Université catholique de Louvain
École Polytechnique de Louvain
ICTEAM - Electrical Engineering
Machine Learning Group

Uncertainty and Label Noise in Machine Learning

Benoît Frénay

thèse présentée en vue de l'obtention du grade de docteur en sciences de l'ingénieur

Composition du Jury:

Prof. **Christophe De Vleeschouwer**, Président, UCL/SST/ICTEAM/ELEN
Prof. **Michel Verleysen**, Promoteur, UCL/SST/ICTEAM/ELEN
Prof. **Thierry Denœux**, Université de Technologie de Compiègne, France
Prof. **Pierre Dupont**, UCL/SST/ICTEAM/INGI
Prof. **Tom Heskes**, Radboud University Nijmegen, Pays-Bas
Prof. **Marco Saerens**, UCL/SSH/LSM/ISYS
Prof. **Louis Wehenkel**, Université de Liège, Belgique

Septembre 2013

Contents

Remerciements

Pour moi, être chercheur, c'est bien plus qu'un métier. C'est avant tout une passion, le but que je m'étais fixé en arrivant à l'Université catholique de Louvain en 2002. A l'époque, je n'avais encore qu'une idée très vague de ce que pouvait être la recherche en général et le machine learning en particulier. Onze années plus tard, ça y est, je vais enfin devenir docteur et c'est l'occasion rêvée de remercier ceux qui m'ont aidé au long d'un cheminement qui ne fait que commencer !

En premier lieu, je veux remercier Michel Verleysen. D'abord parce, bien que Michel soit une personne extraordinairement occupée, il arrive toujours à trouver un moment à vous consacrer. De fait, Michel est quelqu'un de disponible et intègre qui m'a été d'un secours précieux pour me construire en tant que chercheur. Lors des corrections d'articles, ses questions m'ont parfois contrarié : Michel met souvent le doigt où ça fait mal. Ca n'a sûrement pas été de tout repos pour lui, vu mon caractère tétu, mais à chaque fois l'article n'en est ressorti que meilleur. Merci donc à lui d'avoir supporté mon entêtement ! Chaque année, Michel organise à Brugges la conférence ESANN où règne une ambiance bon enfant, pendant laquelle j'ai fait la rencontre de gens intéressants et sympathiques qui reviennent régulièrement à l'ESANN, signe de la qualité de l'évenement. C'est également là que les membres du Machine Learning Group se retrouve chaque année pour une semaine qui crée des liens forts entre ceux-ci. Merci à Michel pour le travail qu'il abat chaque année pour que l'ESANN soit toujours un succès.

Lorsque, comme moi, on est assistant, faire un doctorat signifie aussi consacrer une partie importante de son temps aux étudiants. J'ai eu pour cela l'occasion de travailler avec Piotr Sobieski qui m'a appris énormément. Je lui suis reconnaissant de m'avoir donné ma chance dans des cours de physique pour lesquels je n'étais à priori pas très qualifié. Au départ moyennement convaincu par les APPs, j'ai découvert avec lui tout ce qui se cache derrière et je suis désormais convaincu qu'un étudiant doit être actif dans son apprentissage. A

quelques mois de l'éméritat, Piotr Sobieski a toujours une énergie incroyable qu'il consacre à ses chers *students*. J'espère un jour pouvoir lui ressembler en tant qu'enseignant et apporter autant aux étudiants !

En plus de mon promoteur, mon jury de thèse est composé de six personnes. Marco Saerens m'a permis de mettre le pied dans le monde de la recherche, en supervisant mon mémoire de fin d'études et en me conseillant de rencontrer Michel Verleysen. Merci à lui pour tous ses conseils. Membre actif du Machine Learning Group, Pierre Dupont a toujours une réponse à proposer à celui qui se pose des questions. Je le remercie pour ses conseils. Tom Heskes m'a accueilli à la Radboud University Nijmegen pendant trois mois. Enfin, je remercie Thierry Denœux et Louis Wehenkel d'avoir accepté de faire partie de mon jury et de m'avoir prodigué leurs conseils avisés. Merci également à Christophe De Vleeschouwer d'avoir accepté de présider ma défense de thèse.

La recherche se fait rarement seul. Merci à Gaël de Lannoy pour m'avoir donné la chance d'aborder avec lui les électrocardiogrammes au début de ma thèse. J'ai apprécié travailler avec Gaël et nos travaux communs furent pour moi la source de nombreuses réflexions. Merci également à Gauthier Doquire, qui m'a permis d'aborder le domaine passionnant de la feature selection. Partie d'une simple question, cette collaboration a été pour moi riche et agréable. Merci aussi à Emilie Renard qui, malgré qu'elle ait changé d'équipe à présent, a apporté une présence féminine et sa bonne humeur pendant près d'un an dans notre bureau au Maxwell. Merci enfin aux membres du Machine Learning Group pour les moments passés ensemble et les discussions que j'ai eues avec eux. J'espère pouvoir en faire partie encore longtemps.

Au cours de mes quelques années de thèses, j'ai eu l'occasion de séjourner six semaines à la Altoo University. J'y ai reçu un accueil formidable du Dr Amaury Lendasse et de son équipe; je les en remercie. Durant ce séjour, j'ai pu apprécier le pragmatisme dont fait preuve le Dr Amaury Lendasse, ce qui me pousse aujourd'hui à chercher un équilibre entre théorie et pratique dans mes propres recherches.

La recherche est une passion qui a tendance à déborder des horaires de bureau. Mes amis et ma famille pourront en être témoins et je les remercie pour leur soutien, en particuler Lorianne et Bruno Obsomer.

Enfin, *last but not least*, je veux remercier ma femme, sans qui je ne serais jamais arrivé au bout de cette thèse. Sans son écoute dans les moments heureux ou difficiles de ma thèse, je n'en serais pas là aujourd'hui. Merci Aurélie de me rendre heureux chaque jour et de me soutenir !

Abstract

This thesis addresses three challenge of machine learning: high-dimensional data, label noise and limited computational resources.

Learning is usually hard in high-dimensional spaces, due to the curse of dimensionality and other phenomena like the concentration of distances. One can either handle such data with specific tools or try to reduce their dimensionality using e.g. feature selection. The first contribution of this thesis is to study the adequacy of mutual information to select relevant subsets of features. For both classification and regression problems, mutual information is shown to be a sensible criterion for feature selection in most cases. Counterexamples are discussed, where mutual information fails to select optimal features with respect to common error criteria for classification and regression. However, the probability and impact of such failures is also shown to be limited.

The second contribution of this thesis is a survey of the label noise literature. Indeed, label noise is an important problem in classification, whose consequences are various and complex. For example, this thesis shows that label noise affects the segmentation of electrocardiogram signals and the results of feature selection. In each case, a new algorithm is proposed to deal with label noise using a probabilistic modelling introduced by Lawrence and Schölkopf. Afterwards, a more generic framework is proposed to deal with instances which have a too large influence on learning. This framework is used to robustify several probabilistic learning algorithms.

The last contribution of this thesis is the study of large extreme learning machines. Indeed, extreme learning is a recent trend in machine learning which allows learning non-linear models much faster than other state-of-the-art methods. Extreme learning machines are single layer feedforward neural networks whose hidden layer is randomly initialised and not optimised during learning. Only the output weights of such networks have to be optimised, which explains why learning becomes much faster. This thesis shows that when the number of hidden neurons is large, overfitting can be avoided using regularisation. In this

case, a new kernel can be defined using extreme learning, which is shown to give good results for both classification and regression problems. This kernel offers a compromise between prediction accuracy and computational needs which can be useful in contexts where computational time is precious.

Notations and Acronyms

Here is a list of the common notations used in this dissertation:

- X: random variable;
- x: observation of the random variable X;
- p_X: probability density/mass function of the random variable X;
- $p_{X,Y}$: joint probability density/mass function of X and Y;
- $p_{Y|X}$: conditional probability density/mass function of Y given X;
- $\mathbb{E}_X\{f(X)\}$: expected value of the random variable $f(X)$;
- $H(X)$: entropy of X;
- $H(Y|X)$: conditional entropy of Y given X;
- $H(Y|X=x)$: specific conditional entropy of Y given $X=x$;
- $I(X;Y)$: mutual information between X and Y;
- $P_e(x)$: probability of error when $X=x$;
- $\mathcal{X}$: set of values, typically the domain of the random variable X;
- $|\mathcal{X}|$: size (number of elements) of the set $\mathcal{X}$;
- n: number of training instances;
- d: dimensionality of instances;
- $\|x\|_2$: L2 norm of x;
- $\boldsymbol{\theta}$: vector of model parameters;
- c_d: volume of a d-dimensional unit hypersphere;

- $\epsilon_k(i)$: diameter of the hypersphere containing the k nearest neighbours of x_i;
- $\epsilon_k(i|y)$: diameter of the hypersphere containing the k nearest neighbours of x_i in class y;
- ν: number of degrees of freedom of a Student distribution;
- tanh: hyperbolic tangent;
- erf: error function;
- ψ: digamma function;
- B: beta function.

Here is a list of the common acronyms used in this dissertation:

- AFD: abnormally frequent data;
- ECG: electrocardiogram signal;
- ELM: extreme learning machine;
- EM: expectation maximisation;
- GMM: Gaussian mixture model;
- HMM: hidden Markov model;
- LARS: least angle regression;
- MAE: mean absolute error;
- MI: mutual information;
- MSE: mean square error;
- PCA: principal component analysis;
- SVM: support vector machine;
- SVR: support vector regression.

Part I

Discussion of the Articles

Chapter 1

Introduction

> The most exciting phrase to hear in science, the one that heralds new discoveries, is not "Eureka!" but "That's funny...".
>
> *Isaac Asimov, writer*

This introduction gives the context of the thesis, introduces the main contributions and discusses them in relation with collaborations.

1.1 Context of the Thesis

In recent years, machine learning has become a mature field of research. Efficient methods for standard problems like classification or regression have been developed and allow users to make accurate predictions. The large number of journals and conferences dedicated to machine learning demonstrates its vitality. However, in recent applications, the machine learning specialist often faces challenges like e.g. high-dimensional data, non-uniform label noise and limited computational resources. This thesis focuses on these three challenges and is divided into two parts: the first part contextualises several published articles which are gathered in the second part. The first part of the thesis should be considered as a general introduction to the published papers and therefore contains only a minimal amount of technical material. In the second part, each article is shortly introduced by giving the publication context and linking it to other published papers.

Inference from high-dimensional data is known to be hard, due to the curse of dimensionality. Indeed, in such cases, the number of samples is usually insufficient to fill the high-dimensional space, which makes learning hard. A common solution to this problem is feature selection, which consists in using only a small subset of the features for inference. Mutual information has been used for years in the feature selection community to select relevant features. For example, in classification, mutual information is often seen as a proxy to the risk. However, [1] shows that selecting the feature subset which maximises mutual information does not necessarily minimise the risk. In particular, [2] gives counterexamples and studies under which conditions mutual information can be used in classification, showing that mutual information is in general a good choice. Also, investigations have been carried in regression where it appears that mutual information can be used under reasonable hypotheses [3]. These works on the adequacy of mutual information for feature selection tasks are discussed in Chapter 2.

Label noise is also an important challenge in classification. Indeed, experts often make errors when they label instances, which can hurt the performances of inferred models if the possible errors are not properly handled. However, label noise is difficult to define and many works in the literature do not rigorously describe the effect of label noise on inference. In order to fill this gap, Chapter 3 proposes a thorough survey of the label noise literature [4]. Next, an approach developed by Lawrence and Schölkopf to handle label noise is adapted in the context of automated electrocardiogram segmentation with hidden Markov models [5]. Also, the problem of mutual information estimation in the presence of label noise is considered [3]. Eventually, a generic, non-parametric approach is proposed to deal with data which are abnormally frequent with respect to their theoretical probability of occurrence [6] (including e.g. outliers, instances with wrong labels, etc.). This approach allows to robustify any maximum likelihood-based inference algorithm like e.g. linear regression, logistic regression, kernel Ridge regression or principal component analysis.

Limitation on the computational resources is also a problem faced in machine learning. In response, the extreme learning framework has recently been proposed [7, 8], which allows one to train single-layer neural networks very fast and yet to obtain good prediction performances. In Chapter 4, the behaviour of extreme learning machines is analysed when the number of neurons becomes very large. In particular, it is shown that in this case extreme learning machines can be formulated in terms of a kernel [9, 10].

1.2 Contributions of the Thesis

The main contributions of this thesis are

- theoretical and empirical proofs of mutual information adequacy for feature selection in classification and regression [1, 2, 11, 12];
- a comprehensive survey on label noise [4];
- a method to deal with label noise in the case of ECGs [5];
- a label noise-tolerant estimator of mutual information [3];
- a method to deal with abnormally frequent data like e.g. mislabelled instances or outliers [6];
- empirical and theoretical arguments for using large numbers of neurons in extreme learning machines [9, 10];
- a kernel based on extreme learning machines [10].

Other contributions (not discussed in this dissertation) include

- a method for Q-learning in two-player games [13, 14];
- a methodology to cluster curves in geography [15–17];
- new types of hidden Markov models for ECG segmentation [18–20];
- a feature selection method for non-linear regression problems based on extreme learning machines [21].

1.3 Publications in Relation with Collaborations

During my thesis, I had the opportunity to work with several researchers. Consequently, many of the works discussed in this thesis are the result of collaborations. This section gives more details about these collaborations, including articles which are not included in this thesis.

The works on automated electrocardiogram (ECG) segmentation have been carried with Gael de Lannoy. We have published several papers using wavelets and hidden Markov models to recognise patterns in ECGs [18–20]. We also proposed a solution to deal with label noise in the case of ECGs [5]. Gael provided the data and took care of the wavelet preprocessing of ECGs, whereas we developed Markov models with Gaussian mixture modelling simultaneously.

My contribution was particularly important in [5,20] where I proposed the ideas, developed the models, implemented them and tested them on ECGs.

The results obtained during my master thesis under the supervision of Marco Saerens have been published in [13,14]. I also participated in a research in geography with Isabelle Thomas, where I took care of the data analysis which was in turn interpreted by geographers [15–17].

During a stay at the Altoo University in Finland, I had the opportunity to work on extreme learning machines with Mark van Heeswijk, Yoan Miche and Amaury Lendasse. At Altoo, I finished a journal paper [10] in the line of my ESANN work [9] and pursued research on feature selection with extreme learning [21]. Whilst I implemented the proposed method and tested it on real datasets, Mark van Heeswijk has provided the results for classical feature selection methods for comparison purposes. Also, I helped a master student with her experiments [22].

The publications on mutual information are the results of a fruitful collaboration with Gauthier Doquire. Maximising mutual information is often assumed to be equivalent to minimising the classification risk, but we showed in [1] that this is not necessarily true, even if mutual information seems to be a good criterion in practice [2, 11]. We performed a similar analysis for regression [12] and also proposed a label noise-tolerant estimator of mutual information [3]. These works are the result of frequent discussions and it is therefore not easy to distinguish who did what. On the one hand, Gauthier provided datasets and performed most of the actual feature selection experiments, in particular for [11]. On the other hand, I performed most of the theoretical developments and most of the new implementations, in particular for [2, 3, 12]. Moreover, I also obtained the numerical results and graphs shown in [1–3], which were the subject of many discussions with Gauthier.

Eventually, on my own, I also wrote a survey on label noise [4] and proposed a new method to deal with abnormally frequent data like e.g. mislabelled instances or outliers [6].

1.4 List of Publications

Here is a comprehensive list of the publications written during my PhD thesis, including those which are not discussed in this dissertation. The keys of the entries correspond to the keys in the bibliography

[13] Benoît Frénay and Marco Saerens. Ql2, a simple reinforcement learning scheme for two-player zero-sum markov games. In *Proceedings of the 16th*

International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2008), pages 137–142, 2008

[20] Benoît Frénay, Gael de Lannoy, and Michel Verleysen. Improving the transition modelling in hidden markov models for ecg segmentation. In *Proceedings of the 17th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2009)*, 2009

[18] Benoît Frénay, Gaël de Lannoy, and Michel Verleysen. Emission modelling for supervised ecg segmentation using finite differences. In *4th European Conference of the International Federation for Medical and Biological Engineering*, pages 1212–1216. 2009

[14] Benoît Frénay and Marco Saerens. A simple reinforcement learning scheme for two-player zero-sum markov games. *Neurocomputing*, 72(7-9):1494–1507, 2009

[9] Benoît Frénay and Michel Verleysen. Using svms with randomised feature spaces: an extreme learning approach. In *Proceedings of The 18th European Symposium on Artificial Neural Networks (ESANN)*, pages 315–320, 2010

[5] Benoît Frénay, Gaël de Lannoy, and Michel Verleysen. Label noise-tolerant hidden markov models for segmentation: application to ecgs. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part I*, pages 455–470, Athens, Greece, 2011

[10] Benoît Frénay and Michel Verleysen. Parameter-insensitive kernel in extreme learning for non-linear support vector regression. *Neurocomputing*, 74(16):2526 – 2531, 2011

[3] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. Estimating mutual information for feature selection in the presence of label noise. Accepted in Computational Statistics & Data Analysis, 2013

[1] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. On the potential inadequacy of mutual information for feature selection. In *Proceedings of the 20th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, pages 501–506, 2012

[2] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification. *Neurocomputing*, 112:64–78, 2013

[4] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. Submitted to IEEE Transaction on Neural Networks, 2013

[12] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. Is mutual information adequate for feature selection in regression ? *Neural Networks*, 48:1–7, 2013

[6] Benoît Frénay and Michel Verleysen. Pointwise probability reinforcements for robust statistical inference. Submitted to Neural Networks, 2013

[21] Benoît Frénay, Mark van Heeswijk, Yoan Miche, Michel Verleysen, and Amaury Lendasse. Feature selection for nonlinear models with extreme learning machines. *Neurocomputing*, 102:111–124, 2013

[19] G. Lannoy, B. Frénay, M. Verleysen, and J. Delbeke. Supervised ecg delineation using the wavelet transform and hidden markov models. In *4th European Conference of the International Federation for Medical and Biological Engineering*, pages 22–25. 2009

[11] Gauthier Doquire, Benoît Frénay, and Michel Verleysen. Risk estimation and feature selection. In *Proceedings of the 21th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013)*, 2013

[16] Isabelle Thomas, Pierre Frankhauser, Benoît Frénay, and Michel Verleysen. Clustering fractal urban patterns with curves of scaling behavior. In *Proceedings of the 49th Congress of the European Regional Science Association " Territorial cohesion of Europe and integrative planning" (ERSA 2009)*, 2009

[15] Isabelle Thomas, Pierre Frankhauser, Benoît Frénay, and Michel Verleysen. Clustering patterns of urban builtup areas with curves of fractal scaling behavior. In *Proceedings of ASRDLF 2009, l'Association de Science Régionale de Langue Française*, 2009

[17] Isabelle Thomas, Pierre Frankhauser, Benoît Frénay, and Michel Verleysen. Clustering patterns of urban built-up areas with curves of fractal scaling behaviour. *Environment and Planning B: Planning and Design*, 37(5):942–954, 2010

[22] Laura Kainulainen, Yoan Miche, Emil Eirola, Qi Yu Yu, Benoît Frénay, Eric Séverin, and Amaury Lendasse. Ensembles of local linear models for bankruptcy analysis and prediction. *Case Studies in Business, Industry and Government Statistics (CSBIGS)*, 4(2):116–133, 2011

Chapter 2

Adequacy of Mutual Information for Feature Selection

> The beginning of knowledge is the discovery of something we do not understand.
>
> *Frank Herbert, writer*

Contents

This chapter discusses the adequacy of mutual information for feature selection in classification and regression. First, Section 2.1 discusses high-dimensional data and feature selection. Then, Section 2.2 reviews information theoretic quantities for feature selection, whose estimation is discussed in Section 2.3. Section 2.4 considers feature selection in classification and the adequacy of mutual information in this context is analysed in Section 2.5. Similarly, Section 2.6 tackles regression and the adequacy of mutual information in this context is discussed in Section 2.7. Eventually, Section 2.8 concludes about whether mutual information should be used in feature selection or not. The results discussed in Sections 2.5 and 2.7 have been published in [1, 2, 11, 12].

2.1 Dealing with High-Dimensional Datasets

In machine learning, difficulties occur when the dimensionality of data is large with respect to the number of training instances. In such cases, the data space is only sparsely filled with training instances and learning algorithms can easily overfit the training sample. This problem decreases the generalisation performances on new instances and is well known as the curse of dimensionality [23, 24]. A typical example of application where this problem occurs is micro-array analysis [25–27]. Also, learning is usually slower in high-dimensional spaces and counterintuitive phenomena may occur, like e.g. the concentration of distances [28]. Eventually, interpretability may be important and models may be more difficult to interpret with a large number of features.

There exist various and complementary approaches to deal with high-dimensional training samples. First, one can restrict himself to using simple models which are less likely to overfit. For example, in micro-array analysis, most models are linear [25,26,29–31]. Second, regularisation and other complexity control methods can be used to reduce overfitting. Eventually, preprocessing methods can be used to reduce the dimensionality of data, which are either projection methods like principal component analysis or feature selection methods.

This chapter focuses on feature selection, which consists in using only a subset of the available features for learning and prediction. Feature selection methods are widely used in the literature, since they allow to reduce the risk of overfitting and to obtain easy-to-interpret models which are appreciated in industrial and medical areas. Also, decreasing the number of features speeds up learning and reduces the curse of dimensionality and the concentration of distances. Notice that feature selection is different from sufficient dimension reduction [32], since features which contain information may be discarded if necessary. Feature selection also has the advantage over projection methods

that the original features are not altered.

There exists three major types of feature selection methods [33]: wrappers, filters and embedded methods. Wrappers [34] select the features that allow a given model to obtain the best performances. Wrappers have the advantage of being optimal for a given model, in the sense that the selected features are the best features with respect to the error criterion if an exhaustive search is performed. However, the results of the feature selection are only valid for the chosen model. Moreover, feature selection with wrappers can be time-consuming, since the meta-parameters and parameters of the model must be optimised for each considered feature subset. Filters use criteria like e.g. correlation or mutual information to select features before the model of interest is learnt. In practice, filters use simple models to estimate a measure of relevance of the features, like univariate linear models in the case of the correlation or nearest neighbour probability density estimators in the case of the mutual information (see Section 2.3). Filters are usually much faster than wrappers, but they may produce feature subsets which are suboptimal with respect to a given model. Eventually, embedded methods allow us to directly embed feature selection into the learning process. The most well-known example is the least absolute shrinkage and selection operator (LASSO) [35] which can be performed using the least angle regression (LARS) [36–38]. The pros and cons of embedded methods are similar to those of wrappers, except that they are much faster. Moreover, whereas wrappers are black box methods, feature selection is performed in embedded methods in a structural way. Hence, the parameters of the models can be interpreted to gain more knowledge. Also, prior knowledge and constraints can be more easily be taken into account with embedded methods, like e.g. in the case of the group LASSO [39].

Filter methods for feature selection use various criteria to assess the usefulness of features. For example, correlation can be used to detect linear relationships between a given feature and the target. However, since features are often linked to the predicted quantity in a non-linear and multivariate way, correlation is not sufficient for real-world problems. Instead, one can use mutual information, whose adequacy for feature selection is the main subject of this chapter. This criterion has been used in a large literature of successful applications [40–49] since the seminal work of [50].

2.2 Information-Theoretic Feature Selection

With respect to other filter methods for feature selection, the approaches based on mutual information have the advantage of being theoretically well-grounded.

Indeed, mutual information comes from information theory, where it is used to measure the reduction of uncertainty about a random variable when another variable is observed [51].

In information theory, it can be axiomatically shown [52] that the uncertainty on the value of a random variable X with probability density function p_X is quantified by the entropy

$$H(X) = \mathop{\mathbb{E}}_{X} \{ -\log p_X(X) \} . \tag{2.1}$$

Entropy [53] is a widely used measure of uncertainty in machine learning, which has proven useful in many applications [43]. Two closely related quantities can be used in feature selection: the conditional entropy

$$H(Y|X) = \mathop{\mathbb{E}}_{X,Y} \left\{ -\log p_{Y|X}(Y|X) \right\} \tag{2.2}$$

is the uncertainty on Y once X is known and the mutual information

$$I(X;Y) = H(Y) - H(Y|X) \tag{2.3}$$

can be seen as the reduction of uncertainty on Y when X is known. If X is a subset of features and Y is the quantity to be predicted, $I(X;Y)$ can be used to measure the information contained in X about Y. In other words, according to information theory, mutual information (MI) can be used to select subsets of features which are strongly related to the target [42, 50]. In fact, MI can be alternately written as the Kullback-Leibler divergence between $p_X(X)p_Y(Y)$ and the joint probability $p_{X,Y}(X,Y)$

$$I(X;Y) = \mathop{\mathbb{E}}_{X,Y} \left\{ \log \frac{p_{X,Y}(X,Y)}{p_X(X)p_Y(Y)} \right\} , \tag{2.4}$$

which shows that MI also measures the statistical dependency between the two random variables X and Y. $I(X;Y)$ is zero if X and Y are independent and increases as their mutual dependency grows.

With respect to other criteria such as the correlation, MI has the advantage to naturally extend to multivariate and non-linear problems, which is of fundamental importance if greedy feature subset search procedures (such as forward or backward) and non-linear models are used. Moreover, fast estimators of mutual information are available [42, 54–56]. Eventually, the large literature of successful applications using MI in feature selection shows the interest of this criterion [40–49].

2.3 Fast Entropy Estimation

In order to use MI, one needs an efficient entropy estimator, like e.g. the Kozachenko-Leonenko estimator [54] which is extensively used in this thesis. This estimator uses a nearest neighbour probability density estimator, which assumes that the density remains constant in a small region around each instance. If $\epsilon_k(i)$ is the diameter of the hypersphere containing the k nearest neighbours of the instance x_i, the estimate is

$$\log \hat{p}_X(x_i) = \psi(k) - \psi(n) - \log c_d - d \log \epsilon_k(i) \tag{2.5}$$

where ψ is the digamma function, d is the dimensionality and $c_d = 2\pi^{\frac{d}{2}}/\Gamma(\frac{d}{2})$ is the volume of a d-dimensional unit hypersphere. Equation (2.5) has the advantage over other nearest neighbour probability density estimators to define an actual probability density function. Using Equation (2.5), [54] proposes the empirical entropy estimator

$$\hat{H}(X) = -\frac{1}{n}\sum_{i=1}^{n} \log p_X(x_i) = \psi(n) - \psi(k) + \log c_d + \frac{d}{n}\sum_{i=1}^{n} \log \epsilon_k(i) \tag{2.6}$$

which can used to estimate MI [55, 57] and has proven to give good results for feature selection in classification and regression [46, 57, 58]. Notice that, in theory, the number of neighbours k should be optimised using e.g. cross-validation [59,60], but such methods are computationally consuming [57]. This could greatly limit the interest of MI estimators in filter-based feature selection. Instead, it is advised in the MI estimation literature to use small values for k [46, 55, 58, 61].

2.4 Mutual Information for Classification

The two following sections deal with the use of mutual information for feature selection in classification. In this context, Gómez et al. [57] has proposed a MI estimator based on the Kozachenko-Leonenko entropy estimator. Indeed, the definition of MI is symmetric and one can write

$$I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y). \tag{2.7}$$

Since the class Y is a discrete variable, the conditional entropy is

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p_Y(y) H(X|Y = y) \tag{2.8}$$

where the specific conditional entropy is defined as

$$H(X|Y=y) = \mathop{\mathbb{E}}_{X} \left\{ -\log p_{X|Y} p(X|y) \right\}, \tag{2.9}$$

Gómez et al. propose to estimate MI as

$$\hat{I}(X;Y) = \hat{H}(X) - \sum_{y \in \mathcal{Y}} \hat{p}_Y(y) \hat{H}(X|Y=y) \tag{2.10}$$

where $\hat{p}_Y(y) = n_y/n$ is the observed frequency of class y. Using the entropy estimator given in Section 2.3, Gomez et al. eventually obtain

$$\hat{I}(X;Y) = \psi(n) - \frac{1}{n} \sum_{y \in \mathcal{Y}} n_y \psi(n_y) + \frac{d}{n} \left[\sum_{i=1}^{n} \log \epsilon_k(i) - \sum_{y \in \mathcal{Y}} \sum_{i|y_i=y} \log \epsilon_k(i|y) \right] \tag{2.11}$$

where $\epsilon_k(i|y)$ is the diameter of the hypersphere containing the k nearest neighbours of the instance x_i in class y.

According to the discussion in Section 2.2, $I(X;Y)$ can be interpreted in classification as the reduction in uncertainty about the class Y of an instance when features X are observed. This interpretation is often used to justify the use of MI for feature selection in classification. However, the final goal in classification is usually to minimise the misclassification probability that can achieved by an optimal classifier:

$$P_e = \mathop{\mathbb{E}}_{X,Y} \left\{ 1 - \max_{y \in \mathcal{Y}} p_{Y|X}(y|X) \right\}. \tag{2.12}$$

Hence, MI will be optimal for feature selection in classification if choosing the feature subset which maximises MI always corresponds to choosing the feature subset which minimises the misclassification probability. Since $H(Y)$ only depends on the class priors and is constant for a given classification problem, Equation (2.3) shows that maximizing the mutual information $I(X;Y)$ is equivalent to minimizing the conditional entropy $H(Y|X)$ in a feature selection context. This section and Section 2.5 show theoretical and empirical results for $H(Y|X)$ which also apply to $I(X;Y)$.

There exist two well-known bounds relating the misclassification probability and the conditional entropy $H(Y|X)$. First, the Hellman-Raviv inequality [62] is an upper bound of the misclassification probability (also called the probability of error)

$$P_e \leq \frac{1}{2} H(Y|X). \tag{2.13}$$

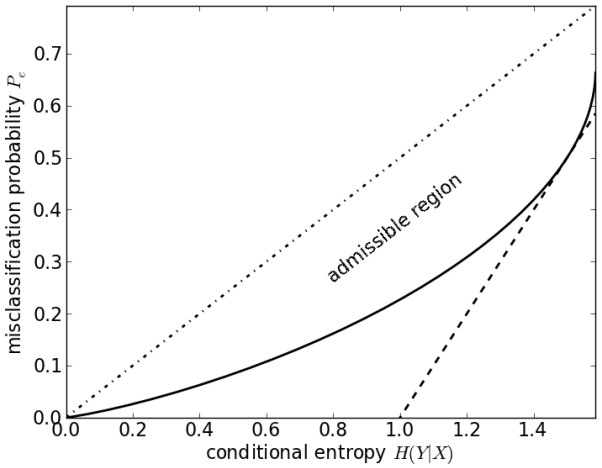


Figure 2.1: Weak Fano bound (dashed line), strong Fano bound (plain line) and Hellman-Raviv bound (dash-dotted line) on the misclassification probability P_e, in terms of the conditional entropy $H(Y|X)$. Figure inspired from [64, 65], reprinted with permission from [1, 2].

Second, the Fano inequalities [63]

$$H(Y|X) \leq 1 + P_e \log_2(|\mathcal{Y}| - 1) \tag{2.14}$$

and

$$H(Y|X) \leq H(P_e) + P_e \log_2(|\mathcal{Y}| - 1) \tag{2.15}$$

allow obtaining lower bounds on the misclassification probability, where $|\mathcal{Y}|$ is the number of classes and $H(P_e) = -P_e \log_2 P_e - (1 - P_e) \log_2(1 - P_e)$ [51]. The weak Fano bound (2.14) is useless in most cases since it is weaker than the trivial bound $P_e \geq 0$ as soon as $H(Y|X) < 1$. The strong Fano bound (2.15) is more useful but difficult to use, since the lower bound on P_e cannot be derived analytically and has to be obtained numerically.

Figure 2.1 shows the Hellman-Raviv bound, the weak Fano bound and the strong Fano bound for a three-class classification problem. In the feature selection literature, several authors use these bounds to justify the use of MI in classification [64–66]. Indeed, since the lower and upper bounds on the misclassification probability decrease as $H(Y|X)$ decreases, they claim that maximising MI is equivalent to minimising the misclassification probability. As shown in the next section, this is not necessarily true.

2.5 Adequacy of Mutual Information for Classification

As discussed in Section 2.4, the final goal in classification is to minimise the misclassification probability. Some authors claim that MI can be used as a proxy for the risk [64–66], which makes MI an optimal criterion for feature selection. However, this thesis shows that this is not necessarily true.

2.5.1 Common Misconceptions about Bounds

Figure 2.2 shows the Hellman-Raviv bound, the weak Fano bound and the strong Fano bound for a binary classification problem. Dots corresponds to the pairs $\langle H(Y|X), P_e \rangle$ computed for several randomly built binary classification problems with two binary features. As detailed in [1, 2], the problems are generated as follows: (i) the two values $P(Y = y)$ (for $y \in [0, 1]$) and the four values $P(X = x|Y = y)$ (for $x \in [0, 1]$ and $y \in [0, 1]$) are drawn from the uniform distribution $\mathcal{U}(0, 1)$, (ii) these values are normalised to ensure that they represent probabilities, i.e. $\sum_y P(Y = y) = 1$ and $\sum_x P(X = x|Y = y) = 1$ for each y and (iii) probabilities $P(X)$ and $P(Y|X)$ are eventually computed using marginalisation and Bayes' theorem. For each problem, P_e and $H(Y|X)$ can be computed exactly since all the necessary probabilities are known.

The pairs are clearly scattered in the region between the strong Fano lower bound and the Hellman-Raviv upper bound, which illustrates that there exists no deterministic relationship between MI and the misclassification probability. A natural question is whether it is possible to find a situation in feature selection where MI is not optimal, i.e. $I(X_1; Y) > I(X_2; Y)$ and $P_e(X_1) > P_e(X_2)$ for two feature subsets X_1 and X_2 which are compared. The answer is positive, as shown in the following example taken from [1, 2].

2.5.2 Example of Mutual Information Failure

Let us consider a binary classification problem ($Y \in \{0, 1\}$) where classes are balanced, i.e. $p_Y(0) = p_Y(1) = 0.5$. Two binary tests $X_1 \in \{0, 1\}$ and $X_2 \in \{0, 1\}$ are available to discriminate between classes, but the user can only afford to use one test and faces a feature selection problem. Also, it has been experimentally observed that the conditional distributions $p_{X_i|Y}$ of both tests

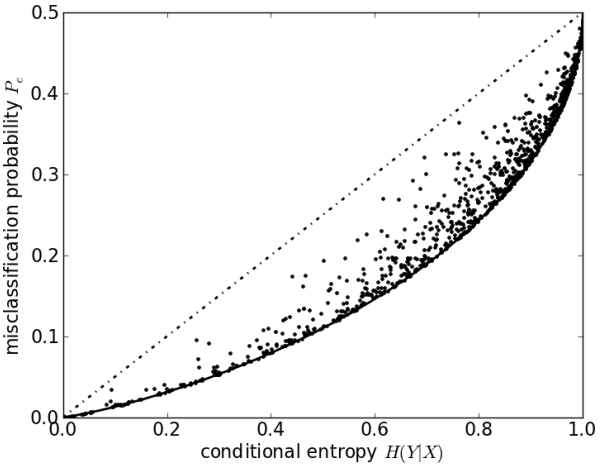


Figure 2.2: Pairs of $\langle H(Y|X), P_e \rangle$ values corresponding to random binary classification problems with two binary features. The strong Fano bound (plain line) and the Hellman-Raviv bound (dash-dotted line) relating P_e to $H(Y|X)$ are shown. Reprinted with permission from [1, 2].

X_1 and X_2 given Y are given by

	$Y=0$	$Y=1$
$X_1=0$	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

.

Using marginalisation and Bayes' theorem, it is straightforward to show that the posteriors $p_{Y|X_i}$ are given by

	$X_1=0$	$X_1=1$
$Y=0$	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

.

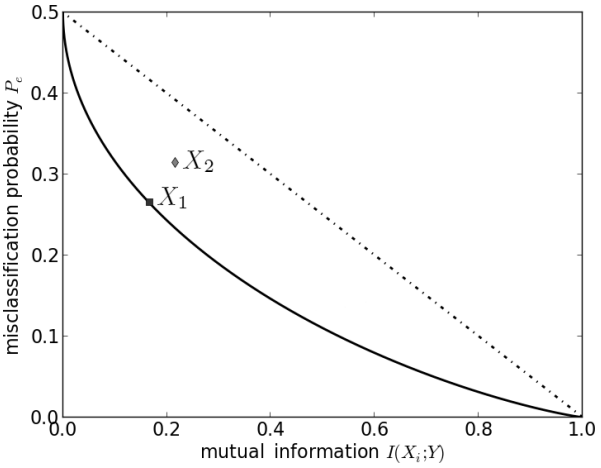


Figure 2.3: Example of mutual information failure for feature selection, with the strong Fano bound (plain line) and the Hellman-Raviv bound (dash-dotted line). X_1 and X_2 are feature subsets. Adapted from [2].

On the one hand, using only test X_1 gives $P_e = 0.265$ while $I(X_1;Y) = 0.167$. On the other hand, using only test X_2 gives $P_e = 0.314$ and $I(X_2;Y) = 0.217$. In this example, a MI-based feature selection algorithm would choose X_2, but P_e is actually smaller when X_1 is used. Selecting X_2 based on MI leads to an increased misclassification probability, i.e. MI fails. This is illustrated in Figure 2.3, where $I(X_2;Y) = H(Y) - H(Y|X_2)$ is larger than $I(X_1;Y) = H(Y) - H(Y|X_1)$ while $P_e(X_2)$ is simultaneously larger than $P_e(X_1)$.

2.5.3 Bound on the Impact of Failures

Even if MI can fail as a feature selection criterion, it is important to quantify how important the failures are. One can obtain (see [2] for details) an upper bound for the supplementary percentage of samples which are misclassified due to an incorrect choice of feature subset. This difference is called the misclassification probability loss in this thesis. Figure 2.4 shows the upper bound on the misclassification probability loss for the above example. The upper bound is concave with respect to $I(X;Y)$, since it is the difference between the Hellman-Raviv bound and the strong Fano bound. The maximum misclassification probability loss is smaller for extreme values of the mutual information, which suggests that mutual information failures have less important consequences in these cases.

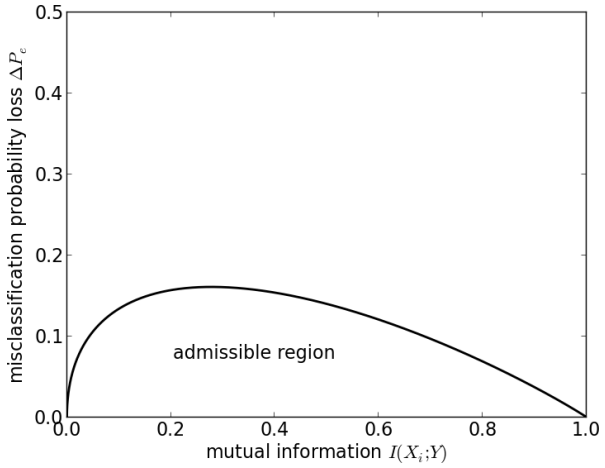


Figure 2.4: Theoretical upper bound on the misclassification probability loss for binary classification with balanced classes. Reprinted with permission from [2].

2.5.4 Empirical Results of Failures

Through extensive experiments, it has been shown in [2] that the impact of MI failures remains limited in practice. This is confirmed in [11] where actual feature selection experiments are performed. Two types of experiments are used in [2], whose results are briefly described here. See [2] for more details about the experimental settings and the results.

First, a large number of discrete and continuous artificial balanced classification problems are built. For each problem, two features X_1 and X_2 are created, whose conditional probability distributions $p_{X_i|Y}$ are randomly built for each class y. This allows to obtain classification problems with very different characteristics. For each pair of features, the feature with the largest mutual information is chosen. If the misclassification probability is also larger for the chosen feature, the pair is an example of failure for mutual information as a feature selection criterion. In case of failure, the difference in probability of error is called the misclassification probability loss ΔP_e, i.e. the percentage of samples which are misclassified due to an incorrect choice of feature. The conditional probability of failure is estimated by counting failures.

Several experimental settings corresponding to various classification problem difficulties are tested in [2]. The main conclusion is that (i) the average probability of mutual information failure is small, (ii) misclassification probability losses remain small and (iii) failures are more likely to occur for moderately

difficult classification problem. For example, Figures 2.5 and 2.6 show the results for such classification problems with discrete and continuous features, respectively. The conditional probabilities of failures are smaller than five percent (see Figure 2.5(b) and 2.6(b)) and the misclassification probability losses ΔP_e are in order of the percent (see Figure 2.5(c) and 2.6(c)). Moreover, the bound given in Section 2.5.3 is met in Figures 2.5(d) and 2.6(d) which also shows that MI failures have smaller impact for large MI values.

The second set of experiments in [2] tackles feature selection for real datasets. The feature selection process is repeated a large number of times for each dataset. For each repetition, 10 features are randomly chosen from the original d features to obtain a different subdataset with smaller dimensionality whose characteristics are similar to those of the original dataset. Then, a forward search using MI is performed, where a forward step is considered as a failure if the selected feature does not minimise the misclassification probability. Figure 2.7 shows the results for the Digits dataset [67], for which the forward search makes it possible to achieve good prediction results with only 5 features (see Figure 2.7(a)). Most of the MI failures occur for very small feature subsets (see Figure 2.7(d)) and their impact remains small (see Figures 2.7(b) and 2.7(c)). These results confirm the conclusion obtained using artificial datasets.

2.5.5 Practical Consequences for Feature Selection

The results obtained in [1, 2] show that MI is not always an optimal criterion for feature selection in classification, if the goal is to minimize the misclassification probability. However, the average percentage of failures is relatively small and the impact of these failures remains limited. Most of the failures occur for intermediate MI values, like e.g. in the first steps of a forward search. Hence, backward search could be a more reliable option, since it directly starts in the region where MI failures are less likely to occur and only reaches the dangerous zone after having found good feature subsets. Another solution consists in using forward search in association with a few backward steps, which could be performed when the feature subset has sufficiently grown. Also, when it is affordable, one could start the feature selection by considering pairs or triplets of features, in order to avoid feature subsets with only one feature. In conclusion, taking some precautions and possibly adapting the search algorithm, mutual information remains a very interesting heuristic for feature selection.

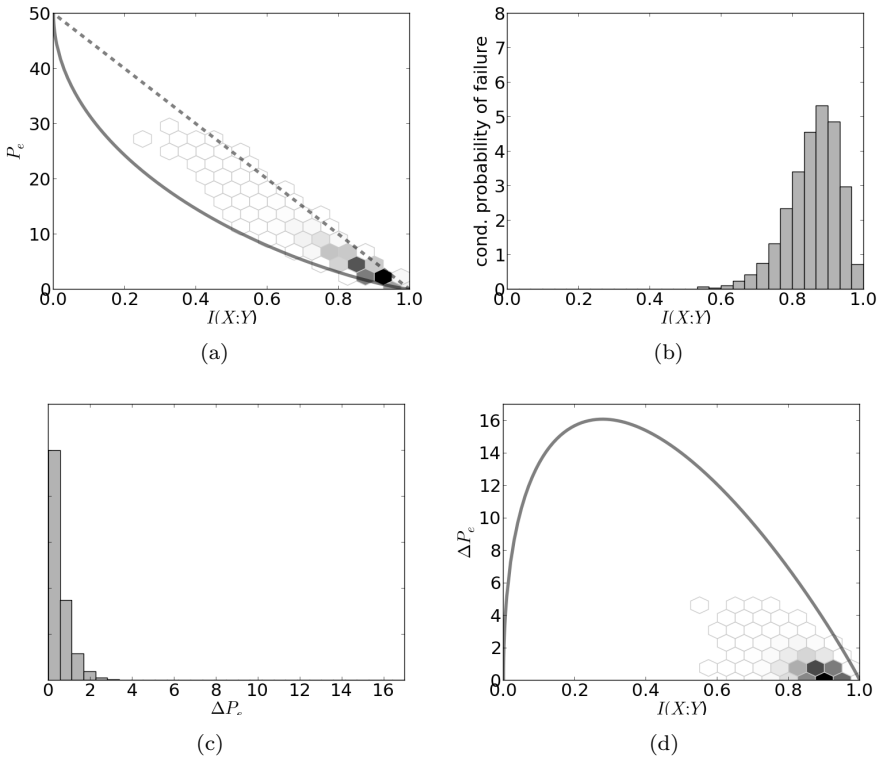


Figure 2.5: Results for artificial binary classification problems with one discrete feature: (a) two-dimensional histogram of the MI and the misclassification probability with the Fano and Hellman-Raviv bounds, (b) conditional probability of failure with respect for several MI values, (c) histogram of the misclassification probability loss in case of failure and (d) two-dimensional histogram of the MI and the misclassification probability loss with the theoretical bound derived in Section 2.5.3. Partially reprinted with permission from [2].

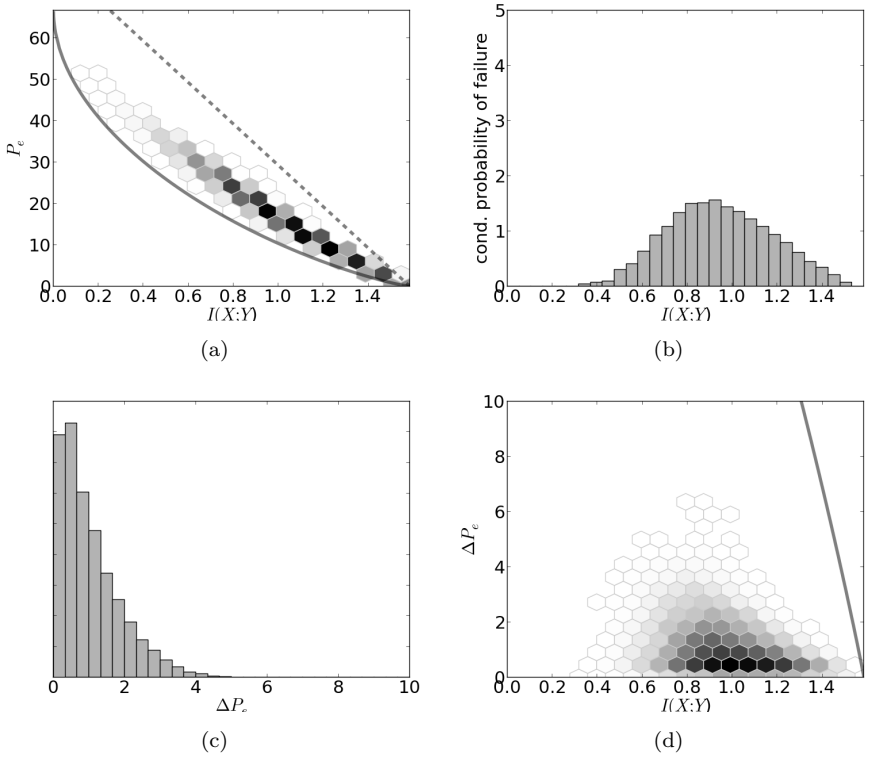


Figure 2.6: Results for artificial three-class classification problems with one continuous feature: (a) two-dimensional histogram of the MI and the misclassification probability with the Fano and Hellman-Raviv bounds, (b) conditional probability of failure with respect for several MI values, (c) histogram of the misclassification probability loss in case of failure and (d) two-dimensional histogram of the MI and the misclassification probability loss with the theoretical bound derived in Section 2.5.3. Partially reprinted with permission from [2].

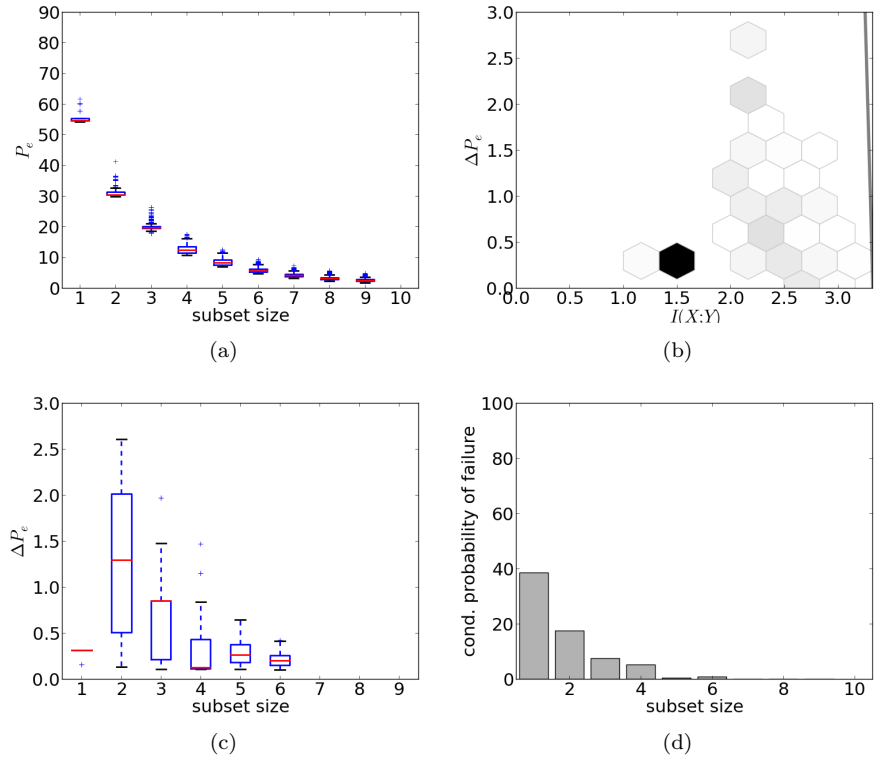


Figure 2.7: Results of MI-based forward search for the Digits dataset: (a) misclassification probability for several feature subset sizes, (b) two-dimensional histogram of the MI and the misclassification probability loss with the theoretical bound derived in Section 2.5.3, (c) misclassification probability loss for several feature subset sizes and (d) conditional probability of failure for several feature subset sizes. Partially reprinted with permission from [2].

2.6 Mutual Information for Regression

In regression, one is typically interested in reducing the estimation error

$$\epsilon = Y - f(X) \tag{2.16}$$

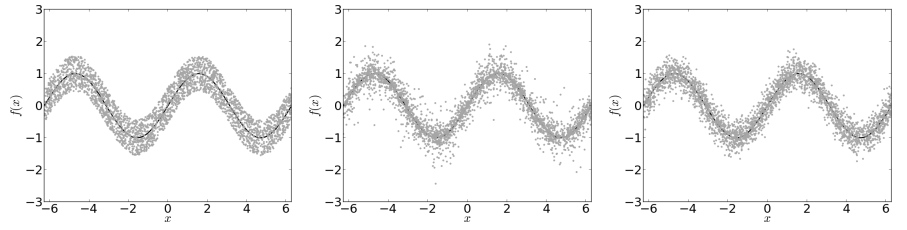
where f is a prediction model. For example, the mean square error (MSE) $\mathbb{E}_{X,Y}\left\{(Y - f(X))^2\right\} = \mathbb{E}_{X,Y}\left\{\epsilon^2\right\}$ and the mean absolute error (MAE) $\mathbb{E}_{X,Y}\left\{|Y - f(X)|\right\} = \mathbb{E}_{X,Y}\left\{|\epsilon|\right\}$ are commonly used to measure the performance of a regression model. Since $H(Y|X) = H(\epsilon|X)$ [12, 51, 52], MI is also related to the estimation error: choosing the feature subset which maximises MI corresponds to minimising the conditional entropy of the estimation error. In intuitive terms, all these criteria measure how much the distribution of the estimation error is spread.

Since the MSE and the MAE are the standard optimality criteria in regression, it is important to known whether using MI is optimal with respect to these two criteria. In other words, is there a deterministic (and if possible monotonic) relationship between the MSE, the MAE and MI ? Surprisingly, this question has not yet been tackled in the machine learning literature. In information theory, there exist relationships between the MSE and MI [68–70], but they are either not relevant in our case or limited to Gaussian estimation errors. The goal of the next section is to answer this question.

2.7 Adequacy of Mutual Information in Regression

This section shows that MI can either be adequate or not for feature selection in regression, depending on the characteristics of the estimation error. In the following examples, it is assumed that when two features are compared at a given feature selection step, the respective conditional entropies of the estimation error belong to the same parametric family. This seems a reasonable hypothesis for example in forward search, since estimation errors at a given forward step will be quite similar.

Figure 2.8 shows a simple functional whose values are polluted by uniform, Laplacian and Gaussian noise. Under the hypothesis that the estimation error belongs to one of these parametric families, [3] shows that there exist a monotonic relationship between the conditional entropy $H(Y|X)$, the MSE and the MAE. This is illustrated by Figure 2.9. Hence, selecting the feature subset which maximises MI also corresponds to minimising the MSE and MAE. In



(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 2.8: Functional $f(x) = \sin(x)$ polluted by uniform, Laplacian or Gaussian target noise with conditional target entropy $H(Y|X) = 0.1$. Reprinted with permission from [3].

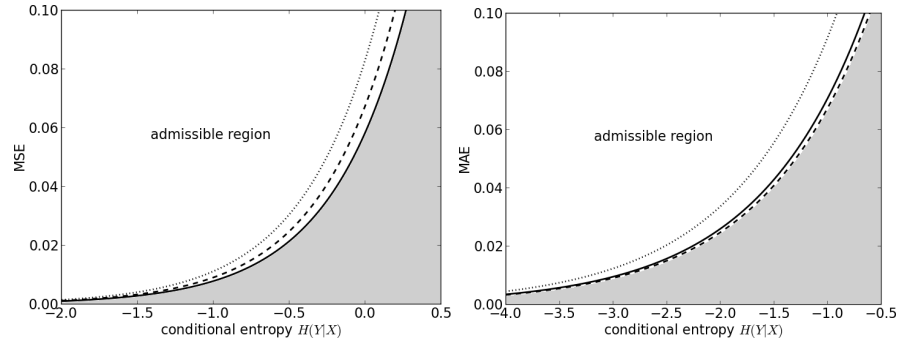
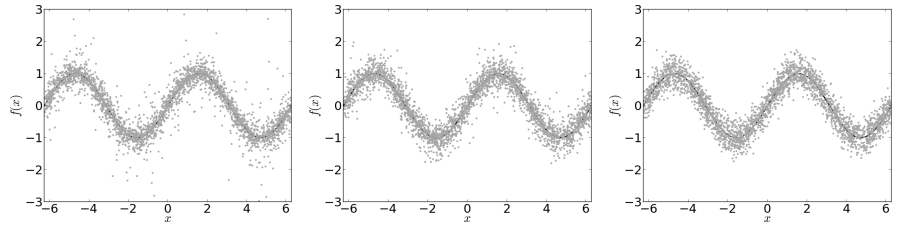


Figure 2.9: MSE and MAE in terms of the conditional target entropy $H(Y|X)$ for identically distributed uniform (dotted line), Laplacian (dashed line) and Gaussian (plain line) estimation error. The Gaussian and Laplacian curves define an admissible region (in white). Reprinted with permission from [3].

such cases, MI is an optimal feature selection criterion, with respect to both MSE and MAE. Notice that if the variance of the estimation error depends on the inputs, then failures may occur like in the case of classification. However, one can conjecture that the frequency and impact of such failures are similar to those observed in the experiments for classification tasks in [2].

Figure 2.10 shows a simple functional whose values have been polluted by Student noise, which is often used for robust modelling [71, 72]. The non-



(a) Student noise $\mathcal{S}(\mu = 0, \nu = 2.3, \sigma = 0.168)$ (b) Student noise $\mathcal{S}(\mu = 0, \nu = 5, \sigma = 0.217)$ (c) Student noise $\mathcal{S}(\mu = 0, \nu = 30, \sigma = 0.259)$

Figure 2.10: Functional $f(x) = \sin(x)$ polluted by Student noises with different parameters and conditional target entropy $H(Y|X) = 0.1$. Reprinted with permission from [3].

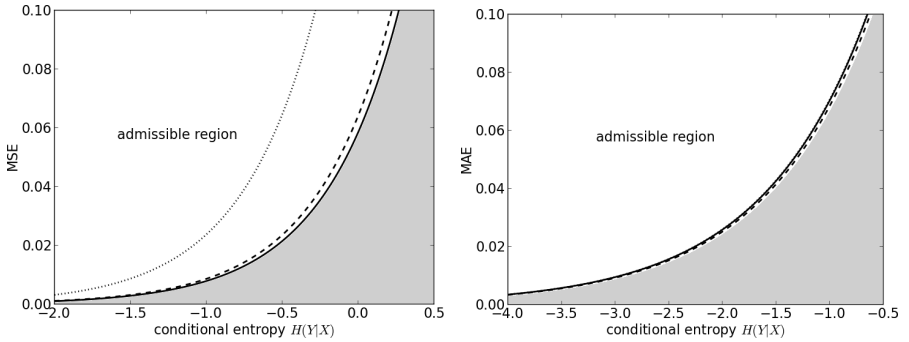
standardised Student distribution is

$$\mathcal{S}\left(\epsilon = e|\mu, \nu, \sigma\right) = \frac{1}{B\left(\frac{1}{2}, \frac{\nu}{2}\right)\sqrt{\nu\sigma^2}}\left[1 + \frac{(e-\mu)^2}{\nu\sigma^2}\right]^{-\frac{\nu+1}{2}} \tag{2.17}$$

where B is the beta function and the two parameters are the number of degrees of freedom ν and the scale σ. Contrarily to the uniform, Laplacian and Gaussian cases, [3] shows that there exists no deterministic relationship between the conditional entropy $H(Y|X)$ and the MSE or MAE when the estimation error follows a Student distribution. This is due to the fact that knowing the value of the conditional entropy only fixes one degree of freedom, whereas the Student distribution has two parameters and therefore two degrees of freedom. This is illustrated by Figure 2.11 where different MSE and MAE values can be achieved for a given conditional entropy, if the ν parameter takes different values.

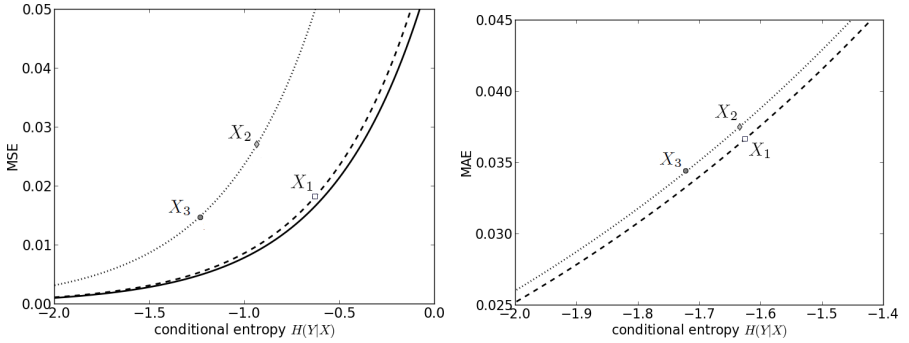
In the Student case, Figure 2.12 shows an example of MI failure with respect to the MSE and MAE criteria. In Figures 2.12(a) and 2.12(b), the feature subset X_2 corresponds to a smaller conditional entropy $H(Y|X_2)$ than the feature subset X_1. However, the MSE and MAE are larger for X_2 than for X_1. Here, the MI fails to select the best feature subset. In the case of MAE, Figure 2.12(b) shows that the impact of the failure is not important, since the curves for different degrees of freedom are quite close. Also, the situation in Figure 2.12(a) is quite extreme, since $\nu = 2.3$ is a very small degree of freedom which is unlikely to occur, unless there is a large number of outliers in the estimation error distribution.

In conclusion, whether MI is adequate for feature selection depends on the characteristics of the estimation error. Moreover, the impact of MI failures



(a) (b)

Figure 2.11: MSE and 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 Gaussian and Laplacian curves define an admissible region (in white). Reprinted with permission from [3].



(a) (b)

Figure 2.12: Example of mutual information failure for Student estimation error with respect to the MSE. The candidate feature subsets correspond to different numbers of degrees of freedom: $\nu = 2.3$ (dotted line) and $\nu = 5$ (dashed line). The curve for $\nu = 30$ (plain line) is also shown for discussion. The symbols X_i refer to feature subsets. Reprinted with permission from [3].

seems to remain limited. Hence, in practice, MI remains a valuable criterion for feature selection in regression.

2.8 Conclusion

This chapter shows that mutual information is not necessarily optimal for feature selection in either classification or regression. Indeed, it is possible in each context to design examples where MI fails to select the best feature subset, with respect to the accuracy in classification and to the MSE or MAE in regression. However, the impact and the frequency of MI failures seem to be limited in practice, as shown in the successive works [1–3, 11, 73].

In classification, the results obtained in [1, 2] show that most of the failures occur for intermediate MI values, like e.g. in the first steps of a forward search. Such problems can be avoided by using e.g. backward search or simple heuristics (see Section 2.5.5). Notice that datasets with thousands of features were not used in the experiments performed in [2]. However, linear models are usually preferred for such problems and mutual information may not be necessary. In regression, whether MI is adequate for feature selection depends on the characteristics of the estimation error (see Section 2.7). For uniform, Laplacian and Gaussian estimation error, mutual information can be used safely.

In conclusion, according to the results obtained in our works [1–3, 11, 73], MI remains a valuable criterion for feature selection. This is confirmed in our view by the large number of successful application of MI like e.g. [11, 40–49]. An interesting question is how much the mutual information failures studied in this chapter would be less likely if one considers pairs or triplets of features at the initial step of the forward search.

Chapter 3

Dealing with Label Noise

> The only real mistake is the one from which we learn nothing.
>
> *John Powell, composer*

Contents

The labels which come with training instances are not always reliable. Indeed, experts make mistakes, communication problems happen and categorisation may be subjective. For all these reasons, label noise may pollute datasets and it is necessary to handle mislabelling in classification problems. This chapter studies label noise and proposes several solutions to reduce the impact of mislabelled instances. First, Section 3.1 defines label noise and reviews its sources and consequences. Then, Section 3.2 reviews existing approaches to handle label noise. In particular, the probabilistic solution of Lawrence et al. [74] is detailed, since it is used in Section 3.3 and Section 3.4 to robustify hidden Markov models inference and mutual information estimation, respectively. Since the negative effects of label noise are mainly due to the occurrence of data which are abnormally frequent (like e.g. outliers), a more general approach is proposed in Section 3.5 to handle abnormally frequent data. The pointwise probability reinforcements allow to robustify any maximum likelihood-based method like e.g. linear regression, kernel ridge regression, logistic regression and principal component analysis. Eventually, Section 3.6 concludes the chapter.

3.1 About Label Noise

Label noise is defined in [75] as anything that obscures the relationship between the features of an instance and its class. This chapter is based on [4] and focuses on mislabelling, which randomly alters the observed labels such that they do not necessarily correspond to true classes. Intentional [76–79] and adversarial [80–89] label noise is not considered. Label noise is often contrasted with feature noise [90–92] which alters the observed values of features, but several works have shown that the label noise is more harmful [90, 92, 93].

3.1.1 Sources and Taxonomy of Label Noise

Mislabelling may be caused by different issues. First, available information may be insufficient to perform reliable labelling [75, 94], for example if the values of some features are unknown [95], if the description language is too limited [96] or if data are of poor quality [97]. Second, experts often make mistakes during labelling [75]. Labelling is a time-consuming and costly task and there is an increasing interest in using cheap, easy-to-get labels from non-expert [98–101]: the wealth of such labels may alleviate quality problems [98]. Third, classification is in some cases subjective [102–104], which results in inter-expert variability [105]. For example, the boundaries provided by experts for

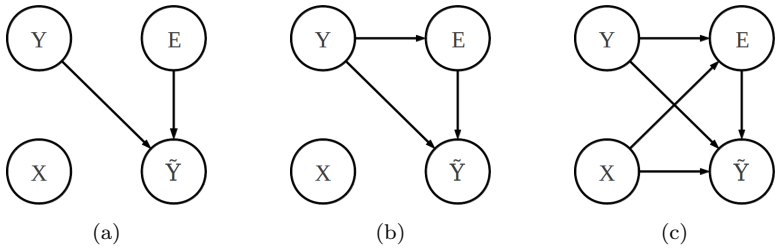


Figure 3.1: Statistical taxonomy of label noise inspired by [112]: (a) noisy completely at random (NCAR), (b) noisy at random (NAR) and (c) noisy not at random (NNAR). Arrows report statistical dependencies between the observed features X, the true class Y, the observed label $\tilde{Y}$ and the random variable E indicating whether an error occurred. The complexity of dependencies in the models increases from left to right. The statistical link between X and Y is not shown for clarity. Reprinted with permission from [4] © 2013 IEEE.

electrocardiogram signals (ECGs) are often slightly different from one expert to the other [106], because there exists no formal definition of the patterns of interest in ECGs. Eventually, incorrect labels may come from communication or encoding problems [92, 94, 107, 108]; real-word databases are estimated to contain around five percent of encoding errors [109–111].

In [4], a new taxonomy of label noise is proposed, inspired by the work of Schafer and Graham [112]. Figure 3.1 shows graphical models for the three types of noise which are distinguished here. First, noisy completely at random (NCAR) label noise occurs independently of the true class and of the values of the instance features. Second, changes which occur with noisy at random (NAR) label noise depend on the true label, which can be used to model situations where some classes are more likely to be mislabelled than others. Third, noisy not at random (NNAR) label noise is the more general case, where the mislabelling probability also depends on the feature values. This allows modelling labelling problems which occur near classification boundaries.

3.1.2 Consequences of Label Noise

Label noise is ubiquitous in real-word datasets, which has several consequences.

First, label noise decreases the prediction performances, which has been theoretically proved for simple models like linear classifiers [113–117], quadratic classifiers [118] or kNN classifiers [119–121]. Many works [90, 95, 108, 122–124] have empirically confirmed this issue for other classifiers like decision trees

induced by C4.5 and support vector machines, as well as in spam filtering. Boosting is also well known to be affected by label noise [125–128]. In particular, the adaptive boosting algorithm AdaBoost tends to give too large weights to mislabelled instances [126, 129, 130]. Most studies only deal with NCAR or NAR label noise, but NNAR label noise is also studied e.g. in [131, 132].

Second, the number of necessary training instances [107, 133–135] may increase in learning situations with label noise. This is also the case for the complexity of inferred models, like e.g. the number of nodes of decision trees [90, 94, 130] and the number of support vectors in SVMs [94, 136].

Third, the observed frequencies of the possible classes may be altered [97, 137–141], which is of particular importance in medical contexts. Indeed, medical studies are often concerned about measuring the incidence of a given disease in a population, whose estimation may be biased by label noise. This is also important in model validation, since performances measures can be poorly estimated in presence of label noise [142]. For example, a spam filter *with a true error rate of* 0.5%*, for example, might be estimated to have an error rate between* 5.5% *and* 6.5% *when evaluated using labels with an error rate of* 6.0%*, depending on the correlation between filter and label errors* [143].

Eventually, other related tasks like feature selection [144, 145] or feature ranking [146] are also impacted by label noise. A label noise-tolerant method for feature selection is proposed to address this problem in Section 3.4.

3.2 State of the Art to Deal with Label Noise

In light of Section 3.1.2, it seems necessary to deal with label noise. There exist three types of approaches in the literature [95,147–153]: label noise-robust models, data cleansing methods and label noise-tolerant learning algorithms.

3.2.1 Label Noise-Robust Models

From a theoretical point of view, learning algorithms are seldom completely robust to label noise [152], except in some simple cases [154–156]. However, in practice, some of them are more robust than the others; see e.g. [157–159] for empirical comparisons. For ensemble methods, bagging achieves better results than boosting [126] and several boosting methods are known to be more robust than AdaBoost [127, 128, 160–165]. For decision trees, the choice of the node split criterion can improve label noise-robustness [130, 166, 167]. In general, robust methods rely on overfitting avoidance to handle label noise [147–149].

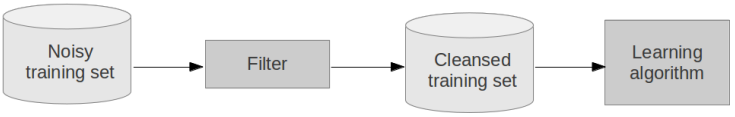


Figure 3.2: General procedure for learning in the presence of label noise with training set cleansing, inspired by [94]. Reprinted with permission from [4] © 2013 IEEE..

3.2.2 Data Cleansing Methods

A simple method to deal with label noise consists in removing instances which seems to be mislabelled, as illustrated in Figure 3.2. There exists a lot of such cleansing methods in the label noise literature. Similarly to outlier detection [168–171] and anomaly detection [172–176], one can e.g. simply use methods based on ad hoc measures of anomaly and remove instances which are above a given threshold [177]. One can also remove instances which disproportionately increase model complexity [178–182].

Model predictions can also be used to filter instances [181, 183]. A simple heuristic consists in removing training instances which are misclassified by a classifier [184–186], but this may remove too many instances [187,188]. Iterative [189] and local model-based [190–192] variants have been proposed, as well as voting filtering. With voting filtering [94, 181, 183, 193–198], an instance is removed when all (or almost all) learners in an ensemble agree to remove it, which can be adapted for large and distributed datasets [199–201].

There exists many other filtering methods. For example, instances which have an abnormally large influence on learning [104, 202, 203] or which seems suspicious [116] can be removed. A lot of kNN-based methods have also been proposed (see e.g. [121, 136, 204–206] for surveys and comparisons), which are mainly based on heuristics [121, 204, 207–215]. For example, the reduced nearest neighbours [208] removes instances whose removal does not cause other instances to be misclassified. Also, since AdaBoost tends to give large weights to mislabelled instances, several approaches use this unwelcome behaviour to detect label noise [197, 216–218]. For example, Verbaeten et al. [197] remove a given percentage of the instances with the largest weighting coefficients.

Eventually, an interesting approach proposed by Hughes et al. [106] consists in deleting the label of the instances (and not the instances themselves) for which experts are less reliable. Thereafter, semi-supervised learning is performed using both the labelled and the (newly) unlabelled instances. Surprisingly, this method has only been used in ECG segmentation; an open research

question is whether it could be applied to other settings.

3.2.3 Label Noise-Tolerant Learning Algorithms

In the probabilistic community, some authors claim that detecting label noise is impossible without making assumptions [139, 219–222]. For example, [139] reports a case where there exists an infinite number of maximum likelihood solutions for a probabilistic model taking label noise into account. In fact, for such identifiability issues [219, 221, 222], prior information is necessary to break ties. For example, Bayesian priors on the mislabelling probabilities [220, 223–226] can be used, but they should be chosen carefully, for *the results obtained depend on the quality of the prior distribution* [227, 228]. Beta priors [144, 220, 223, 224, 229–233] and Dirichlet priors [234, 235] are common choices; Bayesian methods exist for logistic regression [145, 233, 236–238], hidden Markov models [239] and graphical models [240]. Other approaches [232, 241, 242] are based on indicators which tells whether a given label has been flipped.

Frequentist methods also exist to deal with label noise. A simple solution consists in using a mixture of a normal distribution and an anomalous distribution [170, 243, 244]. The anomalous distribution is usually simply a uniform distribution on the instance domain, but other choices are possible. Besides, clustering can be used to detect mislabelled instances [151, 245, 246]. Indeed, instances whose label is not consistent with the label of nearby clusters are likely to be mislabelled. Eventually, belief functions can be used [247–249], since they allow modelling the confidence of the expert in its labels. When this information is not provided by the expert, several approaches have been proposed to infer beliefs directly from data [247–250]. Moreover, many machine learning methods have been adapted to deal with beliefs [247, 248, 251–254].

Several other non-probabilistic models have been modified to become label noise-tolerant. For example, one can prevent instances to take too large weights in neural networks [255–259], support vector machines [260–265] and ensembles obtained with boosting [266–269]. Robust losses [270–275] can also be used, which are theoretically less sensitive to outliers.

3.2.4 Probabilistic Modelling of Lawrence et al.

Lawrence et al. [74] have proposed a probabilistic modelling to deal with label noise, which is extended in [5, 276–278]. Since this methodology is extensively used in this thesis, it is detailed separately in this section.

In [74], labels are assumed to be generated according to the model shown in Figure 3.3. First, the true labels Y are drawn from a prior distribution p_Y.

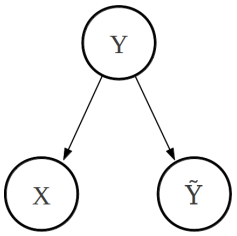


Figure 3.3: Statistical model of label noise, inspired by [74] and reprinted with permission from [4] © 2013 IEEE..

Then, the feature values are drawn from the conditional distribution $p_{X|Y}$ and the observed labels $\tilde{Y}$ are drawn from the conditional distribution $p_{\tilde{Y}|Y}$. The feature values and the observed labels are known, whereas the (hidden) true labels have to be estimated from data. Lawrence et al. [74] propose an EM algorithm to learn a Fisher discriminant while inferring the true labels.

The approach of Lawrence et al. [74] allows making probabilistic models label noise-tolerant. For example, it has been extended to non-Gaussian conditional class distributions [276], multi-class problems [277], logistic regression [278], sequential data [5] and mutual information estimation [3]. Section 3.3 and Section 3.4 are dedicated to the two last works [3, 5].

3.2.5 Experimental Considerations

Methods which deal with label noise must be assessed with datasets where mislabelled are identified and with relevant criteria.

In practice, there exist only a few real-world datasets where mislabelled instances have been identified [104, 143, 241, 279, 280]. In most experiments, label noise is artificially introduced in datasets. Most often, NCAR label noise is introduced by picking instances at random and flipping their label [150]. Several works consider NAR label noise where asymmetric flipping strategies are used [75, 92, 124, 158, 159, 199], in order to simulate situations where some classes are more likely to be polluted than others. Eventually, a few works deal with NNAR label noise which is introduced in ambiguous regions [131,132,156]. Open research questions include how to obtain more real-world datasets where mislabelled instances are clearly identified and what the characteristics of real-world label noise are. In the literature, it not yet clear if and when NCAR, NAR or NNAR label noise is the most realistic.

Algorithms which deal with label noise must be assessed using objective

criteria. Such criteria include e.g. the classification accuracy [94, 180, 181, 189, 193, 194, 197, 199], the model complexity [94, 193, 197], the accuracy of the estimation of true frequencies from observed frequencies [137–139, 142] and the filter precision for data cleansing methods [94, 194, 195, 197, 199, 281].

3.3 Robust HMMs

Electrocardiogram signals (ECGs) measure the electrical activity of the heart. These time series are used by physicians to monitor patients and to diagnose various cardiac diseases. In particular, the duration of certain patterns and the time interval between specific events are intensively used. Because ECGs may last for hours and contain thousands of cycles (called *beats*), their analysis is a tedious and time-consuming task. Automated tools exist to help physicians to segment ECGs, but they are not robust to label noise. This section presents an extension of the approach described in Section 3.2.4 to handle label noise with such models. These results have been published in [5].

3.3.1 Electrocardiogram Segmentation

Figure 3.4 shows an ECG, where several patterns (or *waves*) are delimited: the P wave, the QRS complex and the T wave, which are separated by baselines. Typically, only a few beats are manually segmented by an expert and machine learning tools are used to annotate the rest of the signal. State-of-the-art methods [19, 20, 106, 282, 283] use hidden Markov models (HMMs) for this task, where the ECG is transformed into a multi-dimensional signal with a wavelet transform. This time-frequency analysis allows one to decompose a time series into various scales. In ECG modelling, the resulting multivariate measures are modelled using Gaussian mixtures models (GMMs).

Theoretically, transitions in ECGs occur as shown in Figure 3.5. However, mislabelling may occur during manual annotation; Figure 3.6 shows an example of mislabelling for a real-world ECG. As discussed in Section 3.1.1, such errors can be due to expert errors, the subjectivity of the task (usually, different experts have different implicit definitions of patterns) or encoding problems (e.g. if the expert misclicked or inaccurately reported his segmentation). It is therefore desirable to design label noise-tolerant algorithms for automated ECG segmentation, which is done in the rest of this section.

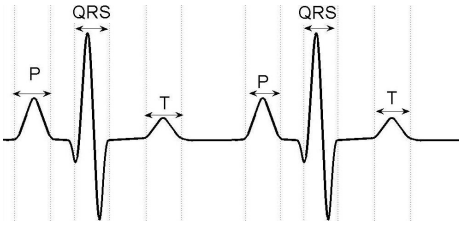


Figure 3.4: Example of ECG with annotations of the P wave, the QRS complex and the T wave. Reprinted with kind permission of Springer Science+Business Media from [5].

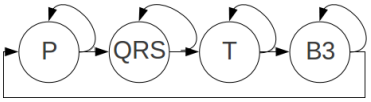


Figure 3.5: Theoretical transitions (arrows) in an ECG. B3 is the baseline between the T wave and the P wave. Reprinted with kind permission of Springer Science+Business Media from [5].

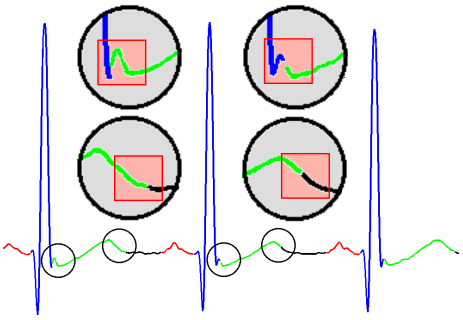


Figure 3.6: Example of real ECG with four enlargements of noisy labelling. From left to right: (i) the end of the QRS complex (blue) is incorrectly labelled as T wave (green), (ii) correct labelling, (iii) correct labelling and (iv) the end of the T wave (green) is incorrectly labelled as B3 baseline (black).

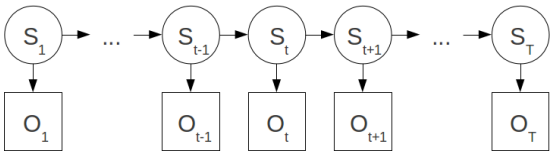


Figure 3.7: Conditional dependencies in a HMM. The label S_t is assumed to be a reliable estimate of the true heart state. Reprinted with kind permission of Springer Science+Business Media from [5].

3.3.2 Label Noise-Tolerant Inference of HMMs

Usually, the conditional dependencies in HMMs relate a sequence of inner states S_t and a sequence of multivariate observations O_t, as shown in Figure 3.7. In the line of Lawrence et al. [74], a more realistic model [5] is shown in Figure 3.8, where S_t is the true state at time t, Y_t is the expert annotation at time t and O_t is the multi-dimensional representation of the observed signal at time t. Three distributions must be learnt: the prior distribution of the true states p_S, the conditional distribution of the annotations $p_{Y|S}$ and the conditional distribution of the observations $p_{O|S}$. For the annotation of new beats, only the distributions p_S and $p_{O|S}$ are necessary, since one is only interested in predicting the true labels S_t (and not the noisy expert labels Y_t).

The above model of label noise assumes that observed expert annotations are independent of each other, which may seem surprising, since actual annotations are linked in practice. However, preliminary experiments have shown that more complex models of mislabelling do not improve the results. On the one hand, in the case of ECG segmentation, the amount of available training data is small and seems to be insufficient to learn additional parameters. On the other hand, when dependencies are added between successive expert annotations, the inference algorithm tends to shift patterns. For example, the right part of the small P wave is sometime interpreted as an expert labelling error and the P wave is moved left. Hence, such modelling is not considered in [5].

In order to learn a HMM in a label noise-tolerant way, Frénay et al. [5] propose an expectation-maximisation (EM) algorithm [284]. Indeed, similar to the approach described in Section 3.2.4, an estimate of the parameters for the HMM and label noise model can be obtained by maximising the incomplete log-likelihood with respect to the parameters Θ

$$\log P(O, Y|\Theta) = \log \sum_S P(O, Y, S|\Theta), \tag{3.1}$$

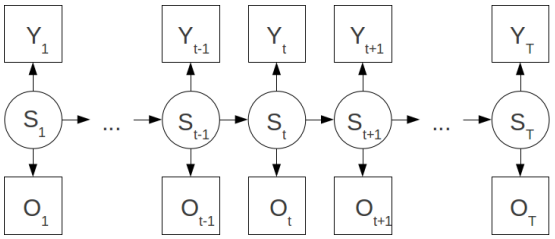


Figure 3.8: Conditional dependencies in a label noise-tolerant HMM. The label Y_t is a (noisy) copy of the heart state S_t. Reprinted with kind permission of Springer Science+Business Media from [5].

for which a closed-form solution does not exist. Instead, one can use an EM algorithm which maximises successive approximations of the incomplete log-likelihood (3.32) [285, 286]. It consists in alternatively (i) estimating

$$Q(\Theta, \Theta^{old}) = \sum_{S} P(S|O, Y, \Theta^{old}) \log P(O, Y, S|\Theta) \tag{3.2}$$

using the current estimate Θ^{old} (E step) and (ii) maximising $Q(\Theta, \Theta^{old})$ with respect to the parameters Θ in order to update their estimate (M step). Since

$$P(O, Y, S|\Theta) = q_{s_1} \prod_{t=2}^{T} a_{s_{t-1}s_t} \prod_{t=1}^{T} b_{s_t}(o_t) \prod_{t=1}^{T} d_{s_t y_t} \tag{3.3}$$

where q_s is the prior probability of state s, $a_{ss'}$ is the transition probability from state s to state s', $b_s(o)$ is the probability density of o in state s modelled with a Gaussian mixture model (GMM) and d_{sy} is the probability of observing label y when the true state is s expressed here as

$$d_{sy} = \begin{cases} 1 - p_e(s) & (s = y) \\ \frac{p_e(s)}{|\mathcal{S}|-1} & (s \neq y) \end{cases} \tag{3.4}$$

in terms of the expert error probability $p_e(s)$, the expression of $Q(\Theta, \Theta^{old})$ is

$$\sum_{s=1}^{|\mathcal{S}|} \gamma_1(s) \log q_s + \sum_{t=2}^{T} \sum_{s=1}^{|\mathcal{S}|} \sum_{s'=1}^{|\mathcal{S}|} \epsilon_t(s, s') \log a_{ss'} + \sum_{t=1}^{T} \sum_{s=1}^{|\mathcal{S}|} \gamma_t(s) \log b_s(o_t) + \sum_{t=1}^{T} \sum_{s=1}^{|\mathcal{S}|} \gamma_t(s) \log d_{sy_t} \tag{3.5}$$

where the posterior probabilities γ and ϵ are defined as

$$\gamma_t(s) = P(S_t = s|O, Y, \Theta^{old}) \tag{3.6}$$

and

$$\epsilon_t(s, s') = P(S_{t-1} = s, S_t = s'|O, Y, \Theta^{old}). \tag{3.7}$$

During the expectation (E) step, the γ and ϵ quantities are estimated using the current model, Equations (3.6) and (3.7) and a forward-backward algorithm [285,286] detailed in [5]. During the maximisation (M) step, the γ and ϵ values can be used to maximise $Q(\Theta, \Theta^{old})$ with respect to Θ: one obtains

$$q_s = \frac{\gamma_1(s)}{\sum_{s=1}^{|\mathcal{S}|} \gamma_1(s)} \tag{3.8}$$

and

$$a_{ss'} = \frac{\sum_{t=2}^{T} \epsilon_t(s, s')}{\sum_{t=2}^{T} \sum_{s'=1}^{|\mathcal{S}|} \epsilon_t(s, s')} \tag{3.9}$$

for the state prior and transition probabilities. For each state s, the distribution b_s is modelled with a GMM whose parameters are the priors π_{sl}, the means μ_{sl} and the covariance matrices Σ_{sl} for each lth Gaussian component. Their estimates are

$$\pi_{sl} = \frac{\sum_{t=1}^{T} \gamma_t(s, l)}{\sum_{t=1}^{T} \gamma_t(s)}, \tag{3.10}$$

$$\mu_{sl} = \frac{\sum_{t=1}^{T} \gamma_t(s, l) o_t}{\sum_{t=1}^{T} \gamma_t(s)} \tag{3.11}$$

and

$$\Sigma_{sl} = \frac{\sum_{t=1}^{T} \gamma_t(s, l)(o_t - \mu_{sl})^T (o_t - \mu_{sl})}{\sum_{t=1}^{T} \gamma_t(s)} \tag{3.12}$$

where

$$\gamma_{sl}(t) = \gamma_s(t) \frac{\pi_{sl} b_{sl}(o_t)}{b_s(o_t)}. \tag{3.13}$$

Eventually, the expert error probabilities are obtained with

$$p_e(s) = \frac{\sum_{t|Y_t \neq s} \gamma_t(s)}{\sum_{t=1}^{T} \gamma_t(s)}. \tag{3.14}$$

The above procedure is similar to the standard Baum-Welch algorithm [285,286] which is usually used to learn HMMs. The main difference is that there are two sequences of observed values: the observations O_t and the expert annotations

Y_t. Also, the part of the HMM which models expert annotations is not used at test time. Indeed, the annotations are not available and one is interested in the most probable sequence of states S_t for a given sequence of observations O_t. The fact that different models are used at training and test time is characteristic of probabilistic label noise-tolerant inference. Indeed, such methods use two models which render the classification task and the contamination by label noise, respectively. Only the classification model is useful at test time.

An important issue with EM algorithms is the initialisation of the model parameters which are updated during the M step. This problem is already addressed in the literature for all parameters, except d [286]. The probability $d_{sy} = P(Y_t = y|S_t = s, \Theta)$ is introduced for each pair of state and annotation (s, y) and is called the annotation probability. In other words, $d_{ss} = 1 - p_e(s)$ is the probability of correct annotation in state s. In [5], d is initialised with

$$d_{sy} = \begin{cases} 1 - p_e & (s = y) \\ \frac{p_e}{|\mathcal{S}|-1} & (s \neq y) \end{cases} \tag{3.15}$$

where p_e is a small probability of expert annotation error. In Section 3.3.3, the value $p_e = .05$ is used as an initial compromise between (i) $p_e = 0$ which corresponds to standard supervised HMMs and (ii) $p_e = (|\mathcal{S}| - 1)/|\mathcal{S}|$ which gives the unsupervised HMMs obtained with the Baum-Welch algorithm. Notice that the EM algorithm for unsupervised HMMs is initialised with expert annotations in Section 3.3.3, which may explain their surprisingly good results. Indeed, preliminary experiments shown that randomly initialised models are very slow to converge with the EM algorithm or converge to unsatisfying solutions because of the local minima of the incomplete log-likelihood.

3.3.3 Experimental Results on ECGs

The results obtained in [5] show that the proposed label noise-tolerant inference algorithm for HMMs yields better results than standard inference. For example, Figure 3.9 shows the experimental results obtained by adding a uniform (NCAR) label noise to the expert annotation for sinus ECGs from the MIT-QT database [287]. The results of two supervised and unsupervised learning procedures [19, 282, 283, 285] are compared to the proposed approach. The recall and precision is improved using label noise-tolerant inference when the amount of label noise increases. Similar results are obtained when a more realistic type of NNAR label noise is used (see Figure 3.10), where pattern boundaries are shifted by a small random number of milliseconds.

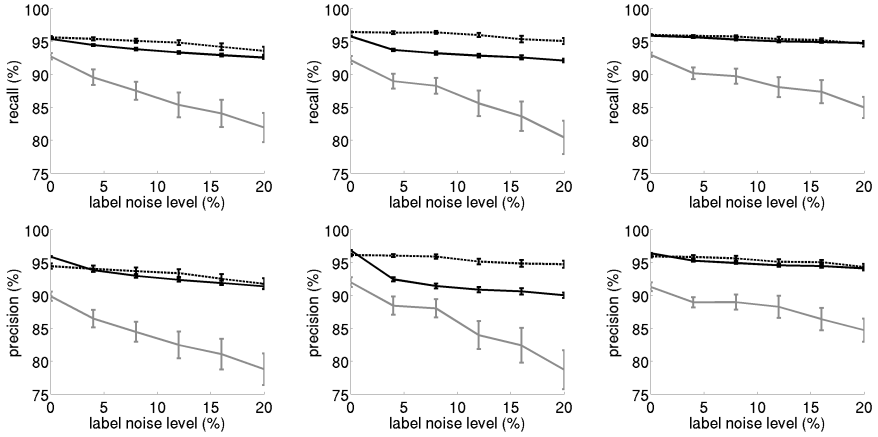


Figure 3.9: Recalls and precisions on sinus ECGs with uniform noise for supervised learning (black plain line), unsupervised learning (grey plain line) and the proposed algorithm (black dashed line), with respect to the percentage of flipped labels (0% to 20%). Reprinted with kind permission of Springer Science+Business Media from [5].

3.4 Feature Selection with Label Noise

As discussed in Section 3.1.2, decrease in model performances is not the only possible consequence of label noise. For example, the results of feature selection [144,145] and feature ranking [146] may be impacted. In the context of feature selection, this section shows that mutual information estimation is biased by label noise and proposes a solution to deal with label noise in such cases. The content of this section has been accepted in [3].

3.4.1 Label Noise and Mutual Information Estimation

As discussed in Section 2.2, mutual information (MI) $I(X;Y)$ measures the statistical relationship between a subset of feature X and a target Y [51]. MI can be used to select feature subsets which are relevant with respect to a classification task [42,50]. In practice, MI must estimated using e.g. the Kozachenko-Leonenko density estimator [54,55]. However, when labels are polluted by label noise, this estimator is biased by mislabelled instances as shown in Figure 3.11 where 40 instances are generated from two classes with Gaussian conditional probability distributions $\mathcal{N}(\mu_0 = -1.5, \sigma_0 = 1)$ and $\mathcal{N}(\mu_1 = 1.5, \sigma_1 = 1)$ and

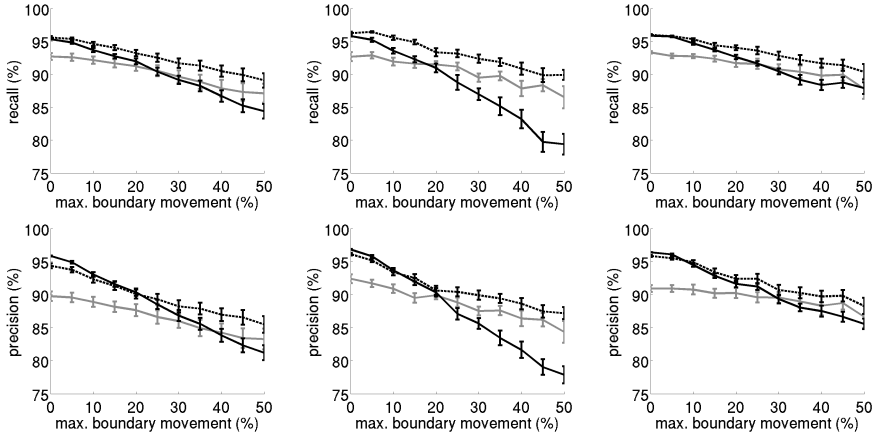


Figure 3.10: Recalls and precisions on sinus ECGs with horizontal noise for supervised learning (black plain line), unsupervised learning (grey plain line) and the proposed algorithm (black dashed line), with respect to the maximum boundary movement (0% to 50% of the modified wave). The actual boundary movement is randomly drawn. Reprinted with kind permission of Springer Science+Business Media from [5].

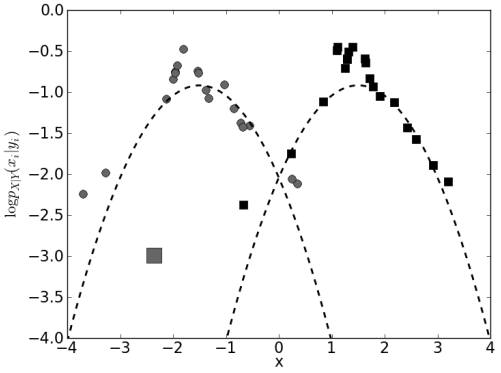


Figure 3.11: Kozachenko-Leonenko estimates of the logarithm of the conditional probability for a binary classification problem. Each class has a Gaussian distribution (dashed lines) and 40 samples are shown (grey circles belong to class 0, black squares belong to class 1); one sample is mislabelled (large grey square). Reprinted with permission from [3].

identical priors $p_Y(0) = p_Y(1) = \frac{1}{2}$. With the clean dataset, the MI estimate is $\hat{I}(X;Y) = 0.63$. When one of the instances is mislabelled, the MI estimate is only $\hat{I}(X;Y) = 0.58$. Hence, label noise affects mutual information estimation; [3] shows that this affects feature selection.

