Learning to Rank Using 1-norm Regularization and Convex Hull Reduction

Xiaofei Nan
Department of Computer and Information Science
University of Mississippi
University, MS 38677
xnan@olemiss.edu

Yixin Chen
Department of Computer and Information Science
University of Mississippi
University, MS 38677
ychen@cs.olemiss.edu

Xin Dang
Department of Mathematics
University of Mississippi
University, MS 38677
xdang@olemiss.edu

Dawn Wilkins
Department of Computer and Information Science
University of Mississippi
University, MS 38677
dwilkins@cs.olemiss.edu

ABSTRACT

The ranking problem appears in many areas of study such as customer rating, social science, economics, and information retrieval. Ranking can be formulated as a classification problem when pair-wise data is considered. However this approach increases the problem complexity from linear to quadratic in terms of sample size. We present in this paper a convex hull reduction method to reduce this impact. We also propose a 1-norm regularization approach to simultaneously find a linear ranking function and to perform feature subset selection. The proposed method is formulated as a linear program. We present experimental results on artificial data and two real data sets, concrete compressive strength data set and Abalone data set.

1. INTRODUCTION

Most inductive learning work has concentrated on classification and regression. However, there are many applications that reside in between the two: it is desirable to order objects or determine preferences rather than to classify instances (classification) or to predict ordinal utility values (regression). Ranking problems arise from such applications. For instance, it is common to rank document search results according to their relevance to a query as "perfect match", "relevant" and "non-relevant". A movie recommendation system gives users a list of recommended films where the most potentially wanted films are placed on the top.

One advantage of formulating an application as a ranking problem is that preference judgments may be much easier to obtain than the labels required for classification learning and the values of the dependent variable in regression [6]. For one example, in the design of the above movie recommendation system, a user might easily express his or her preference of movie A over movie B to generate a training data set. But it is harder to quantify how much he or she likes these two movies. Therefore, learning to rank is a natural choice for many applications in social science, mathematical economics, and information where human preferences play an important role.

Ranking problems share some properties with both classification and regression problems. These problems can be generalized as [11]: Given an iid sample (X, Y) where $X \in \mathbb{X}$ and $Y \in \mathbb{Y}$, and a set $\mathbb{F}$ of mappings $f\colon \mathbb{X} \to \mathbb{Y}$, a learning algorithm is to find an optimal mapping function $f^\star$ so that a predefined loss function is minimized. Normally, the loss function consists of the sum of an empirical risk which measures the training error and a measure that characterizes the capacity of $\mathbb{F}$. If $\mathbb{Y}$ is a finite unordered set, the problem is commonly referred to as a classification problem. Especially, it is a binary classification task when $\mathbb{Y} = \{-1, 1\}$. Because $\mathbb{Y}$ is unordered, the $0-1$ loss is often used as the loss function. It counts the number of cases that the estimate label does not match the given label. On the other hand, if $\mathbb{Y}$ is the set of real numbers, the task is usually referred to as regression. Because $\mathbb{Y}$ is in a metric space, the loss function should involve the corresponding metric information. In ranking tasks, $\mathbb{Y}$ is finite as in classification and has an ordering over these finite elements as in regression. In this sense, ranking lies in between classification and regression.

However, in contrast with regression, $\mathbb{Y}$ in ranking does not have a complete metric structure, in particular, the concept of distance is not available in $\mathbb{Y}$. It hence brings challenges in defining a ranking loss function. A ranking loss function should capture the intuition that it is more expensive to make mistakes such as placing "perfect match" to

"non-relevant" than to "relevant", yet it is difficult to define the distance of two labels y and y'. One approach to tackle this dilemma is to cast a ranking problem as a regression problem that imposes a metric on top of the set of rankings. Balcan et al. [3] proposed and analyzed reductions from ranking to binary classification problems given the significant efforts placed on developing classification algorithms. A commonly used method is to act on pairs of observations, and the loss function measures the probability of misclassification of a randomly drawn pair (x_1, x_2), where the two classes are x_1 being preferred over x_2 and the opposite respectively [11]. However the biggest concern about this method is that it increases the computational complexity from linear to quadratic in terms of the number of samples. In this paper, we incorporate the idea of convex hull reduction to decrease the computational complexity of learning to rank.

