Computer Science and Information Systems 19(1):141–164 https://doi.org/10.2298/CSIS201220042Y

A Graph-based Feature Selection Method for Learning to Rank Using Spectral Clustering for Redundancy Minimization and Biased PageRank for Relevance Analysis*

Jen-Yuan Yeh[1,†] and Cheng-Jung Tsai[2]

[1] Dept. of Operation, Visitor Service, Collection and Information Management,
National Museum of Natural Science,
No. 1, Guanqian Rd., North Dist.,
Taichung City 404, Taiwan (R.O.C.)
jenyuan@nmns.edu.tw
[2] Graduate Institute of Statistics and Information Science,
National Changhua University of Education,
No. 1, Jinde Rd., Changhua City,
Changhua County 500, Taiwan (R.O.C.)
cjtsai@cc.ncue.edu.tw

Abstract. This paper addresses the feature selection problem in learning to rank (LTR). We propose a graph-based feature selection method, named FS-SCPR, which comprises four steps: (i) use ranking information to assess the similarity between features and construct an undirected feature similarity graph; (ii) apply spectral clustering to cluster features using eigenvectors of matrices extracted from the graph; (iii) utilize biased PageRank to assign a relevance score with respect to the ranking problem to each feature by incorporating each feature's ranking performance as preference to bias the PageRank computation; and (iv) apply optimization to select the feature from each cluster with both the highest relevance score and most information of the features in the cluster. We also develop a new LTR for information retrieval (IR) approach that first exploits FS-SCPR as a preprocessor to determine discriminative and useful features and then employs Ranking SVM to derive a ranking model with the selected features. An evaluation, conducted using the LETOR benchmark datasets, demonstrated the competitive performance of our approach compared to representative feature selection methods and state-of-the-art LTR methods.

Keywords: Feature selection, Feature similarity graph, Spectral clustering, Biased PageRank, Learning to rank, Information retrieval.

* This paper is an extended version of the ICCIP 2020 paper "Graph-based Feature Selection Method for Learning to Rank" [70].

† Corresponding author

1. Introduction

Ranking, a crucial task in information retrieval (IR), involves creating an ordered list of documents in which the relative order of documents represents their degree of relevance to the given query or their importance. In the last decade, learning to rank (LTR), which leverages machine learning to build effective ranking models, has received much attention. LTR automatically learns from the training data for tuning model parameters or by combining some features (or ranking models in context) into one more effective model [40]. Existing literature has proposed a variety of approaches, such as McRank [38], PRank [14], Ranking SVM [27][30], RankBoost [21], RankNet [8], FRank [62], AdaRank [67], SVM-MAP [72], and ListNet [9] (see Section 2.1).

As LTR algorithms incorporate more and more features, feature selection for ranking is needed because high-dimensional features tend to include irrelevant and redundant features, which can deteriorate the models' performance and make the models difficult to understand. High-dimensional features also lead to high computational costs in training and prediction. However, feature selection, which constructs and selects useful subsets of features for building a good predictor [25], reduces data dimensionality and eliminates redundant and irrelevant features. Thus, much work has been done in recent years to develop feature selection methods dedicated to LTR since the pioneering work of [22]. See Section 2.2 for an overview of feature selection methods for LTR.

We propose a graph-based feature selection method for LTR, referred to as FS-SCPR (Feature Selection Using Spectral Clustering and Biased PageRank) (see Fig. 3). We then develop a new LTR for IR approach that exploits FS-SCPR as a preprocessor to determine discriminative and useful features. This approach employs Ranking SVM [27][30] to derive a ranking model with the selected features (see Fig. 2). FS-SCPR selects a subset of features that have minimum redundancy with each other and have maximum relevance to the ranking problem. To minimize redundancy, FS-SCPR drops redundant features that are grouped in the same cluster. To maximize relevance, FS-SCPR greedily collects a representative feature with high relevance to the ranking problem from each cluster.

FS-SCPR comprises four steps. First, it uses ranking information to assess the similarity between two features and construct an undirected feature similarity graph. Second, it applies spectral clustering [44] to cluster features based on eigenvectors of matrices derived from the feature similarity graph. Then, it utilizes biased PageRank [26] to create a relevance score with respect to the ranking problem for each feature by analyzing the link structure of the feature similarity graph while incorporating each feature's ranking performance as preference to bias the PageRank computation. Finally, it applies optimization to select the feature from each cluster with both the highest relevance score and most information of the features in the cluster.

The main contributions of this paper are twofold:

1. We propose FS-SCPR, a graph-based feature selection method for LTR, to model feature relationships as a graph and leverage the graph model to select features using spectral clustering for redundancy minimization and biased PageRank for relevance analysis. In addition, we develop a new LTR for IR approach that integrates FS-SCPR and Ranking SVM.
2. We perform extensive experiments to evaluate the performance and effectiveness of the proposed approach using the LETOR benchmark datasets. The experimental results suggest that FS-SCPR helps improve the ranking performance. We

show the performance gains of the proposed approach compared to other feature selection methods and state-of-the-art LTR methods.

The remainder of this paper is structured as follows. Section 2 briefly reviews the related work. Section 3 elaborates the technical details of our LTR for IR approach, which incorporates FS-SCPR. Section 4 presents and discusses the experimental results. Finally, Section 5 concludes and points out possible directions for further work.

2. Related Work

2.1. LTR Methods

An LTR task consists of training and testing processes (see Fig. 1). Suppose that $F = \{f_1, ..., f_{|F|}\}$ is the feature set, $Q = \{q_1, ..., q_{|Q|}\}$ is the query set, and $D = \{d_1, ..., d_{|D|}\}$ is the document set. In the training process, the learning algorithm takes training data as inputs. In IR, the training data $\{(q_i, d_j), y_{i,j}\}$ comprise query-document pairs, each pair $(q_i, d_j) \in Q \times D$ is associated with a relevance label $y_{i,j}$ that indicates the relationship between q_i and d_j. Each query-document pair is modeled by a vector in an $|F|$-dimensional feature space, and each component of the vector denotes the degree of relevance of document d_j to query q_i respecting feature f_k. The training process aims to learn a ranking model (or function) f from the training data and $f(q_i, d_j)$ is assumed to assign the "true" relevance judgment for q_i and d_j. In the testing process, the model f is utilized to decide the relevance between a new query q and each document d_i in D. Then, sorting documents based on the relevance judgments constructs the document ranking list for query q.

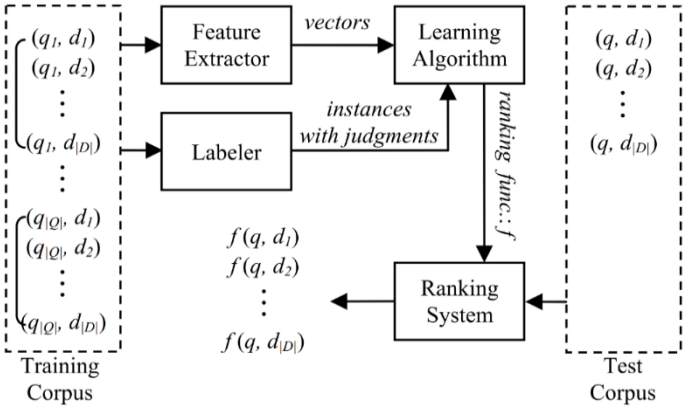

Fig. 1. Framework of LTR for IR [69]

Existing literature has explored three categories of LTR methods [40]: pointwise approaches, pairwise approaches, and listwise approaches.‡ In *pointwise* approaches, the relevance label associated with each instance (q_i, d_j) is either a class of relevance or a

‡ See [40] which provides a comprehensive survey of the literature.

relevance score (ordinal or numerical). The goal is to find a model that assigns each instance a class or a relevance score as close as possible to the instance's true class or relevance score. There are three main streams: classification-based methods (e.g., [43] and McRank [38]) and ordinal regression-based methods (e.g., [55] and PRank [14]) for dealing with classes of relevance; and regression-based methods (e.g., 13) for tackling a relevance score. The *pairwise* approach is based on learning pairwise preferences. This approach views a pair of instances, (q_i, d_j) and (q_i, d_k), as a new single instance and learns a binary classifier that can predict the preference between d_j and d_k for q_i. Example algorithms include Ranking SVM (or RankSVM for short) [27][30], RankBoost [21], RankNet [8], LambdaRank [7], and FRank [62]. The *listwise* approach takes the document ranking lists as instances and builds a model that can directly produce the ordered list (or permutation) of the documents according to a score assigned to every document. Most methods of this type focus on the direct optimization of ranking performance (e.g., AdaRank [67], SVM-MAP [72], SoftRank [61], and PermuRank [68]) or on permutations count (e.g., ListNet [9], ListMLE [66], RankCosine [52], and BoltzRank [63]).

