Bipartite Ranking through Minimization of Univariate Loss

Wojciech Kotłowski[1,3] KOTLOWSK@CWI.NL
Krzysztof Dembczyński[2,3] DEMBCZYNSKI@INFORMATIK.UNI-MARBURG.DE
Eyke Hüllermeier[2] EYKE@INFORMATIK.UNI-MARBURG.DE

[1]Centrum Wiskunde & Informatica, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[2]Mathematics and Computer Science, Marburg University, Hans-Meerwein-Str., 35032 Marburg, Germany
[3]Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland

Abstract

Minimization of the rank loss or, equivalently, maximization of the AUC in bipartite ranking calls for minimizing the number of disagreements between *pairs* of instances. Since the complexity of this problem is inherently quadratic in the number of training examples, it is tempting to ask how much is actually lost by minimizing a simple *univariate* loss function, as done by standard classification methods, as a surrogate. In this paper, we first note that minimization of 0/1 loss is not an option, as it may yield an arbitrarily high rank loss. We show, however, that better results can be achieved by means of a weighted (cost-sensitive) version of 0/1 loss. Yet, the real gain is obtained through margin-based loss functions, for which we are able to derive proper bounds, not only for rank risk but, more importantly, also for rank regret. The paper is completed with an experimental study in which we address specific questions raised by our theoretical analysis.

1. Introduction

Bipartite ranking refers to the problem of learning a ranking function from a training set of positively and negatively labeled examples. Applied to a set of unlabeled instances, a ranking function is expected to establish a total order in which positive instances precede negative ones. The most commonly used criterion for measuring the quality of a ranking function is the *area under the ROC curve*, or AUC for short. Roughly speaking, it corresponds to the fraction of correctly ordered pairs of instances. Focusing on loss minimization in this paper, we shall mostly refer to a reciprocal measure, namely the *rank loss* defined as $1 - AUC$.

Since the rank loss is defined on pairs of instances, most ranking methods use a *pairwise* approach to learning. The basic idea is to reduce ranking to binary classification by treating each pair of instances (x, x') as a single object, to be classified as positive if x should precede x' and as negative if x should ranked below x' (Herbrich et al., 1999; Freund et al., 2003; Agarwal et al., 2005; Duchi et al., 2010). Unfortunately, this approach scales quadratically with the training set size. Only in some special cases (when surrogate convex loss of a very specific form, such as hinge loss or exponential loss, is used), computational tricks allow for reducing complexity, so that the algorithms scale subquadratically with the number of training examples (Freund et al., 2003; Joachims, 2006). But even in these cases, the pairwise approach yields more difficult optimization problems, compared to binary classification, and requires special algorithms to solve the problem.

It seems legitimate, therefore, to ask whether this additional complexity is actually warranted, especially in light of several experimental studies suggesting that simple scoring classifiers, notably those minimizing margin loss functions, perform quite strongly in terms of AUC (Cortes & Mohri, 2003; Joachims, 2005; Steck, 2007). Such classifiers can be used for ranking in an obvious way, namely by sorting instances according to their classification scores. In some specific cases, it can even be proved that minimization of a margin-based loss yields the same solution as the pairwise approach (Rudin et al., 2005). On the other hand, there are also counterexamples showing that a small classification error (0/1 loss) does not necessarily imply a small rank loss, especially when the classes are unbalanced.

In this paper, we therefore seek to answer the follow-

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

ing question: What do we lose in terms of ranking performance when training a simple scoring classifier, seeking to minimize a standard (univariate) loss on the original data, instead of training a ranker on pairs of instances? The main contribution of the paper is the derivation of several upper bounds on the drop in performance, expressed in terms of risk and regret. As univariate loss functions, we analyze the standard 0/1 loss and a balanced version thereof (Section 3), as well as the exponential and logistic loss (Section 4). Finally, we also present results of an experimental study in which we address specific questions raised by our theoretical analysis (Section 5).

2. Setting and Problem Statement

Let $(x, y) \in X \times Y$ be the object-label pair generated according to the distribution $P(x, y)$, where X is a feature space and $Y = \{-1, 1\}$. In the classification problem, the purpose is to construct a *classifier* $c\colon X \to \mathbb{R}$ that accurately predicts values of y. The accuracy is measured in terms of a (univariate) *loss function* $\ell\colon Y \times \mathbb{R} \to \mathbb{R}$, where $\ell(y, z)$ is the penalty for predicting $z \in \mathbb{R}$ when the true label is $y \in Y$. The overall accuracy of the classifier c is measured in terms of its expected classification loss (*classification risk*):

$$L_\ell(c) := \mathbb{E}\left[\ell(y, c(x))\right] = \int \ell(y, c(x)) dP(x, y) \qquad (1)$$

We allow the output of the classifier to be a real number rather than a class label in order to incorporate margin loss functions into this framework.

