Efficient Margin-Based Rank Learning Algorithms for Information Retrieval

Rong Yan and Alexander G. Hauptmann

School of Computer Science
Carnegie Mellon University
Pittsburgh PA, 15213, USA
{yanrong, alex+}@cs.cmu.edu

Abstract. Learning a good ranking function plays a key role for many applications including the task of (multimedia) information retrieval. While there are a few rank learning methods available, most of them need to explicitly model the relations between every pair of relevant and irrelevant documents, and thus result in an expensive training process for large collections. The goal of this paper is to propose a general rank learning framework based on the margin-based risk minimization principle and develop a set of efficient rank learning approaches that can model the ranking relations with much less training time. Its flexibility allows a number of margin-based classifiers to be extended to their rank learning counterparts such as the ranking logistic regression developed in this paper. Experimental results show that this efficient learning algorithm can successfully learn a highly effective retrieval function for multimedia retrieval on the TRECVID'03-'05 collections.[1]

1 Introduction

Many applications have to present their results in form of ranked lists, such as information retrieval that sorts documents according to their relevance to the query and collaborative filtering that sorts items for a user based on the rating provided by other users. All of these applications can benefit if we can automatically learn a better ranked list from some given training examples. In this paper, we specifically consider such a rank learning problem in the context of information retrieval and evaluate it using multimedia retrieval collections. Typically, the training data of a retrieval system include a set of queries, a set of retrieved documents for each query and relevance judgments that manually label some pairs of queries and retrieved documents as relevance and others as irrelevance. Our task is to learn a retrieval utility function to rank the documents using the manual relevance judgments of the training queries.

Previous retrieval models usually cast rank learning into a binary classification problem that treats the relevant query-document pairs as positive data and

[1] This research is partially supported by Advanced Research and Development Activity (ARDA) under contract number H98230-04-C-0406 and NBCHC040037, and by the National Science Foundation under Grant No. IIS-0535056.

H. Sundaram et al. (Eds.): CIVR 2006, LNCS 4071, pp. 113–122, 2006.

irrelevant pairs as negative data. Some examples include the generative models used in the binary independence model [1] and the discriminative models such as the maximum entropy model [2]. Despite its great successfulness, converting retrieval into classification might suffer from several disadvantages. For example, since the classification accuracy has no direct relationship with the retrieval measure, a learning algorithm that can achieves a high classification accuracy might not produce a good performance in terms of ranking. Therefore, there are a few recent attempts to develop learning algorithms that can explicitly account for ranking relations in information retrieval [3,4,5,6,7]. Most of these rank learning approaches attempt to model the pairwise ranking preferences between every pair of relevant and irrelevant training examples. They are built on a solid foundation because it has been shown that minimizing the discordant pairs of examples are closely related to the commonly used ranking criteria. However, the effort of modeling every pair of examples often leads to a prohibitive learning process and thus limits their applications in practice.

In this paper, we propose a general rank learning framework based on the principle of margin-based risk minimization, which can be generalized to a large family of rank learning approaches such as Ranking SVMs [3] and RankBoost [4]. To make the optimization less computational intensive but still keep the ability to model the ranking relations between examples, we further propose an approximate but efficient rank learning framework by bounding the pairwise risk function. In particular, we designed a new learning algorithm called ranking logistic regression(RLR) by plugging in the logit loss function. Experiments show that this efficient learning algorithm can successfully learn a highly effective retrieval function for multimedia retrieval on the TRECVID'03-'05 collections.

2 Related Work

The wide range of applications for rank learning has inspired numeric approaches to handle this problem especially in the context of information retrieval. One typical direction of rank learning is to formulate it into an ordinal regression problem, i.e., mapping the labels to an ordered set of numerical ranks. Herbrich et al. [8] model the ranks as intervals on the real line, and optimize the loss function based on the true ranks and features. Following the similar idea, "PRank" [9] is developed based on an online linear learning algorithm called perceptron that uses one example at a time to update the linear feature weights. The ordinal regression formulation has been proven to be effective in the task of collaborative filtering. However, it might not be suitable for retrieval because the absolute rankings over documents are usually expensive to collect and users are less willing to provide such a detailed feedback in practice. Moreover, all the objects in ordinal regression have to be ranked in the same scale. But for retrieval, the ranking relationships only need to be consistent within a query which can greatly reduce the number of constraints.

