THE UNIVERSITY OF TEXAS AT SAN ANTONIO, COLLEGE OF BUSINESS

Working Paper SERIES

Date August 28, 2014

WP # 0007MSS-061-2014

Combination of Multiple Bipartite Ranking for Multipartite Web Content Quality Evaluation

Xiao-Bo Jin
School of Information Science and Engineering
Henan University of Technology
Zhengzhou, Henan, 450001, China

Guang-Gang Geng*
Computer Network Information Center
Chinese Academy of Science
Beijing, 100190, China

Minghe Sun
College of Business
The University of Texas at San Antonio
San Antonio, 78249, USA

Dexian Zhang
School of Information Science and Engineering
Henan University of Technology
Zhengzhou, Henan, 450001, China

Combination of Multiple Bipartite Ranking for Multipartite Web Content Quality Evaluation

Xiao-Bo Jin

School of Information Science and Engineering
Henan University of Technology
Zhengzhou, Henan, 450001, China

Guang-Gang Geng*

Computer Network Information Center
Chinese Academy of Science
Beijing, 100190, China

Minghe Sun

College of Business
The University of Texas at San Antonio
San Antonio, 78249, USA

Dexian Zhang

School of Information Science and Engineering
Henan University of Technology
Zhengzhou, Henan, 450001, China

Abstract

Web content quality evaluation is crucial to various web content processing applications. Bagging has a powerful classification capacity by combining multiple classifiers. In this study, similar to Bagging, multiple pairwise bipartite ranking learners are combined to solve the multipartite ranking problems for web content quality evaluation. Both encoding and decoding mechanisms are used to combine bipartite rankers to form a multipartite ranker and,

*Corresponding author. Tel: +86-010-58812272
Email addresses: xbjin9801@gmail.com (Xiao-Bo Jin), gengguanggang@cnnic.cn (Guang-Gang Geng), minghe.sun@utsa.edu (Minghe Sun), zdxzzit@hotmail.com (Dexian Zhang)

hence, the multipartite ranker is called MultiRank.ED. Both binary encoding and ternary encoding extend each rank value to an $L-1$ dimensional vector for a ranking problem with L different rank values. Predefined weighting and adaptive weighting decoding mechanisms are used to combine the ranking results of bipartite rankers to obtain the final ranking results. In addition, some theoretical analyses of the encoding and the decoding strategies in the MultiRank.ED algorithm are provided. Computational experiments using the DC2010 datasets show that the combination of binary encoding and predefined weighting decoding yields the best performance in all four combinations. Furthermore, this combination performs better than the best winning method of the DC2010 competition.

Keywords: Web Content Quality Evaluation, Multipartite Ranking, Bipartite Ranking, Encoding Design, Decoding Design
JEL Codes: C61, C63, C81, C88

1. Introduction

Website content quality has become a major research topic. The theme of the Discovery Challenge of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases[1] (DC2010) is to develop techniques to measure the quality of, or to rank, websites. As the general descriptions of DC2010 says, "In this year's Discovery Challenge we try to explore various properties that may determine the overall rank, quality and importance of a website, with the task of developing automatic methods that can be used to estimate web content quality."

Research on web content quality assessment should focus on computational models that can automatically predict the web content quality. However, in the past, most data quality measures were developed on an ad hoc basis to solve specific problems, and fundamental principles necessary for developing stable metrics in practice were insufficient [1]. Pipino et al. [1] describe principles that can help organizations develop usable metrics measuring data quality. Herrera-Viedma et al. [2] use a fuzzy linguistic approach to evaluate the quality of digital libraries. The evaluation method in Herrera-Viedma and Peis [3] generates linguistic recommendations from linguistic

[1] http://www.ecmlpkdd2010.org/indexd7fa.html?md=articles.

evaluation judgments provided by different recommenders on meaningful elements of document type definitions. The method uses two quantifier guided linguistic aggregation operators including the linguistic weighted averaging operator and the linguistic ordered weighted averaging operator. PageRank [4] is what Google uses to determine the importance of a web page and the most important pages on the Internet are the pages with the most links leading to them. Richardson [5] shows that using features that are independent of the link structure of the web such as the data on the frequency at which users visit web pages can significantly outperform PageRank. Most features used by Richardson [5] are heuristic content attributes (e.g., page and anchor text) from commercial service data. The popular features are impracticable to get for most practical applications. Geng et al. [6] explore the cross-language website ranking problem by extracting the multi-modal language-independent features and mapping these features to the eigenspace.

Web spam can significantly deteriorate the quality of the search results of the search engines. However, high quality is much more than just the identification of web spams. DC2010, for example, aims at more aspects of the websites by developing site-level classification or ranking techniques for the genre of the websites such as editorial, news, commercial, educational, deep web or web spam and more, as well as their readability, authoritativeness, trustworthiness and neutrality [7].

Each website is treated as an observation or an instance by the ranking techniques. Traditionally three learning approaches are used to rank objects. In the simplest pointwise approach, each instance is assigned a ranking score as the absolute quantity using classical regression or classification techniques [8, 9]. In the pairwise approach, the order of each pair of instances is treated as a binary variable and is learned by using a classification method (e.g., RankSVM [10], RankNet [11] and SortNet [12]). RankBoost [13], for example, maintains a number of weak ranking functions or weak rankers. Each weak ranking function orders each pair of the instances. RankBoost then combines the results of the weak ranking functions to obtain the final ranking result. Finally, the most complex listwise approaches [14, 15] try to directly optimize a ranking-specific evaluation metric, e.g., the normalized discounted cumulative gain (NDCG) [16].

In DC2010, Geng et al. [17] estimate web quality with the weighted output from bagging of C4.5 (Bagging + C4.5) [18], which can be regarded as the combination of some pointwise ranking methods. Geng et al. [17] achieved the best results among all submitted reports and won the competition at

DC2010. However, the pairwise ranking models often perform better than the pointwise methods but are simpler than the listwise methods.

Bagging [18] is a popular machine learning approach for statistical classification and regression. For classification, it ensembles many classifiers to improve the stability of the algorithm and the accuracy of the classification results. It also reduces variance and helps avoid overfitting. Furthermore, the supervised methods have demonstrated their effectiveness in many classification and regression problems such as spam detection [19, 20, 21] and anti-phishing [22]. Inspired by the idea of Bagging and the success of the supervised methods, we investigate the combination of multiple bipartite rankers to improve the performance of the ranking algorithm as a supervised approach in the evaluation of web content quality. To achieve this objective, we carefully design some encoding and decoding strategies to build multiple bipartite rankers for the evaluation of web content quality. The effectivenesses of the encoding and the decoding strategies of the algorithm are analyzed theoretically and experimentally.

In this study, we propose a new ranking algorithm, called the multipartite ranking algorithm with encoding and decoding (MultiRank.ED) for the web quality assessment by combining the results of multiple binary pairwise ranking models[2] with efficient ranking encoding and decoding mechanisms. This approach divides a multipartite ranking problem into multiple bipartite ranking problems. Specially, RankBoost, a bipartite ranking method, is used as the base learner. The number of ratings in the ranking system is represented by L. In the encoding process, binary encoding and ternary encoding are used to extend each rank value to a vector with $L-1$ dimensions. In the decoding process, predefined weighting and adaptive weighting decoding mechanisms are used to combine the multiple ranking results of multiple bipartite ranking models to obtain the final ranking result.

