A Unified Iterative Optimization Algorithm for Query Model and Ranking Refinement

Yunping Huang[1], Le Sun[1], and Jian-Yun Nie[2]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
{yunping07,sunle}@iscas.ac.cn
[2] Université de Montréal, Montréal, QC, Canada
nie@iro.umontreal.ca

Abstract. Document ranking and query model estimation can be considered as optimization problems. In this paper, we propose an iterative algorithm for optimizing query model and ranking function simultaneously in the context of language model and vector space model, respectively. This algorithm extends the risk minimization framework by incorporating manifold structure of word graph and document graph, and it provides a unified formulation of several existing heuristics for document ranking and query modeling. Moreover, we extend our algorithm by incorporating user's true feedback information, and derive a new ranking model. Experimental results on four TREC collections show that our model is effective.

Keywords: Query Modeling, Document Ranking, Graph-Based Model.

1 Introduction

In order to correctly determine relevant documents for a query, the estimation of query models, the estimation of document models and the definition of document ranking function are three key problems. Document model estimation has been well studied in many studies. In this paper, we address the other two problems together within a unified framework.

Query modeling has attracted much attention in recent years. As the query submitted by a user is usually very short, it needs to be enriched. A commonly used strategy for query modeling is through feedback techniques, such as relevance (pseudo-relevance) feedback [14, 16]. Rocchio feedback formula [14] has been proved to be effective in vector space modeling. In language modeling for information retrieval, two model-based feedback methods have been proposed in [18]: one based on a generative probabilistic model of feedback documents and another on minimization of KL-divergence over feedback documents. Both of them have been proved to be effective. Another commonly used approach to improve query model is to exploit term relations [2]. There are many studies[1,2] that use word relations to improve query models.

Ranking algorithm is another important factor that influences the performance of a retrieval system. In score-based retrieval, a ranking algorithm assigns relevance score

P.-J. Cheng et al. (Eds.): AIRS 2010, LNCS 6458, pp. 514–526, 2010.

to each document with respect to the query. Different methods are defined for estimating relevance scores. Similarity-based methods assume that the relevance is correlated with the similarity/distance between a query and a document [8, 15]. Probabilistic relevance models use a binary random variable to model relevance [7]. Learning to rank is a new family of methods that aims at training a model to rank the document in a supervised learning fashion [5]. However, all the above mentioned models make an assumption that documents are independent. The structure among documents is not taken into consideration. This assumption has been relaxed in several recent studies. [3] utilizes the clustering hypothesis on a graph to smooth the scores and to re-rank the document. Similar studies are described in [10, 12]. These studies demonstrate the advantages of considering the structure among documents. However, query model and ranking function are considered separately in their studies. But in fact, these two elements are strongly dependent. For one thing, query model can be improved with the help of feedback documents. Well ranked documents will be very helpful to refine the query model. For the other thing, query model is a key factor that influences document ranking quality.

In this paper, we propose a unified approach to optimize query model and ranking function simultaneously. We aim to choose the best query model and assign the most appropriate scores to documents. These two elements are optimized together: query model can be improved by well ranked documents, and the ranking function can be improved by a better query model. The previous approaches which fix one of the elements while optimizing another can hit their limit quickly: at some point, it is no longer possible to improve one of the elements (e.g. the ranking function) without also changing the other (query model).

To surpass this limitation, we propose an iterative optimization algorithm: Given a query, a subset D of top-ranked documents in the initial retrieval results and their scores S are used as new evidence to help refine the query model. In turn, the document ranking can be adjusted using the new query model. This process can be done iteratively several times. Different resources, such as word relation, document relation and relevance feedback information, can be incorporated in this process. They are represented as graphs. A constraint onto the smoothness of these graphs will help the optimization algorithm to determine a query model and a ranking function that satisfy the manifold hypothesis: two similar points (terms and documents) in the graph should have similar scores.

In our approach, the above problem is formulated within the risk minimization framework for IR [8]. A unified objective function is defined to integrate multiple criteria: loss function based on risk minimization framework, smoothness of word graphs and document graphs, and fidelity to the original query model. The solution to this optimization problem leads to a new way to define query model and ranking function.

In addition, we extend our approach by incorporating user's true feedback information, which encode the feedback information into a loss function, and combine it with the existing objective function. By minimizing the new objective function, a new ranking method can be derived.

