Multimedia Retrieval via Deep Learning to Rank

Xueyi Zhao, Xi Li, and Zhongfei Zhang

***Abstract*—Many existing learning-to-rank approaches are incapable of effectively modeling the intrinsic interaction relationships between the feature-level and ranking-level components of a ranking model. To address this problem, we propose a novel joint learning-to-rank approach called Deep Latent Structural SVM (DL-SSVM), which jointly learns deep neural networks and latent structural SVM (connected by a set of latent feature grouping variables) to effectively model the interaction relationships at two levels (i.e., feature-level and ranking-level). To make the joint learning problem easier to optimize, we present an effective auxiliary variable-based alternating optimization approach with respect to deep neural network learning and structural latent SVM learning. Experimental results on several challenging datasets have demonstrated the effectiveness of the proposed learning to rank approach in real-world information retrieval.**

***Index Terms*—Deep neural network, joint learning, latent variable, learning to rank, structural SVM.**

I. Introduction

MULTIMEDIA retrieval is typically cast as a problem of learning to rank [1][2] over the data samples (e.g., image and text). In general, learning to rank is carried out at two mutually correlated levels, that is, *feature-level* and *ranking-level*. The former focuses on capturing the intrinsic interaction information on features by effective feature transformation [3]–[7] (e.g., feature selection, feature weighting, and feature mapping), while the latter aims to build effective ranking models for encoding the underlying structural ranking relationships among data samples.

Many conventional information retrieval approaches [8]–[12] only focus on the ranking level and do not involve the feature learning. Recently, several retrieval approaches [13]–[19] include both the feature level and the ranking level. However, they consider the two levels as two separated aspects, and thus are usually incapable of well modeling the latent interaction relationships between the feature-level and ranking-level.

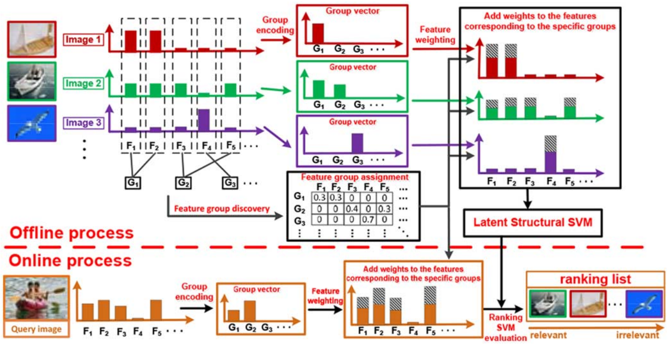

Fig. 1. Workflow of our proposed multimedia retrieval approach. Each data instance (e.g., an image) is represented by a set of features (i.e., F_1, F_2, …). In the feature group discovery process, we extract the feature group assignment matrix in which the features with their assigned groups (i.e., $G_1, G_2, \ldots$) are described. Meanwhile, the weight on each group is obtained via group coding. Subsequently, we add the weights to the features corresponding to the specific groups, and then train the modified feature vectors in the latent structural SVM framework in order to learn a ranking model. For a retrieval task, the feature vector of the query instance is weighted via feature weighting. By using the learned latent structural SVM model, the ranking list corresponding to the query is obtained.

In this work, we seek to construct an effective ranking model (based on existing simple features) for capturing the interaction information about the *feature-level* and *ranking-level* parts. As shown in Fig. 1, our proposed ranking model has the capability of adaptively exploring the latent feature grouping properties as well as the relative importance information for different feature groups, and meanwhile adapts to the learning-to-rank scenario in the sense of characterizing the structural ranking information on data samples. Typically, the structural SVM has been shown to be an effective framework for learning to rank. Furthermore, the structural SVM model is extended by adding latent variables [20], [21] to achieve the best ranking scores. In contrast, the feature grouping latent variables in our model are dependent on deep neural networks.

At the feature level, the feature grouping problem in our model is associated with deep low-rank latent structure learning. Specifically, the low-rank part of our model aims to discover the underlying feature groups (represented by latent variables) and obtain their corresponding group-specific weights for relative importance evaluation. The latent feature grouping variables are dependent on two-view deep neural networks, which seek for the inherent nonlinear mapping relationships between sample inputs and feature grouping from two different viewpoints (i.e., featurewise and samplewise).