The dataset provided by DC2010 [7] is used to validate the developed multipartite ranking algorithm. The task of DC2010 is to rank the webpages in three different languages, i.e., English, French and Germany, according to their quality. The experimental results show that MultiRank.ED achieves better results than any other ranking methods. In particular, the combination of binary encoding and predefined weighting decoding yields the best performance in all four combinations and overpasses the best results of Bag-

[2]The binary pairwise ranking models are also called the bipartite rankers.

ging + C4.5 [18, 23] that won the DC2010 competition [17].

The reminder of this paper is organized as follows. Section 2 describes the multipartite ranking problem and the RankBoost algorithm. Section 3 presents the encoding and decoding mechanisms for multipartite ranking. Section 4 reports the experimental results. Section 5 concludes the paper and outlines future works.

2. Multipartite Ranking

A multipartite ranking algorithm minimizes the expected empirical error or maximizes the receiver operating characteristic curve (AUC) of the dataset when the ranking functions are constructed. Some multipartite ranking algorithms decompose a multipartite problem into multiple bipartite problems and use pairwise ranking functions. Specially, RankBoost minimizes the exponential loss function that is the upper bound of the empirical error under the framework of multipartite ranking.

2.1. The Multipartite Ranking Framework

The instance space is represented by $X \subset \mathbb{R}^D$, where D is the number of features. For any pair of instances $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ such that $\boldsymbol{x}_i \in X$ and $\boldsymbol{x}_j \in X$, $\boldsymbol{x}_i \succ \boldsymbol{x}_j$ indicates that $\boldsymbol{x}_i$ is preferred to $\boldsymbol{x}_j$; $\boldsymbol{x}_i \prec \boldsymbol{x}_j$ indicates that $\boldsymbol{x}_j$ is preferred to $\boldsymbol{x}_i$; while $\boldsymbol{x}_i \equiv \boldsymbol{x}_j$ indicates that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are preferred equally. The ranking algorithm should rank $\boldsymbol{x}_i$ above $\boldsymbol{x}_j$ if $\boldsymbol{x}_i \succ \boldsymbol{x}_j$; should rank $\boldsymbol{x}_j$ above $\boldsymbol{x}_i$ if $\boldsymbol{x}_i \prec \boldsymbol{x}_j$; or should rank $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ equally if $\boldsymbol{x}_i \equiv \boldsymbol{x}_j$.

The dataset of a bipartite ranking problem from the instance space is given by $S = \{S_0, S_1\}$ such that $S \subset X$. Each $\boldsymbol{x}_i^0 \in S_0$ for $1 \leq i \leq |S_0|$ has a rating 0 and each $\boldsymbol{x}_j^1 \in S_1$ for $1 \leq j \leq |S_1|$ has a rating 1. Each $\boldsymbol{x}_j^1 \in S_1$ is preferred to each $\boldsymbol{x}_i^0 \in S_0$, i.e., $\boldsymbol{x}_i^0 \prec \boldsymbol{x}_j^1$, and each $\boldsymbol{x}_j^1 \in S_1$ should be ranked above each $\boldsymbol{x}_i^0 \in S_0$. The purpose of a bipartite ranking algorithm is to develop a ranking function $f : X \to \mathbb{R}$. It is desirable to have $f(\boldsymbol{x}_i^0) < f(\boldsymbol{x}_j^1)$ for all $\boldsymbol{x}_i^0 \in S_0$ and $\boldsymbol{x}_j^1 \in S_1$. For practical problems, however, it is impossible to have $f(\boldsymbol{x}_i^0) < f(\boldsymbol{x}_j^1)$ for all $\boldsymbol{x}_i^0 \in S_0$ and $\boldsymbol{x}_j^1 \in S_1$. A ranking error occurs if $f(\boldsymbol{x}_i^0) \geq f(\boldsymbol{x}_j^1)$ for any $\boldsymbol{x}_i^0 \in S_0$ and $\boldsymbol{x}_j^1 \in S_1$. The objective of the ranking algorithm is to find a ranking function f so as to minimize the expected empirical error represented by $R_\delta(f)$ as defined in (1) in the following

$$R_\delta(f) = \frac{1}{|S_0||S_1|} \sum_{i=1}^{|S_0|} \sum_{j=1}^{|S_1|} \delta(f(\boldsymbol{x}_i^0) - f(\boldsymbol{x}_j^1)), \tag{1}$$

where $\delta(\cdot)$ is the unit step function such that $\delta(\cdot) = 1$ if the argument is nonnegative and $\delta(\cdot) = 0$ otherwise. The AUC [24] is computed as $1 - R_\delta(f)$. As the direct minimization of the expected empirical error (1) is computationally intractable, the ranking algorithm minimizes the loss function, i.e., the convex upper bound of $R_\delta(f)$ represented by $R_\phi(f)$ as defined in the following[3]

$$R_\phi(f) = \frac{1}{|S_0||S_1|} \sum_{i=1}^{|S_0|} \sum_{j=1}^{|S_1|} \phi(f(\boldsymbol{x}_i^0) - f(\boldsymbol{x}_j^1)), \tag{2}$$

where $\phi(\cdot)$ can be any convex bounded function of the unit step function $\delta(\cdot)$.

In a multipartite, more specifically an L-partite, ranking problem, the dataset $S \subset X$ is partitioned into L disjoint subsets $\{S_l\}$ for $0 \leq l \leq L-1$, where $S = \cup_{l=0}^{L-1} S_l$. The preference relation $\boldsymbol{x}_i^l \prec \boldsymbol{x}_j^k$ holds for any $\boldsymbol{x}_i^l \in S_l$ and $\boldsymbol{x}_j^k \in S_k$ such that $0 \leq l < k \leq L-1$. The purpose of a multipartite ranking algorithm is also to develop a ranking function $f : X \to \mathbb{R}$. It is desirable to have $f(\boldsymbol{x}_i^l) < f(\boldsymbol{x}_j^k)$ for any pair $\boldsymbol{x}_i^l$ and $\boldsymbol{x}_j^k$ such that $\boldsymbol{x}_i^l \in S_l$ and $\boldsymbol{x}_j^k \in S_k$ with $0 \leq l < k \leq L-1$. However, it is impossible to have $f(\boldsymbol{x}_i^l) < f(\boldsymbol{x}_j^k)$ for all $0 \leq l < k \leq L-1$ for practical problems. An ranking error occurs if $f(\boldsymbol{x}_i^l) \geq f(\boldsymbol{x}_j^k)$ for any $0 \leq l < k \leq L-1$. The ranking algorithm then determines a ranking function f so as to minimize the expected empirical error also represented by $R_\delta(f)$. For the multipartite ranking problem, $R_\delta(f)$ in (2) is extended to the C-index [25] as shown in the following

$$R_\delta(f) = \frac{1}{Z_0} \sum_{0 \leq l < k \leq L-1} \sum_{i=1}^{|S_l|} \sum_{j=1}^{|S_k|} \delta(f(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_j^k)), \tag{3}$$

where Z_0 is defined as

$$Z_0 = \sum_{0 \leq l < k \leq L-1} |S_l||S_k|. \tag{4}$$

Similarly, the loss function or the convex upper bound of $R_\delta(f)$ in (3) is obtained by extending $R_\phi(f)$ in (2) to the multipartite case as shown in the

[3]The expected empirical error also includes a regularization term in some ranking algorithms such as RankSVM [10].

following

$$R_{\phi}(f) = \frac{1}{Z_0} \sum_{0 \le l < k \le L-1} \sum_{i=1}^{|S_l|} \sum_{j=1}^{|S_k|} \phi(f(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_j^k)), \tag{5}$$

where, like in (2), $\phi(\cdot)$ can be any convex bounded function of the step function $\delta(\cdot)$ in (3).

