Hinge Rank Loss and the Area Under the ROC Curve

Harald Steck

Siemens Medical Solutions, IKM CAD & Knowledge Solutions,
51 Valley Stream Parkway E51, Malvern, PA 19355, USA
harald.steck@siemens.com

Abstract. In ranking as well as in classification problems, the Area under the ROC Curve (AUC), or the equivalent Wilcoxon-Mann-Whitney statistic, has recently attracted a lot of attention. We show that the AUC can be lower bounded based on the hinge-rank-loss, which simply is the rank-version of the standard (parametric) hinge loss. This bound is asymptotically tight. Our experiments indicate that optimizing the (standard) hinge loss typically is an accurate approximation to optimizing the hinge rank loss, especially when using affine transformations of the data, like e.g. in ellipsoidal machines. This explains for the first time why standard training of support vector machines approximately maximizes the AUC, which has indeed been observed in many experiments in the literature.

1 Introduction

The equivalence of the Area under the Receiver Operating Characteristics Curve (AUC) and the Wilcoxon-Mann-Whitney statistic has in recent years sparked a lot of interesting work toward a better understanding of classification and ranking problems. While the AUC is a valuable measure for *assessing* the quality of a given classifier or ranking method, it was typically not used as an objective function when *training / optimizing* a classifier, mainly due to its high computational cost or different preferences concerning performance measures in the past (e.g., 0/1-loss). Only recently, computationally tractable optimization methods were developed for the use of AUC during training. These approaches are reviewed in Section 8.

This paper aims to better understand the interrelationship among the performance measures AUC (cf. Section 3), 0/1-loss, and our new hinge rank loss (cf. Section 4). It is not concerned with algorithms for optimizing these measures. In Section 5, we first show that the AUC is determined by the difference between the hinge rank loss and the 0/1-loss; and secondly, that the hinge rank loss provides an asymptotically-tight lower-bound on the AUC. Thirdly, Section 6 argues that the AUC is approximately maximized by the standard training of support vector machines; this can be improved by using affine transformations of the data, e.g., employing (kernel) PCA [1] or ellipsoidal machines [2]. This is supported by our experiments in Section 7 as well as by many experimental findings in the literature [3,4,5,6], as discussed in Section 8.

J.N. Kok et al. (Eds.): ECML 2007, LNAI 4701, pp. 347–358, 2007.

2 Notation

This section introduces relevant notation concerning classifiers and their parametric (real-valued) vs. rank outputs. Like in much of the machine learning literature, we consider *binary* classification in this paper. Assume we are given data $D = \{(x_i, y_i)\}_{i=1,...,N}$ with N examples, class labels $y_i \in \{-1, +1\}$, and input vectors x_i; the number of positive examples (i.e., where $y_i = +1$) is N^+, and the number of negative examples is $N^- = N - N^+$.

Given a classifier C with real-valued output c_i, we have $c_i = C(x_i)$ for each input x_i, $i = 1, ..., N$. For simplicity, we assume that there are no ties, i.e., $c_i \neq c_j$ for all $i \neq j$.[1] Given the real-valued threshold θ, the classification rule is $\text{sign}(c_i - \theta)$. The rank-version of this classifier is denoted as follows: let the values c_i be ordered in *ascending* order, i.e., the smallest output-value gets assigned the lowest rank. Let $r_i \in \{1, ..., N\}$ be the rank of example $i = 1, ..., N$. Moreover, let r_j^+ denote the ranks of the positive examples, $j = 1, ..., N^+$; and r_k^- be the ranks of the negative ones, $k = 1, ..., N^-$. As the counterpart of the real-valued threshold θ, a natural definition of the rank-threshold is $\tilde{\theta} = \max\{r_i : c_i \leq \theta\} + 1/2 = \min\{r_i : c_i > \theta\} - 1/2$, as it is located half way between the two neighboring ranks at the (real-valued) threshold.[2] The classification rule is $\text{sign}(r_i - \tilde{\theta})$, so that the real-valued version and the rank-version of the classifier yield identical classification results.

