FP-Rank: An Effective Ranking Approach Based on Frequent Pattern Analysis

Yuanfeng Song, Kenneth Leung, Qiong Fang, and Wilfred Ng

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology, Hong Kong, China
{songyf, kwtleung, fang, wilfred}@cse.ust.hk

Abstract. Ranking documents in terms of their relevance to a given query is fundamental to many real-life applications such as document retrieval and recommendation systems. Extensive studies in this area have focused on developing efficient ranking models. While ranking models are usually trained based on given training datasets, besides model training algorithms, the quality of the document features selected for model training also plays a very important aspect on the model performance. The main objective of this paper is to present an approach to discover "significant" document features for learning to rank (LTR) problem. We conduct a systematic exploration of frequent pattern-based ranking. First, we formally analyze the effectiveness of frequent patterns for ranking. Combined features, which constitute a large portion of frequent patterns, perform better than single features in terms of capturing rich underlying semantics of the documents and hence provide good feature candidates for ranking. Based on our analysis, we propose a new ranking approach called *FP-Rank*. Essentially, *FP-Rank* adopts frequent pattern mining algorithms to mine frequent patterns, and then a new pattern selection algorithm is adopted to select a set of patterns with high overall significance and low redundancy. Our experiments on the real datasets confirm that, by incorporating effective frequent patterns to train a ranking model, such as RankSVM, the performance of the ranking model can be substantially improved.

Keywords: Learning to rank, frequent pattern, combined features, feature selection, ranking performance;

1 Introduction

Ranking is a well-recognized problem in the areas of knowledge management and information retrieval, since it is an integral part of many data-intensive applications such as advertising, documents retrieval, recommender systems, and many others. For example, given a query, in a document retrieval system, an effective ranking algorithm is essential to estimate the relevance of each document with respect to this query, so that users can easily find the most relevant documents.

A high-quality ranking method is vital to guarantee the retrieval qualities. The problem of finding effective ranking (or the ranking problem) has attracted a

lot of researchers' attention in recent years. Many empirical ranking models, like the Boolean model, the vector space model, and the probabilistic model, were then adopted to solve the ranking problem [2]. However these methods usually suffer high cost for parameter tuning. Later, machine learning approaches, such as RankSVM[15], RankNet[3], SoftRank[25], CRR[24], etc. have been derived to automatically learn ranking functions, and they are collectively regarded as the *learning to rank (LTR)* methods. By representing the documents with a large amount of features and making use of advanced machine learning techniques, most existing LTR methods give rise to very effective ranking functions.

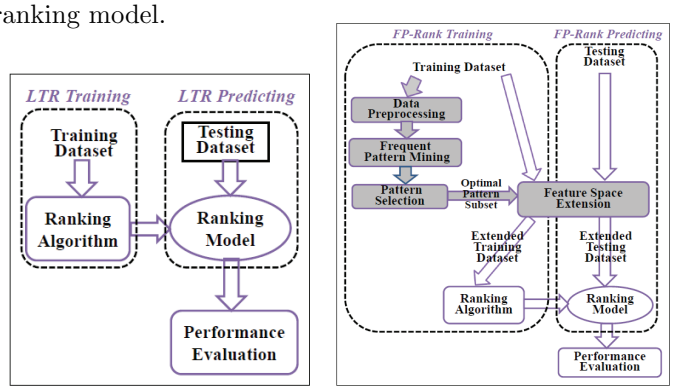
While the majority of the research focuses on the design of more effective ranking models, limited studies are carried out to improve the quality of the document features used in LTR approach. In fact, besides the ranking model training algorithms, the performance of a ranking model is also highly related to the choice of the features used for ranking. In this paper, we systematically investigate the possibility of frequent pattern-based ranking approach, where a ranking model is built in terms of single features as well as significant frequent patterns. We propose a new ranking approach, *FP-Rank*, which optimizes the set of features used in LTR to improve the accuracy of ranking methods.

Combined features, which constitute a large portion of frequent patterns, are proved to be effective to capture underlying semantics of datasets [6] [7]. For ranking problem, a good example is that in order to extract features to represent documents, compared to single words (single feature), phrases (combined features) can better deliver the semantics of the documents. In this paper, we first formally analyze the ranking effectiveness of frequent patterns. In particular, we adopt a well-acknowledged criterion called *pattern significance* to measure the ranking capability of a pattern. Then, we show combined patterns, which consist a large portion of frequent patterns tend to have higher significance than single patterns. Furthermore, we prove the significance of low frequency patterns is limited due to their small coverage in the dataset. This work provides us a theoretical support to use frequent patterns as feature candidates for ranking problem and to filter the infrequent patterns when mining frequent patterns.

Our important observation is that not every frequent pattern is equally helpful for ranking. A good example is stop words which appear frequently in the documents but tends to be useless in differentiating the documents. In addition, due to large amount of possible frequent patterns, including all patterns in the extended feature space not only increases model training time, but also deteriorates the ranking accuracy due to problem of over-fitting the model. These conclusions provide us the necessity to do further feature selection on frequent pattern set after mining frequent patterns. Therefore, we propose a new algorithm to further select a pattern subset with high overall significance and low redundancy after frequent pattern mining.

We now highlight all the components of our ranking approach called *FP-Rank*, as shown in Figure 1(b), which consists of the following three phases: (1) frequent pattern mining, (2) pattern selection, and (3) model training. In this paper, we employ FP-Close [13] as the frequent pattern mining method, which

is shown to be effective to mine closed frequent itemsets. Then, by adopting the pattern significance criterion, our proposed pattern selection method does the further pattern selection. Finally, the selected patterns are used to extend the original feature space of training dataset, and the extended dataset is used to train the ranking model.