At the ranking level, the objective of the latent structure learning part is to model the interaction relationships between low-rank learning and structural learning to rank by using the latent feature grouping variables. Subsequently, the aforementioned three parts are implemented in a joint learning framework that simultaneously learns the parameters for neural networks and structural ranking models. Directly solving the

Manuscript received December 26, 2014; revised February 12, 2015; accepted February 18, 2015. Date of publication March 04, 2015; date of current version March 11, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61472353, the China Knowledge Centre for Engineering Sciences and Technology, the Fundamental Research Funds for the Central Universities, the National Basic Research Program of China under Grant 2012CB316400, and by the Zhejiang Provincial Engineering Center on Media Data Cloud Processing and Analysis. The work of Z. Zhang was also supported by the U.S. National Science Foundation under Grant CCF-1017828. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joao Ascenso.

X. Zhao and Z. Zhang are with Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China (e-mail: xueyizhao@zju.edu.cn; zhongfei@zju.edu.cn).

X. Li is with College of Computer Science, Zhejiang University, Hangzhou, China (e-mail: xilizju@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2015.2410134

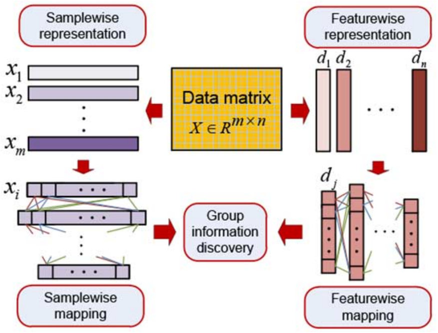

Fig. 2. Workflow of the group information discovery. Each row (or column) corresponds to a samplewise (or featurewise) representation. By achieving the samplewise and featurewise mappings simultaneously, the group information is discovered.

joint learning problem is intractable because the latent feature grouping variables are involved in several complex learning subtasks (e.g., back propagation for non-differentiable structural latent SVM learning). To make the joint learning task tractable, we propose an effective alternating approach that can be easily carried out in an iterative manner.

II. Algorithm Description

Our learning-to-rank model is based on a collection of m input samples (the dimensionality is n) denoted as a data matrix $X \in \mathcal{R}^{m \times n}$. As shown in Fig. 2, each row x_i of the matrix X corresponds to the i-th sample (a feature vector), while each column d_j of X is associated with the j-th feature component (a sample vector). To effectively explore the intrinsic interactions between feature grouping and ranking performance, we introduce our approach in two aspects (i.e., feature-level aspect and ranking-level aspect) stage by stage.

A. Feature-level: Feature Grouping Model

Feature grouping in our model is performed in a deep learning scheme that builds a mapping model from the original data to the latent grouping variables. In principle, the mapping model (as shown in Fig. 2) comprises two types of mappings, that is, featurewise mapping and samplewise mapping: i) The featurewise mapping utilizes the sample vector d_j as an input. Each element of the mapping output vector indicates the feature grouping membership. ii) The samplewise mapping adopts the feature vector x_i as an input. Each element of the mapping output vector is associated with a sample-related weighting factor within the learned feature groups.

For the featurewise deep neural network, the output of the l-th layer is denoted by $a_d^{(l)}$, which is obtained according to the output of the $(l-1)$-th layer. Mathematically, the function is given as $a_d^{(l)} = f(W_d^{(l)} a_d^{(l-1)} + b_d^{(l)})$. The variable $W_d^{(l)}$ denotes the weight matrix of the l-th layer of this view, $b_d^{(l)}$ denotes the bias constant of the l-th layer, and $f(\cdot)$ is the sigmoid function. The number of the layers of the featurewise deep neural network is denoted as L_d and the number of the groups is denoted as G. For the samplewise deep neural network, the notations are similarly defined (by taking the subscript x instead of the subscript d). Therefore, we have the similar units (like $W_x^{(l)}$, $b_x^{(l)}$, $a_x^{(l)}$, and L_x) to those in the featurewise deep neural network. Therefore, the variables with the subscript d correspond to the samplewise neural network and the variables with the subscript x correspond to the featurewise neural network.

