PH.D. THESIS

FEATURE SELECTION BASED ON INFORMATION THEORY

Boyán Ivanov Bonev

Supervised by **Dr. Miguel Ángel Cazorla Quevedo**
and **Dr. Francisco Escolano Ruiz**

Robot Vision Group
Department of Computer Science and Artificial Intelligence
UNIVERSITY OF ALICANTE

June, 2010

BibTEX reference

```
@phdthesis{thesisBonev2010,
    author={Boyan Bonev},
    title={Feature Selection based on Information Theory},
    school={University of Alicante},
    year={2010},
    month={June}
}
```

If you torture the data for long enough,
in the end they will confess.

Ronald H. Coase,
Nobel Prize in Economics in 1991

Acknowledgements

Acknowledgements to my thesis advisors Miguel Ángel Cazorla and Francisco Escolano for their guidance and support. Also, to my colleagues and friends Pablo Suau, Miguel Ángel Lozano, Juan Manuel Sáez, Antonio Peñalver, Daniela Giorgi and Silvia Biasotti for their contributions and collaboration in several experiments.

I would also like to acknowledge the valuable advices of many people from the Information Theory and Pattern Recognition community, Ajit Rajwade, Anand Rangarajan and Edwin Hancock, among many others. Thanks to all my colleagues from the Department of Computer Science and Artificial Intelligence and the Department of Languages and Information Systems.

Thanks to the free software community.

Boyan Bonev
Alicante, June 2010

Universitat d'Alacant
Universidad de Alicante

Abstract

Along with the improvement of data acquisition techniques and the increasing computational capacity of computers, the dimensionality of the data grows higher. Pattern recognition methods have to deal with samples consisting of thousands of features and the reduction of their dimensionality becomes crucial to make them tractable. Feature selection is a technique for removing the irrelevant and noisy features and selecting a subset of features which describe better the samples and produce a better classification performance. It is becoming an essential part of most pattern recognition applications.

In this thesis we propose a feature selection method for supervised classification. The main contribution is the efficient use of information theory, which provides a solid theoretical framework for measuring the relation between the classes and the features. Mutual information is considered to be the best measure for such purpose. Traditionally it has been measured for ranking single features without taking into account the entire set of selected features. This is due to the computational complexity involved in estimating the mutual information. However, in most data sets the features are not independent and their combination provides much more information about the class, than the sum of their individual prediction power.

Methods based on density estimation can only be used for data sets with a very high number of samples and a low number of features. Due to the curse of dimensionality, in a multi-dimensional feature space the amount of samples required for a reliable density estimation is very high. For this reason we analyse the use of different estimation methods which bypass the density estimation and estimate entropy directly from the set of samples. These methods allow us to efficiently evaluate sets of thousands of features.

For high-dimensional feature sets another problem is the search order of

the feature space. All non-prohibitive computational cost algorithms search for a sub-optimal feature set. Greedy algorithms are the fastest and are the ones which incur less overfitting. We show that from the information theoretical perspective, a greedy backward selection algorithm conserves the amount of mutual information, even though the feature set is not the minimal one.

We also validate our method in several real-world applications. We apply feature selection to omnidirectional image classification through a novel approach. It is appearance-based and we select features from a bank of filters applied to different parts of the image. The context of the task is place recognition for mobile robotics. Another set of experiments are performed on microarrays from gene expression databases. The classification problem aims to predict the disease of a new patient. We present a comparison of the classification performance and the algorithms we present showed to outperform the existing ones. Finally, we successfully apply feature selection to spectral graph classification. All the features we use are for unattributed graphs, which constitutes a contribution to the field. We also draw interesting conclusions about which spectral features matter most, under different experimental conditions. In the context of graph classification we also show how important is the precise estimation of mutual information and we analyse its impact on the final classification results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis we present our research on feature selection for supervised classification problems. In the approach that we describe, the use of Information Theory plays an important role as a strong theoretical framework. On the other hand, non-parametric entropy estimation methods make it possible to work in very high dimensional feature spaces which could not be tackled with the already existing feature selection approaches.

1.1. Feature Selection

Feature selection research dates back to the 60's. Hughes used a general parametric model to study the accuracy of a Bayesian classifier as a function of the number of features [Hughes, 1968]. He concludes: [...] *"measurement selection, reduction and combination are not proposed as developed techniques. Rather, they are illustrative of a framework for further investigation."*

Since then the research in feature selection has been a challenging field, and some have been sceptical about it. In the discussion of the paper [Miller, 1984], J.B. Copas pessimistically commented that *"It has been said: ≪if you torture the data for long enough, in the end they will confess≫. Errors of grammar apart, what more brutal torture can there be than subset selection? The data will always confess, and the confession will usually be wrong."* Also, R.L. Plackett stated: *"If variable elimination has not been sorted out after two decades of work assisted by high-speed computing, then perhaps the time has come to move on to other problems."*

Despite the computationally challenging scenario, the research in this

direction continued. *"As of 1997, when a special issue on relevance including several papers on variable and feature selection was published [Blum and Langley, 1997, Kohavi and John, 1997], few domains explored used more than 40 features."*, [Guyon and Elisseeff, 2003].

Nowadays machine learning and data acquisition advances demand the processing of data with thousands of features. An example is microarray processing. Wang and Gotoh work on molecular classification and qualify feature selection as *"one intractable problem* [...] *is how to reduce the exceedingly high-dimensional gene expression data, which contain a large amount of noise"* [Wang and Gotoh, 2009].

Feature selection is a field of Machine Learning and Pattern Recognition and consists in reducing the dimensionality of the data by eliminating those features which are noisy, redundant or irrelevant for a classification problem. Similarly *feature weighting* is a generalization of feature selection, which assigns weights to each feature. For the case of feature selection the weights are 0 or 1. Another term is *feature extraction* which in the literature refer both to the calculus of the feature values from the original data (e.g. the extraction of cornerness or SIFT descriptors from images) and to the combination of basic features into high level features. In the literature the term *feature extraction* also refers to a different pattern recognition process: the transformation of the feature space, also referred to as *feature transform*. Some widely used *feature transform* methods are Principal Component Analysis and Independent Component Analysis.

There are two kinds of classification problems: supervised and unsupervised. This work is focused on the supervised problem. The task of supervised learning is to build a model (find a function) given a set of points (samples). The goal is to minimize the empirical risk of the function. This is usually not possible and an approximation is needed. There are a number of factors playing role in supervised classification. One of them is which features define the samples.

In supervised classification a classifier is built given a training set of samples and their classes (labels). For testing the classifier a testing set of data is usually available. The classifier has to guess the classes of the test samples. Based on this result a classification accuracy is calculated. Two main factor determine the accuracy: the classification method and the set of features which define each sample.

For an ideal classifier, no feature selection would be necessary, because it would classify the samples disregarding the amount of redundant and noisy features. However, in real world classifiers, feature selection is necessary

and in most cases increases the classification accuracy.

For supervised classification there are two main feature selection approaches: filter and wrapper. In this work both of them are discussed, and the main contribution is a novel filter approach which is based on Information Theory [Cover and Thomas, 1991a, Neemuchwala et al., 2006, Escolano et al., 2009b].

1.2. Information theory

"Shannon's definition of channel capacity, information and redundancy was a landmark [...]. *Since these measurable quantities were obviously important to anyone who wanted to understand sensory coding and perception, I eagerly stepped on the boat"*. (Stated by a well-known visual neuroscientist [Barlow, 2001])

The information theory approach has proved to be effective in solving many computer vision and pattern recognition (CVPR) problems, like image matching, clustering and segmentation, extraction of invariant interest points, feature selection, classifier design, model selection, PCA/ICA, projection pursuit [Hastie and Tibshirani, 1996] and many others. Nowadays researchers are widely exploiting information theory (IT) elements to formulate and solve CVPR problems [Escolano et al., 2009b].

Among these elements we find measures, principles and theories. Entropy, mutual information and Kullback-Leibler divergence are well known measures, typically used either as metrics (e.g. [Goldberger et al., 2006]) or as optimization criteria (e.g. [Tu et al., 2006]). Such is the case of this work, in which we exploit mutual information as an optimization criterion. Some examples of IT principles are the minimum descritption length and the minimax entropy principles. The first one, enunciated by Rissanen, deals with the selection of the simplest model in terms of choosing the shortest code (number of parameters), explaining the data well enough. Minimax entropy, formulated by Christensen, has more to do with perceptual learning. Inferring the probability distribution characterizing a set of images may be posed in terms of selecting, among all distributions matching the statistics of the learning examples, the one with maximum entropy [Zhu et al., 1998]. This ensures that the learnt distribution contains no more information than the examples. Regarding IT theories, these consist in mathematical developments. Two examples are Rate Distortion Theory and the Method of Types. Rate Distortion Theory formalizes the question of what is the minimum expected distortsion produced by lowering the bit

rate, for instance, in lossy compression. Closely related is the Information Bottleneck Method [Tishby et al., 1999] which finds the best tradeoff between accuracy and complexity in a clustering problem. The Method of Types, developed by Csiszár and Körner, provides theoretical insights into calculating the probability of rare events. Types are associated to empirical histograms, and the Sanov's theorem bounds the probability that a given type lies within a certain set of types. An example is the work of [Yuille et al., 2001] where they study the detectability of curves in the presence of background clutter. Due to the Sanov's theorem, the expected number of misclassified distributions depends on the Bhattacharyya distance, which, in turn, is bounded by the Chernoff information. Chernoff information quantifies the overlapping between the distributions associated to different classes of patterns. In [Cazorla and Escolano, 2003] it is exploited for obtaining a range of thresholds for edge detection and grouping. Also in [Suau and Escolano, 2008] they study the applicability of their method to a image class, by means of the Chernoff information.

In addition to measures, principles and theories, another facet of the IT field is the estimation of measures. The problem of estimating entropy, which is the fundamental quantity in IT, is an important research field. In this thesis entropy estimation plays an important role. Different estimation methods are discussed. For example, for high-dimensional data it is necessary to use methods which bypass the distribution estimation and directly estimate entropy from the set of samples.

1.3. Applications

After tackling the feature selection problem we present several real-world applications. The generality of the feature selection problem makes it applicable to a very wide range of domains.

In the first place we present a computer vision application. Its context is a mobile robotics task, and the problem is formalized as an image classification method. Vision is useful and necessary in many indoor environments, where GPS is not available. We use appearance-based image classification in two different ways. On the one hand we use it for dividing the environment in different parts, for example rooms, and we recognize the room or the part of the environment in which the robot is located. This is useful for sub-map applications and for switching from one robotic task to another. On the other hand we use the samples of the classifier for establishing a distance between different images. This way, it is possible to find

the most similar images to a new incoming image, and establish its place in a previously recorded trajectory. If the training images are associated to a physical position, it can be the output of the system.

Secondly we show some gene expression microarray experiments. The databases we use are focused on cancer diseases classification, based on the expression levels of a number of genes. These experiments are the most representative of the method we present, because of the high dimensionality of the samples, and the small number of samples. Our method efficiently deals with them and achieves a better performance than other experiments presented in the literature.

Finally we present an application to graph classification. The graphs are extracted from 3D shapes and the features are structural and spectral measures from the graphs. These measures use only structural information (without attributes for the nodes), which is an important challenge in graph classification. We succeed to find suitable features for accurate classifications in multiclass cases. We also draw conclusions about which spectral measures are most important, depending on the shapes involved in the experiments.

1.4. Motivation and objectives

The feature selection problem is present in pattern recognition since the early 70's. Due to its computational complexity it has been a challenge for the researches and still remains an open problem. The importance of feature selection does not only consist in improving classification performance. In some fields feature selection is used for explaining data. There are data sets with many features whose function and relation to the class is not known by the experts of the field. Such is the case of gene selection problems, where the biologists have to find the relation of a disease with a set of genes. Thus, any improvement of the feature selection problem represents an important advance in pattern recognition, with its implications to many other fields.

As the capacity of computers increases, the dimensionality of the collected data also becomes higher. However, analysing all possible combinations of features is impossible, no matter how fast the computers are becoming. The mathematical developments of the last decade offer the possibility to deal with the high number of dimensions of such data sets. The most important advance, which is exploited in this thesis, is the ability to capture the interactions among all of the selected variables, in an

efficient way. Thus, the objective is to accomplish this task in a feasible computational time.

Chapter 2

Feature Selection

This chapter is devoted to detailing our approach to feature selection. In first place we introduce the general problem and present the wrapper approach to feature selection. Then, we present some image classification examples for explaining the problem. After that, we formulate some mutual information criteria for feature selection. These criteria require entropy estimation methods and we analyse those which are feasible in the high-dimensional case. Finally we discuss the search order from an information-theoretic point of view.

2.1. State of the art

Dimensionality reduction of the raw input variable space is a fundamental step in most pattern recognition tasks. Focusing on the most relevant information in a potentially overwhelming amount of data is useful for a better understanding of the data, for example in genomics [Sima and Dougherty, 2006, Xing et al., 2001, Gentile, 2003]. A properly selected features set significantly improves the classification performance. However, the exhaustive evaluation of the feature sepace is prohibitive, due to the *curse of dimensionality* [Bellman, 1957]. Even the evaluation of large feature sets becomes impossible for some feature selection criteria. Thus, the removal of the noisy, irrelevant, and redundant features is a challenging task.

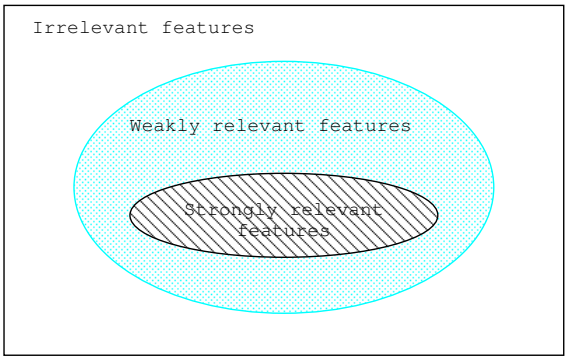

Figure 2.1: Feature space with strongly and weakly relevant features.

2.1.1. General characteristics

There are different definitions of the relevance of a feature. The most commonly used is the one defined by John, Kohavi and Pfleger in [John et al., 1994]. They define two different kinds of relevance (see Figure 2.1):

- Strong relevance: A feature f_i is strongly relevant if its removal degrades the performance of the Bayes Optimum Classifier.
- Weak relevance: A feature f_i is weakly relevant if it is not strongly relevant and there exists a subset of features $\vec{F}'$ such that the performance on $\vec{F}' \cup \{f_i\}$ is better than the performance on just $\vec{F}'$.

The rest of the features are irrelevant. Among them we could say that there are redundant features and noisy features. Both of them could degrade the performance of the classification, depending on the classifier.

A different classification of the features consists of

- Relevant features: those features which, by themselves or in a subset with other features, have information about the class.
- Redundant features: those which can be removed because there is another feature or subset of features which already supply the same information about the class.
- Noisy features: those features which are not redundant and don't have information about the class.

There are many different factors which affect a feature selection algorithm. Blum and Langley [Blum and Langley, 1997] divide them in four classes.

- Search direction. It could be forward feature selection, starting from one feature and adding new ones; backward feature selection, starting from the complete set of features and removing them; and finally a mixed method which adds and removes features (aiming to avoid local minima).
- Search strategy. An exhaustive search is prohibitive, that is why different heuristics can be used as a search strategy.
- Evaluation criterion. The evaluation of the feature subsets is key in feature selection. It is detailed in the next subsection.
- Stopping criterion. When to stop searching in the feature space? Some algorithms complete its execution, while some others need a stopping criterion has to ensure that the solution reached is a good one. The stopping criterion also determines the size of the feature set, which is a delicate issue. The optimal dimension may depend on the number of training instances, as suggested in [Navot et al., 2005].

2.1.2. Filters, wrappers and evaluation criteria

There are two major approaches to dimensionality reduction: feature selection and feature transform. Whilst feature selection reduces the feature set by discarding the features which are not useful for some definite purpose (generally for classification), feature transform methods (also called feature extraction) build a new feature space from the original variables.

Feature selection is a field with increasing interest in machine learning. The literature differentiates among three kinds of feature selection: *filter* method [Guyon and Elisseeff, 2003], *wrapper* method [Blum and Langley, 1997], and *on-line* [Perkins and Theiler, 2003]. Filter feature selection does not take into account the properties of the classifier, as it performs statistical tests to the variables, while wrapper feature selection (see Figure 2.8) tests different feature sets by building the classifier. Finally, on-line feature selection incrementally adds or removes new features during the learning process. All of these methods are based on some feature selection criterion, for example, the criterion of wrappers is the classification performance while the criterion of filters usually is some statistical test on the variables.

Feature selection is a computational problem with combinatorial complexity. Feature selection methods are usually forced to be oriented to finding suboptimal solutions in a feasible number of iterations. This means that the algorithm which selects features from the set cannot be exhaustive. On the other hand, the criterion used for evaluating the feature subsets is a delicate point. It has to estimate the usefulness of a subset accurately and inexpensively.

When there are thousands of features, wrapper approaches become infeasible because the evaluation of large feature sets is computationally expensive. Filter approaches evaluate feature subsets via different statistical measures. Among the filter approaches, a fast way to evaluate individual features is given by their relevance to the classification, by maximizing the mutual information between each single variable and the classification output. As Guyon and Elisseeff state in [Guyon and Elisseeff, 2003], this is usually suboptimal for building a predictor, particularly if the variables are redundant. Conversely, a subset of useful variables may exclude many redundant, but relevant, variables. To overcome this limitation some authors minimize redundancy among the selected features set. Still the problem remains if the criteria are based on individual comparisons between features.

Evaluating all possible interactions among the selected features is a challenge. The reason for this is the fact that estimating mutual information (and entropy) in a continuous multi-dimensional feature space is very complex. As the number of dimensions grows, most estimation methods require an exponentially growing number of samples. Peng et al. [Peng et al., 2005] succeed to estimate mutual information in an incremental way, however, this is only possible when features are selected in an incremental order, starting from zero, and adding one feature per iteration. Backward feature selection would not be possible within their approach.

In this work we use the mutual information criterion and we estimate its value directly from the data. This kind of estimation methods bypass the estimation of the distribution of the samples. Thus, the low number of samples in a high dimensionality is not a problem any more. What actually matters are the distances between samples. In [Hero and Michel, 2002] they use Entropic Spanning Graphs to estimate entropy from a set of samples. However, a method with a lower computational cost is Leonenko's formulation [Leonenko et al., 2008] which uses nearest neighbours for estimating entropy. We study the use of both of them.

Therefore the computational complexity of the estimation does not depend on the number of dimensions, but only on the number of samples. It

allows us to estimate mutual information and thus, maximize dependency between combinations of thousands of features and the class labels. In order to show the efficiency of the method with very high-dimensional feature sets, we perform experiments on gene patterns with thousands of features, as shown in the Applications chapters.

2.1.3. Classical feature selection algorithms

Wrapper approaches often yield better feature sets than filter approaches because they are classifier-dependent. Thus, the bias of the classifier is also taken into account when evaluating a feature set. Some classical wrapper approaches are presented in the works of Blum and Langley [Blum and Langley, 1997] and Kohavi and John [Kohavi and John, 1997, John et al., 1994].

Embedded approaches, in contrast to wrapper approaches, embed the feature selection in the induction algorithm for classification. Some examples are the decision tree algorithms ID3 [Quinlan, 1986] and C4.5 [Quinlan, 1993] by Quinlan. These algorithms use recursive partitioning methods based on the attributes, and perform a greedy search through the search of decision trees. Other embedded approaches are the weighted models [Payne and Edwards, 1998, Perkins and Theiler, 2003].

Filter approaches are classifier-independent. Almuallim and Dietterich's FOCUS algorithm [Almuallim and Dietterich, 1991] was originally designed for the boolean domain. It firstly searches individual features, then pairs of features, then triples, and so on, until it finds the minimal feature set. Another representative filter approach is Kira and Rendell's RELIEF algorithm [Kira and Rendell, 1992]. It evaluates individual features and keeps a set of n best features, but its evaluation criterion is more sophisticate. Gilad-Bachrach et al. describe a margin-based approach to filter feature selection in [Gilad-Bachrach et al., 2004].

There is a large number of relevant works on feature selection. A comprehensive review of many classical feature selection approaches is presented in [Guyon and Elisseeff, 2003]. The performances of several approaches are compared in [Hua et al., 2009].

2.1.4. Evolutionary computation

Evolutionary computation is a field which addresses combinatorial optimization problems by growing a population of candidate solutions. The process is inspired in biological evolutive mechanisms. It iteratively grows

and selects a population by a guided random search. The necessary diversity among the candidate solutions is generated by recombination and mutation. The stochasticity of the algorithm is guided by some evaluation criterion.

A pioneering work with genetic algorithms for feature selection was performed by [Vafaie and Jong, 1995], who proposed an architecture where the feature construction and feature selection are independently selectable modules. Thus, they can control the way in which the two processes interact. In their representation of the data, an individual from the population is a structure of variable length which represents both original features from the data set, and expressions of features. They use the crossover operator which is suitable for variable length hierarchical structures. The evaluation criterion they use is the test of a decision tree built with the selected features. A more recent work on feature selection can be found in [Jirapech-Umpai and Aitken, 2005].

2.1.4.1. Estimation of distribution algorithms

Estimation of distribution algorithms (EDA) is inspired in genetic algorithms. While in a genetic algorithm the population of candidate solutions is typically represented as an array of explicit values, in the EDA paradigm they are a vector representing a probability distribution. Using this vector it is possible to create an arbitrary number of candidate solutions.

In genetic algorithms new candidate solutions are usually generated by stochastically combining and modifying the existing ones. However, the underlying probability distribution is not explicitly specified, while in EDAs, it is. In fact, a population can be obtained by sampling this distribution. Apart from being a more compact representation, it helps to avoid premature convergence of the algorithm.

In [Armañanzas et al., 2008] they explain that the EDA paradigm is applicable to feature subset selection. It is a generalization of feature weighting, ranking and selection. EDA has been used in the context of gene classification, clustering and network inference.

2.1.5. Information theoretic feature selection

Information theory [Cover and Thomas, 1991b, Escolano et al., 2009b] offers a solid theoretical framework for many different machine learning problems. In the case of feature selection, information theoretic methods are usually applied in the filter feature selection way. A classical use of

information theory is found in several feature ranking measures. These consist in statistics from the data which score each feature F_i depending on its relation with the classes c_1, c_2. In this case only two classes are denoted, but these measures could be extended to several classes:

- Shannon entropy

$$H(F_i) = -\int p(f|c_1)\log p(f|c_1)df - \int p(f|c_2)\log p(f|c_2)df$$

- Kullback–Leibler

$$KL(F_i) = -\int p(f|c_1)\log \frac{p(f)}{p(f|c_1)}df - \int p(f|c_2)\log \frac{p(f)}{p(f|c_2)}df$$

- Euclidean distance

$$E(F_i) = \sqrt{\int (p(f|c_1) - p(f|c_2))^2 df}$$

- Kolmogorov dependence

$$KO(F_i) = \int p(f)(p(f|c_1) - p(f|c_2))df$$

One of the most relevant contributions of information theory to the feature selection research is the use of mutual information for feature evaluation. In the following formulation $\vec{F}$ refers to a set of features and C to the class labels.

$$I(\vec{F}; C) = \int\int p(f,c)\log \frac{p(f,c)}{p(f)p(c)}df\,dc.$$

Some approaches evaluate the mutual information between a single feature and the class label. This measure is not a problem. The difficulties arise when evaluating entire feature sets. The necessity for evaluating entire feature sets in a multivariate way is due to the possible interactions among features. While two single features might not provide enough information about the class, the combination of both of them could, in some cases, provide significant information. In [Martínez et al., 2006] they show experimentally that, as $I(F_1; F_2; C)$ decreases, higher is the need of a multivariate model.

There are multivariate filter methods which evaluate the interdependence between pairs of features. Such is the error-weighted uncorrelated shrunken centroid presented by [Yeung and Bumgarner, 2003]. Also in [Wang et al., 2005] they use a multivariate criterion known as correlation-based feature selection. It evaluates a feature set based on the average feature-to-class correlation and the average feature-to-feature correlation. A recent work which uses this filter is [Calvo et al., 2009].

The usual approach for calculating mutual information is to measure entropy and substitute it in the mutual information formula, as detailed in Section 2.4. There are several techniques which estimate entropy in a multi-dimensional space. However, many of them do not perform well for more than three dimensions. This is the case of techniques which first estimate the density underlying the samples, and then plug it into the formula of entropy. A classical example is the Parzen's Window method [Parzen, 1962, Duda et al., 2000]. Density estimation degrades exponentially with respect to the number of dimensions of the patterns. In [Viola, 1995] the width of the Parzen's Window kernel is variable and they estimate the optimal width. Other more recent approaches for density estimation are the wavelet density estimation presented in [Peter and Rangarajan, 2008], which takes a maximum-likelihood approach, as well as the work of [Hong and Schonfeld, 2008], where they use a maximum-entropy maximization-expectation algorithm and estimate densities and entropy from 2D images. Another recent approach is the isosurfaces-based method [Rajwade et al., 2009] which takes a picewise continuous approach for the estimation of image intensity values.

A good alternative for entropy estimation are the techniques which bypass the density estimation, estimating entropy directly from the data. These are usually based on the distances between the data samples. Distances are not more complex in high-dimensional feature space. Then, the complexity of these approaches does not depend on the number of dimensions, but on the number of samples. A classical technique in pattern recognition is the one proposed by Hero and Michel [Hero and Michel, 2002], in which they exploit spanning entropic graphs for entropy estimation. Based on the length of the minimal spanning tree of the data they estimate the Rényi entropy. In [Peñalver et al., 2006, nalver et al., 2009] Peñalver et al. estimate Shannon entropy from the Rényi entropy yielded by Hero and Michel's method. A more straightforward approach is Kozachenko and Leonenko's work [Leonenko et al., 2008] in which they directly estimate Shannon entropy from the data, using the k-nearest neighbour distances.

Following we present some state of the art approaches to feature selection which are based on information theory. Peng et al. [Peng et al., 2005] evaluate feature sets using mutual information. Another information theoretic approach are the Markov Blankets [Koller and Sahami, 1996]. There is also a work in which Torkkola [Torkkola, 2003] uses non-parametric mutual information and quadratic divergence measure for learning discriminative feature transforms. However, their approach is for feature extraction, while the current work is focused on feature selection.

2.1.5.1. Mutual information for feature selection

Mutual information is considered to be a suitable criterion for feature selection [Guyon and Elisseeff, 2003]. Firstly, mutual information is a measure of the reduction of uncertainty about the class labels, due to the knowledge of the features of a data set. Secondly, maximizing the mutual information between the features and the class labels minimizes a lower bound on the Bayes classification error.

However, the estimation of mutual information is a hard task. For this reason univariate techniques have been widely used in the feature selection literature. For instance, correlation based feature selection (CFS) evaluates each feature based on its correlation to the class (its relevancy) and on the correlation among the features in the subset (redundancy). The uncertainty coefficient $U(F_i, F_j)$ is defined, in terms of information theory, as:

$$U(F_i, F_j) = \frac{I(F_i, F_j)}{H(F_i)} = \frac{H(F_i) - H(F_i|F_j)}{H(F_i)}. \tag{2.1}$$

In [Calvo et al., 2009] they successfully adapt this measure to the case of positive and unlabelled samples.

The well-known information gain ratio, which measures the correlation between a feature and the class, is equivalent to the uncertainty coefficient, however, its definition is usually seen as:

$$G_R(C, F_i) = \frac{I(C, F_i)}{H(F_i)} = \frac{H(C) - H(C|F_i)}{H(F_i)}. \tag{2.2}$$

Another similar example is the symmetrical uncertainty measure, which is a normalization of mutual information:

$$SU(F_i, C) = 2\frac{H(F_i) - H(F_i|C)}{H(F_i) + H(C)}, \tag{2.3}$$

where F_i is a single feature and C is a class label. The symmetrical uncertainty is also known as entropy correlation coefficient (ECC).

Alternatively in [Estévez et al., 2009] they propose to use the normalized mutual information:

$$NI_1(F_i, F_j) = \frac{I(F_i; F_j)}{\min\{H(F_i), H(F_j)\}} \tag{2.4}$$

This definition is very similar to the previous one, when used for maximization purposes. A different and less common normalization of mutual information is given by:

$$NI_2((F_i, F_j) = \frac{H(F_i) + H(F_j)}{H(F_i, F_j)}. \tag{2.5}$$

A different work [Liu et al., 2009] proposes to calculate first order dependencies among features but they make a distinction between labelled and unlabelled instances, and iteratively calculate mutual information only on the newly labelled instances.

Other authors [Peng et al., 2005], although performing measures between pairs of features and a single feature and the class label, capture the mutual information between the whole set of features and the class. They achieve this result by calculating the mutual information iteratively in each iteration. The process is explained in detail in Subsection 2.4.5.

Another approximation of mutual information is suggested by [Guo and Nixon, 2009] where they approximate mutual information between the features $\vec{F}$ and the class labels C as:

$$I(\vec{F}; C) \approx \hat{I}(\vec{F}; C) = \sum_i I(F_i; C) - \sum_i \sum_{j>i} I(F_i; F_j) + \sum_i \sum_{j>i} I(F_i; F_j|C), \tag{2.6}$$

where F_i are single features and i and j are in the domain of the feature indexes. This approximation is based on second order product distributions. In [Balagani et al., 2010] they remark that this approximation has two main limitations. Both are attributed to a higher order independence assumption (higher than two). The first one is about not checking for third and higher order dependencies between features. The second one is about not considering third and higher order associations between the features and the class. [Balagani et al., 2010] also remark that the multi-dimensional mutual information can be estimated with the method described in [Bonev et al., 2008].

2.1.5.2. Markov Blankets for feature selection

Markov blankets provide a theoretical framework for proving that some features can be successively discarded (in a greedy way) from the feature set without loosing any information about the class. The Markov blanket of a random variable x_i is a minimal set of variables, such that all other variables conditioned on them, are probabilistically independent on the target x_i. (In a Bayesian network, for example, the Markov Blanket of a node is represented by the set of its parents, children, and the other parents of the children).

Before formally defining a Markov Blanket, let us define the concept of conditional independence. Two variables $\vec{A}$ and $\vec{B}$ are conditionally independent, given a set of variables $\vec{C}$, if $P(\vec{A}|\vec{C},\vec{B}) = P(\vec{A}|\vec{C})$. From this definition some properties of conditional independence can be derived. Let us denote the conditional independence between $\vec{A}$ and $\vec{B}$ given $\vec{C}$ as $\vec{A} \perp \vec{B} \mid \vec{C}$. The properties are:

$$\begin{array}{lll}
\text{Symmetry:} & \vec{A} \perp \vec{B}|\vec{C} & \Longrightarrow \vec{B} \perp \vec{A}|\vec{C} \\
\text{Decomposition:} & \vec{A},\vec{D} \perp \vec{B}|\vec{C} & \Longrightarrow \vec{A} \perp \vec{B}|\vec{C} \text{ and } \vec{D} \perp \vec{B}|\vec{C} \\
\text{Weak union:} & \vec{A} \perp \vec{B},\vec{D}|\vec{C} & \Longrightarrow \vec{A} \perp \vec{B}|\vec{C},\vec{D} \\
\text{Contraction:} & \vec{A} \perp \vec{D}|\vec{B},\vec{C} \text{ and } \vec{A} \perp \vec{B}|\vec{C} & \Longrightarrow \vec{A} \perp \vec{D},\vec{B}|\vec{C} \\
\text{Intersection:} & \vec{A} \perp \vec{B}|\vec{C},\vec{D} \text{ and } \vec{A} \perp \vec{D}|\vec{C},\vec{B} & \Longrightarrow \vec{A} \perp \vec{B},\vec{D}|\vec{C},
\end{array} \tag{2.7}$$

where the Intersection property is only valid for positive probabilities. (Negative probabilities are used in several fields, like quantum mechanics and mathematical finance).

Markov blankets are defined in terms of conditional independence. The set of variables (or features) $\vec{M}$ is a Markov blanket for the variable x_i, if x_i is conditionally independent of the rest of the variables $\vec{F} - \vec{M} - \{x_i\}$, given $\vec{M}$:

$$P(\vec{F} - \vec{M} - \{x_i\}|\vec{M}, x_i) = P(\vec{F} - \vec{M} - \{x_i\}|\vec{M}),$$

or

$$x_i \perp \vec{F} - \vec{M} - \{x_i\} \mid \vec{M}, \tag{2.8}$$

where $\vec{F}$ is the set of features $\{x_1, \ldots, x_N\}$. Also, if $\vec{M}$ is a Markov blanket of x_i, then the class C is conditionally independent of the feature given the Markov blanket: $x_i \perp C \mid \vec{M}$. Given these definitions, if a feature x_i has a Markov blanket among the set of features $\vec{F}$ used for classification, then x_i can safely be removed from $\vec{F}$ without losing any information for predicting the class.

Once a Markov blanket for x_i is found among $\vec{F} = \{x_1, \ldots, x_N\}$ and x_i is discarded, the set of selected (still not discarded) features is $\vec{S} = \vec{F} - \{x_i\}$. In [Koller and Sahami, 1996] it is proven that, if some other feature x_j has a Markov blanket among $\vec{S}$, and x_j is removed, then x_i still has a Markov blanket among $\vec{S} - \{x_j\}$. This property of the Markov blankets makes them useful as a criterion for a greedy feature elimination algorithm. The proof is as follows:

Let $\vec{M}_i \subseteq \vec{S}$ be a Markov blanket for x_i, not necessarily the same blanket which was used to discard the feature. Similarly, let $\vec{M}_j \subseteq \vec{S}$ be a Markov blanket for x_j. It can happen that $\vec{M}_i$ contains x_j, so we have to prove that, after the removal of x_j, the set $\vec{M}'_i = \vec{M}_i - \{x_j\}$, together with the Markov blanket of $\vec{M}_j$, are still a Markov blanket for the initially removed x_i. Intuitively, when we remove x_j, if it forms part of a Markov blanket for some already removed feature x_i, then the Markov blanket of $\vec{M}_j$ will still provide the conditional information that x_j provided in $\vec{M}_i$. By the definition of Markov blankets in Eq. 2.8, we have to show that, given the blanket $\vec{M}'_i \cup \vec{M}_j$, the feature x_i is conditionally independent of the rest of the features; let us denote them as $\vec{X} = \vec{S} - (\vec{M}'_i \cup \vec{M}_j) - \{x_j\}$. We have to show that:

$$x_i \perp \vec{X} \mid \vec{M}'_i \cup \vec{M}_j \tag{2.9}$$

In first place, from the assumption about the Markov blanket of x_j we have that

$$x_j \perp \vec{S} - \vec{M}_j - \{x_j\} \mid \vec{M}_j.$$

Using the Decomposition property (Eq. 2.7) we can decompose the set $\vec{S} - \vec{M}_j - \{x_j\}$ and we obtain

$$x_j \perp \vec{X} \cup \vec{M}'_i \mid \vec{M}_j.$$

Using the Weak union property (Eq. 2.7), we can derive from the last statement:

$$x_j \perp \vec{X} \mid \vec{M}'_i \cup \vec{M}_j. \tag{2.10}$$

For x_i we follow the same derivations and we have

$$x_i \perp \vec{X} \cup (\vec{M}_j - \vec{M}'_i) \mid \vec{M}'_i \cup \{x_j\}$$

and therefore,

$$x_i \perp \vec{X} \mid \vec{M}_j \cup \vec{M}'_i \cup \{x_j\} \tag{2.11}$$

From Eqs. 2.10 and 2.11, and using the Contraction property (Eq. 2.7) we derive that

$$\{x_i\} \cup \{x_j\} \perp \vec{X} \mid \vec{M}_j \cup \vec{M}'_i,$$

which, with the Decomposition property (Eq. 2.7), is equivalent to Eq. 2.9, therefore it is true that after the removal of x_j, the subset $\vec{M}'_i \cup \vec{M}_j$ is a Markov blanket for x_i.

In practice it would result very time-consuming to find a Markov blanket for each feature before discarding it. In [Koller and Sahami, 1996] they propose a heuristic in which they fix a size K for the Markov blankets for which the algorithm searches. The size K depends very much on the nature of the data. If K is too low, it is not possible to find good Markov blankets. If it is too high, the performance is also negatively affected. Among other experiments, the authors of [Koller and Sahami, 1996] also experiment with the "Corral" data set. With the appropriate K they successfully achieve the correct feature selection on it.

2.1.6. Ensemble feature selection

Ensemble classification algorithms build different classifiers in order to achieve a more stable classification. The different classifiers can be built from different samplings of the data set, or from different learning algorithms. The aim of ensembles is classification stability and to avoid overfitting.

A commonly used learner in ensembles is the decision tree. It recursively partitions the feature space into disjoint regions and assigns a response value to each region. The process of building the tree can be considered a feature selection process: in each iteration the decision tree tries all the combinations of variables to achieve the maximum reduction of impurity of a node. The relative importance of the features is based on a multivariate model.

A classical representative of tree ensembles is the Random Forest method [Breiman, 2001]. It builds a forest of random trees based on bagged samples. Its most important characteristic is that it does not overfit. It needs the specification of a number m of features, usually approximate to the square root of the total number of features. Then, each tree of maximum depth is grown on a bootstrap sample of the training set. At each node, m features are selected at random. Finally, from the splits on these m variables the best one is selected and used.

The relative feature ranking of Random Forest, however, does not separate relevant from irrelevant features. Only a rank of the features is gener-

ated. Moreover, trees tend to split on variables with more distinct values. This effect often makes a less relevant feature more prone to splitting on it.

In [Tuv et al., 2006] they combine the use of Random Forest ensemble of fixed-depth trees and re-weight the splits with the samples out of the bag. This helps to produce more unbiased estimation of the features importance. After that they compare statistically the variable importance against artificially constructed noisy variables. Thus, they remove iteratively the effect of important features, allowing the detection of less important variables.

Also in [Genuer et al., 2010] they describe an approach to feature selection with Random Forest. It is a two-steps procedure in which the first one is the preliminary elimination and ranking of features. It consists of sorting the variables by their Random Forest scores of importance and cancelling those with smallest scores. The second step depends on the objective. If the objective is interpretation (finding all the important variables, even if they have a high redundancy), they construct the nested collection of Random Forest using k first variables for all the k's from 1 to the number of features remaining after the first step. The model which leads to the smallest *out-of-the-bag* error determines which variables will constitute the final selection. If, on the contrary, the objective is prediction (finding a small feature set with good classification performance), then they start from the ordered variables and construct an ascending sequence of Random Forest models, by a stepwise use and test of the variables. The variables of the last model are the final selection.

Other works which address feature selection stability are [Saeys et al., 2007a], [Saeys et al., 2008] and [Munson and Caruana, 2009].

2.1.7. Cases when feature selection fails

Is feature selection necessary in all cases? Obviously not. The most obvious scenario in which feature selection is not necessary is a data set in which all the features are relevant and there is no redundancy among them. There could be a number of real data sets with this kind of features.

However, most data sets do have irrelevant features. Again, is feature selection necessary in these cases? It depends on the classifier. An ideal classifier should be able to classify data disregarding the noisy features. Most actual classifiers degrade their performance in the presence of noisy and irrelevant features. There are, however, other meta-learning algorithms whose classification performance is not degraded in the presence of noisy features.

Such is the case of Bagging (bootstrap aggregating,

[Breiman and Breiman, 1996]). It consists of creating sub-samples of the training data and training a model on each sample. Then, the bagging model averages the predictions of these models in order to make predictions. Bagging usually improves the classification performance of the learning algorithm, and has proved to work better when the models have good performance and make uncorrelated mistakes.

In [Munson and Caruana, 2009] they observe that feature selection usually is unnecessary to get good performance from bagged models. In their experiments with Bagging, the bias of the classifier decreases as more features are added. Contrarily, it is the variance which increases with the number of features.

There are three kinds of error sources in a classification experiment:

- noise: the intrinsic error of the data set, the stochasticity about the label
- bias: the error of the predictor, or how close the algorithm is to optimal prediction
- variance: the amount of change of the prediction when changing the training set

In real problems the bias and the noise cannot be measured separately. The experiments of [Munson and Caruana, 2009] show that, as more features are added, the lower is the bias/noise and the higher is the variance. However, they note that this effect is stronger for single trees (the predictor they use) than for bagged trees. Their conclusion is that feature selection finds the feature set that represents the best trade-off between the bias of having too few features and the variance of having too many features. They suggest that feature selection could be best viewed as a model regularization method instead of as a means of distinguishing relevant from irrelevant inputs.

When is bagging better than feature selection or vice versa? In those cases when feature selection is not feasible, a bagging approach is a viable alternative to protect the classification from overfitting problems. On the other hand feature selection is a way for simplifying the data sets.

2.2. Feature extraction from images

The first step in a classification problem is to extract a set of features from the data. In some cases the features come straightforwardly from the

data set but usually the feature extraction design is crucial. Image classification is the example which we will use along this chapter. An image is defined by a sequence of RGB (or HSV, or grayscale) values. However, if these raw values are taken as the feature space of the image, the classification performance would be very poor, no matter how well features are selected. This, of course, depends on the problem. In most computer vision problems the images have important differences among their appearances. The most common one is translation. Rotation and scale are very frequent differences as well. Still more complicated are perspective changes and occlusions. Also, illumination variations cause notable differences in the raw pixel values. All these factors make feature extraction a necessary step in pattern classification. However, there does not exist a set of features which is suitable for any situation. Depending on the task, features have to be designed in one way or another. To ease this design, we propose to use a large feature set and then automatically select the important features.

2.2.1. Low-level filters as image features

In an appearance-based image classification there are many different features which could be extracted from the images. In the following examples we propose to make a collection of features which consist in low-level filters applied to the images.

Biological evidences [Tarr and Bülthoff, 1999, Dill et al., 1993] report that low-level filters play an important role in biological visual recognition systems. For example, Gabor filters model the visual processing carried out by the *simple* and *complex* cells of the primary visual cortex of higher mammals. The organization of these cells results from an unsupervised learning in the visual system, during the first months of life [Meese and Hess, 2004]. In computer vision for object recognition Gabor filters [Escobar and del Solar, 2002], Haar features, steerable filters [Carmichael et al., 2002] and colour co-occurrence histograms [Ekvall et al., 2005, P.Chang and J.Krumm, 1999] are being widely used, providing tolerance to noise and robustness to small perspective changes and occlusions. These methods tend to be probabilistic and machine-learning based, and they are not restricted to any particular environment, nor to some definite set of filters [Schiele and Crowley, 1996].

In this work the low level features are based on filters (edge, colour, corner) which are applied to images. The filter response to an image is a histogram, and several histograms from different filters form a feature vector. A small features subset is selected from a large

bank of filters in order to yield a good recognition rate of the classifier [Blum and Langley, 1997, Jain and Zongker, 1997]. The idea is to perform automatic selection with minimal a priori knowledge about the visual task.

Following we present some experimentation data sets in which filter responses are calculated in a different way. We explain it in the subsequent sections.

2.2.2. Data sets

In this chapter we present five different image classification experiments. Two of them are performed on natural images taken from L.G. Shapiro's ground-truth database [Shapiro, 2005]. Each dataset (examples in the Figures 2.2, 2.3) contains 250 images which belong to 5 classes in the first experiment, and 7 classes in the second one. The image resolutions are about 640×420 pixels.

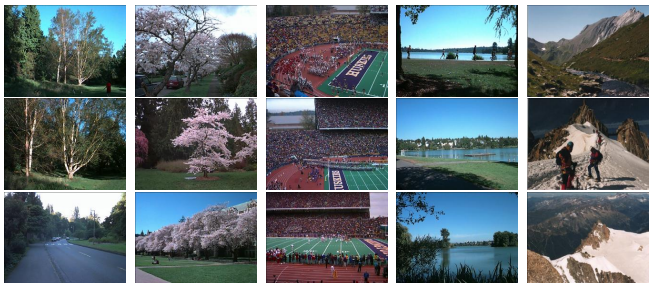

Figure 2.2: Data set 1 examples. Three samples of each one of the 5 classes: Arborgreens, cherries, football, greenlake, Swiss mountains. [Shapiro, 2005]

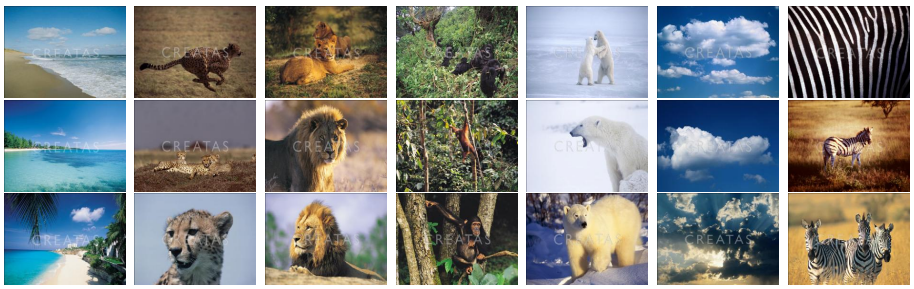

Figure 2.3: Data set 2 examples. Three samples of each one of the 7 classes: Beach, cheetah, lion, monkey, polar bear, sky, zebra. [Shapiro, 2005]

Other two experiments are performed on omnidirectional images taken in indoor and outdoor environments (examples in the Figure.2.4). The

labelling of these images is also performed by hand for supervising the classification. The labels divide the set in different zones of the environment. Each dataset contains 70 images at a 400×400 pixels resolution, 100×600 in their rectified representation. The omnidirectional images were taken at a distance of 1,50m between each other. More detailed information can be read in the Applications chapters.

