Master's Thesis

Ordinal Multi-Class Molecular Classification

Thesis Committee:
Prof.dr.ir. M.J.T. Reinders
Dr.ir. W.F.J. Verhaegh
Dr.ir. R.P.W. Duin
Dr. D.M.J. Tax
Ir. M.H. van Vliet

Author	O.P.Pfeiffer
Email	o.p.pfeiffer@student.tudelft.nl
Student number	1041835
Thesis supervisors	Prof.dr.ir. M.J.T. Reinders Dr.ir. W.F.J. Verhaegh
Date	April 24, 2009

Information and Communication Theory Group [I,C)^T

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Preface

This document is a result of my graduation project done at Philips Research in Eindhoven as part of the MSc degree in Bioinformatics at the Delft University of Technology. It consists of a research article, the supplementary material that comes with the article and a work document.

The research article is the main part, that describes the results and conclusions based on a comparison study, which was done as a part of this graduation project. The supplementary material contains additional information on topics discussed and will be referred to throughout the article. The work document covers more details on the graduation project itself, including the problem description, planning, implementation and any additional experiments that were not included in the research article.

The graduation project started on July 1, 2009 and will be defended on May 7, 2009. It was supervised by dr.ir. W.F.J. Verhaegh from Philips Research and prof.dr.ir. M.J.T. Reinders of the Delft University of Technology. I would like to thank both of my supervisors for their support and input throughout the project.

O.P.Pfeiffer
April 24, 2009

Vol. 01 no. 01 2009
Pages 1–13

Ordinal Multi-Class Molecular Classification

O.P.Pfeiffer[1,2] et al.

[1]Bioinformatics Department, Delft University of Technology, Delft, The Netherlands
[2]Molecular Diagnostics Department, Philips Research, Eindhoven, The Netherlands

April 24, 2009

ABSTRACT

Motivation: When a cancer grows, it progresses from one stage to another, which can been seen as a sequence of ordered phases. Current research on multi-class molecular classification typically treats the classes on a nominal scale and thus does not take any relation between classes into account. The ordering is however valuable information which may be used to improve the predictive power of a classifier. A few ordinal classifiers have been published, but they have not been applied in the analysis of molecular data, where there are only a limited number of samples in comparison to the number of features. This paper describes a comparative study in which current ordinal classifiers are benchmarked in a molecular analysis of gene expression. This helps to determine whether using the relation between classes can help to improve the prediction results.

Results: The results of the comparison study shows that there is not a lot of difference in performance between nominal and ordinal classifiers evaluated on real datasets. Several experiments were executed to further investigate any difference between both types of classifiers. It seems that by selecting monotonous features (i.e. features that correlate linearly with the class labels), the performance of nominal classifiers can be significantly improved. This allows for the usage of well-known and less complex classifiers, which is beneficial in $p > n$ problems.

Contact: o.p.pfeiffer@student.tudelft.nl

1 INTRODUCTION

When dealing with cancers it is important to know the severity to correctly determine the diagnosis and exact treatment for a patient. The severity is based on factors such as the location of the primary tumor, certain histological features and the presence of metastasis. The TNM system [1] is a general staging system which can be used for any type of cancer and is most commonly used. It specifies the current state of the cancer and how it has spread. More specific systems were also developed such as the Dukes' score [2] for colorectal cancer. To help identify how the cancer is growing, a grading system can be used. These systems are specific to the type of cancer, so there is no general system available. An example of a grading system is the Gleason's score [3] for prostate cancer. See Section 1.1 of the Supplementary Material for more details on the different staging and grading systems.

Currently the stage and grade of a cancer is determined by a specialist, which has as a disadvantage that diagnosis is limited to for example, the morphological appearance. In other words, not all information available is used. With the development of technology such as microarrays and mass spectrometry, classification algorithms from other research fields are being used to relate molecular profiles to diseases like cancer.

This study focuses on classification problems in which either the grades or stages of a cancer are used as classes, which means that there are more than two classes, with an ordinal relation between them. These class labels can take values such as "small", "medium" and "large" or numbers like $1, 2, 3$ as long as the ordinal relation between them holds (i.e. "small" $<$ "medium" $<$ "large" and $1 < 2 < 3$, respectively).

Research on multi-class molecular classification usually treats these classes nominally and ignores the ordinal relation. By discarding the ordinal relation, information is being omitted which could help to improve the performance of classification algorithms. The goal of this study is to test this hypothesis.

Datasets of prostate [4], breast [5] and ovarian [6] cancer are studied for which three different types of staging/grading systems were used. In addition, synthetic datasets were included to help to identify properties specific to ordinal classification.

2 METHODS

In multi-class classification problems there are K classes and the goal is to estimate the function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps instances from the feature space $\mathcal{X}$ to a set of discrete labels $\mathcal{Y} = \{\omega_1, ..., \omega_K\}$.

A classifier is an estimator $\hat{f}$ of the function f, which is estimated using samples $X = [x_1, x_2, ..., x_n] \in \mathcal{X}$ and their labels $y = [y_1, y_2, ..., y_n] \in \mathcal{Y}$. Once the $\hat{f}$ has been estimated, a new sample $x \in \mathcal{X}$ can be classified by determining $\hat{f}(x)$.

To evaluate the performance of the classifiers included in this study, an adapted version of the protocol proposed by Wessels et al. [7] is used to minimize bias that could occur. The protocol involves two nested cross-validation [8] loops of which the inner loop is used to optimize any parameters used (for example a threshold used by a classifier, or the number of features used to train a classifier), and the outer loop is used to validate the resulting classifier. These two nested cross-validation steps are permuted a number of times to ensure the results are not biased on the data was split in a training and validation set. The process is depicted in Figure 1.

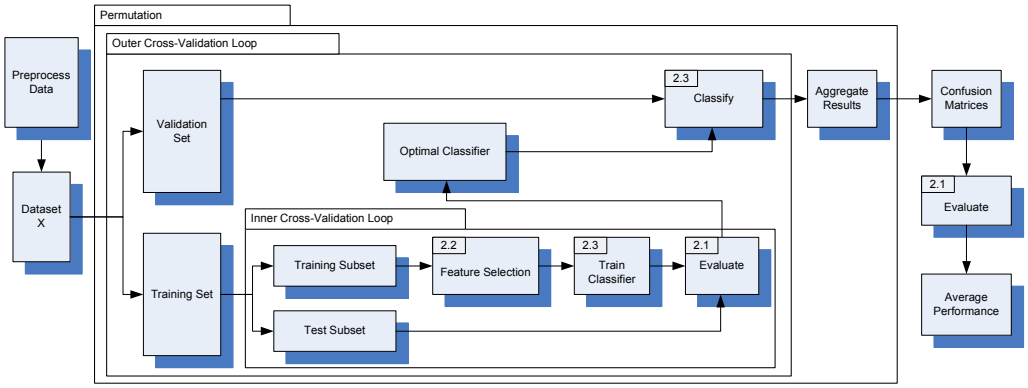


Figure 1: **Evaluation Protocol.** The preprocessed data is split up in a training and validation set in the outer cross-validation loop using stratified sampling. The training set is used to find the optimal feature subset by applying another cross-validation loop. This inner cross-validation loop results in an optimal trained classifier which is evaluated against the untouched validation set from the outer loop. This ensures that the performance is not biased because of how the dataset has been split. The samples and their predicted labels from each outer fold are aggregated into one big pool to create a confusion matrix for the whole data set. Any evaluation measure can then be applied on the confusion matrix to obtain the performance for that evaluation. This whole process is permuted 50 times and averaging the performance of all permutations results in the final performance. In the outer cross-validation loop the data is split in 3 folds, while in the inner cross-validation loop 10 folds are used. The numbers displayed in some of the boxes refer to their corresponding section numbers.

2.1 Evaluation Measures

In the case of nominal classification it is quite common to assess the performance of a classifier by evaluating the number of true positives and negatives. In this study the balanced accuracy will be used, which is the average of the accuracy rate for each class. When using such a measure, the way a sample was misclassified does not matter. In the case of ordinal classification this plays a far more important role.

When a sample is misclassified, one would want the predicted label to be as close as possible to the true label. In other words, the risk as a result of misclassifying a sample should be minimized since the consequences can be quite undesirable (e.g. the dose for radiotherapy can depend on the predicted severity of the cancer [9], thus overclassifying can lead to a more risky dosage than required). To assess the samples that are misclassified, two different measures will be used: a cost-based distance measure m_{cd} and a rank-based measure r_{int}. When a sample is misclassified, one would want the predicted label to be as close as possible to the true label. In other words, the risk as a result of misclassifying a sample should be minimized since the consequences can be quite undesirable (e.g. the dose for radiotherapy can depend on the predicted severity of the cancer [9], thus overclassifying can lead to a more risky dosage than required). To assess the samples that are misclassified, two different measures will be used: a cost-based distance measure m_{cd} and r_{int} coefficient.

To help compare performances, each evaluation measure was transformed into the interval $[0, 1]$ if necessary (where a higher value means a better performance).

Cost-based Distance Measure m_{cd}

This measure assigns costs based on the distance between the predicted and true label, where the distance is defined as the number of classes going from the predicted to the true label on the ordinal scale. By doing an element-wise multiplication between the cost matrix and the prediction's confusion matrix, the costs of that prediction is determined. By dividing these costs by the costs of the worst-case scenario (i.e. when all samples are classified as the class furthest away from the true class), a quotient is obtained that indicates how good the classifier performed.

More formally, given a confusion matrix A and a cost matrix M the m_{cd} can be defined as:

$$m_{\text{cd}}(M, A) = 1 - \frac{c}{t} \tag{1}$$

where c the actual total misclassification cost and t the maximal misclassification cost:

$$\begin{aligned} c &= \sum_{i=1}^{K} \sum_{j=1}^{K} M_{ij} A_{ij} \\ t &= \sum_{i=1}^{K} (\max_{j=1,..,K} M_{ij} \sum_{j=1}^{K} A_{ij}) \end{aligned} \tag{2}$$

Here A_{ij} is the number of samples of class i that were predicted as class j and M_{ij} is the cost of misclassifying a sample as class j while the true class is i. By default M is a linear cost matrix where the diagonal is equal to zero and the costs for misclassifying increases linearly with the distance to the true class; $M_{ij} = |i - j|$ for $i, j = 1, 2, ..., K$. See Equation 3 as an example of a linear cost matrix for $K = 4$.

$$M = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix} \tag{3}$$

When m_{cd} is equal to 0 then the classifier classified the samples in the worst possible way; a value of 1 indicates that the samples were classified perfectly. The definition of m_{cd} given in Equation 2 just looks at the overall situation, but the same can be done on a per-class basis similar to the balanced accuracy rate. The definition of a balanced version of m_{cd} can be found in Section 3.4.2 of the Supplementary Material.

Rank-based measure r_{int}

This ordinal measure [10] looks at how the predicted labels are ordered and compares it with the order of the true labels. Let O be the set of samples that needs to be validated, then any pair i, j for which the relation between the true labels $f(o_i) \leq f(o_j)$ holds, is added to the set S_{t}. Let S_{p} be a similar set but with the relations between the predicted labels $\hat{f}(o_i) \leq \hat{f}(o_j)$ instead. The measure r_{int} can then be defined as:

$$r_{\text{int}}(S_{\text{t}}, S_{\text{p}}) = -1 + 2\frac{|S_{\text{t}} \cap S_{\text{p}}|}{\sqrt{|S_{\text{t}}||S_{\text{p}}|}} \tag{4}$$

Where $|S|$ is the cardinality of a set S. More details and examples are given in Section 3.4.1 of the Supplementary Material.

2.2 Feature Selection

As the datasets contain a large number of features in comparison to the number of samples, it is important to reduce the number of features to ensure that the classes can be separated correctly [11]. To this end, a feature selection procedure takes place in the inner cross-validation loop (see Figure 1). There are several ways to achieve this goal, but for this study a fairly simple procedure is followed consisting of three steps.

Feature filtering is the first step, which removes any feature that is not significant enough (i.e. when it does not show enough difference over variance between the different classes). For this purpose the F-test with significance level $\alpha = 0.05$ was used, which is a special case of the one-way analysis of variance (ANOVA) [12, 13]. See Section 3.3.1 of the Supplementary Material for more information on ANOVA. All features that do not pass the F-test are discarded.

After the insignificant features have been filtered, the remaining ones are ranked based on their F-statistic.

The final step, unlike the previous two, depends on the classifier in question as it is used to evaluate the performance of possible feature subsets. For $i = 1, ..., n-1$ the topmost i features of the filtered and ranked set are used to construct a classifier using the training subset. The $n-1$ classifiers are then evaluated using the test subset to create a performance curve.

Each fold in the inner loop will result in a performance curve. After averaging the performance curves from all folds in the inner loop, the number of features for which the performance is highest will determine the size of the final subset of features (in case of ties the classifier is picked at random). This whole process of obtaining the optimal feature subset will be referred to as the basic feature selection method.

A disadvantage of such a simple procedure is that the true optimal subset of features will typically not be obtained. This is less of an issue here, as the purpose of this study is to compare the relative performance between the different classifiers and not to find the absolute performance of a classifier which might only be optimal in specific situations.

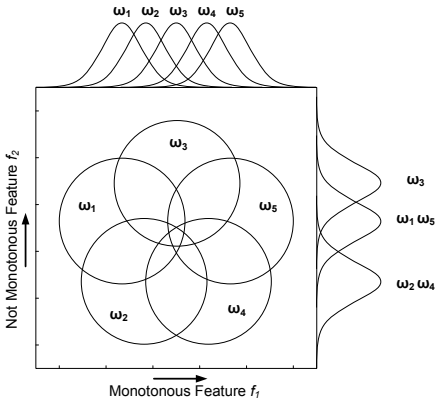


Figure 2: **Monotonous Features.** This synthetic dataset has 5 different classes which are ordered w_1 to w_5 when looking at only feature f_1 on the X-axis, while some classes overlap and disobey the ordinal relationship when looking at feature f_2 on the Y-axis. As a result the ordinal relation between the classes can not be found when looking only at feature f_2, while this is possible in the case of feature f_1. In this given example f_1 is a monotonous feature while f_2 is not.

Monotonous Features

Ordinal classification requires the relationship between the classes to be included. Such information can be included a-priori by using expert knowledge [14]. Assuming expert knowledge is not available, the ordinal relation has to be estimated from the training set's features. Thus the features should correlate with the class labels as much as possible (See Figure 2 for an example) and ideally increase or decrease monotonously in relation to the class labels.

To help investigate the effect of having features that contain the ordinal relation between classes, a variation on the previously described feature selection procedure has been included in this study. This variation will be referred to as the monotonous feature selection method and is similar to the basic one, although the ranking is done differently.

Instead of ranking the filtered genes based on their F-statistic, they are ordered based on the Pearson's correlation coefficient [15] between each feature and the class labels. The definition of the Pearson's correlation coefficient is given

in Section 3.3.2 of the Supplementary Material. The filtered genes are then ranked based on the absolute value of their correlation coefficients.

2.3 Classification

In Table 1 an overview is given of all the classification methods included in this study.

Classifier	Abbr.	Type
Naïve Bayes	NB	Nominal
Support Vector Machine	SVM	Nominal
One-vs-One Pseudo-Classification	OVO	Nominal
One-vs-All Pseudo-Classification	OVA	Nominal
Ordered Pseudo-Classification	OPC	Ordinal
Ordinal Support Vector Machine	OSVM	Ordinal
Cost-Based NB	CNB	Ordinal
Cost-Based OVO	COVO	Ordinal
Proportional Odds Log Regression	POLR	Ordinal

Table 1. A list of classifiers included in this study.

Nominal Classifiers

The naïve Bayes classifier (NB) [16] uses the Bayes' rule while assuming that all features are class-conditionally independent of each other. As a result, the classifier only needs to estimate the probability distribution of each feature individually, for which in this study the Gaussian model is used. It is a simple method which works surprisingly well on $n > p$ problems considering the strong assumptions on the independence.

The multi-class support vector machine (SVM) is a generalized version of the original SVM formulation [17]. Several generalizations are available, but in this study the one by Crammer et al. [18] was used.

One-vs-one pseudo-classification (OVO) [19] constructs binary classifiers for each pair of classes. So in case there are K classes, then there are $\binom{K}{2} = \frac{K(K-1)}{2}$ classifiers needed. To classify an unknown sample, the output of all classifiers needs to be combined and one way to do that is by voting. The most typical example of a voting scheme, and the one used in this study is majority voting [20].

In One-vs-all pseudo-classification (OVA) [21] K classifiers are constructed, one for each of the available classes. For the training of the ith classifier, the samples that belong to class i are used as positives and all other samples are used as negatives. Each classifier outputs the posterior probability of its class being the correct one. The classifier that is most confident in the prediction of its own class, will determine the label assigned to the unknown sample.

A more detailed description on these methods is given in the Supplementary Material in Section 3.1.

Ordered Pseudo-Classification

Ordered pseudo-classification (OPC) [22] is a method which constructs the binary classifiers in such way, that the ordinal relationship between classes can be enforced. As a result this approach can be applied independent of the binary classifiers being used as long as they return posterior class probabilities. $K-1$ binary classifiers are created which return the conditional probability $P(\omega_{\mathrm{T}} > \omega_i|x)$ where $i = 1, ..., K-1$ and ω_{T} is the true class of the unknown sample x (i.e. $\hat{f}(x)$ if the function f is known). In other words, each classifier i determines the chance that the class of the unknown sample is higher than the class ω_i given x. The corresponding probabilities for each class can then be calculated as indicated in Equation 5.

$$\begin{aligned} P(\omega_k|x) =& P(\omega_{\mathrm{T}} > \omega_{k-1}|x) - P(\omega_{\mathrm{T}} > \omega_k|x) \\ & \text{for } k = 1, ..., K \end{aligned} \tag{5}$$

Here $P(\omega_{\mathrm{T}} > \omega_0|x) = 1$ and $P(\omega_{\mathrm{T}} > \omega_K|x) = 0$ as $y \in \{\omega_1, \omega_2, ..., \omega_K\}$. For an unknown sample the label is determined by the class with the highest probability $P(\omega_k|x)$. A naïve Bayes classifier is used to determine the conditional probabilities $P(\omega_{\mathrm{T}} > \omega_k|x)$. See Section 3.2.1 in the Supplementary Material for more details.

Ordinal Support Vector Machine

OSVM [23] tries to solve the original multi-class problem as a binary problem while preserving the ordinal relationship between the classes. The advantage is that well-developed binary SVMs can be used for multi-class classification problems. The idea is based on a regular multi-class SVM formulation in which $K-1$ hyperplanes are constructed. However, instead of creating multiple hyperplanes, OSVM replicates the samples so it can solve the optimization problem in one go. This is done by adding $K-2$ dimensions to the feature space, which allows for the ordinal relation to be included into the problem. This process of replicating data is shown in Equation 6.

$$\begin{aligned} \begin{bmatrix} x_i^{\omega_j} \\ e_{q-1} \end{bmatrix} \in & \begin{cases} \bar{\omega}_1 & j = \max(1, q-s+1), ..., q \\ \bar{\omega}_2 & j = q+1, ..., \min(K, q+s) \end{cases} \\ & \text{for } q = 1, ..., K-1 \end{aligned} \tag{6}$$

Here $x_i^{\omega_j}$ are the samples that belong to class ω_j. If $q = 1$, e_{q-1} is a sequence of $K-2$ zeros, otherwise e_{q-1} is a sequence of $K-2$ symbols $0, ..., 0, h, 0, ..., 0$ with h at position $q-1$. The parameter s specifies the number of classes on both sides of the boundary that should be included. The h parameter is used as a trade-off between the objectives of maximizing the margin of separation and minimizing the distance between the hyperplanes.

Once the data has been replicated, it can be used to train a regular binary SVM. If an unknown sample needs to be classified, then that sample will be replicated once in all $K-2$ dimensions similarly as done in Equation 6, but without assigning a label. Each replica is then classified using the trained binary classifier, resulting in $K-1$ predictions $\in \{\bar{\omega}_1, \bar{\omega}_2\}$. Based on the ordinality assumption, the final prediction is then equal to the number of replicas classified as $\bar{\omega}_2$ plus one. See Section 3.2.2 in the Supplementary Material for a more detailed description of OSVM.

Cost-based Ordinal Classification

It is also possible to enforce the ordinal relation by assigning costs to the probability of misclassifying. By multiplying the costs with the posterior class probabilities, the risk for assigning a label to the unknown sample can obtained. The advantage of this approach is that any well-developed classifier can be used as long as it returns the class posteriors.

Using a constructed $K \times K$-cost matrix M, the risk of choosing class i can be determined using the following equation [24]:

$$R_j(x) = \sum_{i=1}^{K} M_{ij} P(\omega_i|x) \tag{7}$$

Here M_{ij} is the corresponding entry in the cost matrix M. To classify a new sample the risk R_j is calculated for each class j and the one with the lowest risk is the winner. In other words by selecting the class with the lowest risk, the costs of misclassifying that sample is minimized. By default a linear cost matrix is used (see Equation 3 for an example of such a cost matrix for $K = 4$), but other cost matrices can be used as well.

In the case of a cost-based approach of the naïve Bayes classifier (CNB) this is straightforward as it returns the needed posteriors for each class. The pseudo-classification methods (OVO and OVA) and the methods based on support vector machines (SVM and OSVM) on the other hand do not return probabilities. Although it is not uncommon to estimate the class posteriors, it would be less suitable in the context of determining the risk.

A cost-based approach of one-vs-one pseudo-classification (COVO) can be constructed by determining the risk for each individual classifier instead of using a voting-like scheme. By adding the risk from each classifier, the total risk for each class is obtained. The advantage of this approach over voting is that confidence of each individual classifier is taken into account, instead of giving each classifier an equal vote as is the case in majority voting. A similar approach is not possible for the one-vs-all pseudo classification as each binary classifier groups the remaining classes together. As a result the samples will be pushed to the center of the ordinal scale, which is not desirable. See Section 3.2.3 in the Supplementary Material for more details.

Proportional Odds Logistic Regression

The proportional odds logistic regression model [25] uses a continuous function $\tilde{f}$, which is discretized to obtain the estimator $\hat{f}$. It uses thresholds as defined in Equation 8 for this purpose.

$$\hat{f}(x) = \omega_k \qquad \text{if } \alpha_{k-1} \leq \tilde{f}(x) < \alpha_k \qquad \text{for } k = 1, ..., K \tag{8}$$

Here $\alpha_0, ..., \alpha_K$ are the thresholds separating $\tilde{f}(x)$ into K different classes. The two classes at both ends of the ordinal scale are defined by open-ended intervals where $\alpha_0 = -\infty$ and $\alpha_K = \infty$.

It is assumed, that the ratio of the odds of the event $y_i \leq \omega_k$ for any pair of sets of explanatory variables is independent of the choice of values for y. The model formulates the cumulative probabilities $P(y_i \leq \omega_k|x_i)$ as a latent variable $\tilde{y}$ as summarized in Equation 9.

$$\text{logit}(P(y_i \leq \omega_k|x_i)) = \text{logit}(\tilde{y}) = \alpha_k + x_i\beta \tag{9}$$

Each $\text{logit}(P(y_i \leq \omega_k|x_i))$ has its own threshold α_k but shares the same regression coefficients β. After estimating these parameters, the cumulative probabilities can then be used to determine the probabilities of each class (see Equation 10).

$$\begin{aligned} P(y_i = \omega_k|x_i) &= P(y_i \leq \omega_k|x_i) - P(y_i \leq \omega_{k-1}|x_i) \\ &\text{for } k = 1, ..., K \end{aligned} \tag{10}$$

The class with the highest probability will be assigned to the unknown sample. More details can be found in Section 3.2.4 of the Supplementary Material.

3 RESULTS

The classification methods included in this study were evaluated on several datasets; see Table 2 for a brief overview. These datasets are publicly available on-line through the Gene Expression Omnibus offered by the NCBI (`http://www.ncbi.nlm.nih.gov/geo/`).

Dataset	# Features	Class	# Samples
Prostate [4]	11476	Gleason 3	11
		Gleason 4	12
		Gleason 5	8
Breast [5]	10704	Elston I	68
		Elston II	126
		Elston III	55
Ovarian [6]	11533	TNM I	35
		TNM II	11
		TNM III	44
		TNM IV	9
Synthetic I & II	2	1	500
		2	500
		3	500
		4	500
		5	500

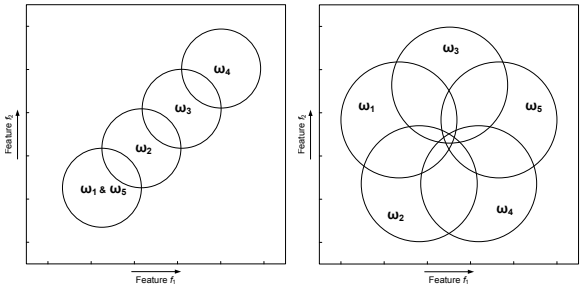
Table 2. Properties of the different datasets used. Gleason [3] and Elston [26] are both grading system which specifies the growth rate of the cancer, while TNM [1] is a staging system which specifies the spread of the cancer. See Section 2 in the Supplementary Material for more information on each dataset and how they were preprocessed.

Besides the real datasets, two synthetic datasets have been constructed as shown in Figure 3.

