MINLIP: Efficient Learning of Transformation Models *

V. Van Belle, K. Pelckmans, J.A.K. Suykens and S. Van Huffel

Katholieke Universiteit Leuven, ESAT-SCD
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
{vvanbell,kpelckma,johan.suykens,vanhuffe}
@esat.kuleuven.be

Abstract. This paper studies a risk minimization approach to estimate a transformation model from noisy observations. It is argued that transformation models are a natural candidate to study ranking models and ordinal regression in a context of machine learning. We do implement a structural risk minimization strategy based on a Lipschitz smoothness condition of the transformation model. Then, it is shown how the estimate can be obtained efficiently by solving a convex quadratic program with $O(n)$ linear constraints and unknowns, with n the number of data points. A set of experiments do support these findings.

Key words: support vector machines, ranking models, ordinal regression

1 Introduction

Non-linear methods based on ranking continue to challenge researchers in different scientific areas, see e.g. [5, 7]. Problems of learning ranking functions come in different flavors, including ordinal regression, bipartite ranking and discounted ranking studied frequently in research on information retrieval. This problem will be considered in the context of Support Vector Machines (SVM) [11, 12, 14] and convex optimization. We study the general problem where the output domain can be arbitrary (with possibly infinite members), but possess a natural ordering relation between the members. This general problem was studied before in [1, 7], and results can be specified to the aforementioned specific settings by proper definition of the domain of the outputs (e.g. restricting its cardinality to $k < \infty$ or $k = 2$).

A main trend is the reduction of a ranking problem to a pairwise classification problem, bringing in all methodology from learning theory. It may however be argued that such an approach deflects attention from the real nature of the ranking problem. It is for example not clear that the complexity control (in a broad sense) which is successful for classification problems is also natural and efficient in the ranking setting. More specifically, it is often taken for granted that the measure of margin - successful in the setting

* KP is a postdoctoral researcher with FWO Flanders (A 4/5 SB 18605). S. Van Huffel is a full professor and J.A.K. Suykens is a professor at the Katholieke Universiteit Leuven, Belgium. This research is supported by GOA-AMBioRICS, CoE EF/05/006, FWO G.0407.02 and G.0302.07, IWT, IUAP P6/04, eTUMOUR (FP6-2002-LIFESCIHEALTH 503094

of binary classification - has a natural counterpart in the ranking setting as the measure of *pairwise margin*, although it remains somewhat arbitrary how this is to be implemented exactly, see e.g. [4]. In order to approach such questions, we take an alternative approach: we will try to learn a single function $u : \mathbb{R}^d \to \mathbb{R}$, such that the natural order on $\mathbb{R}$ induces the desired ranking (approximatively). Such a function is often referred to as a scoring, ranking, utility or health function depending on the context - we will use *utility function* in this text.

In the realizable case, techniques as complexity control, regularization or Occam's razor (in a broad sense) give a guideline to learn a specific function in case there are more functions exactly concordant with the observed data: a simpler function has a better chance of capturing the underlying relation. In short, we will argue that a utility function reproducing the observed order is less complex than another concordant function if the former is more smoothly related to the actual output values. That is, if there is an exact order relation between two variables, one can obviously find (geometrically) a monotonically increasing function between them. This argument relates ranking directly to what is well-studied in the statistical literature as transformation models, see e.g. [6, 9]. Here the monotonically increasing mapping between utility function and output is referred to as the transformation function. Now, we define the complexity of a prediction rule for transformation models as being the Lipschitz constant of this transformation function. When implementing a risk minimization strategy based on these insights, the resulting methods are similar to the binary, hard margin SVMs, but do differ conceptually and computationally with existing ranking approaches. Also similar in spirit to the non-separable case in SVMs, it is indicated how slack variables can be used to relax the realizable case: we assume that an exactly concordant function can be found, were it not for incomplete observation of the patients' covariates.

This paper is organized as follows. Section 2 discusses in some detail the use of transformation models and its relation with ranking methods. Section 3 introduces an efficient estimator of such a transformation function, relying on ideas as thoroughly used in the machine learning literature. Section 4 gives insight how our estimator can be modified in the context of ordinal regression. Section 5 reports experimental results supporting the approach.