2. PREVIOUS WORK

The procedure of robust reduction from ranking to classification was summarized in [3] where the first step is to minimize the number of *inconsistencies* (cases where a less preferred instance is ranked higher than a more preferred instance) and then find the ordering that agrees best with the estimated preferences. Cohen et al. [6] proved that the latter procedure is NP-complete and described a simple greedy algorithm to rank observations based on how many other observations are ranked lower. They also made use of Hedge algorithm to find a good linear combination of multiple "weak" ranking "experts". Herbrich et al. [11] cast the ranking problem as a variation of classification, called ordinal regression, which modeled ranks as intervals on the real line and considered the loss functions based on pairs of observations. The margins of neighbor ranking boundaries had crucial impacts on the performance. Shashua and Levin [13] discussed this large margin principle and introduced two main approaches, fixed margin and sum of margins. The former strategy maximized the margin of the closest neighboring ranks, while the latter allows different margins for different observations and maximizes the sum of margins. Agarwal and Roth [1] stated the large margin principle theoretically and discussed the learnability of bipartite ranking functions. It is shown that the sufficient and necessary conditions of the learnability of a class of ranking functions are related with its rank-shatter coefficients in the same way as the VC-dimension related shatter coefficients for classification functions.

Instead of using SVM-like large margin methods, Freund et al. [8] combined preferences based on the boosting approach. Rudin et al. [12] modified Freund's work using a method analogous to Approximate Coordinate Ascent Boosting and proved that the algorithm makes progress with respect to the ranking margin at each iteration and converges to a maximum margin solution.

Crammer and Singer [7] projected observations on a real line as in [11]. Their online learning algorithm used one example at a time instead of using the pairwise data as in [11]. The loss function was defined as the sum of differences between the estimated ranks and real rank values. Burges et al. [5] built their loss function based on the cross entropy cost and utilized a neural network to solve this optimization problem. All these existing approaches implicitly operate on a non-calibrated utility scale which restricts the expressive power of these methods. Brinker and Hüllermeier [4] extended the conventional ranking framework with a natural zero point.

Clearly, using pairs of observations and casting ranking problems as classifications are a commonly used strategy. However, the major limitations are the high computational complexity and the lack of robustness to noise. The complexity issue is due to the usage of pairs of data. This problem is tackled in our paper by only selecting a small number of representative observations for each rank label, which dramatically reduces the size of the training data. Moreover, in many high dimensional applications, some attributes have little contribution to the ranking function, and these inconsequential attributes are removed automatically by the means of 1-norm SVM learning.

The remainder of the paper is organized as follows. In Section 3, we briefly introduce the concept of Kendall τ rank correlation coefficient as the performance measure and present the formulation of our ranking problem. We review the 1-norm SVM and convex hull construction in Section 4 and propose the ranking algorithm that combines convex hull reduction and 1-norm SVM. In Section 5, we present details about the experiments performed and the results obtained. We conclude in Section 6 with a discussion on future directions.

3. PRELIMINARIES

Performance metrics are fundamental in assessing any learning method. We first discuss a way to quantify the performance of ranking algorithms. We then describe the formulation of the ranking problem used in this work.

3.1 Kendall τ Correlation Coefficient

Fung et al. [9] used generalized Wilcoxon-Mann-Whitney Statistics to measure the probability of any pair of data being ordered correctly. A similar but simpler concept is Kendall τ rank correlation coefficient (simply the Kendall τ), which is a non-parametric statistic used to measure the degree of correspondence between two rankings. It assesses the significance of this correspondence. Considering the occurrence of ties, we use Kendall τ-b variation.

Let a_i and b_i be the rank of observation x_i given by two ranking algorithms, Kendall τ-b rank correlation coefficient for the two ranking algorithms on the given set of observations is defined as

$$\tau = \frac{\sum_{i<j} sgn(a_i - a_j) sgn(b_i - b_j)}{\sqrt{(T_0 - T_1)(T_0 - T_2)}}$$

where $T_0 = n(n-1)/2$, $T_1 = \sum_k t_k(t_k - 1)/2$, and $T_2 = \sum_l u_l(u_l - 1)/2$. The t_k is the number of tied a values in the k-th group, u_l is the number of tied b values in the l-th group, n is the number of observations.

If the two rankings are identical, the Kendall τ-b value is equal to 1. If two rankings are totally opposite, Kendall τ-b equals negative one. It approaches zero when the two rankings are irrelevant.