The risk of a classifier is not always a good indicator of the performance of the learning method, since even the optimal classifier c^* (which has access to the distribution $P(x, y)$) will normally have a non-zero risk. We call c^* a *Bayes classifier*. It has to minimize expected loss conditioned on x:

$$\begin{aligned} c^*(x) &= \arg\min_z \mathbb{E}\left[\ell(y, z)|x\right] \\ &= \arg\min_z \int \ell(y, z) dP(y|x) \end{aligned} \qquad (2)$$

The risk of c^*, denoted L^*_ℓ, is called *Bayes risk*. It offers a reasonable baseline for comparison and suggests to define the *regret* of a classifier c as follows:

$$\mathrm{Reg}_\ell(c) = L_\ell(c) - L^*_\ell \qquad (3)$$

2.1. Ranking

In *bipartite ranking*, the goal is to learn a *ranking function* (ranker) $r\colon X \times X \to \{-1, 0, 1\}$, where $r(x, x') = 1$ means that x is ranked higher than x', while $r(x, x') = 0$ signifies a tie; for consistency, we assume r to satisfy $r(x, x') = -r(x', x)$. The accuracy of a ranking function is measured by the *rank loss*, namely the probability of incorrectly ordering two examples $(x, 1)$ and $(x', -1)$, one positive and one negative, drawn independently from $P(x, y)$ (ties are broken randomly). More precisely,

$$\begin{aligned} L_{\mathrm{rank}}(r) &:= P(r(x, x') = -1 | y > y') \\ &\quad + \frac{1}{2} P(r(x, x') = 0 | y > y') = \\ &= \int \frac{|1 - r(x, x')|}{2} dP(x|y{=}1) dP(x'|y'{=}{-}1). \end{aligned} \qquad (4)$$

Like in the classification case, we introduce the *Bayes ranker* r^*. Given $P(x, y)$, it can be derived explicitly (Clémençon et al., 2008; Balcan et al., 2008; Ailon & Mohri, 2008):

$$r^*(x, x') = \mathrm{sgn}\left(\eta(x) - \eta(x')\right), \qquad (5)$$

where $\eta(x) = P(y = 1 \,|\, x)$, and:

$$\mathrm{sgn}\,(z) = \begin{cases} 1 & z > 0 \\ 0 & z = 0 \\ -1 & z < 0 \end{cases}$$

is a sign function. The *rank regret* of a ranker r is then defined as follows:

$$\mathrm{Reg}_{\mathrm{rank}}(r) = L_{\mathrm{rank}}(r) - L^*_{\mathrm{rank}}, \qquad (6)$$

where $L^*_{\mathrm{rank}} := L_{\mathrm{rank}}(r^*)$.

2.2. Problem Statement

Each classifier c can be turned into a ranker r_c as follows:

$$r_c(x, x') = \mathrm{sgn}\left(c(x) - c(x')\right).$$

By a slight abuse of notation, we will speak about the rank risk of c and denote it by $L_{\mathrm{rank}}(c)$, having in mind the risk of the associated ranker r_c, i.e., $L_{\mathrm{rank}}(c) := L_{\mathrm{rank}}(r_c)$. As an important observation, note that the Bayes ranker r^* can be constructed in this way. Indeed, consider a classifier $\tilde{c}(x)$ of the form $\tilde{c}(x) = f(\eta(x))$, with a strictly monotonically increasing function $f(\cdot)$. Then, (5) implies that $r^* = \mathrm{sgn}\left(\tilde{c}(x) - \tilde{c}(x')\right)$.

The purpose of the paper is to address the following problem: Given a classifier c with classification regret $\mathrm{Reg}_\ell(c)$ for some loss function ℓ, what is the maximum rank regret of c, $\mathrm{Reg}_{\mathrm{rank}}(c)$? In other words, how can we bound $\mathrm{Reg}_{\mathrm{rank}}(c)$ in terms of $\mathrm{Reg}_\ell(c)$? We will also consider a weaker objective, namely bounding the rank risk $L_{\mathrm{rank}}(c)$ in terms of the classification risk $L_\ell(c)$.

Remark. Although we focus on the "generalization error" of the classifier (the expected loss according to P), our analysis equally applies to the training error and the optimization process on the training data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. To this end, P is simply replaced by the empirical distribution (i.e., the relative frequency in D). The risk then simply reduces to the average loss on D.

3. The Case of 0/1 Loss

The simple 0/1 loss function (also called "classification error")

$$\ell_{0/1}(y, z) = \mathbb{1}[zy \leq 0]$$

(where $\mathbb{1}[C]$ is the indicator function, equal to 1 if predicate C holds and 0 otherwise) is by far the most commonly used performance measure for classification. Yet, a simple observation immediately excludes any interesting bound on the regret, i.e., a bound of the form

$$\mathrm{Reg}_{\mathrm{rank}}(c) \leq f\left(\mathrm{Reg}_{0/1}(c)\right), \tag{7}$$

where $\mathrm{Reg}_{0/1}$ denotes the classification regret for 0/1 loss and $f(\cdot)$ is a function that becomes small as $x \to 0$. It follows from (2) that the function $c^*(x) = \mathrm{sgn}\,(\eta(x) - 1/2)$ is a Bayes classifier for 0/1 loss. Plugging this function into (7), the right-hand side is 0, while the left hand-side can be much larger than 0, since $r_{c^*}(x, x') = \mathrm{sgn}\,(c^*(x) - c^*(x'))$ will not be a Bayes ranker for most of the distributions.

