Data-driven rank ordering — a preference-based comparison study

Maria Dobrska [1], Hui Wang [1], William Blackburn [1]

[1] *School of Computing and Mathematics, Univeristy of Ulster,*
Shore Road,
Newtownabbey, BT37 0QB, United Kingdom
E-mail: dobrska-m@email.ulster.ac.uk, {h.wang,wt.blackburn}@ulster.ac.uk

Received: 22-12-2009
Accepted: 15-12-2010

Abstract

Data driven rank ordering refers to the rank ordering of new data items based on the ordering inherent in existing data items. This is a challenging problem, which has received increasing attention in recent years in the machine learning community. Its applications include product recommendation, information retrieval, financial portfolio construction, and robotics. It is common to construct ordering functions based on binary pairwise preferences. The level of dominance within pairs has been modelled in approaches based on statistical models, where strong assumptions about the distributions of the data are present. For learning pairwise preferences from the data we introduce a distribution-independent framework incorporating the level of dominance. We compare our approach with learning to rank order based on binary pairwise preferences through experiments using large margin classifiers.

Keywords: rank ordering, preference learning, preference with level of dominance

1. Introduction

Rankings are very useful sources of information; in fact people often rely on rankings while making decisions in their everyday lives. For example, in sports tournaments participants may be ranked based on their performance, in trading products may be ranked based on their qualities or customer opinions.

Rankings have been extensively studied in economics, psychology, operations research, and other human-centred disciplines, usually under the term "preference" [1]. In recent years rankings have found applications in many areas of artificial intelligence, including recommender systems, e-commerce, multi-agent systems, planning and scheduling, intelligent financial data analysis and fuzzy number-based multicriteria decision making [1,2].

Rankings provide potentially more powerful information about data than, for example, class membership of data. Many researchers agree that rankings based on pairwise preferences are more informative than performance measures for single instances [3]. For example, information about the total number of goals scored and conceded during a football tournament by each team is considered less relevant in performance ranking than the set of game results between the teams.

In the study of rankings the central problem is: given some instances, how should they be ranked in a rational way? In existing studies, it is usually assumed that the pairwise preferences are provided by, for example experts, and the aim is to find a ranking that agrees maximally with the given preferences.

However in some cases such preferences are not directly available, so they need to be learned. The importance of preference learning has been highlighted in [4,1].

In this paper we present our methodology for creating a ranking based on pairwise preferences. We derive the preference function from the historical data and do not rely on any ranking experts or side information. Our preference function not only provides information on which of two given instances is preferred over the other one (we will refer to such information as *binary preference*, because it has binary output), but unlike other well-studied binary problems [5,6] it also provides the level (strength) of such dominance.

Our hypothesis is that with the use of the same (or similar) machine learning methodologies, trained preferences with level of dominance lead to better rank orderings than trained binary preferences. It reduces the number of ties in rankings, reduces ranking error and performs significantly better than binary preferences for highly unbalanced data, when one domain dominates the other one, therefore instances from one dataset are usually (however not always) preferred than instances from the other dataset. We reformulate the problem from binary classification into regression and aim at building a structure and distribution-independent model.

This paper is organised as follows: Section 2 provides background for the problem of rank ordering based on pairwise preferences. In Section 3 we discuss the problem of preference learning, compare approaches based on binary preference with preference incorporating level of dominance and address the key issues and differences. Section 4 provides experimental results and description of data used in our experiments. In Section 5 we conclude this work and discuss future challenges.

2. Preference-based rank ordering — background

The notion of ranking is used in various areas of research as well as everyday life. It is a subject of study in decision making, as well as in artificial intelligence and data mining.

Early research work addressing the subject of rank ordering based on preference judgements was reported by Hans Buhlmann and Peter Huber. In [7] the problem of choosing the best ranking for a given preference matrix is discussed. Since the instances to be rank ordered are participants in a tournament whose scores are known, the availability of preference judgements can be assumed as they are based on the score of each game in the tournament.

William Cohen studied the problem of rank ordering based on pairwise comparisons. In [8] he and his co-workers addressed these issues with the use of preference judgements which provide information about priority between each of two given instances. In this study the availability of "primitive preference functions" for a set of instances to be rank ordered is assumed, i.e. the presence of some "ranking experts". The preference training involves the search for the optimal weighting of such experts, or more precisely their judgements regarding preferences among instances.

The construction of the ranking from the set of pairwise preference judgements requires formalisation of the notion of agreement between a ranking and the preference function. In [8] the agreement is measured as a sum of values of the preference function for those pairs of instances for which the preference function and the ordering agree, i.e. if the preference function indicates that a should be ranked ahead of b, the ordering reflects this preference. It is proved in [8] that finding the ordering whose agreement with preference function exceeds some rational number is NP-complete. A "greedy algorithm" for the search of quasi-optimal ordering is then introduced to address this issue. It is further proved in [8] that the greedy algorithm finds an ordering whose extent of agreement with a given function is not smaller than half the value of the agreement of the optimal ordering.