3.2 A Ranking Problem Formulation

In this work the ranking problem is formulated as below. Given observation set $\mathbb{X} \in \mathbb{R}^n$, let $\mathcal{R}$ be a relation on $\mathbb{X} \times \mathbb{X}$ that for any $(x_i, x_j) \in \mathcal{R}$, x_i is ranked higher than x_j. We are interested in learning a ranking function f to capture this relation, i.e.,

$$f(x_i) > f(x_j), \forall (x_i, x_j) \in \mathcal{R}.$$

If we consider the family of linear ranking functions, $f(x) = w^T x + b$, the given relation $\mathcal{R}$ is linear rankable (similar to linear separable in binary classification) if and only if

$$w^T x_i + b > w^T x_j + b, \forall (x_i, x_j) \in \mathcal{R},$$

which is equivalent to

$$w^T (x_i - x_j) > 0, \forall (x_i, x_j) \in \mathcal{R}.$$

If $\mathcal{R}$ is linear rankable, there exists an n-vector w such that

$$w^T (x_i - x_j) \geq 1, \forall (x_i, x_j) \in \mathcal{R}.$$

For the non-rankable cases, slack variables could be introduced into the model, i.e.,

$$w^T (x_i - x_j) \geq 1 - \eta_{i,j}, \eta_{i,j} \geq 0, \forall (x_i, x_j) \in \mathcal{R}. \quad (1)$$

Therefore, the ranking problem has been cast as a classification problem where the loss function acts on pairs of preferences. In order to have some indication that the learning algorithm will generalize well, the loss function should involve both the complexity of the ranking function class and the empirical error. This complexity can be achieved by an informative quantity, for example, VC dimension or an upper bound on VC dimension.

However, it is clear that the number of constraints grows quickly as the size of observation increases, roughly quadratic in the number of training samples, which poses a significant computational burden and even makes the solution infeasible.

4. RANKING WITH 1-NORM REGULARIZATION AND CONVEX HULL REDUCTION

Our goal is to decrease the number of constraints that are inherent with pairwise data without compromising the ranking relationships. The basic idea is to select a subset of representative preference constraints instead of using all preference pairs.

4.1 Convex Hull Reduction

When there are multiple ranking labels $\{y_1, y_2, \ldots, y_k\}$, let Y_s be the set of all observations with rank label y_s. A preference relation is specified as

$$\mathcal{R} = \{(x_i, x_j) : \forall i, j,\ x_i \in Y_s, x_j \in Y_t, Y_s \prec Y_t\},$$

where $\prec$ denotes "being preferred to".

The size of $\mathcal{R}$ is $|\mathcal{R}| = \sum_{s,t,Y_s \prec Y_t} |Y_s||Y_t|$. For a ranking problem in Section 3.2, there will be $|\mathcal{R}|$ constraints of form (1). For almost all ranking problems, $|\mathcal{R}|$ is much larger than the number of ranking labels. In fact, the size of constraints is approximately the quadratic of the observation size, i.e., the number of constraints is $\mathcal{O}(\ell^2)$ where ℓ is the number of training observations. Therefore, the number of constraints is prohibitively large even for a training set of medium size, say 1000.

Our approach to reduce the number of constraints is to select a small subset of representative preference constraints without compromising the ranking performance significantly. These representative constraints are constructed using observations lying on the convex hulls of Y_s $(s = 1, \ldots, k)$.

A convex hull is the smallest convex polygon containing all the points in a set. It captures the shape of a data set. The convex hull computation could be finished in $\mathcal{O}(|Y|lg(|Y|))$ time for each observations set Y. After finding convex hulls for each set $Y_s (s = 1, \ldots, k)$, constraints in (1) are reduced to

$$w^T (x_i - x_j) \geq 1 - \eta_{i,j}, \eta_{i,j} \geq 0, \forall (x_i, x_j)$$
$$\text{that } x_i \in CH(Y_s), x_j \in CH(Y_t), \text{and } Y_s \prec Y_t,$$

where $CH(Y_s)$ denotes the observations on the convex hull of the set Y_s.

After convex hull reduction, the number of constraints shrinks to $\mathcal{O}(\sum_{s,t,Y_s \prec Y_t} |CH(Y_s)||CH(Y_t)|)$, where $|CH(Y_s)|$ is the size of the convex hull of set Y_s. We will show later that in most applications this procedure will reduce the constraint size dramatically.

