UNIVERSITÉ CATHOLIQUE DE LOUVAIN
ÉCOLE POLYTECHNIQUE DE LOUVAIN
ICTEAM - ELEN
MACHINE LEARNING GROUP

Non-Parametric Feature Selection for Machine Learning in Complex Settings

Gauthier DOQUIRE

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION DU GRADE DE DOCTEUR EN SCIENCES DE L'INGÉNIEUR

Composition du Jury:

Prof. **Michel Verleysen**, Advisor, UCL/SST/ICTEAM/ELEN
Prof. **Christophe De Vleeschouwer**, President, UCL/SST/ICTEAM/ELEN
Prof. **Pierre Dupont**, UCL/SST/ICTEAM/INGI
Prof. **Fabrice Rossi**, Université Paris 1 Panthéon-Sorbonne, France
Prof. **Barbara Hammer**, Universität Bielefeld, Germany
Dr. **Ivan Saeys**, Universiteit Gent, Belgium

September 2013

Contents

Remerciements

*A*u moment de ponctuer par ce texte quatre années de travail, beaucoup de personnes méritent, pour diverses raisons, mes plus chaleureux et sincères remerciements.

*S*ans l'aide et les conseils de mon promoteur, le professeur Michel Verleysen, cette thèse n'aurait jamais pu voir le jour. C'est en effet lui qui m'a proposé de travailler sur la sélection de variables et qui a tout mis en oeuvre pour que je puisse décrocher la bourse ayant permis de financer mes recherches. Ses idées, remarques et commentaires m'ont sans cesse stimulé, m'ont permis de progresser et d'améliorer grandement la qualité de mes travaux, tant d'un point de vue technique que concernant la rédaction, domaine dans lequel il excelle. Michel a également la particularité d'être la personne la plus occupée que je connaisse, tout en étant probablement aussi une des plus disponibles, parvenant toujours à ménager le temps nécessaire à une discussion. Probablement d'ailleurs, la relecture de certains papiers ou chapitres de cette thèse lui auront-ils coûté quelques nuits de sommeil; j'espère qu'il ne m'en voudra pas pour cela. Travailler aux côtés de Michel fut donc au final un réel plaisir et une très grande chance, tant scientifiquement qu'humainement. Je me souviendrai aussi longtemps de l'ambiance si conviviale, presque familiale, que Michel réussit à faire régner lors de l'ESANN. J'ai passé à Brugges d'excellents moments avec toute l'équipe du MLG.

*M*erci également aux professeurs Pierre Dupont et Fabrice Rossi qui ont composé le comité d'accompagnement de cette thèse. Jamais à court d'idées ou de références à consulter, ils furent tout deux d'excellent conseil et leur rigueur scientifique fut très enrichissante pour moi. Je remercie par ailleurs Barbara Hammer, Yvan Saeys et Christophe De Vleeschouwer d'avoir accepté de faire partie de mon jury. Leurs conseils et remarques m'ont été d'une aide précieuse. Il me faut également exprimer ma gratitude au personnel technique et adminstratif de l'ex-DICE, pour une assistance aussi sympathique qu'efficace au long de ces quatre années.

*O*n peut sans mentir affirmer que le MLG est une équipe au sein de laquelle il est très agréable d'évoluer et dans laquelle je me suis immédiatement senti à ma place. Le mérite en revient avant tout à mes collègues et anciens collègues. Merci en particulier à Emilie, Arnaud, Benoît et Gaël, avec qui partager l'un des trois bureaux que j'ai

occupés au cours de ma thèse fut un réel plaisir. Merci également aux autres membres du MLG pour les bons moments et les fructueuses discussions que nous avons pu partager, que ce soit lors de nos réunions, d'un barbecue chez Michel ou autour d'un verre durant l'ESANN. J'ai également eu la chance de collaborer plus directement avec deux de mes collègues. Je remercie donc Gaël, pour m'avoir donné la possibilité de réaliser mes premières publications en me faisant partager ses recherches sur la classification de battements cardiaques. Je remercie également Benoît pour l'enthousiasme dont il a fait preuve lorsque j'ai pour la première fois abordé avec lui le sujet de la sélection de variables. Ce fut à mon sens le début d'une collaboration particulièrement fructueuse à laquelle ses nombreuses idées et sa passion pour le Machine Learning ont grandement contribué.

La recherche est une activité pouvant rapidement se révéler très prenante. Merci donc à mes amis pour leur soutien et pour les précieux moments passés en leur compagnie, qui m'ont souvent permis de lâcher prise et de remettre un peu d'ordre dans mes priorités.

Par dessus tout, je voudrais exprimer à ma famille ma plus profonde gratitude. A mes parents en premier lieu, pour leur soutien indéfectible tout au long des 26 dernières années. Ils ont tout mis en oeuvre pour me permettre de m'épanouir au mieux et je leur dois beaucoup de ce que je suis aujourd'hui. Mon principal regret est probablement de ne pas avoir totalement réussi à les convaincre que oui, la recherche est bel et bien un *vrai métier*. Merci également à mes (beaux-)frères et (belles-)soeurs pour tout ce que nous partageons, et pour avoir parfois pris le temps de s'intéresser (ou de me le faire croire) à mes travaux. Merci enfin à mon épouse, Julie, pour rendre chaque jour ma vie aussi belle. Merci d'avoir été à mes côtés pendant ces quatre ans, pour partager avec moi des moments de joie et d'autres plus difficiles. Notre mariage est aujourd'hui ce dont je suis le plus fier.

Ce travail a été financé par une bourse FRIA délivrée par le FNRS.

Foreword

During my thesis, I had the chance to work on various aspects of feature selection, both experimental and theoretical, and to collaborate with different researchers. An important amount of the work and the publications discussed in this thesis are the results of fruitful exchanges. The aim of this foreword is to give more information about these collaborations, but also to clearly expose which are the original contributions I am responsible for. For each publication, the ideas and comments of my advisor Michel Verleysen have constituted a priceless help.

First, a number of works have been performed on my own. In [54], I experimentally compare different multivariate mutual information estimators, in the context of feature selection. In [52] and [53], I propose two approaches to feature selection for mixed datasets, with both categorical and continuous features. I presented first results on feature selection for missing data in [58] at ESANN 2011, and this work was further selected to be extended for a special issue of the Neurocomputing journal [55]. The presentation of [58] raised the interest of Amaury Lendasse and Emil Eirola, two members of the Environmental and Industrial Machine Learning Group from the Aalto University in Finland. Emil set himself to the derivation of an efficient way to estimate distances between samples for datasets with missing values [61]. I contributed by improving the experimental part, listing related works and helping organizing the text. I also tackled the problem of feature selection for datasets where the sample values are uncertain or noisy. I proposed, for different noise models, an analytical expression of the mutual information in the case of classification problems [50]. Furthemore, I have studied the problem of feature selection fur multi-label datasets, i.e. datasets where each sample can be associated to more than one class label. For continuous datasets, I suggested to use a problem transformation method combined with a mutual information-based feature selection scheme [49]. These results were further analysed and a strategy to fix the pruning parameter for the problem transformation was proposed in [48]. For discrete datasets, I obtained promising results using the Hilbert-Schmidt independance criterion (HSIC), which are waiting to be published. My works eventually focused on feature selection for semi-supervised

regression problems. I proposed in [51] and [56] an approach based on the graph Laplacian. I also obtained interesting results using HSIC to derive multivariate feature selection algorithms for semi-supervised and weakly-supervised problems which have to be published.

I collaborated with Gaël de Lannoy on the problem of feature selection for heart beat classification. Gaël provided the dataset and preprocessed the data while I suggested the methodology and took care of the experiments. This work resulted in a conference paper [46] selected for an extension in a journal [47].

Eventually, the person I had the most succesfull collaboration with is Benoît Frénay. We had lots of rich (and long) discussions in the last two years of our respective theses, making it difficult to exactly determine each one's contribution to our common works. Knowing Benoît was working on label-noise, I asked him about the oportunity to achieve feature selection with mutual information in this context using generative models, as they can provide an estimate of conditional probabilitites which are required to estimate the mutual information. Benoît coordinated the theoretical developments and took care of the implementation. I coordinated the experimental part and helped in the developments with my expertise of the mutual information. The results can be found in [72]. Benoît and I then came to discuss about feature selection for imbalanced classification and quickly reached the conclusion that in the literature, mutual information is often seen as a proxy for the classification risk. We have shown in [7] and [73] that it is not always true even if mutual information can be considered as a good criterion in practice. Benoît and I both performed a deep literature review. The two aforementioned papers being mainly experimental, we carefully determined which simulations should be run; Benoît then implemented them. We both contributed to the analysis of the obtained results. Following an idea of Dr. Amaury Lendasse, we also decided to study the adequacy of mutual information in regression problems, with similar conclusions as for classification problems. The conclusions in [74] are the results of many discussions and exchanges; Benoît implemented the proposed experiments. In [80], we show that, using a nearest-neighbour based mutual information estimator, the classification risk can be directly and reliably estimated. I took care of the experiments after Benoît implemented the risk estimator.

List of Figures

Introduction

Nowadays, in almost any area, the possibilities of acquiring and storing data are continuously increasing. As an example, IBM estimates that about 90 % of the data existing in the world today has been created during the last two years, at a rate of about 2.5 quintillon bites per day [180]. Simultaneously, the per-capita data storage capacity has been reported to have roughly doubled every 40 months since the beginning of the eighties [98].

Lacking of prior information about the relevance of the collected features, practionners often consider the whole available information and have to deal with databases containing a huge number of features; many of them are generally redundant or irrelevant for a given prediction problem. To illustrate this, in near-infrared spectra analysis, the rate of absorbance of objects is traditionally measured for tens or hundreds of different wavelengths [150]. In text classification or spam detection, features often consist in a value indicating the (frequence of) occuring of a collection of words. Again, datasets with hundreds or thousands of dimensions are generally built this way. The task of feature selection, sometimes referred to as variable or characteristic selection, consists in detecting a small set of features which are relevant or informative for the considered problem. Of course the notion of relevance can vary from one situation to another, and mainly results from the many possible motivations one has to perform feature selection. These motivations are described in the rest of this section.

The objectives and benefits of feature selection are numerous in practice. First, as can be easily deduced, removing irrelevant variables will undoubtely positively affect the performances of many learning algorithms, which could be harmed by the presence of too many *noisy* or redundant features.

Then, learning prediction models in high-dimensional spaces is a particularly hard task, due to many indesirable and counter-intuitive effects arising in this context; the most well-known one is the curse of dimensionality [15], basically stating that the number of points needed to partition a space at a given resolution increases exponentially with the dimension of the space. Other problems of working in high-dimensional spaces include the empty space phenomenon (the volume of a hypersphere rapidly

decreases towards 0 when the dimensionality increases), the concentration of the distances and the spreading of Gaussian functions [176]. More generally, learning in high-dimensional spaces traditionnaly requires a huge amount of samples and is prone to overfitting; feature selection is one way to circumvent such problems, by sharply reducing the dimesionality of the studied datset. It is important to note that learning in high-dimensional spaces is mainly an issue for the very popular distance-based models. On the contrary, other learning paradigms perform better when given as much features as possible, as they are considered as additional information sources. This for instance the case in statistical physics-based learning where, in an extreme situation, an infinite number of features make learning deterministic. We do not discuss this point in further details and refer the interested reader to [64].

Feature selection also helps speeding up further processing of the data, including the construction, the training and the application of prediction models. Besides, it allows one reducing the costs related to data acquisition and the required data storage capacity, since useless features obviously do not need to be collected nor stored anymore. Keeping in mind the considerations in [180] and the need for rentability many industries have to deal with, this seems of crucial importance in the current context.

Another important aspect of feature selection is that it allows one to better understand the problems at hand and to better interpret the prediction models, by determining which (groups of) variables mostly impact the outcome one wants to predict. This is of particular importance in the medical field and for instance in biomarker selection; from the thousands of genes whose activity level is obtained through microarray analysis, one tries to identify a small group of genes (a signature) strongly related to a pathology or to the response to a specific treatment (see e.g. [90, 96]). Obviously, such researches have an enormous potential impact in the understanding of complex disease developments as well as in the detection and cure of those diseases.

Closely related, feature selection has the advantage over other dimensionality reduction techniques such as projection [122] or feature extraction [88] that it preserves the original features instead of somehow transforming or combining them. This characteristic makes the aforementioned model interpretation possible. Besides, most of the projection or feature extraction techniques, whose most popular ones are principal components analysis and multidimensional scaling, are unsupervised, meaning that they are not able to consider the output in prediction problems; said otherwise, these approaches are not designed for being considered in combination with a prediction, classification or regression algorithm. They would indeed neglect a considerable amount of informative data.

Eventually, feature selection has to be distinguished from metric learning, where a weighted distance is learnt from the data to increase the accuracy of a prediction method [181, 182]; indeed, using such a distance is equivalent to using the basic Eu-

clidean distance in a new feature space, obtained by combining linearly or not the original features. It can thus be seen as a supervised extension of projection techniques. Feature selection is actually more closely linked to feature weighting [110] where, as a matter of fact, each feature is given a specific weight in distance computation (this is actually a particular case of metric learning where the covariance matrix of the Mahalanobis distance is constrained to be diagonal). A certain number of relevant features can then be kept, according to the weights they have been given.

Non-standard Data

Because of its numerous and important potential benefits, feature selection has become an extremely active reasearch field in Machine Learning and Data Mining, as will be evidenced in Section 1. However, the huge majority of existing works are dedicated to *classical* prediction problems, where a complete dataset of continuous or categorical features as well as a full output target vector are available.

Nevertheless, in many situtations, the data is not encountered in such an ideal and standard form or come with uncertainties on their values, as will be illustrated and detailed in Chapters 4 to 7. For instance, it easily understandable that in practice, some values may be missing (either in the dataset or int the target output). As another example, one can suspect that some of the provided labels in a classification problem are wrong. There is thus a real need for feature selection methods able to deal with such non-standard types of data.

The second part of the thesis is consequently dedicated to the analysis of different unconventional but important problems, including for instance missing data and label noise, for which a feature selection algorithm is proposed. We have chosen to generically denote those non-standard situations as *complex settings*.

Organisation of the thesis

The thesis is divided into three parts. The first one aims at provding the reader with the necessary background knowledge on feature selection. Chapter 1 first describes the three main approaches to feature selection. Chapter 2 introduces the three feature selection criteria on which will be based the subsequent developments, namely the mutual information, the Laplacian score and Hilbert-Schmidt independence criterion. None of the contributions of this part are original, except for the experimental comparison of mutual information estimators, whose results presented in Section 2.1.3 are extracted from a more complete work [54].

In the second part of the thesis, feature selection algorithms are proposed for different types of complex settings. Each chapter begins with an introduction to the

considered problem, an illustration of its practical interest and a brief related state-of-the-art, before the algorithm is introduced and its performances are illustrated. The results are mainly based on the three feature selection criteria introduced in Part 1. The four main types of problems addressed in this paper are semi-supervised learing, multi-label learning, regression with missing data and classification problems with unprecise class labels. They are respectively studied in Chapters 3 to 6.

The last part of the thesis is dedicated to the presentation of some theoretical considerations about the entropy and the mutual information. More precisely, Chapter 8 is dedicated to the study of the relationships between the mutual information, one of the most used feature selection criteria, and the classification risk. The objective is actually to characterize the situations where mutual information is a risk-optimal feature selection criterion and what are the consequences of using it when it is not optimal. This chapter also includes a direct risk classification estimator, which can be used as a criterion for feature selection in place of the mutual information. Similar considerations for regression problems are presented in Chapter 9, where links between the mutual information and the mean squared error and the mean absolute error are given. Again, the goal is to better apprehend the behaviour of the mutual information criterion in order to use it in the most efficient way.

Contributions

We end this chapter by detailing and giving some comments on the contributions of the thesis.

The first contribution, presented in Section 2.1.3, is to compare different MI estimators in the particular context of multivariate feature selection. The results confirm previous and quite natural observations about the inadequacy of traditional density-based MI estimators for multi-dimensional variables. More interestingly, the interest of a nearest neighbors-based MI estimator is also confirmed, after previous works have successfully used it for feature selection. Those conclusions are of great interest for the thesis, since many developments are based on the nearest neighbors-based MI estimator. They also confirm the estimator as an interesting tool for feature selection in traditional settings, which can be of broad interest. Of course, the list of compared estimators is not exhaustive; in particular, other estimators based on a similar nearest-neighbors statistics exist and could potentially provide valuable alternative solutions. Besides, the estimator has been tested with variables of moderate dimensionality, and no guarantee can be given when subsets of thousands of features have to be considered. The related publication is [54].

In Chapter 4, the problem of feature selection for semi-supervised problem is tackled. We first propose a solution for regression problems based on the popular Lapla-

cian Score (LS) criterion, while the whole literature about semi-supervised feature selection is actually dedicated to classification problems. Our proposal seems to behave well when only a few data points are available and to benefit well from the addition of new points. Its main drawback is its important number of parameters. We therefore provide a sensitivity analysis of the criterion regarding the different parameters and conclude that they can roughly be set to reasonable values, without affecting much its performances; this observation makes our criterion quite practical to use in practice. We then propose a way to turn univariate criteria based on the Laplacian Score into multivariate ones, using the Hilbert-Schmidt independence criterion (HSIC). The presented results are preliminary but indicate the interest of the proposed approach; its potential impact is quite important since many feature selection algorithms are actually based on the Laplacian score and can currently only be used to rank features individually. It is important to mention that most of the kernels designed for graph Laplacians possess several parameters whose influence on the feature selection process should be carefully analysed. The corresponding references are [51, 56].

In Chapter 5, we first propose a mutual information-based multivariate feature selection algorithm. It addresses to important lacks of existing methods: the need for discrete features and the impossibility to deal with multi-dimensional variables. The proposed methodology experimentally shows strong advantages over existing methods. However, its application is restricted to continuous problems for which the number of different label combinations is moderate. We also propose a multivariate criterion based on the HSIC and test it for problems with discrete features. First results are promising and there exists no restriction for the use of the criterion, which could easily be adapted to problems with continuous features. Related publications are [49, 48].

Chapter 6 is dedicated to the problem of missing data. We first introduce a straightforward but efficient algorithm to handle such data, which is a direct adaptation of the nearest neighbors-based mutual information estimator introduced in Section 2.1.3. We believe this approach can be of broad interest, especially for moderate rates of missingness. We then present the results of a joined work devoted to the development of a robust distance estimation technique. Besides a greater accuracy than competing approaches, the proposed methodology is able to deal with high rates of missingness and produces distance matrices which can safely be used in kernel methods. The potential impact of the approach is important, since many real-world datasets do contain missing values, while lots of machine learning and data mining methods are actually based on the distance between the data points. The corresponding papers are [58, 55, 61].

The last chapter of Part 2 is devoted to classification problems with uncertain labels. Our first contribution in this field is the proposal of a noise-tolerant entropy and mutual information estimator for problems with label noise. The basic idea is to estimate the true label of each data point and to use this information to compute the size

of the neighborhood containing a given number of points from the same class; this is actually the basis of the mutual information estimator presented in Section 2.1.3. Our approach exhibits good performances and has a clear advantage over noise-intolerant algorithms in the presence of label noise. However, it is based on a particular noise model and could become inadequate if the true underlying model noise is far from the assumed one. In the second part of the chapter we briefly address the problem of feature selection for datasets where the labels are given as probabilities over the different possible classes and propose an approach based on the Laplacian Score. The publications related to this chapter are [72, 57].

While Part 2 essentially proposes practical solutions to feature selection and related problems, Part 3 is dedicated to more theoretical considerations about the mutual information and the entropy. In Chapter 8, we illustrate the potential inadequacy of the mutual information regarding the classification risk but show that in practice, both quantities are often equivalent from a feature selection perspective. These observations are actually important as they (i) correct some wrong statements found in the literature about the complete adequacy between the risk and the mutual information and (ii) confirm the interest of the mutual information for feature selection, even providing additional arguments for its use. Besides, extensive experiments give insights on the behaviour of the mutual information during feature selection procedures, which can be of broad interest for practitionners. While the two main messages of the chapter can seem quite contradictory (MI is not optimal but should however be used), we do not believe it is actually the case, especially because of the small proportion of problems for which MI is actually not optimal. More details can be found in [7, 73].

Chapter 9 also gives considerations on the adequacy of the mutual information, in the particular context of regression problems. We show that under some conditions on the distribution of the prediction errors, the MI can be guaranteed to be optimal from a mean squared or a mean absolute error point of view. While the underlying hypotheses are strong (especially the fact that the error distribution is assumed to be the same for every data point), our objective is mainly to give first results on the topic, while confirming the interest of the mutual information criterion. More extensive results on the potential non-validity of the hypotheses and its consequences are interesting leads for future work. The corresponding publication is [74].

Table 1 summarizes the criterion used for the different complex problems tackled in the thesis.

	MI	LS	HSIC
Semi-supervised		x (univariate)	x (multivariate)
Multi-label	x (continuous)		x (discrete)
Missing values	x		
Label noise	x		
Probabilistic labels		x	

Table 1: Feature selection criteria used for non-standard problems.

Part I

Feature selection: background and tools

Chapter 1

Feature selection: approaches and quality criteria

Contents

As previously mentioned, feature selection is a preprocessing step of major importance in many prediction problems; it has therefore been adressed extensively for decades. This chapter intends to provide the reader with basic feature selection notions and to give a brief overview of the various approaches generally considered in the literature for traditional problems. We postpone the state-of-the-art specific to each particular kind of non-standard data studied in this thesis to the corresponding sections in Part 2. In the end of the chapter, we also give some comments on the way feature selection approaches are generally evaluated.

1.1 Approaches to feature selection

There exist different paradigms according to which feature selection algorithms could be distinguished. Here, we choose to introduce them following the way the subsequent prediction algorithm is considered in the feature selection process. Three main categories can then be distinguished, filters, wrappers and embedded methods, which are introduced in Sections 1.1.1 to 1.1.3. As will be made clear, filters and wrappers generally require a heuristic procedure to build a feature subset based on a relevance criterion. Some popular strategies are described in Section 1.1.1.

The reader interested in a more extensive introduction to feature selection is referred to [86]. A deeper literature survey can be found in [135, 87] and in [154] for the special case of Bioinformatics.

1.1.1 Filters

Filters do not take into account the final prediction model to select features and are rather based on a relevance criterion independent of this model. The most popular of these criteria include the correlation coefficient [91, 187], the mutual information [161, 37], which will be more deeply introduced in Section 2.1 and other information-based criteria such as the Markov blanket [111]. Spectral feature selection algorithms [199] based on the eigensystem of a similarity matrix defined from the dataset and closely related to the notion of Graph Laplacian [93] introduced in Section 2.2 also belong to filters. In addition, many other heuristic relevance criteria have been proposed in the literature; this is the case of the popular Relief algorithm [108] and its extensions, which are based on the ratio of the distance between the samples and their nearest neihgbors of the same and of another class. The idea is that according to good features, close samples are likely to belong to the same class. These methods can also be considered as filters. Filters are generally considered to be fast (they do not require building and training potentially complex prediction models), simple and general, making them extremely popular in practice. It is however important to mention

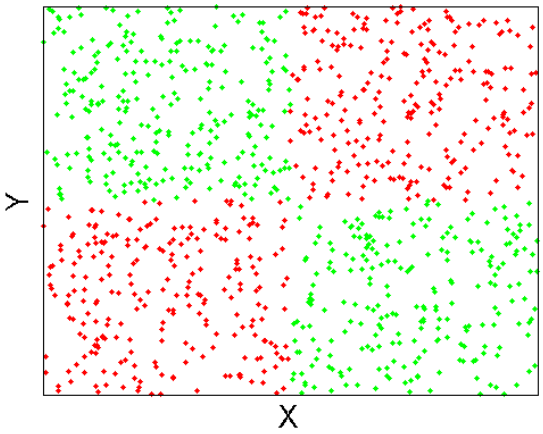


Figure 1.1: Continuous version of a XOR problem. The colour of the points represents their class label.

that the rapidity and the computational efficiency of the filters is obviously dependent on the chosen quality critetion.

Different strategies can be considered to optimize the chosen relevance criterion. Obviously, the optimal solution would be to test every possible feature subset and to chose the one that maximises the considered criterion. This is unfortunately not realistic in practice since the number of subsets to test this way is 2^d, where d is the number of features. As soon as d exceeds 30, more than 1 billion feature subsets already have to be considered.

The most straightforward approach consists in individually ranking the features and in chosing the best rated ones. Despite its rapidity and easiness, this basic strategy suffers from a huge drawback; it does not consider any possible joint relevance or redundancy between the features, while both are in practice extremely important to detect. Indeed, two features can for instance be completely irrelevant when considered individually and be extremely informative when considered together. Think for instance of the XOR problem or of a continuous version of it, as illustrated in Figure 1.1. Obviously, neither the value of X or Y alone gives any clue on the class of a data point. However, knowing both the values of X and Y completely determines the class. Besides, it is as well important to eliminate redundant features, as they do not provide any useful information and increases the dimensionality of the datasets. The best n features are thus not always the best n features and ranking methods are not to be recommended, as acknowledged for decades [36, 37].

In practice, heuristic subset construction algorithms have thus to be considered. The most popular ones are greedy search algorithms, including forward, backward and forward/backward search. In a forward search, one starts with an empty set of features and successively add the feature whose addition to the current set of selected features maximises the chosen evaluation criterion. Well-known examples include maximum relevance minimun redundancy (MRMR) algorithms, where the subset quality criterion is the difference between the relevance of a feature and the sum of its pairwise redundancy with the previously selected features (where both the relevance and the redundancy can be evaluated by information-theoretic quantities) [139], or conditional mutual information [68]. However, these two criteria are univariate, in the sense that even if the feature redundancy is actually taken into account, the relevance between each feature and the outptut is still measured individually. They could thus still not be able to detect jointly relevant features as the ones in Figure 1.1 but they clearly outperform ranking procedures. A possible solution to this problem is to use multivariate criteria; two of them, namely multivariate mutual information [150, 70] and Hilbert-Schmidt independence criterion [166], will be introduced in Sections 2.1 and 2.3, respectively. Such criteria are able to evaluate the joint relevance and redundancy of a susbset of features.

As mentioned above, one sometimes prefers backward elimination procedures, which start with all the features and recursively eliminate the one whose removal optimizes the value of the relevance criterion. Their main advantage regarding forward search is that they will not be affected by a bad choice of the first feature. As an example, a forward search will not be able to detect one the two relevant features of a XOR problem as informative among other (noisy) features. On the contrary, a backward procedure combined with a multivariate criterion will eliminate useless features. However, backward procedures are required to initially evaluate a criterion in a potentially very high-dimensional space, which may become untractable or unreliable in practice. Forward/backward search procedures, performing a backward step after some forward steps have also been proposed (see e.g. [167]).

Besides greedy search, other search algorithms including genetic algorithms [185] or tabu search [191] can also be thought of. Contrarily to greedy procedures, genetic algorithms are random, in the sense that running them on the same dataset twice will not necessarily lead to the same feature subset.

1.1.2 Wrappers

Contrarily to filters, wrappers are feature selection techniques which explicitely take the prediction model of interest into account. The objective is thus often to maximise the performances (e.g. the accuracy) of a specific prediction model, considered here

as a black box, by using an adequate feature subset [109]. Cross-validation procedures are generally considered to assess the performances of a feature subset. Filters are thus conceptually very simple and quite easy to use.

The main criticism often made to wrappers is their the need for massive amounts of computation required to build multiple prediction models. Indeed, wrappers can be highly time-consuming when complex models with hyperparameters to tune are considered or when very large datasets have to be dealt with; for instance, they can hardly be used for text classification problems [69]. On the other hand, they are likely to lead to better prediction performances than filters, as they are specially designed for this objective. Comparative studies between wrappers and filters such as the one in [101], devoted to the particular context of microarray data classification, experimentally support these claims.

The same subset construction procedures as the ones described in Section 1.1.1 for filters can equally be considered for wrappers. However, the questions of relevance and redundancy do not need to be investigated anymore since only the eventual performance of the model actually matters. It is important to note that in [147], the author advocates for the use of simple greedy search strategies in wrapper approaches. They have the advantages over more complicated search procedures to limit the risk of overfitting and to keep the number of evaluated feature subsets quite moderate.

It is worth noting that some works combine both wrapper and filter approaches, trying to get the best of both worlds (the rapidity of the filters and the higher performances of wrappers). Examples of such hybrid methods include [203, 158, 40].

1.1.3 Embedded methods

The last main approach to feature selection is the use of embedded methods. The idea behind such methods is to consider the feature selection as part of the learning procedure. Because of this quite general definition, many feature selection algorithms can actually be considered as embedded methods. One of the first and most popular examples of embedded method is the CART [22], which incorporates a mechanism to perform feature selection.

Another example is the work in [90], where the authors propose a backward elimination procedure using the difference in the norm of the weights of a linear support vector machine (SVM) as the relevance criterion; this of course considerably limits the amount of models that have to be built compared to a classical backward elimination procedure since only one model is built at each step. Other SVM-based feature selection approaches are proposed and studied in [143].

The popular optimum brain damage (OBD) approach [120] and related methods can also be considered as embedded methods. The approach, originally designed to

prune weights in a neural network, uses a second order Taylor expansion to approximate the second derivative of the saliency of a feature, i.e. the effect of the deletion of this feature on the training error. This criterion can be used with a greedy backward elimination scheme.

Today, the most popular embedded methods are probably regularized prediction models where a regularisation term, penalising the number of features actually used, is added to the original objective function of the model. Learning in this framework thus requires finding a compromise between optimising a quantity somehow related to the performances of the model and limiting the number of effective features. The most well-known example of such an approach is the LASSO [172] which proposes learning the weights of a linear regression by simultaneously (i) minimising the sum of the squared prediction errors and (ii) minimising a quantity proportional to the sum of the absolute value of the weights (the 1-norm of the weight vector). The specificity of this particular regularisation tends to make the weight vector sparse (i.e. many weights will be set exactly to 0).

Because of its simplicity and the existence of efficient methods to solve it (such as the Least Angle Regression (LARS) [60]), LASSO has become extremely popular. Many research works have and still focus on the determination of new regularisation criteria leading to sparse regression models [8]. As an example, elastic net [205] penalizes both the 1-norm and the 2-norm of the weight vector, and has been reported to outperform LASSO and to jointly select or discard groups of very correlated features.

Researchers also try to develop regularization that allows the incorporation of prior knowledge about the structure of the features [102]. For instance, the group LASSO [188] explicitely allows the user to choose ensembles of features which will all be given the same weight. In [197], the possibility to define a hierarchical structure between features (i.e. the selection of some features are conditionned to the selection of other ones). LASSO and its extensions have been particularly successful for microarray classification problems whose particular nature (they are described by very few samples and lots of features) makes the use of linear models almost mandatory [154, 42, 129, 96].

In addition to linear regression, regularization has also been proposed for other prediction models. Sparse logistic regression [115, 128], sparse SVM [27, 171] and sparse probit regression [66] have been proposed along this line.

1.1.4 Discussion

It must be underlined that today, the strict distinction between filters and wrappers tends to become less and less crisp. Indeed, it frequently happens that the criterion used in the filter approach is actually based on a particular prediction model, or that

this criterion implicitely assumes the use of a specific model. For instance, the mutual information estimator introduced in Section 2.1.3 is obviously related to a k nearest neighbors model, as will be made clear. The same is also true for the Relief criterion, mentioned in Section 1.1.1. Similarly, the correlation coefficient assumes a linear relationship between the features and the output. This issue and its implications are further discussed at the end of the thesis.

In the remaining of the thesis we essentially focus on filter methods, based on the three criteria described in Chapter 2. The reason why we choose filters instead of both wrappers and embedded methods is their greater independence from a prediction model, and therefore their greater generality. Indeed, as we just mentioned, filters are rarely totally independent from a prediction model. However, a simple model is often only used as a tool to build a more general criterion, which can further be combined with any (other) prediction model. This is for instance the case for the mutual information, whose goal is to detect any kind of dependencies between random variables; the estimator presented in Section 2.1.3, based on a nearest neighbors statistics, is actually used as a proxy for the true mutual information. On the contrary, both wrappers and embedded methods are tight to a particular prediction model.

1.2 Performance evaluation of feature selection methods

In the literature, most feature selection algorithms are only evaluated through the accuracy of a prediction model trained and tested using the selected features; this is also the case for the methods we propose in Chapters 4 to 7 since building accurate prediction models is often the objective one is the most interested in.

It has however to be mentioned that in practice, the stability or the robustness of the feature selection algorithms (i.e. the sensitivity of the selected features to small perturbations in the dataset) is also important to consider [105]. This is for instance particularly true for biomarker selection from genomics data, where few data points are typically available. In this context, the interpretability of the selected features (or biomarkers) from a biological point of view is essential and it is important that a feature selection method returns consistent feature sets when data from a few additional patients are added to the original dataset. Algorithms producing stable subsets (of good predictive power) are thus generally preferred to algorithms returning slightly more predictive but highly volatile subsets in the field of bioinformatics [94, 153]. More generally, if a feature selection algorithm returns very different feature subsets on very similar data, presenting and considering only one of those subsets can obviously be misleading in practice. Of course, stability alone does not give any clue on

the quality of the feature selection; it should ideally be combined with another quality criterion, such as the accuracy of a prediction model.

The reason of the instability of feature selection algorithms is that the values of the objective function guiding the search are actually random variables and that we do not have access to the exact value of the criterion, since it is estimated on the available data. The construction of the feature subset thus drastically depends on the accuracy and the variance of the estimation of the search criterion. Different approaches have been proposed in the literature to develop stable feature selection algorithms, including for instance ensemble feature selection (see e.g. [1]). An overview of stable feature selection techniques for biomarker discovery can be found in [94].

To assess the consistency of feature selection algorithms, stability indexes have been proposed. One of the most popular ones has been proposed by Kuncheva [116]. The index is based on the cardinality of the intersection of the feature subsets and incorporates a term that corrects a bias introduced by the chance of finding common features in two feature subsets chosen at random. Other possible stability indexes include for instance the proposal in [117], where an entropy-based measure is proposed.

Also note that in the context of regularized regression models (such as the LASSO and related methods), the consistency precisely denotes, as the number of data points grows, the ability of the method to recover the sparse vector from which the data is assumed to be generated from. Consistency is an important property, related to the notion of stability, which has for instance been studied in details for the LASSO (see e.g. [198]) and the Group LASSO [11]; adaptive versions of the LASSO which can be proven to be always consistent have been proposed [204].

Chapter 2

Feature selection criteria

Contents

This chapter is dedicated to the presentation of three feature selection criteria which will serve as the basis for most of the remaining developments in this thesis. The first one, mutual information (MI), is a very popular feature selection criterion on which many of the feature selection algorithms presented in this text are based. Besides, in Part 3, we study some if its properties; more specifically we investigate the situations in which it can be an optimal criterion regarding prediction accuracy.

In Section 2.2, we introduce the Laplacian score (LS), a more recent criterion originally introduced for unsupervised problems. We essentially use it to address semi-supervised and wrongly supervised problems in our research, for reasons that will be made clear.

Finally, Section 2.3 introduces the Hilbert-Scmidt independence criterion (HSIC), used as a measure of dependence between two kernels each defining the inner product of the mapping of a domain to a possibly high-dimensional feature space.

2.1 Mutual information

Mutual information is a quantity first introduced by Claude Shannon back to 1948 [161] in his famous *Mathematical Theory of Communication*, for which he is still known as the father of information theory. More gentle introductions to mutual information and information theory can be found in [5, 37].

2.1.1 Definitions

Let us define two discrete random variables X and Y, possibly multidimensional, whose respective domains are $\mathcal{X}$ and $\mathcal{Y}$. Les us denote their probability mass function as p_X and p_Y respectively. We can then define the entropy [161], a fundamental quantity in Shannon's theory of information, as

$$H(X) = -\sum_{x\in\mathcal{X}} p_X(x)\log p_X(x) = \mathbb{E}\left[-\log p_X\right] \tag{2.1}$$

where $\mathbb{E}\left[.\right]$ denotes the expected value. The entropy is a measure of the uncertainty of a random variable, which corresponds to the expected unpredictability in this random variable, measured by its information content $-\log p_X$. Using Eq. (2.1), it is then possible to defive the mutual information (MI) between two random variables as

$$I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y) \tag{2.2}$$

with $H(Y|X)$ being the conditional entropy of Y given X:

$$H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log\frac{p_Y(y)}{p_{X,Y}(x,y)} \tag{2.3}$$

where $p_{X,Y}$ denotes the joint probability mass function of random variables X and Y.

Following Eq. (2.2) and (2.3), the MI of two variables X and Y can equivalently be formulated as the Kulback-Leibler (KL) divergence of the product of the distributions of the random variables $p_X p_Y$ and the joint distribution $p_{X,Y}$ of these variables:

$$\begin{aligned} I(X;Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} \\ &= KL_D(p_{X,Y}||p_X\,p_Y). \end{aligned} \tag{2.4}$$

Noting that $p_{X,Y}(x,y) = p_{X|Y}(x|y)\,p_Y$, it possible to write

$$\begin{aligned} I(X;Y) &= \sum_{y\in\mathcal{Y}} p_Y(y)\sum_{x\in\mathcal{X}} p_{X|Y}(x|y)\log\frac{p_{X|Y}(x|y)}{p_X(x)} \\ &= \sum_{y\in\mathcal{Y}} p_Y(y) KL_D(p_{X|Y}(x|y)||p_X(x)) \\ &= \mathop{\mathbb{E}}_{Y}\left[KL_D(p_{X|Y}(x|y)||p_X(x))\right] = \mathop{\mathbb{E}}_{X}\left[KL_D(p_{Y|X}(y|x)||p_Y(y))\right] \end{aligned} \tag{2.5}$$

This last equation thus expresses mutual information as the expected value of the KL divergence of the marginal distribution p_X and the conditional distribution $p_{X|Y}$.

It is important to note that the exact same developments can be made if the considered random variables are both continuous, or if one is discrete and the other one is continuous. For continuous random variables, the sums in Equations (2.1), (2.3), (2.4) and (2.5) are simply replaced by integrals. In the case of two continuous variables, Eq. (2.4) thus becomes

$$I(X;Y) = \int_{\mathcal{X}}\int_{\mathcal{Y}} p_{X,Y}(x,y)\log\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\,dxdy \tag{2.6}$$

where p_X, p_Y and $p_{X,Y}$ now denote continuous probability density functions. Formulating the entropy and the mutual information as expectations prevents one from using sums or integrals and thus leads to more general definitions for these two quantities.

Eventually, as can be deduced from the above definitions, let us note that MI is a positive and symmetric quantity, i.e. $I(X;Y) = I(Y;X) \geq 0$.

2.1.2 Interpretation and interest for feature selection

Keeping in mind that the entropy is actually a measure of the uncertainty about the values taken by a random variable, MI can be given a nice interpretation from a feature selection point of view. Indeed, Eq. (2.2) defines MI as the difference between two uncertainty measurements; more precisely, if variable X corresponds to a set of features and if variable Y corresponds to an output target, $H(Y) - H(Y|X)$ is the reduction of uncertainy on Y once X is known; it is equivalently the amount of information that X carries about Y. In a feature selection context, if one tries to find a subset of features maximising the MI with the target output vector, she actually tries to select a subset of features which contain the most information about the output. Mutual information is thus intuitively a sound criterion for feature selection. Equation (2.5) confirms this interpretation; if X and Y are independent $p_{Y_X} = p_Y$, whose divergence with itself is obviously equal to zero. On the contrary, as the dependence between X and Y increases, so does the value of $\mathbb{E}_X\left[D_{KL}(p_{Y|X}(y|x)||p_Y(y))\right]$. A similar interpretation can be given for Eq. (2.4).

Other useful properties

Besides this interpretation, MI possesses other desirable properties for feature selection. First, it is able to detect non-linear relationships between variables, which is for instance not the case with the very popular correlation coefficient. This is an important asset since it is often the case in practice that the features are non-linearly related to the target output. On the other hand, it also means that MI should not be used before learning linear models, since the features it will select, if non-linearly dependent to the target output, may actually be irrelevant for such models. Another advantage is that MI can naturally be defined for groups of variables; subsets of variables can thus be jointly evaluated, whereas this is not true for univariate criteria, as already discussed in Section 1.1.1.

The aforementioned properties were essentially the ones originally leading to the popularity of the MI for feature selection, following the seminal paper of Battiti [14]. The Fano and Hellman-Raviv bounds, despites the anteriority of their publication, were only related later to the feature selection problem [136, 67, 24], but support as well the interest of MI for feature selection in classification.

Fano and Hellman-Raviv bounds

In classification problems, the objective one is traditionnaly interested in is to minimise the misclassification probability that a classification algorithm will achieve. The goal is thus that given a sample $x \in \mathcal{X}$, the predicted class label $\hat{y} \in \mathcal{Y}$ be equal to the

actual class label $y \in \mathcal{Y}$. The error probability P_e for a given classifier is then

$$P_e = \mathop{\mathbb{E}}_{Y|X=x}[\mathbb{I}[\hat{y} \neq y]] = \sum_{y \in \mathcal{Y}} p_{Y|X}(y|x)\,\mathbb{I}[\hat{y} \neq y] \tag{2.7}$$

where $\mathbb{I}$ is the indicator function, i.e.

$$\mathbb{I}[\hat{y} \neq y] = \begin{cases} 0 & \text{if } \hat{y} = y, \\ 1 & \text{if } \hat{y} \neq y \end{cases}. \tag{2.8}$$

The Fano and Hellman-Raviv bounds relate the misclassification probability P_e to the conditional entropy, in the particular context of an optimal (i.e. Bayesian) classifier, predicting y^* as:

$$y^* = \max_{\hat{y} \in \mathcal{Y}} p_{Y|X}(\hat{y}|x). \tag{2.9}$$

First, Fano [65] has proven the existence of two lower bounds on P_e. The weaker bound is:

$$H(Y|X) \leq 1 + P_e \log_2(n_Y - 1) \tag{2.10}$$

where n_Y denotes the number of possible classes. The stronger bound is:

$$H(Y|X) \leq H(P_e) + P_e \log_2(n_Y - 1). \tag{2.11}$$

Both Eq. (2.10) and (2.11) originally give bounds on $H(Y|X)$, but can be inverted to provide a lower bound on P_e. In practice, the stronger bound (2.11) is less easy to manipulate because P_e cannot be isolated in a closed-form and the bound on P_e has to be computed numerically. Nevertheless, the strong Fano bound is much more useful than the weak one because (i) the weak bound (2.10) does not apply to binary classification problems since in this case, it trivially reduces to $H(Y|X) \leq 1$ and (ii) the weak bound is generally much looser than the strong one, especially when P_e is small. which is the situation of interest for classifier design [67].

The misclassification probability P_e can also be upper-bounded by the Hellman-Raviv inequality [97]

$$P_e \leq \frac{1}{2} H(Y|X). \tag{2.12}$$

The three aforementioned bounds are illustrated in Figure 2.1 for a 3-class problem ($n_Y = 3$); the weak Fano bound is shown to be quite useless regarding the strong bound. Please note that similarly to the risk, these bounds correspond to a given X and the conditional entropy in Eq. (2.10) to (2.12) is actually the specific conditional entropy $H(Y|X = x)$.

One could rightly argue that the Fano and Hellman-Raviv bounds are actually

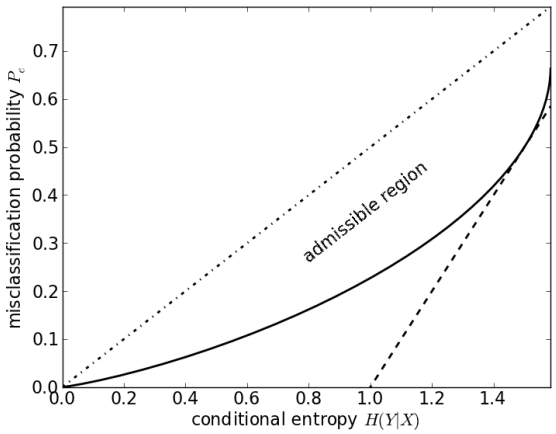


Figure 2.1: Weak Fano bound (dashed line), strong Fano bound (plain line) and Hellman-Raviv bound (dash-dotted line) with three classes.

about the conditional entropy $H(Y|X)$ rather than the MI $I(X;Y)$. Nevertheless, for a given classification problem, the outptut target will remain identical during the feature selection procedure and the term $H(Y)$ in the first part of Eq. 2.2 can be considered as constant. Therefore, maximising the MI becomes equivalent to minimising the conditional entropy. Equations (2.10) to (2.12) thus provide arguments to support the use of MI as feature selection criterion in classification problems, since reducing the conditional entropy actually leads to a lower upper bound and a higher lower bound on the probability of misclassification P_e. However, contrarily to what is sometimes read [136, 67, 24], the Fano and Hellman-Raviv bounds do not guarantee the optimality of the MI for feature selection. This will be further discussed in Section 8.

2.1.3 Estimation

As can be seen in the definitions given in Section 2.1.1, computing the MI between two groups of random variables requires the knowledge of probability density functions. However, in real-world problems, these quantities are likely to be unknown, and MI has to be estimated from the available data.

