RC24129 (W0612-004) December 1, 2006
Computer Science

IBM Research Report

Robust Reductions from Ranking to Classification

Maria-Florina Balcan
Carnegie Melon University
Pittsburgh, PA 15213

Alina Beygelzimer
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

John Langford
Yahoo Research
New York, NY 10011

Gregory B. Sorkin
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Robust Reductions from Ranking to Classification

Maria-Florina Balcan[1], Alina Beygelzimer[2], John Langford[3], and Gregory B. Sorkin[4]

[1] Carnegie Melon University, Pittsburgh, PA 15213
`ninamf@cs.cmu.edu`
[2] IBM Thomas J. Watson Research Center, Hawthorne, NY 10532
`beygel@us.ibm.com`
[3] Yahoo Research, New York, NY 10011
`jl@yahoo-inc.com`
[4] IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598
`sorkin@us.ibm.com`

Abstract. We reduce ranking, as measured by the Area Under the Receiver Operating Characteristic Curve (AUC), to binary classification. The core theorem shows that a binary classification regret of r on the induced binary problem implies an AUC regret of at most $4r$. (The binary problem is to predict, given a random pair of elements in the test set, whether the first element should be ordered before the second.) This is a large improvement over naive approaches such as ordering according to regressed scores, which have a regret transform of $r \rightarrow nr$ where n is the number of elements.

1 Introduction

We study the problem of learning to rank a set of instances, robustly. In the most basic version, we are given a set of unlabeled instances belonging to two classes (0 and 1), and the goal is to rank all instances from class 0 before any instance from class 1. A common measure of success for a ranking algorithm is the area under the ROC curve (AUC). When all 0s are ranked before all 1s, the AUC is exactly 1. The loss, $1 - \text{AUC}$, is greater for mistakes at the beginning and the end of an ordering, which satisfies the intuition that an unwanted item placed at the top of a recommendation list should have a higher associated loss than when placed in the middle. A handy shorthand for understanding this loss is that it is the normalized bubble-sort distance between the predicted ordering and the true ordering (i.e., the number of pairs in the predicted ordering when a 1 comes before a 0, normalized by the number of 1s times the number of 0s).

The classification problem is simply predicting whether a label is 0 or 1 with success measured according to the error rate, i.e., the probability of a misprediction.

These two problems appear quite different: The classification loss function is defined on a per-example basis while the AUC loss is defined for sets of examples. A natural question is whether they are truly different. This paper shows that, in some precise sense, they are not. We prove that the problem of optimizing the AUC can be reduced to classification in such a way that a small number of mis-classifications cannot induce a large AUC loss. The classification problem is to predict, given a random pair of instances in the test set, whether the first instance should be ordered before the second. Thus there is a robust mechanism for translating any classifier learning algorithm into a ranking algorithm.

Several observations help understand the problem and the result better.

Relation to Regression and Classification A common way to generate a ranking is to order examples according to some regressed score or estimated conditional class probability. The problem with this approach is that it is not necessarily robust. The fundamental difficulty is exhibited by highly unbalanced test sets. If we have one 1 and many 0s, a point-wise (i.e., regression or classification) loss on the 1 with perfect prediction for the 0s can greatly harm the AUC while only slightly affecting the point-wise loss with respect to the induced distribution. This observation implies that such schemes transform point-wise loss l to AUC loss nl, where n is the number of elements we want to rank.

A similar observation holds for regrets in place of losses: point-wise regret r translates into AUC regret nr. Regret is the difference between the incurred loss and the loss of the best predictor on the same problem. The motivation behind regret analysis is that it separates errors from unremovable noise in the problem, thus the bounds apply nontrivially even on problems with large conditional noise.

Our results apply to both losses and regrets, but will be stated in terms of regrets. We show that a pairwise classifier with a regret of r on pairs implies an AUC regret of at most $4r$, for arbitrary distributions over instances. The constant 4 has been subsequently improved to 2 [BCS06], which is the best possible (see Section 4). The theorem is a large improvement over the approaches discussed above, which have a dependence on nr. For comparison, the relationship of ranking to classification is functionally tighter than has been proven for regression to binary classification ($r \rightarrow \sqrt{r}$) [LZ05].

Tournaments and Relation to the Feedback Arc Set Problem Consider a tournament where n players each play each other. What is the best way to rank the players from weakest to strongest? A natural desire is to find an ordering which agrees with the tournament on as many player pairs as possible, i.e., minimizes the number of inconsistent pairs where a higher ranked player actually lost to a lower ranked player. This optimization problem is called the minimum feedback arc set problem and it has finally been proved NP-hard in tournaments (see [A06]).