In particular, 0/1 loss is not *consistent*: even if c converges to c^* in terms of the 0/1 loss, r_c does *not* converge to r^* in terms of the rank loss. In fact, this example also shows that consistency cannot be assured unless $r_{c^*} = r^*$, i.e., the Bayes ranker r^* can be derived from the Bayes classifier c^*. Since r^* is any monotonic transformation of the conditional probability η, this means that, in one way or the other, c^* must estimate conditional probabilities.

3.1. Risk Bound

Given that a useful bound cannot be obtained in terms of the regret, let us now turn to the risk itself. Denote by $L_{0/1}$ the classification risk for 0/1 loss.

Lemma 3.1. *For a classifier c, let α and β denote the probability of a false negative and a false positive prediction, respectively. Moreover, let p denote the probability of the positive class. Then,*

$$L_{\mathrm{rank}}(c) \leq \frac{\alpha}{p} + \frac{\beta}{1-p} - \frac{\alpha\beta}{p(1-p)}. \tag{8}$$

Proof. Let $\alpha = P(c(x) \leq 0, y = 1)$ and $\beta = P(c(x) \geq 0, y = -1)$. Consider the event E of selecting two examples (x, y) and (x', y') such that $y = 1$ and $y' = -1$; the probability of E is $p(1-p)$. Moreover, note that a ranking error can only occur if (x, y) is a false negative or if (x', y') is a false positive, since correct classification of both instances implies correct ranking. The probabilities of these two events are α/p and $\beta/(1-p)$, respectively, and since they are independent (conditioned on E), the probability of their union is given by the right-hand side of (8). □

Theorem 3.1. *Let the distribution $P(x, y)$ be such that the prior of the positive class is equal to p. Then, for every classifier c,*

$$L_{\mathrm{rank}}(c) \leq \frac{L_{0/1}(c)}{\min\{p, 1-p\}}. \tag{9}$$

Proof. According to the previous lemma,

$$\begin{aligned} L_{\mathrm{rank}}(c) &\leq \frac{\alpha}{p} + \frac{\beta}{1-p} - \frac{\alpha\beta}{p(1-p)} \\ &\leq \frac{\alpha}{p} + \frac{\beta}{1-p} \\ &\leq \frac{\alpha + \beta}{\min\{p, 1-p\}} \\ &= \frac{L_{0/1}(c)}{\min\{p, 1-p\}}, \end{aligned}$$

which proves the theorem. □

According to both bounds, namely (8) and the less tight version (9), $L_{0/1}(c) \to 0$ does indeed imply $L_{\mathrm{rank}}(c) \to 0$. Both bounds critically depend on the class distribution, however, and may become loose in the case of strongly imbalanced classes. As an illustration, consider the majority classifier, for which $L_{0/1}(c) = \min\{p, 1-p\}$ while $L_{\mathrm{rank}}(c) = 1/2$. Once again, this shows that strong classification (at least in the sense of 0/1 loss, indeed a debatable measure in the case of strongly imbalanced class distributions) does not necessarily imply strong ranking performance.

3.2. Balanced Loss

From the previous discussion, one may conclude that imbalanced class distributions are undesirable from a ranking point of view. Indeed, the bound (8) suggests that false negative and false positive predictions have a different influence on the ranking performance. In fact, ignoring the last term in (8), which involves both α and β, the influence of α on the bound (measured in terms of the derivative) is $1/p$, while the influence of β

is $1/(1-p)$. This suggests that minimizing a *balanced* version of the 0/1 loss, namely a loss of the form

$$\ell_b(y,z) = w(y)\mathbb{1}[zy \le 0],$$

may help improve ranking performance. More specifically, taking $w(1) = (2p)^{-1}$ and $w(-1) = (2(1-p))^{-1}$ guarantees $pw(1) + (1-p)w(-1) = 1$ and thus yields a re-scaling of the loss within $[0,1]$. According to the balanced loss thus defined, a wrong classification of an instance from the minority class is punished stronger than a wrong classification of an instance from the majority class.

Alternatively, the values $w(1)P(y=1) = 1/2 = w(-1)P(y=-1)$ can be considered as new priors, perfectly balancing the class distribution. Indeed, in terms of the balanced 0/1 loss, a bound on the ranking error can be expressed as a special case of (9) with $p = 1/2$:

$$L_{\text{rank}}(c) \le 2L_b(c), \tag{10}$$

where $L_b(c)$ is the expected balanced loss produced by the classifier c. This follows immediately from Lemma 3.1 by omitting the last term in (8).

