UNIVERSIDADE DA CORUÑA

FACULTADE DE INFORMÁTICA

Departamento de Computación

PH.D. THESIS

Novel Machine Learning Methods Based on Information Theory

Author: Iago Porto Díaz

Advisors: Amparo Alonso Betanzos

Oscar Fontenla Romero

A Coruña, September 2015

September 21, 2015
UNIVERSIDADE DA CORUÑA

FACULTADE DE INFORMÁTICA
Campus de Elviña s/n
15071 - A Coruña (Spain)

Copyright notice:

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording and/or otherwise without the prior permission of the authors.

Acknowledgements

To Amparo and Óscar, for your advice and orientation.

To my girlfriend Iria, my parents José Luis and Sara, and my brother David, for your support and dedication.

To past and present LIDIA's members, for the good atmosphere and comradeship, for your advice, and for everything I have learned in all these unforgettable years I have spent with you guys.

Iago Porto Díaz
September 2015

Resumo

A aprendizaxe automática é a área da intelixencia artificial e da ciencia da computación que estuda algoritmos que aprenden a partir de datos, fan predicciónS e producen comportamentos baseados en exemplos. Esta tesis desenvolve novos métodos de aprendizaxe automática baseados en teoría da información (TI) e en *information theoretic learning* (ITL): (1) En primeiro lugar, utilízase TI para selección de características. Específicamente, se desenvolven dous novos algoritmos. O primeiro ten en conta o coste (computacional, económico, etc.) de cada característica —ademais da relevancia—. O segundo fai uso do concepto de *ensemble*, moi común en escenarios de clasificación, pero moi pouco explorado na literatura de selección de características. (2) En segundo lugar, se poden empregar conceptos de TI e ITL como unha función de erro alternativa, o cal permite a exploración doutro campo da literatura non moi estudado: a aproximación de modelado local. Específicamente, desenvólvese un novo algoritmo para clasificación. Este algoritmo está baseado na combinación de redes de neuronas por medio de modelado local e técnicas baseadas en ITL.

Resumen

El aprendizaje automático es el área de la inteligencia artificial y la ciencia de la computación que estudia los algoritmos que aprenden a partir de datos, realizan predicciones y producen comportamientos basados en ejemplos. Esta tesis desarrolla nuevos métodos de aprendizaje automático basados en teoría de la información (TI) y en *information theoretic learning* (ITL): (1) En primer lugar, se utiliza TI para selección de características. Específicamente, se desarrollan dos nuevos algoritmos. El primero tiene en cuenta el coste (computacional, económico, etc.) de cada característica —además de la relevancia—. El segundo hace uso del concepto de *ensemble*, muy común en escenarios de clasificación, pero muy poco explorado en la literatura de selección de características. (2) En segundo lugar, se pueden emplear conceptos de TI e ITL como una función de error alternativa, lo cual permite la exploración de otro campo de la literatura no muy estudiado: la aproximación de modelado local. Específicamente, se desarrolla un nuevo algoritmo para clasificación. Este algoritmo está basado en la combinación de redes de neuronas por medio de modelado local y técnicas basadas en ITL.

Abstract

Machine learning is the area of artificial intelligence and computer science that studies algorithms that can learn from data, make predictions, and produce behaviors based on examples. This thesis develops new methods of machine learning based on information theory (IT) and information theoretic learning (ITL): (1) On the one hand, IT is used for feature selection. Specifically, two new algorithms are developed. The first one takes into account the cost (computational, economic, etc.) of each feature —besides its relevance—. The second one makes use of the concept of ensemble, quite common for classification scenarios, but very little explored in the literature of feature selection. (2) On the other hand, IT and ITL concepts can be employed as an alternative error function, thus allowing the exploration of another not very well studied field in the literature: the local modeling approach. Specifically, a new algorithm for classification is developed. This algorithm is based on the combination of neural networks by means of local modeling and techniques based on ITL.

Contents

List of figures

List of tables

CHAPTER 1

Introduction

Machine learning is the area of artificial intelligence and computer science that studies algorithms that can learn from data, make predictions, and develop behaviors based on examples. The main types of problems machine learning can solve are [15]: (a) classification, where the algorithm must assign unseen inputs to a series of classes; (b) regression, where the focus is predicting a continuous output; (c) clustering, where inputs must be labeled into unknown groups, unlike classification; (d) density estimation, where the goal is finding the distribution of a set of inputs; and (e) dimensionality reduction, where inputs are simplified by mapping them to lower dimensional spaces. These tasks can also be classified, according to the nature of available learning data, in (a) supervised learning, where a set of known patterns are used for training; (b) unsupervised learning, where the objective is to unravel the underlying similarities between data; and (c) reinforcement learning, where the environment provides information about the goodness of the learning.

Supervised classification, the problem in which this thesis is focused, is an area of artificial intelligence concerned with the classification of observations. The objective is to classify data based on a priori knowledge. This knowledge is utilized to learn predictive models from a data set of examples in order to classify unseen instances. Specifically, supervised classification assumes previous knowledge of the class —the value to be predicted— of the instances of the data set. One important aspect of supervised classification is the evaluation of the algorithms by means of an evaluation function. It usually quantifies the generalization ability of the classifier. One of the most important evaluation functions is the classification error, which provides the probability of misclassifying an instance. In real world problems, the true classification error is unknown, and so is its underlying probability distribution. Therefore, it must be estimated from data. In particular, he mean squared error (MSE) is the measure that is typically utilized for evaluating the estimations made by the algorithms. The MSE is the second-order moment of the error, and therefore, it incorporates both the variance and the bias of the estimator. However, the use of evaluation functions based on second-order moments suffers from the limitation of the inherent Gaussian hypothesis. In this dissertation, this impediment is avoided by using a computationally-efficient model, based on information-theoretic descriptors of entropy, di-

vergence and mutual information, combined with non-parametric PDF estimators. This brings robustness and generality to the evaluation function. This model is called Information Theoretic Learning (ITL) [115]. As entropy is defined as the uncertainty of a random variable, it is natural to use it as a tool for applications where the data are incomplete or noisy.

A key aspect for a correct model construction is data preprocessing, which aim is to prepare the data properly to serve as input for learning algorithms. Learning algorithms usually suffer from overfitting (loss of generality) and efficiency problems. Dimensionality reduction techniques are a family of data preprocessing methods that can be applied to reduce the dimensionality of data and improve the performance of learning algorithms. There are two types of dimensionality reduction techniques: feature extraction and feature selection [158].

Feature extraction techniques take the set of features of the original learning data set and build derived features, with the aim of improving the subsequent learning process. In this way, the generated set of features is usually more compact and has more discriminating power. It is widely used in applications such as image analysis, signal processing, and information retrieval. As there is a loss of interpretability (because of the derivation of features), it is more interesting for applications where model accuracy is more important than model understandability.

On the other hand, feature selection removes irrelevant and redundant features, which increases the predictive accuracy of the model learned, reducing the cost of data, improving learning efficiency by reducing storage requirements and computational costs, reducing the complexity and improving the understanding of the resulting model. It is widely used in data mining applications, such as text mining, genetics analysis, and sensor data processing. Unalike feature extraction, feature selection maintains the original features. Therefore, it is useful for applications where the interpretability of the model is important, such as in knowledge extraction. There exists a large amount of feature selection algorithms, some of which are based on information theoretic (IT) principles.

The use of IT and ITL in this thesis is twofold:

- On the one hand, IT is used for the feature selection step. Specifically, two new algorithms are developed. The first one takes into account the cost (computational, economic, etc.) of each feature —besides its relevance—. This fact is important due to the possibility of obtaining similar or better performances while reducing the associated cost. The second algorithm makes use of the concept of ensemble, quite common for classification scenarios, but very little explored in the literature of feature selection. In this case, the

aim is obtaining more stable results than using a single feature selection method and also improving the computational efficiency of the training process by means of distributed computing.

- On the other hand, IT and ITL concepts can be employed as an alternative error function, thus allowing the exploration of another not very well studied field in the literature: the local modeling approach. Specifically, a new algorithm for classification is developed. This algorithm is based on the combination of neural networks by means of local modeling and techniques based on ITL.

1.1 Objectives

In this doctoral thesis, the learning algorithms used are of the supervised type. In this context, selecting an appropriate cost function is a non-trivial problem, where the conflict between parametric and non-parametric modeling appears. The classic mean squared error (MSE) captures all the information of the probability density function (PDF) of the error under normality hypothesis. It provides analytical solutions for lineal model optimization, providing with optimality and ease of implementation.

However, MSE is often utilized in situations where the classifiers are non-linear and the errors are not normally distributed. With this end, the exploration of several possibilities based in the scope of information theory and statistics is posed. Combining non-parametric estimators of PDF with descriptors of information theory's entropy and mutual information, the goal of moving away from the traditional approach of using second-order moments of error is achieved. In this manner, the limitations of the MSE's inherent normality are avoided. Those estimators provide with robust and general cost functions which improve the performance in realistic scenarios. The challenge, therefore, consists on demonstrating that these new learning models can improve the results obtained by current systems in certain circumstances or scenarios.

The thesis is divided in three main parts. The objectives for each of the parts are described as follows:

1. Cost-based feature selection.

    - Solve problems where not only it is interesting to minimize the classification error, but also to reduce costs that may be associated to input features.

- Obtain a trade-off between a feature selection metric and the cost associated to the features, in order to select relevant features with a low associated cost, while keeping the classification accuracy.

2. Ensemble learning for feature selection.

   - Combine ordered rankings of features which are obtained from base selectors.
   - Achieve an improvement in the overall computational performance of the feature selection process, while maintaining the classification accuracy.
   - Release the user from the task of deciding which feature selection method is the most appropriate, while maintaining the classification accuracy.

3. Local classification based on ITL.

   - Build complex classification models for two-class and multiclass problems. Those models are composed of several simpler neural network sub-models.
   - Achieve an improvement of classification performance on real problems.

The rest of this dissertation is organized as follows. Chapter 2 introduces the domain and precedents of this research. Chapter 3 describes a new cost-based feature selection method. Chapter 4 introduces a new ensemble method for feature selection, based in ranking learning. Chapter 5 presents a new classification method based on the combination of neural networks by means of Information Theoretic Learning tools. Finally, Chapter 6 summarizes the obtained contributions and conclusions and the produced publications.

CHAPTER 2

Machine Learning Methods Based on Information Theory

This chapter presents the basis of this thesis. It commences with a description of the most basic foundations, which are Information Theory (IT) —Section 2.1— and Information Theoretic Learning (ITL) —Section 2.2—, and follows with a description of some relevant developments of these two areas on machine learning, specifically in feature selection and classification —Section 2.3—.

2.1 Information Theory

Information theory (IT) is an area of computer science and electrical engineering that deals with quantification of information. It was formulated by Claude E. Shannon in 1948 [129]. Initially, the objectives of this theory were to represent, transmit and store data compactly and reliably. Since then, applications have been found in other fields like neurobiology [122], natural language processing, statistical inference, and machine learning, the latter being the field of interest for this doctoral thesis. The connection between information theory and machine learning comes from the fact that representing data in a compact fashion requires assigning short words to highly usual bit strings, and longer words to less likely bit strings. Moreover, transmitting information over noisy channels requires a good model for the messages. Ultimately, a model to predict which data are likely and which are unlikely is needed, which is a central issue in machine learning.

Next, a series of central concepts of IT —which are used in the algorithms proposed in this dissertation— are defined.

Entropy An important measure of information is entropy, which is the average number of bits

needed to store or communicate one symbol in a message. The entropy of a random variable X with distribution p, denoted by $H(X)$ or $H(p)$ is a measure of its uncertainty. For a discrete variable with K possible values, it is defined by:

$$H(X) = -\sum_{k=1}^{K} p(X=k)\log_2 p(X=k) \tag{2.1}$$

log_2 is used when using binary digits. An important property of entropy is that it is maximum when all the messages are equiprobable: $p(X=k) = 1/K$ and $H(X) = \log_2 K$.

Kullback-Leibler divergence Another important measure of information theory is the Kullback-Leibler divergence (KL divergence) [81], information gain, or relative entropy. It is used to measure the dissimilitude between two probability distributions. It is defined by:

$$KL(p||q) = \sum_{k=1}^{K} p_k \log \frac{p_k}{q_k} \tag{2.2}$$

This can be rewritten as:

$$KL(p||q) = \sum_{k=1}^{K} p_k \log p_k - \sum_{k=1}^{K} p_k \log q_k = -H(p) + H(p,q) \tag{2.3}$$

where $H(p,q)$ is called the cross entropy:

$$H(p,q) = -\sum_{k=1}^{K} p_k \log q_k \tag{2.4}$$

Cross entropy The cross entropy is the average number of bits needed to identify an event drawn from an underlying set of events with "true" distribution p, when using model q. Moreover, the entropy $H(p)$ is the expected number of bits if the true model is used. So, as displayed in (2.3), the KL divergence is the difference between these two. Alternatively, the KL divergence is the average number of additional bits needed, due to using distribution q instead of the true distribution p. This interpretation denotes that $KL(p||q) \geq 0$ and $KL(p||q) = 0$ if $q = p$.

Mutual information In order to define mutual information, which is the next quantity of information to be described, let us consider two random discrete variables, X and Y. In order to know how much information can be obtained from one of the variables by observing the other, it can be determined how similar the joint distribution $p(X,Y)$ is to the factored distribution $p(X)p(Y)$. This is called the mutual information (MI) and is defined as follows:

$$I(X;Y) = KL\left(p(X,Y)\,||\,p(X)\,p(Y)\right) = \sum_{x}\sum_{y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{2.5}$$

where $I(X;Y) \geq 0$ and $I(X;Y) = 0$ if $p(X,Y) = p(X)p(Y)$, that is, the MI is 0 if the variables are independent.

A basic property of the MI is:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{2.6}$$

where $H(Y|X)$ is the conditional entropy, defined as:

$$H(Y|X) = \sum_x p(x)H(Y|X = x). \tag{2.7}$$

Therefore, the MI between X and Y can be interpreted as the reduction in uncertainty about X after observing Y (the opposite is true by symmetry).

2.2 Information Theoretic Learning

How to best extract the information contained in data is a common problem nowadays. We are surrounded by huge amounts of data, which hide the information needed to answer a myriad of questions that the data processing professionals have. The use of computers and the World Wide Web has increased dramatically the accessibility and the amount of data generated. Information Theoretic Learning (ITL) [115] is a framework that utilizes the information theory descriptors of entropy and divergence as non-parametric cost functions for the design of adaptive systems in unsupervised or supervised training models. Data modeling is a process to extract information from data. A model of the data summarizes the process of its generation and allows a better design of subsequent data processing systems. Probabilistic reasoning plays a central role in data modeling. Probability theory is a respected framework to work with uncertain or noisy data. Discovering the structure of the data, and finding dependencies in the data are two sides of the same coin.

When the data sample contains all the information in their distribution, directly using the probability density function (PDF) of the data is a powerful tool. When this is not the case, a possibility is to construct scalar descriptors of the PDF that, under certain assumptions, briefly characterize the data structure. This approach is illustrated by statistical moments, which are the most commonly used descriptors of the PDF. There exist consistent non-parametric estimators for the moments. In particular, if the Gaussian assumption is held, the mean and the variance completely describe the PDF.

There are differences between the application of entropy to communication systems and to machine learning. First, machine learning systems handle not only discrete-valued data, but

may also face continuous processes. Second, machine learning algorithms require smooth cost functions, in order to apply local search algorithms. Third, and last, the PDFs of modern applications usually have long tails and real problems usually have many outliers. This makes the Gaussian assumption a poor descriptor in most situations. Therefore, the information theoretic descriptors must be estimated with continuous and differentiable non-parametric estimators. The non-parametric kernel density estimators by Parzen [106] meet these requirements, besides connecting IT with kernel methods. Next, the kernel-based learning theory is introduced with the definition of Reproducing Kernel Hilbert Spaces.

2.2.1 Reproducing Kernel Hilbert Spaces

A Hilbert space is a generalization of a Euclidean space to any finite or infinite number of dimensions. It is an abstract linear vector space that has the structure of an inner product, and it is normed and complete. A Reproducing Kernel Hilbert Space (RKHS) [7] is a Hilbert space associated with a kernel that reproduces every function in the space. The application of RKHS in signal processing was proposed by Parzen [105]. He developed an analysis of random Gaussian processes. They are approached by geometric methods when studied in terms of their second-order moments (covariance kernel). Parzen demonstrated that the RKHS offers an elegant general framework for minimum variance unbiased estimation. The problems are solved algebraically in the RKHS associated with the covariance functions, with the geometric advantages of its inner product.

Let H_k be a Hilbert space of real-valued functions defined on a set E, equipped with an inner product $< .,. >$ and a real-valued bivariate function $K(x,y)$ on $E \times E$. The function $K(x,y)$ is said to be non-negative definite if for any finite point set $\{x_1, x_2, \ldots, x_n\} \subset E$ and for any not all zero corresponding real numbers $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subset R$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \geq 0 \tag{2.8}$$

Kernel density estimation is central in ITL. There is a large overlap between the mathematical conditions required for a kernel for density estimation and positive definite functions. In fact, any non-negative definite bivariate function $K(x,y)$ is a reproducing kernel, as proved by the theorem of Moore-Aronszajn [7]. Kernel-based learning algorithms use the following idea [127]:

$$\begin{aligned} \Phi : & E \rightarrow H_k \\ & x \rightarrow \Phi(x) \end{aligned} \tag{2.9}$$

Via the non-linear mapping (2.9), the data $\{x_1, x_2, \ldots, x_n\} \subset E$ are mapped into a potentially much higher dimensional feature space H_k with a linear structure. A given learning problem in E is solved in H_k instead, by working with $\{\Phi(x_1), \ldots, \Phi(x_n)\} \subset H$. Because H_k is high dimensional, a linear learning algorithm can solve arbitrarily non-linear problems in the input space (if H_k is rich enough to represent the mapping). The inner product formulation implicitly executes the linear algorithm in the kernel feature space, while the data and the operations are all done in the input space. The Mercer theorem [94] guarantees the existence of the non-linear mapping Φ. This property of the kernels is called the "kernel trick". The kernel trick can be used to develop non-linear generalizations of any algorithm that can be expressed in terms of inner products. A kernel that satisfies the Mercer theorem is known as a Mercer kernel. The most widely used Mercer kernel is the Gaussian function.

2.2.2 RKHS and ITL

From a practical perspective, one must estimate entropy from data. In this subsection, the interest lies in computationally simple, non-parametric estimators that are continuous and differentiable. Alfred Renyi [121] derived a set of estimators to apply entropy and divergence as cost functions in learning. They are described next.

There are many factors that affect the determination of the optimum in the process of learning: gradient noise, learning rates, misadjustment, etc. The bias and variance of the entropy estimator are not as critical as in other fields. In consequence, what matters the most in learning is to develop cost functions that can be derived directly from data without further assumptions, and they must capture as much structure as possible of the PDF.

Renyi information measure of order α or Renyi α entropy has the following expression:

$$H_\alpha(P) = \frac{1}{1-\alpha} \log \left(\sum_{k=1}^{N} p_k^\alpha \right) \tag{2.10}$$

with $\alpha \neq 1$ and $\alpha \geq 0$. It is called entropy because Renyi showed that it is a generalization of Shannon's theory, as it is shown next.

Probability mass functions (PMF) can be visualized geometrically as points in a vector space called the simplex. The simplex Δ_N consists of all possible probability distributions for an N-dimensional random variable.

$$\Delta_N = \left\{ p = (p_1, \ldots, p_N)^T \in R^N, p_i \geq 0, \sum_i p_i = 1, \forall i \right\} \tag{2.11}$$

Any point in the simplex is a different PMF and has a different distance to the origin. Let us define the PMF α-norm as

$$||p(x)||_\alpha = \sqrt[\alpha]{\sum_{k=1}^{N} p_k^\alpha} = \sqrt[\alpha]{V_\alpha(X)} \tag{2.12}$$

where $V_\alpha(X) = \sum_k p_k^\alpha = E\left[p_k^{\alpha-1}\right]$ is called the α information potential (IP_α), and can be interpreted as the α power of the PMF α-norm.

In order to see the relation of Renyi entropy on (2.10) with (2.12), the former can be rewritten as:

$$\begin{aligned} H_\alpha(X) &= \frac{1}{1-\alpha}\log\left(\sum_{k=1}^{N} p_k^\alpha\right) = -\log\left(\sum_{k=1}^{N} p_k^\alpha\right)^{\frac{1}{\alpha-1}} \\ &= -\log\left(\sum_{k=1}^{N} p_k p_k^{\alpha-1}\right)^{\frac{1}{\alpha-1}} \end{aligned} \tag{2.13}$$

The argument of the log can be denoted as the α information potential $V_\alpha(X)$ and allows rewriting (2.13) as:

$$H_\alpha(X) = \frac{1}{1-\alpha}\log\left(V_\alpha(X)\right) = -\log\left(\sqrt[\alpha-1]{V_\alpha(X)}\right) \tag{2.14}$$

Therefore, Renyi α entropy takes the $\alpha-1$ root of $V_\alpha(x)$ and rescales it by the negative of the logarithm. In the simplex, α specifies the norm to measure the distance of $p(x)$ to the origin. The free parameter α changes the importance of small values versus large values in the set. There are three special cases of interest. When $\alpha = 0$, H_0 is the logarithm of the number of non-zero components of the distribution, and it is known as Hartley entropy. H_∞ can be thought of as $\lim_{\alpha\to\infty} H_\alpha$ and is called the Chebyshev entropy. The most interesting special case is obtained for $\lim_{\alpha\to 1} H_\alpha$, which is Shannon entropy, which means that Shannon entropy is the limiting case of the 1-norm of the PMF $p(x)$.

Moreover, it can be generalized that, when $\alpha > 1$, Renyi entropy H_α are monotonic decreasing functions of IP_α. Therefore, entropy maximization is equivalent to IP minimization and viceversa. When $\alpha \leq 1$, Renyi entropy H_α are monotonic increasing functions of IP_α. In this case, entropy maximization is equivalent to IP maximization, and viceversa.

Renyi Quadratic Entropy H_2 is of particular interest, as it is a monotonic decreasing function of the $\alpha = 2$ information potential V_2 of the PMF $p(x)$. H_2 implicitly uses a Euclidean

distance from the point $p(x)$ in the simplex to the origin of the space.

$$H_2(X) = -\log\left(\sum_k p_k^2\right) \tag{2.15}$$

As H_2 is a lower bound of Shannon entropy, it may be more efficient than Shannon entropy for entropy maximization.

As stated, ITL needs to estimate entropy and divergence in a non-parametric way. As these descriptors are based on the PDF, kernel density estimation may be a useful technique. Most of the kernels used in density estimation are non-negative bivariate functions and, therefore, they define a RKHS. Let us define the continuous cross entropy between two PDFs $p(x)$ and $q(x)$ as

$$H(p,q) = -\int p(x)\log q(x)dx = -E_p[\log q(x)] \tag{2.16}$$

which, as explained in Sect. 2.1, measures the average number of bits needed to encode data coming from a source with density p, while using model q to encode data. For Renyi entropy, the equivalent quadratic cross entropy is defined as

$$H_2(p,q) = -\log\int p(x)q(x)dx = -\log E_p[q(x)] \tag{2.17}$$

The argument of the logarithm, called the cross information potential (CIP), is a positive definite function, so it defines a RKHS that provides a functional analysis view of the information theoretic descriptors of entropy and divergence. In this thesis, the CIP is utilized as the basis for similarity in the supervised classification method proposed in Chapter 5.

The Renyi's α-divergence is an extension to the KL divergence (2.2) and is defined as:

$$D_\alpha(f||g) = \frac{1}{\alpha-1}\log\int_{-\infty}^{\infty} f(x)\left(\frac{f(x)}{g(x)}\right)^{\alpha-1}dx \tag{2.18}$$

2.3 Applications in Machine Learning

The concepts of IT and ITL can be applied to machine learning, in particular to two core areas such as feature selection and classification, which are the main topics of this dissertation. Feature selection (FS) is the process of detecting relevant features and discarding irrelevant and redundant ones. Its goal is obtaining a subset of features that describes the problem properly

and causes a minimum degradation or even an improvement in performance in the learning algorithms [58]. Classification is another of the classic activities in machine learning, along with regression, clustering and density estimation. Its main goal is assigning observations to a set of categories or classes [97].

2.3.1 Feature Selection

From a functional point of view, FS methods can work in two different ways [156]. Some methods assign weights to each feature, in such a way that the order corresponding to their theoretical relevance is preserved. Methods that follow this approach are known as continuous, individual evaluation or ranking methods. The second set of methods are known as binary or subset evaluation methods. First, they produce candidate feature subsets using search strategies. Then, the subsets are assessed by an evaluation function which determines the final selected subset of features. Moreover, methods can be uni or multivariate, depending on whether they consider each feature independently of the rest or not.

From a structural point of view, FS methods can be classified in three major groups [58]. Filter methods perform the feature selection step as pre-processing, before the learning step. The filter is independent of the learning algorithm and relies on underlying attributes of data. Wrapper methods use the learning algorithm as a subroutine, measuring the usefulness of the features with the prediction performance of the learning algorithm over a validation set. In embedded methods, the FS process is specifically built into the machine learning method, in such a way that the search is guided by the learning process itself.

Each of these approaches has its advantages and disadvantages. The main factors are the speed of computation and the probability of overfitting. Filters are faster than embedded methods, and the latter are faster than wrappers. Regarding overfitting, wrappers are more likely to overfit than embedded methods, which are more likely to overfit than filter methods. In general, filters are relatively inexpensive in terms of computational efficiency.

Filter methods are defined by a criterion J [36]. This criterion measures how relevant a feature or feature subset is. A measure of correlation between the feature and the class label can be a good criterion. There are several types of criteria. In this thesis, those based on IT are considered. For a class label Y, the mutual information score for a feature X_k is:

$$J_{MI}(X_k) = I(X_k;Y) \tag{2.19}$$

In order to use this criterion, a filter must rank the features in order of their J_{MI} and select the top K features. An important limitation is that this approach assumes that each feature is independent of all other features. In general, a set of features should not only be individually relevant, but also should not be redundant with respect to each other [25]. A possible improvement is the Mutual Information Feature Selection (MIFS) criterion [10], which introduces a penalty to correlations between features:

$$J_{MIFS}(X_k) = I(X_k;Y) - \beta \sum_{X_j \in S/X_k} I(X_k;X_j) \tag{2.20}$$

where S is the candidate set of features. Another criterion is the Joint Mutual Information (JMI), which focuses on the complementary information of features [95] [152]:

$$J_{JMI}(X_k) = \sum_{X_j \in S} I(X_k X_j;Y) \tag{2.21}$$

This is the mutual information between a joint random variable $X_k X_j$ and the class label. The idea is to include features that complement with existing features from the subset S of selected features.

There exist more criteria like, for instance: Koller-Sahami metric (KS) [79], Informative Fragments (IF) [139], Fast Correlation Based Filter (FCBF) [156], Conditional Mutual Info Maximization (CMIM) [46], Minimum Redundancy (MINRED) [36], Interaction Gain Feature Selection (IGFS) [38], Conditional MIFS (CMIFS), and Min-Redundancy Max-Relevance (mRMR) [107]. However, only the relevance and redundance of the features regarding the output is taken into account. But there is another important aspect that is forbidden in these approaches: the (economical, computational...) cost of features. This means that there may exist certain subsets of features that, having the same or similar relevance regarding the output, one of them might allow for computational/economical savings. One of the contributions of this thesis is the extension of one of the most used algorithms, mRMR, in order to consider this factor. This contribution is described in Chapter 3.

2.3.2 Classification

Supervised classification in highly non-linear and multimodal problems has been a challenge for machine learning algorithms through the years. Several previous researchers [75] have analyzed the difficulties found when facing these kind of problems by both classical statistical classifiers (such as Fisher Linear Discriminant [44] and its variations) and machine learning

methods (such as artificial neural networks [16] and decision trees like ID3 [117] or C4.5 [118]). Over the last years, more sophisticated models have come out. These models try to mitigate the weaknesses of classical algorithms in order to being able to deal with more complex classification problems. One of the latest and more well-known approaches are Support Vector Machines (SVM) [30]. These models convert a complex non-linear non-separable problem into a linear problem, by means of a transformation to a higher dimensional space.

Most classifiers are global methods. A *global method* attempts to solve a problem by means of adjusting a single model for the whole feature space. However, there exists another approach to the classification problem, the *combination of classifiers* [83]. This is a relatively recent technique that can be considered a meta-algorithm in the sense that it combines a set of component classifiers in order to obtain a more precise and stable model. The two most important strategies to combine classifiers are fusion and selection. On *fusion of classifiers*, each of the classifiers has knowledge of the totality of the feature space. On the other hand, on *selection of classifiers*, each classifier knows only a part of the feature space.

The methods based on *fusion of classifiers* are also known as *ensemble methods*. The most popular strategies are *Boosting*, *Bagging* and *Stacking*:

- *Boosting* is based on the question enunciated by Kearns [73]: "can a set of weak learners create a single strong learner?" They consist of training several weak classifiers iteratively and adding them to a final strong classifier. After a weak learner is added, data are weighted: misclassified samples gain weight and correctly classified ones lose weight. In this manner, newly added weak learners focus more on previously misclassified samples. Algorithms of this family are, e.g., AdaBoost [49] and its variants AdaBoost.M1 and M2 [48], and AdaBoostR [101].
- *Bagging* [24] randomly generates several data sets from the original one with replacement. The models are trained and combined using voting.
- *Stacking* [148] utilizes an extra classifier that learns to combine the outputs of the base classifiers in order to generate a common final output.