Feature grouping framework: The featurewise network corresponds to the grouping membership of the features while the samplewise network corresponds to the associated weights of the groups. Finally, the feature grouping task is accomplished by solving a reconstruction-based optimization problem, which aims to reconstruct the feature components of the original samples through a weighted combination of feature groups. Hence, we have the following objective function:

$$F(W_x, b_x, W_d, b_d) = \sum_{i=1}^{m} \|x_i - a_{x_i}^{(L_x)} A_d^{(L_d)}\|^2 + R(W_x, W_d, A_d^{(L_d)}).$$

Here, $A_d^{(L_d)}$ is a matrix with the representation $A_d^{(L_d)} = (a_{d_1}^{(L_d)}, a_{d_2}^{(L_d)}, \ldots, a_{d_n}^{(L_d)})$. As with the previous definition, $a_{x_i}^{(L_x)}$ (or $a_{d_i}^{(L_d)}$) denotes the output of the last layer of the corresponding input x_i (or d_i). The multiplication $\{a_{x_i}^{(L_x)} A_d^{(L_d)}\}$ is the reconstruction term based on the outputs of the two-view deep neural networks.

Regularization: $R(W_x, W_d, A_d^{(L_d)})$ in the above objective function is a regularizer that enforces the sparsity constraints on soft feature grouping assignment. For convenience, let $\{\hat{\rho}_j = \frac{1}{n} \sum_{i=1}^{n} a_{d_i}^{(L_d)}\}_{j=1,\ldots,G}$ denote the average output over all the inputs for the featurewise neural network. In order to obtain the sparse solutions, we wish $\hat{\rho}_j$ to approximate a very low threshold ρ. In our approach, the KL-divergence between $\hat{\rho}_j$ and ρ is thus minimized as a soft constraint to ensure the closeness of $\hat{\rho}_j$ and ρ. Finally, the regularization term $R(W_x, W_d, A_d^{(L_d)})$ is formulated as:

$$\frac{\lambda}{2} \sum_{l=1}^{L_x - 1} \|W_x^{(l)}\|^2 + \frac{\mu}{2} \sum_{l=1}^{L_d - 1} \|W_d^{(l)}\|^2 + \beta \sum_{j=1}^{G} KL(\rho \| \hat{\rho}_j). \tag{1}$$

B. Ranking-level: Ranking Connection with Feature Grouping

After extracting the grouping variables that are obtained from our feature-level framework, we need to design a learning to rank scheme to well exploit these variables. In principle, the data from the different groups have different effects on the ranking task. By assigning different weights to the data from different groups, we utilize the feature grouping variables as the *latent* variables to refine the original data.

In the learning to rank application, $\mathcal{Y}$ denotes the set of all possible permutations of the ranking values. Let the operator $|\cdot|$ denote the number of data instances in the dataset and thus $\mathcal{Y} \subset \{1, 0, -1\}^{|X| \times |X|}$. For any ranking $\mathbf{y} \in \mathcal{Y}$, y_{ij} represents the element in the i-th row and the j-th column of $\mathbf{y}$. $y_{ij} = 1$ if vector x_i is ranked ahead of vector x_j, and $y_{ij} = -1$ if vector x_j is ranked ahead of x_i, and $y_{ij} = 0$ if x_i and x_j have an equal rank.

Latent structural SVM: In practice, we embed our two-view deep neural network into the latent structural SVM, in which a prediction function $f_w(x)$ is optimized to obtain the best ranking score as follows:

$$f_w(x) = \arg\max_{\mathbf{y} \in \mathcal{Y}} [w\Phi(x, \mathbf{y}, h)], \tag{2}$$

where the latent variable h corresponds to our latent feature grouping variables that are determined from two deep neural networks, and Φ stands for the combined feature function (the detailed definition of Φ is given in the experiment section for concise expression). In our model, the two-view neural