Note that many learning methods are able to handle weighted training examples as input, and hence can be used for minimizing the balanced loss without a need for major modifications. Practically, a classifier does of course not have access to the class priors in the training phase. However, given their estimates $\hat{P}(y=1)$ and $\hat{P}(y=-1)$, one can use a plug-in estimate $\hat{w}(y) = (2\hat{P}(y))^{-1}$ instead of the true $w(y)$. The error thus produced in the risk will be proportional to $P(y)|w(y) - \hat{w}(y)| = |w(y) - \hat{w}(y)|/w(y)$. This quantity exceeds ϵ only if $|\hat{P}(y) - P(y)| \ge \frac{\epsilon}{1-\epsilon}P(y)$. If $\hat{P}$ is a frequency-based estimator, the Chernoff bound implies that the error decrease exponentially with the sample size:

$$P\left(\frac{|w(y) - \hat{w}(y)|}{w(y)} \ge \epsilon\right) \le 2\exp\left\{-\frac{2P(y)n\epsilon}{1-\epsilon}\right\}$$

4. Margin-based Loss Functions

Let us now proceed to the analysis of general loss functions of the form $\ell(y,z) = \ell(yz)$, called *margin losses* in the literature. For now, we assume that ℓ is nonnegative, strictly monotonically decreasing (better classification incurs less loss) and normalized by $\ell(0) = 1$. Since this means that ℓ upper-bounds the 0/1 loss, Theorem 3.1 implies

$$L_{\text{rank}}(c) \le 2L_\ell(c) \tag{11}$$

in the case of equal priors $P(y=1) = P(y=-1) = 1/2$. For unequal priors, we can use the balanced version of margin loss, similarly as in the case of balanced 0/1 loss in Section 3.2, and obtain the same bound (11).

4.1. Regret Bound

The risk bound, based solely on the fact that the margin loss upper-bounds 0/1 loss, does not suggest any advantage of margin losses over 0/1 loss. On the other hand, intuition suggests that a margin loss may indeed help us decrease the value of rank loss: Since margin loss is strictly monotonically decreasing, it is not only sensitive to the sign of z, but also to the value of z itself. In order to minimize expected loss, a classifier should therefore assign scores $c(x)$ in agreement with the class probabilities $P(y=1 \mid x)$.

In the rest of this section, we will focus on two commonly used margin losses, namely *exponential loss* $\ell_{\exp}(z) = e^{-z}$ and *logistic loss* $\ell_{\log}(z) = \log(1+e^{-z})$. The former is known for its use in the first and most popular boosting algorithm, AdaBoost (Freund & Schapire, 1997), whereas the latter derives from maximum likelihood estimation. Interestingly, both losses share (almost) the same Bayes classifier (Hastie et al., 2003):

$$c^*_{\exp}(x) = \frac{1}{2}\log\frac{\eta(x)}{1-\eta(x)}, \quad c^*_{\log}(x) = \log\frac{\eta(x)}{1-\eta(x)}.$$

Since both classifiers are monotonic functions of the conditional probability $\eta(x)$, the derivation of a bound for the rank regret should be possible. Indeed, in the remainder of this section, we shall prove such bounds, establishing the consistency and the convergence rate for the exponential and the logistic loss. In other words, we show that a minimization of these losses is a consistent procedure for minimizing the rank risk: If the classification regret for the respective loss converges to 0, then so does the rank regret, and the convergence rate is at most quadratically slower.

Theorem 4.1. *Let $\ell(y,z)$ be the balanced margin loss function, i.e., $\ell(y,z) = w(y)\ell(yz)$ with $w(y) = 1/2P(y)$. The following regret bounds hold for the exponential loss $\ell(yz) = e^{-yz}$ and the logistic loss $\ell(yz) = \log(1+e^{-yz})$, respectively:*

$$\text{Reg}_{\text{rank}}(c) \le \frac{3\sqrt{2}}{2}\sqrt{\text{Reg}_{\exp}(c)}, \tag{12}$$

$$\text{Reg}_{\text{rank}}(c) \le 2\sqrt{\text{Reg}_{\log}(c)}, \tag{13}$$

where $\text{Reg}_{\exp}$ and $\text{Reg}_{\log}$ are the classification regrets for balanced exponential and logistic loss, respectively.

Proof. (sketch) Due to space limitations, we only provide a sketch of the proof. We will use a theorem

from (Bartlett et al., 2006) which, for every classifier c, allows for bounding the 0/1 classification regret $\mathrm{Reg}_{0/1}(c)$ by the margin loss ℓ regret $\mathrm{Reg}_\ell(c)$:

$$\mathrm{Reg}_{0/1}(c) \leq f(\mathrm{Reg}_\ell(c)) \tag{14}$$

If ℓ is the exponential loss, we have $f(x) = \sqrt{x(2-x)} \leq \sqrt{2x}$ assuming $\mathrm{Reg}_\ell(c) \leq 1$. For logistic loss, one also can show that $f(x) \leq \sqrt{2x}$.

