Ricardo Jorge Gamelas de Sousa

Multicriteria Learning on Ordinal Data

Ph.D. Thesis

School of Engineering, University of Porto
2012

Thesis submitted to School of Engineering, University of Porto
for the Degree of Doctor of Philosophy

Ricardo Jorge Gamelas de Sousa
(ricardo.jorge.sousa@fe.up.pt)

Thesis submitted under supervision of
Professor Doctor Jaime S. Cardoso (INESC Porto, FEUP),
Professor Doctor Joaquim F. Pinto da Costa (CMUP, FCUP)

Porto, 2012

Abstract

Several operations of recognition and prediction are performed nowadays, many without even people consciousness. Predictive learning has traditionally consisted in constructing rules which discriminate positive from negative, or malign from benign cases depending on the scenario in analysis. Models are constructed by gathering knowledge from data. Data describe the problem through different features, attributes or criteria thereby constituting the feature space. The insight gained will make possible the estimation of a mapping from the feature space into a finite class space. Depending on the cardinality of the finite class space we are left with binary (e.g., positive and negative) or multiclass classification problems. In more complex situations, one has to deal with data where, the presence or absence of a "natural" order among classes, will separate ordinal from nominal problems.

Retrieving information in a way that we can interpret different criteria on data has been playing major roles in the academy and industry. Specially in scenarios where data contains an order relation not only on the classes labels but also on the data itself. Learning models for these settings are referred to as ordinal data problem. The credit scoring problem is an example of that. In this problem, one evaluates how unlikely a client will default with his payments. Client profiles are evaluated, being their results expressed in terms of an ordinal score scale ($Excellent \succ Good \succ Fair \succ Poor$). Intelligent systems have then to take into consideration different criteria such as payment history, mortgages, wages among others in order to accomplish their outcome.

Contributions of this work are three fold. Firstly, we have shown that existing measures for evaluating ordinal classification models suffer from a number of important shortcomings. For this reason, we proposed an alternative measure defined directly in the confusion matrix. An error coefficient appropriate for ordinal data was therefore designed such that it captures how much the result diverges from the ideal prediction and how "inconsistent" the classifier is in regard to the relative order of the classes.

Secondly, we have identified that despite the myriad of schemes for multi-class classification with Support Vector Machine (SVM), little work has been done for the case where the classes are ordered. Hence, a new SVM methodology was proposed based on the unimodal paradigm with the All-at-Once approach for the ordinal classification. In the same manner, the ordinal data problem on k-Nearest Neighbor (k-NN) and Decision Tree (DT) has not evolved significantly. Knowing that a DT consistent with the ordinal setting is often desirable to aid decision making, we proposed a strategy based on constraints defined globally over the feature space. This approach was further extended through a bootstrap technique to improve the accuracy of the baseline solution.

Thirdly, we explored a particular problem where in many scenarios there is the opportunity to label critical items for manual revision, instead of trying to automatically classify every item. Therefore, the development of classifiers with an extra output class, the reject class, in-between the decision classes, is attractive where the ordinal problem can easily fit in. We present three new approaches on Self-Organizing Map (SOM) and a new paradigm initially proposed for the classification of ordinal data to address the classification problem with reject option was delved.

Finally, the proposed methodologies were assessed in two medical applications.

Resumo

Hoje em dia tem-se ao dispor diversas aplicações de reconhecimento e previsão onde grande parte das quais agem sem real perceção dos seus utilizadores. A aprendizagem preditiva tem sido tradicionalmente constituída pela construção de regras que separam os casos positivos dos negativos ou, os casos malignos dos benignos, dependendo do cenário em análise. Modelos inteligentes são assim construídos através da extração de informação existente nos dados. Estes dados descrevem por sua vez o problema por meios de diversas características, atributos ou critérios, constituindo assim o espaço de características. A introspeção adquirida torna assim possível estimar uma função que irá mapear o espaço de características para um conjunto finito das classes. Dependendo da cardinalidade do espaço das classes, estar-se-à perante um problema de classificação binário (i.e., positivo e negativo) ou multi-classe. Em situações mais complexas, ter-se-ão dados cuja presença ou inexistência de uma ordem "natural" entre as classes irá destingir o problema ordinal do nominal.

A extração de informação de modo a que seja possível interpretar os diferentes critérios existentes nos dados tem tomado um papel importante quer na academia quer na indústria. Especialmente em cenários onde não só a relação de ordem das classes é importante como também a dos dados. Deste modo, modelos de aprendizagem para situações com estas características são identificados como problemas ordinais. Um exemplo é o problema de avaliação de créditos onde um analista identifica quão improvável um determinado cliente irá entrar em incumprimento. Os perfis dos clientes são avaliados segundo diversos fatores onde os resultados são expressos em termos de uma escala ordinal ($Excelente \succ Bom \succ Medíocre \succ Mau$). Sistemas inteligentes tem então que ter em consideração diversos critérios tais como o histórico de pagamentos, dívidas, vencimento, entre outros, para efetuar a decisão.

A presente tese tem três contribuições chave. Em primeiro lugar, mostrou-se que as métricas existentes para avaliar o desempenho dos classificadores para dados ordinais comportam diversas limitações. Por este motivo, foi proposto uma métrica alternativa definida diretamente da matriz de confusão. Definiu-se assim um coeficiente de erro apropriado para dados ordinais tal que capturasse o quanto o resultado diverge da previsão ideal e o quanto "inconsistente" o classificador é relativamente à ordem relativa das classes.

Em segundo lugar, identificou-se que apesar das várias metodologias de aprendizagem baseadas em máquinas de suporte vetorial (SVM), poucos trabalhos existem na literatura para o problema ordinal. Deste modo foi proposto uma nova formulação SVM baseado no paradigma unimodal em conjunto com a abordagem All-at-Once. Similarmente foi identificado que o problema ordinal em k-vizinhos mais próximos e árvores de decisão não teve uma evolução significativa. Sabendo que as árvores de decisão consistentes com o problema ordinal são usualmente desejáveis no apoio à decisão, foi proposto uma nova estratégia baseada em restrições globais. Esta proposta foi posteriormente estendida através de técnicas bootstrap para melhorar o desempenho da solução base.

Em terceiro lugar, explorou-se um problema particular para cenários onde existe a oportunidade de etiquetar os itens mais críticos para revisão manual. Após a identificação das vantagens no desenvolvimento de classificadores com uma classe extra, a classe de rejeição, entre as classes de decisão, verificou-se que o problema ordinal facilmente se enquadrava neste cenário. Assim, explorou-se uma nova abordagem para resolver o problema da opção

de rejeição dentro do contexto ordinal.

Por último, as técnicas apresentadas nesta tese foram exploradas em dois casos concretos de aplicação clínica.

Contents

List of Tables

List of Figures

Acronyms

Notation	Description
AHC	Agglomerative Hierarchical Clustering.
AHP	Analytic Hierarchy Process.
AI	Artificial Intelligence.
ANP	Analytic Network Process.
AR curve	Accuracy-Reject curve.
AUC	Area Under Curve.
BCCT	Breast Cancer Conservative Treatment.
BMU	Best Matching Unit.
CAD	Computer Aided Diagnosis.
CM	Confusion Matrix.
DA	Decision Analysis.
DDAG	Decision Directed Acyclic Graph.
DM	Decision Maker.
DRSA	Dominance-based Set Approach.
DT	Decision Tree.
EA	Evolutionary algorithm.
FS	Feature Selection.
K-Means	K-Means.
k-NN	k-Nearest Neighbor.
KDA	Kernel Discriminant Analysis.
KKT	Karush–Kuhn–Tucker.
LVQ	Learning Vector Quantization.
M.H.DIS	Multi-group Hierarchical Discrimination.
MAE	Mean Absolute Error.
MC	Multicriteria.
MCDA	Multicriteria Decision Analysis.
MER	Misclassification Error Rate.
MIL	Multiple Instance Learning.
MIP	Mixed Integer Programming.

Notation	Description
ML	Machine Learning.
MLP	Multi-Layer Perceptron.
MSE	Mean Square Error.
NN	Neural Network.
ok-NN	ordinal k-Nearest Neighbor.
OCI	Ordinal Classification Index.
oNN	ordinal Neural Networks.
OR	Operations Research.
ORT	Outranking Relation Theory.
oSVM	ordinal Support Vector Machine.
oTree	ordinal decision Tree.
OVA	One-Versus-All.
OVO	One-Versus-One.
PCA	Principal Component Analysis.
RBF	Radial Basis Function.
ROC	Receiver Operating Characteristic.
SBC	Single Binary Classifier.
SINPATCO	System for Intelligent Diagnosis of Pathologies on the Vertebral Column.
SMAA	Stochastic Multicriteria Acceptability Analysis.
SOM	Self-Organizing Map.
SRM	Structural Risk Minimization.
SVM	Support Vector Machine.
UTA	Utility Additive Functions.

Part I

Introduction

Chapter 1

Introduction

Decision support systems are becoming ubiquitous in many human activities, most notably in finance and medicine where automatic models are being developed to imitate, as closely as possible, the usual human decision. Within this context, classification is one of the most representative predictive learning tasks. Traditionally, it consists in constructing rules which discriminate positive from negative, or malign from benign cases depending on the scenario in analysis. In a simple way, the classifier is developed to partition the feature space in two regions, discriminating between the two classes. Modeling a learner is performed by gathering knowledge from data with different features, attributes or criteria thereby constituting the feature space. The insight gained will make possible the estimation of a mapping from the feature space into a finite class space. Depending on the cardinality of the finite class space we are left with binary (e.g., positive and negative) or multiclass classification problems. In more complex situations, one has to deal with data where, the presence or absence of a "natural" order among classes, will separate nominal from ordinal problems. This stratification is depicted in Figure 1.1.

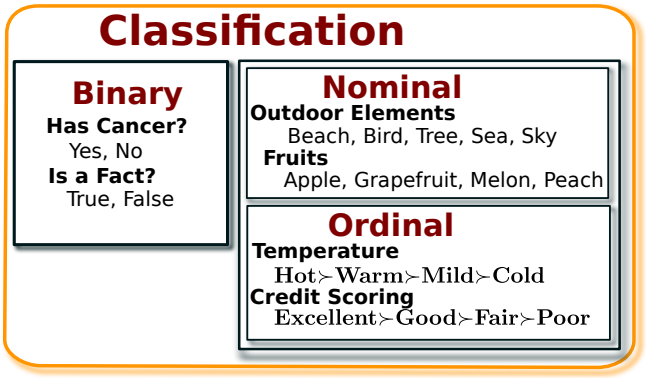


Figure 1.1: Classification problem is divided into binary and multiclass. The latter is further subdivided into nominal and ordinal.

The importance of ordinal data is clear. Nowadays, industry tries to cope with current technological advancements. Hence, more and more personalized products and services to a wider audience are being commercialized. Problems like credit scoring where the system evaluates the capability of one default his debts (Doumpos et al., 2002; Doumpos and Pasiouras, 2005; Xu et al., 2009) by grading a customer credit profile in the scale $Excellent \succ Good \succ Fair \succ Poor$, movies suggestion (Delannay and Verleysen, 2008), breast cancer diagnosis (Cardoso and Cardoso, 2007), or gene analysis through the analysis of hyperactivity on certain proteins (Presson et al., 2011; Pyon and Li, 2009), are some examples of ordinal problems. As a result, new and robust computational methods capable to unravel reasoning's behind decisions also led to new theoretical developments. Moreover,

recent studies have emphasized that the definition of an ordinal classifier leads to better, generalized learners. Furthermore, methods for learning ordinal data have been recently seen as a generalization of some multicriteria techniques (Angilella et al., 2010).

In fact, learning multicriteria (MC) from data has recently gathered a substantial attention. Such trend has its reasons much due to the diverse set of applications from different domains as management (Lahdelma et al., 2002; Rietveld and Ouwersloot, 1992), financial (Doumpos et al., 2002; Doumpos and Pasiouras, 2005) and medicine (Belacel, 2000; Tagliafico et al., 2009), to name a few. Consequently, the very diversity of the multicriteria learning research topic led to a discussion and proposals in several different topics. Decision Analysis (DA), Machine Learning (ML) and statistics/econometrics are some of them. Hence, a rich terminology can be found due to this diverse fields of study. Sorting, ranking, dominance, among others, are some of the many terms referring to multicriteria methods. Even though almost all share the same fundamental principles, it is on the methods assumptions that most differences occur.

This thesis focuses on four main parts: *Measuring Performance of Ordinal Classifiers*, *Multicriteria Learning Models for Ordinal Data*, *Reject Option on an Ordinal Setting* and their usage in two *Medical Applications*.

1.1 Motivation and Objectives

Learning on ordinal data has challenged many researchers to unfold the natural structure of the problem which, at the end, could lead to better performance results when compared with standard learning mechanisms. Considering beyond performance, the development of learning algorithms specific for the ordinal data problem can lead to simpler classifiers. In doing so, it will be possible to capture all important factors with key roles in the classes discrimination. This will result in better generalization capabilities for the learning algorithms developed under these settings.

Despite the rich collection of algorithms presented in the literature concerning to the ordinal data problem (e.g. Cardoso and da Costa (2007); Cheng et al. (2008); Waegeman et al. (2008)), different improvements can be performed. Existing techniques use mappings to convert ranks into real values (Shashua and Levin, 2003) which makes learners sensitive to rank representation than their ordering or are too complex. In general, this is very difficult and makes learners sensitive to the rank value than their pairwise ordering. Some do not totally incorporate or effectively use the additional information of order in the classifier construction (Cheng et al., 2008; Frank and Hall, 2001). Or, by requiring specific optimization algorithms during the classifier construction, they discard classification algorithms that already have been introduced specifically for binary problems (Cardoso and da Costa, 2007; Frank and Hall, 2001). Other approaches (e.g., Potharst and Bioch (2000); Potharst and Feelders (2002)) explore the interpretability capability by investigating the data monotonicity though having as drawback the limitations of the generalization capabilities or the requirements of substantial amounts of data.

How to measure the performance of these learners presents another challenge. Some metrics assume classes equally costly and others disregard order. More recent improvements (Baccianella et al., 2009; Gaudette and Japkowicz, 2009) still do not fully tackle the ordinal data classification models performance problem. By only looking at the relative order relation between the 'true' and 'predicted' values and by being still dependent on the values used to represent the classes, metrics cannot guarantee a fair comparison among competing systems.

Other paradigm is motivated by the fact that even though decision support systems are becoming ubiquitous in many human activities, for instance, prediction of insurance companies' insolvency, has arisen as an important problem in the field of financial research. This

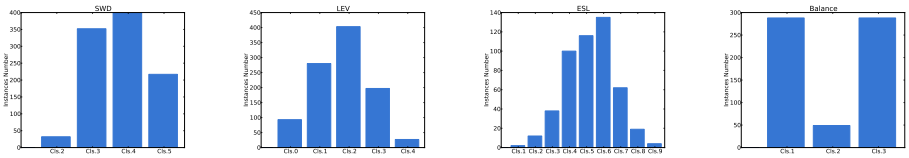
urges the need of automated systems capable to provide decisions as alternative, complementary or as first opinion in many applications. Mostly, in dynamic environments where learning complex items from distinct classes can lead to erroneous outcomes which enhance the requirements to deploy decision support systems capable to label critical items for manual revision. The ordinal data problem can be naturally extended to this scenario where critical items labeled for manual revision are in-between decision classes.

In a nutshell the main objectives of this thesis are:

1. To introduce a new metric which properly considers the ordinal classifiers performance.
2. To develop new learning algorithms appropriate for the ordinal data problem.
3. To propose a new concept of ordinality and new methodologies capable to be interpretable in this new ordinal context.
4. To explore the ordinal data problem in the reject option scenario.

1.2 Datasets

For the experimental study applied to the algorithms under evaluation to the classification of real data, we used mostly the available data on the Weka datasets website and on the UCI Machine Learning repository[1]. In Figure 1.2 it is depicted the classes frequencies for four



(a) Frequency for `SWD` dataset. (b) Frequency for `LEV` dataset. (c) Frequency for `ESL` dataset. (d) Frequency for `Balance` dataset.

Figure 1.2: Real datasets frequency values.

ordinal problems. The first dataset, `SWD`, contains real-world assessments of qualified social workers regarding the risk facing children if they stayed with their families at home and is composed by 10 features and 4 classes. `LEV` dataset contains examples of anonymous lecturer evaluations, taken at the end of MBA courses and is composed by 4 features and 5 classes. These datasets contain 1000 examples each.

Another dataset which we worked on was the `ESL` dataset containing 488 profiles of applicants for certain industrial jobs. Features are based on psychometric tests results and interviews with the candidates performed by expert psychologists. The class assigned to each applicant was an overall score corresponding to the degree of fitness for the type of job.

`Balance` dataset available on UCI machine learning repository was also experimented. Created to model psychological experimental results, each example is labeled as having a balance scale tip to the right, left or balanced. Features encompass on left and right weights, and distances.

Finally, it was also used the `Letter AH` dataset composed of $20,000$ instances with 16 features describing the 26 capital letters. Each instance is mainly defined by statistical moments

[1]For more information, please see: http://www.cs.waikato.ac.nz/~ml/weka/index_datasets.html and http://archive.ics.uci.edu/ml/.

and edge counts. In our experiments we used a subset of the whole dataset comprehending only the discrimination of the letter A versus the letter H. As opposite to the aforementioned datasets, `Letter AH` dataset was used only as benchmark for the binary reject option problem.

1.3 Contributions

This thesis contributed with new methods for the improvement of the multicriteria learning on ordinal data:

1. A new metric for ordinal classifiers.
2. A new SVM methodology for ordinal classification.
3. A new concept of ordinality and new methodologies exploring this concept on DT and k-NN through global constraints.
4. Development of new reject option methods adapted for the ordinal data problem.

List of Publications Related with the Dissertation

The work related with this thesis resulted in the submission of the following articles:

- Ricardo Sousa, Irina Yevseyeva, Joaquim F. Pinto da Costa, and Jaime S. Cardoso. Multicriteria Models for Learning Ordinal Data: A Literature Review. In Xin-She, editor, *Artificial Intelligence, Evolutionary Computation and Metaheuristics (AIECM) –In the footsteps of Alan Turing (Turing 2012)*. Springer, 2012.
- Ajalmar R. R. Neto, Ricardo Sousa, Guilherme Barreto, and Jaime S. Cardoso. Diagnostic of Pathology on the Vertebral Column with Embedded Reject Option. In *Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2011.
- Ricardo Sousa and Jaime S. Cardoso. Ensemble of Decision Trees with Global Constraints for Ordinal Classification. In *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2011.
- Jaime S. Cardoso and Ricardo Sousa. Measuring the Performance of Ordinal Classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(8): 1173–1195, 2011.
- Jaime S. Cardoso and Ricardo Sousa. Classification Models with Global Constraints for Ordinal Data. In *Proceedings of The Ninth International Conference on Machine Learning and Applications (ICMLA)*, 2010.
- Joaquim F. Pinto da Costa, Ricardo Sousa, and Jaime S. Cardoso. An All-at-Once Unimodal SVM Approach for Ordinal Classification. In *Proceedings of The Ninth International Conference on Machine Learning and Applications (ICMLA)*, 2010.
- Ricardo Sousa, Beatriz Mora, and Jaime S. Cardoso. An Ordinal Data Method for the Classification with Reject Option. In *Proceedings of The Eighth International Conference on Machine Learning and Applications (ICMLA)*, 2009.

And the following articles are awaiting the submission outcome:

- Ricardo Sousa and Jaime S. Cardoso. The Data Replication Method for the Classification with Reject Option. (**submitted**).
- Ricardo Sousa, Ajalmar R. da Rocha Neto, Jaime S. Cardoso, and Guilherme A. Barreto. Self-Organizing Maps for Classification with Reject Option. (**submitted**).

1.4 Structure of the Dissertation

This thesis is divided in nine Chapters each one describing the work conducted during the last four years. The remainder of this Section presents the main motivations, benchmark datasets and thesis contributions. Afterwards, a literature review of existing studies on ordinal data unfolds a myriad of several different methods either on Operations Research (OR) or Artificial Intelligence (AI) presented in Chapter 2. In this review, we describe several different techniques presented for over more than five decades on OR and AI disciplines applied to multicriteria problems.

In Chapter 3 we first show that existing measures for evaluating ordinal classification models suffer from a number of important shortcomings. For this reason, we propose an alternative measure defined directly in the confusion matrix. We argue that an error coefficient appropriate for ordinal data should capture how much the result diverges from the ideal prediction and how "inconsistent" the classifier is in regard to the relative order of the classes. The proposed coefficient results from the observation that the performance yielded by the Misclassification Error Rate coefficient is the benefit of the path along the diagonal of the confusion matrix.

A second aspect which was identified concerns to the myriad of schemes for multiclassification with SVM where little work has been done for the case where the classes are ordered. We claim that standard methods usually construct a nominal classifier and define the order afterwards generating rules with ambiguous decision regions. Therefore, in Chapter 4 a new SVM methodology is devised based on the unimodal paradigm with the All-at-Once scheme for the ordinal classification. In the same way, ordinal decision trees have not evolved significantly where conventional trees for regression settings or nominal classification are commonly induced for ordinal classification problems. Claiming that a decision tree consistent with the ordinal setting is often desirable to aid decision making in Chapter 5 we introduce a new rationale to include the information about the order in the design of a classification model. Such was attained by encompassing the inclusion of consistency constraints between adjacent decision regions which were instantiated in a decision tree and in a nearest neighbor algorithm.

As mentioned, decision support system are taking charge in many operations where an human expert was usually the responsible one. A particular example happens in medicine where in the last decades we have witnessed the development of advanced diagnostic systems as alternative, complementary or a first opinion in many applications (Bellazi et al., 2007). Notwithstanding, real world problems still pose challenges which may not be solvable with satisfactory results by many of the existent learning methodologies (Wolpert, 2001). Or, in other words, the automation of decisions can still lead to many wrong predictions. Therefore, systems where the automation occurs only those decisions which can be reliably predicted, labeling the critical ones for a human expert to analyze, is attractive, leading to the development of classifiers with a third output class, the reject class. Having this in mind, in Chapter 6 we present two different proposals on SOM to act as supervised classifiers with reject option. Then, in Chapter 7, a paradigm initially proposed for the classification of ordinal data problems was adapted for the classification problem with reject option. This technique reduces the problem of classifying with reject option to the standard two-class problem. The method here introduced is then mapped into SVM and Neural Network (NN). Finally, the framework is extended to ordinal data problem with reject option.

In Chapter 8, an assessment over the techniques presented along this thesis is conducted into two medical applications: Breast Cancer Conservative Treatment (BCCT) and System for Intelligent Diagnosis of Pathologies on the Vertebral Column (SINPATCO). Finally, conclusions and lines for future research of this dissertation are given in the last chapter.

Chapter 2

Background Knowledge*

2.1 Terminology and Concepts

Learning Multicriteria (MC) on ordinal data has a strong connection with OR and AI (Zopounidis and Doumpos, 2002). Albeit being conceptually different topics, there is an intrinsic connection among them. OR comprises several different areas of study such as decision analysis, mathematical programming among others. Whereas, AI can be described as being composed by machine learning, pattern recognition, data mining (Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, Wei Wang, 2006) etc. Within each area there are concepts borrowed from one another. For instance, machine learning vastly uses techniques from mathematical programming and statistics since its early days (Fisher, 1936; Vapnik, 1998) (Figure 2.1 depicts some of these relations). How these topics interact with each other is not within the scope of this chapter. It is the purpose of Figure 2.1 to illustrate the broad aspects of the area in study. Its usage is so broad that a full coverage is not possible. However, it is interesting

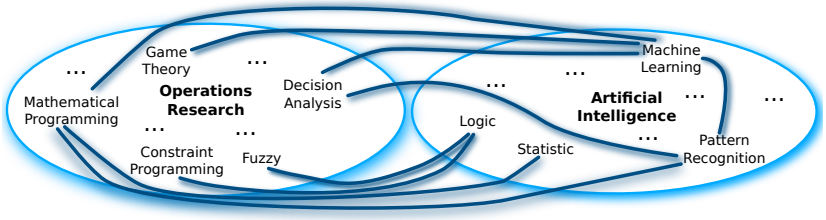


Figure 2.1: Illustration of the different fields that overlap with operations research and artificial intelligence.

to know how MC methods have been used in data analysis to represent knowledge. Such is done in order to understand reasoning's behind decisions (Olafsson et al., 2008), outcome prediction (Doumpos and Zopounidis, 2004), in mimicking behaviors (McGeachie and Doyle, 2004) and planning (Kangas et al., 2003; Rietveld and Ouwersloot, 1992).

Even though MC methods have been thoroughly studied, not much effort has been employed on the particular case where data is presented in a "natural" order. Let us consider the credit score problem. A bank assigns a score of *Excellent* to a client given his wage, good payment history in previous mortgages and the number of credits at the time of the evaluation. The score assessment is clearly rendered over the different criteria: Wage, payment history, among others. Ideally, one wants to find the best function that can capture all this information in order to output the expected outcome.

*Some portions of this Chapter appeared in Sousa et al. (2012).

Definition 2.1 (Classification on Ordinal Data Problems). *(Belacel, 2000; Cardoso and da Costa, 2007; da Costa et al., 2008; Meyer and Roubens, 2005; Mousseau et al., 2001; Zopounidis and Doumpos, 2002) Classifying on ordinal data problems consists on finding the best mapping* $f : \mathbb{R}^d \to \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ *of a given pattern,* $\boldsymbol{x} \in \mathbb{R}^d \subset \mathcal{X}$*, to a finite set of classes, where* $\mathcal{C}_1 \prec \ldots \prec \mathcal{C}_K$*.*

Pattern $\boldsymbol{x}$ is also referred as instance, example or alternative. Moreover, $\boldsymbol{x}$ can be represented in a vector fashion where each entry is identified as a feature, attribute or criterion, i.e., $\boldsymbol{x} = \{x_1, \ldots, x_d\}$. A dataset is a tuple consisted of N patterns and its target classes (or outcomes), $\mathcal{D} = \{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$.

Literature usually differentiates attribute and criterion (Greco et al., 2001). Consequently, the problem in analysis can be substantially different. In an ordinal data problem as the credit scoring, an alternative (to which client a loan should be granted) is represented by several attributes each one representing a level of *importance* of the Decision Maker (DM) (the bank). Here, criteria are used instead of attribute being the former more adequate for the ordinal problem (Greco et al., 2001; Waegeman et al., 2009).

The usage of the term ranking is also common in the MC field. However, such term is usually mentioned to other subjects aside classification.

Definition 2.2 (Ranking). *(Cao-Van and De Baets, 2003; Cossock and Zhang, 2006) A ranking problem consists on finding the best mapping* $f : \mathbb{R}^d \to \{\mathcal{R}_1, \ldots, \mathcal{R}_L\}$ *of a given pattern,* $\boldsymbol{x} \in \mathbb{R}^d \subset \mathcal{X}$*, to a finite set of ranks, where* $\mathcal{R}_1 \prec \ldots \prec \mathcal{R}_L$ *is* not *pre-defined.*

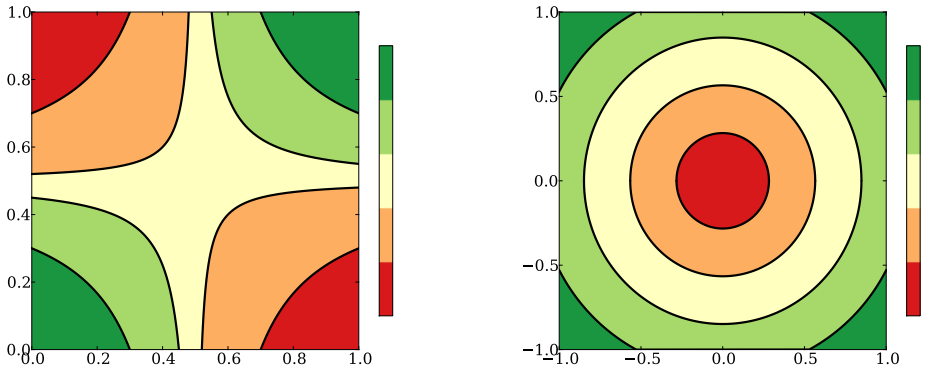
There are subtle differences between the two problems. Whereas in classification the order between classes is already defined and all patterns have to be assigned into at most one class, in ranking such does not hold. Think for instance on the Google™ or Yahoo™ search engines. When entering a search query, the result can vary from user to user for the same query. The search engine will look on its database and will rank the results according to, for instance, user search history. Ranking approaches however go beyond the subject of this chapter.

Depending on the problem, criteria can also represent a magnitude of importance or unimportance, a ratio, among others. This can generate datasets where order may not be explicitly represented. Different works tackled the ordinal problem assuming that data were monotone, i.e., where both criteria and classes were assumed to be ordered (Blaszczynski et al., 2009; Duivesteijn and Feelders, 2008; Potharst and Feelders, 2002). Nevertheless, we argue that monotonicity constraint does not need to be verified. The following synthetic datasets are perfect representatives of an ordinal problem. To each point in Figure 2.2a was assigned a class y from the set $\{1, 2, 3, 4, 5\}$, according to

$$\begin{aligned} y = \min_{r \in \{1,2,3,4,5\}} \{r : b_{r-1} < 10(x_1 - 0.5)(x_2 - 0.5) + \varepsilon \ < b_r\} \\ (b_0, b_1, b_2, b_3, b_4, b_5) = (-\infty, -1, -0.1, 0.25, 1, +\infty) \end{aligned} \tag{2.3}$$

where $\varepsilon \sim N(0; 0.125^2)$ simulates the possible existence of error in the assignment of the true class to $\boldsymbol{x}$. Data in Figure 2.2b is uniformly distributed in the unit-circle, with the class y being assigned according to the radius of the point: $y = \left\lceil \sqrt{x_1^2 + x_2^2} \right\rceil$. These synthetic datasets are examples where order can not be captured directly in the input space, but in an implicit feature space. We will return to this matter in Chapter 5.

Hence, the following question can be posed: How to capture order? Many models have been proposed towards this goal. But before answering that question, first a brief description of the most commonly used models is required. The following concepts will allow a better understanding of the most recent techniques discussed along this chapter.



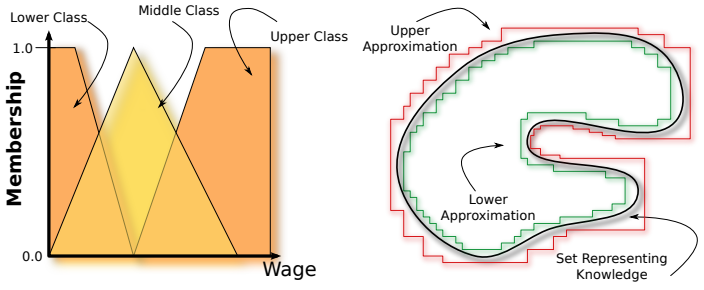
(a) (b)

Figure 2.2: Two synthetic ordinal dataset where the monotonicity property at input data does not hold.

Starting by the OR domain, a classic Multicriteria Decision Analysis (MCDA) approach is done by the representation of a specific aggregation model. Aggregation models are performed by aggregating different value or utility functions in order to be expressed by a single criterion. One aggregation model that we can think of is, for instance, the mean: $\frac{1}{d}\sum_{j=1}^{d} x_j$. The use of utility vs. value depends upon the problem. Whereas, utility functions are used in stochastic problems, value functions are used in deterministic ones (Miettinen, 1999). In brief, an aggregation model is a function $\mathcal{U} : \mathbb{R}^d \rightarrow \mathbb{R}$, that maps criteria of the DM onto outcomes (Miettinen, 1999). Utility functions are widely used, where the one presented in Equation (2.4) is one of several other aggregation models. It has the advantage of considering both qualitative and quantitative criteria. The simplest additive case of an utility function is defined as follows

$$\mathcal{U}(\boldsymbol{x}) = \sum_{j=1}^{d} u_j(x_j) \tag{2.4}$$

where $\mathcal{U} \in [0, 1]$. For the interested reader Siskos et al. (2005) present a good description of these methods.



(a) (b)

Figure 2.3: Fuzzy and Rough Set concept illustrations: (a) An example of a membership function that defines a possible economic class problem in a fuzzy set approach; (b) Lower and Upper approximations of a given set which represent the domain knowledge;

Fuzzy set theory is another topic with increasing interest on the scientific community. Its usage is not restricted only to the MCDA problem being however strongly defended thanks to its capability to handle uncertainty (Greco et al., 2006; Jensen and Shen, 2008). In general,

fuzzy set theory presents a fundamental principle which describes a special type of sets which have *degrees of membership* through simple logical operators. Such can be described by any mapping function $\mu(\boldsymbol{x}) : \mathbb{R}^d \rightarrow [0, 1]$. Figure 2.3a) consists of a valid representation for a given membership function. Moreover, it can represent knowledge in a *if ... then* way in a similar way to DT (Kecman, 2001) which will be described shortly.

In much of the works currently present in the literature, fuzzy set theory usually appears along with rough sets. The latter field is however slightly different from the former. Rough Set theory not just handle uncertainty, but also incomplete information which can be present on data (Jensen and Shen, 2008). Even though new approaches on Utility Additive Functions (UTA)[2] already tackle this problem, it has also been stated that rough and fuzzy set theory are complementary because of dealing with different kinds of uncertainty (Greco et al., 2006). It was initially proposed by Pawlak (1982) with the objective to provide a mathematical formulation of the concept of approximated (rough) equality of sets in a given space. In the rough set theory it is assumed that to every object there is an associated amount of information that describes it. This refers to the view that knowledge has a granular structure (Abraham et al., 2009; Greco et al., 2001; 2006; Pawlak, 1997). Therefore, an important characteristic of rough sets theory is the identification of consistent data and assigning them into lower and upper approximations of sets—see Figure 2.3b).

More on the AI domain, in general, one tries to obtain valid generalization rules, classifier, from data. Once a classifier has been designed, one has to assess its performance by estimating the error of the classifier for unseen examples. Classification error is expressed as a misclassification error defined by a "true misclassification rate" (here denoted as $R^*(d)$). $d(\boldsymbol{x})$ is the learner model with input data $\boldsymbol{x}$. Breiman et al. (1998) defines this function as:

Definition 2.5 (Accuracy Estimation). *(Breiman et al., 1998) Take $(\boldsymbol{x}, y)$, $\boldsymbol{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, to be a new sample from the probability distribution $\mathcal{P}(A, j)$; i.e.,*

- $\mathcal{P}(\boldsymbol{x} \in A, y = j) = \mathcal{P}(A, j)$.
- $(\boldsymbol{x}, y)$ *is independent of* $\mathcal{D}$.

Then define

$$R^*(d) = \mathcal{P}(d(\boldsymbol{x}) \neq y) \tag{2.6}$$

But how can $R^*(d)$ be estimated? There are many approaches. One that this work will use is the cross-validation approach. Dataset $\mathcal{D}$ is randomly divided in sub-samples, with the same size as possible, e.g., $\mathcal{D}_1, \ldots, \mathcal{D}_V$. For each v, $v = 1, \ldots, V$, a learning method is applied to the sample $\mathcal{D} - \mathcal{D}_v$, resulting in the $d^v(\boldsymbol{x})$ model. R is then computed as:

$$R^{cv}(d) = \frac{1}{V} \sum_{v=1}^{V} R^{ts}(d^v) \tag{2.7}$$

where R^{ts} is defined as

$$R^{ts}(d^v) = \frac{1}{N_v} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_v} \mathcal{F}(d^v(\boldsymbol{x}_i), y_i) \tag{2.8}$$

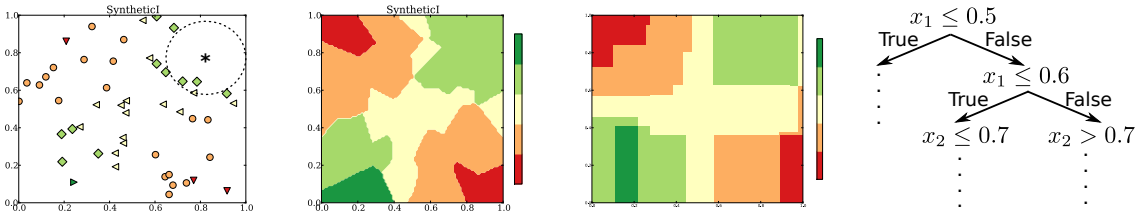
where $N_v \simeq N/V$ and $\mathcal{F}$ any function which penalizes each error[3]. One can now analyze the different learning methods for ordinal data.

k-NN is a simple method that interestingly has not been explored enough in the MCDA setting until very recently. It consists of a non-parametric method with the main objective to

[2] UTilitès Additives (Siskos et al., 2005)

[3] The l_{0-1} *loss* function is the most commonly used one, i. e., $\mathcal{F}(a, b) = I(a \neq b)$ being I the identity function.

estimate the density function from sample patterns (Duda et al., 2001). It extends the local region around a data point $\boldsymbol{x}$ until the k^{th} nearest neighbor is found. The most represented class in the k-closest cases defines the predicted class. Figure 2.4a-b) illustrates such procedure. DT are another method that captured some interest for tackling MCDA problems,

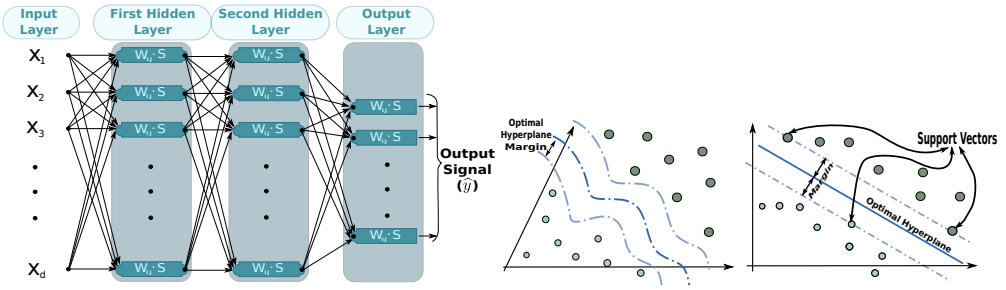


(a) (b) (c) (d)

Figure 2.4: k-NN and DT methods. (a) A test pattern (illustrated as a star) composed by two features checks for, in this example, two closest labeled patterns in order to determine its own class; (b) Prediction over the whole feature domain for an 2-NN on the training data shown in (a); (c) A DT discriminates the feature space (a) by rectangles; (d) A sample of the decision tree for (c).

specially on the OR domain. DT classify a pattern through a sequence of questions where the next question depends on the answer to the previous one. These trees are constructed as logical expressions as is illustrated in Figure 2.4c-d). This ability generates a powerful data analysis tool capable to obtain interpretable results (Duda et al., 2001). Nodes are consecutively split where a stop-splitting rule is required that controls the growth of the tree.

NN are another kind of learning models. Multi-Layer Perceptron (MLP) is the most commonly used. A MLP is a layered structure consisting of nodes or units (called neurons) and one-way connections or links between the nodes of successive layers, such as the structure of Figure 2.5a). The first layer is called the input layer, the last layer is the output layer, while the ones in the middle are called the hidden layers. Input layer of neurons is only a vector where all data are introduced triggering the learning process. Data propagates through the network in a forward direction, on a layer-by-layer basis. Layers are constituted by several neurons which commonly have non-linear and differentiable activation functions. SVM are



(a) (b)

Figure 2.5: MLP and SVM methods: (a) Example of a MLP. This MLP is composed by 2 hidden layers, one input and output layer; (b) A two dimensional dataset is augmented to a higher feature space.

another popular learning mechanism. In its simple form, SVMs uses a linear separating hyperplane to create a binary classifier with a maximal margin. In cases where data can not be linearly separable, data are transformed to a higher dimension than the original feature space (see Figure 2.5b). Such is done by choosing a kernel function, representing the inner product

in some implicit higher dimension space. Formally, a kernel function is defined by $k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}')$. This transformation ($\phi$) can be achieved by several non-linear mappings: e.g., polynomial, sigmoidal and radial basis functions. However, in a multiclass problem the usage of a binary SVM classifier can be limited. In order to improve this some heuristics and new formulations were proposed as an extension to the binary classification problem. Some of them encompass the One-Versus-One (OVO), One-Versus-All (OVA), Decision Directed Acyclic Graph (DDAG), single optimization formulation, among others. Basically, OVO consists on the design of K(K-1)/2 binary classifiers where one class is discriminated against another. Similarly, and as the name suggests, OVA consists on the design of K binary classifiers where one class is compared against all the others. Likewise the former heuristic, DDAG, follow a similar procedure. The major difference is that prediction is made in a graph path manner where each node corresponds to a given binary classifier. In a completely different scenario, there are also techniques that try to define a single optimization problem to solve the multiclass problem on SVM (Cardoso and da Costa, 2007).

This Section provided some key concepts regarding techniques for learning from data. Knowing that still much more has to be covered, the interested reader is advised to OR and AI textbooks (Bishop, 2007; Duda et al., 2001; Haykin, 2008; Jensen and Shen, 2008; Lee, 2004; Russell and Norvig, 2003) for more information. Next Sections will describe different methods using some of the aforementioned methodologies for learning multicriteria models on ordinal data problems.

2.2 Multicriteria Decision Analysis

DA is an important field within OR. It helped researchers to devise new approaches in order to analyze and interpret human's reasoning. Specifically, when handling several usually conflicting criteria towards an outcome. Such methods are generally composed by five phases depicted in Figure 2.6.

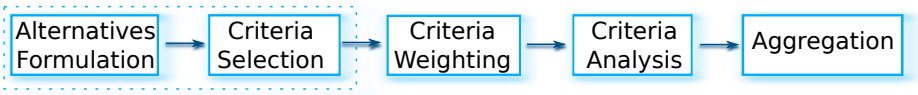


Figure 2.6: Common Diagram of MCDA Methods (Ustinovichius et al., 2007; Wang et al., 2009b).

This Section will review multicriteria decision methods for ordinal data problems. Alternative formulation and criteria selection is usually defined by a set of experts or DMs (Ustinovichius et al., 2007; Wang et al., 2009b) and can depend on the problem in analysis. On the other hand, a given importance (weight) can be defined to each criterion whether through subjective or objective methods. After every criteria being considered, the analysis takes place. In general, MCDA tries to learn about users preferences encompassed in the different criteria considered. One key aspect of such methods is that they do not rely on any statistical assumptions (Zopounidis and Doumpos, 2000). Such highly contrasts with the topic which will be reviewed in Section 2.3. These two views can mark great differences on both topics, but as one shall see, there are points of connection between these two fields. In doing so, one can identify a trend towards filling the gap between OR and AI on MCDA. Finally, all criteria which were considered are then aggregated in order to define a plausible outcome.

It is important to stress that this work is mostly concerned with ordinal data. Hence, not all topics within MCDA can be covered in this Section. The interested reader is referred to Hillier et al. (2004), Figueira et al. (2005a), Taha (2006) and Zopounidis and Pardalos (2010) for more information.

2.2.1 Multicriteria Methods

From Figure 2.6, one can define methodologies which follow the whole process. Analytic Hierarchy Process (AHP) is one of such kind of frameworks (Saaty, 1990). After having the problem analyzed and criteria selected, usually performed by an expert (or DM), it considers through an hierarchical approach each criteria (Saaty, 1990). However, recent reviews have argued that AHP results may not be the most desirable ones (Ishizaka and Labib, 2009; 2011; Ishizaka et al., 2011). Mentioning that there is no clear evidence that AHP provides its users with their "best" decision (Ishizaka et al., 2011), or in more detail, identifying the limitations in each step on the process (Ishizaka and Labib, 2009). Even though the Analytic Network Process (ANP) was introduced as a generalization over AHP (a feedback network capable to adjust weights) (Ishizaka and Labib, 2011; Saaty and Vargas, 2001), few work has been done for the ordinal case.

ELECTRE (Doumpos and Zopounidis, 2002; Roy, 1991) and PROMETHEE (Doumpos and Zopounidis, 2002; 2010; Figueira et al., 2005b) are two well known methods that, like AHP, can consist at most by the five steps illustrated in Figure 2.6 (Ishizaka and Labib, 2009). Both techniques arose from the foundations of the Outranking Relation Theory (ORT) (Doumpos and Zopounidis, 2002). In simple words, it consists of checking the outranking relation among instances which permits to conclude whether an instance $\boldsymbol{x}^{(p)}$ outranks instance $\boldsymbol{x}^{(q)}$. In other words, an instance $\boldsymbol{x}^{(p)}$ will be more adequate for the DM than $\boldsymbol{x}^{(q)}$. This is achieved if there are enough statements to confirm (concordance) or to refute that (discordance). The two aforementioned methods require some preferential information which has to be defined by the DM. However, it may be difficult for the DM to understand the meaning of the preferences (Iryna, 2007). To overcome this, different improvements over the methods have been conducted. One of them was through the usage of evolutionary algorithms.

Evolutionary algorithm (EA) came in a way to reproduce Darwin's theory of the survival of the fittest. EA are also referred as populational meta-heuristics meaning that they work on the population space of solutions (Branke et al., 2008). EA generally encompasses on three major steps: 1) Gather a set of solutions; 2) Select a possible subset of candidates on that set of solutions and allow them to reproduce. Reproduction consists mainly on creating new solutions from the selected ones by crossover and mutation operators; 3) Finally, the process is repeated for the set of new solutions until a stopping criteria is achieved. Siwik and Natanek (2008) in (Siwik and Natanek, 2008; and references therein) introduced an elitist evolutionary agent[4] system to solve multicriteria optimization problems. By trying to reproduce biological mechanisms, an elitist group is introduced in the evolutionary architecture proposal. The final solution identified by the elitist group would indicate the desirable one which will dominate other possible solutions identified by other groups. Some hybrid approaches are also present in the literature (Doumpos et al., 2009; Fernandez et al., 2009). In (Fernandez et al., 2009) an outranking combined with an EA was proposed thanks to an indifference measure. Since preference modeling is cumbersome, authors used a population based meta-heuristic to generate the best solutions. An agent would then decide the best one. An approach proposed by Doumpos et al. (2009) comprehends the usage of concordance and discordance measures into a credibility index of an outranking method. This will assess the outranking relation among several alternatives. Since incomparable relations can occur, an EA is used to infer the parameters of the outranking method.

In a complete different setting, constraint programming tries to explore all possible combination of solutions thoroughly. Despite this being highly computational expensive, Junker (2004; 2008) argues that an interactive approach has its advantages over state of the art techniques. It is also claimed that current existing methods do not express a clear explanation of the reason for one alternative being more preferable than another. In other words, a per-

[4]In a simple way, an agent is a solution vector generated by some sub-optimal learning method.

formance of 98% does not express which option is the best based on the original preferences. Using a special utility function to define preferences order in (Junker, 2008) a lexicographic optimal scheme is applied. Since lexicographic approach establish some ranking over the preferences order (Ehrgott, 2000; Junker, 2008), authors also permute the order of alternatives search. Bouveret and Lemaître (2009) explores the idea in which characterizes good solutions where multiple criteria have to be handled through the use of lexicographic algorithms.

Other methods incorporate cooperative algorithms which take part in the learning process from diverse sources of information and by different decision criteria (Dembczynski et al., 2007; Kotlowski et al., 2008). Methods with such properties are named Dominance-based Set Approach (DRSA) (Dembczynski et al., 2007) which deal with the problem of multicriteria classification using maximum likelihood estimation. The problem is then solved by an optimal object reassignment algorithm. In Kotlowski et al. (2008) a stochastic DRSA approach is introduced. The rationale behind this method is to assess object class probability from an interval of classes.

Rough set theory is another field that one can count with when tackling MCDA. One interesting aspect is that rough set has the ability to produce a model of rule induction similar to data mining, knowledge discovery and machine learning (Greco et al., 2006). In Greco et al. (2006) authors extend the fuzzy set theory to rough sets theory in order to avoid as much as possible meaningless transformation of information. Rule induction is made through decision rules induced from dominance-based rough approximations of preference-ordered decision classes (Greco et al., 2001).

Let us now analyze in more depth contributions made to each node in the multicriteria methods process.

Criteria Weighting

Criteria weighting can be considered one of the most important steps for the decision maker. Once it weights the importance of each criterion, acting as a trade-off between criteria (Iryna, 2007) that will be considered in the decision process, subtle changes can produce different outcome (Wang et al., 2009a).

