constraint $d = \widehat{ID}$ should likely be relaxed in all three declinations of AGNOS, which section 6.4.2 will showcase the impact of.

3. The 2-NN ID estimator being a deterministic algorithm, we cannot simply perform multiple runs on the full dataset and examine the variance of the results.

6.1.2.3 Selection subset size

Following the manifold assumption and the discussion on intrinsic dimension (*chap.* 2), the intrinsic dimension ID of the data provides a lower bound for the selection subset size k by construction. In practice, it is unlikely that any selection subset of size $k = ID$ is able to fully recover the original feature set F. Furthermore, we only have access to an approximation $\hat{ID}$ of the "true" intrinsic dimension, of unknown precision.

In order to avoid selecting too few features and for the sake of cautiousness, we therefore want that $k > \widehat{ID}$. The wider the margin between $\widehat{ID}$ and k, the more likely S_k to be sufficient for recovering F. However, the larger k, the more similar the respective performances of the different considered unsupervised FS algorithms tend to be, as will be shown in section 6.4.

k should therefore ultimately be set to a value larger than the estimated intrinsic dimension of the datasets, but small enough that we can observe a stark contrast in performance among the baselines and the three declinations of AGNOS. As a consequence of preliminary experiments, we choose the default selection subset size to be $\boldsymbol{k = 100}$. For the sake of completeness, we will however study the behavior of the baselines and AGNOS for varying subset sizes in section 6.4.

6.2 Supervised evaluation results

Section 6.2.1 first assesses the stability of the performance of AGNOS (according to the 3 considered supervised performance indicators) w.r.t. the random initialization of the AutoEncoder parameters. AGNOS-S, AGNOS-G and AGNOS-W are then compared to the baselines in section 6.2.2.

6.2.1 Sensitivity w.r.t. initialization of network parameters

All baseline unsupervised FS methods considered are deterministic algorithms. By contrast, AGNOS is stochastic, given that it relies on training a neural network. The inherent randomness comes from the initialization of the AutoEncoder parameters (using the Glorot initialization of weights and biases (Glorot and Bengio, 2010)). A first step consists of examining the stability of AGNOS performance across runs, controlling the reliability of the results under an affordable time complexity budget.

Tables 6.4 and 6.5 display the results respectively obtained with the standard classification and the ACC scores of AGNOS-S, AGNOS-W and AGNOS-G on the benchmark datasets. Each table contains the mean and variance, over 10 runs with different initial network parameters, of the score.

The standard classification score (Table 6.4) is measured with a 5-NearestNeighbor classifier and $k = 100$ selected variables. The variance of the classification score is shown to be small (less than 1% of the mean) for every dataset and declination of AGNOS, establishing its low sensitivity w.r.t. the random initialization of the AutoEncoder.

Similar conclusions can be drawn from Table 6.5, which records the ACC scores using the same setup as above.

Dataset	AgnosS	AgnosW	AgnosG
Arcene	0.81(-)	0.77(0.003)	0.75(0.004)
Isolet	0.83(-)	0.84(-)	0.65(-)
ORL	0.93(-)	0.93(-)	0.89(-)
pixraw10P	0.97(-)	0.93(-)	0.99(-)
ProstateGE	0.76(-)	0.83(-)	0.75(-)
TOX171	0.66(0.001)	0.86(0.006)	0.63(0.007)
warpPIE10P	0.99(-)	0.98(-)	0.98(-)
Yale	0.63(-)	0.60(-)	0.61(-)

TABLE 6.4: Mean and variance of standard classification score of 5-NearestNeighbor classifier on the benchmark datasets for the three declinations of AGNOS, over 10 runs with different Glorot initializations of network parameters. (-) indicates a variance lower than 10^{-3}.

In both cases, and even more so in the ACC case, AGNOS-W and AGNOS-G variances are higher than for AGNOS-S. Specifically, the variance is negligible for AGNOS-S on all datasets but 2 (where it is 10^{-3}). In contrast, the variance is circa $4\,10^{-3}$ on all datasets but two for AGNOS-W and AGNOS-G.

Dataset	AgnosS	AgnosW	AgnosG
Arcene	0.67(-)	0.62(0.002)	0.63(0.002)
Isolet	0.54(0.001)	0.58(-)	0.41(0.001)
ORL	0.57(-)	0.55(-)	0.53(-)
pixraw10P	0.81(0.002)	0.64(0.009)	0.78(0.008)
ProstateGE	0.61(-)	0.59(0.006)	0.57(0.005)
TOX171	0.40(-)	0.29(0.006)	0.36(0.007)
warpPIE10P	0.27(-)	0.36(0.002)	0.42(0.002)
Yale	0.51(-)	0.38(-)	0.53(-)

TABLE 6.5: Mean and variance of ACC score on the benchmark datasets for the three declinations of AGNOS, over 10 runs with different Glorot initializations of network parameters. (-) indicates a variance lower than 10^{-3}.

Likewise, table 6.6 contains the mean and variance of the NMI scores. The variance of the NMI score is also less than 1% of the mean in most cases. There are however some outliers for which the variance is proportionally larger (e.g. $\sim 8\%$ of the mean for AGNOS-W on ProstateGE). This seems to occur only in the event that the mean NMI score is itself small (less than 0.10).

Given the overall stability of the results, we will in the remainder of this chapter neglect the variance due to random network parameter initialization and consider only the expected value.

Dataset	AgnosS	AgnosW	AgnosG
Arcene	0.08(-)	0.02(-)	0.04(-)
Isolet	0.69(-)	0.70(-)	0.58(-)
ORL	0.76(-)	0.77(-)	0.73(-)
pixraw10P	0.87(-)	0.76(0.003)	0.81(0.004)
ProstateGE	0.06(-)	0.13(0.006)	0.01(-)
TOX171	0.23(0.001)	0.15(0.009)	0.08(0.002)
warpPIE10P	0.28(0.001)	0.35(0.003)	0.33(0.003)
Yale	0.54(-)	0.50(-)	0.56(-)

TABLE 6.6: Mean and variance of NMI score on the benchmark datasets for the three declinations of AGNOS, over 10 runs with different Glorot initializations of network parameters. (-) indicates a variance lower than 10^{-3}.

Dataset	AgnosS	AgnosW	AgnosG	Laplacian	MCFS	NDFS	SPEC
Arcene	**0.81**	0.77	0.75	0.67	0.52	0.69	0.70
Isolet	0.83	**0.84**	0.65	0.68	0.65	0.82	0.74
ORL	**0.93**	**0.93**	0.89	0.92	0.90	0.91	0.87
pixraw10P	0.97	0.93	**0.99**	**0.99**	0.95	0.98	0.85
ProstateGE	0.76	**0.83**	0.75	0.75	0.74	0.71	0.70
TOX171	0.66	**0.86**	0.63	0.84	0.74	0.67	0.78
warpPIE10P	**0.99**	0.98	0.98	**0.99**	0.96	0.98	0.98
Yale	**0.63**	0.60	0.61	0.56	0.53	**0.63**	0.58

TABLE 6.7: Supervised classification scores of 5-NearestNeighbor classifier for the three declinations of AGNOS and the baselines on the benchmark datasets. Statistically significantly better (according to a t-test with a p-value of 0.05) results in boldface.

6.2.2 Comparison with the baselines

Tables 6.7, 6.8 and 6.9 respectively contain, for the three variants of AGNOS and the four baselines, the classification, ACC and NMI scores on every benchmark dataset. In particular, AGNOS-S appears to perform better on average than the group-LASSO based AGNOS-W and AGNOS-G, with the default hyperparameter values ($k = 100$, $d = \widehat{ID}$, $\lambda = 1$). The validity of this conclusion in other regions of the hyperparameter space will be investigate during the sensitivity study (*sec.* 6.4).

Interestingly, on the two high-dimensional image datasets *pixraw10P* and *warpPie10P*, the Laplacian method matches respectively AGNOS-G and AGNOS-S w.r.t. the classification score.

On the remaining lower dimensionality image dataset Yale, NDFS matches the results of AGNOS-S.

On *Isolet*, AGNOS-W and to a lesser extent AGNOS-S outperform all other algorithms, with NDFS ranking third.

Dataset	AgnosS	AgnosW	AgnosG	Laplacian	MCFS	NDFS	SPEC
Arcene	**0.67**	0.62	0.63	0.66	0.56	0.51	0.66
Isolet	0.54	**0.58**	0.41	0.48	0.41	0.57	0.57
ORL	**0.57**	0.55	0.53	0.55	0.56	0.54	0.47
pixraw10P	**0.81**	0.64	0.78	0.80	0.75	0.78	0.48
ProstateGE	**0.61**	0.59	0.57	0.58	0.59	0.57	0.59
TOX171	0.40	0.29	0.36	0.45	**0.48**	0.46	0.47
warpPIE10P	0.27	0.36	**0.42**	0.29	0.36	0.29	0.33
Yale	0.51	0.38	**0.53**	0.44	0.40	0.44	0.40

TABLE 6.8: Supervised ACC scores for the three declinations of AGNOS and the baselines on the benchmark datasets. Statistically significantly better (according to a t-test with a p-value of 0.05) results in boldface.