The most popular and straightforward way to do so is to first estimate the required probability densities, before plugging them into Eq. (2.5) or an equivalent expression. Many density estimators have been proposed in the literature, whose most simple one is the basic histogram which consists in dividing the space into bins of fixed size and counting the number of points falling in each bin. To become independent from the size of the bins and their origin and to avoid sharp discontinuities between the

different bins, one can use a kernel density estimator (KDE) instead. The density is then estimated as:

$$\hat{p}_X(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_{i*}}{h}\right), \tag{2.13}$$

where n is the number of observations in the dataset, h is the window width and K is the kernel function required to integrate to one in order for $\hat{f}$ to be a probability density [137]; x_{i*} denotes the i^{th} observation of the data set. One option for K is the Gaussian kernel, leading to the following density estimator:

$$\hat{p}_X(x) = \frac{1}{nh\sqrt{2\pi}}\sum_{i=1}^{n} \exp\left(\frac{-(x - x_{i*})^2}{2h^2}\right). \tag{2.14}$$

The choice of h is crucial and constitutes an open problem, with many existing possible alternatives (see e.g. [163]). Other density estimation strategies involve for instance B-splines [41]. The idea is again to discretise the space into bins, but to allow each point to belong simultaneously to different bins with a degree of belongingness given by the coefficients of a B-Splines function.

The three aforementioned density estimators have the similar limitation that they become unreliable as soon as the dimensionality of the data increases. Indeed, as mentioned earlier, the number of available samples needed to obtain accurate estimates increases exponentially with the dimension of the space. In practice, however, the number of available samples is often limited. Consequently, the majority of the bins will be empty, leading to sharp and inacurate estimations. Besides, for the histogram and the B-splines estimator, the number of bins to consider increases exponentially with the dimension of the space, which becomes quickly untractable if a decent resolution has to be obtained.

Some works try to directly estimate the MI, such as the proposal in [39]. In this paper, the authors use an adaptive partitioning of the observation space to take into account the fact that with basic fixed partitions, much of the bins are not used to estimate the MI and can thus be replaced by fewer bins. The paper furthermore shows that the MI can be approximated arbitrarily closely in probability by calculating relative frequencies on appropriate partitions. However, this approach can hardly be considered for high dimensions because of its computational requirements.

A possible solution is to consider nearest neighbours-based MI estimators, which bypass direct density estimation and space partitioning, the most critical steps in density and MI estimation. The estimator introduced in [114] by Kraskov *et al.* is an example of such approach, which has already been used successfully for feature selection [150].

This estimator is actually based on the Kozachenko-Leonenko estimator of en-

tropy, computed as [113]:

$$\hat{H}(X) = -\psi(k) + \psi(n) + \log(c_d) + \frac{d}{n}\sum_{i=1}^{n}\log(\varepsilon_X(i,k)) \tag{2.15}$$

where k, a parameter of the estimator, is the number of nearest neighbors considered, n is the number of samples in X, d is the dimensionality of these samples, c_d is the volume of a unitary ball of dimension d and $\varepsilon_X(i,k)$ is twice the distance from the i^{th} observation x_{i*} to its k^{th} nearest neighbor, usually determined using the Euclidean distance. Finally, ψ is the digamma function defined as:

$$\psi(k) = \frac{\Gamma'(k)}{\Gamma(k)} = \frac{d}{dk}\ln\Gamma(k)\,,\ \Gamma(k) = \int_0^\infty x^{k-1}e^{-x}dx. \tag{2.16}$$

Function ψ can be shown to satisfy the following recursion:

$$\psi(x+1) = \psi(x) + \frac{1}{x},\ \psi(1) = C,$$

with $C = -0.5772\ldots$ being the Euler-Mascheroni constant.

More precisely, in [113], the authors first propose a density estimator

$$\log\hat{p}_X(\mathbf{x_{i*}}) = \psi(k) - \psi(n) - \log c_d - d\log\varepsilon_X(i,k), \tag{2.17}$$

before deriving an entropy estimator by noting that following the continuous version of Eq. (2.1), the entropy can actually be estimated as:

$$\hat{H}(X) = -\frac{1}{n}\sum_i \log p_X(\mathbf{x_{i*}}). \tag{2.18}$$

The estimator is based on the sole hypothesis that $p_X(\mathbf{x_{i*}})$ remains constant in the hypersphere of diameter $\varepsilon_X(i,k)$ centered in x_{i*}.

Based on (2.15), Kraskov et al. [114] derived two slightly different estimators for regression problems, i.e. problems with a continuous output variable. The most widely used is:

$$\hat{I}(X;Y) = \psi(n) + \psi(k) - \frac{1}{k} - \frac{1}{n}\sum_{i=1}^{n}(\psi(\tau_x(i)) + \psi(\tau_y(i))) \tag{2.19}$$

where $\tau_x(i)$ is the number of points located no further than $\varepsilon(i,k)$ from the i^{th} observation x_{i*} in the X space and $\tau_y(i)$ is defined similarly for y_i in the Y space. Here, $\varepsilon(i,k)$ is defined as $\max(\varepsilon_X(i,k),\varepsilon_Y(i,k))$ which corresponds to the maximum norm in

the joint (X,Y) space Z: $\left\|z_{i*}-z_{j*}\right\| = \max\left(\left\|x_{i*}-x_{j*}\right\|, \left\|y_{i*}-y_{j*}\right\|\right)$. In practice, any norm could theoretically be used.

Based on the same entropy estimator (2.15), Gomez et al. derived a similar MI estimator dedicated to classification problems [82]. Empirically estimating the probability distribution of the discrete class vector as $p(y=y_c)=n_c/n$, whith n_c the number of points whose class value is y_c, and keeping in mind that

$$I(X;Y) = H(X) - H(X|Y) = H(X) - \sum_{y\in\mathcal{Y}} p_Y(y)H(X|Y=y), \tag{2.20}$$

it is quite straightforward to derive the following MI estimator:

$$\hat{I}_{class}(X;Y) = \psi(N) - \frac{1}{N}n_c\psi(n_c) + \frac{d}{N}\left[\sum_{n=1}^{N}\log(\varepsilon(n,K)) - \sum_{c=1}^{C}\sum_{n\in y_c}\log(\varepsilon_c(n,K))\right]. \tag{2.21}$$

In Eq. (2.21), C is the total number of classes while $\varepsilon_c(n,K)$ has the same meaning as $\varepsilon_X(n,K)$ in (2.15), but is limited to samples belonging to class c.

It is important to note that other nearest neighbours-based MI or entropy estimators have recently been proposed [179, 124]. They have not been considered in this work, but are likely to show similar interesting properties as the estimator proposed in [114].

Brief experimental comparison

In the remaining of this section, we briefly illustrate the differences in performances obtained by the previously introduced MI estimators, in the particular context of feature selection. In particular, the behaviour of the different estimators is studied when the dimensionality of the data increases. Indeed, if a forward search strategy is used to build a feature subset, the dimensionality of the data from which MI is estimated will increase at each step; conversely this dimensionality may be very high at the begining of a backward search. The results presented here are part of a more extensive work published in [54], where all technical and implementation details can be found. Comparisons of MI estimators in other contexts have also been proposed in the literature, see e.g. [106, 178].

Accuracy The first criterion of comparison is the accuracy of the MI estimation. To compare the estimators according to this criterion, we have chosen to work with correlated Gaussian features of unit variance and zero mean, as it is one a of the only multivariate distributions for which the MI can be given an analytical expression. Indeed, it can be shown [39] that for m such features $X_1 \ldots X_m$, the high dimensional

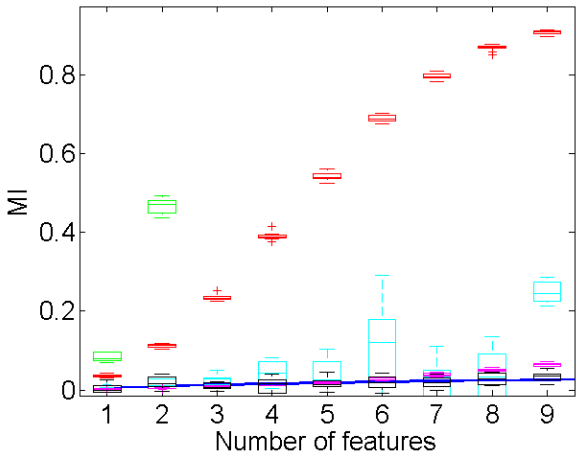


Figure 2.2: Boxplots of MI approximation of the MI for correlated Gaussian vectors by 5 estimators with $r = 0.1$: the basic histogram (green), a KDE (red), an adaptive histogram (cyan), a NN-based estimator (black) and a B-splines estimator (magenta). The solid blue line is the true MI.

redundancies [39, 114] between these m features is:

$$I(X_1,\ldots,X_m) = -0.5 \times \log[det(\sigma)] \tag{2.22}$$

where σ is the feature covariance matrix.

From the definition of MI (2.6), it can also be deduced that:

$$I((X_1,\ldots,X_{m-1});X_m) = I(X_1,\ldots,X_m) - I(X_1,\ldots,X_{m-1}). \tag{2.23}$$

Therefore, Eq. (2.22) allows us to know the exact value of $I((X_1,\ldots,X_{m-1});X_m)$ given the correlation between the features is known. In Figures 2.2 and 2.3, the accuracy of the five described MI estimators is compared; the covariance r between each pair of features is set to an identical value of 0.1 and 0.9, respectively. The sample size is 1000 and the estimation is repeated on 100 artificially generated datasets. The parameter k of the estimator is set to 6.

As it can be seen, the nearest neighbours-based MI estimator shows the more accurate and stable results over the repetitions, especially for highly correlated gaussians, i.e. for problems where there is a more important dependence between the features. The B-splines-based estimator appears as the principal outsider, as it is quite accurate for $r = 0.1$ and has a satisfying behaviour for $r = 0.9$, even if it strongly underestimates

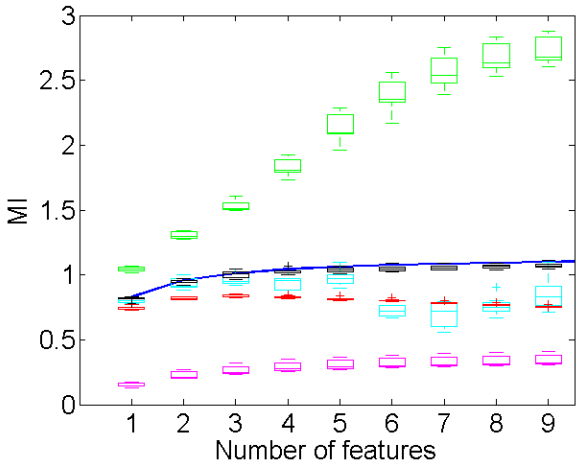


Figure 2.3: Boxplots of MI approximation of the MI for correlated Gaussian vectors by 5 estimators with $r = 0.9$: the basic histogram (green), a KDE (red), an adaptive histogram (cyan), a NN-based estimator (black) and a B-splines estimator (magenta). The solid blue line is the true MI.

the MI. As soon as the dimensionality exceeds 3, the three others compared estimators produce inacurate and unreliable results.

MI between independent features In feature selection, a very desirable property for a MI estimator is to be able to assign a value close to zero to the MI between independent (groups of) variables. Said otherwise, it is important to make sure that a large value of MI actually does translate a strong dependence between the variables and does not result from a weakness or a bias of the estimator.

Besides, the MI is not bounded to a known interval (as $[-1,1]$ for the correlation coefficient); the relevance of each feature subset cannot thus be assessed on the sole basis of the value of the MI. A possible solution is therefore to establish the relevance of a feature subset by estimating the MI between a randomly permuted version of this subset and the target outptut vector; if it is significantly lower than the MI between the subset and the target output, then the feature subset can be considered as relevant, in the sense that it brings more information about the output than a random vector with the same density. Such a strategy has for instance been successfuly used to decide when to stop a forward search procedure and to set parameter k of the MI nearest neighbours-based MI estimator [70].

It is thus important in practice to study how the MI between the *true* features and

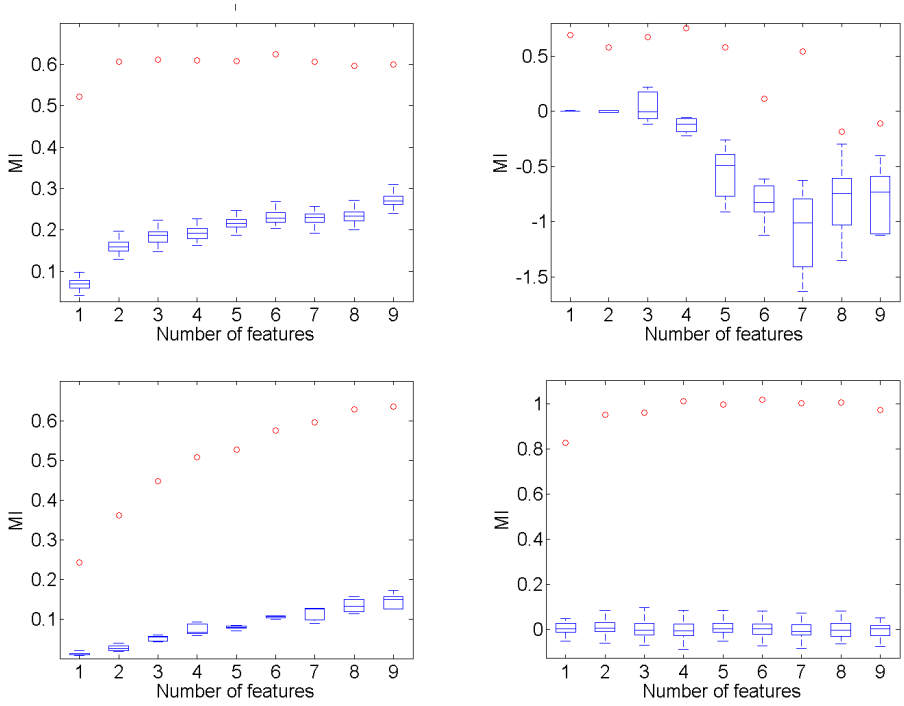


Figure 2.4: Estimated MI between a group of features and the output (circles) and boxplot of the estimated MI between the same group of features and a permuted output: KDE (upper left), adaptive histogram (upper right), B-splines estimator (lower left), nearest neighbours estimator (lower right).

a randomly permuted target ouput is estimated (permuting the features or the output is actually equivalent). Theoretically, the MI estimated in this way should be 0 as no more relationship is supposed to remain between the observations and the permuted output.

The results obtained with the different estimators are illustrated for the Nitrogen dataset (see Chapter 3 for details). The goal is to predict the nitrogen content of grass samples. A forward feature selection procedure using the NN-based estimator has been conducted to select the first best successive feature subsets. For each subset of increasing dimension, the *regular* MI and the MI with the permuted output are then estimated for 100 random permutations of the output. The performances of the estimators are thus compared on the same sets of relevant features. Note that the results for the histogram are not reported here, since this method is obviously not adapted to multidimensional problems and has already shown the worst accuracy.

Figure 2.4 shows the estimated MI as a function of the number of features for the compared estimators. The NN-based estimator returns values very close to 0 for the permuted output. This conclusion is in good agreement with the conjecture made in [114] that equation (2.6) is exact for independent variables; however, the authors did not provide a proof of this result. The same is not true for the other estimators, which all show important deviations from a zero value as the dimensionality increases. For instance, the results for the adaptive histogram [39] are particularly unreliable: the MI for 6,8 or 9 relevant features is lower than the MI with three random features. Besides, the estimations become strongly negative when the dimension exceeds 3, which translates high instability of the algorithm, which could be caused by the relatively small number of features. Results on alternative datasets presented in [54] support these claims.

Let us also notice two undesirable facts about MI estimation. First as already mentioned above, slightly negative values are sometimes produced. This has no theoretical justification since MI is a positive quantity and can easily be dealt with in practice, by setting negative values to 0. Secondly, it can be seen that the MI sometimes decreases after the addition of some variables. Again, this is theoretically not sounded since considering more features can never reduce the information content about the outptut [37], even if a decrease in MI has often been used as a stopping criterion in greedy feature selection algorithms.

Sensitivity to the parameter value To conclude, we illustrate how the choice of the parameter(s) of an estimator can actually influence the MI estimation. To this end, Figure 2.5 shows the MI for a relatively small number of features, estimated using the narest neighbours-based and the B-Splines-based estimators, as the latter appears as a good alternative to the nearest neighbors-based estimator for small dimensions. The parameter in this estimator is k the number of neighbours. In the splines-based estimator, there are two parameters, the degree of the splines and the number of bins per dimension. Here, we focus on the second one, as the degree of the splines has been reported to have less influence on the output. Various plausible values of those parameters are considered in the comparison and the dataset considered is Nitrogen.

In Figure 2.5, the estimator proposed in [114] shows very litle sensitivity to the value of the parameter k. On the contrary, the splines-based estimator appears to be much more influenced by the number of considered bins.

While the considerations presented in this section are based on heuristic results, they remain quite informative about the actual behaviour of traditional MI estimators. Most of them are somehow based on a partitioning of the observation space, or a continuous extension of such a partioning. These methods clearly cannot deal with high-dimensional data for practical problems where the number of available samples is

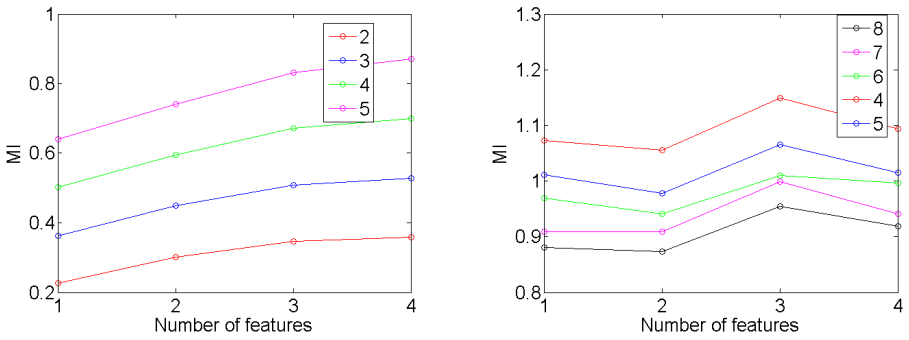


Figure 2.5: Estimated MI for different values of the estimators parameter. Splines-based estimator (left), Nearest neighbours-based estimator (right).

limited. Nearest neighbours-based approaches, as illustrated here through the proposal in [114], are actually better suited for high-dimensional datasets, as the ones encountered in the field of feature selection. This claim is supported by different works in which the estimator in [114] is successfully used for feature selection [149, 150].

2.2 Laplacian score

We now introduce the Laplacian score (LS), a feature selection criterion first introduced in the unsupervised concept [93].

2.2.1 Definitions

The basic idea behind Laplacian score is to somehow define a similarity matrix between the samples of a dataset. Then, the feature relevance is a measure of how much the samples of a feature are coherent with the similarity matrix. Close samples according to the similarity matrix thus have to be close for relevant features, while the contrary is true for samples far away from each other. As will be made clear in the following of this section, one of the major advantages of the LS and related approaches is the freedom to build the similarity or proximity matrix. It can for instance involve the target outputs if available, the samples only or a combination of both. Such approaches are therefore particularly well suited to address semi-supervised or weakly-supervised problems, as will be further illustrated in this thesis. We begin by introducing the unsupervised criterion.

Formally, consider a data set of n samples $\mathbf{x_{i*}}$, each described by d features. Let x_{ir} denote the value of the r^{th} feature of the i^{th} sample ($i = 1 \dots n$) and $\mathbf{x_{*r}}$ the r^{th} feature

of all samples. Build then a similarity graph consisting of n nodes (one for each data sample), which contains an edge between node i and node j if the samples $\mathbf{x_{i*}}$ and $\mathbf{x_{j*}}$ are close. While closennes could be defined in many ways, it will denote here the fact that $\mathbf{x_{i*}}$ is among the k nearest neighbors of $\mathbf{x_{j*}}$ or conversely. Even if any measure of similarity between the samples can theoretically be used, the Euclidean distance will be considered here to identify the nearest neighbors of each sample.

Based on the similarity graph, a similarity matrix S can be built by setting

$$S_{i,j} = \begin{cases} e^{-\frac{\left\|\mathbf{x_{i*}}-\mathbf{x_{j*}}\right\|_2^2}{t}} & \text{if } \mathbf{x_{i*}} \text{ and } \mathbf{x_{j*}} \text{ are close} \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

where t is a positive constant, whose value can be set freely. The next step is to define $D = diag(S\mathbf{1})$, with $\mathbf{1} = [1 \dots 1]^T$, and eventually the graph Laplacian $L = D - S$ [31]. In practice, it is not mandatory to have a zero similarity for pairs of samples whose one is not among the k nearest neighbors of the other, especially since the similarity decreases exponentially with the distance. However, using such a cut-off still enforces the local aspect of the approach and leads to sparse similarity matrices.

It is then important to remove the mean of each feature, possibly weighted by the local density of data points: the new features are called $\tilde{\mathbf{x}}_{*\mathbf{r}}$ and are given by

$$\tilde{\mathbf{x}}_{*\mathbf{r}} = \mathbf{x}_{*\mathbf{r}} - \frac{\mathbf{x_{*r}^T} D\mathbf{1}}{\mathbf{1^T} D\mathbf{1}}\mathbf{1}. \tag{2.25}$$

While a weighted normalisation has a natural interpretation in the field of spectral graph theory [93, 31], traditional normalisation can be used as well, as suggested in [93]. The end of this section gives a detailed argument for the need of normalization.

Eventually the Laplacian score of each feature $\mathbf{x}_{*\mathbf{r}}$ is computed as follows:

$$L_r = \frac{\tilde{\mathbf{x}}_{*\mathbf{r}}^T L \tilde{\mathbf{x}}_{*\mathbf{r}}}{\tilde{\mathbf{x}}_{*\mathbf{r}}^T D \tilde{\mathbf{x}}_{*\mathbf{r}}}, \tag{2.26}$$

and this score is used to rank the features in increasing order. In its original form, LS is thus a univariate criterion. In Section 4.3, a proposal to develop a multivariate criterion based on LS is introduced. It is also interesting to note that the authors of [93] derive a connection between the LS (2.26) and the well-known Fisher criterion.

We now give some brief comments on the reasons why normalisation (2.25) is necessary. The first goal in removing the (weighted) mean is actually to prevent a non-zero constant feature such as $\mathbf{1}$ to be assigned a zero Laplacian score. This is important in practice since, as mentioned above, a low Laplacian score corresponds to a relevant feature. Indeed, a constant feature obviously does not contain any infor-

mation about the output but would be given a zero LS. As detailed in [93], after the proposed normalisation, the numerator of (2.26) cannot be zero for a constant feature if the denominator is not zero. If both the numerator and the denominator are equal to zero, the score of the feature is thus equal to $\frac{0}{0}$, which is an indetermined quantity. It is then possible to detect the feature and to remove it from the problem. While detecting constant features is not a hard task in practice, the main interest of the proposed normalisation is to ensure that the score of the features is not influenced by their range. Indeed, according to Eq. (2.26), the LS directly depends on the feature values. It is thus important to make sure that a good LS actually translates a real relevance of the feature. As the feature range can also negatively impact the computation of the distance between the samples, we actually remove the mean of the features before building the similarity matrix S.

2.2.2 Justification or alternative derivation

In this section, we show an alternative derivation of the Laplacian score; this derivation heuristically justify its use for feature selection.

Let us first mention that the denominator $\tilde{\mathbf{x}}_{*\mathbf{r}}^T D \tilde{\mathbf{x}}_{*\mathbf{r}}$ in (2.26), is the weighted variance of the feature $\mathbf{x}_{*\mathbf{r}}$. Features with a low variance are thus penalized, as the variance is believed to measure the predictive power of a feature.

Then, as explained above, the basic idea of LS is to favour features which are coherent with a previously defined similarity matrix. In the context of unsupervised learning, this matrix is defined by the Euclidean distance between two samples, and the objective is thus to find features which are the most compatible with this proximity measure representing the structure of the data. In other contexts, the similarity matrix could as well translate the fact that two samples belong to the same class or that their associated continuous target outputs are close to each other. A quite natural criterion for which a relevant feature will be low is thus the following one:

$$\min_r \sum_i \sum_j (x_{ir} - x_{jr})^2 S_{ij}. \tag{2.27}$$

Indeed, following Eq. (2.27), if S_{ij} becomes large, $(x_{ri} - x_{rj})^2$ has to decrease for the criterion (2.27) to remain small (remember that criterion (2.26) has to be minimised). Features according to which the samples $\mathbf{x}_{\mathbf{i}*}$ and $\mathbf{x}_{\mathbf{j}*}$ are not close while they actually are in the sense of S_{ij} are thus penalized. The local structure of the data can therefore be preserved. Indeed, a negative exponential is used in the definition (2.26), meaning that for close samples $\mathbf{x}_{\mathbf{i}*}$ and $\mathbf{x}_{\mathbf{j}*}$, S_{ij} will be high.

We now explicitely establish the connection between criteria (2.26) and (2.27). Based on the definition of the diagonal matrix $D = diag(S)\mathbf{1}$, it is easy to deduce that

$D_{ii} = \sum_j S_{ij}$. Remembering that the Laplacian of a graph is defined as $L = D - S$, some basic developments give

$$\begin{aligned}
\sum_i \sum_j (x_{ir} - x_{jr})^2 S_{ij} &= \sum_i \sum_j (x_{ir}^2 + x_{jr}^2 - 2x_{ir}x_{jr}) S_{ij} \\
&= \sum_i \sum_j x_{ir}^2 S_{ij} + \sum_i \sum_j x_{jr}^2 S_{ij} \\
&\quad - 2 \sum_i \sum_j x_{ir} x_{jr} S_{ij} \\
&= 2\, \mathbf{x}_{*\mathbf{r}}^{\mathbf{T}} D\, \mathbf{x}_{*\mathbf{r}} - 2\, \mathbf{x}_{*\mathbf{r}}^{\mathbf{T}} S\, \mathbf{x}_{*\mathbf{r}} \\
&= 2\, \mathbf{x}_{*\mathbf{r}}^{\mathbf{T}} (D - S)\, \mathbf{x}_{*\mathbf{r}} \\
&= 2\, \mathbf{x}_{*\mathbf{r}}^{\mathbf{T}} L\, \mathbf{x}_{*\mathbf{r}}.
\end{aligned} \tag{2.28}$$

Therefore it clearly appears that minimising the numerator $\tilde{\mathbf{x}}_{*\mathbf{r}}^{T} L\, \tilde{\mathbf{x}}_{*\mathbf{r}}$ of (2.26) is equivalent to minimising (2.27); both expressions are indeed equal to a multiplicative factor. Also note that in [199], the authors presents the Laplacian score as a particular case of a more general feature selection algorithm based on spectral graph theory. More generally, they propose an interesting unifying framework for supervised and unsupervised spectral feature selection.

2.3 Hilbert-Schmidt independence criterion

The last feature selection criterion we present is the Hilbert-Schmidt independence criterion (HSIC) introduced in [83]. As illustrated in this section, it is a kernel-based independence criterion which can handle a very large range of data types, making it particularly suitable for feature selection with unconventional or non-standard data. The idea of using kernel-based approaches to measure dependence is actually quite popular in the literature. The general principle of these methods is to define covariance or cross-covariance operators in reproducing kernel Hilbert spaces and to derive statistics from those operators which are well suited to measure the dependence between functions in these spaces. For instance, in [9], the authors introduce the kernel canonical correlation analysis (CCA), a non-linear extension of the canonical correlation analysis, defined using positive-definite kernels. The idea is to extract the information shared by two random variables by finding non-linear mappings $\phi(X)$ and $\psi(X)$, respectively belonging to the reproducing kernel Hilbert spaces $\mathcal{F}$ and $\mathcal{G}$, which maximize their correlation. Kernel CCA can be written in terms of a cross-covariance operator (which will be defined in the next section), making its theoretical analysis easier. From this operator, a regularized corelation operator is derived, whose largest singular value has associated eigenfunctions that are the solutions of the kernel CCA

problem. In another related work, the largest singular value of the cross-covariance operator is used as a measure of independence [84]. The HSIC extends the work in [84] by considering the whole spectrum of the cross-covariance estimator, in order to get a more robust indication of (in)dependence between the variables. To this end, the sum of the squared singular values of the cross-covariance operator, corresponding to the Hilbert-Schmidt norm of this operator, is used as a measure of dependence. Experiments presented in [83] illustrate the superiority of the HSIC regarding previous approaches. An important related work is [77], where the authors also use covariance and cross-covariance operators in RKHSs but do not consider the Hilbert-Schmidt norm. Other related works include [95, 18], which deal with covariance operators instead of cross-covariance operators (the difference being that a cross-covariance operator defines a mapping from one space to another one).

2.3.1 Definitions

Let us again consider two domains $\mathcal{X}$ and $\mathcal{Y}$, which can contain any data type such as (a vector of) real values, strings, graphs, class labels... From these domains, we draw couples of samples (x,y) and we consider a mapping function $\phi(x) \in \mathcal{F}$ from any $x \in \mathcal{X}$, where $\mathcal{F}$ is a feature space such that the inner product $\langle \phi(x), \phi(x') \rangle$ is given by a kernel $k_{\mathcal{X}}(x,x')$. $\mathcal{F}$ is then called a reproducing kernel Hilbert space (RKHS). A second RKHS $\mathcal{G}$ can be defined on $\mathcal{Y}$, with a kernel $k_{\mathcal{Y}}(y,y')$ and a feature map $\psi(y) \in \mathcal{G}$.

We begin by defining the Hilbert-Schmidt (HS) norm of a linear operator $C : \mathcal{G} \longmapsto \mathcal{F}$, provided the sum converges, as:

$$\|C\|_{HS}^2 := \sum_{i,j} \langle Cv_i, u_j \rangle_{\mathcal{F}}^2 , \tag{2.29}$$

where u_j and v_i are elements of orthonormal bases of $\mathcal{F}$ and $\mathcal{G}$ respectively and where $\langle .,. \rangle_{\mathcal{F}}$ denotes the inner product of $\mathcal{F}$. This norm actually generalizes the Frobenius norm on matrices.

We can also define a tensor product operator $f \otimes g : \mathcal{G} \longmapsto \mathcal{F}$ as

$$f \otimes g := f \langle g,h \rangle_{\mathcal{G}} \; \forall h \in \mathcal{G}. \tag{2.30}$$

Following Eq. (2.29), the HS norm of $f \otimes g$ can be computed as

$$\begin{aligned} \|f \otimes g\|_{HS}^2 &= \langle f \otimes g, f \otimes g \rangle_{\mathcal{HS}} = \langle f, (f \otimes g) g \rangle_{\mathcal{F}} \\ &= \langle f,f \rangle_{\mathcal{F}} \langle g,g \rangle_{\mathcal{G}} = \|f\|_{\mathcal{F}}^2 \|g\|_{\mathcal{G}}^2 . \end{aligned} \tag{2.31}$$

Let us also define the mean element $\mu_x \in \mathcal{F}$ as follows:

$$\langle \mu_x, f \rangle_{\mathcal{F}} := \mathop{\mathbb{E}}_{x} \left[\langle \phi(x), f \rangle_{\mathcal{F}} \right] = \mathop{\mathbb{E}}_{x} [f(x)], \tag{2.32}$$

while $\mu_y \in \mathcal{G}$ can be defined similarly. It is therefore possible to compute

$$\|\mu_x\|_F^2 = \mathop{\mathbb{E}}_{x,x'} \left[\langle \phi(x), \phi(x') \rangle_{\mathcal{F}} \right] = \mathop{\mathbb{E}}_{x,x'} \left[k(x,x') \right]. \tag{2.33}$$

Let us note that μ_x and μ_y can be proven to exist and to be unique using Riesz's representation theorem [145], as long as their respective norms in $\mathcal{F}$ and $\mathcal{G}$ are bounded; this condtion is verified when the kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are bounded.

It is now possible to define a (linear) cross-covariance operator $\mathcal{C}_{xy}$ between these features maps as:

$$\mathcal{C}_{xy} : \mathcal{G} \longmapsto \mathcal{F} := \mathop{\mathbb{E}}_{xy} [(\phi(x) - \mu_x) \otimes (\psi(y) - \mu_y)] = Exy\phi(x) \otimes \psi(y) - \mu_x \otimes \mu_y. \tag{2.34}$$

Alternatively, $\mathcal{C}_{xy}$ can be defined as the only operator satisfying [166]

$$\langle \mathcal{C}_{xy}\psi, \phi \rangle_{\mathcal{F}} = \mathop{\mathbb{E}}_{xy} [(\phi(x) - \mu(x))(\psi(y) - \mu(y))]. \tag{2.35}$$

The cross-covariance operator thus expresses the covariance between functions in the RKHS; it also contains all the information regarding the dependence of X and Y that can be expressed by nonlinear functions in the RKHS [78].

The Hilbert-Schmidt independence criterion is formally defined as the square of the Hilbert-Schmidt norm of the cross-covariance estimator [166]:

$$HSIC(\mathcal{F}, \mathcal{G}, P_{xy}) = \|\mathcal{C}_{xy}\|_{HS}^2, \tag{2.36}$$

with P_{xy} being the joint distribution of (x, y).

To ease the computation and the approximation of the HSIC, it is useful to rewrite it only in terms of the kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$. Following the developments in [83] it can be proven that:

$$\begin{aligned} HSIC(\mathcal{F}, \mathcal{G}, P_{xy}) = & \mathop{\mathbb{E}}_{x,x',y,y'} \left[k_{\mathcal{X}}(x,x') k_{\mathcal{Y}}(y,y') \right] + \mathop{\mathbb{E}}_{x,x'} \left[k_{\mathcal{X}(x,x')} \right] \mathop{\mathbb{E}}_{y,y'} \left[k_{\mathcal{Y}(y,y')} \right] \\ & - 2 \mathop{\mathbb{E}}_{x,y} \left[\mathop{\mathbb{E}}_{x'} \left[k_{\mathcal{X}(x,x')} \right] \mathop{\mathbb{E}}_{y'} \left[k_{\mathcal{Y}(y,y')} \right] \right]. \end{aligned} \tag{2.37}$$

Indeed, using Eqs. (2.31) and (2.34), it is possible to write

$$\begin{aligned}\|\mathcal{C}_{xy}\|^2_{HS} &= \left\langle \mathop{\mathbb{E}}_{xy}[\phi(x)\otimes\psi(y)] - \mu_x\otimes\mu_y, \mathop{\mathbb{E}}_{xy}[\phi(x)\otimes\psi(y)] - \mu_x\otimes\mu_y \right\rangle_{\mathcal{HS}} \\ &= \mathop{\mathbb{E}}_{x,y,x',y'}\left[\left\langle \phi(x)\otimes\psi(y), \phi(x')\otimes\psi(y')\right\rangle_{\mathcal{HS}}\right] \\ &\quad - 2\mathop{\mathbb{E}}_{x,y}\left[\left\langle \mu_x\otimes\mu_y, \phi(x)\otimes\psi(y)\right\rangle_{\mathcal{HS}}\right] + \left\langle \mu_x\otimes\mu_y, \mu_x\otimes\mu_y\right\rangle_{\mathcal{HS}}.\end{aligned} \tag{2.38}$$

By substituting the definiton of μ_x and μ_y and by using the fact that $\langle\phi(x),\phi(x')\rangle_{\mathcal{F}} = k(x,x')$, it is possible to obtain Eq. (2.37).

A remarkable property of the HSIC criterion is that if $\mathcal{F}$ and $\mathcal{G}$ are RKHSs with universal kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ as introduced in [169] on the compact spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively, then $HSIC(\mathcal{F},\mathcal{G},P_{xy}) = 0$ if and only if x and y are independent. Examples of universal kernels include the Gaussian or the Laplacian kernels, which can thus be used to detect dependencies between $\mathcal{X}$ and $\mathcal{Y}$. While they will not guarantee that all dependencies will be detected, other (non-universal) kernels can as well be used. By defining them on strings, graphs or other domains, one will thus be able to address a large variety of complex non-standard problems. In [83], the authors experimentally show the advantage of the HSIC over competing methods in the measurement of dependency between variables.

The use of HSIC as a feature selection criterion has first been advocated in [166]. In this work, the authors propose to combine the HSIC with the backward search procedure described above. The idea is to compute only once the kernel for the target output and to evaluate the kernel for the tested features as required by the search procedure, before estimating the HSIC. The authors use a Gaussian kernel for the features as well as for the continuous target outptut of regression problems. They rather apply a linear kernel on the labels for classification problems.

2.3.2 Estimation

In order for the HSIC to be a useful and practical criterion to evaluate the relevance of a (set of) features, it is important to be able to correctly approximate it given a finite sample size. Let us consider a dataset $D := \{(x_i,y_i) \subseteq \mathcal{X}\times\mathcal{Y}, (i = 1\ldots n)\}$, where the n observations are independent and assumed to be drawn from distribution P_{xy}. An estimator of $HSIC(\mathcal{F},\mathcal{G},P_{xy})$, written as $HSIC(\mathcal{F},\mathcal{G},D)$ can then be given as [83]:

$$HSIC(\mathcal{F},\mathcal{G},D) = (n-1)^{-2}\,\mathbf{tr}\left(K^{\mathcal{X}} H K^{\mathcal{Y}} H\right). \tag{2.39}$$

In Eq. (2.39), $K^{\mathcal{X}}_{ij} = k_{\mathcal{X}}(x_i, x_j)$, $K^{\mathcal{Y}}_{ij} = k_{\mathcal{Y}}(y_i, y_j)$ and $H_{ij} = \delta_{ij} - n^{-1}$, with δ being the Kronecker delta.

As pointed out in [83], the computational complexity of the estimator $HSIC(\mathcal{F}, \mathcal{G}, D)$ is $O(n^2)$, which is less than most of other kernel dependence criteria. However, in practice, good approximations to all such criteria can be obtained with a simlar complexity (see e.g. [10]). Another important property of this estimator concerns its bias. Indeed, it can be shown that, considering again the i.i.d. couples of samples $(x_i, y_i) \sim P_{xy}$ that compose the dataset D [83]:

$$HSIC(\mathcal{F}, \mathcal{G}, P_{xy}) = \mathop{\mathbb{E}}_{D}[HSIC(\mathcal{F}, \mathcal{G}, D)] + O(m^{-1}). \tag{2.40}$$

This result actually means that if the variance of $HSIC(\mathcal{F}, \mathcal{G}, D)$ is larger than $O(m^{-1})$, the bias induced by the estimation (2.39) can be considered as neglectable. It is also important to note that assuming non-negative and bounded almost everywhere by 1 kernels, tight bounds on the difference between the estimated and real values of HSIC can be given, showing again the interest of the estimators [83].

2.4 Discussion

We end this chapter with a small discussion on the choice of three particular criteria considered in the thesis.

Obviously, and as already detailed in Section 1.1.1, other possible filter quality criteria were available as the basis of our further developments. The rationale behind the choice of the three criteria (MI, LS and HSIC) is the following one. First, the mutual information has mainly been chosen because of the existence of an estimator able to handle multivariate data and therefore to score feature subsets, both for regression and classification problems. Besides, mutual information is a non-linear criterion, able to detect any kind of dependency between random variables. Then, Laplacian score and related methods are particularly well designed for both weakly supervised and non-standard problems. Indeed, such methods are only based on the determination of a similarity matrix between samples, which can either take supervised samples into account or not. Eventually, we believe the HSIC has a great potential for feature selection in non-standard settings. Indeed, it only requires the determination of a kernel function in both the input and the target output space. As kernels can be defined for many kinds of data (discrete and/or continuous, strings, graphs...), a large amount of problems can potentially be addressed. Besides, HSIC is naturally defined for multivariate data.

The three considered criteria thus possess interesting and complementary properties for feature selection, especially in the context of non-standard data.

For comparison, the very popular correlation coefficient is univariate and limited to the detection of linear dependencies, while the Relief algorithm is univariate and limited to regression problems. Both approaches seem therefore less promising in terms of potential extensions.

Chapter 3

Datasets

We now gather in Table 3.1 all the datasets used in the experiments throughout the thesis; we then briefly introduce them. First, some datasets correspond to regression problems, i.e. problems for which the output to predict is a continuous (real) value. Nitrogen[1] and Orange juice[2] are spectroscopic datasets. As preprocessing, each spectrum of the 1050 original spectrum in the Nitrogen dataset has been represented using its coordinates in a B-splines basis, in order to reduce the amount of features to a reasonable number of 105 (see [149]). For the Delve census dataset[3], only the first 2048 samples have been kept. The mortage dataset can be downloaded from the website of the Federal reserve bank of Saint-Louis [4] while the Housing dataset is available from the UCI website [71]. Then, all classification datasets, except for the multi-label ones, can as well be found on the UCI website [71], where they are presented in details. Eventually, the multi-label datasets can all be downloaded from the website of the Mulan project[5] . In Table 2, the number in brackets denotes the maximum number of potential labels for each data point.

All but the Corel, Enron and Medical datasets are have continuous variables, while the three remaining ones are discrete.

The experimental setting and the way the datasets are used can differ according to the considered problem. Indications on the exact use of the datasets are therefore given when necessary in the corresponding chapter.

Also note that two datasets are called *Yeast*. However, the context will make clear the one that is referred to in the following chapters.

[1] http://kerouac.pharm.uky.edu/asrg/cnirs/

[2] http://mlg.info.ucl.ac.be/index.php?page=DataBases

[3] http://www.cs.toronto.edu/ delve/data/census-house/desc.html

[4] http://research.stlouisfed.org/

[5] http://mulan.sourceforge.net/

	Sample size	Features	Type
Delve Census	2048	104	Regression
Housing	506	13	Regression
Mortage	1049	16	Regression
Nitrogen	141	105	Regression
Orange Juice	218	700	Regression
Breast Tissue	106	10	6-class
Ecoli	336	8	8-class
Glass Identification	214	10	7-class
Image Segmentation	2310	19	7-class
Ionosphere	351	34	2-class
Iris	150	4	3-class
Mines vs. Rocks	208	60	2-class
Page Blocks Classification	5473	10	5-class
Handwritten Digits	10992	16	10-class
Seeds	210	7	3-class
Vehicle Silhouettes	946	18	4-class
Vertebral Column	310	6	4-class
Wall-Following Robot	5456	24	4-class
Waveform	5000	21	3-class
Wine	178	13	3-class
Wisconsin diagnostic Breast Cancer (wdbc)	569	32	2-class
Yeast	1484	8	10-class
Corel	5000	499	multi-label (374)
Enron	1702	1001	multi-label (53)
Medical	978	1149	multi-label (45)
Scene	2407	294	multi-label (6)
Yeast	2417	103	multi-label (14)

Table 3.1: Datasets used in the thesis.

Part II

Feature selection algorithms for complex settings

Chapter 4

Feature selection for semi-supervised learning

Contents

This chapter deals with feature selection in the context of semi-supervised problems. Such problems are characterized by the small amount of output values available regarding the number of samples in the dataset. Semi-supervised learning thus differs from the traditional supervised learning, where the knowledge of the output target value is assumed for every sample in the data set. It also differs from the unsupervised learning paradigm, where no target value is considered and which is thus unrelated to any prediction task.

Semi-supervised learning thus lies between these two extreme situations and actually corresponds to a quite realistic problem in practice. Indeed, it is common for real-world problems that generating lots of samples is an easy task, while obtaining the target value of interest can turn out to be costly, time-consuming and error prone. As an example, consider a practician whose goal is to detect if a patient suffers (or will likely suffer) from a given disease based on analyses or on radiographies. As most patients of an hospital undergo lots of tests and analyses, building a dataset is quite simple. However, most of the patients will not be diagnosed for the pathology of interest. Besides, determining the presence of a certain disease can be a hard task, even for a trained practician. In another field, it is sometimes important in a food quality control context to measure some properties of a food sample (as e.g. the fat rate for meat, the sugar rate for juice or the alcohol degree for wine). Spectroscopic properties of the samples can be used to help automating such predictions (see e.g. [150]). However, building supervised training sets and obtaining the real values of interest can be costly in practice, as destructive tests (such as burning the piece of meat) and chemical analyses often have to be performed.

These two examples illustrate the need to adress the semi-supervised learning problem in the machine learning literature, and more precisely in the context of feature selection. The objective is often to improve algorithms based on the unsupervised part of the data by using the knowledge of the few available labels; two reference books on semi-supervised learning are [28, 202].

While many recent works have adressed the feature selection problem for semi-supervised datasets, they are focused on classification problems. Regression problems have therefore not been considered. This chapter thus first addresses this lack by proposing an approach based on the Laplacian Score (LS) introduced in Section 2.2. The developments are presented in Section 4.2 and are based on the works [51, 56].

As the feature selection criteria based on the LS and on spectral theory are mainly univariate, a solution is proposed in Section 4.3 to develop a multivariate criterion. The method is based on the Hilbert-Schmidt independence criterion (see Section 2.3) and on the possibility to define kernels on graph Laplacians. It can be applied to both regression and classification problems, and actually to a broader range of problems, as will be illustrated. The results have not been published yet.