The purpose of the synthetic I dataset (see Figure 3a) is to distinguish between the results of nominal and ordinal classifiers. The balanced accuracy should not show much difference for both types of classifiers. A 'perfect' nominal classifier would mistake classes 1 and 5 half of the time. A 'perfect' ordinal classifier would be influenced by the

neighboring classes and thus would prefer class 1 over 5 or even classify the samples of both classes as class 3. That way it can ensure that a sample, if misclassified, lies as close to its true class as possible. The difference in both types of classifier should only become apparent when looking at how the samples were misclassified using an ordinal evaluation measure.

The dataset synthetic II consists of a monotonous feature on the X-axis and a non-monotonous feature on the Y-axis (see Figure 3b). The non-monotonous feature has some overlapping classes, so both the nominal and ordinal classifiers should use the monotonous feature as much as possible. This dataset was created to see how well both types of classifiers are able to use the ordinal information to their advantage.



(a) Synthetic I (b) Synthetic II

Figure 3: **Synthetic Datasets.** The synthetic datasets (a) I and (b) II. Each set has 5 classes, labeled $\omega_1, ..., \omega_5$, and 500 samples which are normally distributed. Both datasets only have 2 features, which are both monotonous in the synthetic I dataset. In the case of synthetic II, only feature f_1 on the X-axis is monotonous. More information on the datasets can be found in Section 2.4 of the Supplementary Material.

3.1 Comparison study

The comparison study consists of all classifiers listed in Table 1, using the balanced accuracy for the feature selection process and evaluated on all datasets. This was done using both the basic and monotonous feature selection. The evaluation protocol discussed in Table 1 was used and the results of this experiment are listed in Figure 4.

The box plots for POLR are not shown for the real datasets, as the implemented system was not able to fit the regression model for most of the permutations. This is a result of the limited amount of samples available for training in each permutation.

Looking at the real datasets there does not seem to be much difference from one classifier to another, not even between the nominal and ordinal classifiers. The ovarian dataset does show a little of variation between the classifiers, because it has one class more than the prostate and breast datasets. The high variance of the prostate dataset can be accounted to the limited number of samples.

When evaluated on the synthetic datasets, the classifiers show a lot more difference in performance. Especially for synthetic I there seems to be a distinction between the nominal and ordinal methods. To confirm this conjecture, a one-sided paired t-test was applied between the performance of each classifier with that of all other classifiers. The corresponding t- and p-values are listed in Table 3.

From the paired t-test of synthetic I it can be concluded, that the nominal classifiers outperform the ordinal ones based on the balanced accuracy, while they perform worse than ordinal classifiers based on m_{cd}. This behavior corresponds to the purpose of the synthetic I dataset, where the ordinal classifiers treat the classes ω_1 and ω_5 (see Figure 3) differently than the nominal classifiers would.

In the case of the synthetic II dataset, the nominal classifiers again outperform the ordinal ones based on the balanced accuracy. For m_{cd} however, both ordinal as nominal classifiers seem to perform well. This can be accounted to the way the features are used by each classifier. For example, NB assumes that the features are class-conditionally independent of each other and as a result is able to use feature f_1 (see Figure 3) to its advantage. A classifier like SVM on the other hand, uses both features at the same time and seems to have more difficulty to discriminate between the classes.

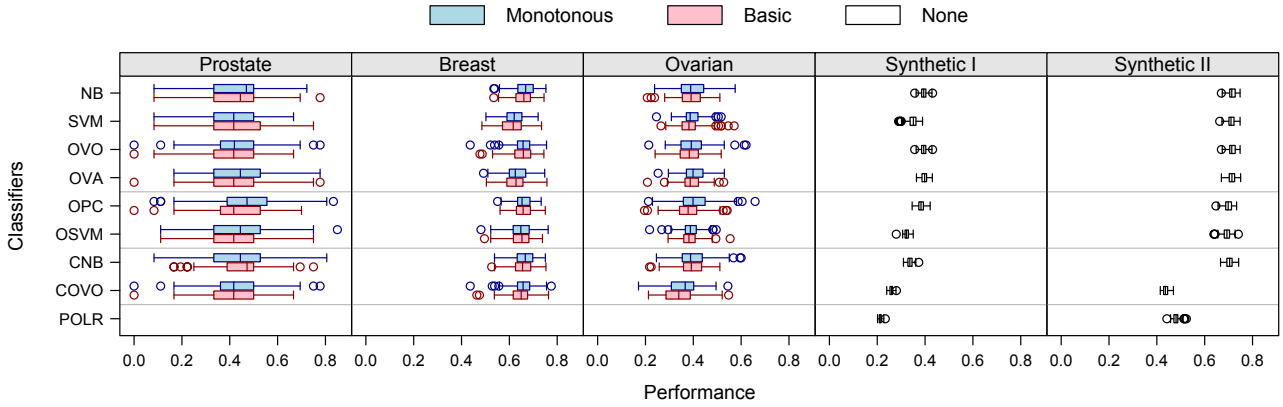
From all ordinal classifiers, only CNB seems to perform well based on m_{cd} in the synthetic II dataset. This can be a result of the classifier being able to exploit feature f_1, while at the same time use the cost matrix to enforce the ordinal relationship. It should be noted though, that some bias might have occurred as a linear cost matrix was used by CNB to predict samples and to evaluate them at the same time.

A downside of POLR becomes apparent when looking at its performance for both synthetic datasets. As the regression model uses the same regression coefficients, but different thresholds for each class, it is not very suitable for non-linear datasets. This would indicate that selecting strictly monotonous features might help improve the performance of POLR. This has not been verified however, as POLR was not able to fit the regression model on the real datasets.

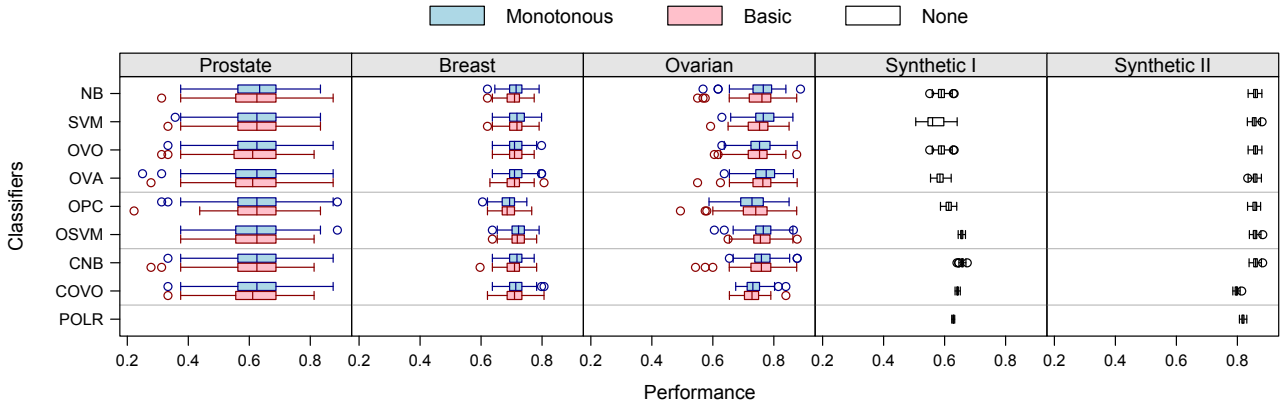
3.2 Evaluation

The classifiers in the previous section were trained with the balanced accuracy rate. In the evaluation protocol, the evaluation measure is however used in two different situations. First during feature selection it is used to evaluate the classifier for the different number of features (see the box 'Evaluate' in the inner cross-validation loop in Figure 1) and as a result determines the final optimal set of features. Later on it is used to evaluate the trained classifier on the test set which leads to the final performance measure after averaging the permutations (see the box 'Evaluate' outside of the cross-validation loops in Figure 1).

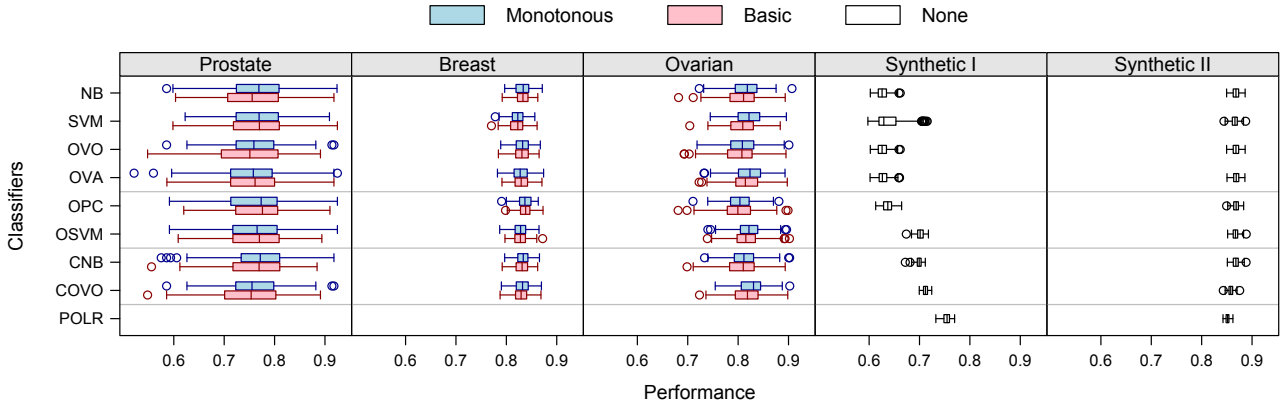
To help determine the overall effect on the performance when using different evaluation measures, all classifiers where trained and evaluated with the any combination of the measures included in this study. Table 4 lists the most



(a) Balanced Accuracy



(b) Cost-based Distance Measure (m_{cd})



(c) r_{int} Coefficient

Figure 4: **Comparison Study Results.** Box plots of the performance using (a) balanced accuracy, (b) cost-based distance measure and (c) r_{int} coefficient as evaluation measure. See Tables 4.1-4.3 in the Supplementary Material for the mean and standard deviations of these results.

		NB	SVM	OVO	OVA	OPC	OSVM	CNB	COVO	POLR
balancedaccuracy	NB	-	**29.62 (0.000)**	0.00 (0.500)	-3.62 (1.000)	**18.18 (0.000)**	**65.13 (0.000)**	**65.81 (0.000)**	**193.00 (0.000)**	**247.14 (0.000)**
	SVM	-29.62 (1.000)	-	-29.52 (1.000)	-29.75 (1.000)	-21.75 (1.000)	**12.29 (0.000)**	**6.77 (0.000)**	**57.99 (0.000)**	**88.37 (0.000)**
	OVO	0.00 (0.500)	**29.52 (0.000)**	-	-3.63 (1.000)	**18.12 (0.000)**	**64.95 (0.000)**	**65.68 (0.000)**	**192.47 (0.000)**	**246.63 (0.000)**
	OVA	**3.62 (0.000)**	**29.75 (0.000)**	**3.63 (0.000)**	-	**27.26 (0.000)**	**65.99 (0.000)**	**69.74 (0.000)**	**192.10 (0.000)**	**252.95 (0.000)**
	OPC	-18.18 (1.000)	**21.75 (0.000)**	-18.12 (1.000)	-27.26 (1.000)	-	**59.32 (0.000)**	**60.06 (0.000)**	**198.36 (0.000)**	**263.99 (0.000)**
	OSVM	-65.13 (1.000)	-12.29 (1.000)	-64.95 (1.000)	-65.99 (1.000)	-59.32 (1.000)	-	-21.07 (1.000)	**96.51 (0.000)**	**160.99 (0.000)**
	CNB	-65.81 (1.000)	-6.77 (1.000)	-65.68 (1.000)	-69.74 (1.000)	-60.06 (1.000)	**21.07 (0.000)**	-	**214.65 (0.000)**	**345.42 (0.000)**
	COVO	-193.00 (1.000)	-57.99 (1.000)	-192.47 (1.000)	-192.10 (1.000)	-198.36 (1.000)	-96.51 (1.000)	-214.65 (1.000)	-	**174.79 (0.000)**
	POLR	-247.14 (1.000)	-88.37 (1.000)	-246.63 (1.000)	-252.95 (1.000)	-263.99 (1.000)	-160.99 (1.000)	-345.42 (1.000)	-174.79 (1.000)	-
m_{cd}	NB	-	**6.70 (0.000)**	-0.00 (0.500)	**8.56 (0.000)**	-37.76 (1.000)	-67.11 (1.000)	-86.64 (1.000)	-68.13 (1.000)	-50.11 (1.000)
	SVM	-6.70 (1.000)	-	-6.69 (1.000)	-5.30 (1.000)	-15.07 (1.000)	-27.80 (1.000)	-31.20 (1.000)	-25.35 (1.000)	-20.35 (1.000)
	OVO	0.00 (0.500)	**6.69 (0.000)**	-	**8.58 (0.000)**	-37.74 (1.000)	-66.91 (1.000)	-86.33 (1.000)	-67.90 (1.000)	-49.96 (1.000)
	OVA	-8.56 (1.000)	**5.30 (0.000)**	-8.58 (1.000)	-	-55.30 (1.000)	-67.86 (1.000)	-84.31 (1.000)	-67.67 (1.000)	-51.51 (1.000)
	OPC	**37.76 (0.000)**	**15.07 (0.000)**	**37.74 (0.000)**	**55.30 (0.000)**	-	-47.26 (1.000)	-58.65 (1.000)	-39.10 (1.000)	-21.11 (1.000)
	OSVM	**67.11 (0.000)**	**27.80 (0.000)**	**66.91 (0.000)**	**67.86 (0.000)**	**47.26 (0.000)**	-	-4.77 (1.000)	**36.81 (0.000)**	**73.63 (0.000)**
	CNB	**86.64 (0.000)**	**31.20 (0.000)**	**86.33 (0.000)**	**84.31 (0.000)**	**58.65 (0.000)**	**4.77 (0.000)**	-	**60.74 (0.000)**	**113.32 (0.000)**
	COVO	**68.13 (0.000)**	**25.35 (0.000)**	**67.90 (0.000)**	**67.67 (0.000)**	**39.10 (0.000)**	-36.81 (1.000)	-60.74 (1.000)	-	**123.08 (0.000)**
	POLR	**50.11 (0.000)**	**20.35 (0.000)**	**49.96 (0.000)**	**51.51 (0.000)**	**21.11 (0.000)**	-73.63 (1.000)	-113.32 (1.000)	-123.08 (1.000)	-

(a) Synthetic I

		NB	SVM	OVO	OVA	OPC	OSVM	CNB	COVO	POLR
balancedaccuracy	NB	-	**6.60 (0.000)**	-Inf (1.000)	**1.85 (0.036)**	**46.85 (0.000)**	**35.31 (0.000)**	**45.59 (0.000)**	**803.35 (0.000)**	**608.87 (0.000)**
	SVM	-6.60 (1.000)	-	-6.60 (1.000)	-5.79 (1.000)	**19.83 (0.000)**	**21.75 (0.000)**	**13.28 (0.000)**	**471.36 (0.000)**	**371.27 (0.000)**
	OVO	-Inf (1.000)	**6.60 (0.000)**	-	**1.85 (0.036)**	**46.85 (0.000)**	**35.31 (0.000)**	**45.59 (0.000)**	**803.35 (0.000)**	**608.87 (0.000)**
	OVA	-1.85 (0.964)	**5.79 (0.000)**	-1.85 (0.964)	-	**57.58 (0.000)**	**32.86 (0.000)**	**37.75 (0.000)**	**904.90 (0.000)**	**711.04 (0.000)**
	OPC	-46.85 (1.000)	-19.83 (1.000)	-46.85 (1.000)	-57.58 (1.000)	-	**9.53 (0.000)**	-17.38 (1.000)	**865.29 (0.000)**	**642.44 (0.000)**
	OSVM	-35.31 (1.000)	-21.75 (1.000)	-35.31 (1.000)	-32.86 (1.000)	-9.53 (1.000)	-	-18.40 (1.000)	**373.85 (0.000)**	**306.36 (0.000)**
	CNB	-45.59 (1.000)	-13.28 (1.000)	-45.59 (1.000)	-37.75 (1.000)	**17.38 (0.000)**	**18.40 (0.000)**	-	**792.90 (0.000)**	**721.26 (0.000)**
	COVO	-803.35 (1.000)	-471.36 (1.000)	-803.35 (1.000)	-904.90 (1.000)	-865.29 (1.000)	-373.85 (1.000)	-792.90 (1.000)	-	-137.26 (1.000)
	POLR	-608.87 (1.000)	-371.27 (1.000)	-608.87 (1.000)	-711.04 (1.000)	-642.44 (1.000)	-306.36 (1.000)	-721.26 (1.000)	**137.26 (0.000)**	-
m_{cd}	NB	-	**13.76 (0.000)**	-Inf (1.000)	**4.06 (0.000)**	**8.10 (0.000)**	**3.23 (0.001)**	-7.83 (1.000)	**406.09 (0.000)**	**248.43 (0.000)**
	SVM	-13.76 (1.000)	-	-13.76 (1.000)	-12.42 (1.000)	-7.60 (1.000)	-9.15 (1.000)	-16.95 (1.000)	**197.52 (0.000)**	**119.38 (0.000)**
	OVO	-Inf (1.000)	**13.76 (0.000)**	-	**4.06 (0.000)**	**8.10 (0.000)**	**3.23 (0.001)**	-7.83 (1.000)	**406.09 (0.000)**	**248.43 (0.000)**
	OVA	-4.06 (1.000)	**12.42 (0.000)**	-4.06 (1.000)	-	**5.70 (0.000)**	0.55 (0.293)	-10.04 (1.000)	**415.70 (0.000)**	**269.18 (0.000)**
	OPC	-8.10 (1.000)	**7.60 (0.000)**	-8.10 (1.000)	-5.70 (1.000)	-	-3.36 (0.999)	-14.45 (1.000)	**444.29 (0.000)**	**290.36 (0.000)**
	OSVM	-3.23 (0.999)	**9.15 (0.000)**	-3.23 (0.999)	-0.55 (0.707)	**3.36 (0.001)**	-	-7.18 (1.000)	**327.30 (0.000)**	**186.43 (0.000)**
	CNB	**7.83 (0.000)**	**16.95 (0.000)**	**7.83 (0.000)**	**10.04 (0.000)**	**14.45 (0.000)**	**7.18 (0.000)**	-	**426.30 (0.000)**	**318.63 (0.000)**
	COVO	-406.09 (1.000)	-197.52 (1.000)	-406.09 (1.000)	-415.70 (1.000)	-444.29 (1.000)	-327.30 (1.000)	-426.30 (1.000)	-	-183.65 (1.000)
	POLR	-248.43 (1.000)	-119.38 (1.000)	-248.43 (1.000)	-269.18 (1.000)	-290.36 (1.000)	-186.43 (1.000)	-318.63 (1.000)	**183.65 (0.000)**	-

(b) Synthetic II

Table 3. The t-values of a one-sided paired t-test between the performance of each classifier and all other classifiers on the datasets (a) synthetic I and (b) synthetic II. The corresponding p-values are shown between parentheses and those less than the significance level $\alpha = 0.05$ are in bold. When a p-value is in bold, the classifier listed in the row performs significantly better than the classifier listed in the column. See Table 4.5 of the Supplementary Material for the results of r_{int}.

interesting results of this comparison when using the ovarian dataset.

It seems that using a different evaluation measure to train a classifier with regard to testing does not have a big effect. The same experiment was applied to the other datasets giving similar results. It seemed that for a lot of permutations, the feature selection process often resulted in almost the same set of features, independent of the evaluation measure used. The feature selection process used in this study is simple and using a different evaluation measure might have more effect when using a more advanced feature selection process.

		Acc.	m_{cd}	r_{int}
SVM	Acc.	39.43% ± 2.13%	76.59% ± 1.81%	82.33% ± 1.18%
	m_{cd}	39.29% ± 1.81%	76.45% ± 1.56%	82.23% ± 1.06%
	r_{int}	39.27% ± 1.82%	76.44% ± 1.61%	82.22% ± 1.09%
OSVM	Acc.	38.85% ± 2.36%	76.59% ± 2.03%	82.35% ± 1.42%
	m_{cd}	39.29% ± 2.27%	76.51% ± 1.89%	82.28% ± 1.31%
	r_{int}	39.31% ± 2.09%	76.72% ± 1.82%	82.44% ± 1.25%

Table 4. The mean ± standard deviation of both SVM and OSVM trained on the ovarian dataset using different evaluation measures (Balanced accuracy, m_{cd}, r_{int}). The measures used to train the classifiers are listed in the rows, while the measures in the columns were used for evaluation.

3.3 Monotonicity

Ordinal methods like OSVM and POLR require the feature data to be correlated with the class labels, while for others, like OPC, CNB and COVO, it could help to improve performance. When features increase or decrease monotonously, they include information of the ordinal relation between the classes in comparison to features that just show high variance between classes.

To determine the effect of using monotonous features, all classifiers have been trained and evaluated using the two different feature selection methods discussed in Section 2.2. Figure 4 shows the results when using either the basic (red) or monotonous (blue) feature selection method.

There seems to be a very slight improvement in performance of the nominal classifiers when using monotonous feature selection over that when using basic feature selection. To confirm this difference, a one-sided paired t-test was applied where the performance of the monotonous results is matched with those of the basic results

for each permutation. The t-values with the corresponding p-values are shown in Table 5.

		t (p)		
		Prostate	Breast	Ovarian
Balanced Accuracy	NB	0.740 (0.231)	1.629 (0.055)	0.777 (0.221)
	SVM	-2.721 (0.996)	1.588 (0.059)	**2.239 (0.015)**
	OVO	1.550 (0.064)	0.701 (0.243)	**2.964 (0.002)**
	OVA	0.662 (0.255)	0.897 (0.187)	**2.712 (0.005)**
	OPC	1.398 (0.084)	-0.162 (0.564)	**3.037 (0.002)**
	OSVM	0.367 (0.358)	0.031 (0.488)	1.055 (0.148)
	CNB	-0.770 (0.777)	**3.028 (0.002)**	0.626 (0.267)
	COVO	**1.691 (0.049)**	**2.862 (0.003)**	**3.223 (0.001)**
m_{cd}	NB	**1.960 (0.028)**	1.377 (0.087)	**2.123 (0.019)**
	SVM	0.102 (0.459)	1.092 (0.140)	**4.675 (0.000)**
	OVO	**3.106 (0.002)**	0.652 (0.259)	**1.976 (0.027)**
	OVA	0.122 (0.452)	**1.895 (0.032)**	**2.969 (0.002)**
	OPC	0.035 (0.486)	-0.070 (0.528)	-2.251 (0.986)
	OSVM	0.045 (0.482)	0.096 (0.462)	1.461 (0.075)
	CNB	1.291 (0.101)	**2.745 (0.004)**	1.145 (0.129)
	COVO	**2.835 (0.003)**	**3.068 (0.002)**	**2.835 (0.003)**
r_{int}	NB	**2.019 (0.025)**	0.650 (0.259)	**2.157 (0.018)**
	SVM	-0.606 (0.726)	1.292 (0.101)	**6.012 (0.000)**
	OVO	**2.332 (0.012)**	0.774 (0.221)	**1.833 (0.036)**
	OVA	-0.803 (0.787)	1.657 (0.052)	**3.049 (0.002)**
	OPC	-0.649 (0.740)	-1.128 (0.868)	0.873 (0.193)
	OSVM	-0.890 (0.811)	-0.188 (0.574)	1.631 (0.055)
	CNB	1.439 (0.078)	**1.748 (0.043)**	1.035 (0.153)
	COVO	**2.104 (0.020)**	**2.974 (0.002)**	**4.773 (0.000)**

Table 5. The t-values of a one-sided paired t-test between the performance using the monotonous feature selection method and that from the basic feature selection method. The corresponding p-values are shown between parentheses and those less than the significance level $\alpha = 0.05$ are in bold. When a p-value is in bold, the classifier for that dataset performs significantly better when using the monotonous feature selection than when using the basic one.

With a significance level $\alpha = 0.05$, selecting monotonous features seems to be advantageous mostly in the ovarian dataset. This can be accounted to the fact that the ovarian dataset has 4 classes in comparison to only 3 in the other datasets. In other words, the ordering is not so much present and consequently does not have enough effect to influence the performance in the case of the prostate and breast datasets.

Looking at just the ovarian dataset, all nominal classifiers perform significantly better based on the ordinal measures m_{cd} and r_{int} when using monotonous feature selection. The ordinal methods on the other hand, do not seem to have this improvement, with the exception of COVO. By selecting strong monotonous features, the nominal classifiers automatically adopt the ordinal relation from the feature space. When not specifically selecting monotonous features, the ordinal classification methods are however able to enforce the ordinal relations between classes. As a result the ordinal classification methods show less improvement than the nominal ones from using monotonous features.