(a) Usual LTR Approach (b) FP-Rank Approach

Fig. 1. Traditional LTR Approach vs. Our Proposed FP-Rank Approach

In summary, the major contributions are fourfold:

- We formally justify that frequent patterns are important in ranking. By incorporating frequent patterns, the quality of training datasets can be improved, and eventually the performance of ranking methods can be boosted.
- We propose a novel pattern selection algorithm to select a pattern set with high overall significance and low redundancy. The pattern set is proved to be effective for ranking.
- We present a new ranking approach called *FP-Rank*. In our proposed approach, the ranking models are built in terms of single features as well as significant frequent patterns.
- We provide experimental evaluation of our proposed algorithm on real datasets. By incorporating the selected patterns as new features for ranking, the ranking performance of current widely-used LTR model such as RankSVM has been greatly improved.

The rest of this paper is organized as follows. The notations and basic concepts are introduced in Section 2. The related work is discussed in Section 3. In Section 4, the details of the frequent pattern-based ranking approach and the *FP-Rank* approach are presented. Extensive experiments on real datasets have been conducted in Section 5. Finally Section 6 concludes the paper.

2 Preliminaries

We introduce notations and basic concepts that are used throughout the paper.

Documents and the Training Dataset. We denote by A the set of m attributes that are used to represent documents, and the domain of each attribute $a_i \in A$ is either a range $[l_i, u_i]$ or a discrete value set R_i. The training dataset is denoted by $\mathcal{D}$, and each record in $\mathcal{D}$ is in the form of $\langle q, \boldsymbol{d}, y \rangle$, where q is a query, $\boldsymbol{d}$

is a document, and y is the relevance score of the document $\boldsymbol{d}$ with respect to the query q. A document $\boldsymbol{d}$ is a set of attribute-value pairs, denoted as $\boldsymbol{d} = \{\langle a_1, v_1\rangle, \ldots, \langle a_m, v_m\rangle\}$, where v_i is the value of attribute a_i for $1 \leq i \leq m$. The relevance score y of a document is a value in the range $[0, K]$, where 0 means no relevance between the query and the document and the (maximum) value K means a "perfect" relevance.

Patterns (Features), single patterns and combined patterns. A *pattern* is a set of attribute-value pairs, and we denote it as $\alpha = \{\langle a_{i_1}, v_{i_1}\rangle, \ldots, \langle a_{i_k}, v_{i_k}\rangle\}$. We call the set of attributes contained in a pattern α the *associated attribute set* of α and denote it as A^α. A^α is a subset of A, i.e., $A^\alpha \subseteq A$. Given a pattern, if the size of its associated attribute set is 1, we call this pattern a *single pattern*; if the size of its associated attribute set is larger than 1, we call it a *combined pattern*. Since the patterns are used as features in *FP-Rank*, we use interchangeably the concepts patterns and features, single patterns and single features, combined patterns and combined features, when no ambiguity arises.

Frequent patterns. Given a pattern α, we denote by $\mathcal{D}_\alpha$ the set of records $\langle q_i, \boldsymbol{d_i}, y_i\rangle$ in $\mathcal{D}$ such that, $\boldsymbol{d_i}$ contains pattern α. For example, suppose we have a record $\langle q, \{\langle a_1, v_1\rangle, \langle a_2, v_2\rangle, \langle a_3, v_3\rangle\}, y\rangle$, the record is said to belong to $\mathcal{D}_\alpha$ with the pattern $\alpha = \{\langle a_1, v_1\rangle, \langle a_3, v_3\rangle\}$. Given a threshold θ_0, a pattern α is said to be a *frequent pattern* if $P(\alpha) = \frac{|D_\alpha|}{|D|} \geq \theta_0$. We use F to denote a set of frequent patterns.

Learning to rank problem. The LTR approach solves the ranking problem in the following way. First, it takes a training dataset $\mathcal{D}$ as the input, and a ranking model is then constructed on $\mathcal{D}$. The testing dataset $\mathcal{T}$ contains the records in the form of $\langle q, \boldsymbol{d}, \bar{y}\rangle$ and $\bar{y}$ is the relevance score to be estimated. Then, the ranking model is applied on $\mathcal{T}$ to estimate $\bar{y}$ of each record in it. Finally, records in $\mathcal{T}$ are given in the form of a list sorted in term of their estimated relevance scores. The LTR approach is shown on Figure 1(a).

MAP and NDCG. MAP and NDCG are two criteria to evaluate the performance of ranking model. The details can be found in [12].

3 Related Works

Frequent pattern mining based classification: Frequent pattern mining has been a focused theme in data mining research, which gives rise to a large number of scalable methods. A comprehensive survey can be found in [14]. Besides traditional techniques of deterministic frequent pattern mining, mining frequent itemsets over uncertain databases has also attracted much attention recently. For example, Tong et al. [26] [27] compare eight representative approaches of uncertain frequent itemset mining and develop a comparable software platform.

The frequent pattern-based classification is inherently related to associative classification. In associative classification, a classifier is built upon high quality rules, such as the ones with high-confidence and high-support. The association between frequent patterns and class labels is then used for prediction. The work related to this area includes: CBA[19], CMAR[18], CPAR[34] etc. These methods differ in their rule selection criteria (confidence, support, etc), number of rules

they select (dataset coverage, top N, etc), and prediction result combination methodology. Cheng [6] provides a theoretically analysis about why frequent patterns are helpful for classification and bridges the gap between pattern's support with its information gain. Recent work in this area focuses on how to mine the discriminative pattern efficiently. For example, Cheng [6] provides a pattern selection method MMRFS to select frequent patterns from the candidate pattern set. HARMONY [32] adopts an instance-centric rule generation approach and achieves high accuracy and efficiency. DDP-Mine [7] provides a more effective pruning technique and directly mines out informative patterns for classification.