Methods for weighting criteria encompass equal weights, rank-order and hybrid approaches where after some considerations from the DM, weighting can be performed by a subjective or objective method (Wang et al., 2009a;b). Equal weights ($w_j = 1/d$) is not valuable once relative importance among the criteria is ignored. Remains rank-order weighting approaches and their derivations to overcome these limitations. Another issue is that when dealing with uncertainty or incomplete information in any decision problem, the DM may not be reliable to define her/his criteria accurately. One way to handle this type of information is to represent preferences by a suitable distribution using Stochastic Multicriteria Acceptability Analysis (SMAA) methods. Several methods have been proposed in the literature—e.g. Lahdelma et al. (2003), Tervonen and Lahdelma (2007), Lahdelma and Salminen (2009) and Durbach (2009) to name a few. SMAA-O proposed in Lahdelma et al. (2003) was an extension of SMAA works (Tervonen and Figueira, 2008; Tervonen and Lahdelma, 2007) applied to ordinal (and cardinal) criteria. The problem is that, in the authors approach, an ordered criteria can not be used directly in MC model. Therefore, it is assumed that exists a cardinal measure that corresponds to the known ordinal criteria and by considering consistent mappings between ordinal and cardinal scales, they randomly simulate such mapping through a Monte Carlo iterations. Or in other words, ordinal data is converted into stochastic cardinal data by simulating consistent mappings between ordinal and cardinal scales that preserve the given labels. In SMAA literature review work of Tervonen and Figueira (2008) they claim that such simulations are not necessary since cardinal values can be interpreted directly.

Criteria Analysis

To the best of our knowledge, one of the first works in criteria analysis was proposed by Herstein and Milnor (1953) where an axiomatic approach was carried out. A set of mathematical axioms was presented in this work to measure preferences order. Maccheroni in their work (Maccheroni et al., 2006) explore the possibility where DM does for certain her/his preferences being therefore unable to rationalize her/his choices.

As previously mentioned, in the outranking approaches inconsistencies may arise when the preferences which are learned by given instances can not be expressed through a model. Belacel (2000) proposes a construction of partial indifference indexes comparing pairs of preferences according to some criteria, aggregating them according to a concordance and non-discordance concept. Mousseau et al. (2001) suggest to discard contradictory information from the preferences through an iterative aggregation-disaggregation scheme.

A number of variants of UTA (Siskos et al., 2005) have been proposed in the literature over the last two decades and many works have been published concerned to this subject (Beuthe and Scannella, 2001; Greco et al., 2008; Hastie and Tibshirani, 1986; Köksalan and Özpeynirci, 2009; Zopounidis and Doumpos, 2002). One related to ordinal problem was proposed in Zopounidis and Doumpos (2000). In this work, additive functions are used discriminating the preferences being evaluated from those that are not. Trying to go through a more natural way to human thinking over their outcomes or goals, some methods also based on utility functions have recently been proposed (McGeachie, 2002; McGeachie and Doyle, 2002; 2004). In this method, the authors developed a model to express logic of preferences in order to determine which of two outcomes is more preferable.

Aggregation

As mentioned, aggregation models are one of the most studied methods within Multicriteria Decision Analysis. For instance, in our credit scoring problem a model has to be designed to aggregate wage, payments history, age among others so that it can express the credit score profile of a given client. However, this approach implies that those functions have to be, among others, *monotone* (Marichal, 1998). Most important of all, the aggregation model has to be able to evince the importance of a criterion (done in the criteria analysis step), but also the interaction and compensation effects between criteria (done in the weighting step) (Huédé et al., 2006). Meaning that one has to design a model such that it can assign weights to a subset of possible criteria in order to capture these relations (Huédé et al., 2006; Sridhar et al., 2008).

As one saw until now, multicriteria methods encompass a variety of different approaches. Many of them address this problem through classification techniques using some sort of aggregation model (Doumpos and Zopounidis, 2004; Figueira et al., 2005a). Afterward, restrictions are then defined to the problem in question. However, despite the existence of the myriad of techniques, many pass through the definition of some objective function which can be delved through mathematical programming approaches.

In Zopounidis and Doumpos (2000) a Multi-group Hierarchical Discrimination (M.H.DIS) method is defined. An error minimization and clear group discrimination utility function is presented. Then, two optimization stages are conducted to avoid high computational complexity of Mixed Integer Programming (MIP) problems with many binary variables. An extension of this work is presented in Doumpos et al. (2002) where the estimation of the additive utility functions in aforementioned work is accomplished through mathematical programming techniques. Two linear and one mixed-integer programs are used in M.H.DIS to estimate optimally the utility functions.

Unsupervised approaches such as the K-Means algorithm or Agglomerative Hierarchical Clustering (AHC) can also be used. The latter performs a hierarchical clustering where given

individual clusters it can merge or split clusters until a stopping criteria is achieved. Given the utility matrix, authors employ clustering algorithms to form groups of alternatives (e.g., customers) with closely related preferences (Lakiotaki et al., 2011; 2009). However, in this phase little or no usage of the ordered criteria is explored.

2.3 Inductive Learning Algorithms

Inductive learning describes a very powerful field of research where machine learning (ML) lies. In ML one tries to obtain valid generalization rules from data instead of the deductive learning approaches where one is already presented with a formalization of the world and constructs (deducts) reasonable conclusions that cover our initial assumptions. Being also referred as a technique that *learns by examples* (instances), it has been another thoroughly studied field which is composed by two main research topics: Regression and classification. A schematic of such problems and some real world scenarios are depicted in Figure 2.7.

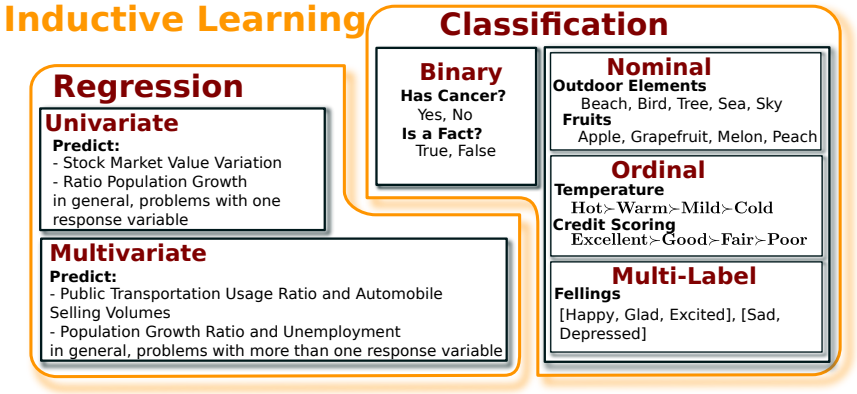


Figure 2.7: Inductive Learning encompasses on two major research topics: *Regression* and *classification*. Both thrives on finding the best function that explains our data. The former renders the reasoning's on a continuous domain whereas the latter on a discrete (finite) domain. Each one is divided in other subtopics being their thoroughly analysis more appropriate for other textbooks (Bishop, 2007; Duda et al., 2001; Haykin, 2008) and here depicted just for context.

Learning mechanisms that solve ordinal problems have been tackled with both regression and classification strategies. Albeit being fundamentally different, both ordinal regression and ordinal classification methods have thrived among the scientific community, e.g. McCullagh (1980), Herbrich et al. (1999), Frank and Hall (2001), Kramer et al. (2001), Shashua and Levin (2003) and Cardoso and da Costa (2007), to name a few.

The first works that tried to solve the classification of ordinal data were based on generalized linear models, as the cumulative model (McCullagh, 1980). Tutz (2003) presents a generic formulation for semi-parametric models extending therefore the additive models (Hastie and Tibshirani, 1986). In the machine learning community, Frank and Hall (2001) have introduced a simpler process which permits to explore information order in classification problems, using conventional binary classifiers as can be depicted in Figure 2.8. In Herbrich et al. (1999) it is applied the minimal structural risk principle (Vapnik, 1998) to derive a learning algorithm based in pairs of points.

Another way to learn ordering relation is by using classical algorithms of classification or regression and mapping the results into an ordinal scale. Kramer et al. (2001) investigate the use of a learning algorithm for regression tasks—more specifically, a regression tree learner—to solve ordinal classification problems. In this case each class needs to be mapped to a numeric

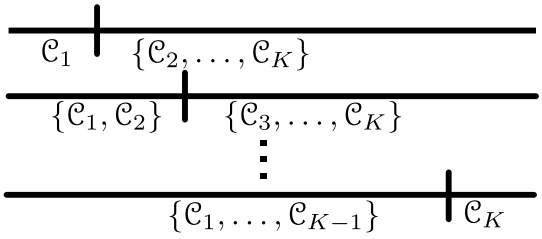


Figure 2.8: Schematic of the proposal presented by (Frank and Hall, 2001). Firstly it is performed a transformation of a K-class problem to a $K-1$ binary class problem. The training of the i^{th} classifier involves the transformation of the K ordinal class into a binary one where the i^{th} discriminator is obtained by separating the classes $\mathcal{C}_1, \ldots, \mathcal{C}_i$ and $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_k$. The i^{th} class represents the test $\mathcal{C}_x > \mathcal{C}_i$.

value. Kramer et al. (2001) compare several different methods for doing this. However, if the class attribute represents a truly ordinal quantity—which, by definition, cannot be represented as a number in a meaningful way—there is no principled way of devising an appropriate mapping and this procedure is necessarily *ad hoc*. Harrington (2003) argues that these type of approaches have many drawbacks as 1) makes regression learners sensitive to rank representation than their ordering and 2) since classification algorithms ignore rank order treating them as classes, it will be required more training data. Consequently, Harrington (2003) presents a perceptron algorithm where its goal it to find a perceptron weight vector $\boldsymbol{w}$ which successfully projects all the instances into the k classes subintervals defined by some thresholds.

Moreover, existing methods incurring ordinal regression approaches fit data in general by a single rule defined by parts through K-1 thresholds (Waegeman et al., 2008). This has a drawback since a mapping is required to convert ranks into real values or *vice-versa*. Hence, determining this mapping function is in general very difficult and makes regression learners more sensitive to rank value than their pairwise ordering. Some of the aforementioned drawbacks were avoided in Shashua and Levin (2003) work where a generalized formulation of SVM applied to ordinal data was proposed. However, such models can be too complex. Cardoso and da Costa (2007) proposed a reduction technique to solve data ordinal problem classification using only one binary classifier. Following this idea, Lin and Li (2009) explored the potential of solving ordinal problems through binary classification methods whereas Cheng et al. (2008) presented an adaptation of the NN towards ordinal problems. In da Costa et al. (2008) an order relation is incorporated among classes by imposing an unimodal distribution. This fundamental principle allowed to delve simpler NN classifiers. Sun et al. (2010) proposed a Kernel Discriminant Analysis (KDA) for ordinal data. Even though authors argued that finding an optimal projection would result in better results, in doing so one would loose its relation to the original features. Hence, in the case of need for interpretable results, through the usage of such methods, one would be unable to understand the reason of the outcome given specific features.

Metric learning is research subject that recently has been gaining increasingly attention, specially in the machine learning community (Weinberger and Saul, 2009; Yang and Jin, 2006; Zhang et al., 2003). The performance of all machine learning algorithms depends critically on the metric that is used over the input space. Some learning algorithms, such as K-Means (K-Means) and k-NN, require a metric that will reflect important relationships between each classes in data and will allow to discriminate instances belonging to one class from others (Rebelo et al., 2011). Schultz and Joachims (2004) and Ouyang and Gray (2008) explored this subject in the ordinal problem. In Ouyang and Gray (2008) by assuming that closer instances in the input space should translate an order of relation, a metric distance is learnt so that pairs of instances are closer than the remainder pairs. However, class label is

discarded in this approach.

Other approaches (Chu and Ghahramani, 2005a;b; Chu et al., 2007; Yu et al., 2006) consisted in probabilistic approaches based on Gaussian processes to learn models for the ordinal problem. In Yu et al. (2006) a collaborative approach is delved towards better, not only in accuracy but also in the context of collaborative preference learning.

Regarding DT for ordinal data, some works consider problems that are monotone, i.e., all attributes have ordered domains. Meaning, if $\boldsymbol{x}, \boldsymbol{z}$ are data points such that $\boldsymbol{x} \leq \boldsymbol{z}$ ($x_i \leq z_i$ for each criteria i) then their classes should satisfy the condition $\widehat{f}(\boldsymbol{x}) \leq \widehat{f}(\boldsymbol{z})$, where $\widehat{f}(.)$ is the labeling function. Potharst and Bioch (1999; 2000) and Potharst and Feelders (2002) propose a method that induces a binary DT from a monotone dataset. Other methods were also proposed for non-monotone datasets (the most likely scenario in the presence of noise) where the resulting tree may be non-monotone. In this scenario, a fuzzy operator was used instead of a entropy function for performance measurement (Dombi and Zsiros, 2005). Works on k-nearest neighbor for ordinal data seems even scarcer. Besides the well-known adaptation of using the median as labeling rule instead of mode for the k labels, literature only presents a modified version of the standard k-NN for the construction of monotone classifiers from data (Duivesteijn and Feelders, 2008). Again, this work continues to be limited by the assumption of monotonocity in the input data.

From the works until now revised, one has encountered several methods that make use of different procedures from operations research field, and other proposals design their learning models so that multicriteria can be rendered in the learning phase. In this setting, multi-criteria assessment is simply performed over a set of diverse unattached reasoning's which renders the desirable outcomes without a clear understanding of which criteria contributed most. To overcome this, Smet and Guzmán (2004) developed a K-Means clustering algorithm in a multicriteria decision analysis perspective.

In this section we have reviewed several learning approaches for the resolution of the ordinal problem. In the end, it is obvious how increasingly this subject has been studied. The reasons can be due to the awareness of its transversal usability in a set of diverse applications. However, due to the background of many researchers, many have tried to solve this problem through regression, classification and ranking methodologies. The work of Fürnkranz and Hüllermeier (2003) in (Fürnkranz and Hüllermeier, 2003; and references therein) despite using a pairwise approach, compared ranking and classification principles in their proposals. In the same way, Lin and Li (2009) were able to establish a relation between ordinal ranking and binary classification. As final remark, one must note how vastly such methods can be employed such it has been explored by Shen and Joshi (2005) and Vanya et al. (2011). In these works, different approaches have been delved towards ranking, ordinal and survival analysis problems. Even though authors performed strict assumptions on data to develop their models, such as monotone data, it still is a good example of the importance of this topic in the inductive learning field.

2.3.1 Feature Selection Algorithms on Ordinal Data

Nowadays, it is relatively easy to solve problems with millions of instances, each of them with a reasonable number of features. However, it is common to have access to datasets with significantly higher number of features than instances leading to the well known problem of the curse of dimensionality. Feature Selection (FS) techniques provide the means to overcome this issue by identifying the most valuable features so that good and simple class discrimination models can be obtained. Furthermore, a noise reduced dataset can be achieved since these methods can "clean" data from features with noise (Doumpos and Salappa, 2005).

There are three types of FS algorithms: Filter, wrapper and embedded. The former is independent of the classifier being usually done before the learning phase. Wrapper algorithms iteratively select subset of features and assess the learning models performance to determine

how useful that set of features are whereas embedded algorithms select automatically features during the model construction (Doumpos and Salappa, 2005; Rodriguez-Lujan et al., 2010). Figure 2.9 succinctly depicts the three approaches.

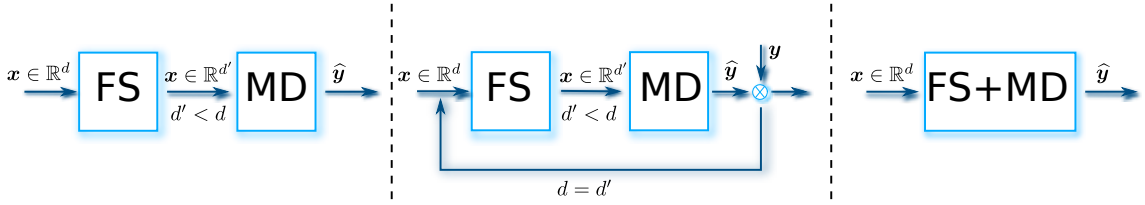


Figure 2.9: Three different standard approaches for feature selection: (left) depicts the *filter* feature selection approach done before the model design (MD); (center) the *wrapper* is consisted on an iterative approach where features are removed step by step until a desirable performance of the model is achieved; and (right) *embedded* method is designed jointly with the learning model algorithm.

FS on ordinal data is a relatively young topic. In Liu and Setiono (1997) a χ^2 statistic method is used to discretize numeric features as a way to select features. Even though the method proposed by Liu and Setiono (1997) was identified as being limited to a first-order feature-class correlation (i.e., are linearly correlated), such should not be seen as a drawback. Once highly complex learning machines could easily cope with the data complexity and infer a linear relation with the features and classes, or more precisely, perform overfitting on data (Seth and Príncipe, 2010; Sousa et al., 2011). Nevertheless, Last et al. (2001) proposed an information-theoretic method for FS by performing a discretization over the features in order to minimize classes entropy. Even though ordinal data can contain only discrete features fitting well to this technique, there are datasets with continuous features (see for instance Cardoso and Cardoso (2007)). In such scenarios, applying a discretization technique can lead to loss of accuracy in the model design. Despite being mentioned the capability to handle ordinal data, no experiment has been conducted, neither their methods were designed for this type of problems. Through a completely different approach, Xia et al. (2007) present a recursive approach to extract features where it learns consecutively new rules from instances represented by the new features.

Other techniques in the ordinal context have been referred to Baccianella et al. (2010a;b). Using only the filter approach for FS, authors used several measures to identify feature relevance through the minimization of the instances variance over all classes, similarity, information gain and negative correlation according to the class label, specifically developed for ordinal problems. Finaly, Sousa et al. (2011) explored a concept introduced by Rodriguez-Lujan et al. (2010) where they tackle the FS problem in one-step process through quadratic programming as represented in Equation (2.9). The quadratic term (Q in Equation (2.9)) would capture the redundancy whereas the linear term (F in Equation (2.9)) would capture the relevance.

$$\min_{\boldsymbol{x}} \left\{ \frac{1}{2}(1-\alpha)\boldsymbol{x}^t Q \boldsymbol{x} - \alpha F^t \boldsymbol{x} \right\} \tag{2.9}$$

Here α is the trade-off between relevance and redundancy which can be empirically defined. In order to capture the ordinal relation on data in this setting, authors chosen the Minimum Spanning Trees (MST) as the linear term (F) to assess the increase of complexity when a subset of features is removed. However, one of the issues identified in this approach concerns to the fact that authors did not take advantage of the ordinal information that could be explicitly included on data (quadratic term).

2.3.2 Performance Measures

After considering the advantages and disadvantages, goals achieved and open issues of the techniques presented in previous sections, the discussion of how to measure the performance of such techniques is still feeble.

Usually, a learning process consists in two main phases: A cross-validation phase and an estimation of the model performance ($\mathcal{F}$ represented in Equation (2.8)) on a real-world scenario (also known as the testing phase). In both situations, one has to analyze the performance of a model given certain parametrization and its behavior in non controllable environment, respectively. Herein, the question that one obviously poses is: How much did the model err? Or, how much the prediction differs from the real outcome? Given certain assumptions of models design, it is clear, as we will shortly show, that the metric chosen for this task is crucial.

It is interesting to see that in contrast to the plethora of existing methods concerning multicriteria learning, only recently we witnessed some concerns to this issue (Frasch et al., 2011; Lee and Liu, 2002), disregarding advances performance made on the broader field of machine learning (Lavesson and Davidsson, 2007). Knowing that "no free lunch" theorems state that there is not an algorithm that can be superior on all problems in regard to classification accuracy (Wolpert, 2001), the assessment of an appropriate learning method given a specific problem is desirable (Lavesson and Davidsson, 2007).

For classification problems, Misclassification Error Rate (MER) is currently one of the most used measures. Its widely use make it a *de facto* standard when comparing different learning algorithms by just counting the misclassifications occurred. In other problems domains, it is usual to penalize the misclassifications by weighting them by the magnitude of the error to avoid uneven results. When such happens, Mean Absolute Error (MAE) and Mean Square Error (MSE) measures are usually the most appropriate choices. Summing, the performance of a classifier can be assessed in a dataset $\mathcal{O}$ through

$$\frac{1}{N}\sum_{\boldsymbol{x}\in\mathcal{O}}|g(\mathcal{C}_{\boldsymbol{x}})-g(\widehat{\mathcal{C}}_{\boldsymbol{x}})|; \qquad \frac{1}{N}\sum_{\boldsymbol{x}\in\mathcal{O}}\left(g(\mathcal{C}_{\boldsymbol{x}})-g(\widehat{\mathcal{C}}_{\boldsymbol{x}})\right)^2,$$

respectively, where $g(.)$ corresponds to the number assigned to a class, $N = \text{card}(\mathcal{O})$, and $\mathcal{C}_{\boldsymbol{x}}$ and $\widehat{\mathcal{C}}_{\boldsymbol{x}}$ are the true and estimated classes. However, this assignment is arbitrary and the numbers chosen to represent the existing classes will evidently influence the performance measurement given by MAE or MSE. A clear improvement on these measures would be to define them directly from the Confusion Matrix (CM) (a table with the true class in rows and the predicted class in columns, with each entry $n_{r,c}$ representing the number of instances from the $r-$th class predicted as being from $c-$th class):

$$MAE = \frac{1}{N}\sum_{r=1}^{K}\sum_{c=1}^{K} n_{r,c}|r-c|; \qquad MSE = \frac{1}{N}\sum_{r=1}^{K}\sum_{c=1}^{K} n_{r,c}(r-c)^2$$

where K is the number of classes. We will always assume that the ordering of the columns and rows of the CM is the same as the ordering of the classes. This procedure makes MAE and MSE independent of the numbers or labels chosen to represent the classes. To a certain degree, these two measures are better than MER because they take values which increase with the absolute differences between 'true' and 'predicted' class numbers and so the misclassifications are not taken as equally costly.

In order to avoid the influence of the numbers chosen to represent the classes on the performance assessment, it has been argued that one should only look at the order relation between 'true' and 'predicted' class numbers. The use of Spearman's rank correlation coefficient, R_s, and specially Kendall's coefficient, τ_b, is a step in that direction (Kendall, 1938; Spearman, 1904). For instance, in order to compute R_s, we start by defining two rank

vectors of length N which are associated with the variables $g(\mathcal{C})$ and $g(\widehat{\mathcal{C}})$. There will be many examples in the dataset with common values for those variables; for these cases average ranks are used. If $\boldsymbol{p}$ and $\boldsymbol{q}$ represent the two rank vectors, then $R_s = \frac{\sum(p_i-\bar{p})(q_i-\bar{q})}{\sqrt{\sum(p_i-\bar{p})^2\sum(q_i-\bar{q})^2}}$. As we can see, Spearman's coefficient is still dependent on the values chosen for the ranks representing the classes and so it is not completely appropriate to measure the performance of ordinal data classifiers. More importantly, R_s looses information about the absolute value of the classes. Kendall's coefficient τ_b has been advocated as a better measure for ordinal variables because it is independent of the values used to represent classes (Kendall, 1938). Its robustness is achieved by working directly on the set of pairs corresponding to different observations. To define τ_b, start with the two N-point vectors, associated with the true and predicted classes, $\mathcal{C}_{\mathbf{x}}$ and $\widehat{\mathcal{C}}_{\mathbf{x}}$, and consider all $\frac{1}{2}N(N-1)$ pairs of data points. Before proceeding, some definitions are required (Press et al., 2002).

Definition 2.10 (Concordant Pair). *We call a pair* (i, j) concordant, c, *if the relative ordering of the true classes* $\mathcal{C}_{\boldsymbol{x}_i}$ *and* $\mathcal{C}_{\boldsymbol{x}_j}$ *is the same as the relative ordering of the predicted classes* $\widehat{\mathcal{C}}_{\boldsymbol{x}_i}$ *and* $\widehat{\mathcal{C}}_{\boldsymbol{x}_j}$.

Definition 2.11 (Discordant Pair). *We call a pair* discordant, d, *if the relative ordering of the true classes is opposite from the relative ordering of the predicted classes.*

Definition 2.12 (Pair Ties). *If there is a tie in either the true or predicted classes, then we do not call the pair either concordant or discordant. However, different concepts applies to different types of ties.*

extra true pair: *If the tie is in the true classes, we will call the pair an* extra true pair, e_t.

extra predicted pair: *If the tie is in the predicted class, we will call the pair an* extra predicted pair, e_p.

ignore pair: *If the tie is both on the true and the predicted classes, we ignore the pair.*

The τ_b coefficient can be computed as

$$\tau_b = \frac{c-d}{\sqrt{c+d+e_t}\sqrt{c+d+e_p}},$$

where c refers to concordant pairs and d for discordant pairs. The τ_b coefficient attains its highest value, 1, when both sequences agree completely, and -1 when the two sequences totally disagree. However, the source of robustness is probably the source of its main limitation: By working only with the relative order of elements, it looses information about the absolute prediction for a given observation.

Other attempts have considered the analysis of the learner behavior on a Receiver Operating Characteristic (ROC) curve or Area Under Curve (AUC). Despite empirical evidences of AUC providing more desirable properties when compared to accuracy (Bradley, 1997) only recently this topic was not only re-proposed but also new evidences of its advantages were shown (Huang and Ling, 2005). In this work, AUC is demonstrated as an objective measure for selecting the best learning model, but, and most important, refers to the need of developing better measures for learner design and performance assessment (Huang and Ling, 2005). In this line of research, in (Waegeman et al., 2006) it is compared different ROC measurements. However, and despite the assumptions made, ROC derived measures that assess a ranking for different performance do not quantify the performance achieved by a learner (Waegeman et al., 2008). Such analysis, although with different purposes, has been conducted by Ben-David (2007) using Cohen's kappa statistic.

On the other way, the discussion was revamped by (Baccianella et al., 2009) through an analysis of different derivations of MSE and MAE metrics for ordinal problems. This work is key since it debates two main issues incurred on the performance measurement of learners for this type of classification problems: Imbalanced classes and classes with equal penalization costs. In order to avoid the former problematic, a derivation from MAE is presented by averaging the deviations per class.

$$MAE^M = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{g(\widehat{\mathcal{C}}_i)}|g(\mathcal{C}_i) - g(\widehat{\mathcal{C}}_i)|$$

In the same line, the coefficient r_{int} was recently introduced, taking into account the expected high number of ties in the values to be compared (da Costa et al., 2008). In fact, the variables $\mathcal{C}$ and $\widehat{\mathcal{C}}$ are two special ordinal variables. Because there are usually very few classes compared to the number of observations, these variables will take many tied values (most of them, in fact). Nevertheless, r_{int} is sufficiently general and, if there are no tied values, it can still be applied as it is. Like τ_b, r_{int} assumes that the only thing that matters is the order relation between such values, which is the same as the order relation between the classes. This coefficient takes values in $[-1, 1]$, in contrary to MAE (and MSE) which are upper-unbounded. The latter can be identified as a limitation. Another observation is that it is fair to compare MAE results in two different applications with a different number of observations, N, since MAE is properly normalized by N. However, if the applications involve a different number of classes, K, it is not clear how to compare the performance obtained in the two settings.

Other techniques can also go through data generators methodologies where one can control the statistical properties herein aiding in the learners benchmark (Frasch et al., 2011). More importantly, techniques capable to manipulate Bayes error rate can foster new lines of research where fair learners comparison (Ben-David, 2007) and the development of new ones take place.

As one knows, the usage of such metrics in the design of classifiers can be done on two distinct situations. A first use is 'externally' to the classifier, using the metric to select the best parametrization of the classifier (usually when performing a cross-validation procedure). A second possibility is to embed the new metric in the classifier design, adapting the internal objective function of the classifier, replacing loss functions based on standard measures by a loss function based on the proposed measure. For instance, the standard loss function of a neural network based on the square of the error or on cross-entropy could be replaced by an error evaluated by an appropriate metric (Huang and Ling, 2005). (Lee and Liu, 2002) accomplished such for the design of ordinal trees, but since then few works have addressed this subject in the ordinal context.

It is interesting that only recently we saw a significant growth of the awareness of this topic importance. Even though some works have already tackled this issue, all lack on concretely assessing the performance of a given ordinal learning model. Until now, new metrics have been designed and compared against MAE followed by some reasoning. The problem resides how close a metric is in expressing accuracy. Different prosaically strategies can pass through the definition of prior costs for each class (Oliveira et al., 2010) or, when using a given set of different metrics, a meta-metric to assess the performance of metrics should be in place as suggested by (Cardoso and Sousa, 2011).

2.4 Discussion

Multicriteria (MC) has been studied for over more than five decades where recent years presented interesting developments. Aside novel methodologies, a trend towards the generalization of this problem was identified where at the same time a new light was shed over this

topic thanks to a niche of applications. In this chapter a thorough review was conducted on two major disciplines: Operations Research (OR) and Artificial Intelligence (AI).

MCDA has a strong connection with OR community. Fuzzy Set theory research community was one that rapidly proposed new models towards these problems. Their capability to handle uncertainty can be identified as an asset in these models. Even though in other research fields MC is giving its first steps, a new trend is appearing as a number of different studies are taking place. On the other hand, evolutionary approaches are still on the very beginning regarding ordinal problems. It also has been claimed that some approaches do not cope well with many criteria or do not capture correctly every rationale taken by the decision maker.

In the AI domain, it was described that albeit the myriad of techniques, some do not totally incorporate or effectively use the additional information of order in the classifier construction. Others have a higher complexity to be useful in real problems or require specific optimization algorithms during the classifier construction. Also, it was identified that is still common the usage of regression approaches to solve the ordinal data problem. Notwithstanding, some improvements have been achieved. Simplifications have been introduced through the usage of a standard binary classification techniques and fundamental principles towards the ordinal data problem. Such theories have proved to be valuable in the design of simpler classifiers and when not possible, in the design of posterior rules to impose ordinality. Another question that has recently been tackled concerns about finding good metrics for measuring learners performance. We reviewed many adaptations of standard metrics and new ones that optimize different criteria of the learner behavior.

In the end, and in spite of much of what has been achieved, a fair comparison between methods of both fields is still lacking. It was also clear that MC is very rich in terms of nomenclature. Having identified what it has been achieved and current open issues, it is expected that this study leads to future technical developments.

Part II

Learning Models for Ordinal Data

Chapter 3

Measuring Performance of Ordinal Classifiers*

In supervised classification problems with ordered classes, it is common to assess the performance of the classifier using measures more appropriate for nominal classes, regression problems or preference learning (Baccianella et al., 2009; Gaudette and Japkowicz, 2009). Baccianella et al. (2009) address the adaptation of existing measures (MAE) to unbalanced data, while Gaudette and Japkowicz (2009) compare existing measures concluding that MAE and MSE are the best performance metrics. Other strategies encompass the use of rank order measures (Lee and Liu, 2002; Vanbelle and Albert, 2009) or the adaptation of the ROC curve (Waegeman et al., 2006). However, the application of these measures faces difficulties in the context of ordinal classification, as we will show next.

In this Chapter our main goal is to propose a new metric specifically adapted to ordinal data problems, problems endowed with a natural order among classes. We argue that standard metrics do not adequately take into account all the information in the assessment process. We also claim that an error coefficient appropriate for ordinal data should capture how much the result diverges from the ideal prediction and how "inconsistent" the classifier is in regard to the relative order of the classes. This "inconsistency" results from discordant results in the relative order given by the classifier and the true relative class order.

3.1 A Preliminary Comparison of the Merits of Existing Metrics

A major difficulty in the design of a new classification performance coefficient lies in the difficulty in demonstrating that the coefficient captures adequately the performance of the classification algorithms. In a first test to check the adequacy of the coefficients discussed in the previous section, we created synthetic classification results and compared the values given by the coefficients with the expected measured performance. The performance of any classification algorithm is conveniently summarized in the CM and any of the coefficients presented in the previous section can be computed directly from it. Suppose that four classifiers A, B, C and D produce the following the CMs ($K = 4$, $N = 13$) in a certain task:

$$CM(A) = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \quad CM(B) = \begin{bmatrix} 0 & 4 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

*Some portions of this Chapter appeared in Cardoso and Sousa (2011).

$$CM(C) = \begin{bmatrix} 0 & 0 & 4 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \quad CM(D) = \begin{bmatrix} 0 & 4 & 0 & 0 \\ 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

One would expect that a valid measure of performance would output for classifier A a perfect performance, for classifier B a inferior performance and for classifier C a performance below B's performance.

Table 3.1 presents the results for the different coefficients. In order to aid our analysis, we have used in first place the metric MER and MAE due to is common use. Afterwards, we have selected two other metrics more appropriate for the ordinal data problem: Kendall's coefficient, τ_b, and Spearman coefficient, R_s. Although Spearman's coefficient does not consider all errors equally costly, it still depends on the values used to represent the classes. Kendall's coefficient does not; it measures the agreement in respect to the *relative* ordering of all possible pairs of data. Finally, we also used r_{int} which was proposed specifically for the ordinal data problem. Note that MER and MAE are indices of dissimilarity while R_s, τ_b and r_{int} are indices of similarity. It is important to remark right now a limitation of MAE (and MSE). Start by noticing that the range of possible values for MAE is an upper-unbounded interval. Nevertheless, it is fair to compare MAE results in two different applications with a different number of observations, N, since MAE is properly normalized by N. However, if the applications involve a different number of classes, K, it is not clear how to compare the performance obtained in the two settings.

Table 3.1: Results for the preliminary comparison, with $\beta_1 = \frac{0.25}{N(K-1)}$ and $\beta_2 = \frac{0.75}{N(K-1)}$. Coefficients $OC^1_{\beta_1}$ and $OC^1_{\beta_2}$ will be introduced later in the text.

classifier	MER	MAE	R_s	τ_b	r_{int}	$OC^1_{\beta_1}$	$OC^1_{\beta_2}$
A	0.0	0.0	1.0	1.0	1.0	0.0	0.0
B	0.77	0.77	1.0	1.0	1.0	0.50	0.63
C	0.77	1.08	0.79	0.75	0.80	0.61	0.78
D	0.77	0.77	0.24	0.11	0.53	0.65	0.72

Note that R_s, τ_b and r_{int} were unable to detect any performance difference between classifiers A and B; that results from the fact that they only measure relative values. We can also conclude that, in this context, $1 - R_s$, $1 - \tau_b$ and $1 - r_{int}$ do not constitute metrics since they do not satisfy the identity of indiscernible property ($d(x, y) = 0$ if and only if $x = y$). The MER coefficient was unable to differentiate classifiers B and C; note that, since classes are ordered, it is worst to predict points from class $\mathcal{C}_1$ to belong to class $\mathcal{C}_3$ rather than to predict them to be from class $\mathcal{C}_2$. The MAE coefficient (MSE would present the same behavior) was unable to differentiate classifiers B and D; note that classifier B was more consistent than classifier D in the sense that the relative order of the predicted classes coincide with the true order of the classes.

Finally, one can discuss the relative merit of C and D classifiers. If the ranking-based error is more relevant than the instance-based error then C should be preferred over D since the relative evaluation of C is consistent with the correct classification. When the instance-based error is prominent over the ranking error then one should prefer classifier D. We will return to this point later.

3.2 The Ordinal Classification Index

Nominal data classification analyzes each item in isolation and it is the closeness of the predicted assignment with respect to the exact one the most relevant criterion. Ranking, which is an aggregate evaluation task, is instead totally focused on respecting the ordering of items, not considering the actual values assigned to them. When applied to ordinal classification, a drawback of any pairwise criteria, such as Kendall's coefficient, is that it does not allow example dependent evaluation.

At the heart of the proposed measure is the incorporation of a ranking-based component to an instance-based evaluation of ordinal classification. Nevertheless, the new metric is still applicable to the evaluation of single points.

An appropriate error coefficient for ordinal data should capture how much the result diverges from the ideal prediction and how much 'inconsistent' the classifier is in regard to the relative order of the instances. We propose to define a metric directly in the CM, capturing these two sources of errors.

For this we adopt the following definition of *non-discordant pair of points*:

Definition 3.1 (Non-Discordant Pairs). *A pair of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is called* non-discordant *if the relative order of the true classes $\mathcal{C}_{\boldsymbol{x}_i}$ and $\mathcal{C}_{\boldsymbol{x}_j}$ is not opposite to the relative order of the predicted classes $\hat{\mathcal{C}}_{\boldsymbol{x}_i}$ and $\hat{\mathcal{C}}_{\boldsymbol{x}_j}$ (if there is a tie in either the true or predicted classes, or both, the pair is still* non-discordant*).*

In the CM the Definition 3.1 is translated into

$$\operatorname{sign}\left((r_{\boldsymbol{x}_i} - r_{\boldsymbol{x}_j}) \times (c_{\boldsymbol{x}_i} - c_{\boldsymbol{x}_j})\right) \geq 0, \tag{3.2}$$

where $r_{\boldsymbol{x}_i}$ and $c_{\boldsymbol{x}_i}$ are the row and column in the CM corresponding to example $\boldsymbol{x}_i$, respectively. Finally, define a path in the CM as a sequence of entries where two consecutive entries in the path are 8-adjacent neighbors. The benefit corresponding to a path is the sum of the values of the entries in the path. In fact, it is useful to consider a graph associated with the CM, where each entry of the matrix corresponds to a vertex and there is an edge connecting vertices corresponding to adjacent entries.

The coefficient to be proposed results from the observation that the performance yielded by the MER coefficient is the benefit of the path along the diagonal of the CM. The MER coefficient only counts the pairs in the main diagonal of the CM to measure the performance; any deviation from the main diagonal is strictly forbidden – see Figure 3.1a.

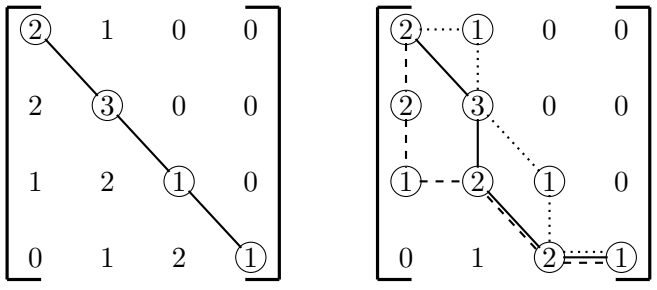


Figure 3.1: Consistent paths over the CM. Figure 3.1a illustrates the benefit of the MER coefficient as the sum of the entries in the main diagonal of the CM. The MER coefficient results as $\frac{N-\text{benefit}}{N}$. Figure 3.1b shows some examples of consistent paths; any pair of observation contributing to the entries in a consistent path are *non-discordant*. The benefit of a path is the sum of the entries in the path.

A more relaxed coefficient can be defined by allowing the pairs to deviate from the diagonal, while staying *non-discordant*. Therefore, we allow all pairs forming a consistent path from (1,1) to (K,K) – see Figure 3.1b. A path is said to be consistent if every pair of nodes in the path is non-discordant. It is trivial to verify that any monotonous path (a path where the row and column indices do not decrease when walking from (1,1) to (K,K)) is consistent. The consistency of the classifier is therefore taken into account by valuing only the *non-discordant* subsets of entries. Still, it is not enough to select the consistent path with the maximum benefit.

One should also penalize the deviation of the path from the main diagonal. We propose then to find the consistent path from (1,1) to (K,K) that maximizes the sum of the entries in the path and minimizes a measure of the deviation from the main diagonal. We propose the Ordinal Classification Index (OCI), OC_β, to take the shape

$$OC_\beta = \min \left\{ (1 - \frac{1}{N}\text{benefit(path)}) + \beta(\text{penalty(path)}) \right\}$$

where the minimization is performed over the set of all consistent paths from (1,1) to (K,K) and $\beta \geq 0$. Tentative solutions for the penalty of the path include the excess on the length of the path over the minimum possible length (penalty(path) = length(path)$-K$), the maximum distance of the path to the main diagonal or the area between the path and the main diagonal. However, it is intuitive that these terms do not meet the required properties. In Figure 3.2a and Figure 3.2b we present two paths that would experience the same penalization under a measure based on the length of the path, the maximum distance to the main diagonal or the area of the path; however, it should be consensual that the CM in Figure 3.2a represents a better performance than the CM in Figure 3.2b.

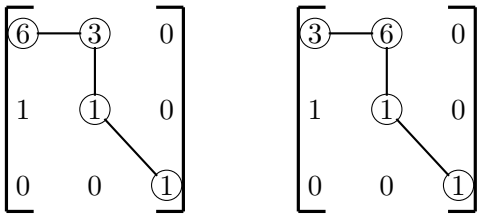


Figure 3.2: The two paths 3.2a and 3.2b would have the same penalization using the length, the maximum distance to the main diagonal or the area to select the cost; however, path a) should be preferred over path b).

A penalization term suggested by the expressions of MAE and MSE is based on penalizing each vertex of the path by its 'distance' to the main diagonal, obtaining

$$OC_\beta^{'\gamma} = \min \left\{ (1 - \frac{1}{N} \sum_{(r,c)\in \text{path}} n_{r,c}) + \beta \sum_{(r,c)\in \text{path}} n_{r,c}|r-c|^\gamma \right\}, \tag{3.3}$$

where $\gamma > 1$. It is clear that $OC_\beta^{'\gamma}$ is always non-negative, as the two terms in Equation (3.3) are both non-negative; $OC_\beta^{'\gamma}$ is also not superior to 1 as $OC_\beta^{'\gamma}$ is always not superior to the cost over the main diagonal, where the path penalty is zero. It is also easy to conclude that if $\beta \geq 1$ then $OC_\beta^{'\gamma}$ will equal the MER: since any deviation from the main diagonal will incur in a cost not inferior to 1, the optimal path is always over the main diagonal.

Nevertheless, this setting is still unsatisfactory; incorporating in the objective function only terms measuring the quality of the path does not capture differences in performance

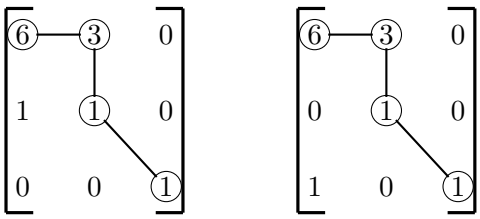


Figure 3.3: The performance represented by CM in Figure 3.3a should be better than the performance represented by CM in Figure 3.3b.

due to the leftover entries – see Figure 3.3a and Figure 3.3b. One needs also to penalize the 'dispersion' of the values from the main diagonal.

A first tentative solution is to add an additional term $\beta_2 \left(\sum_{\forall(r,c)} n_{r,c}|r-c|^\gamma\right)^{1/\gamma}$ to the objective function penalizing such dispersion of the data. This approach suffers from the disadvantages of adding a further parameter whose value needs to be selected and of changing the range of possible values for OC_β^γ from $[0,1]$ to an upper-unbounded interval.

Therefore, we propose to change the definition (3.3) by normalizing the benefit of the path not by N but by $N+M$, where $M = \left(\sum_{\forall(r,c)} n_{r,c}|r-c|^\gamma\right)^{1/\gamma}$ is a measure of the dispersion of the data in the CM:

$$OC_\beta^\gamma = \min\left\{1 - \frac{\sum_{(r,c)\in \text{path}} n_{r,c}}{N + \left(\sum_{\forall(r,c)} n_{r,c}|r-c|^\gamma\right)^{1/\gamma}} + \beta \sum_{(r,c)\in \text{path}} n_{r,c}|r-c|^\gamma\right\} \tag{3.4}$$

Note that M can be interpreted as the Minkowski distance between the two vectors used to build the CM. The parameter β controls the tradeoff between the relevance of the ranking-based component and the instance based evaluation. Small values for β will favor ranking over 'absolute' classification; high values for β will do the opposite. In Table 3.1 we present the results for two different values of β. The only difference is the relative merit of classifiers C and D, in accordance with the preceding discussion.

3.2.1 The Ordinal Classification Index – General Formulation

Thus far, the consistency was valued by working only with non-discordant pairs of points. The feasible paths were constrained under the set of consistent paths. A standard procedure in optimization is to replace a constraint by a penalty term in the goal function. Assume now we extend the set of feasible paths to the set of paths starting in (1,1) and ending in (K,K). Note also that there is always on of such paths going through all the entries in the CM. One can generalize the framework over this set of paths, penalizing now not only the deviation of the path from the main diagonal, but also the inconsistency of the path. One can therefore add an additional penalizing term to the definition of the index, capturing this undesirable attribute. An intuitive penalization term is the number of discordant pairs of vertices in the

path, N_{disc_pos} (see (3.2)):

$$OC^{\gamma}_{\beta_1;\beta_2} = \min \left\{ 1 - \frac{\sum_{(r,c)\in \text{path}} n_{r,c}}{N + \left(\sum_{\forall (r,c)} n_{r,c}|r-c|^{\gamma}\right)^{1/\gamma}} + \right.$$
$$\left. \beta_1 \sum_{(r,c)\in \text{path}} n_{r,c}|r-c|^{\gamma} + \beta_2 N_{disc\_pos} \right\} \quad (3.5)$$

Now the minimization is performed over all possible paths from (1,1) to (K,K). Since N_{disc_pos} is a non-negative integer, setting $\beta_2 \geq 1$ will revert to the initial OC^{γ}_{β}. Note that $OC^{1}_{0;0} = \frac{MAE}{1+MAE}$ is just a normalized version of MAE.

Nevertheless, we will not explore further this generalized index and all the following discussion will be based in the formulation (3.4).

3.2.2 Single Sample-Size

A key distinction between measures such as MAE (MER or MSE) and Kendall's τ_b (or Spearman's rank correlation coefficient R_s or r_{int}) is that the latter cannot be applied to assess the performance in a single object. By working with pairs of observations, τ_b is not applicable to a single observation.

Although OC^{γ}_{β} integrates a ranking-based component, it is straightforwardly applied to a single example evaluation. Assume that the true and predicted classes of the observation correspond to the r-th row and the c-th column in the CM, respectively. Setting in Equation (3.4) $N = 1$, $n_{r,c} = 1$, $n_{r',c'} = 0$ if $r', c' \neq r, c$, then OC^{γ}_{β} equals

$$OC^{\gamma}_{\beta} = \min \left(1; \quad 1 - \frac{1}{1+|r-c|} + \beta|r-c| \right),$$

which increases monotonously from 0 to 1 when the distance of the example to the main diagonal increases from 0 to infinity. Figure 3.4 illustrates this evolution for different values of β. Note that, in this setting, for $\beta = 0.5$, OC already equals the MER.

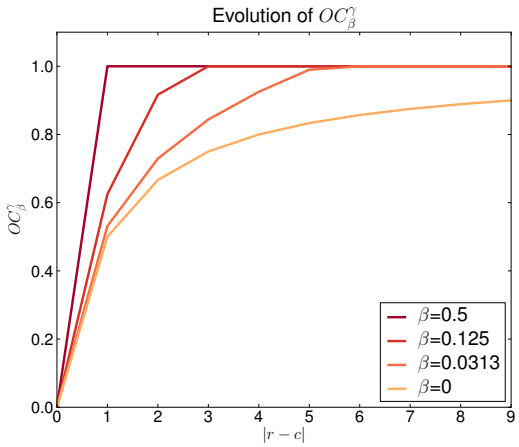


Figure 3.4: Evolution of OC^{γ}_{β} for a single example evaluation.

3.2.3 Properties of OC^{γ}_{β}

Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ be vectors used to construct CMs. It is easily observed from the definition that for $\beta > 0$, $\gamma > 1$ $OC^{\gamma}_{\beta}(\boldsymbol{a}, \boldsymbol{b}) = 0$ if and only if $\boldsymbol{a} = \boldsymbol{b}$.

Since the cost given by (3.4) of any consistent path is always non negative, OC_β^γ is always non negative; since the cost of the path through the main diagonal is always not superior to 1, $OC_\beta^\gamma \leq 1$.

It should be clear that the transposition of the CM does not change the value of OC_β^γ and therefore OC_β^γ is symmetric with respect to the role of the vectors involved in the construction of the CM: $OC_\beta^\gamma(\boldsymbol{a}, \boldsymbol{b}) = OC_\beta^\gamma(\boldsymbol{b}, \boldsymbol{a})$.

These conditions express intuitive notions about the expected properties for a classification performance index. It is also possible to establish that, for sufficiently high values of β, the triangular inequality is also satisfied, meaning that for certain values of β OC_β^γ is a metric. See Appendix A for further details.