3.4.2 Label Noise-Tolerant MI Estimation

The model of Lawrence et al. [74] can be used to make the Kozachenko-Leonenko density estimator label-noise tolerant. This section describes an approach proposed in [3] and based on the concept of true class memberships.

True Class Memberships

In the presence of label noise, one can define the membership $p_{S|X,Y}(s|x,y)$ of an instance x to the true class s, if the observed label is y. In order to simplify mathematical notations, the following notation is introduced:

$$\gamma(s|i) = p_{S|X,Y}(s|x_i, y_i). \tag{3.16}$$

On the one hand, when there is no label noise, one has $\gamma(y_i|i) = 1$ and $\gamma(y|i) = 0$ for all $y \neq y_i$. On the other hand, in the presence of label noise, the memberships indicate which labels are likely to be the true, hidden labels.

Entropy Estimation with True Class Memberships

The Kozachenko-Leonenko density estimator is based on the hypothesis that $p_{X|Y}$ remains constant in a small hypersphere with diameter $\epsilon_k(i|y)$ containing exactly the k nearest neighbours of the ith sample in class y, which gives

$$\log \hat{p}_{X|Y}(x_i|y_i) = \psi(k) - \psi(n_y) - \log c_d - d \log \epsilon_k(i|y). \tag{3.17}$$

As shown in Section 2.3, this allows estimating the specific conditional entropy

$$\hat{H}(X|Y=y) = -\psi(k) + \psi(n_y) + \log c_d + \frac{d}{n_y} \sum_{i|y_i=y} \log \epsilon_k(i|y), \tag{3.18}$$

which can used to estimate MI with the estimator (2.11) of Gómez et al. [57] reviewed in Section 2.4. When some instances are mislabelled, two problems occur. First, the hypersphere diameter $\epsilon_k(i|y)$ for a given class y may be decrease (increase) because a neighbour of x_i is incorrectly labelled in (out of) class y. Second, instances which are incorrectly labelled in (out of) class y are (not) taken into account in (3.18), which can bias the resulting estimate.

In order to alleviate the two above problems, true class memberships can be used to obtain a reliable estimate of mutual information. In [3], it is proposed to use hyperspheres which contain an expected number of k instances really belonging to the target class s, i.e. such that the sum of their memberships to class s is approximately equal to k. These diameters can be computed using Algorithm 1. The resulting label noise-tolerant density estimator is

$$\log \hat{p}_{X|S}(x_i|s_i) = \psi\left(\Gamma(s|i)\right) - \psi\left(\Gamma(s)\right) - \log c_d - d \log \epsilon_{k,\gamma}(i|s) \tag{3.19}$$

where

$$\Gamma(s) = \sum_{i=1}^{n} \gamma(s|i) \tag{3.20}$$

is the expected number of samples which really belong to class s and

$$\Gamma(s|i) = \sum_{j=1}^{k(s|i)} \gamma(s|i_j) \tag{3.21}$$

is the expected of instances in class s in the $k(s|i)$ neighbours of x_i. For each instance, Algorithm 1 computes the number of neighbours $k(s|i)$ which have to be considered in order to obtain $\Gamma(s|i) \approx k$. Figure 3.12 shows the corrected density estimates for the example of Figure 3.11.

Algorithm 1 Label noise-tolerant estimation of hypersphere diameters for Kozachenko-Leonenko density estimation, reprinted with permission from [3].

Input: set of samples $\{x_i\}_{i\in 1\dots n}$ and memberships $\{\gamma(s|i)\}_{i\in 1\dots n, s\in\mathcal{Y}}$
Output: hypersphere diameters $\epsilon_{k,\gamma}(i|s)$ and memberships sums $\Gamma(s|i)$

for all class $s \in \mathcal{Y}$ **do**
  for all sample x_i **do**
    compute the ordering $i_1 \dots i_n$ of samples w.r.t. x_i

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

    while $\Gamma(s|i) < k$ **do**
      $k(s|i) \leftarrow k(s|i) + 1$
      $\Gamma(s|i) \leftarrow \Gamma(s|i) + \gamma(s|i_{k(s|i)})$
    end while

    $\epsilon_{k,\gamma}(i|s) \leftarrow 2\, \|x_{i_{k(s|i)}} - x_i\|_2$
  end for
end for

Since (i) each sample x_i belongs to class s with probability $\gamma(s|i)$ and (ii) $\Gamma(s)$ is the estimated number of samples in class s, Frénay et al. [3] define

$$\hat{H}(X|S=s) = -\frac{1}{\Gamma(s)} \sum_{i=1}^{n} \gamma(s|i) \log \hat{p}_{X|S}(x_i|s_i). \tag{3.22}$$

and eventually obtain the label noise-tolerant estimate

$$\begin{aligned} \hat{H}(X|S=s) = &-\frac{1}{\Gamma(s)} \sum_{i=1}^{n} \gamma(s|i) \psi\left(\Gamma(s|i)\right) + \psi\left(\Gamma(s)\right) \\ &+ \log c_d + \frac{d}{\Gamma(s)} \sum_{i=1}^{n} \gamma(s|i) \log \epsilon_{k,\gamma}(i|s). \end{aligned} \tag{3.23}$$

Mutual Information Estimation

Using the above results and the fact that the estimation of $H(X)$ is not affected by label noise, the MI estimator (2.11) of Gómez et al. [57] becomes

$$\hat{I}(X;S) = \hat{H}(X) - \sum_{s\in\mathcal{S}} \hat{p}_S(s) \hat{H}(X|S=s) \tag{3.24}$$

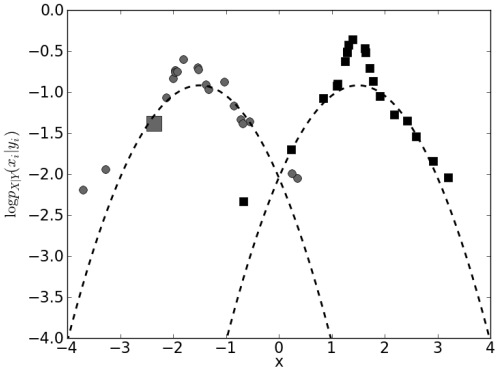


Figure 3.12: Label noise-tolerant estimates of the logarithm of the conditional probability for a binary classification problem. Each class has a Gaussian distribution (dashed lines) and 40 samples are shown (grey circles belong to class 0, black squares belong to class 1), but one sample is mislabelled (large grey square) like in Figure 3.11. Reprinted with permission from [3].

where $\hat{p}_S(s) = \Gamma(s)/n$ and $\hat{H}(X|S = s$ is given by Equation (3.23). Notice that Equation (3.24) gives the mutual information with respect to the true labels.

3.4.3 True Class Memberships Estimation

In order to use the label noise-tolerant MI estimator (3.24), one has to obtain the true class memberships $\gamma(s|i)$. This section proposes an expectation-maximisation (EM) algorithm for this task, which is based on the Lawrence et al. model of label noise with an error probability for each state $p_e(s)$:

$$p_{Y|S}(y|s) = \begin{cases} 1 - p_e(s) & \text{if } s = y \\ \frac{p_e(s)}{|\mathcal{Y}|-1} & \text{if } s \neq y. \end{cases} \tag{3.25}$$

Since label noise tends to decrease MI estimates, Frénay et al. [3] propose to conversely choose the error probabilities which maximise the estimated MI between the features X and the observed labels Y, i.e.

$$\hat{p}_e = \arg\max_{p_e} \hat{I}(X;Y). \tag{3.26}$$

One can show that this is in fact equivalent to the standard likelihood maximisation problem

$$\hat{p}_e = \arg\max_{p_e} \sum_{i=1}^{n} \log \hat{p}_{X|Y}(x_i|y_i). \tag{3.27}$$

In other words, Equation (3.26) corresponds to searching for the mislabelling model which provides the best explanation for the observed labels. There exists no close form solution for the above optimisation problem, since

$$\sum_{i=1}^{n} \log \hat{p}_{X|Y}(x_i|y_i) = \sum_{i=1}^{n} \log \sum_{s\in\mathcal{Y}} \hat{p}_{X,S|Y}(x_i, s|y_i). \tag{3.28}$$

However, a classical solution to deal with such problems is the EM algorithm [285, 286] which is also used in Section 3.3.2. The E step consists in estimating the true class memberships using the current label noise model, i.e.

$$\gamma(s|i) = \frac{p_{X|S}(x_i|s)p_{Y|S}(y_i|s)p_S(s)}{\sum_{s\in\mathcal{S}} p_{X|S}(x_i|s)p_{Y|S}(y_i|s)p_S(s)}. \tag{3.29}$$

During the M step, the error probabilities are updated with

$$p_e(s) = \frac{1}{\Gamma(s)} \sum_{i|y_i\neq s} \gamma(s|i) \tag{3.30}$$

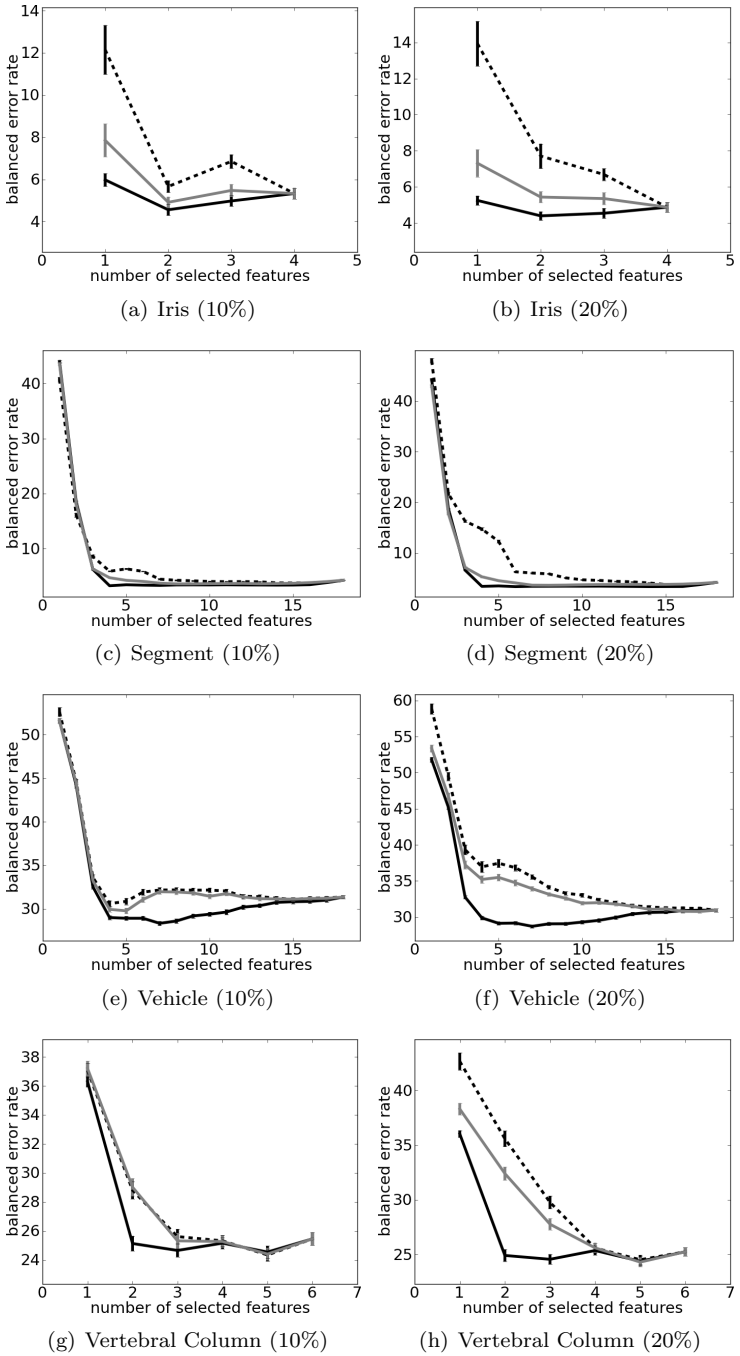
whereas the true class priors become

$$p_S(s) = \frac{1}{n} \sum_{i=1}^{n} \gamma(s|i). \tag{3.31}$$

Since the memberships stabilise after a few iterations of the EM algorithm, the proposed method is only a few times slower than the standard approach.

3.4.4 Experimental Results

Frénay et al [3] show the interest of dealing with label noise in feature selection for several UCI datasets [67]. Feature selection is performed using a backward search algorithm (i) with standard MI estimated on the clean labels (BW-C), (ii) with standard MI estimated on the noisy labels (BW-N) and (iii) with the proposed label noise-tolerant MI estimated on the noisy labels (LNT-BW). Experiments are repeated 100 times. For each feature subset, the training set with clean labels is also used to obtain a kNN classifier and the balanced error rate is computed in order to assess the relevance of the selected features. Figure



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

(c) Segment (10%) (d) Segment (20%)

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

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

Figure 3.13: Experimental results of feature selection for the (a-b) Iris, (c-d) Segment, (e-f) Vehicle and (g-h) Vertebral Column datasets. Balanced error rates in percents are shown in terms of the feature subset size for 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% of flipped labels in the left and right columns, respectively. Reprinted with permission from [3].

3.13 shows the results for four datasets when 10% and 20% of labels are flipped by artificial NCAR label noise (see [3] for more experimental results).

The results in [3] show the label noise can have a significant impact on feature selection: the performances are often significantly worst with BW-N than with BW-C. Also, in many cases, the label noise-tolerant method LNT-BW outperforms the label noise-sensitive approach BW-N. However, this is not always true and LNT-BW is most often outperformed by BW-C, which estimates MI using clean labels. In conclusion, feature selection with the proposed MI estimation algorithm is more label noise-tolerant than the standard method, but it is not yet completely insensitive to label noise.

3.5 Beyond Label Noise: Abnormally Frequent Data

When only a few training instances are available for model inference, it is particularly important to deal with abnormally frequent data (AFDs). Such instances are much more frequent in the training set that they should theoretically be. For example, when observations are drawn from a Gaussian distribution, AFDs include extreme values in the distribution tails. Indeed, such outliers should theoretically not be observed in a small sample. If appropriate measures are not taken to limit the influence of the outliers during Gaussian distribution fitting, the estimated mean and variance are likely to be biased.

The influence of AFDs is especially important in maximum likelihood methods, where the model parameters are obtained by maximising the log-likelihood

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}) = \sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta}). \tag{3.32}$$

Indeed, it can be shown that maximising the log-likelihood of training instances is equivalent to minimising the Kullback-Leibler divergence between the parametric distribution and the empirical distribution [288]. Since AFDs are characterised by too large empirical probabilities with respect to their theoretical probabilities, the learning process is biased towards models which support AFDs. This section proposes a generic method which has been submitted in [6] and which can be used to robustify any maximum likelihood learning algorithm. In particular, the proposed methodology can be used to deal with label noise.

3.5.1 Pointwise Probability Reinforcements

AFDs have a negative impact on likelihood maximisation because parametric models are unable to model them properly. A common solution in the literature consists in using a model where data are either coming from a normal distribution or from a garbage distribution [289, 290]. Data can be then be modelled using a mixture of these two distributions, what reduces the influence of AFDs which are identified as garbage patterns. However, this approach has two drawbacks. First, from a practical point of view, it is usually not trivial to design a relevant garbage distribution. Second, from a conceptual point of view, AFDs are not garbage patterns. There exist other solutions, but they are specific to e.g. label noise in classification or outliers in regression.

A more general solution consists in introducing pointwise probability reinforcements (PPRs). In [6], a small PPR $r_i > 0$ is associated to each instance i and the following reinforced likelihood is maximised:

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{r}) = \sum_{i=1}^{n} \log \left[p(x_i|\boldsymbol{\theta}) + r_i \right]. \tag{3.33}$$

PPRs r_i should be (close to) zero, except for AFDS for which they compensate for small probabilities $p(x_i|\boldsymbol{\theta})$. As shown in this section, maximising (3.33) can be done efficiently and is more robust than maximising (3.32).

3.5.2 Generic Reinforced Likelihood Maximisation

In [6], a generic two-step algorithm is proposed for reinforced likelihood maximisation. First, in order to avoid the trivial solution which consists in using large values for the PPRs, a regularisation term is added to control the amount of reinforcement. Assuming that the penalisation for a PPR is independent from other PPRs and from the model parameters, the new objective is

$$\mathcal{L}_\Omega(\boldsymbol{\theta}; \mathbf{x}, \mathbf{r}) = \sum_{i=1}^{n} \log \left[p(x_i|\boldsymbol{\theta}) + r_i \right] - \alpha \sum_{i=1}^{n} \Omega(r_i) \tag{3.34}$$

where Ω is a penalty function and α controls the amount of PPRS.

Pointwise Probability Reinforcements Optimisation

During the first step of the optimisation of the objective (3.34), the model parameters are kept fixed and only the PPRs are optimised. In [6], closed form expressions are obtained for the particular penalisations $\Omega(r_i) = r_i$ and

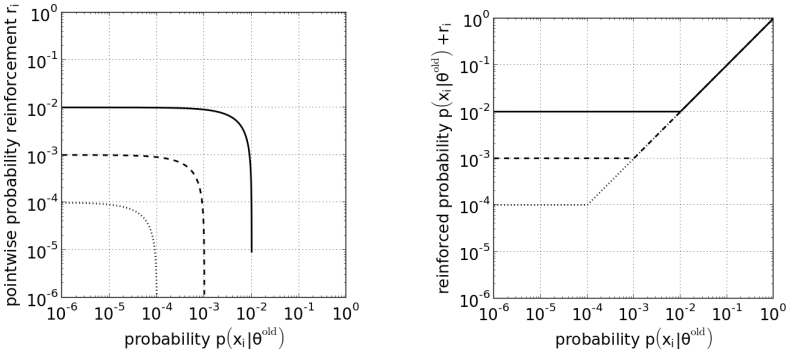


Figure 3.14: PPR r_i (left) and reinforced probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i$ (right) in terms of the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ obtained using L_1 regularisation on PPRs. The reinforcement meta-parameter is $\alpha = 10^2$ (plain line), $\alpha = 10^3$ (dashed line) and $\alpha = 10^4$ (dotted line). Reprinted with permission from [6].

$\Omega\left(r_i\right) = \frac{1}{2}r_i^2$. With L1 regularisation, the PPRs become

$$r_i = \max\left(\frac{1}{\alpha} - p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right), 0\right) \tag{3.35}$$

and the reinforced probabilities are

$$p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i = \max\left(p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right), \frac{1}{\alpha}\right). \tag{3.36}$$

Figure 3.14 shows these quantities in terms of the probability $p\left(x_i|\boldsymbol{\theta}\right)$; it can be seen that L1-regularised PPRs are sparse. Indeed, only PPRs associated with instances characterised by a very small probability are non-zero; the probability of normal instances is not reinforced. With L2 regularisation, the PPRs become

$$r_i = \frac{-p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + \sqrt{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)^2 + \frac{4}{\alpha}}}{2}, \tag{3.37}$$

and the reinforced probabilities are

$$p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i = \frac{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + \sqrt{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)^2 + \frac{4}{\alpha}}}{2}. \tag{3.38}$$

Figure 3.15 shows these quantities in terms of the probability $p\left(x_i|\boldsymbol{\theta}\right)$; it can be seen that L2-regularised PPRs are smoother than L1-regularised ones.

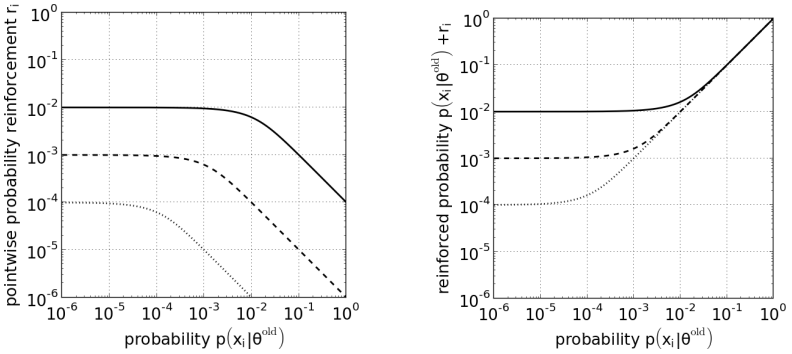


Figure 3.15: PPR r_i (left) and reinforced probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i$ (right) in terms of the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ obtained using L_2 regularisation on PPRs. The reinforcement meta-parameter is $\alpha = 10^4$ (plain line), $\alpha = 10^6$ (dashed line) and $\alpha = 10^8$ (dotted line). Reprinted with permission from [6].

Interestingly, Frénay and Verleysen [6] prove that, in the PPR framework, data which are more probable with respect to the parametric model are going to receive smaller reinforcements. Moreover, the ordering of observations with respect to the parametric model remains identical with reinforced probabilities.

Model Parameters Optimisation

During the second step, the PPRs are kept fixed and only the model parameters are optimised. In general, the exact maximisation of the reinforced likelihood is not trivial. Hopefully, Frénay and Verleysen [6] prove the following theorem:

Theorem 1 (reprinted with permission from [6]) *Let $\boldsymbol{\theta}^{old}$ be the current estimate of the model parameters and, for each observation x_i, let r_i be the optimal PPR with respect to $\boldsymbol{\theta}^{old}$. If one defines the observation weight*

$$w_i = \frac{p\left(x_i|\boldsymbol{\theta}^{old}\right)}{p\left(x_i|\boldsymbol{\theta}^{old}\right) + r_i}, \tag{3.39}$$

then the functional

$$\sum_{i=1}^{n} \left[w_i \log \frac{p\left(x_i|\boldsymbol{\theta}\right)}{p\left(x_i|\boldsymbol{\theta}^{old}\right)} + \log \left[p\left(x_i|\boldsymbol{\theta}^{old}\right) + r_i \right] \right] \tag{3.40}$$

is a lower bound to the reinforced log-likelihood

$$\sum_{i=1}^{n} \log \left[p\left(x_i|\boldsymbol{\theta}\right) + r_i \right]. \tag{3.41}$$

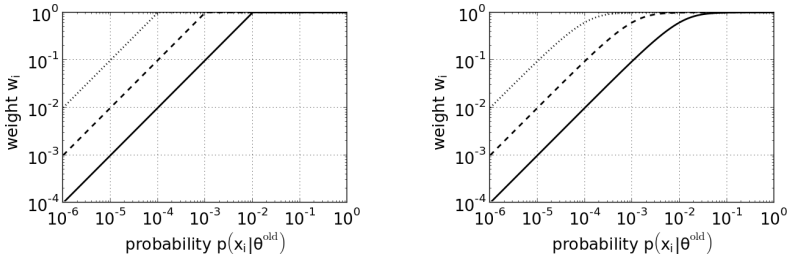


Figure 3.16: Observation weight w_i in terms of the probability $p(x_i|\theta)$ for L_1 (left) and L_2 (right) regularisation. The L_1 reinforcement meta-parameter is $\alpha = 10^2$ (plain line), $\alpha = 10^3$ (dashed line) and $\alpha = 10^4$ (dotted line). The L_2 reinforcement meta-parameter is $\alpha = 10^4$ (plain line), $\alpha = 10^6$ (dashed line) and $\alpha = 10^8$ (dotted line). Reprinted with permission from [6].

Moreover, (3.41) *and* (3.40) *are tangent at* $\boldsymbol{\theta} = \boldsymbol{\theta}^{old}$.

As shown in [6], Theorem 1 can be used to obtain a generic approximate parameters optimisation procedure. Indeed, if one ignores terms in (3.40) which do not depend on $\boldsymbol{\theta}$, it follows that the reinforced log-likelihood maximisation can be approximated by the maximisation of the weighted log-likelihood

$$\mathcal{L}_{\mathbf{w}}(\boldsymbol{\theta}; \mathbf{x}) = \sum_{i=1}^{n} w_i \log p(x_i|\boldsymbol{\theta}) \tag{3.42}$$

whose weights are given by definition (3.39). The weight w_i of each instance can be interpreted as normality degree: AFDs correspond to small weights. Figure 3.16 shows examples of weights obtained with L1 and L2 regularisation.

Literature Review and Links with the Proposed Methodology

There exists a large literature of methods to deal with outliers. For example, many approaches have been proposed to perform outlier detection [168–171] and anomaly detection [176]. Also, it is well known that many statistical inference methods are quite sensitive to outliers, like e.g. linear regression [169, 291, 292], logistic regression [293] and principal component analysis [294, 295].

The approach proposed in this section relies on weighted log-likelihood maximisation, which is often used in the literature to reduce the impact of outliers [296]. For example, there exist such algorithms for kernel ridge regression [297, 298], logistic regression [293] and principal component analysis [299, 300]. The main problem with these approaches is that the weights

are usually obtained through heuristics. Other methods for linear regression include e.g. M-estimators [301], the trimmed likelihood approach [302] and least trimmed squares [303, 304]. One of the main advantages of the proposed method is that the observation weights are automatically computed. Moreover, the method is very generic and can be applied to any inference problem which can be formulated as a likelihood maximisation. The goal is not only to detect the outliers: the aim is rather to make maximum likelihood estimates less sensitive to outliers. Linear regression, kernel ridge regression (a.k.a. least squares support vector machines), logistic regression and principal component analysis are shown in [6] to be easily robustified using the proposed approach.

AFDs have been studied in classification, where labelling errors adversely impact the performances of induced classifiers [92]. Methods have been proposed to limit the influence of each observation during inference, in order to prevent the model parameters to be biased by a few mislabelled instances. However, each method relies on a different way to limit the contribution of observations which is specific to a given model. For example, instances with large dual weights can be identified as mislabelled for support vector machines [260], mislabelled instances can be prevented to trigger updates too frequently for perceptrons [257] and instance weights can be limited in boosting [266]. It has also been proposed to associate each observation with a misclassification indicator variable [232], what is closer to the contribution of this section; the indicators can be used to identify mislabelled observations [144, 242]. The proposed approach has the advantage of being generic, simple to adapt to specific models and not limited to classification problems.

3.5.3 Supervised and Unsupervised Learning with PPRs

PPRs can be used to robustify any learning procedure which can expressed as a likelihood maximisation. In particular, PPRs are used in [6] to robustify several methods for both supervised and unsupervised learning. This section shows the results which are obtained in the case of kernel ridge regression and principal component analysis with the procedure given in Section 3.5.2.

Reinforced Kernel Ridge Regression

Kernel ridge regression [305] (a.k.a. least-square support vector machines [306]) is a ridge regression performed in a feature space. The predictions are

$$f(x_i) = \sum_{j=1}^{n} \alpha_j k(x_i, x_j) + \beta_0. \tag{3.43}$$

where $\boldsymbol{\alpha}$ is the vector of dual weights, β_0 is the bias and k is a kernel which computes dot products in the feature space [307]. The standard solution is obtained by assuming that the prediction errors $\epsilon_i = y_i - f(x_i)$ have a Gaussian distribution with variance σ_ϵ^2 and that the weights in the feature space have a Gaussian prior with variance σ_β^2. The parameters are the solution of the linear system

$$\begin{pmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & K + \frac{1}{\gamma}\mathbf{I}_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{y} \end{pmatrix} \tag{3.44}$$

where $\mathbf{K}$ is the Gram matrix such that $K_{ij} = k(x_i, x_j)$, $\mathbf{1}_n$ is a n-element vector of 1's and $\gamma = \sigma_\beta^2/\sigma_\epsilon^2$ is a meta-parameter. The standard deviation σ_ϵ can be estimated as

$$\sigma_\epsilon^2 = \frac{1}{n}\sum_{i=1}^{n} \epsilon_i^2. \tag{3.45}$$

With PPRs, the parameter optimisation is a weighted likelihood maximisation whose solution is given by the weighted kernel ridge regression [297, 298]

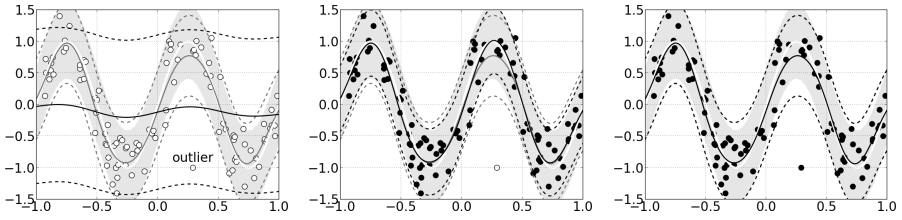
$$\begin{pmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & K + \frac{1}{\gamma}\mathbf{W}^{-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{y} \end{pmatrix} \tag{3.46}$$

where $\mathbf{W}$ is a diagonal weighting matrix whose diagonal terms are $W_{ii} = w_i$. The standard deviation σ_ϵ is obtained with the weighted average

$$\sigma_\epsilon^2 = \frac{\sum_{i=1}^{n} w_i \epsilon_i^2}{\sum_{i=1}^{n} w_i}. \tag{3.47}$$

The only difference between the solutions (3.44) and (3.46) is that the identity matrix $\mathbf{I}_n$ in (3.44) is replaced by the inverse of the weighting matrix $\mathbf{W}$ in (3.46). This allows reducing the impact of outliers on complexity control.

Figure 3.17 shows the results obtained for a non-linear regression problem with one outlier. With standard kernel ridge regression, predictions are biased by the outlier. Several values of the reinforcement parameter α are tested for L1- and L2-regularised PPRs. On the one hand, with small values of α, PPRs are free to take large values and the resulting model is completely irrelevant to the learning task. On the other hand, with large values of α, PPRs are constrained to remain close to zero and the resulting model is very close to the model obtained using standard kernel ridge regression. A compromise is obtained for an intermediate value of α, where the outlier is clearly identified.



(a) $\alpha = 10^{-3}$ (b) $\alpha = 20$ (c) $\alpha = 10^{10}$

(d) $\alpha = 10^{-6}$ (e) $\alpha = 40$ (f) $\alpha = 10^{20}$

Figure 3.17: Results obtained by standard kernel ridge regression (grey line) and reinforced kernel ridge regression (black line). The 95% percent confidence interval associated with the true function (white line) is shown by the grey-shaded area. PPRs are computed using L_1 (upper row) and L_2 (lower row) regularisation. The reinforcement meta-parameter values are indicated below each subfigure. The 30 data are shown by circles whose darkness is proportional to their respective weights. Dashed lines delimit 95% percent confidence intervals. In (c) and (f), standard and reinforced solutions are superimposed. Reprinted with permission from [6].

Reinforced Principal Component Analysis

Principal component analysis (PCA) can also be obtained as the solution of a likelihood maximisation. Indeed, Tipping et al [308] have proposed a probabilistic modelling of this unsupervised projection method, where q hidden, independent sources Z with standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_q)$ are linearly combined to obtain d observed features X with conditional distribution

$$p\left(X = \mathbf{x} | Z = \mathbf{z}, \boldsymbol{\mu}, \sigma\right) = \mathcal{N}\left(\mathbf{x} | \mathbf{A}z + \boldsymbol{\mu}, \sigma^2 \mathbf{I}_d\right) \tag{3.48}$$

where $\mathbf{A}$ is $d \times q$ linear transformation matrix, $\boldsymbol{\mu}$ is a d-dimensional translation vector and σ is the noise standard deviation. The marginal distribution of the observed features is

$$p\left(X = \mathbf{x} | \boldsymbol{\mu}, \sigma\right) = \mathcal{N}\left(\mathbf{x} | \boldsymbol{\mu}, \mathbf{C}\right) \tag{3.49}$$

where $\mathbf{C} = \mathbf{A}\mathbf{A}^T + \sigma^2 \mathbf{I}_d$. Hence, the log-likelihood becomes

$$\mathcal{L}\left(\mathbf{A}, \sigma; \mathbf{x}\right) = -\frac{n}{2}\left[d \log\left[2\pi\right] + \log\left|\mathbf{C}\right| + \operatorname{tr}\left(\mathbf{C}^{-1}\mathbf{S}\right)\right] \tag{3.50}$$

where the sample covariance matrix $\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T$ is obtained with the sample mean $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$. The maximum likelihood solution is

$$\mathbf{A}_{\mathrm{ML}} = \mathbf{U}_q\left(\boldsymbol{\Lambda}_q - \sigma^2 \mathbf{I}_q\right)^{\frac{1}{2}} \tag{3.51}$$

where $\mathbf{U}_q$ contains the q principal eigenvectors of $\mathbf{S}$ as columns and $\boldsymbol{\Lambda}_q$ is a diagonal matrix containing the q corresponding eigenvalues [308]. The maximum likelihood estimator of σ when $\mathbf{A} = \mathbf{A}_{\mathrm{ML}}$ is given for $d > q$ by

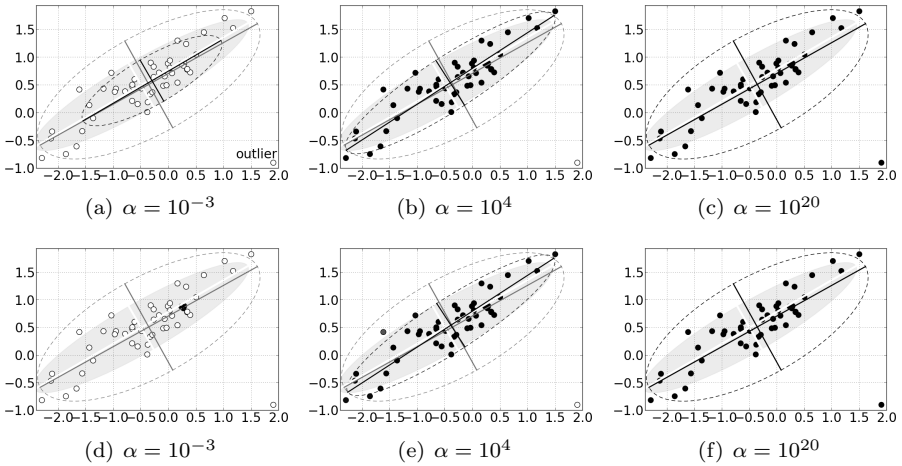
$$\sigma^2_{\mathrm{ML}} = \frac{1}{d-q}\sum_{i=q+1}^{d}\lambda_i. \tag{3.52}$$

PCA is known to be sensitive to outliers [294, 295]. With PPRs, the model parameter optimisation is a weighted PCA [299,300], whose solution is obtained by simply using Equation (3.51) with the weighted sample covariance matrix

$$\mathbf{S} = \frac{\sum_{i=1}^{n} w_i \left(\mathbf{x}_i - \boldsymbol{\mu}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}\right)^T}{\sum_{i=1}^{n} w_i} \tag{3.53}$$

where $\boldsymbol{\mu}$ is the weighted sample mean

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{n} w_i \mathbf{x}_i}{\sum_{i=1}^{n} w_i}. \tag{3.54}$$

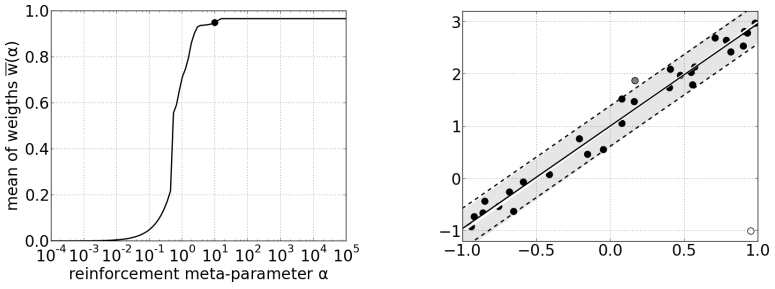


(a) $\alpha = 10^{-3}$ (b) $\alpha = 10^{4}$ (c) $\alpha = 10^{20}$

(d) $\alpha = 10^{-3}$ (e) $\alpha = 10^{4}$ (f) $\alpha = 10^{20}$

Figure 3.18: Axes obtained by standard PCA (grey lines) and reinforced PCA (black lines), with respect to the true axes of the hidden data model (white lines) for which the grey-shaded area shows the true 95% percent confidence region. PPRs are computed using L_1 (upper row) and L_2 (lower row) regularisation. The reinforcement meta-parameter values are indicated below each subfigure. The 50 data are shown by circles whose darkness is proportional to their respective weights. Dashed lines are level curves of the Gaussian distribution which delimit the 95% percent confidence region for each model, except in (d) where the reinforced PCA estimates an almost zero variance in the second principal axis direction. In (c) and (f), standard and reinforced solutions are superimposed. Reprinted with permission from [6].

Figure 3.18 shows results for a small Gaussian dataset with one outlier. The axes given by standard PCA are slightly rotated and stretched with respect to the real axes of the data cloud. Several values of the reinforcement parameter α are tested for L1- and L2-regularised PPRs. On the one hand, with small values of α, PPRs take large values and the resulting model only take a small part of the data into account. On the other hand, with large values of α, PPRs are constrained to remain close to zero and the axes are very close to those obtained with standard PCA. A compromise is obtained for an intermediate value of α, where the outlier is clearly identified and correct axes are obtained.



(a) (b)

Figure 3.19: Reinforcement meta-parameter selection for linear regression with L_1 regularisation on PPRs. The left panel shows the mean of weights in terms of α for optimal model parameters. The right panel shows the linear regression which is obtained with $\alpha = 10.19$ (black line), with respect to the true function (white line). The 30 data are shown by circles whose darkness is proportional to their respective weights. The estimated and true 95% percent confidence intervals are shown by dashed lines and the shaded region, respectively. Reprinted with permission from [6].

3.5.4 Selection of the Regularisation Meta-Parameter α

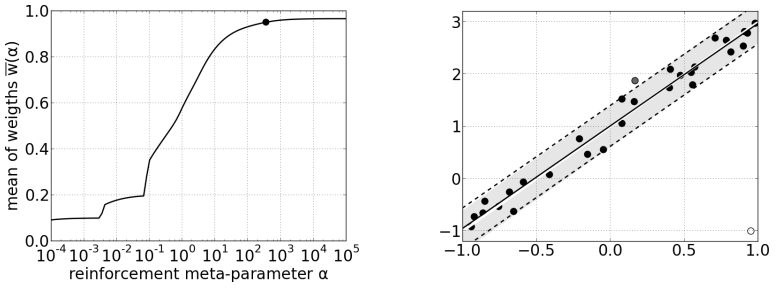
As shown in the two above examples, it is important to carefully select an appropriate value for the regularisation meta-parameter α. In [6], it is proposed to optimise α to obtain

$$\overline{w}(\alpha) = \frac{1}{n} \sum_{i=1}^{n} w_i \approx 0.95. \tag{3.55}$$

which is equivalent to assuming that 95% of the data are normal. Figures 3.19 and 3.20 show the results of the α optimisation for linear regression with the two simple procedures proposed in [6] for L1 and L2 regularisation, respectively.

3.6 Conclusion

This chapter reviews the literature on label noise and shows that this complex phenomenon has many potential sources: low-quality information, expert errors, subjective nature of classes and communication or encoding problems. The consequences of label noise are shown to be diverse, including decrease of the prediction performances, increase of the number of necessary training instances and alteration of observed frequencies. Hence, it is not surprising that



(a) (b)

Figure 3.20: Reinforcement meta-parameter selection for linear regression with L_2 regularisation on PPRs. The left panel shows the mean of weights in terms of α for optimal model parameters. The right panel shows the linear regression which is obtained with $\alpha = 323.08$ (black line), with respect to the true function (white line). The 30 data are shown by circles whose darkness is proportional to the sample weights. The estimated and true 95% percent confidence intervals are shown by dashed lines and the shaded region, respectively. Reprinted with permission from [6].

a lot of methods have been designed to deal with label noise, including label noise-robust models, data cleansing methods and label noise-tolerant learning algorithms. However, an open problem is the current difficulty to assess methods in the presence of label noise, since very few appropriate datasets exist.

Probabilistic modelling of label noise allows one to deal with mislabelled instances in various situations. For example, label noise-tolerant algorithms are designed in this chapter for sequential data modelled with HMMs and for feature selection with mutual information. These two algorithms are based on the probabilistic model of Lawrence et al. [74], which is also used in other works including [276–278]. We believe that probabilistic modelling has many desirable advantages over other approaches based e.g. on heuristics. First, they are generic and can applied to diverse learning methods, which has been done for the Lawrence et al. model. They are more theoretically sound than heuristics used e.g. in data cleansing methods, which allows performing theoretical analysis of their behaviour. Second, they allow one to derive probabilistic information about instances, like e.g. the probability that a given label has been switched. Third, prior information about the label noise can be easily embedded in such methods. Bayesian priors can be used, but the probabilistic model itself can be adapted to better model NCAR, NAR or NNAR label noise.

Many learning algorithms can be formulated in terms of likelihood maximisation, including e.g. linear regression, kernel ridge regression, logistic regression, principal component analysis, etc. For all these methods, this chapter proposes the generic PPR approach which allows dealing with abnormally frequent data. Even if PPRs are not probabilities, the PPR approach shares many advantages with the probabilistic modelling discussed in the above paragraph. First, this method is generic as it can be used to robustify any maximum likelihood-based algorithm. Second, the PPRs can be used to obtain weights, which characterise the degree of normality of the instances. This information can be used to analyse the dataset. Third, prior information about the AFDs can be embedded using the regularisation term. For example, L1 regularisation is suitable when training data are expected to contain only a few AFDs, whereas L2 regularisation is more suitable when many training instances are anomalous. We believe that PPRs provide an easy-to-implement and interesting solution to deal with outliers in many supervised and unsupervised contexts.

Chapter 4

Extreme Learning Machines

> Everything should be made as simple as possible, but not simpler.
>
> *Albert Einstein, physicist*

Contents

This chapter discusses extreme learning, which offers fast methods to learn neural models without the need for tuning meta-parameters. First, Section 4.1 explains why fast models may be necessary in real-world applications. Then, Section 4.2 reviews extreme learning machines (ELMs) and their basic algorithms. Section 4.3 discusses the behaviour of ELMs when their number of neurons becomes very large and proposes a new kernel based on ELMs. Section 4.4 extends this analysis to regression and proposes an analytical form for the ELM kernel when the number of neurons is infinite. Eventually, Section 4.6 concludes this chapter in the light of other recent results in extreme learning. The results presented in this chapter have been published in [9, 10]

4.1 The Need for Fast Models

In real-world applications, computational time is often a critical resource. For example, if an image processing algorithm is used for a real-time application, each classification has to be performed in a few milliseconds. In applications like diesel engine modelling [309], a large number of models may have to be learned in a small amount of time. In these contexts, fast models are necessary.

For most models, the most expensive step is the optimisation of the meta-parameters. Indeed, this optimisation is typically achieved by minimising a generalisation error estimated using e.g. cross-validation or bootstrap [43, 310], which requires the learning of a large number of models. Consequently, models which have many meta-parameters may become useless because the computational cost of their meta-parameters selection is too high. For example, although theoretically appealing, the state-of-the-art non-linear support vector regression is seldom used in practice because its three meta-parameters have to be tuned and tens of thousands of regression models have to be learnt.

In conclusion, it is worth of interest to obtain models with a small number of meta-parameters and which are fast to train and to use for prediction.

4.2 Extreme Learning

Extreme learning offers a solution to the two problems discussed in Section 4.1. This recent trend of machine learning proposes a compromise between learning speed and prediction performances, where non-linear models can be obtained at roughly the same computational cost than linear regression.

4.2.1 Single Layer Feedforward Neural Networks

Extreme learning [7, 8] proposes fast methods to learn single layer feedforward neural networks (SLFNs) [311]. As shown in Figure 4.1, a SLFN consists of three layers of neurons: (i) an input layer where each neuron corresponds to a dimension (or feature) in the data space, (ii) a hidden layer whose number of neurons determines the learning abilities of the SLFN and (iii) an output layer where each neuron corresponds to a predicted quantity. In this thesis, multi-output neural networks are not considered and the output layer of SLFNs always contains only one neuron. The output of the pth hidden neuron is

$$h_p(\mathbf{x}) = \sigma\left(\sum_{j=1}^{d} W_{jp} x_j + b_p\right) \tag{4.1}$$

where σ is the non-linear activation function of hidden neurons, W_{jp} is the weight between the jth neuron in the input layer and the pth hidden neuron and b_p is the bias of the pth hidden neuron. In extreme learning, σ is usually a sigmoid function like e.g. tanh [7] or erf [10, 312]. The output of the SLFN is

$$f(\mathbf{x}) = \sum_{p=1}^{m} w_p h_p(\mathbf{x}) \tag{4.2}$$

where m is the number of hidden neurons and w_p is the weight between the pth hidden neuron and the output neuron. Notice that the output neuron has no bias. The weights are called hidden weights and output weights, respectively.

Usually, the hidden and output weights of a SLFN are optimised using a gradient descent algorithm like e.g. the well-known backpropagation algorithm [311]. However, gradient descent algorithms are slow and their behaviour is controlled by parameters which are difficult to tune. Moreover, the error surface of a SLFN may contain several local minima and the gradient descent is therefore not guaranteed to converge to a global minimum [7].

4.2.2 Extreme Learning Machines

Huang et al. [7, 8, 313–316] introduce a new efficient way to optimise SLFNs, which are called extreme learning machines (ELMs) in their works. In order to solve the difficulties encountered by traditional SLFN learning algorithms, the hidden weights of ELMs are not optimised. Instead, Huang et al. [7, 313, 314] propose to draw these weights randomly from a given distribution, which is usually either uniform or Gaussian. During learning, the hidden weights remain

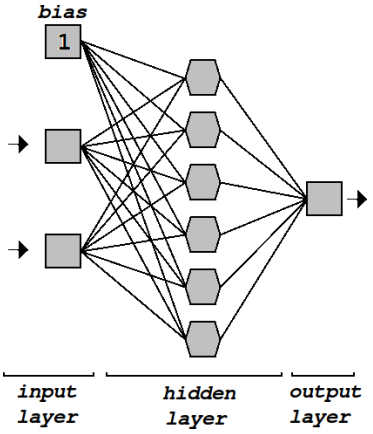


Figure 4.1: Architecture of a single layer feedforward neural network. Shapes are neurons and lines are connections. Reprinted with permission from [21].

fixed and only the output weights are optimised. The solution is

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{t} - \mathbf{H}\mathbf{w}\|_2^2 \tag{4.3}$$

for a training set $\{(\mathbf{x_i}, t_i)\}_{i=1}^n$, where $\mathbf{t}$ is the vector of target values and

$$\mathbf{H} = \begin{pmatrix} h_1(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_n) & \cdots & h_m(\mathbf{x}_n) \end{pmatrix} \tag{4.4}$$

is the hidden layer output matrix. Since Equation (4.3) is simply a linear regression, ELMs can be trained much faster than traditional non-linear methods. Several works [7,314] report that ELMs can be up to hundred times faster than backpropagation or support vector machines. Interestingly, ELMs achieve results which are close to those of state-of-the-art methods for both classification and regression [7,314,317–319]. Hence, extreme learning offers are a good compromise between computation needs and prediction accuracy. The idea of using random neurons is not new, as discussed in [320], even if Huang [321] claims that there are several differences between ELMs and previous works. According [7], the name *extreme learning* reflects the fact that ELMs run extremely fast.

4.2.3 Algorithms for Extreme Learning Machines

The ELM literature proposes several ways of learning ELMs [8]. The very first ELM implementation [7, 314] simply consists in using the pseudo-inverse method to solve Equation (4.3). ELMs can also be learned incrementally [315, 322, 323] by adding neurons one by one. Depending on the implementation, additional neurons may be chosen in a large pool of random neurons in order to obtain more relevant neurons (with respect to the learning task). Regularisation techniques have also been proposed, using e.g. L1 [324] and L2 [9, 325] regularisation. In the former case, neurons are ranked and only a subset of them is used, which is called OP-ELM. In the latter case, all neurons are used, but their weights are prevented to take very large values. TROP-ELM [326] merges these two approaches. Also, ELMs are linked to kernel machines [9, 10, 319, 325], as discussed in the rest of this chapter.

4.3 SVMs with Randomised Feature Spaces

This section discusses how extreme learning can be used with support vector machines for classification and is based on the results published in [9].

4.3.1 Support Vector Machines

Support vector machines (SVMs) [256, 327, 328] are state-of-the-art models for binary classification which maximise the margin between the two classes. SVMs are linear classifiers whose decision function is $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ where $\mathbf{w}$ is a weight vector and b is a bias. If some of the training instances are allowed to be misclassified during training, the parameters of a SVM are the solution of

$$\left| \begin{array}{lll} \min\limits_{\mathbf{w},b,\xi_i} \|\mathbf{w}\|_2^2 + C\sum_i \xi_i & & \\ \text{s.t.} \quad t_i(\mathbf{w}\cdot\mathbf{x}_i + b) & \geq & 1-\xi_i \\ \xi_i & \geq & 0 \end{array} \right. , \tag{4.5}$$

where C is a regularisation constant and ξ_i is the distance between $\mathbf{x}_i$ and the hyperplane corresponding to $\mathbf{w} \cdot \mathbf{x} + b = t_i$ if $\mathbf{x}_i$ is misclassified. The regularisation allows finding a compromise between maximising the margin (or equivalently minimising $\|\mathbf{w}\|_2^2$) and minimising the classification errors.

One of the keys of the success of SVMs is that they can easily be kernelised [256, 327, 328]. The basic idea of kernel methods [307] is to first map data in a high-dimensional space, called the feature space, and to learn in that representational space using regularisation. In the case of SVMs, a dual version

of (4.5) can be obtained where only the dot products between instances in the feature space are necessary. These dots products are given by a kernel $k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ which can usually be computed without actually computing the mapping ϕ. For example, a common kernel is the RBF kernel

$$k(\mathbf{x}, \mathbf{z}; \gamma) = e^{-\gamma \|\mathbf{x}-\mathbf{z}\|_2^2} \tag{4.6}$$

where γ is the kernel width and the feature space is actually infinite-dimensional.

4.3.2 The Extreme Learning Machines Kernel

The ELMs discussed in Section 4.1 can be interpreted in terms of kernels [9, 319, 325]. Indeed, an ELM can be used to define a mapping

$$\phi(\mathbf{x}) = \left(h_1(\mathbf{x})/\sqrt{m}, \ldots, h_m(\mathbf{x})/\sqrt{m}\right) \tag{4.7}$$

where $h_p(\mathbf{x})$ is the output of the pth hidden neuron. Here, each neuron defines a new dimension in a feature space whose dimensionality is equal to the number of neurons m. Solving Equation (4.3) can be seen as the solution of a linear regression in the feature space defined by the ELM. The corresponding kernel

$$k(\mathbf{x}, \mathbf{z}; m) = \frac{1}{m} \sum_{p=1}^{m} h_p(\mathbf{x}) h_p(\mathbf{z}) \tag{4.8}$$

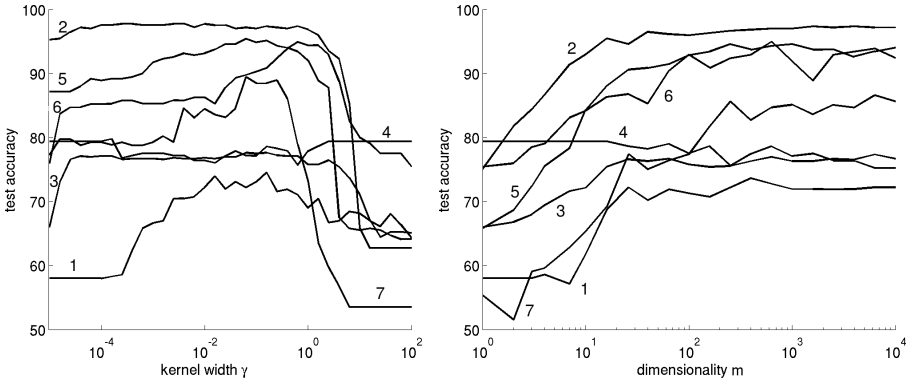
is called the ELM kernel in [9] and is studied in the rest of this chapter.

4.3.3 Support Vector Machines with the ELM Kernel

The ELM kernel is tested in [9] with SVMs. Figure 4.2 shows the accuracies obtained for UCI datasets [67, 329, 330] with (i) the RBF kernel for several kernel widths γ and (ii) the ELM kernel with several number of neurons m. One the one hand, the results with the RBF kernel are sensitive to γ, whose optimal value depends on the dataset. On the other hand, the results with the ELM kernel are optimal for large dimensionalities m. Consequently, whereas the kernel parameter γ has to be optimised, the number of neurons in the ELM kernel can be fixed to a large value without tuning. Table 4.1 shows the results obtained with $m = 1000$; the ELM kernel achieves results which are comparable to those of the RBF kernel in a much smaller amount of training time.

4.4 ELM Kernel for Support Vector Regression

Section 4.3 shows that large ELMs can be used to define a new kernel, which allows obtaining good SVM classifiers in a short amount of time. This section



(a) (b)

Figure 4.2: Test accuracy with (a) the kernel RBF with respect to the kernel width γ and (b) the ELM kernel with respect to the number of neurons m. The curve labels are given in Table 4.1. Adapted with permission from [9].

is based on the results published in [10] and shows that the ELM kernel admits an analytical form when the number of neurons is infinite. This asymptotic ELM kernel is shown to give good results in regression.

4.4.1 Support Vector Regression

Support vector regression (SVR) adapts SVMs for regression using the ϵ-sensitive loss, which only penalises prediction errors outside a tube and is given by

$$|y-t|_\epsilon = \begin{cases} 0 & \text{if} \quad |y-t| \leq \epsilon \\ |y-t|-\epsilon & \text{if} \quad |y-t| > \epsilon \end{cases} \tag{4.9}$$

where ϵ denotes the half-width of the tube, as illustrated in Figure 4.3(a). SVR finds a compromise between model complexity (quantified by $\|\mathbf{w}\|_2^2$) and large estimation errors. Its parameters are the solution of the problem

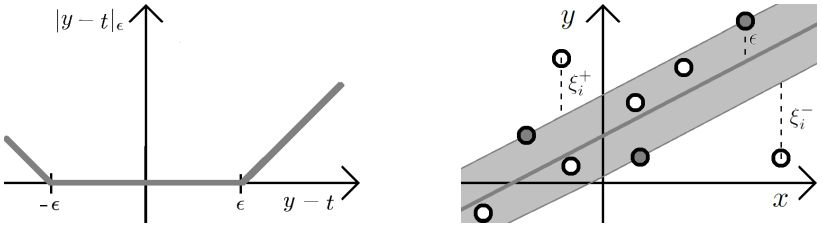
$$\left| \begin{array}{lrcl} \min\limits_{\mathbf{w},b,\xi_i^+,\xi_i^-} & \|\mathbf{w}\|_2^2 + C\sum_{i=1}^n (\xi_i^+ + \xi_i^-) & & \\ & \mathbf{w}\cdot\mathbf{x}_i + b - t_i & \leq & \epsilon + \xi_i^+ \\ \text{s.t.} & t_i - \mathbf{w}\cdot\mathbf{x}_i + b & \leq & \epsilon + \xi_i^- \\ & \xi_i^+, \xi_i^- & \geq & 0 \end{array} \right. , \tag{4.10}$$

where $\mathbf{w}$ is the weight vector, b is the bias, C is the regularisation constant and ξ_i^+, ξ_i^- are positive slack variables (see Figure 4.3(b)). SVR can easily be

		RBF kernel		ELM kernel ($m = 1000$)	
		acc.	comp. time	acc.	comp. time
1	Bupa	72 (69−76)	140 (139−141)	72 (64−80)	5 (5−5)
2	Cancer	97 (95−98)	292 (290−293)	97 (96−98)	3 (3−3)
3	Diabetes	76 (73−80)	594 (586−602)	76 (74−79)	27 (27−28)
4	Heart	77 (72−82)	100 (98−101)	77 (72−83)	1 (1−1)
5	Ion	95 (92−98)	169 (168−170)	95 (92−97)	1 (1−2)
6	Parkinsons	93 (91−96)	36 (35−36)	92 (88−96)	1 (1−1)
7	Sonar	89(83−95)	96 (95−96)	85 (83−88)	1 (1−1)

Table 4.1: Test accuracies and computational times for RBF and ELM kernels with SVMs for classification. Confidence intervals are obtained with 10-fold cross-validation. Computational times include meta-parameters selection and the learning of the final model. Adapted with permission from [9].

kernelised, but it is seldom used in practice due to its three meta-parameters: the tube half-width ϵ, the regularisation parameter C and the kernel parameter. Indeed, the meta-parameter selection requires to perform SVR tens of thousands of times, which is unaffordable in most applications.



(a) (b)

Figure 4.3: Reprinted with permission from [10]: (a) ϵ-sensitive loss and (b) example of linear SVR with two estimation errors lying outside the ϵ-tube.

4.4.2 The Asymptotic ELM Kernel

Frénay et al. [10] show that the ELM kernel defined in Section 4.3.2 admits an analytical form when the number of hidden neurons is infinite. Indeed, in that case, the ELM kernel is called the *asymptotic ELM kernel* and becomes

$$k(\mathbf{x}, \mathbf{z}; m \to \infty) = \lim_{m \to +\infty} k(\mathbf{x}, \mathbf{z}; m) = \lim_{m \to +\infty} \frac{1}{m} \sum_{p=1}^{m} h_p(\mathbf{x}) h_p(\mathbf{z}), \qquad (4.11)$$

which can be interpreted as the covariance between the activations of a random hidden unit alternatively fed with $\mathbf{x}$ and $\mathbf{z}$. If the weights and bias of the hidden neurons of the ELM are drawn from an isotropic Gaussian distribution with variance $\sigma_{\mathbf{w}}^2$ and their activation function is the sigmoid function

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt, \tag{4.12}$$

then one can obtain using [331] the following analytical expression

$$k(\mathbf{x}, \mathbf{z}; m \to \infty) = \frac{2}{\pi} \arcsin \frac{1 + \mathbf{x} \cdot \mathbf{z}}{\sqrt{\left(\frac{1}{2\sigma_{\mathbf{w}}^2} + 1 + \|\mathbf{x}\|_2^2\right)\left(\frac{1}{2\sigma_{\mathbf{w}}^2} + 1 + \|\mathbf{z}\|_2^2\right)}}, \tag{4.13}$$

which has independently been found by Frénay et al. [10] and Parviainen et al. [312]. Interestingly, Equation (4.13) only depends on dots products and $\sigma_{\mathbf{w}}^2$.

4.4.3 Support Vector Regression with the ELM Kernel

In [10], SVR is used with the normalised asymptotic ELM kernel

$$\tilde{k}(x, z; m \to \infty) = \frac{k(x, z; m \to \infty)}{\sqrt{k(x, x; m \to \infty) k(z, z; m \to \infty)}} \tag{4.14}$$

which can be interpreted as a correlation. Experiments are performed on several datasets of various sizes, where several values of σ_w are tested. If data are normalised before learning, it appears that the values $\sigma_w = 1$ and $\sigma_w = 10$ allow one to obtain results which are close if not identical to the results obtained using the RBF kernel. Hence, the proposed kernel is parameter-insensitive, in the sense that it is not necessary to tune the parameter σ_w to obtain good results. With the asymptotic ELM kernel, SVR has only two meta-parameters, which speeds up the meta-parameter optimisation step, so that non-linear problems can be tackled at the same computational cost than linear problems.

4.5 Impact of the Solver on Computational Times

When using kernel methods, the choice of the solver may have an important impact on the computational needs. This section compares LIBSVM [332] and LIBLINEAR [333] to perform classification with the ELM kernel. Data are normalised and 300 neurons with random weights and bias drawn from

a standard normal distribution $\mathcal{N}(0,1)$ are used to build the feature space. For both LIBSVM and LIBLINEAR, the 300 new features are explicitly used with a linear kernel, i.e. the hidden layer output matrix $\mathbf{H}$ is used instead of the design matrix $\mathbf{X}$. LIBSVM is additionally tested with the precomputed Gram matrix $\mathbf{K} = \mathbf{H}\mathbf{H}^T$. Table 4.2 shows the results obtained with the three above settings. The accuracies are obtained using double cross-validation and the computational times include the meta-parameter selection. Notice that only the computational time used by the solvers are shown. For example, the computation of the Gram matrix $\mathbf{K}$ is not taken into account. This allows a fair comparison of the different uses of LIBSVM and LIBLINEAR.

		LIBSVM with expl. feat. space		LIBSVM with precomp. Gram matrix		LIBLINEAR with expl. feat. space	
name	size	acc.	time	acc.	time	acc.	time
Parkinsons	195×22	79.9 (±9.0)	13	79.9 (±9.0)	2	77.3 (±11.9)	57
Bupa	345×6	70.5 (±5.7)	56	70.5 (±5.7)	9	71.6 (±6.4)	88
Ionosphere	351×34	93.2 (±3.6)	28	93.2 (±3.6)	4	91.5 (±4.0)	33
Vehicle	390×18	98.7 (±1.4)	52	98.7 (±1.4)	5	94.6 (±3.3)	90
Wdbc	569×30	93.7 (±3.1)	75	93.7 (±3.1)	8	91.9 (±3.9)	83
Segment	660×18	100.0 (±0.0)	21	100.0 (±0.0)	8	99.8 (±0.5)	11
Pima	768×8	69.7 (±5.4)	277	69.7 (±5.4)	44	69.1 (±5.1)	237
Masses	830×5	78.2 (±3.8)	309	78.2 (±3.8)	20	78.3 (±3.2)	253
Yeast	889×8	65.5 (±4.8)	445	65.5 (±4.8)	91	66.2 (±5.2)	280
Optdigits	1125×64	99.9 (±0.3)	50	99.9 (±0.3)	25	99.7 (±0.6)	15
Waveform	3347×40	91.8 (±1.3)	6788	91.8 (±1.3)	5254	91.7 (±1.0)	366
Robot	4302×24	90.7 (±1.2)	11212	90.7 (±1.2)	7528	90.5 (±1.5)	838
Page	5242×10	98.5 (±0.7)	907	98.5 (±0.7)	811	98.6 (±0.8)	548

Table 4.2: Test accuracies in percents and computational times in seconds for ELM kernel with LIBSVM and LIBLINEAR. Standard deviations are shown for accuracies. Computational times only include the calls to solvers. Datasets are sorted by numbers of instances. For datasets with more than 2 classes, only the two larger classes are used to obtain binary classification tasks.

For the different UCI datasets [67] considered in Table 4.2, LIBSVM and LIBLINEAR obtain similar results in terms of accuracy. However, the three compared settings correspond to quite different computational needs. For small datasets, the fastest approach consists in using LIBSVM with the precomputed Gram matrix $\mathbf{K} = \mathbf{H}\mathbf{H}^T$. LIBLINEAR is often the slowest solver for such datasets. However, for large datasets, using the precomputed Gram matrix becomes less attractive from a computational point of view. In fact, the computation of the Gram matrix itself also becomes very expensive and may become

unaffordable for very large datasets with hundreds of thousands of instances. For large datasets, LIBLINEAR seems to be by far the less expensive solver.

4.6 Conclusion

The literature in extreme learning shows that ELMs offer a good compromise between prediction accuracy and computational needs. ELMs allows one to obtain non-linear models in roughly the same time than linear regression, while the performances of ELMs are close to those of other state-of-the-art methods. Sections 4.3 and 4.4 show that the size of ELMs can be chosen large, if regularisation is used to control the model complexity. Since the hidden layer of an ELM can be seen as a non-linear mapping to a feature space, one can define an ELM kernel. When the number of neurons is infinite, this kernel admits an analytical form and can be used in classification and regression.

The experiments in Section 4.5 show that the choice of the solver depends on the characteristic of the dataset. In some cases, using the most appropriate solver allows learning 20 times faster. For large datasets, building an explicit feature space like in [9] seems to be more suitable than using a kernel like in [10].

An open question in extreme learning is whether the distribution of the hidden weights matters. Results in [10] show that the same isotropic Gaussian distribution can be used for most datasets, which is confirmed by the large number of successful application of ELMs. However, Parviainen et al. [312] show that the variance of the weight distribution may have an important impact on the performance of ELMs for some datasets. We believe that ELMs remain a good solution for applications where computational time is a critical resource. However, we also believe that further work should be done for high-dimensional datasets. For example, the worst results in [312] are obtained for high-dimensional datasets ($d = 10000$ and $d = 1558$). Since counterintuitive phenomena like the concentration of distances [28] occur in high-dimensional spaces, we believe that it is necessary to investigate what happens when random hidden weights are drawn from a Gaussian distribution. For example, it is likely that all vector of hidden weights will have roughly the same norm, which may cause the hidden neurons to have similar behaviours.

In conclusion, even if the researchers in extreme learning are sometimes overconfident in the efficiency of ELMs, extreme learning is a valuable tool for practical applications. In our opinion, most of the work has been done in the ELM literature: these models have to remain simple to be faster than other state-of-the-art models and it leaves not much space to improve them. We believe that the research should focus on theoretical problems related to

the properties of ELMs, since it is not yet clear how they actually work. As outlined by Parviainen et al. [312], the asymptotic ELM kernel allows one to obtain deterministic results (the model no longer depends on a particular set of random neurons), yet it raises a fundamental question: what is the role of hidden units besides increasing the variance of the ELM output ?

Chapter 5

Conclusion

I don't pretend we have all the answers. But the questions are certainly worth thinking about.

Arthur C. Clarke, writer

Contents

This section contextualises the results of this thesis with respect to the problems highlighted in the introduction and proposes a lead for future research.

5.1 Main Results of the Thesis

Three challenges of machine learning are highlighted in the introduction of this thesis: dealing with high-dimensional data, reducing the consequences of label noise in classification and obtaining satisfying results with limited computational resources. This thesis studies these problems and proposes several solutions.

As discussed in Sections 2.1 and 2.2, feature selection with mutual information allows reducing the dimensionality of data while keeping most of the informational content. Accordingly, many works have successfully used this approach to tackle high-dimensional datasets. However, until recently, the suitability of mutual information for feature selection has never been studied from a theoretical point of view in machine learning. Chapter 2 fills this gap and confirms that mutual information is indeed a valuable criterion for feature selection. Counterexamples are given where mutual information fails to reveal the optimal features with respect to the accuracy in classification or the mean square error in regression. However, such failures are theoretically and empirically shown to remain uncommon and of limited impact. We believe that the results of this thesis enhance the legitimacy of mutual information-based feature selection, while clearing up some common misunderstandings.

Chapter 3 proposes a survey of the label noise literature which shows that label noise has many potential sources and consequences. For example, the accuracy of classifiers is decreased and feature selection is altered in the presence of label noise. In our view, it is therefore surprising that standard machine learning methods for classification do not take label noise into account. Fortunately, many methods have been developed to deal with label noise (see Section 3.2), which can be divided in label noise-robust methods, data cleansing methods and label noise-tolerant methods. Two new label noise-tolerant methods based on a probabilistic modelling of label noise introduced by Lawrence et al. [74] (see Section 3.2.4) are proposed in Sections 3.3 and 3.4. First, since label noise has not yet been studied in the case of sequential data, Section 3.3 proposes a label noise-tolerant algorithm for hidden Markov model inference. Experiments on electrocardiogram signals show that a proper probabilistic modelling allows one to deal with label noise in more complex situations than i.i.d. data. Second, Section 3.4 shows that the popular Kozachenko-Leonenko entropy estimator is altered by label noise. Since mutual information is often estimated using this

estimator, a label noise-tolerant entropy estimator is derived and is shown to improve feature selection results. Sections 3.3 and 3.4 show that label noise is an important concern which should be (and can be) taken into account in various situations like e.g. feature selection. Chapter 3 also proposes a generic framework to deal with outliers, which is discussed in Section 5.2.

For certain real-world applications, computational resources are limited. In such cases, the extreme learning machines discussed in Chapter 4 allows one to obtain good results at a very low computational cost. Indeed, non-linear regression can be tackled with extreme learning at the cost of linear regression. Chapter 4 shows that the random hidden layer of an extreme learning machine can be seen as a mapping defining an ELM kernel. When regularisation is used to prevent overfitting, experiments show that one can choose a very large number of hidden neurons and yet obtain good results with this kernel. Actually, when the number of neurons becomes infinite, an analytical form can be derived for the ELM kernel which also gives good results. As for standard extreme learning machines, the prediction accuracy is pretty much insensitive to the width of the hidden weights prior. The developments of Chapter 4 are interesting because they offer an alternative to the popular RBF kernel for support vector machines and support vector regression. Indeed, in the literature, support vector regression is usually either not used or used incorrectly. In particular, the half-width of the tube ϵ is often fixed *a priori*. With the ELM kernel, proper meta-parameters tuning is no longer unaffordable, since only two meta-parameters have to be tuned for support vector regression (instead of three with the RBF kernel). We believe that extreme learning in general (and the ELM kernel in particular) offer a good compromise between computational cost and prediction accuracy. However, extreme learning is a relatively new field and many questions still await for answers. From our experience in extreme leaning, it seems very unlikely that the decrease in computational cost is not counterbalanced by at least a small decrease in generalisation abilities for some datasets. As a matter of fact, the results of Parviainen et al. [312] discussed in Section 4.6 show that it is at least the case for high-dimensional datasets. In our view, the ELM literature is sometimes overly optimistic, in the sense that there exist very few publications about the limitations of extreme learning. This state of affairs may harm the credibility of ELMs and prevent researchers to consider them. We believe that publications like e.g. [312] raise interesting questions which should be answered by the extreme learning community.

5.2 Going Further in Uncertainty Reduction

Uncertainty is one of the prevalent concept in this thesis. For example, uncertainty measures are used in Chapters 2 and **??** to select relevant features so as to obtain better and interpretable classifiers. Also, Chapter 3 shows that label noise is the cause of an additional uncertainty which obscures the relationship between the observed label and true class of instances. This thesis shows that proper probabilistic modelling of uncertainty allows one to deal with it. In particular, Chapter 2 confirms that mutual information (which measures uncertainty reduction in feature selection) is a relevant criterion to select features in classification and regression. Moreover, the simple probabilistic modelling of label noise proposed by Lawrence et al. [74] allows reducing the effects of label noise in electrocardiogram signal segmentation and feature selection.

One of the major drawbacks of the probabilistic modelling of label noise proposed by Lawrence et al. [74] is its inability to model more complex types of label noise. Indeed, the survey in Chapter 3 shows that label noise is not necessarily uniform and that labels may be e.g. less reliable near (or far from) the classification boundary. Unfortunately, on the one hand, it is seldom trivial to describe label noise in probabilistic terms. Indeed, the sources of label noise may be complex and various as discussed in Section 3.1.1. Moreover, most datasets are hard to visualise appropriately in order to characterise label noise. In our view, the characterisation of real-world label noise remains an important, yet open challenge. On the other hand, since the final goal is to reduce the undesirable effects of label noise, it is not obvious that modelling label noise accurately is necessary to deal with its consequences.

The method proposed in Section 3.5 is in our view a good alternative to deal with label noise, as well as any other source of uncertainty which pollutes observations. Indeed, pointwise probability reinforcements (PPRs) allows one to deal with abnormally frequent data (AFDs) which are hard to explain with the chosen probabilistic model. The influence of very unlikely training data (with respect to the considered model) can be easily reduced in problems like classification, regression and even unsupervised learning. Rather than specifying a parametric model for AFDs, what may require a lot of expert knowledge, the PPR approach instead specifies e.g. whether there are only a few AFDs.

The PPR approach has the advantage of being simple to implement. Moreover, this non-parametric approach provides a generic framework to study AFDs using similar tools in different contexts. The user can easily extract a list of suspicious instances from the training set. For all these reasons, we believe that it is necessary to further investigate PPRs. In particular, we are currently

adapting PPRs to Bayesian modelling, where they could allow mitigating the impact of outliers on the posterior distribution of model parameters.

Bibliography

[1] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. On the potential inadequacy of mutual information for feature selection. In *Proceedings of the 20th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, pages 501–506, 2012.

[2] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification. *Neurocomputing*, 112:64–78, 2013.

[3] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. Estimating mutual information for feature selection in the presence of label noise. Accepted in Computational Statistics & Data Analysis, 2013.

[4] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. Submitted to IEEE Transaction on Neural Networks, 2013.

[5] Benoît Frénay, Gaël de Lannoy, and Michel Verleysen. Label noise-tolerant hidden markov models for segmentation: application to ecgs. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part I*, pages 455–470, Athens, Greece, 2011.

[6] Benoît Frénay and Michel Verleysen. Pointwise probability reinforcements for robust statistical inference. Submitted to Neural Networks, 2013.

[7] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[8] G.B. Huang, D.H. Wang, and Y. Lan. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, 2(2):107–122, 2011.

[9] Benoît Frénay and Michel Verleysen. Using svms with randomised feature spaces: an extreme learning approach. In *Proceedings of The 18th European Symposium on Artificial Neural Networks (ESANN)*, pages 315–320, 2010.

[10] Benoît Frénay and Michel Verleysen. Parameter-insensitive kernel in extreme learning for non-linear support vector regression. *Neurocomputing*, 74(16):2526 – 2531, 2011.

[11] Gauthier Doquire, Benoît Frénay, and Michel Verleysen. Risk estimation and feature selection. In *Proceedings of the 21th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013)*, 2013.

[12] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. Is mutual information adequate for feature selection in regression ? *Neural Networks*, 48:1–7, 2013.

[13] Benoît Frénay and Marco Saerens. Ql2, a simple reinforcement learning scheme for two-player zero-sum markov games. In *Proceedings of the 16th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2008)*, pages 137–142, 2008.

[14] Benoît Frénay and Marco Saerens. A simple reinforcement learning scheme for two-player zero-sum markov games. *Neurocomputing*, 72(7-9):1494–1507, 2009.

[15] Isabelle Thomas, Pierre Frankhauser, Benoît Frénay, and Michel Verleysen. Clustering patterns of urban builtup areas with curves of fractal scaling behavior. In *Proceedings of ASRDLF 2009, l'Association de Science Régionale de Langue Française*, 2009.

[16] Isabelle Thomas, Pierre Frankhauser, Benoît Frénay, and Michel Verleysen. Clustering fractal urban patterns with curves of scaling behavior. In *Proceedings of the 49th Congress of the European Regional Science Association " Territorial cohesion of Europe and integrative planning" (ERSA 2009)*, 2009.

[17] Isabelle Thomas, Pierre Frankhauser, Benoît Frénay, and Michel Verleysen. Clustering patterns of urban built-up areas with curves of fractal scaling behaviour. *Environment and Planning B: Planning and Design*, 37(5):942–954, 2010.

[18] Benoît Frénay, Gaël de Lannoy, and Michel Verleysen. Emission modelling for supervised ecg segmentation using finite differences. In *4th European Conference of the International Federation for Medical and Biological Engineering*, pages 1212–1216. 2009.

[19] G. Lannoy, B. Frénay, M. Verleysen, and J. Delbeke. Supervised ecg delineation using the wavelet transform and hidden markov models. In *4th European Conference of the International Federation for Medical and Biological Engineering*, pages 22–25. 2009.

[20] Benoît Frénay, Gael de Lannoy, and Michel Verleysen. Improving the transition modelling in hidden markov models for ecg segmentation. In *Proceedings of the 17th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2009)*, 2009.

[21] Benoît Frénay, Mark van Heeswijk, Yoan Miche, Michel Verleysen, and Amaury Lendasse. Feature selection for nonlinear models with extreme learning machines. *Neurocomputing*, 102:111–124, 2013.

[22] Laura Kainulainen, Yoan Miche, Emil Eirola, Qi Yu Yu, Benoît Frénay, Eric Séverin, and Amaury Lendasse. Ensembles of local linear models for bankruptcy analysis and prediction. *Case Studies in Business, Industry and Government Statistics (CSBIGS)*, 4(2):116–133, 2011.

[23] Richard E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, 1961.

[24] Michel Verleysen. Learning high-dimensional data. *Limitations and Future Trends in Neural Computation*, 186:141–162, 2003.

[25] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46:389–422, 2002.

[26] Jae Won Lee, Jung Bok Lee, Mira Park, and Seuck Heun Song. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, 48(4):869 – 885, 2005.

[27] Thibault Helleputte and Pierre Dupont. Feature selection by transfer learning with linear regularized models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, pages 533–547, 2009.

[28] Damien Francois, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Trans. on Knowl. and Data Eng.*, 19(7):873–886, July 2007.

[29] Sayan Mukherjee. *A Practical Approach to Microarray Data Analysis*, chapter Classifying microarray data using support vector machines. Kluwer Academic Publishers, 2003.

[30] Lawrence Carin, Balaji Krishnapuram, and Alexander Hartemink. *Kernel Methods in Computational Biology*, chapter Gene expression analysis: Joint feature selection and classifier design. Bradford Bks, 2004.

[31] Ping Xu, Guy N. Brock, and Rudolph S. Parrish. Modified linear discriminant analysis approaches for classification of high-dimensional microarray data. *Computational Statistics & Data Analysis*, 53(5):1674 – 1687, 2009.

[32] Amir Globerson and Naftali Tishby. Sufficient dimensionality reduction. *J. Mach. Learn. Res.*, 3:1307–1331, 2003.

[33] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[34] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[35] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc. B.*, 58:267–288, 1996.

[36] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[37] Tim C. Hesterberg, Nam H. Choi, Lukas Meier, and Chris Fraley. Least angle and l1 penalized regression: A review. *Statistics Surveys*, 2008.

[38] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2008.

[39] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Royal. Statist. Soc. B.*, 68:49–67, 2006.

[40] I. Kojadinovic and T Vottka. Comparison between a filter and a wrapper approach to variable subset selection in regression problems. In *In Proceedings of ESIT*, 2000.

[41] N. Benoudjit, D. Francois, M. Meurens, and M. Verleysen. Spectrophotometric variable selection by mutual information. *Chemometrics and Intelligent Laboratory Systems*, 74:243–251, 2004.

[42] Francois Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[43] Ivan Kojadinovic. On the use of mutual information in data analysis : an overview. *Proceedings of 11th International Symposium on Applied Stochastic Models and Data Analysis ASMDA05*, page 738747, 2005.

[44] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238, 2005.

[45] Gert Van Dijck and Marc M. Van Hulle. Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis. In *In Proceedings of the 16th International Conference on Artificial Neural Networks*, pages 31–40, 2006.

[46] Fabrice Rossi, Amaury Lendasse, Damien Francois, Vincent Wertz, and Michel Verleysen. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics and Intelligent Laboratory Systems*, 80(2):215–226, 2006.

[47] Carlos Guerrero-Mosquera, Michel Verleysen, and Angel Navia Vazquez. Eeg feature selection using mutual information and support vector machine: A comparative analysis. In *Proceedings of EMBC 2010*, 2010.

[48] Gauthier Doquire and Michel Verleysen. Feature selection with mutual information for uncertain data. In *Data Warehousing and Knowledge Discovery*, pages 330–341. 2011.

[49] Gauthier Doquire and Michel Verleysen. Mutual information for feature selection with missing data. In *Proceedings of ESANN'11*, 2011.

[50] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550, 1994.

[51] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[52] R.B. Ash. *Information Theory*. Dover Publ., 1965.

[53] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, 1948.

[54] L. F. Kozachenko and N. Leonenko. Sample estimate of the entropy of a random vector. *Problems Inform. Transmission*, 23:95–101, 1987.

[55] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69:066138, 2004.

[56] Janett Walters-Williams and Yan Li. Estimation of mutual information: A survey. In *Proceedings of the 4th International Conference on Rough Sets and Knowledge Technology*, pages 389–396, Berlin, Heidelberg, 2009.

[57] Vanessa Gómez-Verdejo, Michel Verleysen, and Jérôme Fleury. Information-theoretic feature selection for the classification of hysteresis curves. In *Proceedings of the 9th international work conference on Artificial neural networks*, pages 522–529, Berlin, Heidelberg, 2007.

[58] Gauthier Doquire and Michel Verleysen. A comparison of multivariate mutual information estimators for feature selection. In *Proceeding of ICPRAM'12*, 2012.

[59] 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):1276–1288, 2007.

[60] Michel Verleysen, Fabrice Rossi, and Damien FranÃ§ois. Advances in feature selection with mutual information. In *Similarity-Based Clustering*, pages 52–69. 2009.

[61] Harald Stögbauer, Er Kraskov, Sergey A. Astakhov, and Peter Grassberger. Least-dependent-component analysis based on mutual information. *Phys. Rev. E*, 70:066123, 2004.

[62] M. Hellman and J. Raviv. Probability of error, equivocation, and the chernoff bound. *IEEE Transactions on Information Theory*, 16(4):368–372, July 1970.

[63] R. Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, 1961.

[64] J. W. Fisher, M. Siracusa, and T. Kihn. Estimation of signal information content for classification. In *Proceedings of DSP/SPE 2009*, 2009.

[65] Gavin Brown. An information theoretic perspective on multiple classifier systems. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, pages 344–353, Berlin, Heidelberg, 2009.

[66] U. Ozertem, D. Erdogmus, and R. Jenssen. Spectral feature projections that maximize Shannon mutual information with class labels. *Pattern Recogn.*, 39:1241–1252, 2006.

[67] D.J. Newman A. Asuncion. UCI machine learning repository.

[68] S. Ihara. *Information Theory for Continuous System*. World Scientific, 1993.

[69] Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1261–1282, 2005.

[70] Badong Chen, Jinchun Hu, Hongbo Li, and Zengqi Sun. Adaptive filtering under maximum mutual information criterion. *Neurocomputing*, 71(16-18):3680 – 3684, 2008.

[71] D. Peel and G. J. McLachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 10:339–348, 2000.

[72] Cédric Archambeau, Nicolas Delannay, and Michel Verleysen. Mixtures of robust probabilistic principal component analyzers. *Neurocomputing*, 71(7-9):1274–1282, 2008.

[73] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. Mutual information: an adequate tool for feature selection. In *Proceedings of the 22nd Belgian-Dutch Conference on Machine Learning*, 2013.

[74] Neil D. Lawrence and Bernhard Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 306–313, San Francisco, CA, USA, 2001.

[75] Ray J. Hickey. Noise modelling and evaluating learning from examples. *Artificial Intelligence*, 82(1-2):157–179, 1996.

[76] Robert Hanner, Sven Becker, Natalia V Ivanova, and Dirk Steinke. Fish-bol and seafood identification: geographically dispersed case studies reveal systemic market substitution across canada. *Mitochondrial DNA*, 22:106–122, 2011.

[77] Eva Garcia-Vazquez, Gonzalo Machado-Schiaffino, Daniel Campo, and Francis Juanes. Species misidentification in mixed hake fisheries may lead to overexploitation and population bottlenecks. *Fisheries Research*, 114(0):52 – 55, 2012.

[78] Claudia Lopez-Vizcón and Felisa Ortega. Detection of mislabelling in the fresh potato retail market employing microsatellite markers. *Food Control*, 26(2):575 – 579, 2012.

[79] Donna-MareÃ¨ Cawthorn, Harris A. Steinman, and Louwrens C. Hoffman. A high incidence of species substitution and mislabelling detected in meat products sold in south africa. *Food Control*, 32(2):440 – 449, 2013.

[80] L. G. Valiant. Learning disjunction of conjunctions. In *Proceedings of the 9th international joint conference on Artificial intelligence - Volume 1*, pages 560–566, San Francisco, CA, USA, 1985.

[81] Michael Kearns and Ming Li. Learning in the presence of malicious errors. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 267–280, Chicago, Illinois, United States, 1988.

[82] Scott Evan Decatur. Statistical queries and faulty pac oracles. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 262–268, New York, NY, USA, 1993.

[83] Scott E. Decatur. *Learning from Data: AI and Statistics V*, chapter Learning in Hybrid Noise Environments Using Statistical Queries, pages 175–185. Springer Verlag, 1995.

[84] Robert H. Sloan. Four types of noise in data for pac learning. *Information*, 54(3):157–162, 1995.

[85] Peter Auer and Nicolò Cesa-Bianchi. On-line learning with malicious noise and the closure algorithm. *Annals of Mathematics and Artificial Intelligence*, 23(1-2):83–99, 1998.

[86] Nicolò Cesa-Bianchi, Eli Dichterman, Paul Fischer, Eli Shamir, and Hans Ulrich Simon. Sample-efficient strategies for learning in the presence of noise. *Journal of the ACM*, 46(5):684–719, 1999.

[87] Rocco A. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.

[88] Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Proceedings of the 3rd Asian Conference on Machine Learning*, pages 97–112, Taoyuan, Taiwan, 2011.

[89] Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *ECAI*, pages 870–875, 2012.

[90] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[91] Xindong Wu. *Knowledge acquisition from databases*. Ablex Publishing Corp., 1996.

[92] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22:177–210, 2004.

[93] JosèA. Sàez, Mikel Galar, Julián Luengo, and Francisco Herrera. Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. *Knowledge and Information Systems*, pages 1–28, 2012.

[94] Carla E. Brodley and Mark A. Friedl. Identifying mislabeled training data. *Journal Of Artificial Intelligence Research*, 11:131–167, 1999.

[95] Mykola Pechenizkiy, Alexey Tsymbal, Seppo Puuronen, and Oleksandr Pechenizkiy. Class noise and supervised learning in medical domains: The effect of feature extraction. In *Proceedings of the 19th IEEE International Symposium on Computer-Based Medical Systems*, pages 708–713, 2006.

[96] P. B. Brazdil, K. Konolige, Boston Kluwer, Pavel Brazdil Peter, and Peter Clark. *Machine Learning, Meta-Reasoning and Logics*, chapter Learning from Imperfect Data, pages 207–232. Kluwer Academic Publishers, 1990.

[97] A P Dawid and A M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

[98] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast - but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Stroudsburg, PA, USA, 2008.

[99] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67, Washington DC, DC, USA, 2010.

[100] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

[101] Man-Ching Yuen, I. King, and Kwong-Sak Leung. A survey of crowdsourcing systems. In *Proceedings of the IEEE Third International Conference on Social Computing*, pages 766 –773, Boston, MA, USA, 2011.

[102] Padhraic Smyth, Usama M. Fayyad, Michael C. Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems 7*, pages 1085–1092, 1994.

[103] Padhraic Smyth. Bounds on the mean classification error rate of multiple experts. *Pattern Recognition Letters*, 17(12):1253–1257, 1996.

[104] Andrea Malossini, Enrico Blanzieri, and Raymond T. Ng. Detecting potential labeling errors in microarrays by data perturbation. *Bioinformatics*, 22(17):2114–2121, 2006.

[105] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

[106] Nicholas P Hughes, Stephen J Roberts, and Lionel Tarassenko. Semi-supervised learning of probabilistic models for ecg segmentation. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 434–437, San Francisco, California, USA, 2004.

[107] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988.

[108] D. Sculley and Gordon V. Cormack. Filtering email spam in the presence of noisy user feedback. In *Proceedings of the fifth conf on email and anti-spam*, 2008.

[109] Ken Orr. Data quality and systems theory. *Communications of the ACM*, 41(2):66–71, February 1998.

[110] Thomas Redman. The impact of poor data quality on the typical enterprise. *Communications of the ACM*, 2(2):79–82, 1998.

[111] Jonathan I. Maletic and Andrian Marcus. Data cleansing: Beyond integrity analysis. In *Proceedings of the Conference on Information Quality*, pages 200–209, 2000.

[112] Joseph L. Schafer and John W. Graham. Missing data: our view of the state of the art. *Psychological methods*, 7(2):147–177, June 2002.

[113] Peter A. Lachenbruch. Discriminant analysis when the initial samples are misclassified. *Technometrics*, 8(4):657–662, 1966.

[114] G. J. McLachlan. Asymptotic results for discriminant analysis when the initial samples are misclassified. *Technometrics*, 14(2):415–422, 1972.

[115] Joel E. Michalek and Ram C. Tripathi. The effect of errors in diagnosis and measurement on the estimation of the probability of an event. *Journal of the American Statistical Association*, 75(371):713–721, 1980.

[116] Tom Heskes. The use of being stubborn and introspective. In *Proceedings of the ZiF Conference on Adaptative Behavior and Learning*, pages 55–65, Bielefeld, Germany, 1994.

[117] Yingtao Bi and Daniel R. Jeske. The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise. *Journal of Multivariate Analysis*, 101(7):1622–1637, 2010.

[118] Peter A. Lachenbruch. Note on initial misclassification effects on the quadratic discriminant function. *Technometrics*, 21(1):129–132, 1979.

[119] Seishi Okamoto and Yugami Nobuhiro. An average-case analysis of the k-nearest neighbor classifier for noisy domains. In *Proceedings of the 15th international joint conference on Artifical intelligence - Volume 1*, pages 238–243, San Francisco, CA, USA, 1997.

[120] J.S. Sánchez, F. Pla, and F.J. Ferri. Prototype selection for the nearest neighbour rule through proximity graphs. *Pattern Recognition Letters*, 18(6):507–513, 1997.

[121] D Randall Wilson and Tony R Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

[122] Gary M. Weiss. Learning with rare cases and small disjuncts. In *Proceedings of Twelfth International Conference on Machine Learning*, pages 558–565, 1995.

[123] Jian Zhang and Yiming Yang. Robustness of regularized linear classification methods in text categorization. In *Proceedings the Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 190–197, 2003.

[124] David Nettleton, Albert Orriols-Puig, and Albert Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306, 2010.

[125] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[126] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[127] Ross A. McDonald, David J. Hand, and Idris A. Eckley. An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *Proceedings of the Fourth International Workshop on Multiple Classifier Systems*, pages 35–44, Surrey, UK, 2003.

[128] Prem Melville, Nishit Shah, Lilyana Mihalkova, and Raymond J. Mooney. Experiments on ensembles with missing and noisy data. In *Proceedings of the Fifth International Workshop on Multi Classifier Systems*, pages 293–302, Cagliari, Italy, 2004.

[129] Wenxin Jiang. Some theoretical aspects of boosting in the presence of noisy data. In *Proceedings of The Eighteenth International Conference on Machine Learning*, pages 234–241, Williamstown, MA, USA, 2001.

[130] Joaquín Abellán and Andrés R. Masegosa. Bagging decision trees on data sets with classification noise. In *Proceedings of the 6th international conference on Foundations of Information and Knowledge Systems*, pages 248–265, Berlin, Heidelberg, 2010.

[131] Peter A. Lachenbruch. Discriminant analysis when the initial samples are misclassified ii: Non-random misclassification models. *Technometrics*, 16(3):419–424, 1974.

[132] Raj S. Chhikara and Jim McKeon. Linear discriminant analysis with misallocation in training samples. *Journal of the American Statistical Association*, 79(388):899–906, 1984.

[133] Philip D. Laird. *Learning from good and bad data.* Kluwer Academic Publishers, 1988.

[134] Javed A. Aslam. On the sample complexity of noise-tolerant learning. *Information Processing Letters*, 57(4):189–195, 1996.

[135] Claudio Gentile. Improved lower bounds for learning from noisy examples: an information-theoretic approach. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 104–115, Madison, WI, USA, 1998.

[136] Giampaolo L. Libralon, André Carlos Ponce de Leon Ferreira de Carvalho, and Ana Carolina Lorena. Pre-processing for noise detection in gene expression classification data. *Journal of the Brazilian Computer Society*, 15(1):3–11, 2009.

[137] Irwin Bross. Misclassification in 2 x 2 tables. *Biometrics*, 10(4):478–486, 1954.

[138] Aaron Tenenbein. A double sampling scheme for estimating from binomial data with misclassifications. *Journal of the American Statistical Association*, 65(331):1350–1361, 1970.

[139] Anil Gaba and Robert L. Winkler. Implications of errors in survey data: A bayesian model. *Management Science*, 38(7):913–925, 1992.

[140] Peter F. Thall, Derek Jacoby, and Stuart O. Zimmerman. Estimating genomic category probabilities from fluorescent in situ hybridization counts with misclassification. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 45(4):431–446, 1996.

[141] Susan L. Stewart, Karen C. Swallen, Sally L. Glaser, Pamela L. Horn-Ross, and Dee W. West. Adjustment of cancer incidence rates for ethnic misclassification. *Biometrics*, 54(2):774–781, 1998.

[142] Chuck P. Lam and David G. Stork. Evaluating classifiers by means of test data with noisy labels. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 513–518, San Francisco, CA, USA, 2003.

[143] Gordon V. Cormack and Aleksander Kolcz. Spam filter evaluation with imprecise ground truth. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 604–611, New York, NY, USA, 2009.