2.2. The Evaluation Measure

NDCG is used to measure the quality of the ranking results. Different definitions of NDCG have been provided and used in the literature such as those in Järvelin and Kekäläinen [26], Valizadegan et al. [14] and Wang et al. [16]. In this study, the NDCG specified by DC2010 is used to measure the quality of the rankings. For an L-partite ranking problem, the set of the ratings is represented by $Q = \{0, 1, \cdots, L-1\}$ and the rating of a specific observation i is represented by $l_i \in Q$. Given a specific set of the rankings, denoted by ξ, for all observations in S, the discounted cumulative gain (DCG) denoted by DCG_{ξ} is defined as

$$DCG_{\xi} = \sum_{i=1}^{|S|} l_i(|S| - i), \tag{6}$$

where $(|S| - i)$ is the discount function meaning that highly relevant documents appearing lower in a search result list should be penalized. An ideal permutation π for the ratings of the observations in S is obtained by sorting the ratings in a descending order. The DCG of the ideal permutation π is denoted by DCG_{π}, hence, $DCG_{\xi} \le DCG_{\pi}$. The NDCG of the specific set of ratings ξ for all observations in S denoted by $NDCG_{\xi}$ is given by

$$NDCG_{\xi} = \frac{1}{DCG_{\pi}} DCG_{\xi}. \tag{7}$$

As Chen et al. [27] pointed out, although most ranking methods learn the ranking functions by minimizing loss functions, the performance measures such as NDCG are used to evaluate and report the performance of the ranking methods. Chen at al. [27] showed that the loss functions are upper bounds of the empirical ranking errors as measured with performance measures, e.g., $1 - NDCG$. As a result, the minimization of the loss functions is equivalent to the maximization of the performance measures, e.g., the NDCG.

2.3. RankBoost for Ranking

RankBoost [13] maintains a distribution D_t over $X \times X$ for $1 \leq t \leq T$, where T is the number of weak rankers. The distribution specifies the weight attached to each pair of instances in the input data. A pair of instances $(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k)$ is crucial if $l < k$ and the weight assigned to a crucial pair $D_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k)$ is positive under the distribution D_t. The distribution D_t is passed to the weak learners and the weak learners generate the weak ranking functions or weak rankers. RankBoost then approximates the true rankings of the instances by combining the ranking results of the T weak rankers. RankBoost minimizes the convex upper bound of the expected empirical error, also called the loss function, defined in the following

$$R_{rb}(f) = \frac{1}{Z_0} \sum_{0 \leq l < k \leq L-1} \sum_{i=1}^{|S_l|} \sum_{j=1}^{|S_k|} \exp(f(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_j^k)), \tag{8}$$

where Z_0 is defined in (4) and $1/Z_0$ may be regarded as a normalization coefficient. $R_{rb}(f)$ in (8) is a special form of $R_\phi(f)$ in (5) with $\phi(x) = \exp(x)$. In the loss function (8), $f(\boldsymbol{x})$ is the final ranking function that is a weighted sum of the weak rankers as shown in the following

$$f(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}). \tag{9}$$

In this ranking function, $h_t(\boldsymbol{x})$ represents the t-th weak ranker and α_t is the weight of $h_t(\boldsymbol{x})$ for $1 \leq t \leq T$.

A weak ranker $h_t(\boldsymbol{x})$ focuses on the binary rating producing a 0 or 1 that gives the relative ordering of $\boldsymbol{x}$ rather than a specific ranking score. A weak ranker has the following simple form

$$h_t(\boldsymbol{x}) = \begin{cases} 1, & \text{if } x_d \geq \theta_t \\ 0, & \text{if } x_d < \theta_t \\ q_t, & \text{if } x_d \text{ missing}, \end{cases} \tag{10}$$

where x_d is the value of the d-th component of $\boldsymbol{x}$, θ_t is the threshold and q_t is the default value of the weak ranker.

With the convexity of $e^{\alpha x}$, it is easily verified that $((1-u_t)e^{\alpha_t} + (1+u_t)e^{-\alpha_t})/2$ is the upper bound of Z_t [13] in Algorithm 1 where

$$u_t = \sum_{0 \leq l < k \leq L-1} \sum_{i=1}^{|S_l|} \sum_{j=1}^{|S_k|} D_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k)(h_t(\boldsymbol{x}_j^k) - h_t(\boldsymbol{x}_i^l)). \tag{11}$$

The upper bound of Z_t in each iteration is minimized when $\alpha_t = \frac{1}{2}\ln((1+u_t)/(1-u_t))$, which yields $Z_t \leq \sqrt{1-u_t^2}$. In the training process, the weak ranker scans all candidate threshold values and chooses the optimal values for d, θ_t and q_t to maximize $|u_t|$. In the extreme case, the candidate threshold values may be composed of all features from the dataset. Algorithm 1 shows the framework of RankBoost[4].

Algorithm 1 The RankBoost Algorithm

Input all pairs $\{(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k) | 0 \leq l < k \leq L-1, 1 \leq i \leq |S_l|, 1 \leq j \leq |S_k|\}$.
Initialize each $D_1(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k) = 1/Z_0$.
for $t = 1, 2, \cdots, T$ **do**
Train the weak learner $h_t(\boldsymbol{x})$ using the distribution $D_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k)$.
Choose $\alpha_t = \frac{1}{2}\ln((1+u_t)/(1-u_t))$, where u_t is defined in (11).
Update

$$D_{t+1}(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k) = \frac{1}{Z_t} D_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k) \exp(-\alpha_t(h_t(\boldsymbol{x}_j^k) - h_t(\boldsymbol{x}_i^l))),$$

where

$$Z_t = \sum_{0 \leq l < k \leq L-1} \sum_{i=1}^{|S_l|} \sum_{j=1}^{|S_k|} D_t(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k) \exp(-\alpha_t(h_t(\boldsymbol{x}_j^k) - h_t(\boldsymbol{x}_i^l))).$$

end for
Output the final ranking function $f(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$.

3. Multipartite Ranking with Encoding and Decoding

In this section, the decomposition of a multipartite ranking problem into to multiple bipartite ranking problems and the encoding and the decoding mechanisms are discussed. When encoding, each rating is encoded as a K-bit sequence[5], where each bit comes from $\{-1, 0, +1\}$. Each binary ranker may be a pointwise or a pairwise ranking function. When decoding, the final

[4]For the bipartite ranking, Freund et al. [13] gives a more efficient implementation of RankBoost called RankBoost.B.

[5]We use the term 'bit' to describe the encoding sequence although a bit may be -1.

ranking score of an instance is a weighted sum of the results of all binary rankers. The ranking scores will be sorted in a descending order to obtain the ranking results. RankBoost is used as the base ranker or base learner in MultiRank.ED. The MultiRank.ED algorithm[6] is presented in Algorithm 2.

Algorithm 2 The Multipartite Ranking Algorithm with Encoding and Decoding

Input all pairs $\{(\boldsymbol{x}_i^l, \boldsymbol{x}_j^k)|0 \leq l < k \leq L-1, 1 \leq i \leq |S_l|, 1 \leq j \leq |S_k|\}$.
Encoding: encode each rating with a K-bit code (see subsection 3.1).
for $k = 1, 2, \cdots, K$ **do**
  Train ranker $g_k(\boldsymbol{x})$ with the k-th column of the encoding matrix using all instances.
  Decoding: determine the weighting function $w_k(\boldsymbol{x})$ (see subsection 3.2).
end for
Output the final ranking function $H(\boldsymbol{x}) = \sum_{k=1}^{K} w_k(\boldsymbol{x}) g_k(\boldsymbol{x})$.