3 Area Under the Curve

This section briefly reviews the Area under the Receiver Operating Characteristics (ROC) Curve (AUC). While the AUC, denoted by A, had been used as a measure for assessing *classifier performance* in machine learning (e.g., see [7] for an overview), in recent years it has also become popular as a quality measure in *ranking problems*. This is because of its well-known equivalence to the Wilcoxon-Mann-Whitney (WMW) statistic [9,10]; it can be written in terms of pairwise comparisons of ranks:

$$A = \frac{1}{N^+ N^-} \sum_{j=1}^{N^+} \sum_{k=1}^{N^-} \mathbf{1}_{r_j^+ > r_k^-}, \tag{1}$$

where $\mathbf{1}$ is the indicator function: $\mathbf{1}_a = 1$ if a is true and 0 otherwise. Essentially, it counts the number of pairs of examples that are ranked correctly, i.e., positive examples are supposed to have a higher rank than negative ones. The AUC takes values $A \in [0, 1]$; $A = 1$ indicates perfect classification/ranking, and a random classification/ranking results in $A = 0.5$. The AUC is independent of the threshold value θ used in classification.

[1] Given continuous inputs, and assuming that the classifier does not discretize or use a step-function internally, there are no ties in the outputs to be expected in general.

[2] Again, no ties concerning the c_i's are assumed.

Even though the computational cost of evaluating Eq. 1 for all pairs of positive and negative examples (of which there are N^+N^-) seems to grow quadratically with N at first glance, it is easy to see that the WMW statistic can be evaluated in linear time in N given the ranks (if the continuous outputs c_i are given, the ranks can be determined by sorting them in time $N \log N$) [7,8]:

$$A = \frac{1}{N^+N^-} \sum_{j=1}^{N^+} (r_j^+ - j) = \frac{1}{N^+N^-} \left[\left(\sum_{j=1}^{N^+} r_j^+ \right) - \binom{N^+ + 1}{2} \right]. \qquad (2)$$

4 Hinge Rank Loss

In this section, we define the *hinge rank loss* as a rank-version of the standard (parametric) hinge loss, which is commonly used for learning support vector machines (SVM). We show that it can be calculated by summing over the ranks of the positive examples (or equivalently of the negative ones), similar to Eq. 2.

While classification accuracy is often *assessed* in terms of the 0/1-loss, the 0/1-loss is computationally expensive to optimize when *training* the classifier. For the optimization task, researchers thus typically resort to approximations or bounds of the 0/1-loss. Among other loss functions, the (linear) *hinge loss* [11] (plus a penalty for regularization) is commonly used as objective function when learning SVMs. The hinge loss has several favorable properties, including: (1) it is an upper bound on the 0/1-loss, (2) it is differentiable everywhere except for one point, and (3) leads to a convex optimization problem. The hinge loss [11] of the real-valued classifier-outputs c_i, given the threshold θ and the data D, is typically defined as $L_\theta^{\mathrm{H}} = \sum_{i=1}^{N} [1 - y_i(c_i - \theta)]_+$, where $[\cdot]_+$ denotes the positive part, i.e., $[a]_+ = a$ if $a > 0$, and 0 otherwise. In analogy, we propose the following rank-version of the standard hinge loss:

Definition 1 (Hinge Rank Loss). *We define as the* hinge rank loss*, based on the ranks r_i w.r.t. the rank-threshold $\tilde{\theta}$*:

$$L_{\tilde{\theta}}^{\mathrm{HR}} = \sum_{i=1}^{N} \left[\frac{1}{2} - y_i(r_i - \tilde{\theta}) \right]_+ . \qquad (3)$$

Note that $r_i - \tilde{\theta} \in \{\pm 1/2, \pm 3/2, ...\}$, cf. Section 2, so that $[1/2 - y_i(r_i - \tilde{\theta})]_+ \in \{0, 1, 2, ...\}$. Both L^{H} and L^{HR} share the same relevant properties: the loss incurred due to each misclassified example i is at least 1 (hence both are an upper bound on the 0/1-loss), and it increases linearly in r_i.[3] Conversely, no loss L^{HR} is incurred for any correctly classified example, as desirable (in contrast to hinge loss). Note that the 'hinge ranking loss' defined in [12] is different from our definition, as discussed in Section 8.

Next, we re-write the hinge rank loss in terms of the sum over the ranks of the positive examples. For notational convenience, we will use the definition

[3] If we had defined the hinge rank loss with 1 in place of 1/2, the results in this paper would hold with only minor changes.

$\bar{\theta} = \tilde{\theta} - 1/2 \in \mathbb{N}$ for the rank-threshold in place of the (equivalent) definition in Section 2. Hence, the examples with ranks $r_i \leq \bar{\theta}$ get classified as negatives, and the examples with ranks $r_i \geq \bar{\theta} + 1$ as positives. Now we can present

Proposition 1. *For the hinge rank loss from Definition 1 holds*

$$L_{\bar{\theta}}^{\mathrm{HR}} = N_{\bar{\theta}}^{\mathrm{fn}} + N^+ \bar{\theta} + \binom{N - \bar{\theta} + 1}{2} - \sum_{j=1}^{N^+} r_j^+ \ , \tag{4}$$

with the number of false negatives $N_{\bar{\theta}}^{\mathrm{fn}} = \sum_{j=1}^{N^+} \mathbf{1}_{r_j^+ \leq \bar{\theta}}$.

