Ranking with a P-Norm Push

Cynthia Rudin

Center for Neural Science and Courant Institute of Mathematical Sciences
New York University / Howard Hughes Medical Institute
4 Washington Place, Room 809, New York, NY 10003-6603
rudin@nyu.edu

Abstract. We are interested in supervised ranking with the following twist: our goal is to design algorithms that perform especially well near the top of the ranked list, and are only required to perform sufficiently well on the rest of the list. Towards this goal, we provide a general form of convex objective that gives high-scoring examples more importance. This "push" near the top of the list can be chosen to be arbitrarily large or small. We choose ℓ_p-norms to provide a specific type of push; as p becomes large, the algorithm concentrates harder near the top of the list. We derive a generalization bound based on the p-norm objective. We then derive a corresponding boosting-style algorithm, and illustrate the usefulness of the algorithm through experiments on UCI data. We prove that the minimizer of the objective is unique in a specific sense.

1 Introduction

The problem of supervised ranking is useful in many application domains, e.g., document processing, customer service routing, and drug discovery. Many of these domains require the construction of a ranked list, yet often, only the top portion of the list is used in practice. For instance, in the setting of supervised movie ranking, the learning algorithm provides the user (an avid movie-goer) with a ranked list of movies based on preference data. We expect the user to examine the top portion of the list as a recommendation. It is possible that she never looks at the rest of the list, or examines it only briefly. Thus, we wish to make sure that the top portion of the list is correctly constructed. This is the problem on which we concentrate.

Naturally, the design of these rankings requires a tradeoff. Given the option, we would correct a misrank towards the top of the list at the expense of possibly making a new misrank towards the bottom. This type of sacrifice will have to be made; assuming a learning machine with finite capacity, the best total ranking will not often correspond to the best ranking near the top of the list. The trick is to design an algorithm that knows when a misrank occurs at the top and forces us to pay a high price for it, relative to other misranks.

We have developed a somewhat general and fairly flexible technique for solving these types of problems. In our framework, a specific price is assigned for each misrank; the misranks at the top are given higher prices, and the ones towards

G. Lugosi and H.U. Simon (Eds.): COLT 2006, LNAI 4005, pp. 589–604, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the bottom are less expensive. Thus, the choice of these prices determines how much emphasis (or "push") is placed closer to the top. We may only desire to incorporate a small push; it is possible, for example, that our movie-goer has seen all of the movies near the top of the list and needs to look farther down in order to find a movie she has not seen. It is important that the rest of the list be sufficiently well-constructed in this case. The desired size of the push might be anywhere between very large and very small depending on the application. There is simply a tradeoff between the size of the push and the sacrifice made farther down the list. As mentioned, some sacrifice must always be made since, as usual, we take our algorithm to have limited capacity in order to enable generalization ability. Using the form of ranking objective introduced in Section 2, one can make the prices very high for misranking near the top (a big push), moderately high (a little push), or somewhere in between.

The algorithms we develop are motivated in the usual setting of supervised bipartite ranking. In this setting, each training instance has a label of +1 or -1, i.e., each movie is either a good movie or a bad movie. Here, we want to push the bad movies away from the top of the list where the good movies are desired. The quality of the ranking can be determined by examining the Receiver Operator Characteristic (ROC) curve. In the setting where all misranks are equally priced (no push), the AUC (Area Under the ROC Curve) is precisely a constant times one minus the total standard misranking error (see [4]). However, the quantity we measure in our problem is different. We care mostly about the leftmost portion of the ROC curve for this problem, corresponding to the top of the ranked list. This is precisely the sacrifice we must make; in order make the leftmost portion of the curve higher, we must sacrifice on the total area underneath the curve.

This problem is highly asymmetric with respect to the positive and negative classes. It is interesting to consider generalization bounds for such an asymmetric problem; we should not rely on a symmetrization step which requires natural symmetry. The generalization bound presented here holds even under such asymmetric conditions. The measure of complexity is the L_∞ covering number.

Recently, there has been a large amount of interest in the supervised ranking problem, and especially in the bipartite problem. Freund et al. have developed the RankBoost algorithm for the general setting [8]. We inherit the setup of RankBoost, since our algorithms will also be boosting-style algorithms. Oddly, there is a recent theoretical proof that Freund and Schapire's classification algorithm called AdaBoost [9] performs just as well for bipartite ranking as RankBoost; i.e., both algorithms achieve equally good values of the AUC [13, 14]. There are a number of algorithms designed to maximize variations of the AUC, for instance Mozer et al. [11] aim to manipulate specific points of the ROC curve in order to study "churn" in the telecommunications industry. Perhaps the closest algorithm to ours is the one proposed by Dekel et al. [6], who have used a similar form of objective with different specifics for the score to achieve a different goal, namely to rank labels. The work of Yan et al. [17] contains a brief mention of a method that optimizes the lower left corner of the ROC curve with a multi-layer perceptron approach that is highly non-convex. There is much

recent work on generalization bounds for supervised ranking [8, 2, 1, 16, 13], though only the covering number bounds [13] can be naturally adapted to this setting due to the asymmetry of the problem.

In Section 2, we present a general form of objective function, allowing us to incorporate a push near the top of the ranked list. One must choose a loss function ℓ and a convex price function g to specify the objective function. If the price function is steep (e.g., the power law $g(r) = r^p$), then the push near the top is very strong. In Section 3, we provide a generalization bound for the objective function, for the "0-1" loss and the power law price function. In Section 4, we derive the "P-Norm Push" Algorithm, which is a coordinate descent algorithm based on the objective function. In Section 5, we prove that the minimizer of the algorithm's objective function is unique in a specific sense. This result is based on conjugate duality and the theory of Bregman distances [7], and is analogous to the result of Collins et al. [3] for AdaBoost. In Section 6, we demonstrate the P-Norm Push algorithm on UCI data. In Section 7, we use the generalization bound of Section 3 to indicate the limit of the algorithm's problem domain; we aim to find when the algorithm should (and should not) be used.

2 A General Objective for Ranking with a Push

The set of instances with positive labels is $\{\mathbf{x}_i\}_{i=1,\ldots,I}$, where $\mathbf{x}_i \in \mathcal{X}$. The negative instances are $\{\tilde{\mathbf{x}}_k\}_{k=1,\ldots,K}$, where $\tilde{\mathbf{x}}_k \in \mathcal{X}$. We always use i for the index over positive instances and k over negative instances. Our goal is to construct a ranking function $f : \mathcal{X} \to \mathcal{R}$, $f \in \mathcal{F}$ that gives a score to each instance in $\mathcal{X}$. Unlike in classification, we do not care about the exact values of each instance, only the relative values; for positive-negative pair $\mathbf{x}_i, \tilde{\mathbf{x}}_k$, we do not care if $f(\mathbf{x}_i) = .4$ and $f(\tilde{\mathbf{x}}_k) = .1$, but we do care that $f(\mathbf{x}_i) > f(\tilde{\mathbf{x}}_k)$, or that $f(\mathbf{x}_i) - f(\tilde{\mathbf{x}}_k) = .3$.

