Generalization Bounds for *k*-Partite Ranking

Shyamsundar Rajaram
Beckman Institute
University of Illinois
Urbana, IL 61801, USA
rajaram1@ifp.uiuc.edu

Shivani Agarwal
Computer Science & AI Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
shivani@csail.mit.edu

Abstract

We study generalization properties of ranking algorithms in the setting of the k-*partite* ranking problem. In the k-partite ranking problem, one is given examples of instances labeled with one of k ordered 'ratings', and the goal is to learn from these examples a real-valued ranking function that ranks instances in accordance with their ratings. This form of ranking problem arises naturally in a variety of applications and, formally, constitutes a generalization of the bipartite ranking problem that has recently been studied. We start by defining notions of ranking error suitable for measuring the quality of a ranking function in the k-partite setting. We then give distribution-free probabilistic bounds on the expected error of a ranking function learned by a k-partite ranking algorithm.

1 Introduction

Consider the following scenario. Kim, an avid reader, is always on the look-out for good books to read. She has read several books in the last few years, and each time she reads a book, she sums up her opinion on the book in the form of a 'rating', which can range from one star (poor) to five stars (excellent). She wonders if, based on these past ratings, someone could generate for her an ordered list of all the books at her local library, such that books she is likely to give a high rating to appear at the top of the list, while books she is likely to give a lower rating to appear at the bottom of the list. She could then use this list as a guide in selecting her next book.

Kim's problem is clearly an instance of a learning problem. How can this problem be formalized? There is an instance space $\mathcal{X}$ (which in Kim's case is the space of all books), and there is a set $\mathcal{Y}$ of k ordered 'ratings', which we can take to be $\mathcal{Y} = \{1, \ldots, k\}$, with k representing the highest rating and 1 the lowest (in Kim's case, $k = 5$). The input to the problem is a finite sequence of labeled training examples $S = ((x_1, y_1), \ldots, (x_M, y_M)) \in (\mathcal{X} \times \mathcal{Y})^M$. What form should the output of a learning algorithm for this problem take?

We could ask that a learning algorithm predict ratings of new instances. The goal of the algorithm would then be to learn from S a function $h : \mathcal{X} \to \mathcal{Y}$; given a new instance $x \in \mathcal{X}$, the algorithm would predict the rating $h(x)$. There could be different ways to measure the quality of such a function h. One possibility is to simply count an error if $h(x)$ differs from the true rating y of x; the error of h with respect to the training sample S would then be

$$\widehat{L}_0(h; S) \quad = \quad \frac{1}{M} \sum_{i=1}^{M} \mathbf{I}_{\{h(x_i) \neq y_i\}} \,, \tag{1}$$

where $\mathbf{I}_{\{\cdot\}}$ denotes the indicator variable whose value is one if its argument is true and zero otherwise. This, however, reduces the problem to one of multi-class classification, in which the k ratings are treated simply as k 'classes'. Such a formulation ignores the order over the ratings: predicting four stars instead of five incurs the same penalty as predicting two stars instead of five. Another possibility is to weigh the error by the difference between the predicted rating $h(x)$ and the true rating y; in this case, the error of h w.r.t. S would be

$$\widehat{L}_1(h;S) \quad = \quad \frac{1}{M}\sum_{i=1}^{M}|h(x_i)-y_i|\,. \tag{2}$$

This corresponds to ordinal regression, and comes closer to fitting our problem.

However, remember that what Kim wants is actually an ordered list of books; in other words, the desired output is an ordering over the instances in $\mathcal{X}$. We could construct an ordering over $\mathcal{X}$ using a function $h : \mathcal{X}\rightarrow\mathcal{Y}$, that predicts ratings of instances in $\mathcal{X}$, by placing instances that are given a higher rating by h above instances that are given a lower rating by h, breaking ties at random; this would place instances predicted to have rating k at the top (in an arbitrary order), followed by instances predicted to have rating $k-1$ (again in an arbitrary order), and so on. A more natural approach, however, is to ask that a learning algorithm output directly an ordering over the instance space. Formally, we require that the algorithm learn from S a real-valued ranking function $f : \mathcal{X}\rightarrow\mathbb{R}$ that assigns scores to instances and thereby induces an ordering or *ranking* over $\mathcal{X}$: an instance $x \in \mathcal{X}$ is ranked higher by f than an instance $x' \in \mathcal{X}$ if $f(x) > f(x')$, and lower if $f(x) < f(x')$.