As an alternative of learning the absolute numerical ranks, the approaches that model the relative ranking preferences between pairs of training data has

also been investigated recently. In the setting of collaborative filtering, Freund et al. [4] proposed the RankBoost algorithm which learns to rank a set of objects by combining multiple "weak" classifiers to build up a more accurate composite classifiers. The ranking SVMs proposed by Joachims [3] is constructed on a risk-minimization framework with the goal to minimize the number of misorderings between the predicted ranks and target ranks. Bearing resemblance to the common classification SVMs, ranking SVMs can be solved with similar optimization techniques. Based on a simple probabilistic cost function, Burges et al. [5] investigated a gradient descent method called RankNet to learn ranking functions with a neural network implementation. More recently, Chua et al. [6] developed a ranking maximal figure-of-merit(MFoM) algorithm by maximizing the area under the ROC curve. This approach has gained its success in the domain of video semantic feature extraction. In essence, aforementioned rank learning algorithms transform ranks into a set of pairwise relationships between relevant and irrelevant examples and thus cast it into a classification problem built on example pairs. However, above algorithms usually suffer from a expensive training process due to the explosive amount of training data after coupling each relevant and irrelevant documents, especially when the number of underlying training documents is large. For example, a query with 100 relevant documents and 900 irrelevant ones will result in 90,000 pairs of training examples, which is computationally intensive for many learning algorithms. It would be helpful to develop an efficient rank learning algorithm that is able to capture the ranking relationship while with a less learning time.

3 A Margin-Based Framework for Learning Ranks

We begin by introducing the basic notations and terminologies used in this work. The term *document* is referred to as the basic unit of retrieval throughout this paper. For example, in the TRECVID video retrieval task, the documents stands for video shots. A query collection $\mathcal{Q}$ contains a set of queries $\{q_1, ..., q_t, ..., q_{M_Q}\}$ where q_t can have either a set of keywords, a detailed text descriptions or even possibly image, audio, video query examples. A search collection $\mathcal{D}$ contains a set of documents $\{d_1, ..., d_j, ..., d_{M_D}\}$. D_q^+ is the collection of relevant documents and D_q^- is the collection of irrelevant documents for query q. M_D^+ and M_D^- are the number of documents in each collection. For each query q and document d, we can generate a bag of ranking features denoted as $f_i(d, q), i = 1..N$. For instance, in the context of multimedia retrieval, the features can be generated from multiple knowledge sources including either the uni-modal retrieval experts built from various modalities or the indexing of predefined semantic video concepts.

Formally, a ranking (partially ordered list) is a binary relation defined on $D \times D$ with the properties of weak ordering. The goal of a retrieval system is to find a ranking function r_f to approximate the optimal ranking r^*, where r_f means the documents are sorted in a descending order of the retrieval function $f(d_i, q)$. But the prerequisite of optimizing the ranking function is to define an appropriate similarity measure between two rankings. As pointed out by

Joachims [3], Kendall's τ is one of the most frequently used criteria to compare ordinal correlations of two random variables. To explain the Kendall's τ measure, let us define concordant pairs as the document pair (d_i, d_j) when r_1 and r_2 agree on their orders, otherwise discordant pairs. Based on the number of discordant pairs Q, Kendall's τ can be defined as $\tau(r_1, r_2) = 1 - \frac{4Q}{M_D(M_D-1)}$. In the case of binary relevance scale, i.e., all of the documents are judged as either relevant or irrelevant, maximizing $\tau(r_1, r_2)$ is the same as minimizing the average ranks of relevant documents. Since the definition of Kendall's τ only depends on Q, maximizing the $\tau(r_1, r_2)$ is also equivalent to minimizing the number of inversions Q. More importantly, the inverse of Q provides a lower bound of another frequently used performance measure in information retrieval called average precision [3]. Therefore, it is reasonable to develop a rank learning algorithm that attempts to minimize the number of inversions between the predicted ranking r_f and the target ranking r^* in the training data.