Let us now derive the general form of objective function as promised in the introduction. For a particular negative example, we wish to reduce its *Height*, i.e., the number of positive examples that are ranked beneath it. That is, for each k, we wish to make Height(k) small, where:

$$\text{Height}(k) := \sum_{i=1}^{I} \mathbf{1}_{[f(\mathbf{x}_i) \leq f(\tilde{\mathbf{x}}_k)]}.$$

Let us now add the push. We want to concentrate harder on negative examples with large Height's; we want to push these examples down from the top. Thus, for convex, non-negative, monotonically increasing function $g : \mathcal{R}_+ \to \mathcal{R}_+$, we place the price $g(\text{Height}(k))$ on negative example k. If g is very steep, we pay an extremely large price for a high-ranked negative example. Examples of steep functions include $g(r) = \exp(r)$ and $g(r) = r^p$ for p large; the latter price function will be used for the P-Norm Push. Thus we have derived an objective to minimize:

$$R_{g,\mathbf{1}}(f) := \sum_{k=1}^{K} g\left(\sum_{i=1}^{I} \mathbf{1}_{[f(\mathbf{x}_i) \leq f(\tilde{\mathbf{x}}_k)]}\right).$$

If $R_{g,\mathbf{1}}(f)$ is small, then no negative example is ranked very highly; this is exactly our design. It is hard to minimize $R_{g,\mathbf{1}}$ directly due to the 0-1 loss in the inner sum. Instead, we minimize an upper bound, $R_{g,\ell}$, which incorporates $\ell : \mathcal{R} \to \mathcal{R}_+$, a convex, non-negative, monotonically decreasing upper bound on the 0-1 loss. Popular loss functions include the exponential, logistic, and hinge losses. We can now define the general form of objective:

$$R_{g,\ell}(f) := \sum_{k=1}^{K} g\left(\sum_{i=1}^{I} \ell(f(\mathbf{x}_i) - f(\tilde{\mathbf{x}}_k))\right).$$

To construct a specific version of this objective, one chooses the loss ℓ, the price function g, and an appropriate hypothesis space $\mathcal{F}$ over which to minimize $R_{g,\ell}$.

For the moment, assume we care only about the very top of the list, that is, we wish to push the most offending negative example as far down the list as possible. Equivalently, we wish to minimize $R_{\max}$, the number of positives below the highest ranked negative example: $R_{\max}(f) := \max_k \text{Height}(k)$. It is hard to minimize $R_{\max}(f)$ directly, but $R_{g,\ell}$ can give us some control over this quantity. Namely, the following relationships exist between $R_{g,\ell}$, $R_{g,\mathbf{1}}$ and $R_{\max}$.

Theorem 1

$$Kg\left(\frac{1}{K}R_{\max}(f)\right) \leq R_{g,\mathbf{1}}(f) \leq R_{g,\ell}(f) \quad \textit{and} \quad R_{g,\mathbf{1}}(f) \leq Kg(R_{\max}(f)).$$

The proof uses Jensen's inequality for convex function g, monotonicity of g, and the fact that ℓ is an upper bound on the 0-1 loss. Theorem 1 suggests that $R_{g,\ell}$ is a reasonable quantity to minimize in order to incorporate a push at the top, e.g., in order to diminish $R_{\max}$. If g is especially steep, e.g., $g(r) = r^p$ for p large, then $g^{-1}(\sum_{k=1}^{K} g(r_k)) \approx \max_k r_k$, i.e., $g^{-1}(R_{g,\mathbf{1}}) \approx R_{\max}$. From now on, we specifically consider the power law (or "p-norm") objectives. Since the user controls p, the amount of push can be specified to match the application.

3 A Generalization Bound for the p-Norm Objective

This bound is an adaptation of previous work [14, 13] inspired by works of Koltchinskii and Panchenko [10] and Cucker and Smale [5]. Assume that the positive instances $\{\mathbf{x}_i \in \mathcal{X}\}_{i=1,\ldots,I}$ are chosen independently and at random (iid) from a fixed but unknown probability distribution $\mathcal{D}_+$ on $\mathcal{X}$. The negative instances $\{\tilde{\mathbf{x}}_k \in \mathcal{X}\}_{k=1,\ldots,K}$ are chosen iid from $\mathcal{D}_-$. The notation $\mathbf{x} \sim \mathcal{D}$ means $\mathbf{x}$ is chosen randomly according to $\mathcal{D}$. The notation $S_+ \sim \mathcal{D}_+^I$ means each of the I elements of the training set S_+ are chosen iid according to $\mathcal{D}_+$. Similarly for $S_- \sim \mathcal{D}_-^K$. We now define the "true" objective function for which our algorithm has been designed. Our goal is to make this quantity small:

$$\begin{aligned} R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f &:= \left(\mathbb{E}_{\mathbf{x}_-\sim\mathcal{D}_-}\left(\mathbb{E}_{\mathbf{x}_+\sim\mathcal{D}_+}\mathbf{1}_{[f(\mathbf{x}_+)-f(\mathbf{x}_-)\leq 0]}\right)^p\right)^{1/p} \\ &= \|\mathbb{P}_{\mathbf{x}_+\sim\mathcal{D}_+}(f(\mathbf{x}_+) - f(\mathbf{x}_-) \leq 0|\mathbf{x}_-)\|_{L_p(\mathcal{X},\mathcal{D}_-)}\,. \end{aligned}$$

The empirical loss associated with $R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f$ is:

$$R^p_{S_+,S_-}\mathbf{1}_f := \left(\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{I}\sum_{i=1}^{I}\mathbf{1}_{[f(\mathbf{x}_i)-f(\tilde{\mathbf{x}}_k)\leq 0]}\right)^p\right)^{1/p}.$$

Here, for a particular $\tilde{\mathbf{x}}_k$, $R^p_{S_+,S_-}\mathbf{1}_f$ takes into account the average number of positive examples that have scores below $\tilde{\mathbf{x}}_k$. It is a monotonic function of $R_{g,\mathbf{1}}$. To make this notion more general, consider the average number of positive examples that have scores *close to* or below $\tilde{\mathbf{x}}_k$, namely:

$$R^p_{S_+,S_-}\mathbf{1}^\theta_f := \left(\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{I}\sum_{i=1}^{I}\mathbf{1}_{[f(\mathbf{x}_i)-f(\tilde{\mathbf{x}}_k)\leq \theta]}\right)^p\right)^{1/p}.$$

This terminology incorporates the "margin" value θ. Now we can state our generalization bound:

Theorem 2. *For all $\epsilon > 0, \theta > 0$, and $f \in \mathcal{F}$:*

$$\mathbb{P}_{S_+\sim\mathcal{D}_+^I, S_-\sim\mathcal{D}_-^K}\left[R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f \leq R^p_{S_+,S_-}\mathbf{1}^\theta_f + \epsilon\right]$$

$$\geq 1 - 2\mathcal{N}\left(\mathcal{F}, \frac{\epsilon\theta}{8}\right)\left[\exp\left[-2\left(\frac{\epsilon}{4}\right)^{2p}K\right] + \exp\left[-\frac{\epsilon^2}{8}I\right]\right].$$

Here $\mathcal{N}(\mathcal{F},\epsilon)$ is the L_∞ covering number for $\mathcal{F}$. The theorem says that if I and K are large, then with high probability, the true error $R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f$ is not too much more than the empirical error $R^p_{S_+,S_-}\mathbf{1}^\theta_f$. The proof is in Appendix A.