Proof: Decomposing the sum in the definition in Eq. 3 into one for either class, and summing only over the non-zero arguments, one obtains $L_{\bar{\theta}}^{\mathrm{HR}} = \sum_{j=1:r_j^+ \leq \bar{\theta}}^{N^+} \left(1 - r_j^+ + \bar{\theta}\right) + \sum_{k=1:r_k^- > \bar{\theta}}^{N^-} \left(r_k^- - \bar{\theta}\right)$. Concerning the right-most sum, we use the following identity regarding all the ranks greater than $\bar{\theta}$, $\sum_{j=1:r_j^+ > \bar{\theta}}^{N^+} (r_j^+ - \bar{\theta}) + \sum_{k=1:r_k^- > \bar{\theta}}^{N^-} (r_k^- - \bar{\theta}) = \binom{N - \bar{\theta} + 1}{2}$. Now the former equation can be rewritten in terms of sums over the positive examples only (or equivalently over the negative ones only). Merging the two sums over the positive examples into one, it follows $L_{\bar{\theta}}^{\mathrm{HR}} = \sum_{j=1:r_j^+ \leq \bar{\theta}}^{N^+} 1 - \sum_{j=1}^{N^+} (r_j^+ - \bar{\theta}) + \binom{N - \bar{\theta} + 1}{2}$, which yields Eq. 4. □

5 Hinge Rank Loss and AUC

In this section, we decompose the AUC or Wilcoxon-Mann-Whitney-statistic used in ranking problems in terms of the hinge rank loss and the 0/1-loss used in classification tasks. From Eqs. 2 and 4, it follows immediately:

Proposition 2. *The AUC is related to the hinge rank loss and the number of false negatives as follows:*

$$A = 1 - \frac{L_{\bar{\theta}}^{\mathrm{HR}} - \mathrm{const}_{D,\bar{\theta}} - N_{\bar{\theta}}^{\mathrm{fn}}}{N^+ N^-} \quad \text{and} \quad A \geq 1 - \frac{L_{\bar{\theta}}^{\mathrm{HR}} - \mathrm{const}_{D,\bar{\theta}}}{N^+ N^-}, \tag{5}$$

where $\mathrm{const}_{D,\bar{\theta}} = \binom{N^- - \bar{\theta} + 1}{2}$ *if* $N^- \geq \bar{\theta}$ *and* $\mathrm{const}_{D,\bar{\theta}} = \binom{\bar{\theta} - N^-}{2}$ *otherwise;* $\mathrm{const}_{D,\bar{\theta}}$ *is a constant given the data* D *(and thus* N^+ *and* N^-*) and the rank-threshold* $\bar{\theta}$*, i.e., it is independent of the classifier* C*.*

Not only is the hinge rank loss $L_{\bar{\theta}}^{\mathrm{HR}}$ an upper bound on the 0/1-loss (as discussed earlier), but also it is the decisive term in the lower bound on the AUC, as apparent from the non-negativity of $N_{\bar{\theta}}^{\mathrm{fn}}$ in Eq. 5.

Proposition 3. *The lower bound in Eq. 5 is tight in the asymptotic limit* $N \to \infty$ *under the mild assumption that* $N^+/N \to \mathrm{const}_D^+$*, where* $0 < \mathrm{const}_D^+ < 1$ *is a constant.*

Proof: It has to be shown that $N_{\bar{\theta}}^{\mathrm{fn}} / (N^+ N^-) \to 0$ as $N \to \infty$, as this is the only term omitted from Eq. 5 as to obtain the bound in Eq. 5. This is trivial because

$0 \leq N^{\text{fn}}_{\bar{\theta}} \leq N$, and $N/(N^+N^-) \to 0$ as $N \to \infty$ under the assumption $N^+/N \to \text{const}^+_D > 0$. Hence, in the non-separable case, we have $1 - A > \text{const} > 0$, while $N^{\text{fn}}_{\bar{\theta}}/(N^+N^-) \to 0$, which hence becomes negligible for large N. In the separable case, we have $A = 1$ and from Eq. 5 thus $L^{\text{HR}}_{\bar{\theta}} - \text{const}_{D,\bar{\theta}} = N^{\text{fn}}_{\bar{\theta}}$, so that the bound indeed approaches 1 in the asymptotic limit and thus becomes tight. □

The asymptotic tightness of this bound implies that the minimum of the hinge rank loss indeed coincides with the maximum of the AUC in the asymptotic limit (our experiments in Section 7 indicate that this holds already for rather small data sets in excellent approximation). This desirable property is not guaranteed for the loose bounds on the AUC used in the literature, cf. Section 8.