Returning to the AUC problem, consider running a tournament on the set of instances U we want to rank. The outcome of each play is determined by a classifier c trained to predict which of the two given instances should be ordered first. The tournament induced by c on U is not necessarily consistent with a linear ordering while a ranking algorithm must predict an ordering (or equivalently, a transitive tournament).

It is best to think of c as an adversary trying to induce a large AUC regret without paying much in classification regret: The adversary c specifies a tournament on U. There is some realized bipartition of U into a set of 0s and 1s (drawn from the underlying conditional distribution of label sequences given U). The bipartition is known to the adversary but unknown to the ranking algorithm. The adversary starts with a tournament of its choice where every 1 beats every 0, and it can invert (and is charged for) the outcomes of some games between a 0 and a 1. Again, the adversary can choose any subtournaments on the 1s and on the 0s for free. Given c's tournament, a ranking algorithm orders the elements of U, possibly introducing additional mistakes, i.e., pairs where a 0 beats a 1. A ranking algorithm is robust if c cannot cause the algorithm to make many mistakes without making many mistakes itself.

One way to predict an ordering is to solve the feedback arc set problem. A basic guarantee holds for a solution to this problem: If c makes at most k mistakes, then the ordering minimizing the number of inconsistent pairs will make at most $2k$ mistakes; furthermore, no solution can do better.

Another natural way to break cycles is to rank instances according to the number of wins in the tournament. (The way ties are broken is inessential; but for definiteness, assume they are broken randomly.) Coppersmith, Fleischer, and Rudra [CFR06] proved that this algorithm provides a factor of 5-approximation for the feedback arc set problem. An approximation, however, does not generally imply any finite regret transform for the AUC problem. For example, c may make no mistakes on the 0-1 pairs while inducing a non-transitive tournament on the 0s or the 1s, so an approximation that does not know the labeling can incur a non-zero number of 0-1 mistakes.

We show, however, that the algorithm that orders the elements by their number of wins in the tournament, transforms classification regret k into AUC regret of at most $4k$. Bansal, Coppersmith, and Sorkin subsequently improved the constant to 2 [BCS06], which is the best possible (see a lower bound in Section 4).

This shows that there is an alternative to solving the NP-hard problem with the same optimality guarantee: Ordering by the number of wins has *exactly* the same regret and loss transform as an optimal solution to the feedback arc set problem.

Relation to generalization bounds A number of papers analyze generalization properties of ranking algorithms (see, e.g., [FIS+03,SHD05,SN05,RCM+05]). These results analyze ranking directly by estimating the rate of convergence of empirical estimates of the ranking loss to its expectation. The bounds typically involve some complexity parameter of the class of functions searched by the algorithms (which serves as a regularizer), and some additional quantities considered relevant for the analysis. The examples are assumed to be drawn independently from some fixed distribution and the labels are often assumed to be deterministic.

The type of results in this paper is different. We bound the realized AUC performance in terms of the realized classification performance. Since the analysis is relative, it does not have to rely on any assumptions about the way the world produces data. In particular, the bounds apply when there are arbitrary high-order dependencies between examples. This seems important in a number of real-world applications where ranking is of interest.

Our analysis does not say anything about the number of samples needed to achieve a certain level of performance. Instead it says that achieved performance can be robustly transferred from classification to ranking.

2 Preliminaries

A *binary classification problem* is defined by a distribution P over $X \times \{0,1\}$, where X is some feature space and $\{0,1\}$ is the binary prediction space. The goal is to find a classifier $c : X \to \{0,1\}$ minimizing the *classification loss*,

$$e(c,P) = \mathbf{Pr}_{(x,y)\sim P}[c(x) \neq y].$$

Let $\pi : X \times X \to \{0,1\}$ be a *preference function* that, given as input any two instances in X, outputs 1 if it agrees with the ordering of its arguments, and 0 otherwise. We say that π is an ordering of a set S if it is transitive on S, i.e., its pairwise preferences are consistent with some linear ordering of elements in S. The *AUC loss* of an ordering π on a set $S \in (X \times \{0,1\})^n$ is defined as

$$l_{\mathrm{AUC}}(\pi, S) = \frac{\sum_{i\neq j} \mathbf{1}(y_i > y_j)\pi(x_i,x_j)}{\sum_{i<j} \mathbf{1}(y_i \neq y_j)}.$$

Algorithm 1 Auc-Train (labeled set S, binary learning algorithm A)

```
1. Let $S' = \{\langle (x_1, x_2), \mathbf{1}(y_1 > y_2)\rangle : (x_1, y_1), (x_2, y_2) \in S$ and $y_1 \neq y_2\}$
2. return $c = A(S')$.
```

Algorithm 2 Degree (unlabeled set U, pairwise classifier c)

```
1. For $x \in U$, let $\deg(x) = |\{x' : c(x, x') = 1, x' \in U\}|$.
2. Sort $U$ in the descending order of $\deg(x)$, breaking ties randomly.
```

(Indices i and j in the summations range from 1 to n, and $\mathbf{1}(\cdot)$ is the indicator function which is 1 if its argument is true, and 0 otherwise.)

