Ordinal Data Classification Using Kernel Discriminant Analysis: A Comparison of Three Approaches

Jaime S. Cardoso*, Ricardo Sousa†, Inês Domingues*

*INESC TEC and Faculdade de Engenharia, Universidade do Porto, Portugal

Email: {jaime.cardoso,ines.c.domingues}@inescporto.pt

†Instituto de Telecomunicações, Faculdade de Ciências, Universidade do Porto, Portugal

Email: rsousa@dcc.fc.up.pt

***Abstract*—Ordinal data classification (ODC) has a wide range of applications in areas where human evaluation plays an important role, ranging from psychology and medicine to information retrieval. In ODC the output variable has a natural order; however, there is not a precise notion of the distance between classes. The recently proposed method for ordinal data, Kernel Discriminant Learning Ordinal Regression (KDLOR), is based on Linear Discriminant Analysis (LDA), a simple tool for classification. KDLOR brings LDA to the forefront in the ODC field, motivating further research.**

This paper compares three LDA based algorithms for ODC. The first method uses the generic framework of Frank and Hall for ODC instantiated with a kernel version of LDA. Similarly, the second method is based on the also generic Data Replication framework for ODC instantiated with the same kernel version of LDA. Both the Frank and Hall and Data Replication methods address the ODC problem by the use of a base binary classifier. Finally, the third method under comparison is KDLOR. The experiments are carried out on synthetic and real datasets. A comparison between the performances of the three systems is made based on t-statistics. The performance and running time complexity of the methods do not support any advantage of KDLOR over the other two methods.

***Keywords*-Ordinal Data, Classification, Kernel Discriminant Analysis, Linear Discriminant Analysis**

I. Introduction

Learning from examples is one of the most successful areas in machine learning. This research area encompasses two fundamental learning frameworks, i.e., supervised and unsupervised learning, each with different assumptions on the uncertainty in the training data. Supervised learning attempts to learn a concept to correctly label unseen instances, where the training instances have known labels, and therefore the ambiguity is at a minimum. Unsupervised learning attempts to learn the structure of the underlying sources of instances, where the labels of the training instances are unknown, and therefore the ambiguity is maximum.

One of the most representative problems of supervised learning is classification, consisting of the estimation of a mapping from the feature space into a finite class space. Depending on the cardinality of the output space, we are left with binary or multiclass classification problems. Finally, the presence or absence of a "natural" order among classes will separate nominal from ordinal problems. As an example, consider the credit score problem. A bank assigns a score to a client given his wage, good payment history in previous mortgages and the number of credits at the time of the evaluation. The score assessment is clearly rendered over the different criteria: wage, payment history, among others. Ideally, one wants to find the best function that can capture all this information in order to predict the true score. It is clear that the scores do carry order information (excellent is definitely better than fair), but scores do not carry numerical information (good is not, for example, half excellent).

Towards the formalization of this learning problem, let us assume that a training set of labeled patterns is available where each pair $\{\boldsymbol{x}_i, y_i\} \in \mathbf{R}^d \times \mathcal{Y}$ has been generated independently from an unknown distribution. The goal is to induce a classifier, i.e., a function from patterns to labels $f : \mathbf{R}^d \rightarrow \mathcal{Y}$. In this paper, we will focus on the ordinal case of $\mathcal{Y} = \{y_1, \cdots, y_K\}$, where $y_1 \prec \cdots \prec y_K$ and $\prec$ is a linear order relation in $\mathcal{Y}$.

One of the first works that addresses the classification of ordinal data was based on generalized linear models, the cumulative model [1]. Later, Frank and Hall [2] introduced a simple process, which explores order in classification problems using conventional binary classifiers. The problem is transformed from a K-class ordinal problem to $K-1$ binary class problems. The main advantage of this scheme is that any binary classifier can be used as the building block.

In 2005, Cardoso [3] presented a technique that overcomes the main limitation of the Frank and Hall method, by imposing a constraint that the individual boundaries do not intersect. This method, called the Data Replication method, can be framed under the single binary classifier reduction (SBC), an approach for solving multi-class problems via binary classification relying on a single, standard binary classifier.

More recently, the Kernel Discriminant Learning Ordinal Regression (KDLOR) method has been proposed [4]. KDLOR is an adaptation of the conventional Linear Discriminant Analysis (LDA) method with a ranking constraint. A complete survey on the topic of ordinal data classification can be found elsewhere [5].