As noted, this is a generalization bound for a compulsorily asymmetric problem. It is important to note the implications of this bound for scalability. Since we are concentrating on the negative examples near the top of the ranked list (corresponding to a small chunk of negative input space), we must require more negative examples to achieve high accuracy, as we discuss in Section 7.

Theorem 2 provides a theoretical justification for our choice of objective. Let us now write an algorithm for minimizing that objective.

4 A Boosting-Style Algorithm

We choose a specific form for $R_{g,\ell}$ by specifying ℓ as the exponential loss, $\ell(r) = \exp(-r)$. One could easily choose another loss; we chose the exponential loss in order to compare with RankBoost, which corresponds to the $p = 1$ case for our price function $g(r) = r^p$. Our family of objective functions is thus:

$$F_p(f) := \sum_{k=1}^{K}\left(\sum_{i=1}^{I}\exp[-f(\mathbf{x}_i) + f(\tilde{\mathbf{x}}_k)]\right)^p.$$

Note that F_p is not normalized to approximate $R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f$, but this can easily be accomplished via $\frac{1}{I(K)^{1/p}}(F_p(f))^{1/p}$, which is monotonically related to $F_p(f)$.

Now we describe our boosting-style approach. The hypothesis space $\mathcal{F}$ is the class of linear combinations of "weak" rankers $\{h_j\}_{j=1,\ldots,n}$, where $h_j : \mathcal{X} \rightarrow [0,1]$. The function f is constructed as: $f = \sum_j \lambda_j h_j$, where $\boldsymbol{\lambda} \in \mathcal{R}^n$. At iteration t, the coefficient vector (denoted by $\boldsymbol{\lambda}_t$) is updated. To describe how each individual weak ranker j ranks each positive-negative pair i, k, we use a structure $\mathbf{M}$ defined element-wise by: $M_{ikj} := h_j(\mathbf{x}_i) - h_j(\tilde{\mathbf{x}}_k)$. Thus, $M_{ikj} \in [-1, 1]$. To define right multiplication, we write the product element-wise as: $(\mathbf{M}\boldsymbol{\lambda})_{ik} := \sum_{j=1}^n M_{ikj}\lambda_j = \sum_{j=1}^n \lambda_j h_j(\mathbf{x}_i) - \lambda_j h_j(\tilde{\mathbf{x}}_k)$ for $\boldsymbol{\lambda} \in \mathcal{R}^n$. Thus, $\ell(f(\mathbf{x}_i) - f(\tilde{\mathbf{x}}_k))$ can now be written as $\exp(-\mathbf{M}\boldsymbol{\lambda})_{ik}$. By construction, F_p is convex in $\boldsymbol{\lambda}$ (but not strictly convex).

We now derive a boosting-style coordinate descent algorithm for minimizing F_p as a function of $\boldsymbol{\lambda}$, notating F_p now as $F_p(\boldsymbol{\lambda})$. We start with the objective at iteration t: $F_p(\boldsymbol{\lambda}_t) := \sum_{k=1}^K \left(\sum_{i=1}^I \exp[(-\mathbf{M}\boldsymbol{\lambda}_t)_{ik}]\right)^p$. We then compute the variational derivative along each "direction", and choose weak ranker j_t to have largest variational derivative. Define the vector $\mathbf{q}_t$ on pairs i, k as: $q_{t,ik} := \exp[(-\mathbf{M}\boldsymbol{\lambda}_t)_{ik}]$, and $\mathbf{d}_t$ as: $d_{t,ik} := q_{t,ik}/\sum_{ik} q_{t,ik}$. Let the vector $\mathbf{e}_j$ be 1 in position j and 0 elsewhere. Then j_t becomes:

$$j_t \in \operatorname*{argmax}_j \left[-\frac{dF_p(\boldsymbol{\lambda}_t + \alpha\mathbf{e}_j)}{d\alpha}\Big|_{\alpha=0}\right] = \operatorname*{argmax}_j \left[\sum_{k=1}^K \left[\left(\sum_{i=1}^I d_{t,ik}\right)^{p-1} \sum_{i=1}^I d_{t,ik} M_{ikj}\right]\right].$$

To update the coefficient of weak ranker j_t, we now perform a linesearch for the minimum of F_p along the j_t^{th} direction. The distance to travel in the j_t^{th} direction, denoted α_t, solves $0 = \frac{dF_p(\boldsymbol{\lambda}_t + \alpha\mathbf{e}_{j_t})}{d\alpha}\Big|_{\alpha_t}$, or incorporating normalization,

$$0 = \sum_{k=1}^K \left[\left(\sum_{i=1}^I d_{t,ik} \exp[-\alpha_t M_{ikj_t}]\right)^{p-1} \left(\sum_{i=1}^I M_{ikj_t} d_{t,ik} \exp[-\alpha_t M_{ikj_t}]\right)\right]. \quad (1)$$

The value of α_t can be computed analytically in special cases, but more generally, we use a linesearch to solve for α_t. The full algorithm is shown in Figure 1.

5 Uniqueness of the Minimizer

One might hope that a function $f = \sum_j \lambda_j h_j$ (or limit of functions) minimizing our objective is unique in some sense. Since $\mathbf{M}$ is not required to be invertible (and often is not), a minimizing $\boldsymbol{\lambda}$ may not be unique. Furthermore, elements of $\boldsymbol{\lambda}_t$ and $\mathbf{M}\boldsymbol{\lambda}_t$ may approach $\pm\infty$ or ∞ respectively, so it would seem difficult to prove (or even define) uniqueness. It is useful to consider the set $Q' := \{\mathbf{q}' \in \mathcal{R}_+^{IK} | q'_{ik} = e^{-(\mathbf{M}\boldsymbol{\lambda})_{ik}} \text{ for some } \boldsymbol{\lambda} \in \mathcal{R}^n\}$; with the help of convex analysis, we show that our objective function yields a unique minimizer in the closure of $\mathcal{Q}'$.

1. **Input:** $\{\mathbf{x}_i\}_{i=1,...,I}$ positive examples, $\{\tilde{\mathbf{x}}_k\}_{k=1,...,K}$ negative examples, $\{h_j\}_{j=1,...,n}$ weak classifiers, $t_{\max}$ number of iterations, p power.
2. **Initialize:** $\lambda_{1,j} = 0$ for $j = 1, ..., n$, $d_{1,ik} = 1/IK$ for $i = 1, ..., I$, $k = 1, ..., K$ $M_{ikj} = h_j(\mathbf{x}_i) - h_j(\tilde{\mathbf{x}}_k)$ for all i, k, j
3. **Loop for** $t = 1, ..., t_{\max}$
   (a) $j_t \in \text{argmax}_j \left[\sum_{k=1}^{K} \left[\left(\sum_{i=1}^{I} d_{t,ik}\right)^{p-1} \sum_{i=1}^{I} d_{t,ik} M_{ikj}\right]\right]$.
   (b) Find a value α_t that solves (1). That is, perform a linesearch for α_t.
   (c) $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \alpha_t \mathbf{e}_{j_t}$, where $\mathbf{e}_{j_t}$ is 1 in position j_t and 0 elsewhere.
   (d) $z_t = \sum_{ik} d_{t,ik} \exp[-\alpha_t M_{ikj_t}]$
   (e) $d_{t+1,ik} = d_{t,ik} \exp[-\alpha_t M_{ikj_t}]/z_t$ for $i = 1, ..., I$, $k = 1, ..., K$
4. **Output:** $\boldsymbol{\lambda}_{t_{\max}}$