Figure 2.4: Datasets 3 and 4: Examples of a) omnidirectional indoor view and b) omnidirectional outdoor view. The images are a rectified representation of the original omnidirectional views.

Finally there is an experiment on a data set of 721 (training set) + 470 (test set) images (samples) with a 320x240 resolution, labelled with 6 different classes. This data set was taken with a stereo camera, so range information (depth) is also available.

2.2.3. Filters

The filters we use have a low computational cost. The colour filters are calculated in the HSB colour space in order to maintain some tolerance to lighting variations. These filters return the probability distribution of some definite colour H (from the HSB colour space). The 17 filters are:

- Nitzberg
- Canny
- Horizontal Gradient
- Vertical Gradient

Figure 2.5: Data set 5 examples. The training set contains 721 images taken during an indoor-outdoor walk.

- Gradient Magnitude
- 12 Color Filters $H_i,\ 1 \leq i \leq 12$
- Depth information (for Data set 5)

Some of them are redundant, for example the magnitude and the gradients. Others are similar, like Canny and the magnitude of gradient. Finally, some filters may overlap, for example the colour filters. The colour filters return the probability distribution of some definite colour H (from the HSB colour space).

From the response of the filters we obtain a histogram. The features consist of single bins of the histograms of the filters. An example of filter responses is shown in Figure 2.6.

In the case of data sets 1, 2 and 3, filters are directly applied to each image. However, in the case of omnidirectional images, the filters are applied to four different parts of the image, as shown on the Figure 2.7. The image is divided in four concentric rings to keep rotation invariance (the idea comes from the Transformation Ring Projection, [Tang et al., 1991]). This division also provides additional information by establishing different features for different height levels of the environment. For example, the inner ring has information only about the nearest floor, while the outer ring informs about buildings, trees and sky, but not about the floor.

Finally, when a filter is applied to an image (or to a ring, for omnidirectional images), a histogram is obtained. For the learning process, each bin

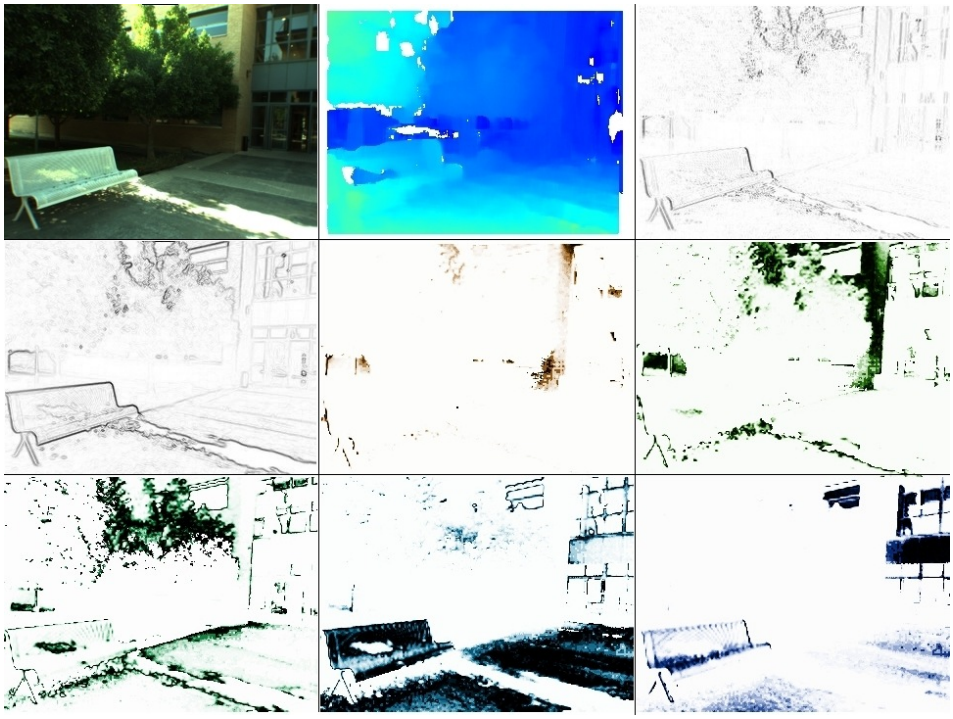

Figure 2.6: Responses of some filters applied to an image. From top-bottom and left-right: input image, depth, vertical gradient, gradient magnitude, and five colour filters. The rest of the filters are not represented as they yield null output for this input image.

of the histogram is a feature of the image. The total number of features N_F per image is

$$N_F = C * K * (B - 1) \tag{2.12}$$

where C is the number of rings (1 for datasets 1 and 2), K the number of filters (17) and B is the number of histogram bins.

2.3. Wrapper and the Cross Validation Criterion

Wrapper feature selection consists in selecting features according to the classification results that these features yield. Therefore wrapper feature selection is a classifier-dependent approach. Contrarily, *filter feature selection* is classifier independent, as it is based on statistical analysis on the input variables (features), given the classification labels of the samples. In

Figure 2.7: An image divided in concentric rings notated as ring 1, ring 2, ring 3 and ring 4, respectively.

filter feature selection the classifier itself is built and tested once the features are selected. Wrappers build classifiers each time a feature set has to be evaluated. This makes them more prone to overfitting than filters. It is also worth mentioning that wrappers are usually applied as a multivariate technique, which means that they test whole sets of features. On the contrary, most filter techniques in the literature are univariate.

Let us place a simple example of wrapping for feature selection. For a supervised[1] classification problem with four features and two classes, suppose we have the following data set containing 9 samples:

	Features				Class
sample 1 = (	x_{11}	x_{12}	x_{13}	x_{14}),	C1
sample 2 = (	x_{21}	x_{22}	x_{23}	x_{24}),	C1
sample 3 = (	x_{31}	x_{32}	x_{33}	x_{34}),	C1
sample 4 = (	x_{41}	x_{42}	x_{43}	x_{44}),	C1
sample 5 = (	x_{51}	x_{52}	x_{53}	x_{54}),	C2
sample 6 = (	x_{61}	x_{62}	x_{63}	x_{64}),	C2
sample 7 = (	x_{71}	x_{72}	x_{73}	x_{74}),	C2
sample 8 = (	x_{81}	x_{82}	x_{83}	x_{84}),	C2
sample 9 = (	x_{91}	x_{92}	x_{93}	x_{94}),	C2

A wrapper feature selection approach could consist in evaluating different combinations of features. In the previous table each single feature is represented by a column: $F_j = (x_{1j}, x_{2j}, \cdots, x_{9j})$, $j \in \{1, \cdots, 4\}$. The evaluation of a feature set involves building a classifier with the selected features and testing it, so we have to divide the data set in two disjoint sets: the train set for building the classifier, and the test set for testing it. For large data sets a good proportion is 75% for the train set and 25% for

[1] In supervised classification, a classifier is built given a set of samples, each one of them labelled with the class to which it belongs. In this section the term classification refers to supervised classification.

the test set. Usually it is adequate to perform the partition on randomly ordered samples. On the other hand, for small data sets there exists a strategy which consists of taking only one sample for the test and repeating the process for all the samples, as explained later in this section.

It is very important to note that, even if we are provided with a separate test set, we can not use it for the feature selection process. In other words, during the wrapper feature selection we use the train set which has to be divided in sub-train and sub-test sets in order to build classifiers and evaluate them. Once finished this process, there is the need to test the final results with a data set which has not been used during the feature selection process.

For example, for the wrapper evaluation of the feature sets (F_1, F_2) and (F_1, F_3), two classifiers have to be built and tested. Let us take as train set the samples $\{S_1, S_3, S_6, S_8, S_9\}$ and the rest, $\{S_2, S_4, S_5, S_7\}$ as test set. Then, the classifiers $\mathcal{C}_1$ and $\mathcal{C}_2$ have to be built with the following data:

$\mathcal{C}_1$: Features		Class	$\mathcal{C}_2$: Features		Class
x_{11}	x_{12}	C1	x_{11}	x_{13}	C1
x_{31}	x_{32}	C1	x_{31}	x_{33}	C1
x_{61}	x_{62}	C2	x_{61}	x_{63}	C2
x_{81}	x_{82}	C2	x_{81}	x_{83}	C2
x_{91}	x_{92}	C2	x_{91}	x_{93}	C2

and tested with the follwing data:

$T_{\mathcal{C}_1}$: Features		$T_{\mathcal{C}_2}$: Features		$Output$: Class
x_{21}	x_{22}	x_{21}	x_{23}	C1
x_{41}	x_{42}	x_{41}	x_{43}	C1
x_{51}	x_{52}	x_{51}	x_{53}	C2
x_{71}	x_{72}	x_{71}	x_{73}	C2

Denoted as *Output* is the set of labels that are expected to be returned by the classifiers for the selected samples. The accuracy of the classifiers is evaluated based on the similarity between its actual output and the desired output. For example, if the classifier $\mathcal{C}_1$ returned $C1, C2, C2, C2$, while the classifier $\mathcal{C}_2$ returned $C1, C2, C1, C1$, then $\mathcal{C}_1$ would be more accurate. The conclusion would be that the feature set (F_1, F_2) works better than (F_1, F_3). This wrapper example is too simple. Actually, drawing such conclusion from just one classification test would be statistically unreliable, and cross validation techniques have to be applied in order to decide which feature set is better than another.

2.3.1. Cross Validation

Cross Validation, also known as rotation estimation, is a validation technique used in statistics and particularly in machine learning. It consists in partitioning a sample of data in several subsets, and performing statistical analysis on different combinations of these subsets. In machine learning and pattern recognition, cross validation (CV) is generally used for estimating the error of a classifier, given a sample of the data. There are two most frequently used CV methods: the *10-fold cross validation* and the *leave-one-out cross validation* (LOOCV).

The 10-fold cross validation (10-fold CV) method consists in dividing the training set in 10 equally sized partitions, and performing 10 classification experiments for calculating their mean error. For each classification, nine of the partitions are put together and used for training (building the classifier), and the other one partition is used for testing. In the next classification, another partition is designed for testing the classifier built with the rest of the partitions. Ten classification experiments are performed so that each partition is used for testing a classifier. Note that the partitioning of the data set is performed only once. It is also important to have a random sample order before partitioning, in order to distribute the samples homogeneously among the partitions.

Leave-one-out cross validation (LOOCV) is used in the cases of very reduced training sets. It is equivalent to K-fold CV with K equal to the total number of samples present the data set. For example, in the well-known NCI60 microarray data set there are 60 samples and 14 classes, where some classes are represented by only two samples. With 10-fold CV there would be cases in which all the samples that represent a class would be in the test set, and the class would not be represented in the training set. Instead, LOOCV would perform 60 experiments, in each one of which, one of the samples would test the classifier built with the resting 59 samples. The LOOCV error would be the mean error of the 60 classification errors. Also,in the cases when the data set is so reduced that there is no separate test set available, the classification results are usually reported in terms of CV error over the training set.

2.3.2. Wrapper feature selection

There are different strategies for generating feature combinations. The only way to assure that a feature set is optimum is the exhaustive search among feature combinations. The *curse of dimensionality* limits this

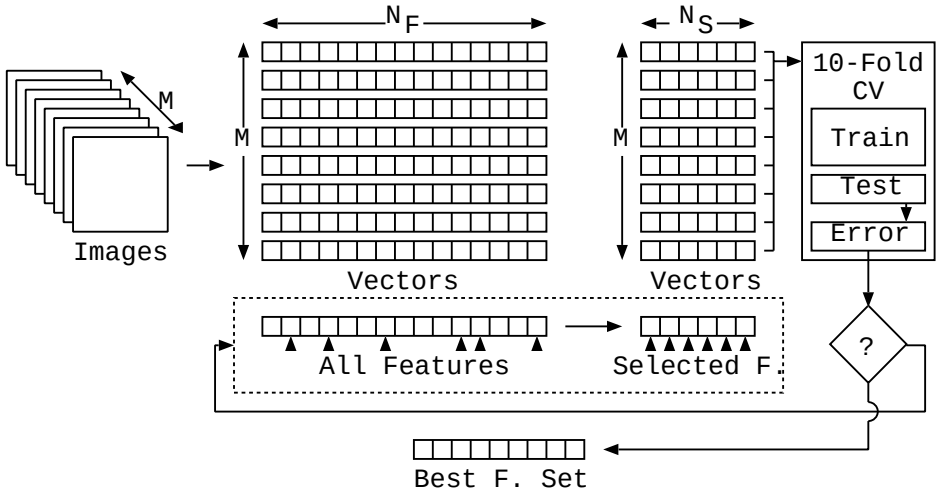

Figure 2.8: The Wrapper Feature Selection process.

search, as the complexity is

$$O\left(\sum_{i=1}^{n}\left(\begin{array}{c} n \\ i \end{array}\right)\right) \tag{2.13}$$

If we model the filters response with histograms of only 2 bins, we have a total of 17 filters for the first two datasets, which is still plausible for an exhaustive search. For the omnidirectional images, if we perform the rings division, we have a total of 68 features, and the total number of combinations is 2.9515×10^{20}. It would take trillions of centuries to evaluate all of them. The maximum number of features which we were able to search exhaustively is $N_F = 3$, that is, feature sets of 3 features maximum.

The fastest way to select from a large amount of features is a greedy strategy. Its computational complexity is

$$O\left(\sum_{i=1}^{n} i\right) \tag{2.14}$$

and the algorithm is as follows:

$DATA_{M \times N_F} \leftarrow$ vectors of all (M) samples
$F_S = \emptyset$
$F = \{feature_1, feature_2, \ldots features_{N_F}\}$
`while` $F \neq \emptyset$

$\forall i \mid feature_i \in F$
$D_S = DATA(F_S \cup \{feature_i\})$
$E_i = \texttt{10FoldCrossValid}(D_S)$
$selected = \arg\min_i E_i$
`/* and also store` E_i `*/`
$F_S = F_S \cup \{feature_{selected}\}$
$F = F \sim \{feature_{selected}\}$
`end`

At the end of each iteration a new feature is selected and its CV error is stored. The process is also outlined in Figure 2.8.

Let us have a look at the data sets presented in the previous subsection. For the first two of them, the best three features are colour filters, as the images are different enough to be discriminated by colour histograms. With these three features the Cross Validation (CV) error is 12,96%. In the omnidirectional images the situation is different, as images from different classes have similar colours. The best three features for the 8-classes indoor experiment are: Canny, Color Filter 1, and Color Filter 5, all of them applied to the ring 2 (notation illustrated in the Figure 2.7). The Cross Validation (CV) error yielded with only 3 features is 24,52%, much better than the 30,57% CV error yielded by the complete set of 68 features. For the outdoor experiment the best three features are Color Filter 9 on ring 2, Nitzberg and Color Filter 8 on ring 3. In this case 3 features are not enough (see Table 2.1) for decreasing the error, as this experiment needs more features to be selected (graphically illustrated in Figure 2.10).

In Table 2.1 we can see that feature selection on the nature datasets performs in a different way than the localization images. These images present more significant improvements with feature selection. This is due to a greater similarity among localization images, and the need of a subtler discrimination. On the other hand, the total number of features for these experiments is larger, as we divide the images into four different zones. This makes feature selection play an important role for a good classification.

These differences are better visualized in the Figures 2.9 and 2.10. There are plotted the CV classification errors for different number of features, from 0 to N_F. By comparing these curves we can see that recognition error rates for the nature datasets do not rise much when using the complete feature set. On the other hand, the curve is not as parabolic as in the localization dataset.

There also are differences between the indoor and outdoor experiments. We can see that the indoor problem needs just 45 features (out of 204) to

Experiment			All Features		Feature Sel.	
	Dataset	Classes	Features.	CV.Err.	Features.	CV.Err.
nature		5	17	9,86%	3	12,96%
	Data set 1	5	51	8,41%	13	7,15%
	Data set 2	7	51	22.98%	39	17,77%
indoor	Data set 3	8	68	30.57%	3	24.52%
		8	68	30.57%	38	13.33%
		8	204	22.38%	45	7.51%
		8	748	26.95%	234	3.33%
		16	68	49.00%	31	26.43%
		16	204	51.67%	66	22.86%
outdoor	Data set 4	8	68	12.42%	3	16.07%
		8	68	12.42%	19	3.93%
		8	204	7.96%	138	1.42%
		8	748	12.50%	184	1.25%

Table 2.1: Cross Validation errors without and with Feature Selection, as well as the number of selected features. Several experiments are presented: two data sets with the nature images, and the indoor and outdoor datasets, with different number of classes and different number of bins in the filters histogram (2 bins for 68 feat., 4 bins for 204 feat. and 12 bins for 748 feat.)

obtain a good classification (detailed results on Table 2.1). The outdoor problem needs 138 features for a better performance. This is due to a higher complexity of the natural environment, and the presence of noise coming from people and several objects.

The number of bins used to generate features is an important factor for the feature selection results. A larger number of bins can (but not necessarily) improve the classification. In the case of localization datasets the use of 12 bins (748 total features) do improve classification, while 2 bins (68 total features) do worsen classification. See Figure 2.11 where this difference is illustrated. On the Table 2.1 are represented several feature selection results using different number of histogram bins. An excessive number of bins overfits the classifier.

Another important factor is the number of classes. As the number of classes increases, the classification performance decays, and more features are needed. This can be observed in the difference of the performance of the nature images with 5 classes and those with 7 classes. Also, in the Figure 2.12 we compare feature selection performance on the 8-classes and

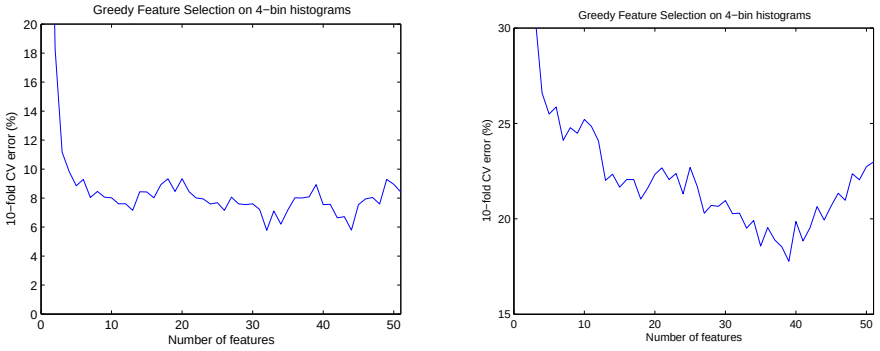

Figure 2.9: Error of Feature Selection on data sets 1 and 2.

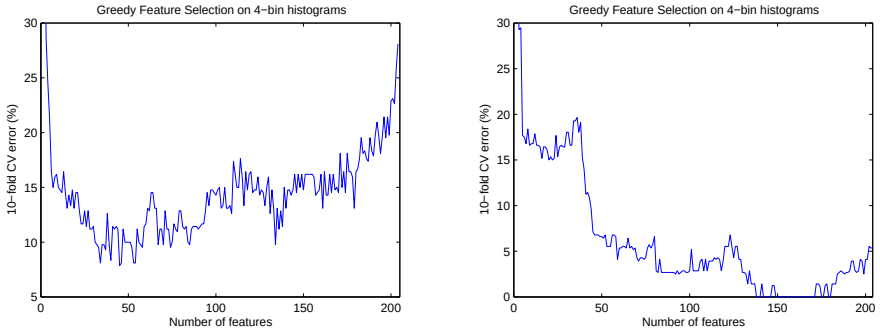

Figure 2.10: Comparison of the Feature Selection results for the 8-classes indoor (data set 3) and 8-classes outdoor (data set 4) experiments.

the 16-classes experiments. The evolution of the CV error is similar, with a vertical offset.

Finally it is worth commenting out some observations about the selected filters. For the natural images, there is no structure defined (as explained in Section 2.2) The feature selection yields colour filters, for feature subspaces with less than 5 features. For larger feature subspaces, gradient filters become important, in particular the Gradient Magnitude and the Nitzberg filter.

For the case of omnidirectional images, not only filters are selected but also the rings to which filters have to be applied. The rings whose features are selected are usually the ring 2 and ring 3. The ring 1 and the ring 4 always have a little lower number of selected features.

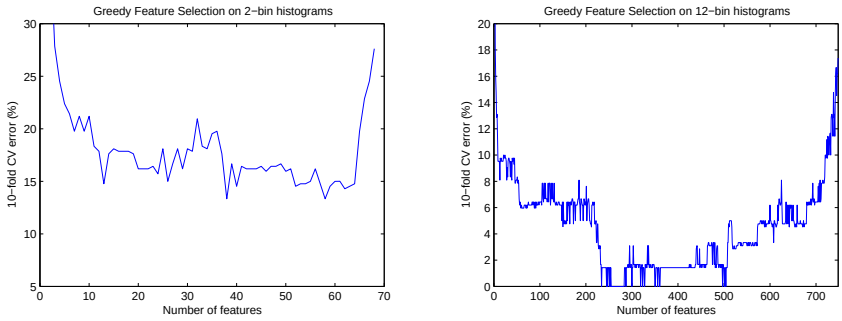

Figure 2.11: Comparison of Feature Selection using 2 bins and 12 bins histograms, on the 8-classes indoor experiment.

Figure 2.12: Comparison of 8-classes and 16-classes in the same environment (indoor). The 16-classes experiment reports a higher errors but similar performance evolution.

2.4. Mutual Information Criterion

In this section we detail our feature selection approach. First, we discuss the feature selection criteria we use. There is an important difference between multivariate and univariate criteria, and this is an important point in this work. Next we detail the entropy estimation method that we use for evaluating the mutual information of the data sets.

2.4.1. Feature Selection criteria

Wrappers offer a good approach to multivariate feature selection, as shown in the previous section. They are, however, classifier-dependent and prone to overfitting effects in the results. The alternative is to use a filter

feature selection criterion. In the following subsections we present and discuss our approach to multivariate feature selection.

2.4.2. Criteria for Filter Feature Selection

In most Feature Selection approaches there are two well differentiated points: the search algorithm and the selection criterion. Another important issue is the stopping criterion used to determine when an algorithm has achieved a good maximum in the feature space. This section is centred on some IT-based feature selection criteria. Regarding the way that feature combinations are generated, or search order, Section 2.6 discusses it. An exhaustive search among the features set combinations would have a combinatorial complexity with respect to the total number of features. In the following subsections we assume the use of a Greedy Forward Feature Selection algorithm, which starts from a small feature set, and adds one feature in each iteration.

In the presence of thousands of features the Wrapper approaches are infeasible because the evaluation of large feature sets is computationally expensive. In Filter Feature Selection the feature subsets are statistically evaluated. Univariate filter methods evaluate single features without taking into account the interactions among them. A way to measure the relevance of a feature for the classification is to evaluate its mutual information with the classification labels [Cover and Thomas, 1991b]. This is usually suboptimal for building predictors [Guyon and Elisseeff, 2003] due to the possible redundancy among variables. Peng et al. [Peng et al., 2005] use not only information about the relevance of a variable but also an estimation of the redundancy among the selected variables. This is the min-Redundancy Max-Relevance (mRMR) feature selection criterion. For their measures they estimate mutual information between pairs of variables. Another feature selection approach estimates the mutual information between a whole set of features and the classes for using the infomax criterion. The idea of maximizing the mutual information between the features and the classes is similar to the example illustrated in the Figure 2.13, where we can see that in the first plot the classifier is the optimal, as well as the mutual information between the two dimensions of the data and the classes (black and white) is maximum. The mutual information is a mathematical measure which captures the dependencies which provide information about the class labels, disregarding those dependencies among features which are irrelevant to the classification.

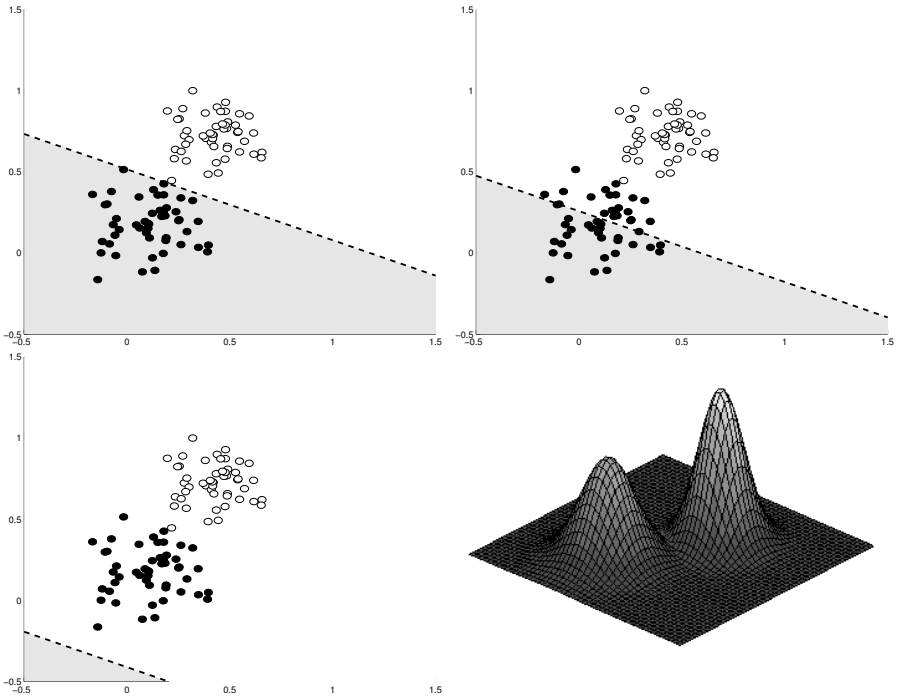

Figure 2.13: Three different classifiers applied to the same data, obtained from two different Gaussian distributions (represented at bottom right of the figure). Mutual information between inputs and class labels, obtained by means of entropic graphs (as explained in chapters 3 and 4), are 6.2437, 3.5066 and 0, respectively. The highest value is achieved in the first case; it is the *Infomax* classifier.

2.4.3. Mutual Information for Feature Selection

The primary problem of feature selection is the criterion which evaluates a feature set. It must decide whether a feature subset is suitable for the classification problem, or not. The optimal criterion for such purpose would be the Bayesian error rate for the subset of selected features:

$$E(S) = \int_{\vec{S}} p(\vec{S}) \left(1 - \max_i (p(c_i|\vec{S}))\right) d\vec{S}, \tag{2.15}$$

where $\vec{S}$ is the vector of selected features and $c_i \in C$ is a class from all the possible classes C existing in the data.

The Bayesian error rate is the ultimate criterion for discrimination, however, it is not useful as a cost, due to the non-linearity of the $\max(\cdot)$ function. Then, some alternative cost function has to be used. In the literature there are many bounds on the Bayesian error. An upper bound obtained by Hellman and Raviv (1970) is:

$$E(\vec{S}) \leq \frac{H(C|\vec{S})}{2}.$$

This bound is related to mutual information, becuase mutual information can be expressed as

$$I(\vec{S}; C) = H(C) - H(C|\vec{S})$$

and $H(\vec{C})$ is the entropy of the class labels which do not depend on the feature subspace $\vec{S}$. Therefore the mutual information maximization is equivalent to the maximization of the upper bound (Eq. 2.15) of the Bayesian error. There is a Bayesian error lower bound as well, obtained by Fano (1961), and is also related to mutual information.

The relation of mutual information with the Kullback-Leibler (KL) divergence also justifies the use of mutual information for feature selection. The KL divergence is defined as:

$$KL(P||Q) = \int_{\vec{x}} p(\vec{x}) \log \frac{p(\vec{x})}{q(\vec{x})} d\vec{x}$$

for the continuous case and

$$KL(P||Q) = \sum_{\vec{x}} p(\vec{x}) \log \frac{p(\vec{x})}{q(\vec{x})}$$

for the discrete case. From the definition of mutual information, and given that the conditional entropy can be expressed as $p(x|y) = p(x,y)/p(y)$, we have that:

$$
\begin{aligned}
I(X;Y) &= \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} && (2.16)\\
&= \sum_{y\in Y} p(y)\sum_{x\in X} p(x|y)\log\frac{p(x|y)}{p(x)} && (2.17)\\
&= \sum_{y\in Y} p(y)KL\left(p(x|y)||p(x)\right) && (2.18)\\
&= E_Y(KL\left(p(x|y)||p(x)\right)). && (2.19)
\end{aligned}
$$

Then, maximizing mutual information[2] is also equivalent to maximizing the expectation of the KL divergence between the class-conditional densities $P(\vec{S}|\vec{C})$ and the density of the feature subset $P(\vec{S})$. In other words, the density over all classes has to be as distant as possible from the density of each class in the feature subset. Mutual information maximization provides a trade-off between discrimination maximization and redundancy minimization.

There are some practical issues involved into the maximization of mutual information between the features and the classes. Nowadays feature selection problems involve thousands of features, in a continuous feature space. Estimating mutual information between a high-dimensional continuous set of features, and the class labels, is not straightforward, due to the entropy estimation. There exist graph-based methods which do not need the density estimation of the data, thus, allowing to estimate the entropy of high-dimensional data with a feasible computational complexity.

2.4.4. Individual features evaluation, dependence and redundancy

Some works on feature selection avoid the multi-dimensional data entropy estimation by working with single features. This, of course, is not equivalent to the maximization of $I(\vec{S};C)$. In the approach of Peng et al. [Peng et al., 2005] the feature selection criterion takes into account the mutual information of each separate feature and the classes, but also subtract

[2] Some authors refer to the maximization of the mutual information between the features and the classes as *infomax criterion*.

the redundancy of each separate feature with the already selected ones. It is explained in the next subsection.

A simpler approach is to limit the cost function to evaluate only the mutual information between each selected feature $x_i \in \vec{S}$ and the classes C:

$$I(\vec{S}^*; C) \approx \sum_{x_i \in \vec{S}} I(x_i; C) \tag{2.20}$$

Such cost can be effective for some concrete cases, as reason Vasconcelos et al. in [Vasconcelos and Vasconcelos, 2004]. The expression of the mutual information of the optimal feature subset $\vec{S}^* = \{x_1^*, x_2^*, \cdots, x_{N_{S^*}}^*\}$ of size N_{S^*} can be decomposed into the following sum:

$$\begin{aligned} I(\vec{S}^*; C) &= \sum_{i=1}^{N} I(x_i^*; C) - \\ &- \sum_{i=2}^{N} \Big(I(x_i^*; \vec{S}_{1,i-1}^*) - I(x_i^*; \vec{S}_{1,i-1}^* | C) \Big), \end{aligned} \tag{2.21}$$

where x_i^* is the i^{th} most important feature and $\vec{S}_{1,i-1}^* = \{x_1^*, \ldots, x_{i-1}^*\}$ is the set of the first $i-1$ best features, which have been selected before selecting x_i^*. This expression is obtained by applying the chain rule of mutual information. For the mutual information between N variables $X_1, \ldots, X_N$, and the variable Y, the chain rule is:

$$I(X_1, X_2, \ldots, X_N; Y) = \sum_{i=1}^{N} I(X_i; Y | X_{i-1}, X_{i-2}, \ldots, X_1)$$

The property from Eq. 2.21 is helpful for understanding the kind of trade-off between discriminant power maximization and redundancy minimization which is achieved by $I(\vec{S}^*; C)$. The first summation measures the individual discriminant power of each feature belonging to the optimal set. The second summation penalizes those featues x_i^* which, together with the already selected ones $\vec{S}_{1,i-1}^*$, are jointly informative about the class label C. This means that if $\vec{S}_{1,i-1}^*$ is already informative about the class label, the the informativeness of the feature x_i^* is the kind of redundancy which is penalized. However, those features which are redundant, but do not inform about the class label, are not penalized.

Given this property, Vasconcelos et al. [Vasconcelos and Vasconcelos, 2004] focus the feature selection problem on visual processing with low level features. Several studies report

that there exist universal patterns of dependence between the features of biologically plausible image transformations. These universal statistical laws of dependence patterns are independent of the image class. This conjecture implies that the second summation in 2.21 would probably be close to zero, because of the assumption that the redundancies which carry information about the class, are insignificant. In this case, only the first summation would be significant for the feature selection process, and the approximation in Eq. 2.20 would be valid. This is the most relaxed feature selection cost, in which the discriminant power of each feature is individually measured.

An intermediate strategy was introduced by Vasconcelos et al. They sequentially relax the assumption that the dependencies are not informative about the class. By introducing the concept of l-decomposable feature sets they divide the feature set into disjoint subsets of size l. The constraint is that any dependence which is informative about the class label, has to be between the features of the same subset, but not between susbsets. If $\vec{S^*}$ is the optimal feature subset of size N and it is l-decomposable into the subsets $T_1, \ldots, T_{\lceil N/l \rceil}$, then

$$\begin{aligned} I(\vec{S^*}; C) &= \sum_{i=1}^{N} I(x_i^*; C) - \\ &- \sum_{i=2}^{N} \sum_{j=1}^{\lceil i-1/l \rceil} \left(I(x_i^*; \vec{\tilde{T}}_{j,i}) - I(x_i^*; \vec{\tilde{T}}_{j,i} | C) \right), \end{aligned} \tag{2.22}$$

where $\vec{\tilde{T}}_{j,i}$ is the subset of $\vec{T}_j$ containing the features of index smaller than k. This cost function makes possible an intermediate strategy which is not as relaxed as Eq. 2.20, and is not as strict as Eq. 2.21. The gradual increase of the size of the subsets $\vec{T}_j$ allows to find the l at which the assumption about non-informative dependences between the subsets becomes plausible.

The assumption that the redundancies between features are independent of the image class is not realistic in many feature selection problems, even in the visual processing field. Following we analyse some approaches which do not make the assumption of Eq. 2.20. Instead they take into consideration the interactions between all the features.

2.4.5. The min-Redundancy Max-Relevance Criterion

Peng et al. present in [Peng et al., 2005] a Filter Feature Selection criterion based on mutual information estimation. Instead of estimating the

mutual information $I(\vec{S};C)$ between a whole set of features and the class labels (also called prototypes), they estimate it for each one of the selected features separately. On the one hand they maximize the relevance $I(x_j;C)$ of each individual feature $x_j \in \vec{F}$. On the other hand they minimize the redundancy between x_j and the rest of selected features $x_i \in \vec{S}, i \neq j$. This criterion is known as the min-Redundancy Max-Relevance (mRMR) criterion and its formulation for the selection of the m-th feature is:

$$\max_{x_j \in \vec{F}-\vec{S}_{m-1}} \left[I(x_j;C) - \frac{1}{m-1} \sum_{x_i \in \vec{S}_{m-1}} I(x_j;x_i) \right]. \tag{2.23}$$

This criterion can be used by a greedy algorithm, which, in each iteration takes a single feature and decides whether to add it to the selected features set, or to discard it. This strategy is called forward feature selection. With the mRMR criterion each evaluation of a new feature consists of estimating the mutual information (MI) between a feature and the prototypes, as well as the MI between that feature and each one of the already selected ones (Eq. 2.23). An interesting property of this criterion is that is equivalent to first-order incremental selection using the Max-Dependency (MD) criterion. The MD criterion, presented in the next subsection, is the maximization of the mutual information between all the selected features (together) and the class, $I(\vec{S},C)$.

First-order incremental selection consists in starting with an empty feature set and add, incrementally, a single feature in each subsequent iteration. This implies that by the time the m-th feature x_m has to be selected, there already are $m-1$ selected features in the set of selected features $\vec{S}_{m-1}$. By defining the following measure for the $x_1, x_2, \ldots, x_n$ scalar variables (i.e., single features),

$$\begin{aligned} &J(x_1, x_2, \cdots, x_n) = \\ &= \int \cdots \int p(x_1, x_2, \ldots, x_n) \log \frac{p(x_1, x_2, \ldots, x_n)}{p(x_1)p(x_2)\cdots p(x_n)} dx_1 \cdots dx_n, \end{aligned}$$

it can be seen that selecting the m-th feature with mRMR first-order incremental search is equivalent to maximizing the mutual information between $\vec{S}_m$ and the class C. Eqs. 2.24 and 2.25 represent the simultaneous maximization of their first term and minimization of their second term.

Following we show the equivalence with mutual information (Eq. 2.26).

$$
\begin{aligned}
I(\vec{S}_m;C) &= J(\vec{S}_m,C) - J(\vec{S}_m) && (2.24)\\
&= J(\vec{S}_{m-1},x_m,C) - J(\vec{S}_{m-1},x_m) && (2.25)\\
&= J(x_1,\ldots,x_{m-1},x_m,C) - J(x_1,\ldots,x_{m-1},x_m)\\
&= \int\cdots\int p(x_1,\ldots,x_m,C)\log\frac{p(x_1,\ldots,x_m,C)}{p(x_1)\cdots p(x_m)p(C)}dx_1\cdots dx_m dC\\
&\quad -\int\cdots\int p(x_1,\ldots,x_m)\log\frac{p(x_1,\ldots,x_m)}{p(x_1)\cdots p(x_m)}dx_1\cdots dx_m\\
&= \int\cdots\int p(x_1,\ldots,x_m,C)\log\left(\frac{p(x_1,\ldots,x_m,C)}{p(x_1)\cdots p(x_m)p(C)}\right.\\
&\quad \left.\cdot\frac{p(x_1)\cdots p(x_m)}{p(x_1,\ldots,x_m)}\right)dx_1\cdots dx_m dC\\
&= \int\cdots\int p(x_1,\ldots,x_m,C)\log\frac{p(x_1,\ldots,x_m,C)}{p(x_1,\ldots,x_m)p(C)}dx_1\cdots dx_m dC\\
&= \int\int p(\vec{S}_m,C)\log\frac{p(\vec{S}_m,C)}{p(\vec{S}_m)p(C)}d\vec{S}_m dC = I(\vec{S}_m;C). && (2.26)
\end{aligned}
$$

This reasoning can also be denoted in terms of entropy. We can write $J(\cdot)$ as:

$$J(x_1,x_2,\cdots,x_n) = H(x_1)+H(x_2)+\cdots+H(x_n)-H(x_1,x_2,\cdots,x_n),$$

therefore

$$J(\vec{S}_{m-1},x_m) = J(\vec{S}_m) = \sum_{x_i\in\vec{S}_m} H(x_i) - H(\vec{S}_m)$$

and

$$J(\vec{S}_{m-1},x_m,C) = J(\vec{S}_m,C) = \sum_{x_i\in\vec{S}_m} H(x_i) + H(C) - H(\vec{S}_m,C),$$

which, substituted in Eq. 2.25 results in:

$$
\begin{aligned}
&J(\vec{S}_{m-1},x_m,C) - J(\vec{S}_{m-1},x_m) =\\
&\sum_{x_i\in\vec{S}_m} H(x_i) + H(C) - H(\vec{S}_m,C) - \left[\sum_{x_i\in\vec{S}_m} H(x_i) - H(\vec{S}_m)\right]\\
&= H(C) - H(\vec{S}_m,C) + H(\vec{S}) = I(\vec{S},C)
\end{aligned}
$$

There is a variant of the mRMR criterion. In [Ponsa and López, 2007] it is reformulated using a different representation of redundancy. They propose to use a coefficient of uncertainty which consists of dividing the MI between two variables x_j and x_i by the entropy of $H(x_i)$, $x_i \in \vec{S}_{m-1}$.

$$\frac{I(x_j;x_i)}{H(x_i)} = \frac{H(x_i) - H(x_i|x_j)}{H(x_i)} = 1 - \frac{H(x_i|x_j)}{H(x_i)}$$

This is a non-symmetric definition which quantifies the redundancy with a value between 0 and 1. The highest value possible for the negative term $H(x_i|x_j)/H(x_i)$ is 1, which happens when x_i and x_j are independent, then $H(x_i|x_j) = H(x_i)$. The lowest value is 0, when both variables are completely dependent, disregarding their entropy. With this redundancy definition the mRMR criterion expression 2.23 becomes:

$$\max_{x_j \in \vec{F} - \vec{S}_{m-1}} \left[I(x_j;C) - \frac{1}{m-1} \sum_{x_i \in \vec{S}_{m-1}} \frac{I(x_j;x_i)}{H(x_i)} \right] \tag{2.27}$$

2.4.6. The Max-Dependency criterion

The Max-Dependency (MD) criterion consists of maximizing the mutual information between the set of selected features $\vec{S}$ and the class labels C:

$$\max_{\vec{S} \subseteq \vec{F}} I(\vec{S};C) \tag{2.28}$$

Then, the m-th feature is selected according to:

$$\max_{x_j \in \vec{F} - \vec{S}_{m-1}} I(\vec{S}_{m-1}, x_j; C) \tag{2.29}$$

Whilst in mRMR the mutual information is incrementally estimated by estimating it between two variables of one dimension, in MD the estimation of $I(\vec{S};C)$ is not trivial because $\vec{S}$ could consist of a large number of features. In [Bonev et al., 2008] such estimation is performed with the aid of Entropic Spanning Graphs for entropy estimation [Hero and Michel, 2002]. This entropy estimation is suitable for data with a high number of features and a small number of samples, because its complexity depends on the number n_s of samples ($O(n_s \log(n_s))$), but not on the number of dimensions. The MI can be calculated from the entropy estimation in two different ways, with the conditional entropy and with the joint entropy:

$$I(\vec{S};C) = \sum_{x\in\vec{S}}\sum_{c\in\vec{C}} p(x,c)log\frac{p(s,c)}{p(x)p(c)} \quad (2.30)$$

$$= H(\vec{S}) - H(\vec{S}|C) \quad (2.31)$$

$$= H(\vec{S}) + H(C) - H(\vec{S},C) \quad (2.32)$$

where x is a feature from the set of selected features $\vec{S}$ and c is a class label belonging to the set of prototypes C.

Provided that entropy can be estimated for high-dimensional data sets, different IT-based criteria can be designed, depending on the problem. For example, the Max-min-Dependency (MmD) criterion (Eq. 2.33), in addition to the Max-Dependency (MD) maximization, also minimizes the mutual information between the set of discarded features and the classes:

$$\max_{\vec{S}\subseteq\vec{F}} \left[I(\vec{S};\vec{C}) - I(\vec{F}-\vec{S};\vec{C}) \right] \quad (2.33)$$

Then, for selecting the m-th feature, Eq. 2.34 has to be maximized:

$$\max_{x_j\in\vec{F}-\vec{S}_{m-1}} \left[I(\vec{S}_{m-1}\cup\{x_j\};\vec{C}) - I(\vec{F}-\vec{S}_{m-1}-\{x_j\};\vec{C}) \right] \quad (2.34)$$

The aim of the MmD criterion is to avoid leaving out features which have information about the prototypes. In the Figure 2.14 we show the evolution of the criterion as the number of selected features increases, as well as the relative values of the terms $I(\vec{S};\vec{C})$ and $I(\vec{F}-\vec{S};\vec{C})$, together with the 10-fold CV and test errors of the feature sets.

2.4.7. Mutual Information Estimation

Mutual information (MI) is used in filter feature selection as a measure of the dependency between a set of features $\vec{S}$ and the classification prototypes $\vec{C}$. MI can be calculated in different ways. In [Neemuchwala et al., 2006], Neemuchwala et al. study the use of entropic graph for MI estimation. In the present approach MI calculation is based on conditional entropy estimation:

$$I(\vec{S};\vec{C}) = H(\vec{S}) - H(\vec{S}|\vec{C}).$$

Using the Eq. 2.31 the conditional entropy $H(\vec{S}|\vec{C})$ has to be calculated. To do this, $\sum H(\vec{S}|C=c)p(C=c)$ entropies have to be calculated, and

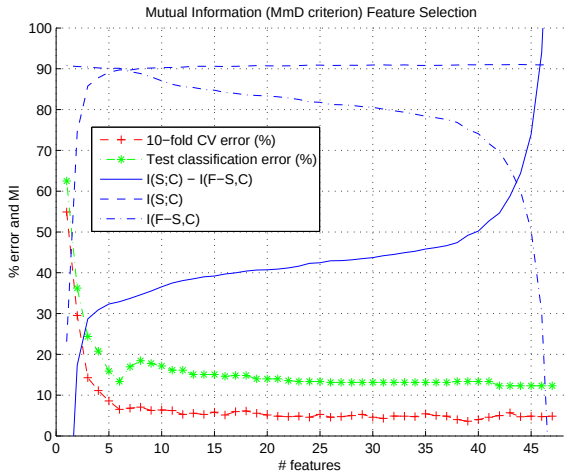

Figure 2.14: Maximum-Minimum Dependency Feature Selection criterion on image data with 48 features.

this is feasible insofar $\vec{C}$ is discrete ($\vec{C}$ consists of the class labelling). On the other hand, using Eq. 2.32 implies estimating the joint entropy. In the following experiments we used Eq. 2.31 because it is faster, due to the complexity of the entropy estimator, which depends on the number of samples as shown in the following section.