Apart from that, Eq. 5 relates the AUC, which is independent of threshold $\bar{\theta}$, with the terms $L^{\text{HR}}_{\bar{\theta}}$, $\text{const}_{D,\bar{\theta}}$ and $N^{\text{fn}}_{\bar{\theta}}$, which all depend on $\bar{\theta}$. The validity of Eq. 5 implies that the effect of different values $\bar{\theta}$ cancels out among those terms.

An interesting special case of Eq. 5 is obtained for the natural choice of the threshold $\bar{\theta} = N^-$, so that the *predicted number* of positive (negative) examples equals the *true number* of positives (negatives); or equivalently, the number of false positives equals the number of false negatives. This choice has two effects: (1) it minimizes the constant $\text{const}_{D,\bar{\theta}}$, namely it vanishes; (2) it holds that $N^{\text{fn}}_{N^-} = L^{0/1}_{N^-}/2$, where the latter is the 0/1-loss. We thus obtain the

Corollary: *For the choice $\bar{\theta} = N^-$, the relation among AUC, hinge rank loss and 0/1-loss reads:*

$$A = 1 - \frac{L^{\text{HR}}_{N^-} - \frac{1}{2}L^{0/1}_{N^-}}{N^+N^-} \quad \text{and} \quad A \geq 1 - \frac{L^{\text{HR}}_{N^-}}{N^+N^-}. \tag{6}$$

6 Hinge Loss as a Parametric Approximation

In this section, we argue that minimizing the (standard) hinge loss—as it is the parametric counterpart of the hinge rank loss—can be expected to be a good approximation to maximizing the AUC, especially after pre-processing the data by an affine transformation, like (kernel-) PCA (principal component analysis) [1] with subsequent rescaling along each principal component, or the ellipsoidal machine [2].

While minimizing the hinge rank loss during training would provide an asymptotically tight bound on the AUC (as shown in the previous section), this is computationally expensive due to its discrete nature. For computational reasons, a standard approach is to approximate a discrete function by a continuous (and possibly differentiable or convex) one, which then can be optimized more efficiently. For instance, the Wilcoxon-Mann-Whitney-statistic has been approximated by various differentiable functions in [14,15] for efficient optimization by means of gradient descent (but then suffered from quadratic computational complexity due to the gradient).

Our propositions suggest an alternative approximation for efficient optimization: as the asymptotically-tight lower bound on the AUC is maximized by

minimizing the hinge rank loss, the latter may simply be approximated by its parametric counterpart: the standard hinge loss, which is computationally less costly to minimize (as done e.g. in standard SVM training). The rank-threshold $\bar{\theta}$ is accordingly replaced by (real-valued) threshold θ, cf. Sections 2 and 4.

Note that there exist rank and parametric versions of many commonly-used statistics, like the Pearson correlation coefficient and the Spearman rank correlation; or the paired Student's t-test and the Wilcoxon signed-rank test. In general, the ranked and the parametric versions of a statistic were found to yield similar results in experiments, even though no theoretical proofs exist in general. Further experimental insights include that there can also be some differences; in particular, rank statistics are independent of an (assumed) distribution of the examples, and are typically less affected by outliers in the data.

Even though the validity of the approximation of the hinge rank loss by the (standard) hinge loss cannot be proven theoretically, it is strongly supported not only by our experiments in Section 7 but also by many experimental findings in the literature, where standard SVM training yielded surprisingly high AUC values [3,4,5,6], cf. also Section 8.

