Exploiting Ranking Factorization Machines for Microblog Retrieval

Runwei Qiang
qiangrw@pku.edu.cn

Feng Liang*
liangfeng@pku.edu.cn

Jianwu Yang†
yangjw@pku.edu.cn

Institute of Computer Science and Technology
Peking University, Beijing 100080,China

ABSTRACT

Learning to rank method has been proposed for practical application in the field of information retrieval. When employing it in microblog retrieval, the significant interactions of various involved features are rarely considered. In this paper, we propose a Ranking Factorization Machine (Ranking FM) model, which applies Factorization Machine model to microblog ranking on basis of pairwise classification. In this way, our proposed model combines the generality of learning to rank framework with the advantages of factorization models in estimating interactions between features, leading to better retrieval performance. Moreover, three groups of features (*content relevance features*, *semantic expansion features* and *quality features*) and their interactions are utilized in the Ranking FM model with the methods of stochastic gradient descent and adaptive regularization for optimization. Experimental results demonstrate its superiority over several baseline systems on a real Twitter dataset in terms of P@30 and MAP metrics. Furthermore, it outperforms the best performing results in the TREC'12 Real-Time Search Task.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval—*Retrieval models*

Keywords

Learning to Rank; Microblog Retrieval; Ranking FM; Optimization Method

1. INTRODUCTION

Microblog, a popular form of social media service, provides people a convenient way to post and share their activities, emotions and statuses [6]. In Twitter, people can post short messages limited within 140 characters. Besides, shortened URLs and some common signs (e.g. '@', '#' and 'RT') can be embedded in a tweet for further content representation or user interaction. To explore information retrieval (IR) in microblogging environment such as Twitter, TREC firstly introduced a Real-Time Search Task in 2011 [11], which doesn't mean that the search results are simply ranked chronologically. It means that an information need arrives at a specific time and concerns something happening right now [15].

IR for microblog is a non-trivial problem which involves various kinds of factors, such as query expansion, link-based document expansion and social network analysis. However, a single retrieval model can hardly utilize these factors perfectly at the same time. Recently, learning to rank has become one of the most active research topics in IR [9] and a great number of features have been proved useful in IR for microblog [2]. However, there still exists much space for improvement in feature utilization of prior work. First, some features with the potential of enhancing the search efficiency are not fully exploited. For instance, as a crucial factor in IR, the link feature is usually treated as a binary one by which we can in fact obtain more semantic information from the linked page. Second, features are considered independent when applied to most learning to rank approaches while it cannot be neglected that some features are closely related to each other. For instance, RT and @ symbols occur in the same tweet frequently, revealing their intrinsic relationship. However, existing models which consider nested feature interactions are inefficient and not quite flexible such as the non-linear SVM model with polynomial kernel.

To remedy the above drawbacks, we propose employing an optimization for Ranking FM to improve the performance of system for Real-Time Search Task, which applies Factorization Machines (FM) model [13] to microblog ranking on basis of pairwise classification. The main contributions in this paper are concluded as follows: (1) we employ Ranking FM, which adopts FM as the ranking function to model interactions between features, and apply it to pairwise learning to rank approach; (2) we utilize several effective features which are neglected in existing work to boost the microblog retrieval performance; (3) we optimize the Ranking FM by two methods of stochastic gradient descent and adaptive regularization. The proposed approach is analyzed empirically on the Tweet11 corpus used in TREC'11 and TREC'12 Microblog Track. Experimental results indicate that Ranking FM outperforms several baseline systems as well as the best performed system in the TREC'12 Real-Time Search Task.

*First two authors contribute to this work equally.

†Corresponding author.

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2505515.2505648 .

The remainder of this paper is organized as follows. We give an overview of the related work in Section 2. In Section 3, we first introduce the Factorization Machines [13], and then describe our Ranking FM approach to deal with the microblog retrieval problem, followed by two optimization methods to estimate Ranking FM model parameters. Section 4 presents three categories of features used in our method. In Section 5, experimental results and comparison are presented in detail. Finally, we conclude the paper in Section 6.