2.5. Entropy estimation from graphs

Entropy is a basic concept in information theory [Cover and Thomas, 1991b]. It is a concept related to predictability with several possible interpretations. One of them is that entropy measures the amount of information that an event provides. For example, a very unusual event is more informative than a very probable and frequent event. A related interpretation is that entropy measures the uncertainty in the outcome of an event. In this sense, one very common event provides less entropy than many different but equiprobable events. Another interpretation is that entropy measures the dispersion in the probability distribution. Thus, an image with many different colours has a more disperse (and entropic) histogram than an image with a few colours, or with some limited range of colours. Some other statistical measures could be useful for quantifying the dispersion, like kurtosis, which measures the

"peakedness" of a probability distribution.

2.5.1. Shannon and Rényi entropies

There are many different entropy definitions. The well known Shannon entropy of a discrete variable $\vec{Y}$ with a set of values $y_1, ..., y_N$, is defined as:

$$\begin{aligned} H(\vec{Y}) &= -E_y[\log(p(\vec{Y}))] \\ &= -\sum_{i=1}^{N} p(\vec{Y} = y_i) \log p(\vec{Y} = y_i), \end{aligned} \tag{2.35}$$

which can be extended to the continuous case for a probability density function (pdf) f:

$$H(f) = -\int_{-\infty}^{\infty} f(z) \log f(z) dz. \tag{2.36}$$

A generalization of the Shannon entropy is the Rényi's α-entropy defined as:

$$H_\alpha(\vec{Y}) = \frac{1}{1-\alpha} \log \sum_{i=1}^{n} y_j^\alpha, \tag{2.37}$$

and for the continuous case:

$$H_\alpha(f) = \frac{1}{1-\alpha} \ln \int_z f^\alpha(z) dz, \tag{2.38}$$

and in both cases the domain of the order α is:

$$\alpha \in (0, 1) \cup (1, \infty).$$

The Rényi entropy tends to the Shannon entropy when its order α tends to 1. However, for $\alpha = 1$ the latter definition has a discontinuity because there is a division by 0. This discontinuity is very significant. It can be shown, analytically and experimentally, that as α approaches 1, H_α tends to the value of the Shannon entropy. The analytical proof of the limit

$$\lim_{\alpha \to 1} H_\alpha(f) = H(f) \tag{2.39}$$

is straightforward by using the l'Hôpital's rule. Let $f(z)$ be a pdf of z. Its Rényi entropy, in the limit $\alpha \to 1$ is:

$$\lim_{\alpha \to 1} H_\alpha(f) = \lim_{\alpha \to 1} \frac{\log \int_z f^\alpha(z) dz}{1-\alpha}$$

In $\alpha = 1$ we have that $\log \int_z f^1(z)dz = \log 1 = 0$ (note that $f(z)$ is a pdf, then its integral over z is 1). This, divided by $1-\alpha = 0$ is an indetermination of the type $\frac{0}{0}$. By the l'Hôpital's rule we have that if

$$\lim_{x\to c} g(x) = \lim_{x\to c} h(x) = 0,$$

then

$$\lim_{x\to c} \frac{g(x)}{h(x)} = \lim_{x\to c} \frac{g'(x)}{h'(x)}.$$

Substituting the expression of the limit of the Rényi entropy:

$$\lim_{\alpha\to 1} \frac{\log \int_z f^\alpha(z)dz}{1-\alpha} = \lim_{\alpha\to 1} \frac{\frac{\partial}{\partial\alpha}(\log \int_z f^\alpha(z)dz)}{\frac{\partial}{\partial\alpha}(1-\alpha)}$$

The derivative of the divisor is $\frac{\partial}{\partial\alpha}(1-\alpha) = -1$, and the derivative of the dividend is:

$$\begin{aligned}\frac{\partial}{\partial\alpha}\left(\log \int_z f^\alpha(z)dz\right) &= \frac{1}{\int_z f^\alpha(z)dz}\frac{\partial \int_z f^\alpha(z)dz}{\partial\alpha} \\ &= \frac{1}{\int_z f^\alpha(z)dz}\int_z f^\alpha(z)\log f(z)dz.\end{aligned}$$

The first term of this expression goes to 1 in the limit because $f(z)$ is a pdf:

$$\begin{aligned}\lim_{\alpha\to 1} H_\alpha(f) &= \lim_{\alpha\to 1} -\frac{\int_z f^\alpha(z)\log f(z)dz}{\int_z f^\alpha(z)dz} \\ &= -\frac{\int_z f^1(z)\log f(z)dz}{1} \\ &= -\int_z f(z)\log f(z)dz \equiv H(f)\end{aligned}$$

The discontinuity at $\alpha = 1$ also marks a change from concave to convex in the α-entropy function. In Figure 2.15 it can be seen that for the interval $\alpha \in (0,1)$, the entropy function H_α is concave, while for the interval $\alpha \in (1,\infty)$, it is neither concave, nor convex. Also, for the first interval, it is smaller or equal to the Shannon entropy, $H_\alpha(\vec{P}) \geq H(\vec{P})$, $\forall\alpha \in (0,1)$, given that H_α is a non-increasing function of α.

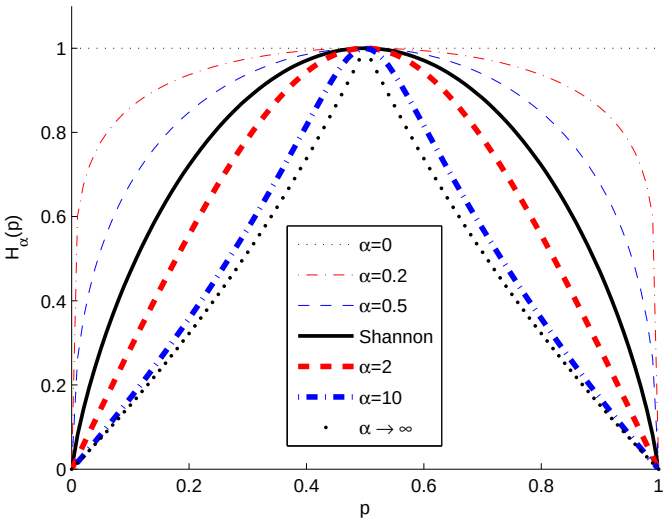

Figure 2.15: Rényi and Shannon entropies of a Bernoulli distribution $\vec{P} = (p, 1-p)$.

2.5.2. Bypass entropy estimation

Entropy estimation is critical in IT-based pattern recognition problems. The estimation of the Shannon entropy of a probability density given a set of samples has been widely studied in the past [Paninski, 2003, Viola and Wells-III, 1995, Viola et al., 1996, Hyvarinen and Oja, 2000, Wolpert and Wolf, 1995]. Entropy estimators can be divided in two categories: "plug-in" and "non plug-in". Plug-in methods [Beirlant et al., 1996] first estimate the density function, for example, the construction of a histogram, or the Parzen's windows method. The non plug-in methods [Hero and Michel, 2002], on the contrary, estimate entropy directly from a set of samples, bypassing the estimation of a density. The latter are also referred to as "bypass entropy estimation".

With the plug-in methods a serious problem arises when de number of the dimensions of the data is high. Due to the *curse of dimensionality*, it is not possible to obtain a good estimate the underlying density of a set of samples when there are more than two or three dimensions. On the one hand, the number of samples needed to define the density increases exponentially. On the other hand, the density estimation methods have to be tuned depending not only on the number of dimensions, but also on

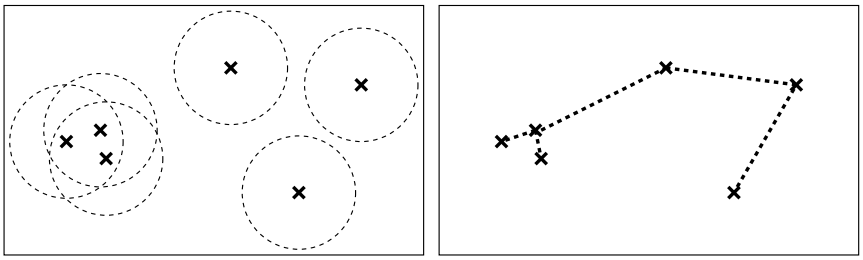

Figure 2.16: Estimation of a distribution of some samples, where the window width (Left) is not enough for capturing the distribution of some samples with respect to the rest. On the contrary, the minimal spanning tree (Right) of these samples connects them no matter how far from each other.

the data. The Parzen's windows approach, for example, is a widely used method for estimating pdfs for a finite set of patterns. It has a parameter which sets the width of the (usually Gaussian) kernels used in the method. An inadequate value of this parameter affects the resulting estimation. In [Viola and Wells-III, 1997] Viola and Wells propose a method for adjusting the widths of the kernels by iteratively maximizing the likelihood. However, the method does not perform well with high-dimensional patterns.

Among the bypass entropy estimation methods, the most commonly used are those based on entropic graphs and nearest neighbors graphs. These methods are capable of estimating entropy despite a high number of dimensions of the data space. Entropic and neighbors graphs are based on distances between the data patterns; once calculated the distances, no matter how high-dimensional the space is, the problem is simplified to distance measures. Thus, the computational complexity does not depend on the dimensionality, but on the number of patterns. On the other hand, in the Parzen windows approach there is a problem with data sparseness. Gaussian kernels, for example, though being infinite, do not consider those samples which lie at a distance larger than three standard deviatons, because their weight is close to 0. This effect (Figure 2.16) is less harmful when using entropic graphs for direct entropy estimation.

The most used graphs in the state of the art of entropy estimation are the minimal spanning trees and the k-nearest neighbour graphs. In both of them the vertices correspond to patterns in the feature space, and the length of the edges which connect those vertices, are the distances between patterns. A spanning tree graph is an undirected graph which connects all the vertices without cycles. The minimal spanning tree is the one which has minimum total sum of its edges. Entropy can be estimated from the

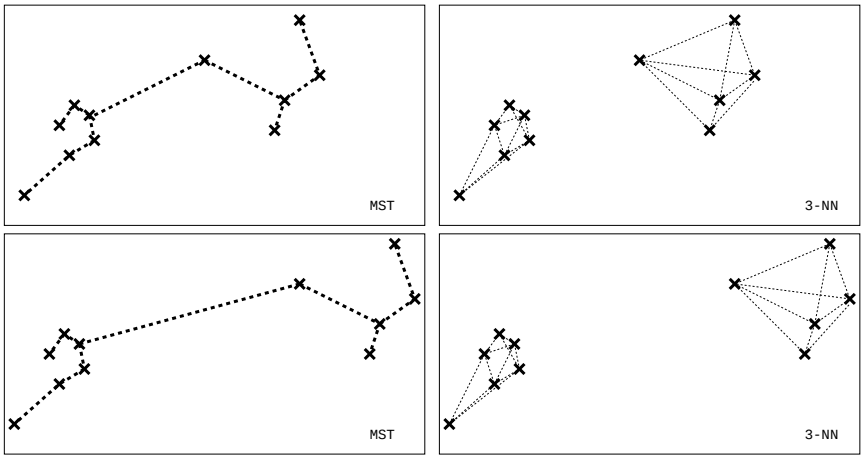

Figure 2.17: Toy example of entropic graphs. Left: minimal spanning tree. Right: k-nearest neighbour graph with $k = 3$. Top: bimodal distribution. Bottom: bimodal distribution with larger intermodal separation. Note that for this k, the k-NN graph remains the same.

weighted sum of those edges, as explained in the next section. The k-nearest neighbour graph is also undirected, however, it is not necessarily connected, and it is not acyclic. Its vertices are connected to the k nearest vertices. Thus, an important difference arises between the estimation based on minimal spanning trees and the based on k-nearest neighbour one. If a multimodal distribution has its modes separated enough, the nearest neighbour graph does not connect them with any edge and the distance between both modes is not taken into account (Figure 2.17-right). Contrarily, the minimal spanning tree always connects the different modes of a distribution (Figure 2.17-left).

Another recent but promising bypass entropy estimation method is the k-d partitioning by Stowell and Plumbley [Stowell and Plumbley, 2009]. It is a non-parametric method performs a recursive rectilinear partitioning of the space (Figure 2.18). The partitioning is adaptive and its $\Theta(N \log N)$ computational complexity also depends on the number of samples N, as happens with the estimation based on graphs. In each branch the recursivity stops depending on a heuristic which performs a test for uniformity of the density of the samples. The entropy is estimated from the volumes of the partitions:

$$\hat{H} = \sum_{j=1}^{m} \frac{n_j}{N} \log \left(\frac{N}{n_j} \mu(A_j) \right), \tag{2.40}$$

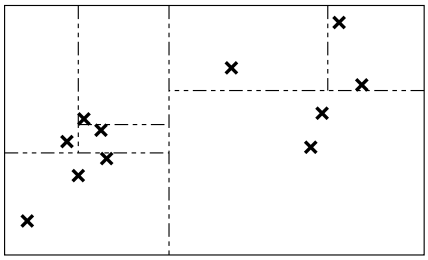

Figure 2.18: Toy example of k-d partitioning entropy estimation. A heuristical density uniformity test decides when to stop dividing the space in each branch. Entropy depends on the volumes of the partition.

for m disjoint partitions A_j, $j = 1, \ldots, m$ whose union is the whole considered space. (The lower an upper limits of each dimension have to be established so that the partitions are finite.) N is the number of samples, $\mu(A_j)$ are the D-dimensional volumes of the partitions and n_j are their respective numbers of samples.

2.5.3. Rényi's entropy estimation

The Rényi entropy (Eq. 2.38) of a set of samples can be estimated from the length of their minimal spanning tree (MST) in a quite straightforward way. This method, based on entropic spanning graphs [Hero and Michel, 2002], belongs to the non plug-in (or bypass) methods of entropy estimation.

The MST graphs have been used for testing the randomness of a set of points (see Figure 2.19). In [Hero and Michel, 1999] it was showed that in a d-dimensional feature space, with $d \geq 2$, the α-entropy estimator

$$H_\alpha(X_n) = \frac{d}{\gamma}\left[\ln \frac{L_\gamma(X_n)}{n^\alpha} - \ln \beta_{L_\gamma,d}\right] \tag{2.41}$$

is asymptotically unbiased and consistent with the PDF of the samples. Here, the function $L_\gamma(X_n)$ is the length of the MST, and γ depends on the order α and on the dimensionality: $\alpha = (d-\gamma)/d$. The bias correction $\beta_{L_\gamma,d}$ depends on the graph minimization criterion, but it is independent of the PDF. There are some approximations which bound the bias by: (i) Monte Carlo simulation of uniform random samples on unit cube $[0,1]^d$; (ii) approximation for large d: $(\gamma/2)\ln(d/(2\pi e))$ in [Bertsimas and Ryzin, 1990].

Regarding the length $L_\gamma(X_n)$ of the MST, it is defined as the length of the acyclic spanning graph with minimal length of the sum of the weights

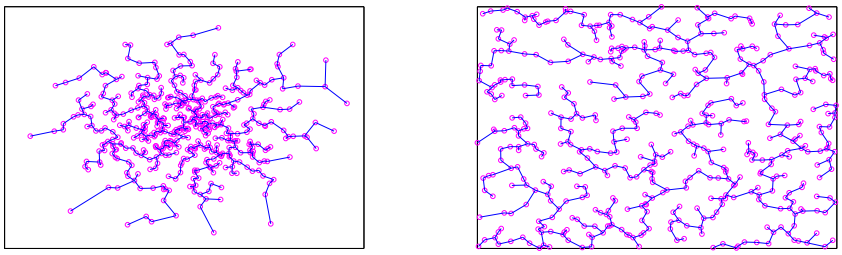

Figure 2.19: Minimal spanning trees of samples with a Gaussian distribution (top) and samples with a uniform distribution (bottom). The length L_γ of the first MST is visibly shorter than the length of the second MST.

(in this case the weights are defined as $\mid e \mid^\gamma$) of its edges $\{e\}$:

$$L_\gamma^{MST}(X_n) = \min_{M(X_n)} \sum_{e \in M(X_n)} \mid e \mid^\gamma, \tag{2.42}$$

where $\gamma \in (0, d)$. Here, $M(X_n)$ denotes the possible sets of edges of a spanning tree graph, where $X_n = \{x_1, ..., x_n\}$ is the set of vertices which are connected by the edges $\{e\}$. The weight of each edge $\{e\}$ is the distance between its vertices, powered the γ parameter: $\mid e \mid^\gamma$. There are several algorithms for building a MST, for example, the Prim's MST algorithm has a straightforward implementation.

2.5.4. Shannon from Rényi's entropy estimation

Sec. 2.5.1 presented the proof that Rényi's α-entropy tends to the Shannon entropy when $\alpha \to 1$:

$$\lim_{\alpha \to 1} H_\alpha(f) = H(f). \tag{2.43}$$

Therefore, in order to obtain a Shannon entropy approximation from α-entropy, α must have a value close to 1 but not equal to 1. It is convenient a value which is strictly less than 1, as for $\alpha > 1$ the Rényi entropy is no more concave, as shows Figure 2.15. The problem is which value close to 1 is the optimal, given a set of samples.

The experiments presented in [nalver et al., 2009] show that it is possible to model H_α as a function of α, independently on the on the size and nature of the data. The function is monotonic decreasing. For any point of this function, a tangent straight line $y = mx + b$ can be calculated. This

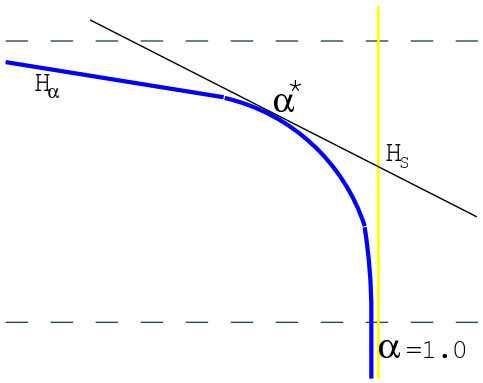

Figure 2.20: The line tangent to H_α in the point α^*, gives the Shannon entropy approximated value at $\alpha = 1$.

tangent is a continuous function and can give us a value at its intersection with $\alpha = 1$, as shown in Figure 2.20. Only one of all the possible tangent lines is the one which gives us a correct estimation of the Shannon entropy at $\alpha = 1$; let us say that this line is tangent to the function at some point α^*. Then, by knowing the correct α^*, one can obtain the Shannon entropy estimation. As H_α is a monotonous decreasing function, the α^* value can be estimated by means of a dichotomic search between two well separated values of α, for a constant number of samples and dimensions.

In [nalver et al., 2009] it has been experimentally verified that α^* is almost constant for diagonal covariance matrices with variance greater than 0.5. The optimal α^* depends on the number of dimensions D and samples N. This function is monotonous decreasing and can be modelled as:

$$\alpha^* = 1 - \frac{a + b \cdot e^{cD}}{N}. \tag{2.44}$$

Its values a, b and c have to be experimentally estimated. Peñalver et al. calibrated them for a set of 1000 distributions with random $2 \leq d \leq 5$ and number of samples [nalver et al., 2009]. The resulting function is Eq. 2.44 with values $a = 1.271$, $b = 1.3912$ and $c = -0.2488$. A limitation of this approach to Shannon entropy estimation is the need to adjust these values depending on the dimensionality and the nature of the data.

2.5.5. k-nearest neighbour entropy estimation

The k-nearest neighbour methods for entropy estimation [Kozachenko and Leonenko, 1987, Wang et al., 2006, Kraskov et al., 2004]

are another bypass entropy estimation approach which has a low computational complexity and is asymptotically consistent. Provided that these methods are based on distances, the k-NN approach is useful for high dimensionalities. Also, as happens with the MST-based methods, not having to estimate a density is an advantage in high dimensionalities.

In 1987 Kozachenko and Leonenko published the proofs [Kozachenko and Leonenko, 1987] for convergence and consistence of k-NN estimators of differential entropy. Recently Leonenko et al. published an extensive study [Leonenko et al., 2008] about Rényi and Tsallis entropy estimation, also considering the case of the limit of $\alpha \to 1$ for obtaining the Shannon entropy. Their construction relies on the integral

$$I_\alpha = E\{f^{\alpha-1}(\vec{X})\} = \int_{\mathbb{R}^d} f^\alpha(x)dx, \tag{2.45}$$

where f^α refers to the density of a set of n independent and identically distributed (i.i.d.) samples $\vec{X} = \{X_1, X_2, \ldots, X_N\}$. The latter integral is valid for $\alpha \neq 1$, however, the limits for $\alpha \to 1$ are also calculated in order to consider the Shannon entropy estimation.

2.5.5.1. Entropy-maximizing distributions

The α-entropy maximizing distributions are only defined for $0 < \alpha < 1$, where the entropy H_α is a concave function. The maximizing distributions are defined under some constraints. The uniform distribution maximizes α-entropy under the constraint that the distribution has a finite support. For distributions with a given covariance matrix the maximizing distribution is Student-t, if $d/(d+2) < \alpha < 1$, for any number of dimensions $d \geq 1$ [Vignat et al., 2004]. This is a generalization of the property that the normal distribution maximizes the Shannon entropy H (see proof in Appendix A). The entropy-maximizing properties are key in the derivation of the non-parametric entropy estimation.

A multi-dimensional Student distribution $T(\nu, \Sigma, \mu)$ with mean $\mu \in \mathbb{R}^d$, correlation matrix Σ, covariance matrix $C = \nu\Sigma/(\nu-2)$ and ν degrees of freedom has a probability density function (pdf):

$$f_\nu(x) = \frac{1}{(\nu\pi)^{d/2}} \times \frac{\Gamma(\frac{d+\nu}{2})}{\Gamma(\frac{\nu}{2})} \frac{1}{|\Sigma|^{\frac{1}{2}}[1 + (x-\mu)^\top[\nu\Sigma]^{-1}(x-\mu)]^{\frac{d+\nu}{2}}},$$

with $x \in \mathbb{R}^d$ and $\Gamma(\cdot)$ the Gamma function, a generalization of the factorial function $n! = \Gamma(n+1), n \in \mathbb{N}$ to the complex domain, defined as $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ for those z whose real part is positive.

As $\nu \to \infty$, the Rényi entropy of f_ν converges to the Rényi entropy of the normal distribution $\mathcal{N}(\mu, \Sigma)$ [Leonenko et al., 2008].

2.5.5.2. Rényi, Tsallis and Shannon entropy estimates

Leonenko et al. derive the following estimator of the integral (2.45). For $\alpha \neq 1$, the estimated I is:

$$\hat{I}_{N,k,\alpha} = \frac{1}{N} \sum_{i=1}^{n} (\zeta_{N,i,k})^{1-\alpha}, \tag{2.46}$$

with

$$\zeta_{N,i,k} = (N-1) C_k V_d (\rho_{k,N-1}^{(i)})^d, \tag{2.47}$$

where $\rho_{k,N-1}^{(i)}$ is the Euclidean distance from X_i to its k-th nearest neighbour from among the resting $N-1$ samples.

$$V_d = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} \tag{2.48}$$

is the volume of the unit ball $\mathcal{B}(0,1)$ in $\mathbb{R}^d$ and C_k is

$$C_k = \left[\frac{\Gamma(k)}{\Gamma(k+1-\alpha)} \right]^{\frac{1}{1-\alpha}}. \tag{2.49}$$

The estimator $\hat{I}_{N,k,\alpha}$ is asymptotically unbiased, which is to say that $E\hat{I}_{N,k,\alpha} \to I_q$ as $N \to \infty$, for $\alpha < 1$ if $I\alpha$ exists and for any $\alpha \in (1, k+1)$ if f is bounded. It is also consistent as $N \to \infty$, for $\alpha < 1$ if $I2\alpha - 1$ exists and, if f is bounded, for any $\alpha \in (1, (k+1)/2)$, $k \geq 2$.

Given these conditions, the estimated Rényi entropy H_α of f is

$$\hat{H}_{N,k,\alpha} = \frac{\log(\hat{I}_{N,k,\alpha})}{1-\alpha}, \quad \alpha \neq 1, \tag{2.50}$$

and the estimator of the Tsallis entropy $S_\alpha = \frac{1}{q-1}(1 - \int_x f^\alpha(x)dx)$ is

$$\hat{S}_{N,k,\alpha} = \frac{1 - \hat{I}_{N,k,\alpha}}{\alpha - 1}, \quad \alpha \neq 1, \tag{2.51}$$

and as $N \to \infty$, both estimators are asymptotically unbiased and consistent. The proofs are presented in [Leonenko et al., 2008].

The limit of the Tsallis entropy estimator (Eq. 2.51) as $\alpha \to 1$ gives the Shannon entropy H estimator:

$$\hat{H}_{N,k,1} = \frac{1}{N}\sum_{i=1}^{N} \log \xi_{N,i,k}, \tag{2.52}$$

with

$$\xi_{N,i,k} = (N-1)e^{-\Psi(k)} V_d (\rho_{k,N-1}^{(i)})^d, \tag{2.53}$$

where V_d is defined in Eq. 2.48 and $\Psi(k)$ is the digamma function which can be defined in terms of the $\Gamma(\cdot)$ function and its derivative, or for integers positive values k in terms of the Euler constant γ:

$$\Psi(z) = \frac{\Gamma'(z)}{\Gamma(z)} \tag{2.54}$$

$$\Psi(1) = -\gamma \simeq 0.5772, \tag{2.55}$$

$$\Psi(k) = -\gamma + A_{k-1} \tag{2.56}$$

with

$$A_0 = 0, \tag{2.57}$$

$$A_j = \sum_{i=1}^{j} \frac{1}{i}. \tag{2.58}$$

2.5.5.3. Kozachenko-Leonenko entropy estimation

A simpler way to understand the k-NN entropy estimation is to look at the Shannon entropy formula

$$H(X) = -\int f(x) \log f(x) dx, \tag{2.59}$$

as an average of $\log f(x)$, being $f(x)$ an existing pdf. The Shannon entropy estimator of Kozachenko-Leonenko [Kozachenko and Leonenko, 1987], whose publication was previous to [Leonenko et al., 2008], provides a slightly different form. Its estimates return almost the same values when the data has several dimensions. Only for a very low dimensionality a small difference between [Kozachenko and Leonenko, 1987] and [Leonenko et al., 2008] can be observed. This difference is not significant, for this reason in the experiments we only use [Leonenko et al., 2008].

The estimation of $\widehat{\log f(x)}$ would allow the estimation of

$$\hat{H}(X) = -N^{-1} \sum_{i=1}^{N} \widehat{\log f(x)} \tag{2.60}$$

For this purpose the probability distribution $P_k(\epsilon)$ for the distance between a sample x_i and its k-NN is considered. If a ball of diameter ϵ is centred at x_i and there is a point within distance $\epsilon/2$, then there are $k-1$ other points closer to x_i and $N-k-1$ points farther from it. The probability of this to happen is

$$P_k(\epsilon)d\epsilon = k\binom{N-1}{k} \frac{dp_i(\epsilon)}{d\epsilon} p_i^{k-1}(1-p_i)^{N-k-1} \tag{2.61}$$

being p_i the mass of the ϵ-ball and

$$p_i(\epsilon) = \int_{||\xi - x_i|| < \epsilon/2} f(\xi) d\xi. \tag{2.62}$$

The expectation of of $\log p_i(\epsilon)$ is

$$\begin{aligned} E(\log p_i) &= \int_0^\infty P_k(\epsilon) \log p_i(\epsilon) d\epsilon & (2.63)\\ &= k\binom{N-1}{k} \int_0^1 p^{k-1}(1-p)^{N-k-1} \log p \cdot dp & (2.64)\\ &= \psi(k) - \psi(N), & (2.65) \end{aligned}$$

where $\psi(\cdot)$ is the digamma function already defined in Eq. 2.54. If assumed that $f(x)$ is constant in the entire ϵ-ball, then the approximation

$$p_i(\epsilon) \approx \frac{V_d}{2^d} \epsilon^d \mu(x_i)$$

can be formulated. Here d is the dimension and V_d is the volume of the unit ball $\mathcal{B}(0,1)$, defined in Eq. 2.48. From the previous approximation and using the expectation of $\log p_i(\epsilon)$, the approximation of $\log f(\epsilon)$ is

$$\log f(\epsilon) \approx \psi(k) - \psi(N) - dE(\log \epsilon) - \log \frac{V_d}{2^d}, \tag{2.66}$$

and finally,

$$\hat{H}(X) = -\psi(k) + \psi(N) + \log \frac{V_d}{2^d} + \frac{d}{N} \sum_{i=1}^{N} \log \epsilon_i \tag{2.67}$$

is the estimation of $H(X)$, where $\epsilon_i = 2||x_i - x_j||$ is twice the distance between the sample x_i and its k-NN x_j. It is suggested that the error for Gaussian and uniform distributions is $\sim k/N$ or $\sim k/N \log(N/k)$.

2.5.6. Comparison

In this section we present an experimental comparative of the Kozachenko-Leonenko k-NN entropy estimator and the Stowell-Plumbley k-d partitioning entropy estimator. The estimator based on minimal spanning trees is not included in the experiment due to its requirement to tune the optimal α^* for different ranges of dimensions.

For comparing the estimators, a number of different distributions can be generated. In the following experiments the entropy estimations are performed on data following Gaussian (normal) distribution. The Gaussian distribution is the one which has maximum entropy relative to all probability distributions which, having finite mean and finite variance, cover the entire real line (or real space of any dimension, for multivariate distributions). A multivariate Gaussian distribution is described by the pdf:

$$f_{\mathcal{N}}(x_1, \cdots, x_N) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right), \quad (2.68)$$

where $|\Sigma|$ is the determinant of the covariance matrix Σ, which is a non-singular $d \times d$ matrix. The Gaussian is a useful distribution for entropy estimation experiments because it has a theoretical entropy value, which depends on the number of dimensions d:

$$H(f_{\mathcal{N}}) = \frac{1}{2}\log\left((2\pi e)^d|\Sigma|\right). \quad (2.69)$$

The data generated for the experiments was generated with a covariance matrix with ones in its diagonal, that is, the identity matrix, $\Sigma = I_{d\times d}$. This means that the generating distribution has the same variance in all the d dimensions. However, the generated data does not have exactly the same variance because infinite data would be necessary to perfectly define the actual distribution. If the data samples are very sparse in their d-dimensional space, their actual covariance matrix Σ could be very different from I. In fact, as the number of dimensions d increases, the number of samples N necessary to define well the distribution, increases exponentially, it is the *curse of dimensionality*. That is why in some figures we present both the theoretical value (Eq. 2.69) for the covariance $\Sigma = I$, and the theoretical value for the real Σ calculated from the data points, after their generation.

This sparseness effect can be observed in Figure 2.21, where the theoretical entropy value for $\Sigma = I$ is a dotted line and the theoretical entropy value for the real Σ is plotted in continuous black line. For a low number of

samples these values differ much more than for a high number of samples. The number of samples varies along the horizontal axis.

Figure 2.21 also represents the estimation results of the k-NN estimator in blue, and the k-d partitioning estimator in gray. Both estimators become more consistent as the number of samples increases. Left and the right columns correspond to the same plots, with fixed scales on the left, and zoomed-in vertical (entropy) scale on the right. The four rows correspond to four different dimensionalities: 1, 2, 3 and 4 dimensions. Note that, for a fixed number of samples, the estimation of both estimators becomes worse for an increasing number of dimensions. However, it can be seen that the data spasedness effect causes the k-d partition estimator to diverge much faster than the k-NN estimator.

A better analysis of the dimensionality effect on the entropy difference between the distributions which were used for generating data and both estimators can be seen in Figure 2.22. In this figure the entropy is measured for data consisting of the fixed number of 100 samples. These samples were distributed following a Gaussian distribution in spaces with different dimensionality, from 1 dimension to 200 dimensions. The discontinuous black line represents the theoretical entropy of the generating distribution. The four smaller plots correspond to the same results of the top figure, zoomed in for a better examination.

In the same figure (Figure 2.22) we also represent several different k values for the k-NN estimator. (Note that the k in k-d partition is not a parameter, it just belongs to the name we use for referring to the estimator of Stowell and Plumbley.) We use it to show that different k values offer quite a similar entropy estimation. In all the experiments we set $k = 5$ which has experimentally shown to be an appropriate value. However, the rest of the k values offer a similar results. In these plots, several different k values are represented in blue line, and only the $k = 5$ line is represented in black, in order to see that this value is consistent with the rest of k values. In the last plot of the figure it can be seen that the difference between them is very small.

From Figure 2.22, a conclusion can only be drawn for those dimensions in which 100 samples are still enough for defining the distribution, that is, 1 and 2 dimensions. In 3 dimensions 100 samples are a small number for defining well a Gaussian distribution. In the zoomed-in plots we can see that the estimations based on k-d partitions (gray line) keep close to the theoretical entropy (black discontinuous line) for 1 and 2 dimensions, but not for 3 dimensions and more. The k-NN estimation keeps close to

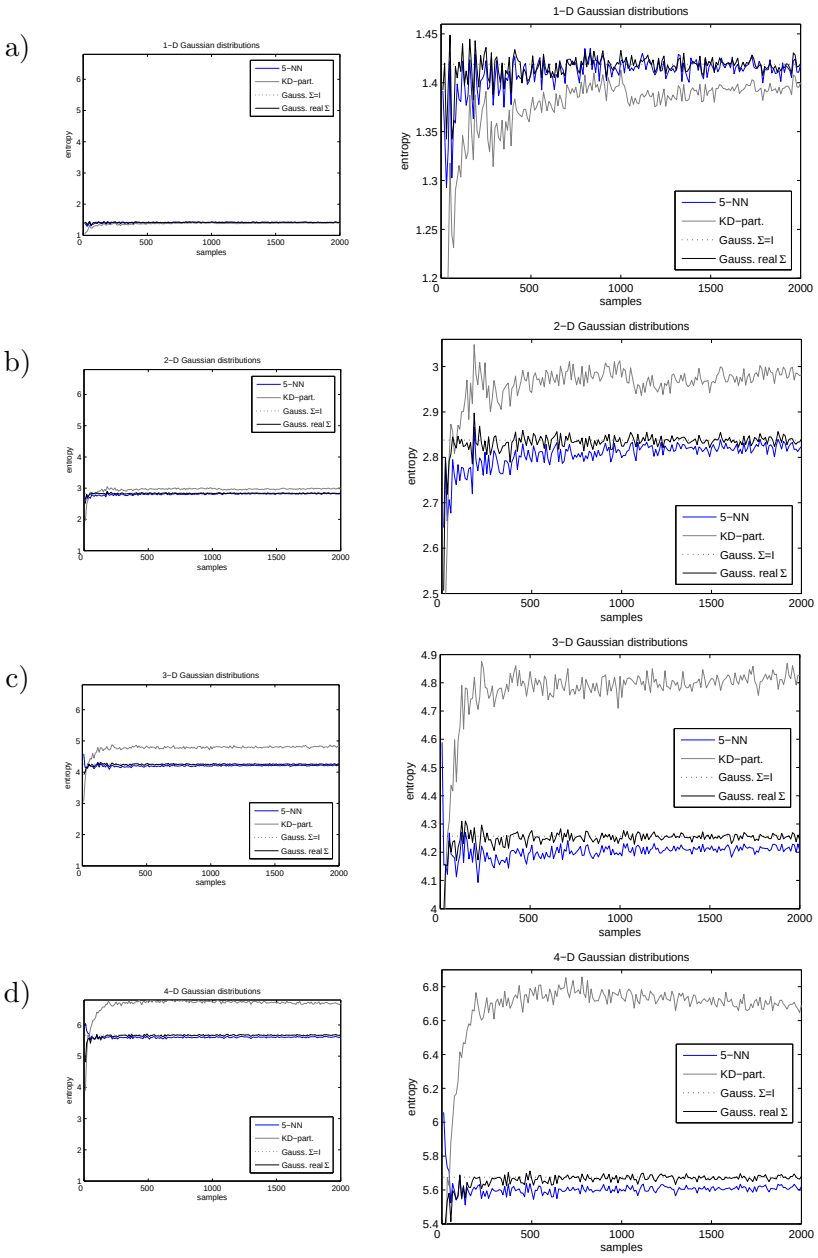

Figure 2.21: Comparison of the estimated entropy by the k-NN estimator and the k-d partitioning estimator. Both columns plot the same data, with fixed scales on the left, and zoomed-in entropy scale on the right. The distributions are Gaussian, a) 1-D, b) 2-D, c) 3-D, d) 4-D.

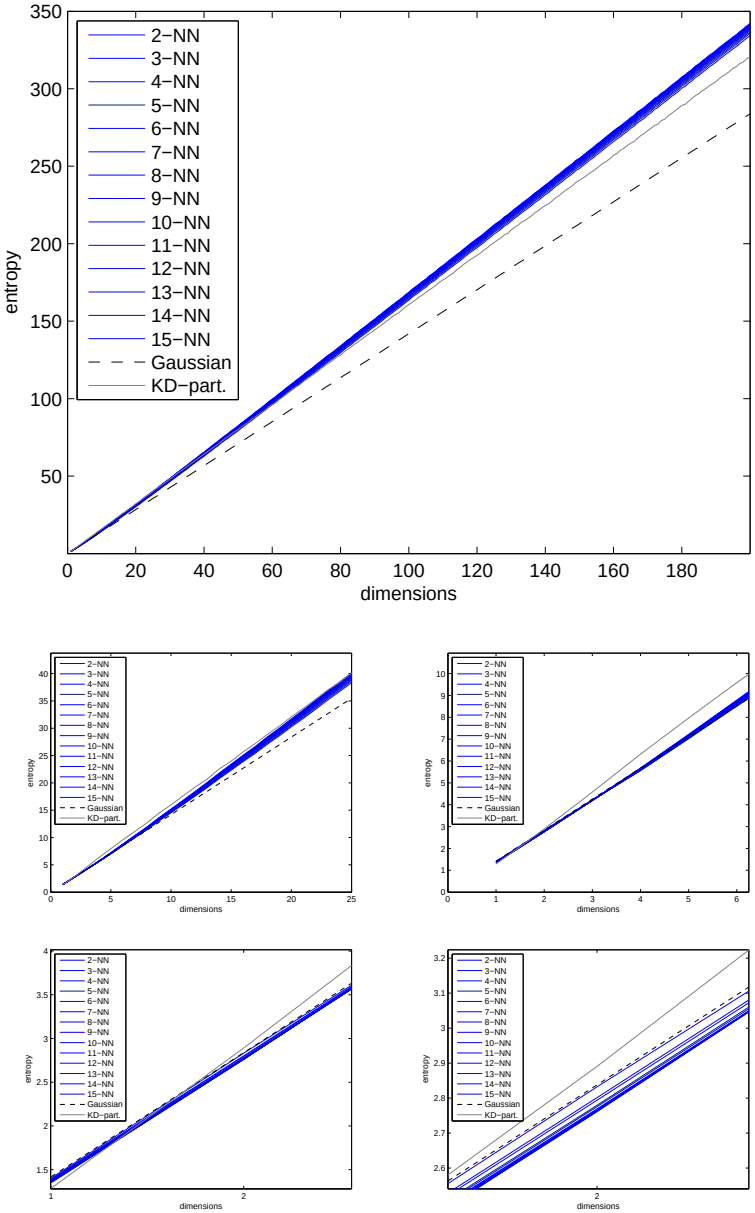

Figure 2.22: Comparison of the estimated entropy by the k-NN estimator and the k-d partitioning estimator. All the plots represent the same results, with different scales. The data for the experiment consists of Gaussian distributions for 100 samples, and different number of dimensions (represented in the horizontal axis).

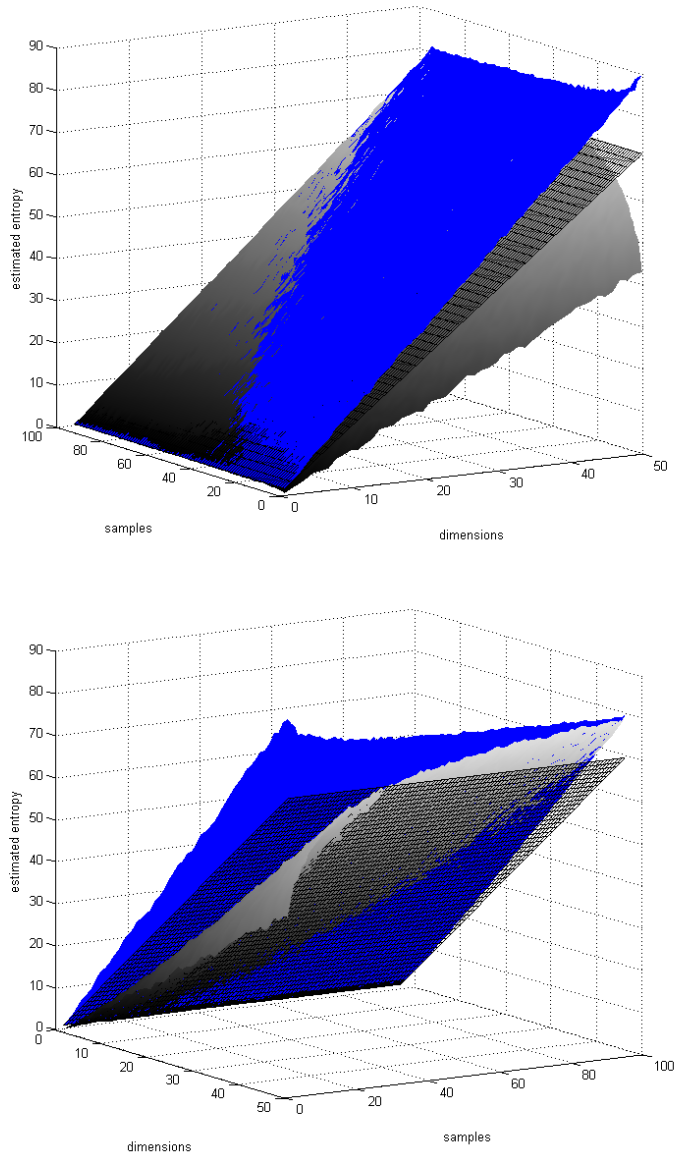

Figure 2.23: Comparison of the estimated entropy of Gaussian distributions by the k-NN estimator and the k-d partitioning estimator. The vertical axis represents the entropy, while the parameters of the surface are the number of samples and the number of dimensions. k-NN estimator (blue surface); k-d partitions estimator (gray surface); theoretical Gaussian entropy (black grid). Front view (top) and rear view (bottom).

the theoretical for serveral dimensions more and its estimations differ when there are 9 or more dimensions. After that, both estimators present a slightly changing behaviour, but no conclusion can be drawn, as 100 samples in such a high number of dimensions are very sparse for representing the distribution.

All these observations are valid only for small sampled Gaussian distributions (100 samples). If we exponentially increase the number of samples the plots would present a similar behaviour, but there is an offset in the dimensions axis. In Figure 2.23 we represent a surface plot in which the vertical axis, again, represents the entropy, while the parameters of the surface are the number of samples and the number of dimensions. In the front view of the plot it can be seen that Kozachenko-Leonenko's k-NN estimator (blue surface) keeps close to the theoretical entropy value (black grid) only for the first 8 dimensions. It can also be seen that the Stowell-Plumbley's k-d partition estimator is far from the correct values unless there is a sufficient number of samples. In [Stowell and Plumbley, 2009] they point out that a minimum of $N \geq 2^d$ samples are needed in relation with the dimensionality (d) in order to produce a reasonable estimate. This effect is observed in the gray surface, whose curvature with respect to the samples changes depending on the dimensionality.