What would be a good way to measure the quality of a ranking function f? To answer this question, let us consider the bipartite ranking problem which has recently received some attention [7, 1]. In the bipartite ranking problem, instances in the space $\mathcal{X}$ come from two categories, 'positive' and 'negative'. The learner is given a finite number of examples of instances labeled as positive or negative, and the goal is to learn a ranking function $f : \mathcal{X}\rightarrow\mathbb{R}$ that ranks positive instances higher than negative ones. It is easy to see that this problem corresponds exactly to our problem in the special case when $k = 2$, with positive examples viewed as having the higher rating, 2, and negative examples the lower rating, 1. If n_1, n_2 denote the number of examples in the training sample S with ratings 1 and 2, respectively, then the bipartite ranking error (w.r.t. S) of a ranking function f, used to measure the quality of a ranking function in the bipartite setting, can be expressed as

$$\widehat{R}(f;S) \quad = \quad \frac{1}{n_1 n_2}\sum_{\{i:y_i=1\}}\sum_{\{j:y_j=2\}}\lambda(f,x_j,x_i)\,, \tag{3}$$

where

$$\lambda(f,x,x') \quad = \quad \mathbf{I}_{\{f(x)<f(x')\}} + \frac{1}{2}\mathbf{I}_{\{f(x)=f(x')\}}\,. \tag{4}$$

The bipartite ranking error effectively counts an error each time an instance of rating 2 is ranked lower by f than an instance of rating 1 (assuming ties are broken at random). To measure the quality of a ranking function in the general case ($k \geq 2$), we would like to use a quantity that extends naturally the bipartite ranking error above; in particular, we would like the quantity to reduce to the bipartite ranking error in the case $k = 2$.

One way to do this is to count a ranking error each time an instance of a higher rating is ranked lower by f than an instance of a lower rating:

$$\widehat{R}_0(f;S) \quad = \quad \frac{1}{C(S_Y)}\sum_{l=1}^{k-1}\sum_{m=l+1}^{k}\sum_{\{i:y_i=l\}}\sum_{\{j:y_j=m\}}\lambda(f,x_j,x_i)\,, \tag{5}$$

where we have decomposed S into $S_X = (x_1,\ldots,x_M) \in \mathcal{X}^M$ and $S_Y = (y_1,\ldots,y_M) \in \mathcal{Y}^M$, and taken $C(S_Y) = \sum_{l=1}^{k-1}\sum_{m=l+1}^{k} n_l n_m$, with n_l denoting the number of ratings in S_Y equal to l. However, this quantity does not adequately distinguish ranking errors

made between instances of ratings 4 and 5 from ranking errors made between instances of ratings 2 and 5. Such distinctions can be taken into account by measuring the error of f as

$$\widehat{R}_1(f;S) \;=\; \frac{1}{C(S_Y)} \sum_{l=1}^{k-1} \sum_{m=l+1}^{k} \sum_{\{i:y_i=l\}} \sum_{\{j:y_j=m\}} (m-l)\, \lambda(f, x_j, x_i)\,. \qquad (6)$$

If we represent the training sample S as a graph in which there is a vertex for each instance in S and an edge between each pair of vertices for which a mis-ranking of the corresponding pair of instances would contribute a non-zero error term, then under any of the above two forms of ranking error (Eqs. (5) and (6)), the resulting graph is k-partite. For this reason, we refer to this form of ranking problem as the *k-partite* ranking problem.