Dataset	AgnosS	AgnosW	AgnosG	Laplacian	MCFS	NDFS	SPEC
Arcene	0.08	0.02	0.04	0.09	**0.20**	0.01	0.09
Isolet	0.69	**0.70**	0.58	0.62	0.56	**0.70**	0.69
ORL	0.76	0.77	0.73	0.76	**0.78**	0.74	0.70
pixraw10P	**0.87**	0.76	0.81	0.86	0.86	0.84	0.66
ProstateGE	0.06	**0.13**	0.01	0.02	0.02	0.01	0.02
TOX171	0.23	0.15	0.08	0.27	0.22	**0.33**	0.24
warpPIE10P	0.28	**0.35**	0.33	0.30	0.30	0.34	0.34
Yale	0.54	0.50	**0.56**	0.49	0.52	0.48	0.44

TABLE 6.9: Supervised NMI scores for the three declinations of AGNOS and the baselines on the benchmark datasets. Statistically significantly better (according to a t-test with a p-value of 0.05) results in boldface.

On *ORL*, AGNOS-S and AGNOS-W outperform others, though Laplacian and NDFS obtain close performances. On Arcene, AGNOS-S significantly outperforms all other methods, while AGNOS-W does so on ProstateGE and TOX171, medical datasets with comparatively high intrinsic dimension.

Overall, AGNOS is therefore shown to be competitive with the baselines in terms of supervised evaluation ; specifically, the best recorded performance is achieved by a declination of AGNOS on all datasets w.r.t. the classification score, 7 out of 8 datasets w.r.t. the ACC score, and 5 out of 8 w.r.t. the NMI score.

These first *sup.* results are encouraging. However, based on the claim that the *sup.* assessment is brittle, we shall delay the discussion regarding the respective performances of AGNOS-S, AGNOS-W and AGNOS-G to their *unsup.* assessment in *sec.* 6.3.

Discussion

On one hand, we set the number of clusters κ used in *sup.* clustering to the minimal value c corresponding to the number of *sup.* classes in the dataset. On the other hand, the NMI score tends to be positively correlated with κ (*sec.* 5.2.2.3). Consequently, we expect the NMI to be positively correlated with c.

	AgnosS	AgnosW	AgnosG	Laplacian	MCFS	NDFS	SPEC
ACC	-0.01	0.14	-0.15	-0.11	-0.14	-0.12	-0.20
NMI	0.72	0.79	0.73	0.71	0.70	0.71	0.80

TABLE 6.10: Correlation between scores and number of classes in the dataset, for the three declinations of AGNOS and the baselines.

Table 6.10 contains the correlations between the ACC and NMI scores on one hand, and c on the other hand. Expectedly, the NMI score is strongly correlated to c. The correlation between the ACC score and c is much lower than for the NMI, for all algorithms[4]. The ACC score should accordingly be prioritized over the NMI score if no prior knowledge is available to tune κ.

6.3 Unsupervised evaluation results

Section 6.3.1 first assesses the stability of the FIT score of AGNOS w.r.t. the randomness of the initial conditions ; section 6.3.2 provides a comparison of the three declinations of AGNOS with the baselines, which is further discussed in section 6.3.3.

6.3.1 Sensitivity w.r.t. initialization of network parameters

Similarly as for *sup.* assessment, the variance in the results due to the Glorot initialization is minimal (less than 1% of the mean value) for all datasets and declinations of AGNOS, though AGNOS-W and AGNOS-G appear to be slightly less stable than AGNOS-S. Overall, the FIT score is shown to be hardly sensitive w.r.t. initial conditions. We will therefore neglect this source of variance in the remainder of this chapter, recording only the mean value.

6.3.2 Comparison with the baselines

Table 6.12 contains the respective FIT scores of the considered FS algorithms over the benchmark datasets. The proposed AGNOS-S is shown to achieve a higher FIT score than the baselines on all datasets. These results empirically demonstrate that the selection subsets induced by AGNOS-S retain more information about the features *on average* than the baselines.

4. Interestingly enough, this correlation is slightly positive for AGNOS-W and negative for the baselines and AGNOS-G. This suggests that the baselines and AGNOS-G are less adequate for multi-label classification than AGNOS-W and AGNOS-S.

Dataset	AgnosS	AgnosW	AgnosG
Arcene	0.610(-)	0.460(6×10^{-4})	0.560(5×10^{-4})
Isolet	0.763(-)	0.762(-)	0.701(-
ORL	0.800(-)	0.795(-)	0.780(-)
pixraw10P	0.855(-)	0.782(4×10^{-4})	0.832(3×10^{-4})
ProstateGE	0.662(-)	0.620(3×10^{-4})	0.606(3×10^{-4})
TOX171	0.581(-)	0.580(2×10^{-4})	0.528(3×10^{-4})
warpPIE10P	0.910(-)	0.897(-)	0.901(-)
Yale	0.703(-)	0.696(-)	0.671(-)

TABLE 6.11: Mean and variance of FIT score on the benchmark datasets for the three declinations of AGNOS, over 10 runs with different Glorot initializations of network parameters. (-) indicates a variance lower than 10^{-4}.

	Arcene	Isolet	ORL	pixraw10P	ProstateGE	TOX171	warpPIE10P	Yale
AgnoS-S	**0.610**	**0.763**	**0.800**	**0.855**	**0.662**	**0.581**	**0.910**	**0.703**
AgnoS-W	0.460	0.762	0.795	0.782	0.620	0.580	0.897	0.696
AgnoS-G	0.560	0.701	0.780	0.832	0.606	0.528	0.901	0.671
Laplacian	0.576	0.680	0.789	0.840	0.655	0.563	0.903	0.601
MCFS	0.275	0.720	0.763	0.785	0.634	0.549	0.870	0.652
NDFS	0.490	0.747	0.796	0.835	0.614	0.520	0.904	0.677
SPEC	0.548	0.733	0.769	0.761	0.646	0.559	0.895	0.659

TABLE 6.12: FIT score of 5-NearestNeighbors regressor using the top 100 ranked features. Statistically significantly (according to a t-test with a p-value of 0.05) better results in boldface.

By constrast, AGNOS-W and AGNOS-G are both outperformed by at least one baseline on every benchmark dataset (with the exception of AGNOS-W on TOX171 and Yale). Our interpretation is that this is due to a key difference between the LASSO regularization and the slack variables. One one hand, the importance of initial feature $\boldsymbol{f_j}$ is in AGNOS-W and AGNOS-G obtained by taking the maximum of a d-dimensional vector. This process therefore ignores the behavior of $d-1$ latent features. On the other hand, the importance of $\boldsymbol{f_j}$ is directly given by the single positive real value $|a_j|$, summarizing its influence over all d latent variables simultaneously, which tentatively explains why this allows the slack variable layer to better reflect the influence of the original variables.

6.3.3 Discussion

Stability of results is not stability of selection While the three variants of AGNOS obtain very stable results according to both *sup.* and *unsup.* performance indicators (w.r.t. the random initialization of the Auto-Encoder parameters), the selected features themselves are not. Quite the contrary, the overlap between

selection subsets resulting of two different runs of AGNOS may be as low as 5% (95% of features selected in one run are rejected in the other), for all datasets.

Given that both *sup.* and *unsup.* performance indicators are stable across runs however, suggesting that different selected subsets carry the same information, the variability of the selected features is explained from the feature redundancy within a dataset : typically, in the case of several copies of a same feature, one of these copies should be selected indifferently by AGNOS. More generally, AGNOS indiscriminately picks one representative per cluster of correlated features.

Leaving no feature behind Figure 6.1 depicts the respective cumulative distribution functions of the R^2 scores achieved by a 5-NearestNeighbors regressor using the top 100 ranked features by the baseline methods and the three AGNOS variants, on *Arcene*. A first observation is that every FS algorithm leads to accurate fitting (R^2 score > 0.8) of *some* features and poor fitting (R^2 score < 0.2) on *some* other features. This shows that the quality of the model predictions is very sensitive w.r.t. the target variable, which is an additional supportive argument to our claim that *sup.* assessment of *unsup.* FS (dealing with a *single* target) is unreliable. Most importantly, FS algorithms differ in the number of poorly fitted features. R^2 scores < 0.2 are achieved for less than 20% of features using any declination of AGNOS and more than 35% of features using MCFS. This shows that on this example dataset, AGNOS retains information about more features than MCFS.