2.2. Feature Selection Methods for LTR

Three general categories of feature selection methods for LTR are filter, wrapper, and embedded approaches. A *filter* approach performs feature ranking based on a relevance criterion. As a preprocessing step, it selects subsets of features independently of the chosen LTR algorithm. Feature selection for ranking was pioneered in [22], which addressed a multi-objective optimization problem in greedily finding a feature subset with minimum total similarity scores and maximum total importance scores. Two method variants, GAS-E and GAS-L, were proposed that utilize performance measures and loss functions in ranking, respectively, to assess feature importance. A hierarchical feature selection strategy was developed in [28] by which clusters of features are constructed and the best performing feature is selected from each cluster. RankFilter [71] extends Relief [32] to compute feature weights from multi-level relevance judgments. In [24], the authors selected the subset of features according to their expected divergence over relevance classes and their importance derived from evaluation scores. The work in [42] exploited greedy result diversification techniques, including maximal marginal relevance (MMR), max-sum dispersion (MSD), and modern portfolio theory (MPT). In [56], the subset of features was selected via minimum redundancy maximum relevance (mRMR) based on their importance and similarity. [23] devised several algorithms, including NGAS that greedily selects the subset of features by minimizing similarity and maximizing relevance, XGAS (an extension of NGAS) that considers more features at each selection iteration, and HCAS that selects the feature with the largest relevance score from each feature cluster, built through hierarchical clustering. In [49], an architecture-agnostic neural feature selection approach was proposed based on a neural LTR model. The approach consists of neural model training, feature group mining based on saliency map, and feature selection based on hierarchical clustering.

With an LTR algorithm as a "black box," a *wrapper* approach scores subsets of features according to their ranking performance. In [28], the authors proposed a

hierarchical feature selection strategy that builds feature clusters with a linear ranking model trained per cluster to select the feature of the highest model weight. Methods using boosted regression trees were explored in [47], including two greedy approaches (selecting the features with the highest relative importance as computed by boosted trees and discounting importance by feature similarity) and a randomized approach with feature-importance-based backward elimination. RankWrapper [71] extends Relief [32] to compute the feature weights from relative orderings. The best first search was used in [15] to greedily partition features into subsets and coordinate ascent was then used to combine features in each subset into one single feature. Greedy RankRLS [45] selects the feature subset of the maximal ranking performance for RankRLS [46] based on greedy forward selection and leave-query-out cross-validation. In [39], language modeling smoothing approaches with different parameters were proposed for selecting the ranking features. [16] considered a multi-objective Pareto-efficient method that optimizes both risk-sensitive evaluation and ranking performance. MOFSRank [11] is a multi-objective evolutionary algorithm consisting of an instance selection strategy, a multi-objective feature selection algorithm, and an ensemble strategy. [17] adopted forward stepwise selection and chose Akaike's information criterion [1] to decide which feature to be added to the selected subset. In [4], a subset of features was viewed as a state in the search space, and simulated annealing was utilized to find the best subset of features.

In an *embedded* approach, the feature selection procedure is integrated into the LTR algorithm. SuperSelRank [33] is a general framework for sparse LTR based on a hierarchical Bayesian model. RSRank [59] performs ℓ_1 regularization using truncated gradient descent to achieve sparsity in ranking models. FenchelRank [34], a primal-dual algorithm for sparse LTR, minimizes the ℓ_1 regularized pairwise ranking loss while simultaneously conducting model selection. SparseRank [35] is a gradient descent algorithm for minimizing the ranking errors with the ℓ_1 regularization. FSMRank [36] is a one-stage method for solving a joint convex optimization problem in which the ranking errors are minimized and meanwhile feature selection is conducted. A general framework using SVM (support vector machines) with sparse regularizations to handle nonconvex penalties was presented in [37]. EGRank [18] uses exponentiated gradient updates to solve a convex optimization problem on a sparsity-promoting ℓ_1 constraint and a pairwise ranking loss. In [53], a deep neural LTR model was provided. The authors used group ℓ_1 regularization to optimize the weights of a neural network, select the relevant features with active neurons at the input layer, and remove inactive neurons from hidden layers. The work in [19] incorporated the ℓ_1 regularized sparse term into the cost-sensitive ListMLE model proposed in [41], and an efficient proximal gradient descent learning method with adaptive Lipschitz constant was applied to obtain the global optimal parameters of the model.

Feature extraction is another technique for dimension reduction. In contrast to feature selection which selects a subset of the original features, feature extraction creates a small set of new features to represent the input data by merging or transforming the original features. LifeRank [48], for instance, views the input dataset as a matrix and constructs a new low-rank dataset with the projection of a transformation matrix that is optimized for the original dataset by minimizing the pairwise ranking loss. More examples of feature extraction methods for LTR can refer to [2] and [20].

3. Proposed Method

FS-SCPR identifies a subset of features that can accurately represent the data, reduce the complexity of the feature space, and enhance performance in ranking problems. This study develops a new LTR for IR approach by extending the framework of LTR for IR in Fig. 1 with the proposed feature selection method, FS-SCPR. See Fig. 2.

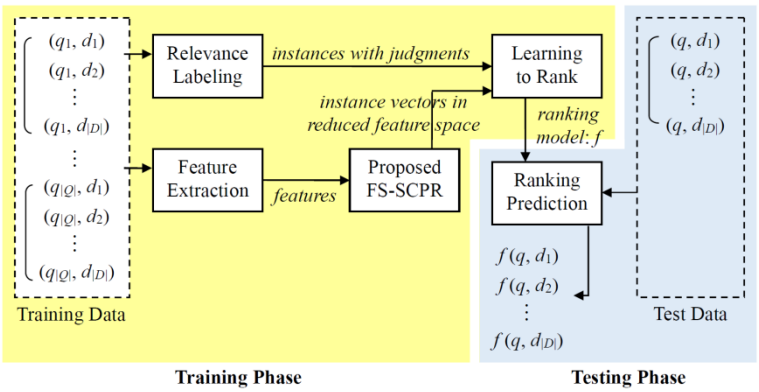

Fig. 2. The proposed LTR for IR approach that incorporates FS-SCPR

3.1. Relevance Labeling

Relevance labeling, which is often done by human annotators, assigns each instance a proper relevance judgment, which plays the role of answers (or observations) that guide the learning algorithm to learn an effective ranking model. Possible relevance judgments include (1) a class; (2) an ordinal rating; (3) a ranking order; and (4) a relevance score [69]. For the labeling scheme, this study adopts an n-star rating. To be specific, each relevance label $y_{i,j} \in \{0, 1, \ldots, n-1\}$, 0 indicates not relevant, $n-1$ means definitely relevant, and higher $y_{i,j}$ indicates higher relevance.

3.2. Feature Extraction

Feature extraction transforms the data into numerical values of ad hoc features. Let fv_k be the feature extraction function for feature f_k, and $w_{i,j,k} = \text{fv}_k(q_i, d_j)$ denote the degree of relevance of document d_j to query q_i respecting feature f_k. The value of $w_{i,j,k}$ is normalized via query-level min-max normalization as follows:

$$w_{i,j,k} = \frac{\text{fv}_k(q_i, d_j) - \min\{\text{fv}_k(q_i, d_l)\}}{\max\{\text{fv}_k(q_i, d_l)\} - \min\{\text{fv}_k(q_i, d_l)\}}, \tag{1}$$

where all $d_l \in D$, $\min\{\cdot\}$ and $\max\{\cdot\}$ respectively stand for the minimum and maximum values of $\text{fv}_k(q_i, d_l)$.

The extracted features in this study cover low-level content features (e.g., the occurrences of a query term in a document and the document length), high-level content features (e.g., BM25 [54] and LMIR [73]), and other features (e.g., the number of out- or in-links of a webpage and the PageRank centrality [6] of a webpage). See Section 4.1.