Learning to rank: Ranking is a fundamental problem in many application areas such as recommendation systems, document retrieval and advertising etc. Previous work such as boolean models, vector models and probabilistic models [2] usually suffers high cost of parameter tuning since we usually consider a large number of relevant features for documents and queries.

Machine learning techniques provide many feasible solutions, since they can automatically learn parameters and make use of a large part of features in the model learning process, and this approach is referred *Learning to rank* (LTR) approach. According to [4], [5], current LTR methods can be classified into three categories: (i) Pointwise approach, (ii) Pairwise approach and (iii) Listwise approaches. In pointwise approach, each training example is treated as an independent instance and a model is trained to map each document's features to its relevance score which could be based on regression [9] or classification [20] [17]. The pairwise approach train ranking function to minimize a loss function which is based on pair-wise preferences. The ranking problem is then transformed into binary classification problem. Typical examples of such models includes RankSVM [15], RankNet [3], FRank [28], MHR [23], RankBoost[11], and CRR[24]. etc. In listwise approach, the models consider the whole document list instead of document pairs by either directly optimizing the IR measures, or indirectly optimizing the IR measures by employing a loss function correlated to IR measures. Directly optimizing the IR measures is difficult since they depend on the rank and are not differentiable. Example methods include [8], SVM^{map} [35], AdaRank [33], Boltzrank [31], NDCG-Boost [29], and [16]. Indirectly optimizing the IR measures includes RankCosine [21], and ListNet [5].

Beside the above approaches, association rules have also been applied to solve the LTR problem by Veloso [30]. When predicting the orders, several high confidence rules are used and the final relevance score is computed by weighted combination of the relevance score of all these selected rules. Our approach is inspired by the success of existing frequent pattern based classification approaches, however, we differ from these approaches in the following three aspects: (1) We use frequent patterns to extend the feature space instead of only using association rules [30]. (2) Rather than only considering confidence or support of patterns or association rules, we consider the characteristic of ranking problem and provide pattern selection method to select high significance, low redundancy pattern set for effective ranking. (3) Our approach is compatible with most of current LTR algorithms and it demonstrates significant ranking improvement.

4 Frequent Pattern-Based Ranking Approach

In this section, we present the frequent pattern based-ranking approach *FP-Rank*, which carries out ranking by the following phases: (1) frequent pattern mining, (2) pattern selection, and (3) model training. We first prove the effectiveness of frequent patterns for ranking, and adopt the frequent pattern mining methods such as FP-Close [13] to mine frequent patterns. By adopting the pattern significance criterion, a greedy method is developed to select the pattern set with high overall significance and low redundancy. Finally, the selected patterns are used to extend the original feature space of training dataset, and the extended dataset is used to train the ranking model.

4.1 The Effectiveness of Frequent Pattern for Ranking

Frequent patterns have two essential properties: *combined patterns* and *high frequency*. We analyze how these properties contribute to the ranking problem.

The significance of combined patterns A large portion of frequent patterns are combined patterns. Compared with single patterns, combined patterns are better at capturing the underlying semantics of the documents, and thus they can be more effective for producing more accurate ranking.

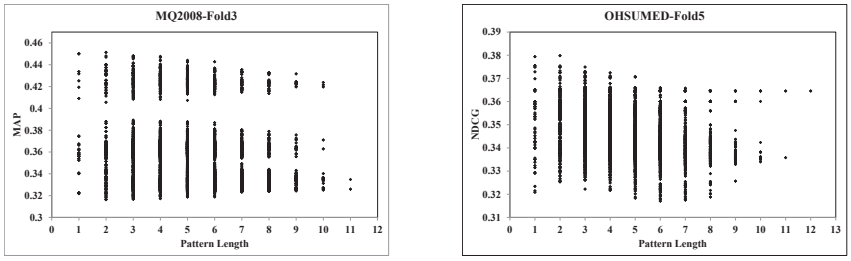
In order to formally analyze ranking capability of frequent patterns, we adopt a well-acknowledged criterion called *pattern significance* for ranking.

Pattern significance. Given a pattern α, pattern significance $S(\alpha)$ measures the correlation between α *w.r.t* relevance score. For the ranking problem, MAP and $NDCG$ are used to evaluate the effectiveness of a feature, which are proved to be helpful in [12]. Here we adopt the same methodology and define pattern significance be a pattern's MAP or $NDCG$, denoted as $MAP(\alpha)$ and $NDCG(\alpha)$, and they can be computed by MAP and $NDCG$ of the ranking model trained solely based on this pattern (using RankSVM, RankNet, etc.).

We utilize the Microsoft LETOR MQ2008 and OHSUMED datasets [22], and plot the MAP and $NDCG$ of single patterns as well as combined patterns. We can see that the combined patterns tend to have higher significance, e.g. Figures 2(a) and 2(b).

Pattern significance vs. pattern frequency We now study the relationship between the significance of a frequent pattern and its frequency, and demonstrate that the significance of patterns with low frequency is limited. In addition, patterns with low frequency may lower the ranking accuracy due to model over-fitting. We provide the following lemma for detailed illustration.