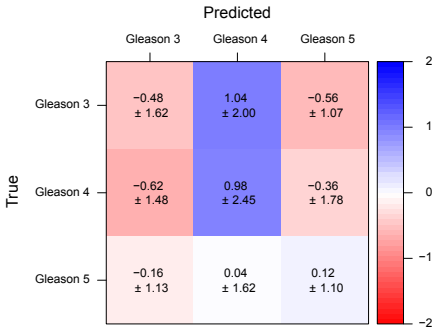


Figure 5: Difference in confusion matrix for COVO evaluated on the prostate dataset. For each permutation the confusion matrix from the results using basic feature selection was subtracted element-wise from the confusion matrix when using monotonous feature selection. With n permutations, this results in n matrices that represent the difference in the confusion of samples between both types of feature selection methods. Each element in the colored matrix in this figure shows the mean $\pm$ standard deviation of that element over the n permutations. An element i, j with a positive mean (blue) would indicate that samples of class i are more often predicted as class j when using monotonous feature selection, than with basic feature selection. See Figure 4.2 of the Supplementary Material for the average confusion matrices of the prediction results for both basic as monotonous feature selection.

As a positive side effect, the nominal classifiers SVM, OVO and OVA also seem to perform significantly better when looking at the balanced accuracy. This could be a result of those classifiers, being able to discriminate better between the classes when using the selected monotonous features. There is not enough supporting evidence, indicating that this a structural improvement from using monotonous features.

In all three datasets and for all evaluation measures, COVO performs significantly better when using monotonous feature selection than when using basic feature selection. Confusion matrices of the prediction results were inspected to determine the cause of the better performance. See Figure 5 for a matrix that shows the difference in confusion between both types of feature selection methods. From this figure it can be concluded that for class Gleason 5, it does not matter whether monotonous features were selected or not. The samples from the other two classes are influenced by the feature selection method, where more samples are pushed to class Gleason 4 when using monotonous features. This means more samples from class 4 are classified as 4 and less samples from class 3 are classified as 5, resulting in an increase in the balanced accuracy, m_{cd} and r_{int}.

In Figure 4.1 of the Supplementary Material the differences in confusion matrices is displayed as well but for all real datasets.

3.4 Learning Curves

The number of available training samples can have a large influence on the performance of a classifier. To measure this influence, learning curves [27] can be used, which plot the evaluation measure (for example accuracy) against the number of training samples. In other words, a classifier is trained using an increasing number of samples and evaluated against the same test set. The resulting plot can give some indication of the minimum amount of training samples needed to reach a certain result.

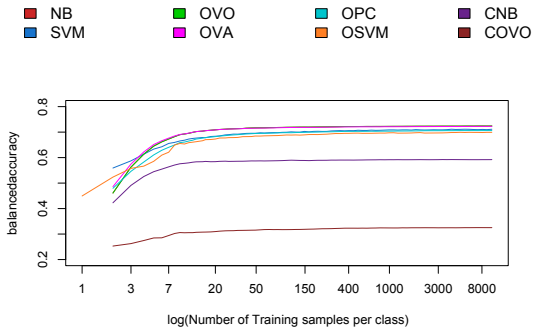
There are some drawbacks to this approach as it originates from a research area where the number of samples are abundant and the number of classes are typically limited to one or two. In the problems discussed here, the number of samples is limited, so after splitting the data set into both a training and test set there are not many samples left to generate the learning curve. In addition there are multiple classes, which leads to the question on how to increase the number of samples as the results can be biased by the class of the sample picked. Consequently the learning curves were only generated for the synthetic datasets as the number of samples are not limited. The number of training samples used, is increased each time by picking an additional sample from each class.

Additional samples were generated for the synthetic datasets from Figure 3 totaling up to 10.000 samples in each class. A learning curve was created using 90% of the samples for training while using the remainder for validation purposes. This process was permuted 100 times and the resulting curves were averaged to obtain the final curves. Three learning curves of the Synthetic II dataset are displayed in Figure 6 and the remaining curves can be found in the Supplementary Material in Figures 4.3-4.4.

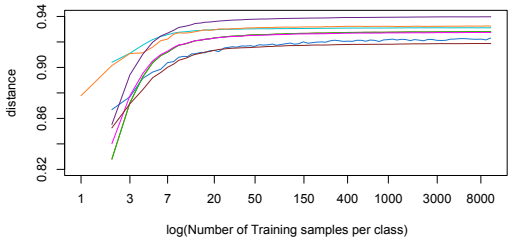
The curve of OSVM starts before the others which can be accounted to the fact that it replicates the samples and thus has more samples to work with from the beginning. As seen in the comparative study COVO does not perform well on the Synthetic II dataset, where it has a very low balanced accuracy rate while having both ordinal evaluation measures more up to par with the other classifiers.

The ordinal methods OSVM and OPC have a less steep m_{cd} learning curve and seem to require fewer samples than the other classifiers.

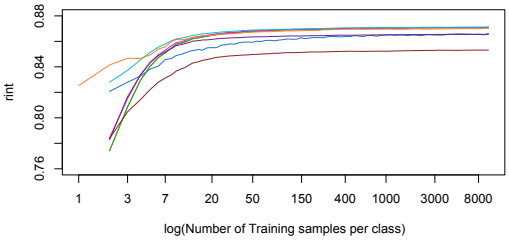
The classifiers do not improve a lot once above 20 samples per class. Each curve still improves slowly, but it is apparent that the limits of the dataset has been reached.



(b) Synthetic II (Balanced Accuracy)



(c) Synthetic II (m_{cd})



(d) Synthetic II (r_{int})

Figure 6: **Synthetic II Learning Curves.**The learning curves of the classifiers trained on Synthetic II dataset using (a) balanced accuracy, (b) cost-based distance measure and (c) r_{int}.

4 DISCUSSION

When using ordinal classification on real datasets, there are several issues to keep in mind. Depending of the type of tumor, a specific staging or grading system is used to assign one of the classes along the ordinal scale. Most of these systems require specialists to determine the label of each sample, but only after the patient is deceased the true label can be determined. This may lead to a rather subjective labeling of the dataset which can drastically influence the use of such data.

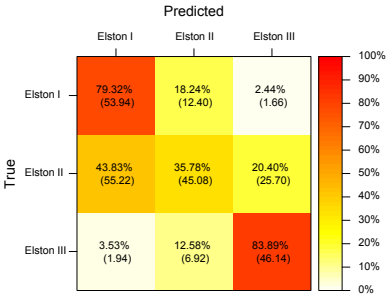
In addition, some systems like Elston grading (breast) only have three classes, which minimizes the possible benefits ordinal classification might have. Other systems like the Gleason score (prostate) have more classes, but only have a limited amount of samples of the lowest and highest class and for some classes even none. This can be accounted to the time of diagnosis, as the patient often already is past the first few stages or deceased before the final stage of the tumor. As a result, datasets can have fewer usable classes than the grading or staging system would allow as some classes have to be merged or omitted.

The balanced accuracy scores of the classifiers are a lot better on the breast dataset in comparison to the other real datasets, which seems to indicate that the classes are easier to separate. Taking the averaged confusion matrix of NB as an example (see Figure 7a); it seems that most samples can be found either on or right next to the diagonal of the confusion matrix, suggesting it is only difficult to separate each class from its neighboring classes. Consequently, the improvement an ordinal classifier could offer will be limited, as the predicted labels of most samples are next to their true label.

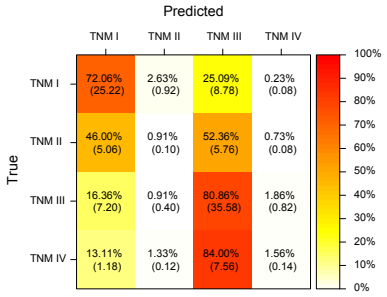
The ovarian dataset consists of classes 1 to 4 which have $35, 11, 44, 9$ samples respectively. Classes 2 and 4 are underrepresented in comparison to the other classes. Most of the classifiers are not very good at dealing with such imbalance as can be seen for OSVM in Figure 7b. Because of the imbalance, the classifier prefers classes 1 and 3 over the other two most of the times, explaining the two white columns. OSVM was however able to force most of the samples close to the true class and thus still achieves a performance comparable to the other classifiers. OPC seems to be the least affected by the class imbalance, although it is still noticeable as can be seen in Figure 7c.

Another issue comes into play, as a result of the selected parameters used for the evaluation protocol. The data has to be split up two times; first into 3 folds in the outer loop and then into 10 folds in the inner loop. In other words during the feature selection process in the evaluation protocol only 60% of the available samples can be used for training. When dealing with datasets such as the ones included in this study, it might be better to use fewer folds in the inner cross-validation loop instead of the 10-folds used in this study. Leave-one-out cross-validation would be an option, as it would ensure more samples for training but at the cost of more variance in the results.

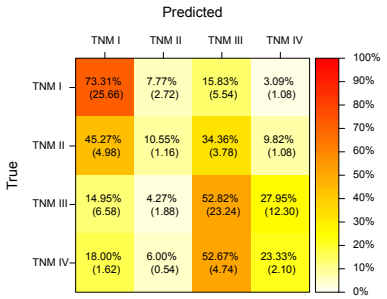
POLR has been successfully applied to ordinal classification in other fields of research. It is however used in a setting where samples are abundant and only a few variables come into play (and usually those variables are on a ordinal scale such as ratings given in a questionnaire). In molecular analysis however, POLR cannot always be applied directly, as there are typically not enough samples to base the model on.



(a) NB evaluated on the breast dataset using monotonous feature selection



(b) OSVM evaluated on the ovarian dataset using monotonous feature selection



(c) OPC evaluated on the ovarian dataset using monotonous feature selection

Figure 7: **Averaged confusion matrices.** The displayed confusion matrices are averaged over all permutations. The element i, j shows the average percentage of samples from class i that were predicted as class j, and the number between parentheses is the average number of samples.

5 CONCLUSION

At first sight, the results of this comparison study do not show much difference in performance between the studied classification methods. Using two synthetic datasets, it was possible to show the advantage of using an ordinal classifier. This advantage however, is rather limited for real datasets, which typically only have a small number of samples and classes. The limited number of samples is a common problem in molecular classification, as it makes it more difficult to separate the classes with the huge amount of features available. Over the years more effort is put into creating larger datasets, which would help minimize this limitation. The limited number of classes is on the other hand a more fundamental problem. For example, when using a staging or grading systems for cancers, there are effectively only three to five classes available. Ideally, there should be ten or more

classes, before the benefits of ordinal classification can be fully used.

From this study it can be concluded that a more simple approach is sufficient in comparison to a more experimental method specifically geared towards ordinal classification. By selecting features that correlate with the class labels, it was possible to improve the performance based on the ordinal evaluation measure. This would allow one to use a well known classifier such as the Naïve Bayes classifier and still perform well based on all evaluation measures. A rather simple feature selection procedure was used in this study, but future work could help in improving the selection of monotonous features.

CONTRIBUTION

In this paper a comparison study is presented, which benchmarks current ordinal classification methods in a molecular analysis setting. In general, previous work on this topic only studied ordinal classification methods on non-biological or synthetic datasets which do not correctly represent the kind of data found in molecular analysis.

These studies typically focus on the true and false positives and do not use a criterion with regards to the way samples were misclassified. For this purpose the cost-based distance measure was introduced, which helps to better identify the advantage of using ordinal classification.

The effect of selecting certain types of features in the context of ordinal classification is typically ignored. Using the cost-based distance measure, the potential benefit of using monotonous features was investigated. Based on experiments in this study, there does seem to be a benefit in selecting such features for both ordinal and nominal classifiers.

REFERENCES

[1]F. Greene, D. Page, and I. Fleming, *AJCC cancer staging handbook: from the AJCC cancer staging manual*. Springer, 6th ed., 2002.

[2]C. Dukes, "The classification of cancer of the colon," *Journal of Pathological Bacteriology*, vol. 35, pp. 323–332, 1932.

[3]D. Gleason, "Histologic grading and clinical staging of prostatic carcinoma," *Urologic Pathology: The Prostate*, pp. 171–197, 1977.

[4]L. True, I. Coleman, S. Hawley, C.-Y. Huang, D. Gifford, R. Coleman, T. M. Beer, E. Gelmann, M. Datta, E. Mostaghel, B. Knudsen, P. Lange, R. Vessella, D. Lin, L. Hood, and P. S. Nelson, "A molecular correlate to the gleason grading system for prostate adenocarcinoma.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, pp. 10991–10996, Jul 2006.

[5]L. D. Miller, J. Smeds, J. George, V. B. Vega, L. Vergara, A. Ploner, Y. Pawitan, P. Hall, S. Klaar, E. T. Liu, and J. Bergh, "An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival.," *Proc Natl Acad Sci U S A*, vol. 102, pp. 13550–13555, Sep 2005.

[6]N. D. Hendrix, R. Wu, R. Kuick, D. R. Schwartz, E. R. Fearon, and K. R. Cho, "Fibroblast growth factor 9 has oncogenic activity and is a downstream target of wnt signaling in ovarian endometrioid adenocarcinomas.," *Cancer Res*, vol. 66, pp. 1354–1362, Feb 2006.

[7]L. F. A. Wessels, M. J. T. Reinders, A. A. M. Hart, C. J. Veenman, H. Dai, Y. D. He, and L. J. vant Veer, "A protocol for building and evaluating predictors of disease state based on microarray data," *Bioinformatics*, vol. 21, p. 37553762, 2005.

[8]R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, vol. 14, pp. 1137–1145, 1995.

[9]U. Ganswindt, F. Paulsen, A. Anastasiadis, A. Stenzl, M. Bamberg, and C. Belka, "70 gy or more: which dose for which prostate cancer?," *Journal of Cancer Research and Clinical Oncology*, vol. 131, no. 7, pp. 407–419, 2005.

[10]J. F. P. da Costa, H. Alonso, and J. S. Cardoso, "The unimodal model for the classification of ordinal data.," *Neural Netw*, vol. 21, pp. 78–91, Jan 2008.

[11]R. Bellman, *Adaptive control processes*. Princeton University Press, 1961.

[12]R. Fisher, *Statistical methods for research workers*. Edinburgh London, 1925.

[13]R. Lomax, *Statistical concepts: A second course for education and the behavioral sciences*. Lawrence Erlbaum Associates, 2000.

[14]A. Sinha and H. Zhao, "Incorporating domain knowledge into data mining classifiers: An application in indirect lending," *Decision Support Systems*, vol. 46, no. 1, pp. 287–299, 2008.

[15]J. Rodgers and W. Nicewander, "Thirteen ways to look at the correlation coefficient," *American Statistician*, pp. 59–66, 1998.

[16]R. Duda, P. Hart, and D. Stork, *Pattern classification*. Wiley-Interscience, 2000.

[17]V. Vapnik *et al.*, *The nature of statistical learning theory*. Springer, 1995.

[18]K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *The Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.

[19]T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *The Annals of Statistics*, vol. 26, no. 2, pp. 451–471, 1998.

[20]D. Tax and R. Duin, "Using two-class classifiers for multi-class classification," *International Conference on Pattern Recognition*, 2002.

[21]R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[22]E. Frank and M. Hall, "A simple approach to ordinal classification," in *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, pp. 145–156, Springer, 2001.

[23]J. S. Cardoso, J. F. P. da Costa, and M. J. Cardoso, "Modelling ordinal relations with svms: an application to objective aesthetic evaluation of breast cancer

conservative treatment," *Neural Netw*, vol. 18, no. 5-6, pp. 808–817, 2005.

[24]S. Kotsiantis and P. Pintelas, "A cost sensitive technique for ordinal classification problems," in *SETN 2004: Methods and Applications of Artificial Intelligence*, Springer, 2004.

[25]P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society*, vol. 42, pp. 109–142, 1980.

[26]C. Elston, "Grading of invasive carcinoma of the breast," *Diagnostic histopathology of the breast*, pp. 300–311, 1987.

[27]A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, p. 437, 2000.

Ordinal Multi-Class Molecular Classification Supplementary Material

O.P. Pfeiffer

April 24, 2009

Contents

Chapter 1

Background Information

1.1 Severity of Cancers

When looking at the severity of cancers doctors look at two different aspects: its current spread and growth. The spread is specified by the stage of a cancer and indicates whether the cancer is confined to its organ of origin or has already spread to other organs. The growth is specified by the grade of a cancer and indicates how the likelihood of the cancer growing.

1.1.1 Staging Systems

Tumor, Node, Metastasis (TNM)

The TNM system uses three different criteria to help determine the severity of the cancer. For each criterion a score is assigned and when combined they describe the state of the cancer, see Table 1.1 for a brief description.

Score	Description
The degree of invasion of the intestinal wall	
T0	No evidence of tumor
Tis	Cancer in situ (tumor present, but no invasion)
T1-T4	Tumor size and the extent of spread.
The degree of lymphatic node involvement	
N0	No regional lymph nodes involved
N1-N4	The extent to which lymph nodes are involved.
The degree of metastasis	
M0	Metastasis absent
M1	Metastasis present

Table 1.1: Tumor, Node, Metastasis (TNM) system which can be applied to any type of cancer, although scores T1-T4 and N1-N4 depend on the type.

TNM Grouping

The American Joint Committee for Cancer Classification commonly groups the scores from the TNM system [1] into several stages (0-IV). See Table 1.2 for an overview of how the different scores are grouped together.

Stage	TNM equivalent
0	Tis, N0, M0
I	T1-T2, N0, M0
IIA	T3, N0, M0
IIB	T4, N0, M0
IIIA	T1-T2, N1, M0
IIIB	T3-T4, N1, M0
IIIC	Any T, N2, M0
IV	Any T, Any N, M1

Table 1.2: Grouping of the different TNM scores.

Dukes

Dukes' system [2] originally consisted of only three stages (A,B,C), but over the years several adaptations have been suggested. Turnbull et al.[3] adds a fourth stage so distant metastasis could be included (A,B,C,D), while Astler-Coller [4] separates both stage B and C into three different stages (A,B1-B3,C1-C3,D). The system for the colorectal dataset in this study uses the four stage system by Turnbull et al., see Table 1.3 for a brief description of the different stages.

Stage	Description	TNM equivalent
A	Tumor confined to the intestinal wall	Tis-T2, N0, M0
B	Tumor invading through the intestinal wall	T3, N0, M0
C	Tumor has spread outside the colon to one or more lymph nodes	T1-T4,N1,M0
D	Tumor with distant metastasis	Any T, Any N, M1

Table 1.3: Dukes' system.

1.1.2 Grading Systems

Grading is usually done by a pathologist who looks at the histological features of a tumor sample taken from a biopsy. The histological features include for example the degree of differentiation (i.e. how closely do the cancer cell look like the original ones), but also the size, shape and number of nuclei in the cancer cells. These features are highly dependent on the type of cancer so specific systems are needed for each type.

Gleason

The Gleason score [5] was developed to determine the grade of prostate cancer. The original system assigns a score between 1 and 5 based on the architectural patterns in the tissue sample, see

Table 1.4 for a brief description. Later on it appeared that the prostate cancer typically shows several patterns so now it is more common to assign the Gleason score to the primary and secondary pattern (i.e. the most common and second most common pattern seen in the tissue sample) and sum these values. As a result the combined Gleason score will range between 2 and 10.

Score	Description
1	Cells in tissue resemble original cells and the glands are still small and closely packed
2	Tissue still has well-formed glands although there is more space between them
3	Tissue still has recognizable glands and some cells have started invading surrounding tissue
4	Some glands in tissue are still recognizable and many cells are invading surrounding tissue
5	Glands in tissue are not recognizable

Table 1.4: Gleason's score.

1.2 Elston

The Elston system [6] is used to assess the histological grade of breast cancer. It uses three different parameters to obtain a final grade of either I,II or III. A short description of each of the parameters is given in Table 1.5. A pathologist assigns each property either $1, 2$ or 3 points, which are then summed. A total of $3-5$ points results in an Elston grade of I, $6-7$ points a grade of II and $8-9$ points a grade of III.

Parameter	Description
Tubules	Indicates how bad the tissue looks, by determining the cellular organization.
Nuclear Variation	Indicates how bad the cells look, by describing the state of the cells based on cytological properties.
Mitotic Activity	Describes the growth rate, by looking at the number of mitotic cells.

Table 1.5: The three different parameters used for the Elston grade.

Chapter 2

Datasets

2.1 Prostate (True et al.)

The dataset used by True et al. [7] was created using cDNA microarrays which compared tumor samples with normal samples. For each spot in each channel, the median background intensity was subtracted from the median foreground intensity. Afterwards the background-subtracted intensities of both channels were divided and $\log_2$ taken. Spots were removed when the median background intensity was greater than the median foreground intensity. Printtip-specific lowess curve [8] was used to normalize the log-ratio data over the different microarrays. The data set consist of 31 samples with 15488 features in total of which only 9491 features are present in all microarrays slides. The class labels can be summarized as:

Gleason's Grade	# samples
3	11
4	12
5	8
Total	31

URL: `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5132`

2.2 Breast (Miller et al.)

This dataset of breast tumors were created using two Affymetrix GeneChips (U133A/U133B) [9]. The data of both chips were normalized using the global mean method MAS5. The resulting values were $\ln$ transformed and scaled around the mean value of $\log(500)$. There are 253 samples with an Elston grade label and there are 44792 features. The class labels of the tumor samples can be summarized as:

Elston's Grade	# samples
I	68
II	128
III	55
Total	249

URL: `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4922`

2.3 Ovarian (Hendrix et al.)

This dataset was published by Hendrix et al. [10] and constructed using oligonucleotide microarrays. A quantile normalization procedure was performed to adjust for differences in the probe intensity distribution across different chips and finally the data was transformed by taking $\log_2$. The total dataset consists of 103 samples of which 99 are tumor samples and the other are normal. Of all available oligonucleotides the expression levels of 22283 genes are available. The class labels of the 99 tumor samples can be summarized as:

TNM Stage	# samples
1	5
1A	15
1C	15
2	1
2A	4
2B	1
2C	5
3	8
3B	1
3C	34
3D	1
4	9
Total	99

Stage Group	# samples
I	35
II	11
III	44
IV	9
Total	99

Grade	# Samples
1	25
2	26
2 or 3	5
3	43
Total	99

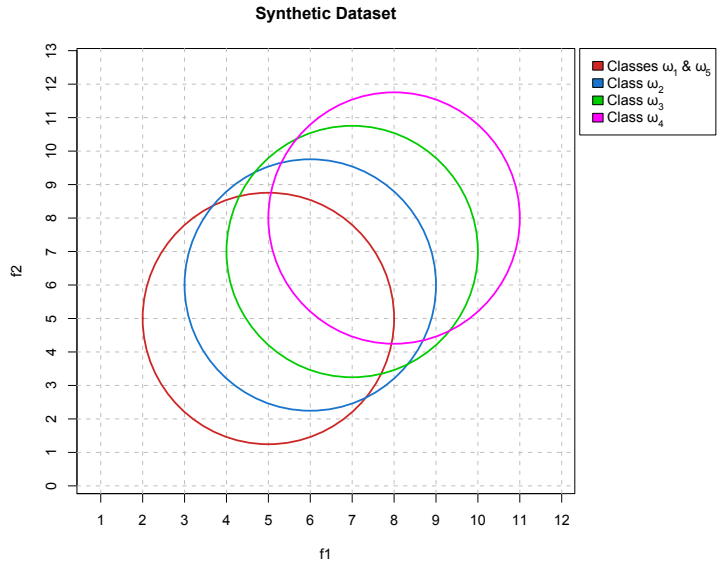
URL: `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE6008`

2.4 Synthetic I & II

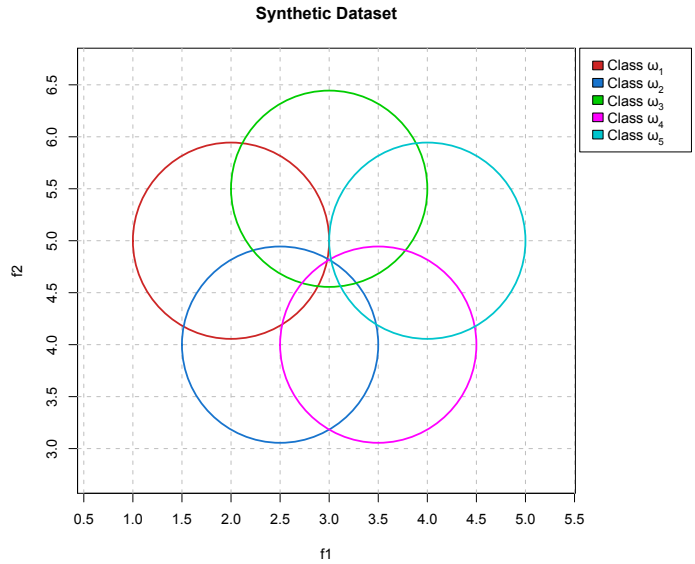
Both datasets consist of 500 samples in each class, totaling up to 2500 samples for the 5 classes generated. In the case of learning curves the number of samples per class was 10000. The samples are distributed normally $\mathcal{N}(\mu, \sigma^2)$; the means and standard deviations for each class can be found in the following list:

Class	Synthetic I			Synthetic II		
	μ_{f_1}	μ_{f_2}	σ	μ_{f_1}	μ_{f_2}	σ
1	5	5	1.5	2	5	0.5
2	6	6	1.5	2.5	4	0.5
3	7	7	1.5	3	5.5	0.5
4	8	8	1.5	3.5	4	0.5
5	5	5	1.5	4	5	0.5

Figure 2.1 shows the visual representation of the classes for both datasets. As there are only two features, no feature selection is required.