We first prove the theorem for equal priors, i.e., $P(y = 1) = P(y = -1) = 1/2$. In this case, we can skip the balancing and prove the theorem for ordinary exponential and logistic losses. We start by reducing the ranking problem to a classification problem in a typical way (Cohen et al., 1999; Balcan et al., 2008; Ailon & Mohri, 2008; Herbrich et al., 1999). This is done by constructing objects $(\tilde{x}, \tilde{y})$ in the new problem from i.i.d. pairs of objects (x, y) and (x', y') in the original problem, namely by setting $\tilde{x} = (x, x')$ and $\tilde{y} = \frac{y-y'}{2}$. Then, objects with $\tilde{y} = 0$ are removed. It can be shown that the new distribution is given by $\tilde{P}(\tilde{x}, \tilde{y}) = \frac{P(x|y=\tilde{y})P(x'|y'=-\tilde{y})}{2}$. Moreover, if we define the classifier in the new problem $\tilde{c}$: $X \times X \to \mathbb{R}$ as $\tilde{c}(x, x') = c(x) - c(x')$, then $L_{0/1}(\tilde{c}) = L_{\mathrm{rank}}(c)$, i.e., the 0/1 loss of $\tilde{c}$ in the new problem and the rank loss of c in the original problem P coincide. Also, the Bayes classifier $\tilde{c}^*$ in the new problem is the Bayes ranker r^* in the original problem. Thus, we obtain $\mathrm{Reg}_{0/1}(\tilde{c}) = \mathrm{Reg}_{\mathrm{rank}}(c)$.

The longest and most involved part of the proof consists of relating $\mathrm{Reg}_\ell(\tilde{c})$ with $\mathrm{Reg}_\ell(c)$, where ℓ is either the exponential loss or the logistic loss. For exponential loss, one can show that

$$L_{\exp}(\tilde{c}) \leq L^2_{\exp}(c), \qquad L_{\exp}(\tilde{c}^*) = L^2_{\exp}(c^*),$$

which implies

$$\begin{aligned}\mathrm{Reg}_{\exp}(\tilde{c}) &\leq L^2_{\exp}(c) - L^2_{\exp}(c^*) \\ &= (\mathrm{Reg}_{\exp}(c) + 2L_{\exp}(c^*))\mathrm{Reg}_{\exp}(c) \\ &\leq \frac{9}{4}\mathrm{Reg}_{\exp}(c).\end{aligned}$$

The last inequality holds because, for every distribution, $L_{\exp}(c^*) \leq 1$, and because we can assume $\mathrm{Reg}_{\exp}(c) \leq \frac{1}{4}$ (if $\mathrm{Reg}_{\exp}(c) > \frac{1}{4}$, then (12) is trivially satisfied because $\mathrm{Reg}_{\mathrm{rank}}(c) \leq 1$ for all c and for all distributions). We can now use the bound (14) to derive

$$\begin{aligned}\mathrm{Reg}_{\mathrm{rank}}(c) = \mathrm{Reg}_{0/1}(\tilde{c}) &\leq f(\mathrm{Reg}_{\exp}(\tilde{c})) \\ &\leq \sqrt{2\mathrm{Reg}_{\exp}(\tilde{c})} \leq \frac{3\sqrt{2}}{2}\sqrt{\mathrm{Reg}_{\exp}(c)}.\end{aligned}$$

For logistic loss, one cannot separately bound $L_{\log}(\tilde{c})$ by $L_{\log}(c)$ and $L_{\log}(c^*)$ by $L_{\log}(\tilde{c}^*)$. However, with the help of a more involved analysis, one can show that

$$\mathrm{Reg}_{\log}(\tilde{c}) \leq 2\mathrm{Reg}_{\log}(c),$$

which implies

$$\begin{aligned}\mathrm{Reg}_{\mathrm{rank}}(c) = \mathrm{Reg}_{0/1}(\tilde{c}) &\leq f(\mathrm{Reg}_{\log}(\tilde{c})) \\ &\leq \sqrt{2\mathrm{Reg}_{\log}(\tilde{c})} \leq 2\sqrt{\mathrm{Reg}_{\log}(c)}.\end{aligned}$$

Thus, we proved theorem for equal priors. To prove it for arbitrary priors, let us introduce the distribution $P'(x, y)$, such that $P'(y) := 1/2$ and $P'(x|y) := P(x|y)$. Note, that when we change P to P', the distribution on pairs $P(\tilde{x}, \tilde{y})$ will not change, as it only depends on $P(x|y)$. Therefore, the rank regret and the margin loss regret for $\tilde{c}$ will not change as well.