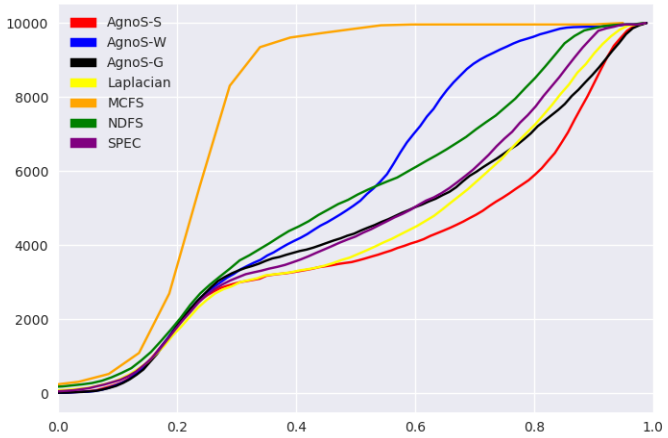

FIGURE 6.1: Cumulative distribution functions of the R^2 scores of a 5-NearestNeighbors regressor using the top 100 ranked features on *Arcene*. If a point has coordinates (x, y), then the goodness-of-fit of the regressor is $\leq x$ for y initial features (the lower, the better).

Unreliabilty of *sup.* assessment Table 6.13 contains the respective frequencies of ranks attained by each selection method w.r.t. the R^2 scores of each feature on the *warpPIE10P* dataset. A first observation is that

not only is AGNOS-S more often ranked first than the baselines, it is also least often ranked last. This property also holds true for AGNOS-W and AGNOS-G, although the contrast with the baselines is less pronounced. This is once again in line with the idea of reconstructing every initial variable.

	1	2	3	4	5	6	7
AgnoS-S	0.37	0.09	0.12	0.09	0.17	0.07	0.08
AgnoS-W	0.16	0.11	0.09	0.23	0.07	0.21	0.13
AgnoS-G	0.12	0.17	0.11	0.18	0.05	0.24	0.13
LAP	0.12	0.17	0.20	0.17	0.04	0.17	0.13
MCFS	0.10	0.23	0.11	0.19	0.04	0.16	0.17
NDFS	0.08	0.21	0.13	0.11	0.11	0.09	0.27
SPEC	0.04	0.03	0.25	0.03	0.52	0.05	0.09

TABLE 6.13: Frequency of ranks of selection methods w.r.t. R^2 scores of each feature on warpPIE10P with a 5-NearestNeighbors regressor using the top 100 ranked features. For instance, AGNOS-S obtains the lowest reconstruction error among the 7 candidate methods for 37% of the original features.

Most importantly, every FS algorithm is able to achieve *any* rank for some original features. Therefore, ***the ranking of unsup. FS w.r.t. sup. assessment is extremely brittle, depending on the target variable considered***. This confirms our claim (*chap.* 5) that *sup.* assessment is *partial*, thus unreliable. It remains to show that the FIT score itself is stable w.r.t. hyperparameters d, k and λ (*sec.* 6.4).

6.4 Sensitivity study

We will study the sensitivity of the 3 aforementioned supervised performance indicators, as well as the proposed FIT score, w.r.t. three parameters :

— The size k of the selection subset (section 6.4.1).

— The size d of the hidden layer of the AutoEncoder (section 6.4.2).

— The strength λ of the sparsity penalty term in the AutoEncoder loss function (section 6.4.3).

In order to conduct this sensitivity study, we will record the performance indicators with 2 of the 3 parameters set to their respective default values and the remaining parameter varying across a wide range [5]. The default values are the same as in section 6.1, namely : $\{k = 100; d = \widehat{ID}, \lambda = 1\}$.

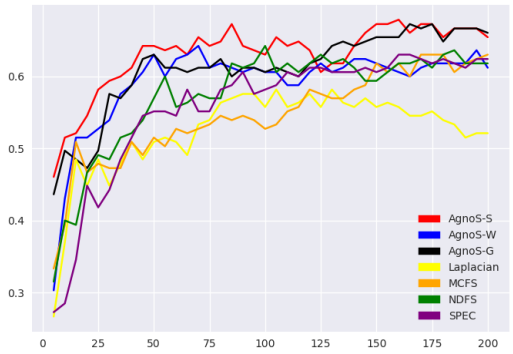

FIGURE 6.2: Prediction score of 5-NearestNeighbor classifier w.r.t. selection subset size k, on Yale

6.4.1 Sensitivity w.r.t. number of selected features k

6.4.1.1 Classification-based criterion

Figure 6.2 depicts the classification accuracy achieved by the 5-NearestNeighbor classifier trained only from the selection subset S_k as a function of k, on the Yale dataset. A first observation is that this supervised score expectedly appears to increase with k overall. Additionally, AGNOS-S outperforms all other methods for $k \leq 100$. Thereafter, it is locally overtaken by NDFS and globally matched by AGNOS-G. There is therefore a clear-cut best candidate on this dataset. However, the ranking of "middle of the pack" algorithms is unstable : AGNOS-W, NDFS and SPEC regularly overtake each other. This makes it difficult to precisely gauge the efficiency of the respective FS methods with this performance indicator.

Similar results are visible on *fig.* 6.3, depicting the results on ProstateGE. On this dataset, the clear-cut best candidate is AGNOS-W. For the other six methods, the performances are brittle and may be highly non monotonous w.r.t. k ; note for instance the sharp decreasef in classification accuracy of NDFS when k passes from 10 to 15. This non-monotony is blamed on the addition of features irrelevant for predicting the target, hindering the classifier.

In conclusion, the supervised performance indicator is too sensitive w.r.t. k to provide a consistent ranking of methods. [6]

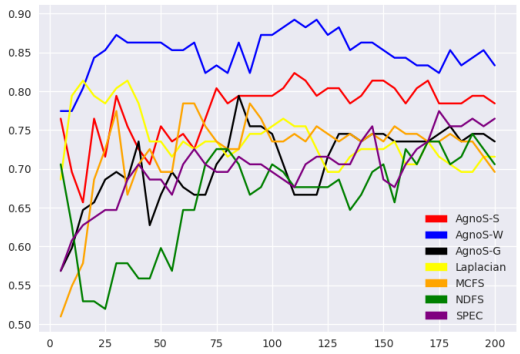

FIGURE 6.3: Prediction score of 5-NearestNeighbor classifier w.r.t. selection subset size k, on ProstateGE

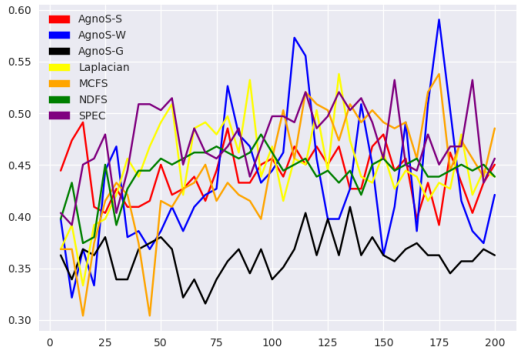

FIGURE 6.4: ACC score w.r.t. selection subset size k, on TOX171

6.4.1.2 Clustering-based criteria

Figures 6.4 and 6.5 depict the sensitivity w.r.t. k of the ACC score on TOX171 (resp. of the NMI score on warpPIE10P). Both these performance criteria appear to be highly sensitive w.r.t. the selection subset size, and they do not support any consistent (dataset-dependent) ranking of the methods. Similarly chaotic performance curves were observed on all other benchmark datasets. This sensitivity could admittedly be

5. This technique will allow us to measure the sensitivity of the results in a small region of the $\mathbb{R}^3$ hyperparameter space, centered around the coordinates corresponding to the default values. In order to obtain a more comprehensive overview of result sensitivity across the hyperparameter landscape, one should instead turn to a fine-grained grid search.

6. Note that, even if the resulting ranking were stable w.r.t. k, the *partiality* problem exposed in chapter 5 would remain.

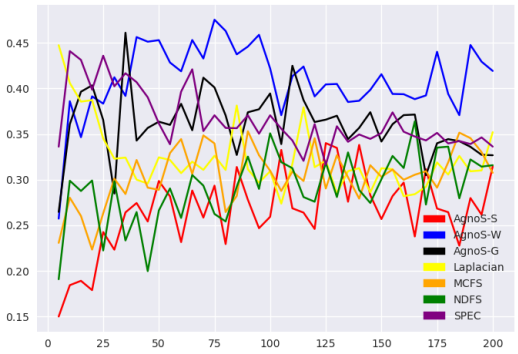

FIGURE 6.5: NMI score w.r.t. selection subset size k, on warpPIE10P

decreased by fine-tuning the number κ of clusters considered beforehand ; nevertheless these experiments suggest that these two criteria are ill-suited to compare FS algorithms fairly.

6.4.1.3 FIT criterion

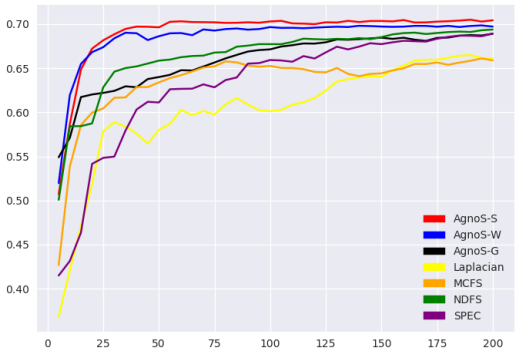

FIGURE 6.6: FIT score w.r.t. selection subset size k, on Yale

Figure 6.6 depicts the FIT score attained by the 3 declinations of AGNOS and the baselines as a function of the selection subset size k, on the Yale dataset.

