Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

Ball Ranking Machines for Content-Based Multimedia Retrieval

Dijun Luo, Heng Huang
The University of Texas at Arlington, Arlington, Texas, USA
dijun.luo@gmail.com, heng@uta.edu

Abstract

In this paper, we propose the new Ball Ranking Machines (BRMs) to address the supervised ranking problems. In previous work, supervised ranking methods have been successfully applied in various information retrieval tasks. Among these methodologies, the Ranking Support Vector Machines (Rank SVMs) are well investigated. However, one major fact limiting their applications is that Ranking SVMs need optimize a margin-based objective function over all possible document pairs within all queries on the training set. In consequence, Ranking SVMs need select a large number of support vectors among a huge number of support vector candidates. This paper introduces a new model of of Ranking SVMs and develops an efficient approximation algorithm, which decreases the training time and generates much fewer support vectors. Empirical studies on synthetic data and content-based image/video retrieval data show that our method is comparable to Ranking SVMs in accuracy, but use much fewer ranking support vectors and significantly less training time.

1 Introduction

In history of Support Vector Machines (SVMs) research, many original models and variants have been well studied and have attracted impacts from various areas, including classification [Vapnik, 1995], clustering [Xu and Schuurmans, 2005], and ranking [Herbrich *et al.*, 1999]. In this paper, we focus on ranking which is central issue of many applications, such as content based image/video retrieval, collaborative filtering, expert search, anti web spam, sentiment analysis, *etc*. However, the volume of multimedia content and user data is increasing in exponential pattern on the internet. For example, there have been over 500 million users on *Facebook.com*, spending over 700 billion minutes per month, generating over 900 million objects that people interact with (pages, groups, events and community pages) and huge amount of other content in various formats [1]. How to efficiently perform machine learning tasks on such heavily accumulated data is a challenging problem. In this paper, we try to take an ambitious step on the problem of supervised ranking with large scale content-based multimedia retrieval.

Supervised ranking, taking image (or key frames of videos) retrieval for example, is a task as follows. Assume that there is a collection of images. In retrieval (*i.e.*, ranking), given a query, the **ranking function** assigns a score to each image in a database, and ranks the images in descending order of the scores. The ranking order represents the relevance of images with respect to the query. In learning, a number of queries are provided; each query is associated with a ranking list of images; a ranking function is then created using the training data, such that the model can precisely predict the ranking lists in the training data. Due to its importance, supervised ranking has been drawing broad attention in the multimedia community recently. Several methods based on what we call the pairwise approach have been developed and successfully applied to image/video retrieval. This approach takes data pairs as instances in learning, and formalizes the problem of supervised ranking as that of classification. Specifically, in learning it collects data pairs from the ranking lists, and for each data pair it assigns a label representing the relative relevance of the two data points. It then trains a classification model with the labeled data and makes use of the classification model in ranking. The uses of Support Vector Machines (SVMs), Boosting, and Neural Network as the classification model lead to the methods of Ranking SVMs [Herbrich *et al.*, 1999], RankBoost [Freund *et al.*, 2003], and RankNet [Burges *et al.*, 2005]. In this paper, we focus on Ranking SVMs.

There are several advantages of the pairwise approach. First, existing methodologies on classification can be directly applied. Second, the training instances of data pairs can be easily obtained in certain scenarios [Joachims, 2002], and third, the models are much reliable and robust since the pairwise relationships bring rich redundant information (even some of the pairs are ranked incorrectly, the final results often remain corrected by the other pairs). However, the pairwise approach also limits the application of Ranking SVMs in the sense that it generates (1) a huge number of possible pairs and (2) a large number of support vectors remain in the ranking models. For example, if there are 1000 referent images and 1000 irreverent images within a single query, there are

[1] http://www.facebook.com/press/info.php?statistics

1000,000 pairs which need to be considered in the training process of the ranking machine. Such a huge number of pairs also make the trained ranking model very complex which hinders Ranking SVMs from responding an online query in a reasonable time since the number of support vectors is a critical element in the prediction of SVMs.