4.2 1-norm Support Vector Machines

Support Vector Machines (SVMs) are a supervised learning technique based on statistical learning theory. The standard two-norm Support Vector Machines have been proven effective in learning classification, regression and ranking functions.

In the cases of classification, a SVM is formulated as the solution of a quadratic program. Given a set of training data $(x_1, y_1), \ldots, (x_l, y_l)$, where the input $x_i \in \mathbb{R}^n$, and the output $y_i \in \{1, -1\}$ has binary values, the standard SVM with linear kernel is found by solving the following quadratic program:

$$\begin{array}{rl} \min_{w,b,\eta} & \lambda \|w\|_2^2 + \sum_{i=1}^{\ell} \eta_i \\ s.t. & y_i (w^T x_i + b) + \eta_i \geq 1 \\ & \eta_i \geq 0, i = 1, \ldots, \ell \end{array} \quad (2)$$

where $\lambda > 0$ is a fixed penalty parameter that specifies the tradeoff between empirical misclassification error and the complexity of the classifier. Zhu et al. [14] argued that the 1-norm SVM may have some advantage over the standard 2-norm SVM, especially when there are redundant noise features. Similarly, 1-norm SVM can be formulated by replacing the $\|\cdot\|_2$ regularization operator in (2) with $\|\cdot\|_1$ regularization. Hence we have the following optimization problem:

$$\begin{array}{rl} \min_{w,b,\eta} & \lambda \|w\|_1 + \sum_{i=1}^{\ell} \eta_i \\ s.t. & y_i (w^T x_i + b)_T + \eta_i \geq 1 \\ & \eta_i \geq 0, i = 1, \ldots, \ell \end{array} \quad (3)$$

where $\|w\|_1 = \sum_{j=1}^{n} \|w_i\|$.

The optimization problem (3) can be formed as a linear

program. We rewrite $w_j = u_j - v_j$ where $u_j, v_j \geq 0$. If either u_j or v_j has to equal zero, then $\mid w_j \mid = u_j + v_j$. Then (3) becomes

$$\begin{array}{rl} \min_{u,v,b,\eta} & \lambda \sum_{j=1}^{n}(u_j + v_j) + \sum_{i=1}^{\ell} \eta_i \\ s.t. & y_i[(u-v)^T x_i + b] + \eta_i \geq 1 \\ & u_j, v_j \geq 0, j = 1, \ldots, n \\ & \eta_i \geq 0, i = 1, \ldots, \ell. \end{array} \quad (4)$$

Solving (4) yields solutions equivalent to those obtained by (3) because any optimal solution to (4) has at least one of the two variables u_j, v_j equal to 0 for all $j = 1, \ldots, n$. Otherwise, assume $u_j > v_j > 0$ without loss of generality, and we can find a better solution by setting $u_j = u_j - v_j$ and $v_j = 0$, which contradicts to the optimality of (u, v).

The popularity of standard two-norm SVM is based on the concept of kernel, which may create non-linear classifiers by transforming the original input space to high dimensional feature space though the transformation functions are unknown. One of the kernel trick's sufficient conditions is the existence of dot product. Although the kernel trick does not apply to 1-norm SVM, it has a major advantage over the 2-norm SVM: 1-norm regularization favors sparse solution hence provides an embedded feature subset selection capability [14]. Learning accuracy does not always go up along with an increase in the number of features. In many applications for which data sets with tens or hundreds of thousands variables are available, it is important but difficult to find a subset of features that maximizes the learning accuracy. This problem is known as feature selection. The related literature can be found in [10]. 1-norm regularization is especially suitable for data sets that have redundant and/or irrelevant features.

4.3 Ranking with 1-norm Regularization

Combing the results of (2) and (4), the 1-norm ranking problems could be regularized as

$$\begin{array}{rl} \min_{u,v,b,\eta} & \lambda \sum_{j=1}^{n}(u_j + v_j) + \sum_{i=1}^{\ell} \eta_{i,j} \\ s.t. & w^T(x_i - x_j) \geq 1 - \eta_{i,j}, \eta_{i,j} \geq 0 \\ & x_i \in CH(Y_s), x_j \in CH(Y_t), Y_s \prec Y_t, \\ & u_j, v_j \geq 0, j = 1, \ldots, n \\ & \eta_{i,j} \geq 0, i = 1, \ldots, \ell. \end{array}$$

Slack variables are necessary in the non-rankable applications. But the above formula shows that the size of slack variables is equal to the number of convex hull point pairs, which is another computational burden, especially when the sample points are mostly distributed on the surface of the high dimensional cube and convex hull reduction cannot relieve the computational load a lot.