2 Transformation Models and Ranking Methods

In order to make the discussion more formal, we adopt the following notation. We work in a stochastic context, so we denote random variables and vectors as capital letters, e.g. $X, Y, \ldots$, which follow an appropriate stochastic law $P_X, P_Y, \ldots$, abbreviated (generically) as P. Deterministic quantities as constants and functions are represented in lower case letters (e.g. $d, h, u, \ldots$). Matrices are denoted as boldface capital letters (e.g. $\mathbf{X}, \mathbf{D}, \ldots$). Now we give a definition of a *transformation model*.

Definition 1 (Transformation Model) *Let $h : \mathbb{R} \to \mathbb{R}$ be a* strictly *increasing function, and let $u : \mathbb{R}^d \to \mathbb{R}$ be a function of the covariates $X \in \mathbb{R}^d$. A Transformation Model (or TM) takes the following form*

$$Y = h(u(X)). \tag{1}$$

Let ϵ be a random variable ('noise') independent of X, with cumulative distribution function $F_\epsilon(e) = P(\epsilon \leq e)$ for any $e \in \mathbb{R}$. Then a Noisy Transformation Model (NTM) takes the form

$$Y = h(u(X) + \epsilon). \tag{2}$$

Now the question reads as how to estimate the utility function $u : \mathbb{R}^d \to \mathbb{R}$ and the transformation model h from i.i.d. samples $\{(X_i, Y_i)\}_{i=1}^n$ without imposing any distributional (parametric) assumptions on the noise terms $\{\epsilon_i\}$.

Transformation models are often considered in the context of failure time models and survival analysis [8]. It should be noted that the approach which will be outlined sets the stage for deriving predictive models in this context. Note that in this context [3, 6, 9] one considers transformation models of the form $h^-(Y) = u(X) + \epsilon$, which are equivalent in case h is invertible, or $h^-(h(z)) = h(h^-(z)) = z$ for all z.

The relation with empirical risk minimization for ranking and ordinal regression goes as follows. The risk of a ranking function with respect to observations is often expressed in terms of Kendall's τ, Area Under The Curve or a related measure. Here we consider the (equivalent) measure of *disconcordance* (or one minus concordance) for a fixed function $u : \mathbb{R}^d \to \mathbb{R}$, where the probability concerns the two i.i.d. copies (X, Y) and (X', Y'):

$$\mathcal{C}(u) = P((u(X) - u(X'))(Y - Y') < 0). \tag{3}$$

Given a set of n i.i.d. observations $\{(X_i, Y_i)\}_{i=1}^n$,

$$\mathcal{C}_n(u) = \frac{2}{n(n-1)} \sum_{i<j} I((u(X_i) - u(X_j))(Y_i - Y_j) < 0), \tag{4}$$

where the indicator function $I(z)$ equals one if z holds, and equals zero otherwise. Empirical Risk Minimization (ERM) is then performed by solving

$$\hat{u} = \arg\min_{u \in \mathcal{U}} \mathcal{C}_n(u), \tag{5}$$

where $\mathcal{U} \subset \{u : \mathbb{R}^d \to \mathbb{R}\}$ is an appropriate subset of ranking functions, see e.g. [5] and citations. This approach however results in difficult and combinatorial optimization problems, and the current solution is to majorize the discontinuous indicator function with the Hinge loss, i.e. $I(z) \leq \max(0, 1 - z)$ yielding rankSVM [7]. The intrinsic problem with such an approach is that one has $O(n^2)$ number of constraints or unknowns in the final optimization problem, obstructing applicability (computationally) to many real life cases.

Now, there is an intrinsic relation with transformation models which circumvent such problems. The crucial observation here (again) is that *if a function $u : \mathbb{R}^d \to \mathbb{R}$ exists such that $\mathcal{C}_n(u) = 0$, one describes implicitly a monotonically increasing transformation function* (see Figure 1). In the case that $\mathcal{C}_n(u) = 0$ is not satisfied, we will adopt the noisy transformation model and use the error terms (slack variables) to model the deviance from this assumption. This reasoning is entirely similar as is used in formulating the hard margin Support Vector Machine, and its soft-margin variation.