3.1. Encoding

In this subsection, the encoding mechanism used in the training algorithm is described. Given a set of L ratings to be learned for an L-partite problem, the dataset is partitioned into L subsets. A codeword is an encoding matrix $M_{L\times K}$, where $M_{lk} \in \{-1, 0, +1\}$ for $0 \leq l \leq L-1$ and $1 \leq k \leq K$. Row l of $M_{L\times K}$ corresponds to the rating l and column k corresponds to the dichotomizer g_k. A dichotomizer is like a binary classifier in classification. Informally, a bipartite ranker is introduced into a machine-learned ranking algorithm as a dichotomizer. Two encoding mechanisms, binary encoding and ternary encoding, are discussed in the following.

3.1.1. Binary Encoding

Binary encoding is used for the one-vs-all strategy [28] in multi-class classification, where each dichotomizer is built to distinguish one class from the rest. Each rating in an L-partite ranking problem is extended to a vector with $K = L - 1$ dimensions. Formally, a rating l, for $0 \leq l \leq L-1$, is

[6]Note that $H(\boldsymbol{x})$ instead of $f(\boldsymbol{x})$ is used to denote the final ranking function in Algorithm 2 to emphasize that it is a weighted sum of the binary rankers.

encoded to a vector $\boldsymbol{r}_l$ by comparing l with each $1 \leq k \leq L-1$ as defined in the following

$$M_{lk} = \begin{cases} -1, & l < k \\ +1, & l \geq k. \end{cases} \tag{12}$$

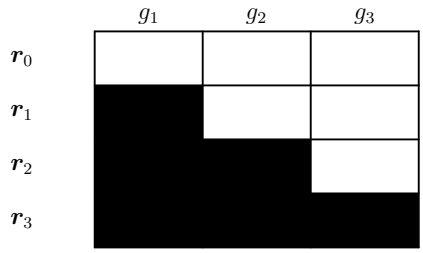

Figure 1: Binary encoding design for a problem with $L = 4$ ratings (white: -1, black: $+1$)

Figure 1 illustrates the binary encoding design with dichotomizers g_1, g_2 and g_3 for a problem with $L = 4$ ratings. The ranking algorithm sequentially executes the dichotomizers g_1, g_2 and g_3. For each dichotomizer, the rating of the instance is $+1$ or -1. More formally, each dichotomizer is a bipartite ranker. Given an observation with a rating l, the dichotomizer g_1 determines whether $l > 0$ is true. If yes, then g_2 determines whether $l > 1$ is true, and so on. The F&H method [29] uses this encoding mechanism to implement ordinal regression. Instead of the pointwise model used in the F&H method, pairwise models (RankBoost) are used in this study. In practice, pairwise models often perform better than pointwise models.

3.1.2. Ternary Encoding

Ternary encoding corresponds to the one-vs-one strategy [30] and the sparse random strategy [31] in classification. With the one-vs-one strategy, each pair of the classes is used to train a two-class model and to construct a classification function. A code of 0 in the encoding means that a particular pair of classes is not considered in a given classifier. A K-bit (K is often set to $L-1$) code is used for each pair of ratings.

Column k of the encoding matrix is used for building the dichotomizer g_k using the instances from $\{S_n\}_{n=0}^{k}$. An encoding vector $\boldsymbol{r}_l$ for the rating l is defined in a way similar to that in the binary encoding. The element M_{lk} in row l for $0 \leq l \leq L-1$ and column k for $1 \leq k \leq L-1$ is formally defined

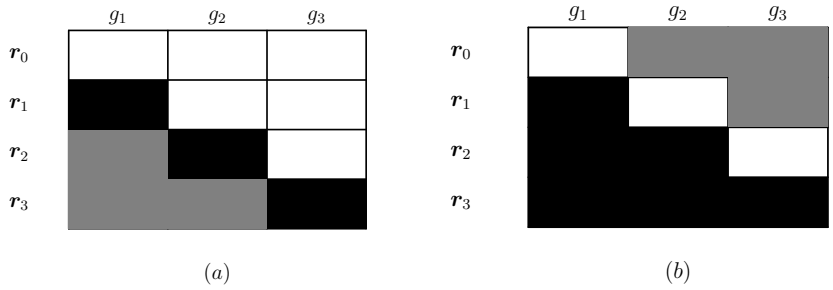

Figure 2: Ternary encoding design for $L = 4$ ratings (white: -1, black: $+1$, gray: 0); (a) Upper triangular encoding: the dichotomizer g_k is built using the instances in $\{S_n\}_{n=0}^{k}$; (b) Lower triangular encoding: the dichotomizer g_k is built using the instances in $\{S_n\}_{n=k-1}^{L-1}$.

in the following

$$M_{lk} = \begin{cases} -1, & l < k \\ +1, & l = k \\ 0, & l > k. \end{cases} \tag{13}$$

With this encoding, the upper triangular elements of the encoding matrix are all non-zero. The encoding is called the upper triangular encoding.

Alternatively, the dichotomizer g_k can also be built using all the instances from $\{S_n\}_{n=k-1}^{L-1}$. With this encoding, the lower triangular elements of the encoding matrix are all non-zero, hence, this encoding is called the lower triangular encoding. The element M_{lk} in the lower triangular encoding is defined as

$$M_{lk} = \begin{cases} 0, & l < k - 1 \\ -1, & l = k - 1 \\ +1, & l > k - 1. \end{cases} \tag{14}$$

Both of the ternary encoding mechanisms are depicted in Figure 2. In Figure 2, the matrix encodes three dichotomizers g_1, g_2 and g_3, for a 4-partite problem. The white cells represent -1, the black cells represent $+1$, and the gray cells represent 0. The gray cells are not considered by the respective dichotomizer g_k.

The Learning by Pairwise Comparison (LPC) [25] can be formulated in ternary encoding with $L(L-1)/2$ bits for each rating. As an example, Figure 3 shows an encoding with 6 bits when LPC is used to solve a 4-partite problem. Each dichotomizer $g_{l,k}$ corresponds to a pair of ratings (l, k) such that $0 \leq l < k \leq L - 1$ and is built only using the instances with ratings l

and k.

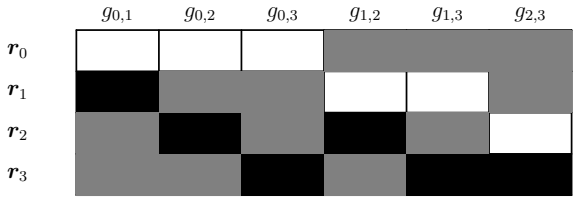

Figure 3: Ternary encoding for $L = 4$ ratings (white: -1, black: $+1$, gray: 0) in LPC: Dichotomizer $g_{l,k}$ is built using the instances in S_l and S_k.

3.2. Decoding

In classification, the most frequently used decoding mechanisms are built on certain distance metrics. Each class has a K-bit encoding vector and each bit is used for building a classifier (commonly a binary classifier). Given any input vector $\boldsymbol{x}$, a K-bit output vector $\boldsymbol{y}$ is obtained by classifying it with all K classifiers. The instance with input vector $\boldsymbol{x}$ is assigned to the class whose encoding vector is nearest to the output $\boldsymbol{y}$, according to some distance metrics. The classic decoding method for classification are Hamming decoding [32] for the binary decoding and the loss-based decoding [31] for the ternary decoding. Error-Correcting Output Codes (ECOC) [28] classifies an observation to the class whose encoding vector is nearest to the output vector of the observation in the specific distance metric. For the ranking problem, a K-bit encoding of the ratings builds K dichotomizers and the output vector with length K can be obtained for any input $\boldsymbol{x}$. However, the objective of the decoding mechanism is to fuse the outputs of multiple dichotomizers into a final ranking score instead of predicting the class membership. Hence, the decoding process is to determine the weights of the dichotomizers.