3.2.4 Computational Remarks

Noting from Equation (3.4) that there is a cost $w_{r,c}$ corresponding to each vertex (entry in the matrix) of the graph given by

$$w_{r,c} = -\frac{n_{r,c}}{N + \left(\sum_{\forall(r,c)} n_{r,c}|r-c|^\gamma\right)^{1/\gamma}} + \beta n_{r,c}|r-c|^\gamma$$

the optimal consistent path can be found using dynamic programming. The first step is to traverse the matrix from the first entry to the last entry and compute the cumulative minimum weight W for all possible connected consistent paths for each entry (r, c):

$$W_{r,c} = w_{r,c} + \min\{W_{r-1,c-1}, W_{r-1,c}, W_{r,c-1}\}$$

With the adequate initialization ($W_{1,1} = 1 + w_{1,1}$) and the adequate attention for the entries in the first row and column. At the end of this process, the value $W_{K,K}$ will equal OC_β^γ. The computational complexity of this process is $O(K^2)$.

For typical values of N and K, the overall complexity will be dominated by the cost of constructing the confusion matrix (N). This is also the complexity of MAE and MSE. Note also that the complexity of τ_b and r_{int} is not inferior to the complexity of OC.

3.3 Experimental Study

In this section we evaluate the behavior of the different coefficients in some additional cases, where it is possible to define a reasonable reference behavior. Typically, in the Minkowski distance, γ is rarely used for values other than 1, 2, and infinity. Since the overall conclusions do not differ for different γ values, we only present the experimental study for $\gamma = 1$. Simultaneously, the β values tested in this study are a percentage of the maximum possible value for the penalization term, $N(K-1)^\gamma$. Since the choice for β is likely to be application dependent, balancing the tradeoff between the ranking and absolute classification, we present the results for two values of β, in the low and high range of the interval: $\beta_1 = \frac{0.25}{N(K-1)^\gamma}$ and $\beta_1 = \frac{0.75}{N(K-1)^\gamma}$.

Tridiagonal matrices

Consider CMs that are tridiagonal, taking the form

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 1 & 0 & \cdots & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & \cdots & 0 & 0 & 1 & 1 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Figure 3.5 plots the values of the coefficients for different number of classes. As the figure suggests and is analytically possible to conclude, r_{int}, R_s and τ_b all converge to 1 (perfect performance) as $K \to \infty$. In opposition, MER, MAE converge to 2/3 and OC^1_β converge to 0.6. Our subjective evaluation of the performance of a classification result corresponding to

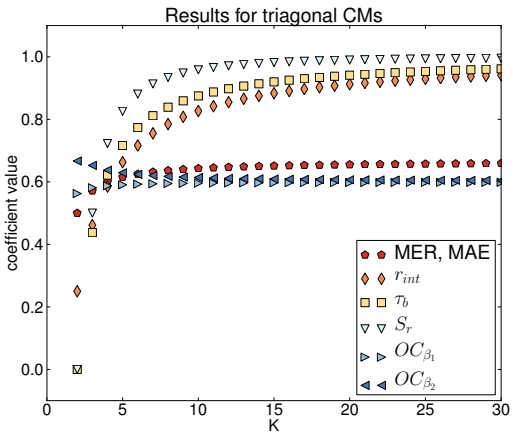


Figure 3.5: Results for tridiagonal CMs, with $\beta_1 = \frac{0.25}{N(K-1)}$ and $\beta_2 = \frac{0.75}{N(K-1)}$.

a tridiagonal matrix would hardly correspond to the perfect performance. The r_{int}, R_s and τ_b coefficients seem therefore to present an unintuitive behavior. It is also interesting to discuss if the performance should improve with the increase of K. Subjectively, one may argue that with the increase of K errors to the sub- and super-diagonals of the CM become less significant and the performance should improve. Under this assumption, $OC^1_{\frac{0.75}{N(K-1)}}$ presents the desired behavior.

Dispersed examples

To select the following examples, we randomly generated pairs of CMs and analyzed those where the relative performance as measure by OC_β did not agree with some of the other coefficients. Then, we tried to subjectively criticize the results.

A first pair of CMs is

$$CM_1 = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 0 \\ 2 & 1 & 2 \end{bmatrix} \quad CM_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

The values for the coefficients we have been considering are provided in Table 3.2. All

Table 3.2: Results for CM_1 and CM_2, with $\beta_1 = \frac{0.25}{N(K-1)}$ and $\beta_2 = \frac{0.75}{N(K-1)}$.

CM	MER	MAE	R_s	τ_b	r_{int}	$OC^1_{\beta_1}$	$OC^1_{\beta_2}$
CM_1	0.50	0.80	0.20	0.19	0.39	0.63	0.69
CM_2	0.40	0.60	0.10	0.11	0.45	0.53	0.58

coefficients, except R_s and τ_b, seem to be in agreement with the expected conclusion that the performance corresponding to CM_2 is better than the performance corresponding to CM_1.

Consider now the pair of CMs

$$CM_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 3 & 2 & 0 \end{bmatrix} \quad CM_4 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

The values for the coefficients we have been considering are provided in Table 3.3. Now all

Table 3.3: Results for CM_3 and CM_4, with $\beta_1 = \frac{0.25}{N(K-1)}$ and $\beta_2 = \frac{0.75}{N(K-1)}$.

CM	MER	MAE	R_s	τ_b	r_{int}	$OC^1_{\beta_1}$	$OC^1_{\beta_2}$
CM_3	0.86	1.43	-0.26	-0.254	0.34	0.79	0.93
CM_4	0.57	0.85	-0.25	-0.250	0.08	0.71	0.75

coefficients, with the exception of r_{int}, seem to be in agreement with the expected conclusion that the performance corresponding to CM_4 is better than the performance corresponding to CM_3.

In a third example, consider the following CMs

$$CM_5 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad CM_6 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and the values in Table 3.4. This time MER and MAE were unable to capture the degrada-

Table 3.4: Results for CM_5 and CM_6, with $\beta_1 = \frac{0.25}{N(K-1)}$ and $\beta_2 = \frac{0.75}{N(K-1)}$.

CM	MER	MAE	R_s	τ_b	r_{int}	$OC^1_{\beta_1}$	$OC^1_{\beta_2}$
CM_5	0.86	1.00	0.89	0.84	0.81	0.58	0.75
CM_6	0.71	1.00	-0.29	-0.26	0.06	0.74	0.79

tion of performance from CM_5 to CM_6. Note that CM_6 corresponds to an almost random classifier.

Evaluation of real classifiers Following Herbrich et al. (1999), we generated a synthetic dataset composed by 400 example points $\mathbf{x} = [x_1\ x_2]^t$ in the unit square $[0,1] \times [0,1] \subset \mathbb{R}^2$ according to a uniform distribution. This dataset is referred to the `synthetic` as presented in Chapter 2 and depicted in Figure 2.2a.

We compared the performance of three classifiers: the recently proposed data replication method (Cardoso and da Costa, 2007), instantiated both in Support Vector Machines (ordinal Support Vector Machine (oSVM)) and Neural Networks (ordinal Neural Networks (oNN)) and the method by Frank and Hall (2001). For completeness, we will briefly describe these learning techniques.

The data replication method for ordinal data can be framed under the Single Binary Classifier (SBC), an approach for solving multiclass problems via binary classification relying on a single, standard binary classifier. SBC reductions can be obtained by embedding the original problem in a higher-dimensional space consisting of the original features, as well as one or more other features determined by fixed vectors, designated here as *extension features*. This embedding is implemented by replicating the training set points so that a copy of the original point is concatenated with each of the extension features' vectors. The binary labels of the replicated points are set to maintain a particular structure in the extended space. This construction results in an instance of an artificial binary problem, which is fed to a binary learning algorithm that outputs a single binary classifier. To classify a new point, the point is replicated and extended similarly and the resulting replicas are fed to the binary classifier, which generates a number of signals, one for each replica. This method can be instantiated in

two important machine learning algorithms: support vector machines and neural networks. For more details, the reader should consult (Cardoso and da Costa, 2007).

Using the aforementioned techniques, the dataset was split in 40% for training ($\mathcal{D}$) and 60% for testing ($\mathcal{D}^*$). Algorithm 1 illustrates the experimental procedure. The splitting of the data was repeated fifty times in order to obtain more stable results for performance estimation. In line 7 and line 12 of Algorithm 1 one can use any of the metrics discussed in this Chapter in order to obtain the best parametrization of the model or estimate the final performance.

Algorithm 1: Experimental procedure to design the models. This procedure was repeated fifty times in order to obtain more stable results for performance estimation.

Data: $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ the training dataset and $D^* = \{\mathcal{X}^*, \mathcal{Y}^*\}$ the testing set.
Result: $\mathcal{M}$, trained model, accuracy accuracy result for $\mathcal{D}^*$ and respective CM.

1 Best_Accuracy $\leftarrow 0$;
2 Partition training data $\mathcal{D}$ in five equal subsets so that $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\} = \{(\mathcal{X}^1, \mathcal{Y}^1) \cup \ldots \cup (\mathcal{X}^5, \mathcal{Y}^5)\}$
3 **forall** *parametrization values* p **do**
4 **forall** $fold = 1$ **to** 5 **do**
5 $\mathcal{M} \leftarrow$ `Train_Model`$(\mathcal{X}^i, \mathcal{Y}^i, p)$ where $i = \{1, \ldots, 5\} \backslash fold$;
6 $\mathcal{Y}_1 \leftarrow$ `Test_Model`$(\mathcal{M}, \mathcal{X}^{fold})$;
7 accuracy$^{fold} \leftarrow$ assess performance according a given measure, m, $(\mathcal{Y}_1, \mathcal{Y}^{fold})$;
8 $\overline{\text{accuracy}} \leftarrow 1/5 \sum_{i=1}^{5}$ accuracy^i;
9 **if** $\overline{\text{accuracy}} >$ Best_Accuracy **then**
10 Best_Accuracy $\leftarrow \overline{\text{accuracy}}$;
11 Best_Parameterization $\leftarrow p$;
12 $\mathcal{M} \leftarrow$ `Train_Model`$(\mathcal{X}, \mathcal{Y},$ Best_Parameterization$)$;
13 $(\mathcal{Y}_1, CM) \leftarrow$ `Test_Model`$(\mathcal{M}, \mathcal{X}^*)$;
14 accuracy $\leftarrow$ assess performance according a given measure, m, $(\mathcal{Y}_1, \mathcal{Y}^*)$;

In the results of Table 3.5, CM_{10} represents the results for oSVM, CM_{11} the result for oNN and CM_{12} the performance for Frank&Hall. The CMs are as follows:

$$CM_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 7 & 0 & 0 \\ 0 & 2 & 94 & 2 & 0 \\ 0 & 0 & 11 & 39 & 0 \\ 0 & 0 & 0 & 5 & 30 \end{bmatrix} \quad CM_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 45 & 12 & 0 \\ 0 & 0 & 2 & 87 & 9 \\ 0 & 0 & 0 & 6 & 44 \\ 0 & 0 & 0 & 0 & 35 \end{bmatrix}$$

$$CM_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 7 & 0 & 0 \\ 0 & 2 & 94 & 2 & 0 \\ 0 & 0 & 21 & 29 & 0 \\ 0 & 0 & 0 & 29 & 6 \end{bmatrix}$$

A subjective analysis of the CMs places CM_{10} as the best result and CM_{11} in the bottom. Although all indices capture this relative performance, R_s, τ_b and r_{int} almost do not differentiate CM_{11} from CM_{12}. The OCI, on the other hand, portrays a significant difference in performance, in spite of also incorporating a ranking term.

Table 3.5: Results for CM_{10}, CM_{11} and CM_{12}, with $\beta_1 = \frac{0.25}{N(K-1)}$ and $\beta_2 = \frac{0.75}{N(K-1)}$.

CM	MER	MAE	R_s	τ_b	r_{int}	$OC^1_{\beta_1}$	$OC^1_{\beta_2}$
CM_{10}	0.11	0.11	0.93	0.91	0.91	0.12	0.13
CM_{11}	0.82	0.91	0.89	0.85	0.84	0.55	0.66
CM_{12}	0.25	0.25	0.90	0.86	0.86	0.23	0.26

Experiments with real datasets To further evaluate the impact of using OCI, we performed the following experiments with sets of real ordinal data, testing our method on the `SWD`, `LEV`, `ESL` and `Balance` datasets.

To assess the merit of OCI in an ordinal data classification setting, we trained three different classifiers on the five mentioned datasets:

- A conventional multiclass classifier, based on the one-against-one rationale. The baseline binary classifier was the binary SVM, as deployed in libSVM (Chang and Lin, 2001).

- The multiclass classifier adapted for ordinal data based on the proposal by Frank&Hall, as described previously. The baseline binary classifier was again the binary SVM, as deployed in libSVM. Previous works have shown the advantage of this method over conventional approaches (Frank and Hall, 2001; Herbrich et al., 1999).

- The data replication method, instantiated in SVM (oSVM), as also described before. Previous works have shown the advantage of this method over both conventional approaches and the Frank and Hall (2001) method (Cardoso and Cardoso, 2007; da Costa et al., 2008; 2010).

Once again the experimental study followed the setting illustrated in Algorithm 1. The datasets were split in 40% for training and 60% for testing; the optimization of the parameters using cross-validation over the training set was based on the OCI metric; the final assessment of the performance of the models in the test set was done again using OCI. A linear kernel was used in all learning schemes. The results are presented in Table 3.6.

Dataset	oSVM	Frank&Hall	Conventional
`SWD`	0.49 (0.02)	0.47 (0.01)	0.49 (0.02)
`LEV`	0.44 (0.02)	0.46 (0.02)	0.47 (0.02)
`ESL`	0.36 (0.00)	0.36 (0.01)	0.36 (0.01)
`Balance`	0.13 (0.01)	0.13 (0.01)	0.14 (0.02)

Table 3.6: Performance average (std. dev.) results for the five datasets using the OCI measure.

A first main assertion is that OCI correctly captures the superiority of both algorithms specific to ordinal data over the conventional method. The learning and the assessment with OCI are in accordance with the expected relative performance. The relative merit of oSVM and Frank&Hall method is not that strong, with a potentially slightly advantage of oSVM, both in average and in variance. It is also important to notice that oSVM produces simpler models than Frank&Hall method, since all boundaries share the same direction (the boundaries) are parallel. Likewise, Frank&Hall method produce simpler and more robust classifiers than the one-against-one generic model implemented in libSVM.

3.4 Discussion

We have proposed the use of a metric defined directly on the CM to evaluate the performance in ordinal data classification. The metric chooses the non-discordant pairs of observations that minimize the cost of a global optimization procedure on the CM, minimizing deviation of the pairs to the main diagonal while maximizing the benefit. The adoption of this measure thus guarantees fair comparison among competing systems, and more correct optimization procedures for classifiers.

Arguing in favor of a new metric against current ones is a difficult task, almost requiring a meta-metric to assess the performance of metrics. To overcome this difficulty we started by trying to motivate the interest of the proposed metric with intuitive settings and completed with the application in real datasets.

Finally, OCI measure was developed in a time frame subsequent to the methods presented in the following chapters. Due to this reason, the usage of the OCI was limited to the writing time of this document and therefore it was not possible to apply it in the different studies described next.

Chapter 4

An All-at-Once Unimodal SVM Approach for Ordinal Classification*

In this Chapter it is introduced a new All-at-Once SVM methodology specifically devised for supervised classification on ordinal data. An extension of the unimodal paradigm proposed in da Costa et al. (2008) and da Costa and Cardoso (2005) is here presented for SVM. Basically, the paradigm assumes that a posteriori probabilities of the K classes should follow an unimodal distribution so that order relationship can be taken into account. One will present the solution to this mathematical optimization problem which takes two forms: a basic and a sophisticated architecture. Afterwards, it is delved a formulation of this paradigm by introducing the appropriate constraints in the usual All-at-Once soft margin SVM optimization functions, both in its primal and dual forms. The remainder of this Chapter is concerned with the performance assessment of this approach on synthetic and real datasets.

4.1 Unimodal Paradigm

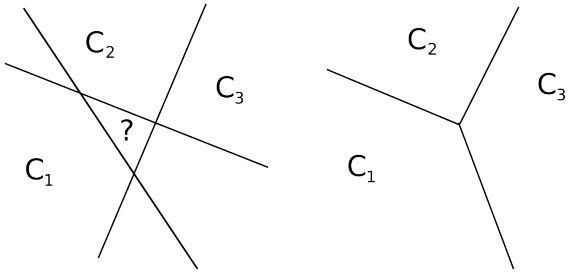
This Section recovers the idea of the unimodal paradigm presented in da Costa et al. (2008) and da Costa and Cardoso (2005). In the presence of a supervised multiclassification problem where the classes are ordered, like for instance the four classes (Cardoso and Cardoso, 2007), $Excellent \succ Good \succ Fair \succ Poor$, if for a particular instance the class with highest a posteriori probability is *Fair*, then its neighboring classes, *Good* and *Poor*, should have the second and third highest probabilities. This is the unimodal paradigm which states that the probabilities output by a prediction method should increase monotonically, until reaching a maximum value, and then decrease monotonically. In simple words, it does not make sense that the most likely class is *Fair* and that the second most likely is *Excellent*; it should be one of the classes closest to *Fair*. This unimodal paradigm has already been introduced in the context of NN in da Costa et al. (2008) and da Costa and Cardoso (2005) and this work follows as a extension of it in another context, namely all-at-once SVM.

4.2 All-at-Once Methods

The all-at-once methods were proposed to the scientific community to overcome some vicissitudes present on the standard procedures like the pairwise, one-against-one, one-against-all schemes, DDAG, among others (Abe, 2005). One of the problems presented on stan-

*The work presented in this Chapter follows the line of research of da Costa et al. (2008). Moreover, some portions of this Chapter appeared in da Costa et al. (2010).

dard heuristics for supervised multiclass classification problems are the unclassifiable regions. These classifiers have the feature of not being capable of classifying a point which is within



(a) Example of an unclassifiable region.

(b) All-at-Once decision discriminators.

Figure 4.1: Different Decision Discriminators.

a particular decision region—see Figure 4.1a—since each decision function gives a different value for that point. All-at-once schemes solve this issue by determining all the decision functions simultaneously, and therefore do not generate these ambiguity regions.

4.2.1 Standard Approaches

The standard approaches follow closely the formulation proposed in Crammer and Singer (2002). However, it should be stated that this work did not focus on the study of the algorithm complexity that led Crammer and Singer (2002) to propose an iterative method. The methods implemented in this work are therefore a straightforward implementation of the mathematical formulation.

As referred previously, the technique proposed by Crammer and Singer (2002) tries to determine all the decision functions simultaneously. More specifically,

$$\boldsymbol{w}_i^T g(\boldsymbol{x}) + b_i > \boldsymbol{w}_j^T g(\boldsymbol{x}) + b_j, \quad j \neq i, \quad i = 1, \ldots, K \tag{4.1}$$

where $g(\boldsymbol{x})$ is the mapping function, $\boldsymbol{w}_i$ the weight vector for the i^{th} class and b_i its bias term. There are two strategies to attain all the decision planes which we will describe in some detail in the following Sections. These are the basic and sophisticated architectures, as presented in Abe (2005).

Basic and Sophisticated Architectures

All-at-once techniques accomplish the capability to determine simultaneously K discriminant functions through the definition of one single optimization function. That is attained by incorporating K conditions which will serve to separate each class.

In the basic approach the objective function to be minimized is

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) = \frac{1}{2} \sum_{j=1}^{K} \|\boldsymbol{w}\|_j^2 + C \sum_{i=1}^{N} \sum_{j=1}^{K} \xi_{i,j}, \tag{4.2}$$

which uses $N \times K$ slack variables and, for each point $(\boldsymbol{x}_i, y_i)$ of the data set, is subject to the constraints

$$(\boldsymbol{w}_{y_i} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{y_i} - b_j \geq 1 - \xi_{i,j}, \quad \forall j \neq y_i, j = 1, \ldots, K, \quad i = 1, \ldots, N \tag{4.3}$$

An alternative to this approach consists in using only N slack variables. This follows the suggestion of Crammer and Singer (2002) which replaces the slack variables ξ_{ij} by $\xi_i =$

$\max_j \xi_{ij}$. The objective function becomes therefore,

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) = \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}\|_j^2 + C\sum_{i=1}^{N}\xi_i \tag{4.4}$$

subject to the constraints,

$$(\boldsymbol{w}_{y_i} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{y_i} - b_j \geq 1 - \xi_i, \quad \forall j \neq y_i, j = 1, \ldots, K, \quad i = 1, \ldots, N \tag{4.5}$$

As it is known, this last problem is easier to solve in the dual Lagrangian formalism. Focusing for the moment on the basic architecture, the optimization function becomes,

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \\ \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{N}\sum_{j=1}^{K}\xi_{i,j} - \sum_{i=1}^{N}\sum_{j=1}^{K}\beta_{i,j}\xi_{i,j} - \\ \sum_{i=1}^{N}\sum_{j=1}^{K}\alpha_{ij}((\boldsymbol{w}_{y_i} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{y_i} - b_j - 1 + \xi_{i,j})\end{aligned}$$

After some calculus, one obtains the following dual problem,

$$\begin{aligned}\max \mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\sum_{j=1, j\neq y_i}^{K}\alpha_{ij} - \frac{1}{2}\sum_{i,k=1}^{N}\sum_{j=1}^{K} z_{ij} z_{kj} H(\boldsymbol{x}_i, \boldsymbol{x}_k) \\ s.t \begin{cases} \sum_{i=1}^{N} z_{ij} = 0 \quad j = 1, \ldots, K \\ 0 \leq \alpha_{ij} \leq C, \quad i = 1, \ldots, N \quad j \neq y_i, j = 1, \ldots, K \end{cases}\end{aligned} \tag{4.6}$$

where $H(\boldsymbol{x}_i, \boldsymbol{x}_k)$ is the kernel function and

$$z_{ij} = \begin{cases} \sum_{k=1}^{K}\alpha_{ik}, & j \neq y_i \\ -\alpha_{ij}, & \text{otherwise} \end{cases} \tag{4.7}$$

The decision functions are given by

$$D_j(\boldsymbol{x}) = \sum_{i=1}^{N} z_{ij} H(\boldsymbol{x}_i, \boldsymbol{x}) + b_j, \quad j = 1, \ldots, K \tag{4.8}$$

and a new instance $\boldsymbol{x}$ is classified into the class $\arg\max_{j=1,\ldots,K} D_j(\boldsymbol{x})$. See Appendix B for further details.

4.2.2 Unimodal Approaches

In the previous Sections it was recovered the all-at-once SVM definition. However, its applicability to ordinal classification is not really appropriate (da Costa et al., 2008), since the order between the classes is not taken into account. The development of ordinal classifiers can lead to more interpretable results and a better generalization capability.

In a problem with K ordered classes, $\mathcal{C}_1 \prec \ldots \prec \mathcal{C}_K$, if the maximum *a posteriori* probability is attained at $\mathcal{P}(\mathcal{C}_i|\boldsymbol{x})$, the predicted class is $\mathcal{C}_i$. Then, the unimodal paradigm states that the probabilities should monotonically decrease through $\mathcal{P}(\mathcal{C}_{i+1}|\boldsymbol{x}) \geq \ldots \geq \mathcal{P}(\mathcal{C}_K|\boldsymbol{x})$ and $\mathcal{P}(\mathcal{C}_{i-1}|\boldsymbol{x}) \geq \ldots \geq \mathcal{P}(\mathcal{C}_1|\boldsymbol{x})$. This property motivated us to extend the all-at-once methods to the unimodal paradigm (da Costa et al., 2008).

In the following sections a natural derivation to ordinal classification will be developed inspired by the standard methods presented in the previous section.

Basic Architecture

The basic architecture comes naturally by reformulating the decision functions defined in equation (4.1) to ordinal problem towards the property mentioned in the Section 4.1. Therefore, the unimodal paradigm for class i is,

$$\begin{aligned} \boldsymbol{w}_{j+1}^T g(\boldsymbol{x}) + b_{j+1} \geq \boldsymbol{w}_j^T g(\boldsymbol{x}) + b_j, \quad j = 1, \ldots, i-1 \\ \boldsymbol{w}_j^T g(\boldsymbol{x}) + b_j \geq \boldsymbol{w}_{j+1}^T g(\boldsymbol{x}) + b_{j+1}, \quad j = i, i+1, \ldots, K-1 \end{aligned} \tag{4.9}$$

Consequently, the L_1 soft margin SVM can be obtained by minimizing

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) = \frac{1}{2} \sum_{j=1}^{K} \|\boldsymbol{w}\|_j^2 + C \sum_{i=1}^{N} \sum_{j=1}^{K-1} \xi_{i,j} \tag{4.10}$$

constrained to

$$\begin{aligned} (\boldsymbol{w}_{j+1} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{j+1} - b_j \geq 1 - \xi_{i,j}, \\ \forall j = 1, \ldots, y_i - 1 \\ (\boldsymbol{w}_j - \boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i) + b_j - b_{j+1} \geq 1 - \xi_{i,j}, \\ \forall j = y_i, \ldots, K-1 \end{aligned} \tag{4.11}$$

To solve this optimization problem, it was used the Lagrange formalism by introducing the non-negative Lagrange multipliers $\alpha_{i,j}$ and $\beta_{i,j}$ and the quantity to be minimized becomes,

$$\begin{aligned} &\mathcal{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \\ &\frac{1}{2} \sum_{j=1}^{K} \|\boldsymbol{w}_j\|^2 + C \sum_{i=1}^{N} \sum_{j=1}^{K-1} \xi_{i,j} - \sum_{i=1}^{N} \sum_{j=1}^{K-1} \beta_{i,j} \xi_{i,j} \\ &- \sum_{i=1}^{N} \sum_{j=1}^{y_i - 1} \alpha_{ij} ((\boldsymbol{w}_{j+1} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{j+1} - b_j - 1 + \xi_{i,j}) \\ &- \sum_{i=1}^{N} \sum_{j=y_i}^{K-1} \alpha_{ij} ((\boldsymbol{w}_j - \boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i) + b_j - b_{j+1} - 1 + \xi_{i,j}) \end{aligned}$$

and after some calculus one obtains the following dual problem:

$$\begin{aligned} \max \mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \sum_{j=1}^{K} \alpha_{ij} - \frac{1}{2} \sum_{i,k=1}^{N} \sum_{j=1}^{K} z_{ij} z_{kj} H(\boldsymbol{x}_i, \boldsymbol{x}_k) \\ s.t \begin{cases} \sum_{i=1}^{N} z_{ij} = 0 \quad j = 1, \ldots, K-1 \\ 0 \leq \alpha_{ij} \leq C, \quad i = 1, \ldots, N \quad j = 1, \ldots, K-1 \end{cases} \end{aligned} \tag{4.12}$$

where

$$\begin{aligned} z_{ij} = \alpha_{ij-1} I(j \geq 2) I(j \leq y_i) - \alpha_{ij} I(j \leq y_i - 1) \\ + \alpha_{ij} I(j \geq y_i) I(j \leq K-1) - \alpha_{ij-1} I(j \geq y_i + 1) \end{aligned} \tag{4.13}$$

and $H(\boldsymbol{x}_i, \boldsymbol{x}_k) = g(\boldsymbol{x}_i)^T . g(\boldsymbol{x}_k)$ is the kernel function. The decision functions are given by

$$D_j(\boldsymbol{x}) = \sum_{i=1}^{N} z_{ij} H(\boldsymbol{x}_i, \boldsymbol{x}) + b_j, \quad j = 1, \ldots, K \tag{4.14}$$

and a new instance $\boldsymbol{x}$ is classified into the class $\arg\max_{j=1,\ldots,K} D_j(\boldsymbol{x})$.

Sophisticated Architecture

Following Crammer and Singer (2002) suggestion, one replaces slack variables ξ_{ij} by $\xi_i = \max_j \xi_{ij}$. This produces significant differences in the initial formulation. Therefore, the optimization function becomes

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) = \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{N}\xi_i \tag{4.15}$$

restricted to

$$\begin{aligned}(\boldsymbol{w}_{j+1} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{j+1} - b_j \geq 1 - \xi_i, \quad \forall j = 1, \ldots, y_i - 1 \\ (\boldsymbol{w}_j - \boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i) + b_j - b_{j+1} \geq 1 - \xi_i, \quad \forall j = y_i, \ldots, K - 1\end{aligned} \tag{4.16}$$

The decision functions are given by

$$\begin{aligned}\max \mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\sum_{j=1}^{K-1}\alpha_{ij} - \frac{1}{2}\sum_{i,k=1}^{N}\sum_{j=1}^{K} z_{ij} z_{kj} H(\boldsymbol{x}_i, \boldsymbol{x}_k) \\ s.t \begin{cases} \sum_{i=1}^{N} z_{ij} = 0 \quad j = 1, \ldots, K-1 \\ 0 \leq \sum_{j=1}^{K-1}\alpha_{ij} \leq C, \quad i = 1, \ldots, N \end{cases}\end{aligned} \tag{4.17}$$

And the decision functions are given in the same manner as in Equation (4.14).

4.3 Experimental Study

In order to assess the performance of the approach here proposed, several experiments were performed. Firstly, a synthetic dataset was generated where the optimal discriminator was known (in this experiment it is only needed to find the best parameters values for the objective and kernel function). Afterwards, the method was evaluated in four real datasets.

On the synthetic dataset, randomly example points $\boldsymbol{x} = (x_1, x_2)^t$ in the unit square $[0, 1] \times [0, 1] \in \mathbb{R}^2$ were generated according to the uniform distribution. This dataset is referred to the `synthetic` as presented in Chapter 2 and depicted in Figure 2.2a.

All the algorithms where put under the same conditions, so that the results could be discussed fairly. The data was divided randomly and distributed through all algorithms. Classes were also equally divided on train (80 instances), validation and test sets to assure that each class was evenly represented. A 5-fold cross validation was performed. In order to assess the variability of the algorithms the experiments were repeated 100 times.

A straightforward implementation of the formulations presented in Section 4.2 were carried out and so it did not focus at present with performance issues. A grid search over C = $2^{-3}, \ldots, 2^{10}$ and $\gamma = 2^{-3}, \ldots, 2^3$ was performed and four measures were used to assess the performance of the models. C is a penalty factor for each point misclassified and γ controls the fitting of the kernel to the data.

The MER, although not very appropriate to these problems with ordered classes (because it considers all errors equally costly) was used due to its popularity. In the experiments a Radial Basis Function (RBF) kernel was used and also polynomial kernels with degrees 2 and 3.

Table 4.1 and Table 4.2 present the best overall results for the four schemes. Note that the postfix I or II refers to the basic and sophisticated architectures, respectively. First an

Method	standard I	standard II	unimodal I	unimodal II
MER	**0.35 (0.09)**	**0.35 (0.08)**	0.38 (0.09)	0.39 (0.11)
OCI	0.51 (0.37)	0.53 (0.38)	**0.30 (0.32)**	0.48 (0.40)

(a) mean (std. dev.) for `synthetic` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = (1 + \boldsymbol{x}'\boldsymbol{y})^2$.

Method	standard I	standard II	unimodal I	unimodal II
MER	0.49 (0.03)	0.49 (0.03)	**0.47 (0.03)**	0.51 (0.03)
OCI	0.40 (0.34)	0.40 (0.32)	**0.17 (0.29)**	0.41 (0.33)

(b) mean (std. dev.) for `SWD` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Method	standard I	standard II	unimodal I	unimodal II
MER	0.47 (0.04)	0.48 (0.04)	**0.46 (0.04)**	0.50 (0.04)
OCI	**0.52 (0.04)**	0.54 (0.03)	0.92 (0.00)	0.56 (0.03)

(c) mean (std. dev.) for `LEV` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Method	standard I	standard II	unimodal I	unimodal II
MER	0.49 (0.19)	**0.46 (0.12)**	0.55 (0.17)	0.50 (0.08)
OCI	**0.43 (0.02)**	0.50 (0.16)	1.00 (0.00)	0.53 (0.11)

(d) mean (std. dev.) for `ESL` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Method	standard I	standard II	unimodal I	unimodal II
MER	0.14 (0.02)	0.14 (0.02)	0.16 (0.03)	**0.13 (0.02)**
OCI	0.25 (0.04)	**0.23 (0.04)**	0.76 (0.00)	**0.23 (0.05)**

(e) mean (std. dev.) for `Balance` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Table 4.1: Results for MER and OCI measures.

Method	standard I	standard II	unimodal I	unimodal II
R_s	0.86 (0.06)	0.85 (0.07)	**0.87 (0.05)**	0.86 (0.05)
τ_b	0.80 (0.06)	0.78 (0.08)	**0.81 (0.05)**	0.79 (0.06)

(a) mean (std. dev.) for each method, `synthetic` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = (1 + \boldsymbol{x}'\boldsymbol{y})^2$.

Method	standard I	standard II	unimodal I	unimodal II
R_s	0.45 (0.16)	0.47 (0.06)	**0.51 (0.06)**	0.47 (0.07)
τ_b	0.41 (0.07)	0.41 (0.06)	**0.46 (0.05)**	0.42 (0.05)

(b) mean (std. dev.) for each method, `SWD` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Method	standard I	standard II	unimodal I	unimodal II
R_s	0.60 (0.05)	0.58 (0.05)	**0.63 (0.04)**	0.61 (0.05)
τ_b	0.53 (0.05)	0.52 (0.06)	**0.57 (0.04)**	0.54 (0.05)

(c) mean (std. dev.) for each method, `LEV` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Method	standard I	standard II	unimodal I	unimodal II
R_s	0.13 (0.91)	0.75 (0.33)	0.72 (0.42)	**0.81 (0.26)**
τ_b	0.02 (0.89)	0.69 (0.34)	0.64 (0.39)	**0.76 (0.20)**

(d) mean (std. dev.) for each method, `ESL` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Method	standard I	standard II	unimodal I	unimodal II
R_s	0.85 (0.03)	**0.86 (0.03)**	**0.86 (0.03)**	**0.86 (0.03)**
τ_b	0.82 (0.04)	**0.83 (0.03)**	0.82 (0.04)	0.83 (0.04)

(e) mean (std. dev.) for each method, `Balance` dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Table 4.2: Results for Spearman and Kendall's coefficients.

analysis with the MER measure was performed, due to its common use in classifiers evaluation and with OCI. Afterwards the other two measures of accuracy will be considered.

As can be seen in the results on Table 4.1, the benefits of this approach are not clear (values in bold correspond to the best results). The methods here proposed obtained the best results in 50% of the datasets, according to MER. One of the reasons is due to the measure used not being appropriate for this problem since it does not take into account the order of the classes. Therefore, the same experiments were conducted but by measuring the performance using the Spearman and Kendall's coefficients (see Table 4.2).

On the `synthetic` dataset once again the difference between the methods are very dim, although slightly better for the unimodal method. For the real datasets all unimodal schemes attained slightly better results than the corresponding standard all-at-once with exception on the `Balance`, where there are almost no differences.

Despite the results on the Balance dataset being very similar amongst all of the methods it is interesting to see this kind of performance with measures that take into account the order between the classes, whereas with MER, the unimodal approach gives slightly better results. This may be due to the class frequencies distribution on this dataset—see Figure 1.2d—because class #2 is slimly represented when compared with the other two.

4.4 Discussion

A new ordinal classification formulation for ordinal data was presented. Based on the unimodal paradigm proposed in da Costa et al. (2008) and da Costa and Cardoso (2005) it was extended onto the SVM context using the all-at-once strategies. This paradigm states that the probabilities output by a prediction method should increase monotonically until reaching a maximum value and then decrease monotonically. With such a strategy it is possible to enforce the ordinal relation amongst the classes.

We have also performed an extensive experimentation where these methods were tested against all-at-once standard techniques. The Unimodal all-at-once approach was tested on one synthetic and 4 real datasets where, in overall, the approach expressed superior results when comparing with standard all-at-once strategies. Finally, the classifier performance was assessed with four measures: MER, OCI, Spearman and Kendall's coefficients showing consistent results.

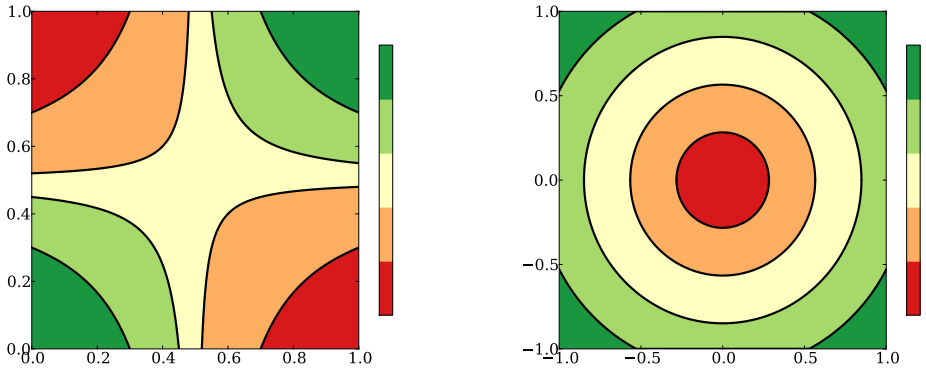
Chapter 5

Global Constraints for Ordinal Classification*

In this chapter we first present a novel rationale to capture and impose the order constraints in the design of a supervised classifier. The proposed formulation tries to objectify the imprecise notion of natural order. A second contribution lies on the instantiation of that underlying principle in the design of a new decision tree and a new nearest neighbor algorithms. Finally, we improve this formulation in order to diminish over-regularized and over-smoothed decision boundaries impact. Through the usage of ensemble learning techniques applied to decision trees, we can join the set of resulting trees into a single one. By applying a new formulation for the global constraints in order to impose the order, we can avoid over-regularized output decision regions.

5.1 Capturing the Order Constraints between Classes

Assume that examples in a classification problem come from one of K ordered classes, labeled from $\mathcal{C}_1$ to $\mathcal{C}_K$, corresponding to their natural order. Unlike the monotone learning problem, where both the input attributes and the class attribute are assumed to be ordered, the setting considered in this work does not assume that the inputs are ordered. Consider the two datasets in Figure 5.1 here repeated for better reading. Each point in Figure 2.2a was



(a) (b)

Figure 5.1: Two synthetic ordinal dataset where the monotonicity property at input data does not hold.

*Some portions of this Chapter appeared in Cardoso and Sousa (2010) and Sousa and Cardoso (2011).

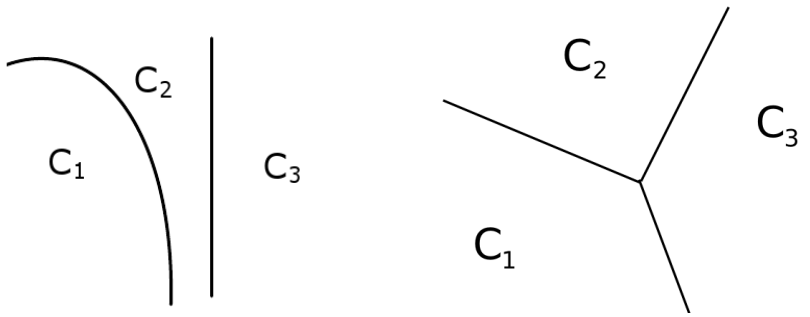
assigned a class y from the set $\{1, 2, 3, 4, 5\}$, according to

$$y = \min_{r \in \{1,2,3,4,5\}} \{r : b_{r-1} < 10(x_1 - 0.5)(x_2 - 0.5) + \varepsilon \ < b_r\}$$
$$(b_0, b_1, b_2, b_3, b_4, b_5) = (-\infty, -1, -0.1, 0.25, 1, +\infty) \tag{5.1}$$

where $\varepsilon \sim N(0; 0.125^2)$ simulates the possible existence of error in the assignment of the true class to $\boldsymbol{x}$. The data in Figure 5.1b is uniformly distributed in the unit-circle, with the class y being assigned according to the radius of the point: $y = \left\lceil \sqrt{x_1^2 + x_2^2} \right\rceil$ In neither of the datasets the monotonicity constraint is verified; however, we argue that these datasets are perfectly representatives of an ordinal setting, where the order is not captured directly in the input space, but in an implicit feature space. In fact the dataset in Figure 5.1a has been used to validate algorithms for ordinal data classification (Cardoso and da Costa, 2007; Herbrich et al., 1999).

How to capture then the order relation in the output? Let $f(\boldsymbol{x})$ be a decision rule that assigns each value of $\boldsymbol{x}$ to one of the available classes[2]. Such a rule will divide the input space into regions $\mathcal{R}_k$ called decision regions, such that all points in $\mathcal{R}_k$ are assigned to class $\mathcal{C}_k$. The boundaries between decision regions are called decision boundaries or decision surfaces. Note that each decision region need not be contiguous but could comprise any number of disjoint regions. Intuitively, for ordinal data, in a sufficiently small neighborhood of $\boldsymbol{x}$, $\mathcal{V}_\epsilon(\boldsymbol{x})$, the decision function should only take at most two consecutive values: $\max f(\boldsymbol{x}) - \min f(\boldsymbol{x}) \leq 1$. The motivation for this is that a small change in the input data should not lead to a 'big jump' in the output decision. Therefore, we say that a decision function is *consistent* with an ordinal data classification setting in a point $\boldsymbol{x}_0$ if $\exists \epsilon > 0 \ \forall \boldsymbol{x} \in \mathcal{V}_\epsilon(\boldsymbol{x}_0) \ \max f(\boldsymbol{x}) - \min f(\boldsymbol{x}) \leq 1$. A decision function is consistent in the whole input space if the above condition is verified for every point in the input space: $\forall \boldsymbol{x}_0 \exists \epsilon > 0 \ \forall \boldsymbol{x} \in \mathcal{V}_\epsilon(\boldsymbol{x}_0) \ \max f(\boldsymbol{x}) - \min f(\boldsymbol{x}) \leq 1$[3].

Decision functions consistent with the ordinal setting lead to the very pleasant result that a region $\mathcal{R}_i$ where one decides for $\mathcal{C}_i$ can only be adjacent to regions $\mathcal{R}_{i+1}$ and $\mathcal{R}_{i-1}$—see Figure 5.2.



(a) Regions consistent with the ordinal setting.

(b) Regions inconsistent with the ordinal setting.

Figure 5.2: Consequence of the consistency constraint in the arrangement of the decision regions.

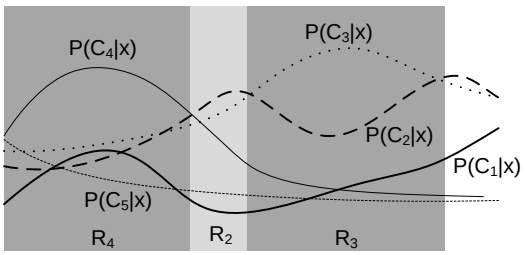
The rationale here introduced is a generalization of the formulation of parallel boundaries adopted in linear SVM for ordinal data (Shashua and Levin, 2003) and the non-intersecting

[2] A remark should be made. Since we are dealing with ordered classes, we shall consider that the output of the decision function is one of the K labels $\{\mathcal{C}_1, \cdots, \mathcal{C}_K\}$ or one number in $\{1, \cdots, K\}$ resulting from the bijective map $g : \{\mathcal{C}_i\}_{i=1}^K \to \{1, \cdots, K\}$ which assigns the number k to the class $\mathcal{C}_k$, i.e., $g(\mathcal{C}_k) = k$. The context should make it clear which of the two output formats is being considered.

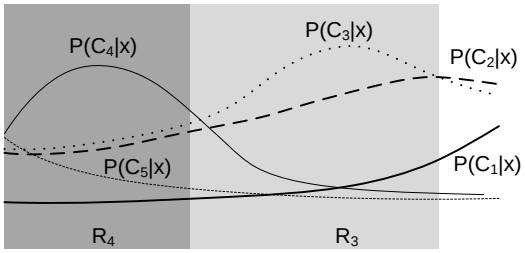
[3] This definition of consistency precludes decision functions such as $f(x) = 1, x < 0; f(x) = 2, x = 0; f(x) = 3, x > 0$, where the region corresponding to class 2 is a measure-zero set.

boundaries approach adopted in Cardoso and da Costa (2007). We also notice that the approach by Frank and Hall (2001) may lead to inconsistent solution under the adopted formulation since the design of independent classifiers will likely result in intersecting boundaries.

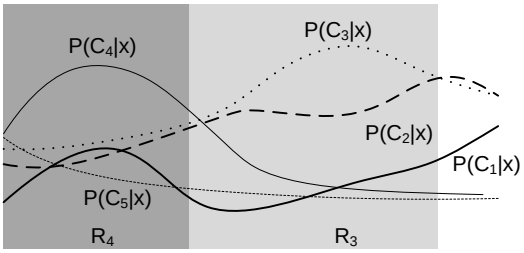
It is also interesting to establish a parallel with the probabilistic framework introduced previously by da Costa et al. (2008). The unimodal model assumes that for any given point $\boldsymbol{x}$ the posterior probabilities $\mathcal{P}(\mathcal{C}_k|\boldsymbol{x})$ follow a unimodal distribution. Given a point $\boldsymbol{x}$, if the highest a posteriori probability is, for instance, $\mathcal{P}(\mathcal{C}_k|\boldsymbol{x})$, then we should have, given that there is an order relation between the classes, $\mathcal{P}(\mathcal{C}_1|\boldsymbol{x}) < \cdots < \mathcal{P}(\mathcal{C}_{k-1}|\boldsymbol{x}) < \mathcal{P}(\mathcal{C}_k|\boldsymbol{x}) > \mathcal{P}(\mathcal{C}_{k+1}|\boldsymbol{x}) > \cdots > \mathcal{P}(\mathcal{C}_K|\boldsymbol{x})$: $\mathcal{C}_{k-1}$ and $\mathcal{C}_{k+1}$ are closer to $\mathcal{C}_k$ and therefore the second highest a posteriori probability should be attained in one of these classes, see Figure 5.3b. Had one used a classifier which does not take into account the order relation between the classes, the second highest a posteriori probability can be, for instance, $\mathcal{P}(\mathcal{C}_{k-2}|\boldsymbol{x})$, see Figure 5.3a.



(a) Illustrative posteriori class distribution for a conventional nominal data problem.



(b) Illustrative posteriori class distribution for the unimodal model for ordinal data.



(c) Illustrative posteriori class distribution sufficient to assure the consistency property for ordinal data.

Figure 5.3: Illustrative posteriori class distributions for different models.

While the unimodal model imposes an order relationship between any two consecutive class probabilities, such a strict condition is not required to observe the consistency property we introduce in this work. In fact, the consistency property will be observed if the following conditions, in-between the conventional formulation for nominal data and the unimodal model, are true:

$$\begin{aligned} \mathcal{P}(\mathcal{C}_k|\boldsymbol{x}) > \mathcal{P}(\mathcal{C}_{k-1}|\boldsymbol{x}) > \mathcal{P}(\mathcal{C}_i|\boldsymbol{x}), \quad \forall\ 1 < i < k-1 \\ \mathcal{P}(\mathcal{C}_k|\boldsymbol{x}) > \mathcal{P}(\mathcal{C}_{k+1}|\boldsymbol{x}) > \mathcal{P}(\mathcal{C}_i|\boldsymbol{x}), \quad \forall\ k+1 < i < K \end{aligned} \tag{5.2}$$

Intuitively, one just needs to impose that the second higher probability is the 'right' one. This is sufficient (although not necessary) to assure that, at the decision boundaries the decision rule will change for an adjacent class.

5.2 Imposing the Ordinal Constraints in a Decision Function

Consistency is a global property, i.e., it involves a relation between different decision regions of the space. A key challenge is how to use this information during the design process of

a learning algorithm. In this section we consider that a decision function has already been obtained by, possibly, standard methods and use the consistency property to relabel the decision regions.

It is convenient at this point to define some notation to describe the assignment of labels to different decision regions. Let $\mathcal{R}_n$, $n = 1, \cdots, N$, represent the contiguous decision regions created by some model[4]. For each region $\mathcal{R}_n$ we introduce a corresponding set of binary indicator variables $x_{n,k} \in \{0, 1\}$, where $k = 1, \cdots, K-1$ describing which of the K ordinal labels is assigned to region $\mathcal{R}_n$, so that if data points in $\mathcal{R}_n$ are assigned the label k then $x_{n,j} = 1$ for $j < k$, and $x_{n,j} = 0$ otherwise. So, for instance if we have a setting with 5 classes, $K = 5$, and to a particular region happens to be assigned the label 3, then $\boldsymbol{x}$ will be represented by $\boldsymbol{x} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^t$. Note that this is different from the often used 1-of-K coding scheme and we find it more convenient for the introduction of the constraints in what follows.