An *AUC problem* is defined by a distribution D over $(X \times \{0,1\})^*$. The goal is to find a total ordering $\pi : X \times X \to \{0, 1\}$ minimizing the expected AUC loss on D,

$$l(\pi, D) = \mathbf{E}_{S\sim D}\, l(\pi, S).$$

Note that D may encode arbitrary dependencies between examples.

The *classification regret* of classifier c on distribution P on binary examples is defined as

$$r(c, P) = e(c, P) - \min_{c^*} e(c^*, P).$$

Similarly, the *AUC regret* of preference function π on distribution D over $(X \times \{0,1\})^*$ is given by

$$r_{\mathrm{AUC}}(\pi, D) = l(\pi, D) - \min_{\pi^*} l(\pi^*, D).$$

Our goal is to design a ranking algorithm (that uses a preference function as an oracle) such that a small classification regret of the oracle cannot imply a large AUC regret.

Finally, a pair $(x_1, y_1), (x_2, y_2)$ will be called *mixed* if $y_1 \neq y_2$.

3 Ordering by the Number of Wins: Regret Transform

The reduction consists of two components. The training part, Auc-Train (Algorithm 1), transforms mixed pairs of labeled examples into binary data.

For any process D generating datasets $S \in (X \times \{0,1\})^*$, we can define an induced distribution on binary examples in $(X \times X) \times \{0,1\}$ by first drawing S from D, and then applying Auc-Train to S. We denote this induced distribution by Auc-Train(D).

The test portion, Degree (Algorithm 2), uses the pairwise classifier (i.e., a preference function) c learned in Algorithm 1 to run a tournament on the test set, and then creates an ordering according to the number of wins in the tournament, breaking ties randomly.

There are several important practical points that follow from the fact that the analysis is independent of how the oracle is learned. Most importantly, the complexity of the test portion does not have to be quadratic in n. If, for example, $\pi(x_i, x_j) = \mathbf{1}(f(x_i) > f(x_j))$ for some learned score function $f : X \to [0, 1]$, the complexity is linear.

The remainder of this section proves the following theorem.

Theorem 1. *For all joint distributions D and all pairwise classifiers c,*

$$r_{\mathrm{AUC}}(\textsc{Degree}(\cdot, c), D) \leq 4r(c, \textsc{Auc-Train}(D)).$$

Note the quantification in the above theorem: it applies to *all* settings where Algorithms 1 and 2 are used; in particular, D may encode arbitrary dependences between examples.

Proof. Given an unlabeled test set $x^n \in X^n$, the joint distribution D induces a conditional distribution $D(Y_1, \ldots, Y_n \mid x^n)$ over the set of label sequences $\{0,1\}^n$. We prove the theorem for any fixed x^n, and then take the expectation over the draw of x^n at the end. In the remainder of the proof $Q(y^n) = D(y^n|x^n)$ is the conditional distribution over y^n given x^n. Similarly, we replace x_i with i where it is unambiguous.

The first step is to rewrite the regrets in terms of a sum over pairwise regrets. A pairwise loss is defined by

$$l_Q(i,j) = \mathbf{E}_{y^n \sim Q(Y^n)} \frac{\mathbf{1}(y_i > y_j)}{\sum_{i<j} \mathbf{1}(y_i \neq y_j)}.$$

If $l_Q(i,j) < l_Q(j,i)$, the *regret* $r_Q(i,j)$ of ordering i before j is 0; otherwise, $r_Q(i,j) = l_Q(i,j) - l_Q(j,i)$.

We can assume without loss of generality that the ordering minimizing the AUC loss (thus having zero AUC regret) is $x_1 x_2 \ldots x_n$. All regret-zero pairwise predictions must be consistent with the ordering; i.e., $r_Q(i,j) = 0$ for all $i < j$.