(a) Synthetic I



(b) Synthetic II

Figure 2.1: The datasets (a) Synthetic I and (b) Synthetic II. In each dataset the 5 classes have the same standard deviation: $\sigma = 1.5$ and $\sigma = 0.5$ for Synthetic I and Synthetic II respectively.

Chapter 3

Methods

3.1 Nominal Classification Methods

3.1.1 Naïve Bayes

The Naïve Bayes classifier uses a decision rule based on the Bayes' theorem with the additional assumption that all features are class-conditionally independent of each other. It is a simple method which works surprisingly well considering the strong assumptions on the independence. Because of this independence assumption only the samples of class k are needed to train the kth model (instead of using the whole training set), making the computation faster.

Using the Bayes' theorem the conditional probability ω_k given a set of p features $v_1, v_2, ...v_p$ can be written as shown in Equation 3.1.

$$P(\omega_k|v_1, v_2, ..., v_p) = \frac{P(\omega_k)P(v_1, v_2, ..., v_p|\omega_k)}{P(v_1, v_2, ..., v_p)} \qquad \text{for } k = 1, ...K \tag{3.1}$$

The probability $P(v_1, v_2, ..., v_p)$ is independent on the class labels and will be constant for each class label ω_k. As a result, it can be disregarded as only the numerator is important. The conditional probability $P(v_1, v_2, ..., v_p|\omega_k)$ is more difficult to estimate, but using the independence assumption it can be rewritten as in Equation 3.2:

$$P(v_1, v_2, ..., v_p|\omega_k) = \prod_{i=1}^{p} P(v_i|\omega_k) \tag{3.2}$$

By substituting the previous equation into Equation 3.1, the decision rule in Equation 3.3 can be obtained:

$$C(x) = \arg\max_k P(\omega_k|v_1, v_2, ..., v_p) = \arg\max_k P(\omega_k) \prod_{i=1}^{p} P(v_i|\omega_k) \tag{3.3}$$

The probability $P(\omega_k)$ and conditional probabilities $P(v_i|\omega_k)$ can be estimated by computing the relative frequency of each class from the training set.

3.1.2 Support Vector Machine

Support vector machines (SVM) [11] try to fit a hyperplane, defined by $w^T x + b = 0$, that optimally separates between classes. This is done by maximizing the distance between the hyperplane and

the support vectors of each class, where the support vectors are the samples that lie on the edge of the class distributions. By maximizing these distances, the support vector machine are expected to generalize better and thus should be able to more accurately classify any new unseen samples. Only the support vectors are needed, so constructing a support vector machine does not require the usage of all data samples to train an accurate classifier.

The support vector machines were originally developed for binary classification problems, which can be written as an optimization problem which needs to be maximized [12](written in quadratic form using Lagrange multipliers):

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \mathcal{K}(x_i, x_j) \tag{3.4}$$

$$\begin{aligned} \text{subject to } & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \qquad (i = 1, ..., n) \end{aligned}$$

Here $\mathcal{K}(x_i, x_j)$ is a kernel function (for example linear, polynomial or Guassian) which is used to transform the data into a higher dimensional space (allowing the hyperplane to be fit between more complex class boundaries). α is the variable being optimized which is used to fit the hyperplane. The classification rule can then be defined as:

$$C(x) = \begin{cases} \omega_1 & \text{if } f(x) > 0 \\ \omega_2 & \text{if } f(x) < 0 \end{cases} \tag{3.5}$$

$$\text{where} \quad f(x) = (\sum_{i=1}^{n} \alpha_i y_i \mathcal{K}(x, x_i)) + b \tag{3.6}$$

If $f(x)$ is equal to 0, then the class label is undefined and either class label can be assigned to the unknown sample arbitrarily.

The original SVM can also be generalized to multi-class classification problems. One way of generalizing involves creating a hyperplane for each class [12]. It is similar to the one-vs-all method discussed in Section 3.1.4 in the sense that there are K decision functions, but the optimization problem is adjusted to consider all decision functions at once. The SVM formulation in Equation 3.4 can be re-derived as the following optimization problem which needs to be minimized:

$$\sum_{i,j} \left[\frac{1}{2} c_j^{y_i} A_i A_j - \sum_{k=1}^{K} \alpha_i^k \alpha_j^{y_i} + \frac{1}{2} \sum_{k=1}^{K} \alpha_i^k \alpha_j^k \right] \mathcal{K}(x_i, x_j) - 2 \sum_{i,k} \alpha_i^k \tag{3.7}$$

$$\begin{aligned} \text{subject to } & \sum_{i=1}^{n} \alpha_i^k = \sum_{i=1}^{n} c_i^k A_i \\ & 0 \leq \alpha_i^m \leq C \\ & \alpha_i^{y_i} = 0 \\ & k = 1, ..., K \\ & m \in 1, ..., k \backslash y_i \\ & i = 1, ..., n \end{aligned}$$

Here $c_j^{y_i}$ is 1 if $y_i = y_j$ and 0 otherwise. The minimization in Equation 3.7 will result in the optimal α which can be used to determine w of each hyperplane $w^T x + b_k$. When classifying, the hyperplane that is furthest away from the sample determines the class label:

$$C(x) = \arg\max_k (\sum_{i=1}^{n} (c_i^k A_i - \alpha_i^k) \mathcal{K}(x_i, x) + b_k) \tag{3.8}$$

A disadvantage of SVMs is the fact they do not output posterior probabilities, but only return class labels instead. The posterior probabilities are needed for several purposes such as determining the confidence in the predicted class or combining several SVMs. There are a number of possible ways to calculate the posterior probabilities and one way in doing that is calculating the distance from the sample to the hyperplane [13]. The further a sample if from the hyperplane, the more confident a classifier is in whether the sample belongs to a certain assigned class.

3.1.3 One-vs-One

This approach, also known as all-vs-all or pairwise, constructs binary classifiers for each pair of classes. So in case there are K classes, then there are $\binom{K}{2} = \frac{K(K-1)}{2}$ classifiers needed. To classify an unknown sample, the output of all classifiers need to be combined and one way to do that is by voting. The most typical example of such method is majority voting, where the predicted classes from each classifier are counted. The class that is most represented, is the winner:

$$C(x) = \arg\max_k \sum_{i \neq k} \mathrm{I}\Big(P_{ki}(\lambda = k|x) \geq 0.5 \Big) \tag{3.9}$$

Here P_{ki} is the posterior probability output of the classifier that was trained to classify class k against class i. $\mathrm{I}(b)$ is 1 if b is true, and 0 otherwise.

The one-against-one approach can require some computation when the number of classes is high, as the number of classifiers needed is $\frac{K(K-1)}{2}$. However it has the advantage that on average fewer samples are required to train each classifier. So depending on which type of classifier is used, the increase in computation is compensated by the time that is needed to train each classifier.

Another type of voting is the winner-takes-all[14], also known as max-wins voting. In this case the posterior probabilities need to available, as the class with the highest probability from any classifier is the winner. This approach can result in bias towards specific classes when some classifiers are over-confident compared to the others. The corresponding classification rule is:

$$C(x) = \arg\max_k \sum_{i \neq k} \mathrm{I}\left(\frac{p_k}{p_k + p_i} > \frac{p_i}{p_k + p_i} \right) \tag{3.10}$$

Where p_i is the class posterior probabilities $P(\lambda = i|x)$ (p_k is the same probability but then for class k). $\mathrm{I}(b)$ is 1 if b is true, and 0 otherwise.

3.1.4 One-vs-All

In this approach, also known as one-vs-rest, K classifiers are constructed for each of the available classes. For the training of the ith classifier, the samples that belong to class i are used as positives and while all other samples are used as negatives. Each classifier outputs the posterior probability of its class being the correct one. The classifier that is most confident in its prediction will determine the label assigned to the unknown sample:

$$C(x) = \arg\max_k P_k(\lambda = k|x) \tag{3.11}$$

Where P_k is the posterior probability output of the classifier that was trained to classify class k. One disadvantage of this setup strategy is the high imbalance in the class sizes when training a classifier[15]. Although it is a rather simple approach, it seems to perform just as well as more advanced multi-class classifiers, making it a good alternative[16, 17].

3.2 Ordinal Classification Methods

3.2.1 Ordered Pseudo-Classification

In an article by Frank et al. [18] a method is proposed which tries to enforce the ordering by constructing the binary classifiers in a specific way. As a result this approach can be applied independent of the binary classifiers being used. $K-1$ binary classifiers are created which return the conditional probability $P(\omega_{\mathrm{T}} > \omega_i|X)$ where $i = 1, ..., K-1$ and ω_{T} is the true class of the unknown sample i. In other words classifier i determines the chance that the class of the unknown sample is higher than the class ω_i given X. In the case of four classes the following three binary classifiers will be constructed:

$$\begin{array}{lcl} \{\omega_1\} & \text{vs} & \{\omega_2, \omega_3, \omega_4\} \\ \{\omega_1, \omega_2\} & \text{vs} & \{\omega_3, \omega_4\} \\ \{\omega_1, \omega_2, \omega_3\} & \text{vs} & \{\omega_4\} \end{array}$$

The corresponding probabilities for each class can then be calculated as indicated in Equation 3.12.

$$P(\omega_k) = P(\omega_{\mathrm{T}} > \omega_{k-1}|X) - P(\omega_{\mathrm{T}} > \omega_k|X) \qquad \text{for } k = 1, ..., K \tag{3.12}$$

Here $P(\omega_{\mathrm{T}} > \omega_0|X) = 1$ and $P(\omega_{\mathrm{T}} > \omega_K|X) = 0$ as $y \in \{\omega_1, \omega_2, ..., \omega_K\}$. To determine the remaining conditional probabilities $P(\omega_{\mathrm{T}} > \omega_0|X)$ the Bayes' rule will be applied to Equation 3.12 which leads to the equation shown in Equation 3.13.

$$\begin{aligned} P(\omega_k) &= P(\omega_{\mathrm{T}} > \omega_{k-1}|X) - P(\omega_{\mathrm{T}} > \omega_k|X) \\ &= \frac{P(\omega_{\mathrm{T}} > \omega_{k-1})P(X|\omega_{\mathrm{T}} > \omega_{k-1})}{P(X)} - \frac{P(\omega_{\mathrm{T}} > \omega_k)P(X|\omega_{\mathrm{T}} > \omega_k)}{P(X)} \\ &= \frac{P(\omega_{\mathrm{T}} > \omega_{k-1})P(X|\omega_{\mathrm{T}} > \omega_{k-1}) - P(\omega_{\mathrm{T}} > \omega_k)P(X|\omega_{\mathrm{T}} > \omega_k)}{P(X)} \end{aligned} \tag{3.13}$$

Here $k = 1, ..., K$, while the prior probability and the likelihood should be estimated. The prior can be estimated using $P(\omega_Y > \omega_k) = \frac{n_k}{n}$, where n_k is the number of samples belonging to class ω_k and n is the total number of samples. The likelihood will be estimated by assuming a Gaussian distribution for each feature per each possible value of $\mathcal{Y}$, which will result in pK different Gaussian distributions. The evidence $P(X)$ is the same for all $P(\omega_k)$ as it does not depend on the ω_k, as a result Equation 3.13 can be simplified to the naïve Bayes approach where:

$$\begin{aligned} &\arg\max_k P(\omega_k) = \\ &\arg\max_k P(\omega_{\mathrm{T}} > \omega_{k-1})P(X|\omega_{\mathrm{T}} > \omega_{k-1}) - P(\omega_{\mathrm{T}} > \omega_k)P(X|\omega_{\mathrm{T}} > \omega_k) \end{aligned} \tag{3.14}$$

Modified Probability

Equation 3.12 assumes that the conditional probabilities are ordered (i.e. $P(\omega_{\mathrm{T}} > \omega_1|X) \leq P(\omega_{\mathrm{T}} > \omega_2|X) \leq ... \leq P(\omega_{\mathrm{T}} > \omega_K|X)$). Another approach would be to assume that $P(\omega_k) = P(\omega_{\mathrm{T}} > \omega_{k-1} \cap \omega_{\mathrm{T}} <= \omega_k|X)$. In that case the class probabilities can be obtained by determining the intersection of two binary classifiers:

$$\begin{aligned} P(\omega_k) &= P(\omega_{\mathrm{T}} > \omega_{k-1} \cap \omega_{\mathrm{T}} \leq \omega_k|X) \\ &= P(\omega_{\mathrm{T}} > \omega_{k-1}|X)P(\omega_{\mathrm{T}} \leq \omega_k|X) \qquad \text{for } k = 1, ..., K \\ &= P(\omega_{\mathrm{T}} > \omega_{k-1}|X)(1 - P(\omega_{\mathrm{T}} > \omega_k|X)) \end{aligned} \tag{3.15}$$

Where $P(\omega_{\mathrm{T}} > \omega_0|X) = 1$ and $P(\omega_{\mathrm{T}} > \omega_K|X) = 0$ as $y \in \{\omega_1, \omega_2, ..., \omega_K\}$ still hold. In this case there's an independence assumption which might not be very accurate in an ordinal setting.

3.2.2 Ordered-SVM

An article by Cardoso et al. [19] discusses a method in which the data is replicated so that the original multi-class problem can be solved as a binary problem. The advantage is that the well-developed binary SVM's can be used for multi-class classification problems.

The idea is based on the regular multi-class SVM methods in which $K - 1$ hyperplanes are constructed. These hyperplanes can be seen as $K - 1$ hyperplanes, but by replicating the samples this algorithm tries to solve this problem in one go. This is done by adding $K - 2$ dimensions to the feature space, so that the ordinal relation can be included into the problem. Figure 3.1 visually gives an example in the case of $K = 3$.

The data is replicated using Equation 3.16. Instead of creating $K-1$ hyperplanes, this equation replicates the data $K - 1$ times while adding the additional $K - 2$ dimensions.

$$\begin{bmatrix} x_i^{\omega_j} \\ e_{q-1} \end{bmatrix} \in \begin{cases} \bar{\omega}_1 & j = \max(1, q - s + 1), ..., q \\ \bar{\omega}_2 & j = q + 1, ..., \min(K, q + s) \end{cases} \qquad \text{for } q = 1, ..., K - 1 \tag{3.16}$$

Here $x_i^{\omega_j}$ are the samples that belong to class ω_j. e_0 is a sequence of $K - 2$ zeros, while e_{q-1} is a sequence of $K - 2$ symbols $0, ..., 0, h, 0, ..., 0$ with h in the position $q - 1$. The parameter s specifies the number of classes on both ways of the boundary that should be taken included (see Figure 3.2). The h parameter is used as a trade-off between the objectives of maximizing the margin of separation and minimizing the distance between the hyperplanes. It's not really clear whether the h parameter really has the stated effect, as it should not influence the results when all data is scaled with the same h.

Using $K = 4$ and $s = 2$ as an example, the data is replicated as shown in Equation 3.17.

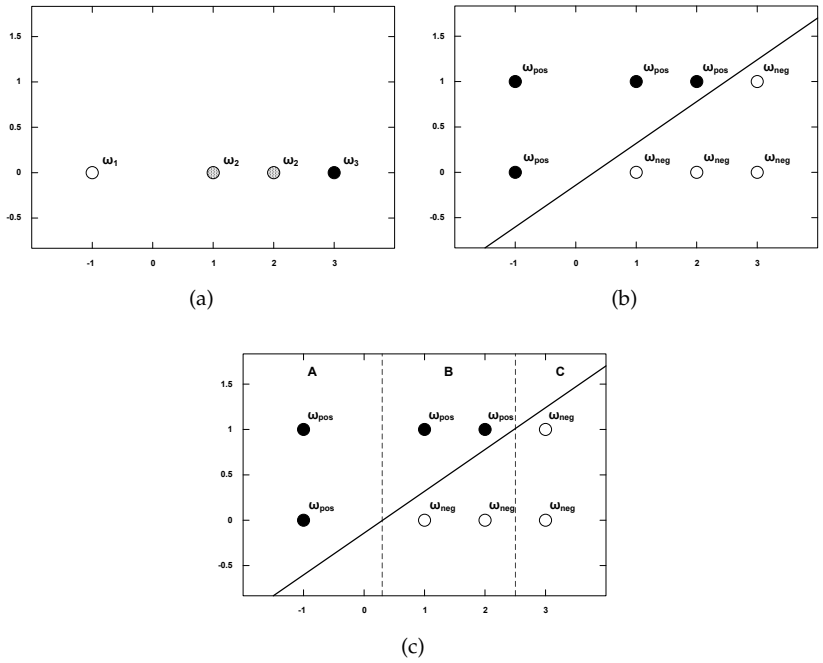


Figure 3.1: (a) 1 dimensional example with the original data for $K = 3$ in (a). The replicated data is shown as a binary problem in (b). In (c) the resulting solution to the original multi-class problem is shown. The dashed lines represent the class boundaries, where area A,B,C result in classes ω_1, ω_2, ω_3 respectively.

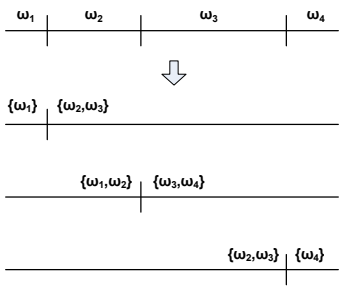


Figure 3.2: This figure shows how the original four classes can be divided using three different hyperplanes. The parameter s is used specify the number of classes on each side of the boundary. If $s = K - 1$ then all classes will be taken into account for each hyperplane. In the case of the example used in this figure $K = 4$ and $s = 2$.

$$
\begin{array}{rll}
 & \bar{\omega}_1: & \begin{bmatrix} x_i^{\omega_1} \\ 0 \\ 0 \end{bmatrix} \\
q=1: & & \\
 & \bar{\omega}_2: & \begin{bmatrix} x_i^{\omega_2} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_3} \\ 0 \\ 0 \end{bmatrix} \\
 & \bar{\omega}_1: & \begin{bmatrix} x_i^{\omega_1} \\ h \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_2} \\ h \\ 0 \end{bmatrix} \\
q=2: & & \\
 & \bar{\omega}_2: & \begin{bmatrix} x_i^{\omega_3} \\ h \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_4} \\ h \\ 0 \end{bmatrix} \\
 & \bar{\omega}_1: & \begin{bmatrix} x_i^{\omega_2} \\ 0 \\ h \end{bmatrix} \begin{bmatrix} x_i^{\omega_3} \\ 0 \\ h \end{bmatrix} \\
q=3 & & \\
 & \bar{\omega}_2: & \begin{bmatrix} x_i^{\omega_4} \\ 0 \\ h \end{bmatrix}
\end{array}
\tag{3.17}
$$

These samples can be combined to create the binary problem with the classes $\bar{\omega}_1$ and $\bar{\omega}_2$:

$$
\begin{array}{rl}
\bar{\omega}_1: & \begin{bmatrix} x_i^{\omega_1} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_1} \\ h \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_2} \\ h \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_2} \\ 0 \\ h \end{bmatrix} \begin{bmatrix} x_i^{\omega_3} \\ 0 \\ h \end{bmatrix} \\
\bar{\omega}_2: & \begin{bmatrix} x_i^{\omega_2} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_3} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_3} \\ h \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_4} \\ h \\ 0 \end{bmatrix} \begin{bmatrix} x_i^{\omega_4} \\ 0 \\ h \end{bmatrix}
\end{array}
\tag{3.18}
$$

Using the replicated data, the original multi-class problem can be solved using equation 3.19:

$$
\min_{w,b_i,\xi_i} \frac{1}{2} w'w + \frac{1}{h^2} \sum_{i=2}^{K-1} \frac{(b_i - b_1)^2}{2} + C \sum_{q=1}^{K-1} \sum_{j=\max(1,q-s+1}^{\min(K,q+s)} \sum_{i=1}^{n_j} \xi_{i,q}^{(j)} \tag{3.19}
$$

s.t.

$$
\begin{array}{rl}
-(w'x_i^{(j)} + b_1) \geq 1 - \xi_{i,1}^{(j)} & j = 1 \\
(w'x_i^{(j)} + b_1) \geq 1 - \xi_{i,1}^{(j)} & j = 2, ..., \min(k, 1+s) \\
\vdots & \\
-(w'x_i^{(j)} + b_q) \geq 1 - \xi_{i,q}^{(j)} & j = \max(1, q-s+1), ..., q \\
(w'x_i^{(j)} + b_q) \geq 1 - \xi_{i,q}^{(j)} & j = q+1, ..., \min(k, q+s) \\
\vdots & \\
-(w'x_i^{(j)} + b_{K-1}) \geq 1 - \xi_{i,K-1}^{(j)} & j = \max(1, K-s), ..., K-1 \\
(w'x_i^{(j)} + b_{K-1}) \geq 1 - \xi_{i,K-1}^{(j)} & j = K \\
\xi_{i,q}^{(j)} \geq 0 &
\end{array}
\tag{3.20}
$$

The boundary of the binary problem can then be mapped to the original space into $K-1$ boundaries:

$$b_i = \begin{cases} b & \text{if } i = 1 \\ hw_{p+i-1} + b & \text{if } i > 1 \end{cases} \tag{3.21}$$

Once the data has been replicated it can be used to train regular binary SVM. If an unknown sample needs to be classified, then that sample will be replicated as well (similarly as done in Equation 3.16). Each replica is then classified using the trained binary classifier, resulting in $K-1$ predictions $\in \{\bar{\omega}_1, \bar{\omega}_2\}$. It can be assumed that when the classes are truly ordered, the boundaries are ordered as well: $0 \geq hw_{p+1} \geq hw_{p+2} \geq ... \geq hw_{p+K-2}$. That means that the final prediction is equal to the number of replicas classified as $\bar{\omega}_2 + 1$.

One thing to consider with this method is that the used features should correlate with the increase of classes on the ordinal scale as much as possible. If not, the boundaries might not be ordered as assumed and could result in poor performance. A non-linear SVM might also help to make the classifier more robust.

3.2.3 Cost-based Methods

A different approach from developing a native ordinal classifier, is one which tries to use (mis) classification costs to include the ordinal relations when using nominal classifiers. The advantage of this approach that any well-developed classifier that returns posteriors can be used, although having to select the correct costs for each (mis) classification requires an additional configuration step as the optimal costs differ from one problem to another.

To create a cost-based version of classifier, a cost-matrix needs to be defined which will be used to assign costs to a misclassification based on distance between the predicted classes and the true class. In other words the bigger the distance, the higher the costs should be. The matrix below on the left side is the definition of a linear $K \times K$-cost-matrix being used and the one on the right is an example of such matrix for $K = 4$.

$$\begin{pmatrix} 0 & \cdots & k-1 \\ \vdots & \ddots & \vdots \\ k-1 & \cdots & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix} \tag{3.22}$$

Using the constructed cost-matrix M the risk of choosing class i can be determined using the following equation [20]:

$$R_j(x) = \sum_{i=1}^{K} m_{ij} P(\omega_i|x) \tag{3.23}$$

Here m_{ij} is the corresponding entry in the cost-matrix M. To classify a new sample the risk R_j is calculated for each class j and the one with the lowest risk is the winner. In other words by selecting the class with the lowest risk, the costs of misclassifying that sample is minimized. The way of constructing the cost matrix can be done several ways, but the matrix proposed in the original article was defined as shown in Equation 3.22.

3.2.4 Proportional Odds Logistic Regression

Regression methods return a response $\hat{y}$ on a continuous interval, where in the case of binary classification $\hat{y}$ is split into two discrete outputs using a specified threshold. This can be extended to the multi-class case by adding additional thresholds as indicated in Equation 3.24.

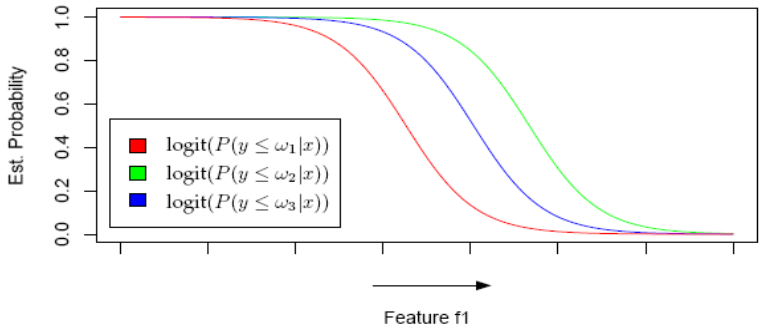


Figure 3.3: This figure illustrates a model trained for an example where $K = 4$. The β-coefficients are the same for each logit curve, although they have been shift to right by their α-threshold.

$$y_i \rightarrow \omega_k \qquad \text{if } \alpha_{k-1} \leq \hat{y}_i < \alpha_k \qquad \text{for } k = 1, ..., K \tag{3.24}$$

Here y_i is the class label of sample i and $\alpha_0, ..., \alpha_K$ are the thresholds separating the response $\hat{y}$ into K different classes. The two classes at both ends of the ordinal scale are defined by open-ended intervals where $\alpha_0 = -\infty$ and $\alpha_K = \infty$.