In ordinal data settings, the loss associated with a region $\mathcal{R}_n$ when deciding for class $\mathcal{C}_k$ is usually captured with the absolute error, the sum over all points lying in $\mathcal{R}_n$ of the absolute difference between the true class of the point and the predicted class for the region:

$$c_{n,k} = \sum_{i=1}^{K} |i - k| p_{n,i},$$

where $p_{n,i}$, $n = 1, \cdots, N$, $i = 1, \cdots, K$ represent the number of observations (from the data used in creating the region by some learning algorithm) from class k satisfying the conditions for region $\mathcal{R}_n$, (that is, lying inside $\mathcal{R}_n$). Nevertheless, the following model is generic for any costs $c_{n,k}$.

The optimal labeling of the regions can then be found by minimizing the following objective function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} c_{n,k}(x_{n,k-1} - x_{n,k}), \tag{5.3}$$

where the *constants* $x_{n,0} = 1$ and $x_{n,K} = 0$ have been introduced for notational convenience, with the constraints

$$x_{n,k+1} - x_{n,k} \leq 0, k = 1, \cdots, K-2, \quad n = 1, \cdots, N \tag{5.4}$$

and

$$x_{n,k} \in \{0, 1\}, k = 1, \cdots, K-1, \quad n = 1, \cdots, N \tag{5.5}$$

It is easily seen that Equation (5.3) can be rewritten as

$$J = \sum_{n=1}^{N} \left\{ c_{n,1} + \sum_{k=1}^{K-1} x_{n,k}(c_{n,k+1} - c_{n,k}) \right\}, \tag{5.6}$$

Without any constraints relating the labels of the regions, the optimization of the loss J over the whole space leads to the standard solution of predicting the median of the values in each region.

Now, we want to impose that adjacent regions have labels that differ at most by one. Therefore we are led to the optimization of the loss of the decision function constrained by

[4]Note the change of notation: so far we have used $\mathcal{R}_k$ to represent the decision region, contiguous or not, corresponding to class $\mathcal{C}_k$. From now on $\mathcal{R}_n$ just represents a continuous region of the space with all points inside that region being assigned the same class. Therefore, different regions $\mathcal{R}_n$ and $\mathcal{R}_m$ may be assigned the same class and the number of regions is likely greater than the number of classes.

the consistency of it. Consistency imposes that, for any pair of adjacent regions $\mathcal{R}_n$ and $\mathcal{R}_{n'}$, the following inequality must be verified:

$$\left| (1 + \sum_{k=1}^{K-1} x_{n,k}) - (1 + \sum_{k=1}^{K-1} x_{n',k}) \right| \leq 1 \tag{5.7}$$

Inequality (5.7) can be written as

$$\begin{aligned} \sum_{k=1}^{K-1} x_{n,k} - \sum_{k=1}^{K-1} x_{n',k} \leq 1 \\ \sum_{k=1}^{K-1} x_{n',k} - \sum_{k=1}^{K-1} x_{n,k} \leq 1 \end{aligned} \tag{5.8}$$

The optimization of (5.6), subject to constraints (5.4), (5.5) and (5.8) constitutes a linear binary integer programming problem.

Although the resulting constraint matrix is not totally unimodular (which would allow the relaxation of the linear binary integer programming problem to a much easier linear programming problem), we found experimentally that the actual shape and sparsity of the constraint matrix of typical problems favor the efficiency of the algorithm. Nevertheless, further research on the computational complexity of the method is required.

5.2.1 Algorithms for Solving the 0-1 Linear Model

In this section we focus on two algorithms for solving the 0-1 linear model. Although for small problems the 0-1 formulation can be used directly, this approach becomes prohibitive with the increase of the dimension of the data, the increase of the size of training set or with the increase of the number of classes.

Iterative Algorithm

The observation that decision regions for class $\mathcal{C}_k$ are more likely to be adjacent to regions labeled for $\mathcal{C}_j$ with $|j - k|$ small, suggests a block coordinate optimization procedure, where the consistency constraints are imposed iteratively to a different subset of regions.

Initializing the region labels to the conventional value obtained from the median label of the points assigned to the region, we propose to iteratively select a subset of regions with labels in the interval $\mathcal{C}_j, \cdots, \mathcal{C}_{j+W-1}$ and re-label those regions with the output of the optimization problem restricted to those regions. The simplest solution is to simply iterate j from 1 to $K - W + 1$. Note that if we select $W = K$ we would be solving the complete original problem; if we select $W = 2$ no constraint will be imposed and one stays in the solution without consistency constraints.

Note that the global consistency of the solution obtained at the end of the iterative process is not assured.

Approximation Algorithm based on LP Relaxation

A relaxation procedure starts by choosing and solving a relaxation problem for obtaining an approximated solution; then, it uses a rounding procedure to extract a feasible solution to the original 0-1 problem from the approximate solution. The relaxation step has an important role in the whole algorithm. For example, if the approximation solution is in fact feasible for the original problem, then it is exactly an optimal solution. On the other hand, when the approximation solution is not feasible regarding the original problem, we have to use a rounding procedure to extract a feasible solution.

The relaxed model for our 0-1 problem is obtained by replacing the constraint (5.5) by

$$x_{n,k} \in [0,1], k = 1, \cdots, K-1, \quad n = 1, \cdots, N \tag{5.9}$$

Solving now (5.6), subject to constraints (5.4), (5.9) and (5.8) finds the solution to our relaxed problem.

Noting now that (5.4), together with the monotonicity of the round function, assures that the rounded solution is a valid coding for the class — although not necessarily a feasible solution since the constraints (5.8) may not be observed —, that terminates the relaxation method. Again, the global consistency of the solution obtained at the end of the iterative process is not assured.

5.3 An Ordinal k-Nearest-Neighbor: the okNN Model

The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms. This algorithm belongs to a set of techniques called Instance Based Learning. It starts by extending the local region around a data point until the k^{th} nearest neighbor is found. For nominal data, an object is classified by a majority vote scheme, with the object being assigned to the class most common amongst its k-nearest neighbors; for ordinal data, the median is usually preferred.

In the simplest case, consider $k = 1$ and a given set of points S. Each training point $\boldsymbol{x}_i$ defines a Voronoi cell R_i, a convex polytope, consisting of all points closer to $\boldsymbol{x}_i$ than to any other training point $\boldsymbol{x}_j$. The label assigned to a given Voronoi cell R_i is the label of the corresponding training point $\boldsymbol{x}_i$.

The consistency constraints for ordinal data introduced before are also easily integrated in the 1-NN classifier. Now the regions involved in the optimization process are the Voronoi cells; the cost $c_{n,k}$ is simply $c_{n,k} = |k - i|$, where i is the class of the training point in the cell. The adjacency can be tested by testing the adjacency of the corresponding polytopes.

The extension to the k-NN can be accomplished in two ways. One option is to apply the consistency constraints directly on the generalized Voronoi cells corresponding to the k-NN as a post-processing, identically to what was just proposed for the 1-NN. Another option is to use the above procedure on 1-NN as a pre-processing before applying a standard k-NN. It is possible to show that, under some conditions, the resulting decision function is consistent.

Consider the neighborhood $V_k(\boldsymbol{x})$ containing the k nearest training points of the (test) point $\boldsymbol{x}$. Let m be the minimum and M the maximum of those k labels. Under the assumption that the training points have been relabeled by imposing the consistency constraints in the 1-NN classifier, the set of the k labels contains every label between m and M. Consider the Voronoi cells from 1-NN that intersect $V_k(\boldsymbol{x})$ and a graph with a vertex in each of the k training points and an edge for each pair of adjacent training points (for which the cells are adjacent). Then there is a path between any pair of vertices, and in particular between a point labeled with m and a point labeled with M. Since the Voronoi cells are consistent, the path must go throw each possible label between m and M. Now, adjacent regions in the k-NN differing at a single of the k points will then also differ at most by one in the median of the k points. When adjacent regions differ at more than 1 of the k point due to, for instance, coincident training points, the consistency is not assured.

5.4 An Ordinal Decision Tree

The root of the majority of the work on decision trees is in Breiman's work (Breiman et al., 1984) and Quinlan's ID3 algorithm (Quinlan, 1986) from statistical and machine learning perspectives. Decision trees are hierarchical decision systems in which conditions are sequentially tested until a class is accepted. To this end, the feature space is split into unique

regions, corresponding to the classes, in a sequential manner. Upon the arrival of a feature vector, the searching of the region to which the feature vector will be assigned is achieved via a sequence of decisions along a path of nodes of an appropriately constructed tree. The most popular schemes among decision trees are those that split the space into hyper-rectangles with sides parallel to the axes. The sequence of decisions is applied to individual features, and the questions to be answered are of the form "is feature $x_k \leq \alpha$?" where α is a threshold value. Such trees are known as ordinary binary classification trees (OBCTs).

An algorithm for the induction of a decision tree from a training dataset contains the following ingredients:

- a splitting rule: At each node, the set of candidate questions to be asked has to be decided. Each question corresponds to a specific binary split into two descendant nodes. A splitting criterion must be adopted according to which the best split from the set of candidate ones is chosen.

- a stopping rule: A stop-splitting rule is required that controls the growth of the tree and a node is declared as a terminal (leaf). The most commonly used approach is to grow the tree up to a large size first and then prune nodes according to a pruning criterion. A number of pruning criteria have been suggested. A commonly approach is to combine an estimate of the error probability with a complexity measuring term (e.g. number of terminal nodes) (Ripley, 1986).

- a labeling rule: a rule is required that assigns each leaf to a specific class.

5.4.1 Imposing the Ordinal Constraints in a Decision Tree: the oTree Model

If the consistency is measured for each possible split during tree construction, the order in which nodes are expanded becomes important. For example, a depth-first search strategy will generally lead to a different tree than a breadth-first search. Also, and perhaps more importantly, a non-consistent tree may become consistent after additional splits.

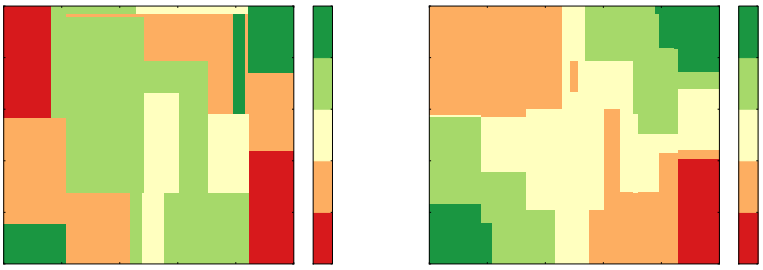
In view of these difficulties, in this work we consider imposing consistency only during the labeling assignment step. Future work will address other mechanisms. Consider an already constructed tree, using any standard technique such as C4.5 (Quinlan, 1993), perhaps already pruned according to a pre-specified strategy.

We can now apply the rationale developed in the previous section to the regions corresponding to each leaf of the tree. In this scenario, each region is a hyper-rectangle. In Figure 5.4 is depicted the decision regions obtained by growing a tree without pruning from 300 random observations generated according to Equation (5.1). In Figure 5.4b is visible the benefits of imposing the consistency constraints by relabeling the leaves. It is also interesting to interpret the consistency constraints as a regularization factor in the tree building process.

5.4.2 Avoiding Over-Regularized Decision Spaces

Even if this baseline framework has the potential to improve the performance of a model, that did not always happen in the experiments reports in (Cardoso and Sousa, 2010). We conjecture that the use of the consistency property only as a post-processing operation may lead to 'over-regularized' or over-smoothed decision functions, effectively hurting or attenuating the positive impact on the generalization performance of the model. This over-regularization could be especially true with small datasets, precisely when it is more needed.

One way to try overcoming this problem is to force an over-partition of the space prior to the relabeling for global consistency. One would expect that the global optimization would



(a) Without consistency constraints. (b) With consistency constraints.

Figure 5.4: Decision regions for a fully-grown tree.

then compensate this initial over-refinement. Resampling techniques (Zoubir and Iskander, 2007), noise induction (Wang and Principe, 1999), or other similar approaches could be used to induce this over-partition of the space. In here we explore the resampling approach on the context of ensemble learning.

Although the bootstrap technique is a general tool for assessing statistical accuracy, it can also be used to improve the accuracy of a prediction scheme. The basic idea is to randomly draw datasets with replacement from the training data, each sample the same size as the original training set. This is done B times ($B = 100$ say), producing B bootstrap datasets. Then we fit a DT to each of the bootstrap datasets. Typically bootstrap aggregation or bagging would then select the class with the most "votes" from the B DT. In here we will consider the option of working directly with the partition of the space corresponding to each DT—see Figure 5.5.

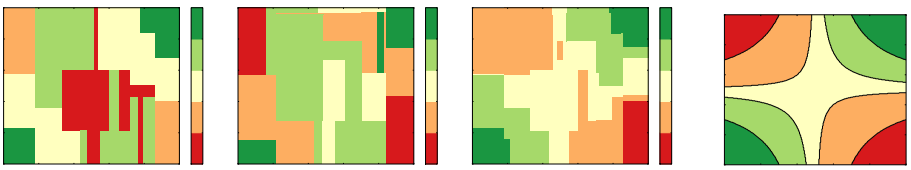


Figure 5.5: Example of individual models and their aggregation under an ensemble algorithm. First two figures: two distinct models; remaining figures: aggregated regions of the two models and optimal decision boundaries, respectively.

Instead of bagging directly the output of the B DT we propose to group first the B DT in groups of M DT and to compute the fusion (intersection) of the M corresponding space partitions, see Figure 5.6. Each merged partition will then be relabeled according to the consistency optimization procedure described earlier. Finally, we bag the relabeled models. Since we are dealing with ordinal data, we use the median of the B/M votes as the final decision. A natural question to ask is if the model induced by the bagging procedure is still consistent according to our previous definition. That this is indeed true is easily confirmed.

Theorem 1. *Aggregation of consistent decisions produces a consistent decision when using the 'median voting' as the fusion rule.*

Proof 1. *Consider* $\boldsymbol{x}$ *and the* $L = B/M$ *predictions* $y_1, \cdots, y_L$ *at* $\boldsymbol{x}$ *by the* L *models, which are by construction consistent. Consider* $\boldsymbol{x}+\boldsymbol{\delta}$ *in a small enough neighborhood of* $\boldsymbol{x}$ *so that the* L *predictions* $z_1, \cdots, z_L$ *at* $\boldsymbol{x}+\boldsymbol{\delta}$ *obey the consistency constraint, namely* $z_i \in \{y_i-1, y_i, y_i+1\}$. *The consistency of the 'median voting' scheme results from the simple observation that since*

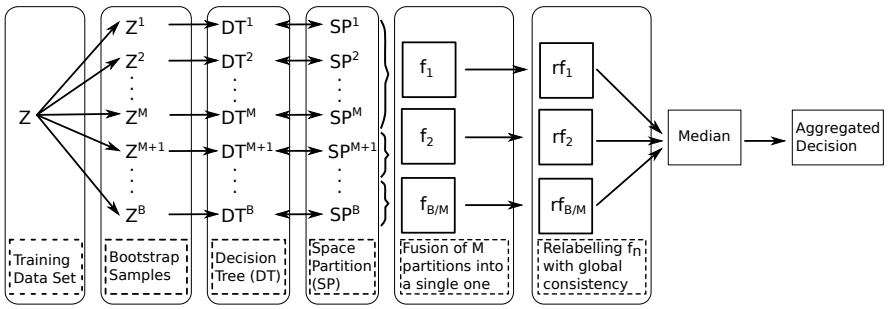


Figure 5.6: Schematic of the proposed aggregation process.

$y_i - 1 \leq z_i \leq y_i + 1$ *then*
$median(y_1, \cdots, y_L) - 1 = median(y_1 - 1, \cdots, y_L - 1) \leq median(z_1, \cdots, z_L) \leq median(y_1 + 1, \cdots, y_L + 1) = median(y_1, \cdots, y_L) + 1$

□

Global consistency with empty regions

The fusion mechanism is likely to produce empty regions, i.e., regions without instances from the training set. A direct consequence is that the optimization procedure provided early becomes ill-defined, in the sense that there are multiple optimal labellings. In fact, any relabeling of the empty regions that is still consistent does not change the value of the objective function, see Figure 5.7. We set additional constraints on the labels of the empty

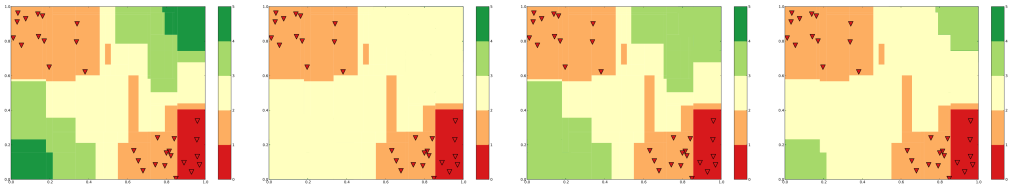


Figure 5.7: Different labeling with the same value for the optimization function (objective function in Equation (5.6) s.t. (5.4), (5.5) and (5.8)).

regions so that the optimization problem becomes again well defined. Intuitively, we argue that adjacent empty regions would share the same label. Instead of forcing hard constraints, we suggest to penalize in the objective function any deviation of this goal. The constraints given in Equation (5.8) are re-written for pairs of regions involving empty regions as in Equation (5.10):

$$\begin{aligned} \sum_{k=1}^{K-1} x_{n,k} - \sum_{k=1}^{K-1} x_{n',k} &\leq \delta_{(n,n')} \\ \sum_{k=1}^{K-1} x_{n',k} - \sum_{k=1}^{K-1} x_{n,k} &\leq \delta_{(n,n')} \end{aligned} \qquad \forall (n, n') \in \Delta \tag{5.10}$$

$$\delta_{(n,n')} \in \{0, 1\} \qquad \forall (n, n') \in \Delta \tag{5.11}$$

where Δ contains all empty adjacent regions. The objective function is also updated with a regularization factor as represented in Equation (5.12):

$$J = \sum_{n=1}^{N} \left\{ c_{n,1} + \sum_{k=1}^{K-1} x_{n,k}(c_{n,k+1} - c_{n,k}) \right\} + C \sum_{\forall (n,n') \in \Delta} \delta_{(n,n')} \tag{5.12}$$

where $C > 0$ controls the tradeoff between the smoothness over the labels of the empty regions, which we want to impose and the need to satisfy the consistency property. Since the new term in the objective function has the single purpose of, among the solutions satisfying the consistency property, favor the solutions with 'almost' constant labels in the empty regions, C should be 'sufficiently' small so that inconsistent solutions (but very smooth over the empty regions) are not preferred. However, in this formulation, pairs of adjacent regions where both are empty and pairs which have exactly one empty region are treated equally in terms of the relabeling cost. Take for instance the possible labellings in Table 5.1. Assume

Case 1:	$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_3$
Case 2:	$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_3$
Case 3:	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_3$

Table 5.1: Different possible labellings.

that the decision regions on the first and last columns are populated with some instances of the training set whereas the remaining decision regions are empty. The training observations in the first and last columns are such that the optimal decision is those regions is $\mathcal{C}1$ and $\mathcal{C}3$, respectively.

All three labeling are equivalent by the baseline optimization criterion (Cardoso and Sousa, 2010). However, the last two are preferred over the first one by the re-formulation in Equations (5.10),(5.11),(5.12), since both minimize the number of label transitions.

Intuitively, empty regions adjoin with non-empty regions should share the label of the non-empty region. The rationale is similar to the margin maximization of other learning schemes, putting the transition between labels further away from the data points. Therefore, pairs of empty regions should have a lower penalty than pairs which have exactly one empty region.

Letting Δ_1 be the set containing only pairs of empty regions and Δ_2 the set of pairs which have exactly one empty region[5], we penalize differently the deviation of the aforementioned objective:

$$J = \sum_{n=1}^{N} \left\{ c_{n,1} + \sum_{k=1}^{K-1} x_{n,k}(c_{n,k+1} - c_{n,k}) \right\} + C_1 \sum_{\forall (n,n') \in \Delta_1} \delta_{(n,n')} + C_2 \sum_{\forall (n,n') \in \Delta_2} \delta_{(n,n')} \quad (5.13)$$

with $C_2 > C_1 > 0$. We defined C_1 with value of $1/(N(K-1))$ and C_2 with $1/(N(K-1)0.9)$. The factor 0.9 was set empirically. The formulation presented in Equation (5.13) constrained to (5.4), (5.5), (5.8), (5.10) and (5.11) in conjugation with the aggregation approach represented in Figure 5.6, results in our proposal titled *oTreeBagger*.

5.5 Experimental Study

We started by conducting an empirical comparison in an artificial dataset between a standard classification tree (cTree), a standard k-NN and the ordinal decision Tree (oTree) and ordinal k-Nearest Neighbor (ok-NN) models proposed in this work. The comparison study is based on the MER. The experimental study was conducted in Matlab R2009b. The conventional tree model was based on the `classregtree` class, with the labeling rule adapted to use the median of the values instead of the mode. The k-NN used the `knnclassify` function.

We began by generating 1000 examples from the dataset presented in Section 5.1, given by Equation (5.1), and randomly split 50 times the generated dataset into training and test sets. Each model parametrization, namely the pruning level of the tree and the size k of the

[5] $\Delta = \Delta_1 \cup \Delta_2$ and $\Delta_1 \cap \Delta_2 = \emptyset$.

neighborhood of k-NN was selected by 5-fold cross-validation on the training set. Results were averaged over the 50 setups in order to get more robust estimates. This was repeated taking $\ell \in \{100, 300, 500\}$ for size of the training set and $1000 - \ell$ for the test set size. The small size of the dataset allowed us to use directly the 0-1 exact formulation for the relabeling procedure. The test results for are shown in Table 5.2. It can be seen that there

	Training sets size		
Model	$\ell = 100$	$\ell = 300$	$\ell = 500$
cTree	0.47 (0.11)	0.30 (0.05)	0.22 (0.03)
oTree	0.40 (0.10)	0.27 (0.04)	0.22 (0.02)
kNN	0.29 (0.03)	0.24 (0.02)	0.22 (0.02)
okNN	0.28 (0.02)	0.23 (0.02)	0.21 (0.01)

Table 5.2: Mean (standard deviation) of MER over 50 setups of the synthetic dataset.

are no significant differences between the conventional and the proposed models, with only a slightly advantage for the latter. Nevertheless, the proposed models also show higher stability (lower variance) and produce smaller and more consistent models.

Once again we used 2 of the datasets presented in Section 1.2. The test results are shown in Table 5.3, for the MER criterion.

	Datasets	
Model	`SWD`	`LEV`
cTree	0.48 (0.03)	0.45 (0.02)
oTree	0.47 (0.03)	0.45 (0.02)
kNN	0.57 (0.03)	0.58 (0.05)
okNN	0.57 (0.04)	0.56 (0.04)

Table 5.3: Mean (standard deviation) of MER over 50 setups of the datasets.

Again, the same relative behavior is observed in these real datasets. It is also visible that the DT usually attains better results than the k-NN. Even if the proposed framework seems to help improving the performance of a model, that did not always happen. We conjecture that the use of the consistency property only as a post-processing operation may lead to 'over-regularized' or over-smoothed decision functions, effectively hurting or attenuating the positive impact on the generalization performance of the model.

In order to clarify these claims the global constraints approach was extended into the resampling approach on the context of ensemble learning. The baseline method (*TreeBagger*) used in our experiments consisted on the bagging approach with decision trees available in Matlab^{TM} Statistical Toolbox. We opted to use the Gini index as splitting criterion. The grouping size M was evaluated from 1 to 5. The results presented in Figure 5.8 and Figure 5.9 show only the performance for a subset of these values for easier interpretation of the results. In these figures it is also clear the evolution of the learners throughout the increasing number of ensemble components. Due to the sensibility of these learners in regards to the number of training instances used, we conducted our experiments in 10%, 30% and 50% of training data. Our proposal outperformed the standard ensemble learner obtaining considerable gains in terms of performance. Logically, when the number of training instances increases this gain is more subtle, though.

5.6 Discussion

We have provided a new rationale for the incorporation of the order information in the design of classification models intended for ordinal data. The fundamental idea is that adjacent decision region should have equal or consecutive labels. The rationale was then used as a

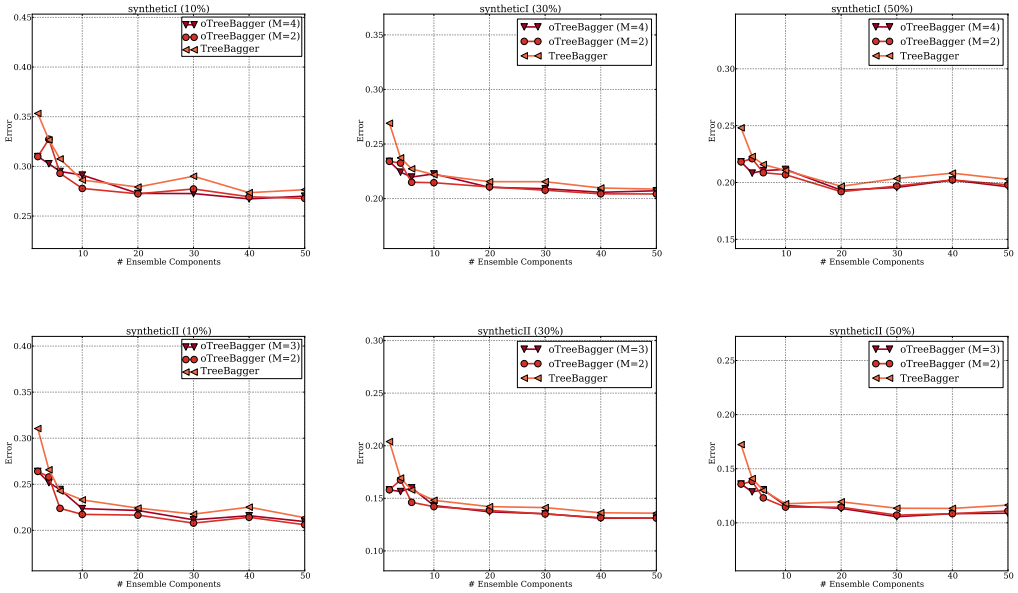


Figure 5.8: Results for synthetic datasets. Models trained with 10%, 30% and 50% of the 1000 instances in the left, center and right plots, respectively.

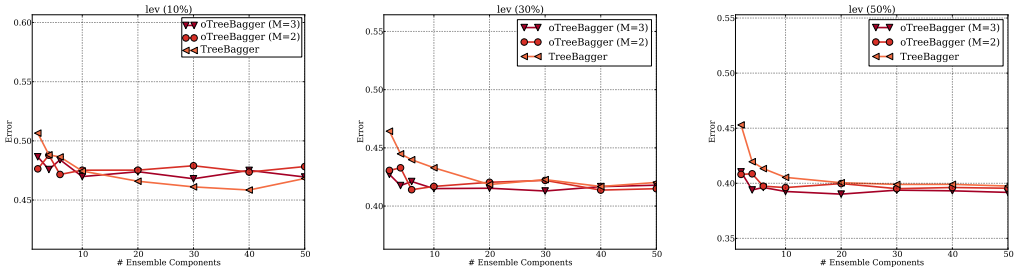


Figure 5.9: Results for a real dataset. Models trained with 10%, 30% and 50% of the 1000 instances in the left, center and right plots, respectively.

post-processing mechanism of a standard decision tree and as a pre- or post-processing step for the k-NN. We have conducted several experiments where our method was tested against standard models from where our method was derived. The results show some advantages of the proposed method. We have also proposed an improvement of Cardoso and Sousa (2010) in order to reduce the over-regularized decision regions artifact through the usage of ensemble learning techniques. Results shown the benefits of our proposal in terms of accuracy gained when compared to a standard ensemble learning technique.

Part III

Reject Option on an Ordinal Setting

Chapter 6

Self-Organizing Maps for Classification with Reject Option*

Real world problems still pose challenges which may not be solvable satisfactorily by the existing learning methodologies used by automatic decision support systems (Gama and de Carvalho, 2012; Goldszmidt et al., 2005; Han and Gao, 2009), leading to many incorrect predictions. This is particularly true for conventional learning systems (e.g. neural networks), in which the number of possible outputs is equal to the number of class labels. For instance, in a binary classification task, the possible outputs are encoded as good (normal) or bad (abnormal) categories.

However, there are situations in which the decision can be postponed, giving the support system the opportunity to identify critical items for posterior revision, instead of trying to automatically classify every and each item. In such cases, the system automates only those decisions which can be reliably predicted, letting the critical ones for a human expert to analyze. Therefore, the development of binary classifiers with a third output class, usually called the *reject class*, is attractive. This approach is known as classification with reject option (Chow, 1970; El-Yaniv and Wiener, 2010; Herbei and Wegkamp, 2006) or soft decision making (Ishibuchi and Nii, 2000).

Roughly speaking, reject option comprises a set of techniques aiming at improving the classification reliability in decision support systems, being originally formalized in the context of statistical pattern recognition in Chow (1970), under the minimum risk theory. Basically, it consists in withholding the automatic classification of an item, if the decision is considered not sufficiently reliable. Rejected patterns can then be handled by a different classifier, or manually by a human. Implementation of reject option strategies requires finding a trade-off between the achievable reduction of the cost due to classification errors, and the cost of handling rejections (which are application-dependent).

Despite its potential advantage, the problem of classification with a reject option has been tackled only occasionally in machine learning literature, in most cases using supervised learning methods, such as the SVM and MLP classifiers. For example, one can reformulate or adapt the SVM method to deal with the reject option problem (Fumera and Roli, 2002; Sousa et al., 2009), by learning the reject region during training. Other SVM-based approaches encompass the use of Neyman-Pearson hypothesis testing on SVMs (Bounsiar et al., 2008), or even the formulation of a new loss function (Bartlett and Wegkamp, 2008). Modifications of supervised neural network classifiers to deal with the reject option date back to the first half of the 1990s (Cordella et al., 1995a;b; Vasconcelos et al., 1993), but there are also more recent works on this issue (De Stefano et al., 2000; Fumera et al., 2003; Gasca et al., 2011; Lotte et al., 2008; Santos-Pereira and Pires, 2005; Suutala et al., 2004).

*Some portions of this Chapter appeared in Sousa et al.

As mentioned, classification strategies with reject option are implemented using supervised classifiers (e.g. SVM, MLP and LVQ). As a feasible alternative to them, the SOM (Kohonen, 1990), originally an unsupervised learning algorithm, has been successfully applied to supervised pattern classification tasks (Mattos and Barreto, 2011; Sim and Sági-Kiss, 2011; Souza Júnior et al., 2011; Turky and Ahmad, 2010). Much before, Kohonen himself had already introduced the neural phonetic typewriter (Kohonen, 1988), in which the SOM is applied to a supervised speech recognition problem. To the best of our knowledge, the SOM has not been evaluated before as a classifier with rejection option.

From the exposed, in this chapter we develop two novel variants of the SOM network to act as supervised classifiers with reject option, and compare their performances with that of the MLP classifier. For this purpose, we promote a comprehensively evaluation of the performances of the proposed SOM-based classifiers on two synthetic and one real-world data set.

6.1 Basics of Classification with Reject Option

As mentioned before, in possession of a "complex" dataset (e.g. from a medical diagnosis problem), every classifier is bound to misclassify some data samples. Depending on the costs of the errors, misclassification can lead to very poor classifier's performance. Therefore, techniques where the classifier can abstain from providing a decision by delegating it to a human expert (or to another classifier) is very appealing. In the following, we limit the discussion of reject option strategies to the binary classification problem. For that, we assume that the problem (and hence, the data) involves only two classes, say $\{\mathcal{C}_{-1}, \mathcal{C}_{+1}\}$, but the classifier must be able to output a third one, the reject class $\{\mathcal{C}_{-1}, \mathcal{C}_{\text{Reject}}, \mathcal{C}_{+1}\}$).

The design of classifiers with reject option can be systematized in three different approaches:

1. **Method 1**: It involves the design of a single, standard binary classifier. If the classifier provides some approximation to the a posteriori class probabilities, $\mathcal{P}(\mathcal{C}_k|\boldsymbol{x})$, $k = 1, 2, ..., K$, then a pattern is rejected if the largest value among the K posterior probabilities is lower than a given threshold, say β $(0 \leq \beta \leq 1)$ (Fumera and Roli, 2002). More formally, according to Chow (1970) one holds a decision if

$$\max_k [\mathcal{P}(\mathcal{C}_k|\boldsymbol{x})] < \beta, \tag{6.1}$$

   or, equivalently,

$$\max_k [\mathcal{P}(\boldsymbol{x}|\mathcal{C}_k)\mathcal{P}(\mathcal{C}_k)] < \beta, \tag{6.2}$$

   where $\mathcal{P}(\mathcal{C}_k)$ is the a priori probability distribution of the k-th class and $\mathcal{P}(\boldsymbol{x}|\mathcal{C}_k)$ is the conditional probability density for the pattern $\boldsymbol{x}$ given the k-th class. If the classifier does not provide probabilistic outputs, then a rejection threshold targeted to the particular classifier's output should be used (Ishibuchi and Nii, 2000). In this case, reject the classification of $\boldsymbol{x}$ if

$$\max_k \{o_k\} < \beta, \tag{6.3}$$

   where o_k is the k-th output of the classifier, $k = 1, 2..., K$. For the binary classification problem, we have $K = 2$.

   For this method, the classifier is trained as usual (i.e. without referring to an explicit rejection class); but rather, the rejection region is determined *after* the training phase, heuristically or based on the optimization of some post-training criterion that weighs the trade-off between the costs of misclassification and rejection.

2. **Method 2**: The design of two, *independent*, classifiers. A first classifier is trained to output $\mathcal{C}_{-1}$ only when the probability of $\mathcal{C}_{-1}$ is high and a second classifier trained to output $\mathcal{C}_{+1}$ only when the probability of $\mathcal{C}_{+1}$ is high. When both classifiers agree on the decision, the corresponding class is outputted. Otherwise, in case of disagreement, the reject class is the chosen one. The intuitive idea behind this approach is that if both classifiers have high levels of confidence in their decisions then the aggregated decision should be correct in case of agreement. In case of disagreement, the aggregated decision is prone to be unreliable and hence rejection would be preferable (Chow, 1970; Fumera et al., 2000a;b).

3. **Method 3**: The design of a single classifier with embedded reject option; that is, the classifier is trained following optimality criteria that automatically take into account the costs of misclassification and rejection in their loss functions, leading to the design of algorithms specifically built for this kind of problem (Bounsiar et al., 2008; Fumera and Roli, 2002).

Later in this chapter, we will introduce two SOM-based strategies that instantiate the classification with reject option paradigms described above as Methods 1 and 2.

6.1.1 Related Works

In one of the first works to analyze the tradeoffs between erring and rejecting, Chow (1970) derived a general error and reject tradeoff relation for the Bayes optimum recognition system. This derivation assumed a complete knowledge of the a priori probability distribution of the classes and the posterior probabilities which, in real problems, are usually unknown. Fumera et al. (2000a;b) shows that Chow's rule does not perform well if a significant error in probability estimation is present, proposing the use of multiple reject thresholds related to the data classes.

The incorporation of reject option opens new fields of applications for a learning method. For instance, application to Multiple Instance Learning (MIL) for image categorization as presented in Zhang and Metaxas (2006), the improvement of reliability in banknote neuro-classifier (Ahmadi et al., 2004) through the use of Principal Component Analysis (PCA) and a Learning Vector Quantization (LVQ), among others.

The introduction of the reject option in a classifier also demands the introduction of new evaluation measures. In Ferri and Hernández-Orallo (2004) new measures are developed to find a relation between the reduction of the number of misclassified instances and the reduction of the number of unclassified instances. Despite the results obtained and presented, they claim that their measures can not be statistically interpreted and henceforth no formal interpretation can be taken (Ferri and Hernández-Orallo, 2004). Following this idea, in Ferri et al. (2004) the concept of delegating classifiers in a systematic way is developed. These type of methods follow the concept of divide-to-conquer (Ferri and Hernández-Orallo, 2004; Ferri et al., 2004; Gama and Brazdil, 2000), where a more generic classifier abstains on a part of the examples and delegates them to a second, more specific, classifier. However, such approaches could potentially delegate only a small number of instances to the second classifier which will lead to overfitting (Ferri et al., 2004).

Based on the ROC curve principle, as in Ferri and Hernández-Orallo (2004), a cost-sensitive reject rule for SVM classifiers is introduced in Tortorella (2004). Other strategies are taken in Tortorella (2005) and Pietraszek (2005) where a reject rule based on the ROC curve is specially designed for binary classifiers.

In Landgrebe et al. (2004) the authors explored the idea of combining one-class learning models with supervised learning. They further evaluated their strategy concerning the incorporation of a reject option on classification tasks through ROC analysis (Landgrebe et al., 2006). The measures delved in Landgrebe et al. (2006) aid in choosing and optimizing a

classifier that reduces the risk of misclassifying an unseen class (outlier). Another system to identify outliers, in contrast with those proposed in Landgrebe et al. (2004; 2006), is presented in Tax and Duin (2008). The authors propose a heuristic which combines any type of one-class models for solving multi-class classification problem with outlier rejection. This is achieved through the use of two models: density and distance based class models. In this scheme, PCA is used to avoid the dimensionality problem. Instead of rejecting outlier instances, in Le Capitaine and Fré andlicot (2010) it is suggested a new rejection scheme. Their technique encompasses the rejection of instances from one class determined as outlier and the assignment of instances to the remaining classes.

Other approaches can be taken. If the probability density functions of classes are known, pattern recognition is a problem of statistical hypothesis testing (Fukunaga, 1990). Keeping in mind the minimization of the empirical risk principle, in Bounsiar et al. (2006) it is proposed a kernel learning method. This technique consists in a likelihood ratio based classifier where a Parzen window estimator is used to estimate the probability densities. In Bounsiar et al. (2008), the authors follow the statistical hypothesis testing rationale a little further through the use of the Neyman-Pearson (NP) criterion. NP does not introduce any new decision theory since it relies on the likelihood test as Bayes theory (Fukunaga, 1990). However, this criterion has a more natural way to specify a constraint on the false alarm (type I error) probability than to assign costs to the different kinds of errors. Based on this, a reject option method based on the Neyman-Pearson criterion is presented as an extension of the Chow's rule.

Although several learning methods exist addressing the reject option, only a few tackle the assessment of the sensibility. Devarakota et al. (2008) present a generic approach where, through the quantification of uncertainty of a decision made by a statistical learning scheme, the method computes a confidence interval which can afterward be used on several learning techniques.

Despite the myriad of techniques that handle the incorporation of a reject option in their approaches, many of them do not fully account the pioneer work of Chow (1970). Also, the principle issue usually used in pattern recognition, which is the minimization of the empirical risk, is feebly explored on the reject option case. Moreover, a major difficulty with these approaches is that the resulting formulations are no longer standard optimization procedures and cannot be solved efficiently, lacking some appealing features like convexity and sparsity. In this line Bartlett and Wegkamp (2008); Yuan and Wegkamp (2010) consider a convex surrogate of the generalized loss function to efficiently solve the resulting problem under SVM and of the convex loss functions. As an extension of this, in Grandvalet et al. (2008) it is proposed a double hinge function and a probabilistic viewpoint of the SVM fitting. Without changing the loss function, in (Fumera and Roli, 2002) it is proposed a modified SVM.

6.2 The Self-Organizing Map

The SOM (Kohonen, 1982; 1990) is one of the most popular neural network architectures. It belongs to the category of unsupervised competitive learning algorithms and it is usually designed to build an ordered representation of spatial proximity among vectors of an unlabeled data set. The SOM has been widely applied to pattern recognition and classification tasks, such as clustering, vector quantization, data compression and data visualization. In these applications, the weight vectors are called *prototypes* or *centroids* of clusters of input vectors, being obtained usually through a process of learning.

The neurons in the SOM are put together in an output layer, $\mathcal{A}$, in one-, two- or even three-dimensional arrays. Each neuron $j \in \mathcal{A}$, $j = 1, 2, \ldots, q$, has a weight vector $\boldsymbol{w}_j \in \mathbb{R}^n$ with the same dimension of the input vector $\boldsymbol{x} \in \mathbb{R}^n$. The network weights are trained according to a competitive-cooperative learning scheme in which the weight vector of a winning neuron (also

called, the Best Matching Unit (BMU)) and its neighbors in the output array are updated after the presentation of an input vector. Roughly speaking, the functioning of this type of learning algorithm is based on the concept of *winning neuron*, defined as the neuron whose weight vector is the closest to the current input vector.

Using Euclidean distance, the simplest strategy to find the winning neuron, $i(k)$, is given by:

$$i(k) = \arg\min_{\forall j} \|\boldsymbol{x}(k) - \boldsymbol{w}_j(k)\| \tag{6.4}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ denotes the current input vector, $\boldsymbol{w}_j(k) \in \mathbb{R}^n$ is the weight vector of neuron j, and k denotes the current iteration of the algorithm. Accordingly, the weight vectors are adjusted by the following recursive equation:

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \eta(k)h(j,i;k)[\boldsymbol{x}(k) - \boldsymbol{w}_j(k)], \tag{6.5}$$

where $h(j,i;k)$ is a Gaussian function which control the degree of change imposed to the weight vectors of those neurons in the neighborhood of the winning neuron:

$$h(j,i;k) = \exp\left(-\frac{\|\boldsymbol{r}_j(k) - \boldsymbol{r}_i(k)\|^2}{\sigma^2(k)}\right) \tag{6.6}$$

where $\sigma(k)$ defines the radius of the neighborhood function, $\boldsymbol{r}_j(k)$ and $\boldsymbol{r}_i(k)$ are, respectively, the coordinates of neurons j and i in the array. The learning rate, $0 < \eta(k) < 1$, should decrease gradually with time to guarantee convergence of the weight vectors to stable states. In this chapter, we use $\eta(k) = \eta_0 \left(\eta_T/\eta_0\right)^{(k/T)}$, where η_0 and η_T are the initial and final values of $\eta(k)$, respectively. The variable $\sigma(k)$ should also decrease with time similarly to the learning rate $\eta(k)$.

The SOM has several features which make it a valuable tool in data mining applications (Peng and Zhu, 2007). For instance, the use of a neighborhood function imposes an order to the weight vectors, so that, at the end of the training phase, input vectors that are close in the input space are mapped onto the same winning neuron or onto winning neurons that are close in the output array. This is the so-called *topology-preserving property* of the SOM, which has been particularly useful for data visualization purposes (Flexer, 2001).

Once the SOM converges, the set of ordered weight vectors summarizes important statistical characteristics of the input (see Figure 6.1). The SOM should reflect variations in the statistics of the input distribution: regions in the input space $\mathcal{X}$ from which a sample $\boldsymbol{x}$ are drawn with a high probability of occurrence are mapped onto larger domains of the output space A, and therefore with better resolution than regions in $\mathcal{X}$ from which sample vectors are drawn with a low probability of occurrence.

For the interested reader, further information about the SOM and applications can be found in van Hulle (2010) and Yin (2008).

6.2.1 SOM for Supervised Classification

In order to use the SOM for supervised classification, modifications are necessary in its original learning algorithm. There are many ways to do that (see Mattos and Barreto (2011) and references therein), but in the present chapter we will resort to two well-known strategies.

Strategy 1: The first strategy involves a post-training neuron labeling. It consists firstly in training the SOM in the usual unsupervised way until convergence of the weights. Once training is finished, one has to present the whole training data once again to the SOM in order to find the winning neuron for each pattern vector.

A given neuron can be selected the winner for pattern vectors belonging to different classes. However, among all the patterns a given neuron was selected the winner, the number of exemplars of a given class usually is higher that the number of exemplars of other classes.

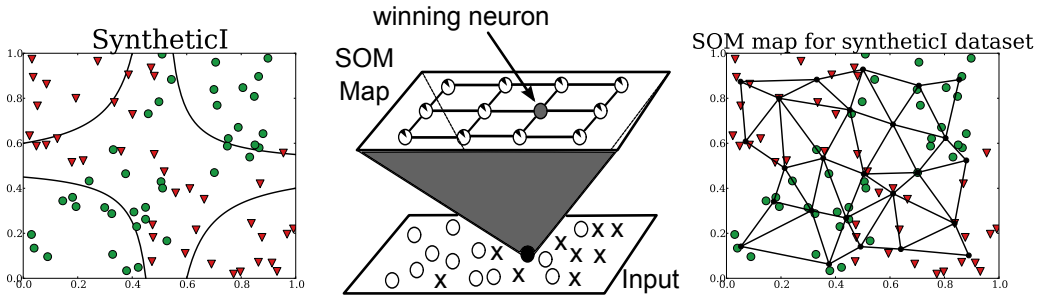


Figure 6.1: Example of a SOM as a compact, topology-preserving, representation of a synthetic dataset (left figure). A mapping (ϕ) is learned in order to reflect the input data distribution (center figure). Representation of the distribution of the weight vectors of the SOM in the input space, where neighboring prototypes in the output grid are shown connected in the input space (right figure).

Hence, a class label is assigned to a neuron on a majority voting basis, i.e. a neuron receives the label of the class with the highest number of exemplars.

Two undesirable situations may occur: (*i*) ambiguity or (*ii*) dead neurons. Ambiguity occur when the frequency of the class labels of the patterns mapped to a given neuron are equivalent. Dead neurons are those never selected as winner for any of the input patterns. In these cases, the neuron can be pruned (i.e. disregarded) from the map, or even be tagged with a "rejection class" label. This rejection option approach is somewhat too empirical and, hence, not considered here. Instead, we adopt a more systematic and principled approach based on Chow's work (Chow, 1970).

Strategy 2: The second strategy, usually called the *self-supervised SOM* training scheme, is the one used by Kohonen for the neural phonetic typewriter (Kohonen, 1988). According to this strategy, the SOM is made supervised by adding class information to each input pattern vector. Specifically, the input vectors $\boldsymbol{x}(k)$ are now formed of two parts, $\boldsymbol{x}_p(k)$ and $\boldsymbol{x}_l(k)$, where $\boldsymbol{x}_p(k)$ is the pattern vector itself, while $\boldsymbol{x}_l(k)$ is the corresponding class label of $\boldsymbol{x}_p(k)$. During training, these vectors are concatenated to build augmented vectors $\boldsymbol{x}(k) = [\boldsymbol{x}_p(k)\ \boldsymbol{x}_l(k]^T$ which are used as inputs to the SOM. The corresponding augmented weight vectors, $\boldsymbol{w}_j(k) = [\boldsymbol{w}_j^p(k)\ \ \boldsymbol{w}_j^l(k)]^T$, are adjusted as in the usual SOM training procedure.

Usually, the label vector $\boldsymbol{x}_l(k)$ is represented as a unit-length binary vector; that is, only one of its components is set to "1", while the others are set to "0". The index of the "1" position indicates the class of the pattern vector $\boldsymbol{x}_p(k)$. For example, if three classes are available, then three label vectors are possible: one for the first class ([1 0 0]), one for the second class ([0 1 0]) and one for the third class ([0 0 1]).

For the classification of an unknown pattern $\boldsymbol{x}(k)$, the $\boldsymbol{x}_l(k)$ part is not considered, i.e. only its $\boldsymbol{x}_p$ part is compared with the corresponding part of the weight vectors. However, the class label of the unknown pattern vector is decided on the basis of the $\boldsymbol{w}_i^l(k)$ part of the winning weight vector $\boldsymbol{w}_i(k)$. The index of the component of $\boldsymbol{w}_i^l(k)$ with largest value defines the class label of the unknown pattern vector $\boldsymbol{x}_p$.

6.2.2 Learning SOM with Costs

A natural extension of the aforementioned approaches can be performed by merging LVQ techniques (referred in Section 6.2.1) with SVM approaches (Graepel et al., 1998). Graepel et al. (1998) present a set of different LVQ SOM models through the learning of a cost function. Knowledge is incorporated from data and neighborhood information leading to a reformulation of the function expressed in Equation (6.5). Similar technique was also

employed by Hammer et al. (2002). Fuzzy approaches (Abonyi et al., 2003; Pascual-Marqui et al., 2001) are also very common to define a better neighborhood contribution for the map adaptation rule. However, the computational complexity involved in the design and application of these models may be unbearable.