Although the KDLOR method seems to be the only one proposed in the literature based on discriminant analysis, the fact is that proposals exist for ordinal data that are frameworks that can be instantiated with different base models, including (linear or kernel) discriminant analysis. The above mentioned Frank and Hall and Data Replication methods are two such examples.

The major contributions of this paper are a) the instantiation for the first time of the Frank and Hall and Data Replication methods with kernel discriminant analysis (KDA) and b) the comparative performance of three models for ordinal data classification based on discriminant analysis. All systems are based on the KDA and their efficiency is tested on both synthetic and real datasets using the same experimental protocol. Moreover, their performance is further assessed by using t-statistics.

II. Three KDA-based Models for Ordinal Data Classification

One way to view a linear binary classification model is in terms of dimensionality reduction. In general, the projection onto one dimension leads to a considerable loss of information and classes that are well separated in the original space may strongly overlap in one dimension. However, by proper selection of the projection, one can find a direction that maximizes the class separation. The simplest measure of the separation of the classes, when projected onto $\boldsymbol{w}$, is the separation of the projected class means. However, strongly nondiagonal covariances of the class distributions pose difficulties to this simple approach [6]. LDA explicitly attempts to model the difference between the classes of data, maximizing the following objective function:

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^t S_B \boldsymbol{w}}{\boldsymbol{w}^t S_W \boldsymbol{w}} \tag{1}$$

where S_B and S_W are the Between-Class Scatter Matrix and Within-Class Scatter Matrix, respectively. The optimal solution can be found by computing the eigenvalues of $S_B^{-1} S_W$ and taking the eigenvectors corresponding to the largest eigenvalues to form a new basis for the data. KDA is an extension of LDA to non-linear distributions. The objective of KDA is also to find a transformation maximizing the between-class variance and minimizing the within-class variance. In non-linearly separable datasets, it is difficult to directly compute the discriminating features between the two classes of patterns in the original input space. By defining a non-linear mapping from the input space to a high-dimensional feature space, we (expect to) obtain a linearly separable distribution in the feature space. Then LDA, the linear technique, can be performed in the feature space to extract the most significant discriminating features. However, the computation may be problematic or even impossible in the feature space owing to the high dimensionality. By introducing a kernel function which corresponds to the non-linear mapping, all the computation can conveniently be carried out in the input space. The problem can be finally solved as an eigen-decomposition problem like LDA. The binary LDA or KDA can be used as a base model to solve multiclass, and in particular ordinal data, classification problems. The design of classifiers adapted for ordinal data classification can be systematized in three different approaches:

- A standard approach to solve multiclass classification problems is to reduce them to binary classification. Commonly, several binary classifiers are designed in the training set, and the multiclass problem is solved by combining the predictions of the ensemble of binary classifiers. One-versus-all and one-versus-one are key examples of this approach.
- A less known reduction technique is to train a single binary classifier over an extended training set, containing multiple replicas of the original data. The predictions of the single binary classifier on the multiple replicas are combined to create the multiclass prediction.
- Some algorithms that have been originally proposed to solve the binary classification problem are naturally extended to the ordinal case.

Next, we overview the three methods under comparison, which span the three enumerated alternatives.

A. The Frank and Hall Framework

Frank and Hall [2] proposed using $(K-1)$ standard binary classifiers to address the K-class ordinal data problem. Toward that end, the training of the i-th classifier is performed by converting the ordinal dataset with classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$ into a binary dataset, discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$. To predict the class value of an unseen instance, the $(K-1)$ outputs are combined to produce a single estimation. If the i-th classifier predicts $\mathcal{C}_X > \mathcal{C}_i$ with probability p_i, [2] suggest to estimate the probability values of each of the K classes as

$$\begin{aligned} p_{\mathcal{C}_1} &= 1 - p_1 \\ p_{\mathcal{C}_j} &= p_{j-1} - p_j \qquad j = 2, \cdots, K-1 \\ p_{\mathcal{C}_K} &= p_{K-1} \end{aligned} \tag{2}$$

Note however that this approach may lead to estimates of negative estimates of probability values. A solution to that problem is to identify the output p_i of the i-th classifier with the conditional probability $p(\mathcal{C}_X > \mathcal{C}_i \,|\, \mathcal{C}_X > \mathcal{C}_{i-1})$. This can be exploited to rank the classes according to:

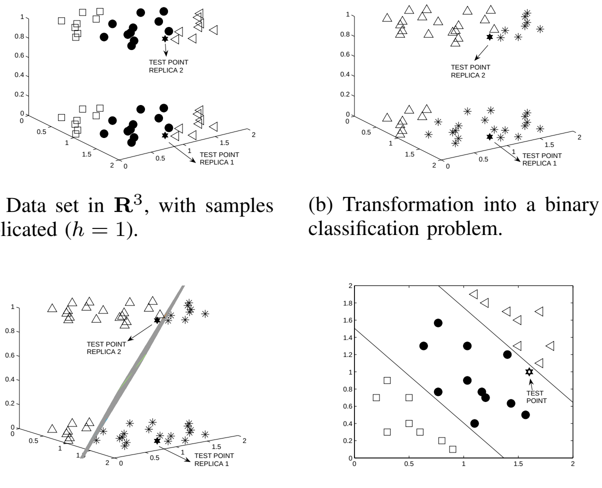
$$\begin{aligned} p(\mathcal{C}_X > \mathcal{C}_1) &= p_1 \\ p_{\mathcal{C}_1} &= 1 - p_1 \\ p(\mathcal{C}_X > \mathcal{C}_j) &= p_j \, p(\mathcal{C}_X > \mathcal{C}_{j-1}) \\ p_{\mathcal{C}_j} &= (1 - p_j) \, p(\mathcal{C}_X > \mathcal{C}_{j-1}) \quad j = 2, \cdots, K-1 \\ p_{\mathcal{C}_K} &= p(\mathcal{C}_X > \mathcal{C}_{K-1}) \end{aligned} \tag{3}$$

Any binary classifier can be used as the building block of this scheme. Observe that, under the Data Replication method to be presented next, the i-th boundary is also discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$; the major difference lies in the *independence* of the boundaries found with Frank and Hall's method. This independence is likely to lead to intersecting boundaries.

B. The Data Replication Method

The Data Replication method for ordinal data can be framed under the single binary classifier (SBC) reduction, an approach for solving multiclass problems via binary classification relying on a single, standard binary classifier. SBC reductions can be obtained by embedding the original problem in a higher-dimensional space consisting of the original features, as well as one or more other features determined by fixed vectors, designated here as *extension features*. This embedding is implemented by replicating the training set points so that a copy of the original point is concatenated with each of the extension features vectors. The binary labels of the replicated points are set to maintain a particular structure in the extended space. This construction results in an instance of an artificial binary problem, which is fed to a binary learning algorithm that outputs a single binary classifier. To classify a new point, the point is replicated and extended similarly and the resulting replicas are fed to the binary classifier, which generates a number of signals, one for each replica. The class is determined as a function of these signals [7].

Consider a simplified toy example with just three classes in $\mathbf{R}^2$. Here, the task is to find two parallel hyperplanes, the first one discriminating class $\mathcal{C}_1$ against classes $\{\mathcal{C}_2, \mathcal{C}_3\}$ and the second hyperplane discriminating classes $\{\mathcal{C}_1, \mathcal{C}_2\}$ against class $\mathcal{C}_3$. These hyperplanes will correspond to the solution of two binary classification problems but with the additional constraint of parallelism. The Data Replication method solves both problems simultaneously in an augmented feature space [3].



(a) Data set in $\mathbf{R}^3$, with samples replicated ($h = 1$).

(b) Transformation into a binary classification problem.

(c) Linear solution to the binary problem.

(d) Linear solution in the original dataset.

Figure 1: Data Replication model in a toy example (from [3]).

In the toy example, using a transformation from the $\mathbf{R}^2$ initial feature-space to a $\mathbf{R}^3$ feature space (Figure 1a), replicate each original point, according to the rule:

$$\mathbf{x}\in\mathbf{R}^2 \begin{array}{l}\nearrow \left[\begin{smallmatrix}\mathbf{x}\\ h\end{smallmatrix}\right]\in\mathbf{R}^3\\ \searrow \left[\begin{smallmatrix}\mathbf{x}\\ 0\end{smallmatrix}\right]\in\mathbf{R}^3\end{array},\quad \text{where } h = \text{const} \in \mathbf{R}^+ \tag{4}$$