This approach has several potential advantages. First, it can lead to better query model and more appropriate document scores by the unified iterative optimization algorithm which can integrate graph-based optimization into risk minimization framework. Second, the optimization of the query model and document ranking function becomes dependent by systematically incorporating the query term weight and document score as components within the same algorithm. Document scores provide evidence for query model refinement, and query model can influence ranking performance. Once one of them changes, it can be considered as a piece of new evidence for optimizing the other. Finally, it provides a principled general formulation of the query model and ranking function refinement. New resources such as implicit and true relevance feedback information can be flexibly embedded into the approach. Experimental results on four TREC collections show that our model is effective.

The remaining sections are organized as follows. In section 2, we propose a unified iterative optimization algorithm for query model and ranking refinement with graph structure. In section 3, we further integrate user's relevance judgments into our framework. In section 4, we report the experimental results of these methods. We discuss the related work in section 5. We conclude our paper in section 6.

2 Iterative Optimization Algorithm for Query Model and Ranking Refinement

2.1 Risk Minimization Framework

The problem of query model and ranking refinement is essentially an optimization problem. This problem can be formulated within the risk minimization framework for IR [8] by the following loss function:

$$(D^*, S^*, Q^*) = \arg\min_{D,S,Q} \int_{\Theta} L(D,S,Q,\theta)\, p(\theta \mid q,U,C)\, d\theta \tag{1}$$

where $D = \{d_1,\ldots d_n\}$ is a subset of the document collection C; q is a query; $S=\{s_1,\ldots s_n\}$ is the scores of documents in D; Q is the query model; U is a user variable; θ is the set of parameters of the document models; $p(\theta \mid q,U,C)$ is the posterior probability distribution of all the parameters; $L(D,S,Q,\theta)$ is a loss function, which can be defined as follows:

$$L(D,S,Q,\theta) = \sum_{di \in D} Si * \Delta(Q,\theta_{di}) \tag{2}$$

where $\Delta(Q,\theta d)$ is a distance measure between Q and θd, $S(d)$ is the document score of document d. The global risk is defined as a weighted sum of the risk of individual documents. The larger Si is, the larger the contribution of di to the global loss.

In the context of language modeling using KL-divergence, Q and θd are usually represented as unigram language model. This leads to

$$\Delta(Q,\theta d) = \sum_t p(t \mid Q) \log \frac{p(t \mid Q)}{p(t \mid d)} \propto -\sum_t p(t \mid Q) \log p(t \mid d) \tag{3}$$

In the context of vector space modeling, Q and θd are represented as vectors. Within this setting, we have:

$$\Delta(Q, \theta d) = -\sum_t tfidf(q,t) * tfidf(d,t) \tag{4}$$

In this paper, we do not change the document model θ, but focus on the optimization of query model and ranking function. We thus define the following loss function. where θ is the document model we selected.

$$r(D,S,Q) = L(D,S,Q,\theta)$$

2.2 Label Smoothness on Data Graph

Manifold hypothesis states that similar points on the graph are likely to have similar scores. Following this hypothesis, we define a cost function: $G(f)$, which penalizes inconsistency between related points, as follows.

$$G(f) = \sum_{i,j} w_{ij}\left(\frac{f_i}{\sqrt{m_{ii}}} - \frac{f_j}{\sqrt{m_{jj}}}\right)^2 \tag{5}$$

where f_i is the score of point i, W is an affinity matrix, w_{ij} represents the affinity between points i and j and $w_{ii} = 0$, and M is a diagonal normalizing matrix such that $m_{ii} = \sum_j w_{ij}$. This matrix M allows us to normalize the affinity. Such normalization has been shown to result in superior convergence properties than unnormalized affinities for tasks such as spectral clustering [17]. We can use any type of graph to constraint the above function. Points i and j in a graph can be documents, words or users. In this study, we will use word graph and document graph.

2.3 Computing the Optimal Scores

In this section, we introduce our method to refine the query model and ranking function within the context of language modeling and vector space modeling, respectively.