The uniform distribution is another entropy maximizing distribution. It is the distribution of maximum entropy from among those distributions whose samples have a finite range support, that is, a minimum and maximum in each dimension. The entropy of the uniform distribution depends on the number of dimensions d and on the size of the range of each dimensions, which is given by its upper and lower limits, $l_{d,2}$ and $l_{d,1}$, respectively.

$$H(f_{\mathcal{U}}) = \log \prod_{i=1}^{d} (l_{i,2} - l_{i,1}). \tag{2.70}$$

The k-d partitioning based estimator of Stowell-Plumbley is more adequate for the estimation of this kind of distributions, as it parts a regions of the space which is given by the bounding box of the data, that is, the limits of the range. In Figure 2.24 an experiment with uniform distributions is represented. Their limits are $(l_{d,1}, l_{d,2}) = (0, 1)$ in any number d of dimensions. Then, according to Eq. 2.70 the entropy of the distributions is 0 for any dimension. In Figure 2.24 the plots represent how the estimation of the entropy converges to the real entropy value (0 in this case), as the number of samples defining the distribution increases. The experiment is presented in four different dimensionalities and, as expected, in higher dimensions the

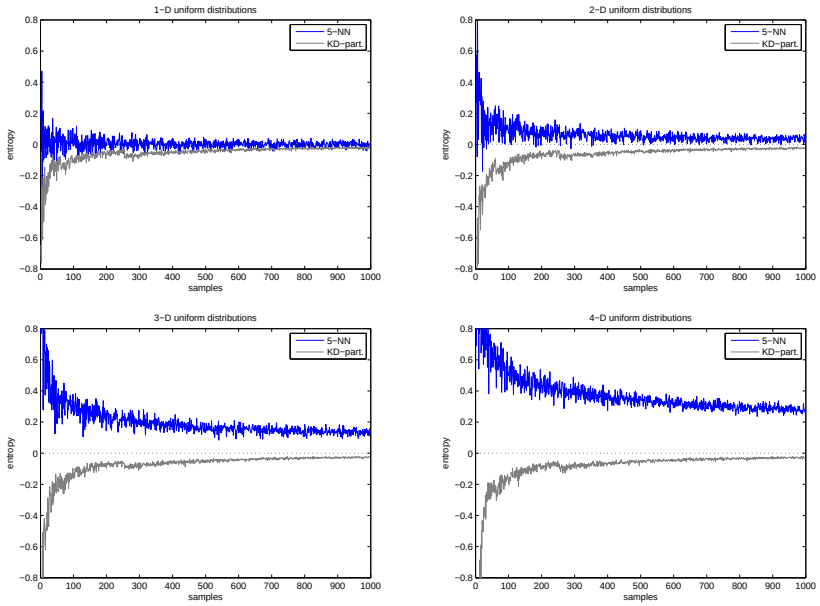

Figure 2.24: Comparison of the estimated entropy of uniform distributions by the k-NN estimator and the k-d partitioning estimator. The vertical axis represents the entropy, whose estimation becomes better as the number of samples (horizontal axis) increases. The theoretical entropy value of the uniform distributions generated is zero for any dimension. Top left: 1D, top right: 2D, bottom left: 3D, bottom right: 4D.

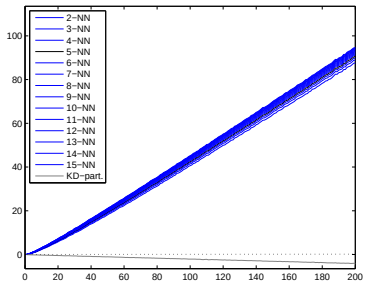

Figure 2.25: Comparison of the estimated entropy of uniform distributions by the k-NN estimator and the k-d partitioning estimator. The vertical axis represents the entropy, whose estimation becomes worse as the number of dimensions (horizontal axis) increases for a fixed number of samples, in this experiment, 100. The theoretical entropy value of the uniform distributions generated is zero for any dimension.

convergence needs a higher number of samples. For a better illustration of the dimensionality effect on the estimation Figure 2.25 represents the estimations of uniform distributions with a fixed number of samples (100), in different dimensions from 1 to 200. While the k-NN method estimates higher entropies as the number of dimensions increases, the k-d partitioning estimator keeps a good lower bound of the entropy.

In conclusion, the entropy estimators that have to be used in high-dimensonal problems are of the bypass kind, because the estimation of the distribution is not feasible. Among the estimators based on graphs, the k-NN based estimator of Kozachenko-Leonenko offers a good alternative to the MST based estimators. However, the k-NN estimator may fail to capture well enough the entropy if the different modes of a distribution are very separated. The k-NN entropy estimator is useful for distributions in $\mathbb{R}^d$, while the Stowell-Plumbley's k-d partitioning entropy estimator is useful for distributions with a finite support. The latter tends to underestimate entropy, while Kozachenko-Leonenko's tends to overestimate it.

2.6. Search order

2.6.1. Limitations of the Greedy Search

The term "greedy" refers to the kind of searches in which the decisions can not be undone. In many problems, the criterion which guides the search does not necessarily lead to the optimal solution and usually falls into a local maximum (minimum). This is the case of forward feature selection. In the previous subsections we presented different feature selection criteria. With the following toy problem we show an example of incorrect (or undesirable) feature selection.

Suppose we have a categorical data set. The values of categorical variables are labels and these labels have no order: the comparison of two categorical values can just tell whether they are the same of different. Note that if the data is not categorical but it is ordinal, regardless if it is discrete or continuous, then a histogram has to be built for the estimation of the distribution. For continuous data, a number of histogram bins has to be chosen necessarily, and for some discrete, but ordinal data, it is also convenient. For example, the distribution of the variable $x = \{1, 2, 1002, 1003, 100\}$ could be estimated by a histogram with 1003 bins (or more) where only five bins would have a value of 1. This kind of histogram is too sparse. A histogram with 10 bins offers a more compact representation, though less precise, and the distribution of x would look like $(\frac{2}{5}, \frac{1}{5}, 0, 0, 0, 0, 0, 0, 0, \frac{2}{5})$.

There also are entropy estimation methods which bypass the estimation of the probability distribution. These methods, however, are not suitable for categorical variables. For simplicity we present an example with categorical data, where the distribution of a variable $x = \{A, B, \Gamma, A, \Gamma\}$ is $Pr(x = A) = \frac{2}{3}$, $Pr(x = B) = \frac{1}{3}$, $Pr(x = \Gamma) = \frac{2}{3}$.

The data set of the toy-example contains 5 samples defined by 3 features, and classified in 2 classes.

	x_1	x_2	x_3	C
	A	Z	Θ	C_1
	B	Δ	Θ	C_1
	Γ	E	I	C_1
	A	E	I	C_2
	Γ	Z	I	C_2

The mutual information between each single feature x_i, $1 \leq i \leq 3$ and the class C is:

$$\begin{aligned} I(x_1, C) &= 0.1185 \\ I(x_2, C) &= 0.1185 \\ I(x_3, C) &= 0.2911 \end{aligned} \tag{2.71}$$

Therefore, both mRMR and MD criteria would decide to select x_3 first. For the next feature which could be either x_1 or x_2, mRMR would would have to calculate the redundancy of x_3 with each one of them:

$$\begin{aligned} I(x_1, C) - I(x_1, x_3) &= 0.1185 - 0.3958 = -0.2773 \\ I(x_1, C) - I(x_1, x_3) &= 0.1185 - 0.3958 = -0.2773 \end{aligned} \tag{2.72}$$

In this case both values are the same and it does not matter which one to select. The feature sets obtained by mRMR, in order, would be: $\{x_3\}$, $\{x_1, x_3\}$, $\{x_1, x_2, x_3\}$.

To decide the second feature (x_1 or x_2) with the MD criterion, the mutual information between each one of them with x_3, and the class C, has to be estimated. According to the definition of MI, in this discrete case, the formula is:

$$I(x_1, x_3; C) = \sum_{C} \sum_{x_3} \sum_{x_1} \left[p(x_1, x_3, C) \log \frac{p(x_1, x_3, C)}{p(x_1, x_3)p(C)} \right]$$

The joint probability $p(x_1, x_3, C)$ is calculated with a 3D histogram where the first dimension has the values of x_1, i.e., A, B, Γ, the second dimension has the values Θ, I, and the third dimension has the values C_1, C_2. The resulting MI values are:

$$\begin{aligned} I(x_1, x_3; C) &= 0.3958 \\ I(x_2, x_3; C) &= 0.3958 \end{aligned} \tag{2.73}$$

Then, MD would also select any of them, as first-order forward feature selection with MD and mRMR are equivalent. However, MD can show us that selecting x_3 in first place was not a good decision, given that the combination of x_1 and x_2 has much higher mutual information with the class:

$$I(x_1, x_2; C) = 0.6730 \tag{2.74}$$

Therefore in this case we should have used MD with a higher-order forward feature selection, or another search strategy (like backward feature selection or some non-greedy search). A second-order forward selection would have first yielded the set $\{x_1, x_2\}$. Note that the mutual information of all the features and C, does not outperform it:

$$I(x_1, x_2, x_3; C) = 0.6730, \tag{2.75}$$

and if two feature sets provide the same information about the class, the preferred is the one with less features: x_3 is not informative about the class, given x_1 and x_2.

The MmD criterion would have selected the features in the right order in this case, because it not only calculates the mutual information about the selected features, but it also calculates it for the non-selected features. Then, in the case of selecting x_3 and leaving unselected x_1 and x_2, MD would prefer not to leave together an unselected pair of features which jointly inform so much about the class. However, MmD faces the same problem in other general cases. Some feature selection criteria could be more suitable for one case or another. However,there is not a criterion which can avoid the local maxima when used in a greedy (forward or backward) feature selection. Greedy searches with a higher-order selection, or algorithms which allow both addition and deletion of features, can alleviate the local minima problem.

2.6.2. Greedy Backward Search

Even though greedy searches can fall into local maxima, it is possible to achieve the highest mutual information possible for a feature set, by means of a greedy backward search. However, the resulting features set which provides this maximum mutual information about the class, is usually suboptimal.

There are two kinds of features which can be discarded: irrelevant features and redundant features. If a feature is simply irrelevant to the class label, it can be removed from the feature set and this would have no impact on the mutual information between the rest of the features and the class. It is easy to see that removing other features from the set is not conditioned by the removal of the irrelevant one.

However, when a feature x_i is removed due to its redundancy given other features, it is not so intuitive if we can continue removing from the remaining features, as some subset of them made x_i redundant. By using the mutual information chain rule we can easily see the following. We remove a feature x_i from the set $\vec{F}_n$ with n features, because that feature provides no additional information about the class, given the rest of the features $\vec{F}_{n-1}$. Then, we remove another feature $x_{i'}$ because, again, it provides no information about the class given the subset $\vec{F}_{n-2}$. In this case the previously removed one, x_i, will not be necessary any more, even after the removal of $x_{i'}$. This process can continue until it is not possible to remove any feature because otherwise the mutual information would decrease. Let us illustrate it with the chain rule of mutual information:

$$I(\vec{F}; C) = I(x_1, \ldots, x_n; C) = \sum_{i=1}^{n} I(x_i; C | x_{i-1}, x_{i-2}, \ldots, x_1) \tag{2.76}$$

With this chain rule the mutual information of a multi-dimensional variable and the class is decomposed into a sum of conditional mutual information. For simplicity, let us see an example with 4 features:

$$\begin{aligned} I(x_1, x_2, x_3, x_4; C) \quad = \quad & I(x_1; C) + \\ & I(x_2; C | x_1) + \\ & I(x_3; C | x_1, x2) + \\ & I(x_4; C | x_1, x_2, x_3) \end{aligned}$$

If we decide to remove x_4, it is because it provides no information about C, given the rest of the features, that is: $I(x_4; C | x_1, x_2, x_3) = 0$. Once

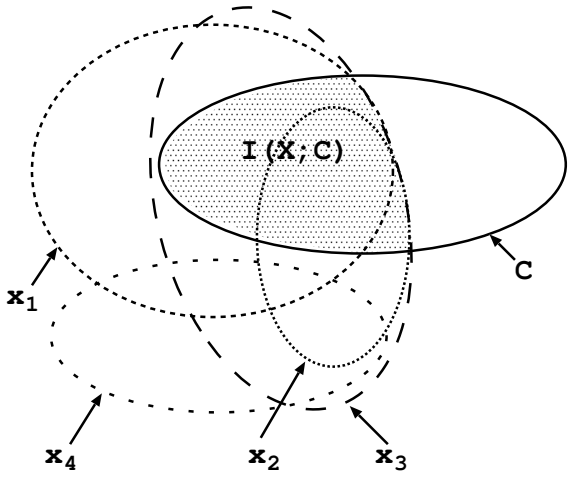

Figure 2.26: A Venn diagram representation of a simple feature selection problem where C represents the class information, and $X = \{x_1, x_2, x_3, x_4\}$ is the complete feature set. The coloured area represents all the mutual information between the features of X and the class information. The feature x_4 does not intersect this area, this means that it is irrelevant.

removed, it can be seen that x_4 does not appear in any other terms, so, x_3, for example, could be removed if $I(x_3; C|x_1, x2) = 0$, without worrying about the previous removal of x_4.

This backward elimination of features does not usually lead to the minimum feature set. In Figure 2.26 we have illustrated a sample feature selection problem with 4 features. The feature x_4 can be removed because $I(x_4; C|x_1, x_2, x_3) = 0$. Actually, this feature is not only irrelevant given the other features, but it is also irrelevant by itself, because $I(x_4; C) = 0$. The next feature which could be removed is either x_1, x_2, or x_3 because we have that $I(x_1; C|x_2, x_3) = 0$, $I(x_2; C|x_1, x_3) = 0$ and $I(x_3; C|x_1, x_2) = 0$. In such situation greedy searches take their way randomly. See Figure 2.27 to understand that, if x_3 is taken, the search falls into a local minimum, because neither x_1, nor x_2 can be removed, if we do not want to miss any mutual information with the class. However, if instead of removing x_3, one of the other two features is removed, the final set is $\{x_3\}$, which is the smallest possible one for this example.

The artificial data set "Corral" [John et al., 1994] illustrates well

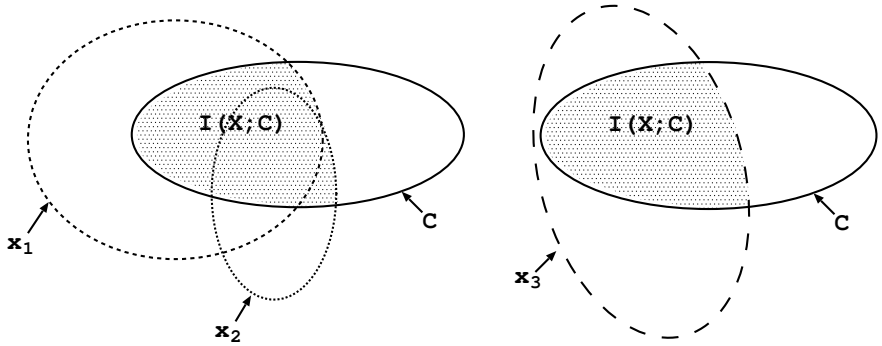

Figure 2.27: In the previous Figure 2.26, the features $\{x_1, x_2\}$ (together) do not provide any further class information than x_3 provides by itself, and vice versa: $I(x_1, x_2; C|x_3) = 0$ and $I(x_3; C|x_1, x_2) = 0$. Both feature sets, $\{x_1, x_2\}$ (left) and x_3 (right), provide the same information about the class as the full feature set.

the difference between forward and backward greedy searches with mutual information. In this data set there are 6 binary features, $\{x_1, x_2, x_3, x_4, x_5, x_6\}$. The class label is also binary and it is the result of the operation:

$$C = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$$

Therefore x_1, x_2, x_3, and x_4 fully determine the class label C. The feature x_5 is irrelevant, and x_6 is a feature highly (75%) correlated with the class label. Some samples could be the following:

x_1	x_2	x_3	x_4	x_5	x_6	C
0	1	1	0	1	0	0
0	1	1	1	1	1	1
1	0	0	0	0	1	0
1	0	1	1	0	1	1

Most feature selection approaches, and in particular those which perform a forward greedy search, first select the highly correlated feature, which is an incorrect decision. Contrarily, when evaluating the mutual information (the MD criterion) in a backward greedy search, the first features to discard are the irrelevant and the correlated ones. Then, only the four features defining the class label remain selected.

In practice, the mutual information estimations are not perfect; moreover, the train set usually does not contain enough information to perfectly define the distribution of the features. Then, rather than maintaining a zero decrease of the mutual information when discarding features, the objective is rather to keep it as high as possible, accepting small decreases.

To illustrate the impact of the search order, we present an experiment on real data. The data consists of spectral graph features, as explained in Section 5.3. The error plot (Figure 2.28-top) represents the 10-fold cross validation errors yielded by both feature selection orders: forward and backward. They were run on the same data set, and using the same evaluation criterion, i.e, the mutual information. Although there might be data sets for which the search order does not have such a big impact, in this case it does, and the minimum error yielded by the backward search order is 25%, while the minimum error achieved by the forward search order is 34%. Note that feature selection didn't work well (in terms of CV error) for the latter, because the minimum error is reached with a very large feature set, 534. Contrarily the backward process obtained the minimum error with only 88 features. This is due to important differences between the selected features in both cases. See Figure 2.28-bottom where the area plots represent the participation of each feature with respect to the others along the feature selection process: from 1 to 540 features. It can be observed that for very small feature sets, the forward strategy starts with bins of the same feature, while the backward approach yielded more equilibrated feature sets containing different features.

2.6.2.1. Bootstrapping

Bootstrap methods [Efron, 1981] resample the data in order to estimate the properties of an estimator, for example, its variance and mean. A number of resamples is drawn from the original data set, by random sampling with replacement. It is assumed that the original set of observations are independent and identically distributed. The bootstrap method is general-purpose and it is much simpler than analytical methods. It tends to be optimistic even though it is asymptotically consistent.

In Figure 2.29 we show the results of a bootstrapping experiment consisting of 25 bootstraps of a set of 225 samples. For the testing the errors there is a separate test set of 75 samples which is the same for all the bootstraps so that their evaluation criterion is the same and independent on the bootstraps. The resamplings consist of a random sampling with repetition of size 225, from the original train set of 225 samples. In practice

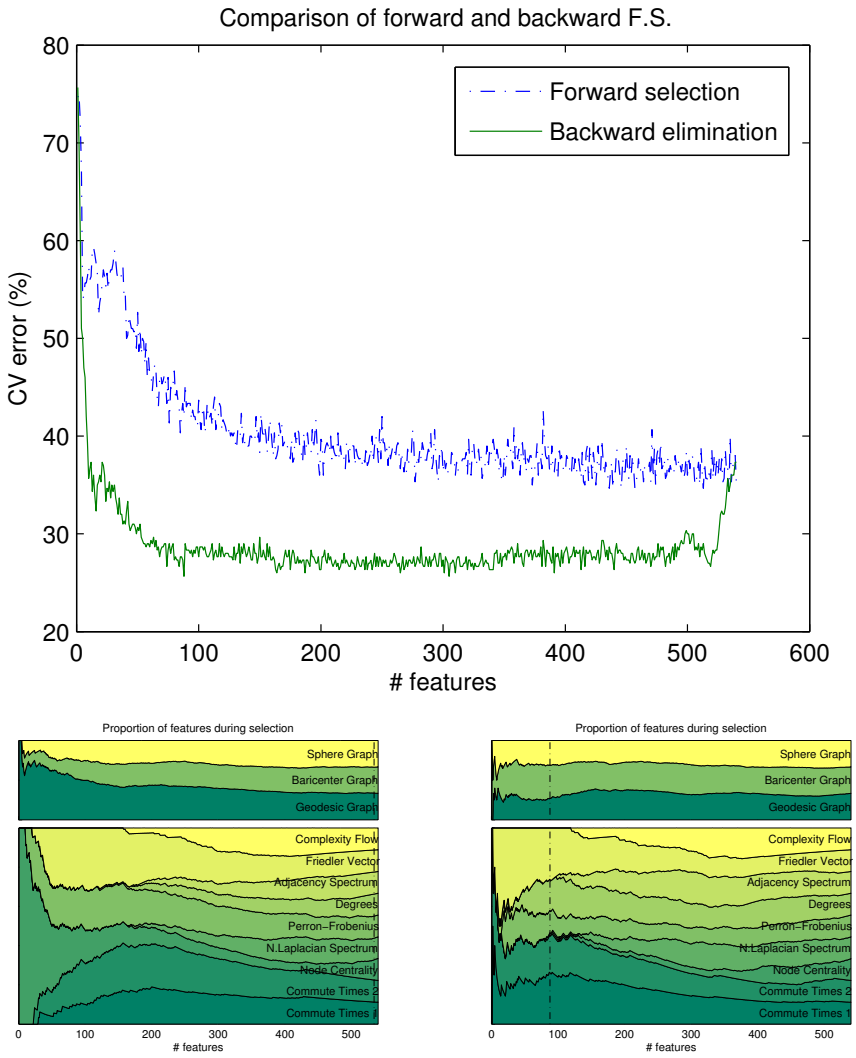

Figure 2.28: Comparison of forward selection and backward elimination. Top: 10-fold cross validation errors for all the feature set sizes produced by both search orders on the same data set (structural data, explained in Section 5.3).
Bottom: The participation of each type of feature during the process of feature selection. Left: forward feature selection. Right: backward feature elimination. These area plot represent, for each feature set size (X axis), the participation of each feature in the selected set. The order of the features on the Y axis is arbitrary; instead, their areas represent their presence in the feature set. Note that in the complete feature set (540 features) all features have the same participation because there are no more features to select.

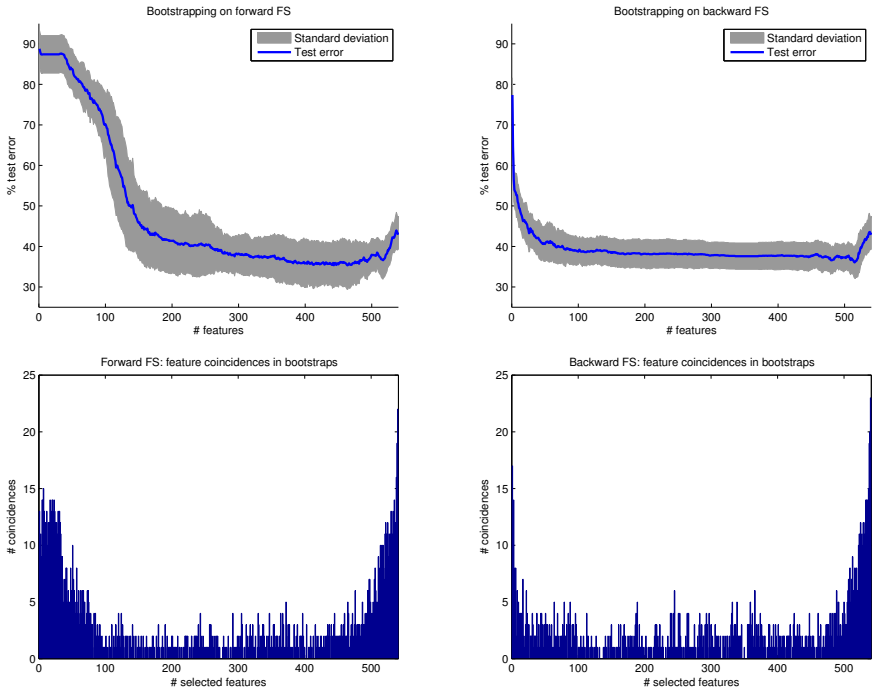

Figure 2.29: Bootstrapping experiments on forward (on the left) and backward (on the right) feature selection, with 25 bootstraps. Train and test samples are separate sets and the bootstraps are extracted (with repetition) from the train set. Top: mean test errors and their standard deviation.
Bottom: the number of features which are selected by several bootstraps in the same iteration of the feature selection process.

this means that in each bootstrap some of the 225 samples are missing, and some other are repeated.

With these bootstraps 25 feature selection processes are run and each one of them yields slightly different feature sets. These differences can be observed in Figure 2.29-bottom, where we represent the number of features which coincide among two or more bootstraps. The horizontal axis represents the iteration of the feature selection, in this case from 1 to 540 because there are 540 features. In forward feature selection (left part of Figure 2.29) the first iteration yields a feature set of size 1, the second, size 2, and so on. Contrarily, backward feature elimination (right part of Figure 2.29) yields its feature set of size 1 in the last iteration, while the first iteration yields a feature set of size 539 (the number of features minus one). The X axis of all the plots in this thesis are ordered by the size of the feature sets,

disregarding the search order of the algorithm. Figure 2.29-top represents the test errors and their standard deviation.

The test errors are with the 10-fold cross validation errors in Figure 2.28. The latter are lower as all the 300 samples are involved in the classification. Also, the forward and backward selection have similar CV error and test error curves. Apart from the lower error of the backward search, it is interesting to point out its smaller standard deviation (Figure 2.29-top). This means that the backward search yields more stable feature selection results than the forward search, which coincides with the idea that we theoretically analysed in the present section. However, it can be seen that the number of feature coincidences (Figure 2.29-bottom) is higher in the early stage of the forward selection processes.

Another interesting phenomenon is that in both cases, most of the feature coincidences occur in the firstly selected features and in those which are the last to be selected; in the case of backward search – the firstly discarded and the last to discard. In the middle stages of the process the features among bootstraps are quite different.

2.7. Conclusions

In this chapter we presented the main contribution of this work, which is the efficient use of mutual information for evaluating subsets in supervised feature selection. Although the idea of using mutual information as a feature selection criterion has been present in the literature for years, it has been used on separate features instead of evaluating entire feature sets. Thus, the interactions among features are not taken into account. Some approaches like Markov Blankets calculate k-th order interactions, however, higher order interactions are not evaluated still. The reason for not evaluating the mutual information on feature subsets is the difficulty of estimating it with the traditional parametric entropy estimation methods. For estimating the density of a distribution the number of samples has to be very high in relation to the number of features. This is usually not the case of the data sets with more than 3 or 4 features. The method of [Peng et al., 2005] estimates the multi-dimensional mutual information, however, they do it in an incremental way. This makes it mandatory to follow a greedy forward search order, as already explained in this chapter. Contrarily, the approach we presented can be used with any search order.

Our approach consists of estimating the mutual information based on entropy estimation methods which bypass the density estimation and esti-

mate it directly from the samples, based on the distances between them. We considered three of them: one based on minimal spanning trees, one based on k-nearest neighbors, and one based on k-d partitions. The first one is not suitable because it estimates Rényi's α-entropy. It is possible to approximate it to the Shannon entropy but there are some parameters which depend on the data. In this sense the methods based on k-d partitions and k-nn are more general. We analyzed the estimations of both methods for different number of dimensions and samples and concluded that Leonenko's k-nn entropy estimator yields better estimations.

Finally, we argue that, even though the greedy search is obviously limited by local minima, it is not the same to perform a greedy forward search or a greedy backward search, with the mutual-information-based criterion. We showed that from a theoretical point of view, backward greedy selection can preserve all the mutual information between the features and the class label, while safely discarding irrelevant features. In practice we just try to minimize the loss of mutual information when discarding a feature. Forward feature selection, on the other hand, is very prone to limitations due to the features selected in the first iterations. For instance, the decision to select the first features does not actually take into accound other features, as the set of selected features is of size 1. The second feature to select takes the previous into account, and so on. Thus the feature sets selected by a forward process are more arbitrary than those yielded by a backward process. We presented a bootstrap experiment that proves this effect and shows that backward greedy feature selection yields better small-size sets than the greedy one. Also, the standard deviation of the errors is larger for forward feature selection. This phenomenon agrees with our conclusion that the forward process produces more arbitrary feature sets in the first iterations, and that these first features highly condition the rest of the process. Thus, the backward process should be used whenever possible. In some situations it is not feasible because of the dimensionality of the original feature set. The reason for this is that backward elimination takes more time in its first iterations (that is, the first features to discard) while forward selection is faster in its first iterations.

Chapter 3

Application to image classification

In this chapter we present a feature selection application to a mobile robot using only vision. This is a low-cost approach which can be used for many different mobile robots, vehicles, and even people. Usually installing sonars, lasers, or other range sensors is more expensive than installing a video camera. GPS is inexpensive, however, it only works outdoors. Regarding indoor environments, there are many localization and navigation systems which are designed to work in a constrained environment with special landmarks installation.

The approaches we present in this work are oriented to work in both indoor and outdoor environments. The localization method [Bonev and Cazorla, 2006a] is able to learn from any kind of environment. Also a simple navigation method [Bonev et al., 2007a] is presented so that the robot can move autonomously across the environment. The motivation of this application is that vision provides much richer information than other kind of sensors, however, dealing with its information is a challenging task. In the following sections we present our approach and results.

3.1. Introduction

Mobile robots are becoming an essential part of our lives. They are present not only in industries and laboratories, but also in less controlled environments, such like museums, at home, and some mobile robots are even destined to navigate outdoors.

Many kinds of the mobile robots have wheels because these are cheaper and faster than legged robots. In this research we present experiments

with three robots, all of them are fully autonomous wheeled robots. An autonomous robot is characterized by its skill to understand an environment and perform the appropriate actions without any external information or guidance. This means that it needs the adequate sensors, which could be sonars, laser, cameras, contact sensors, to gather information from the world. It also needs a computer with software which processes all this information, and finally its actuators, which receive orders from the computer (or processor). Actuators are the wheels, legs, robotic arm, and all the accessories of the robot which are able to physically modify the world, or the state (position) of the robot.

In such a challenging task, where the world is dynamic, sensors return noisy information, and actuators are imperfect, there are two main Mobile Robotics' problems to consider: Localization and Navigation. Navigation [Bonev et al., 2007b, Lambrinos et al., 2000] is a necessary skill for any mobile robot. It consists of being able to move around an environment without colliding [Borenstein and Koren, 1989], in order to avoid harming people, other objects, or the robot itself. Even though in some navigation approaches localization is necessary, in many other it is not. However localization is fundamental when a robot is supposed to perform some particular task.

The Global Positioning System (GPS) could be used in many cases. However, it does not work indoors, unless some special installations are present in the building. On the other hand GPS provides coordinates, but does not provide any conceptual information such like "in a corridor" or "in front of a door". For these cases vision, sonars, or even lasers, report much richer information. There are different kinds of localization, depending on the task and depending on the environment. In this work we differentiate between fine localization and coarse localization. Fine localization is usually implemented with artificial landmarks [Buschka et al., 2000], or for small environments. For larger environments coarse localization is needed before fine-localizing the robot [Bonev et al., 2007d]. Coarse localization is also useful for topological navigation [Boada et al., 2004]. In topological navigation the robot does not need to be localized all the time, but only in some decisive places or moments. For example, it could follow a corridor until finding a door. There it would decide what to do, whether to enter or to go on. At such point the robot needs to know its localization, otherwise its decision could be wrong.

The localization approach adopted in this work is based on supervised classification. Feature selection turns to be a feasible technique for adapting

the system to a variety of environments. An example can be found in [Bonev et al., 2007d], where coarse localization is used to obtain the kind of environment (class) where the robot is located. Once the environment is known, the input image is compared (via graph matching) to a set of images of the training set. Without coarse localization it would be infeasible to perform graph matching, as the same graphs could be found in different environments.

On the other hand this approach is appropriate for real time classification, as the feature selection is an offline process. Then, applying the selected filters to a test image and classifying it, can take about 0.1 seconds on a standard PC, depending on the number of filters and the classifier.

3.2. Experimental setup

3.2.1. Hardware

The experiments we present in this chapter have been tested on different robots (Figure 3.1). One of them is the ActivMedia PowerBot robot via the Player-Stage open source library. We have also successfully run the navigation and localization software on a Magellan Pro mobile robot which is also supported by the Player-Stage library. Finally we have also tested navigation on the lightweight Evolution Robotics ER-1 robot. For this one the only way is to use the manufacturer's API which is only available for Windows.

Figure 3.1: ER-1, PowerBot and Magellan Pro mobile robots equipped with omnidirectional cameras.

The omnidirectional mirror is the Remote Reality's OneShot360 which

needs a camera has to be connected to it. Some localization tests were performed using a compact Canon PowerShot photographic camera. For navigation we started using a low resolution Logitech webcam, however, a better resolution video camera is needed in order to perform both navigation and localization. The high-resolution camera is a GreyPoint Flea2 firewire camera (Figure 3.2) with Pentax lens, connected to the omnidirectional mirror.

Figure 3.2: Omnidirectional mirror and Flea2 camera

Finally the laptop computer used in the experiments has an Intel Centrino 1.6 processor with 512 MB of RAM. A PCMCIA card with firewire adapter had to be installed, as the firewire of the laptop does not have the power line. The firewire adapter allows to connect an external power supply of 12V, which we take from the robot's power supply.

3.2.2. Resolution

The resolution of the images used for the following experiment is different depending on the task:

- Localization: Omnidirectional images, 400×400 pixels
- Navigation (direction): Rectified panorama images, 100×600 pixels
- Navigation (obstacle avoidance): Omnidirectional images, 600×600 pixels

These resolutions can be fixed depending on the quality of the camera and the processing time constraints.

The time resolution of the image acquisition is 10Hz for navigation, as the time to process an image, estimate the direction of the corridor and find obstacles for their avoidance takes less than 100 milliseconds. Localization takes from 100 to 1000 milliseconds, depending on the image resolution and the number of selected filters. However, localization does not have to be executed every second, so it is not a very time-consuming task.

3.2.3. Software

The software was programmed in C++ (GNU C++ compiler) due to processing time restrictions. The vision library used to capture from firewire and webcam cameras is OpenCV. The library used for controlling the robot is Player-Stage. The operating system on which all the software runs is a Debian GNU/Linux distribution. Given that all the software is open source, it could have be run a different operating system.

For the Feature Selection experiments we used The MathWorks Matlab and the machine learning toolbox Spider (open source) for Matlab.

3.2.4. Environments

The environments where the experiments were run are the indoor (Figures 3.4) and outdoor (Figures 3.3) environment of the building III of the "Escuela Politécnica Superior" of the University of Alicante.

Figure 3.3: Outdoor environment. Escuela Politécnica Superior, edificios II y III, University of Alicante.

3.3. Localization

A fundamental problem in mobile robotics is localization. Cameras provide much richer information than range sensors like laser beams and sonars, informing about textures and illumination. However, the correct interpretation of that information is a complex problem which still remains unsolved. Structural-description models and image-based models are two well-differentiated approaches to visual recognition. Structural descriptions represent objects and scenes as 3D or 2D parts which have relative positions among them, or in the space. This kind of data is very useful for mobile

Figure 3.4: Indoor environment. Escuela Politécnica Superior, edificio III, University of Alicante.

robotics. Though, to retrieve such high-level information from an image has proved to be a hard task, where computationally expensive algorithms are involved. Moreover such systems are very sensitive to illumination variations, noise, and other environmental changes.

Image-based recognition is causing great interest in computer vision. In robotics it has received most attention in the field of visual navigation. Jogan and Leonardis [Jogan and Leonardis, 2003], for example, construct an omnidirectional appearance-based model of the environment for localization, with reduced dimensionality (PCA) and a mechanism for rotational invariance. Menegatti et al. [Menegatti et al., 2004] weight omnidirectional samples according to similarity among images, to implement then a Monte-Carlo localization for indoor environments. On the other hand, appearance-based approaches tend to be more suitable for outdoor and arbitrary environments (like those in Figure 3.5).

In the present application we adopt an appearance-based visual learning approach by extracting low-level information from omnidirectional camera images. Such information is extracted by applying a large set of low-level filters (edge detectors, colour filters, etc) to the image. Feature selection is used to automatically select the most appropriate set of filters for any particular environment. The computational complexity of the method does not depend on the image size, but on the number of filters used to evaluate an image. The approach is based on the idea that a very small number of filters are sufficient for many classification tasks, moreover, they usually outperform classifications with larger number of features [Blum and Langley, 1997, Jain and Zongker, 1997].

A possible application is *"the kidnapped robot problem"* in which the robot is placed in a random position of the map and is turned on. It is

useful to supply robot's localization with a higher level information ("You are in the living room, near to the window.") than its sonars could provide.

3.3.1. Input images

For the localization purpose we present experiments performed on two data sets: indoor and outdoor (as shown in Figure 3.5). These image sets were already presented in Chapter 2. The source of the data is a camera with an omnidirectional mirror. This kind of sensor receives much more information from the environment than a local view (a traditional camera). Thus, the problems of angle of view and coverage are discarded and the only problems that remain unsolved are occlusions and noise.

Figure 3.5: An example of a sequence of indoor images

Images were taken a) along the corridors of a building and b) around a building. The corridors and the route around the building are represented in the Figure 3.6. The number of images taken were 70 for each one of the experiments. The physical distance between pictures is roughly 1,50m for the indoor experiment and 3m for the outdoor experiment.

Once the images are collected and hand-labelled, the filters bank is used to calculate a large number of features for each image. Then feature selection takes place. The process is explained in the following sections. In the end, the constructed system consists of a set of selected filters and a built classifier. The system is capable of identifying the label of a new (unlabelled) image. The only computational cost relies upon the number of selected filters, which will have to be extracted from each new test image.

3.3.2. Feature extraction

The filters we use have a low computational cost, and are rotation invariant. The colour filters are calculated in the HSB colour space in order to maintain some tolerance to lighting variations. The 17 filters are:

Figure 3.6: Supervised learning experiments: indoor and outdoor. Class division and labelling are represented. (Polytechnic School at the University of Alicante)

- Nitzberg
- Canny
- Horizontal Gradient
- Vertical Gradient
- Gradient Magnitude
- 12 Color Filters H_i, $1 \leq i \leq 12$

Gradients are applied to a rectified panoramic image in order to provide rotation invariance.

These filters are applied to four different parts of the image, as shown in Figure 3.7. The image is divided in four concentric rings to keep rotation invariance The idea comes from the Transformation Ring Projection [Tang et al., 1991] and is explained in Section 2.2.3.

From the filter responses we extract histograms in order to make the features independent to the order of the pixels. We performed experiments with 2, 4, 12 and 256 bins in the histograms. Each feature contains the value of a definite histogram's bin, of a definite filter, applied to one of the four rings of the image. The whole process is illustrated in Figure 3.7. The total number of features N_F per image is

$$N_F = C * K * (B - 1) \tag{3.1}$$

where C is the number of rings (4), K the number of filters (17) and B is the number of bins.

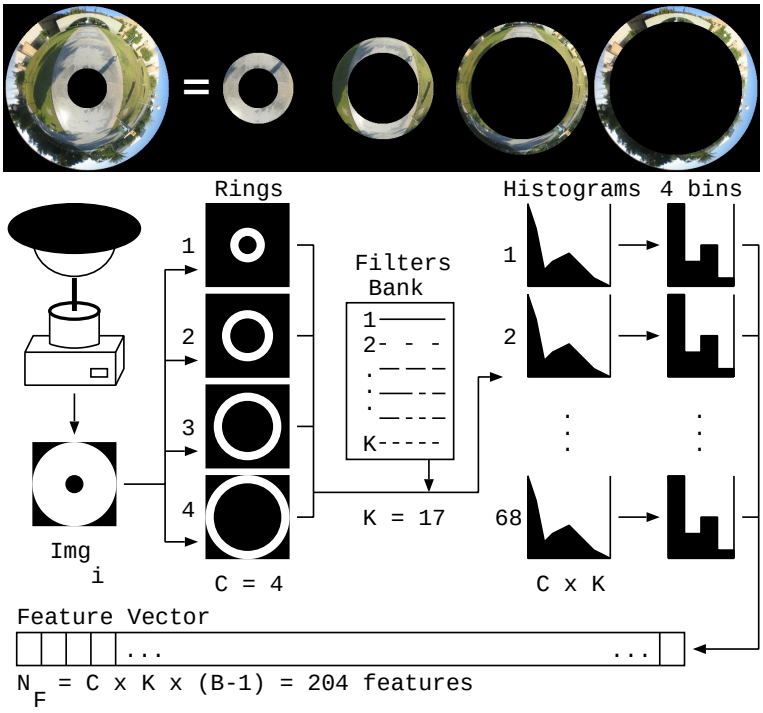

Figure 3.7: The feature extraction process

3.3.3. Supervised learning

The different parts of each environment are labelled by hand.Labellings are represented in Figure 3.6. The classes we establish are not clusters inherent to the data, but are established by a human criterion. For example one class could be, "The beginning of the corridor with blue columns", another could be "Near to the entrance".

In Table 3.1 we can see classification errors for experiments with different number of classes. For a higher number of classes we can expect worse classification error. We can also see that 2-bins histograms present some lack of information for several experiments. On the other hand, 256-bins histograms are excessive and do not improve results. Based on these experimental results it can be estimated that using 4 bins is reasonable. With 12 bins the amount of features is very large and feature selection has a much higher computational cost.

		Number of bins			
	L	2	4	12	256
indoor	2	8.809	9.142	9.142	5.619
	4	11.571	9.190	8.714	15.714
	8	30.571	22.381	26.952	26.904
	16	49.000	51.667	60.571	55.905
outdoor	2	1.392	2.785	1.964	5.500
	4	4.714	5.785	3.892	7.142
	8	12.428	7.964	12.500	18.464

Table 3.1: Errors in % : Supervised Learning (without Feature Selection) - Indoor and Outdoor experiments. 10-fold Cross Validation errors calculated for experiments labelled with 2, 4, 8 and 16 classes (L), and histograms with 2, 4, 12 and 256 bins.

3.3.3.1. Clustering analysis

We also analysed clusters in the data with the K-means algorithm for different number of classes. Some of the resulting clusters coincide with the classes we establish (with a small offset) and other do not. Disagreements are usually due to some object present in the image, shades and light. We calculated that for the 8-classes experiment, the percentage of disagreement is 14% indoor, and 19% outdoor. For the 16-classes indoor experiment the labels and the clustering disagree in 27%.

3.3.3.2. Feature Selection

The best three features for the 8-classes indoor experiment are: Canny, Color Filter 1, and Color Filter 5, all of them applied to the ring 2 (notation illustrated in Figure 3.7). The Cross Validation (CV) error yielded with only 3 features is 24,52%, much better than the 30,57% CV error yielded by the complete set of 68 features. For the outdoor experiment the best three features are Color Filter 9 on ring 2, Nitzberg and Color Filter 8 on ring 3. In this case 3 features are not enough (see Table 3.2) for decreasing the error, as this experiment needs more features to be selected.

Finally it is worth commenting out some statistics about the selected filters. The rings whose features are selected are usually the ring 2 and ring 3. The ring 1 and the ring 4 always have a little lower number of selected features. On the other hand when a bin from a definite filter

		All Features		Feature Sel.	
	L	Feat.	Err.%	Feat.	Err.%
indoor	8	68	30.57	3	24.52
	8	68	30.57	38	13.33
	8	204	22.38	45	7.51
	8	748	26.95	234	3.33
	16	68	49.00	31	26.43
	16	204	51.67	66	22.86
outdoor	8	68	12.42	3	16.07
	8	68	12.42	19	3.93
	8	204	7.96	138	1.42
	8	748	12.50	184	1.25

Table 3.2: Cross Validation errors without and with Feature Selection, as well as the number of selected features for several experiments indoor and outdoor, with different number of classes (L) and different number of bins (2 bins for 68 feat., 4 bins for 204 feat. and 12 bins for 748 feat.)

is selected, it is highly probable that the rest of the bins of that filter will also be selected. Another interesting phenomenon is that in some cases the Nitzberg, Canny, Horizontal Gradient, Vertical Gradient, and Gradient Magnitude are selected all together, even though some of them are redundant. However, if we force the number of selected features to be low, this does not occur. For three features always the selected features are some edge detector and two colour filters.

About the differences between the indoor and the outdoor experiments, it can be said that the indoor environment requires less different colour filters (roughly 2 colours). For the ring 2, only colour filters are selected, while for the ring 3 the filters selected are edge detectors. In the outdoor case, the selected features are more homogeneously distributed among rings and filters.

3.3.3.3. Image Data

The image data experiment consists in classifying the images of Data set 5 (Section 2.2), which were taken from a human point of view, around rooms, corridors, stairs, and an outdoor route. Its application is environment recognition, as well as finding the most similar image in the training set. In this experiment the data set is labelled with 6 different classes.

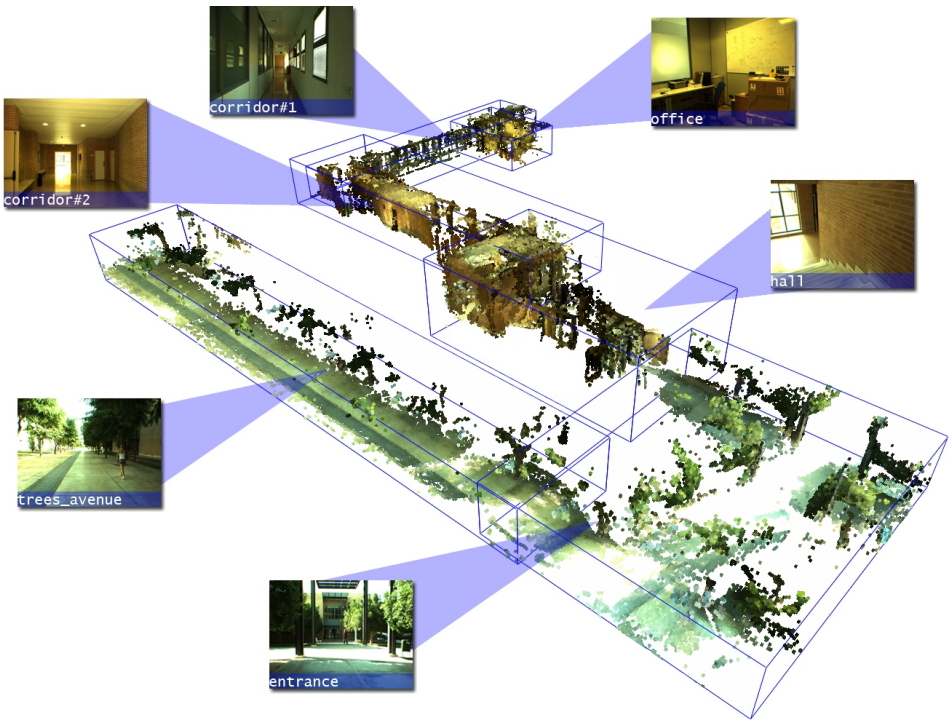

Figure 3.8: A 3D reconstruction of the route followed during the acquisition of the Data set 5, and examples of each one of the six classes. Image obtained with 6-DOF SLAM, by courtesy of Juan Manuel Sáez.