The methods based on *selection of classifiers* are also known as *local methods*. The idea of using different classifiers for different inputs was suggested by Dasarathy, B.V. and Sheela, B.V. [31], who combined a linear classifier and a *k-Nearest Neighbor*. Rastrigin [119], in 1981, already proposed a methodology for selection of classifiers that is virtually similar to the one used these days. The philosophy of local methods consists of splitting up the feature space

in several subspaces and adjusting a model for each of these subspaces. Each subproblem is supposed to be simpler than the original model and may be solved with simpler classification models, i.e., linear ones. In this manner, large and complex problems, like the ones dealt with in this chapter, are more approachable. Therefore, a correct division of the original problem is very important for the correct operation of the system. The most straightforward way of splitting up the data is a division in regular regions, which is posible, but it may happen that some of them contain few or no data at all. In order to ensure that the regions always contain some patterns, it is usual to employ a clustering algorithm to split up the data [82, 91].

On unsupervised learning, there exist two schools of thought:

- Methods that build generative models to describe the observed data.
  - Methods that adjust the parameters in order to optimize the likelihood of data with constraints on model architecture, i.e., Bayesian Inference Models [43], and Maximum Likelihood Competitive Learning [120].
  - Methods that require some form of regularization to select a proper model, i.e., Minimum Description Length [123], Bayesian Information Criterion [14], and Akaike Information Criterion [2].
- Methods that use self-organization principles. In this approach, minimization of entropy leads to a featureless solution given by the collapse of all the samples to a single point in space. The idea is to construct energy functions that combine two competing aspects —information preservation and redundancy reduction—. This school of thought has the advantage of not imposing statistical models on data, instead allowing samples the freedom to interact with one another, which in the end reveals the hidden structure of data through self-organization.

The latter approach was used in [125, 116] to develop a simple framework for unsupervised learning based on Information Theory, the Principle of Relevant Information (PRI).

2.3.2.1 Principle of Relevant Information

The classical unsupervised learning algorithms are solutions to the following optimization problem:

$$L[p(x|x_o)] = \min_X (H(X) + \lambda D_{KL}(X||X_o)) \tag{2.22}$$

where $x_o \in X_o$ is the original data set, $x \in X$ is the compressed version of the original data (the clusters), λ is a parameter of variation, $H(X)$ is the entropy between the original and the compressed data, and $D_{KL}(X||X_o)$ is the Kullback-Leibler divergence.

The formulation of the Principle of Relevant Information (PRI) addresses the entropy of a single data set. The solution is specified as an optimization over the compressed data given the original data. The PRI generalizes the classical algorithms (clustering, principal curves, vector quantization), as each of them is represented by a different value of λ. This generalizing principle for unsupervised learning is formulated in terms of information-theoretic quantities.

The family of data x obtained by means of (2.22) is controlled by the variational parameter λ. This parameter controls the level of distortion in compressed data. The estimators of ITL from Sect. 2.2 can be used in this formulation to derive algorithms to obtain the different solutions. Rewriting (2.22) with Renyi's formulation of entropy:

$$L[p(x|x_o)] = \min_X (H_\alpha(X) + \lambda D_\alpha(X||X_o)) \tag{2.23}$$

where Renyi's entropy H_α and Renyi's divergence D_α are respectively defined in (2.10) and (2.18). Cauchy-Schwarz divergence is defined as:

$$D_{CS}(f,g) = \log \int f(x)^2 dx + \log \int g(x)^2 dx - 2\log \int f(x)g(x)dx \tag{2.24}$$

Cauchy-Schwarz divergence can be rewritten in terms of Renyi's quadratic entropy as

$$\begin{aligned} D_{CS}(X,Y) &= -2\log \int f(x)g(x)dx + \log \int f(x)^2 dx + \log \int g(x)^2 dx \\ &= 2H_2(X;Y) - H_2(X) - H_2(Y) \end{aligned} \tag{2.25}$$

Continuing with the formulation of the PRI, redundancy will be measured by Renyi's quadratic entropy $H_2(x)$ and divergence will be measured by the Cauchy-Schwarz divergence $D_{CS}(X,X_o)$:

$$\begin{aligned} J(X) &= \min_X (H_2(X) + \lambda D_{CS}(X,X_o)) \\ &= \min_X [(1-\lambda)H_2(X) + 2\lambda D_{CEF}(X,X_o) - \lambda H_2(X_o)] \end{aligned} \tag{2.26}$$

where $D_{CS}(X,X_o) = 2D_{CEF}(X,X_o) - H_2(X) - H_2(X_o)$, $D_{CEF} = -\log V(X,X_o)$ is the logarithm of the cross-information potential (CIP) and λ is the variational parameter. J is the cost function, with X as its argument. Therefore, $\lambda H_2(X_o)$, the last term in (2.26), is constant with respect to X and can be removed from the optimization problem:

$$J(X) = \min_X [(1-\lambda)H_\alpha(X) - 2\lambda \log V(X,X_o)] \tag{2.27}$$

Hereafter, all these quantities can be estimated directly from samples.

CHAPTER 3

A New Method for Cost Feature Selection Based on Information Theory

The proliferation of high-dimensional data has become a trend in the last few years. Data sets with dimensionality over the tens of thousands are constantly appearing in applications such as medical image, text retrieval or genetic data. In fact, analyzing the dimensionality of the data sets posted in the UCI Machine Learning Repository [8] in the last decades, one can observe that in the 1980s, the maximum dimensionality of data is around 100 features; increasing to more than 1500 features in the 1990s; and finally, in the 2000s, it further increases to about 3 million features [158].

The high dimension of data has an important impact in learning algorithms, since their performance is degraded when a number of irrelevant and redundant features are present. In fact, this phenomenon is known as the curse of dimensionality [69], because unnecessary features increase the size of the search space and make generalization more difficult. For overcoming this obstacle, researchers usually employ dimensionality reduction techniques. In this manner, the set of features required for describing the problem is reduced, most of the times along with an improvement in the performance of the models. Feature selection is arguably the most utilized dimension reduction technique. It consists of detecting the relevant features and discarding the irrelevant ones. Its goal is to obtain a subset of features that describes properly the given problem with a minimum degradation in performance [58], with the implicit benefits of improving data and model understanding and the reduction in the need for data storage. With this technique, the original features are maintained, contrary to what usually happens in other techniques such as feature extraction, where the generated data set is represented by a newly generated set of features, different than the original.

There are some situations where a user is not only interested in maximizing the merit of a subset of features, but also in reducing costs that may be associated to features. For example, for medical diagnosis, symptoms observed with the naked eye are costless, but each diagnostic value extracted by a clinical test is associated with its own cost and risk. In other

fields, such as image analysis, the computational expense of features refers to the time and space complexities of the feature acquisition process [42]. This is a critical issue, specifically in real-time applications, where the computational time required to deal with one or another feature is crucial, and also in the medical domain, where it is important to save economic costs and to also improve the comfort of a patient by preventing risky or unpleasant clinical tests (variables that can be also treated as costs).

Feature selection methods, filters in particular, are mainly based in measures of relevance and redundance of features. There exists a large variety of methods that explore several measures. However, the existence of a feature selection method that takes cost into account is unbeknownst to the author.

Among all the feature selection methods, Minimal Redundancy Maximal Relevance (mRMR) is one of the most relevant. mRMR is a ranked filter based on information theory. In this chapter, the metric function of this algorithm is modified in order to having into account the cost associated to the input features. The goal is to obtain a trade-off between a filter metric and the cost associated to the selected features, in order to select relevant features with a low associated cost while keeping the accuracy. The contents of this chapter have been published in [17].

The remainder of this chapter is organized as follows: Section 3.1 summarizes previous research on the subject. Section 3.2 describes the proposed method in detail. Sections 3.3 and 3.4 describe the experimental study performed and the obtained results, respectively. Finally, Section 3.5 sums up the contents of the chapter.

3.1 Background

Feature selection has been an active and effective tool in numerous fields such as DNA microarray analysis [20, 34], intrusion detection [19, 99], medical diagnosis [3] or text categorization [47]. New feature selection methods are constantly appearing, however, the great majority of them only focuses on removing irrelevant and redundant features but not on the costs for obtaining the input features.

The cost associated to a feature can be related to different concepts. For example, in medical diagnosis, a pattern consists of observable symptoms (such as age, sex, etc.) along with the results of some diagnostic tests. Contrary to observable symptoms, which have no cost, diagnostic tests have associated economical costs and risks. On the other hand, cost can also

be related to computational issues. In the medical imaging field, extracting a feature from a medical image can have a high computational cost.

As one may notice, features with an associated cost can be found in many real-life applications. However, this has not been the focus of much attention for machine learning researchers. As mentioned above, the purpose of this research is to contribute to the problem of cost-based feature selection, trying to balance the correlation of the features with the class and their cost. There have been similar attempts to balance the contribution of different terms in other areas. For instance, in classification, Friedman et al. [50] included a regularization term to the traditional Linear Discriminant Analysis (LDA). The left side term of their cost function evaluates the error and the right side term would be the regularization one, which is weighted with λ. This provides a framework in which, according to the λ value, different regularized solutions can be obtained. Related to feature extraction, in [155] a criterion is proposed to select kernel parameters based on maximizing between-class scattering and minimizing within-class scattering. Applied to face recognition, Wright et al. [149] proposed a general classification framework to study feature extraction and robustness to occlusion via obtaining a sparse representation. Instead of measuring the correlation between a feature and the class, this method evaluates the representation error.

However, the objective of this chapter is completely different, as it is to provide a framework for feature selection where features with an inherent cost could be dealt with. Despite the previous attempts in classification and feature extraction, to the best knowledge of the author, there are only a few attempts to deal with this issue in feature selection. In the early 90s, Feddema et al. [42] were developing methodologies for the automatic selection of image features to be used by a robot. For this selection process, they employed a weighted criterion that took into account the computational expense of features, i.e., the time and space complexities of the feature extraction process. Several years later, Yang et al. [153] proposed a genetic algorithm to perform feature subset selection where the fitness function combined two criteria: the accuracy of the classification function realized by the neural network and the cost of performing the classification (defined by the cost of measuring the value of a particular feature needed for classification, the risk involved, etc.). A similar approach was presented in [66], where a genetic algorithm is used for feature selection and parameters optimization for a support vector machine. In this case, classification accuracy, the number of selected features and the feature cost were the three criteria used to design the fitness function. Another proposal can be found in [131] by presenting a hybrid method for feature subset selection based on ant colony optimization and artificial neural networks. The heuristic that enables ants to select features is the inverse of the cost parameter.

The methods found in the literature that deal with cost associated to the features, which were described above, have the disadvantage of being computationally expensive by having interaction with a classifier, which prevents their use in large databases, a trending topic in the past few years [70]. However, the idea proposed in this paper is applied together with the filter model, which is known to have a low computational cost and be independent of any classifier. By being fast and with a good generalization ability, filters using this cost-based feature selection framework will be suitable for application to databases with a great number of input features like, e.g., microarray DNA data sets.

In light of the above, the novelty of this approach lies in that the research in cost-based selection is extremely scarce in the literature. As a matter of fact, no cost methods can be found in the most popular machine learning and data mining tools. For instance, in Weka [60] we can only find some methods that address the problem of cost associated to the instances (not to the features), and they were incorporated in the latest release. RapidMiner [96] does in fact include some methods that take cost into account, but they are quite simple. One of them selects the attributes that have a cost value which satisfies a given condition and another one just selects the k attributes with the lower cost. Therefore, the cost-based feature selection method proposed in this chapter intends to cover this necessity.

3.2 Description of the method

mRMR (Minimal Redundancy Maximal Relevance) [107] is one of the most employed multivariate ranker filters, due to obtaining good results in several fields [100, 26, 71, 140]. The evaluation function combines two constraints (as the name of the method indicates), maximal relevance and minimal redundancy. The former is denoted by the letter D, it corresponds with the mean value of all mutual information values between each feature x_i and class c, and has the following expression:

$$D(S,c) = \frac{1}{|S|} \sum_{x_i \in S} I(x_i;c) \tag{3.1}$$

where S is a set of features and $I(x_i;c)$ is the mutual information between the feature x_i and the class c. The expression of $I(x;y)$ is:

$$I(x;y) = \int \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{3.2}$$

The constraint of minimal redundancy is denoted by R, and has the following expression:

$$R(S) = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) \tag{3.3}$$

The evaluation function to be maximized combines the two constraints (3.1) and (3.3). It is called Minimal Redundancy Maximal Relevance (mRMR):

$$\Phi(D,R) = \frac{1}{|S|} \sum_{x_i \in S} I(x_i;c) - \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j) = D(S,c) - R(S) \tag{3.4}$$

In practice, this is an incremental search method that selects, on each iteration, the feature that maximizes the evaluation function. Suppose we already have S_{m-1}, the feature set with m - 1 features, the m^{th} selected feature will optimize the following condition:

$$\max_{x_j \in X - S_{m-1}} \left[I(x_j;c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j;x_i) \right] \tag{3.5}$$

The modification of mRMR which is proposed in this chapter consists of adding a term to the condition to be maximized so as to take into account the cost of the feature to be selected:

$$\max_{x_j \in X - S_{m-1}} \left[\left(I(x_j;c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j;x_i) \right) - \lambda C_j \right] \tag{3.6}$$

where C_j is the cost of the feature j, and λ is a parameter introduced to weight the influence of the cost in the evaluation function. λ is a positive real number. If λ is 0, the cost is ignored and the method works as the regular mRMR. If λ is between 0 and 1, the influence of the cost is smaller than the one from the other term. If λ is 1, both terms have the same influence and, finally, if λ is greater than 1, the influence of the cost is greater than the influence of the other term.

3.2.1 Generalization

Ultimately, the general idea consists on adding a term to the evaluation function of the filter to take into account the cost of the features. Since, to the best knowledge of the author, all filters use an evaluation function, this evaluation function could be modified to contemplate costs in

the following manner. Let M_S be the merit of the set of k features S, that is, the value originally returned by the function.

$$M_S = EF(S) \tag{3.7}$$

where EF is the evaluation function. Let C_S be the average cost of S.

$$C_S = \frac{\sum_{i=1}^{k} C_i}{k} \tag{3.8}$$

where C_i is the cost of feature i. The evaluation function can be modified to become:

$$MC_S = M_S - \lambda C_S \tag{3.9}$$

where λ is a parameter introduced in order to weight the influence of the cost in the evaluation.

Notice that when a ranker method that selects features one at a time, such as mRMR, is used, the cardinality of S is 1 and C_S in (3.8) results in the cost of that single feature.

3.3 Experimental study

The experiment is performed over three blocks of data sets (Table 3.1). The data sets in the first and second blocks are available at the UCI Machine Learning Repository [8]. The data sets in the third block are DNA microarray data sets and are available at the web site of the Broad Institute [68]. The main feature of the first block of data sets is that they have intrinsic cost associated to the input features. For the second and third blocks, as these data sets do not have intrinsic cost associated, random cost for their input features has been generated. This decision has been taken because no data sets with cost, other than the four ones of the first block, exist publicly available, to the best knowledge of the author. For each feature, the cost was generated as a random number between 0 and 1. As an example, on Table 3.2, the costs for each feature of Yeast data set are displayed.

Overall, the chosen classification data sets are very heterogeneous. They present a variable number of classes, ranging from two to twenty six. The number of samples and features range from single digits to the tens of thousands. Notice that data sets in the first and second blocks have a larger number of samples than features, whilst data sets in the third block have a much larger number of features than samples, which poses a big challenge for feature selection researchers. This variety of data sets allows for a better understanding of the behavior of the proposed method.

Data set	No. features	No. samples	No. classes
Hepatitis	19	155	2
Liver	6	345	2
Pima	8	768	2
Thyroid	20	3772	3
Letter	16	20000	26
Magic04	10	19020	2
Optdigits	64	5620	10
Pendigits	16	7494	10
Sat	36	4435	6
Segmentation	19	2310	7
Waveform	21	5000	3
Yeast	8	1033	10
Brain	12625	21	2
CNS	7129	60	2
Colon	2000	62	2
DLBCL	4026	47	2
Leukemia	7129	72	2

Table 3.1: Description of the data sets

Feature	Cost
1	0.5093
2	0.1090
3	0.5890
4	0.2183
5	0.8112
6	0.6391
7	0.2741
8	0.1762

Table 3.2: Random costs of the features of Yeast data set

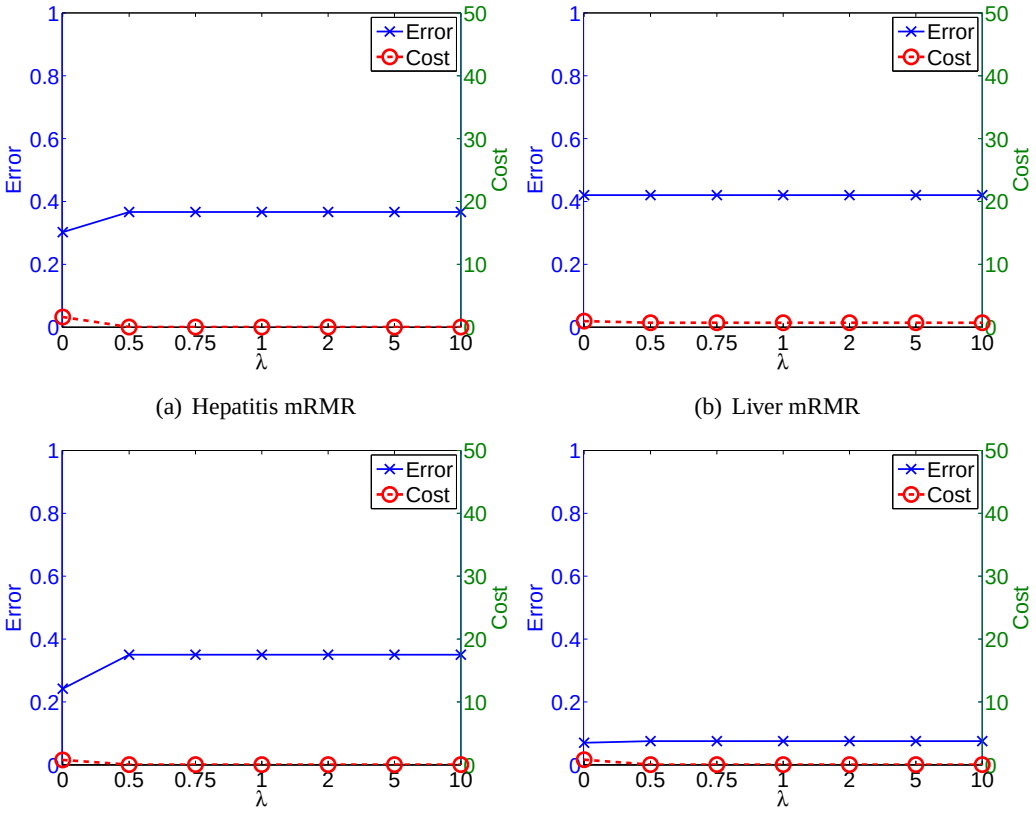
The experiment consists of performing feature selection with Cost mRMR over the data sets. The goal of the experiment is to study the behavior of the method under the influence of the λ parameter. The performance is evaluated in terms of both the total cost of the selected features and the classification error committed by a SVM classifier trained only with the selected features (estimated with a 10-fold cross-validation). It is expected that, the larger the λ, the lower the cost and the higher the error, because increasing λ gives more weight to cost at the expense of correlation between features. Moreover, a Kruskal-Wallis statistical test and a multiple comparison test (based on Tukey's honestly significant difference criterion [136]) [65] have been run on the obtained results. The results of the tests can help the user to choose the value of the λ parameter. As mRMR is a ranker, it does not return a subset of selected features. It returns all the features sorted by the evaluation function for each feature. In consequence, a threshold must be chosen in order to train the SVM classifier. This threshold is obtained by executing a subset feature selection method —CFS [61] in particular— over the data sets. The number of features CFS selects for each data set is utilized as a threshold for mRMR.

3.4 Experimental results

Figures 3.1, 3.3 and 3.4 show the cost and error for several values of λ. The solid line with 'x' represents the error (referenced on the left Y axis) and the dashed line with 'o' represents the cost (referenced on the right Y axis). Notice that when $\lambda = 0$ the cost has no influence on the behavior of the method and it behaves as if it were the non-cost version.

Figure 3.1 plots the classification error/cost of the four data sets with cost associated found at the UCI repository (see Table 3.1). The behavior expected when applying cost feature selection is that the higher the λ, the lower the cost and the higher the error. The results obtained for the first block of data sets, in fact, show that cost value behaves as expected (although the magnitude of the cost does not change too much because these data sets have few features and the set of selected ones is often very similar). The error, however, remains constant in most of the cases. This may happen because these data sets are quite simple and the same set of features is often chosen. The Kruskal-Wallis statistical test run on the results displays that the errors are not significantly different, except for Pima data set. This fact can be caused because this data set has very few expensive features (which are often associated with a higher predictive power), as can be seen on Table 3.3. Therefore, removing them has a greater effect on the classification accuracy.

Fig. 3.2 displays the results of the Kruskal-Wallis statistical test for Pima data set. The



(a) Hepatitis mRMR

(b) Liver mRMR

(c) Pima mRMR

(d) Thyroid mRMR

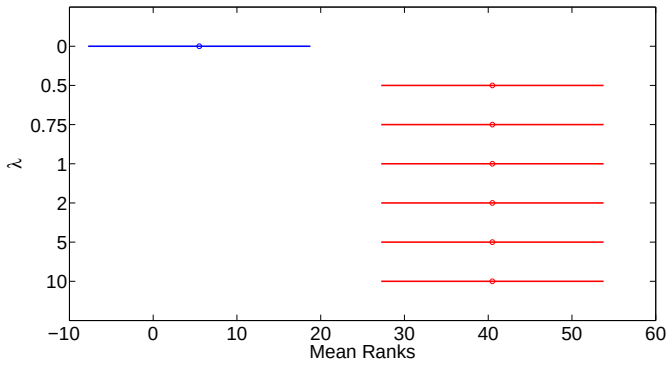
Figure 3.1: Error / cost plots of first block of data sets for cost feature selection mRMR

Feature	Cost
1	0.0100
2	0.7574
3	0.0100
4	0.0100
5	0.9900
6	0.0100
7	0.0100
8	0.0100

Table 3.3: Costs of the features of Pima data set (normalized to 1)

Kruskal-Wallis ANOVA Table					
Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	10.500,0	6	1.750,00	25,91	2,00E-04
Error	17.467,5	63	277,26		
Total	27.967,5	69			

(a) ANOVA Table (Cost mRMR).



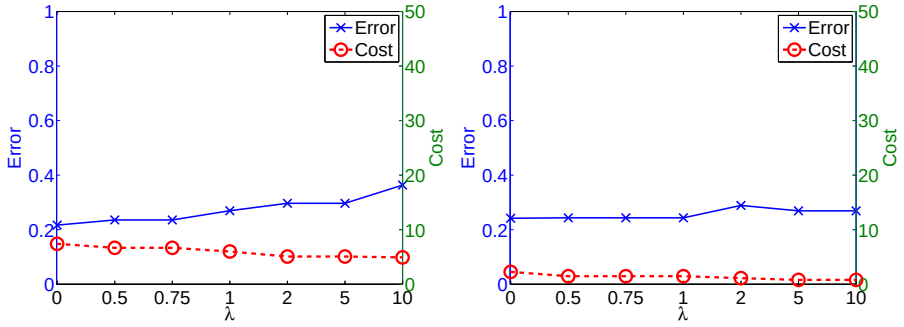
(b) Graph of multiple comparison (Cost mRMR).

Figure 3.2: Kruskal-Wallis statistical test results of Pima data set

entries in the ANOVA (ANalysis Of VAriance) Table (Fig. 3.2(a)) are the usual sums of squares (SS), degrees of freedom (df), mean square estimator (MS), chi-square statistic (Chi-sq) and the *p*-value that determines the significance of the chi-square statistic (Prob>Chi-sq).

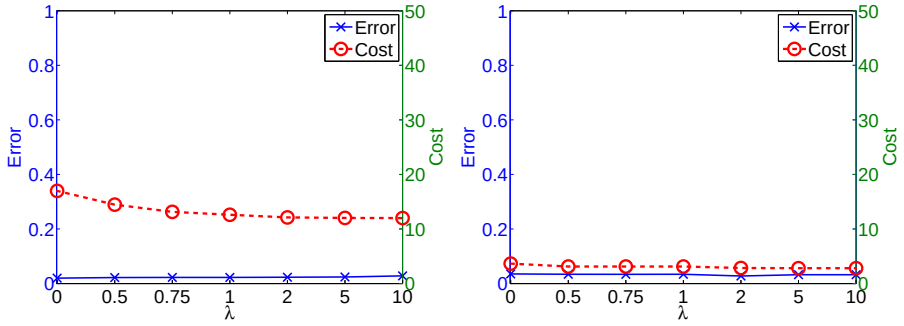
As can be seen, the *p*-value is 2×10^{-4} for Cost mRMR, as displayed in Fig. 3.2(a). This indicates that there exist values significantly different than others. In Fig. 3.2(b), it is shown which groups of errors are significantly different, information that can be helpful for the user to decide which value of λ utilize. When using Cost mRMR, a reduction in cost can not be achieved without worsening the error measure. For Cost mRMR, when λ is 0 (and hence, the cost is not taken into account), the second feature is selected, which has a high cost (see Table 3.3). However, when the method is forced to decrease the cost (by increasing the value of λ), this feature is not selected anymore and prevents the classifier to obtain a high prediction accuracy.

The error/cost graphs of the second block of data sets are displayed in Fig. 3.3. It can be seen how cost decreases, according to expected, and how, contrary to first block, error usually raises when λ increases. In the cases when error raises (see Fig. 3.3(a), for example), there exist significant error changes (p-values are close to zero), therefore the user has to make a choice to find an appropriate trade-off between cost and error.