6.2.3 Incorporating the Reject Option into the SOM: Two Proposals

Devising SOM-based algorithms with the reject option for supervised pattern classification is appealing mainly due to the SOM's properties of density estimation and topology preservation (for data visualization). If we take advantage of these properties, it is possible to devise new SOM-based approaches to learn rejection regions. We argue that endeavoring this in an (originally) unsupervised learning method, can permit further analysis of the results towards better decision making. Bearing this in mind, we will introduce two different strategies to incorporate the reject option into the SOM.

The rationale for the two proposals is based on the intuitive idea that if in a specific region of the input manifold one has a major cluster of neurons, one can easily realize that a high concentration of patterns is present, and if all they share the same label, one can say for a certain degree of confidence which label is likely to define that region.

Both proposals to be described require the estimation of $\mathcal{P}(\boldsymbol{x}|\mathcal{C}_k)$ (or, equivalently, $\mathcal{P}(\mathcal{C}_k|\boldsymbol{x})$) using the distribution of SOM's weight vectors. An optimal threshold value has to be determined in order to re-tag some of the weight vectors with the rejection class label. In this chapter we will provide three techniques to obtaining suitable estimates of the posterior probability $\mathcal{P}(\boldsymbol{x}|\mathcal{C}_k)$.

The first proposal will be referred to as the *ROSOM-1C* methodology, since it requires only one SOM network, trained in the usual unsupervised way. The second proposal consists in training two SOMs, one is trained to become specialized on the class of negative examples, say, class $\mathcal{C}_{-1}$, while the other is trained to become specialized on the class of positive examples, say, class $\mathcal{C}_{+1}$. The decision to reject a given pattern will be determined based on the combination of results provided by the outputs of each map. This approach will be referred as the *ROSOM-2C* methodology along the remainder of the chapter.

6.3 SOM with Reject Option Using One Classifier

Roughly speaking, the ROSOM-1C works as the standard supervised SOM classifier described in Section 6.2.1, except for the fact that some of the neurons are tagged with the *rejection class* label. The main idea behind the proposal of the ROSOM-1C approach relies exactly on developing formal techniques to assign the rejection class label to a given neuron. In greater detail, the design of the ROSOM-1C requires the following steps.

STEP 1 - For a given data set, a number of training realizations are carried out using a single SOM network in order to find the best number of neurons and suitable map dimensions. For this purpose, the conventional unsupervised SOM training is adopted.

STEP 2 - Present the training data once again and label the prototypes $\boldsymbol{w}_j$, $j = 1, ..., q$, according to the mode of the class labels of the patterns mapped to them. No weight adjustments are carried out at this step.

STEP 3 - Based on the SOM's ability to approximate the input data density, we approximate $\mathcal{P}(\boldsymbol{x}|\mathcal{C}_k)$ with $\mathcal{P}(\boldsymbol{w}_j|\mathcal{C}_k, \boldsymbol{x})$, for $j = 1, ..., q$ and $k = 1, ..., K$. In Subsection 6.3.1, we describe two techniques to compute $\mathcal{P}(\boldsymbol{w}_j|\mathcal{C}_k, \boldsymbol{x})$ based on standard statistical techniques, namely, Parzen Windows and Gaussian Mixture Models.

STEP 4: Finding an optimum value for the rejection threshold β requires the minimization of the empirical risk as proposed in Chow (1970):

$$\widehat{R} = w_r R + E \tag{6.7}$$

where R and E are, respectively, the ratio of rejected and misclassified patterns (computed using validation data), while w_r is the rejection cost. The searching procedure is described as follows.

STEP 4.1 - For a given rejection cost w_r, vary β from an initial value β_i to a final value β_f in fixed increments $\Delta\beta$. Typical values are: $\beta_i = 0.55$, $\beta_f = 1.00$ and $\Delta\beta = 0.05$.

STEP 4.2 - Then, for each value of β, do

(*i*) Compute $R(\beta) = \dfrac{\text{number of rejected patterns}}{\text{total number of patterns}}$

(*ii*) Compute $E(\beta) = \dfrac{\text{number of misclassified patterns}}{\text{total number of patterns}}$

(*iii*) Compute $\widehat{R}(\beta)$ as in Equation (6.7).

STEP 4.3 - Select the optimum rejection threshold (β_o) according to the following rule:

$$\beta_o = \arg\min_{\beta}\{\widehat{R}(\beta)\}. \tag{6.8}$$

STEP 5: Re-label the prototypes using the following rule:

$$\begin{aligned} &\text{IF} && \max_k\{\mathcal{P}(\mathcal{C}_k)\mathcal{P}(\boldsymbol{w}_i|\mathcal{C}_k, \boldsymbol{x})\} < \beta_o \\ &\text{THEN} && \text{change class}(\boldsymbol{w}_i) \text{ to } \textit{Rejection Class}, \\ &\text{ELSE} && \text{keep class}(\boldsymbol{w}_i) \text{ as determined in STEP 2.} \end{aligned} \tag{6.9}$$

Once the prototypes have been re-labeled, the following decision rule is used for classifying new incoming patterns:

$$\begin{aligned} &\text{IF } \boldsymbol{w}_i \text{ is the winning prototype for pattern } \boldsymbol{x}(n), \\ &\text{THEN reject } \boldsymbol{x}(n) \text{ if class}(\boldsymbol{w}_i) = \textit{Rejection Class}, \\ &\text{ELSE class}(\boldsymbol{x}(n)) \leftarrow \text{class}(\boldsymbol{w}_i). \end{aligned} \tag{6.10}$$

6.3.1 On the Estimation of $\mathcal{P}(\mathbf{w}_j|\mathcal{C}_k, \mathbf{x})$

The first approach to be used to compute SOM-based estimates of $\mathcal{P}(\boldsymbol{w}_j|\mathcal{C}_k)$ is through the Parzen windows nonparametric method. The estimation is usually performed by some kernel function, usually a Gaussian, averaged by the number of points belonging to a given class. It is therefore given by

$$\mathcal{P}(\boldsymbol{w}_j|\mathcal{C}_k, \boldsymbol{x}) = \frac{1}{N_k}\sum_{i=1}^{N_k} \frac{1}{h^d(2\pi)^{\frac{d}{2}}|\boldsymbol{C}_k|^{\frac{1}{2}}} \exp\left(-\frac{Q(\boldsymbol{x}^{(k)}, \boldsymbol{w}_j)}{2h^2}\right) \tag{6.11}$$

with

$$Q(\boldsymbol{x}^{(k)}, \boldsymbol{w}_j) = (\boldsymbol{x}_i^{(k)} - \boldsymbol{w}_j)^T \boldsymbol{C}_k^{-1}(\boldsymbol{x}_i^{(k)} - \boldsymbol{w}_j), \tag{6.12}$$

where h is the width of the Gaussian window, $\boldsymbol{x}_i^{(k)}$ is the ith pattern of the kth class, $\boldsymbol{C}_k$ is the covariance matrix estimated from the training instances belonging to the k-th class (i.e. $\mathcal{C}_k$), N_k the number of elements of the kth class and d is the dimension of $\boldsymbol{x}_i^{(k)}$ and $\boldsymbol{w}_j$.

Another approach that can also be used to estimate $\mathcal{P}(\boldsymbol{w}_j|\mathcal{C}_k, \boldsymbol{x})$ based on the distribution of weight vectors of the SOM is the Gaussian Mixture Models (GMM) (Alhoniemi et al., 1999; Holmström and Hämäläinen, 1993; Riveiro et al., 2008; Seo and Obermayer, 2002; Utsugi, 1998; Yin and Allinson, 2001). In this chapter we follow the approach developed by Alhoniemi et al. (1999), which is implemented in the SOM toolbox[2].

6.3.2 Neuron Re-Labeling Based on Gini Index

For the application of the decision rule in (6.9), one has to store all the values of the posterior probabilities estimates $\mathcal{P}(\mathcal{C}_k|\boldsymbol{w}_j, \boldsymbol{x}) \propto \mathcal{P}(\mathcal{C}_k)\mathcal{P}(\boldsymbol{w}_j|\mathcal{C}_k, \boldsymbol{x})$ for each neuron j. The quantity $\mathcal{P}(\mathcal{C}_k|\boldsymbol{w}_j, \boldsymbol{x})$ express the probability of an instance that has fallen within the Voronoi cell of neuron j to belong to class $\mathcal{C}_k$. By means of concepts borrowed from information theory, it is possible to merge all the probabilities $\mathcal{P}(\mathcal{C}_k|\boldsymbol{w}_j, \boldsymbol{x})$, $k = 1, ..., K$, associated with a given neuron, into a single quantity to be called *cell impurity*.

Roughly, the impurity of neuron (or cell) j is a measure of the entropy of the class labels of the patterns mapped to this neuron. If the entropy is high, the distribution of class labels is more or less uniform (i.e. no class label dominates over the others). If the entropy is low, one class label clearly dominates over the others. In order to quantify the inequality of class labels distribution within a neuron, one can resort to the Gini coefficient (Giles, 2004; Gini, 1921). In the present context, this measurement is given by

$$G_j = 1 - \sum_{k=1}^{K} \mathcal{P}^2(\mathcal{C}_k|\boldsymbol{w}_j, \boldsymbol{x}), \quad j = 1, ..., q \tag{6.13}$$

where $\mathcal{P}(\mathcal{C}_k|\boldsymbol{w}_j, \boldsymbol{x})$ can be, for simplicity, computed as the frequency of instances within the Voronoi cell belonging to the class $\mathcal{C}_k$. Ideally, the desirable situation is to have always low values for the Gini coefficient, indicating predominance of a certain class label within neuron j. Neurons located at the borders of decision regions usually have high Gini coefficients, indicating higher entropy in the frequency of class labels within those neurons and, hence, a lower confidence in labeling them with a specific class label.

Using the Gini coefficient measure, the decision rule in (6.9) is now written as the following decision rule:

$$\begin{array}{rl} \text{IF} & G_i > \beta_o \\ \text{THEN} & \text{reject } \boldsymbol{x}(n), \\ \text{ELSE} & \text{class}(\boldsymbol{x}(n)) = \text{class}(\boldsymbol{w}_i). \end{array} \tag{6.14}$$

where i is the index of the winning neuron for the current input pattern $\boldsymbol{x}(n)$.

Labeling neurons as reject based on the Gini coefficient can be governed by the following rule:

$$\text{reject if} \quad G > t, \text{or equivalently}, 1 - G < 1 - t \Leftrightarrow p_1^2 + p_2^2 < 1 - t \tag{6.15}$$

Noticing that both the functions $f(p_1, p_2) = \max(p_1, p_2)$ and $g(p_1, p_2) = p_1^2 + p_2^2$, defined in $p_1 + p_2 = 1$, are symmetric, convex and attain the minimum of 0.5 at $(p_1; p_2) = (0.5; 0.5)$, it is trivial to show that for any decision rule based on Equation (6.1) with a certain threshold t_1 there exists an equivalent rule based on Equation (6.15) with a certain threshold t_2 leading to exactly the same reject region. As a side note, this would not be true for more than 2 classes, since the max function and Gini-based function have level curves of different 'shapes' in higher dimensions (selecting a threshold corresponds to selecting a given level curve).

Figure 6.2 shows the results of a ROSOM-1C classifier for synthetic dataset (see Section 6.5) using the Gini coefficient approach. Each neuron has been initially trained and labeled, respectively, according to Steps 1 and 2 of the design procedure. Once the optimum rejection threshold has been determined, decision for rejection are made based on (6.19).

[2]Available for download at `http://www.cis.hut.fi/somtoolbox/`

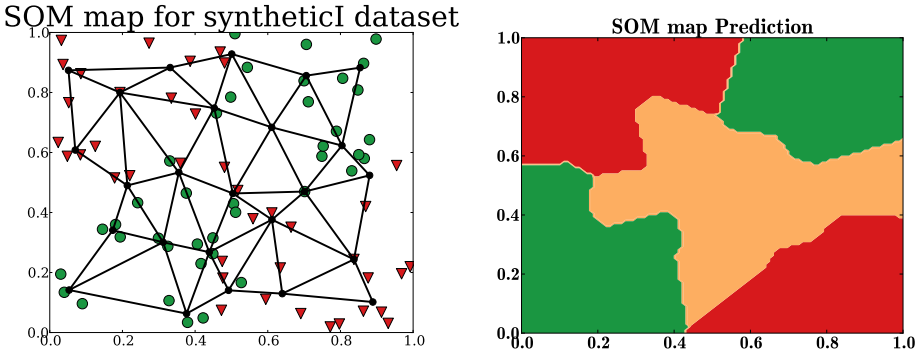


Figure 6.2: On the lefthand figure it is shown a trained ROSOM-1C classifier using the Gini coefficient approach for a synthetic dataset. The righthand figure depicts a class prediction results for a given testing data, where the red and green colors denote the decision classes and beige the reject decisions.

$$\boldsymbol{w}_j(n+1) = \boldsymbol{w}_j(n) + \begin{cases} \eta(n)h(i,j;n)[\boldsymbol{x}(n) - \boldsymbol{w}_j(n)]w_r, & \text{if class}(\boldsymbol{x}(n)) = \mathcal{C}_{+1} \\ \eta(n)h(i,j;n)[\boldsymbol{x}(n) - \boldsymbol{w}_j(n)](1-w_r), & \text{if class}(\boldsymbol{x}(n)) = \mathcal{C}_{-1}. \end{cases} \tag{6.16}$$

6.4 SOM with Reject Option Using Two Classifiers

As mentioned before, the second proposal requires two SOMs. One is trained to become specialized on the class of negative examples, say, class $\mathcal{C}_{-1}$, while the other is trained to become specialized on the class of positive examples, say, class $\mathcal{C}_{+1}$. An explicit estimation of the posterior class probability is not required since, in the case of using two SOMs, the maps will be tuned for a specific class.

Thus, by stating that one SOM will be trained to become specialized on the $\mathcal{C}_{-1}$ class we mean that instances from this class will be "preferred" (i.e. will be given more importance during training) over the patterns belonging to the other class. This preference may be expressed in terms of a weight, which will be equivalent to a cost C_{high} and C_{low} ($C_{low} < C_{high}$) for the patterns of the classes $\mathcal{C}_{-1}$ and $\mathcal{C}_{+1}$, respectively—see Figure 6.3 (left) and Figure 6.3 (center). These costs are related to the rejection cost w_r:

$$w_r = \frac{C_{low}}{C_{high}}, \tag{6.18}$$

where a low (high) w_r indicates a low (high) rejection cost; that is, many (few) patterns are rejected. However, to incorporate these costs on SOMs in a principled (mathematically oriented) way may be difficult due to the lack of a suitable objective function that gives rise to the learning rules in Equation (6.5). As a consequence, our proposal consists in including these costs directly on the SOM learning rule and evaluate empirically the resulting classifier.

The design ofthe ROSOM-2C classifier requires the following steps.

STEP 1 - Choose a rejection cost $w_r = C_{low}/C_{high}$.

STEP 2 - Train two SOM networks following the self-supervised SOM training scheme describe in Subsection 6.2.1.

STEP 2.1 - Train the first SOM network, henceforth named SOM-1 classifier, to become specialized on the class $\mathcal{C}_{-1}$. For that, we replace the standard SOM learning rule with Equation (6.16).

$$\boldsymbol{w}_j(n+1) = \boldsymbol{w}_j(n) + \begin{cases} \eta(n)h(i,j;n)[\boldsymbol{x}(n) - \boldsymbol{w}_j(n)](1-w_r), & \text{if class}(\boldsymbol{x}(n)) = \mathcal{C}_{+1} \\ \eta(n)h(i,j;n)[\boldsymbol{x}(n) - \boldsymbol{w}_j(n)]w_r, & \text{if class}(\boldsymbol{x}(n)) = \mathcal{C}_{-1}. \end{cases} \tag{6.17}$$

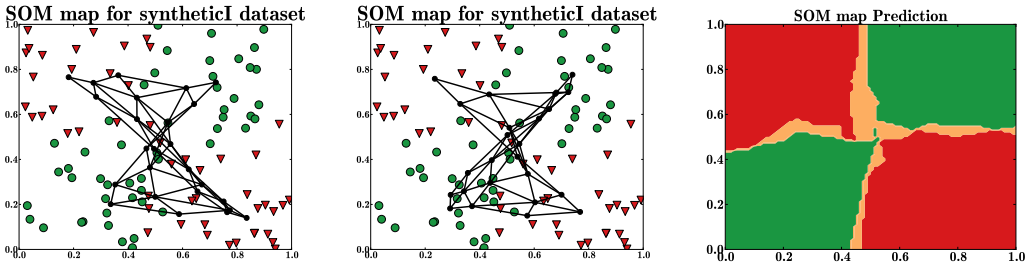


Figure 6.3: The figures on the left and center present the trained SOM-1 and SOM-2 networks, respectively. If both agree on the outcome a decision is emitted (green or red). Otherwise, instances are rejected (beige).

STEP 2.2 - Train the second SOM network, henceforth named SOM-2 classifier, to become specialized on the class $\mathcal{C}_{+1}$. For that, we replace the standard SOM learning rule with Equation (6.17).

STEPS 3, 4 and 5 - The same as the ones described for the ROSOM-C1 classifier. The Gini coefficient approach can also be used to re-label the prototypes of the ROSOM-2C classifier.

Once the ROSOM-2C classifier is trained, a new incoming pattern $\boldsymbol{x}(n)$ can be classified or rejected by the application of the following procedure:

- Find the winning prototype $\boldsymbol{w}_{i_1}$ for $\boldsymbol{x}(n)$ in SOM-1.
- Find the winning prototype $\boldsymbol{w}_{i_2}$ for $\boldsymbol{x}(n)$ in SOM-2

$$\begin{aligned} &\text{IF} && \text{class}(\boldsymbol{w}_{i_1}) = \text{class}(\boldsymbol{w}_{i_2}), \\ &\text{THEN} && \text{class}(\boldsymbol{x}(n)) \leftarrow \text{class}(\boldsymbol{w}_{i_1}), \\ &\text{ELSE} && \text{reject } \boldsymbol{x}(n). \end{aligned} \tag{6.19}$$

Figure 6.3 illustrates the decision regions found produced by a ROSOM-2C classifier for synthetic dataset (details are given in Section 6.5).

6.5 Experimental Study

The performance of the classification methods were assessed over five datasets. The first two were synthetically generated; the remainder datasets includes real-world data.

As in Cardoso and da Costa (2007), for the synthetic dataset (`syntheticI`), we began by generating 400 points $\mathbf{x} = [x_1\ x_2]^T$ in the unit square $[0,1] \times [0,1] \subset \mathbb{R}^2$ following a uniform distribution. Then, we assigned to each example $\mathbf{x}$ a class $y \in \{-1, +1\}$ corresponding to

$$y = \begin{cases} t, & t \neq 0 \\ +1, & t = 0 \wedge \varepsilon_2 < \alpha \\ -1, & t = 0 \wedge \varepsilon_2 > \alpha \end{cases},$$

where $t = \min_{r \in \{-1,0,+1\}} \{r : b_{r-1} < \alpha + \varepsilon_1 < b_r\}$, $\alpha = 10(x_1 - 0.5)(x_2 - 0.5)$, $\varepsilon_1 \sim N(0, 0.125^2)$, $\varepsilon_2 \sim \text{Uniform}(b_{-1}, b_0)$ and $(b_{-2}, b_{-1}, b_0, b_1) = (-\infty; -0.5; 0.25; +\infty)$.

w_r	0.44		0.24		0.04	
	Rej.	Perf.	Rej.	Perf.	Rej.	Perf.
(ROSOM-1C) Parzen	0.13	0.90	0.28	0.96	0.49	0.99
(ROSOM-1C) Gini	0.08	0.87	0.25	0.94	0.47	0.98
(ROSOM-1C) GMM	0.06	0.83	0.15	0.85	0.99	1.00
(MLP-1C)	0.29	0.91	0.40	0.96	0.56	0.99

(a) Performance for `syntheticI` dataset with 60% of training data using one classifier.

w_r	0.44		0.24		0.04	
	Rej.	Perf.	Rej.	Perf.	Rej.	Perf.
(ROSOM-1C) Parzen	0.11	0.91	0.27	0.96	0.48	0.99
(ROSOM-1C) Gini	0.07	0.87	0.28	0.94	0.51	0.98
(ROSOM-1C) GMM	0.07	0.83	0.18	0.87	0.95	1.00
(MLP-1C)	0.26	0.92	0.39	0.96	0.57	0.99

(b) Performance for `syntheticI` dataset with 80% of training data using one classifier.

Table 6.1: Performances achieved for `syntheticI` dataset using one classifier.

This distribution creates two plateau uniformly distributed and a transition zone of linearly decreasing probability, delimited by hyperbolic boundaries. A second synthetic dataset of 400 points—`syntheticII`—was generated from two Gaussian in $\mathbb{R}^2$: $\boldsymbol{y}_{-1} \sim N\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}9 & 0\\0 & 9\end{bmatrix}\right)+\varepsilon$ and $\boldsymbol{y}_{+1} \sim N\left(\begin{bmatrix}+2\\+2\end{bmatrix}, \begin{bmatrix}25 & 0\\0 & 25\end{bmatrix}\right)+\varepsilon$ corresponding to classes $\{-1, +1\}$ respectively, where ε follows a uniform distribution in $[0.025, 0.25]$. The real-world dataset is a subset of `letter` problem as described in Section 1.2.

In the computer experiments, we used the SOM toolbox for implementing the ROSOM-1C and ROSOM-2C classifiers and the Matlab$^{\text{TM}}$ Neural Networks toolbox for MLP-based classifiers. For fair performance comparison, we have instantiated the same rejection option strategies used for the SOM-based classifiers into the MLP-based classifiers, giving rise to the MLP-1C and MLP-2C classifiers. Since we have trained the MLP-based classifiers to estimate the posterior probabilities, decisions for the MLP-1C classifier are obtained simply through the application of the rule in 6.1. For the MLP-2C classifier, each individual network penalizes differently the misclassifications according to the same costs as presented for the ROSOM-2C classifier.

For the SOM-based classifiers a two-dimensional map was used in the experiments with a hexagonal neighborhood structure and a Gaussian neighborhood function. For determining the best parameterization, we conducted a 5-fold cross validation in order to find the best number of neurons and the initial radius size for the neighborhood function. Our search considered a squared map spanning 5×5 to 25×25 neurons. The learning phase stopped after 200 epochs.

For the MLP-based classifiers, we performed a "grid search" over the number of the neurons that composed the network. The tested range encompassed 5 to 20 neurons with one hidden layer, a single output neuron, and logistic sigmoid as activation function for all neurons. We defined a maximum number of 15 epochs as the stopping criterion in order to avoid overfitting (Caruana et al., 2000). The resilient back-propagation training algorithm was used.

It is important to point out that, in the absence of further insights about the problem at our disposal (other than the data itself), we cannot select only one value for w_r, since its selection is intrinsically application-dependent. Thus, we started by running the classifiers

w_r	0.44		0.24		0.04	
	Rej.	Perf.	Rej.	Perf.	Rej.	Perf.
(ROSOM-2C) Parzen	0.07	0.88	0.12	0.90	0.30	0.96
(ROSOM-2C) Gini	0.04	0.88	0.13	0.91	0.32	0.96
(ROSOM-2C) GMM	0.07	0.89	0.17	0.92	0.44	0.97
(MLP-2C)	0.09	0.90	0.30	0.96	0.66	0.99

(a) Performance for `syntheticI` dataset with 60% of training data using two classifiers.

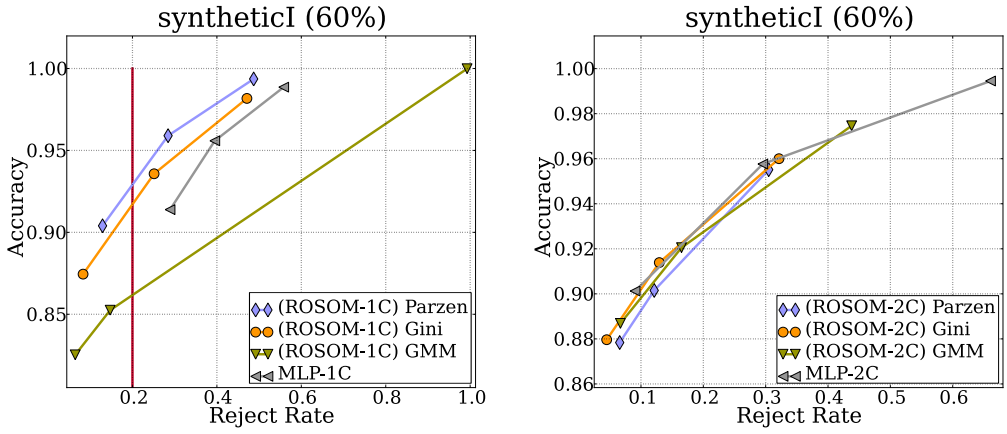
w_r	0.44		0.24		0.04	
	Rej.	Perf.	Rej.	Perf.	Rej.	Perf.
(ROSOM-2C) Parzen	0.08	0.89	0.13	0.91	0.33	0.97
(ROSOM-2C) Gini	0.04	0.88	0.13	0.91	0.32	0.96
(ROSOM-2C) GMM	0.07	0.88	0.15	0.91	0.43	0.98
(MLP-2C)	0.10	0.91	0.30	0.95	0.64	1.00

(b) Performance for `syntheticI` dataset with 80% of training data using two classifiers.

Table 6.2: Performances achieved for `syntheticI` dataset using two classifiers.

spanning three values for w_r in Equation (6.7): 0.04, 0.24 and 0.44[3]. As mentioned the w_r value is directly related to how many patterns an expert is willing to reject. For high values of w_r each pattern will have high rejection costs and, in consequence, we will eventually have a low number of rejected patterns. To assess the stability of the proposed approaches the experiments were repeated 50 times by averaging the results.

Table 6.1 and Table 6.2 illustrate the implications of an incorrect choice of the w_r value. As an example, in Table 6.2 for the MLP-2C classifier (the same argument applies for the SOM-C2) we can have three times more patterns rejected with subtle improvements on the performance when selecting $w_r = 0.24$ instead of $w_r = 0.44$.



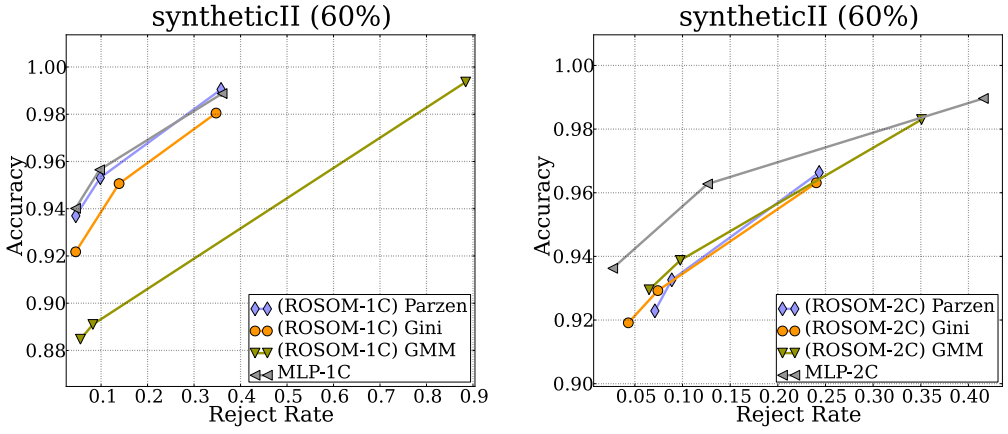
(a) (b)

Figure 6.4: The A-R curves for the `SyntheticI` dataset using 60% of training data.

By analyzing Table 6.1 and Table 6.2 we note that it is difficult to identify the overall gain of the proposed methods in comparison with the MLP-based classifiers. What follows next is a set of figures that allow a better understanding of the performances through the Accuracy-Reject (A-R) curve, whose major advantage resides on the straightforward interpretation of

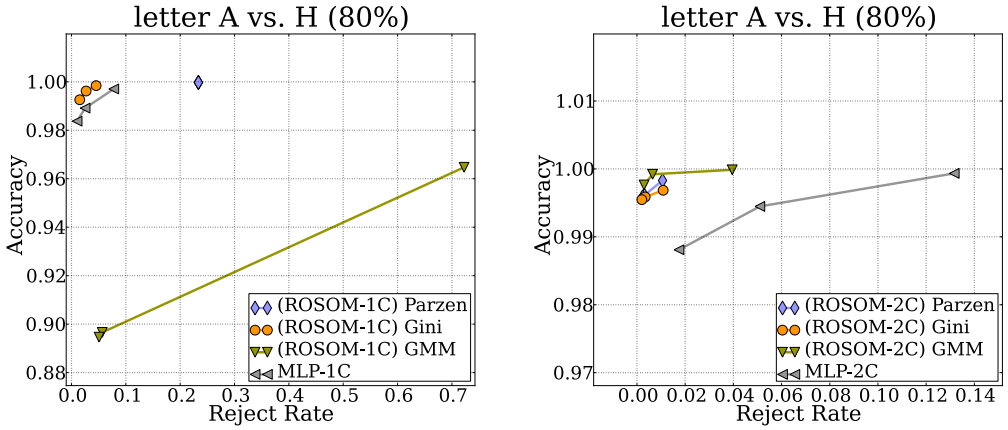
[3]Values of w_r higher than 0.5 are equivalent to random guesses.

the results over the rejection costs presented by the A-R curve. In Figure 6.4 to Figure 8.3 we present the experimental results for each of the aforementioned data sets. In each plot the results of the proposed approaches compared to the MLP-based counterparts are presented. Each point break in the curves corresponds to a given w_r value: 0.04, 0.24 and 0.44.



(a) (b)

Figure 6.5: The A-R curves for the `SyntheticII` dataset using 60% of training data.



(a) (b)

Figure 6.6: The A-R curves for the `Letter AH` dataset using 80% of training data.

By analyzing the performance on an A-R curve one can easily read the performance achieved by a given method and how much it was rejected for a given w_r: the highest the curve, the better the performance is.

For example, for the A-R curves shown in Fig. 6.4a, the ROSOM-1C using the Parzen and Gini coefficient approaches achieved the best overall results. Note that for a reject rate of 0.2 (red vertical line) these classifiers achieved accuracies higher than 0.90; in other words, by rejecting 20% of the patterns, the accuracies of these classifiers go higher than 90% for the `SyntheticI` dataset, both performing much better than the MLP-1C classifier. In Fig. 6.4b, we can see that the performances of all ROSOM-2C variants and the MLP-2C were equivalent.

For the `SyntheticII` dataset, the A-R curves in Fig. 6.5a reveal that the ROSOM-1C/Parzen and the MLP-1C performed equivalently, followed closely by the ROSOM-1C/Gini. The A-R curves in Fig. 6.5b show that the best performance was achieved by the MLP-2C, while all the ROSOM-2C variants achieved equivalent performance.

For the `Letter AH` dataset, the A-R curves in Fig. 6.6a reveal that the best performance was achieved by the ROSOM-1C/Gini, followed closely by the MLP-1C. Both classifiers achieve very high accuracy rates, rejecting less than 5% of the patterns. The A-R curves in Fig. 6.6b show that all the ROSOM-2C variants performed better than the MLP-2C.

It is worth mentioning that to verify that the performances of the SOM-based and MLP-based classifiers are equivalent is *not* a bad thing for the SOM-based classifiers. On the contrary, it is a good thing. Let us recall that the SOM is being adapted to work as a supervised classifier, since it is originally an unsupervised learning algorithm. But even so, the proposed SOM-based approaches achieved very competitive results in comparison with the MLP-based approaches.

For all datasets the ROSOM-1C/GMM achieved in average the worst results. However, the ROSOM-2C/GMM achieved competitive results in comparison with the other approaches based on two classifiers. Such behavior can be partly explained by the fact that the proposed modified learning rules in (6.16) and (6.17) provide additional improvement over the raw estimates of the posterior probabilities in the performances of the ROSOM-2C classifier.

As a general conclusion, although neither the Parzen windows nor the Gini coefficient approaches outperformed one another over all datasets, Parzen and Gini attained better performances than the MLP-based counterparts. For instance, on the vertebral column dataset—see Figure 8.7i—, one can achieve a performance of more than 85% rejecting less than 20% for both the ROSOM-1C and ROSOM-2C approaches.

6.6 Discussion

Reject option comprises a set of techniques aiming at improving the classification reliability in decision support systems. However, the problem of classification with a reject option has been tackled only occasionally in machine learning literature, in most cases using supervised learning methods, such as the SVM and MLP classifiers. In this chapter we presented two SOM-based pattern classifiers that incorporate the rejection class option and compared their performances with MLP-based counterparts. To the best of our knowledge, this is the first time such approach is developed for the self-organizing map or similar neural networks.

The first proposal, called the ROSOM-C1 classifier, requires a single SOM network trained in the usual unsupervised way. The second proposed classifier, called ROSOM-C2 classifier, requires two SOMs which are trained in the self-supervised learning scheme. Both proposals require the estimation of $\mathcal{P}(\boldsymbol{x}|\mathcal{C}_k)$ (or, equivalently, $\mathcal{P}(\mathcal{C}_k|\boldsymbol{x})$) using the distribution of SOM's weight vectors. An optimal threshold value has to be determined in order to re-tag some of the weight vectors with the rejection class label. We have described three techniques to obtaining suitable estimates of the posterior probability $\mathcal{P}(\boldsymbol{x}|\mathcal{C}_k)$ based, namely, on Parzen Window, Gaussian mixture model and Gini coefficient.

For the ROSOM-C2, in particular, the SOM learning rules were modified by the introduction of the rejection cost as a weight. The goal is to train one of the SOMs to become specialized on the class $\mathcal{C}_{-1}$, while the other is trained to become specialized on the class $\mathcal{C}_{+1}$. The decision to accept or reject a given pattern is determined based on the combination of results provided by the outputs of each map.

We carried out a comprehensively evaluation of the performances of the proposed SOM-based classifiers on two synthetic and three real-world data sets. The simulations have indicated that the proposed approaches achieved results that are equivalent to or even better than those obtained by the MLP-based classifiers.

Chapter 7

An Ordinal Data Approach for Detecting Reject Regions*

Having motivated in the previous Chapter the development of classifiers with a third output class, the reject class, in-between the good and bad classes, this particular structure can be further explored. Such can be done through methods presented in the literature for the classification of ordinal data extending them to the reject option paradigm. Therefore, and for completeness, we start by reviewing the data replication method followed by the novel aspects introduced in this Chapter.

7.1 Problem Statement and Standard Solutions

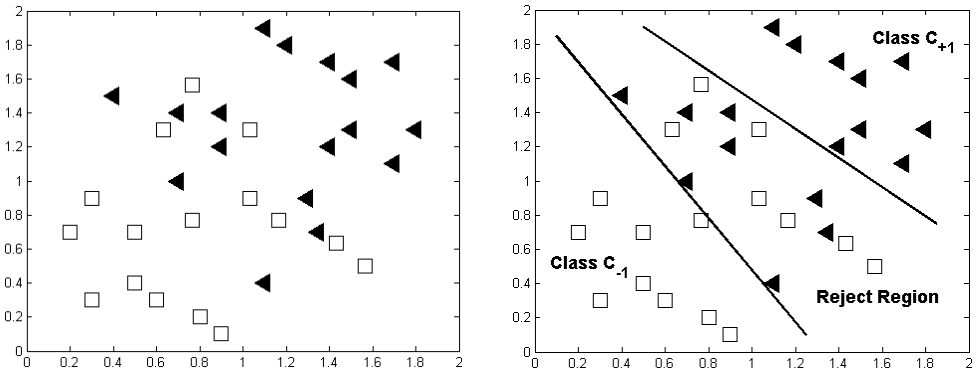
Predictive modeling tries to find good models for predicting the values of one or more variables in a dataset from values of other variables. Our target can assume only two values, represented by 'good' and 'bad' classes. When in possession of a "complex" dataset, a simple separator is bound to misclassify some points. Two types of errors are possible, 'false positives' and 'false negatives'. The construction of a model can be conducted to optimize some adopted measure of business performance, be it profit, loss, volume of acquisitions, market share, etc, by giving appropriate weights to the two types of errors. When the weights of the two types of errors are heavily asymmetric, the boundary between the two classes will be pushed near values where the most costly error seldom happens.

This fact suggests a simple procedure to construct a three-class output classifier: training a first binary classifier with a set of weights heavily penalizing the false negative errors, we expect that when this classifier predicts an item as negative, it will be truly negative. Likewise, training a second binary classifier with a set of weights heavily penalizing the false positive errors, we expect that when this classifier predicts an item as positive, it will be truly positive. When a item is predicted as positive by the first classifier and negative by the second, it will be labeled for review. This setting is illustrated in Figure 7.1.

A problem arises when an item is predicted as negative by the first classifier and positive by the second classifier as in Figure 7.2a. That can happen because the two separator lines intersect each other, generating therefore regions with a *non-logical decision* (regions where individual classifiers are inconsistent, individually deciding for different classes). A convenient workaround is then to avoid this problematic state by imposing that the two boundaries of the classifiers do not intersect, Figure 7.2b.

Before delving into the proposed method, it is worth discussing the simple solution of using a single classifier. If more than just discriminating between the two classes, the model to use yields the posterior probability for each target class, then two cutoffs can be defined

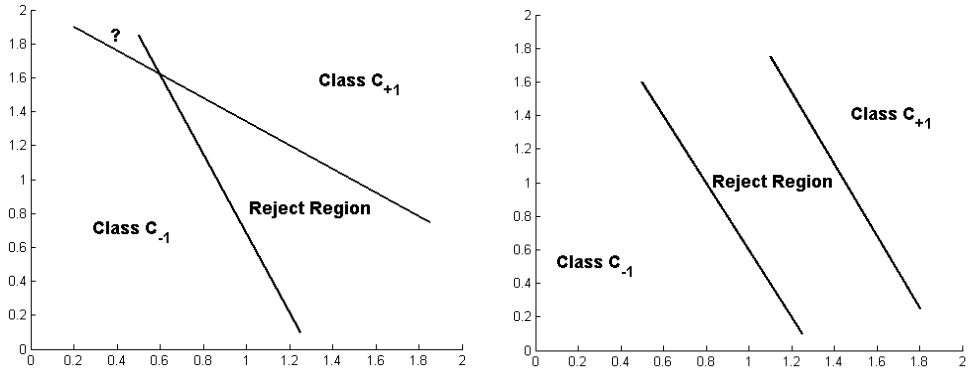
*Some portions of this Chapter appeared in Sousa et al. (2009) and Sousa and Cardoso.



(a) Overlapping regions.

(b) Typical separator lines, obtained with two independent binary classifiers.

Figure 7.1: Illustrative setting with overlapping classes.



(a) Intersecting separating lines.

(b) Non-intersecting separating lines.

Figure 7.2: Potential discriminative boundaries. The advantage of the approach depicted in Figure 7.2b on an ordinal setting has already been stated in Cardoso and da Costa (2007).

on this value. All items with predicted probability of belonging to class $\mathcal{C}_{-1}$ less than a low threshold are labeled as $\mathcal{C}_{+1}$, items with predicted probability of belonging to class $\mathcal{C}_{-1}$ higher than a high threshold are labeled as $\mathcal{C}_{-1}$, items with predicted probability of belonging to class $\mathcal{C}_{-1}$ in-between the low and high threshold are labeled for review. Two issues can be identified with this approach. First, we need to estimate the probability of each class, which is by itself a problem harder than the problem of discriminating classes. Second, the estimation of the two cutoffs is not straightforward nor can be easily fitted into standard frameworks.

The proposed solution is based on the extension of a technique developed for ordinal data, which, for completeness, we start by reviewing this work.

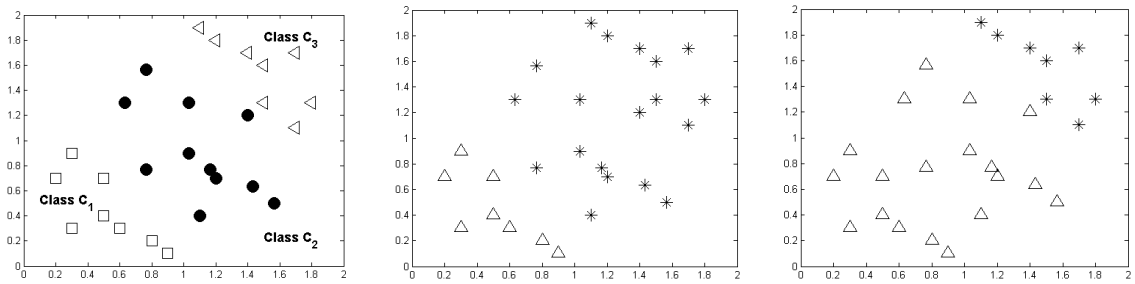
7.2 The Data Replication Method for Ordinal Data

The data replication method for ordinal data can be framed under the SBC reduction, an approach for solving multiclass problems via binary classification relying on a single, standard binary classifier. SBC reductions can be obtained by embedding the original problem in a higher-dimensional space consisting of the original features, as well as one or more other features determined by fixed vectors, designated here as *extension features*. This embedding

is implemented by replicating the training set points so that a copy of the original point is concatenated with each of the extension features vectors. The binary labels of the replicated points are set to maintain a particular structure in the extended space. This construction results in an instance of an artificial binary problem, which is fed to a binary learning algorithm that outputs a single binary classifier. To classify a new point, the point is replicated and extended similarly and the resulting replicas are fed to the binary classifier, which generates a number of signals, one for each replica. The class is determined as a function of these signals (El-Yaniv et al., 2008).

To introduce the data replication method, assume that examples in a classification problem come from one of K ordered classes, labeled from $\mathcal{C}_1$ to $\mathcal{C}_K$, corresponding to their natural order. Consider the training set $\{\mathbf{x}_i^{(k)}\}$, where $k = 1, \ldots, K$ denotes the class number, $i = 1, \ldots, \ell_k$ is the index within each class, and $\mathbf{x}_i^{(k)} \in \mathbb{R}^d$, with d the dimension of the feature space. Let $\ell = \sum_{k=1}^{K} \ell_k$ be the total number of training examples.

Let us consider a very simplified toy example with just three classes, as depicted in Figure 7.3a. Here, the task is to find two parallel hyperplanes, the first one discriminating class $\mathcal{C}_1$ against classes $\{\mathcal{C}_2, \mathcal{C}_3\}$ and the second hyperplane discriminating classes $\{\mathcal{C}_1, \mathcal{C}_2\}$ against class $\mathcal{C}_3$. These hyperplanes will correspond to the solution of two binary classification problems but with the additional constraint of parallelism—see Figure 7.3. The data replication method suggests solving both problems simultaneously in an augmented feature space (Cardoso and da Costa, 2007).



(a) Original dataset in $\mathbb{R}^2$, $K = 3$.

(b) Binary problem $\mathcal{C}_1$ against classes $\{\mathcal{C}_2, \mathcal{C}_3\}$.

(c) Binary problem $\{\mathcal{C}_1, \mathcal{C}_2\}$ against class $\mathcal{C}_3$.

Figure 7.3: Binary problems to be solved simultaneously with the data replication method.

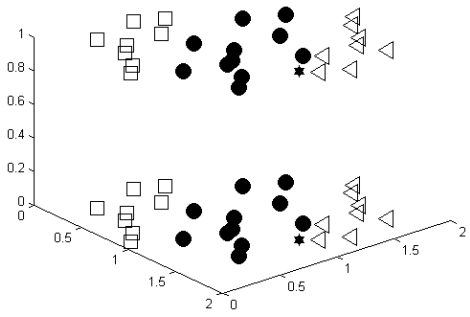
In the toy example, using a transformation from the $\mathbb{R}^2$ initial feature-space to a $\mathbb{R}^3$ feature space, replicate each original point, according to the rule (see Figure 7.4a):

$$\mathbf{x} \in \mathbb{R}^2 \begin{cases} \nearrow \begin{bmatrix} \mathbf{x} \\ h \end{bmatrix} \in \mathbb{R}^3 \\ \searrow \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \in \mathbb{R}^3 \end{cases}, \text{ where } h = \text{const} \in \mathbb{R}^+$$

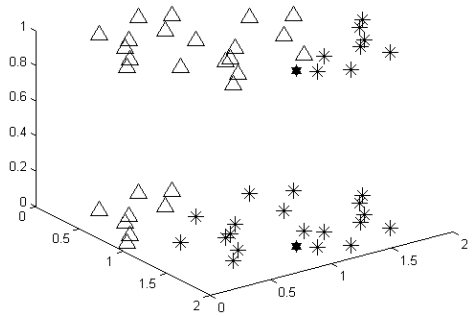
Observe that any two points created from the same original point differ only in the extension feature. Define now a binary training set in the new (higher dimensional) space according to (see Figure 7.4b):

$$\begin{bmatrix} \mathbf{x}_i^{(1)} \\ 0 \end{bmatrix} \in \overline{\mathcal{C}}_1, \ \begin{bmatrix} \mathbf{x}_i^{(2)} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{x}_i^{(3)} \\ 0 \end{bmatrix} \in \overline{\mathcal{C}}_2$$
$$\begin{bmatrix} \mathbf{x}_i^{(1)} \\ h \end{bmatrix}, \begin{bmatrix} \mathbf{x}_i^{(2)} \\ h \end{bmatrix} \in \overline{\mathcal{C}}_1, \ \begin{bmatrix} \mathbf{x}_i^{(3)} \\ h \end{bmatrix} \in \overline{\mathcal{C}}_2 \tag{7.1}$$

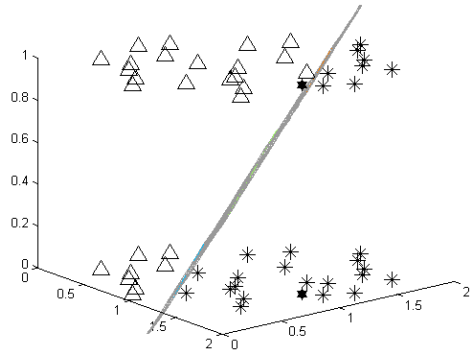
In this step we are defining the two binary problems as a single binary problem in the augmented feature space. A linear two-class classifier can now be applied on the extended dataset, yielding a hyperplane separating the two classes, see Figure 7.4c. The intersection



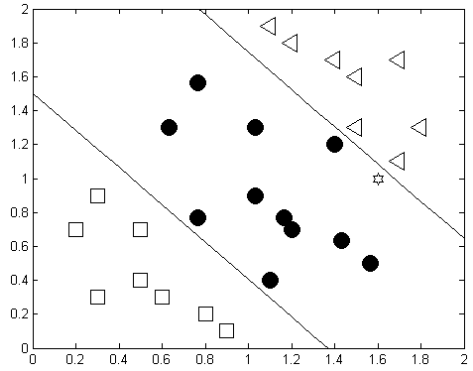
(a) Dataset in $\mathbb{R}^3$, with samples replicated ($h = 1$).



(b) Transformation into a binary classification problem.



(c) Linear solution to the binary problem.



(d) Linear solution in the original dataset.

Figure 7.4: Data replication model in a toy example (from Cardoso and da Costa (2007)).

of this hyperplane with each of the subspace replicas can be used to derive the boundaries in the original dataset, as illustrated in Figure 7.4d.

To predict the class of an unseen example, classify both replicas of the example in the extended dataset with the binary classifier. From the sequence of binary labels one can infer the predicted label on the original ordinal classes

$$\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_1 \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_2 \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_2 \Longrightarrow \mathcal{C}_3$$

Note that only three sequences are possible (Cardoso and da Costa, 2007). The generalization for any problem in $\mathbb{R}^d$, with K ordinal classes and nonlinear boundaries can be found in Cardoso and da Costa (2007).

Summing up, $(K-1)$ replicas in a $\mathbb{R}^{d+K-2}$ dimensional space are used to train a binary classifier. The target class of an unseen example can be obtained by adding one to the number of $\mathcal{C}_2$ labels in the sequence of binary labels resulting from the classification of the $(K-1)$ replicas of the example.