The problem of ranking, and the related problem of ordinal regression, have recently begun to be studied in machine learning[1] [5, 8, 6, 7, 11, 1, 4]. In the most general setting of the ranking problem [5, 7], the learner is given training examples consisting of ordered pairs of instances, each labeled with a ranking preference that indicates the importance of ranking that pair correctly, and the goal is to learn from these examples a real-valued ranking function that ranks future instances accurately. The bipartite setting of the ranking problem is perhaps one of the simplest, and several theoretical results have recently been derived for it [7, 1, 2, 3]. The k-partite setting described above is a natural extension of the bipartite setting that encompasses a wider range of applications; these include, for example, the book ranking problem described earlier, as well as many other problems that involve ranking based on ordered ratings, such as ranking movies or ranking music albums.

An important question for ranking algorithms, as for all learning algorithms, concerns generalization ability: how well does the empirical error of a learned ranking function, with respect to the training sample from which it is learned, generalize to its expected error on future data? This question has been addressed recently for the bipartite setting in [7, 1, 2], and, in the case of zero empirical error, for a more general setting in [12]. In this paper, we address this question for the k-partite setting. We start by defining in Section 2 notions of the empirical and expected error of a k-partite ranking function, and then present in Section 3 our generalization bounds for k-partite ranking algorithms.

2 *k*-Partite Ranking Error

The notions of empirical k-partite ranking error defined in Eqs. (5) and (6) can be generalized as follows:

Definition 1 (Empirical k-partite ranking error). *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$. Let $S = ((x_1, y_1), \ldots, (x_M, y_M)) \in (\mathcal{X} \times \mathcal{Y})^M$, and for each l, let n_l denote the number of examples in S with rating l. Define the* empirical k-partite ranking error *of f with respect to S, parametrized by $\alpha \geq 0$ and denoted by $\widehat{R}_\alpha(f;S)$, as*

$$\widehat{R}_\alpha(f;S) \;=\; \frac{1}{C(S_Y)} \sum_{l=1}^{k-1} \sum_{m=l+1}^{k} \sum_{\{i:y_i=l\}} \sum_{\{j:y_j=m\}} (m-l)^\alpha\, \lambda(f, x_j, x_i)\,.$$

For $k = 2$, $\widehat{R}_\alpha(f;S)$ reduces to the bipartite ranking error for each $\alpha \geq 0$. In order to define a notion of the expected k-partite ranking error of a ranking function, we need to introduce

[1] The problem considered in [8] actually lies between ordinal regression and k-partite ranking; the setting is similar to that of k-partite ranking described above, but the goal is to learn a function $h : \mathcal{X} \to \mathcal{Y}$ that assigns ratings to new instances, and the error of such a function is measured by

$$\widehat{Q}(h;S) \;=\; \frac{1}{C(S_Y)} \sum_{l=1}^{k-1} \sum_{m=l+1}^{k} \sum_{\{i:y_i=l\}} \sum_{\{j:y_j=m\}} \mathbf{I}_{\{h(x_j) \leq h(x_i)\}}\,.$$

some notation; our notation will follow largely that of [1]. As is standard in classification, we shall assume that all examples (x, y) are drawn randomly and independently according to some fixed (but unknown) distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. For each $l \in \{1, \ldots, k\}$, the notation $\mathcal{D}_l$ will be used to denote the 'class-conditional' distribution $\mathcal{D}_{x|y=l}$. Several of our results will involve the conditional distribution $\mathcal{D}_{S_X|S_Y=\mathbf{y}}$ for some label sequence $\mathbf{y} = (y_1, \ldots, y_M) \in \mathcal{Y}^M$; this distribution is simply $\mathcal{D}_{y_1} \times \ldots \times \mathcal{D}_{y_M}$.