The model is known as the proportional odds logistic regression model [21], because the ratio of the odds of the event $y_i \leq \omega_k$ for any pair of sets of explanatory variables is independent of the choice of values for y. The model formulates the cumulative probabilities $P(y_i \leq \omega_k|x_i)$ as a latent variable $\hat{y}$ as summarized in Equation 3.25.

$$\text{logit}(\hat{y}) = \text{logit}(P(y_i \leq \omega_k|x_i)) = \log \frac{P(y_i \leq \omega_k|x_i)}{1 - P(y_i \leq \omega_k|x_i)} = \alpha_k + x_i\beta \tag{3.25}$$

Each logit$(P(y_i \leq \omega_k|x_i))$ has its own threshold α_k but shares the same coefficients β. Both the thresholds α_1, ..., α_k and coefficients β can be estimated by using for example maximum likelihood estimation or a more numerical approach like Newton's method. In this study the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method [22] is used for parameters estimation.

Once the parameters for the regression model have been estimated, the cumulative probabilities of an unknown sample x_i can then be calculated using Equation 3.26. See Figure 3.3 for an example where the cumulative probabilities are plotted.

$$\hat{y} = P(y_i \leq \omega_k|x_i) = \frac{1}{1 + e^{-(\alpha_k - x_i\beta)}} \qquad \text{for } k = 1, ..., K \tag{3.26}$$

These cumulative probabilities can be used to determine the probabilities of each class (see Equation 3.27).

$$P(y_i = \omega_k|x_i) = P(y_i \leq \omega_k|x_i) - P(y_i \leq \omega_{k-1}|x_i) \qquad \text{for } k = 1, ..., K \tag{3.27}$$

The class with the highest probability will be assigned to the unknown sample. The advantage of the proportional odds model is that the classes are separated without having to specify the thresholds a priori, but it requires the classes to be ordered in the feature space.

3.3 Feature Selection Methods

3.3.1 Analysis of Variance (ANOVA)

The one-way analysis of variance (ANOVA) [23, 24] can be used to test the equality of two or more classes. It is based on a couple of assumptions; the classes are a normally distributed, the variances of the class are equal and the samples are independent.

ANOVA tries to estimate the variance between the different classes and the variance within each class. The ratio of both variances has an F-distribution and can be used to determine how similar the different classes are.

The sum of squares between groups (SS_b) and within groups (SS_w) are used to describe the variance between the different classes and the variance within each class respectively. The definition of both sums of squares is given in Equation 3.28

Let the weighted average of the sample means be , then the sums of squares can be defined as:

$$\begin{aligned} \text{SS}_b &= \sum_{i=1}^{K} n_i (\bar{x}_i - \bar{X})^2 \\ \text{SS}_w &= \sum_{i=1}^{K} (n - K)\sigma_i^2 \end{aligned} \tag{3.28}$$

where

$$\bar{X} = \frac{\sum_{i=1}^{K} n_i \bar{x}_i}{\sum n_i} \tag{3.29}$$

Here n_i is the number of samples in class i, while $\bar{x}_i$ and σ_i are respectively the sample mean and standard deviation of that class. The ratio between both variances is referred to as the F-statistic and can be determined using the definition listed in Equation 3.30:

$$F = \frac{n - K}{K - 1} \frac{\text{SS}_b}{\text{SS}_w} \tag{3.30}$$

The larger the F-statistic is, the more separated the classes are. Once the F-statistic has been determined, the corresponding p-value can be obtained from the F-distribution. If the p-value is less than the significance level α, then the classes are separated from each other significantly.

3.3.2 Pearson's Correlation Coefficient

The Pearson's correlation coefficient [25], also referred to as Pearson product-moment correlation coefficient, is used to measure the linear correlation between two variables. The coefficient ranges from -1 to 1, where -1 would indicates that the variables are negatively correlated, while 1 would indicate a positive correlation. A value of 0 on the other hand would mean that there's no relation between both variables.

The Pearson's correlation coefficient can be defined as follows:

$$\rho(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{n \sigma_X \sigma_Y} \tag{3.31}$$

Where $\bar{X}$ and σ_X are the mean and standard deviation of the variable X respectively. The same definitions are used for $\bar{Y}$ and σ_Y but the variable Y. Using a feature as one variable and the

class labels as the other, the Pearson's correlation can be used to determine which features are monotonous or not. The higher the absolute value of the coefficient is for a certain feature, the more correlated it is.

3.4 Evaluation Methods

3.4.1 r_{int} rank coefficient

The article by La Costa et al. [26] also discusses ways to measure the performance of ordinal data classifiers. The Spearman's coefficient and Kendal's coefficient are mentioned and they propose their own measure r_{int}. The defined measure $_{\text{int}}$ tries compare any two ordinal variables. For each variable a set of ordered relations between each of the elements is constructed (i.e. the relations define which elements are higher than or equal to the others on the ordinal scale). In the case of two ordinal variables $a = o_1, o_2, o_3, o_4 = 1, 1, 2, 3$ and $b = o_1, o_2, o_3, o_4 = 2, 1, 2, 3$ their corresponding sets of ordered relations would be defined as follows:

$$S_a = \{(o_1, o_2), (o_1, o_3), (o_1, o_4), (o_2, o_1), (o_2, o_3), (o_2, o_4), (o_3, o_4)\} \tag{3.32}$$
$$S_b = \{(o_1, o_3), (o_1, o_4), (o_2, o_1), (o_2, o_3), (o_2, o_4), (o_3, o_1), (o_3, o_4)\} \tag{3.33}$$

To determine the similarity between both variables the subsets are used in the following equation:

$$r_{\text{int}} = -1 + 2\frac{\varsigma(S_1 \cap S_2)}{\sqrt{\varsigma(S_1)\varsigma(S_2)}} \tag{3.34}$$

Here S_1 and S_2 are the subsets of both ordinal variables and $\varsigma(X)$ is the function that returns the cardinality of the set X. The resulting r_{int} lies in the range $[-1, 1]$ where -1 would mean that both vectors are complete opposite and 1 would mean that they are identical. For the two example variables listed earlier, the intersection of both subsets consists of all pairs except $(o_1, o_2), (o_3, o_1)$ which would result in: $r_{\text{int}} = -1+2\frac{6}{\sqrt{7 \cdot 7}} = \frac{5}{7}$. This measure can be used where one ordinal vectors consists of the true class labels of all test samples and the other vector consists of the predicted class labels.

In the article a quicker way of computing r_{int} using the confusion matrix is discussed. This approach will be used to implement the r_{int} measure. One thing to note is that using r_{int} as a measure on its own can have some drawbacks. For example if the two ordinal vectors are $1, 2, 3, 4$ and $2, 3, 4, 5$ the resulting r_{int} will be 1. This would mean that the order within the elements of each vector are identical, however it does not mean that the samples were correctly classified.

3.4.2 Cost-Based Distance Measure

This method uses a cost matrix in combination with a confusion matrix to evaluate how close the classes of the predicted samples lie to their true classes. This is done by dividing the costs of the misclassified samples by the costs in case all samples were classified in the worst possible way. In the most general case a cost matrix that increases quadratically with the distance can be used. If a more conservative classifiers needs to constructed, an asymmetric cost matrix can be used instead.

The cost-based distance measure can be determined in two different ways: overall and balanced. It can be integrated into the evaluation protocol by calculating the cost-based distance

measure for each fold and averaging them. To illustrate both approaches an example is given in Equation 3.35 with the confusion matrix A of a trained classifier and M a cost matrix.

$$M = \begin{pmatrix} 0 & 1 & 4 & 9 \\ 1 & 0 & 1 & 4 \\ 4 & 1 & 0 & 1 \\ 9 & 4 & 1 & 0 \end{pmatrix}, A = \begin{pmatrix} 3 & 5 & 1 & 0 \\ 2 & 2 & 3 & 1 \\ 0 & 4 & 1 & 2 \\ 1 & 2 & 1 & 0 \end{pmatrix} \tag{3.35}$$

Overall

To determine the costs of the misclassifications, the confusion matrix A multiplied by the cost matrix M is summed. This would lead to the mislcassification costs $c = 42$. The worst case scenario can then determined by multiplying the maximum of each row in the cost matrix by the number of samples in each class. The sum of these values are the total worst-case costs $t = 177$. By dividing the misclassification costs c by the total cost b the overall cost-based distance m_{cd} is obtained. The overall cost-based distance measure can be defined as shown in Equation 3.37, which would result in $m_{\text{cd}} = 1 - \frac{42}{177} = 76.27\%$ for the given example.

More formally, given a confusion matrix A and a cost matrix M the m_{cd} can be defined as:

$$m_{\text{cd}}(M, A) = 1 - \frac{c}{t} \tag{3.36}$$

where c the actual total misclassification cost and t the maximal misclassification cost:

$$\begin{aligned} c &= \sum_{i=1}^{K} \sum_{j=1}^{K} M_{ij} A_{ij} \\ t &= \sum_{i=1}^{K} (\max_{j=1,..,K} M_{ij} \sum_{j=1}^{K} A_{ij}) \end{aligned} \tag{3.37}$$

Here A_{ij} is the number of samples of class i that were predicted as class j and M_{ij} is the cost of misclassifying a sample as class j while the true class is j. By default M is a linear cost matrix where the diagonal is equal to zero and the costs for misclassifying increases linearly with the distance to the true class; $M_{ij} = |i - j|$ for $i, j = 1, 2, ..., K$. When m_{cd} is equal to 0 then the classifier classified the samples in the worst way possible, while a value of 1 would indicate that the samples were classified perfectly.

Balanced

The balanced cost-based distance measure is quite similar, although the costs are determined for each class. In the case of the misclassification costs, the confusion matrix A is also multiplied by the cost matrix M. Although now the rows of the resulting matrix are summed which leads to the costs for each class. In the given example this would result in the misclassification costs $c = \{8, 7, 6, 8\}$. The worst case scenario can then determined by multiplying the maximum of each row in the cost matrix by the number of samples in each class. The resulting vector can be seen as the maximum possible cost for each class. In the given example this would result in the total costs $t = \{3, 2, 2, 3\}\{9, 8, 7, 4\} = \{27, 16, 14, 12\}$.

By dividing the misclassification costs c by the total cost b the balanced cost-based distances are obtained for each class. By averaging the class costs the overall costs of the classifier is given. The balanced cost-based distance measure $\bar{m}_{\text{cd}}$ can be defined as shown in Equations 3.38 and 3.39, which would result in $\bar{m}_{\text{cd}} = 1 - \frac{1}{4}(\frac{8}{27} + \frac{7}{16} + \frac{6}{14} + \frac{8}{12}) = 54.27\%$ for the given example.

First determine the actual total misclassification cost c and the maximal misclassification cost t:

$$\begin{aligned} c_i &= \sum_{j=1}^{K} M_{ij} A_{ij} \\ t_i &= (\max_{j=1,..,K} M_{ij}) \sum_{j=1}^{K} A_{ij} \\ &\text{for } i = 1, ..., K \end{aligned} \tag{3.38}$$

After that the balanced cost-based distance measure $\bar{m}_{\text{cd}}$ can be calculated as follows:

$$\bar{m}_{\text{cd}}(M, A) = 1 - \frac{1}{K} \sum_{i=1}^{K} (\frac{c_i}{t_i}) \tag{3.39}$$

Where M is a pre-defined cost matrix and A the confusion matrix of the classifier being evaluated. When $\bar{m}_{\text{cd}}$ is equal to 0 then the classifier classified the samples in the worst way possible, while a value of 1 would indicate that the samples were classified perfectly.

Chapter 4

Results

Additional figures and tables, which are referred to in the article, are listed here:

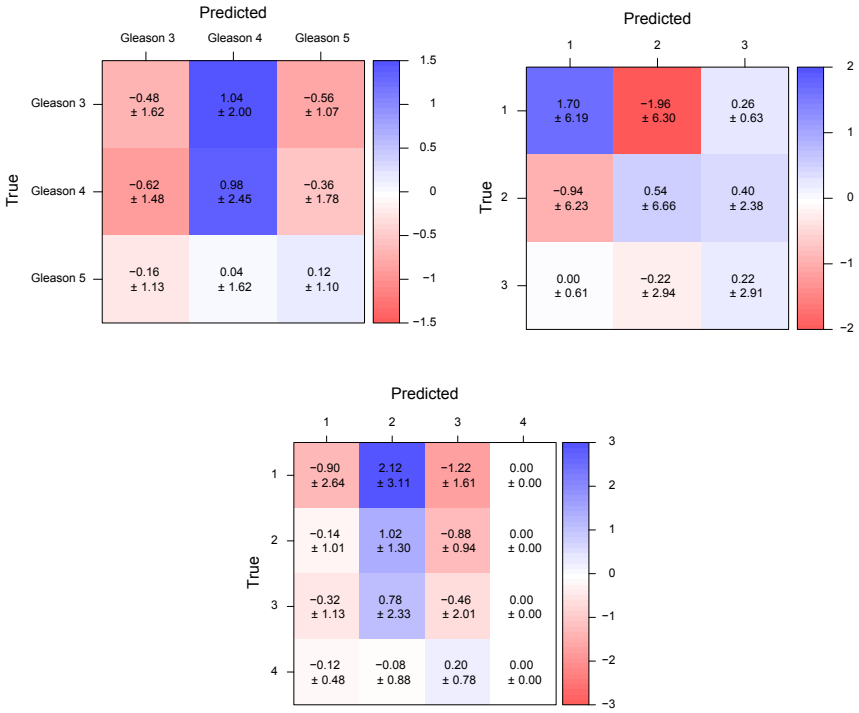


Figure 4.1: Difference in confusion matrices between basic and monotonous feature selection for COVO. Each colored matrix represents this difference in confusion matrices for the (a) prostate, (b) breast and (c) ovarian dataset. For each element, the mean and standard deviation over all permutations is shown. A positive mean (blue) for element i, j indicates more samples from class i were classified as class j when using monotonous feature selection. The same goes for a negative mean (red), but then for the basic feature selection.

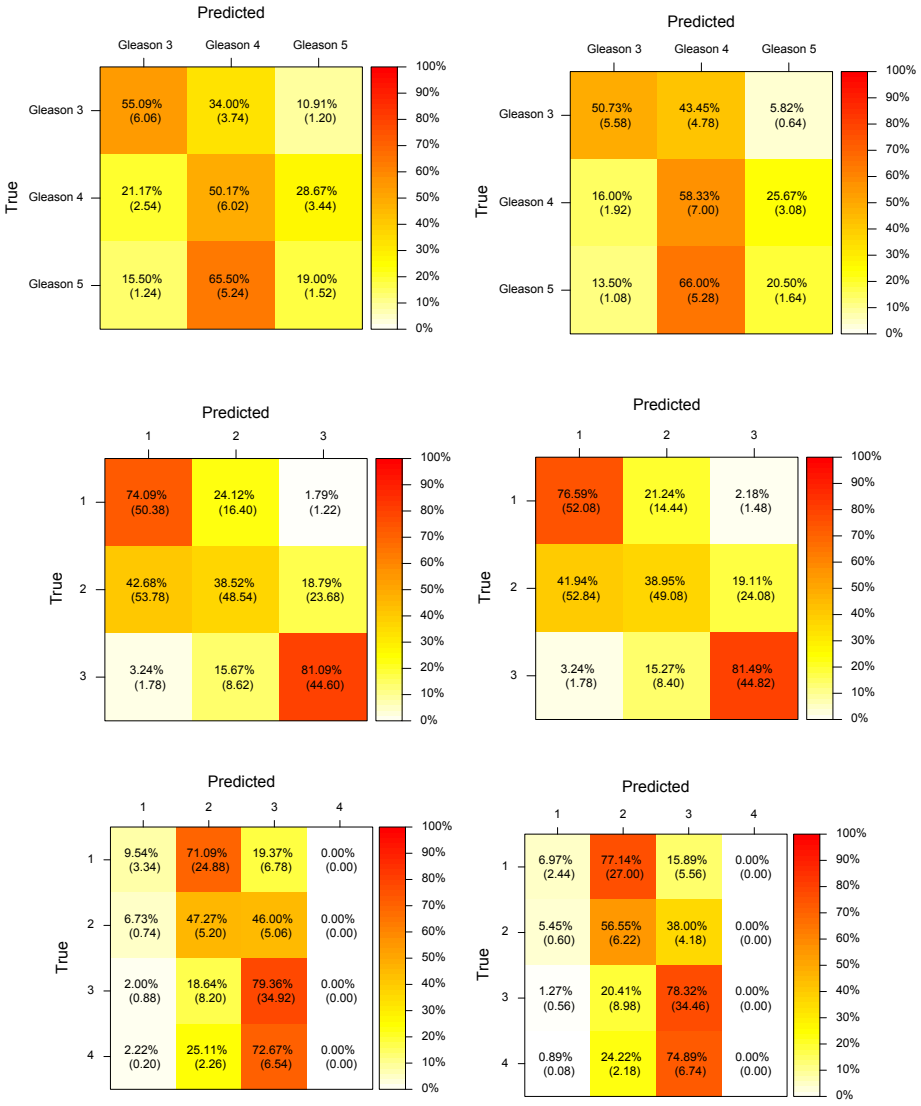
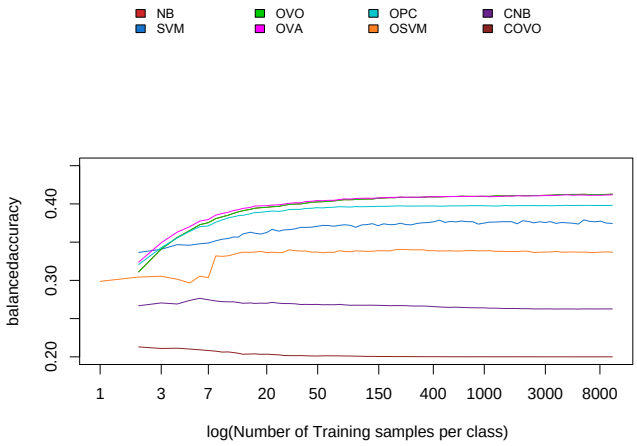
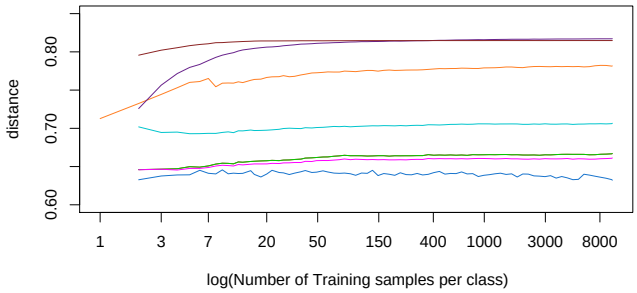


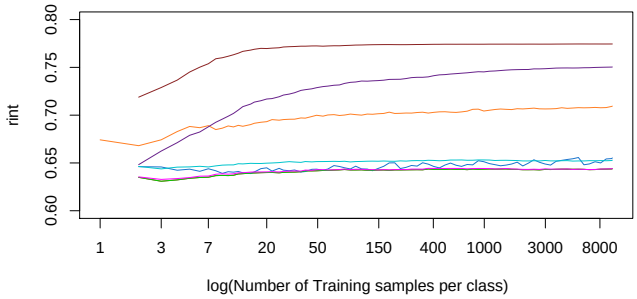
Figure 4.2: The confusion matrices averaged over all permutations for COVO evaluated on the (a) prostate, (b) breast and (c) ovarian dataset. On the left side the average confusion matrix is shown for the results using basic feature selection, while on the right side the matrix is shown when monotonous feature selection was used instead. Each element i, j shows the percentage of samples from class i that were classified as class j, while the number in parenthesis is the actual number of samples.



(b) Synthetic I (Balanced Accuracy)

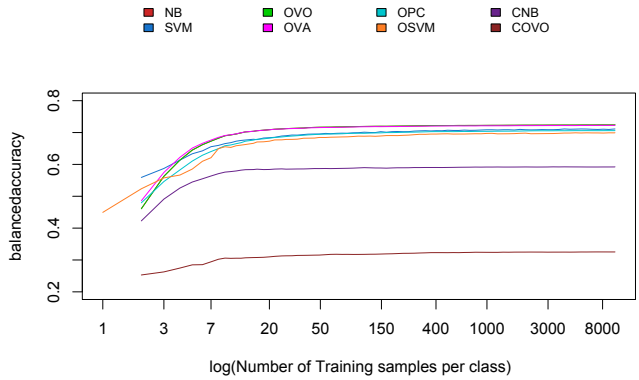


(c) Synthetic I (m_{cd} (Quadratic Cost-Matrix))

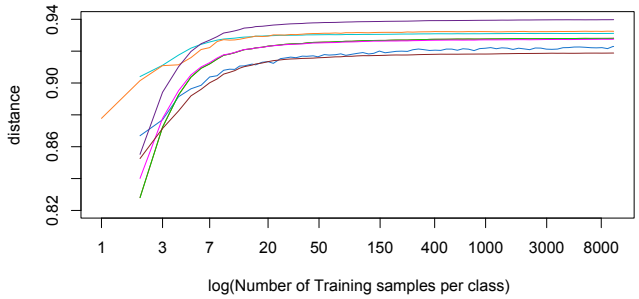


(d) Synthetic I (r_{int})

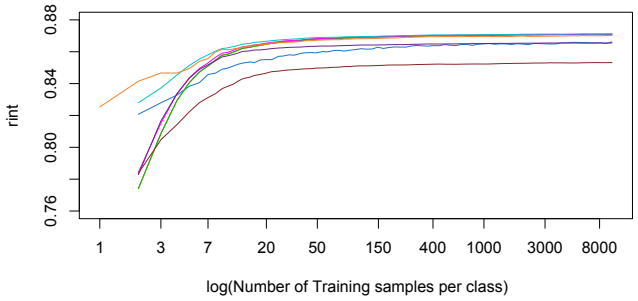
Figure 4.3: The learning curves of the classifiers trained on Synthetic I dataset using balanced accuracy (a), cost-based distance measure (b) and the r_{int} (c).



(b) Synthetic II (Balanced Accuracy)



(c) Synthetic II (m_{cd} (Quadratic Cost-Matrix))



(d) Synthetic II (r_{int})

Figure 4.4: The learning curves of the classifiers trained on Synthetic II dataset using balanced accuracy (a), cost-based distance measure (b) and the r_{int} (c).