On the other hand, the expected balanced loss ℓ_b according to P is equal to the expected ordinary loss ℓ according to P':

$$\begin{aligned}L_b(c) &= \int w(y)\ell(yc(x))dP(x|y)dP(y) \\ &= \int \ell(yc(x))dP'(x|y)dP'(y) = L'_b(c),\end{aligned}$$

so we can apply what we proved so far for P' and get the bound in terms of the balanced loss for original distribution P. □

4.2. Balancing

In Section 3.2, we have seen that, in the case of 0/1 loss, balancing the positive and negative class is likely to have a positive influence on ranking performance. An obvious question, therefore, is whether a similar effect can be expected in the case of a margin loss.

Interestingly, the answer appears to be negative, at least on a population level. Since the expected loss produced by a prediction $c = c(x)$ is given by

$$\eta(x) \cdot \ell(c) + (1 - \eta(x)) \cdot \ell(-c),$$

where $\eta(x) = P(y = 1 \,|\, x)$, the risk minimizing prediction c^* is implicitly determined by the equation

$$R = \frac{\eta(x)}{1 - \eta(x)} = -\frac{\partial\ell/\partial c(-c)}{\partial\ell/\partial c(c)},$$

where $\partial\ell/\partial c$ denotes the derivative of ℓ with respect to c. Now, in the case of balancing, the only change concerns the probability ratio R on the left-hand side, which is replaced by αR with $\alpha = w(1)/w(-1)$. Consequently, if a risk minimizer can be written in the

form $c = c(x) = \log(R)$, as it is the case for the exponential and the logistic loss, then the risk minimizer for the balanced loss is of the form

$$c_b(x) = \log(\alpha R) = \log(R) + \log(a) = c(x) + c_0.$$

In other words, the two classifiers only differ by a constant c_0, which is of no practical relevance as long as the underlying model class is closed under addition of a constant term (a property exhibited by essentially all learning methods minimizing a margin loss).

For the exponential loss, it can furthermore be shown that weighing does not have any practical effect even on the level of empirical risk minimization, as long as the family of classifier $\mathcal{C}$ over which we optimize, has the following closure property: if $c \in \mathcal{C}$, then also $c + c_0 \in \mathcal{C}$, where c_0 is a constant. Indeed, fix $c \in \mathcal{C}$ and let $L_w(c_0) = \sum_{i=1}^{n} w(y_i) e^{-y_i(c(x_i)+c_0)}$ be the empirical risk of $c + c_0$. We minimize over c_0 by setting the derivative to zero:

$$\frac{\partial L_w(c_0)}{\partial c_0} = -\sum_{i=1}^{n} w(y_i) y_i e^{-y_i(c(x_i)+c_0)} = 0$$

$$\iff e^{2c_0} = \frac{w(1)\sum_{y_i=1} e^{-c(x_i)}}{w(-1)\sum_{y_i=0} e^{c(x_i)}}$$

By plugging the optimal value for c_0 into the expression for $L_w(c_0)$, we eliminate c_0 from the optimization problem, and instead minimize the "profiled" loss

$$L_w = 2\sqrt{w(-1)w(1)} \left(\sum_{y_i=1} e^{-c(x_i)} \right) \left(\sum_{y_i=0} e^{c(x_i)} \right).$$

Obviously, the optimal solution minimizing this objective does not depend on the weights, which only appear as a constant multiplier. In summary, this means that the theoretical guarantees given for the balanced version of the loss remain valid, even when minimizing the unbalanced exponential loss. We conjecture that this phenomenon approximately holds for logistic loss, too (i.e., shifting the classifier effectively changes the class weights).

5. Computational Experiments

In this section, we verify our theoretical results by means of computational experiment on both artificial and real datasets. More specifically, our goal is to verify the main claim of the paper, namely that minimization of pointwise losses is sufficient to achieve low rank loss. To this end, we compare the performance of a linear classifier trained by minimizing the following three objectives: (1) exponential loss, (2) logistic loss, and (3) pairwise hinge loss. The reason we chose (1) and (2) is clear, while the choice of (3) is due to the fact that pairwise hinge loss is the tightest convex upper bound on the rank loss, and it is the objective used by one of the most popular algorithms for rank loss minimization, SVM^{perf} for ordinal regression (SVM-OR) (Joachims, 2006).

We did not compare to the other popular ranking algorithm, RankBoost (Freund et al., 2003), as for the bipartite ranking it becomes equivalent to exponential loss minimization while trained until convergence (Rudin et al., 2005). We trained all the methods using L_2 regularization. We ran the experiments with regularization constant $\lambda \in \{0.01, 0.1, 1, 10, 100, 1000\}$[1] and chose the best result for each method.