An improvement of the greedy algorithm can be found in [9]. The greedy algorithm treats the instances as nodes of a graph and weights of directed edges between those nodes are derived from preference judgements. Improvement of the algorithm involves dividing the initial graph into subgraphs referred to as strongly connected components. The ordering for

each of the subgraphs is conducted with the use of the greedy algorithm.

According to [3] pairwise preferences are very useful information for ranking creation, as they are simple and potentially sharper than other approaches. The search for optimal ranking based on pairwise preferences involves maximisation of the probability of full agreement between an ordering and pairwise preferences. Pairwise preferences are based on the estimated probability of one instance outperforming another.

There are also several approaches for ranking creation based on preferences which utilise information about historical data and do not employ any ranking experts. In [6] Chu and Ghahramani propose a probabilistic kernel approach to preference learning based on a Gaussian process. This assumes the existence of latent function values associated with each instance to be rank ordered, which preserves pairwise dependencies. The values of such functions are assumed to be realisations of random variables in a zero-mean Gaussian process. Such assumption weakens the applicability of the methodology for ranking instances derived from different domains with different distributions.

In [10] the problem of ranking of labels for given instances is addressed. The main point is to design a methodology which ranks the labels (from a given fixed set). It is then compared with classification methodology, where only one label is associated with each instance. The problem of rank ordering is approached from the pairwise preference angle. For each pair of class labels a separate model is created, based on training instances for which preferences among labels are known. The models are trained with the use of the decision tree learner C4.5. The proposed solution does not take into account the strength of preference between two labels. It has a binary output of a form "label l_i is preferred over l_j or label l_i is not preferred over l_j", which may lead to omission of some important information. The ranking in this case is obtained by a voting scenario: each time a label is preferred over another one it gets one vote: the more votes it has, the higher the rank it gets. All experimental results are based on artificial data.

In [5] a methodology based on pairwise preferences for ordinal regression is proposed. Ordinal regression is a problem which lies between multi-class classification and regression — there is a fixed number of rank labels which are to be associated with each instance, and there is an ordering among these labels, i.e. misclassifying an instance into an adjacent class is less harmful than into a remote one. The proposed framework based on Support Vector Machine classification methodology is (unlike statistical models) distribution independent. For a pair of instances a preference is obtained via large margin classifier, i.e. one class contains pairs whose first member is preferred over the second one and the other class contains pairs where the second instance should be ranked ahead of the first one. Similar to the approach in [10] the methodology does not take into account the strength of preference. The theory is tested on artificial data and also applied into Information retrieval problem.

There are several applications of a learning to rank order methodology. One of them is the area of feature selection. In the real world availability of a vast variety of features does not necessarily provide better outcomes [11]. Feature ranking can be regarded as one of the feature selection methods [12]. This technique can be applied in learning in Bayesian networks to reduce its computational complexity. The method proposed in [12] using a feature ranking algorithm improves the learning of a Bayesian network classifier from data. An algorithm for learning the structure and conditional probability distribution of directed probabilistic models based on ordering-search for Bayesian networks is proposed in [13]. According to experimental findings in [14] ordering-search performed on a space of orderings is at least as good as (and more time efficient) than structure-search for learning Bayesian networks. The algorithm in [13] incorporates these results and performs ordering-search for directed probabilistic models. The optimal ordering is searched with the use of relational regression trees which predict the gain in likelihood of a training dataset. An industrial application of rankings is presented in [15] where the problem of creating a global ranking of products based on partial rankings is addressed. The partial rank-

ings illustrate different aspects of the performance of various products and a technique to merge such a set of rankings into a global one is introduced.

3. Learning to rank order based on pairwise preference function

Many researchers agree that rankings based on preference judgements are potentially sharper than those based on a performance measure of each single instance to be rank ordered [3]. In real world applications expert opinions about pairwise preferences are usually not available. The lack of ready-to-use preference judgements can, however, be overcome with the use of a preference function. Such a function can be obtained if some training data are available.

3.1. Data and notation

Let us consider a number of sets $S_1, \ldots S_m$, each containing instances deriving from different domains and following different distributions,

$$S_i = \{(X_{i_1}, Y_{i_1}), \cdots, (X_{i_n}, Y_{i_n})\}. \quad (1)$$

Each X_{i_t} is a l-dimensional vector, we will refer to it as an independent variable; Y_{i_t} is a dependent variable associated with each independent variable, i.e. there exist functions $f_i : \mathbb{R}^l \to \mathbb{R}$, such that $Y_{i_t} = f_i(X_{i_t})$. The functions f_i are not known.