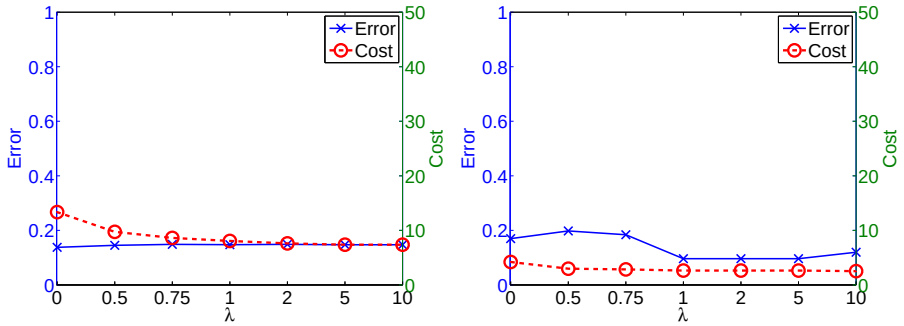
(a) Letter mRMR

(b) Magic04 mRMR



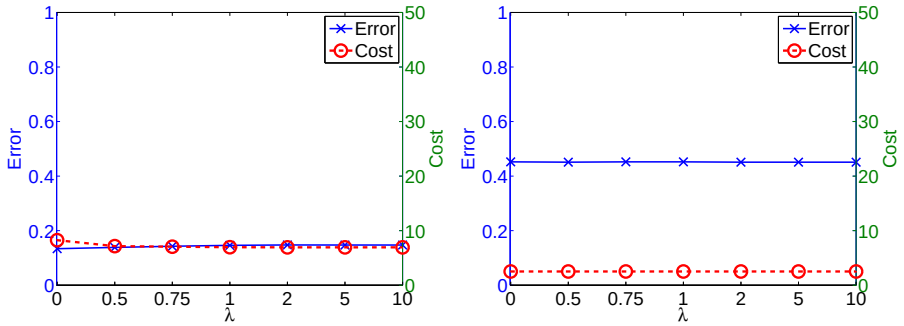
(c) Optdigits mRMR

(d) Pendigits mRMR



(e) Sat mRMR

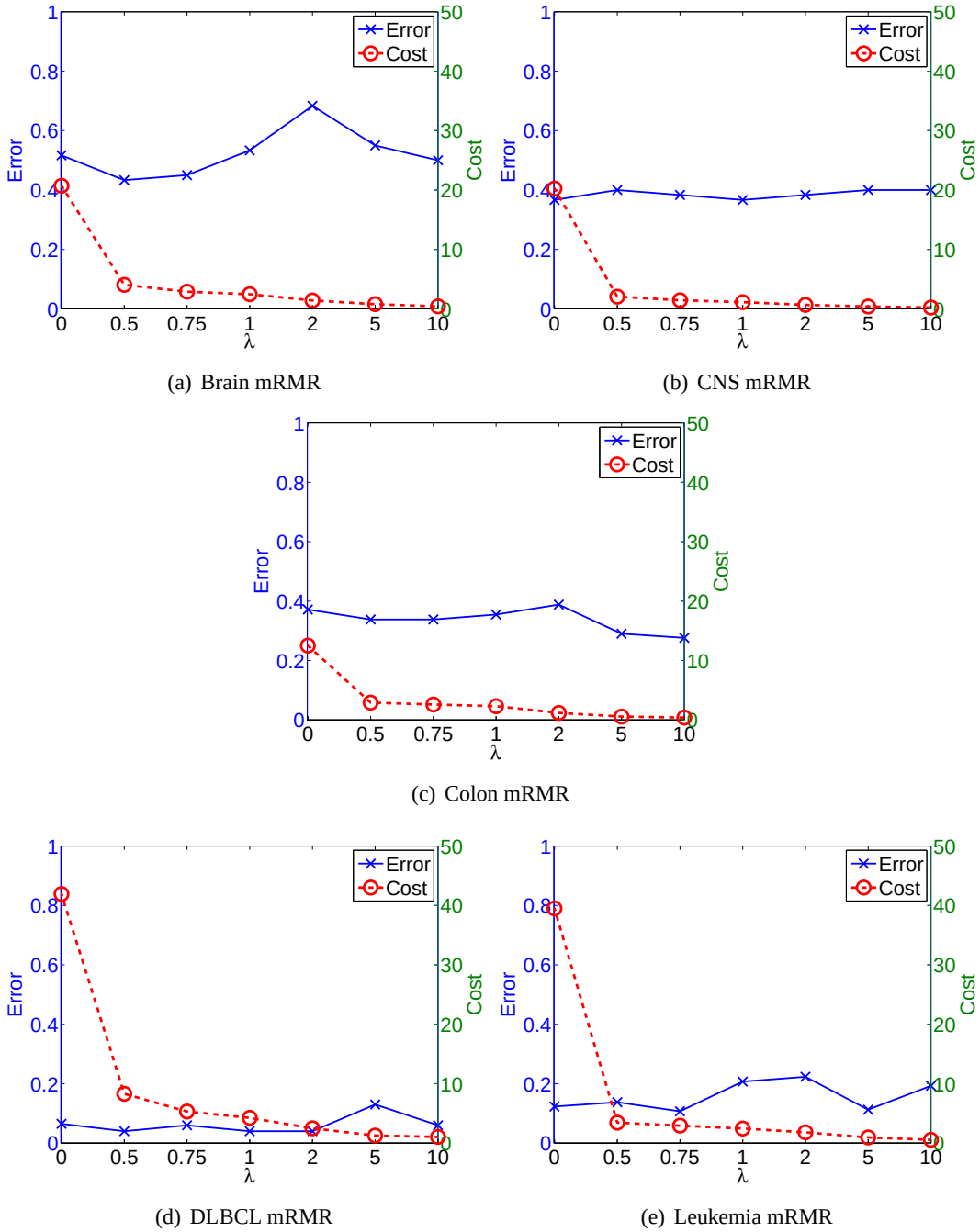
(f) Segment mRMR



(g) Waveform mRMR

(h) Yeast mRMR

Figure 3.3: Error / cost plots of second block of data sets for cost feature selection with mRMR



(a) Brain mRMR

(b) CNS mRMR

(c) Colon mRMR

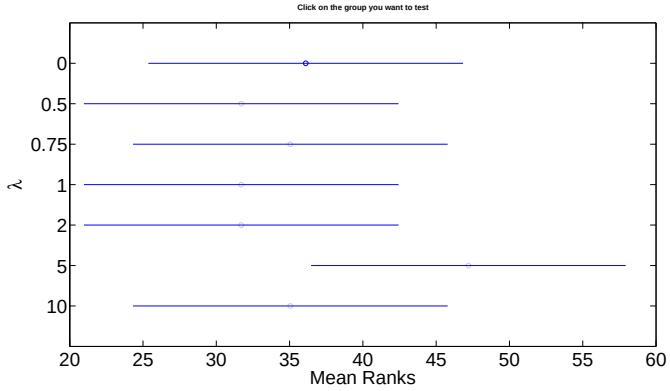
(d) DLBCL mRMR

(e) Leukemia mRMR

Figure 3.4: Error / cost plots on third block of data sets for cost feature selection with mRMR

Kruskal-Wallis ANOVA Table					
Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	1.809,8	6	301,63	6,83	0,3371
Error	16.481,3	63	261,61		
Total	18.291,0	69			

(a) ANOVA Table.



(b) Graph of multiple comparison.

Figure 3.5: Kruskal-Wallis error statistical test of DLBCL data set with Cost mRMR

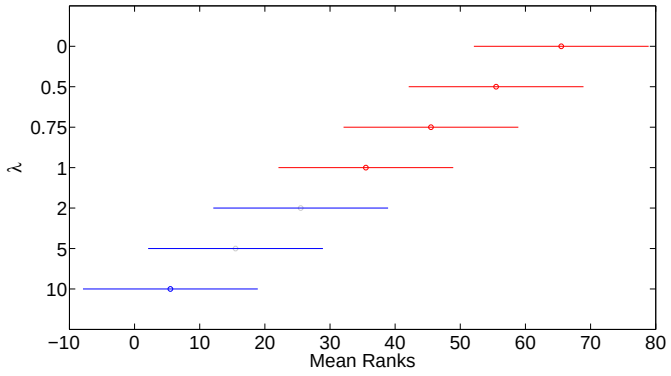
Finally, Fig. 3.4 presents the results for the third block of data sets, corresponding with the well-known DNA microarray domain, with much more features than samples. As expected, cost decreases as λ increases, and since these data sets have larger number of input attributes than the ones in previous blocks, cost experiments larger variability (see, for instance, Figs. 3.4(d), 3.4(e)). For instance, for the DLBCL data set, it can be chosen $\lambda = 10$, as the errors are not significantly different (see Fig. 3.5) and the cost for $\lambda = 10$ is significantly lower than the one for the four first λ (0, 0.5, 0.75 and 1).

Notwithstanding, the behavior of the error, in some cases, and contrary to expected, remains almost constant (see, for instance, Fig. 3.4(b)). The reason why the error is not raising can be two-fold:

- On the one hand, it is necessary to remind that in this research the proposed method is a filter feature selection method. This approach has the benefit of being fast and computationally inexpensive. This characteristic of filters can cause that the selected features, according to particular criteria, would not be the more suitable for a given classifier to obtain the highest accuracy. Therefore, forcing a filter to select features according to another criterion rather than correlation (or the one used for each particular filter) may

Kruskal-Wallis ANOVA Table					
Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	28.000,0	6	4.666,67	67,61	1,27E-12
Error	577,5	63	9,17		
Total	28.577,5	69			

(a) ANOVA Table.



(b) Graph of multiple comparison.

Figure 3.6: Kruskal-Wallis cost statistical test of DLBCL data set with Cost mRMR

cause the selection of features to be more suitable for minimizing classification error. For example, in [21, 78], a synthetic data set called Monk3 is dealt with. Among others, this data set contains three relevant features. However, some classifiers obtain a better classification accuracy when filters only had selected two relevant features than when selecting the three relevant ones. This fact demonstrates that the behavior of some filters is somewhat unpredictable and not always the one expected.

- On the other hand, it has to be noted that DNA microarray data sets are a difficult challenge for feature selection methods, due to the enormous amount of features they present. In fact, the filters evaluated in this research are usually retaining a maximum of 2% of features. Therefore, irregular results are expected with such an important reduction in number of features.

3.5 Summary

In this chapter, a new cost-based feature selection method is proposed. The objective is solving problems where not only it is interesting to minimize the classification error, but also reducing

costs that may be associated to input features. The approach consists of adding a new term to the evaluation function of mRMR so that it is possible to reach a trade-off between the error and the cost associated to the selected features. A new parameter, called λ, is introduced in order to adjust the influence of the cost into the evaluation function, allowing the user fine control of the process according to his needs.

In order to test the adequacy of the proposed idea, experimentation is performed over a broad suite of different data sets. Results after performing classification with a SVM display that the approach is sound and allows the user to reduce the cost without compromising the classification error significantly, which can be very useful in fields such as medical diagnosis or real-time applications.

CHAPTER 4

A New Ensemble Approach for Feature Selection Based on Ranking Learning

In the previous chapter, the problem of the absence of cost criteria in filter FS methods was confronted. In this one, two problems are addressed: (1) the non-existence of a "best" method, which causes that the user has to search and choose a specific method for each specific problems; (2) the heterogeneity of data sets, which makes it difficult to obtain good results with one single method.

In the past, machine learning methods used to employ a single learning model. However, it has been observed that the technique of using multiple prediction models for solving the same problem, known as ensemble learning, is effective [83, 84]. The idea builds on the assumption that combining the output of multiple experts is better than the output of a single expert. Typically, ensemble learning has been applied to classification. However, ensemble learning can also be thought as a means of improving other machine learning disciplines such as feature selection.

In this chapter, the feature rankings obtained by each member of the ensemble are combined prior to the classification stage, by using ranking function learning [54], a technique that allows to learn the ranking of features from the individual rankings provided by the components of the ensemble. The use of an ensemble instead of a single method induces diversity. The objective is to reduce the variance associated to using regular feature selection methods, since the proposed ensemble takes advantage of the strengths of the single selectors and overcomes their weak points. Two approaches are presented, depending on how data is distributed and the variety of feature selectors to be used. Experimental validation of the methodology on a range of UCI data sets [9] shows the adequacy of the proposed ensembles, paving the way to their application to other real-world data sets, and releasing the user from the decision of which feature selection algorithm is the most appropriate for a given problem.

Besides, machine learning methods have come to be a necessity for many companies, in order to obtain useful information and knowledge from their increasingly massive databases. Real life data sets come in diverse flavors and sizes, and so their nature imposes several substantial restrictions for both learning models and feature selection algorithms [137]. Data sets may be very large in samples and number of features and, also, there might be problems with redundant, noisy, multivariate and non-linear scenarios. Thus, most methods alone are not capable of confronting these problems, and something like "the best feature selection method" simply does not exist, making it difficult for users to select one method over another. In order to make a correct choice, a user not only needs to know the domain well and the characteristics of each data set, but is also expected to understand technical details of available algorithms [90]. As experts of this type are not universally available, more user-friendly methods are necessary. In this sense, a possible way to confront this situation is to use an ensemble of feature selection algorithms, which is the idea proposed in this chapter. Using an ensemble avoids the need to choose a specific method for solving a problem. Specifically, methods that follow the ranking approach are used, i.e., they return an ordered ranking of all the features. Notice that methods that return a ranking of features are less computationally expensive than those which return a subset of selected features, and this is of vital importance when the current tendency is toward Big Data problems. Then, the outputs of all the components of the ensemble have to be combined in order to produce a common final output. The ensemble proposed in this chapter combines these rankings using Ranking SVM [72], which is a SVM-based method for learning of ranking functions.

In the case of ensemble feature selection, each individual component is known as a base selector. If the base selectors are all of the same kind, the ensemble is known as homogeneous. Otherwise, it is known as heterogeneous. There are several ways in which an ensemble can be formed. In this chapter, two of them are explored: (a) N selections using a variety of different feature selection algorithms, all using the same training data and (b) N selections using the same feature selection algorithm, using different training data. Feature selection can also take advantage of data distribution. Most feature selection methods do not scale well when the number of features grows. Processing multiple subsets concurrently means that the training of feature selection methods is faster. This advantage is achieved with option (b). Part of the contents of this chapter have been published in [128].

The remainder of this chapter is organized as follows: Section 4.1 summarizes previous research on the subject. Section 4.2 introduces the proposed ensemble and its algorithm, as well as the individual ranker methods and the Ranking SVM method used to join the individual rankers. Next, Section 4.3 describes the data sets, the experimental design, and the experimental results. Finally, in Section 4.4, the contents of the chapter are summarized.

4.1 Background

Feature selection has been applied in many machine learning and data mining problems. The aim of feature selection is to select a subset of features that minimizes the prediction error obtained by a given classifier. Previous works, as those presented by Guyon and Elisseeff [57] or Hall and Holmes [62] collect different approaches used for feature selection, including feature construction, feature ranking, multivariate feature selection, efficient search methods and feature validity assessment methods.

Along the last few years, it has been observed that, by using and combining different learning models on the same problem, better results could be obtained. This combination of machine learning methods for solving problems is called *ensemble learning*. Moreover, combining classifiers appears as a natural step forward when a critical mass of knowledge of single classifier models has been accumulated, and have been rapidly growing and enjoying a lot of attention from pattern recognition and machine learning communities [83].

As mentioned before, ensemble learning has been typically applied to classification, where the most popular methods are *bagging* [24] and *boosting* [126]. Bagging creates an ensemble by training individual classifiers on bootstrap samples of the training set. Each bootstrap sample is generated by randomly selecting, with replacement, n instances from the training set where n is the size of the training set. As a result of the sampling with replacement procedure, each classifier is trained on the average of 63.2% of the training instances. The prediction of each classifier is combined using simple voting. On the other hand, in the boosting approach the sampling is proportional to an instance's weight. Bagging and boosting are two of the most well-known ensemble learning methods due to their theoretical performance guarantees and strong experimental results. Although these models are the most used to improve the classification results, new ensemble learning techniques on the feature subspace have been proposed. The *Random Subspace* [64] method is a simple random selection of feature subsets derived from the theory of stochastic discrimination. Optiz [104] describes an ensemble feature selection technique for neural networks called *Genetic Ensemble Feature Selection*. Another ensemble method for decision trees is called *Stochastic Attribute Selection Committees* [159], while *Multiple Feature Subsets* [12] is a combining algorithm for nearest neighbor classifiers. Finally, for steganalysis of digital media, an ensemble of classifiers implemented as random forests [77] has been proposed, since this ensemble is ideally suited for this kind of problems.

In recent works it is proposed to improve the robustness of a feature selection algorithm by using multiple feature selection evaluation criteria. Several studies have been performed in

this general area, in order to achieve better classification accuracy. One of these studies [135] has been conducted on 21 UCI data sets [9], comparing five measures of diversity with regard to their possible use in ensemble feature selection. This study considers four search strategies for ensemble feature selection together with the simple random subspacing: genetic search, hill-climbing, and ensemble forward and backward sequential selection. Based on the idea of multiple feature selection evaluation criteria, many ensembles of feature selection methods have appeared. A *Multicriterion Fusion-based Recursive Feature Elimination* [151] *(MCF-RFE)* algorithm is developed with the goal of improving both the classification performance and the stability of the feature selection results. A feature ranking scheme for *Multi-layer Perceptron* [145] *MLP* ensembles is proposed, along with a stopping criterion based upon the *out-ofbootstrap (OOB)* estimate. Experimental results on benchmark data demonstrate the versatility of the MLP base classifier in removing irrelevant features.

Finally, there are some other works in which all the feature selection methods of the final ensemble are ranker methods. Diversity can be achieved by using various rankers, combined afterwards to yield more stable and robust results. Three commonly used filter-based feature ranking techniques for text classification problems were used by Olsson and Oard [103], where the combining methods employed are lowest, highest and average rank.

Wang et al. perform a few outstanding papers in this area, providing two interesting studies. The first one examines the ensembles of six commonly used filter-based rankers [141] and the second one studies seventeen different ensembles of feature ranking techniques [142], with six commonly-used rankers, the signal-to-noise filter technique *(S2N)* [150], and eleven threshold-based rankers. In their second paper, the ensembles are composed of different numbers of rankers, ranging from two to eighteen single feature selection methods. Also, other studies collect different methods to combine the single generated rankings, with the aim of obtaining a final ensemble. This combination of single rankings covers from simple —as mean, median, minimal, etc.— to more complex methods —as *Weighted mean aggregation* [1] *(WMA)*, *Complete linear aggregation* [1] *(CLA)* and *Robust ensemble feature selection* [13] *Rob-EFS*—.

4.2 Proposed method

The method proposed in this chapter is an ensemble of feature selection methods that obtain a ranking of the features (individual evaluation methods). The outputs of the components of the ensemble have to be combined in order to produce a common final output. This is performed using Ranking SVM [72], which is a SVM-based method of learning of ranking functions.

The problem of ranking is formalized as follows: for a query q and a data collection $D = \{d_1, \ldots, d_n\}$, the system should return a ranking r^* that orders the data in D according to their relevance to the query. An optimal ordering r^* can not be achieved. Instead, an operational function f is evaluated by how closely its ordering $r_{f(q)}$ approximates the optimum. If a datum d_i is ranked higher than d_j for an ordering r, i.e. $d_i <_r d_j$, then $(d_i, d_j) \in r$, otherwise $(d_i, d_j) \notin r$. The similarity between the ranking $r_{f(q)}$ and the target ranking r^* is measured by using Kendall's τ [74]. For two finite strict orderings $r_a \subset D \times D$ and $r_b \subset D \times D$, Kendall's τ is defined based on the number P of concordant pairs and the number Q of discordant pairs. A pair $d_i \neq d_j$ is concordant if both r_a and r_b agree in how they order d_i and d_j. It is discordant if they disagree. Therefore, τ can be defined as:

$$\tau(r_a, r_b) = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{\binom{m}{2}} \tag{4.1}$$

where m is the cardinality of D, and $\binom{m}{2}$ is the sum of P and Q for strict orderings.

The algorithm selects a ranking function f that maximizes:

$$\tau_S(f) = \frac{1}{n} \sum_{i=1}^{n} \tau(r_{f(q_i)}, r_i^*) \tag{4.2}$$

The function f must maximize (4.2) and must generalize well beyond the training data. Consider the class of linear ranking functions (4.3), where $\mathbf{w}$ is a weight vector that is adjusted by learning, and $\Phi(q, d)$ is a mapping onto features that describes the match between query q and datum d.

$$(d_i, d_j) \in f_{\mathbf{w}}(q) \Longleftrightarrow \mathbf{w}\Phi(q, d_i) > \mathbf{w}\Phi(q, d_j) \tag{4.3}$$

The task of the learner is to minimize the number of discordant ranking pairs. For the class of linear ranking functions (4.3), this is equivalent to finding the weight vector $\mathbf{w}$ so that the maximum number of the following inequalities (4.4) is satisfied.

$$\begin{aligned} &\forall (d_i, d_j) \in r_1^* : \mathbf{w}\Phi(q_1, d_i) > \mathbf{w}\Phi(q_1, d_j) \\ &\ldots \\ &\forall (d_i, d_j) \in r_n^* : \mathbf{w}\Phi(q_n, d_i) > \mathbf{w}\Phi(q_n, d_j) \end{aligned} \tag{4.4}$$

Unfortunately, this problem is known to be NP-hard, however it is possible to approximate the solution by introducing non-negative slack variables $\xi_{i,j,k}$ and minimizing the upper bound $\sum \xi_{i,j,k}$. Therefore, the above problem is optimized, obtaining the approximation shown in

(4.5).

$$\begin{aligned}
&\text{minimize:} && V(\mathbf{w},\xi) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i,j,k}\xi_{i,j,k} \\
&\text{subject to:} && \\
&\forall (d_i,d_j) \in r_1^* : && \mathbf{w}\Phi(q_1,d_i) \geq \mathbf{w}\Phi(q_1,d_j) + 1 - \xi_{i,j,1} \\
& && \dots \\
&\forall (d_i,d_j) \in r_n^* : && \mathbf{w}\Phi(q_n,d_i) \geq \mathbf{w}\Phi(q_n,d_j) + 1 - \xi_{i,j,n} \\
&\forall i \forall j \forall k : && \xi_{i,j,k} \geq 0
\end{aligned} \tag{4.5}$$

C is a parameter that controls the trade-off between the margin size and the training error. By rearranging the constraints in (4.5) as

$$\mathbf{w}(\Phi(q_k,d_i) - \Phi(q_k,d_j)) \geq 1 - \xi_{i,j,k} \tag{4.6}$$

it becomes equivalent to that of SVM classification on pairwise difference vectors $\Phi(q_k,d_i) - \Phi(q_k,d_j)$. For each query-model pair, features are calculated to measure the similarity between them. The ranking order of the model objects is also known. Thus, the input to the SVM learning algorithm, to learn the optimal ranking function, are the training data presented above. Given a new query q, the model objects can be sorted based on their value of

$$rsv(q,d_i) = \mathbf{w}\Phi(q,d_i) = \sum_{k,l}\alpha_{k,l}^*\Phi(q_k,d_l)\Phi(q,d_j). \tag{4.7}$$

The $\alpha_{k,l}^*$ can be derived from the values of the dual variables at the solution.

There are several ways to design an ensemble [23]. In this thesis, two of them are used:

1. N models generated using the same method, all with different training data (See Fig. 4.1). An important problem of ensemble methods is the computation time they take in comparison to individual methods. One way to deal with this is to distribute the data set in order to parallelize the task of training. Therefore, this variation of the method consists in distributing the training data among a number of nodes. The training samples are randomly distributed in disjoint sets without replacement. The same method is then executed on each of the nodes and the ranking obtained is thereafter combined using the Ranking SVM union method.

2. N models generated using different methods, all with the same training data (See Fig. 4.2). The second variation of the method trains several different methods over the same training data. The output obtained from the methods is then combined using the Ranking SVM union method.

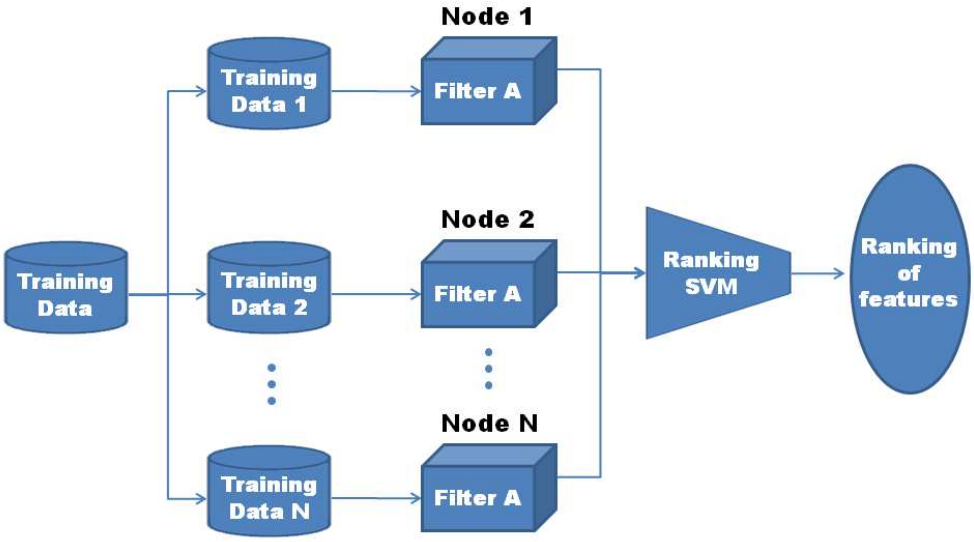


Figure 4.1: First design: same filter, different training data.

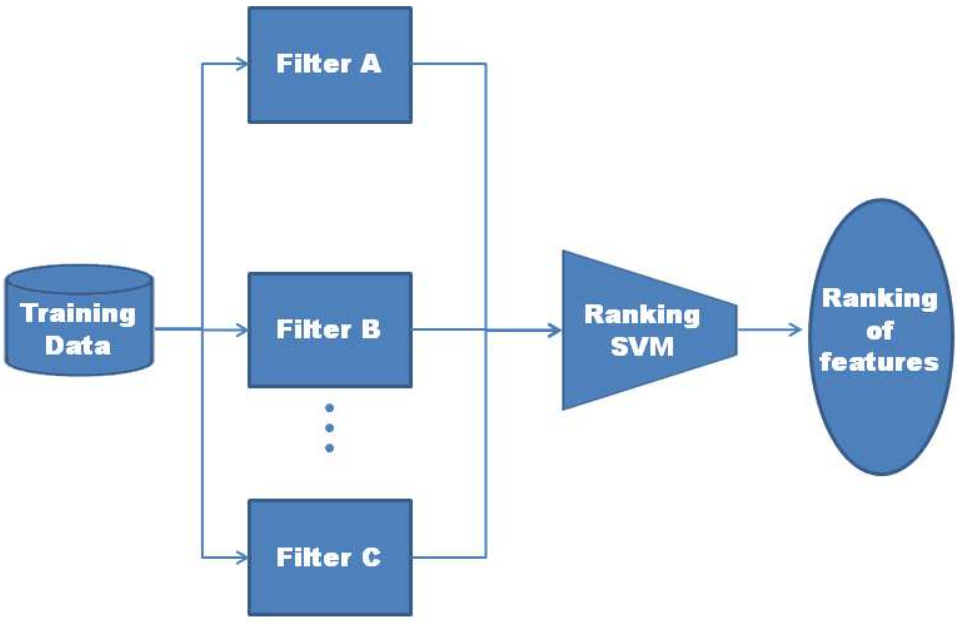


Figure 4.2: Second design: different filters, same training data.

Among the broad suite of feature selection methods available in the literature, three filters and two embedded methods were chosen as candidate components of the ensemble:

- *Information Gain* [117] (filter): this is one of the most common univariate methods of evaluation based on IT. It evaluates the features according to their information gain, only taking into account one feature at each time. The measure utilized to rank variables is the entropy. If the observed values of a variable Y in the training data set are partitioned according to the values of a second feature X, and the entropy of Y with respect to the partitions induced by X is less than the entropy of Y prior to partitioning, then there is a relationship between features Y and X. Then, the entropy of Y after observing X is:

$$H(Y|X) = \sum p(x) \sum p(y|x) log_2(p(y|x)) \tag{4.8}$$

  where $p(y|x)$ is the conditional probability of y given x. Given the entropy as a criterion of "impurity" in a training set S, a measure reflecting additional information about Y provided by X can be defined. This measure represents the amount by which the entropy of Y decreases. It is known as IG and it is an indicator of the dependency between X and Y:

$$IG = H(Y) - H(Y|X) = H(X) - H(X|Y) \tag{4.9}$$

  IG is a symmetrical measure. The method provides an orderly classification of all the features, and then a threshold is required to select a certain number of them according to the order obtained. A weakness of the IG criterion is that it is biased in favor of features with more values even when they are not more informative.

- *ReliefF* [80] (filter): this method is an extension of the original Relief algorithm [76] that can handle multiclass problems. It is more robust and capable of dealing with incomplete and noisy data. As the original Relief, ReliefF works by randomly selecting an instance R_i from the data and then locating the k nearest neighbors from the same class (named "nearest hits", H_j) and the k nearest neighbors from each of the other different classes (named "nearest misses", $M_j(C)$). It updates the quality estimation $W[A]$ for all attributes A depending on their values for R_i, hits H_j and misses $M_j(C)$. If the instances R_i and H_j have different values for the attribute A, then this attribute separates instances of the same class, which is not desirable, and thus the quality estimation $W[A]$ has to be decreased. On the other hand, if instances R_i and M_j have different values for the attribute A for a class then the attribute A separates two instances with different class values, which is desirable, and therefore the quality estimation $W[A]$ is increased. Since ReliefF considers multiclass problems, the contribution of all the hits and all the misses is averaged. Besides, the contribution for each class of the misses is weighted with the

prior probability of that class $P(C)$ (estimated from the training set). The whole process is repeated m times where m is a user-defined parameter.

This method may be applied in all situations, has low bias, includes interaction among features and may capture local dependencies which other methods may miss.

- *mRMR* [107] (filter): Minimum Redundancy Maximum Relevance uses mutual information to select the features that have the highest relevance with the class and are minimally redundant between them. As stated before, it constitutes one of the most used multivariate filter methods based on IT. A most thorough description of the method can be seen in Sect. 3.2.

- *SVM-RFE* [59] (embedded): Recursive Feature Elimination for Support Vector Machines (SVM-RFE) performs feature selection by iteratively training a SVM classifier with the current set of features. It removes the least important feature, as indicated by the weights in the SVM solution.

- *FS-P* [93] (embedded): Feature Selection - Perceptron (FS-P) trains a Perceptron in a supervised manner and uses its interconnection weights to rank the features. A Perceptron is a simple type of linear feed-forward artificial neural network.

This set of ranker methods was selected because (i) they are based on different metrics so they ensure diversity in the final ensemble; and (ii) they are widely used by researchers in feature selection.

4.3 Experimental study

The performance of the proposed ensemble is tested over five well-known data sets, which are listed in Table 4.1. The number of samples ranges from 1 484 to 67 557 and the number of features oscillates from 8 to 617. These data sets conform an interesting suite to check the adequacy of the ensemble.

The experimental study is split in two parts, according to each of the designs proposed in Sect. 4.2 (see Figs. 4.1 and 4.2).

Data set	Samples	Features	Classes
Connect4	67,557	42	3
Madelon	2,400	500	2
Spambase	4,601	57	2
Yeast	1,484	8	10
Isolet	7,797	617	26

* All data sets can be downloaded at [8] .

Table 4.1: Data sets employed in the experimental study

4.3.1 Experimental study for the distributed approach

The experiment performed consists of a comparison between the use of single feature selection methods and the use of an ensemble over a 10-fold cross validation. In the case of the ensemble, the training samples are randomly split in four packages and the feature selection method execution is parallelized. The pseudo-code of this approach can be seen in Algorithm 1.

Algorithm 1 Pseudo-code of the proposed method

Inputs: number of nodes N, threshold of the number of features to be selected T.
Output: classification prediction P.

1. Split training data between N training nodes. The training samples are randomly distributed in disjoint sets without replacement.
2. For each n from 1 to N, obtain ranking A_n in node n.
3. Combine rankings A_n, $n = 1..N$ with Ranking SVM, obtaining A.
4. Select T first attributes from A, obtaining A_t.
5. Build a SVM classifier with the selected A_t attributes.
6. Obtain prediction P.

The results of the experiment (both average training time and average test error) are displayed in Tables 4.2, 4.3, 4.4, and 4.5. Test error is measured using a Support Vector Machine (SVM) classifier, with a RBF kernel, gamma 0.01, and C 1. The first table (Table 4.2) displays the average training times in seconds for the five feature selection methods in the five data sets.

It can be seen how the distributed strategy improves the training times considerably. The next three tables (Tables 4.3, 4.4, and 4.5) display the average test errors. Remember that the feature selection methods used in this chapter are rankers, i.e., they do not select a subset of features: they sort all the features. Therefore, it is necessary to establish a threshold in order to obtain a practical subset of features. Three thresholds are utilized: 10%, 25%, and 50%, and test errors corresponding with each of them are shown respectively in the mentioned three tables.