Given a linear classifier (in feature space), consider an arbitrary orientation of its hyperplane: regarding the hinge rank loss, the contribution of an individual misclassified example grows monotonically with its distance from the hyperplane for both the parametric hinge loss and the hinge rank loss, however at possibly different rates. Note that the maximum possible contribution of a misclassified example to the hinge rank loss is a constant (determined by the number of examples) for *any* orientation of the hyperplane; this does not hold for the standard hinge loss, especially if the examples are squished along some dimensions. This suggests a necessary condition for the hinge loss to be a good approximation to the hinge rank loss: rescaling the (feature) space such that the examples have the *same spread in every direction*. This can be achieved by the ellipsoidal machine [2], which determines the bounding ellipsoid of the examples (possibly allowing for outliers for robustness) and transforms the examples such that they lie within a *sphere*. Alternatively, a similar affine transformation can be obtained by using principal component analysis and rescaling / normalizing along every principal component j by a factor N/std_j (where std_j is the standard deviation along principal component j, cf. Section 7 for an example), or using its kernelized version [1]. The resulting improvement is illustrated in Section 7, which also shows that such an affine transformation can make the hinge loss more robust against outliers. Moreover, note that such an affine transformation is independent of and has a different effect than the (standard) penalty term for regularization; for instance, the latter would not solve the issue in the third example in Fig. 1.

7 Experiments

In this section, we experimentally evaluated how the hinge rank loss, the (standard) hinge loss and the 0/1-loss are related to the AUC. We assessed this in two different ways, as outlined in the following, using artificial data as well as 8

Table 1. This table shows the AUC values obtained after optimizing the following measures: AUC, hinge rank loss (L^{HR}), (standard) hinge loss (L^{H}),(standard) hinge loss after affine transformation of data (aff. L^{H}), and 0/1-loss ($L^{0/1}$) on artificial data (cf. Fig. 1) and 8 data sets from the UCI repository

data set	dim.	examples	AUC	L^{HR}	L^{H}	aff. L^{H}	$L^{0/1}$
artificial data 1	2	1000	0.998	0.998	0.998	0.998	0.998
artificial data 2	2	1000	0.848	0.848	0.848	0.848	0.848
artif. with outliers	2	1000	0.796	0.796	0.574	0.796	0.796
Sonar	60	208	0.996	0.996	0.973	0.996	0.950
Glass	10	214	0.992	0.992	0.987	0.988	0.957
Ionosphere	33	351	0.983	0.983	0.976	0.981	0.958
SPECTF	44	267	0.956	0.952	0.930	0.937	0.910
Pima	8	768	0.841	0.841	0.819	0.837	0.835
Hayes-Roth	4	129	0.730	0.730	0.704	0.710	0.703
Hepatitis	19	80	1.000	1.000	1.000	1.000	0.972
Echocardiogram	7	107	0.826	0.826	0.724	0.798	0.793

data sets from the UCI machine learning repository.[4] In a pre-processing step, we discarded the examples with missing values for simplicity, as the remaining examples still provide a 'real-world distribution' for comparing the measures.

As a linear classifier, we used a hyperplane. When learning this classifier w.r.t. the various measures, only the hinge loss is not invariant under re-scaling of the data. We thus re-scaled the data in two different ways, as to illustrate the improvements due to the affine transformation discussed in Section 6. In the first version, we re-scaled each dimension k by the factor N/std_k, where std_k is the standard deviation of the examples regarding dimension k. In the second version, we applied PCA and re-scaled along each principal component analogously (called the affine transformation in the remainder).

Tab. 1 summarizes the AUC-values achieved after optimizing the various performance measures.[5,6] As expected, direct maximization of the AUC resulted in the highest AUC-values, but the difference to minimizing the hinge rank loss appeared negligible, as expected due to the asymptotic tightness of our bounds in Eqs. 5 and 6. Moreover, minimizing the standard hinge loss also yielded quite good AUC-scores. As expected, the affine transformation leads to a notable improvement. In comparison, optimizing the 0/1-loss was clearly inferior to minimizing the hinge rank loss, as expected. Moreover, the 0/1-loss was also slightly worse than the (standard) hinge loss applied to the data after the affine transformation. This suggests that the latter can indeed serve as a useful parametric

[4] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[5] We omitted the standard penalty term here, as regularization is an important but different problem. It can simply be included when classifying *unseen* examples, after the preprocessing step regarding the affine transformation.

[6] For ease of implementation, we used a simulated annealing scheme with random distortions of the hyperplane as to optimize the 'discrete' measures directly.