In information retrieval, most of the learning approaches simplify the rank learning to be a binary classification problem and many of them can be derived from a learning framework that aims at minimizing the following regularized empirical risk [10],

$$\min_f R_{reg}(f) = \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} L(y_j f(d_j, q_t)) + \nu \Omega(\|f\|_{\mathcal{H}}), \tag{1}$$

where y_j is the binary relevance label for j^{th} training document d_j, L is the empirical loss function, $\Omega(\cdot)$ is some monotonically increasing regularization function and ν is the regularization constant. The component of $yf(d, q)$ is usually called "margin" in the literature [10] and hence the learning framework is called *margin-based risk minimization* framework. However, such a classification framework might have difficulties in dealing with the retrieval task. For example, because there are only a small fraction of relevant examples in the collection and many others are left as irrelevant ones, a classification algorithm that always provides negative prediction will unfortunately achieve a high predictive accuracy. Moreover, the classification accuracy has no relationship with the retrieval measure such as average precision. Maximizing the classification accuracy does not necessarily imply a higher ranking effectiveness. To address this issue, we can consider switching the learning criterion to optimize the number of discordant pairs Q between the predicted ranking and the target ranking, i.e., $\sum_{q_t} \sum_{d_j \in D^+_{q_t}} \sum_{d_k \in D^-_{q_t}} I(f(d_j, q_t) - f(d_k, q_t))$ where $I(\cdot)$ is the indicator function. Unfortunately, a direct optimization on the general form of above equation has been shown to be NP-hard. But following the similar idea of margin-based risk minimization, we can replace the binary misclassification error $I(\cdot)$ into a continuous, convex and monotonically decreasing loss function $L(\cdot)$ in an attempt to facilitate the learning process. By introducing an additional regularization term, we can obtain the following unified margin-based rank learning framework,

$$\min_f RR_{reg}(f) = \sum_{q_t \in Q} \sum_{d_j \in D^+_{q_t}} \sum_{d_k \in D^-_{q_t}} L(f(d_j, q_t) - f(d_k, q_t)) + \nu \Omega(\|f\|_{\mathcal{H}})$$

$$= \sum_{q_t \in Q} \sum_{d_j \in D_{qt}^+} \sum_{d_k \in D_{qt}^-} L\left(\sum_{i=1}^{n} \lambda_i [f_i(d_j, q_t) - f_i(d_k, q_t)]\right) + \nu\Omega(\|f\|_{\mathcal{H}}), \quad (2)$$

where the retrieval function $f(d_j, q)$ is expressed as a linear function of the ranking features due to its retrieval effectiveness and simple presentation, i.e., $f(d_j, q) = \sum_{i=1}^{n} \lambda_i f_i(d_j, q)$. By optimizing the risk function, we can compute the risk minimization estimator λ_i^* for each ranking feature $f_i(d_j, q)$. With different choices of loss functions and regularization terms, a large family of rank learning algorithms can be derived from Eqn(2). For example, ranking support vector machines can be obtained by setting loss function to be the hinge loss and regularization factor to be $\|w\|_2^2$. RankBoost can be viewed as a rank learning algorithm with the exponential loss function. A recent proposed linear discriminant ranking model(LDM) [7] can be derived by using a binary loss function without regularization terms and setting $f(d_j, q)$ to be a linear function.

The rank learning framework presented in Eqn(2) lends itself to another advantage over the margin-based classification framework. Before further discussions, let us define a useful property called rank consistency,

Definition 1 (Rank consistency). *If a risk minimization estimator λ_i^* satisfies the following conditions: 1) $\lambda_i^* \geq 0$ when $\forall d_j \in D_q^+, \forall d_k \in D_q^-, f_i(d_j, q) \geq f_i(d_k, q)$, and similarly 2) $\lambda_i^* \leq 0$ when $f_i(d_j, q) \leq f_i(d_k, q)$, we will call the estimator is consistent with the data ranking. Note that we assume $f_i(\cdot)$ does not take a trivial constant value.*

It is intuitive to expect the parameters estimated from a rank learning algorithm to satisfy the property of rank consistency. For instance, let us assume the binary outputs of an anchor person detector is one of the ranking features in the multimedia retrieval system, where $f_a = 0$ means no anchor available and $f_a = 1$ otherwise. For a specific query, if we find all of the relevant documents do not contain any anchor shots, i.e., $f_a(d_j, q) \leq f_a(d_k, q)$, then it is naturally to expect the corresponding weight λ_a to be lower than 0, because a negative λ_a can push the relevant examples closer to the top ranked positions.