Fig. 1. Pseudocode for the "P-Norm Push Algorithm"

Theorem 3. *Define* $Q' := \{\mathbf{q}' \in \mathcal{R}_+^{IK} | q'_{ik} = e^{-(M\boldsymbol{\lambda})_{ik}}$ *for some* $\boldsymbol{\lambda} \in \mathcal{R}^n\}$ *and define* closure(Q') *as the closure of* $\mathcal{Q}'$ *in* $\mathcal{R}^{IK}$. *Then,* $\mathbf{q}'^* \in$ closure(Q') *is uniquely determined by:*

$$\mathbf{q}'^* = \text{argmin}_{\mathbf{q}' \in \text{closure}(Q')} \sum_k \left(\sum_i q'_{ik}\right)^p.$$

Our uniqueness proof (in Appendix B) depends mainly on the theory of convex duality for a class of Bregman distances, as defined by Della Pietra et al. [7]. This proof is inspired by Collins et al. [3] who have proved uniqueness of this type for AdaBoost. In the case of AdaBoost, the primal optimization problem corresponds to a minimization over relative entropy. In our case, the primal is not a common function.

6 Experiments

We will now show the effect of adding a push by examining the leftmost portion of the ROC curve. Our goal is to illustrate the effect of the price g on the quality of the solution; the choice of g as a power law allows us to explore this effect. We hope that $R_{\max}$, or more generally, the leftmost portion of the ROC curve, increases steadily with p. Our demonstration shows this firmly; $R_{\max}$ does often increase (fairly dramatically) with p, for both training and testing.

Data for these experiments were obtained from the UCI machine learning repository [15]. Settings chosen were: **pima-indians-diabetes** with threshold features (Figure 2), **wdbc - Wisconsin Breast Cancer** (Figure 3) and **housing** (Figure 4). The (normalized) features themselves were used as the weak rankers. Results from other datasets can be found in the longer version of this paper [12]. The linesearch for α_t was performed using matlab's 'fminunc' subroutine. The total number of iterations, $t_{\max}$, was fixed at 200. In agreement

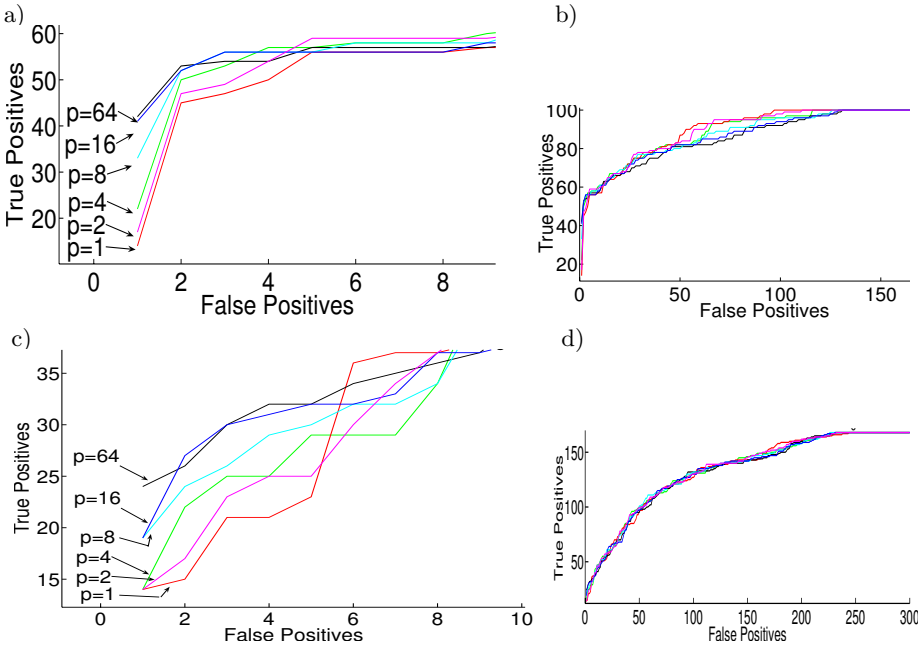

Fig. 2. pima-indians-diabetes with threshold features: 4 threshold features were obtained from each real valued feature, via $h_{\text{thresh}}(\mathbf{x}) = 1$ iff $h(\mathbf{x}) >$ thresh, and $h_{\text{thresh}}(\mathbf{x}) = 0$ otherwise. Thresholds used were chosen so that no two threshold features would be equivalent with respect to the training data. Of 768 examples, 300 randomly chosen examples were used for training, and the rest for testing. (a) Leftmost portion of scaled ROC curve for training, up to and including the crossover point where the sacrifice begins. (b) Full scaled ROC training curve. (c) Leftmost portion of scaled ROC curve for testing. (d) Full scaled ROC testing curve.

with our algorithm's derivation, a larger push (p large) causes the algorithm to perform better near the top of the ranked list. As discussed, this ability to correct the top of the list is not without sacrifice; we do sacrifice the ranks of items farther down on the list, but we have made this choice on purpose. We believe it is important to show this sacrifice explicitly, thus full ROC curves have been included for all experiments. The **housing** setting yields the clearest view of the effect of the algorithm. The trend in $R_{\max}$ from $p = 1$ to $p = 64$ is clearly present and close to monotonic. There is a distinct crossover region, showing exactly what parts of the ROC curve are gained and what parts are sacrificed.

7 Limitations

We have included this section in order to more explicitly describe the problem domain for which the algorithm is useful. As no one algorithm is the best