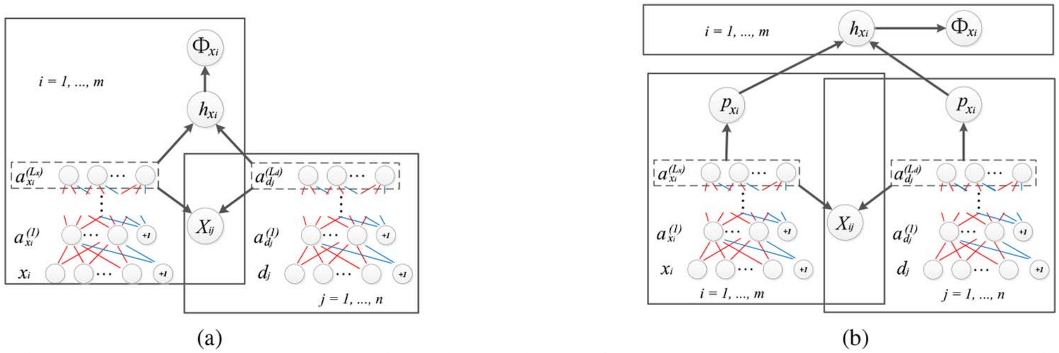

Fig. 3. Illustration of different joint learning schemes. The red lines denote the weights W in the neural networks and the blue lines denote the bias value b. (a) shows the directly joint learning scheme while (b) displays the joint learning scheme with an auxiliary layer (represented by p_{x_i} and p_{d_j}), which makes the optimization procedure easier. (a) Directly joint learning. (b) Joint learning with auxiliary layer.

network outputs $a_x^{(L_x)}$ and $a_d^{(L_d)}$ are utilized as the hidden variables of latent structural SVM. For clarity, we denote the joint latent variables $\{a_x^{(L_x)}, A_d^{(L_d)}\}$ as a single hidden variable $h = (a_x^{(L_x)}, A_d^{(L_d)})$ in its samplewise representation.

To explore the intrinsic interactions between neural networks module and latent structural SVM module, we present a joint learning scheme to simultaneously optimize these two modules. Thus, we have the following joint learning problem:

$$\min_{w} L(w,h) + F(W_x, b_x, W_d, b_d), \tag{3}$$

where the function $L(w,h)$ is the latent structural SVM formulation, which is formulated as:

$$\frac{\gamma}{2}\|w\|^2 + C\sum_{i=1}^{m}(\max_{\mathbf{y}\in\mathcal{Y}}[w\Phi(x_i,\mathbf{y},h) + \Delta(\mathbf{y}_i,\mathbf{y})] - w\Phi(x_i,\mathbf{y}_i,h)),$$

where Δ measures the prediction loss (which is defined in the experiment part). Clearly, the above optimization problem is difficult to solve since the function $\max_{\mathbf{y}\in\mathcal{Y}}[w\Phi(x_i,\mathbf{y},h) + \Delta(\mathbf{y}_i,\mathbf{y})]$ is nondifferentiable with respect to h. Alternatively, we derive a relaxed optimization problem by introducing a new auxiliary variable p to approximate the original h. As shown in Fig. 3, we insert a new auxiliary layer between the deep neural networks module and latent structural SVM module. In this case, the last layer outputs of the neural networks become the approximate inputs of the latent structural SVM module with a soft regularization. In principle, this strategy is inspired by ADMM [22] principle for decorrelating the dependent variables. Hence, the optimization problem becomes to minimize:

$$\begin{aligned} &L(w,\{p_x,P_d\}) + F(W_x,b_x,W_d,b_d) \\ &\quad + \frac{1}{2}\sum_{i=1}^{m}\|p_{x_i} - a_{x_i}^{(L_x)}\|^2 + \frac{1}{2}\|P_d - A_d^{(L_d)}\|^2. \end{aligned} \tag{4}$$

Apparently, the grouping matrix discovers the weight description of the image that emphasizes the valuable areas for ranking. By featurewise weighting, the joint feature function in the $L(w,\{p_x,P_d\})$ (p_x with the subscript x represents the auxiliary variable of samplewise neural network output corresponding to the input x) is given as:

$$\Phi(x,\mathbf{y},h) = \Phi(x,\mathbf{y},p_x,P_d) = \Phi((p_xP_d)diag(x),\mathbf{y}), \tag{5}$$

where $diag(x)$ is the diagonal matrix with the elements of x on the main diagonal. Since (p_xP_d) is the feature weights corresponding to the feature vector x and $(p_xP_d)diag(x) = \sum_i (p_xP_d)_i x_i$, Eq. (5) is introduced for weighting the feature vector x. The optimization problem in Eq. (4) can be effectively solved by the back propagation and a standard structural SVM scheme that is given in our supplementary file[1].

III. Experiments

A. Experimental Setup

Datasets: Five challenging datasets in different modalities are used in the experiments, namely NUSWIDE (with 500-dimensional SIFT feature), WEB10K (with 136-dimensional provided feature), Caltech256 (with 512-dimensional color histogram), LabelMe (with 512-dimensional color histogram), and Newsgroup (with 8964-dimensional TF-IDF feature).