5.1. Artificial Data

We consider two different models for generating the data, a linear and a nonlinear one. In both cases, a 50-feature input vector $x = (x_1, \ldots, x_{50}) \in [0,1]^{50}$ is first drawn from a uniform distribution on a cube. From a prediction point of view, the most important characteristic of the data is the underlying "target" function, generating an output from the inputs. In our case, we assume the output y is generated by thresholding a function $f(x)$, i.e., $y = 2\mathbb{1}[f(x) \geq 0] - 1$. For the linear model, f is a linear function of the inputs, namely $f(x) = a_0 + \sum_{i=1}^{50} a_i x_i \geq 0$. For the nonlinear model, $f(x) = a_0 + \sum_{i=1}^{50} a_i r_i(x)$ is a linear combination of "decision rules" $r_i(x)$, defined as $r_i(x) = \prod_{j=1}^{m_i} \theta(x_{k_j} - b_j)$, where $\theta(z) = \mathbb{1}[z \geq 0]$ or $\theta(z) = \mathbb{1}[z \leq 0]$, $k_j \in \{1, \ldots, 50\}$ and $b_j \in [0,1]$. In other words, decision rules are conjunctions of single-feature threshold functions, and are axis-parallel hyper-rectangles in the feature space. All the coefficients a_i, $i > 0$, are drawn from a Gaussian $N(0,1)$, m_i are drawn from a geometric distribution with rate $1/2$, and the features and thresholds in the rules are drawn uniformly at random. Then, the noise is introduced by randomly relabeling the objects with a fixed probability κ, which is tuned to obtain the risk of the Bayes ranker equal to 0.1. The threshold a_0 is chosen to obtain the desired class priors $(p, 1-p)$. Since we investigate the case of equal priors ($p = 0.5$) and imbalanced data ($p = 0.9$), we end up with four combinations of the models (linear/nonlinear and balanced/imbalanced).

For each combination, 30 random models (i.e. target functions f) are generated in the way described above to decrease variation w.r.t. random choice of model parameters. For each model, 30 training sets of size 1000

[1] For pairwise optimization, we rescaled the values properly to account for a larger range of empirical risk.

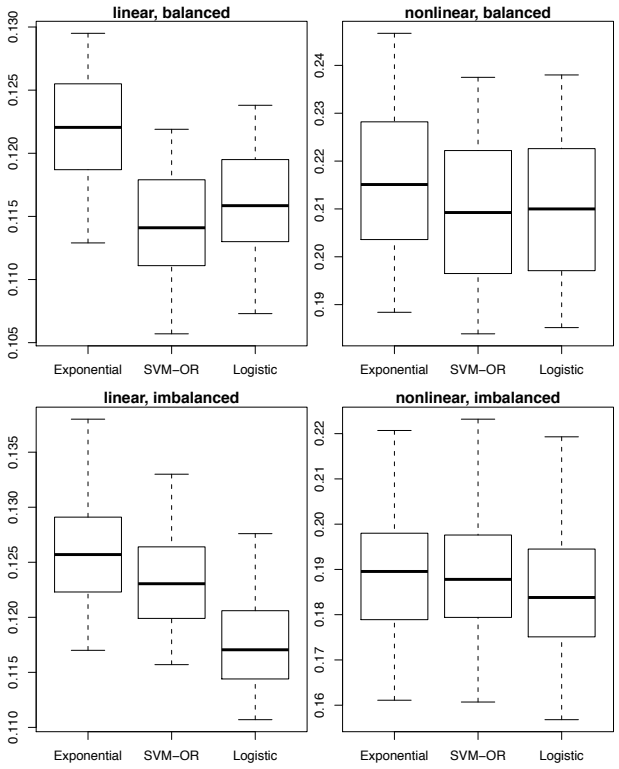

Figure 1. Results of the experiment for different linear/nonlinear and balanced/imbalanced combination of the data models.

and a test set of size 10000 are generated. Thus, each algorithm is trained on $4 \times 30 \times 30 = 3600$ datasets. The performance of each algorithm (rank loss) for each random model is averaged over the 30 training sets. The results for different models are shown as boxplots in Fig. 5.1. Thus, each box is made out of 30 observations, for 30 random models.

As we can see from the plots, there is no clear winner in the experiment. SVM slightly outperforms the logistic loss classifier for balanced data, while the logistic loss classifier performs better for imbalanced data. The exponential loss classifier always exhibits the worst performance, however, the differences among the algorithms are very small, at most around 0.01. Note that highly imbalanced class priors do not harm the results of the logistic loss classifier, despite the fact that no loss balancing is introduced. Since we also noticed worse performance of the exponential loss classifier in terms of 0/1 loss, compared to the logistic loss classifier, we believe that slightly worse results for exponential loss are due to its lack of robustness against the noise (Hastie et al., 2003). Summarizing, there seems to be no benefit from optimizing the pairwise loss in terms of ranking performance.

Table 1. Results of the experiment on real datasets. For first nine datasets, 10-fold cross-validation repeated 10 times is used. For *covtype* and *kdd04*, the error estimate is computed over a single split into train and test data.