To tackle this challenge, this paper presents a novel objective function of Ranking SVMs and develop an efficient algorithm to optimize the presented objective function, according to the idea of core set approximation of Support Vector Machines. The dual form or our Ranking Machines (Ball Ranking Machines) can be formalized as Minimum Enclosing Ball (MEB) problem. Our efficient raking machine algorithm is much less expensive in computation complexity and generates much fewer support vectors. Empirical studies on both synthetic data, content-based image/video retrieval data show that our method is comparable with Ranking SVMs in accuracy, but uses much fewer ranking support vectors and significantly less training time.

One should notice that in [Tsang *et al.*, 2007], the authors also released the options for ranking their source codes, but the ranking is just a variant of regression, and not based on the pairwise approach and our approach is totally different from theirs.

2 Ranking SVMs

In this section, we first begin our discussion with a brief introduction of the ranking SVMs and related work. And in next section we will present our version of the ranking SVM model and develop a novel algorithm to solve the related optimization problems.

In classification problems, the SVMs [Vapnik, 1995] are considered as one of the state-of-the-art approaches which offers relatively robust and accurate results among all well-known algorithms. Ranking SVM is a variant of SVM which addresses the supervised ranking problem [Joachims, 2002]. In Ranking SVMs, given a query q the document d are ranked using linear **ranking function** as,

$$F_w(q, d) = w^T \mathbf{f}(q, d) \qquad (1)$$

where $\mathbf{f}(q, d)$ is a feature vector which describes the matching between query q and document d such as in the description-oriented retrieval approach of Fuhr *et al.* [Fuhr, 1989], and w is the model parameters which needs to be determined by learning. As a pairwise method, the ranking SVM aims to satisfy:

$$F_w(q_k, d_i) > F_w(q_k, d_j), \forall (q_k, d_i, d_j) \in \Omega \qquad (2)$$

where Ω is defined as $\Omega = \{(q, d_i, d_j)$: document d_i is more relevant to query q than d_j $\}$. This suggests that in the ranking model, we want to use $F_w(q, d)$ to represent how much the document matches the query.

As a maximum margin method, the model of ranking SVMs is represented by the following optimization problem:

$$\begin{aligned} \min_w \quad & \frac{1}{2}\|w\|^2 + C\xi_\omega & (3) \\ \text{s.t.} \quad & F_w(q_k, d_i) - F_w(q_k, d_j) \geq 1 - \xi_\omega, & \\ & \forall \omega = (q_k, d_i, d_j) \in \Omega & \end{aligned}$$

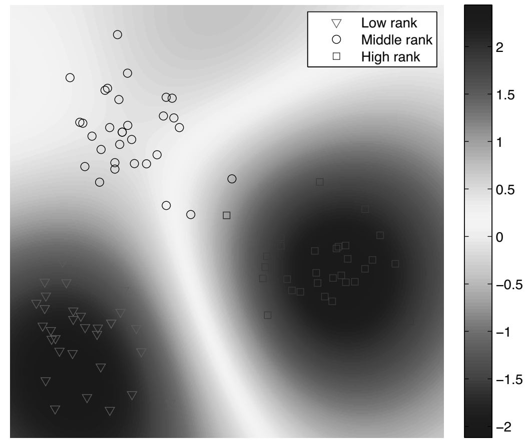

Figure 1: Ranking function by Gaussian kernel Ranking SVM.

In the rest of the paper, ω denotes the index of all the triples $(q_k, d_i, d_j) \in \Omega$.

Ranking SVMs enjoy the following properties: (1) The optimization problem is convex, thus we are guaranteed to obtain a unique global solution. (2) Kernel trick is allowed such that non-linear ranking can be obtained.

Fig. 1 shows a ranking function on a toy dataset using Gaussian kernel ranking SVM. In this cases, we have three levels of relevance for queries: high, middle, and low. From the figure, we can see that the non-linear ranking function ranks the objects in reasonable orders.

Notice that here the size of Ω is usually very large, since that for each query q, we have to consider all the possible pairs of d_i, d_j in all relevance levels. Unfortunately, the worse case of the training complexity of ranking SVMs is $|\Omega|^3$, and in practical applications, the complexity is approximately $|\Omega|^{2.3}$. This difficulty makes ranking SVMs almost impractical in real world applications.