3.3. Proposed Feature Selection Method, FS-SCPR

The proposed feature selection method, FS-SCPR, is a filter approach. It targets at selecting a subset of features that have minimum redundancy with each other and have maximum relevance to the ranking problem. To minimize redundancy, FS-SCPR drops redundant features, which are grouped into the same cluster. To maximize relevance, FS-SCPR greedily collects a representative feature with high relevance to the ranking problem from each cluster. To produce a feature subset F^* ($F^* \subseteq F$), the process flow of FS-SCPR (see Fig. 3) involves the steps below.

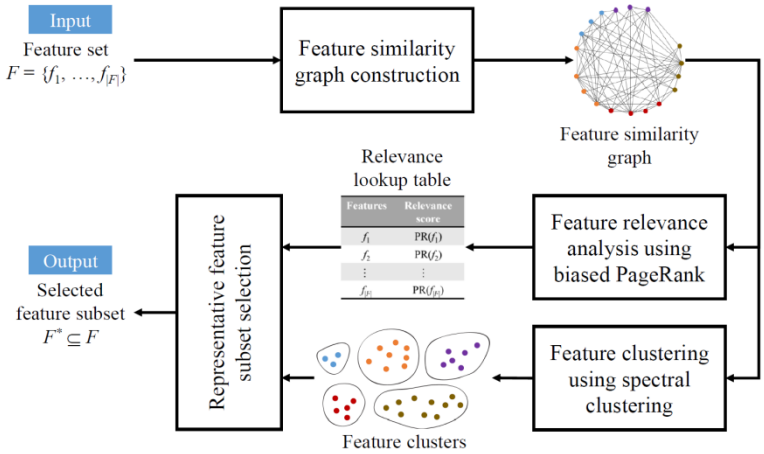

Fig. 3. Process flow of FS-SCPR

1. *Feature similarity graph construction*. Features are modeled as an undirected feature similarity graph. A vertex refers to a feature and an edge indicates that the corresponding features relate to each other. The pairwise feature similarity is measured relying on the correlation of two features' ranking results.
2. *Feature clustering using spectral clustering*. To find redundant features, similar features are grouped into clusters. This study applies spectral clustering [44] that groups data based on eigenvectors of matrices extracted from the feature similarity graph.
3. *Feature relevance analysis using biased PageRank*. The PageRank [6] centrality is utilized to capture the relative "importance" of features by analyzing the link structure of the feature similarity graph. This study further incorporates the ranking performance of each feature as preference to bias the

PageRank computation, giving each feature a more accurate relevance score with respect to the ranking problem.

4. *Representative feature subset selection*. Representative features are selected from each cluster to form the feature subset F^*. For each cluster, this study selects the feature that not only has the highest relevance score but also contains most information of the features in the cluster.

With the feature subset F^* comprising an $|F^*|$-dimensional space, every query-document pair (q_i, d_j) is depicted as a vector in the *reduced* feature space. Every component of the vector is obtained using Eq. (1).

Feature Similarity Graph Construction

Given a query q, there are $|F|$ document ranking lists $\{R_{q,1}, \ldots, R_{q,|F|}\}$. Here, $R_{q,i}$ is established by sorting (in descending order) the retrieved documents D_q according to their feature values regarding feature f_i. Widely-used non-parametric measures of ordinal association, e.g., Spearman's *rho* (ρ) [57] and Kendall's *tau* (τ) [31], can assess the degree of correlation (or similarity) between two ranking lists. This study refers to the correlation between two document ranking lists as the similarity between two features with respect to the given query.

This study chooses Kendall's τ. For two document ranking lists $R_{q,i}$ and $R_{q,j}$, the Kendall's τ value is computed as

$$\tau(R_{q,i}, R_{q,j}) = \frac{|\{(d_s, d_t) \mid d_s \prec_{R_{q,i}} d_t \text{ and } d_s \prec_{R_{q,j}} d_t\}|}{|\{(d_s, d_t)\}|}, \quad \textbf{(2)}$$

where d_s, $d_t \in D_q$, (d_s, d_t) represents a document pair, $d_s \prec_{R_{q,i}} d_t$ denotes that d_t is ranked ahead of d_s in $R_{q,i}$. For a set of queries, the overall similarity between features f_i and f_j is defined in Eq. (3) as the average of their Kendall's τ values for all the queries:

$$sim(f_i, f_j) = \frac{1}{|Q|} \sum_{q \in Q} \tau(R_{q,i}, R_{q,j}). \quad \textbf{(3)}$$

Given the pairwise similarities between features, this study thus represents features as an undirected similarity graph $G = (V, E)$. A vertex denotes a feature, i.e., $V = \{f_1, ..., f_{|F|}\}$, and $E \subseteq V \times V$. Two vertices f_i and f_j are connected if $sim(f_i, f_j) \geq \sigma$[§], and the edge weight is given by $sim(f_i, f_j)$. The graph G can be represented by an adjacency matrix $W = [w_{i,j}]_{i,j=1, \ldots, |F|}$, and each element $w_{i,j}$ is denoted by

$$w_{i,j} = \begin{cases} sim(f_i, f_j) & \text{if } i \neq j \text{ and } sim(f_i, f_j) \geq \sigma \\ 0 & \text{otherwise} \end{cases}. \quad \textbf{(4)}$$

Note that the matrix W is symmetric since $w_{i,j} = w_{j,i}$ holds.

[§] This study empirically sets σ to 0.1.

Feature Clustering Using Spectral Clustering

To cluster features into k subsets, this study applies the normalized spectral clustering algorithm in [44]. The algorithm uses k eigenvectors of a normalized graph Laplacian simultaneously for spectral graph partitioning, as the eigenvectors carry clustering information. By the spectral graph theory [12], the normalized graph Laplacian matrix L is formulated as

$$L = A^{-1/2}(A-W)A^{-1/2} = I - A^{-1/2}WA^{-1/2}, \quad (5)$$

where W is the aforementioned adjacency matrix, I is the unit matrix, and A = diag(a_1, …, $a_{|F|}$) is the diagonal matrix whose every diagonal element $a_i = \sum_j^{|F|} w_{i,j}$. Since both matrices W and A are symmetric real matrices, L is also symmetric and real. Additionally, L has $|F|$ eigenvalues, λ_1, …, $\lambda_{|F|}$, and $0 = \lambda_1 \leq ... \leq \lambda_{|F|}$.

In our problem, the inputs contain the feature set F accompanying the matrix W and the number k of clusters to build. The output is the set of clusters of features {C_1, …, C_k}. Algorithm 1 states the steps of the normalized spectral clustering algorithm.

Algorithm 1 Normalized Spectral Clustering [44]

Input: The feature set F = {f_1, …, $f_{|F|}$} accompanying the matrix W and the number k of clusters to build.

Output: The feature clusters {C_1, …, C_k}. $\forall f_i, \exists j$ s.t. $f_i \in C_j$.

Procedure:

1. Compute the matrix $L = I - A^{-1/2}WA^{-1/2}$.
2. Build the matrix $X = [x_1\ x_2\ \ldots\ x_k] \in \mathbb{R}^{|F|\times k}$ whose columns x_1, …, x_k are the k smallest eigenvectors of L.
3. Build from X the matrix $Y \in \mathbb{R}^{|F|\times k}$ whose every element $y_{i,j} = \frac{x_{i,j}}{\sqrt{\sum_j x_{i,j}^2}}$.
4. Let Y's every row be a data point in $\mathbb{R}^k$, and build k clusters via bisecting K-means.
5. Assign feature f_i to cluster C_j if Y's row i is in cluster C_j.

There are two points to note. First, [44] constructs the matrix $I - L$ in Step 1, which only changes the eigenvalues (from λ_i to $1 - \lambda_i$) and not the eigenvectors. Thus, in Step 2, [44] finds the k largest eigenvectors (referring to the k largest eigenvalues), we instead consider the k smallest eigenvectors (referring to the k smallest eigenvalues). Second, [44] uses K-means in Step 4. This study utilizes bisecting K-means [58] because it in practice produces better-quality clustering results (see [60]).