Recall that the training dataset S includes L subsets $\{S_l\}$ for $0 \leq l \leq L-1$, according to the ratings of the instances. Inspired by McRank [8] where the ranking algorithm fused the posterior probability of an instance conditioned on the class label into a score value, we define the scoring function $H(\boldsymbol{x})$ as

$$H(\boldsymbol{x}) = \sum_{k=1}^{K} w_k(\boldsymbol{x}) g_k(\boldsymbol{x}), \tag{15}$$

where $w_k(\boldsymbol{x})$ represents the weighting function of the dichotomizer $g_k(\boldsymbol{x})$ for the instance $\boldsymbol{x} \in S$. The instances are then sorted in the descending order

of $H(\boldsymbol{x})$ after they are computed for all instances. The F&H method [29] sets $w_k(\boldsymbol{x}) = 1$, independent of the index k and the instance $\boldsymbol{x}$. The computational results [8] show that obviously slightly better results are obtained with $w_k(\boldsymbol{x}) = k$ than with $w_k(\boldsymbol{x}) = 1$ if predefined weights are used. A property of (15) is that a linear transformation of the scoring function $H(\boldsymbol{x})$, i.e., multiplying it by a positive constant or adding a constant to it, will not change the ranking results.

An adaptive weighting function measuring the abilities of the dichotomizers appears to be more intuitive than the predefined weighting function. For each dichotomizer, the NDCG of the three-holdout validation result, instead of one of the three-crossfold validation results, is used as the adaptive weight, i.e., $w_k(\boldsymbol{x})$. The use of the three-fold holdout validation result is for the purpose of saving running time. In the F&H method [29], $w_k(\boldsymbol{x}) = 1$ is empirically determined. LPC [25] trains a separate ranker $g_{l,k}(\boldsymbol{x})$ on the instances with the ratings l and k for $0 \leq l < k \leq L-1$ and the prior probability $p_l p_k$ of the pair is used as the weights for the scoring function $H(\boldsymbol{x})$

$$H(\boldsymbol{x}) = \sum_{0 \leq l < k \leq L-1} p_l p_k g_{l,k}(\boldsymbol{x}), \tag{16}$$

where p_k is the probability of the k-th ranker estimated by the relative frequency in the training data.

3.3. Discussions

From (10), it can be noticed that the model $h_t(\boldsymbol{x})$ in RankBoost is in the interval $[0, 1]$. To facilitate the discussion, the output of $h_t(\boldsymbol{x})$ from the k-th dichotomizer $g_k(\boldsymbol{x})$ is represented by $h_{k,t}(\boldsymbol{x})$ for $1 \leq k \leq K$ and $1 \leq t \leq T$. The posterior probability that the rating of the instance $\boldsymbol{x}$ is k represented by $p(k|\boldsymbol{x})$ is obtained by normalizing $h_{k,t}(\boldsymbol{x})$ (also see (9))

$$p(k|\boldsymbol{x}) = \frac{\sum_{t=1}^{T} \alpha_{k,t} h_{k,t}(\boldsymbol{x})}{\sum_{t=1}^{T} \alpha_{k,t}}, \tag{17}$$

where $\alpha_{k,t}$ is a weight corresponding to the t-th weak ranker of the dichotomizer $g_k(\boldsymbol{x})$.

Because $\sum_{k=0}^{L-1} p(k|\boldsymbol{x}) = 1$ with $K = L-1$, a large value of $p(k|\boldsymbol{x})$ implies small values of $p(k'|\boldsymbol{x})$ for $k' \neq k$. In fact, the posterior probabilities $p(k'|\boldsymbol{x})$ are determined by the different dichotomizers $g_{k'}(\boldsymbol{x})$, for $1 \leq k' \leq K$, that

are independent of each other. After adding the constraints $\sum_{k=0}^{L-1} p(k|\boldsymbol{x}) = 1$ and $w_k(\boldsymbol{x}) = k$, the scoring function $H(\boldsymbol{x})$ in (15) can be rewritten as

$$H(\boldsymbol{x}) = \sum_{k=1}^{L-1} kp(k|\boldsymbol{x}) = \sum_{k=0}^{L-1} kp(k|\boldsymbol{x}). \tag{18}$$

In the decoding mechanism, the weighting function $w_k(\boldsymbol{x}) = k$ can be explained in the framework of regression analysis [33]. A loss $\Omega(H(\boldsymbol{x}), l)$ is incurred when the specific output $H(\boldsymbol{x})$ is produced from the input $\boldsymbol{x}$ with a rating l. The expected loss $\mathbb{E}(H)$ is given by

$$\mathbb{E}(H) = \int\int \Omega(H(\boldsymbol{x}), l)p(\boldsymbol{x}, l)\mathrm{d}\boldsymbol{x}\mathrm{d}l, \tag{19}$$

where $p(\boldsymbol{x}, l)$ is the joint density function on the instance $\boldsymbol{x}$ with a rating l. When the loss function is the squared loss given by $\Omega(H(\boldsymbol{x}), l) = (H(\boldsymbol{x}) - l)^2$, the expected loss can be written as

$$\mathbb{E}(H) = \int\int (H(\boldsymbol{x}) - l)^2 p(\boldsymbol{x}, l)\mathrm{d}\boldsymbol{x}\mathrm{d}l. \tag{20}$$

By calculus of variations, the following is obtained

$$H(\boldsymbol{x}) = \int lp(l|\boldsymbol{x})\mathrm{d}l = \mathbb{E}_l(l|\boldsymbol{x}), \tag{21}$$

where $\mathbb{E}_l(l|\boldsymbol{x})$ is the conditional expectation of l conditioned on $\boldsymbol{x}$. Obviously $H(\boldsymbol{x})$ in (18) is related to that in (21) in the case of the discrete variable l. It can be seen that MultiRank.ED approximates the rating by combining multiple rankers using the weighting function $w_k(\boldsymbol{x}) = k$.

As for $w_k(\boldsymbol{x}) = 1$, $p(k|\boldsymbol{x})$ in (17) is rescaled to the range $[0, L-1]$ by multiplying it by $L-1$. Hence, each dichotomizer gives a rating to the instance $\boldsymbol{x}$ and the final result $H(\boldsymbol{x})$ is the weighted average of $L-1$ ratings given by all $L-1$ dichotomizers. However, different dichotomizers have different rating capabilities and, hence, should be assigned different weights [8]. The adaptive weighting approach estimates the rating capabilities of the dichotomizers based on the NDCGs given by the dichotomizer on the hold-out dataset. Hence, only the results with $w_k(\boldsymbol{x}) = k$ are reported for the predefined weighting and the adaptive weighting approaches in the experiments of this study.

The encoding strategy tries to estimate the posterior probability $p(k|\boldsymbol{x})$. Intuitively, the more information is used from the data, the more precise the estimation is. It is likely that MultiRank.ED with the binary encoding performs better than with the ternary encodings because it builds the ranking functions to approximate the posterior probability $p(k|\boldsymbol{x})$ on all instances. The better performance is evidenced by the results of the computational experiments of this study.

4. Experiments

In the computational experiments, the performance of the proposed MultiRank.ED is compared with that of Bagging + C4.5 under different conditions for the estimation of the web content quality. Furthermore, different parameter settings in MultiRank.ED are explored to demonstrate the robustness of the ranking algorithm. As stated earlier, the DC2010 datasets are used for the computational experiment. The ranking algorithm was implemented in Java. The experiments were performed on a workstation with a 3.2GHZ Intel Xeon processor and 8GB RAM under the Window 7 operating system. RankBoost is used as the base ranker in MultiRank.ED. For all results, the number of weak learners in Bagging and RankBoost is set to $T = 100$.