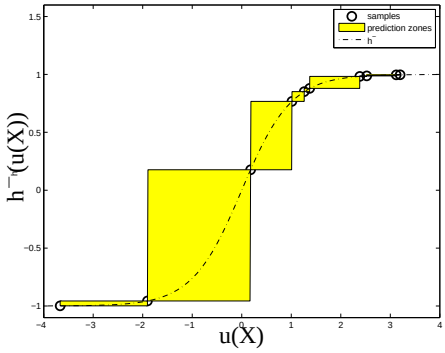


Fig. 1. The main observation relating ranking and transformation models is that if two variables $u(x)$ and y are perfectly *concordant*, they describe (implicitly) a monotonically increasing function $y = h(u(x))$. This means that a perfect ranking function corresponds with a (noiseless) transformation model. Moreover, if the samples are pairwise Lipschitz, there exists a Lipschitz transformation function. The yellow zones indicate possible function values on test samples.

3 MINLIP: a Convex Approach to Learning a Transformation Model

3.1 Lipschitz Smooth Functions and Transformation Models

In order to overcome the difficulties of implementing the estimator given in equation (5), we need one final ingredient. This concept will play a similar role as the margin in Support Vector Machines for classification. We will say that the univariate function h has a Lipschitz constant of $L \geq 0$ if $|h(z) - h(z')| \leq L|z - z'|$ for all $z, z' \in \mathbb{R}$, or equivalently

$$|h(u(x)) - h(u(x'))| \leq L\,|u(x) - u(x')|\,, \quad \forall x, x' \in \mathbb{R}^d. \tag{6}$$

Now, since h is monotonically increasing one has also $h(z) - h(z') \leq z - z'$ for all $z \geq z'$, and restricting attention to the samples $\{(x_i, y_i)\}_{i=1}^n$, one has the necessary and sufficient conditions $h(u(X_{(i)})) - h(u(X_{(i-1)})) \leq L\left(u(X_{(i)}) - u(X_{(i-1)})\right)$ for all $i = 2, \ldots, n$. Here, we assume that the data obey a noiseless transformation model (as in (1)), and the samples are reindexed as $\{(X_{(i)}, Y_{(i)})\}_{i=1}^n$ where $Y_{(i-1)} \leq Y_{(i)}$ for all $i = 2, \ldots, n$. Wrapping up results thus far gives us the following proposition:

Proposition 1 (Existence of h) *Given a set of samples $\{(X_{(i)}, Y_{(i)})\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ and a function $u : \mathbb{R}^d \to \mathbb{R}$, such that $Y_{(i)} \geq Y_{(i-1)}$ for all $i = 2, \ldots, n$, and*

$$Y_{(i)} - Y_{(i-1)} \leq L\left(u(X_{(i)}) - u(X_{(i-1)})\right), \quad \forall i = 2, \ldots, n, \tag{7}$$

Then there exists a monotonically increasing function $h : \mathbb{R} \to \mathbb{R}$ such that the mapping x to y obeys $y = h(u(x))$ and h has Lipschitz constant L following (6) (see Figure 1).

Before using non-linear utility functions, we will consider only linear utilities in the next two sections.

3.2 Kernel based Model

Since the function $u(x) = w^T x$ can be arbitrarily rescaled such that the corresponding transformation function has arbitrary Lipschitz constant (i.e. for any $c > 0$, one has $h(u(x)) = h'(u'(x))$ where $h'(z) = h(c^{-1}z)$ and $u'(x) = cu(x)$), we fix the norm $w^T w$ and try to find $u(x) = v^T x$ with $v^T v = 1$. Hence learning a transformation model with minimal Lipschitz constant of h can be written as

$$\min_{v,L} L^2 \text{ s.t. } \|v\|_2 = 1,\ Y_{(i)} - Y_{(i-1)} \leq L\left(v^T X_{(i)} - v^T X_{(i-1)}\right),\ \ \forall i = 2, \ldots, n \quad (8)$$

and equivalently substituting $w = Lv$ as

$$\min_{w} \frac{1}{2} w^T w \quad \text{s.t. } Y_{(i)} - Y_{(i-1)} \leq w^T X_{(i)} - w^T X_{(i-1)},\ \ \forall i = 2, \ldots, n \qquad (9)$$

which goes along similar lines as the hard margin SVM (see e.g.[11]). Remark that there is no need for an intercept term here. Observe that this problem has $n-1$ linear constraints. We will refer to this estimator of w as MINLIP. We can rewrite this problem compactly as