7.3 The Data Replication Method for Detecting Reject Regions

The scenario of designing a classifier with reject option shares many characteristics with the classification of ordinal data. It is also reasonable to assume for the reject option scenario that the three output classes are naturally ordered as $\mathcal{C}_1, \mathcal{C}_{reject}, \mathcal{C}_2$. As the intersection point of the two boundaries would indicate an example with the three classes equally probable—one would be equally uncertain between assigning $\mathcal{C}_1$ or $\mathcal{C}_{reject}$ and between assigning $\mathcal{C}_{reject}$ or $\mathcal{C}_2$—it is plausible to adopt a strategy imposing non-intersecting boundaries. In fact,

as reviewed in Section 7.1, methods have been proposed with exactly such assumption. In the scenario of designing a classifier with reject option, we are interested on finding two boundaries: a boundary discriminating $\mathcal{C}_1$ from $\{\mathcal{C}_{reject}, \mathcal{C}_2\}$ and a boundary discriminating $\{\mathcal{C}_1, \mathcal{C}_{reject}\}$ from $\mathcal{C}_2$.

We proceed exactly as in the data replication method for ordinal data. We start by transforming the data from the initial space to an extended space, replicating the data, according to the rule (see Figure 7.5a and Figure 7.5b):

$$\mathbf{x} \in \mathbb{R}^d \nearrow \begin{bmatrix} \mathbf{x} \\ h \end{bmatrix} \in \mathbb{R}^{d+1} \quad \searrow \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \in \mathbb{R}^{d+1}, \text{ where } h = \text{const} \in \mathbb{R}^+$$

If we design a binary classifier on the extended training data, without further considerations, one would obtain the same classification boundary in both data replicas. Therefore, we modify the misclassification cost of the observations according to the data replica they belong to. In the first replica (the extension feature assumes the value zero), we will discriminate $\mathcal{C}_1$ from $\{\mathcal{C}_{reject}, \mathcal{C}_2\}$; therefore we give higher costs to observations belonging to class $\mathcal{C}_2$ than to observations belonging to class $\mathcal{C}_1$. This will bias the boundary towards the minimization of errors in $\mathcal{C}_2$. In the second replica (the extension feature assumes the value h), we will discriminate $\{\mathcal{C}_1, \mathcal{C}_{reject}\}$ from $\mathcal{C}_2$; therefore we give higher costs to observations belonging to class $\mathcal{C}_1$ than to observations belonging to class $\mathcal{C}_2$. This will bias the boundary towards the minimization of errors in $\mathcal{C}_1$. In Figure 7.5c this procedure is illustrated by filling the marks of the observations with higher costs. Table 7.1 summarizes this procedure.

Replica #	points from $\mathcal{C}_1$	points from $\mathcal{C}_2$
1	$-1; C_\ell$	$+1; C_h$
2	$-1; C_h$	$+1; C_\ell$

Table 7.1: Labels and costs (C_ℓ and C_h represent a low and a high cost value, respectively) for points in different replicas in the extended dataset.

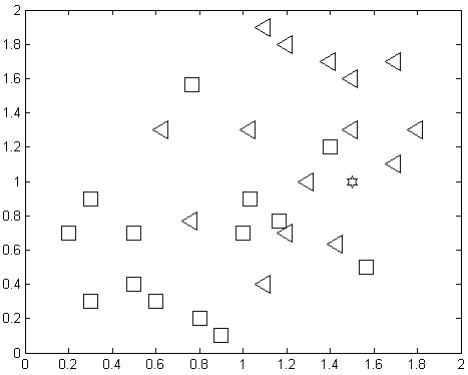
A two-class classifier can now be applied on the extended dataset, yielding a boundary separating the two classes, see Figure 7.5d. The intersection of this boundary with each of the subspace replicas can be used to derive the boundaries in the original dataset, as illustrated in Figure 7.5e.

Summing up, with a proper choice of costs, the data replication method can be used to learn a reject region, defined by two non-intersecting boundaries. Note that the reject region is optimized during training and not heuristically defined afterward. Nonlinear (and non-intersecting) boundaries are treated exactly as the ordinal data scenario. Likewise, prediction follows the same rationale.

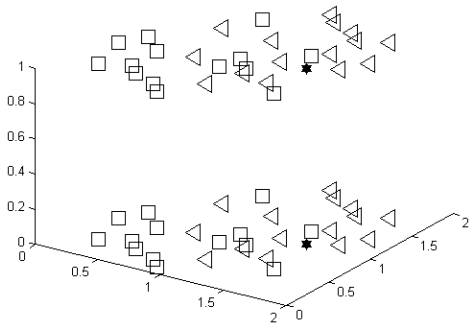
7.3.1 Selecting the Misclassification Costs

In the reject option scheme, one aims to obtain a minimum error while minimizing the number of rejected cases. However, when the number of rejected cases decreases the classification error increases, and to decrease the classification error one typically has to increase the reject region. The right balance between these two conflicting goals depends on the relation of the associated costs.

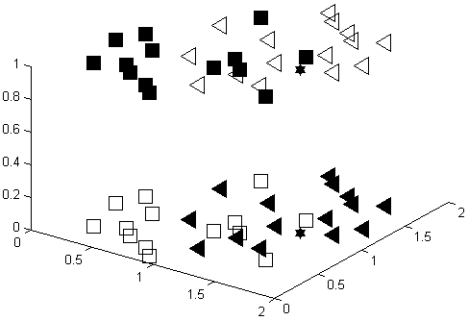
Let $C_{i,q}^{(k)}$ represent the cost of erring a point $\boldsymbol{x}_i$ from class k in data replica q (or, equivalently, by hyperplane q). Points from class $\mathcal{C}_1$ misclassified by the first hyperplane ($\mathbf{w}^t\mathbf{x} + b_1 = 0$) but correctly classified by the second hyperplane ($\mathbf{w}^t\mathbf{x} + b_2 = 0$) incur in a loss $C_{i,1}^{(1)}$; points from class $\mathcal{C}_1$ misclassified by both hyperplanes incur in a loss $C_{i,1}^{(1)} + C_{i,2}^{(1)}$. Likewise, points from class $\mathcal{C}_2$ misclassified by the hyperplane 2 ($\mathbf{w}^t\mathbf{x} + b_2 = 0$) but correctly



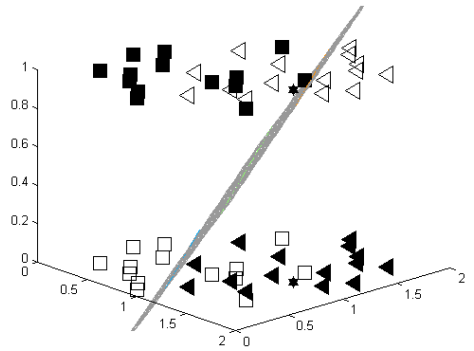
(a) Original binary dataset in $\mathbb{R}^2$.



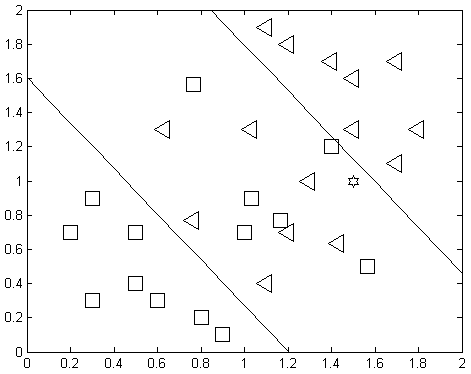
(b) Dataset in $\mathbb{R}^3$, with samples replicated ($h = 1$).



(c) Binary problem in $\mathbb{R}^3$, with filled points representing observations with higher cost of misclassification.



(d) Solution to the binary problem in $\mathbb{R}^3$.



(e) Solution with reject region in the original dataset.

Figure 7.5: Proposed reject option model in a toy example.

classified by the first hyperplane ($\mathbf{w}^t\mathbf{x} + b_1 = 0$) incur in a loss $C^{(2)}_{i,2}$; points from class $\mathcal{C}_2$ misclassified by both hyperplanes incur in a loss $C^{(2)}_{i,1} + C^{(2)}_{i,2}$. The resulting loss matrix is given by

		predicted		
		$\mathcal{C}_1$	$\mathcal{C}_{reject}$	$\mathcal{C}_2$
true	$\mathcal{C}_1$	0	$C^{(1)}_{i,1}$	$C^{(1)}_{i,1} + C^{(1)}_{i,2}$
	$\mathcal{C}_2$	$C^{(2)}_{i,1} + C^{(2)}_{i,2}$	$C^{(2)}_{i,2}$	0

The typical adoption of the same cost for erring and rejecting on the two classes leads to the following simplified loss matrix:

		predicted		
		$\mathcal{C}_1$	$\mathcal{C}_{reject}$	$\mathcal{C}_2$
true	$\mathcal{C}_1$	0	C_l	C_h
	$\mathcal{C}_2$	C_h	C_l	0

Therefore, $C_{reject} = \frac{C_l}{C_h} = w_r$ is the cost of rejecting (normalized by the cost of erring). The data replication method with reject option tries to minimize the empirical risk $w_r R + E$, where R accounts for the rejection rate and E for the misclassification rate.

7.3.2 Prediction

To predict the class of an unseen example, classify both replicas of the example in the extended dataset with the binary classifier. From the sequence of binary labels one can infer the predicted label on the original ordinal classes

$$\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_1 \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_{reject} \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_2 \Longrightarrow \mathcal{C}_2$$

Henceforth, the target class can be obtained by counting the number of $\overline{\mathcal{C}}_2$ labels in the sequence, $N_{\overline{\mathcal{C}}_2}$: if $N_{\overline{\mathcal{C}}_2}/2 + 1$ is integer, it yields the target class; otherwise the option is to reject.

7.4 Mapping the Data Replication Method to Learning Algorithms

In this section the method just introduced is instantiated in two important machine learning algorithms: support vector machines and multilayer perceptrons.

7.4.1 Mapping the Data Replication Method with Reject Option to SVMs

The learning task in a classification problem is to select a prediction function $f(\mathbf{x}, \alpha)$ from a family of possible functions that minimizes the expected *loss*, where α is a parameter denoting a particular function in the set.

The SVM classification technique has been originally derived by applying the Structural Risk Minimization (SRM) principle to a two-class problem using the 0/1 (indicator) loss function:

$$L(\mathbf{x}, \alpha, y) = \begin{cases} 0, & \text{if } f(\mathbf{x}, \alpha) = y \\ 1, & \text{if } f(\mathbf{x}, \alpha) \neq y \end{cases}$$

The simplest generalization of the indicator loss function to classification with reject option is the following loss function

$$L(\mathbf{x}, \alpha, y) = \begin{cases} 0, & \text{if } f(\mathbf{x}, \alpha) = y \\ w_r, & \text{if } f(\mathbf{x}, \alpha) = reject \\ 1, & \text{if } f(\mathbf{x}, \alpha) \neq y \text{ and } f(\mathbf{x}, \alpha) \neq reject \end{cases}$$

where w_r denotes the cost of rejection (with the cost of erring normalized to 1). Obviously $0 \leq w_r \leq 1$. The corresponding expected risk is

$$R = w_r P(reject) + P(error)$$

The expression of the empirical risk (R_{emp}) is

$$R_{\text{emp}} = w_r R + E \tag{7.2}$$

Let us formulate the problem of classifying with reject option in the spirit of SVM. Starting from the generalization of the two-class separating hyperplane presented in the beginning of previous section, let us look for 2 parallel hyperplanes represented by vector $\mathbf{w} \in \mathbb{R}^d$ and scalars b_1, b_2, such that the feature space is divided into 3 regions by the decision boundaries $\mathbf{w}^t\mathbf{x} + b_r = 0,\ r = 1, 2$.

A pair of parallel hyperplanes which minimizes the empirical risk can be obtained by minimizing the following functional (where sgn (x) returns $+1$ if x is greater than zero; 0 if x equals zero; -1 if x is less than zero)

$$\min_{\mathbf{w},b_i,\xi_i} \quad \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_{q=1}^{2}\sum_{k=1}^{2}\sum_{i=1}^{\ell_k} C_{i,q}^{(k)}\ \text{sgn}\ (\xi_{i,q}^{(k)}) \tag{7.3}$$

under the constraints

$$\begin{aligned} -(\mathbf{w}^t\mathbf{x}_i^{(1)} + b_1) &\geq +1 - \xi_{i,1}^{(1)} \\ +(\mathbf{w}^t\mathbf{x}_i^{(2)} + b_1) &\geq +1 - \xi_{i,1}^{(2)} \\ -(\mathbf{w}^t\mathbf{x}_i^{(1)} + b_2) &\geq +1 - \xi_{i,2}^{(1)} \\ +(\mathbf{w}^t\mathbf{x}_i^{(2)} + b_2) &\geq +1 - \xi_{i,2}^{(2)} \\ \xi_{i,q}^{(k)} &\geq 0 \end{aligned}$$

In practice the regularization term sgn $(\xi_{i,q}^{(k)})$ is usually replaced by $\xi_{i,q}^{(k)}$ mainly for computational efficiency.

It is important to note that, although the formulation was constructed from the two-class SVM, it is no longer solvable with the same algorithms. Let us now examine the mapping of the data replication method with reject option on SVMs, which is solvable with a single standard binary SVM classifier.

The rejoSVM The insight gained from studying the toy example paves the way for the formal presentation of the instantiation of the data replication method with reject region in SVMs, rejoSVM.

Following the same procedure delineated in Cardoso and da Costa (2007), it is straightforward to conclude that the formulation corresponding to the mapping of the data replication method with reject option in SVMs results in

$$\min_{\mathbf{w},b_i,\xi_i} \quad \frac{1}{2}\mathbf{w}^t\mathbf{w} + \frac{1}{2}\frac{1}{h^2}(b_2 - b_1)^2 + C\sum_{q=1}^{2}\sum_{k=1}^{2}\sum_{i=1}^{\ell_k} C_{i,q}^{(k)}\ \text{sgn}\ (\xi_{i,q}^{(k)}) \tag{7.4}$$

with $b_2 = b_1 + w_{d+1}h$ and with the same set of constraints as in (7.3).

This formulation for the high-dimensional data set matches the previous formulation (7.3) up to an additional regularization member in the objective function. This additional member is responsible for the unique determination of the thresholds (Cardoso and da Costa, 2007). We see that the rejoSVM captures the essence of the SRM of SVMs, while being solvable with existing binary SVM classifiers.

7.4.2 Mapping the Data Replication Method with Reject Option to Neural Networks

The mapping of the data replication method with reject option to NNs, rejoNN, is easily accomplished with the architecture proposed for ordinal data in Cardoso and da Costa (2007). Non-intersecting boundaries were enforced by making use of a partially linear function $\overline{G}(\overline{\mathbf{x}}) = G(\mathbf{x}) + \underline{\mathbf{w}}^t\mathbf{e}_i$ defined in the extended space. Setting $G(\mathbf{x})$ as the output of a neural network,

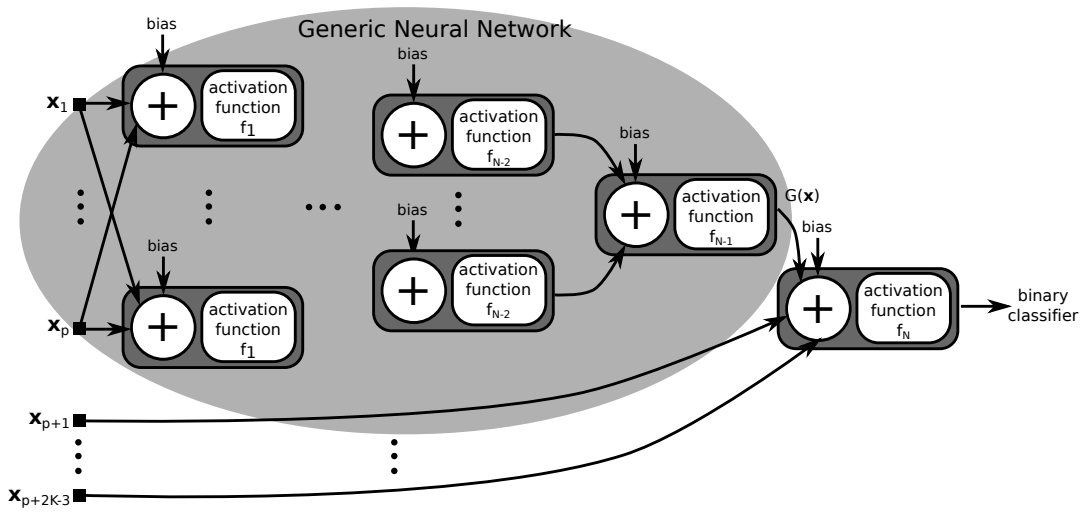


Figure 7.6: Data replication method for neural networks with reject option (adapted from Cardoso and da Costa (2007)).

a flexible architecture for classification with reject option can be devised, as represented diagrammatically in Figure 7.6.

For the mapping of the data replication method with reject option in SVMs and NNs, rejoSVM and rejoNN, if we allow the samples in all the classes to contribute to each threshold, the order inequalities on the thresholds are satisfied automatically, in spite of the fact that such constraints on the thresholds are not explicitly included in the formulation. The proof follows closely the derivation presented in Cardoso and da Costa (2007) for the oNN algorithm.

7.5 Classifying Ordinal Data with Reject Option – a General Framework

Although the reject option is usually only considered on binary data, it makes sense to extend it to multiclass data. In particular, the proposed approach extends nicely to ordinal data. In settings where we have K ordered classes, it may be interesting to define $K-1$ reject regions, between class k and class $k+1$, $k = 1, \ldots, K-1$.

In the standard data replication method for ordinal data, one would have a data replica for each boundary to be defined ($K-1$ data replicas), requiring $K-2$ extension features. Now, as we need to have two boundaries between consecutive classes, we will use $2(K-1)$ data replicas, requiring $2(K-1)-1$ extension features. The goal is to find $2(K-1)$ boundaries $\boldsymbol{w}^t\boldsymbol{x} + b_i$, $i = 1, \ldots, 2(K-1)$, with reject regions defined between boundaries $2j-1$ and $2j$, $j = 1, \ldots, K-1$.

Replicas q and $q+1$, $q = 1, 3, 5, \ldots$ will have exactly the same binary labels, but different costs. Replicas q and $q+1$, $q = 2, 4, 6, \ldots$ will have exactly the same costs, but different binary labels. The boundaries obtained from replicas $2q-1$ and $2q$ will both discriminate $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$. Table 7.2 summarizes this setting.

Similarly to the binary case, the prediction of the target class for an unseen examples uses the sequence of $2(K-1)$ labels $\in \{\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_2\}^{2(K-1)}$ by classifying each of the $2(K-1)$ replicas in the extended dataset with the binary classifier. The target class can be obtained by counting the number of $\overline{\mathcal{C}}_2$ labels in the sequence, $N_{\overline{\mathcal{C}}_2}$: if $N_{\overline{\mathcal{C}}_2}/2 + 1$ is integer, is yields the target class; otherwise the option is to reject.

Replica #	points from $\mathcal{C}_1$	points from $\mathcal{C}_2$	...	points from $\mathcal{C}_{K-1}$	$\mathcal{C}_K$
1	$-1; C_\ell$	$+1; C_h$	$+1; C_h$	$+1; C_h$	$+1; C_h$
2	$-1; C_h$	$+1; C_\ell$	$+1; C_h$	$+1; C_h$	$+1; C_h$
...					
2(K-1)-1	$-1; C_h$	$-1; C_h$	$-1; C_h$	$-1; C_\ell$	$+1; C_h$
2(K-1)	$-1; C_h$	$-1; C_h$	$-1; C_h$	$-1; C_h$	$+1; C_\ell$

Table 7.2: Labels and costs (C_ℓ and C_h represent a low and a high cost value, respectively) for points in different replicas in the extended dataset.

7.6 Two Classifiers Approach for Ordinal Data with Reject Option

In this section, and for experimental comparison purposes, we introduce an extension to ordinal data of the two-classifier approach for binary data with reject option. The extension involves a simple adaptation of the method for ordinal data presented in Frank and Hall (2001). Frank and Hall (2001) proposed to use $(K-1)$ standard binary classifiers to address the K-class ordinal data problem. Toward that end, the training of the i^{th} classifier is performed by converting the ordinal dataset with classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$ into a binary dataset, discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$ (see Figure 7.7). The i^{th} classifier represents

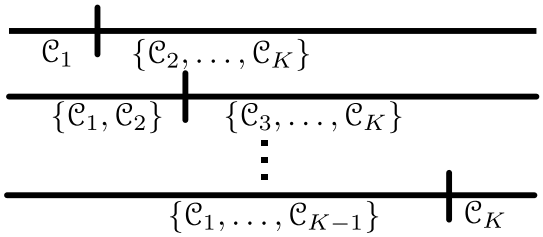


Figure 7.7: Transformation of an ordinal data classification problem in (K-1) binary problems.

the test $\mathcal{C}_x > \mathcal{C}_i$. To predict the class value of an unseen instance, the $K-1$ binary outputs are combined to produce a single estimation. The extension of the two classifiers approach for reject option to ordinal data involves replacing the i^{th} classifier in Frank&Hall method by two classifiers, both discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$ but trained with different costs, exactly as given in Table 7.2 for our proposal. Observe that, under our approach, the $(2i-1)^{th}$ and $(2i)^{th}$ boundaries are also discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$; the major difference lies in the independence of the boundaries found with Frank&Hall's method. This independence is likely to lead to intersecting boundaries.

7.7 Implementation

In the following subsections we will outline three algorithms regarding the reject option approaches identified in Chapter 6, Section 7.1. First we outline in Section 7.7.1 the general setup of the experiments conducted in this work. In Section 7.7.2 and in Section 7.7.3 we present the algorithms for the one and two classifiers approach extended to the multiclass ordinal problem according to the description given in Section 7.6. Finally, in Section 7.7.4 it is presented the algorithm for the method for learning the reject region in an ordinal setting.

7.7.1 Methodology

We randomly split each dataset into training and test sets; in order to study the effect of varying the size of the training set, we considered three possibilities: 5%, 25% and 40% of all the data available. The splitting of the data into training and test sets was repeated 50 times in order to obtain more stable results for accuracy by averaging and also to assess the variability of this measure. The best parametrization of each model was found by 'grid-search', based on a 5-fold cross validation scheme conducted on the training set. Finally, the error of the model was estimated on the test set. The 'grid-search' was performed over the $C = 2^{-5}, \ldots, 2^{3}$ and $\gamma = 2^{-3}, \ldots, 2^{1}$ values when using the RBF kernel for the SVM methods on the `LEV` datasets and polynomial of degree 2 for the synthetic datasets. For the neural network techniques, we performed a 'grid-search' over the number of neurons (5 to 25) with one-hidden layer. Regarding specifically to rejoNN, we also had to tune the h and s parameters. The range of tested values were 1, 1.5 and 2 for h, and 2 and 4 for s in the binary datasets. We fixed the values for $h = 10$ and $s = 3$ in the ordinal datasets. To train the networks on all methods we used the resilient back-propagation algorithm available in MATLAB ^{TM}. For the binary datasets the number of epochs for all methods was set to be 15 whereas for the ordinal datasets we had to tune the best number without degrading the overall results. rejoNN and remaining MLP techniques were trained with at most 100 epochs. The rationale behind the low number of epochs is that it served as an early stopping criterion to attain better generalization results. We have also used a network with K outputs, one corresponding to each class, and target values of 1 for the correct class and 0 otherwise.

7.7.2 Design of Two *Independent* Classifiers

One of the standard procedures identified in Section 7.1 to define the reject region is through the design of independent classifiers. This approach can be straightforwardly extended to the ordinal problems and is described in Algorithm 2. We first train a first classifier with a set of weights heavily penalizing the false negative errors in order to obtain truly negative predictions; and, train a second classifier with a set of weights heavily penalizing the false positive errors in order to obtain truly positive predictions—see Table 7.2 (here the replicas correspond to the different discriminants). In the end, we will have two classifiers, each one specialized to a given class.

7.7.3 Design of a Single Classifier

The algorithm structure for learning the reject region with a single classifier is described in Algorithm 3. First we train a model and the reject region is determined only *after*. If the classifier provides some approximation to the posterior class probabilities, then a pattern is rejected if the maximum of the two posterior probabilities is lower than a given threshold. Otherwise, it is used a rejection threshold targeted to a particular classifier.

7.7.4 Design of rejoSVM

To learn the reject option based on the data replication method proposed in Cardoso and da Costa (2007), we have to modify the misclassification costs of the observations according to the data replica they belong to. Such is performed according Table 7.2 as already mentioned in Section 7.3. This can be easily done by adjusting the C tradeoff with the misclassification costs as represented in Equation (7.4).

For the neural network approach, rejoNN, we changed the error function, $e_k(n)$, where we modify the misclassification costs according to the data replica as before. Formally,

$$e_k(n) = (d_k(n) - y_k(n))C_n \tag{7.5}$$

Algorithm 2: Algorithm structure for the two classifiers approach.

Data: $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}, \mathcal{D}^* = \{\mathcal{X}^*, \mathcal{Y}^*\}$ the training and testing datasets, respectively ($\mathcal{D}, \mathcal{D}^*$ are disjoin datasets).
Result: $\mathcal{Y}^*_{w_r}$ testing set prediction $\forall w_r \in]0, \ldots, 0.5[$.

1 **forall** $w_r \in]0, \ldots, 0.5[$ **do**
2 **forall** *possible combinations of model parameters,* p_i **do**
3 Split $\mathcal{D}$ in 5 equal partitions, $\mathcal{D}^{(v)} = \{\mathcal{X}^{(v)}, \mathcal{Y}^{(v)}\}, v = \{1, \ldots, 5\}$, such that $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(5)}$ are disjoin sets;
4 **foreach** $v = 1$ **to** 5 **do**
5 **foreach** $k = 1$ **to** $K - 1$ **do**
6 Costs $\leftarrow$ set costs according Table 7.2 ;
7 $\mathcal{Y}_o = \begin{cases} -1, & y \leq k \\ +1, & y > k \end{cases}, \quad \forall y \in \mathcal{Y}^{(1,\ldots,5)\setminus v}$;
8 $\mathcal{M}_{2k-1} \leftarrow$ `Train_Model` $(\mathcal{X}, \mathcal{Y}_o$,Costs);
9 $\mathcal{M}_{2k} \leftarrow$ `Train_Model` $(\mathcal{X}, \mathcal{Y}_o$,Costs);
10 validate $\mathcal{M}_1 \cup \ldots \cup \mathcal{M}_{2(K-1)}$ performance according Equation (7.2) given $\mathcal{D}^v$;
11 save the parametrization resulting of the best mean validation performance;
12 train the 2(K-1) models, $\mathcal{M}_k$, with dataset $\mathcal{D}$ according lines 2–10;
13 **forall** *models* $\mathcal{M}_k, k = \{1, \ldots, 2(K-1)\}$ **do**
 `/* predict and change negative responses to zero */`
14 $\mathcal{Y}_k \leftarrow$ `Test_Model` $(\mathcal{X}^*, \mathcal{M}_k)$;
15 $\mathcal{Y}^*_{w_r} = \begin{cases} 1 + \left(\sum_{k=1}^{2(K-1)} \mathcal{Y}_k\right)/2, & \mod \left(\sum_{k=1}^{2(K-1)} \mathcal{Y}_k, 2\right) = 0 \\ \text{Reject}, & \text{otherwise} \end{cases}$

Algorithm 3: Algorithm structure for the one classifier approach.

Data: $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ the training dataset and $\mathcal{X}^*$ the testing set.
Result: $\mathcal{Y}^*_{w_r}$ testing set prediction $\forall w_r \in]0, \ldots, 0.5[$.

`/* train model according a standard 5 fold cross-validation procedure to find best model parametrization */`
1 $\mathcal{M} \leftarrow$ `Train_Model`$(\mathcal{X}, \mathcal{Y})$;
2 Obtain the posterior probabilities $(\mathcal{P}_1, \ldots, \mathcal{P}_K)$ of $\mathcal{X}$ given model $\mathcal{M}$;
3 **forall** $w_r \in]0, \ldots, 0.5[$ **do**
4 obtain BestThreshold $\in [0.5, \ldots, 1]$, that minimizes Equation (7.2) given $\mathcal{D}$ and $\mathcal{P}$;
5 $(\mathcal{Y}_{pred}, \mathcal{P}_{max}) \leftarrow$ `Test_Model` $(\mathcal{X}^*, \mathcal{M})$, where $\mathcal{P}_{max} = \max(\mathcal{P}_1, \ldots, \mathcal{P}_K)$;
6 $\mathcal{Y}^*_{w_r} = \begin{cases} \text{Reject}, & \mathcal{P}_{max} < \text{BestThreshold} \\ \mathcal{Y}_{pred}, & \text{otherwise} \end{cases}$

where $d_k(n)$ is the response given by output neuron k for the input pattern n and $y_k(n)$ the desire response (true label). C_n corresponds to a given $C^{(k)}_{i,q}$ from Equation (7.4) represented here for syntax simplicity.

The algorithm structure for learning the reject region as proposed in here is described in Algorithm 4. Function `Train_Model` in line 4 of Algorithm 4 can be a single binary classifier according Equation (7.4) in the case of a binary SVM. The formulation for the multiclass case can be found in Cardoso and da Costa (2007) subject to the costs present in Table 7.2.

Algorithm 4: Algorithm structure for the rejoSVM classifier approach.

Data: $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ the training dataset and $\mathcal{X}^*$ the testing set composed by N instances.

Result: $\mathcal{Y}^*_{w_r}$ testing set prediction $\forall w_r \in]0, \ldots, 0.5[$.

forall $w_r \in]0, \ldots, 0.5[$ **do**

  /* C^{rep} is all the $C^{(k)}_{i,j}$ as represented in Table 7.2 and in Equation (7.4) */

  $(\mathcal{X}^{rep}, \mathcal{Y}^{rep}, C^{rep}) \leftarrow$ replicate dataset $\mathcal{D}$ according Table 7.2;

  $(\mathcal{X}^{*rep}) \leftarrow$ replicate dataset $\mathcal{D}^*$;

  /* Optimize function from Equation (7.4) or the NN represented in Figure 7.6 */

  $\mathcal{M} \leftarrow$ Train_Model$(\mathcal{X}^{rep}, \mathcal{Y}^{rep}, C^{rep})$;

  $\mathcal{Y}_1 \leftarrow$ Test_Model $(\mathcal{X}^{*rep}, \mathcal{M})$;

  /* convert $\mathcal{Y}_1$ replicas prediction to a single K class */

  $Y^{*(j)}_{w_r} \leftarrow 1 + \sum_{i=1}^{p+K-2} y_1^{(i)}, \quad \forall j = 1, \ldots, N, \quad y_1 \in \mathcal{Y}_1$;

7.8 Experimental Study

In the following subsections, experimental results are provided for several models based on SVMs and NNs, when applied to diverse data sets, ranging from synthetic to real data, for binary and ordinal data. The set of models under comparison include the proposed rejoSVM and rejoNN methods, the "one classifier" approach and "two classifiers" approach (SVM and MLP, hereafter referred to SVM-1C and SVM-2C, and MLP-1C and MLP-2C respectivelly), and Fumera and Roli (2002) method.

The major reason for comparing our proposal (rejoSVM, rejoNN) against Fumera and Roli (2002) resides on the most fundamental principles which both methods share. The minimization of the empirical risk with the optimum reject rule proposed by Chow (1970) as succinctly presented in Chapter 6, represents the same basis for both methods. However, and to the best of our knowledge, the most recent works do not explore this concept and hence a fair comparison would not be possible.

SVM-1C, SVM-2C, MLP-1C and MLP-2C are naïve reject option learning schemes as referred in Section 7.1. The SVM-1C was also used in Fumera and Roli (2002) as baseline. As a remark, the SVM-2C and MLP-2C approaches are formed by 2(K-1) classifiers.

The work was performed in a reproducible research manner, and the MATLAB ™code needed to reproduce all reported results is available at `http://www.inescporto.pt/~rsousa/software/`[2]. The proposed rejoSVM is based on the binary SVM from the Bioinformatics Toolbox and the rejoNN is based on the Neural Network Toolbox. We thank G. Fumera for providing the source code (in C/C++) of his method. Note that this method is for SVMs only and the provided implementation works only with linear kernels.

7.8.1 Multiclass data

To evaluate the generalization of our approach, we tailored the `syntheticI` dataset into another different dataset, `syntheticIII`, generated similarly as `syntheticI` (see Chapter 6).

$$(b_{0.5}, b_1, b_{1.5}, b_2, b_{2.5}, b_3, b_{3.5}, b_4, b_{4.5}, b_5) = (-\infty; -1.5; -1.25; -1; -0.5; -0.1; 0.1, 0.5; 1.1; +\infty)$$

[2] Page under construction.

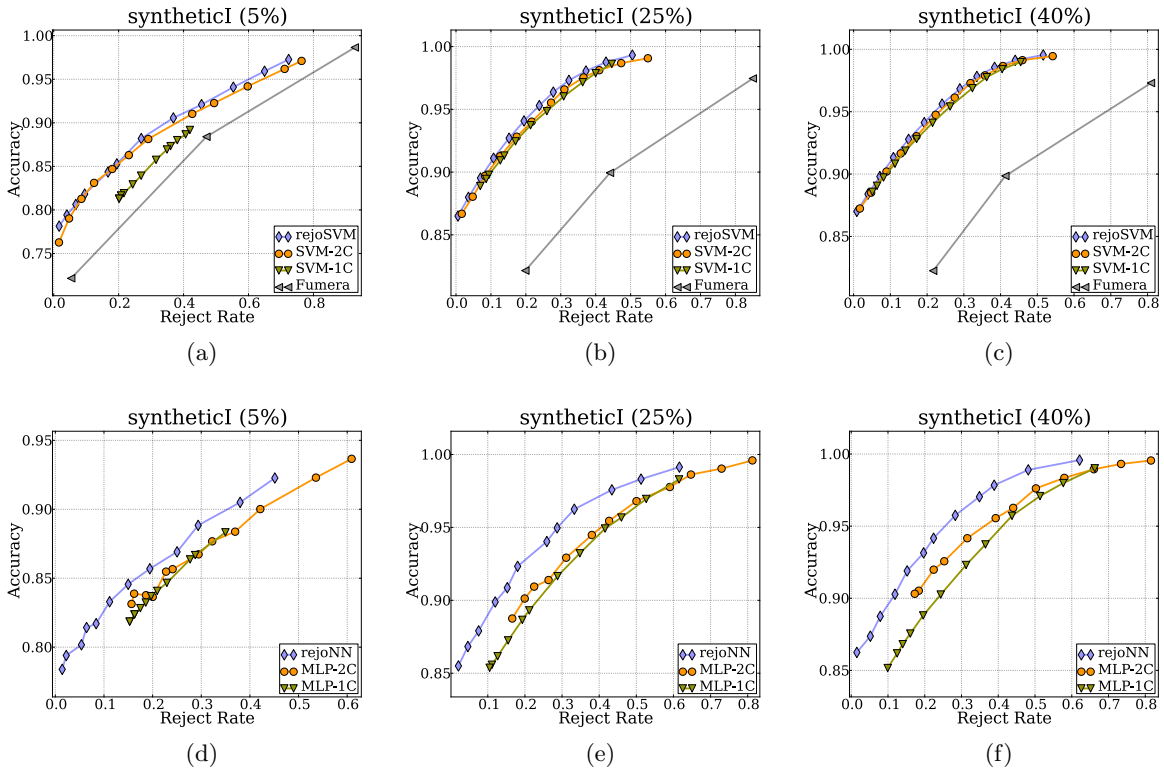


Figure 7.8: The A-R curves for the `syntheticI` dataset. (a)–(c): SVM methods only; (d)–(f): NN methods only. 5%, 25% and 40% of training data, respectively.

Another dataset named `syntheticIV` was used in our experiments. This dataset is an extension of the `syntheticII` with one additional class generated accordingly to the Gaussian distribution with mean $[7\ 7]^t$ and covariance $\Sigma = 4\mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

7.8.2 Results

The performance of a classifier with reject option can be represented by the classification accuracy achieved for any value of the reject rate (the so-called Accuracy-Reject curve (AR curve)). The trade-off between errors and rejections depends on the cost of a rejection w_r. This implies that different points of the AR curve correspond to different values of w_r. We considered values of w_r less than 0.5, as above this value it is preferable to just try to guess randomly (Chow, 1970). In some cases, only three values of w_r were used due to computational issues.

Figure 7.8 to Figure 7.13 summarize the results obtained for all datasets. A first main assertion is that in overall rejoSVM and rejoNN performed better than any of the other methods under comparison, over the full range of values for w_r, specially, on the binary datasets. Moreover, since that in Fumera method only linear kernels were implemented, we extended the datasets with second order terms $x_i x_j$ when evaluating this method. In this extended space, the optimal solutions for the synthetic datasets are indeed linear. On the ordinal datasets, rejoSVM and rejoNN achieved competitive results with standard procedures.

With the increase of the training dataset size, as expected, we see that all methods do not perform each other. A major conclusion based on this empirical analysis is that rejoSVM performs well with few training instances. Nonetheless, this can cause some irregularities on the curves, specially on NN, as can be depicted in Figure 7.8d and Figure 7.9d.

It is also observable that, in general, SVM based methods outperform the neural network

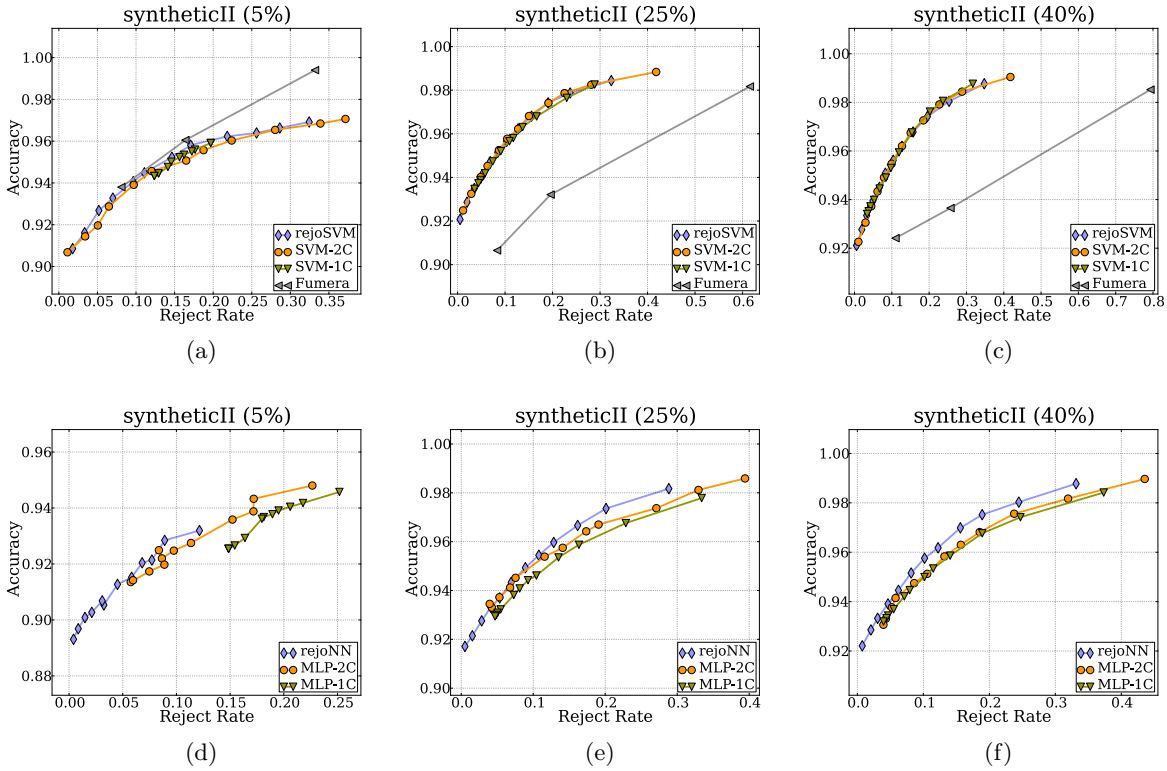


Figure 7.9: The A-R curves for the `syntheticII` dataset. (a)–(c): SVM methods only; (d)–(f) NN methods only. 5%, 25% and 40% of training data, respectively.

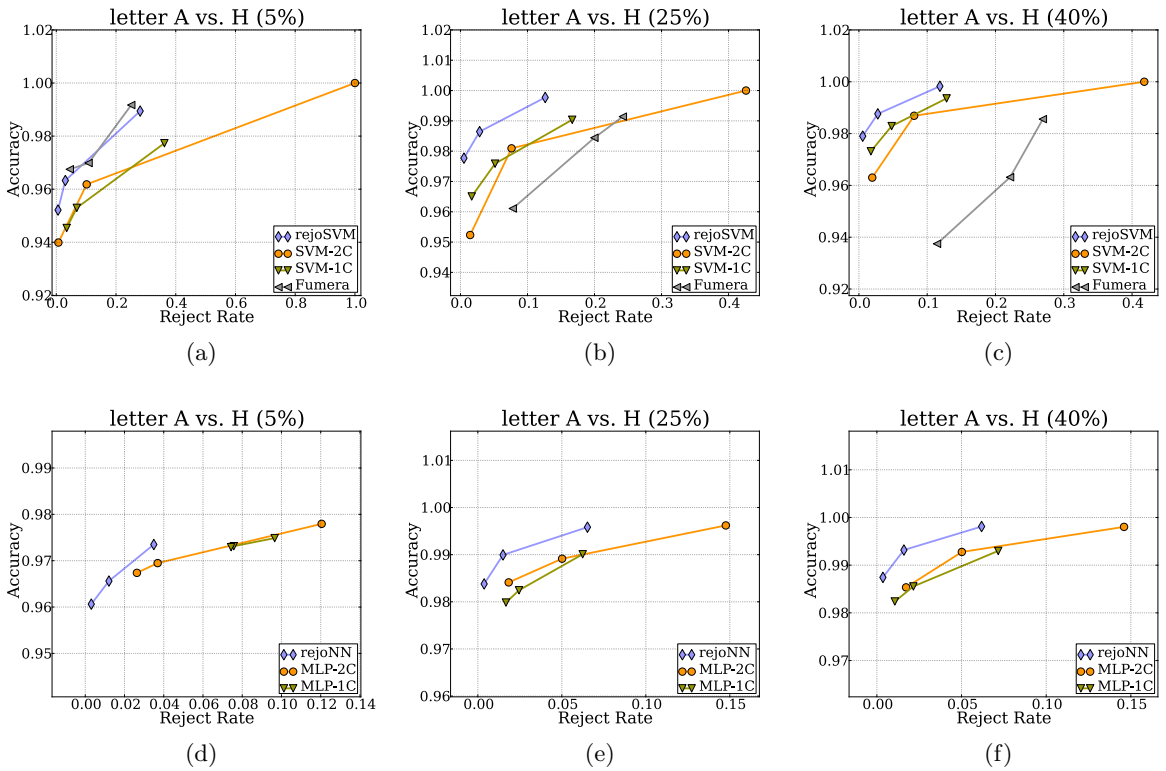


Figure 7.10: The A-R curves for the `letter AH` dataset. (a)–(c): SVM methods only; (d)–(f) NN methods only. 5%, 25% and 40% of training data, respectively.

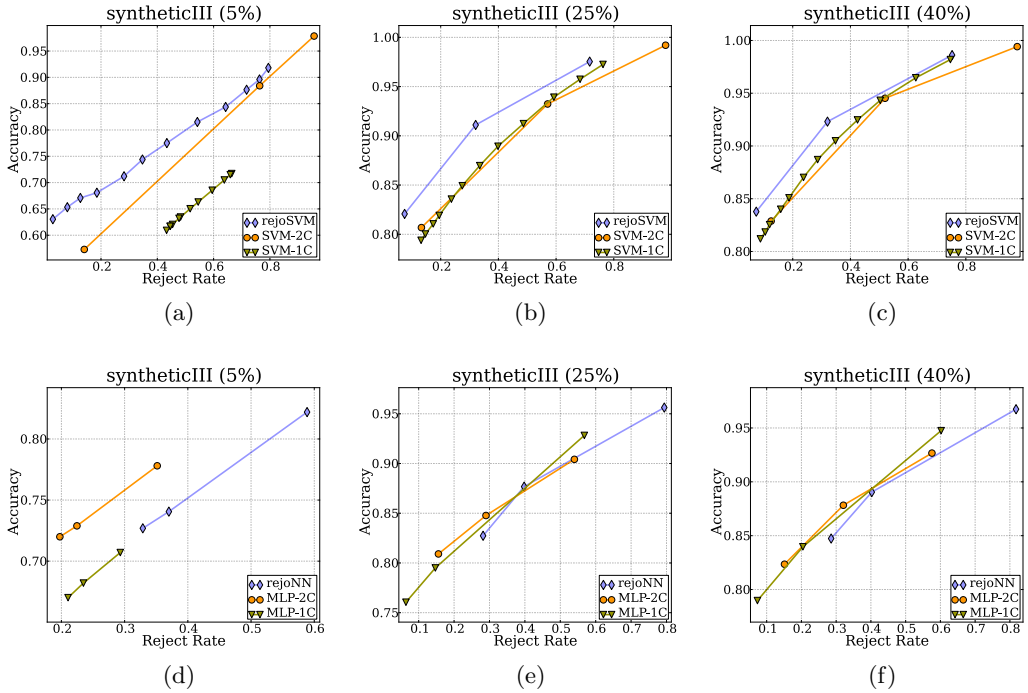


Figure 7.11: The A-R curves for the `syntheticIII` dataset. (a)–(c): SVM methods only. (d)–(f): NN methods only. 5%, 25% and 40% of training data, respectively.

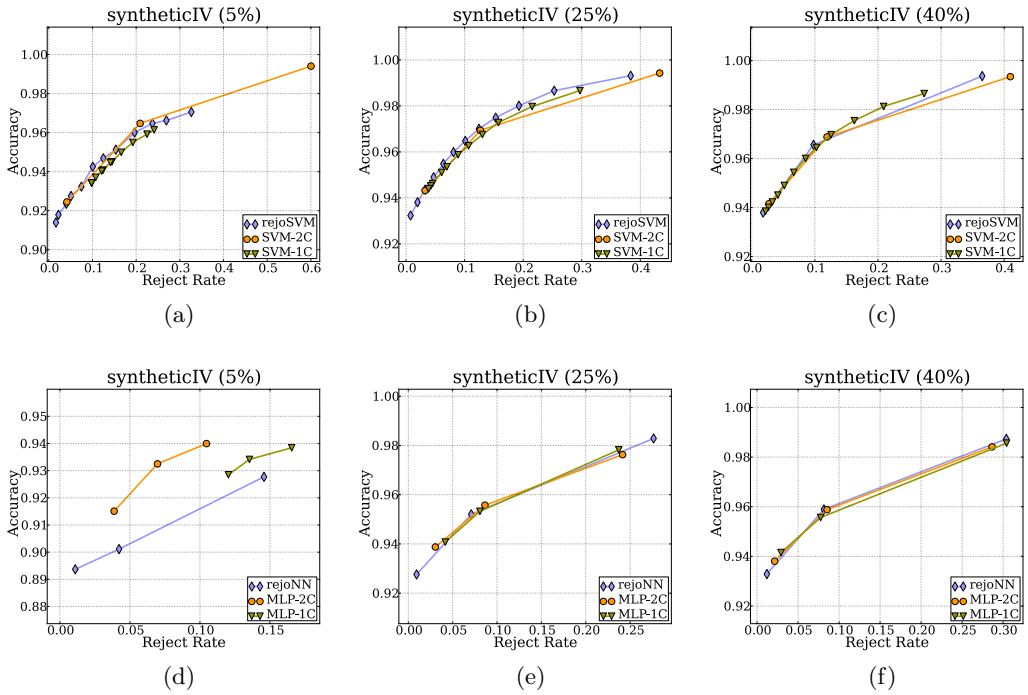


Figure 7.12: The A-R curves for the `syntheticIV` dataset. (a)–(c): SVM methods only. (d)–(f): NN methods only. 5%, 25% and 40% of training data, respectively.