For all considered FS algorithms, the FIT score appears to be a non-decreasing function of k, nonwithstanding a few exceptions (e.g. the decrease between 30 and 40 selected features for the Laplacian approach).

Additionally, the ranking of selection methods is shown to be robust w.r.t. k[7], with AGNOS-S attaining the top rank for any $k > 20$. Concordant results were observed on the other benchmark datasets.

Following these two observations, the proposed FIT score appears much more reliable for ranking unsupervised FS approaches than the 3 considered supervised performance indicators, in the absence of prior knowledge regarding the desired size of the selection subset.

The smoothness of the performance curves was expected as the FIT indicator is an average over thousands of elements (the original features), whereas the supervised criteria are obtained from a *single* target, leading to the irregular curves shown previously.

Furthermore, the gap in performance between the top and bottom ranked algorithms expectedly decreases as k increases (it would eventually be 0 for $k = D$). However, an additional interesting result is that FS approaches do not all benefit equally from a larger selection subset size. Typically, SPEC generally obtains low results for small values of k. This is explained as SPEC is ill-suited to handle redundant features, with a tendency to select features correlated to each other. This is visible on Yale, and all the more so on the Arcene dataset, for which the results are provided in figure 6.7.

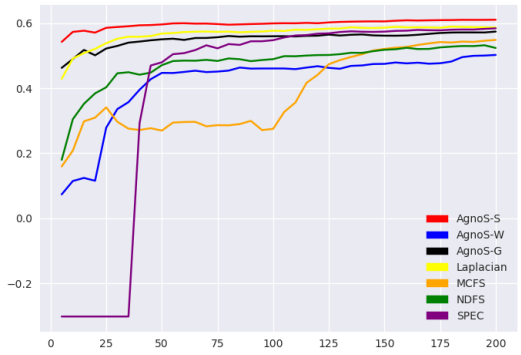

FIGURE 6.7: FIT score w.r.t. selection subset size k, on Arcene

Arcene is the only benchmark dataset for which negative FIT scores are recorded. The 5-NearestNeighbor regressor trained from SPEC is worse than a constant regressor for $k \leq 30$. However, its performance then abruptly improves as k increases to 50.

Given that the performance of AGNOS-S is already close to the observed global maximum for $k = 5$, it appears than very few variables are required to accurately predict the whole feature set on this dataset. However, the top 30 ranked features w.r.t. SPEC are likely highly redundant (even duplicates), explaining the constant FIT score for $k \in [5, \ldots, 30]$. Actually relevant features occupy the next 20 spots in the SPEC ranking, which is the reason for the sudden jump in fitting accuracy. The counterperformance of AGNOS-W is interpreted as the sparsity penalty being insufficiently strong to handle the known redundancy of the features.

7. The rank of each method changes at most 3 times with k, and the top ranked algorithms are mostly invariant across the considered range $[5, \ldots, 200]$.

Ongoing experiments will clarify this phenomenon. By contrast, this issue of inefficient selection for small subset sizes has not been encountered by AGNOS-S.

6.4.2 Sensitivity w.r.t. dimension of hidden layer d

6.4.2.1 Classification-based criterion

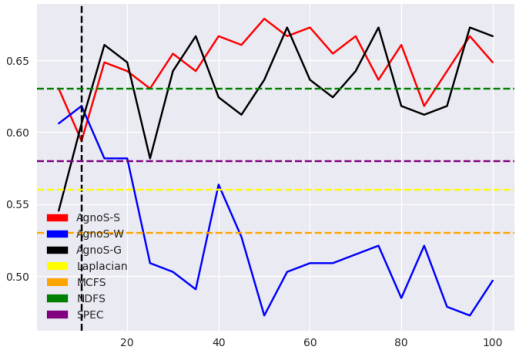

FIGURE 6.8: Prediction score of 5-NearestNeighbor classifier w.r.t. hidden layer size d, on Yale. The vertical black dotted line corresponds to the estimated intrinsic dimension. The colored horizontal dotted line correspond to the respective prediction scores of the baselines (independent of d)

Figure 6.8 depicts the classification accuracy attained by the 3 declinations of AGNOS as a function of the size d of the hidden layer of the AutoEncoder, on the Yale dataset. The performance of all three declinations of AGNOS w.r.t. this criterion appears sensitive w.r.t. d. The supervised predictive accuracy of AGNOS-W diminishes as d increases, which is attributed to the same phenomenon as in *fig.* 6.7. This variant of AGNOS eventually performs worse than all baselines. On the other hand, AGNOS-S remains a better candidate than all baselines for all d in a reasonable range (in $[10, 85]$).

6.4.2.2 Clustering-based criteria

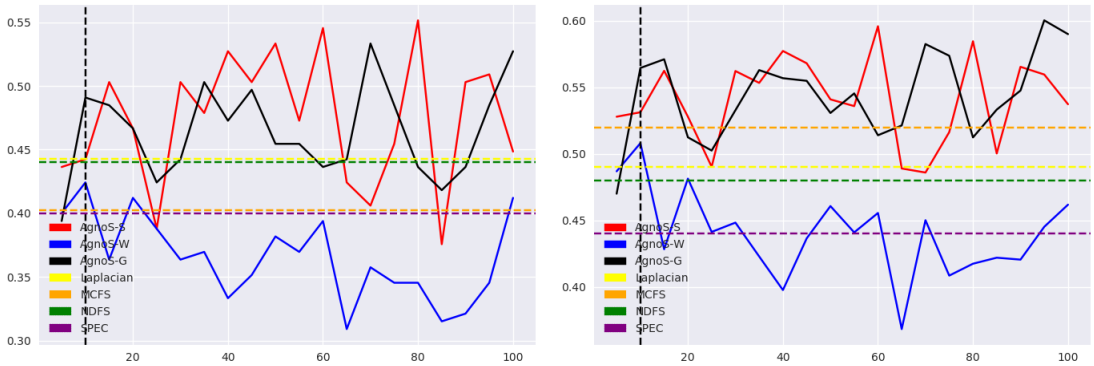

FIGURE 6.9: ACC score (left panel) and NMI score (right panel) w.r.t. hidden layer size d, on Yale. The vertical black dotted line corresponds to the estimated intrinsic dimension. The colored horizontal dotted line correspond to the respective FIT scores of the baselines (independent of d).

Figure 6.9 records the ACC and NMI scores as function of d, on Yale. The sensitivity of the results w.r.t. d is higher than for the classification criterion, with AGNOS-S generally being the best candidate and AGNOS-G a close second. However, all three declinations of AGNOS can perform worse than at least one baseline for some values of d. The brittleness of the resulting ranking confirms that these two performance criteria are ill-advised for comparing FS algorithms.

6.4.2.3 FIT criterion

Figure 6.10 depicts the FIT score attained by the 3 declinations of AGNOS as a function of the size d of the hidden layer of the AutoEncoder, on the Yale dataset. The respective performances of both AGNOS-W and AGNOS-G are s.t. the rank of both variants among the considered FS methods fluctuates with d. By contrast, the performance of AGNOS-S appears to be stable w.r.t. d, and remains higher than all baselines across the entire range $d \in [5, \ldots, 100]$. Concordant results were observed on the other benchmark datasets.

The fact that the performance of AGNOS-W decreases for $d > \widehat{ID}$ is congruent with the reasoning exposed in chapter 4 ; if the size of the hidden layer is larger than the intrinsic dimension, then the set of latent features likely contains redundancy. In turn, original features that are important to build only superfluous latent features may be wrongfully selected, negatively impacting the results.

By construction, all latent features matter for $d = \widehat{ID}$, meaning the above issue is averted and only the most relevant initial features are selected. However, this assumption does not seem to hold in practice, as highlighted by the fact that the performance of AGNOS-G is shown to increase for $d > \widehat{ID}$. In order to address

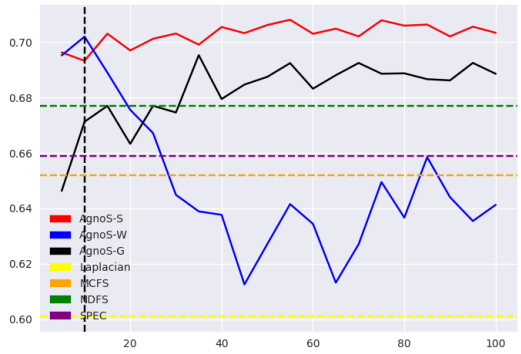

FIGURE 6.10: FIT score w.r.t. hidden layer size d, on Yale. The vertical black dotted line corresponds to the estimated intrinsic dimension. The colored horizontal dotted line correspond to the respective FIT scores of the baselines (independent of d).

this shortcoming, a potential direction for further work is to take into account the varying importance of the latent features in the selection criterion (*chap.* 7).