[144] Wensheng Zhang, Romdhane Rekaya, and Keith Bertrand. A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. *Bioinformatics*, 22(3):317–325, 2006.

[145] Richard Gerlach and James Stamey. Bayesian model selection for logistic regression with misclassified outcomes. *Statistical Modelling*, 7(3):255–273, 2007.

[146] Ahmad Abu Shanab, Taghi M. Khoshgoftaar, and Randall Wald. Robustness of threshold-based feature rankers with data sampling on noisy and imbalanced data. In *FLAIRS Conference*, 2012.

[147] Choh-Man Teng. Evaluating noise correction. In *Proceedings of the 6th Pacific Rim international conference on Artificial intelligence*, pages 188–198, 2000.

[148] Choh-Man Teng. A comparison of noise handling techniques. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pages 269–273, 2001.

[149] Choh-Man Teng. Dealing with data corruption in remote sensing. In *Proceedings of the 6th international conference on Advances in Intelligent Data Analysis*, pages 452–463, 2005.

[150] Shahram Golzari, Shyamala Doraisamy, Md Nasir Sulaiman, and Nur Izura Udzir. The effect of noise on rwtsairs classifier. *European Journal of Scientific Research*, 31(4):632–641, 2009.

[151] Charles Bouveyron and Stéphane Girard. Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recognition*, 42(11):2649–2658, 2009.

[152] Naresh Manwani and P. S. Sastry. Noise tolerance under risk minimization. *CoRR*, 2011.

[153] Hua Yin and Hongbin Dong. The problem of noise in classification: Past, current and future work. In *IEEE 3rd International Conference on Communication Software and Networks*, pages 412 –416, Xi'an, China, 2011.

[154] M.A.L. Thathachar and P.S. Sastry. *Networks of learning automata: techniques for online stochastic optimization*. Kluwer Academic, 2004.

[155] Eyal Beigman and Beata Beigman Klebanov. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 280–287, Suntec, Singapore, 2009.

[156] P. S. Sastry, G. D. Nagendra, and Naresh Manwani. A team of continuous-action learning automata for noise-tolerant learning of half-spaces. *Transactions on Systems, Man, and Cybernetics: Part B*, 40(1):19–28, February 2010.

[157] Andres Folleco, Taghi M. Khoshgoftaar, Jason Van Hulse, and Lofton A. Bullard. Software quality modeling: The impact of class noise on the random forest classifier. In *IEEE Congress on Evolutionary Computation*, pages 3853–3859, Hong Kong, China, 2008.

[158] Andres Folleco, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Identifying learners robust to low quality data. *Informatica*, 33:245–259, 2009.

[159] Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors. *IEEE Transactions on Neural Networks*, 21:813–830, May 2010.

[160] Gunnar Rätsch, Takashi Onoda, and Klaus-Robert Müller. Regularizing adaboost. In *Advances in Neural Information Processing Systems 11*, pages 564–570, 1998.

[161] Gunnar Rätsch, Takashi Onoda, and Klaus Robert Müller. An improvement of adaboost to avoid overfitting. In *Proceedings of the Fifth International Conference on Neural Information Processing*, pages 506–509, 1998.

[162] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–374, 2000.

[163] Gunnar Rätsch, Bernhard Schölkopf, Alex J. Smola, Sebastian Mika, Takashi Onoda, and Klaus-Robert Müller. Robust ensemble learning for data mining. In *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, pages 341–344, London, UK, 2000.

[164] Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293–318, June 2001.

[165] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. *Machine Learning*, 42(3):287–320, 2001.

[166] Joaquín Abellán and Serafín Moral. Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12):1215–1225, 2003.

[167] Joaquín Abellán and Andrés R. Masegosa. Bagging schemes on the presence of class noise in classification. *Expert Systems with Applications*, 39(8):6827–6837, 2012.

[168] D. M. Hawkins. *Identification of outliers*. Chapman and Hall, 1980.

[169] R. J. Beckman and R. D. Cook. Outlier.....s. *Technometrics*, 25(2):119–149, 1983.

[170] Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. Wiley, 1994.

[171] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[172] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems 12*, pages 582–588, 1999.

[173] Paul Hayton, Bernhard Schölkopf, Lionel Tarassenko, and Paul Anuzis. Support vector novelty detection applied to jet engine vibration spectra. In *Advances in Neural Information Processing Systems 13*, pages 946–952, Denver, CO, USA, 2000.

[174] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[175] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[176] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, 2009.

[177] Jiang-wen Sun, Feng-ying Zhao, Chong-jun Wang, and Shi-fu Chen. Identifying and correcting mislabeled training instances. In *Proceedings of the Future Generation Communication and Networking - Volume 1*, pages 244–250, Washington, DC, USA, 2007.

[178] Dragan Gamberger, Nada Lavrač, and Sašo Džeroski. Noise elimination in inductive concept learning: A case study in medical diagnosis. In *Algorithmic Learning Theory*, pages 199–212. 1996.

[179] Dragan Gamberger and Nada Lavrač. Conditions for occam's razor applicability and noise elimination. In *Proceedings of the 9th European Conference on Machine Learning*, pages 108–123, Prague, Czech Republic, 1997.

[180] Dragan Gamberger and Nada Lavrač. Noise detection and elimination applied to noise handling in a krk chess endgame. In *Selected Papers from the 6th International Workshop on Inductive Logic Programming*, pages 72–88. 1997.

[181] Dragan Gamberger, Rudjer Boskovic, Nada Lavrac, and Ciril Groselj. Experiments with noise filtering in a medical domain. In *Proceedings of the 16th International Conference on Machine Learning*, pages 143–151, Bled, Slovenia, 1999.

[182] Dragan Gamberger, Nada Lavrač, and Sašo Džeroski. Noise detection and elimination in data preprocessing: experiments in medical domains. *Applied artificial intelligence*, 14:205–223, 2000.

[183] Taghi M. Khoshgoftaar and Pierre Rebours. Generating multiple noise elimination filters with the ensemble-partitioning filter. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration*, pages 369–375, Las Vegas, NV, USA, 2004.

[184] Jaree Thongkam, Guandong Xu, Yanchun Zhang, and Fuchun Huang. Support vector machine for outlier detection in breast cancer survivability prediction. In *Advanced Web and NetworkTechnologies, and Applications*, pages 99–109. 2008.

[185] André L. Miranda, Luís Paulo Garcia, André C. Carvalho, and Ana C. Lorena. Use of classification algorithms in noise detection and elimination. In *Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems*, pages 417–424, Berlin, Heidelberg, 2009.

[186] Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung. Data cleaning for classification using misclassification analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 14(3):297–302, 2010.

[187] N. Matic, I. Guyon, L. Bottou, J. Denker, and V. Vapnik. Computer aided cleaning of large databases for character recognition. In *Proceedings of the 11th IAPR International Conference on Pattern Recognition, Conference B: Pattern Recognition Methodology and Systems*, pages 330–333, 1992.

[188] Isabelle Guyon, Nada Matic, and Vladimir Vapnik. Advances in knowledge discovery and data mining. pages 181–203. 1996.

[189] George H. John. Robust decision trees: Removing outliers from databases. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 174–179, Montreal, Quebec, Canada, 1995.

[190] Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

[191] Nicola Segata, Enrico Blanzieri, and Pàdraig Cunningham. A scalable noise reduction technique for large case-based systems. In *Proceedings of the 8th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, pages 328–342, 2009.

[192] Nicola Segata, Enrico Blanzieri, Sarah Delany, and Pàdraig Cunningham. Noise reduction for instance-based learning with a local maximal margin approach. *Journal of Intelligent Information Systems*, 35(2):301–331, 2010.

[193] Carla E. Brodley and Mark A. Friedl. Identifying and eliminating mislabeled training instances. In *Proceedings of the thirteenth national conference on Artificial intelligence*, pages 799–805, Portland, Oregon, 1996.

[194] C. E. Brodley and M. A. Friedl. Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data. In *Proceedings of the 1996 International Geoscience and Remote Sensing Symposium*, pages 27–31, Lincoln, Nebraska, USA, 1996.

[195] Harald Berthelsen and Beáta Megyesi. Ensemble of classifiers for noise detection in pos tagged corpora. In *Proceedings of the Third International Workshop on Text, Speech and Dialogue*, pages 27–32, London, UK, 2000.

[196] S. Verbaeten. Identifying mislabeled training examples in ilp classification problems. In *Proceedings of Twelfth Belgian-Dutch Conference on Machine Learning*, pages 71–78, 2002.

[197] Sofie Verbaeten and Anneleen Van Assche. Ensemble methods for noise elimination in classification problems. In *Proceedings of the 4th international conference on Multiple classifier systems*, pages 317–325, Berlin, Heidelberg, 2003.

[198] Borut Sluban, Dragan Gamberger, and Nada Lavrac. Advances in class noise detection. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 1105–1106, 2010.

[199] Xingquan Zhu, Xindong Wu, and Qijun Chen. Eliminating class noise in large datasets. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 920–927, 2003.

[200] Y. Xiao, T.M. Khoshgoftaar, and N. Seliya. The partitioning- and rule-based filter for noise detection. In *IEEE International Conference on Information Reuse and Integration*, pages 205–210, Las Vegas, NV, USA, 2005.

[201] Xingquan Zhu, Xindong Wu, and Qijun Chen. Bridging local and global data cleansing: Identifying class noise in large, distributed data datasets. *Data Mining and Knowledge Discovery*, 12(2-3):275–308, 2006.

[202] Chen Zhang, Chunguo Wu, Enrico Blanzieri, You Zhou, Yan Wang, Wei Du, and Yanchun Liang. Methods for labeling error detection in microarrays based on the effect of data perturbation on the regression model. *Bioinformatics*, 25(20):2708–2714, 2009.

[203] You Zhou, Chong Xing, Wei Shen, Ying Sun, Jianan Wu, and Xu Zhou. A fast algorithm for outlier detection in microarray. In *Advances in Computer Science, Environment, Ecoinformatics, and Education - International Conference*, pages 513–519, 2011.

[204] D. Randall Wilson and Tony R. Martinez. Instance pruning techniques. In *Proceedings of the International Conference on Machine Learning*, pages 403–411, 1997.

[205] Giampaolo Libralon, André Carvalho, and Ana Lorena. Ensembles of pre-processing techniques for noise detection in gene expression data. In *Proceedings of the 15th international conference on Advances in neuro-information processing - Volume Part I*, pages 486–493, 2009.

[206] Sarah Jane Delany, Nicola Segata, and Brian Mac Namee. Profiling instances in noise reduction. *Knowledge-Based Systems*, 31:28–40, 2012.

[207] P. Hart. The condensed nearest neighbor rule (corresp.). *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.

[208] Geoffrey W. Gates. The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18(3):431–433, 1972.

[209] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions On Systems Man And Cybernetics*, 2(3):408–421, 1972.

[210] Ivan Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):448 –452, 1976.

[211] J. Koplowitz. On the relation of performance to editing in nearest neighbor rules. *Pattern Recognition*, 13(3):251–255, 1981.

[212] Ricardo Barandela and Eduardo Gasca. Decontamination of training samples for supervised pattern recognition methods. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 621–630, 2000.

[213] Sarah Jane Delany and Padraig Cunningham. An analysis of case-base editing in a spam filtering system. In *Proceedings of the 7th European Conference on Case Based Reasoning*, pages 128–141, Madrid, Spain, 2004.

[214] Annalisa Franco, Davide Maltoni, and Loris Nanni. Data pre-processing through reward-punishment editing. *Pattern Analysis and Applications*, 13(4):367–381, 2010.

[215] Loris Nanni and Annalisa Franco. Reduced reward-punishment editing for building ensembles of classifiers. *Expert Systems with Applications*, 38(3):2395–2400, 2011.

[216] Virginia Wheway. Using boosting to detect noisy data. In *Revised Papers from the PRICAI 2000 Workshop Reader, Four Workshops held at PRICAI 2000 on Advances in Artificial Intelligence*, pages 123–132, London, UK, 2001.

[217] Amitava Karmaker and Stephen Kwek. A boosting approach to remove class label noise. *International Journal of Hybrid Intelligent Systems*, 3(3):169–177, 2006.

[218] Yunlong Gao, Feng Gao, and Xiaohong Guan. Improved boosting algorithm with adaptive filtration. In *Proceedings of the 8th World Congress on Intelligent Control and Automation*, pages 3173–3178, 2010.

[219] A. Srinivasan, S. Muggleton, and M. Bain. Distinguishing exceptions from noise in non monotonic learning. In *Proceedings of the 2nd International Workshop on Inductive Logic Programming*, pages 97–107, 1992.

[220] Lawrence Joseph, Theresa W. Gyorkos, and Louis Coupal. Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard. *American Journal of Epidemiology*, 141(3):263–272, 1995.

[221] M. Evans, I. Guttman, Y. Haitovsky, and T. Swartz. *Bayesian Analysis in Statistics and Econometrics: Essays in Honor of Arnold Zellner*, chapter Bayesian Analysis of Binary Data Subject to Misclassification, pages 67–77. Wiley, 1996.

[222] Tim Swartz, Yoel Haitovsky, Albert Vexler, and Tae Yang. Bayesian identifiability and misclassification in multinomial data. *The Canadian Journal of Statistics*, 32(3):285–302, 2004.

[223] Anil Gaba. Inferences with an unknown noise level in a bernoulli process. *Management Science*, 39(10):1227–1237, 1993.

[224] R. L. Winkler. *Bayesian Statistics 2*, chapter Information Loss in Noisy and Dependent Processes, pages 559–570. Elsevier Science Publishers, 1985.

[225] María-Gloria Basànez, Clare Marshall, Hélène Carabin, Theresa Gyorkos, and Lawrence Joseph. Bayesian statistics for parasitologists. *Trends in Parasitology*, 20(2):85–91, 2004.

[226] C. J. Perez, F. J. Giron, J. Martin, M. Ruiz, and C. Rojano. Misclassified multinomial data: a bayesian approach. *Rev. R. Acad. Cien. Serie A. Mat.*, 101(1):71–80, 2007.

[227] Alula Hadgu, Nandini Dendukuri, and Joergen Hilden. Evaluation of nucleic acid amplification tests in the absence of a perfect gold-standard test: A review of the statistical and epidemiologic issues. *Epidemiology*, 16(5):604–612, 2005.

[228] Martin Ladouceur, Elham Rahme, Christian A. Pineau, and Lawrence Joseph. Robustness of prevalence estimates derived from misclassified data from administrative databases. *Biometric*, 63(1):272–279, 2007.

[229] Wesley O. Johnson and Joseph L. Gastwirth. Bayesian inference for medical screening tests: Approximations useful for the analysis of acquired immune deficiency syndrome. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(2):427–439, 1991.

[230] Lawrence Joseph and Theresa W. Gyorkos. Inferences for likelihood ratios in the absence of a "gold standard". *Medical Decision Making*, 16(4):412–417, 1996.

[231] Paul Gustafson, Nhu D. Le, and Refik Saskin. Case-control analysis with partial knowledge of exposure misclassification probabilities. *Biometrics*, 57(2):598–609, 2001.

[232] R. Rekaya, K. A. Weigel, and D. Gianola. Threshold model for misclassified binary responses with applications to animal breeding. *Biometrics*, 57(4):1123–1129, 2001.

[233] Carlos Daniel Paulino, Paulo Soares, and John Neuhaus. Binomial regression with misclassification. *Biometrics*, 59(3):670–675, 2003.

[234] M. Ruiz, F. J. Girón, C. J. Pérez, J. Martín, and C. Rojano. A bayesian model for multinomial sampling with misclassified data. *Journal of Applied Statistics*, 35(4):369–382, 2008.

[235] Juxin Liu, Paul Gustafson, Nicola Cherry, and Igor Burstyn. Bayesian analysis of a matched case-control study with expert prior information on both the misclassification of exposure and the exposure-disease association. *Statistics in Medicine*, 28(27):3411–3423, 2009.

[236] Jorge Alberto Achcar, E. Z Martinez, and F Louzada-Neto. Binary data in the presence of misclassifications. In *16th Symposium of the International Association for Statistical Computing*, pages 581–587, Praga, Czech Republic, 2004.

[237] Pat McInturff, Wesley O Johnson, David Cowling, and Ian A Gardner. Modelling risk when binary outcomes are subject to error. *Statistics in Medicine*, 23(7):1095–1109, 2004.

[238] Carlos Daniel Paulino, Giovani Silva, and Jorge Alberto Achcar. Bayesian analysis of correlated misclassified binary data. *Computational Statistics & Data Analysis*, 49(4):1120–1131, 2005.

[239] M. J. García-Zattera, T. Mutsvari, A. Jara, D. Declerckc, and E. Lesaffrea. Correcting for misclassification for a monotone disease process with an application in dental research. *Statistical Medecine*, 29(30):3103–3117, 2010.

[240] Frederik O. Kaster, Bjoern H. Menze, Marc-André Weber, and Fred A. Hamprecht. Comparative validation of graphical models for learning tumor segmentations from noisy manual annotations. In *Proceedings of the 2010 international MICCAI conference on Medical computer vision: recognition techniques and applications in medical imaging*, pages 74–85, Beijing, China, 2011.

[241] K Robbins, S Joseph, W Zhang, R Rekaya, and JK Bertrand. Classification of incipient alzheimer patients using gene expression data: Dealing with potential misdiagnosis. *Online Journal of Bioinformatics*, 7(1):22–31, 2006.

[242] Daniel Hernandez-Lobato, José Miguel Hernandez-Lobato, and Pierre Dupont. Robust multi-class gaussian process classification. In *Advances in Neural Information Processing Systems 24*, pages 280–288, Granada, Spain, 2011.

[243] Yishay Mansour and Michal Parnas. Learning conjunctions with noise under product distributions. *Information Processing Letters*, 68(4):189–196, 1998.

[244] Eleazar Eskin. Detecting errors within a corpus using anomaly detection. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 148–153, Stroudsburg, PA, USA, 2000.

[245] Neamat El Gayar, Friedhelm Schwenker, and Günther Palm. A study of the robustness of knn classifiers trained using soft labels. In *Proceedings of the Second international conference on Artificial Neural Networks in Pattern Recognition*, pages 67–80, Ulm, Germany, 2006.

[246] Umaa Rebbapragada and Carla E. Brodley. Class noise mitigation through instance weighting. In *Proceedings of the 18th European conference on Machine Learning*, pages 708–715, Berlin, Heidelberg, 2007.

[247] T. Denœux. A k-nearest neighbor classification rule based on dempster-shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5):804–813, 1995.

[248] Thierry Denœux. A neural network classifier based on dempster-shafer theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 30(2):131–150, 2000.

[249] Mahdi Tabassian, Reza Ghaderi, and Reza Ebrahimpour. Knitted fabric defect classification for uncertain labels based on dempster-shafer theory of evidence. *Expert Systems with Applications*, 38(5):5259–5267, 2011.

[250] Zoulficar Younes, Fahed abdallah, and Thierry Denœux. Evidential multi-label classification approach to learning from data with imprecise labels. In *Proceedings of the Computational intelligence for knowledge-based systems design, and 13th international conference on Information processing and management of uncertainty*, pages 119–128, 2010.

[251] P. Vannoorenberghe and T. Denœux. Handling uncertain labels in multiclass problems using belief decision trees. In *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1919–1926, 2002.

[252] Etienne Côme, Latifa Oukhellou, Thierry Denœux, and Patrice Aknin. Mixture model estimation with soft labels. In *Soft Methods for Handling Variability and Imprecision*, pages 165–174. 2008.

[253] E. Côme, L. Oukhellou, T. Denœux, and P. Aknin. Learning from partially supervised data using mixture models and belief functions. *Pattern Recognition*, 42:334–348, 2009.

[254] Benjamin Quost and Thierry Denœux. Learning from data with uncertain labels by boosting credal classifiers. In *Proceedings of the 1st ACM SIGKDD Workshop on Knowledge Discovery from Uncertain Data*, pages 38–47, Paris, France, 2009.

[255] W. Krauth and Mézard. Learning algorithms with optimal stability in neural networks. *Journal of Physics A*, 20:L745–L752, 1987.

[256] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines And Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[257] A. Kowalczyk, A. J. Smola, and R. C. Williamson. Kernel machines and boolean functions. In *Advances in Neural Information Processing Systems 14*, pages 439–446, Vancouver, British Columbia, Canada, 2001.

[258] Yi Li and Philip M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1-3):361–387, 2002.

[259] Roni Khardon and Gabriel Wachman. Noise tolerant variants of the perceptron algorithm. *Journal of Machine Learning Research*, 8:227–248, 2007.

[260] Aravind Ganapathiraju, Joseph Picone, and Mississippi State. Support vector machines for automatic data cleanup. In *Proceedings of the Sixth International Conference on Spoken Language Processing*, pages 210–213, Beijing, China, 2000.

[261] Chun-fu Lin and Sheng-de Wang. Training algorithms for fuzzy support vector machines with noisy data. *Pattern Recognition Letters*, 25(14):1647–1656, 2004.

[262] Ding-Fang Li, Wen-Chao Hu, Wei Xiong, and Jin-Bo Yang. Fuzzy relevance vector machine for learning from unbalanced data and noise. *Pattern Recognition Letters*, 29(9):1175–1181, 2008.

[263] Romer Rosales, Glenn Fung, and Wei Tong. Automatic discrimination of mislabeled training points for large margin classifiers. In *Proceedings of the Snowbird Machine Learning Workshop*, 2009.

[264] Mostafa Sabzekar, Hadi Sadoghi Yazdi, Mahmoud Naghibzadeh, and Sohrab Effati. Emphatic constraints support vector machine. *International Journal of Computer and Electrical Engineering*, 2(2):296–306, 2010.

[265] Wenjuan An and Mangui Liang. Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises. *Neurocomputing*, (0):–, 2013.

[266] Carlos Domingo and Osamu Watanabe. Madaboost: A modification of adaboost. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 180–189, San Francisco, CA, USA, 2000.

[267] Nikunj C. Oza. Boosting with averaged weight vectors. In *Proceedings of the 4th international conference on Multiple classifier systems*, pages 15–24, Berlin, Heidelberg, 2003.

[268] Nikunj C. Oza. Aveboost2: Boosting for noisy data. In *Proceedings of the 5th international conference on Multiple classifier systems*, pages 31–40, 2004.

[269] Vanessa Gómez-Verdejo, Manuel Ortega-Moral, Jerónimo Arenas-García, and Aníbal R. Figueiras-Vidal. Boosting by weighting critical and erroneous samples. *Neurocomputing*, 69(7-9):679–685, 2006.

[270] L Mason, J Baxter, P Bartlett, and M Frean. *Advances in Large Margin Classifiers*, chapter Functional gradient techniques for combining hypotheses, pages 221–246. MIT Press, Cambridge, 2000.

[271] Nir Krause and Yoram Singer. Leveraging the margin more carefully. In *Proceedings of the twenty-first international conference on Machine learning*, pages 63–70, New York, NY, USA, 2004.

[272] Linli Xu, Koby Crammer, and Dale Schuurmans. Robust support vector machine training via convex outlier ablation. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, pages 536–542, Boston, Massachusetts, USA, 2006.

[273] Hamed Masnadi-Shirazi and Nuno Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Advances in Neural Information Processing Systems 21*, pages 1049–1056, 2008.

[274] Guillaume Stempfel and Liva Ralaivola. Learning svms from sloppily labeled data. In *Proceedings of the 19th International Conference on Artificial Neural Networks: Part I*, pages 884–893, Berlin, Heidelberg, 2009.

[275] Hamed Masnadi-Shirazi, Vijay Mahadevan, and Nuno Vasconcelos. On the design of robust classifiers for computer vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 779–786, Los Alamitos, CA, USA, 2010.

[276] Yunlei Li, Lodewyk F.A. Wessels, Dick de Ridder, and Marcel J.T. Reinders. Classification in the presence of class noise using a probabilistic kernel fisher method. *Pattern Recognition*, 40(12):3349–3357, 2007.

[277] Jakramate Bootkrajang and Ata Kaban. Multi-class classification in the presence of labelling errors. In *Proceedings of the 19th European Symposium on Artificial Neural Networks*, Bruges, Belgium, 2011.

[278] Mattias Rantalainen and Chris C. Holmes. Accounting for control mislabeling in case-control biomarker studies. *Journal of Proteome Research*, 10(12):5562–5567, 2011.

[279] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of M*, 5:361–397, 2004.

[280] Shuiwang Ji and Jieping Ye. Generalized linear discriminant analysis: A unified framework and efficient model selection. *IEEE Transactions on Neural Networks*, 19(10):1768–1782, October 2008.

[281] Luis Daza and Edgar Acuna. An algorithm for detecting noise on supervised classification. In *Proceedings of the World Congress on Engineering and Computer Science 2007*, pages 701–706, San Francisco, USA, 2007.

[282] N. P. Hughes, L. Tarassenko, and S. J. Roberts. Markov models for automated ECG interval analysis. In *NIPS 2004: Proceedings of the 16th Conference on Advances in Neural Information Processing Systems*, pages 611–618, 2004.

[283] G. D. Clifford, F. Azuaje, and P. McSharry. *Advanced Methods And Tools for ECG Data Analysis*. Artech House, Inc., 2006.

[284] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[285] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[286] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[287] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[288] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[289] Murray Aitkin and Granville Tunnicliffe Wilson. Mixture models, outliers, and the em algorithm. *Technometrics*, 22(3):325–331, 1980.

[290] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 255–262, San Francisco, CA, USA, 2000.

[291] R. Dennis Cook. Influential observations in linear regression. *Journal of the American Statistical Association*, 74(365):pp. 169–174, 1979.

[292] Ali S. Hadi and Jeffrey S. Simonoff. Procedures for the identification of multiple outliers in linear models. *Journal of the American Statistical Association*, 88(424):1264–1272, 1993.

[293] Peter J Rousseeuw and Andreas Christmann. Robustness against separation and outliers in logistic regression. *Computational Statistics & Data Analysis*, 43(3):315 – 332, 2003.

[294] Hong Xu and Philippe Smets. Generating explanations for evidential reasoning. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 574–581, San Francisco, CA, USA, 1995.

[295] Cédric Archambeau, Nicolas Delannay, and Michel Verleysen. Robust probabilistic projections. In *Proceedings of the 23rd international conference on Machine learning*, pages 33–40, New York, NY, USA, 2006.

[296] Feifang Hu and James V. Zidek. The weighted likelihood. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 30(3):pp. 347–371, 2002.

[297] Wen Wen, Zhifeng Hao, and Xiaowei Yang. Robust least squares support vector machine based on recursive outlier elimination. *Soft Comput.*, 14(11):1241–1251, September 2010.

[298] Jingli Liu, Jianping Li, Weixuan Xu, and Yong Shi. A weighted lq adaptive least squares support vector machine classifiers - robust and sparse approximation. *Expert Systems with Applications*, 38(3):2253 – 2259, 2011.

[299] P. J. Huber. *Robust Statistics*. 1981.

[300] Zizhu Fan, Ergen Liu, and Baogen Xu. Weighted principal component analysis. In *AICI (3)*, pages 569–574, 2011.

[301] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):pp. 73–101, 1964.

[302] Ali S. Hadi and Alberto Luce no. Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics & Data Analysis*, 25(3):251 – 272, 1997.

[303] David Ruppert and Raymond J. Carroll. Trimmed least squares estimation in the linear model. *Journal of the American Statistical Association*, 75(372):pp. 828–838, 1980.

[304] Peter J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):pp. 871–880, 1984.

[305] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 515–521, San Francisco, CA, USA, 1998.

[306] J.A.K. Suykens. *Least Squares Support Vector Machines*. World Scientific, 2002.

[307] K R Muller, S Mika, G Ratsch, K Tsuda, and B Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[308] Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[309] P. K. Wong, C. M. Vong, C. C. Cheung, and K. I. Wong. Diesel engine modelling using extreme learning machine under scarce and exponential data sets. . *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 2013.

[310] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[311] Simon Haykin. *Neural Networks*. Pearson, 1998.

[312] Yoan Miche Elina Parviainen, Jaakko Riihimäki and Amaury Lendasse. Interpreting extreme learning machine as an approximation to an infinite neural network. In *KDIR 2010: Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, 2010.

[313] G.B. Huang, Q.Y. Zhu, and C.K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, pages 985–990, 2004.

[314] Guang-bin Huang and Chee-kheong Siew. Extreme learning machine with randomly assigned rbf kernels. *International Journal*, (Icarcv 2004):16–24, 2005.

[315] Guang-Bin Huang and Lei Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16-18):3056–3062, 2007.

[316] Guang-Bin Huang, Ming-Bin Li, Lei Chen, and Chee-Kheong Siew. Incremental extreme learning machine with fully complex hidden nodes. *Neurocomput.*, 71:576–583, 2008.

[317] Ying Liu, HanTong Loh, and ShuBeng Tor. Comparison of extreme learning machine with support vector machine for text classification. In *Innovations in Applied Artificial Intelligence*, pages 390–399. 2005.

[318] Xun-Kai Wei, Ying-Hong Li, and Yue Feng. Comparative study of extreme learning machine and support vector machine. In *Advances in Neural Networks - ISNN 2006*, pages 1089–1095. 2006.

[319] Guang-Bin Huang, Xiaojian Ding, and Hongming Zhou. Optimization method based extreme learning machine for classification. *Neurocomputing*, 74:155–163, 2010.

[320] Lipo P Wang and Chunru R Wan. Comments on the extreme learning machine. *IEEE Transactions on Neural Networks*, 19(8):1494–1495, 2008.

[321] Guang-Bin Huang. Reply to comments on the extreme learning machine. *IEEE Transactions on Neural Networks*, pages 1495–1496, 2008.

[322] Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *Neural Networks, IEEE Transactions on*, 17(4):879–892, 2006.

[323] Guang-Bin Huang and Lei Chen. Enhanced random search based incremental extreme learning machine. *Neurocomput.*, 71(16-18):3460–3468, October 2008.

[324] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse. Op-elm: Optimally pruned extreme learning machine. *Neural Networks, IEEE Transactions on*, 21(1):158–162, December 2009.

[325] Qiuge Liu, Qing He, and Zhongzhi Shi. Extreme support vector machine classifier. In *Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining*, pages 222–233, Berlin, Heidelberg, 2008.

[326] Yoan Miche, Mark van Heeswijk, Patrick Bas, Olli Simula, and Amaury Lendasse. Trop-elm: A double-regularized elm using lars and tikhonov regularization. *Neurocomputing*, 74(16):2413 – 2421, 2011.

[327] Bernhard Schölkopf and Alex J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimisation and Beyond.* The MIT Press, 2001.

[328] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[329] R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[330] M. A. Little, P. E. McSharry, E. J. Hunter, and L. O. Ramig. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 56(4):1015–1022, 2009.

[331] C. Williams. Computing with infinite networks. In *Advances in Neural Information Processing Systems 9*, pages 295–301, 1996.

[332] Chih-Chung Chang and Chih-Jen Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):2:27:1–27:27, 2011.

[333] C.-J. Hsieh X.-R. Wang R.-E. Fan, K.-W. Chang and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Part II

Publications

Chapter 6

On the Potential Inadequacy of Mutual Information for Feature Selection

The following article has been presented at the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, 25-27 April 2012. Whereas mutual information is often seen as a proxy to the risk in classification, this paper shows that is not necessarily true. Related papers about the adequacy of mutual information for feature selection include [2, 11, 12]. Reprinted with permission from [1].

On the Potential Inadequacy of Mutual Information for Feature Selection

Benoît Frénay, Gauthier Doquire and Michel Verleysen *

Université catholique de Louvain - ICTEAM/ELEN - Machine Learning Group
Place du Levant 3, 1348 Louvain-la-Neuve - Belgium

Abstract. Despite its popularity as a relevance criterion for feature selection, the mutual information can sometimes be inadequate for this task. Indeed, it is commonly accepted that a set of features maximising the mutual information with the target vector leads to a lower probability of misclassification. However, this assumption is in general not true. Justifications and illustrations of this fact are given in this paper.

1 Introduction

For a lot of machine learning and data mining applications, feature selection is a task of major importance. In particular, many regression or classification algorithms perform particularly bad when faced to high-dimensional data, due to the so-called *curse of dimensionality*. By reducing the dimensionality of the dataset while preserving the original features (by opposition to projection techniques), feature selection allows building efficient and easy-to-interpret models.

Filter methods, which are based on a statistical criterion to evaluate the relevance of a set of features, are often used in practice; this is mainly due to their low computational cost and their independence from any prediction model, in comparison to wrapper approaches which directly optimize the performances of a specific prediction model. Indeed, filter methods can be used prior to the construction of any prediction model.

As it is well-known, the mutual information (MI) [1] is a quantity measuring the dependency between two (groups of) random variables. Many reasons detailed below, including bounds relating it to the probability of classification error, made the MI criterion very popular for filter based feature selection [2]. However, despite its popularity, there exists a significant number of problems for which the MI should probably not be the criterion of choice. Indeed, the subset of features maximising the MI with a target class vector may not always minimise the probability of misclassification, which is often the final quantity of interest in real-world applications. The objective of the paper is to clearly point out and illustrate this fact. A sufficient condition for the MI criterion to be relevant for a certain problem is also given.

Section 2 briefly recalls basic notions about MI and presents the reasons why it is popular for feature selection. In Section 3, the possible inadequacy of the MI for this task is discussed, the potential problems are illustrated and a sufficient condition for optimality is given. Section 4 concludes the work.

*Gauthier Doquire is funded by a Belgian F.R.I.A grant.

2 Mutual Information

This section introduces mutual information in the context of feature selection.

2.1 Basic Definitions

Shannon's mutual information (MI) [1] measures the dependency between two discrete random variables X and Y. If X (resp. Y) takes on n_X (n_Y) possible values x_i (y_i) with probability $P(X = x_i)$ ($P(Y = y_i)$), MI is defined as

$$I(X;Y) = \sum_{i=1}^{n_X} \sum_{j=1}^{n_Y} P(X = x_i, Y = y_j) \log_2 \frac{P(X = x_i, Y = y_j)}{P(X = x_i)P(Y = y_j)} \tag{1}$$

where $P(X, Y)$ is the joint probability of X and Y. Equation (1) can be seen as the Kullback-Leibler divergence [1] between $P(X)$ $P(Y)$ and the joint probability $P(X, Y)$. Knowing that the entropy of a discrete random variable is

$$H(X) = -\sum_{i=1}^{n_x} P(X = x_i) \log_2 P(X = x_i), \tag{2}$$

it is possible to show [1] from Eq. (1) that the MI can be rewritten as

$$I(X;Y) = H(Y) - H(Y|X) \tag{3}$$

with $H(Y|X)$ being the conditional entropy of Y given X. Similar definitions can derived for continuous variables, the sums being then replaced by integrals.

2.2 Use for Feature Selection

Since the seminal paper of Battiti [2], the MI criterion has been used extensively for filter feature selection as it possesses many desirable properties for this task.

First, as detailed in [2], the MI has a natural interpretation in terms of uncertainty reduction. Indeed, it is well known that the entropy of a random variable measures the uncertainty on the values taken by this variable. Let Y be a target class vector and X a (set of) feature(s). Equation (3) translates the fact that $I(X;Y)$ is the reduction of uncertainty about the value of Y once X is known; this appears to be a natural criterion for feature selection. Equation (1) can also be interpreted in the same way. If X and Y are independent, $P(X, Y) = P(X)P(Y)$ and the MI is zero. On the contrary, as the dependency between X and Y grows so does the divergence (1) and thus the MI.

Then, it is also stressed in [2] that MI has the advantage over other popular criteria (such as the correlation coefficient) that it is able to detect non-linear relationships between variables. Moreover, the MI criterion can naturally be defined for multivariate random variables, which again is not true for correlation. This property is of fundamental importance if greedy search procedures (such as forward or backward) have to be used to construct the feature subset.

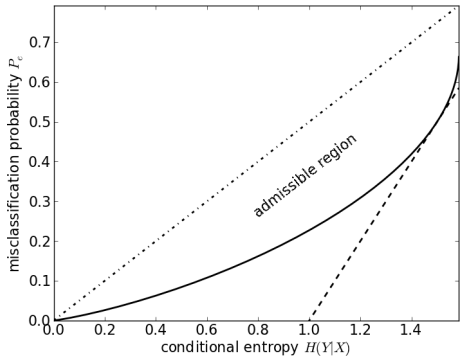


Figure 1: Weak Fano bound (dashed line), strong Fano bound (plain line) and Hellman-Raviv bound (dash-dotted line) on the probability of misclassification P_e for an optimal classifier with three balanced classes ($n_Y = 3$), with respect to the conditional entropy $H(Y|X)$. This figure is inspired by [4, 6].

Eventually, the use of MI is also supported by the existence of bounds relating the probability of misclassification P_e for an optimal classifier to the conditional entropy $H(Y|X)$. More specifically, Fano [3] derived two lower bounds on P_e. The weak Fano bound states that

$$H(Y|X) \leq 1 + P_e \log_2(n_Y - 1) \tag{4}$$

where n_Y is the number of classes, whereas the strong Fano bound is

$$H(Y|X) \leq H(P_e) + P_e \log_2(n_Y - 1). \tag{5}$$

The two above upper bounds on $H(Y|X)$ can be inverted to obtain lower bounds on P_e. It is important to notice that the weak bound (4) is useless in binary classification problems, since it cannot be inverted to get a lower bound on P_e when $n_Y = 2$. Moreover, the bound (4) is generally much looser than the bound (5), especially if P_e is small, which is precisely the situation of interest for classifier design [4]. However, the strong bound (5) on P_e is less easy to manipulate in practice since it has no closed-form and must be solved numerically. An upper bound on P_e is also given by the Hellman-Raviv inequality [5]

$$P_e \leq \frac{1}{2} H(Y|X). \tag{6}$$

As can be seen in Figure 1 inspired by [4, 6], decreasing the conditional entropy decreases both the upper and the lower bound on P_e, motivating the use of this criterion for feature selection. Since $H(Y)$ is a constant value for a given classification problem, Equations (4), (5) and (6), together with Equation (3), also give a justification to the maximisation of the MI for feature selection.

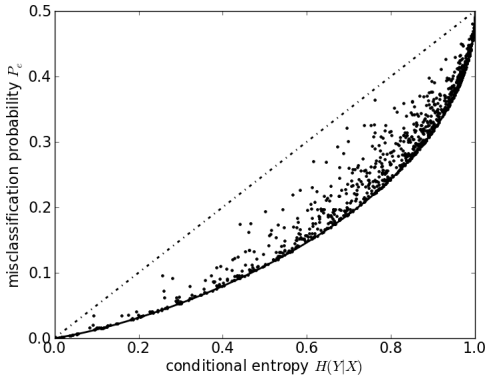


Figure 2: Examples of pairs $\langle H(Y|X), P_e \rangle$ corresponding to random binary classification problems with two binary features. The strong Fano bound (plain line) and the Hellman-Raviv bound (dash-dotted line) on P_e are shown.

3 Potential Inadequacy of Mutual Information

As mentioned in Section 2, the actual goal in many applications is to minimise the probability of misclassification. In other words, the utility of a feature subset can be quantified using P_e, which is a tight lower bound for the misclassification probability of any classifier. Using Figure 1 and Equations (3), (4), (5) and (6), several papers [4, 6, 7] conclude (i) that minimising MI is equivalent to minimising P_e and (ii) that MI can therefore be used equivalently for feature selection. This section shows that both claims are not necessarily true.

3.1 Relationships Between Misclassification Probability and Entropy

Figure 2 shows (i) the strong Fano bound and the Hellman-Raviv bound for P_e in terms of $H(Y|X)$ and (ii) several examples of actual pairs $\langle H(Y|X), P_e \rangle$. The pairs correspond to random binary classification problems with two binary features. For each problem, both P_e and $H(Y|X)$ are computed exactly, which is possible since all necessary probabilities are known. The problems are drawn as follows: (i) the values $P(Y = y)$ and $P(X = x|Y = y)$ are randomly drawn from the uniform distribution $\mathcal{U}(0, 1)$, (ii) these values are normalised to enforce $\sum_y P(Y = y) = 1$ and $\sum_x P(X = x|Y = y) = 1$ for each y and (iii) probabilities $P(X)$ and $P(Y|X)$ are computed using marginalisation and Bayes' theorem.

Figure 2 shows that the pairs $\langle H(Y|X), P_e \rangle$ are scattered between the strong Fano lower bound and the Hellman-Raviv upper bound. Moreover, it is possible to find two pairs such that the entropy $H(Y|X)$ decreases and the probability of misclassification P_e increases (and *vice versa*). In other words, contrary to what is often claimed, it is not sufficient to reduce $H(Y|X)$ in order to reduce P_e. It suggests that minimising MI may not be sufficient, which is illustrated below.

3.2 Illustration of Mutual Information Failure for Feature Selection

Let us now review a simple, artificial example of mutual information failure. In a context of disease diagnosis, two classes are distinguished with prior

$$P(Y) = \begin{pmatrix} 0.316 & 0.684 \end{pmatrix} \tag{7}$$

where columns correspond to possible values of $Y \in \{0, 1\}$. Furthermore, two tests are available to classify a new patient, whose binary outcomes are denoted $X_1 \in \{0, 1\}$ and $X_2 \in \{0, 1\}$. However, the practician can only perform one of these tests. In terms of feature selection, he has to select the best feature.

Through experimentation, the practician discovers that the conditional distributions of X_1 and X_2 with respect to Y are

$$P(X_1|Y) = \begin{pmatrix} 0.417 & 0.104 \\ 0.583 & 0.896 \end{pmatrix} \quad \text{and} \quad P(X_2|Y) = \begin{pmatrix} 0.991 & 0.479 \\ 0.009 & 0.521 \end{pmatrix} \tag{8}$$

where rows correspond to values of X_i and columns correspond to values of Y. Hence, using marginalisation and Bayes' theorem, one obtains the posteriors

$$P(Y|X_1) = \begin{pmatrix} 0.649 & 0.231 \\ 0.351 & 0.769 \end{pmatrix} \quad \text{and} \quad P(Y|X_2) = \begin{pmatrix} 0.489 & 0.008 \\ 0.511 & 0.992 \end{pmatrix} \tag{9}$$

where rows correspond to values of Y and columns correspond to values of X_i. On one hand, the test with outcome X_1 allows discriminating between both classes, but there is an important error probability ($P_e = .351$ if $X_1 = 0$ and $P_e = .231$ if $X_1 = 1$). On the other hand, the test with outcome X_2 allows discriminating almost perfectly when it is positive ($P_e = .008$ if $X_2 = 1$), but it is almost useless when it is negative ($P_e = .489$ if $X_2 = 0$).

Using the first test, one obtains $P_e = 0.255$ and $I(X_1; Y) = 0.089$. Using the second test, one obtains $P_e = 0.316$ and $I(X_2; Y) = 0.236$. Here, the MI is significantly larger using X_2. However, P_e is also larger, which means that selecting X_2 based on mutual information leads here to an increase in error. This phenomenon is not rare: using pairs of random problems drawn as explained in the previous subsection, about 20% of the pairs violate the common belief that increasing mutual information decreases the misclassification probability.

Figure 3 illustrates the example. Each pair $\langle H(Y|X), P_e \rangle$ stands between the Fano and Hellman-Raviv bounds. It is clear that $I(X_2; Y) = H(Y) - H(Y|X_2)$ is larger than $I(X_2; Y) = H(Y) - H(Y|X_2)$, whereas $P_e(X_2)$ is larger than $P_e(X_1)$.

3.3 Conditions of Optimality

According to the above discussion, mutual information seems to be more a heuristic than a never-failing criterion. However, it is possible to guarantee whether MI is valuable or not. Indeed, let us define two feature subsets $\mathcal{X}_1$ and $\mathcal{X}_2$ which must be compared. If the value of the Hellman-Raviv bound for $\mathcal{X}_1$ is smaller than the value of the strong Fano bound for $\mathcal{X}_2$, then an increase in mutual information always leads to a decrease in misclassification probability.

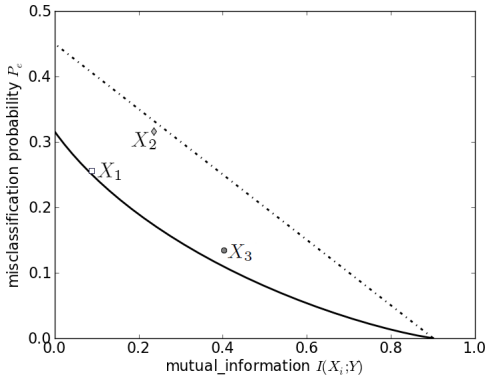


Figure 3: Example of mutual information failure for feature selection, with the strong Fano bound (plain line) and the Hellman-Raviv bound (dash-dotted line).

In the above example, the Fano bound for X_1 is $P_e \geq .250$, whereas the Hellman-Raviv bound for X_2 is $P_e \leq .332$. Here, it is not possible to guarantee that mutual information is a relevant criterion to choose between X_1 and X_2. Figure 3 also shows an other candidate X_3 for which the Hellman-Raviv bound is $P_e \leq .249$. Here, the new feature X_3 is guaranteed to be a better choice.

4 Conclusion

This paper shows that mutual information is not necessarily an optimal criterion to select features, if the actual goal is to achieve minimal probability of misclassification. The behaviour of mutual information is described and related to Fano and Hellman-Raviv bounds. An example of MI failure is given, which shows that increasing MI can sometimes increase the misclassification probability as well.

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 99th edition, August 1991.

[2] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550, 1994.

[3] R. Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.

[4] J. W. Fisher, M. Siracusa, and T. Kihn. Estimation of signal information content for classification. In *Proceedings of DSP/SPE 2009*, 2009.

[5] M. E. Hellman and J. Raviv. Probability of error, equivocation and the chernoff bound. *IEEE Transactions on Information Theory*, 16:368–372, 1970.

[6] G. Brown. An information theoretic perspective on multiple classifier systems. In *Proceedings of MCS 2009*, pages 344–353, Berlin, Heidelberg, 2009. Springer-Verlag.

[7] 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.

Chapter 7

Theoretical and Empirical Study on the Potential Inadequacy of Mutual Information for Feature Selection in Classification

The following article has been published in Volume 112 (2013) of the Neurocomputing journal and is an extended version of [1]. The paper illustrates the potential inadequacy of mutual information in feature selection. More importantly, through extensive experiments, it also confirms the general interest of the mutual information for feature selection and helps to better apprehend the behaviour of mutual information in order to make a better use of it. Related papers about the adequacy of mutual information for feature selection include [1, 11, 12]. Reprinted with permission from [2].

Neurocomputing 112 (2013) 64–78

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification

Benoît Frénay *, Gauthier Doquire [1], Michel Verleysen

Machine Learning Group—ICTEAM, Université catholique de Louvain, Place du Levant 3, 1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Available online 7 March 2013

Keywords:
Mutual information
Feature selection
Classification
Probability of misclassification
Hellman–Raviv and Fano bounds

ABSTRACT

Mutual information is a widely used performance criterion for filter feature selection. However, despite its popularity and its appealing properties, mutual information is not always the most appropriate criterion. Indeed, contrary to what is sometimes hypothesized in the literature, looking for a feature subset maximizing the mutual information does not always guarantee to decrease the misclassification probability, which is often the objective one is interested in. The first objective of this paper is thus to clearly illustrate this potential inadequacy and to emphasize the fact that the mutual information remains a heuristic, coming with no guarantee in terms of classification accuracy. Through extensive experiments, a deeper analysis of the cases for which the mutual information is not a suitable criterion is then conducted. This analysis allows us to confirm the general interest of the mutual information for feature selection. It also helps us better apprehending the behaviour of mutual information throughout a feature selection process and consequently making a better use of it as a feature selection criterion.

1. Introduction

Feature selection is known to be a preprocessing technique of fundamental importance for many applications in machine learning, pattern recognition or data mining. Indeed, dealing with high-dimensional data is a particularly hard task, in practice, due to many problems and counter-intuitive phenomena such as the empty space phenomenon and the concentration of distances [1,2]. Reducing the dimensionality of the datasets to a relatively low number of features is thus often necessary if one wants to build, for instance, efficient classification models. While efficient projection techniques can be used for dimensionality reduction, feature selection has the advantage of preserving the original features, which makes it possible to build easily interpretable models. Such an interpretability is highly appreciated, for instance, in the industrial and medical areas.

Among the various approaches to feature selection, filter methods are very popular and often used in practice. Filter methods are based on a relevance criterion independent of any classification model. They are thus easy to use and generally exhibit a low computational cost, especially when compared with wrapper methods which try to directly maximize the performances of a given prediction model. Filter methods have the additional advantage of being more general than wrapper or embedded methods, which perform simultaneously feature selection and prediction, in the sense that filters can be used in combination with any prediction model. The reader interested in feature selection is referred to [3] for a nice overview of this topic.

Since the seminal work of Battiti [4], the mutual information [5] has become one of the most widely used criteria for feature selection; see for example the following works [6–8]. In [6,7], the authors try to determine a set of maximally informative features which are mutually as non-redundant as possible, using the maximum relevance minimum redundancy principle and the conditional mutual information, respectively. In [8], a forward/backward search procedure is used to find the most relevant variables in spectroscopic modelling.

Besides performing often well in practice, the mutual information possesses other properties, detailed later in the paper, making it particularly well-suited for the feature selection task. These properties include the existence of bounds relating the mutual information to the probability of classification error. However, for a certain number of classification problems, mutual information is not the most appropriate choice of relevance criterion. Indeed, despite what is sometimes hypothesized, choosing a subset of features maximising the mutual information is not always equivalent to choosing a subset of features minimizing the

* Corresponding author. Tel.: +32 10 47 81 33; fax: +32 10 47 25 98.
E-mail addresses: benoit.frenay@uclouvain.be (B. Frénay), gauthier.doquire@uclouvain.be (G. Doquire), michel.verleysen@uclouvain.be (M. Verleysen).
[1] The author is funded by a Belgian F.R.I.A. Grant.

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.12.051

misclassification probability, which is generally the quantity one is eventually interested in.

The first objective of this paper is thus to clearly point out this fact, by illustrating it through an intuitive example. Moreover, this work also aims at characterizing the problems for which the mutual information criterion is likely to fail, and in this case to which extend the loss in misclassification probability is important. To this end, extensive experiments have been carried out on both continuous and categorical datasets, either artificially generated or corresponding to real-world problems. The idea is to eventually assess the potential interest of the mutual information as a feature selection criterion, despite its non-optimality regarding the misclassification probability. This work extends preliminary results presented in [9]. Balanced datasets are considered and new experiments are conducted to gain a better insight on the behaviour of mutual information. A forward feature selection procedure is also analysed while only pairwise comparisons between features were considered in [9].

The rest of the paper is organised as follows. Section 2 briefly recalls basic definitions about the mutual information and details some of the reasons of its popularity for feature selection. Section 3 discusses and illustrates the potential inadequacy of the mutual information for feature selection; a problem for which mutual information is not appropriate is presented and a simple sufficient condition for its optimality is given. Sections 4–6 present the experimental results for artificial datasets with discrete features, artificial datasets with continuous features and real-world datasets with continuous features, respectively. Section 7 summarises the observations drawn from the experiments and Section 8 concludes the work.

2. Mutual information

The aim of this section is to remind fundamental notions about mutual information and to justify its interest for feature selection in classification problems.

2.1. Formal definitions

Shannon's mutual information [10,5] is a measure of the dependency existing between two random variables X and Y, considered to be discrete in this section. Let us assume that X (resp. Y) can take n_X (n_Y) possible different values x_i (y_i), each with probability $P_X(X=x_i)$ ($P_Y(Y=y_i)$). The mutual information is then defined as

$$I(X;Y) = \sum_{i=1}^{n_X}\sum_{j=1}^{n_Y} P_{XY}(X=x_i,Y=y_j) \times \log_2 \frac{P_{XY}(X=x_i,Y=y_j)}{P_X(X=x_i)P_Y(Y=y_j)} \quad (1)$$

where $P_{XY}(X,Y)$ is the joint probability of the X and Y variables. Eq. (1) actually defines the Kullback–Leibler divergence [5] between the product of the two distributions $P_X(X) \times P_Y(Y)$ and the joint probability $P_{XY}(X,Y)$. As can be deducted from Eq. (1), the mutual information is a symmetric criterion, i.e. $I(X;Y) = I(Y;X)$.

Since the entropy of a discrete random variable X is defined as

$$H(X) = -\sum_{i=1}^{n_x} P_X(X=x_i)\log_2 P_X(X=x_i), \quad (2)$$

it can be shown [5] from Eq. (1) that the mutual information can be equivalently rewritten as

$$I(X;Y) = H(Y) - H(Y|X) \quad (3)$$

with

$$H(Y|X) = \sum_{i=1}^{n_X}\sum_{j=1}^{n_Y} P_{XY}(X=x_i,Y=y_j) \times \log_2 \frac{P_X(X=x_i)}{P_{XY}(X=x_i,Y=Y_j)} \quad (4)$$

being the conditional entropy of Y once X is given. While the developments have been presented for discrete variables, similar definitions can as well be derived for continuous random variables. In this case, the sums are then replaced by integrals.

2.2. Interest for feature selection

Since the work of Battiti [4], the mutual information criterion has been used extensively for filter feature selection because of many desirable properties it possesses for this task.

The first important property of mutual information, as detailed in [4], is its natural interpretation in terms of uncertainty reduction. Indeed, the entropy is a measure of the uncertainty on the values taken by a random variable. Consequently, if Y denotes a target class vector and X is a (set of) feature(s), Eq. (3) shows that $I(X;Y)$ can be interpreted as the reduction of uncertainty about the value of Y once X is known. In this regard, mutual information is thus a quite intuitive criterion to maximize for a feature subset to be considered as good. If there is no dependency between X and Y, then $H(Y) = H(Y|X)$ and $I(X;Y) = 0$. Similarly in Eq. (1), if X and Y are independent, $P_{XY}(X,Y) = P_X(X)P_Y(Y)$ and again $I(X;Y) = 0$. On the contrary, if $Y = f(X)$, then the mutual information is maximal and $I(X;Y) = H(Y)$.

The second main advantage of the mutual information, as also stressed in [4], is that it is able to measure non-linear relationships between variables. Other criteria, such as the correlation coefficient, are limited to the detection of linear dependencies. The ability to detect non-linear dependencies is obviously a strong advantage since many of the most popular classification algorithms, such as support vector machines (with a non-linear kernel) and k-nearest-neighbors, are effectively able to model non-linear relationships between the features and the class label.

In addition, the mutual information criterion can naturally be defined for multivariate random variables (and thus for subsets of features), which is not true e.g. for the correlation coefficient. This is a property of major importance since greedy search procedures (such as forward, backward and forward/backward) are often used in practice to build a feature subset. This is because, in some situations, some features are only relevant or redundant when considered together. For example, in the well-known XOR problem, both features individually do not contain any information about the output, but together completely determine it. For such problem, a univariate criterion will never be able to detect any of the two features as relevant.

Finally, the use of mutual information for feature selection in classification problems is supported by the existence of bounds relating the misclassification probability P_e for an optimal classifier that achieves the Bayes risk to the conditional entropy $H(Y|X)$, where Y is again the class label and X the feature subset. Firstly, Fano [11] derived two lower bounds on P_e. The weaker bound is

$$H(Y|X) \leq 1 + P_e \log_2(n_Y - 1) \quad (5)$$

where n_Y is the number of possible classes. The stronger bound states that

$$H(Y|X) \leq H(P_e) + P_e \log_2(n_Y - 1), \quad (6)$$

where $H(P_e) = -P_e \log_2 P_e - (1-P_e)\log_2(1-P_e)$ [5].

Both Eqs. (5) and (6) give bounds on $H(Y|X)$, but they can be inverted to provide a lower bound on P_e. However, the stronger

bound is less easy to manipulate in practice than the weaker bound because P_e cannot be isolated in Eq. (5) in a closed form, the bound on P_e has thus to be computed numerically. Nevertheless, the stronger Fano bound in practice is much more useful than the weak one. Indeed, the weak bound (5) does not apply to binary classification problems, since in this case, Eq. (5) trivially reduces to $H(Y|X) \leq 1$. Eq. (5) cannot thus be inverted to get a lower bound on P_e when $n_Y=2$. Moreover, the weak bound is generally much looser than the strong one. This is particularly true when P_e is small, which is however precisely the situation of interest for classifier design [12].

The probability of misclassification P_e can also be upper-bounded by the Hellman–Raviv inequality [13]

$$P_e \leq \tfrac{1}{2} H(Y|X). \tag{7}$$

Fig. 1, inspired from [12,14], illustrates the three bounds introduced in Eqs. (5)–(7). As can be seen, decreasing the conditional entropy $H(Y|X)$ obviously decreases both the upper and the lower bound on P_e, which motivates the use of this criterion for feature selection. Notice that $H(Y)$ is a constant value for a given classification problem since it depends only on the class labels and not on the selected features. According to Eq. (3), maximizing the mutual information $I(X;Y)$ is thus equivalent to minimizing the conditional entropy $H(Y|X)$ in this context. Eqs. (5)–(7) give a justification to the maximisation of the mutual information for feature selection.

Notice that the upper bound (7) on P_e is an increasing concave, since it is linear with respect to $H(Y|X)$. Also, the lower bound (6) on P_e (as well as its weak form (5) which is linear with respect to $H(Y|X)$) is an increasing convex, since the converse upper bound on $H(Y|X)$

$$H(P_e) + P_e \log_2(n_Y - 1) \geq H(Y|X) \tag{8}$$

is increasing concave with respect to P_e. Indeed, it can easily be shown that its first-order derivative

$$-\log_2 P_e + \log_2(1-P_e) + \log_2(n_Y - 1) \tag{9}$$

is positive and that its second-order derivative

$$\frac{-\log_2 e}{P_e(1-P_e)} \tag{10}$$

is negative when $P_e \leq ((n_Y-1)/n_Y)$, which is the case since P_e is the misclassification probability for an optimal classifier. These properties are used in the next section.

3. Potential inadequacy of mutual information

As mentioned in Section 1, the actual objective of feature selection is often to reduce as much as possible the probability of misclassification of a model built on the selected feature subset. In other words, the quality and the utility of a feature subset can be measured through P_e, which actually gives a lower bound for the misclassification probability of any (suboptimal) classification model. Based on the convex lower bound and the concave upper bound in Fig. 1 and Eq. (3), several papers, e.g. [12,14,15], claim that a feature subset having a higher mutual information with the output than another one will lead to a smaller probability of misclassification P_e. Those papers conclude that the mutual information can therefore be used as a proxy for P_e in a feature selection context. The objective of this section is to show that such a conclusion is not always valid in practice. A simple condition for the optimality of the mutual information as a feature selection criterion is also given. Eventually, we also derive a bound relating (i) the maximum value of mutual information between two feature subsets and the output to (ii) the loss in misclassification probability induced by the selection of one subset instead of the other one.

3.1. Relationship between misclassification probability and conditional entropy

In Fig. 2, the strong Fano bound and the Hellman–Raviv bound for P_e in terms of $H(Y|X)$ are again illustrated. Moreover, the figure also shows many examples of $\langle H(Y|X), P_e \rangle$ couples of values. Each point in Fig. 2 corresponds to a different random binary classification problem with two binary features. The problems are generated as follows: (i) the two values $P(Y=y)$ (for $y \in [0,1]$) and the four values $P(X=x|Y=y)$ (for $x \in [0,1]$ and $y \in [0,1]$) are randomly drawn from the uniform distribution $\mathcal{U}(0,1)$, (ii) these values are normalised to ensure that they represent probabilities, i.e. $\sum_y P(Y=y)=1$ and $\sum_x P(X=x|Y=y)=1$ for each y and (iii) probabilities $P(X)$ and $P(Y|X)$ are eventually computed using marginalisation and the Bayes' theorem. For each problem, it is thus possible to compute exactly both P_e and $H(Y|X)$ because all the necessary probabilities are known.

As expected, the couples $\langle H(Y|X), P_e \rangle$ all lie in the area defined by the strong Fano lower bound and the Hellman–Raviv upper bound. Given the value of the mutual information $I(X;Y)$, or equivalently the value of the conditional entropy $H(Y|X)$, the two bounds thus define an interval where P_e belongs. Obviously, given two different values of conditional entropy, the intervals for

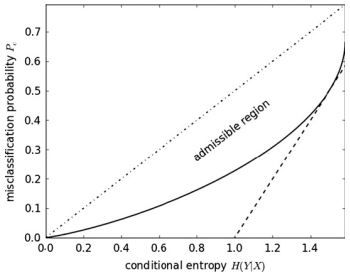


Fig. 1. Weak Fano bound (dashed line), strong Fano bound (plain line) and Hellman–Raviv bound (dash-dotted line) on the misclassification probability P_e of an optimal classifier with three classes ($n_Y=3$), in terms of the conditional entropy $H(Y|X)$; figure inspired from [12,14], reprinted with permission from [9].

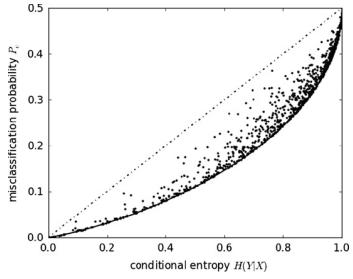


Fig. 2. Several pairs of $\langle H(Y|X), P_e \rangle$ values corresponding to random binary classification problems with two binary features. The strong Fano bound (plain line) and the Hellman–Raviv bound (dash-dotted line) relating P_e to $H(Y|X)$ are shown. From [9], reprinted with permission.

P_e defined by the bounds could strongly overlap. Therefore, given two subsets of features $\mathcal{X}_1$ and $\mathcal{X}_2$ such that $H(Y|\mathcal{X}_1) < H(Y|\mathcal{X}_2)$, it could theoretically be possible that $\mathcal{X}_1$ leads to a higher probability of misclassification P_e than $\mathcal{X}_2$. Fig. 2 illustrates the fact that, in practice, this situation could actually happen. Indeed, even if the pairs $\langle H(Y|X), P_e \rangle$ mainly lie near the lower bound, they scatter the whole area between the two bounds; for a given conditional entropy, actual values of misclassification probability can thus be obtained in the whole interval defined by the bounds. Consequently, choosing between two feature sets based on the mutual information criterion could not be optimal (in terms of misclassification probability), as shown through a simple example in Section 3.2.

3.2. Illustration of mutual information failure for feature selection

A simple example is now presented, to illustrate the potential inadequacy of the mutual information in a feature selection context. Let us consider a disease diagnosis, where two classes have the same prior probability

$$P(Y) = (0.5 \;\; 0.5). \tag{11}$$

In (11), each column corresponds to one of the two possible values of $Y \in \{0,1\}$. Let us further assume that the results of two different tests are available to help classifying a new patient. Both tests are binary and their outcomes are denoted as $X_1 \in \{0,1\}$ and $X_2 \in \{0,1\}$. For some practical reasons, the practician can only perform one of those two tests. This choice is clearly a feature selection problem, each test corresponding to a feature and the practician having to chose the best test.

Through previous experimentation, the practician is able to establish that the conditional distributions $P(X_i|Y)$ of both tests X_1 and X_2 given Y are given by

	$Y=0$	$Y=1$
$X_1=0$	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

.

The rows correspond to the possible values of X_i and the columns again correspond to the values of Y. Using marginalisation and the Bayes' theorem, it is straightforward to obtain the posteriors $P(Y|X_i)$ given by

	$X_1=0$	$X_1=1$
$Y=0$	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

.

Again, rows correspond to values of Y and columns correspond to values of X_i. It can be understood from the last two probability tables that the test whose outcome is X_1 allows discriminating fairly well between the classes, whatever its output is. There remains however a quite important misclassification probability using X_1 (P_e=0.275 if $X_1=0$ and P_e=0.254 if $X_1=1$). The second test, with the outcome X_2, allows discriminating almost perfectly when it is positive (P_e=0.001 if $X_2=1$). When it is negative ($X_2=0$), it is however much less discriminative than the first test since the misclassification probability is P_e=0.385 in that case.

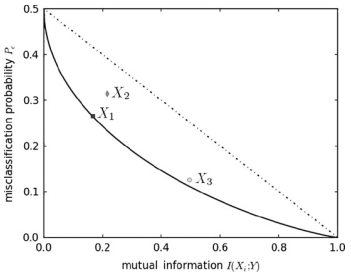


Fig. 3. Example of mutual information failure for feature selection, with the strong Fano bound (plain line) and the Hellman–Raviv bound (dash-dotted line).

When the results of the first tests are used to select the feature, one eventually obtains a global misclassification probability of P_e=0.265 while $I(X_1;Y) = 0.167$. Using the second test, one obtains P_e=0.314 and $I(X_2;Y) = 0.217$. Here, it appears that the mutual information is larger using X_2. However, P_e is smaller when X_1 is used, meaning that selecting X_2 based on mutual information leads here to an increased probability of misclassification.

The above example is illustrated in Fig. 3, where each point $\langle H(Y|X), P_e \rangle$ is again shown to lie between the Fano and Hellman–Raviv bounds. Obviously, $I(X_2;Y) = H(Y) - H(Y|X_2)$ is larger than $I(X_1;Y) = H(Y) - H(Y|X_1)$ while $P_e(X_2)$ is simultaneously larger than $P_e(X_1)$.

3.3. A condition of optimality

As illustrated by the previous example, and as shown in the following sections, the mutual information appears to be a heuristic with no obvious way to assess its potential interest. However, in some situations, it is possible to guarantee that the mutual information is actually an adequate criterion. Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two features sets that have to be compared. If the value of the Hellman–Raviv bound for $\mathcal{X}_1$ is smaller than the value of the strong Fano bound for $\mathcal{X}_2$, then it can be deduced from the bounds in Fig. 2 that the feature set $\mathcal{X}_1$ leads to a smaller misclassification probability P_e than the feature set $\mathcal{X}_2$ does. Thus, if the values of the conditional entropies for two subsets are different enough, the corresponding possible intervals for P_e cannot overlap and ranking feature subsets with the mutual information criterion is optimal.

In the above example, the Fano bound for X_1 is $P_e \geq 0.264$, whereas the Hellman–Raviv bound for X_2 is $P_e \leq 0.391$; it is not possible to guarantee that mutual information is a relevant criterion to choose between X_1 and X_2. Fig. 3 also shows another candidate X_3 for which the Hellman–Raviv bound is $P_e \leq 0.252$. In this case, the new feature X_3 is guaranteed to be a better choice.

3.4. Upper bound on the misclassification probability loss

It is also possible to give an upper bound for the difference in misclassification probability in case of failure, i.e. the supplementary percentage of samples which are misclassified due to an incorrect choice of feature subset only. This difference is called the misclassification probability loss in the following of the paper. Indeed, the worst case of mutual information failure occurs when (i) both feature subsets have almost identical mutual information, (ii) the selected feature subset stands on the Hellman–Raviv bound (maximum misclassification probability) and (iii) the other

feature subset stands on the strong Fano bound (minimum misclassification probability). In such a case, the misclassification probability loss is simply the difference between the Hellman–Raviv bound and the strong Fano bound. The upper bound on the misclassification probability loss is concave with respect to $I(X;Y)$, since the Hellman–Raviv and Fano bounds are increasing concave and convex with respect to $H(Y|X)$, respectively. Fig. 4 shows the upper bound on the misclassification probability loss for the above example. Here, the misclassification probability loss is bounded by 0.159 for the selected feature X_2, whereas the actual misclassification probability loss is 0.049. Interestingly, the maximum misclassification probability loss decreases for extreme (small or large) values of the mutual information. It suggests that mutual information failures have less important consequences in these cases.

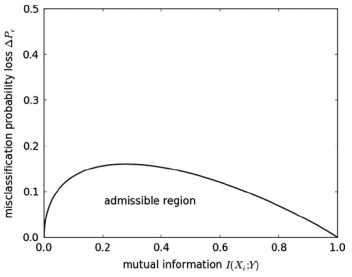


Fig. 4. Theoretical upper bound on the misclassification probability loss for binary classification with balanced classes.

4. Artificial classification problems with discrete features

This section discusses the use of mutual information for feature selection using three simple artificial monovariate binary classification problems. The input of the classifier is a discrete feature with 2^d possible modalities. This may be viewed as equivalent to a binary classification problem with d binary features.

4.1. Experimental settings

The three artificial problems discussed in this section are designed to simulate low, medium and high levels of difficulty in binary classification. This is achieved by choosing different

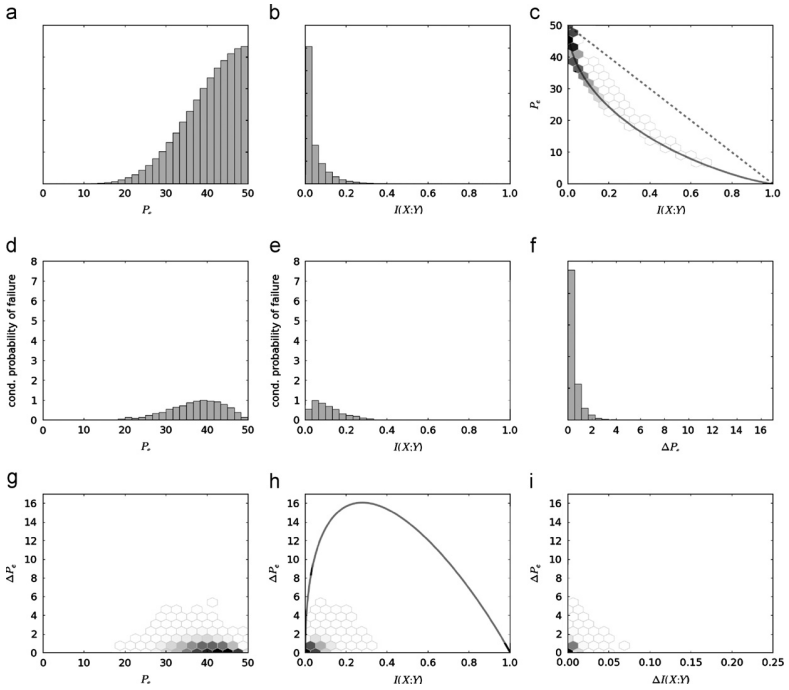


Fig. 5. Results for artificial binary classification problems with a 2-value discrete feature and a Dirichlet prior with $\alpha=4$ for the conditional probabilities $P(X=x_i|Y)$.

domain sizes for the discrete feature X and different prior distributions for its conditional probabilities $P(X|Y)$. For each of the three problems, a large number of pairs of possible features are generated, which are compared pairwise to assess whether mutual information is consistent with the misclassification probability. Notice that the classifier remains univariate: the possible features are compared by pairs, to decide in each pair which feature will be used as input to the classifier. The conditional probabilities $P(X=x_i|Y)$ of each feature are drawn from a symmetric Dirichlet distribution

$$f(x_1,\ldots,x_m|\alpha)=\Gamma(\alpha m)\prod_{i=1}^{m}\frac{x_i^{\alpha-1}}{\Gamma(\alpha)} \tag{12}$$

where x_i is the ith modality of feature X, Γ is the gamma function and α is the concentration parameter. Conditional probabilities $P(X|Y)$ are drawn instead of conditional probabilities $P(Y|X)$ because it allows us to keep constant the class prior $P(Y)$. Indeed, the proportion of instances in each class should not depend on the feature which is used to classify them. Moreover, this is necessary to compute the bounds which are visualised in the figures below. Large values of α correspond to conditional distributions of X given Y where almost all probabilities are equal, whereas only one probability is non-zero for small values of α. In other words, class discrimination is expected to be easier with small values of α. In addition, classes are usually easier to discriminate in high-dimensional spaces. By choosing the problem parameters $\langle m=2,\alpha=4\rangle$, $\langle m=8,\alpha=1\rangle$ and $\langle m=128,\alpha=0.06\rangle$, three families of problems are obtained with high, medium and low levels of difficulty, respectively. The class prior is uniform, i.e. $P(Y=y)=\frac{1}{2}$ for each y, in order to avoid class imbalance effects.

For each set of binary classification problem parameters, 10^6 pairs of features are generated. For each pair, the features are compared in terms of mutual information with the class Y and misclassification probability, which can be computed exactly since all the required probabilities are known. The feature with the largest mutual information is chosen. If the misclassification probability is also larger for the chosen feature, the pair is an example of failure for mutual information as a feature selection criterion. In case of failure, the difference in misclassification probability is called the misclassification probability loss ΔP_e, i.e. the percentage of samples which are misclassified due to an incorrect choice of feature. The average and conditional probabilities of failure can be estimated by counting failures among the pairs.

Each artificial problem is illustrated in Figs. 5–7, respectively. Mutual information is computed in base 2, whereas all probabilities are given in percents. The first row consists of a histogram of

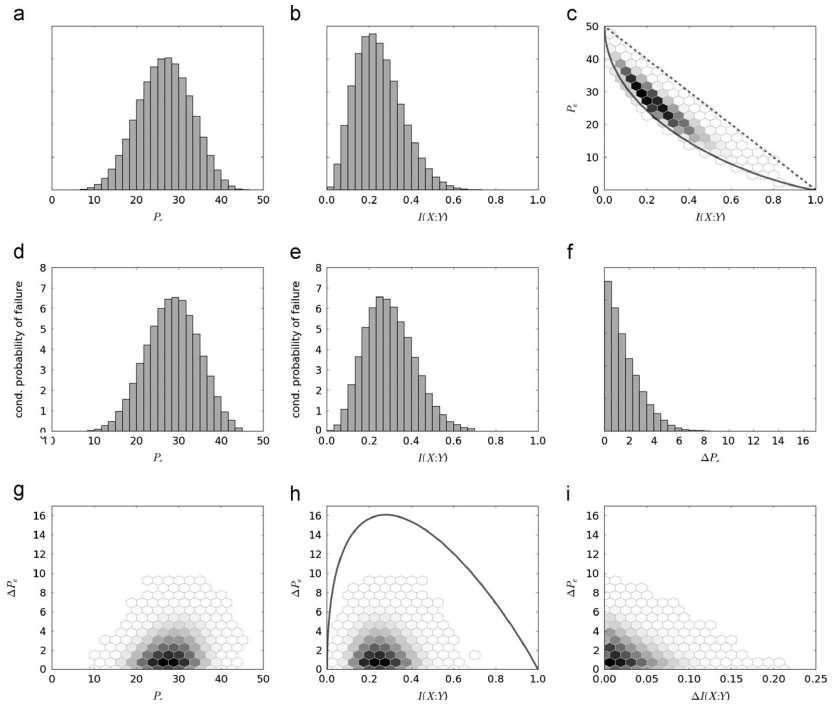


Fig. 6. Results for artificial binary classification problems with a 8-value discrete feature and a Dirichlet prior with $\alpha=1$ for the conditional probabilities $P(X=x_i|Y)$.

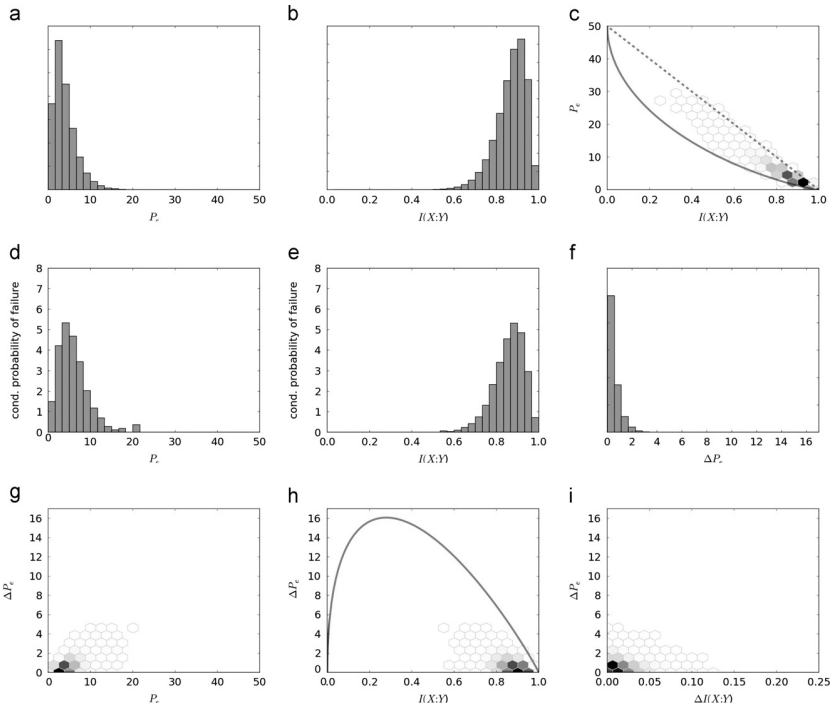


Fig. 7. Results for artificial binary classification problems with a 128-value discrete feature and a Dirichlet prior with $\alpha = 0.06$ for the conditional probabilities $P(X = x_i|Y)$.

the misclassification probability, a histogram of the mutual information and a two-dimensional histogram (with hexagonal bins whose opacity indicates the number of samples in each bin) of these two quantities, with the Fano and Hellman–Raviv bounds. The second row shows an estimate of the conditional probability of failure given the misclassification probability and given the mutual information, and a histogram of the misclassification probability loss in case of failure. Eventually, for failures, the last row shows two-dimensional histograms of (i) the misclassification probability and the misclassification probability loss, (ii) the mutual information and the misclassification probability loss (with the theoretical bound derived in Section 3.4) and (iii) the mutual information difference and the misclassification probability loss. In the last two rows, which correspond to mutual information failures, the mutual information and the misclassification probability are those of the feature which is selected using mutual information. Indeed, what we are mainly interested in is to know when a feature selected by mutual information is likely to be a bad choice.

4.2. Results

For $m=2$ and $\alpha = 4$, classes are very difficult to discriminate, as seen in Fig. 5(a) and (b), but only 0.6% of the pairs are failures. The conditional probability of failure remains small in Fig. 5(d) and (e), where the failure probability decreases for small mutual information values and large misclassification probabilities. Fig. 5(f) shows that 95% of the misclassification probability losses remain below 1.5%. In Fig. 5(g) and (h), the misclassification probability loss decreases for small mutual information values and large misclassification probabilities. Eventually, Fig. 5(i) shows that failures 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 moderately difficult to discriminate, as seen in Fig. 6(a) and (b). The percentage of failure is 4.6, what is higher than in the $m=2$ and $\alpha = 4$ case. The conditional probability of failure is also larger in Fig. 6(d) and (e); Fig. 6(f) shows that the misclassification probability loss is larger, but remains below 4.3% in 95% of the failures. Fig. 6(g)–(i) leads to similar conclusions than with $m=2$ and $\alpha = 4$.

For $m=128$ and $\alpha = 0.06$, classes are quite easy to discriminate, as seen in Fig. 7(a) and (b). The percentage of failure is 3.8 and the conditional failure probability decreases for large mutual information values and small misclassification probabilities. Similarly, Figs. 7(g) and (h) show that the misclassification probability loss decreases for large mutual information values and small misclassification probabilities. In Fig. 7(f), 95% of the misclassification probability losses remain below 1.5%.

4.3. Discussion

In the above experiments, mutual information fails to select the feature with the best misclassification probability in only a few percents of the cases. Moreover, such failures do not lead to large misclassification probability losses, which means that the consequences of the failures are not too important. Failures appear to be more probable when classes are moderately difficult to discriminate, i.e. for intermediate values of mutual information and misclassification probability. In such cases, the misclassification probability loss is also larger. For all problems, failures mostly occur for pairs of features which are close in terms of both mutual information and misclassification probability.

5. Artificial classification problems with continuous features

This section discusses the use of mutual information for feature selection using three simple artificial three-class classification problems with a single continuous feature.

5.1. Experimental settings

Similar to Section 4, the three artificial problems discussed in this section are designed to simulate low, medium and high levels of difficulty in three-class classification. For each of the three problems, a large number of features are generated, which are compared pair-wise to assess whether mutual information is consistent with the misclassification probability. For each class, each feature has a unidimensional Gaussian conditional distribution. The standard deviations of the feature values are randomly drawn from a gamma distribution

$$f(\sigma|k,\theta) = \frac{\sigma^{k-1}}{\theta^k \Gamma(k)} e^{-\sigma/\theta}, \tag{13}$$

where k is the shape parameter and θ is the scale parameter. In the experiments, the parameter values $k=2$ and $\theta=0.5$ are used in order to obtain realistic and diversified standard deviations. The means of the feature values in the three classes are $\mu_0=-\Delta$, $\mu_1=0$ and $\mu_2=\Delta$, where Δ is a parameter which determines the difficulty of the classification problem. Large values of Δ correspond to easy problems with well-separated Gaussian distributions, whereas difficult problems with overlapping Gaussian distributions are obtained for small values of Δ. The problem parameters are $\Delta=0.5$, $\Delta=2$ and $\Delta=4$ and the class priors are uniform, i.e. $P(Y=y)=\frac{1}{3}$ for each y. Fig. 8 shows an example for each difficulty of three-class classification problem.

Similar to Section 4, 10^6 pairs of features are generated for each of the three-class classification problems. In each pair, the two features are compared in terms of mutual information and misclassification probability. Mutual information is computed in base 2, whereas all probabilities are given in percents. Since the distribution $P(X)$ is a mixture of Gaussian distributions, it is impossible to obtain exact values for the entropy $H(X)$ and the mutual information. Only the conditional entropy $H(Y|X)$ and the conditional probabilities $P(X|Y)$ can be computed analytically. In order to solve this problem, for each feature, 10^4 samples are drawn from each class. The conditional probabilities $P(X|Y)$ of the samples are computed analytically and used to obtain an estimate of the mutual information and the misclassification probability. Given the large number of samples and the low dimensionality of the data, the estimates are expected to be accurate. However, to deal with approximation errors, failures with a mutual information difference below 0.01 or a misclassification probability loss below 0.1% are ignored. Remaining computations and Figs. 9–11 are obtained similarly to Section 4.

5.2. Results

For $\Delta=0.5$, the three classes are quite difficult to discriminate, as seen in Fig. 9(a) and (b). The percentage of failures is 1.2 and the failure probability decreases for extreme (i.e. small or large) mutual information values and extreme misclassification probabilities in Fig. 9(d) and (e). Fig. 9(f) shows that 95% of the misclassification probability losses remain below 3.2%. In Fig. 9(g) and 9(h), the misclassification probability loss decreases for extreme mutual information values and misclassification probabilities. Eventually, Fig. 9(i) shows that failures mostly occur for pairs of features which are close in terms of both mutual information and misclassification probability.

For $\Delta=2$, classification is of medium difficulty, as seen in Fig. 10(a) and (b). The percentage of failure is 0.9 and the failure probability decreases for large mutual information values and small misclassification probabilities in Fig. 10(d) and (e). Fig. 10(f) shows that 95% of the misclassification probability losses remain below 2.7%. In Fig. 10(g) and (h), the misclassification probability loss decreases for large mutual information values and small misclassification probabilities. Again, failures occur for pairs of features which are close in terms of both mutual information and misclassification probability, as seen in Fig. 10(i).

For $\Delta=4$, the three classes are quite easy to discriminate, as seen in Fig. 11(a) and (b). The percentage of failure is 0.2 and the failure probability decreases for large mutual information values and small misclassification probabilities in Fig. 11(d) and (e). Fig. 11(f) shows that 95% of the misclassification probability losses remain below 1%. Figs. 11(g)–(i) lead to similar conclusions than for $\Delta=2$.

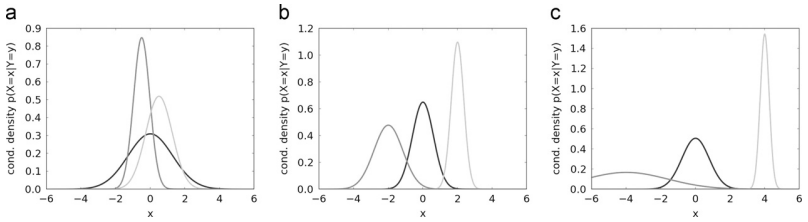


Fig. 8. Examples of three-class balanced classification problems of various difficulties. Standard deviations of the Gaussian distributions are randomly drawn from a gamma distribution with shape $k=2$ and scale $\theta=0.5$, whereas centers are chosen using $\Delta=0.5$, $\Delta=2$ and $\Delta=4$.

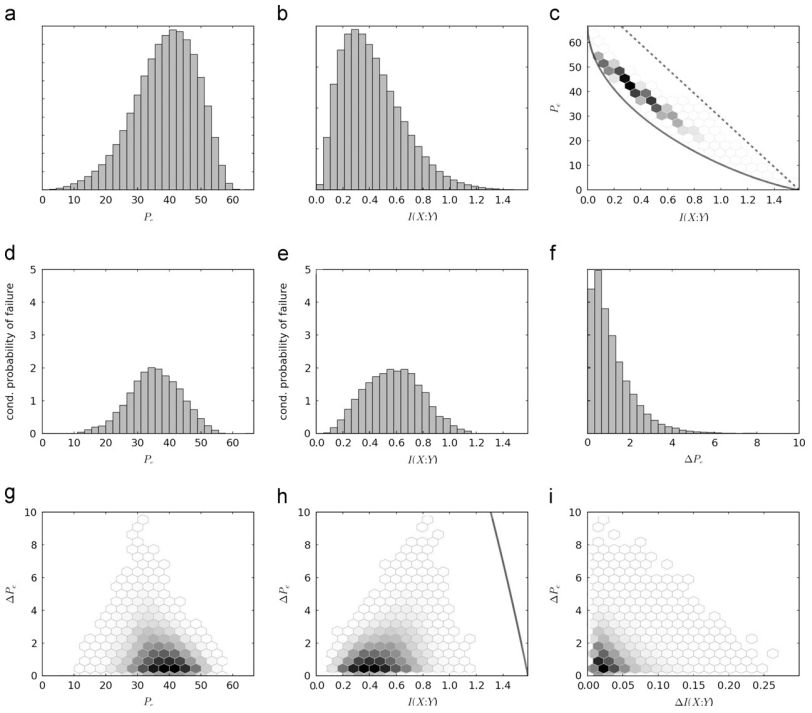


Fig. 9. Results for 10^6 pairs of artificial three-class problems with one continuous feature. Class distributions are Gaussians centered at $x=-0.5$, $x=0$ and $x=0.5$ and whose widths are randomly drawn from a gamma distribution. Mutual information values and misclassification probabilities are estimated using 10^4 samples from each class.

5.3. Discussion

The lessons of the above experiments are similar to those of the experiments in Section 4. Mutual information fails to select the feature with the best misclassification probability in only a few percents of the cases and the misclassification probability loss remains quite small. Again, failures appear to be more probable and to have more important consequences when classes are moderately difficult to discriminate, i.e. for intermediate values of mutual information and misclassification probability. For all problems, failures mostly occur for pairs of features which are close in terms of both mutual information and misclassification probability.

6. Real-world classification problems with continuous features

This section discusses the use of mutual information for feature selection using three real-world classification problems with continuous features. Feature selection is performed using a mutual information-based forward search algorithm [16], with the aim of assessing whether mutual information failures are more likely to occur at certain stages of a multivariate feature selection process.

6.1. Experimental settings

This section presents the results obtained with real-world datasets from the UCI repository [17]. Three balanced datasets with a large number of instances are chosen, in order to obtain reliable estimates of the mutual information and the misclassification probability. Firstly, Digits is a 10-class digit recognition dataset which contains 10,992 instances with 16 continuous features. Secondly, Wallrobot is a two-class robot navigation dataset which contains 4302 instances with 24 continuous features. The original dataset contains four classes, but only the two majority classes are kept in order to obtain a balanced dataset. Eventually, Wave is a three-class waveform dataset which contains 5000 instances with 21 continuous features. All datasets are almost perfectly balanced.

For each dataset, the feature selection process is repeated 5000 times. For each repetition, 10 features are randomly chosen among the set of available features in order to obtain a sub-problem whose characteristics remain similar to the full problem. Then, a forward search is performed to find feature subsets of

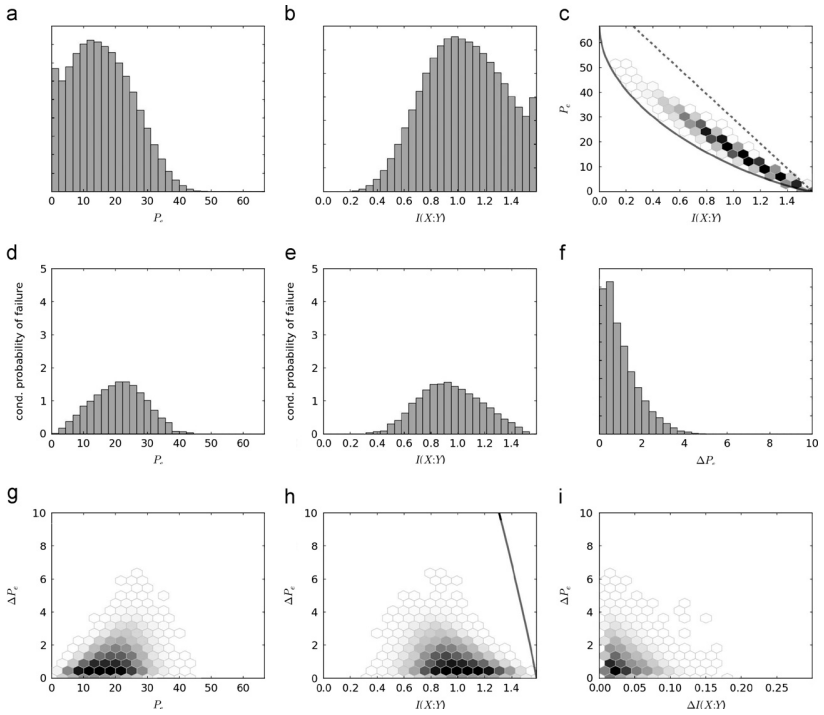


Fig. 10. Results for 10^6 pairs of artificial three-class problems with one continuous feature. Class distributions are Gaussians centered at $x=-2$, $x=0$ and $x=2$ and whose widths are randomly drawn from a gamma distribution. Mutual information values and misclassification probabilities are estimated using 10^4 samples from each class.

increasing sizes. The selection criterion is the mutual information, which is estimated as detailed below. For each forward step in each repetition, the misclassification probabilities are also estimated. A forward step is a failure if the feature subset which is selected in order to maximise the mutual information does not minimise the misclassification probability, i.e. if there exists a feature subset with a lower misclassification probability at this step. The mutual information and the misclassification probabilities are directly estimated using the conditional probabilities $P(Y|X)$ of each sample. These conditional probabilities are obtained from the conditional probabilities $P(X|Y)$, which are estimated using the Kozachenko–Leonenko estimator [18], by using the Bayes rule and marginalisation. Mutual information is computed in base 2, whereas all probabilities are given in percents. Moreover, in case of failure, the mutual information difference and the misclassification probability loss are computed between the feature with the best mutual information and the feature with the best misclassification probability. Fig. 12 shows a two-dimensional histogram of the mutual information and the misclassification probability for each dataset. The Fano and Hellman–Raviv bounds hold, what illustrates the validity of the above procedure.

Figs. 13–15 show several plots for each real-world problem. The first row consists of a histogram of the misclassification probability, a histogram of the mutual information and the misclassification probability for different feature subset sizes. The second row shows an estimate of the conditional probability of failure given the misclassification probability and given the mutual information, and the mutual information for different feature subset sizes. The third row shows two-dimensional histograms of (i) the misclassification probability and the misclassification probability loss, (ii) the mutual information and the misclassification probability loss (with the theoretical bound derived in Section 3.4) and (iii) the mutual information difference and the misclassification probability loss. The fourth row shows the mutual information difference and the misclassification probability loss for different feature subset sizes, and an estimate of the conditional probability of failure given the feature subset size. In the last three rows, which correspond to mutual information failures, the mutual information and the misclassification probability are those of the feature which is selected using mutual information. Indeed, what we are mainly interested in is to know when a feature selected by mutual information is likely to be a bad feature in terms of misclassification probability.