The total number of features extracted from the images can be varied by changing the number of bins of the filter histograms. Experiments showed that for the present experiment a good number of bins is 4. In Figure 3.10 it can be seen that 2-bins generate such a small number of features, that the 10-fold Cross Validation (CV) error remains too high. On the other hand, more than 4 bins, while increasing the total number of feature unnecessarily, do not minimize the error very much. The classification performance also depends on the number of classes in which the data set is divided. Figure 3.10 shows that a lower number of classes yield better classification performances. For the present experiment we have divided the images in 6 different environments: an office, two corridors, stairs, entrance, and a tree avenue.

After the filter feature selection process, the feature sets have to be tested [1] with a classifier in order to see the classification performances they

[1]Note that this is a filter feature selection method, therefore the classifier does not

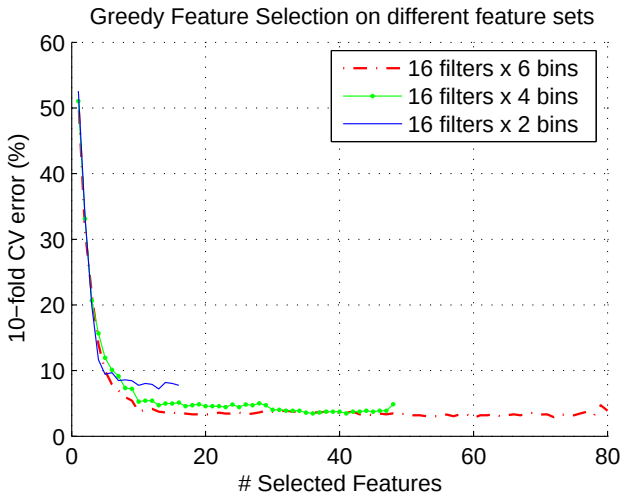

Figure 3.9: Finding the optimal number of bins N_b.

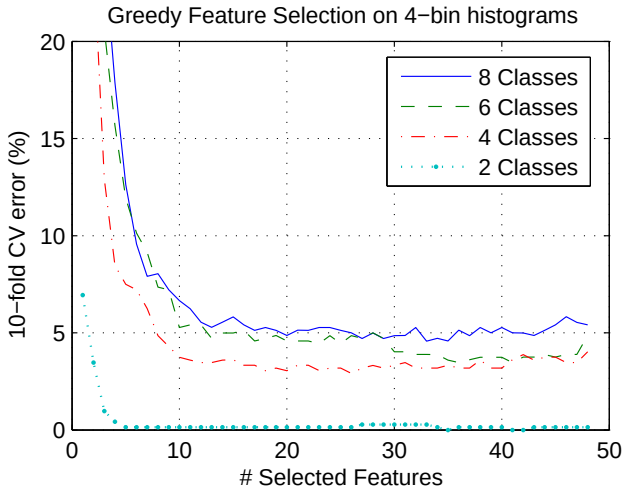

Figure 3.10: Evolution of the CV error for different number of classes N_c.

actually yield. An example with test images and their nearest neighbours from the training set, is shown on Figure 3.11. The k-nearest neighbours algorithm (k-NN) is usually suitable for classification problems where the number of samples is not high. Feature selection can increase the linear separability of the classes, avoiding the need for a more complex classifier [Vidal-Naquet and Ullman, 2003]. We have compared the performances of the k-NN and a Support Vector Machine (SVM) classifier and the latter does not outperform K-NN for the present classification problem. In Table 3.3 we show the confusion matrices of 6-class classifications for k-NN and SVM.

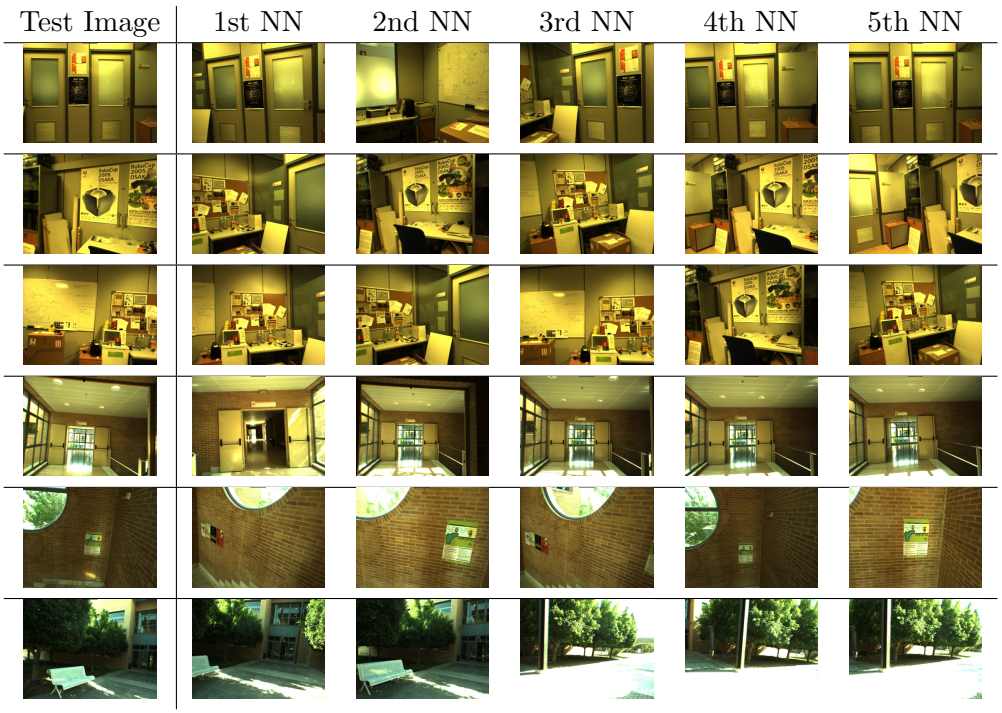

Figure 3.11: The Nearest Neighbours of different test images. The training set from which the neighbours are extracted contains 721 images taken during an indoor-outdoor walk. The amount of low-level filters selected for building the classifier is 13, out of 48 in total. Note that the test images of the first column belong to a different set of images.

In Figure 3.13 we show the classification errors of the feature sets selected with the MD, MmD and mRMR criteria. Only 20 selected features

need to be built for selecting the features.

	C#1	C#2	C#3	C#4	C#5	C#6
C#1	26	0	0	0	0	0
C#2	2/3	63/56	1/4	0/3	0	0
C#3	0	1/0	74/67	1/9	0	0
C#4	4/12	5/6	10/0	96/95	0/2	0
C#5	0	0	0	0	81	0
C#6	0	0	0	0	30/23	78/85

Table 3.3: K-NN / SVM Confusion Matrix

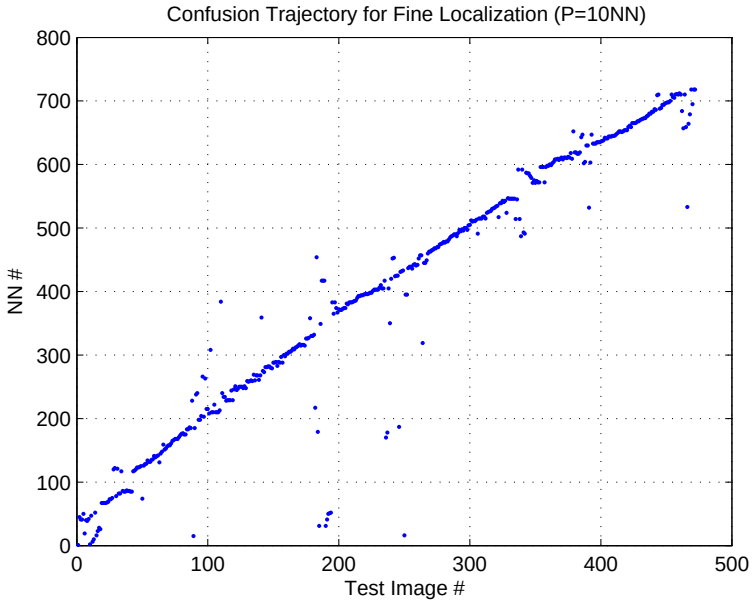

Figure 3.12: The nearest neighbour (from among the train images, Y axis) of each one of the test images, X axis. In the ideal case the line should be almost straight, as the trajectories of both train and test sets are similar. This figure illustrates the appropriateness of the presented classification method for coarse visual localization.

are represented, as for larger features sets the error does not decrease. The 10-fold Cross Validation error of K-NN is represented, as well as a test error, which was calculated using the additional test set of images, containing 470 samples.

With mRMR, the lowest CV error (8.05%) is achieved with a set of 17

features, while with MD a CV error of 4.16% is achieved with a set of 13 features. The CV errors yielded by MmD are very similar to those of MD. On the other hand, test errors present a slightly different evolution. The MmD test error descends faster than the MD test error, only for the first 7 feature sets. For the rest of the feature sets the best error is given by MD, and the worst one is given by mRMR. In Peng et al. [Peng et al., 2005],

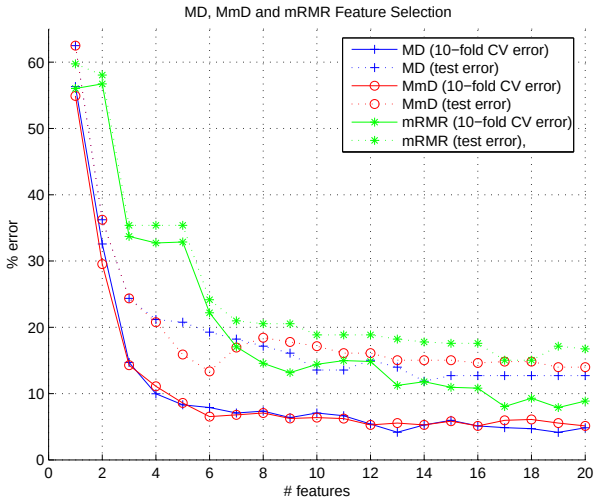

Figure 3.13: Feature Selection performance on image histograms data with 48 features. Comparison between the Maximum-Dependency (MD), Maximum-Minimum-Dependency (MmD) and the Minimum-Redundancy Maximum-Relevance (mRMR) criteria.

the experiments yielded better results with the mRMR criterion than with the MD criterion. However, we obtain better performance using MD. This may be explained by the fact that the entropy estimator we use does not degrade its accuracy as the number of dimensions increases.

3.4. Navigation

The ability to navigate in a structured environment without any prior knowledge is crucial for mobile robots. Vision-based navigation systems have the advantage of not needing any additional hardware to be installed on the robot. The installation of a range sensor on a mobile robot has an additional cost, while a vision sensor provides more abundant and richer

information. Catadioptric sensors provide the robot with a 360°field of vision, information which can be exploited in different ways for navigation.

Some approaches are oriented toward extracting 3D information, for example the structure from motion approach [Geyer and Daniilidis, 2001]. This requires the catadioptric sensor to be calibrated. Catadioptric projection can also be modelled by a spherical projection [Lopez-Franco and Bayro-Corrochano, 2004], providing a mathematical framework for working with omnidirectional vision.

Other approaches to navigation do not need any 3-D information about the structure of the target [Benhimane and Malis, 2006, Mariottini et al., 2006]). In [Gaspar and Santos-Victor, 2000] the original 2-D image is transformed into a 2-D orthographic image (Figure 3.14.b).

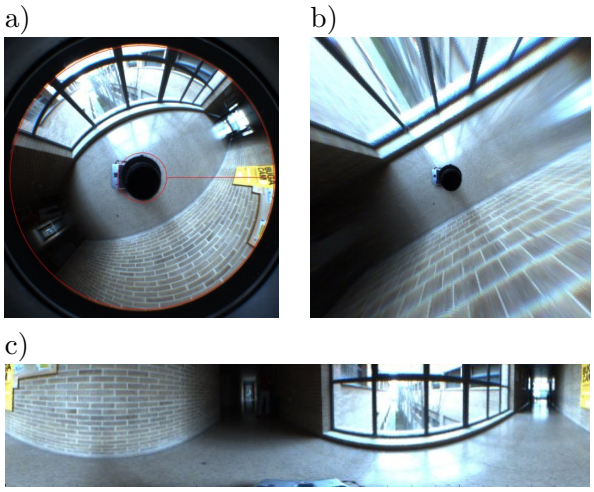

Figure 3.14: a) Omnidirectional image obtained with a camera and a hyperbolic mirror. b) Transformation to bird's view (orthographic image). c) Rectified panorama image.

3.4.1. OmniVisual sonars

For the present application we propose a vision-based method inspired in sonar rings, and refer to it as *OmniVisual sonars* (OV-sonars). OV-sonars consist of virtual rays (similarly to [Lenser and Veloso, 2003]) which are launched radially from the center of the omnidirectional image and reach the first point where the gradient of the image is high (more than 0.1 in our experiments). Each k-th OV-sonar $\overrightarrow{V}_k = \{v_1, \cdots, v_r\}$ explores a

maximum of r pixel of the omnidirectional image. Each pixel v_i corresponds to:

$$\begin{aligned} v_i &= I_O(i \sin\alpha + c_x,\ i \cos\alpha + c_y) \\ \alpha &= k \cdot 2\pi / N_s \end{aligned} \tag{3.2}$$

where I_O is the original 2D omnidirectional image with radius r and center in (c_x, c_y) and N_s is the number of OV-sonars launched for each image. In the implementation we also perform a horizontal and vertical sub-pixel interpolation.

The $\lambda(\vec{V})$ function stands for the number of pixels that the OV-sonar $\vec{V}$ explores before the Δv_{max} threshold is exceeded:

$$\lambda(\vec{V}) = \arg\min_i |v_i - v_{i-1}| > \Delta v_{max} \tag{3.3}$$

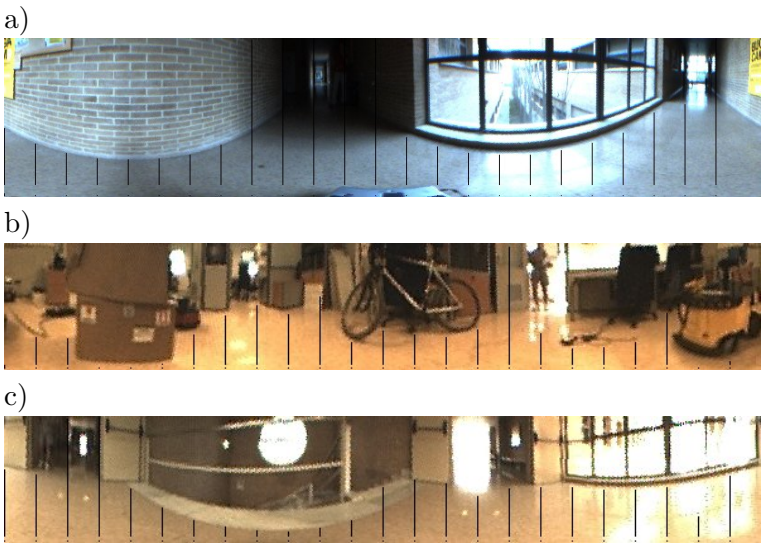

Figure 3.15: *OmniVisual sonars* with gradient threshold on different images. On a) we can see some false negatives in the dark zones of the image. On c) there is a false positive due to saturation. Note that the picture c) was taken near to stairs and a handrail which are a true positive obstacle in the example, due to the detection of a border on the floor.

In Figure 3.15 we present three examples of OV-sonars using the rectified representation. In the experiments we launched 24 OV-sonars per frame. Larger quantities of OV-sonars would help only for the detection of small obstacles while increasing the computational load.

OV-sonars detect obstacles and free zones in the environment. The following subsection details how this information is used for navigation.

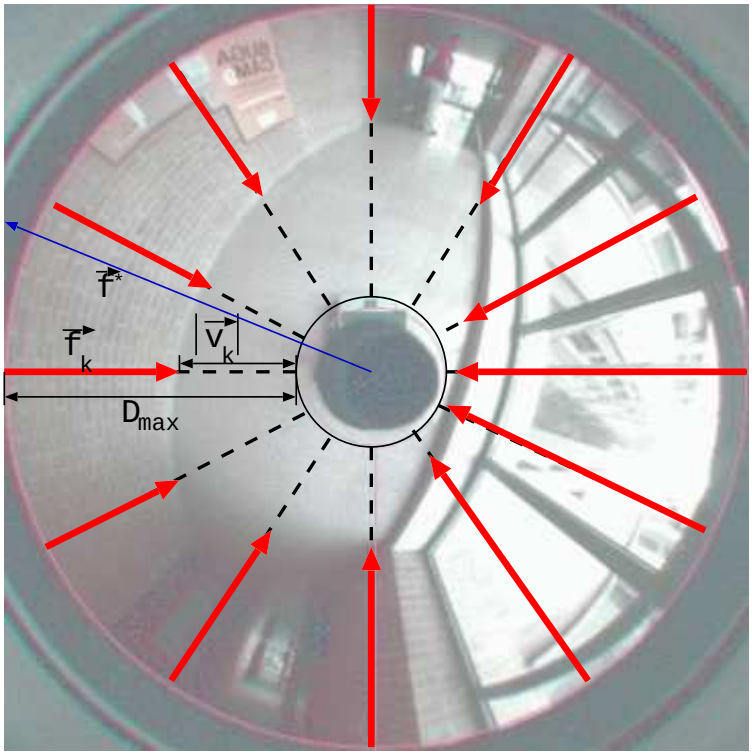

Figure 3.16: The force vectors $\vec{f}_k$ (red thick arrows) represent the difference between the maximum distance d_{max} and the vectors of the OV-sonars $\vec{v}_k$ (dashed lines). The sum of all force vectors $\vec{f}^*$ is represented with a blue thin arrow.

3.4.2. Navigation

Behaviour-based architectures have proved to be convenient for systems which simultaneously have various objectives and environmental restrictions [Arkin, 1999]. In the previous subsections we presented two basic behaviours, one of them for guiding the robot, and the other one for avoiding collisions. Following we propose the combination of both of them in the same system.

A simple navigation method for robots with a sonar ring, is the sum of force vectors. The algorithm can be used with the OV-sonars in the

vision-based system.

$$\begin{aligned} \vec{f}_k &= d_{max} \cdot \widehat{v}_k - \vec{v}_k \\ \vec{f}^* &= \sum_{k=1}^{N_s} \vec{f}_k \end{aligned} \tag{3.4}$$

where N_s is the number of OV-sonars and $\vec{f}^*$ is the resulting vector whose direction would avoid the collision. We set $d_{max} = r$ the radius of the omnidirectional image. See Figure 3.16.

3.5. Conclusions

The experiments presented in this chapter are mainly oriented to localization in a mobile robotics system. Image-based classification with feature selection proved to be feasible for the visual localization task. The objective of this approach is to use a large set of low-level filters which are applied to the images and then their responses are transformed into histograms. The set of filters is as general as possible, so that feature selection can decide which filters to use in each problem. This approach is useful for coarse localization [Bonev et al., 2007d, Escolano et al., 2007] in both indoor and outdoor environments. It is very general and it is suitable for a variety of environments, however, we showed that the classification performance is degraded as the number of classes increases.

Regarding the relevance of filters, we observed that the three most predictive features always are a corner or edge detector, and two different colour filters. In the omnidirectional camera experiments we divided the image in four rings which are related to the proximity to the robot. Most of the filters were selected from the second and the third ring (see Figure 3.7).

Finally, we also used the same approach to build a k-NN classifier which was used for a fine-loclization purpose. In Figure 3.11 we show that the nearest neighbours are very similar to the test images. In order to illustrate the error of this experiment we present a confusion plot of the trajectory (Figure 3.12). For most of the test images the nearest neighbor is the correct one. There are some failures, however, which yield an image which is far away in the sequence of images.

Chapter 4

Application to gene selection

"As of 1997, when a special issue on relevance including several papers on variable and feature selection was published [Blum and Langley, 1997, Kohavi and John, 1997], few domains explored used more than 40 features.", [Guyon and Elisseeff, 2003].

The aim of the experiments in this application is to show the performance of the feature selection criteria on high-dimensional data found in a real pattern recognition problem.

4.1. Microarray analysis

Advances in gene expression microarray technology have enabled the measurement of the expression levels of tens of thousands of genes in a single experiment. Researchers look for molecular markers which can be exploited for diagnosing, prognosing and predicting cancer. They use the data of gene expression levels related to multiple individuals and multiple tissue and tumor samples. Moreover, microarray studies are bringing about a revolution in the understanding of the molecular mechanisms underlying biological processes.

The basic setup of a microarray platform is a glass slide which is spotted with DNA fragments or oligonucleotides representing some specific gene coding regions. Then, purified RNA is labelled fluorescently or radioactively and it is hybridized to the slide. The hybridization can be done simultaneously with reference RNA to enable comparison of data with multiple experiments. Then washing is performed and the raw data is obtained by scanning with a laser or autoradiographing imaging. The obtained ex-

pression levels are numerically quantified and used for statistical studies. An illustration of the process can be seen in Figure 4.1, where ADNc denotes complementary ADN. When hybridization is done with ARN and ADN, the complementary ADN is used instead of genetic ADN.

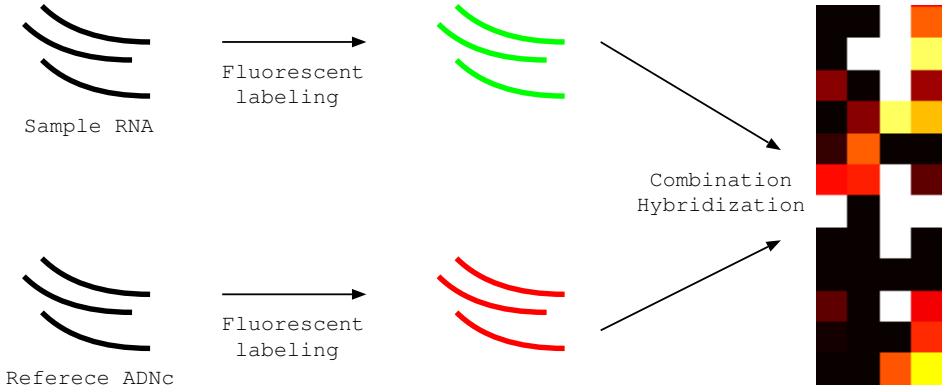

Figure 4.1: An illustration of the basic idea of a microarray platform.

An delicate issue in the setup of a cancer study microarray experiment is the reference RNA. It should be extracted from the same kind of tissue in order to compare cancer-tissue with non-cancer tissue. The selection of the individuals is important too. There are many other factors which could vary the levels of expressions of the genes. Some factors are inherent to the individuals (diet, age, etc). Some other are due to differences among instruments and methods in different laboratories. Thus, experimental conditions could be very different among different data bases. To help repeatability it is a good idea to do all the experiments in the same laboratory.

In many microarray databases the genes which are stored are a preselected set of genes. These genes are not automatically selected but are a human decision based on previous experience in the type of disease which has to be studied. For example, there are a number of genes which get overexpressed in several cancer diseases, so these are included in cancer-related microarray experiments. Other genes which are not considered relevant, are not included. However, the number of genes included in the final data is still of the order of thousands. This high dimensionality of the patterns turns their analysis into a challenging pattern recognition problem. As stated in [Wang and Gotoh, 2009], "one intractable problem [...] is how to reduce the exceedingly high-dimensional gene expression data, which

contain a large amount of noise". In this chapter we present our work toward making this problem more tractable.

The process of measuring the gene expression levels has a high economical cost. Many genomics researchers send their data to specialized laboratories which perform these experiments at an affordable cost. In this work we have used databases which are publicly available. Their public availability is convenient for comparing our resluts with those presented in the literature [Jirapech-Umpai and Aitken, 2005, Pavlidis and Poirazi, 2006, Díaz-Uriate and de Andrés, 2006, Gentile, 2003, Díaz-Uriate and de Andrés, 2006, Ruiz et al., 2006, Singh et al., 2002, Pirooznia et al., 2008].

Other state-of-the-art feature selection and machine learning methods for gene selection and classification can be found in [Saeys et al., 2007b] and [Larrañaga et al., 2006]. An alternative, inspired in genetic algorithms, is the use of estimation distribution algorithms (EDAs). As explained in [Armañanzas et al., 2008], this paradigm is not only appropriate for DNA microarray classification, but also for DNA clustering and for the inference of genetic networks.

4.2. Microarray experiments

In order to illustrate the dimensionality independence of the proposed filter feature selection method, we present classification experiments on some well-known microarray data sets. The objective is to identify small sets of genes with good predictive performance for diagnostic purposes. Traditional gene selection methods often select genes according to their individual discriminative power. Such approaches are efficient for high-dimensional data but cannot discover redundancy and basic interactions among genes. The contribution of the current work is the efficient evaluation of whole sets of features.

Firstly, we will evaluate the MD and the MmD criteria with experiments on the NCI data set and next we will show results for other microarray data sets. The NCI data set contains 60 samples (patients), each one containing 6,380 dimensions (genes). The samples are labelled with 14 different classes of human tumor diseases. The purpose of Feature Selection is to select those genes which are useful for predicting the disease.

In Figure 4.3 there are represented the increase of Mutual Information and the resulting Leave One Out Cross Validation Error [1] (LOOCV), al-

[1] LOOCV measure is used when the number of samples is so small that a test set

ready explained in Section 2.3. The error decreases until 39 features are selected, and then it slowly increases, due to the addition of redundant and noisy genes.

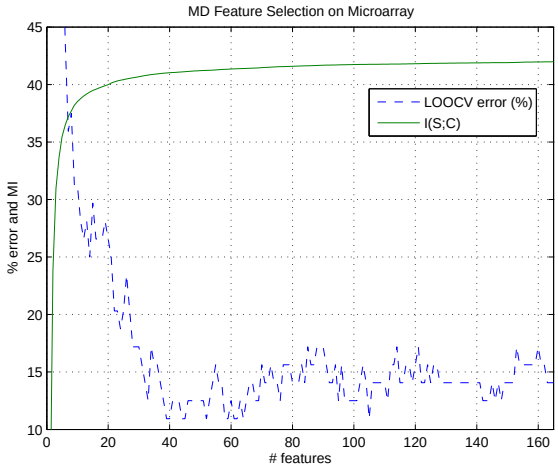

Figure 4.2: Maximum Dependency Feature Selection performance on the NCI microarray data set with 6,380 features. The Mutual Information of the selected features is represented. The FFS algorithm using the Maximum Dependency criterion obtained the lowest LOOCV error with a set of 39 features.

We also tested the MmD criterion on the microarray data sets, and the resulting performances are similar (see Table 4.1), being slightly better the error rates yielded by MD. In Figure 4.3 we can see that for the NCI data set, the first 16 selected features are the same for both criteria. For the subsequent feature sets, firstly the MmD criterion achieves lower error, but with the addition of more features MD wins. In order to understand better the behaviour of both criteria we have represented the gene expression matrix of the best feature sets selected by MD and MmD on the Figure 4.5. There are 24 genes selected by both criteria, and despite the rest of the genes are different, the overall expression matrices keep a similar aspect. This means that there are different genes with similar expression, and therefore with similar information about the prototypes. We present the same kind of comparative with the mRMR criterion, and the errors yielded are presented in Figure 4.4. The MD criterion outperforms the rest, due to the precision in the estimation of mutual information. The selected genes are represented in Figure 4.6.

cannot be built.

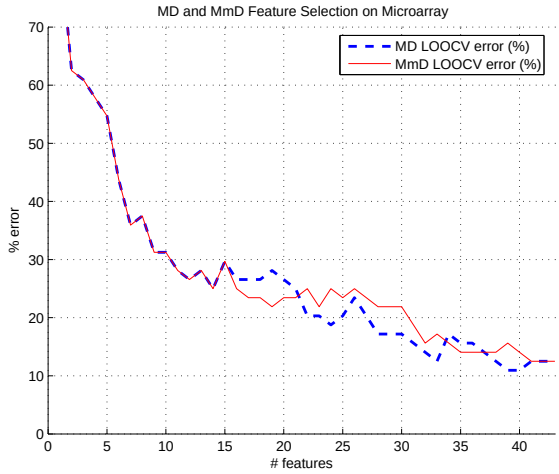

Figure 4.3: Maximum Dependency and Maximum-Minimum Dependency Feature Selection performance on the NCI microarray. Only the first 43 features are represented, out of 6,380.

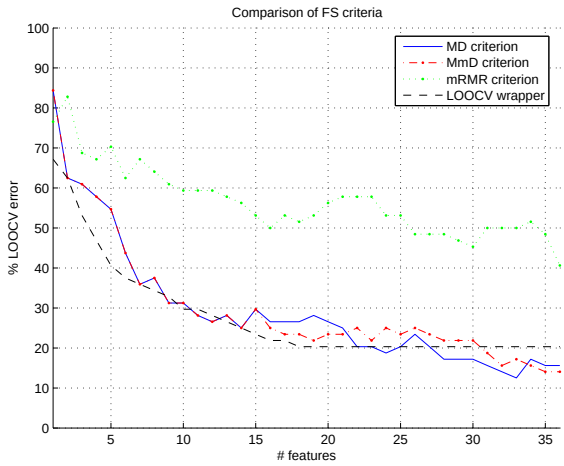

Figure 4.4: Performance of Maximum Dependency, Maximum-Minimum Dependency, Minimum-Redundancy Maximum-Relevance and LOOCV Feature Selection criteria on the NCI microarray. Only the selection of the first 36 features (out of 6,380) is represented.

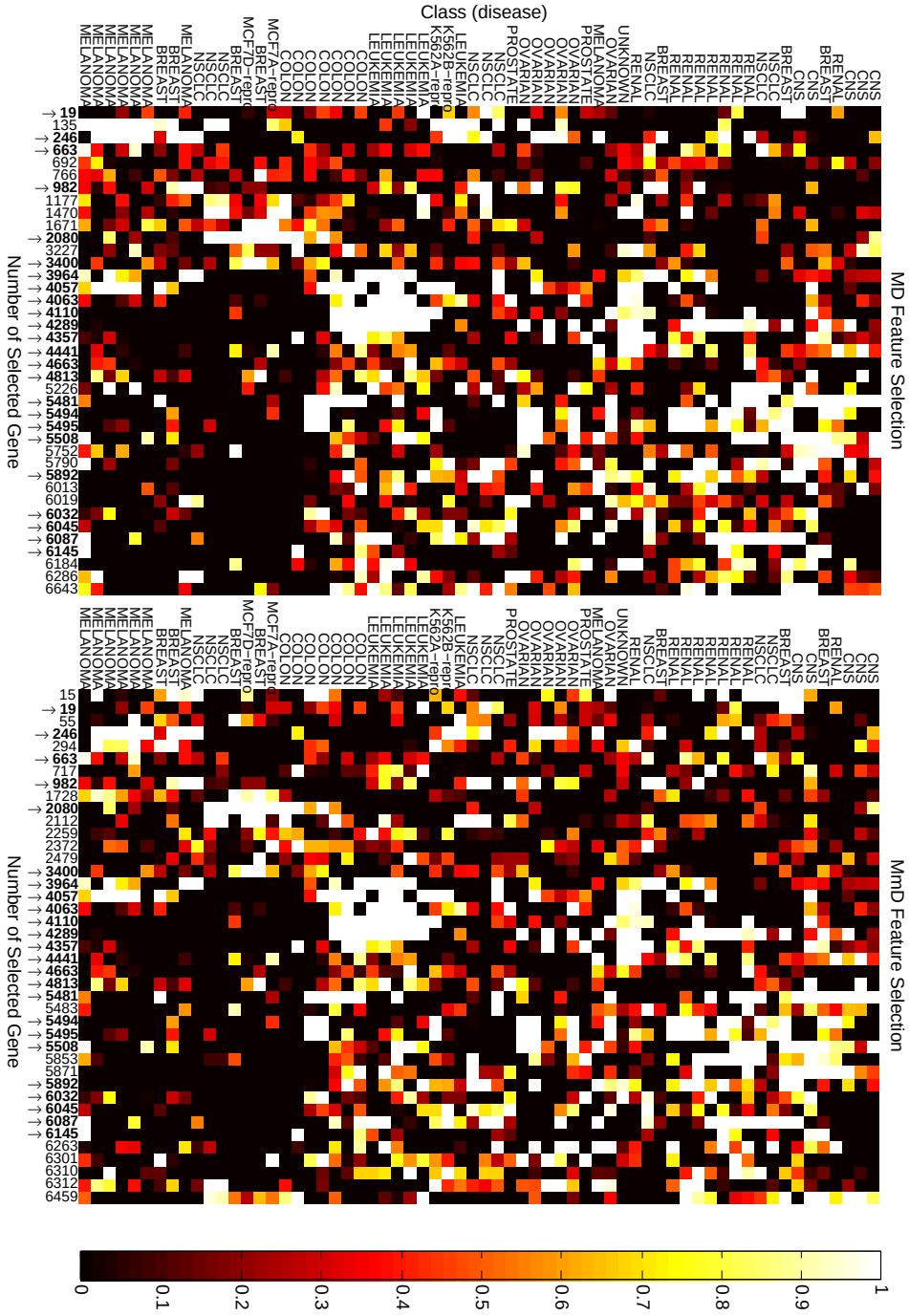

Figure 4.5: Feature Selection on the NCI DNA microarray data. The MD (on the left) and MmD (on the right) criteria were used. Features (genes) selected by both criteria are marked with an arrow.

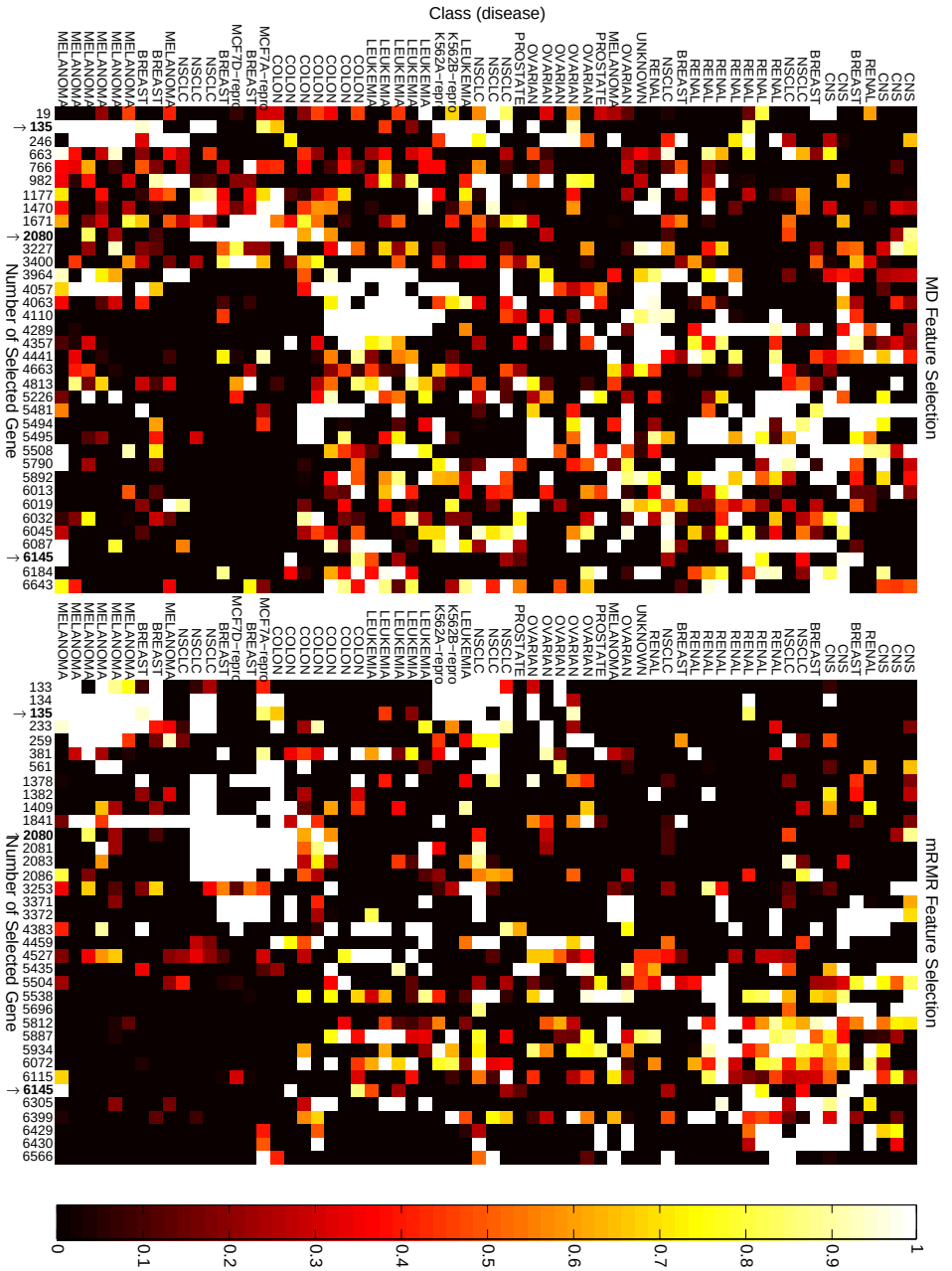

Figure 4.6: Feature Selection on the NCI DNA microarray data. The MD (on the left) and mRMR (on the right) criteria were used. Features (genes) selected by both criteria are marked with an arrow.

Figure 4.2 shows the Leave One Out Cross Validation errors for the selected feature subsets, using the Max-Dependency criterion. Only the best 220 genes (out of 6,380) are on the X axis, and it took about 24 hours on a PC with Matlab to select them. During this time MI was calculated $\sum_{i=1}^{220}(6380 - i + 1) = 1,385,670$ times.

In [Jirapech-Umpai and Aitken, 2005] an evolutionary algorithm is used for feature selection and the best LOOCV error achieved is 23.77% with a set of 30 selected features. In the experiments we achieve a 10.94% error with 39 selected features. In addition to the experiments with the NCI data set, we have performed FS experiments on other four well-known microarray expression data sets [2].

- Leukemia: The training data set consists of 38 bone marrow samples labelled with two classes, 27 Acute Lymphoblastic Leukemia (ALL) and 11 Acute Myeloid Leukemia (AML), over 7,129 probes (features) from 6,817 human genes. Also 34 samples testing data is provided, with 20 ALL and 14 AML. We obtained a 2.94% test error with 7 features, while [Pavlidis and Poirazi, 2006] report the same error selecting 49 features via individualized markers. Other recent works [Díaz-Uriate and de Andrés, 2006, Gentile, 2003] report test errors higher than 5% for the Leukemia data set.

- Colon: This data set contains 62 samples collected from colon-cancer patients. Among them, 40 tumor biopsies are from tumors and 22 normal biopsies are from healthy parts of the colons of the same patients. From 6,500 genes, 2,000 were selected for this data set, based on the confidence in the measured expression levels. In the feature selection experiments we performed, 15 of them were selected, resulting in a 0% LOOCV error, while recent works report errors higher than 12% [Díaz-Uriate and de Andrés, 2006, Ruiz et al., 2006].

- Central Nervous System Embryonal Tumors: This data set contains 60 patient samples, 21 are survivors (patients who are alive after treatment) and 39 are failures (those who succumbed to their disease). There are 7,129 genes in the data set. We selected 9 genes achieving a 1.67% LOOCV error. In [Pavlidis and Poirazi, 2006], a 8.33% CV error with 100 features is reported.

[2] Datasets can be downloaded from the Broad Institute http://www.broad.mit.edu/, Stanford Genomic Resources http://genome-www.stanford.edu/, and Princeton University http://microarray.princeton.edu/.

- Prostate: This data set contains two classes for tumor versus normal classification. The training set contains 52 prostate tumor samples and 50 non-tumor prostate samples with 12,600 genes. An independent set of testing samples is also available, but it is from a different experiment and has a nearly 10-fold difference in overall microarray intensity from the training data, so we have applied a 0.1 correction for our experiments. The test set is the one used in [Singh et al., 2002], where extra genes contained in the testing samples are removed, and there are 25 tumor and 9 normal samples. Our experiments on the Prostate data set achieved the best error with only 5 features, with a test error of 5.88%. Other works [Díaz-Uriate and de Andrés, 2006, Gentile, 2003] report test errors higher than 6%.

Only a work of Gentile [Gentile, 2003] reports a little better classification error for the Leukemia (2.50% with 100 features) data set, but we cannot compare it to the presented results because their results refer to different (custom) training/test splits. The best feature selection results achieved with the proposed method are summarized in the Tables 4.1 and 4.2.

4.3. Conclusions

Microarray subset selection is one of the most challenging problems due to its low number of samples defined by a high number of features. With the experiments presented in this chapter we prove the efficacy of the method on high-dimensional data sets.

We compare the MD, MmD and the mRMR criteria on microarray data sets and the results show that the MD criterion outperforms the other. The MD and the MmD have a similar formulation and present similar results. Moreover, their outputs have many coincidences, as shown in the gene expression matrix in Table 4.5. However, the outputs of the mRMR have little coincidences with the other two. This effect should be due to differences in the entropy estimation. In the following chapter (Subsection 5.5.3) we show the impact of entropy estimation on the selected subsets and on the final classification results.

Another conclusion is that on high-dimensional data sets, different subsets can have similar classification performances. Therefore it is difficult to state that a selected subset of genes is actually related to some type of disease. On the one hand, the number of samples should be much higher

Criterion	Errors (%) LOOCV	10FCV	Test	Selected Features $\|S\|$	S
Colon data set, $\|F\| = 2,000$					
MD/MmD	0.00	0.00	N/A	15	1 87 295 371 534 625 653 698 858 1024 1042 1058 1161 1212 1423
NCI data set, $\|F\| = 6,830$					
MD	10.94	10.48	N/A	39	19 135 246 663 692 766 982 1177 1470 1671 2080 3227 3400 3964 4057 4063 4110 4289 4357 4441 4663 4813 5226 5481 5494 5495 5508 5752 5790 5892 6013 6019 6032 6045 6087 6145 6184 6286 6643
MmD	12.50	13.81	N/A	35	15 19 246 294 663 717 982 1728 2080 2112 3400 3964 4057 4063 4110 4289 4357 4441 4663 4813 5481 5494 5495 5508 5853 5871 5892 6032 6045 6087 6145 6263 6310 6312 6459
Leukemia (ALL-AML) data set, $\|F\| = 7,129$					
MD	0.00	0.00	2.94	7	1779 2015 2288 4847 6182 6277 7093
MmD	0.00	0.00	5.88	6	1779 2015 4847 6182 6277 7093 [a]

[a] The first 6 features selected are the same as for MD, however, the next feature selected is different (the 2402), and the test error increases to 8.82%.

Table 4.1: Feature Selection results on different microarray data sets. Continued on Table 4.2

Criterion	Errors (%) LOOCV	10FCV	Test	Selected Features $\|S\|$	S
Central Nervous System, $\|F\| = 7,129$ data set					
MD	1.67	0.00	N/A	15	454 491 560 863 1506 1667 1777 1879 2474 2548 4994 6063 6143 6165
MmD	1.67	1.67	N/A	15	6634
Prostate data set, $\|F\| = 12,600$					
MD	0.00	0.91	5.88 [a]	14	289 1770 3160 3824 4567 6185 6359 6443 6462 7756 8351 9172 9332 10234

[a] This test error is also achieved with only 5 features (160 6185 7756 9172 10234), but the LOOCV error they produce is higher (2.91% LOOCV).

Table 4.2: Feature Selection results on different microarray data sets. (Continued from Table 4.1). Test, Leave One Out Cross Validation (LOOCV) and 10-fold Cross Validation (10FCV) errors are provided. For data sets which do not provide a test set, such error is annotated as not available (N/A). S represents the selected features subset, $|S|$ the number of selected features, and $|F|$ the total number of features in the data set. Max-Dependency (MD) and Max-Min-Dependency (MmD) criteria are compared. The experiments where both criteria yield the same results, are written in one row (MD/MmD).

than the number of features in order to draw a reliable conclusion. On the other hand, even if there were enough samples, there could still exist several different combinations of genes which are able to predict the class, due to biological reasons.

The experiments were performed on some well-known microarray data sets which are publicly available and are used by other authors. We compare our results to the classification errors reported in the literature. The Prostate and Leukemia experiments report the test errors, while the rest of the data sets (CNS, NCI, Colon) only report the LOOCV error because they do not provide a separate test set. The LOOCV is used insted of 10-fold CV because of the low number of samples. The results of the experiments are very promising (Tables 4.1 and tab:genes2), as they outperform most of the already published results [Díaz-Uriate and de Andrés, 2006, Gentile, 2003, Singh et al., 2002, Pavlidis and Poirazi, 2006, Ruiz et al., 2006, Jirapech-Umpai and Aitken, 2005]. The comparison is published in [Bonev et al., 2008].