$$\min_{w} \frac{1}{2} w^T w \quad \text{s.t.} \quad \mathbf{D}\mathbf{X}w \geq \mathbf{D}\mathbf{Y}, \qquad (10)$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a matrix with each row containing a sample, i.e. $\mathbf{X}_i = X_{(i)} \in \mathbb{R}^d$, $\mathbf{Y}_i = Y_{(i)} \in \mathbb{R}$. The matrix $\mathbf{D} \in \{-1, 0, 1\}^{(n-1)\times n}$ gives the first order differences of a vector, i.e. assuming no ties in the output, $\mathbf{D}_j \mathbf{Y} = Y_{(j+1)} - Y_{(j)}$ for all $j = 1, \ldots, n-1$, with $\mathbf{D}_j$ the jth row of $\mathbf{D}$. In the presence of ties $Y_{(j+1)}$ is replaced by $Y_{(i)}$, with i the smallest output value with $Y_{(i)} > Y_{(j)}$. Solving this problem as a convex QP can be done efficiently with standard mathematical solvers as implemented in MOSEK [1] or R-quadprog [2].

3.3 The Agnostic Case

The agnostic case deals with the case where one is not prepared to make the assumption that a function exists which will exactly extract in all cases the most relevant element. To model this, we impute a random variable ϵ with expected value zero, which acts additive on the contribution of the covariates (hence nonadditive on the final output for general function h). Hence our model becomes

$$Y = h(u(X) + \epsilon) = h(w^T X + \epsilon), \qquad (11)$$

as in (2). Now we suggest how one can integrate the agnostic learning scheme with the Lipschitz-based complexity control. We will further specify the loss function $\ell : \mathbb{R} \to \mathbb{R}$ to the absolute value loss, or $\ell(\epsilon) = |\epsilon|$. The reason for doing so is threefold. At first, this loss function is known to be more robust to model misspecification and outliers (leverage points) than e.g. the squared loss $\ell(\epsilon) = \epsilon^2$. Secondly, this loss will result

[1] http://www.mosek.org

[2] http://cran.r-project.org/web/packages/quadprog/index.html

in sparse terms, i.e. many of the estimated error terms will be zero. This in turn can be exploited in order to obtain a compact representation of the estimate through the dual (as is the case for Support Vector Machines (SVMs) [14], and see the following subsection). Thirdly, the one-norm loss is found to perform well in the binary classification case as implemented in the SVMs. However, we stress that the choice of this loss is in some sense arbitrary, and should be tailored to the case study at hand. One can formalize the learning objective for a fixed value of $\gamma > 0$ with errors $\epsilon = (\epsilon_1, \ldots, \epsilon_{n-1})^T \in \mathbb{R}^{n-1}$:

$$\min_{w,\epsilon} \frac{1}{2} w^T w + \gamma \|\epsilon\|_1 \quad \text{s.t.} \quad \mathbf{D}(\mathbf{X}w + \epsilon) \geq \mathbf{DY}, \tag{12}$$

where $\|\epsilon\|_1 = \sum_{i=1}^{n} |\epsilon_i|$. This problem can again be solved as a convex quadratic program.

3.4 A Nonlinear Extension using Mercer Kernels

Consider the model

$$u(x) = w^T \varphi(x), \tag{13}$$

where $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_\varphi}$ is a mapping of the data to a high dimensional feature space (of dimension d_φ, possibly infinite). Now $w \in \mathbb{R}^{d_\varphi}$ is a (possibly) infinite dimensional vector of unknowns. Let $\mathbf{\Phi} = [\varphi(X_{(1)}), \ldots, \varphi(X_{(n)})]^T \in \mathbb{R}^{n \times d_\varphi}$. Then we can write the learning problem concisely as

$$\min_{w} \frac{1}{2} w^T w \quad \text{s.t.} \quad \mathbf{D\Phi}w \geq \mathbf{DY}, \tag{14}$$

with the matrix $\mathbf{D}$ defined as before. This problem can be solved efficiently as a convex Quadratic Programming (QP) problem. The Lagrange dual problem becomes

$$\min_{\alpha} \frac{1}{2} \alpha^T \mathbf{DKD}^T \alpha - \alpha^T \mathbf{DY} \quad \text{s.t.} \quad \alpha \geq 0_{n-1} \tag{15}$$