Let

$$P_t = \{(X_{1_t}, Y_{1_t}), \ldots, (X_{m_t}, Y_{m_t})\}, \quad (2)$$

$t \leqslant n$ be a set containing one representative instance from each set S_i. For each P_t we are looking for an ordering of instances (X_{i_t}, Y_{i_t}).

Let Λ_t be a model which rank orders pairs utilising information about X_{i_t}'s only, $i \leqslant m$. Such a model associates unique rank label $r^i_{\Lambda_t} \in \{1, \ldots, m\}$ with each instance (X_{i_t}, Y_{i_t}). By $r^i_t \in \{1, \ldots, m\}$ we denote the (unique) actual rank label of the instance. The vector of actual rank labels $\mathbf{r}_t =< r^1_t, \ldots, r^m_t >$ satisfies the following constraint:

$$r^i_t < r^j_t \Leftrightarrow Y_{i_t} \geqslant Y_{j_t}. \quad (3)$$

We are looking for a model Λ^*_t which minimises the mean absolute error between the predicted and actual rankings:

$$E(\mathbf{r}_t, \mathbf{r}_{\Lambda^*_t}) = \frac{1}{m} \sum_{i=1}^{m} |r^i_t - r^i_{\Lambda^*_t}|. \quad (4)$$

3.2. Pairwise preference function

We approach the problem of learning to rank order from the pairwise preference angle. Such preferences constitute the partial order of the data which can be translated into the total order, e.g. with the use of the greedy algorithm from [9].

In this work we concentrate on the problem of deriving pairwise preferences from historical data. Our hypothesis is that learning additional information about the strength of such preferences (and not only their direction) can reduce the mean absolute error between predicted and actual rankings. Our aim is to improve the formulation of the problem based on binary classification methodology [10] and define it as a regression problem. We also do not want to impose any constraints regarding structure and distribution of the data, i.e. we search for a flexible and, unlike [6], distribution-independent model.

We define the preference function $Pref : P_t \times P_t \to \mathbb{R}$ and impose the following constraints:

- if $Pref(X_{i_t}, X_{j_t}) > 0$, X_{i_t} is preferred over X_{j_t};
- if $Pref(X_{i_t}, X_{j_t}) < 0$, X_{j_t} is preferred over X_{i_t};
- if $Pref(X_{i_t}, X_{j_t}) = 0$, there is no preference between X_{i_t} and X_{j_t}.

Therefore the sign of the function indicates which of its arguments is preferred over the other one, and the absolute value of the function represents the strength of the preference.

The graph on the left hand-side of Fig. 1 shows binary preferences among five data items. Using the voting (where an instance gets when it is preferred over another instance) the following ranking is induced by the given preferences: $B \to C, D, E \to A$. There is no distinction between instances C, D and E.

Now let us assume that we obtained additional information about strength of preferences. On the