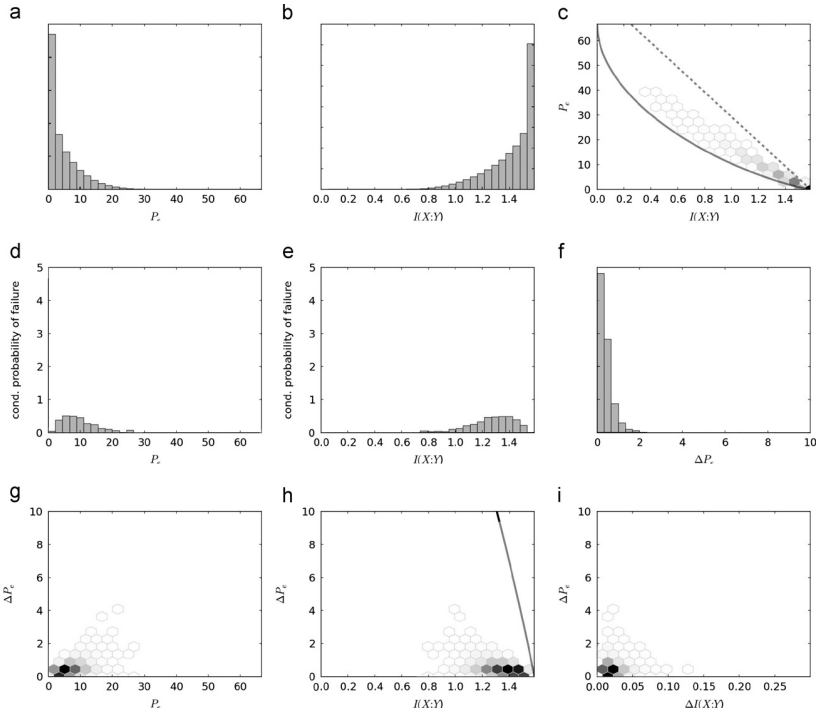


Fig. 11. Results for 10^6 pairs of artificial three-class problems with one continuous feature. Class distributions are Gaussians centered at $x=-4$, $x=0$ and $x=4$ and whose widths are randomly drawn from a gamma distribution. Mutual information values and misclassification probabilities are estimated using 10^4 samples from each class.

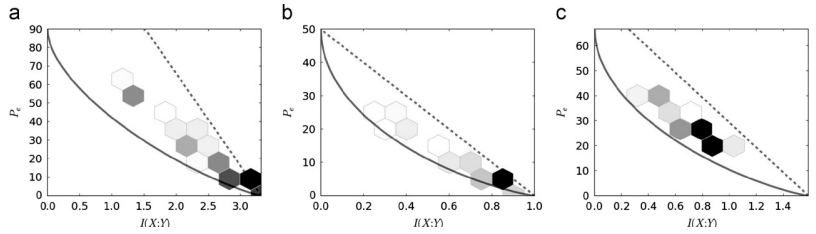


Fig. 12. Mutual information values and misclassification probabilities with random subsets of 10 features for the Digits, Wallrobot and Wave datasets, with Fano and Hellman–Raviv bounds.

6.2. Results

Figs. 13(a)–(c) and (f) show that forward search goes through a wide range of problem difficulties for the Digits dataset. As the feature subset size increases, the misclassification probability decreases slowly. The percentage of failures is 7.8, but Figs. 13(g)–(i) show that 95% of the misclassification probability loss remain below 2%. The dark hexagonal bin in these figures indicates the failures 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. The misclassification probability loss decreases for large mutual information values and small misclassification probabilities. In Fig. 13(d) and (e), the

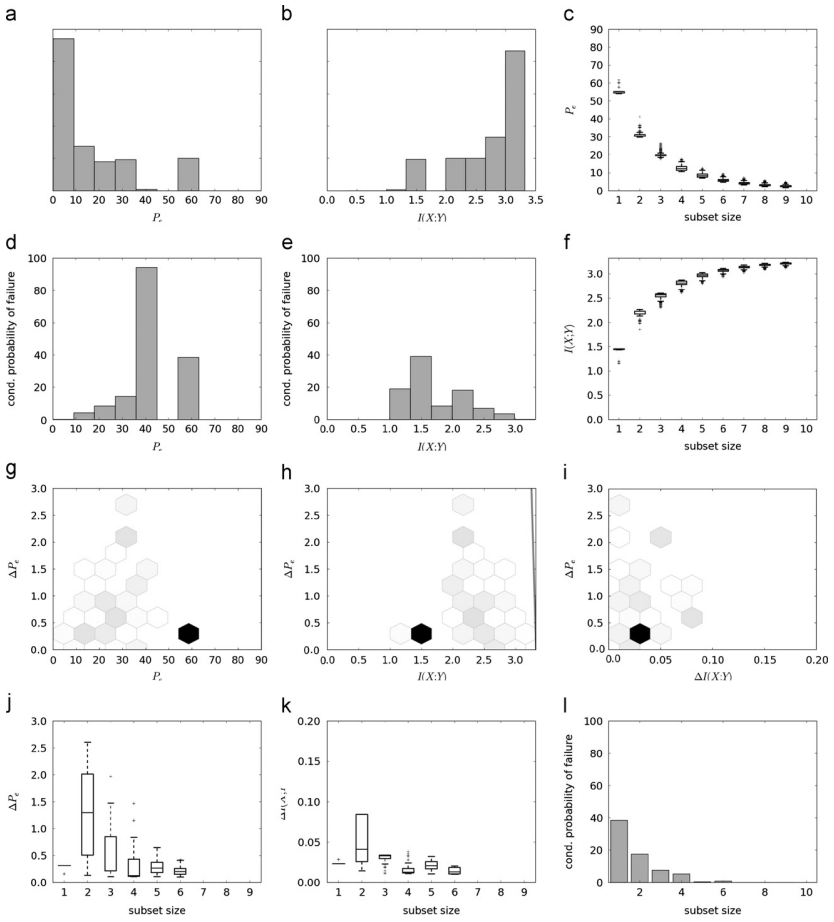


Fig. 13. Results of mutual information-based forward search with random subsets of 10 features for the Digits dataset.

misclassification probability loss decreases for large mutual information values and small misclassification probabilities. Figs. 13(j)–(l) show that the probability of failure is maximum at the beginning of the forward search, where the misclassification probability loss is small, and decreases quickly as the features subset size increases.

The results for the Wallrobot dataset are similar to the result for the Digits dataset, except (i) that classification performances are already optimal with about three features, as seen in Fig. 14(c) and (f), and (ii) that failures are much more likely to occur at the first step of the forward search, as seen in Fig. 14(l), what corresponds to intermediate values of mutual information and the misclassification probability. Consequently, Fig. 14(j) shows that the misclassification probability loss is only significant for subsets with a single feature. It corresponds to the peak of probability of failure in Fig. 14(d) and (e) and to the dark hexagonal bin in Fig. 14(g)–(i). The percentage of failures is 2.4 and 95% of the misclassification probability losses remain below 1%.

Figs. 15(a)–(c) and (f) show that the Wave dataset corresponds to a quite difficult problem. The percentage of failure is 0.2. Contrary to the Digits and Wallrobot datasets, the conditional probability of failure in Fig. 15(l) first increases for small features subset sizes, achieves its maximum for three features and then quickly decreases. The misclassification probability is large for the two first feature subset sizes in Fig. 15(c), what suggests again that failures more likely occur for intermediate mutual information

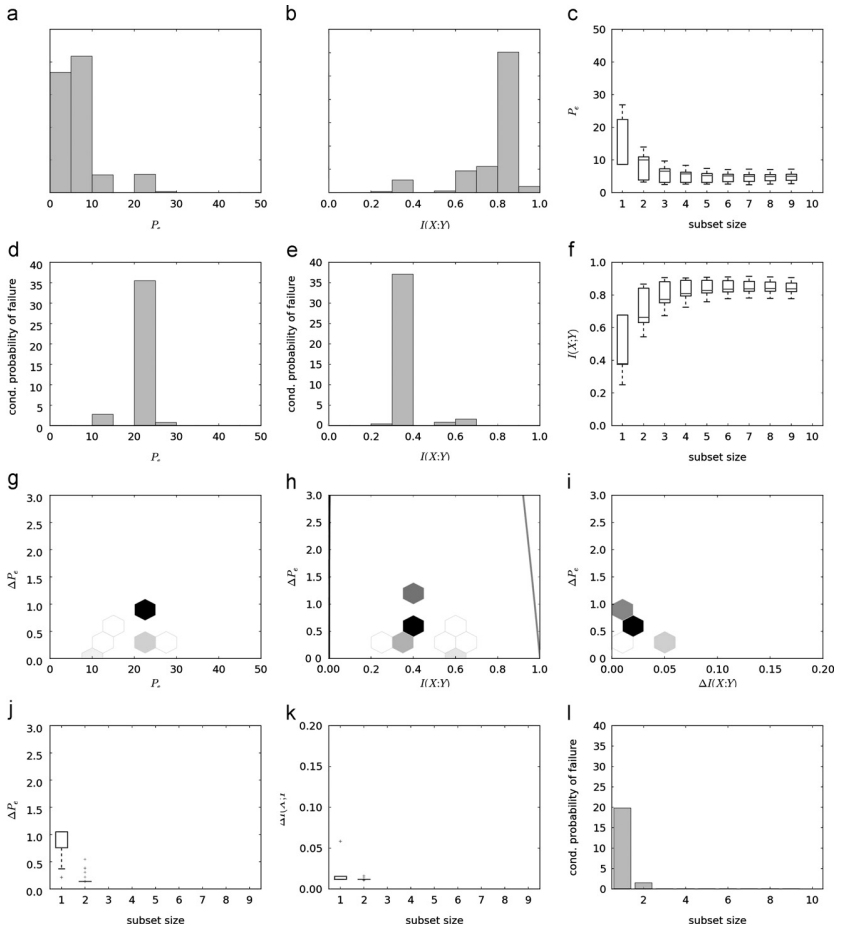


Fig. 14. Results of mutual information-based forward search with random subsets of 10 features for the Wallrobot dataset.

values and misclassification probabilities. Figs. 15(d) and (e) show a peak of probability of failure which corresponds to the dark hexagonal bin in Fig. 15(g)–(i), where 95% of the misclassification probability losses remain below 0.4%.

6.3. Discussion

The results of the above experiments on real-world datasets show that mutual information is more likely to fail in the first stages of the forward search. These situations correspond to intermediate values of mutual information. For the three datasets, misclassification probability losses remain in the order of the percent. This shows that mutual information failures do not have important consequences in practice. In each experiment, failures 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.

7. Meta-analysis of the experimental results

This section reviews and summarises the experimental results and the elements discussed in Sections 4–6, in order to extract several general conclusions. Firstly, the experiments show that

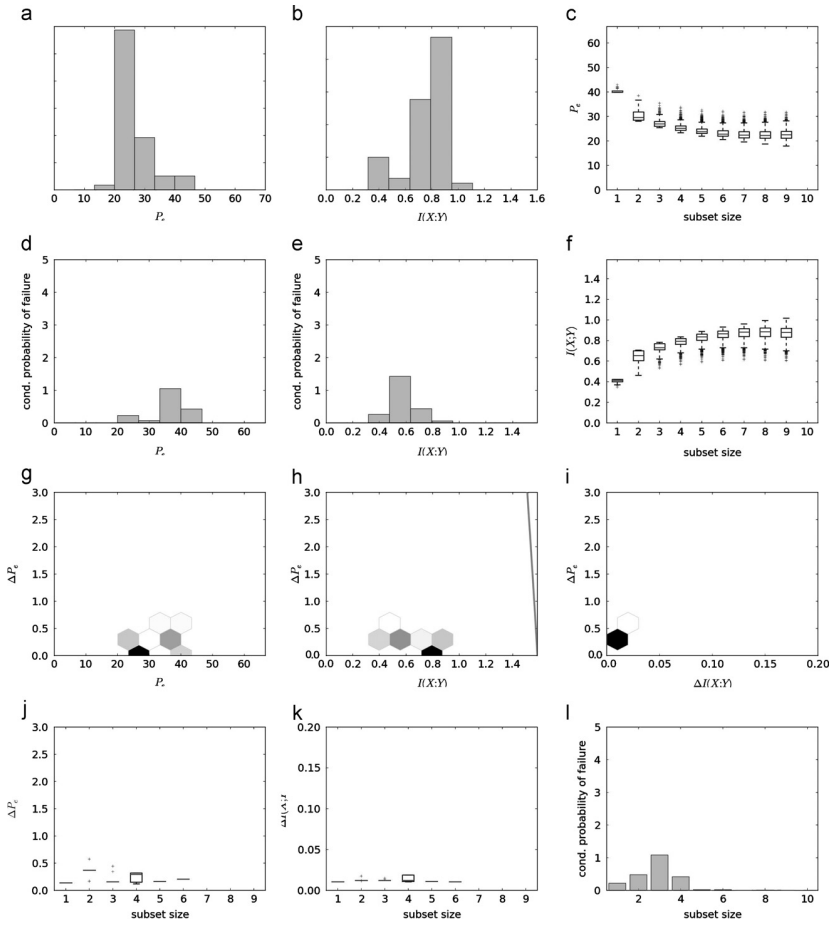


Fig. 15. Results of mutual information-based forward search with random subsets of 10 features for the Wave dataset.

mutual information can fail for feature selection in a wide range of artificial and real-world problems. However, the average percentage of failure is relatively small (often below 5%) and the misclassification probability loss remains in the order of a few percents. In particular, for the three real-world problems, the misclassification probability loss remains below 2% for 95% of the failures. Secondly, mutual information failures are more probable for intermediate values of mutual information. In forward selection, this case occurs in the first steps, when the feature subset size is still small. Hence, on a practical point of view, it could be a good idea to perform several backward steps after the first steps of the forward search, when the algorithm has reached a region where mutual information is more likely to be a reliable criterion for feature selection. Another effective option is to start a forward search with all combinations of 2 or 3 features (when computationally affordable) rather than with a single feature. Moreover, experiments suggest that the backward search algorithm could obtain more reliable feature subsets, since it directly starts in the region where mutual information is reliable and only reaches the dangerous zone after having found satisfying feature subsets. Thirdly, failures occur when comparing features which are close in terms of both mutual information and misclassification probability. Fourthly, in all experiments, the misclassification probability loss remains below the theoretical bound given in Section 3 and the Fano and Hellman–Raviv bounds are satisfied, what supports the validity of the experimental results.

8. Conclusion

This paper shows that in a classification context, mutual information is not always an optimal criterion to achieve feature selection, if the actual goal is eventually to minimize the probability of misclassification. Indeed, as it is first illustrated through a simple example, the Fano and Hellman–Raviv bounds do not guarantee such an optimality, contrary to what can be read in the literature. Extensive experiments on both continuous and discrete datasets confirm this fact and allow detecting the situations for which the mutual information criterion is the more likely to fail. It results that, taking some precautions and possibly adapting the search algorithm, mutual information remains a very interesting heuristic for feature selection.

Acknowledgments

The authors would like to thank the ESANN'12 reviewers and attendees for their fruitful discussions and comments on the subject of this paper, in particular Pierre Dupont, Amaury Lendasse, Fabrice Rossi and Jochen J. Steil.

References

[1] R.E. Bellman, Adaptive Control Processes—A Guided Tour, Princeton University Press, Princeton, New Jersey, USA, 1961.
[2] M. Verleysen, Learning high-dimensional data, Limitations Future Trends Neural Comput. 186 (2003) 141–162.
[3] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[4] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Networks 5 (1994) 537–550.
[5] T.M. Cover, J.A. Thomas, Elements of Information Theory, 99th edition, Wiley-Interscience, 1991.
[6] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1226–1238.
[7] F. Fleuret, Fast binary feature selection with conditional mutual information, J. Mach. Learn. Res. 5 (2004) 1531–1555.
[8] F. Rossi, A. Lendasse, D. Francoi, V. Wertz, M. Verleysen, Mutual information for the selection of relevant variables in spectrometric nonlinear modelling, Chemom. Intell. Lab. Syst. 80 (2006) 215–226.
[9] B. Frénay, G. Doquire, M. Verleysen, On the potential inadequacy of mutual information for feature selection, in: Proceedings of ESANN 2012, 2012.
[10] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423 623–656.
[11] R. Fano, Transmission of Information: A Statistical Theory of Communications, The MIT Press, Cambridge, MA, 1961.
[12] J.W. Fisher, M. Siracusa, T. Kihn, Estimation of signal information content for classification, in: Proceedings of DSP/SPE 2009, 2009.
[13] M.E. Hellman, J. Raviv, Probability of error, equivocation and the Chernoff bound, IEEE Trans. Inf. Theory 16 (1970) 368–372.
[14] G. Brown, An information theoretic perspective on multiple classifier systems, in: Proceedings of MCS 2009, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 344–353.
[15] U. Ozertem, D. Erdogmus, R. Jenssen, Spectral feature projections that maximize Shannon mutual information with class labels, Pattern Recognition 39 (2006) 1241–1252.
[16] D. Francois, F. Rossi, V. Wertz, M. Verleysen, Resampling methods for parameter-free and robust feature selection with mutual information, Neurocomputing 70 (7–9) (2007) 1276–1288.
[17] D.N.A. Asuncion, UCI machine learning repository, 2007 URL 〈http://www.ics.uci.edu/~mlearn/MLRepository.html〉.
[18] L.F. Kozachenko, N. Leonenko, Sample estimate of the entropy of a random vector, Probl. Inf. Transm. 23 (1987) 95–101.

Benoît Frénay received the Engineer's degree from the Université catholique de Louvain (UCL), Belgium, in 2007. He is now a Ph.D. student at the UCL Machine Learning Group. His main research interests in machine learning include support vector machines, extreme learning, graphical models, classification, data clustering, probability density estimation and label noise.

Gauthier Doquire was born in 1987 in Belgium. He received the M.S. in Applied Mathematics from the Université catholique de Louvain (Belgium) in 2009. He is currently a Ph.D. student at the Machine Learning Group of the same university. His research interests include machine learning, feature selection and mutual information estimation.

Michel Verleysen received the M.S. and Ph.D. degrees in electrical engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He was an invited professor at the Swiss E.P.F.L. (Ecole Polytechnique Fédérale de Lausanne, Switzerland) in 1992, at the Université d'Evry Val d'Essonne (France) in 2001, and at the Université Paris1-Panthéon-Sorbonne from 2002 to 2011, respectively. He is now a Full Professor at the Université catholique de Louvain, and Honorary Research Director of the Belgian F.N.R.S. (National Fund for Scientific Research). He is editor-in-chief of the *Neural Processing Letters* journal (published by Springer), chairman of the annual ESANN conference (European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning), past associate editor of the *IEEE Transactions on Neural Networks* journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He was the chairman of the IEEE Computational Intelligence Society Benelux chapter (2008–2010), and member of the executive board of the European Neural Networks Society (2005–2010). He is author or co-author of more than 250 scientific papers in international journals and books or communications to conferences with reviewing committee. He is the co-author of the scientific popularization book on artificial neural networks in the series "Que Sais-Je?", in French, and of the "Nonlinear Dimensionality Reduction" book published by Springer in 2007. His research interests include machine learning, artificial neural networks, self-organization, time-series forecasting, nonlinear statistics, adaptive signal processing, and high-dimensional data analysis.

Chapter 8

Risk Estimation and Feature Selection

The following article has been presented at the 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, 24-26 April 2013. Two ways of estimating the risk are proposed using the Kozachenko-Leonenko probability density estimator, which is usually used to estimate mutual information. Experiments show that using an estimator of either the risk or the mutual information give similar results. Related papers about the adequacy of mutual information for feature selection include [2, 11, 12]. Reprinted with permission from [11].

Risk Estimation and Feature Selection

Gauthier Doquire, Benoît Frénay and Michel Verleysen *

Université catholique de Louvain - ICTEAM/ELEN - Machine Learning Group
Place du Levant 3, 1348 Louvain-la-Neuve - Belgium

Abstract.
For classification problems, the risk is often the criterion to be eventually minimised. It can thus naturally be used to assess the quality of feature subsets in feature selection. However, in practice, the probability of error is often unknown and must be estimated. Also, mutual information is often used as a criterion to assess the quality of feature subsets, since it can be seen as an imperfect proxy for the risk and can be reliably estimated. In this paper, two different ways to estimate the risk using the Kozachenko-Leonenko probability density estimator are proposed. The resulting estimators are compared on feature selection problems with a mutual information estimator based on the same density estimator. Along the line of our previous works, experiments show that using an estimator of either the risk or the mutual information give similar results.

1 Introduction

In classification, model performances are usually assessed by the risk or, equivalently, the probability of error. In the context of feature selection, this criterion can e.g. be used to select the best subset of features, for a given number of features. However, the risk is usually not available and has to be estimated from training data. Risk estimation has been tackled in different works [1, 2, 3], which mostly rely on discretising features [4] or on counting errors made by a k nearest-neighbours classifier [5]. Alternatively, mutual information can also be used instead, since it is strongly related to the risk [6, 7]. Based upon the Kozachenko-Leonenko density estimator [8], the variant of the Kraskov estimator [9] proposed by Gomez et al. [10] can be used in classification. Using mutual information gives good results in feature selection [11], even if maximising the mutual information is not always equivalent to minimising the risk [12, 13]. In line with [12, 13], this paper tackles direct risk estimation in a way which allows a fair comparison between risk and mutual information in feature selection.

In this paper, it is proposed to use the Kozachenko-Leonenko estimator [8] to estimate the risk in two different ways. These two estimators and the mutual information estimator of Gomez et al. [10] are compared on feature selection problems. The goal of this paper is to assess whether it is interesting to directly estimate the risk instead of using mutual information and how the risk should be estimated. Since the Kozachenko-Leonenko probability density estimator is at the heart of these three estimators, it allows a fair comparison.

This paper is organised as follows. Section 2 reviews the literature on risk estimation and discusses the use of mutual information as a proxy to the risk.

*Gauthier Doquire is funded by a Belgian F.R.I.A grant.

Section 3 proposes two new estimators based on the Kozachenko-Leonenko estimator and discusses a mutual information estimator for classification. These three estimators are compared in Section 4 and Section 5 concludes the paper.

2 Risk and Mutual Information in Feature Selection

Given the random variables $X \in \Re^d$ and $Y \in \mathcal{Y}$ corresponding to the associated class, the classification risk [14] for a given classifier $f : \Re^d \to \mathcal{Y}$ is defined as

$$R(f) = \mathop{\mathbb{E}}_{X,Y} \left[\mathbb{I} \left[y \neq f(x) \right] \right]. \tag{1}$$

where x and y are the values taken by X and Y and $\mathbb{I}[.]$ is the indicator function. The Bayes risk is the optimal risk which can be achieved, i.e.

$$R^* = \min_f R(f) = \mathop{\mathbb{E}}_{X} \left[1 - \max_{y \in \mathcal{Y}} p_{Y|X}(y|x) \right] \tag{2}$$

where $P_{Y|X}$ is the conditional distribution of Y given X. In the above equation, the label $y_{\max}$ which maximises $p_{Y|X}(y|x)$ for a given x is called the Bayes decision. In the rest of this paper, the risk always refers to the Bayes risk, since we are interested in selecting features which lead to the best possible classification performances, i.e. when they are used by an optimal classifier.

The idea of feature selection through risk estimation is not new and dates back to [1, 2]. These papers are based on rectangular Parzen density estimation and require the features to first be discretised, which leads to a loss of information. Moreover, each possible combination of discretised feature values has to be considered, which is not tractable for high-dimensional datasets. Feature discretisation is also needed in [4] which focuses on cancer classification problems. In [3], binary classification problems are tackled through Parzen or k-NN density estimation procedures. Related works also include [5] which counts the number of mistakes made by a weighted 1-NN classifier and [15] which establishes relationships between risk minimisation and the well-known Relief algorithm. Contrarily to the risk estimators reviewed above, those proposed in this paper are able to deal with continuous features and multi-label classification problems.

Instead of the risk, mutual information (MI) has often been used as a feature selection criterion. MI is a symetrical quantity measuring the amount of information that two variables carry about each other. It is formally defined as

$$I(X;Y) = H(X) - H(X|Y), \tag{3}$$

where

$$H(X) = -\int_X p_X(x) \log p_X(x) dx \tag{4}$$

is the entropy of the continuous random variable X and

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p_Y(y) H(X|Y = y) \tag{5}$$

is the conditional entropy of X given Y is known (for Y assumed to be discrete) [16]. One of the main reasons for the use of MI in feature selection is the existence of an upper and a lower bound on the Bayes risk R^* as a function of the conditional entropy (and thus equivalently of the MI) [6, 7]. However, as demonstrated in [12, 13], MI is not an ideal proxy for mutual information in feature selection. Indeed, in some specific situations, a feature subset having a higher MI with the class labels than another one could actually lead to a higher risk. MI is thus not always optimal from the risk point of view.

3 Using the Kozachenko-Leonenko Estimator

The Kozachenko-Leonenko estimator [8] is a nearest neighbours density estimator which can e.g. be used to estimate mutual information [9, 10]; it assumes that p_X remains constant in a small hypersphere with diameter $\epsilon_k(i)$ containing exactly the k nearest neighbours of the ith sample. Using this hypothesis, Kozachenko and Leonenko obtain the following estimate

$$\log \hat{p}_X(x_i) = \psi(k) - \psi(n) - \log c_d - d \log \epsilon_k(i) \tag{6}$$

where ψ is the digamma function and c_d is the volume of the d-dimensional unit hypersphere. The Kozachenko-Leonenko estimator can been used to estimate mutual information [9, 10], since one can write

$$\hat{I}(X;Y) = \hat{H}(X) - \sum_{y \in \mathcal{Y}} \hat{p}_Y(y) \hat{H}(X|Y=y); \tag{7}$$

using the density estimator defined in Equation (6), one eventually obtains

$$\hat{I}(X;Y) = \psi(n) - \frac{1}{n} \sum_{y \in \mathcal{Y}} n_y \psi(n_y) + \frac{d}{n} \left[\sum_{i=1}^{n} \log \epsilon_k(i) - \sum_{y \in \mathcal{Y}} \sum_{i|y_i=y} \log \epsilon_k(i|y) \right] \tag{8}$$

where n_y is the number of samples which belong to class y and $\epsilon_k(i|y)$ is the diameter of the hypersphere containing the k nearest neighbours in that class.

This paper proposes to estimate the risk using the Kozachenko-Leonenko estimator. Indeed, Bayes' rule allows one to obtain the estimate

$$\hat{p}_{Y|X}(y|x) = \frac{\hat{p}_{X|Y}(x|y)\hat{p}_Y(y)}{\sum_{y \in \mathcal{Y}} \hat{p}_{X|Y}(x|y)\hat{p}_Y(y)}, \tag{9}$$

which can in turn be used to estimate the risk in two possible ways.

Firstly, one can simply count misclassifications using Equation (9), i.e. use

$$\widehat{R^*} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left[y_i \neq \arg\max_{y \in \mathcal{Y}} \hat{p}_{Y|X}(y|x) \right] \tag{10}$$

which is an empirical estimator [17] of the true risk and is very similar to what is commonly used to estimate the risk of classifiers on test instances.

Secondly, one can also rely on the alternative empirical estimator of the risk

$$\widehat{R^*} = \frac{1}{n} \sum_{i=1}^{n} \left[1 - \max_{y \in \mathcal{Y}} \hat{p}_{Y|X}(y|x_i) \right]. \tag{11}$$

The main difference between the estimators (10) and (11) is that the former uses the training labels, whereas the latter uses the estimated class memberships.

The two risk estimators discussed in this section are compared for feature selection in the rest of this paper. The estimators (10) and (11) are similar to the approaches used e.g. in [1, 2, 3, 4, 5], but they rely on the Kozachenko-Leonenko estimator which (i) is an actual density estimator contrarily to some k-neighbours estimators, (ii) gives good results in feature selection [11] and (iii) can deal with high-dimensional data. Moreover, using the Kozachenko-Leonenko estimator for the estimators (8), (10) and (11) allows a fair comparison in Section 4. Notice that using Equation (11) is more costly than Equations (8) and (10), since $n|\mathcal{Y}|$ conditional probabilities have to be estimated in the former case, whereas only n conditional probabilities are needed in the latter case.

4 Experiments

This section compares the three quantities[1] introduced in Section 3, i.e. the mutual information (8) and the two risk estimators (10) and (11), as feature selection criteria. Feature selection has consequently been carried out using these three criteria with a greedy backward search procedure. Backward search starts with all features and recursively eliminates the one whose removal leads to the highest value of mutual information or to the lowest value of risk, according to the considered criterion. While other procedures such as the forward search could be used instead, it has been suggested in [12, 13] that with the mutual information, backward procedures are expected to produce better results.

The criterion of comparison is the balanced classification rate (the class-mean of the percentage of the samples of a particular class correctly classified) of a 1-nearest neighbor classifier, as a function of the number of selected features, obtained on a test set independent of the training set. The 1-NN classifier has been chosen for both its simplicity and its sensitivity to irrelevant features. Indeed, it gives the same weight to each feature and is not able to perform any kind of embedded feature selection. The results have been obtained through a 10-fold cross-test procedure. To avoid any problem in the determination of the nearest neighbours in the MI or risk estimators, a small random zero-mean Gaussian noise with variance 10^{-3} has been added to the features of each training set before the feature selection process. The noisy datasets are only used for feature selection, while the noise-free datasets are considered for classification.

Figure 1 shows the performances of the three approaches on 6 datasets from the UCI repository [18]. As it can be seen, the results obtained with the three methods are quite similar. Mutual information (8) and error counting (10)

[1] MATLAB and Python implementations are available at `http://www.ucl.ac.be/mlg`.

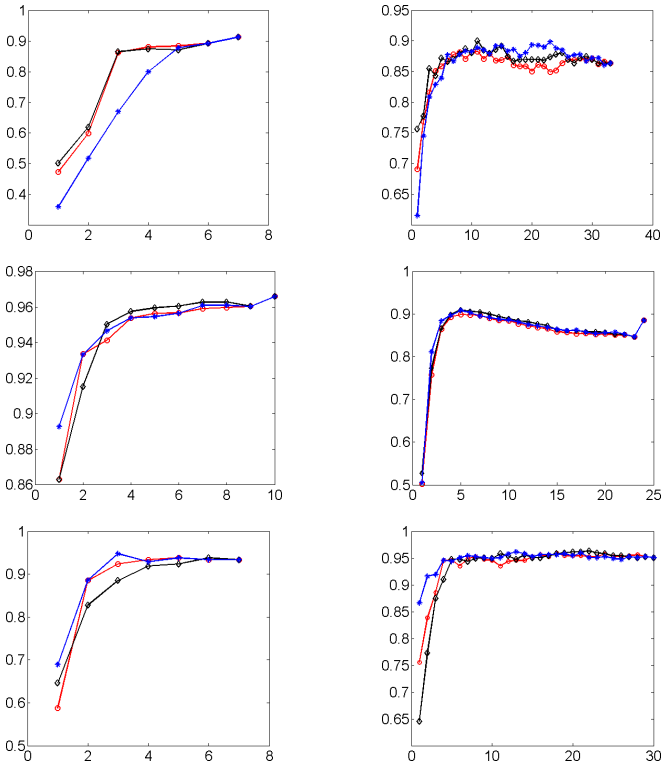


Figure 1: Balanced classification rate of a 1-nearest neighbour classifier as function of the number of selected features obtained with the mutual information (8) (o), the misclassification count (10) ($\diamond$) and the direct risk estimation (11) (*).

perform better on the Ecoli dataset (Fig. 1(a)), while risk estimation (11) is slightly better on the Ionosphere (Fig. 1(b)) and Seeds (Fig. 1(e)) datasets. Performances are equivalent on the other datasets. To asses the significance of the results, a two sample test of means has been carried out, following prescriptions in [19]. MI performances are significantly better for the first three feature subsets for the Ecoli Dataset. No other significative differences can be observed, except for very small features subsets of one or two features in some datasets. It is worth noting that the error counting (10) seems sufficient to get good results.

5 Conclusion

This paper proposes two estimation procedures for the Bayes classification risk using the Kozachenko-Leonenko density estimator. The risk estimators do not

require any feature discretisation and can deal with multi-class problems. The interest of the proposed estimators is illustrated in a feature selection context, where their performances are shown to be comparable to the ones of the mutual information criterion estimated based on the same entropy estimator. This observation is in good agreement with our previous work and shows again the strong relationships existing between these two criteria. Besides feature selection, the proposed risk estimators could as well be used in another area; for instance they could easily be applied to instance selection in active learning.

References

[1] P.J. Min. A non-parametric method for feature selection. In *Adaptive Processes, 1968. Seventh Symposium on*, volume 7, page 34, 1968.

[2] K. S. Fu, P. J. Min, and T. J. Li. Feature selection in pattern recognition. *IEEE T. Syst. Man Cyb.*, 6:33–38, 1970.

[3] Keinosuke Fukunaga and Donald M. Hummels. Bayes error estimation using parzen and k-nn procedures. *IEEE T. Pattern Anal.*, 9(5):634 –643, 1987.

[4] Jian Li, Jin-Mao Wei, Tian Yu, and Hai-Wei Zhang. Feature selection based on bayes minimum error probability. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 706 –710, 2012.

[5] Peng-Fei Zhu, Tian-Hang Meng, Yun-Long Zhao, Rui-Xian Ma, and Qing-Hua Hu. Feature selection via minimizing nearest neighbor classification error. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, pages 506 –511, 2010.

[6] R. Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.

[7] M. E. Hellman and J. Raviv. Probability of error, equivocation and the chernoff bound. *IEEE T. Inform. Theory*, 16:368–372, 1970.

[8] L. F. Kozachenko and N. Leonenko. Sample estimate of the entropy of a random vector. *Problems Inform. Transmission*, 23:95–101, 1987.

[9] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69(6):066138, 2004.

[10] Vanessa Gómez-Verdejo, Michel Verleysen, and Jérôme Fleury. Information-theoretic feature selection for functional data classification. *Neurocomputing*, 72:3580–3589, 2009.

[11] F. Rossi, A. Lendasse, D. Francois, V. Wertz, and M. Verleysen. Mutual Information for the Selection of Relevant Variables in Spectrometric Nonlinear Modelling. *Chemometr. Intell. Lab.*, 80:215–226, 2006.

[12] B. Frénay, G. Doquire, and M. Verleysen. Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification. *Accepted for publication in the Special Issue for ESANN 2012 of Neurocomputing*.

[13] B. Frénay, G. Doquire, and M. Verleysen. On the potential inadequacy of mutual information for feature selection. In *Proceeding of ESANN*, 2012.

[14] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2001.

[15] Shuang Hong Yang and Bao-Gang Hu. Discriminative feature selection by nonparametric bayes error minimization. *IEEE T. Knowl. Data En.*, 24(8):1422 –1434, 2012.

[16] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[17] B. Schölkopf and A.J. Smola. *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Mit Press, 2002.

[18] D.J. Newman A. Asuncion. UCI machine learning repository, 2007.

[19] R.H. Riffenburgh. *Statistics in Medicine*. Academic Press, 2012.

Chapter 9

Is Mutual Information Adequate for Feature Selection in Regression ?

The following article has been published in Volume 48 (2013) of the Neural Networks journal. This paper demonstrates that under some reasonable assumptions, features selected with the mutual information criterion are the ones minimising the mean squared error and the mean absolute error. It is also shown that the mutual information criterion can fail in selecting optimal features in some particular situations. This paper is expected to lead in practice to a critical and efficient use of the mutual information for feature selection. Related papers about the adequacy of mutual information for feature selection include [1, 2, 11]. Reprinted with permission from [12].

Neural Networks 48 (2013) 1–7

Contents lists available at SciVerse ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Neural networks letter

Is mutual information adequate for feature selection in regression?

Benoît Frénay *,[1], Gauthier Doquire [1], Michel Verleysen

Machine Learning Group - ICTEAM, Université catholique de Louvain, Place du Levant 3, 1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:
Received 17 September 2012
Revised and accepted 4 July 2013

Keywords:
Mutual information
Feature selection
Regression
MSE
MAE

ABSTRACT

Feature selection is an important preprocessing step for many high-dimensional regression problems. One of the most common strategies is to select a relevant feature subset based on the mutual information criterion. However, no connection has been established yet between the use of mutual information and a regression error criterion in the machine learning literature. This is obviously an important lack, since minimising such a criterion is eventually the objective one is interested in. This paper demonstrates that under some reasonable assumptions, features selected with the mutual information criterion are the ones minimising the mean squared error and the mean absolute error. On the contrary, it is also shown that the mutual information criterion can fail in selecting optimal features in some situations that we characterise. The theoretical developments presented in this work are expected to lead in practice to a critical and efficient use of the mutual information for feature selection.

1. Introduction

In many regression problems, input data are originally high-dimensional. As an example, in the field of near-infrared spectroscopy analysis, each sample is described by tens or hundreds of features, corresponding to its spectrum components. Much of these features are in practice either redundant or irrelevant to the considered regression problem (Rossi, Lendasse, Francois, Wertz, & Verleysen, 2006).

It is well known that learning with a huge number of features and limited sample size is a hard task because of the so-called *curse of dimensionality* (Bellman, 1961) and its consequences (Verleysen, 2003). In such settings, the risk of overfitting is high, especially when complex models (with numerous parameters) have to be inferred from the data.

In order to address the aforementioned issues, one of the most popular methods is to perform feature selection before any further learning step. The idea is to select a small subset of features which are together highly relevant with the output to predict (see Guyon & Elisseeff, 2003 for a nice introduction). Feature selection has the advantage over projection methods (which project the features onto a space of small dimension) that the original features are not transformed, which allows one to subsequently build easy-to-interpret models.

The feature selection problem should be distinguished from the one of sufficient dimension reduction (Globerson & Tishby, 2003), where the objective is to obtain a subspace of minimal dimension containing the whole information about the output. On the contrary, the goal of feature selection is to select only a few of the original features, even if the price to pay is a small loss of information. Feature selection allows one to considerably reduce the dimension of the dataset, which can improve the performance of the prediction model by reducing the effects of the curse of dimensionality. It thus also speeds up the learning process and leads to a better understanding of the considered problem.

An intuitive and appealing idea is to select the features based on the performance of an inference model. This approach, called *wrapper* in the literature (Kohavi & John, 1997), often leads to good prediction performances but also suffers from two main drawbacks. First, it can be very computationally demanding since many prediction models with different feature subsets have to be built. Then, the results of the wrapper strategy lack generality as their use is limited to a specific regression model. To circumvent both problems, filter methods are often used in practice. Such methods are based on a relevance criterion measuring the quality of feature subsets; this criterion is independent of any prediction algorithm. Filters are traditionally much faster than wrappers and can be used with any regression algorithm. Among the numerous solutions proposed in the literature, mutual information (Shannon, 1948) is one of the most popular relevance criteria, due to many advantages for the feature selection task which will be detailed in the next section. It has therefore been used in a large number of works (see e.g. Dijck & Hulle, 2006, Fleuret, 2004, Kojadinovic & Wottka, 2000, Rossi, François, Wertz, Meurens, & Verleysen, 2007) since the seminal paper Battiti (1994).

* Corresponding author. Tel.: +32 10 47 81 33; fax: +32 10 47 25 98.
E-mail addresses: benoit.frenay@uclouvain.be (B. Frénay), gauthier.doquire@uclouvain.be (G. Doquire), michel.verleysen@uclouvain.be (M. Verleysen).

[1] Both authors contributed equally to this work.

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.neunet.2013.07.003

The eventual objective in a regression problem is to reduce as much as possible an error criterion; the most frequently used ones are the mean squared error (MSE) and the mean absolute error (MAE). However, to the best of our knowledge, no explicit connection has been established in the machine learning literature between the use of the mutual information as a feature selection criterion and the MSE or the MAE. In information theory, the MSE has been e.g. related to the derivative of mutual information with respect to the signal to noise ratio (Guo, Shamai, & Verdú, 2005). Moreover, there exists a relationship between MSE and mutual information (see Section 3.2 and e.g. Chen, Hu, Li, & Sun, 2008, Ihara, 1993), but this relation is only valid in the case of Gaussian estimation errors. This paper addresses the previously discussed lack of connection in machine learning by showing that, assuming some realistic hypotheses on the estimation errors on the target, the mutual information criterion is actually optimal from a MSE or a MAE point of view. This result confirms that the mutual information is a criterion which is worth considering for feature selection. In addition, the paper also illustrates the fact that in some situations, the features selected using the mutual information criterion are not the ones minimising the considered error criterion. In such cases, mutual information should not necessarily be the criterion of choice, and the results of the feature selection procedure should be carefully analysed. This work thus intends to present both theoretical arguments and case studies of the potential interest of mutual information for feature selection in regression problems; the final goal is to better apprehend its behaviour in order to use it in the most efficient and sensitive way. It should be noted that the results in this paper assume the knowledge of the distributions of the random variables, and can thus be seen as infinite-sample arguments. The variance of the mutual information, MSE and MAE estimators are not considered here. A preliminary study on the adequation between mutual information and misclassification probability for classification problems was published in Frénay, Doquire, and Verleysen (2012, 2013). This paper extends the approach and focus on regression tasks.

The remaining of the paper is organised as follows. Section 2 recalls basic notions about mutual information and entropy for feature selection. The well known MSE and MAE error criteria are presented and linked to the estimation error on the target output in the context of feature selection. Section 3 theoretically demonstrates the optimality of the mutual information for three popular models of the estimation error. Section 4, on the contrary, illustrates the potential inadequacy of mutual information and characterises the situations where this criterion is likely to fail. Section 5 briefly discusses the results while Section 6 concludes the work.

2. Theory and notations

This section briefly gives basic definitions about mutual information and entropy. The MSE and the MAE, the two error criteria considered in this work, are then briefly reviewed. Next, Sections 3 and 4 will establish a connection between the use of mutual information for feature selection and the two error criteria.

2.1. Mutual information and entropy

Mutual information (Shannon, 1948) is a quantity measuring the dependency existing between two (groups of) random variables, assumed to be continuous in this work. Let X and Y be random variables whose respective probability density functions are f_X and f_Y and whose domains are $\mathcal{X}$ and $\mathcal{Y}$. Let us also define the joint probability density function $f_{X,Y}$. The mutual information between X and Y is defined as

$$I(X;Y) = -\int_{\mathcal{X}}\int_{\mathcal{Y}} f_{X,Y}(x,y)\ \log\frac{f_{X,Y}(x,y)}{f_X(x)f_Y(y)}\,dxdy. \tag{1}$$

Eq. (1) can actually be rewritten in terms of entropy and conditional entropy, respectively defined as

$$H(X) = -\int_{\mathcal{X}} f_X(x)\ \log f_X(x)\,dx \tag{2}$$

and

$$H(Y|X) = -\int_{\mathcal{X}}\int_{\mathcal{Y}} f_{X,Y}(x,y)\ \log\frac{f_X(x)}{f_{X,Y}(x,y)}\,dxdy. \tag{3}$$

Using Eqs. (1)–(3), it is possible to write

$$I(X;Y) = H(Y) - H(Y|X). \tag{4}$$

The mutual information can thus be understood as the reduction of uncertainty (measured by the entropy) on the values of Y once X is known. If Y denotes the target output to predict and X a subset of features, mutual information has a quite natural interpretation as a feature selection criterion. Indeed, a feature subset having a high mutual information with the target output is likely to reduce the uncertainty on the values taken by the output, what is obviously desirable. Besides an intuitive interpretation, mutual information also has the advantage of detecting non-linear relationships between variables while some other popular criteria (such as the correlation coefficient) are essentially limited to linear dependencies. Eventually, mutual information can naturally be defined for groups of variables, making it possible to evaluate subsets of features; this last property is of crucial importance when jointly redundant or relevant features make univariate criteria useless.

For a given regression problem between inputs X and output Y, $H(Y)$ is fixed and does not depend on the choice of features. Therefore, from a feature selection point of view, Eq. (4) indicates that selecting features X in order to maximise $I(X;Y)$ can be achieved by selecting features which minimise $H(Y|X)$. In the remainder of the paper, the discussion will be about $H(Y|X)$, while the same conclusions can be drawn about $I(X;Y)$.

2.2. Regression error criteria

As mentioned in Section 1, the final objective in a regression problem is to minimise an error criterion, measuring in some way the difference between the predicted value and the actual value of the output. In this section, two popular error criteria are reviewed and linked to the estimation error on the target output.

Let us assume that for a given subset of d features, the output $Y \in \Re$ depends probabilistically on the input $X \in \Re^d$. Moreover, the function f provides an estimate $\hat{Y} = f(X)$ of Y given X. Then, the estimation error is

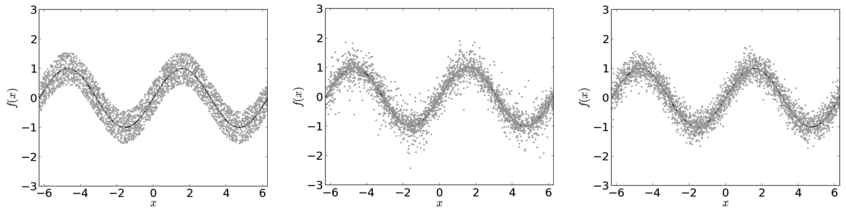
$$\epsilon = f(X) - Y, \tag{5}$$

whose zero-mean distribution depends on the choice of features. Two popular regression error criteria can be rewritten in terms of ϵ, which are discussed below.

The mean square error (MSE) of the estimate f is defined as the variance $\mathrm{E}\left\{(f(X)-Y)^2\right\} = \mathrm{E}\left\{\epsilon^2\right\}$ of ϵ, where $\mathrm{E}\{\cdot\}$ denotes the expected value. Another popular error criterion is the mean absolute error (MAE), defined as the expected absolute value $\mathrm{E}\{|f(X)-Y|\} = \mathrm{E}\{|\epsilon|\}$ of ϵ. In feature selection, one is typically interested in feature subsets which allow one to obtain estimates achieving low MSE or MAE values.

2.3. Error criteria and entropy of estimation error

For a given estimate f, it is well known (see e.g. Ash, 1990, Cover & Thomas, 1991) that the conditional entropy of Y given X can be



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

Fig. 1. Functional $f(x) = \sin(x)$ polluted by uniform, Laplacian or Gaussian target noise with identical conditional target entropy $H(Y|X) = 0.1$.

rewritten in terms of ϵ as

$$H(Y|X) = H(\epsilon|X). \tag{6}$$

To show this relationship, let us rewrite $H(Y|X)$ as

$$\begin{aligned} H(Y|X) &= \int_{\mathcal{X}} f_X(x) H(Y|X = x) dx \\ &= \int_{\mathcal{X}} f_X(x) H(f(X) + \epsilon|X = x) dx. \end{aligned} \tag{7}$$

Since $f(X)$ is fixed when X is known and since the differential entropy is translation invariant ($H(X + k) = H(X)$ for a constant k, see e.g. Emmert-Streib & Dehmer, 2008), Eq. (6) follows directly from Eq. (7).

The rest of this paper considers the relationship between the mutual information and the two error criteria in different settings. Assuming f and the entropy or mutual information estimator can be accurately estimated using the available data (see e.g. Kozachenko & Leonenko, 1987, Kraskov, Stögbauer, & Grassberger, 2004 for mutual information estimation), the results obtained in this paper show the interest of using mutual information as a feature selection criterion for real-world problems.

3. Mutual information adequacies

This section shows how mutual information can be an adequate criterion for feature selection in regression. More specifically, when the conditional distribution of the estimation error is uniform, Laplacian or Gaussian, choosing the feature subset which minimises the conditional target entropy $H(Y|X)$ is equivalent to minimising either the MSE or the MAE criterion. Here, X corresponds to a specific subset of features and $H(Y|X)$ depends on the choice of this subset X. Moreover, it is assumed that when two feature subsets are compared for a given dataset, the distributions of the estimation error belong to the same parametric family (uniform, Laplacian or Gaussian) in both cases. This hypothesis is realistic when feature subsets which are not too different (in terms of informative contain) are compared, like e.g. at a given step of a forward or backward search.

3.1. Specification of regression examples

As explained in Section 2, mutual information is adequate for feature selection in regression with respect to an error criterion if minimising the conditional target entropy $H(Y|X)$ always improves the criterion. The choice of the criterion depends on the application, so it could also be true for mutual information adequacy.

In this section, three realistic estimation error distributions are considered: a uniform, a Laplacian and a Gaussian distribution. The estimation error is assumed to be identically distributed for any $x \in \mathcal{X}$, which means that the conditional entropy $H(Y|X)$ is equal to the specific conditional entropy $H(Y|X = x)$ for any $x \in \mathcal{X}$. Since the means of the estimation error distributions are zero, they only have 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. The value of the estimation error distribution parameter (Δ, λ or σ) depends on the feature subset which corresponds to X. Considering these estimation error distributions, the conditional target entropies $H(Y|X)$ are $\ln \Delta$, $\ln[2e\lambda]$ and $\frac{1}{2}\ln\left[2\pi e\sigma^2\right]$, respectively (Cover & Thomas, 1991).

In order to visualise each type of estimation error, Fig. 1 shows an example of functional $f(x) = \sin(x)$ which is polluted by three types of noise, with identical conditional target entropy $H(Y|X) = 0.1$. Under uniform noise, the function values stay inside a tube around the real function f. Under Laplacian or Gaussian noise, the distribution of function values spreads around the real function f, with more or less thick tails.

3.2. Adequacy assessment for the MSE criterion

As shown in Section 2, the MSE can be interpreted as the expected variance of the estimation error. Since the estimation error is assumed to be identically distributed for any $x \in \mathcal{X}$, its variance is precisely equal to the MSE. For the uniform, Laplacian and Gaussian estimation errors, the variance can be written in terms of their only free parameter as $\frac{\Delta^2}{12}$, $2\lambda^2$ and σ^2, respectively (Cover & Thomas, 1991; Kotz, Kozubowski, & Podgórski, 2001). Using the expressions for the conditional target entropies and these relationships, it is possible to express the MSE in sole terms of $H(Y|X)$. For the uniform, Laplacian and Gaussian estimation error, the MSE becomes $\frac{1}{12}\exp[2H(Y|X)]$, $\frac{1}{2e^2}\exp[2H(Y|X)]$ and $\frac{1}{2\pi e}\exp[2H(Y|X)]$, respectively. Notice that the MSE no longer depends explicitly on the parameter of the estimation error distribution. In the Gaussian case, similar relationships between MSE and conditional entropy have been reported e.g. in Chen et al. (2008), Guo et al. (2005), Ihara (1993).

In the above relationships, the MSE is a monotonically increasing function of the conditional target entropy, which depends on the selected feature subset. It means that if different feature subsets are compared and if the distribution of the estimation errors belongs to the parametric family (uniform, Laplacian or Gaussian) in each case, choosing the feature subset which minimises the conditional target entropy $H(Y|X)$ necessarily corresponds to minimising the MSE. This is illustrated in Fig. 2 which shows the MSE in terms of the conditional target entropy $H(Y|X)$ for the uniform, Laplacian and Gaussian estimation errors. Since the Gaussian distribution is the maximum entropy distribution for a given estimation error variance σ^2 (Cover & Thomas, 1991), the curve corresponding to the Gaussian error gives a lower bound for the MSE and defines an admissible region for the MSE as shown in Fig. 2.

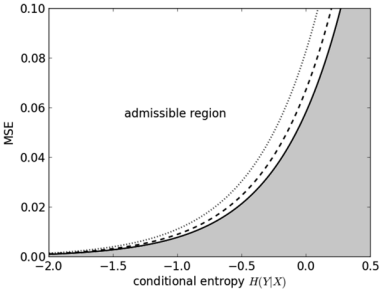


Fig. 2. MSE in terms of the conditional target entropy $H(Y|X)$ for identically distributed uniform (dotted line), Laplacian (dashed line) or Gaussian (plain line) estimation error. The Gaussian curve gives a lower bound and defines an admissible region (in white).

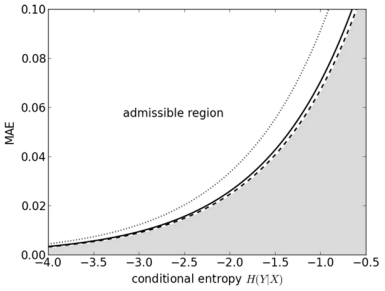


Fig. 3. MAE in terms of the conditional target entropy $H(Y|X)$ for identically distributed uniform (dotted line), Laplacian (dashed line) or Gaussian (plain line) estimation error. The Laplacian curve gives a lower bound and defines an admissible region (in white).

3.3. Adequacy assessment for the MAE criterion

Since the estimation error is assumed to be identically distributed, the MAE is by definition equal to the expected absolute value of the estimation error for any $x \in \mathcal{X}$. For the uniform, Laplacian and Gaussian estimation errors, the expected absolute value can be written in terms of their only free parameter as $\frac{\Delta}{4}$, λ and $\sqrt{\frac{2}{\pi}}\sigma$, respectively (Kotz et al., 2001). Using the expressions for the conditional target entropies and these relationships, it is possible to express the MAE in sole terms of $H(Y|X)$. For the uniform, Laplacian and Gaussian estimation error, the MAE becomes $\frac{1}{4}\exp\left[H(Y|X)\right]$, $\frac{1}{2e}\exp\left[H(Y|X)\right]$ and $\frac{1}{\pi\sqrt{e}}\exp\left[H(Y|X)\right]$, respectively. Notice that the MAE no longer depends explicitly on the estimation error distribution parameter.

In the above relationships, the MAE is a monotonically increasing function of the sole conditional target entropy, which depends on the selected feature subset. It means that if different feature subsets are compared and if the distribution of the estimation errors belongs to the parametric family (uniform, Laplacian or Gaussian) in each case, choosing the feature subset which minimises the conditional target entropy $H(Y|X)$ necessarily corresponds to minimising the MAE. This is illustrated in Fig. 3 which shows the MAE in terms of the conditional target entropy $H(Y|X)$ for the uniform, Laplacian and Gaussian estimation errors. Since the Laplacian distribution is the maximum entropy distribution for a given expected estimation error absolute value λ (Kotz et al., 2001), the corresponding curve gives a lower bound for the MAE and defines an admissible region.

3.4. Short discussion

This section shows that mutual information is an adequate criterion for feature selection with respect to the MSE and the MAE, when the estimation error is identically distributed for any $x \in \mathcal{X}$ with a uniform, Laplacian or Gaussian distribution. Indeed, maximising mutual information is equivalent to minimising the conditional target entropy $H(Y|X)$, which in the above settings also corresponds to minimising either the MSE or the MAE.

4. Mutual information inadequacies

This section shows that mutual information is not always adequate for feature selection in regression. In the proposed example, the conditional distribution of the estimation error is assumed to be a Student distribution. For this particular setting, it is shown that 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.

4.1. Specification of regression examples

In this section, it is assumed that the estimation error follows an identical Student distribution for any $x \in \mathcal{X}$. This distribution is often used for the robust modelling of random variables which look Gaussian, but whose distribution has ticker tails (Archambeau, Delannay, & Verleysen, 2008; Peel & McLachlan, 2000). The density of the non-standardised Student distribution is

$$\mathcal{S}\left(\epsilon = e|\mu, \nu, \sigma\right) = \frac{1}{B\left(\frac{1}{2}, \frac{\nu}{2}\right)\sqrt{\nu\sigma^2}}\left[1 + \frac{(e-\mu)^2}{\nu\sigma^2}\right]^{-\frac{\nu+1}{2}} \tag{8}$$

where B is the beta function. Since the mean of the estimation error is zero, so is the parameter μ and the Student distribution only has two effective parameters: the number of degrees of freedom ν and the scale σ. The value of the distribution parameters ν and σ depend on the selected feature subset X. The conditional target entropy $H(Y|X)$ for a Student estimation error is

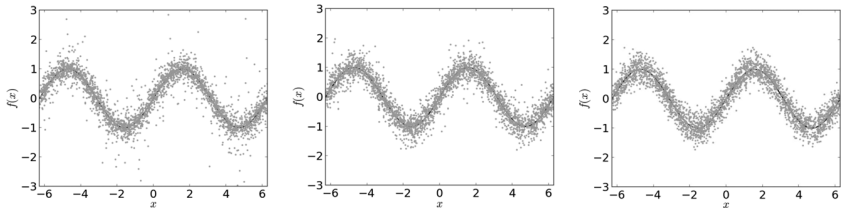
$$\left(\frac{\nu+1}{2}\right)\left[\Psi\left(\frac{\nu+1}{2}\right) - \Psi\left(\frac{\nu}{2}\right)\right] + \ln\left[\sqrt{\nu}B\left(\frac{1}{2}, \frac{\nu}{2}\right)\right] + \ln\sigma \tag{9}$$

where Ψ is the digamma function (Cover & Thomas, 1991). Fig. 4 shows an example of functional $f(x) = \sin(x)$ which is polluted by Student noises with different numbers of degrees of freedom but identical conditional target entropy $H(Y|X) = 0.1$. The spread of the distribution and the thickness of its tail depend on ν.

4.2. Inadequacy assessment for the MSE criterion

For a Student estimation error which is identical for any $x \in \mathcal{X}$, the variance can be expressed in terms of its two free parameters as $\frac{\nu}{\nu-2}\sigma^2$. Hence, using this relationship and the expression of the conditional target entropy, the MSE can be rewritten in terms of the number of degrees of freedom ν and $H(Y|X)$ as

$$\mathrm{E}\left\{(Y - f(X))^2\right\} = \frac{1}{(\nu-2)\,B\left(\frac{1}{2}, \frac{\nu}{2}\right)^2}\exp\left\{2H(Y|X) - (\nu+1)\left[\Psi\left(\frac{\nu+1}{2}\right) - \Psi\left(\frac{\nu}{2}\right)\right]\right\}. \tag{10}$$



(a) Student noise $\mathcal{S}(\mu = 0, \nu = 2.3, \sigma = 0.168)$. (b) Student noise $\mathcal{S}(\mu = 0, \nu = 5, \sigma = 0.217)$. (c) Student noise $\mathcal{S}(\mu = 0, \nu = 30, \sigma = 0.259)$.

Fig. 4. Functional $f(x) = \sin(x)$ polluted by Student noises with different parameters but identical conditional target entropy $H(Y|X) = 0.1$.

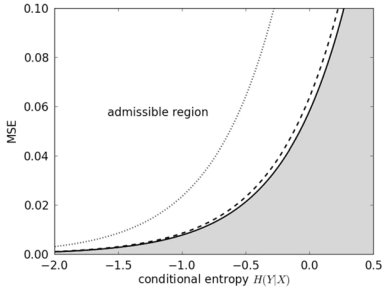


Fig. 5. MSE in terms of the conditional target entropy $H(Y|X)$ for Student estimation error with different numbers of degrees of freedom: $\nu = 2.3$ (dotted line), $\nu = 5$ (dashed line) or $\nu = 30$ (plain line). The admissible region defined by the Gaussian curve appears in white.

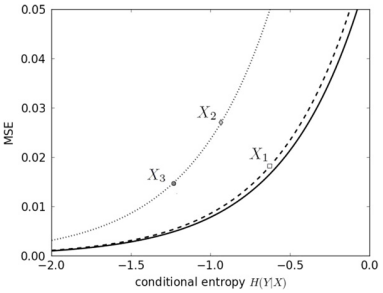


Fig. 6. Example of mutual information failure for Student estimation error with respect to the MSE. The candidate feature subsets correspond to different numbers of degrees of freedom: $\nu = 2.3$ (dotted line) and $\nu = 5$ (dashed line). The curve for $\nu = 30$ (plain line) is also shown for discussion (see text). The symbols X_i are feature subsets.

It shows that the MSE cannot be written in sole terms of the conditional target entropy. Indeed, the MSE still depends on the number of degrees of freedom ν (one could alternatively use σ, but ν is easier to understand intuitively). This is illustrated in Fig. 5 which shows the MSE in terms of the conditional target entropy $H(Y|X)$ for different numbers of degrees of freedom. Since the numbers of degrees of freedom of the Student estimation error distribution for different feature subsets are not necessarily identical (for example, data which are outliers with respect to some features can look normal with respect to other features), it is possible to decrease the conditional target entropy while increasing the MSE. This means that choosing the feature subset which minimises the conditional target entropy $H(Y|X)$ does not necessarily correspond to minimising the MSE.

Fig. 6 shows an example of mutual information failure with respect to the MSE, where two candidate features subsets X_1 and X_2 are characterised by a Student estimation error with parameters $\nu = 2.3$ and $\nu = 5$, respectively. Using mutual information, X_2 will be chosen rather than X_1, since $H(Y|X_2)$ is smaller than $H(Y|X_1)$. However, because of the different degrees of freedom of the estimation error affecting both feature subsets, the MSE is larger for X_2 than for X_1. Hence, selecting X_2 based on mutual information leads here to an increase in the criterion which should be minimised; mutual information fails as feature selection criterion. Notice that, in Fig. 6, X_3 is characterised by the same degree of freedom than X_2, but mutual information does not fail when choosing between X_1 and X_3. Indeed, selecting the feature subset X_3 because $H(Y|X_3)$ is smaller than $H(Y|X_1)$ effectively minimises the MSE.

In practice, the case of mutual information failures discussed in this section may be of little impact. Indeed, Fig. 6 also shows that the curve for $\nu = 30$ is quite close to the curve for $\nu = 5$. Hence, it is less dangerous to compare feature subsets with such estimation error distributions, which are common in practice. Since $\nu = 2.3$ is quite extreme as can be seen in Fig. 4, one can use mutual information in most practical cases with a quite reasonable confidence.

4.3. Inadequacy assessment for the MAE criterion

For the Student estimation error, the expected absolute value can be expressed in terms of its two free parameters (Psarakis & Panaretos, 1990) as

$$\mathrm{E}\left\{|Y - f(X)|\right\} = \frac{2\sqrt{\nu\sigma^2}}{B\left(\frac{1}{2}, \frac{\nu}{2}\right)(\nu - 1)}. \tag{11}$$

Hence, in terms of the number of degrees of freedom ν and the conditional target entropy $H(Y|X)$, the MAE is

$$\mathrm{E}\left\{|Y - f(X)|\right\} = \frac{2}{(\nu - 1)\,B\left(\frac{1}{2}, \frac{\nu}{2}\right)^2} \exp\left\{H(Y|X) - \left(\frac{\nu + 1}{2}\right)\left[\Psi\left(\frac{\nu + 1}{2}\right) - \Psi\left(\frac{\nu}{2}\right)\right]\right\}. \tag{12}$$

As for the MSE, the MAE still depends on the number of degrees of freedom ν and cannot be written in sole terms of the conditional target entropy. This is illustrated in Fig. 7 which shows the MAE in

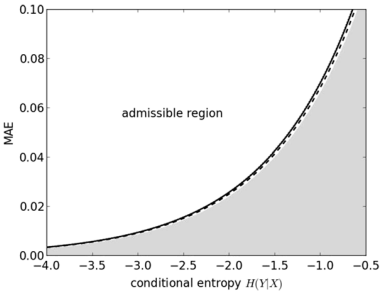


Fig. 7. MAE in terms of the conditional target entropy $H(Y|X)$ for Student estimation error with different numbers of degrees of freedom: $\nu = 2.3$ (dotted line), $\nu = 5$ (dashed line) or $\nu = 30$ (plain line). The admissible region defined by the Laplacian curve appears in white.

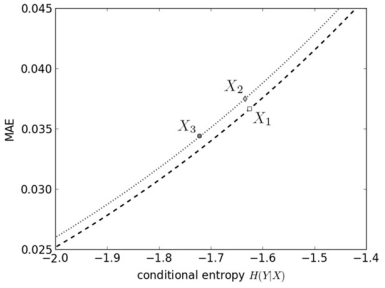


Fig. 8. Example of mutual information failure for Student estimation error with respect to the MAE. The candidate feature subsets correspond to different numbers of degrees of freedom: $\nu = 2.3$ (dotted line) and $\nu = 5$ (dashed line). The symbols X_i are feature subsets.

terms of the conditional target entropy $H(Y|X)$ for different numbers of degrees of freedom. The different curves are very close, yet they do not coincide and it is possible to decrease the conditional target entropy while increasing the MAE. Minimising the conditional target entropy $H(Y|X)$ does not necessarily correspond to minimising the MAE, but the problem is less important than for the MSE because the curves are very close from each other.

Fig. 8 shows an example of mutual information failure with respect to the MAE, where two candidate features subsets X_1 and X_2 are characterised by a Student estimation error with parameters $\nu = 2.3$ and $\nu = 5$, respectively. Using mutual information, X_2 will be chosen rather than X_1, since $H(Y|X_2)$ is smaller than $H(Y|X_1)$. However, selecting X_2 leads here to an increase in the MAE, which should rather be minimised. Fig. 8 also shows a counterexample: mutual information does not fail when choosing between the feature subsets X_1 and X_3, with two different Student estimation errors.

4.4. Short discussion

This section shows that mutual information is not always an adequate criterion to perform feature selection in regression. Indeed, when the estimation error has several parameters, it may be possible to obtain different values of the criterion for a given conditional target entropy $H(Y|X)$ (and *vice versa*). Intuitively, the knowledge of the value of the conditional target entropy $H(Y|X)$ only fixes one degree of freedom of the estimation error distribution parameters. Hence, problems may occur when the estimation error distribution is characterised by several parameters. In such a case, it becomes possible to select a feature subset which decreases the conditional target entropy with respect to other feature subsets while simultaneously increasing the MSE or the MAE. However, the impact of mutual information issues is likely to remain limited in terms of MSE and MAE for Student estimation errors. This is the case when the number of degrees of freedom is not too small, which is verified unless there is a large number of outliers.

5. Discussion

The examples in Section 3 show that mutual information is often a valuable criterion for feature selection in regression. Indeed, for realistic estimation errors with e.g. uniform, Laplacian or Gaussian distribution, choosing a feature subset which minimises the mutual information corresponds to minimising either the MSE or the MAE. Since it is often assumed (i) that the estimation error follows one of these distributions and (ii) that the MSE or the MAE is a sensible criterion, mutual information can in most cases be used safely. In fact, one can postulate that mutual information can be used whenever the estimation error is identically distributed for any $x \in \mathcal{X}$ and the estimation error distribution can be characterised by only one parameter.

Unfortunately, mutual information is not always optimal. Indeed, the example in Section 4 shows that when the estimation error distribution is characterised by multiple parameters like e.g. the Student distribution, it may be possible to obtain different values of the MSE or the MAE for a given value of the mutual information. Hence, minimising the mutual information does not necessarily correspond to minimising either the MSE or the MAE. However, it must be noticed that the impact of this issue may be of various importance. Indeed, in Section 4, the MSE can be quite different for a given conditional target entropy when the number of degrees of freedom of the Student distribution changes, whereas the difference remains quite small for the MAE. However, since it is not common to observe very small degrees of freedom (unless there is a large number of outliers), one can expect the impact of mutual information failures to remain small in practice, as discussed in Sections 4.2 and 4.4.

In Sections 3 and 4, the estimation error is assumed to be identically distributed for any $x \in \mathcal{X}$. However, this is not necessarily the case; the distribution of the estimation error may depend on x. For example, let us consider a simple estimation error which 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. In terms of the standard deviations σ_1 and σ_2, the conditional target entropies $H(Y|X)$ for this non-identically distributed (n.i.d.) Gaussian estimation error is $\frac{1}{2}\ln[2\pi e\sigma_1\sigma_2]$, whereas the MSE is $\frac{1}{2}\left(\sigma_1^2 + \sigma_2^2\right)$ and the MAE is $\frac{1}{\sqrt{2\pi}}(\sigma_1 + \sigma_2)$. Here, it is possible to rewrite the MSE and the MAE by replacing e.g. σ_2, what gives

$$\frac{1}{2}\left(\sigma_1^2 + \frac{\exp[4H(Y|X)]}{4\pi^2 e^2 \sigma_1^2}\right) \tag{13}$$

for the MSE and

$$\frac{1}{\sqrt{2\pi}}\left(\sigma_1 + \frac{\exp[2H(Y|X)]}{2\pi e\sigma_1}\right) \tag{14}$$

for the MAE. Hence, for a given value of the conditional target entropy, it is possible to obtain feature subsets with different MSE or MAE values. In this setting, mutual information may therefore also fail.

6. Conclusion

The goal of this paper is to study the adequacy of mutual information for feature selection in regression. The conclusion is that mutual information remains optimal in many cases, yet may sometimes also give non-optimal results. On the one hand, mutual information is optimal for commonly assumed estimation error distributions like e.g. the uniform, Laplacian or Gaussian distributions. In such a case, if the estimation error is identically distributed for any $x \in X$, the feature subset with the maximum mutual information is always the feature subset with the lowest MSE or MAE. On the other hand, mutual information may select feature subsets with non-optimal MSE or MAE when e.g. a Student distribution can be assumed for the estimation error. In such a case, feature subsets with identical mutual information values may correspond to different MSE or MAE values, even if the importance of the mutual information failure remains limited in practice.

In practice, it seems that the nature of the estimation errors is an important factor to determine whether mutual information is optimal for feature selection in regression. Hence, any study using mutual information in this context should assess the hypotheses which can be made about the conditional estimation error.

Acknowledgements

The authors thank the ESANN'12 reviewers and attendees and the Neural Networks anonymous reviewers for their fruitful discussions and comments on the subject of this paper, in particular Pierre Dupont, Amaury Lendasse, Fabrice Rossi and Jochen J. Steil.

Gauthier Doquire is funded by a Belgian F.R.I.A. grant.

References

Archambeau, C., Delannay, N., & Verleysen, M. (2008). Mixtures of robust probabilistic principal component analyzers. *Neurocomputing*, *71*(7–9), 1274–1282.

Ash, R. (1990). *Information theory*. Dover Publications.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, *5*, 537–550.

Bellman, R. E. (1961). *Adaptive control processes—a guided tour*. Princeton University Press.

Chen, B., Hu, J., Li, H., & Sun, Z. (2008). Adaptive filtering under maximum mutual information criterion. *Neurocomputing*, *71*(16–18), 3680–3684.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory* (1st ed.). Wiley-Interscience.

Dijck, G. V., & Hulle, M. M. V. (2006). Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis. In *Proceedings of the 16th international conference on artificial neural networks* (pp. 31–40). Springer.

Emmert-Streib, F., & Dehmer, M. (2008). *Information theory and statistical learning*. Springer.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, *5*, 1531–1555.

Frénay, B., Doquire, G., & Verleysen, M. (2012). On the potential inadequacy of mutual information for feature selection. In *Proceedings of ESANN*.

Frénay, B., Doquire, G., & Verleysen, M. (2013). Theoretical and empirical study on the potential inadequacy of mutual information for feature selection in classification. *Neurocomputing*, *112*, 64–78.

Globerson, A., & Tishby, N. (2003). Sufficient dimensionality reduction. *Journal of Machine Learning Research*, *3*, 1307–1331.

Guo, D., Shamai, S., & Verdú, S. (2005). Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory*, *51*(4), 1261–1282.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Ihara, S. (1993). *Information theory for continuous system*. World Scientific.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*, 273–324.

Kojadinovic, I., & Wottka, T. (2000). Comparison between a filter and a wrapper approach to variable subset selection in regression problems. In *Proceedings of ESIT*.

Kotz, S., Kozubowski, T., & Podgórski, K. (2001). *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Birkhäuser.

Kozachenko, L. F., & Leonenko, N. (1987). Sample estimate of the entropy of a random vector. *Problems of Information Transmission*, *23*, 95–101.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, *69*, 066138.

Peel, D., & McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, *10*(4), 339–348.

Psarakis, S., & Panaretos, J. (1990). The folded t distribution. *Communications in Statistics: A Theory and Methods*, *19*(7), 2717–2734.

Rossi, F., François, D., Wertz, V., Meurens, M., & Verleysen, M. (2007). Fast selection of spectral variables with b-spline compression. *Chemometrics and Intelligent Laboratory Systems*, *86*(2), 208–218. 4.

Rossi, F., Lendasse, A., Francois, D., Wertz, V., & Verleysen, M. (2006). Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics and Intelligent Laboratory Systems*, *80*, 215–226.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379–423.

Verleysen, M. (2003). Learning high-dimensional data. In *Limitations and future trends in neural computation, Vol. 186* (pp. 141–162).

Chapter 10

Classification in the Presence of Label Noise: a Survey

The following article has been submitted to the IEEE Transactions on Neural Networks and Learning Systems journal. This paper proposes a comprehensive survey on the different types of label noise, their consequences and the algorithms that take label noise into account. Related papers about label noise include [3, 5, 6].

Classification in the Presence of Label Noise: a Survey

Benoît Frénay and Michel Verleysen, *Senior Member, IEEE*

***Abstract*—Label noise is an important issue in supervised classification, with many potential negative consequences. For example, the accuracy of predictions may decrease, whereas the complexity of inferred models and the number of necessary training samples may increase. Many works in the literature have been devoted to the study of label noise and the development of techniques to deal with label noise. However, the field lacks a comprehensive survey on the different types of label noise, their consequences and the algorithms that take label noise into account. This paper proposes to fill this gap. Firstly, the definitions and sources of label noise are considered and a taxonomy of the types of label noise is proposed. Secondly, the potential consequences of label noise are discussed. Thirdly, label noise-robust, label noise cleansing and label noise-tolerant algorithms are reviewed. For each category of approaches, a short discussion is proposed in order to help the practitioner to choose the most suitable technique in its own particular field of application. Eventually, the design of experiments is also discussed, what may interest the researchers who would like to test their own algorithms. In this survey, label noise consists of mislabelled instances: no additional information is assumed to be available, like e.g. confidences on labels.**

***Index Terms*—classification, label noise, class noise, mislabelling, robust methods, survey.**

I. Introduction

CLASSIFICATION has been widely studied in machine learning. In that context, the standard approach consists in learning a classifier from a labelled dataset, in order to predict the class of new samples. However, real-word datasets may contain noise, which is defined in [1] as anything that obscures the relationship between the features of an instance and its class. In [2], noise is also described as consisting of non-systematic errors. Among other consequences, many works have shown that noise can adversely impact the classification performances of induced classifiers [3]. Hence, the ubiquity of noise seems to be an important issue for practical machine learning, e.g. in medical applications where most medical diagnosis tests are not 100 percent accurate and cannot be considered a gold standard [4]–[6]. Indeed, classes are not always as easy to distinguish as *lived* and *died* [4]. It is therefore necessary to implement techniques which eliminate noise or reduce its consequences. It is all the more necessary since reliably labelled data are often expensive and time consuming to obtain [4], what explains the commonness of noise [7].

In the literature, two types of noise are distinguished: feature (or attribute) noise and class noise [2], [3], [8]. On the one hand, feature noise affects the observed values of the features, e.g. by adding a small Gaussian noise to each feature during measurement. On the other hand, class noise alters the observed labels assigned to instances, e.g. by incorrectly setting a negative label on a positive instance in binary classification. In [3], [9], it is shown that class noise is potentially more harmful than feature noise, what highlights the importance of dealing with this type of noise. The prevalence of the impact of label noise is explained by the fact 1) that there are many features, whereas there is only one label and 2) that the importance of each feature for learning is different, whereas labels always have a large impact on learning. Similar results are obtained in [2]: feature noise appears to be less harmful than class noise for decision trees, except when a large number of features are polluted by feature noise.

Even if there exists a large literature about class noise, the field still lacks a comprehensive survey on the different types of label noise, their consequences and the algorithms that take label noise into account. This work proposes to cover the class noise literature. In particular, the different definitions and consequences of class noise are discussed, as well as the different families of algorithms which have been proposed to deal with class noise. As in outlier detection, many techniques rely on noise detection and removal algorithms, but it is shown that more complex methods have emerged. Existing datasets and data generation methods are also discussed, as well as experimental considerations.

In this work, class noise refers to observed labels which are incorrect. It is assumed that no other information is available, contrarily to other contexts where experts can e.g. provide a measure of confidence or uncertainty on their own labelling or answer with sets of labels. It is important to make clear that only the observed label of an instance is affected, not its true class. For this reason, class noise is called here label noise.

The survey is organised as follows. Section II discusses several definitions and sources of label noise, as well as a new taxonomy inspired by [10]. The potential consequences of label noise are depicted in Section III. Section IV distinguishes three types of approaches to deal with label noise: label noise-robust methods, label noise cleansing methods and label noise-tolerant methods. The three families of methods are discussed in Sections V, VI and VII, respectively. Section VIII discusses the design of experiments in the context of label noise and Section IX concludes the survey.

The authors are with the ICTEAM institute, Université catholique de Louvain, Place du Levant 3, 1348 Louvain-la-Neuve, Belgium. E-mails: {benoit.frenay, michel.verleysen}@uclouvain.be.

II. Definition, Sources and Taxonomy of Label Noise

Label noise is a complex phenomenon, as shown in this section. First, Section II-A defines label noise and specifies the scope of the survey. Similarities and differences with outliers and anomalies are also highlighted, since outlier detection methods can be used to detect mislabelled instances. Next, Section II-B reviews various sources of label noise, including insufficient information, expert labelling errors, subjectivity of the classes and encoding and communication problems. Eventually, a taxonomy of the types of label noise is proposed in Section II-C in order to facilitate further discussions. The proposed taxonomy highlights the potentially complex relationships between the features of instances, their true class and their observed label. This complexity should be taken into account when designing algorithms to deal with label noise, for they should be adapted to the characteristics of label noise.

A. Definition of Label Noise and Scope of the Survey

Classification consists in predicting the class of new samples, using a model inferred from training data. In this survey, it is assumed that each training sample is associated with an observed label. This label often corresponds to the true class of the sample, but it may be *subjected to a noise process before being presented to the learning algorithm* [11]. It is therefore important to distinguish the true class of an instance from its observed label. The process which pollutes labels is called label noise and must be separated from feature (or attribute) noise [2], [3], [8] which affects the value of features. Some authors also consider outliers which are correctly labelled as label noise [12], what is not done here.

In this survey, label noise is considered to be a stochastic process, i.e. the case where the labelling errors may be intentionally (like e.g. in the food industry [13]–[16]) and maliciously induced by an adversary agent [17]–[26] is not considered. Moreover, labelling errors are assumed to be independent from each other [11]. Edmonds [27] shows that noise in general is a complex phenomenon. In some very specific contexts, stochastic label noise can be intentionally introduced e.g. to protect people privacy, in which case its characteristics are completely under control [28]. However, a fully specified model of label noise is usually not available, what explains the need for automated algorithms which are able to cope with label noise. Learning situations where label noise occurs can be called *imperfectly supervised*, i.e. *pattern recognition applications where the assumption of label correctness does not hold for all the elements of the training sample* [29]. Such situations are between supervised and unsupervised learning.

Dealing with label noise is closely related to outlier detection [30]–[33] and anomaly detection [34]–[38]. Indeed, mislabelled instances may be outliers, if their label has a low probability of occurrence in their vicinity. Similarly, such instances may also look anomalous, with respect to the class which corresponds to their incorrect label. Hence, it is natural that many techniques in the label noise literature are very close to outlier and anomaly detection techniques; this is detailed in Section VI. In fact, many of the methods which have been developed to deal with outliers and anomalies can also be used to deal with label noise (see e.g. [39], [40]). However, it must be highlighted that mislabelled instances are not necessarily outliers or anomalies, which are subjective concepts [41]. For example, if labelling errors occur in a boundary region where all classes are equiprobable, the mislabelled instances neither are rare events nor look anomalous. Similarly, an outlier is not necessarily a mislabelled sample [42], since it can be due to feature noise or simply be a low-probability event.

B. Sources of Label Noise

As outlined in [1], the identification of the source(s) of label noise is not necessarily important, when the focus of the analysis is on the consequences of label noise. However, when a label noise model has to be embedded directly into the learning algorithm, it may be important to choose a modelling which accurately explains the actual label noise.

Label noise naturally occurs when human experts are involved [43]. In that case, possible causes of label noise include imperfect evidence, patterns which may be confused with the patterns of interest, perceptual errors or even biological artefacts. See e.g. [44], [45] for a philosophical account on probability, imprecision and uncertainty. More generally, potential sources of label noise include four main classes.

Firstly, the information which is provided to the expert may be insufficient to perform reliable labelling [1], [46]. For example, the results of several tests may be unknown in medical applications [12]. Moreover, the description language may be too limited [47], what reduces the amount of available information. In some cases, the information is also of poor or variable quality. For example, the answers of a patient during anamnesis may be imprecise or incorrect or even may be different if the question is repeated [48].

Secondly, as mentioned above, errors can occur in the expert labelling itself [1]. Such classification errors are not always due to human experts, since automated classification devices are used nowadays in different applications [12]. Also, since collecting reliable labels is a time-consuming and costly task, there is an increasing interest in using cheap, easy-to-get labels from non-expert using frameworks like e.g. the Amazon Mechanical Turk[1] [49]–[52]. Labels provided by non-expert are less reliable, but Snow et al. [49] show that the wealth of available labels may alleviate this problem.

Thirdly, when the labelling task is subjective, like e.g. in medical applications [53] or image data analysis [54], [55], there may exist an important variability in the labelling by several experts. For example, in electrocardiogram analysis, experts seldom agree on the exact boundaries of signal patterns [56]. The problem of inter-expert variability was also noticed during the labelling of the Penn Treebank, an annotated corpus of over 4.5 million words [57].

Eventually, label noise can also simply come from data encoding or communication problems [3], [11], [46]. For example, in spam filtering, sources of label noise include *misunderstanding the feedback mechanisms and accidental click* [58]. Real-world databases are estimated to contain around five

[1]https://www.mturk.com

percents of encoding errors, all fields taken together, when no specific measures are taken [59]–[61].

C. Taxonomy of Label Noise

In the context of missing values, Schafer and Graham [10] discuss a taxonomy which is adapted below to provide a new taxonomy for label noise. Similarly, Nettleton et al. [62] characterise noise generation in terms of its distribution, the target of the noise (features, label, etc.) and whether its magnitude depends on the data value of each variable. Since it is natural to consider label noise from a statistical point of view, Fig. 1 shows three possible statistical models of label noise. In order to model the label noise process, four random variables are depicted: X is the vector of features, Y is the true class, $\tilde{Y}$ is the observed label and E is a binary variable telling whether a labelling error occurred ($Y \neq \tilde{Y}$). The set of possible feature values is $\mathcal{X}$, whereas the set of possible classes (and labels) is $\mathcal{Y}$. Arrows report statistical dependencies: for example, $\tilde{Y}$ is assumed to always depend on Y (otherwise, there is no sense in using the labels).

1) The Noisy Completely at Random Model: In Fig. 1(a), the relationship between Y and $\tilde{Y}$ is called *noisy completely at random* (NCAR): the occurrence of an error E is independent of the other random variables, including the true class itself. In the NCAR case, the observed label is different from the true class with a probability $p_e = P(E = 1) = P(Y \neq \tilde{Y})$ [11], sometime called the error rate or the noise rate [63]. In the case of binary classification, NCAR noise is necessarily symmetric: the same percentage of instances are mislabelled in both classes. When $p_e = \frac{1}{2}$, the labels are useless, since they no longer carry any information [11]. The NCAR setting is similar to the absent-minded professor discussed in [64].

In the case of multiclass classification, it is usually assumed that the incorrect label is chosen at random in $\mathcal{Y} \setminus \{y\}$ when $E = 1$ [11], [65]. In other words, a biased coin is firstly flipped in order to decide whether the observed label is correct or not. If the label is wrong, a fair dice with $|\mathcal{Y}| - 1$ faces (where $|\mathcal{Y}|$ is the number of classes) is tossed to choose the observed, wrong label. This particularly simple model is called the *uniform label noise*.

2) The Noisy at Random Model: In Fig. 1(b), it is assumed that the probability of error depends on the true class Y, what is called here *noisy at random* (NAR). E is still independent of X, but this model allows modelling asymmetric label noise, i.e. when instances from certain classes are more prone to be mislabelled. For example, in medical case-control studies, control subjects may be more likely to be mislabelled. Indeed, the test which is used to label case subjects may be too invasive (e.g. a biopsy) or too expensive to be used on control subjects and is therefore replaced by a suboptimal diagnostic test for control subjects [66]. Since one can define the labelling probabilities

$$P(\tilde{Y} = \tilde{y}|Y = y) = \sum_{e \in \{0,1\}} P(\tilde{Y} = \tilde{y}|E = e, Y = y)P(E = e|Y = y), \quad (1)$$

the NAR label noise can equivalently be characterised in terms of the labelling (or transition) matrix [67], [68]

$$\gamma = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1n_{\mathcal{Y}}} \\ \vdots & \ddots & \vdots \\ \gamma_{n_{\mathcal{Y}}1} & \cdots & \gamma_{n_{\mathcal{Y}}n_{\mathcal{Y}}} \end{pmatrix} = \begin{pmatrix} P(\tilde{Y} = 1|Y = 1) & \cdots & P(\tilde{Y} = n_{\mathcal{Y}}|Y = 1) \\ \vdots & \ddots & \vdots \\ P(\tilde{Y} = 1|Y = n_{\mathcal{Y}}) & \cdots & P(\tilde{Y} = n_{\mathcal{Y}}|Y = n_{\mathcal{Y}}) \end{pmatrix} \quad (2)$$

where $n_{\mathcal{Y}} = |\mathcal{Y}|$ is the number of classes. Each row of the labelling matrix must sum to 1, since $\sum_{\tilde{y} \in \mathcal{Y}} P(\tilde{Y} = \tilde{y}|Y = y) = 1$. For example, the uniform label noise corresponds to the labelling matrix

$$\begin{pmatrix} 1 - p_e & \cdots & \frac{p_e}{n_{\mathcal{Y}}-1} \\ \vdots & \ddots & \vdots \\ \frac{p_e}{n_{\mathcal{Y}}-1} & \cdots & 1 - p_e \end{pmatrix}. \quad (3)$$

Notice that NCAR label noise is a special case of NAR label noise. When true classes are known, the labelling probabilities can be directly estimated by the frequencies of mislabelling in data, but it is seldom the case [48]. Alternately, one can also use the incidence-of-error matrix [48]

$$\begin{pmatrix} \pi_1\gamma_{11} & \cdots & \pi_1\gamma_{1n_{\mathcal{Y}}} \\ \vdots & \ddots & \vdots \\ \pi_{n_{\mathcal{Y}}}\gamma_{n_{\mathcal{Y}}1} & \cdots & \pi_{n_{\mathcal{Y}}}\gamma_{n_{\mathcal{Y}}n_{\mathcal{Y}}} \end{pmatrix} = \begin{pmatrix} P(Y = 1, \tilde{Y} = 1) & \cdots & P(Y = 1, \tilde{Y} = n_{\mathcal{Y}}) \\ \vdots & \ddots & \vdots \\ P(Y = n_{\mathcal{Y}}, \tilde{Y} = 1) & \cdots & P(Y = n_{\mathcal{Y}}, \tilde{Y} = n_{\mathcal{Y}}) \end{pmatrix} \quad (4)$$

where $\pi_y = P(Y = y)$ is the prior of class y. The entries of the incidence-of-error matrix sum to one and may be of more practical interest.

With the exception of uniform label noise, NAR label noise is the most commonly studied case of label noise in the literature. For example, Lawrence and Schölkopf [67] consider arbitrary labelling matrices. In [3], [69], pairwise label noise is introduced: 1) two classes c_1 and c_2 are selected, then 2) each instance of class c_1 has a probability to be incorrectly labelled as c_2 and vice versa. For this label noise, only two non-diagonal entries of the labelling matrix are non-zero.

In the case of NAR label noise, it is no longer trivial to decide whether the labels are helpful or not. One solution is to compute the expected probability of error

$$p_e = P(E = 1) = \sum_{y \in \mathcal{Y}} P(Y = y)P(E = 1|Y = y) \quad (5)$$

and to require that $p_e < \frac{1}{2}$, similarly to NCAR label noise. However, this condition does not prevent the occurrence of very small correct labelling probabilities $P(\tilde{Y} = y|Y = y)$ for some class $y \in \mathcal{Y}$, in particular if the prior probability $P(y)$ of this class is small. Instead, conditional error probabilities $p_e(y) = P(E = 1|Y = y)$ can also be used.

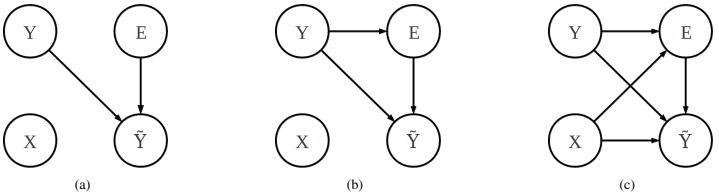


Fig. 1. Statistical taxonomy of label noise inspired by [10]: (a) noisy completely at random (NCAR), (b) noisy at random (NAR) and (c) noisy not at random (NNAR). Arrows report statistical dependencies. Notice the increasing complexity of statistical dependencies in the label noise generation models, from left to right. The statistical link between X and Y is not shown for clarity.

3) The Noisy not at Random Model: Most works on label noise consider that the label noise affects all instances with no distinction. However, it is not always realistic to assume the two above types of label noise [11], [70]. For example, samples may be more likely mislabelled when they are similar to instances of another class [70]–[76], as illustrated e.g. in [77] where empirical evidence is given that more difficult samples in a text entailment dataset are labelled randomly. It also seems natural to expect less reliable labels in regions of low density [78]–[80], where experts predictions may be actually based on a very small number of similar previously encountered cases.

Let us consider a more complex and realistic model of label noise. In Fig. 1(c), E depends on both variables X and Y, i.e. mislabelling is more probable for certain classes and in certain regions of the X space. This *noisy not at random* (NNAR) model is the most general case of label noise [81], [82]. For example, mislabelling near the classification boundary or in low density regions can only be modelled in terms of NNAR label noise. Such a situation occurs e.g. in speech recognition, where automatic speech recognition is more difficult in case of phonetic similarity between the correct word and the recognised word [83]. The context of each word can be considered in order to detect incorrect recognitions. Notice that the medical literature distinguishes differential (feature-dependent, i.e. NNAR) label noise and non-differential (feature-independent, i.e. NCAR or NAR) label noise [84].

The reliability of labels is even more complex to estimate that for NCAR or NAR label noise. Indeed, the probability of error also depends in that case on the value of X. As before, one can define an expected probability of error which becomes

$$p_e = P(E=1) = \sum_{y \in \mathcal{Y}} P(Y=y) \int_{x \in \mathcal{X}} P(X=x|Y=y) P(E=1|X=x, Y=y) dx \quad (6)$$

if X is continuous. However, this quantity does not reflect the local nature of label noise: in some cases, p_e can be almost zero although the density of labelling errors shows important peaks in certain regions. The quantity $p_e(x,y) = P(E=1|X=x, Y=y)$ may therefore be more appropriate to characterise the reliability of labels.

III. Consequences of Label Noise on Learning

In this section, the potential consequences of label noise are described to show the necessity to take label noise into account in learning problems. Section III-A reviews theoretical and empirical evidences of the impact of label noise on classification performances, which is the most frequently reported issue. Section III-B shows that the presence of label noise also increases the necessary number of samples for learning, as well as the complexity of models. Label noise may also pose a threat for related tasks, like e.g. class frequencies estimation and feature selection, which are discussed in Section III-C and Section III-D, respectively.

This section presents the negative consequences of label noise, but artificial label noise also has potential advantages. For example, label noise can be added in statistical studies to protect people privacy: it is e.g. used in [28] to obtain statistics for questionnaires, while making impossible to recover individual answers. In [85]–[89], label noise is added to improve classifier results. Whereas bagging produces different training sets by resampling, these works copy the original training set and switch labels in new training sets to increase the variability in data.

A. Deterioration of Classification Performances

The more frequently reported consequence of label noise is a decrease in classification performances, as shown in the theoretical and experimental works described below.

1) Theoretical Studies of Simple Classifiers: There exist several theoretical studies of the consequences of label noise on prediction performances. For simple problems and symmetric label noise, the accuracy of classifiers may remain unaffected. Lachenbruch [71] consider e.g. the case of binary classification when both classes have Gaussian distribution with identical covariance matrix. In such a case, a linear discriminant function can be used. For a large number of samples, the consequence of uniform noise is noticeable only if the error rates α_1 and α_2 in each class are different. In fact, the change in decision boundary is completely described in terms of the difference $\alpha_1 - \alpha_2$. These results are also discussed asymptotically in [90].

The results of Lachenbruch [71] are extended in [91] for quadratic discriminant functions, i.e. Gaussian conditional distributions with unequal covariance matrices. In that case,

prediction is affected even when label noise is symmetric among classes ($\alpha_1 = \alpha_2$). Consequences worsen when differences in covariance matrices or misclassification rates increase. Michalek and Tripathi [92] and Bi and Jeske [93] show that label noise affects normal discriminant and logistic regression: their error rates are increased and their parameters are biased. Logistic regression seems to be less affected.