Data set		IG	ReliefF	mRMR	SVM-RFE	FS-P
Connect4	Single	0.05 ± 0.01	37.57 ± 3.86	1.38 ± 0.07	691.62 ± 90.16	13.60 ± 2.43
	Ens.	0.02 ± 0.01	0.48 ± 0.03	0.23 ± 0.01	7.01 ± 0.66	0.84 ± 0.17
Madelon	Single	0.02 ± 0.01	0.69 ± 0.04	510.90 ± 25.37	1744.28 ± 218.17	4.91 ± 0.18
	Ens.	0.10 ± 0.25	0.12 ± 0.23	218.35 ± 3.91	6.51 ± 13.51	0.54 ± 0.04
Spambase	Single	0.02 ± 0.02	0.20 ± 0.04	13.54 ± 3.89	0.12 ± 0.06	0.73 ± 0.12
	Ens.	0.01 ± 0.01	0.01 ± 0.00	8.67 ± 2.58	0.01 ± 0.01	0.06 ± 0.06
Yeast	Single	0.01 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.05 ± 0.03	0.30 ± 0.09
	Ens.	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.00	0.03 ± 0.03	0.03 ± 0.02
Isolet	Single	0.18 ± 0.01	8.35 ± 0.44	59.64 ± 0.38	2662.18 ± 249.78	179.35 ± 16.19
	Ens.	0.06 ± 0.01	0.20 ± 0.01	25.49 ± 0.14	37.82 ± 9.57	18.63 ± 0.35

Table 4.2: Average training times in seconds. Single method (Single) and ensemble (Ens.) strategies

Data set		IG	ReliefF	mRMR	SVM-RFE	FS-P
Connect4	Single	30.76 ± 0.54	30.70 ± 0.50	32.29 ± 0.49	33.92 ± 0.59	34.18 ± 0.60
	Ens.	30.76 ± 0.37	30.09 ± 0.53	33.08 ± 0.54	33.93 ± 0.68	34.16 ± 0.52
Madelon	Single	33.62 ± 3.50	33.17 ± 3.13	46.42 ± 3.46	31.71 ± 2.56	33.96 ± 2.94
	Ens.	34.42 ± 3.93	34.13 ± 3.36	53.46 ± 2.71	34.92 ± 3.37	34.29 ± 3.48
Spambase	Single	13.39 ± 1.24	20.08 ± 3.14	22.78 ± 2.24	12.50 ± 1.41	12.17 ± 1.52
	Ens.	13.28 ± 1.76	16.97 ± 3.07	22.76 ± 3.89	11.78 ± 1.77	12.23 ± 1.93
Yeast	Single	55.13 ± 4.99	55.13 ± 4.99	55.13 ± 4.99	54.31 ± 6.50	54.66 ± 4.33
	Ens.	60.30 ± 6.05	54.92 ± 4.72	54.92 ± 4.72	50.81 ± 4.92	54.31 ± 4.87
Isolet	Single	48.62 ± 2.30	58.38 ± 2.23	47.15 ± 1.71	51.58 ± 3.33	64.95 ± 4.53
	Ens.	49.04 ± 2.03	57.52 ± 1.36	49.96 ± 1.38	65.77 ± 2.32	72.66 ± 2.73

Table 4.3: 10% threshold: average estimated percentage test errors. Single method (Single) and ensemble (Ens.) strategies

It can be seen how the errors remain stable after the distribution process. The reduction in time is especially important for multivariate filters, which are the ones that usually provide the best results, e.g. mRMR.

Table 4.6 shows the variations in training time and error between the single method and the ensemble approach for the 50% threshold data. It can be seen how the average training time is

Data set		IG	ReliefF	mRMR	SVM-RFE	FS-P
Connect4	Single	26.34 ± 0.53	25.55 ± 0.46	32.05 ± 0.47	32.79 ± 1.32	34.00 ± 0.64
	Ens.	26.46 ± 0.47	25.62 ± 0.54	31.96 ± 0.55	33.18 ± 1.03	34.17 ± 0.47
Madelon	Single	36.08 ± 2.37	35.58 ± 2.74	49.83 ± 3.11	31.12 ± 2.89	33.08 ± 1.76
	Ens.	39.04 ± 7.81	38.58 ± 8.35	52.08 ± 4.50	37.92 ± 8.20	37.17 ± 8.75
Spambase	Single	17.67 ± 1.90	19.34 ± 3.69	15.65 ± 1.95	18.04 ± 3.99	14.45 ± 5.53
	Ens.	18.28 ± 1.61	16.63 ± 2.05	17.10 ± 1.29	13.11 ± 5.07	11.28 ± 4.16
Yeast	Single	53.17 ± 6.89	53.17 ± 6.89	53.17 ± 6.89	53.56 ± 6.45	53.03 ± 4.51
	Ens.	59.97 ± 3.89	53.91 ± 2.91	53.91 ± 2.91	51.42 ± 4.57	52.97 ± 2.96
Isolet	Single	43.68 ± 1.94	54.08 ± 3.64	46.72 ± 2.71	42.20 ± 1.77	61.45 ± 4.34
	Ens.	42.88 ± 1.67	57.73 ± 2.07	49.10 ± 2.34	59.51 ± 4.20	73.80 ± 4.95

Table 4.4: 25% threshold: average estimated percentage test errors. Single method (Single) and ensemble (Ens.) strategies

Data set		IG	ReliefF	mRMR	SVM-RFE	FS-P
Connect4	Single	24.82 ± 0.50	23.51 ± 0.50	30.94 ± 0.55	31.96 ± 2.06	32.61 ± 1.13
	Ens.	24.74 ± 0.60	23.32 ± 0.52	29.52 ± 1.04	31.37 ± 1.84	34.09 ± 0.67
Madelon	Single	39.00 ± 2.78	38.21 ± 2.44	39.17 ± 2.84	32.71 ± 2.94	34.92 ± 2.73
	Ens.	37.92 ± 2.94	38.46 ± 4.33	39.58 ± 3.80	38.92 ± 2.62	38.08 ± 3.14
Spambase	Single	16.95 ± 1.28	16.17 ± 1.15	13.17 ± 1.43	18.11 ± 1.66	16.06 ± 4.18
	Ens.	16.93 ± 1.67	17.78 ± 1.85	14.63 ± 1.63	17.93 ± 1.76	17.84 ± 1.92
Yeast	Single	54.05 ± 4.31	54.05 ± 4.31	54.05 ± 4.31	52.57 ± 6.65	54.80 ± 5.87
	Ens.	59.44 ± 7.30	53.10 ± 6.26	53.10 ± 6.26	51.42 ± 4.99	52.83 ± 6.10
Isolet	Single	37.98 ± 2.36	50.33 ± 3.45	46.75 ± 3.36	37.70 ± 3.12	48.39 ± 4.95
	Ens.	38.54 ± 2.51	48.07 ± 4.25	43.45 ± 3.06	47.98 ± 4.87	64.44 ± 9.74

Table 4.5: 50% threshold: average estimated percentage test errors. Single method (Single) and ensemble (Ens.) strategies

Data set		IG	ReliefF	mRMR	SVM-RFE	FS-P
Connect4	Time	-60%	-99%	-83%	-99%	-94%
	Error	0%	-1%	-5%	-2%	4%
Madelon	Time	80%	-83%	-57%	-99%	-89%
	Error	-3%	1%	1%	16%	8%
Spambase	Time	-50%	-95%	-36%	-92%	-92%
	Error	0%	9%	10%	-1%	10%
Yeast	Time	0%	-50%	0%	-40%	-90%
	Error	9%	-2%	-2%	-2%	-4%
Isolet	Time	-67%	-98%	-57%	-99%	-90%
	Error	1%	-2%	-7%	21%	25%

* Negative percentages are favorable to the ensemble.

Table 4.6: Variation in training time and error between single method and ensemble strategies (50% threshold)

greatly improved in most of the cases, while the error is close, except in some cases where the ensemble performs worse, and others where the ensemble even slightly improves the results of the single method.

4.3.2 Experimental study for the pure ensemble approach

For this part of the study, the same set of ranker methods described above —Information Gain, mRMR, ReliefF, SVM-RFE, and FS-P— is utilized. R rankings are generated using the aforementioned feature selection methods, all of them with the same training data. The pseudo-code of this approach can be seen in Algorithm 2.

The A_r outputs obtained from the different methods are combined using the Ranking SVM union method to obtain a single ranking list. Since the individual methods used for feature selection are rankers, it is necessary to establish a threshold T in order to obtain a practical subset of features. After obtaining this practical subset of features A_t, a SVM is used for checking the adequacy of the proposed ensemble in terms of classification error. The SVM utilizes a RBF kernel, with gamma 0.01, and C 1.

The performance of the proposed ensemble method is tested over the five well-known data

Algorithm 2 Pseudo-code of the proposed method

Inputs: number of ranker methods R, threshold of the number of features to be selected T, training set.
Output: classification prediction P.

1. For each r from 1 to R, obtain ranking A_r using method r.
2. Combine rankings A_r, $r = 1..R$ with Ranking SVM, obtaining A.
3. Select T first attributes from A, obtaining A_t.
4. Build a SVM classifier with the selected A_t attributes.
5. Obtain prediction P.

sets listed in Table 4.1. The experiment performed consisted of a comparison between the use of different feature selection methods individually and the use of an ensemble. Remember that all the feature selection methods used in this chapter are rankers, i.e. they do not select a subset of features, but they sort all the features. Therefore it is necessary to establish a threshold in order to obtain a practical subset of features. In this study, the same thresholds are used —10%, 25% and 50%—. Moreover, the ensemble is composed by six methods.

A Support Vector Machine (SVM) is chosen for checking the adequacy of the proposed ensemble in terms of classification error. A 10-fold cross validation is performed for estimating the error.

The next three tables (Tables 4.7, 4.8 and 4.9) display the average test errors. Having ten different errors as a result of the 10-fold cross validation, a Kruskal-Wallis test was applied to check if there were significant differences for a level of significance $\alpha = 0.05$ [65]. Then, a multiple comparison test (based on Tukey's honestly significant difference criterion [136]) is applied and those algorithms whose error average test results are not significantly worse than the best are labeled with a cross.

The experimental results demonstrate the adequacy of the proposed ensemble, since they match or improve upon the results achieved by the feature selection methods alone. It can be seen that, as the threshold is increased, the results obtained are not as positive. Despite this, the proposed ensemble obtains favorable results in four out of five data sets when the threshold is fixed to 25 % (indicated in Table 4.8). Finally, when the threshold is increased to 50 % (Table 4.9), only two out of five data sets have results not significantly different from the

Ranker method	Yeast	Spambase	Madelon	Connect4	Isolet
Ensemble	53.24 † ± 4.62	11.39 † ± 2.34	35.46 † ± 4.05	31.14 † ± 0.51	50.13 † ± 2.03
InfoGain	55.13 † ± 4.99	13.39 † ± 1.24	33.62 † ± 3.50	30.76 † ± 0.54	48.62 † ± 2.30
mRMR	55.13 † ± 4.99	22.78 ± 2.24	46.42 ± 3.46	32.29 ± 0.49	47.15 † ± 1.71
ReliefF	55.13 † ± 4.99	20.08 ± 3.14	33.17 † ± 3.13	30.70 † ± 0.50	58.38 ± 2.23
SVM-RFE	54.31 † ± 6.50	12.50 † ± 1.41	31.71 † ± 2.56	33.92 ± 0.59	51.58 † ± 3.33
FS-P	54.66 † ± 4.33	12.17 † ± 1.52	33.96 † ± 2.94	34.18 ± 0.60	64.95 ± 4.53

* The cross shows results that are not significantly different than the best.

Table 4.7: 10% threshold: average estimated percentage test errors

Ranker method	Yeast	Spambase	Madelon	Connect4	Isolet
Ensemble	53.24 † ± 4.62	19.19 † ± 2.00	36.21 † ± 4.42	26.96 † ± 0.67	48.77 ± 2.11
InfoGain	53.17 † ± 6.89	17.67 † ± 1.90	36.08 † ± 2.37	26.34 † ± 0.53	43.68 † ± 1.94
mRMR	53.17 † ± 6.89	15.65 † ± 1.95	49.83 ± 3.11	32.05 ± 0.47	46.72 † ± 2.71
ReliefF	53.17 † ± 6.89	19.34 † ± 3.69	35.58 † ± 2.74	25.55 † ± 0.46	54.08 ± 3.64
SVM-RFE	53.56 † ± 6.45	18.04 † ± 3.99	31.12 † ± 2.89	32.79 ± 1.32	42.20 † ± 1.77
FS-P	53.03 † ± 4.51	14.45 † ± 5.53	33.08 † ± 1.76	34.00 ± 0.64	61.45 ± 4.34

* The cross shows results that are not significantly different than the best.

Table 4.8: 25% threshold: average estimated percentage test errors

Ranker method	Yeast	Spambase	Madelon	Connect4	Isolet
Ensemble	53.24 † ± 4.62	16.93 ± 1.91	39.29 ± 2.65	25.27 ± 0.59	43.21 † ± 3.01
InfoGain	54.05 † ± 4.31	16.95 ± 1.28	39.00 ± 2.78	24.82 † ± 0.50	37.98 † ± 2.36
mRMR	54.05 † ± 4.31	13.17 † ± 1.43	39.17 ± 2.84	30.94 ± 0.55	46.75 ± 3.36
ReliefF	54.05 † ± 4.31	16.17 † ± 1.15	38.21 † ± 2.44	23.51 † ± 0.50	50.33 ± 3.45
SVM-RFE	52.57 † ± 6.65	18.11 ± 1.66	32.71 † ± 2.94	31.96 ± 2.06	37.70 † ± 3.12
FS-P	54.80 † ± 5.87	16.06† ± 4.18	34.92 † ± 2.73	32.61 ± 1.13	48.39 ± 4.95

* The cross shows results that are not significantly different than the best.

Table 4.9: 50% threshold: average estimated percentage test errors

Threshold	Yeast	Spambase	Madelon	Connect4	Isolet
10 %	$53.24^{\dagger} \pm 4.62$	$11.39^{\dagger} \pm 2.34$	$35.46^{\dagger} \pm 4.05$	31.14 ± 0.51	50.13 ± 2.03
25 %	$53.24^{\dagger} \pm 4.62$	19.19 ± 2.00	$36.21^{\dagger} \pm 4.42$	$26.96^{\dagger} \pm 0.67$	48.77 ± 2.11
50 %	$53.24^{\dagger} \pm 4.62$	16.93 ± 1.91	39.29 ± 2.65	$25.27^{\dagger} \pm 0.59$	$43.21^{\dagger} \pm 3.01$

* The cross shows results that are not significantly different than the best.

Table 4.10: Ensemble methods: average estimated percentage test errors

lowest average error. Even so, in the three data sets in which significative differences between the ensemble method and the best single method, it can be seen that the estimated percentage error of the ensemble is lower than the one presented by several single rankers.

However, if focusing on the behavior of the feature selection rankers individually (six last rows of each table), none of the six methods tested was able to significantly outperform the results obtained by the ensemble for all combinations. This fact proves that, although in some specific cases there is a single method that performs better than the ensemble, there is not a better feature selection ranker in general, and the ensemble seems to be the most reliable alternative when a feature selection process has to be carried out. Moreover, notice the adequacy of using Ranking SVM as a method to combine different rankings.

A last experiment is performed, consisting of the analysis of the behavior of the ensemble with the different thresholds, with independence of the actual feature selection methods. Table 4.10 displays the average test errors obtained with the different thresholds. A Kruskal-Wallis test plus Tukey's multiple comparison procedure was also applied and those algorithms whose error average test results are not significantly worse than the best are labeled with a cross.

This analysis demonstrates that an optimal threshold value does not exist such that its results stand out over the others. The three thresholds analyzed in this research show very similar results, since each one of the thresholds was significantly better than the others in three out of five data sets. Thus, it can be concluded that the most appropriate threshold depends on the nature of the data sets and their features. In this regard, the users cannot be released from this decision, and must select an appropriate threshold according to the particularities of each specific data set.

4.4 Summary

In the last few years, ensemble learning has been the focus of much attention mainly in classification tasks, based on the assumption that combining the output of multiple experts is better than the output of any single expert. This idea of ensemble learning can be adapted for feature selection, in which different feature selection algorithms act as different experts. In this chapter, two ways of building ensembles are explored: (a) N selections using the same feature selection algorithm, using different training data and (b) N selections using a variety of different feature selection algorithms, all using the same training data. Feature selection can also take advantage of data distribution. Most feature selection methods do not scale well when the number of features grows. Processing multiple subsets concurrently means that the training of feature selection methods is faster. This advantage is achieved with option (a). In both options, the results of the individual rankings are combined with SVM Rank, and the adequacy of the ensemble is subsequently tested using SVM as classifier. Results obtained in an experimental study performed over five UCI data sets show that both options are able to obtain good results. Option (a) improves training times over the individual feature selection methods, while maintaining errors. Option (b) obtains the best average results regardless of the data set and thresholds chosen. Notice the implications of this result, since it can release the user from the task of deciding which feature selection method is more appropriate for a given problem.

CHAPTER 5

A New Local Method for Classification Based on Information Theoretic Learning

The two previous chapters focused in new proposals for feature selection methods. This chapter is dedicated to the development of a new local classification method. The general aim, however, is the same: trying to confront diversity in data sets through the application of new ideas based on IT. The proposed algorithm performs classification based on the combination of neural networks by means of local modeling and techniques based on ITL [116] (See Sect. 2.2). First, a modified ITL clustering algorithm is applied in order to identify the local models. Second, since the problem is simplified by splitting it into smaller parts, a simple but effective model, the one-layer neural network, is applied. This approach is related to the one followed in the previous chapter, which dealt with ensemble learning applied to feature selection.

VQIT (Vector Quantization using Information Theoretic concepts) [86] is an information theoretic clustering algorithm that is able to distribute a set of nodes in such a way that the mutual information between the nodes and the data set is maximized. The result of this self-organizing task can be subsequently used for clustering or quantization purposes. In this chapter, VQIT is modified in order to perform classification tasks. This new algorithm is called FVQIT (Frontier Vector Quantization based on Information Theoretic concepts). It builds local models in a similar fashion to VQIT and then classifies using one-layer neural networks on each local model. In the first part of the chapter, the model for two-class (binary) classification is described. Later on, the concept utilized in the stage of local model building is expanded in order to being able to deal with muticlass problems. The contents of this chapter have been published in [92, 110, 111, 112, 113, 114].

The remainder of this chapter is organized as follows: Section 5.1 describes the VQIT method. Section 5.2 contains the binary version of FVQIT classification method. This version has been applied to several high dimensional problems, both in samples and features, such as intrusion detection and microarray gene expression. Section 5.3 contains the extension of FVQIT for multiclass problems, which has been studied over several microarray gene expres-

sion problems. Finally, Section 5.4 sums up the contents of the chapter.

5.1 Background: VQIT Clustering Algorithm

The VQIT (Vector Quantization Using Information Theory) clustering algorithm [86] is designed to take the statistical distribution of data into account. The objective is to place a series of nodes in the input space in such a way that the distribution of the nodes matches the distribution of the data. The algorithm considers that both data points and nodes are particles that have an information potential field associated. The information potential field created by a particle can be described by a kernel of the form $K(\cdot)$. The information potential field of data and nodes is of different sign, respectively. Placing a kernel on each particle (data point), the information potential energy at a point x in space is:

$$p(x) = \frac{1}{N}\sum_{i=1}^{N} K(x - x_i) \tag{5.1}$$

where N is the number of particles of a particular sign. If the kernel decays with distance $\left(K(x) \propto \frac{1}{(x-x_i)}\right)$ the potential is equivalent to physical potentials like gravitation and electric ones.

As there are two different types of particles (data and nodes), the energy of the system is defined by three terms:

1. Interactions between the data points: since the data points are fixed, these interactions have no influence over the energy.
2. Interactions between the data and the nodes: due to the opposite signs of the information potentials, these particles attract each other and maximize the correlation between the distribution of data and the distribution of nodes.
3. Interactions between nodes: the nodes' information potentials are of the same sign, which causes the nodes to repel each other. This helps to distribute the nodes across the input space, avoiding unnecessary concentrations on the same region of the input space.

Eq. (5.1) is Parzen density estimator [106]. In order to match the nodes with the data, (5.1) is used to estimate their PDF and then the divergence between them is minimized. Using Gaussian kernels, the distribution of the data points (x_i) is

$$f(x) = \sum_i G(x - x_i, \sigma_f) \tag{5.2}$$

The distribution of nodes (w_i) is

$$g(x) = \sum_i G(x - w_i, \sigma_g) \tag{5.3}$$

VQIT algorithm uses the Kullback-Leibler divergence, defined in (2.2). This divergence can be linearly approximated by the Cauchy-Schwarz inequality (C-S):

$$|f(x)g(x)| \leq ||f(x)||\ ||g(x)|| \tag{5.4}$$

Therefore, maximizing $\frac{|f(x)g(x)|}{||f(x)||\ ||g(x)||}$ is equivalent to minimizing the divergence between $f(x)$ and $g(x)$. Using logarithms in order to remove the division, the expression to minimize the divergence between the distributions $f(x)$ and $g(x)$ is the following:

$$\begin{aligned} D_{C-S}(f,g) &= -\log \frac{(\int f(x)g(x)dx)^2}{\int f^2(x)dx \int g^2(x)dx} \\ &= \log \int f^2(x)dx - 2\log \int f(x)g(x)dx + \log \int g^2(x)dx \end{aligned} \tag{5.5}$$

$V = \int g^2(x)dx$ is the information potential of the nodes, $C = \int f(x)g(x)dx$ is the cross information potential between the distributions of the data and the nodes, and $H = -\log \int g^2(x) = -\log V$ is the Renyi quadratic entropy of the nodes. In consequence, minimizing the divergence between f and g is equivalent to maximizing the sum of the entropy of the nodes and the cross information potential between the densities of the nodes and the data.

The algorithm uses the gradient descent method to minimize (5.5). This clustering algorithm is the basis for the classification algorithm proposed in the following section.

5.2 Learning Model for Binary Classification Problems

Using the ideas of VQIT, a supervised local classification algorithm for binary data sets is developed [92]. The method is composed of two stages. First, a set of nodes, which are points placed in the same space as data, are moved from their initial random positions to the

frontier between classes. This part of the algorithm is a modification of VQIT algorithm [86]. Second, a set of local models, associated to the nodes, based on one-layer neural networks are trained using the efficient algorithm described in [27], in such a way that a piecewise borderline between the classes is built. Therefore, the final system consists of a set of local experts, each of which will be trained to solve a subproblem of the original. In this manner, the method benefits from a finer adaptation to the characteristics of the training set. This architecture can be seen on Fig. 5.1. The following subsections describe both stages in detail.

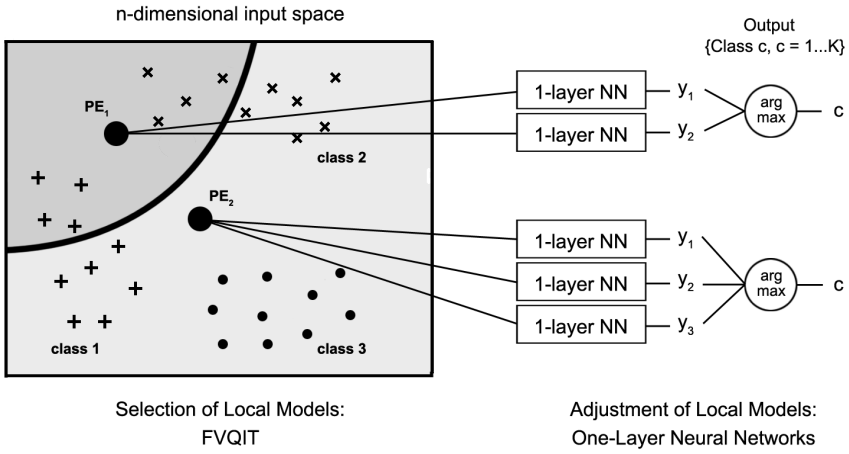


Figure 5.1: Architecture of the proposed learning model.

5.2.1 Creation of Local Models

The VQIT algorithm, which FVQIT is based on, was developed for vector quantization, that is, for representing a large data set with a smaller number of vectors in an appropriate way [86]. However, in our approach, the original algorithm has been modified in order to be able to build a piecewise representation of the borderline between classes in a classification problem. Therefore, the objective is placing a set of nodes on the frontier between the two classes, in such a way that each node will represent a local model.

The algorithm minimizes the energy function that calculates the divergence between the Parzen estimator of the distribution of data points and the estimator of the distribution of the nodes. Under this premise, a physical interpretation can be made. Both data points and nodes are considered two kinds of particles with a potential field associated. These fields induce repulsive and attractive interactions between particles, depending on its sign. In the original

VQIT algorithm, data and nodes had different signs. In FVQIT, data particles belonging to different classes have different signs. In this manner, a series of forces converge upon each node. Training patterns of a class exert an attractive force on a node and training patterns of the other class induce a repulsive force on it. Which class attracts and which class repels is decided using the Euclidean distance and k-NN (k-Nearest Neighbor) [28] as a rule of thumb. The closest class to the node (called 'own class') repels it and the furthest one attracts it. These roles alternate during the iterations as nodes move. An example of the movement of a node until it reaches its stability point can be seen in Fig. 5.2. Moreover, there exists a third force of repulsion between the nodes, which favors a better distribution, avoiding the accumulation of several nodes on a point.

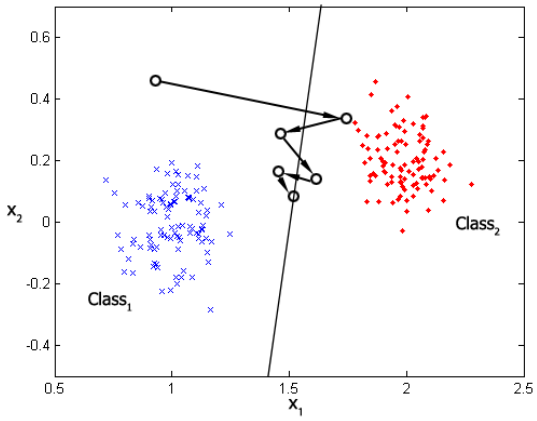


Figure 5.2: Evolution of a node from a random position to a position on the frontier between classes

In this context, the Parzen density estimators of the distribution of data points $f(\mathbf{x})$ and nodes $g(\mathbf{x})$ are:

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{N}\sum_{i=1}^{N} K\left(\mathbf{x}-\mathbf{x}_i, \sigma_f^2\right) \\ g(\mathbf{x}) &= \frac{1}{N}\sum_{i=1}^{N} K\left(\mathbf{x}-\mathbf{w}_i, \sigma_g^2\right) \end{aligned} \tag{5.6}$$

where N is the number of data points, K is any kernel function, σ_f^2 and σ_g^2 are the variances of the kernel functions, $\mathbf{x}_i \in \Re^n$ are data points, and $\mathbf{w}_i \in \Re^n$ are the weights associated to the nodes.

The function of energy $J(\mathbf{w})$ that calculates the divergence between the estimators is:

$$\begin{aligned} J(\mathbf{w}) = \quad & \log \int f^2(\mathbf{x})\,dx + 2\log \int f^+(\mathbf{x})\,g(\mathbf{x})\,dx \\ & -2\log \int f^-(\mathbf{x})\,g(\mathbf{x})\,dx + \log \int g^2(\mathbf{x})\,dx \end{aligned} \tag{5.7}$$

where $f^+(\mathbf{x})$ and $f^-(\mathbf{x})$ are the estimators of the distributions of data for each of the classes.

The first term of (5.7) is the information potential among data. Since data are stationary during the learning process, this term will not be considered from now on. The second and third terms are the crossed correlations between the distributions of data and nodes. The fourth term is the information potential of the nodes. Note that $H(\mathbf{x}) = -\log \int g^2(\mathbf{x})d\mathbf{x}$ is the Renyi quadratic entropy of the nodes. Consequently, minimizing the divergence between $f(\mathbf{x})$ and $g(\mathbf{x})$ is equivalent to maximizing the sum of the entropy of the nodes and the cross-information potentials between the distribution of data and nodes.

Assuming this formulation, when the nodes are placed on the minimum of the energy function $J(\mathbf{w})$, they are situated on a frontier area. Therefore, we utilize the gradient descent method to obtain the minimum of the function and, in consequence, to move the nodes toward such situation. To develop this, the derivative of (5.7) is calculated. For simplicity, the derivation of $J(\mathbf{w})$ is divided in three parts: (a) calculation of the contribution of the data from the own class (the closest one), (b) calculation of the contribution of the data from the other class (the furthest one) and (c) calculation of the contribution of the interactions between nodes.

Developing the last three terms in (5.7):

- Data from the own class:

$$\begin{aligned} C_+ &= \int f^+(\mathbf{x})g(\mathbf{x})dx \\ &= \frac{1}{MN_+}\int \sum_i^{N_+} G(\mathbf{x}-\mathbf{x}_i^+,\sigma_f^2)\sum_j^M G(\mathbf{x}-\mathbf{w}_j,\sigma_g^2)dx \\ &= \frac{1}{MN_+}\sum_i^{N_+}\sum_j^M \int G(\mathbf{x}-\mathbf{x}_i^+,\sigma_f^2)G(\mathbf{x}-\mathbf{w}_j,\sigma_g^2)dx \\ &= \frac{1}{MN_+}\sum_j^M\sum_i^{N_+} G(\mathbf{w}_j-\mathbf{x}_i^+,\sigma_a^2) \end{aligned} \tag{5.8}$$

  where M is the number of nodes, N_+ is the number of objects from the class of the node, $\mathbf{x}_i^+$ are the data from the own class, $\mathbf{w}_j$ are the weights of the nodes and the covariance of the Gaussian after integration is $\sigma_a^2 = \sigma_f^2 + \sigma_g^2$.