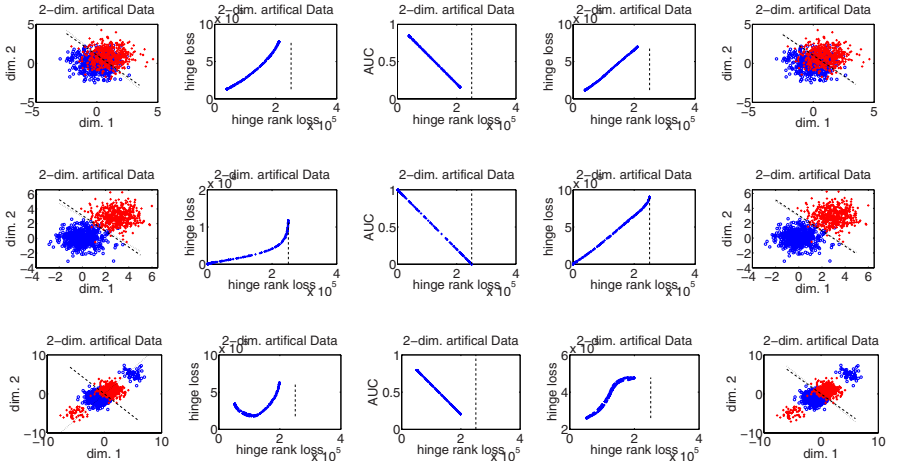

Fig. 1. Two-dimensional artificial data: either class (+, ∘) is represented by 500 examples, sampled from a standard normal distribution. The first two experiments differ in the distance between the centroids of the two classes. In the third experiment, both classes contain 10% of outliers. The two leftmost columns show the results for the hinge loss without the affine transformation, and the two rightmost columns show the results (in the original space) when using the affine transformation. The column in the center applies to both cases. The dashed line is the optimal hyperplane based on the hinge rank loss, while the dotted line is optimal w.r.t. the (standard) hinge loss.

approximation to the hinge rank loss, as it is computationally less expensive to optimize.

While the previous evaluation is only concerned with the *optimum* AUC-value, in our second assessment we evaluated the relationship between the hinge loss and the AUC for *all possible* AUC-values. For each data set, we randomly sampled various orientations of the hyperplane. We chose the rank-threshold $\bar{\theta} = N^-$, as in the Corollary, and determined the corresponding (parametric) threshold as the average parametric classifier-output for the two examples with ranks N^- and $N^- + 1$.[7] Then we calculated—for each orientation of the hyperplane—the parametric hinge loss, the hinge rank loss and the AUC; the scatter-plots in Figs. 1 and 2 illustrate the relationship between these measures (each point corresponds to a different random orientation of the hyperplane). The vertical dashed line indicates the largest possible value of the hinge rank loss given N^+ positive and N^- negative examples: it equals $N^+N^- + \min\{N^+, N^-\}$, but its value may not be attained for a given data set due to the configuration of the examples.

Concerning the first two artificial data sets in Fig. 1, the scatter-plots indicate a clear monotonic relationship between the parametric hinge loss (with and

[7] As long as the parametric and rank thresholds correspond to each other, any other choice may have been used as well.

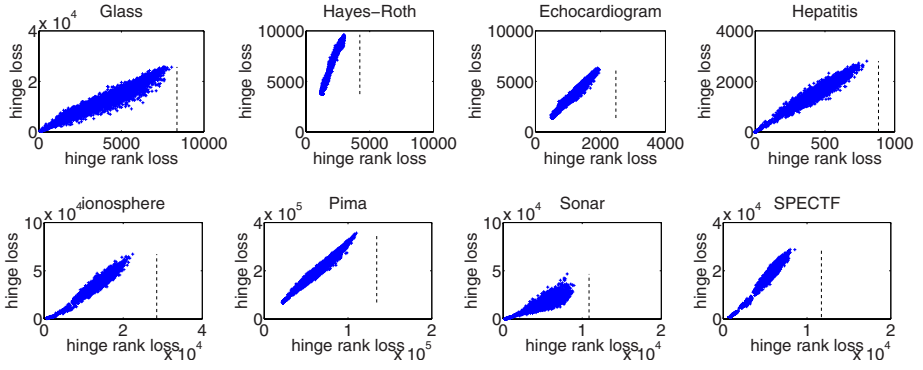

Fig. 2. The scatter-plots show the relationship of our hinge rank loss with the hinge loss on 8 data sets from the UCI machine learning repository

without the affine transformation) and the hinge rank loss, and between the hinge rank loss and the AUC. Hence, minimizing the hinge loss is an excellent approximation to minimizing the hinge rank loss, and to maximizing the AUC. Only in the third experiment, where many outliers are present, the optimal hyperplane w.r.t. the hinge loss (without affine transformation) differs significantly from the result based on the hinge rank loss, as expected (cf. Section 6); the corresponding scatter-plot shows a non-monotonic relationship, illustrating that the minima of the two loss functions are vastly different. The graphs on the lower right in Fig. 1 show the benefit of the affine transformation: when the (standard) hinge loss is applied to the data after the affine transformation, the scatter-plot shows a monotonic relationship, as desired, and the optimal hyperplanes w.r.t. either loss function are very similar. Apart from that, note that the relationship of the hinge rank loss and the AUC is well approximated by a linear function, as expected from the asymptotic tightness of the lower bounds in Eqs. 5 and 6.