Lemma 2 establishes a basic property of pairwise regrets: For any pair $i < j$, the regret $r_Q(j,i)$ can be decomposed as

$$r_Q(j,i) = \sum_{k=i}^{j-1} r_Q(k+1, k).$$

The AUC regret of π on Q can thus be decomposed as a sum of pairwise regrets:

$$\begin{aligned}
r_{\mathrm{AUC}}(\pi, Q) &= l(\pi, Q) - \min_{\pi^*} l(\pi^*, Q) = \mathbf{E}_{y^n \sim Q} l(\pi, S) - \min_{\pi^*} \mathbf{E}_{y^n \sim Q} l(\pi^*, S) \\
&= \mathbf{E}_{y^n \sim Q} \frac{\sum_{i,j} \mathbf{1}(y_i > y_j)\pi(i,j)}{\sum_{i<j} \mathbf{1}(y_i \neq y_j)} - \min_{\pi^*} \mathbf{E}_{y^n \sim Q} \frac{\sum_{i,j} \mathbf{1}(y_i > y_j)\pi^*(i,j)}{\sum_{i<j} \mathbf{1}(y_i \neq y_j)} \\
&= \max_{\pi^*} \mathbf{E}_{y^n \sim Q} \frac{\sum_{i,j} \mathbf{1}(y_i > y_j)\pi(i,j) - \mathbf{1}(y_i > y_j)\pi^*(i,j)}{\sum_{i<j} \mathbf{1}(y_i \neq y_j)} \\
&= \sum_{i<j:\pi(j,i)=1} r_Q(j,i) = \sum_{k=1}^{n-1} |\{i \leq k < j : \pi(j,i) = 1\}| \cdot r_Q(k+1, k).
\end{aligned}$$

The last equality follows from the repeated use of Lemma 2.

The classification regret can also be written in terms of pairwise regrets:

$$\begin{aligned} r(c, \textsc{Auc-Train}(Q)) &= e(c, \textsc{Auc-Train}(Q)) - \min_{c^*} e(c^*, \textsc{Auc-Train}(Q)) \\ &= \max_{c^*} \mathbf{E}_{y^n \sim Q} \left[\frac{\sum_{i,j} \mathbf{1}(y_i > y_j) c(i,j) - \mathbf{1}(y_i > y_j) c^*(i,j)}{\sum_{i<j} \mathbf{1}(y_i \neq y_j)} \right] \\ &= \sum_{i<j : c(j,i)=1} r_Q(j,i) = \sum_{k=1}^{n-1} |\{i \leq k < j : c(j,i) = 1\}| \cdot r_Q(k+1,k). \end{aligned}$$

Let o_k and c_k be the coefficients with which $r_Q(k+1,k)$ appears in the above decompositions of $r_{\mathrm{AUC}}(\pi, Q)$ and $r(c, \textsc{Auc-Train}(Q))$ respectively. The proof is done if we can show that $o_k/c_k \leq 4$ for each k.

Fix k and consider a bipartition of n nodes $1,\ldots,n$ into a nonempty set W of k "winners" and a nonempty set L of $n-k$ "losers". Let T_0 be a tournament on these nodes, with the property that W dominates L: every node $j \in W$ beats every node $i \in L$. Let T_c be the tournament corresponding to our classifier c. Both tournaments are on the same set of nodes.

The coefficient o_k is the number of pairs (j,i) such that $j \leq k < i$ but i is ordered before j in π:

$$\begin{aligned} o_k &= \sum_{i \in L} \sum_{j \in W} \left[\mathbf{1}\left(\deg_c(i) > \deg_c(j)\right) + \tfrac{1}{2} \cdot \mathbf{1}\left(\deg_c(i) = \deg_c(j)\right) \right] \\ &\leq \sum_{i \in L} \sum_{j \in W} \mathbf{1}\left(\deg_c(i) \geq \deg_c(j)\right), \end{aligned}$$

where $\deg_c(i)$ is the number of wins in T_c.

Also, given the two tournaments T_0 and T_c, let $\rho(i,j) = 0$ if T_0 and T_c agree on the direction of (i,j), and 1 otherwise. The classifier's cost function is then

$$c_k = \sum_{i \in L} \sum_{j \in W} \rho(i,j).$$

To complete the proof we need to show that $o_k/c_k \leq 4$. Theorem 2 proves precisely that. ∎

We will need the following lemma, due to Landau [Lan53].

Lemma 1. *There exists a tournament with outdegree sequence* $\deg(1) \leq \deg(2) \leq \cdots \leq \deg(n)$ *if and only if, for all* $1 \leq i \leq n$, $\sum_{j=1}^{i} \deg(j) \geq \sum_{j=1}^{i} (j-1)$, *with equality for* $i = n$.

We can now prove the remaining theorem relating the coefficients o_k and c_k.