For NUSWIDE dataset, 3,000 query samples are randomly selected to be the training set and 2,000 to be the validation set. The rest are used to form the testing set. For the other four datasets, 1,500 query samples are randomly selected to be the training set and 1,000 to be the validation set.

Evaluation Criteria: We use Mean Average Precision (MAP), Normalized Discounted Cumulative Gain ($NDCG$), and Precision-Recall Curve as the performance measures. The three criteria are the commonly used criteria such that a larger area under the Precision-Recall Curve (or higher MAP, $NDCG$) is associated with a better result. Typically, we report the mean NDCG@20 over a set of queries ($NDCG$ for short in the rest of the paper).

Parameter Settings: Let $p = rank(\mathbf{y})$ denote the true rankings of the retrieved vectors. The loss function in our model is given as $\Delta(\mathbf{y}_i,\mathbf{y}) = 1 - MAP(rank(\mathbf{y}_i,\mathbf{y}))$. The definition of the combined feature function is $\Phi(x,y) = \sum_{x_i\in X^+}\sum_{x_j\in X^-} y_{ij}\frac{\phi(q,x_i)-\phi(q,x_j)}{|X^+||X^-|}$, where X^+ and X^- denote the sets of the relevant and irrelevant vectors, respectively. q is the query vector and the structure feature mapping function $\phi(q,x)$ is given as $\phi(q,x) = \frac{<q,x>\cdot x}{\|q\|\cdot\|x\|}$ in our experiments. The parameters W_x, b_x, W_d, and b_d are initialized in random range from 0 to 1. For setting the parameters $(\gamma,\mu,\lambda,\beta,C)$, we perform ten trials. The values of these five parameters are selected from the ranges that $\gamma \in (0,10)$, $C \in (0,100)$, and $(\mu,\lambda,\beta) \in (0,1)$. The setting with the best performance on the validation set is selected. Both neural networks are set as three layers with 10 output nodes and 1000 hidden nodes.

[1] http://dsec.zju.edu.cn/static/DLSSVM_SuppFile.pdf

TABLE I
THE INVESTIGATION OF THE ADVANTAGE OF THE DEEP NEURAL NETWORKS FOR LEARNING THE GROUP PROPERTY OF THE FEATURES. CLEARLY, OUR APPROACH WITH DEEP NEURAL NETWORKS ACHIEVES BETTER RANKING PERFORMANCES THAN SVD-LSSVM, WHICH EXPLOITS SINGULAR VALUE DECOMPOSITION TO DISCOVER THE GROUP INFORMATION

Metric	Approach	NUSWIDE	Caltech256	WEB10K	LabelMe	Newsgroup
MAP	DL-SSVM	**0.375**	**0.253**	**0.424**	**0.483**	**0.350**
	SVD-LSSVM	0.292	0.172	0.338	0.327	0.253
$NDCG$	DL-SSVM	**0.534**	**0.383**	**0.561**	**0.594**	**0.442**
	SVD-LSSVM	0.458	0.331	0.513	0.527	0.363

TABLE II
THE RANKING PERFORMANCES WITH OUR JOINT APPROACH AND THE SEPARATE LEARNING APPROACH. CLEARLY, OUR JOINT LEARNING APPROACH IS CAPABLE OF ACHIEVING BETTER RANKING RESULTS THAN THE SEPARATE LEARNING APPROACH

Metric	Approach	NUSWIDE	Caltech256	WEB10K	LabelMe	Newsgroup
MAP	DL-SSVM	**0.375**	**0.253**	**0.424**	**0.483**	**0.350**
	S-DL-SSVM	0.302	0.186	0.341	0.326	0.268
$NDCG$	DL-SSVM	**0.534**	**0.383**	**0.561**	**0.594**	**0.442**
	S-DL-SSVM	0.460	0.343	0.517	0.524	0.372

B. Algorithm Analysis and Evaluation

Investigation of the Two-view Neural Networks: In order to investigate the advantage of the deep neural networks for learning the group property of the features, we exploit singular value decomposition as a baseline tool to achieve the formulation $X \approx A_x^{(L_x)} A_d^{(L_d)}$ which can be considered as a one-layer linear neural network without activation function (in our approach, the activation function is the sigmoid function). This approach is referred to as SVD-LSSVM for the comparison purpose. Table I indicates that our approach with deep neural networks achieves better ranking performances than SVD-LSSVM. The results demonstrate that our two-view deep neural networks, which are capable of well capturing the underlying feature grouping, achieve better solutions than low-level linear mapping such as SVD.