- Data from the other class:

$$\begin{aligned} C_- &= \int f^-(\mathbf{x}) g(\mathbf{x}) dx \\ &= \frac{1}{MN_-} \int \sum_i^{N_-} G(\mathbf{x}-\mathbf{x}_i^-, \sigma_f^2) \sum_j^M G(\mathbf{x}-\mathbf{w}_j, \sigma_g^2) dx \\ &= \frac{1}{MN_-} \sum_i^{N_-} \sum_j^M \int G(\mathbf{x}-\mathbf{x}_i^-, \sigma_f^2) G(\mathbf{x}-\mathbf{w}_j, \sigma_g^2) dx \\ &= \frac{1}{MN_-} \sum_j^M \sum_i^{N_-} G(\mathbf{w}_j-\mathbf{x}_i^-, \sigma_a^2) \end{aligned} \tag{5.9}$$

  where N_- is the number of objects from the class of the node, $\mathbf{x}_i^-$ are the data from the other class, $\mathbf{w}_j$ are the weights of the nodes and the covariance of the Gaussian after integration is $\sigma_a^2 = \sigma_{f^-}^2 + \sigma_g^2$.

- Interactions between nodes (entropy):

$$\begin{aligned} V &= \int g(\mathbf{x})^2 dx \\ &= \frac{1}{M^2} \sum_i^M \sum_j^M G(\mathbf{w}_i-\mathbf{w}_j, \sqrt{2}\sigma_g) \end{aligned} \tag{5.10}$$

  where $\mathbf{w}_i$ and $\mathbf{w}_j$ are the weights of the nodes.

The contributions to the gradient update for each of the previous terms in an iteration are:

- Data from the own class:

$$\frac{\partial}{\partial \mathbf{w}_k} 2\log C_+ = -2\frac{\nabla C_+}{C_+} \tag{5.11}$$

  where the term ∇C_+ denotes the derivative of C_+ with respect to $\mathbf{w}_k$.

$$\nabla C_+ = -\frac{1}{MN_+} \sum_i^{N_+} G(\mathbf{w}_k-\mathbf{x}_i^+, \sigma_a) \sigma_a^{-1} (\mathbf{w}_k-\mathbf{x}_i^+) \tag{5.12}$$

- Data from the other class:

$$\frac{\partial}{\partial \mathbf{w}_k} 2\log C_- = -2\frac{\nabla C_-}{C_-} \tag{5.13}$$

  where the term ∇C_- denotes the derivative of C_- with respect to $\mathbf{w}_k$.

$$\nabla C_- = -\frac{1}{MN_-} \sum_i^{N_-} G(\mathbf{w}_k-\mathbf{x}_i^-, \sigma_a) \sigma_a^{-1} (\mathbf{w}_k-\mathbf{x}_j^-) \tag{5.14}$$

- Interactions between nodes (entropy):

$$\frac{\partial}{\partial \mathbf{w}_k} 2\log V = \frac{\nabla V}{V} \tag{5.15}$$

where the term ∇V denotes the derivative of V with respect to $\mathbf{w}_k$.

$$\nabla V = -\frac{1}{M^2}\sum_{j}^{M} G(\mathbf{w}_j - \mathbf{w}_k, \sqrt{2}\sigma_g)\sigma_g^{-1}(\mathbf{w}_k - \mathbf{w}_j) \tag{5.16}$$

Therefore, using equations (5.11), (5.13) and (5.15), and through gradient descent, the weight update rule for the node $\mathbf{w}_k$ becomes:

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) - \eta\left(\frac{\nabla V}{V} + \frac{\nabla C_+}{C_+} - \frac{\nabla C_-}{C_-}\right) \tag{5.17}$$

where n is the iteration and η is the step size.

As with self-organizing maps, a good starting point is to choose high-variance kernels and a large η parameter such that all particles interact with one another. This allows a fast distribution of nodes along the feature space. Gradually, in order to obtain stability and a smooth convergence, the variances of the kernels and the parameter η are decreased or annealed at each step.

Once FVQIT is trained, the nodes, ideally, will find themselves well distributed on the frontiers between classes. Each node defines a region, a local model in the feature space which is in charge of classifying the data inside. Those models are defined by proximity: the local model associated to each node is composed of the nearest points (according to Euclidean distance) in the feature space, independently of their class. Therefore, data from both classes could coexist in the same local model. Algorithm 3 summarizes the pseudocode of the training process of FVQIT.

Algorithm 3 Training algorithm for the binary version of FVQIT

Inputs: Training set, number of nodes M, learning rate η, covariance matrices σ_f and σ_g, annealing rates η_{dec} and σ_{dec}, maximum number of iterations p, number of neighbors k.

1. Initialize the weights of the M nodes randomly in the data range.
2. Calculate which class repels the node and which class attracts it by calculating the Euclidean distances from each node to every data point and using the k-NN (k-Nearest Neighbor) rule.
3. Evaluate the cross information potential C_+ between each node and the data from the repelling class, as in (5.8).
4. Calculate the cross information potential C_- between each node and the data from the attracting class, using (5.9).
5. Evaluate the entropy V between nodes as described in (5.10).
6. Calculate the derivatives ∇C_+, ∇C_- y ∇V, utilizing (5.12), (5.14) and (5.16), respectively.
7. Evaluate the weight update for each node using (5.17).
8. Reduce learning rate η in the proportion shown by η_{dec}.
9. Reduce σ_f and σ_g in the proportion shown by σ_{dec}.
10. Repeat from 2 until the predefined maximum number of iterations p is reached.

The method employs several input parameters. Some of them can be assigned to a standard value or do not significantly affect the final performance of the method. The covariance matrices σ_f and σ_g are assigned to the covariance matrices of the patterns in the training set. This assignment is derived from the work in [86] and has obtained good results in the experiments in [92]. The parameter k of the k-NN (k-Nearest Neighbor) rule does not present a great impact on performance as its effect when the nodes are near the frontier between classes is compensated due to the subsequent moves of the nodes. It may take any typical value between 1 and 10. The parameter η controls the magnitude of node movements in each learning step. With high values, a significant oscillation of the nodes in the first learning steps will be observed and it will take longer to converge to a stable situation in the frontier. This parameter usually takes values in the interval $[range(X)/2, range(X)]$ being $range(X) = abs(\max(X) - \min(X))$

and being X the training set. η_{dec} and σ_{dec} control the smoothness of the convergence to the frontier. They may take a value in the interval $(0,1)$, although they typically take values close to 1 to ensure a smooth evolution. The maximum number of iterations p is a stopping condition added to the method. If a poor performance is observed, it can be increased to let the method converge to the frontier. The number of nodes M is usually selected using cross validation.

5.2.2 Adjustment of Local Models

In the first stage a set of local models was constructed by moving the nodes to their optimal position. Since each local model covers the closest points to the position of its associated node, the input space is completely filled, as input data are always assigned to a local model. In this second stage, the goal is to construct a classifier for each local model. This classifier will be in charge of classifying points in the region assigned to its local model and will be trained only with the points of the training set in this region.

As local modeling algorithms may suffer from temporal efficiency problems, caused by the process of training several local classifiers, we have decided to use a lightweight classifier. We have chosen one-layer neural networks, trained with the efficient algorithm presented in [27]. This algorithm allows rapid supervised training for one-layer feed-forward neural networks. The key idea is to measure the error prior to the nonlinear activation functions. In this manner, it is proven in [27] that the minimization based on the MSE can be rewritten in equivalent fashion in terms of the error committed prior to the application of the activation function, which produces a system of equations with $I+1$ equations and unknowns. This kind of systems can be solved computationally with a complexity of $O(M^2)$, where $M = I+1$ is the number of weights of the network. Thus, it requires much less computational resources than classic methods.

5.2.3 Operation of the Model

After the training process, when a new pattern arrives to be classified, the method first calculates the closest node $\mathbf{w}_k$ to a new pattern $\mathbf{x}_n$ using the Euclidean distance and then classifies it using the neural network associated to the local model $\mathbf{w}_k$.

In Fig. 5.3, a simple two-class bi-dimensional example is displayed. Data from one class is displayed with 'x'-mark and data from the other class with circles. FVQIT nodes are represented with squares. The division in local models is shown with dotted lines and the solid lines

depict the decision regions defined by each neural network.

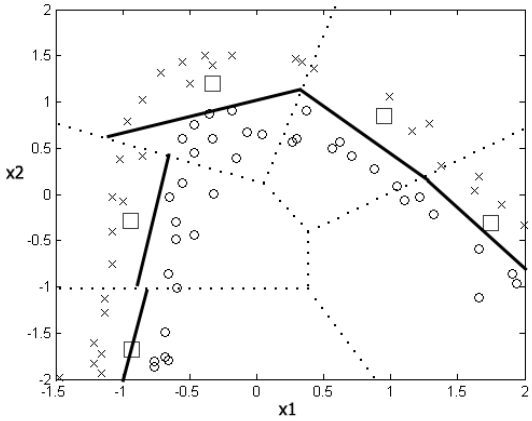


Figure 5.3: Example of operation of FVQIT. Local models and frontier between classes

5.2.4 Applications of the Binary Version

The binary version of FVQIT has been studied over several problems. First, an illustrative example over a two-dimensional problem; second, the study over several data sets from the UCI Machine Learning Repository [8]; third, the study on a very large real problem, intrusion detection, particularly the KDD Cup 99 data set, which has a very large amount of data; and last, the method is applied on a high dimensional real problem, microarray gene expression data sets, which have a very large amount of features (in the order of the thousands) and very few samples (in the order of the tens).

5.2.4.1 An Illustrative Example: 2D Spiral Classification Problem

To illustrate the power of the method in a visually perceptible problem, results for the classical 2D Spiral Classification problem are presented. This problem is highly non linear. It was reported in previous papers that, though being apparently simple, classical pattern recognition methods as multilayer perceptrons have problems when dealing with it [11, 45, 40].

The generated data set has 1200 two-dimensional patterns with a 50% for each class. A 5-fold cross validation is run to measure the accuracy of our method compared to a SVM with RBF kernel [30] and a Multilayer Perceptron (MLP) trained with the Scaled Conjugate

Gradient method [98]. To generate the data set, the code presented in the broadly known SVM Spider Toolbox 1.71 was taken [144]. It was added to the classical two spiral problem an uniform distributed random noise perturbation in the interval $[0, 0.35]$.

Figure 5.4 shows the final distribution of FVQIT nodes. It can be noted that the nodes are finally distributed along the border line between the two classes. The results obtained were satisfactory for both SVM and the proposed architecture. Both methods obtained an accuracy of 99.50% in test. In terms of efficiency, in this case, FVQIT solved the problem in 11.34 sec. while SVM solved the problem in 14.19 sec.

However, MLP was tested with a hidden layer from 5 up to 25 hidden neurons and in no case was capable of solving the problem. It obtained an accuracy of 50%, the same as random assignation of a class label in this case. Our results for the MLP are similar to those obtained in [11, 45, 40]. In these papers, it was stated that MLPs were not capable of solving this problem. The only way is to use a number of neurons in the hidden layer almost equal to the number of patterns to classify, highly increasing the complexity of the system.

Finally, the noise energy was increased to the interval $[0, 1.0]$ and our method and SVM were tested again with a 5-fold cross validation. As expected, the accuracy of both methods decreased, but while our method obtained an accuracy of 91.83% in 10.18 sec. the SVM obtained 89.83% in 22.55 sec.

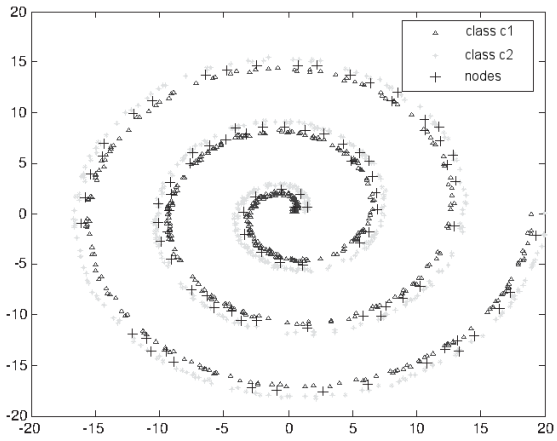


Figure 5.4: Final distribution of nodes for the 2D Spiral Problem

5.2.4.2 Real Data Sets from the UCI Repository

In this section, we discuss some experiments that demonstrate the performance of our method in databases. We coded the algorithm in Matlab® and ran these experiments on a 2.13 GHz Intel Pentium having 2GB of memory.

We tested our algorithms on three real life data sets available on UCI repository [8] and on the Wisconsin Data Mining Institute [146]. Their sizes and numbers of attributes and classes are detailed in Table 5.1. For the mushroom data set we used the transformation reported in [146], and for the adult data set the patterns with unknowns were deleted. Data sets including only pure categorical, and only pure numerical attributes were used, so as to test their influence in the results obtained. The proposed method was compared with the available results obtained by other methods, both regarding performance and training times. Accuracy is obtained using five 5-fold cross validation to evaluate the real error.

Data Set	**# of instances**	**# of numerical attributes**	**# of categorical attributes**	**# of classes**
Galaxy Dim	4192	14	0	2
Spambase	4601	57	0	2
Mushroom	8134	0	22	2

Table 5.1: Data sets used in the experiments

The SpamBase data set contains only numerical attributes, and it is a classification problem that aims at detecting whether a mail is spam or not. The data set has a reported a misclassification error of approximately 7%, which is in fact the error rate obtained by the other methods that are shown in Table 5.2, and that can be found in [33]. The methods tested are: FVQIT, AdaBoost + MLP, RL-Mix + MLP, Mixture of Experts + MLP, and MLP alone. As it can be seen, our method is the one that obtains the best performance results (less than 5% error). As training time was not available for the other methods, these results are not displayed in the table. For the case of the proposed method we employed an average time of 63 s.

The Mushroom data set contains only categorical attributes, and it is a binary classification problem. The data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The results corresponding to the RSVM

Method	% accuracy
FVQIT	**95.58%**
AdaBoost + MLP	93.52%
RL-Mix. + MLP	92.59%
Mix of Exp. + MLP	92.35%
MLP	91.67%

Table 5.2: Results for SpamBase data set

(Reduced Support Vector Machines) are reported in [85], while all the others were obtained implementing the methods in MatLab. The methods tested were: FVQIT, Reduced Support Vector Machines (RSVM) [85], Scaled Conjugate Gradient (SCG) [98] , Least-Squares Support Vector Machines (LS-SVM) [133], Proximal Support Vector Machines (PSVM) [53], one-layer neural networks [27], and Linear Discriminant Analysis (LDA) [44]. As can be seen in table 5.3, the proposed method is again the one that obtains the best performance results, while maintaining the efficiency.

Method	% accuracy	Training Time (sec.)
FVQIT	**89.24%**	13.68
RSVM	89.04%	466.20
SCG	81.63%	15.25
LS-SVM	80.90%	263.61
PSVM	80.79%	0.20
One-layer NN	80.77%	0.03
LDA	62.02%	0.08

Table 5.3: Results for Mushroom data set

The Galaxy Dimension Data set contains only numerical attributes. The aim for this data set is to classify stellar and non-stellar objects based on 14 image parameters computed for each object detected by the University of Minnesota Automated Plate Scanner (APS) operating in a threshold densitometry mode. The proposed method is again compared with several other methods, of which one-layer NN, PSVM, LDA, LS-SVM and SCG were implemented in Matlab, while the results of Minimal Support Vector Machine (MSVM), 1-Norm SVM and 1-Norm Support Vector Machine with Feature Selection (FSV) were extracted from [52]. Although the best training times are those of LDA, PSVM and the one layer neural network, these are linear methods which accuracy is significantly worse than the one of the proposed method.

Compared to the remaining non linear approaches, our method is the most efficient.

As can be seen on table 5.4, the proposed method obtains, once more, the best results regarding performance/efficiency.

Method	% accuracy	Training Time (sec.)
FVQIT	**94.94**%	6.63
SCG	94.82%	16.77
MSVM	94.70%	193.0
FSV	94.70%	541.0
1-Norm SVM	94.40%	774.0
One-layer NN	93.38%	0.02
LDA	93.37%	0.02
LS-SVM	92.21%	28.63
PSVM	92.53%	0.18

Table 5.4: Results for Galaxy Dimension data set

5.2.4.3 Experimental Study over Intrusion Detection

The KDD Cup 99 data set is a processed version of the DARPA 1998 data set, which was constructed from a simulation performed by the Defense Advanced Research Projects Agency (DARPA) through the Intrusion Detection Evaluation Program (IDEP) in 1998. The KDD Cup 99 data set was released for a classifier learning contest, which task was to distinguish between legitimate and illegitimate connections in a computer network [39], at the KDD (Knowledge Discovery and Data Mining) Conference in 1999. The training data set consists of about five million connection records (although a reduced training data set containing around five hundred thousand records exists) [87]. Each record contains values of forty one variables which describe different aspects of the connection, and the value of the class label (either normal, either the specific attack type). The test data set comprises three hundred thousand records and its data are not from the same probability distribution as training data.

Following the KDD Cup contest, the data set has been extensively used as a benchmark for developing machine learning algorithms for intrusion detection systems. The data set is very demanding not only because of its size but also due to the great inner variability among features. For those reasons, the KDD Cup 99 data set is a challenging classification problem.

Despite that KDD Cup 99 is a multiclass data set, it can be treated as a binary data set, simply by considering attack or no attack, instead of the different attack types. This approach is interesting in the sense that, most of the time, it is enough to distinguish between normal connections and attacks. This transformation has been carried out by other authors [6, 51], and there exist several results in the literature which are utilized as part of the comparative study.

The experimental study performed involves applying the proposed FVQIT algorithm to the binary version of the KDD Cup 99 data set [113]. As a preliminary stage, discretization and feature selection were both performed on the data set. The motivation for using discretization is that some features of the KDD Cup 99 data set present high imbalance and variability. This situation may cause a malfunction in most feature selection methods and classifiers. The problem is softened up by using discretization methods. In substance, the process of discretization involves putting continuous values into groups, by means of a number of discrete intervals. Two discretization methods will be employed in this study: PKID (Proportional k-Interval Discretization) [154] and EMD (Entropy Minimization Discretization) [41].

In order to reduce input dimensionality and improve the computational efficiency of the classifier, feature selection was performed. Filter methods were chosen because they are computationally cheaper than wrapper methods, and computational efficiency is a desirable feature given the large size of the data set [22]. The filters that will be used in this study are INTERACT [157] and Consistency based Filter [32]. These filters are widely used, with good results.

The discretization methods (PKID and EMD) are considered in combination with the above-named filters (INTERACT and Consistency-based). Thus, four combinations of discretizator plus filter are analyzed in order to check which subset of features works best with FVQIT method.

The model is trained with the KDD Cup 99 reduced training data set —494,021 samples— and is tested using the standard KDD Cup 99 test data set of 311,029 samples. Three performance measures are employed:

- Test Error (TE): indicates the overall percentage error rate for both classes (Normal and Attack).
- True Positive Rate (TP): shows the overall percentage of detected attacks.
- False Positive Rate (FP): indicates the percentage of normal patterns classified as attacks.

The results of the proposed method are compared with those obtained by other authors [6, 22, 39, 51], as can be seen in Table 5.5. Specifically, the classification methods to be compared are decision trees (C4.5), functional networks (FN), Support Vector Machines (SVM), ANalysis Of VAriance (ANOVA) (ANOVA ens.) and linear perceptrons (LP). Font in boldface indicates best results considering all three measures altogether. Table columns show the test error (TE), the true positive rate (TP), the false positive rate (FP) and the number of features employed (NF). Both error and rates are shown in percentage (%). These measures (TE, TP and FP) are typical in the field of intrusion detection.

As can be seen in Table 5.5, the combination PKID+Cons +FVQIT obtains the best result as it improves the performance obtained by the KDD Cup Winner in all three measures used, using a considerably reduced number of features (six instead of the forty one original features).

In addition, this combination outperforms all other results included in this study. Despite the fact that individual values of error and TP for the combination EMD+Cons +FVQIT are better than those for the above mentioned combination —4.73 versus 5.95 and 94.50 versus 92.73—, it must be noted that the variations in percentage between these quantities are quite small —20% and 2% respectively— in contrast to the variation between the values of FP —1.54 versus 0.48 (300%)—. On the other hand, error and TP for EMD+ INT+FVQIT, EMD+Cons+FVQIT, and PKID+INT+FVQIT are good, but unfortunately at the expense of FP, which happens to be high for all of them.

5.2.4.4 Experimental Study over Microarray Gene Expression

In this experimental study, FVQIT classifier is employed to classify twelve DNA gene-expression microarray data sets of different kinds of cancer. These data sets present features of the order of thousands and very few samples (tens or hundreds). A comparative study with other well-known classifiers is carried out [111, 112]. The number of features and samples for each data set are shown in Table 5.6.

Since the number of input features of these kind of data sets is huge, as can be seen on Table 5.6, feature selection is applied again, as in the previous problem [124]. Two different kinds of filter methods are employed: subset filters and rankers. Subset filters provide a subset of selected features, while rankers make use of a scoring function in order to build a feature ranking, where all features of the data set are sorted in decreasing relevance order. In the first experiment (subset filters), the performance of the method is tested. The aim of the second

Method	TE(%)	TP(%)	FP(%)	NF
PKID+Cons+FVQIT	**5.95**	**92.73**	**0.48**	**6**
EMD+INT+FVQIT	5.40	93.50	0.85	7
EMD+Cons+FVQIT	4.73	94.50	1.54	7
PKID+INT+FVQIT	5.68	93.61	2.75	7
KDD Winner	6.70	91.80	0.55	41
PKID+Cons+C4.5	5.14	94.08	1.92	6
EMD+INT+C4.5	6.69	91.81	0.49	7
FNs_poly	6.48	92.45	0.86	41
FNs_fourier	6.69	92.72	0.75	41
FNs_exp	6.70	92.75	0.75	41
SVM Linear	6.89	91.83	1.62	41
SVM RBF	6.86	91.83	1.43	41
ANOVA ens.	6.88	91.67	0.90	41
LP 2cl.	6.90	91.80	1.52	41

Table 5.5: KDD Cup data set: results obtained by the four versions of the proposed method and by other authors

experiment (ranker methods), is to check the stability of the performance reached by FVQIT, independently of the number of features selected.

Experiment 1: Study of Performance Using Subset Filters In the first experimental setting, FVQIT method is compared with other classifiers with the objective of finding out which classifier obtains the best performance. Thus, five well-known machine learning classifiers —naive Bayes (NB), k-Nearest Neighbor (k-NN), C4.5, Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP)— are also applied over the filtered data sets. The implementation of these methods can be found in [88], except for MLP, where the Matlab Neural Networks Toolbox was used. Three filters have been chosen in order to consider different behaviors. In previous works, values obtained by filters were shown to be influenced by discretization [18], thus in consequence we are using two discretizers —Entropy Minimization Discretization (EMD) [41] and Proportional k-Interval Discretization (PKID) [154]— in combination with the subset filters CFS (Correlation-based Feature Selection) [61], Consistency-based Filter [32] and INTERACT [157], which can be found in the Weka tool [147].

Data set	No. features	Total samples
Brain [102]	12,625	21
Breast [138]	24,481	97
CNS [109]	7,129	60
Colon [5]	2,000	62
DLBCL [4]	4,026	47
GLI [63]	22,283	85
Leukemia [55]	7,129	72
Lung [56]	12,533	181
Myeloma [134]	12,625	173
Ovarian [108]	15,154	253
Prostate [130]	12,600	136
SMK [132]	19,993	187

Table 5.6: Description of the binary microarray data sets

The data sets have been divided using $2/3$ for training and $1/3$ for test. A 10-fold cross-validation has been performed on the training sets, in order to estimate the validation error to choose a good configuration of parameters. The results of FVQIT have been compared with those obtained by other classifiers. Table 5.7 shows the estimated test errors (TE in the table) as well as the sensitivity (Se) and specificity (Sp) rates —in percentage— and the number of features (NF) selected by each method tested. Moreover, the ranking is displayed between parentheses. The ranking assigns a position between 1 and 6 to each method in each data set, taking into account the ties among them. Also, the best error obtained for each data set is emphasized in bold font. Despite having executed all six combinations of discretizer + filter, only the best result for each classifier in each data set is shown.

As can be seen in Table 5.7, FVQIT obtains good performance on all data sets, with an adequate number of selected features. Specially remarkable are the results obtained on the data sets DLBCL and Leukemia, where FVQIT classifier is the only method able to achieve 0% of test error. The result obtained on the Prostate data set is also important. Its test set is unbalanced (26% of one class and 74% of the other). C4.5, naive Bayes and k-NN are assigning all the samples to the majority class and SVM is assigning all the samples to the minority class, whereas FVQIT is able to do something different and better, which results in a lower test error.

In Table 5.8 the average rankings (obtained from the rankings displayed in Table 5.7 between parentheses) are shown. In average, the proposed method is clearly preferable above the

Data set		FVQIT	SVM	NB	MLP	k-NN	C4.5
Brain	TE	**0.00** (1)	14.29 (4)	14.29 (4)	28.57 (6)	**0.00** (1)	**0.00** (1)
	Se	100.00 (1)	100.00 (1)	100.00 (1)	0.00 (6)	100.00 (1)	100.00 (1)
	Sp	100.00 (1)	83.33 (4)	83.33 (4)	71.43 (6)	100.00 (1)	100.00 (1)
	NF	1	45	45	1	1	45
Breast	TE	**21.05** (1)	**21.05** (1)	26.32 (5)	**21.05** (1)	26.32 (5)	**21.05** (1)
	Se	75.00 (5)	83.33 (1)	83.33 (1)	83.33 (1)	83.33 (1)	66.70 (6)
	Sp	85.71 (2)	71.43 (3)	57.10 (5)	71.43 (3)	57.10 (5)	100 (1)
	NF	17	119	5	17	5	3
CNS	TE	**25.00** (1)	35.00 (3)	**25.00** (1)	35.00 (3)	35.00 (3)	35.00(3)
	Se	69.20 (3)	71.43 (2)	69.20 (3)	68.75 (6)	69.20 (3)	76.90 (1)
	Sp	85.70 (1)	50.00 (4)	85.70 (1)	50.00 (4)	57.10 (3)	42.90 (6)
	NF	4	60	4	60	4	47
Colon	TE	**10.00** (1)	**10.00** (1)	15.00 (3)	40.00 (6)	15.00 (3)	15.00 (3)
	Se	80.00 (4)	80.00 (4)	87.50 (1)	50.00 (6)	87.50 (1)	87.50 (1)
	Sp	100.00 (1)	100.00 (1)	83.30 (3)	61.11 (6)	83.30 (3)	83.30 (3)
	NF	12	12	3	12	3	3
DLBCL	TE	**0.00** (1)	6.67 (2)	6.67 (2)	6.67 (2)	6.67 (2)	13.33 (6)
	Se	100.00 (1)	100.00 (1)	85.70 (4)	100.00 (1)	85.70 (4)	85.70 (4)
	Sp	100.00 (1)	88.89 (4)	100.00 (1)	88.89 (4)	100.00 (1)	87.50 (6)
	NF	36	36	36	47	36	2
GLI	TE	**10.71** (1)	14.29 (3)	**10.71** (1)	17.86 (5)	14.29 (3)	21.43 (6)
	Se	85.71 (1)	85.00 (3)	85.71 (1)	78.26 (5)	81.82 (4)	75.00 (6)
	Sp	100.00 (1)	87.50 (6)	100.00 (1)	100.00 (1)	100.00 (1)	100.00 (1)
	NF	113	23	23	23	122	3
Leukemia	TE	**0.00** (1)	2.94 (2)	5.88 (3)	5.88 (3)	8.82 (6)	5.88 (3)
	Se	100.00 (1)	100.00 (1)	100.00 (1)	92.86 (5)	100.00 (1)	92.86 (5)
	Sp	100.00 (1)	95.24 (2)	90.00 (5)	95.00 (3)	90.00 (5)	95.00 (3)
	NF	2	18	18	2	1	2
Lung	TE	0.67 (2)	1.34 (4)	4.70 (5)	0.67 (2)	**0.00** (1)	18.12 (6)
	Se	100.00 (1)	99.26 (3)	94.80 (5)	99.26 (3)	100.00 (1)	82.80 (6)
	Sp	93.75 (4)	93.33 (5)	100.00 (1)	100.00 (1)	100.00 (1)	73.30 (6)
	NF	40	40	1	40	40	1
Myeloma	TE	21.05 (2)	21.05 (2)	21.05 (2)	21.05 (2)	29.82 (6)	**19.30** (1)
	Se	84.00 (1)	81.48 (3)	81.48 (3)	80.36 (6)	82.20 (2)	80.70 (5)
	Sp	42.86 (1)	33.33 (2)	33.33 (2)	0.00 (5)	25.00 (4)	0.00 (5)
	NF	2	40	2	2	2	2
Ovarian	TE	**0.00** (1)	**0.00** (1)	**0.00** (1)	**0.00** (1)	**0.00** (1)	1.19 (6)
	Se	100.00 (1)	100.00 (1)	100.00 (1)	100.00 (1)	100.00 (1)	98.10 (6)
	Sp	100.00 (1)	100.00 (1)	100.00 (1)	100.00 (1)	100.00 (1)	100.00 (1)
	NF	3	3	3	17	3	
Prostate	TE	**20.59** (1)	73.53 (6)	26.47 (3)	23.53 (2)	26.47 (3)	26.47 (3)
	Se	56.25 (2)	26.47 (3)	0.00 (4)	100.00 (1)	0.00 (4)	0.00 (4)
	Sp	100.00 (1)	0.00 (6)	100.00 (1)	75.76 (5)	100.00 (1)	100.00 (1)
	NF	64	3	2	3	2	2
SMK	TE	**25.81** (1)	33.87 (3)	40.32 (6)	32.26 (2)	33.87 (3)	33.87 (3)
	Se	78.79 (2)	71.88 (4)	67.85 (6)	89.47 (1)	75.00 (3)	68.42 (5)
	Sp	68.97 (1)	60.00 (3)	52.94 (6)	58.14 (5)	58.82 (4)	62.50 (2)
	NF	21	3	3	21	21	3

Table 5.7: Best estimated test errors (TE), sensitivity (Se), specificity (Sp) and number of features selected (NF). The rankings are displayed between parentheses

other methods studied. It is shown that the proposed method is the most specific (it correctly identifies most of the negatives) and the most sensitive (it correctly identifies most of the positives). Therefore, in light of the above, we can conclude that FVQIT classifier is suitable to be combined with discretizers and filters to deal with problems with a much higher number of features than instances, such as DNA microarray gene-expression problems.