		Monotonous Features		
		Prostate	Breast	Ovarian
Nominal	NB	43.88% ± 7.45%	66.33% ± 1.51%	39.40% ± 3.08%
	SVM	41.69% ± 7.14%	61.78% ± 2.43%	39.43% ± 2.13%
	OVO	43.22% ± 7.10%	65.63% ± 1.90%	39.62% ± 3.17%
	OVA	43.31% ± 7.06%	62.90% ± 2.61%	40.46% ± 2.69%
Ordinal	OPC	45.28% ± 8.53%	65.54% ± 1.13%	40.00% ± 3.64%
	OSVM	43.70% ± 7.42%	64.67% ± 2.51%	38.85% ± 2.36%
	CNB	43.51% ± 6.91%	66.25% ± 1.51%	39.60% ± 3.66%
	COVO	43.19% ± 7.04%	65.68% ± 2.05%	35.46% ± 3.38%

(a) Balanced accuracy (monotonous features)

		Non-Monotonous Features		
		Prostate	Breast	Ovarian
Nominal	NB	42.85% ± 7.31%	65.93% ± 1.13%	39.02% ± 2.87%
	SVM	44.32% ± 6.77%	61.00% ± 2.55%	38.42% ± 2.83%
	OVO	41.43% ± 7.52%	65.35% ± 2.02%	38.14% ± 3.08%
	OVA	42.57% ± 7.25%	62.47% ± 2.76%	39.15% ± 2.44%
Ordinal	OPC	43.30% ± 7.36%	65.58% ± 1.40%	38.03% ± 3.86%
	OSVM	43.25% ± 6.60%	64.65% ± 2.03%	38.45% ± 2.20%
	CNB	44.37% ± 6.88%	65.51% ± 1.53%	39.16% ± 2.93%
	COVO	41.42% ± 6.47%	64.57% ± 2.00%	34.04% ± 2.70%

(b) Balanced accuracy (Non-monotonous features)

		No feature selection	
		Synthetic 1	Synthetic 2
Nominal	NB	39.51% ± 0.48%	71.45% ± 0.19%
	SVM	34.79% ± 1.06%	71.10% ± 0.39%
	OVO	39.51% ± 0.48%	71.45% ± 0.19%
	OVA	39.70% ± 0.49%	71.41% ± 0.16%
Ordinal	OPC	38.36% ± 0.42%	69.88% ± 0.14%
	OSVM	32.41% ± 0.41%	69.22% ± 0.46%
	CNB	33.73% ± 0.25%	70.41% ± 0.15%
	COVO	25.92% ± 0.12%	43.65% ± 0.16%

(c) Balanced accuracy (no feature selection)

Table 4.1: The mean and standard deviations of the performance of all classifiers evaluated using the balanced accuracy rate

		Monotonous Features		
		Prostate	Breast	Ovarian
Nominal	NB	63.52% ± 5.49%	71.12% ± 0.99%	76.17% ± 2.18%
	SVM	62.68% ± 5.06%	72.02% ± 1.51%	76.59% ± 1.81%
	OVO	62.84% ± 5.39%	71.28% ± 1.13%	75.30% ± 2.46%
	OVA	60.68% ± 5.77%	71.13% ± 1.51%	77.02% ± 1.71%
Ordinal	OPC	63.16% ± 6.92%	68.85% ± 0.76%	72.74% ± 2.41%
	OSVM	62.00% ± 5.47%	72.05% ± 1.39%	76.59% ± 2.03%
	CNB	63.68% ± 5.53%	71.11% ± 1.04%	75.95% ± 2.12%
	COVO	63.00% ± 5.44%	71.43% ± 1.22%	73.41% ± 1.60%

(a) cdm (monotonous features)

		Non-Monotonous Features		
		Prostate	Breast	Ovarian
Nominal	NB	61.40% ± 5.72%	70.88% ± 0.89%	75.32% ± 2.26%
	SVM	62.60% ± 5.27%	71.67% ± 1.70%	74.95% ± 2.25%
	OVO	59.68% ± 6.12%	71.14% ± 1.10%	74.64% ± 2.43%
	OVA	60.56% ± 5.60%	70.63% ± 1.25%	76.07% ± 1.60%
Ordinal	OPC	63.12% ± 6.10%	68.87% ± 0.98%	73.73% ± 2.86%
	OSVM	61.96% ± 4.79%	72.03% ± 1.15%	76.16% ± 2.00%
	CNB	62.52% ± 5.88%	70.64% ± 1.00%	75.41% ± 2.23%
	COVO	60.32% ± 5.50%	70.84% ± 0.97%	72.79% ± 1.61%

(b) cdm (Non-monotonous features)

		No feature selection	
		Synthetic 1	Synthetic 2
Nominal	NB	59.01% ± 0.54%	85.87% ± 0.10%
	SVM	56.97% ± 2.05%	85.50% ± 0.20%
	OVO	59.01% ± 0.55%	85.87% ± 0.10%
	OVA	58.55% ± 0.60%	85.82% ± 0.09%
Ordinal	OPC	61.35% ± 0.51%	85.74% ± 0.07%
	OSVM	65.63% ± 0.24%	85.81% ± 0.14%
	CNB	65.82% ± 0.18%	85.95% ± 0.08%
	COVO	64.24% ± 0.06%	79.68% ± 0.07%

(c) Balanced accuracy (no feature selection)

Table 4.2: The mean and standard deviations of the performance for all classifiers evaluated using the cost-distance measure m_{cd}

		Monotonous Features		
		Prostate	Breast	Ovarian
Nominal	NB	76.13% ± 2.95%	83.25% ± 0.47%	81.38% ± 1.55%
	SVM	76.25% ± 2.58%	82.00% ± 0.73%	82.33% ± 1.18%
	OVO	75.62% ± 3.14%	83.06% ± 0.61%	80.73% ± 1.88%
	OVA	74.87% ± 3.27%	82.35% ± 0.77%	82.31% ± 1.22%
Ordinal	OPC	75.76% ± 3.87%	83.65% ± 0.42%	80.07% ± 1.24%
	OSVM	75.66% ± 3.17%	82.63% ± 0.78%	82.35% ± 1.42%
	CNB	76.30% ± 3.29%	83.21% ± 0.46%	81.12% ± 1.59%
	COVO	75.74% ± 3.24%	83.08% ± 0.62%	82.75% ± 1.30%

(a) cdm (monotonous features)

		Non-Monotonous Features		
		Prostate	Breast	Ovarian
Nominal	NB	74.87% ± 3.36%	83.20% ± 0.35%	80.75% ± 1.67%
	SVM	76.57% ± 3.19%	81.79% ± 0.82%	80.84% ± 1.62%
	OVO	74.10% ± 3.68%	82.96% ± 0.63%	80.26% ± 1.77%
	OVA	75.37% ± 3.00%	82.11% ± 0.75%	81.59% ± 1.17%
Ordinal	OPC	76.20% ± 3.60%	83.73% ± 0.42%	79.84% ± 1.84%
	OSVM	76.12% ± 2.90%	82.65% ± 0.61%	82.02% ± 1.38%
	CNB	75.47% ± 3.62%	83.07% ± 0.49%	80.76% ± 1.70%
	COVO	74.41% ± 3.44%	82.72% ± 0.63%	81.74% ± 1.67%

(b) cdm (Non-monotonous features)

		No feature selection	
		Synthetic 1	Synthetic 2
Nominal	NB	62.58% ± 0.51%	86.79% ± 0.08%
	SVM	62.86% ± 1.00%	86.55% ± 0.16%
	OVO	62.58% ± 0.51%	86.79% ± 0.08%
	OVA	62.60% ± 0.54%	86.83% ± 0.07%
Ordinal	OPC	63.56% ± 0.49%	86.82% ± 0.06%
	OSVM	69.89% ± 0.38%	86.68% ± 0.10%
	CNB	69.89% ± 0.30%	86.78% ± 0.06%
	COVO	71.21% ± 0.10%	85.78% ± 0.08%

(c) Balanced accuracy (no feature selection)

Table 4.3: The mean and standard deviations of the performance for all classifiers evaluated using the r_{int} coefficient.

		NB	SVM	OVO	OVA	OPC	OSVM	CNB	COVO
Balanced Accuracy	NB	-	-0.07 (0.527)	-0.49 (0.686)	-1.95 (0.971)	-1.04 (0.848)	1.15 (0.129)	-0.33 (0.629)	**6.94 (0.000)**
	SVM	0.07 (0.473)	-	-0.39 (0.651)	-2.15 (0.982)	-1.15 (0.873)	1.46 (0.075)	-0.31 (0.622)	**7.17 (0.000)**
	OVO	0.49 (0.314)	0.39 (0.349)	-	-1.50 (0.930)	-0.68 (0.751)	1.56 (0.063)	0.03 (0.486)	**7.26 (0.000)**
	OVA	**1.95 (0.029)**	**2.15 (0.018)**	1.50 (0.070)	-	0.75 (0.229)	**3.91 (0.000)**	1.56 (0.063)	**7.50 (0.000)**
	OPC	1.04 (0.152)	1.15 (0.127)	0.68 (0.249)	-0.75 (0.771)	-	**1.91 (0.031)**	0.66 (0.255)	**6.96 (0.000)**
	OSVM	-1.15 (0.871)	-1.46 (0.925)	-1.56 (0.937)	-3.91 (1.000)	-1.91 (0.969)	-	-1.34 (0.907)	**6.31 (0.000)**
	CNB:L	0.33 (0.371)	0.31 (0.378)	-0.03 (0.514)	-1.56 (0.937)	-0.66 (0.745)	1.34 (0.093)	-	**6.54 (0.000)**
	COVO:L	-6.94 (1.000)	-7.17 (1.000)	-7.26 (1.000)	-7.50 (1.000)	-6.96 (1.000)	-6.31 (1.000)	-6.54 (1.000)	-
m_{cd}	NB	-	-1.18 (0.879)	**2.47 (0.008)**	-2.71 (0.995)	**8.09 (0.000)**	-1.31 (0.902)	0.58 (0.281)	**10.20 (0.000)**
	SVM	1.18 (0.121)	-	**3.25 (0.001)**	-1.25 (0.891)	**10.32 (0.000)**	-0.00 (0.500)	1.68 (0.050)	**9.89 (0.000)**
	OVO	-2.47 (0.992)	-3.25 (0.999)	-	-5.29 (1.000)	**7.95 (0.000)**	-3.24 (0.999)	-1.90 (0.968)	**5.89 (0.000)**
	OVA	**2.71 (0.005)**	1.25 (0.109)	**5.29 (0.000)**	-	**11.14 (0.000)**	1.54 (0.065)	**3.28 (0.001)**	**12.13 (0.000)**
	OPC	-8.09 (1.000)	-10.32 (1.000)	-7.95 (1.000)	-11.14 (1.000)	-	-9.78 (1.000)	-7.85 (1.000)	-1.79 (0.960)
	OSVM	1.31 (0.098)	0.00 (0.500)	**3.24 (0.001)**	-1.54 (0.935)	**9.78 (0.000)**	-	1.55 (0.064)	**9.77 (0.000)**
	CNB:L	-0.58 (0.719)	-1.68 (0.950)	**1.90 (0.032)**	-3.28 (0.999)	**7.85 (0.000)**	-1.55 (0.936)	-	**7.51 (0.000)**
	COVO:L	-10.20 (1.000)	-9.89 (1.000)	-5.89 (1.000)	-12.13 (1.000)	**1.79 (0.040)**	-9.77 (1.000)	-7.51 (1.000)	-
r_{int}	NB	-	-3.83 (1.000)	**2.47 (0.009)**	-4.35 (1.000)	**5.16 (0.000)**	-4.37 (1.000)	0.96 (0.172)	-6.66 (1.000)
	SVM	**3.83 (0.000)**	-	**5.47 (0.000)**	0.10 (0.461)	**11.12 (0.000)**	-0.06 (0.526)	**4.43 (0.000)**	-1.87 (0.966)
	OVO	-2.47 (0.991)	-5.47 (1.000)	-	-6.60 (1.000)	**2.94 (0.003)**	-5.35 (1.000)	-1.48 (0.928)	-7.69 (1.000)
	OVA	**4.35 (0.000)**	-0.10 (0.539)	**6.60 (0.000)**	-	**10.39 (0.000)**	-0.19 (0.577)	**4.93 (0.000)**	-2.44 (0.991)
	OPC	-5.16 (1.000)	-11.12 (1.000)	-2.94 (0.997)	-10.39 (1.000)	-	-9.81 (1.000)	-3.98 (1.000)	-11.17 (1.000)
	OSVM	**4.37 (0.000)**	0.06 (0.474)	**5.35 (0.000)**	0.19 (0.423)	**9.81 (0.000)**	-	**3.93 (0.000)**	-1.63 (0.945)
	CNB:L	-0.96 (0.828)	-4.43 (1.000)	1.48 (0.072)	-4.93 (1.000)	**3.98 (0.000)**	-3.93 (1.000)	-	-6.64 (1.000)
	COVO:L	**6.66 (0.000)**	**1.87 (0.034)**	**7.69 (0.000)**	**2.44 (0.009)**	**11.17 (0.000)**	1.63 (0.055)	**6.64 (0.000)**	-

Table 4.4: The t-values of a one-sided paired t-test between the performance of each classifier and all other classifiers on the ovarian dataset. The corresponding p-values are shown between parentheses and those less than the significance level $\alpha = 0.05$ are in bold. When a p-value is in bold, the classifier listed in the row performs significantly better then the classifier listed in the column.

			NB	SVM	OVO	OVA	OPC	OSVM	CNB	COVO	POLR
Synthetic I	Balanced Accuracy	NB	-	**29.62 (0.000)**	0.00 (0.500)	-3.62 (1.000)	**18.18 (0.000)**	**65.13 (0.000)**	**65.81 (0.000)**	**193.00 (0.000)**	**247.14 (0.000)**
		SVM	-29.62 (1.000)	-	-29.52 (1.000)	-29.75 (1.000)	-21.75 (1.000)	**12.29 (0.000)**	**6.77 (0.000)**	**57.99 (0.000)**	**88.37 (0.000)**
		OVO	0.00 (0.500)	**29.52 (0.000)**	-	-3.63 (1.000)	**18.12 (0.000)**	**64.95 (0.000)**	**65.68 (0.000)**	**192.47 (0.000)**	**246.63 (0.000)**
		OVA	**3.62 (0.000)**	**29.75 (0.000)**	**3.63 (0.000)**	-	**27.26 (0.000)**	**65.99 (0.000)**	**69.74 (0.000)**	**192.10 (0.000)**	**252.95 (0.000)**
		OPC	-18.18 (1.000)	**21.75 (0.000)**	-18.12 (1.000)	-27.26 (1.000)	-	**59.32 (0.000)**	**60.06 (0.000)**	**198.36 (0.000)**	**263.99 (0.000)**
		OSVM	-65.13 (1.000)	-12.29 (1.000)	-64.95 (1.000)	-65.99 (1.000)	-59.32 (1.000)	-	-21.07 (1.000)	**96.51 (0.000)**	**160.99 (0.000)**
		CNB:L	-65.81 (1.000)	-6.77 (1.000)	-65.68 (1.000)	-69.74 (1.000)	-60.06 (1.000)	**21.07 (0.000)**	-	**214.65 (0.000)**	**345.42 (0.000)**
		COVO:L	-193.00 (1.000)	-57.99 (1.000)	-192.47 (1.000)	-192.10 (1.000)	-198.36 (1.000)	-96.51 (1.000)	-214.65 (1.000)	-	**174.79 (0.000)**
		POLR	-247.14 (1.000)	-88.37 (1.000)	-246.63 (1.000)	-252.95 (1.000)	-263.99 (1.000)	-160.99 (1.000)	-345.42 (1.000)	-174.79 (1.000)	-
	m_{cd}	NB	-	**6.70 (0.000)**	-0.00 (0.500)	**8.56 (0.000)**	-37.76 (1.000)	-67.11 (1.000)	-86.64 (1.000)	-68.13 (1.000)	-50.11 (1.000)
		SVM	-6.70 (1.000)	-	-6.69 (1.000)	-5.30 (1.000)	-15.07 (1.000)	-27.80 (1.000)	-31.20 (1.000)	-25.35 (1.000)	-20.35 (1.000)
		OVO	0.00 (0.500)	**6.69 (0.000)**	-	**8.58 (0.000)**	-37.74 (1.000)	-66.91 (1.000)	-86.33 (1.000)	-67.90 (1.000)	-49.96 (1.000)
		OVA	-8.56 (1.000)	**5.30 (0.000)**	-8.58 (1.000)	-	-55.30 (1.000)	-67.86 (1.000)	-84.31 (1.000)	-67.67 (1.000)	-51.51 (1.000)
		OPC	**37.76 (0.000)**	**15.07 (0.000)**	**37.74 (0.000)**	**55.30 (0.000)**	-	-47.26 (1.000)	-58.65 (1.000)	-39.10 (1.000)	-21.11 (1.000)
		OSVM	**67.11 (0.000)**	**27.80 (0.000)**	**66.91 (0.000)**	**67.86 (0.000)**	**47.26 (0.000)**	-	-4.77 (1.000)	**36.81 (0.000)**	**73.63 (0.000)**
		CNB:L	**86.64 (0.000)**	**31.20 (0.000)**	**86.33 (0.000)**	**84.31 (0.000)**	**58.65 (0.000)**	**4.77 (0.000)**	-	**60.74 (0.000)**	**113.32 (0.000)**
		COVO:L	**68.13 (0.000)**	**25.35 (0.000)**	**67.90 (0.000)**	**67.67 (0.000)**	**39.10 (0.000)**	-36.81 (1.000)	-60.74 (1.000)	-	**123.08 (0.000)**
		POLR	**50.11 (0.000)**	**20.35 (0.000)**	**49.96 (0.000)**	**51.51 (0.000)**	**21.11 (0.000)**	-73.63 (1.000)	-113.32 (1.000)	-123.08 (1.000)	-
	r_{rint}	NB	-	-1.79 (0.960)	-0.01 (0.504)	-0.32 (0.624)	-17.73 (1.000)	-73.78 (1.000)	-100.50 (1.000)	-117.50 (1.000)	-177.28 (1.000)
		SVM	**1.79 (0.040)**	-	**1.78 (0.040)**	**1.73 (0.045)**	-4.68 (1.000)	-46.35 (1.000)	-51.40 (1.000)	-59.45 (1.000)	-88.41 (1.000)
		OVO	0.01 (0.496)	-1.78 (0.960)	-	-0.32 (0.624)	-17.70 (1.000)	-73.71 (1.000)	-100.03 (1.000)	-117.01 (1.000)	-176.40 (1.000)
		OVA	0.32 (0.376)	-1.73 (0.955)	0.32 (0.376)	-	-21.58 (1.000)	-73.53 (1.000)	-94.94 (1.000)	-113.39 (1.000)	-170.56 (1.000)
		OPC	**17.73 (0.000)**	**4.68 (0.000)**	**17.70 (0.000)**	**21.58 (0.000)**	-	-68.46 (1.000)	-86.34 (1.000)	-106.51 (1.000)	-175.82 (1.000)
		OSVM	**73.78 (0.000)**	**46.35 (0.000)**	**73.71 (0.000)**	**73.53 (0.000)**	**68.46 (0.000)**	-	0.21 (0.419)	-22.61 (1.000)	-89.65 (1.000)
		CNB:L	**100.50 (0.000)**	**51.40 (0.000)**	**100.03 (0.000)**	**94.94 (0.000)**	**86.34 (0.000)**	-0.21 (0.581)	-	-31.39 (1.000)	-130.77 (1.000)
		COVO:L	**117.50 (0.000)**	**59.45 (0.000)**	**117.01 (0.000)**	**113.39 (0.000)**	**106.51 (0.000)**	**22.61 (0.000)**	**31.39 (0.000)**	-	-197.77 (1.000)
		POLR	**177.28 (0.000)**	**88.41 (0.000)**	**176.40 (0.000)**	**170.56 (0.000)**	**175.82 (0.000)**	**89.65 (0.000)**	**130.77 (0.000)**	**197.77 (0.000)**	-
Synthetic II	Balanced Accuracy	NB	-	**6.60 (0.000)**	-Inf (1.000)	**1.85 (0.036)**	**46.85 (0.000)**	**35.31 (0.000)**	**45.59 (0.000)**	**803.35 (0.000)**	**608.87 (0.000)**
		SVM	-6.60 (1.000)	-	-6.60 (1.000)	-5.79 (1.000)	**19.83 (0.000)**	**21.75 (0.000)**	**13.28 (0.000)**	**471.36 (0.000)**	**371.27 (0.000)**
		OVO	-Inf (1.000)	**6.60 (0.000)**	-	**1.85 (0.036)**	**46.85 (0.000)**	**35.31 (0.000)**	**45.59 (0.000)**	**803.35 (0.000)**	**608.87 (0.000)**
		OVA	-1.85 (0.964)	**5.79 (0.000)**	-1.85 (0.964)	-	**57.58 (0.000)**	**32.86 (0.000)**	**37.75 (0.000)**	**904.90 (0.000)**	**711.04 (0.000)**
		OPC	-46.85 (1.000)	-19.83 (1.000)	-46.85 (1.000)	-57.58 (1.000)	-	**9.53 (0.000)**	-17.38 (1.000)	**865.29 (0.000)**	**642.44 (0.000)**
		OSVM	-35.31 (1.000)	-21.75 (1.000)	-35.31 (1.000)	-32.86 (1.000)	-9.53 (1.000)	-	-18.40 (1.000)	**373.85 (0.000)**	**306.36 (0.000)**
		CNB:L	-45.59 (1.000)	-13.28 (1.000)	-45.59 (1.000)	-37.75 (1.000)	**17.38 (0.000)**	**18.40 (0.000)**	-	**792.90 (0.000)**	**721.26 (0.000)**
		COVO:L	-803.35 (1.000)	-471.36 (1.000)	-803.35 (1.000)	-904.90 (1.000)	-865.29 (1.000)	-373.85 (1.000)	-792.90 (1.000)	-	-137.26 (1.000)
		POLR	-608.87 (1.000)	-371.27 (1.000)	-608.87 (1.000)	-711.04 (1.000)	-642.44 (1.000)	-306.36 (1.000)	-721.26 (1.000)	**137.26 (0.000)**	-
	m_{cd}	NB	-	**13.76 (0.000)**	-Inf (1.000)	**4.06 (0.000)**	**8.10 (0.000)**	**3.23 (0.001)**	-7.83 (1.000)	**406.09 (0.000)**	**248.43 (0.000)**
		SVM	-13.76 (1.000)	-	-13.76 (1.000)	-12.42 (1.000)	-7.60 (1.000)	-9.15 (1.000)	-16.95 (1.000)	**197.52 (0.000)**	**119.38 (0.000)**
		OVO	-Inf (1.000)	**13.76 (0.000)**	-	**4.06 (0.000)**	**8.10 (0.000)**	**3.23 (0.001)**	-7.83 (1.000)	**406.09 (0.000)**	**248.43 (0.000)**
		OVA	-4.06 (1.000)	**12.42 (0.000)**	-4.06 (1.000)	-	**5.70 (0.000)**	0.55 (0.293)	-10.04 (1.000)	**415.70 (0.000)**	**269.18 (0.000)**
		OPC	-8.10 (1.000)	**7.60 (0.000)**	-8.10 (1.000)	-5.70 (1.000)	-	-3.36 (0.999)	-14.45 (1.000)	**444.29 (0.000)**	**290.36 (0.000)**
		OSVM	-3.23 (0.999)	**9.15 (0.000)**	-3.23 (0.999)	-0.55 (0.707)	**3.36 (0.001)**	-	-7.18 (1.000)	**327.30 (0.000)**	**186.43 (0.000)**
		CNB:L	**7.83 (0.000)**	**16.95 (0.000)**	**7.83 (0.000)**	**10.04 (0.000)**	**14.45 (0.000)**	**7.18 (0.000)**	-	**426.30 (0.000)**	**318.63 (0.000)**
		COVO:L	-406.09 (1.000)	-197.52 (1.000)	-406.09 (1.000)	-415.70 (1.000)	-444.29 (1.000)	-327.30 (1.000)	-426.30 (1.000)	-	-183.65 (1.000)
		POLR	-248.43 (1.000)	-119.38 (1.000)	-248.43 (1.000)	-269.18 (1.000)	-290.36 (1.000)	-186.43 (1.000)	-318.63 (1.000)	**183.65 (0.000)**	-
	r_{rint}	NB	-	**10.87 (0.000)**	-Inf (1.000)	-4.70 (1.000)	-2.32 (0.988)	**8.37 (0.000)**	0.88 (0.191)	**75.05 (0.000)**	**136.08 (0.000)**
		SVM	-10.87 (1.000)	-	-10.87 (1.000)	-13.46 (1.000)	-10.65 (1.000)	-5.09 (1.000)	-10.86 (1.000)	**31.49 (0.000)**	**59.65 (0.000)**
		OVO	-Inf (1.000)	**10.87 (0.000)**	-	-4.70 (1.000)	-2.32 (0.988)	**8.37 (0.000)**	0.88 (0.191)	**75.05 (0.000)**	**136.08 (0.000)**
		OVA	**4.70 (0.000)**	**13.46 (0.000)**	**4.70 (0.000)**	-	1.14 (0.130)	**9.74 (0.000)**	**5.04 (0.000)**	**74.14 (0.000)**	**155.32 (0.000)**
		OPC	**2.32 (0.012)**	**10.65 (0.000)**	**2.32 (0.012)**	-1.14 (0.870)	-	**8.84 (0.000)**	**3.27 (0.001)**	**74.23 (0.000)**	**178.30 (0.000)**
		OSVM	-8.37 (1.000)	**5.09 (0.000)**	-8.37 (1.000)	-9.74 (1.000)	-8.84 (1.000)	-	-7.36 (1.000)	**60.70 (0.000)**	**101.13 (0.000)**
		CNB:L	-0.88 (0.809)	**10.86 (0.000)**	-0.88 (0.809)	-5.04 (1.000)	-3.27 (0.999)	**7.36 (0.000)**	-	**73.51 (0.000)**	**170.80 (0.000)**
		COVO:L	-75.05 (1.000)	-31.49 (1.000)	-75.05 (1.000)	-74.14 (1.000)	-74.23 (1.000)	-60.70 (1.000)	-73.51 (1.000)	-	**56.92 (0.000)**
		POLR	-136.08 (1.000)	-59.65 (1.000)	-136.08 (1.000)	-155.32 (1.000)	-178.30 (1.000)	-101.13 (1.000)	-170.80 (1.000)	-56.92 (1.000)	-

Table 4.5: The t-values of a one-sided paired t-test between the performance of each classifier and all other classifiers on the synthetic I and II datasets. The corresponding p-values are shown between parentheses and those less than the significance level $\alpha = 0.05$ are in bold. When a p-value is in bold, the classifier listed in the row performs significantly better then the classifier listed in the column.

Bibliography

[1] F. Greene, D. Page, and I. Fleming, *AJCC cancer staging handbook: from the AJCC cancer staging manual*. Springer, 6th ed., 2002.

[2] C. Dukes, "The classification of cancer of the colon," *Journal of Pathological Bacteriology*, vol. 35, pp. 323–332, 1932.

[3] R. B. Turnbull, K. Kyle, F. R. Watson, and J. Spratt, "Cancer of the colon: the influence of the no-touch isolation technic on survival rates.," *Ann Surg*, vol. 166, pp. 420–427, Sep 1967.

[4] V. B. Astler and F. A. Coller, "The prognostic significance of direct extension of carcinoma of the colon and rectum.," *Ann Surg*, vol. 139, pp. 846–852, Jun 1954.

[5] D. Gleason, "Histologic grading and clinical staging of prostatic carcinoma," *Urologic Pathology: The Prostate*, pp. 171–197, 1977.

[6] C. Elston, "Grading of invasive carcinoma of the breast," *Diagnostic histopathology of the breast*, pp. 300–311, 1987.

[7] L. True, I. Coleman, S. Hawley, C.-Y. Huang, D. Gifford, R. Coleman, T. M. Beer, E. Gelmann, M. Datta, E. Mostaghel, B. Knudsen, P. Lange, R. Vessella, D. Lin, L. Hood, and P. S. Nelson, "A molecular correlate to the gleason grading system for prostate adenocarcinoma.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, pp. 10991–10996, Jul 2006.

[8] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed, "Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation.," *Nucleic Acids Res*, vol. 30, p. e15, Feb 2002.

[9] L. D. Miller, J. Smeds, J. George, V. B. Vega, L. Vergara, A. Ploner, Y. Pawitan, P. Hall, S. Klaar, E. T. Liu, and J. Bergh, "An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival.," *Proc Natl Acad Sci U S A*, vol. 102, pp. 13550–13555, Sep 2005.