In [64], the single-unit perceptron is studied in the presence of label noise. If the teacher providing learning samples is absent-minded, i.e. labels are flipped with a given probability (uniform noise), the performances of a learner who takes the labels for granted are damaged and even get worse than the performances of the teacher.

Classification performances of the k nearest neighbours (kNN) classifier are also affected by label noise [94], [95], in particular when $k = 1$ [96]. Okamoto and Nobuhiro [96] present an average-case analysis of the kNN classifier. When k is optimised, the consequences of label noise are reduced and remain small unless a large amount of label noise is added. The optimal value of k depends on both the number of training instances and the presence of label noise. For small noise-free training sets, 1NN classifiers are often optimal. But as soon as label noise is added, the optimal number of neighbours k is shown to monotonically increase with the number of instances even for small training sets, what seems natural since 1NN classifiers are particularly affected by label noise.

2) Experimental Assessment of Specific Models: Apart from theoretical studies, many works show experimentally that label noise may be harmful. First of all, the impact of label noise is not identical for all types of classifiers. As detailed in Section V, this fact can be used to cope (at least partially) with label noise. For example, Nettleton et al. [62] compare the impact of label noise on four different supervised learners: naive Bayes, decision trees induced by C4.5, kNNs and support vector machines (SVMs). In particular, naive Bayes achieves the best results, what is attributed to the conditional independence assumption and the use of conditional probabilities. This should be contrasted with the results in [12], where naive Bayes is sometime dominated by C4.5 and kNNs. The poor results of SVMs are attributed to its reliance on support vectors and the feature interdependence assumption.

In text categorization, Zhang and Yang [97] consider the robustness of regularized linear classification methods. Three linear methods are tested by randomly picking and flipping labels: linear SVMs, Ridge regression and logistic regression. The experiments show that the results are dramatically affected by label noise for all three methods, which obtain almost identical performances. Only 5% of flipped labels already leads to a dramatic decrease of performances, what is explained by the presence of relatively very small classes with only a few samples in their experiments.

Several studies have shown that boosting [98] is affected by label noise [99]–[102]. In particular, the adaptive boosting algorithm AdaBoost tends to spend too much efforts on learning mislabelled instances [100]. During learning, successive weak learners are trained and the weights of instances which are misclassified at one step are increased at the next step. Hence, in the late stages of learning, AdaBoost tends to increase the weights of mislabelled instances and starts overfitting [103], [104]. Dietterich [100] clearly shows that the mean weight per training sample becomes larger for mislabelled samples than for correctly labelled samples as learning goes on. Interestingly, it has been shown in [105]–[108] that AdaBoost *tends to increase the margins of the training examples* [109] and *achieves asymptotically a decision with hard margin, very similar to the one of SVMs for the separable case* [108]. This may not be a good idea in the presence label noise and may explain why AdaBoost overfits noisy training instances. In [110], it is also shown that ensemble methods can fail simply because the presence of label noise affects the ensembled models. Indeed, learning through multiple models becomes harder for large levels of label noise, where some samples *become more difficult for all models* and are therefore seldom correctly classified by an individual model.

In systems which learn Boolean concepts with disjuncts, Weiss [111] explains that small disjuncts (which individually cover only a few examples) are more likely to be affected by label noise than large disjuncts covering more instances. However, only large levels of label noise may actually be a problem. For decision trees, it appears in [2] that *destroying class information produces a linear increase in error*. Taking logic to extremes, *when all class information is noise, the resulting decision tree classifies objects entirely randomly*.

Another example studied in [58] is spam filtering where performances are decreased by label noise. Spam filters tend to overfit label noise, due to aggressive online update rules which are designed to quickly adapt to new spam.

3) Additional Results for More Complex Types of Label Noise: The above works deal with NAR label noise, but more complex types of label noise have been studied in the literature. For example, in the case of linear discriminant analysis (LDA), i.e. binary classification with normal class distributions, Lachenbruch [70] considers that mislabelling systematically occurs when samples are too far from the mean of their true class. In that NNAR label noise model, *the true probabilities of misclassification are only slightly affected*, whereas the populations are better separated. This is attributed to the reduction of the effects of outliers. However, the apparent error rate [112] of LDA is highly influenced, what may cause the classifier to overestimate its own efficiency.

LDA is also studied in the presence of label noise by [72], which generalises the results of [70], [71], [90]–[92]. Let us define 1) the misallocation rate α_y for class y, i.e. the number of samples with label y which belong to the other class and 2) a z-axis which passes through the center of both classes and is oriented towards the positive class, such that each center is located at $z = \pm\frac{\Delta}{2}$. In [72], three label noise models are defined and characterised in terms of the *probability of misallocation* $g_y(z)$, which is a monotone decreasing (increasing) function of z for positive (negative) samples. In random misallocation, $g_y(z) = \alpha_y$ is constant for each class, what is equivalent to the NAR label noise. In truncated label noise, $g(z)$ is zero as long as the instance is close enough to the mean of its class. Afterwards, the mislabelling probability is equal to a small constant. This type of NNAR label noise is equivalent to the model of

[70] when the constant is equal to one. Eventually, in the exponential model, the probability of misallocation becomes for the negative class

$$g_y(z) = \begin{cases} 0 & \text{if } z \leq -\frac{\Delta}{2} \\ 1 - \exp\left(-\frac{1}{2}k_y(z+\frac{\Delta}{2})^2\right) & \text{if } z > -\frac{\Delta}{2} \end{cases} \quad (7)$$

where Δ is the distance between the centres of both classes and $k_y = (1-2\alpha_y)^{-2}$. A similar definition is given for the positive class. For equivalent misallocation rates α_y, random misallocation has more consequences than truncated label noise, in terms of influence on the position and variability of the discriminant boundary. In turn, truncated label noise itself has more consequences than exponential label noise. The same ordering appears when comparing misclassification rates.

B. Consequences on Learning Requirements and Model Complexity

Label noise can affect learning requirements (e.g. number of necessary instances) or the complexity of learned models. For example, Quinlan [2] warns that the size of decision trees may increase in case of label noise, making them overly complicated, what is confirmed experimentally in [46]. Similarly, Abellán and Masegosa [104] show that the number of nodes of decision trees induced by C4.5 for bagging is increased, while the resulting accuracy is reduced. Reciprocally, Brodley and Friedl [46] and Libralon et al. [113] show that removing mislabelled samples reduces the complexity of SVMs (number of support vectors), decision trees induced by C4.5 (size of trees) and rule-based classifiers induced by RIPPER (number of rules). Post-pruning also seems to reduce the consequences of label noise [104]. Noise reduction can therefore produce models which are easier to understand, what *is desirable in many circumstances* [114]–[116]

In [11], it is shown that the presence of uniform label noise in the probably approximately correct (PAC) framework [117] increases the number of necessary samples for PAC identification. An upper bound for the number of necessary samples is given, which is strengthened in [118]. Similar bounds are also discussed in [65], [119]. Also, Angluin and Laird [11] discuss the feasibility of PAC learning in the presence of label noise for propositional formulas in conjunctive normal form (CNF), what is extended in [120] for Boolean functions represented by decision trees and in [73], [121] for linear perceptrons.

C. Distortion of Observed Frequencies

In medical applications, it is often necessary to perform medical tests for disease diagnosis, to estimate the prevalence of a disease in a population or to compare (estimated) prevalence in different populations. However, label noise can affect the observed frequencies of medical test results, what may lead to incorrect conclusions. For binary tests, Bross [4] shows that mislabelling may pose a serious threat: the observed mean and variance of the test answer is strongly affected by label noise. Let us consider a simple example taken from [4]: if the minority class represents 10% of the dataset and 5% of the test answers are incorrect (i.e. patients are mislabelled), the observed proportion of minority cases is $0.95\times10\%+0.05\times90\% = 14\%$ and is therefore overestimated by 40%. Significance tests which assess the difference between the proportions of both classes in two populations are still valid in case of mislabelling, but their power may be strongly reduced. Similar problems occur e.g. in consumer survey analysis [122].

Frequency estimates are also affected by label noise in multiclass problems. Hout and Heijden [28] discuss the case of artificial label noise, which can be intentionally introduced after data collection in order to preserve privacy. Since the label noise is fully specified in this case, it is possible to adjust the observed frequencies. When a model of the label noise is not available, Tenenbein [123] proposes to solve the problem pointed by [4] using double sampling, which uses two labellers: an expensive, reliable labeller and a cheap, unreliable labeller. The model of mislabelling can thereafter be learned from both sets of labels [124], [125]. In [48], the case of multiple experts is discussed in the context of medical anamnesis; an algorithm is proposed to estimate the error rates of the experts.

Evaluating the error rate of classifiers is also important for both model selection and model assessment. In that context, Lam and Stork [126] show that label noise can have an important impact on the estimated error rate, when test samples are also polluted. Hence, mislabelling can also bias model comparison. As an example, a spam filter *with a true error rate of* 0.5%*, for example, might be estimated to have an error rate between* 5.5% *and* 6.5% *when evaluated using labels with an error rate of* 6.0%*, depending on the correlation between filter and label errors* [127].

D. Consequences for Related Tasks

The aforementioned consequences are not the only possible consequences of label noise. For example, Zhang et al. [128] show that the consequences of label noise are important in feature selection for microarray data. In an experiment, only one mislabelled sample already leads to about 20% of not identified discriminative genes. Notice that in microarray data, only a few data are available. Similarly, Shanab et al. [129] show that label noise decreases the stability of feature rankings. The sensitivity of feature selection to label noise is also illustrated for logistic regression in [130]. A methodology to achieve feature selection for classification problems polluted by label noise is proposed in [131], based on a probabilistic label noise model combined with a nearest neighbours-based estimator of the mutual information.

E. Conclusion

This section shows that the consequences of label noise are important and diverse: decrease in classification performances, changes in learning requirements, increase in the complexity of learned models, distortion of observed frequencies, difficulties to identify relevant features, etc. The nature and the importance of the consequences depend, among others, on the type and the level of label noise, the learning algorithm and the characteristics of the training set. Hence, it seems important

for the machine learning practitioner to deal with label noise and to consider these factors, prior to the analysis of polluted data.

IV. Methods to Deal with Label Noise

In light of the various consequences detailed in Section III, it seems important to deal with label noise. In the literature, there exist three main approaches to take care of label noise [12], [82], [132]–[137]; these approaches are described below. Manual review of training samples is not considered in this survey, because it is usually prohibitively costly and time consuming, if not impossible in the case of large datasets.

A first approach relies on algorithms which are naturally robust to label noise. In other words, the learning of the classifier is assumed to be not too sensitive to the presence of label noise. Indeed, several studies have shown that some algorithms are less influenced than others by label noise, what advocates for this approach. However, label noise is not really taken into account in this type of approach. In fact, label noise handling is entrusted to overfitting avoidance [132]–[134].

Secondly, one can try to improve the quality of training data using filter approaches. In such a case, noisy labels are typically identified and being dealt with before training occurs. Mislabelled instances can either be relabelled or simply removed [138]. Filter approaches are cheap and easy to implement, but some of them are likely to remove a substantial amount of data.

Eventually, there exist algorithms which directly model label noise during learning or which have been modified to take label noise into account in an embedded fashion. The advantage of this approach is to separate the classification model and the label noise model, what allows using information about the nature of label noise.

The literature for the three above trends of approaches is reviewed in the three next sections. In some cases, it is not always clear whether an approach belongs to one category or the other. For example, some of the label noise-tolerant variants of SVMs could also be seen as filtering. At the end of each section, a short discussion of the strengths and weaknesses of the described techniques is proposed, in order to help the practitioner in its choice. The three following sections are strongly linked with Section III. Indeed, the knowledge of the consequences of label noise allows one to avoid some pitfalls and to design algorithms which are more robust or tolerant to label noise. Moreover, the consequences of label noise themselves can be used to detect mislabelled instances.

V. Label Noise-Robust Models

This section describes models which are robust to the presence of label noise. Even if label noise is neither cleansed nor modelled, such models have been shown to remain relatively effective when training data are corrupted by small amounts of label noise. Label noise-robustness is discussed from a theoretical point of view in Section V-A. Then, the robustness of ensembles methods and decision trees are considered in Section V-B and V-C, respectively. Eventually, various other methods are discussed in Section V-D and Section V-E concludes about the practical use of label noise-robust methods.

A. Theoretical Considerations on the Robustness of Losses

Before we turn to empirical results, a first, fundamental question is whether it is theoretically possible (and under what circumstances) to achieve perfect label noise-robustness. In order to have a general view of label noise-robustness, Manwani and Sastry [82] study learning algorithms in the empirical risk minimisation (ERM) framework for binary classification. In ERM, the cost of wrong predictions is measured by a loss and classifiers are learned by minimising the expected loss for future samples, which is called the risk. The more natural loss is the 0-1 loss, which gives a cost of 1 in case of error and is zero otherwise. However, the 0-1 loss is neither convex nor differentiable, what makes it intractable for real learning algorithms. Hence, others losses are often used in practice, which approximate the 0-1 loss by a convex function, called a surrogate [139].

In [82], risk minimisation under a given loss function is defined as label noise-robust if the probability of misclassification of inferred models is identical, irrespective of label noise presence. It is demonstrated that the 0-1 loss is label noise-robust for uniform label noise [140] or when it is possible to achieve zero error rate [81]; see e.g. [74] for a discussion in the case of NNAR label noise. The least-square loss is also robust to uniform label noise, which guarantees the robustness of the Fisher linear discriminant in that specific case. Other well-known losses are shown to be not robust to label noise, even in the uniform label noise case: 1) the exponential loss, which leads to AdaBoost, 2) the log loss, which leads to logistic regression and 3) the hinge loss, which leads to support vector machines. In other words, one can expect most of the recent learning algorithms in machine learning to be not completely label noise-robust.

B. Ensemble Methods: Bagging and Boosting

In the presence of label noise, bagging achieves better results than boosting [100]. On the one hand, mislabelled instances are characterised by large weights in AdaBoost, which spends too much effort in modelling noisy instances [104]. On the other hand, mislabelled samples increase the variability of the base classifiers for bagging. Indeed, since each mislabelled sample has a large impact on the classifier and bagging repeatedly selects different subsets of training instances, each resampling leads to a quite different model. Hence, the diversity of base classifiers is improved in bagging, whereas the accuracy of base classifiers in AdaBoost is severely reduced.

Several algorithms have been shown to be more label noise-robust than AdaBoost [101], [102], e.g. LogitBoost [141] and BrownBoost [142]. In [108], [143]–[145], boosting is casted as a margin maximisation problem and slack variables are introduced in order to allow a given fraction of patterns to stand in the margin area. Similarly to soft-margin SVMs, these works propose to allow boosting to misclassify some of the training samples, what is not directly aimed at dealing with label noise but robustifies boosting. Moreover, this approach can be used to find difficult or informative patterns [145].

C. Decision trees

It is well-known that decision trees are greatly impacted by label noise [2], [104]. In fact, their instability makes them well suited for ensemble methods [146]–[148]. In [148], different node split criteria are compared for ensembles of decision trees in the presence of label noise. The imprecise info-gain [149] is shown to improve accuracy, with respect to the information gain, the information gain ratio and the Gini index. Compared to ensembles of decision trees inferred by C4.5, Abellán and Masegosa [104] also show that the imprecise info-gain allows reducing the size of the decision trees. Eventually, they observe that post-pruning of decision trees can reduce the impact of label noise. The approach is extended for continuous features and missing data in [150].

D. Other Methods

Most of the studies on label noise robustness have been presented in Section III. They show that complete label noise robustness is seldom achieved, as discussed in Section V-A. An exception is [81], where the 0-1 loss is directly optimised using a team of continuous-action learning automata: 1) a probability distribution is defined on the weights of a linear classifier, then 2) weights are repetitively drawn from the distribution to classify training samples and 3) the 0-1 losses for the training samples are used at each iteration as a reinforcement to progressively tighten the distribution around the optimal weights. In the case of separable classes, the approach converges to the true optimal separating hyperplane, even in the case of NNAR label noise. In [151], eleven classifiers are compared on imbalanced datasets with asymmetric label noise. In all cases, the performances of the models are affected by label noise. Random forests [147] are shown to be the most robust among the eleven methods, what is also the case in another study by the same authors [152]. C4.5, radial basis function (RBF) networks and rule-based classifiers obtain the worst results. The sensitivity of C4.5 to label noise is confirmed in [153], where multilayer perceptrons are shown to be less affected. In [135], a new artificial immune recognition system (AIRS) is proposed, called RWTSAIRS, which is shown to be less sensitive to label noise. In [154], two procedures based on argumentation theory are also shown to be robust to label noise. In [12], it is shown that feature extraction can help to reduce the impact of label noise. Also, Sàez et al. [9], [155] shows that using one-vs-one decomposition in multiclass problems can improve the robustness, which could be due to the *distribution of the noisy examples in the subproblems*, the *increase of the separability of the classes* and *collecting information from different classifiers*.

E. Discussion

Theoretically, common losses in machine learning are not completely robust to label noise [139]. However, overfitting avoidance techniques like e.g. regularisation can be used to partially handle label noise [132]–[134], even if label noise *may interfere with the quality of the classifier, whose accuracy might suffer and the representation might be less compact* [132]. Experiments in the literature show that the performances of classifiers inferred by label noise-robust algorithms are still affected by label noise. Label noise-robust methods seem to be adequate only for simple cases of label noise, which can be safely managed by overfitting avoidance.

VI. Data Cleansing Methods for Label Noise-Polluted Datasets

When training data is polluted by label noise, an obvious and tempting solution consists in cleansing the training data themselves, what is similar to outlier or anomaly detection. However, detecting mislabelled instances is seldom trivial: Weiss and Hirsh [156] show e.g. in the context of learning with disjuncts that true exceptions may be hard to distinguish from mislabelled instances. Hence, many methods have been proposed to cleanse training sets, with different degrees of success. The whole procedure is illustrated by Fig. 2, which is inspired by [46]. This section describes several methods which detect, remove or relabel mislabelled instances. First, simple methods based on thresholds are presented in Section VI-A. Model prediction-based filtering methods are discussed in Section VI-B, which includes classification filtering, voting filtering and partition filtering. Methods based on measures of the impact of label noise and introspection are considered in Section VI-C. Sections VI-D, VI-E and VI-F address methods based on nearest neighbours, graphs and ensembles. Eventually, several other methods are discussed in Section VI-G and a general discussion about data cleansing methods is proposed in Section VI-H.

A. Measures and Thresholds

Similarly to outlier detection [30]–[33] and anomaly detection [34]–[38], several methods in label noise cleansing are based on ad hoc measures. Instances can e.g. be removed when the *anomaly measure* exceeds a predefined threshold. For example, in [157], the entropy of the conditional distribution $P(Y|X)$ is estimated using a probabilistic classifier. Instances with a low entropy correspond to confident classifications. Hence, such instances for which the classifier disagrees with the observed label are relabelled using the predicted label.

As discussed in Section III, label noise may increase the complexity of inferred models. Therefore, complexity measures can be used to detect mislabelled instances, which disproportionately increase model complexity when added to the training set. In [158], the complexity measure for inductive concept learning is the number of literals in the hypothesis. A cleansing algorithm is proposed, which 1) finds for each literal the minimal set of training samples whose removal would allow going without the literal and 2) awards one point to each sample in the minimal set. Once all literals have been reviewed, the sample with the higher score is removed, if the score is high enough. This heuristic produces less complex models. Similarly, Gamberger and Lavrač [159] measure the complexity of the least complex correct hypothesis (LCCH) for a given training set. Each training set is characterised by a LCCH value and is *saturated* if its LCCH value is equal to the complexity of the target hypothesis. Mislabelled samples

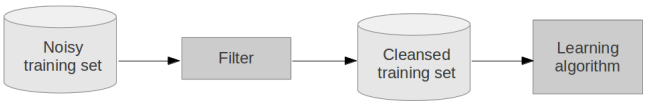


Fig. 2. General procedure for learning in the presence of label noise with training set cleansing, inspired by [46].

are removed to obtain a saturated training set. Gamberger et al. [160]–[162] elaborate on the above notions of complexity and saturation, which result in the so-called saturation filter.

B. Model Predictions-Based Filtering

Several data cleansing algorithms rely on the predictions of classifiers: classification filtering, voting filtering and partition filtering. In [163], such methods are extended in the context of cost-sensitive learning, whereas Khoshgoftaar and Rebours [164] propose a generic algorithm which can be specialised to classification filtering, voting filtering or partition filtering by a proper choice of parameters.

1) Classification Filtering: The predictions of classifiers can be used to detect mislabelled instances, what is called *classification filtering* [161], [164]. For example, Thongkam et al. [165] learn a SVM using the training data and removes all instances which are misclassified by the SVM. A similar method is proposed in [166] for neural networks. Miranda et al. [167] extend the approach of [165]: four classifiers are induced by different machine learning techniques and are combined by voting to detect mislabelled instances. The above methods can be applied to any classifier, but it eliminates all instances which stand on the wrong side of the classification boundary, what be can dangerous [168], [169]. In fact, as discussed in [170], classification filtering (and data cleansing in general) suffers from a *chicken-and-egg* dilemma, since 1) good classifiers are necessary for classification filtering and 2) learning in the presence of label noise may precisely produce poor classifiers. An alternative is proposed in [169], which 1) defines a pattern as informative if *it is difficult to predict by a model trained on previously seen data* and 2) sent a pattern to the human operator for checking if its informativeness is above a threshold found by cross-validation. Indeed, such patterns can either be atypical *patterns that are actually informative or garbage patterns*. The *level of surprise is* considered to be *a good indication of how informative a pattern is*, what is quantified by the information gain $-\log P(Y = y|X = x)$.

In [171], an iterative procedure called robust-C4.5 is introduced. At each iteration, 1) a decision tree is inferred and pruned by C4.5 and 2) training samples which are misclassified by the pruned decision tree are removed. The procedure is akin to regularisation, in that the model is repeatedly made simpler. Indeed, each iteration removes training samples, what in turn allows C4.5 to produce smaller decision trees. Accuracy is slightly improved, whereas the mean and variance of the tree size are decreased. Hence, smaller and more stable decision trees are obtained, which also perform better. Notice that caution is advised when comparing sizes of decision trees in data cleansing [172], [173]. Indeed, Oates and Jensen [172] show that the size of decision trees naturally tends to increase linearly with the number of instances. It means that the removal of randomly selected training samples already leads to a decrease in tree sizes. Therefore, Oates and Jensen [172] propose the measure

$$100 \times \left(\frac{\text{initial tree size} - \text{tree size with random filtering}}{\text{initial tree size} - \text{tree size with studied filtering}} \right) \tag{8}$$

to estimate the percentage of decrease in tree size which is simply due to a reduction in the number of samples. For example, Oates and Jensen [172] show experimentally for robust-C4.5 that 42% of the decrease in tree size can be imputed to the sole reduction in training set size, whereas the remaining 58% are due to an appropriate choice of the instances to be removed. A similar analysis could be done for other methods in this section.

Local models [174] can also be used to filter mislabelled training samples. Such models are obtained by training a standard model like e.g. LDA [175] or a SVM [176], [177] on a training set consisting of the k nearest neighbours of the sample to be classified. Many local models have to be learnt, but the respective local training sets are very small. In [116], local SVMs are used to reject samples for which the prediction is not confident enough. In [115], the local SVM noise reduction method is extended for large datasets, by reducing the number of SVMs to be trained. In [178], a sample is removed if it is misclassified by a k nearest centroid neighbours classifier [179] trained when the sample itself is removed from the training set.

2) Voting Filtering: Classification filtering faces the risk to remove too many instances. In order to solve this problem, ensembles of classifiers are used in [46], [138], [180] to identify mislabelled instances, what is inspired by outlier removal in regression [181]. The first step consists in using a K-fold cross-validation scheme, which creates K pairs of distinct training and validation datasets. For each pair of sets, m learning algorithms are used to learn m classifiers using the training set and to classify the samples in the validation set. Therefore, m classifications are obtained for each sample, since each instance belongs to exactly one validation set. The second step consists in inferring from the m predictions whether a sample is mislabelled or not, what is called voting filtering in [173] or ensemble filtering in [164]. Two possibilities are studied in [46], [138], [180]: a majority vote and a consensus vote. Whereas majority vote classifies a sample as mislabelled if a majority of the m classifiers misclassified it, the consensus vote requires that all classifiers have misclassified the sample. One can also require high agreement of classifiers, i.e. misclassification by more than a given percentage of the classifiers [182]. The consensus

vote is more conservative than the majority vote and results in fewer removed samples. The majority vote tends to throw out too many instances [183], but performs better than consensus vote, because keeping mislabelled instances seems to harm more than removing too many correctly labelled samples.

The K-fold cross-validation is also used in [161]. For each training set, a classifier is learnt and directly filters its corresponding validation set. The approach is intermediate between [165] and [46], [138], [180] and has been shown to be non-selective, i.e. *too many samples are detected as being potentially noisy* [161]. Eventually, Verbaeten [173] performs an experimental comparison of some of the above methods and proposes several variants. In particular, m classifiers from the same type are learnt using all combinations of the $K-1$ parts in the training set. Voting filters are also iterated until no more samples are removed. In [184], voting filters are obtained by generating the m classifiers using bagging: m training sets are generated by resampling and the inferred classifiers are used to classify all instances in the original training set.

3) Partition Filtering: Classification filtering is adapted for large and distributed datasets in [69], [185], which proposes a partition filter. In the first step, samples are partitioned and rules are inferred for each partition. A subset of *good* rules are chosen for each partition using two factors which measure the classification precision and coverage for the partition. In a second step, all samples are compared to the good rules of all partitions. If a sample is not covered by a set of rules, it is not classified, otherwise it is classified according to these rules. This mechanism allows distinguishing between exceptions (not covered by the rules) and mislabelled instances (covered by the rules, but misclassified). Majority or consensus vote is used to detect mislabelled instances. Privacy is preserved in distributed datasets, since each site (or partition) only shares its good rules. The approach is experimentally shown to be less aggressive than [161]. In [186], partitioning is repeated and several classifiers are learned for each partition. If all classifiers predict the same label which is different from the observed label, the instance is considered as potentially mislabelled. Votes are summed over all iterations and can be used to order the instances.

C. Model Influence and Introspection

Mislabelled instances can be detected by analysing their impact on learning. For example, Malossini et al. [53] define the leave-one-out perturbed classification (LOOPC) matrix where the (i, j) entry is the label predicted for the jth training sample if 1) the jth sample itself is removed from the training set and 2) the label of the ith sample is flipped. The LOOPC matrix is defined only for binary classification. Two algorithms are proposed to analyse the LOOPC matrix in search for wrong labels. The classification-stability algorithm (CL-stability) analyses each column to detect suspicious samples: good samples are expected to be consistently classified even in the case of small perturbation in training data. The leave-one-out-error-sensitivity (LOOE-sensitivity) algorithm detects samples whose label flip improves the overall results of the classifier. The computation of the LOOPC matrix is expensive, but it can be afforded for small datasets. Experiments show that CL-stability dominates LOOE-sensitivity. The approach is extended in [187], [188].

Based on introspection, Heskes [64] proposes an online learning algorithm for the single-unit perceptron, when labels coming from the teacher are polluted by uniform noise. The presented samples are accepted only when the confidence of the learner in the presented labelled sample is large enough. The propensity of the learner to reject suspicious labels is called the stubbornness: the learner only accepts to be taught when it does not contradict its own model too much. The stubbornness of the learner has to be tuned, since discarding too many samples may slow the learning process. An update rule is proposed for the student self-confidence: the stubbornness is increased by learner-teacher contradictions, whereas learner-teacher agreements decrease stubbornness. The update rule itself depends on the student carefulness, which reflects the confidence of the learner and can be chosen to outperform any absent-minded teacher.

D. k Nearest Neighbours-Based Methods

The k nearest neighbours (kNN) classifiers [189], [190] are sensitive to label noise [94], [95], in particular for small neighbourhood sizes [96]. Hence, it is natural that several methods have emerged in the kNN literature for cleansing training sets. Among these methods, many are presented as *editing methods* [191], what may be a bit misleading: most of these methods do not edit instances, but rather edit the training set itself by removing instances. Such approaches are also motivated by the particular computational and memory requirements of kNN methods for prediction, which linearly depend on the size of the training set. See e.g. [192] for a discussion on instance selection methods for case-based reasoning.

Wilson and Martinez [95], [193] provide a survey of kNN-based methods for data cleansing, propose several new methods and perform experimental comparisons. Wilson and Martinez [95] show that mislabelled training instances degrade the performances of both the kNN classifiers built on the full training set and the instance selection methods which are not designed to take care of label noise. This section presents solutions from the literature and is partially based on [95], [193]. See e.g. [194] for a comparison of several instance-based noise reduction methods.

kNN-based instance selection methods are mainly based on heuristics. For example, the condensed nearest neighbour (CNN) rule [195] builds a subset of training instances which allows classifying correctly all other training instances. However, such a heuristic systematically keeps mislabelled instances in the training set. There exist other heuristics which are more robust to label noise. For example, the reduced nearest neighbours (RNN) rule [196] successively removes instances whose removal do not cause other instances to be misclassified, i.e. it removes noisy and internal instances. The blame-based noise reduction (BBNR) algorithm [197] removes all instances which contribute to the misclassification of another instance and whose removal does not cause any

instance to be misclassified. In [198], [199], instances are ranked based on a score *rewarding the patterns that contribute to a correct classification and punishing those that provide a wrong one*. An important danger of instance selection is to remove too many instances [200], if not all instances in some pathological cases [95].

More complex heuristics exist in the literature; see e.g. [113], [201] for an experimental comparison for gene expression data. For example, Wilson [202] removes instances whose label is different from the majority label in its $k = 3$ nearest neighbours. This method is extended in [203] by the all-kNN method. In [95], [193], six heuristics are introduced and compared with other methods: DROP1-6. For example, DROP2 is designed to reduce label noise using the notion of instance *associates*, which have the instance itself in their k nearest neighbours. DROP2 removes an instance if its removal does not change the number of its associates which are incorrectly classified in the original training set. This algorithm tends to retain instances which are close to the classification boundary. In [200], generalised edition (GE) checks whether there are at least k' samples in the locally majority class among the k neighbours of an instance. In such a case, the instance is relabelled with the locally majority label, otherwise it is simply removed from the training set. This heuristic aims at keeping only instances with strong support for their label. Barandela and Gasca [29] show that a few repeated applications of the GE algorithm improves results in the presence of label noise.

Other instance selection methods designed to deal with label noise include e.g. IB3 which *employs a significance test to determine which instances are good classifiers and which ones are believed to be noisy* [204], [205]. Lorena et al. [206] propose to use Tomek links [207] to filter noisy instances for splice junction recognition. Different instance selection methods are compared in [114]. In [192], a set of instances are selected by using Fisher discriminant analysis, while maximising the diversity of the reduced training set. The approach is shown to be robust to label noise for a simple artificial example. In [208], different heuristics are used to distinguish three types of training instances: normal instances, border samples and instances which should be misclassified (ISM). ISM instances are such that, *based on the information in the dataset, the label assigned by the learning algorithm is the most appropriate even though it is incorrect*. For example, one of the heuristics uses a nearest neighbours approach to estimate the hardness of a training sample, i.e. how hard it is to classify correctly. ISM instances are simply removed, what results in the so-called PRISM algorithm.

E. Graph-Based Methods

Several methods in the data cleansing literature are similar to kNN-based editing methods, except that they represent training sets by *neighbourhood graphs* [209], where the instances (or nodes) are linked to other close instances. The edge between two instances can be weighted depending on the distance between them. Such methods work directly on the graphs to detect noisy instances. For example, Sánchez et al. [94] propose variants of kNN-based algorithms which use Gabriel graphs and relative neighbourhood graphs [210], [211]. In [212], [213], mode filters, which preserve edges and remove impulsive noise in images, are extended to remove label noise in datasets represented by a graph. In [209], [214], the ith instance is characterised by its *local cut edge weight statistic* J_i, which is the sum of the weights of edges linking the instance to its neighbours with a different label. Three types of instances are distinguished: *good* samples with a small J_i, *doubtful* samples with an intermediate J_i and *bad* samples with a large J_i. Two filtering policies are considered: 1) to relabel doubtful samples and to remove bad samples or 2) to relabel doubtful and bad samples using the majority class in good neighbours (if any) and to remove doubtful and bad samples which have no good neighbours.

F. Ensemble and Boosting-Based Methods

As discussed in Section III-A2, AdaBoost is well known to overfit noisy datasets. Indeed, the weights of mislabelled instances tend to become much larger than the weights of normal instances in the late iterations of AdaBoost. Several works presented below show that this propensity to overfitting can be exploited in order to remove label noise.

A simple data cleansing method is proposed in [184], which removes a given percentage of the samples with the highest weights after m iterations of AdaBoost. Experiments show that the precision of this boosting-based algorithm is not very good, what is attributed to the dynamics of Adaboost. In the first iterations, mislabelled instances quickly obtain large weights and are correctly spotted as mislabelled. However, consequently, several correctly labelled instances then obtain large weights in late iterations, what explains that they are incorrectly removed from the training set by the boosting filter.

A similar approach is pursued in [215]. Outlier removal boosting (ORBoost) is identical to AdaBoost, except that instance weights which are above a certain threshold are set to zero during boosting. Hence, data cleansing is performed while learning and not after learning as in [184]. ORBoost is sensitive to the choice of the threshold, which is performed using validation. In [216], mislabelled instances are also removed during learning, if they are misclassified by the ensemble with high confidence.

In [217], edge analysis is used to detect mislabelled instances. The edge of an instance is defined as the sum of the weights of weak classifiers which misclassified the instance [218]. Hence, an instance with a large edge is often misclassified by the weak learners and is classified by the ensemble with a low confidence, what is the contrary of the margin defined in [106]. Wheway [217] observes a homogenisation of the edge as the number of weak classifiers increases: the mean of the edge stabilises and its variance goes to zero. It means that *observations which were initially classified correctly are classified incorrectly in later rounds in order to classify harder observations correctly*, what is consistent with results in [106], [218]. Mislabelled data have *edge values which remain high due to persistent misclassification*. It is therefore proposed to remove the instances corresponding e.g. to the 5% top edge values.

G. Others Methods

There exist other methods for data cleansing. For example, in ECG segmentation, Hughes et al. [56] delete the label of the instances (and not the instances themselves) which are close to classification boundaries, since experts are known to be less reliable in that region. Thereafter, semi-supervised learning is performed using both the labelled and the (newly) unlabelled instances. In [219], a genetic algorithm approach based on a class separability criterion is proposed. In [220], [221], the automatic data enhancement (ADE) method and the automatic noise reduction (ANR) method are proposed to relabel mislabelled instances with a neural network approach. A similar approach is proposed in [222] for decision trees.

H. Discussion

One of the advantages of label noise cleansing is that removed instances have absolutely no effects on the model inference step [158]. In several works, it has been observed that simply removing mislabelled instances is more efficient than relabelling them [167], [223]. However, instance selection methods may remove too many instances [132]–[134], [200], if not all instances in some pathological cases [95]. On the one hand, Matic et al. [168] show that *overcleansing* may reduce the performances of classifiers. On the other hand, it is suggested in [46] that keeping mislabelled instances may harm more than removing too many correctly labelled samples. Therefore, a compromise has to be found. The overcleansing problem is of particular importance for imbalanced datasets [224]. Indeed, minority instances may be more likely to be removed by e.g. classification filtering (because they are also more likely to be misclassified), what makes learning even more difficult. In [225], it is shown that dataset imbalance can affect the efficiency of data cleansing methods. Label noise cleansing can also reduce the complexity of inferred models, but it is not always trivial to know if this reduction is not simply due to the reduction of the training set size [172], [173].

Surprisingly, to the best of our knowledge, the method in [56] has not been generalised to other label noise cleansing methods, what would be easy to do. Indeed, instead of completely removing suspicious instances, one could only delete their labels and perform semi-supervised learning on the resulting training set. The approach in [56] has the advantage of keeping the distribution of the instances unaltered (what is not the case for their conditional distributions, though), what is of particular interest for generative approaches. An interesting open research question is whether this method would improve the results with respect to the classical solution of simply removing suspicious instances. Another alternative would be to resubmit the suspicious samples to a human expert for relabelling as proposed in [168]. However, this may reveal too costly or even impossible in most applications, and there is no guarantee that the new labels will actually be noise-free.

VII. Label Noise-Tolerant Learning Algorithms

When some information is available about label noise or its consequences on learning, it becomes possible to design models which take label noise into account. Typically, one can learn a label noise model simultaneously with a classifier, what uncouples both components of the data generation process and improves the resulting classifier. In a nutshell, the resulting classifier learns to classify instances according to their true, unknown class. Other approaches consist in modifying the learning algorithm in order to reduce the influence of label noise. Data cleansing can also be embedded directly into the learning algorithm, like e.g. for SVMs. Such techniques are described in this section and are called label noise-tolerant, since they can tolerate label noise by modelling it. Section VII-A reviews probabilistic methods, whereas model-based methods are discussed in Section VII-B.

A. Probabilistic Methods

Many label noise-tolerant methods are probabilistic, in a broad sense. They include Bayesian and frequentist methods, as well as methods based on clustering or belief functions. An important issue which is highlighted by these methods is the identifiability of label noise. The four families of methods are discussed in the following four subsections.

1) Bayesian Approaches: Detecting mislabelled instances is a challenging problem. Indeed, there are identifiability issues [226]–[228], as illustrated in [122], where consumers answer a survey with some error probability. Under the assumption that it results in a Bernoulli process, it is possible to obtain an infinite number of maximum likelihood solutions for the true proportions of answers and the error probabilities. In other words, in this simple example, it is impossible to identify the correct model for observed data. Several works claim that prior information is strictly necessary to deal with label noise. In particular, [5], [122], [227], [228] propose to use Bayesian priors on the mislabelling probabilities to break ties. Label noise identifiability is also considered for inductive logic programming in [226], where a minimal description length principle prevents the model to overfit on label noise.

Several Bayesian methods to take care of label noise are reviewed in [68] and summarised here. In medical applications, it is often necessary to assess the quality of binary diagnosis tests with label noise. Three parameters must be estimated: the population prevalence (i.e. the true proportion of positive samples) and the sensitivity and specificity of the test itself [5]. Hence, the problem has one degree of freedom in excess, since only two data-driven constraints can be obtained (linked to the observed proportions of positive and negative samples). In [5], [229], [230], it is proposed to fix the degree of freedom using a Bayesian approach: setting a prior on the model parameters disambiguates maximum likelihood solutions. Indeed, whereas the frequentist approach considers that parameters have fixed values, the Bayesian approach considers that *all unknown parameters have a probability distribution that reflects the uncertainty in their values* and that *prior knowledge about unknown parameters can be formally included* [231]. Hence, the Bayesian approach can be seen as a generalisation of constraints on the parameters values, where the uncertainty on the parameters is taken into account through priors.

Popular choices for Bayesian priors for label noise are Beta priors [5], [128], [229], [230], [232]–[236] and Dirichlet

priors [237], [238], which are the conjugate priors of binomial and multinomial distributions, respectively. Bayesian methods have also been designed for logistic regression [130], [236], [239]–[241], hidden Markov models [84] and graphical models for medical image segmentation [242]. In the Bayesian approaches, *although the posterior distribution* of parameters *may be difficult (or impossible) to calculate directly*, efficient implementations are possible using Markov chain Monte Carlo (MCMC) methods, which allows approximating the posterior of model parameters [68]. A major advantage of using priors is the ability to include any kind of prior information in the learning process [68]. However, the priors should be chosen carefully, for *the results obtained depend on the quality of the prior distribution used* [243], [244].

In the spirit of the above Bayesian approaches, an iterative procedure is proposed in [128] to correct labels. For each sample, Rekaya et al. [235] define an indicator variable α_i which is equal to 1 if the label of the ith instance was switched. Hence, each indicator follows a Bernoulli distribution parametrised by the mislabelling rate (which itself follows a Beta prior). In [128], the probability that $\alpha_i = 1$ is estimated for each sample and the sample with the higher mislabelling probability is relabelled. The procedure is repeated as long as the test is significant. Indicators are also used in [245] for Alzheimer disease prediction, where four out of sixteen patients are detected as potentially misdiagnosed. The correction of the supposedly incorrect labels leads to a significant increase in predictive ability. A similar approach is used in [246] to robustify multiclass Gaussian process classification. If the indicator for a given sample is zero, then the label of that sample is assumed to correspond to a latent function. Otherwise, the label is assumed to be randomly chosen. The same priors as in [235] are used and the approach is shown to yield better results than other methods which assume that the latent function is polluted by a random Gaussian noise [247] or which use Gaussian processes with heavier tails [248].

2) Frequentist Methods: Since label noise is an inherently stochastic process, several frequentist methods have emerged to deal with it. A simple solution consists in using mixture models, which are popular in outlier detection [32]. In [249], each sample is assumed to be generated either from a majority (or normal) distribution or an anomalous distribution, with respective priors $1-\lambda$ and λ. The expert error probability λ is assumed to be relativity small. Depending on prior knowledge, any appropriate distribution can be used to model the majority and anomalous distributions, but the anomalous distribution may be simply chosen as uniform. The set of anomalous samples is initially empty, i.e. all samples initially belong to the majority set. Samples are successively tested and added to the anomalous set whenever the increase in log-likelihood due to this operation is higher than a predefined threshold. Mansour and Parnas [250] also consider the mixture model and propose an algorithm to learn conjunctions of literals.

Directly linked with the definition of NAR label noise in Section II-C, Lawrence and Schölkopf [67] propose another probabilistic approach to label noise. The label of an instance is assumed to correspond to two random variables (see Fig. 3, inspired by [67]): the true hidden label Y and the observed label $\tilde{Y}$, which is possibly noisy. $\tilde{Y}$ is assumed to depend only on the true label Y, whose relationship is described by a labelling matrix (see Section II-C2). Using this simple model of label noise, a Fisher discriminant is learned using an EM approach. Eventually, the approach is kernelised and is shown to effectively deal with label noise. Interestingly, the probabilistic modelling also leads to an estimation of the noise level. Later, Li et al. [251] extended this model by relaxing the Gaussian distribution assumption and carried out extensive experiments on more complex datasets, which convincingly demonstrated the value of explicit label noise modelling. More recently the same model has been extended to multiclass datasets [252] and sequential data [253]. Asymmetric label noise is also considered in [66] for logistic regression. It is shown that conditional probabilities are altered by label noise and that this problem can be solved by taking a model of label noise into account. A similar approach was developed for neural networks in [254], [255] for uniform label noise. Repeatedly, a neural network is trained to predict the conditional probability of each class, what allows optimising the mislabelling probability before retraining the neural network. The mislabelling probability is optimised either using a validation set [254] or a Bayesian approach with a uniform prior [255]. In [256], Gaussian processes for classification are also adapted for label noise by assuming that each label is potentially affected by a uniform label noise. It is shown that label noise modelling increases the likelihood of observed labels when label noise is actually present.

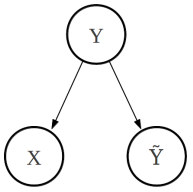


Fig. 3. Statistical model of label noise, inspired by [67].

Valizadegan and Tan [257] propose a method based on a weighted KNN. Given the probability p_i that the ith training example is mislabelled, the binary label y_i is replaced by its expected value $-p_i y_i + (1-p_i)y_i = (1-2p_i)y_i$. Then, the sum of the consistencies

$$\delta_i = (1-2p_i)y_i \frac{\sum_{j\in N(x_i)} w_{ij}(1-2p_j)y_j}{\sum_{j\in N(x_i)} w_{ij}} \tag{9}$$

between the expected value of y_i and the expected value of the weighted KNN prediction is maximised, where $N(x_i)$ contains the neighbours of x_i and w_{ij} is the weight of the jth neighbour. To *avoid declaring all the examples from one of the two classes as mislabelled*, a $L1$ regularisation is enforced on the probabilities p_i.

Contrarily to the methods described in Section VII-A1, Bayesian priors are not used in the above frequentist methods. We hypothesise that the identifiability problem discussed in Section VII-A1 is solved by using a generative approach and

setting constraints on the conditional distribution of X. For example, in [67], Gaussian distributions are used, whereas Li et al. [251] consider mixtures of Gaussian distributions. The same remark applies to Section VII-A3.

3) Clustering-Based Methods: In the generative statistical models of Section VII-A2, it is assumed that the distribution of instances can help to solve classification problems. Classes are not arbitrary: they are linked to a latent structure in the distribution of X. In other words, clusters in instances can be used to build classifiers, what is done in [136]. Firstly, a clustering of the instances [258] is performed using an unsupervised algorithm. Labels are not used and the procedure results in a mixture of K models $p_k(x)$ with priors π_k for components $k = 1 \dots K$. Secondly, instances are assumed to follow the density

$$p(x) = \sum_{y \in \mathcal{Y}} \sum_{k=1}^{K} r_{yk} \pi_k p_k(x) \tag{10}$$

where r_{yk} can be interpreted as the probability that the kth cluster belongs to the yth class. The coefficients r_{yk} are learned using a maximum likelihood approach. Eventually, classification is performed by computing the conditional probabilities $P(Y = y|X = x)$ using both the unsupervised (clusters) and supervised (r_{yk} probabilities) parts of the model. When a Gaussian mixture model is used to perform clustering, the mixture model can be interpreted as a generalisation of mixture discriminant analysis (MDA, see [259]). In this case, the model is called robust mixture discriminant analysis (RMDA) and is shown to improve classification results with respect to MDA [136], [260]. In [261], the method is adapted to discrete data for DNA barcoding and is called robust discrete discriminant analysis. In that case, data are modelled by a multivariate multinomial distribution. A clustering approach is also used in [262] to estimate a confidence on each label, where *each instance inherits the distribution of classes within its assigned cluster*. Confidences are averaged over several clusterings and a weighted training set is obtained.

In this spirit, El Gayar et al. [263] propose a method which is similar to [136]. Labels are converted into soft labels in order to reflect the uncertainty on labels. Firstly, a fuzzy clustering of the training instances is performed, which gives a set of cluster and the membership of each instance to each cluster. Then, the membership L_{yk} of the kth cluster to the yth class is estimated using the fuzzy memberships. Each instance with label y increases the membership L_{yk} by its own membership to cluster k. Eventually, the fuzzy label of each instance is computed using the class memberships of the clusters where the instance belongs. Experiments show improvements with respect to other label fuzzification methods like kNN soft labels and Keller soft labels [264].

4) Belief Functions: In the belief function theory, each possible subset of classes is characterised by a belief mass, which is the amount of evidence which supports the subset of classes [265]. For example, let us consider an expert who 1) thinks that a given case is positive, but 2) has a very low confidence in its own prediction. In the formalism of belief functions, one can translate the above judgement by a belief mass function (BMF, also called basic probability assignment) m such that $m(\{-1, +1\}) = 0.8$, $m(\{-1\}) = 0$ and $m(\{+1\}) = 0.2$. Here, there is no objective uncertainty on the class itself, but rather a subjective uncertainty on the judgement itself. For example, if a coin is flipped, the BMF would simply be $m(\{\text{head}, \text{tail}\}) = 1$, $m(\{\text{head}\}) = 0$ and $m(\{\text{tail}\}) = 0$ when the bias of the coin is unknown. If the coin is known to be unbiased, the BMF becomes $m(\{\text{head}, \text{tail}\}) = 0$, $m(\{\text{head}\}) = \frac{1}{2}$ and $m(\{\text{tail}\}) = \frac{1}{2}$. Again, this simple example illustrates how the belief function theory allows distinguishing subjective uncertainty from objective uncertainty. Notice that Smets [266] argues that it is necessary to fall back to classical probabilities in order to make decisions. Different decision rules are analysed in [79]. Interestingly, the belief function formalism can be used to modify standard machine learning methods like e.g. kNN classifiers [78], neural networks [80], decision trees [267], mixture models [268], [269] or boosting [270].

In the context of this survey, belief functions cannot be used directly, since the belief masses are not available. Indeed, they are typically provided by the expert itself as an attempt to quantify its own (lack of) confidence, but we made the hypothesis in Section I that such information is not available. However, several works have proposed heuristics to infer belief masses directly from data [78], [80], [271].

In [78], a kNN approach based on Dempster-Shafer theory is proposed. If a new sample x_s has to be classified, each training sample (x_i, y_i) is considered as an evidence that the class of x_s is y_i. The evidence is represented by a BMF $m_{s,i}$ such that $m_{s,i}(\{y_i\}) = \alpha$, $m_{s,i}(\mathcal{Y}) = 1 - \alpha$ and $m_{s,i}$ is zero for all other subsets of classes, where

$$\alpha = \alpha_0 \Phi(d_{s,i}) \tag{11}$$

such that $0 < \alpha_0 < 1$ and Φ is a monotonically decreasing function of the distance $d_{s,i}$ between both instances. There are many possible choices for Φ;

$$\Phi(d) = \exp -\gamma d^{\beta} \tag{12}$$

is chosen in [78], where $\gamma > 0$ and $\beta \in \{1, 2\}$. Heuristics are proposed to select proper values of α_0 and γ. For the classification of the new sample x_s, each training sample provides an evidence. These evidences are combined using the Dempster rule and it becomes possible to take a decision (or to refuse to take a decision if the uncertainty is too high). The case of mislabelling is experimentally studied in [78], [272] and the approach is extended to neural networks in [80].

In [271], a kNN approach is also used to infer BMFs. For a given training sample, the frequency of each class in its k nearest neighbours is computed. Then, the sample is assigned to a subset of classes containing 1) the class with the maximum frequency and 2) the classes whose frequency is not too different from the maximum frequency. A neural network is used to compute beliefs for test samples.

B. Model-Based Methods

Apart from probabilistic methods, specific strategies have been developed to obtain label noise-tolerant variants of

popular learning algorithms, including e.g. support vector machines, neural networks, decision trees, boosting and semi-supervised algorithms. The five families of methods are discussed in the following five subsections.

1) Support Vector Machines and Robust Losses: SVMs are not robust to label noise [62], [82], even if instances are allowed to be misclassified during learning. Indeed, instances which are misclassified during learning are penalised in the objective using the hinge loss

$$[1 - y_i \langle x_i, w \rangle]_+ \tag{13}$$

where $[z]_+ = \max(0, z)$ and w is the weight vector. The hinge loss increases linearly with the distance to the classification boundary and is therefore significantly affected by mislabelled instances which stand far from the boundary.

Data cleansing can be directly implemented into the learning algorithm of SVMs. For example, instances which correspond to very large dual weights can be identified as potentially mislabelled [273]. In [274], k samples are allowed to be not taken into account in the objective function. For each sample, a binary variable (indicating whether or not to consider the sample) is added and the sum of the indicators is constrained to be equal to k. An opposite approach is proposed in [275] for aggregated training sets, which consists of several distinct training subsets labelled by different experts. The percentage of support vectors in training samples is constrained to be identical in each subset, in order to decrease the influence of low-quality teachers which tend to require more support vectors due to more frequent mislabelling. In [276], [277], SVMs are adapted by weighting the contribution of each training sample in the objective function. The weights (or *fuzzy memberships*) are computed using heuristics. Similar work is done in [278] for relevance vector machines (RVMs). Empathetic constraints SVMs [279] relax the constraints of suspicious samples in the SVM optimisation problem.

Xu et al. [280] propose a different approach, which consists in using the loss

$$\eta_i \left[1 - y_i \langle x_i, w \rangle\right]_+ + (1 - \eta_i) \tag{14}$$

where $0 \leq \eta_i \leq 1$ indicates whether the ith sample is an outlier. The η_i variables must be optimised together with the weights vector, what is shown to be equivalent to using the robust hinge loss

$$\min(1, [1 - y_i \langle x_i, w \rangle]_+). \tag{15}$$

Notice that there exist other bounded, non-convex losses [281]–[284] which could be used similarly. A non-convex loss is also used in [285] to produce label noise-tolerant SVMs without filtering. For binary classification with $y \in \{-1, +1\}$, the loss is

$$K_{p_e} \left[(1 - p_e(-y_i)) \left[1 - y_i \langle x_i, w \rangle\right]_+ - p_e(y_i) \left[1 + y_i \langle x_i, w \rangle\right]_+\right] \tag{16}$$

where $K_{p_e} = \frac{1}{1 - p_e(+1) - p_e(-1)}$. Interestingly, the expected value of the proposed loss (with respect to all possible mislabellings of the noise-free training set) is equal to the hinge loss computed on the noise-free training set. In other words, it is possible to *estimate the noise-free* [...] *errors from the noisy data*. Theoretical guarantees are given and the proposed approach is shown to outperform SVMs, but error probabilities must be known *a priori*.

2) Neural Networks: Different label noise-tolerant variants of the perceptron algorithm are reviewed and compared experimentally in [286]. In the standard version of this algorithm, samples are presented repeatedly (on-line) to the classifier. If a sample is misclassified, i.e.

$$y_i \left[w x_i + b\right] < 0 \tag{17}$$

where w is the weight vector and b is the bias, then the weight vector is adjusted towards this sample. Eventually, the perceptron algorithm converges to a solution.

Since the solution of the perceptron algorithm can be biased by mislabelled samples, different variants have been designed to reduce the impact of mislabelling. With the λ-trick [287], [288], if an instance has already been misclassified, the adaptation criterion becomes $y_i \left[w x_i + b\right] + \lambda \|x_i\|_2^2 < 0$. Large values of λ may prevent mislabelled instances to trigger updates. Another heuristic is the α-bound [289], which does not update w for samples which have already been misclassified α times. This simple solution limits the impact of mislabelled instances. Although not directly designed to deal with mislabelling, Khardon and Wachman [286] also describe the perceptron algorithm using margins (PAM, see [290]). PAM updates w for instances with $y_i \left[w x_i + b\right] < \tau$, similarly to support vector classifiers and to the λ-trick.

3) Decision Trees: Decision trees can easily overfit data, if they are not pruned. In fact, learning decision trees *involves a trade-off between accuracy and simplicity*, which are two requirements for good decision trees in real-world situations [291]. It is particularly important to balance this trade-off in the presence of label noise, what makes the overfitting problem worse. For example, Clark and Niblett [291] propose the CN2 algorithm which learns a disjunction of logic rules while avoiding too complex ones.

4) Boosting Methods: In boosting, an ensemble of weak learners h_t with weights α_t is formed iteratively using a weighted training set. At each step t, the weights $w_i^{(t)}$ of misclassified instances are increased (resp. decreased for correctly classified samples), what progressively reduces the ensemble training error because the next weak learners focus on the errors of the previous ones. As discussed in Section III, boosting methods tend to overfit label noise. In particular, AdaBoost obtains large weights for mislabelled instances in late stages of learning. Hence, several methods propose to update weights more carefully to reduce the sensitivity of boosting to label noise. In [292], MadaBoost imposes an upper bound for each instance weight, which is simply equal to the initial value of that weight. The AveBoost and AveBoost2 [293], [294] algorithms replace the weight $w_i^{(t+1)}$ of the ith instance at step $t+1$ by

$$\frac{t w_i^{(t)} + w_i^{(t+1)}}{t+1}. \tag{18}$$

With respect to AdaBoost, AveBoost2 obtains larger training errors, but smaller generalisation errors. In other words, AveBoost2 is less prone to overfitting than AdaBoost, what

improves results in the presence of label noise. Kim [295] proposes another ensemble method called Averaged Boosting (A-Boost), which 1) does not take instances weights into account to compute the weights of the successive weak classifiers and 2) performs similarly to bagging on noisy data. Other weighting procedures have been proposed in e.g. [296], but they were not assessed in the presence of label noise.

In [297], two approaches are proposed to reduce the consequences of label noise in boosting. Firstly, AdaBoost can be early-stopped: limiting the number of iterations prevents AdaBoost from overfitting. A second approach consists in *smoothing* the results of AdaBoost. The proposed BB algorithm combines bagging and boosting: 1) K training sets consisting of ρ percents of the training set (sub-sampled with replacement) are created, 2) K boosted classifiers are trained for M iterations and 3) the predictions are aggregated. In [297], it is advised to use $K = 15$, $M = 15$ and $\rho = \frac{1}{2}$. The BB algorithm is shown to be less sensitive to label noise than AdaBoost. A similar approach is proposed in [298]: the multiple boosting (MB) algorithm.

A *reverse boosting* algorithm is proposed in [299]. In adaptive boosting, weak learners may have difficulties to obtain good separation frontiers because correctly classified samples get lower and lower weights as learning goes on. Hence, safe, noisy and borderline patterns are distinguished, whose weights are respectively increased, decreased and unaltered during boosting. Samples are classified into these three categories using parallel perceptrons, a specific type of committee machine whose margin allows to separate the input space into three regions: a safe region (beyond the margin), a noisy region (before the margin) and a borderline region (inside the margin). The approach improves the results of parallel perceptrons in the presence of label noise, but is most often dominated by classical perceptrons.

5) Semi-Supervised Learning: In [7], a particle competition-based algorithm is proposed to perform semi-supervised learning in the presence of label noise. Firstly, the dataset is converted into a graph, where instances are nodes with edges between similar instances. Each labelled node is associated with a labelled particle. Particles walk through the graph and cooperate with identically-labelled particles to label unlabelled instances, while staying in the neighbourhood of their home node. What interests us in [7] is the behaviour of mislabelled particles: they are pushed away by the particles of near instances with different labels, what prevents a mislabelled instance to influence the label of close unlabelled instances. In [300], unlabelled instances are firstly labelled using a semi-supervised learning algorithm, then the new labels are used to filter instances. Other works on label noise for semi-supervised learning include e.g. [301] or [302]–[304], which are particular because they model the label noise induced by the labelling of unlabelled samples.

C. Discussion

The probabilistic methods to deal with label noise are grounded in a more theoretical approach than robust or data cleansing methods. Hence, probabilistic models of label noise can be directly used and allow to take advantage of prior knowledge. Moreover, the model-based label noise-tolerant methods allow us to use the knowledge gained by the analysis of the consequences of label noise. However, the main problem of the approaches described in this section is that they increase the complexity of learning algorithms and can lead to overfitting, because of the additional parameters of the training data model. Moreover, the identifiability issue discussed in Section VII-A1 must be addressed, what is done explicitly in the Bayesian approach (using Bayesian priors) and implicitly in the frequentist approach (using generative models).

As highlighted in [1], different models should be used for training and testing in the presence of label noise. Indeed, a complete model of the training data consists of a label noise model and a classification model. Both parts are used during training, but only the classification model is useful for prediction: one has no interest in making noisy predictions. Dropping the label noise model is only possible when label noise is explicitly modelled, as in the probabilistic approaches discussed in Section VII-A. For other approaches, the learning process of the classification model is supposed to be robust or tolerant to label noise and to produce a good classification model.

VIII. Experiments in the Presence of Label Noise

This section discusses how experiments are performed in the label noise literature. In particular, existing datasets, label noise generation techniques and quality measures are highlighted.

A. Datasets with Identified Mislabelled Instances and Label Noise Generation Techniques

There exist only a few datasets where incorrect labels have been identified. Among them, Lewis et al. [305] provide a version of the Reuters dataset with corrected labels and Malossini et al. [53] propose a short analysis of the reliability of instances for two microarray datasets. In spam filtering, where the expert error rate is usually between 3% and 7%, the TREC datasets have been carefully labelled by experts adhering to the same definition of *spam*, with a resulting expert error rate of about 0.5% [127]. Mislabelling is also discussed for a medical image processing application in [306] and Alzheimer disease prediction in [245]. However, artificial label noise is more common in the literature. Most studies on label noise use NCAR label noise, which is introduced in real datasets by 1) randomly selecting instances and 2) changing their label into one of the other remaining labels [135]. In this case, label noise is independent of Y. In [307], it is also proposed to simulate label noise for artificial datasets by 1) computing the membership probabilities $P(Y = y|X = x)$ for each training sample x, 2) adding a small uniform noise to these values and 3) choosing the label corresponding to the largest polluted membership probability.

Several methods have been proposed to introduce NAR label noise. For example, in [62], label noise is artificially introduced by changing the labels of some randomly chosen instances from the majority class. In [3], [69], label noise is

introduced using a pairwise scheme. Two classes c_1 and c_2 are selected, then each instance of class c_1 has a probability P_e to be incorrectly labelled as c_2 and vice versa. In other words, this label noise models situations where only certain types of classes are mislabelled. In [1], label noise is introduced by increasing the entropy of the conditional mass function $P(\tilde{Y}|X)$. The proposed procedure is called majorisation: it leaves the probability of the majority class unchanged, but the remaining probability is spread more evenly on the other classes, with respect to the true conditional mass function $P(Y|X)$. In [151], [153], the percentage of mislabelled instances is firstly chosen, then the proportions of mislabelled instances in each class are fixed.

NNAR label noise is considered in much less works than NCAR and NAR label noise. For example, Chhikara and McKeon [72] introduce the truncated and the exponential label noise models which are detailed in Section III-A3 and where the probability of mislabelling depends on the distance to the classification boundary. A special case of truncated label noise is studied in [70]. In [81], two features are randomly picked and the probability of mislabelling depends on which quadrant (with respect to the two selected features) the sample belongs to.

In practice, it would be very interesting to obtain more real-world datasets where mislabelled instances are clearly identified. Also, an important open research problem is to find what the characteristics of real-world label noise are. Indeed, it is not yet clear in the literature if and when NCAR, NAR or NNAR label noise is the most realistic.

B. Validation and Test of Algorithms in the Presence of Label Noise

An important issue for methods which deal with label noise is to prove their efficiency. Depending on the consequence of label noise which is targeted, different criteria can be used. In general, a good method must either 1) maintain the value of the quality criterion when label noise is introduced or 2) improve the value of the criterion with respect to other methods in the presence of label noise. In the literature, most experiments assess the efficiency of methods to take care of label noise in terms of accuracy (see e.g. [46], [69], [138], [160], [161], [171], [180], [184]), since a decrease in accuracy is one of the main consequences of label noise, as discussed in Section III-A.

Another common criterion is the model complexity [46], [138], [184], e.g. the number of nodes for decision trees or the number of rules in inductive logic. Indeed, as discussed in Section III-B, some inference algorithms tend to overfit in the presence of label noise, what results in overly complex models. Less complex models are considered better, since they are less prone to overfitting.

In some contexts, the estimated parameters of the models themselves can also be important, as discussed in Section III-C. Several works focus on the estimation of true frequencies from observed frequencies [4], [122], [123], [126], what is important e.g. in disease prevalence estimation.

Eventually, in the case of data cleansing methods, one can also investigate the filter precision. In other words, do the removed instances actually correspond to mislabelled instances and conversely ? Different measures are used in the literature, which can be explained using Fig. 4 inspired by [46], [138]. In [46], [69], [180], [184], [308], two types of errors are distinguished. Type 1 errors are correctly labelled instances which are erroneously removed. The corresponding measure is

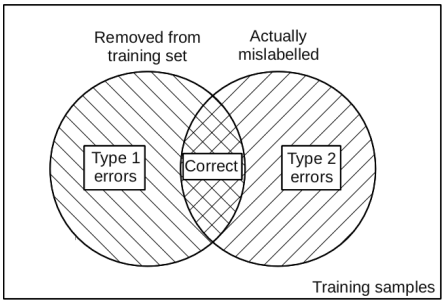


Fig. 4. Types of errors in data cleansing for label noise, inspired by [46], [138].

$$ER_1 = \frac{\text{\# of correctly labelled instances which are removed}}{\text{\# of correctly labelled instances}}. \tag{19}$$

Type 2 errors are mislabelled instances which are not removed. The corresponding measure is

$$ER_2 = \frac{\text{\# of mislabelled instances which are not removed}}{\text{\# of mislabelled instances}}. \tag{20}$$

The percentage of removed samples which are actually mislabelled is also computed in [46], [69], [180], [183], [184], [308], what is given by the noise elimination precision

$$NEP = \frac{\text{\# of mislabelled instances which are removed}}{\text{\# of removed instances}}. \tag{21}$$

Also, Verbaeten and Van Assche [184] compute the percentage of mislabelled instances in the cleansed training set.

Notice that a problem which is seldom mentioned in the literature is that model validation can be difficult in the presence of label noise. Indeed, since validation data are also polluted by label noise, methods like e.g. cross-validation or bootstrap may poorly estimate generalisation errors and choose meta-parameters which are not optimal (with respect to clean data). For example, the choice of the regularisation constant in regularised logistic regression will probably be affected by the presence of mislabelled instances far from the classification boundary. We think that this is an important open research question.

IX. Conclusion

This survey shows that label noise is a complex phenomenon with many potential consequences. Moreover, there exist many different techniques to address label noise, which

can be classified as label noise-robust methods, label noise cleansing methods or label noise-tolerant methods. As discussed in Section VII-A1, an identification problem occurs in practical inference: mislabelled instances are difficult to distinguish from correctly labelled instances. In fact, *without additional information beyond the main data, it is not possible to take into account the effect of* mislabelling [84]. A solution is to make assumptions which allow selecting a compromise between naively using instances as they are and seeing any instance as possibly mislabelled.

All methods described in this survey can be interpreted as making particular assumptions. Firstly, in label noise-robust methods described in Section V, overfitting avoidance is assumed to be sufficient to deal with label noise. In other words, mislabelled instances are assumed to cause overfitting in the same way as any other instance would. Secondly, in data cleansing methods presented in Section VI, different heuristics are used to distinguish mislabelled instances from exceptions. Each heuristic is in fact a definition of *what* is label noise. Thirdly, label noise-tolerant methods described in Section VII impose different constraint using e.g. Bayesian priors or structural constraints (i.e. in generative methods) or attempt to make existing methods less sensitive to the consequences of label noise.

In conclusion, the machine learning practitioner has to choose the method whose definition of label noise seems more relevant in his particular field of application. For example, if experts can provide prior knowledge about the values of the parameters or the shape of the conditional distributions, probabilistic methods should be used. On the other hand, if label noise is only marginal, label noise-robust methods could be sufficient. Eventually, most data cleansing methods are easy to implement and have been shown to be efficient and to be good candidates in many situations. Moreover, underlying heuristics are usually intuitive and easy-to-interpret, even for the non-specialist who can look at removed instances.

Open research questions related to label noise include whether the semi-supervised approach in [56] can be generalised to other label noise cleansing methods (see the discussion in Section VI-H). Also, it would be very interesting to obtain more real-world datasets where mislabelled instances are clearly identified. It is also important to find what the characteristics of real-world label noise are, since it is not yet clear if and when NCAR, NAR or NNAR label noise is the most realistic. The problem of meta-parameter selection in the presence of label noise is also an important open research problem, since estimated error rates are also biased by label noise [112], [126], [127].

REFERENCES

[1] R. J. Hickey, "Noise modelling and evaluating learning from examples," *Artif. Intell.*, vol. 82, no. 1-2, pp. 157–179, 1996.
[2] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[3] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artif. Intell. Rev.*, vol. 22, pp. 177–210, 2004.
[4] I. Bross, "Misclassification in 2 x 2 tables," *Biometrics*, vol. 10, no. 4, pp. 478–486, 1954.
[5] L. Joseph, T. W. Gyorkos, and L. Coupal, "Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard," *Am. J. Epidemiol.*, vol. 141, no. 3, pp. 263–272, 1995.
[6] A. Hadgu, "The discrepancy in discrepant analysis," *The Lancet*, vol. 348, no. 9027, pp. 592–593, 1996.
[7] F. A. Breve, L. Zhao, and M. G. Quiles, "Semi-supervised learning from imperfect data through particle cooperation and competition," in *Proc. Int. Joint Conf. Neural Networks*, Barcelona, Spain, Jul. 2010, pp. 1–8.
[8] X. Wu, *Knowledge acquisition from databases*. Greenwich, CT: Ablex Publishing Corp., 1996.
[9] J. Sàez, M. Galar, J. Luengo, and F. Herrera, "Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition," *Knowl. Inf. Syst.*, pp. 1–28, in press.
[10] J. L. Schafer and J. W. Graham, "Missing data: our view of the state of the art," *Psychol. methods*, vol. 7, no. 2, pp. 147–177, 2002.
[11] D. Angluin and P. Laird, "Learning from noisy examples," *Mach. Learn.*, vol. 2, pp. 343–370, 1988.
[12] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy, "Class noise and supervised learning in medical domains: The effect of feature extraction," in *Proc. 19th IEEE Int. Symp. Computer-Based Medical Systems*, Washington, DC, Jun. 2006, pp. 708–713.
[13] R. Hanner, S. Becker, N. V. Ivanova, and D. Steinke, "Fish-bol and seafood identification: geographically dispersed case studies reveal systemic market substitution across canada," *Mitochondr. DNA*, vol. 22, pp. 106–122, 2011.
[14] E. Garcia-Vazquez, G. Machado-Schiaffino, D. Campo, and F. Juanes, "Species misidentification in mixed hake fisheries may lead to overexploitation and population bottlenecks," *Fish. Res.*, vol. 114, pp. 52 – 55, 2012.
[15] C. Lopez-Vizcón and F. Ortega, "Detection of mislabelling in the fresh potato retail market employing microsatellite markers," *Food Control*, vol. 26, no. 2, pp. 575 – 579, 2012.
[16] D.-M. Cawthorn, H. A. Steinman, and L. C. Hoffman, "A high incidence of species substitution and mislabelling detected in meat products sold in south africa," *Food Control*, vol. 32, no. 2, pp. 440 – 449, 2013.
[17] L. G. Valiant, "Learning disjunction of conjunctions," in *Proc. 9th Int. Joint Conf. Artificial Intelligence - Vol. 1*, Los Angeles, CA, Aug. 1985, pp. 560–566.
[18] M. Kearns and M. Li, "Learning in the presence of malicious errors," in *Proc. 20th Ann. ACM Symp. Theory of computing*, Chicago, IL, May 1988, pp. 267–280.
[19] S. E. Decatur, "Statistical queries and faulty pac oracles," in *Proc. 6th Ann. Conf. Computational Learning Theory*, Santa Cruz, CA, Jul. 1993, pp. 262–268.
[20] ——, "Learning in hybrid noise environments using statistical queries," in *Learning from Data: AI and Statistics V*, D. Fisher and H.-J. Lenz, Eds. Berlin: Springer Verlag, 1995, pp. 175–185.
[21] R. H. Sloan, "Four types of noise in data for pac learning," *Inform. Process. Lett.*, vol. 54, no. 3, pp. 157–162, 1995.
[22] P. Auer and N. Cesa-Bianchi, "On-line learning with malicious noise and the closure algorithm," *Ann. Math. Artif. Intel.*, vol. 23, no. 1-2, pp. 83–99, 1998.
[23] N. Cesa-Bianchi, E. Dichterman, P. Fischer, E. Shamir, and H. U. Simon, "Sample-efficient strategies for learning in the presence of noise," *J. ACM*, vol. 46, no. 5, pp. 684–719, 1999.
[24] R. A. Servedio, "Smooth boosting and learning with malicious noise," *J. Mach. Learn. Res.*, vol. 4, pp. 633–648, 2003.
[25] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Proc. 3rd Asian Conf. Machine Learning*, Taoyuan, Taiwan, Nov. 2011, pp. 97–112.
[26] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines," in *20th Eur. Conf. Artificial Intelligence*, Montpellier, France, Aug. 2012, pp. 870–875.
[27] B. Edmonds, "The nature of noise," in *Epistemological Aspects of Computer Simulation in the Social Sciences*, F. Squazzoni, Ed. Berlin: Springer, 2009, pp. 169–182.
[28] A. v. d. Hout and P. G. M. v. d. Heijden, "Randomized response, statistical disclosure control and misclassification: A review," *Int. Stat. Rev.*, vol. 70, no. 2, pp. 269–288, 2002.
[29] R. Barandela and E. Gasca, "Decontamination of training samples for supervised pattern recognition methods," in *Proc. Joint IAPR Int. Workshops Advances in Pattern Recognition*, Alicante, Spain, Aug.–Sep. 2000, pp. 621–630.

[30] D. M. Hawkins, *Identification of outliers*. London, UK: Chapman and Hall, 1980.
[31] R. J. Beckman and R. D. Cook, "Outlier.....s," *Technometrics*, vol. 25, no. 2, pp. 119–149, 1983.
[32] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York, NY: Wiley, 1994.
[33] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.
[34] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems 12*, Denver, CO, Aug.–Sep. 1999, pp. 582–588.
[35] P. Hayton, B. Schölkopf, L. Tarassenko, and P. Anuzis, "Support vector novelty detection applied to jet engine vibration spectra," in *Advances in Neural Information Processing Systems 13*, Denver, CO, Nov. 2000, pp. 946–952.
[36] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[37] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recogn.*, vol. 40, no. 3, pp. 863–874, 2007.
[38] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput.Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.
[39] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, "Enhancing data analysis with noise removal," *IEEE Trans. Knowl. Data Eng*, vol. 18, pp. 304–319, Mar. 2006.
[40] H. Lukashevich, S. Nowak, and P. Dunker, "Using one-class svm outliers detection for verification of collaboratively tagged image training sets," in *Proc. 2009 IEEE Int. Conf. Multimedia and Expo*, Piscataway, NJ, Jun.–Jul. 2009, pp. 682–685.
[41] D. Collett and T. Lewis, "The subjective nature of outlier rejection procedures," *J. Roy. Stat. Soc. C. - App.*, vol. 25, no. 3, pp. 228–237, 1976.
[42] X. Liu, G. Cheng, and J. X. Wu, "Analyzing outliers cautiously," *IEEE Trans. Knowl. Data Eng*, vol. 14, pp. 432–437, Mar.–Apr. 2002.
[43] D. McNicol, *A primer of signal detection theory*. London, UK: Allen & Unwin, 1972, ch. What are statistical decisions, pp. 1–17.
[44] P. Smets, "Imperfect information: Imprecision and uncertainty," in *Uncertainty Management in Information Systems*, A. Motro and P. Smets, Eds. Berlin: Springer Verlag, 1997, pp. 225–254.
[45] B. de Finetti, *Philosophical lectures on probability: Collected, Edited, and Annotated by Alberto Mura*. Berlin: Springer, 2008.
[46] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, pp. 131–167, 1999.
[47] P. B. Brazdil and P. Clark, "Learning from imperfect data," in *Machine Learning, Meta-Reasoning and Logics*, P. B. Brazdil and K. Konolige, Eds. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1990, pp. 207–232.
[48] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *J. Roy. Stat. Soc. C. - App.*, vol. 28, no. 1, pp. 20–28, 1979.
[49] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast - but is it good?: evaluating non-expert annotations for natural language tasks," in *Proc. Conf. Empirical Methods in Natural Language Processing*, Waikiki, Honolulu, Hawaii, Oct. 2008, pp. 254–263.
[50] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proc. ACM SIGKDD Workshop Human Computation*, Washington, DC, Jul. 2010, pp. 64–67.
[51] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, 2010.
[52] M.-C. Yuen, I. King, and K.-S. Leung, "A survey of crowdsourcing systems," in *Proc. IEEE 3rd Int. Conf. Social Computing*, Boston, MA, Oct. 2011, pp. 766 –773.
[53] A. Malossini, E. Blanzieri, and R. T. Ng, "Detecting potential labeling errors in microarrays by data perturbation," *Bioinformatics*, vol. 22, no. 17, pp. 2114–2121, 2006.
[54] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi, "Inferring ground truth from subjective labelling of venus images," in *Advances in Neural Information Processing Systems 7*, Denver, CO, Nov.–Dec. 1994, pp. 1085–1092.
[55] P. Smyth, "Bounds on the mean classification error rate of multiple experts," *Pattern Recog. Lett.*, vol. 17, no. 12, pp. 1253–1257, 1996.
[56] N. P. Hughes, S. J. Roberts, and L. Tarassenko, "Semi-supervised learning of probabilistic models for ecg segmentation," in *Ann. Int. Conf. IEEE Engineering in Medicine and Biology Society*, San Francisco, CA, Sep. 2004, pp. 434–437.
[57] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: the penn treebank," *Comput. Linguist.*, vol. 19, no. 2, pp. 313–330, 1993.
[58] D. Sculley and G. V. Cormack, "Filtering email spam in the presence of noisy user feedback," in *Proc. 5th Conf. Email and Anti-spam*, Mountain View, CA, Aug. 2008.
[59] K. Orr, "Data quality and systems theory," *Commun. ACM*, vol. 41, no. 2, pp. 66–71, 1998.
[60] T. Redman, "The impact of poor data quality on the typical enterprise," *Commun. ACM*, vol. 2, no. 2, pp. 79–82, 1998.
[61] J. I. Maletic and A. Marcus, "Data cleansing: Beyond integrity analysis," in *Proc. Conf. Information Quality*, Cambridge, MA, Oct. 2000, pp. 200–209.
[62] D. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 275–306, 2010.
[63] A. T. Kalai and R. A. Servedio, "Boosting in the presence of noise," *J. Comput. Syst. Sci.*, vol. 71, no. 3, pp. 266–290, 2005.
[64] T. Heskes, "The use of being stubborn and introspective," in *Proc. ZiF Conf. Adaptative Behavior and Learning*, Bielefeld, Germany, Apr. 1994, pp. 55–65.
[65] J. A. Aslam, "On the sample complexity of noise-tolerant learning," *Inform. Process. Lett.*, vol. 57, no. 4, pp. 189–195, 1996.
[66] M. Rantalainen and C. C. Holmes, "Accounting for control mislabeling in case-control biomarker studies," *J. Proteome Res.*, vol. 10, no. 12, pp. 5562–5567, 2011.
[67] N. D. Lawrence and B. Schölkopf, "Estimating a kernel fisher discriminant in the presence of label noise," in *Proc. of the 18th Int. Conf. Machine Learning*, Williamstown, MA, Jun.–Jul. 2001, pp. 306–313.
[68] C. J. Perez, F. J. Giron, J. Martin, M. Ruiz, and C. Rojano, "Misclassified multinomial data: a bayesian approach," *Rev. R. Acad. Cien. Serie A. Mat.*, vol. 101, no. 1, pp. 71–80, 2007.
[69] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *Proc. 20th Int. Conf. Machine Learning*, Washington, DC, Aug. 2003, pp. 920–927.
[70] P. A. Lachenbruch, "Discriminant analysis when the initial samples are misclassified ii: Non-random misclassification models," *Technometrics*, vol. 16, no. 3, pp. 419–424, 1974.
[71] ——, "Discriminant analysis when the initial samples are misclassified," *Technometrics*, vol. 8, no. 4, pp. 657–662, 1966.
[72] R. S. Chhikara and J. McKeon, "Linear discriminant analysis with misallocation in training samples," *J. Am. Stat. Assoc.*, vol. 79, no. 388, pp. 899–906, 1984.
[73] E. Cohen, "Learning noisy perceptrons by a perceptron in polynomial time," in *Proc. 38th Ann. Symp. Foundations of Computer Science*, Oct. 1997, pp. 514–523.
[74] E. Beigman and B. B. Klebanov, "Learning with annotation noise," in *Proc. Joint Conf. 47th Ann. Meeting ACL and 4th Int. Joint Conf. Natural Language Processing AFNLP: Vol. 1*, Suntec, Singapore, Aug. 2009, pp. 280–287.
[75] B. Beigman Klebanov and E. Beigman, "From annotator agreement to noise models," *Comput. Linguist.*, vol. 35, no. 4, pp. 495–503, 2009.
[76] A. Kolcz and G. V. Cormack, "Genre-based decomposition of email class noise," in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Paris, France, Jun.–Jul. 2009, pp. 427–436.
[77] B. B. Klebanov and E. Beigman, "Some empirical evidence for annotation noise in a benchmarked dataset," in *Human Language Technologies: 2010 Ann. Conf. North American Chapter ACL*, Los Angeles, CA, Jun. 2010, pp. 438–446.
[78] T. Denœux, "A k-nearest neighbor classification rule based on dempster-shafer theory," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 804–813, May 1995.
[79] ——, "Analysis of evidence-theoretic decision rules for pattern classification," *Pattern Recogn.*, vol. 30, no. 7, pp. 1095–1107, 1997.
[80] ——, "A neural network classifier based on dempster-shafer theory," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, pp. 131–150, Mar. 2000.
[81] P. S. Sastry, G. D. Nagendra, and N. Manwani, "A team of continuous-action learning automata for noise-tolerant learning of half-spaces," *IEEE Trans. on Syst., Man, Cybern. B, Cybern.*, vol. 40, pp. 19–28, Feb. 2010.
[82] N. Manwani and P. S. Sastry, "Noise tolerance under risk minimization," *IEEE Trans. on Syst., Man, Cybern.*, in press.
[83] A. Sarma and D. D. Palmer, "Context-based speech recognition error detection and correction," in *Proc. Human Language Technology Conf.*

/ North American chapter of the AACL Ann. Meeting, Boston, MA, May 2004, pp. 85–88.

[84] M. J. García-Zattera, T. Mutsvari, A. Jara, D. Declerckc, and E. Lesaffrea, "Correcting for misclassification for a monotone disease process with an application in dental research," *Stat. Med.*, vol. 29, no. 30, pp. 3103–3117, 2010.

[85] L. Breiman, "Randomizing outputs to increase prediction accuracy," *Mach. Learn.*, vol. 40, no. 3, pp. 229–242, 2000.

[86] G. Martínez-Muñoz and A. Suárez, "Switching class labels to generate classification ensembles," *Pattern Recogn.*, vol. 38, no. 10, pp. 1483–1494, 2005.

[87] G. Martínez-Muñoz, A. Sánchez-Martínez, D. Hernández-Lobato, and A. Suárez, "Building ensembles of neural networks with class-switching," in *Proc. 16th Int. Conf. Artificial Neural Networks - Vol. I*, Athens, Greece, Sep. 2006, pp. 178–187.

[88] ——, "Class-switching neural network ensembles," *Neurocomputing*, vol. 71, no. 13-15, pp. 2521–2528, 2008.

[89] D. P. Williams, "Label alteration to improve underwater mine classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, pp. 488–492, May 2011.

[90] G. J. McLachlan, "Asymptotic results for discriminant analysis when the initial samples are misclassified," *Technometrics*, vol. 14, no. 2, pp. 415–422, 1972.

[91] P. A. Lachenbruch, "Note on initial misclassification effects on the quadratic discriminant function," *Technometrics*, vol. 21, no. 1, pp. 129–132, 1979.

[92] J. E. Michalek and R. C. Tripathi, "The effect of errors in diagnosis and measurement on the estimation of the probability of an event," *J. Am. Stat. Assoc.*, vol. 75, no. 371, pp. 713–721, 1980.

[93] Y. Bi and D. R. Jeske, "The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise," *J. Multivariate Anal.*, vol. 101, no. 7, pp. 1622–1637, 2010.

[94] J. Sánchez, F. Pla, and F. Ferri, "Prototype selection for the nearest neighbour rule through proximity graphs," *Pattern Recog. Lett.*, vol. 18, no. 6, pp. 507–513, 1997.

[95] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Mach. Learn.*, vol. 38, no. 3, pp. 257–286, 2000.

[96] S. Okamoto and Y. Nobuhiro, "An average-case analysis of the k-nearest neighbor classifier for noisy domains," in *Proc. 15th Int. Joint Conf. Artifical intelligence - Vol. 1*, Nagoya, Aichi, Japan, Aug. 1997, pp. 238–243.

[97] J. Zhang and Y. Yang, "Robustness of regularized linear classification methods in text categorization," in *Proc. 26th Ann. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Toronto, Canada, Jul.–Aug. 2003, pp. 190–197.

[98] Y. Freund and R. Schapire, "A short introduction to boosting," *J. Jpn. Soc. Artif. Intell*, vol. 14, no. 5, pp. 771–780, 1999.

[99] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, 1999.

[100] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.

[101] R. A. McDonald, D. J. Hand, and I. A. Eckley, "An empirical comparison of three boosting algorithms on real data sets with artificial class noise," in *Proc. 4th Int. Workshop Multiple Classifier Systems*, Guilford, UK, Jun. 2003, pp. 35–44.

[102] P. Melville, N. Shah, L. Mihalkova, and R. J. Mooney, "Experiments on ensembles with missing and noisy data," in *Proc. 5th Int. Workshop Multi Classifier Systems*, Cagliari, Italy, Jun. 2004, pp. 293–302.

[103] W. Jiang, "Some theoretical aspects of boosting in the presence of noisy data," in *Proc. 18th Int. Conf. Machine Learning*, Williamstown, MA, Jun.–Jul. 2001, pp. 234–241.

[104] J. Abellán and A. R. Masegosa, "Bagging decision trees on data sets with classification noise," in *Proc. 6th Int. Conf. Foundations of Information and Knowledge Systems*, Sofia, Bulgaria, Feb. 2010, pp. 248–265.

[105] R. E. Schapire, Y. Freund, P. Barlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," in *Proc. 14th Int. Conf. Machine Learning*, Nashville, TN, Jul. 1997, pp. 322–330.

[106] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Ann. Stat.*, vol. 26, no. 5, pp. 1651–1686, 1998.

[107] T. Onoda, G. Rätsch, and K.-R. Müller, "An asymptotic analysis of adaboost in the binary classification case," in *Proc. Int. Conf. Artificial Neural Networks*, Skövde, Sweden, Sep. 1998, pp. 195–200.

[108] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Mach. Learn.*, vol. 42, no. 3, pp. 287–320, 2001.

[109] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[110] K. M. Ali and M. J. Pazzani, "Error reduction through learning multiple descriptions," *Mach. Learn.*, vol. 24, pp. 173–202, 1996.

[111] G. M. Weiss, "Learning with rare cases and small disjuncts," in *Proc. 12th Int. Conf. Machine Learning*, Tahoe City, CA, Jul. 1995, pp. 558–565.

[112] M. Hills, "Allocation rules and their error rates," *J. Roy. Stat. Soc. B Met.*, vol. 28, no. 1, pp. 1–31, 1966.

[113] G. L. Libralon, A. C. P. de Leon Ferreira de Carvalho, and A. C. Lorena, "Pre-processing for noise detection in gene expression classification data," *J. Brazil. Comput. Soc.*, vol. 15, no. 1, pp. 3–11, 2009.

[114] A. C. Lorena and A. C. Carvalho, "Evaluation of noise reduction techniques in the splice junction recognition problem," *Genet. Mol. Biol.*, vol. 27, no. 4, pp. 665–672, 2004.

[115] N. Segata, E. Blanzieri, and P. Cunningham, "A scalable noise reduction technique for large case-based systems," in *Proc. 8th Int. Conf. Case-Based Reasoning: Case-Based Reasoning Research and Development*, Seattle, WA, Jul. 2009, pp. 328–342.

[116] N. Segata, E. Blanzieri, S. Delany, and P. Cunningham, "Noise reduction for instance-based learning with a local maximal margin approach," *J. Intell. Inf. Syst.*, vol. 35, no. 2, pp. 301–331, 2010.

[117] L. G. Valiant, "A theory of the learnable," in *Proc. 16th Ann. ACM Symp. Theory of computing*, Washington, DC, Apr.–May 1984, pp. 436–445.

[118] P. D. Laird, *Learning from good and bad data*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1988.

[119] C. Gentile, "Improved lower bounds for learning from noisy examples: an information-theoretic approach," in *Proc. 11th Ann. Conf. Computational Learning Theory*, Madison, WI, Jul. 1998, pp. 104–115.

[120] Y. Sakakibara, "Noise-tolerant occam algorithms and their applications to learning decision trees," *Mach. Learn.*, vol. 11, no. 1, pp. 37–62, 1993.

[121] T. Bylander, "Learning linear threshold functions in the presence of classification noise," in *Proc. 7th Ann. Workshop Computational Learning Theory*, New Brunswick, NJ, Jul. 1994, pp. 340–347.

[122] A. Gaba and R. L. Winkler, "Implications of errors in survey data: A bayesian model," *Manage. Sci.*, vol. 38, no. 7, pp. 913–925, 1992.

[123] A. Tenenbein, "A double sampling scheme for estimating from binomial data with misclassifications," *J. Am. Stat. Assoc.*, vol. 65, no. 331, pp. 1350–1361, 1970.

[124] P. F. Thall, D. Jacoby, and S. O. Zimmerman, "Estimating genomic category probabilities from fluorescent in situ hybridization counts with misclassification," *J. Roy. Stat. Soc. C APP.*, vol. 45, no. 4, pp. 431–446, 1996.

[125] S. L. Stewart, K. C. Swallen, S. L. Glaser, P. L. Horn-Ross, and D. W. West, "Adjustment of cancer incidence rates for ethnic misclassification," *Biometrics*, vol. 54, no. 2, pp. 774–781, 1998.

[126] C. P. Lam and D. G. Stork, "Evaluating classifiers by means of test data with noisy labels," in *Proc. 18th Int. Joint Conf. Artificial intelligence*, Acapulco, Mexico, Aug. 2003, pp. 513–518.

[127] G. V. Cormack and A. Kolcz, "Spam filter evaluation with imprecise ground truth," in *Proc. 32nd Int. ACM SIGIR Conf. Research and Development In Information Retrieval*, Boston, MA, Jul. 2009, pp. 604–611.

[128] W. Zhang, R. Rekaya, and K. Bertrand, "A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer," *Bioinformatics*, vol. 22, no. 3, pp. 317–325, 2006.

[129] A. A. Shanab, T. M. Khoshgoftaar, and R. Wald, "Robustness of threshold-based feature rankers with data sampling on noisy and imbalanced data," in *Proc. 25th Int. Florida Artificial Intelligence Research Society Conf.*, Marco Island, FL, May 2012.

[130] R. Gerlach and J. Stamey, "Bayesian model selection for logistic regression with misclassified outcomes," *Stat. Model.*, vol. 7, no. 3, pp. 255–273, 2007.

[131] B. Frénay, G. Doquire, and M. Verleysen, "Feature selection with imprecise labels: Estimating mutual information in the presence of label noise," *Comput. Stat. Data An.*, submitted for publication.

[132] C.-M. Teng, "Evaluating noise correction," in *Proc. 6th Pacific Rim Int. Conf. Artificial intelligence*, Melbourne, Australia, Aug.–Sep. 2000, pp. 188–198.

[133] ——, "A comparison of noise handling techniques," in *Proc. 14th Int. Florida Artificial Intelligence Research Society Conf.*, Key West, FL, May 2001, pp. 269–273.

[134] ——, "Dealing with data corruption in remote sensing," in *Proc. 6th Int. Symp. Advances in Intelligent Data Analysis*, Madrid, Spain, Sep. 2005, pp. 452–463.

[135] S. Golzari, S. Doraisamy, M. N. Sulaiman, and N. I. Udzir, "The effect of noise on rwtsairs classifier," *Eur. J. Sci. Res.*, vol. 31, no. 4, pp. 632–641, 2009.

[136] C. Bouveyron and S. Girard, "Robust supervised classification with mixture models: Learning from data with uncertain labels," *Pattern Recogn.*, vol. 42, no. 11, pp. 2649–2658, 2009.

[137] H. Yin and H. Dong, "The problem of noise in classification: Past, current and future work," in *IEEE 3rd Int. Conf. Communication Software and Networks*, Xi'an, China, May 2011, pp. 412–416.

[138] C. E. Brodley and M. A. Friedl, "Identifying and eliminating mislabeled training instances," in *Proc. 13th Nat. Conf. Artificial intelligence*, Portland, Oregon, Aug. 1996, pp. 799–805.

[139] P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe, "Convexity, classification, and risk bounds," *J. Am. Stat. Assoc.*, vol. 101, no. 473, pp. 138–156, 2006.

[140] M. Thathachar and P. Sastry, *Networks of learning automata: techniques for online stochastic optimization*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2004.

[141] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–374, 2000.

[142] Y. Freund, "An adaptive version of the boost by majority algorithm," *Mach. Learn.*, vol. 43, no. 3, pp. 293–318, 2001.

[143] G. Rätsch, T. Onoda, and K.-R. Müller, "Regularizing adaboost," in *Advances in Neural Information Processing Systems 11*, Denver, CO, Nov.–Dec. 1998, pp. 564–570.

[144] G. Rätsch, T. Onoda, and K. R. Müller, "An improvement of adaboost to avoid overfitting," in *Proc. 5th Int. Conf. Neural Information Processing*, Kitakyushu, Japan, Oct. 1998, pp. 506–509.