3 Ball Ranking Machine

In this section, we present a new model of ranking SVMs which is a variant of the model defined in Eq. (3). The reason why we modify Eq. (3) is that after our modification, the dual problem is equivalent to a well studied problem (called Minimal Enclosing Ball problem), which can be solved very efficiently. Instead of solving the prime problem of Eq. (3) or its dual, we solve the corresponding Minimal Enclosing Ball problem.

3.1 Ranking Model

We propose a new version of Ranking SVMs, named Ball Ranking Machine, which is guaranteed to generate sparse solution in the dual form problem. Instead of solving Problem (3), we solve

$$\begin{aligned} \min_w \quad & \frac{1}{2}\|w\|^2 - \rho + C\xi_\omega^2 & (4) \\ \text{s.t.} \quad & F_w(q_k, d_i) - F_w(q_k, p_j) \geq \rho - \xi_\omega, & \\ & \forall \omega = (q_k, d_i, d_j) \in \Omega & \end{aligned}$$

The corresponding dual is,

$$\max_{\alpha} \quad -\alpha^T H\alpha, \qquad (5)$$
$$\text{s.t.} \quad \alpha^T\mathbf{1} = 1, \alpha \geq 0,$$

where

$$H_{\omega\omega'} = (\mathbf{f}(q_k, d_i) - \mathbf{f}(q_k, d_j))^T \left(\mathbf{f}(q'_k, d'_i) - \mathbf{f}(q'_k, d'_j)\right) + \frac{\delta(\omega, \omega')}{C}, \qquad (6)$$

$\omega = (q_k, d_i, d_j) \in \Omega, \omega' = (q'_k, d'_i, d'_j) \in \Omega$, and $\delta(\omega, \omega')$ is 1 if $\omega = \omega'$, 0 for other case.

Assume that α is the optimal solution of Eq. (5), then given a query q and a document d, the final ranking function is

$$\begin{aligned} F_w(q, d) &= w^T\mathbf{f}(q, d) \\ &= \sum_{\omega=1}^{|\Omega|} \alpha_\omega (\mathbf{f}(q_k, d_i) - \mathbf{f}(q_k, d_j))^T \mathbf{f}(q, d) \end{aligned}$$

Here we should notice that the size of the H is huge, even for some medium size training data. The total number of variables in H is $(Qn^2)^2 = Q^2n^4$, and the total number of optimization variables is Qn^2, which is almost impossible to handle. However, we also know that the solution of Eq. (5) should be very sparse, *i.e.* most of the values of α_i should be zeros, $i = 1, 2, \cdots, Qn^2$. Notice that typical SVM implementations have a training time complexity that scales is around $O(m^{2.3})$ where m is the number of dual variables [Platt, 1999]. The main purpose of this paper is to develop an efficient approach to optimize the problem in Eq. (5).

3.2 Optimization

The major reason one can avoid the large scale optimization is that most of the optimization variables are not involved in the final optimal solution, (simply because the corresponding dual variable is zero, which does not contribute to the objective function in Eq. (5)). However, for most optimization techniques, we can determine whether one variable is involved in the optimal solution only after we obtain the global solution. In contrast, our solution determines it at the very beginning stage and avoid the complex optimization procedure over all the optimization variables.

Given m data points in Euclidean spaces $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]$ and consider the following problem,

$$\min_{\mathbf{c}, r} \quad r^2,$$
$$\text{s.t.} \quad \|\mathbf{c} - \mathbf{x}_i\| \leq r, i = 1, 2, \cdots, m. \qquad (7)$$

One can easily check that the dual problem of Eq. (7) is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{m} \alpha_i \mathbf{x}_i^T\mathbf{x}_i - \sum_{i,j=1}^{m} \alpha_i\alpha_j \mathbf{x}_i^T\mathbf{x}_j, \\ = \quad & \alpha^T\mathbf{diag}K - \alpha^T K\alpha \\ \text{s.t.} \quad & \alpha^T\mathbf{1} = 1, \alpha \geq 0. \end{aligned} \qquad (8)$$

where

$$\mathbf{c} = \sum_i \alpha_i\mathbf{x}_i, r = \sqrt{\alpha^T\mathbf{diag}K - \alpha^T K\alpha}, \qquad (9)$$

and K is a $m \times m$ matrix such that $K_{ij} = \mathbf{x}_i^T\mathbf{x}_j$ and $\mathbf{diag}K = [K_{11}, K_{22}, \cdots, K_{mm}]^T$. By comparing Eq. (8) and Eq. (5), we see that they are almost identical.