To relieve the computational load, we not only reduce the number of constraints but also reduce the number of variables in the optimization problem. To fulfill the second purpose, we assign each convex hull point a slack variable. Therefore, the original slack variable for each pair (x_i, x_j) can be represented by the addition of the slack variables of x_i and x_j. And the number of slack variables is dramatically reduced from $|\mathcal{R}| = \sum_{s,t,Y_s \prec Y_t} |Y_s||Y_t|$ to $\sum_s |Y_s|$.

As a result, the ranking function is specified as the solution of the following linear program:

$$\begin{array}{rl} \min_{u,v,b,\eta} & \lambda \sum_{j=1}^{n}(u_j + v_j) + \sum_{s,t}(\eta_s + \eta_t) \\ s.t. & (u-v)^T(x_s - x_t) + (\eta_s + \eta_t) \geq 1 \\ & x_s \in CH(Y_s), x_t \in CH(Y_t), Y_s \prec Y_t \\ & u_j, v_j \geq 0, j = 1, \ldots, n \\ & \eta_s, \eta_t \geq 0, s, t = 1, \ldots, n. \end{array} \quad (5)$$

Moreover, we could further reduce the number of constraints using the transitive property of the preference over ranking labels. For example, we have preference relation denoting as $Y_s \prec Y_t \prec Y_u$. Instead of considering all the preference constraints in the relation $\mathcal{R}_1 = \{(Y_s, Y_t), (Y_s, Y_u), (Y_t, Y_u)\}$, relations $\mathcal{R}_2 = \{(Y_s, Y_t), (Y_t, Y_u)\}$ is sufficient to express $\mathcal{R}_1$ in the sense that the transitive closure of $\mathcal{R}_2$ is $\mathcal{R}_1$. Therefore, the number of constraints after reduction is bounded by $\mathcal{O}(\sum_{s,Y_s \prec Y_{s+1}} |CH(Y_s)||CH(Y_{s+1})|)$.

5. EXPERIMENTS AND RESULTS

We tested our approach on three data sets: an artificial data set, a concrete compressive strength data set, and an Abalone data set. The artificial data set consists of 2000 samples in two size-balanced groups with dimension 6. Three features out of six are generated according to the uniform distribution on two regions with little overlapping so that the two groups are linear separable. The other three features are random noises with little discriminative power.

The two real data sets are publicly available from UCI machine learning repository [1]. They were used as benchmark data sets for ordinal regression methods in the literature. The dependent variable of the concrete strength data set is continuous. In the Abalone data set, the dependent variable is discrete taking integer values between 1 and 29. The concrete compressive strength data set has 1030 instances and 8 attributes. We discretize the values of continuous dependent variable into five bins of equal size. Observations in the same bin share the same ranking label. Although the dependent variable of the Abalone data set is discrete, considering the fact that most applications do not have such a large number of ranking labels as 29, we grouped the observations into three rank classes. All observations whose dependent variable is smaller than 6 are assigned a ranking label 1. All observations whose dependent variable is greater than 15 are assigned a ranking label 3. All the remaining observations belong to rank class 2. The Abalone data set has 4177 instances and 8 attributes.

We used 5-fold cross validation in all data sets. On the artificial data set, our approach showed perfect ranking performance with Kendall τ coefficient being 1. Moreover, the weights for the three noise features were all 0, which implies that the noise features were rejected by the ranking algorithm. This confirms that 1-norm regularization favors sparse solution and is able to select the most discriminative features under a proper value of the λ parameter.