Chapter 5

Application to structure classification

5.1. Introduction

Although feature selection (FS) plays a fundamental role in pattern classification [Guyon and Elisseeff, 2003], there are few studies about this topic in structured patterns, mainly when graphs are not attributed (pure structure). One exception is the work of Luo et al. [Luo et al., 2003] where different spectral features are investigated, but for embedding purposes. Regarding application areas, graph-based descriptors have been used for 3D object retrieval and classification. In this application we study Reeb graphs [Biasotti, 2005] obtained from different functions. What is the role of each function? What is the role of each spectral feature, beyond the ones studied so far? Answering these questions, through an information-theoretic [Escolano et al., 2009b] method, is the main contribution of this application.

5.2. Reeb Graphs

Given a surface $\mathcal{S}$ and a real function $f : \mathcal{S} \to \mathbb{R}$, the *Reeb graph* (RG), [Reeb, 1946], represents the topology of $\mathcal{S}$ through a graph structure whose nodes correspond to the critical points of f. When f is differentiable, the critical points are located in correspondence of topological changes of S, such as birth, join, split and death of connected components of the surface.

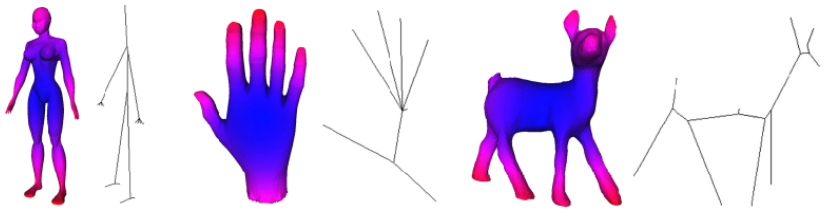

Figure 5.1: Graphs from 3D shapes. Image by courtesy of Daniela Giorgi and Silvia Biasotti.

Hence, RGs describe the *global* topological structure of $\mathcal{S}$, while also coding *local* features identified by f. The graph representation we adopt in this application is the *Extended Reeb Graph* (ERG) proposed in [Biasotti, 2004, Biasotti, 2005] for triangle meshes representing closed surfaces embedded in $\mathbb{R}^3$. The salient feature of ERG is the approximation of the RG by using of a fixed number of level sets (63 in this work) that divide the surface into a set of regions; critical regions, rather than critical points, are identified according to the behaviour of f along level sets; ERG nodes correspond to critical regions, while the arcs are detected by tracking the evolution of level sets.

The most interesting aspect of RGs is their parametric nature. By changing f, we have different descriptions of the same surface $\mathcal{S}$ that highlight different shape properties. Here we choose three alternative scalar functions f, namely the integral geodesic distance defined in [Hilaga et al., 2001] and the two distance functions $f(\mathbf{p}) = ||\mathbf{p} - \mathbf{b}||_2$, with $\mathbf{b}$ the center of mass and the center of the sphere circumscribing the triangle mesh respectively. Figure 5.2 exemplifies our three ERG representations on a hand model, namely a) using geodesic distance [Hilaga et al., 2001], b) the distance from the mass center, and c) from the center of the circumscribing sphere.

5.3. Features from graph spectra

Given a graph $G = (V, E)$, the degree distribution is a major source of statistical information. A more elaborate feature is the *subgraph node centrality* [Estrada and Rodriguez, 2005], which quantifies the degree of participation of a node i in structural subgraphs. It is defined in terms of the spectrum of the adjacency matrix $\mathbf{A}$, i.e. $C_S(i) = \sum_{k=1}^{n} \phi_k(i)^2 e^{\lambda_k}$, where $n = |V|$, λ_k the k-th eigenvalue of $\mathbf{A}$ and ϕ_k its corresponding eigen-

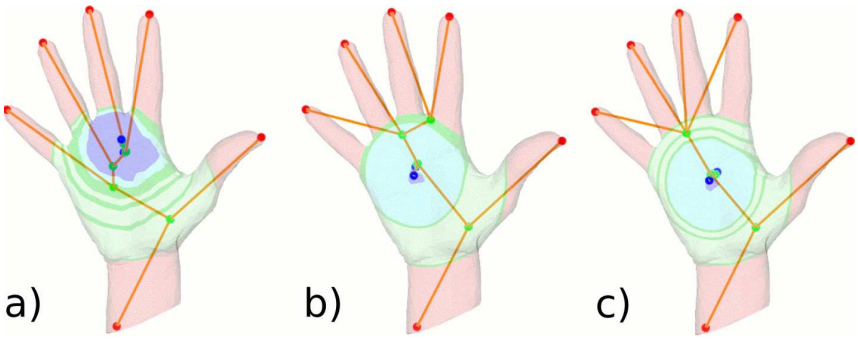

Figure 5.2: Extended Reeb Graphs. Image by courtesy of Daniela Giorgi and Silvia Biasotti.

vector. In this regard, ϕ_n (the eigenvector corresponding to the largest eigenvalue) is the so called *Perron-Frobenius eigenvector*. The components of the latter vector denote the degree of importance of each node in a connected component and they are closely related to subgraph centrality [Estrada and Rodriguez, 2005]. Furthermore, the magnitudes $|\phi_k|$ of the (leading) eigenvalues of $\mathbf{A}$ have been been experimentally validated for graph embedding [Luo et al., 2003]. Besides the study of the adjacency matrix, it is also interesting to exploit the *spectrum of the Laplacian* $\mathbf{L} = \mathbf{D} - \mathbf{A}$ or the *spectrum of the normalized Laplacian* $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal degree matrix. These spectra encode significant structural information. For instance, in the case of $\mathcal{L}$ we have $\lambda_k \leq 2,\ 2 \leq k \leq n$, and the Laplacian spectrum plays a fundamental role in the design of regularization graph kernels. Such kernels encode a family of dissimilarity measures between the nodes of the graph. Regarding the eigenvectors of the Laplacian, the *Friedler vector*, that is, the eigenvector corresponding to the first non-trivial eigenvalue, ϕ_2 in connected graphs, encodes the connectivity structure of the graph (actually its analysis is the core of graph-cuts methods) and it is related to the Cheeger constant. In addition, both the eigenvectors and eigenvalues of the Laplacian are key to defining a metric between the nodes of the graph, namely the *commute time*, $CT(i,j)$. It is the average time taken by a random walk starting at i to reach j and then returning. If we use the un-normalized Laplacian, we have that $CT(i,j) = vol \sum_{k=2}^{n} (1/\lambda_k)(\phi_k(i) - \phi_k(j))^2$, where vol is the volume of the graph, that is, the trace of $\mathbf{D}$. In the normalized case $CT(i,j) = vol \sum_{k=2}^{n} (1/\lambda_k)(\phi_k(i)/\sqrt{d_i} - \phi_k(j)/\sqrt{d_j})^2$, where d_i and d_j are the degrees of i and j respectively. Since the commute time is a metric, and

because of its utility for graph embedding [Qiu and E.R.Hancock, 2007], the path-length structure of the graph is partially encoded. Finally, considering diffusion kernels on graphs, which belong to the family of regularization kernels, a recent characterization of the diffusion process is the *the flow complexity trace* [Escolano et al., 2009a], a fast version of polytopal complexity [Escolano et al., 2008]. The complexity trace encodes the amount of heat flowing through the edges of G for a set of inverse temperatures β: from $\beta = 0$ (no flow) to $\beta \to \infty$ (flow equal to $2|E|$) there is a phase-transition point. More precisely, the instantaneous flow for a given β is $F(G;\beta) = \sum_{i=1}^{n} \sum_{j \neq i}^{n} A_{ij} (\sum_{k=1}^{n} \phi_k(i)\phi_k(j) e^{-\beta\lambda_k})$ and the trace element for this inverse temperature is the instantaneous complexity $C(G;\beta) = \log_2(1 + F(G;\beta)) - \log_2(n)$.

5.4. Feature Selection

In *filter feature selection* methods, the criterion for selecting or discarding features does not depend on any classifier. We estimate the mutual information (MI) between the features set and the class label: $I(\vec{S};\vec{C}) = H(\vec{S}) - H(\vec{S}|\vec{C})$. Here $\vec{S}$ is a matrix of size $m \times n$ and $\vec{C}$ of size $m \times 1$ where m is the number of samples and n the number of features of the feature subset. Traditionally the MI has been evaluated between a single feature and the class label. Here we calculate the MI using the entire set of features to select. This is an important advantage in FS, as the interactions between features are also taken into account.The entropies $H(\cdot)$ of a set with a large n number of features can be efficiently estimated using the k-NN-based method developed by Leonenko [Leonenko et al., 2008]. Thus, we take the data set with all its features and determine which feature to discard in order to produce the smallest decrease of $I(\vec{S_{n-1}};\vec{C})$. We then repeat the process for the features of the remaining feature set, until only one feature is left. A similar information-theoretic selection approach is described in detail in [Bonev et al., 2008].

In the present application each sample is originally a 3D object represented by a triangle mesh. From each 3D object, three types of graphs (Sec. 5.2) are extracted (labelled in the figures as a) *Sphere*, b) *Baricenter* and c) *Geodesic*). Only the structural graph information is used for classification. For each graph, 9 different measures (listed in the area plots in Figure 5.8) are calculated, as described in Sec. 5.3. They are transformed into histograms after normalizing them by the volume of the graph. Commute time is normalized twice, 1) linearly and 2) quadratically. Only the

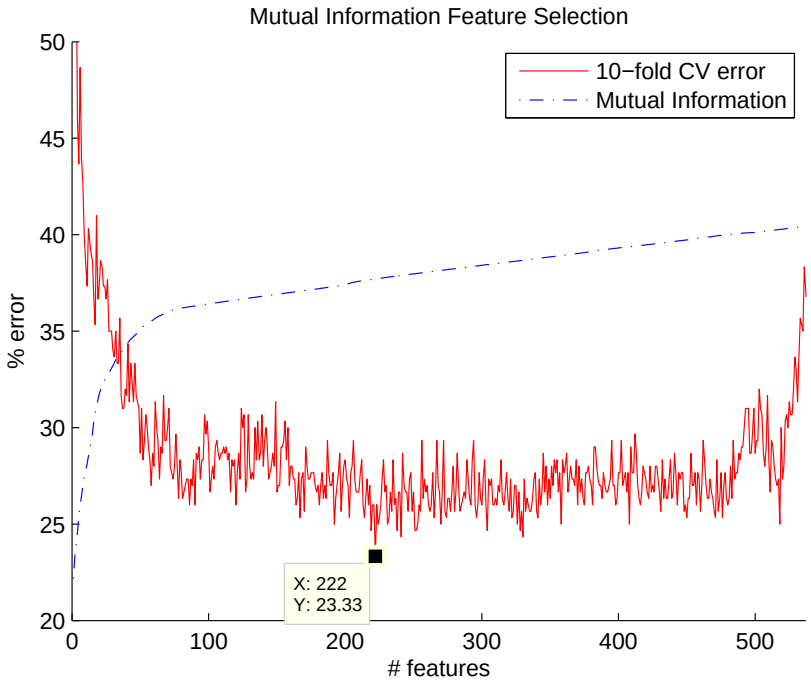

Figure 5.3: Mutual Information and 10-fold cross validation error for 1 to 540 selected features.

complexity flow curve is not histogrammed, for the sake of order preservation. Since there is no optimal way to select tthe number of bins, we perform several different binnings on each measure (2, 4, 6 and 8 bins). All histograms form a bag of features, of size $9 \cdot 3 \cdot 20 = 540$ features (see Figure 5.4). We let the FS process decide which binning from which measure and from which graph to discard.

5.5. Results

The experiments are performed on the pre-classified 3D shapes database [Attene and Biasotti, 2008]. It consists of 15 classes $\times$ 20 objects. Each one of the 300 samples is characterized by 540 features, and has a class label $l \in \{$*human, cup, glasses, airplane, chair, octopus, table, hand, fish, bird, spring, armadillo, buste, mechanic, four-leg*$\}$.

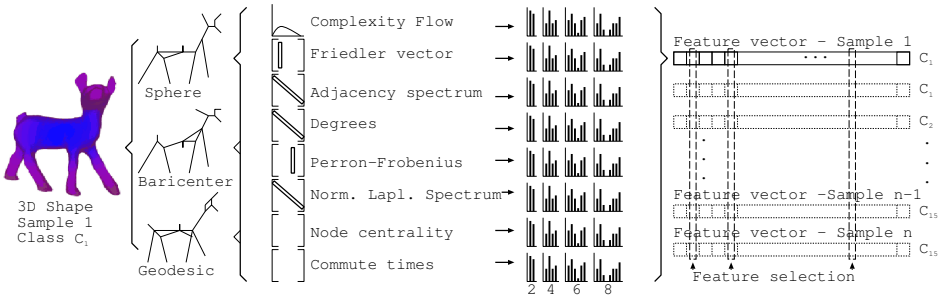

Figure 5.4: The process of extracting from the 3D object the three graph representations, unattributed graph features, histogram bins, and finally selecting the features.

5.5.1. Classification errors

The errors are measured by 10-fold cross validation (10-fold CV). In Figure 5.6 we show how MI is maximized as the number of selected features grows, and its relation to the decrease in error. The figure shows how a high number of features degrades the classification performance. For the 15-class problem, the optimal error $(23,3\%)$ is achieved with a set of 222 features. This error is lower for 10 classes $(15,5\%)$, 5 classes (6%) and 2 classes problems (0%). These results depend on the classifier used for measuring the error. However, the MI curve, as well as the selected features, do not depend on the classifier, as it is a purely information-theoretic measure.

5.5.2. Features analysis

Several different unattributed graph measures are used in this work. We aim to determine which measures are most important and in which combinations. In Figure 5.8 we show the evolution of the proportion of selected features. The coloured areas in the plot represent how much a feature is used with respect to the remaining ones (the height on the Y axis is arbitrary). For the 15-class experiment, in the feature sets smaller than 100 features, the most important is the Friedler vector, *in combination* with the remaining features. Commute time is also an important feature. Some features that are not relevant are the node centrality and the complexity flow. Turning our attention to the graphs type, all three appear relevant. In Figure 5.7 we show the proportion of features selected for the 222-feature set, which yielded the lowest error in our experiments. (The dashed vertical line in Figure 5.8 also shows the 222-feature set) In the plot representing the

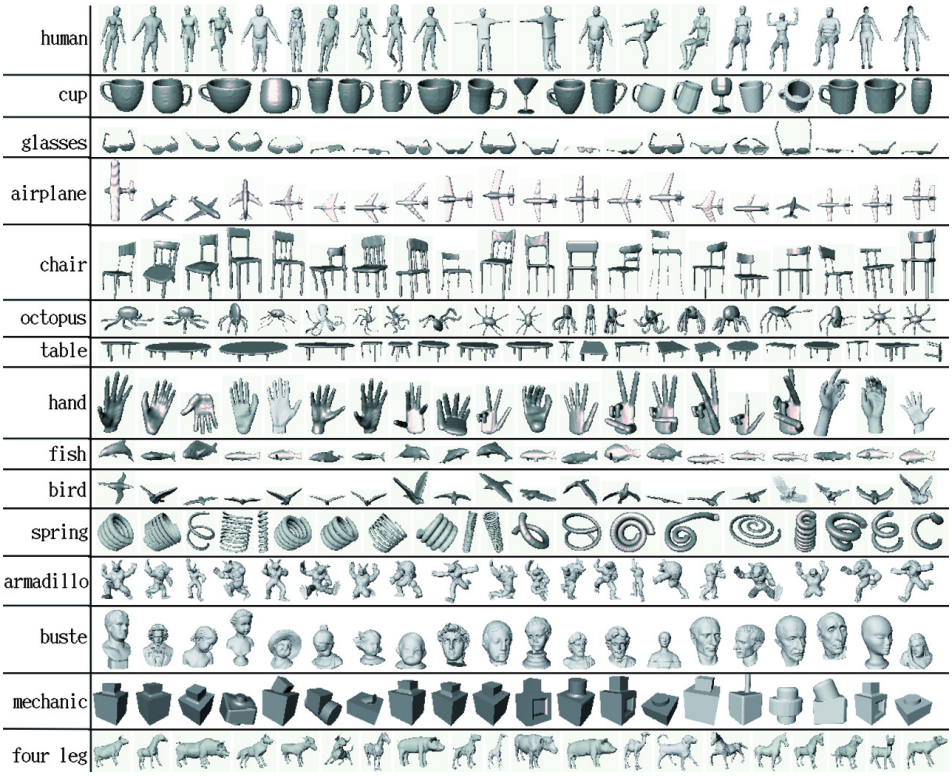

Figure 5.5: The 3D shapes database [Attene and Biasotti, 2008].

selected binnings we can see that the four different binnings of the features do have importance for graph characterization.

These conclusions concerning the relevance of each feature cannot be drawn without performing some additional experiments with different groups of graph classes. For this purpose in Figure 5.9 we present four different 3-class experiments. The classes share some structural similarities, for example the 3 classes of the first experiment have a head and limbs. Although in each experiment the minimum error is achieved with very different numbers of features, the participation of each feature is highly consistent with the 15-class experiment. The main difference among experiments (Figure 5.9) is that node centrality seems to be more important for discerning among elongated sharp objects. Although all three graph types are relevant, the *sphere graph* performs best for blob-shaped objects.

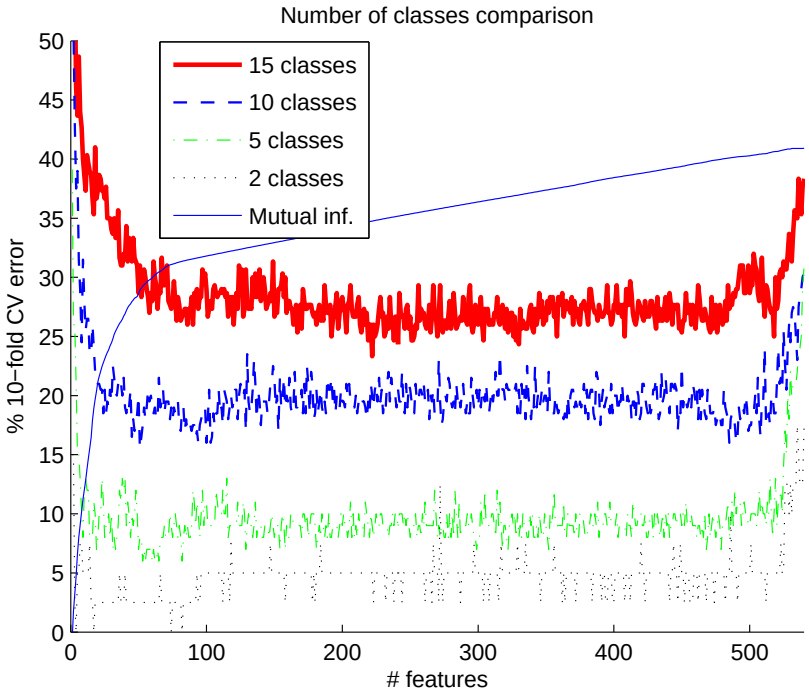

Figure 5.6: Classification errors.

5.5.3. The impact of feature selection

In the Figure 5.6 it is obvious that, if the number of features used for classification is too low, the error is higher, and on the other hand if the number of features is too high, the error could also rise. However, this depends on the order of features are added. In this work we use mutual information as the evaluation criterion because it is related to the minimization of the Bayesian error. What would happen if a worse criterion is used? To what extent the precision of the mutual information estimation is important? What is its impact on the final classification error?

Following we present some experiments which answer these questions. All of them refer to the 15-classes experiment. In order to vary the precision of the mutual information criterion we change the error bound ϵ of the ANN algorithm which is used for entropy estimation. ANN builds a kd-tree structure, whose cells are visited in increasing order of distance from the query point. A stop condition of the search algorithm occurs when the distance is colser than an error bound ϵ. This premature stop can save

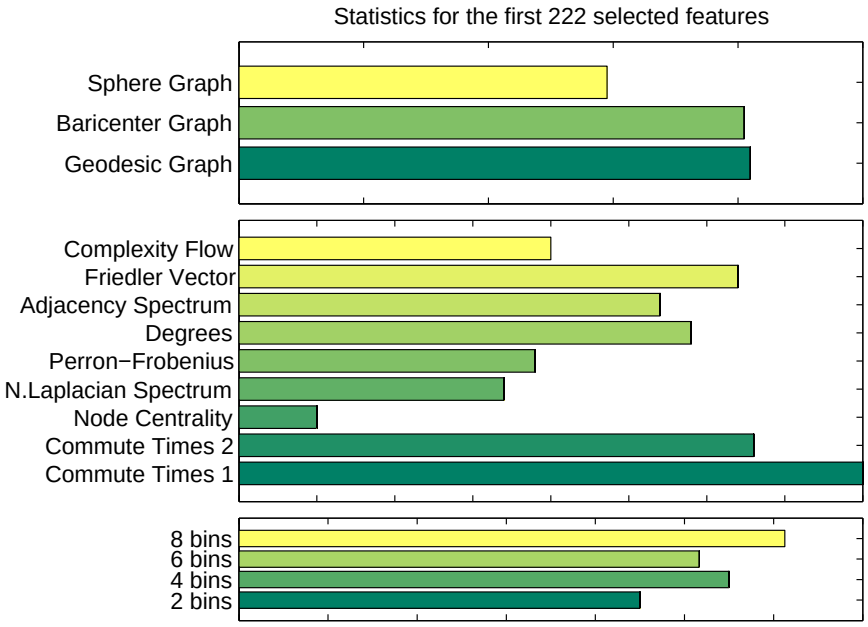

Figure 5.7: Statistics of the first 222 features selected in the 15-classes experiment.

computational time, as shown in the Figure 5.11[1]. It also causes a decrease in the precision of the k-NN computation. Thus, the entropy estimation, and so, the mutual information estimations, are degraded. To what extent? This is shown in Figure 5.10. It is interesting to see that the error bound $\epsilon = 0$ yields significantly better feature selection results, in terms of 10-fold Cross Validation error. Also, the increment of the error bound is not linear with respect to the increment of the 10-fold CV error.

The differences in the classification performance are due to small differences in the feature sets. For example, the difference among the feature sets yielded by $\epsilon = 0$ and $\epsilon = 1$ are significant (see Figure 5.12,-top-left). Then, before the error bound ϵ arrives the 0.5 value, the feature sets remain very similar. Other significant changes in the feature sets are plotted in Figure 5.12. Each one of the figures compares two different feature selection processes, as a consequence of different ϵ values. The first process

[1]The experiments were optimized for multiprocessors and they were run on an 1.6GHz Intel Centrino processor and DDR2 RAM.

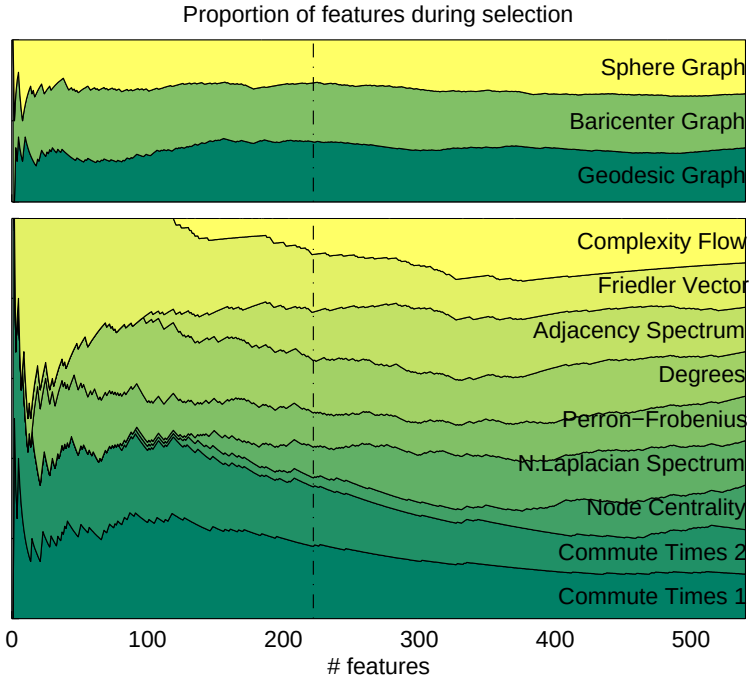

Figure 5.8: Area plot representing the proportion of features in the 15-classes experiment, for 1 to 540 features.

is represented as a coloured area plot, and the second one is represented with black solid lines.

The most important differences are observed in the early stage of feature selection (before the first 200 features are selected). After that, the proportion among the different features selected converges, because there are no more features left for selecting. However, the early stage of the selection process determines the maximum error which could be achieved, as shown in Figure 5.10: a good run ($\epsilon = 0$) yields an error plot which decreases to $23,3\%$, and after that increases to $37,33\%$. A run which yields poor results is the case of $\epsilon = 0.5$, for instance. In this case the error decreases progressively until achieving $37,33\%$, but none of the feature subsets produces a lower error.

It is also worth observing that the node centrality and the Friedler vector features are always important in the beginning of the process, disregarding the precision of the feature selection criterion. Shortly after the beginning, commute times start to play an important role. Regarding the reeb graph

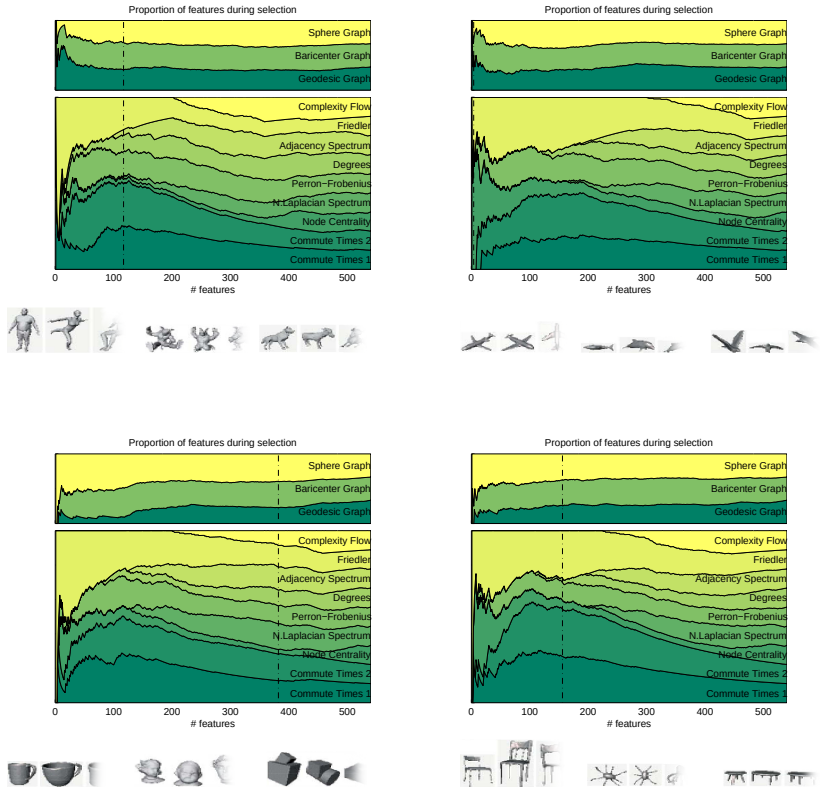

Figure 5.9: Feature Selection on 3-class experiments: Human/Armadillo/Four-legged, Aircraft/Fish/Bird, Cup/Bust/Mechanic, Chair/Octopus/Table.

types, most of the features are selected from the "sphere graph" type.

5.6. Conclusions

The contributions of this application to graph classification are twofold. First, it demonstrates the feasibility of multiclass classification based on purely structural spectral features. Secondly, an information-theoretic feature analysis suggests that similar features are selected for very different sets of objects. Moreover, the feature selection experiments show that even if the precision of the selection criterion is degraded, the most important features are still the same.

On the other hand this work demonstrates some important effects of feature selection. In the first place we prove that the precision of the

mutual information estimation has a great impact on the final classification performance. The same experiments show how very small changes in the order of the selected features can also affect the classification result.

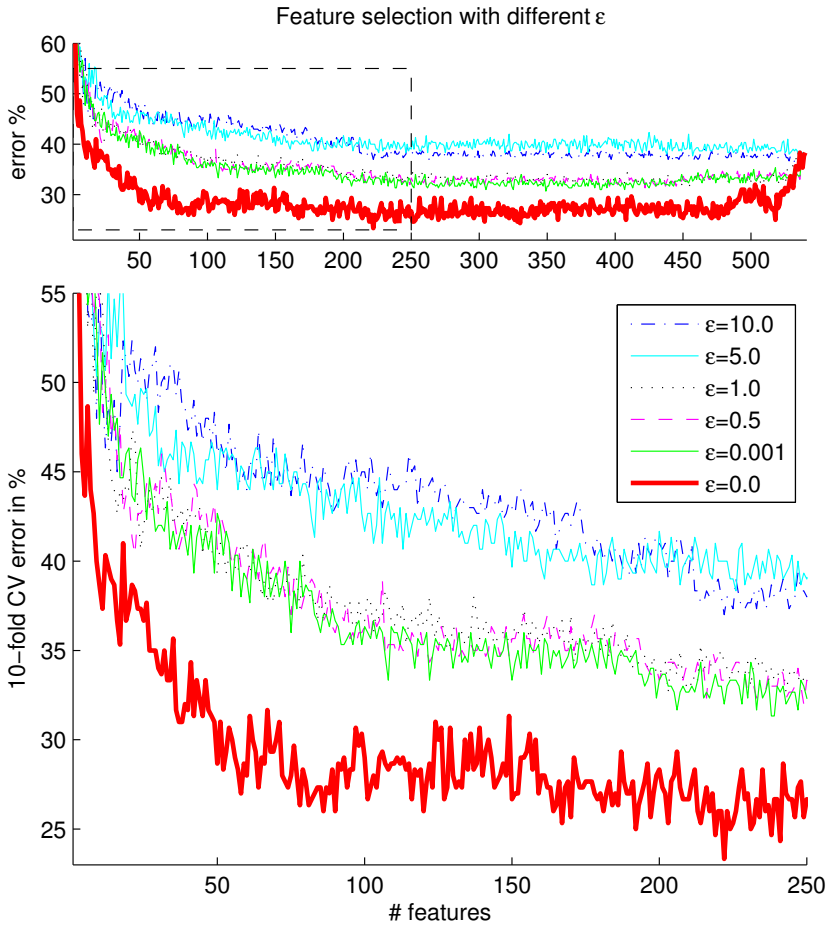

Figure 5.10: The 10-fold CV errors yielded by several feature selection runs, with different ANN error bound values (ϵ).

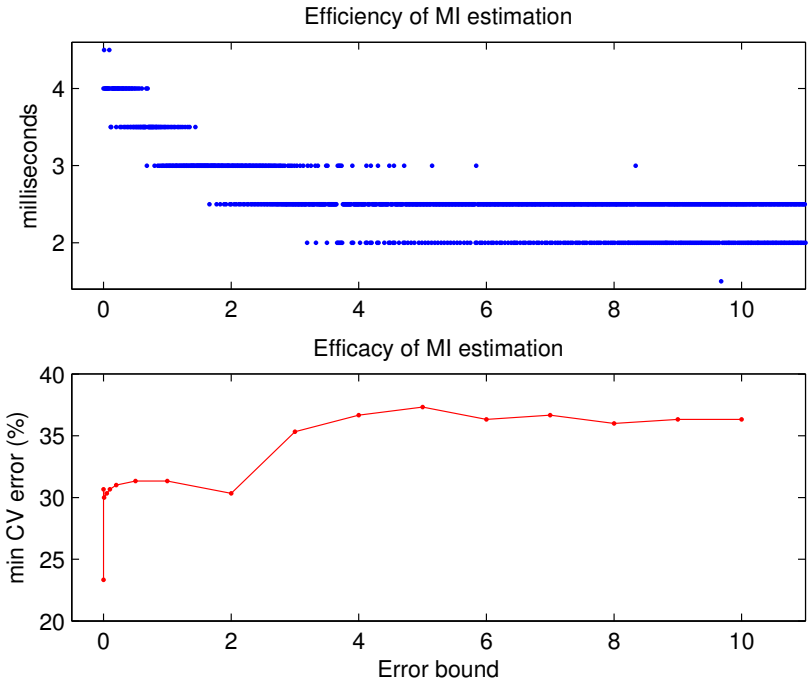

Figure 5.11: Top: the milliseconds it takes to evaluate the mutual information between the 300 samples with 540 features, and the 15 class labels. Bottom: the minimal errors achieved in the 15-class feature selection, for different Error bound (ϵ) values.

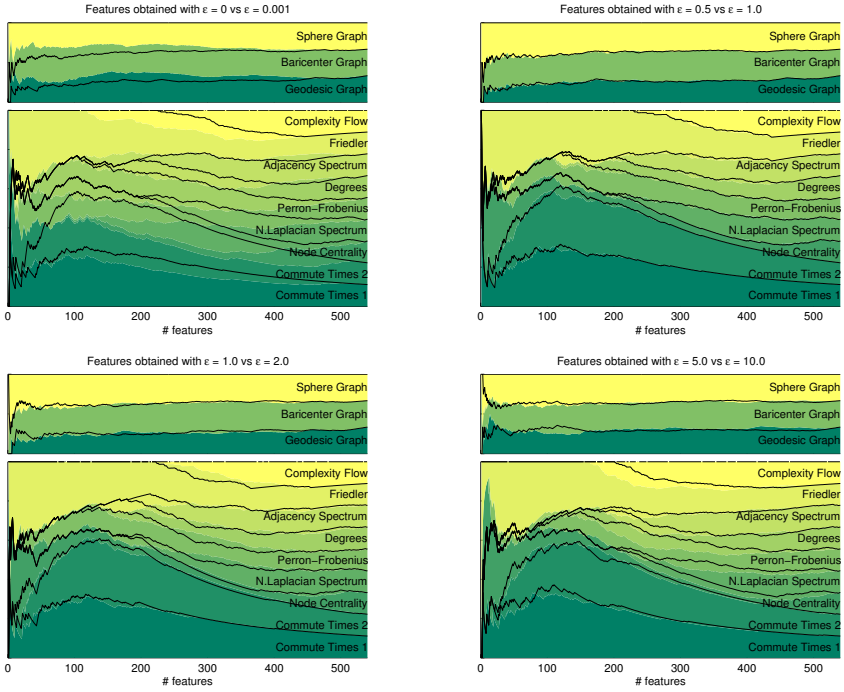

Figure 5.12: Four feature selection comparisons of different pairs of ϵ values. The first feature selection process is represented as a coloured area plot, while the second one is plotted with black solid lines.

Chapter 6

Conclusions and future work

6.1. Contributions

The main contributions of the thesis are stated in this section. In the first place we state the contributions related to the feature selection method. Secondly we enumerate some contributions of the method to other fields of pattern recognition.

6.1.1. Method

In this research we present a novel filter feature selection method based on information theory. Although the idea of using mutual information for evaluating the prediction power of features has been present in the literature for years [Guyon and Elisseeff, 2003], it has been used on separate features without the ability to capture their prediction power as a set. Some techniques have been designed to capture higher order interactions among features [Koller and Sahami, 1996]. In this thesis we study the efficient estimation of mutual information for evaluating entire sets of features. There are criteria that estimate mutual information incrementally [Peng et al., 2005], by adding one feature to the set in each iteration. Our formulation of the criterion does not depend on any particular search order in the feature space.

The estimation of mutual information is based on entropy estimation. The entropy estimators which do not estimate the distribution but are based on the distances among the samples [Leonenko et al., 2008], provide independence on the number of dimensions. This makes it possible to com-

pare the prediction power of sets of thousands of features. We present an analysis of non-parametric entropy estimation methods in order to compare estimators for different number of samples and dimensions of the data.

We show the importance of the precise estimation of mutual information when used as a feature selection criterion. Our analysis of different error bounds in the nearest neighbours calcula for entropy estimation shows that it has a big impact on the final classification errors. Although accepting an error bound for the nearest neighbours estimation [Mount and Arya, 1997] could save computational time when evaluating a feature set, our experiments show that this is not feasible and a zero error bound is preferable.

The search order also determines the success of a feature selection method. We do not propose a new search order but we show that, from an information-theoretical point of view, a greedy backward elimination search can preserve all the mutual information possible with the criterion we propose. On the contrary, a greedy forward search is not capable of guaranteeing that a set of selected features has the maximum information possible about the class, because a newly added feature could provide new information in combination with the already selected.

6.1.2. Applications

Feature selection for supervised classification plays an important role in many different fields related to pattern recognition. The applications we present in this thesis have several objectives. On the one hand we use them to show how the theory is successfully applied to real data which we use for experiments that show some of the important aspects of feature selection performance (classification error with different number of classes, number of samples, number of dimensions, number of histogram bins, precision of the estimation, computational time). On the other hand we use the applications for showing successful feature selection and classification on real-world high-dimensional data, and for comparing our results to other results published in the literature. With the structure classification experiment we show how feature selection can be used to explain the features in a classification problem and provide new information to the field of spectral graph classification.

6.1.2.1. Computer vision

The experiments on image data are mainly oriented to localization and mobile robotics tasks. Image-based classification with feature selection

proved to be appropriate for the visual localization task, with both orthographic and omnidirectional cameras. In our experiments we aim to extract a wide range of different visual filters which are applied to the whole image and their responses are histogrammed. Thus, we make the filter responses independent to the position on the image, an thus, to rotation and small offsets. Some of the filters are appropriate for a given task, and some others are not, and this is the motivation for using feature selection. This localization approach is very general, as it can be trained for any environment. However, its performance can vary depending on the environment and the number of classes. This approach is useful for coarse localization and it has proved to be useful for hierarchical localization systems [Bonev et al., 2007d, Escolano et al., 2007] where it provides context information for further fine-localization steps.

6.1.2.2. Gene microarrays

With the experiments on microarray data we prove the feasibility of evaluating high-dimensional patterns with the evaluation criterion we propose. The data of the microarrays comes from the genetic code of cancer or tumors of different types, extracted from both healthy and affected tissues of different patients. These gene selection experiments are performed on publicly available data sets in order to compare the classification performance with the results published in the literature. From these gene selection experiments we obtained promising results, which outperform most of the state-of-the-art classification performances [Bonev et al., 2008].

6.1.2.3. Structural graphs

In the literature of graph classification most algorithms rely on the information present in the attributes of the graph nodes. There is little work relying on purely structural features. Our experiments with spectral graph features proof the feasibility of classification based on purely structural spectral features. We successfully classify graphs extracted from 3D objects which are divided in 15 different classes. On the other hand the information-theoretic feature selection yields an analysis that suggests that similar features are selected for very different sets of objects. Thus, feature selection gives an insight into the use of spectral features in graph classification.

6.2. Limitations

We find two main limitations in the approach we present. One of them is related to the search order. There is also a complexity issue of the evaluation criterion which is worth to be remarked.

6.2.1. Search order

The main contribution of this thesis is focused on the evaluation criterion. However, a major limitation is the greedy character of the search order we use. We showed examples in which the greedy forward selection falls into local minima. Greedy backward elimination can better preserve the information about the class, but the local minima are still present and the resulting feature set is not optimal.

6.2.2. Computationally complex cases

There are two complexity issues. The first is about the estimation of mutual information, which is involved into the evaluation criterion based on mutual information. We pointed out that the complexity of the non-parametric estimations methods that we discuss does not depend on the number of dimensions. It only depends on them insomuch distances among samples have to be calculated. However, the entropy estimation does depend on the number of samples. The complexity is due to the use of the k-nearest neighbour algorithm which, in this case, has an $n \log n$ complexity for building it, and $\log n$ complexity in the queries, where n is the number of samples. Therefore it can be seen that if too many samples are present in the data set, their evaluation could take a prohibitive computational time. In most of our experiments we have been working with data sets which have less than 1000 samples.

It is interesting to note that depending on the number of classes of the data set, the computational time is different. For the mutual information estimation we use the conditional entropy which involves a sum of the entropies of the samples belonging to each class: $I(\vec{S}; C) = H(\vec{S}) - \sum H(\vec{S}|C = c)p(C = c)$. Then, if we have the samples divided in many classes, the second term consists of the weighted sum of the entropies of many small-sized sets of samples. On the contrary, if there are only two classes, the sum consists of the entropies of two larger sets of samples and this has a higher complexity than the multiclass case. The first term, which estimates the entropy of the whole set of samples, is present in both cases.

Another complexity issue is related to the search order. We state that this approach is capable of selecting features from high-dimensional data sets. However, if there were, for instance, one million features, even though the evaluation criterion would be capable of evaluating the mutual information in this dimensionality, the greedy search order would force us to test the addition or elimination of each one of them, that is, one million evaluations for adding or removing the first feature. The next feature would take one evaluation less, the next two less, and so on, until the last feature to be decided would take just one evaluation.

Related to this complexity issue, there is a decision which has to be taken for some large data sets. If the data set doesn't have a very large number of features (of the order of thousand), we recommend to use greedy backward elimination, as we already reasoned. However, if the number of features is very high (of the order of ten thousand or more), it is not feasible to start a backward elimination process because it takes many iterations to eliminate all the features. In this case it would be feasible to start a forward feature selection because in many problems a small number of features would be enough for a plausible classification. This way it would not be necessary to wait for all the features to be added.

6.3. Future work

Based on the results presented in this thesis, we are planning to continue our research on several topics.

6.3.1. Search order

As we already discussed the search order is one of the limitations of this work. Feature selection researchers have proposed a number of heuristics for search orders which escape local minima. However, we believe that further research on this topic is necessary. The suitability of a search order often depends on the evaluation criterion used. We want to research on this conjecture.

6.3.2. Hierarchical selection

The method presented in this thesis will be significantly faster if the feature set is divided in several smaller feature sets. Traditionally many feature selection methods have assumed independence among features, or

among small feature sets. The presented method performs well with high-dimensional feature sets. For sets with many thousands of features few partitions would be enough in order to decrease computational time. These experiments are not under the focus of this thesis and are left for the future work.

6.3.3. Stopping criterion

This is another topic which is not covered by the presented research. In our experiments we always take as the best feature set the one which yields the minimum classification error, disregarding its size. This kind of criterion is classifier-dependent and requires evaluating the complete range of feature set sizes in order to know which is the best performing one. This strategy is actually feasible because the greedy search order has a definite number of iterations. However there exist search orders which need a stopping criterion in order to finish the search. Information theory might provide a good theoretical framework for developing a reliable stopping criterion.

6.3.4. Generative and discriminative learning

Traditionally there have been two schools of thought in machine learning: the generative and the discriminative learning. Generative methods learn models which "explain" the data and are more suitable for introducing conditional independence assumptions, priors, and other parameters. Contrarily, discriminative learning methods only learn from data to make accurate predictions [Jebara, 2004]. The combination of both is becoming a widely explored topic in machine learning. The feature selection method presented in this thesis is a purely discriminative method. We plan to work on feature selection methods which combine the generative and discriminative approaches. We want to explore the combination of bottom-up and top-down estimation processes [Tu and Zhu, 2002]. This kind of mutual feedback has proved to function in the human visual perception [Ullman, 2000]. Information theory offers a number of tools (maximum entropy discrimination, minimum description length, divergence measures), which have contributed to important advances in the research on these topics.

Appendix A

Entropy of the Gaussian distribution

From among the distributions which cover the entire real line $\mathbb{R}$ and have a finite mean μ and a finite variance σ^2, the maximum entropy distribution is the Gaussian. Following we present the analytical proof, using optimization under constraints. The Lagrange multipliers method offers quite a straightforward way for finding the Gaussian probability distribution.