Dataset	Exponential	SVM-OR	Logistic
Breast-w	0.0051	0.0049	0.0054
Breast-c	0.3077	0.2955	0.3005
Colic	0.1251	0.1352	0.1179
Diabetes	0.1724	0.1702	0.1804
Haberman	0.3684	0.3153	0.3820
Heart-h	0.0887	0.1005	0.0929
Hepatitis	0.1289	0.1321	0.1230
Ionosphere	0.0811	0.0773	0.0884
Vote	0.0098	0.0103	0.0096
Covtype	0.1635	0.1604	0.1623
KDD04	0.2114	0.2083	0.2143

5.2. Real Data

The second part of the experiment uses benchmark datasets taken from the UCI repository (Asuncion & Newman, 2007). We consider nine small-size and two relatively large datasets. For small-size datasets, we perform ten times 10-fold cross-validation. For the large datasets, we use a single split into training and testing examples. The first large problem is the prediction of class 1 in the *covertype* dataset that contains 54 features. There are 582 012 examples in total, which we split, similarly as in (Joachims, 2006), into 522 911 training examples and 58 101 test examples. The second large problem is the KDD04 Physics task from the KDD-Cup 2004 (Caruana et al., 2004) with 78 features. We used the original training set that contains 50 000 examples and split it by using 60% of examples for training and the rest for testing.

In Table 1, we present the average rank loss for three algorithms used in the experiment. We can observe a similar performance of the algorithms. The difference in performance on particular datasets is usually small, with an exception of the *haberman* dataset, for which we see the superiority of SVM-OR. In general, however, we cannot observe a consistent advantage of any of the algorithms.

Again, the results indicate that univariate loss minimization is competitive to algorithms that aim at directly minimizing the pairwise rank loss.

6. Conclusions

In this paper, we studied the problem of minimizing the rank loss in bipartitie ranking and argued that this problem can be solved effectively by minimizing

a simple (univariate) loss function. Roughly speaking, this means that bipartite ranking can effectively be reduced to standard classification with conditional probability estimation. Since classification algorithms are computationally much more efficient than ranking algorithms, a result of this kind is clearly of practical relevance.

Our theoretical results, establishing loss and regret bounds on the ranking performance of a classifier, confirm this conjecture, at least for margin-based loss functions such as exponential and logistic loss. Further evidence has been provided by experimental studies with real and synthetic data, which are in complete agreement with our theoretical results.

In a sense, our results are not very surprising, especially since the strong ranking performance achieved through margin loss (as opposed to 0/1 loss) minimization has already been observed in several empirical studies. Besides, since probability estimation establishes a close connection between rank loss minimization (instances need to be ranked by probability) and minimization of margin loss, the results are also intuitively plausible. Still, we consider them as an important contribution, as they provide a sound theoretical explanation of previous observations and arguably help to gain further insights into the ranking problem.

Of course, the paper also leaves a number of open questions. One of them concerns the existence of regret bound for all margin-based losses estimating conditional probabilities. A corresponding extension of the study presented in this paper is planned as future work.

References

Agarwal, S., Graepel, T., Herbrich, R., Har-Pelend, S., and Roth, D. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

Ailon, N. and Mohri, M. An efficient reduction of ranking to classification. In *COLT*, pp. 87–98, 2008.

Asuncion, A. and Newman, D.J. UCI Machine Learning Repository, 2007. URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

Balcan, M.-F., Bansal, N., Beygelzimer, A., Coppersmith, D., Langford, J., and Sorkin, G. B. Robust reductions from ranking to classification. *Machine Learning*, 82:139–153, 2008.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473): 138–156, 2006.

Caruana, R., Joachims, T., and Backstrom, L. KDD-Cup 2004: Results and analysis. *SIGKDD Explor. Newsl.*, 6:95–108, 2004.

Clémençon, S., Gábor, L., and Vayatis, N. Ranking and empirical minimization of U-statistics. *Annals of Statistics*, 36:844–874, 2008.

Cohen, W. W., Schapire, R. E., and Singer, Y. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

Cortes, C. and Mohri, M. AUC optimization vs. error rate minimization. In *NIPS*. MIT Press, 2003.

Duchi, J., Mackey, L., and Jordan, M. On the consistency of ranking algorithms. In *ICML 2010*, 2010.

Freund, Y. and Schapire, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4: 933–969, 2003.

Hastie, T., Tibshirani, R., and Friedman, J. H. *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2003.

Herbrich, R., Graepel, T., and Obermayer, K. Regression models for ordinal data: A machine learning approach. Technical report TR-99/03, Technical University of Berlin, 1999.

Joachims, T. A support vector method for multivariate performance measures. In *ICML 2005*, pp. 377–384, 2005.

Joachims, T. Training linear SVMs in linear time. In *ACM SIGKDD*, pp. 217–226, 2006.

Rudin, C., Cortes, C., Mohri, M., and Schapire, R. E. Margin-based ranking meets boosting in the middle. In *COLT*, pp. 63–78, 2005.

Steck, H. Hinge rank loss and the area under the ROC curve. In *ECML*, Lecture Notes in Computer Science, pp. 347–358. Springer-Verlag, 2007.