[10] N. D. Hendrix, R. Wu, R. Kuick, D. R. Schwartz, E. R. Fearon, and K. R. Cho, "Fibroblast growth factor 9 has oncogenic activity and is a downstream target of wnt signaling in ovarian endometrioid adenocarcinomas.," *Cancer Res*, vol. 66, pp. 1354–1362, Feb 2006.

[11] V. Vapnik *et al.*, *The nature of statistical learning theory*. Springer, 1995.

[12] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," *Proceedings of the Seventh European Symposium On Artificial Neural Networks*, vol. 4, p. 6, 1999.

[13] A. Madevska-Bogdanova and D. Nikolic, "A new approach of modifying SVM outputs," *Proceedings of the International Joint Conference on Neural Networks*, vol. 6, pp. 395–398, 2000.

[14] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *The Annals of Statistics*, vol. 26, no. 2, pp. 451–471, 1998.

[15] C. Huang, L. Davis, and J. Townshend, "An assessment of support vector machines for land cover classification," *International Journal of Remote Sensing*, vol. 23, no. 4, pp. 725–749, 2002.

[16] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[17] C. H. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R. M. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov, and T. Golub, "Molecular classification of multiple tumor types.," *Bioinformatics*, vol. 17 Suppl 1, pp. S316–S322, 2001.

[18] E. Frank and M. Hall, "A simple approach to ordinal classification," in *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, pp. 145–156, Springer, 2001.

[19] J. S. Cardoso, J. F. P. da Costa, and M. J. Cardoso, "Modelling ordinal relations with svms: an application to objective aesthetic evaluation of breast cancer conservative treatment," *Neural Netw*, vol. 18, no. 5-6, pp. 808–817, 2005.

[20] S. Kotsiantis and P. Pintelas, "A cost sensitive technique for ordinal classification problems," in *SETN 2004: Methods and Applications of Artificial Intelligence*, Springer, 2004.

[21] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society*, vol. 42, pp. 109–142, 1980.

[22] M. Avriel, *Nonlinear programming: analysis and methods*. Dover Publications, 2003.

[23] R. Fisher, *Statistical methods for research workers*. Edinburgh London, 1925.

[24] R. Lomax, *Statistical concepts: A second course for education and the behavioral sciences*. Lawrence Erlbaum Associates, 2000.

[25] J. Rodgers and W. Nicewander, "Thirteen ways to look at the correlation coefficient," *American Statistician*, pp. 59–66, 1998.

[26] J. F. P. da Costa, H. Alonso, and J. S. Cardoso, "The unimodal model for the classification of ordinal data.," *Neural Netw*, vol. 21, pp. 78–91, Jan 2008.

Ordinal Multi-Class Molecular Classification
Work Document

O.P. Pfeiffer

April 24, 2009

Chapter 1

Problem Statement

1.1 Goal

In the standard approach of multi-class classification the classes are assumed to be distinct. If one would want to classify the progression of a disease, such as prostate cancer, the classes do however show order among the classes as the tumor progresses from a benign to a malignant stage. This thesis looks into which key properties characterize ordinal multi-class molecular classification problems and presents a comparison study of classification algorithms that can be used for this purpose.

1.2 Research Questions

Based on the goal several research questions were posed. This section covers each of those questions and the corresponding answers based on the results of this study:

- *Which objectives can be used to evaluate ordinal classifiers?*
  Two different criteria were identified to evaluate ordinal classifiers. The first criterion is the same as the one used for nominal classifiers, where the number of true positives and negatives is used to evaluate the performance. The balanced accuracy rate was used for this purpose.
  The second criterion is based on the distance between the true and predicted labels on the ordinal scale. For most general purposes this scale can be linear, but others can be used as well depending on the problem at hand. Two different measures were included in this study: r_{int} by Costa et al. [1] and m_{cd}, a new measure proposed in this study.

- *Which types of classifiers are most suitable for ordinal classification problems?*
  Based on the results of the synthetic datasets included in the comparison study, there seems to be a difference between nominal and ordinal classifiers. Nominal classifiers outperform the ordinal classifiers based on the balanced accuracy. Ordinal classifiers on the other hand perform better based on a measure like m_{cd}.
  In the case of the real datasets, this difference between both types of classifiers is however not significant. As a result, there is no specific type of classifiers that would be suitable for ordinal classification in molecular analysis. Instead more focus should be put into the feature selection process, to ensure that information of the ordinal relationship between classes is present in the feature data.

  - *Which types of feature selection methods would be suitable for such classifiers?*
    There is a wide range of feature selection methods available, with each one having its pro's and con's. The comparison study did not focus on how the optimal subset of features needs to be selected, but rather what kind of features to select. In the case of ordinal classi-

fication selecting features that are correlated with the class labels, can positively influence the performance results. In other words, any feature selection method that is able to select features based on this correlation would be most suitable for ordinal classification.

- Does taking the order of classes into account improve the results significantly in comparison to nominal classification methods?

The benefit of taking the ordinal relation between classes into account is minimal when the number of available samples and classes is limited. Typically, this is the case in the kind of biological datasets that this study focused on (i.e. only 3-5 classes with some of them having less than 10 samples). Using synthetic datasets, it was possible to show the improvement in results when using ordinal methods instead of nominal ones. This difference was however insignificant in the case of the real datasets used.

- And in comparison to regression methods?

The proportional odds logistic regression used in this comparison study had problems fitting the regression model on the real datasets. As a result, it was not possible to assess the performance of this type of regression and thus comparison was not possible.

Chapter 2

Planning

Five different milestones were used as a framework for the original planning shown in Figure 2.1. A short description of each milestone is given here:

Milestone 1 - Research Proposal During this phase publicly available datasets with ordinal labels will be collected. The data will be preprocessed using unlabeled filtering methods so that they can be directly used for feature selection and classification. Additionally literature on current ordinal multi-class classification methods will be gathered so they can be used as a basis for this thesis. At the end of this milestone the problem statement document will be written.
Deliverables: Problem statement, data sets, literature.

Milestone 2 - Develop Comparison Study The feature selection / classification / evaluation methods found in the previous phase will be designed and implemented during this phase. Additionally, preliminary comparison results will be generated during development.
Deliverables: Source code of implemented functions and preliminary comparison results.

Milestone 3 - Execute Comparison Study During this phase the available methods will be applied on the chosen datasets. With the interpretation and visualization of the comparison results several research questions (see Section 1) can be answered and will also help identify any possible improvements.
Deliverables: Present results of comparison study and possible improvement, source code of implemented functions.

Milestone 4 - Develop Improvements During this phase the proposed improvements will be validated.
Deliverables: Draft report.

Milestone 5 - Finish Report The last phase will include rounding up the project and finishing the final report.
Deliverables: Final report.

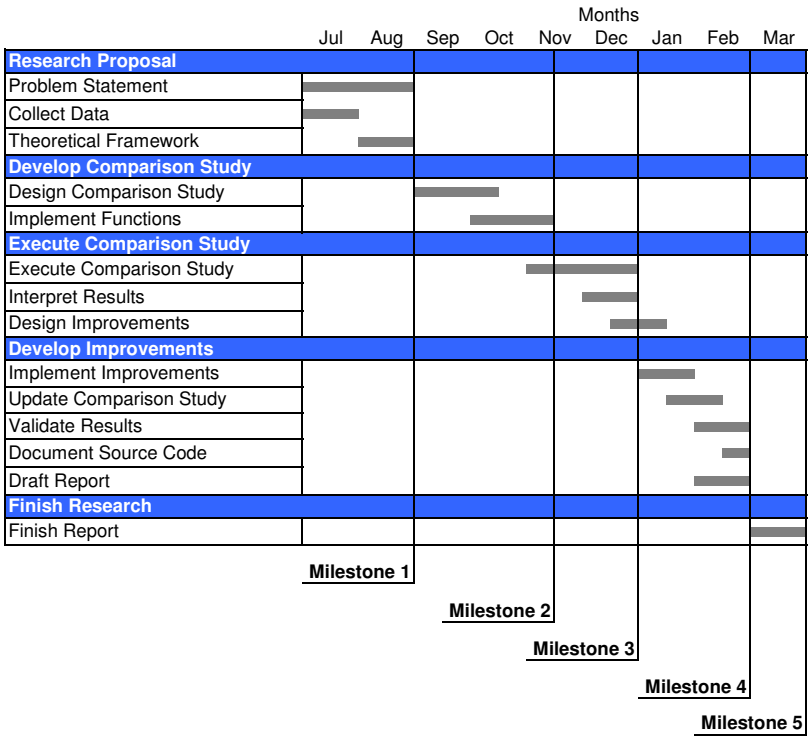


Fig. 2.1: The planning of the project.

Chapter 3

Implementation

The open-source statistical program R[1] was used for the comparison study and the implementation of the studied algorithms. It is a scripting like system similar to Matlab in which queries can be executed in a shell-environment. The reason for choosing this platform is that it is freely available and a wide range of 3rd-party packages specific for the field of bioinformatics have been developed for the Bioconductor project[2].

The R platform supports classes through the use of S4-objects, although it is not always as straightforward to implement an object-oriented design, like one would with programming languages like C# or Java. However, an object-oriented system was developed as shown in the class diagram found in Figure 3.1, as it will allow for adding additional classification, feature selection or evaluation methods in the future. In addition, R is not a strict language when looking at evaluation or type casting, but by applying the object-oriented design this could be enforced (ensuring that variables do not change when least expected).

3.1 Classification Methods

NB - Naïve Bayes The `klaR`-package[3] available in R contains an implementation of the naïve bayes classifier. When using default parameters, the classifier assumes independence of the predictor variables and a Gaussian distribution for these variables.

SVM - Support Vector Machine The `kernlab`-package[4] in R contains the method `ksvm` which includes several implementations of SVM's. The one used in this study is the multi-clas SVM by Crammer et al. [2].

OVO - One-vs-One The one-vs-one trains binary classifiers for every possible pair of classes. In this implementation the naïve Bayes classifier was used for this purpose. Majority voting is used to combine the output of the different classifiers into a final prediction.

OVA - One-vs-All The one-vs-all constructs binary classifiers by training each class against all other classes. In this implementation the naïve Bayes classifier was used for this purpose as well. The binary classifier with the highest probability determines the winning class.

OPC - Ordered Pseudo-Classification This method splits the original multi-class problem into $K - 1$ binary problems. The binary problem can be solved using any classifier that returns posterior probabilities and the naïve bayes classifier from the `klaR`-package was used for this purpose.

[1] `http://cran.r-project.org/`
[2] `http://www.bioconductor.org/`
[3] `http://cran.r-project.org/web/packages/klaR/index.html`
[4] `http://cran.r-project.org/web/packages/kernlab/index.html`

OSVM - Ordered SVM The main principle of this method consists of replicating the data into $K-2$ additional dimensions (see pseudo-code of this process in Algorithm 2). Once replicated the data consists of samples from classes C_1 and C_2 which can be used to a binary SVM. The same implementation of SVM will be used as the one used for the multi-class SVM discussed earlier (*kernlab*-package in R).

Once the binary SVM has been trained any unknown sample can be classified by replicating it in all $K-2$ additional dimensions and predicting their class labels. The prediction of the sample in the original multi-class problem can be determined by counting the number of replicas labeled as C_2 and adding 1 to it. See the pseudo-code listed in Algorithm 3 for a description of the prediction process.

CNB - Cost-based method The cost-based version of NB determines the risk of labeling an unknown sample as class i, by multiplying column i with the class posteriors and summing the result. Once the risk for each class has been calculated, the class with the lowest risk will be the winner. The pseudo-code of this procedure is listed in Algorithm 4.

COVO - Cost-based method The cost-based version of OVO determines the risk for each individual binary classifier. By summing these risks, the total risk of each class is obtained. The class with the lowest total risk will be the winner. The pseudo-code of this procedure is listed in Algorithm 5.

POLR - Proportional Odds Logistic Regression The `MASS`-package[5] available in R contains an implementation of proportional odds logistic regression. It is based on the model by Agresti [3] and uses a quasi-Newton method to estimate its parameters.

3.2 Evaluation Measures

m_{cd} - Cost-based Distance Measure The measure m_{cd} is calculated using the confusion matrix from the results and a predefined cost matrix. See Algorithm 6 for the pseudo-code of this process.

r_{int} - Rank-based Measure The r_{int} measure can be calculated in a quicker way by using a confusion matrix instead [1]. If C is the confusion matrix and $C_{i\bullet}$ and $C_{\bullet j}$ are row i and column j of the matrix C respectively, then the cardinalities can be calculated as follows:

$$|S_1| = \sum_{i=1}^{K} C_{i\bullet}\left(\sum_{j=1}^{K} C_{j\bullet} - 1\right) \tag{3.1}$$

$$|S_2| = \sum_{i=1}^{K} C_{\bullet i}\left(\sum_{j=1}^{K} C_{\bullet j} - 1\right) \tag{3.2}$$

$$|S_{1\cap 2}| = \sum_{i=1}^{K}\sum_{j=1}^{K} C_{i,j}\left(\sum_{i'=i}^{K}\sum_{j'=j}^{K} C_{i',j'} - 1\right) \tag{3.3}$$

3.3 Pseudo-code

[5] `http://cran.r-project.org/web/packages/VR/index.html`

Algorithm 1 Ordered Pseudo Classification

Output: Index of the winning class

 # Construct the binary classifiers C_1 to C_{K-1}
$K \leftarrow$ number of classes
for $i = 1$ to $K - 1$ **do**
  $s_{\text{pos}} \leftarrow$ samples from class 1 to i
  $s_{\text{neg}} \leftarrow$ samples from class $i + 1$ to K
  train classifier C_i using s_{pos} as positive samples and s_{neg} as negative samples
end for
 # Predict an unknown sample x using the individual classifiers
for $i = 1$ to $K - 1$ **do**
  $b_i \leftarrow$ determine posterior probabilities using classifier C_i
end for
 # Determine the combined probability for each class
 # Class 1
$p_1 \leftarrow 1 - b_1$
 # Class 2 to $K - 1$
for $i = 2$ to $K - 1$ **do**
  $p_i \leftarrow b_{i-1} - b_i$
end for
 # Class K
$p_K \leftarrow b_{K-1}$
 # Determine winner and return it
return $\arg\max(p)$

Algorithm 2 Ordered SVM - Train

Input: s: parameter to control the number of neighboring classes
Output: Returns the replicated data

 # First replicate the training set X
$C_1 \leftarrow C_2 \leftarrow$ NULL
$K \leftarrow$ number of classes
for $q = 1$ to $K - 1$ **do**
  $v \leftarrow$ 0-vector of length $K - 2$
  $v[q - 1] \leftarrow 1$
  # Replicate positive samples
  for $j = max(1, q - s + 1)$ to q **do**
    $r \leftarrow$ All samples from class j
    Append v to each sample in r
    Add r to C_1
  end for
  # Replicate negative samples
  for $j = (q + 1)$ to $min(K, q + s)$ **do**
    $r \leftarrow$ All samples from class j
    Append v to each sample in r
    Add r to C_2
  end for
end for
 # Train the SVM using the replicated data
$C \leftarrow$ Train binary SVM using the replicas from C_1 and C_2

Algorithm 3 Ordered SVM - Predict

Input: Trained classifier C
Output: Index of the winning class
Replicate the unknown sample x
$r \leftarrow$ empty vector
$K \leftarrow$ number of classes
for $q = 0$ to $K - 1$ **do**
  $v \leftarrow$ 0-vector of length $K - 2$
  if $q > 0$ **then**
    $v[q - 1] \leftarrow 1$
  end if
  Append v to unknown sample x and add to r
end for
# *Predict replicas*
Predict classes of the replicas in r using trained classifier C
return Number of replicas in r predicted as $C_2 + 1$

Algorithm 4 Cost-based version of NB (CNB)

Output: Index to the winning class
$K \leftarrow$ number of classes
Construct NB
Train NB classifier C
for $i = 1$ to K **do**
  $p_i \leftarrow$ use C to determine the probability of class i given sample x
end for
# *Define the cost matrix*
$M \leftarrow$ a KxK-cost matrix
# *Calculate the risk for each class*
for $j = 1$ to K **do**
  $R_j \leftarrow$ sum$(M[, j] * p)$
end for
# *Determine winner and return it*
return $\arg\min(R)$

Algorithm 5 Cost-based version of OVO (COVO)

Output: Index to the winning class
$K \leftarrow$ number of classes
Construct OVO
Train OVO classifier C
Define the cost matrix
$M \leftarrow$ a KxK-cost matrix
Calculate the risk for each binary classifier
$R \leftarrow$ 0-vector of length K
for each binary classifier in C **do**
  $p \leftarrow$ determine the posterior probabilities given sample x
  for $j = 1$ to K **do**
    $R_j \leftarrow R_j +$ sum$(M[, j] * p)$
  end for
end for
# *Determine winner and return it*
return $\arg\min(R)$

Algorithm 6 Cost-based Distance Measure m_{cd}

Output: A numeric value in the range $[0, 1]$

$A \leftarrow$ determine the confusion matrix from the prediction results
$K \leftarrow$ number of classes
 # Construct a cost-matrix
$M \leftarrow$ create linear $K \times K$-cost matrix
 # Determine the actual total misclassification cost C
$c \leftarrow$ sum the element-wise multiplication of M and A
 # Determine the maximum misclassification cost t
$m \leftarrow$ the maximum of each row in the cost matrix M
$p \leftarrow$ the sum of each column in the confusion matrix A
$t \leftarrow m \cdot p$
 # Determine winner and return it
return $1 - \frac{c}{t}$

Algorithm 7 r_{int}

Output: A numeric value in the range $[0, 1]$

$A \leftarrow$ determine the confusion matrix from the prediction results
$K \leftarrow$ number of classes
 # Determine the cardinality of S_1
for $i = 1$ to K **do**
  $S_1 \leftarrow S_1 +$ sum($A[i,]$) * (sum($A[i$ to $K,]$) - 1)
end for
 # Determine the cardinality of S_2
for $j = 1$ to K **do**
  $S_2 \leftarrow S_2 +$ sum($A[, j]$) * (sum($A[, j$ to $K]$) - 1)
end for
 # Determine the cardinality of $S_{1 \cap 2}$
for $i = 1$ to K **do**
  for $j = 1$ to K **do**
    $S_{1 \cap 2} \leftarrow S_{1 \cap 2} + A[i, j]$ * ((sum($A[i$ to K, j to $K]$) - 1)
  end for
end for
return $S_{1 \cap 2}$ / sqrt(S_1 * S_2)

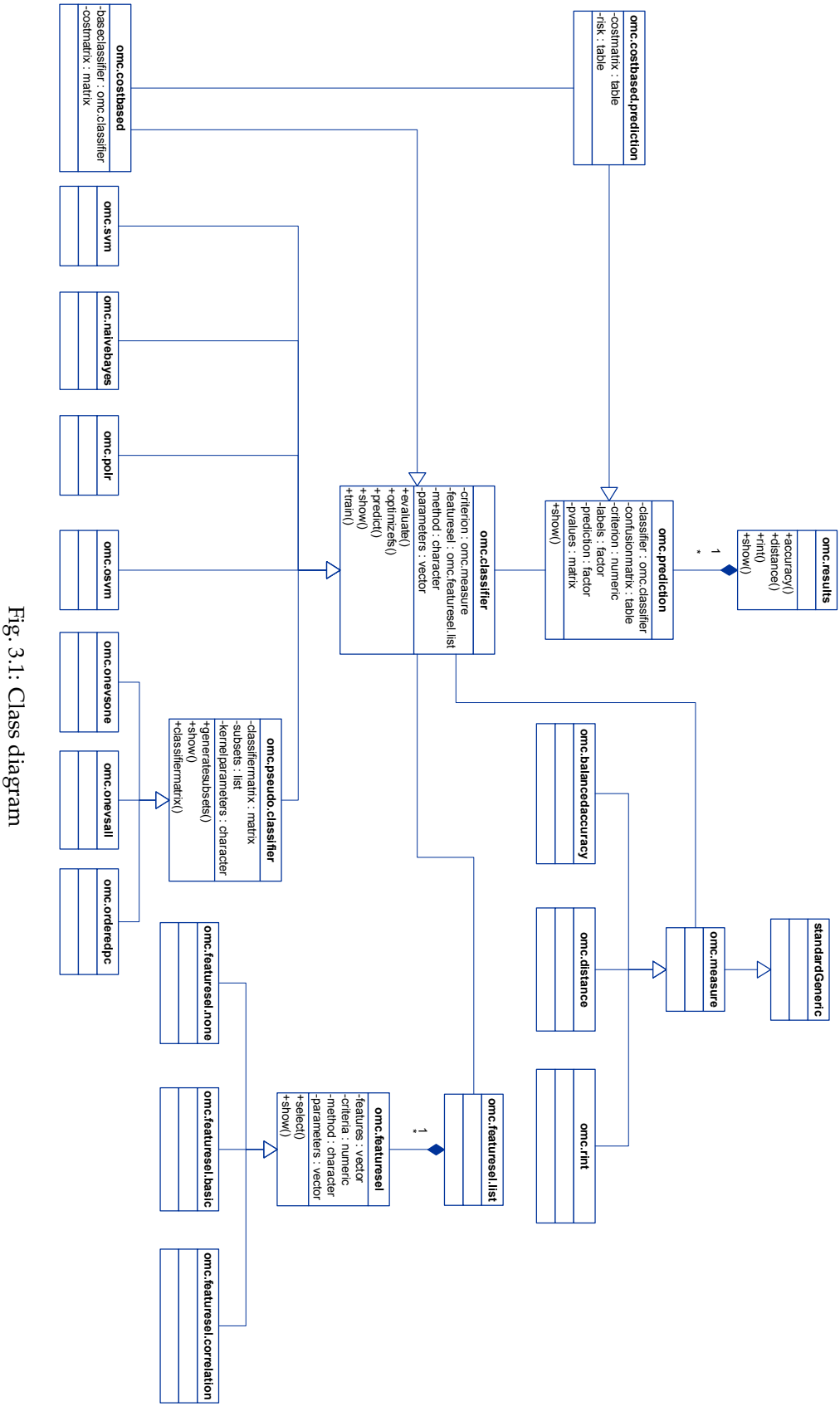


Fig. 3.1: Class diagram

Chapter 4

Additional Experiments

4.1 Synthetic Datasets

Several additional synthetic datasets were experimented with; see Figure 4.1 for a graphical representation of each one of them. The preliminary results of these synthetic datasets were not promising, as there did not seem to be a distinction between the results of the nominal and ordinal classifiers. These datasets were dropped in the end and replaced by the ones included in the comparison study.

4.2 Cost Matrices

A number of different cost matrices were used to investigate, the effect they might have on the prediction results. Depending on the dataset it might be important to have either really high or low costs. In other cases, it might be important to have an asymmetric cost matrix to make the classifier more conservative in its predictions.

Figures 4.3 to 4.5 show the class boundaries of CNB using the different cost matrices. As expected, the choice of the used cost matrix has noticable effect on the result.

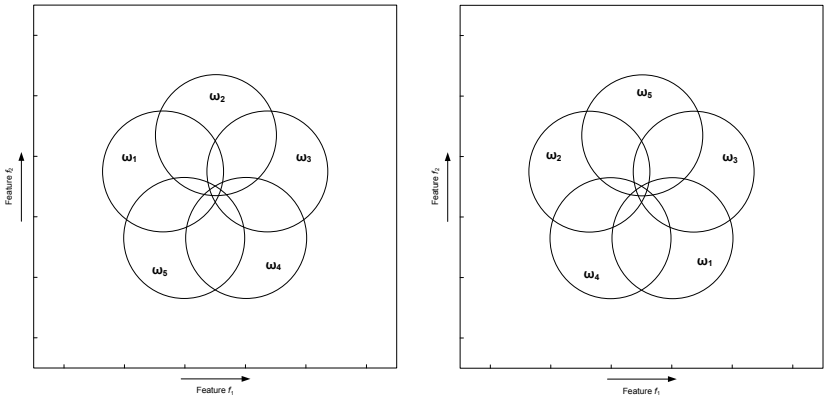
When using the linear cost matrix class 1 is still present, but that's not the case when using either the quadratic or exponential cost matrix. By using a cost matrix with higher costs, like the quadratic cost matrix, the classifier gives preference to classes $2, 3$ or 4 as the costs of misclassifying a sample as class 1 has become too high.

Similar effects can be seen in the asymmetric cost matrices, where the classifier will give preference to misclassifying either up or down the ordinal scale.

4.3 Ordered Pseudo-Classification Variants

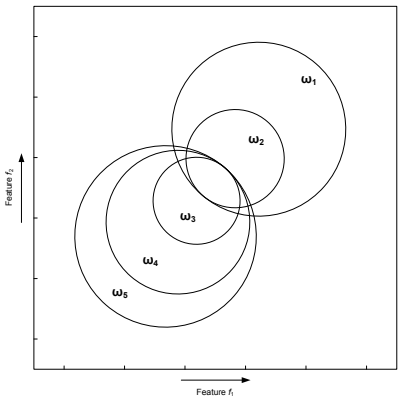
The original OPC has a disadvantage as it assumes that the conditional probability $P(\omega_T > \omega_{k-1}|x)$ is always larger than $P(\omega_T > \omega_k|x)$ (see Equation 5 of the thesis). This is not always the case, as it depends on the distribution of the classes and can lead to negative probabilities if the assumption does not hold. In the implementation used in this comparison study, any negative class posteriors were set to 0. See Figure 4.6c where the curves of all classes are cut off at 0.