Lemma 1 *Given dataset D and pattern α, suppose pattern frequency $P(\alpha) = \frac{|D_\alpha|}{|D|} = \theta$. To simplify our analysis, we further assume relevance score $y \in \{0, 1\}$, the percentage of relevant documents $P(y = 1) = \frac{|D_{y=1}|}{|D|} = p$, the possible significance upper bound of α, denoted as $S(\alpha)_{ub}$, is monotonically increasing with θ, when θ is small.* ***i.e.,*** *$0 \leq \theta < min\{1 - p, p\}$.*



(a) MQ2008 Fold3 MAP (b) OHSUMED Fold5 NDCG

Fig. 2. Pattern Significance vs. Pattern Length on LETOR dataset

In order to prove Lemma 1, we now cast ranking problem into a multiple classification problem by treating relevance scores as class labels, since perfect classifications lead to perfect DCG scores according to the definition of DCG in Section 2. This view connects two intrinsically different problems of ranking and classification. In addition, Li et al.[17] further proved that a model's DCG error is bounded by the converted classification error by Lemma 2.

Lemma 2 *Suppose there are n documents $\{d_1, d_2, \ldots, d_n\}$. Given a query q, the ground truth ranked list of documents is G, which is produced by ranking documents in terms of their true relevance scores. Suppose a classifier estimates the relevance score $\bar{y_i}$ of document d_i to be an integer in $[0, K]$, for $1 \leq i \leq n$. Then the documents are sorted in terms of their estimated relevance scores to produce the estimated ranked list R. The corresponding DCG error of R with respect to G is bounded by the square root of the classification error, that is,*

$$DCG_G - DCG_R \leqslant \left(2^K - 1\right) \left(\sum_{i=1}^{n} c_{[i]}^2 - n \prod_{i=1}^{n} c_{[i]}^{2/n}\right)^{1/2} \left(\sum_{i=1}^{n} 1_{y_i \neq \bar{y_i}}\right)^{1/2}. \quad (1)$$

Based on Lemma 2, we now prove that the significance of patterns with low frequency is limited. To simplify our analysis, we further assume relevance score $y \in \{0, 1\}$. Given a dataset D, let $P(\alpha) = \frac{|D_\alpha|}{|D|} = \theta$, $P(y = 1) = \frac{|D_{y=1}|}{|D|} = p$, where $D_{y=1}$ is the set of documents with relevance score $y = 1$, and $P(y = 1|\alpha) = \frac{|D_\alpha \cap D_{y=1}|}{|D_\alpha|} = q$. Then

$$1 - S(\alpha) = 1 - \frac{DCG(\alpha)}{DCG_G} \leq \lambda * \left(\frac{\sum_{i=1}^{n} 1_{y_i \neq \bar{y_i}}}{|D|}\right)^{1/2}, \quad (2)$$

where $\frac{\sum_{i=1}^{n} 1_{y_i \neq \bar{y_i}}}{|D|}$ is the relevant classification error of the classifier built solely on α, denoted as $\mathcal{E}(\alpha)$, and

$$\lambda = \frac{\left(2^K - 1\right) \left(\sum_{i=1}^{n} c_{[i]}^2 - n \prod_{i=1}^{n} c_{[i]}^{2/n}\right)^{1/2} * |D|^{1/2}}{DCG_G} \quad (3)$$

is a constant for a given dataset.

From the above assumption, we deduce that $P(\alpha, y=1) = q\theta$, $P(\bar{\alpha}, y=1) = p - q\theta$, $P(\alpha, y=0) = \theta - q\theta$ and $P(\bar{\alpha}, y=0) = 1 - p - \theta + q\theta$. So the error of the classifier built on α is given by:

$$\mathcal{E}(\alpha) = min\left\{\theta + p - 2\theta q, 1 - (\theta + p - 2\theta q)\right\}. \tag{4}$$

For fixed p and θ, $\mathcal{E}(\alpha)$ varies with q, and reaches the lower bound at the following conditions. When $p \leq 0.5$,

$$\mathcal{E}(\alpha)_{lb} = \begin{cases} p - \theta, & \text{for } q = 1, 0 \leq \theta < p \\ \theta - p, & \text{for } q = \frac{p}{\theta}, p \leq \theta < 0.5 \\ 1 - \theta - p, & \text{for } q = 0, 0.5 \leq \theta < 1 - p \\ \theta + p - 1, & \text{for } q = 1 - \frac{1-p}{\theta}, 1 - p \leq \theta \leq 1 \end{cases}, \tag{5}$$

and when $p > 0.5$,

$$\mathcal{E}(\alpha)_{lb} = \begin{cases} 1 - \theta - p, & \text{for } q = 0, 0 \leq \theta < 1 - p \\ \theta + p - 1, & \text{for } q = 1 - \frac{1-p}{\theta}, 1 - p \leq \theta < 0.5 \\ p - \theta, & \text{for } q = 1, 0.5 \leq \theta < p \\ \theta - p, & \text{for } q = \frac{p}{\theta}, p \leq \theta \leq 1 \end{cases}. \tag{6}$$

We take one case of $\mathcal{E}(\alpha)_{lb}$ as an example, i.e., $p \leq 0.5$ and $0 \leq \theta < p$. $\mathcal{E}(\alpha)$ gets its lower bound when $q = 1$. The partial derivative of $\mathcal{E}(\alpha)_{lb|q=1}$ w.r.t.θ is

$$\frac{\partial \mathcal{E}(\alpha)_{lb|q=1}}{\partial \theta} = -1 < 0. \tag{7}$$

The above analysis demonstrates that when $p \leq 0.5$, $\mathcal{E}(\alpha)_{lb}$ is a function of the pattern frequency θ. When θ is small, i.e., $0 \leq \theta < p$, $\mathcal{E}(\alpha)_{lb|q=1}$ is monotonically decreasing with θ, i.e., the smaller θ is, the larger $\mathcal{E}(\alpha)_{lb|q=1}$ is, and according pattern significance $S(\alpha)$ is likely to be smaller as well. The conclusion is the same for the cases with $p > 0.5$. When θ is small, i.e., $0 \leq \theta < 1 - p$, $\mathcal{E}(\alpha)_{lb|q=0}$ is monotonically decreasing with θ. Therefore, the significance of patterns with low frequency is bounded by a small value.