We would like to define the expected k-partite ranking error of a ranking function $f : \mathcal{X} \to \mathbb{R}$ as its expected 'ranking loss' on a pair of instances drawn from distinct classes. It turns out that, unlike the bipartite case, the expected loss in the general k-partite case depends on the class probabilities. Thus, we define the expected error as follows:

Definition 2 (Expected k-partite ranking error). *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$. Let $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_k) \in [0, 1]^k$ be a 'class probability' vector satisfying $\sum_{l=1}^{k} \rho_l = 1$. Define the* expected k-partite ranking error *of f with respect to $\boldsymbol{\rho}$, parametrized by $\alpha \geq 0$ and denoted by $R_\alpha(f; \boldsymbol{\rho})$, as*

$$R_\alpha(f; \boldsymbol{\rho}) \;=\; \frac{1}{\gamma(\boldsymbol{\rho})} \sum_{l=1}^{k-1} \sum_{m=l+1}^{k} \rho_l \rho_m \mathbf{E}_{x \sim \mathcal{D}_l, x' \sim \mathcal{D}_m} \Big\{ (m-l)^\alpha \lambda(f, x', x) \Big\} ,$$

where $\gamma(\boldsymbol{\rho}) = \sum_{l=1}^{k-1} \sum_{m=l+1}^{k} \rho_l \rho_m$.

In an ideal situation, one would like to estimate the expected error of a learned ranking function w.r.t. the true class probabilities (under $\mathcal{D}$). This is difficult for two reasons: first, the true class probabilities are generally unknown; second, the true probabilities may be different from the empirical class probabilities. In this paper, we study how well one can estimate the expected error w.r.t. the *empirical* class probabilities.

Definition 3 (Skew vector). *Let $\mathbf{y} = (y_1, \ldots, y_M) \in \mathcal{Y}^M$ be a finite rating sequence of length $M \in \mathbb{N}$, and for each $l \in \{1, \ldots, k\}$, let $n_l = |\{i : y_i = l\}|$. Define the* skew vector *of $\mathbf{y}$, denoted by $\boldsymbol{\rho}(\mathbf{y})$, as*

$$\boldsymbol{\rho}(\mathbf{y}) \;=\; (n_1/M, \ldots, n_l/M) .$$

For each l, we shall denote by $\rho_l(\mathbf{y})$ the lth component of $\boldsymbol{\rho}(\mathbf{y})$, i.e., $\rho_l(\mathbf{y}) = n_l/M$.

We shall be interested in bounding the deviation of the empirical error of a learned ranking function f_S, w.r.t. the training sample S from which it is learned, from the expected error of f_S with respect to the skew vector of S_Y.

3 Generalization Bounds

We give here two generalization bounds for k-partite ranking algorithms. The first bound applies to algorithms that select a ranking function from a finite function class; the second bound can be applied to algorithms that search potentially infinite function classes. Note that our results are all distribution-free, in the sense that they hold for any distribution $\mathcal{D}$.

Theorem 1. *Let $\mathcal{F}$ be a finite class of real-valued functions on $\mathcal{X}$, and let $\mathcal{A}$ be a k-partite ranking algorithm that, given a training sample $S \in (\mathcal{X} \times \mathcal{Y})^M$, returns a ranking function $f_S \in \mathcal{F}$. Let $\alpha \geq 0$. Then for any $0 < \delta \leq 1$, with probability at least $1 - \delta$ (over the draw of S according to $\mathcal{D}^M$), we have*

$$\left| \widehat{R}_\alpha(f_S; S) - R_\alpha(f_S; \boldsymbol{\rho}(S_Y)) \right| \;<\; \sqrt{\left(\frac{\ln |\mathcal{F}| + \ln\left(\frac{2}{\delta}\right)}{2M} \right) \sum_{l=1}^{k} \rho_l(S_Y) \left(C_{l,\alpha}(S_Y) \right)^2} ,$$

where for $\mathbf{y} \in \mathcal{Y}^M$, $C_{l,\alpha}(\mathbf{y}) = \Big(\sum_{m \neq l} |m-l|^\alpha \rho_m(\mathbf{y}) \Big) \Big/ \gamma(\boldsymbol{\rho}(\mathbf{y}))$.