[145] G. Rätsch, B. Schölkopf, A. J. Smola, S. Mika, T. Onoda, and K.-R. Müller, "Robust ensemble learning for data mining," in *Proc. 4th Pacific-Asia Conf. Knowledge Discovery and Data Mining, Current Issues and New Applications*, Kyoto, Japan, Apr. 2000, pp. 341–344.

[146] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Machine Learning*, Bari, Italy, Jul. 1996, pp. 148–156.

[147] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[148] J. Abellán and A. R. Masegosa, "An experimental study about simple decision trees for bagging ensemble on datasets with classification noise," in *Proc. 10th Eur. Conf. Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Verona, Italy, Jul. 2009, pp. 446–456.

[149] J. Abellán and S. Moral, "Building classification trees using the total uncertainty criterion," *Int. J. Intell. Syst.*, vol. 18, no. 12, pp. 1215–1225, 2003.

[150] J. Abellán and A. R. Masegosa, "Bagging schemes on the presence of class noise in classification," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6827–6837, 2012.

[151] A. Folleco, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, "Identifying learners robust to low quality data," *Informatica*, vol. 33, pp. 245–259, 2009.

[152] A. Folleco, T. M. Khoshgoftaar, J. V. Hulse, and L. A. Bullard, "Software quality modeling: The impact of class noise on the random forest classifier," in *IEEE Cong. Evolutionary Computation*, Hong Kong, China, Jun. 2008, pp. 3853–3859.

[153] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors," *IEEE Trans. Neural Netw.*, vol. 21, pp. 813–830, May 2010.

[154] M. Wardeh, F. Coenen, and T. Bench-Capon, "Arguing from experience to classifying noisy data," in *Proc. 11th Int. Conf. Data Warehousing and Knowledge Discovery*, Linz, Austria, Aug.–Sep. 2009, pp. 354–365.

[155] J. Sàez, M. Galar, J. Luengo, and F. Herrera, "A first study on decomposition strategies with data with class noise using decision trees," in *Proc. 7th Int. Conf. Hybrid Artificial Intelligent Systems: Part I*, Salamanca, Spain, Mar. 2012, pp. 25–35.

[156] G. M. Weiss and H. Hirsh, "The problem with noise and small disjuncts," in *Proc. Int. Conf. Machine Learning*, Madison, WI, Jul. 1998, pp. 574–578.

[157] J.-w. Sun, F.-y. Zhao, C.-j. Wang, and S.-f. Chen, "Identifying and correcting mislabeled training instances," in *Proc. Future Generation Communication and Networking - Vol. 1*, Jeju-Island, Korea, Dec. 2007, pp. 244–250.

[158] D. Gamberger, N. Lavrač, and S. Džeroski, "Noise elimination in inductive concept learning: A case study in medical diagnosis," in *Proc. 7th Int. Workshop Algorithmic Learning Theory*, Sydney, Australia, Oct. 1996, pp. 199–212.

[159] D. Gamberger and N. Lavrač, "Conditions for occam's razor applicability and noise elimination," in *Proc. 9th Eur. Conf. Machine Learning*, Prague, Czech Republic, Apr. 1997, pp. 108–123.

[160] ——, "Noise detection and elimination applied to noise handling in a krk chess endgame," in *Proc. 5th Int. Workshop Inductive Logic Programming*, Leuven, Belgium, Sep. 1997, pp. 59–75.

[161] D. Gamberger, R. Boskovic, N. Lavrac, and C. Groselj, "Experiments with noise filtering in a medical domain," in *Proc. 16th Int. Conf. Machine Learning*, Bled, Slovenia, Jun. 1999, pp. 143–151.

[162] D. Gamberger, N. Lavrac, and S. Dzeroski, "Noise detection and elimination in data preprocessing: experiments in medical domains," *Appl. Artif. Intell.*, vol. 14, pp. 205–223, 2000.

[163] X. Zhu and X. Wu, "Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering," *IEEE Trans. Knowl. Data Eng*, vol. 18, pp. 1435–1440, Oct. 2006.

[164] T. M. Khoshgoftaar and P. Rebours, "Generating multiple noise elimination filters with the ensemble-partitioning filter," in *Proc. 2004 IEEE Int. Conf. Information Reuse and Integration*, Las Vegas, NV, Nov. 2004, pp. 369–375.

[165] J. Thongkam, G. Xu, Y. Zhang, and F. Huang, "Support vector machine for outlier detection in breast cancer survivability prediction," in *Advanced Web and Network Technologies, and Applications*, Y. Ishikawa, J. He, G. Xu, Y. Shi, G. Huang, C. Pang, Q. Zhang, and G. Wang, Eds. Berlin: Springer, 2008, pp. 99–109.

[166] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Data cleaning for classification using misclassification analysis," *J. Adv. Comput. Intell. and Intell. Informatics*, vol. 14, no. 3, pp. 297–302, 2010.

[167] A. L. Miranda, L. P. Garcia, A. C. Carvalho, and A. C. Lorena, "Use of classification algorithms in noise detection and elimination," in *Proc. 4th Int. Conf. Hybrid Artificial Intelligence Systems*, Salamanca, Spain, Jun. 2009, pp. 417–424.

[168] N. Matic, I. Guyon, L. Bottou, J. Denker, and V. Vapnik, "Computer aided cleaning of large databases for character recognition," in *Proc. 11th IAPR Int. Conf. Pattern Recognition, Conf. B: Pattern Recognition Methodology and Systems*, The Hague, Netherlands, Aug.–Sep. 1992, pp. 330–333.

[169] I. Guyon, N. Matic, and V. Vapnik, "Discovering informative patterns and data cleaning," in *Advances in knowledge discovery and data mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Cambridge, MA: AAAI/MIT Press, 1996, pp. 181–203.

[170] A. Angelova, Y. Abu-mostafa, and P. Perona, "Pruning training sets for learning of object categories," in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 2005, pp. 494–501.

[171] G. H. John, "Robust decision trees: Removing outliers from databases," in *Proc. 1st Int. Conf. Knowledge Discovery and Data Mining*, Montreal, Quebec, Canada, Aug. 1995, pp. 174–179.

[172] T. Oates and D. Jensen, "The effects of training set size on decision tree complexity," in *Proc. 14th Int. Conf. Machine Learning*, Nashville, TN, Jul. 1997, pp. 254–262.

[173] S. Verbaeten, "Identifying mislabeled training examples in ilp classification problems," in *Proc. 12th Belgian-Dutch Conf. Machine Learning*, Utrecht, The Netherlands, Dec. 2002, pp. 71–78.

[174] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Comput.*, vol. 4, no. 6, pp. 888–900, 1992.

[175] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 607–616, Jun. 1996.

[176] E. Blanzieri and F. Melgani, "An adaptive svm nearest neighbor classifier for remotely sensed imagery," in *IEEE Int. Conf. Geoscience and Remote Sensing Symp.*, Denver, CO, Jul.–Aug. 2006, pp. 3931–3934.

[177] ——, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, pp. 1804–1811, May 2008.

[178] J. S. Sánchez, R. Barandela, A. I. Marqués, R. Alejo, and J. Badenas, "Analysis of new techniques to obtain quality training sets," *Pattern Recog. Lett.*, vol. 24, pp. 1015–1022, 2003.

[179] B. Chaudhuri, "A new definition of neighborhood of a point in multi-dimensional space," *Pattern Recog. Lett.*, vol. 17, no. 1, pp. 11–17, 1996.

[180] C. E. Brodley and M. A. Friedl, "Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data," in *Proc. 1996 Int. Geoscience and Remote Sensing Symp.*, Lincoln, NE, May 1996, pp. 27–31.

[181] S. Weisberg, *Applied linear regression*. New York, NY: Wiley, 1985.

[182] B. Sluban, D. Gamberger, and N. Lavrac, "Advances in class noise detection," in *Proc. 19th Eur. Conf. Artificial Intelligence*, Lisbon, Portugal, Aug. 2010, pp. 1105–1106.

[183] H. Berthelsen and B. Megyesi, "Ensemble of classifiers for noise detection in pos tagged corpora," in *Proc. 3rd Int. Workshop Text, Speech and Dialogue*, Brno, Czech Republic, Sep. 2000, pp. 27–32.

[184] S. Verbaeten and A. Van Assche, "Ensemble methods for noise elimination in classification problems," in *Proc. 4th Int. Conf. Multiple Classifier Systems*, Guildford, UK, Jun. 2003, pp. 317–325.

[185] X. Zhu, X. Wu, and Q. Chen, "Bridging local and global data cleansing: Identifying class noise in large, distributed data datasets," *Data Min. Knowl. Disc.*, vol. 12, no. 2-3, pp. 275–308, 2006.

[186] Y. Xiao, T. Khoshgoftaar, and N. Seliya, "The partitioning- and rule-based filter for noise detection," in *IEEE Int. Conf. Information Reuse and Integration*, Las Vegas, NV, Aug. 2005, pp. 205–210.

[187] C. Zhang, C. Wu, E. Blanzieri, Y. Zhou, Y. Wang, W. Du, and Y. Liang, "Methods for labeling error detection in microarrays based on the effect of data perturbation on the regression model," *Bioinformatics*, vol. 25, no. 20, pp. 2708–2714, 2009.

[188] Y. Zhou, C. Xing, W. Shen, Y. Sun, J. Wu, and X. Zhou, "A fast algorithm for outlier detection in microarray," in *Proc. Int. Conf. Advances in Computer Science, Environment, Ecoinformatics, and Education*, Wuhan, China, Aug. 2011, pp. 513–519.

[189] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, pp. 21–27, Jan. 1967.

[190] B. Dasarathy, *Nearest neighbor (NN) norms: nn pattern classification techniques*. Washington, DC: IEEE Computer Society Press, 1991.

[191] P. Devijver and J. Kittler, *Pattern recognition: a statistical approach*. Englewood Cliffs, London, UK: Prentice-Hall, 1982.

[192] R. Pan, Q. Yang, and S. J. Pan, "Mining competent case bases for case-based reasoning," *Artif. Intell.*, vol. 171, no. 16-17, pp. 1039–1068, 2007.

[193] D. R. Wilson and T. R. Martinez, "Instance pruning techniques," in *Proc. Int. Conf. Machine Learning*, Nashville, TN, Jul. 1997, pp. 403–411.

[194] S. J. Delany, N. Segata, and B. M. Namee, "Profiling instances in noise reduction," *Knowl.-Based Syst.*, vol. 31, pp. 28–40, 2012.

[195] P. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. 14, pp. 515–516, May 1968.

[196] G. W. Gates, "The reduced nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. 18, pp. 431–433, May 1972.

[197] S. J. Delany and P. Cunningham, "An analysis of case-base editing in a spam filtering system," in *Proc. 7th Eur. Conf. Case Based Reasoning*, Madrid, Spain, Aug.–Sep. 2004, pp. 128–141.

[198] A. Franco, D. Maltoni, and L. Nanni, "Data pre-processing through reward-punishment editing," *Pattern Anal. Appl.*, vol. 13, no. 4, pp. 367–381, 2010.

[199] L. Nanni and A. Franco, "Reduced reward-punishment editing for building ensembles of classifiers," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2395–2400, 2011.

[200] J. Koplowitz, "On the relation of performance to editing in nearest neighbor rules," *Pattern Recogn.*, vol. 13, no. 3, pp. 251–255, 1981.

[201] G. Libralon, A. Carvalho, and A. Lorena, "Ensembles of pre-processing techniques for noise detection in gene expression data," in *Proc. 15th Int. Conf. Advances in neuro-information processing - Vol. I*, Auckland, New Zealand, Nov. 2009, pp. 486–493.

[202] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. on Syst., Man, Cybern.*, vol. 2, pp. 408–421, Jul. 1972.

[203] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. 6, pp. 448 –452, Jun. 1976.

[204] D. W. Aha and D. Kibler, "Noise-tolerant instance-based learning algorithms," in *Proc. 11th Int. Joint Conf. Artificial intelligence - Vol. 1*, Detroit, MI, Aug. 1989, pp. 794–799.

[205] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

[206] A. C. Lorena, G. E. A. P. A. Batista, A. C. P. L. F. de Carvalho, and M. C. Monard, "The influence of noisy patterns in the performance of learning methods in the splice junction recognition problem," in *Proc. 7th Brazilian Symp. Neural Networks*, Recife, Brazil, Nov. 2002, pp. 31–37.

[207] I. Tomek, "Two modifications of cnn," *IEEE Trans. Syst., Man, Cybern.*, vol. 6, pp. 769 –772, Nov. 1976.

[208] M. R. Smith and T. Martinez, "Improving classification accuracy by identifying and removing instances that should be misclassified," in *Proc. Int. Joint Conf. Neural Networks*, San Jose, CA, Jul.–Aug. 2011, pp. 2690–2697.

[209] F. Muhlenbach, S. Lallich, and D. A. Zighed, "Identifying and handling mislabelled instances," *J. Intell. Inf. Syst.*, vol. 22, no. 1, pp. 89–109, 2004.

[210] M. Tüceryan and T. Chorzempa, "Relative sensitivity of a family of closest-point graphs in computer vision applications," *Pattern Recogn.*, vol. 24, no. 5, pp. 361–373, 1991.

[211] J. W. Jaromczyk and G. T. Toussaint, "Relative neighborhood graphs and their relatives," *Proc. of the IEEE*, vol. 80, pp. 1502–1517, Sep. 1992.

[212] W. Du and K. Urahama, "Error-correcting semi-supervised learning with mode-filter on graphs," in *12th Int. Conf. Computer Vision Workshops*, Kyoto, Japan, Sep.–Oct. 2009.

[213] ——, "Error-correcting semi-supervised pattern recognition with mode filter on graphs," in *2nd Int. Symp. Aware Computing*, Tainan, Taiwan, Nov. 2010, pp. 6–11.

[214] S. Lallich, F. Muhlenbach, and D. A. Zighed, "Improving classification by removing or relabeling mislabeled instances," in *Proc. 13th Int. Symp. Foundations of Intelligent Systems*, Lyon, France, Jun. 2002, pp. 5–15.

[215] A. Karmaker and S. Kwek, "A boosting approach to remove class label noise," *Int. J. Hybrid Intell. Syst.*, vol. 3, no. 3, pp. 169–177, 2006.

[216] Y. Gao, F. Gao, and X. Guan, "Improved boosting algorithm with adaptive filtration," in *Proc. 8th World Cong. Intelligent Control and Automation*, Jinan, China, Jul. 2010, pp. 3173–3178.

[217] V. Wheway, "Using boosting to detect noisy data," in *Advances in Artificial Intelligence. PRICAI 2000 Workshop Reader*, R. Kowalczyk, S. W. Loke, N. E. Reed, and G. J. Williams, Eds. Berlin: Springer Verlag, 2001, pp. 123–132.

[218] L. Breiman, "Arcing the edge," Univ. California, Berkeley, CA, Tech. Rep. 486, 1997.

[219] N. Ghoggalii and F. Melgani, "Automatic ground-truth validation with genetic algorithms for multispectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, pp. 2172–2181, Jul. 2009.

[220] X. Zeng and T. R. Martinez, "An algorithm for correcting mislabeled data," *Intell. Data Anal.*, vol. 5, pp. 491–502, 2001.

[221] X. Zeng and T. Martinez, "A noise filtering method using neural networks," in *IEEE Int. Workshop Soft Computing Techniques in Instrumentation, Measurement and Related Applications*, Provo, UT, May 2003, pp. 26–31.

[222] X. Zeng and T. R. Martinez, "Using decision trees and soft labeling to filter mislabeled data," *J. Intell. Syst.*, vol. 17, no. 4, pp. 331–354, 2011.

[223] S. Cuendet, D. Hakkani-Tür, and E. Shriberg, "Automatic labeling inconsistencies detection and correction for sentence unit segmentation in conversational speech," in *4th Int. Conf. Machine Learning for Multimodal Interaction*, Brno, Czech Republic, Jun. 2008, pp. 144–155.

[224] J. Van Hulse and T. Khoshgoftaar, "Knowledge discovery from imbalanced and noisy data," *Data Knowl. Eng.*, vol. 68, no. 12, pp. 1513–1542, 2009.

[225] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Folleco, "An empirical study of the classification performance of learners on imbalanced and noisy software quality data," in *Proc. IEEE Int. Conf. Information Reuse and Integration*, Las Vegas, NV, Aug. 2007, pp. 651–658.

[226] A. Srinivasan, S. Muggleton, and M. Bain, "Distinguishing exceptions from noise in non monotonic learning," in *Proc. 2nd Int. Workshop Inductive Logic Programming*, Tokyo, Japan, Jun. 1992, pp. 97–107.

[227] M. Evans, I. Guttman, Y. Haitovsky, and T. Swartz, "Bayesian analysis of binary data subject to misclassification," in *Bayesian Analysis in Statistics and Econometrics: Essays in Honor of Arnold Zellner*, D. Berry, K. Chaloner, and J. Geweke, Eds. New York, NY: Wiley, 1996, pp. 67–77.

[228] T. Swartz, Y. Haitovsky, A. Vexler, and T. Yang, "Bayesian identifiability and misclassification in multinomial data," *Can. J. Stat.*, vol. 32, no. 3, pp. 285–302, 2004.

[229] A. Gaba, "Inferences with an unknown noise level in a bernoulli process," *Manage. Sci.*, vol. 39, no. 10, pp. 1227–1237, 1993.

[230] R. L. Winkler, "Information loss in noisy and dependent processes," in *Bayesian Statistics 2*, J. M. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M.Smith, Eds. Amsterdam: North-Holland, 1985, pp. 559–570.

[231] M.-G. Basànez, C. Marshall, H. Carabin, T. Gyorkos, and L. Joseph, "Bayesian statistics for parasitologists," *Trends Parasitol.*, vol. 20, no. 2, pp. 85–91, 2004.

[232] W. O. Johnson and J. L. Gastwirth, "Bayesian inference for medical screening tests: Approximations useful for the analysis of acquired immune deficiency syndrome," *J. Roy. Stat. Soc. B Met.*, vol. 53, no. 2, pp. 427–439, 1991.

[233] L. Joseph and T. W. Gyorkos, "Inferences for likelihood ratios in the absence of a "gold standard"," *Med. Decis. Making*, vol. 16, no. 4, pp. 412–417, 1996.

[234] P. Gustafson, N. D. Le, and R. Saskin, "Case-control analysis with partial knowledge of exposure misclassification probabilities," *Biometrics*, vol. 57, no. 2, pp. 598–609, 2001.

[235] R. Rekaya, K. A. Weigel, and D. Gianola, "Threshold model for misclassified binary responses with applications to animal breeding," *Biometrics*, vol. 57, no. 4, pp. 1123–1129, 2001.

[236] C. D. Paulino, P. Soares, and J. Neuhaus, "Binomial regression with misclassification," *Biometrics*, vol. 59, no. 3, pp. 670–675, 2003.

[237] M. Ruiz, F. J. Girón, C. J. Pérez, J. Martín, and C. Rojano, "A bayesian model for multinomial sampling with misclassified data," *J. Appl. Stat.*, vol. 35, no. 4, pp. 369–382, 2008.

[238] J. Liu, P. Gustafson, N. Cherry, and I. Burstyn, "Bayesian analysis of a matched case-control study with expert prior information on both the misclassification of exposure and the exposure-disease association," *Stat. Med.*, vol. 28, no. 27, pp. 3411–3423, 2009.

[239] J. A. Achcar, E. Z. Martinez, and F. Louzada-Neto, "Binary data in the presence of misclassifications," in *16th Symp. Int. Association for Statistical Computing*, Praga, Czech Republic, Aug. 2004, pp. 581–587.

[240] P. McInturff, W. O. Johnson, D. Cowling, and I. A. Gardner, "Modelling risk when binary outcomes are subject to error," *Stat. Med.*, vol. 23, no. 7, pp. 1095–1109, 2004.

[241] C. D. Paulino, G. Silva, and J. A. Achcar, "Bayesian analysis of correlated misclassified binary data," *Comput. Stat. Data An.*, vol. 49, no. 4, pp. 1120–1131, 2005.

[242] F. O. Kaster, B. H. Menze, M.-A. Weber, and F. A. Hamprecht, "Comparative validation of graphical models for learning tumor segmentations from noisy manual annotations," in *Proc. 2010 Int. MICCAI Conf. Medical Computer Vision: Recognition Techniques and Applications in Medical Imaging*, Beijing, China, Sep. 2011, pp. 74–85.

[243] A. Hadgu, N. Dendukuri, and J. Hilden, "Evaluation of nucleic acid amplification tests in the absence of a perfect gold-standard test: A review of the statistical and epidemiologic issues," *Epidemiology*, vol. 16, no. 5, pp. 604–612, 2005.

[244] M. Ladouceur, E. Rahme, C. A. Pineau, and L. Joseph, "Robustness of prevalence estimates derived from misclassified data from administrative databases," *Biometrics*, vol. 63, no. 1, pp. 272–279, 2007.

[245] K. Robbins, S. Joseph, W. Zhang, R. Rekaya, and J. Bertrand, "Classification of incipient alzheimer patients using gene expression data: Dealing with potential misdiagnosis," *Online J. Bioinformatics*, vol. 7, no. 1, pp. 22–31, 2006.

[246] D. Hernandez-Lobato, J. M. Hernandez-Lobato, and P. Dupont, "Robust multi-class gaussian process classification," in *Advances in Neural Information Processing Systems 24*, Granada, Spain, Dec. 2011, pp. 280–288.

[247] H.-C. Kim and Z. Ghahramani, "Bayesian gaussian process classification with the em-ep algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1948–1959, Dec. 2006.

[248] F. L. Wauthier and M. I. Jordan, "Heavy-tailed process priors for selective shrinkage," in *Advances in Neural Information Processing Systems 23*, Vancouver, British Columbia, Canada, Dec. 2010, pp. 2406–2414.

[249] E. Eskin, "Detecting errors within a corpus using anomaly detection," in *Proc. 1st North American Chapter ACL Conf.*, Seattle, WA, May 2000, pp. 148–153.

[250] Y. Mansour and M. Parnas, "Learning conjunctions with noise under product distributions," *Inform. Process. Lett.*, vol. 68, no. 4, pp. 189–196, 1998.

[251] 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.*, vol. 40, no. 12, pp. 3349–3357, 2007.

[252] J. Bootkrajang and A. Kaban, "Multi-class classification in the presence of labelling errors," in *Proc. 19th Eur. Symp. Artificial Neural Networks*, Bruges, Belgium, Apr. 2011, pp. 345–350.

[253] B. Frénay, G. de Lannoy, and M. Verleysen, "Label noise-tolerant hidden markov models for segmentation: application to ecgs," in *Proc. 2011 Eur. Conf. Machine learning and Knowledge Discovery in Databases - Vol. I*, Athens, Greece, Sep. 2011, pp. 455–470.

[254] J. Larsen, L. N. Andersen, M. Hintz-madsen, and L. K. Hansen, "Design of robust neural network classifiers," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Seattle, WA, May 1998, pp. 1205–1208.

[255] S. Sigurdsson, J. Larsen, L. K. Hansen, P. A. Philipsen, and H. C. Wulf, "Outlier estimation and detection: Application to skin lesion classification," in *Int. Conf. Acoustics, Speech and Signal Processing*, Orlando, FL, May 2002, pp. 1049–1052.

[256] H.-C. Kim and Z. Ghahramani, "Outlier robust gaussian process classification," in *Proc. 2008 Joint IAPR Int. Workshop Structural, Syntactic, and Statistical Pattern Recognition*, Orlando, FL, Dec. 2008, pp. 896–905.

[257] H. Valizadegan and P.-N. Tan, "Kernel based detection of mislabeled training examples," in *SIAM Conf. Data Mining*, Minneapolis, MN, Apr. 2007.

[258] R. Xu and D. I. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, pp. 645–678, May 2005.

[259] T. Hastie and R. Tibshirani, "Discriminant analysis by gaussian mixtures," *J. Roy. Stat. Soc. B Met.*, vol. 58, no. 1, pp. 155–176, 1996.

[260] C. Bouveyron, "Weakly-supervised classification with mixture models for cervical cancer detection," in *Proc. 10th Int. Work-Conf. Artificial Neural Networks: Part I: Bio-Inspired Systems: Computational and Ambient Intelligence*, Salamanca, Spain, Jun. 2009, pp. 1021–1028.

[261] C. Bouveyron, S. Girard, and M. Olteanu, "Supervised classification of categorical data with uncertain labels for DNA barcoding," in *17th Eur. Symp. Artificial Neural Networks*, Bruges, Belgique, Apr. 2009, pp. 29–34.

[262] U. Rebbapragada and C. E. Brodley, "Class noise mitigation through instance weighting," in *Proc. 18th Eur. Conf. Machine Learning*, Warsaw, Poland, Sep. 2007, pp. 708–715.

[263] N. El Gayar, F. Schwenker, and G. Palm, "A study of the robustness of knn classifiers trained using soft labels," in *Proc. 2nd Int. Conf. Artificial Neural Networks in Pattern Recognition*, Ulm, Germany, Aug.–Sep. 2006, pp. 67–80.

[264] J. M. Keller, M. R. Gray, and J. J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, pp. 580–585, Jul.–Aug. 1985.

[265] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton Univ. Press, 1976.

[266] P. Smets, "Decision making in the tbm: the necessity of the pignistic transformation," *Int. J. Approx. Reason.*, vol. 38, no. 2, pp. 133–147, 2005.

[267] P. Vannoorenberghe and T. Denœux, "Handling uncertain labels in multiclass problems using belief decision trees," in *Proc. 9th Int. Conf. Information Processing and Management of Uncertainty*, Annecy, France, Jul. 2002, pp. 1919–1926.

[268] E. Côme, L. Oukhellou, T. Denœux, and P. Aknin, "Mixture model estimation with soft labels," in *Soft Methods for Handling Variability and Imprecision*, D. Dubois, M. A. Lubiano, H. Prade, M. Ángeles Gil, P. Grzegorzewski, and O. Hryniewicz, Eds. Berlin: Springer, 2008, pp. 165–174.

[269] ——, "Learning from partially supervised data using mixture models and belief functions," *Pattern Recogn.*, vol. 42, pp. 334–348, 2009.

[270] B. Quost and T. Denœux, "Learning from data with uncertain labels by boosting credal classifiers," in *Proc. 1st ACM SIGKDD Workshop Knowledge Discovery from Uncertain Data*, Paris, France, Jun. 2009, pp. 38–47.

[271] M. Tabassian, R. Ghaderi, and R. Ebrahimpour, "Knitted fabric defect classification for uncertain labels based on dempster-shafer theory of evidence," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5259–5267, 2011.

[272] Z. Younes, F. abdallah, and T. Denœux, "Evidential multi-label classification approach to learning from data with imprecise labels," in *Proc. 13th Int. Conf. Information Processing and Management of Uncertainty*, Dortmund, Germany, Jun.–Jul. 2010, pp. 119–128.

[273] A. Ganapathiraju, J. Picone, and M. State, "Support vector machines for automatic data cleanup," in *Proc. 6th Int. Conf. Spoken Language Processing*, Beijing, China, Oct. 2000, pp. 210–213.

[274] R. Rosales, G. Fung, and W. Tong, "Automatic discrimination of mislabeled training points for large margin classifiers," in *Proc. Snowbird Machine Learning Workshop*, Clearwater, FL, Apr. 2009, pp. 1–2.

[275] O. Dekel and O. Shamir, "Good learners for evil teachers," in *Proc. 26th Ann. Int. Conf. Machine Learning*, Montreal, Quebec, Canada, Jun. 2009, pp. 233–240.

[276] C.-f. Lin and S.-d. Wang, "Training algorithms for fuzzy support vector machines with noisy data," *Pattern Recog. Lett.*, vol. 25, no. 14, pp. 1647–1656, 2004.

[277] W. An and M. Liang, "Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises," *Neurocomputing*, in press.

[278] D.-F. Li, W.-C. Hu, W. Xiong, and J.-B. Yang, "Fuzzy relevance vector machine for learning from unbalanced data and noise," *Pattern Recog. Lett.*, vol. 29, no. 9, pp. 1175–1181, 2008.

[279] M. Sabzekar, H. S. Yazdi, M. Naghibzadeh, and S. Effati, "Emphatic constraints support vector machine," *Int. J. Comput. Elec. Eng.*, vol. 2, no. 2, pp. 296–306, 2010.

[280] L. Xu, K. Crammer, and D. Schuurmans, "Robust support vector machine training via convex outlier ablation," in *Proc. 21st Nat. Conf. Artificial intelligence - Vol. 1*, Boston, MA, Jul. 2006, pp. 536–542.

[281] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Functional gradient techniques for combining hypotheses," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 221–246.

[282] N. Krause and Y. Singer, "Leveraging the margin more carefully," in *Proc. 21st Int. Conf. Machine learning*, Banff, Alberta, Canada, Jul. 2004, pp. 63–70.

[283] H. Masnadi-Shirazi and N. Vasconcelos, "On the design of loss functions for classification: theory, robustness to outliers, and savageboost," in *Advances in Neural Information Processing Systems 21*, Dec. 2008, pp. 1049–1056.

[284] H. Masnadi-Shirazi, V. Mahadevan, and N. Vasconcelos, "On the design of robust classifiers for computer vision," in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, Jun. 2010, pp. 779–786.

[285] G. Stempfel and L. Ralaivola, "Learning svms from sloppily labeled data," in *Proc. 19th Int. Conf. Artificial Neural Networks: Part I*, Limassol, Cyprus, Sep. 2009, pp. 884–893.

[286] R. Khardon and G. Wachman, "Noise tolerant variants of the perceptron algorithm," *J. Mach. Learn. Res.*, vol. 8, pp. 227–248, 2007.

[287] A. Kowalczyk, A. J. Smola, and R. C. Williamson, "Kernel machines and boolean functions," in *Advances in Neural Information Processing Systems 14*, Vancouver, British Columbia, Canada, Dec. 2001, pp. 439–446.

[288] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," *Mach. Learn.*, vol. 46, no. 1-3, pp. 361–387, 2002.

[289] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines And Other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge Univ. Press, 2000.

[290] W. Krauth and Mézard, "Learning algorithms with optimal stability in neural networks," *J. Phys. A: Math. Gen.*, vol. 20, pp. L745–L752, 1987.

[291] P. Clark and T. Niblett, "The cn2 induction algorithm," *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, 1989.

[292] C. Domingo and O. Watanabe, "Madaboost: A modification of adaboost," in *Proc. 13th Ann. Conf. Computational Learning Theory*, San Francisco, CA, Jun. 2000, pp. 180–189.

[293] N. C. Oza, "Boosting with averaged weight vectors," in *Proc. 4th Int. Conf. Multiple classifier systems*, Guildford, UK, Jun. 2003, pp. 15–24.

[294] ——, "Aveboost2: Boosting for noisy data," in *Proc. 5th Int. Conf. Multiple Classifier Systems*, Cagliari, Italy, Jun. 2004, pp. 31–40.

[295] Y. Kim, "Averaged boosting: A noise-robust ensemble method," in *Proc. of the 7th Pacific-Asia conference on Advances in knowledge discovery and data mining*, Seoul, Korea, Apr.–May 2003, pp. 388–393.

[296] V. Gómez-Verdejo, M. Ortega-Moral, J. Arenas-García, and A. R. Figueiras-Vidal, "Boosting by weighting critical and erroneous samples," *Neurocomputing*, vol. 69, no. 7-9, pp. 679–685, 2006.

[297] A. Krieger, C. Long, and A. Wyner, "Boosting noisy data," in *Proc. 18th Int. Conf. Machine Learning*, Williamstown, MA, Jun.–Jul. 2001, pp. 274–281.

[298] G. I. Webb, "Multiboosting: A technique for combining boosting and wagging," *Mach. Learn.*, vol. 40, no. 2, pp. 159–196, 2000.

[299] I. Cantador and J. R. Dorronsoro, "Boosting parallel perceptrons for label noise reduction in classification problems," in *Proc. 1st Int. Work-Conf. Interplay Between Natural and Artificial Computation*, Las Palmas, Canary Islands, Spain, Jun. 2005, pp. 586–593.

[300] D. Guan, W. Yuan, Y.-K. Lee, and S. Lee, "Identifying mislabeled training data with the aid of unlabeled data," *Appl. Intell.*, vol. 35, no. 3, pp. 345–358, 2011.

[301] Y. Duan, Y. Gao, X. Ren, H. Che, and K. Yang, "Semi-supervised classification and noise detection," in *Proc. 6th Int. Conf. Fuzzy Systems and Knowledge Discovery - Vol. 1*, Tianjin, China, Aug. 2009, pp. 277–280.

[302] M.-R. Amini and P. Gallinari, "Semi-supervised learning with explicit misclassification modeling," in *Proc. 18th Int. Joint Conf. Artificial intelligence*, Acapulco, Mexico, Aug. 2003, pp. 555–560.

[303] M. Amini and P. Gallinari, "Semi-supervised learning with an imperfect supervisor," *Knowl. Inf. Syst.*, vol. 8, no. 4, pp. 385–413, 2005.

[304] A. Krithara, M. Amini, J.-M. Renders, and C. Goutte, "Semi-supervised document classification with a mislabeling error model," in *Proc. 28th Eur. Conf. IR Research*, London, UK, Apr. 2008, pp. 370–381.

[305] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, 2004.

[306] S. Ji and J. Ye, "Generalized linear discriminant analysis: A unified framework and efficient model selection," *IEEE Trans. Neural Netw.*, vol. 19, pp. 1768–1782, Oct. 2008.

[307] E. Niaf, R. Flamary, C. Lartizien, and S. Canu, "Handling uncertainties in svm classification," in *IEEE Workshop Statistical Signal Processing*, Nice, France, Jun. 2011, pp. 757–760.

[308] L. Daza and E. Acuna, "An algorithm for detecting noise on supervised classification," in *Proc. World Cong. Engineering and Computer Science 2007*, San Francisco, CA, Oct. 2007, pp. 701–706.

Benoît Frénay received the Engineer's degree from the Université catholique de Louvain (UCL), Belgium, in 2007. He is now Ph.D. student at the UCL Machine Learning Group. His main research interests in machine learning include support vector machines, extreme learning, graphical models, classification, data clustering, probability density estimation, feature selection and label noise.

Michel Verleysen received the M.S. and Ph.D. degrees in electrical engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He is Full Professor at the Université catholique de Louvain, and Honorary Research Director of the Belgian F.N.R.S. (National Fund for Scientific Research). He is editor-in-chief of the Neural Processing Letters journal (published by Springer), chairman of the annual ESANN conference (European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning), past associate editor of the IEEE Trans. on Neural Networks journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He is author or co-author of more than 250 scientific papers in international journals and books or communications to conferences with reviewing committee. He is the co-author of the scientific popularization book on artificial neural networks in the series "Que Sais-Je?", in French, and of the "Nonlinear Dimensionality Reduction" book published by Springer in 2007. His research interests include machine learning, artificial neural networks, self-organization, time-series forecasting, nonlinear statistics, adaptive signal processing, and high-dimensional data analysis.

Chapter 11

Label Noise-Tolerant Hidden Markov Models for Segmentation: Application to ECGs

The following article has been presented at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Athens, Greece, 5-9 September 2011. This paper extends the pioneer work of Lawrence and Schölkopf in order to obtain label noise-tolerant hidden Markov models. Experiments are conducted in the context of electrocardiogram signals segmentation. Related papers on electrocardiogram signals segmentation include [18–20]. Reprinted with kind permission of Springer Science+Business Media from [5].

Label Noise-Tolerant Hidden Markov Models for Segmentation: Application to ECGs

Benoît Frénay, Gaël de Lannoy*, and Michel Verleysen

Machine Learning Group, ICTEAM Institute, Université catholique de Louvain
3 place du Levant, B-1348 Louvain-la-Neuve, Belgium

Abstract. The performance of traditional classification models can adversely be impacted by the presence of label noise in training observations. The pioneer work of Lawrence and Schölkopf tackled this issue in datasets with independent observations by incorporating a statistical noise model within the inference algorithm. In this paper, the specific case of label noise in non-independent observations is rather considered. For this purpose, a label noise-tolerant expectation-maximisation algorithm is proposed in the frame of hidden Markov models. Experiments are carried on both healthy and pathological electrocardiogram signals with distinct types of additional artificial label noise. Results show that the proposed label noise-tolerant inference algorithm can improve the segmentation performances in the presence of label noise.

Keywords: label noise, hidden Markov models, expectation maximisation algorithm, segmentation, electrocardiograms.

1 Introduction

In standard situations, supervised machine learning algorithms learn their parameters to fit previously labelled data, called training observations, as best as possible. In real situations, however, it is difficult to guarantee perfect labelling, e.g. because of the subjectivity of the labelling task, of the lack of information or of communication noise. In particular, label errors are likely to arise in biomedical applications involving the tedious and time-consuming labelling of a large amount of data by one or several medical experts. The label noise issue is typically addressed in regression problems by assuming independent Gaussian noise on the regression target. In classification problems, although standard algorithms such as support vector machines are able to cope with outliers and feature noise to some degree, the label noise issue is however mostly left untreated.

Previous work addressing the label noise issue incorporated a noise model into a generative model which assumes independent and identically distributed (i.i.d.) observations [1–3]. Nevertheless, this issue is mostly left untreated in the case of models for the segmentation of sequential (non i.i.d.) observations such as hidden Markov models (HMMs). In this work, a variant of HMMs which is

* Gaël de Lannoy is funded by a Belgian F.R.I.A. grant.

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part I, LNAI 6911, pp. 455–470, 2011.

robust to the label noise is proposed. To illustrate the relevance of the proposed model, artificial electrocardiogram (ECG) signals generated using ECGSYN [4] and real ECG recordings from the Physiobank database [5] are used in the experiments. The label noise issue is indeed known affect the segmentation of waveform boundaries by experts in ECG signals [6]. Nevertheless, the proposed model also applies to any kind of sequential data facing the label noise issue, for example biomedical signals such as EEGs, EMGs and many others.

This paper is organised as follows. Section 2 reviews related work. Section 3 introduces hidden Markov models and two standard inference algorithms. Section 4 derives a new, label noise-tolerant algorithm. Section 5 quickly reviews electrocardiogram signals and details the experimental settings. Finally, empirical results are presented in Section 6 and conclusions are drawn in Section 7.

2 Related Work

Before presenting the state-of-the-art in classification with label noise, it is first important to distinguish the label noise issue from the semi-supervised paradigm where some data points in the training set are completely left unlabelled. Here, we rather consider the framework where an unknown proportion of the observations are wrongly labelled. To our knowledge, existing approaches to this problem are relatively few. These approaches can be divided in three categories: filtering approaches, model-based approaches and plausibilistic approaches.

Filtering techniques act as a preprocessing of the training set to either remove noisy observations or correct their labels. These methods involve the use of a criterion to detect mislabelled observations. For example, [7] uses disagreement in ensemble methods. Furthermore, [8] introduces an algorithm to iteratively modify the examples whose class label disagrees with the class labels of most of their neighbours. Eventually, [9] uses information gain to detect noisy labels.

On the other hand, model-based approaches tackle the label noise by incorporating the mislabelling process as an integral part of the probabilistic model. Pioneer work by [1] incorporated a probabilistic noise model in a kernel-Fisher discriminant for binary classification. Later, [2] extended this model by relaxing the Gaussian distribution assumption and carried out extensive experiments on more complex datasets, which convincingly demonstrated the value of explicit label noise modeling. More recently the same model has been extended to multi-class datasets [10]. Bouveyron et al. also proposes a distinct robust mixture discriminant analysis [3], which consists in two steps: (i) learning an unsupervised Gaussian mixture model and (ii) computing the probability that each cluster belongs to a given class.

Eventually, plausibilistic approaches assume that the experts have explicitly provided uncertainties over labels. Specific algorithms are then developed to integrate and to focus on such uncertainties [11].

This work concentrates on model-based approaches to embed the noise process into classifiers. Model-based approaches have a sound theoretical foundation and tackle the noise issue in a more principled and transparent manner without

discarding potentially useful observations. Our contribution in this field is the development of a label noise-tolerant hidden Markov model for labelling of sequential (non i.i.d.) observations.

3 Hidden Markov Models for Segmentation

This section introduces hidden Markov models for segmentation. Two widely used inference algorithms are detailed: supervised learning and the Baum-Welch algorithm. Their application to ECG segmentation is discussed in Section 5.

3.1 Hidden Markov Models

HMMs are probabilistic models of time series generating processes where two distinct sequences are considered: the states $S_1 \dots S_T$ and observations $O_1 \dots O_T$. Here T is the length of these sequences (see Fig. 1). At a given time step t, the current observation O_t and the next state S_{t+1} are considered to depend only on the current state S_t. For example, in the case of ECGs, the process under study is the human heart. Hence, states and observations correspond to the inner state and electrical activity of the heart, respectively (see Section 5 for more details).

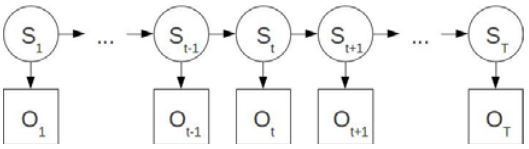


Fig. 1. Conditional dependencies in an hidden Markov model

Using the independence assumptions introduced above, an HMM is completely specified by its set of parameters $\Theta = (q, a, b)$ where q_i is the prior of state i, a_{ij} is the transition probability from state i to state j and b_i is the observation distributions for state i [12]. Usually, b_i is modelled by a Gaussian mixture model (GMM) with parameters $(\pi_{ik}, \mu_{ik}, \Sigma_{ik})$ where π_{ik}, μ_{ik} and Σ_{ik} are the prior, mean and covariance matrix of the kth Gaussian component, respectively.

Given a sequence of observations with expert annotations, the HMM inference problem consists in learning the parameters Θ from data. The remaining of this section presents two approaches for estimating the parameters. Once that an HMM is inferred, the segmentation of new signals can be done using the Viterbi algorithm, which looks for the most probable state sequence [12].

3.2 Algorithms for Inference

A simple solution to infer an HMM from data consists in assuming that the expert annotations are correct and trustworthy. Given this assumption, q and a are simply obtained by counting the state occurrences and transitions in the

data. Then, each observation distribution is fitted using the observations labelled accordingly. This approach has the advantage of being simple to implement and having a very low computational cost. However, if the labels are not perfect and polluted by some label noise, the produced HMM may be significantly altered.

The Baum-Welch algorithm is another, unsupervised algorithm [12]. More precisely, it assumes that the true labels are unknown, i.e. it ignores the expert annotations. The likelihood of the observations is maximised using an expectation-maximisation (EM) scheme [13], since no closed-form maximum likelihood estimator is available in this case. During the E step, the posteriors $P(S_t = i|O_1 \dots O_T)$ and $P(S_{t-1} = i, S_t = j|O_1 \dots O_T)$ are estimated for each time step t and states i and j. Then, these posteriors are used during the M step in order to estimate the prior vector q, the transition matrix a and the observation distributions b_i.

The main advantage of Baum-Welch is that wrong expert annotations should have no impact on the inferred HMM. However, in practice, expert annotations are used to compute a initial estimate of the HMM parameters, which is necessary for the first E step. Moreover, ignoring expert annotations can also be a disadvantage: if the expert uses a specific decomposition of the ECG dynamic, such subtleties may be lost in the unsupervised learning process.

4 A Label Noise-Tolerant Algorithm

Two algorithms for HMM inference have been introduced in Section 3. However, neither of them is satisfying when label noise is introduced. On the one hand, supervised learning is bound to trust blindly the expert annotations. Therefore, as shown in Section 6, label noise can degrade the segmentation quality. On the other hand, the Baum-Welch algorithm fails to encode precisely the expert knowledge. Indeed, as shown experimentally in Section 6, even predictions on clean, easy-to-segment signals do not match accurately the expert annotations.

This section introduces a new algorithm for HMM inference which lies in-between supervised learning and the Baum-Welch algorithm: expert annotations are used, but the label noise is modelled during the inference process in order to decrease the influence of wrong annotations.

4.1 Label Noise Modelling

Previous works showed the value of explicit label noise modelling for i.i.d. data in classification [1–3]. Here, a similar approach is used for non-independent sequential data. Two distinct, yet related sequences of states are considered (see Fig. 2): the sequence of observed, noisy annotations Y and the sequence of hidden, true labels S. In this paper, Y_t is assumed to depend only on S_t, i.e. Y_t is a (possibly noisy) copy of S_t.

An additional quantity $d_{ij} = P(Y_t = j|S_t = i, \Theta)$ is introduced for each pair of states (i, j), which is called the annotation probability. In order to avoid overfitting, the annotations probabilities take in this paper the restricted form

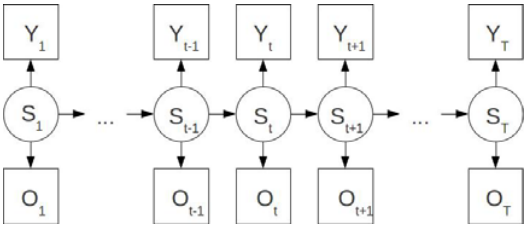


Fig. 2. Conditional dependencies in a label noise-tolerant hidden Markov model

$$d_{ij} = \begin{cases} 1 - p_i & (i = j) \\ \frac{p_i}{|\mathcal{S}|-1} & (i \neq j) \end{cases} \tag{1}$$

where p_i is the probability that the expert makes an error in state i and $|\mathcal{S}|$ is the number of possible states. Hence $d_{ii} = 1 - p_i$ is the probability of correct annotation in state i. Notice that d_{ij} is only used during inference. Here, Y is an extra layer put on a standard HMM to model the label noise. For segmentation, only the parameters linked to S and O are used, i.e. q, a, π, μ and Σ.

4.2 Finding the HMM Parameters with a Label Noise Model

Finding an estimate for both the HMM and label noise model parameters is achieved by maximising the incomplete log-likelihood

$$\log P(O, Y|\Theta) = \log \sum_S P(O, Y, S|\Theta), \tag{2}$$

where the sum spans all possible sequences of true states. As a closed-form solution does not exist, one can use the EM algorithm which is derived in the rest of this section. Notice that only approximate solutions are obtained, for EM algorithms are iterative procedures and may converge to local minima [13].

Definition of the $Q(\Theta, \Theta^{old})$ Function. The EM algorithm builds successive approximations of the incomplete log-likelihood in (2) and use them to maintain an estimate of the parameters [12, 14]. In the settings introduced above, it consists in alternatively (i) estimating the functional

$$Q(\Theta, \Theta^{old}) = \sum_S P(S|O, Y, \Theta^{old}) \log P(O, Y, S|\Theta) \tag{3}$$

using the current estimate Θ^{old} (E step) and (ii) maximising $Q(\Theta, \Theta^{old})$ with respect to the parameters Θ in order to update their estimate (M step). Since

$$P(O, Y, S|\Theta) = q_{s_1} \prod_{t=2}^{T} a_{s_{t-1}s_t} \prod_{t=1}^{T} b_{s_t}(o_t) \prod_{t=1}^{T} d_{s_t y_t}, \tag{4}$$

where o_t, y_t, s_1, s_{t-1} and s_t are the actual values taken by the random variables O_t, Y_t, S_1, S_{t-1} and S_t, the expression of $Q(\Theta, \Theta^{old})$ becomes

$$\sum_{i=1}^{|\mathcal{S}|} \gamma_1(i) \log q_i + \sum_{t=2}^{T} \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} \epsilon_t(i,j) \log a_{ij} + \sum_{t=1}^{T} \sum_{i=1}^{|\mathcal{S}|} \gamma_t(i) \log b_i(o_t) + \sum_{t=1}^{T} \sum_{i=1}^{|\mathcal{S}|} \gamma_t(i) \log d_{iy_t} \quad (5)$$

where the posterior probabilities γ and ϵ are defined as

$$\gamma_t(i) = P(S_t = i | O, Y, \Theta^{old}) \quad (6)$$

and

$$\epsilon_t(i,j) = P(S_{t-1} = i, S_t = j | O, Y, \Theta^{old}). \quad (7)$$

E Step. The γ and ϵ variables must be computed in order to evaluate (5), which is necessary for the M step. In standard HMMs, these quantities are estimated during the E step by the forward-backward algorithm [12, 14]. Indeed, if forward variables α, backward variables β and scaling coefficients c are defined as

$$\alpha_t(i) = P(S_t = i | O_{1\ldots t}, Y_{1\ldots t}, \Theta^{old}) \quad (8)$$

$$\beta_t(i) = \frac{P(O_{t+1\ldots T}, Y_{t+1\ldots T} | S_t = i, \Theta^{old})}{P(O_{t+1\ldots T}, Y_{t+1\ldots T} | O_{1\ldots t}, Y_{1\ldots t}, \Theta^{old})} \quad (9)$$

$$c_t = P(O_t, Y_t | O_{1\ldots t-1}, Y_{1\ldots t-1}, \Theta^{old}), \quad (10)$$

one eventually obtains

$$\gamma_t(i) = \alpha_t(i) \beta_t(i) \quad (11)$$

and

$$\epsilon_t(i,j) = \alpha_{t-1}(i) c_t^{-1} a_{ij} b_j(o_t) d_{jy_t} \beta_t(j). \quad (12)$$

Here, the scaling coefficients c_t are introduced in order to avoid numerical issues. Indeed, for sufficiently large T (i.e. 10 or more), the dynamic range of both α and β will exceed the precision range of any machine. The scaling factors c_t are therefore introduced to keep the values within reasonable bounds [12]. The incomplete likelihood can be computed using $P(O, Y | \Theta^{old}) = \prod_{t=1}^{T} c_t$.

The forward-backward algorithm consists in using the recursive relationship

$$\alpha_t(i) c_t = \begin{cases} q_i b_i(o_1) d_{iy_1} & (t = 1) \\ b_i(o_t) d_{iy_t} \sum_{j=1}^{|\mathcal{S}|} a_{ji} \alpha_{t-1}(j) & (t > 1). \end{cases} \quad (13)$$

linking the α and c variables and the recursive relationship

$$\beta_t(i) = \begin{cases} 1 & (t = T) \\ \frac{1}{c_{t+1}} \sum_{j=1}^{|\mathcal{S}|} a_{ij} b_j(o_{t+1}) d_{jy_{t+1}} \beta_{t+1}(j) & (t < T) \end{cases} \quad (14)$$

linking the β and c variables. The scaling coefficients can be computed using the constraint $\sum_{i=1}^{|\mathcal{S}|} \alpha_t(i) = 1$ jointly with (13).

M Step. The values of the γ and ϵ computed during the E step can be used to maximise $Q(\Theta, \Theta^{old})$. Using (5), one obtains

$$q_i = \frac{\gamma_1(i)}{\sum_{i=1}^{|\mathcal{S}|} \gamma_1(i)} \tag{15}$$

and

$$a_{ij} = \frac{\sum_{t=2}^{T} \epsilon_t(i,j)}{\sum_{t=2}^{T} \sum_{j=1}^{|\mathcal{S}|} \epsilon_t(i,j)} \tag{16}$$

for the state prior and transition probabilities. The GMMs parameters become

$$\pi_{il} = \frac{\sum_{t=1}^{T} \gamma_t(i,l)}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{17}$$

$$\mu_{il} = \frac{\sum_{t=1}^{T} \gamma_t(i,l) o_t}{\sum_{t=1}^{T} \gamma_t(i)} \tag{18}$$

and

$$\Sigma_{il} = \frac{\sum_{t=1}^{T} \gamma_t(i,l)(o_t - \mu_{il})^T (o_t - \mu_{il})}{\sum_{t=1}^{T} \gamma_t(i)} \tag{19}$$

where

$$\gamma_{il}(t) = \gamma_i(t) \frac{\pi_{il} b_{il}(o_t)}{b_i(o_t)}. \tag{20}$$

Eventually, the expert error probabilities are obtained using

$$p_i = \frac{\sum_{t|Y_t \neq i} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{21}$$

and the annotations probabilities can be computed using (1).

The EM Algorithm. The EM algorithm can be implemented using the equations detailed above. Θ must be initialised before the first E step. This problem is already addressed in the literature for all the parameters, except d. A simple solution, used in this paper, consists in initialising d using

$$d_{ij} = \begin{cases} 1 - p_e & (i = j) \\ \frac{p_e}{|\mathcal{S}|-1} & (i \neq j) \end{cases} \tag{22}$$

where p_e is a small probability of expert annotation error. Equivalently, one can set $p_i = p_e$. For example, $p_e = .05$ is used in the experiments in Section 6.

5 ECG Segmentation

This section (i) quickly reviews ECG segmentation and the use of HMMs in this context and (ii) details the methodology used for the experiments in Section 6.

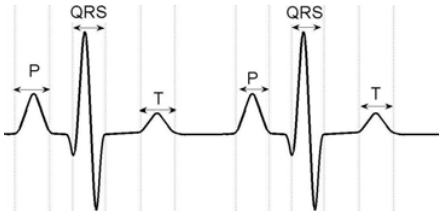


Fig. 3. Example of ECG signal, with annotations

5.1 ECG Signals

Electrocardiograms (ECGs) are periodic signals measuring the electrical activity of the heart. These time series are typically associated to a sequence of labels, called annotations (see Fig. 3). Indeed, physicians distinguish different kind of patterns called *waves*: P waves, QRS complexes and T waves. Moreover, physicians talk about baselines when the signal is flat, outside of waves. Here, only the B3 baseline between T and P waves is considered.

The ECG segmentation problem consists in predicting the labels for unlabeled observations, using the annotated part of the signal. Indeed, ECGs usually last for hours and it is of course impossible to annotate the entire signal manually.

In the context of ECG segmentation, (15) cannot be used directly. Indeed, only one ECG is available for HMM inference: the ECG of the patient under treatment. This is due to large inter-patient differences which prevent generalisation from one patient to the other. Here, q is simply estimated as the percentage of each observed annotation, as in the case of the supervised learning.

5.2 State of the Art

One of the most widely used, successful tool for ECG segmentation are HMMs [6, 15]. Typically, each ECG is firstly filtered using a 3-30 Hz band-pass filter. Then it is transformed using a continuous wavelet transform (WT) with an order 2 coiflet wavelet. The dyadic scales from 2^1 to 2^7 are kept in order to build the observations. Eventually, the resulting observations may be normalised component-wise, which is done in this paper. Fig. 4 shows the theoretical transitions in a HMM modelling an ECG.

Label noise has already been considered by [16] in the ECG context by using a semi-supervised approach. Annotations around boundaries are simply deleted,

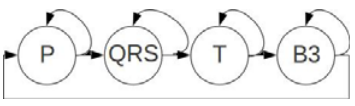


Fig. 4. Theoretical transitions in an ECG signal

which results in an intermediate situation between supervised learning and the Baum-Welch algorithm. Indeed, the remaining annotations are considered trustworthy and only the posteriors of the deleted annotations are estimated by EM. The width of the deletion window has to be selected, whereas this paper uses a noise model where the level of label noise is automatically estimated.

5.3 Experimental Settings

The two algorithms which are used for comparison are supervised learning and the Baum-Welch algorithm. Each emission model uses a GMM with 5 components. The EM algorithms are repeated 10 times and each repetition consists of at most 300 iterations. The initial mean of each GMM component is randomly chosen among the data in the corresponding class; the initial covariance matrix of each GMM component is set as a small multiple of the covariance matrix of the corresponding class.

Three classes of ECGs are used. Firstly, a set of 10 artificial ECGs are generated and annotated using the popular ECG waveform generator ECGSYN [4]. Secondly, 10 real ECGs are selected in the sinus MIT-QT database from Physiobank [5]. These Holter ECG recordings have been manually annotated by cardiologists with waveform boundaries for 30 to 50 selected beats in each recording. All recordings are sampled at 250 Hz. These ECGs were measured on real patients, but they are quite clean and easy to segment, for the patients were healthy. Thirdly, 10 ECGs are selected in the arrhythmia MIT-QT database from Physiobank [5]. These ECGs are more difficult to segment and the annotations are probably less reliable. Indeed, these patients were being treated for cardiac diseases and their ECG often differ significantly from the text-book ECGs. Only P waves, QRS complexes, T waves and B3 baselines are annotated.

Each ECG is segmented before and after the addition of artificial label noise. Different types and levels of artificial label noise are added to the annotations in order to test the robustness of each algorithm. The two label noises which are used here are called horizontal and uniform noise:

- The horizontal noise moves the boundaries of P and T waves by a random number of milliseconds drawn from a uniform distribution. This type of noise is particularly interesting in the context of ECG segmentation since it mimics the errors made by medical experts in practice. The uniform distribution used in the experiments is symmetric around zero and its half width is a given percentage of the considered wave duration.
- The uniform noise consist in randomly flipping a given percentage of the labels. This type of noise is the same as in previous experiments [1].

For each experiment, two measures are given: the average recall and precision. For each wave, the recall is the percentage of observations belonging to that wave which are correctly classified. The precision is the percentage of predicted labels which are correct, for a given label. Both measures are estimated using the ECGSYN annotations for the artificial ECGs, whereas human expert annotations are used for the real ECGs. In Section 5, recalls and precisions are systematically averaged over the four possible labels (P, QRS, T and B3).

For each algorithm, ECGs are split into training and test sets. The training set is used to learn the HMM, whereas the test set allows testing the HMM on independent data. For artificial ECGs, 10% of the signal is used for training, whereas the remaining 90% is used for test. For real ECGs, 50% of the signal is used for training, whereas the remaining 50% is used for test. This way, the size of the training sets are roughly equal for both artificial and real ECGs.

6 Experimental Results

This section compares the label noise-tolerant algorithm proposed in Section 4 to the two standard algorithms described in Section 3. The tests are carried out on three classes of ECGs, which are altered by two types of label noise. See Section 5 for more details about ECGs and the methodology used in this section.

6.1 Noise-Free Results

Tables 1 and 2 respectively show the recalls and precisions obtained on test beats for artificial, sinus and arrhythmia ECGs using supervised learning, Baum-Welch and the proposed algorithm. The annotations are the original annotations, without additional noise. Each ECG signal is segmented 40 times in order to evaluate the variability of the results. The results in the three first rows average the results of all runs for all ECGs, whereas other rows average the results of all runs for two selected ECGs. For the two selected ECGs, standard deviations are given. The standard deviations shown on the three first lines are the average of the standard deviations obtained for each ECG.

Results show that if one completely discards the available labels (i.e. with the Baum-Welch algorithm), the information loss is important. Indeed, the unsupervised algorithm always achieves significantly lower recalls and precisions. The results in terms of recall and precision are approximatively equal for the

Table 1. Recalls on original artificial, sinus and arrhythmia ECGs for supervised learning, Baum-Welch and the proposed algorithm

		supervised learning	Baum-Welch	proposed algorithm
average	artificial	95.21 ± 0.31	89.17 ± 2.20	95.14 ± 0.43
	sinus	95.35 ± 0.28	92.71 ± 1.60	95.60 ± 0.59
	arrhythmia	89.51 ± 0.69	82.48 ± 2.57	89.10 ± 0.90
ECG 1	artificial	94.98 ± 0.24	88.09 ± 2.33	94.87 ± 0.40
	sinus	95.75 ± 0.30	92.28 ± 1.50	96.44 ± 0.44
	arrhythmia	94.36 ± 0.46	81.77 ± 3.36	92.80 ± 1.79
ECG 2	artificial	93.34 ± 0.88	87.44 ± 3.17	93.50 ± 1.04
	sinus	95.88 ± 0.14	93.43 ± 0.74	96.01 ± 0.29
	arrhythmia	90.07 ± 0.35	88.01 ± 1.42	91.22 ± 0.46

Table 2. Precisions on original artificial, sinus and arrhythmia ECGs for supervised learning, Baum-Welch and the proposed algorithm

		supervised learning	Baum-Welch	proposed algorithm
average	artificial	95.53 ± 0.27	87.58 ± 3.06	95.34 ± 0.39
	sinus	95.86 ± 0.26	89.85 ± 2.23	94.38 ± 1.14
	arrhythmia	87.28 ± 0.75	77.37 ± 2.73	84.56 ± 1.45
ECG 1	artificial	95.51 ± 0.17	86.16 ± 3.10	95.14 ± 0.43
	sinus	96.81 ± 0.19	91.89 ± 1.50	95.96 ± 0.89
	arrhythmia	95.11 ± 0.96	72.13 ± 3.15	87.08 ± 4.03
ECG 2	artificial	94.76 ± 0.63	86.17 ± 4.26	94.67 ± 0.73
	sinus	96.42 ± 0.11	91.33 ± 1.87	95.82 ± 0.56
	arrhythmia	88.25 ± 0.32	86.14 ± 0.05	89.51 ± 0.49

proposed algorithm and supervised learning. One exception is the precision on arrhythmia signals, where the labels themselves are less reliable, making the performance assessment less reliable too.

6.2 Results with Horizontal Noise

Fig. 5, 6 and 7 show the results obtained for artificial, sinus and arrhythmia ECGs, respectively. The annotations are polluted by a horizontal noise, with the maximum boundary movement varying from 0% to 50% of the modified wave. For each figure, the first row shows the recall, whereas the second row shows the precision, both obtained on test beats. Each ECG signal is noised and segmented 40 times in order to evaluate the variability of the results. The curves in the first column average the results of all runs for all ECGs, whereas the curves in the second and third columns average the results of all runs for two selected ECGs. For the two last plots of each row, the error bars show the 95 % confidence interval around the mean on the 40 runs. The error bars shown on the first plot of each line are the average of the error bars obtained for each ECG.

Again, the performances of the unsupervised algorithm are the worst ones for small levels of noise. However, the results obtained using Baum-Welch seem to be less affected by the label noise. In most cases, the unsupervised algorithm achieves better results than supervised learning for large levels of noise. The effect of the noise level is probably due to the fact that the EM algorithm is initialised using the labelled observations. Therefore, the final result is also influenced by the label noise, for it depends on the initial starting point.

The performances of supervised learning and the label noise-tolerant algorithm are both affected by the increasing label noise. However, for large levels of noise, the label noise-tolerant algorithm achieves significantly better recalls and precisions than supervised learning. Supervised learning is only better in terms of precision for low levels of noise. Since the horizontal noise mimics errors made by medical experts, the above results suggest that using the proposed algorithm can improve the segmentation quality when the expert is not fully reliable.

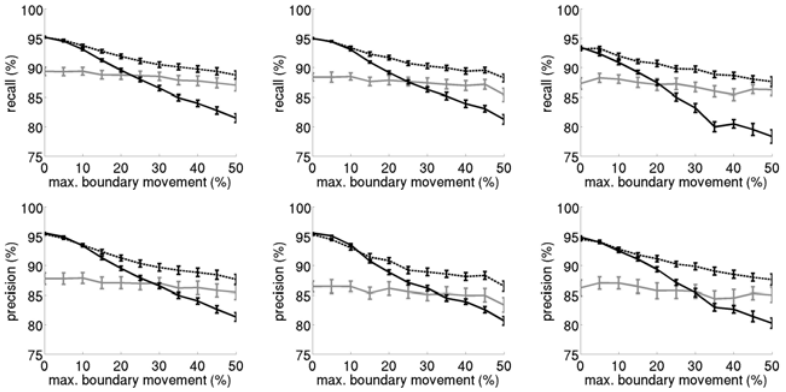


Fig. 5. Recalls and precisions on artificial ECGs with horizontal noise for supervised learning (black plain line), Baum-Welch (grey plain line) and the proposed algorithm (black dashed line), with respect to the percentage of the maximum boundary movement (0% to 50% of the modified wave). See text for details.

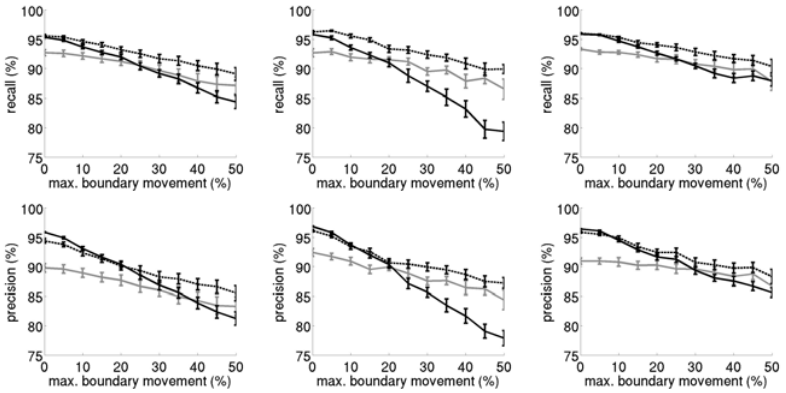


Fig. 6. Recalls and precisions on sinus ECGs with horizontal noise for supervised learning (black plain line), Baum-Welch (grey plain line) and the proposed algorithm (black dashed line), with respect to the percentage of the maximum boundary movement (0% to 50% of the modified wave). See text for details.

6.3 Results with Uniform Noise

Fig. 8, 9 and 10 shows the recalls and precisions obtained for artificial, sinus and arrhythmia ECGs, respectively. The annotations are polluted by a uniform noise, with a percentage of flipped labels varying from 0% to 20%. For each figure, the first row shows the recall, whereas the second row shows the precision, both obtained on test beats. Each ECG signal is noised and segmented 40 times in

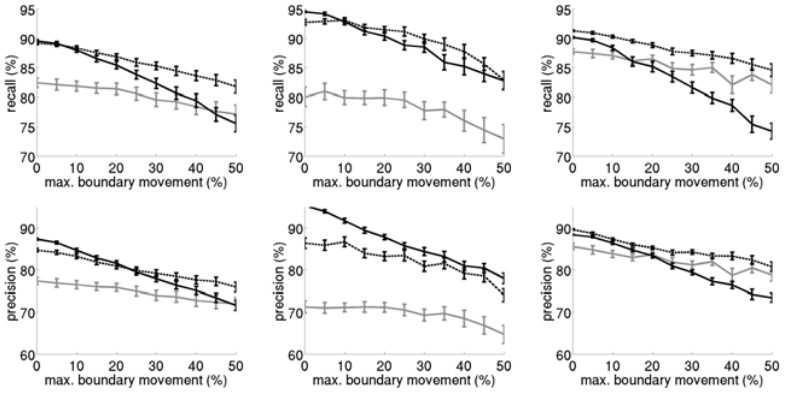


Fig. 7. Recalls and precisions on arrhythmia ECGs with horizontal noise for supervised learning (black plain line), Baum-Welch (grey plain line) and the proposed algorithm (black dashed line), with respect to the percentage of the maximum boundary movement (0% to 50% of the modified wave). See text for details.

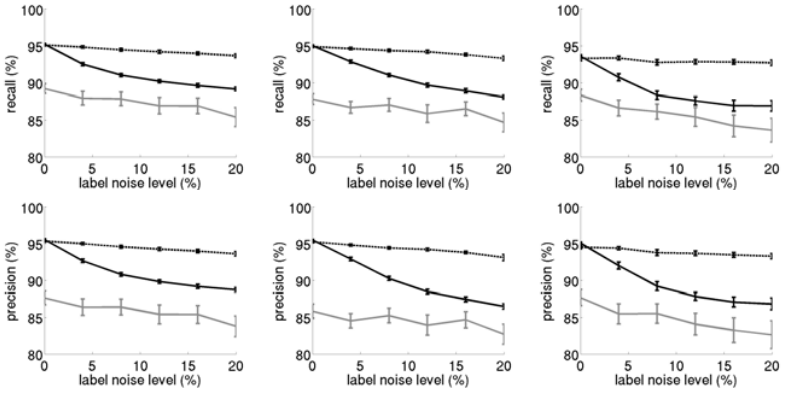


Fig. 8. Recalls and precisions on artificial ECGs with uniform noise for supervised learning (black plain line), Baum-Welch (grey plain line) and the proposed algorithm (black dashed line), with respect to the percentage of flipped labels (0% to 20%). See text for details.

order to evaluate the variability of the results. The curves in the first column average the results of all runs for all ECGs, whereas the curves in the second and third columns average the results of all runs for two selected ECGs. For the two last plots of each row, the error bars show the 95 % confidence interval around the mean on the 40 runs. The error bars shown on the first plot of each line are the average of the error bars obtained for each ECG.

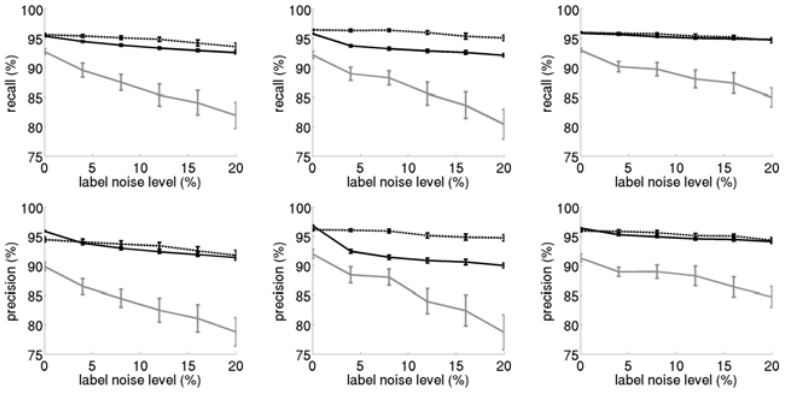


Fig. 9. Recalls and precisions on sinus ECGs with uniform noise for supervised learning (black plain line), Baum-Welch (grey plain line) and the proposed algorithm (black dashed line), with respect to the percentage of flipped labels (0% to 20%). See text for details.

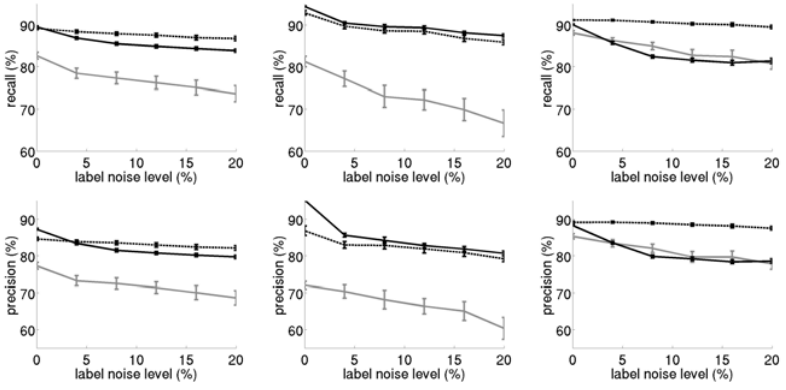


Fig. 10. Recalls and precisions on arrhythmia ECGs with uniform noise for supervised learning (black plain line), Baum-Welch (grey plain line) and the proposed algorithm (black dashed line), with respect to the percentage of flipped labels (0% to 20%). See text for details.

As for horizontal noise, the performances of Baum-Welch are significantly worse and decrease as the percentage of label noise increases. For the proposed algorithm, the recall and precision seem to be almost unaffected by the increasing level of label noise. For supervised learning, the recall and precision slowly decrease as the label noise increases. In terms of both recall and precision, the label noise-tolerant algorithm performs better than supervised learning when the level of noise is larger than 5%.

7 Conclusion

In this paper, a variant of the EM algorithm for label noise-tolerant HMM inference is proposed. More precisely, each observed label is assumed to be a noisy copy of the true, unknown state. The proposed EM algorithm relies on two steps to automatically estimate the level of noise in the set of available labels. First, during the E step, the posterior of the hidden state is estimated for each sample. Next, the M step computes the HMM parameters using the hidden true states, and not the noisy labels themselves, which results in a model which is less impacted by label noise.

Experiments are carried on both healthy and pathological ECGs signals artificially polluted by distinct types of label noise. Three types of inference algorithms for HMMs are compared: supervised learning, the Baum-Welch algorithm and the proposed noise-tolerant algorithm. The results show that the performances of the three approaches are adversely impacted by the level of label noise. However, the proposed noise-tolerant algorithm can yield better performances than the other two algorithms, which confirms the benefit of embedding the noise process into the inference algorithm. This improvement is particularly pronounced when the artificial label noise mimics errors made by medical experts, which suggests that the proposed algorithm could be useful when expert annotations are less reliable. The recall is improved for any label noise level, and the precision is improved for large levels of noise.

References

1. Lawrence, N.D., Schölkopf, B.: Estimating a kernel fisher discriminant in the presence of label noise. In: Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001, pp. 306–313. Morgan Kaufmann Publishers Inc, San Francisco (2001)
2. Li, Y., Wessels, L.F.A., de Ridder, D., Reinders, M.J.T.: Classification in the presence of class noise using a probabilistic kernel fisher method. Pattern Recognition 40, 3349–3357 (2007)
3. Bouveyron, C., Girard, S.: Robust supervised classification with mixture models: Learning from data with uncertain labels. Pattern Recognition 42, 2649–2658 (2009)
4. McSharry, P.E., Clifford, G.D., Tarassenko, L., Smith, L.A.: Dynamical model for generating synthetic electrocardiogram signals. IEEE Transactions on Biomedical Engineering 50(3), 289–294 (2003)
5. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 101(23), e215–e220 (2000)
6. Hughes, N.P., Tarassenko, L., Roberts, S.J.: Markov models for automated ECG interval analysis. In: NIPS 2004: Proceedings of the 16th Conference on Advances in Neural Information Processing Systems, pp. 611–618 (2004)
7. Brodley, C.E., Friedl, M.A.: Identifying Mislabeled Training Data. Journal of Artificial Intelligence Research 11, 131–167 (1999)

8. Barandela, R., Gasca, E.: Decontamination of training samples for supervised pattern recognition methods. In: Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition, pp. 621–630. Springer, London (2000)
9. Guyon, I., Matic, N., Vapnik, V.: Discovering informative patterns and data cleaning. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 181–203 (1996)
10. Bootkrajang, J., Kaban, A.: Multi-class classification in the presence of labelling errors. In: Proceedings of the 19th European Conference on Artificial Neural Networks, pp. 345–350 (2011)
11. Côme, E., Oukhellou, L., Denoeux, T., Aknin, P.: Mixture model estimation with soft labels. In: Proceedings of the 4th International Conference on Soft Methods in Probability and Statistics, pp. 165–174 (2008)
12. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
13. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
14. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics), 1st ed. 2006. corr. 2nd printing edition. Springer, Heidelberg (2007)
15. Clifford, G.D., Azuaje, F., McSharry, P.: Advanced Methods And Tools for ECG Data Analysis. Artech House, Inc., Norwood (2006)
16. Hughes, N.P., Roberts, S.J., Tarassenko, L.: Semi-supervised learning of probabilistic models for ecg segmentation. In: IEMBS 2004: Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, pp. 434–437 (2004)

Chapter 12

Estimating Mutual Information for Feature Selection in the Presence of Label Noise

The following article has been accepted for publication in the Computational Statistics & Data Analysis journal. Since the performances of feature selection algorithms often decrease when instances are wrongly labelled, this paper proposes a way to achieve feature selection for classification problems polluted by label noise. A method based on a probabilistic label noise model combined with a nearest neighbours-based entropy estimator is introduced to robustly evaluate the mutual information. Related papers about label noise include [3–5]. Reprinted with permission from [3].

ARTICLE IN PRESS

Computational Statistics and Data Analysis ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Estimating mutual information for feature selection in the presence of label noise

Benoît Frénay *, Gauthier Doquire, Michel Verleysen

Machine Learning Group, ICTEAM Institute, Université catholique de Louvain, Place du Levant 3, BE 1348, Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:
Received 4 April 2012
Received in revised form 1 May 2013
Accepted 1 May 2013
Available online xxxx

Keywords:
Label noise
Mutual information
Entropy estimation
Feature selection

ABSTRACT

A way to achieve feature selection for classification problems polluted by label noise is proposed. The performances of traditional feature selection algorithms often decrease sharply when some samples are wrongly labelled. A method based on a probabilistic label noise model combined with a nearest neighbours-based entropy estimator is introduced to robustly evaluate the mutual information, a popular relevance criterion for feature selection. A backward greedy search procedure is used in combination with this criterion to find relevant sets of features. Experiments establish that (i) there is a real need to take a possible label noise into account when selecting features and (ii) the proposed methodology is effectively able to reduce the negative impact of the mislabelled data points on the feature selection process.

1. Introduction

Performing feature selection is an essential preprocessing step for many data mining and pattern recognition applications, including classification (Guyon and Elisseeff, 2003; Dash and Liu, 1997). The objective is to determine, among the original set of features of a data set, which are the most relevant ones to achieve a particular task. In practice, the benefits of feature selection are numerous. First, it helps reducing the dimensionality of a data set. This aspect is particularly important when the data are high-dimensional. Indeed, learning in this context is a hard task, due to many difficulties known under the generic term *curse of dimensionality* (Bellman, 1961). In addition, it is likely that for a specific problem, some features are either irrelevant or redundant. Discarding these features generally improves the performances of classification models. Last, feature selection has the advantage over other dimensionality reduction strategies, such as feature extraction (Guyon et al., 2006), that it preserves the original features. This is of crucial importance in many industrial and medical applications, where the interpretation of the models is important.

Among the different possible solutions, filter methods are often preferred to achieve feature selection. Filters are based on the optimisation of a criterion which is independent of any prediction model; in practice, this makes them particularly fast compared to wrapper methods, which directly optimise the performances of a specific prediction model. Moreover, filter methods can be used in combination with any prediction model; for these reasons, they will be considered in this work. As a criterion of relevance, Shannon's mutual information (MI) (Shannon, 1948) is one of the most popular and successful choices for filter-based feature selection. Due to several reasons described in Section 2.1, MI possesses many required qualities for this task and has strong advantages over other well-known criteria such as the correlation coefficient.

The major problem when using MI is that, in general, it cannot be computed analytically but has to be estimated from the available data. Even if estimating MI has been intensively studied for one-dimensional features, estimating the MI between high-dimensional groups of features still remains a challenging task; however, it can prove to be very useful in practice

* Correspondence to: ICTEAM/ELEN, Université catholique de Louvain, Place du Levant 3, BE 1348, Louvain-la-Neuve, Belgium. Tel.: +32 10 478133; fax: +32 10 472598.
E-mail address: benoit.frenay@uclouvain.be (B. Frénay).

0167-9473/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.csda.2013.05.001

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

for feature selection. Recent works have addressed this problem, by showing the interest of a nearest-neighbours based MI estimator (Kraskov et al., 2004; Gómez-Verdejo et al., 2009).

Even if feature selection for traditional classification problems has been widely studied in the literature, it is somehow surprising that the impact of label noise on this task has not been investigated yet. To our knowledge, problems with feature selection were only mentioned by Zhang et al. (2006) and Shanab et al. (2012). In the particular context of gene selection, they show that only a few mislabelled samples cause a large percentage of the most discriminative genes to be not identified and that label noise decreases the stability of feature rankings. It is quite common when working with real-world datasets that some of the class labels are wrong (Brodley and Friedl, 1999). This can be due to the fact that, for many applications, human expertise is needed to assign class labels. Moreover, some errors can be made when labels are encoded in a data set. As label noise is known to have a negative impact on the performances of supervised classification algorithms, it is reasonable to assume that it will also degrade the performances of supervised feature selection algorithms. In this case, a label noise-tolerant feature selection algorithm would undoubtedly be of great interest.

First, the impact of label noise on a traditional MI-based filter feature selection algorithm is analysed, which shows how the performances of such an algorithm can decrease when the label noise increases. A solution to make a nearest neighbours based entropy estimator less sensitive to errors in the class labels is then proposed; the solution is based on a statistical model of the label noise and an expectation-maximisation algorithm.

The rest of the paper is organised as follows. Section 2 briefly reviews basic notions about MI-based feature selection and about the label noise problem; the impact of label noise on feature selection is also illustrated. Section 3 introduces a label noise-tolerant entropy estimator, assuming the true class memberships are known. An expectation-maximisation algorithm to estimate these memberships is derived in Section 4. The complete label noise-tolerant feature selection procedure is introduced in Section 5 and its interest is experimentally illustrated in Section 6. Section 7 concludes the paper.

2. Imprecise labels and feature selection

This section reviews basic concepts about mutual information (MI)-based feature selection and methods to handle label noise. The impact of the label noise on the performances of a classical MI-based supervised feature selection algorithm is eventually illustrated in an example.

2.1. Mutual information: definitions and interest for feature selection

Filter-based feature selection requires the use of a statistical criterion, measuring the relevance of a feature set for predicting the class labels. In this work, the mutual information (MI) (Shannon, 1948) criterion is considered. Let X denote a (group of) real-valued random variable(s) on domain $\mathcal{X}$ and Y a discrete random variable on domain $\mathcal{Y}$. In a feature selection context, X is a (group of) feature(s) and Y the associated class label. The MI between X and Y is defined as

$$I(X;Y) = H(X) - H(X|Y), \tag{1}$$

where $H(X)$ is called the entropy of X. The entropy is

$$H(X) = -\int_{\mathcal{X}} p_X(x) \log p_X(x) dx, \tag{2}$$

p_X being the probability density function of X. In Eq. (1), $H(X|Y)$ is the conditional entropy of X given Y:

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p_Y(y) H(X|Y = y), \tag{3}$$

where p_Y is the probability mass function of Y. In the last equation, $H(X|Y = y)$ is the classical entropy of X, but limited to the points whose class label is y.

The MI criterion has many desirable properties for feature selection. First it has a natural interpretation in terms of uncertainty reduction. Indeed, it is symmetric and Eq. (1) can be equivalently rewritten as

$$I(X;Y) = H(Y) - H(Y|X). \tag{4}$$

Since the entropy measures the uncertainty on the observed values of a random variable, the MI can be seen as the reduction of uncertainty on the class labels once a (group of) feature(s) is known. This is obviously a sound criterion to assess the interest of a subset of features. Moreover, the MI has the advantage over other well-known criteria (such as the popular correlation coefficient, see e.g. Yu and Liu (2003)) that it is able to detect non-linear relationships between variables; it is thus more powerful in practice. Eventually, the MI can be naturally defined for multidimensional variables, which again is not the case for other popular criteria. This property can be particularly helpful for feature selection, since some features are often only relevant or redundant when considered together.

2.2. Search procedures

The objective of the feature selection method that is considered in this paper is to find the subset of the original features which together maximise the MI with the output Y. The most straightforward strategy is to try all possible feature subsets. However, such an exhaustive search is intractable in practice as the number of features gets large.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

An efficient alternative often encountered in the literature is to use greedy procedures, whose most popular ones are the forward and the backward searches (Caruana and Freitag, 1994). A forward search begins with an empty set of features. Then, at each step, the feature whose addition to the set of already selected features leads to the highest MI with the output is selected. On the contrary, a backward search starts with all features. At each step of the procedure, the feature whose removal leads to the subset with the highest MI with the output vector is eliminated.

2.3. Mutual information estimation

As can be understood from Eq. (3), the MI cannot be directly computed for real-world problems, as the probability density function (PDF) of X is not known in practice. The MI has thus to be estimated from the dataset. Traditional approaches to MI estimation first start by estimating the PDF, in order to get an approximation of the entropy of X. Eqs. (1) and (3) can then be used to estimate the MI. Estimating a PDF for one-dimensional variables is a widely studied task for which many satisfactory solutions exist, e.g. the kernel-based density estimation (Steuer et al., 2002), the B-splines approach (Daub et al., 2004) or even the basic histogram (Battiti, 1994). When the dimension of the data increases, however, these methods are likely to fail, since they are strongly affected by the curse of dimensionality (Bellman, 1961). As a consequence, they cannot be used in combination with the multivariate greedy search procedures described above.

A possible alternative is to consider nearest neighbours based entropy estimators, such as the one proposed by Kozachenko and Leonenko (1987). Indeed, such estimators are expected to be less sensitive to the dimensionality of the data (Rossi et al., 2006). For this reason, the results in Kozachenko and Leonenko (1987) have been extended to the estimation of the MI for both continuous (Kraskov et al., 2004) and categorical (Gómez-Verdejo et al., 2009) output vectors. Feature selection results obtained through these estimators are particularly encouraging (see e.g. Rossi et al., 2006). The entropy estimator proposed in Kozachenko and Leonenko (1987) is

$$\hat{H}(X) = -\psi(k) + \psi(n) + \log c_d + \frac{d}{n}\sum_{i=1}^{n} \log \epsilon_k(i), \tag{5}$$

where k, the only parameter of the estimator, is the number of nearest neighbours considered, n is the total number of samples of X, d is the dimensionality of the samples (or equivalently the number of features), $c_d = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})}$ is the volume of the unitary ball of dimension d (where Γ is the Gamma function) and $\epsilon_k(i)$ is twice the distance from the ith sample in X to its kth nearest neighbour (in terms of the Euclidean distance). Notice that $\epsilon_k(i)$ can also be interpreted as the diameter of the hypersphere containing the k-nearest neighbours of the ith sample. Eventually, ψ is the digamma function.

Combining Eqs. (1), (3) and (5), Gómez-Verdejo et al. (2009) derived the estimator

$$\begin{aligned}\hat{I}(X;Y) &= \hat{H}(X) - \sum_{y\in\mathcal{Y}} p_Y(y)\hat{H}(X|Y=y) \\ &= \psi(n) - \frac{1}{n}\sum_{y\in\mathcal{Y}} n_y\psi(n_y) + \frac{d}{n}\left[\sum_{i=1}^{n}\log\epsilon_k(i) - \sum_{y\in\mathcal{Y}}\sum_{i|y_i=y}\log\epsilon_k(i|y)\right],\end{aligned} \tag{6}$$

with n_y being the number of samples whose observed label is y and $\epsilon_k(i|y)$ being defined similarly as $\epsilon_k(i)$ but considering only the points belonging to class y. Eq. (6) assumes that $p_Y(y)$ can be adequately estimated by $\frac{n_y}{n}$.

2.4. Label noise

In the literature, three main approaches can be distinguished to deal with label noise in the context of classification problems.

First, some authors have proposed to filter the noisy data, in order to detect the wrongly labelled samples. Those samples are then removed or their label is corrected. Different criteria indicating the existence of mislabelled data points can be thought of. As a few examples, Guyon et al. (1996) considers the information gain, while Barandela and Gasca (2000) rather makes use of the labels of the neighbouring samples. Eventually, in Brodley and Friedl (1999), the authors use disagreement in ensemble methods. In these examples, the *decontamination* of the data set is made prior to any further classification step.

Second, a quite different strategy is the model-based approach, where an explicit label noise model is considered. The first work to propose such a strategy was Lawrence and Schölkopf (2001), where a probabilistic noise model is combined with a Fisher Kernel discriminant to perform binary classification. This model has been extended by getting rid of the limiting assumption of Gaussian distribution (Li et al., 2007) and adapted to multi-class problems (Bootkrajang and Kaban, 2011). Other models have also been proposed as e.g. Paulino et al. (2005) and Bouveyron and Girard (2009).

Eventually, different but related works address problems where uncertainties over labels are assumed to be readily available (Côme et al., 2008, 2009).

This work focuses on model-based approaches and develops a noise-tolerant MI estimator to achieve feature selection. Indeed, model-based approaches are theoretically sound and have the advantage of not discarding any sample containing potentially valuable information.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

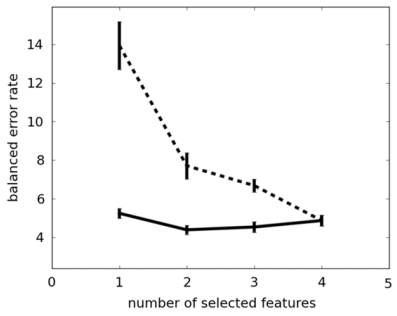


Fig. 1. Balanced classification error rate of a k-nearest neighbours classifier for the Iris dataset as a function of the number of selected features for noise-free (plain line) and noisy (dashed line) data. The error bars correspond to 95% confidence intervals.

2.5. Impact of label noise on feature selection

It is well known that label noise has a negative impact on the performances of supervised classification models. Nevertheless, there is to the best of our knowledge no evidence that label noise also degrades the performances of feature selection algorithms, except Zhang et al. (2006) which shows in the context of gene selection that only a few mislabelled samples cause a large percentage of the most discriminative genes to be not identified and Shanab et al. (2012) which shows that label noise decreases the stability of feature rankings. Notice that gene selection is particular, since there are only a few tens of training samples, what may make this application especially sensitive to label noise. The goal of this section is to show that the performances of feature selection are actually degraded by label noise and that taking label noise into account when performing feature selection can be important and useful in practice.

To this end, a greedy backward feature selection algorithm based on the MI criterion, estimated as detailed in Section 2.3, was run on the well-known Iris dataset from the UCI repository (Frank and Asuncion, 2010). The same experiment was also carried out, but after 20% of the class labels have randomly been switched to another class, chosen equiprobable among the other classes. To measure the quality of feature selection, the performances of a k-nearest neighbours (kNN) classifier are observed for each obtained feature subset, since the kNN classifier is known to be very sensitive to the presence or irrelevant features. It is important to notice that only the feature selection is made with noisy data, while the training, the validation and the prediction steps of the kNN are made using the noise-free data. This allows us to compare the results on the actual problem of interest, feature selection. The experiment was repeated 100 times and samples are split into training and test sets (70%–30%). The optimal value of the meta-parameter k was chosen using ten-fold cross-validation. All the technical details are discussed in Section 6.

Fig. 1 shows that even for the simple Iris dataset, the performances of the kNN classifier are considerably affected by the presence of label noise. Indeed, the balanced classification error rate (the average of the classification error rates obtained for each class) is largely and significantly higher in the noisy case, whatever the number of selected features is. Moreover, the stability of the feature selection is also degraded by the presence of label noise, as shown by the much larger confidence intervals.

3. Label noise-tolerant entropy estimation

As discussed in the previous section, feature selection in classification can be affected by label noise. Indeed, the Gomez MI estimator (Gómez-Verdejo et al., 2009) uses the observed labels, which may be incorrect. In turn, the incorrect MI values disrupt feature selection, which may lead to the selection of less informative feature subsets. This section proposes a label noise-tolerant entropy estimator. Since the next three sections are highly interdependent, a short introduction is given to help the reader.

3.1. Short summary of the three next sections

In this section and the two following ones, a methodology is proposed to deal with label noise for feature selection in classification. Concepts and algorithms are progressively introduced in these three sections to make developments easier to follow.

Section 3 shows that the Kozachenko–Leonenko estimator is affected by label noise. Consequently, a label noise-tolerant estimator of the entropy is proposed by relaxing the observed labels. This estimator requires the knowledge of the true memberships of each instance to the different classes, which are of course usually unknown in practice but can be estimated

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

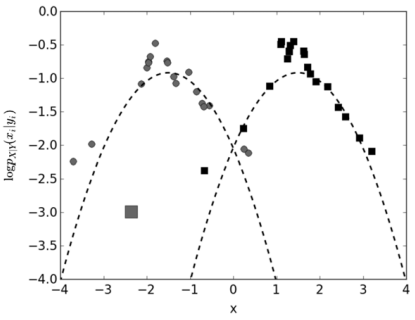


Fig. 2. Estimates of the logarithm of the conditional probability for a binary classification problem. Each class has a Gaussian distribution (dashed lines) and 40 samples are shown (grey circles belong to class 0, black squares belong to class 1); one sample is mislabelled (large grey square).

as seen in Section 4. This allows to estimate mutual information in a label noise-tolerant way to perform feature selection, as seen in Section 5.

The true memberships which are necessary in Section 3 are obtained in Section 4. A statistical model of label noise introduced in Lawrence and Schölkopf (2001) is used, whose parameters are optimised using a new expectation-maximisation algorithm. At each iteration of this EM algorithm, the class memberships are estimated using the current label noise model parameters, which are in turn updated. The resulting values can be used to evaluate the entropy estimator proposed in Section 3.

Eventually, a backward search algorithm is proposed in Section 5 to perform feature selection in the presence of label noise. This feature selection algorithm uses the results of Section 4 to estimate the class memberships and the results of Section 3 to estimate mutual information.

3.2. Effects of label noise on the Kozachenko–Leonenko estimator

As explained in Section 2, the MI between the features X and the target Y can estimated for classification problems using

$$\hat{I}(X;Y) = \hat{H}(X) - \sum_{y\in\mathcal{Y}} \hat{p}_Y(y)\hat{H}(X|Y=y) \tag{7}$$

where probabilities $\hat{p}_Y(y)$ are estimated from data. One has to estimate the entropy $H(X)$ of the features and their partial conditional entropy $H(X|Y=y)$ for each class $y \in \mathcal{Y}$. In Gómez-Verdejo et al. (2009), the Kozachenko–Leonenko estimator of entropy (5) is used to obtain the Gomez estimator of MI for classification (6), which implements Eq. (7). When instances are mislabelled, two different, yet related problems occur with both Eq. (7) and the Gomez estimator (6).