The trick of spectral clustering is to embed the data in a low-dimensional space wherein the data's cluster properties become prominent. The method's success is mainly owing to that no assumptions are made on the form of the clusters and their statistics [64] (as opposed to, for example, K-means, where the clusters are convex sets). Thus, spectral clustering very often outperforms conventional clustering algorithms. Additionally, spectral clustering is simple to implement, can be solved efficiently by standard linear algebra software, is efficient to obtain near-optimal partitions, and is reasonably fast for large sparse data sets [64]. Furthermore, spectral clustering does not necessarily need the data in the embedded form (i.e., featured objects) [65]. The data can be represented as relationships between objects, as in this work features are

modeled by a feature similarity graph. For these reasons, we choose spectral clustering to obtain feature clusters instead of conventional clustering methods.

Feature Relevance Analysis Using Biased PageRank

This study assesses feature relevance to the ranking problem via biased PageRank [26]. Given a feature similarity graph $G = (V, E)$, each vertex (i.e., feature in this context) is scored by applying biased PageRank on the graph. The score s for a vertex f_i is assigned by the recursive equation

$$s(f_i) = (1-\alpha) \times p(f_i) + \alpha \times \sum_{f_j \in M(f_i)} \frac{w_{i,j}}{\sum_{f_k \in M(f_j)} w_{j,k}} \times s(f_j), \quad (\mathbf{6})$$

where α is a damping factor ($0 \le \alpha < 1$)[**], $p(f_i)$ is the preference weight[††] assigned to vertex f_i, and $M(f_i)$ is the set of those vertices that have links to the vertex f_i.

The feature relevance analysis approach iterates until convergence is achieved. When iterations stop, a score is associated with every vertex as its feature relevance. In each iteration, function $p(\cdot)$ introduces additional preferences to the appropriate vertices. By selecting the appropriate preference weights, the PageRank computation can be made to prefer certain vertices. This study assigns larger preference weights to those features that have better ranking performance. The idea is to incorporate ranking performance into the biased PageRank to better capture feature relevance. This study uses MAP (see Section 4.2 for its definition) for $p(\cdot)$. Note that the preference weights of features are normalized by $\frac{p(f_i)}{\sum_j p(f_j)}$ before they are used in Eq. (6).

Representative Feature Subset Selection

After feature clustering, redundant features are grouped into the same cluster. Additionally, each feature is scored for its relevance to the ranking problem after feature relevance analysis. This study selects one representative feature from each cluster, according to which feature that not only has the highest relevance score but also contains most information of the features in the cluster. All the other features in the cluster are discarded. Thus, the resulting feature subset contains features that have minimum redundancy with each other and have maximum relevance to the ranking problem.

Algorithm 2 depicts the steps of our feature subset selection approach. In Step 2.1, this study measures the similarity between features using the matrix Y in Algorithm 1. To determine which feature in a cluster has the highest relevance score and contains most information of the other features, we deal with the multi-objective problem using a linear combination of a feature's relevance score and its sum of pairwise similarities to the other features, as shown in Step 2.2.

[**] This study sets α to 0.85 according to [6].

[††] In the original PageRank [6], each vertex is weighted with an equal preference of $1/|V|$.

Algorithm 2 Representative Feature Subset Selection

Input: The feature clusters $\{C_1, \ldots, C_k\}$.
Output: The selected feature subset F^*.
Procedure:

1. Set F^* to an empty set, $F^* = \varnothing$.
2. For each cluster C_i, do:
   2.1. For each feature f in C_i, compute *SSim*(f), i.e., the sum of its pairwise similarities to the other features in C_i.
   2.2. From C_i, find feature f that has not only the largest *SSim*(f) but also the highest feature relevance, as scored by Eq. (6). That is, $f = \arg\max\left[0.5\times s(f) + 0.5\times \frac{SSim(f)}{|C_i|-1}\right]$. Note that *SSim*($f$) is normalized by $|C_i| - 1$.
   2.3. Assign feature f to F^*, i.e., $F^* = F^* \bigcup \{f\}$.

3.4. Ranking Model Learning and Prediction

This study employs Ranking SVM [27][30] to derive a ranking model since previous studies have demonstrated its feasibility and effectiveness. Ranking SVM views the LTR problem as binary classification on pairs of documents and applies SVM (support vector machines) to solve the classification problem. In other words, Ranking SVM targets binary ordering relations between documents with respect to queries and learns, based on parts of the observations of the target (or optimal) ranking lists, a model that minimizes the count of discordant pairs. Considering the class of *linear* ranking functions, the following optimization problem is solved in Ranking SVM [30]:

$$\begin{aligned} &minimize: \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum \xi_{i,j,q} \\ &subject\ to: \\ &\quad \forall q, \forall (d_i, d_j) \in r_q^* : \vec{w}\Phi(q, d_i) \geq \vec{w}\Phi(q, d_j) + 1 - \xi_{i,j,q} \\ &\quad \forall i \forall j \forall q : \xi_{i,j,q} \geq 0 \end{aligned} \quad (7)$$

Here, the weight vector $\vec{w}$ is arranged in learning; C trades-off between margin and training error; $\xi_{i,j,q}$ is a non-negative slack variable; r_q^* is the target ranking list, given query q; and $\Phi(q, d_i)$ is a feature vector that depicts the relevance of document d_i to query q in terms of features.

For all the queries, pairs of instances and their relative preferences are inputted into Ranking SVM for training. Note that each instance is modeled as a vector in the reduced feature space (see Section 3.3). Regarding ranking prediction, the learned ranking model decides for a new query whether pairs of documents are in concordant order. The final document ranking list can thus be established according to the outputted binary ordering relations between documents.

4. Evaluation

4.1. Datasets

To evaluate the performance and effectiveness of the proposed LTR for IR approach, we conducted experiments on the publicly available LETOR[‡‡] benchmark collections. We selected the following four datasets: HP2004, NP2004, OHSUMED, and MQ2008. The first three datasets are from LETOR 3.0 and the last one is from LETOR 4.0. The datasets come as query-document pairs. A pair contains a feature vector and its relevance judgment. For cross-validation, each dataset is split into five subsets. In each fold, three subsets are used for learning, one subset for validation, and the other one for testing. See [51] and [50] for details on the selection of document corpora, the sampling of documents, the extraction of features and meta-information, and the finalization of datasets. Table 1 depicts the statistics of the datasets. Table 2 illustrates some sample data; each row stands for a query-document pair.

Table 1. Statistics of the datasets. For HP2004 and NP2004, the relevance judgments are on two levels (relevant and not relevant); for OHSUMED and MQ2008, the relevance judgments are on three levels (definitely relevant, possibly relevant, and not relevant)

	HP2004	NP2004	OHSUMED	MQ2008
No. of queries	75	75	106	784
No. of query-document pairs (i.e., instances)	74,409	73,834	16,140	15,211
No. of features	64	64	45	46
Relevance levels	2	2	3	3

Table 2. Sample data excerpted from MQ2008

Label	Query	f_1	…	f_{46}	Note
2	qid:10032	1:0.056537	…	46:0.076923	#doc: GX029-35-5894638
0	qid:10032	1:0.279152	…	46:1.000000	#doc: GX030-77-6315042
0	qid:10032	1:0.130742	…	46:1.000000	#doc: GX140-98-13566007
1	qid:10032	1:0.593640	…	46:0.000000	#doc: GX256-43-0740276
⋮	⋮	⋮	⋮	⋮	⋮

4.2. Evaluation Measures

We use two common measures, namely, MAP (mean average precision) [5] and NDCG (normalized discounted cumulative gain) [29], as the evaluation measures.

[‡‡] https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/.

Eq. (8) denotes the average precision (AvgP) for a query, and for all the queries the mean of their average precisions is the MAP.

$$\text{AvgP} = \frac{\sum_{n=1}^{N} \text{P@}n \times rel(n)}{\text{\# of relevant documents for the query}}, \tag{8}$$

In the equation, N is the number of retrieved documents, P@n (namely, precision at position n) is the fraction of relevant documents among the top n results, and $rel(n) \in \{0, 1\}$ implies that the document at position n is relevant or not.

For a query's ranking list, the NDCG at position n is calculated by

$$\text{NDCG@}n = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{\log_2(1+j)}, \tag{9}$$

in which Z_n is a normalization parameter that allows producing an NDCG of 1.0 for the perfect list, and $r(j)$ means the rating of the document at position j. The NDCG@n values for all queries are averaged and reported.