Observe that any two points created from the same original point differ only in the extension feature. Now define a binary training set in the new (higher dimensional) space (Figure 1b) according to:

$$\begin{array}{l}\begin{bmatrix}\mathbf{x}_i^{(1)}\\ 0\end{bmatrix}\in\overline{\mathcal{C}}_1,\ \begin{bmatrix}\mathbf{x}_i^{(2)}\\ 0\end{bmatrix},\begin{bmatrix}\mathbf{x}_i^{(3)}\\ 0\end{bmatrix}\in\overline{\mathcal{C}}_2\\ \begin{bmatrix}\mathbf{x}_i^{(1)}\\ h\end{bmatrix},\begin{bmatrix}\mathbf{x}_i^{(2)}\\ h\end{bmatrix}\in\overline{\mathcal{C}}_1,\ \begin{bmatrix}\mathbf{x}_i^{(3)}\\ h\end{bmatrix}\in\overline{\mathcal{C}}_2\end{array} \tag{5}$$

In this step, we are defining the two binary problems as a single binary problem in the augmented feature space. A linear two-class classifier can now be applied on the extended dataset, yielding a hyperplane separating the two classes, see Figure 1c. The intersection of this hyperplane with each of the subspace replicas can be used to derive the boundaries in the original dataset, as illustrated in Figure 1d.

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

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

Note that only three sequences are possible. The generalization for any problem in $\mathbf{R}^p$, with K ordinal classes and nonlinear boundaries can be found in [3].

C. KDLOR

The Kernel Discriminant Learning Ordinal Regression (KDLOR) algorithm [4] is an extension of KDA for ordinal data classification. The main goal can be described as finding the optimal linear projection for classification (from which different classes can be well separated), preserving at the same time the ordinal information of classes, i.e., the average projection of the samples from the higher rank classes should be larger than that of lower rank classes [4].

The original LDA optimization problem is transformed and extended with a penalty term (C) to account for the constraint in the projected means:

$$\min\ J(\boldsymbol{w}, \rho) = \boldsymbol{w}^T S_W \boldsymbol{w} - C\rho$$

$$\text{subject to } \boldsymbol{w}^T(\mathbf{m}_{k+1} - \mathbf{m}_k) \geq \rho$$

where $\mathbf{m}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} \mathbf{x}_i$, N_k is the sample size, and ρ represents the minimum difference of the projected means between consecutive classes. So, if $\rho > 0$, the projected means are ranked according to the ordinal scale.

To accommodate nonlinear problems, the algorithm is modified with the kernel trick resulting in

$$\min\ f(\alpha) = \sum_{k=1}^{K-1} \alpha_k(\mathbf{m}_{k+1} - \mathbf{m}_k)^T S_W^{-1} \sum_{k=1}^{K-1} \alpha_k(\mathbf{m}_{k+1} - \mathbf{m}_k)$$

$$\text{subject to } \alpha_k \geq 0, k = 1, \ldots, K-1 \text{ and } \sum_{k=1}^{K-1} \alpha_k = C$$

This is a Quadratic Programming (QP) optimization problem with linear constraints [4] that can be solved by any QP tool.

III. Experimental Study

In order to compare the performance of the aforementioned algorithms, we performed experiments on artificial and real-world datasets. The Frank and Hall (FH_LDA) and Data Replication (oLDA) methods were instantiated with the KDA method, as made available in the Statistical Pattern Recognition Toolbox for Matlab (STPRtool)[1].

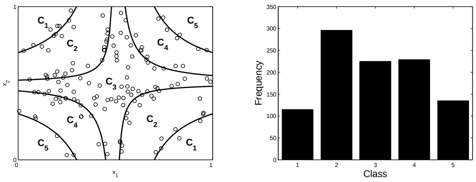
1) Datasets: For the synthetic datasets we have started by the SyntheticI dataset which is uniformly distributed in the unit square $[0,1] \times [0,1] \subset \mathbf{R}^2$ as used in previous studies for benchmarking ordinal methods [8], [3], [9] consisting of 1000 samples. For this dataset we assigned to each example $\boldsymbol{x}$ a class $y \in \{1,\ldots,5\}$ corresponding to

$$y = \min_{r\in\{1,\ldots,5\}} \{r : b_{r-1} < 10(x_1-0.5)(x_2-0.5)+\varepsilon \ < b_r\}$$

where

$$(b_0, b_1, b_2, b_3, b_4, b_5) = (-\infty, -1, -0.1, 0.25, 0.4, 1, +\infty).$$

ε is a random variable with normal distribution with zero mean and 0.125^2 of variance (see Figure 2).



(a) Scatter plot of the 14.2% data points wrongly classified. Also shown are the class boundaries.

(b) Class distribution.

Figure 2: Synthetic dataset with 5 classes in $\mathbf{R}^2$.