where the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ contains the kernel evaluations such that $\mathbf{K}_{ij} = \varphi(X_i)^T \varphi(X_j)$ for all $i, j = 1, \ldots, n$. The estimated $\hat{u}$ can be evaluated at any point $x \in \mathbb{R}^d$ as

$$\hat{u}(x) = \hat{\alpha}^T \mathbf{D} K_n(x), \tag{16}$$

where $\hat{\alpha}$ solves (15), and $K_n(x) = (K(X_1, x), \ldots, K(X_n, x))^T \in \mathbb{R}^n$. A similar argument gives the dual of the agnostic learning machine of Subsection 3.3 (12), see e.g. [11, 12, 14]:

$$\min_{\alpha} \frac{1}{2} \alpha^T \mathbf{DKD}^T \alpha - \alpha^T \mathbf{DY} \quad \text{s.t.} \quad \begin{cases} -\gamma 1_n \leq \mathbf{D}^T \alpha \leq \gamma 1_n \\ \alpha \geq 0_{n-1}\,, \end{cases} \tag{17}$$

with $\mathbf{K}$ as above and the resulting estimate can be evaluated as in (16) without computing explicitly $\hat{w}$. It is seen that the nonlinear model can be estimated using a pre-defined kernel function, and without explicitly defining the mapping $\varphi(\cdot)$.

4 Learning for Ordinal Regression

Consider now the situation where the output takes a finite number of values - say $k \in \mathbb{N}$ - and where the k different classes possess a natural ordering relation. Instead of ranking all samples with its closest sample, one has to enumerate the rankings of all samples with certain output levels with all samples possessing the closest non-equal output level. However, when only observing a constant number k different output levels, this procedure can increase the number of constraints in the estimation problem to $\mathcal{O}(n^2)$. To cope with this issue, we introduce unknown thresholds $\{v_j\}_{j=1}^{k-1}$ on the utility function, corresponding with known output levels $z_j = Y^j + \frac{1}{2}(Y^{j+1} - Y^j)$. This implies that one has to compare each sample only twice, namely with thresholds z_j and z_{j+1} for each data point in class j. This problem can be formulated as

$$\min_{\overline{w},\epsilon} \frac{1}{2} w^T w + \gamma \|\epsilon\|_1 \quad \text{s.t.} \quad \begin{cases} \mathbf{D}(\overline{\mathbf{\Phi}}\overline{w} + \epsilon) \geq \mathbf{D}\overline{\mathbf{Y}}, \\ v_j \geq v_{j-1},\ \forall j = 2, \ldots, k-1\,, \end{cases} \tag{18}$$

with

$$\overline{w} = \begin{bmatrix} w \\ v \end{bmatrix} \qquad \overline{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi} & 0 \\ 0 & I \end{bmatrix} \qquad \overline{\mathbf{Y}} = \begin{bmatrix} \mathbf{Y} \\ z \end{bmatrix}, \tag{19}$$

where $\mathbf{D}$ needs to be build in such a way that $\mathbf{D}\overline{\mathbf{\Phi}}\overline{w}$ equals the difference between the utility of each point and the utility of the nearest threshold.

5 Application Studies

5.1 Ordinal Regression

In a first example 6 regression datasets[3] are used to compare the performance of the minlip model with two methods described in [4] (see Table 1). Both of these methods optimize multiple thresholds to define parallel discriminant hyperplanes for the ordinal levels. The first method (EXC) explicitly imposes the ordering of the thresholds, whereas this is done implicitly in the second method (IMC). Tuning of the Gaussian kernel parameter and the regularization parameter was performed with 10-fold cross-validation on an exponential grid. After an initial search, a finer search was performed in the neighborhood of the initial optimum. The datasets are divided into 20 folds with 10 equal-frequency bins, as in [4]. The generalization performance of the minlip method is clearly better than for the other methods. The IMC method performs best on the small dataset, but the minlip performance is better on larger datasets. Remark that the results on EXC and IMC obtained here are better than reported in [4].

In a second experiment, the performance and calculation time of the minlip model and standard rankSVM are compared on the pyrimidines dataset. Figure 2 shows the concordance, mean average error and calculation time when varying the number of training data points from 5 to 50. The concordance and error of both methods are comparable but for an increasing number of training data points the calculation time is considerably higher for the rankSVM method.