Theorem 2. *For every* n, *every bipartition of* $\{1,\ldots,n\}$ *into nonempty sets* W *and* L, *every tournament* T_0 *in which every* $j \in W$ *dominates every* $i \in L$, *and every tournament* T_c, *with* $k = |W|$,

$$\frac{o_k}{c_k} \leq \frac{\sum_{i \in L} \sum_{j \in W} \mathbf{1}\left(\deg_c(i) \geq \deg_c(j)\right)}{\sum_{i \in L} \sum_{j \in W} \rho(i,j)} \leq 4. \tag{1}$$

The bound has subsequently been improved to $o_k/c_k \leq 2$ [BCS06], which is the best possible (see Section 4).

The proof of Theorem 2 comprises the remainder of this section.

We think of maximizing the ratio (1) over the space described by the theorem, and showing that the maximum is at most 4. The numerator of (1) depends only on T_c. If we simply transform T_0 into T_c by flipping the edges that disagree, the denominator is the number of edge reversals *between L and W*. Note that the denominator is unchanged if we replace T_0 with the tournament ${T_0}'$ which agrees with T_c on $L \times L$ and on $W \times W$, and (like T_0) has W dominating L. Thus, we may equivalently perform the maximization only over tournaments T_0 and T_c which agree on $L \times L$ and $W \times W$. For such a pair of tournaments, each edge reversal $\rho(i, j)$ contributing 1 to the denominator has the effect of increasing the degree of $i \in L$ by 1, and decreasing the degree of $j \in W$ by 1.

Thus, we may rewrite the ratio in (1) as

$$\frac{\sum_{i \in L} \sum_{j \in W} \mathbf{1}\left(\deg_c(i) \geq \deg_c(j)\right)}{\frac{1}{2}\left[\sum_{i \in L}(\deg_c(i) - \deg_0(i)) + \sum_{j \in W}(\deg_0(j) - \deg_c(j))\right]}. \tag{2}$$

Instead of maximizing the ratio only over degree sequences corresponding to tournaments satisfying the conditions of Theorem 2, we will maximize it over the broader class of sequences satisfying the following two conditions:

1. The sequence $\deg_0$ satisfies Landau's condition (Lemma 1); i.e., it is the degree sequence of some tournament T_0.
2. For all $i \in L$, $\deg_c(i) \geq \deg_0(i)$, and for all $j \in W$, $\deg_c(j) \leq \deg_0(j)$.

Note that both conditions are satisfied by tournaments obeying the theorem's conditions. This maximization is thus a relaxation of the original maximization problem; we will show that its maximum is at most 4, thus establishing the theorem.[5]

For convenience, let $\ell_1, \ldots, \ell_{|L|}$ be the nodes of L ordered so that $\deg_0(\ell_i) \geq \deg_0(\ell_{i+1})$, so for example ℓ_1 is the best of the losers (or tied for that status). Similarly, let $w_1, \ldots, w_{|W|}$ be the nodes of W ordered so that $\deg_0(w_j) \leq \deg_0(w_{j+1})$, so w_1 is the worst of the winners.

Without loss of generality we may assume that $\deg_c(\ell_i)$ is a nonincreasing sequence (like $\deg_0(\ell_i)$) and $\deg_c(w_j)$ is a nondecreasing sequence (like $\deg_0(w_j)$). This follows because we may replace any sequences $\deg_c(\ell_i)$ and $\deg_c(w_j)$ with their sorted equivalents. Clearly such a replacement does not affect the value of the denominator of (2). Also, if the original sequences satisfied condition (B), so do their sorted equivalents.

This simple fact has a nice "structural" consequence for the set of points (i, j) contributing to the numerator, call it $S = \{(i, j) \colon \mathbf{1}\left(\deg_c(\ell_i) \geq \deg_c(w_j)\right)\}$. First, if $(i, j) \in S$, then for all $i' \leq i$ and $j' \leq j$, $(i', j') \in S$ as well.

It may be helpful to imagine S as an area drawn in the positive quadrant of a sheet of graph paper: the cell $[i-1, i] \times [j-1, j]$ is filled iff $(i, j) \in S$. The condition just established asserts that in this representation of S there are no "holes": the region is a solid one running from some

[5] An easy construction shows that in this relaxation the ratio can be equal to 4, asymptotically. Thus a proof that the ratio is at most 2, has to rely on stronger properties of the pair of tournaments (see [BCS06]).

point on the j axis down in some sort of staircase pattern to some point on the i axis. (See Figures 1 and 2.)