Prior to these developments, the chapter begins with a brief state-of-the-art about semi-supervised feature selection.

4.1 Related work

Various solutions for feature selection have been proposed in the particular context of semi-supervised learning. Among many others, Zhao and Liu proposed a very popular approach using spectral analysis [199]. The idea is basically to turn the features into cluster indicators for the min-cut clustering method [162]. The quality of the features is then measured by their ability to produce clusters which are (i) well-separated and (ii) coherent with the few available class-labels. Quinzan et al. introduced an algorithm based on feature clustering, conditional mutual information and conditional entropy [142]. In [26], the authors use the notion of Markov blanket. In [184], the authors propose to solve a class margin maximization problem involving a manifold regularization term, which allows the user to take the structure of the data into account. Globally, features are selected according to their class-margin maximisation power and their ability to preserve the structure of the data. In [200], Zhong et al. introduce a *hybrid* method that selects an initial set of features using the supervised samples, before expending and correcting this set using the unsupervised samples and label propagation techniques. In [196], the concept of graph Laplacian is also exploited, similarly to what is done in the next section. Sample proximity matrices are defined through the class memberships if available or through the distance between samples if one of the two samples is not supervised. Again, a compromise is thus seeked between features that are in good agreement with the few class labels and with the global structure of the (unsupervised) data. Eventually, let us also mention the work in [146], which belongs to the wrapper category. The idea is to perform a greedy forward search to select at each step the feature whose addition leads to the better classification performances. To evaluate these performances, a random set of unsupervised samples are first given a class label, evaluated by the prediction algorithm. This procedure is repeated many times, leading to a very time-consuming method.

As previously mentioned, one common aspect of all the aforementioned works is that they are designed for classification problems, and that there exists no obvious way of extending them for regression problems. One of the main reasons for that is that class labels naturally define a cluster structure in the data, while this is not the case for regression problems. Besides, the notions of class-margin or label propagation have no obvious counterpart in regression problems.

While this chapter is dedicated to semi-supervised problems, some works on feature selection for unsupervised or weakly-supervised problems also have to be mentioned. First, the most obvious and simple example of unsupervised method is simply

the evaluation of each feature variance, which gives an indication of the features predictive power, as already explained in Section 2.2.

However, more interesting unsupervised methods rather focus on (i) finding subsets of features that lead to the determination of sound clusters of samples or (ii) removing redundancy between features. For instance, in [134] the authors introduce a feature similarity measurement called maximum information compression index and perform feature clustering using this criterion; they then remove all but one features of the most compact cluster and proceed recursively. Other works include for instance [59], which is based on expectation-maximisation clustering or [130] where it is also proposed to use the dependency between features to eliminate redundant ones. Of course, the Graph Laplacian based ranking method proposed in [93] and presented in details in Section 2.2 is another popular approach on which are based the subsequent developments in this chapter. An important more recent work is [25], where the basic idea is again to find features preserving the best the (cluster) structure of the data. The authors first propose to compute a graph Laplacian L before solving the generalized eigenvalue problem $L\mathbf{Y} = \lambda D\mathbf{Y}$, where D has the same meaning as in Eq. (2.26). Each row of $\mathbf{Y}$ actually corresponds to a flat embedding of a data point or, said otherwise, to the unfolded data point. The K eigenvectors Y_{*k} with the largest eigenvalues (coresponding to the K first columns of $\mathbf{Y}$) are then kept, where K ideally corresponds to the number of clusters in the data. For each eigenvector, a LASSO-like problem is solved:

$$\min_{a_k} \left\| Y_{*k} - X^T a_k \right\|^2 + \beta \left| a_k \right|,$$

where X is the original dataset. The maximum of the absolute value of the coefficient corresponding to each feature is determined across the K LASSO-like problems. The features are then ranked according to this value, in decreasing order.

Let us eventually mention another form of weak supervision closely related to the semi-supervised problem: pairwise constraints [190]. The idea is that the class labels are not available, but that it is known, for a limited amount of pairs of samples, whether or not they belong to the same class. An approach based on a graph Laplacian, very similar to the work in [196] for semi-supervised problems has been proposed to address feature selection for problems with pairwise constraints.

4.2 Semi-supervised Laplacian score

This section introduces the Semi-Supervised Laplacian Score (SSLS), following the developments in [51, 56]. SSLS is actually a semi-supervised feature selection criterion, designed for regression problems. The first part of this section will be dedicated to the derivation of a supervised criterion, which will then be used to produce the

semi-supervised one.

4.2.1 Supervised Laplacian score

While the goal of the Supervised Laplacian score (SLS) is essentially to serve as an intermediate step in the derivation of the SSLS, it possesses by itself interesting properties for feature selection, as illustrated below.

Let us consider again a dataset described by n samples $\mathbf{x_{i*}}$ and by d features $\mathbf{x_{*i}}$. A target output vector $Y = [y_1 \dots y_n] \in \Re^m$ is also given. Assuming that the outputs y_i are given by a continuous and smooth function of the samples $\mathbf{x_{i*}}$ (possibly limited to a subset of features), it is quite natural to expect that for close samples $\mathbf{x_{i*}}$ and $\mathbf{x_{j*}}$, the output values y_i and y_j are close too. In that regard, we can define good features as ones having close values for data points whose outputs are close too. Let us note that this assumption is quite common in machine learning and that many popular regression models including the k nearest neighbors and the radial basis functions networks are based on a similar idea. Based on this principle, and keeping in mind Eq. (2.28) and the justifications in Section 2.2.2, a feature selection criterion can be constructed following the next guidelines.

Define a similarity matrix S^{sup} as

$$S_{i,j}^{sup} = \begin{cases} e^{-\frac{(y_i - y_j)^2}{t}} & \text{if } \mathbf{x_{i*}} \text{ and } \mathbf{x_{j*}} \text{ are close} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

and $D^{sup} = diag(S^{sup}\mathbf{1})$, $L^{sup} = D^{sup} - S^{sup}$, $\tilde{\mathbf{x}}_{*\mathbf{r}} = \mathbf{x}_{*\mathbf{r}} - \frac{\mathbf{x}_{*\mathbf{r}}^{\mathbf{T}} D^{sup}\mathbf{1}}{\mathbf{1}^{\mathbf{T}} D^{sup}\mathbf{1}}\mathbf{1}$. In eq. (4.1), two points $\mathbf{x_{i*}}$ and $\mathbf{x_{j*}}$ are considered as being close if one is among the k nearest neighbors of the other, while t is a suitable (positive) constant.

The Supervised Laplacian Score (SLS) can then be defined as follows:

$$SLS_r = \frac{\tilde{\mathbf{x}}_{*\mathbf{r}}^T L^{sup} \tilde{\mathbf{x}}_{*\mathbf{r}}}{\tilde{\mathbf{x}}_{*\mathbf{r}}^T D^{sup} \tilde{\mathbf{x}}_{*\mathbf{r}}}. \tag{4.2}$$

It can be used to rank the features, by sorting them in decreasing order of their score SLS_r.

Illustration

In order to illustrate the interest of SLS, it is briefly compared to two other feature selection criteria: the correlation coefficient and the univariate MI. From a practical point of view, in this section, the value of the parameter k in (4.1) is set to 5 and the MI is estimated using the nearest neighbors-based estimator in [114] (see Eq. (2.19)).

	Y_1	Y_2
Correlation	100	32
SLS	100	100
MI	100	100

Table 4.1: Percentage of experiments for which the relevant features are ranked first on two artificial problems.

Artificial datasets SLS is first tested on two artificial problems for which the relevant features are known in advance.

The first problem has 6 features $X_1 \dots X_6$ uniformly distributed on $[0;1]$. Its output is a linear combination of three features

$$Y_1 = 5X_1 + 7X_2 - 10X_3. \tag{4.3}$$

Obviously, only features X_1, X_2 and X_3 are informative.

The second problem consists of 4 features $X_1 \dots X_4$ uniformly distributed on $[0;1]$, where only the two first ones are useful to build the output, since it is defined as

$$Y_2 = X_1^2 X_2^{-2}. \tag{4.4}$$

In both problems, the sample size is 1000 and 1000 datasets are randomly generated. The criterion of comparison is the percentage of the experiments where the informative features are the best ranked. The results are summarized in Table 4.1. Obviously, when the relationship between the features and the output is non-linear the proposed SLS performs much better than the correlation coefficient, which is restricted to linear dependencies. SLS is equivalent to the MI for this problem. For the linear problem (4.3), all the three criteria perform equally well.

Real-world datasets We also test SLS on real-world data sets. For such datasets, a prediction model has to be used to assess the performances of the feature selection criteria. Here, a 5-nearest neighbors (5-NN) model is considered because it is both simple to use and sensitive to irrelevant features. Two data sets are studied, Delve Census and Orange Juice (see Chapter 3 for details).

For the two datasets, Figures 4.1 and 4.2 show the root mean squared error (RMSE) (estimated through a 5-fold cross-validation procedure) of a 5-NN model as a function of the number of selected features. For comparison, the results with the unsupervised Laplacian score are also presented. The interest of the SLS against the correlation coefficient is again illustrated, while as could be expected, the performances of the unsupervised method are the worst. MI outperforms its competitors for the Juice dataset and is equivalent to the SLS for the Delve dataset. However, as will be illustrated

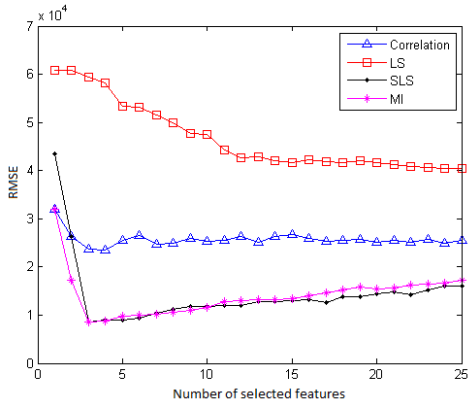


Figure 4.1: RMSE of a 5-NN model for 4 feature selection criteria on the Delve Census data set.

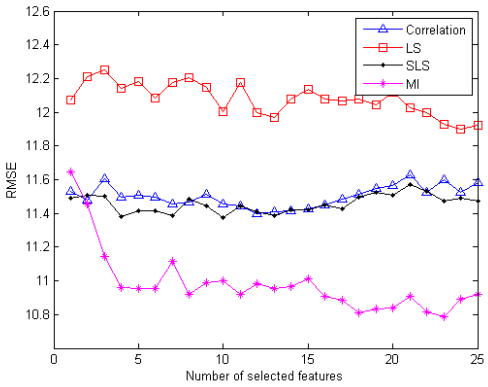


Figure 4.2: RMSE of a 5-NN model for 4 feature selection criteria on the Orange Juice data set.

below, the performances of the MI greatly decrease when the number of labelled samples is low; the performances of the SLS are then comparable to the ones of the MI. A possible explanation is that because of its complexity, the MI criterion requires a consequent amount of samples for being reliably estimated.

4.2.2 Semi-supervised Laplacian score

We now introduce a semi-supervised feature selection algorithm based on the SLS; such an algorithm is necessary in practice as taking the unlabeled samples into account is particularly important when the number of labeled samples is low. As previously explained, the LS and the SLS are both based on the idea of preserving the local structure of the data. The difference between both methods only lies in the way we define *local structure*, using the data samples, the target outputs or both. An intuitive idea is to compute the distance between two samples from their outputs if both are known, and from the unsupervised part of the data otherwise. This way, the important supervision information is always taken into account, while the strucutre of the data is also considered to help dealing with the small number of supervised samples.

Let us consider a semi-supervised regression problem, where a data set of n samples is considered, only s of them being supervised: $Y = [y_1 \dots y_s] \in \Re^s$. We futhermore assume that $s << m$.

The developments begin by defining the pairwise distance matrix d:

$$d_{i,j}^2 = \begin{cases} (y_i - y_j)^2 & \text{if } y_i \text{ and } y_j \text{ are known} \\ \frac{1}{n}\sum_{k=1}^{n}(f_{k,i} - f_{k,j})^2 & \text{otherwise.} \end{cases} \tag{4.5}$$

In the second case, the distance is normalized by the number of features n to keep every value of d^2 in a comparable range. This would not be the case otherwise, since the dimension of the data set is generally much larger than one.

Based on d^2, a matrix S^{semi} is then built as follows

$$S_{i,j}^{semi} = \begin{cases} e^{-\frac{d_{i,j}^2}{t}} & \text{if } \mathbf{x_{i*}} \text{ and } \mathbf{x_{j*}} \text{ are close and } y_i \text{ and/or } y_j \text{ is unknown,} \\ C \times e^{-\frac{d_{i,j}^2}{t}} & \text{if } \mathbf{x_{i*}} \text{ and } \mathbf{x_{j*}} \text{ are close and } y_i \text{ and } y_j \text{ are known,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

Again, two points are close if one is among the k nearest neighbors of the other one.

A positive constant C has been introduced in Eq. (4.6) to weight the values of $S_{i,j}^{semi}$ corresponding to two supervised samples. Since the SLS has been shown to outperform the unsupervised LS (see Figures 4.1 and 4.2), it actually seems reasonable to assume that the labels are in practice more important than the unsupervised samples

and to give them more importance in determination of relevant features.

We then define $D^{semi} = diag(S^{semi}\mathbf{1})$, $L^{semi} = D^{semi} - S^{semi}$ and $\tilde{\mathbf{f}}_{*\mathbf{r}} = \mathbf{x}_{*\mathbf{r}} - \frac{\mathbf{x}_{*\mathbf{r}}^{\mathbf{T}} D^{semi}\mathbf{1}}{\mathbf{1}^{\mathbf{T}} D^{semi}\mathbf{1}}\mathbf{1}$.

The Semi-Supervised Laplacian Score (SSLS) is eventually defined for each feature $\mathbf{x}_{*\mathbf{r}}$ as

$$SSLS_r = \frac{\tilde{\mathbf{x}}_{*\mathbf{r}}^T L^{semi} \tilde{\mathbf{x}}_{*\mathbf{r}}}{\tilde{\mathbf{x}}_{*\mathbf{r}}^T D^{semi} \tilde{\mathbf{x}}_{*\mathbf{r}}} \times SLS_r. \tag{4.7}$$

In Eq. (4.7), SLS_r is the Supervised Laplacian Score computed using the supervised samples only.

The proposed criterion thus takes both the unsupervised and the supervised part of the data into account, giving the supervised part more importance. The SSLS can actually be seen as the SLS corrected with a term slightly influenced by values of the data samples. Experimental results will show that even such a small influence can greatly improve the feature selection performances compared to the SLS. Both terms are actually important in Eq. 4.7 since the semi-supervised term alone is not guaranteed to fully use all the available information provided by the supervised samples.

Experimental results and discussion

We now illustrate the interest of the SSLS criterion through two different axes. First, the performances of a prediction model using the features selected by SSLS are studied. Then, we ensure that SSLS efficiently uses the few supervised samples.

Prediction performances While we only focus on the two same datasets as the ones used for the SLS (see Section 4.2.1), more results can be found in the corresponding journal paper [56]. The SSLS is compared with the MI, the correlation coefficient and the SLS. The results with the unsupervised LS are not reproduced here, as they do not compete with those of the supervised and semi-supervised methods. Again, the feature selection criteria are compared through the RMSE of a 5-NN model. A few supervised samples are first randomly selected from the training set, based on which the MI, the correlation coefficient and the SLS are computed. All samples are used for the SSLS. For the prediction step, all samples in the training set are considered to be supervised because the model would perform badly with a low number of supervised samples, making the results unmeaningful. The compared algorithms are tested with 3% and 5% of randomly selected supervised samples. The RMSE is estimated through a 5-fold cross-test procedure repeated 10 times. The different parameters are set as follows. The value of t is set to 1 while 5 neighbors are considered for computing the unsupervised (2.24) and supervised (4.2) score. 30 neighbors are considered for S^{semi} (4.6) in order to increase the probability of taking supervised samples into account. C in Eq. (4.6) is set to the moderate value of 5, giving significant weight to supervised

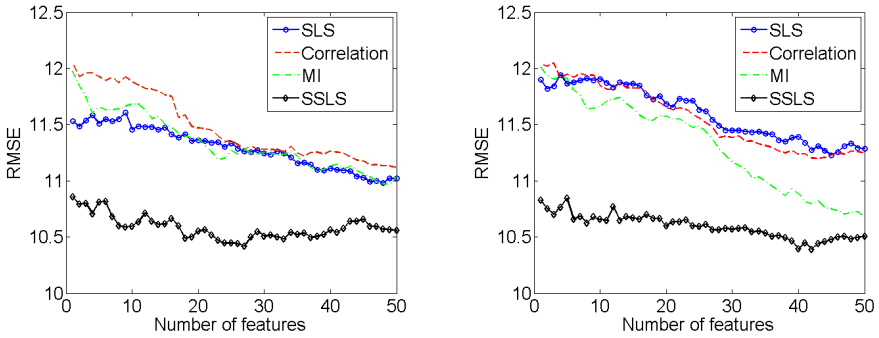


Figure 4.3: RMSE of a 5-NN model as a function of the number of selected features with 3% (left) and 5% (right) supervised samples for the Orange Juice data set.

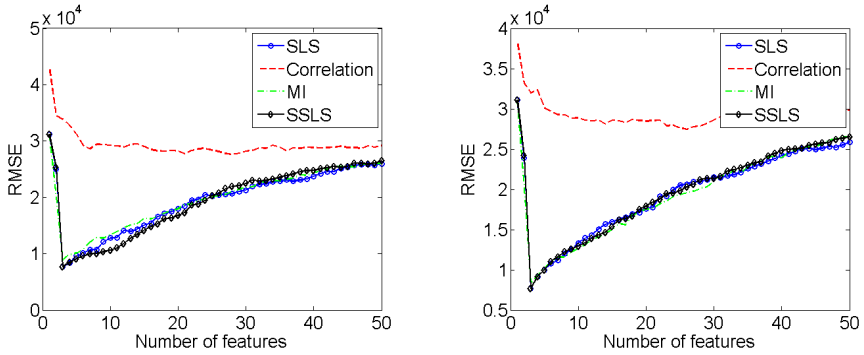


Figure 4.4: RMSE of a 5-NN model as a function of the number of selected features with 3% (left) and 5% (right) supervised samples for the Delve Census data set.

samples but nevertheless allowing one to take the information coming from the unsupervised data points into account. Before any distance computation, the features and the output vector are normalized by removing their mean and dividing them by their standard deviation. For the prediction step, the outputs are not normalized.

Figures 4.3 and 4.4 illustrate the results. First, the semi-supervised criterion outperforms its supervised counterpart for the Juice dataset, while the performances are identical for the Delve dataset. These observations and further results in [56] indicate the interest of considering the information coming from the unsupervised part of the data. It also looks like the MI performs quite bad when given a small number of supervised samples (the Delve dataset is much larger than the Juice dataset in terms of the number of samples). In this regard, the SSLS appears to be a possible alternative. Indeed, the SSLS also performs much better than the correlation coefficient. The SSLS thus seems to have an advantage over both the unsupervised and supervised compared approaches to feature selection for semi-supervised problems.

Efficient use of the supervised samples Another property a good semi-supervised algorithm should possess is the capacity to efficiently exploit the precious and rare information given by the labelled samples. It is thus interesting to see how the performances of the SSLS depend on the number of supervised samples, especially when a very small amount of them is available. Figure 4.5 shows the results on the Delve dataset. As can be seen, the SSLS clearly benefits from the addition of supervised samples, as further illustrated in [56]. It is reasonable to assume that this observation will remain true only as long as the number of supervised samples remains small and that the behaviour of both the SSLS and the SLS will tend to become similar when the amount of supervised samples increase. It is worth noting that the proposed methodology has interest only if the supervised samples can be assumed to be distributed similarly to the unsupervised ones, i.e. if the samples are i.i.d. Indeed, otherwise, the feature relevance computed from the supervised samples would be meaningless for the unsupervised ones.

Discussion on the parameters

This section ends with a few comments on the different parameters involved in the definiton of the SSLS.

First, 5 neighbors are used to build the the proximity matrices for the unsupervised and supervised scores, in order for the graph to accurately reflect the local structure of the data. This value is suggested in related works such as [93] and [196]. For the semi-supervised similarity matrix (4.6), the number of neighbors is increased to 30, while in practice any intermediate value could as well be used. Experiments using 50

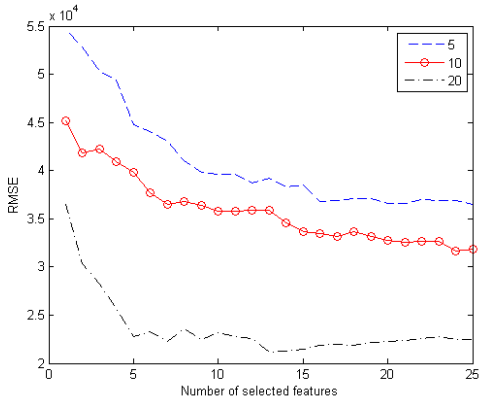


Figure 4.5: Performances of a 5-NN model as a function of the number of features selected with the SSLS criterion and different numbers of supervised samples on the Delve Census data set.

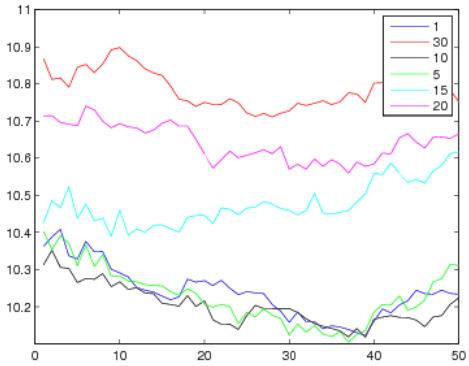


Figure 4.6: Performances of a 5-NN model as a function of the number of features selected with the SSLS criterion and different values of the parameter C on the Orange Juice data set.

to 100 neighbors confirmed this intuition with non significant differences observed in the final performances.

The parameter t in Equations (2.24), (4.1) and (4.6) has actually mainly been introduced to be consistent with the notations of the original paper about the LS [93]. We have however set it to 1 because (i) the data we deal with are assumed to have zero mean and unit variance and (ii) a low number of neighbors are used to build the proximity matrix and using a quickly decaying exponential function to represent the local structure of the data is not needed.

Eventually, C gives more weight to the supervised information in S^{semi} (4.6). This parameter can prove useful in practice but experiments have shown that its value only slightly affects the results. To illustrate this, Figure 4.6 presents the performances of the SSLS for 10 supervised samples and different values of C. When the value of C remains low, the performances of the SSLS remain close. On the contrary, when the value of C increases, the performances of the SSLS become similar to the ones of the SLS. The value of C can thus be set to 1 in practice and this parameter can be ignored. However, when the proportion of supervised samples is extremely low (e.g. less than 1%), we rather suggest a moderate value of $C = 5$.

4.3 Multivariate LS-based criteria

As explained above and as can be understood from the presentation of the SLS and SSLS, feature selection criteria based on the graph Laplacian are traditionally univariate, which can be a serious drawback in practice. A possible solution to this problem is the method proposed in [25] and already discussed in Section 4.1. Indeed, even if originally designed for unsupervised data, the similarity matrix on which the graph Laplacian is built can equally be constructed using target outputs if available. However, one of the main limitations of the method in [25] is that it cannot be used to evaluate a subset of features as it rather solves an optimisation problem and outputs what it considers as the best feature subset of a given cardinality. For instance, it cannot been used in greedy forward or backward searches where subsets of features are comapred at each step. In practice, this limitation can restrict the application of the method to criteria which involve more than one similarity matrix and graph Laplacian, as it is made clear below.

For instance, as mentioned above, the constraint score [190] assumes that class labels are not known but that it is known for a certain couple of samples whether they belong to the same class (there is then a must-link constraint between these points) or if they belong to different classes (there is then a cannot-linnk constraint). Two similarity matrices S^{ml} and S^{cl} are built such that S^{ml}_{ij} (S^{cl}_{ij}) is 0 if there exists a must-link (cannot-link) constraint between the i^{th} and j^{th} samples. Two graph Laplacians (L^{ml} and L^{cl})

are then built following the usual procedure. Two variants of the univariate feature selection criterion exist:

$$C_r^1 = \frac{\mathbf{x}_{*\mathbf{r}}{}^T L^{ml} \mathbf{x}_{*\mathbf{r}}}{\mathbf{x}_{*\mathbf{r}}{}^T L^{cl} \mathbf{x}_{*\mathbf{r}}}, \tag{4.8}$$

and

$$C_r^2 = \mathbf{x}_{*\mathbf{r}}{}^T L^{ml} \mathbf{x}_{*\mathbf{r}} - \lambda \mathbf{x}_{*\mathbf{r}}{}^T L^{cl} \mathbf{x}_{*\mathbf{r}}. \tag{4.9}$$

The idea of these two criteria is to favour features for which samples with a must-link constraint are close and samples with a cannot-link constraint are far from each other.

The criterion in (4.9) can be reduced to $C_r^2 = \mathbf{x}_{*\mathbf{r}}{}^T (L^{ml} - \lambda \mathbf{L}^{cl}) \mathbf{x}_{*\mathbf{r}}$ and can therefore be plugged into the framework proposed by [25]. On the contrary, it is impossible to summarize the criterion (4.8) using a single Laplacian matrix. Therefore, the approach in [25] cannot be adapted to measure the relevance of a subset of features regarding criterion 4.8, as it involves a quotient of Laplacian Scores.

The solution we propose to produce a multivariate criterion generalising the criterion (4.8) is to use the HSIC (see Section 2.3). Indeed, it has been shown in [165] how kernels can be defined on graph Laplacians based on regularization functions on graphs. As an example, if $\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ is a normalized graph Laplacian, than a p-step random walk kernel can be defined as:

$$K_L = (aI - \tilde{L})^p, \tag{4.10}$$

where a and p are parameters to be set. Following the guidelines in [165] we fixed their value to 2 and 4 respectively.

Using such a Laplacian kernel as the target output kernel K_L and the Gaussian kernel as the samples kernel K_X, it is possible to use the HSIC feature selection criterion presented in Section 2.3 in a backward search procedure. At each step of the procedure, $HSIC(K_X, K_L^{ml}, D)$ as well as $HSIC(K_X, K_L^{cl}, D)$ are evaluated for all the features subsets that have to be tested. The feature whose removal leads to the largest value of $\frac{HSIC(K_X, K_L^{ml}, D)}{HSIC(K_X, K_L^{cl}, D)}$ is then eliminated.

Another criterion for which the proposed methodology can be proven to be useful is the semi-supervised feature selection method for classification problems introduced in [196]. In this work, two similarity matrices are also introduced. The between-class matrix S_b is such that $S_{b,ij}$ is 1 if the i^{th} and the j^{th} samples have different labels, and 0 otherwise. The within-class matrix S_w is such that $S_{w,ij}$ is γ if the samples have the same label, 1 if one or both samples are unlabeled but that they are close and 0 otherwise. The finall criterion is:

$$L_r^{semi} = \frac{\mathbf{x}_{*\mathbf{r}}{}^T L^b \mathbf{x}_{*\mathbf{r}}}{\mathbf{x}_{*\mathbf{r}}{}^T L^w \mathbf{x}_{*\mathbf{r}}}, \tag{4.11}$$

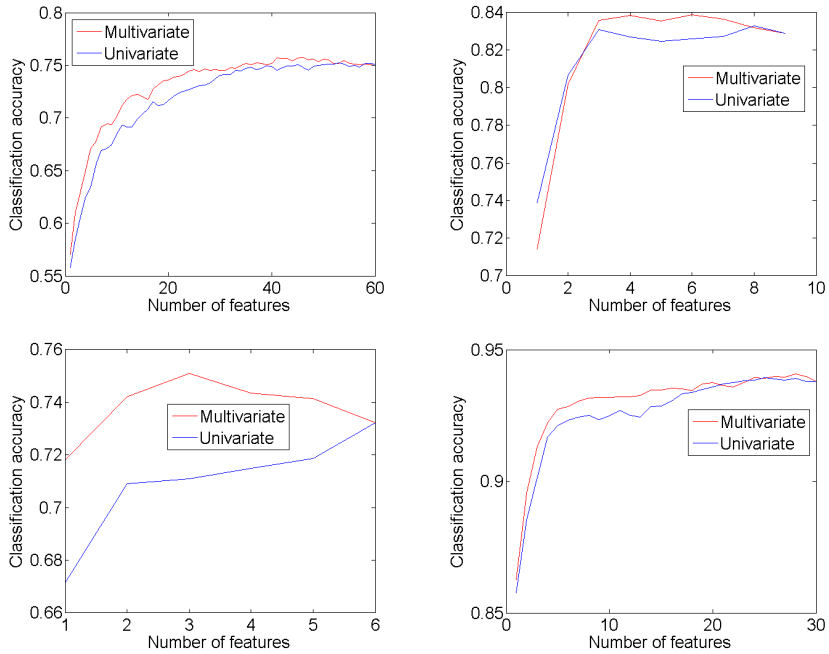


Figure 4.7: Accuracy of a nearest-neighbor classifier with features selected by the Constraint Score and multivariate version of it. From top to bottom and left to right: Mines vs. rocks, Breast Tissue, Vertebral Column and wdbc dataset.

with the exact same interpretation as for (4.8). To illustrate the potential interest of the method we propose, we have studied the performances of a nearest-neighbor classifier using the features selected by criteria (4.8), (4.11) and their respective multivariate counterpart. We present the results for four datasets from the UCI repository: Mines vs. Rocks, Breast Tissue, Vetebral Column and wdbc. The experiments are repeated 50 times. Each time a training set containing 70% of the samples and a test set containing the remaining 30% are randomly drawn from the dataset. 50 constraints are randomly drawn in the training set, while 10 % of the samples are considered as supervised for the semi-supervised problem. Figures 4.7 and 4.8 illustrate the potential interest of the proposed approach, for the two feature selection algorithms. Indeed, even if they are non-significant, large accuracy differences can be observed especially for the Vertebral column dataset. There thus appears to be a real interest in further investigating the suggested methodology, for instance by considering other kernels on graph Laplacians.

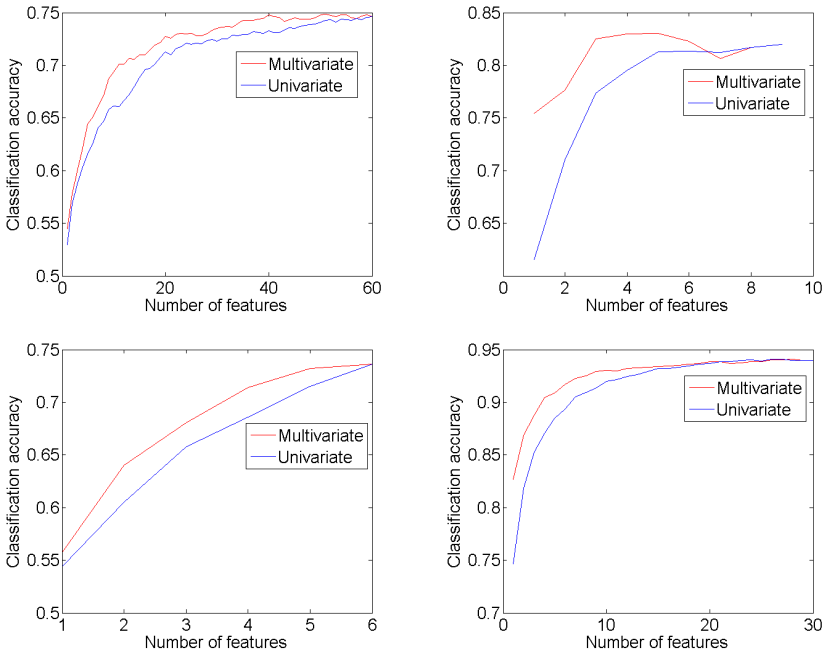


Figure 4.8: Accuracy of a nearest-neighbor classifier with features selected by the semi-supervised algorithm in [196] and a multivariate version of it. From top to bottom and left to right: Mines vs. Rocks, Breast Tissue, Vertebral Column and wdbc dataset.

4.4 Conclusion

This chapter first introduces a feature selection criterion for semi-supervised regression problems, inspired from the well-known Laplacian score and called semi-supervised Laplacian score (SSLS). The idea is to look for features that preserve the local structure of the data, defined either based on the target output if available, or on the unsupervised samples. The second part of this chapter suggests a possible way to extend univariate graph Laplacian-based criteria, extremely popular in practice, to multivariate criteria. To this end, it is proposed to use the HSIC feature selection framework introduced in Section 2.3 and to consider a p-step random walk kernel on a graph Laplacian as the output kernel. First results indicate the approach is promising; it must however be kept in mind that it introduces two additional parameters.

Chapter 5

Feature selection for multi-label learning

Contents

Figure 5.1: Illustration of the multi-label paradigm. Picture belonging to both beach and mountain categories.

This chapter tackles the problem of feature selection for multi-label datasets. By opposition to more conventional single-lable classification problems, the multi-label paradigm assumes that each data sample in a dataset can be simultaneaously associated to multiple classes. The samples thus really belong to different classes and multi-label learning is not related to any kind of uncertainty. One of the most illustrative examples of application is the categorization of pictures or texts, where one is interested in the detection of more than one topic or characteristic. In newspapper articles categorization, for instance, one could be interested in both the main topic of the text (sport, politics...) and the place where the events took place. Similarly, in picture classification, one could be interested in determining the scene of the picture [20]. In this context, Figure 5 can be associated to both beach and mountain categories.

The problem of multi-label classification can actually be seen a generalization of the single-label one; it has been extensively studied recently due to its interest in many domains. Besides the aforementioned classification of visual scenes [20] and text categorization [156], these domains also include classification of music into emotions [173] or protein function classification [45].

The next section gives a brief overview of state-of-the-art approaches to classification and feature selection for multi-label learning. A feature selection algorithm for continuous problems is then presented in Section 5.2; a way to tune its parameter is proposed in Section 5.2.1 and its performances are evaluated in Section 5.2.2. A multivariate algorithm for discrete datasets is also introduced in Section 5.3. This chapter is based on the works [49, 48] as well as on unpublished results.

5.1 Related work

Two main directions are usually considered to tackle multi-label learning. One consists in adapting existing classification algorithms to the multi-label paradigm. Among the most popular examples of this strategy, let us mention extensions of AdaBoost [156], support vector machines [62], C4.5 [32] and K nearest neighbors [194]. Adaptation can de done through the use of a multi-label errror criterion [62], the design of a multi-label entropy-like criterion [32] or more heuristic procedures.

The other popular approach is to transform the considered multi-label classification problem into one or several single-label problems; each subproblem can then be addressed using any existing method. The simplest transformation method that can be thought of is the binary relevance (BR). The idea is to learn a separate classification model for each possible label. As many prediction models as the number of possible labels are thus created this way and each of these models predict whether or not a sample belongs to a specific class. The prediction of each classifier is independent of the prediction of the other classifiers. Eventually, the predicted label set for each sample is obtained by combining the decisions of all the classifiers. One important dawback of this approach is that BR does not take into account the potential dependence that could exist between the class labels, since the decision for each label is made individually. One possible solution to this problem is to see each unique combination of labels in a training set as a label for a single-label classification algorithm. This approach, introduced in [144], is called label power set (LP). A problem is that the number of class labels created this way can potentially be very high (up to 2^c, where c is the original number of labels). Read et al. [144] thus suggested to prune the problem by getting rid of the classes represented by a too small number of samples after the problem transformation. Points belonging to these classes can be removed or can be assigned to one or several other class(es). This methodology is called pruned problem transformation (PPT) [144].

Concerning the feature selection problem we are interested in, it has only been addressed in few works. One of the most popular and straightforward approaches is to measure the relevance of each feature for each of the labels independently using a χ^2 statistics [173, 30]. The global relevance of the features can then be obtained by somehow combining the scores relative to the different labels. This methodology has essentially been used for text categorization [30]. The χ^2 statistics has also been considered in combination with the LP transformation rather than the BR as in [30]. This combination showed promising results in the field of music classification [173] and has been shown to outperform the work in [30]. The main reason of this superiority is the fact that the LP takes the label dependency into account.

The two aforementioned approaches suffer from two important restrictions. First,

the χ^2 statistics is defined for discrete features. It is thus necessary to discretize them if the training set contains continuous variables. This obviously leads to a loss of information and makes the results of the feature selection highly dependent of the chosen discretization procedure. A criterion designed for continuous variables could therefore prove very useful. Then, the χ^2 statistics is a univariate relevance criterion that scores features individually. Neither jointly relevant nor jointly redundant features can therefore be detected. It is therefore important to propose multivariate feature selection criteria, which could detect jointly relevant or redundant features.

Other related works include [118], where a graphical model and the symmetrical uncertainty are used to represent the relationships among the labels and the features; it is designed for discrete datasets only. In [121], the authors decompose and approximate the multivariate MI as a sum of (conditional) entropies involving subsets of variables of lower dimensionality. Again, discrete datasets are required. In [193], it is proposed to combine PCA-based feature extraction with a wrapper feature selection method using a genetic algorithm. This approach is rather a feature extraction than a feature selection method, as the original features are transformed and combined.

Let us also mention that some works consider other information than a simple dependency between the class labels. For instance, in the context of gene classifications, it can be pertinent to introduce a hierarchical structure between the labels, meaning that some labels cannot be associated with a sample if some other ones are not. Such considerations are for instance adressed in [13, 175]. A very comprehensive survey covering all aspects of multi-label learning can be found in [192].

5.2 MI-based feature selection with continuous datasets

Because it is naturally able to both deal with continuous features and consider multivariate variables, MI is well-suited to address the drawbacks of most current multi-label feature selection algorithms.

Let us consider a dataset $D \in \Re^{n \times d}$, whith d being the original number of features. Let us also consider a multi-label output vector $O \in \{0,1\}^{n \times c}$, whith c being the number of possible labels. Each column of O thus codes for the belonging of the samples to one of the c classes.

Let us first transform the problem using the PPT approach. Every unique combination of class labels in O is thus considered as a single class label. Points belonging to created classes containing less than p samples are then removed from the dataset. The transformed dataset and output vector are denoted D_t and O_t, respectively. D_t contains $n_t \leq n$ samples described by the original number of features d.

As mentioned above, it is possible to keep the points whose class is shared by less than p samples by duplicating them and assigning each copy a new class label, chosen

from the labels still existing in the dataset after the pruning [144]. However, as we will work with the nearest neighbors-based MI estimator designed for classification problems (2.21), this solution is not acceptable. Indeed, this would induce the existence of samples described by the exact same values but belonging to different classes. This would of course confuse the determination of the class label of the k^{th} nearest neighbor of the samples and hurt the MI estimator.

Once the original problem has been transformed to a single-label one, a forward search using the MI criterion is considered to select relevant features. It is stopped after a predetermined number of features has been selected or when another stopping criterion is met. Because the MI is estimated between a subset of features and the output, the above strategy is able to consider the possible interaction or redundancy between features, which is a great advantage over most existing methods, as already mentioned.

The benefits of the problem transformation are two-fold for the proposed methodology. First, the PPT eases the problem by limiting the possible number of classes. It thus prevents the MI estimator from suffering from the presence of many rare classes, which do not accurately represent the original dataset. In this regard, the PPT prevents from a certain kind overfitting. Then, the PPT ensures that each class of the transformed dataset is described by a minimum number of p samples. This is of great importance since the considered MI estimator requires that the distance between each sample and its K^{th} nearest neighbor in the same class is known. Controlling the pruning parameter p allows us to make sure that $K < p$, as required. This would obviously not be the case for the BR transformation for which no guarantee can be given regarding the minimal cardinality of the classes.

5.2.1 Determination of the pruning parameter

The proposed feature selection algorithm can be seen to have only one parameter (in addition to the one of the MI estimator): the pruning parameter p corresponding to the minimum number of samples belonging to a class after the PPT procedure. This section is dedicated to the presentation of a heuristic procedure to determine it. The procedure is based on the work in [70], where a way to set the k parameter of the nearest neighbors-based MI estimator is proposed.

Intuitively, a good value of p should lead to a a compromise between two requirements that may seem quite incompatible. First, it is important to preserve the information contained in the data as much as possible, which is equivalent to choosing a relatively low value for p. Indeed, selecting a too high value for p would mean removing many of the original samples. It could make the transformed dataset uninformative and therefore useless, as it would not represent anymore the original dataset.

Only a few leading classes and therefore only the global trends of the dataset would actually still be represented.

Secondly,feature selection algorithms will probably not perform well in the presence of many classes containing very few samples. Indeed, it can reasonably be assumed that most of such small classes actually represent extremely rare events which are probably not relevant for the considered problem and which can easily lead to overfitting.

A quite natural idea to set the value of p is to consider the feature selection criterion, i.e. the MI in our case. We therefore look for a value of p for which the MI (i) could efficiently use as much relevant information as possible to determine the important features and (ii) would not be harmed by too many small classes. Said otherwise, we are interested in values of p for which the MI criterion will be the most able to discriminate between relevant and irrelevant features. One solution is to look for values of p which will best differentiate the MI estimated between the output and relevant features and the MI estimated between the output and irrelavant features. Following the work in [70], it is proposed to compare the distribution of $I(x^t_{*i};O_t)$ with the distribution $I(U;O_t)$, where x^t_{*i} is the transformed i^{th} feature and U denotes a useless feature. The value of p which best separates these distributions can be chosen. This procedure has quite an intuitive justification. If there are too many classes with a low number of samples (in an extreme situation, each sample could be assigned a different class label), the labels will lose any kind of meaning and there will be no significant difference between a relevant and an irrelevant feature. The same will be true if only a small number of classes with many samples exist (only one class in an extreme situation), since the dataset would then contain a very limited amount of information.

Of course, in practice, the distribution of $I(x^t_{*i};O_t)$ is unknown for any feature, while an irrelevant feature still has to be defined. We therefore suggest to use a permutation test combined with a k-fold procedure to estimate the mean and the variance of the MI estimator and thus the distribution $I(x^t_{*i};O_t)$. Following the guidelines in [70], the idea is the following. For a feature x^t_{*i} assumed to be relevant, the distribution of $I(x^t_{*i};O_t)$ and of $I(x^{t,\Pi}_{*i};O_t)$ are built, where $x^{t,\Pi}_{*i}$ denotes a randomly permuted version of x^t_{*i}. Because of the random permutation, $x^{t,\Pi}_{*i}$ is assumed to be completely independent from O_t and therefore irrelevant for the classification task. To concretely build the required distributions, 20 estimations of $I(x^t_{*i};O_t)$ and of $I(x^{t,\Pi}_{*i};O_t)$ are made on non-overlapping subsets of the dataset, using a 20-fold cross-validation scheme.

The value of p is then chosen as the one best separating the two distributions in the sense of a Student measure:

$$t^i_x = \frac{\mu_x - \mu^\Pi_x}{\sqrt{\sigma^2_x + (\sigma^\Pi_x)^2}}. \tag{5.1}$$

In Equation (5.1), μ_x and σ_x^2 are respectively the estimated mean and variance of the distribution of $I(x_{*i}^t; O_t)$ obtained with a pruning parameter p equal to x; μ_x^π and $(\sigma_x^\pi)^2$ are the corresponding quantities for the distribution of $I(x_{*i}^{t,\Pi}; O_t)$.

Once t_x^i is obtained for every feature x_{*i}^t and every parameter value x in a chosen range, the best value for p is selected as the one corresponding to the global highest value of t_x^i across all the features. This ensures that useless features are not taken into account in the choice of the value of p, as it would be the case if t_x^i would for instance be averaged over all features. Considering useless features would indeed make no sense in the described procedure. Other strategies are nevertheless possible and one could also consider the m highest values of t_x^i, or the features for which t_x^i is above a fixed threshold. This would however require the introduction of an additional parameter which would need to be tuned.

As mentioned above, the parameter k of the MI estimator could as well be set according to the same methodology. To do so, t_x^i should simply be made dependent to k and become $t_x^i(k)$ (with of course $k < p$). The maximum value of $t_x^i(k)$ over all features would then give the best couple of parameters for (p,k). The computational cost of the procedure would consequently be multiplied by k when compared to the situation where only p has to be determined. In the remaining developments of this chapter, the value of k is set arbitrarily because we mainly focus on the influence of p.

5.2.2 Experimental evaluation

This section gives experimental evidences for the interest of the suggested feature selection methodology. In the present text, we only show results for two real-world classification problems. More extensive results can be read in [48] where experiments on artificial datasets clearly illustrate the ability of the proposed criterion to detect jointly relevant and redundant features.

The proposed methodology is compared to the one introduced in [173] denoted as χ^2. The discretisation needed for the χ^2 approach follows [127]. The parameter k is set to a relatively small value of 4 in the MI estimator, as it is needed that $p > k$. The value of the pruning parameter p is chosen in the range 5 to 20.

The first dataset we consider is called Yeast, whose objective is to associate genes with a set of functional classes. The sample size is 1500 for the training set and 917 for the test set. The second one is the Scene dataset [20], concerned with the semantic classification of pictures. The sample size is 1211 for the training set and 1196 for the test set.