2. RELATED WORK

Recently, more and more machine learning technologies have been used to train the ranking model, and learning to rank has become one of the most active research topics in IR [9]. Joachims *et al.* [7] applied Ranking SVM to optimize the retrieval quality of search engines with users' click-through data. Besides, Cao *et al.* [1] adapted Ranking SVM to document retrieval by modifying the loss function.

Factorization Machines, a new model class combining the advantages of SVM with factorization models, has been proposed to model all nested variable interactions [12]. Factorization Machines are able to work with any real-value feature vector, and it also can mimic most state-of-the-art factorization models by feature engineering. As a general predictor, Factorization Machines can be applied to a variety of prediction tasks including regression, binary classification and ranking.

Many attempts have been made in exploiting IR in the microblogosphere. Massoudi *et al.* [10] proposed incorporating query expansion and quality indicators in microblog retrieval. In their work, quality indicators such as emoticons, post length, shouting, capitalization, hyperlinks, reposts, followers and recency are taken into consideration to build the retrieval model. An integrated retrieval model under language model framework is presented in [8]. The proposed approach described a two-stage pseudo-relevance feedback query expansion to estimate the query language model and proposed two ways to expand document with shortened URLs in microblog. In [2], Duan *et al.* employed learning to rank algorithms to determine the best set of features. Han *et al.* [3] adopted query expansion, document expansion and learning to rank technique to fuse the scores of the tweet text and the linked URL to improve retrieval performance. Specifically, they used a logistic regression model to learn a pairwise ranking for twitter retrieval. However, neither of Duan and Han took the pair interactions between features into consideration.

3. RANKING FACTORIZATION MACHINES FOR MICROBLOG RETRIEVAL

In this section, we first briefly introduce Factorization Machines. Then we present our Ranking FM framework which incorporates learning to rank approach with Factorization Machines. At last, two optimization methods are conducted: stochastic gradient descent and adaptive regularization.

3.1 Factorization Machines

Factorization Machine (FM), proposed by Rendle in [12], models all nested interactions up to order d among n input variables in $\mathbf{x}$ using factorized interaction parameters. The FM model of order $d = 2$ is defined as:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i \, x_j \qquad (1)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \qquad \mathbf{w} \in \mathbb{R}^n, \qquad \mathbf{V} \in \mathbb{R}^{n \times k}$$

And$\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \qquad (2)$$

A row vector $\mathbf{v}_i$ of $\mathbf{V}$ represents the i-th variable with k factors. $k \in \mathbb{N}_0^+$ is a hyper-parameter that defines the factorization dimensionality. A 2-way FM captures all single and pairwise interactions between variables with w_0 as the global bias. The strength of the i-th variable is measured by w_i and $\hat{w}_{i,j} := \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ models the interaction between the i-th and j-th variables. Instead of using an own model parameter $w_{i,j} \in \mathbb{R}$ for interaction estimation, FM models the interaction by factorizing it, which allows high quality parameter estimates of high-order interactions under sparsity.

In [12], it proves that FM (eq.(1)) can be computed efficiently with the computational complexity of $O(k \cdot n)$ as it is equivalent to:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right) \qquad (3)$$

3.2 Ranking FM Framework

FM can be applied to a variety of prediction problems with variables whose interactions are hard for estimation. We can realize a Ranking FM approach by incorporating the learning to rank approach with FM as follows. Assume that there exists an input space $X \in \mathbb{R}^n$, where n is the number of features. Meanwhile there is an output space of ranks/categories represented by labels $Y = \{r_1, r_2, \cdots, r_q\}$ with the number of ranks q. They keep a fixed order as $r_q \succ r_{q-1} \succ \cdots \succ r_1$, where $\succ$ represents a preference relation. A family of ranking functions $f \in F$ exist and each of the candidate function can determine the preference relations between instances:

$$\mathbf{x}^{(i)} \succ \mathbf{x}^{(j)} \Leftrightarrow f(\mathbf{x}^{(i)}) > f(\mathbf{x}^{(j)}) \qquad (4)$$