Define $\mathsf{L}(i,j) = \{(i',j'), i,j \geq 0 \colon i' = i \text{ and } j' \leq j, \text{ or } j' = j \text{ and } i' \leq i\}$, i.e., the point (i,j) together with all points directly left of it and all points below it. Note that if $(i,j) \in S$ then $\mathsf{L}(i,j) \subset S$.

Claim. For any "staircase" region S there exists a (not necessarily perfect) matching M of rows i to columns j such that $S = \bigcup_{(i,j)\in M} \mathsf{L}(i,j)$.

That is, there is a set of Ls which form a cover of S (it is permissible for them to overlap), and whose defining "corners" all lie in distinct rows and columns.

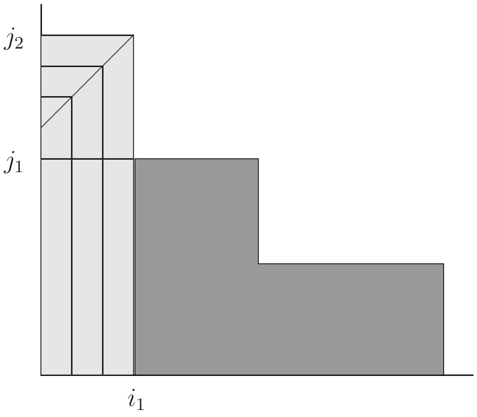

Fig. 1. Tall and skinny protrusion

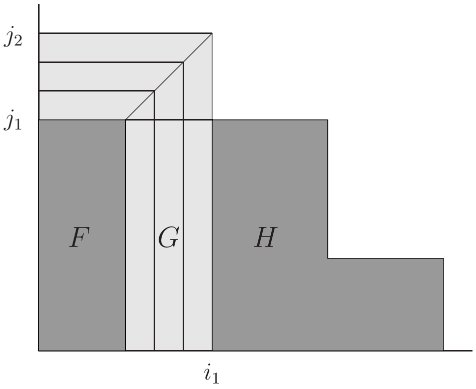

Fig. 2. Short and wide protrusion

Proof. We write (i,j) to denote a point in $\mathbb{N}^2$, and $[i,j] = \{i, i+1, \ldots, j\}$ to denote an interval in $\mathbb{N}$. The proof is by induction on the cardinality of S. Consider the topmost protrusion of S, just down to the level of the next "step" to the right. (See Figures 1 and 2.) That is, say it extends from $i = 0$ to i_1, and from $j = 0$ to j_2, with j_1 defining the height of the next-highest bit off to the right. Start covering from the protrusion's top-right corner (i_1, j_2) with nested Ls, working down and left to a point to be specified.

If the protrusion is taller than it is wide (if $j_2 - j_1 > i_1$; see Figure 1), go until you bump into the left edge (the j axis). This covers the entire leftmost tower (from $(0,0)$ to (i_1, j_2)) with Ls whose supporting columns are precisely the range $[0, j_1]$ and whose supporting rows are the range $[j_2 - j_1, j_2]$. What's left uncovered is an area right of i_1 and below j_1. By induction it can be covered with Ls with supporting columns right of i_1 and rows below j_1. This second set of Ls can thus be safely unioned with the first set, without any duplication of supporting columns or rows. All the area is covered. (The area $[0, i_1] \times [0, j_1]$ is doubly covered, which is allowed.)

If the protrusion is wider than it is tall (Figure 2), go until you bump into the horizontal line $j = j_1$. This covers the top protrusion, uses up all the rows $[j_1, j_2]$, and also uses up columns $[i_1 - (j_2 - j_1), i_1]$. The remaining area is thus all below j_1, and consists of the rectangle "F" from

$(0,0)$ to $(i_1-(j_2-j_1), j_1)$; a "gap G" (of covered area and forbidden rows) from $(i_1-(j_2-j_1), 0)$ to (i_1, j_1); and then some more complex structure "H" to the right of j_1. "Glue" the first rectangle F to the area H at the right (deleting the gap). Inductively cover this shape FH with Ls. Then pull the shape apart again (any L anchored in H now extends across the gap G). The new Ls use rows below j_1 (thus not conflicting with the first set, which were above j_1), and use columns in either F or in H (thus not conflicting with the first set, which were in G). The top rectangle and G are covered by the first set of Ls; F and H are covered by the second set; and thus the whole area is covered.

Corollary 1. *If M is a matching covering S (in the sense of Claim 3) then the numerator of* (2) *is* $\leq \sum_{(i,j)\in M}(i+j-1)$.

Proof. S is the union of the Ls, and $\mathsf{L}(i,j)$ has cardinality $i+j-1$.