Unfortunately, simple margin-based classifiers do not offer any guarantees on this intuitive property. In other words, for a ranking feature, even when its values in the relevant documents are always lower than that in the irrelevant documents, the corresponding weight estimator can still be positive. This is because general classification algorithms did not take the ranking information into consideration and the violation of rank consistency might sometimes provide better separability between positive/negative examples rather than better retrieval performance. In contrast, the proposed margin-based rank learning framework preserves such a property, namely, the λ^* learned from Eqn(2) is always consistent with the data ranking if $L(\cdot)$ is a monotonically decreasing function. The proof is given in Appendix. This fact further explains why the proposed margin-based rank learning framework is a better candidate for the retrieval problem.

4 Efficient Rank Learning Algorithms

The above margin-based rank learning framework is quite general, but as mentioned before, optimizing the pairwise risk function in Eqn(2) in a brute force manner need to take care of an explosive number of training pairs between every relevant and irrelevant documents. Therefore, it is desirable to develop a more efficient algorithm to speed up the learning process. In this section, we will describe one of such types of efficient learning algorithms derived from the general rank learning framework. Unless stated otherwise, the following discussions assume the loss function $L(\cdot)$ is convex and satisfies $2L(x/2) \geq L(x)$. Under this assumption[2], we can have the following inequality,

$$RR_{prox}(f) \geq RR'_{reg}(f) \geq \frac{1}{2}[RR_{prox}(f) - RR_{prox}(-f)], \tag{3}$$

where $RR'_{reg}(f)$ is the pairwise ranking risk defined in Eqn(2) without the regularization factor and $RR_{prox}(f)$ is the approximate ranking risk function based on a shifted retrieval function $f^{\alpha}(d_j, q) = \sum_{i=1}^{n} \lambda_i [f_i(d_j, q) - \alpha_i]$,

$$RR_{prox}(f) = \sum_{q_t} \left\{ \sum_{d_j \in D_{qt}^{+}} M_D^{-} L\left(f^{\alpha}(d_j, q_t)\right) + \sum_{d_k \in D_{qt}^{-}} M_D^{+} L\left(-f^{\alpha}(d_k, q_t)\right) \right\}. \tag{4}$$

The proof of inequality Eqn(3) is provided in the Appendix. Both bounds are tight in the sense that all three parts are equal when $L(\cdot)$ is a linear function. Therefore, in lieu of optimizing the pairwise ranking function, we can consider minimizing the $RR_{prox}(f)$ as a reasonable surrogate. Meanwhile, it is instructive to compare and contrast $RR_{prox}(f)$ with the margin-based classification risk function presented in Eqn(1). As can be seen, if we set the label y of d_j to be $+1$ and that of d_k to be -1 in Eqn(1), these two risk functions have a similar form with each other. Therefore, minimizing $RR_{prox}(f)$ has a small computational complexity $O(M_D^{+} + M_D^{-})$, which is much faster than minimizing the pairwise ranking risk function with a complexity $O(M_D^{+} M_D^{-})$. However, $RR_{prox}(f)$ also bears some major differences with the classification risk R_{reg}, because (1) it weights the relevant documents more heavily by a ratio of M_D^{-}/M_D^{+}; (2) it drops the constant feature term, which is usually available for classification to capture the shifts of decision boundary; 3) it shifts each feature vector by the parameter α_i. These differences has made the $RR_{prox}(f)$ a better choice for the rank learning such as the advantage of balanced data distributions.

In the following implementation, we specially adopt the logit loss $L_R(x) = \log(1 + \exp(-x))$ as the empirical loss function due to its retrieval effectiveness and optimization simpleness. But before proceeding we need to decide the value of the shifting parameters α. One idea is choose α to minimize the gaps between the lower bound and the upper bound, i.e., $\min_{\alpha}[RR_{prox}(f) + RR_{prox}(-f)]/2$ so as to make RR_{prox} a tight approximation for RR'_{reg}. We approach this by utilizing the inequality $L_R(x) + L_R(-x) \leq 2 + |x|$ and thus we can transform the optimization problem into a series of minimization problem w.r.t. each α_i,

[2] This is a very general condition with a large family of loss functions satisfied, such as the hinge loss(SVMs), logistic loss and binary loss function.