It is important to note that the proposed divisions of the samples into the training and test sets are the ones suggested on the website of the Mulan project[1] where the

[1] http://mulan.sourceforge.net/

datasets can be downloaded from in ARFF format. Such divisions are often used in the multi-label literature.

Performance criteria

Four multi-label performance criteria are considered in this work to evaluate the performances of the proposed feature selection algorithm. Let us call M the number of points in the test set, O_{i*} ($i = 1 \dots M$) the set of true class labels for sample i, $\hat{O}_{i*}$ the set of class labels predicted by a multi-label classifier h for sample i and $\hat{N}_i$, the set of labels non-predicted by the classifier h. Let us also define, for sample i, $r_i(o)$ as the position of label o in the ranking of the most probable labels provided by the classifier.

The Hamming loss is defined as:

$$HL(h,M) = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{1}{c}|O_{i*}\,\Delta\,\hat{O_{i*}}|, \tag{5.2}$$

where Δ denotes the symmetric difference between two sets and c the maximum number of possible labels. The Hamming loss actually measures the number of times a label not associated to a sample is predicted or a label associated to a sample is not predicted.

The second criterion is the accuracy, defined as:

$$Accuracy(h,M) = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{|O_{i*} \cap \hat{O_{i*}}|}{|O_{i*} \cup \hat{O_{i*}}|}. \tag{5.3}$$

For the accuracy to be defined (and not infinite), it is important to make sure that all samples belong to at least one class.

The ranking loss is defined as :

$$RL(h,M) = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{\left|(o,o') \in \hat{O}_{i*} \times \hat{N}_i \mid r_i(o) > r_i(o')\right|}{|\hat{O}_{i*}|\,|\hat{N}_i|} \tag{5.4}$$

and corresponds to the average fraction of pairs of labels which are not correctly ordered.

Finally, coverage is defined as:

$$Co(h,M) = \frac{1}{|M|}\sum_{i=1}^{|M|}\max_{o \in O_{i*}} r_i(o) - 1 \tag{5.5}$$

and corresponds to the mean number of predicted labels required to cover the complete set of true labels.

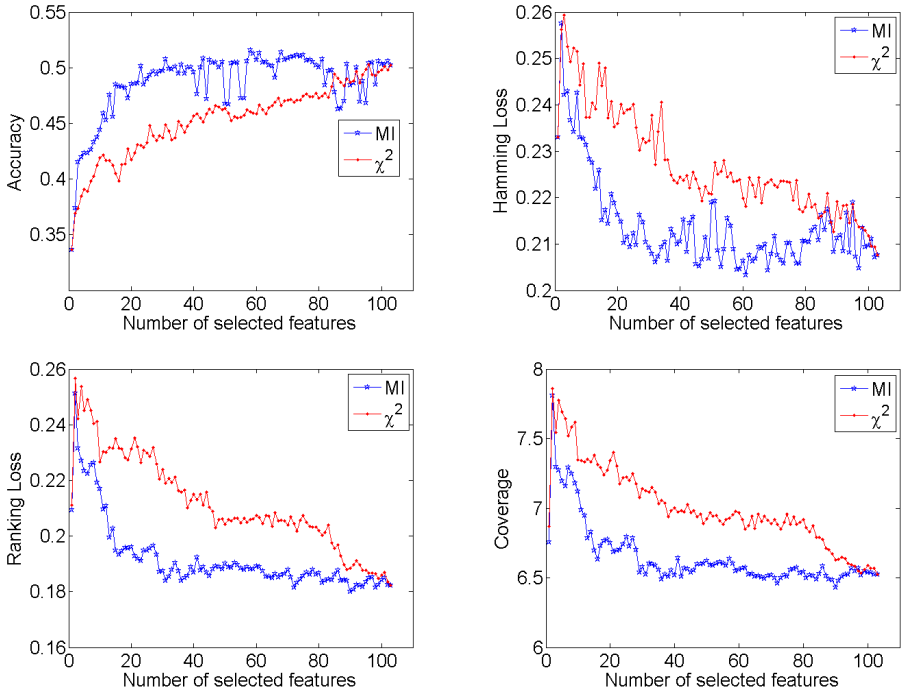


Figure 5.2: Four quality criteria of the k nearest neighbors classifier as a function of the number of selected features for the Yeast dataset.

Results and discussions

To compare the feature selection algorithms, the k nearest neighbors-based multi-label classifier introduced in [194] is used. Its principle is to first identify the k nearest neighbors of the sample to be classified before using the maximum a posteriori principle to predict the label set. This algorithm has been chosen for both its simplicity and its high sensitivity to the presence of irrelevant features.

An important point is that the PPT has only been used for the feature selection step. Once the features have been ranked, the original dataset containing all instances is considered for the classification step. We are then sure to address the same classification problem and to use the same learning set as for the χ^2-based algorithm. The values 8 and 12 have been obtained for the parameter p for the Yeast and Scene datasets, respectively. They lie in the middle of the possible range and therefore confirm this range of tested values is reasonable.

Figures 5.2 and 5.3 show the accuracy, the Hamming loss, the ranking loss and the

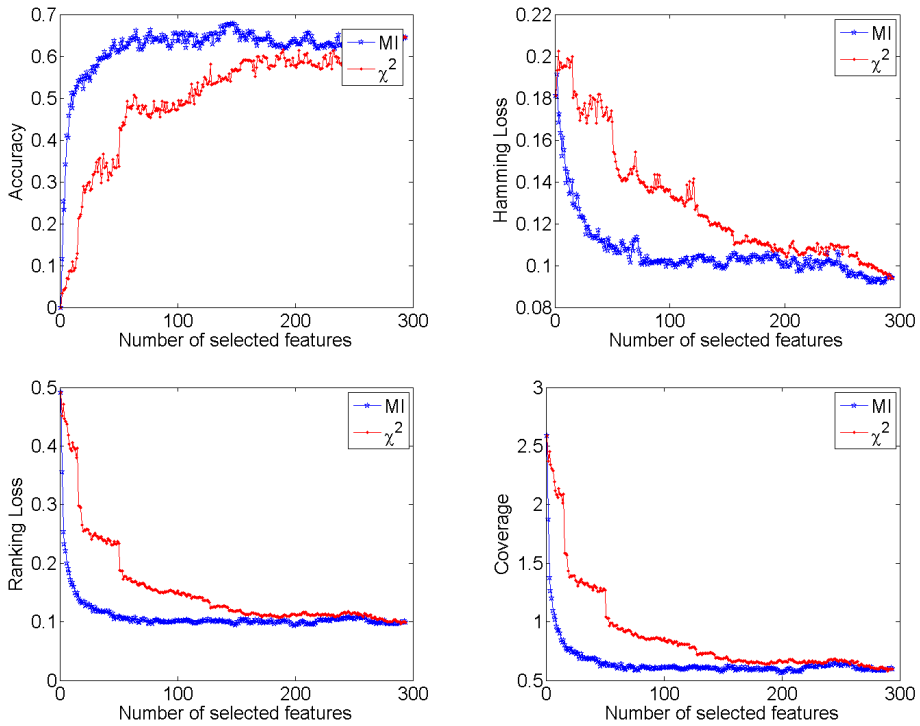


Figure 5.3: Four quality criteria of the k nearest neighbors classifier as a function of the number of selected features for the Scene dataset.

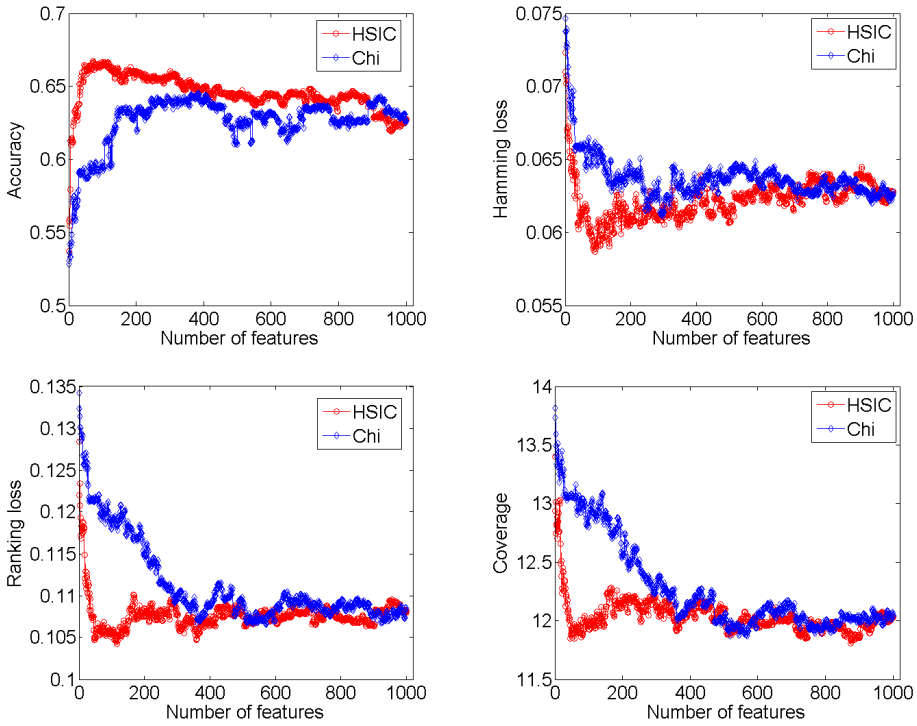


Figure 5.4: Four quality criteria of the k nearest neighbors classifier as a function of the number of selected features for the Enron dataset.

coverage of the ML-KNN classifier as a function of the number of selected features for the two datasets. The results indicate the superiority of the proposed methodology as the MI-based approach (i) leads to improved classification performance for the two datasets and the four criteria, when compared to the case where all features are used, (ii) always leads to the global highest accuracy and smallest Hamming loss, ranking loss and coverage and (iii) leads to better prediction performances than its competitor for a very large majority of the numbers of selected features. Point (i) is particularly important as it demonstrates that the method is effectively able to detect relevant features and to globally improve the whole classification procedure.

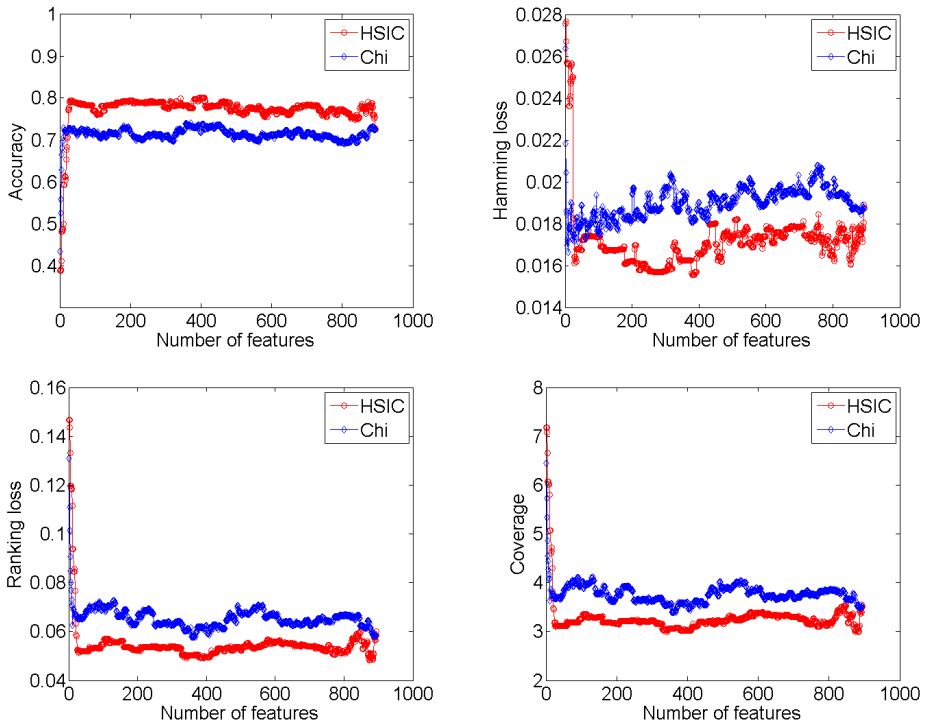


Figure 5.5: Four quality criteria of the k nearest neighbors classifier as a function of the number of selected features for the Medical dataset.

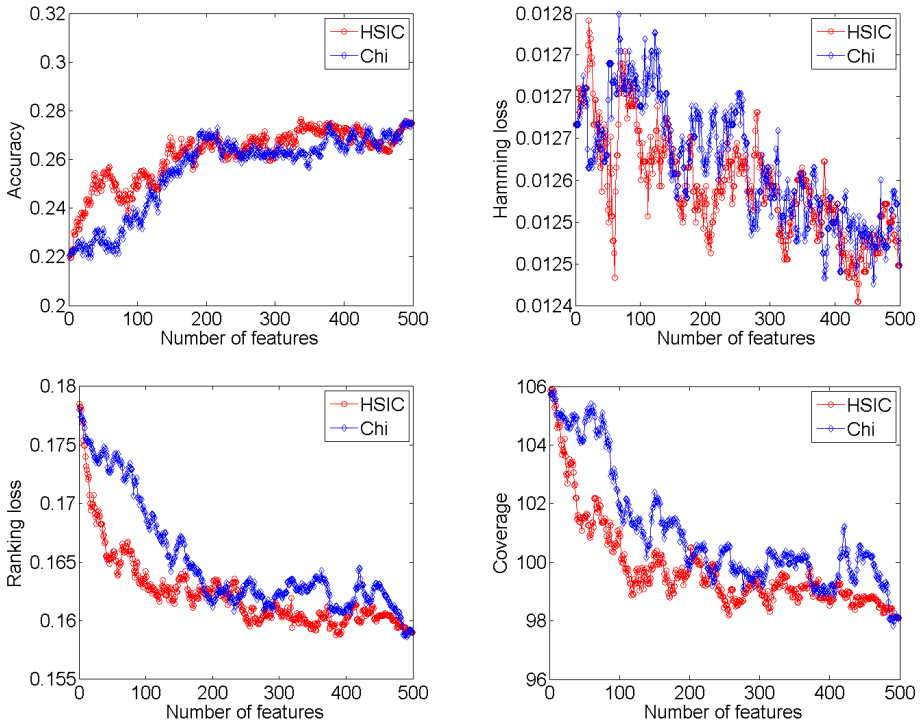


Figure 5.6: Four quality criteria of the k nearest neighbors classifier as a function of the number of selected features for the Corel dataset.

5.3 A multivariate criterion for discrete datasets

As mentioned above, text categorization is one of the main areas for which multi-label learning can prove useful. In such an area, the features are very often discrete, for instance if they correspond to the presence of a particular word in the text. It is thus important in practice to propose feature selection algorithms designed to handle such data.

In this section, we briefly illustrate how the HSIC criterion can be useful in this context. More precisely, we extend the idea presented in [195], where a projection method for dimensionality reduction is proposed. In this paper, Zhang and Zhou propose two possible kernels for the multi-dimensional target output space. The first one consists simply in taking the scalar product of two output vectors and therefore does not take any dependency between the class labels into account.

On the contrary, the second one does allow one to take the label dependencies into account. It is defined as

$$L = O^T BO,$$

where O again denotes the target output matrix. The B matrix is a between samples correlation matrix which can be built in many ways. One solution is to define each entry B_{ij} as the Gaussian kernel of O_{i*} and O_{j*}, the output vectors for samples i and j, respectively. The varaince parameter of the Gaussian kernel can be set to the average squared distance between any pair of samples. The kernel for the sample space can also be defined in the exact same way. Once the two kernels are defined, the HSIC criterion can be used. Here, we combine it with a greedy forward search procedure.

Three discrete datasets are used in the experiments. The Enron and the Medical datasets are both concerned with text categorization while The Corel dataset is a picture classification dataset. The same four quality criteria as in Section 5.2.2 are studied.

Figures 5.5 and 5.6 show the performances of the suggested approach, again compared to the χ^2 method already used in Section 5.2.2. The advantage of the suggested approach seems obvious for the three tested datasets, suggesting that it could effectively be used in practice. Indeed, it consistently leads to better performances than the χ^2 approach. Moreover, it always allows one to improve the classification performances compared with the case all the features are used.

5.4 Conclusion

This chapter is concerned with the development of feature selection algorithms in the context of multi-label learning. An algorithm based on the mutual information and specifically designed for continuous problems is first proposed and successfully tested. It combines a problem transformation strategy taking the label dependency into account and a MI-based greedy search strategy.

An approach for discrete datasets based on the HSIC is then proposed. It requires the determination of a kernel function on both the sample and the output spaces. First results show extremely promising results on the three considered datasets. Further experiments with other choices of kernels could confirm this good behaviour and possibly make the approach much more general, if for instance kernels designed for the hierachical structure described above could be designed.

Chapter 6

Feature selection for missing data

Contents

Missing data is an extremely common problem occuring in many data mining, machine learning or pattern recognition applications [155, 126]. Indeed, there are actually plenty of reasons that can cause a value in a dataset to be missing. For instance, in an industrial context, data can be missing because of the dysfunction of a measure equipment or because a sensor device has a too low resolution to capture the desired information. In the socio-economic domain where data are often collected from surveys and polls, it is very likely that some people will refuse to answer too personal questions, for instance about their income or their political opinions. This will also lead to missing values in the datasets [107]. As an other example, in the medical domain, it is not always possible to conduct all the desired experiments on some patients. The reasons for this can be numerous and include the patients' condition, the fact that they left the hospital or died or simply their age or their sex. Eventually, some data can also be lost or wrongly recorded.

The aforementioned situations illustrate the three different missingness patterns often distinguished in the literature [152] and that we now briefly detail. Let us denote by M the matrix indicator coding for the fact that values are observed or missing. Here, M has thus the same size as the dataset and its elements are 1 or 0, depending if the values are missing or not. Let us then denote the total dataset by $\Delta = \{\Delta_{obs}, \Delta_{mis}\}$ where Δ_{obs} is the observed part of the data and Δ_{mis} is the missing part.

The first missingness mechanism is missing at random (MAR) data, making the assumption that M (or its distribution) does not depend on the (unobserved) values of the missing data:

$$P(M|\Delta) = P(M|\Delta_{obs}). \tag{6.1}$$

In the medical example, this situation corresponds for instance to the fact that a particular experiment has not been realized because the patient is a woman, as can be indicated or not by a feature in the dataset.

A special case of MAR data is *missing completely at random data* (MCAR). This situation, very popular in practice, corresponds to a matrix M independent from both the missing values and from the observed values:

$$P(M|\Delta) = P(M). \tag{6.2}$$

The MCAR assumption is verified for lost data or when an equipment or device unexpectedly stops working. As another example, it is also verified when one voluntarily prevents all the participants of a test from undergoing a huge number of experiments or from answering lots of questions, but randomly selects a few ones for each participant for time or cost reasons [2].

When Equation (6.1) (and consequently Equation (6.2)) does not hold, the data are said to be *missing not at random* (MNAR). For such data, the probability to observe

a missing data depends on Δ_{mis} and can also possibly depend on Δ_{obs}. For example, such a situation happens when people having a too low or a too high income refuse to communicate it. In practice, the MNAR missingness is a much harder problem to address than MAR or MCAR because the missingness distribution has to be carefully modelled. In this chapter, the data are assumed to be MCAR and experiments are conducted accordingly.

It is important to note that the presence of missing values in a dataset does not automatically mean that a great amount of information is lost. However missing data always prevent one from using traditional data mining or machine learning tools, designed for *regular* data. As a revealing example, pairwise sample distances, which are at the core of many algorithms, cannot be computed directly anymore in the presence of missing data. More specifically, for the topic we are interested in, most feature selection algorithms are designed to be used with complete dataset and cannot be trivially adapted to problems with missing data. Of course one simple solution would be to remove all samples (or features) containg missing data, which is obviously unrealistic when the proportion of missing data grows. Another option is to impute the data before applying classical feature selection procedures. However, imputation would add a bias whose effect on feature selection would be really hard to estimate; in practice we thus look for a way to select features independently of any imputation procedure.

The goal of this chapter is thus precisely to propose a way to achieve feature selection in the presence of missing data, without the need for any imputation algorithm. Besides the first motivation explained above, it is also important to note that very few prediction models are actually able to handle missing data. Therefore, if one wants to achieve regression or classification with an originally incomplete dataset, it is very likely that an imputation phase will still be necessary after the feature selection step. In this situation, it appears obvious that useless features would probably harm the imputation procedure and should therefore be removed beforehand. This will be particularly the case when distance-based imputation strategies including the very popular k nearest neighbors imputation and its variants will be considered.

To propose an imputation-free feature selection algorithm, the mutual information (MI) criterion will be used. More precisely, MI will be directly estimated from the dataset, with an adapted version of the k nearest neighbors-based MI estimator [114] introduced in Section 2.1. A more elaborated way to estimate distance between samples with missing values will then be introduced.

The remaining of the chapter is organized as follows. Section 6.1 draws a brief literature review about missing data analysis. The proposed feature selection algorithm is introduced in Section 6.2 and is experimentally tested in Section 6.2.2. Section 6.3 is devoted to the presentation of a distance estimation methodology for data with missing values.

Section 6.2 is based on publications [58] and [55] and Section 6.3 is based on [61].

6.1 Related work

It has already been explained that the missing data problem is frequently encountered in machine learning. It has thus been extensively studied for many different purposes. Three main approaches are generally considered to tackle missing data: case deletion, imputation of the missing values or learning directly with missing data.

Case deletion or complete-case analysis only takes into account the samples for which the values are available for all features, deleting the incomplete ones [141]. As already said, such a strategy can lead to a huge loss of information, especially when there is a high proportion of missing data. On the other hand, the case deletion approach can be used in combination with any traditionnal algorithm designed for complete case problems and does not require developing specific procedures for missing values.

Contrarily, imputation methods try to recover the values which are missing, with the goal of producing complete datasets. Traditional data mining or machine learning methods can then be applied. Those techniques are extremely popular and have been adressed by a huge amount of works in the literature. They have been used successfully in many areas including DNA microarray [174] and environmental [104] data analysis.

The most widely used imputation technique is probably the k nearest neighbors imputation. It consists in finding the k nearest *complete* samples of an incomplete data point. The missing values of this point are then set to the average of the values of the k neighbors for the corresponding samples. It is easy to understand that the performances of this strategy are very limited when the proportion of missing data is high, because only a small amount of samples will then be complete and will serve as the basis for imputation. A simple improvement to this approach is to also look for *incomplete* neighbors of a sample, which are however observed for the features whose value is missing in the sample to impute. This strategy is called incomplete case k nearest neighbors imputation (ICkNNI) [99].

Another basic imputation technique is to replace each missing value with the mean of the observed values for the corresponding feature. More sophisticated imputation methods include an expectation maximization (EM) algorithm and regularized versions of it [157], as well as multiple imputation [151]. However, multiple imputation requires to fit a model to the data and is generally considered as hard to conduct in practice [155]. Least-squares imputation [81, 6] is another possibility which estimates the missing values from a linear prediction model built from the data.

Methods have also been proposed to directly deal with missing data; they do not

require any imputation or deletion phase. For instance, the authors of [183] perform logistic regression with missing data by estimating conditional density functions using a Gaussian mixture model. For clustering purposes, [177] proposed an algorithm called KSC, where the partially observed features are encoded as a set of supplementary soft constraints.

Quite surpisingly, the important problem of feature selection for data with missing values has been barely addressed. Indeed, only two works are, to the best of our knowledge, dedicated to this issue. In [189], Zaffalon *et al.* infer the MI distribution using a second-order Dirichlet prior distribution to conduct feature selection. The developments are limited to the classification problem. In [131], the authors combine feature selection and imputation of missing values. Nevertheless, the feature selection only aims at increasing the performances of a k nearest neighbors imputation algorithm; it is thefore not clear if such a strategy can increase the performances of a prediction algorithm. We refer the interested reader to [126] for a more complete overview of the treatment of missing data.

6.2 A feature selection algorithm for continuous regression problems

This section introduces the proposed methodology to achieve feature selection for regression problems with missing values. It is based on the MI criterion, estimated using the nearest neighbors-based estimator [114]. As detailed in Section 2.1.3, the estimator (2.19) is fully determined by the distances between each sample and its k nearest neighbors in both the input and the target output space.

To estimate the MI using (2.19), imputing the missing values is therefore not required since only the pairwise distances between samples is needed. To estimate these distances, the partial distance strategy (PDS) is used in this work. The quite simple principle of PDS is to compute the distances using the available values only and thus ignoring the missing ones.

To illustrate this idea, let us consider two samples $a, b \in \Re^f$ and let us denote by M_a (M_b) the indices of the missing values for a (b). The distance between samples a and b is:

$$dist(a,b) = \sqrt{\frac{1}{d - |M_a \cup M_b|} \sum_{i \notin M_a \cap M_b} (a_i - b_i)^2}, \tag{6.3}$$

whith $|\,.\,|$ being the cardinality of a set. In Equation (6.3), we have chosen to normalize the estimated distance by the number of features (or dimensions) effectively involved in its estimation. The reason for this normalization is to make the range of ε_X and ε_Y comparable in (2.19). Indeed, the output target variable is of dimension 1 while,

conversely, the original dimension of the samples is often much higher in practice. Without the proposed normalisation, ε_X would likely always be greater than ε_y because the distance in the sample space would be computed using much more dimensions; the MI estimation would therefore be harmed.

As a very simple example, the estimated distance between the samples $[3 \bullet 8\ 7\ 2]$ and $[1\ 9 \bullet 3\ 4]$, with $\bullet$ denoting a missing value, is computed by only taking into account the first, fourth and fifth elements of each sample; it is equal to

$$\sqrt{\frac{(3-1)^2+(7-3)^2+(2-4)^2}{3}}.$$

Once the pairwise distances between samples have been obtained, the nearest neighbors can be determined in both the sample and the target output spaces, and the MI can be traditionnaly estimated using Equation (2.19). Here, we combine the MI criterion estimated as described above with a greedy forward search procedure to build the subsets of relevant features. It is interesting to note that the PDS cannot be related to any kind of imputation method.

Despites its simplicity, PDS has been successfully used in other topics than feature selection, including clustering [92], inference of prediction problems [3] and self-organizing maps [168]. Let us note that in the PDS acronym, the term distance is used abusively. Indeed, the similarity measure defined by the PDS is not guaranteed to obey the triangle inequality and cannot thus be considered formally as a distance measure.

6.2.1 Practical considerations

Some practical details about the proposed methodology deserve to be clarified.

First, the parameter k of the MI estimator, corresponding to the number of nearest neighbors, is set to a moderate value of 6, as suggested in the original paper [114].

Then, the PDS distance estimation is obviously not able to produce a distance estimate if two samples have no common features for whose values are available. In this case, the distance between the samples is considered as infinite and the points cannot therefore be considered as neighbors.

Finally, prior to the MI estimation, the real-world datasets have to be normalized such that each feature has zero mean and unit variance. Indeed, we do not want that because of their range, some features are given more weight than other ones in the estimation of the distance between samples.

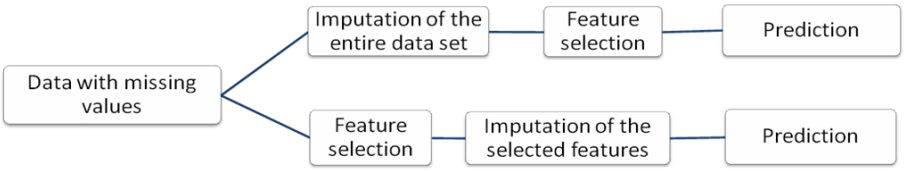


Figure 6.1: Two different approaches to the regression problem with missing values.

6.2.2 Experimental results

We now provide some experimental evidences about the interest of the suggested feature selection algorithm. More specifically, we compare the proposed approach with the strategy consisting in imputing the dataset before feature selection and using the same MI estimator. Since for real-world problems we compare the feature selection algorithms through the prediction performances of regression models, the data have to be imputed before the prediction step. Therefore, our experimental framework allows us to compare the two main approaches to prediction with missing data, illustrated in Figure 6.1. Our intuition is that the strategy in the lower part of Fig. 6.1 (feature selection THEN imputation) is computationally more efficient and potentially leads to more accurate predictions.

Two imputation strategies are used for comparison: the ICkNNI with 10 neighbors and a regularized version of the EM algorithm [157]. Note that the ICkNNI method fails to produce an estimation of the missing values if for a given sample whose values are missing for some features, it cannot be found neighbours with observed values for those features; the mean value of the feature is used instead in such situations.

Artificial datasets

Two artificial datasets are first used to evaluate the performances of the proposed algorithm. Both datasets consist in ten continuous features, whose values are uniformly distributed between 0 and 1. They are built such that only a subset of the ten features are useful to predict the associated continuous target output.

The first problem is inspired from [75]; its output is defined as:

$$Y_1 = 10\sin(X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \varepsilon, \tag{6.4}$$

where ε is a Gaussian noise with unit variance. The second output is computed as:

$$Y_2 = X_1 X_2 + \sin(X_3) + 0.5X_4 + \varepsilon. \tag{6.5}$$

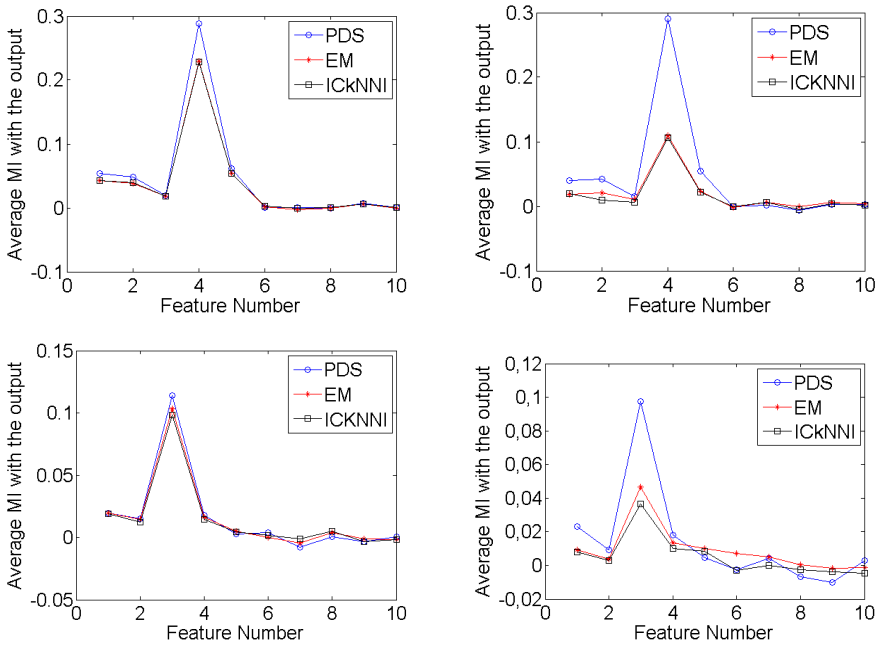


Figure 6.2: Average MI between each individual feature for, from top to bottom, Y_1 and Y_2 with 10% (left) and 40% (right) of missing values; $\circ$: PDS, $\square$: ICkNNI, $\star$: regularized EM.

Obviously, only the five first features are useful for the prediction of Y_1 while only the four first ones are needed to predict Y_2. For both problems (6.4) and (6.5), 50 datasets of sample size 100 have been generated. Each dataset has been randomly filled with various percentages of missing values. As explained above, three strategies are compared: the proposed approach, feature selection after imputation by the EM algorithm and feature selection after imputation by ICkNNI.

Feature ranking The first experiment we conduct is simply to rank the features according to the MI criterion. This is indeed important to make sure that the estimated MI is still a valuable criterion in the context of missing data. Besides, in a forward search procedure, the choice of the first feature is actually based on univariate MI estimations. It is important to note that in the context of ranking procedures, the PDS approach is equivalent to considering only the available values for the MI estimation.

Figure 6.2 shows the average MI between each feature and the output, over the 50

	10% of missing data					40% of missing data				
	5	4	3	2	1	5	4	3	2	1
PDS	14	49	30	6	1	8	40	43	9	0
Regularized EM	0	1	6	30	49	0	1	23	45	28
ICkNNI	0	1	5	39	43	0	0	9	43	40

Table 6.1: Percentage of experiments for which a certain number of relevant features (second line of the table) are ranked in the five first positions for the Y_1 problem with 10 and 40 % of missing data.

repetitions of the experiment. Table 6.1 also presents the percentage of experiments for which five, four, three, two and one relevant features have respectively been ranked in the five first positions for the Y_1 problem. Results are similar for the other problem and are not shown here for concision reasons.

Results clearly indicate the advantage of the PDS-based approach. First, the results are similar when only 10% of the values are missing and the three approaches all discriminate well between irrelevant and relevant features in this situation. Conversely, when the proportion of missing values becomes higher (40%), imputation based methods do not seem able anymore to correctly assess the relevance of a feature. As an example, for the second problem, the MI between feature 5 and the output is greater than the MI between the output and feature 1 or 2 according to both imputation techniques. Such strategies could therefore be misleading about the relative importance of the features for a regression problem. When the PDS is used, the average MI is always higher for the relevant features than for the irrelevant ones. These claims are confirmed by Table 6.1 which clearly shows that using the PDS method leads to the selection of a more important fraction of relevant features than the one obtained with imputation-based methods.

Forward search Results are now presented for a forward search on the artificial datasets. The search has been stopped when the number of relevant features had been selected, i.e. five for the first problem and four for the second one. Table 6.2 summarizes the performances of the three compared approaches for the Y_1 output and various amount of missing data. For the other problem, relevant features are always selected first by all methods. As can be seen, the proposed methodology largely outperforms its competitors regarding the correct determination of the entire set of relevant features.

Real-World Datasets

We now compare the performances of the feature selection methods on two real-world datasets; more extensive results can be found in [55].

The first dataset is the Housing dataset; from its 506 instances, 337 are kept for

% of missingness	PDS	Regularized EM	ICkNNI
10	100	100	100
20	84	70	62
30	66	46	44

Table 6.2: Percentage of correct selection of the relevant set of features by a forward search procedure for the Y_1 artificial problem.

		KNN			
	%	EM	PDS then EM	ICkNNI	PDS then ICkNNI
Housing	1	2,9366 ± 0,1277	2,8252 ± 0,0690	2,8896 ± 0,1783	2,8522 ± 0,0590
	5	3,1365 ± 0,1341	3,1083 ± 0,1548	3,9726 ± 0,2323	**2,9732 ± 0,1551**
	10	3,5232 ± 0,3061	3,4998 ± 0,3617	4,4350 ± 0,2238	**3,4646 ± 0,1942**
	20	3,6588 ± 0,5177	3,7271 ± 0,3508	6,0218 ± 1,0507	**4,0720 ± 0,3173**
Mortgage	1	0,3709 ± 0,0193	0,3700 ± 0,0149	0,3515 ± 0,0401	0,3571 ± 0,0250
	5	0,3986 ± 0,0640	0,3545 ± 0,0394	0,4339 ± 0,0227	**0,3778 ± 0,0595**
	10	0,3908 ± 0,0337	0,3519 ± 0,0388	0,4950 ± 0,0986	**0,3589 ± 0,0428**
	20	0,4854 ± 0,0501	**0,3680 ± 0,0686**	0,8355 ± 0,0946	**0,4057 ± 0,0435**

Table 6.3: Best RMSE reached by a KNN prediction model on two real-world datasets for various rates of data missingness. Significantly better results for one of the two compared approches are shown in bold.

training the model while the 169 remaining ones constitute the test set. Experiments are also conducted on the Mortgage dataset; from the 1049 data points available, the 700 first ones are used as the training set.

As the relevant features are not known in advance, the criterion of comparison is the root mean square error (RMSE) of a regression model built from the selected features. As almost no work has been done to propose prediction models able to handle missing values (see [138] for an exception in classification), it is necessary to impute the data after the feature selection step if the PDS-based feature selection strategy is used. Two regression models are tested: the k nearest neighbors predictor, with $k = 10$, and a Radial Basis Functions Network model (RBFN) whose parameters are optimised according to [16]. For each complete dataset, 1, 5, 10 and 20 % of missing values are randomly introduced and this procedure is repeated 10 times, producing 40 new datasets. The two branches of Figure 6.1 are again compared.

Tables 6.3 and 6.4 give the average lowest RMSE obtained over the 10 incomplete datasets, together with the standard deviation, for the four missing data rates. In those tables, EM or ICkNNI means that the data have first been imputed using the corrsponding method; PDS then EM or ICkNNI means that the feature selection has been conducted with missing data and indicates the imputation method used to build the prediction model. Cases where one of the two strategies performs significantly better than the other one according to a paired test with a 95% confidence level are shown in bold.

Again, the interest of the proposed approach is demonstrated by the results. In-

		RBFN			
	%	EM	PDS then EM	ICkNNI	PDS then ICkNNI
Housing	1	7,7308 ± 0,6103	7,6343 ± 0,7922	7,7187 ± 0,4537	7,6061 ± 1,2242
	5	8,1533 ± 0,9341	**7,0395 ± 0,2474**	7,0163 ± 0,7397	6,9950 ± 0,4565
	10	7,6845 ± 0,3980	7,3737 ± 0,2737	7,8922 ± 2,1218	7,4886 ± 0,5520
	20	8,3647 ± 0,4642	**7,6461 ± 0,4159**	9,3573 ± 0,9904	**7,6116 ± 0,3985**
Mortgage	1	0,1819 ± 0,0692	0,1517 ± 0,0521	0,1714 ± 0,0919	0,2184 ± 0,098
	5	0,2680 ± 0,0750	0,2668 ± 0,0646	0,3568 ± 0,1576	0,2664 ± 0,1344
	10	0,3068 ± 0,0757	**0,1901 ± 0,0624**	1,006 ± 0,463	**0,2556 ± 0,1106**
	20	0,3964 ± 0,1784	**0,2244 ± 0,0722**	1,6213 ± 0,284	**0,5071 ± 0,2719**

Table 6.4: Best RMSE reached by a RBFN prediction model on two real-world datasets for various rates of data missingness. Significantly better results for one of the two compared approches are shown in bold.

deed, it leads to an average best regression performance than the imputation-based techniques in all but 3 of the 32 comparisons. An important point is that the PDS-based approach seems much less sensitive to the increase of the missingness rate. For instance, on the Mortgage dataset, the mean lowest RMSE is reduced by more than 25% in comparison with the ICkNNI imputation and by more than 50% in comparison with the regularized EM imputation for both prediction models and 20% of missing values. Globally, the RBFN model always performs better with the PDS-based method when 5% of values at least are missing. Eventually, the results obtained with the proposed strategy are statistically significantly better for almost half of the considered comparisons.

6.2.3 Further considerations

The obtained results are clearly in favor of the proposed PDS-based feature selection approach. However, some points have to be discussed.

Indeed, the proposed methodology could suffer from a few drawbacks. First, and similarly to most methods dealing with missing data, the performances of the algorithm are likely to decrease when the missingness rate becomes high, even if the methodology could theoretically be applied with any percentage of missing values. Indeed, for high missingness rates, the first steps of the forward procedure could be based on a very few number of samples and therefore be unreliable. One solution is to rather use a backward search strategy, or to begin the forward procedure with groups of two or three features instead of singletons; this way, the whole procedure would not be affected by a bad choice of the first few features.

Then, it is not possible to determine whether the estimated MI values are meaningful. Said otherwise, the proposed methodology can estimate the MI for any rate of missing value; however there probably exists a certain rate of missingness above which the estimated MI values will not reflect true dependencies anymore since the estimator

will be given too few information. Knowing the relevance of an estimated value is obviously of great importance for feature selection. One solution could be found in the permutation test [70]. Comparing the MI estimations for relevant and random features could give very valuable information on whether or not the estimations correspond to true dependencies.

6.3 Distance estimation with missing data

Despites its efficiency, the PDS distance estimation method remains quite basic. In particular, this strategy tends to underestimate the distances, beacause it ignores the general variability of the data, and only considers the locally known components. There is thus room for improvements and the development of more sophisticated methods. The solution described in the remaining of the chapter aims at estimating the expected squared distance between samples and is based on the assumption of a multivariate normal distribution, even if experiments show the robustness of the approach even when this assumption is violated.

6.3.1 Theoretical developments

For a sample $\mathbf{x_{i*}} \in \Re^d$, let us denote by M_i the set of indices denoting the dimensions for which the values of $\mathbf{x_{i*}}$ are missing. The first idea is to rewrite the squared distance by identifying four terms according to the missing attributes for both samples:

$$\begin{aligned}\left\|\mathbf{x_{i*}} - \mathbf{x_{j*}}\right\|^2 = &\sum_{k \notin M_i \cup M_j} (x_{ik} - x_{jk})^2 + \sum_{k \in M_j \setminus M_i} (x_{ik} - x_{jk})^2 + \\ &\sum_{k \in M_i \setminus M_j} (x_{ik} - x_{jk})^2 + \sum_{k \in M_i \cap M_j} (x_{ik} - x_{jk})^2.\end{aligned} \tag{6.6}$$

The first term of Eq. (6.6) corresponds to the known components for both samples and can be computed directly. The second (third) term corresponds to components known for $\mathbf{x_i}$ ($\mathbf{x_j}$) and not for $\mathbf{x_j}$ ($\mathbf{x_i}$). The fourth term corresponds to components missing for both samples. Let us then consider the missing values as random variables $X_{ik},\ k \in M_i$

and take the expectation of Eq. (6.6).

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{x_{i*}}-\mathbf{x_{j*}}\right\|^2\right] &= \sum_{k\notin M_i\cup M_j}(x_{ik}-x_{jk})^2+\sum_{k\in M_j\setminus M_i}\mathbb{E}\left[(x_{ik}-X_{jk})^2\right] \\
&+\sum_{k\in M_i\setminus M_j}\mathbb{E}\left[(X_{ik}-x_{jk})^2\right]+\sum_{k\in M_i\cap M_j}\mathbb{E}\left[(X_{ik}-X_{jk})^2\right] \\
&=\sum_{k\notin M_i\cup M_j}(x_{ik}-x_{jk})^2 \\
&+\sum_{k\in M_j\setminus M_i}((x_{ik}-\mathbb{E}\left[X_{jk}\right])^2+\mathrm{Var}\left[X_{jk}\right] \\
&+\sum_{k\in M_i\setminus M_j}(\mathbb{E}\left[(x_{ik}\right]-X_{jk})^2+\mathrm{Var}\left[X_{ik}\right] \\
&+\sum_{k\in M_i\cap M_j}(\mathbb{E}\left[(X_{ik}\right]-\mathbb{E}\left[X_{jk}\right])^2)+\mathrm{Var}\left[X_{ik}\right]+\mathrm{Var}\left[X_{jk}\right].
\end{aligned} \tag{6.7}$$

The developments in Eq. (6.7) result from the linearity of the expectation operator. For clarity, the expansion for the second term is now detailed:

$$\begin{aligned}
\mathbb{E}\left[(x_{ik}-X_{jk})^2\right] &= \mathbb{E}\left[x_{ik}^2+X_{jk}^2-2\,x_{ik}\,X_{jk}\right]=x_{ik}^2+\mathbb{E}\left[X_{jk}^2\right]-2x_{ik}\,\mathbb{E}\left[X_{jk}\right] \\
&=x_{ik}^2-2x_{ik}\,\mathbb{E}\left[X_{jk}\right]+\mathbb{E}\left[X_{jk}\right]^2-\mathbb{E}\left[X_{jk}\right]^2+\mathbb{E}\left[X_{jk}^2\right] \\
&=\left(x_{ik}-\mathbb{E}\left[X_{jk}\right]\right)^2+\mathbb{E}\left[X_{jk}^2-\mathbb{E}\left[X_{jk}\right]^2\right] \\
&=\left(x_{ik}-\mathbb{E}\left[X_{jk}\right]\right)^2+\mathrm{Var}\left[X_{jk}\right].
\end{aligned} \tag{6.8}$$

Following Eq. (6.7), the expectation and variance of each random variables are thus sufficient to estimate the distance between two samples. If we assume the samples $\mathbf{x_{i*}}$ are drawn independently from a multivariate distribution, the distribution of each random variable X_{ik} is equal to the conditional distribution of the random variable, conditioned on the observed samples. This implies that when estimating the expected squared distance, it is sufficient to estimate the mean and covariance (conditionned to the observed values) for each missing value separately. To formalize this idea, let us define x'_{ik} and σ^2_{ik} as:

$$x'_{ik}=\begin{cases}\mathbb{E}\left[X_{ik}|x_{i,obs}\right] & \text{if } k\in M_i\\ x_{ik} & \text{otherwise}\end{cases} \tag{6.9}$$

and

$$\sigma_{ik}^2 = \begin{cases} \text{Var}\left[X_{ik}|x_{i,obs}\right] & \text{if } k \in M_i \\ 0 & \text{otherwise.} \end{cases} \tag{6.10}$$

Using the above notations, the expected square distance can be rewritten as:

$$\mathbb{E}\left[\left\|\mathbf{x_{i*}} - \mathbf{x_{j*}}\right\|^2\right] = \left\|\mathbf{x'_{i*}} - \mathbf{x'_{j*}}\right\|^2 + \sum_{k\in M_i} \sigma_{ik}^2 + \sum_{k\in M_j} \sigma_{jk}^2. \tag{6.11}$$

This last expression clearly emphasizes the fact that the uncertainty of the missing values is taken into account and that the estimated distance is not limited to the difference between imputed samples.