2.3.1 Query Model Refinement

We define the following objective function for query model refinement in the context of language model.

$$\Phi(f) = \frac{1}{2} * (1-\alpha) * \sum_i^n (f_i - y_i)^2 + \alpha * \left(\frac{\beta}{2} * G(f) + (1-\beta) * \left(\frac{r(D,S,f)}{\sum_{d_k \in D} S_k} + \lambda * \sum_i f_i \log P(t_i \mid C) + \frac{1}{2} * \sum_i f_i^2\right)\right) \tag{6}$$

where f and y are query models, $f_i = p(t_i \mid Q)$, $y_i = p_{ml}(t_i \mid Q)$; α, β and $\lambda \in [0,1)$ are weighting parameters. The first term guarantees that the refined language model does not deviate too much from its original value; $G(f)$, also known as harmonic function in semi-supervised learning [19], guarantees the consistency of the query model on the graph; $r(D,S,f)$ represent risk function over feedback document, which is defined in section 2.1, where D and S are the feedback documents and the corresponding scores; $p(t|C)$ is the collection language model; $\sum_i f_i \log P(t_i \mid C)$ measures how different the query model is from the collection language model, and $\sum_i f_i^2$ is a regularization term.

In general, to minimize the objective function in Equation 6, we can compute the first-order partial derivative of it, which is

$$\frac{\partial\Phi(f)}{\partial f_i}=(1-\alpha)\times(f_i-y_i)+\alpha\times(\beta\times\sum_j^n\frac{w_{ij}}{\sqrt{m_{ii}}}(\frac{f_i}{\sqrt{m_{ii}}}-\frac{f_j}{\sqrt{m_{jj}}})+(1-\beta)\times(\lambda\times\log p(t_i\mid C)-\sum_{d_j\in D}\frac{S_j}{\sum_{d_k\in D}S_k}\times\log P(t_i\mid\theta_d)+f_i))$$

Using the gradient decent method to optimize the objective function, let

$$f_i^{t+1}=f_i^t-\frac{\partial\Phi(f)}{\partial f_i}$$

we have

$$f_i^{t+1}=(1-\alpha)\times y_i+\alpha\times(\beta\times\sum_j^n\frac{w_{ij}\times f_j^t}{\sqrt{m_{ii}}\times\sqrt{m_{jj}}}+(1-\beta)\times(\sum_{d_j\in D}\frac{S_j}{\sum_{d_k\in D}S_k}\times\log P(t_i\mid\theta_d)-\lambda\times\log P(t_i\mid C))) \tag{7}$$

From Equation 7, we can see that the new query model f_i has three components, corresponding to the three objective terms in equation 6: The first term is the original query model, the second term is obtained through score regularization on the graph, and the third term is obtained from the document model of feedback documents which is similar to the divergence minimization feedback model in [18]. The difference between the above formulation and that in [18] is that our approach considers the document score S_j as a component.

Similar to the above language model, we define another objective function for vector space model:

$$\Phi(f)=\frac{1}{2}\times(1-\alpha)\times\sum_i^n(f_i-y_i)^2+\alpha\times(\frac{\beta}{2}\times G(f)+(1-\beta)\times(\frac{r(D,S,f)}{\sum_{d_k\in D}S_k}+\frac{1}{2}\times\sum_i f_i^2)) \tag{8}$$

where f and y are two vectors, $y_i=tfidf(q,t_i)$ is the *tfidf* weight of the term t_i in query q, and the other parameters are the same as in equation 6. We do not use the collection model in this objective function, because the *idf* value can play a similar role. Using the same manipulation as in language modeling, we have

$$f_i^{t+1}=(1-\alpha)\times y_i+\alpha\times(\beta\times\sum_j^n\frac{w_{ij}\times f_j^t}{\sqrt{m_{ii}}\times\sqrt{m_{jj}}}+(1-\beta)\times(\sum_{d_j\in D}\frac{S_j}{\sum_{d_k\in D}S_k}\times tfidf(d,t_i))) \tag{9}$$

2.3.2 Ranking Refinement

In score-based retrieval, the retrieval system assigns each document a relevance score. Document ranking can be considered as a decision problem, which is optimized so as to minimize a loss function within the risk minimization framework. Manifold hypothesis implies that neighbor documents on document graph should have similar scores. So we define the following loss function for ranking refinement as follows:

$$\Phi(S)=a*(r(D,S,Q)+\frac{1}{2}\sum_i S_i^2)+\frac{1}{2}*b*G(S) \tag{10}$$

where a and b are two weighting parameters with $a+b=1$; S_i is the score of document d_i. The first term is derived from risk minimization framework, Q is the query model;

$r(D,S,Q)$ is the loss function defined in 2.1; and D is the set of top-ranked documents from the initial retrieval results. The last term guarantees the smoothness of the document scores on the graph; $\sum_i S_i^2$ is a regularization term.