We have discussed the effectiveness of combined patterns for ranking in Section 4.1. One possible way to generate combined patterns from the original dataset is to enumerate all the combinations of the single patterns. This naive method suffers from the high cost due to large number of combinations ($O(2^n)$). The formal analysis in this section indicates that we can use frequent patterns with frequency large than some threshold *min_sup* instead of all the single pattern combinations without suffering too much performance loss, since significance of patterns with low frequency is limited.

4.2 Feature Selection on Frequent Pattern Set

Although frequent patterns are useful for improving accuracy of ranking, it does not mean that every frequent pattern is equally helpful. A good example is stop words which appear quite a lot in most of the documents, but almost helpless

Algorithm 1 FP-Rank Feature Selection

Input: Frequent pattern set F; Training dataset D; Pattern Number N;
Output: Pattern set F_s;

$F_s = \Phi$;
while ($|F_s| < N$) **do**
  $\alpha = arg\max_{\alpha \in F - F_s} \Phi(\alpha)$;
  if α can correctly cover at lease one instance in D **then**
    $F_s = F_s \cup \{\alpha\}$;
  end if
  $F = F - \{\alpha\}$;
  if $F = \Phi$ **then**
    break;
  end if
end while
return F_s

in differentiating documents. Since frequent patterns are generated by only considering frequency, the mined frequent patterns may contain a large portion of insignificant patterns. Including insignificant patterns for model training does not only increase the model training time, but also leads to the reduction of the ranking performance due to model overfitting. The objective of pattern selection is to find a pattern set from all the mined frequent patterns, such that the overall pattern significance is high, while the redundancy among the patterns in the set is low. This problem is known to be NP-hard [12]. Since the number of mined frequent patterns is usually extremely large, we therefore need to devise an efficient pattern selection method, which searches for the pattern set in a greedy way. We have defined pattern significance in Section 4.1, and the redundancy criterion is defined as follows.

Redundancy between two patterns. Given two patterns α and β, redundancy $R(\alpha, \beta)$ measures the correlation between these two patterns. Particularly, we consider the redundancy between two patterns based on the prediction results given by the models solely built on each of them. Many methods have been proposed to measure the distance between two ranked lists, such as Sperman's footrule, Kendall's tau distance, etc [12]. We choose the Kendall's tau distance, which has been proved to be effective in measuring distance of ranked lists [12], and thus the $R(\alpha, \beta)$ is defined as follows:

$$R(\alpha, \beta) = \tau(\alpha, \beta) \times min(S(\alpha), S(\beta)), \text{ with } \tau(\alpha, \beta) = \frac{\sum_{q \in Q} \tau_q(\alpha, \beta)}{|Q|}. \quad (8)$$

$\tau_q(\alpha, \beta)$ is the Kendall's tau value between two rankings respectively generated based on two patterns for query q, which is defined as follows:

$$\tau_q(\alpha, \beta) = \frac{|\{(d_i, d_j) \in D_q\}| d_i \prec_\alpha d_j \; and \; d_i \prec_\beta d_j|}{|\{(d_i, d_j) \in D_q\}|}, \quad (9)$$

where D_q denotes the set of documents given by query q. $\tau(\alpha, \beta)$ is the average Kendall's tau value over all the queries in set Q.

We define a score for a pattern α, denoted as $\Phi(\alpha)$, as follows:

$$\Phi(\alpha) = S(\alpha) - max_{\beta \in F_s} R(\alpha, \beta). \tag{10}$$

The greedy pattern selection algorithm is presented in Algorithm 1. It searches over all the mined frequent patterns in F and find the one with maximal Φ value (Line 3), and if this pattern can correctly cover at least one instance in the training dataset, we include it to the selected pattern set F_s (Lines 4-5). We keep searching the mined frequent pattern set until N patterns are found (Line 2) or set F is empty (Lines 8-10).

4.3 FP-Rank Approach

We present the two algorithms in our *FP-Rank* Approach: *FP-Rank* Training (Algorithm 2) and *FP-Rank* Predicting (Algorithm 3). In the training part, after we preprocess the dataset (Line 1), the frequent pattern mining algorithm, such as FP-Close [13], is adopted for mining frequent patterns (Line 2). Our proposed pattern selection algorithm 1 is used to select a set of patterns F_s (Line 3). The selected patterns are used to extend the original feature space of the dataset (Line 4), and extended dataset is used to train a ranking model M, using RankSVM, RankNet, and etc (Line 5). In the prediction part, we use the pattern set F_s to extend the feature space of the testing instances (Line 1), and then ranking model M is used to predict the relevance scores of testing instances (Line 2).