We present the results of NDCG@1, NDCG@3, NDCG@5, NDCG@10, and MAP for comparisons.

4.3. Experimental Setup

We conducted experiments to verify whether FS-SCPR helps improve the ranking performance and to understand whether FS-SCPR outperforms other baseline feature selection methods and state-of-the-art LTR approaches. Five-fold cross-validation is conducted, and all the presented results are the average performance on the testing set. In each fold, we use the training set to select features, and train a ranking model from the training set with the selected features. The validation set is utilized for parameter tuning and model selection. The above two steps are repeated to identify the best ranking model. Then, the obtained ranking model is evaluated on the testing set.

For simplicity, we denote the proposed approach as FS-SCPR and use "feature selection" and "feature selection for LTR" interchangeably for the remainder of this paper. Additionally, for efficient learning, we use RankSVM-Primal [10] (an efficient version of Ranking SVM) instead of Ranking SVM.

4.4. Baseline Algorithms

We tested two groups of baseline algorithms.[§§] The first group tested LTR methods without using feature selection. This study selects AdaRank-MAP (a listwise method) [67], RankSVM-Primal (a pairwise method) [10], ListNet (a listwise method) [9], and RankBoost (a pairwise method) [21]. AdaRank-MAP, RankSVM-Primal, and ListNet learn linear ranking models, while RankBoost learns a non-linear ranking model.

The second group tested feature selection methods, including GAS-E (a filter method)

[§§] The presented results of the baselines are cited from the LETOR datasets and the original papers.

[22], FSMSVM (a wrapper method) [36], and FSMRank (an embedded method) [36]. See Section 2.2 for a brief description of these methods. As the proposed approach adopts RankSVM-Primal as the learning algorithm, the implementation of GAS-E in this work also chooses RankSVM-Primal (instead of Ranking SVM or RankNet used in [22]). As a simple feature selection method, FSMSVM selects top features with large weights according to their weights in a pre-trained model (which in [36] is learned by FSMRank) and uses the selected features to learn a ranking model by RankSVM-Primal.

4.5. Results

Comparison with RankSVM-Primal

This experiment compares the ranking performance of FS-SCPR with RankSVM-Primal. FS-SCPR considers only the selected features, while RankSVM-Primal uses all the features. The objective of this experiment is to empirically justify whether the proposed feature selection method helps enhance the performance of ranking predictions. Tables 3–6 present the results on four datasets. The row named "Imp." in each table denotes the relative improvement[***] of FS-SCPR versus RankSVM-Primal.

Table 3. Ranking performance of FS-SCPR and RankSVM-Primal on HP2004 (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.6337**	**0.7590**	**0.7893**	**0.8179**	**0.7216**
RankSVM-Primal	0.5733	0.7129	0.7528	0.7720	0.6712
Imp.	+10.54%	+6.47%	+4.85%	+5.95%	+7.51%

Table 4. Ranking performance of FS-SCPR and RankSVM-Primal on NP2004 (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.5527	**0.7603**	**0.7842**	**0.8159**	**0.6809**
RankSVM-Primal	**0.5600**	0.7236	0.7719	0.7950	0.6755
Imp.	–1.3%	+5.07%	+1.59%	+2.63%	+0.8%

Table 5. Ranking performance of FS-SCPR and RankSVM-Primal on OHSUMED (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.5459	**0.4959**	**0.4785**	**0.4584**	**0.4491**
RankSVM-Primal	**0.5460**	0.4855	0.4689	0.4504	0.4446
Imp.	–0.02%	+2.14%	+2.05%	+1.78%	+1.01%

[***] When b is compared to a, the relative improvement is calculated as $(b - a) / a \times 100\%$.

Table 6. Ranking performance of FS-SCPR and RankSVM-Primal on MQ2008 (best performance bold-faced)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.3692	**0.4375**	**0.4770**	**0.2318**	**0.4776**
RankSVM-Primal	**0.3725**	0.4333	0.4765	0.2309	0.4744
Imp.	–0.89%	+0.97%	+0.1%	0.39%	+0.67%

The results of FS-SCPR are significantly boosted by the proposed feature selection method. Looking at NDCG@10, the performance of FS-SCPR is enhanced by 5.95% on HP2004, by 2.63% on NP2004, by 1.78% on OHSUMED, and by 0.39% on MQ2008. In terms of MAP, FS-SCPR has relative increases of 7.51% on HP2004, 0.8% on NP2004, 1.01% on OHSUMED, and 0.67% on MQ2008. Similar enhancements can be seen in other measures. For each dataset, the maximum enhancements over distinct measures are increases of 10.54% in NDCG@1 on HP2004, 5.07% in NDCG@3 on NP2004, 2.14% in NDCG@3 on OHSUMED, and 0.97% in NDCG@3 on MQ2008. However, there are few exceptions, including the NDCG@1 scores on NP2004, OHSUMED, and MQ2008. In these cases, the performance of FS-SCPR deteriorated by 1.3%, 0.02%, and 0.89%, respectively, compared to the performance of RankSVM-Primal.

Comparison with Feature Selection Methods

This experiment focuses on understanding how effectively FS-SCPR performs compared to other feature selection methods. Tables 7–10 present the comparison results. In each column, the methods are ranked by their scores, and the rankings are shown in parentheses.

First, FS-SCPR is observed in most cases to have superior performance to GAS-E (a filter method) and FSMSVM (a wrapper method). The few exceptions are the cases of NDCG@1 on NP2004 and OHSUMED and the case of NDCG@5 on MQ2008. Taking NDCG@10 as an example, FS-SCPR outperforms GAS-E by 4.2%, 2.53%, 1.82%, and 1.22% on HP2004, NP2004, OHSUMED, and MQ2008, respectively. Compared to FSMSVM, FS-SCPR has performance gains of 3.81%, 1.23%, 3.08%, and 2.98% in NDCG@10 on HP2004, NP2004, OHSUMED, and MQ2008, respectively. Regarding MAP, FS-SCPR outperforms GAS-E by 4.2%, 1.08%, 0.36%, and 0.19% on HP2004, NP2004, OHSUMED, and MQ2008, respectively. Compared to FSMSVM, FS-SCPR has performance gains of 2.88%, 0.9%, 1.13%, and 0.67% in MAP on HP2004, NP2004, OHSUMED, and MQ2008, respectively.

Second, compared to FSMRank (an embedded method), FS-SCPR performs competitively only in a few cases. For instance, it outperforms FSMRank in MAP on HP2004 and MQ2008 with increases of 0.15% and 0.1%, respectively. As another example, its NDCG@1 scores on the four datasets are superior to those of FSMRank with increases of 3.33% on HP2004, 1.1% on NP2004, 1.21% on OHSUMED, and 0.16% on MQ2008. The comparison results in most cases demonstrate that FS-SCPR does not perform better than FSMRank, especially in NDCG@3, NDCG@5, and NDCG@10. These observations are not unexpected since an embedded method that

conducts feature selection inside the LTR algorithm generally tends to have superior performance to a filter method that selects features independently of the LTR algorithm.

Finally, from an overall perspective, the experimental results show that FS-SCPR practically performs well. In terms of MAP, FS-SCPR ranks first on HP2004 and MQ2008 and ranks second on NP2004 and OHSUMED. Considering NDCG@10, FS-SCPR is the second best performer on the four datasets. To further identify which method demonstrates the best results in various measures on different datasets, Table 11 presents a unified ranking of the methods. According to [3], the unified rank of a method is defined by

$$Rank = \sum_{r=1}^{M} \frac{(M - r + 1) \times R_r}{M}, \tag{10}$$

in which M is the number of compared methods and R_r is the count the method appears in the r-th rank. From Table 11, a unified ranking of the methods is obtained: FSMRank $\succ$ FS-SCPR $\succ$ GAS-E $\succ$ FSMSVM, in which the proposed FS-SCPR ranks second.