The conditional mean and variance

As the distribution of the random variables is obviously unknown in practice, hypotheses have to be made in order to evaluate the required conditional mean and variance. Here, the assumption is made that the data samples are drawn independently from a parametric distribution: the multivariate normal distribution. Among other advantages discussed in more details in [61], the normal distribution has the advantage of being fully determined by the mean and the covariance matrix.

The conditional mean and variance can be directly computed under the multivariate normal assumption. Indeed, let us divide the components of a multivariate normal random variable X in $X^{(1)}$ (the missing values) and $X^{(2)}$ (the observed values). Let us proceed similarly for the mean μ and the covariance matrix Σ such that:

$$X = \begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix}, \mu = \begin{bmatrix} \mu^{(1)} \\ \mu^{(2)} \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}.$$

It can then be shown [4] that the conditional distribution $\left\{X^{(1)}|X^{(2)} = \mathbf{x}^{(2)}\right\}$ follows a normal distribution with mean and covariance matrix respectively given by:

$$\mu'_1 = \mu^{(1)} + \Sigma_{12}\Sigma_{22}^{-1}\left(x^{(2)} - \mu^{(2)}\right) \tag{6.12}$$

and

$$\Sigma'_{11} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \tag{6.13}$$

The only remaining task is thus now to estimate the covariance matrix, which is not trivial in practice when dealing with missing values. One popular solution is to use maximum likelihood estimation. For instance, it is detailed in [126] how the EM algorithm can be used to produce an estimate. Here, the expectation conditional

maximisation (ECM) algorithm [132] with the improvements proposed in [160] is considered.

6.3.2 Summary of the proposed approach

We now sum up the consecutive steps of the expected squared distance estimation procedure we suggest. As a reminder, we consider a dataset of n samples $\mathbf{x_{i*}} \in \Re^d$; the indices of the missing components in each sample are denoted as M_i.

1. Estimate the mean μ and the covariance Σ of the dataset using the ECM algorithm.

   Repeat the following steps for each sample $\mathbf{x_{i*}}$ with missing values.

2. Compute the conditional mean $\mu_1' = \mu^{(1)} + \Sigma_{12}\Sigma_{22}^{-1}\left(\mathbf{x_{i*}^{(2)}} - \mu^{(2)}\right)$

3. Define $\mathbf{X_{i*}}'$ by replacing the missing values in $\mathbf{x_{i*}}$ by values from μ_1'

4. Compute the covariance matrix $\Sigma_{11}' = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ and define $s_i^2 = \sum_{k=1}^{|M_i|}(\Sigma_{11}')_{kk}$

   For each couple of samples $\mathbf{x_{i*}}$, $\mathbf{x_{j*}}$ repeat the following steps

5. Compute $P_{ij} = \sum_{k=1}^{d}\left(x_{ik}' - x_{jk}'\right)^2$

6. Add the sum of the conditional variances to the estimated distances $P_{ij} = P_{ij} + s_i^2 + s_j^2$

The matrix P built this way contains the estimated expected squared distance between each pair of samples.

6.3.3 Further considerations

We end this section by briefly giving some practical considerations on the proposed distance estimation method.

The normal hypothesis The Gaussian hypothesis is at the heart of the proposed method since it allows most of the presented developments. However, it is very likely that the Gaussian hypothesis will not hold in practice. Considering the effects of dealing with non-Gaussian distributions is therefore important. First, it is important to note that Equationss (6.12) and (6.13) are exact in some non-Gaussian cases. For instance, the equations hold exactly if all the variables are independent and if their

variance is positive and finite. Indeed, the covariance matrix would then be diagonal for any distribution. Equations (6.12) and (6.13) can also be shown to hold if a subset of variables follow a multivariate normal distribution while the remaining variables are (i) mutually independent and (ii) independent of the normally distributed variables.

If the data does not follow a Gaussian distribution, determining the mean and covariance will likely produce a normal distribution covering too large areas of the space. The conditional variance terms will then probably be overestimated, and so will the distances with high uncertainty (i.e. the distances between samples with many missing values). In practical applications, more importance is often given to small distances or to the determination of nearest neighbors. In this context, overestimating distances for which the estimations are not likely to be accurate is actually not a drawback. Indeed, it is often better no to detect the closeness between two samples than to treat two samples as close when it is not the case.

Extensive experiments presented in [61] confirm the interest of the method for both distance estimation and nearest neighbors determination, even for data far from being Gaussian.

Special case and further use An important special case is when the features of the dataset can be assumed to be independent. The proposed algorithm then becomes much simpler as only the mean and the variance of each variable then need to be estimated. In particular, no matrix inversion is required anymore and the algorithm runs much more quickly.

Then, let us also mention that the proposed methodology can be easily extended to any squared weighted distance in the form $((\mathbf{x}_{\mathbf{i}*} - \mathbf{x}_{\mathbf{j}*})^T S((\mathbf{x}_{\mathbf{i}*} - \mathbf{x}_{\mathbf{j}*})$. It is only required to estimate the Cholesky decomposition $S = LL^T$ and to slightly adapt the mean and covariance terms. Let us also note that the estimated distance matrix can safely be used in kernel methods considering a RBFN or Gaussian kernel. Indeed, it is shown in [61] that the kernel matrix built this way can be guaranteed to be positive definite.

6.4 Conclusion

This chapter introduces a feature selection algorithm for regression problems with missing values. The idea is to adapt a nearest neighbors-based MI by using a distance measure able to directly deal with missing values. Experiments on both artificial and real-world data show that this strategy often leads to improved performances compared to the case where data are first imputed (before the feature selection step).

An improved distance estimation procedure between samples with missing values is then presented. The developments are based on the hypothesis of a multivariate

Gaussian distribution while experimental results presented in [61] actually indicate that the proposed strategy works well, even with non-Gaussian data. The developments are theoretically sound and the procedure can easily be adapted to the compution of weighted distances; it can also safely be used for kernel-based methods.

Chapter 7

Feature selection for uncertain class labels

Contents

This chapter is concerned with feature selection for problems where the target class labels are uncertain. In particular, we address the cases (i) where some labels can be wrong (label noise) in Section 7.1 and (ii) where some labels are provided with a probability of correctedness in Section 7.2.

7.1 Feature selection in the presence of label noise

Label noise is a frequently encountered problem in machine learning and data mining [23]. Indeed, it is common in practice that class labels are wrongly assigned to samples, for instance when a human expertise is needed to assign class labels. As an example, in medicine, practicians have to make diagnoses based e.g. on blood analyses, on X-ray images or on electrocardiograms monitoring. Such tasks can be hard to perform, repetitive and time-consuming; they are obviously error-prone. Besides, errors can also be made when labels are encoded in a data set.

Label noise is known to have a negative impact on the behaviour of most supervised classification algorithms. It is therefore likely that it will as well degrade the performances of supervised feature selection algorithms. In this regard, proposing a label noise-tolerant feature selection algorithm would therefore be of considerable interest. Quite surprisingly, this task has not been addressed in the literature.

We first begin by analysing the impact of label noise on a traditional MI-based filter feature selection algorithm, to make completely sure that the problem is interesting to consider. A proposal to make the nearest neighbors-based MI estimator less sensitive to label noise is then introduced. The approach is based on a statistical label noise model combined with an expectation-maximisation (EM) algorithm.

Section 7.1.1 briefly reviews basic notions about the label noise problem. The impact of label noise on feature selection is illustrated in Section 7.1.2. Section 7.1.3 introduces a label noise-tolerant entropy estimator, requiring the knowledge of the true class memberships and on which is based the MI estimator (see Section 2.1.3). These memberships can be estimated as described in Section 7.1.5. The label noise-tolerant MI estimator is introduced in Section 7.1.4 and is experimentally tested in Section 7.1.6.

7.1.1 Related work

Two main approaches are generally thought of in the literature to deal with label noise in classification problems.

One popular approach consists in filtering the noisy data, with the objective of detecting the wrongly labelled samples. These samples can then be removed or their

label can be corrected. Many criteria indicating mislabelled data points can be considered. For instance, [89] uses the information gain, while [12] observes the class labels of the closest samples whose majority is supposed to belong to the class of the studied sample.

A completely different strategy is to build an explicit model for the label noise. The first and most well-known work following this strategy is [119], where the authors propose a probabilistic noise model and combine it with a Fisher Kernel discriminant to perform binary classification. The original model in [119] was later extended in [125] where the authors get rid of the limiting assumption of normal distribution. The same model was more recently further adapted to multi-class problems [19]. Besides the model in [119], other models can be found in the literature (see e.g. [21]). Such a model-based approach is considered in this paper as it has the advantage of being theoretically sound and does not require discarding potentially valuable samples.

Eventually, in a slightly different context, some works assume uncertain or partial information about the class labels to be readily available [33].

7.1.2 Impact of label noise on feature selection

The objective of this small section is to illustrate that the problem we consider actually has an interest and that label noise really does impact the performances of supervised feature selection algorithms.

To this end, we have conducted a greedy backward feature selection procedure using the nearest neighbors-based MI estimator on the well-known Iris dataset from the UCI repository [71]. We have then run the same experiment but after 20% of the class labels have randomly been switched to another class, chosen equiprobable among the other classes. To assess the quality of feature selection, the balanced classification error rate of a k-nearest neighbors classifier is studied. Let us notice that noisy data have only been considered for the feature selection step, while the whole classification step has been conducted with the noise-free data. This allows us to compare the results on the actual problem of interest. We delay the technical details to Section 7.1.6.

Even for a dataset as simple as Iris, Figure 7.1 shows that the performances of the feature selection algorithm are highly (and negatively) affected by the presence of label noise. Indeed, as can be observed, the balanced classification error rate (which is the average of the classification accuracy obtained for each class) is significantly larger for the noisy feature selection. Besides, another important aspect is that the stability of the feature selection also seems to be affected by the presence of label noise, as the confidence intervals on the values of the error criterion are much larger in this situation.

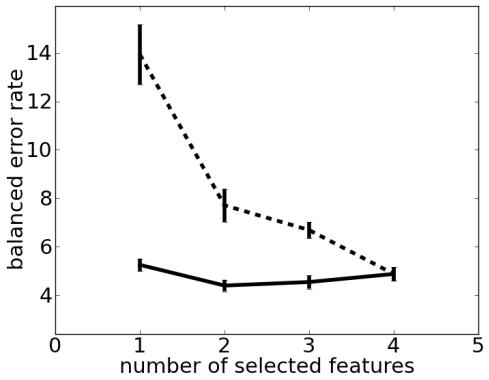


Figure 7.1: Balanced classification error rate of a k-nearest neighbors classifier as a function of the number of selected features for noise-free (plain line) and noisy (dashed line) data. The error bars correspond to a 95% confidence interval on the mean of the values of the error criterion.

7.1.3 The need for a label noise-tolerant entropy estimator

As explained above, designing a noise-tolerant feature selection algorithm is an important task. To this end, we propose to adapt the nearest neighbors-based MI estimator designed for classification (see Section 2.1.3). As a reminder, this estimator uses the Kozachenko-Leonenko entropy estimator (2.15). There are actually two ways in which the MI and entropy estimators can be affected by label noise.

First, remember that empirical estimators of conditional entropy $\hat{H}(X|Y=y)$ are required to estimate the MI (see Eq. (2.20)). Obviously, instances with an incorrect label will be used in the estimation of a wrong quantity. Indeed, assume that the sample $\mathbf{x_{i*}}$ has 0 as true class label but is wrongly given the label 1 in the dataset. It will be used to estimate $H(X|Y=1)$ where it should actually be used for $H(X|Y=0)$, which affects both estimations.

Then, closely related, the estimations of $H(X|Y=y)$ using the estimator (2.15) are likely to be biased. Indeed, the entropy estimator (2.15) and therefore the density estimator (2.17) are based on $\varepsilon_y(i,k)$, the diameter of the hypersphere containing the k nearest neighbors of $\mathbf{x_{i*}}$ with label y. It is easy to understand that this diameter will be affected by label noise. A particularly problematic situation would be the case of a wrongly labelled sample *lost* in the middle of a large amount of samples from another class. In such a case, the volume of the hypershere containing k neighbors of the same class would probably be very (and abnormally) large; the estimate of $p_{X|Y}(x_i|y)$ would then be close to zero and large negative values would occur in (2.18) and equivalently

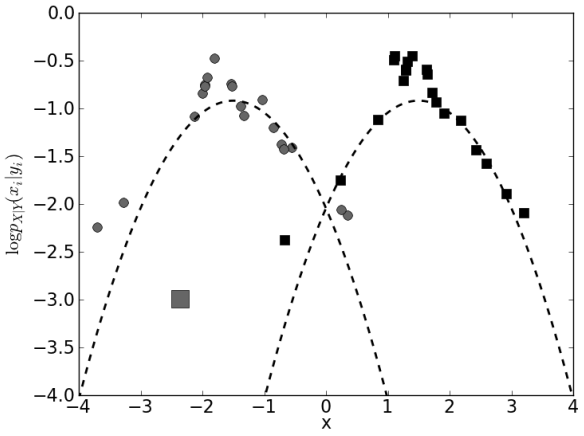


Figure 7.2: Estimates of the logarithm of the conditional probability $p_{X|Y}$ for a classification problem. Both class have a Gaussian distribution (dashed lines). Circles belong to class 0 and black squares belong to class 1: the large grey square sample is mislabelled.

in the third term of Equation (2.21).

To illustrate this issue, 40 samples are generated from two classes with Gaussian distributions $\mathcal{N}(\mu = \pm 1.5, \sigma = 1)$ and identical priors. One of the samples with label 0 is switched to 1. Figure 7.2 shows estimates of $\log p_{X|Y}(x_i|y_i)$ computed using the Kozachenko-Leonenko approach. For the mislabelled sample a large negative value can be observed, while it is not the case for the other correctly labelled samples. The estimate of MI is only 0.58 with the switched label and 0.63 for a clean version of the dataset. An important difference can thus be observed, even with only one mislabelled sample.

To address the two aforementioned issues, one possibility is to associate each sample with what we consider is its *true* class label, which can of course differ from the observed label. We thus consider each of these observed labels as noisy versions of true hidden labels. For each sample, we would therefore like to have an estimate of the membership $p_{S|X,Y}(s|x,y)$ of an instance $\mathbf{x}$ to the true class s, if the observed label is y. Note that those memberships are not crisp but can take any value between 0 and 1, provided that sum of the memberships for a given sample is always 1.

To estimate the required memberships a sound strategy consists in (i) choosing a label noise model representing the nature of the noise and (ii) estimating the most probable class memberships given the available data. For the subsequent developments, we make the assumption that these true class memberships are available, while

they are usually unknown in practice. The estimation procedure of these quantities is delayed to Section 7.1.5.

To keep things simple, we introduce the following notation:

$$\gamma(s|i) = p_{S|X,Y}(s|x_i, y_i). \tag{7.1}$$

7.1.4 An entropy estimator based on the class memberships

Remember from Section 2.1.3 that the density estimator (2.17) for $p_{X|Y}$ relies on the hypothesis that $p_{X|Y}$ is constant in the hypersphere of diameter $\varepsilon_x(i,k)$ which contains the k nearest neighbors of the i^{th} sample. Of course, rather than estimating $H(X|Y = y)$, one would prefer estimating $H(X|S = s)$ using the true class labels.

The solution proposed in this paper is to consider a hypersphere containing an expected number of k instances truly belonging to the target class s. As we have assumed the class memberships to be available, they can be used to determine the *true* hypersphere diameter. Indeed, for any class $s \in \mathcal{Y}$ and any sample $\mathbf{x_{i*}}$, one can define the hypersphere containing enough neighbours of $\mathbf{x_{i*}}$ such that the sum of these neighbors memberships to class s is approximately equal to k.

For each class $s \in \mathcal{Y}$ and each sample $\mathbf{x_{i*}}$, let us first consider the standard hypersphere containing the k nearest neighbors of $\mathbf{x_{i*}}$. In this hypersphere, there is an expected number of

$$\sum_{j=1}^{k} \gamma(s|i_j) \tag{7.2}$$

instances belonging to target class s, with i_j being the index of the j^{th} neighbor of sample $\mathbf{x_{i*}}$. Intuitively, for correctly labelled samples, (7.2) is expected to be close to k for their observed class and close to zero for other classes; an exception is for samples near the classification boundary, where the number of instances of each class tends to be proportional to the classes prior. On the contrary, for mislabelled samples, the number of samples from the observed class contained in the hypersphere is expected to be very low. Indeed, mislabelled samples are supposed to lie in a region where samples of the same class should not be observed.

The idea is then to increase the diameter of the hypersphere $\varepsilon_{s,\gamma}(k,i)$ until the sum of the memberships

$$\Gamma(s|i) = \sum_{j=1}^{k(s|i)} \gamma(s|i_j) \tag{7.3}$$

is at least equal to k. In Eq. (7.3), $k(s|i)$ denotes the number of considered neighbors. The obtained hypersphere contains an expected number of $\Gamma(s|i) \approx k$ instances belonging to the target class s.

Intuitively, the new diameter $\varepsilon_{k,\gamma}(i|y)$ of an hypersphere associated with a given observed label y is likely to be much larger for mislabelled samples than for correctly labelled ones. Indeed, for mislabelled samples, the hypersphere has to become large enough to encompass a part of the space containing samples with sufficiently high memberships to the class y. Contrarily, for the hypersphere associated with the true class of a mislabelled sample, the diameter should be close to the diameter for correctly labelled samples of the same class.

Using the newly derived hypersphere diameters, the following noise-tolerant density estimator is proposed, based on the original estimator (2.17):

$$\log \hat{p}_{X|S}(x_i|s_i) = \psi(\Gamma(s|i)) - \psi(\Gamma(s)) - \log c_d - d \log \varepsilon_{s,\gamma}(k,i) \tag{7.4}$$

with

$$\Gamma(s) = \sum_{i=1}^{n} \gamma(s|i). \tag{7.5}$$

The quantity $\Gamma(s)$ can be interpreted as the expected number of samples really belonging to class s, and it is used instead of the number of samples with label s considered in the original estimator. In addition, the sum of the memberships $\Gamma(s|i)$ is used instead of the parameter k while the new hypersphere diamteer is considered instead of the original one. Since $\gamma(s|i)$ represents the probability with which $\mathbf{x_{i*}}$ belongs to class s, it is possible to write a noise-tolerant expression for the estimation of the conditional entropy by combining Equations (2.18) and (7.4) as:

$$\begin{aligned} \hat{H}(X|S=s) = &- \frac{1}{\Gamma(s)} \sum_{i=1}^{n} \gamma(s|i) \psi(\Gamma(s|i)) + \psi(\Gamma(s)) \\ &+ \log c_d + \frac{d}{\Gamma(s)} \sum_{i=1}^{n} \gamma(s|i) \log \varepsilon_{k,\gamma}(i|s). \end{aligned} \tag{7.6}$$

For feature selection we are mainly interested in the MI criterion, for which we now trivially derive a noise-tolerant version from (7.6). Indeed, in the last part of Eq. (2.20) the entropy $H(X)$ is not affected by label noise as it only involves the samples while the conditional entropies $H(X|Y=y)$ can be estimated using Eq. (7.6). As a result, the following noise-tolerant version of the MI estimator (2.21) is proposed:

$$\hat{I}(X;S) = \hat{H}(X) - \sum_{s \in \mathcal{S}} \hat{p}_S(s) \hat{H}(X|S=s) \tag{7.7}$$

where $\hat{p}_S(s) = \Gamma(s)/n$. This estimator thus measures the dependencies between the features and the true classes rather than noisy versions of them. It is therefore expected to be more reliable for feature selection in the presence of label noise.

7.1.5 True class memberships estimation

Up to now, the knowledge of the true class memberships has been assumed, which is an unrealistic hypothesis in practice (the label noise problem would therefore be useless to consider). This section introduces an expectation maximisation (EM) algorithm to estimate the required memberships.

Label noise modelling

The first step in estimating the memberships is to consider a model for the label noise. Here, we consider the model introduced in [119]. Its underlying assumptions are that (i) according to its true class s, an instance has a probability $p_e(s)$ to be mislabelled and that (ii) if an instance is mislabelled, the incorrect label is randomly chosen amongst the wrong labels $\mathcal{Y} \setminus \{s\}$. Under both hypotheses, one can write:

$$p_{Y|S}(y|s) = \begin{cases} 1 - p_e(s) & \text{if } s = y \\ \frac{p_e(s)}{|\mathcal{Y}|-1} & \text{if } s \neq y. \end{cases} \tag{7.8}$$

The question is then how to estimate the error probabilities $p_e(s)$ and therefore the true class labels.

Derivation of an objective criterion for label noise estimation

Remember from Section 7.1.3 that one of the main effects of label noise is to increase the estimated partial conditional entropy

$$\hat{H}(X|Y=y) = -\frac{1}{n_y} \sum_{i|y_i=y} \log \hat{p}_{X|Y}(\mathbf{x_{i*}}|y). \tag{7.9}$$

by introducing large negative values for $\log \hat{p}_{X|Y}(\mathbf{x_{i*}}|y)$ which increase the partial conditional entropies and eventually decrease the estimated MI. These problems come from the difficulty of standard models to explain abnormal instances which bias the (estimated) class conditional distributions.

The interest of label noise modelling precisely lies in its capacity to explain noisy observations, helping to increase $\hat{p}_{X|Y}(x|y)$ for misclassified instances and to limit the decrease of the estimated MI value. This last remark suggests that looking for values of p_e that leads to high values of MI can be an interesting way to determine the label noise model. We thus propose to determine values for p_e as follows:

$$\hat{p}_e = \arg\max_{p_e} \hat{I}(X;Y). \tag{7.10}$$

Combining Equations (2.20) and (7.9), it can be shown that the optimisation problem (7.10) is completely equivalent to

$$\hat{p}_e = \arg\max_{p_e} \sum_{i=1}^{n} \log \hat{p}_{X|Y}(\mathbf{x_{i*}}|y_i). \tag{7.11}$$

An EM algorithm for label noise model estimation

Equation (7.11) can further be rewritten by noting that

$$\sum_{i=1}^{n} \log \hat{p}_{X|Y}(\mathbf{x_{i*}}|y_i) = \sum_{i=1}^{n} \log \sum_{s\in\mathcal{Y}} \hat{p}_{X,S|Y}(\mathbf{x_{i*}}, s|y_i) \tag{7.12}$$

The second term of Eq. (7.12) is called the incomplete log-likelihood, because it involves unknown true class labels. Because of the logarithms the optimisation problem is non-convex, meaning multiple local maxima may exist. As there exists no closed-form solution for a global maximum of Eq. (7.12), a possible solution is to use a EM algorithm [43]. It will consider S as a latent random variable to recursively build an approximation of the incomplete log-likelihood and use it to estimate the error probabilities $p_e(s)$. For the particular case of Equation (7.12), the EM algorithm alternatively (i) estimates

$$Q(p_e, p_e^{old}) = \sum_{i=1}^{n} \sum_{s\in\mathcal{Y}} \gamma(s|i) \log \hat{p}_{X,S|Y}(\mathbf{x_{i*}}, s|y_i) \tag{7.13}$$

based on the current estimate p_e^{old} (E step) and (ii) update the estimate of p_e by finding the value maximizing $Q(p_e, p_e^{old})$ (M step).

The E step consists in estimating the true class memberships using the current label noise model:

$$\gamma(s|i) = \frac{p_{X|S}(x_i|s)p_{Y|S}(y_i|s)p_S(s)}{\sum_{s\in\mathcal{S}} p_{X|S}(x_i|s)p_{Y|S}(y_i|s)p_S(s)} \tag{7.14}$$

using Equations (7.8) and (7.4).

During the M step, the following updates are made:

$$p_e(s) = \frac{1}{\Gamma(s)} \sum_{i|y_i\neq s} \gamma(s|i), \quad p_S(s) = \frac{1}{n}\sum_{i=1}^{n} \gamma(s|i). \tag{7.15}$$

To initialize the algorithm, the error probabilities $p_e(s)$ are randomly selected while the true class priors are set to the value of the label priors: $p_Y(s) = \frac{n_y}{n}$.

More details on the proposed algorithm can be found in [72].

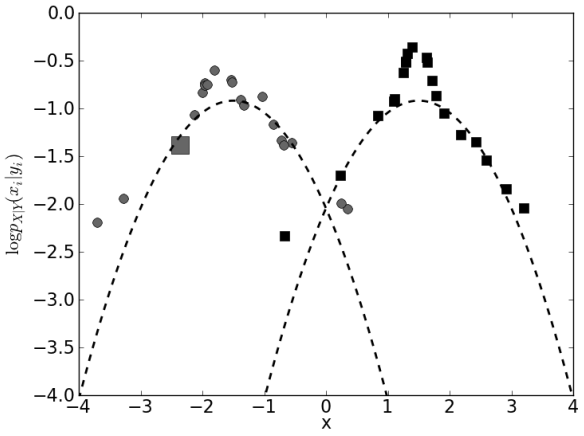


Figure 7.3: Estimates of the logarithm of the conditional probability $p_{X|Y}$ for a classification problem. Both class have a Gaussian distribution (dashed lines). Circles belong to class 0 and black squares belong to class; the large grey square sample is mislabelled.

Illustration of the proposed approach

Consider again the problem illustrated in Figure 7.2. To show the interest of the proposed methodology, Figure 7.3 shows the estimates the conditional probabilities obtained with the estimated true class labels.

The conditional probability of the mislabelled sample has increased to a value similar to the one of the samples of its true class. Besides, the estimated error probabilities are $p_e(0) = 0.02$ and $p_e(1) = 0.00$, which is close to the true error probabilities in the dataset. Eventually, the MI estimate obtained using Equation (7.7) is $\hat{I}(X;Y) = 0.61$, which is above the value obtained by the standard estimator.

7.1.6 Experiments

We now illustrate the interest of the proposed label noise-tolerant feature selection algorithm. To do so, we compare the two following approaches: i) a backward search procedure based on the MI criterion estimated with the noisy samples (BW-N) and (ii) the same backward search based on the proposed label noise-tolerant MI estimator (LNT-BW). Both algorithms are thus identical, except for the way the MI is estimated. This allows us to really measure the impact of the noise-tolerant MI estimator on the feature selection process.

The classification accuracy of a k nearest neighbors algorithm is used as the criterion of comparison between the feature selection algorithms and the experiments are performed on eleven UCI datasets [71] described in Table 3.1. For each dataset, features are first normalised to have zero mean and unit variance. Samples are then randomly split into a training and a test set whose respective sizes are 70% and 30% of the total amount of samples. Labels of the training set are then polluted by label noise: a given percentage of labels are randomly flipped. Feature selection is then conducted on the training set following the two aforementioned strategies. For comparison purposes, feature selection is also conducted on the true original labels as given in the dataset (BW-C).

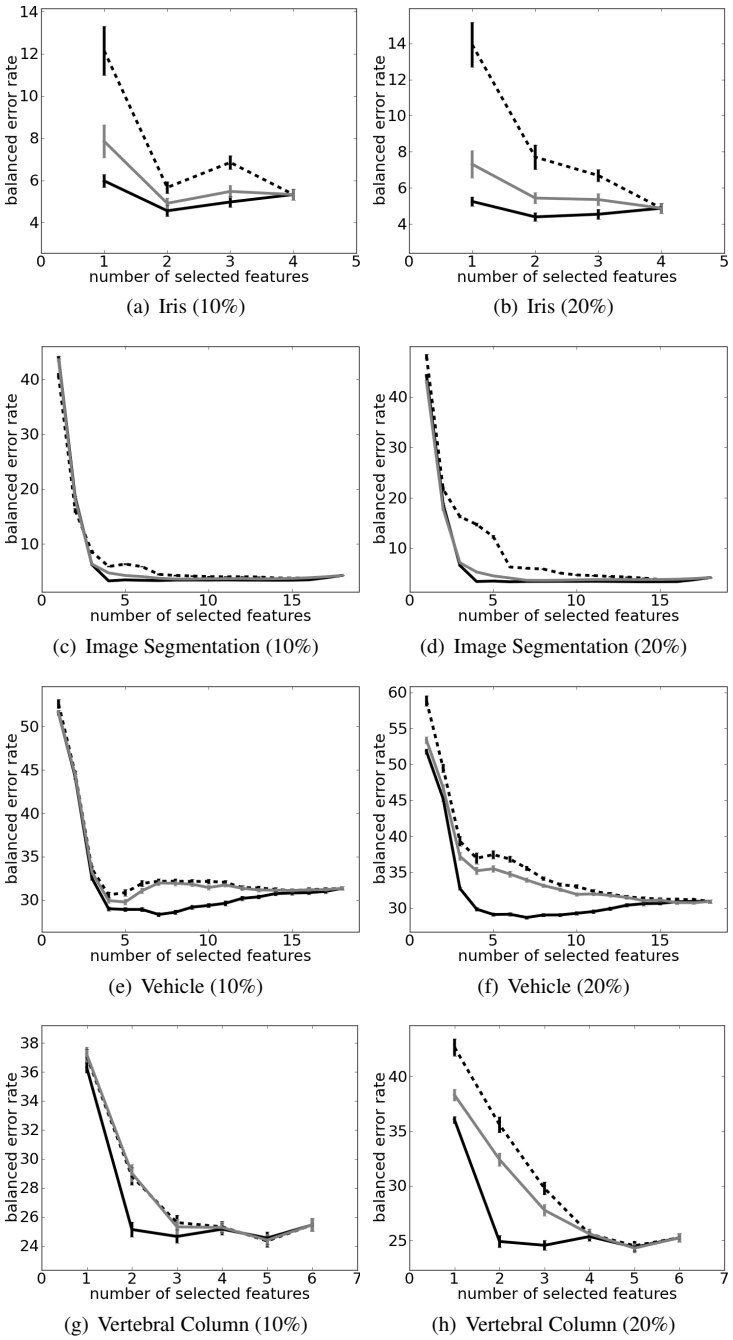
For each feature subset obtained by the feature selection algorithms, the training set with *unpolluted* labels is then used to (i) select the optimal value of the classifier meta-parameter k using ten-fold cross-validation and (ii) build the k NN classifier. The performances of the classifier is eventually obtained on the test set, which is not polluted by label noise. The exact criterion of comparison is the balanced error rate, which corresponds to the average of the percentages of misclassification for each class. Experiments are repeated 100 times. The levels of label noise are 10% and 20% of flipped labels while the parameter for the MI estimator is $k = 8$.

The parameter for the label noise modelling is $k = 3$, meaning the label noise is estimated locally. Preliminary experiments indicated that samples from minority classes tend to be considered as belonging to majority classes during the estimation of the true class memberships with larger values of k. Please refer to the related paper [72] for more details on the experimental protocol.

Figures 7.4, 7.5 and 7.6 show the results for some classification datasets introduced in Chapter 3. In those figures, the error bars show the 95% confidence interval for the means. To make the presentation of the results as clear as possible, datasets are split into two groups. Figures 7.4 and 7.5 show the results for the datasets for which the proposed feature selection method produces significantly better feature subsets than BW-N while Figure 7.6 shows the results for datasets on which both methods perform similarly.

Figures 7.4, 7.5 and 7.6 first indicate that the label noise can have a significant impact on feature selection since the observed performances are almost always significantly better when feature subsets are obtained using BW-N than when using BW-C. Large differences between BW-C and BW-N can particularly be observed in Figures 7.4(f), 7.4(h), 7.5(d) and 7.5(f). For the Ecoli and Yeast datasets, the differences are less important.

Figures 7.4 and 7.5 illustrate that the proposed feature selection method is able to considerably improve the feature selection performances compared to noise-intolerant methods. Indeed, when 20% of the labels are flipped, LNT-BW is significantly better



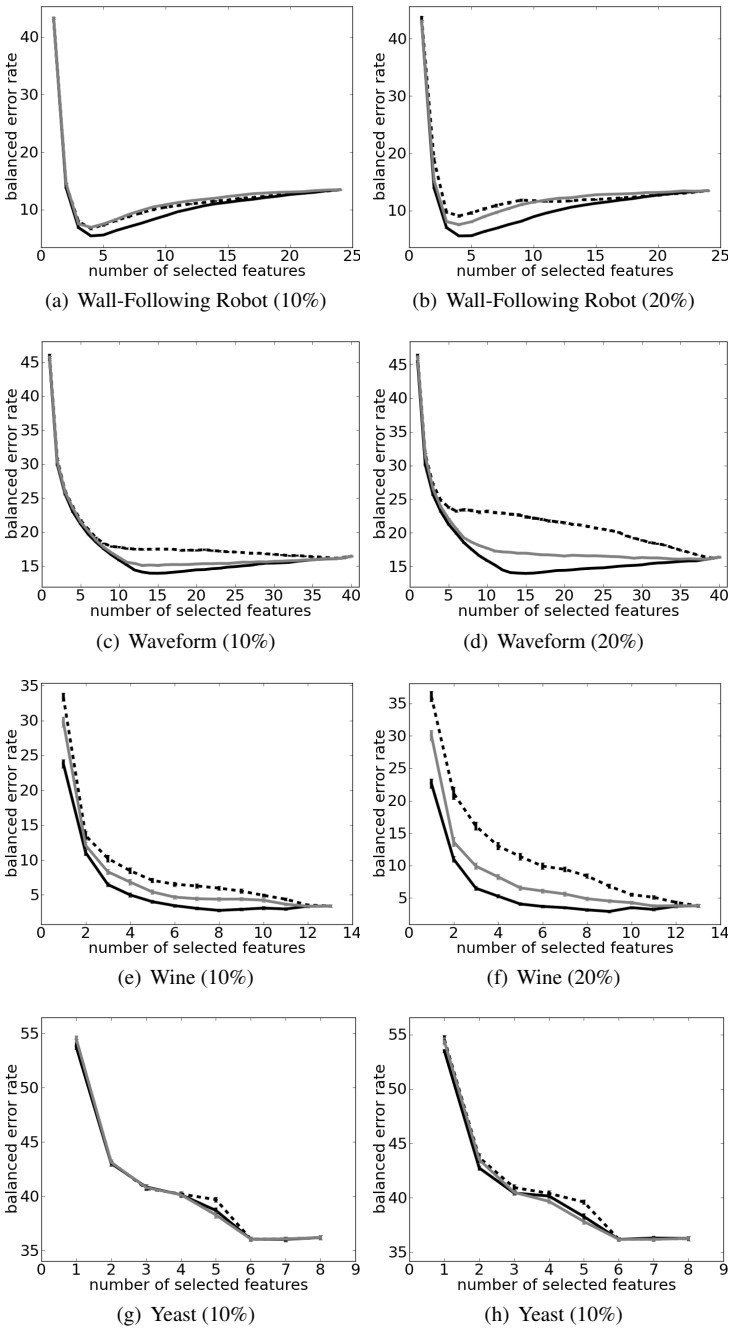
(a) Iris (10%) (b) Iris (20%)

(c) Image Segmentation (10%) (d) Image Segmentation (20%)

(e) Vehicle (10%) (f) Vehicle (20%)

(g) Vertebral Column (10%) (h) Vertebral Column (20%)

Figure 7.4: Balanced error rates for the (a-b) Iris, (c-d) Image Segmentation, (e-f) Vehicle and (g-h) Vertebral Column datasets using BW-C (plain black line), BW-N (dashed black line) and LNT-BW (plain grey line). The levels of label noise are 10% and 20% for the left and right columns, respectively.



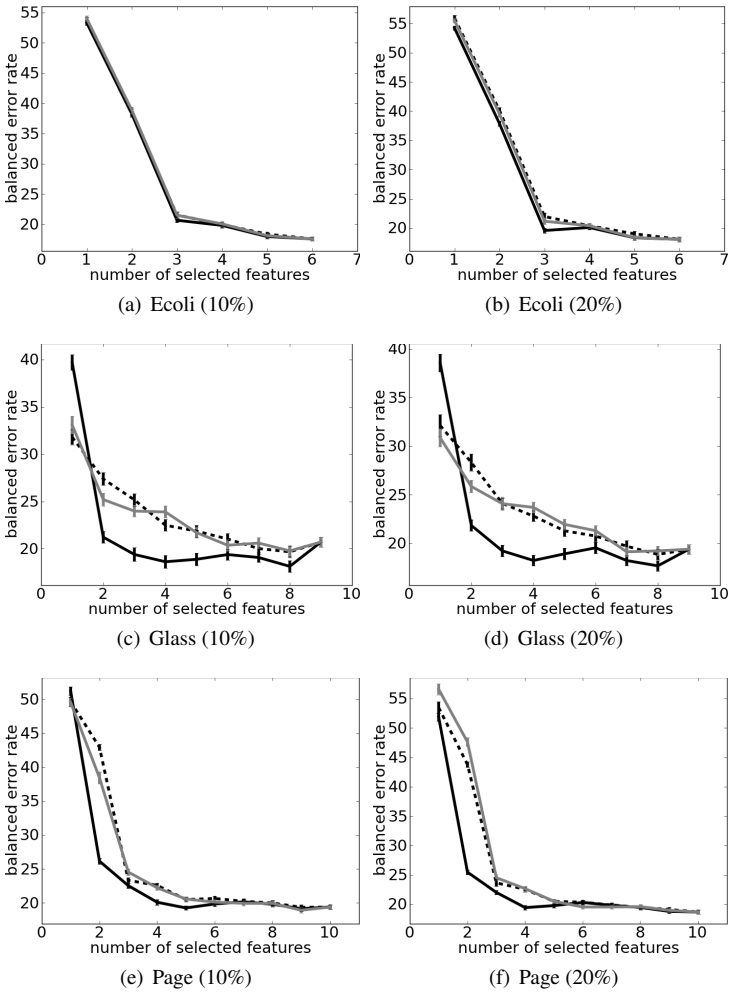
(a) Wall-Following Robot (10%) (b) Wall-Following Robot (20%)

(c) Waveform (10%) (d) Waveform (20%)

(e) Wine (10%) (f) Wine (20%)

(g) Yeast (10%) (h) Yeast (10%)

Figure 7.5: Balanced error rates for the (a-b) Wall-Following Robot, (c-d) Waveform, (e-f) Wine and (g-h) Yeast datasets using BW-C (plain black line), BW-N (dashed black line) and LNT-BW (plain grey line). The levels of label noise are 10% and 20% for the left and right columns, respectively.



(a) Ecoli (10%) (b) Ecoli (20%)

(c) Glass (10%) (d) Glass (20%)

(e) Page (10%) (f) Page (20%)

Figure 7.6: Balanced error rates for the (a-b) Ecoli, (c-d) Glass and (e-f) Page datasets using BW-C (plain black line), BW-N (dashed black line) and LNT-BW (plain grey line). The levels of label noise are 10% and 20% for the left and right columns, respectively.

than BW-N in terms of the balanced error rate for a very large number of intermediate feature subsets. Obviously, when the number of selected features gets close to d, the performances of each feature selection strategy tend to become similar. Besides, when only a small number of features are selected, the amount of information is too low to achieve satisfying and relevant classification performances. When 10% of the labels are flipped, LNT-BW remains better than BW-N, except for the Vertebral Column and the Wall Robot datasets.

For the datasets whose results are shown in Figure 7.6, the performances of the LNT-BW and BW-N methods cannot be distinguished. For the Ecoli dataset, the balanced error rates are identical for the three methods. For the Glass and Page datasets, both LNT-BW and BW-N are affected by label noise, but none of them is significantly better over a large range of feature subset sizes.

The proposed noise-tolerant feature selection algorithm thus exhibits obvious experimental advantages over its competitors in the presence of label noise. The algorithms have not been compared on noise-free datasets as in this case, the classical estimator has to be prefered. Indeed, the noise-tolerant estimator is (i) more time-consuming and (ii) more complex, which could lead to overfitting problems. It should therefore be used only when one does suspect the presence of label noise.

7.2 Feature selection for probabilistic labels

We now give some results concerning another form of uncertain supervision. We consider classification problems for which the labels are not exactly known but are rather given as probabilities over the different class labels. As an illustration, it is frequent that a human supervision is required to assign labels to sample points, for instance in the medical domain where a diagnostic has to be made based on a microarray or a radiography. As such a task can be hard to perform, experts sometimes hesitate between different labels or furnish a single class label and give a measure of the confidence they have in this label.

More precisely, this section addresses the problem of feature selection in the case where an expert gives for each sample $\mathbf{x_{i*}}$ ($i = 1 \dots n$), a probability p_{ij} to every possible class c_j ($j = 1 \dots c$); it is assumed that $\sum_{j=1}^{l} p_{ij} = 1 \, \forall i$, while this is not mandatory in practice. Such labels are called possibilistic labels in the literature. Rather than being directly given by a single expert, they can also be obtained by combining the opinion of several experts reagrding the membership of a point to a given class. As a concrete illustration, [44] mentions the recognition of some transient phenomena in EEG data, whose shapes can be really hard to distinguish from the EEG background activity, even for trained practicians. Experts are almost never sure about the presence of such phenomena but are nevertheless able to give a probability of this presence.

Probabilistic labels can also be found in fuzzy classification problems. Indeed, classes which are not well defined can often be better represented by labels measuring the degree of membership of the samples to the classes. For example, probabilistic labels can be obtained through the fuzzy c-means clustering algorithm [17].

We propose to extend the Laplacian score introduced in section 2.2 to this uncertain label framework in order to achieve feature selection in this context. The developments are organized as follows. Section 7.2.1 describes the few related works on uncertain labels. The proposed feature selection criterion, called weighted Laplacian Score (WLS), is introduced in Section 7.2.2 and is briefly experimentally tested in Section 7.2.3.

7.2.1 Related work

The problem of classification with uncertain labels has been barely addressed in the literature. The most relevant proposition of classification algorithms make use of the Dempster-Shafer theory (DS) of belief functions, which has proven to be very successfull in this context [44, 103, 34]. DS theory offers a very general and flexible framework to model beliefs about the class label of a sample. Beliefs can either take the form of probabilities such as in the present work or more general aspects as a set of possible class labels or the probability of a class with no additional information. Moreover, DS theory can conveniently be used in combination with the Transferable Belief Model [164] which permits to elegantly combine the different pieces of belief concerning the class memberships of samples.

Feature selection with imprecise class labels has similarly received very little attention in the literature. Among the few related works, let us mention [159], where the authors consider a fully supervised classification problem before relabelling automatically all the data points in order to take the classification ambiguity into account. In [179], the Hilbert-Schmidt independence criterion is used to achieve feature selection with uncertain labels. Label noise and probabilistic labels are however not studied in this paper which is rather focused on multi-label like problems.

7.2.2 The weighted Laplacian Score for probabilistic labels

We now introduce the proposed feature selection criterion for possibilistic labels. While we assume here that each class label comes with a probability for each sample, the developments could equivalently be applied to the case where an expert only provides one class label for each point and associates this label with a value of the confidence he has on this prediction.

Consider a data set containing n samples $\mathbf{x}_{i*} \in \Re^d$. Each sample actually belongs to *a single true* class c_j, $j \in 1 \dots C$ but this label is not precisely known. Instead, the

samples are associated with a probability value p_{ij} for each possible class label such that $\sum_j p_{ij} = 1 \, \forall i \in 1 \dots n$. It is then possible to compute the probability that two points $\mathbf{x_{i*}}$ and $\mathbf{x_{j*}}$ share the same *true* class label as:

$$S_{ij}^{sim} = \sum_k p_{i,k} p_{j,k}. \tag{7.16}$$

Said otherwise, $S_{i,j}^{sim}$ is the scalar product between the vectors of label probability for samples $\mathbf{x_{i*}}$ and $\mathbf{x_{j*}}$. Following the developments presented in Section 2.2, we define a diagonal matrix $D^{sim} = diag(S^{sim}\mathbf{1})$ and the graph Laplacian as $L^{sim} = D^{sim} - S^{sim}$.

Let us then proceed similarly for a matrix $S^{dis} \in \Re^{n \times n}$, corresponding to the probabilities that two samples belong to different classes:

$$S_{ij}^{dis} = 1 - \sum_k p_{i,k} p_{j,k} = 1 - S_{ij}^{sim} \tag{7.17}$$

and let us define $D^{dis} = diag(S^{dis}\mathbf{1})$ as well as $L^{dis} = D^{dis} - S^{dis}$.

The importance of each feature $\mathbf{f_r}$ is eventually computed by the weighted Laplacian score (WLS) that we define as

$$WLS_r = \frac{\mathbf{f_r^T} L^{sim} \mathbf{f_r}}{\mathbf{f_r^T} L^{dis} \mathbf{f_r}}. \tag{7.18}$$

Features are ranked according to this score, in increasing order.

The idea of the criterion is thus to select features according to which samples belonging to the same class with a high probability are close while samples belonging to different classes are far away from each other. The alternative formulation proposed in Section 2.2.2 can therefore justify why the proposed criterion accurately translates these feature quality criteria.