[3] These regression datasets are available at `http://www.liacc.up.pu/~ltorgo/Regression/DataSets.html`

Table 1. Test results of minlip, EXC and IMC using a Gaussian kernel. The targets of the datasets were discretized by 10 equal-frequency bins. The results are averaged over 20 trials.

dataset	mean zero-one error			mean absolute error		
	minlip	EXC	IMC	minlip	EXC	IMC
pyrimidines	0.65±0.09	0.70±0.09	**0.62 ± 0.07**	1.01±0.16	1.22±0.22	**1.00±0.12**
triazines	**0.66±0.06**	0.72 ±0.00	0.71±0.02	**1.19±0.12**	1.34±0.00	1.27±0.07
wisconsin	0.91±0.03	0.89±0.03	**0.88±0.03**	2.33±0.11	2.30±0.17	**2.25±0.13**
machine CPU	**0.36±0.04**	0.55±0.06	0.42±0.09	**0.54±0.09**	0.77±0.07	0.69±0.11
auto MPG	**0.49±0.04**	0.55±0.02	0.55±0.03	**0.62±0.14**	0.76±0.05	0.75±0.06
Boston housing	**0.44±0.04**	0.50±0.03	0.48±0.03	**0.54±0.08**	0.71±0.06	0.63±0.05

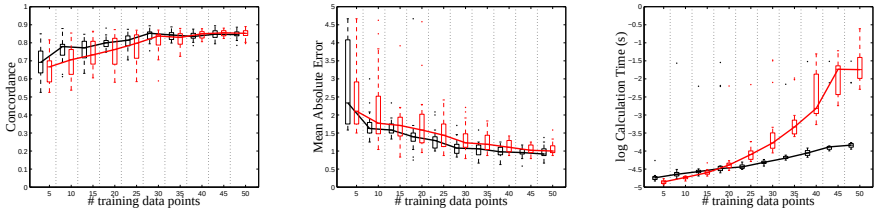


Fig. 2. Comparison between minlip (black) and the standard rankSVM (grey) on the pyrimidines dataset. The performance (concordance and mean absolute error are illustrated) of both methods is comparable, but for a reasonable number of training points, the calculation time is considerably lower for the first method.

5.2 Movie Recommendations

Our last application is a movie-recommendation task[4]. The data consists of the scores for 6040 viewers on 3952 movies. The goal is to predict the scoring of user i on movie j. We use the scorings of 1000 viewers as covariates to predict the scoring of the other viewers as follows

$$\hat{s}_{i,k} = \sum_{j=1}^{1000} w_{i,j} s_{j,k} \,,$$

where $\hat{s}_{i,k}$ indicates the predicted score of user i on movie k, $w_{i,j}$ is the weight or "importance" of user j to predict the score given by user i. $s_{j,k}$ represents the score of movie k given by user j. The 1000 viewers with the highest number of rated movies were selected as reference viewers. Another 1000 (random) viewers were used as a validation set to tune the regularization parameterand the imputation value for scores in case a reference viewer did not score a certain movie. The values for the regularization parameter were selected after 10-fold cross-validation on an exponential grid. We chose two possible values for the imputation parameter: 3, which is the mean of all possible scores, and 2, which is one score lower than the previous one, indicating that the reason for not seeing a movie could be that one is not interested in the movie. For the 4040 remaining viewers, the first half of the rated movies were used for training, the second

[4] Data available on `http://www.grouplens.org/node/73`

half for testing. The performance of the minlip method was compared with 3 other methods:

- **linear regression (LREG)**: The score of the new user is found as a linear combination of the scores of the 1000 reference users.
- **nearest neighbor classification (NN)**: This method searches the reference viewer for whom the scores are most similar to the scores of the new user. The score of the most similar reference viewer is considered as predicted score for the new viewer.
- **vector similarity (VSIM)**: This algorithm [2] is based on the notion of similarity between two datapoints. The correlation between the new user and the reference users are used as weights $w_{k,i}$ in the formula:$\hat{s}_{k,j} = \bar{s}_k + a \sum_{i=1} w_{k,i}(s_{i,j} - \bar{s}_i)$, where $\bar{s}_i$ represents the mean score for viewer i and a is a normalization constant such that $\sum_i |w_{k,i}| = 1$.