Table 7. Ranking performance of FS-SCPR and other feature selection for LTR methods on HP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.6337** (1)	*0.7590* (2)	*0.7893* (2)	*0.8179* (2)	**0.7216** (1)
GAS-E	0.6133 (3)	0.7280 (3)	0.7679 (3)	0.7849 (4)	0.6925 (4)
FSMSVM	*0.6267* (2)	0.7136 (4)	0.7635 (4)	0.7879 (3)	0.7014 (3)
FSMRank	0.6133 (3)	**0.8070** (1)	**0.8187** (1)	**0.8383** (1)	*0.7205* (2)

Table 8. Ranking performance of FS-SCPR and other feature selection for LTR methods on NP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	*0.5527* (2)	*0.7603* (2)	*0.7842* (2)	*0.8159* (2)	*0.6809* (2)
GAS-E	**0.5600** (1)	0.7236 (4)	0.7617 (4)	0.7958 (4)	0.6736 (4)
FSMSVM	0.5467 (3)	0.7538 (3)	0.7830 (3)	0.8060 (3)	0.6748 (3)
FSMRank	0.5467 (3)	**0.7784** (1)	**0.8000** (1)	**0.8279** (1)	**0.6837** (1)

Table 9. Ranking performance of FS-SCPR and other feature selection for LTR methods on OHSUMED (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.5459 (3)	*0.4959* (2)	*0.4785* (2)	*0.4584* (2)	*0.4491* (2)
GAS-E	**0.5547** (1)	0.4794 (3)	0.4720 (3)	0.4502 (3)	0.4475 (3)
FSMSVM	*0.5492* (2)	0.4690 (4)	0.4640 (4)	0.4447 (4)	0.4441 (4)
FSMRank	0.5394 (4)	**0.5013** (1)	**0.4824** (1)	**0.4613** (1)	**0.4498** (1)

Table 10. Ranking performance of FS-SCPR and other feature selection for LTR methods on MQ2008 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.3692** (1)	*0.4375* (2)	0.4770 (3)	*0.2318* (2)	**0.4776** (1)
GAS-E	0.3601 (4)	0.4345 (3)	*0.4772* (2)	0.2290 (3)	0.4767 (3)
FSMSVM	0.3652 (3)	0.4278 (4)	0.4701 (4)	0.2251 (4)	0.4744 (4)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FSMRank	*0.3686* (2)	**0.4399** (1)	**0.4791** (1)	**0.2327** (1)	*0.4771* (2)

Table 11. Unified ranking of methods (the higher Rank value, the better)

	$R_r = 1$	$R_r = 2$	$R_r = 3$	$R_r = 4$	Rank
FS-SCPR	4	14	2	0	15.5
GAS-E	2	1	10	7	9.5
FSMSVM	0	2	8	10	8
FSMRank	14	3	2	1	17.5

Comparison with State-of-the-Art LTR Methods

This experiment compares the ranking performance of FS-SCPR with other state-of-the-art LTR methods (all without using feature selection). Tables 12–15 present the results. In each column, the methods are ranked by their scores, and the rankings are shown in parentheses.

Table 12. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on HP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	**0.6337** (1)	*0.7590* (2)	*0.7893* (2)	*0.8179* (2)	*0.7216* (2)
AdaRank-MAP	*0.6133* (2)	**0.8164** (1)	**0.8277** (1)	**0.8328** (1)	**0.7219** (1)
RankSVM-Primal	0.5733 (4)	0.7129 (4)	0.7528 (4)	0.7720 (4)	0.6712 (4)
ListNet	0.6000 (3)	0.7213 (3)	0.7694 (3)	0.7845 (3)	0.6899 (3)
RankBoost	0.5067 (5)	0.6989 (5)	0.7211 (5)	0.7428 (5)	0.6251 (5)

Table 13. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on NP2004 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	*0.5527* (2)	**0.7603** (1)	*0.7842* (2)	**0.8159** (1)	**0.6809** (1)
AdaRank-MAP	0.4800 (4)	0.6979 (4)	0.7310 (4)	0.7497 (4)	0.6220 (4)
RankSVM-Primal	**0.5600** (1)	0.7236 (3)	0.7719 (3)	0.7950 (3)	*0.6755* (2)
ListNet	0.5333 (3)	*0.7587* (2)	**0.7965** (1)	*0.8128* (2)	0.6720 (3)
RankBoost	0.4267 (5)	0.6274 (5)	0.6512 (5)	0.6914 (5)	0.5640 (5)

Table 14. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on OHSUMED (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	*0.5459* (2)	**0.4959** (1)	**0.4785** (1)	**0.4584** (1)	**0.4491** (1)
AdaRank-MAP	0.5388 (3)	0.4682 (4)	0.4613 (3)	0.4429 (3)	*0.4487* (2)
RankSVM-Primal	**0.5460** (1)	*0.4855* (2)	*0.4689* (2)	*0.4504* (2)	0.4446 (4)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
ListNet	0.5326 (4)	0.4732 (3)	0.4432 (5)	0.4410 (4)	0.4457 (3)
RankBoost	0.4632 (5)	0.4555 (5)	0.4494 (4)	0.4302 (5)	0.4411 (5)

Table 15. Ranking performance of FS-SCPR and other state-of-the-art LTR methods on MQ2008 (best performance bold-faced; second best in italics)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10	MAP
FS-SCPR	0.3692 (5)	**0.4375** (1)	*0.4770* (2)	**0.2318** (1)	**0.4776** (1)
AdaRank-MAP	*0.3754* (2)	*0.4370* (2)	**0.4794** (1)	0.2288 (4)	0.4764 (4)
RankSVM-Primal	0.3725 (4)	0.4333 (3)	0.4765 (3)	*0.2309* (2)	0.4744 (5)
ListNet	*0.3754* (2)	0.4324 (4)	0.4747 (4)	0.2303 (3)	*0.4775* (2)
RankBoost	**0.3856** (1)	0.4288 (5)	0.4666 (5)	0.2255 (5)	*0.4775* (2)

Table 16. Unified ranking of methods (the higher Rank value, the better)

	$R_r = 1$	$R_r = 2$	$R_r = 3$	$R_r = 4$	$R_r = 5$	Rank
FS-SCPR	11	8	0	0	1	17.6
AdaRank-MAP	5	4	3	8	0	13.2
RankSVM-Primal	2	5	5	7	1	12
ListNet	1	4	10	4	1	12
RankBoost	1	1	0	1	17	5.6

On the different datasets, FS-SCPR ranks either first or second in various measures with the only exception being that it is ranked fifth in NDCG@1 on MQ2008. For example, for OHSUMED, FS-SCPR is the best performer for NDCG@3, NDCG@5, NDCG@10, and MAP, and is ranked second for NDCG@1. We briefly highlight some statistics. In terms of NDCG@10, FS-SCPR performs the best on NP2004, OHSUMED, and MQ2008, and is the second best performer on HP2004. On NP2004, it performs 0.38% higher compared to the second best method (ListNet) and 2.63% higher compared to the third best method (RankSVM-Primal). On OHSUMED, it outperforms the second best method (RankSVM-Primal) by 1.78% and outperforms the third best method (AdaRank-MAP) by 3.50%. On MQ2008, it performs better than the second best method (RankSVM-Primal) with a 0.39% improvement and performs better than the third best method (ListNet) by 0.65%. On HP2004, it outperforms the third best method (ListNet) with a 4.26% improvement.

Regarding MAP, FS-SCPR is ranked first on NP2004, OHSUMED, and MQ2008, and second on HP2004. On NP2004, it is superior to the second best method (RankSVM-Primal) and the third best method (ListNet) by 0.8% and 1.32%, respectively. On OHSUMED, it performs 0.09% better than the second best method (AdaRank-MAP) and performs 0.76% better than the third best method (ListNet). On MQ2008, it outperforms the second best methods (ListNet and RankBoost) by 0.02%. On HP2004, it outperforms the third best method (ListNet) by 4.59%.

Overall, the comparison results indicate that FS-SCPR performs very competitively and has stable performance on ranking on different datasets compared to other baselines. Table 16 demonstrates the following unified ranking of the methods: FS-

SCPR $\succ$ AdaRank-MAP $\succ$ RankSVM-Primal $=$ ListNet $\succ$ RankBoost, in which the proposed FS-SCPR ranks first.