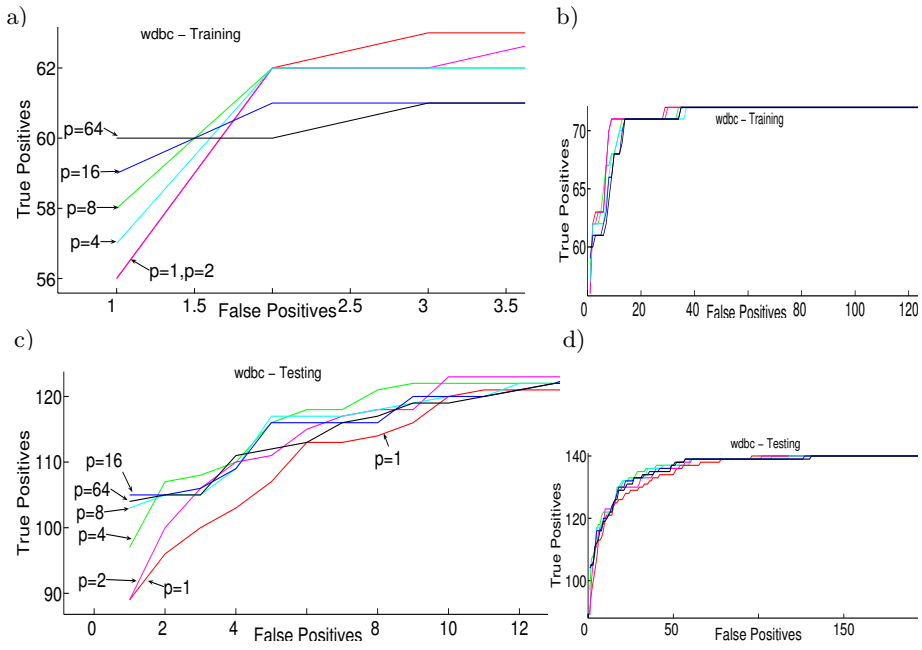

Fig. 3. wdbc (Wisconsin Breast Cancer): 569 total examples, 200 used for training. To ensure the algorithm would not achieve a separable solution, only the first six features (columns 3-8) were used. All features were normalized to $[0, 1]$. (a) Leftmost portion of scaled ROC curve for training (b) Full scaled ROC training curve. (c) Leftmost portion of scaled ROC curve for testing. (d) Full scaled ROC testing curve.

for every problem setting, we wish to make as clear as possible the settings in which our algorithm is meant to succeed, and in which domains it is not meant to be used. The most definitive boundary of the problem domain involves the sample size. The generalization bound of Theorem 2 indicates that for larger values of p, many more examples are needed in order to allow generalization ability; we are concentrating on a smaller region of the probability distribution, so this is natural. When the sample size is too small, the algorithm may still be able to generalize for smaller values of p, but for larger values, we cannot expect the training curve to represent the testing curve. For the settings shown in Section 6, we have used a few hundred examples per experiment, which is enough to allow the algorithm to generalize. In contrast, we now present a setting that compliments our theoretical prediction; the setting is the pima-indian-diabetes dataset with normalized real-valued features, but only 50 training examples. Above a certain p value, the performance degrades as p increases as shown in Figure 5. This shows (what we believe is) the main cautionary note to experimentalists when using this algorithm, and for that matter, when using any other algorithm that concentrates on a small part of the input space.

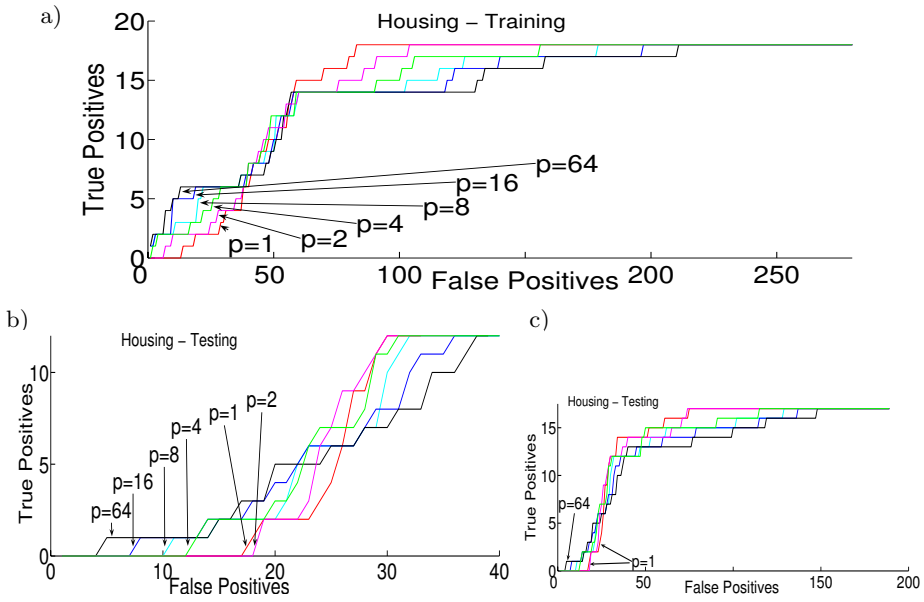

Fig. 4. housing (Boston Housing): 506 total examples, 300 used for training, 13 (normalized) features. The fourth column (which is binary) was used as the label y. The label specifies whether a tract bounds the Charles River. Since there is some correlation between the label and the features, it is reasonable for our learning algorithm to predict whether a tract bounds the river. This data set is skewed; there are significantly fewer positive examples than negative examples. (a) Full scaled ROC training curve. (b) Leftmost portion of scaled ROC curve for testing. (c) Full scaled ROC testing curve.

8 Discussion and Open Problems

In Section 6, we have shown that an increase in p tends to increase $R_{\max}$, but how severe is the sacrifice that we make farther down the ranked list? All of the full ROC training curves in Section 6 (with perhaps the exception of housing) do not show any significant sacrifice, even between the $p = 1$ and $p = 64$ curves. To explain this observation, recall that we are working with learning machines of very limited capacity. The number of real valued features has not exceeded 13, i.e., there is not too much flexibility in the set of solutions that yield good rankings; the algorithm chooses the best solution from this limited choice. A high capacity learning machine generally is able to produce a consistent (or nearly consistent) ranking, so it is a delicate matter to find a dataset and hypothesis space such that an increase in p causes a dramatic change in the full ROC curve. It is an open problem to find such a dataset and function space.

Another important direction for future research is the choice of loss function ℓ and price function g. The choice of loss function is a thoroughly-studied topic, however, the choice of price function adds a new dimension to this problem. One appealing possibility is to choose a non-monotonic function for g. The only

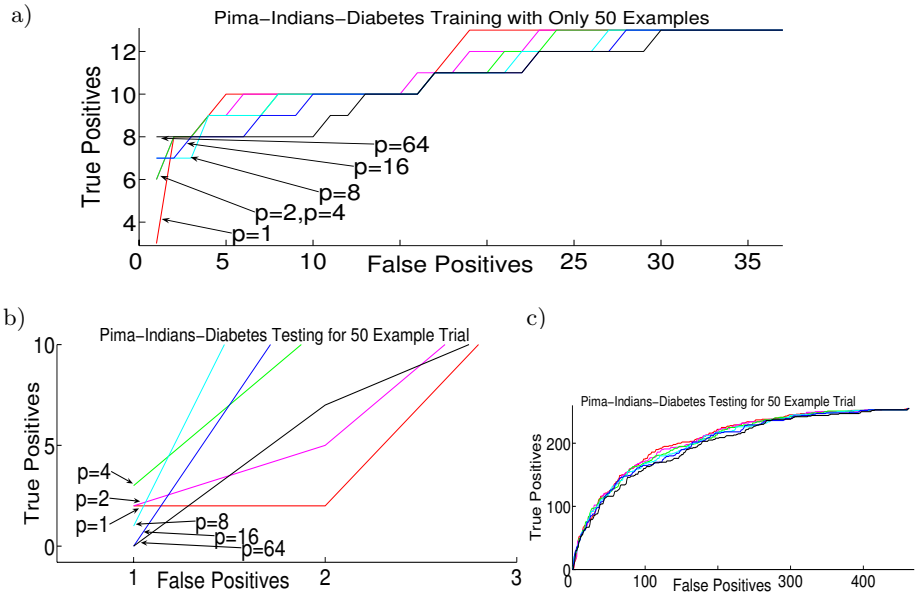

Fig. 5. The pima-indians-diabetes dataset with only 50 training examples. The algorithm is able to generalize for early values of p, but it does not generalize for large values of p. This underscores the need for a sufficiently large training set for large p values. (a) Full training ROC curve. (b) Leftmost portion of ROC testing curve. (c) Full ROC testing curve.

algorithmic requirement is that g be convex. Also, it is possible to use variations of our basic derivation in Section 2 to derive other specialized objectives. Of our experiments, the algorithm's most dramatic effect was arguably seen on the housing dataset, which is a very uneven dataset. It would be interesting to understand the algorithm's effect as a function of the unevenness of the data.