Fig. 2 shows the relationship of the hinge rank loss and the (standard) hinge loss applied to 8 data sets from the UCI repository after the affine transformation: also here, it is notably monotonic. The fact that this relationship is different for each data set is irrelevant for minimization as long as it is monotonic. Apart from that, this relationship is much more 'noisy' for these real-world data sets than for our artificial data sets. Interestingly, the 'noise level' typically decreases as the hinge loss decreases, so that minimizing the hinge loss appears to be a good approximation to optimizing the hinge rank loss, and hence the AUC. Only for the data set 'Echocardiogram', the noise level is large for small values of the hinge loss, and thus the optimal hyperplanes with respect to the two loss functions are notably different, possibly due to dominating outliers.

Like in Fig. 1, we also found for these real-world data sets that the relationship between the hinge rank loss and the AUC is linear in excellent approximation (the plots have to be omitted due to lack of space), as expected.

8 Related Work

This section describes related work concerning boosting and SVMs in the context of ranking, and points out differences to our approach. A 'hinge ranking loss' was first defined in [12]. Its main difference to our Definition 1 is that they measure the difference in the ranks among all incorrectly ordered *pairs* of examples, so their measure is essentially quadratic in the rank-differences, while our measure is linear. As our hinge rank loss provides an asymptotically tight bound on the AUC, it is clear that the 'hinge ranking loss' does not.

Boosting can be understood as gradient descent, and the various flavors of boosting essentially differ in the objective function or in the (heuristic) minimization method, e.g., [16]. The objective function minimized by RankBoost [17], L^{RBoost}, provides a lower bound on the AUC [13], namely $1-L^{\mathrm{RBoost}}/(N^+N^-) \leq A$. It has essentially the same form as our bounds in Eqs. 5 or 6 involving the hinge rank loss. While our bounds are asymptotically tight, RankBoost optimizes a loose bound, as $L^{\mathrm{RBoost}} = \sum_{j=1}^{N^+}\sum_{k=1}^{N^-} \mathrm{e}^{c_k^- - c_j^+} \geq \sum_{j=1}^{N^+}\sum_{k=1}^{N^-} \mathbf{1}_{c_j^+ < c_k^-} = N^+N^-(1-A)$, cf. [13], where each c_j^+ (or c_k^-) denotes the weighted sum over the weak learners' outputs for the positive (or negative) example j (or k). In [13], it was also shown that AdaBoost's loss function equals the one of RankBoost in the case where the positive and negative examples contribute equally to the loss.

The average and the variance of the AUC statistic were derived in [18], revealing interesting relations to the misclassification rate, among other properties. In [19], confidence intervals for the AUC were obtained from these results by applying Chebyshev's inequality. The average and variance of the AUC was calculated with respect to all possible rankings with fixed misclassification rate, where each ranking got implicitly assigned the same probability/weight. This average-case analysis of a combinatorial problem may only be of limited use in practice: given fixed data, it is unlikely that all possible rankings occur with the same probability; in fact, many rankings may not occur at all in a given data set (e.g., cf. the scatter-plots in Section 7, where the extreme (small and large) values are actually not reached in many data sets). A different kind of generalization bounds were derived in [20].

It was mentioned in [3] that optimizing standard SVMs leads to maximizing the AUC in the special (trivial) case when the given data is separable. As a perfect separation implies an AUC of 1 (which is maximal), the more interesting case is non-separable data. Our results are derived without any assumptions on the kind of classifier used or the (non-)separability of the given data.

Apart from that, it was experimentally observed in [3] that there was no significant difference in AUC-scores between SVMs trained in the standard way and other approaches tailored to directly maximize the AUC, like RankBoost [17], AUCsplit (local optimization of AUC) [21], or ROC-SVM [3]. This provides additional support for the point made in this paper, namely that the hinge rank loss can indeed be accurately approximated by its parametric counterpart, the standard hinge loss.

In [4], the objective was to directly maximize the AUC when learning SVMs, which led to slight experimental improvements over the standard SVM training. This approach may be considered a special case of the SVM-approach to ordinal regression [22]. Both gradient-descent methods suffered from quadratic computational complexity, which made additional approximations necessary for computational reasons.