Measure	FVQIT	SVM	NB	MLP	k-NN	C4.5
TE	**1.17**	2.67	3.00	2.92	3.08	3.50
Sensitivity	**1.92**	2.25	2.58	3.50	2.17	4.17
Specificity	**1.33**	3.42	2.58	3.67	2.92	3.00

Table 5.8: Average rankings of error, sensitivity and specificity for all data sets

Experiment 2: Study of Performance Stability Using Rankers When using feature selection, sometimes it is difficult to compare performance between classifiers because there are two variables involved: test error and number of features selected. Depending on the application, sometimes it may be desirable to choose the minimum test error regardless the number of features, but sometimes a somewhat larger error may be accepted in the interest of a smaller number of features. In this context, the aim of the second experiment is to check the stability of the performance reached by FVQIT classification method independently of the number of features selected. Therefore, in this case, it is advisable to utilize rankers, so as to compare the performance of the classifiers in a wide range of selected features. Four rankers have been chosen in order to consider different behaviors. The ranker methods we have chosen are the following, the implementation of which can be found in [88]: Fisher Score [37], Chi-square [89], Information Gain [29], and mRMR (Minimal Redundancy Maximal Relevance) [35].

Since ranker methods provide a sorted list of features according to a score, there is a decision to make regarding the number of features to be selected. As of this, in this experiment we are going to test the classifiers with different numbers of features. Thus, we are going to select the first 1, 3, 5, 10, 15, 20, 30, 40, 50 and 100 features from the sorted list of features that the rankers provide.

First, the overall results of the comparative study for each data set are presented and then we focus on the overall results for each feature number. As the number of experimental results is very large (all the combinations of four rankers, seven classifiers and ten different feature numbers over twelve data sets account for 3360 experiments), some summary of results needs

to be used. In a similar way as in the first part of the experimental section, the methods are sorted in a table using a ranking in which ties have been taken into consideration. The average rankings of test error, sensitivity and specificity for all twelve data sets are represented on Table 5.9. As can be seen, FVQIT method is the classifier that obtains the best average performance for all data sets, as well as the best sensitivity and specificity. However, FVQIT does not obtain the best performance in every data set. Since these data sets are a hard challenge, obtaining the best result in average is an important achievement for FVQIT, especially when comparing it with popular and well-tested methods such as the ones employed in this work.

Data set	FVQIT			SVM			NB			MLP			k-NN			C4.5		
	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp
Brain	2.1	1.4	2.6	2.3	2.4	2.5	2.9	3.3	2.5	3.7	3.4	3.2	3.9	4.6	3.9	1.3	3.5	1.0
Breast	2.3	2.6	2.4	2.4	2.6	2.8	3.3	3.7	3.7	4.0	4.4	3.9	3.5	3.8	3.6	2.9	2.8	2.9
CNS	1.6	2.2	3.0	2.9	5.4	1.0	4.6	3.8	4.9	3.3	3.5	3.9	2.4	2.8	3.5	2.3	2.6	3.5
Colon	2.0	3.3	2.1	6.0	1.0	6.0	1.4	2.6	1.4	2.0	3.0	2.3	3.5	4.4	3.6	2.0	3.1	2.0
DLBCL	1.4	1.0	1.9	2.1	1.1	2.3	1.7	1.7	1.7	1.4	1.7	1.4	2.7	2.0	3.3	3.6	4.2	3.6
GLI	1.1	1.4	2.2	6.0	6.0	1.0	1.5	1.8	2.9	2.2	2.7	2.5	2.0	2.4	2.8	4.5	4.5	5.9
Leukemia	1.3	2.3	1.3	6.0	1.0	6.0	1.8	3.3	1.4	3.3	2.5	3.8	1.2	2.1	1.6	4.2	5.6	3.9
Lung	1.9	2.3	2.6	4.5	6.0	1.0	2.0	1.2	2.9	2.4	2.1	3.1	2.3	1.8	3.2	5.2	5.0	5.9
Myeloma	3.9	2.5	3.4	1.5	3.3	2.1	5.3	5.5	5.3	1.6	3.5	2.6	4.5	3.1	4.3	2.5	2.6	2.5
Ovarian	2.2	1.3	2.1	1.1	1.2	1.0	3.8	2.0	3.6	1.5	1.5	1.2	4.7	2.3	4.7	2.2	2.0	1.5
Prostate	1.6	2.5	1.9	3.1	1.3	3.9	4.9	4.8	3.4	1.9	3.5	2.4	1.8	3.5	2.6	4.0	4.9	2.9
SMK	3.7	3.8	3.9	1.7	2.7	2.1	2.5	3.0	3.0	4.0	2.4	4.3	3.4	4.1	3.8	3.9	4.8	3.7
Average	**2.09**	**2.22**	**2.45**	3.30	2.83	2.64	2.98	3.06	3.06	2.61	2.85	2.88	2.99	3.07	3.41	3.22	3.80	3.28

Table 5.9: Average ranking of test error (TE), sensitivity (Se) and specificity (Sp) for all data sets

In a second step, the results in function of the number of features are analyzed. Again, the same processing is made, in such a way that the average ranking of test error, sensitivity and specificity for all features are represented in Table 5.10.

On Table 5.10 can be seen how FVQIT classifier outperforms the other methods for all feature numbers except for 100 features, where it obtains the second best result, behind of MLP. In light of the above it can be concluded that FVQIT is the most stable classifier because it obtains good results both with few and many features, in contrast with other classifiers. For instance, k-NN performs correctly between 15 and 50 features but it does not obtain good results with smaller numbers (less than 15) and higher ones (100). On the other hand, C4.5 performs adequately with few features but its performance decreases when the number of features increases. Last, MLP shows stable behavior for all the feature numbers (although it is better for few features), but, in average, FVQIT performs better. Besides, FVQIT method is the most sensitive and specific in average. For further details, please refer to [112].

No. features	FVQIT			SVM			NB			MLP			k-NN			C4.5		
	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp	TE	Se	Sp
1	1.7	2.0	1.8	2.9	2.7	2.4	2.4	2.1	2.6	1.9	2.7	2.7	3.6	4.0	3.7	2.3	3.4	2.1
3	2.0	3.1	2.5	2.9	2.9	2.2	2.8	2.8	2.8	2.3	2.7	2.4	3.6	2.9	3.3	2.6	3.6	2.9
5	2.3	1.9	2.6	3.3	3.3	3.1	2.7	3.3	2.8	2.4	3.4	2.4	3.6	3.8	3.8	2.4	3.3	2.8
10	2.3	2.1	2.7	3.3	3.0	2.4	3.0	2.9	3.3	2.8	3.1	3.3	3.2	3.4	3.9	3.2	3.3	3.4
15	2.3	2.8	2.5	3.3	2.9	2.5	3.2	3.3	3.4	2.8	2.7	3.2	3.1	2.7	3.8	3.4	3.2	3.6
20	2.2	2.2	2.6	3.8	2.8	3.2	3.0	3.3	2.8	3.0	3.2	3.2	2.8	2.8	3.3	3.1	3.8	3.1
30	1.8	1.8	2.2	3.3	2.6	2.7	3.3	3.8	3.5	3.2	2.4	3.3	2.6	3.0	3.2	3.7	4.0	3.7
40	2.2	2.4	2.3	3.4	2.8	2.8	2.9	2.8	3.0	2.7	2.7	3.2	2.3	2.7	2.8	3.5	4.2	3.3
50	1.8	1.4	2.6	3.2	2.7	2.5	3.2	3.1	3.2	2.8	3.3	3.2	2.1	2.8	2.6	4.2	5.0	4.1
100	2.3	2.5	2.8	3.5	2.8	2.8	3.3	3.4	3.4	2.1	2.5	2.2	3.1	2.8	3.6	3.8	4.3	3.9
Average	**2.09**	**2.22**	**2.45**	3.30	2.83	2.64	2.98	3.06	3.06	2.61	2.85	2.88	2.99	3.08	3.41	3.22	3.80	3.28

Table 5.10: Average ranking of test error (TE), sensitivity (Se) and specificity (Sp) for all features

5.3 Extension for the Multiclass Problem

In this section, the previous binary FVQIT algorithm is extended to deal with multiclass scenarios. The training process of multiclass FVQIT is very similar to the binary one. In the first stage of the training process of the binary version, the closest class to each node in each iteration repelled the node and the other class attracted it. In the multiclass version, for each node, the two nearest classes are chosen using the same k-NN (k-Nearest Neighbor) rule of thumb. From among them, the closest one repels the node; the second closest one attracts it (Alg. 4); the other classes have no effect. The rest of the training of the first stage is the same as in binary FVQIT, employing the two closest classes in order to generate the crossed information potentials (see Sect. 5.2).

Algorithm 4 Mechanism of selection of the classes that attract and repel

Inputs: Training set, number of classes, distance from each node to each data point, number of neighbors k.

1. For each node $\mathbf{w}_i$,
   (a) Sort the data points by increasing Euclidean distance to the node.
   (b) Take the classes of the k closest points and calculate its mode. The mode will be the repelling class for that node.
   (c) Take the classes of the k closest points to each node that do not belong to the repelling class and calculate its mode. The mode will be the attracting class for that node.

In the second stage of the training of binary FVQIT, a one-layer neural network was trained in each local model. In the multiclass version, instead of just one neural network, we will have several one-layer neural networks in each local model, each of them associated with one of the classes of the problem. In each local model there can exist a variable number of one-layer neural networks according to the number of classes of the data in that model. Por instance, if a local model contains one hundred data which belong to four classes, it will have four associated networks. If another model has two hundred data classified in five classes, it will have five networks. Thereafter, the training is performed following a one-versus-rest strategy, that is to say, each neural network is trained to recognize the patterns of "its" class against the points of the rest of classes.

Once the model is trained, when a new pattern needs to be classified, in binary FVQIT the pattern was assigned to the nearest local model (using Euclidean distance) and the associated network classified it into one of the two classes. In multiclass FVQIT, the pattern is assigned to a local model in the same manner. However, after that, the outputs of the one-layer neural networks associated to this local model are evaluated.

The pattern is classified in the class associated to the network that produces the highest output ($c_i = \arg\max_j net_j$).

5.3.1 Results of the Multiclass Version

The multiclass version of FVQIT has been applied over several real world data sets. In the following sections, two experimental studies are described. First, the study over several data sets from the UCI Machine Learning Repository [8]. Second, and analogously to the previous binary version, the study over several microarray gene expression data sets.

5.3.1.1 Real Multiclass Data Sets from the UCI Repository

In this subsection, a comparative study in terms of the test error between the proposed classifier and other representative techniques of the field is performed. These techniques are: k-NN, Naive Bayes, C4.5, MLP, SVM and Bagging C4.5. The study uses several benchmark data sets, obtained from the UCI Machine Learning Repository [8], which are shown, along with a brief description of their main characteristics, on Table 5.11. These data sets have been selected with the aim of achieving variety of the number of samples, features and classes.

Data set	Number of samples	Number of features	Number of classes
Iris	150	4	3
Wine	178	13	3
Glass Identification	214	9	6
Vowel Recognition	990	10	11
Image Segmentation	2310	18	7
Landstat Satellite	6435	36	6
Letter Recognition	20000	16	26

Table 5.11: Data sets employed in the first experiment of the multiclass version

The methodology utilized for the comparative study is the *k*-fold cross-validation. In this work, $k = 10$ is taken, as recommended by [143]. However, some of these data sets are already split up in training and test set. In these cases (Vowel Recognition and Landstat Satellite data sets) this approach is respected and the *k*-fold cross-validation technique is not used so as to be able to compare our results with those of other authors. The parameters for SVM and FVQIT have been tuned up.

Table 5.12 shows the errors committed, in percentage, by each method on each data set. Best results are enhanced in bold font. The last column shows the average error committed by each method in the experimental study. In Table 5.13, it can be observed the ranking for each method on each comparative of the data sets. The last column of the table shows the average position of each method in the ranking. In these tables can be seen how FVQIT achieves the best test error in average and obtains the best average ranking as well.

Classifier	Iris	Wine	Glass	Vowel	Image	Landstat	Letter	Average
FVQIT	**1.33**	**0.55**	**26.65**	46.10	4.07	11.45	10.16	**14.33**
Bagging	4.67	3.33	28.82	46.54	**2.38**	12.90	5.85	14.93
SVM	2.67	0.56	28.40	**43.94**	3.25	**8.55**	17.68	15.01
k-NN	4.01	3.33	32.23	49.35	3.38	11.50	**3.90**	15.39
MLP	3.33	2.25	30.84	51.08	3.94	12.90	17.41	17.39
C4.5	5.33	8.99	34.59	54.76	2.94	15.30	11.94	19.12
Naive Bayes	5.33	2.81	50.04	48.27	19.70	20.50	35.91	26.08

Table 5.12: Error committed (%) by each method on each benchmark data set

Classifier	Iris	Wine	Glass	Vowel	Image	Landstat	Letter	Average
FVQIT	**1st**	**1st**	**1st**	2nd	6th	2nd	3rd	**2.29**
SVM	2nd	2nd	2nd	**1st**	3rd	**1st**	6th	2.43
Bagging	5th	6th	3rd	3rd	**1st**	4th	2nd	3.43
k-nn	4th	5th	5th	5th	4th	3rd	**1st**	3.86
MLP	3rd	3rd	4th	6th	5th	4th	5th	4.29
C4.5	6th	7th	6th	7th	2nd	6th	4th	5.43
Naive Bayes	6th	4th	7th	4th	7th	7th	7th	6.00

Table 5.13: Ranking for each method on the comparative study of benchmark data sets

5.3.1.2 Experimental Study over Multiclass Microarray Gene Expression

In this study, five multiclass DNA microarray data sets have been chosen. The main characteristics of these data sets are shown on Table 5.14. Three of them (CLL-SUB, GLA-BRA and TOX) have been obtained from the web site of feature selection of the Arizona State University [88]. The remaining data sets (GCM and Lymphoma) are available at the Broad Institute Cancer Program Data Sets Repository [67]. The methods compared with FVQIT are the following: MLP, SVM —note that a one-versus-all strategy is used—, k-NN, NB, and C4.5.

Data set	Number of samples	Number of features	Number of classes
CLL-SUB	74	11,340	3
GCM	144	16,063	14
GLA-BRA	120	49,151	4
Lymphoma	64	4,026	9
TOX	114	5,748	4

Table 5.14: Multiclass DNA microarray data sets employed in the experiment

As can be seen, the multiclass DNA microarray data sets also present many more features than instances. Therefore, again, feature selection methods are utilized. For this experiment, the INTERACT filter [157] is applied to those data sets as a preprocessing step in order to make them manageable. This filter has been previously used with success on binary microarray data sets [111]. The number of features selected for each data set can be seen on Table 5.15.

Data set	No. features
CLL-SUB	61
GCM	78
GLA-BRA	150
Lymphoma	160
TOX	80

Table 5.15: Number of features selected by the INTERACT filter

The data sets have been divided using $2/3$ for training and $1/3$ for testing. Table 5.16 shows the estimated test errors (in percentage) for each classifier and data set.

Classifier	CLL-SUB	GCM	GLA-BRA	Lymphoma	TOX	**Average**
FVQIT	**21.62**	45.65	**33.33**	**12.50**	**12.28**	**26.41**
k-NN	29.73	54.35	41.67	15.63	22.81	32.84
Naive Bayes	27.03	50.00	36.67	40.63	26.32	36.13
SVM	37.84	73.91	48.33	25.00	15.79	40.17
MLP	45.95	**39.13**	35.00	43.75	38.60	40.49
C4.5	43.24	63.04	55.00	46.88	52.63	52.16

Table 5.16: Error committed (%) by each method on each multiclass DNA microarray data set

A 10-fold cross-validation is performed upon the training sets in order to choose a good configuration of parameters. The k in the k-NN method ranges from 1 to 5. The SVM utilizes a Radial Basis Function kernel and its parameters C and γ range from 1 to $10,000$ and 0.1 to 40, respectively. The MLP (Multi-Layer Perceptron) has one hidden layer which contains between 3 and 50 neurons. FVQIT utilizes between 10 and 40 nodes, 100 iterations, initial η between 1 and 5 and η decrement between 0.7 and 0.99.

On Table 5.16 can be seen that FVQIT obtains the best test errors in four out of five data sets. On table 5.17 a ranking of the performance results for all the compared methods is shown. The ranking assigns a position between 1 and 6 to each method for each data set. The proposed method is clearly preferable, as it obtains an average ranking of 1.2 opposed to the ranking of 3.2 of the second classified.

Classifier	CLL-SUB	GCM	GLA-BRA	Lymphoma	TOX	**Average**
FVQIT	1st	2nd	1st	1st	1st	**1.2**
Naive Bayes	2nd	3rd	3rd	4th	4th	3.2
k-NN	3rd	4th	4th	2nd	3rd	3.2
MLP	6th	1st	2nd	5th	5th	3.8
SVM	4th	6th	5th	3rd	2nd	4
C4.5	5th	5th	6th	6th	6th	5.6

Table 5.17: Ranking for each method on the comparative study of multiclass DNA microarray data sets

5.4 Summary

In this chapter a local classifier based on ITL is presented. The classifier is able to obtain complex classification models via a two-step process that first defines local models by means of a modified clustering algorithm and, subsequently, several one-layer neural networks, assigned to the local models, construct a piecewise borderline between classes.

Two versions of the method are detailed: binary (two-class problems) and multiclass. Using the divide-and-conquer approach, it has been shown that the proposed method is able to successfully classify complex and unbalanced data sets, high dimensional in data samples and/or features, achieving good average results. Several experiments have been performed over the complex domains of intrusion detection and microarray gene expression.

The intrusion detection data set employed is KDD Cup 99. It is very large (five million samples), highly unbalanced and has forty one features. The most important contribution of the method is the considerable reduction in the number of false positives (an important measure in this field of application), with a drastic reduction in the number of features used (6 vs 41) in comparison with the KDD Winner and the results obtained by other authors.

On the other hand, microarray data sets have a large amount of features (thousands or tens of thousands) but very few samples (tens or hundreds), which is a difficult challenge for most machine learning methods. In this case, the method has been compared with several state-of-the-art classifiers, achieving the best average values of all the performance measurements used, exhibiting an important difference with the second best method, both in the binary and the multiclass experiments. Furthermore, as different feature selection methods can select

different features, the stability of the proposed method has also been tested for different ranges of features, again showing the best behavior compared with the other classifiers.

CHAPTER 6

Conclusions

In this chapter, on Sect. 6.1, the general conclusions of this dissertation are presented. On Sect. 6.2, the publications in conferences and journals are presented.

6.1 Contributions

This dissertation discusses the application of information theory (IT) and information theoretic learning (ITL) to classification and feature selection. The new algorithms proposed are centered in two aspects of machine learning: feature selection and classification, with the common aim of confronting the diversity and heterogeneity of data sets. With that goal in mind, diversity in the cost of the features and heterogeneity in the samples are treated by the feature selection methods proposed. Specifically, two new algorithms for feature selection are developed. The first one takes into account the cost of each feature —besides its relevance—. The second algorithm makes use of the concept of ensemble, quite common for classification scenarios, but very little explored in the literature of feature selection. On the other hand, IT and ITL concepts can be employed as an alternative error function, thus allowing the exploration of another not very well studied field in the literature: the local modeling approach. Specifically, a new algorithm for classification is developed. This algorithm is based on the combination of neural networks by means of local modeling and techniques based on ITL, allowing for the treatment of complex and diverse data sets.

In light of the above, the conclusions obtained are the following:

- Not only features have different relevance/redundance with others and the output class, but they may also have a different importance regarding (economical, risk, computational, etc) cost. This last fact has not been explored in the scientific literature. In this thesis, a new cost-based feature selection method is proposed. The objective is solving

feature selection problems where reducing costs is important. The approach consists of adding a new term to the evaluation function of mRMR —an information theory based feature selection method— so that it is possible to reach a trade-off between the filter metric and the cost associated to the selected features. Results display that the approach is sound and allows the user to reduce the cost without compromising the classification error significantly, which can be useful in fields such as medical diagnosis or real-time applications.

- Diversity and heterogeneity in data sets prevents the users of FS of having a "best" method, and thus it can be hard to cope with all available ones to select the most adequate for each scenario. Trying to solve this problem, in this thesis an ensemble for feature selection is designed. Two ways of building ensembles are explored: (a) N selections using the same feature selection algorithm, using different training data and (b) N selections using a variety of different feature selection algorithms, all using the same training data. The particularity of the proposed ensemble is that it works with ordered rankings of features, which is a natural approach for feature selection methods. The individual rankings obtained for each of the packages were combined using ranking function learning, Ranking SVM in particular. Option (a) improves training times over the individual feature selection methods, while maintaining errors. Option (b) obtains the best average results regardless of the data set and thresholds chosen.

- Finally, the complexity and heterogeneity of data sets makes it difficult for a global machine learning approach to work properly. In this thesis, a new local classifier based on ITL is presented. The classifier is able to obtain complex classification models via a two-step process. This process first defines local models by means of a modified clustering algorithm and, second, trains several one-layer neural networks, assigned to the local models, in order to construct a piecewise borderline between classes. It has been shown that the proposed method is able to successfully classify complex and unbalanced data sets, high dimensional in data samples and/or features, achieving good average results. Several experiments have been performed over the complex domains of intrusion detection and microarray gene expression.

The following lines of research are proposed as future work:

- Extend the feature selection cost framework developed for mRMR to other feature selection methods.

- Experiment with other methods of ranking function learning for ensembles of feature

selection, in such a way that the ensemble gets more diversity and is able to handle better different types of data sets.

- Automatic estimation of parameters for FVQIT.
- Employ other algorithms than the one-layer neural network for the local models of FVQIT.

6.2 Publications

As a consequence of the research performed in this thesis, the following publications have been produced.

6.2.1 Journals

- Porto-Díaz, Iago and Bolón-Canedo, Verónica and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *A Study of Performance on Microarray Data Sets for a Classifier Based on Information Theoretic Learning*. Neural Networks (vol. 24, pp. 888–896, 2011)
- Porto-Díaz, Iago and Martínez-Rego, David and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *Information Theoretic Learning and Local Modeling for Binary and Multiclass Classification*. Progress in Artificial Intelligence (vol. 1, no. 4, pp. 315–328, 2012)
- Bolón-Canedo, Verónica and Porto-Díaz, Iago and Sánchez-Maroño, Noelia and Alonso-Betanzos, Amparo. *A Framework for Cost-Based Feature Selection*. Pattern Recognition (vol. 47, no. 7, pp. 2481–2489, 2014)

6.2.2 Conferences

- Martínez-Rego, David and Fontenla-Romero, Oscar and Alonso-Betanzos, Amparo and Porto-Díaz, Iago. *A New Supervised Local Modelling Classifier Based on Information Theory*. Proceedings of International Joint Conference on Neural Networks (IJCNN) 2009 (pp. 2014–2020, 2009)

- Porto-Díaz, Iago and Martínez-Rego, David and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *Combining Feature Selection and Local Modelling in the KDD Cup 99 Data set*. Proceedings of the International Conference on Artificial Neural Networks (ICANN) 2009 (pp. 824–833, 2009)

- Porto-Díaz, Iago and Bolón-Canedo, Verónica and Fontenla-Romero, Oscar and Alonso-Betanzos, Amparo. *Local Modeling Classifier for Microarray Gene-Expression Data*. Proceedings of the International Conference on Artificial Neural Networks (ICANN) 2010 (pp. 11-20, 2010)

- Porto-Díaz, Iago and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *A Multiclass Classifier Based on Local Modeling and Information Theoretic Learning*. Proceedings of the Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA) 2011.

- Seijo-Pardo, Borja and Bolón-Canedo, Verónica and Porto-Díaz, Iago and Alonso-Betanzos, Amparo. *Ensemble Feature Selection for Rankings of Features*. Advances in Computational Intelligence. Lecture Notes in Computer Science Vol. 9095. Proceedings of the 14th International Work Conference on Artificial Neural Networks (IWANN) (pp. 29–42, 2015)

APPENDIX

Summary in English

Machine learning is the area of artificial intelligence and computer science that studies algorithms that can learn from data, make predictions, and develop behaviors based on examples. The types of problems machine learning can solve are [15]: (a) classification, where the algorithm must assign unseen inputs to a series of classes; (b) regression, where the focus is predicting a continuous output; (c) clustering, where inputs must be classified into unknown groups, unlike classification; (d) density estimation, where the goal is finding the distribution of a set of inputs; and (e) dimensionality reduction, where inputs are simplified by mapping them to lower dimensional spaces. These tasks can also be classified, according to the nature of available learning data, in (a) supervised learning, where a set of known patterns are used for training; (b) unsupervised learning, where the objective is to unravel the underlying similarities between data; and (c) reinforcement learning, where the environment provides information about the goodness of the learning.

In supervised classification, the problem in which this thesis is focused, the mean squared error (MSE) is the measure that is typically utilized for evaluating the estimations made by the algorithms. However, the use of cost functions based on second-order moments (MSE) suffers from the limitation of the inherent Gaussian hypothesis. In this dissertation, this impediment is avoided by using a computationally-efficient model, based on information-theoretic descriptors of entropy, divergence and mutual information, combined with non-parametric PDF estimators. This brings robustness and generality to the cost function. This model is called Information Theoretic Learning (ITL) [116, 115]. As entropy is defined as the uncertainty of a random variable, it is natural to use it as a tool for applications where the data are incomplete or noisy.

The use of information theory (IT) and ITL in this thesis is twofold: (1) On the one hand, IT is used for the preprocessing step of a data mining pipeline. Specifically, two new algorithms for feature selection are developed. The first one takes into account the cost (computational, economic, etc.) of each feature —besides its relevance—. This fact is important due to the possibility of obtaining similar or better performances while reducing the associated cost. The

second algorithm makes use of the concept of ensemble, quite common for classification scenarios, but very little explored in the literature of feature selection. In this case, the aim is obtaining more stable results than using a single feature selection method and also improving the computational efficiency of the training process by means of distributed computing. (2) On the other hand, IT and ITL concepts can be employed as an alternative error function, thus allowing the exploration of another not very well studied field in the literature: the local modeling approach. Specifically, a new algorithm for classification is developed. This algorithm is based on the combination of neural networks by means of local modeling and techniques based on ITL.

I.1 Cost Feature Selection Based on Information Theory

The first part of this dissertation presents a new method for cost-based feature selection. Over the last few years, the dimensionality of data sets involved in data mining applications has increased dramatically. In this situation, feature selection becomes indispensable as it allows for dimensionality reduction and relevance detection. The method proposed in this part broadens the scope of feature selection by taking into consideration not only the relevance of the features but also their associated costs. Despite the previous attempts in classification and feature extraction, to the best knowledge of the author, there are only a few attempts to deal with this issue in feature selection. A new framework is proposed, which consists of adding a new term to the evaluation function of a filter method called Minimal Redundancy Maximal Relevance (mRMR), so that cost is taken into account. mRMR is one of the most employed multivariate ranker filters, due to obtaining good results in several fields. The evaluation function combines two constraints (as the name of the method indicates), maximal relevance and minimal redundancy.

In light of the above, the novelty of this approach lies in that the research in cost-based selection is extremely scarce in the literature. As a matter of fact, no cost methods can be found in the most popular machine learning and data mining tools. For instance, in Weka we can only find some methods that address the problem of cost associated to the instances (not to the features), and they were incorporated in the latest release. RapidMiner does in fact include some methods that take cost into account, but they are quite simple. One of them selects the attributes that have a cost value which satisfies a given condition and another one just selects the k attributes with the lower cost. Therefore, the cost-based feature selection method proposed in this thesis intends to cover this necessity. The behavior of the proposed method is tested on 17 heterogeneous classification data sets, employing a Support Vector Machine (SVM) as

a classifier. The results of the experimental study show that the approach is sound and that it allows the user to reduce the cost without compromising the classification error.