The proof of the above bound relies on the derivation of a large deviation result for the k-partite ranking error, which in turn relies on a powerful concentration inequality of McDiarmid [9] and is similar to the derivation of such a result for the bipartite case [1]. Lack of space precludes a full discussion here; complete details of the proof can be found in [10].

Our second generalization bound, which is applicable to k-partite ranking algorithms that select a ranking function from a potentially infinite function class, is expressed in terms of a new set of combinatorial parameters that we term the *k-partite rank-shatter coefficients*. These coefficients extend naturally the bipartite rank-shatter coefficients of [1].

Definition 4 (Bipartite rank matrix [1]). *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$, let $m, n \in \mathbb{N}$, and let $\mathbf{x} = (x_1, \ldots, x_m) \in \mathcal{X}^m$, $\mathbf{x}' = (x'_1, \ldots, x'_n) \in \mathcal{X}^n$. The* bipartite rank matrix *of f with respect to $(\mathbf{x}, \mathbf{x}')$, denoted by $\mathbf{B}_f(\mathbf{x}, \mathbf{x}')$, is defined to be the matrix in $\{0, 1/2, 1\}^{m \times n}$ whose (i, j)-th element (for $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$) is given by*

$$[\mathbf{B}_f(\mathbf{x}, \mathbf{x}')]_{ij} \quad = \quad \mathbf{I}_{\{f(x_i) > f(x'_j)\}} + \frac{1}{2}\mathbf{I}_{\{f(x_i) = f(x'_j)\}} \,.$$

Definition 5 (k-Partite rank matrix set). *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$, let $n_1, \ldots, n_k \in \mathbb{N}$, and for each $l \in \{1, \ldots, l\}$, let $\mathbf{x}^{(l)} = (x_1^{(l)}, \ldots, x_{n_l}^{(l)}) \in \mathcal{X}^{n_l}$. Define the k-*partite rank matrix set *of f with respect to $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)})$, denoted by $\mathbf{K}_f(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)})$, to be the following set of $\binom{k}{2}$ bipartite rank matrices:*

$$\mathbf{K}_f(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}) \quad = \quad \left\{ \mathbf{B}_f(\mathbf{x}^{(l)}, \mathbf{x}^{(m)}) \,\middle|\, 1 \leq l < m \leq k \right\}.$$

Definition 6 (k-Partite rank-shatter coefficient). *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$, and let $n_1, \ldots, n_k \in \mathbb{N}$. Define the $(n_1, \ldots, n_k)$-*th k-partite rank-shatter coefficient *of $\mathcal{F}$, denoted by $r(\mathcal{F}, n_1, \ldots, n_k)$, as follows:*

$$r(\mathcal{F}, n_1, \ldots, n_k) \quad = \quad \max_{\mathbf{x}^{(l)} \in \mathcal{X}^{n_l}} \left| \left\{ \mathbf{K}_f(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}) \mid f \in \mathcal{F} \right\} \right|.$$

Clearly, for finite $\mathcal{F}$, we have $r(\mathcal{F}, n_1, \ldots, n_k) \leq |\mathcal{F}|$ for all $n_1, \ldots, n_k$. In general, $r(\mathcal{F}, n_1, \ldots, n_k) \leq \prod_{l=1}^{k-1} \prod_{m=l+1}^{k} 3^{n_l n_m}$ for all $n_1, \ldots, n_k$.