The function to maximize is the Shannon entropy of a pdf $f(x)$:

$$H(f) = -\int_{-\infty}^{\infty} f(x) \log f(x) dx \tag{A.1}$$

with respect to $f(x)$, under the constraints that

a) $f(x)$ is a pdf, that is, the probabilities of all x sum 1:

$$\int_{-\infty}^{\infty} f(x) dx = 1, \tag{A.2}$$

b) it has a finite mean μ:

$$\int_{-\infty}^{\infty} x f(x) dx = \mu \tag{A.3}$$

c) and a finite variance σ^2:

$$\int_{-\infty}^{\infty} x^2 f(x) dx - \mu^2 = \sigma^2 \tag{A.4}$$

The Lagrangian function under these constraints is:

$$\begin{aligned}\Lambda(f,\lambda_0,\lambda_1,\lambda_2) &= -\int_{-\infty}^{\infty} f(x)\log f(x)dx && \text{(A.5)}\\ &+ \lambda_0\left(\int_{-\infty}^{\infty} f(x)dx - 1\right) && \text{(A.6)}\\ &+ \lambda_1\left(\int_{-\infty}^{\infty} xf(x)dx - \mu\right) && \text{(A.7)}\\ &+ \lambda_2\left(\int_{-\infty}^{\infty} x^2 f(x)dx - \mu^2 - \sigma^2\right) && \text{(A.8)}\end{aligned}$$

The critical values of Λ are zero-gradient points, that is, when the partial derivatives of Λ are zero:

$$\begin{aligned}\frac{\partial\Lambda(f,\lambda_0,\lambda_1,\lambda_2)}{\partial f(x)} &= -\log f(x) - 1 + \lambda_0 + \lambda_1 x + \lambda_2 x^2 = 0 && \text{(A.9)}\\ \frac{\partial\Lambda(f,\lambda_0,\lambda_1,\lambda_2)}{\partial\lambda_0} &= \int_{-\infty}^{\infty} f(x)dx - 1 = 0 && \text{(A.10)}\\ \frac{\partial\Lambda(f,\lambda_0,\lambda_1,\lambda_2)}{\partial\lambda_1} &= \int_{-\infty}^{\infty} xf(x)dx - \mu = 0 && \text{(A.11)}\\ \frac{\partial\Lambda(f,\lambda_0,\lambda_1,\lambda_2)}{\partial\lambda_1} &= \int_{-\infty}^{\infty} x^2 f(x)dx - \mu^2 - \sigma^2 = 0 && \text{(A.12)}\end{aligned}$$

These equations form a system from which $\lambda_0, \lambda_1, \lambda_2$ and, more importantly, $f(x)$, can be calculated. From Eq. A.9:

$$f(x) = e^{\lambda_2 x^2 + \lambda_1 x + \lambda_0 - 1} \tag{A.13}$$

Now $f(x)$ can be substituted in Eqs. A.10, A.11, A.12:

$$\int_{-\infty}^{\infty} e^{\lambda_2 x^2 + \lambda_1 x + \lambda_0 - 1}dx = 0 \tag{A.14}$$

$$\int_{-\infty}^{\infty} xe^{\lambda_2 x^2 + \lambda_1 x + \lambda_0 -}dx = \mu \tag{A.15}$$

$$\int_{-\infty}^{\infty} x^2 e^{\lambda_2 x^2 + \lambda_1 x + \lambda_0 - 1}dx = \mu^2 + \sigma^2 \tag{A.16}$$

For solving the latter system of three equations with three unknowns $(\lambda_0, \lambda_1, \lambda_2)$, the three improper integrals have to be computed. In 1778 Laplace proved that

$$I = \int_{-\infty}^{\infty} e^{-x^2}dx = \sqrt{\pi}. \tag{A.17}$$

This result is obtained by switching to polar coordinates:

$$I^2 = \left(\int_{-\infty}^{\infty} e^{-x^2} dx\right)^2 = \int_{-\infty}^{\infty} e^{-x^2} dx \int_{-\infty}^{\infty} e^{-y^2} dy \tag{A.18}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-(x^2+y^2)} dxdy \tag{A.19}$$

$$= \int_{0}^{\infty}\int_{0}^{2\pi} e^{-r^2} r dr d\theta \tag{A.20}$$

$$= 2\pi\left[-\frac{1}{2}e^{-r^2}\right]_0^{\infty} = \pi \tag{A.21}$$

In Eq. A.18,the differential $dxdy$ represents an element of area in Cartesian coordinates in the xy-plane. This is represented in polar coordinates r, θ which are given by $x = r\cos\theta$, $y = r\sin\theta$ and $r^2 = x^2 + y^2$. The element of area turns into $rdrd\theta$. The 2π factor in Eq. A.21 comes from the integration over θ. The integral over r can be calculated by substitution, $u = r^2$, $du = 2rdr$.

The more general integral of $x^n e^{-ax^2+bx+c}$ has the following closed form [Weisstein, 1998]:

$$\int_{-\infty}^{\infty} x^n e^{-ax^2+bx+c} dx = \sqrt{\frac{\pi}{a}} e^{b^2/(4a)+c} \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{n!}{k!(n-2k)!} \frac{(2b)^{n-2k}}{(4a)^{n-k}} \tag{A.22}$$

for integer $n > 0$, the variables a, b belonging to the punctured plane (the complex plane with the origin 0 removed), and the real part of a being positive.

Provided the previous result, the system of equations becomes:

$$\sqrt{\frac{\pi}{-\lambda_2}} e^{-\lambda_1^2/(4\lambda_2)+\lambda_0-1} = 1 \tag{A.23}$$

$$\sqrt{\frac{\pi}{-\lambda_2}} e^{-\lambda_1^2/(4\lambda_2)+\lambda_0-1} \frac{-\lambda_1}{2\lambda_2} = \mu \tag{A.24}$$

$$\sqrt{\frac{\pi}{-\lambda_2}} e^{-\lambda_1^2/(4\lambda_2)+\lambda_0-1} \left(\frac{\lambda_1^2}{4\lambda_2^2} - \frac{1}{2\lambda_2}\right) = \mu^2 + \sigma^2 \tag{A.25}$$

By applying the natural logarithm to the equations,

$$\log e^{-\lambda_1^2/(4\lambda_2)+\lambda_0-1+\log\sqrt{-\pi/\lambda_2}} = \log 1, \tag{A.26}$$

the unknown λ_0 can be isolated from each equation of the system:

$$\lambda_0 = \frac{\lambda_1^2}{4\lambda_2} + 1 - \log\sqrt{\frac{\pi}{-\lambda_2}} \tag{A.27}$$

$$\lambda_0 = \frac{\lambda_1^2}{4\lambda_2} + 1 - \log\sqrt{\frac{\pi}{-\lambda_2}} + \log\frac{-2\mu\lambda_2}{\lambda_1} \tag{A.28}$$

$$\lambda_0 = \frac{\lambda_1^2}{4\lambda_2} + 1 - \log\sqrt{\frac{\pi}{-\lambda_2}} + \log\frac{\mu^2+\sigma^2}{\frac{\lambda_1^2}{4\lambda_2^2} - \frac{1}{2\lambda_2}} \tag{A.29}$$

$$\tag{A.30}$$

From which it is deduced that λ_1 and λ_2 have the following relation:

$$0 = \log\frac{-2\mu\lambda_2}{\lambda_1} = \log\frac{\mu^2+\sigma^2}{\frac{\lambda_1^2}{4\lambda_2^2} - \frac{1}{2\lambda_2}}, \tag{A.31}$$

then,

$$\lambda_1 = -2\mu\lambda_2 \tag{A.32}$$

$$\mu^2 + \sigma^2 = \frac{\lambda_1^2}{4\lambda_2^2} - \frac{1}{2\lambda_2} \tag{A.33}$$

provides the values of λ_1 and λ_2 and by subsituting them in any equation of the system, λ_0 is also obtain. The result is

$$\lambda_0 = -\frac{1}{2}\frac{\mu^2}{\sigma^2} - \log\sqrt{2\pi\sigma^2} + 1 \tag{A.34}$$

$$\lambda_1 = \frac{\mu}{\sigma^2} \tag{A.35}$$

$$\lambda_2 = -\frac{1}{2\sigma^2} \tag{A.36}$$

These solutions can be substituted in the $f(x)$ expression (Eq. A.13):

$$p(x) = e^{-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x + \frac{1}{2}\frac{\mu^2}{\sigma^2} - \log\sqrt{2\pi\sigma^2}} \tag{A.37}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\mu^2 - 2\mu x + x^2)} = \tag{A.38}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mu-\sigma)^2}{2\sigma^2}}, \tag{A.39}$$

which, indeed, is the pdf of the Gaussian distribution. Finally, the negative sign of the second derivative of Λ ensures that the obtained solution is a

maximum, and not a minimum or an inflection point.

$$\frac{\partial^2 \Lambda(f, \lambda_0, \lambda_1, \lambda_2)}{\partial f(x)^2} = -\frac{1}{f(x)} \tag{A.40}$$

Appendix B

Implementation details

B.1. Introduction

Most of the experiments in this thesis were performed in Matlab due to its plotting facilities and mathematical tools. However, in order to optimize the computational time, part of the software was implemented in C++, namely the forward/backward feature selection based on mutual information estimation with the Leonenko estimator of entropy.

The source code is available for downloading at

http://www.rvg.ua.es/ITinCVPR

under the GNU Lesser General Public License, Version 2.1 (LGPLv2). The program accepts as an input a matrix of real values representing the feature values of all the samples, and the class labels. The output is the feature selected in each iteration.

The binary can be run by command line. Its syntax is the following:

```
./lmifs [options] [resultlog.txt] < data.txt
Options:
  -K int        : The number of K nearest neighbours, default 5.
  -E float      : The error bound for ANN, default 0.
  -V true/false: Verbose output on screen within each iteration.
  -D f/b        : Direction of the FS search: (f)orward or
                  (b)ackward (default).
  -H            : This help.
```

```
data.txt format: a text file with three integers in the first
                 line and a matrix of Nx(D+1) floats:
                 N D C
                 n11 n12 ... n1D c1
                 ...
                 nN1 nN2 ... nND cN
                 with N number of samples, D dimensions and
                 C classes where c1..cN contain C different
                 classes in ascendant order.

Output in resultlog.txt:
  #F   selectedF   bestMI
  ...
  (one line for each #F)
```

The code can be compiled with a standard C++ compiler. We work with the GNU GCC 4.4 compiler.

B.2. Nearest neighbours

The main source of computational load are the calculi of the nearest neighbours. For this purpose we use the open source library Approximate Nearest Neighbours (ANN) [Mount and Arya, 1997]. Its algorithm constructs a kd-tree by dividing the space of the samples. The cells are visited in increasing order of distance from the query point. The search can be stopped when a maximum error bound is reached, in order to save computational time. However, we proved that maximum precision is important for our algorithm, so we use the error bound $\epsilon = 0$. The ANN library is also very efficient when queried with the maximum precision.

We needed to modify the code of the ANN Version 1.1.1 in order to adapt the code for parallelization, as explained in the following section.

B.3. Parallelization support

One of the characteristics of many feature selection algorithms is the possibility for parallelization. When several feature sets have to be evaluated, the evaluations usually are independent. Therefore they can be performed in parallel.

Our code is adapted for parallelization and can be compiled both for multiprocessor and monoprocessor machines. The parallelization standard that we use is OpenMP.

OpenMP is a portable, scalable model that gives shared-memory parallel programmers a simple and flexible interface for developing parallel applications. The OpenMP Application Program Interface (API) supports multi-platform shared-memory parallel programming in C/C++ and Fortran on all architectures, including Unix platforms and Windows NT platforms.

Our code checks during preprocessing time whether the OpenMP support is required and includes or not the OpenMP headers. The definition of the parallel code sections is also done with precompiler directives. Thus, the code can be successfully compiled both with compilers which support OpenMP and those which do not. The GNU GCC compiler supports OpenMP since the version 4.2.

At each iteration of the feature selection a number of feature sets have to be evaluated by estimating the mutual information. This consists of a number of iterations in a `for` loop. The OpenMP model can decide how to parallelize these iterations, depending on the number of processors, on their use, and depending on the time each iterations takes. This is done by placing the directive:

```
#pragma omp parallel for
for( ; ; ){ }
```

before the definition of the loop. The code which goes inside the loops has to be adapted for parallel computing. In this case the estimation of entropy is involved, which includes calls to the ANN library. We had to modify its code because the version 1.1.1 uses some global variables. In the presence of them, the parallel executions would be inconsistent because several different calls would try to overwrite and use the same variables.

With the OpenMP support enabled the efficiency of the program depends on the number of processors. For example, in an Intel Centrino with two processors, the computational time is almost twice faster.

B.4. Calculus optimization

The operations which most times are repeaded in the code are those related to the entropy estimation of Leonenko. As detailed in Subsec-

tion 2.5.5, the calculus consist of:

$$\hat{H}_{N,k,1} = \frac{1}{N}\sum_{i=1}^{N} \log \xi_{N,i,k}, \tag{B.1}$$

with

$$\xi_{N,i,k} = (N-1)e^{-\Psi(k)} V_d (\rho^{(i)}_{k,N-1})^d, \tag{B.2}$$

and

$$V_d = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}, \tag{B.3}$$

where $\Psi(k)$ is the digamma function and $\Gamma(\cdot)$ is the gamma function, which are already present in the C/C++ mathematical libraries.

The previous formulae can be rewritten using the logarithm properties in order to minimize the number of operations. In our code we use a lookup table which we fill in at the beginning of the execution because it only depends on the data set. It is of size $N \times D$, the number of samples and the number of dimensions. Each cell $L_{n,d}$ of this table has the pre-calculated value of:

$$L_{n,d} = \log\left((n-1)e^{-\psi(k)}\right) + \frac{d}{2}\log\pi - \log\Gamma\left(\frac{m}{2}+1\right) \tag{B.4}$$

Thus when estimating entropy, the distance to the k-th nearest neighbour is weighted by $L_{n,d}$, accordingly to Leonenko's formulation.

Appendix C

Publications

The developments and results of this Ph.D. thesis are published in some journals as listed below. The work on this thesis also generated some of the contents of a monograph devoted to information theory. This book has a chapter devoted to feature selection. It also discusses the entropy estimation topic in several chapters. Finally, our methods have been successfully applied in several pattern recognition tasks published in the international journals and conferences listed below.

C.1. Journals

- B. Bonev, F. Escolano and M. Cazorla (2008) [Bonev et al., 2008]
  "Feature Selection, Mutual Information, and the Classification of High-Dimensional Patterns"
  Pattern Analysis and Applications, Special Issue on Non-parametric Distance-based Classification Techniques and their Applications (Springer). February 2008

  In this work we present the forward feature selection approach, with mutual information estimation based on minimal spanning trees. We present experiments on both natural images and gene microarrays.

- M.A. Lozano, F. Escolano, B. Bonev, P.Suau, W. Aguilar, J.M. Saez and M.A. Cazorla (2008) [Lozano et al., 2008]
  "Region and constellations based categorization of images with unsupervised graph learning"
  Image and Vision Computing, November 2008

In this work feature selection plays an important role. Unsupervised graph learning is not applicable to large datasets. Feature selection is used for a coarse categorization which is the first step in the presented system, while graph learning is used for fine categorization. Some of the experiments of this work are performed in the domain of visual localization.

- B. Bonev, M. Cazorla and F. Escolano (2007) [Bonev et al., 2007a]
  "Robot Navigation Behaviors based on Omnidirectional Vision and Information Theory"
  Journal of Physical Agents, September 2007

  The autonomous navigation of a robot, based only on visual information, was developed for providing autonomy to the application presented in Chapter 3. Information theory is used for detecting the direction of the navigation in semi-structured environments.

C.2. Books

- F. Escolano, P. Suau and B. Bonev (2009) [Escolano et al., 2009b]
  "Information Theory in Computer Vision and Pattern Recognition"
  Springer: Computer Imaging, Vision, Pattern Recognition and Graphics. Hardcover, 420 pages.

  This text introduces and explores measures, principles, thoeries and entropy estimators from Information Theory underlying modern Computer Vision and Pattern Recognition algorithms. Several topics of the book are closely related to this thesis. The Feature Selection Chapter explains part of the contributions of this thesis, as well as other state-of-the-art methods which exploit information theory.

C.3. International conferences

- B. Bonev, F. Escolano, D. Giorgi, S. Biasotti (2010)
  "Information-theoretic Feature Selection from Unattributed Graphs"
  International Conference on Pattern Recognition, Istanbul, Turkey, August 2010

  This publication is the outcome of the graph classification application of this thesis. The backward feature selection method with the k-nn entropy estimator are exploited.

- B. Bonev, F. Escolano, D. Giorgi, S. Biasotti (2010)
  "High-dimensional Spectral Feature Selection for 3D Object Recognition based on Reeb Graphs"
  Statistical, Structural and Syntactic Pattern Recognition, Cezme, Turkey, August 2010

  The contribution of this work is the experimental study of the k-nn error bound, the precision of the mutual information estimation, and its impact on the final classification result after feature selection.

- B. Bonev, F. Escolano and M. Cazorla (2007) [Bonev et al., 2007c]
  "A novel Information Theory method for Filter Feature Selection"
  Mexican International Conference on Artificial Intelligence,Aguascalientes, Mexico, November 2007.

  The backward feature elimination method is exploited here for image classification and microarray selection.

- F. Escolano, B. Bonev, P. Suau, W. Aguilar, Y. Frauel, J.M. Saez and M. Cazorla (2007) [Escolano et al., 2007]
  "Contextual visual localization: cascaded submap classification, optimized saliency detection, and fast view matching"
  IEEE International Conference on Intelligent Robots and Systems. San Diego, California, USA, October 2007

  Feature selection plays role in a minimal-complexity classifier which performs submap classification for a coarse localization of a mobile robot. The outcome of this classification restricts the domain of an optimized saliency detector which exploits visual statistics of the submap. Finally, a fast view-matching algorithm filters the initial matchings with a structural criterion, for fine localization.

- B. Bonev, F. Escolano, M.A. Lozano, P. Suau, M.A. Cazorla and W. Aguilar (2007) [Bonev et al., 2007d]
  "Constellations and the Unsupervised Learning of Graphs"
  6th IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition. Alicante, Spain. June 2007

  In this work feature selection is used for natural images classification and is compared to structural matching methods.

C.4. National conferences

- B. Bonev , M. Cazorla (2008) [Bonev and Cazorla, 2008]
  "Large Scale Environment Partitionaing in Mobile Robotics Recognition Tasks"
  IX Workshop de Agentes Físicos, Vigo, Spain, September 2008

  We use information-theoretic methods for visual localization. In large sequences of images classification may not work, because the visual appearance of the images may present similarities along the sequence. We partition the sequence by using the Jensen-Rényi divergence. Once partitioned feature selection is used for producing different classifiers for each subsequence. Finally, in order to deal with the kidnapped robot problem, an algorithm keeps several hypotheses about the location, and as the robot moves, they converge to a single one.

- B. Bonev , M. Cazorla (2006) [Bonev and Cazorla, 2006b]
  "Towards Autonomous Adaptation in Visual Tasks"
  VII Workshop de Agentes Físicos, Las Palmas de Gran Canaria, Spain, 2006

  This work presents the idea of using a large set of low-level filters for image classification, and using feature selection for adapting the filters set to some particular image classification problem. The experiments are performed on indoor and outdoor data sets for mobile robot localization.

C.5. Projects

This thesis has been developed during the participation in several projects.

- MAP3D - Mapping with mobile robots using computer vision techniques (MCYT: TIC2002-02792, 2003-2005)

  One of the aspects of this project was the need for efficient classification of images. Here we exploited the idea of using a set of low-level filters for characterizing the images, and then performing feature selection for improving the classification performance. We used an omnidirectional camera in both indoor and outdoor environments.

- OUTSLAM: Simultaneous Localization and Mapping (SLAM) in outdoor environments via Computer Vision and Cooperative Robotics (MEC: DPI2005-01280, 2005-2008)

  The research presented in this thesis contributed in two ways to this project. On the one hand, feature selection was performed on colour images and stereo images for classification and localization. On the other hand, the visual navigation method was developed for tackling the problem of outdoor navigation.

- HLSLAM: Hybrid metric-topological SLAM integrating computer vision and 3D LASER for large environments (MCI: TIN2008-04416/TIN)

  Feature research also contributed to this project in the domain of topological localization and mapping. We used feature selection for detecting natural landmarks. We also used information-theoretic measures and divergences in order to find suitable partitionings of the environment. This thesis contributed in the direction of both coarse and fine localization in submaps.

Apéndice D

Resumen

D.1. Selección de características basada en teoría de la información

En esta tesis doctoral[1] presentamos nuestra investigación en *selección de características* para problemas de *clasificación supervisada*. Proponemos un método computacionalmente eficiente, cuya eficacia se prueba tanto analítica como experimentalmente. El método está basado en la *teoría de la información*, que ofrece un sólido marco teórico en el campo del *reconocimiento de patrones*.

Hoy en día los avances en aprendizaje automático y en la adquisición de datos exigen procesar datos compuestos por miles de características. Un ejemplo es el proceso de *microarrays*. La selección de características es un campo de *aprendizaje automático y reconocimiento de patrones* y consiste en la reducción de la dimensionalidad de los datos. Esto se consigue eliminando las características redundantes, irrelevantes o ruidosas para la tarea de clasificación. Hay dos tipos de clasificación: la supervisada y la no supervisada. Este trabajo está enfocado hacia la clasificación supervisada.

En clasificación supervisada el clasificador se construye a partir de un conjunto de ejemplos de entrenamiento y las etiquetas de sus correspondientes clases. El clasificador debe adivinar la clase de los ejemplos de test. En base a ese resultado se puede medir el error de clasificación. Hay dos factores que influyen en éste: el método de clasificación y el conjunto de

[1] Esta traducción a castellano es un resumen de la tesis doctoral. Para ver en detalle la metodología y experimentación, véase el resto del documento con la versión original en inglés.

características que definen los ejemplos.

Una vez propuesto un método de selección de características y presentadas una serie de pruebas empíricas y teóricas de su eficacia, en esta tesis presentamos un conjunto de aplicaciones a problemas reales de reconocimiento de patrones. En primer lugar aplicamos la selección de características a un problema de visión artificial enmarcado en una tarea de robótica móvil. En segundo lugar probamos nuestro método en microarrays de datos genéticos para la clasificación y detección de enfermedades de cáncer. Con este experimento probamos datos de muy alta dimensionalidad, del orden de miles de características. Finalmente presentamos un experimento con datos estructurales. Se trata de grafos obtenidos a partir de formas tridimensionales. El tipo de características que extraemos son principalmente las espectrales, y se basan puramente en información de estructura. No usar información de atributos en los grafos es todo un desafío. La mayoría de los trabajos que clasifican grafos con éxito, utilizan atributos, aparte de la estructura. Aparte de abordar con éxito esta tarea de clasificación estructural, la selección de características nos permite realizar un detallado análisis de qué características importan más que otras, y bajo qué condiciones.

D.1.1. Motivación y objetivos

La selección de características está presente en el campo del reconocimiento de patrones desde principios de los años 70. Desde entonces han sido un problema desafiante para los investigadores, debido a la complejidad computacional del problema. La importancia de la selección de características no consiste sólo en maximizar los porcentajes de acierto de la clasificación. En muchos campos la selección de características se utiliza para explicar los datos. Hay datos con un elevado número de características cuya función y relación con la clase no se conoce a priori por los expertos de dichos campos. Este es el caso de muchos datos biológicos. Por tanto, cualquier mejora en el campo de la selección de características representa un avance importante en reconocimiento de patrones y las implicaciones llegan a diversos campos de la ciencia.

Conforme aumenta la capacidad de las máquinas de computación, la dimensionalidad de los datos también, por su lado, es cada vez mayor. Sin embargo el análisis de todas las posibles combinaciones de los datos no es posible, independientemente del incremento de la capacidad computacional de las máquinas actuales. Los desarrollos matemáticos de los últimos años ofrecen la posibilidad de tratar el problema de la alta dimensionalidad de

maneras eficientes. El objetivo de esta tesis es capturar las interacciones entre todas las variables seleccionadas, de una manera eficiente. Se pretende conseguir un criterio de evaluación mejor que los existentes para cuantificar en qué medida un conjunto de características es mejor que otro para dado problema de clasificación. Todo este procesamiento debe realizarse en un tiempo computacional viable.

D.2. Criterio basado en *información mutua*

Los criterios de selección de características supervisada se dividen principalmente en tres tipos: de *filtro*, de *envoltorio* e *incrustados*. Los criterios de envoltorio consisten en evaluar conjuntos de características construyendo clasificadores y midiendo sus tasas de acierto. Por otro lado los criterios incrustados se denominan así por formar parte en el propio algoritmo de clasificación, es decir, las características se seleccionan durante la construcción del clasificador. Ambos criterios son dependientes de determinado clasificador y el conjunto de características seleccionado es más apropiado para dicho clasificador, y no para otro. Por otro lado, el *sobreaprendizaje* cometido puede ser mayor que con las técnicas de filtro [Guyon and Elisseeff, 2003]. Las técnicas basadas en filtro consisten en aplicar una serie de tests estadísticos sobre las características, sin utilizar ningún clasificador determinado. La clasificación es posterior a la selección con filtro.

El criterio óptimo para evaluar un conjunto de datos de cara a un problema de clasificación sería el error Bayesiano cometido:

$$E(S) = \int_{\vec{S}} p(\vec{S}) \left(1 - \max_i(p(c_i|\vec{S}))\right) d\vec{S}, \tag{D.1}$$

donde $\vec{S}$ es el vector de características seleccionadas y $c_i \in C$ es una clase de entre todas las clases C presentes en el conjunto de datos.

Debido a que la función máx$(\cdot)$ no es lineal, el criterio de minimizar el error Bayesiano no es una función viable. En la literatura hay diferentes cotas del error Bayesiano. Una cota superior obtenida por Hellman y Raviv (1970) es:

$$E(\vec{S}) \leq \frac{H(C|\vec{S})}{2}$$

Esta cota está relacionada con la información mutua porque ésta última puede expresarse como:

$$I(\vec{S}; C) = H(C) - H(C|\vec{S})$$

y $H(\vec{C})$ es la entropía de las etiquetas de las clases, que no depende del espacio de características $\vec{S}$. Por tanto la maximización de la información mutua es equivalente a la maximización de la cota superior del error Bayesiano. Por otro lado también existe una cota inferior, obtenida por Fano (1961), que también está relacionada con la información mutua.

Por otro lado la relación de la información mutua con la divergencia de Kullback-Leibler también justifica su uso como criterio de selección de características. Dicha divergencia se define:

$$KL(P||Q) = \sum_{\vec{x}} p(\vec{x}) \log \frac{p(\vec{x})}{q(\vec{x})}$$

para el caso discreto. Dada la definición de información mutua, y dado que la entropía condicional puede expresarse como $p(x|y) = p(x,y)/p(y)$, tenemos que:

$$\begin{aligned} I(X;Y) &= \sum_{y\in Y}\sum_{x\in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} && \text{(D.2)} \\ &= \sum_{y\in Y} p(y) \sum_{x\in X} p(x|y) \log \frac{p(x|y)}{p(x)} && \text{(D.3)} \\ &= \sum_{y\in Y} p(y) KL\left(p(x|y)||p(x)\right) && \text{(D.4)} \\ &= E_Y\left(KL\left(p(x|y)||p(x)\right)\right) && \text{(D.5)} \end{aligned}$$

Por tanto maximizar la información mutua también es equivalente a maximizar la *esperanza* de la divergencia de Kullback-Leibler entre las densidades $P(\vec{S}|\vec{C})$ y $P(\vec{S})$. En otras palabras, la densidad de todas las clases debe ser todo lo distante posible de la densidad de cada una de las clases en el conjunto de características. La maximización de la información mutua ofrece un equilibrio entre maximización de la discriminación y minimización de la redundancia.

El criterio Max-Dependency (MD) que proponemos utilizar consiste en maximizar la información mutua entre el conjunto de características $\vec{S}$ y las etiquetas de la clase C:

$$\max_{\vec{S}\subseteq\vec{F}} I(\vec{S};C) \qquad \text{(D.6)}$$

Así, la m-ésima característica es seleccionada según el criterio:

$$\max_{x_j\in\vec{F}-\vec{S}_{m-1}} I(\vec{S}_{m-1}, x_j; C) \qquad \text{(D.7)}$$

En [Peng et al., 2005] se presenta un criterio que, cuando se aplica a un orden de selección incremental (empezar desde un conjunto vacío de características e ir añadiendo de una en una) es equivalente al anterior. Es el criterio *minimum redundancy - maximum relevance* (mRMR) que para la selección de la m-ésima característica se define:

$$\max_{x_j \in \vec{F}-\vec{S}_{m-1}} \left[I(x_j;C) - \frac{1}{m-1} \sum_{x_i \in \vec{S}_{m-1}} I(x_j;x_i) \right] \tag{D.8}$$

Este criterio sólo puede ser utilizado en un algoritmo voraz y que añada características de una en una. El criterio MD que nosotros proponemos, por el contrario, puede evaluar conjuntos de características independientemente del orden de selección ya que no funciona de forma incremental, sino que evalúa cada conjunto de características independientemente de los anteriores.

Hay algunas dificultades a la hora de aplicar este criterio en la práctica. Estimar la información mutua en un espacio continuo de miles de características no es trivial. En la mayoría de los problemas la información mutua se estima entre una característica y la clase, pero no con un conjunto de características completo. En esta tesis utilizamos métodos de estimación de entropía que no se basan en estimación de densidades, sino en grafos entrópicos y en vecinos próximos, para estimar la entropía de un conjunto de datos. A partir de esa entropía calculamos la información mutua. En [Bonev et al., 2008] la entropía se calcula usando *minimal spanning trees* (MSTs) [Hero and Michel, 2002]. Este tipo de estimación es apropiada para datos de alta dimensionalidad y un pequeño número de ejemplos, ya que su complejidad depende del número de ejemplos n_s, siendo la complejidad $O(n_s \log(n_s))$ independiente del número de dimensiones. La información mutua puede calcularse de dos maneras: a partir de la entropía condicional o a partir de la entropía conjunta:

$$\begin{aligned} I(\vec{S};C) &= \sum_{x \in \vec{S}} \sum_{c \in \vec{C}} p(x,c) log \frac{p(s,c)}{p(x)p(c)} && \text{(D.9)} \\ &= H(\vec{S}) - H(\vec{S}|C) && \text{(D.10)} \\ &= H(\vec{S}) + H(C) - H(\vec{S},C), && \text{(D.11)} \end{aligned}$$

donde x es una característica del conjunto de características seleccionadas $\vec{S}$ y c es la etiqueta de la clase perteneciente al conjunto de clases o prototipos C.

En el presente enfoque la información mutua se calcula en base a la entropía condicional, $H(\vec{S}) - H(\vec{S}|\vec{C})$. Para ello primero deben estimarse las entropías $\sum H(X|C = c)p(C = c)$, cálculo que es factible ya que $\vec{C}$ es una variable discreta.

D.3. Estimación de entropía

La entropía es un concepto básico en la teoría de la información [Cover and Thomas, 1991b]. Está relacionada con la predecibilidad, y lleva a varias interpretaciones posibles. Una de ellas es que la entropía mide la cantidad de información de la que nos provee un evento. Una interpretación relacionada es que la entropía mide la incertidumbre del resultado de un evento. En este sentido, un evento muy común aporta menor entropía que muchos eventos diferentes, pero equiprobables. Otra interpretación es que la entropía mide la dispersión en una distribución de probabilidades. Así, una imagen con muchos colores diferentes tiene un histograma más disperso (más entrópico) que una imagen con pocos colores, o con un rango de colores bien diferenciado. Algunas otras medidas estadísticas también pueden ser útiles para cuantificar la dispersión, como la *kurtosis*, que mide cómo de "picuda" es una distribución de probabilidades.

Hay diferentes definiciones matemáticas de entropía. La famosa entropía de Shannon se define, para una variable discreta $\vec{Y}$ con un conjunto de valores $y_1, ..., y_N$, como:

$$\begin{aligned} H(\vec{Y}) &= -E_y[\log(p(\vec{Y}))] \\ &= -\sum_{i=1}^{N} p(\vec{Y} = y_i) \log p(\vec{Y} = y_i), \end{aligned} \tag{D.12}$$

Una de las generalizaciones de la entropía de Shannon es la entropía de Rényi's, también conocida como α-entropía:

$$H_\alpha(\vec{Y}) = \frac{1}{1-\alpha} \log \sum_{i=1}^{n} y_j^\alpha, \tag{D.13}$$

con el valor $\alpha \in (0, 1) \cup (1, \infty)$. La entropía de Rényi tiende a la de Shannon cuando α tiende a 1. Sin embargo cuando $\alpha = 1$, la anterior ecuación tiene una discontinuidad debido a la división por 0. Se puede demostrar, tanto analítica como experimentalmente, que cuando α tiende a 1, H_α tiende al valor de la entropía de Shannon. Por otro lado la discontinuidad marca

un cambio de cóncava a convexa en la función de α-entropía. Este cambio está ilustrado en la figura 2.15.

La estimación de entropía es crucial en los problemas de reconocimiento de patrones. La estimación de la entropía de Shannon ha sido muy estudiada en el pasado [Paninski, 2003, Viola and Wells-III, 1995, Viola et al., 1996, Hyvarinen and Oja, 2000, Wolpert and Wolf, 1995]. Los estimadores de entropía se dividen en dos tipos principales: los que estiman primero una distribución, y los que evitan tener que estimarla, conocidos como métodos *bypass*. Los estimadores basados en estimación de distribuciones sufren el problema de dimensionalidad en presencia de pocos ejemplos o bien de muchas dimensiones. Un ejemplo de ellos son las ventanas de Parzen [Viola and Wells-III, 1997]. Un ejemplo de método bypass es la estimación realizada por [Hero and Michel, 2002]. Ellos estiman la entropía a partir de la altura de un minimal spanning tree (MST). Para un espacio d-dimensional con $d > 2$, el estimador de α-entropía

$$H_\alpha(X_n) = \frac{d}{\gamma}\left[\ln \frac{L_\gamma(X_n)}{n^\alpha} - \ln \beta_{L_\gamma,d}\right] \tag{D.14}$$

es asintóticamente consistente con la función de densidad de probabilidad. Aquí la función $L_\gamma(X_n)$ es la altura del MST y γ depende del orden α y de la dimensionalidad: $\alpha = (d-\gamma)/d$. La corrección del bias (o sesgo) $\beta_{L_\gamma,d}$ depende del criterio de minimización de grafos utilizado, pero es independiente de la función de densidad de la probabilidad. Este bias se puede acotar a través de una aproximación para d grandes: $(\gamma/2)\ln(d/(2\pi e))$ [Bertsimas and Ryzin, 1990]. El valor de la entropía de Shannon se puede aproximar a patir de valores de la entropía de Rényi, como hacen en [nalver et al., 2009].

Otro método más eficiente computacionalmente, y que directamente estima la entropía de Shannon sin tener que pasar por la de Rényi, es el desarrollado por [Kozachenko and Leonenko, 1987]. Está basado en vecinos próximos o k-nearest neighbours (k-NN), y el estimador se puede expresar como

$$\hat{H}(X) = -\psi(k) + \psi(N) + \log \frac{V_d}{2^d} + \frac{d}{N}\sum_{i=1}^{N} \log \epsilon_i, \tag{D.15}$$

donde $\epsilon_i = 2||x_i - x_j||$ es el doble de la distancia entre el ejemplo x_i y su k-NN x_j. En las figuras 2.21, 2.22, 2.23, 2.24 y 2.25 se presenta un estudio comparativo de las estimaciones de entropía y sus errores. Como conclusión, el uso del estimador de entropía basado en vecinos próximos es

el más adecuado para datos de alta dimensionalidad, a la vez que también sirve para datos de pocas dimensiones.

D.4. Orden de búsqueda

El orden de búsqueda es el orden en el que se recorre el espacio de características. Un recorrido exhaustivo del espacio de características es inviable debido a que el número de combinaciones de características crece exponencialmente. Así, en la literatura hay diferentes métodos, desde *voraces* y de *enfriamiento simulado*, hasta algoritmos que añaden y quitan características debiendo garantizar la tendencia hacia una convergencia, como por ejemplo los algoritmos genéticos. Los experimentos presentados en esta tesis utilizan un recorrido voraz. Hay dos formas de realizarlo: hacia delante y hacia atrás. La primera forma consiste en empezar desde el conjunto vacío e ir añadiendo características, mientras que la segunda consiste en empezar desde el conjunto completo de características e ir eliminándolas.

La principal limitación de los algoritmos voraces es la caída en mínimos locales. Los mínimos locales se deben a que hay conjuntos de características que por separado no tienen mucha información sobre la clase, pero en conjunto sí. Así, lo que puede ocurrir es que se seleccionen otras características previamente en lugar de éstas. Otro caso que puede darse es el de seleccionar una característica que tiene mucha información sobre la clase, sin contar con que alguna de las siguientes características aportaría todavía más, pero ésta última deja de seleccionarse debido a redundancia con la ya seleccionada.

Aunque las búsquedas voraces caen en mínimos locales, es posible alcanzar la máxima información mutua posible para determinado número de características, utilizando el recorrido voraz hacia atrás. Aún así, este conjunto de características normalmente sería subóptimo, ya que no sería el mínimo posible. Hay dos tipos de características que se pueden eliminar: las irrelevantes y las redundantes. Si una característica simplemente es irrelevante para la etiqueta de la clase, puede ser eliminada del conjunto de características y ello no tendría ningún impacto sobre la información mutua entre el resto de características y la clase.

Qué ocurre cuando una característica x_i se elimina por ser redundante dadas las demás características seleccionadas: ¿se pueden seguir eliminando características? Para comprobarlo vamos a utilizar la regla de la cadena de la información mutua. Supongamos que eliminamos una característica x_i del conjunto $\vec{F}_n$ de n características porque ésta no provee información

adicional acerca de la clase, dadas el reso de características $\vec{F}_{n-1}$. Entonces quitamos otra característica $x_{i'}$ porque, una vez más, ésta no aporta información acerca de la clase, dado $\vec{F}_{n-2}$. En este caso la eliminada anteriormente, x_i, no podrá hacerse necesaria, ni siquiera tras la eliminación de x_{i-1}. Según la regla de la cadena, tenemos que para un conjunto multidimensional de características,

$$I(\vec{F};C) = I(x_1,\ldots,x_n;C) = \sum_{i=1}^{n} I(x_i;C|x_{i-1},x_{i-2},\ldots,x_1) \qquad \text{(D.16)}$$

la información mutua entre las características y la clase se puede descomponer en el sumatorio de informaciones mutuas condicionales entre una característica x_i y la clase, dadas el resto de características. Un ejemplo con sólo 4 características quedaría así:

$$\begin{aligned} I(x_1,x_2,x_3,x_4;C) \quad = \quad & I(x_1;C) + \\ & I(x_2;C|x_1) + \\ & I(x_3;C|x_1,x2) + \\ & I(x_4;C|x_1,x_2,x_3) \end{aligned}$$

Si se decide eliminar x_4 es porque no aporta información acerca de la clase C dadas el resto de características, es decir, que $I(x_4;C|x_1,x_2,x_3) = 0$. Una vez eliminada, se puede ver que x_4 no aparece en el resto de los términos, de manera no habría inconveniente en que otra, por ejemplo x_3, fuera eliminada si se diera que $I(x_3;C|x_1,x2) = 0$.

Este método conservaría la máxima información posible entre el conjunto de características y la clase. Por otro lado, al ser voraz no asegura que el conjunto de características obtenido sea el mínimo. En la figura 2.26 se ilustra un ejemplo de 4 características con un diagrama de Venn. En el ejemplo la característica x_4 puede ser eliminada porque $I(x_4;C|x_1,x_2,x_3) = 0$. En realidad esta característica no sólo es irrelevante dadas el resto de características, sino que es irrelevante por si sola, ya que $I(x_4;C) = 0$. La siguiente característica que podría ser eliminada es, o bien x_1, x_2, o x_3, porque tenemos que $I(x_1;C|x_2,x_3) = 0$, $I(x_2;C|x_1,x_3) = 0$ y $I(x_3;C|x_1,x_2) = 0$. En tal situación la búsqueda voraz podría decidir cualquiera de ellas indistintamente. Véase la figura 2.27, donde se ve que si se toma x_3, la búsqueda cae en un mínimo local porque después ni x_1, ni x_2 pueden ser eliminadas sin perder información acerca de la clase. Sin embargo si en lugar de eliminar x_3, se hubiera eliminado alguna de las otras dos, el conjunto de características final sería x_3, que es el conjunto de características mínimo para este ejemplo.

El conjunto de datos "Corral" [John et al., 1994] es adecuado para ver la diferencia entre la búsqueda hacia atrás y hacia delante cuando el criterio es la información mutua. En este conjunto de datos hay 6 características binarias, $\{x_1, x_2, x_3, x_4, x_5, x_6\}$. La etiqueta de la clase también es binaria y es el resultado de la operación

$$C = (x_1 \wedge x_2) \vee (x_3 \wedge x_4).$$

Por tanto x_1, x_2, x_3, y x_4 determinan por completo la etiqueta de la clase, C. La característica x_5 es irrelevante y x_6 es una característica que tiene alta correlación (del 75 %) con la etiqueta de la clase. Algunos ejemplos del conjunto de datos son:

x_1	x_2	x_3	x_4	x_5	x_6	C
0	1	1	0	1	0	0
0	1	1	1	1	1	1
1	0	0	0	0	1	0
1	0	1	1	0	1	1

Muchos métodos de selección de características, y particularmente aquellos que realizan una búsqueda voraz hacia delante, primero seleccionan la característica que tiene alta correlación, decisión que no es la correcta. Por el contrario, cuando se evalúa la información mutua en una búsqueda voraz hacia atrás, las primeras características en descartar son las irrelevantes y también las que no aporten información en presencia de las demás, a pesar de que estén correlacionadas. Así, el resultado es que quedan seleccionadas las cuatro primeras características. En la práctica las estimaciones de la información mutua no son perfectas. Además los conjuntos de datos no suelen tener ejemplos suficientes como para definir perfectamente la distribución de características. Así, más que buscar un decremento nulo en la información mutua, lo que se hace es permitir el decremento mínimo con el fin de mantener la máxima información posible acerca de la clase.

D.5. Aplicaciones

En esta tesis aportamos tres aplicaciones de selección de características. La generalidad del problema nos permite aplicarlo a campos muy distintos, como son la visión artificial, la genética y la clasificación de formas tridimensionales. Cada aplicación tiene un propósito distinto. La de visión artificial enseña cómo diseñar un problema de extracción y selección de características desde cero, en función de la información de la que se puede

disponer, y experimenta con el método obteniendo unos resultados prácticos enmarcados en una tarea de robótica móvil. La aplicación a selección de genes tiene como propósito compararse con otros métodos del estado del arte, así como experimentar con datos de muy alta dimensionalidad, del orden de miles de características. Finalmente en la aplicación de clasificación de formas tridimensionales se utiliza información estructural en forma de grafos, y las características extraídas de estos grafos se componen de un conjunto de características espectrales y de complejidad. En este caso se consigue resolver un problema muy complicado y poco tratado en la literatura, que es la clasificación basada en información puramente estructural, es decir, clasificación de grafos sin atributos en los nodos ni en las aristas. Por otro lado esta aplicación tiene como propósito utilizar la selección de características no sólo para optimizar porcentajes de acierto, sino también explorar qué características espectrales juegan un papel más importante en la clasificación de grafos y en qué condiciones de clasificación. Este tipo de estudio aporta información novedosa al campo de clasificación de estructuras.

D.5.1. Clasificación de imágenes

Este experimento está enmarcado en una tarea de robótica móvil. Los robots móviles están cada vez más presentes, tanto en entornos industriales y laboratorios, como en otro tipo de entornos menos controlados, como museos, el hogar, o incluso entornos de exterior. Dos de los problemas primordiales en un robot móvil son la localización y la navegación. La localización indica al robot en qué posición del entorno se encuentra, y es necesaria para proporcionarle autonomía y funcionalidad en muchas de las tareas del robot. Hay dos tipos de localización: localización de grano fino, que suele ser métrica, y localización de grano grueso, que está más orientada a reconocer entornos o balizas naturales. La localización de grano grueso se puede utilizar para identificar el tipo de entorno en el que se encuentra el robot, para una posterior localización más fina dentro de este entorno, como se hace en [Bonev et al., 2007d]. La localización por GPS es una alternativa que no descarta el uso de visión, ya que el GPS no funciona en todos los instantes de tiempo, además de no ser operativo en entornos de interior. El método de localización que proponemos está descrito en detalle en [Bonev and Cazorla, 2006a]. Para completar la aplicación también se ha implementado un sencillo método de navegación del robot que sirve para evitar obstáculos, y que se explica en [Bonev et al., 2007a].

Los experimentos presentados en esta aplicación han sido probados en

diferentes robots móviles (ActivMedia PowerBot, Magellan Pro, Evolution Robotics ER-1, figura 3.1). La cámara utilizada es de tipo omnidireccional. Algunos experimentos se realizaron con cámara fotográfica Canon PowerShot y un espejo omnidireccional montado en el objetivo. Otros se realizaron con la cámara Flea2 de GreyPoint y el espejo omnidireccional de la figura 3.2. La resolución de las imágenes para localización fue de 400×400 píxeles, y para navegación de 600×600. La resolución temporal es de 10Hz para la navegación, mientras que el cálculo de localización requiere entre 100 y 1000 milisegundos, dependiendo de la cantidad de características previamente seleccionadas.

Los entornos utilizados para tomar las imágenes de entrenamiento y de test están representados en la figura 3.6. Se trata de un entorno de interiores y un entorno de exteriores, el primero dividido en 16 clases y el segundo dividido en 8 clases.