Three different performance measure were used for comparison of the methods:

- **mean zero-one error (MZOE)**
- **mean absolute error (MAE)**
- **concordance (CONC)**: measuring the concordance of the test set within the training set, defined as:
  $\mathrm{CONC}_n(u) = \frac{\sum_{i=1}^{n_t} \sum_{j=1}^{n} I[(u(X_j)-u(X_i))(T_j-T_i)>0]}{n_t n}$, with n and n_t the number of datapoints in the training and test set respectively.

Figure 3 compares all 4 methods for the 3 considered performance measures. The mean zero-one and mean absolute error should be as small as possible, while the concordance should be close as large as possible. The LREG method performs the least on all measures. The VSIM method results in a good average precision and low error measures, whereas the NN methods is better in obtaining a high concordance. The advantage of the minlip method is that is performs good on all the measures.

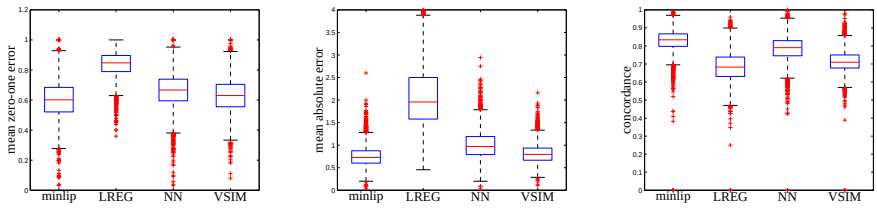


Fig. 3. Performance comparison of 4 methods: minlip (linear kernel), linear regression (LREG), nearest neighbor (NN) and vector similarity (VSIM). Three different performance measure were used. LREG performs the least on all measures. VSIM has low errors, whereas the NN method has a high concordance. The advantage of the minlip method is that it performs well on all the investigated performance measures.

6 Conclusions

This paper proposed an efficient estimator of a transformation model from noisy observations. The motivation for considering this problem is given by describing its relation to (i) the problem of learning ranking functions, and (ii) its relevance to estimating statistical models e.g. in a context of survival analysis. The latter topic will be the focus of subsequent work. We conducted two experiments to illustrate the use of this estimator: a first example on the prediction of the rankings of movies showed a good performance on different measures where other methods performed worse regarding at least one measure. In a second example on ordinal regression, we illustrate the reduction in calculation time in comparison with the standard rankSVM method, without reduction in performance.

References

1. S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.
2. J.S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
3. S.C. Cheng, L.J. Wei, and Z. Ying. Predicting Survival Probabilities with Semiparametric Transformation Models. *Journal of the American Statistical Association*, 92(437), 1997.
4. W. Chu and S.S. Keerthi. New approaches to support vector ordinal regression. *ICML*, pages 145–152, 2005.
5. S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and Scoring Using Empirical Risk Minimization. *Learning Theory: 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30: Proceedings*, pages 1–15, 2005.
6. D.M. Dabrowska and K.A. Doksum. Partial likelihood in transformation models with censored data. *Scandinavian Journal of Statistics*, 15(1):1–23, 1988.
7. R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, pages 115–132, 2000. MIT Press, Cambridge, MA.
8. J.D. Kalbfleisch and R.L. Prentice. *The Statistical Analysis of Failure Time Data*. Wiley series in probability and statistics. Wiley, 2002.
9. R. Koenker and O. Geling. Reappraising Medfly Longevity: A Quantile Regression Survival Analysis. *Journal of the American Statistical Association*, 96(454), pages 458–468, 2001.
10. K. Pelckmans, J.A.K. Suykens and B. De Moor. *A Risk Minimization Principle for a Class of Parzen Estimators*. Advances in Neural Information Processing Systems 20. MIT Press, pages 1–8, 2008.
11. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
12. J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.
13. V. Van Belle, K. Pelckmans, J.A.K. Suykens and S. Vanhuffel. Support Vector Machines for Survival Analysis. *Third International Conference on Computational Intelligence in Medicine and Healthcare, CIMED, 2007, Plymouth, UK, July 25-27: Proceedings*, pages 1–6, 2007.
14. V.N. Vapnik. *Statistical Learning Theory*. Wiley and Sons, 1998.