Using the gradient decent method,

$$S_i^{t+1} = S_i^t - \frac{\partial \Phi(S)}{\partial S_i} = a * x_i + b * \sum_j w_{ij} * \frac{S_j^t}{\sqrt{m_{ii}} * \sqrt{m_{jj}}} \tag{11}$$

$$x_i = -\Delta(Q, \theta_{di})$$

where $\Delta(Q, \theta_{di})$ is defined in Equations 3 or 4 for language model or vector space model respectively.

In order to normalize the value of x_i we replace x_i in Equation 11 by the following normalized value:

$$x_i = \frac{x_i - \min\{x_j,\ j = 1,...N\}}{\max\{x_j,\ j = 1,...N\} - \min\{x_j,\ j = 1,...N\}}$$

2.3.3 The Iterative Optimization Algorithm for Query Model and Ranking Refinement

We observe that the query model Q is a component for refining document scores, and the document scores s_d is also a component for refining query model. We see clearly that these two elements mutually influence each other. The best solution for one element is condition to the other element. A way to solve this problem is to use an iterative algorithm to optimize each of them in turn. This process is expected to improve the query model and ranking iteratively and simultaneously.

We derive two models here: one in the context of language modeling using KL-divergence, denoted by **IOA-KL** (Iterative Optimization Algorithm-KL); another in the context of vector space model using TF-IDF weighting, denoted by **IOA-TFIDF** (Iterative Optimization Algorithm -TFIDF). To obtain a more precise and smoothed query model and document score model, IOA-KL and IOA-TFIDF start with $f = y$, the initial document scores assigned in the initial retrieval step. The algorithm alternates the optimization of query model and ranking model until convergence. The iterative optimization algorithm is defined as follows:

The Iterative Algorithm for Query Model and Ranking Refinement

1. Use the initial retrieval model to obtain the top ranked documents and their scores.
2. Use equation 7 (IOA-KL) or 9 (IOA-TFIDF) to compute the new query model.
3. Use equation 11 to compute the new document scores.
4. Compute the change c between two iterations as follows:

   $$c = \sum_i (S_i^{t+1} - S_i^t)^2$$

   If c is smaller than some predefined value or the number of iterations reaches the predefined limit, the algorithm stops. Otherwise, we go to step 2.

For the sake of efficiency, we limit the gradient decent in steps 2 and 3 to only one iteration in our implementation. The above iterative algorithm stops when the change of document scores is smaller than some predefined threshold ((e.g. 10^{-6}). In practice, we do not need to wait till complete convergence. A few iterations can already give an improved query model and ranking. So we set the maximum iteration number to 5.

2.3.4 Construction of the Data Graph

For every query, we construct a k-Nearest-Neighbor (kNN) document graph of all initial documents. The weight of the edge connecting two documents is measured with cosine similarity, which is the same as in [10]. We also construct a word graph for the query model. The weight of an edge is set to be the mutual information between the two words. In order to control the scale of the word graph, in each iteration, the word graph is based on the query model obtained from the last iteration, which only includes the terms in the query model obtained from the last iteration.

3 Considering True Relevance Feedback

The models defined in the previous sections rely on a word graph or a document graph for optimization. The document graph can be constructed using the top *N* retrieved results. In this section we consider a particular case in which we have true user relevance feedback. True relevance feedback is known to be effective [14, 16]. Previous work using true relevance feedback focuses on query updating such as query term re-weighting and query expansion. Here, we consider such information as encoding preference orders between documents.

When a user provides true relevance judgments for the top *N* results of the initial retrieval, the relative relevance of each document pair inside the judged set can be obtained. We encode it within a matrix *R* as follows.

$$R_{ij} = \begin{cases} 1 & d_i \succ d_j \\ 0 & otherwise \end{cases}$$

It is similar to the pairwise relationship in learning to rank approach [5]. In [5], the relative relevance information from the implicit feedback information (clickthrough information) is used to train a model. Previous studies showed that relative relevance preferences can be derived more reliably from implicit feedback than absolute relevance [6].