9 Conclusions

We have provided a method for constructing a ranked list where correctness at the top of the list is most important. Our main contribution is a general set of convex objective functions determined by a loss ℓ and price function g. A boosting-style algorithm based on a specific family of these objectives is derived. We have demonstrated the effect of a number of different price functions, and it is clear, both theoretically and empirically, that a steeper price function concentrates harder at the top of the list.

Acknowledgements. Thanks to Rob Schapire, Sinan Güntürk, and Eero Simoncelli. Funding for this research is provided by an NSF postdoctoral fellowship.

References

1. Shivani Agarwal, Thore Graepel, Ralf Herbich, Sariel Har-Peled, and Dan Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.
2. Stéphan Clemençon, Gabor Lugosi, and Nicolas Vayatis. Ranking and scoring using empirical risk minimization. In *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory*, 2005.
3. Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1/2/3), 2002.
4. Corinna Cortes and Mehryar Mohri. AUC optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems 16*, 2004.
5. Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.*, 39:1–49, 2002.
6. Ofer Dekel, Christopher Manning, and Yoram Singer. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems 16*, 2004.
7. Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Duality and auxiliary functions for Bregman distances. Technical Report CMU-CS-01-109R, School of Computer Science, Carnegie Mellon University, 2002.
8. Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.
9. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.
10. Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1), February 2002.
11. M. C. Mozer, R. Dodier, M. D. Colagrosso, C. Guerra-Salcedo, and R. Wolniewicz. Prodding the ROC curve: Constrained optimization of classifier performance. In *Advances in Neural Information Processing Systems 14*, pages 1409–1415, 2002.
12. Cynthia Rudin. Ranking with a p-norm push. Technical Report TR2005-874, New York University, 2005.
13. Cynthia Rudin, Corinna Cortes, Mehryar Mohri, and Robert E. Schapire. Margin-based ranking meets boosting in the middle. In *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory*, 2005.
14. Cynthia Rudin and Robert E. Schapire. Margin-based ranking and why Adaboost is actually a ranking algorithm. in progress, 2006.
15. C.L. Blake S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.
16. Nicolas Usunier, Massih-Reza Amini, and Patrick Gallinari. A data-dependent generalisation error bound for the AUC. In *Proceedings of the ICML 2005 Workshop on ROC Analysis in Machine Learning*, 2005.
17. Lian Yan, Robert H. Dodier, Michael Mozer, and Richard H. Wolniewicz. Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic. In *Proc. ICML*, pages 848–855, 2003.

A Proof of Theorem 2

We follow the outline of Rudin et al. [13]. Define a Lipschitz function $\phi : \mathcal{R} \rightarrow \mathcal{R}$ (with Lipschitz constant $\text{Lip}(\phi)$). Later we use a piecewise linear ϕ

(see [10]), but for now, take $0 \leq \phi(z) \leq 1\ \forall z$ and $\phi(z) = 1$ for $z < 0$. Since $\phi(z) \geq \mathbf{1}_{[z \leq 0]}$, we have an upper bound on $R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f$, namely, $R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_f := \left(\mathbb{E}_{\mathbf{x}_-\sim\mathcal{D}_-}(\mathbb{E}_{\mathbf{x}_+\sim\mathcal{D}_+}\phi(f(\mathbf{x}_+) - f(\mathbf{x}_-)))^p\right)^{1/p}$. The empirical error is thus:

$$R^p_{S_+,S_-}\phi_f := \left(\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i) - f(\tilde{\mathbf{x}}_k))\right)^p\right)^{1/p}.$$

First, we upper bound $R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_f$ by two terms: the empirical error term $R^p_{S_+,S_-}\phi_f$, and a term characterizing the deviation of $R^p_{S_+,S_-}\phi_f$ from $R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_f$ uniformly:

$$R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f \leq R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_f \leq \sup_{\bar{f}\in\mathcal{F}}\left(R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_{\bar{f}} - R^p_{S_+,S_-}\phi_{\bar{f}}\right) + R^p_{S_+,S_-}\phi_f.$$

The proof involves an upper bound on the first term. Let $L(f) := R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_f - R^p_{S_+,S_-}\phi_f$. The following lemma is true for every training set S:

Lemma 1. *For any two functions $f_1, f_2 \in L_\infty(\mathcal{X})$, $L(f_1) - L(f_2) \leq 4\mathrm{Lip}(\phi)||f_1 - f_2||_\infty$.*

The proof uses Minkowski's inequality twice and some algebraic manipulation. The following step is due to Cucker and Smale [5]. Let $\ell_\epsilon := \mathcal{N}\left(\mathcal{F}, \frac{\epsilon}{8\mathrm{Lip}(\phi)}\right)$, the covering number of $\mathcal{F}$ by L_∞ disks of radius $\frac{\epsilon}{8\mathrm{Lip}(\phi)}$. Define $f_1, f_2, ..., f_{\ell_\epsilon}$ to be the centers of such a cover, i.e., the collection of L_∞ disks B_r centered at f_r and with radius $\frac{\epsilon}{8\mathrm{Lip}(\phi)}$ is a cover for $\mathcal{F}$. The center of each disk will act as a representative for the whole disk. Now, the following lemma is not difficult to prove (see [5] or [13]).

Lemma 2. *For all $\epsilon > 0$,*

$$\mathbb{P}_{S_+\sim\mathcal{D}_+^I, S_-\sim\mathcal{D}_-^K}\{\sup_{f\in B_r} L(f) \geq \epsilon\} \leq \mathbb{P}_{S_+\sim\mathcal{D}_+^I, S_-\sim\mathcal{D}_-^K}\{L(f_r) \geq \frac{\epsilon}{2}\}.$$

Here is a small lemma from calculus that will be useful in the next proof.

Lemma 3. *For $a, b \in \mathcal{R}_+$, it is true that $|a^{1/p} - b^{1/p}| \leq |a - b|^{1/p}$.*

We now incorporate the fact that the training set is chosen randomly.