$$\min_{\alpha_i} \sum_{q_t} \Big\{ \sum_{d_j \in D^+_{qt}} M^-_D |f_i(d_j, q_t) - \alpha_i| + \sum_{d_k \in D^-_{qt}} M^+_D |f_i(d_k, q_t) - \alpha_i| \Big\}. \quad (5)$$

The optimal estimator α_i^* can be written as follows,

$$\alpha_i^* = \text{median} \left[\bigcup_{\forall j,t} \Big\{ f_i(d_j, q_t) \Big\}_{M^-_D} \cup \bigcup_{\forall k,t} \Big\{ f_i(d_k, q_t) \Big\}_{M^+_D} \right], \quad (6)$$

where $\{x\}_n$ denote a set of n elements with the same value x. By substituting the optimal α_i^* and the logit loss into Eqn(4), we can proceed to optimize the combination parameter λ_i^* as follows,

$$\min_{\lambda} \sum_{q_t} \Big\{ \sum_{d_j \in D^+_{qt}} M^-_D L_R \Big(\sum_i \lambda_i f^*_{ijt} \Big) + \sum_{d_k \in D^-_{qt}} M^+_D L_R \Big(- \sum_i \lambda_i f^*_{ikt} \Big) \Big\} + \nu \sum_i \lambda_i^2, \quad (7)$$

where $f^*_{ijt} = f_i(d_j, q_t) - \alpha_i^*$. The optimal estimation of λ_i can be achieved by using any gradient descent methods such as iterative reweighted least squares(IRLS) algorithm [11]. We also prove in the Appendix that the estimation from Eqn(7) is consistent with the data ranking. In the rest of the paper, we will call this algorithm *ranking logistic regression*(RLR).

5 Experiments

Our experiments are designed based on the guidelines of the manual retrieval task in the TREC video retrieval evaluation(TRECVID), which requires an automatic video retrieval system to search relevant documents without any human feedbacks. To evaluate the proposed learning algorithms, we used TRECVID'03-'05 video collections which officially provide 25 multimodal queries and around 70,000 shots every year[3]. Each of these video collections is split into a development set and a search set chronologically by source. For each query topic, the relevance judgment on the search set was provided officially by NIST and the judgment on the development set was collaboratively collected by several human annotators using the Informedia client [13]. Although we cannot guarantee all the relevant shots can be found in the development set, this collection effort generally provides a high coverage for the relevance data based on our experience. As the building blocks of the retrieval task, we generated a number of ranking features on the search set including 14 high-level semantic features learned from development data (face, anchor, commercial, studio, graphics, weather, sports, outdoor, person, crowd, road, car, building, motion), and 5 uni-modal retrieval experts (text retrieval, face recognition, image-based retrieval based on color, texture and edge histograms). The detailed descriptions on the feature generation can be found in [13].

We compare four different types of algorithms on all three video collections in Table 1, i.e., logistic regression(LR), ranking logistic regression(RLR), full rank-

[3] Information about these collections can be found at the TRECVID web site [12].

Table 1. Retrieval performance on TRECVID'03 - '05 data. TrainAP is the mean average precision on the development set. TestAP is the mean average precision on the search set. Prec10, Prec30 and Prec100 indicate the mean precisions at the top 10, 30 and 100 retrieved shots on the search set.

Data	Algorithms	TrainAP	TestAP	Prec10	Prec30	Prec100
t05	F-RLR	0.453	0.217	0.535	0.451	0.341
	RLR	0.447	0.217	0.529	0.433	0.341
	LR	0.389	0.207	0.506	0.433	0.341
	NB	0.409	0.204	0.535	0.410	0.334
t04	F-RLR	0.292	0.143	0.269	0.262	0.192
	RLR	0.283	0.141	0.269	0.264	0.192
	LR	0.261	0.132	0.238	0.241	0.184
	NB	0.236	0.129	0.231	0.215	0.182
t03	F-RLR	0.379	0.189	0.433	0.360	0.221
	RLR	0.371	0.186	0.433	0.342	0.224
	LR	0.358	0.185	0.431	0.358	0.229
	NB	0.348	0.181	0.344	0.338	0.230