Suppose that we are given a set of ranked instances $S = \left\{ \left(\mathbf{x}^{(i)}, y^{(i)} \right) \right\}_{i=1}^{t}$ from the space $X \times Y$ where $y^{(i)}$ is a rank with preference. The task here is to select the best function $f^* \in F$ that minimizes the loss function for the given ranked instances. This problem is formalized as learning for classification *on pairs of instances* by Herbrich *et al.* [4]. To incorporate FM into pairwise learning to rank approach, we assume that f is represented with FM function ($d = 2$):

$$f_\Theta(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j \sum_{f=1}^{k} v_{i,f} v_{j,f} \qquad (5)$$

Next, we take any instance pair and their relation to create a new instance with a new label. Let $\mathbf{p}$ and $\mathbf{q}$ denote the

first and second instances in the pair instance, and y_p and y_q denote their ranks, then we have:

$$\left((\mathbf{p},\mathbf{q}), z = \begin{cases} +1 & y_p \succ y_q \\ -1 & y_q \succ y_p \end{cases}\right) \tag{6}$$

In this way, from a given training data set S, we create a new set $S' = \{(\mathbf{p}^{(t)},\mathbf{q}^{(t)}), z^{(t)}\}_{i=1}^{l}$ containing l labeled instances. Next, we can calculate the empirical Hinge Loss by the t-th instance pair of S', where subscript "+" indicates the positive part:

$$l_i(f;\mathbf{p}^{(t)},\mathbf{q}^{(t)},z) = [1 - z \times (f_\Theta(\mathbf{p}^{(t)}) - f_\Theta(\mathbf{q}^{(t)}))]_+ \tag{7}$$

The difference of $f_\Theta(\mathbf{p}^{(t)})$ and $f_\Theta(\mathbf{q}^{(t)})$ can be computed intuitively with the computational complexity of $O(k \cdot n)$ by applying eq.(3).

$$\begin{aligned} f_\Theta(\mathbf{p}^{(t)}) - f_\Theta(\mathbf{q}^{(t)}) := & \sum_{i=1}^{n} w_i(p_i - q_i) \\ & + \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}p_i\right)^2 - \left(\sum_{i=1}^{n} v_{i,f}q_i\right)^2\right) \\ & - \frac{1}{2}\sum_{f=1}^{k}\left(\sum_{i=1}^{n} v_{i,f}^2 p_i^2 - \sum_{i=1}^{n} v_{i,f}^2 q_i^2\right) \end{aligned} \tag{8}$$

Now, we define a global loss function over all training data S' on the basis of Hinge loss,

$$\min_{\Theta} L(\Theta) = \sum_{t=1}^{l} l_t(f;\mathbf{p}^{(t)},\mathbf{q}^{(t)},z^{(t)}) + \sum_{\theta\in\Theta}\lambda_\theta\theta^2 \tag{9}$$

where λ_θ is a regularization (hyper-)parameter for the model parameter θ. Theoretically, the regularization term can be chosen individually for each model parameter. However, in practical cases, it makes sense to use the same regularization parameters for similar model parameters (i.e. $\lambda_{\mathbf{w}}$ for w_i and $\lambda_{\mathbf{v}}$ for $v_{i,f}$).

From the framework of Ranking FM, we can see that: (1) it is capable of modeling all nested variable interactions which are neglected in linear ranking model like Ranking SVM with linear kernel. (2) compared with other non-linear ranking models, its computational complexity is reduced to $O(k \cdot n)$. Considering that $k \ll n$ satisfies in most scenarios, we can estimate Ranking FM efficiently in linear time.

3.3 Optimization Methods

In this section, we present two methods to optimize the loss function in eq.(9). The two learning algorithms are stochastic gradient descent and adaptive regularization, both of which have been used to optimize FM [13, 14] and achieved good performance.