The matrix *R* is then normalized as follows to make the sum of each row equal to 1:

$$R_{ij} = \begin{cases} R_{ij} / \sum_{j'} R_{ij'} & \sum_{j'} R_{ij'} \neq 0 \\ 0 & otherwise \end{cases}$$

A cost function *F* can then be defined as follows:

$$F(S) = \sum_i \sum_j R_{ij} * e\mathrm{xp}(S_j - S_i)$$

This cost function is similar to the loss function defined in [4] [13]. The element $\exp(S_j - S_i)$ only contributes when its corresponding R_{ij} is not zero, which means that document i is more relevant than document j. In that case, if document i has a larger score than document j, then the value of the cost function will be small; in contrast, if document i has a smaller score than document j, then the value of the cost function will be large. Approximating $e\mathrm{xp}(S_j - S_i)$ using the Taylor expansion, we have:

$$F(S) = \sum_i \sum_j R_{ij}(1 + S_j - S_i + \frac{1}{2}(S_j - S_i)^2) \tag{12}$$

Appending this cost function to the existing objective function (10) discussed in section 2.3.2 leads to a new objective function.

$$\Phi(S) = a * (r(D,S,Q) + \frac{1}{2}\sum_i S_i^2) + \frac{1}{2} * b * G(S) + c * F(S) \tag{13}$$

Using the decent gradient algorithm, we have

$$\begin{aligned} S_i^{t+1} &= S_i^{t+1} - \frac{1}{Z_i} * \frac{\partial \Phi(S)}{\partial S_i} = \frac{a}{Z_i} * x_i + \frac{b}{Z_i} * \sum_j w_{ij} * \frac{S_j^t}{\sqrt{m_{ii}} * \sqrt{m_{jj}}} \\ &+ \frac{c}{Z_i} * (\sum_j R_{ij}(1 + S_j^t) + \sum_j R_{ji}(S_j^t - 1)) \\ Z_i &= a + b + c * (\sum_j R_{ij} + \sum_j R_{ji}) \end{aligned} \tag{14}$$

where $a+b+c=1$; x_i is the same as defined in section 2, and it is also normalized. Z_i is used to control the learning rate.

Based on IOA-KL, IOA-TFIDF and the relevance feedback information, we derive two new models: IOA-KL+REL and IOA-TFIDF+REL. The new models only change the method to re-compute the document scores.

4 Experiments

In Section 2, we introduce our algorithm and two instantiations: IOA-KL in language modeling and IOA-TFIDF in vector space modeling. In Section 3, we extend our approach by integrating true relevance feedback information, and derive two further models: IOA-KL+REL and IOA-TFIDF+REL, which incorporate relative relevance feedback information into IOA-KL and IOA-TFIDF respectively. We will evaluate the effectiveness of these methods empirically in this section.

In this study, we fix the document model. Specifically, in IOA-KL, we use a Dirichlet prior (with a setting of 600) to estimate the document language models. In IOA-TFIDF, we use the TFIDF value to weight terms, The TF formula used is the one based on the BM25 retrieval formula as in [18].

4.1 Experimental Setup

We evaluate the proposed method over four TREC data sets: AP (Associated Press news 1988-90), LA (LA Times), SJMN (San Jose Mercury News 1991), and TREC8 (the ad hoc data used in TREC8). They are identical to the data sets used in [10], and are preprocessed in the same way. We used the title field of a query/topic description to simulate short keyword queries.

Because our methods need an initial retrieval document set, the Lemur toolkit[1] is used to retrieval the initial 3000 documents for each query. Our methods will be used to re-rank these documents. For IOA-KL, KL-divergence based on language model using Dirichlet smoothing strategy is selected as our initial retrieval model. For IOA-TFIDF, vector space model using BM25 TF is used as the initial retrieval model. When updating the query model, we set the upper bound of the number of terms to 20, and ignored all terms having a weight less than 0.001.

In all our experiments, the cutoff of relevant documents is set to 1000. The following two performance measures are used in our evaluation: (1) non-interpolated Mean Average Precision (MAP). (2) Precision at 10 documents (P@10).