In this paper, our main purpose is to convert problem Eq. (5) to problem Eq. (8) then equivalently solve Eq. (7) which can be solved very efficiently by approximation. To do so, we let

$$\mathbf{x}_\omega = \begin{pmatrix} \mathbf{f}(q_k, d_i) - \mathbf{f}(q_k, d_j) \\ \frac{\mathbf{e}_\omega}{\sqrt{C}} \end{pmatrix}, \qquad (10)$$

where $\mathbf{e}_\omega$ is an $|\Omega|$-dimensional column vector in which all elements are zeros except that the position at ω is 1. Then one can easily show that $K = H$ where K and H are defined as in Eq. (8) and Eq. (5), respectively. If $\alpha^T\mathbf{diag}K$ in Eq. (8) is a constant, respective to α, Eq. (8) and Eq. (5) are equivalent.

Now consider a kernel version of Problem (5) where

$$\begin{aligned} &H_{\omega\omega'} \\ =&k\left((\mathbf{f}(q_k, d_i) - \mathbf{f}(q_k, d_j), \mathbf{f}(q'_k, d'_i) - \mathbf{f}(q'_k, d'_j)\right) \\ &+ \frac{\delta(\omega, \omega')}{C}. \end{aligned} \qquad (11)$$

and $k(u, v)$ is a kernel function such that $k(u, v)$ is a constant if $u = v$. Fortunately this constraint is satisfied in most of the real world application [Tsang *et al.*, 2005]. A simple and widely used example is the Gaussian kernel: $k(u, v) = \exp(-\|u - v\|^2/\sigma^2)$. We define $\mathbf{x}_\omega$ as,

$$\mathbf{x}_\omega = \begin{pmatrix} \phi(q_k, d_i) - \phi(q_k, d_j) \\ \frac{\mathbf{e}_\omega}{\sqrt{C}} \end{pmatrix} \qquad (12)$$

where $\phi(\cdot)$ is the kernel mapping function which defines the kernel $k(\cdot, \cdot)$: $k(u, v) = \langle\phi(u), \phi(v)\rangle$. Then we have the following theorem,

Theorem 3.1 *With Eq. (12), problems of Eq. (8) and (5) are equivalent.*

3.3 Rank Function via MEB Solution

Eq. (7) is a minimum enclosing ball (MEB) problem, which have been studied well in literacy [Badoiu and Clarkson, 2008; Welzl, 1991; Fischer and Gartner, 2003; Fischer *et al.*, 2003]. In the previous discussion, the complexity analysis suggests that exact MEB solution is expensive and is not necessary. Thus here we employ an approximate solver of MEB problem which is similar with paper [Badoiu and Clarkson, 2008] or [Tsang *et al.*, 2005].

By employing Theorem 2.2 in paper [Badoiu and Clarkson, 2008], a MEB problem can be solved approximately by the following,

(0) Initiate a set of support set $\mathbb{S}_0$ and a center $\mathbf{c}_0, \mathbb{S} \leftarrow \mathbb{S}_0, \mathbf{c} \leftarrow \mathbf{c}_0$.

(1) Solve Eq. (8) on $\mathbb{S}$, let $(\mathbf{c}_\mathbb{S}, r)$ be the optimal solution.

(2) $\mathbf{c} \leftarrow \mathbf{c}_\mathbb{S}$, let x^∞ be the data point which is the furthest to $\mathbf{c}$,

(3) If all the data points are in the ball $(\mathbf{c}, (1 + \epsilon)r)$ then terminate the algorithm and output $\mathbf{c}$, else $\mathbb{S} = \{\mathbb{S}, x^\infty\}$, and go to (1).