3.3.1 Stochastic Gradient Descent

By differentiating eq.(9) with respect to parameter θ, we can obtain:

$$\frac{\partial L}{\partial\theta} = \sum_{t=1}^{l}\frac{\partial l_t(f;\mathbf{p}^{(t)},\mathbf{q}^{(t)},z^{(t)})}{\partial\theta} + 2\lambda_\theta\theta \tag{10}$$

$$\frac{\partial l_t}{\partial\theta} = \begin{cases} 0 & \text{if } \xi(t) \geq 1 \\ -z^{(t)}(p_i^{(t)} - q_i^{(t)}) & \text{if } \theta \text{ is } w_i \\ -z^{(t)}\left(G(p_i^{(t)}, q_i^{(t)})\right) & \text{if } \theta \text{ is } v_{i,f} \end{cases} \tag{11}$$

where

$$\xi(t) = z^{(t)} \times (f_\Theta(\mathbf{p}^{(t)}) - f_\Theta(\mathbf{q}^{(t)})) \tag{12}$$

$$G(p_i^{(t)}, q_i^{(t)}) = \sum_{j=1}^{n} v_{j,f}(p_i^{(t)}p_j^{(t)} - q_i^{(t)}q_j^{(t)}) - v_{i,f}(p_i^{(t)2} - q_i^{(t)2}) \tag{13}$$

Note that $\sum_{j=1}^{n} v_{j,f}p_j^{(t)}$ and $\sum_{j=1}^{n} v_{j,f}q_j^{(t)}$ are independent of i and thus they can be precomputed.

One of the most popular algorithms for gradient descent is stochastic gradient descent (**SGD**), in which for each triple $((\mathbf{p},\mathbf{q}),z) \in S'$, an update is performed as:

$$\theta = \theta - \eta(\frac{\partial l_t}{\partial\theta} + 2\lambda_\theta\theta) \tag{14}$$

3.3.2 Adaptive Regularization

In eq.(9), we utilize L2 regularization to overcome the overfitting problem. The effectiveness of such regularization approach depends largely on the choice of the regularization parameter λ. However, the grid search on validation set for finding the best λ is time-consuming. Thus, to further enhance the efficiency of Ranking FM, we follow the work of [14], and modify the adaptive regularization method to estimate the Ranking FM model parameters.

The Adaptive Regularization (**AR**) method can be summarized as a nested optimization task [14]. In **Step 1**, on the training set S_T, the future model parameters $\Theta^{(t+1)}$ are optimized for the regularized loss objective with a current regularization constant $\lambda^{(t)}$; In **Step 2**, on the validation set S_V, the objective is to determine the best future regularization values $\lambda^{(t+1)}$ with the updated model $\Theta^{(t+1)}$ that minimizes the loss. In summary, with adaptive regularization, the optimization method is simple and only requires a little extension to the standard SGD algorithm.

4. FEATURE DESCRIPTION

Several features have been proved effective in the prior work [2]. However, these features are not fully utilized to further improve the performance of learning to rank approach in the microblogosphere. In this section, we describe the features used in Ranking FM in detail. We classify all the features into three groups as *content relevance features*, *semantic expansion features* and *quality features*.

4.1 Content Relevance Features

Content relevance features measure the relevance between a tweet and a specific query by analyzing the content of tweets. These features are *QueryTOTweet* (the term overlap between a query and a tweet), *QueryBM25Tweet*(the standard BM25 weighting function is adopted to measure the content relevance between query Q and tweet T), *QueryTFIDFTweet* (the cosine similarity distance between a query and a tweet in the *Vector Space Model* with the TFIDF weighting method), *QueryLMTweet* (the KL-divergence language model based retrieval method is utilized to measure the relevance between query language model $\hat{\theta}_Q$ and tweet language model $\hat{\theta}_T$).