To investigate the influence of the number of classes and data dimension on models' relative performance, the described experiment was repeated for a dataset SyntheticII with 10 classes in $\mathbf{R}^4$. This time 2000 example points $\boldsymbol{x} = [x_1 \ x_2 \ x_3 \ x_4]^t$ were generated uniformly at random in the unit square in $\mathbf{R}^4$. The rank of each example was assigned according to the rule

$$y = \min_{r\in\{1,\ldots,10\}} \{r : b_{r-1} < 1000\prod_{i=1}^{4}(x_i - 0.5) + \varepsilon \ < b_r\}$$

where

$$(b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_0, b_{10}) = \\ (-\infty, -5, -2.5, -1, -0.4, 0.1, 0.5, 1.1, 3, 6, +\infty).$$

For the real datasets we used available data from the Weka datasets website and the UCI Machine Learning repository[2] [10]. The first dataset, SWD, contains real-world assessments of qualified social workers regarding the risk facing children if they stayed with their families at home and is composed by 10 features and 4 classes. The LEV dataset contains examples of anonymous lecturer evaluations, taken at the end of MBA courses and is composed by 4 features and 5 classes.These datasets contain 1000 examples each. Another dataset which we worked on was the ESL dataset containing 488 profiles of applicants for certain industrial jobs. Features are based on psychometric tests results and interviews with the candidates performed by expert psychologists. The class assigned to each applicant was an overall score corresponding to the degree of fitness for the type of job. The Balance dataset available on UCI machine learning repository was also tested. This was created to model psychological experimental results, where each example is labeled as having a balance scale tip to the right, left or balanced.

2) Methodology: We randomly split each dataset into training and test sets. In order to study the effect of varying the size of the training set, we considered different trainings sizes of all the available data: 20%, 40%, and 60%. The splitting of the data into training and test sets was repeated 100 times in order to assess the variability of the performance. For the synthetic datasets, data were generated in each repetition whereas for the real data, points were randomly shuffled. The best parameterization for each model was found by 'grid-search', based on a 5-fold cross validation scheme conducted on the training set. Finally, the error of the models was estimated on the test set. We have chosen the RBF kernel for all methods and the grid search was performed over C = $2^5, \ldots, 2^{12}$ and $\gamma = 2^{-3}, \ldots, 2^1$. The code needed to reproduce all reported results is available upon request to the authors. The performance of the learning methodologies was assessed with the mean absolute error (MAE). Throughout we speak of two results as being "significantly different" if the difference is statistically significant at the 1% level according to a paired two sided t-test, where each pair of data points consists of the estimates obtained in one of the 100 runs of the two learning schemes being compared [11].

3) Results: Table I and Table II show the MAE estimates for the three methods under comparison. Standard deviations are also shown (based on the 100 MAE estimates). Results are marked with $\star$ if they are statistically better than the results of the two other methods, and with $\bullet$ if they show statistically significant degradation over the two other methods. The results show that the KDLOR was never statistically significantly better than oLDA and FH_LDA. In fact, KDLOR is statistically significantly worse than the two other models for all experiments, except in the training with 60% of the second synthetic dataset (which produced unexpected results). Additionally, oLDA and FH_LDA seem to be comparable in terms of accuracy, with a slight advantage of oLDA. For some experiments, their difference in terms of performance is not statistically relevant. A legitimate conclusion is that it is enough to, in a specific application scenario, test and compare oLDA and FH_LDA, since KDLOR is never

[1] http://cmp.felk.cvut.cz/cmp/software/stprtool/

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

		Training Sizes					
	Methods	20%		40%		60%	
syntheticI	oLDA	0.291 ± 0.025		0.293 ± 0.027		0.314 ± 0.044	
	FH_LDA	0.279 ± 0.022	⋆	0.272 ± 0.021	⋆	0.269 ± 0.027	⋆
	KDLOR	0.536 ± 0.078	●	0.520 ± 0.066	●	0.506 ± 0.057	●
syntheticII	oLDA	0.715 ± 0.034	⋆	0.628 ± 0.126	⋆	1.612 ± 0.520	●
	FH_LDA	0.844 ± 0.046		0.901 ± 0.043		1.005 ± 0.051	⋆
	KDLOR	1.895 ± 0.163	●	1.443 ± 0.121	●	1.279 ± 0.128	

Table I: Average MAE ± standard deviations on 100 simulations for the synthetic datasets.