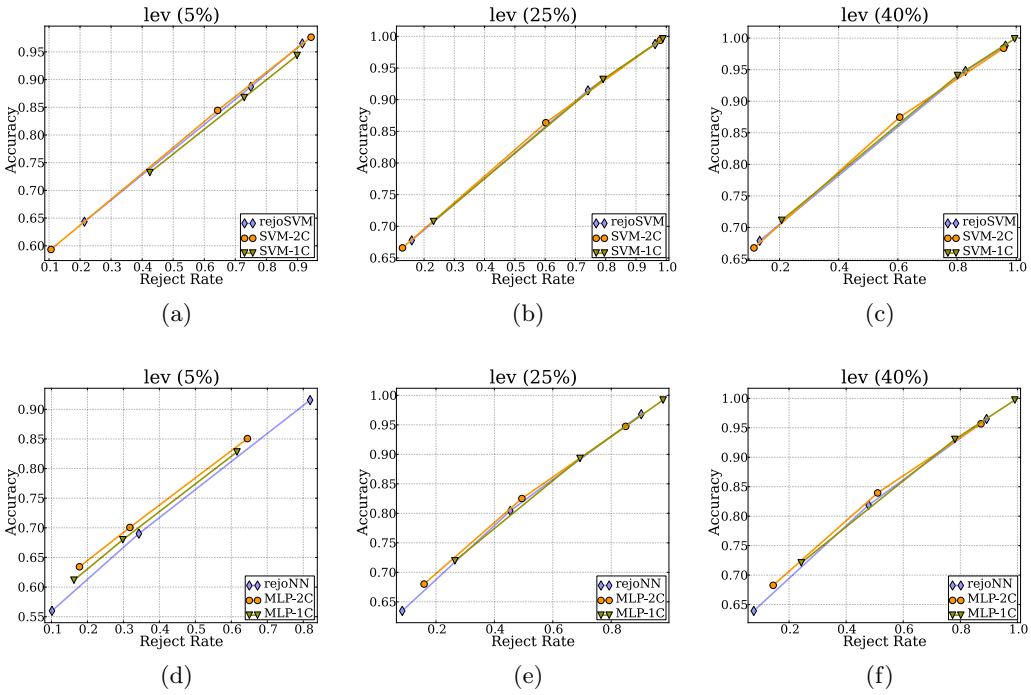


Figure 7.13: The A-R curves for the `LEV` dataset. (a)–(c): SVM methods only. (d)–(f): NN methods only. 5%, 25% and 40% of training data, respectively.

counterparts, in line with the current view in the research community. When restricting the attention to neural network methods, the proposed rejoNN exhibits often the best performance. Moreover, it is important to emphasize that rejoSVM and rejoNN approaches have the advantage of simplicity, using a single direction for all boundaries, and interpretability. The insight of looking to the reject option problem as an ordinal class setting can promote new lines of research.

Finally, we highlight that the proposed framework: 1) has the capability to detect reject regions with a single standard binary classifier; 2) does not required the addition of any confidence level, or thresholds, to define the trust regions; and 3) does not generate ambiguity regions as the "two classifiers" approach, as it was presented in Figure 7.2a.

7.9 Discussion

Despite the myriad of techniques that handle the incorporation of a reject option in their approaches, many of them do not fully account the pioneer work of Chow (1970). In this chapter, we proposed an extension of the data replication method (Cardoso and da Costa, 2007) that directly embeds reject option. This extension was derived by taking a new perspective of the classification with reject option problem, viewing the three output classes as naturally ordered. A pair of non-intersecting boundaries delimits the rejection region provided by our model. Our proposal has the advantages of using a standard binary classifier and embedding the design of the reject region during the training process. Moreover, the method allows a flexible definition of the position and orientation of the boundaries, which can change for different values of the cost of rejections w_r. This method was mapped into NN and SVM with very positive results.

Part IV

Multicriteria Learning on Medical Applications

Chapter 8

Applications*

This Chapter will be dedicated to the assessment of the several learning models presented in this thesis into two medical applications: Breast Cancer Conservative Treatment and Diagnosis of Pathologies in the Vertebral Column. We will start by firstly describing both problems so that afterwards we can conduct a thoroughly assessment of the methods presented along this thesis. Considering the advantages already outlined for each method, a discussion will be provided in the end towards their incorporation on Computer Aided Diagnosis (CAD) systems.

8.1 Breast Cancer Conservative Treatment (BCCT)

One of the first problems that we will consider is regarding breast cancer. As one might know, breast cancer treatments have evolved in the last decades where the use of breast conservative techniques to treat early breast cancer cases have considerably increased. These techniques have a major advantage over mastectomy in the preservation of the breast with equivalent oncological results. Nonetheless, the non-existence of standard methods as for instance quantity of tissue to be excised around the tumor and the type of incision contribute to different final aesthetical results.

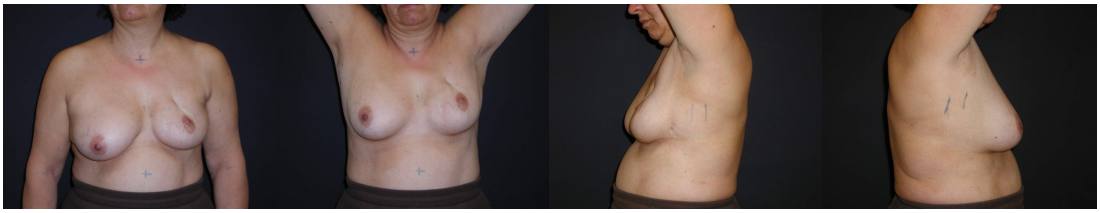
Traditionally Harris et al. (1979), Beadle et al. (1984) and Pierquin et al. (1991), cosmetic assessment has been subjectively performed by a group of observers. However, this evaluation procedure is poorly accurate since it depends highly on the experience of the observers because different and complementary variables and estimations are combined synergistically in assigning an evaluation score. Moreover, human group behavior have shown that will exist a predominant individual that will try to make others to agree with him and by that influencing the evaluation. Consequently, the inherent subjective decisions that are associated to every human will result in a questionable evaluation. By that, this form of assessment is poorly reproducible.

Objective methods of evaluation have emerged to overcome the reproducibility which exist in the subjective assessment and consisted in measurements taken from patients or from photographs, being essentially based on asymmetries between treated and non-treated breast.

A dataset for this problem was constructed containing 150 patients and was divided in two sets: 120 from different institutions from Portugal and 30 from two different European institutions. Breast images were obtained employing a 4Megapixel digital camera where patients were photographed in four positions: facing, arms down; facing, arms up; operated side, arms up; contra-lateral side, arms up (see Figure 8.1). A mark was made on the skin at the suprasternal notch and at the mid-line 25 cm below the first mark. These two marks

*Some portions of this Chapter appeared in Sousa (2008) and Neto et al. (2011).

create a correspondence between pixels measured on the digital photograph and the length in centimeters on the patient.



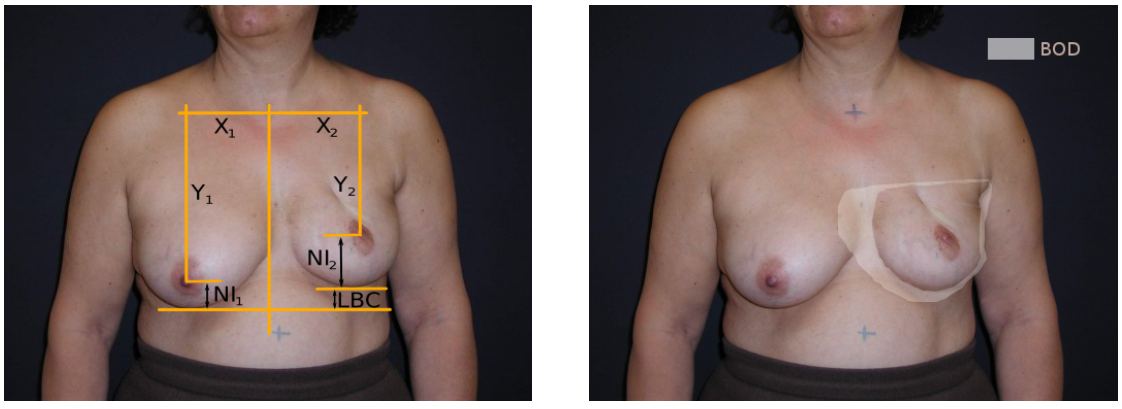
(a) (b) (c) (d)

Figure 8.1: Positions used in the photographs: (a) Facing, arms down; (b) Facing, arms up; (c) Operated side, arms up; and, (d) Contra-lateral side, arms up.

To define a method which could be reproducible, making use of objective measures, a set of patients with known overall classification was required. Collecting this type of evaluation from different areas of the world would provide the desired reference classification since ideally the overall aesthetic assessment should correlate coherently with experts' assessment. The evaluation was done according Harris scale divided in four levels: *excellent* (treated breast nearly identical to untreated breast), *good* (treated breast slightly different than untreated), *fair* (treated breast clearly different from untreated but not seriously distorted) and *poor* (treated breast seriously distorted).

In order to obtain a consensus among the observers, the Delphi process Jones and Hunter (1995) and Hasson et al. (2000) was used. Experts are recruited individually and anonymously. Being the survey conducted over several rounds where the results are next analyzed and then reported to the group, this process is only completed when there is a convergence of opinion or when a point of diminishing returns is reached.

In the evaluation of the aesthetical result of breast cancer conservative treatment, an observer identifies and evaluates color, shape, geometry, irregularity and roughness of the visual appearance of the treated breast. These identified characteristics can be described by the following three major features: breast asymmetry, color difference and scar visibility which can be seen on the whole in Figure 8.2.



(a) (b)

Figure 8.2: Assessment used measures: (a) Reference points and some measures; and, (b) Breast Overlap difference.

For more details concerning the description of each feature, the reader should consult (Cardoso and Cardoso, 2007; Sousa, 2008).

8.1.1 Results

First, we applied the corresponding learning methods to the ordinal data problem BCCT starting by the Unimodal All-at-Once approach presented in Chapter 4. A far more clear

Method	standard I	standard II	unimodal I	unimodal II
MER	0.47 (0.02)	0.47 (0.02)	0.47 (0.02)	0.47 (0.02)
OCI	0.41 (0.41)	0.41 (0.39)	**0.25 (0.33)**	0.39 (0.42)
R_s	0.33 (0.04)	0.33 (0.04)	**0.25 (0.23)**	0.27 (0.23)
τ_b	0.30 (0.04)	0.30 (0.04)	**0.24 (0.19)**	0.23 (0.25)

(a) mean (std. dev.) for each method, BCCT dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$.

Method	standard I	standard II	unimodal I	unimodal II
MER	0.20 (0.04)	0.20 (0.04)	0.20 (0.04)	0.20 (0.04)
OCI	0.26 (0.30)	0.23 (0.30)	**0.16 (0.29)**	0.20 (0.30)
R_s	**0.85 (0.04)**	0.85 (0.05)	0.84 (0.04)	0.84 (0.04)
τ_b	0.81 (0.04)	0.81 (0.05)	**0.82 (0.04)**	**0.82 (0.04)**

(b) mean (std. dev.) for each method, BCCT dataset, $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma|\boldsymbol{x} - \boldsymbol{y}|^2)$ with selected features ρLBC, ρBCE, $cEMD_a$ and $s\chi^2 Lab_{3D}$.

Table 8.1: Unimodal results for BCCT dataset.

difference is presented on the results for the BCCT dataset with feature selection. In Cardoso and Cardoso (2007) the authors performed a FS technique in order to select the best sub-set of features. Based on that study a selection of the same features (ρLBC, ρBCE, $cEMD_a$ and $s\chi^2 Lab_{3D}$) was performed and then the proposed classifiers were evaluated. Results are presented in Table 8.1b. Comparing the results without FS (Table 8.1a) and with FS (Table 8.1b) one can assess the improvement not only on the overall performance of all the classifiers but also on these approaches. Even though these methods do not outperform the standard All-at-Once techniques on this particular dataset, they attain similar results.

BCCT dataset was also used to assess the performance of the Global Constraints presented in Chapter 5. Conclusions drawn in Chapter 5 can be applied as well to BCCT dataset. DT

Model	Datasets
	BCCT
cTree	0.45 (0.04)
oTree	0.42 (0.05)
kNN	0.53 (0.04)
okNN	0.54 (0.02)

Table 8.2: Mean (standard deviation) of MER over 50 setups of the datasets.

attained a better result than the k-NN and the proposed improvement on the latter did not aid to attain better performance. Unfortunately it was not possible to assess the improved version of global constraints framework on BCCT dataset due to computational issues.

Finally, remains the analysis of these methodologies in the reject option setting. Once again, each point break in each curve correspond to a given w_r value: 0.4, 0.24 and 0.44. Each value corresponds to the cost of rejecting and can be defined as how willing one is to reject a portion of the dataset. First, and for the experimental work with the binary models, the multiclass problem was transformed into a binary one, by aggregating *Excellent* and *Good* in one class, and the *Fair* and *Poor* cases in another class. The A-R curves in Fig. 8.4c reveal that the best performance was achieved by the ROSOM-1C/Parzen. For a small range of reject rate values (around 0.3) the performances of the ROSOM-1C/Parzen and the ROSOM-1C/Gini overlap. The A-R curves in Fig. 8.4f show that all ROSOM-2C variants and the

Figure 8.3: The A-R curves for the BCCT dataset using 80% of training data.

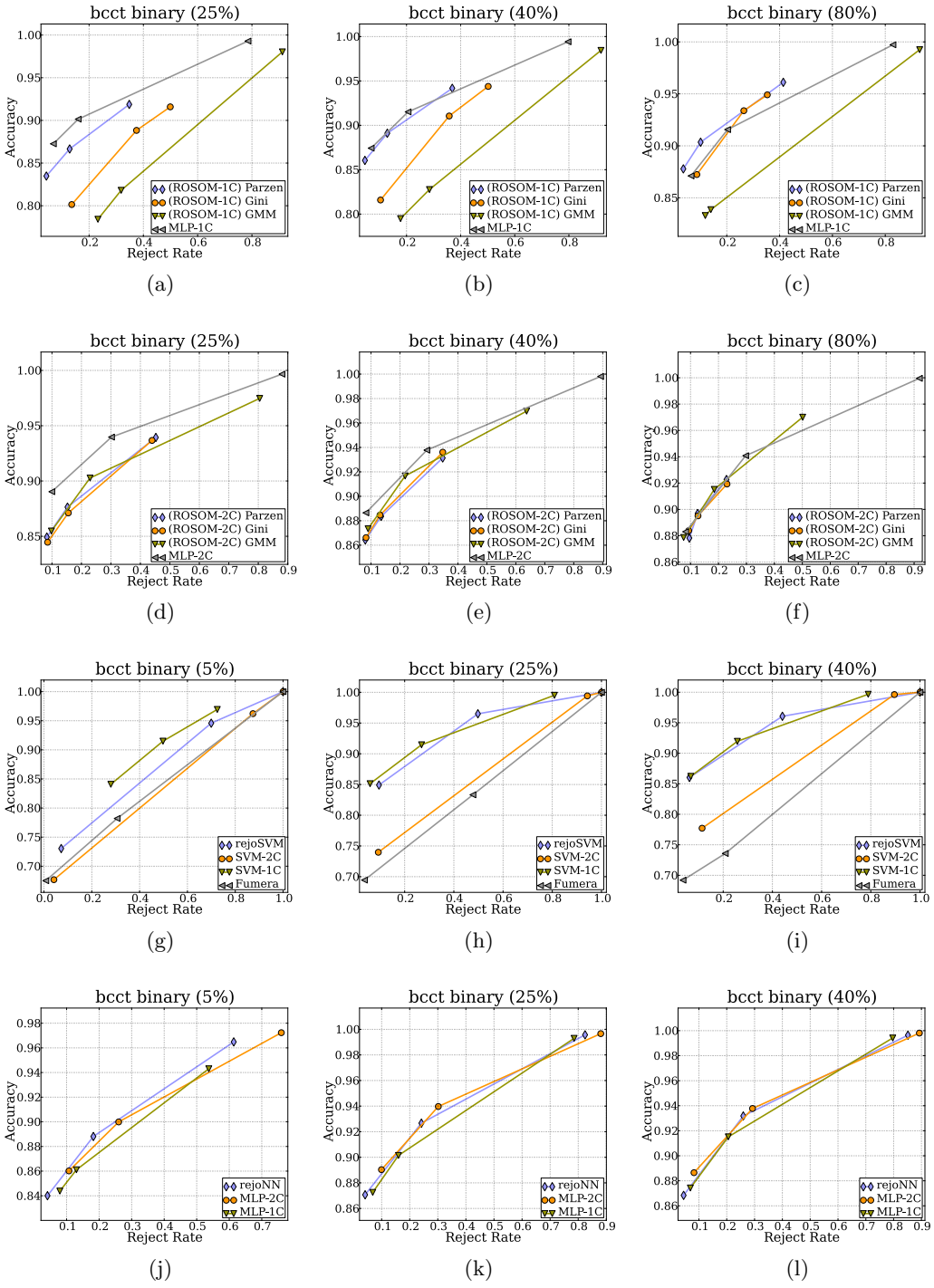


Figure 8.4: The A-R curves for the binary BCCT dataset. Figure 8.4a–Figure 8.4c: SOM methods with one classifier. Figure 8.4d–Figure 8.4f: SOM methods with two classifiers. 25%, 40% and 80% of training data, respectively. (g)–(i): SVM methods only; (j)–(l): NN methods only. 5%, 25% and 40% of training data, respectively.

MLP-2C performed equivalently. Regarding rejoSVM and rejoNN methodologies, in the same scenario as before, one can attain an accuracy on the order of more than 85%—see Figure 8.4h and Figure 8.4k. On the full BCCT class set depicted in Figure 8.5a through Figure 8.5c,

despite all methods performing increasingly better with an increasing training dataset size, "one classifier" approach attains the best results.

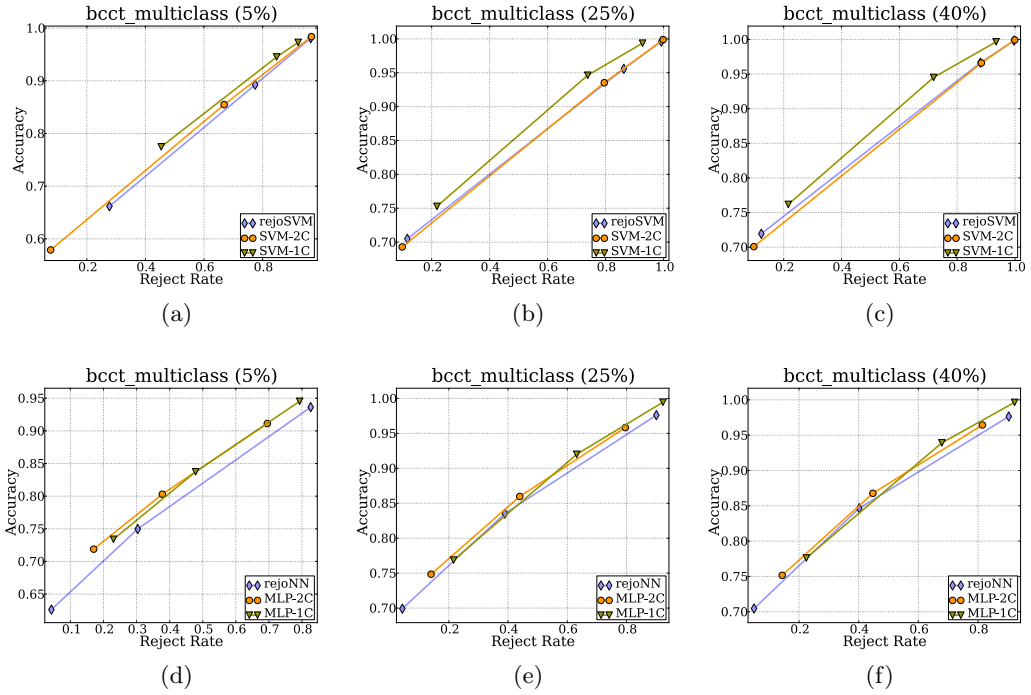


Figure 8.5: The A-R curves for the multiclass BCCT dataset. (a)–(c): SVM methods only. (d)–(f): NN methods only. 5%, 25% and 40% of training data, respectively.

8.2 Intelligent System for Diagnosis of Pathologies in the Spine (SINPATCO)[2]

The second problem that we will describe is concerned to the diagnosis of pathologies on the vertebral column. Of particular interest is that, in general, the application of machine learning techniques in Traumatic Orthopedics is rather sparse in the literature. This fact is due to the absence of numerical attributes that quantitatively describe the pathologies of interest to the field of orthopedics, to generate a suitable database for the design of classifiers (Neto and Barreto, 2009).

8.2.1 Pathologies of the Vertebral Column

The vertebral column is a system composed by a group of vertebrae, invertebrate discs, nerves, muscles, medulla and joints. The main functions of the vertebral column are as follow: (i) human body support axle; (ii) osseous protector of the spine medulla and nervous roots; and (iii) body's movement axles, making movement possible in three levels: frontal, sagittal and transversal.

This complex system can suffer dysfunctions that cause backaches with very different intensities. Disc hernia and spondylolisthesis are examples of pathologies of the vertebral column that cause intense pain. They result of small or several traumas in the column that gradually injures the structure of the inter-vertebral disc.

Disc hernia appears when the core of the inter-vertebral disc migrates from its place (from the center to the periphery of the disc). Once heading towards the medullary channel

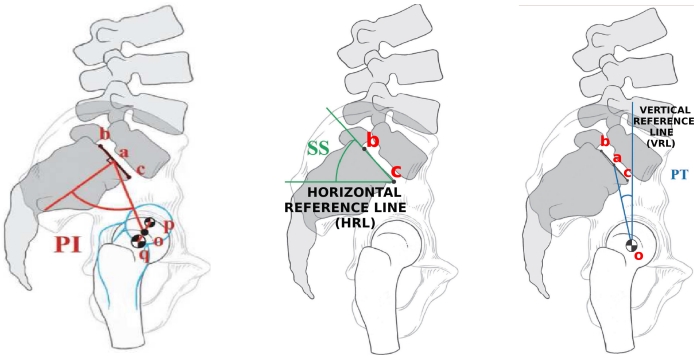
[2] *Sistema Inteligente para Diagnóstico de Patologias da Coluna Vertebral*

or to the spaces where the nervous roots lie, this leads inevitability to their compression. Spondylolisthesis occurs when one of the 33 vertebrae of the vertebral column slips in relation to the others. This slipping occurs generally towards the base of the spine in the lumbar region, causing pain or symptomatology irritation of the nervous roots. In the following section we will briefly describe characteristics (attributes) that are used to quantitatively describe each patient.

8.2.2 Biomechanical Attributes

The database applied in this work was kindly supplied by Dr. Henrique da Mota, who collected it during a medical residence in spine surgery at the *Centre Médico-Chirurgical de Réadaptation des Massues*, placed in Lyon, France. This database contains data about 310 patients obtained from sagittal panoramic radiographies of the spine. From this, 100 patients are volunteers that do not have any pathology in their spines (normal patients). The remaining data are from the patients operated due to disc hernia (60 patients) or spondylolisthesis (150 patients). Therefore, the database is composed of 210 abnormal patients.

Each patient in this database is represented as a vector (or pattern) with six biomechanical attributes, which correspond to the following parameters of the spino-pelvic system: angle of pelvic incidence, angle of pelvic tilt, lordosis angle, sacral slope, pelvic radius and grade of slipping. The correlation between the vertebral column pathologies and this attributes was originally proposed in reference (Berthonnaud et al., 2005).



(a) (b) (c)

Figure 8.6: Spino-pelvic system.

Pelvic incidence (PI) is defined as an angle subtended by line $\overline{oa}$, which is drawn from the center of the femoral head to the midpoint of the sacral endplate and a line perpendicular to the center of the sacral endplate in Figure 8.6a. The sacral endplate is defined by the line segment $\overline{bc}$ constructed between the posterior superior corner of the sacrum and the anterior tip of the $S1$ endplate at the sacral promontory. For the case when the femoral heads are not superimposed, the center of each femoral head is marked, and a connecting line segment will connect the centers of the femoral heads. Pelvic radius (RP) $\overline{ao}$ will be drawn from the center of this line to the center of the sacral endplate (Figure 8.6a).

Lordosis angle is the bigger sagittal angle between the sacrum superior plate and the lumbar vertebra superior plate or thoracic limit. Sacral Slope (SS) is defined as the angle between the sacral endplate ($\overline{bc}$) and the horizontal reference line (HRL), in Figure 8.6b, while Pelvic Tilt (PT) is defined as the angle between the vertical reference line (VRL) and the line joining the middle of the sacral endplate and the axis of the femoral heads in Figure 8.6c. It is positive when the hip axis lies in front of the middle of the sacral endplate. Finally, the level of slipping is the percentage level of slipping between the inferior plate of the fifth lumbar vertebra and the sacrum.

The occurrence of pathologies in the vertebral column is conditioned to the morphological types of the pelvis-spine system. The pelvic incidence, being in an elevated level, is conditioned to a higher sacral slope, that generates increasing shear by the increase of the support plan inclination for lumbar lordosis, besides facilitating the conflict of posterior structures, leading to the appearing of a fracture of fatigue in the arc that supports the vertebra and generating a slope called Spondylitics. The low pelvic incidences lead to the contrary effect, with the occurrence of an increasing pressure in the intervertebral disc and facilitate the occurrence of degeneration and disc hernias. The incidence angle determines a normal condition.

The design of automatic classifiers based in biomechanical attributes of real clinical cases allows that linear and/or non-linear relations, as well as their influences in the diagnostic, are captured in a transparent way for the orthopedist, in a way to help him in the decision making.

8.2.3 Results

We will now conduct an assessment of the reject approaches presented in this thesis applied to the `SPINE`[3] dataset. Within the SINPATCO context, the incorporation of a reject option can be an asset. Moreover, tools like SINPATCO are designed as decision aiding system which could be used on healthcare offices located on remote areas with limited access to modern resources and funding. In this way, systems with high rates of True Positive (sensitivity) and True Negatives (specificity) are required. Such techniques besides imposing high accuracies rates and a higher confidence on the diagnosis, they also avoid misclassifications. In doing so, there will not be any influence by SINPATCO on the expert to take wrong decisions which could lead to some interventions (being invasive or not). As a final remark, we could verify that rejoSVM and two classifiers do not outperform the other. However, and as a feature of this work, rejoSVM benefits of simplicity and interpretability which could aid the medical expert in future evaluations. Regard SOM methodologies, we can see an upwards trend in almost all of them when applied to `SPINE` datasets depicted in Figure 8.7 for the *ROSOM-1C* and *ROSOM-2C*. The A-R curves in Fig. 8.7i indicate that the ROSOM-1C/Gini achieved the best overall performance. The A-R curves in Fig. 8.7l show that all the ROSOM-2C variants performed better than the MLP-2C.

[3]SPINE dataset is available online at the Machine Learning UCI repository.

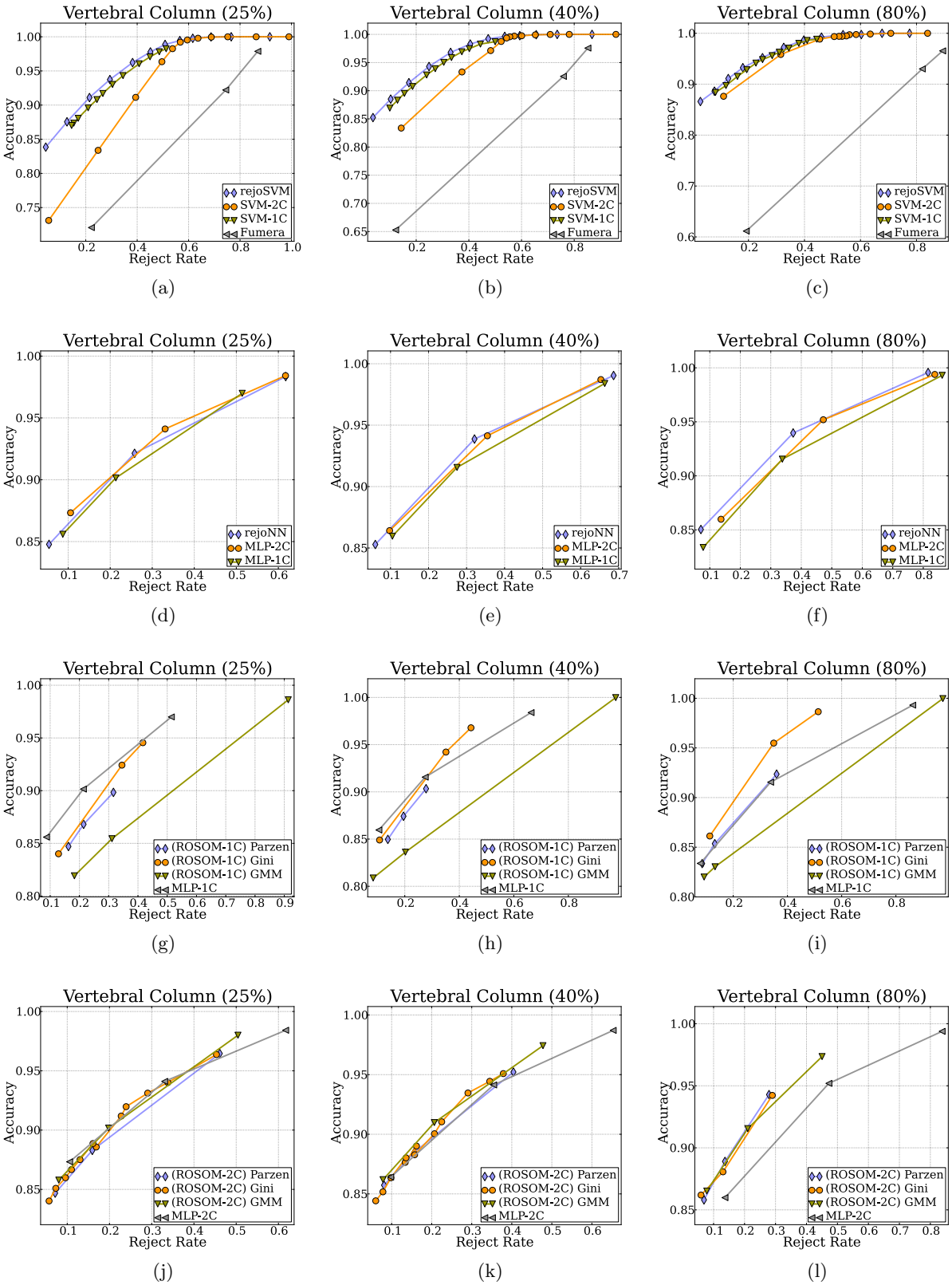


Figure 8.7: The A-R curves for the `Vertebral Column` dataset using 80% of training data.

Part V

Conclusion and Future Work

Chapter 9

Conclusion

MC models for ordinal data has increasingly captured research community attention where its properties led to very interesting and appealing niches in the broader ML field. One of the main reasons for this trend is mostly due to the capability to deduce simple and straightforward classification strategies. Nevertheless, a gap related to some classical methods still exist and was tackled in this thesis.

One of the first issues concerns to the quality assessment for ordinal classifiers which had not been thoroughly delved. The usual metrics, MER, MSE or MAE, to name a few, although very popular, are not appropriate for ordinal data problems. Hence, the need for robust metrics urges as these classification schemes are becoming intrinsic to the decision support field. In this thesis we have proposed a new metric for assessing ordinal classifiers performance. Being defined directly on the CM to evaluate the performance in ordinal data classification, this metric chooses the non-discordant pairs of observations that minimize the cost of a global optimization procedure on the CM, minimizing deviation of the pairs to the main diagonal while maximizing the benefit. The adoption of this measure thus guarantees fair comparison among competing systems, and more correct optimization procedures for classifiers.

It was also identified that despite the multitude of approaches already tackling the ordinal data problem, some still suffer from some issues: e.g., do not incorporate totally the order. By extending the unimodal paradigm for SVMs where one assumes that the a posteriori probabilities of the K classes should follow an unimodal distribution, we were able to create a learning model capable to take into account the order relationship. Afterwards, we have also considered the unimodal paradigm in the design of a new k-NN and DT methods where ordinal data learning algorithms seems even scarcer. To do so, we have first introduced a new concept of ordinality where the order is not captured directly in the input space, but in an implicit feature space. Secondly, we have delved a new method which instantiates this new reasoning for ordinality through global constraints. Such leads us to the fundamental idea that adjacent decision regions should have equal or consecutive labels. Finally, taking advantage of ordinal data setting, it was possible to extend the ordinal data learning paradigms to the reject option problem.

Future Work

Theoretical conclusions and experimental results presented in this thesis can be extended in several different directions. It is, however, important to state first the following. Notwithstanding the well defined results with important implications, the definition possible future lines of research can encourage others to use and explore the analysis conducted in this document. For this reason, the next paragraphs will succinctly refer some of the likely upswing that can be attained.

Regarding to the proposed metric, we have argued that it should not be seen only as a tool for comparing but also to design better classifiers. It can be done on two different settings. A first use is 'externally' to the classifier, using the metric to select the best parametrization of the classifier; and, a second possibility is to embed the new metric in the classifier design. In the latter, an adaptation is conducted in the internal objective function of the classifier, replacing loss functions based on standard measures by a loss function based on the proposed measure. For instance, the standard loss function of a MLP based on the square of the error or on cross-entropy could be replaced by an error evaluated by OCI which may be pursued in future research.

The Unimodal All-at-Once methodology proposed can be improved by using different strategies. Crammer and Singer (2002) suggested an iterative optimization technique since the computation of the full problem is highly computationally expensive. This scheme decomposes the problem into sub-problems having therefore the major advantage of being capable to compute for larger datasets. Also, a comparison with Tsochantaridis et al. (2004) approach which uses a similar technique as Crammer and Singer (2002), among others, can be performed.

Concerning the global constraints approach, some extensions may encompass the adaptation of the pruning or splitting strategies of tree models. Dyadic trees (Scott and Nowak, 2006) may provide an adequate environment to research some of the previous topics. In fact, although the proposed consistency underlying principle has been applied as a pre- and post-processing of the result of a standard method, nothing prevents its application during the design of the decision model. The connection established with the unimodal model may provide some suggestions in that direction. Finally, further studies may be taken in order to reduce the number of variables and constrains towards complexity diminution.

Regarding the reject option paradigm, the overall good results achieved in our experiments with SOMs express promissory future results for the development of an embedded SOM reject option method. The design of such algorithm would thereby allow to capture automatically the reject region during the training phase. In doing so, such would allow a direct comparison against rejoSVM and rejoNN methodologies which were also proposed in this thesis.

Appendix A

Measures for Ordinal Data

A.1 Triangular inequality

For sufficiently high values of β ($\beta \geq \frac{1}{N+1}$) the optimal path is always over the main diagonal and the OCI simplifies to $1-\frac{\sum_{(r,c)\in\text{main diagonal}} n_{r,c}}{N+\left(\sum_{\forall(r,c)} n_{r,c}|r-c|^{\gamma}\right)^{1/\gamma}} = \frac{M+H}{M+N} = \frac{M}{M+N}+\frac{H}{M+N}$, where H and M are the Hamming and Minkowski distances, respectively. This is easily seen to be a metric:

- the positive definiteness and symmetry have already been established in the main body of the article;
- Knowing that if d_1 and d_2 are metrics and $d_1(\boldsymbol{a},\boldsymbol{b}) \leq d_2(\boldsymbol{a},\boldsymbol{b}),\ \forall \boldsymbol{a},\boldsymbol{b}$, then
  1. $\frac{d_2}{1+d_2}$ is a metric;
  2. $\frac{d_1}{1+d_2} \leq \frac{d_2}{1+d_2}$ is a metric;
  3. $d_1 + d_2$ is a metric;

It just lacks to prove that for $\beta \geq 1/(N+1)$ the optimal path is indeed the main diagonal. Let p be a consistent path and b_1 be the part of benefit of the path on the main diagonal and $b_2 > 0$ the part of benefit of the path not in the main diagonal. If $\beta \geq \frac{1}{N+1}$ then the following is true for the cost C of the path:

$C = 1-\frac{b_1+b_2}{N+M}+\beta\sum_{(r,c)\in\text{path}} n_{r,c}|r-c|^{\gamma} \geq 1-\frac{b_1}{N+M}-\frac{b_2}{N+M}+\frac{1}{N+1}\sum_{(r,c)\in\text{path}} n_{r,c}|r-c|^{\gamma} \geq 1-\frac{b_1}{N+M}-\frac{b_2}{N+M}+\frac{b_2}{N+1} \geq 1-\frac{b_1}{N+M}$.

This last value is clearly not inferior to the cost of the path over the main diagonal.

To finalize, it is easy to conclude that for small values of β, OC_{β}^{γ} is not a metric. Consider the vectors (K=2)

$$\boldsymbol{a} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ 2 \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \boldsymbol{c} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 2 \\ 1 \end{bmatrix}.$$

The corresponding confusion matrices are

$$CM(\boldsymbol{a},\boldsymbol{b}) = \begin{bmatrix} N-1 & 0 \\ 1 & 0 \end{bmatrix} \quad CM(\boldsymbol{b},\boldsymbol{c}) = \begin{bmatrix} N-1 & 1 \\ 0 & 0 \end{bmatrix} \quad CM(\boldsymbol{a},\boldsymbol{c}) = \begin{bmatrix} N-2 & 1 \\ 1 & 0 \end{bmatrix}$$

It is easy to confirm that for $\beta < \frac{N-1}{(N+1)(N+2)}$ we have $OC_{\beta}^{\gamma}(\boldsymbol{a},\boldsymbol{b}) + OC_{\beta}^{\gamma}(\boldsymbol{b},\boldsymbol{c}) < OC_{\beta}^{\gamma}(\boldsymbol{a},\boldsymbol{c})$ and therefore OC_{β}^{γ} does not obey the triangular inequality.

A.2 Source Code Listing

For reference, it is presented in Listing 1 a Matlab implementation of OC_{β}^{γ}.

```
% input: confusion matrix and number of classes
% size(cMatrix) must be [K K]
function oc=OrdinalClassificationIndex(cMatrix, K)
  N = sum(cMatrix(:));
  ggamma = 1;
  bbeta  = 0.75/(N*(K-1)^ggamma);

  helperM2 = zeros(K,K);
  for r=1:K
    for c=1:K
      helperM2(r,c) = cMatrix(r,c) * ((abs(r-c))^ggamma);
    end
  end
  TotalDispersion=(sum(helperM2(:))^(1/ggamma));
  helperM1        =cMatrix/(TotalDispersion+N);

  errMatrix(1,1) = 1 - helperM1(1,1) + bbeta*helperM2(1,1);
  for r=2:K
    c=1;
    errMatrix(r,c) = errMatrix(r-1, c) - helperM1(r,c) + bbeta*helperM2(r,c);
  end
  for c=2:K
    r=1;
    errMatrix(r,c) = errMatrix(r,c-1) - helperM1(r,c) + bbeta*helperM2(r,c);
  end

  for c=2:K
    for r=2:K
      costup      = errMatrix(r-1, c);
      costleft    = errMatrix(r, c-1);
      lefttopcost = errMatrix(r-1, c-1);
      [aux,idx]   = min([costup costleft lefttopcost]);
      errMatrix(r,c) = aux  - helperM1(r,c) + bbeta*helperM2(r,c);
    end
  end
  oc = errMatrix(end,end);
  return
```

Listing A.1: Ordinal Classification Index computation.

Appendix B

Unimodal

B.1 Unimodal All-at-Once Support Vector Machine

Our first approach consists modeling the All-at-Once technique to the ordinal data problem. This can be done in straightforward manner by adding the following restrictions:[1]

$$\begin{aligned} \boldsymbol{w}_{j+1} g(\boldsymbol{x}_i) + b_{j+1} > \boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j, \quad j = 1, \ldots, y_i - 1 \\ \boldsymbol{w}_j g(\boldsymbol{x}_i) + b_j > \boldsymbol{w}_{j+1}^T g(\boldsymbol{x}_i) + b_{j+1}, \quad j = y_i, \ldots, K - 1 \end{aligned} \tag{B.1}$$

The following Sections will be concerned to the extension of the All-at-Once SVM concept in the basic and sophisticated architectures.

B.1.1 Basic Architecture

Conditions defined in Equation (B.1) define the decision function according the unimodal paradigm. Hence, an unimodal All-at-Once SVM formulation is defined as:

$$\begin{aligned} \min \mathcal{L}(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) = \frac{1}{2} \sum_{j=1}^{K} \|\boldsymbol{w}_j\|^2 + C \sum_{i=1}^{n} \sum_{j=1}^{K-1} \xi_{i,j} \\ s.t. \begin{cases} (\boldsymbol{w}_{j+1} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{j+1} - b_j \geq 1 - \xi_{i,j}, \quad j = 1, \ldots, y_i - 1 \\ (\boldsymbol{w}_j - \boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i) + b_j - b_{j+1} \geq 1 - \xi_{i,j}, \quad j = y_i, \ldots, K - 1 \\ \xi_{i,j} > 0 \end{cases} \end{aligned} \tag{B.2}$$

where $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K)$, $\boldsymbol{b} = (b_1, \ldots, b_K)$ and $\boldsymbol{\xi} = (\xi_{1,1}, \ldots, \xi_{1,K-1}, \xi_{2,1}, \ldots, \xi_{N,K-1})$ with $\xi_{i,j} > 0$.

To solve this optimization problem the nonnegative Lagrange multipliers are introduced:

[1] For simplicity of notation, a pattern $\boldsymbol{x}$ belonging to i^{th} class will be identified by the subscript index i. From the text context it should be clear when the subscript is referred to i^{th} class or to the i^{th} pattern.

$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The quantity $\mathcal{L}(.)$ to be minimized now becomes:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) =& \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 + C\sum_{i=1}^{N}\sum_{j=1}^{K-1}\xi_{i,j} \\ &- \sum_{i=1}^{N}\sum_{j=1}^{y_i-1}\alpha_{i,j}((\boldsymbol{w}_{j+1}-\boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{j+1} - b_j - 1 + \xi_{i,j}) \\ &- \sum_{i=1}^{N}\sum_{j=y_i}^{K-1}\alpha_{i,j}((\boldsymbol{w}_j-\boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i) + b_j - b_{j+1} - 1 + \xi_{i,j}) \\ &- \sum_{i=1}^{N}\sum_{j=1}^{K-1}\beta_{i,j}\xi_{i,j}\end{aligned} \tag{B.3}$$

Simplifying Equation (B.3) a little more, it becomes:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = & \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{K-1}\xi_{i,j}(\alpha_{i,j}+\beta_{i,j}-C) \\ &- \sum_{i=1}^{N}\sum_{j=2}^{y_i}\alpha_{i,j-1}(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ &- \sum_{i=1}^{N}\sum_{j=1}^{y_i-1}(-\alpha_{i,j})(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ &- \sum_{i=1}^{N}\sum_{j=y_i}^{K-1}\alpha_{i,j}(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ &- \sum_{i=1}^{N}\sum_{j=y_i+1}^{K-1}(-\alpha_{i,j-1})(w_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{K-1}\alpha_{i,j}\end{aligned} \tag{B.4}$$

which can be reduced to

$$\begin{aligned}\mathcal{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = & \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{K-1}\xi_{i,j}(\alpha_{i,j}+\beta_{i,j}-C) + \sum_{i=1}^{n}\sum_{j=1}^{K-1}\alpha_{ij} \\ &- \sum_{i=1}^{N}\sum_{j=1}^{K} z_{i,j}(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1)\end{aligned} \tag{B.5}$$

$$\begin{aligned}\text{where } z_{i,j} = & \alpha_{i,j-1}I(j\geq 2)I(j\leq y_i) - \alpha_{i,j}I(j\leq y_i - 1) \\ & + \alpha_{i,j}I(j\geq y_i)I(j\leq K-1) - \alpha_{i,j-1}I(j\geq y_i+1)\end{aligned}$$

Setting the respective derivatives to zero we get:

$$\frac{\partial\mathcal{L}}{\partial b_j} = 0 \Leftrightarrow \sum_{i=1}^{N} z_{i,j} = 0, \quad j = 1,\ldots,K \tag{B.6}$$

$$\frac{\partial\mathcal{L}}{\partial \boldsymbol{w}_j} = 0 \Leftrightarrow \boldsymbol{w}_j - \sum_{i=1}^{n} z_{i,j} g(\boldsymbol{x}_i) = 0 \Leftrightarrow \boldsymbol{w}_j = \sum_{i=1}^{N} z_{i,j} g(\boldsymbol{x}_i), \quad j = 1\ldots,K \tag{B.7}$$

$$\frac{\partial\mathcal{L}}{\partial \xi_{i,j}} = 0 \Leftrightarrow (\alpha_{i,j}+\beta_{i,j}-C) = 0 \Leftrightarrow \alpha_{i,j} = C - \beta_{i,j}, \quad i = 1\ldots,N \tag{B.8}$$

with the Karush–Kuhn–Tucker (KKT) complementary conditions:

$$\alpha_{i,j}((\boldsymbol{w}_{j+1}-\boldsymbol{w}_j)^T g(\boldsymbol{x}_i)+b_{j+1}-b_j-1+\xi_{i,j})=0, \quad j=1,\ldots,y_i-1 \tag{B.9}$$

$$\alpha_{i,j}((\boldsymbol{w}_j-\boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i)+b_j-b_{j+1}-1+\xi_{i,j})=0, \quad j=y_i,\ldots,K-1 \tag{B.10}$$

$$\beta_{i,j}\xi_{i,j}=0 \tag{B.11}$$

for $i=1,\ldots,N$.

Thus we obtain the following dual problem

$$\begin{aligned}\max \mathcal{L}(\boldsymbol{\alpha}) &= \sum_{i=1}^{N}\sum_{j=1}^{K-1}\alpha_{i,j}-\frac{1}{2}\sum_{i,k=1}^{N}\sum_{j=1}^{K} z_{i,j}z_{k,j}H(\boldsymbol{x}_i,\boldsymbol{x}_k)\\ s.t &\begin{cases}\sum_{i=1}^{N} z_{i,j}=0 \quad j=1,\ldots,K-1\\ 0\le\alpha_{i,j}\le C, \quad i=1,\ldots,n\end{cases}\end{aligned} \tag{B.12}$$

The decision functions are given by

$$D_j(\boldsymbol{x})=\sum_{i=1}^{N} z_{i,j}H(\boldsymbol{x}_i,\boldsymbol{x})+b_j, \quad j=1,\ldots,K \tag{B.13}$$

and a pattern $\boldsymbol{x}$ will be classified as the class $\arg\max_{j=1,\ldots,K} D_j(\boldsymbol{x})$.

B.1.2 Sophisticated Architecture

Following Crammer and Singer (2002) formulation where they replace the slack variables $\xi_{i,j}$ with $\xi_i=\max_j \xi_{i,j}$, we extend this scheme to the new unimodal paradigm. Hence, the L_1 soft margin support vector machine can be obtained by minimizing the quantity

$$\mathcal{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi})=\frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2+C\sum_{i=1}^{N}\xi_i \tag{B.14}$$

where $\boldsymbol{w}=(\boldsymbol{w}_1,\ldots,\boldsymbol{w}_K)$, $=(b_1,\ldots,b_K)$ and $\boldsymbol{\xi}=(\xi_1,\ldots,\xi_n)$ where $\xi_i>0$.