4.2 Semantic Expansion Features

As microblog retrieval suffers severely from the vocabulary-mismatch problem (i.e. term overlap between query and tweet is relatively small), different semantic

expansion techniques can be leveraged to improve the retrieval effectiveness. However, these expansion resources are not fully explored to obtain more informative evidence in the prior work (e.g. [2]). In this section, we introduce several novel semantic expansion features on basis of query expansion and document expansion.

To extract features related to query expansion, we first follow the work [8] and employ a two-stage pseudo-relevance feedback query expansion approach to obtain a group of query-expansion (QE) terms. Then we treat the QE terms as a "new query" and calculate the semantic similarity between QE terms and the original tweet content using the aforementioned similarity measures. Thus we obtain the following features: *QETOTweet*, *QEBM25Tweet*, *QETFIDFTweet* and *QELMTweet*.

Shortened URLs within tweets can enrich the representation of the original tweets by leading users to an informative, topic-related web page. Many learning to rank approaches only leverage this resource as a binary feature or the count of the URLs. Inspired by the work [8], we expand the shortened URLs by extracting the content of the $<TITLE>$ tag from the raw HTML markup, and we name this content as *Topic Information* (i.e. TopicInfo). With the help of Topic Information, we extract the following document-expansion related features by replacing the original tweet with TopicInfo: *QueryTOTopicInfo*, *QETOTopicInfo*, *QueryBM25TopicInfo*, *QEBM25TopicInfo*, *QueryTFIDFTopicInfo*, *QETFIDFTopicInfo*, *QueryLMTopicInfo* and *QELMTopicInfo*.

4.3 Quality Features

Unlike content relevance and semantic expansion features which are query-biased, quality features are tended to estimate the quality of a tweet. Some specific features of social network service can be used to measure the quality and potential popularity in the entire social network. Based on the assumption that users prefer those tweets that are related with their query or popular in the social network, we can conclude the following features: mention count, retweet count, hashtag count, shortened URL count and length of tweet (after stopword removal).

Previous work [2, 5] also demonstrates the usefulness of user-specific features in the microblogosphere, such as user activeness defined as tweet frequency and registration time, follower number, popularity score [2] computed by PageRank, etc. However, these features are not available in our experiment database (i.e. Twitter11 Corpus), and we will explore them in our future work.

5. EXPERIMENTS

Several experiments are conducted to measure the performance of Ranking FM for microblog retrieval. In this section, we firstly describe the experimental setup. Secondly, we evaluate the performance of Ranking FM and compare it with several baseline methods. Lastly, we conduct analysis on (1) the importance of each feature group by a feature ablation study, (2) the influence of hyper-parameter k which defines the factorization dimensionality and (3) the performance comparison between two optimization methods.

5.1 Experimental Setup

The Tweet11 corpus was obtained using a donation of the unique identifiers of a sample of tweets from Twitter [11]. The Tweet11 has a sample of about 16 million tweets. In addition, we also crawled all the shortened URLs contained in Tweet11 corpus to enrich the representation of original tweets, and extracted each piece of topic information from the corresponding URL to generate the TopicInfo corpus. For both corpora, we discarded the non-English tweets by using a language detector with infinity-gram, named ldig[1]. Also, we removed the simple retweeted tweets beginning with 'RT' based on the assumption that such tweets have no extra information beyond the original ones. Moreover, each tweet was stemmed using the Porter algorithm and stop words were removed using the INQUERY words stoplist.

In TREC'11 Microblog Track, NIST created 50 topics and provided the corresponding assessments conducted by NIST assessors. Tweets were judged on the basis of the defined information need using a three-point scale [11]: *Not Relevant*, *Minimally Relevant* and *Highly Relevant*. We took advantages of these topics along with assessments as training set in learning to rank approaches. We used another 60 topics, which are the official queries in the TREC'12 Microblog Track, as test queries to measure the performance of Ranking FM. The main evaluation metrics in our experiment are precision at N (i.e. P@N) and MAP, which are widely used in IR. In TREC'12, tweets are also judged based on three grades as in TREC'11 Microblog Track. The evaluation measures are P@30 and MAP with respect to *highrel* (i.e. tweet set judged as highly relevant) and P@30 is the official main metric for the real-time search task in 2011 and 2012.