4.1. Description of the Datasets

The dataset[7] of DC2010 are used for the computational experiments in this study. Three datasets for three different languages, i.e., English, French and German, are provided by DC2010. An observation is a website and the label of the observation is its rating or rank. All observations with labels in the training set of the English language are used to train the rankers. Only limited numbers of observations with labels are provided in the training sets of the French and the German languages to emphasize the cross-lingual nature of the developed methods. Hence, all the observations with labels of the English language are included in the French and the German training sets to make the training sets sufficiently large.

Because of website redirection, there exist duplicate observations with different ratings in the datasets. For such duplicate observations, only the

[7]https://dms.sztaki.hu/en/letoltes/ecmlpkdd-2010-discovery-challenge-data-set.

one with the highest rating is kept in and all others are removed from the datasets. After the duplicate observations are removed, the English, French and German training sets have 2113, 2334 and 2238 observations, respectively. As stated above, the observations in the English training set are included in the French and German training sets. The datasets have ten ratings, i.e., $L = 10$, ranging from 0 to 9. The rating of an observation is an aggregate measure of genre, trust, factuality, bias, spam, etc. The ratings of the DC2010 datasets are more delicate than the LETOR dataset which has only 3 ratings [34].

The organizers of DC2010 provided a different suite of datasets with known ratings for the observations as validation sets for the participants to optimize their ranking algorithms. They then provided another suite of datasets as the test datasets to test the ranking algorithms. The results of this suite of test sets are reported as the final results for the competition. Hence, we combine the training set of each language (English, French and German) with the validation and test sets to obtain six datasets. For notational convenience, they are denoted as $E_1, E_2, \cdots, E_6$ in Table 1.

Table 1: Description of dataset

	Training	Validation		Test	
DataSet	Size	Symbol	Size	Symbol	Size
English	2,113	E_1	131	E_2	1,314
French	2,238	E_3	138	E_4	274
German	2,334	E_5	75	E_6	234

4.2. Features

All attributes [17] as shown in Figure 4 are used as features to construct the ranking functions. Four types of multi-scale features, including content-based features, link-based features, host features and term frequency inverse document frequency (TFIDF) were extracted from the DC2010 datasets.

The content-based features and link-based features used in this study are provided by DC2010 [7]. The content features are computed from the text content of the page. These features include number of words in the home page, average word length, average length of the title, etc. The link-based features contain two types of features, i.e., link-based and transformed link-based features. The link-based features include in-degree, out-degree,

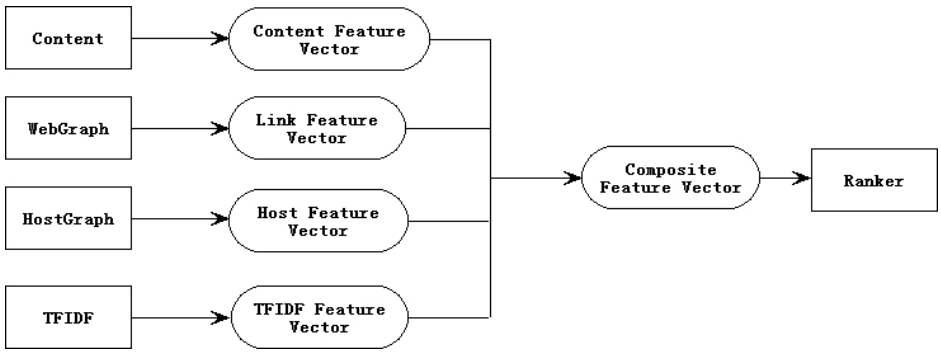

Figure 4: Flow chart of learning from multi-scale features

PageRank, edge reciprocity, assortativity coefficient, TrustRank, Truncated PageRank, estimation of supporters, etc. The transformed link-based features were found to work better for classification in practice than the raw link-based features. The transformed features include mostly ratios between features such as in-degree/PageRank or TrustRank/PageRank, and the logarithms of several features.

Link analysis algorithms usually assume that every link represents an endorsement. That is, if there is a link from page x_1 to page x_2, then x_1 is recommending x_2. It is likely that benign nodes tend to link to other high quality nodes and malicious nodes tend to link to low quality nodes. Because all hosts are connected in a graph, called the host graph, a series of host link analysis features can be extracted and used to mine the quality relations from the topology dependency. These host link analysis features may include HostRank (the PageRank value of the host), DomainPR (the rank of the domain related to host h), Truncated PageRank (the length of path is set to $1, 2, 3, 4$) and Adaptive Estimation of Supporters (the number of iterations is set to $1, 2, 3, 4$) [35]. Let $\omega(u, v)$ represent the number of hyperlinks from host u to host v and $\mathbb{M}(h)$ the value of any of the above host link analysis features, then five types of host level features can be extracted from the host graph as follows

$$F_1(h) = \mathbb{M}(h), \tag{22}$$

$$F_2(h) = \frac{\sum_{v \in \mathbb{I}(h)} \mathbb{M}(v)}{|\mathbb{I}(h)|}, \tag{23}$$

$$F_3(h) = \frac{\sum_{v\in\mathbb{O}(h)}\mathbb{M}(v)}{|\mathbb{O}(h)|}, \tag{24}$$

$$F_4(h) = \frac{\sum_{v\in\mathbb{I}(h)}\mathbb{M}(v)*\omega(v,h)}{\sum_{v\in\mathbb{I}(h)}\omega(v,h)}, \tag{25}$$

$$F_5(h) = \frac{\sum_{v\in\mathbb{O}(h)}\mathbb{M}(v)*\omega(h,v)}{\sum_{v\in\mathbb{O}(h)}\omega(h,v)}, \tag{26}$$

where $\mathbb{I}(h)$ is the set of hosts linked to host h and $\mathbb{O}(h)$ is the set of hosts linked from host h. Finally, we extracted 50 host level link features on host graph, where each type has 10 features.

Information gain (IG) [36] measures the number of bits of information obtained for the prediction of the category of a document, i.e., a webpage, by knowing the presence or absence of a word in it. IG has been proved to be one of the most effective features for text categorization, statistical spam filtering and information retrieval, and so on. After computing the document frequency of each word, the class frequency and the co-occurrence frequency of each word and each class from the webpage documents, we selected 500 dimensions of TFIDF features [17] with the top information gain values.

The content features, link features, host features and TFIDF features are summarized in Table 2.

Table 2: Description of multi-scale features

Type	Size	Description
Content	96	Average word length, average length of the title, etc.
Link	176	Link-based features, transformed link-based features
Host	50	HostRank, Truncated PageRank, etc.
TFIDF	500	Features with the top 500 IG values
Summary	882	Content, Link, Host, TFIDF

4.3. Computational Results and Discussions

For a bipartite ranking problem, the ranker should rank all the observations with positive labels above those with negative labels in the rank sequence. The rank problem will become easier and easier when the ratio of the number of the positive instances over the number of the negative ones increases. In the extreme case, any permutations will be regarded as correct when all instances have positive labels. Figure 5 shows the adaptive

weights $w_k(\boldsymbol{x})$ using the combination of the binary encoding and adaptive weighting decoding for three different languages. The NDCG measure of the dichotomizer $g_k(\boldsymbol{x})$ on the holdout set[8] is used as the adaptive weight $w_k(\boldsymbol{x})$. For the binary encoding, the weight $w_k(\boldsymbol{x})$ decreases when the index k increases.