The low sensitivity of performance w.r.t. d means that studying the intrinsic dimension of the dataset as preprocessing is of lesser importance for AGNOS-S than for AGNOS-G and AGNOS-W. This is consequently another argument in favor of this declination of AGNOS. We interpret this robustness property in the same manner as in section 6.3 ; in AGNOS-W, the encoder weights (resp. the encoder gradients in AGNOS-G) are simultaneously affected by the reconstruction penalty and the group LASSO penalty. Therefore, the value of d directly governs the number of parameters responsible for enforcing sparsity, and in turn has a strong influence on the final selection criteria. On the other hand, sparsity is enforced in AGNOS-S by the slack variable layer, which size is unaffected by d. The quality of the selection criterion in AGNOS-S is therefore less reliant on finetuning d.

6.4.3 Sensitivity w.r.t. penalization strength λ

Figure 6.11 depicts the FIT scores of the three AGNOS variants as function of λ (represented in log scale) on Isolet .

The performance of AGNOS appear to be sensitive w.r.t. λ, with all three variants possibly being overtaken by some of the baselines. However, the suitable range for λ differs for each declination of AGNOS. For both AGNOS-S and AGNOS-W, the default value $\lambda = 1$ leads to a FIT score close to the recorded maximum. However, this default value appears to be too large for AGNOS-G, requiring $\lambda = 10^{-3}$ to reach its best performance instead. This is related to the phenomenon of *fig.* 6.7, which is under study.

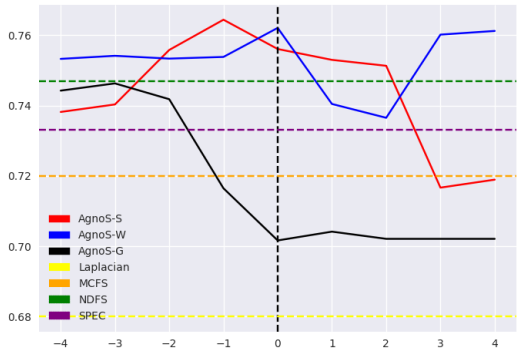

FIGURE 6.11: FIT score w.r.t. penalization strength λ (represented in log scale), on Isolet. The vertical black dotted line corresponds to the default value of $\lambda = 1$. The colored horizontal dotted line correspond to the respective FIT scores of the baselines (independent of λ). Similar curves are obtained for the other benchmark datasets, as well as for *sup.* assessment.

	arcene	Isolet	ORL	pixraw10P	ProstateGE	TOX171	warpPie10P	Yale
AGNOS-S	265	25	29	242	145	143	31	14
AGNOS-W	422	31	40	389	191	180	47	18
AGNOS-G	428	32	42	394	195	184	48	18
Laplacian	<1	<1	<1	<1	<1	<1	<1	<1
SPEC	3	9	<1	2	1	2	1	<1
MCFS	<1	2	<1	<1	<1	<1	<1	<1
NDFS	130	16	17	193	80	76	18	7

TABLE 6.14: Empirical runtimes on a single Nvidia Geforce GTX 1060 GPU, in seconds.

Main drawback of AGNOS : the computational cost

Table 6.14 contains the empirical runtimes of the baselines and the three variants of AGNOS on each dataset. AGNOS-S is shown to be between 25% and 100% slower than NDFS, and several orders of magnitude slower than Laplacian score, SPEC and NDFS. Training an Auto-Encoder with a number of parameters of the order of $D \times d$ thus appears more expensive than spectral clustering-based optimization[8]. AGNOS-W and AGNOS-G are even slower, being on average 50% slower than AGNOS-S. This is explained by the fact that the respective loss functions of AGNOS-W and AGNOS-G involve $D \times d$ parameters (*resp.* weights and gradients) compared to the D slack variables computed in AGNOS-S.

The computational effort thus constitutes the main limitation of the proposed algorithmic contribution. Consequently, a perspective for future research is to lower the complexity of the approach, either with early

8. Arguably, the use of GPU for neural computation is significally more advanced than for spectral clustering, although some announcements from Nvidia (nvi, 2017) suggest that appropriate libraries for spectral clustering with CUDA would be available soon.

stopping or recursive feature elimination (*chap.* 7).

6.5 Partial Conclusion

This experimental study has shown that AGNOS is able to consistently select a subset sufficient to recover the whole original feature set, and outperform the considered baseline methods w.r.t. both *sup.* and *unsup.* performance criteria. A remaining question is which one of the three AGNOS declinations is best suited to the dataset at hand, particularly so among AGNOS-S and AGNOS-G. As said, the comparative lesser performance of AGNOS-W is under study.

The second contribution of the chapter is to show the merits of the proposed FIT criterion, in terms of stability w.r.t. the target feature by construction and also w.r.t. the hyper-parameters of FS such as the selection subset size. This robustness property, highlights the reliability of the FIT evaluation scheme for comparing unsupervised FS algorithms.

Finally, the *unsup.* part of this empirical study (*sec.* 6.3) has also underlined the *partiality* issue inherent to *sup.* validation of *unsup.* FS, as claimed in chapter 5, the ranking of *unsup.* FS algorithms w.r.t. any *sup.* scoring function arbitrarily depending on the considered learning goal.

Chapitre 7

Perspectives and conclusion

Two different aspects of unsupervised feature selection have been explored in this thesis. On one hand, a novel unsupervised FS algorithm has been proposed. On the other hand, we have devised a new performance evaluation framework for comparing unsupervised FS techniques.

This chapter first recalls the main results and lessons learned for the unsupervised feature selection problem (section 7.1), then discusses the research perspectives opened by this work in section 7.2.

7.1 Summary of contributions

7.1.1 Unsupervised Dimensionality Reduction

As shown (*chap.* 2), the curse of dimensionality (Pestov, 1999) effectively renders the Euclidean distance ineffective to assess similarity between high-dimensional datapoints. Moreover, the efficiency of state-of-the-art *unsup.* FS approaches (*chap.* 3) is noticeably hindered, admittedly to a varying extent, when faced with features carrying redundant information. Thirdly, features are typically retained in view of a single particular learning goal (*chap.* 5), even though all original variables are potential learning goals in the *unsup.* context (LeCun, 2016).

Taking note from the above three remarks, the proposed algorithmic contribution hinges on Auto-Encoding neural networks to simultaneously suppress the need for a high-dimensional similarity metric and perform *agnostic* feature selection. In doing so, *Agnostic Feature Selection* (AGNOS, *chap.* 4) essentially bridges the gap between unsupervised feature construction and selection. Three variants of this algorithmic contribution (named AGNOS-W, AGNOS-G and AGNOS-S) have been proposed, each enhancing the AutoEncoder with a different form of structural regularization enforcing sparse selection, thereby efficiently addressing the feature redundancy issue.

7.1.2 Assessment of unsupervised Feature Selection

As seen (*chap.* 5), we claim that typical *sup.* performance indicators for *unsup.* FS lack in reliability and stability. In order to provide a stable and reliable performance indicator, we propose the methodological contribution of this thesis, the *unsup. FIT* scoring criterion (*chap.* 5).

7.1.3 Empirical evidence

A systematic study has been conducted to back the claims of the thesis. On the one hand, it is shown that the proposed AGNOS algorithm outperforms state-of-the-art *unsup.* FS methods (He et al., 2005; Zhao and Liu, 2007; Cai et al., 2010; Li et al., 2012) w.r.t. both typical *sup.* assessment metrics and the novel FIT score.

The main two lessons learned from the empirical study concern both unsupervised FS and the validation methodology :

— On the algorithmic side, AGNOS favourably compares to the most impactful *unsup.* FS techniques on representative datasets illustrating different application domains (medical, text and face image data) and an artificial dataset known to hinder independent scoring methods.

— On the methodological side, we establish that *sup.* performance indicators generally used to assess *unsup.* FS provide brittle results. The exploitation of the intrinsic dimensionality of a dataset can also be considered a worthy ingredient for *unsup.* FS in the data compression perspective.

These findings were the subject of an accepted paper at the 2019 edition of the European Conference on Machine Learning (ECML) [1].

7.1.4 Strengths and weaknesses

Pros The main benefits of our contribution are :

— Escaping the curse of dimensionality by avoiding usage of a high-dimensional pairwise similarity metric.

— Capturing much more information than spectral clustering-based methods : the objective is to recover D original features rather than a single pseudo-label variable. As a result, the selected subset is relevant w.r.t. *any* learning goal.

— Efficiently handling redundant feature sets, thanks to the sparsity-enforcing term in the Auto-Encoder loss.

Cons Nevertheless, AGNOS currently suffers from a sizeable drawback : its empirical time complexity is shown to be larger by at least a factor two, and often an order of magnitude, than state-of-the-art *unsup.* FS methods (*chap.* 6). Lowering the computational cost of the approach is our first perspective for future research (*sec.* 7.2).

7.2 Towards more robust and computationally efficient agnostic feature selection

This work opens three perspectives for further research.