5.2 Ranking FM Performance

To demonstrate the performance of our approach, we compare our system with three baseline methods. The first baseline method is a real-time ranking model under language model framework, proposed by Liang *et al.* [8]. We realize this unsupervised model and denote it as **KL2SFBLoc**. The second baseline method is a state-of-the-art ranking SVM model (denoted as **RSVM_Full**) with all the introduced features in Section 4. Specifically, we set Dirichlet smoothing parameter $\mu = 100$ when computing the Language model score and use default parameters in Lemur[2] to extract BM25 and TFIDF Model Score. A toolkit named SVM$^{\text{rank}}$ [3] implemented by Thorsten Joachim is used to train this model (C is tuned using five-fold cross-validation with training data). The best performed model in TREC'2012 Real-Time Search task (i.e. **hitURLrun3** proposed by Han *et al.* [3]) is also reported for comparison.

As for the proposed method, we adopt all the features introduced in Section 4 and implement both stochastic gradient descent and adaptive regularization learning methods to optimize Ranking FM with $k = 3$. The corresponding two methods are labeled as **RFM_FullSGD** and **RFM_FullAR**, respectively. In practice, when training Ranking FM with stochastic gradient descent, we use the same regularization parameters $\lambda_{\mathbf{w}}$ and $\lambda_{\mathbf{v}}$ for $\mathbf{W}$ and $\mathbf{V}$. In addition, we conduct five-fold cross validation to search the regularization parameters. The best regularization parameters are chosen based on the MAP score on validation set.

[1]https://github.com/shuyo/ldig
[2]http://lemurproject.org/lemur/
[3]http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Table 1: Performance comparison between Ranking FM and three baseline systems. Among them, hitLRrun3 is the best run in TREC'12 Real-Time Search Task. The best performance of each line is shown in bold. †, ‡, ¶ mean the corresponding improvements over hitURLrun3, KL2SFBLoc and RSVM_Full are significant (with p value < 0.05) respectively.

Metric	KL2SFBLoc	RSVM_Full	hitURLrun3	RFM_FullSGD	RFM_FullAR
P@30	0.2441	0.2616	0.2701	**0.2808**†‡¶	0.2746‡¶
MAP	0.2506	0.2597	0.2642	**0.2694**‡¶	0.2678‡¶

Table 1 shows the performance comparison for aforementioned approaches. It can be clearly observed that both **RFM_FullSGD** and **RFM_FullAR** outperform **KL2SFBLoc** and **RSVM_Full** remarkably. More specifically, **RFM_FullSGD** improves the P@30 over **KL2SFBLoc** and **RSVM_Full** by 15.03% and 7.34%, respectively. Moreover, **RFM_FullAR** is comparable with **hitURLrun3** and increases P@30 by 3.96%.

5.3 Feature Study

In this section, we empirically evaluate the effectiveness of each feature group with a feature ablation study. As shown in Section 4, we classify all the features into three groups as content relevance features, semantic expansion features and quality features. Semantic expansion features can be further classified into query expansion features and document expansion features. In this experiment, we first train the Ranking FM of $k = 3$ with all the features by stochastic gradient descent, and then remove one group of features each time. Notice that removing query expansion features means excluding the features with *QE* prefix (e.g. *QELMTopicInfo*) while removing document expansion features means excluding the features with *TopicInfo* suffix (e.g. *QueryLMTopicInfo*).