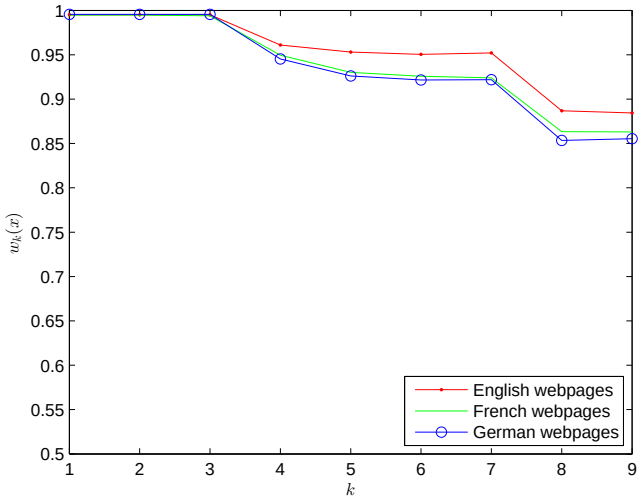

Figure 5: In decoding with the adaptive weighting, the weights $w_k(\boldsymbol{x})$ change as the index k of the dichotomizers $g_k(\boldsymbol{x})$ increases for the ranking tasks of the three languages

Figure 6 gives a comparison between the NDCGs of the predefined and the adaptive weighting decoding mechanisms using the binary encoding mechanism. The NDCGs of the adaptive weighting mechanism are consistently lower than those of the predefined weighting mechanism for all sets. These facts are evidences that the predefined weighting mechanism outperforms the adaptive weighting mechanism.

Figure 7 gives the comparisons of the NDCGs among three encoding mechanisms under two decoding mechanisms. For both of the predefined weighting and the adaptive weighting mechanisms, it is clear that the binary encoding outperforms the ternary encoding. Moreover, the NDCGs of the lower triangular ternary encoding and the upper triangular ternary encoding

[8]We holdout 2/3 of the instances in the training set and build the ranking model, then tested the model on the rest of the instances to obtain a NDCG measure.

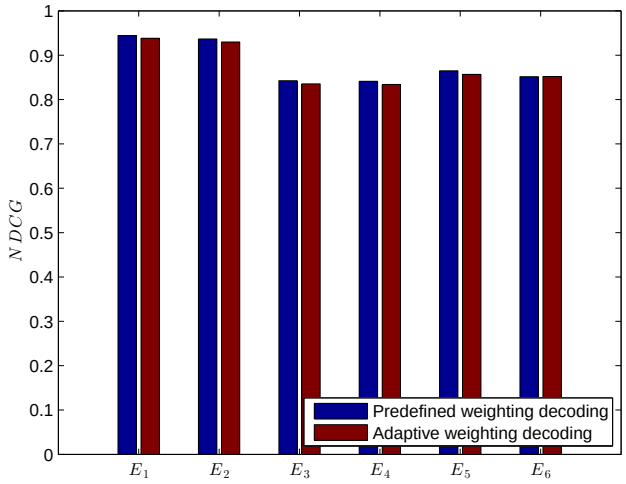

Figure 6: Comparisons of the NDCGs between two decoding mechanisms using the binary encoding mechanism

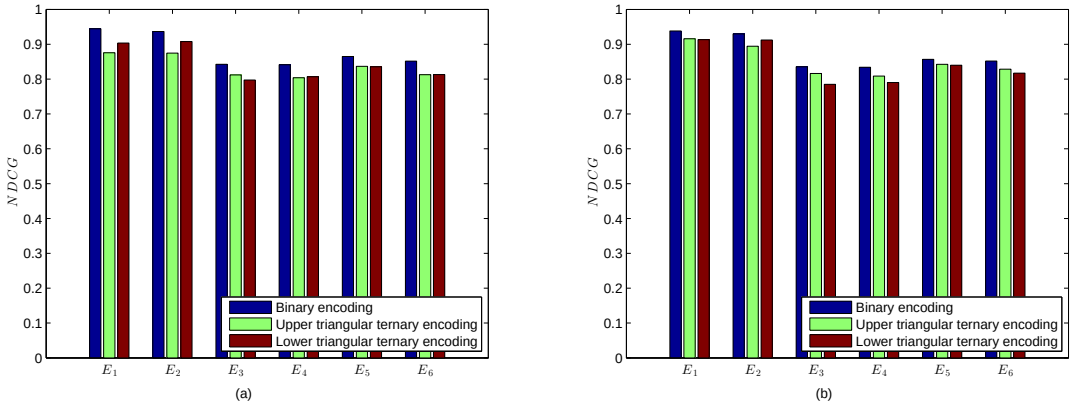

Figure 7: Comparisons of NDCGs among three encoding methods with the two decoding mechanisms (a) predefined weighting decoding, (b) adaptive weighting decoding

mechanisms are also compared in Figure 7. It is interesting to see that the lower triangular ternary encoding is more effective than the upper triangular ternary encoding for the predefined weighting decoding mechanism and the opposite is true for the adaptive weighting decoding mechanism.

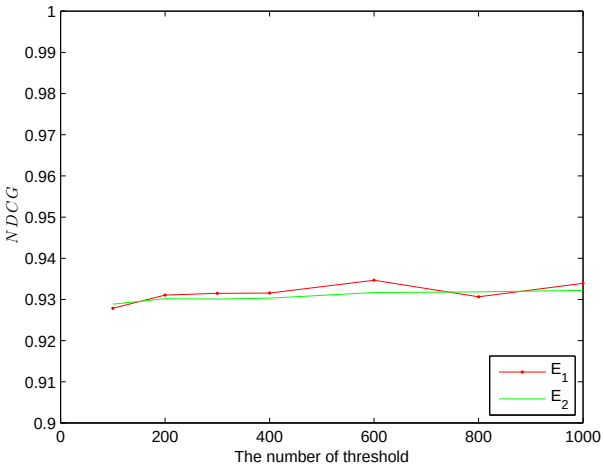

Figure 8: Changes in the NDCGs with the number of threshold values

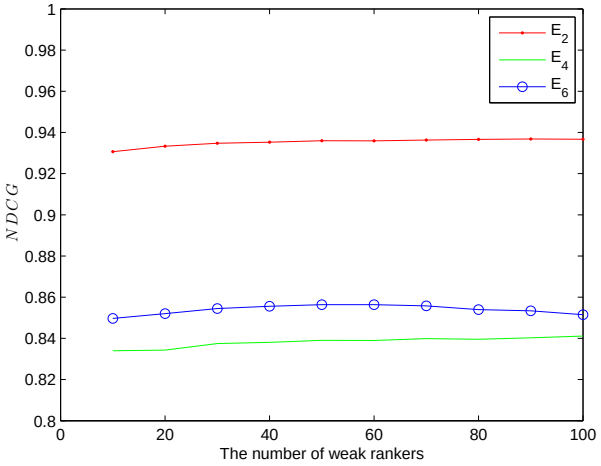

Figure 9: Performance of MultiRank.ED with the binary encoding and the predefined weighting decoding mechanisms as the number of weak rankers increases

Figure 8 shows that the NDCG increases slightly when more threshold values are given. This fact shows that the algorithm more likely finds an optimal threshold value if more discrete threshold values are provided.

AdaBoost [37, 38], a classification algorithm, usually does not overfit the tranining data even when the number of weak classifiers becomes large. RankBoost can be regarded as the application of AdaBoost to the ranking problem. Figure 9 shows that the NDCG varies gently and MultiRank.ED resists overfitting as the number of weak rankers increases. The result is consistent with that reported in Freund et al. [13].