5. Conclusion and Future Work

This paper addresses the feature selection problem in LTR. We proposed a graph-based filter feature selection method, FS-SCPR (see Fig. 3). FS-SCPR selects a subset of features that have minimum redundancy with each other and have maximum relevance to the ranking problem. In practice, FS-SCPR models feature relationships as a feature similarity graph. Based on such a graph model, FS-SCPR selects features using spectral clustering for redundancy minimization and biased PageRank for relevance analysis. Furthermore, we developed a new LTR for IR approach that integrates FS-SCPR and Ranking SVM (see Fig. 2). This approach exploits FS-SCPR as a preprocessor to determine discriminative and useful features and utilizes Ranking SVM to derive a ranking model with the selected features. We evaluated the proposed approach using four LETOR datasets (namely, HP2004, NP2004, OHSUMED, and MQ2008) and found that it performed well with competitive results. We presented the performance gains of the proposed approach compared to representative feature selection methods (namely, GAS, FSMSVM, and FSMRank) and state-of-the-art LTR methods (namely, AdaRank, Ranking SVM, ListNet, and RankBoost). The experimental results showed that (1) FS-SCPR can significantly boost the ranking performance; (2) FS-SCPR has superior performance to GAS (a filter method) and FSMSVM (a wrapper method), and is competitive to FSMRank (an embedded method) in a few cases; and (3) FS-SCPR performs very competitively compared to several LTR baselines and has stable performance on ranking on different datasets.

It is also worth noting that similar to other filter methods, this study tries to find a feature subset with minimum total similarity and maximum total relevance. However, the graph-based selection strategy makes this work quite distinct from the existing studies. Our approach is the first graph-based feature selection technique that uses spectral clustering for redundancy minimization and biased PageRank for relevance analysis. To the best of our knowledge, there was little graph-based attempt to tackle feature selection for LTR and this research contributes to this gap in the literature.

There remains room for improvement. First, it would be valuable to study whether improving relationships between features in the feature similarity graph will directly profit FS-SCPR. Other measures of ordinal association, such as Spearman's *rho* (ρ), are worth exploring to evaluate feature relationships. Second, methods of evaluating the goodness of a clustering can be utilized to help automatically decide the number of feature clusters. Another interesting issue to investigate is what kinds of ranking performance of features besides MAP contribute to FS-SCPR regarding biasing the PageRank computation. FS-SCPR could also be integrated with other learning methods, e.g., AdaRank-MAP. Finally, verifying the effectiveness of FS-SCPR using additional datasets would be beneficial.

References

1. Akaike, H.: Information Theory and an Extension of the Maximum Likelihood Principle. In Proceedings of the 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, 267–281. (1973)
2. Albuquerque, A., Amador, T., Ferreira, R., Veloso, A., Ziviani, N.: Learning to Rank with Deep Autoencoder Features. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN 2018), Rio de Janeiro, Brazil. (2018)
3. Aliguliyev, R. M.: Performance Evaluation of Density-based Clustering Methods. Information Sciences, Vol. 179, No. 20, 3583–3602. (2009)
4. Allvi, M. W., Hasan, M., Rayan, L., Shahabuddin, M., Khan, M. M., Ibrahim, M.: Feature Selection for Learning-to-Rank Using Simulated Annealing. International Journal of Advanced Computer Science and Applications, Vol. 11, No. 3, 699–705. (2020)
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley. (1999)
6. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems, Vol. 30, No. 1–7, 107–117. (1998)
7. Burges, C. J. C., Ragno, R., Le, Q. V.: Learning to Rank with Nonsmooth Cost Functions. In Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS 2006), Vancouver, BC, Canada, 193–200. (2006)
8. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to Rank Using Gradient Descent. In Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany, 89–96. (2005)
9. Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H.: Learning to Rank: From Pairwise Approach to Listwise Approach. In Proceedings of the 24th International Conference on Machine Learning (ICML 2007), Corvallis, OR, 129–136. (2007)
10. Chapelle, O., Keerthi, S. S.: Efficient Algorithms for Ranking with SVMs. Information Retrieval, Vol. 13, No. 3, 201–215. (2010)
11. Cheng, F., Guo, W., Zhang, X.: MOFSRank: A Multiobjective Evolutionary Algorithm for Feature Selection in Learning to Rank. Complexity, Vol. 2018, Article: 7837696. (2018)
12. Chung, F. R. K.: Spectral Graph Theory. American Mathematical Society. (1997)
13. Cossock, D., Zhang, T.: Subset Ranking Using Regression. In Proceedings of the 19th Annual Conference on Learning Theory (COLT 2006), Pittsburgh, PA, 605–619. (2006)
14. Crammer, K., Singer, Y.: Pranking with Ranking. In Proceedings of the 15th Annual Conference on Neural Information Processing Systems (NIPS 2001), Vancouver, BC, Canada, 641–647. (2001)
15. Dang, V., Croft, W. B.: Feature Selection for Document Ranking Using Best First Search and Coordinate Ascent. In Proceedings of the SIGIR 2010 Workshop on Feature Generation and Selection for Information Retrieval, Geneva, Switzerland, 28–31. (2010)
16. de Sousa, D. X., Canuto, S. D., Rosa, T. C., Martins, W. S., Gonçalves, M. A.: Incorporating Risk-Sensitiveness into Feature Selection for Learning to Rank. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016), Indianapolis, IN, 257–266. (2016)
17. Dhake, N., Raut, S., Rahangdale, A.: Identification of Efficient Algorithms for Web Search through Implementation of Learning-to-Rank Algorithms. Sādhanā, Vol. 44, No. 4, Article: 97. (2019)
18. Du, L., Pan, Y., Ding, J., Lai, H., Huang, C.: EGRank: An Exponentiated Gradient Algorithm for Sparse Learning-to-Rank. Information Sciences, Vol. 467, 342–356. (2018)
19. Du, D., Zhou, F., Xiong, W.: Cost-Sensitive ListMLE Ranking Approach Based on Sparse Representation. Journal of Information Science and Engineering, Vol. 35, No. 1, 1–22. (2019)
20. Duh, K., Kirchhoff, K.: Learning to Rank with Partially-Labeled Data. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, 251–258. (2008)

21. Freund, Y., Iyer, R., Schapire, R. E., Singer, Y.: An Efficient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research, Vol. 4, 933–969. (2003)
22. Geng, X., Liu, T.-Y., Qin, T., Li, H.: Feature Selection for Ranking. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 407–414. (2007)
23. Gigli, A., Lucchese, C., Nardini, F. M., Perego, R.: Fast Feature Selection for Learning to Rank. In Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR 2016), Newark, DE, 167–170. (2016)
24. Gupta, P., Rosso, P.: Expected Divergence Based Feature Selection for Learning to Rank. In Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), Mumbai, MH, India, 431–439. (2012)
25. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, Vol. 3, 1157–1182. (2003)
26. Haveliwala, T. H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 4, 784–796. (2003)
27. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Smola, A. J., Bartlett, P. L., Schölkopf, B., Schuurmans, D. (eds.): Advances in Large Margin Classifiers. The MIT Press, 115–132. (2000)
28. Hua, G., Zhang, M., Liu, Y., Ma, S., Ru, L.: Hierarchical Feature Selection for Ranking. In Proceedings of the 19th International Conference on World Wide Web (WWW 2010), Raleigh, NC, 1113–1114. (2010)
29. Järvelin, K., Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems, Vol. 20, No. 4, 422–446. (2002)
30. Joachims, T.: Optimizing Search Engines Using Clickthrough Data. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002), Edmonton, AB, Canada, 133–142. (2002)
31. Kendall, M. G.: A New Measure of Rank Correlation. Biometrika, Vol. 30, No. 1–2, 81–93. (1938)
32. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In Proceedings of the 7th European Conference on Machine Learning (ECML 1994), Catania, Italy, 171–182. (1994)
33. Krasotkina, O., Mottl, V.: A Bayesian Approach to Sparse Learning-to-Rank for Search Engine Optimization. In Proceedings of the 11th International Conference on Machine Learning and Data Mining (MLDM 2015), Hamburg, Germany, 382–394. (2015)
34. Lai, H., Pan, Y., Liu, C., Lin, L., Wu, J.: Sparse Learning-to-Rank via an Efficient Primal-Dual Algorithm. IEEE Transactions on Computers, Vol. 62, No. 6, 1221–1233. (2013)
35. Lai, H., Pan, Y., Tang, Y., Liu, N.: Efficient Gradient Descent Algorithm for Sparse Models with Application in Learning-to-Rank. Knowledge-Based Systems, Vol. 49, 190–198. (2013)
36. Lai, H.-J., Pan, Y., Tang, Y., Yu, R.: FSMRank: Feature Selection Algorithm for Learning to Rank. IEEE Transactions on Neural Networks and Learning Systems, Vol. 24, No. 6, 940–952. (2013)
37. Laporte, L., Flamary, R., Canu, S., Déjean, S., Mothe, J.: Nonconvex Regularizations for Feature Selection in Ranking with Sparse SVM. IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, No. 6, 1118–1130. (2014)
38. Li, P., Burges, C. J. C., Wu, Q.: McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS 2007), Vancouver, BC, Canada, 897–904. (2007)
39. Lin, Y., Lin, H., Xu, K., Sun, X.: Learning to Rank Using Smoothing Methods for Language Modeling. Journal of the American Society for Information Science and Technology, Vol. 64, No. 4, 818–828. (2013)
40. Liu, T.-Y.: Learning to Rank for Information Retrieval. Springer. (2011)