4.2 Parameter Selection

Our methods for query model and ranking refinement contain a set of parameters: α and β in Equation 7 and 9 to balance the importance of each component; and the parameter b in Equation 11. In order to determine the values of these parameters, we use the AP collection as our training collection. We performed an exhaustive grid search to tune the three parameters on the training set, and then used the parameters on other test collections. The three parameters were swept over [0, 1] with a step size of 0.1. For IOA-KL+REL and IOA-TFIDF+REL, we will have five main parameters: { α , β a, b, c }. When searching for the optimal parameters, α , β were swept over [0, 1] with a step size of 0.1; for *a,b,c*, the parameter *c* was swept over [0,1] with a step size of 0.1; since *a*+*b*+*c*=1, we set $a = \sigma \times (1-c)$, $b = (1-\sigma) \times (1-c)$, σ was swept over [0,1] with a step of 0.1. We select the parameter values which optimize the MAP. The number of neighbors is set to 60 through the grid search method.

4.3 The Effectiveness of the Iterative Optimization Algorithm

In order to see the effect of our proposed methods, we compare these two methods with other feedback model. IOA-KL is compared with KL-Divergence minimization feedback (DIV-MIN), which is a model-based feedback method used in [18]. IOA-TFIDF is compared with Rocchio feedback method [14] based on vector space model. For DIV-MIN feedback model, we varied the two main parameters: the coefficient that controls the influence of the feedback model and the noise parameter that controls the influence of the collection model. For Rocchio feedback, the TF formula used is the one based on the BM25 retrieval formula, which is the same as the TF formula used in [18]. We vary the coefficient in Rocchio formula to tune the result to its best.

In Table 1, we can see that IOA-KL outperforms DIV-MIN FB consistently and significantly in most cases when using the top 10 documents for pseudo-feedback. The increase in map is between 6% and 10% in most cases. IOA-TFIDF can also outperform Rocchio feedback model in most cases.

---

[1] http://www.lemurproject.org/

Table 1. Basic results. Top 10 documents are selected as feedback documents. ***,**, and * mean significant improvement in paired t-test at the level of p<0.01, p<0.05 and p<0.1, respectively. For IOA-KL, the optimal values of α , β and b are 0.5, 0.8 and 0.3 respectively; for IOA-TFIDF, the optimal values of α , β and b are also the same.

Data		DIV-MIN FB	IOA-KL	Rocchio FB	IOA-TFIDF
AP88~90	MAP	0.2524	0.2763(+9.47%***)	0.2580	0.2741(+6.24%***)
	p@10	0.4525	0.4828(+6.70%**)	0.4303	0.4646 (+7.97%**)
LA	MAP	0.2505	0.2671 (+6.63%)	0.2455	0.2738(+11.53%**)
	p@10	0.2847	0.2888 (+1.44%)	0.2755	0.3061(+11.11**)
SJMN	MAP	0.2310	0.2490(+7.79%***)	0.2497	0.2486 (-0.44%)
	p@10	0.3340	0.3649 (+9.25%**)	0.3734	0.3691(-1.15%)
TREC8	MAP	0.2622	0.2782 (+6.10%*)	0.2609	0.2752 (+5.48%*)
	p@10	0.4600	0.4580(-0.43%)	0.4540	0.4580 (+0.88%)

We also compared IOA-KL with other language models: the basic KL-Divergence model with Dirchlet smoothing (KL Dir), the mixture feedback (Mix FB) model[1] ; FB+QMWG proposed in [10], a graph-based query model smoothing method combined with mixture feedback. In order to evaluate the impact of considering smoothness in word and document graphs, we remove the smoothness constraint from IOA-KL model. This simplified model is denoted by IOA-KL*.

Table 2. Performance (MAP) comparison with related methods. The result of FB+QMWG is from [10]. For the methods need feedback documents, top 5 documents are used as in [10].

	KL Dir	Mix FB	FB+QMWG	IOA-KL*	IOA-KL
AP88~90	0.217	0.266	0.273	0.260	**0.276**
LA	0.247	0.257	0.267	0.266	**0.271**
SJMN	0.204	0.241	0.246	0.241	**0.251**
TREC8	0.257	0.278	0.280	0.271	**0.281**

In Table 2, comparing IOA-KL with KL Dir, we can see that the increase in effectiveness is very large. IOA-KL can also consistently outperform other existing methods. This indicates that unifying multiple resources such as document graph, word graph can help improve the retrieval results. Comparing IOA-KL to IOA-KL*, when the smoothness criterion is removed from the loss function, the retrieval effectiveness decreases. This demonstrates the importance of graph smoothness.