7.2.3 Experimental results

The performances of the proposed WLS are now briefly illustrated on both artificial and real-world problems. Before presenting the results, we detail the procedure used to simulate the uncertainty of an expert and the possible errors (s)he makes when labelling data points. First, n values β_i are drawn from a Beta distribution of mean μ and variance $\sigma = 0.1$. With probability β_i, the *true* label l_j of sample $\mathbf{x_{i*}}$ is switched to one of the other possible labels $l_{s \neq j}$, with the same probability for each label. The probability value associated with the true class label is set to $p_{ij} = 1 - \beta_i$ and the probability of the possibly switched label is set to $p_{is} = \beta_i$.

Similarly to what has been done for the label noise problem (see Section 7.1.6),

the performances of the WLS are compared to two similar strategies which both neglect the uncertainty of the class labels. The first strategy consists in using the labels $y_i^{error}(i = 1 \dots n)$ obtained after the procedure described above. The second one considers only one label (which has thus a 100% probability) for each sample point. This label is the most probable label $y_i^{max}(i = 1 \dots n)$ after the switch procedure described above.

Two matrices $S^{sim,error}$ (resp. $S^{sim,max}$) and $S^{dis,error}$ (resp. $S^{dis,max}$) are built from the switched (resp. most probable) labels by setting

$$S_{i,j}^{sim,error(max)} = \begin{cases} 1 & \text{if } y_i^{error(max)} = y_j^{error(max)} \\ 0 & \text{otherwise} \end{cases} \tag{7.19}$$

and $S_{ij}^{dis,error(max)} = 1 - S_{ij}^{sim,error(max)}$. The Laplacian scores for both problems are then constructed as usual, and the relevance of the features is computed similarly to Eq. (7.18).

Experiments are first conducted on two artificial data sets, for which the relevant features are known.

Both problems have 10 randomly and uniformly distributed features $\mathbf{x}_{*1} \dots \mathbf{x}_{*10}$ and a sample size of 300. The true labels are generated by discretizing the following outputs:

$$Y_1 = \cos(2\mathbf{x}_{*1}) \cos(\mathbf{x}_{*2}) \exp(2\mathbf{x}_{*3}) \exp(2\mathbf{x}_{*4}) \tag{7.20}$$

and

$$Y_2 = 10 \sin(\mathbf{x}_{*1}\, \mathbf{x}_{*2}) + 20(\mathbf{x}_{*3} - 0.5)^2 + 10\mathbf{x}_{*4} + 5\mathbf{x}_{*5}, \tag{7.21}$$

this last problem being derived from [75]. Precisely, the sample points are ranked in increasing value of Y_1 or Y_2 and then respectively divided into three and two classes containing the same number of consecutive points.

The criterion of comparison is the percentage of relevant features among the d_r best ranked features, where d_r is the number of relevant features for a given problem. Both artificial datasets have been randomly generated 50 times to repeat the experiments.

Tables 7.1 and 7.2 present the results obtained with various values of μ or equivalently for various levels of uncertainty. It can be observed that for both problems and each value of μ, the proposed WLS always outperforms its two competitors by more accurately detecting the relevant features. Besides, the WLS performances appear to be very satisfactory with, for example, more than 90% of relevant features selected for $\mu \leq 0.3$. Such observations indicate the adequateness of the proposed feature selection strategy for problems with uncertain labels. Generally, the method based on the observed class labels y^{error} performs the worst.

μ	WLS	y^{max}	y^{error}
0.25	95.5	94.5	88.5
0.30	95	89	87
0.35	89.5	82.5	82
0.40	84.5	75	74

Table 7.1: Percentage of relevant features obtained by the three feature selection techniques on the Y_1 data set.

μ	WLS	y^{max}	y^{error}
0.25	96.8	93.6	91.6
0.30	94	90	84.8
0.35	84.8	79.2	74.4
0.40	76.4	72.4	59.6

Table 7.2: Percentage of relevant features obtained by the three feature selection techniques on the Y_2 data set.

To further assess the interest of the proposed feature selection criterion, experiments are also conducted on two real-world datasets, namely *Iris* and *Mines vs. Rocks*, both available from the UCI repository [71].

The criterion of comparison is the accuracy of a 1-nearest neighbor (1NN) classifier using the features selected by the three methods. The exact class labels will be used for the classification step while the feature selection will be achieved with the possibly permuted labels and the expert information. This way the ability of the methods to select relevant features for the true original problems can be compared. Figures 7.7 and 7.8 show the accuracy of the 1NN classifier as a function of the number of selected features for $\mu = 20\%$ and $\mu = 30\%$. For each data set and each value of the parameter μ, the label permutation step is randomly repeated 50 times and the results are obtained as an average over a five-fold cross-test procedure.

The results tend to confirm the interest of the proposed *WLS*, even if the impact is less obvious than for the artificial problems. For the *Iris* dataset, the WLS never leads to lower classification performances than its two competitors for any number of features and both contamination rates, even if the differences in performance are larger when $\mu = 30\%$. The *Mines* data set confirms that the *WLS* detects relevant features more quickly than the methods which do not take label uncertainty into account. For this dataset, the *WLS* outperfoms both its competitors for the first 12 and the first 16 features when μ equals 20% and 30%, respectively. For both datasets, there exists a reduced number of features leading to better performances than the full set of features.

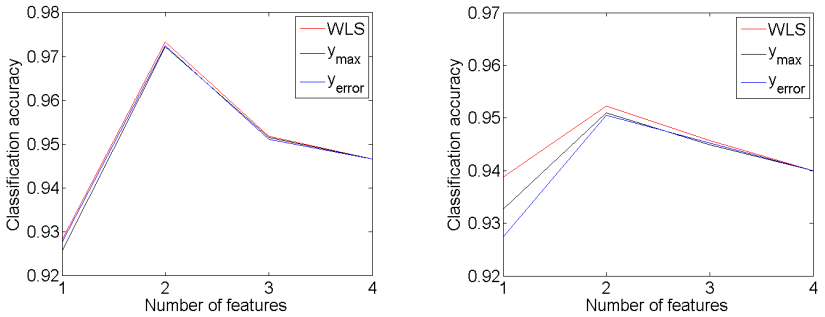


Figure 7.7: Accuracy of a 1NN classifier as a function of the number of selected features for the *Iris* dataset with $\mu = 0.2$ (left) and $\mu = 0.3$ (right).

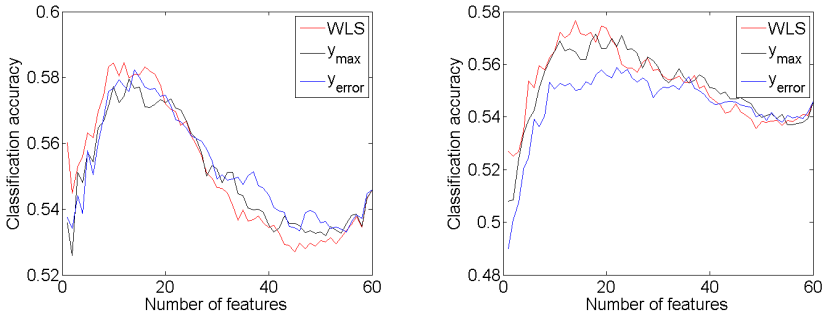


Figure 7.8: Accuracy of a 1NN classifier as a function of the number of selected features for the *Mines vs. Rocks* dataset with $\mu = 0.2$ (left) and $\mu = 0.3$ (right).

7.3 Conclusion

This chapter tackles the problem of wrongly or unprecisely labelled datasets, which can be frequently encountered in practice. The first part of the chapter is dedicated to the label noise problem, i.e. the situation where some of the provided class labels are wrong. To achieve feature selection in this context, a way to make the MI estimator proposed in [114] and presented in Section 2.1.3 tolerant to label noise is proposed. The idea is to obtain an estimation of the true class membership of the samples using a label noise model and an EM algorithm. These memberships are then used to adapt the volume of an hypersphere supposed to contain exactly k nearest neighbors of a given class in the original algorithm. Results indicate that (i) the label noise actually affects the feature selection procedure and that (ii) the proposed approach is efficent in taking the possibility of label noise into account.

The second part is dedicated to the problems where class memberships are given as probability values for each sample. An approach based on the Laplacian score is proposed to take the label uncertainty into account. First results indicate the potential interest of the method for such problems.

Part III

Theoretical considerations on the entropy and the mutual information

Chapter 8

On the relationships between mutual information and risk classification in the context of feature selection

Contents

In the second part of the thesis, the mutual information has been widely used as a relevance criterion for feature selection. As mentioned in Section 2.1, numerous arguments actually motivate this choice. Besides its ability to detect non-linear dependencies, its natural definition for multivariate variables and its good performances in practice, the mutual information is also closely related to the classification risk through the Fano and Hellman-Raviv bounds, as detailed in Section 2.1.2.

Despite what is sometimes written in the literature, the aforementioned bounds do not guarantee that selecting a subset of features maximising the mutual information is always equivalent to selecting a subset of features minimizing the classification risk, the quantity one is generally interested in. This chapter first illustrates this observation in Section 8.1.

We then try to more precisely characterize the problems for which the mutual information is likely to be not optimal as a feature selection criterion, and what can be the consequences in terms of risk or misclassification probability. Extensive experiments are carried out to this end, on both continuous and categorical datasets, some being artificial and other ones corresponding to real-world problems. The results are presented in Section 8.2. The global objective of these developments is to eventually assess what is the real interest of the mutual information as a feature selection criterion, despite the fact that it is non-optimal from a classification risk point of view.

This chapter ends with the presentation of two risk estimators, based on the same density estimators as the nearest neighbors-based MI estimator. They allow one to directly estimate the risk rather than the mutual information, for instance in a feature selection context. The developments and brief experimental results are given in Section 8.5.

This chapter is based on first results presented in [7], later extended in [73] as well as [80].

8.1 Inadequacy of the mutual information for feature selection

When facing a classification problem, the main objective one is often interested in is to reduce as much as possible the classification risk, or equivalently, to build prediction models which are as accurate as possible. In this case, feature selection should also be conducted with the same objective. The quality of a particular feature subset should therefore be measured through the risk induced by this subset. More precisely, we are interested in the optimal (Bayes) risk P_e, which gives a lower bound on the misclassification. Using the bounds illustrated in Figure 2.1 and the related considerations, some papers including [67, 24, 136] conclude that if the feature subset fs_1 has a higher

mutual information with the output than the subset fs_2, then fs_1 will lead to a smaller classification risk P_e than fs_2. Those papers therefore conclude that the mutual information can safely be used as a proxy for the risk as a feature selection criterion. In this section, we show through a simple example that such a conclusion is not always true in practice and give a condition for the optimality of the mutual information. We eventually derive a bound which relates the maximum value of mutual information between two feature subsets and the output to the difference of classification risk induced by the use of one subset instead of the other one.

8.1.1 Illustration of mutual information failure for feature selection

From Figure 2.1 which illustrates the bounds on the Bayes classification risk given the conditional entropy $H(Y|X)$, it is easy to understand that for different values of $H(Y|X)$ (and thus for different values of the mutual information $I(X;Y)$), the bounds are likely to strongly overlap. Therefore, P_e could obviously take values in the same intervals for two different subset of features X_1 and X_2 being compared. Because of this overlap, it could be possible that $I(X_1;Y) > I(X_2;Y)$ while X_1 would simultaneously lead to a higher classification risk than X_2. This would be considered as a mutual information failure for feature selection, in the sense that the choice of a feature subset would not be optimal from the Bayes risk point of view. We show through the simple following example that such a failure can happen in practice.

Let us imagine a situation where a practician has to make a diagnosis Y, choosing between two diseases y_1 and y_2 which have the same prior probability ($p_Y(y_1) = p_Y(y_2) = 0.5$). To help him in his task, the practician can choose to run only one out of two possible tests. Both tests X_1 and X_2 are binary and can therefore only take the values 0 and 1. The practician has thus clearly to deal with a feature selection problem.

Based on previous cases, the conditional distributions $P(X_i|Y)$ of both tests have been obtained:

	$Y=0$	$Y=1$
$X_1=0$	0.287	0.758
$X_1=1$	0.713	0.242

and

	$Y=0$	$Y=1$
$X_2=0$	0.627	0.999
$X_2=1$	0.373	0.001

,

where the rows correspond to the possible values of X_i and the columns correspond to the possible values of Y. Through marginalisation and the Bayes theorem, one can easily obtain $P(Y|X_i)$, the posterior probabilities of both diseases given each of the

tests:

	$X_1 = 0$	$X_1 = 1$
$Y = 0$	0.275	0.746
$Y = 1$	0.725	0.254

and

	$X_2 = 0$	$X_2 = 1$
$Y = 0$	0.385	0.999
$Y = 1$	0.615	0.001

.

Those posterior probabilities indicate that X_1 allows discriminating quite well between the classes, whether its output is 0 or 1. The remaining misclassification probability is almost equal for both outputs ($P_e = 0.275$ if $X_1 = 0$ and $P_e = 0.254$ if $X_1 = 1$). On the contrary X_2, allows discriminating almost perfectly when its output is 1 since $P_e = 0.001$ in this case. When $X_2 = 0$, this feature is much less discriminative than X_1 as the misclassification probability is still $P_e = 0.385$ in that case.

Eventually, it is possible to compute that for the first test X_1, the classification risk $P_e = 0.265$ while the mutual information $I(X_1;Y) = 0.167$. Using the second test, it can be obtained that $P_e = 0.314$ and $I(X_2;Y) = 0.217$. It thus appears that the mutual information between X_2 and Y is larger than the one between X_1 and Y. However, X_2 also incurs a higher risk P_e than X_1; selecting the test to perform (X_2) based the on mutual information thus leads here to an increased classification risk.

The example is more clearly illustrated in Figure 8.1. Again, it is shown that $I(X_2;Y) = H(Y) - H(Y|X_2)$ is larger than $I(X_1;Y) = H(Y) - H(Y|X_1)$ while $P_e(X_2)$ is larger than $P_e(X_1)$. As expected, both points lie between the bounds.

8.1.2 A condition of optimality

As illustrated by the previous example, it is not always possible in practice to asses the interest of the mutual information for feature selection. In some situations, however, the optimality of the mutual information can be guaranteed. Let X_1 and X_2 be two features sets that have to be compared. If the value of the Hellman-Raviv bound for X_1 is smaller than the value of the strong Fano bound for X_2, then the bounds in Figure 8.1 show that X_1 leads to a smaller classification risk P_e than X_2 does. This actually translates the quite natural idea that if the conditional entropies (or the mutual information) for two subsets are different enough, than there cannot be any overlap between the possible values of P_e for these subsets; ranking feature subsets with the mutual information criterion is optimal in that case.

In the example of Section 8.1.1, the aforementioned condition is violated since the Fano bound for X_1 is $P_e \geq 0.264$, whereas the Hellman-Raviv bound for X_2 is $P_e \leq 0.391$; mutual information cannot threfore be guaranteed to be optimal to choose

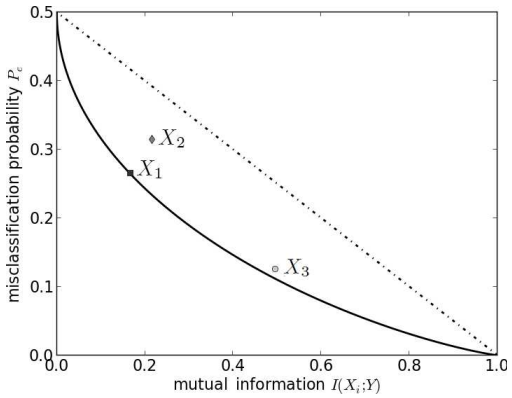


Figure 8.1: Example of mutual information failure for feature selection, with the strong Fano bound (plain line) and the Hellman-Raviv bound (dash-dotted line).

between X_1 and X_2. On the contrary, Figure 8.1 shows an other candidate subset X_3 for which the Hellman-Raviv bound is $P_e \leq 0.252$. Feature subset X_3 thus respects the optimality condition and is guaranteed to be a better choice.

8.1.3 Upper bound on the misclassification probability loss

We end this section by giving an upper bound for the difference in misclassification probability or classification risk in case of a failure. This difference is called the misclassification probability loss or the risk loss.

The worst possible situation for a mutual information failure is obviously when the two compared feature subsets have almost identical mutual information with the output while one of the subsets incurs a maximum risk (corresponding to the Hellman-Raviv bound) and the other one incurs a minimum risk (corresponding to the strong Fano bound). In this case, the risk loss is simply the difference between the Hellman-Raviv bound and the strong Fano bound. In our example, the risk loss is bounded by 0.159 for X_2 and the actual risk loss is 0.049. It is interesting to note that the bounds on the risk loss are tightened for small or large values of the mutual information. This suggests that mutual information failures have less important consequences in these cases. Said otherwise, making a wrong choice between two extremely informative or two completely irrelevant feature subsets has no major consequences in practice, which is quite easy to understand.

Note eventually that the bound on the classification risk loss is concave with respect to $I(X;Y)$, since the Hellman-Raviv and Fano bounds are increasing concave and

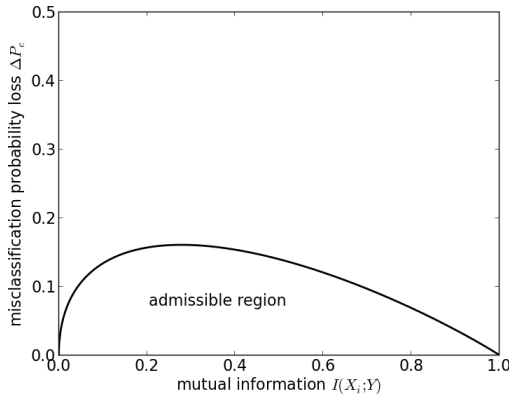


Figure 8.2: Theoretical upper bound on the misclassification probability loss for binary classification problems with balanced classes.

convex with respect to $H(Y|X)$, respectively. Figure 8.2 shows the upper bound on the classification risk loss for a binary classification problem with balanced classes.

8.2 Experimental results

This section illustrates experimentally the connections between the classification risk and the mutual information. More precisely, the potential failure of mutual information is studied on a variety of problems of very different nature to provide the most complete analysis as possible.

8.2.1 Artificial discrete datasets

In this section, three artificial monovariate binary classification problems are considered. More precisely, it is assumed that a classifier is given only one discrete feature with 2^d possible modalities. The reason for this choice is that such a feature is actually equivalent to the combination of d binary features. We thus actually consider classification problems based on d binary features.

We consider three datasets with different domain sizes for the discrete feature X (depending on d) and different prior distributions for the conditional probabilities $p_{X|Y}$. The idea is to obtain classification problems of different levels of difficulty. For each problem, the idea is to generate and compare a huge number of pairs of features, in terms of both mutual information with the output and classification risk.

To generate the problems, we draw conditional probabilities $p_{X|Y}(X = x_i|Y)$ for each feature from a symmetric Dirichlet distribution

$$f(x_1,\ldots,x_m|\alpha) = \Gamma(\alpha m)\prod_{i=1}^{m}\frac{x_i^{\alpha-1}}{\Gamma(\alpha)} \tag{8.1}$$

where x_i denotes the i^{th} modality of feature X, Γ denotes the gamma function and α is the concentration parameter. The interpretation of this parameter is the following one. For large values of α, the distribution will amost be constant and the generated probabilities of X given Y will be very similar. On the contrary, for small values of α, the distribution will be more concentrated and only one probability will eventually be non-zero. The classification problems are thus expected to be easier with small values of α. Besides, it can also be expected that classes are likely to be more easily discriminated in high-dimensional spaces. Therefore, three families of problems are considered $\langle m = 2\ (d = 1), \alpha = 4\rangle$, $\langle m = 8\ (d = 3), \alpha = 1\rangle$ and $\langle m = 128\ (d = 7), \alpha = 0.06\rangle$; they correspond respectively to high, medium and low levels of difficulty. Both classes have the same prior, i.e. $P(Y = y) = \frac{1}{2}$ for each y, in order to avoid any class imbalance effect. It should be noted that the relative difficulty of the three problems is shown to comply with intuition in Figures 8.3(a), 8.4(a) and 8.5(a).

For each family of problem, 10^6 pairs of features are generated, which are compared in terms of both mutual information and risk. These quantities can be computed exactly because all the required probabilities are known. For each comparison, we check whether there is a mutual information failure. In this case, the risk loss ΔP_e is also computed.

The results are presented in Figures 8.3, 8.4 and 8.5. The first row shows, from left to right, a histogram of the misclassification probability, a histogram of the mutual information and a two-dimensional histogram of these two quantities where the opacity of the hexagonal bins indicates the number of samples in the bin. The second row shows an estimate of the conditional probability of failure given P_e and given the mutual information, as well as a histogram of ΔP_e in case of a failure. Eventually, in the case of a failure, the third row shows two-dimensional histograms of (i) P_e and ΔP_e, (ii) the mutual information and ΔP_e and (iii) the mutual information difference and ΔP_e. In the last two rows, both the mutual information and the misclassification probability or the risk P_e are those of the feature which is selected using mutual information. This is quite intuitive since we are mainly interested in studying when selecting a feature by mutual information is likely to be a bad idea.

Results

As expected, classes are very difficult to discriminate in the setting $m = 2$ and $\alpha = 4$; this is confirmed by Figures 8.3(a) and 8.3(b) where we see that the risk is generally high and the mutual information generally low. However, only 0.6% of the comparisons lead to a failure. The conditional probability of failure given both P_e and the MI remains small, as can be seen in Figures 8.3(d) and 8.3(e). It can also be observed that the failure probability decreases for small mutual information values and large misclassification probabilities. The losses are also very limited, since Figure 8.3(f) shows that 95% of the misclassification probability losses remain below 1.5%. The loss decreases for small mutual information values and large misclassification probabilities, as illustrated in Figures 8.3(g) and 8.3(h). Eventually, Figure 8.3(i) illustrates the important fact that failures mostly occur when comparing pairs of features which are close in terms of both mutual information and misclassification probability.

For $m = 8$ and $\alpha = 1$, classes are a little easier to determine, as seen in Figures 8.4(a) and 8.4(b). The percentage of failure in this setting is 4.6 and is thus higher than for $m = 2$ and $\alpha = 4$. The conditional probability of failure is also larger (see Figure 8.4(d) and 8.4(e)); Figure 8.4(f) shows that the misclassification probability loss is larger, but remains below 4.3% in 95% of the cases a failure occur. Eventually, Figures 8.4(g), 8.4(h) and 8.4(i) lead to similar conclusions than what has been observed with $m = 2$ and $\alpha = 4$.

Classes are quite easy to separate for $m = 128$ and $\alpha = 0.06$, as can be seen in Figures 8.5(a) and 8.5(b). A percentage of failure of 3.8% is observed and the conditional failure probability decreases for large mutual information values and small misclassification probabilities. Similarly, Figures 8.5(g) and 8.5(h) show that ΔP_e decreases for large values of mutual information and small risk values. In Figure 8.5(f), it can be shown that 95% of the misclassification probability losses remain below 1.5%.

Discussion for artificial discrete problems

Mutual information failures is observed in a very small percentage of the experiments. Besides, these failures do not incur important misclassification probability losses; the consequences of the failures are thus not too important. Failures actually appear to be more frequent when the classes are moderately difficult to discriminate, i.e. when the compared features have a moderate value of mutual information with the output and lead to a moderate misclassification probability. The misclassification probability loss is also larger in such cases. Eventually, for all problems, failures mostly occur for pairs of features which are close in terms of both mutual information and misclassification probability. This indicates that if the value of the MI is very different for two feature subsets, using this criterion for feature selection is actually safe from a risk

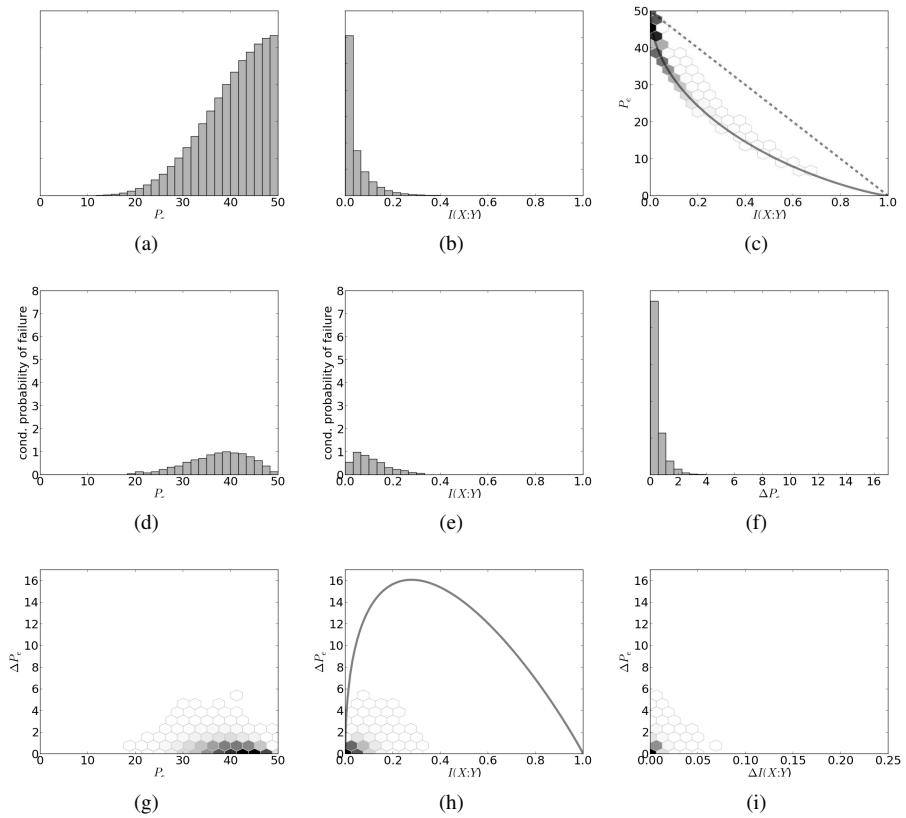


Figure 8.3: Results for artificial binary classification problems with a 2-value discrete feature and a Dirichlet prior with $\alpha = 4$ for the conditional probabilities $P(X = x_i|Y)$: histogram of the misclassification probability (a), histogram of the mutual information (b), two-dimensional histogram of these two quantities (c), estimate of the conditional probability of failure given P_e (d) and given the mutual information (e), histogram of ΔP_e in case of a failure (f), histogram of P_e and ΔP_e (g), histogram of the mutual information and ΔP_e (h) and histogram of the mutual information difference and ΔP_e (g).

perspective.

8.2.2 Artificial continuous datasets

Similarly to what has been done in Section 8.2.1, three continuous artificial problems are now considered to simulate low, medium and high levels of difficulty. All are uni-

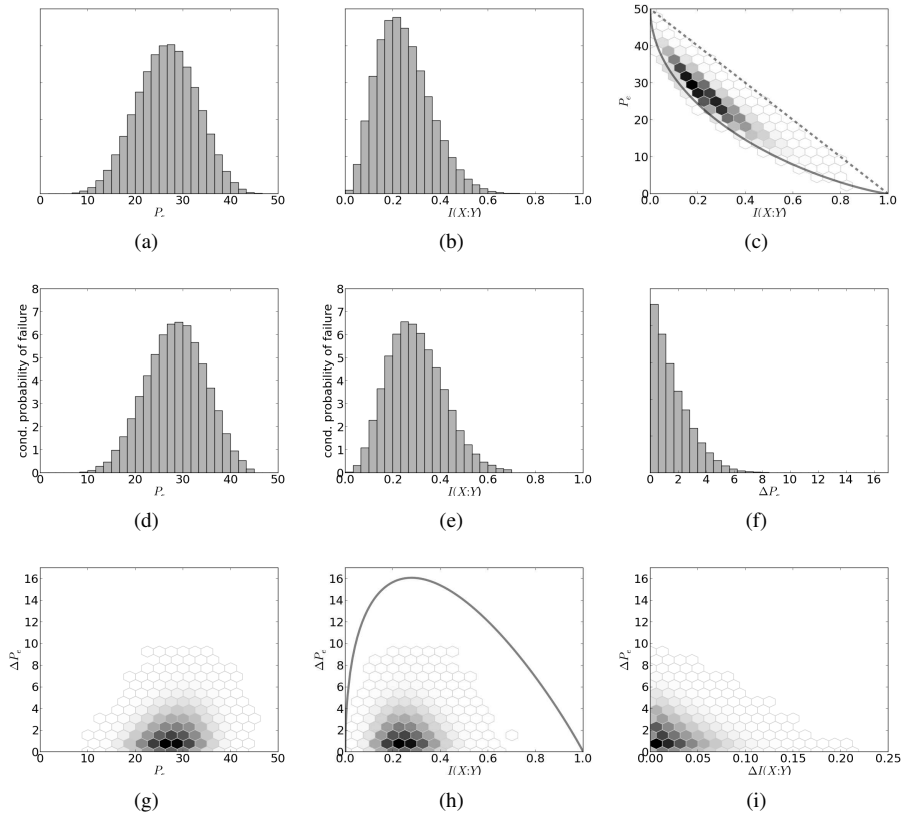


Figure 8.4: Results for artificial binary classification problems with a 8-value discrete feature and a Dirichlet prior with $\alpha = 1$ for the conditional probabilities $P(X = x_i|Y)$: histogram of the misclassification probability (a), histogram of the mutual information (b), two-dimensional histogram of these two quantities (c), estimate of the conditional probability of failure given P_e (d) and given the mutual information (e), histogram of ΔP_e in case of a failure (f), histogram of P_e and ΔP_e (g), histogram of the mutual information and ΔP_e (h) and histogram of the mutual information difference and ΔP_e (g).

variate and correspond to three-class classification problems. Each generated feature follows a class-wise unidimensional Gaussian conditional distribution and the standard deviations of the feature values are randomly drawn from a gamma distribution

$$f(\sigma|k,\theta) = \frac{\sigma^{k-1}}{\theta^k \Gamma(k)} e^{-\frac{\sigma}{\theta}}, \tag{8.2}$$

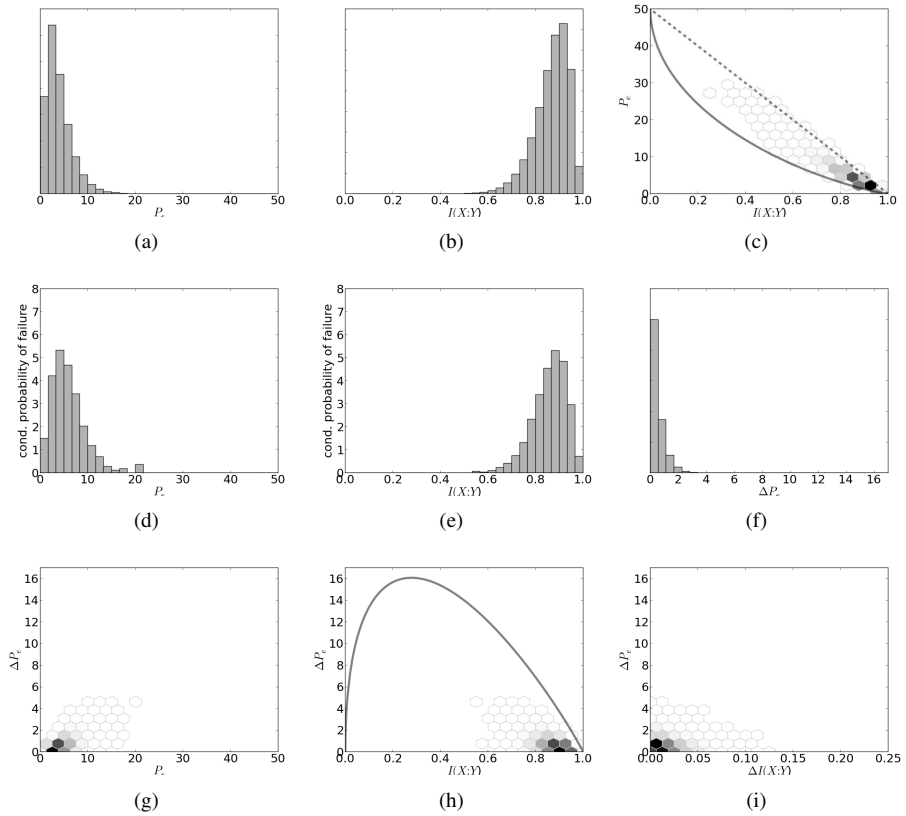
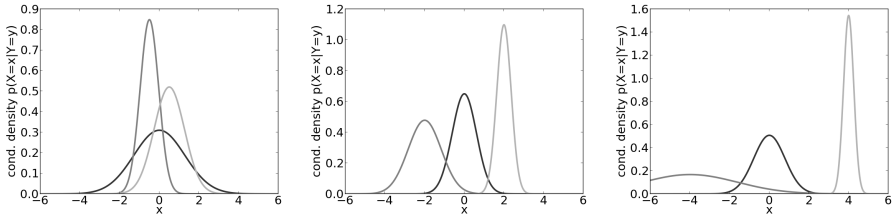


Figure 8.5: Results for artificial binary classification problems with a 128-value discrete feature and a Dirichlet prior with $\alpha = 0.06$ for the conditional probabilities $P(X = x_i|Y)$: histogram of the misclassification probability (a), histogram of the mutual information (b), two-dimensional histogram of these two quantities (c), estimate of the conditional probability of failure given P_e (d) and given the mutual information (e), histogram of ΔP_e in case of a failure (f), histogram of P_e and ΔP_e (g), histogram of the mutual information and ΔP_e (h) and histogram of the mutual information difference and ΔP_e (g).

where k is the shape parameter and θ is the scale parameter. The values $k = 2$ and $\theta = 0.5$ are used in the experiments. The feature values for each class are centered on three different values given by $\mu_0 = -\Delta$, $\mu_1 = 0$ and $\mu_2 = \Delta$, where Δ is a parameter controling the difficulty of the problem. If the value of Δ is large, the classes are well separated and the problem is easy. Small values of Δ lead to overlapping Gaussian distributions and therefore correspond to difficult problems. Here, the three problems



(a) $\Delta = 0.5$ (b) $\Delta = 2$ (c) $\Delta = 4$

Figure 8.6: Examples of the three class-conditionnal feature distribution. Standard deviations of the Gaussian distributions are drawn at random from a gamma distribution with $k = 2$ and $\theta = 0.5$; the centers are respectively set to $\Delta = 0.5$, $\Delta = 2$ and $\Delta = 4$. Subfigures correspond to a: histogram of the misclassification probability (a), histogram of the mutual information (b), two-dimensional histogram of these two quantities (c), estimate of the conditional probability of failure given P_e (d) and given the mutual information (e), histogram of ΔP_e in case of a failure (f), histogram of P_e and ΔP_e (g), histogram of the mutual information and ΔP_e (h) and histogram of the mutual information difference and ΔP_e (g).

are obtained by considering the following values: $\Delta = 0.5$, $\Delta = 2$ and $\Delta = 4$; the class priors are uniform, i.e. $P(Y = y) = \frac{1}{3}$ for each y. The class distribution and therefore the difficulty of the three problems is illustrated on Figure 8.6.

Again, 10^6 pairs of features are generated for each problem. Each feature distribution is actually a mixture of Gaussian distributions, making it impossible to obtain closed-form expressions for the entropy $H(X)$ and the mutual information. However, the conditional entropy $H(Y|X)$ and the conditional probabilities $P(X|Y)$ can be computed analytically. To address this problem, 10^4 samples are drawn from each class for each feature. The conditional probabilities $P(X|Y)$ are then computed analytically and the samples are used to estimate the required quantities, in order to obtain an estimate of the mutual information and the misclassification probability. This procedure is expected to be accurate, given the large number of samples and the low dimensionality of the data. However, to prevent the results from being affected by estimation errors, failures with a mutual information difference below 0.01 or a misclassification probability loss below 0.1% are not taken into account. The quantities presented in Figures 8.7, 8.8 and 8.9 correspond to the ones shown for discrete problems.

Results

For $\Delta = 0.5$, Figures 8.7(a) and 8.7(b) confirm that the three classes are quite difficult to discriminate. A failure occurs in 1.2% of the cases and the failure probability is low

for extreme (i.e. small or large) mutual information values and extreme classification risk (see Figures 8.7(d) and 8.7(e)). In Figure 8.7(f), it can be observed that 95% of the misclassification probability losses remain below 3.2%. From Figures 8.7(g) and 8.7(h), it can be deduced that the misclassification probability loss decreases for extreme mutual information values and misclassification probabilities. Eventually, Figure 8.7(i) shows that failures occur mainly for pairs of features which are close in terms of both mutual information and misclassification probability.

For $\Delta = 2$, classification is easier, as illustrated in Figures 8.8(a) and 8.8(b). The probability of failure is 0.9 and it decreases as mutual information values increase and the misclassification probabilities decrease (Figures 8.8(d) and 8.8(e)). Figure 8.8(f) shows that 95% of the misclassification probability losses remain below 2.7%. In Figures 8.8(g) and 8.8(h), ΔP_e decreases for large mutual information values and small misclassification probabilities. Failures occur for pairs of features which are close in terms of both mutual information and misclassification probability (see 8.8(i)).

The setting $\Delta = 4$ corresponds to the easiest problem. The percentage of failure is only 0.2 and the failure probability decreases for large mutual information values and small misclassification probabilities (see Figures 8.9(d) and 8.9(e)). In Figure 8.9(f), it is shown that 95% of the misclassification probability losses remain below 1%. Figures 8.9(g), 8.9(h) and 8.9(i) lead to similar conclusions than for $\Delta = 2$.

Discussion for artificial continuous problems

The conclusions of the above experiments are very close to those already drawn in Section 8.2.1 for discrete problems. The failure probability for the MI is very low, and the associated risk losses are very limited. Again, failures are more probable and have more important consequences when classes are moderately difficult to discriminate, i.e. for intermediate values of mutual information and misclassification probability. For all three problems, failures occur essentially for pairs of features which are close in terms of both mutual information and misclassification probability.

8.2.3 Real-world continuous datasets

To get insights into the adequacy of mutual information for feature selection in the case of real-world problems, three datasets from the UCI [71] repository are now considered. All are balanced and contain an important number of samples in order to get reliable estimates of the mutual information and the classification risk: Handwritten Digits, Wall-following Robot and Waveform.

Here, feature selection is conducted with a greedy forward search algorithm using the mutual information criterion. The objective is to determine if mutual information failures are more likely to happen at certain steps of greedy multivariate feature

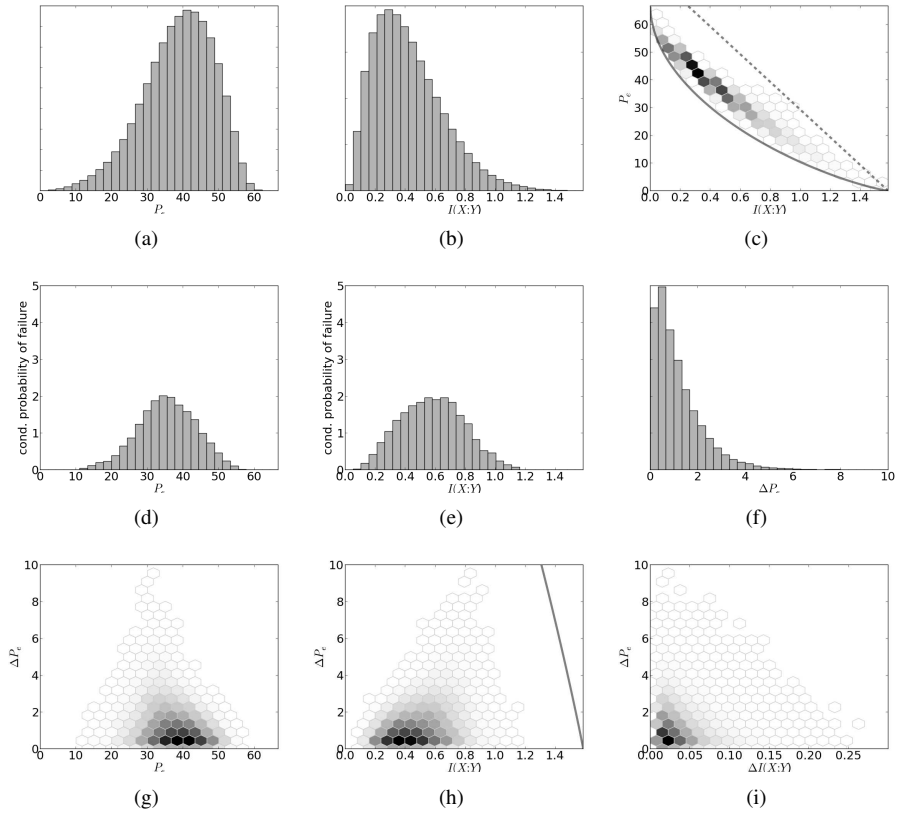


Figure 8.7: Results for an artificial three-class problems with one continuous feature. Class distributions are Gaussians centered at $x = -0.5$, $x = 0$ and $x = 0.5$. Mutual information values and misclassification probabilities are estimated using 10^4 samples from each class. Subfigures correspond to: an histogram of the misclassification probability (a), an histogram of the mutual information (b), a two-dimensional histogram of these two quantities (c), an estimate of the conditional probability of failure given P_e (d) and given the mutual information (e), an histogram of ΔP_e in case of a failure (f), an histogram of P_e and ΔP_e (g), an histogram of the mutual information and ΔP_e (h) and an histogram of the mutual information difference and ΔP_e (g).

selection procedures, such procedures being very popular in practice.

More precisely, the experiments are conducted as follows, and repeated 5000 times for each dataset. First, 10 features are chosen at random among all the features; the goal is to obtain a sub-problem of lower dimension but with similar characteristics to the original problem. We then conduct the forward search to obtain 10 feature subsets

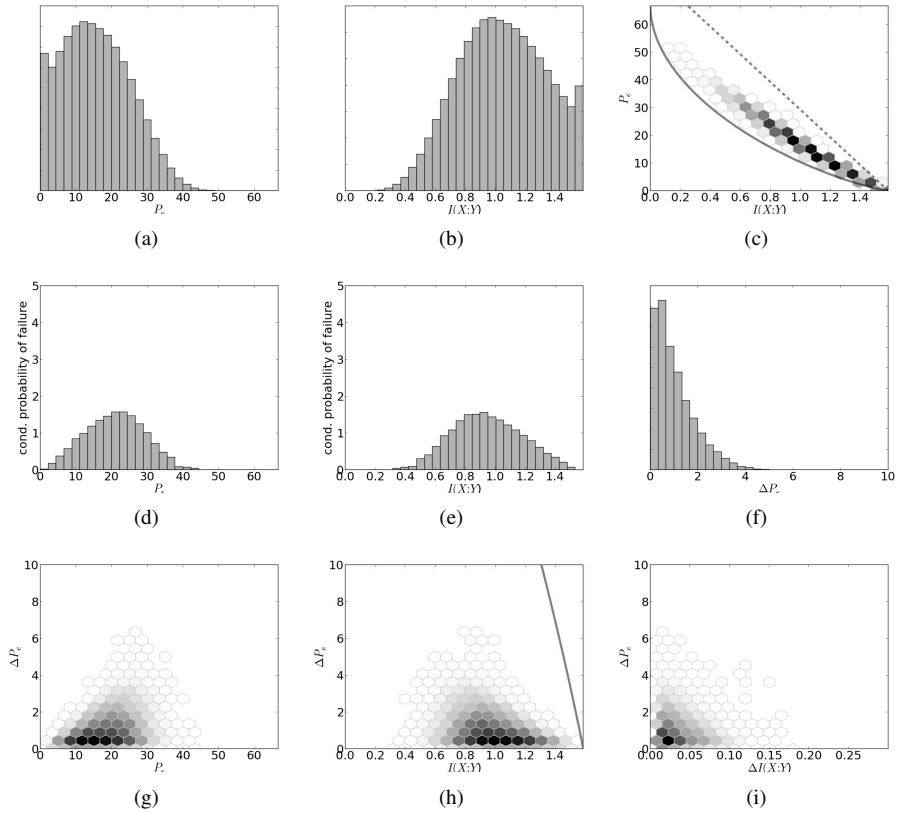


Figure 8.8: Results for an artificial three-class problems with one continuous feature. Class distributions are Gaussians centered at $x = -2$, $x = 0$ and $x = 2$. Mutual information values and misclassification probabilities are estimated using 10^4 samples from each class. Subfigures correspond to: an histogram of the misclassification probability (a), an histogram of the mutual information (b), a two-dimensional histogram of these two quantities (c), an estimate of the conditional probability of failure given P_e (d) and given the mutual information (e), an histogram of ΔP_e in case of a failure (f), an histogram of P_e and ΔP_e (g), an histogram of the mutual information and ΔP_e (h) and an histogram of the mutual information difference and ΔP_e (g).

of increasing sizes. At each step of the forward procedure, we also estimate the risk associated to each possible feature subset. We say that a forward step is a failure if the subset selected based on the mutual information is not the one leading to the smallest classification risk at this step.