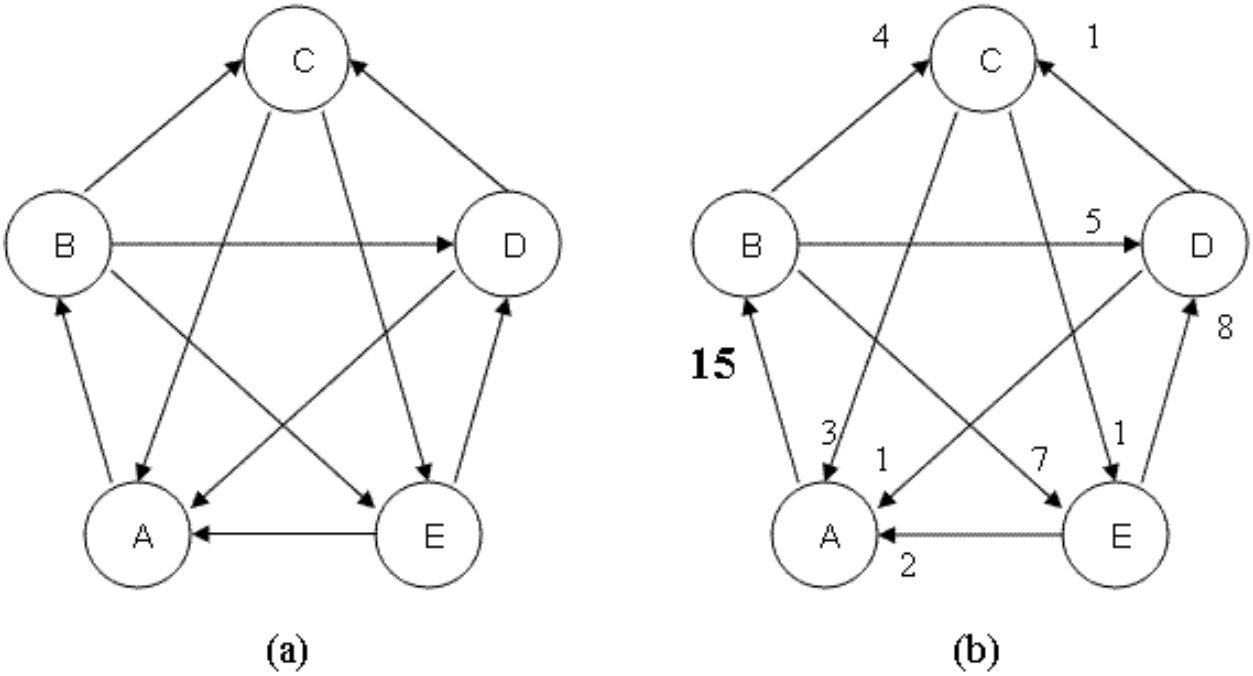

Figure 1: Pairwise dependencies: without (a) and with (b) the level of preference

graph on the right hand-side of Figure 1 these values are represented as weights of directed edges. Given the strong preference of instance A over B and weak preference of the remaining instances over A, ranking A in last place becomes questionable. (Note that we do not impose any constraints on consistency of preferences, therefore the preference relation is not transitive.)

There are various techniques for turning partial order into total order. In this work we use a greedy algorithm from [9]. The total order on the left hand-side of Figure 1 is derived as: A,B,E,D,C. It should be noted that employing the level of dominance in the pairwise preference learning makes the occurrence of ties less likely.

3.3. Preference training

It should be noted that in our problem formulation the instances to be rank ordered derive from different domains, therefore their preferences should be modelled independently for each pair. Based on training (historical data) we build regression models for all $m(m-1)/2$ pairs from the m domains.

In our training scheme the actual preferences for each training pair $\{(X_{i_t},Y_{i_t}),(X_{j_t},Y_{j_t})\}$ are obtained as follows:

$$F((X_{i_t},Y_{i_t}),(X_{j_t},Y_{j_t})) = Y_{i_t} - Y_{j_t}. \qquad (5)$$

We obtain $m(m-1)$ models for predicting the preference $Pref_{ij}(X_{i_t},X_{j_t})$ for unseen instances whose dependant variables Y_{i_t}, Y_{j_t} are not (yet) known.

Pairwise preferences obtained with the trained models constitute a partial order of instances. The total order (ranking) is then obtained with the use of the greedy algorithm proposed by William Cohen in [9]. Models are evaluated with the use of mean absolute error as in (4).