In step (1), the $\mathbf{c}_\mathbb{S}$ is computed by $\mathbf{c}_\mathbb{S} = \sum_{\omega\in\mathbb{S}} \alpha_\omega\mathbf{x}_\omega$. In general case of kernel representation, $\mathbf{x}_i$ can not be explicitly computed. Yet, we can still compute the distance between $\mathbf{c}_{\mathbb{S}}$ and any data points $\mathbf{x}_{\omega'}$. Notice that $\mathbf{x}$ is defined in Eq. (12), thus for any $\mathbf{x}$,

$$\|\mathbf{x}\| = \sqrt{\|\phi_{ki} - \phi_{kj}\|^2 + \|\frac{\mathbf{e}_\omega}{\sqrt{C}}\|^2} = \sqrt{2 - 2\langle\phi_{ki}, \phi_{kj}\rangle + \frac{1}{C}},$$

where $\phi_{ki} = \phi(q_k, d_i)$ and $\phi_{ki} = \phi(q_k, d_j)$. Then

$$\begin{aligned} & \|\mathbf{c}_{\mathbb{S}} - \mathbf{x}_{\omega'}\|^2 \\ = \; & \|\mathbf{c}_{\mathbb{S}}\|^2 - 2\sum_{\omega}\langle\mathbf{x}_\omega, \mathbf{x}_{\omega'}\rangle + \|\mathbf{x}_{\omega'}\|^2 \\ = \; & \|\mathbf{c}_{\mathbb{S}}\|^2 - 2(\alpha H)_{\omega'} + 2 - 2\langle\phi_{ki}, \phi_{kj}\rangle + \frac{1}{C} \quad (13) \end{aligned}$$

where $\omega = (i, j, k)$, $\omega' = (i', j', k')$ and H is defined in Eq. (11). Now all the terms in Eq. (13) can be computed directly given the kernel function $k(\cdot, \cdot)$.

The output of the above algorithm is $\mathbf{c}^*$ and R^*, then $\mathbf{c}_{\mathbb{S}}$ is only

$$\mathbf{c}^* = \sum_{\omega=1}^{|\Omega|} \alpha_\omega \begin{pmatrix} \phi(q_k, d_i) - \phi(q_k, d_j) \\ \frac{\mathbf{e}_\omega}{\sqrt{C}} \end{pmatrix} \qquad (14)$$

and

$$R^* = \sqrt{\alpha^T \mathbf{diag} H - \alpha^T H \alpha}.$$

We summarize the Ball Ranking Machine in Algorithm 1.

Algorithm 1 Ball Ranking Machine Algorithm.

Input
All query-document pair $(q, d)_i, i = 1, 2, ...N$, feature map $\mathbf{f}$, kernel function k.
for each query q_k
  for each document-pair (d_i, d_j)
    compute $\mathbf{f}(q, d_j), \mathbf{f}(q_k, d_j)$.
  end for
end for
Construct problem in Eq. (8) using Eq. (12).
Find $\mathbf{c}$ and R by MEB solver.
Output
Ranking function $F(q, d) = \mathbf{c}^T \phi(q, d)$.

The solution of $\mathbf{c}$ is represented as a linear combination of a subset of the training triples (q_k, d_i, d_j). This subset is called as **core set**. This is similar with support vectors in SVMs. In our experiments, by support vectors we mean the triples in this core set. Suppose $C = [\omega_1, \cdots, \omega_s]$ is the core set where s is the size of the core set, and the corresponding combination coefficients are $\alpha_1, \cdots, \alpha_s$, then the final ranking function can be written as

$$F(q, d) = \sum_{\omega \in C} \alpha_\omega \langle\phi(q, d), \phi(q_k, d_i) - \phi(q_k, d_j)\rangle,$$

where ω is the index of the triple (q_k, d_i, d_j).