Algorithm 2 FP-Rank Training

Input: Training dataset D;
Output: Ranking model M. Pattern set F_s
1: D' =Preprocessing(D). //data discretization etc.
2: F =FP-Close(D'). //closed frequent pattern mining.
3: F_s =FeatureSelection(F). //pattern selection (Algorithm 1).
4: D'' =FeatureSpaceExtension(F_s, D). //feature space extension using F_s and D.
5: M =ModelTraning(D'') //model training based on extended dataset.
6: **return** F_s and M

Algorithm 3 FP-Rank Predicting

Input: Pattern set F_s, Ranking model M, Testing instance t
Output: Predicted relevance score y for t
1: t' =FeatureSpaceExtension(F_s, t) //feature space extension for t using F_s.
2: y =Prediction(M, t') //relevance score prediction for t' using model M
3: **return** y

5 Experiments

In this section, we evaluate the effectiveness of *FP-Rank* framework. We introduce the datasets and the relevant setup algorithms used in the experiments in Section 5.1. Then, we evaluate the ranking performance in Section 5.2.

5.1 Experimental Setup

Dataset In our experiments, the Microsoft's LETOR benchmark [22] is used. LETOR is a benchmark for research on LTR, which composes of several data subsets, evaluation tools, and baseline evaluation results (such as RankSVM, RankBoost, etc) for ranking performance evaluation. Each data subset contains a set of queries, a set of features for query document pairs, and a set of corresponding relevance scores for the evaluation. We choose the LETOR4.0 MQ2008 dataset, the statistics of which is listed in Table 1. For each fold, the training set is first used to learn a ranking model. The validation set is used for model parameters tuning, and the ranking model is then used on testing set. The estimated relevance scores on the testing set are employed to derive the standard $NDCG@n$, $P@n$, and MAP measures in the ranking evaluation.

Table 1. Statistics of the MQ2008 dataset

No. of Features	No. of Queries	No. of Query-Document	No. of Document
46	784	15211	14384

Ranking model In our experiments, *RankSVM* is employed to derive the ranking model. It utilizes instance pairs and their preference labels in the training. The optimization formulation of RankSVM is given by:

$$\min \frac{1}{2} w^T w + C \sum_{i,j,q} \varepsilon_{i,j,q}$$
$$s.t. \forall (d_i, d_j) \in r_q^* : \omega \emptyset (q, d_i) \geq \omega \emptyset (q, d_j) + 1 - \varepsilon_{i,j,q}.$$

We employ RankSVM$^{\text{Struct}}$ [15] in the *FP-Rank* framework. RankSVM$^{\text{Struct}}$ is the most up-to-data implementation with optimized speed and performance, and previous studies [15] have already shown the effectiveness of RankSVM$^{\text{Struct}}$.

Data preprocessing Most pattern mining algorithms, such as Apriori [1], FP-Close [13], can only handle discrete attributes. However, since the attributes of most of the ranking datasets (e.g, Microsoft's LETOR datasets, Yahoo's LTR competition[1] datasets) are continuous, data discretization should be performed before frequent pattern mining. Naive discretization methods such as binary discretization or n-equal-width bins discretization suffers from two major problems: 1) information loss, which decreases the significance of frequent patterns, and 2) useless patterns, which are patterns that have limited effect for ranking but make mining and pattern selection more expensive. Since if the discretization is not fine enough, it assigns many different values into the same bins, and thus generating noise patterns with information loss. In our experiment we compared several discretization methods, and we use MDL methods [10], which gives the best results due to the minimal information loss.

[1] http://learningtorankchallenge.yahoo.com/datasets.php

Frequent pattern mining algorithm Frequent pattern mining is a well-studied theme with various available algorithms and software tools. Based on the redundancy definition in section 4.2, instead of frequent patterns, we use closed frequent patterns as features in our framework, since a closed pattern is a concise representation of all its redundant non-closed sub-patterns. We choose FP-Close [13] to mine closed frequent patterns in our experiment. To maximize the number of significant patterns, we divide each dataset into several partitions according to the relevance scores. We first mine the frequent patterns in each partition. The mined patterns are merged together, and pattern selection is then applied on the merged pattern set to find the pattern subset.

To compare different pattern selection criteria, we also adopt information gain, which is a widely-used feature quality measurement for classification, to measure significance of a pattern, and adopt an extension based on Jaccard distance for measuring the redundancy. This criterion is effective for classification according to [6].

5.2 Ranking

Accuracy The ranking results in terms of MAP and $NDCG@n$ for the MQ2008 dataset are presented in Figures 3 and Table 2. From the results, we observe that the newly added frequent patterns can significantly improve the ranking performance. Both the two feature selection criteria (i.e.,IG+Jaccard, MAP+KenTau) achieve much better results compared to the baseline method (RankSVM with no pattern added). This aligns with our claim in Section 4.1 that ranking performance can be improved by including selected frequent patterns subset.