Both the mutual information and the misclassification probabilities are thus es-

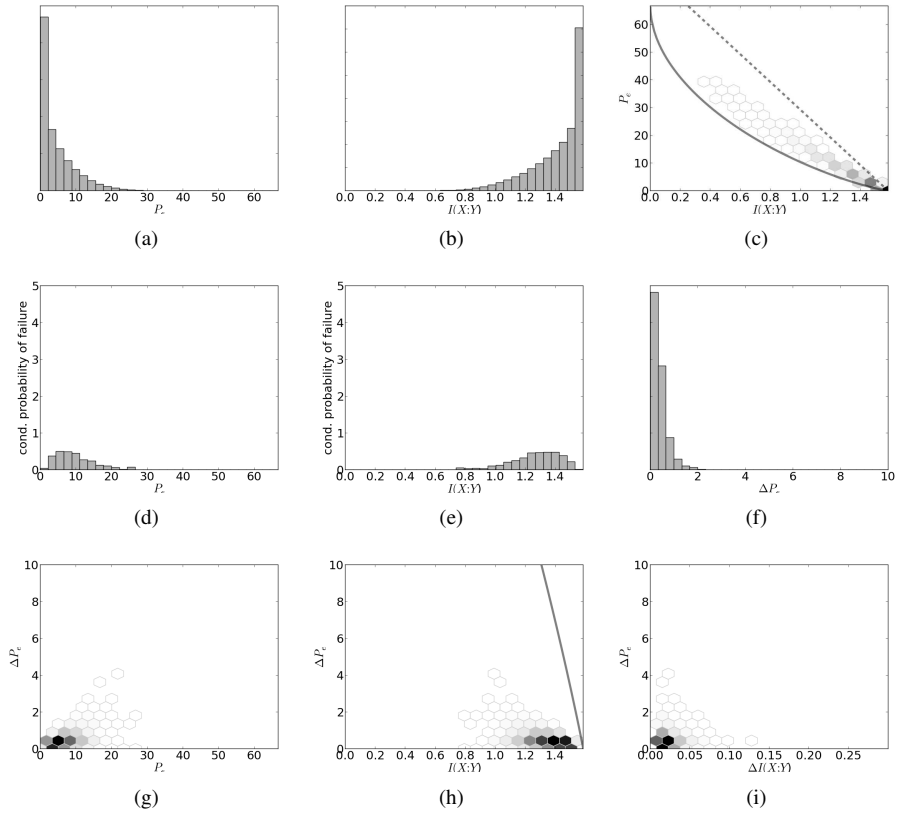
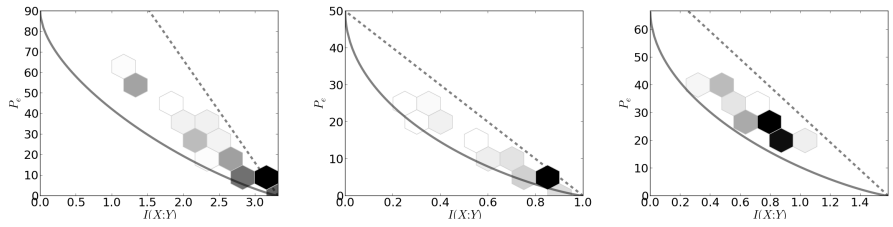


Figure 8.9: Results for an artificial three-class problems with one continuous feature. Class distributions are Gaussians centered at $x = -4$, $x = 0$ and $x = 4$. Mutual information values and misclassification probabilities are estimated using 10^4 samples from each class. Subfigures correspond to: an histogram of the misclassification probability (a), an histogram of the mutual information (b), a two-dimensional histogram of these two quantities (c), an estimate of the conditional probability of failure given P_e (d) and given the mutual information (e), an histogram of ΔP_e in case of a failure (f), an histogram of P_e and ΔP_e (g), an histogram of the mutual information and ΔP_e (h) and an histogram of the mutual information difference and ΔP_e (g).

timated for whole feature subsets, and thus for multivariate variables, contrarily to what has been done for artificial problems. More precisely, those quantities are directly estimated by first estimating the class-conditional probabilities $P(Y|X)$ of each sample. These class-conditional probabilities are obtained using the Bayes rule and marginalisation from the conditional probabilities $P(X|Y)$, which are estimated using



(a) Digits (b) Wall-following robot (c) Waveform

Figure 8.10: Mutual information and classification risk with random subsets of ten features for the Digits, Wall-following robot and Waveform datasets, with Fano and Hellman-Raviv bounds.

the Kozachenko-Leonenko density estimator (2.17). As we use estimates for the MI and the classifiifcation risk, it is important to check that they are reliable before conducting more in-depth experiments. In this regard, Figure 8.10 shows a 2D histogram of the mutual information and the misclassification probability for each dataset and random subsets of 10 features. It can be observed that the Fano and Hellman-Raviv bounds hold, what gives indications about the validity of the above procedure.

The results of the experiments are presented in Figures 8.11, 8.12 and 8.13, which show several interesting plots for each problem. From left to right, the first row presents a histogram of the classification risk P_e, a histogram of the mutual information as well as P_e for different feature subset sizes. The second row shows an estimate of the conditional probability of failure given (i) P_e and (ii) the mutual information. It also shows the mutual information for different feature subset sizes. The third row shows 2D histograms of P_e *vs.* the risk loss ΔP_e, the mutual information *vs.* ΔP_e and the mutual information difference *vs.* ΔP_e. The fourth row shows the mutual information difference and ΔP_e for different feature subset sizes, and the conditional probability of failure given the feature subset size. If a step is a mutual information failure, the risk loss is computed between the feature subset with the highest mutual information and the feature subset with the lowest misclassification probability.

Results

In Figures 8.11(a), 8.11(b), 8.11(c) and 8.11(f), it can be shown that for the Digits dataset, the forward procedure is first faced with quite difficult problems, which become eaiser as the number of selected features increases. The total percentage of failures is 7.8 and Figures 8.11(g), 8.11(h) and 8.11(i) indicate that 95% of the the risk losses are below 2%. The dark hexagonal bin in these figures further shows that the failures occur mostly when the compared feature subsets are close in terms of both

mutual information and misclassification probability. Besides, ΔP_e decreases for large mutual information values and small misclassification probabilities as can be seen in Figures 8.11(d) and 8.11(e). From Figures 8.11(j), 8.11(k) and 8.11(l), we can deduce that the probability of failure is maximum at the beginning of the forward procedure; it corresponds to small risk loss and decreases quickly as the number of considered features increases.

The results obtained on the Wall-following robot dataset are close to the ones observed for the Digits dataset, with two exceptions. First, the classification performances are already maximal with about only three features, as observed in Figures 8.12(c) and 8.12(f). Then, MI failures are much more likely to happen at the first step of the forward search (see Figure 8.12(l)). This step corresponds to intermediate values of mutual information and the classification risk. Therefore, it is shown in Figure 8.12(j) that ΔP_e is only significant for subsets of only one feature. This first step corresponds to the peak of probability of failure in Figures 8.12(d) and 8.12(e) as well as to the dark hexagonal bin in Figures 8.12(g), 8.12(h) and 8.12(i). Globally, failures occur in 2.4% of the cases and 95% of the misclassification probability losses remain below 1%.

The Waveform dataset is actually a harder problem, as shown in Figures 8.13(a), 8.13(b), 8.13(c) and 8.13(f). However, the percentage of failure is only 0.2. As shown in Figure 8.13(l), the conditional probability of failure increases for small feature subsets before quickly decreasing after three features have been selected. The misclassification probability is large for subsets of one and two features (see Figure 8.13(c)), which indicates again that failures mostly occur for intermediate mutual information values and misclassification probabilities. The peak of probability of failure in Figures 8.13(d) and 8.13(e) corresponds to the dark hexagonal bin in Figures 8.13(g), 8.13(h) and 8.13(i), where it can be seen that 95% of the misclassification probability losses remain below 0.4%.

Discussion for real-world problems

The main conlusion from the above results on real-world datasets is that that mutual information is more likely to fail in the first stages of the forward search, even if the probability of a failure remains limited. Those first steps actually correspond to intermediate values of mutual information. On the three datasets, the observed misclassification probability losses are low (around 1 percent). The mutual information failures thus seem not to have important consequences in practice, regarding the classification risk. Besides, failures mainly occur when the feature with the best mutual information and the feature with the best misclassification probability are close in terms of both mutual information and misclassification probability.

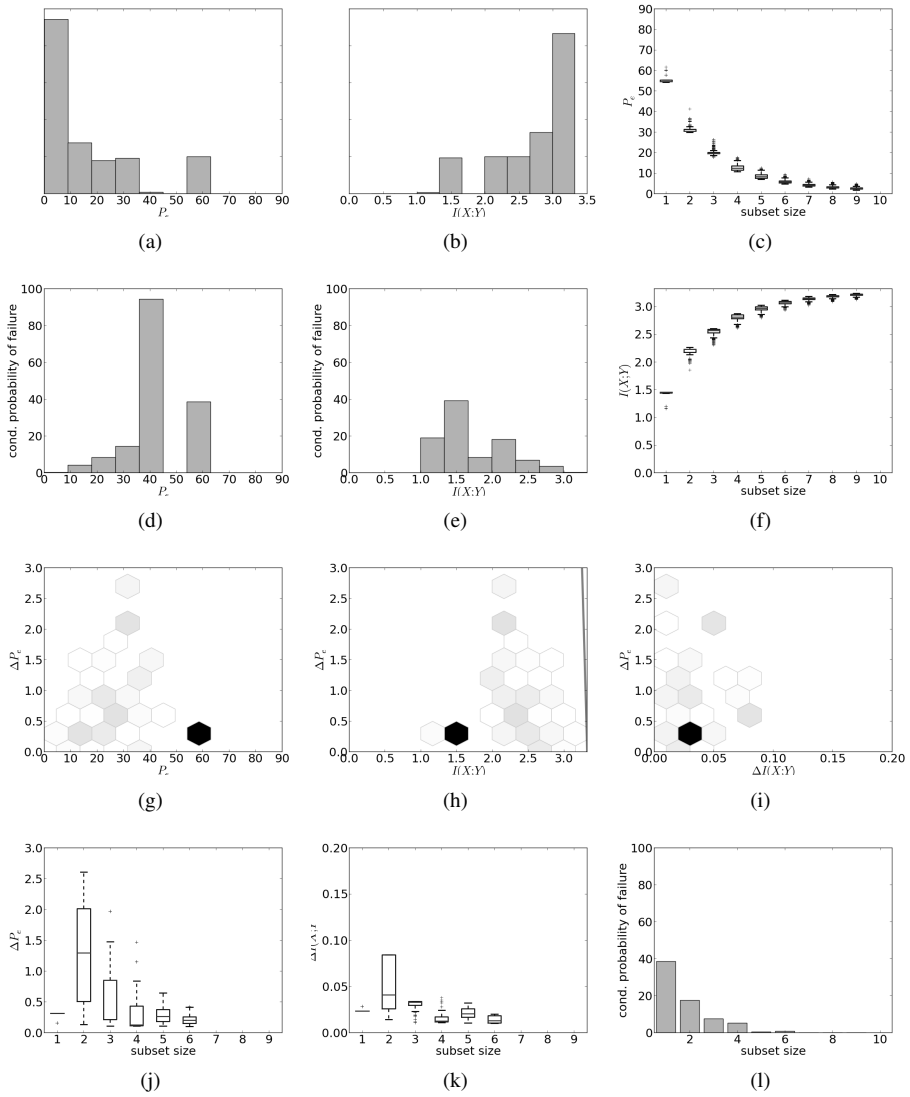


Figure 8.11: Results of mutual information-based forward search with random subsets of ten features for the Digits dataset: histogram of the classification risk P_e (a), histogram of the mutual information (b) as well as P_e (c) for different feature subset sizes, estimate of the conditional probability of failure given P_e (d) and the mutual information (e), mutual information for different feature subset sizes (f), histogram of P_e *vs.* the risk loss ΔP_e (g), histogram of the mutual information *vs.* ΔP_e (h), histogram of the mutual information difference *vs.* ΔP_e (i), mutual information difference (j) and ΔP_e (k) for different feature subset sizes, conditional probability of failure given the feature subset size (l).

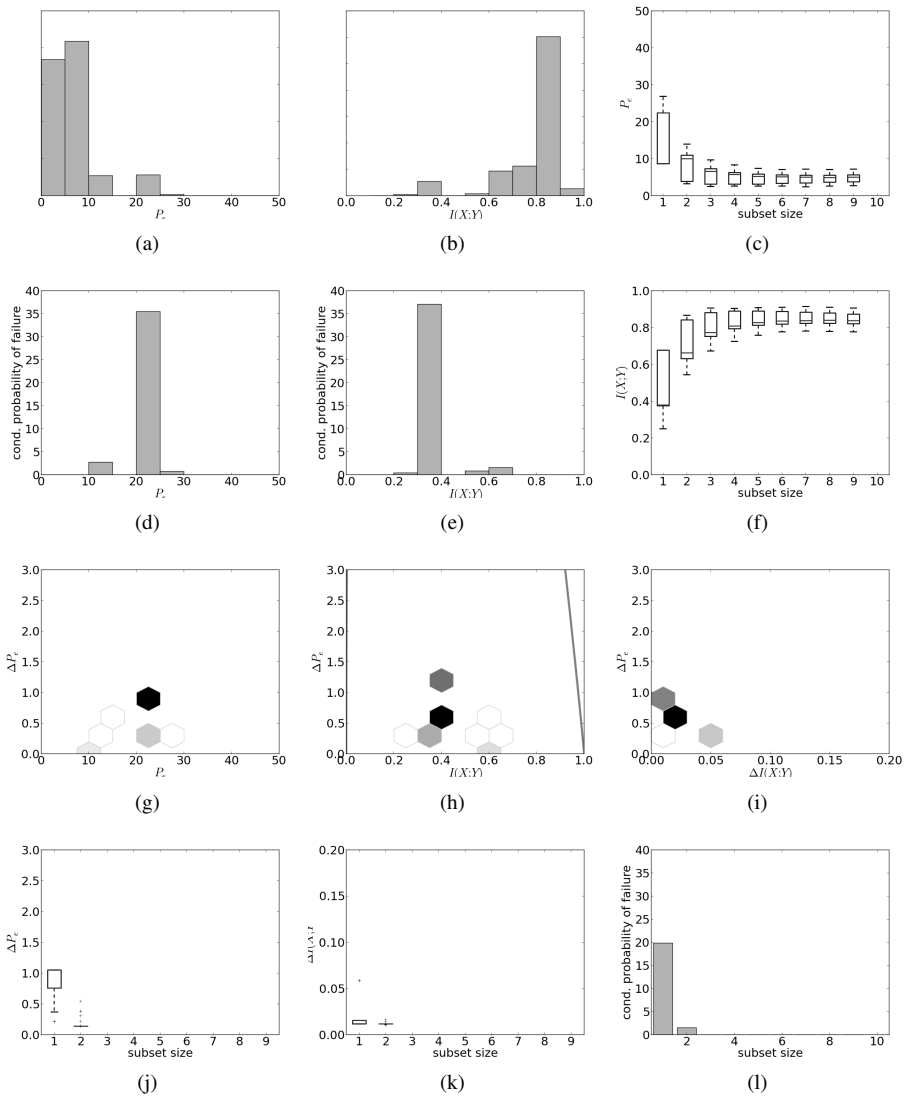


Figure 8.12: Results of mutual information-based forward search with random subsets of ten features for the Wall-following robot dataset: histogram of the classification risk P_e (a), histogram of the mutual information (b) as well as P_e (c) for different feature subset sizes, estimate of the conditional probability of failure given P_e (d) and the mutual information (e), mutual information for different feature subset sizes (f), histogram of P_e *vs.* the risk loss ΔP_e (g), histogram of the mutual information *vs.* ΔP_e (h), histogram of the mutual information difference *vs.* ΔP_e (i), mutual information difference (j) and ΔP_e (k) for different feature subset sizes, conditional probability of failure given the feature subset size (l).

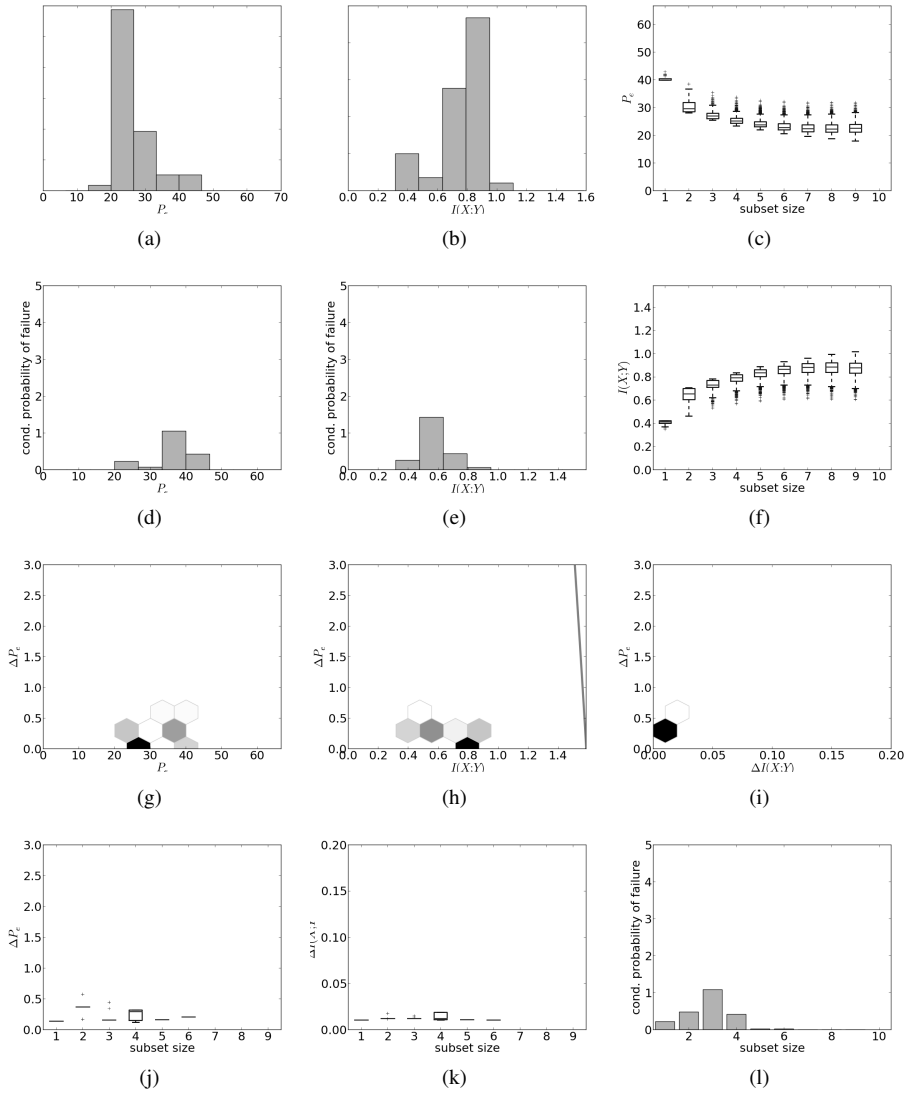


Figure 8.13: Results of mutual information-based forward search with random subsets of ten features for the Waveform dataset: histogram of the classification risk P_e (a), histogram of the mutual information (b) as well as P_e (c) for different feature subset sizes, estimate of the conditional probability of failure given P_e (d) and the mutual information (e), mutual information for different feature subset sizes (f), histogram of P_e *vs.* the risk loss ΔP_e (g), histogram of the mutual information *vs.* ΔP_e (h), histogram of the mutual information difference *vs.* ΔP_e (i), mutual information difference (j) and ΔP_e (k) for different feature subset sizes, conditional probability of failure given the feature subset size (l).

8.3 General analysis of the results

We now summarize the observations and conclusions drawn from the experimental results obtained on the three kinds of problems.

The first important point is that mutual information can actually fail as a feature selection criterion from the classification risk point of view. This however does not mean that this criterion has no interest for feature selection. Indeed, the proportion of failures is quite limited (often below 5%) and so is the misclassification probability loss, which often lies around a few percents. For instance, on the three real-world problems, the misclassification probability loss remains below 2% for 95% of the failures.

Then, the failures mostly occur for intermediate values of mutual information. In the context of a forward selection, this corresponds to the first steps of the procedure, when the number of selected features is still small. Therefore, practically, it could be interesting to perform some backward steps after the first few steps of the forward search, when the procedure lies in a region where mutual information is likely to be a more reliable criterion for feature selection. In the same spirit, another potential option is to start the forward search with all combinations of 2 or 3 features if this can be afforded. Eventually, a backward search algorithm starts with all features, in a region where mutual information is reliable. It only reaches the dangerous zone after good feature subsets have been found. It could therefore also prevent the drawbacks mentioned for the forward search.

The last important fact is that failures occur the most when comparing feature subsets which are close in terms of both mutual information and misclassification probability. Therefore, the misclassification probability losses remain low, and in any case below the theoretical bound given in Section 8.1.3

8.4 Comments on the stability

In this section, we briefly discuss the stability of greedy MI-based feature selection algorithms. Our objective is to evaluate how stable the procedure is and what are the potential consequences of its unstability. We also would like to quantify the relative influence of the unstability on the classification risk, compared to the use of mutual information (as studied from the beginning of this chapter). To this end, we consider the popular Kuncheva stability index. For two feature subsets A and B of n_f features chosen out of d possible ones, such that $|A \cap B| = r$, the Kuncheva index is defined as:

$$I_C(A,B) = \frac{rd - n_f^2}{n_f(d-n_f)}. \tag{8.3}$$

When K feature sequences are used, the stability index becomes:

$$\frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} I_C(S_i(k), S_j(k)), \tag{8.4}$$

where $S_i(k)$ denotes the k best ranked features of the sequence S_i.

Figures 8.14 and 8.15 show the Kuncheva index for a MI-based forward search procedure, respectively obtained on the Waveform and the Wall-following robot datasets. The results have been obtained through a 10-fold cross-test procedure. Both figures show that the stability remains quite high at the beginning of the procedure, before decreasing as the dimensionality of the feature subsets increases. This means that the same most relevant features are always selected first; then as could be expected, when less relevant features have to be selected, the choice is not obvious anymore and slight differences can be observed from run to run, as the dataset is slightly modififed. These results comply quite well with the intuition and indicate that the stability of the MI-based procedures does not seem to be an issue.

To confirm this intuition, we have also computed the standard deviation of the risk estimation obtained using the 10 feature sequences built during the cross-test procedure. For each number of selected features and both datasets, the standard deviation of the estimated risk is below 1 %. This is a very low value, especially regarding the average classification risk obtained on the datasets (see Figures 8.13(c) and 8.12(c)). The standard deviation of the risk using the different feature subsets is indeed much lower than the average risk obtained on the datasets. This means that the conclusions given in the previous sections of this chapter are probably not too much influenced by the stability of the method and that unstability has no major negative impact on the MI-based feature selection procedure. This conclusion should however be confirmed by further investigations as the experimental protocol of Section 8.2.3 consisted in randomly preselecting 10 features before conducting the feature selection procedure, in order to limit the time required to conduct the experiments.

8.5 Direct risk estimation

Results in the previous section indicate that the MI is not an ideal proxy for the classification risk, despite strong similarities. Therefore, if one is interested by the classification risk only, estimating it correctly and using such an estimate to select relevant features can be of great interest in practice. However, beacause of the lack of efficient and reliable risk estimators, such an approach is almost never considered. Instead, the mutual information is often used, mainly because of its strong connections with the risk presented in Section 8.1.

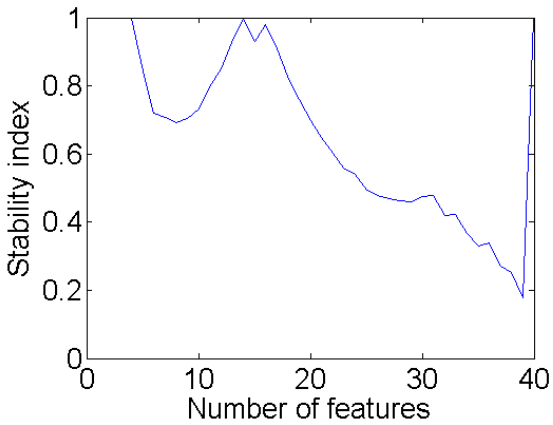


Figure 8.14: Kuncheva stability index on the Waveform dataset for a greedy forward search procedure.

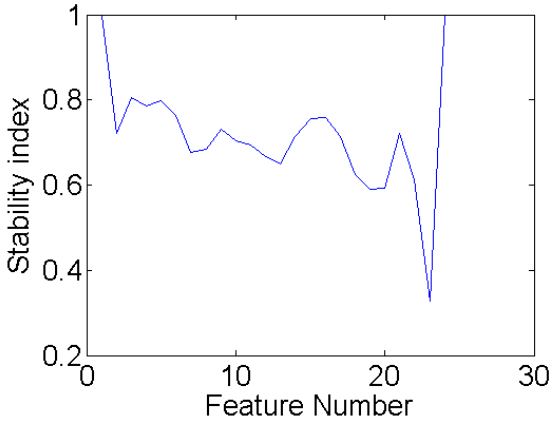


Figure 8.15: Kuncheva stability index on the Wall-following robot dataset for a greedy forward search procedure.

The idea of feature selection using risk estimation is not new. It actually dates back to the works [133, 76] which use a rectangular Parzen density estimation and require the features to be discretised; this obviously leads to a loss of information. Besides, every possible combination of discrete feature values has to be considered, which becomes not tractable for high-dimensional datasets. Discrete features are also required in [123] which is devoted to cancer classification problems. In [79], Parzen or k-NN density estimators are used but only binary problems are tackled. Other related works include [201] which is based on counting the number of mistakes made by a 1-NN classifier and [186] which gives connections between risk minimisation and the well-known Relief feature selection algorithm. Existing methods thus suffer from many drawbacks since none of them is simultaneously able to (i) deal with continuous variables, (ii) be reliable with high-dimensional variables, (iii) consider multi-class problems and (iv) be independant from a specific classification algorithm.

These observations motivate the development of a new risk estimator. To this end, we use the Kozachenko-Leonenko estimator (2.17) to address the aforementioned limitations. The developments are introduced in Section 8.5.1. We also compare the performances of the proposed estimator with the ones of the MI for feature selection purposes in Section 8.5.2.

8.5.1 Derivation of the risk estimator

In this section, we are mainly interested in estimating the Bayes risk, i.e. the optimal achievable risk (see Section 2.1.2). As can be deduced from Equations (2.7) and (2.9), the Bayes risk can be written as

$$R^* = \min_f R(f) = \mathop{\mathbb{E}}_X \left[1 - \max_{y \in \mathcal{Y}} p_{Y|X}(y|x) \right] \tag{8.5}$$

where f denotes a classifier. The Bayes risk thus involves the conditional probabilities of the (discrete) class labels given the observed samples.

However, as discussed in the first part of the thesis (see more precisely Section 2.1.3), the Kozachenko-Leonenko density estimator can be used with continuous variables only. Indeed, with discrete variables, the determination of the nearest neighbors would probably not be univoque, harming the estimation process. A possible solution is therefore to consider the density estimator in order to get estimates for $p_{X|Y}$ using Equation (2.17), before considering the Bayes rule to obtain the estimate

$$\hat{p}_{Y|X}(y|x) = \frac{\hat{p}_{X|Y}(x|y)\hat{p}_Y(y)}{\sum_{y \in \mathcal{Y}} \hat{p}_{X|Y}(x|y)\hat{p}_Y(y)}. \tag{8.6}$$

Naturally, p_Y can be estimated for each class as the proportion of samples belonging

to the class in the training set. With an estimation of $p_{Y|X}$, it is now possible to derive two risk estimators.

The first one simply consists in counting the misclassifications made by a Bayes classifier:

$$\widehat{R^*} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left[y_i \neq \arg\max_{y\in\mathcal{Y}} \hat{p}_{Y|X}(y|x)\right]. \tag{8.7}$$

This empirical estimator of the true risk actually corresponds, for a Bayesian classifier, to the quantity that is frequently used in the literature to estimate the classification risk on a test set.

The second estimator directly estimates the quantity in Eq. (8.5):

$$\widehat{R^*} = \frac{1}{n}\sum_{i=1}^{n} \left[1 - \max_{y\in\mathcal{Y}} \hat{p}_{Y|X}(y|x_i)\right]. \tag{8.8}$$

As can be deduced from Equations (8.7) and (8.8), the first one does consider the training labels while the latter rather uses the estimated class memberships.

Using the Kozachenko-Leonenko density estimator which is an actual density estimator contrarily to some k nearest neighbours estimators, it is expected that the introduced risk estimators propose a solution to the drawbacks of existing estimators. Besides, they are also expected to give good results in feature selection, as MI based on the same density estimators proved to be very successful at this task, even when dealing with high-dimensional data [150].

8.5.2 Experiments

In this section, we compare the two proposed risk estimators with the mutual information estimated using the same density estimator (2.21) in the particular context of feature selection. To this end, we have conducted a greedy backward search procedure using these criteria. The criterion of comparison between these methods is the balanced classification rate of a 1-nearest neighbor classifier. The results have been obtained through a 10-fold cross test procedure. Despite we use continuous datasets in the experiments, some features in these datasets have equal values for a large number of samples. Therefore, to avoid any problem in the determination of the nearest neighbors for risk or MI estimation, a random Gaussian noise with variance 10^{-3} and zero-mean has been added to the features of each training set before the feature selection process. Note that the noisy datasets are only used for feature selection and that classification is performed on the noise-free datasets.

Figure 8.16 shows the results on 6 datasets from the UCI repository [71], presented in Chapter 3. The results obtained with the three methods can be seen to be quite similar. While mutual information and error counting (8.7) perform better on the Ecoli

dataset (Fig. 8.16(a)), direct risk estimation (8.8) is slightly better on the Ionosphere (Fig. 8.16(b)) and Seeds (Fig. 8.16(e)) datasets. Performances are equivalent on the other datasets. To detect any significant difference in the results, a two sample test of means has been carried out, following [148]. It appears that MI performances are significantly better for the first three feature subsets for the Ecoli Dataset. There is no other significant difference, except for very small subsets of one or two features in some datasets. An interesting observation is that the simple error counting (8.7) seems sufficient in practice to obtain good results. It is less costly than the second risk estimator since it requires the estimation of only n conditional probabilities while $n \times |\mathcal{Y}|$ conditional probabilities have to be estimated in the second estimator.

8.6 Conclusion

This chapter is devoted to the derivation of connections between the risk classification and the mutual information, especially in the context of feature selection.

The first part studies the adequacy between the two quantities through various experimental procedures. The first conclusion is that the MI is not an ideal proxy for the classification risk, in the sense that a feature subset having a higher mutual information with the output than another one could at the same time incur a higher classification risk. Extensive experiments confirm these possible failures but confirms the interest of mutual information for feature selection as the failures (i) are not frequent in practice and (ii) lead to a very small risk increase compared to the risk-optimal feature subset. The consequences of using the MI instead of the risk thus remain very limited in practice.

The second part of the chapter is dedicated to the development of a new risk estimator, based on the Kozachenko-Leonenko density estimator. The two proposed solutions are able to handle continuous features, multi-class problems and are expected to be reliable as the dimension of the data increases. Experimentally, they seem to be very close to the MI as feature selection criteria. This confirms the previous observations on the similarity between risk and mutual information for feature selection. The proposed risk estimators could also be useful in other contexts, as for instance active learning, where it could serve as a criterion to lead the sample selection process.

It is important to note that Bayes risk estimation does not necessarily require the estimation of densities. Indeed, some classification algorithms are said to be consistent, meaning that they asymptotically converge to the Bayes risk, as the number of available data points grows to infinity. This is for instance the case of the SVM [170] or the KNN, while the error of a 1NN classifier asymptotically tends to twice the Bayes risk [35]. Estimating densities, a hard and computationaly demanding task in practice, is therefore not always required to estimate the Bayes risk. The estimators proposed

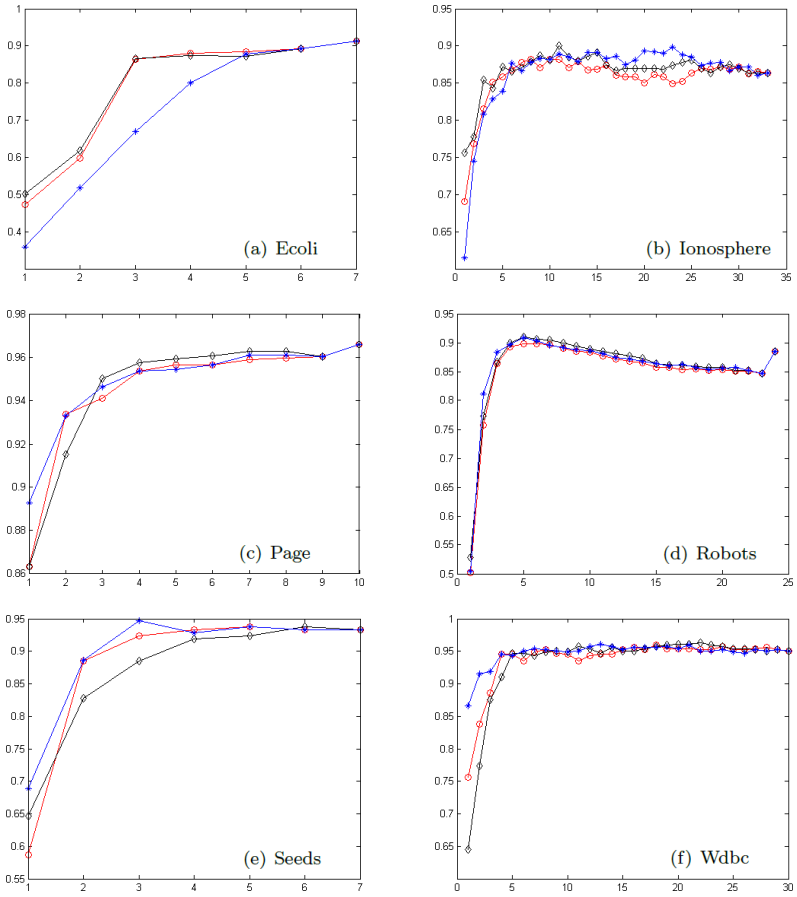


Figure 8.16: Balanced classification rate of a 1-nearest neighbour classifier as function of the number of selected features obtained with the mutual information (o), the misclassification count (8) (◇) and the direct risk estimation (9) (*).

in this chapter do estimate densities, but are based on the non-parametric KNN classifier and are therefore compationally equivalent. Despite their connection to a nearest neighbours statistics, the estimators are not equivalent to the sole use of this classification algorithm and are expected to prevent the user from explicitely building and training prediction models.

Chapter 9

Adequation of the mutual information for feature selection in regression problems

Contents

While the previous chapter was dedicated to the adequation of the mutual information for feature selection from a risk perspective, we now try to develop a similar reflexion in the important context of regression. Here, the objective to minimize is not the classification risk anymore but rather the mean squared error (MSE) or the mean asolute error (MAE); we therefore study links between these two error criteria and the mutual information. More precisely, we show that under reasonable assumptions on the distribution of the prediction errors, the mutual information is actually optimal from a MSE or a MAE point of view.

Some works somehow address similar considerations, as e.g. [85] where the MSE is related to the derivative of the mutual information with respect to the signal to noise ratio. In another related work, connections between MSE and mutual information are shown (see [100, 29]). However, the relationships are only true in the case of normal estimation errors. In the further developments, we address this limitation by generalizing the results for more realistic hypotheses on the estimation errors on the target. In addition, we also present situations where the mutual information fail for feature selection (in the sense mentioned in the previous chapter). This way, we aim to provide both theoretical arguments and experimental evidences for the opportunity of using mutual information for feature selection in regression problems. In the end, we hope helping the interested user to better apprehend the behaviour of the MI to more efficiently use it in a feature selection context.

In the remaining of this chapter, we first present the well-known MSE and MAE error criteria and we connect them to the estimation or prediction error (Section 9.1). In Section 9.2, we theoretically demonstrate the adequacy of the mutual information for three distinct distributions of the estimation error. In Section 9.3, the potential inadequacy of the mutual information is illustrated through an example, and we try to characterize the conditions leading to this criterion to fail. The results are eventually discussed in Section 9.4. This chapter is based on the work [74].

9.1 Error criteria and link to the mutual information

Before presenting the two considered error criteria, please recall that in the context of feature selection, maximising the MI $I(X;Y)$ or minimising the conditional entropy $H(Y|X)$ is actually equivalent. Indeed, as detailed in Section 2.1, $I(X;Y) = H(Y) - H(Y|X)$. During the feature selection process, $H(Y)$ remains constant as it does not depend on any choice of features. In the remaining of this chapter, the discussion will therefore be about $H(Y|X)$ for simplicity and concision reasons, while the exact same conclusions can be drawn about $I(X;Y)$.

Assume that in a regression problem, for a given subset of d features, the continuous target output $Y \in \mathfrak{R}$ depends probabilistically on $X \in \mathfrak{R}^d$. Also assume that the

prediction is made through a function f which provides an estimate $\hat{Y} = f(X)$ of Y given X. The prediction or estimation error is then

$$\varepsilon = f(X) - Y, \tag{9.1}$$

whose distribution is supposed to be zero-mean and obviously depends on the choice of the features. Based on ε, two very popular regression error criteria can be derived.

The first one is the mean squared error (MSE), defined as the variance of ε,

$$\mathrm{E}\left\{(f(X)-Y)^2\right\} = \mathrm{E}\left\{\varepsilon^2\right\},$$

where $\mathrm{E}\{\cdot\}$ denotes the expected value. The second criterion is the mean absolute error (MAE), which is defined as the expected absolute value of ε:

$$\mathrm{E}\{|f(X)-Y|\} = \mathrm{E}\{|\varepsilon|\}.$$

An interesting observation is that, for a given function f, it is possible to rewrite the conditonal entropy $H(Y|X)$ in terms of ε [5, 38]:

$$H(Y|X) = H(\varepsilon|X). \tag{9.2}$$

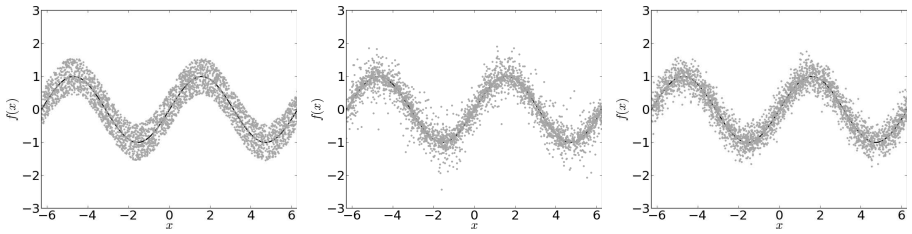
To show the validity of (9.2), we can rewrite $H(Y|X)$ as

$$\begin{aligned} H(Y|X) &= \int_{\mathcal{X}} p_X(x) H(Y|X=x)dx \\ &= \int_{\mathcal{X}} p_X(x) H(f(X)+\varepsilon|X=x)dx. \end{aligned} \tag{9.3}$$

Then, noting that $f(X)$ is fixed when X is known and that the differential entropy is translation invariant ($H(X+k) = H(X)$ for a constant k [63]), it comes that $H(f(X)+\varepsilon|X=x) = H(\varepsilon|X=x)$. Eq. (9.2) then follows directly from Eq. (9.3) since $\int_{\mathcal{X}} p_X(x)dx$ is obviously equal to 1. Based on Equation (9.2), we now study more precisely the connection between the conditional entropy or the mutual information and the two introduced error criteria.

9.2 Adequacy of the mutual information

We now show that it is possible to characterize some situations where mutual information is guaranteed to be an adequate criterion for feature selection in regression. More pecisely, we prove that when the conditional distribution of the estimation error is uniform, Laplacian or Gaussian, choosing the feature subset with the lowest condi-



(a) uniform noise $\mathcal{U}(\mu = 0, \Delta = 1.105)$ (b) Laplacian noise $\mathcal{L}(\mu = 0, \lambda = 0.203)$ (c) Gaussian noise $\mathcal{N}(\mu = 0, \sigma = 0.267)$

Figure 9.1: $\sin(x)$ polluted by uniform, Laplacian or Gaussian target noise with identical conditional target entropy $H(Y|X) = 0.1$.

tional target entropy $H(Y|X)$ or the highest mutual information is actually equivalent to minimising the MSE and the MAE criteria. This conclusion holds under the assumption that when two feature subsets are compared, the distributions of the estimation error for both subsets belong to the same parametric family (uniform, Laplacian or Gaussian). In practice, this hypothesis is realistic when the feature subsets which are compared are not too different from an informative contain perspective. This is for instance the case at a given step of a forward or backward search.

Assume that the estimation error is identically distributed for any $x \in \mathcal{X}$. In this case, the conditional entropy $H(Y|X)$ is obviously equal to the specific conditional entropy $H(Y|X = x)$ for any $x \in \mathcal{X}$. Because the mean of the estimation error distribution is zero, the distribution has only one effective parameter: the width Δ for the uniform estimation error, the scale λ for the Laplacian estimation error and the standard deviation σ for the Gaussian estimation error. It is clear that the value of the aforementioned parameters depends on the considered feature subset, which corresponds to X. Using the parameters, the conditional target entropies $H(Y|X)$ become $\ln\Delta$, $\ln[2e\lambda]$ and $\frac{1}{2}\ln\left[2\pi e\sigma^2\right]$, respectively [38].

The three noise distributions are illustrated in Figure 9.1, where the functional $f(x) = \sin(x)$ is polluted by the three types of noise, with identical conditional target entropy $H(Y|X) = 0.1$. As expected, under the uniform noise, the function values stay inside a tube around the real function f. Under Laplacian or Gaussian noise, the distribution of the function values spreads around the real value of the function f.

9.2.1 Theoretical considerations

Remember that the MSE can be interpreted as the variance of the estimation error. However, the estimation error is assumed to be identically distributed for any $x \in \mathcal{X}$.

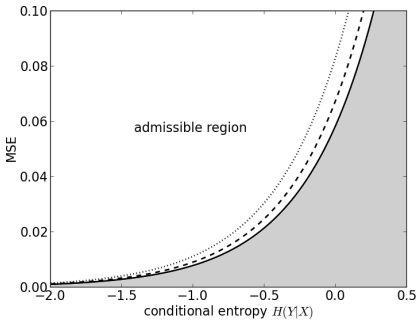


Figure 9.2: MSE as a function of the conditional target entropy $H(Y|X)$ for identically distributed uniform (dotted line), Laplacian (dashed line) and Gaussian (plain line) estimation errors.

Its variance is thus precisely equal to the MSE and can be written as $\frac{\Delta^2}{12}$, $2\lambda^2$ and σ^2 for uniform, Laplacian and Gaussian estimation errors, respectively [38, 112]. Using these relationships and the expressions for the condtional entropy derived in the previous section, it is possible to rewrite the MSE in sole terms of $H(Y|X)$. The MSE becomes $\frac{1}{12}\exp[2H(Y|X)]$, $\frac{1}{2e^2}\exp[2H(Y|X)]$ and $\frac{1}{2\pi e}\exp[2H(Y|X)]$ for uniform, Laplacian and Gaussian estimation errors, respectively.

The MSE expressed this way is a monotonically increasing function of $H(Y|X)$ and depends on the selected feature subset. Therefore, when comparing feature subsets which incur estimation errors belonging to one of the parametric families (uniform, Laplacian or Gaussian), choosing the feature subset which minimises the conditional target entropy $H(Y|X)$ or maximises the mutual information is completely equivalent to choosing a feature subset minimising the MSE. This is illustrated in Figure 9.2 where the MSE in terms of the conditional target entropy $H(Y|X)$ is shown for the three considered error distributions. As the Gaussian distribution is the one maximising the entropy for a given estimation error variance σ^2 [38], the corresponding curve is a lower bound for the MSE for which it defines an admissible region.

It is possible to conduct similar developments for the MAE criterion. Indeed, remember that the estimation error is assumed to be identically distributed. The MAE is thus by definition equal to the expected absolute value of the estimation error for any $x \in \mathcal{X}$. For uniform, Laplacian and Gaussian estimation errors, the expected absolute value can be written as $\frac{\Delta}{4}$, λ and $\sqrt{\frac{2}{\pi}}\sigma$, respectively [112]. Using these relationships and the expressions for the conditional target entropies derived previously, one can express the MAE in sole terms of $H(Y|X)$. The MAE becomes $\frac{1}{4}\exp[H(Y|X)]$, $\frac{1}{2e}\exp[H(Y|X)]$ and $\frac{1}{\pi\sqrt{e}}\exp[H(Y|X)]$ for uniform, Laplacian and Gaussian estimation errors, respectively.

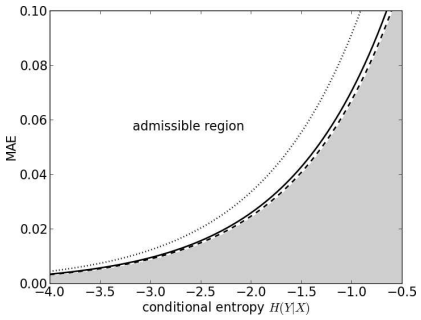


Figure 9.3: MAE as a function of the conditional target entropy $H(Y|X)$ for identically distributed uniform (dotted line), Laplacian (dashed line) or Gaussian (plain line) estimation error.