1. *Agnostic Feature Selection*, G. Doquet and M. Sebag, ECML PKDD 2019

7.2.1 Computational cost

A short-term research perspective is to reduce the computational cost of AGNOS. A first option is to transform AGNOS from a filter-based approach to an embedded online selection method, taking inspiration from Guyon et al. (2002). One possible way of iteratively eliminating the original features least contributing towards learning the constructed features in the latent AutoEncoder data representation during network training, is to set the associated slack variables to 0 in AGNOS-S. Given this modification, the dimensionality of the input continuously decreases, hereby greatly reducing computational cost both in terms of time and space complexity, and allowing AGNOS to better scale to large real-life datasets. Such an iterative process can support an automatic stopping criterion in the approach, s.t. the number of features to ultimately retain is determined on the fly, as is already implemented in supervised neural network-based approaches such as Drop-Out-One (Ye and Sun, 2018).

A second option is to use an early stopping of the Auto-Encoder, e.g. when the feature ranks as computed from the slack variables and/or the weights or gradients have not changed for some consecutive epochs : indeed, a perfect reconstruction accuracy is a means rather than an end for the AE learning.

7.2.2 Probabilistic AGNOS

A longer-term perspective consists of replacing the deterministic AutoEncoder relied upon by AGNOS with a Variational AutoEncoder (Kingma and Welling, 2013) (appendix .1). Given this modified neural architecture, original features would be selected w.r.t. their usefulness for generating realistic new samples, rather than reconstructing the existing datapoints.

The goal of this extension is to provide a more robust feature selection approach in the case where the application domain contains very few samples comparatively to the number of features, e.g. in DNA-based bioinformatics research.

7.2.3 Better exploiting the latent features

As said, AGNOS relies on the implicit assumption that all constructed features are equally important for reconstructing the original data (*chap.* 4). This assumption is unlikely to hold in practice, as underlined by the lower empirical performance (*chap.* 6) of AGNOS-W and AGNOS-G (where feature importance is derived from one constructed feature) comparatively to AGNOS-S (where feature importance simultaneously involves all constructed features).

In the short term, the extension of AGNOS-G to consider the gradients from $\widehat{\boldsymbol{f}_i}$ w.r.t. $\boldsymbol{f_i}$ ($\partial\widehat{\boldsymbol{f}_i}$) is a way to seamlessly handle the importance of the latent variables. In a medium term, the importance of the latent variables ϕ_j w.r.t. $\widehat{\boldsymbol{f}_i}$ can be used to weight the importance of the $\boldsymbol{f_i}$s.

7.2.4 Causal discovery

Lastly, a long-term perspective is to explore the link between the proposed unsupervised feature selection paradigm and the neighboring field of causal inference (Pearl, 2009). Causal feature selection (Guyon and

Aliferis, 2007; Peters et al., 2017) has insofar and to the best of our knowledge only been considered in the supervised context. However, the central motivation behind AGNOS of selecting features sufficient to recover the whole feature set is strongly reminiscent of finding a minimal functional causal model (Goudet et al., 2018) explaining all variables. Bridging the gap between causal discovery and unsupervised feature selection for the purpose of interpretability is therefore an especially interesting prospect in view of *Fair, Transparent* and *Accountable* learning (*chap.* 1).

Bibliographie

(2017). Cluster analysis. https ://developer.nvidia.com/discover/cluster-analysis.

(2018). The scikit-feature project. http://featureselection.asu.edu/index.php.

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd : An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11) :4311–4322.

Ando, R. K. and Zhang, T. (2005a). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov) :1817–1853.

Ando, R. K. and Zhang, T. (2005b). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov) :1817–1853.

Bach, F. R. (2008). Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9(Jun) :1179–1225.

Balasubramanian, M. and Schwartz, E. L. (2002). The isomap algorithm and topological stability. *Science*, 295(5552) :7.

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. *In Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49.

Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis : Learning from examples without local minima. *Neural networks*, 2(1) :53–58.

Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. (2005). Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(Sep) :1345–1382.

Bauer, K. W., Alsing, S. G., and Greene, K. A. (2000). Feature screening using signal-to-noise ratios. *Neurocomputing*, 31 :29–44.

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. *In Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning : A review and new perspectives. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 35(8) :1798–1828.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1987). Occam's razor. *Information processing letters*, 24(6) :377–380.

Borg, I. and Groenen, P. (2003). Modern multidimensional scaling : Theory and applications. *Journal of Educational Measurement*, 40(3) :277–280.

Boureau, Y. and LeCun, Y. (2008). Sparse feature learning for deep belief networks. *In Advances in neural information processing systems*, pages 1185–1192.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1994). Signature verification using a siamese time delay neural network. *In Advances in neural information processing systems*, pages 737–744.

Bronsteina, A. M., Bronstein, M. M., and Kimmel, R. (2006). Generalized multidimensional scaling : a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5) :1168–1172.

Cai, D., Zhang, C., and He, X. (2010). Unsupervised feature selection for multi-cluster data. *KDD*.

Camastra, F. and Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on pattern analysis and machine intelligence*, 24(10) :1404–1407.

Campadelli, P., Casiraghi, E., Ceruti, C., and Rozza, A. (2015). Intrinsic dimension estimation : Relevant techniques and a benchmark framework. *Mathematical Problems in Engineering*.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). *Robust principal component analysis ?*, 58(3).

Chang, X., Nie, F., Yang, Y., and Huang, H. (2014). A convex formulation for semi-supervised multi-label feature selection. *In Twenty-eighth AAAI conference on artificial intelligence*.

Chen, C. H. (2015). *Handbook of pattern recognition and computer vision*. World Scientific.

Chen, J., Stern, M., Wainwright, M. J., and Jordan, M. I. (2017). Kernel feature selection via conditional covariance minimization. *In Advances in Neural Information Processing Systems*, pages 6946–6955.

Chávez, E., Navarro, G., Baeza-Yates, R., and Marroquín, J. L. (2001). Searching in metric spaces. *ACM computing surveys (CSUR)*, 33(3) :273–321.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20 :273–297.

Costa, J. and Hero, A. O. (2004). Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. on Signal Processing*, 52(8) :2210–2221.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley Sons.

Crowder, J. A. and Carbone, J. N. (2011). Occam learning through pattern discovery : Computational mechanics in ai systems. *In Proceedings on the International Conference on Artificial Intelligence (ICAI)*.

De, R. K., Pal, N. R., and Pal, S. K. (1997). Feature analysis : neural network and fuzzy set theoretic approaches. *Pattern Recognition*, 30(10) :1579–1590.

Deepmind, G. (2019). Alphago. https ://deepmind.com/research/alphago/.

Devroye, L. (1986). *Sample-based non-uniform random variate generation*. Springer-Verlag.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv :1702.08608*.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). Pattern classification. *Wiley-Interscience*, 2.

Díaz-Uriarte, R. and Andres, S. A. D. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1).

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3) :211–218.

Facco, E., d'Errico, M., Rodriguez, A., and Laio, A. (2017). Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Nature*, 7(1).

Feng, J. and Simon, N. (2017). Sparse-input neural networks for high-dimensional nonparametric regression and classication. *arXiv :1711.07592*.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2) :179–188.

FTA (2018). Fair, transparent and accountable learning. https ://www.partnershiponai.org/fta-wg-launch/.

Gates, A. J., Wood, I. B., Hetrick, W. P., and Ahn, Y. Y. (2018). On comparing clusterings : an element-centric framework unifies overlaps and hierarchy. *arXiv :1706.06136*.

Gaudel, R. and Sebag, M. (2010). Feature selection as a one-player game.

Ghassabeh, Y. A., Rudzicz, F., and Moghaddam, H. A. (2015). Fast incremental lda feature extraction. *Pattern Recognition*, 48(6) :1999–2012.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *International conference on Artificial Intelligence and Statistics*, pages 249–256.

Goldberger, A. S. (1964). Classical linear regression. *Econometric Theory*, page 156–212.

Golu, G. H. and Reinsch, C. (1971). Singular value decomposition and least squares solutions. *In Linear Algebra*, pages 134–151.

Gorban, A. N., Golubkov, A., Grechuk, B., Mirkes, E. M., and Tyukin, I. Y. (2018). Correction of ai systems by linear discriminants : Probabilistic foundations. *Information Sciences*, 466 :303–322.

Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., and Sebag, M. (2018). Learning functional causal models with generative neural networks. *In Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 39–80.

Gu, Q., Li, Z., and Han, J. (2012). Generalized fisher score for feature selection. *arXiv preprint arXiv :1202.3725*.

Guyon, I. and Aliferis, C. (2007). Causal feature selection. In *Computational methods of feature selection*. Chapman and Hall/CRC.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3) :389–422.

Hartigann, J. A. and Wong, M. A. (1979). Algorithm as 136 : A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1) :100–108.

Haykin, S. (1994). *Neural networks : a comprehensive foundation*. Prentice Hall PTR.

He, X., Cai, D., and Niyogi, P. (2005). Laplacian score for feature selection. *Advances in Neural Information Processing Systems*.

Hinton, G. E., Srivastava, N., Krizhevsky, A. andSutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02) :107–116.

Hofmann, T., Scholkopf, B., and Smola, A. J. (2008). *Kernel Methods in Machine Learning*.