Now we establish a simple condition on the degree sequence $\deg_0$. As W dominates L, it is immediate that $\deg_0(w_j) \geq |L|$ and $\deg_0(\ell_i) \leq |L|-1$.

Claim. For all i and j, $\deg_0(w_j) \geq |L| + (j-1)/2$ and $\deg_0(\ell_i) \leq |L| - (i+1)/2$.

Proof. Restricting T_0 to W gives a tournament ${T_0}^W$ whose outdegrees are $\deg'_0(w_j) = \deg_0(w_j) - |L|$. By Lemma 1, for any j, $\binom{j}{2} \leq \sum_{k=1}^{j} \deg'_0(w_k)$, which by the nondecreasing nature of W's degree sequence is $\leq j \cdot \deg'_0(w_j)$. This gives $(j-1)/2 \leq \deg'_0(w_j) = \deg_0(w_j) - |L|$, yielding the claim's first inequality.

Similarly, restricting T_0 to L gives a tournament ${T_0}^L$ with the same outdegrees, $\deg'_0(\ell_i) = \deg_0(\ell_i)$. Consider the *indegrees* within ${T_0}^L$, and note that $\operatorname{ind}'(\ell_i) + \deg'_0(\ell_i) = |L|-1$. Just as above, by Landau's theorem, for any i, $(i-1)/2 \leq \operatorname{ind}'(\ell_i) = |L| - 1 - \deg_0(\ell_i)$, yielding the claim's second inequality.

Corollary 2. *If M is a matching covering S (in the sense of Claim 3) then the denominator of* (2) *is* $\geq \frac{1}{4}\sum_{(i,j)\in M}(i+j)$.

Proof. By definition, $(i,j) \in M$ implies $(i,j) \in S$, meaning that

$$\begin{aligned} \deg_c(\ell_i) &\geq \deg_c(w_j) \\ \deg_0(\ell_i) + x(i) &\geq \deg_0(w_j) - y(j) \end{aligned}$$

and, by Claim 3,

$$x(i) + y(j) \geq \deg_0(w_j) - \deg_0(\ell_i) \geq \frac{i+j}{2}. \qquad (3)$$

In our new notation, the denominator of (2) is simply

$$\tfrac{1}{2}\left[\sum_{i=1}^{|L|} x(i) + \sum_{j=1}^{|W|} y(j)\right] \geq \tfrac{1}{2} \sum_{(i,j)\in M} \left[x(i) + y(j)\right],$$

because M is a matching and the $x(i)$ and $y(j)$ are all nonnegative. From (3), this is

$$\geq \tfrac{1}{2} \sum_{(i,j)\in M} \frac{i+j}{2}.$$

The theorem is immediate from Corollary 1 and Corollary 2.

Auxiliary Lemma We finally prove the lemma used in the proof of Theorem 1.

Lemma 2. *For any i, j, and k in x^n,*

$$r_Q(i,j) + r_Q(j,k) = r_Q(i,k).$$

Proof. Let d_{ijk} be a short-hand for the restriction of $D(Y_1, \ldots, Y_n \mid x^n)$ to $\{Y_i, Y_j, Y_k\}$. A simple algebraic manipulation verifies the claim.

$$\begin{aligned}
&r_Q(i,j) + r_Q(j,k) \\
&\qquad = d_{ijk}(100) + d_{ijk}(101) - d_{ijk}(010) - d_{ijk}(011) \\
&\qquad\qquad + d_{ijk}(010) + d_{ijk}(110) - d_{ijk}(001) - d_{ijk}(101) \\
&\qquad = d_{ijk}(100) + d_{ijk}(110) - d_{ijk}(001) - d_{ijk}(011) \\
&\qquad = r_Q(i,k),
\end{aligned}$$

Notice that all label assignments above have exactly two mixed pairs, so the factor of 1/2 is cancelled. ■

4 A Lower Bound

The following example gives a simple lower bound of $2 - \frac{4}{n+2}$ on the regret transform, both for the algorithm that orders by the number of wins and an optimal solution to the feedback arcset problem.

Example. Assume for simplicity that n is divisible by 4. Consider a bipartition of $\{1, \ldots, n\}$ into a set $U = \{1, \ldots, n/2\}$ of 0s and a set $V = \{n/2+1, \ldots, n\}$ of 1s. Consider a tournament where every node in U beats $n/4$ other nodes in U and $\frac{n}{4} - 1$ nodes in V; and every node in V beats $n/4$ nodes in U and $n/4$ nodes in V. Thus the tournament assigns $n/2$ wins to every element in V and $(n-2)/2$ wins to every element in U.