To solve the problem stated in Equation (4.15) restricted to the conditions of the sophisticated architecture approach we introduce the nonnegative Lagrange multipliers $\alpha_{i,j}$ and β_i.

$$\begin{aligned}\mathcal{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) =& \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2+C\sum_{i=1}^{N}\xi_i\\ &-\sum_{i=1}^{N}\sum_{j=1}^{y_i-1}\alpha_{i,j}((\boldsymbol{w}_{j+1}-\boldsymbol{w}_j)^T g(\boldsymbol{x}_i)+b_{j+1}-b_j-1+\xi_i)\\ &-\sum_{i=1}^{N}\sum_{j=y_i}^{K-1}\alpha_{i,j}((\boldsymbol{w}_j-\boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i)+b_j-b_{j+1}-1+\xi_i)\\ &-\sum_{i=1}^{N}\beta_i\xi_i\end{aligned} \tag{B.15}$$

Doing similarly as before, one obtains:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 - \sum_{i=1}^{N}\xi_i((\beta_i - C) + \sum_{j=1}^{K-1}\alpha_{i,j}) \\ - \sum_{i=1}^{N}\sum_{j=2}^{y_i}\alpha_{i,j-1}(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ - \sum_{i=1}^{N}\sum_{j=1}^{y_i-1}(-\alpha_{i,j})(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ - \sum_{i=1}^{N}\sum_{j=y_i}^{K-1}\alpha_{i,j}(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ - \sum_{i=1}^{N}\sum_{j=y_i+1}^{K-1}(-\alpha_{i,j-1})(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ + \sum_{i=1}^{N}\sum_{j=1}^{K-1}\alpha_{i,j}\end{aligned} \tag{B.16}$$

Which can be reduced to:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\sum_{j=1}^{K}\|\boldsymbol{w}_j\|^2 - \sum_{i=1}^{N}\xi_i((\beta_i - C) + \sum_{j=1}^{K-1}\alpha_{i,j}) + \sum_{i=1}^{N}\sum_{j=1}^{K-1}\alpha_{i,j} \\ - \sum_{i=1}^{N}\sum_{j=1}^{K} z_{i,j}(\boldsymbol{w}_j^T g(\boldsymbol{x}_i) + b_j - 1) \\ \text{where } z_{i,j} = \alpha_{ij-1}I(j \geq 2)I(j \leq y_i) - \alpha_{i,j}I(j \leq y_i - 1) \\ + \alpha_{i,j}I(j \geq y_i)I(j \leq K-1) - \alpha_{i,j-1}I(j \geq y_i + 1)\end{aligned} \tag{B.17}$$

The conditions of optimality are given by:

$$\frac{\partial\mathcal{L}}{\partial b_j} = 0 \Leftrightarrow \sum_{i=1}^{N} z_{i,j} = 0, \quad j = 1,\ldots,K \tag{B.18}$$

$$\frac{\partial\mathcal{L}}{\partial \boldsymbol{w}_j} = 0 \Leftrightarrow \boldsymbol{w}_j - \sum_{i=1}^{N} z_{i,j} g(\boldsymbol{x}_i) = 0 \Leftrightarrow \boldsymbol{w}_j = \sum_{i=1}^{n} z_{i,j} g(\boldsymbol{x}_i), \quad j = 1\ldots,K \tag{B.19}$$

$$\frac{\partial\mathcal{L}}{\partial \xi_i} = 0 \Leftrightarrow ((\beta_i - C) + \sum_{j=1}^{K-1}\alpha_{i,j} = 0 \Leftrightarrow \sum_{j=1}^{K-1}\alpha_{i,j} = C - \beta_i, \quad i = 1\ldots,N \tag{B.20}$$

And the KKT complementary conditions:

$$\alpha_{i,j}((\boldsymbol{w}_{j+1} - \boldsymbol{w}_j)^T g(\boldsymbol{x}_i) + b_{j+1} - b_j - 1 + \xi_i) = 0, \quad j = 1,\ldots,y_i - 1 \tag{B.21}$$

$$\alpha_{i,j}((\boldsymbol{w}_j - \boldsymbol{w}_{j+1})^T g(\boldsymbol{x}_i) + b_j - b_{j+1} - 1 + \xi_i) = 0, \quad j = y_i,\ldots,K \tag{B.22}$$

$$\beta_i\xi_i = 0, \quad i = 1,\ldots,n \tag{B.23}$$

for $i = 1,\ldots,N$.

Thus we obtain the following dual problem

$$\max \mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \sum_{j=1}^{K-1} \alpha_{i,j} - \frac{1}{2} \sum_{i,k=1}^{N} \sum_{j=1}^{K} z_{i,j} z_{k,j} H(\boldsymbol{x}_i, \boldsymbol{x}_k)$$
$$s.t \begin{cases} \sum_{i=1}^{N} z_{i,j} = 0 \quad j = 1, \ldots, K-1 \\ 0 \leq \sum_{j=1}^{K-1} \alpha_{i,j} \leq C, \quad i = 1, \ldots, N \end{cases} \tag{B.24}$$

Decision functions are given by

$$D_j(\boldsymbol{x}) = \sum_{i=1}^{N} z_{i,j} H(\boldsymbol{x}_i, \boldsymbol{x}) + b_j, \quad j = 1, \ldots, K \tag{B.25}$$

as before, a pattern $\boldsymbol{x}$ will be classified as the class $\arg\max_{j=1,\ldots,K} D_j(\boldsymbol{x})$.

Bibliography

Shigeo Abe. *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*. Springer, 2005. ISBN 978-1852339296.

Janos Abonyi, Sandor Migaly, and Ferenc Szeifert. Fuzzy Self-Organizing Map based on Regularized Fuzzy C-Means Clustering. In Jose Benitez, Oscar Cordon, Frank Hoffmann, and Rajkumar Roy, editors, *Advances in Soft Computing - Engineering, Design and Manufacturing*, pages 99–108. Springer, London, 2003.

Ajith Abraham, Ajith Abraham, Rafael Falcn, and Rafael Bello. *Rough Set Theory: A True Landmark in Data Analysis*. Springer Publishing Company, Incorporated, 2009. ISBN 3540899200, 9783540899204.

Ali Ahmadi, Sigeru Omatu, Toru Fujinaka, and Toshihisa Kosaka. Improvement of Reliability in Banknote Classification Using Reject Option and Local PCA. *Information Sciences*, 168 (1-4):277–293, 2004. ISSN 0020-0255.

E. Alhoniemi, J. Himberg, and J. Vesanto. Probabilistic Measures for Responses of Self-Organizing Map Units. In *Proceedings of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99*, pages 286–290. ICSC Academic Press, 1999.

Silvia Angilella, Salvatore Greco, and Benedetto Matarazzo. Non-Additive Robust Ordinal Regression: A Multiple Criteria Decision Model Based on the Choquet Integral. *European Journal of Operational Research*, 201(1):277 – 288, 2010. ISSN 0377-2217.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Evaluation Measures for Ordinal Regression. In *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications*, pages 283–287, 2009.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Feature Selection for Ordinal Regression. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1748–1754, New York, NY, USA, 2010a. ACM. ISBN 978-1-60558-639-7.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Selecting Features for Ordinal Text Classification. In *Proceedings of the 1st Italian Information Retrieval Workshop*, pages 13–14, 2010b.

Peter L. Bartlett and Marten H. Wegkamp. Classification with a Reject Option using a Hinge Loss. *Journal of Machine Learning Research*, 9:1823–1840, 2008.

Geoffrey F. Beadle, Barbara Silver, Leslie Botnick, Samuel Hellman, and Jay R. Harris. Cosmetic Results Following Primary Radiation Therapy for Early Breast Cancer. *Cancer*, 54(12):2911–2918, 1984.

Nabil Belacel. Multicriteria Assignment Method PROAFTN: Methodology and Medical Application. *European Journal of Operational Research*, 125(1):175–183, 2000. ISSN 0377-2217.

Ricardo Bellazi, Ameen Abu-Hanna, and Jim Hunter, editors. *Artificial Intelligence in Medicine*, 2007.

Arie Ben-David. A Lot of Randomness is Hiding in Accuracy. *Engineering Applications of Artificial Intelligence*, 20(7):875–885, 2007. ISSN 0952-1976.

E. Berthonnaud, J. Dimnet, P. Roussouly, and H. Labelle. Analysis of the Sagittal Balance of the Spine and Pelvis Using Shape and Orientation Parameters. *Journal of Spinal Disorders & Techniques*, 18(1):40–47, 2005.

Michel Beuthe and Giuseppe Scannella. Comparative Analysis of UTA Multicriteria Methods. *European Journal of Operational Research*, 130(2):246–262, 2001. ISSN 0377-2217.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, 2007. ISBN 0387310738.

Jerzy Blaszczynski, Salvatore Greco, Roman Slowinski, and Marcin Szelg. Monotonic Variable Consistency Rough Set Approaches. *International Journal of Approximate Reasoning*, 50 (7):979–999, 2009. ISSN 0888-613X. Special Section on Graphical Models and Information Retrieval.

Abdenour Bounsiar, Edith Grall-Maës, and Pierre Beauseroy. A Kernel Based Rejection Method for Supervised Classification. In *International Journal of Computational Intelligence*, pages 312–321, 2006.

Abdenour Bounsiar, Pierre Beauseroy, and Edith Grall-Maës. General Solution and Learning Method for Binary Classification with Performance Constraints. *Pattern Recognition Letters*, 29(10):1455–1465, 2008. ISSN 0167-8655.

Sylvain Bouveret and Michel Lemaître. Computing Leximin-Optimal Solutions in Constraint Networks. *Artificial Intelligence*, 173(2):343–364, 2009. ISSN 0004-3702.

Andrew P. Bradley. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, 30(7):1145 – 1159, 1997. ISSN 0031-3203.

Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 978-3-540-88907-6.

L. Breiman, JH Friedman, R. Olshen, and CJ Stone. *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.

Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman & Hall, 1998.

Kim Cao-Van and Bernard De Baets. Consistent Representation of Rankings. In Harrie de Swart, Ewa Orlowska, Gunther Schmidt, and Marc Roubens, editors, *Theory and Applications of Relational Structures as Knowledge Instruments*, volume 2929 of *Lecture Notes in Computer Science*, pages 1966–1967. Springer Berlin / Heidelberg, 2003. ISBN 978-3-540-20780-1.

Jaime S. Cardoso and Maria J. Cardoso. Towards an Intelligent Medical System for the Aesthetic Evaluation of Breast Cancer Conservative Treatment. *Artificial Intelligence in Medicine*, 40:115–126, 2007.

Jaime S. Cardoso and Joaquim F. Pinto da Costa. Learning to Classify Ordinal Data: the Data Replication Method. *Journal of Machine Learning Research*, 8:1393–1429, 2007.

Jaime S. Cardoso and Ricardo Sousa. Classification Models with Global Constraints for Ordinal Data. In *Proceedings of The Ninth International Conference on Machine Learning and Applications (ICMLA)*, 2010.

Jaime S. Cardoso and Ricardo Sousa. Measuring the Performance of Ordinal Classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(8):1173–1195, 2011.

R. Caruana, S. Lawrence, and C. L. Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Neural Information Processing Systems Conference*, pages 402–408, 2000.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a Library for Support Vector Machines*, 2001.

Jianlin Cheng, Zheng Wang, and G. Pollastri. A Neural Network Approach to Ordinal Regression. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1279 –1284, 2008.

C. Chow. On Optimum Recognition Error and Reject Tradeoff. *Information Theory, IEEE Transactions on*, 16(1):41–46, 1970.

Wei Chu and Zoubin Ghahramani. Preference Learning with Gaussian Processes. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 137–144, New York, NY, USA, 2005a. ACM. ISBN 1-59593-180-5.

Wei Chu and Zoubin Ghahramani. Gaussian Processes for Ordinal Regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005b.

Wei Chu, Vikas Sindhwani, Zoubin Ghahramani, and S. Sathiya Keerthi. Relational Learning with Gaussian Processes. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 289–296. MIT Press, Cambridge, MA, 2007.

L.P. Cordella, C. De Stefano, C. Sansone, and M. Vento. An Adaptive Reject Option for LVQ Classifiers. In *Image Analysis and Processing*, volume LNCS 974/1995, pages 68–73. Springer, 1995a.

L.P. Cordella, C. De Stefano, F. Tortorella, and M. Vento. A Method for Improving Classification Reliability of Multilayer Perceptrons. *IEEE Transactions on Neural Networks*, 6 (5):1140–1147, 1995b.

David Cossock and Tong Zhang. Subset Ranking Using Regression. In Gábor Lugosi and Hans Simon, editors, *Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 605–619. Springer Berlin / Heidelberg, 2006. ISBN 978-3-540-35294-5.

Koby Crammer and Yoram Singer. On the Learnability and Design of Output Codes for Multiclass Problems. *Machine Learning*, 47(2-3):201–233, 2002.

Joaquim F. Pinto da Costa and Jaime S. Cardoso. Classification of Ordinal Data Using Neural Networks. *Lecture Notes in Artificial Intelligence*, 3720:690–697, 2005.

Joaquim F. Pinto da Costa, Hugo Alonso, and Jaime S. Cardoso. The Unimodal Model for the Classification of Ordinal Data. *Neural Networks*, 21:78–91, 2008.

Joaquim F. Pinto da Costa, Ricardo Sousa, and Jaime S. Cardoso. An All-at-Once Unimodal SVM Approach for Ordinal Classification. In *Proceedings of The Ninth International Conference on Machine Learning and Applications (ICMLA)*, 2010.

C. De Stefano, C. Sansone, and M. Vento. To Reject or Not to Reject: That is the Question - An Answer in Case of Neural Classifiers. *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 30(1):574–585, 2000.

Nicolas Delannay and Michel Verleysen. Collaborative Filtering with Interlaced Generalized Linear Models. *Neurocomputing*, 71(7-9):1300–1310, 2008. ISSN 0925-2312.

Krzysztof Dembczynski, Salvatore Greco, Wojciech Kotlowski, and Roman Slowinski. Statistical Model for Rough Set Approach to Multicriteria Classification. In *PKDD 2007: Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*, pages 164–175, Berlin, Heidelberg, 2007. Springer-Verlag.

Pandu Ranga Rao Devarakota, Bruno Mirbach, and Björn Ottersten. Reliability Estimation of a Statistical Classifier. *Pattern Recognition Letters*, 29:243–253, 2008. ISSN 0167-8655.

Józef Dombi and Ákos Zsiros. Learning Multicriteria Classification Models from Examples: Decision Rules in Continuous Space. *European Journal of Operational Research*, 160(3): 663–675, 2005. ISSN 0377-2217. Decision Analysis and Artificial Intelligence.

M. Doumpos and C. Zopounidis. *Multicriteria Decision Aid Classification Methods*. Kluwer Academic Publishers, Dordrecht, 2002.

M. Doumpos, K. Kosmidou, G. Baourakis, and C. Zopounidis. Credit Risk Assessment Using a Multicriteria Hierarchical Discrimination Approach: A Comparative Analysis. *European Journal of Operational Research*, 138(2):392–412, 2002. ISSN 0377-2217.

M. Doumpos, Y. Marinakis, M. Marinaki, and C. Zopounidis. An Evolutionary Approach to Construction of Outranking Models for Multicriteria Classification: The Case of the ELECTRE TRI Method. *European Journal of Operational Research*, 199(2):496–505, 2009. ISSN 0377-2217.

Michael Doumpos and Fotios Pasiouras. Developing and Testing Models for Replicating Credit Ratings: A Multicriteria Approach. *Computational Economics*, 25:327–341, 2005. ISSN 0927-7099.

Michael Doumpos and Athina Salappa. Feature Selection Algorithms in Classification Problems: An Experimental Evaluation. In *Proceedings of the 4th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering Data Bases*, pages 36:1–36:6. World Scientific and Engineering Academy and Society (WSEAS), 2005. ISBN 960-8457-09-2.

Michael Doumpos and Constantin Zopounidis. A multicriteria Classification Approach based on Pairwise Comparisons. *European Journal of Operational Research*, 158(2):378–389, 2004. ISSN 0377-2217. Methodological Foundations of Multi-Criteria Decision Making.

Michael Doumpos and Constantin Zopounidis. A Multicriteria Decision Support System for Bank Rating. *Decision Support Systems*, 50(1):55 – 63, 2010. ISSN 0167-9236.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2001. ISBN 0471056693.

Wouter Duivesteijn and Ad Feelders. Nearest Neighbour Classification with Monotonicity Constraints. In *ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pages 301–316, Berlin, Heidelberg, 2008. Springer-Verlag.

Ian N. Durbach. The Use of the SMAA Acceptability Index in Descriptive Decision Analysis. *European Journal of Operational Research*, 196(3):1229–1237, 2009.

M. Ehrgott. *Multicriteria Optimization.* Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 2000.

R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Ran El-Yaniv, Dmitry Pechyony, and Elad Yom-Tov. Better Multiclass Classification via a Margin-Optimized Single Binary Problem. *Pattern Recognition Letters*, 29:1954–1959, 2008.

Eduardo Fernandez, Jorge Navarro, and Sergio Bernal. Multicriteria Sorting Using a Valued Indifference Relation under a Preference Disaggregation Paradigm. *European Journal of Operational Research*, 198(2):602–609, 2009. ISSN 0377-2217.

César Ferri and José Hernández-Orallo. Cautious Classifiers. In *ROCAI*, pages 27–36, 2004.

César Ferri, Peter Flach, and José Hernández-Orallo. Delegating Classifiers. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning.* ACM, 2004. ISBN 1581138285.

J. Figueira, S. Greco, and M. Ehrgott. *Multiple Criteria Decision Analysis: State of the Art Surveys.* Springer Verlag, Boston, Dordrecht, London, 2005a.

José Figueira, Salvatore Greco, Matthias Ehrogott, Jean-Pierre Brans, and Bertrand Mareschal. PROMETHEE Methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, pages 163–186. Springer New York, 2005b. ISBN 978-0-387-23081-8.

Ronald Aylmer Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

A. Flexer. On the Use of Self-Organizing Maps for Clustering and Visualization. *Intelligent Data Analysis*, 5(5):373–384, 2001.

Eibe Frank and Mark Hall. A Simple Approach to Ordinal Classification. In *EMCL' 01: Proceedings of the 12th European Conference on Machine Learning*, pages 145–156, London, UK, 2001. Springer-Verlag.

Janick V. Frasch, Aleksander Lodwich, Faisal Shafait, and Thomas M. Breuel. A Bayes-true Data Generator for Evaluation of Supervised and Unsupervised Learning Methods. *Pattern Recognition Letters*, 32(11):1523–1531, 2011. ISSN 01678655.

Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press Professional, Inc., San Diego, CA, USA, 1990. ISBN 0-12-269851-7.

G. Fumera, I. Pillai, and F. Roli. Classification with Reject Option in Text Categorisation Systems. In *Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'2003)*, pages 582–587. IEEE Computer Society, 2003.

Giorgio Fumera and Fabio Roli. Support Vector Machines with Embedded Reject Option. In *SVM '02: Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, pages 68–82, London, UK, 2002. Springer-Verlag.

Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Multiple Reject Thresholds for Improving Classification Reliability. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 863–871, London, UK, 2000a. Springer-Verlag. ISBN 3-540-67946-4.

Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Reject Option with Multiple Thresholds. *Pattern Recognition*, 33(12):2099–2101, 2000b.

Johannes Fürnkranz and Eyke Hüllermeier. Pairwise Preference Learning and Ranking. In *Proceedings of the 14th European Conference on Machine Learning*, pages 145–156. Springer-Verlag, 2003.

Johannes Fürnkranz and Eyke Hüllermeier. Pairwise Preference Learning and Ranking. Technical report, Austrian Research Institute for Artificial Intelligence, Wien, Austria, 2003.

J. Gama and A. C.P.L.F. de Carvalho. Machine Learning. In *Machine Learning: Concepts, Methodologies, Tools and Applications*, pages 13–22. IGI-Global, 2012.

João Gama and Pavel Brazdil. Cascade Generalization. *Machine Learning*, 41(3):315–343, 2000.

A. E. Gasca, T. S. Salda na, G. J. S. Sánchez, G. V. Velasquéz, L. E. Rendón, B. I. M. Abundez, R. R. M. Valdovinos, and R. R. Cruz. A rejection option for the multilayer perceptron using hyperplanes. In *Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA'2011)*, volume LNCS 6593/2011, pages 51–60. Springer, 2011.

Lisa Gaudette and Nathalie Japkowicz. Evaluation Methods for Ordinal Classification. In Yong Gao and Nathalie Japkowicz, editors, *Proceedings of the 2nd Canadian Conference on Artificial Intelligence*, Lecture Notes in Computer Science, pages 207–210. Springer, 2009.

D. Giles. Calculating a standard error for the gini coefficient: Some further results. *Oxford Bulletin of Economics and Statistics*, 66(3):124–126, 2004.

C. Gini. Measurement of inequality of incomes. *The Economic Journal*, 31(121):124–126, 1921.

M. Goldszmidt, I. Cohen, A. Fox, and S. Zhang. Three research challenges at the intersection of machine learning, statistical induction, and systems. In *Proceedings of the 10th conference on Hot Topics in Operating Systems (HOTOS'05)*, volume 10, pages 1–6, 2005.

Thore Graepel, Matthias Burger, and Klaus Obermayer. Self-Organizing Maps: Generalizations and New Optimization Techniques. *Neurocomputing*, 21(1-3):173 – 190, 1998. ISSN 0925-2312.

Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Stéphane Canu. Support Vector Machines with a Reject Option. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Neural Information Processing Systems Conference*, pages 537–544. MIT Press, 2008.

Salvatore Greco, Benedetto Matarazzo, and Roman Slowinski. Rough Sets Theory for Multicriteria Decision Analysis. *European Journal of Operational Research*, 129(1):1–47, 2001. ISSN 0377-2217.

Salvatore Greco, Masahiro Inuiguchi, and Roman Slowinski. Fuzzy Rough Sets and Multiple-Premise Gradual Decision Rules. *International Journal of Approximate Reasoning*, 41(2): 179–211, 2006. ISSN 0888-613X. Advances in Fuzzy Sets and Rough Sets.

Salvatore Greco, Vincent Mousseau, and Roman Slowinski. Ordinal Regression Revisited: Multiple Criteria Ranking Using a Set of Additive Value Functions. *European Journal of Operational Research*, 191(2):416–436, 2008.

Barbara Hammer, Marc Strickert, and Thomas Villmann. Learning Vector Quantization for Multimodal Data. In *Proceedings of the International Conference on Artificial Neural Networks*, ICANN '02, pages 370–376, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44074-7.

J. Han and J. Gao. Research Challenges for Data Mining in Science and Engineering. In H. Kargupta, J. Han, P. S. Yu, R. Motwani, and V. Kumar, editors, *Next Generation of Data Mining*, pages 1–18. Chapman & Hall / CRC Press, 2009.

Edward F. Harrington. Online Ranking/Collaborative Filtering Using the Perceptron Algorithm. In *Proceedings of the 20th International Conference on Machine Learning*, pages 250–257, 2003.

J. R. Harris, M. B. Levene, G. Svensson, and S. Hellman. Analysis of Cosmetic Results Following Primary Radiation Therapy for Stages I and II Carcinoma of the Breast. *International journal of radiation oncology, biology, physics*, 5(2):257–261, 1979.

Felicity Hasson, Sinead Keeney, and Hugh McKenna. Research Guidelines for the Delphi Survey Technique. *Journal of Advanced Nursing*, 32(4):1008–1015, 2000.

T. Hastie and R. Tibshirani. Generalized Additive Models. *Statistical Science*, 1:297–318, 1986.

Simon Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 3 edition, 2008. ISBN 0131471392.

R. Herbei and M. H. Wegkamp. Classification with Reject Option. *The Canadian Journal of Statistics*, 34(4):709–721, 2006.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Regression Models for Ordinal Data: A Machine Learning Approach. Technical report, 1999.

I. N. Herstein and John Milnor. An Axiomatic Approach to Measurable Utility. *Econometrica*, 21(2):291–297, 1953. ISSN 00129682.

Frederick S. Hillier, Gerald J. Lieberman, Frederick Hillier, and Gerald Lieberman. *MP Introduction to Operations Research*. McGraw-Hill Science/Engineering/Math, 2004. ISBN 0073017795.

L. Holmström and A. Hämäläinen. The Self-Organizing Reduced Kernel Density Estimator. In *Proceedings of the 1993 IEEE International Conference on Neural Networks (ICNN'93)*, pages 417–421, 1993.

Jin Huang and C.X. Ling. Using AUC and Accuracy in Evaluating Learning Algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 17(3):299 – 310, 2005. ISSN 1041-4347.

F. Huédé, M. Grabisch, C. Labreuche, and P. Savéant. Integration and Propagation of a Multi-Criteria Decision Making Model in Constraint Programming. *Journal of Heuristics*, 12(4-5):329–346, 2006. ISSN 1381-1231.

Yevseyeva Iryna. *Solving Classification Problems with Multicriteria Decision Aiding Approaches*. University of Jyväskylä, 2007. ISBN 978-951-39-3049-3.

H. Ishibuchi and M. Nii. Neural Networks for Soft Decision Making. *Fuzzy Sets and Systems*, 34(115):121–140, 2000.

Alessio Ishizaka and Ashraf Labib. Analytic Hierarchy Process and Expert Choice: Benefits and Limitations. *OR Insight*, 22(4):201–220, 2009.

Alessio Ishizaka and Ashraf Labib. Review of the Main Developments in the Analytic Hierarchy Process. *Expert Systems with Applications*, 38(11):14336 – 14345, 2011. ISSN 0957-4174.

Alessio Ishizaka, Dieter Balkenborg, and Todd Kaplan. Does AHP Help Us Make a Choice? An Experimental Evaluation. *JORS*, 62(10):1801–1812, 2011.

Richard Jensen and Qiang Shen. Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. 2008.

Jeremy Jones and Duncan Hunter. Consensus Methods for Medical and Health Services Research. *British medical journal*, 331:376–380, 1995.

Ulrich Junker. Preference-Based Search and Multi-Criteria Optimization. *Annals of Operations Research*, 130(1):75—115, 2004.

Ulrich Junker. Preference-Based Problem Solving for Constraint Programming. pages 109–126, 2008.

Jyrki Kangas, Mikko Kurttila, Miika Kajanus, and Annika Kangas. Evaluating the Management Strategies of a Forestland Estate–the S-O-S approach. *Journal of Environmental Management*, 69(4):349–58, 2003.

Vojislav Kecman. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262112558.

M. Kendall. A New Measure of Rank Correlation. *Biometrika*, 30:81–89, 1938.

T. Kohonen. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 43(1):59–69, 1982.

T. Kohonen. The 'Neural' Phonetic Typewriter. *Computer*, 21(3):11–22, 1988.

T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

Wojciech Kotlowski, Krzysztof Dembczynski, Salvatore Greco, and Roman Slowinski. Stochastic Dominance-based Rough Set Model for Ordinal Classification. *Information Sciences*, 178(21):4019–4037, 2008.

Stefan Kramer, Gerhard Widmer, Bernhard Pfahringer, and Michael De Groeve. Prediction of Ordinal Classes Using Regression Trees. *Fundamenta Informaticae*, 47(1-2):1–13, 2001.

Murat Köksalan and Selin Bilgin Özpeynirci. An Interactive Sorting Method for Additive Utility Functions. *Computers & Operations Research*, 36(9):2565–2572, 2009. ISSN 0305-0548.

Risto Lahdelma and Pekka Salminen. Prospect Theory and Stochastic Multicriteria Acceptability Analysis (SMAA). *Omega*, 37(5):961–971, 2009.

Risto Lahdelma, Pekka Salminen, and Joonas Hokkanen. Locating a Waste Treatment Facility by Using Stochastic Multicriteria Acceptability Analysis with Ordinal Criteria. *European Journal of Operational Research*, 142(2):345 – 356, 2002. ISSN 0377-2217.

Risto Lahdelma, Kaisa Miettinen, and Pekka Salminen. Ordinal Criteria in Stochastic Multicriteria Acceptability Analysis (SMAA). *European Journal of Operational Research*, 147 (1):117–127, 2003.

K. Lakiotaki, N.F. Matsatsinis, and A. Tsoukià ands. Multicriteria User Modeling in Recommender Systems. *Intelligent Systems, IEEE*, 26(2):64 –76, 2011. ISSN 1541-1672.

Kleanthi Lakiotaki, Pavlos Delias, Vangelis Sakkalis, and Nikolaos Matsatsinis. User Profiling based on Multi-Criteria Analysis: the Role of Utility Functions. *Operational Research*, 9: 3–16, 2009. ISSN 1109-2858.

Thomas C. W. Landgrebe, David M. J. Tax, Pavel Paclik, Robert P.W. Duin, and Colin Andrew. A Combining Strategy for Ill-Defined Problems. In *Fifteenth Annual Symposium of the Pattern Recognition Association of South Africa*, pages 57–62, 2004.

Thomas C. W. Landgrebe, David M. J. Tax, Pavel Paclík, and Robert P. W. Duin. The Interaction Between Classification and Reject Performance for Distance-Based Reject-Option Classifiers. *Pattern Recognition Letters*, 27:908–917, 2006. ISSN 0167-8655.

Mark Last, Abraham Kandel, and Oded Maimon. Information-Theoretic Algorithm for Feature Selection. *Pattern Recognition Letters*, 22(6-7):799 – 811, 2001. ISSN 0167-8655.

Niklas Lavesson and Paul Davidsson. Evaluating Learning Algorithms and Classifiers. *International Journal of Intelligent Information and Database Systems*, 1:37–52, 2007. ISSN 1751-5858.

H. Le Capitaine and C. Fré andlicot. An Optimum Class-Rejective Decision Rule and Its Evaluation. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3312–3315, 2010.

J. W. T. Lee and Da-Zhong Liu. Induction of Ordinal Decision Trees. In *Machine Learning and Cybernetics, International Conference on*, volume 4, pages 2220–2224, 2002.

K. H. Lee. *First Course On Fuzzy Theory And Applications*. SpringerVerlag, 2004. ISBN 3540229884.

H.-T. Lin and L. Li. Combining Ordinal Preferences by Boosting. In *Proceedings ECML/PKDD 2009 Workshop on Preference Learning*, pages 69–83, 2009.

Huan Liu and R. Setiono. Feature Selection via Discretization. *Knowledge and Data Engineering, IEEE Transactions on*, 9(4):642 –645, 1997.

F. Lotte, H. Mouchère, and A. Lécuyer. Pattern rejection strategies for the design of self-paced EEG-based brain-computer interfaces. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR'2008)*, pages 1–5, 2008.

Fabio Maccheroni, Massimo Marinacci, and Aldo Rustichini. Ambiguity Aversion, Robustness, and the Variational Representation of Preferences. *Econometrica*, 74(6):1447–1498, 2006. ISSN 00129682.

J. L. Marichal. *Aggregation Operators for Multicriteria Decision Aid*. PhD thesis, Institute of Mathematics, University of Liège, Liège, Belgium, 1998.

C. L. C. Mattos and G. A. Barreto. ARTIE and MUSCLE models: building ensemble classifiers from fuzzy art and som networks. *Neural Computing & Applications*, pages 1–13, 2011. ISSN 0941-0643.

Petter McCullagh. Regression Models for Ordinal Data. *Journal of the Royal Statistical Society*, 42(2):109–142, 1980.

Michael McGeachie. Utility Functions for Ceteris Paribus Preferences. Master's thesis, Department of Electrical Engineering and Computer Science, MIT, 2002.

Michael McGeachie and Jon Doyle. Efficient Utility Functions for Ceteris Paribus Preferences. In *Eighteenth national conference on Artificial intelligence*, pages 279–284, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence. ISBN 0-262-51129-0.

Michael McGeachie and Jon Doyle. Utility Functions for Ceteris Paribus Preferences. *Computational Intelligence*, 20(2):158–217, 2004.

P. Meyer and M. Roubens. Choice, Ranking and Sorting in Fuzzy Multiple Criteria Decision Aid. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 471–506. Springer Verlag, Boston, Dordrecht, London, 2005.

K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Dordrecht, 1999.

V. Mousseau, J. Figueira, and J. Ph. Naux. Using Assignment Examples to Infer Weights for ELECTRE TRI Method: Some Experimental Results. *European Journal of Operational Research*, 130(2):263–275, 2001. ISSN 0377-2217.

A. R. Rocha Neto and G. A. Barreto. On the Application of Ensembles of Classifiers to the Diagnosis of Pathologies of the Vertebral Column: A Comparative Analysis. *IEEE Transactions on Latin America*, 7(4):487–496, 2009. ISSN 1548-0992.

Ajalmar R. R. Neto, Ricardo Sousa, Guilherme Barreto, and Jaime S. Cardoso. Diagnostic of Pathology on the Vertebral Column with Embedded Reject Option. In *Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2011.

Sigurdur Olafsson, Xiaonan Li, and Shuning Wu. Operations Research and Data Mining. *European Journal of Operational Research*, 187(3):1429 – 1448, 2008. ISSN 0377-2217.

Helder Oliveira, Andre Magalhaes, Maria J. Cardoso, and Jaime S. Cardoso. An Accurate and Interpretable Model for BCCT.core. In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6158–6161, 2010.

Hua Ouyang and Alex Gray. Learning Dissimilarities by Ranking: From SDP to QP. In *International Conference on Machine Learning*, pages 728–735, 2008.

R.D. Pascual-Marqui, A.D. Pascual-Montano, K. Kochi, and J.M. Carazo. Smoothly Distributed Fuzzy C-Means: A New Self-Organizing Map. *Pattern Recognition*, 34(12):2395 – 2402, 2001. ISSN 0031-3203.

Z. Pawlak. Rough Sets. *International Journal of Computer and Information Sciences*, 11(5): 341–356, 1982.

Zdzislaw Pawlak. Rough Set Approach to Knowledge-based Decision Support. *European Journal of Operational Research*, 99(1):48–57, 1997. ISSN 0377-2217.

H. Peng and S. Zhu. Handling of Incomplete Data Sets Using ICA and SOM in Data Mining. *Neural Computing & Applications*, 16(2):167–172, 2007.

Bernard Pierquin, Judith Huart, Michel Raynal, Yves Otmezguine, Elie Calitchi, Jean-Jacques Mazeron, Gerard Ganem, Jean-Paul Le Bourgeois, Ginette Marinello, Michel Julien, Bernard Brun, and Franck Feuilhade. Conservative Treatment for Breast Cancer: Long-Term Results (15 years). *Radiotherapy and Oncology*, 20(1):16–23, 1991.

Tadeusz Pietraszek. Optimizing Abstaining Classifiers using ROC Analysis. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 665–672, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5.

Rob Potharst and Jan C. Bioch. A Decision Tree Algorithm for Ordinal Classification. In *Advances in Intelligent Data Analysis*, pages 187–198, 1999.

Rob Potharst and Jan C. Bioch. Decision Trees for Ordinal Classification. *Intelligent Data Analysis*, 4(2):97–111, 2000.

Rob Potharst and A. J. Feelders. Classification Trees for Problems with Monotonicity Constraints. *SIGKDD Explorations Newsletter*, 4(1):1–10, 2002.

William Press, Brian Flannery, Saul Teukolsky, and William Vetterling. *Numerical Recipes in C: the Art of Scientific Computing*. Cambridge University Press, Cambridge, 2002.

Angela Presson, Nam Yoon, Lora Bagryanova, Vei Mah, Mohammad Alavi, Erin Maresh, Ayyappan Rajasekaran, Lee Goodglick, David Chia, and Steve Horvath. Protein Expression based Multimarker Analysis of Breast Cancer Samples. *BMC Cancer*, 11(1):230, 2011. ISSN 1471-2407.

Yoon Soo Pyon and Jing Li. Identifying Gene Signatures from Cancer Progression Data Using Ordinal Analysis. In *Bioinformatics and Biomedicine, 2009. BIBM '09. IEEE International Conference on*, pages 136 –141, 2009.

J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986. ISSN 0885-6125.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Belmont, California, 1993.

Ana Rebelo, Jakub Tkaczuk, Ricardo Sousa, and Jaime S. Cardoso. Metric Learning for Music Symbol Recognition. In *The tenth International Conference on Machine Learning and Applications*, 2011.

P. Rietveld and H. Ouwersloot. Ordinal Data in Multicriteria Decision Making: A Stochastic Dominance Approach to Siting Nuclear Power Plants. *European journal of operational research*, 56(2):249–262, 1992.

Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge Press, 1986.

M. Riveiro, F. Johansson, G. Falkman, and T. Ziemke. Supporting maritime situation awareness using self organizing maps and gaussian mixture models. In *Proceedings of the 2008 Conference on 10th Scandinavian Conference on Artificial Intelligence (SCAI'08)*, pages 84–91. IOS Press, 2008.

Irene Rodriguez-Lujan, Ramon Huerta, Charles Elkan, and Carlos Santa Cruz. Quadratic Programming Feature Selection. *Journal of Machine Learning Research*, 11:1491–1516, 2010.

Bernard Roy. The Outranking Approach and the Foundations of ELECTRE Methods. *Theory and Decision*, 31:49–73, 1991. ISSN 0040-5833.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. ISBN 0137903952.

Thomas L. Saaty. How to Make a Decision: The Analytic Hierarchy Process. *European Journal of Operational Research*, 48(1):9 – 26, 1990. ISSN 0377-2217.

Thomas L. Saaty and Luis G. Vargas. The Seven Pillars of the Analytic Hierarchy Process. In *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*, volume 34 of *International Series in Operations Research & Management Science*, pages 27–46. Springer US, 2001. ISBN 978-1-4615-1665-1.

C. M. Santos-Pereira and A. M. Pires. On optimal reject rules and ROC curves. *Pattern Recognition Letters*, 26(7):943–952, 2005.

Matthew Schultz and Thorsten Joachims. Learning a Distance Metric from Relative Comparisons. In *Neural Information Processing Systems Conference*. MIT Press, 2004.

Clayton Scott and Robert D. Nowak. Minimax-Optimal Classification With Dyadic Decision Trees. *IEEE Transactions on Information Theory*, 52(4):1335–1353, 2006.

Sambu Seo and Klaus Obermayer. Soft Learning Vector Quantization. *Neural Computation*, 15:1589–1604, 2002.

Sohan Seth and José C. Príncipe. Variable Selection: A Statistical Dependence Perspective. In *Proceeding of the Ninth International Conference on Machine Learning and Applications*, pages 931—936, 2010.

A. Shashua and A. Levin. Ranking with Large Margin Principle: Two Approaches. In Thrun and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 937–944, Cambridge, MA, 2003. MIT Press.

Libin Shen and Aravind Joshi. Ranking and Reranking with Perceptron. *Machine Learning*, 60:73–96, 2005.

S. F. Sim and V. Sági-Kiss. Multiple self-organising maps (mSOMs) for simultaneous classification and prediction: Illustrated by spoilage in apples using volatile organic profiles. *Chemometrics and Intelligent Laboratory Systems*, 109(1):57–64, 2011.

Y. Siskos, E. Grigoroudis, and N.F. Matsatsinis. UTA Methods. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 297–344. Springer Verlag, Boston, Dordrecht, London, 2005.

L. Siwik and S. Natanek. Elitist Evolutionary Multi-Agent System in Solving Noisy Multi-Objective Optimization Problems. *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, pages 3319–3326, 2008.

Yves De Smet and Linett Montano Guzmán. Towards Multicriteria Clustering: An Extension of the K-Means Algorithm. *European Journal of Operational Research*, 158(2):390 – 398, 2004. ISSN 0377-2217. Methodological Foundations of Multi-Criteria Decision Making.

Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, Wei Wang. Data Mining Curriculum: a Proposal (Version 1.0). 2006. Retrieved January 23, 2012.

Ricardo Sousa. Automatic Aesthetic Evaluation of Breast Cancer Conservative Treatment. Master's thesis, Universidade do Porto, 2008.

Ricardo Sousa and Jaime S. Cardoso. The Data Replication Method for the Classification with Reject Option. (**submitted**).

Ricardo Sousa and Jaime S. Cardoso. Ensemble of Decision Trees with Global Constraints for Ordinal Classification. In *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2011.

Ricardo Sousa, Ajalmar R. da Rocha Neto, Jaime S. Cardoso, and Guilherme A. Barreto. Self-Organizing Maps for Classification with Reject Option. (**submitted**).

Ricardo Sousa, Beatriz Mora, and Jaime S. Cardoso. An Ordinal Data Method for the Classification with Reject Option. In *Proceedings of The Eighth International Conference on Machine Learning and Applications (ICMLA)*, 2009.

Ricardo Sousa, Helder P. Oliveira, and Jaime S. Cardoso. Feature Selection with Complexity Measure in a Quadratic Programming Setting. In *Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, pages 524–531, 2011.

Ricardo Sousa, Irina Yevseyeva, Joaquim F. Pinto da Costa, and Jaime S. Cardoso. Multicriteria Models for Learning Ordinal Data: A Literature Review. In Xin-She, editor, *Artificial Intelligence, Evolutionary Computation and Metaheuristics (AIECM) –In the footsteps of Alan Turing (Turing 2012)*. Springer, 2012.

A. H. Souza Júnior, G. A. Barreto, and A. T. Varela. A Speech Recognition System for Embedded Applications Using the SOM and TS-SOM networks. In J. I. Mwasiagi, editor, *Self-Organizing Maps - Applications and Novel Algorithm Design*, pages 97–108. InTech Open, 2011.

C. Spearman. The Proof and Measurement of Association Between two Things. *American Journal of Psychology*, 15:72–101, 1904.

P. Sridhar, A.M. Madni, and M. Jamshidi. Multi-Criteria Decision Making in Sensor Networks. *Instrumentation Measurement Magazine, IEEE*, 11(1):24 –29, 2008. ISSN 1094-6969.

Bing-Yu Sun, Jiuyong Li, D.D. Wu, Xiao-Ming Zhang, and Wen-Bo Li. Kernel Discriminant Learning for Ordinal Regression. *Knowledge and Data Engineering, IEEE Transactions on*, 22(6):906 –910, 2010. ISSN 1041-4347.

J. Suutala, S. Pirttikangas, J. Riekki, and J. R$'$oning. Reject-Optional LVQ-Based Two-Level Classifier to Improve Reliability in Footstep Identification. In *Pervasive Computing*, pages 182–187. Springer, 2004.

Alberto Tagliafico, Giulio Tagliafico, Simona Tosto, Fabio Chiesa, Carlo Martinoli, Lorenzo E. Derchi, and Massimo Calabrese. Mammographic Density Estimation: Comparison Among BI-RADS Categories, a Semi-Automated Software and a Fully Automated One. *The Breast*, 18(1):35–40, 2009.

Hamdy A. Taha. *Operations Research: An Introduction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. ISBN 0131889230.

D. M. J. Tax and R. P. W. Duin. Growing a Multi-Class Classifier with a Reject Option. *Pattern Recognition Letters*, 29:1565–1570, 2008. ISSN 0167-8655.

Tommi Tervonen and José Rui Figueira. A Survey on Stochastic Multicriteria Acceptability Analysis Methods. *Journal of Multi-Criteria Decision Analysis*, 15:1–14, 2008.

Tommi Tervonen and Risto Lahdelma. Implementing Stochastic Multicriteria Acceptability Analysis. *European Journal of Operational Research*, 178(2):500–513, 2007.

Francesco Tortorella. Reducing the Classification Cost of Support Vector Classifiers through an ROC-based Reject Rule. *Pattern Analysis and Applications*, 7:128–143, 2004. ISSN 1433-7541.

Francesco Tortorella. A ROC-based Reject Rule for Dichotomizers. *Pattern Recognition Letters*, 26:167–180, 2005. ISSN 0167-8655.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support Vector Learning for Interdependent and Structured Output Spaces. In *International Conference on Machine Learning*, 2004.

A. M. Turky and M. S. Ahmad. The Use of SOM for Fingerprint Classification. In *IEEE International Conference on Information Retrieval & Knowledge Management (CAMP'2010)*, pages 287–290, 2010.

G. Tutz. Generalized Semiparametrically Structured Ordinal Models. *Biometrics*, 59:263–273, 2003.

L. Ustinovichius, E. K. Zavadskas, and V. Podvezko. The Application of a Quantitative Multiple Criteria Decision Making (MCDM-1) Approach to the Analysis of Investments in Construction. *Control and cybernetics*, 36, 2007.

A. Utsugi. Density Estimation by Mixture Models with Smoothing Priors. *Neural Computation*, 10:2115–2135, 1998.

M. van Hulle. Self-Organizing Maps. In G. Rozenberg, T. Baeck, and J. Kok, editors, *Handbook of Natural Computing: Theory, Experiments, and Applications*, pages 1–45. Springer-Verlag, 2010.

S. Vanbelle and A. Albert. A Note on the Linearly Weighted Kappa Coefficient for Ordinal Scales. *Statistical Methodology*, 6(2):157–163, 2009.

Van Belle Vanya, Pelckmans Kristiaan, Suykens Johan A. K., and Van Huffel Sabine. Learning Transformation Models for Ranking and Survival Analysis. *Journal of machine learning research*, 12:819–862, 2011.

Vladimir Naumovich Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

G. C. Vasconcelos, M. C. Fairhurst, and D. L. Bisset. Enhanced Reliability of Multilayer Perceptron Networks through Controlled Pattern Rejection. *Electronics Letters*, 29(3): 261–263, 1993.

Willem Waegeman, Bernard De Baets, and Luc Boullart. A Comparison of Different ROC Measures for Ordinal Regression. In *Proceedings of the CML 2006 workshop on ROC Analysis in Machine Learning*, 2006.

Willem Waegeman, Bernard De Baets, and Luc Boullart. ROC Analysis in Ordinal Regression Learning. *Pattern Recognition Letters*, 29(1):1 – 9, 2008. ISSN 0167-8655.

Willem Waegeman, Bernard De Baets, and Luc Boullart. Kernel-based Learning Methods for Preference Aggregation. *4OR: A Quarterly Journal of Operations Research*, 7:169–189, 2009. ISSN 1619-4500.

Chuan Wang and J.C. Principe. Training Neural Networks with Additive Noise in the Desired Signal. *Neural Networks, IEEE Transactions on*, 10(6):1511 –1517, 1999. ISSN 1045-9227.

Jiang-Jiang Wang, You-Yin Jing, and Chun-Fa Zhang. Weighting Methodologies in Multi-Criteria Evaluations of Combined Heat and Power Systems. *International Journal of Energy Research*, 33(12):1023–1039, 2009a. ISSN 1099-114X.

Jiang-Jiang Wang, You-Yin Jing, Chun-Fa Zhang, and Jun-Hong Zhao. Review on Multi-Criteria Decision Analysis Aid in Sustainable Energy Decision-Making. *Renewable and Sustainable Energy Reviews*, 13(9):2263 – 2278, 2009b. ISSN 1364-0321.

Kilian Q. Weinberger and Lawrence K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 10:207–244, 2009. ISSN 1532-4435.

D. H. Wolpert. The Supervised Learning No-Free-Lunch Theorems. In *In Proc. 6th Online World Conference on Soft Computing in Industrial Applications*, pages 25–42, 2001.

Fen Xia, Qing Tao, Jue Wang, and Wensheng Zhang. Recursive Feature Extraction for Ordinal Regression. *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 78–83, 2007.

Xiujuan Xu, Chunguang Zhou, and Zhe Wang. Credit Scoring Algorithm based on Link Analysis Ranking with Support Vector Machine. *Expert Systems with Applications*, 36: 2625–2632, 2009. ISSN 0957-4174.

Liu Yang and Rong Jin. Distance Metric Learning: A Comprehensive Survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.

H. Yin. The Self-Organizing Maps: Background, Theories, Extensions and Applications. In J. Fulcher and L. C. Jain, editors, *Computational Intelligence: A Compendium*, volume 115 of *Studies in Computational Intelligence*, pages 715–762. Springer-Verlag, 2008.

H. Yin and N. M. Allinson. Self-Organizing Mixture Networks for Probability Density Estimation. *IEEE Transactions on Neural Networks*, 12(2):405–411, 2001.

Shipeng Yu, Kai Yu, Volker Tresp, and Hans-Peter Kriegel. Collaborative Ordinal Regression. In *Proceedings of the 23rd international conference on Machine learning*, ICML, pages 1089–1096. ACM, 2006. ISBN 1-59593-383-2.

Ming Yuan and Marten Wegkamp. Classification Methods with Reject Option Based on Convex Risk Minimization. *Journal of Machine Learning Research*, 11:111–130, 2010. ISSN 1532-4435.

R. Zhang and D. Metaxas. RO-SVM: Support Vector Machine with Reject Option for Image Categorization. In *Proceedings of the British Machine Vision Conference*, pages 123.1–123.10, 2006.

Zhihua Zhang, James T. Kwok, and Dit-Yan Yeung. Parametric Distance Metric Learning with Label Information. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 1450–1452, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

C. Zopounidis and M. Doumpos. Building Additive Utilities for Multi-Group Hierarchical Discrimination: The M.H.DIS method. *Optimization Methods and Software*, 14(3):219–240, 2000.

C. Zopounidis and M. Doumpos. Multicriteria Classification and Sorting Methods: A Literature Review. *European Journal of Operational Research*, 138(2):229–246, 2002. ISSN 0377-2217.

Constantin Zopounidis and Panos M. Pardalos. *Handbook of Multicriteria Analysis*. Applied Optimization 103. Berlin: Springer. xxv, 2010.

A. M. Zoubir and D. Robert Iskander. Bootstrap Methods and Applications. *IEEE Signal Processing Magazine*, 24(4):10–19, 2007.