ing logistic regression(FRLR) which directly optimizes the pairwise risk function in Eqn(2) and naïve Bayes(NB) [1] which is an example of the generative retrieval models. For each algorithm, we learned the combination weights on a per query basis using the development data. To reduce the learning complexity, we choose the top 1000 shots with the highest text retrieval scores as the training examples. The learned models are evaluated based on the same query using the search set. By averaging the performance on all queries, we report the retrieval performance in terms of the mean average precision(MAP) and precision at top 10, 30 and 100 retrieved shots. To guarantee the learning process being supported by sufficient training data, we intentionally removed the queries with less than 10 positive examples in the training process, which typically decrease the query number to around 20 for each data collection. As shown in Table 1, the discriminative models such as LR are usually superior to the generative model, i.e., NB, in terms of both the training/testing MAP on three collections. Among the discriminative models, the ranking versions of LR provide an additional 3-6% boost on the training MAP and 1% boost on the testing MAP compared with LR, which demonstrated the benefits of ranking-based learning in multimedia retrieval. The less significant improvement in the search set is partially due to the insufficiency of the training data for a single query. Since the difference between RLR and LR is not statistically significant, further experiments might be needed to verify the performance improvement of the proposed methods on other information retrieval tasks. Finally, we also observe that RLR, as an efficient approximation of its fully optimization version FRLR, achieved a fairly close performance to FRLR. Their differences on MAP are always less than 1% on three collections, which demonstrates RLR is a reasonable approximation for its fully optimization counterpart with a ten-fold speedup in the learning process.

6 Conclusions

This paper presents a general margin-based rank learning framework for the information retrieval task, which aims to optimize the number of discordant pairs between the predicted ranking and the target ranking rather than minimizing the classification errors. We also propose an efficient approximation for the margin-based rank learning framework which can significantly reduce the computational complexity with a negligible loss in the performance. Both the exact and approximated rank learning algorithms are able to preserve the rank consistency in the data while the binary classification is not. Our experiments on three TRECVID collections demonstrate the superiority of the proposed rank learning algorithms over the generative/discriminative classification algorithms in the context of retrieval tasks. As the future work, we can consider extending the experiments to the other types of loss functions such as the hinge loss function and other scenarios related to ranking optimization such as collaborative filtering and modeling implicit user feedback.

References

1. S. E. Robertson and K. Sparck Jones, "Relevance weighting of search terms," *Journal of the American Society for Informaiton Science*, vol. 27, 1977.
2. R. Nallapati, "Discriminative models for information retrieval," in *Proc. of the 27th SIGIR conf. on information retrieval*, 2004, pp. 64–71.
3. T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the 8th ACM SIGKDD intl. conf. on knowledge discovery and data mining*, New York, NY, USA, 2002, pp. 133–142, ACM Press.
4. Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," in *Proc. of the 15th Intl. Conf. on Machine Learning*, San Francisco, CA, USA, 1998, pp. 170–178.
5. C. Burges and et al., "Learning to rank using gradient descent," in *Proceedings of the 22nd intl. conf. on machine learning*, 2005, pp. 89–96.
6. T.-S. Chua, S.-Y. Neo, H.-K. Goh, M. Zhao, Y. Xiao, and G. Wang, "Trecvid 2005 by nus pris," in *NIST TRECVID-2005*, Nov 2005.
7. J. Gao, H. Qi, X. Xia, and J.-Y. Nie, "Linear discriminant model for information retrieval," in *Proceedings of the 28th international ACM SIGIR conference*, New York, NY, USA, 2005, pp. 290–297, ACM Press.
8. R. Herbrich, T. Graepel, and K. Obermayer., "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, A. J. Smola, P. L. Bartlett, B. Scholkopf, and D. Schuurmans, Eds. MIT Press, 2000.
9. K. Crammer and Y. Singer, "Pranking with ranking," in *Proc. of the Advanced Neural Information Processing Systems (NIPS)*, 2001.
10. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Springer Series in Statistics*, Springer Verlag, Basel, 2001.
11. M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, pp. 181–214, 1994.
12. A.F. Smeaton and P. Over, "TRECVID: Benchmarking the effectiveness of information retrieval tasks on digital video.," in *Proc. of the Intl. Conf. on Image and Video Retrieval*, 2003.

13. A. Hauptmann, M.-Y. Chen, M. Christel, C. Huang, W.-H. Lin, T. Ng, N. Papernick, A. Velivelli, J. Yang, R. Yan, H. Yang, and H. D. Wactlar, "Confounded expectations: Informedia at trecvid 2004," in *Proc. of TRECVID*, 2004.