Theorem 2. *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$, and let $\mathcal{A}$ be a k-partite ranking algorithm that, given a training sample $S \in (\mathcal{X} \times \mathcal{Y})^M$, returns a ranking function $f_S \in \mathcal{F}$. Let $\alpha \geq 0$. Then for any $0 < \delta \leq 1$, with probability at least $1 - \delta$ (over the draw of S according to $\mathcal{D}^M$), we have*

$$\left| \widehat{R}_\alpha(f_S; S) - R_\alpha(f_S; \boldsymbol{\rho}(S_Y)) \right|$$

$$< \quad \sqrt{8 \left(\frac{\ln r\big(\mathcal{F}, 2M\rho_1(S_Y), \ldots, 2M\rho_k(S_Y)\big) + \ln\left(\frac{4}{\delta}\right)}{M} \right) \sum_{l=1}^{k} \rho_l(S_Y) \left(C_{l,\alpha}(S_Y)\right)^2} \,,$$

where $C_{l,\alpha}(S_Y)$ is as defined in Theorem 1.

The proof of the above bound relies on the derivation of a uniform convergence result for the k-partite ranking error, and makes use of techniques similar to those used for the bipartite case [1]. Details (excluded here due to lack of space) can be found in [10].

The above bound is meaningful only if $r(\mathcal{F}, 2n_1, \ldots, 2n_k)$ grows sufficiently slowly with $n_1, \ldots, n_k$. Using properties of the bipartite rank-shatter coefficients [1], it can be shown that this is the case for certain function classes $\mathcal{F}$. In particular, one can obtain the following polynomial upper bound on the k-partite rank-shatter coefficients for linear ranking functions (see [10] for details):

Theorem 3. *For $d \in \mathbb{N}$, let $\mathcal{F}_{\text{lin}(d)}$ denote the class of linear ranking functions on $\mathbb{R}^d$. Then for all $n_1, \ldots, n_k \in \mathbb{N}$,*

$$r(\mathcal{F}_{\text{lin}(d)}, n_1, \ldots, n_k) \;\leq\; \prod_{l=1}^{k-1} \prod_{m=l+1}^{k} \left(\frac{2en_l n_m}{d} \right)^d .$$

A similar result can be shown for higher-order polynomial ranking functions.

4 Conclusion

The k-partite setting of the ranking problem is a natural setting that encompasses a wide range of applications. Our goal in this paper has been to initiate a formal study of this setting; we have defined notions of k-partite ranking error suitable for measuring the quality of ranking functions in the k-partite setting, and have obtained generalization bounds for k-partite ranking algorithms. While the basic techniques used to derive our results are similar to those used in the bipartite setting (which constitutes a special case of the more general k-partite setting), there are several important differences, including in particular the need in the k-partite case to define the expected error of a ranking function with respect to a vector $\boldsymbol{\rho}$ of 'class probabilities', and to consider convergence of the empirical error to the expected error with respect to an appropriate vector. It remains an open problem to derive bounds on the expected error with respect to the unknown vector of true class probabilities. Indeed, our results should be viewed as a first step in understanding the generalization behaviour of k-partite ranking algorithms. We expect that they will open the door to further analyses. For example, it can be shown that the (empirical) k-partite ranking error can be expressed as a U-statistic, which suggests that it may be possible to use tools of U-statistics [4].

Acknowledgments

We would like to thank Thore Graepel and Ralf Herbrich for pointing us to the k-partite generalization of the bipartite ranking problem. We would also like to express our gratitude to an anonymous reviewer for many insightful comments and suggestions.

References

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

[2] S. Agarwal and P. Niyogi. Stability and generalization of bipartite ranking algorithms. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

[3] S. Agarwal and D. Roth. Learnability of bipartite ranking functions. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

[4] S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

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

[6] K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, 2002.

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

[8] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[9] C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics 1989*, pages 148–188. Cambridge University Press, 1989.

[10] S. Rajaram and S. Agarwal. Generalization bounds for k-partite ranking. Report, 2005. Available from `http://www.ifp.uiuc.edu/~rajaram1/k-ranking-report.pdf`.

[11] S. Rajaram, A. Garg, X. S. Zhou, and T. S. Huang. Classification approach towards ranking and sorting problems. In *Proc. of the 14th European Conference on Machine Learning*, 2003.

[12] C. Rudin, C. Cortes, M. Mohri, and R. E. Schapire. Margin-based ranking meets boosting in the middle. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.