Firstly, since each term in the sum of Eq. (7) requires an empirical estimator of the form

$$\hat{H}(X|Y=y) = -\frac{1}{n_y}\sum_{i|y_i=y} \log \hat{p}_{X|Y}(x_i|y), \tag{8}$$

where the sum is taken on all instances with observed label y, the instances with incorrect labels are used to estimate the wrong quantity. For example, if x belongs to class 0 but is labelled in class 1, it is used to estimate $H(X|Y=1)$ instead of $H(X|Y=0)$. In that case, both estimates are altered.

Secondly, and consequently, the Kozachenko–Leonenko estimates of the partial conditional entropies $H(X|Y=y)$ are biased. Indeed, since the label noise results in the removal and addition of neighbours with label y, it can modify the diameter $\epsilon_k(i|y)$ of the hypersphere containing the k nearest neighbours of x_i with label y. Depending on the altered labels, the diameter can increase or decrease. Moreover, and more importantly, the diameter of the hypersphere may get large for mislabelled instances. Indeed, if x_i is incorrectly labelled in class 1 while it is surrounded by instances from its true class 0, the k nearest neighbours of x_i in class 1 will probably be far away. In such a case, the estimate of $p_{X|Y}(x_i|y)$ is almost zero and a large negative value occurs in Eq. (8).

The problem of large negative values is illustrated in Fig. 2. In that experiment, 40 samples are generated from two classes with Gaussian conditional probability distributions $\mathcal{N}(\mu_0=-1.5, \sigma_0=1)$ and $\mathcal{N}(\mu_1=1.5, \sigma_1=1)$ and identical priors $p_Y(0)=p_Y(1)=\frac{1}{2}$. Moreover, one of the samples with label 0 is incorrectly assigned the label 1. Fig. 2 shows estimates of the $\log p_{X|Y}(x_i|y_i)$ terms in Eq. (8), which are computed using the Kozachenko–Leonenko approach with $k=8$. The values reflect the conditional probability of each sample, except for the only mislabelled sample which corresponds to a large negative value. Consequently, the resulting estimate of MI is only $\hat{I}(X;Y)=0.58$. By way of comparison, the estimate for

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

a clean version of this dataset (with no mislabelling) is $\hat{I}(X;Y) = 0.63$. Hence, a single mislabelled instance can already influence the estimation of MI, even for a simple problem.

3.3. True class memberships and label noise modelling

In order to address the two above problems, it is proposed to associate each sample to a true class which may be different from the observed label. In other words, each observed label is a noisy copy of a true hidden label. For each instance, one can therefore estimate the membership $p_{S|X,Y}(s|x,y)$ of an instance x to the true class s, if the observed label is y. If there is no label noise, then one simply obtains

$$p_{S|X,Y}(s|x,y) = \begin{cases} 1 & \text{if } s = y \\ 0 & \text{if } s \neq y. \end{cases} \tag{9}$$

In other cases, it is necessary (i) to choose a model of the label noise and (ii) to estimate the most probable true class memberships, given the observed data. For example, in the situation where the classes correspond to separate clusters with no overlap, it is reasonable to assume that an instance which is obviously in the wrong cluster is mislabelled. The estimated memberships depend on the label noise model, which itself expresses hypotheses on the nature of the label noise.

The rest of this section assumes that true class memberships are available. Of course, true class memberships are usually unknown in practice. Hence, an approach is proposed to estimate these quantities in Section 4. In order to simplify mathematical notations, the following notation is introduced:

$$\gamma(s|i) = p_{S|X,Y}(s|x_i, y_i). \tag{10}$$

3.4. An entropy estimator based on the true class memberships

The Kozachenko–Leonenko estimator is based on the hypothesis that $p_{Y|X}$ remains constant in a small hypersphere with diameter $\epsilon_k(i|y)$ containing exactly the k nearest neighbours of the ith sample. Using this assumption, Kozachenko and Leonenko obtain the following estimate

$$\log \hat{p}_{X|Y}(x_i|y_i) = \psi(k) - \psi(n_y) - \log c_d - d \log \epsilon_k(i|y). \tag{11}$$

Hence, the partial conditional entropy in Eq. (8) can be estimated by

$$\hat{H}(X|Y=y) = -\psi(k) + \psi(n_y) + \log c_d + \frac{d}{n_y} \sum_{i|y_i=y} \log \epsilon_k(i|y). \tag{12}$$

However, as explained in Section 3.2, problems can occur when observed labels are polluted by label noise. Instead of $H(X|Y=y)$, one would rather prefer to estimate $H(X|S=s)$. In other words, one is interested in the entropy of X given the true class S, rather than the entropy of X given the observed label Y which is potentially incorrect. In this paper, the proposed solution consists in using the hypersphere which contains approximately an expected number of k instances really belonging to the target class s. Since true class memberships are assumed to be available, they can be used to determine the new hypersphere diameter. For each class $s \in \mathcal{Y}$ and each sample x_i, one can pick the hypersphere which contains enough neighbours of x_i so that the sum of their memberships to class s is approximately equal to k. The resulting algorithm is shown in Algorithm 1.

For each class $s \in \mathcal{Y}$ and each sample x_i, Algorithm 1 starts with the standard hypersphere with diameter $\epsilon_k(i)$ which contains the k nearest neighbours of x_i. Notice that the observed labels are not taken into account. The initial hypersphere contains an expected number

$$\sum_{j=1}^{k} \gamma(s|i_j) \tag{13}$$

of instances of the target class s, where i_j is the index of the jth neighbour of the ith sample. For correctly labelled samples, this quantity is expected to be (i) close to k for their observed class and (ii) close to zero for other classes, except near the classification boundary where the expected number of instances for each class is proportional to the class prior. For mislabelled examples, the initial hypersphere is expected to contain almost no samples of their observed class, since they stand in a region where that class should not be observed.

Thereafter, the hypersphere diameter $\epsilon_{k,\gamma}(i|s)$ is increased until the sum of the memberships

$$\Gamma(s|i) = \sum_{j=1}^{k(s|i)} \gamma(s|i_j) \tag{14}$$

becomes at least equal to k, where $k(s|i)$ is the number of actually considered neighbours. The sum $\Gamma(s|i)$ of memberships estimates the actual number of neighbours of the ith sample which really belong to class s. The resulting hypersphere with diameter $\epsilon_{k,\gamma}(i|s)$ contains an expected number of $\Gamma(s|i) \approx k$ instances of the target class s.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

Algorithm 1 Label noise-tolerant estimation of hypersphere diameters

Input: set of samples $\{x_i\}_{i\in 1\ldots n}$ and memberships $\{\gamma(s|i)\}_{i\in 1\ldots n, s\in\mathcal{Y}}$
Output: hypersphere diameters $\epsilon_{k,\gamma}(i|s)$ and memberships sums $\Gamma(s|i)$

```
for all class s ∈ 𝒴 do
  for all sample x_i do
    compute the ordering i_1 ... i_n of samples w.r.t. x_i

    k(s|i) ← k
    Γ(s|i) ← Σ_{j=1}^{k} γ(s|i_j)

    while Γ(s|i) < k do
      k(s|i) ← k(s|i) + 1
      Γ(s|i) ← Γ(s|i) + γ(s|i_{k(s|i)})
    end while

    ε_{k,γ}(i|s) ← 2 ‖x_{i_{k(s|i)}} − x_i‖_2
  end for
end for
```

The hypersphere diameter $\epsilon_{k,\gamma}(i|y)$ associated with the observed label y is expected to be much larger for mislabelled samples than for correctly labelled samples. Indeed, the hypersphere has to grow much in order to comprehend a region where samples have sufficient memberships to the class y. In contrast, the hypersphere diameter $\epsilon_{k,\gamma}(i|y)$ associated with the true class of mislabelled samples should be close to the diameter for correctly labelled samples of the same class. Hence, hypersphere diameters can be used to deal with noisy instances, as shown below.

With the new robust estimation of the hypersphere diameters, one obtains the following label noise-tolerant estimate

$$\log \hat{p}_{X|S}(x_i|s_i) = \psi\left(\Gamma(s|i)\right) - \psi\left(\Gamma(s)\right) - \log c_d - d\log\epsilon_{k,\gamma}(i|s) \tag{15}$$

where

$$\Gamma(s) = \sum_{i=1}^{n} \gamma(s|i) \tag{16}$$

can be interpreted as the expected number of samples which really belong to class s. Notice that the instance x_i itself is included in sums in both Eqs. (14) and (16), so that $0 \leq \Gamma(s|i) \leq \Gamma(s) \leq n$ for each ith instance and class s. Also, Γ is not the Gamma function used in Section 2.3 to compute the volume of the unitary ball of dimension d. In the rest of this paper, the notation Γ always refers to Eqs. (14) and (16).

Since (i) each sample x_i belongs to class s with probability $\gamma(s|i)$ and (ii) $\Gamma(s)$ is the estimated number of samples in class s, Eq. (8) becomes

$$\hat{H}(X|S=s) = -\frac{1}{\Gamma(s)}\sum_{i=1}^{n}\gamma(s|i)\log\hat{p}_{X|S}(x_i|s_i). \tag{17}$$

Using Eq. (15) together with Eq. (17), one eventually obtains the following label noise-tolerant estimate of the partial conditional entropy

$$\hat{H}(X|S=s) = -\frac{1}{\Gamma(s)}\sum_{i=1}^{n}\gamma(s|i)\psi\left(\Gamma(s|i)\right) + \psi\left(\Gamma(s)\right) + \log c_d + \frac{d}{\Gamma(s)}\sum_{i=1}^{n}\gamma(s|i)\log\epsilon_{k,\gamma}(i|s). \tag{18}$$

3.5. A label noise-tolerant estimator for mutual information

For feature selection, the actual quantity of interest is the mutual information. Using the above results, it is possible to derive a label noise-tolerant estimate of this quantity. Indeed, Eq. (18) allows us to estimate the partial conditional entropy $H(X|S=s)$ in a label noise-tolerant way for each class $s \in \mathcal{Y}$. Since the estimation of the entropy $H(X)$ is not affected by label noise, because labels are not taken into account, one can replace the Gomez estimator (6) by the following new label noise-tolerant estimator of the MI

$$\hat{I}(X;S) = \hat{H}(X) - \sum_{s\in\mathcal{S}}\hat{p}_S(s)\hat{H}(X|S=s) \tag{19}$$

where the partial conditional entropies $H(X|S=s)$ are estimated using Eq. (18) and $\hat{p}_S(s) = \Gamma(s)/n$. This new MI estimator measures the relationship between X and the true class S, whereas Eq. (7) estimates the relationship between X and the

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

observed label Y. Hence, Eq. (19) is more reliable for feature selection in the context of label noise, which is shown through experiments in Section 6.

4. True class memberships estimation

In the above developments, true class memberships are assumed to be known. However, in practice, it is seldom the case. Hence, this section proposes a new expectation maximisation (EM) algorithm to estimate the true class memberships.

4.1. Label noise modelling

In order to estimate the true class memberships, it is necessary to model the label noise. This paper uses the model introduced in Lawrence and Schölkopf (2001) which assumes that the observed label Y is a noisy copy of the true class S. Under the hypothesis that (i) an instance with true class s has a probability $p_e(s)$ to be mislabelled and that (ii) in that case the incorrect label is randomly chosen amongst the remaining labels $\mathcal{Y} \setminus \{s\}$, one obtains

$$p_{Y|S}(y|s) = \begin{cases} 1 - p_e(s) & \text{if } s = y \\ \dfrac{p_e(s)}{|\mathcal{Y}| - 1} & \text{if } s \neq y. \end{cases} \tag{20}$$

In the rest of this section, it is shown how to estimate the error probabilities $p_e(s)$ and the true class memberships through a probabilistic approach.

4.2. Derivation of an objective criterion for label noise estimation

Label noise tends to increase the estimate of the partial conditional entropy

$$\hat{H}(X|Y = y) = -\frac{1}{n_y} \sum_{i|y_i=y} \log \hat{p}_{X|Y}(x_i|y). \tag{21}$$

Indeed, as illustrated in Section 3.2, the most important effect of label noise is to introduce large negative values in the sum in Eq. (21). Eventually, this increase of the partial conditional entropies decreases the estimate of the MI.

The large negative values in Eq. (21) are due to the difficulty of standard models to explain misclassified instances. Indeed, such observations typically arise in regions where instances of the corresponding class should not exist. Consequently, misclassified instances are seen by standard methods as outliers that (i) bias the estimated class distributions whose tails get heavier and (ii) have a very small estimated conditional probability $\hat{p}_{X|Y}(x|y)$.

Label noise modelling allows increasing $\hat{p}_{X|Y}(x|y)$ for misclassified instances, since the label noise model can help to explain such noisy observations. Eventually, it reduces the decrease of the estimated MI value. It is proposed here to take advantage of this behaviour to define an objective criterion for selecting p_e. More precisely, one should seek the model which maximises the estimated MI, i.e.

$$\hat{p}_e = \arg\max_{p_e} \hat{I}(X; Y). \tag{22}$$

In such a case, the label noise model is used to reduce the uncertainty coming from the label noise itself. Using Eqs. (7) and (21), one can show that the above formulation is equivalent to

$$\hat{p}_e = \arg\max_{p_e} \sum_{i=1}^{n} \log \hat{p}_{X|Y}(x_i|y_i). \tag{23}$$

This formulation is similar to objective function considered in Lawrence and Schölkopf (2001). However, there are two main differences: (i) the justification of Eq. (23) is based on MI maximisation, whereas the formulation in Lawrence and Schölkopf (2001) is based on maximum likelihood and (ii) only the label noise model is optimised in our case, whereas a classification model is also optimised in Lawrence and Schölkopf (2001). Indeed, Lawrence and Schölkopf (2001) is about label noise-tolerant classification, whereas this paper is about MI estimation.

4.3. An expectation maximisation algorithm for label noise estimation

There exists no closed-form solution to Eq. (23). Indeed, one can write

$$\sum_{i=1}^{n} \log \hat{p}_{X|Y}(x_i|y_i) = \sum_{i=1}^{n} \log \sum_{s \in \mathcal{Y}} \hat{p}_{X,S|Y}(x_i, s|y_i) \tag{24}$$

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

where the second sum spans all possible true classes. Quantity (24) is called the incomplete log-likelihood, because the true classes are unknown. Because of the occurrence of logarithms in the sum, the function is non-convex and multiple local maxima may exist. As a closed-form expression for the global maximum of Eq. (24) does not exist, one can rather use an expectation maximisation (EM) algorithm (Dempster et al., 1977) which considers S as a latent random variable in order (i) to build successive approximations of the incomplete log-likelihood and (ii) to use them to maintain an estimate of the error probabilities $p_e(s)$. For Eq. (24), the EM algorithm alternatively (i) estimates the functional

$$Q(p_e, p_e^{\text{old}}) = \sum_{i=1}^{n} \sum_{s \in \mathcal{Y}} \gamma(s|i) \log \hat{p}_{X,S|Y}(x_i, s|y_i) \tag{25}$$

using the current estimate p_e^{old} (E step) and (ii) maximises $Q(p_e, p_e^{\text{old}})$ with respect to p_e in order to update its estimate (M step). See e.g. Sections 9.3 and 9.4 of Bishop (2007) for details about the EM algorithm and the link between Eqs. (24) and (25). At the beginning of the EM algorithm, the error probabilities $p_e(s)$ are randomly initialised and the true class priors are equal to the label priors, i.e. $p_Y(s) = \frac{n_y}{n}$.

The E step consists in estimating the true class memberships using the current label noise model, i.e.

$$\gamma(s|i) = \frac{p_{X|S}(x_i|s) p_{Y|S}(y_i|s) p_S(s)}{\sum_{s \in \mathcal{S}} p_{X|S}(x_i|s) p_{Y|S}(y_i|s) p_S(s)}. \tag{26}$$

The labelling probabilities $p_{Y|S}(y_i|s)$ are provided by Eq. (20), whereas the likelihoods $p_{X|S}(x_i|s)$ can be obtained from Eq. (15) as

$$\hat{p}_{X|S}(x_i|s_i) = \exp\big(\psi(\Gamma(s|i)) - \psi(\Gamma(s)) - \log c_d - d \log \epsilon_{k,\gamma}(i|s)\big). \tag{27}$$

During the M step, the error probabilities are updated (see Appendix) as

$$p_e(s) = \frac{1}{\Gamma(s)} \sum_{i|y_i \neq s} \gamma(s|i) \tag{28}$$

whereas the true class priors become

$$p_S(s) = \frac{1}{n} \sum_{i=1}^{n} \gamma(s|i). \tag{29}$$

Notice that EM is an iterative algorithm that may converge to local maxima. Hence, the E step and M step must be repeated until convergence and the whole EM must be repeated several times with random initial values for p_e and $p_S(s)$. The best solution is then selected by evaluating the incomplete log-likelihood with Eq. (24), which is the objective function. This quantity can also be used to stop the EM, e.g. by detecting that a (local) maximum has been reached.

4.4. Time complexity comparison of the entropy estimation algorithms

The results derived in Sections 3 and 4 allow estimating mutual information in the presence of label noise. Let us now compare the proposed method and the standard method in terms of time complexity. In order to simplify the discussion, the estimation of $H(X)$ can be ignored since it is identically performed in both methods. Similarly, both methods require an efficient data structure to sort instances, whose computational cost of creation can be ignored. In fact, the neighbours search is the more computationally demanding operation in both cases. The exact cost of the neighbour search depends on the implementation (using e.g. k-d trees introduced in Bentley, 1975; Friedman et al., 1977), but it is possible to perform a simple comparative analysis.

For the Gomez estimator (6), a kth neighbour search has to be done n times. In the proposed methodology, neighbours are searched in Algorithm 1: for each class $s \in \mathcal{Y}$ and each sample x_i, the hypersphere diameter $\epsilon_{k,\gamma}(i|s)$ is increased until $\Gamma(s|i)$ becomes at least equal to k. This process is equivalent to the kth neighbour search for a correctly labelled instance, whereas it takes in the worst case n iterations for a mislabelled instance. Hopefully, the number of mislabelled instances which fall in the *worst case* category is likely to remain small. Since Algorithm 1 is used in each E step of the EM algorithm, the average computation cost of the proposed method is $r|\mathcal{Y}|$ times larger than the cost of the Gómez estimator, where r is the number of EM iterations.

In practice, memberships stabilise after only a few iterations of the EM algorithm. For a small number of classes $|\mathcal{Y}|$ and a reasonable number of mislabelled instances, the proposed methodology is only a few times slower than the standard approach.

4.5. Illustration

Let us consider again the problem of large negative values discussed in Section 3.2 and illustrated for a simple binary classification problem in Fig. 2. With the above procedure for estimating true class memberships, Fig. 3 shows the estimates

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

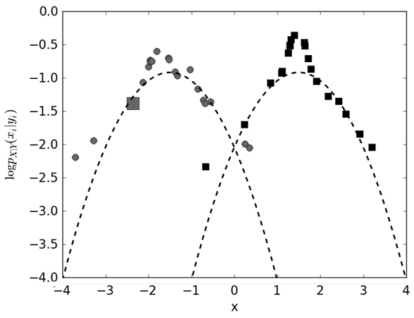


Fig. 3. Estimates of the logarithm of the conditional probability for a binary classification problem. Each class has a Gaussian distribution (dashed lines) and 40 samples are shown (grey circles belong to class 0, black squares belong to class 1), but one sample is mislabelled (large grey square).

of the terms

$$\log p_{X|Y}(x_i|y_i) = \log \sum_{s \in \mathcal{Y}} p_{X,S|Y}(x_i, s|y_i). \tag{30}$$

It can be seen that the conditional probability of the mislabelled sample has significantly increased and is now at the same level as the samples of its true class. Moreover, the estimated error probabilities are $p_e(0) = 0.02$ and $p_e(1) = 0.00$, which is close to the true error frequencies in the dataset. Eventually, the new MI estimate obtained using Eq. (19) is $\hat{I}(X;Y) = 0.61$, which is better than the MI estimate obtained by the standard estimator in Section 3.2.

5. Label noise-tolerant backward search for feature selection

In Sections 3 and 4, tools were developed for estimating MI in the presence of label noise. This section proposes a simple variant of the backward search, which is a well-known feature selection algorithm. The choice of this algorithm is motivated by the fact that backward search starts with large groups of features. In that case, the available information is higher at the beginning and both feature selection and label noise modelling should be easier. On the contrary, forward search has the drawback that it is largely influenced by the choice of the first feature. Indeed, different choices for the first feature generally lead to very different feature subsets. However, the MI between different features and the output vector is not always significantly different, which means that the choice of the first feature to be added is difficult. The backward search traditionally leads to a greater stability regarding the selected features. Notice that backward search may however not be suitable for problems with very large dimensionality, since (i) the number of necessary MI estimations grows in $\mathcal{O}(d^2)$ and (ii) the MI estimation itself becomes less reliable in such cases.

Algorithm 2 shows the proposed label noise-tolerant backward search. At each iteration, the algorithm firstly estimates a model of the label noise and, thereafter, chooses the feature to be removed. Hence, the label noise model is identical for all features within an iteration. There are two advantages to this approach, compared to estimating a new label noise model for each tested feature. Firstly, the computational load is much smaller. Indeed, $\mathcal{O}(d)$ label noise models are estimated in the former case instead of $\mathcal{O}(d^2)$ label noise models in the latter case. Secondly, the same label noise model is used to compare all features which can still be removed. Notice that $\mathcal{S}_i$ is the subset of feature indices of size $i \in 1 \ldots d$, which is obtained at the step $d - i$ of the algorithm.

6. Experiments

Let us recall the two questions raised in the introduction which are addressed in this paper: (i) what are the effects of label noise on feature selection and (ii) is it possible to reduce these effects? In Sections 2.5 and 3.2, it has been shown that label noise adversely impacts feature selection, because the MI estimation itself is affected. In Section 5, a label-noise tolerant algorithm has been derived using a new label noise-tolerant entropy estimator, which is shown in Section 4.5 to be less sensitive to label noise.

The goal of this section is not to compare the proposed feature selection method with all existing feature selection methods. Instead, this section addresses the two above questions for several real-world datasets, in order to confirm the results obtained in previous sections for simple examples. Hence, only two feature selection algorithms are compared: (i) backward search with standard MI (BW) and (ii) the proposed label noise-tolerant backward search (LNT-BW). Since both algorithms are identical, except for the evaluation of MI, it allows focusing the analysis on the two above questions about feature selection.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

Algorithm 2 Label noise-tolerant backward search for feature selection

Input: set of samples $\{x_i\}_{i\in 1\dots n}$
Output: subsets of feature indices $\{\mathcal{S}_i\}_{i\in 1\dots d}$

$\mathcal{S}_d \leftarrow \{1, \dots, d\}$

for all number of features $i \in d-1\dots 1$ **do**
  estimate true class memberships γ with the EM proposed in Section 4

  for all remaining feature with index $j \in \mathcal{S}_{i+1}$ **do**
    compute $\hat{I}_j = \hat{I}(X_{\mathcal{S}_{i+1}\setminus\{j\}}; S)$ with the label noise-tolerant estimator (19)
  end for

  $\mathcal{S}_i \leftarrow \mathcal{S}_{i+1} \setminus \left\{\arg\max_j \hat{I}_j\right\}$
end for

6.1. Experimental setting

In the following experiments, the k-nearest neighbours (kNN) classifier (Kononenko and Kukar, 2007) is used to compare selected subsets of features. Given a new sample, this classifier predicts the majority class in its k nearest neighbours. Despite its simplicity, this classifier usually achieves excellent results (Cover and Hart, 1967). Moreover, it is fast and only one meta-parameter has to be tuned: the number k of neighbours. Since the kNN classifier uses all features to compute distances in order to locate nearest neighbours, it is particularly well suited to assess the two questions recalled at the beginning of this section. Indeed, it is theoretically unable to perform any embedded feature selection and is sensitive to the presence of irrelevant features, which alter distances. If an irrelevant feature is selected, pairwise distances between instances become noisy. In turn, this distance noise causes the neighbourhood of an instance to be less appropriate to predict its class and leads to a decrease in performances.

The experiments are performed on the eleven UCI datasets (Frank and Asuncion, 2010) described in Table 1, where the last column gives the following measure of class imbalance

$$\sqrt{\frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}}\left(\frac{n_y}{n}-\frac{1}{|\mathcal{Y}|}\right)^2}, \tag{31}$$

which is inspired from the measure introduced in Pinto et al. (2009). The fourth feature of the Ecoli dataset and the third feature of the Segment dataset are constant and have been removed, since they have no predictive power and can be ignored beforehand. Eq. (31) measures the difference between the observed frequency of each class and the frequency $\frac{1}{|\mathcal{Y}|}$ which corresponds to the perfectly balanced case. Balanced datasets obtain a zero value, whereas imbalanced datasets correspond to larger values. The Glass and Yeast datasets originally contain some very small classes which are not used, since they do not even consist of enough instances to estimate the standard MI. Classes {1, 2, 7} and classes {1, 2, 3, 4} were kept for the Glass and Yeast datasets, respectively. Notice that very high-dimensional datasets are not considered in our experiments, like e.g. micro-array datasets which contain only a few tens of instances with hundreds of thousands of features. In such cases, specific feature selection methods have to be used. For example, in the micro-array case, linear models provide good results (Guyon et al., 2002; Mukherjee, 2003; Carin et al., 2004; Lee et al., 2005; Xu et al., 2009) since they are about the more complex models which can be used, because of the very small number of instances. Other methods based on correlation, statistical tests or regularisation are commonly used (Guyon et al., 2002; Li and Cheng, 2004; Helleputte and Dupont, 2009; Wang et al., 2010; Hall and Xue, in press). Investigating our method for very high-dimensional datasets is thus left for further work and could be the topic of a specific paper, due to the special characteristics of such problems.

For each dataset, all features are normalised. Then, samples are split into training and test sets (70%–30%). Training labels are then polluted by a random label noise, i.e. a given percentage of labels are flipped. For each flipped label, the new label is chosen among the other possible classes with uniform probability. The training set is used to perform feature selection in three different ways: (i) backward search with standard MI estimated on the clean labels (BW-C), (ii) backward search with standard MI estimated on the noisy labels (BW-N) and (iii) the proposed label noise-tolerant backward search (LNT-BW) with label noise-tolerant MI estimated on the noisy labels.

For each feature subset obtained by the three above feature selection algorithms, the training set with clean labels is also used to obtain a kNN classifier. This step consists in (i) selecting the optimal value of the meta-parameter k using ten-fold cross-validation and (ii) building the kNN classifier. Eventually, the performances of each kNN are assessed on the test set, which is not polluted by label noise. Both validation and test errors are measured by the balanced error rate criterion, which is the average of the percentages of misclassification for each class. This criterion avoids situations where a classifier could assign all instances to the majority class and yet obtain a good score. Indeed, some of the datasets in Table 1 are imbalanced and measures like e.g. the error rate are not appropriate, since they are relevant for balanced classification problems only.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

Table 1
Detailed list of datasets used for experiments, ordered by name.

Name	Size	Dimensionality	nb. of classes	Imbalance
Ecoli	327	6	5	0.13
Glass	175	9	3	0.12
Iris	150	4	3	0.00
Page	5473	10	5	0.35
Segment	2310	18	7	0.00
Vehicle	752	18	4	0.01
Vertebral Column	310	6	3	0.12
Wall Robot	5456	24	4	0.15
Waveform	5000	40	3	0.00
Wine	178	13	3	0.05
Yeast	1296	8	4	0.10

In order to achieve statistically significant conclusions, experiments are repeated 100 times and the dataset is shuffled before each run. The mean and standard deviation of the balanced error rate are computed. The levels of label noise are 10% and 20% of flipped labels, respectively. The parameter for the MI estimator is $k = 8$, which is standard in MI estimation (Stögbauer et al., 2004). In theory, the parameter k should be optimised using e.g. cross-validation (François et al., 2007; Verleysen et al., 2009), but such methods are computationally consuming (Gómez-Verdejo et al., 2009). This could greatly limit the interest of MI estimators in filter-based feature selection. Instead, it is advised in the MI estimation literature to use small values for k (Kraskov et al., 2004; Stögbauer et al., 2004; Rossi et al., 2006), which has been shown to give reliable MI estimates (Doquire and Verleysen, 2012). Similarly, the parameter for the label noise modelling is $k = 3$, which constrains the label noise to be estimated locally. Indeed, it was noticed during preliminary experiments that minority classes tend to be integrated in majority classes with larger values of k during the estimation of the true class memberships. The tested values for the meta-parameter k of kNN classifiers are considered incrementally between 1 neighbour and 50 neighbours, with increasing step size. Between 1 and 10 neighbours, the step size is 1, whereas it is 2 and 5 between 10 and 20 neighbours and between 20 and 50 neighbours, respectively.

Notice that the Kraskov estimator suffers from numerical problems when all neighbours of a sample stand at zero distance from it. Indeed, in that case, the terms $\log \epsilon_k(i|s)$ and $\log \epsilon_{k,\gamma}(i|s)$ are infinite in Eq. (12) and Eq. (18), respectively. This situation can occur when one selects only a few features which contain many repeated values. In order to avoid this situation, a small Gaussian noise with zero mean and standard deviation $\sigma = 10^{-3}$ is added to each feature before (and only for) the feature selection step, so that zero distances no longer occur.

6.2. Results on real datasets

Figs. 4–6 show the results for each dataset in Table 1. In each figure, 10% of the labels have been flipped to simulate label noise in the left column, whereas this percentage increases to 20% in the right column. The curves show the mean over 100 runs of the balanced error rate in terms of the feature subset size. Error bars show the 95% confidence interval for the means. The training sets used in the case of BW-C are not polluted by artificial label noise, contrarily to the training sets for BW-N and LNT-BW. Moreover, the training sets used to build kNN classifiers and the test sets are not polluted by artificial label noise.

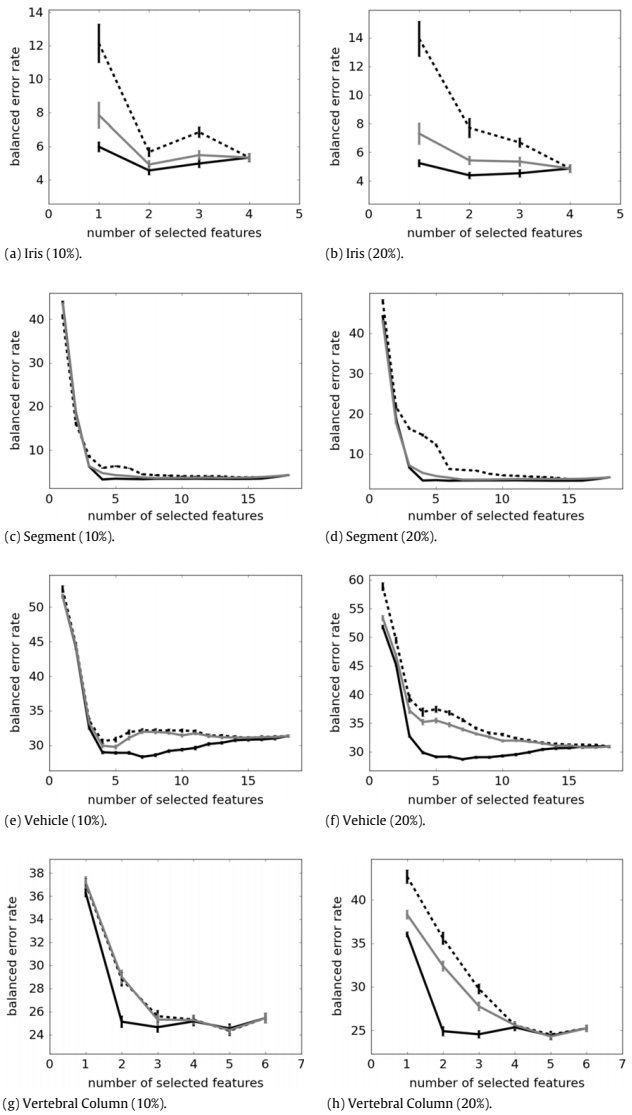
In order to facilitate the discussion of the experimental results, datasets are split into two groups: (i) Figs. 4 and 5 show those for which the proposed feature selection method produces better feature subsets than BW-N, whereas (ii) Fig. 6 shows those for which the balanced error rate is not significantly different between BW-N and LNT-BW.

6.3. Discussion

The results in Figs. 4–6 show that the label noise can have a significant impact on feature selection. Indeed, the performances obtained by the kNN classifier are almost always significantly worst when feature subsets are obtained using BW-N than with BW-C. This can be explained by the fact that the artificial label noise affects the MI estimation, which in turns leads the backward search in a wrong direction. The final consequence is a decrease of classification performances, which can be important in practice. In particular, large differences in balanced error rates between BW-C and BW-N are shown in Figs. 4(f), (h), 5(d) and (f). The only exceptions are the Ecoli and Yeast datasets for which the difference is either small or non-existent.

For the datasets illustrated in Figs. 4 and 5, the proposed feature selection method is able to improve the classification performances. When 20% of the labels are flipped, LNT-BW is always significantly better than BW-N, in terms of the balanced error rate. Notice that this is only true for intermediate feature subset sizes, i.e. $1 \ll |\mathcal{S}| \ll d$. Indeed, when $|\mathcal{S}| = d$, all features are being selected, whatever the feature selection algorithm, which means that their respective performances must be identical. This also explains that the performances of the three feature selection algorithms become more and more similar as the feature subset size tends towards d. When $|\mathcal{S}| = 1$, only one feature is selected and the amount of available

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001



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

(c) Segment (10%). (d) Segment (20%).

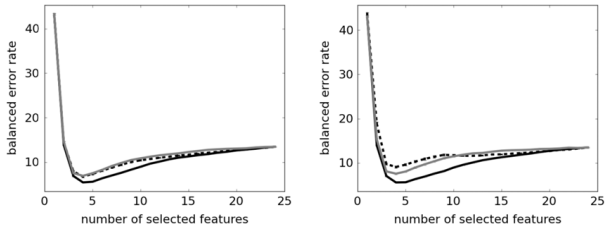
(e) Vehicle (10%). (f) Vehicle (20%).

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

Fig. 4. Results for the (a–b) Iris, (c–d) Segment, (e–f) Vehicle and (g–h) Vertebral Column datasets. Balanced error rates in percentages are shown in terms of the feature subset size for 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% of flipped labels in the left and right columns, respectively.

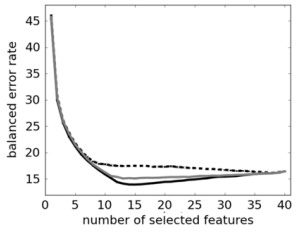
information is too low to achieve satisfying classification performances. Therefore, models achieve bad performances with any feature selection when the number of selected features becomes too small.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

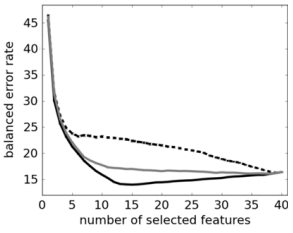


(a) Wall Robot (10%).

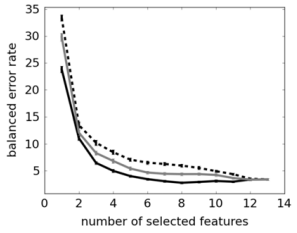
(b) Wall Robot (20%).



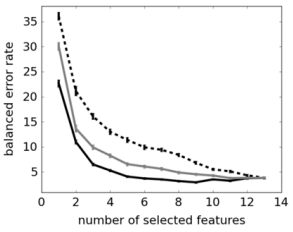
(c) Waveform (10%).



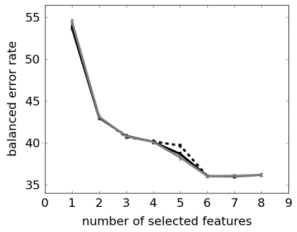
(d) Waveform (20%).



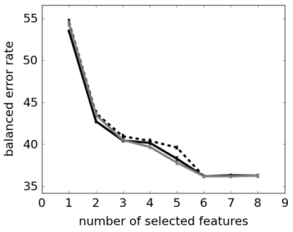
(e) Wine (10%).



(f) Wine (20%).



(g) Yeast (10%).

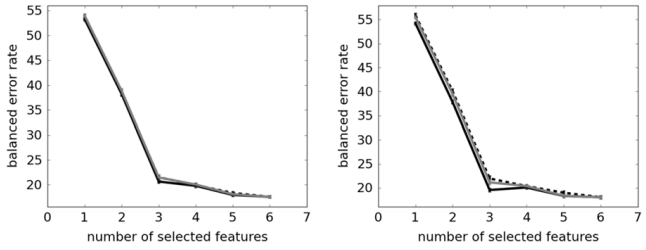


(h) Yeast (20%).

Fig. 5. Results for the (a–b) Wall Robot, (c–d) Waveform, (e–f) Wine datasets and (g–h) Yeast datasets. Balanced error rates in percentages are shown in terms of the feature subset size for 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% of flipped labels in the left and right columns, respectively.

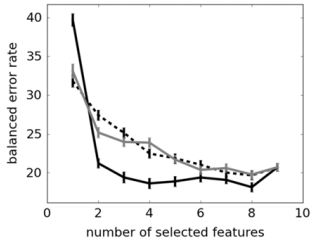
Notice that BW-C is better than LNT-BW in most cases, which means that it could be possible to further reduce the impact of label noise on feature selection. When 10% of the labels are flipped, LNT-BW remains better than BW-N, except in the case of the Vertebral Column and Wall Robot datasets.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

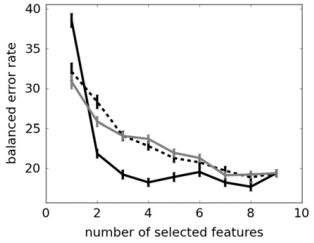


(a) Ecoli (10%).

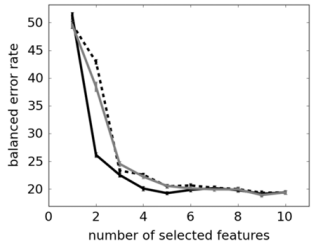
(b) Ecoli (20%).



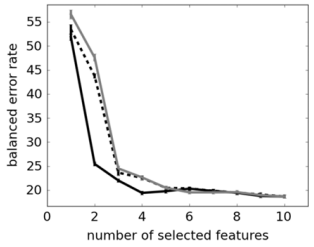
(c) Glass (10%).



(d) Glass (20%).



(e) Page (10%).



(f) Page (20%).

Fig. 6. Results for the (a–b) Ecoli, (c–d) Glass and (e–f) Page. Balanced error rates in percentages are shown in terms of the feature subset size for 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% of flipped labels in the left and right columns, respectively.

For the datasets illustrated in Fig. 6, it is impossible to distinguish the performances of the LNT-BW and BW-N methods. For the Ecoli dataset, the balanced error rates are identical for the three methods. In other words, the artificial label noise has no impact on feature selection for this dataset. For the Glass and Page datasets, both LNT-BW and BW-N are affected by label noise, but neither of them is significantly better over a large range of feature subset sizes.

6.4. Summary of results

According to the above results and discussion, it is possible to answer the two questions asked in the beginning of this experimental section. Firstly, the label noise adversely impacts feature selection. By degrading the MI estimation, it causes backward search to take wrong decisions when the MI estimate does not take label noise into account. In turn, this leads to less informative feature subsets and the classification performances are eventually decreased. The resulting decrease in performances can be important. Secondly, it has been shown that the proposed approach can improve feature selection results. In most cases, it significantly improves the classification performances. At the very worst, performances are not degraded when the method does not improve the feature selection.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

7. Conclusion

A new entropy estimator is proposed to be used in the context of label noise-tolerant feature selection. Indeed, experiments show that label noise often has a significant negative influence on the quality of selected features for MI-based algorithms, because the MI estimators themselves are altered by label noise. The proposed entropy estimator is used to derive a new label noise-tolerant MI estimator. In turn, it is shown how to use this estimator to perform label noise-tolerant feature selection. Experimental results show that the proposed label noise-tolerant feature selection algorithm obtains feature subsets which allow improving the performances of classification models.

Acknowledgements

Gauthier Doquire is funded by a Belgian F.R.I.A grant. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI) funded by the Fond de la Recherche Scientifique de Belgique (FRS-FNRS).

Appendix

Proof of the update rule for error probabilities

The update rule for error probabilities (28) can be obtained by maximising

$$Q(p_e, p_e^{\text{old}}) = \sum_{i=1}^{n} \sum_{s \in \mathcal{Y}} \gamma(s|i) \log p_{X,S|Y}(x_i, s|y_i) \tag{32}$$

with respect to $p_e(s)$ for each $s \in \mathcal{Y}$. Since

$$p_{X,S|Y}(x_i, s|y_i) = \frac{p_{X|S}(x_i|s) p_{Y|S}(y_i|s) p_S(s)}{p_Y(y_i)} \tag{33}$$

where only $p_{Y|S}(y_i|s)$ depends on $p_e(s)$, this is equivalent to maximising

$$\sum_{i=1}^{n} \sum_{s \in \mathcal{Y}} \gamma(s|i) \log p_{Y|S}(y_i|s) = \sum_{i=1}^{n} \left[\gamma(y_i|i) \log(1 - p_e(y_i)) + \sum_{s \in \mathcal{Y} \setminus \{y_i\}} \gamma(s|i) \log \frac{p_e(s)}{|\mathcal{Y}| - 1} \right]. \tag{34}$$

Setting the derivative of the above expression with respect to $p_e(s)$ to zero gives

$$-\sum_{i|y_i=s} \frac{\gamma(y_i|i)}{1 - p_e(s)} + \sum_{i|y_i \neq s} \frac{\gamma(s|i)}{p_e(s)} = 0 \tag{35}$$

which eventually gives the update rule for error probabilities

$$p_e(s) = \frac{1}{\Gamma(s)} \sum_{i|y_i \neq s} \gamma(s|i). \tag{36}$$

References

Barandela, R., Gasca, E., 2000. Decontamination of training samples for supervised pattern recognition methods. In: SSPR/SPR. In: Lecture Notes in Computer Science, vol. 1876. Springer, pp. 621–630.
Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks 5, 537–550.
Bellman, R.E., 1961. Adaptive Control Processes—A Guided Tour. Princeton University Press.
Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. Communications of the ACM 18 (9), 509–517.
Bishop, C.M., 2007. Pattern Recognition and Machine Learning, first ed. In: Information Science and Statistics, Springer.
Bootkrajang, J., Kaban, A., 2011. Multi-class classification in the presence of labelling errors. In: Proceedings of the 19th European Symposium on Artificial Neural Networks.
Bouveyron, C., Girard, S., 2009. Robust supervised classification with mixture models: learning from data with uncertain labels. Pattern Recognition 42, 2649–2658.
Brodley, C.E., Friedl, M.A., 1999. Identifying mislabeled training data. Journal of Artificial Intelligence Research 11, 131–167.
Carin, L., Krishnapuram, B., Hartemink, A., 2004. Gene expression analysis: joint feature selection and classifier design. In: Kernel Methods in Computational Biology. Bradford Books.
Caruana, R., Freitag, D., 1994. Greedy attribute selection. In: Proceedings of the Eleventh International Conference on Machine Learning. Morgan Kaufmann, pp. 28–36.
Côme, E., Oukhellou, L., Denoeux, T., Aknin, P., 2008. Mixture model estimation with soft labels. In: SMPS. In: Advances in Soft Computing, vol. 48. Springer, pp. 165–174.
Côme, E., Oukhellou, L., Denoeux, T., Aknin, P., 2009. Learning from partially supervised data using mixture models and belief functions. Pattern Recognition 42 (3), 334–348.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13 (1), 21–27.
Dash, M., Liu, H., 1997. Feature selection for classification. Intelligent Data Analysis 1, 131–156.
Daub, C., Steuer, R., Selbig, J., Kloska, S., 2004. Estimating mutual information using b-spline functions—an improved similarity measure for analysing gene expression data. BMC Bioinformatics 5 (1), 118.
Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39 (1), 1–38.
Doquire, G., Verleysen, M., 2012. A comparison of multivariate mutual information estimators for feature selection. In: ICPRAM'12. pp. 176–185.
François, D., Rossi, F., Wertz, V., Verleysen, M., 2007. Resampling methods for parameter-free and robust feature selection with mutual information. Neurocomputing 70 (7–9), 1276–1288.
Frank, A., Asuncion, A., 2010. UCI machine learning repository. http://archive.ics.uci.edu/ml.
Friedman, J.H., Bentley, J.L., Finkel, R.A., 1977. An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software 3 (3), 209–226.
Gómez-Verdejo, V., Verleysen, M., Fleury, J., 2009. Information-theoretic feature selection for functional data classification. Neurocomputing 72, 3580–3589.
Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182.
Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A., 2006. Feature Extraction: Foundations and Applications. Springer-Verlag New York, Inc.
Guyon, I., Matic, N., Vapnik, V., 1996. Discovering informative patterns and data cleaning. In: Advances in Knowledge Discovery and Data Mining. American Association for Artificial Intelligence, pp. 181–203.
Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine Learning 46 (1–3), 389–422.
Hall, P., Xue, J.-H., 2012. On selecting interacting features from high-dimensional data. Computational Statistics & Data Analysis, in press (http://dx.doi.org/10.1016/j.csda.2012.10.010).
Helleputte, T., Dupont, P., 2009. Feature selection by transfer learning with linear regularized models. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I. ECML PKDD'09. pp. 533–547.
Kononenko, I., Kukar, M., 2007. Machine Learning and Data Mining: Introduction to Principles and Algorithms. Horwood Publishing.
Kozachenko, L.F., Leonenko, N., 1987. Sample estimate of the entropy of a random vector. Problems of Information Transmission 23, 95–101.
Kraskov, A., Stögbauer, H., Grassberger, P., 2004. Estimating mutual information. Physical Review E 69 (6), 066138.
Lawrence, N.D., Schölkopf, B., 2001. Estimating a kernel Fisher discriminant in the presence of label noise. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML'01. Morgan Kaufmann Publishers Inc., pp. 306–313.
Lee, J.W., Lee, J.B., Park, M., Song, S.H., 2005. An extensive comparison of recent classification tools applied to microarray data. Computational Statistics & Data Analysis 48 (4), 869–885.
Li, C.-S., Cheng, C., 2004. Stable classification with applications to microarray data. Computational Statistics & Data Analysis 47 (3), 599–609.
Li, Y., Wessels, L.F.A., de Ridder, D., Reinders, M.J.T., 2007. Classification in the presence of class noise using a probabilistic Kernel Fisher method. Pattern Recognition 40, 3349–3357.
Mukherjee, S., 2003. Classifying microarray data using support vector machines. In: A Practical Approach to Microarray Data Analysis. Kluwer Academic Publishers.
Paulino, C.D., Silva, G., Achcar, J.A., 2005. Bayesian analysis of correlated misclassified binary data. Computational Statistics & Data Analysis 49 (4), 1120–1131.
Pinto, D., Rosso, P., Jiménez-Salazar, H., 2009. On the assessment of text corpora. In: NLDB. pp. 281–290.
Rossi, F., Lendasse, A., François, D., Wertz, V., Verleysen, M., 2006. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. Chemometrics and Intelligent Laboratory Systems 80 (2), 215–226.
Shanab, A.A., Khoshgoftaar, T.M., Wald, R., 2012. Robustness of threshold-based feature rankers with data sampling on noisy and imbalanced data. In: FLAIRS Conference.
Shannon, C.E., 1948. A mathematical theory of communication. Bell System Technical Journal 27, 379–423. 623–656.
Steuer, R., Kurths, J., Daub, C.O., Weise, J., Selbig, J., 2002. The mutual information: detecting and evaluating dependencies between variables. Bioinformatics 18 (suppl. 2), S231–S240.
Stögbauer, H., Kraskov, E., Astakhov, S.A., Grassberger, P., 2004. Least-dependent-component analysis based on mutual information. Physical Review E 70, 066123.
Verleysen, M., Rossi, F., François, D., 2009. Advances in feature selection with mutual information. In: Biehl, M., Hammer, B., Verleysen, M., Villmann, T. (Eds.), Similarity-Based Clustering. In: Lecture Notes in Computer Science, vol. 5400. Springer, Berlin, Heidelberg, pp. 52–69.
Wang, X., Park, T., Carriere, K., 2010. Variable selection via combined penalization for high-dimensional data analysis. Computational Statistics & Data Analysis 54 (10), 2230–2243.
Xu, P., Brock, G.N., Parrish, R.S., 2009. Modified linear discriminant analysis approaches for classification of high-dimensional microarray data. Computational Statistics & Data Analysis 53 (5), 1674–1687.
Yu, L., Liu, H., 2003. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: ICML. pp. 856–863.
Zhang, W., Rekaya, R., Bertrand, K., 2006. A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. Bioinformatics 22, 317–325.

Please cite this article in press as: Frénay, B., et al., Estimating mutual information for feature selection in the presence of label noise. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.001

Chapter 13

Pointwise Probability Reinforcements for Robust Statistical Inference

The following article has been submitted to the Neural Networks journal. Statistical inference using machine learning techniques may be difficult with small datasets because of abnormally frequent data (AFDs). AFDs are observations that are much more frequent in the training sample that they should be, with respect to their theoretical probability, and include e.g. outliers. Estimates of parameters tend to be biased toward models which support such data. This paper introduces pointwise probability reinforcements (PPRs): the probability of each observation is reinforced by a PPR and a regularisation allows controlling the amount of reinforcement which compensates for AFDs. This paper is related to [4]. Reprinted with permission from [6].

Title Page (With all author details listed)

Pointwise Probability Reinforcements for Robust Statistical Inference

Benoît Frénay[a,*], Michel Verleysen[a]

[a]*Machine Learning Group - ICTEAM, Université catholique de Louvain, Place du Levant 3, 1348 Louvain-la-Neuve, Belgium*

Abstract

Statistical inference using machine learning techniques may be difficult with small datasets because of abnormally frequent data (AFDs). AFDs are observations that are much more frequent in the training sample that they should be, with respect to their theoretical probability, and include e.g. outliers. Estimates of parameters tend to be biased toward models which support such data. This paper proposes to introduce pointwise probability reinforcements (PPRs): the probability of each observation is reinforced by a PPR and a regularisation allows controlling the amount of reinforcement which compensates for AFDs. The proposed solution is very generic, since it can be used to robustify any statistical inference method which can be formulated as a likelihood maximisation. Experiments show that PPRs can be easily used to tackle regression, classification and projection: models are freed from the influence of outliers. Moreover, outliers can be filtered manually since an abnormality degree is obtained for each observation.

Keywords: maximum likelihood, outliers, robust inference

1. Introduction

In statistical inference and machine learning, the goal is often to learn a model from observed data in order to predict a given quantity. In a training sample $\mathbf{x} = (x_1, \ldots, x_n)$, the n observations $x_i \in \mathcal{X}$ are typically assumed to be i.i.d. drawn from the distribution $p(x)$ of the random variable $\mathbf{X}$, whereas the model belongs to a certain parametric family with parameters $\boldsymbol{\theta} \in \Theta$. In particular, many machine learning techniques can be cast as maximum likelihood methods. In this probabilistic framework, learning of the model parameters can be achieved by maximising the data log-likelihood

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}) = \sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta}) \tag{1}$$

where $p(x_i|\boldsymbol{\theta})$ is the probability of the observation x_i under parameters $\boldsymbol{\theta}$. In order to penalise too complex models which could overfit training data, regularisation methods or Bayesian priors can also be used as a complement.

A common problem when the training sample size n is small is that some data may be much more frequent in the training sample that they should be, with respect to their theoretical probability of occurrence $p(x_i)$. These *abnormally frequent data* (AFDs) may pose a threat to statistical inference when maximum likelihood or similar methods are used. Indeed, maximising the log-likelihood corresponds to minimising the Kullback-Leibler divergence between the empirical distribution of observed data and the considered parametric distribution (Barber, 2012), in the hope that the empirical distribution is close to the real (unknown) distribution. Since the empirical probability of AFDs is much larger than their real probability, the parameter estimation is affected and biased towards parameter values which support the AFDs. For example, AFDs are well known to hurt Gaussian distribution fitting. In this paper, a method is proposed to deal with AFDs by considering that it is better to fit for instance 95% of the data well than to fit 100% of the data incorrectly. Notice that outliers are a subclass of AFDs. Indeed, outliers are observations which should theoretically never appear in a training sample, with respect to the parametric model being used (which reflect hypotheses being made about the data generating process). This includes e.g. data which are very far from the mean in Gaussian distribution fitting or data with incorrect labels in classification. Outliers are known to noticeably affect statistical inference. This paper addresses AFDs in general; experi-

*Corresponding author

Email addresses: benoit.frenay@uclouvain.be (Benoît Frénay), michel.verleysen@uclouvain.be (Michel Verleysen)

ments focus on the specific subclass of outliers.

In many applications, regularisation or Bayesian methods are used to deal with data which are not correctly described by the model, by penalising overly complex models and avoiding overfitting. However, these methods are only suited for the control of model complexity, not for the control of AFD effects. These two problems should be dealt with different methods. Hence, many approaches have been proposed to perform outlier detection (Hawkins, 1980; Beckman and Cook, 1983; Barnett and Lewis, 1994; Hodge and Austin, 2004) and anomaly detection (Chandola et al., 2009). It is well-known that many statistical inference methods are quite sensitive to outliers, like e.g. linear regression (Cook, 1979; Beckman and Cook, 1983; Hadi and Simonoff, 1993), logistic regression (Rousseeuw and Christmann, 2003) or principal component analysis (Xu and Yuille, 1995; Archambeau et al., 2006). The approach proposed in this paper relies in part on weighted log-likelihood maximisation, which is often used in the literature to reduce the impact of some of the data (Hu and Zidek, 2002). For example, there exist such algorithms for kernel ridge regression (Wen et al., 2010; Liu et al., 2011), logistic regression (Rousseeuw and Christmann, 2003) and principal component analysis (Huber, 1981; Fan et al., 2011). The main problem with these approaches is that the weights are usually obtained through heuristics. Other methods for linear regression include e.g. M-estimators (Huber, 1964), the trimmed likelihood approach (Hadi and Luceo, 1997) and least trimmed squares (Ruppert and Carroll, 1980; Rousseeuw, 1984). One of the main advantages of the method proposed in this paper is that the observation weights are automatically computed. Moreover, the method is very generic and can be applied to any inference problem which can be formulated as a likelihood maximisation.

AFDs have been widely studied in the classification literature, where labelling errors adversely impact the performances of induced classifiers (Zhu and Wu, 2004). For example, the information gain can be used to detect such AFDs (Guyon et al., 1996). Similarly to the proposed approach, it has also been proposed in the classification literature to limit the influence of each observation during inference, in order to prevent the model parameters to be biased by only a few incorrectly labelled instances. However, each method relies on a different way to limit the contribution of observations which is specific to a given model. For example, instances with large dual weights can be identified as mislabelled for support vector machines (Ganapathiraju et al., 2000), on-line learning of perceptrons can be robustified by preventing mislabelled instances to trigger updates too frequently (Kowalczyk et al., 2001) and boosting algorithms can impose an upper bound on instance weights (Domingo and Watanabe, 2000). It has also been proposed to associate each observation with a misclassification indicator variable which follows a Bernoulli model (Rekaya et al., 2001), what is closer to the contribution of this paper; the indicators can be used to identify mislabelled observations (Zhang et al., 2006; Hernandez-Lobato et al., 2011). The approach proposed in this paper has the advantage of being generic, simple to adapt to specific statistical models and not limited to classification problems.

This paper introduces pointwise probability reinforcements (PPRs), which allow the learner to deal with AFDs in a specific way. The probability of each observation is reinforced by a PPR and a regularisation allows one to control the amount of reinforcement which is awarded to compensate for AFDs. The proposed method is very generic, for it can be applied to any statistical inference method which is the solution of a maximum likelihood problem. Moreover, classical regularisation methods can still be used to further control the model complexity. Eventually, abnormality degrees are obtained, which can be e.g. used to manually screen outliers. In the literature, many outlier detection techniques exist; see e.g. Hawkins (1980); Beckman and Cook (1983); Barnett and Lewis (1994); Hodge and Austin (2004) for a survey. However, the primary goal of the method proposed in this paper is not only to detect the outliers: the aim is rather to make maximum likelihood estimates less sensitive to observations which are abnormally frequent (including outliers) in the training sample, with respect to their theoretical probability. Consequently, common machine learning techniques like linear regression, kernel ridge regression (a.k.a. least squares support vector machines), logistic regression and principal component analysis are shown in this paper to be easily robustified using the proposed approach.

This paper is organised as follows. Section 2 introduces PPRs and motivates their formulation. Section 3 proposes a generic algorithm to compute PPRs and to use them during the statistical inference of model parameters. The proposed algorithm is adapted to several supervised and unsupervised problems in Section 4. It is shown that PPRs allow one to efficiently deal with outliers and Section 5 discusses how to choose the amount of reinforcement to use. Eventually, Section 6 concludes the paper.

2. Pointwise Probability reinforcements: Definition and Concepts

As explained in Section 1, the problem with AFDs is that their empirical probability is much larger than their actual probability. As a consequence, the parameters of models inferred from data with AFDs are biased toward values which overestimate the probability of AFDs. For small training samples, this can have an important impact on the resulting model. For example, in linear regression, outliers can significantly bias the slope and the intercept of an estimated model.

In this paper, it is proposed to deal with AFDs by introducing *pointwise probability reinforcements* (PPRs) $r_i \in \Re^+$. The log-likelihood becomes

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}, \mathbf{r}) = \sum_{i=1}^{n} \log\left[p\left(x_i|\boldsymbol{\theta}\right) + r_i\right] \tag{2}$$

where each observation x_i is given a PPR r_i which acts as a reinforcement to the probability $p\left(x_i|\boldsymbol{\theta}\right)$, resulting in a *reinforced probability*. The above log-likelihood is called here the *reinforced log-likelihood*. The PPRs should remain small (or even zero), except for AFDs for which they will compensate for the difference between their large empirical probability and their small probability under a model with parameters $\boldsymbol{\theta}$. The spirit of the proposed method is similar to the one of M-estimators (Huber, 1964) and related approaches (Chen and Jain, 1994; Liano, 1996; Chuang et al., 2000). In regression, instead of minimising the sum of the squared residuals, the M-estimator approach consists in minimising another function of the residuals which is less sensitive to extreme residuals. Similarly, PPRs allows one to make maximum likelihood less sensitive to extremely small probabilities. However, there exist many different M-estimators and it is not necessarily easy to choose among them. Moreover, their use is limited to regression. On the contrary, PPRs can be used to robustify maximum likelihood methods for e.g. regression, classification or projection, as shown in Section 4. Moreover, Section 3 shows that PPRs can be easily controlled using regularisation, for example by introducing a notion of sparsity.

Equation (2) can be motivated by considering methods which are used in the literature to deal with outliers. In classification, data consists of pairs $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ where x_i is a vector of observed feature values and y_i is the observed label. Label noise occurs when a few data have incorrect labels (e.g. false positives in medical diagnosis). In such a case, Lawrence and Schölkopf (2001) introduce a labelling error probability π_e which can be used to write

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \pi_e; \mathbf{x}, \mathbf{y}) &= \sum_{i=1}^{n} \log\left[\left(1-\pi_e\right) p\left(y_i|x_i, \boldsymbol{\theta}\right)\right. \\ &\qquad \left. +\pi_e\left(1 - p\left(y_i|x_i, \boldsymbol{\theta}\right)\right)\right] \\ &= \sum_{i=1}^{n} \log\left[p\left(y_i|x_i, \boldsymbol{\theta}\right) + \frac{\pi_e}{1-2\pi_e}\right] \\ &\quad + n\log\left[1-2\pi_e\right].\end{aligned} \tag{3}$$

Since π_e is small (incorrect labels are not majority), it follows that $\log\left[1-2\pi_e\right] \approx 0$ and the log-likelihood (3) can be approximated with (2).

Another possibility to deal with outliers (Aitkin and Wilson, 1980; Eskin, 2000) consists in assuming that data actually come from a mixture of two processes: the actual process of interest and a garbage process generating outliers. The log-likelihood becomes

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\theta}, \pi_g, \boldsymbol{\theta}_g; \mathbf{x}\right) &= \sum_{i=1}^{n} \log\left[\left(1-\pi_g\right) p\left(x_i|\boldsymbol{\theta}\right) + \pi_g p\left(x_i|\boldsymbol{\theta}_g\right)\right] \\ &= \sum_{i=1}^{n} \log\left[p\left(x_i|\boldsymbol{\theta}\right) + \frac{\pi_g}{1-\pi_g} p\left(x_i|\boldsymbol{\theta}_g\right)\right] \\ &\quad + n\log\left[1-\pi_g\right]\end{aligned} \tag{4}$$

where π_g is the prior probability of garbage patterns and $p\left(x|\boldsymbol{\theta}_g\right)$ is the probability of a garbage pattern x. Since garbage patterns are assumed to be in minority, it follows that π_g is small and that $\log\left[1-\pi_g\right] \approx 0$. Therefore, the above log-likelihood can be also approximated with (2).

It should be stressed that PPRs are not probabilities. Indeed, they can take any positive value, even if they should normally remain small. PPRs are intended to reinforce the probability $p\left(x_i|\boldsymbol{\theta}\right)$ of AFDs whose empirical probability is too large with respect to their true probability $p(x_i)$. This way, the probability $p\left(x_i|\boldsymbol{\theta}\right)$ can remain small for AFDs, what is natural since their true probability is also small. Using PPRs, maximum likelihood parameter estimation is expected to be less sensitive to AFDs and to provide more accurate parameter estimates. The advantage of using PPRs over the two above approaches discussed in Lawrence and Schölkopf (2001); Aitkin and Wilson (1980); Eskin (2000) is that it is no longer necessary to modify the data generation model. In other words, AFDs do not have to fit into the parametric model which is learnt for prediction. Indeed, a non-parametric method is proposed in Section 3 to compute PPRs using regularisation. Of course, there

is no such thing as a free lunch and it is necessary to control the amount of reinforcement which is given. This point is further discussed in details in Section 5.

3. Statistical Inference with Pointwise Probability Reinforcements

This section shows how to perform maximum likelihood statistical inference with PPRs using a generic two-step iterative algorithm. This algorithm is adapted to supervised and unsupervised problems in Section 4.

3.1. Non-Parametric Pointwise Probability Reinforcements

Without restriction on the PPRs, the reinforced log-likelihood (2) is unbounded. Indeed, one can simply choose large PPR values and obtain an arbitrary large reinforced log-likelihood, whatever the choice of the model parameters $\boldsymbol{\theta}$. A first solution to this problem is to assume a parametric form $r_i = r(x_i|\boldsymbol{\theta}_r)$ for the PPRs, where $\boldsymbol{\theta}_r$ are fixed parameters. However, this solution requires prior knowledge on the reinforcement $r(x_i|\boldsymbol{\theta}_r)$ which is necessary for a given observation x_i. This may depend on the problem which is addressed and the model which is used to solve it. Such prior knowledge is not necessarily available and it is not trivial to define meaningful distributions for PPRs.

In this paper, it is rather proposed to use a regularisation scheme to control the PPRs. In such a case, for a given family of models indexed by parameters $\boldsymbol{\theta} \in \Theta$, the parameter estimation consists in maximising

$$\mathcal{L}_\Omega(\boldsymbol{\theta}; \mathbf{x}, \mathbf{r}) = \sum_{i=1}^{n} \log\left[p(x_i|\boldsymbol{\theta}) + r_i\right] - \alpha\Omega(\mathbf{r}) \quad (5)$$

where $\mathbf{r}$ is the vector or PPRs, α is a reinforcement meta-parameter and Ω is a penalisation function. The meta-parameter α controls the compromise between (i) fitting data using the model (large α values) and (ii) using large reinforcements to deal with data as if they were AFDs (small α values).

Using regularisation to control PPRs has several advantages: this approach remains very generic and almost no prior knowledge is required. As shown in Section 3.3, the choice of the penalisation function Ω determines the properties of the vector of PPRs, like e.g. its sparseness. Hence, it is only necessary to specify e.g. if data are expected to contain only a few *strongly* AFDs or if a lot of *weakly* AFDs are expected. This paper shows that existing statistical inference methods in machine learning can be easily adapted to use PPRs.

3.2. Generic Algorithm for Using Pointwise Probability Reinforcements

In the regularised reinforced log-likelihood (5), the penalisation function Ω only depends on the PPRs in order to avoid overfitting, which could occur if Ω was also depending on the probabilities $p(x_i|\boldsymbol{\theta})$. It also allows one to separate the optimisation of (5) in two independent steps. During the first step, the model parameters $\boldsymbol{\theta}$ are fixed to $\boldsymbol{\theta}^{\text{old}}$ and (5) is maximised only with respect to the PPRs. If Ω can be written as a sum of independent terms

$$\Omega(\mathbf{r}) = \sum_{i=1}^{n} \Omega(r_i), \quad (6)$$

then each PPR can be optimised independently. The above condition on Ω is assumed to hold in the rest of the paper. During the second step, the PPRs are kept fixed and (5) is maximised only with respect to the model parameters $\boldsymbol{\theta}$. Since the penalisation function does not depend on the parametric probabilities, the regularisation term has no influence. The second step only works with reinforced probabilities and simply becomes

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log\left[p(x_i|\boldsymbol{\theta}) + r_i\right]. \quad (7)$$

The two steps are further detailed in the two following subsections. Before starting the above alternate optimisation procedure, $\boldsymbol{\theta}$ has to be initialised. This can be achieved by using a method maximising the classical log-likelihood (1): the parameter estimate will be sensitive to AFDs, yet it will provide a satisfying starting point. Notice that it is important to choose a suitable value of the reinforcement meta-parameter α. A solution to this question is proposed in Section 5.

3.3. Optimisation of Pointwise Probability Reinforcements

The expression and properties of the PPRs obtained during the first step only depend on the form of the penalisation function Ω (and not on the parametric model). Indeed, the probabilities $p(x_i|\boldsymbol{\theta})$ are treated as fixed quantities $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ and Ω is assumed to be independent of the model parameters. In this section, L_1 and L_2 PPR regularisations are considered, which lead to sparse and non-sparse PPRs, respectively. The properties of PPRs and reinforced probabilities are also considered for a general class of penalisation functions.

3.3.1. Sparse Pointwise Probability Reinforcements using L_1 Regularisation

In linear regression, it is well known that a L_1 regularisation on the weights can be used to obtain sparse weight vectors (Efron et al., 2004). Similarly, one can regularise PPRs using the penalisation function

$$\Omega(\mathbf{r}) = \sum_{i=1}^{n} r_i \tag{8}$$

which forces PPRs to shrink towards zero (remember that $r_i \geq 0$). The maximisation of the regularised reinforced log-likelihood leads to the Lagrangian

$$\sum_{i=1}^{n} \log\left[p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i\right] - \alpha \sum_{i=1}^{n} r_i + \sum_{i=1}^{n} \beta_i r_i, \tag{9}$$

what gives the optimality condition

$$\frac{1}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i} - \alpha + \beta_i = 0 \tag{10}$$

for each PPR r_i. When $r_i > 0$, the Lagrange multiplier β_i becomes zero and

$$r_i = \frac{1}{\alpha} - p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right). \tag{11}$$

Otherwise, when $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) > \frac{1}{\alpha}$, β_i has to be non-zero, what causes r_i to become zero. Hence, the PPR for the ith instance is

$$r_i = \max\left(\frac{1}{\alpha} - p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right), 0\right), \tag{12}$$

whereas the corresponding reinforced probability is

$$p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i = \max\left(p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right), \frac{1}{\alpha}\right). \tag{13}$$

Fig. 1 shows the PPR and reinforced probability in terms of the probability for different values of the reinforcement meta-parameter α. As long as $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ remains small, the PPR is approximately equal to $\frac{1}{\alpha}$ and the reinforced probability is exactly equal to $\frac{1}{\alpha}$. Such observations are considered as potential AFDs. However, as soon as $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) \geq \frac{1}{\alpha}$, the PPR becomes zero and the reinforced probability becomes exactly equal to $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$. In such a case, the observation is no longer considered as an AFD. Interestingly, for fixed parameters $\boldsymbol{\theta}^{\text{old}}$, using a L_1 PPR regularisation is equivalent to clipping the penalised probabilities which are below $\frac{1}{\alpha}$.

In conclusion, using a L_1 regularisation leads to a very simple optimisation step for the PPRs. Moreover, the resulting PPRs are sparse and only a few of them are non-zero. Indeed, only the observations for which the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ is smaller than $\frac{1}{\alpha}$ are considered as potential AFDs and reinforced accordingly. The inverse of the reinforcement meta-parameter α corresponds to the threshold applied to the reinforced probabilities.

3.3.2. Smooth Non-Sparse Pointwise Probability Reinforcements using L_2 Regularisation

A drawback of L_1 regularisation is that discontinuities may occur during the optimisation: probabilities which are reinforced at a given iteration may become unreinforced at the next iteration. Similarly to linear regression, the L_2 regularisation provides similar but smoother solutions than the L_1 regularisation. In that case, the penalisation function is

$$\Omega(\mathbf{r}) = \frac{1}{2} \sum_{i=1}^{n} r_i^2 \tag{14}$$

and has the advantage of having a zero derivative with respect to r_i when $r_i = 0$, what leads to a smoother solution. The maximisation of the regularised reinforced log-likelihood leads to the Lagrangian

$$\sum_{i=1}^{n} \log\left[p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i\right] - \frac{\alpha}{2} \sum_{i=1}^{n} r_i^2 + \sum_{i=1}^{n} \beta_i r_i, \tag{15}$$

what gives for each PPR r_i the optimality condition

$$\frac{1}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i} - \alpha r_i + \beta_i = 0. \tag{16}$$

Since $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) \geq 0$ and $\beta_i \geq 0$, it is impossible to have $r_i = 0$ and the Lagrange multiplier β_i is therefore always zero. Hence, the PPR is

$$r_i = \frac{-p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + \sqrt{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)^2 + \frac{4}{\alpha}}}{2}, \tag{17}$$

whereas the the corresponding reinforced probability is

$$p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i = \frac{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + \sqrt{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)^2 + \frac{4}{\alpha}}}{2}. \tag{18}$$

Fig. 2 shows the PPR and reinforced probability in terms of the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ for different values of the reinforcement meta-parameter α. As long as $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ remains small, the PPR and the reinforced probability are approximately equal to $\sqrt{\alpha^{-1}}$. However, as soon as $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ gets close to $\sqrt{\alpha^{-1}}$, the PPR starts decreasing towards zero and the reinforced probability

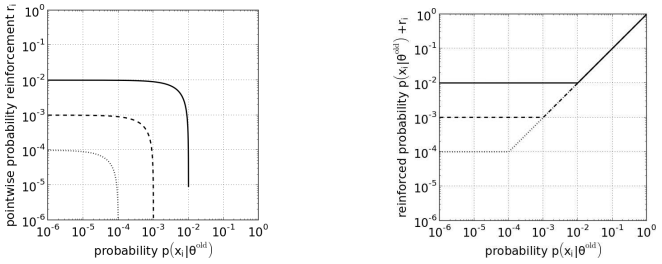


Figure 1: PPR r_i (left) and reinforced probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i$ (right) in terms of the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ obtained using L_1 regularisation on PPRs. The reinforcement meta-parameter is $\alpha = 10^2$ (plain line), $\alpha = 10^3$ (dashed line) and $\alpha = 10^4$ (dotted line).

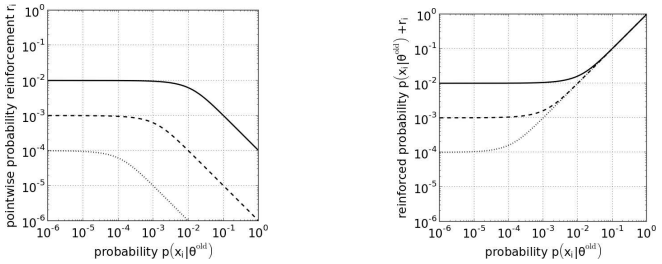


Figure 2: PPR r_i (left) and reinforced probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i$ (right) in terms of the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ obtained using L_2 regularisation on PPRs. The reinforcement meta-parameter is $\alpha = 10^4$ (plain line), $\alpha = 10^6$ (dashed line) and $\alpha = 10^8$ (dotted line).

tends to $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$. A comparison with Fig. 1 shows that L_2 regularisation is similar to L_1 regularisation, with smoother results and a threshold $\sqrt{\alpha^{-1}}$ instead of $\frac{1}{\alpha}$.

In conclusion, using L_2 regularisation seems to give similar results than L_1 regularisation, except that the PPRs cannot be sparse and that all observations are considered as AFDs to a certain degree. Also, the PPRs and the reinforced probabilities change more smoothly when $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ increases.

3.3.3. Properties of Pointwise Probability Reinforcements and Reinforced Probabilities

The above derivations show that simple expression for PPRs can be obtained using L_1 or L_2 penalisation. General results can be obtained for PPRs under reasonable requirements, e.g. that Ω is an increasing convex penalisation function which can be written as

$$\Omega(\mathbf{r}) = \sum_{i=1}^{n} \Omega(r_i) \tag{19}$$

and that the quantity $\log\left[p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i\right] - \alpha\Omega(r_i)$ achieves a maximum value for a finite PPR $r_i \geq 0$ (which means that the PPR optimisation admits a solution).

In the following theorems, penalisation functions are assumed to comply with the above requirements. Also, the notation $p_i = p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ is used in order to make the developments easier to follow.

Theorem 1. *Let Ω be an increasing penalisation function of the form* (19)*, $\boldsymbol{\theta}^{old}$ be a fixed parameter value and r_1 and r_2 be the finite optimal PPRs with respect to $\boldsymbol{\theta}^{old}$ for the observations x_1 and x_2, respectively. Then one has $r_2 \leq r_1$ if the probabilities satisfy $p\left(x_2|\boldsymbol{\theta}^{old}\right) > p\left(x_1|\boldsymbol{\theta}^{old}\right)$.*

Proof. Let us prove the theorem by showing that any PPR $r > r_1$ is suboptimal for the observation x_2. Since $p_2 > p_1$, it follows that $p_2 + r > p_2 + r_1 > p_1 + r_1$ and that $p_2 + r > p_1 + r > p_1 + r_1$. Because the logarithm is a strictly concave function, one therefore obtains the

inequalities

$$\log[p_2 + r_1] > \frac{\log[p_2 + r] - \log[p_1 + r_1]}{(p_2 + r) - (p_1 + r_1)}(p_2 - p_1) + \log[p_1 + r_1] \quad (20)$$

and

$$\log[p_1 + r] > \frac{\log[p_2 + r] - \log[p_1 + r_1]}{(p_2 + r) - (p_1 + r_1)}(r - r_1) + \log[p_1 + r_1]. \quad (21)$$

Since r_1 is the optimal PPR for p_1, it also follows that

$$\log[p_1 + r_1] - \alpha\Omega(r_1) \geq \log[p_1 + r] - \alpha\Omega(r) \quad (22)$$

for any PPR r. Summing (20), (21) and (22) eventually gives

$$\log[p_2 + r_1] - \alpha\Omega(r_1) > \log[p_2 + r] - \alpha\Omega(r), \quad (23)$$

which means that any PPR $r > r_1$ is necessarily suboptimal with respect to the probability p_2, since the PPR r_2 satisfies by definition

$$\log[p_2 + r_2] - \alpha\Omega(r_2) \geq \log[p_2 + r_1] - \alpha\Omega(r_1). \quad (24)$$

□

The above theorem means that data which are more probable with respect to the parametric model are going to be less reinforced, what seems natural. Indeed, reinforcements are supposed to support unlikely observations. Notice that if there is an observation with a zero reinforcement, the reinforcements for observations with larger probabilities are also zero.

Theorem 2. *Let Ω be an increasing convex penalisation function of the form* (19)*, $\boldsymbol{\theta}^{old}$ be a fixed parameter value and r_1 and r_2 be the finite optimal PPRs with respect to $\boldsymbol{\theta}^{old}$ for the observations x_1 and x_2, respectively. Then the reinforced probabilities are such that $p\left(x_2|\boldsymbol{\theta}^{old}\right) + r_2 \geq p\left(x_1|\boldsymbol{\theta}^{old}\right) + r_1$ if the probabilities satisfy $p\left(x_2|\boldsymbol{\theta}^{old}\right) > p\left(x_1|\boldsymbol{\theta}^{old}\right)$.*

Proof. Let us prove the theorem by considering two cases: $r_1 - (p_2 - p_1) \leq 0$ and $r_1 - (p_2 - p_1) > 0$. In the first case, since r_2 must be positive, it necessarily follows that $r_1 - (p_2 - p_1) \leq r_2$ or, by rearranging terms, that $p_2 + r_2 \geq p_1 + r_1$. In the second case, it follows from the condition $p_2 > p_1$ that $r_1 > 0$. Since r_1 is the optimal PPR for p_1, this implies that the derivative of $\log[p_1 + r] - \alpha\Omega(r)$ with respect to r is zero at $r = r_1$, i.e.

$$\frac{1}{p_1 + r_1} - \alpha\Omega'(r_1) = 0. \quad (25)$$

Moreover, the derivative of $\log[p_2 + r] - \alpha\Omega(r_2)$ at $r = r_1 - (p_2 - p_1)$ is

$$\frac{1}{p_1 + r_1} - \alpha\Omega'(r_1 - (p_2 - p_1)) \quad (26)$$

and, since $r_1 - (p_2 - p_1) < r_1$ and Ω is a convex function, it also follows that

$$\Omega'(r_1 - (p_2 - p_1)) \leq \Omega'(r_1). \quad (27)$$

Using the three above results, one can show that

$$\frac{1}{p_1 + r_1} - \alpha\Omega'(r_1 - (p_2 - p_1)) \geq 0, \quad (28)$$

i.e. that the derivative of $\log[p_2 + r] - \alpha\Omega(r)$ at $r_1 - (p_2 - p_1)$ is positive. Since this function is strictly concave in terms of r, it has only one maximum and the optimal PPR r_2 must therefore be larger or equal to this value, i.e. $r_2 \geq r_1 - (p_2 - p_1)$ or, by rearranging terms, $p_2 + r_2 \geq p_1 + r_1$. □

The above theorem means that observations which are more probable with respect to the parametric model also correspond to larger reinforced probabilities. All other things being equal, the opposite would mean that the ordering of observations with respect to their parameterised and reinforced probabilities could be different, what seems counter-intuitive.

To illustrate the above results, let us again consider L_1 and L_2 regularisation, which use increasing and convex penalisation functions. It can be seen in Fig. 1 and 2 that the resulting PPRs and reinforced probabilities behave according to Theorems 1 and 2. A simple counter-example is $L_{\frac{1}{2}}$ regularisation, where the increasing but concave penalisation function is $\Omega(\mathbf{r}) = 2\sum_{i=1}^{n}\sqrt{r_i}$. Fig. 3 shows the PPR and reinforced probability in terms of the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ for different values of the reinforcement meta-parameter α. In particular, for small values of $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$, the reinforced probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i$ decreases when the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ increases, what is a rather counter-intuitive behaviour. The PPR and the reinforced probability present a discontinuity when $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) \approx 0.65/4\alpha^2$.

3.4. Learning Model Parameters with Pointwise Probability Reinforcements

The reinforced log-likelihood may be hard to maximise with respect to the model parameters. For example, if a member of the exponential family

$$p(x_i|\boldsymbol{\theta}) = h(x_i)\exp\left[\boldsymbol{\eta}(\boldsymbol{\theta})^T\mathbf{T}(x_i) - \psi(\boldsymbol{\theta})\right] \quad (29)$$

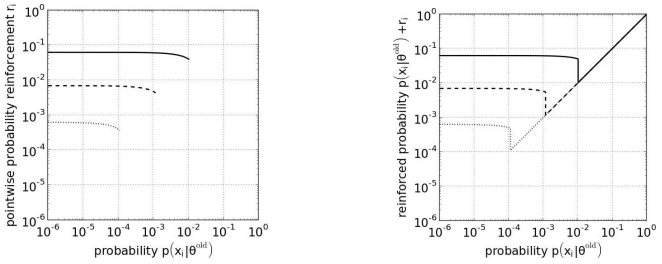


Figure 3: PPR r_i (left) and reinforced probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i$ (right) in terms of the probability $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ obtained using $L_{\frac{1}{2}}$ regularisation on PPRs. The reinforcement meta-parameter is $\alpha = 4$ (plain line), $\alpha = 12$ (dashed line) and $\alpha = 40$ (dotted line). Discontinuities occur at $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) \approx 10^{-4}$, $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) \approx 10^{-3}$ and $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) \approx 10^{-2}$, respectively.

is used to model data (what includes e.g. Gaussian, exponential, gamma, beta or Poisson distributions (Duda and Hart, 1973; Bishop, 2006; Bernardo and Smith, 2007; DasGupta, 2011)), the optimality condition becomes

$$\sum_{i=1}^{n} \frac{p\left(x_i|\boldsymbol{\theta}^{\text{new}}\right)}{p\left(x_i|\boldsymbol{\theta}^{\text{new}}\right) + r_i} \left[\boldsymbol{\eta}'\left(\boldsymbol{\theta}^{\text{new}}\right)^T \mathbf{T}(x_i) - \psi'(\boldsymbol{\theta}^{\text{new}})\right] = 0 \tag{30}$$

where h, $\boldsymbol{\eta}$, $\mathbf{T}$ and ψ depend on the distribution. Obviously, it will in general not be trivial to find a solution satisfying the above condition. For example, in the particular case of a univariate Gaussian distribution with unknown mean μ and known width σ, one can e.g. obtain the parameterisation (DasGupta, 2011)

$$h(x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x_i^2}{2\sigma^2}}; \qquad \boldsymbol{\eta}(\boldsymbol{\theta}) = \left(\tfrac{\mu}{\sigma^2}\right); \tag{31}$$

$$\mathbf{T}(x_i) = \left(x_i\right); \qquad \psi(\boldsymbol{\theta}) = \frac{\mu^2}{2\sigma^2} \tag{32}$$

and eventually obtain the optimality condition

$$\sum_{i=1}^{n} \frac{\mathcal{N}\left(x_i|\mu^{\text{new}}, \sigma\right)}{\mathcal{N}\left(x_i|\mu^{\text{new}}, \sigma\right) + r_i} \left(x_i - \mu^{\text{new}}\right) = 0 \tag{33}$$

where μ^{new} cannot be isolated.

Rather than directly optimising the reinforced log-likelihood, this paper proposes to indirectly maximise it by iteratively (i) finding a close lower bound to the reinforced log-likelihood and (ii) maximising this bound with respect to $\boldsymbol{\theta}$. This procedure is similar to the widely used expectation maximisation (EM) algorithm (Dempster et al., 1977). The following theorem shows that it is easily possible to find such a lower bound for any parametric model.

Theorem 3. *Let $\boldsymbol{\theta}^{old}$ be the current estimate of the model parameters and, for each observation x_i, let r_i be the optimal PPR with respect to $\boldsymbol{\theta}^{old}$. If one defines the observation weight*

$$w_i = \frac{p\left(x_i|\boldsymbol{\theta}^{old}\right)}{p\left(x_i|\boldsymbol{\theta}^{old}\right) + r_i}, \tag{34}$$

then the functional

$$\sum_{i=1}^{n} \left[w_i \log \frac{p\left(x_i|\boldsymbol{\theta}\right)}{p\left(x_i|\boldsymbol{\theta}^{old}\right)} + \log\left[p\left(x_i|\boldsymbol{\theta}^{old}\right) + r_i\right]\right] \tag{35}$$

is a lower bound to the reinforced log-likelihood

$$\sum_{i=1}^{n} \log\left[p\left(x_i|\boldsymbol{\theta}\right) + r_i\right]. \tag{36}$$

Moreover, (36) *and* (35) *are tangent at $\boldsymbol{\theta} = \boldsymbol{\theta}^{old}$.*

Proof. When $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{old}}$, the value of (35) and (36) is

$$\sum_{i=1}^{n} \log\left[p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i\right], \tag{37}$$

whereas their derivative with respect to model parameters is

$$\sum_{i=1}^{n} \frac{1}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i} \frac{\delta p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)}{\delta \boldsymbol{\theta}}. \tag{38}$$

Since their value and derivative are identical in $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{old}}$, it follows that (35) and (36) are tangent at that point. Let us now prove that (35) is a lower bound to (36) by considering their terms. Indeed, if each ith term of (36)

is lower bounded by the ith term of (35), (35) is necessarily a lower bound to (36). Let us first rewrite the inequality

$$\log\left[p\left(x_i|\boldsymbol{\theta}\right)+r_i\right] \geq w_i \log\frac{p\left(x_i|\boldsymbol{\theta}\right)}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)}+\log\left[p\left(\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i\right]\right. \tag{39}$$

as

$$\left[p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i\right]\log\left[\frac{p\left(x_i|\boldsymbol{\theta}\right)+r_i}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i}\right] \geq p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)\log\frac{p\left(x_i|\boldsymbol{\theta}\right)}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)}. \tag{40}$$

When $r_i = 0$, it is easily shown that both sides of the inequality are equal, what is natural since $w_i = 1$ in such a case. Hence, since r_i is always positive, it is sufficient to show that the derivative of the left side with respect to r_i is always larger than the derivative of the right side to ensure that the inequality (39) is verified for any $r_i \geq 0$. This condition can be written as

$$\log\left[\frac{p\left(x_i|\boldsymbol{\theta}\right)+r_i}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i}\right]+\frac{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i}{p\left(x_i|\boldsymbol{\theta}\right)+r_i}-1 \geq 0. \tag{41}$$

Using the standard logarithm inequality $\log x \geq \frac{x-1}{x}$, one can show that

$$\log\left[\frac{p\left(x_i|\boldsymbol{\theta}\right)+r_i}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i}\right] \geq \frac{p\left(x_i|\boldsymbol{\theta}\right)-p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)}{p\left(x_i|\boldsymbol{\theta}\right)+r_i} \tag{42}$$

and it follows that

$$\log\left[\frac{p\left(x_i|\boldsymbol{\theta}\right)+r_i}{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i}\right]+\frac{p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)+r_i}{p\left(x_i|\boldsymbol{\theta}\right)+r_i}-1 \geq \frac{p\left(x_i|\boldsymbol{\theta}\right)+r_i}{p\left(x_i|\boldsymbol{\theta}\right)+r_i}-1=0, \tag{43}$$

what proves the inequality (41) and concludes the proof. □

Based on the above theorem, a maximisation step can easily be found. Indeed, since (35) is a lower bound to the reinforced log-likelihood and both are tangent at $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{old}}$, maximising the former will necessarily increases the latter with respect to its value in $\boldsymbol{\theta}^{\text{old}}$. Hence, the approximate maximisation step (with respect to the model parameters $\boldsymbol{\theta}$) of the proposed algorithm consists in maximising the weighted log-likelihood

$$\mathcal{L}_{\mathbf{w}}\left(\boldsymbol{\theta};\mathbf{x}\right) = \sum_{i=1}^{n} w_i \log p\left(x_i|\boldsymbol{\theta}\right) \tag{44}$$

where weights are computed using (34) and the current estimate of the model parameters $\boldsymbol{\theta}^{\text{old}}$. The advantage of this maximisation step is that weighted log-likelihoods are typically much easier to maximise than reinforced log-likelihoods. For example, in the case of the above Gaussian distribution with known width σ, one obtains the optimality condition

$$\sum_{i=1}^{n} w_i\left(x_i-\mu^{\text{new}}\right)=0 \tag{45}$$

where

$$w_i = \frac{\mathcal{N}\left(x_i|\mu^{\text{old}},\sigma\right)}{\mathcal{N}\left(x_i|\mu^{\text{old}},\sigma\right)+r_i}. \tag{46}$$

Hence, the mean of the Gaussian is estimated by the weighed sample mean

$$\mu^{\text{new}} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}. \tag{47}$$

Interestingly, the weighted log-likelihood is often used in the literature to reduce the impact of some of the data (Hu and Zidek, 2002). Using L_1 regularisation, the proposed approach is similar to the trimmed likelihood approach (Hadi and Luceo, 1997), where only a subset of the observations are used to compute the log-likelihood.

As discussed in Section 3.2, the maximisation step does not depend on the penalisation function Ω. Fig. 4 shows examples of lower-bounded reinforced log-likelihoods for 50 observations drawn from a univariate Gaussian distribution with mean $\mu = 0$ and width $\sigma = 1$. The PPRs r_i are computed using the current estimates $\mu^{\text{old}} = 0.2$ and $\sigma^{\text{old}} = 1.3$. L_1 regularisation is used in Fig. 4(a) and 4(b), whereas L_2 regularisation is used in Fig. 4(c) and 4(d). The log-likelihoods are computed for different values of the mean μ in Fig. 4(a) and 4(c) and different values of the width σ in Fig. 4(b) and 4(d). Reinforced log-likelihoods and lower bounds are tangent at $\mu = \mu^{\text{old}}$ and $\sigma = \sigma^{\text{old}}$, in accordance with Theorem 3.

Notice that at optimum, the solution $\boldsymbol{\theta}^*$ of existing weighted maximum likelihood approaches can be seen as reinforced maximum likelihood solutions where the equivalent PPR for the ith observation would be

$$r_i = \frac{1-w_i}{w_i} p\left(x_i|\boldsymbol{\theta}^*\right), \tag{48}$$

which follows from the inversion of (34) defining weights.

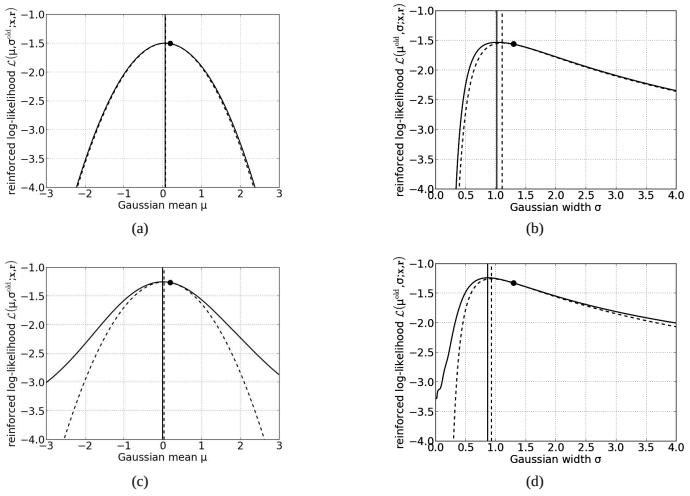


Figure 4: Examples of tangent lower bounds (dashed line) for the reinforced log-likelihood (plain line) in a Gaussian case. The reinforced probabilities are $\mathcal{N}\left(x_i|\mu,\sigma^{\text{old}}\right)+r_i$ (left) and $\mathcal{N}\left(x_i|\mu^{\text{old}},\sigma\right)+r_i$ (right), where PPRs are computed using either L_1 (up) or L_2 (down) regularisation. Vertical lines indicate the position of the maximum for the reinforced log-likelihood (plain line) and its lower bound (dashed line). Dots indicate points of tangency.

3.5. Weights and Degrees of Abnormality

Aside from its interesting properties for the optimisation of the model parameters, the weighted log-likelihood (44) also provides us a useful tool to analyse data. Indeed, when an optimum is reached, the weighted log-likelihood and the reinforced log-likelihood are equal. In such a case, the weight $w_i \in [0,1]$ measures the contribution of the likelihood of the ith observation to these quantities. The quantity $a_i = 1 - w_i \in [0,1]$ can be interpreted as an *abnormality degree* and may be easier to handle than the PPR r_i that belongs to $[0,\infty[$. When the PPR of an instance gets close to zero, its weight approaches one and its abnormality degree becomes zero. On the contrary, when the PPR increases, the weight tends to zero and the abnormality degree tends to one. In other words, data whose probability is highly reinforced (because they appear to be AFDs) are characterised by small weights and large abnormality degrees.