4.4 Effect of Considering Relevance Feedback Information

In section 3, we derive a new optimization algorithm by appending relevance feedback information, and two new document ranking refinement methods **IOA-KL+REL** and **IOA-TFIDF+REL** are derived. We will test the performance of these two models. This experiment simulates a situation where a user accepts to provide relevance judgments for a small set of documents.

In our experiment, we provide true relevance judgments for the top 10 documents. Thus we can get relative relevance between each document pair among the top 10 documents, and encode them in the matrix R described in section 3.

To evaluate the effect of incorporating true relevance feedback information, we compare IOA-KL+REL with DIV-MIN relevance feedback (DIV-MIN+REL) model, which also uses the relevance judgments of the top 10 documents for feedback. In the same way, IOA-TFIDF+REL is compared with the Rocchio relevance feedback (Rocchio REL) model. For KL+REL and DIV-MIN REL model, we use the same initial retrieval method to obtain a document set for each query, and then provide relevance judgments for the top 10 documents. The same processing method is used by IOA-TFIDF+REL and Rocchio REL method.

Table 3. Comparing IOA-KL+Rel and IOA-TFIDF+Rel with other relevance feedback model. For IOA-KL+REL and IOA-TFIDF+REL, the optimal parameters are α =0.6, β = 0.8, c=0.7, and a:b=7:3 respectively.

Data		DIV-MIN+REL	IOA-KL+REL	Rocchio+REL	IOA-TFIDF+REL
AP88~90	MAP	0.2672	0.3107(+16.28%***)	0.2973	0.3053(+2.69%)
	p@10	0.5010	0.5970(+19.16%***)	0.5526	0.5788(+4.74%**)
LA	MAP	0.3356	0.3873(+15.41%***)	0.3741	0.3855 (+3.05%**)
	p@10	0.3561	0.3837 (+7.75%**)	0.3714	0.3929 (+5.79%*)
SJMN	MAP	0.2768	0.3097(+11.89%***)	0.3087	0.3120(+1.07%***)
	p@10	0.4160	0.4766(+14.57%***)	0.4617	0.4743(+2.73%***)
TREC8	MAP	0.2974	0.3375(+13.48%***)	0.3170	0.3409 (+7.54%**)
	p@10	0.5400	0.6280(+16.30%***)	0.5920	0.6140(+3.72%**)

Table 3 shows that our IOA-KL+REL model can outperform DIV-MIN REL consistently and significantly on all the four datasets. The IOA-TFIDF+REL method can also outperform Rocchio REL. A possible explanation to this comparison is that, in traditional relevance feedback model, the relevance feedback information is used only to update query model, and then the updated query model is used to re-rank the documents. When re-ranking the documents, the same ranking function is used, i.e. the relevance judgments did not affect the general ranking function. However, in our model, we interpret the relevance feedback information as relative relevance of document pair. It can be used as a constraint when re-ranking the documents. Thus our method can makes thorough use of the relevance feedback information.

5 Related Work

Risk minimization [8] provides a general framework to further improve a retrieval model. Formally, it treats the task of information retrieval as a statistical decision problem. It can unify several existing retrieval models such as KL-divergence language model and vector space model. KL-Divergence minimization feedback model described in [18] can be considered as one of its instantiation. Our approach is based on risk minimization framework. However, we take the graph structure into consideration, also, our approach can contain document scores as a component, making it possible to use the document score as evidence to help update query model. This allows us to unify query model and document ranking refinement within the same framework.

Graph-based learning has attracted much attention in recent years [3, 9, 19]. Manifold ranking method is proposed in [19]. Our cost function on graph structure is similar to the objective function proposed in [19]. In information retrieval, there are also some investigations using graph structures [3, 10]. These studies focused on the following aspects, however, separately - document score regularization or query model. In contrast, our approach combines graph structure into the risk minimization framework. This makes it easy to unify the refinements of query model and ranking function. We combined both word graph and document graph in our approach, which can lead to a smoothed query model and ranking function. Through the iterative process, we can make use of the mutual influence between the query model and the document ranking in order to optimize both of them.