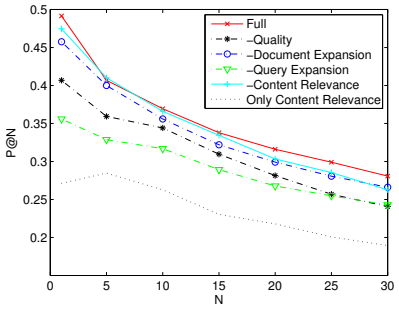

Figure 1: P@N performance with different feature groups. 'Full' means all the features, while '-' means removing a specific group from the full feature set

We can observe from Figure 1 that when removing all the quality features, P@N drops substantially (p-value < 0.01 by t-test). This indicates the significance of quality features in our ranking FM model. Removing query expansion features would also lead to a dramatic decrease in precision (p-value < 0.01), which reveals the effectiveness of query expansion techniques in short-text retrieval. Document expansion and content relevance features, though not as important as query expansion features, also play an important role in gaining a good performance. On the other hand, when only employing content relevance features, the performance is much worse than that of the model with the full feature set (p-value < 0.01).

5.4 Influence of the Hyper-parameter k

The hyper-parameter k in eq.(3) defines the factorization dimensionality. In this section, we analyze the influence of k in Ranking FM. When adapting learning to rank approaches to microblog retrieval, we consider feature interactions very significant in improving the retrieval performance and the presentation of interactions is also important. To verify this assumption, we trained Ranking FM model with different hyper-parameter k with all features involved and stochastic gradient descent.

It can be observed from Figure 2 that the Ranking FM model can achieve optimal P@30 and MAP values with $k = 3$. In addition, the improvements over Ranking FM with $k = 0$ (i.e. a 1-way Ranking FM neglecting pairwise feature interactions) are significant, which also reveals that the 2-way Ranking FM considering interactions between features is more effective in microblog retrieval. In fact, FM model with $k = 0$ is identical to a linear SVM model with the simple linear kernel $K_l(x, \mathbf{z}) := 1 + < x, \mathbf{z} >$ [12], which can also explain the superiority of our 2-way Ranking FM over the state-of-the-art Ranking SVM with linear kernel.

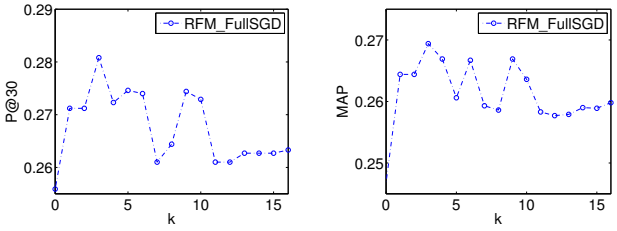

Figure 2: Ranking FM performance with different hyper-parameter k

On the other hand, when k is large and close to the feature size, the performance of 2-way Ranking FM drops while it is still better than that of Ranking FM with $k = 0$. This reveals the importance of selecting an appropriate k. Unlike the Ranking SVM with polynomial kernel which models each pair of interaction parameters independently, the interaction parameters of a 2-way FM are factorized and thus $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ and $\langle \mathbf{v}_i, \mathbf{v}_k \rangle$ have overlaps by sharing the vector $\mathbf{v}_i$. This factorization also makes the 2-way Ranking FM more flexible by adjusting the number of factors k according to specific applications.

5.5 SGD v.s. AR

We study different behaviors of two optimization methods, i.e. SGD and AR. Table 2 lists the retrieval results of Ranking FM of $k = 3$ with different optimization methods in

Table 2: Effectiveness comparison between SGD and AR for Ranking FM

Method	P@5	P@10	P@30	MAP
RFM_FullSGD	0.4068	0.3695	0.2808	0.2694
RFM_FullAR	0.4034	0.3678	0.2746	0.2678

terms of P@5, P@10, P@30 and MAP. We can observe that both methods perform very well in the Twitter11 Corpus.

Then, we experimentally analyze the efficiency of Ranking FM with different optimization methods. We train our model with full feature set and set hyper-parameter $k = 3$. For SGD, the training time includes the model selection time with five-fold cross validation (λ_w and λ_v are both chosen from $\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$) and the final training time with the selected best λ_w and λ_v; when it comes to AR, training and validation are proceeding at the same time. Note that learning Ranking FM with AR does not require any predefined regularization values which are crucial for SGD and cost much time for determination.