I.2 Ensemble Method for Feature Selection Based on Ranking Learning

The second part introduces a new ensemble for feature selection. In the last few years, ensemble learning has been the focus of much attention mainly in classification tasks, based on the assumption that combining the output of multiple experts is better than the output of any single expert. This idea of ensemble learning can be adapted for feature selection, in which different feature selection algorithms act as different experts. In this part, two problems are addressed: (1) the non-existence of a "best" method, which causes that the user has to search and choose a specific method for each problem; (2) the heterogeneity of data sets, which makes it difficult to obtain good results with one single method.

Machine learning methods have come to be a necessity for many companies, in order to obtain useful information and knowledge from their increasingly massive databases. Besides, real life data sets come in diverse flavors and sizes, and so their nature imposes several substantial restrictions for both learning models and feature selection algorithms. Data sets may be very large in samples and number of features and, also, there might be problems with redundant, noisy, multivariate and non-linear scenarios. Thus, most methods alone are not capable of confronting these problems, and something like "the best feature selection method" simply does not exist, making it difficult for users to select one method over another. In order to make a correct choice, a user not only needs to know the domain well and the characteristics of each data set, but also is expected to understand technical details of available algorithms. As experts of this type are not universally available, more user-friendly methods are necessary. In this sense, a possible way to confront this situation is to use an ensemble of feature selection algorithms, which is the idea proposed in this chapter. Specifically, methods that follow the ranking approach are used, i.e., they return an ordered ranking of all the features. Notice that methods that return a ranking of features are less computationally expensive than those which return a subset of selected features, and this is of vital importance when the current tendency is toward Big Data problems. Then, the outputs of all the components of the ensemble have to be combined in order to produce a common final output. The ensemble proposed in this chapter combines these rankings using Ranking SVM, which is a SVM-based method for learning of ranking functions.

Two ways of building ensembles are explored: (a) N selections using the same feature selection algorithm, using different training data and (b) N selections using a variety of different feature selection algorithms, all using the same training data. The adequacy of the ensemble is tested using SVM as a classifier. Both options are able to obtain good results. Option (a) improves training times over the individual feature selection methods, while maintaining errors. Option (b) obtains the best average results regardless of the data set and thresholds chosen.

I.3 Local Method for Classification Based on Information Theoretic Learning

The third part is dedicated to the development of a new local classification method, named *Frontier Vector Quantization based on IT* (FVQIT). The general aim, however, is the same: trying to confront diversity in data sets through the application of new ideas based on IT. The proposed algorithm performs classification based on the combination of neural networks by means of local modeling and techniques based on ITL. First, a modified ITL clustering algorithm is applied in order to identify the local models. Second, since the problem is simplified by splitting it into smaller parts, a simple but effective model, the one-layer neural network, is applied. This approach is related to the one followed in the previous chapter, which dealt with ensemble learning applied to feature selection.

More specifically, the training algorithm for the model works on two stages:

1. A set of nodes are placed on the frontiers between classes using a modified clustering algorithm based on ITL. Each of these nodes defines a local model. The algorithm minimizes the energy function that calculates the divergence between the Parzen estimator of the distribution of data points and the estimator of the distribution of the nodes. Under this premise, a physical interpretation can be made. Both data points and nodes are considered two kinds of particles with a potential field associated. These fields induce repulsive and attractive interactions between particles, depending on its sign. In the original VQIT algorithm, data and nodes had different signs. In FVQIT, data particles belonging to different classes have different signs. In this manner, a series of forces converge upon each node. Training patterns of a class exert an attractive force on a node and training patterns of the other class induce a repulsive force on it. Which class attracts and which class repels is decided using the Euclidean distance and k-NN (k-Nearest Neighbor) [28] as a rule of thumb. The closest class to the node (called 'own class') repels it

and the furthest one attracts it. These roles alternate during the iterations as nodes move. Moreover, there exists a third force of repulsion between the nodes, which favors a better distribution, avoiding the accumulation of several nodes at the same region.

2. Several one-layer neural networks, associated with these local models, are trained to locally classify the points in its proximity. Since each local model covers the closest points to the position of its associated node, the input space is completely filled, as input data are always assigned to a local model. In this second stage, the goal is to construct a classifier for each local model. This classifier will be in charge of classifying points in the region assigned to its local model and will be trained only with the points of the training set in this region. As local modeling algorithms may suffer from temporal efficiency problems, caused by the process of training several local classifiers, we have decided to use a lightweight classifier, the one-layer neural network. Its training algorithm allows rapid supervised training. The key idea is to measure the error prior to the nonlinear activation functions. In this manner, the minimization based on the MSE can be rewritten in equivalent fashion in terms of the error committed prior to the application of the activation function, which produces a system of equations with $I+1$ equations and unknowns, being I the dimension of the input. This kind of systems can be solved computationally with a complexity of $O(M^2)$, where $M = I+1$ is the number of weights of the network. Thus, it requires much less computational resources than classic methods.

The FVQIT method is successfully applied to problems with a large amount of instances and high dimension like intrusion detection and microarray gene expression. The intrusion detection data set employed is KDD Cup 99. It is very large (five million samples), highly unbalanced and has forty one features. The most important contribution of the method is the considerable reduction in the number of false positives (an important measure in this field of application), with a drastic reduction in the number of features used (6 vs 41), in comparison with results obtained by other authors.

Microarray gene expression is a technology that allows the examination of tens of thousands of genes at a time. For this reason, manual observation is not feasible and machine learning methods are suitable to face these types of data. Specifically, since the number of genes is very high, feature selection methods have proven valuable to deal with these unbalanced —-high dimensionality and low cardinality— data sets. The proposed classifier is employed to classify twelve DNA gene expression microarray data sets of different kinds of cancer. A comparative study with other well-known classifiers is performed. The proposed approach shows competitive results outperforming all other classifiers.

I.4 Structure

This thesis consists of the following chapters:

1. Chapter 1 presents the introduction, objectives, and structure of the thesis.
2. Chapter 2 introduces the domain of the research: information theory, information theoretic learning, and its applications in feature selection and classification.
3. Chapter 3 describes a new cost-based feature selection method.
4. Chapter 4 introduces a new ensemble method for feature selection, based in ranking learning.
5. Chapter 5 presents a new classification method based on the combination of neural networks by means of Information Theoretic Learning tools.
6. Chapter 6 summarizes the obtained contributions and conclusions and the produced publications.

I.5 Objectives

The objectives for each of the three main parts of this thesis are the following:

1. Cost-based feature selection.
    - Solve problems where not only it is interesting to minimize the classification error, but also to reduce costs that may be associated to input features.
    - Obtain a trade-off between a feature selection metric and the cost associated to the features, in order to select relevant features with a low associated cost, while keeping the classification accuracy.
2. Ensemble learning for feature selection.
    - Combine ordered rankings of features which are obtained from base selectors.
    - Achieve an improvement in the overall computational performance of the feature selection process, while maintaining the classification accuracy.

- Release the user from the task of deciding which feature selection method is the most appropriate, while maintaining the classification accuracy.

3. Local classification based on information theoretic learning.
   - Build complex classification models for two-class and multiclass problems. Those models are composed of several simpler neural network sub-models.
   - Achieve an improvement of classification performance on real problems.

I.6 Conclusions

The conclusions obtained are the following:

- Not only features have different relevance/redundance with others and the output class, but they may also have a different importance regarding (economical, risk, computational, etc) cost. This last fact has not been explored in the scientific literature. In this thesis, a new cost-based feature selection method is proposed. The objective is solving feature selection problems where reducing costs is important. The approach consists of adding a new term to the evaluation function of mRMR —an information theory based feature selection method— so that it is possible to reach a trade-off between the filter metric and the cost associated to the selected features. Results display that the approach is sound and allows the user to reduce the cost without compromising the classification error significantly, which can be useful in fields such as medical diagnosis or real-time applications.
- Diversity and heterogeneity in data sets prevents the users of FS of having a “best” method, and thus it can be hard to cope with all available ones to select the most adequate for each scenario. Trying to solve this problem, in this thesis an ensemble for feature selection is designed. Two ways of building ensembles are explored: (a) N selections using the same feature selection algorithm, using different training data and (b) N selections using a variety of different feature selection algorithms, all using the same training data. The particularity of the proposed ensemble is that it works with ordered rankings of features, which is a natural approach for feature selection methods. The individual rankings obtained for each of the packages were combined using ranking function learning, Ranking SVM in particular. Option (a) improves training times over the individual feature selection methods, while maintaining errors. Option (b) obtains the best average results regardless of the data set and thresholds chosen.

- Finally, the complexity and heterogeneity of data sets makes it difficult for a global classification approach to work properly. In this thesis, a new local classifier based on ITL is presented. The classifier is able to obtain complex classification models via a two-step process. This process first defines local models by means of a modified clustering algorithm and, second, trains several one-layer neural networks, assigned to the local models, in order to construct a piecewise borderline between classes. It has been shown that the proposed method is able to successfully classify complex and unbalanced data sets, high dimensional in data samples and/or features, achieving good average results. Several experiments have been performed over the complex domains of intrusion detection and microarray gene expression.

I.7 Future work

The following lines of research are proposed as future work:

- Extend the feature selection cost framework developed for mRMR to other feature selection methods.

- Experiment with other methods of ranking function learning for ensembles of feature selection, in such a way that the ensemble gets more diversity and is able to handle better different types of data sets.

- Automatic estimation of parameters for FVQIT.

- Employ other algorithms than the one-layer neural network for the local models of FVQIT.

I.8 Publications

As a consequence of the research performed in this thesis, the following publications have been produced.

I.8.1 Journals

- Porto-Díaz, Iago and Bolón-Canedo, Verónica and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *A Study of Performance on Microarray Data Sets for a Classifier Based on Information Theoretic Learning*. Neural Networks (vol. 24, pp. 888–896, 2011)

- Porto-Díaz, Iago and Martínez-Rego, David and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *Information Theoretic Learning and Local Modeling for Binary and Multiclass Classification*. Progress in Artificial Intelligence (vol. 1, no. 4, pp. 315–328, 2012)

- Bolón-Canedo, Verónica and Porto-Díaz, Iago and Sánchez-Maroño, Noelia and Alonso-Betanzos, Amparo. *A Framework for Cost-Based Feature Selection*. Pattern Recognition (vol. 47, no. 7, pp. 2481–2489, 2014)

I.8.2 Conferences

- Martínez-Rego, David and Fontenla-Romero, Oscar and Alonso-Betanzos, Amparo and Porto-Díaz, Iago. *A New Supervised Local Modelling Classifier Based on Information Theory*. Proceedings of International Joint Conference on Neural Networks (IJCNN) 2009 (pp. 2014–2020, 2009)

- Porto-Díaz, Iago and Martínez-Rego, David and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *Combining Feature Selection and Local Modelling in the KDD Cup 99 Data set*. Proceedings of the International Conference on Artificial Neural Networks (ICANN) 2009 (pp. 824–833, 2009)

- Porto-Díaz, Iago and Bolón-Canedo, Verónica and Fontenla-Romero, Oscar and Alonso-Betanzos, Amparo. *Local Modeling Classifier for Microarray Gene-Expression Data*. Proceedings of the International Conference on Artificial Neural Networks (ICANN) 2010 (pp. 11-20, 2010)

- Porto-Díaz, Iago and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *A Multiclass Classifier Based on Local Modeling and Information Theoretic Learning*. Proceedings of the Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA) 2011.

- Seijo-Pardo, Borja and Bolón-Canedo, Verónica and Porto-Díaz, Iago and Alonso-Betanzos, Amparo. *Ensemble Feature Selection for Rankings of Features*. Advances in

Computational Intelligence. Lecture Notes in Computer Science Vol. 9095. Proceedings of the 14th International Work Conference on Artificial Neural Networks (IWANN) (pp. 29–42, 2015)

APPENDIX II

Resumen en castellano

El aprendizaje automático es el área de la inteligencia artificial y de la computación que estudia algoritmos que pueden aprender a partir de datos, hacer predicciones y desarrollar comportamientos basados en ejemplos. Los tipos de problemas que el aprendizaje automático puede resolver son [15]: (a) clasificación, donde el algoritmo debe asignar nuevas entradas a una serie de clases; (b) regresión, donde el objetivo es predecir una salida continua; (c) agrupamiento (*clustering*), donde las entradas deben ser clasificadas en grupos desconocidos, al contrario que clasificación; (d) estimación de densidad, donde el objetivo es encontrar la distribución de un conjunto de entradas y (e) reducción de la dimensión, donde las entradas se simplifican mediante el mapeo a espacios de menor dimensión. Estas tareas pueden también ser clasificadas, de acuerdo a la naturaleza de los datos de aprendizaje disponibles, en (a) aprendizaje supervisado, donde un conjunto de patrones conocidos se utiliza para el entrenamiento; (b) aprendizaje no supervisado, donde el objetivo es desentrañar las similitudes subyacentes entre datos y (c) aprendizaje por refuerzo, donde es el entorno el que proporciona información sobre la efectividad del aprendizaje.

En la clasificación supervisada, el problema en el que se centra esta tesis, el error cuadrático medio (ECM) es la medida que se utiliza típicamente para evaluar los estimadores construidos por los algoritmos. Sin embargo, el uso de funciones de coste basadas en momentos de segundo orden (ECM) sufre de la limitación de la hipótesis gaussiana inherente. En este trabajo, este impedimento se evita usando un modelo computacionalmente eficiente, basado en descriptores de la entropía, divergencia e información mutua de teoría de la información, combinados con estimadores no paramétricos de la función de densidad de probabilidad. Esto aporta robustez y generalidad a la función de coste. Este modelo se denomina *Information Theoretic Learning* (ITL) [116, 115]. Como la entropía se define como la incertidumbre de una variable aleatoria, es natural utilizarla como una herramienta para aplicaciones donde los datos son incompletos o presentan ruido.

El uso de teoría de la información (IT) e ITL en esta tesis se desglosa en dos partes: (1) en primer lugar, IT se utiliza para la fase de preprocesado. Específicamente, se desarrollan

dos nuevos algoritmos para selección de características. El primero tiene en cuenta el coste (computacional, económico, etc.) de cada característica (además de su relevancia). Este detalle es importante debido a la posibilidad de obtener rendimientos similares o mejores mientras se reduce el coste asociado. El segundo algoritmo hace uso del concepto de *ensemble*, bastante común en escenarios de clasificación, pero muy poco explorado en la literatura de selección de características. En este caso, el objetivo es obtener resultados más estables que los que se obtienen utilizando un método único de selección de características y también mejorar la eficiencia computacional del proceso de entrenamiento por medio de computación distribuida. (2) Por otra parte, los conceptos de IT e ITL se pueden utilizar como una función de error alternativa, permitiendo la exploración de otro campo no muy estudiado en la literatura: la aproximación basada en modelos locales. Específicamente, se desarrolla un nuevo algoritmo para clasificación, el cual está basado en la combinación de redes de neuronas por medio de modelado local y técnicas basadas en ITL.

II.1 Selección de características con coste basada en teoría de la información

La primera parte de esta tesis presenta un nuevo método para selección de características con coste. A lo largo de los últimos años, la dimensión de los conjuntos de datos que se utilizan en minería de datos ha aumentado dramáticamente. En esta situación, la selección de características se convierte en indispensable, ya que permite reducir la dimensión detectando relevancia. El método propuesto en esta parte amplía el ámbito de la selección de características teniendo en consideración no solo la relevancia de las características, sino también sus costes asociados. A pesar de que existen intentos previos en clasificación y extracción de características, existen pocos intentos para tratar con este problema en selección de características. Se propone un nuevo *framework*, que consiste en añadir un nuevo término a la función de evaluación de un método filtro de selección de características llamado *Minimal Redundancy Maximal Relevance* (mRMR), de tal manera que el coste se tenga en cuenta. mRMR es uno de los filtros multivariados más utilizados, debido a la obtención de buenos resultados en varios campos. La función de evaluación combina dos restricciones (como el propio nombre del método indica), relevancia máxima y mínima redundancia.

A la luz de lo anterior, la novedad de esta aproximación radica en que la investigación en selección de características con coste es extremadamente escasa en la literatura. De hecho, las herramientas de aprendizaje automático y minería de datos más habituales no incluyen ningún

método para tratar con coste. Por ejemplo, en *Weka* solo se pueden encontrar algunos métodos que abordan el problema del coste asociado a las muestras (no a las características), y fueron añadidos en la última versión. *RapidMiner* de hecho incluye algunos métodos que tienen el coste en cuenta, pero son bastante simples. Uno de ellos tan solo selecciona los *k* atributos con el coste más bajo. Por lo tanto, el método de selección de características con coste propuesto en esta tesis pretende cubrir esta necesidad. El comportamiento del método propuesto se prueba en 17 conjuntos heterogéneos de clasificación, empleando una máquina de vectores soporte (SVM) como clasificador. Los resultados del estudio experimental realizado muestran que la aproximación es sólida y que permite al usuario reducir el coste sin comprometer el error de clasificación.

II.2 Método *ensemble* para selección de características basado en aprendizaje de rankings

La segunda parte presenta un nuevo *ensemble* para selección de características. En los últimos años, el aprendizaje basado en *ensembles* ha sido el foco de mucha atención, principalmente en tareas de clasificación, centrándose en el supuesto de que combinar la salida de varios expertos es mejor que la salida de un único experto. Esta idea del aprendizaje *ensemble* se puede adaptar para selección de características, en la que diferentes algoritmos de selección actúan como diferentes expertos. En esta parte, se abordan dos problemas: (1) la no existencia de un método "mejor", lo que provoca que el usuario tenga que buscar y elegir un método específico para cada problema; (2) la heterogeneidad de los conjuntos de datos, que hace que sea difícil obtener buenos resultados con un único método.

Los métodos de aprendizaje automático se han convertido en una necesidad para muchas empresas para obtener información y conocimiento útil a partir de sus masivas bases de datos. Además, los conjuntos de datos de la vida real se presentan en muchas formas y tamaños, por lo que su naturaleza impone varias restricciones substanciales tanto para modelos de aprendizaje como para algoritmos de selección de características. Los conjuntos de datos pueden ser muy grandes y de alta dimensión y también puede haber problemas con escenarios redundantes, ruidosos, multivariados y no lineales. Así, la mayoría de los métodos por sí solos no son capaces de enfrentarse a estos problemas, y algo como el "mejor método de selección de características" simplemente no existe, haciendo difícil para los usuarios la elección de un método sobre otros. Con idea de hacer una elección correcta, un usuario no solo necesita conocer bien el dominio y características de cada conjunto de datos, sino que también debe entender detalles

técnicos de los algoritmos disponibles. Ya que los expertos de este tipo no están universalmente disponibles, son necesarios más métodos amigables con el usuario. En este sentido, un posible modo de enfrentarse a esta situación es utilizar un *ensemble* de algoritmos de selección de características, y esa es la idea propuesta en este capítulo. Específicamente, se utilizan métodos que siguen la aproximación ranking, es decir, que devuelven una lista ordenada de todas las características. Nótese que los métodos que se comportan de esta manera son más baratos computacionalmente que aquellos que devuelven un subconjunto de características seleccionadas, y esto es de vital importancia cuando la tendencia actual va hacia grandes conjuntos de *Big Data*. Entonces, las salidas de todos los componentes del *ensemble* tienen que combinarse para producir una salida final común. El *ensemble* propuesto en esta parte de la tesis combina estos rankings utilizando *Ranking SVM*, que es un método basado en SVM para el aprendizaje de funciones ranking.

Se exploran dos formas de construir *ensembles*: (a) *N* selecciones utilizando el mismo algoritmo de selección de características, con diferentes datos y (b) *N* selecciones utilizando una variedad de métodos de selección de características, con los mismos datos de entrenamiento. La idoneidad de esta aproximación se prueba utilizando una SVM como clasificador. Ambas opciones obtienen buenos resultados. La opción (a) mejora los tiempos de entrenamiento sobre los obtenidos por los métodos de selección individuales, manteniendo los errores. La opción (b) obtiene los mejores resultados medios independientemente del conjunto de datos y umbrales elegidos.

II.3 Método local de clasificación basado en ITL

La tercera parte se dedica al desarrollo de un nuevo método de clasificación local, denominado *Frontier Vector Quantization based on IT* (FVQIT). El objetivo general, sin embargo, es el mismo: intentar enfrentarse a la diversidad en los conjuntos de datos a través de la aplicación de nuevas ideas basadas en TI. El algoritmo propuesto lleva a cabo tareas de clasificación mediante de la combinación de redes de neuronas utilizando técnicas de modelado local y basadas en ITL. En primer lugar, se aplica un algoritmo de agrupamiento (*clustering*) modificado para identificar los modelos locales. En segundo lugar, dado que el problema se simplifica al dividirlo en partes más pequeñas, se aplica un modelo simple pero efectivo, la red de neuronas de una sola capa. Esta aproximación se relaciona con la seguida en la parte anterior, que trataba con aprendizaje *ensemble* aplicado a selección de características.

Más en detalle, el algoritmo de entrenamiento para el modelo trabaja en dos fases:

1. Se sitúa un conjunto de nodos en las fronteras entre clases utilizando un algoritmo de agrupamiento basado en ITL modificado. Cada uno de estos nodos define un modelo local. El algoritmo minimiza la función de energía que calcula la divergencia entre el estimador de Parzen de la distribución de los datos y el estimador de la distribución de los nodos. Bajo esta premisa, se puede hacer una interpretación física. Tanto los datos como los nodos se consideran dos tipos de partículas con un campo potencial asociado. Estos campos inducen interacciones repulsivas y atractivas entre partículas, en función de su signo. En FVQIT, los datos que pertenecen a distintas clases tienen diferente signo. De este modo, una serie de fuerzas convergen sobre cada nodo. Los patrones de entrenamiento de una clase ejercen una fuerza atractiva sobre un nodo, mientras que los patrones de entrenamiento de la otra clase inducen una fuerza repulsiva sobre él. Qué clase atrae y qué clase repele se decide utilizando la distancia euclídea y el algoritmo *k-Nearest Neighbor* (k-NN) [28]. La clase más cercana al nodo (llamada la "clase propia") lo repele y la clase más lejana lo atrae. Estos roles se alternan durante las iteraciones, mientras los nodos se mueven. Además, existe una tercera fuerza de repulsión entre los nodos, la cual favorece una mejor distribución, evitando la acumulación de varios nodos en una misma región.

2. Se entrenan varias redes de neuronas de una sola capa, asociadas con estos modelos locales, para clasificar localmente los datos en su proximidad. Dado que cada modelo local cubre los puntos más cercanos a la posición de su nodo asociado, el espacio de entrada está completamente cubierto, ya que los datos de entrada siempre se asignan a un modelo local. En esta segunda fase, el objetivo es construir un clasificador para cada modelo local. Este clasificador se encarga de clasificar datos en la región asignada a su modelo local y se entrena con solo los datos del conjunto de entrenamiento en esta región. Como los algoritmos de modelado local pueden tener problemas de eficiencia temporal, causados por el proceso de entrenar varios clasificadores locales, se ha decidido utilizar un clasificador ligero, las redes de una sola capa. Su algoritmo de entrenamiento permite un rápido entrenamiento supervisado. La idea clave es medir el error a priori de las funciones de activación no lineales. De esta manera, la minimización basada en el ECM puede ser reescrita de forma equivalente en términos del error cometido a priori, lo que produce un sistema de ecuaciones con $I+1$ ecuaciones e incógnitas, siendo I la dimensión de la entrada. Este tipo de sistemas se pueden resolver computacionalmente con una complejidad de $O(M^2)$, donde $M = I+1$ es el número de pesos de la red. Así, se requieren muchos menos recursos computacionales que los que requieren otros métodos clásicos.

El método FVQIT se aplica con éxito a problemas con una gran cantidad de muestras y alta dimension como detección de intrusos y expresión génica *microarray* (*microarray gene expression*). El conjunto de datos de detección de intrusos es el KDD Cup 99. Es muy grande (cinco millones de muestras), muy desbalanceado y tiene 41 características. La contribución más importante del método propuesto es la reducción considerable del número de falsos positivos (una medida importante en este campo de aplicación), con una reducción drástica del número de características utilizadas (seis contra 41), en comparación con resultados obtenidos por otros autores.

La expresión génica *microarray* (*microarray gene expression*) es una tecnología que permite examinar decenas de miles de genes al mismo tiempo. Por esta razón, la observación manual no es factible y los métodos de aprendizaje automático son adecuados para enfrentarse a este tipo de datos. Específicamente, ya que el número de genes es muy alto, los métodos de selección de características han demostrado ser valiosos para tratar con estos conjuntos de datos tan desbalanceados (alta dimensión y poca cardinalidad). El clasificador propuesto se utiliza para clasificar doce conjuntos de datos *microarray* de diferentes tipos de cáncer. Se lleva a cabo un estudio comparativo con otros clasificadores comunes. La aproximación propuesta muestra resultados competitivos, consiguiendo mejores resultados que todos los demás clasificadores.

II.4 Estructura

Esta tesis consta de los siguientes capítulos:

1. El capítulo 1 presenta la introducción, objetivos y estructura de la tesis.
2. El capítulo 2 presenta el dominio de la investigación: teoría de la información, *information theoretic learning* y sus aplicaciones en selección de características y clasificación.
3. El capítulo 3 describe un nuevo método de selección de características con coste.
4. El capítulo 4 presenta un nuevo método *ensemble* para selección de características, basado en aprendizaje *ranking*.
5. El capítulo 5 presenta un nuevo método de clasificación basado en la combinación de redes de neuronas por medio de herramientas de *information theoretic learning*.
6. El capítulo 6 resume las contribuciones y conclusiones obtenidas y las publicaciones producidas.

II.5 Objetivos

Los objetivos para cada una de las tres partes principales de esta tesis son los siguientes:

1. Selección de características basada en coste.
    - Resolver problemas donde no solo es interesante minimizar el error de clasificación, sino también reducir costes que pueden estar asociados a las características de entrada.
    - Obtener una compensación entre una métrica de selección de características y el coste asociado a las características, para seleccionar características relevantes con un coste bajo asociado, mientras se mantiene la precisión de la clasificación.
2. Aprendizaje *ensemble* para selección de características.
    - Combinar rankings ordenados de características que se obtienen a partir de selectores base.
    - Obtener una mejora en el rendimiento computacional del proceso de selección de características, manteniendo la precisión en la clasificación.
    - Liberar al usuario de la tarea de decidir qué método de selección de características es el más apropiado, mientras se mantiene la precisión en la clasificación.
3. Clasificación local basada en *information theoretic learning*.
    - Construir modelos de clasificación complejos para problemas de dos clases y multiclase. Estos modelos se componen de varios submodelos de redes neuronales más simples.
    - Lograr una mejora en el rendimiento en clasificación en problemas reales.

II.6 Conclusiones

Las conclusiones obtenidas son las siguientes:

- No solo las características tienen diferente relevancia/redundancia con otras y con la clase de salida, sino también pueden tener una diferente importancia en función de su

coste (económico, computacional, riesgo, etc.). Este último hecho no ha sido explorado en la literatura científica. En esta tesis, se propone un nuevo método de selección de características basado en coste. El objetivo es resolver problemas de selección de características donde reducir costes es importante. La aproximación consiste en añadir un nuevo término a la función de evaluación de mRMR (un método de selección de características basado en teoría de la información), de tal modo que es posible alcanzar una compensación entre la métrica del método y el coste asociado con las características seleccionadas. Los resultados obtenidos muestran que la aproximación es sólida y permite al usuario reducir el coste sin comprometer significativamente el error de clasificación, lo cual puede ser útil en campos como el diagnóstico médico o las aplicaciones en tiempo real.

- La diversidad y la heterogeneidad de los conjuntos datos impide que los usuarios de selección de características dispongan de un "mejor" método. En consecuencia, puede ser difícil enfrentarse con todos los disponibles para seleccionar el más adecuado para cada escenario. Con la intención de resolver este problema, en esta tesis se diseña un *ensemble* para selección de características. Se exploran dos maneras de construir *ensembles*: (a) *N* selecciones utilizando el mismo algoritmo de selección de características con diferentes datos de entrenamiento y (b) *N* selecciones utilizando una variedad de algoritmos de selección de características, todos ellos con los mismos datos de entrenamiento. La particularidad del *ensemble* propuesto es que trabaja con rankings ordenados de características, lo cual es una aproximación natural para los métodos de selección. Los rankings individuales obtenidos para cada uno de los paquetes se combinaron utilizando aprendizaje de funciones ranking, en particular *Ranking SVM*. La opción (a) mejora los tiempos de entrenamiento sobre los métodos de selección individuales, manteniendo los errores. La opción (b) obtiene los mejores resultados independientemente del conjunto de datos y los umbrales elegidos.

- Finalmente, la complejidad y heterogeneidad de los conjuntos de datos dificulta que un clasificador automático global funcione correctamente. En esta tesis se presenta un nuevo clasificador local basado en *information theoretic learning*. El clasificador es capaz de obtener modelos de clasificación complejos mediante un proceso de dos etapas. Este proceso define, en primer lugar, modelos locales por medio de un algoritmo de agrupamiento modificado y, en segundo lugar, entrena varias redes de neuronas de una sola capa, asignadas a los modelos locales, para construir una frontera a trozos entre clases. Se ha demostrado que el método propuesto es capaz de clasificar con éxito conjuntos de datos complejos y desbalanceados, con alta dimensión y gran cardinalidad, obteniendo buenos resultados medios. Se han llevado a cabo varios experimentos sobre los complejos dominios de detección de intrusos y expresión génica *microarray*.