Lemma 4. *For all $\epsilon_1 > 0$,*

$$\mathbb{P}_{S_+\sim\mathcal{D}_+^I, S_-\sim\mathcal{D}_-^K}(L(f) \geq \epsilon_1) \leq 2\exp\left[-2\left(\frac{\epsilon_1}{2}\right)^{2p}K\right] + 2\exp\left[-\frac{\epsilon_1^2}{2}I\right].$$

Proof. Define $R^p_{S_+,\mathcal{D}_-}\phi_f := \left(\mathbb{E}_{\mathbf{x}_-\sim\mathcal{D}_-}\left(\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i) - f(\mathbf{x}_-))\right)^p\right)^{1/p}$. Now,

$$\begin{aligned}\mathbb{P}_{S_+\sim\mathcal{D}_+^I, S_-\sim\mathcal{D}_-^K}(L(f) \geq \epsilon_1) \leq\ & \mathbb{P}_{S_+\sim\mathcal{D}_+^I}\left(R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_f - R^p_{S_+,\mathcal{D}_-}\phi_f \geq \frac{\epsilon_1}{2}\right) \\ & + \mathbb{P}_{S_+\sim\mathcal{D}_+^I, S_-\sim\mathcal{D}_-^K}\left(R^p_{S_+,\mathcal{D}_-}\phi_f - R^p_{S_+,S_-}\phi_f \geq \frac{\epsilon_1}{2}\right) \\ =:\ & \mathrm{term}_1 + \mathrm{term}_2. \end{aligned} \quad (2)$$

Let us bound term$_2$. Since ϕ_f is bounded between 0 and 1, the largest possible change in $(R^p_{S_+,S_-}\phi_f)^p$ that one negative example can cause is $1/K$. Thus, McDiarmid's Inequality applied to the negative examples implies that for all $\epsilon_2 > 0$:

$$\mathbb{P}_{S_-\sim\mathcal{D}_-^K}\left[\left|\mathbb{E}_{\mathbf{x}_-\sim\mathcal{D}_-}\left(\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\mathbf{x}_-))\right)^p-\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\tilde{\mathbf{x}}_k))\right)^p\right|\geq\epsilon_2\right]$$
$$\leq 2\exp\left[\frac{-2\epsilon_2^2}{K\frac{1}{K^2}}\right]=2\exp\left[-2\epsilon_2^2K\right]. \quad (3)$$

The following is true for any S_+, due to Lemma 3 above:

$$R^p_{S_+,\mathcal{D}_-}\phi_f-R^p_{S_+,S_-}\phi_f$$
$$\leq\left|\mathbb{E}_{\mathbf{x}_-\sim\mathcal{D}_-}\left(\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\mathbf{x}_-))\right)^p-\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\tilde{\mathbf{x}}_k))\right)^p\right|^{1/p}. \quad (4)$$

Combining (3) and (4) yields a bound on term$_2$. Namely, for all $\epsilon_3 > 0$:

$$\mathbb{P}_{S_-\sim\mathcal{D}_-^K}\left(R^p_{S_+,\mathcal{D}_-}\phi_f-R^p_{S_+,S_-}\phi_f\geq\epsilon_3\right)\leq 2\exp\left[-2\epsilon_3^{2p}K\right]. \quad (5)$$

Letting $\epsilon_3 := \epsilon_1/2$ finishes our work on term$_2$. Now we consider term$_1$ of (2).

$$\mathbb{P}_{S_+\sim\mathcal{D}_+^I}\left(R^p_{\mathcal{D}_+\mathcal{D}_-}\phi_f-R^p_{S_+,\mathcal{D}_-}\phi_f\geq\frac{\epsilon_1}{2}\right)$$
$$=\mathbb{P}_{S_+\sim\mathcal{D}_+^I}\left(\left\|\mathbb{E}_{\mathbf{x}_+\sim\mathcal{D}_+}\phi(f(\mathbf{x}_+)-f(\cdot))\right\|_{L_p(\mathcal{X},\mathcal{D}_-)}-\left\|\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\cdot))\right\|_{L_p(\mathcal{X},\mathcal{D}_-)}\geq\frac{\epsilon_1}{2}\right)$$
$$\leq\mathbb{P}_{S_+\sim\mathcal{D}_+^I}\left(\left\|\mathbb{E}_{\mathbf{x}_+\sim\mathcal{D}_+}\phi(f(\mathbf{x}_+)-f(\cdot))-\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\cdot))\right\|_{L_\infty(\mathcal{X},\mathcal{D}_-)}\geq\frac{\epsilon_1}{2}\right).$$

We use McDiarmid's Inequality again to complete the proof. The largest possible change in $\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\mathbf{x}_-))$ due to the replacement of one positive example is $1/I$. Thus, for all $\mathbf{x}_-$,

$$\mathbb{P}_{S_+\sim\mathcal{D}_+^I}\left(\left|\mathbb{E}_{\mathbf{x}_+\sim\mathcal{D}_+}\phi(f(\mathbf{x}_+)-f(\mathbf{x}_-))-\frac{1}{I}\sum_{i=1}^{I}\phi(f(\mathbf{x}_i)-f(\mathbf{x}_-))\right|\geq\frac{\epsilon_1}{2}\right)\leq 2\exp\left[-\frac{\epsilon_1^2I}{2}\right].$$

Combining this result with (2) and (5) yields the statement of Lemma 4. □

Proof. (Of Theorem 2) First applying the union bound over balls, then applying Lemma 2, and then Lemma 4 (as in [13]), we find:

$$\mathbb{P}_{S_+\sim\mathcal{D}^I,S_-\sim\mathcal{D}^K}\left\{\sup_{f\in\mathcal{F}}L(f)\geq\epsilon\right\}\leq\mathcal{N}\left(\mathcal{F},\frac{\epsilon}{8\mathrm{Lip}(\phi)}\right)\left[2\exp\left[-2\left(\frac{\epsilon}{4}\right)^{2p}K\right]+2\exp\left[-\frac{\epsilon^2}{8}I\right]\right].$$

Now we put everything together. With probability at least:

$$1 - \mathcal{N}\left(\mathcal{F}, \frac{\epsilon}{8\text{Lip}(\phi)}\right)\left[2\exp\left[-2\left(\frac{\epsilon}{4}\right)^{2p}K\right] + 2\exp\left[-\frac{\epsilon^2}{8}I\right]\right], \text{ we have:}$$

$$R^p_{\mathcal{D}_+\mathcal{D}_-}\mathbf{1}_f \leq R^p_{S_+,S_-}\phi_f + \epsilon. \tag{6}$$

Let us choose $\phi(z) = 1$ for $z \leq 0$, $\phi(z) = 0$ for $z \geq \theta$, and linear in between, with slope $-1/\theta$. Thus, $\text{Lip}(\phi) = 1/\theta$. Since $\phi(z) \leq 1$ for $z \leq \theta$, we have $R^p_{S_+,S_-}\phi_f \leq R^p_{S_+,S_-}\mathbf{1}^\theta_f$. Incorporating this into equation (6) finishes the proof of the theorem. □

B Proof of Theorem 3

We will use a theorem of Della Pietra et al. [7], and follow their definitions leading to this theorem. Consider function $\phi : S \subset \mathcal{R}^{IK} \to [-\infty, \infty]$ which is *Legendre* (see [7]). The *effective domain* of ϕ, denoted Δ_ϕ, is the set of points where ϕ is finite. The *Bregman Distance* associated with ϕ is $B_\phi : \Delta_\phi \times \text{int}(\Delta_\phi) \to [0, \infty]$ defined as:

$$B_\phi(\mathbf{p}, \mathbf{q}) := \phi(\mathbf{p}) - \phi(\mathbf{q}) - < \nabla\phi(\mathbf{q}), \mathbf{p} - \mathbf{q} > .$$