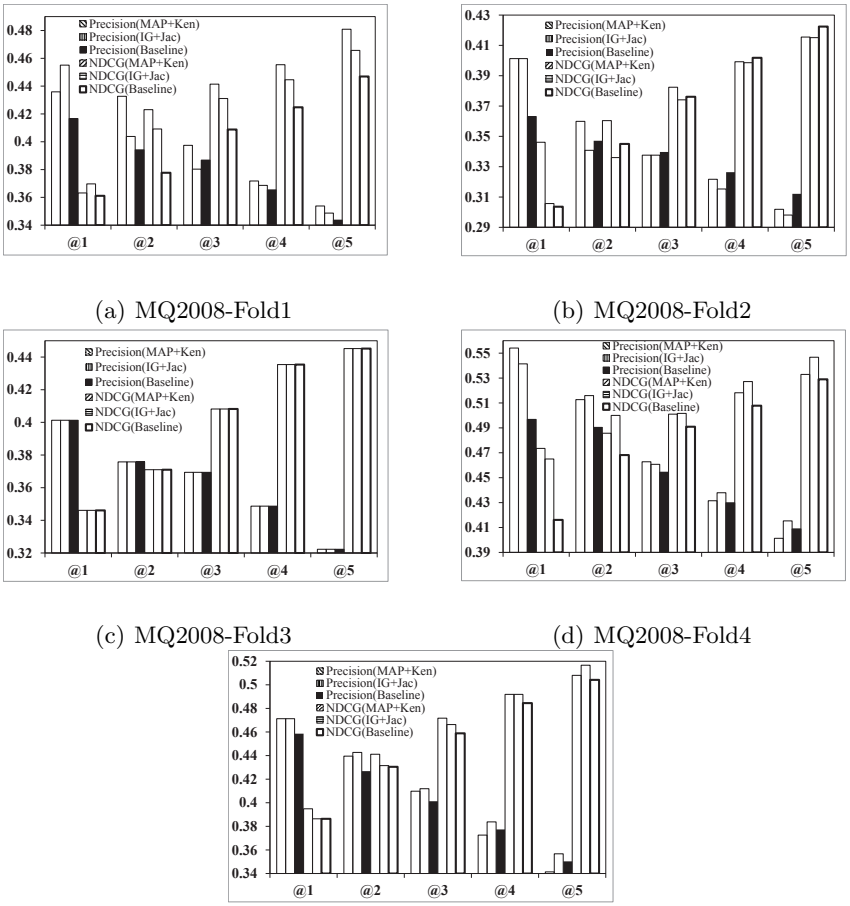
Table 2. Summary of Ranking Improvement on MQ2008 dataset

	MAP			NDCG		
Fold	Baseline	FP-Rank	Improv.	Baseline	FP-Rank	Improv.
F1	0.4502	0.4672	3.78%	0.4577	0.4784	4.52%
F2	0.4213	0.4377	3.89%	0.4296	0.4378	1.91%
F3	0.4529	0.4529	0%	0.4686	0.4686	0%
F4	0.5284	0.5472	3.56%	0.5442	0.5604	2.98%
F5	0.495	0.5059	2.20%	0.5159	0.5232	1.42%
Ave.	0.46956	0.48172	2.69%	0.4832	0.4931	2.17%

We find that our proposed MAP significance with Kendall tau redundancy criterion in *FP-Rank* achieve better results compared to IG with Jaccrad methods, showing that our proposed ranking pattern selection method is more effective comparing to methods (e.g., IG and Jaccrad) for classification (Figure 3). We observe that our method significantly improves the ranking performance (Maximum: 4.52% and Average: 2.17% in terms of $NDCG@n$; Maximum 3.89% and Average: 2.69% in terms of MAP) compared to the baseline RankSVM$^{\text{Struct}}$ method (Table 2).

The effect of pattern set size N In our pattern selection algorithm, parameter N denotes the subset size of the selected pattern. In our experiment, we try different N to train the model with training set, and the models with the best

performance on the validation set are used. As N varies, the ranking results in terms of MAP and $NDCG@n$ for the MQ2008 dataset are presented in Figure 4. Besides confirming the effectiveness of the new added patterns and our pattern selection algorithm, we conclude that the subset size N of the new added pattern is small (less than 20), which makes the model training time similar as the baseline RankSVM$^{\text{Struct}}$ method.



(a) MQ2008-Fold1

(b) MQ2008-Fold2

(c) MQ2008-Fold3

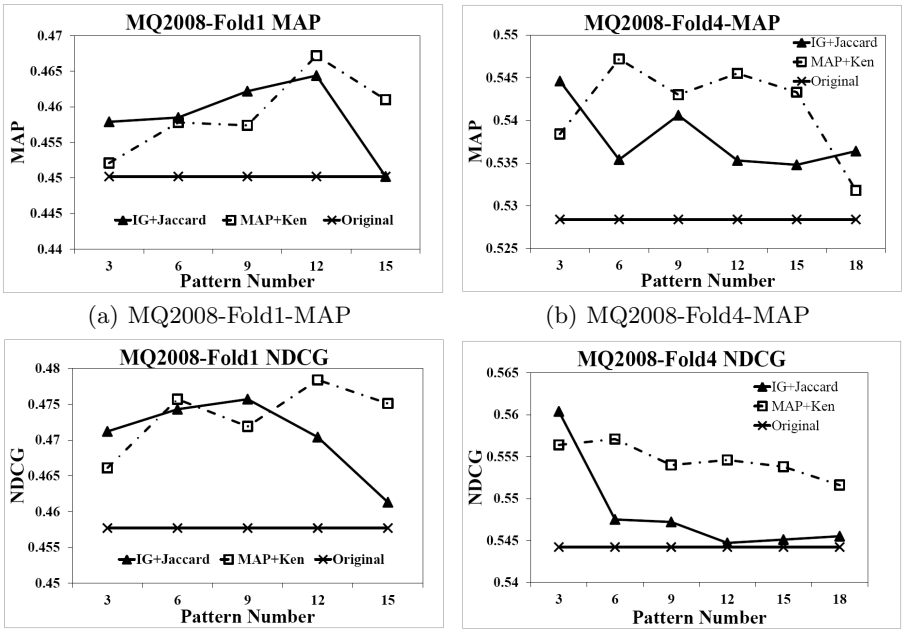
(d) MQ2008-Fold4

(e) MQ2008-Fold5

Fig. 3. Detailed Ranking Improvement on LETOR MQ2008 dataset

6 Conclusions

In this paper, we propose a new approach *FP-Rank* that aims to achieve a more effective learning to rank approach by using frequent patterns. Our study confirms that frequent patterns offer high quality features that can be used to improve the performance of a ranking model. Compared with commonly used feature selection approaches, our ranking feature selection method is able to find a pattern subset that is specific for a ranking problem. The improvement



(a) MQ2008-Fold1-MAP (b) MQ2008-Fold4-MAP

(c) MQ2008-Fold1-NDCG (d) MQ2008-Fold4-NDCG

Fig. 4. Ranking Performance Improvement vs. Pattern Number N

is clearly evidenced by the ranking accuracy measured by MAP and $NDCG$ in *FP-Rank* in a spectrum of experiments.