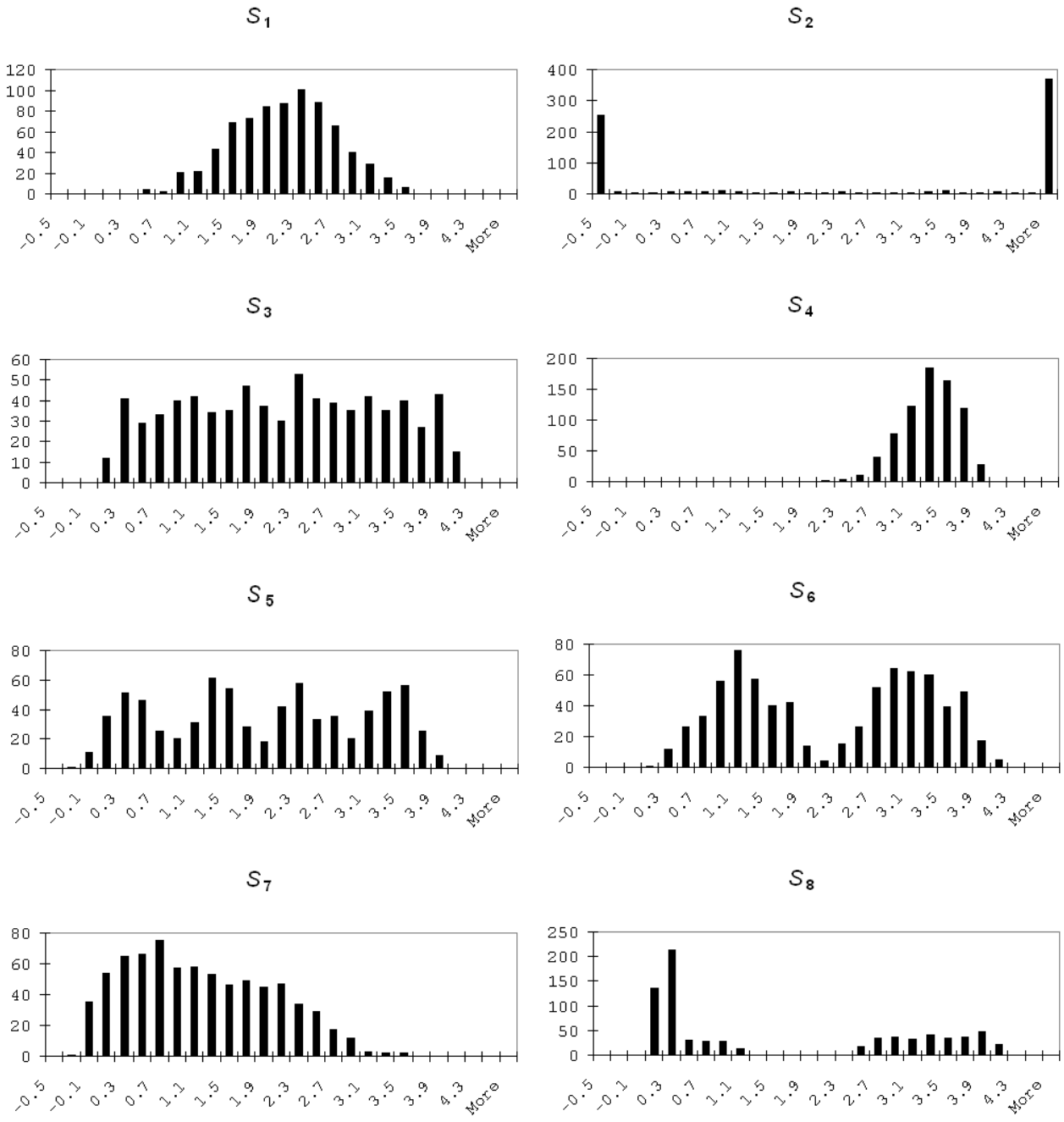

Figure 2: Histograms of dependant variables for artificial datasets used for experimental evaluation

4. Experimental results

In this section we present initial results of our studies on employing the stregth of preference into ranking construction model.

4.1. Data deriving from different domains

In this section we present results of applying our methodology on data deriving from different domains. The experiments were conducted on artficial data and on stock market data (Dow Jones Industrial Average index components).

4.1.1. Artificial data

We generated the experimental data artificially. It consists of eight datasets, each containing 750 instances, whose independent variables $X_{i_t} =< x_{i_{t_1}}, \ldots x_{i_{t_4}} > \in \mathbb{R}^4$ are 4-dimensional vectors with $x_{i_{t_k}}$ randomly generated from $U(0,1)$. Dependant variables $f_i(X_{i_t})$ have different ranges and follow different distributions for different sets. We used the following functions to generate dependant variables:

1. $f_1(X_{1_t}) = x_{1_{t_1}} + x_{1_{t_2}} + x_{1_{t_3}} + x_{1_{t_4}}$

2. $f_2(X_{2_t}) = \dfrac{2 \times x_{2_{t_1}} \times x_{2_{t_2}}}{x_{2_{t_3}} \times x_{2_{t_4}} + 0.00001} - 1$

3. $f_3(X_{3_t}) = \begin{cases} 4 \times x_{3_{t_2}}, & x_{3_{t_1}} \leqslant 0.33 \\ 4 \times x_{3_{t_3}}, & 0.33 < x_{3_{t_1}} \leqslant 0.67 \\ 4 \times x_{3_{t_4}}, & 0.67 < x_{3_{t_1}} \end{cases}$

4. $f_4(X_{4_t}) = \sqrt[4]{x_{4_{t_1}}} + \sqrt[4]{x_{4_{t_2}}} + \sqrt[4]{x_{4_{t_3}}} + \sqrt[4]{x_{4_{t_4}}}$

5. $f_5(X_{5_t}) = \begin{cases} 0.5 \times (x_{5_{t_1}} + x_{5_{t_2}} + x_{5_{t_3}} - 1), & x_{5_{t_4}} \leqslant 0.25 \\ 0.5 \times (x_{5_{t_1}} + x_{5_{t_2}} + x_{5_{t_3}} + 1), & 0.25 < x_{5_{t_4}} \leqslant 0.5 \\ 0.5 \times (x_{5_{t_1}} + x_{5_{t_2}} + x_{5_{t_3}} + 3), & 0.5 < x_{5_{t_4}} \leqslant 0.75 \\ 0.5 \times (x_{5_{t_1}} + x_{5_{t_2}} + x_{5_{t_3}} + 5), & 0.75 < x_{5_{t_4}} \end{cases}$

6. $f_6(X_{6_t}) = \begin{cases} x_{6_{t_1}} + x_{6_{t_3}}, & x_{6_{t_2}} \leqslant 0.5 \\ x_{6_{t_3}} + x_{6_{t_4}} + 2, & 0.5 < x_{6_{t_2}} \end{cases}$