In this example, ordering minimizing the number of inconsistent pairs corresponds to ordering by the number of wins. Both algorithms order all $n/2$ 1s before any of the $n/2$ 0s, and therefore pay $(n/2)^2$ in inversions. The total number of wins in U is $\frac{n-2}{2} \cdot \frac{n}{2}$, but $\binom{n/2}{2}$ of them have to be spent internally on edges from U to U. Thus the number of cross-component edges that the adversary can direct correctly is at most $\frac{n-2}{2} \cdot \frac{n}{2} - \binom{n/2}{2} = \frac{n(\frac{n}{2}-1)}{4}$, giving the desired bound of $\frac{n^2}{n^2 - n(\frac{n}{2}-1)} = \frac{n^2}{\frac{n^2}{2}+n} = 2 - \frac{4}{n+2}$ on the ratio.

5 Relation to Other Related Work

Cortes and Mohri [CM04] tried to analyze the relationship between the AUC and the error rate on the same classification problem, treating the two as different loss functions. They derived expressions for the expected value and the standard deviation of the AUC over all classifications with a fixed number of errors, under the assumption that all such classifications are equiprobable (i.e., the classifier is as likely to err on any one example as on any other). These expressions are of little relevance to the results presented here.

Cohen, Schapire, and Singer [CSS99], similarly, use a two-stage approach to ranking: They first learn a preference function that takes a pair of instances and returns a score predicting how certain it is that the first instance should be ranked before the second. The learned function is then evaluated on all pairs of instances in the test set and an ordering approximating [6] the largest l_1 agreement possible with the predictions is created. They show that the agreement achieved by an optimal ordering is at most twice the agreement obtained by their algorithm. To translate this result into the language of losses, let Mfa be the AUC loss of a minimum feedback arcset ordering and Approx be the AUC loss of the approximation. Then the result says that $1 - \text{Approx} \geq \frac{1}{2}(1 - \text{Mfa})$ or $\text{Approx} \leq \frac{1}{2} + \text{Mfa}/2$. The settings are different making the results given here difficult to compare. Applying the result in our setting requires specializations and yields results that are weaker than ours.

References

[SHD05] S. Agarwal, S. Har-Peled, and D. Roth. A uniform convergence bound for the area under the ROC curve, *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[SN05] S. Agarwal, P. Niyogi. Stability and Generalization of Bipartite Ranking Algorithms, *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory* (COLT), 2005.

[A06] N. Alon. Ranking tournaments, SIAM J. Discrete Math. **20**: 137–142, 2006.

[BCS06] N. Bansal, D. Coppersmith, G. Sorkin. A Tournament Has at Most Twice as Many Order Misrankings as Pair Misrankings, manuscript, 2006.

[CLV05] S. Clemencon, G. Lugosi and N. Vayatis. Ranking and Scoring Using Empirical Risk Minimization, *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory* (COLT), 2005.

[CSS99] W. Cohen, R. Schapire, and Y. Singer. Learning to order things, *Journal of Artificial Intelligence Research*, 10: 243–270, 1999.

[CFR06] D. Coppersmith, L. Fleischer and A. Rudra. Ordering by Weighted Number of Wins Gives a Good Ranking for Weighted Tournaments. *Proceeding of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 776–782, 2006.

[CM04] C. Cortes and M. Mohri. AUC Optimization vs. Error Rate Minimization, *Advances in Neural Information Processing Systems* (NIPS 2003), 2004.

[FIS+03] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences, *Journal of Machine Learning Research*, 4: 933–969, 2003.

[Lan53] H. G. Landau. On Dominance Relations and the Structure of Animal Societies, *III. The Condition for a Score Structure. Bull. Math. Biophys.* 15, 143–148, 1953.

[LB05] J. Langford and A. Beygelzimer. Sensitive Error Correcting Output Codes, *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory* (COLT), 2005.

[LZ05] J. Langford and B. Zadrozny. Estimating Class Membership Probabilities Using Classifier Learners, AI+STATS 2005.

[RCM+05] C. Rudin, C. Cortes, M. Mohri, and R. Schapire. Margin-based ranking meets Boosting in the middle, *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory* (COLT), 2005.

[FS97] Y. Freund, R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139, 1997

[6] The approximation algorithm they use orders by the weighted sum of wins minus the weighted sum of loses, in the induced tournament obtained by eliminated instances that have already been ordered.

[ZLA03] B. Zadrozny, J. Langford, and N. Abe. Cost Sensitive Learning by Cost-Proportionate Example Weighting, *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, 435–442, 2003.