Appendix

Theorem 1. *The risk minimization estimators λ^* learned from both the margin-based rank learning framework presented in Eqn(2) and the ranking logistic regression algorithm presented in Eqn(7) are consistent with the data ranking.*

Proof: Let us first consider the Eqn(2). When there is a ranking feature f_a satisfies $f_a(d_j, q_t) \geq f_a(d_k, q_t), \forall q_t, \forall d_j \in D_{qt}^+, \forall d_k \in D_{qt}^-$, we can prove λ_a^* is not lower than 0 by contradiction. Assume $\lambda_a^* < 0$ in this case, since $L(\cdot)$ is monotonically decreasing, we can have

$$L\left(\sum_{i\neq a} \lambda_i f_{ijt} + \lambda_a^* f_{ajt}\right) \geq L\left(\sum_{i\neq a} \lambda_i f_{ijt} + (-\lambda_a^*) f_{ajt}\right), \forall j, t \tag{8}$$

where $f_{ijt} = f_i(d_j, q_t) - f_i(d_k, q_t)$ and $f_{ajt} \geq 0$ with at least one $f_{aj't'} > 0$. Therefore, this leads to a contradiction that λ_a^* is a risk minimization estimator. The case of $f_a(d_j, q) \geq f_a(d_k, q)$ can be proved similarly. This complete the proof for Eqn(2).

Next let us consider the Eqn(7). When there is a ranking feature f_a satisfies $f_a(d_j, q_t) \geq f_a(d_k, q_t), \forall q_t, \forall d_j \in D_{qt}^+, \forall d_k \in D_{qt}^-$, we are sure that the optimal $\alpha_i^* \in [\max(f_a(d_k, q_t)), \min(f_a(d_j, q_t))]$, because there are exactly $M_D^+ \cdot M_D^-$ elements larger than $\min(f_a(d_j, q_t))$ and $M_D^+ \cdot M_D^-$ elements smaller than $\max(f_a(d_k, q_t))$ in the union set of the right hand side of Eqn(6). Therefore, for all d_j, the shifted ranking feature $f_{ijt}^* = f_i(d_j, q_t) - \alpha_i^* \geq 0$. Similarly, for all d_k, the shifted ranking feature $-f_{ikt}^* \geq 0$. This recovers to the setting discussed above and thus we can have $\lambda_a^* \geq 0$. The case of $f_a(d_j, q) \geq f_a(d_k, q)$ can be proved similarly. This complete the proof for Eqn(7).

Theorem 2. *If $2L(x/2) \geq L(x)$, the inequality shown in Eqn(3) holds.*

Proof: We first provide a useful lemma as follows as a basis to prove the inequalities: for any $A, B \in \mathcal{R}$, based on the condition of $2L(x/2) \geq L(x)$ and the convexity of L, we can have $L(A) + L(B) \geq 2L(\frac{A+B}{2}) \geq L(A + B)$. On the other hand, we can slightly modify the lemma to be $L(A + B) + L(-A) \geq L(B)$ and $L(A + B) + L(-B) \geq L(A)$. Summing both inequalities together yields, $L(A+B) \geq \frac{1}{2}(L(A)+L(B)-L(-A)-L(-B))$. Next we go ahead to show the inequalities shown in Eqn(3) holds. If we set $A = f^\alpha(d_j, q) = \sum_{i=1}^n \lambda_i[f_i(d_j, q)-\alpha_i]$ and $B = -f^\alpha(d_k, q)$, both lemmas can be rewritten as,

$$\begin{aligned} &L(f^\alpha(d_j, q)) + L(-f^\alpha(d_k, q)) \geq L(f(d_j, q) - f(d_k, q)) \\ &\geq \tfrac{1}{2}[L(f^\alpha(d_j, q)) + L(-f^\alpha(d_k, q)) - L(-f^\alpha(d_j, q)) - L(f^\alpha(d_k, q))] \end{aligned} \tag{9}$$

By summing all of the cases when $\forall q_t, \forall d_j \in D_{qt}^+, \forall d_k \in D_{qt}^-$ on both sides, we can get $RR_{prox}(f) \geq RR'_{reg}(f) \geq \frac{1}{2}[RR_{prox}(f) - RR_{prox}(-f)]$.