7. $f_7(X_{7_t}) = \begin{cases} x_{7_{t_4}} - 0.3, & x_{7_{t_1}} \leqslant 0.25 \\ x_{7_{t_4}} + x_{7_{t_3}} - 0.3, & 0.25 < x_{7_{t_1}} \leqslant 0.5 \\ x_{7_{t_4}} + x_{7_{t_3}} + x_{7_{t_2}} - 0.3, & 0.5 < x_{7_{t_1}} \leqslant 0.75 \\ x_{7_{t_4}} + x_{7_{t_3}} + x_{7_{t_2}} x_{7_{t_1}} - 0.3, & 0.75 < x_{7_{t_1}} \end{cases}$

8. $f_8(X_{8_t}) = \begin{cases} 0.25 \times x_{8_{t_2}}, & x_{8_{t_1}} \leqslant 0.4 \\ x_{8_{t_3}}, & 0.4 < x_{8_{t_1}} \leqslant 0.6 \\ 4 \times x_{8_{t_1}}, & 0.6 < x_{8_{t_1}} \end{cases}$

We tested the response to increasing noise of our methodology compared with learning to rank order based on binary preferences. We constructed five experimental scenatios, where different amount of noise was added to dependant variables. The noise follows uniform distribution $U(-\nu\sigma_i, \nu\sigma_i)$, $i \leqslant m$ where σ_i is the standard deviation of the values generated by the related dependant variable in the set S_i, and $\nu < 1$ is a common factor for all sets. We repeated experiments with the following values of ν corresponding to 10%, 20% and 30%. Figure 2 shows histograms of dependant variables for each dataset with 10% noise. Our aim was to design sets with different characteristics and behaviours.

4.1.2. Stock market data

Similar experiments have been conducted for the stock market data. Thirty DOW index components have been used in the experiment. Five three-year-long experimental periods have been taken into consideration, where the first two years have been used for training and the last year for testing. The first experimental period was from 1 January 2004 until 31 December 2006, the successive periods beginning four months later (1 April 2004, 1 July 2004, etc.) On each trading day the ranking of stocks was created based on pairwise preferences between vectors of five technical indicators:

1. $s_{i_1}(t)$ — Relative Strength Index (RSI),
   $\text{RSI} = 100 - \dfrac{100}{1 + \text{RS}}$, (RS = relative strength)

SUPPORT VECTOR CLASSIFIER	with level of dominance (RBFK)	binary preference (RBFK)	with level of dominance (NPK)	binary preference (NPK)
10% noise	1.127	1.528	1.111	1.528
20% noise	1.190	1.547	1.198	1.547
30% noise	1.221	1.557	1.266	1.557
40% noise	1.238	1.569	1.297	1.569
50% noise	1.264	1.572	1.333	1.572
DOW	9.93	9.98	9.95	9.98

Table 1: Preference with level of dominance vs. binary preference — mean absolute error with support vector classifier with Radial Basis Function Kernel (RBFK) and Normalised PolyKernel (NPK)

$$\text{RS} = \frac{\text{average gain in previous } x \text{ trading days}}{\text{average loss in previous } x \text{ trading days}}$$

2. $s_{i_2}(t)$ — Money Flow Index (MFI),
$\text{MFI} = 100 - \frac{100}{1+\text{MR}}$, (MR = money ratio)
$\text{MR} = \frac{\text{positive money flow}}{\text{negative money flow}}$,
$\text{money flow} = \frac{\text{high}+\text{low}+\text{close}}{3} \times \text{volume}$

3. $s_{i_3}(t)$ — Fast Stochastic Oscillator (%K)
$\%\text{K} = \frac{\text{close} - \text{lowest in past } x \text{ days}}{\text{highest in past } x \text{ days} - \text{lowest in past } x \text{ days}}$

4. $s_{i_4}(t)$ — Bollinger Band (%b)
$\%\text{b} = \frac{\text{close} - \text{lowerBB}}{\text{upper BB} - \text{lower BB}}$

5. $s_{i_5}(t)$ — Bollinger Band Width (BBW),
$\text{BBW} = \frac{\text{upper BB} - \text{lower BB}}{\text{middle BB}}$

Detailed discussion on technical indicators can be found in [16,17,19].

The predicted ranking was compared with the actual ranking based on one day returns.

4.1.3. Results

Due to the high range of preference function values for some pairs, the actual preference defined in (5) has been reformulated into:

$$F^*((X_{i_t}, Y_{i_t}), (X_{j_t}, Y_{j_t})) = sgn(Y_{i_t} - Y_{j_t})\sqrt{|Y_{i_t} - Y_{j_t}|}.$$

to improve regression training.

All the experiments were performed with the use of Weka 3.6.1, datamining software [18], with three-fold cross validation. We used different machine learning methodologies for classification and regression to compare rankings based on binary preferences with these based on preferences with level of dominance. Configurations of classification and regression functions can be found in Appendix A.