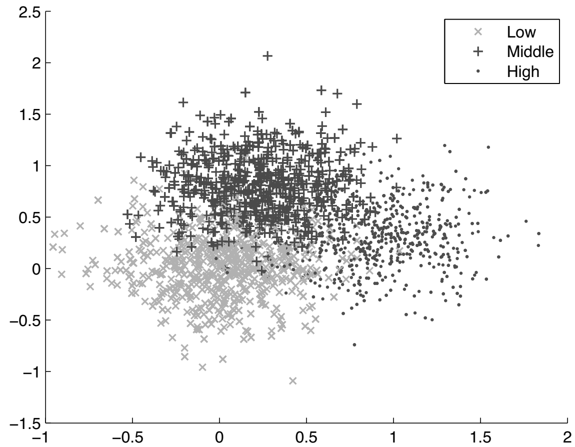

Figure 2: Query-document features of the synthetic data used in our experiments. Three different symbols denotes three level of irrelevance. Within the same level the features are generated by standard unit Gaussian.

4 Empirical Evaluation

We validate the efficiency of BRMs in three different types of applications: (1) synthetic data ranking, (2) image retrieval, and (3) video retrieval. We compare BRMs with the tradition ranking SVMs in three measurements: training time, number of support vectors, and prediction accuracy.

4.1 Datasets

Synthetic data. We randomly generate 20 query and for each query we generate 100 high relevant, 100 low relevant, and 100 irrelevant documents. For each query-document pair, we draw a two-dimension feature space from unit Gaussian mixtures. For the same level of relevance, the feature are drawn under the same center of the Gaussian (see Fig. 2).
WANG's dataset is a set of photos which contain 10 categories (Africa, Bench, Buildings, Buses, Dinosaurs, Elephants, Flowers, Houses, Mountains, and Food) and 100 images for each category [Wang *et al.*, 2001].
TRECVID 2005 is for video retrieval [Smeaton *et al.*, 2006]. We construct video sequences as following. We choose the shots in which there are at least 5 sub-shot key frames and select the first 5 key frames to form the sequence. In order to make a convenient evaluation, we ignore the shots which are not labeled in the ground-truth data. Finally we generate 347 video sequences for 10 topics. Here we use videos (key frames) as a ranking objects.

4.2 Experimental Settings

For the synthetic dataset, the query-document features are generated randomly. For WANG's dataset, we use color and shape features. For color features, we use color histograms and for shape features, we first detect the edge on 16 directions and use the number of pixels on the edge as features. For TRECVID 2005, we put all the 5 key frames together to form a $40 \times 40 \times 5$ three dimensional tensor for each video.