Acknowledgments This work is partially supported by GRF under grant numbers HKUST 617610 and 618509. We also wish to thank the anonymous reviewers for their comments.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB '94. pp. 487–499 (1994)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley (1999)
3. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: ICML'05. pp. 89–96 (2005)
4. Cao, Y., Xu, J., Liu, T.Y., Li, H., Huang, Y., Hon, H.W.: Adapting ranking svm to document retrieval. In: SIGIR '06. pp. 186–193 (2006)
5. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: ICML '07. pp. 129–136 (2007)
6. Cheng, H., Yan, X., Han, J., Hsu, C.W.: Discriminative frequent pattern analysis for effective classification. In: ICDE'07. pp. 169–178 (2007)
7. Cheng, H., Yan, X., Han, J., Yu, P.S.: Direct discriminative pattern mining for effective classification. In: ICDE'08. pp. 169–178 (2008)
8. C.J. Burges, R.Ragno, E.: Learning to rank with nonsmooth cost functions. In: NIPS '06. pp. 193–200 (2006)
9. Cossock, D., Zhang, T.: Subset ranking using regression. In: Learning Theory, LNCS'06, vol. 4005, pp. 605–619 (2006)

10. Fayyad, Irani: Multi-interval discretization of continuous-valued attributes for classification learning. In: UAI '93. pp. 1022–1027 (1993)
11. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. J. Mach. Learn. Res. 4, 933–969 (December 2003)
12. Geng, X., Liu, T.Y., Qin, T., Li, H.: Feature selection for ranking. In: SIGIR'07. pp. 407–414 (2007)
13. Grahne, G., Zhu, J.: Efficiently using prefix-trees in mining frequent itemsets. In: FIMI'03 (2003)
14. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. Knowl. Discov. 15(1), 55–86 (2007)
15. Joachims, T.: Training linear svms in linear time. In: KDD'06. pp. 217–226 (2006)
16. Karimzadehgan, M., Li, W., Zhang, R., Mao, J.: A stochastic learning-to-rank algorithm and its application to contextual advertising. In: WWW '11. pp. 377–386 (2011)
17. Li, P., Burges, C.J.C., Wu, Q.: Mcrank: Learning to rank using multiple classification and gradient boosting. In: NIPS'07. pp. 845–852 (2007)
18. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: ICDM'01. vol. 0, p. 369 (2001)
19. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: KDD'98. pp. 80–86 (1998)
20. Nallapati, R.: Discriminative models for information retrieval. In: SIGIR '04. pp. 64–71 (2004)
21. Qin, T., yan Liu, T., feng Tsai, M., dong Zhang, X., Li, H.: Learning to search web pages with query-level loss functions. Tech. rep. (2006)
22. Qin, T., Liu, T.Y., Xu, J., Li, H.: Letor: A benchmark collection for research on learning to rank for information retrieval. Information Retrieval 13, 346–374 (2010)
23. Qin, T., Zhang, X.D., Wang, D.S., Liu, T.Y., Lai, W., Li, H.: Ranking with multiple hyperplanes. In: SIGIR '07. pp. 279–286 (2007)
24. Sculley, D.: Combined regression and ranking. In: KDD '10. pp. 979–988. ACM, New York, NY, USA (2010)
25. Taylor, M., Guiver, J., Robertson, S., Minka, T.: Softrank: optimizing non-smooth rank metrics. In: WSDM '08. pp. 77–86 (2008)
26. Tong, Y., Chen, L., Cheng, Y., Yu, P.S.: Mining frequent itemsets over uncertain databases. PVLDB'12 5(11), 1650–1661 (2012)
27. Tong, Y., Chen, L., Ding, B.: Discovering threshold-based frequent closed itemsets over probabilistic data. ICDE'12 (270-281) (2012)
28. Tsai, M.F., Liu, T.Y., Qin, T., Chen, H.H., Ma, W.Y.: Frank: a ranking method with fidelity loss. In: SIGIR '07. pp. 383–390 (2007)
29. Valizadegan, H., Jin, R., Zhang, R., Mao, J.: Learning to rank by optimizing ndcg measure. In: NIPS '09 (2009)
30. Veloso, A.A., Almeida, H.M., Gonçalves, M.A., Meira Jr., W.: Learning to rank at query-time using association rules. In: SIGIR '08. pp. 267–274 (2008)
31. Volkovs, M.N., Zemel, R.S.: Boltzrank: learning to maximize expected ranking gain. In: ICML '09. pp. 1089–1096 (2009)
32. Wang, J., Karypis, G.: On mining instance-centric classification rules. IEEE Trans. on Knowl. and Data Eng. 18, 1497–1511 (2006)
33. Xu, J., Li, H.: Adarank: a boosting algorithm for information retrieval. In: SIGIR '07. pp. 391–398 (2007)
34. Yin, X., Han, J.: Cpar: Classification based on predictive association rules. In: SDM'03 (2003)
35. Yue, Y., Finley, T., Radlinski, F., Joachims, T.: A support vector method for optimizing average precision. In: SIGIR'07. pp. 271–278 (2007)