Table 3 compares the results of Bagging + C4.5 and MultiRank.ED. A number in parentheses of a column heading represents the number of threshold values θ_t used by the weak ranker for each attributes[9]. The third column without a number in the column heading indicates that a weak ranker selects a candidate threshold value from all attribute values of the training set. For all results, the number of weak learners is set to $T = 100$. For these results, the combination of binary encoding and predefined weighting decoding mechanisms is used in MultiRank.ED. As the results in the first two columns show, MultiRank.ED performed better than Bagging + C4.5 for all test sets except for E_2. Bagging + C4.5 obtained the best results among all submitted reports to DC2010. The last two columns present the results of MultiRank.ED with different numbers of threshold values.

Table 3: NDCGs of Bagging + C4.5 and MultiRank.ED with the binary encoding and the predefined weighting decoding mechanisms

dataset	Bagging+C4.5	MultiRank.ED	MultiRank.ED (100)	MultiRank .ED (1000)
E_1	0.9325	**0.9442**	0.9279	0.9339
E_2	**0.9378**	0.9367	0.9289	0.9322
E_3	0.8359	0.8425	**0.8430**	0.8397
E_4	0.8405	0.8411	**0.8424**	0.8402
E_5	0.8620	0.8649	0.8610	**0.8657**
E_6	0.8484	**0.8515**	0.8515	0.8502

[9]We use the same number of thresholds for each attribute although the thresholds of different attributes are possibly set to different values.

5. Conclusion

In this study, a ranking algorithm, called MultiRank.ED, is developed for the website content quality evaluation problem using multiple bipartite pairwise ranking models together with efficient encoding and decoding mechanisms. Both binary encoding and ternary encoding mechanisms are presented. For a ranking problem with L ratings, each rating is encoded into an $L-1$ dimensional vector. Both predefined weighting and adaptive weighting mechanisms are used for decoding. The DC2010 datasets containing web pages in English, French and German languages are used to validate and test the proposed ranking algorithm. Factors affecting the performance of the ranking algorithm measured with NDCG, including the number of weak rankers, the number of the threshold values and the different encoding and decoding mechanisms, are experimentally tested through computation using the DC2010 datasets. The computational results show that MultiRank.ED using the combination of binary encoding and predefined weighting decoding mechanisms outperforms Bagging + C4.5, the winning method of the DC2010 competition.

The ways of effectively combining multiple ranking sequences may be explored as a future work. Exploring other weighting strategies for MultiRank.ED is another interesting research direction.

Acknowledgments

This work was partially supported by the Innovation Funds of Henan University of Technology (11CXRC09) and the National Natural Science Foundation of China (61103138, 61005029 and 61375039).

References

[1] L. L. Pipino, Y. W. Lee, R. Y. Wang, Data quality assessment, Communications of the ACM 45 (4) (2002) 211–218.

[2] E. Herrera-Viedma, E. Peis, J. M. Morales-del Castillo, S. Alonso, K. Anaya, A fuzzy linguistic model to evaluate the quality of web sites that store XML documents, International Journal of Approximate Reasoning 46 (1) (2007) 226–253.

[3] E. Herrera-Viedma, E. Peis, Evaluating the informative quality of documents in SGML format from judgements by means of fuzzy linguistic techniques based on computing with words, Information Processing & Management 39 (2) (2003) 233–249.

[4] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web, Technical report.

[5] M. Richardson, A. Prakash, E. Brill, Beyond PageRank: Machine learning for static ranking, in: Proceedings of the 15th International Conference on World Wide Web, 2006, p. 707–715.

[6] G.-G. Geng, L.-M. Wang, W. Wang, A.-L. Hu, S. Shen, Statistical cross-language web content quality assessment, Knowledge-Based Systems 35 (2012) 312–319.

[7] A. A. Benczur, C. Castillo, M. Erdelyi, Z. Gyongyi, J. Masanes, M. Matthews, ECML/PKDD 2010 discovery challenge data set, in: Crawled by the European Archive Foundation, 2010.

[8] P. Li, C. Burges, Q. Wu, Mcrank: Learning to rank using multiple classification and gradient boosting, in: NIPS, 2007, pp. 897–904.

[9] K. Crammer, Y. Singer, Pranking with ranking, in: Advances in Neural Information Processing Systems 14, 2001, p. 641–647.

[10] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 133–142.

[11] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to rank using gradient descent, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, p. 89–96.

[12] L. Rigutini, T. Papini, M. Maggini, F. Scarselli, SortNet: Learning to rank by a neural preference function, IEEE Transactions on Neural Networks 22 (9) (2011) 1368–1380.

[13] Y. Freund, R. Iyer, R. E. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, Journal of Machine Learning Research 4 (2003) 933–969.

[14] H. Valizadegan, R. Jin, R. Zhang, J. Mao, Learning to rank by optimizing NDCG measure, in: Advances in Neural Information Processing Systems 22, 2009, pp. 1883–1891.

[15] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, H. Li, Learning to rank: From pairwise approach to listwise approach, in: Proceedings of the 24th International Conference on Machine Learning, 2007, p. 129–136.

[16] Y. Wang, L. Wang, Y. Li, D. He, T.-Y. Liu, A theoretical analysis of NDCG type ranking measures 30 (2013) 25–54.

[17] G.-G. Geng, X.-B. Jin, X.-C. Zhang, D.-X. Zhang, Evaluating web content quality via multi-scale features, in: ECML/PKDD 2010 Workshop on Discovery Challenge 2010, 2010.

[18] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

[19] C. Castillo, D. Donato, A. Gionis, V. Murdock, F. Silvestri, Know your neighbors: Web spam detection using the web topology, in: SIGIR 2007, 2007, pp. 423–430.

[20] G.-G. Geng, Q. Li, X. Zhang, Link based small sample learning for web spam detection, in: Proceedings of the 18th International Conference on World Wide Web, 2009, p. 1185–1186.

[21] C.-C. Lai, An empirical study of three machine learning methods for spam filtering, Knowledge-Based Systems 20 (3) (2007) 249–254.

[22] S. Abu-Nimeh, D. Nappa, X. Wang, S. Nair, A comparison of machine learning techniques for phishing detection, in: Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit, 2007, p. 60–69.

[23] J. R. Quinlan, C4.5: Programs for machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[24] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

[25] J. Fürnkranz, E. Hüllermeier, S. Vanderlooy, Binary decomposition methods for multipartite ranking, ECML PKDD '09, 2009, p. 359–374.

[26] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (4) (2002) 422–446.

[27] W. Chen, T.-y. Liu, Y. Lan, Z. Ma, H. Li, Ranking measures and loss functions in learning to rank, in: NIPS 2009, 2009.

[28] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (1995) 263–286.

[29] E. Frank, M. Hall, A simple approach to ordinal classification, in: Proceedings of the 12th European Conference on Machine Learning, 2001, p. 145–156.

[30] T. Hastie, R. Tibshirani, Classification by pairwise coupling, The Annals of Statistics 26 (2) (1998) 451–471.

[31] E. L. Allwein, R. E. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, in: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers, 2000, p. 9–16.

[32] N. J. Nilsson, Learning Machines, McGraw-Hill, 1965.

[33] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[34] T. Qin, T.-Y. Liu, J. Xu, H. Li, LETOR: A benchmark collection for research on learning to rank for information retrieval, in: Information Retrieval Journal, 2010.

[35] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, R. Baeza-Yates, Using rank propagation and probabilistic counting for link-based spam detection, in: Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD), 2006.

[36] Y. Yang, J. O. Pedersen, A comparative study on feature selection in text categorization, in: ICML, 1997, p. 412–420.

[37] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.

[38] L. Reyzin, R. E. Schapire, How boosting the margin can also boost classifier complexity, in: ICML'06, 2006, p. 753–760.