Another approach would be to assume that $P(\omega_k) = P(\omega_T > \omega_{k-1} \cap \omega_T <= \omega_k|X)$. In that case the class probabilities can be obtained by determining the intersection of two binary classifiers:

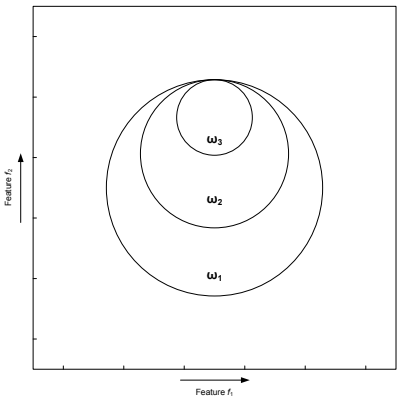


(a) Ordered Circle

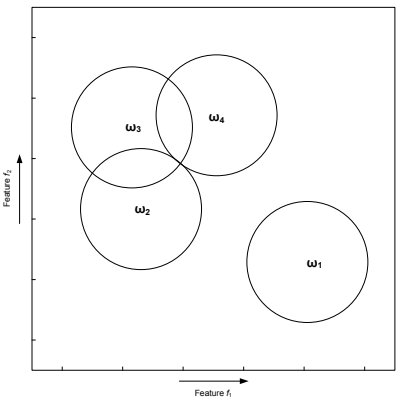
(b) Unordered Circle



(c) Low/High Grade



(d) Subtypes



(e) Outlier

Fig. 4.1: Five different synthetic datasets: (a) Ordered circle (b) Unorderd circle (c) Low/High grade (d) Subtypes (e) Outlier.

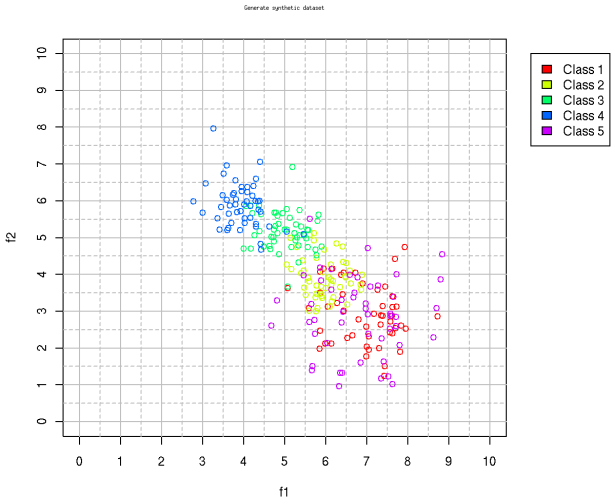


Fig. 4.2: The predecessor of the synthetic I dataset from the comparison study, consisting of 2 features with 5 ordered classes.

$$\begin{aligned} P(\omega_k) &= P(\omega_T > \omega_{k-1} \cap \omega_T \leq \omega_k | X) \\ &= P(\omega_T > \omega_{k-1}|X) P(\omega_T \leq \omega_k | X) \qquad \text{for } k = 1, ..., K \\ &= P(\omega_T > \omega_{k-1}|X)(1 - P(\omega_T > \omega_k | X)) \end{aligned} \tag{4.1}$$

Where $P(\omega_T > \omega_0|X) = 1$ and $P(\omega_T > \omega_K|X) = 0$ as $y \in \{\omega_1, \omega_2, ..., \omega_K\}$ still hold. In this case there's an independence assumption which might not be very accurate.

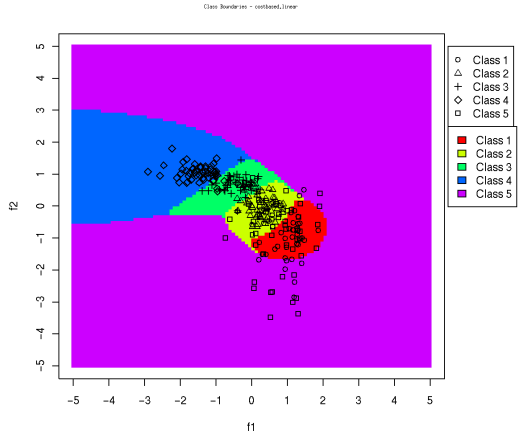
Figure 4.7 shows how a test set of 50 samples performed. As expected the samples of the classes on both ends of the ordinal scale (in this case $C1$ and $C5$) are classified the same for OPC and OPC with modified posteriors. Looking at the middle classes, the OPC with modified posteriors gives more preference to the lower classes, which would represent the original dataset better in comparison to the original OPC that classified most of those samples as $C4$.

4.4 Class Boundaries

To help understand the working of each classifier, the class boundaries were generated to see how all classes are separated. Figures 4.8 and 4.9 show the boundaries of the classifiers evaluated on the synthetic I dataset.

When looking at the class boundaries of an ordinal classifier, one would expect them to transition from one class to a neighboring class on the ordinal scale. For a nominal classifier on the other hand, the class boundaries could switch from one class to any other class. In the case of OSVM, COVO, COVA and POLR this expectation seems to hold true, while for OPC not. This can be accounted to the way OPC determines the class posteriors, as they do not sum up to 1 (which is a result of the way it splits the original problem into several binary problems).

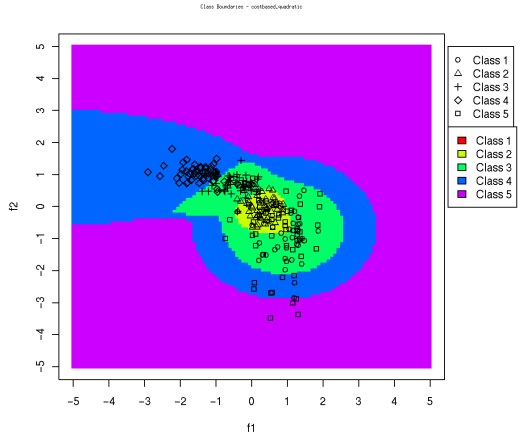
$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 16 & 7 & 0 & 0 & 2 \\ 0 & 19 & 6 & 0 & 0 \\ 0 & 1 & 19 & 5 & 0 \\ 0 & 0 & 4 & 21 & 0 \\ 9 & 8 & 1 & 0 & 7 \end{pmatrix}$$

(a) Linear

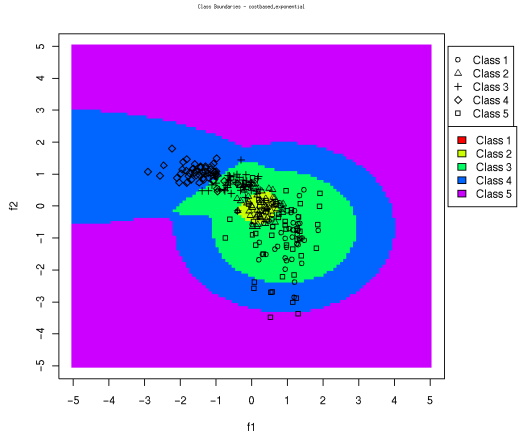
$$\begin{pmatrix} 0 & 1 & 4 & 9 & 16 \\ 1 & 0 & 1 & 4 & 9 \\ 4 & 1 & 0 & 1 & 4 \\ 9 & 4 & 1 & 0 & 1 \\ 16 & 9 & 4 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 7 & 17 & 1 & 0 \\ 0 & 19 & 6 & 0 & 0 \\ 0 & 1 & 19 & 5 & 0 \\ 0 & 0 & 4 & 21 & 0 \\ 0 & 8 & 11 & 5 & 1 \end{pmatrix}$$

(b) Quadratic

$$\begin{pmatrix} e^0 & e^1 & e^2 & e^3 & e^4 \\ e^1 & e^0 & e^1 & e^2 & e^3 \\ e^2 & e^1 & e^0 & e^1 & e^2 \\ e^3 & e^2 & e^1 & e^0 & e^1 \\ e^4 & e^3 & e^2 & e^1 & e^0 \end{pmatrix}$$

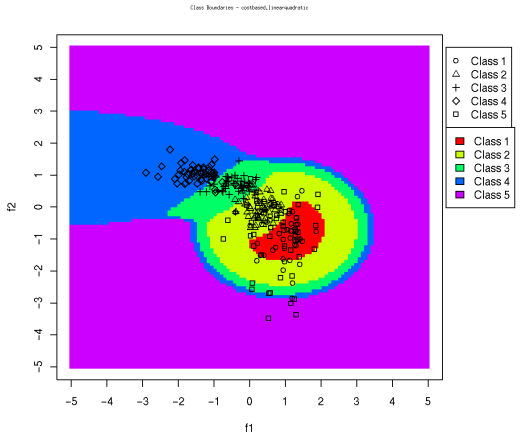


$$\begin{pmatrix} 0 & 3 & 21 & 1 & 0 \\ 0 & 14 & 11 & 0 & 0 \\ 0 & 1 & 19 & 5 & 0 \\ 0 & 0 & 4 & 21 & 0 \\ 0 & 3 & 19 & 2 & 1 \end{pmatrix}$$

(c) Exponential

Fig. 4.3: Each sub figure shows a different symmetric cost matrix (left) with its corresponding class boundaries (middle) and confusion matrix (right). Half of the samples were used to train CNB, while the remaining were used to determine the confusion matrix.

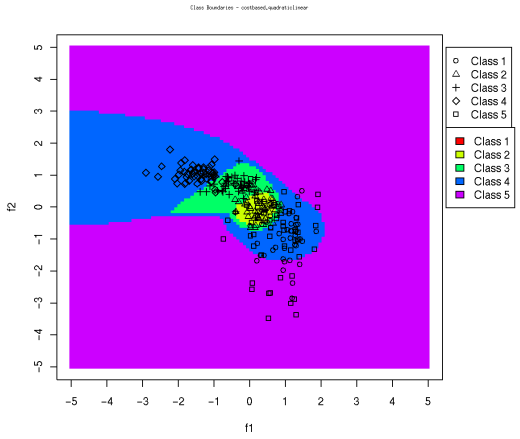
$$\begin{pmatrix} 0 & 1 & 4 & 9 & 16 \\ 1 & 0 & 1 & 4 & 9 \\ 2 & 1 & 0 & 1 & 4 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 16 & 8 & 0 & 1 & 0 \\ 0 & 19 & 6 & 0 & 0 \\ 0 & 2 & 18 & 5 & 0 \\ 0 & 0 & 4 & 21 & 0 \\ 9 & 12 & 3 & 0 & 1 \end{pmatrix}$$

(a) Linear/Quadratic

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 4 & 1 & 0 & 1 & 2 \\ 6 & 4 & 1 & 0 & 1 \\ 9 & 6 & 4 & 1 & 0 \end{pmatrix}$$

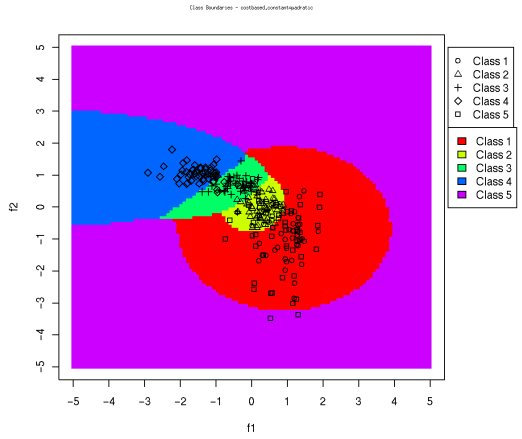


$$\begin{pmatrix} 0 & 3 & 1 & 19 & 2 \\ 0 & 16 & 9 & 0 & 0 \\ 0 & 1 & 19 & 5 & 0 \\ 0 & 0 & 4 & 21 & 0 \\ 0 & 3 & 2 & 13 & 7 \end{pmatrix}$$

(b) Quadratic/Linear

Fig. 4.4: Each sub figure shows a different asymmetric cost matrix (left) with its corresponding class boundaries (middle) and confusion matrix (right). Half of the samples were used to train CNB, while the remaining were used to determine the confusion matrix. The upper and lower triangle of the cost matrices are different, which makes it possible to prefer either a higher or a lower class when the classifier is in doubt.

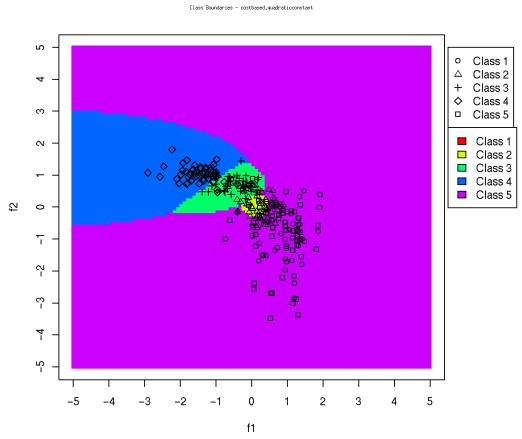
$$\begin{pmatrix} 0 & 1 & 4 & 9 & 16 \\ 1 & 0 & 1 & 4 & 9 \\ 1 & 1 & 0 & 1 & 4 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 20 & 5 & 0 & 0 & 0 \\ 0 & 19 & 6 & 0 & 0 \\ 0 & 2 & 18 & 5 & 0 \\ 0 & 0 & 4 & 21 & 0 \\ 18 & 5 & 1 & 0 & 1 \end{pmatrix}$$

(a) Constant/Quadratic

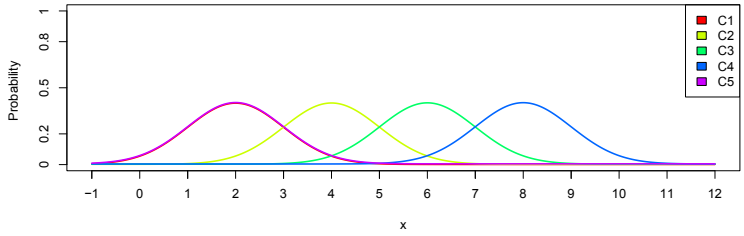
$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 4 & 1 & 0 & 1 & 1 \\ 6 & 4 & 1 & 0 & 1 \\ 9 & 6 & 4 & 1 & 0 \end{pmatrix}$$



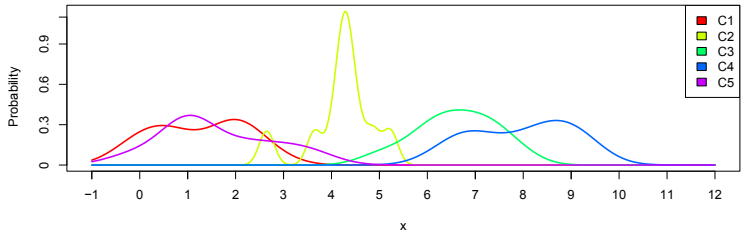
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 24 \\ 0 & 8 & 6 & 0 & 11 \\ 0 & 1 & 19 & 5 & 0 \\ 0 & 0 & 4 & 21 & 0 \\ 0 & 3 & 1 & 0 & 21 \end{pmatrix}$$

(b) Quadratic/Constant

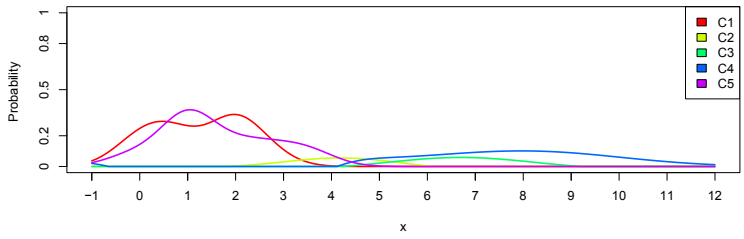
Fig. 4.5: Each sub figure shows a different asymmetric cost matrix (left) with its corresponding class boundaries (middle) and confusion matrix (right). Half of the samples were used to train CNB, while the remaining were used to determine the confusion matrix. The upper and lower triangle of the cost matrices are different, which makes it possible to prefer either a higher or a lower class when the classifier is in doubt.



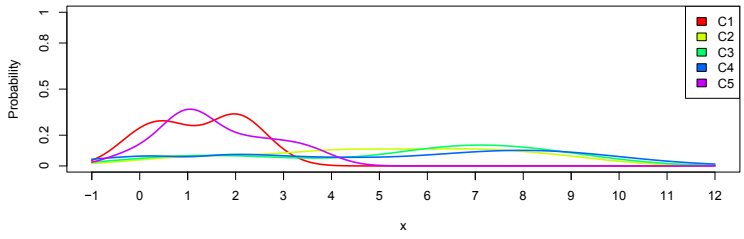
(a) Original Dataset



(b) NB



(c) OPC



(d) OPC (Modified)

Fig. 4.6: A 1 dimensional version of the synthetic I dataset, see (a) for the distribution of the classes (note that classes $C1$ and $C5$ overlap). The class posteriors of three different classifiers are shown: (b) NB, (c) OPC and (d) OPC with modified posteriors (see Equation 4.1)

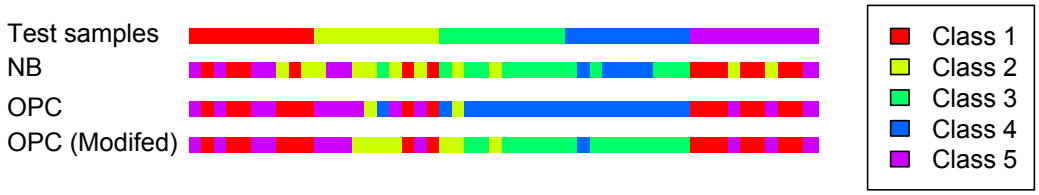
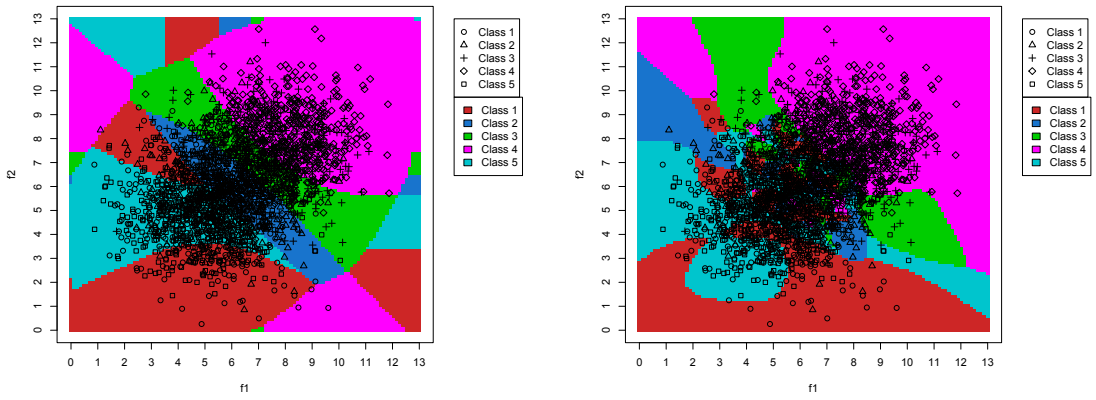
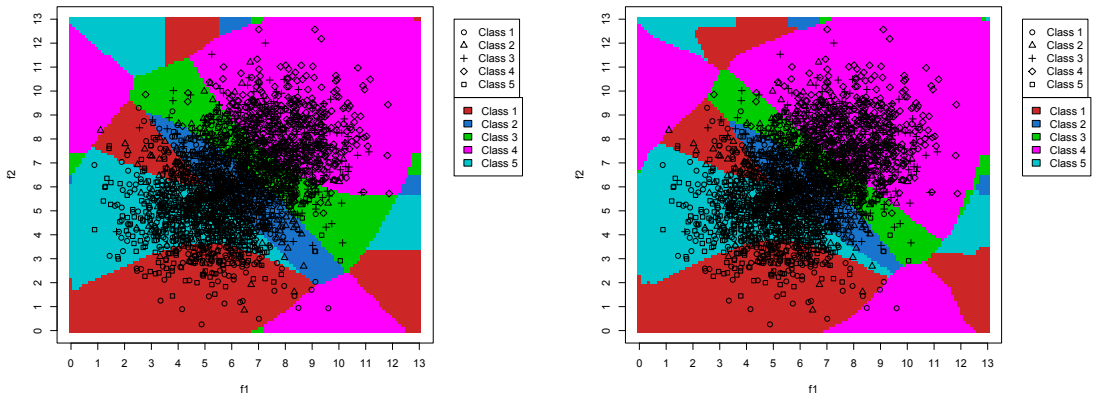


Fig. 4.7: 50 test samples predicted using the three different classifiers (b) NB, (c) OPC and (d) OPC with modified posteriors (see Equation 4.1). Each sample is color coded with their labels, where in (a) they represent the true labels and in (b),(c) and (d) the predicted labels.

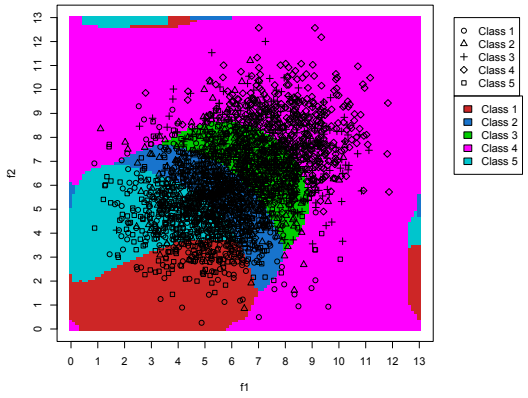


(a) NB (b) SVM

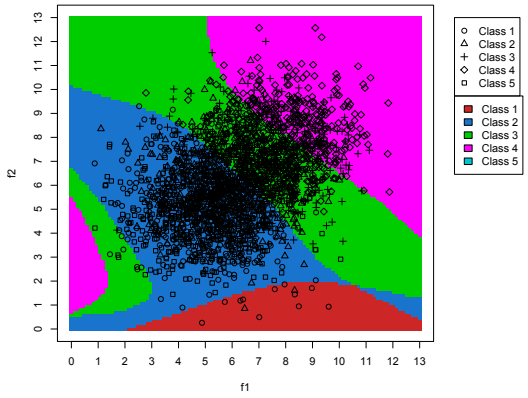


(c) OVO (d) OVA

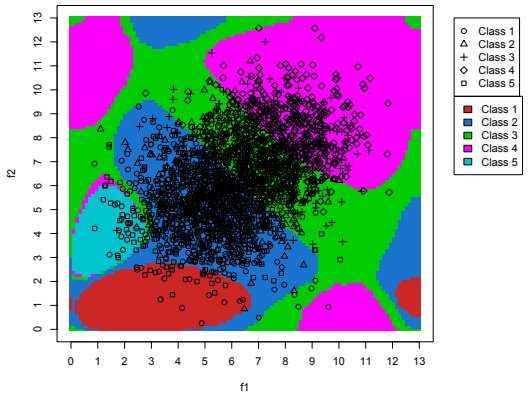
Fig. 4.8: The class boundaries of the nominal classifiers evaluated on the synthetic I dataset.



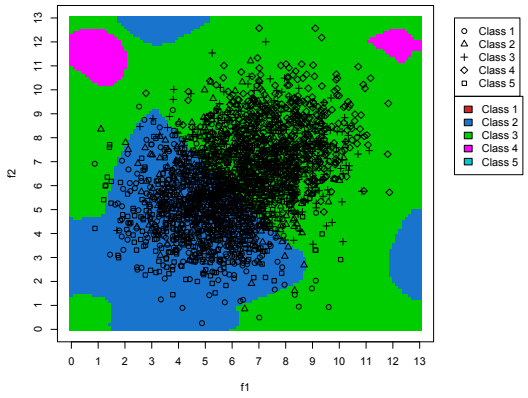
(a) OPC



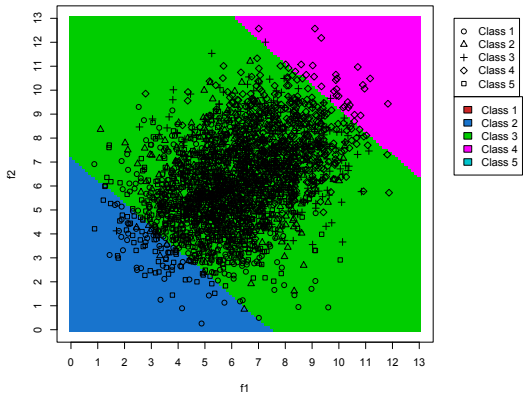
(b) OSVM



(c) CNB



(d) COVO



(e) POLR

Fig. 4.9: The class boundaries of the ordinal classifiers evaluated on the synthetic II dataset.

Bibliography

[1] J. F. P. da Costa, H. Alonso, and J. S. Cardoso, "The unimodal model for the classification of ordinal data.," *Neural Netw*, vol. 21, pp. 78–91, Jan 2008.

[2] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *The Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.

[3] A. Agresti, *Categorical data analysis*. Wiley New York, 2002.