It should be noted, that the expected value of mean absolute error for randomly assigned rank labels for m objects is

$$\text{E}_m = \text{E}(E(\mathbf{r}, \mathbf{r}_{\Lambda^*})) = \frac{1}{m^2} \times \sum_{k=1}^{m} (\sum_{i=0}^{m-k} i + \sum_{i=0}^{k-1} i). \quad (6)$$

Derivation of the formula can be found in Appendix B. Hence in our experiments $\text{E}_5 = 1.6$. (The maximum absolute mean error for ranking of five objects is $E_{\max,5} = 2.4$.)

Experiments were performed with the use of support vector classifier with Radial Basis Function Kernel and Normalised PolyKernel for both the binary preference approach and that using preference with a level of dominance. Table 1 contains mean errors as defined in (4). Results obtained with the use of Radial Basis Function Kernel are marked (RBFK) and (NPK) refers to experiments with the use of Normalised PolyKernel. In all experimental cases our methodology yields better results, specifically smaller mean rank label error as in (4), for the data with and without the noise.

4.1.4. Unbalanced preferences

It should be noted that error in identifying the direction of preference for one pair only can influence the total order substantially. Therefore it is crucial to minimise the number of pairs for which the direction has been misclassified.

Binary classification algorithms often struggle with unbalanced data, i.e. when in terms of size one class substantially dominates the other (e.g. when one class contains over 90% of all training instances). In such cases when every unseen instance is classified as a member of the larger class high accuracy is likely.

For example this is the case for the preference between sets S_4 and S_7. As is clear from Figure 2, S_4 should dominate over S_7 for the vast majority of instances. Preferences learned with binary classifiers assigned preference of an instance from S_4 over one from S_7 for every single training pair.

When the model learns the strength of preference in addition to the binary pairwise preference, it is more likely that it preserves correct direction of preference for members of the smaller class, and our experiments support this claim.

4.2. Data deriving from one domain

Additional experiments have been performed on numeric regression datasets. When all data derive from the same domain, only one preference function has to be trained. In our experiments we used only datasets with continous attributes and without missing values (all entries with missing values were removed). For each set 10 independent experiments were conducted, where a random third of instanced were used for testing, and all remaining instances were used for training.

Table 2 contains mean errors for results obtained with the use of Radial Basis Function Kernel (marked RBFK) and Normalised PolyKernel (marked NPK). In the majority of cases, the approach utilising the level of preference yields better results.

[*] http://www.cs.waikato.ac.nz/ml/weka/

[†] http://www.liaad.up.pt/ ltorgo/Regression/DataSets.html

5. Conclusions and future work

We proposed a methodology for learning to rank order based on pairwise preferences with a level of dominance. We evaluated this methodology against the approach based on binary preferences with the use of Support Vector Machines. We conducted our experiments on artificially generated datasets which followed different distributions, and also where unbalanced preferences over domains were present. Additional experiments were conducted on stock market data as well as numeric regression datasets.

In the case of artificial data, our methodology always outperformed the one based on binary preferences for noisy data. In stock market experiment, the method employing the strength of preference outperforms the binary one, however it should be noted that results for both methods are very close to random. Experiments on regression datasets confirm advantages of utilising streght of preference in the ranking process.

In the future we will perform additional experiments with the use of additional machine learning techniques for classification and regression. We will also apply the methodology of learning to rank order based on pairwise preferences with level of dominance to financial portfolio construction and updating. Also the problem of computational compexity will be addressed.

Appendix A

A.1. Experimental configurations in Weka 3.6.1

Support Vector Classifier (for classification) with RBFK:
weka.classifiers.functions.SMO -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K ”weka.classifiers.functions.supportVector.RBFKernel -C 250007 -G 0.01”

Support Vector Classifier (for regression) with RBFK:
weka.classifiers.functions.SMOreg -C 1.0 -N 0 -I

dataset	with level of dominance (RBFK)	binary preference (RBFK)	with level of dominance (NPK)	binary preference (NPK)
autoPrice *	4.63	4.53	4.53	4.53
baskball *	7.1	7.21	7.37	7.22
bolts *	2.43	2.98	2.14	2.98
gascons *	0.96	2.27	0.58	2.27
pollution *	2.87	3.45	3.17	3.45
pwLinear *	6.06	7.21	6.72	7.21
vineyard *	2.76	3.2	2.71	3.2
diabetes †	3.64	4.57	3.49	4.57
machine †	6.68	7.28	8.12	7.28
pyrim †	3.52	4.28	3.78	4.28

Table 2: Preference with level of dominance vs. binary preference — mean absolute error with support vector classifier with Radial Basis Function Kernel (RBFK) and Normalised PolyKernel (NPK)

"weka.classifiers.functions.supportVector.RegSMO Improved -L 0.0010 -W 1 -P 1.0E-12 -T 0.0010 -V" -K "weka.classifiers.functions.supportVector. RBFKernel -C 250007 -G 0.01"

Support Vector Classifier (for classification) with NPK:
weka.classifiers.functions.SMO -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector. NormalizedPolyKernel -C 250007 -E 2.0"