41. Lu, M., Xie, M., Wang, Y., Liu, J., Huang, Y.: Cost-Sensitive Listwise Ranking Approach. In Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2010), Hyderabad, India, 358–366. (2010)
42. Naini, K. D., Altingovde, I. S.: Exploiting Result Diversification Methods for Feature Selection in Learning to Rank. In Proceedings of the 36th European Conference on Information Retrieval (ECIR 2014), Amsterdam, The Netherlands, 455–461. (2014)
43. Nallapati, R.: Discriminative Models for Information Retrieval. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), Sheffield, South Yorkshire, UK, 64–71. (2004)
44. Ng, A. Y., Jordan, M. I., Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. In Proceedings of the 15th Annual Conference on Neural Information Processing Systems (NIPS 2001), Vancouver, BC, Canada, 849–856. (2001)
45. Pahikkala, T., Airola, A., Naula, P., Salakoski, T.: Greedy RankRLS: A Linear Time Algorithm for Learning Sparse Ranking Models. In Proceedings of the SIGIR 2010 Workshop on Feature Generation and Selection for Information Retrieval, Geneva, Switzerland, 11–18. (2010)
46. Pahikkala, T., Tsivtsivadze, E., Airola, A., Järvinen, J., Boberg, J.: An Efficient Algorithm for Learning to Rank from Preference Graphs. Machine Learning, Vol. 75, No. 1, 129–165. (2009)
47. Pan, F., Converse, T., Ahn, D., Salvetti, F., Donato, G.: Feature Selection for Ranking Using Boosted Trees. In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, China, 2025–2028. (2009)
48. Pandey, G., Ren, Z., Wang, S., Veijalainen, J., de Rijke, M.: Linear Feature Extraction for Ranking. Information Retrieval Journal, Vol. 21, No. 6, 481–506. (2018)
49. Purpura, A., Buchner, K., Silvello, G., Susto, G. A.: Neural Feature Selection for Learning to Rank. In Proceedings of the 43rd European Conference on Information Retrieval (ECIR 2021), 342–349. (2021)
50. Qin, T., Liu, T.-Y.: Introducing LETOR 4.0 Datasets. arXiv preprint (arXiv:1306.2597) (2013). [Online]. Available: https://arxiv.org/abs/1306.2597 (current May 2021)
51. Qin, T., Liu, T.-Y., Xu, J., Li, H.: LETOR: A Benchmark Collection for Research on Learning to Rank for Information Retrieval. Information Retrieval, Vol. 13, No. 4, 346–374. (2010)
52. Qin, T., Zhang, X.-D., Tsai, M.-F., Wang, D.-S., Liu, T.-Y., Li, H.: Query-Level Loss Functions for Information Retrieval. Information Processing & Management, Vol. 44, No. 2, 838–855. (2008)
53. Rahangdale, A., Raut, S.: Deep Neural Network Regularization for Feature Selection in Learning-to-Rank. IEEE Access, Vol. 7, 53988–54006. (2019)
54. Robertson, S. E.: Overview of the Okapi Projects. Journal of Documentation, Vol. 53, No. 1, 3–7. (1997)
55. Shashua, A., Levin, A.: Ranking with Large Margin Principle: Two Approaches. In Proceedings of the 16th Annual Conference on Neural Information Processing Systems (NIPS 2002), Vancouver, BC, Canada, 961–968. (2002)
56. Shirzad, M. B., Keyvanpour, M. R.: A Feature Selection Method Based on Minimum Redundancy Maximum Relevance for Learning to Rank. In Proceedings of the 5th Conference on Artificial Intelligence and Robotics (2015 AI & Robotics), Qazvin, Iran. (2015)
57. Spearman, C.: The Proof and Measurement of Association Between Two Things. The American Journal of Psychology, Vol. 15, No. 1, 72–101. (1904)
58. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In Proceedings of the KDD 2000 Workshop on Text Mining, Boston, MA, 109–110. (2000)
59. Sun, Z., Qin, T., Tao, Q., Wang, J.: Robust Sparse Rank Learning for Non-Smooth Ranking Measures. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009), Boston, MA, 259–266. (2009)

60. Tan, P.-N., Steinbach, M., Karpatne, A., Kumar, V.: Introduction to Data Mining (2nd edition). Pearson. (2019)
61. Taylor, M., Guiver, J., Robertson, S., Minka, T.: SoftRank: Optimizing Non-Smooth Rank Metrics. In Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM 2008), Palo Alto, CA, 77–86. (2008)
62. Tsai, M.-F., Liu, T.-Y., Qin, T., Chen, H.-H., Ma, W.-Y.: FRank: A Ranking Method with Fidelity Loss. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 383–390. (2007)
63. Volkovs, M. N., Zemel, R. S.: BoltzRank: Learning to Maximize Expected Ranking Gain. In Proceedings of the 26th International Conference on Machine Learning (ICML 2009), Montreal, QC, Canada, 1089–1096. (2009)
64. von Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing, Vol. 17, No. 4, 395–416. (2007)
65. Wierzchoń, S. T., Kłopotek, M. A.: Modern Algorithms of Cluster Analysis. Springer. (2018)
66. Xia, F., Liu, T.-Y., Wang, J., Zhang, W., Li, H.: Listwise Approach to Learning to Rank - Theory and Algorithm. In Proceedings of the 25th International Conference on Machine Learning (ICML 2008), Helsinki, Finland, 1192–1199. (2008)
67. Xu, J., Li, H.: AdaRank: A Boosting Algorithm for Information Retrieval. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 391–398. (2007)
68. Xu, J., Liu, T.-Y., Lu, M., Li, H., Ma, W.-Y.: Directly Optimizing Evaluation Measures in Learning to Rank. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008), Singapore, 107–114. (2008)
69. Yeh, J.-Y., Lin, J.-Y., Ke, H.-R., Yang, W.-P.: Learning to Rank for Information Retrieval Using Genetic Programming. In Proceedings of the SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007), Amsterdam, The Netherlands, 41–48. (2007)
70. Yeh, J.-Y., Tsai, C.-J.: Graph-based Feature Selection Method for Learning to Rank. In Proceedings of the 6th International Conference on Communication and Information Processing (ICCIP 2020), Tokyo, Japan, 70–73. (2020)
71. Yu, H., Oh, J., Han, W.-S.: Efficient Feature Weighting Methods for Ranking. In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, China, 1157–1166. (2009)
72. Yue, Y., Finley, T., Radlinski, F., Joachims, T.: A Support Vector Method for Optimizing Average Precision. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), Amsterdam, The Netherlands, 271–278. (2007)
73. Zhai, C., Lafferty, J.: A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001), New Orleans, LA, 2001, 334–342. (2001)

Jen-Yuan Yeh is currently an Associate Researcher of the Dept. of Operation, Visitor Service, Collection and Information Management at the National Museum of Natural Science. His research interests include text mining and summarization, information retrieval and extraction, digital libraries and museums, and natural language processing.

Cheng-Jung Tsai is currently a professor in the Graduate Institute of Statistics and Information Science at National Changhua University of Education, Chang-Hua,

Taiwan, R.O.C. His research interests include data mining, big data analysis, information security, e-learning, and digital image processing.

Received: December 20, 2020; Accepted: July 25, 2021.