Similarly to what has been observed for the MSE, the MAE is a monotonically increasing function of the sole conditional target entropy, which in turn depends on the selected feature subset. Consequently, if some feature subsets all leading to estimation errors following one of (and the same) the considered distribution (uniform, Laplacian or Gaussian) are compared, choosing the feature subset minimising $H(Y|X)$ is equivalent to choosing the subset minimising the MAE. This conclusion is illustrated in Figure 9.3, which shows the MAE in terms of $H(Y|X)$ for the three families of estimation errors. The Laplacian distribution can be proven to be the maximum entropy distribution for a given expected estimation error absolute value λ [112]; the corresponding curve is a lower bound for the MAE and defines an admissible region for it.

9.3 Inadequacy of the mutual information

We now show that when the aforementioned hypotheses are violated, mutual information is not always adequate for feature selection in regression.

For instance, we exhibate an example where the conditional distribution of the estimation error is assumed to be a Student distribution for any $x \in \mathcal{X}$. In this case, choosing the feature subset which minimises the conditional target entropy $H(Y|X)$ is not necessarily equivalent to minimising either the MSE or the MAE criterion. The density of the non-standardised Student distribution is

$$\mathcal{S}(\varepsilon = e|\mu,\nu,\sigma) = \frac{1}{B\left(\frac{1}{2},\frac{\nu}{2}\right)\sqrt{\nu\sigma^2}}\left[1+\frac{(e-\mu)^2}{\nu\sigma^2}\right]^{-\frac{\nu+1}{2}} \tag{9.4}$$

where B is the beta function. Since the mean of the estimation error is zero, so is the parameter μ. The Student distribution has thus two remaining effective parameters: the number of degrees of freedom ν and the scale σ. Obviously, the value of the parameters ν and σ depend on the feature subset X. The conditional target entropy for a Student estimation error can be computed as

$$H(Y|X) = \left(\frac{\nu+1}{2}\right)\left[\Psi\left(\frac{\nu+1}{2}\right) - \Psi\left(\frac{\nu}{2}\right)\right] + \ln\left[\sqrt{\nu}B\left(\frac{1}{2},\frac{\nu}{2}\right)\right] + \ln\sigma \qquad (9.5)$$

where Ψ denotes the digamma function [38].

9.3.1 The MSE criterion

In the case of an identical Student estimation error for any $x \in \mathcal{X}$, the variance is equal to $\frac{\nu}{\nu-2}\sigma^2$. Using this last relationship and the expression of the conditional target entropy given above, the MSE can be rewritten as

$$\mathrm{E}\left\{(Y-f(X))^2\right\} = \frac{1}{(\nu-2)\,B\left(\frac{1}{2},\frac{\nu}{2}\right)^2} \times \exp\left\{2H(Y|X) - (\nu+1)\left[\Psi\left(\frac{\nu+1}{2}\right) - \Psi\left(\frac{\nu}{2}\right)\right]\right\}. \qquad (9.6)$$

The MSE thus cannot be written in sole terms of the conditional target entropy, contrarily to what was possible for the distributions considered in Section 9.2. Indeed, the MSE still depends on the number of degrees of freedom ν, while it could equivalently be made dependent on σ. It is however not possible to get rid of the dependence from the two parameters simultaneously. Figure 9.4 illustrates this problem by showing the MSE as a function of $H(Y|X)$ for different numbers of degrees of freedom. As the numbers of degrees of freedom of the estimation error distribution for different feature subsets can be different (for instance, a data point which is an outlier with respect to some features can look normal with respect to other features), the conditional target entropy can be decreased while increasing the MSE.

This potential failure is illustrated in Figure 9.5 where two feature subsets X_1 and X_2 characterised by a Student estimation error with parameters $\nu = 2.3$ and $\nu = 5$ are shown. Using mutual information (or conditional entropy), X_2 will be chosen. However, the MSE is larger for X_2 than for X_1 and mutual information fails as a feature selection criterion. In Figure 9.5, we also illustrate another subset X_3, characterised by the same degree of freedom that X_2 such that mutual information does not fail when choosing between X_1 and X_3. Failures are thus not immediate when the distributions considered Section 9.2 are not observed.

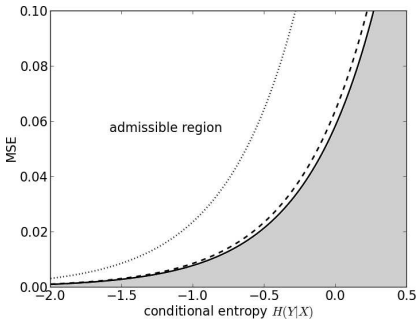


Figure 9.4: MSE in terms of the conditional target entropy $H(Y|X)$ for Student estimation error with different numbers of degrees of freedom: $\nu = 2.3$ (dotted line), $\nu = 5$ (dashed line) or $\nu = 30$ (plain line). The admissible region defined by the Gaussian curve appears in white.

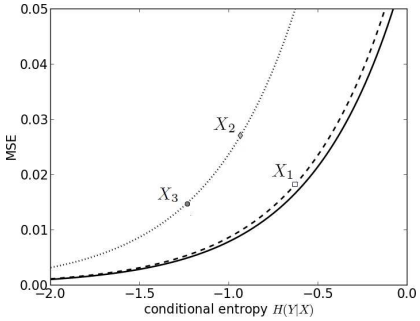


Figure 9.5: Example of mutual information failure for Student estimation error. The feature subsets correspond to different degrees of freedom of the distribution: $\nu = 2.3$ (dotted line) and $\nu = 5$ (dashed line).

In the case of a failure, the impact of using the mutual information is quite limited. Indeed, Figure 9.5 shows that the curve for $\nu = 30$ and $\nu = 5$ are very close. It is therefore not dangerous to compare feature subsets with such estimation error distributions. Since $\nu = 2.3$ leads to a quite extreme distribution, it is less encountered in practice. Mutual information can therefore be used in most practical cases with a reasonable confidence.

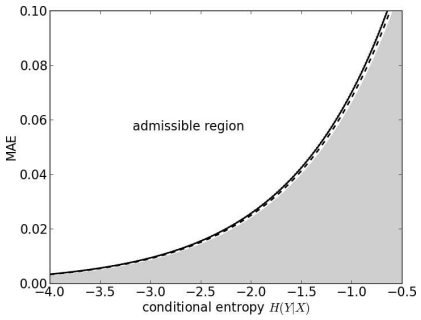


Figure 9.6: MAE in terms of the conditional target entropy $H(Y|X)$ for Student estimation error with different numbers of degrees of freedom: $\nu = 2.3$ (dotted line), $\nu = 5$ (dashed line) or $\nu = 30$ (plain line). The admissible region defined by the Laplacian curve appears in white.

9.3.2 The MAE Criterion

For the same Student distribution, the expected error absolute value can be expressed as [140]

$$\mathrm{E}\{|Y - f(X)|\} = \frac{2\sqrt{\nu\sigma^2}}{B\left(\frac{1}{2}, \frac{\nu}{2}\right)(\nu - 1)} \tag{9.7}$$

or equivalently as

$$\mathrm{E}\{|Y - f(X)|\} = \frac{2}{(\nu - 1)B\left(\frac{1}{2}, \frac{\nu}{2}\right)^2} \times \exp\left\{H(Y|X) - \left(\frac{\nu+1}{2}\right)\left[\Psi\left(\frac{\nu+1}{2}\right) - \Psi\left(\frac{\nu}{2}\right)\right]\right\}. \tag{9.8}$$

Similarly to the MSE, the MAE still depends on one of the two parameters of the distribution and cannot be written in sole terms of the conditional target entropy. Figure 9.6 illustrates this fact by showing the MAE in terms of $H(Y|X)$ for various numbers of degrees of freedom. Even if the different curves are very close, they are not equivalent and it is possible to decrease the conditional target entropy while increasing the MAE.

A concrete example of mutual information failure is shown in Figure 9.7. Two features subsets X_1 and X_2 are illustrated and are characterised by a Student estimation error with parameters $\nu = 2.3$ and $\nu = 5$, respectively. Using mutual information, X_2 will be chosen. However, selecting X_2 leads here to an increase in the MAE. Figure 9.7 also shows a counterexample: mutual information does not fail when choosing

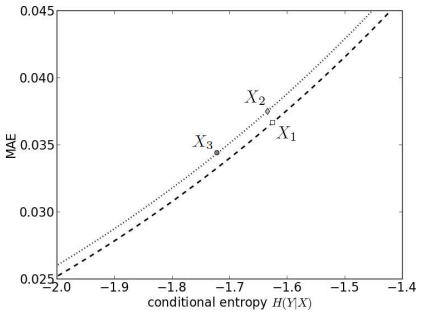


Figure 9.7: Example of mutual information failure for Student estimation error. The feature subsets correspond to different degrees of freedom: $\nu = 2.3$ (dotted line) and $\nu = 5$ (dashed line).

between the feature subsets X_1 and X_3.

9.4 Conclusion and discussion

This chapter first shows that mutual information is often a valuable criterion for feature selection in regression. Indeed, estimation errors with a uniform, Laplacian or Gaussian distribution are realisitc, and choosing a feature subset which minimises the mutual information corresponds in this case to minimising both the MSE and the MAE. More generally, one can postulate that mutual information can safely be used whenever the estimation error is identically distributed for any $x \in \mathcal{X}$ and that its distribution can be characterised by only one parameter.

Mutual information is however not always optimal. Indeed, when the estimation error distribution is characterised by multiple parameters (as it is the case for the Student distribution), it is possible to obtain different values of the MSE or the MAE for a given value of the mutual information. Therefore, minimising the mutual information does not necessarily correspond to minimising either the MSE or the MAE. The impact of this issue may be of various importance. For instance, the MSE can be quite different for a given conditional target entropy when the number of degrees of freedom of the Student distribution changes. On the contrary, the difference remains quite small for the MAE. However, as it is not common to observe very small degrees of freedom in practice (this could actually correspond to the presence of a large number of outliers), one can expect the impact of mutual information failures to remain small.

We eventually discuss an important hypothesis made in Sections 9.2 and 9.3. To derive the results, the estimation error is assumed to be identically distributed for any $x \in \mathcal{X}$. In practice, this is not always the case and the distribution of the estimation

error could depend on x. Consider for instance a situation where the estimation error follows a Gaussian distribution $\mathcal{N}(0,\sigma_1)$ for one half of the samples and $\mathcal{N}(0,\sigma_2)$ for the other half. The conditional entropies $H(Y|X)$ for this non-identically distributed (n.i.d.) estimation error is $\frac{1}{2}\ln[2\pi e\sigma_1\sigma_2]$, the MSE is $\frac{1}{2}\left(\sigma_1^2+\sigma_2^2\right)$ and the MAE is $\frac{1}{\sqrt{2\pi}}(\sigma_1+\sigma_2)$. The MSE and the MAE can be rewritten by replacing e.g. σ_2:

$$\frac{1}{2}\left(\sigma_1^2+\frac{\exp[4H(Y|X)]}{4\pi^2e^2\sigma_1^2}\right) \tag{9.9}$$

for the MSE and

$$\frac{1}{\sqrt{2\pi}}\left(\sigma_1+\frac{\exp[2H(Y|X)]}{2\pi e\sigma_1}\right) \tag{9.10}$$

for the MAE. Consequently, for a given value of $H(Y|X)$, it is possible to obtain feature subsets with different MSE and MAE values. The mutual information may therefore also fail in that case.

In practice it thus appears that the nature of the estimation error is important to assess the interest of MI for feature selection. Trying to characterize it can therefore be interesting if one intends to perform feature selection based on mutual information.

Conclusion

Feature selection is a task of great importance in many data mining and machine learning problems. Reducing the dimensionality of a dataset by detecting the most relevant variables is of great interest, both from a prediction accuracy and from a interpretation point of view. Indeed, it is often the case that the data at hand contains irrelevant or redundant features, or is simply too high-dimensional for a prediction model to reliably handle it. Learning with high-dimensional data is generally acknowledged as a hard task involving counter-intuitive phenomena.

While deeply studied in the literature, feature selection has mainly been addressed for classical data, which are not always representative of actual real-world situations. Indeed, as illustrated throughout the second part of this thesis, data are for instance often unprecise, missing or not fully supervised to name a few potential issues. It is therefore important in practice to propose feature selection methods for such actual problems. This is the first main objective of the thesis, whose second part presents and evaluates feature selection algorithms for a variety of poblems.

The proposed algorithms are mainly based on three feature selection criteria: the mutual information (MI), the Laplacian score (LS) and the Hilbert-Schmidt independence criterion (HSIC).

In the context of semi-supervised learning, we propose an algorithm for regression problems based on the graph Laplacian of a dataset. The graph is constructed based on both the available target outputs and the unsupervised data points. The proposed method is thus able to consider both sources of information and is experimentally shown to outperform many potential competitors. One of the main drawbacks of the approach based on the LS is that they rank the features individually rather than scoring subsets of features. To tackle this lack, we propose to combine the HSIC criterion with kernels defined on graph Laplacian. The introduced multivariate criterion leads to promising performances when it is used to adapt two well-knwon univariate feature selection criteria based on the LS.

The following problem we tackle is multi-label learning, where each sample can simultaneously belong to different classes. The proposed solution is to first reduce

the problem to a classical single-label problem, before using the MI criterion in a greedy forward procedure. The considered transformation of the problem allows us to make sure that the MI estimator will work properly. The transformation procedure we apply has a parameter controling the minimum number of samples a class contains after transformation. We propose an intuitive way to tune this parameter based on the permutation test. Results indicate that the proposed approach is better than existing ones. We then propose a multivariate strategy for discrete datasets. It is based on the HSIC criterion and shows interesting performances on benchmark datasets.

Another common problem in machine learning is the presence of missing data. To perform feature selection in this context, we adapt a nearest neighbors-based MI estimator by considering a distance measure directly able to handle missing values, the partial distance strategy. This way, we avoid any imputation phase before the feature selection process. Such a phase is however often required in practice since almost no prediction model is designed to deal with missing values. We thus compare two full prediction methodologies: performaing feature selection before or after the imputation phase. Results clearly indicate that the suggested approach has strong advantages over the other one. We also propose a more elaborate distance estimation procedure for samples with missing values. It is based on the assumption of a multivariate Gaussian distribution but can be shaown to produce accurate results even when dealing with non-Gaussian data.

The last specific problem for which we present results in this thesis is the presence of uncertainty in the provided labels. We first tackle the well-known label noise problem. We propose to adapt the number of nearest neighbors considered in the mutual information estimator we use by taking into account the estimated membership of each sample to the different classes. These memberships are first estimated by considering a noise model optimized through an expectation-maximisation algorithm. Results indicate the interest of considering a noise-tolerant version of the MI estimator compared to the use of the classical estimator in the presence of label noise. We also propose a feature selection criterion based on the Laplacian score for problems where the samples are given with a probability of belonging to each of the different classes.

The second main objective of the thesis is to study in more details aspects related to some information theoretic quantities. More precisely, we are first interested in the relationship between the mutual information, one of the most popular feature selection criterion, and common quality criteria for prediction models. These criteria are the mean squared arror and the mean absolute error for regression, and the (Bayes) classification risk for classification. Our goal is to better understand the situations where the mutual information is likely to be a reliable proxy for the risk and what are the consequences of using mutual information when it not the case.

For classification problems, we first show that the Fano and Hellman-Raviv bounds

on the risk as a function of the conditional entropy (and therefore the mutual information) do not guarantee the complete equivalence between the risk and the mutual information. The MI is thus not guaranteed to be an optimal feature selection criterion from a risk perspective. We nevertheless provide some conditions of optimality for the MI and an upper bound on the risk loss induced by a choice of features driven by the MI criterion. Extensive experiments indicate that the MI failures remain quite rare in practice and that their consequences are of limited impact, as they mainly occur between feature subsets close in terms of both MI and risk. Failures mostly occur for intermediate values of MI, corresponding to subsets of low cardinality. In this regard, backward search strategies could therefore reveal more reliable than forward ones, as they begin with a more important number of features, which correspond to a zone where the MI is less likely to fail.

For regression problems, we demonstrate that the adequacy of the MI regarding the MSE or the MAE can be studied from the distribution of the prediction errors perspective. Indeed, we show that if the prediction errors follow a uniform, Gaussian or Laplacian distribution, than the MI is optimal as a feature selection criterion. This is due to the fact that for these distributions, it is possible to rewrite the MSE or the MAE as a decreasing function of the sole MI. On the contrary, this is not possible for a Student distribution since it possesses too many parameters. The MI cannot be guaranteed to be optimal anymore. As for the classification, the consequences of the MI failures are likely to remain moderate in practice.

In this thesis, the different complex settings considered have been tackled individually. However, it is important to note that the PDS strategy used to handle missing data could as well be used in combination with the approach proposed for the label noise and the continuous mulit-label problems. It could also be used for Laplacian-based methods provided some minor modifications, like for instance the definition of a vector-matrix product with missing values. The other complex problems mentioned in the thesis are all about a specific form of supervision and we believe the proposed solutions could not be trivially combined to handle both kind of problems simultaneously.

Then, it is also necessary to briefly discuss the distinction between the wrapper and the filter approaches to feature selection. As already explained, the main difference between both approaches is that wrappers do consider the performances of the final prediction model during the feature selection step. For years, one of the main advantages of the filters over wrappers has been their ease and their lower computational requirements as filters do not require learning and tuning many prediction models. Today, because of the increase of both the available computational facilities and the complexity of the filter criteria, the distinction between filters and wrappers has become less obvious. For instance, in this thesis, we mainly focused on filters, with

a particular interest for the mutual information criterion estimated through a nearest neighbors statistics. This estimator is clearly related to a k nearest neighbors classifier and has a similar complexity. However, the classifier is only used as a tool to build a specific criterion, which can subsequently be used in combination with any other classification model and which possesses by itself interesting properties. As an example, the authors that have introduced the nearest neighbors-based MI estimator make the conjecture that their criterion is exactly zero for independent random variables; this is obviously a very desirable property and the MI estimator can prove useful even if no subsequent classification task is performed. Besides, the nearest neighbors algorithm is a simple and non-parametric classifier, easy to use in practice. Therefore, if one could argue that the estimated MI is not purely a filter criterion, we actually believe that this distinction is not of major importance and that what actually matters is to design appropriate and easy-to-handle criteria, whether they are based on a prediction model or not. Of course, since the MI estimator strongly relies on a distance-based criterion, distance-based prediction algorithms are expected to work particularly well with the features selected using this criterion. However, we do not see any major restriction for the use of other kinds of prediction models.

Since the thesis spreads in many different directions, lots of future work perspectives can be thought of. To cite a few, first, as already mentioned, the HSIC criterion seems well suited to transform univariate feature selection criteria into multivariate ones. The promising preliminary results presented in this thesis suggest that more extensive experiments should be conducted, using for instance other kernels than the ones considered. In the more specific context of problems with missing values, the proposed distance estimation procedures should also be tested with the MI estimator, as it has been shown to more accurately detect nearest neighbors than the partial distance strategy. More extensive experiments could also be carried out to assess the validity of the hypotheses made in the study of the adequation of the MI for regression problems. Of course, the most important future work perspectives concern the study of new complex settings. For instance, when dealing with missing data, it is frequent in practice that there exists a clear missingness structure, most values being missing for some given features. In multi-label learning, as already mentioned, a hierarchical structure is often assumed between the class labels. These two examples illustrate the amount of situations for which feature selection methods still have to be developped. We hope that this thesis will modestly help to contribute in these future developments.

Bibliography

[1] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2010.

[2] A. C. Acock. Working With Missing Values. *J.Marriage Fam.*, 67:1012–1028, 2005.

[3] R. J. Almeida, U. Kaymak, and J. M. C. Sousa. A new approach to dealing with missing values in data-driven fuzzy modeling. In *FUZZ*, pages 1–7, Barcelona, July 2010.

[4] T. W. Anderson. *An introduction to multivariate statistical analysis*. Wiley, 2nd edition, 1984.

[5] R. B. Ash. *Information theory*. Dover Publications, New York, 1990.

[6] Grung B. and Manne R. Missing values in principal component analysis. *Chemom. Intell. Lab.*, 42(1):125–139, 1998.

[7] G. Doquire B. Frenay and M. Verleysen. On the potential inadequacy of mutual information for feature selection. In *ESANN*, Brugges, April 2012.

[8] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[9] F. Bach and M. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48, 2003.

[10] F. Bach and M. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48, March 2003.

[11] F. R. Bach. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008.

[12] Ricardo Barandela and Eduardo Gasca. Decontamination of training samples for supervised pattern recognition methods. In *SSPR/SPR*, volume 1876 of *Lecture Notes in Computer Science*, pages 621–630, Alicante, August 2000.

[13] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

[14] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE T. Neural Networ.*, 5:537–550, 1994.

[15] R. E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

[16] N. Benoudjit and M. Verleysen. On the kernel widths in radial-basis function networks. *Neural Process. Lett.*, 18(2):139–154, 2003.

[17] J. C. Bezdez and S. K. Pal. *Fuzzy models for pattern recognition*. IEEE Press, Piscataway, 1992.

[18] G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Mach. Learn.*, 66(2-3):259–294, 2007.

[19] J. Bootkrajang and A. Kaban. Multi-class classification in the presence of labelling errors. In *ESANN*, Brugge, April 2011.

[20] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recogn.*, 37(9):1757–1771, 2004.

[21] C. Bouveyron and S. Girard. Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recogn.*, 42:2649–2658, November 2009.

[22] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman & Hall, 1984.

[23] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *J. Artif. Intell. Res.*, 11:131–167, 1999.

[24] G. Brown. An information theoretic perspective on multiple classifier systems. In *MCS*, pages 344–353, Reykjavik, June 2009.

[25] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *KDD*, pages 333–342, Washington, July 2010.

[26] R. Cai, Z. Zhang, and Z. Hao. Bassum: A bayesian semi-supervised method for classification feature selection. *Pattern Recogn.*, 44(4):811–820, 2011.

[27] O. Chapelle and S. Keerthi. Multi-Class Feature Selection with Support Vector Machines, 2008.

[28] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[29] B. Chen, J. Hu, H. Li, and Z. Sun. Adaptive filtering under maximum mutual information criterion. *Neurocomputing*, 71(16-18):3680 – 3684, 2008.

[30] W. Chen, J. Yan, B. Zhang, Z. Chen, and Q. Yang. Document transformation for multi-label feature selection in text categorization. In *ICDM*, pages 451–456, Omaha, October 2007.

[31] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[32] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In *PKDD*, pages 42–53, Freiburg, September 2001.

[33] E. Côme, L. Oukhellou, T. Denoeux, and P. Aknin. Mixture model estimation with soft labels. In *SMPS*, volume 48 of *Advances in Soft Computing*, pages 165–174, Toulouse, 2008. Springer.

[34] E. Côme, L. Oukhellou, T. Denoeux, and P. Aknin. Learning from partially supervised data using mixture models and belief functions. *Pattern Recogn.*, 42:334–348, March 2009.

[35] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, 1967.

[36] T. M. Cover. The best two independent measurements are not the two best. *IEEE Trans. Syst. Man, Cybern.*, SMC-4:116–117, 1974.

[37] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[38] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1st edition, 1991.

[39] G. A. Darbellay and I. Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE T. Inform. Theory*, 45(4):1315–1321, 1999.

[40] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML*, pages 74–81, Williamstown, June 2001.

[41] R. Daub, C.and Steuer, J. Selbig, and S. Kloska. Estimating mutual information using b-spline functions - an improved similarity measure for analysing gene expression data. *Bioinformatics*, 5(1), 2004.

[42] C. De Mol, S. Mosci, M.i Traskine, and A. Verri. A regularized method for selecting nested groups of relevant genes from microarray data. *J. Comput. Biol.*, 16(5):677–690, 2009.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. series B*, 39(1):1–38, 1977.

[44] T. Denoeux and L. M. Zouhal. Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Set. Syst.*, 122(3):47–62, 2001.

[45] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas. Protein classification with multiple algorithms. In *PCI*, volume 3746 of *Lecture Notes in Computer Science*, pages 448–456, Volos, November 2005. Springer.

[46] G. Doquire, G. de Lannoy, D. François, and M. Verleysen. Feature selection for inter-patient supervised heart beat classification. In *BIOSIGNALS*, pages 67–73, Roma, January 2011.

[47] G. Doquire, G. De Lannoy, D. François, and M. Verleysen. Feature selection for interpatient supervised heart beat classification. *Comput. Intell. Neuroscience*, 2011:1–9, 2011.

[48] G. Doquire and M. Verleysen. Mutual information-based feature selection for multilabel classification. *Accepted for publication in Neurocomputing*.

[49] G. Doquire and M. Verleysen. Feature selection for multi-label classification problems. In *IWANN*, volume 6691 of *Lecture Notes in Computer Science*, Torremolinos, June 2011. Springer.

[50] G. Doquire and M. Verleysen. Feature selection with mutual information for uncertain data. In *DaWaK*, volume 6862 of *Lecture Notes in Computer Science*, Toulouse, August 2011. Springer.

[51] G. Doquire and M. Verleysen. Graph laplacian for semi-supervised feature selection in regression problems. In *IWANN*, volume 6691 of *Lecture Notes in Computer Science*, Torremolinos, June 2011. Springer.

[52] G. Doquire and M. Verleysen. An hybrid approach to feature selection for mixed categorical and continuous data. In *KDIR*, Paris, October 2011.

[53] G. Doquire and M. Verleysen. Mutual information based feature selection for mixed data. In *ESANN*, Brugges, April 2011.

[54] G. Doquire and M. Verleysen. A comparison of multivariate mutual information estimators for feature selection. In *ICPRAM*, Vilamoura, February 2012.

[55] G. Doquire and M. Verleysen. Feature selection with missing data using mutual information estimators. *Neurocomputing*, 90:3–11, 2012.

[56] G. Doquire and M. Verleysen. Graph laplacian based approach to semi-supervised feature selection for regression problems. *Neurocomputing, in press*, 2012.

[57] G. Doquire and M. Verleysen. Handling imprecise labels in feature selection with graph laplacian. In *ICPRAM*, Vilamoura, February 2012.

[58] G. Doquire and M.l Verleysen. Mutual information for feature selection with missing data. In *ESANN*, Brugges, April 2011.

[59] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *J. Mach. Learn. Res.*, 5:845–889, 2004.

[60] B. Efron, T.r Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32:407–499, 2004.

[61] E. Eirola, G. Doquire, A. Lendasse, and M. Verleysen. Distance estimation in datasets with missing values. *Information Sciences*, 240:115–128, 2013.

[62] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687, Vancouver, December 2001.

[63] F. Emmert-Streib and M. Dehmer. *Information Theory and Statistical Learning*. Springer, 2008.

[64] A. Engel and C. P. Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, New York, NY, USA, 2001.

[65] R. Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.

[66] M. Figueiredo and A. Jain. Bayesian learning of sparse classifiers. In *CVPR 2001*, pages 35–41, Hawaii, December 2001.

[67] J. W. Fisher, M. Siracusa, and T. Kihn. Estimation of signal information content for classification. In *DSP/SPE*, Marco Island, January 2009.

[68] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[69] G. Forman, I. Guyon, and A. Elisseeff. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003.

[70] D. Francois, F. Rossi, V. Wertz, and M. Verleysen. Resampling methods for parameter-free and robust feature selection with mutual information. *Neurocomputing*, 70(7-9, Sp. Iss. SI):1276–1288, 2007.

[71] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[72] B. Frenay, G. Doquire, and M. Verleysen. Feature selection with imprecise labels: estimating mutual information in the presence of label noise. *Computational Statistics and Data Analysis, in press*.

[73] B. Frenay, G. Doquire, and M. Verleysen. Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification. *Neurocomputing*, 112:64–78, 2012.

[74] B. Frenay, G.r Doquire, and M. Verleysen. Is mutual information adequate for feature selection in regression ? *Accepted for publication in Neural Networks*.

[75] J. H. Friedman. Multivariate adaptive regression splines. *Ann. Stat.*, 19(1):1–67, 1991.

[76] K. S. Fu, P. J. Min, and T. J. Li. Feature selection in pattern recognition. *IEEE T. Syst. Sci. and Cyb.*, 6:33–38, 1970.

[77] K. Fukumizu, F. Bach, and M. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *J. Mach. Learn. Res.*, 5:73–99, December 2004.

[78] K.i Fukumizu, F. R. Bach, and A. Gretton. Statistical consistency of kernel canonical correlation analysis. *J. Mach. Learn. Res.*, 8:361–383, 2007.

[79] K. Fukunaga and D. M. Hummels. Bayes error estimation using parzen and k-nn procedures. *IEEE T. Patter Anal.*, 9(5):634 –643, 1987.

[80] B. Frenay G. Doquire and M. Verleysen. Risk estimation and feature selection. In *ESANN*, Brugges, April 2013.

[81] K. Ruben Gabriel and S. Zamir. Lower Rank Approximation of Matrices by Least Squares with Any Choice of Weights. *Technometrics*, 21(4):489–498, 1979.

[82] V. Gomez-Verdejo, M. Verleysen, and J. Fleury. Information-theoretic feature selection for functional data classification. *Neurocomputing*, 72(16-18, Sp. Iss. SI):3580–3589, 2009.

[83] A. Gretton, O. Bousquet, A. Smola, and B. SchÃ¶lkopf. Measuring statistical dependence with hilbert-schmidt norms. In *ALT*, pages 63–77, Singapore, October 2005.

[84] A. Gretton, A. J. Smola, O. Bousquet, R. Herbrich, A. Belitski, M. Augath, Y. Murayama, J. Pauls, B. Schölkopf, and N. K. Logothetis. Kernel constrained covariance for dependence measurement. In *AISTATS*, pages 112–119, Barbados, January 2005.

[85] D. Guo, S. Shamai, and S. Verdú. Mutual information and minimum mean-square error in gaussian channels. *IEEE T. Inform. Theory*, 51(4):1261–1282, 2005.

[86] . Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[87] I. Guyon. Practical Feature Selection: from Correlation to Causality. In *Mining Massive Data Sets for Security*. IOS Press, 2008.

[88] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., 2006.

[89] I. Guyon, N. Matic, and V. Vapnik. Advances in knowledge discovery and data mining. chapter Discovering informative patterns and data cleaning, pages 181–203. American Association for Artificial Intelligence, 1996.

[90] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46(1-3):389–422, 2002.

[91] M. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.

[92] R. J. Hathaway and J. C. Bezdek. Fuzzy c-means clustering of incomplete data. *IEEE T. Syst. Man Cybern. part B*, 31(5):735–44, 2001.

[93] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, Vancouver, December 2005.

[94] Z. He and W. Yu. Review article: Stable feature selection for biomarker discovery. *Comput. Biol. Chem.*, 34(4):215–225, 2010.

[95] M. Hein and O. Bousquet. Kernels, associated structures and generalizations, 2004.

[96] T. Helleputte and P. Dupont. Feature selection by transfer learning with linear regularized models. In *ECML PKDD*, pages 533–547, Bled, September 2009. Springer-Verlag.

[97] M. E. Hellman and J. Raviv. Probability of error, equivocation and the chernoff bound. *IEEE T. Inform. Theory*, 16:368–372, 1970.

[98] M. Hilbert and P. Lopez. The world technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, 2011.

[99] J. Van Hulse and T.i M. Khoshgoftaar. Incomplete-case nearest neighbor imputation in software measurement data. In *IRI*, pages 630–637, Las Vegas, August 2007.

[100] S. Ihara. *Information Theory for Continuous System*. World Scientific, 1993.

[101] I.i Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intell. Med.*, 31(2):91–103, 2004.

[102] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *J. Mach. Learn. Res.*, 12:2777–2824, November 2011.

[103] I. Jenhani, N. B. Amor, and Z. Elouedi. Decision trees as possibilistic classifiers. *Int. J. Approx. Reasoning*, 48:784–807, 2008.

[104] H. Junninen, H.i Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen. Methods for imputation of missing values in air quality data sets. *Atmos. Environ.*, 38(18):2895–2907, 2004.

[105] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms. In *ICDM*, pages 218–225, Houston, November 2005.

[106] S. Khan, S. Bandyopadhyay, A. R. Ganguly, S. Saigal, D. J. Erickson, V. Protopopescu, and G. Ostrouchov. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Phys. Rev. E*, 76:026209, 2007.

[107] G. King, J Honaker, A. Joseph, and K. Scheve. Analyzing incomplete political science data: An alternative algorithm for multiple imputation. *Am. Polit. Sci. Rev.*, 95:49–69, 2000.

[108] K.i Kira and L. A. Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *AAAI*, pages 129–134, San Jose, July 1992.

[109] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1):273–324, 1997.

[110] R. Kohavi, P. Langley, and Y. Yun. The utility of feature weighting in nearest-neighbor algorithms. In *ECML*, pages 85–92, Prague, April 1997.

[111] D. Koller and M. Sahami. Toward optimal feature selection. In Lorenza Saitta, editor, *ICML*, pages 284–292, Bari, July 1996.

[112] S. Kotz, T.J. Kozubowski, and K. Podgórski. *The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance*. 2001.

[113] L. F. Kozachenko and N. Leonenko. Sample estimate of the entropy of a random vector. *Problems Inform. Transmission*, 23:95–101, 1987.

[114] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Phys. Rev. E*, 69(6):066138, 2004.

[115] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE T. Pattern Anal.*, 27(6):957–968, June 2005.

[116] L. I. Kuncheva. A stability index for feature selection. In *IASTED - AIAP*, pages 390–395, February 2007.

[117] P. Křížek, J. Kittler, and V. Hlaváč. Improving stability of feature selection methods. In *CAIP*, pages 929–936, Vienna, August 2007.

[118] G. Lastra, O. Luaces, J. R. Quevedo, and A. Bahamonde. Graphical feature selection for multilabel classification tasks. In *IDA*, pages 246–257, Porto, October 2011.

[119] N. D. Lawrence and B. Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. In *ICML*, pages 306–313, Williamstown, June 2001.

[120] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *NIPS*, pages 598–605, Denver, November 1989.

[121] J. Lee and D.-W. Kim. Feature selection for multi-label classification using multivariate mutual information. *Pattern Recogn. Lett.*, 34(3):349–357, 2013.

[122] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer, 2007.

[123] J. Li, J.-M. Wei, T. Yu, and H.-W. Zhang. Feature selection based on bayes minimum error probability. In *FSKD*, pages 706 –710, Chongqing, May 2012.

[124] S. Li, R. M. Mnatsakanov, and M. E. Andrew. k-nearest neighbor based consistent entropy estimation for hyperspherical distributions. *Entropy*, 13(3):650–667, 2011.

[125] Y. Li, L. F. Wessels, D. de Ridder, and M. J. Reinders. Classification in the presence of class noise using a probabilistic kernel fisher method. *Pattern Recogn.*, 40:3349–3357, 2007.

[126] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data, Second Edition*. Wiley-Interscience, 2 edition, 2002.

[127] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *ICTAI*, pages 388–391, Washington, November 1995.

[128] Jun Liu, Jianhui Chen, and Jieping Ye. Large-scale sparse logistic regression. In *KDD*, pages 547–556, Paris, June 2009.

[129] S. Ma, . Song, and J. Huang. Supervised group lasso with applications to microarray data analysis. *Bioinformatics*, 8(1), 2007.

[130] N. S. Madsen, C. Thomsen, and J. M. Pena. Unsupervised Feature Subset Selection. In *Workshop on Probabilistic Graphical Models for Classification (PKDD'03)*, pages 71–82, 2003.

[131] P. Meesad and K. Hengpraprohm. Combination of knn-based feature selection and knnbased missing-value imputation of microarray data. In *ICICICl*, pages 341–344, Dalian, December 2008.

[132] X. L. Meng and D. B. Rubin. Maximum Likelihood Estimation via the ECM Algorithm: A General Framework. *Biometrika*, 80(2):267–278, 1993.

[133] P.J. Min. A non-parametric method for feature selection. In *Seventh Symposium on Adaptive Processes*, volume 7, page 34, Los Angeles, July 1968.

[134] P. Mitra, C. A. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE T. Pattern Anal.*, 24:301–312, 2002.

[135] L. C. Molina, L. Belanche, and À. Nebot. Feature selection algorithms: A survey and experimental evaluation. In *ICDM*, Maebashi City, December 2002.

[136] U. Ozertem, D. Erdogmus, and R. Jenssen. Spectral feature projections that maximize Shannon mutual information with class labels. *Pattern Recogn.*, 39:1241–1252, July 2006.

[137] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33(3):1065–1076, 1962.

[138] K. Pelckmans, J. De Brabanter, J. A. K. Suykens, and B. De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684–692, 2005.

[139] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE T. Pattern Anal.*, 27:1226–1238, 2005.

[140] S. Psarakis and J. Panaretos. The folded t distribution. *Communications in Statistics: A Theory and Methods*, 19(7):2717–2734, 1990.

[141] J. R. Quinlan. Unknown attribute values in induction. In *Proceedings of the Sixth International Machine Learning Workshop*, pages 164–168, 1989.

[142] I. Quinzan, J. M. Sotoca, and F. Pla. Clustering-based feature selection in semi-supervised problems. In *ISDA*, pages 535–540, Pisa, 2009.

[143] A. Rakotomamonjy. Variable selection using svm based criteria. *J. Mach. Learn. Res.*, 3:1357–1370, 2003.

[144] J. Read. A Pruned Problem Transformation Method for Multi-label classification. In *NZCSRS*, pages 143–150, 2008.

[145] M. Reed and B. Simon. *Functional Analysis*. Academic Press, 1 edition, 1981.

[146] Jiangtao Ren, Zhengyuan Qiu, Wei Fan, Hong Cheng, and Philip S. Yu. Forward semi-supervised feature selection. In *PAKDD*, pages 970–976, Osaka, June 2008.

[147] J. Reunanen. Overfitting in making comparisons between variable selection methods. *J. Mach. Learn. Res.*, 3:1371–1382, 2003.

[148] R.H. Riffenburgh. *Statistics in Medicine*. Academic Press, 2012.

[149] F. Rossi, N. Delannay, B. Conan-Guez, and M. Verleysen. Representation of functional data in neural networks. *Neurocomputing*, 64:183 – 210, 2005.

[150] F. Rossi, A. Lendasse, D. François, V. Wertz, and M. Verleysen. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometr. Intell. Lab*, 80(2):215–226, 2006.

[151] D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons., New York, 1987.

[152] Donald B. Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.

[153] Y. Saeys, T. Abeel, and Y. Peer. Robust feature selection using ensemble feature selection techniques. In *ECML PKDD*, pages 313–325, September 2008.

[154] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[155] J. L. Schafer and J. W. Graham. Missing data: Our view of the state of the art. *Psychol. Methods*, 7(2):147–177, 2002.

[156] R. E. Schapire and Y. Singer. Boostexter: A boosting-based systemfor text categorization. *Mach. Learn.*, 39(2-3):135–168, 2000.

[157] T. Schneider. Analysis of Incomplete Climate Data: Estimation of Mean Values and Covariance Matrices and Imputation of Missing Values. *J. Climate*, 14(5), 2001.

[158] M. Sebban and R. Nock. A hybrid filter/wrapper approach of feature selection using information theory. 35:835–846, 2002.

[159] D. Semani, C. Frélicot, and P. Courtellemont. Combinaison d'étiquettes floues/possibilistes pour la sélection de variables. In *RFIA*, pages 479–488, Toulouse, January 2004.

[160] J. Sexton and A. R. Swensen. Ecm algorithms that converge at the rate of em. *Biometrika*, 87(3):651–662, 2000.

[161] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 623–656, 1948.

[162] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE T. Pattern Anal.*, 22(8):888–905, 2000.

[163] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1986.

[164] P. Smets, Y. Hsia, A. Saffiotti, R. Kennes, H. Xu, and E. Umkehren. The transferable belief model. pages 91–96. 1991.

[165] A. J. Smola and R. I. Kondor. Kernels and regularization on graphs. In *COLT*, volume 2777 of *Lecture Notes in Computer Science*, pages 144–158, Washington, August 2003. Springer.

[166] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *ICML*, pages 823–830, Corvallis, June 2007.

[167] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse. Methodology for long-term prediction of time series. *Neurocomputing*, 70(16-18):2861–2869, 2007.

[168] A. Sorjamaa, A. Lendasse, and E. Séverin. Combination of SOMs for fast missing value imputation. In *MASHS*, Lille, June 2010.

[169] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.*, 2:67–93, 2001.

[170] I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Trans. Inf. Theor.*, 51(1):128–142, 2005.

[171] M.i Tan, L. Wang, and I. W. Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *ICML*, pages 1047–1054, Haifa, June 2010.

[172] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc., Series B*, 58:267–288, 1994.

[173] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multilabel Classification of Music into Emotions. In *ISMIR*, Philadelphia, September.

[174] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[175] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Mach. Learn.*, 73(2):185–214, 2008.

[176] M. Verleysen. Learning high-dimensional data. *Limitations and Future Trends in Neural Computation*, 186:141–162, 2003.

[177] .i Wagstaff. Clustering with Missing Values: No Imputation Required. In *IFCS*, pages 649–658, Chicago, July 2004.

[178] J. Walters-Williams and Y. Li. Estimation of mutual information: A survey. In *RSKT*, pages 389–396, Gold Coast, June 2009.

[179] Q. Wang, S. R. Kulkarni, and S. Verdu. Divergence Estimation for Multidimensional Densities Via k-Nearest Neighbor Distances. *IEEE T. Inform. Theory*, 55(5):2392–2405, April 2009.

[180] IBM Big Data webpage: http://www 01.ibm.com/software/data/bigdata/, February 2013.

[181] K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[182] K.Q. Weinberger, F. Sha, and L.K. Saul. Convex optimizations for distance metric learning and pattern classification. *IEEE Signal Proc. Mag.*, 27:146–158, 2010.

[183] D. Williams, X. Liao, Y. Xue, and L. Carin. Incomplete-data classification using logistic regression. In *ICML*, pages 972–979, Bonn, August 2005.

[184] Z. Xu, I. King, M. R. Lyu, and R. Jin. Discriminative semi-supervised feature selection via manifold regularization. *IEEE T. Neural Networ.*, 21(7), 2010.

[185] J. Yang and V. G. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intell. Syst.*, 13(2):44–49, 1998.

[186] S. H. Yang and B.-G. Hu. Discriminative feature selection by nonparametric bayes error minimization. *IEEE T. Knowl. Dat. En.*, 24(8):1422 –1434, 2012.

[187] L. Yu and H. Liu. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In *ICML*, pages 856–863, August 2003.

[188] Ming Yuan, Ming Yuan, Yi Lin, and Yi Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc., Series B*, 68:49–67, 2006.

[189] M. Zaffalon and M. Hutter. Robust feature selection by mutual information distributions. In *UAI*, pages 577–584, Edmonton, August 2002.

[190] D. Zhang, S. Chen, and Z.-H. Zhou. Constraint score: A new filter method for feature selection with pairwise constraints. *Pattern Recogn.*, 41:1440–1451, 2008.

[191] H. Zhang and G. Sun. Feature selection using tabu search method. 35:701–711, 2002.

[192] M. Zhang and Z. Zhou. A review on multi-label learning algorithms, 2013.

[193] M.-L. Zhang, J. M. Peña, and V. Robles. Feature selection for multi-label naive bayes classification. *Inform. Sciences*, 179(19):3218–3229, 2009.

[194] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recogn.*, 40(7):2038–2048, 2007.

[195] Y. Zhang and Z.-H. Zhou. Multilabel dimensionality reduction via dependence maximization. *ACM Trans. Knowl. Discov. Data*, 4(3):1–21, 2010.

[196] J. Zhao, K. Lu, and X. He. Locality sensitive semi-supervised feature selection. *Neurocomputing*, 71(10-12):1842–1849, 2008.

[197] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Ann. Stat.*, pages 3468–3497, 2009.

[198] P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006.

[199] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, pages 1151–1157, Corvallis, June 2007.

[200] H. Zhong, S. Xie, W. Fan, J. Ren, J. Peng, and K. Zhang. Graph-based iterative hybrid feature selection. In *ICDM*, pages 1133–1138, Pisa, December 2008.

[201] P.-F. Zhu, T.-H. Meng, Y.-L. Zhao, R.i-X. Ma, and Q.-H. Hu. Feature selection via minimizing nearest neighbor classification error. In *ICMLC*, pages 506 – 511, Qingdao, July 2010.

[202] X. Zhu and A. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

[203] Z. Zhu, Y. S. Ong, and M. Dash. Wrapper–filter feature selection algorithm using a memetic framework. *IEEE T. Syst. Man Cyb. Part B*, 37(1):70–76, 2007.

[204] H. Zou. The adaptive lasso and its oracle properties. *J. Am. Stat.l Assoc.*, 101(476):1418–1429, 2006.

[205] H.i Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc., Series B*, 67:301–320, 2005.