Huang, J., Y. Cai, Y., and Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters*, 28(13) :1825–1844.

Ivanoff, S., .Picard, F., and Rivoirard, V. (2016). Adaptive lasso and group-lasso for functional poisson regression. *The Journal of Machine Learning Research*, 17(1) :1903–1948.

Jenatton, R., Audibert, J., and Bach, F. (2011). Structured variable selection with sparsity-inducing norms. *JMLR 12*, page 2777–2824.

Jianya, Y. Y. G. (1999). An efficient implementation of shortest path algorithm based on dijkstra algorithm. *Journal of Wuhan Technical University of Surveying and Mapping (Wtusm)*, 3(004).

Kantardzic, M. (2003). *Data Reduction*. Wiley Online Library.

Kearns, M. J. and Vazirani, U. V. (1994). *An introduction to computational learning theory*.

Kim, H. and Mnih, A. (2018). Disentangling by factorising. *arXiv :1802.05983*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv :1312.6114*.

Kira, K. and Rendell, L. A. (1992). The feature selection problem : Traditional methods and a new algorithm. *AAAI*, 2 :129–134.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *In IJCAI*, 14 :1137–1145.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2) :273–324.

Kononenko, I. (1994). Estimating attributes : analysis and extensions of relief. *In European conference on machine learning*, pages 171–182.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, pages 1097–1105.

Kvalseth, T. O. (1987). Entropy and correlation : Some comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3) :517–519.

Kégl, B. (2003). Intrinsic dimension estimation using packing numbers. *In Advances in neural information processing systems*, pages 697–704.

LeCun, Y. (2016). The next frontier in AI : Unsupervised learning. https://www.youtube.com/watch?v=IbjF5VjniVE.

Leray, P. and Gallinari, P. (1999). Feature selection with neural networks. *Behaviormetrika*, 26(1) :145–166.

Levina, E. and Bickel, P. J. (2005). Maximum likelihood estimation of intrinsic dimension. *In Advances in neural information processing systems*, pages 777–784.

Li, B., Wang, C., and Huang, D. S. (2009). Supervised feature extraction based on orthogonal discriminant projection. *Neurocomputing*, 73(1-3) :191–196.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2018a). Feature selection : A data perspective. *ACM Computing Surveys (CSUR)*, 50(6).

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2018b). Feature selection : A data perspective. *ACM Computing Surveys (CSUR)*, 50(6), 94.

Li, J. and Liu, J. (2017). Challenges of feature selection for big data analytics. *IEEE Intelligent Systems*, 32 :9–15.

Li, J., Tang, J., and Liu, H. (2017). Reconstruction-based unsupervised feature selection : an embedded approach. *In Proceedings of the 26th International Joint Conference on Artificial Intelligence*.

Li, Y., Chen, C. Y., and Wasserman, W. W. (2016). Deep feature selection : theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5) :322–336.

Li, Z., Liu, J., Yang, Y., Zhou, X., and Lu, H. (2014). Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 26(9) :2138–2150.

Li, Z., Yang, Y., Liu, Y., Zhou, X., and Lu, H. (2012). Unsupervised feature selection using non-negative spectral analysis. *AAAI*.

Liu, B., Yu, X., Zhang, P., Yu, A., Fu, Q., and Wei, X. (2018). Supervised deep feature extraction for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4) :1909–1921.

Liu, J. and Ye, J. (2010). Moreau-yosida regularization for grouped tree structure learning. *In NIPS*, page 1459–1467.

Liu, Z., Guo, Z., and Tan, M. (2008). Constructing tumor progression pathways and biomarker discovery with fuzzy kernel kmeans and dna methylation data. *Cancer informatics*, 6.

Ma, Z., Nie, F., Yang, Y., Uijlings, J. R. R., and Sebe, N. (2012). Web image annotation via subspace-sparsity collaborated feature selection. *IEEE Trans. Multimedia*, 14(4) :1021–1030.

Mandelbrot, B. (1967). How long is the coast of britain ? statistical self-similarity and fractional dimension. *Science*, 156(3775) :636–638.

Mandelbrot, B. B. (1983). The fractal geometry of nature. *New York : WH freeman*, 173 :51.

Martinez, A. M. and Zhu, M. (2005). Where are linear feature extraction methods applicable ? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12) :1934–1944.

McCuen, R. H., Knight, Z., and Cutter, A. G. (2006). Evaluation of the nash–sutcliffe efficiency index. *Journal of Hydrologic Engineering*, 11(6) :597–602.

Meier, L., Geer, S. V. D., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 70(1) :53–71.

Meilă, M. (2003). Comparing clusterings by the variation of information. *In Learning theory and kernel machines*, pages 173–187.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K. R. (1999). Fisher discriminant analysis with kernels. *In Neural networks for signal processing IX : Proceedings of the 1999 IEEE signal processing society workshop*, pages 41–48.

Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11) :2227–2240.

Ng, A. (2001). On spectral clustering : Analysis and an algorithm. *Advances in Neural Information Processing Systems*.

Nie, F., Zhu, W., and Li, X. (2016). Unsupervised feature selection with structured graph optimization. *In AAAI*, pages 1302–1308.

Olshausen, B. and Field, D. (1997). Sparse coding with an overcomplete basis set : a strategy employed by v1 ? *Vision Research*, 37 :3311–3325.

O'neil, C. (2016). *Weapons of math destruction : How big data increases inequality and threatens democracy*.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *In International conference on machine learning*, pages 1310–1318.

Pearl, J. (2009). Causal inference in statistics : An overview. *Statistics surveys*, 3 :96–146.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2 (11) :559–572.

Pestov, V. (1999). On the geometry of similarity search : dimensionality curse and concentration of measure. *arXiv preprint cs/9901004*.

Pestov, V. (2007). Intrinsic dimension of a dataset : what properties does one expect ? *International Joint Conference on Neural Networks*, pages 2959–2964.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference : foundations and learning algorithms*. MIT press.

Pettis, K. W., Bailey, T. A., Jain, A. K., and Dubes, R. C. (1979). An intrinsic dimensionality estimator from near-neighbor information. *IEEE Trans. on PAMI*, 1 :25–37.

Poultney, C., Chopra, S., and LeCun, Y. (2007). Efficient learning of sparse representations with an energy-based model. *In Advances in neural information processing systems*, pages 1137–1144.

Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., and Cari, L. (2016). Variational autoencoder for deep learning of images, labels and captions. *In Advances in neural information processing systems*, pages 2352–2360.

Qian, M. and Zhai, C. (2013). Robust unsupervised feature selection. *In IJCAI*, pages 1621–1627.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011). Contractive auto-encoders : Explicit invariance during feature extraction. *In Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 833–840.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, New Series*, 290 :2323–2326.

Roy, D., Murty, K. S. R., and Mohan, C. K. (2015). Feature selection using deep neural networks. *International Joint Conference on Neural Networks (IJCNN)*.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv :1609.04747*.

Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3) :660–674.

Saul, L. K. and Roweis, S. T. (2003). Think globally, fit locally : unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun) :119–155.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT press.

Setiono, R. and Liu, H. (1997). Neural-network feature selector. *IEEE transactions on neural networks*, 8(3) :654–662.

Shi, J. and Malik, J. (1997). Normalized cuts and image segmentation. *CVPR*.

Shi, L., Du, L., and Shen, Y. D. (2014). Robust spectral learning for unsupervised feature selection. *In Data Mining (ICDM), IEEE*, pages 977–982.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2) :231–245.

Simpson, E. H. (1951). The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society*, 13 :238–241.

Skeem, J. L., L., J., and Lowenkamp, C. T. (2016). Risk, race, and recidivism : predictive bias and disparate impact. *Criminology*, 54(4) :680–712.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2014). Dropout : a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1) :1929–1958.

Steppe, J. and Bauer, K. W. (1996). Improved feature screening in feedforward neural networks. *Neurocomputing*, 13 :47–58.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec) :583–617.

Tenenbaum, J. B., Silva, V. D., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500) :2319–2323.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B (Methodological)*, pages 267–288.

Torgerson, W. S. (1958). Theory and methods of scaling.

Trunk, G. V. (1976). Stastical estimation of the intrinsic dimensionality of a noisy signal collection. *IEEE Transactions on Computers*, 100(2) :165–171.

van der Maaten, L. J. P., Postma, E. O., and van den Herik, H. J. (2008). Dimensionality reduction : A comparative review.

Varga, D., Csiszárik, A., and Zombori, Z. (2017). Gradient regularization improves accuracy of discriminative models. *arXiv :1712.09936*.

Verikas, A. and Bacauskiene, M. (2002). Feature selection with neural networks. *Pattern Recognition Letters*, 23(11) :1323–1335.

Verveer, P. and Duin, R. (1995). An evaluation of intrinsic dimensionality estimators. *IEEE Trans. on PAMI*, 17(1) :81–86.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. *In Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. A. (2010). Stacked denoising autoencoders : Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec) :3371–3408.

Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison. *Proceedings of the 26th Annual International Conference on Machine Learning*.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison : Variants, properties, normalization and correction for chance. *JMLR*.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4) :395–416.

Wang, D., Nie, F., and Huang, H. (2014). Unsupervised feature selection via unified trace ratio formulation and k-means clustering (track). *In Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 306–321.

Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv :1505.00853*.

Yao, Y. Y. (2003). Information-theoretic measures for knowledge discovery and data mining. *In Entropy measures, maximum entropy principle and emerging applications*, pages 115–136.

Ye, M. and Sun, Y. (2018). Variable selection via penalized neural network : a drop-out-one loss approach. *In International Conference on Machine Learning*, pages 5616–5625.

Yu, S. and Shi, J. (2003). Multiclass spectral clustering. *IEEE*.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68(1) :49–67.

Zhang, M. L. and Zhou, Z. H. (2007). Ml-knn : A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7) :2038–2048.

Zhang, M. L. and Zhou, Z. H. (2013). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8) :1819–1837.

Zhao, H., Sun, S., Jing, Z., and Yang, J. (2006a). Local structure based supervised feature extraction. *Pattern Recognition*, 39(8) :1546–1550.

Zhao, H., Sun, S., Jing, Z., and Yang, J. (2006b). Local structure based supervised feature extraction. *Pattern Recognition*, 39(8) :1546–1550.

Zhao, Z. and Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. *ICML*.

Zhao, Z. and Liu, H. (2008). Multi-source feature selection via geometry-dependent covariance analysis. *In New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, pages 36–47.

Zhao, Z., Wang, L., Liu, H., and Ye, J. (2013). On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 25(3) :619–632.

Zurada, J. M., Malinowski, A., and Usui, S. (1997). Pertubation method for deleting redundant inputs of perceptron networks. *Neurocomputing*, 14 :177–193.

.1 Appendix A : Variational Auto-Encoders

Variational AutoEncoders (VAE) (Kingma and Welling, 2013) rapidly gained popularity over the past five years as a *data augmentation* method (Pu et al., 2016). The mathematical basis of VAEs is actually quite remote from classical AutoEncoders. The name of the approach is therefore slightly ambiguous, as the resulting network merely *resembles* an AutoEncoder.

The core idea behind VAEs is to interpret the d constructed features as being the parameters of d probability distribution functions. The most common choice corresponds to a multivariate Gaussian distribution, so that each constructed feature in Z_d^* consists of a tuple (μ, σ) reflecting the mean and standard deviation of a scalar Gaussian.

During training, latent variables are sampled from their respective distributions, and the decoder part of the neural network is tasked with recreating the original data X [2] from the stochastic samples. The success of this reconstruction hinges on the crucial observation that a set of d Gaussian random variables can be mapped to an arbitrarily close approximation of *any* d-dimensional distribution (including that of X, provided the manifold assumption from chapter 2 holds), provided a sufficiently complex function (Devroye, 1986). An illustration of this result is provided in figure 1.

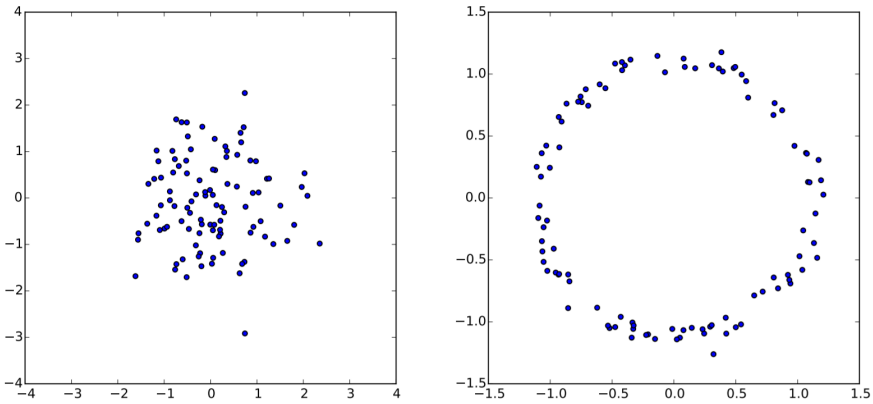

FIGURE 1: The 2D random variable z on the left panel can be mapped to a ring-shape distribution X through the function $\psi(z) = z/10 + z/||z||$

Provided with this setup, one could sample from the model without any input. However, obtaining a satisfactory reconstruction of X this way is unaffordable in terms of time complexity, as highlighted by figure 2 and the following discussion.

One would expect the model generating the digit in panel (b) to be deemed mediocre, given the apparent dissimilarity with the original MNIST datapoint of panel (a). On the other hand, the reconstruction of panel (c) (identical to (a) but shifted down and to the right by one pixel) is perceived to be better. Unfortunately, (b) is much closer to (a) than (c) is, w.r.t. the MSE loss. Therefore, one would need to obtain a model *significantly*

2. More precisely, the decoder is tasked with *generating* datapoints that *look like* those of X, which is why the approach is used to perform Data Augmentation. The *decoder* designation is therefore only used to draw the parallel between VAEs and traditional AEs ; a less ambiguous name for this network component would be *generator*.

(a) (b) (c)

FIGURE 2: The reconstructed digit of panel (c) is perceptually much closer to the original MNIST digit of panel (a) than the sample in panel (b). However, the MSE loss provides the reverse conclusion, that (b) is the far better approximation

better than (c) in order to discard results such as (b). This is consequently likely to require an unreasonable amount of samples.

In order to accelerate the sampling procedure, an encoder ϕ mapping X to the constructed features $(\mu(X), \sigma(X))$ is added to the pipeline, s.t. the distribution parameters are learned through backpropagation of the reconstruction error[3] (fig. 3).

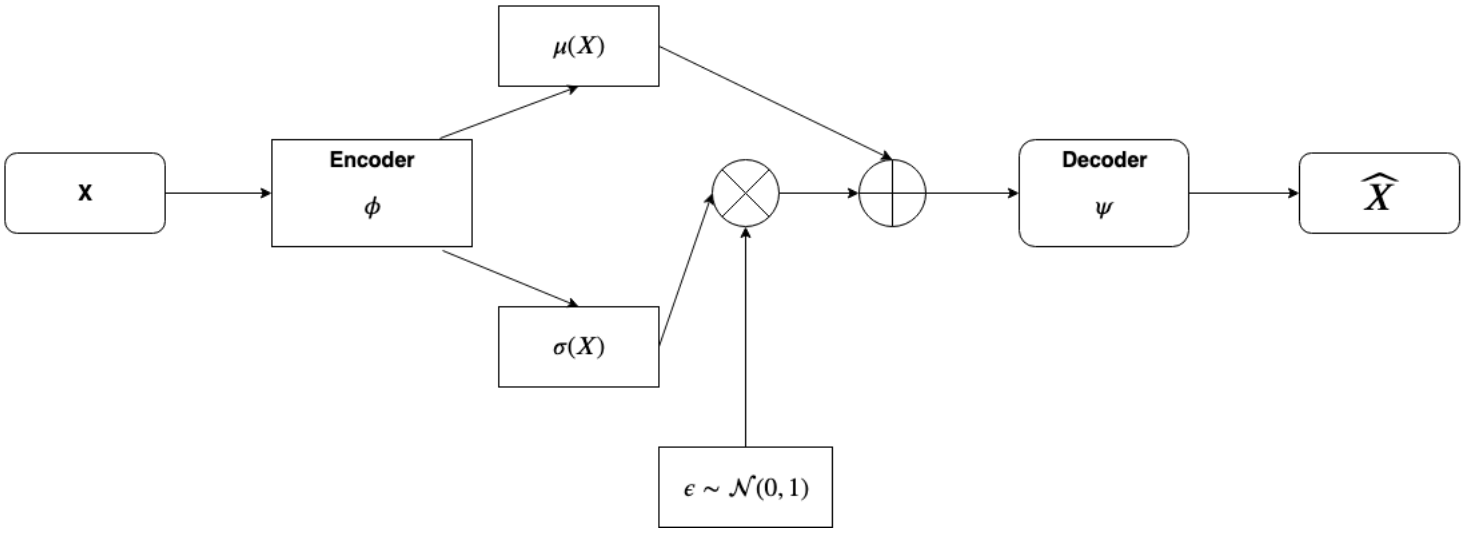

FIGURE 3: Illustration of the final VAE pipeline

3. Arguably, an alternate solution would be to design a similarity metric better suited to depict likeness of images than the MSE. However, not only are such metrics challenging to define in domains such as computer vision, but also hardly interpretable without label information indicating which images are similar, as is the case in unsupervised FC

The sole purpose of the encoder part is thus to ensure that training is affordable in terms of time complexity. After training is complete, ϕ is discarded and new samples can be generated using only the constructed features and ψ, therefore achieving data augmentation. This constitutes a significant methological difference with classical AutoEncoders : the encoder is merely a convenient tool rather than the end goal of learning. Moreover, the constructed features do not necessarily contain any information related to X, which is why we consider VAEs to be remote from other FC methods.