(Do not confuse the vector $\mathbf{p} \in \mathcal{R}^{ik}$ with the scalar power p.) The *Legendre-Bregman Conjugate* associated with ϕ is ℓ_ϕ defined as: $\ell_\phi(\mathbf{q}, \mathbf{v}) := \sup_{\mathbf{p}\in\Delta_\phi}(< \mathbf{v}, \mathbf{p} > - B_\phi(\mathbf{p}, \mathbf{q}))$. For fixed $\mathbf{q}$, the Legendre-Bregman conjugate is the convex conjugate of $B_\phi(\cdot, \mathbf{q})$. The *Legendre-Bregman Projection* is the argument of the sup whenever it is well-defined, $\mathcal{L}_\phi : \text{int}(\Delta_\phi) \times \mathcal{R}^{IK} \to \Delta_\phi$, $\mathcal{L}_\phi(\mathbf{q}, \mathbf{v}) := \text{argmax}_{\mathbf{p}\in\Delta_\phi}(< \mathbf{v}, \mathbf{p} > - B_\phi(\mathbf{p}, \mathbf{q}))$. Della Pietra et al. [7] showed that equivalently, $\mathcal{L}_\phi(\mathbf{q}, \mathbf{v}) = (\nabla\phi)^{-1}(\nabla\phi(\mathbf{q}) + \mathbf{v})$.

The domains of the primal and dual problems will be defined with respect to a matrix $\mathbf{M} \in \mathcal{R}^{IK\times n}$, and vectors $\mathbf{q}_0, \mathbf{p}_0 \in \Delta_\phi$. The domain of the primal problem is: $\mathcal{P} = \{\mathbf{p} \in \mathcal{R}^{IK} | \mathbf{p}^T\mathbf{M} = \mathbf{p}_0^T\mathbf{M}\}$. The domain of the dual problem is:

$$\mathcal{Q}(\mathbf{q}_0, \mathbf{M}) := \{\mathbf{q} \in \Delta_\phi | \mathbf{q} = \mathcal{L}_\phi(\mathbf{q}_0, -\mathbf{M}\boldsymbol{\lambda}) \text{ for some } \boldsymbol{\lambda} \in \mathcal{R}^n\}.$$

The following theorem will give us uniqueness within the closure of $\mathcal{Q}$.

Theorem 4. *(from Proposition 3.2 of [7]) Let ϕ satisfy the technical conditions A1.-A5. of [7] and suppose there is $\mathbf{p}_0$ and $\mathbf{q}_0 \in \Delta_\phi$ with $B_\phi(\mathbf{p}_0, \mathbf{q}_0) < \infty$. Then there exists a unique $\mathbf{q}^* \in \Delta_\phi$ satisfying:*

1. $\mathbf{q}^* = \text{argmin}_{\mathbf{p}\in\mathcal{P}} B_\phi(\mathbf{p}, \mathbf{q}_0)$ *(primal problem)*
2. $\mathbf{q}^* = \text{argmin}_{\mathbf{q}\in\text{closure}(\mathcal{Q})} B_\phi(\mathbf{p}_0, \mathbf{q})$ *(dual problem)*

If we can prove that our objective function fits into this framework, this theorem will provide uniqueness in the closure of $\mathcal{Q}$, which is related to $\mathcal{Q}'$. Let us now

do this. Consider function $\phi : \mathcal{R}^{IK}_{>0} \rightarrow [-\infty, \infty]$, which is Legendre (see [12] for details):

$$\phi(\mathbf{q}) := \sum_{ik} q_{ik} g(q_{ik}, \mathbf{q}), \text{ where } g(q_{ik}, \mathbf{q}) := \ln \left(\frac{q_{ik}}{p^{1/p} (\sum_{i'} q_{i'k})^{(p-1)/p}} \right).$$

Reducing carefully, one can show: $\mathcal{L}_\phi(\mathbf{q}, \mathbf{v})_{ik} = e^{v_{ik}} q_{ik} \left(\sum_{i'} e^{v_{i'k}} q_{i'k}\right)^{(p-1)} \frac{1}{(\sum_{i'} q_{i'k})^{(p-1)}}$. Choosing $\mathbf{q}_0$ to be constant, $q_{0ik} = q_0$ for all i, k, we can now obtain $\mathcal{Q}$:

$$\mathcal{Q}(\mathbf{q}_0, \mathbf{M}) = \left\{ \mathbf{q} \in \Delta_\phi \,\middle|\, \mathbf{q} = e^{-(\mathbf{M}\boldsymbol{\lambda})_{ik}} \left(\sum_{i'} e^{-(\mathbf{M}\boldsymbol{\lambda})_{i'k}} \right)^{(p-1)} \frac{q_0}{I^{(p-1)}} \text{ for some } \boldsymbol{\lambda} \in \mathcal{R}^n \right\}.$$

In order to make the last fraction 1, let $q_0 = I^{(p-1)}$. The domain for the primal problem is fixed by choosing $\mathbf{p}_0 = \mathbf{0}$, namely $\mathcal{P} = \{\mathbf{p} \in \mathcal{R}^{IK} | \mathbf{p}^T \mathbf{M} = \mathbf{0}\}$. The dual objective is $B_\phi(\mathbf{0}, \mathbf{q})$. If $\mathbf{q} \in \mathcal{Q}$, i.e., $\mathbf{q}_{ik} = e^{-(\mathbf{M}\boldsymbol{\lambda})_{ik}} \left(\sum_{i'} e^{-(\mathbf{M}\boldsymbol{\lambda})_{i'k}}\right)^{(p-1)}$, then simplifying yields:

$$B_\phi(\mathbf{0}, \mathbf{q}) = (1/p)\, F_p(\boldsymbol{\lambda}).$$

Thus, we have arrived at exactly the objective function for our algorithm. That is, ϕ was carefully chosen so the dual objective would be exactly as we wished, modulo the constant $1/p$ which does not affect minimization. The technical conditions A1.-A5. are verified in [12]. Part (2) of Theorem 4 states that the objective function has a unique minimizer in closure($\mathcal{Q}$). It is not difficult to show that a vector in closure($\mathcal{Q}$) corresponds uniquely to a vector in closure($\mathcal{Q}'$). This finishes the proof. □

It was unnecessary to state the primary objective $B_\phi(\mathbf{p}, \mathbf{q}_0)$ explicitly to prove the theorem, however, we state it (details omitted) in order to compare with the relative entropy case where $p = 1$.

$$B_\phi(\mathbf{p}, \mathbf{q}_0) = \sum_{ik} p_{ik} \ln \left[\frac{p_{ik}}{p^{1/p} (\sum_{i'} p_{i'k})^{(p-1)/p}} \right] - \frac{1}{p}(1 - \ln p) \sum_{ik} p_{ik} + \frac{1}{p} I^p K$$

By inspection, one can see that for $p = 1$ this reduces to the relative entropy case.

One interesting note is how to find a function ϕ to suit such a problem. We discovered the function ϕ again via convex duality. We knew the desired dual problem was precisely our objective F_p, thus, we were able to recover the primal problem and thus ϕ by convex conjugation.