Similar observations were obtained on the other two data sets. The top plots in Figure 1 and Figure 2 show the feature selection process as the λ parameter in (5) varies. The

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

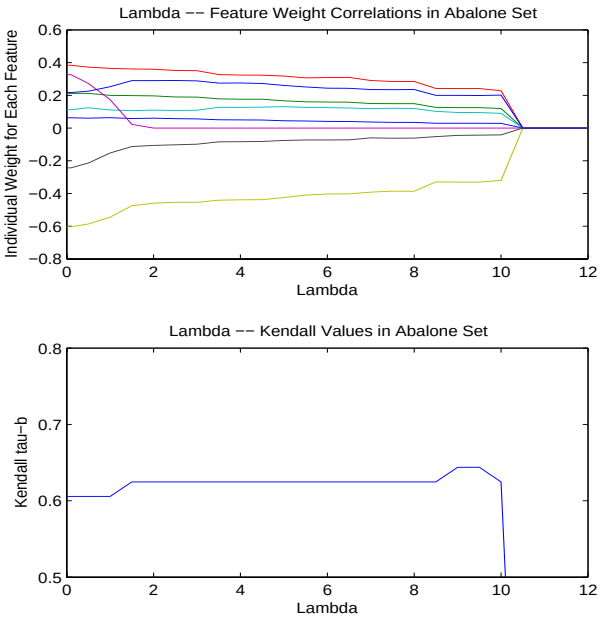

Figure 1: The relationship between λ, feature weights (w_j's), and Kendall τ-b rank correlation coefficient on the Abalone data set.

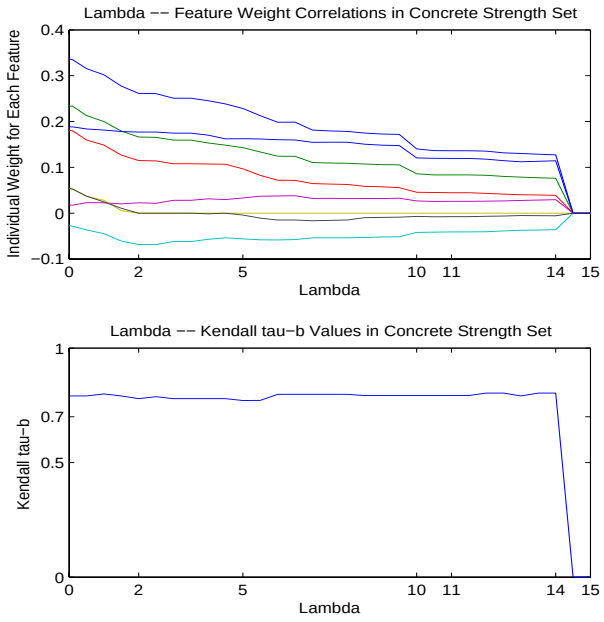

Figure 2: The relationship between λ, feature weights (w_j's), and Kendall τ-b rank correlation coefficient on the Concrete Strength data set.

horizontal axis indicates the value of λ. The vertical axis represents the values of the weights. We can see from both figures that as the value of λ increases the absolute values of feature weights decrease. Eventually all the feature weights drop to zero. However, some feature weights decrease more quickly than others. Thus the non-zero-weight features are selected. The bottom plots in Figure 1 and Figure 2 present the ranking performance as the λ parameter changes. The vertical axis represents the Kendall τ-b value. In the concrete strength data set, 2 feature weights out of 8 drop to zero after λ is equal to 2, which indicates that these two features are "useless" compared with others. The best Kendall τ result, 0.80325, is achieved when λ ranges from 12 to 14. Similar scenario happens in Abalone set. One feature weight drops to zero when λ is greater than 2. It hits the best performance 0.643775 when λ is within the range between 9 and 9.5.

Next, we look at the effect of convex hull reduction on the number of constraints. In our experiment, the preference pairs are specified as between group observations. We use Quick Hull [2] as the convex hull algorithm. The numbers of preference constraints before and after convex hull reduction are given in Table 1. Because 5-fold cross validation is used, we average the number from each run and provide the standard deviation as well. The number of constraint for concrete strength data set is reduced by 34.16%. For the artificial data set and Abalone data set, the reduction ratios are 72.69% and 83.90%, respectively. Although the reduction ratio differs in different data sets, convex hull seems to be an effective option in reducing the constraint set.

[2] `http://www.qhull.org/`

Theoretically predicting the number of observations on a convex hull is not an easy task because it depends on the number of observations, the dimension of feature space, and the distribution of the observations. Hueter [15] developed a central limit theorem for the convex hull size in high dimensional spaces. To further investigate how well convex hull reduction works on the ranking problems, we randomly generate 1000 observations with feature dimension varies from 2 to 9, and record the number of observations on the convex hull. Table 2 shows that the convex hull size increases with the feature dimension. In a 9-dimensional feature space, the number of observations on the convex hull is close to the size of the whole data set. This is an empirical validation of the curse of dimensionality, which limits the use of convex hull reduction in applications with high feature dimension.