Evaluation of the Joint Learning: To investigate the advantage of the joint learning strategy that simultaneously optimizes the neural networks and the latent structural SVM, we implement an approach that conducts learning separately in the two modules (referred as S-DL-SSVM) for the comparison purpose, which extracts the hidden group information first by exploiting the deep neural networks and then assigns the hidden variables to the ranking algorithms as with two independent steps. The implementation is able to be obtained by following the solutions given in this paper and we omit the details here. Table II reports the ranking results of DL-SSVM and S-DL-SSVM on the five datasets. The results indicate that the joint learning approach is capable of achieving better ranking results than the separate learning approach due to the fact that the joint learning framework captures the interaction information about the feature-level and ranking level parts and contributes to achieving an effective ranking model.

C. Comparison with Competing Algorithms

In the following, we compare our approaches with the competing algorithms in the multimedia ranking and information retrieval application, followed by the scalability investigation. The competing algorithms include RankSVM [8], SVM$_{MAP}$[10], AdaRank [23], ListNet [11], and RPCAH [24].

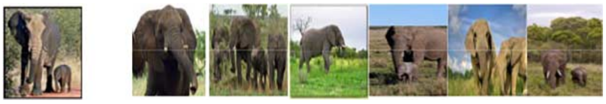

Fig. 4. One retrieval example on the NUSWIDE dataset. The left part shows the query sample while the right part displays a few retrieved images using our approach.

TABLE III
THE RANKING RESULTS OF DIFFERENT LEARNING TO RANK ALGORITHMS IN $NDCG$ ON THE FIVE DATASETS. CLEARLY, OUR APPROACH ACHIEVES THE BEST RANKING PERFORMANCES

Metric	Approach	NUSWIDE	Caltech256	WEB10K	LabelMe	Newsgroup
$NDCG$	DL-SSVM	**0.534**	**0.383**	**0.561**	**0.594**	**0.442**
	RankSVM	0.453	0.324	0.505	0.518	0.359
	SVM$_{MAP}$	0.326	0.315	0.511	0.535	0.372
	AdaRank	0.481	0.336	0.523	0.527	0.384
	ListNet	0.485	0.332	0.514	0.541	0.402
	RPCAH	0.479	0.346	0.528	0.536	0.407

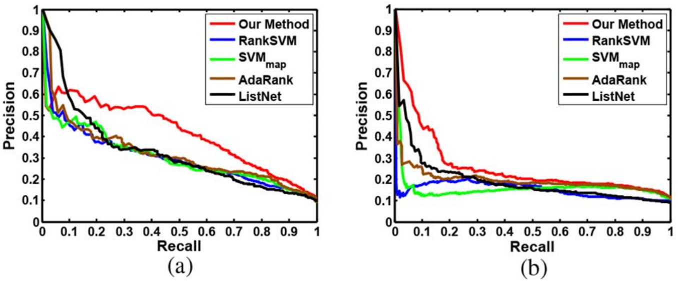

Fig. 5. Precision-Recall Curve retrieval performances of different algorithms on the NUSWIDE and Caltech256 datasets. Clearly, our approach obtains better precision-recall retrieval results in most cases (a) NUSWIDE (b) Caltech256.

Comparison on Ranking Performance: Table III shows the ranking results of different learning to rank algorithms in $NDCG$ on the five datasets. Clearly, our approach achieves the best ranking performances. Since the competing methods do not exploit the interaction between the feature-level learning and ranking-level learning, DL-SSVM has advantage over the competing methods. The experimental results indicate the DL-SSVM is the best ranking model due to its deep interaction information utilization. Besides, we show a retrieval example on the NUSWIDE dataset in Fig. 4.

Precision-Recall Curve Performance: Fig. 5 shows the average Precision-Recall Curves of different ranking algorithms on the two datasets (i.e., NUSWIDE and Caltech256) in the multimedia information retrieval application. From Fig. 5, we observe that our approach obtains higher retrieval precisions than the other four competing ranking algorithms with a fixed recall rate in most cases. The outperform of our approach demonstrate the advantage of modeling the intrinsic interaction between feature grouping and latent structural SVM approach.