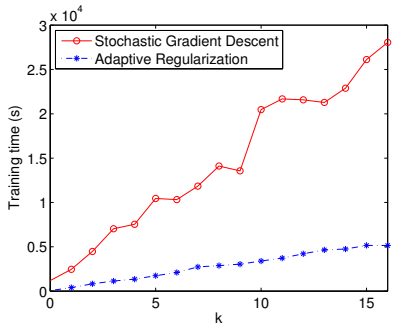

Figure 3: Efficiency comparison between SGD and AR for Ranking FM

The efficiency comparison between SGD and AR for Ranking FM is shown in Figure 3. The results indicate that AR is much more efficient than SGD though they both have an approximately linear time cost with respect to k. Instead of searching a grid of candidate regularization values, AR has an unrestricted search space to choose the appropriate regularization parameters, thus it can significantly enhance the efficiency. Furthermore, as described in Section 3.3, SGD only uses two regularization values due to the exponential complexity of grid search, while AR can update $k + 1$ regularization values at the same time without costing expensive search time.

6. CONCLUSION AND FUTURE WORK

In this study, Ranking FM model is employed which incorporates pairwise learning to rank approach with Factorization Machines for microblog retrieval. Theoretically, we use Factorization Machines as ranking function and utilize the hinge loss function to infer the Ranking FM model. Besides, we suggest two optimization methods, namely stochastic gradient descent and adaptive regularization, to estimate the model parameters. In addition, several effective features, including *content relevance features*, *semantic expansion features* and *quality features* are utilized to boost the retrieval performance. Experimental results on Tweet11 corpus demonstrate the effectiveness of Ranking FM approach as it achieves significant improvements compared with several baseline methods. Furthermore, Ranking FM which leverages all features outperforms the best performing runs of TREC'12 Real-Time Search Task. The feature ablation study indicates the effectiveness of each feature group and the analysis of hyper-parameter k shows the importance of expressing the pairwise interactions between features in learning to rank approaches for microblog retrieval. Besides, the proposed Ranking FM is very flexible as k can be adjusted for different applications. Moreover, we evaluate the different behaviors of two optimization methods in terms of effectiveness and efficiency.

7. ACKNOWLEDGMENTS

The work reported in this paper was supported by the National Natural science Foundation of China Grant 60875033.

8. REFERENCES

[1] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. SIGIR '06. ACM, 2006.

[2] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. COLING '10. ACL, 2010.

[3] Z. Han, X. Li, M. Yang, H. Qi, S. Li, and T. Zhao. Hit at TREC 2012 Microblog Track. In *TREC'12*, 2013.

[4] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. MIT Press, 2000.

[5] M. Huang, Y. Yang, and X. Zhu. Quality-biased ranking of short texts in microblogging services. In *IJCNLP*, 2011.

[6] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. WebKDD/SNA-KDD '07. ACM, 2007.

[7] T. Joachims. Optimizing search engines using clickthrough data. KDD '02. ACM, 2002.

[8] F. Liang, R. Qiang, and J. Yang. Exploiting real-time information retrieval in the microblogosphere. JCDL'12. ACM, 2012.

[9] T.-Y. Liu. *Learning to rank for information retrieval*. Springer, 2011.

[10] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *ECIR'11*. Springer, 2011.

[11] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. In *TREC'11*, 2012.

[12] S. Rendle. Factorization machines. In *Proceedings of the 10th IEEE International Conference on Data Mining*. IEEE Computer Society, 2010.

[13] S. Rendle. Factorization machines with libfm. *ACM TIST*, 3(3):57, 2012.

[14] S. Rendle. Learning recommender systems with adaptive regularization. WSDM '12. ACM, 2012.

[15] I. Soboroff, I. Ounis, and J. Lin. Overview of the TREC-2012 Microblog Track. In *TREC'12*, 2013.