6. CONCLUSIONS

We proposed a ranking algorithm on pair-wise preference constraints. A convex hull reduction algorithm is used to reduce the number of constraints. A linear ranking function is computed by minimizing the sum of ranking error and a 1-norm regularization term. The 1-norm regularization favors sparse solutions, hence selects a subset of features. Experimental results demonstrate good performance on the data sets tested.

Due to the curse of dimensionality, in a high dimensional space, almost all observations lie on the surface of a convex hull. Therefore we don't expect significant improvements from the proposed ranking algorithm when the feature dimension is large (10 or higher).

7. ACKNOWLEDGMENTS

Table 1: Number of constraints before and after convex hull reduction. The reduction ratio is the percentage of the constraints that are removed.

Data Set	Before Convex Hull Reduction	After Convex Hull Reduction	Reduction ratio
Artificial	640000	174816 ± 2670	72.69%
Concrete Strength	271590 ± 466	178822 ± 4440	34.16%
Abalone	1105700 ± 2556	178060 ± 8833	83.90%

Table 2: Convex hull reduction on uniform and normal distributions. There are 1000 observations before the reduction. The right two columns contain the number of observations that are on the convex hull.

Dimension of Feature Space	Normal Distribution	Uniform Distribution
2 Dimension	13	19
3 Dimension	32	72
4 Dimension	85	152
5 Dimension	159	306
6 Dimension	274	496
7 Dimension	358	661
8 Dimension	527	759
9 Dimension	637	852

Xiaofei Nan, Yixin Chen, and Dawn Wilkins were supported in part by the US National Science Foundation under award number EPS-0903787. Xin Dang was supported by the US National Science Foundation under award number DMS-0707074. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

8. REFERENCES

[1] S. Agarwal and D. Roth. Learnability of Bipartite Ranking Functions In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 16–31, 2005.

[2] G. Anthony, H. Ruther. Comparison of Feature Selection Techniques for SVM Classification. In *10th International Symposium on Physical Measurements and Signatures in Remote Sensing*, poster 11, 6 pages, 2007.

[3] M. Balcan, N. Bansal, A. Beygelzimer, D. Coppersmith, J. Langford, G.B. Snorkin. Robust Reductions from Ranking to Classification. In *Machine Learning*, 72(1-2), pages 139–153, 2008.

[4] K. Brinker, E. Hüllermeier. Calibrated label-Ranking. In *Proceedings of the NIPS 2005 Workshop on Learning to Rank*, pages 1–6, 2005.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender. Learning to Rank using Gradient Descent. In *Proceedings of the 22nd international conference on machine Learning*, pages 89–196, 2005.

[6] W. W. Cohen, R. E. Schapire, Y. Singer. Learning to Order Things. In *Journal of Artificial Intelligence Research*, 10(1), pages 243–270, 1999.

[7] K. Crammer and Y. Singer. Pranking with Ranking. In *Proceedings of the Neural Information Processing Systems Conference*, pages 641–647,2001.

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

[9] G. Fung, R. Rosales, B. Krishnapuram. Learning Rankings via Convex Hull Separation. In *Advances in Neural Information Processing Systems*, 18, pages 395–402, 2006.

[10] I. Guyon, A. Elisseeff. An Introduction to Variable and Feature Selection. In *Journal of Machine Learning Research*, Vol.3, pages 1157–1182, 2003.

[11] R. Herbrich, T. Graepel, K. Obermayer. Large Margin Rank Boundaries for Ordinal Regression. In *Advances in Large margin Classifiers*, pages 115–132, 2000, Cambridge, MA, MIT Press.

[12] C. Rudin, C. Cortes, M. Mohri, R. E. Schapire. Margin-Based Ranking Meets Boosting in the Middle. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 63–78, 2005.

[13] A. Shashua, A. Levin. Ranking with large Margin Principle: Two Approaches. In *Advances in Neural Information Processing Systems*, 15, pages 961–968, 2003.

[14] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani. 1-norm Support Vector Machines. In *Neural Information Processing Systems*, 16, 2003.

[15] I. Hueter. Limit Theorems for the Convex Hull of Random Points in Higher Dimensions. In *Transactions of American Mathematical Society*, 351(11), pages 4337–4363, 1999.