In [5], a generalized SVM approach was developed that is able to optimize multivariate non-linear performance measures in polynomial time, including AUC among others. The experiments focused on 4 data sets with unbalanced class distributions: in this scenario, their new approach was superior to standard SVMs when assessed with respect to the F_1-score or the precision/recall breakeven area. However, when assessed with respect to the AUC, the superiority of their new approach over standard SVMs appeared less convincing on the 4 data sets presented.

Among the many performance measures compared experimentally in [6], it was found that '... maximum margin methods such as boosting and SVMs ... surprisingly ... also yield excellent performance on the ordering metrics.'

In summary, the experimental observations in the literature, e.g., [3,4,5,6], suggest that—despite the various sophisticated methods tailored to directly maximize the AUC—standard SVMs could not be consistently outperformed when assessed with respect to the AUC. This paper provides a simple explanation: minimizing the (standard) hinge loss typically is an accurate approximation to maximizing the AUC.

9 Conclusions

We have derived a simple equation that relates the Area under the ROC Curve (AUC) with the hinge-rank-loss and the number of false negatives. This immediately yields an asymptotically-tight lower bound on the AUC, based on the hinge rank loss. While the surprisingly high AUC-scores after standard SVM training in the literature provide indirect evidence, our experiments corroborate directly that minimization of the (standard) hinge loss typically is an accurate approximation to minimizing the hinge rank loss, especially after applying affine transformations like in ellipsoidal machines. In summary, this suggests that standard SVM training typically is a simple, yet effective and computationally efficient way of approximately maximizing the AUC.

Acknowledgments. I am grateful to R. Bharat Rao for encouragement and support of this work, and to the anonymous reviewers for excellent comments.

References

1. Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
2. Shivaswamy, P., Jebara, T.: Ellipsoidal machines. In: Proc. Int. Conf. on Artificial Intelligence and Statistics, pp. 481–488 (2007)

3. Rakotomamonjy, A.: Optimizing area ROC curve with SVMs. In: workshop "ROC Analysis in AI" at the European Conference on Artificial Intelligence (2004)
4. Brefeld, U., Scheffer, T.: AUC maximizing support vector learning. In: workshop "ROC Analysis in Machine Learning" at Int. Conf. on Machine Learning (2005)
5. Joachims, T.: A support vector method for multivariate performance measures. In: Proc. Int. Conf. on Machine Learning, pp. 377–384 (2005)
6. Caruana, R., Niculescu-Mizil, A.: Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: Proc. Int. Conf. on Knowledge Discovery and Data Mining, pp. 69–78 (2004)
7. Hand, D.J., Till, R.J.: A simple generalization of the area under the ROC curve for multiple class classification problems. Machine Learning 45, 171–186 (2001)
8. Wu, S., Flach, P.: A scored AUC metric for classifier evaluation and selection. In: workshop "ROC Analysis in Machine Learning" at Int. Conf. on Machine Learning (2005)
9. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics 1, 80–83 (1945)
10. Mann, H.B., Whitney, D.R.: On a test whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 18, 50–60 (1947)
11. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
12. Agarwal, S., Niyogi, P.: Stability and generalization of bipartite ranking algorithms. In: Proc. Conf. on Learning Theory, pp. 32–47 (2005)
13. Rudin, C., Cortes, C., Mohri, M., Schapire, R.: Margin-based ranking meets boosting in the middle. In: Proc. Conf. on Learning Theory, pp. 63–78 (2005)
14. Yan, L., Dodier, R., Mozer, M.C., Wolniewicz, R.: Optimizing classifier performance via the Wilcoxon-Mann-Whitney statistics. In: Proc. Int. Conf. on Machine Learning, pp. 848–855 (2003)
15. Herschtal, A., Raskutti, B.: Optimising the area under the ROC curve using gradient descent. In: Proc. Int. Conf. on Machine Learning, pp. 49–56 (2004)
16. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. The Annals of Statistics 38, 337–374 (2000)
17. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933–969 (2003)
18. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. Advances in Neural Information Processing Systems 16, 313–320 (2003)
19. Cortes, C., Mohri, M.: Confidence intervals for the area under the ROC curve. Advances in Neural Information Processing Systems 17, 305–312 (2004)
20. Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., Roth, D.: Generalization bounds for the area under the ROC curve. Journal of Machine Learning Research 6, 393–425 (2005)
21. Ferri, C., Flach, P., Hernandez-Orallo, J.: Learning decision trees using the area under the ROC curve. In: Proc. Int. Conf. on Machine Learning, pp. 139–146 (2002)
22. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: Proc. Int. Conf. on Neural Networks, pp. 97–102 (1999)