		Training Sizes					
	Methods	20%		40%		60%	
balance	oLDA	0.165 ± 0.026	⋆	0.097 ± 0.018	⋆	0.076 ± 0.019	⋆
	FH_LDA	0.175 ± 0.024		0.121 ± 0.018		0.097 ± 0.018	
	KDLOR	0.222 ± 0.021	●	0.198 ± 0.018	●	0.191 ± 0.024	●
esl	oLDA	0.359 ± 0.024	⋆	0.331 ± 0.026		0.321 ± 0.030	⋆
	FH_LDA	0.374 ± 0.030		0.331 ± 0.024		0.330 ± 0.029	
	KDLOR	0.464 ± 0.074	●	0.407 ± 0.041	●	0.387 ± 0.035	●
lev	oLDA	0.488 ± 0.032		0.439 ± 0.025	⋆	0.407 ± 0.022	⋆
	FH_LDA	0.489 ± 0.031		0.442 ± 0.024		0.434 ± 0.023	
	KDLOR	0.569 ± 0.043	●	0.529 ± 0.029	●	0.518 ± 0.028	●
swd	oLDA	0.560 ± 0.028		0.495 ± 0.022		0.483 ± 0.032	⋆
	FH_LDA	0.556 ± 0.027	⋆	0.489 ± 0.019	⋆	0.496 ± 0.026	
	KDLOR	0.623 ± 0.052	●	0.630 ± 0.049	●	0.611 ± 0.033	●

Table II: Average MAE ± standard deviations on 100 simulations for the real datasets.

better than both.

A final note in terms of the time taken to train the three models. The simple model of Frank and Hall was the fastest to train since it does not impose constraints among the base models; they are all trained independently, facilitating the training. The Data Replication method was the slowest to train, both because of the data augmentation to impose constraints and the additional parameters to optimize through cross-validation.

IV. Conclusions

This paper discusses and compares three methods for ordinal data based on discriminant analysis. The first model is based on the Frank and Hall framework that converts the original ordinal class problem into a series of binary class problems. The second model trains a single binary model in an extended dataset. The third model, KDLOR, has been recently proposed and adapts the formulation of discriminant analysis to incorporate constraints in the order of the projected means. The mapping of the Frank and Hall and Data Replication frameworks into discriminant analysis had never been attempted before. Moreover, the recent KDLOR specific for discriminant analysis, raised the question of its comparative advantage over existing methods. Our empirical findings seem to demonstrate that KDLOR, the most recent method, is in general the less accurate model.

Acknowledgment

This work was partially funded by Fundação para a Ciência e a Tecnologia (FCT) - Portugal through project PTDC/SAU-ENB/114951/2009.

References

[1] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society*, vol. 42, no. 2, pp. 109–142, 1980.

[2] E. Frank and M. Hall, "A simple approach to ordinal classification," in *12th European Conference on Machine Learning*, 2001, pp. 145–156.

[3] J. S. Cardoso and J. F. P. d. Costa, "Learning to classify ordinal data: the data replication method," *Journal of Machine Learning Research*, vol. 8, pp. 1393–1429, 2007.

[4] B. Sun, J. Li, D. D. Wu, X. Zhang, and W. Li, "Kernel discriminant learning for ordinal regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 906–910, 2010.

[5] R. Sousa, I. Yevseyeva, J. F. P. d. Costa, and J. S. Cardoso, "Multicriteria models for learning ordinal data: a literature review," in *Artificial Intelligence, Evolutionary Computation and Metaheuristics - In the footsteps of Alan Turing*, X. Yang, Ed., 2012.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[7] R. El-Yaniv, D. Pechyony, and E. Yom-Tov, "Better multiclass classification via a margin-optimized single binary problem," *Pattern Recognition Letters*, vol. 29, pp. 1954–1959, 2008.

[8] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," *Ninth International Conference on Artificial Neural Networks*, vol. 1, pp. 97–102, 1999.

[9] R. Sousa and J. S. Cardoso, "Ensemble of decision trees with global constraints for ordinal classification," in *11th International Conference on Intelligent Systems Design and Applications*, 2011, pp. 1164–1169.

[10] A. Ben-David and L. Sterling, "Generating rules from examples of human multiattribute decision making should be simple," *Expert Systems with Applications*, vol. 31, no. 2, pp. 390–396, 2006.

[11] T. M. Mitchell, *Machine Learning*, 1st ed. USA: McGraw-Hill, Inc., 1997.