One of the concrete instantiations of our algorithm is proposed in section 3, which is related to some learning to rank approaches [4, 13]. [4] uses user feedback information to refine ranking. [13] utilizes document relations to obtain a ranking function. Our method also encodes user feedback into document preference relations, and combines them with other cost function to form a unified objective function. It is also related to the problem of relational object ranking. Our method takes the label smoothness on the graph into consideration. This enables user's relevance judgments to propagate in the graph. In addition, the iterative optimization process further enables the relevance information to propagate between query model and document ranking.

6 Conclusions

In this paper, we proposed a unified iterative optimization algorithm which provided a principled way to refine both the query model and the document ranking. It combines the risk minimization framework and the manifold structure of both the word graph and the document graph. The combination of these multiple criteria forms a new optimization problem. The solution to the optimization problem leads to some new methods. We derived two methods in the context of language model and vector space model respectively. Moreover, we extended the algorithm by combining user's true relevance judgments.

The approach has several potential advantages. First, this approach can lead to better query model and more appropriate document scores by our proposed unified iterative optimization algorithm. Second, the optimization of the query model and document ranking function becomes dependent by systematically incorporating the query model and document score as components within the same framework. Finally, it provides a principled formulation of the query model and ranking refinement, which can serve as a general framework for systematically exploring other methods.

The evaluated results on four test collections show that these methods are more effective than the existing ones in the literature. In our future work, we will consider the problem of parameter optimization as well as the problem of document model refinement. Moreover, we can use extend the optimization algorithm by exploring other kind of retrieval model such as dependent ranking model.

Acknowledgements

This work was supported by the National Science Foundation of China (Grants No.60736044, 60773027, 90920010), as well as 863 Hi-Tech Research and Development Program of China (Grants No. 2008AA01Z145, 2006AA010108).

References

1. Bai, J., Nie, J., Bouchard, H., Cao, G.: Using Query Contexts in Information Retrieval. In: Proceedings of SIGIR, pp. 15–22 (2007)
2. Cao, G., Nie, J., Bai, J.: Integrating word relationships into language models. In: Proceedings of SIGIR, pp. 298–305 (2005)
3. Diaz, F.: Regularizing ad hoc retrieval scores. In: CIKM, pp. 672–679 (2005)
4. Jin, R., Valizadegan, H., Li, H.: Ranking refinement and its application to information retrieval. In: Proceedings of WWW, pp. 397–406 (2008)
5. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD, pp. 133–142 (2002)
6. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of SIGIR, pp. 154–161 (2005)
7. Sparck Jones, K., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments—part 1and part 2. Information Processing and Management, 36(6), 779–808 (2000)
8. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: Proceedings of SIGIR, pp. 111–119 (2001)
9. Mihalcea, R., Radev, D.R.: TextRank – bringing order into texts. In: EMNLP. pp. 404–411 (2004)
10. Mei, Q., Zhang, D., Zhai, C.: A General Optimization Framework for Smoothing Language Models on Graph Structures. In: Proceedings of SIGIR, pp. 611–618 (2008)
11. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of SIGIR 1998, pp. 275–281 (1998)
12. Qin, T., Liu, T.-Y., et al.: A study of relevance propagation for web search. In: Proceedings of SIGIR, pp. 408–415 (2005)
13. Qin, T., Liu, T.-Y., et al.: Learning to Rank Relational Objects and Its Application to Web Search. In: Proceedings of WWW 2008, pp. 407–416 (2008)
14. Rocchio, J.J.: Relevance feedback in information retrieval. In: The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313–323 (1971)
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24, 513–523 (1988)
16. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science 41(4), 288–297 (1990)
17. Von Luxburg, U., Bousquet, O., Belkin, M.: On the convergence of spectral clustering on random samples: The normalized case. In: Shawe-Taylor, J., Singer, Y. (eds.) COLT 2004. LNCS (LNAI), vol. 3120, pp. 457–471. Springer, Heidelberg (2004)
18. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of ACM CIKM, pp. 403–410 (2001)
19. Zhou, D., Weston, J., Gretton, A., Bousquet, O., Scholkopf, B.: Ranking on data manifolds. In: Proceedings of NIPS, pp. 169–176 (2003)