IV. CONCLUSION

We have proposed an effective learning to rank approach called Deep Latent Structural SVM (DL-SSVM), which simultaneously solves the optimization problems of learning the effective features and learning to rank. Our joint learning approach is capable of exploring the intrinsic interactions between deep neural networks and latent structural SVM learning to rank. DL-SSVM is also capable of discovering the latent group information which is well fit for the latent structural SVM model. As a result, our approach results in an impressive effectiveness for learning to rank. Experimental results on five datasets have demonstrated the promise of the proposed approach in the information retrieval application.

REFERENCES

[1] C. J. Burges, “From ranknet to lambdarank to lambdamart: An overview,” *Learning*, vol. 11, pp. 23–581, 2010.

[2] J. Weston, C. Wang, R. Weiss, and A. Berenzweig, “Latent collaborative retrieval,” in *arXiv preprint arXiv:1206.4603*, 2012.

[3] Y. Zhou and K. E. Barner, “Locality constrained dictionary learning for nonlinear dimensionality reduction,” *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 335–338, 2013.

[4] A. Coates and A. Y. Ng, “Learning feature representations with k-means,” *J. Neural Netw.: Tricks of the Trade*, pp. 561–580, 2012.

[5] C. Chen, J. Cai, W. Lin, and G. Shi, “Surveillance video coding via low-rank and sparse decomposition,” in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 713–716.

[6] J. Feng, B. Ni, Q. Tian, and S. Yan, “Geometric lp-norm feature pooling for image classification,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn*, 2011, pp. 2609–2704.

[7] Y. Bengio, “Learning deep architectures for AI,” *J. Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[8] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proc. ACM Int. SIGKDD Conf*, 2002, pp. 133–142.

[9] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, “An efficient boosting algorithm for combining preferences,” *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, 2003.

[10] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, “A support vector method for optimizing average precision,” in *Proc. ACM Int. SIGIR Conf*, 2007, pp. 271–278.

[11] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, “Learning to rank: From pairwise approach to listwise approach,” in *Proc. ACM Int. Conf. Mach. Learn.*, 2007, pp. 129–136.

[12] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, “Listwise approach to learning to rank: Theory and algorithm,” in *Proc. ACM Int. Conf. Mach. Learn.*, 2008, pp. 1192–1199.

[13] Y. Fujiwara, M. Nakatsuji, H. Shiokawa, T. Mishima, and M. Onizuka, “Fast and exact top-k algorithm for pagerank,” *Proc. Assoc. Advancement Artif. Intell.*, 2013.

[14] W. Gao and Z.-H. Zhou, “Uniform convergence, stability and learnability for ranking problems,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1337–1343, AAAI Press.

[15] L. Zheng and S. Wang, “Visual phraselet: Refining spatial constraints for large scale image search,” *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 391–394, 2013.

[16] V. Sharmanska, N. Quadrianto, and C. H. Lampert, “Learning to rank using privileged information,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 825–832.

[17] W. Pan and L. Chen, “Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2691–2697, AAAI Press.

[18] S. Clémençon and S. Robbiano, “Anomaly ranking as supervised bipartite ranking,” in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 343–351.

[19] E. F. Can, W. B. Croft, and R. Manmatha, “Incorporating query-specific feedback into learning-to-rank models,” in *Proc. ACM Int. SIGIR Conf. Research and Development in Information Retrieval*, 2014, pp. 1035–1038.

[20] D. Chen, D. Batra, W. Freeman, and M. Johnson, “Group norm for learning latent structural svms,” in *Proc. NIPS Workshop Optim. Mach. Learn.*, 2011.

[21] C.-N. J. Yu and T. Joachims, “Learning structural svms with latent variables,” in *Proc. ACM Int. Conf. Mach. Learn.*, 2009, pp. 1169–1176.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *J. Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, 2011.

[23] J. Xu and H. Li, “Adarank: A boosting algorithm for information retrieval,” in *Proc. ACM Int. SIGIR Conf*, 2007, pp. 391–398.

[24] C. Leng, J. Cheng, and H. Lu, “Random subspace for binary codes learning in large scale image retrieval,” in *Proc. ACM SIGIR Conf*, 2014, pp. 1031–1034.