Support Vector Classifier (for regression) with NPK:
weka.classifiers.functions.SMOreg -C 1.0 -N 0 -I "weka.classifiers.functions.supportVector.RegSMO Improved -L 0.0010 -W 1 -P 1.0E-12 -T 0.0010 -V" -K "weka.classifiers.functions.supportVector. NormalizedPolyKernel -C 250007 -E 2.0"

A.2. Derivation of expected error formula

Let us assume (without the loss of generality) that the actual rank order is r $=< 1,2,\ldots,m >$. Let us take into consideration the k-th instance ($k \leqslant m$) whose actual rank label $r_k = k$. There are $(m-1)!$ orderings (permutations) where the predicted rank of a k-th instance $r_k^{pred} = 1$, generating the error of $|k-1|$. Similarly there are $(m-1)!$ rank orderings where the k-th instance has a predicted rank label $r_k^{pred} = l$, generating the error of $|k-l|$, where $l \leqslant m$. Hence for all possible permutations the k-th instance generates the following mean absolute error: $\frac{1}{m}(m-1)!((k-1)+\ldots+1+0+1+\ldots+(m-k))$.

Therefore the expected value of the mean absolute error is:

$$\mathrm{E}_m = \frac{1}{m!} \times \frac{1}{m} \times (m-1)!\Big(\big(0+1+\ldots+(n-1)\big)+ +\big(1+0+1+\ldots+(n-2)\big)+\ldots+\big((n-2)+(n-3)+\ldots+1+0+1\big)+\big((n-1)+(n-2)+\ldots+1+0\big)\Big) = \frac{(m-1)!}{m\times m!} \times \sum_{k=1}^{m}(\sum_{i=0}^{m-k} i + \sum_{i=0}^{k-1} i)$$

$$= \frac{1}{m^2} \times \sum_{k=1}^{m}(\sum_{i=0}^{m-k} i + \sum_{i=0}^{k-1} i).$$

1. J. Goldsmith and U.Junker, "Preference handling for artificial intelligence," *AI Magazine*, **29**(4), 9–12 (2008).
2. C. Kahraman and A. C. Tolga, "An Alternative Ranking Approach and Its Usage in Multi-Criteria Decision-Making," *International Journal of Computational Intelligence Systems*, **2**(3), 219–235 (2009).
3. Y. Hochberg and R. Rabinovitch, "Ranking by pairwise comparisons with special reference to ordering portfolios," *American Journal of Mathematical and Management Sciences*, **20**, 231–253 (2000).
4. J. Furnkranz and E. Hullermeier, "Preference learning," *Kunstliche Intelligenz*, **19**(1), 60–61 (2005).
5. R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," In *International Conference on Artificial Neural Networks*, 97–102 (1999).
6. W. Chu and Z. Ghahramani, "Preference learning with

gaussian processes," In *Proc. ICML*2005*, 137–144 (2005).
7. H. Buhlmann and P. J. Huber, "Pairwise comparison and ranking in tournaments," *The Annals of Mathematical Statistics*, **34**(2), 501 – 510 (1963).
8. W. W. Cohen, R. E. Schapire, and Y. Singer, "Learning to order things," In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, 451–457 (1998).
9. W. W. Cohen, R. E. Schapire, and Y. Singer, "Learning to order things," *Journal of Artificial Intelligence Research*, **10**, 243–270 (1999).
10. J. Furnkranz and E. Hullermeier, "Pairwise preference learning and ranking," In *Proceedings of the 14th European Conference on Machine Learning*, 145–156 (2003).
11. D. Koller and M. Sahami, "Toward optimal feature selection," In *Proceedings of the 13th International Conference on Machine Learning*, 284–292 (1996).
12. E. R. Hruschka. and N. F. F. Ebecken, "Towards efficient variables ordering for bayesian networks classifier," *Data Knowl. Eng.*, **3**(2), 258–269 (2007).
13. J. Ramon, T. Croonenborghs, D. Fierens, H. Blockeel, and M. Bruynooghe, "Generalized ordering-search for learning directed probabilistic logical models," *Mach. Learn.*, **70**(2-3), 169–188 (2008).
14. M. Teyssier and D. Koller, "Ordering-based search: A simple and effective algorithm for learning bayesian networks," In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 584–590 (2005).
15. L. Corain and L. Salmaso, "A non-parametric method for defining a global preference ranking of industrial products," *Journal of Applied Statistics*, **34**(2), 203–216 (2007).
16. BollingerBands. http://www.bollingerbands.com.
17. M. J. Pring, *Technical Analysis Explained*, McGraw-Hill, Inc., (1991).
18. I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, San Francisco, 2nd edition edition, (2005).
19. StockCharts.com. http://stockcharts.com/school/doku.php?id=chart_school.