D.5.1.1. Extracción de características

La estrategia de extracción de características está gráficamente descrita en la figura 3.7: la idea es aplicar filtros básicos a partes de la imagen. Los filtros son los siguientes:

- Nitzberg
- Canny
- Gradiente horizontal
- Gradiente vertical
- Magnitud del gradiente
- 12 filtros de color $H_i,\ 1 \leq i \leq 12$

Los gradientes se aplican a la imagen omnidireccional rectificada en modo de imagen panorámica, para proveerlos de invarianza a la rotación. Los filtros de color H_i se aplican a un espacio HSV de colores donde la i indica una posición en el valor de la H, dividida a intervalos regulares, desde 1 hasta 12. El resto de los filtros se aplican a partes concéntricas de la imagen, como se ilustra en la figura. La idea de los anillos concéntricos está basada en la *transformation ring projection* [Tang et al., 1991]. El objetivo de tal representación es hacer diferencia entre los filtros que se aplican a zonas de suelo, zona de proximidad inmediata, zonas de proximidad media y zonas de horizonte/cielo. Una aplicados los filtros a cada una de las 4 zonas de la imagen, sus respuestas de deben contabilizar en histogramas, para aportar invarianza a su posición x,y en la imagen. Las imágenes omnidireccionales pueden tomarse con distintas orientaciones con respecto al norte

y sur (aunque siempre orientadas hacia el suelo). Los histogramas nos permiten independizarlas de la rotación y de pequeños desplazamientos. Así, el número total N_F de filtros sería

$$N_F = C * K * (B - 1) \tag{D.17}$$

donde C es el número de anillos (4), K es el número de filtros (17) y B es el número de bins de los histogramas.

D.5.1.2. Selección de características

Los resultados de la selección de características se muestran en las tablas Las tres mejores características para los experimentos de 8 clases de interior son Canny, Filtro de color 1, y Filtro de color 5, todas ellas aplicadas al anillo 2 (proximidad inmediata). El *error de validación cruzada* (CV) producido con tan sólo 3 características es del $24,52\,\%$, mucho mejor que el error de $30,57\,\%$ producido por el conjunto completo de características. Para los experimentos de exteriores las mejores características resultantes fueron el Filtro de color 9 sobre el anillo 2, y Filtro de color 8 y Nitzberg sobre el anillo 3 (proximidad media), lo cual tiene sentido puesto que en exteriores las distancias son mayores. En conjuntos de características de mayor cardinalidad normalmente las características seleccionadas pertenecen a los anillos 2 y 3. El 1 y el 4 cuentan con pocas características seleccionadas. Por otro lado se da el fenómeno de que cuando un bin de una característica es seleccionado, es muy probable que el resto de bins del histograma de dicha característica también sean seleccionados. Otro fenómeno interesante es que en algunos casos Nitzberg, Canny los Gradientes Horizontal y Vertical, así como su Magnitud, se seleccionan todas juntas (a pesar de que son, hasta cierto punto, redundantes en conjunto). Para evitar eso, se debe trabajar con conjuntos de característica de menor cardinalidad. Siempre que hemos seleccionado conjuntos de sólo tres características, dos de ellas han sido filtros de color y una ha sido un detector de aristas.

Otro experimento que hemos realizado consiste en tomar imágenes a lo largo de una trayectoria mixta, de interior y exterior, y comparar las imágenes de test con las ya obtenidas durante el entrenamiento. Esta comparación se hace en un espacio de características previamente seleccionado. Las imágenes pueden verse en las figuras 3.8 y 3.11. En esta última se han representado los vecinos próximos de varias de las imágenes de test. El resultado puede ser observado en la figura 3.12 y en la tabla 3.3. Por otro lado también se ha realizado un estudio comparativo de diferentes criterios de selección, y los resultados están explicados en la figura 3.13. También

es interesante ver la diferencia entre los problemas con diferentes números de clase. Como se puede intuir, a mayor número de clases, peor resultado de clasificación. Este efecto está experimentalmente ilustrado en la figura 3.10. Por último, uno de los parámetros que también se debe justificar es el número de bins utilizados. En la figura 3.9 se realiza una comparativa de la cual se deduce que 4 ó 6 bins son cantidades convenientes, en cuanto a mínimos errores de clasificación alcanzados.

D.5.2. Selección de genes

Los avances en tecnología de *microarrays* de expresiones genéticas permiten medir la expresión de decenas de miles de genes en cada experimento. Los investigadores buscan marcadores moleculares que se puedan explotar para diagnosticar, pronosticar y predecir el cáncer. Ellos usan los datos de la expresión genética de ejemplos de múltiples individuos y múltiples tejodos y tipos de tumor. Además los estudios de microarrays están revolucionando los conocimientos de los mecanismos moleculares subyacentes en los procesos biológicos.

En muchas bases de datos de microarrays los genes que se almacenan son un conjunto preseleccionado. Estos genes no se seleccionan automáticamente sino que son fruto de la decisión humana, en base a experiencias previas y rangos de genes que se conocen como relevantes para determinado estudio. Aún así el número de genes incluidos en las bases de datos es considerable, del orden de decenas de miles. Esta alta dimensionalidad convierte el análisis de los datos en todo un desafío para el campo del reconocimiento de patrones. Estudios en selección de características de los últimos años [Wang and Gotoh, 2009] lo califican como "un problema intratable".

Para ilustrar la aplicabilidad de nuestro método a datos de alta dimensionalidad hemos realizado una serie de experimentos en conjuntos de datos conocidos y disponibles públicamente. El objetivo es identificar pequeños conjuntos de genes con buen rendimiento de predicción acerca de la etiqueta, en este caso, el tipo de cáncer o tumor. Muchos de los métodos tradicionales seleccionan los genes en función de su poder de predicción individual. Este tipo de enfoques son eficientes para datos de alta dimensionalidad pero no pueden descubrir las interacciones entre genes y la redundancia. La aportación de nuestro método es la evaluación eficiente de conjuntos enteros de características.

En primer lugar evaluamos nuestro criterio de selección en los experimentos de la famosa base de datos NCI (National Cancer Institute). Es-

tos datos consisten en 60 ejemplos de 60 pacientes, donde cada uno de ellos está definido por 6380 genes, que son la dimensionalidad de las características. Los 60 ejemplos están etiquetados con 14 clases diferentes de tipos de cáncer o de tumores. En la figura 4.3 están representados el error de validación cruzada *leave one out* (LOOCV) junto con el incremento de la información mutua. El error decrece hasta que se seleccionan 39 características, y después poco a poco se incrementa, debido a que se añaden características ruidosas. En la figura 4.3 comparamos los resultados de selección de los dos criterios basados en información mutua que presentamos en esta tesis, y podemos ver que los errores son muy similares, debido a que bastantes de los genes seleccionados coinciden. Este efecto se puede observar mejor en la figura 4.5, donde están ilustrados los niveles de expresión de genes seleccionados por los métodos. En la figura 4.4 se comparan ambos criterios con el criterio mRMR. Los niveles de expresión de los genes seleccionados están representados en la figura 4.6.

D.5.2.1. Comparación con otros resultados

En [Jirapech-Umpai and Aitken, 2005] se utiliza un algoritmo evolutivo para selección de características y el mejor error LOOCV alcanzado es del 23,77 % con un conjunto de 30 características. En nuestros experimentos alcanzamos un 10,94 % de error con 39 características. Además de los experimentos con los datos del NCI, hemos realizado otros experimentos en otros cuatro conjuntos de datos público que pueden ser descargados de Broad Institute `http://www.broad.mit.edu/`, Stanford Genomic Resources `http://genome-www.stanford.edu/`, y Princeton University `http://microarray.princeton.edu/`.

- Leukemia: El conjunto de entrenamiento consiste en 38 ejemplos etiquetados en dos clases, 27 Acute Lymphoblastic Leukemia (ALL) y 11 Acute Myeloid Leukemia (AML), sobre 7,129 características. También incluye 34 ejemplos de test, con 20 ALL y 14 AML. Hemos obtenido un error del 2,94 % con 7 características mientras que [Pavlidis and Poirazi, 2006] obtiene el mismo error seleccionando 49 características a través de una técnica basada en marcadores individuales. Otros trabajos recientes [Díaz-Uriate and de Andrés, 2006, Gentile, 2003] obtienen errores de test más altos del 5 % en este conjunto de datos.

- Colon: Este conjunto de datos contiene 62 ejemplos recogidos de pacientes con cáncer de colon. Entre ellos hay 40 biopsias de tumor y 22

biopsias normales provenientes de partes sanas del colon de los mismos pacientes. De entre 6500 genes, 2000 han sido preseleccionados para este conjunto de datos, basándose en la confianza en los niveles de expresión medidos. En nuestros experimentos de selección de características, 15 de ellos han sido seleccionados, resultando un error LOOCV del 0 %, mientras que otros trabajos informan de errores superiores al 12 % [Díaz-Uriate and de Andrés, 2006, Ruiz et al., 2006].

- Central Nervous System Embryonal Tumors: Este conjunto de datos contiene 60 ejemplos de pacientes, 21 son supervivientes tras determinado tratamiento, y 39 son de pacientes que no lo han superado. Hay 7129 genes que definen cada ejemplo, y nosotros hemos seleccionado 9 de ellos para alcanzar un error LOOCV del 1,67 %. En [Pavlidis and Poirazi, 2006] informan del 8,33 % de error de validación cruzada, utilizando 100 características.

- Prostate: Este conjunto de datos contiene dos clases: tumor, y normales. El conjunto de entrenamiento contiene 52 ejemplos de tumor de próstata y 50 sin tumor, definidos por 12600 genes. También hay conjunto independiente de test, pero éste proviene de un experimento distinto y tiene tiene diferentes intensidades con respectro al conjunto de entrenamiento. El test que hemos utilizado es el mismo que el que utilizan en [Singh et al., 2002], con 25 ejemplos de tumor y 9 normales. Nuestros experimentos alcanzaron el mejor error con sólo 5 características, con un error del 5,88 %. Otros trabajos [Díaz-Uriate and de Andrés, 2006, Gentile, 2003] informan de errores superiores al 6 %.

En un trabajo [Gentile, 2003] se informa de errores ligeramente mejores para el conjunto de datos Leukemia (2,50 % con 100 características), pero no podemos compararnos con ellos porque los resultados de sus experimentos se refieren a diferentes conjuntos de entrenamiento y de test, diseñados por ellos. Los mejores resultados de selección de característica los hemos resumido en las tablas 4.1 y 4.2.

D.5.3. Clasificación de estructuras

Aunque la selección de características juega un papel fundamental en el reconocimiento de patrones, hay pocos estudios acerca de este tema en clasificación de estructuras. En esta aplicación estudiamos los grafos de

Reeb [Biasotti, 2005] obtenidos a través de funciones diferentes, desde formas (objetos) tridimensionales. Cuál es el papel de cada tipo de grafo Reeb, y cuál es el papel de cada característica en la clasificación, son respuestas en las que nos puede ayudar el proceso de selección de características. La clasificación de grafos basada en características puramente estructurales es un problema poco explorado ya que tradicionalmente se ha utilizado la información de atributos en los nodos para ayudar a la clasificación. En esta aplicación utilizamos características espectrales basadas sólo en la estructura del grafo.

Los grafos Reeb [Reeb, 1946] representan la topología de una superficie a través de una estructura cuyos nodos se corresponden con puntos críticos de determinada función. Si esta función es derivable, los puntos críticos están localizados en cambios topológicos de la superficie. Así, los grafos Reeb describen estructura topológica *global* de la superficie, así como características locales identificadas por la función que se utiliza. La representación de grafos que se utiliza en esta aplicación es la de Grafos Reeb Extendidos propuesta por [Biasotti, 2004, Biasotti, 2005] para mallas triangulares que representan superficies cerradas en espacio tridimensional. La característica más interesante de los grafos Reeb es su naturaleza paramétrica. Cambiando la función se pueden obtener descriptores diferentes de una superficie, que describan propiedades distintas de la forma. Aquí se utilizan tres funciones escalares alternativas: la distancia geodésica integral, y otras dos funciones de distancia basadas en el centro de masas y en el centro de la esfera que circunscribe la malla triangular. En la figura 5.2 se ejemplifican los tres casos.

Las características que extraemos de estos tres grafos generados a partir de cada superficie, son principalmente espectrales. No es el caso del histograma de grados, que es una de las características más sencillas pero ampliamente utilizada en la literatura. Es una importante fuente de información estadística sobre el grafo, y consiste en contar las aristas que salen de cada nodo. Una característica más elaborada es la centralidad de nodos de subgrafos que cuantifica el grado de participación de un nodo en subgrafos estructurales. Se define en términos del espectro de la matriz de adyacencia del grafo. Otra característica relacionada es el llamado autovector de Perron-Frobenius. Los componentes de este vector denotan el grado de importancia de cada nodo en una componente conexa y están íntimamente relacionados con a la centralidad de subgrafos. Aparte del estudio de la matriz de adyacencia, también es interesante explotar el especto de la matriz Laplaciana, o el espectro de la Laplaciana normalizada. Estos espec-

tros codifican información estructural significativa del grafo. El autovector de Friedler que se corresponde con el primer valor no trivial del espectro en grafos conexos, codifica información de la conectividad de la estructura y está relacionado con la constante de Cheeger. Además tanto los autovalores como los autovectores de la Laplaciana son importantes para la definición de una métrica entre nodos del grafo, llamada *commute times*. Se trata del número de pasos medio que le lleva a un camino aleatorio llegar desde un nodo a otro, y volver. Finalmente también consideramos los núcleos de difusión de grafos, que pertenecen a la familia de núcleos de regularización. La traza del *flujo de complejidad* caracteriza el proceso de difusión y es una versión rápida de la complejidad de politopos.

Dados los tres grafos por cada superficie 3D se extraen las características descritas – 9 por cada grafo. Se transforman en histogramas y se normalizan por el volumen del grafo para independizarse de la escala. La curva del flujo de complejidad no se convierte en histograma sino que se discretiza. Puesto no hay un número de bins óptimo para los histogramas, se utilizan simultáneamente varios histogramas diferentes: de 2, 4, 6 y 8 bins. En total, sumando todos los bins de todos los histogramas de cada característica aplicada a cada uno de los tres grafos, se obtiene un total de $9 \cdot 3 \cdot 20 = 540$ características por cada objeto 3D. Véase la figura 5.4 que describe el proceso de extracción.

La selección de características se ha realizado sobre el conjunto de datos llamado SHREC [Attene and Biasotti, 2008], ilustrado en la figura 5.5. Consite en 15 clases $\times$ 20 objetos. Cada uno de los 300 objetos se caracteriza por 540 características que extraemos, y las clases están etiquetados por una etiqueta $l \in \{$*human, cup, glasses, airplane, chair, octopus, table, hand, fish, bird, spring, armadillo, buste, mechanic, four-leg*$\}$. Los errores de clasificación que obtenemos al clasificar las 15 clases están ilustrados en la figura 5.3. El mínimo es del $23,3\,\%$ y se alcanza con 222 características. Este error es inferior cuando se clasifican sólo 10 clases ($15,5\,\%$), 5 clases ($6\,\%$) y 2 clases ($0\,\%$).

En esta aplicación se analiza el papel de cada una de las diferentes características en el problema de clasificación de grafos. En la figura 5.8 representamos la evolución de la selección de características. Las áreas coloreadas representan la relación de la participación entre las diferentes características en cada una de las iteraciones durante el proceso de selección de características, desde el conjunto vacío de características, hasta el conjunto completo de características. La altura en el eje Y es arbitraria, es decir, el orden de cada una de las características en el gráfico no importa, sólo se

representa su participación a través del área. Para el experimento de 15 clases el autovector de Friedler es el más importante, en combinación con el resto de características. Commute times también es una característica importante. Sin embargo la centralidad de nodos y el flujo de complejidad no tienen tanto papel en la clasificación. En cuanto a los tres tipos de grafos Reeb, los tres juegan un papel importante desde el estado inicial de selección de características. En la figura 5.7 representamos la proporción de características seleccionadas cuando se alcanzó el menor error. Por otro lado en la figura 5.9 presentamos cuatro experimentos diferentes con subconjuntos de la base de datos que consisten en problemas de clasificación de tres clases, cada uno de ellos. Hemos diseñado estos experimentos de manera que cada experimento contenga clases que compartan algún tipo de parecido estructural. La participación de cada una de las características es bastante consistente con el experimento inicial de 15 clases. Hay pequeñas diferencias en cuanto a la centralidad de nodos, que parece ser más importante cuando hay objetos con formas puntiagudas. Por otro lado los grafos obtenidos con la función esférica tienen más participación con objetos con formas más recogidas y conexas.

Otro experimento realizado con esta base de datos es referido a la importancia de la buena estimación de la información mutua. Para dicha estimación se requiere un cálculo de vecinos próximos que nosotros realizamos utilizando la librería ANN [Mount and Arya, 1997]. Esta librería permite variar el nivel de precisión de la estimación de vecinos próximos, a cambio de ganar tiempo computacional, véase la gráfica de la figura 5.11. Estos experimentos demuestran que la precisión utilizada para la estimación de la información mutua debe ser la máxima porque tiene un importante impacto en los errores finales de clasificación (figura 5.10). Otro aspecto importante de estos experimentos es el estudio de cómo cambia el tipo de características seleccionadas con la disminución de la precisión de los vecinos próximos. Los resultados se presentan en la figura 5.12 donde se pueden ver comparaciones de cuatro parejas de diferentes *error bounds* de los vecinos próximos. Están representados con área de color el primero, y con líneas negras superpuestas el segundo. Se puede observar que hay un cambio significativo, sobre todo en el primer decremento de precisión. La conclusión es que la máxima precisón es fundamental en la selección de características basada en información mutua.

D.6. Conclusiones y trabajo futuro

En esta tesis doctoral presentamos un novedoso método de selección de características basado en información mutua. Este enfoque es capaz de seleccionar características desde conjuntos de características de muy alta dimensionalidad, del orden de miles de características, porque la dependencia entre los conjuntos de características y la etiqueta de la clase se estima con grafos entrópicos o con vecinos próximos, evitando tener que estimar una distribución. Este procedimiento tiene una complejidad computacional que ya no depende del número de dimensiones, sino que pasa a depender principalmente del número de ejemplos.

Presentamos un estudio sobre las estimaciones que ofrecen dos métodos no-paramétricos de estimación de entropía, que es necesaria para calcular la información mutua. Uno está basado en árboles entrópicos mínimos, mientras que el otro está basado en vecinos próximos. El segundo ha resultado ser más apropiado para nuestro método al no tener parámetros dependientes de los datos y tener menor error. Además tiene menor complejidad computacional.

Tres aplicaciones reales de clasificación de datos nos permiten probar la eficacia del método. Con los experimentos sobre microarrays genéticos demostramos la viabilidad del método para dimensionalidades muy altas. Por otro lado los resultados de clasificación que obtenemos son muy prometedores, al superar la mayoría de los resultados de clasificación del estado del arte [Bonev et al., 2008]. El método de selección de características también es comparado con otros criterios de selección en estos experimentos.

Los experimentos con imágenes están principalmente orientados a localización de un robot móvil. La clasificación basada en apariencia con selección de características ha demostrado ser apropiada para la tarea de localización visual, tanto con imágenes de cámara monocular, como con imágenes omnidireccionales. Este enfoque para localización es muy general, ya que puede ser entrenado para una amplia variedad de entornos, tanto de interiores como de exteriores. Sin embargo su rendimiento puede depender de del entorno y del número de clases en las que se desee clasificarlo. Este tipo de localización es de grano grueso y ha probado ser útil para sistemas de localización jerárquica [Bonev et al., 2007d, Escolano et al., 2007] donde ésta aporta información de contexto importante para la posterior localización de grano fino.

Los experimentos con características espectrales de grafos demuestran la viabilidad de la clasificación multiclase basada únicamente en información estructural. La selección basada en teoría de la información nos permite

un análisis cuyos resultados sugieren que características similares son seleccionadas para un amplio rango de conjuntos de objetos. Así podemos estudiar qué características espectrales y qué representaciones de grafos aportan más a los problemas de clasificación de estructuras.

Como trabajo futuro estamos considerando el desarrollo de un criterio de parada basado en teoría de la información. El error mínimo cometido siempre va a ser dependiente del clasificador, no obstante se puede buscar un criterio que sea lo suficientemente general como para funcionar bien con distintos tipos de clasificadores, y para ello la teoría de la información podría jugar un papel clave. También queremos explorar diferentes tipos de orden de búsqueda en el espacio de características. Es necesario algún orden que no sea completamente voraz, pero que sea de una complejidad computacional viable. Creemos que el criterio de evaluación tiene un impacto importante sobre la decisión de qué recorrido de búsqueda realizar en el espacio de características, y queremos estudiar esta relación. También estamos estudiando la posibilidad de combinar modelos generativos con el modelo discriminativo que estudiamos en esta tesis. La retroalimentación entre ambos podría ofrecernos nuevas respuestas. La selección de características es un problema abierto que ofrece un amplio abanico de nuevas direcciones en la investigación[2].

[2]Esta traducción a castellano es un resumen de la tesis doctoral. Para ver en detalle la metodología y experimentación, véase el resto del documento con la versión original en inglés.

Bibliography

[Almuallim and Dietterich, 1991] Almuallim, H. and Dietterich, T. G. (1991). Learning with many irrelevant features. In *In Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552. AAAI Press.

[Arkin, 1999] Arkin, R. (1999). *Behavior-based robotics*. MIT Press.

[Armañanzas et al., 2008] Armañanzas, R., Inza, I., Santana, R., Saeys, Y., Flores, J. L., Lozano, J. A., de Peer, Y. V., Blanco, R., Robles, V., Bielza, C., and Larrañaga, P. (2008). A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, 1.

[Attene and Biasotti, 2008] Attene, M. and Biasotti, S. (2008). Shape retrieval contest 2008: Stability of watertight models. In *SMI'08*, pages 219–220.

[Balagani et al., 2010] Balagani, K. S., Phoha, V. V., Iyengar, S. S., and Balakrishnan, N. (2010). On guo and nixon's criterion for feature subset selection: Assumptions, implications, and alternative options. *IEEE Transactions on Systems Man and Cybernetics Part A: Systems and Humans*, PP(99):1–5.

[Barlow, 2001] Barlow, H. B. (2001). Redundancy reduction revisited. *Network: computation in neural systems*, 12:241–253.

[Beirlant et al., 1996] Beirlant, E., Dudewicz, E., Gyorfi, L., and der Meulen, E. V. (1996). Nonparametric entropy estimation. *International Journal on Mathematical and Statistical Sciences*, 6(1):17–39.

[Bellman, 1957] Bellman, R. (1957). Dynamic programming. In *Princeton University Press*.

[Benhimane and Malis, 2006] Benhimane, S. and Malis, E. (2006). A new approach to vision-based robot control with omni-directional cameras. In *IEEE International Conference on Robotics and Automation, Orlando*.

[Bertsimas and Ryzin, 1990] Bertsimas, D. and Ryzin, G. V. (1990). An asymptotic determination of the minimum spanning tree and minimum matching constants in geometrical probability. *Operations Research Letters*, 9(1):223–231.

[Biasotti, 2004] Biasotti, S. (2004). *Computational Topology Methods for Shape Modelling Applications*. PhD thesis, Universitá degli Studi di Genova.

[Biasotti, 2005] Biasotti, S. (2005). Topological coding of surfaces with boundary using Reeb graphs. *Computer Graphics and Geometry*, 7(1):31–45.

[Blum and Langley, 1997] Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. In *Artificial Intelligence*.

[Boada et al., 2004] Boada, B., Blanco, D., and Moreno, L. (2004). Symbolic place recognition in voronoi-based maps by using hidden markov models. In *Journal of Intelligent and Robotic Systems, Springer Netherlands*, volume 39.

[Bonev and Cazorla, 2006a] Bonev, B. and Cazorla, M. (2006a). Towards autonomous adaptation in visual tasks. In *WAF 2006 (VII Workshop de Agentes FÃsicos). Las Palmas de Gran Canaria (Spain)*, pages 59–66.

[Bonev and Cazorla, 2006b] Bonev, B. and Cazorla, M. (2006b). Towards autonomous adaptation in visual tasks. In *WAF 2006 (VII Workshop de Agentes FÃsicos). Las Palmas de Gran Canaria (Spain)*, pages 59–66.

[Bonev and Cazorla, 2008] Bonev, B. and Cazorla, M. (2008). Large scale environment partitionaing in mobile robotics recognition tasks. In *WAF 2008 (IX Workshop de Agentes Físicos) - Vigo (Spain)*.

[Bonev et al., 2007a] Bonev, B., Cazorla, M., and Escolano, F. (2007a). Robot navigation behaviors based on omnidirectional vision and information theory. *Journal of Physical Agents*, 1(1):27–35.

[Bonev et al., 2007b] Bonev, B., Cazorla, M., and Escolano, F. (2007b). Robot navigation behaviors based on omnidirectional vision and information theory. In *WAF 2007 (VIII Workshop de Agentes FÃsicos) - CEDI 2007 Zaragoza (Spain)*.

[Bonev et al., 2007c] Bonev, B., Escolano, F., and Cazorla, M. (2007c). A novel information theory method for filter feature selection. In *MICAI 2007, Aguascalientes, Mexico, LNAI*.

[Bonev et al., 2008] Bonev, B., Escolano, F., and Cazorla, M. (2008). Feature selection, mutual information, and the classification of high-dimensional patterns. *Pattern Analysis and Applications*, pages 309–319.

[Bonev et al., 2007d] Bonev, B., Escolano, F., Lozano, M., Suau, P., Cazorla, M., and Aguilar, W. (2007d). Constellations and the unsupervised learning of graphs. In *6th IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition. Alicante (Spain)*, pages 340–350.

[Borenstein and Koren, 1989] Borenstein, J. and Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 19, pages 1179–1187.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Breiman and Breiman, 1996] Breiman, L. and Breiman, L. (1996). Bagging predictors. In *Machine Learning*, pages 123–140.

[Buschka et al., 2000] Buschka, P., Saffiotti, A., and Wasik, Z. (2000). Fuzzy landmark-based localization for a legged robot. In *IEEE/RSJ*, pages 1205–1210, Takamatsu, Japan.

[Calvo et al., 2009] Calvo, B., Larrañaga, P., and Lozano, J. A. (2009). Feature subset selection from positive and unlabelled examples. *Pattern Recognition Letters*, 30(11):1027–1036.

[Carmichael et al., 2002] Carmichael, O., Mahamud, S., and Hebert, M. (2002). Discriminant filters for object recognition, technical report. CMU-RI-TR-02-09.

[Cazorla and Escolano, 2003] Cazorla, M. and Escolano, F. (2003). Two bayesian methods for junction classification. *IEEE Transactions on Image Processing*, 12(3):317–327.

[Cover and Thomas, 1991a] Cover, M. and Thomas, J. (1991a). *Elements of Information Theory*. Wiley Interscience.

[Cover and Thomas, 1991b] Cover, T. and Thomas, J. (1991b). *Elements of Information Theory*. J. Wiley and Sons.

[Díaz-Uriate and de Andrés, 2006] Díaz-Uriate, R. and de Andrés, S. A. (2006). Gene selection and classification of microarray data using random forest. In *BMC Bioinformatics*, volume 7:3.

[Dill et al., 1993] Dill, M., Wolf, R., and Heisenberg, M. (1993). Visual pattern recognition in drosophila involves retinotopic matching. In *Nature*, number 365(6448):639-4.

[Duda et al., 2000] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience Publication.

[Efron, 1981] Efron, B. (1981). Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika*, 68(3):589–599.

[Ekvall et al., 2005] Ekvall, S., Kragic, D., and Hoffmann, F. (2005). Object recognition and pose estimation using color cooccurrence histograms and geometric modeling. In *Image and Vision Computing*, number 23, pages 943–955.

[Escobar and del Solar, 2002] Escobar, M. and del Solar, J. R. (2002). Biologically-based face recognition using gabor filters and log-polar images. In *Int. Joint Conf. on Neural Networks - IJCNN 2002, Honolulu, USA*.

[Escolano et al., 2007] Escolano, F., Bonev, B., Suau, P., Aguilar, W., Frauel, Y., Saez, J., and Cazorla, M. (2007). Contextual visual localization: cascaded submap classification, optimized saliency detection, and fast view matching. In *IEEE International Conference on Intelligent Robots and Systems. San Diego, California. USA*.

[Escolano et al., 2009a] Escolano, F., Giorgi, D., Hancock, E. R., Lozano, M. A., and Falcidieno, B. (2009a). Flow complexity: Fast polytopal graph complexity and 3d object clustering. In *GbRPR*, pages 253–262.

[Escolano et al., 2008] Escolano, F., Hancock, E. R., and Lozano, M. A. (2008). Birkhoff polytopes, heat kernels and graph complexity. In *ICPR*, pages 1–5.

[Escolano et al., 2009b] Escolano, F., Suau, P., and Bonev, B. (2009b). *Information Theory in Computer Vision and Pattern Recognition*. Springer, Computer Imaging, Vision, Pattern Recognition and Graphics, New York.

[Estévez et al., 2009] Estévez, P. A., Tesmer, M., Perez, C. A., and Zurada, J. M. (2009). Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201.

[Estrada and Rodriguez, 2005] Estrada, E. and Rodriguez, J. A. (2005). Subgraph centrality in complex networks. *Physical Review E*, 71(5).

[Gaspar and Santos-Victor, 2000] Gaspar, J. and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omni-directional camera. In *IEEE Transactions on Robotics and Automation*, volume 16.

[Gentile, 2003] Gentile, C. (2003). Fast feature selection from microarray expression data via multiplicative large margin algorithms. In *Neural Information Processing Systems*.

[Genuer et al., 2010] Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, In Press, Accepted Manuscript.

[Geyer and Daniilidis, 2001] Geyer, C. and Daniilidis, K. (2001). Structure and motion from uncalibrated catadioptric views. In *IEEE Conf. on Computer Vision and Pattern Recognition, Hawaii*, pages 11–13.

[Gilad-Bachrach et al., 2004] Gilad-Bachrach, R., Navot, A., and Tishby, N. (2004). Margin based feature selection - theory and algorithms. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 43, New York, NY, USA. ACM.

[Goldberger et al., 2006] Goldberger, J., Gordon, S., and Greenspan, H. (2006). Unsupervised image-set clustering using an information theoretic framework. *IEEE Transactions on Image Processing*, 15(2):449–458.

[Guo and Nixon, 2009] Guo, B. and Nixon, M. (2009). Gait feature subset selection by mutual information. *IEEE Transactions on Systems, Man, and Cybernetics–part A: Systems and Humans*, 39(1):36–46.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. In *Journal of Machine Learning Research*, volume 3, pages 1157–1182.

[Hastie and Tibshirani, 1996] Hastie, T. and Tibshirani, R. (1996). Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society, Series B*, 58:155–176.

[Hero and Michel, 1999] Hero, A. and Michel, O. (1999). Asymptotic theory of greedy aproximations to minnimal k-point random graphs. *IEEE Trans. on Infor. Theory*, 45(6):1921–1939.

[Hero and Michel, 2002] Hero, A. and Michel, O. (2002). Applications of spanning entropic graphs. *IEEE Signal Processing Magazine*, 19(5):85–95.

[Hilaga et al., 2001] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. (2001). Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH'01*, pages 203–212, Los Angeles, CA.

[Hong and Schonfeld, 2008] Hong, H. and Schonfeld, D. (2008). Maximum-entropy expectation-maximization algorithm for image reconstruction and sensor field estimation. *IEEE Transactions on Image Processing*, 17(6):897–907.

[Hua et al., 2009] Hua, J., Tembe, W. D., and Dougherty, E. R. (2009). Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42(3):409 – 424.

[Hughes, 1968] Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, IT-14(1):55–63.

[Hyvarinen and Oja, 2000] Hyvarinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430.

[Jain and Zongker, 1997] Jain, A. and Zongker, D. (1997). Feature selection: Evaluation, application and small sample performance. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19.

[Jebara, 2004] Jebara, T. (2004). *Machine Learning: Discriminative and Generative*. Springer.

[Jirapech-Umpai and Aitken, 2005] Jirapech-Umpai, T. and Aitken, S. (2005). Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. In *BMC Bioinformatics*, number 6:148.

[Jogan and Leonardis, 2003] Jogan, M. and Leonardis, A. (2003). Robust localization using an omnidirectional appearance-based subspace model of environment. In *Robotics and Autonomous Systems*, volume 45, pages 51–72. Elsevier Science.

[John et al., 1994] John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. *International Conference on Machine Learning*, pages 121–129.

[Kira and Rendell, 1992] Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *ML '92: Proceedings of the Ninth International Workshop on Machine Learning*, pages 249–256, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Kohavi and John, 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection.

[Koller and Sahami, 1996] Koller, D. and Sahami, M. (1996). Toward optimal feature selection. pages 284–292.

[Kozachenko and Leonenko, 1987] Kozachenko, L. F. and Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. *Problems Information Transmission*, 23(1):95–101.

[Kraskov et al., 2004] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Phys. Rev. E*, 69(6):066138.

[Lambrinos et al., 2000] Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., and Wehner, R. (2000). A mobile robot employing insect strategies for navigation. In *Robotics and Autonomous Systems, special issue on Biomimetic Robots*, volume 30, pages 39–65.

[Larrañaga et al., 2006] Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Martínez, A. P., and Robles, V. (2006). Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112.

[Lenser and Veloso, 2003] Lenser, S. and Veloso, M. (2003). Visual sonar: Fast obstacle avoidance using monocular vision. In *IEEE/RSJ IROS*.

[Leonenko et al., 2008] Leonenko, N., Pronzato, L., and Savani, V. (2008). A class of rényi information estimators for multidimensional densities. *The Annals of Statistics*, 36(5):2153–2182.

[Liu et al., 2009] Liu, H., Sun, J., Liu, L., and Zhang, H. (2009). Feature selection with dynamic mutual information. *Pattern Recognition*, 42(7):1330–1339.

[Lopez-Franco and Bayro-Corrochano, 2004] Lopez-Franco, C. and Bayro-Corrochano, E. (2004). Unified model for omnidirectional vision using the conformal geometric algebra framework. In *17th International Conference on Pattern Recognition (ICPR'04)*, volume 4, pages 48–51.

[Lozano et al., 2008] Lozano, M., Escolano, F., Bonev, B., P.Suau, Aguilar, W., Saez, J., and Cazorla, M. (2008). Region and constellations based categorization of images with unsupervised graph learning. *Image and Vision Computing*.

[Luo et al., 2003] Luo, B., Wilson, R., and Hancock, E. (2003). Spectral embedding of graphs. *Pattern Recognition*, 36(10):2213–2223.

[Mariottini et al., 2006] Mariottini, G., Prattichizzo, D., and Oriolo, G. (2006). Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera. In *IEEE International Conference on Robotics and Automation, Orlando*.

[Martínez et al., 2006] Martínez, A. P., Larrañaga, P., and Inza, I. (2006). Information theory and classification error in probabilistic classifiers. In *Discovery Science*, pages 347–351.

[Meese and Hess, 2004] Meese, T. and Hess, R. (2004). Low spatial frequencies are suppressively masked across spatial scale, orientation, field position, and eye of origin. In *Journal of Vision*, volume 4, pages 843–859.

[Menegatti et al., 2004] Menegatti, E., Zoccarato, M., Pagello, E., and Ishiguro, H. (2004). Image-based monte-carlo localisation with omnidirectional images. In *Robotics and Autonomous Systems*. Elsevier Science.

[Miller, 1984] Miller, A. J. (1984). Selection of subsets of regression variables. *Journal of the Royal Statistical Society. Series A (General)*, 147(3):389–425.

[Mount and Arya, 1997] Mount, D. and Arya, S. (1997). Ann: A library for approximate nearest neighbor searching.

[Munson and Caruana, 2009] Munson, M. A. and Caruana, R. (2009). On feature selection, bias-variance, and bagging. In *ECML PKDD '09: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 144–159, Berlin, Heidelberg. Springer-Verlag.

[nalver et al., 2009] nalver, A. P., Escolano, F., and Sáez, J. (2009). Learning gaussian mixture models with entropy based criteria. *IEEE Transactions on Neural Networks*.

[Navot et al., 2005] Navot, A., Gilad-Bachrach, R., Navot, Y., and Tishby, N. (2005). Is feature selection still necessary? In Saunders, C., Grobelnik, M., Gunn, S. R., and Shawe-Taylor, J., editors, *SLSFS*, volume 3940 of *Lecture Notes in Computer Science*, pages 127–138. Springer.

[Neemuchwala et al., 2006] Neemuchwala, H., Hero, A., and Carson, P. (2006). Image registration methods in high-dimensional space. In *International Journal on Imaging*.

[Paninski, 2003] Paninski, I. (2003). Estimation of entropy and mutual information. *Neural Computation*, 15(1).

[Parzen, 1962] Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

[Pavlidis and Poirazi, 2006] Pavlidis, P. and Poirazi, P. (2006). Individualized markers optimize class prediction of microarray data. In *BMC Bioinformatics*, volume 7, page 345.

[Payne and Edwards, 1998] Payne, T. R. and Edwards, P. (1998). Implicit feature selection with the value difference metric. In *In ECAI 98 Conference Proceedings*, pages 450–454. John Wiley and Sons.

[P.Chang and J.Krumm, 1999] P.Chang and J.Krumm (1999). Object recognition with color cooccurrence histograms. In *IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO*.

[Peñalver et al., 2006] Peñalver, A., Escolano, F., and Sáez, J. (2006). Ebem: An entropy-based em algorithm for gaussian mixture models.

In *International Conference on Pattern Recognition, Hong-Kong, China*, pages 451–455.

[Peng et al., 2005] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27, pages 1226–1238.

[Perkins and Theiler, 2003] Perkins, S. and Theiler, J. (2003). Online feature selection using grafting. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC*.

[Peter and Rangarajan, 2008] Peter, A. M. and Rangarajan, A. (2008). Maximum likelihood wavelet density estimation with applications to image and shape matching. *IEEE Trans Image Process*, 17(4):458–68.

[Pirooznia et al., 2008] Pirooznia, M., Yang, J. Y., Yang, M. Q., and Deng, Y. (2008). A comparative study of different machine learning methods on microarray gene expression data. *BMC Medical Genomics*, 9(Suppl 1).

[Ponsa and López, 2007] Ponsa, D. and López, A. (2007). Feature selection based on a new formulation of the minimal-redundancy-maximal-relevance criterion. In *IbPRIA*, pages 47–54.

[Qiu and E.R.Hancock, 2007] Qiu, H. and E.R.Hancock (2007). Clustering and embedding using commute times. *IEEE Transactions on PAMI*, 29(11):1873–1890.

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

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Rajwade et al., 2009] Rajwade, A., Banerjee, A., and Rangarajan, A. (2009). Probability density estimation using isocontours and isosurfaces: Applications to information-theoretic image registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:475–491.

[Reeb, 1946] Reeb, G. (1946). Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus*, 222:847–849.

[Ruiz et al., 2006] Ruiz, R., Riquelme, J. C., and Aguilar-Ruiz, J. S. (2006). Incremental wrapper-based gene selection from microarray data for cancer classification. In *Pattern Recognition*, number 12, pages 2383–2392.

[Saeys et al., 2008] Saeys, Y., Abeel, T., and de Peer, Y. V. (2008). Robust feature selection using ensemble feature selection techniques. In *ECML/PKDD (2)*, pages 313–325.

[Saeys et al., 2007a] Saeys, Y., Inza, I., and Larrañaga, P. (2007a). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

[Saeys et al., 2007b] Saeys, Y., Inza, I., and Larrañaga, P. (2007b). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

[Schiele and Crowley, 1996] Schiele, B. and Crowley, J. (1996). Probabilistic object recognition using multidimensional receptive field histograms. In *ICPR'96, International Conference on Pattern Recognition, Vienna, Austria*.

[Shapiro, 2005] Shapiro, L. (2005). Object and concept recognition for content-based image retrieval. Groundtruth Database available at http://www.cs.washington.edu/research/imagedatabase/ groundtruth/.

[Sima and Dougherty, 2006] Sima, C. and Dougherty, E. (2006). What should be expected from feature selection in small-sample settings. In *Bioinformatics*, volume 22, pages 2430–2436.

[Singh et al., 2002] Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T., and Sellers, W. (2002). Gene expression correlates of clinical prostate cancer behavior. In *Cancer Cell*, volume 1.

[Stowell and Plumbley, 2009] Stowell, D. and Plumbley, M. D. (2009). Fast multidimensional entropy estimation by k-d partitioning. *IEEE Signal Processing Letters*, 16(6):537–540.

[Suau and Escolano, 2008] Suau, P. and Escolano, F. (2008). Bayesian optimization of the scale saliency filter. *Image and Vision Computing*, 26(9):1207–1218.

[Tang et al., 1991] Tang, Y., Cheng, H., and Suen, C. (1991). Transformation-ring-projection (trp) algorithm and its vlsi implementation. In *International Journal on Pattern Recognition and Artifical Intelligence*, volume 5, pages 25–56.

[Tarr and Bülthoff, 1999] Tarr, M. and Bülthoff, H. (1999). Object recognition in man, monkey, and machine. In *Cognition Special Issues*. MIT Press.

[Tishby et al., 1999] Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. pages 368–377.

[Torkkola, 2003] Torkkola, K. (2003). Feature extraction by non-parametric mutual information maximization. In *Journal of Machine Learning Research*, volume 3, pages 1415–1438.

[Tu et al., 2006] Tu, Z., Chen, X., Yuille, A. L., and Zhu, S. C. (2006). Image parsing: Unifying segmentation, detection, and recognition. In *Toward Category-Level Object Recognition*, pages 545–576.

[Tu and Zhu, 2002] Tu, Z. and Zhu, S. (2002). Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:657–673.

[Tuv et al., 2006] Tuv, E., Borisov, A., and Torkkola, K. (2006). Feature selection using ensemble based ranking against artificial contrasts. In *IJCNN*, pages 2181–2186.

[Ullman, 2000] Ullman, S. (2000). *High-level vision: object recognition and visual cognition*. MIT Press.

[Vafaie and Jong, 1995] Vafaie, H. and Jong, K. D. (1995). Genetic algorithms as a tool for restructuring feature space representations. In *In Proc. of the International Conference on Tools with Artificial Intelligence. IEEE Computer Soc*, pages 8–11. Press.

[Vasconcelos and Vasconcelos, 2004] Vasconcelos, N. and Vasconcelos, M. (2004). Scalable discriminant feature selection for image retrieval and recognition. In *Computer Vision and Pattern Recognition Conference (CVPR04)*, pages 770–775.

[Vidal-Naquet and Ullman, 2003] Vidal-Naquet, M. and Ullman, S. (2003). Object recognition with informative features and linear classification. In

ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision, page 281, Washington, DC, USA. IEEE Computer Society.

[Vignat et al., 2004] Vignat, C., III, A. O. H., and Costa, J. A. (2004). About closedness by convolution of the tsallis maximizers. *Physica A: Statistical Mechanics and its Applications*, 340(1-3):147 – 152. News and Expectations in Thermostatistics.

[Viola, 1995] Viola, P. (1995). *Alignment by Maximization of Mutual Information*. PhD thesis, AI Lab. MIT.

[Viola et al., 1996] Viola, P., Schraudolph, N. N., and Sejnowski, T. J. (1996). Empirical entropy manipulation for real-world problems. *Adv. in Neural Infor. Proces. Systems*, 8(1).

[Viola and Wells-III, 1995] Viola, P. and Wells-III, W. M. (1995). Alignment by maximization of mutual information. In *5th Intern. Conf. on Computer Vision*. IEEE.

[Viola and Wells-III, 1997] Viola, P. and Wells-III, W. M. (1997). Alignment by maximization of mutual information. In *5th International Conference on Computer Vision*, volume 2, pages 137–154. IEEE.

[Wang et al., 2006] Wang, Q., Kulkarni, S. R., and Verdú, S. (2006). A nearest-neighbor approach to estimating divergence between continuous random vectors. In *IEEE Int. Symp. Information Theory, Seattle, WA*, volume 23, pages 95–101.

[Wang and Gotoh, 2009] Wang, X. and Gotoh, O. (2009). Accurate molecular classification of cancer using simple rules. *BMC Medical Genomics*, 64(2).

[Wang et al., 2005] Wang, Y., Tetko, I. V., Hall, M. A., Frank, E., Facius, A., Mayer, K. F. X., and Mewes, H. W. (2005). Gene selection from microarray data for cancer classification-a machine learning approach. *Comput. Biol. Chem.*, 29(1):37–46.

[Weisstein, 1998] Weisstein, E. W. (1998). *Concise Encyclopedia of Mathematics*. CRC Press.

[Wolpert and Wolf, 1995] Wolpert, D. and Wolf, D. (1995). Estimating function of probability distribution from a finite set of samples. *Phisical Review E*, 52(6).

[Xing et al., 2001] Xing, E., Jordan, M., and Karp, R. (2001). Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608.

[Yeung and Bumgarner, 2003] Yeung, K. Y. and Bumgarner, R. E. (2003). Multiclass classification of microarray data with repeated measurements: application to cancer. *Genome Biology*, R83(4).

[Yuille et al., 2001] Yuille, A. L., Coughlan, J. M., Wu, Y., and Zhu, S. C. (2001). Order parameters for detecting target curves in images: When does high level knowledge help? *Int. J. Comput. Vision*, 41(1-2):9–33.

[Zhu et al., 1998] Zhu, S. C., Wu, Y. N., and Mumford, D. (1998). Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126.

Index