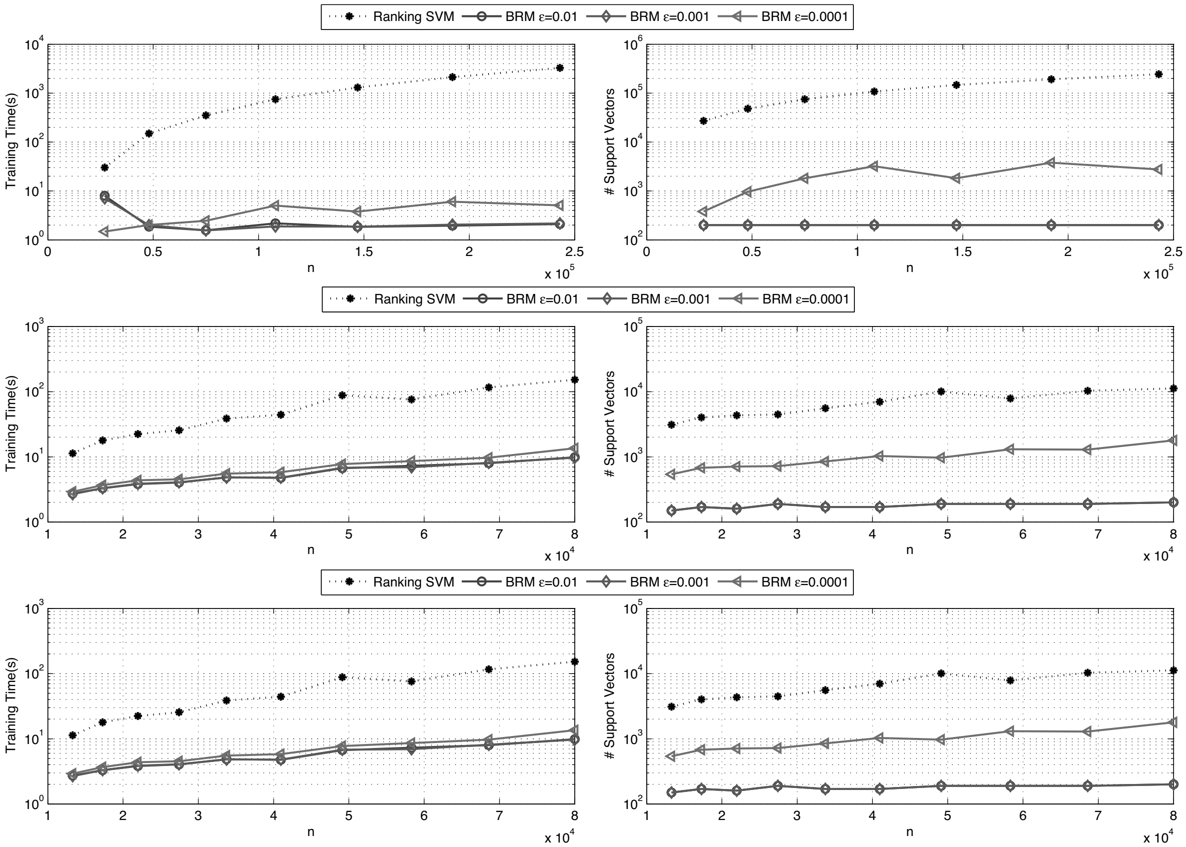

Figure 3: Training time (left) and number of support vectors (right) used by Ranking SVMs and BRMs in synthetic data(top), WANG's image dataset (middle), and TRECVID 2005 (bottom).

Then employ $D-1$ Orthogonal Tensor Decomposition to extract features [Ding *et al.*, 2008]. We compress a video to a $5 \times 5 \times 4$ tensor and reshape compressed tensor to a single column as features (which is a 100×1 vector). For all the datasets, we use Gaussian kernels.

For Ranking SVMs, we use the package in SVM light [Joachims, 1999], which can be downloaded at http://svmlight.joachims.org/.

4.3 Experimental Results and Discussions

We first evaluate the training efficiency of BRMs comparing to Ranking SVMs. We choose different sizes of subset of the data as training data. For Ranking SVMs, we use the default parameters. For BRMs, we try three different ϵ : $[0.0001, 0.001.0.01]$. The parameter ϵ is explained in [Tsang *et al.*, 2007]. Results are shown in the left part of Fig. 3. Notice that the time computing ranking function is linear to the number of support vectors which are also compared in the right part of Fig. 3. From the figures, we can see that our new method is significantly faster than Ranking SVMs and uses much fewer support vectors.

Since our algorithm is an approximation of Ranking SVM, we are also interested in how much accuracy our approach may lose. In this experiment, we use two fold cross-validation and set $\epsilon = 0.001$ for our method. We compare the Normalized Discounted Cumulative Gain (NDCG) which is a standard measurement in retrieval [Jarvelin, Kalervo and Kekalainen, Jaana, 2000]. Results are shown in Fig. 4. In our experiments, BRMs are comparable with, and sometimes better than Rank SVMs. We believe that the reason of the better performance might benefit from lower number of support vectors, which suggests lower level of model complexity and less overfitting to the training data.

5 Conclusions

Among all variations of SVMs, Ranking SVMs are the most expensive one which heavily requires new efficient optimization techniques. This paper proposed a Ball Ranking Machine to address the supervised ranking problem with much lower computational cost. We reformulate the objective function of Ranking SVM to a Minimum Enclosing Ball problem, which can be solved in an efficient way. Our new approach is applied to both image and video retrieval tasks. Empirical studies show that Ball Ranking Machine outperforms Ranking SVM significantly in terms of training time complexity and the number of support vectors without losing accuracy.

Acknowledgments

This research was supported by NSF-CCF 0830780, NSF-CCF 0917274, NSF-DMS 0915228.