Fig. 5 shows the weight w_i in terms of the parametric probability $p\left(x_i|\theta\right)$ for L_1 and L_2 regularisation. The weight is close to one for large probabilities, but it decreases quickly as the probability decreases. On the contrary, notice that the abnormality degree a_i would be close to one for small probabilities, then decrease quickly as the probability increases.

4. Supervised and Unsupervised Inference using Pointwise Probability Reinforcements

This section adapts several standard statistical inference methods to reinforce them with PPRs: linear regression, kernel ridge regression (a.k.a. least-square support vector machines), logistic regression and principal component analysis. These four techniques tackle regression, classification and projection problems, what shows that PPRs allow one to easily deal with AFDs in various supervised or unsupervised contexts. For each method, experiments target outliers, since outliers are commonly recognized as harmful in the above applications.

4.1. Reinforced Linear Regression

Linear regression consists in fitting a linear prediction model $f(x_i) = \sum_{j=1}^{d} \beta_j x_{ij} + \beta_0$ to observed target values, where x_{ij} is the value of the jth feature for the ith observation and d is the dimensionality of data. Under the

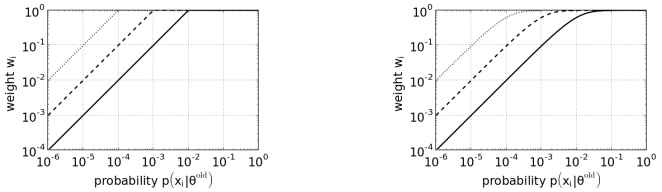


Figure 5: Observation weight w_i in terms of the probability $p(x_i|\theta)$ for L_1 (left) and L_2 (right) regularisation. The L_1 reinforcement meta-parameter is $\alpha = 10^2$ (plain line), $\alpha = 10^3$ (dashed line) and $\alpha = 10^4$ (dotted line). The L_2 reinforcement meta-parameter is $\alpha = 10^4$ (plain line), $\alpha = 10^6$ (dashed line) and $\alpha = 10^8$ (dotted line).

assumption that a Gaussian noise pollutes the observations, the maximum likelihood solution is given by the well-known ordinary least squares (OLS) estimator

$$\boldsymbol{\beta}_{\text{OLS}} = \left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{y} \tag{49}$$

where $\tilde{\mathbf{X}}$ is the $n \times (1+d)$ matrix of data with an additional column of ones and $\mathbf{y}$ is the vector of target values. For small datasets, the above estimator is quite sensitive to outliers (Cook, 1979; Beckman and Cook, 1983; Hadi and Simonoff, 1993), but it can easily be reinforced using PPRs. As discussed in Section 3.4, the model parameters optimisation step can be achieved by maximising a weighted log-likelihood $\mathcal{L}_{\mathbf{w}}(\boldsymbol{\beta}, \sigma; \mathbf{x}, \mathbf{y})$, which becomes

$$\frac{1}{2}\log\left[2\pi\sigma^2\right]\sum_{i=1}^{n} w_i + \frac{1}{2\sigma^2}\left(\tilde{\mathbf{X}}\boldsymbol{\beta} - \mathbf{y}\right)^T \mathbf{W}\left(\tilde{\mathbf{X}}\boldsymbol{\beta} - \mathbf{y}\right) \tag{50}$$

where σ is the Gaussian noise variance and $\mathbf{W}$ is a diagonal weighting matrix whose diagonal terms are $W_{ii} = w_i$. The solution maximising the above log-likelihood is similar to weighted least squares (WLS) estimator, except that the noise variance has to be also estimated. Indeed, the probabilities $p(y_i|x_i, \boldsymbol{\beta}, \sigma)$ must be estimated in order to obtain PPRs. The estimates are

$$\boldsymbol{\beta}_{\text{PPR}} = \left(\tilde{\mathbf{X}}^T\mathbf{W}\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{W}\mathbf{y} \tag{51}$$

and

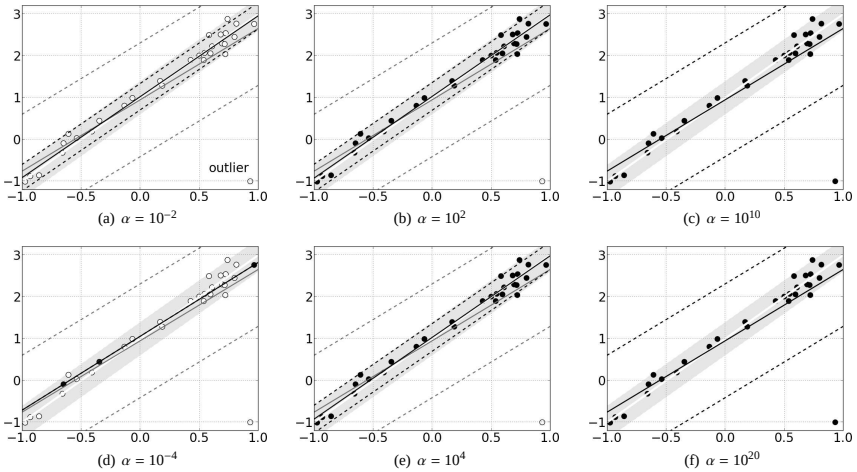
$$\sigma^2_{\text{PPR}} = \frac{\sum_{i=1}^{n} w_i\left(f(x_i) - y_i\right)^2}{\sum_{i=1}^{n} w_i}. \tag{52}$$

When L_1 regularisation is used, reinforced linear regression is similar to least trimmed squares (Ruppert and Carroll, 1980; Rousseeuw, 1984), which may be expensive for large datasets (Rousseeuw and Van Driessen, 2006).

Fig. 6 shows results obtained using reinforced linear regression with L_1 and L_2 regularisation on the PPRs. 29 data are generated from a unidimensional linear model with $\boldsymbol{\beta} = [2, 1]^T$ and $\sigma = 0.2$. Moreover, one outlier is added in order to interfere with inference. As a result, Fig. 6 shows that standard linear regression is biased and that its confidence interval for prediction is quite wide. On the one hand, Fig. 6(a) and 6(d) show that when the reinforcement meta-parameter α is too small, the outlier is not detected. For L_1 penalisation, all weights are identical, whereas they seem quite arbitrary for L_2 regularisation. Moreover, the confidence interval for the prediction of the reinforced linear regression is very narrow in that latter case. On the other hand, Fig. 6(c) and 6(f) show that when α is very large, the reinforced linear regression obtains results which are similar to standard linear regression results, since PPRs are forced to take very small values. A good compromise is obtained in Fig. 6(b) and 6(e) where the intermediate value of α allows PPRs to detect the outlier. Hence, the model is freed from its influence, what results in a more reliable model and improved 95% percent confidence intervals for predictions. Section 5 shows how to find an intermediate value of α corresponding to this compromise.

In order to illustrate the effect of the reinforcement of probabilities, let us consider the probability $\mathcal{N}(\epsilon_i|0, 0.2)$ of the residual $\epsilon_i = y_i - f(x_i)$. Fig. 7 shows the reinforced probability $\mathcal{N}(\epsilon_i|0, 0.2) + r_i$ obtained using L_1 and L_2 regularisation; the effect of the probability reinforcement is to clip the Gaussian distribution in the tails. Indeed, when $\mathcal{N}(\epsilon_i|0, 0.2)$ gets too small, it is replaced by $\frac{1}{\alpha}$ for L_1 regularisation and $\sqrt{\alpha^{-1}}$ for L_2 regularisation.

The reinforced linear regression allows us to deal with outliers, but it can also easily be adapted to penalise large feature weights β_j. Indeed, if a penalisation $\frac{\gamma}{2}\|\boldsymbol{\beta}\|^2$ is added to the reinforced log-likelihood, one obtains a reinforced ridge regression (Hoerl and Kennard,



(a) $\alpha = 10^{-2}$ (b) $\alpha = 10^{2}$ (c) $\alpha = 10^{10}$

(d) $\alpha = 10^{-4}$ (e) $\alpha = 10^{4}$ (f) $\alpha = 10^{20}$

Figure 6: Results obtained by standard linear regression (grey line) and reinforced linear regression (black line). The 95% percent confidence interval associated with the true function (white line) is shown by the grey-shaded area. PPRs are computed using L_1 (upper row) and L_2 (lower row) regularisation. The reinforcement meta-parameter values are indicated below each subfigure. The 30 data are shown by circles whose darkness is proportional to their respective weights. Dashed lines delimit 95% percent confidence intervals for prediction of each model, except in (d) where reinforced linear regression estimates an almost zero noise variance. In (c) and (f), standard and reinforced solutions are superimposed.

1970) where the weights are estimated by

$$\boldsymbol{\beta}_{\text{PPR}} = \left(\tilde{\mathbf{X}}^T\mathbf{W}\tilde{\mathbf{X}} + \gamma\mathbf{I}_d\right)^{-1}\tilde{\mathbf{X}}^T\mathbf{W}\mathbf{y} \tag{53}$$

where $\mathbf{I}_d$ is the $d \times d$ identity matrix. Interestingly, it can be seen that both regularisations coexist without problem in the above solution. Each regularisation takes independently care of one problem, what allows to readily separate outlier control and complexity control in a theoretically sound way.

4.2. Reinforced Kernel Ridge Regression

Kernel ridge regression (Saunders et al., 1998) (also called least squares support vector machine (Suykens et al., 2002)) is an extension of ridge regression where data are first mapped in a feature space. This kernel-based non-linear regression method can be formulated as a maximum likelihood problem. Indeed, ridge regression corresponds to assuming that the n errors $\epsilon_i = y_i - f(x_i)$ follow a Gaussian distribution $\mathcal{N}\left(\mathbf{0}, \sigma_\epsilon^2\mathbf{I}_n\right)$ and that the m weights in the feature space have a Gaussian prior $\mathcal{N}\left(0, \sigma_\beta^2\mathbf{I}_m\right)$, excluding the bias β_0. In such a case, it turns out that the prediction function is

$$f(x_i) = \sum_{j=1}^{n}\alpha_j k(x_i, x_j) + \beta_0 \tag{54}$$

where $\alpha_j > 0$ are dual weights and the kernel function k computes the dot product between two observations in the feature space (Muller et al., 2001). If ones introduces the meta-parameter $\gamma = \sigma_\beta^2/\sigma_\epsilon^2$ which controls the compromise between errors and model complexity and whose value is chosen using e.g. cross-validation, the parameter estimate of $\boldsymbol{\alpha}$ and β_0 is the solution of the linear system

$$\begin{pmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & K + \frac{1}{\gamma}\mathbf{I}_n \end{pmatrix}\begin{pmatrix}\beta_0 \\ \boldsymbol{\alpha}\end{pmatrix} = \begin{pmatrix}0 \\ \mathbf{y}\end{pmatrix} \tag{55}$$

where $\mathbf{K}$ is the Gram matrix such that $K_{ij} = k(x_i, x_j)$ and $\mathbf{1}_n$ is a n-element vector of 1's. Moreover, the standard deviation σ_ϵ can be estimated as

$$\sigma_\epsilon^2 = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2. \tag{56}$$

The above kernel ridge regression can easily be reinforced. Indeed, the weighted maximum likelihood prob-

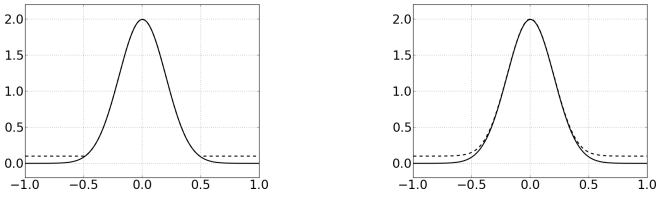


Figure 7: Comparison of the probability $\mathcal{N}(\epsilon_i|0, 0.2)$ (plain line) and the reinforced probability $\mathcal{N}(\epsilon_i|0, 0.2) + r_i$ (dashed line) of the residual $\epsilon_i = y_i - f(x_i)$ obtained using L_1 (left) and L_2 (right) regularisation. The reinforcement meta-parameter values are $\alpha_1 = 25$ and $\alpha_2 = 625$, respectively. In tails, the reinforced probability tends to $\frac{1}{\alpha_1} = 0.04$ and $\sqrt{\alpha_2^{-1}} = 0.04$.

lem solved in the parameter optimisation step is equivalent to a weighted kernel ridge regression (Wen et al., 2010; Liu et al., 2011). It follows that the parameter estimate of $\boldsymbol{\alpha}$ and β_0 is the solution of the linear system

$$\begin{pmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & K + \frac{1}{\gamma}\mathbf{W}^{-1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{y} \end{pmatrix}, \tag{57}$$

whereas the standard deviation σ_ϵ can be estimated as

$$\sigma_\epsilon^2 = \frac{\sum_{i=1}^n w_i \epsilon_i^2}{\sum_{i=1}^n w_i}. \tag{58}$$

The solutions in (55) and (57) are almost identical. The only difference is that the identity matrix $\mathbf{I}_n$ in (55) is replaced by the inverse of the weighting matrix $\mathbf{W}$ in (57). Since outliers correspond to large diagonal entries in $\mathbf{W}^{-1}$, the complexity control is less affected by them. Hence, the outlier control allows reducing the impact of outliers on complexity control.

Fig. 8 shows results obtained using reinforced kernel ridge regression with L_1 and L_2 regularisation on the PPRs. 29 data are generated from a unidimensional sinus polluted by a Gaussian noise with $\sigma = 0.1$. Moreover, one outlier is added in order to interfere with inference. For both the standard and the reinforced kernel ridge regression, the value of the meta-parameter is $\gamma = 10$ and the Gaussian kernel $k(x, z) = \exp \|x - z\|^2/0.1$ is used. The result of standard kernel ridge regression appears to be biased by the outlier. For small α values, Fig. 8(a) and 8(d) show that irrelevant models are obtained, because the $K\boldsymbol{\alpha}$ term is negligible with respect to the term $\frac{1}{\gamma}\mathbf{W}^{-1}\boldsymbol{\alpha}$ in the linear system (57). Fig. 8(c) and 8(f) show that reinforced kernel ridge regression performs similary to standard kernel ridge regression when α is very large. A good compromise is obtained in Fig. 8(b) and 8(e) where the intermediate value of α allows PPRs to detect outliers. Indeed, the outlier is clearly detected and the resulting non-linear regression model is freed from its influence. Moreover, 95% percent confidence intervals for predictions are also more reliable using PPRs. Section 5 shows how to find an intermediate value of α for a good compromise.

4.3. Reinforced Logistic Regression

Logistic regression is a standard classification model which linearly discriminates between two classes (0 and 1 here). Conditional class probabilities for this model are obtained using

$$\begin{aligned} p(Y_i = 1|X_i = x_i) &= 1 - p(Y_i = 0|X_i = x_i) \\ &= \frac{1}{1 + e^{-\sum_{j=1}^d \beta_j x_{ij} - \beta_0}} \end{aligned} \tag{59}$$

where Y_i is the class of the ith observation. Using the iterative reweighted least squares (IRLS) algorithm (Bishop, 2006), logistic regression can be efficiently performed. This quasi-Newton approach method consists in using the estimate

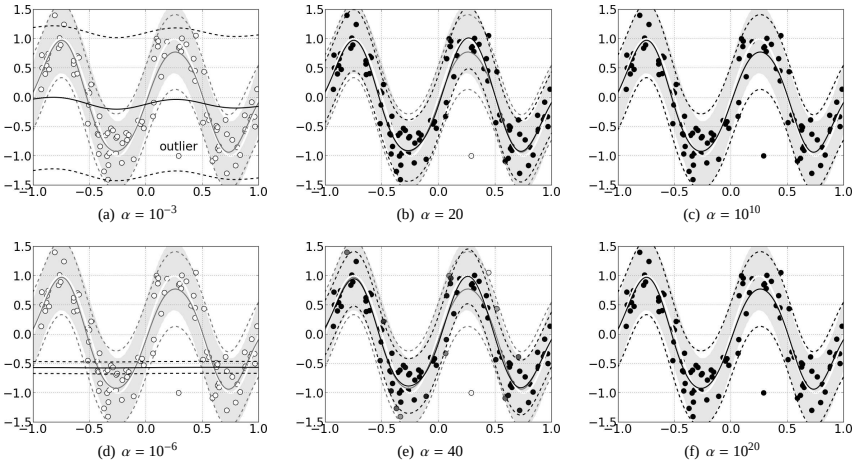
$$\boldsymbol{\beta}_{\text{IRLS}} = \left(\tilde{\mathbf{X}}^T \mathbf{R} \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^T \mathbf{R} \mathbf{z} \tag{60}$$

iteratively, where $\tilde{\mathbf{X}}$ is the $n \times (1 + d)$ matrix of data with an additional column of ones, $\mathbf{R}$ is a diagonal matrix whose diagonal terms are $R_{ii} = \sigma_i(1 - \sigma_i)$ with $\sigma_i = p(Y_i = 1|X_i = x_i)$ and $\mathbf{z}$ is a vector of altered targets

$$\mathbf{z} = \tilde{\mathbf{X}}\boldsymbol{\beta} - \mathbf{R}^{-1}(\boldsymbol{\sigma} - \mathbf{y}). \tag{61}$$

Logistic regression is sensitive to outliers (Rousseeuw and Christmann, 2003), but it can be reinforced using PPRs. Since the model parameter optimisation step is in fact a weighted logistic regression (Rousseeuw and Christmann, 2003), it can also be performed by an IRLS-like algorithm. The only modification is that the iterative update becomes

$$\boldsymbol{\beta}_{\text{IRLS}} = \left(\tilde{\mathbf{X}}^T \mathbf{W} \mathbf{R} \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^T \mathbf{W} \mathbf{R} \mathbf{z}. \tag{62}$$



(a) $\alpha = 10^{-3}$ (b) $\alpha = 20$ (c) $\alpha = 10^{10}$

(d) $\alpha = 10^{-6}$ (e) $\alpha = 40$ (f) $\alpha = 10^{20}$

Figure 8: Results obtained by standard kernel ridge regression (grey line) and reinforced kernel ridge regression (black line). The 95% percent confidence interval associated with the true function (white line) is shown by the grey-shaded area. PPRs are computed using L_1 (upper row) and L_2 (lower row) regularisation. The reinforcement meta-parameter values are indicated below each subfigure. The 30 data are shown by circles whose darkness is proportional to their respective weights. Dashed lines delimit 95% percent confidence intervals. In (c) and (f), standard and reinforced solutions are superimposed.

Fig. 9 shows results obtained using reinforced logistic regression with L_1 and L_2 regularisation on the PPRs. 30 data are generated from two classes with Gaussian distributions $\mathcal{N}(\mu = \pm 2, \sigma = 1.7)$. In order to introduce an outlier, the label of one observation from class 1 is flipped, i.e. a labelling error is introduced which alters the result of standard logistic regression. On the one hand, Fig. 9(a) and 9(d) show that when the reinforcement meta-parameter α is too small, the outlier is not detected. On the other hand, Fig. 9(c) and 9(f) show that when α is very large, the reinforced logistic regression obtains results which are similar to standard logistic regression results with polluted data, since PPRs are forced to take very small values. A good compromise is obtained in Fig. 9(b) and 9(e) where the reinforced logistic regression produces a model which is very close to the model obtained by standard logistic regression with no labelling error. Section 5 shows how to find an intermediate value of α which allows a good compromise.

4.4. Reinforced Principal Component Analysis

Principal component analysis (PCA) finds the q principal (or maximum variance) axes of a data cloud. This unsupervised procedure projects data onto a smaller dimensional space, while keeping the most of the feature variance. PCA can be cast as a probabilistic method (Tipping and Bishop, 1999) by assuming that (i) data are generated by q hidden independent Gaussian sources Z with distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_q)$ and (ii) that one only observes d features X whose conditional distribution is

$$p\left(X = x|Z = z, \boldsymbol{\mu}, \sigma\right) = \mathcal{N}\left(x|\mathbf{A}z + \boldsymbol{\mu}, \sigma^2\mathbf{I}_d\right) \tag{63}$$

where $\mathbf{A}$ is $d \times q$ linear transformation matrix, μ is a d-dimensional translation vector and σ is the noise standard deviation. Tipping and Bishop (1999) show that the observed features have a marginal distribution

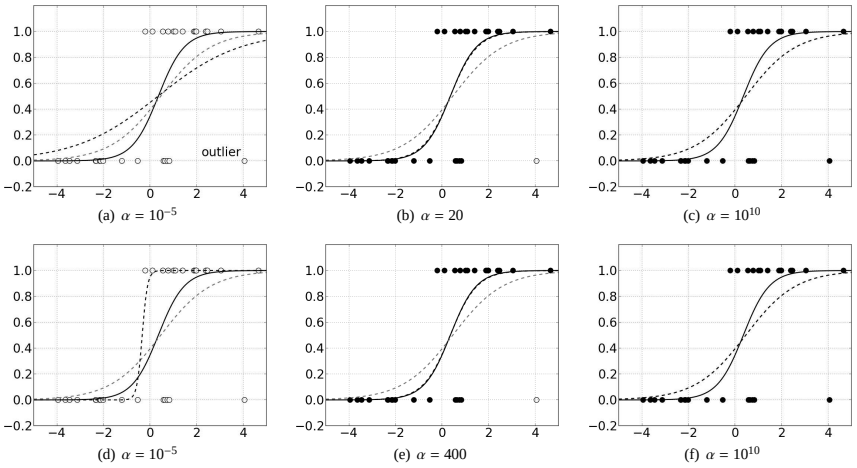
$$p\left(X = x|\boldsymbol{\mu}, \sigma\right) = \mathcal{N}\left(x|\boldsymbol{\mu}, \mathbf{C}\right) \tag{64}$$

where $\mathbf{C} = \mathbf{A}\mathbf{A}^T + \sigma^2\mathbf{I}_d$. Hence, the data log-likelihood is

$$\mathcal{L}\left(\mathbf{A}, \sigma; \mathbf{x}\right) = -\frac{n}{2}\left[d\log\left[2\pi\right] + \log|\mathbf{C}| + \mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{S}\right)\right] \tag{65}$$

where the sample covariance matrix $\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \boldsymbol{\mu}\right)\left(x_i - \boldsymbol{\mu}\right)^T$ is computed using the sample mean $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$. The maximum likelihood solution is

$$\mathbf{A}_{\mathrm{ML}} = \mathbf{U}_q\left(\boldsymbol{\Lambda}_q - \sigma^2\mathbf{I}_q\right)^{\frac{1}{2}} \tag{66}$$



(a) $\alpha = 10^{-5}$ (b) $\alpha = 20$ (c) $\alpha = 10^{10}$

(d) $\alpha = 10^{-5}$ (e) $\alpha = 400$ (f) $\alpha = 10^{10}$

Figure 9: Results obtained by standard logistic regression (grey dashed line) and reinforced logistic regression (black dashed line) from polluted data, with respect to the result obtained by standard logistic regression from clean data (black plain line). PPRs are computed using L_1 (upper row) and L_2 (lower row) regularisation. The reinforcement meta-parameter values are indicated below each subfigure. The 30 data are shown by circles whose darkness is proportional to their weights. The observed (noisy) labels are indicated by the y-value of each point (0 or 1). In (b) and (e), reinforced and standard from clean data solutions are superimposed. In (c) and (f), standard from polluted data and reinforced solutions are superimposed.

where $\mathbf{U}_q$ contains the q principal eigenvectors of $\mathbf{S}$ as columns and $\mathbf{\Lambda}_q$ is a diagonal matrix containing the q corresponding eigenvalues (Tipping and Bishop, 1999). Moreover, the maximum likelihood estimator of σ when $\mathbf{A} = \mathbf{A}_{\text{ML}}$ is given for $d > q$ by

$$\sigma^2_{\text{ML}} = \frac{1}{d-q} \sum_{i=q+1}^{d} \lambda_i. \tag{67}$$

When $d = q$, data are reconstructed with no error and σ_{ML} is zero. PCA is known to be sensitive to outliers (Xu and Yuille, 1995; Archambeau et al., 2006), but this method can be easily reinforced. Indeed, it turns out that the parameter optimisation step is a weighted PCA (Huber, 1981; Fan et al., 2011), which simply consists in using (66) and (67) with the eigenvectors and eigenvalues of the weighted sample covariance matrix

$$\mathbf{S} = \frac{\sum_{i=1}^{n} w_i \, (x_i - \boldsymbol{\mu}) \, (x_i - \boldsymbol{\mu})^T}{\sum_{i=1}^{n} w_i} \tag{68}$$

where $\boldsymbol{\mu}$ is the weighted sample mean

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}. \tag{69}$$

Similarly to the other reinforced methods, the weights are obtained using the definition (34) and the marginal probabilities given by (64).

Fig. 10 shows the results obtained using reinforced PCA with L_1 and L_2 regularisation on the PPRs. 49 data are (i) generated from a two-dimensional isotropic Gaussian distribution with mean $[0, 0]^T$ and covariance matrix $\mathbf{I}_2$ and (ii) transformed using the linear transformation matrix

$$\mathbf{A} = \begin{pmatrix} \cos\frac{\pi}{6} & -\sin\frac{\pi}{6} \\ \sin\frac{\pi}{6} & \cos\frac{\pi}{6} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0.2 \end{pmatrix} = \begin{pmatrix} 0.87 & -0.1 \\ 0.5 & 0.87 \end{pmatrix} \tag{70}$$

and the translation matrix $\mu = [-0.5, 0.5]^T$. Moreover, one outlier is added in order to interfere with inference. As a result, the axes given by standard PCA are slightly rotated and stretched. On the one hand, Fig. 10(a) and 10(d) show that when the reinforcement meta-parameter α is too small, unconvincing results are obtained because PPRs are free to take large values. On the other hand, Fig. 10(c) and 10(f) show that when α is very large, our approach obtains results which are similar to standard PCA results, since PPRs are forced to take very small values. A good compromise is obtained in Fig. 10(b) and 10(e) with an intermediate value of α. Sec-

tion 5 shows how to find such an intermediate value of α.

5. Choice of the Reinforcement Meta-Parameter

As mentioned in Section 3.2, the reinforcement meta-parameter α determines the amount of reinforcement which is allowed. As illustrated by the results of the various reinforced methods presented in Section 4, small values of α lead to large PPRs, whereas large values of α lead to small PPRs. In the former case, the parametric model is insensitive to AFDs but may poorly fit data, since all of them are seen as AFDs with large PPRs. In the latter case, none of the data can be seen as an AFD and the parametric model is given by the standard maximum likelihood solution, which is sensitive to AFDs. In conclusion, it is important to choose a good intermediate value of α where only a few data can be considered as AFDs with large PPRs, what results in model parameters which make sense and are less sensitive to e.g. outliers. This important problem is discussed in this section.

5.1. Meta-Parameter Optimisation Schemes

Two common approaches to deal with meta-parameters are validation and Bayesian priors. Validation consists in choosing the best meta-parameter with respect to the performances obtained by the model on test data. This approach includes cross-validation, leave-one-out validation, bootstrap, etc. However, it is in practice impossible to obtain test data which are guaranteed to be clean from outliers. Hence, since using PPRs produces models which are less sensitive to AFDs and may therefore give them very small probabilities, a validation approach would probably choose a very large α value, what is undesirable. It is not easy either to define a sensible Bayesian prior for the α parameter. This paper proposes an alternative way to optimise α. As discussed in Section 3.5, the instance weights measure the contribution of each observation to the parameter estimation and their mean can be seen as the percentage of the training sample which is actually used. Equivalently, the mean of the abnormality degrees a_i can be seen as the percentage of observations which are considered as AFDs. Hence, a simple solution to the α optimisation problem consists in using the α value which correspond to a plausible percentage of data supporting the model and a plausible percentage of outliers. A good choice is for example 95% for the former quantity or, equivalently, 5% for the latter. Several works suggest that it is more harmful to keep too many outliers than to remove too many correct data (Brodley and Friedl, 1999). Moreover, in the literature, real-world databases are estimated to contain around five percents of encoding errors (Redman, 1998; Maletic and Marcus, 2000). Of course, if prior knowledge suggests that more or less outliers are present in data, another percentage could be used.

Along this idea, meta-parameter α can be adapted as follows. Model parameters are initialised using a maximum likelihood approach, what corresponds to an infinite α value. Then, in the first iteration, an α value is searched (see Sections 5.2 and 5.3) in order to produce PPRs which are consistent with the constraint

$$\overline{w}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} w_i \approx 0.95. \tag{71}$$

The resulting PPRs are used to optimise model parameters and the algorithm iterates until convergence. At each iteration, a new α value is computed, since the weights w_i have changed meanwhile. In the end, PPRs and model parameters are obtained, whereas the condition (71) is satisfied.

5.2. The L_1 Regularised Case

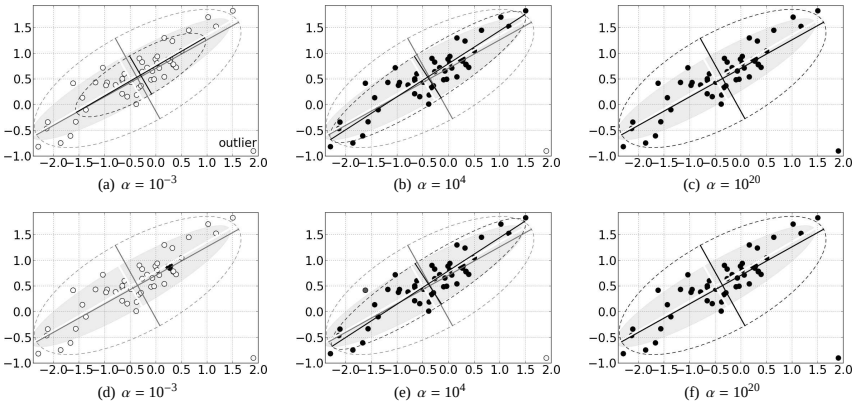
When L_1 regularisation is used, observations must be first ordered according to their probability $p(x_i|\boldsymbol{\theta})$. Indeed, it follows from (13) and (34) that any observation whose probability is smaller than $\frac{1}{\alpha}$ has a weight $w_i = \alpha p(x_i|\boldsymbol{\theta})$, whereas all other instances have unitary weight. Then, the α search consists in looking for the smallest number k of observations with subunitary weight such that

$$\begin{aligned}\overline{w}(\alpha) &= \frac{1}{n}\sum_{i=1}^{n} w_i \\ &= \frac{1}{n}\sum_{i=1}^{k} \frac{1}{p(x_{k+1}|\boldsymbol{\theta})} p(x_i|\boldsymbol{\theta}) + \frac{1}{n}\sum_{i=k+1}^{n} 1 \leq 0.95,\end{aligned} \tag{72}$$

where α has been replaced by $1/p(x_{k+1}|\boldsymbol{\theta})$ since x_{k+1} is one of the instances with unitary weight $w_{k+1} = 1 = \alpha p(x_{k+1}|\boldsymbol{\theta})$. Since the $n-k$ last observations (and only them) necessarily have a unitary weight for $\overline{w}(\alpha) = 0.95$, the value of α which satisfies (71) can be estimated as

$$\alpha \approx \frac{\frac{k}{n} - 0.05}{\frac{1}{n}\sum_{i=1}^{k} p(x_i|\boldsymbol{\theta})}. \tag{73}$$

Fig. 11 shows an example of reinforcement meta-parameter choice for linear regression. Fig. 11(a) shows



(a) $\alpha = 10^{-3}$ (b) $\alpha = 10^{4}$ (c) $\alpha = 10^{20}$

(d) $\alpha = 10^{-3}$ (e) $\alpha = 10^{4}$ (f) $\alpha = 10^{20}$

Figure 10: Axes obtained by standard PCA (grey lines) and reinforced PCA (black lines), with respect to the true axes of the hidden data model (white lines) for which the grey-shaded area shows the true 95% percent confidence region. PPRs are computed using L_1 (upper row) and L_2 (lower row) regularisation. The reinforcement meta-parameter values are indicated below each subfigure. The 50 data are shown by circles whose darkness is proportional to their respective weights. Dashed lines are level curves of the Gaussian distribution which delimit the 95% percent confidence region for each model, except in (d) where the reinforced PCA estimates an almost zero variance in the second principal axis direction. In (c) and (f), standard and reinforced solutions are superimposed.

the mean of weights in terms of α for optimal model parameters, whereas Fig. 11(b) shows the linear regression which is obtained for the mean of weights $\overline{w}(\alpha) = 0.95$ with $\alpha = 10.19$. This solution is obtained with 3 iterations of the two-step iterative algorithm proposed in Sections 3 and 4.1.

5.3. The L_2 Regularised Case

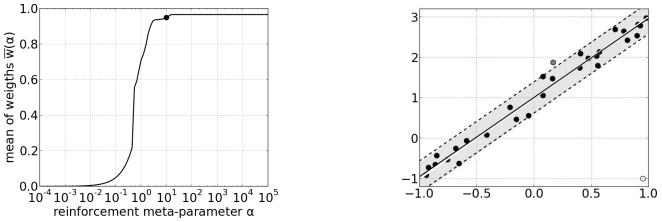
For L_2 regularisation, a dichotomy approach can be used. Indeed, the mean of the weights is an increasing function of α, since its first-order derivative can easily be shown to be always positive. Initially, two very small and very large initial values of α (e.g. $\alpha_1 = 10^{-4}$ and $\alpha_3 = 10^{20}$) are picked. Then, one computes the intermediate value $\alpha_2 = \sqrt{\alpha_1 \alpha_3}$ and only the two values $\alpha_i < \alpha_j$ such that $0.95 \in \left[\overline{w}(\alpha_i), \overline{w}(\alpha_j)\right]$ are kept. The algorithm is iterated until α_1 and α_3 are close enough, where the value α_2 can be chosen, since (71) is sufficiently close to be satisfied for $\overline{w}(\alpha_2)$. Fig. 12(a) shows the mean of weights in terms of α for optimal model parameters, whereas Fig. 12(b) shows the linear regression which is obtained for the mean of weights $\overline{w}(\alpha) = 0.95$ with $\alpha = 323.08$. This solution is obtained with 18 iterations of the two-step iterative algorithm proposed in Sections 3 and 4.1.

5.4. Computational Cost of Reinforcing Probabilites

This section analyses the computational cost of the PPR methodology, with the above method for the choice of the reinforcement meta-parameter α. At each iteration of the algorithm proposed in Section 3.2, irrespective of the model type, three steps are performed.

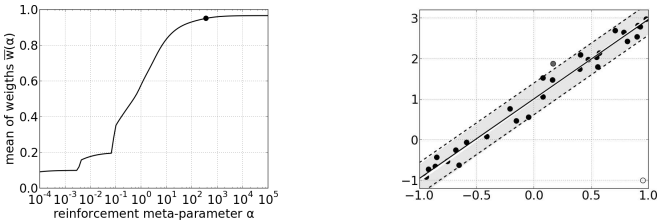
First, the probabilities $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ are computed using the current estimate $\boldsymbol{\theta}^{\text{old}}$ of the model parameters. The computational cost depends on the considered model and is therefore difficult to characterise precisely. However, evaluating the probabilities $p\left(x_i|\boldsymbol{\theta}^{\text{old}}\right)$ is expected to be much faster than learning the estimate $\boldsymbol{\theta}^{\text{old}}$ itself. For example, in the case of linear regression, the former only requires to estimate the model output for each observation and to compute its probability, whereas the latter requires to compute a pseudo-inverse.

Second, the reinforcement meta-parameter α is optimised and the PPRs r_i are obtained from the probabilities. The computational cost depends on the type of regularisation. For L_1 and L_2 regularisation, Equations (12) and (17) provide cheap closed-form expression for the PPRs. With L_1 regularisation, the reinforcement meta-parameter is first estimated as explained in Section 5.2 and then the PPRs are computed. With L_2 regularisation, the PPRs must be computed at each step of the reinforcement meta-parameter search. The com-



(a) (b)

Figure 11: Reinforcement meta-parameter selection for linear regression with L_1 regularisation on PPRs. The left panel shows the mean of weights in terms of α for optimal model parameters. The right panel shows the linear regression which is obtained with $\alpha = 10.19$ (black line), with respect to the true function (white line). The 30 data are shown by circles whose darkness is proportional to their respective weights. The estimated and true 95% percent confidence intervals are shown by dashed lines and the shaded region, respectively.



(a) (b)

Figure 12: Reinforcement meta-parameter selection for linear regression with L_2 regularisation on PPRs. The left panel shows the mean of weights in terms of α for optimal model parameters. The right panel shows the linear regression which is obtained with $\alpha = 323.08$ (black line), with respect to the true function (white line). The 30 data are shown by circles whose darkness is proportional to the sample weights. The estimated and true 95% percent confidence intervals are shown by dashed lines and the shaded region, respectively.

putational cost with L_2 regularisation is higher, but the computation of the reinforcement meta-parameter and of the PPRs only consists of simple operations.

Third, the instance weights w_i are computed and the model parameters are optimised with a weighted log-likelihood algorithm. Whereas weights are obtained using the simple closed-form expression (34), the model parameter optimisation is the most computationally expensive step in the proposed methodology. Indeed, learning algorithms usually involve costly operations like matrix inversion, gradient descent or convex optimisation with often non-linear time complexities.

Overall, the cost of reinforcing probabilities is dominated by the optimisation of the model parameters, since all other operations involve computations which are comparatively more simple. In practice, only a few iterations of the proposed algorithm are necessary before convergence. For example, in the two problems discussed in Sections 5.2 and 5.3, 3 and 18 iterations are necessary to converge when L_1 or L_2 regularisation is used, respectively. Experimentally, it is observed that the number of iterations decreases as the number of training instances increases. For small sample sizes, modern machine learning techniques are fast enough to cope with training the model a few times. For iterative learning procedures like gradient descent or convex optimisation, the proposed methodology can be considerably sped up by using the model parameters $\boldsymbol{\theta}^{\text{old}}$ obtained at a given iteration as a seed for the model parameters optimisation in the next iteration. Convergence of such methods should be much faster this way.

6. Conclusion

This paper introduces a generic method to deal with outliers in the case of probabilistic models. Probabilities are reinforced by pointwise probability reinforcements

(PPRs), whose properties can be controlled by regularisation. It is shown that L_1 regularisation induces sparse PPRs, what results in a few observations being considered as potential abnormally frequent data. Using L_2 regularisation, a similar, yet smoother solution is obtained. The adaptation of four standard probabilistic techniques (linear regression, kernel ridge regression, logistic regression and PCA) shows the generality of the proposed approach. Outliers can be detected in supervised or unsupervised contexts and a degree of abnormality can be obtained for each observation. The average degree of abnormality can be easily controlled using a meta-parameter α, what can be used to select an adequate compromise in the regularisation.

References

Aitkin, M., Wilson, G. T., 1980. Mixture models, outliers, and the EM algorithm. Technometrics 22 (3), 325–331.

Archambeau, C., Delannay, N., Verleysen, M., Jun. 2006. Robust probabilistic projections. In: Proceedings of the 23rd Int. Conf. on Machine Learning. Pittsburgh, PA, pp. 33–40.

Barber, D., 2012. Bayesian Reasoning and Machine Learning. Cambridge University Press, Cambridge, UK.

Barnett, V., Lewis, T., 1994. Outliers in Statistical Data. Wiley, New York, NY.

Beckman, R. J., Cook, R. D., 1983. Outlier.....s. Technometrics 25 (2), 119–149.

Bernardo, J., Smith, A., 2007. Bayesian Theory. Wiley, New York, NY.

Bishop, C., 2006. Pattern recognition and machine learning. Springer, Berlin.

Brodley, C. E., Friedl, M. A., 1999. Identifying mislabeled training data. Journal of Artificial Intelligence Research 11, 131–167.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Computing Surveys 41 (3), 15:1–15:58.

Chen, D., Jain, R., 1994. A robust backpropagation learning algorithm for function approximation. IEEE Transactions on Neural Networks 5 (3), 467–479.

Chuang, C.-C., Su, S.-F., Hsiao, C.-C., 2000. The annealing robust backpropagation (arbp) learning algorithm. IEEE Transactions on Neural Networks 11 (5), 1067–1077.

Cook, R. D., 1979. Influential observations in linear regression. Journal of the American Statistical Association 74 (365), 169–174.

DasGupta, A., 2011. Probability for Statistics and Machine Learning. Springer, Berlin, Ch. The Exponential Family and Statistical Applications, pp. 498–521.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 39 (1), 1–38.

Domingo, C., Watanabe, O., Jun.–Jul. 2000. Madaboost: A modification of adaboost. In: Proceedings of the 13th Ann. Conf. on Computational Learning Theory. Palo Alto, CA, pp. 180–189.

Duda, R. O., Hart, P. E., 1973. Pattern Classification and Scene Analysis. Wiley, New York, NY.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Annals of Statistics 32, 407–499.

Eskin, E., Jun.–Jul. 2000. Anomaly detection over noisy data using learned probability distributions. In: Proceedings of the 17th Int. Conf. on Machine Learning. Stanford, CA, pp. 255–262.

Fan, Z., Liu, E., Xu, B., Sep. 2011. Weighted principal component analysis. In: Proceedings of the 3rd Int. Conf. on Artificial Intelligence and Computational Intelligence, Part III. Taiyuan, China, pp. 569–574.

Ganapathiraju, A., Picone, J., State, M., Oct. 2000. Support vector machines for automatic data cleanup. In: Proceedings of the 6th Int. Conf. on Spoken Language Processing. Beijing, China, pp. 210–213.

Guyon, I., Matic, N., Vapnik, V., 1996. Discovering informative patterns and data cleaning. In: Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds.), Advances in knowledge discovery and data mining. AAAI/MIT Press, Cambridge, MA, pp. 181–203.

Hadi, A. S., Luceo, A., 1997. Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. Computational Statistics & Data Analysis 25 (3), 251–272.

Hadi, A. S., Simonoff, J. S., 1993. Procedures for the identification of multiple outliers in linear models. Journal of the American Statistical Association 88 (424), 1264–1272.

Hawkins, D. M., 1980. Identification of outliers. Chapman and Hall, London, UK.

Hernandez-Lobato, D., Hernandez-Lobato, J. M., Dupont, P., Dec. 2011. Robust multi-class gaussian process classification. In: Advances in Neural Information Processing Systems 24. Granada, Spain, pp. 280–288.

Hodge, V., Austin, J., 2004. A survey of outlier detection methodologies. Artificial Intelligence Review 22 (2), 85–126.

Hoerl, A. E., Kennard, R. W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12 (1), 55–67.

Hu, F., Zidek, J. V., 2002. The weighted likelihood. Canadian Journal of Statistics 30 (3), 347–371.

Huber, P. J., 1964. Robust estimation of a location parameter. Annals of Mathematical Statistics 35 (1), 73–101.

Huber, P. J., 1981. Robust Statistics. Wiley, New York, NY.

Kowalczyk, A., Smola, A. J., Williamson, R. C., Dec. 2001. Kernel machines and boolean functions. In: Advances in Neural Information Processing Systems 14. Vancouver, British Columbia, Canada, pp. 439–446.

Lawrence, N. D., Schölkopf, B., Jun.–Jul. 2001. Estimating a kernel fisher discriminant in the presence of label noise. In: Proceedings of the 18th Int. Conf. Machine Learning. Williamstown, MA, pp. 306–313.

Liano, K., 1996. Robust error measure for supervised neural network learning with outliers. IEEE Transactions on Neural Networks 7 (1), 246–250.

Liu, J., Li, J., Xu, W., Shi, Y., 2011. A weighted lq adaptive least squares support vector machine classifiers robust and sparse approximation. Expert Systems with Applications 38 (3), 2253–2259.

Maletic, J. I., Marcus, A., Oct. 2000. Data cleansing: Beyond integrity analysis. In: Proceedings of the Conf. on Information Quality. Cambridge, MA, pp. 200–209.

Muller, K. R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B., 2001. An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12 (2), 181–201.

Redman, T., 1998. The impact of poor data quality on the typical enterprise. Communications of the ACM 2 (2), 79–82.

Rekaya, R., Weigel, K. A., Gianola, D., 2001. Threshold model for misclassified binary responses with applications to animal breeding. Biometrics 57 (4), 1123–1129.

Rousseeuw, P., 1984. Least median of squares regression. Journal of the American Statistical Association 79 (388), 871–880.

Rousseeuw, P., Christmann, A., 2003. Robustness against separation and outliers in logistic regression. Computational Statistics & Data Analysis 43 (3), 315–332.

Rousseeuw, P., Van Driessen, K., 2006. Computing lts regression for large data sets. Data Mining and Knowledge Discovery 12, 29–45.
Ruppert, D., Carroll, R. J., 1980. Trimmed least squares estimation in the linear model. Journal of the American Statistical Association 75 (372), 828–838.
Saunders, C., Gammerman, A., Vovk, V., Jul. 1998. Ridge regression learning algorithm in dual variables. In: Proceedings of the 15th Int. Conf. on Machine Learning. Madison, WI, pp. 515–521.
Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J., 2002. Least Squares Support Vector Machines. World Scientific, Singapore.
Tipping, M. E., Bishop, C., 1999. Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 61, 611–622.
Wen, W., Hao, Z., Yang, X., 2010. Robust least squares support vector machine based on recursive outlier elimination. Soft Computing 14 (11), 1241–1251.
Xu, L., Yuille, A., 1995. Robust principal component analysis by self-organizing rules based on statistical physics approach. IEEE Transactions on Neural Networks 6 (1), 131–143.
Zhang, W., Rekaya, R., Bertrand, K., 2006. A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. Bioinformatics 22 (3), 317–325.
Zhu, X., Wu, X., 2004. Class noise vs. attribute noise: A quantitative study. Artificial Intelligence Review 22, 177–210.

Chapter 14

Using SVMs with Randomised Feature Spaces: an Extreme Learning Approach

The following article has been presented at the 18th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, 28-30 April 2010. This paper proposes to merge extreme learning machine (ELM) and support vector machines by defining a new kernel. It shows what happen when the number of neurons in ELMs becomes very large and is related to [10]. Reprinted with permission from [9].

Using SVMs with randomised feature spaces: an extreme learning approach

Benoît Frénay and Michel Verleysen

Université catholique de Louvain - Machine Learning Group
EPL/ELEC/DICE - Place du Levant, 3 1348 Louvain-la-Neuve, Belgium

Abstract. Extreme learning machines are fast models which almost compare to standard SVMs in terms of accuracy, but are much faster. However, they optimise a sum of squared errors whereas SVMs are maximum-margin classifiers. This paper proposes to merge both approaches by defining a new kernel. This kernel is computed by the first layer of an extreme learning machine and used to train a SVM. Experiments show that this new kernel compares to the standard RBF kernel in terms of accuracy and is faster. Indeed, experiments show that the number of neurons of the ELM behind the randomised kernel does not need to be tuned and can be set to a sufficient value without altering the accuracy significantly.

1 Introduction

In classification, support vector machines (SVMs) are state-of-the-art models which have been used to solve many problems. Their success is due to three factors. Firstly, they are maximum-margin classifiers. Secondly, the dual form of the SVM optimisation problem is quadratic and its computational complexity depends on the number of data, not on their dimensionality. Thirdly, since the optimisation of their dual form only needs the inner products of data points and not the data points themselves, kernels can easily be plugged into SVMs.

When using SVMs, three choices must be made: the kernel type, the kernel parameters and the regularisation parameter. These choices are critical for the quality of the results and are usually done using cross-validation. However, this process can be computationally intensive since many models have to be built.

This paper presents a new approach inspired by the extreme learning machines (ELMs) which allows focusing on the regularisation. It proposes to build an explicit feature space using random neurons, as in extreme learning. Then, the corresponding Gram matrix is computed and used with a standard SVM. Experiments show that this approach obtains results which compare to those obtained with RBF kernels. Only the computational time is smaller in our case, because we have no kernel parameters to tune. Indeed, it appears experimentally that the dimensionality of the randomised feature space can be set to a sufficient value, e.g. 10^3, without decreasing the classification accuracy significantly. Therefore, only the regularisation constant has to be tuned.

The remaining of this paper is organised as follows. Sections 2 and 3 quickly review the SVMs and ELMs. Section 4 presents our approach mixing these frameworks. Section 5 is concerned with the experimental comparison of our approach with standard SVMs. Eventually, Section 6 concludes this paper.

2 SVMs in a nutshell

This section quickly reviews the SVM classifiers and the RBF kernel.

2.1 Maximum-margin classifiers

In binary classification, a SVM [1] is a classifier which separates the two classes by a maximum-margin hyperplane. The margin is the distance between the hyperplane and its closest data point. The idea behind SVMs is to maximise that margin, i.e. to move the hyperplane away from the data points while keeping them separated. This optimisation problem can be stated formally as

$$\left| \begin{array}{l} \min\limits_{w,b,\xi_i} \|w\|_2^2 + C \sum_i \xi_i \\ \text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{array} \right. \tag{1}$$

where w is the weight vector (maximising the margin is equivalent to minimising the norm $\|w\|_2^2$), ξ_i is the distance between the ith data point and the hyperplane corresponding to $y_i(w \cdot x_i + b) = 1$ and C is a regularisation constant. The regularisation is necessary because the data are usually not separable. The higher the value of C, the higher the constraint and the less likely the overfitting.

There are theoretical arguments in favor of SVMs [2] and their implementation is fast. Indeed, the dual form of Eq. 1 is a quadratic optimisation problem which only involves the inner products between data points. Hence, its computational complexity depends on the number of data points, not on their dimension.

2.2 SVMs and kernels

In practice, SVMs are often used jointly with kernels. Indeed, the dual form of Eq. 1 only needs the inner products of the data. They can be computed by any valid kernel. The joint use of SVMs and kernels opens the way for using SVMs in a wide range of domains: there exist kernels to deal with real numbers, sequences, graphs etc. However, the practitioner has to (i) choose the right kernel, (ii) tune the kernel parameters (if any) and (iii) tune the regularisation parameter. This can be done using e.g. cross-validation, but it is usually computationally intensive.

In this paper, we focus on applications where the inputs are numerical. In practice, linear, polynomial and radial basis function (RBF) kernels are often used. We will compare our approach to the RBF kernel

$$k(x, z) = \exp(-\gamma \|x - z\|_2^2) \tag{2}$$

which is very common and usually gives very good results. Its only parameter is the kernel width γ which regulates the scale at which the SVM is looking at the data points. The larger the kernel width γ, the larger the scale.

3 Basis of extreme learning

This section quickly reviews the basis and implementation of extreme learning.

3.1 Extreme learning machines

In its most standard form, a MLP is a two-layer feedforward neural network with a non-linear activation function for the hidden layer and a linear activation function for the output layer. MLPs can solve classification problems, but their training relies on the slow backpropagation algorithm which is difficult to tune.

In [3], Huang et al. propose to keep the MLP structure with a different learning algorithm. First, they scale the inputs in $[-1, 1]$ and initialise the first layer weights and biases at random, e.g. in $[-3, 3]$. The n data points are then transformed by the p neurons of the hidden layer from $x_i \in \Re^d$ to $\phi(x_i) \in \Re^p$. Here, ϕ is the sigmoid function $\phi(x_i) = 1/(1+\exp(-w \cdot x_i - b))$. The output layer being linear, the problem boils down to solving a linear system with n equations and p variables

$$\Phi w = Y \tag{3}$$

where Φ is a $n \times p$ matrix which rows are the transformed data points, w is the weight vector and Y contains the labels y_i. Usually, p is much larger than d, hence the name *extreme learning machines*. The classification of a new data point x is given by $y = \text{sign}(x \cdot w)$.

3.2 Implementations

In [3, 4], Huang et al. solve Eq. 3 using the Moore-Penrose pseudoinverse and an iterative algorithm. They show that ELMs with a large enough number of neurons obtain satisfactory results in a much shorter computation time than SVMs. The number of neurons has to be tuned using e.g. cross-validation. In [5], they applied successfully the same approach to RBFs. Moreover, Miche et al. [6] and Liu et al. [7] have shown that ELMs can be pruned using LASSO or regularised w.r.t. the L2 norm of the output weights vector.

4 Proposed approach

This section describes the proposed approach merging both the SVMs and ELMs. Indeed, the transformation performed by the first layer of an ELM can be seen as a kernel which can be plugged into a SVM. The subsequent section will show that the size of the ELM is not critical and can be set to a sufficient, large value.

4.1 The ELM kernel

ELMs are fast, but they do not search for maximum-margin hyperplanes. Instead, they minimise a sum of squared errors between the class labels and the MLP output. This kind of criterion is not really suitable for classification. In this paper, we propose to merge both the SVM and ELM approaches in order to obtain models which (i) are fast to train and (ii) are maximum-margin classifiers.

In the ELM framework, we can think of the hidden layer as a mapping ϕ from the data space $\Re^d$ to a feature space $\Re^p$ where we have to solve a linear problem. This is very similar to the idea behind the use of kernels: statistically,

the data points are easier to separate in a higher dimensional space. Hence, the first layer of an ELM can be thought as defining some kind of randomised kernel, which will be subsequently called the *ELM kernel*.

Given two data points x and z and an ELM with p neurons defining a mapping ϕ from $\Re^d$ to $\Re^p$, the corresponding ELM kernel function is defined as

$$k(x, z) = \frac{1}{p}\phi(x) \cdot \phi(z). \tag{4}$$

Therefore, the Gram matrix corresponding to this ELM kernel and a set of data points can be computed as

$$G = \frac{1}{p}\Phi\Phi'. \tag{5}$$

In [7], Liu et al. also propose to use the explicit mapping described above. However, their extreme SVMs use neither the maximum-margin principle nor the advantages of the dual form. Indeed, they are rather doing a Ridge regression in the ELM feature space which has a time complexity depending on the dimensionality of the data. To our knowledge, the ELM kernel has never been used jointly with real SVMs which (i) search for maximum-margin hyperplanes and (ii) offer a time complexity which is particularly interesting when using an ELM kernel computed in a high dimensional feature space.

4.2 Merging extreme learning and SVMs

Since Huang et al. showed that the ELM kernel gives good results if jointly used with a linear classifier, it makes sense to question whether the same holds for SVMs. Therefore, in the rest of this paper, we will assess the results obtained by plugging the ELM kernel into a standard SVM. At this point, we still have to tune the number of hidden neurons. In fact, we experimentally show in Section 5 that this parameter is not critical. It can be set to a large value without significantly altering the classification results, thanks to the regularisation.

5 Experiments

In this section, we study the evolution of the classification results and show that it is sufficient to set the dimensionality p of the randomised feature space to a large value p_s, e.g. 10^3.

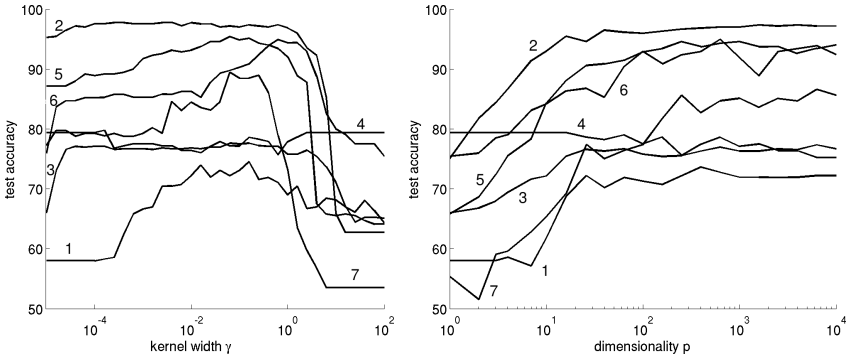
The seven datasets used here are coming from the UCI machine learning repository: Bupa liver disorders (Bupa), Wisconsin diagnostic breast cancer (Cancer), Pima indians diabetes (Diabetes), spectf heart (Heart), Johns Hopkins University ionosphere dataset (Ion), Parkinsons disease (Parkinsons, see [8]) and sonar mines vs rocks (Sonar, see [9]).

For each dataset, we used a double 10-fold cross-validation. Each dataset is split into 10 folds which are alternatively used as test sets for the models built on the 9 remaining folds. These training data are then in turn split into 10 folds which are alternatively used as validation sets for the models built on the 9 remaining folds.

For SVMs using the RBF kernel, C values are $10^{-1:.2:3}$ and γ values are $10.^{-5:.2:2}$. For SVMs using the ELM kernel, C values are $10^{-1:.2:4}$ and p values are $\lceil 10^{0:.2:4} \rceil$. The SVMs have been trained using the LIBSVM library [10].

5.1 Behaviour during optimisation

FIG. 1(a) and FIG. 1(b) show the evolution of the test accuracy obtained by SVM classifiers using RBF and ELM kernels as the kernel width γ and the dimensionality p increase. Using RBF kernels, the accuracy increases, reaches its maximum and then decreases. In contrast, the accuracy with ELM kernels quickly stabilises for each dataset.



(a) (b)

Figure 1: Test accuracy for (a) RBF and (b) ELM kernels, with respect to dimensionality p. The curve labels are given in TAB. 1.

5.2 Classification results

TAB. 1 shows the results obtained with standard SVMs using RBF and ELM kernels, i.e. the test accuracy in % and the computational time in seconds with 95% confidence intervals into parentheses. Each result is provided for the sufficient dimensionality $p_s = 10^3$ and for the optimal dimensionality p^*, which is different for each problem and selected using cross-validation.

The results show that the three models obtain very similar results, except for the training time which is much smaller for our approach when using $p = p_s$. This difference is due to the absence of additional parameters to tune. Moreover, we can see that the choice of the dimensionality p is not significantly critical for the classification performances. In fact, this is due to the regularisation which prevents the SVM from overfitting the data. This regularisation results in a model that efficiently benefits from the high-dimensional information from the ELM kernel.

		RBF kernel		ELM kernel (p^*)	ELM kernel (p_s)	
		Acc.	Comp. time	Acc.	Acc.	Comp. time
1	Bupa	72 (69−76)	140 (139−141)	71 (64−77)	72 (64−80)	5 (5−5)
2	Cancer	97 (95−98)	292 (290−293)	97 (96−98)	97 (96−98)	3 (3−3)
3	Diabetes	76 (73−80)	594 (586−602)	76 (74−78)	76 (74−79)	27 (27−28)
4	Heart	77 (72−82)	100 (98−101)	75(70−80)	77 (72−83)	1 (1−1)
5	Ion	95 (92−98)	169 (168−170)	95 (92−97)	95 (92−97)	1 (1−2)
6	Parkinsons	93 (91−96)	36 (35−36)	92 (88−96)	92 (88−96)	1 (1−1)
7	Sonar	89(83−95)	96 (95−96)	87 (83−90)	85 (83−88)	1 (1−1)

Table 1: Test accuracies and computational times for RBF and ELM kernels.

6 Conclusion

This paper proposes an approach merging both the SVM and ELM framework. Using a kernel defined using the first layer of an ELM, experiments show that the accuracy of SVM classifiers compares to the accuracy obtained with standard RBF kernels. Moreover, it appears that the only parameter of this ELM kernel, i.e. the number of neurons of the ELM, can be set to a large value without altering significantly this accuracy. As a consequence, the computational time is reduced since there are no kernel parameters left to tune.

Possible directions for further work include testing the ELM kernel on more datasets (e.g. biological datasets) and using the ELM kernel for regression.

References

[1] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[2] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001.

[3] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[4] G.-B. Huang, L. Chen, and C.-K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892, 2006.

[5] G.-B. Huang and C.-K. Siew. Extreme learning machine with randomly assigned RBF kernels. *International Journal of Information Technology*, 11(1):16–24, 2005.

[6] Y. Miche, A. Sorjamaa, and A. Lendasse. OP-ELM: Theory, experiments and a toolbox. In *ICANN*, volume 5163 of *Lecture Notes in Computer Science*, pages 145–154. Springer, 2008.

[7] Q. Liu, Q. He, and Z. Shi. Extreme support vector machine classifier. In *PAKDD*, pages 222–233, 2008.

[8] M. A. Little, P. E. McSharry, E. J. Hunter, and L. O. Ramig. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 56(4):1015–1022, 2009.

[9] R. P. Gorman and T. J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[10] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chapter 15

Parameter-Insensitive Kernel in Extreme Learning for Non-linear Support Vector Regression

The following article has been published in Volume 74 (2011) of the Neurocomputing journal. In the line of [9], this paper proposes a new parameter-insensitive kernel inspired from extreme learning for non-linear support vector regression. The kernel is obtained by using an infinite number of neurons in order to compute the similarity between two instances, which is possible since there exist an analytical form. Reprinted with permission from [10].

Neurocomputing 74 (2011) 2526–2531

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Parameter-insensitive kernel in extreme learning for non-linear support vector regression

Benoît Frénay [a,b,*], Michel Verleysen [a]

[a] Machine Learning Group, ICTEAM institute, Université catholique de Louvain, Louvain-la-Neuve, BE 1348, Belgium
[b] Aalto University School of Science and Technology, Department of Information and Computer Science, P.O. Box 15400, FI-00076 Aalto, Finland

ARTICLE INFO

Available online 12 May 2011

Keywords:
Extreme learning machine
Support vector regression
ELM kernel
Infinite number of neurons

ABSTRACT

Support vector regression (SVR) is a state-of-the-art method for regression which uses the ε-sensitive loss and produces sparse models. However, non-linear SVRs are difficult to tune because of the additional kernel parameter. In this paper, a new parameter-insensitive kernel inspired from extreme learning is used for non-linear SVR. Hence, the practitioner has only two meta-parameters to optimise. The proposed approach reduces significantly the computational complexity yet experiments show that it yields performances that are very close from the state-of-the-art. Unlike previous works which rely on Monte-Carlo approximation to estimate the kernel, this work also shows that the proposed kernel has an analytic form which is computationally easier to evaluate.

1. Introduction

In machine learning, regression is a well-known problem which has been thoroughly studied. In the inductive setting, it boils down to making hypotheses on the underlying model, choosing an objective function and then estimating the model parameters. In that process, the squared error loss is often used for mathematical convenience, since it is differentiable. However, the squared error loss leads to non-sparse models. In order to get sparse models, Vapnik proposed to use the ε-sensitive loss [1]. This loss allows data lying within a thick tube at no cost and linearly penalises data outside the tube. Integrating the ε-sensitive loss into a regularised linear model leads to support vector regression (SVR, see e.g. [2–4]). Used in conjunction with kernels, SVRs are powerful non-linear models for regression which have been shown competitive in a wide number of applications.

However, even if SVRs are conceptually appealing, their training is difficultly affordable in practice. Indeed, three meta-parameters have to be tuned for non-linear problems (as detailed in Section 2): the regularisation constant, the tube width for the loss and the kernel parameter. Therefore, one rather uses least-square support vector machines (LS-SVMs, see e.g. [5,6]). However, LS-SVMs lack the sparsity of SVRs, since they relax the quadratic programming problem solved in SVRs by using a squared error loss instead of the ε-sensitive loss.

A similar problem occurs in non-linear classification when using Gaussian kernels. Recently, [7–9] have provided a solution by proposing a new parameter-insensitive kernel inspired from extreme learning. In other words, the choice of the meta-parameter associated to the kernel does not seem to affect the quality of classification. This paper extends this concept to regression. The proposed approach implies the optimisation of only two meta-parameters: the regularisation constant and the tube width. Therefore, the computational cost of non-linear SVR is significantly reduced. Experiments show that the approach yields state-of-the-art performances on various datasets.

The paper is organised as follows. Section 2 reviews SVR in the regression framework. Section 3 shortly introduces the basics of extreme learning. Section 4 derives the new kernel and shows that it has an analytical form under some assumptions. Eventually, the experiments carried in Section 5 show that our computationally cheaper approach achieves state-of-the-art results.

2. Support vector regression

Given a dataset $\mathcal{D}=\{(x_i,t_i)\}_{i\in 1..n}$ where $x\in\Re^d$ and $t\in\Re$ are respectively the inputs and targets, regression consists in building a model $f:\Re^d\to\Re$ which gives a good estimate $y=f(x)$. Usually, models are obtained by minimising an estimate of the expected value of the loss $\mathcal{L}(t,y)$. This estimate is called the empirical risk $\mathcal{R}_{emp}(f)$ and is computed from $\mathcal{D}$. In practice, the empirical risk

* Corresponding author at: Batiment Maxwell, place du Levant, 3, Louvain-la-Neuve, BE 1348, Belgium. Tel.: +32 10 47 81 33; fax: +32 10 47 2598.
E-mail address: benoit.frenay@uclouvain.be (B. Frénay).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.11.037

corresponding to the squared error loss, i.e. the mean squared error

$$\mathcal{R}_{emp}(f)=\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(t_i,y_i)=\frac{1}{n}\sum_{i=1}^{n}(y_i-t_i)^2 \tag{1}$$

is often used because it is mathematically convenient, but it leads to non-sparse models taking into account every single data. Vapnik has developed an alternative loss, called the ε-sensitive loss [1], which leads to sparse models depending only on a small subset of the whole dataset. This section reviews how to embed this loss into a linear model, called support vector regression (SVR, see e.g. [2–4]), which can be extended to deal with non-linear problems.

2.1. Linear support vector regression

The ε-sensitive loss allows the estimate y lying in a thick tube around the observed target t at no cost and penalises it linearly outside the tube, i.e.

$$|y-t|_\varepsilon=\begin{cases}0 & \text{if } |y-t|\leq\varepsilon,\\ |y-t|-\varepsilon & \text{if } |y-t|>\varepsilon,\end{cases} \tag{2}$$

where ε denotes the half-width of the tube, as illustrated in Fig. 1(a). SVRs are linear models which try to find a compromise between the model complexity and the total ε-sensitive loss, i.e.

$$\left|\min_{w,b,\xi_i}\|w\|_2^2+C\sum_{i=1}^{n}(\xi_i^+ +\xi_i^-)\text{s.t.}\begin{cases}\langle w,x_i\rangle+b-t_i\geq\varepsilon+\xi_i^+,\\ t_i-\langle w,x_i\rangle+b\geq\varepsilon+\xi_i^-,\\ \xi_i^+,\xi_i^-\geq 0,\end{cases}\right. \tag{3}$$

where w is the weight vector, b is the bias, C is the regularisation constant and ξ_i^+,ξ_i^- are positive slack variables. The above optimisation problem minimises the norm $\|w\|_2^2$ in order to control the model complexity. Moreover, the objective function is regularised by the ε-sensitive loss: the regularisation constant C determines the compromise between model complexity and errors. Notice that the sum of ξ_i^+ and ξ_i^- is equal to the ε-sensitive loss term for the ith data.

Using the dual form of its Lagrangian, it can be shown [2] that Eq. (3) is equivalent to a quadratic programming problem expressed only in terms of the Lagrange multipliers α_i^+,α_i^- corresponding to the tube constraints in the primal form. The weight vector can then be expressed as

$$w=\sum_{i=1}^{n}(\alpha_i^- -\alpha_i^+)x_i, \tag{4}$$

where only data with different dual variables α_i^+,α_i^- are used to estimate w. These particular data are called *support vectors*. Fig. 1(b) shows a simple example of SVR, where f and the corresponding tube is shown, as well as the three support vectors and two slack variables.

Here, the compromise between model complexity, sparsity and over-fitting is determined by the regularisation constant C. Low values of C correspond to simpler and sparser models, whereas models inferred for large values of C better fit the training data. Moreover, the ε constant controls the width of the tube. Therefore, both the values of C and ε have to be selected simultaneously from training data. For example, a grid search can be done using 10-fold cross-validation (see e.g. [2]).

2.2. Non-linear support vector regression

SVR can be easily extended to handle non-linear regression problems. Indeed, only dot products appear in the dual of Eq. (3). Moreover, the model prediction for a new point is simply

$$f(x)=w\cdot x=\sum_{i=1}^{n}(\alpha_i^- -\alpha_i^+)\langle x_i,x\rangle. \tag{5}$$

Given a non-linear mapping ϕ to a high-dimensional space, the corresponding kernel is defined as

$$k(x,z)=\langle\phi(x),\phi(z)\rangle. \tag{6}$$

Since k computes a dot product, the mapping itself is not necessary. In practice, the Gaussian kernel

$$k(x,z)=\exp(-\gamma\|x-z\|_2^2), \tag{7}$$

which corresponds to an infinite dimensional space is often used and gives good results. However, using SVRs with a Gaussian kernel induces the need to tune three meta-parameters: the regularisation constant C, the tube half-width ε and the kernel scale γ. Using 10-fold cross-validation and only 10 possible values for each meta-parameter, not less than 10^4 SVRs have to be trained. The rest of this paper shows that it is possible to obtain results of the same quality by using a new parameter-insensitive kernel and therefore to reduce strongly the number of SVRs to be trained.

3. Extreme learning

This section introduces extreme learning, a recent trend in machine learning which aims at producing fast, but accurate models. The next section shows how to extend it to kernel-based machines.

3.1. Extreme learning machines

In [10,11], Huang et al. propose a new way to optimise networks with a single hidden layer of units that they call extreme learning. Firstly, a large hidden layer is created with random weights and biases, e.g. picked up uniformly in $[-3,3]$. Then, the output weights and bias are optimised by solving a linear system. As opposed to back-propagation [12], this approach does not get stuck in local minima and is very fast.

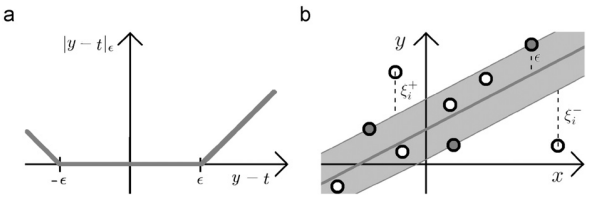


Fig. 1. (a) ε-sensitive loss and (b) example of SVR trained on a linear problem with two outliers.

More formally, let us define W and w being respectively the hidden and output weights. Moreover, we choose an activation function σ for hidden units and define X as the matrix whose rows are data. An additional column of ones is added to X to embed the biases directly in W, resulting in X_b. Therefore, the activation of hidden units corresponding to the data is simply

$$H = \sigma(X_b W). \tag{8}$$

Other types of hidden units can be used, e.g. radial basis function (RBF) hidden units [13–15]. If hidden weights are held fixed and the output activation is linear, the output weights are given by

$$\underset{w}{\operatorname{argmin}} \mathcal{L}(T, H_b w), \tag{9}$$

where $\mathcal{L}$ is the loss and H_b is the matrix H with an additional column of ones added in order to embed the bias in w. Using the squared error loss, the above equation becomes

$$\underset{w}{\operatorname{argmin}} \|H_b w - T\|^2. \tag{10}$$

The resulting model is called an extreme learning machine (ELM) and can be trained easily at a very low computational cost. Indeed, Eq. (10) is simply a linear regression whose solution is

$$w = H_b^{\dagger} T, \tag{11}$$

where $H_b^{\dagger} = (H_b' H_b)^{-1} H_b'$ is the Moore–Penrose pseudo-inverse of the complete activation matrix H_b. Huang et al. [10,11] have shown that, using the above setting, it is possible to solve regression problems very fast and yet to obtain satisfying results. Huang et al. [16] have also shown that the extreme learning machines (ELMs) are universal approximators and can be built incrementally, starting with one unit in the network.

3.2. Algorithms to train extreme learning machines

The number of units of an ELM cannot be set arbitrarily. For example, let us consider the approximation of a simple sine function, using 20 noisy training samples. Fig. 2(a) shows the mean squared error on an independent test set for different ELM sizes. Fig. 2(b) shows the resulting approximations with 2, 9 and 20 units, where 9 units is the optimal ELM size in Fig. 2(a). Here, choosing too many or not enough units leads respectively to overfitting or underfitting.

To our knowledge, three approaches have been proposed so far to choose the number of units in ELMs. First, Huang et al. [10,11,16] propose to build networks with different sizes, either incrementally or not, and to pick the best size using e.g. cross-validation methods. Second, Miche et al. [17,18] propose the OP-ELM framework which uses the LARS/LASSO to compute a L-1 regularisation path for the output weights, then pick up the set of units with the lowest LOO error and eventually retrain the model. Third, Liu et al. [7] use L-2 regularisation, i.e. Ridge regression.

These algorithms do not produce sparse models since they are essentially solving a linear regression with a squared error loss. Notice that OP-ELM is sparse in terms of units, not in terms of data. Indeed, only a few units are selected using L-1 regularisation, but output weights are computed from all the data. The next section shows how to introduce the ε-sensitive loss in ELMs in order to obtain sparse models.

4. Extreme learning with kernel-based models

Recently, several papers [7–9] have emphasised the similarities between ELMs and kernel-based methods. This section reviews ELM from the kernel point of view and shows how to create a new family of kernels inspired from extreme learning. An analytical expression is derived for one of these kernels and it is shown in Section 5 that the kernel parameter does not need to be tuned.

4.1. A change of perspective: the ELM kernel

In standard ELMs, the role of the hidden layer is to map the input to the hidden units. In other words, the first, hidden layer of the ELM is mapping data from the data space to some higher dimensional space, where each dimension corresponds to a hidden unit. Hence, this new high-dimensional space can be viewed as a feature space where the ELM just solves a linear regression. This new interpretation leads to the definition of the so-called ELM kernel.

Let us define the mapping $\phi : \Re^d \to \Re^p$ such that the ith component of $\phi(x)$ is equal to the activation of the ith hidden unit. Given two data points x and z and an ELM with p units, the corresponding ELM kernel function is defined as

$$k(x, z|p) = \frac{1}{p} \langle \phi(x), \phi(z) \rangle, \tag{12}$$

which is simply the dot product in the feature space, normalised by the number of hidden units p. By construction, k is a valid kernel and can be used by any kind of kernel-based method. Indeed, it corresponds to the dot product of x and z in the feature space defined by the mapping ϕ, i.e. the hidden layer of an ELM.

The feature space associated to the ELM kernel is built randomly, since each dimension corresponds to a hidden unit

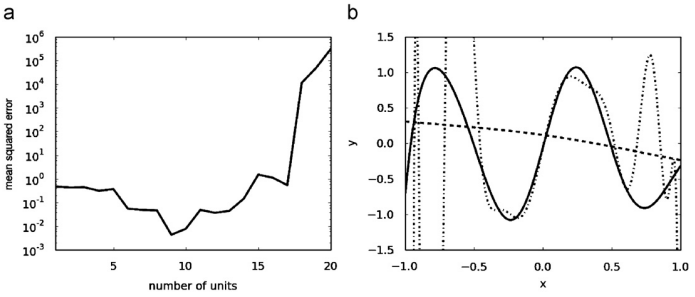


Fig. 2. Sine function approximation from 20 noisy data using ELMs: (a) evolution of the mean squared error computed on an independent test set of 10^3 samples as the number of units increases and (b) models obtained with 2 units (dashed line), 9 units (plain line) and 20 units (dotted line).

with random weights and bias. However, each data is mapped to the same feature space using the hidden layer of the same ELM.

4.2. Limit case and analytical form of the ELM kernel

It has been shown in [8,9] that the number of units used to evaluate the ELM kernel does not matter for classification tasks, as long as it is large enough. Let us assume a network with a single hidden layer of units, whose hidden layer weights are picked randomly from a given prior. If we let the number of units p grow to infinity, the ELM kernel becomes

$$\lim_{p\to+\infty} k(x,z|p) = \lim_{p\to+\infty} \frac{1}{p}\langle \phi(x),\phi(z)\rangle = \lim_{p\to+\infty} \frac{1}{p}\sum_{i=1}^{p} \sigma(w_i x)\sigma(w_i z) = \mathbb{E}_w[\sigma(wx)\sigma(wz)]. \quad (13)$$

In other words, the limit $k(x,z|p\to+\infty)$ can be interpreted as the expected value of $\sigma(wx)\sigma(wz)$, i.e. the covariance between the activations of a random hidden unit alternatively fed with x and z. In the rest of this paper, $k(\cdot,\cdot|p\to+\infty)$ is referred to as the *asymptotic ELM kernel*.

Let us consider a particular case where (i) the weights and biases of the hidden layer are picked randomly from an isotropic Gaussian distribution with variance σ_w^2 and (ii) the activation function is the sigmoid *erf* function defined as

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)\,dt. \quad (14)$$

It can be shown [19] that the asymptotic ELM kernel admits the following analytical expression:

$$k(x,z|p\to+\infty) = \frac{2}{\pi}\arcsin\frac{1+\langle x,z\rangle}{\sqrt{\left(\frac{1}{2\sigma_w^2}+1+\langle x,x\rangle\right)\left(\frac{1}{2\sigma_w^2}+1+\langle z,z\rangle\right)}}. \quad (15)$$

Since an analytical expression is available, it is no longer necessary to actually build ELMs in order to implement this kernel. Therefore, the implementation is straightforward. In the rest of this paper, we use the normalised version of the asymptotic ELM kernel

$$\tilde{k}(x,z|p\to+\infty) = \frac{k(x,z|p\to+\infty)}{\sqrt{k(x,x|p\to+\infty)k(z,z|p\to+\infty)}} = \frac{\mathbb{E}_w[\sigma(wx)\sigma(wz)]}{\sqrt{\mathbb{E}_w[\sigma(wx)\sigma(wx)]\mathbb{E}_w[\sigma(wz)\sigma(wz)]}}, \quad (16)$$

which can be interpreted as the correlation between the activations of a random hidden unit alternatively fed with x and z. Hence, we have $\tilde{k}(x,x|p\to+\infty)=1$. Fig. 3 shows the shape of the resulting kernel evaluated around zero for different values of σ_w. Even if the shape itself changes, the scale of the kernel is constant and does not depend on σ_w. Section 5 shows that the σ_w parameter does not affect the results obtained when using SVR with this kernel, if it is chosen large enough.

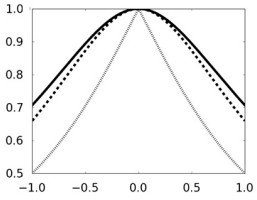


Fig. 3. Normalised asymptotic ELM kernel evaluated around zero for different values of σ_w: $\sigma_w = 10^{-3}$ (plain line), $\sigma_w = 1$ (dashed line) and $\sigma_w = 10^3$ (dotted line).

5. Experiments

This section aims to answer two questions: (i) is it necessary to tune the σ_w parameter for the normalised asymptotic ELM kernel and (ii) does SVR get state-of-the-art results with that kernel?

5.1. Experimental setting

SVR results are obtained using the LIBSVM library [20] and several datasets of various sizes coming from the UCI machine learning repository [21] (Abalone, Cancer and Machine CPU) and from [22] (Ailerons, CompActs, Elevators, Stock and Triazines); see Table 1 for details. The datasets are separated into two groups: small datasets (several hundred of instances) and large datasets (several thousands of instances).

For each dataset, we used 10-fold cross-test (see Fig. 4). Each dataset is split into 10 folds which are alternatively used as independent test sets for the models built on the nine remaining folds. These training data are then in turn split into 10 folds which are alternatively used as validation sets for the models built on the nine remaining folds. Validation sets are used to select the best values for the meta-parameters. The criterion used here to compare models is the mean squared error (MSE).

The standard Gaussian kernel and the normalised asymptotic ELM kernel are used respectively with values in a logarithmic scale from 10^{-3} to 10^{0} for γ and from 10^{-3} to 10^{3} for σ_w. The other meta-parameter values are respectively taken logarithmically from 10^{-2} to 10^{6} for C and from 10^{-5} to 10^{1} for ε. For the meta-parameters γ, C and ε, each logarithmic step consists in multiplying their value by $\sqrt{10}$: C takes e.g. 17 different values. For each trained SVR, the inputs and targets are normalised using the mean and standard deviations computed from the training data. However, the differences between true targets and predictions are multiplied afterwards by the standard deviation in order to be expressed in terms of original units.

Table 1
Detailed list of datasets used for experiments, ordered by size and split into two groups.

Short name	Group	Size	Dimensionality	Full name
Triazines	Small	186	60	Inhibition of dihydrofolate reductase by triazines
Cancer		192	32	Wisconsin prognostic breast cancer
CPU		209	6	Relative CPU performance data
Stock		950	9	Daily stock prices dataset
Abalone	Large	4177	8	Abalone data
Ailerons		7129	5	Delta ailerons control
CompActs		8192	21	Computer activity database
Elevators		9517	6	Delta elevators control

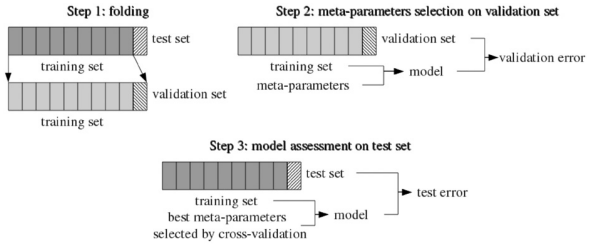


Fig. 4. Illustration of the cross-test method.

Table 2
Means and 95% confidence intervals for the test MSE obtained on small datasets using SVR with (i) the Gaussian kernel and (ii) the normalised asymptotic ELM kernel for different σ_w. Results which are not significantly different from the best result (underlined) are in bold font.

	Gaussian kernel	Normalised asymptotic ELM kernel						
		$\sigma_w = 1e{-}3$	$\sigma_w = 1e{-}2$	$\sigma_w = 1e{-}1$	$\sigma_w = 1e{+}0$	$\sigma_w = 1e{+}1$	$\sigma_w = 1e{+}2$	$\sigma_w = 1e{+}3$
Cancer	**1.1e+3**	**1.0e+3**	**1.0e+3**	**1.1e+3**	**1.0e+3**	**1.0e+3**	**1.1e+3**	**1.0e+3**
	[8.6e+2, 1.3e+3]	[8.9e+2, 1.2e+3]	[8.9e+2, 1.2e+3]	[9.4e+2, 1.2e+3]	[8.7e+2, 1.2e+3]	[8.7e+2, 1.2e+3]	[9.3e+2, 1.2e+3]	[8.6e+2, 1.2e+3]
CPU	**4.5e+3**	1.3e+4	1.0e+4	**3.0e+3**	**3.9e+3**	**3.9e+3**	**4.1e+3**	**4.1e+3**
	[−1.0e+3, 1.0e+4]	[3.9e+3, 2.2e+4]	[3.5e+3, 1.8e+4]	[1.4e+3, 4.6e+3]	[1.1e+3, 6.7e+3]	[1.1e+3, 6.8e+3]	[1.1e+3, 7.1e+3]	[1.2e+3, 7.0e+3]
Stock	4.2e−1	5.0e+0	4.8e+0	6.8e−1	4.3e−1	**3.5e−1**	**3.6e−1**	**3.8e−1**
	[3.7e−1, 4.8e−1]	[4.3e+0, 5.7e+0]	[4.5e+0, 5.2e+0]	[6.1e−1, 7.6e−1]	[3.8e−1, 4.9e−1]	[3.2e−1, 3.8e−1]	[3.2e−1, 4.0e−1]	[3.5e−1, 4.2e−1]
Triazines	3.0e−2	**2.2e−2**	**2.2e−2**	**2.2e−2**	**2.1e−2**	**2.1e−2**	**2.0e−2**	**2.0e−2**
	[1.4e−2, 4.7e−2]	[1.8e−2, 2.7e−2]	[1.5e−2, 2.9e−2]	[1.6e−2, 2.8e−2]	[1.3e−2, 2.9e−2]	[1.5e−2, 2.7e−2]	[1.4e−2, 2.6e−2]	[1.4e−2, 2.6e−2]

Table 3
Means and 95% confidence intervals for the test MSE obtained on large datasets using SVR with (i) the Gaussian kernel and (ii) the normalised asymptotic ELM kernel for different σ_w. Results which are not significantly different from the best result (underlined) are in bold font.

	Gaussian kernel	Normalised asymptotic ELM kernel						
		$\sigma_w = 1e{-}3$	$\sigma_w = 1e{-}2$	$\sigma_w = 1e{-}1$	$\sigma_w = 1e{+}0$	$\sigma_w = 1e{+}1$	$\sigma_w = 1e{+}2$	$\sigma_w = 1e{+}3$
Abalone	**4.4e+0**	5.2e+0	5.0e+0	**4.5e+0**	**4.4e+0**	**4.4e+0**	**4.4e+0**	**4.4e+0**
	[4.1e+0, 4.7e+0]	[4.8e+0, 5.6e+0]	[4.6e+0, 5.4e+0]	[4.2e+0, 4.7e+0]	[4.2e+0, 4.6e+0]	[4.0e+0, 4.8e+0]	[4.1e+0, 4.7e+0]	[4.2e+0, 4.7e+0]
Ailerons	**2.7e−8**	3.6e−8	3.6e−8	**2.7e−8**	**2.8e−8**	**2.8e−8**	**2.8e−8**	**2.8e−8**
	[2.6e−8, 2.9e−8]	[3.4e−8, 3.9e−8]	[3.4e−8, 3.8e−8]	[2.5e−8, 2.9e−8]	[2.6e−8, 3.1e−8]	[2.7e−8, 2.9e−8]	[2.6e−8, 3.0e−8]	[2.5e−8, 3.1e−8]
CompActs	**8.8e+0**	5.9e+1	5.2e+1	**8.9e+0**	1.2e+1	1.2e+1	1.2e+1	1.2e+1
	[8.3e+0, 9.3e+0]	[5.2e+1, 6.7e+1]	[4.6e+1, 5.8e+1]	[8.0e+0, 9.8e+0]	[1e+1, 1.4e+1]	[1.0e+1, 1.3e+1]	[1.0e+1, 1.3e+1]	[1.0e+1, 1.3e+1]
Elevators	**2.0e−6**	2.2e−6	2.2e-6	**2.0e−6**	**2.0e−6**	**2.0e−6**	**2.0e−6**	**2.0e−6**
	[2.0e−6, 2.1e−6]	[2.1e−6, 2.2e−6]	[2.1e−6, 2.3e−6]	[1.9e−6, 2.1e−6]	[2.0e−6, 2.1e−6]	[1.9e−6, 2.1e−6]	[1.9e−6, 2.1e−6]	[1.9e−6, 2.1e−6]

5.2. Results on real datasets

Tables 2 and 3 show the results obtained on the small and large datasets, respectively. For each dataset and kernel, the mean and 95% confidence interval of the MSE computed using 10-fold cross-test are given. For each dataset, the MSE is expressed in terms of original units. Best results are underlined and results which are not significantly different are in bold font. Here, a result is significantly different from the best result if its mean does not belong to the confidence interval of the best result. The best results are comparable to the results obtained in [18], except for Triazines which is not used in that work.

Tables 2 and 3 show that the results obtained with different values of σ_w are very similar. For six out of the eight datasets (Cancer, CPU, Triazines, Abalone, Ailerons and Elevators), the results are not significantly different when σ_w is equal to or larger than 10^{-1}. Moreover, these results are not significantly different from the results obtained using the standard Gaussian kernel, except for Triazines where the Gaussian kernel result is slightly worst. For CompActs, the result obtained with $\sigma_w = 10^{-1}$ is still not significantly different from the result obtained using the standard Gaussian kernel. Moreover, the results obtained using larger values of σ_w are only slightly worst. For Stock, the results obtained with $\sigma_w \geq 10$ are not significantly different from the best result. Moreover, the result obtained using the standard Gaussian kernel is slightly worst.

According to Tables 2 and 3, it seems that using the normalised asymptotic ELM kernel with e.g. $\sigma_w = 1$ or 10 allows obtaining results which are close if not identical to the results obtained using the Gaussian kernel. It means that the proposed kernel is in fact parameter-insensitive, in the sense that it is not necessary to tune the parameter σ_w to obtain good results. In practice, this observation reduces the number of meta-parameters from three to two: the regularisation constant C and the tube half-width ε.

Therefore, the proposed approach speeds up the meta-parameter optimisation step, so that non-linear problems can be tackled at the same computational cost than linear problems.

6. Conclusion

This paper shows that the ELM kernel can be successfully used for support vector regression. Using this kernel with SVR is in fact equivalent to optimising extreme learning machines using the ε-sensitive loss. Moreover, this paper proposes a new asymptotic view for the ELM kernel when the number of units grows to infinity. In that limit case, this paper also shows that the proposed kernel has an analytical form under certain assumptions on the hidden units of the extreme learning machine.

Experimental results suggest that the performances do not depend strongly on the only parameter of the ELM kernel. Indeed, performances which are close if not identical to the state-of-the-art performances are obtained as soon as the kernel parameter is chosen large enough, e.g. $\sigma_w = 1$ or 10. Therefore, the proposed approach reduces the number of meta-parameters to be tuned from three to two. As a matter of fact, the computational cost of non-linear SVR is strongly reduced with almost no consequence on the obtained performances.

Acknowledgements

The authors would like to thank Prof. Schrauwen (Ghent University, Belgium) for his useful remarks concerning the ELM kernel convergence, Mr. de Lannoy for his useful comments on this paper and Mr. Durvaux from BELNET-BEgrid for his precious technical help to get the experimental results on time. The authors also used the UCL CISM in order to get additional experimental results and would like to thank Mr. Van Renterghem for his technical help. Benoît Frénay would like to thank the FNRS which supported him during his stay at the Altoo University.

References

[1] V. Vapnik, The Nature of Statistical Learning, Springer, New York, 1995.
[2] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (3) (2004) 199–222.
[3] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[4] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, The MIT Press, 2001.
[5] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters 9 (3) (1999) 293–300.
[6] J.A.K. Suykens, T.V. Gestel, J.D. Brabanter, B.D. Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific Publishing Co., Singapore, 2002.
[7] Q. Liu, Q. He, Z. Shi, Extreme support vector machine classifier, in: PAKDD, Lecture Notes in Computer Science2008, pp. 222–233.
[8] B. Frénay, M. Verleysen, Using SVMs with randomised feature spaces: an extreme learning approach, in: Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN)2010, pp. 315–320.
[9] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing, in Press, Corrected Proof.
[10] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[11] G.-B. Huang, C.-K. Siew, Extreme learning machine with randomly assigned RBF kernels, International Journal of Information Technology 11 (1) (2005) 16–24.
[12] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, 1998.
[13] G.-B. Huang, C.-K. Siew, Extreme learning machine with randomly assigned RBF kernels, International Journal of Information Technology 11 (1) (2005) 16–24.
[14] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16–18) (2007) 3056–3062.
[15] G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (16–18) (2008) 3460–3468.
[16] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Transactions on Neural Networks 17 (4) (2006) 879–892.
[17] Y. Miche, A. Sorjamaa, A. Lendasse, OP-ELM: theory, experiments and a toolbox, ICANN, Lecture Notes in Computer Science, vol. 5163, Springer, 2008, pp. 145–154.
[18] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally-pruned extreme learning machine, IEEE Transactions on Neural Networks 21 (1) (2010) 158–162.
[19] C. Williams, Computing with infinite networks, in: Advances in Neural Information Processing Systems, MIT Press, 1996, pp. 295–301.
[20] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at 〈http://www.csie.ntu.edu.tw/∼cjlin/libsvm〉.
[21] UCI machine learning repository. 〈http://archive.ics.uci.edu/ml/datasets.html〉.
[22] 〈http://www.liaad.up.pt/∼ltorgo/Regression/DataSets.html〉.

Benoît Frénay received the Engineer's degree from the Université catholique de Louvain (UCL), Belgium, in 2007. He is now Ph.D. student at the UCL Machine Learning Group. His main research interests in machine learning include support vector machines, extreme learning, graphical models, classification, data clustering, probability density estimation and label noise.

Michel Verleysen was born in 1965 in Belgium. He received the M.S. and the Ph.D. degrees in Electrical Engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He was an Invited Professor at the Swiss Ecole Polytechnique Fédérale de Lausanne (E.P.F.L.), Switzerland, in 1992, at the Université d'Evry Val d'Essonne (France) in 2001, and at the Université Paris 1—Panthéon-Sorbonne in 2002–2004. He is a former Research Director with the Belgian FNRS (Fonds National de la Recherche Scientifique) and a Professor at the Université catholique de Louvain. He is Editor-in-Chief of the Neural Processing Letters journal, Chairman of the Annual European Symposium on Artificial Neural Networks (ESANN) Conference, Associate Editor of the IEEE Transactions on Neural Networks journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He is the author or the co-author of about 200 scientific papers in international journals and books or communications to conferences with reviewing committee. He is the co-author of the scientific popularisation book on artificial neural networks in the series "Que Sais-Je?," in French. His research interests include machine learning, artificial neural networks, self-organisation, timeseries forecasting, non-linear statistics, adaptive signal processing, and high-dimensional data analysis.