II.7 Trabajo futuro

Se proponen las siguientes líneas de investigación como trabajo futuro:

- Extender el *framework* de selección de características con coste desarrollado para mRMR a otros métodos de selección de características.
- Experimentar con otros métodos de aprendizaje de funciones *ranking* para *ensembles* de selección de características, de tal modo que el *ensemble* obtenga más diversidad y sea capaz de manejar mejor diferentes tipos de conjuntos de datos.
- Estimación automática de parámetros para el FVQIT.
- Emplear otros algoritmos distintos de la red de neuronas de una sola capa para los modelos locales del FVQIT.

II.8 Publicaciones

Como consecuencia de la investigación llevada a cabo en esta tesis, se han producido las siguientes publicaciones.

II.8.1 Revistas

- Porto-Díaz, Iago and Bolón-Canedo, Verónica and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *A Study of Performance on Microarray Data Sets for a Classifier Based on Information Theoretic Learning*. Neural Networks (vol. 24, pp. 888–896, 2011)
- Porto-Díaz, Iago and Martínez-Rego, David and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *Information Theoretic Learning and Local Modeling for Binary and Multiclass Classification*. Progress in Artificial Intelligence (vol. 1, no. 4, pp. 315–328, 2012)
- Bolón-Canedo, Verónica and Porto-Díaz, Iago and Sánchez-Maroño, Noelia and Alonso-Betanzos, Amparo. *A Framework for Cost-Based Feature Selection*. Pattern Recognition (vol. 47, no. 7, pp. 2481–2489, 2014)

II.8.2 Congresos

- Martínez-Rego, David and Fontenla-Romero, Oscar and Alonso-Betanzos, Amparo and Porto-Díaz, Iago. *A New Supervised Local Modelling Classifier Based on Information Theory*. Proceedings of International Joint Conference on Neural Networks (IJCNN) 2009 (pp. 2014–2020, 2009)

- Porto-Díaz, Iago and Martínez-Rego, David and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *Combining Feature Selection and Local Modelling in the KDD Cup 99 Data set*. Proceedings of the International Conference on Artificial Neural Networks (ICANN) 2009 (pp. 824–833, 2009)

- Porto-Díaz, Iago and Bolón-Canedo, Verónica and Fontenla-Romero, Oscar and Alonso-Betanzos, Amparo. *Local Modeling Classifier for Microarray Gene-Expression Data*. Proceedings of the International Conference on Artificial Neural Networks (ICANN) 2010 (pp. 11-20, 2010)

- Porto-Díaz, Iago and Alonso-Betanzos, Amparo and Fontenla-Romero, Oscar. *A Multiclass Classifier Based on Local Modeling and Information Theoretic Learning*. Proceedings of the Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA) 2011.

- Seijo-Pardo, Borja and Bolón-Canedo, Verónica and Porto-Díaz, Iago and Alonso-Betanzos, Amparo. *Ensemble Feature Selection for Rankings of Features*. Advances in Computational Intelligence. Lecture Notes in Computer Science Vol. 9095. Proceedings of the 14th International Work Conference on Artificial Neural Networks (IWANN) (pp. 29–42, 2015)

Bibliography

[1] ABEEL T., HELLEPUTTE T., VAN DE PEER Y., DUPONT P., AND SAEYS Y. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics* **26**(3), 392–398 (2010).

[2] AKAIKE H. Information theory and an extension of the maximum likelihood principle. In "Selected Papers of Hirotugu Akaike", pages 199–213. Springer (1998).

[3] AKAY M. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications* **36**(2), 3240–3247 (2009).

[4] ALIZADEH A., EISEN M., DAVIS R., MA C., LOSSOS I., ROSENWALD A., BOLDRICK J., SABET H., TRAN T., YU X., ET AL.. Distinct Types of Diffuse Large B-cell Lymphoma Identified by Gene Expression Profiling. *Nature* **403**(6769), 503–511 (2000).

[5] ALON U., BARKAI N., NOTTERMAN D., GISH K., YBARRA S., MACK D., AND LEVINE A. Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. *Proceedings of the National Academy of Sciences of the United States of America* **96**(12), 6745–6750 (1999).

[6] ALONSO-BETANZOS A., SANCHEZ-MARONO N., CARBALLAL-FORTES F., SUAREZ-ROMERO J., AND PEREZ-SANCHEZ B. Classification of computer intrusions using fuctional networks. a comparative study. In "Proc ESANN", pages 25–27 (2007).

[7] ARONSZAJN N. Theory of reproducing kernels. *Transactions of the American mathematical society* pages 337–404 (1950).

[8] ASUNCION A. AND NEWMAN D. J. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/. Last access: November 2013 (2010). University of California, Irvine, School of Information and Computer Sciences.

[9] ASUNCION A. AND NEWMAN D. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. `http://mlearn.ics.uci.edu/MLRepository.html`. Last access: November 2013.

[10] BATTITI R. Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on* **5**(4), 537–550 (1994).

[11] BAUM E. B. AND LANG K. J. Constructing hidden units using examples and queries. In "Advances in neural information processing systems", pages 904–910 (1991).

[12] BAY S. Combining nearest neighbor classifiers through multiple feature subsets. In "ICML", volume 98, pages 37–45. Citeseer (1998).

[13] BEN BRAHIM A. AND LIMAM M. Robust ensemble feature selection for high dimensional data sets. In "High Performance Computing and Simulation (HPCS), 2013 International Conference on", pages 151–157. IEEE (2013).

[14] BHAT H. S. AND KUMAR N. On the derivation of the bayesian information criterion. *School of Natural Sciences, University of California* (2010).

[15] BISHOP C. M. "Pattern recognition and machine learning". springer (2006).

[16] BISHOP C. "Neural networks for pattern recognition". Clarendon press Oxford (1995).

[17] BOLÓN-CANEDO V., PORTO-DÍAZ I., SÁNCHEZ-MAROÑO N., AND ALONSO-BETANZOS A. A framework for cost-based feature selection. *Pattern Recognition* **47**(7), 2481–2489 (2014).

[18] BOLÓN-CANEDO V., SÁNCHEZ-MARONO N., AND ALONSO-BETANZOS A. On the Effectiveness of Discretization on Gene Selection of Microarray Data. In "Proceedings of International Joint Conference on Neural Networks, IJCNN", pages 3167–3174 (2010).

[19] BOLÓN-CANEDO V., SÁNCHEZ-MAROÑO N., AND ALONSO-BETANZOS A. Feature selection and classification in multiple class datasets: An application to kdd cup 99 dataset. *Expert Systems with Applications* **38**(5), 5947–5957 (2011).

[20] BOLÓN-CANEDO V., SÁNCHEZ-MAROÑO N., AND ALONSO-BETANZOS A. An ensemble of filters and classifiers for microarray data classification. *Pattern recognition* **45**(1), 531–539 (2012).

[21] BOLÓN-CANEDO V., SÁNCHEZ-MARONO N., AND ALONSO-BETANZOS A. A review of feature selection methods on synthetic data. *Knowledge and Information Systems (in press)* (2012).

[22] BOLÓN-CANEDO V., SANCHEZ-MAROO N., AND ALONSO-BETANZOS A. A combination of discretization and filter methods for improving classification performance

in kdd cup 99 dataset. In “Neural Networks, 2009. IJCNN 2009. International Joint Conference on”, pages 359–366. IEEE (2009).

[23] BRAMER M. “Principles of data mining”. Springer (2013).

[24] BREIMAN L. Bagging predictors. *Machine learning* **24**(2), 123–140 (1996).

[25] BROWN G., POCOCK A., ZHAO M., AND LUJÁN M. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research* **13**(1), 27–66 (2012).

[26] CAI Y., HUANG T., HU L., SHI X., XIE L., AND LI Y. Prediction of lysine ubiquitination with mrmr feature selection and analysis. *Amino acids* **42**(4), 1387–1395 (2012).

[27] CASTILLO E., FONTENLA-ROMERO O., GUIJARRO-BERDIÑAS B., AND ALONSO-BETANZOS A. A Global Optimum Approach for One-Layer Neural Networks. *Neural Computation* **14**(6), 1429–1449 (2002).

[28] COVER T. AND HART P. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on* **13**(1), 21–27 (1967).

[29] COVER T. M. AND THOMAS J. A. “Elements of Information Theory”. Wiley (1991).

[30] CRISTIANINI N. AND SHAWE-TAYLOR J. “An introduction to support Vector Machines: and other kernel-based learning methods”. Cambridge Univ Pr (2000).

[31] DASARATHY B. AND SHEELA B. A composite classifier system design: concepts and methodology. *Proceedings of the IEEE* **67**(5), 708–713 (1979).

[32] DASH M. AND LIU H. Consistency-based search in feature selection. *Artificial intelligence* **151**(1), 155–176 (2003).

[33] DIMITRAKAKIS C. AND BENGIO S. Online adaptive policies for ensemble classifiers. *Neurocomputing* **64**, 211–221 (2005).

[34] DING C. AND PENG H. Minimum redundancy feature selection from microarray gene expression data. In “Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE”, pages 523–528. IEEE (2003).

[35] DING C. AND PENG H. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. *Journal of Bioinformatics and Computational Biology* **3**(2), 185–206 (2005).

[36] DUCH W. Feature extraction: foundations and applications. *Studies in Fuzziness and Soft Computing. Springer: Berlin Heidelberg New York* pages 89–117 (2006).

[37] DUDA R., HART P., AND STORK D. "Pattern Classification". John Wiley & Sons, New York, 2 edition (2001).

[38] EL AKADI A., EL OUARDIGHI A., AND ABOUTAJDINE D. A powerful feature selection approach based on mutual information. *International Journal of Computer Science and Network Security* **8**(4), 116 (2008).

[39] ELKAN C. Results of the kdd'99 classifier learning. *ACM SIGKDD Explorations Newsletter* **1**(2), 63–64 (2000).

[40] FAHLMAN S. E. AND LEBIERE C. The cascade-correlation learning architecture. *Advances in Neural Information Processing Systems II* (1989).

[41] FAYYAD U. AND IRANI K. Multi-interval discretization of continuous-valued attributes for classification learning. In "Proceedings of the 13th International Joint Conference on Artificial Intelligence", pages 1022–1029. Morgan Kaufmann (1993).

[42] FEDDEMA J., LEE C., AND MITCHELL O. Weighted selection of image features for resolved rate visual feedback control. *Robotics and Automation, IEEE Transactions on* **7**(1), 31–47 (1991).

[43] FERGUSON T. S. A bayesian analysis of some nonparametric problems. *The annals of statistics* pages 209–230 (1973).

[44] FISHER R. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics* **7**(2), 179–188 (1936).

[45] FLAKE G. W. Square unit augmented radially extended multilayer perceptrons. In "Neural Networks: Tricks of the Trade", pages 145–163. Springer (1998).

[46] FLEURET F. Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research* **5**, 1531–1555 (2004).

[47] FORMAN G. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research* **3**, 1289–1305 (2003).

[48] FREUND Y. AND SCHAPIRE R. Experiments with a new boosting algorithm. In "Machine Learning - International Workshop then Conference -", pages 148–156. Morgan Kaufmann Publishers, Inc. (1996).

[49] FREUND Y. AND SCHAPIRE R. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences* **55**(1), 119–139 (1997).

[50] FRIEDMAN J. H. Regularized discriminant analysis. *Journal of the American statistical association* **84**(405), 165–175 (1989).

[51] FUGATE M. AND GATTIKER J. Computer intrusion detection with classification and anomaly detection, using svms. *International Journal of Pattern Recognition and Artificial Intelligence* **17**(3), 441–458 (2003).

[52] FUNG G. AND MANGASARIAN O. L. Data selection for support vector machines classifiers. In "Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining", pages 64–70. ACM (2000).

[53] FUNG G. AND MANGASARIAN O. L. Proximal support vector machine classifiers. In "Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining", pages 77–86. ACM (2001).

[54] FÜRNKRANZ J. AND HÜLLERMEIER E. "Preference learning". Springer (2010).

[55] GOLUB T., SLONIM D., TAMAYO P., HUARD C., GAASENBEEK M., MESIROV J., COLLER H., LOH M., DOWNING J., CALIGIURI M., ET AL.. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *science* **286**(5439), 531–537 (1999).

[56] GORDON G., JENSEN R., HSIAO L., GULLANS S., BLUMENSTOCK J., RAMASWAMY S., RICHARDS W., SUGARBAKER D., AND BUENO R. Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer research* **62**(17), 4963–4971 (2002).

[57] GUYON I. AND ELISSEEFF A. An introduction to variable and feature selection. *The Journal of Machine Learning Research* **3**, 1157–1182 (2003).

[58] GUYON I., GUNN S., NIKRAVESH M., AND ZADEH L. "Feature Extraction. Foundations and Applications". Springer (2006).

[59] GUYON I., WESTON J., BARNHILL S., AND VAPNIK V. Gene selection for cancer classification using support vector machines. *Machine learning* **46**(1-3), 389–422 (2002).

[60] HALL M., FRANK E., HOLMES G., PFAHRINGER B., REUTEMANN P., AND WITTEN I. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter* **11**(1), 10–18 (2009).

[61] HALL M. "Correlation-based Feature Selection for Machine Learning". PhD thesis, University of Waikato, Hamilton, New Zealand (1999).

[62] HALL M. AND HOLMES G. Benchmarking attribute selection techniques for discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on* **15**(6), 1437–1447 (2003).

[63] HAMILL B. freij-affy-human-91666. `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4412` [Last access: October 2010] (2006).

[64] HO T. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **20**(8), 832–844 (1998).

[65] HOCHBERG Y. AND TAMHANE A. "Multiple Comparison Procedures". John Wiley & Sons (1987).

[66] HUANG C. AND WANG C. A ga-based feature selection and parameters optimizationfor support vector machines. *Expert Systems with applications* **31**(2), 231–240 (2006).

[67] INSTITUTE B. Broad Institute Cancer Program Data Sets. `http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi`. [Last access: November 2010].

[68] INSTITUTE B. Cancer program datasets. `http://\-www.\-broadinstitute.\-org/\-cgi-bin/\-cancer/\-datasets\-.cgi` [Last access: May 2015] (2015).

[69] JAIN A. AND ZONGKER D. Feature selection: Evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **19**(2), 153–158 (1997).

[70] JIAWEI H. AND KAMBER M. "Data mining: concepts and techniques". San Francisco, CA, itd: Morgan Kaufmann (2001).

[71] JIN X., MA E. W., CHENG L. L., AND PECHT M. Health monitoring of cooling fans based on mahalanobis distance with mrmr feature selection. *Instrumentation and Measurement, IEEE Transactions on* **61**(8), 2222–2229 (2012).

[72] JOACHIMS T. Optimizing search engines using clickthrough data. In "Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining", pages 133–142. ACM (2002).

[73] KEARNS M. J. Thoughts on hypothesis boosting. *ML class project* **319**, 320 (1988).

[74] KENDALL M. G. "Rank correlation methods." Charles Griffin and Company (1948).

[75] KIANG M. A comparative assessment of classification methods. *Decision Support Systems* **35**(4), 441–454 (2003).

[76] KIRA K. AND RENDELL L. The feature selection problem: Traditional methods and a new algorithm. In "AAAI", pages 129–134 (1992).

[77] KODOVSKY J., FRIDRICH J., AND HOLUB V. Ensemble classifiers for steganalysis of digital media. *Information Forensics and Security, IEEE Transactions on* **7**(2), 432–444 (2012).

[78] KOHAVI R. AND JOHN G. Wrappers for feature subset selection. *Artificial intelligence* **97**(1-2), 273–324 (1997).

[79] KOLLER D. AND SAHAMI M. Toward optimal feature selection. In "13th International Conference on Machine Learning", pages 284–292 (1996).

[80] KONONENKO I. Estimating attributes: analysis and extensions of relief. In "Machine Learning: ECML-94", pages 171–182. Springer (1994).

[81] KULLBACK S. AND LEIBLER R. A. On information and sufficiency. *The annals of mathematical statistics* pages 79–86 (1951).

[82] KUNCHEVA L. Clustering-and-selection model for classifier combination. In "Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on", volume 1, pages 185–188. IEEE (2000).

[83] KUNCHEVA L. Combining pattern classifiers: methods and algorithms (2004).

[84] KUNCHEVA L. AND WHITAKER C. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning* **51**(2), 181–207 (2003).

[85] LEE Y. J. AND MANGASARIAN O. L. Rsvm: Reduced support vector machines. In "SDM", volume 1, pages 325–361. SIAM (2001).

[86] LEHN-SCHIØLER T., HEGDE A., ERDOGMUS D., AND PRINCIPE J. Vector quantization using information theoretic concepts. *Natural Computing* **4**(1), 39–51 (2005).

[87] LEVIN I. Kdd-99 classifier learning contest: Llsoft's results overview. *SIGKDD explorations* **1**(2), 67–75 (2000).

[88] LIU H. Feature Selection at Arizona State University, Data Mining and Machine Learning Laboratory. `http://featureselection.asu.edu/index.php`[Last access: October 2010] (2010).

[89] LIU H. AND SETIONO R. Chi2: Feature selection and discretization of numeric attributes. In "Proceedings of the Seventh IEEE International Conference on Tools with

Artificial Intelligence, November 5-8, 1995", pages 388–391. IEEE Computer Society (1995).

[90] LIU H. AND YU L. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on* **17**(4), 491–502 (2005).

[91] LIU R. AND YUAN B. Multiple classifiers combination by clustering and selection. *Information Fusion* **2**(3), 163–168 (2001).

[92] MARTINEZ-REGO D., FONTENLA-ROMERO O., PORTO-DIAZ I., AND ALONSO-BETANZOS A. A New Supervised Local Modelling Classifier Based on Information Theory. In "International Joint Conference on Neural Networks, 2009. IJCNN 2009.", pages 2014–2020. IEEE (2009).

[93] MEJÍA-LAVALLE M., SUCAR E., AND ARROYO G. Feature selection with a perceptron neural net. In "Proceedings of the international workshop on feature selection for data mining", pages 131–135 (2006).

[94] MERCER J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character* pages 415–446 (1909).

[95] MEYER P. E. AND BONTEMPI G. On the use of variable complementarity for feature selection in cancer classification. In "Applications of Evolutionary Computing", pages 91–102. Springer (2006).

[96] MIERSWA I., WURST M., KLINKENBERG R., SCHOLZ M., AND EULER T. Yale: Rapid prototyping for complex data mining tasks. In UNGAR L., CRAVEN M., GUNOPULOS D., AND ELIASSI-RAD T., editors, "KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining", pages 935–940, New York, NY, USA (August 2006). ACM.

[97] MOHRI M., ROSTAMIZADEH A., AND TALWALKAR A. "Foundations of machine learning". MIT press (2012).

[98] MØLLER M. F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks* **6**(4), 525–533 (1993).

[99] MUKKAMALA S. AND SUNG A. Feature selection for intrusion detection with neural networks and support vector machines. *Transportation Research Record: Journal of the Transportation Research Board* **1822**(-1), 33–39 (2003).

[100] MUNDRA P., RAJAPAKSE J. C., ET AL.. Svm-rfe with mrmr filter for gene selection. *NanoBioscience, IEEE Transactions on* **9**(1), 31–37 (2010).

[101] NOCK R. AND NIELSEN F. A real generalization of discrete adaboost. In "Proceeding of the 2006 conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy", pages 509–515. IOS Press (2006).

[102] NUTT C. L., MANI D. R., BETENSKY R. A., TAMAYO P., CAIRNCROSS J. G., LADD C., POHL U., HARTMANN C., MCLAUGHLIN M. E., BATCHELOR T. T., ET AL.. Gene Expression-based Classification of Malignant Gliomas Correlates Better with Survival than Histological Classification. *Cancer Research* **63**(7), 1602–1610 (2003).

[103] OLSSON J. AND OARD D. Combining feature selectors for text classification. In "Proceedings of the 15th ACM international conference on Information and knowledge management", pages 798–799. ACM (2006).

[104] OPITZ D. Feature selection for ensembles. In "AAAI/IAAI", pages 379–384 (1999).

[105] PARZEN E. "Statistical inference on time series by Hilbert space methods". Stanford Univ. (1959).

[106] PARZEN E. On estimation of a probability density function and mode. *The annals of mathematical statistics* pages 1065–1076 (1962).

[107] PENG H., LONG F., AND DING C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27**(8), 1226–1238 (2005).

[108] PETRICOIN III E., ARDEKANI A., HITT B., LEVINE P., FUSARO V., STEINBERG S., MILLS G., SIMONE C., FISHMAN D., KOHN E., ET AL.. Use of Proteomic Patterns in Serum to Identify Ovarian Cancer. *The Lancet* **359**(9306), 572–577 (2002).

[109] POMEROY S., TAMAYO P., GAASENBEEK M., STURLA L., ANGELO M., MCLAUGHLIN M., KIM J., GOUMNEROVA L., BLACK P., LAU C., ET AL.. Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression. *Nature* **415**(6870), 436–442 (2002).

[110] PORTO-DÍAZ I., ALONSO-BETANZOS A., AND FONTENLA-ROMERO O. A multiclass classifier based on local modeling and information theoretic learning. In "Proceedings of the Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA)" (2011).

[111] PORTO-DÍAZ I., BOLÓN-CANEDO V., ALONSO-BETANZOS A., AND FONTENLA-ROMERO Ó. Local modeling classifier for microarray gene-expression data. *Artificial Neural Networks–ICANN 2010* pages 11–20 (2010).

[112] PORTO-DÍAZ I., BOLÓN-CANEDO V., ALONSO-BETANZOS A., AND FONTENLA-ROMERO O. A study of performance on microarray data sets for a classifier based on information theoretic learning. *Neural Networks* **24**(8), 888–896 (2011).

[113] PORTO-DÍAZ I., MARTÍNEZ-REGO D., ALONSO-BETANZOS A., AND FONTENLA-ROMERO O. Combining feature selection and local modelling in the kdd cup 99 dataset. *Artificial Neural Networks–ICANN 2009* pages 824–833 (2009).

[114] PORTO-DÍAZ I., MARTÍNEZ-REGO D., ALONSO-BETANZOS A., AND FONTENLA-ROMERO O. Information theoretic learning and local modeling for binary and multiclass classification. *Progress in Artificial Intelligence* **1**(4), 315–328 (2012).

[115] PRINCIPE J. C. AND XU D. Information-theoretic learning using renyi's quadratic entropy. In "Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, Aussois", pages 407–412 (1999).

[116] PRINCIPE J. "Information Theoretic Learning: Renyi's entropy and kernel perspectives". Springer Verlag (2010).

[117] QUINLAN J. Induction of decision trees. *Machine learning* **1**(1), 81–106 (1986).

[118] QUINLAN J. "C4. 5: programs for machine learning". Morgan kaufmann (1993).

[119] RASTRIGIN L. A. AND ERENSTEIN R. H. Method of collective recognition. *Energoizdat, Moscow* (1981).

[120] REDNER R. A. AND WALKER H. F. Mixture densities, maximum likelihood and the em algorithm. *SIAM review* **26**(2), 195–239 (1984).

[121] RENYI A. On measures of information and entropy. In "Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability 1960", pages 547–561 (1961).

[122] RIEKE F., WARLAND D., DE RUYTER VAN STEVENINCK R., AND BIALEK W. Exploring the neural code (1997).

[123] RISSANEN J. Modeling by shortest data description. *Automatica* **14**(5), 465–471 (1978).

[124] SAEYS Y., INZA I., AND LARRAÑAGA P. A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**(19), 2507–2517 (2007).

[125] SANCHEZ GIRALDO L. AND PRINCIPE J. C. A reproducing kernel hilbert space formulation of the principle of relevant information. In "Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on", pages 1–6. IEEE (2011).

[126] SCHAPIRE R. The strength of weak learnability. *Machine learning* **5**(2), 197–227 (1990).

[127] SCHÖLKOPF B. AND SMOLA A. J. "Learning with kernels: Support vector machines, regularization, optimization, and beyond". MIT press (2002).

[128] SEIJO-PARDO B., BOLÓN-CANEDO V., PORTO-DÍAZ I., AND ALONSO-BETANZOS A. Ensemble feature selection for rankings of features. In "Advances in Computational Intelligence", pages 29–42. Springer (2015).

[129] SHANNON C. A mathematical theory of communication. *The Bell System Technical Journal* **27**, 379–423 (1948).

[130] SINGH D., FEBBO P., ROSS K., JACKSON D., MANOLA J., LADD C., TAMAYO P., RENSHAW A., D'AMICO A., RICHIE J., ET AL.. Gene Expression Correlates of Clinical Prostate Cancer Behavior. *Cancer cell* **1**(2), 203–209 (2002).

[131] SIVAGAMINATHAN R. AND RAMAKRISHNAN S. A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert systems with applications* **33**(1), 49–60 (2007).

[132] SPIRA A., BEANE J., SHAH V., STEILING K., LIU G., SCHEMBRI F., GILMAN S., DUMAS Y., CALNER P., SEBASTIANI P., ET AL.. Airway Epithelial Gene Expression in the Diagnostic Evaluation of Smokers with Suspect Lung Cancer. *Nature medicine* **13**(3), 361–366 (2007).

[133] SUYKENS J. A. K., VAN GESTEL T., DE BRABANTER J., DE MOOR B., VANDEWALLE J., SUYKENS J. A. K., AND VAN GESTEL T. "Least squares support vector machines", volume 4. World Scientific (2002).

[134] TIAN E., ZHAN F., WALKER R., RASMUSSEN E., MA Y., BARLOGIE B., AND SHAUGHNESSY JR J. The Role of the Wnt-signaling Antagonist DKK1 in the Development of Osteolytic Lesions in Multiple Myeloma. *The New England Journal of Medicine* **349**(26), 2483–2494 (2003).

[135] TSYMBAL A., PECHENIZKIY M., AND CUNNINGHAM P. Diversity in search strategies for ensemble feature selection. *Information fusion* **6**(1), 83–98 (2005).

[136] TUKEY J. Comparing individual means in the analysis of variance. *Biometrics* pages 99–114 (1949).

[137] TUV E., BORISOV A., RUNGER G., AND TORKKOLA K. Feature selection with ensembles, artificial variables, and redundancy elimination. *The Journal of Machine Learning Research* **10**, 1341–1366 (2009).

[138] VAN'T VEER V., LAURA J., HONGYUE D., VAN DE VIJVER M. J., HE Y. D., HART A. A. M., ET AL.. Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer. *Nature* **415**(6871), 530–536 (2002).

[139] VIDAL-NAQUET M. AND ULLMAN S. Object recognition with informative features and linear classification. In "ICCV", volume 3, page 281 (2003).

[140] VOZARIKOVA E., LOJKA M., JUHAR J., AND CIZMAR A. Performance of basic spectral descriptors and mrmr algorithm to the detection of acoustic events. In "Multimedia Communications, Services and Security", pages 350–359. Springer (2012).

[141] WANG H., KHOSHGOFTAAR T., AND GAO K. Ensemble feature selection technique for software quality classification. In "SEKE", pages 215–220 (2010).

[142] WANG H., KHOSHGOFTAAR T., AND NAPOLITANO A. A comparative study of ensemble feature selection techniques for software defect prediction. In "Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on", pages 135–140. IEEE (2010).

[143] WEISS S. M. AND KULIKOWSKI C. "Computer Systems that Learn - Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems". Morgan Kaufmann Inc. (1991).

[144] WESTON J., ELISSEEFF A., BAKIR G., AND SINZ F. Spider svm toolbox (2006).

[145] WINDEATT T., DUANGSOITHONG R., AND SMITH R. Embedded feature ranking for ensemble mlp classifiers. *IEEE Transactions on Neural Networks* **22**(6), 988–994 (2011).

[146] Data Mining Institute. http://www.cs.wisc.edu/dmi. Last access: 11-26-2008 (2010). University of Wisconsin Madison.

[147] WITTEN I. AND FRANK E. "Data Mining: Practical machine learning tools and techniques". Morgan Kaufmann Pub (2005). `http://www.cs.waikato.ac.nz/ml/weka/`. [Last Access: October 2010].

[148] WOLPERT D. Stacked generalization. *Neural networks* **5**(2), 241–259 (1992).

[149] WRIGHT J., YANG A. Y., GANESH A., SASTRY S. S., AND MA Y. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **31**(2), 210–227 (2009).

[150] YANG C., HUANG C., WU K., AND CHANG H. A novel ga-taguchi-based feature selection method. In "Intelligent Data Engineering and Automated Learning–IDEAL 2008", pages 112–119. Springer (2008).

[151] YANG F. AND MAO K. Robust feature selection for microarray data based on multicriterion fusion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **8**(4), 1080–1092 (2011).

[152] YANG H. H. AND MOODY J. E. Data visualization and feature selection: New algorithms for nongaussian data. In "NIPS", pages 687–702. Citeseer (1999).

[153] YANG J. AND HONAVAR V. Feature subset selection using a genetic algorithm. *Intelligent Systems and Their Applications, IEEE* **13**(2), 44–49 (1998).

[154] YANG Y. AND WEBB G. Proportional k-interval discretization for naive-bayes classifiers. *Machine Learning: ECML 2001* pages 564–575 (2001).

[155] YOU D., HAMSICI O. C., AND MARTINEZ A. M. Kernel optimization in discriminant analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **33**(3), 631–638 (2011).

[156] YU L. AND LIU H. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research* **5**, 1205–1224 (2004).

[157] ZHAO Z. AND LIU H. Searching for interacting features. In "Proceedings of the 20th international joint conference on Artifical intelligence", pages 1156–1161. Morgan Kaufmann Publishers Inc. (2007).

[158] ZHAO Z. AND LIU H. "Spectral Feature Selection for Data Mining". Chapman & Hall / CRC (2012).

[159] ZHENG Z. AND WEBB G. "Stochastic attribute selection committees". Springer (1998).