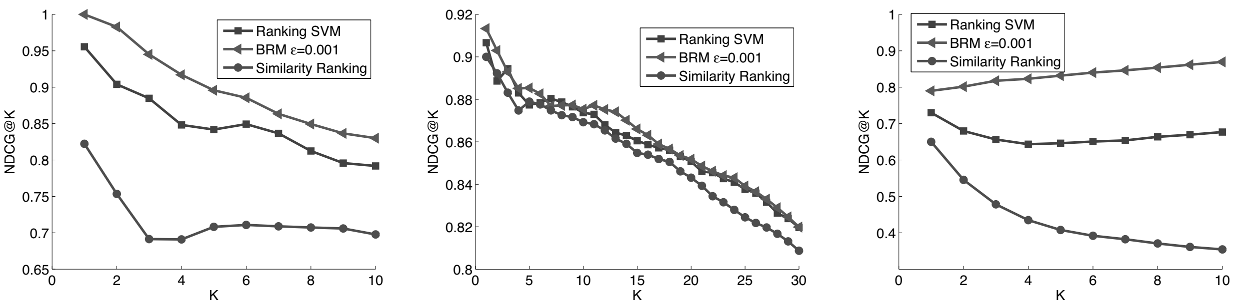

Figure 4: NDCG comparison of Ranking SVM and Ball Ranking Machine on synthetic data (top), WANG (middle), and TRECVID 2005 (bottom). For the WANG data set, Ball Ranking Machines are comparable with Ranking SVMs, for the synthetic data and TRECVID 2005 data sets, our method outperforms Ranking SVMs.

References

[Badoiu and Clarkson, 2008] M. Badoiu and K.L. Clarkson. Optimal core-sets for balls. *Computational Geometry*, 40(1):14–22, 2008.

[Burges *et al.*, 2005] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA, 2005. ACM Press.

[Ding *et al.*, 2008] C. Ding, H. Huang, and D. Luo. Tensor reduction error analysis: Applications to video compression and classification. In *CVPR*, pages 1–8, 2008.

[Fischer and Gartner, 2003] K. Fischer and B. Gartner. The smallest enclosing ball of balls: combinatorial structure and algorithms. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 292–301. ACM, 2003.

[Fischer *et al.*, 2003] K. Fischer, B. Gartner, and M. Kutz. Fast smallest-enclosing-ball computation in high dimensions. *Algorithms-ESA 2003*, pages 630–641, 2003.

[Freund *et al.*, 2003] Yoav Freund, Raj Iyer, R. E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 2003.

[Fuhr, 1989] Norbert Fuhr. Optimal polynomial retrieval functions based on the probability ranking principle. *ACM Transactions on Information Systems*, 7(3):183–204, July 1989.

[Herbrich *et al.*, 1999] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. *IJCNN*, pages 97–102, 1999.

[Jarvelin, Kalervo and Kekalainen, Jaana, 2000] Jarvelin, Kalervo and Kekalainen, Jaana. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Evaluation, pages 41–48, 2000.

[Joachims, 1999] T. Joachims. Making large–scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

[Joachims, 2002] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proc. KDD'02*, 2002.

[Platt, 1999] J.C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, pages 185–208. MIT press, 1999.

[Smeaton *et al.*, 2006] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and TRECVid, October 2006.

[Tsang *et al.*, 2005] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.

[Tsang *et al.*, 2007] Ivor W. Tsang, András Kocsor, and James T. Kwok. Simpler core vector machines with enclosing balls. In Zoubin Ghahramani, editor, *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 911–918. ACM, 2007.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[Wang *et al.*, 2001] James Ze Wang, Jia Li, and Gio Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture LIbraries. *IEEE Trans. Pattern Anal. Mach. Intell*, 23(9):947–963, 2001.

[Welzl, 1991] E. Welzl. Smallest enclosing disks (balls and ellipsoids). *New results and new trends in computer science*, pages 359–370, 1991.

[Xu and Schuurmans, 2005] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *Proceedings Of The National Conference On Artificial Intelligence*, volume 20, page 904. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.