

## 查询依赖的有序多超平面排序学习模型\*

孙鹤立<sup>1+</sup>, 黄健斌<sup>2</sup>, 冯博琴<sup>1</sup>, 赵志勤<sup>1</sup>, 刘均<sup>1</sup>, 郑庆华<sup>1</sup>

<sup>1</sup>(西安交通大学 计算机科学与技术系, 陕西 西安 710049)

<sup>2</sup>(西安电子科技大学 软件学院, 陕西 西安 710071)

### Query Dependent Learning to Model Based on Ordered Multiple Hyperplanes

SUN He-Li<sup>1+</sup>, HUANG Jian-Bin<sup>2</sup>, FENG Bo-Qin<sup>1</sup>, ZHAO Zhi-Qin<sup>1</sup>, LIU Jun<sup>1</sup>, ZHENG Qing-Hua<sup>1</sup>

<sup>1</sup>(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

<sup>2</sup>(School of Software, Xidian University, Xi'an 710071, China)

+ Corresponding author: E-mail: helisun.sunny@gmail.com

Sun HL, Huang JB, Feng BQ, Zhao ZQ, Liu J, Zheng QH. Query dependent learning to model based on ordered multiple hyperplanes. *Journal of Software*, 2011, 22(11): 2773-2781. <http://www.jos.org.cn/1000-9825/3908.htm>

**Abstract:** This paper proposes a ranking model that trains different hyperplanes for different queries and optimizes hyperplanes with the order relations. It aims at solving the problem of most existing rank methods that do not consider the significant differences between queries and only resort to a single function that is time consuming. Next, a weighted voting method is proposed to aggregate the ranking lists of the hyperplanes as the final rank. The weights reflect the degree of precision. Effectiveness is tested by the benchmark data set LETOR OHSUMED and is compared with other ranking models. The proposed method shows improved ranking performance with a significant reduction of training time.

**Key words:** learning to rank; query dependent; multiple hyperplane; order relation; aggregation; weighted voting

**摘要:** 针对当前基于支持向量机的排序学习方法训练时间长以及不考虑查询之间差异、模型单一的问题,提出一种查询依赖的有序多超平面排序学习模型.根据不同查询,利用其对应训练数据所属等级之间的序关系构建多个超平面.此外,提出了一种加权表决方法对多个超平面的排序列表进行聚合,根据各超平面的排序精度赋予其不同权重,计算最终排序结果.在标准数据集 LETOR OHSUMED 上对所提出的模型性能进行了综合评测,并与相关排序模型进行了对比分析.实验结果显示,所提出的模型排序性能有较大提升.同时,训练时间明显缩短.

**关键词:** 排序学习;查询依赖;多超平面;序关系;聚合;加权表决

中图法分类号: TP391 文献标识码: A

排序函数是决定信息检索系统查询结果相关度的主要指标.如何为用户呈现与其查询最相关、最权威的信息,对提高信息检索的性能影响巨大.因此,查询结果排序方法的研究至关重要.信息检索中的排序函数通常利用相关度和权威度等指标,计算给定实例的得分,然后根据其得分多少进行排序.概率模型、BM25<sup>[1]</sup>、语言模

\* 基金项目: 国家高技术研究发展计划(863)(2008AA01Z131); 陕西省自然科学基金研究计划(SJ08-ZT14)

收稿时间: 2009-10-10; 定稿时间: 2010-07-06

型<sup>[2]</sup>等传统无监督排序方法通常利用一些简单的特征进行排序,例如词频(tf)、倒排文档频率(idf)等.这些方法的优点在于容易进行经验参数调整.但是,随着待排实例的结构、链接等大量复杂特征的出现,传统的无监督排序方法变得力不从心:首先,手工的参数调整非常困难,当参数很多并且评估指标非平滑时,该问题尤其严重;其次,手工的参数调整通常导致过拟合(overfitting);最后,难以整合其他新的排序算法.

最近,使用机器学习方法从训练数据中自动创建排序模型方法的出现,使得上述问题的解决成为可能:机器学习方法能够整合大量复杂特征并自动进行参数调整,还能通过众多有效手段规避过拟合.通常,将使用机器学习技术解决排序问题的方法称作排序学习(learning to rank).该方法所使用的待排实例以特征的形式表示,这些特征可以是已有排序方法的输出结果.例如,可以使用 BM25 值作为待排实例的一个特征.排序学习方法能够学习得到所有特征的最优组合,这使得该方法能够吸纳排序领域的任何新进展,只需将新方法的输出作为其所使用的一个特征即可.在真实的网络环境中,搜索引擎每天都会收到很多用户的反馈信息和使用日志,因此,自动地利用反馈持续改善排序机制至关重要.排序学习方法所采用的判别训练能够利用这些反馈信息进行学习,持续改善排序模型,并允许反馈数据具有多个相关性层次.这恰好满足了真实搜索引擎环境的需要.

排序学习方法的重要性,使其在信息检索及相关领域得到了广泛的关注.目前,针对信息检索提出的一些排序学习模型包括:基于支持向量机(SVM)的排序模型构建方法,例如 RankSVM<sup>[3]</sup>,RSVM-IR<sup>[4]</sup>,MHR<sup>[5]</sup>等;使用相对熵作为损失函数,并利用梯度下降算法来训练神经网络排序模型,例如 RankNet<sup>[6]</sup>,ListNet<sup>[7]</sup>等;以及使用提升策略(boosting)进行排序函数构建的方法,例如 RankBoost<sup>[8]</sup>,AdaRank<sup>[9]</sup>等.此外,损失函数的优化<sup>[10]</sup>、关系数据排序<sup>[11]</sup>以及排序指标的直接优化<sup>[12]</sup>也是当前的研究热点.目前,排序模型创建中存在的主要问题在于模型的复杂度较高,训练时间长,对不同查询的文档排序缺乏针对性.特别是 RankSVM 方法,虽然有较好的排序效果,但模型训练时间较长,这对实际应用较为不利.此外,当前的排序方法大多采用单一模型处理所有查询,这显然不适合真实 Web 信息检索环境的需要.Web 信息检索中,不同用户提交的查询往往具有不同语义,反映了不同的用户兴趣,并通常使用不同的表现形式,而它们的相关文档也有所不同.查询可能是人名、产品名或术语.查询的形式可能是一个短语,也可能是几个短语或用自然语言表示的句子.查询可长可短,其主题可能很热门也可能很罕见.使用单一模型处理所有查询的相关文档势必会导致排序结果缺乏针对性,降低排序准确率.据此,本文提出一种查询依赖的有序多超平面排序模型(query dependent ordered multiple hyperplanes,简称 QDOMH),利用排序等级间的序关系,针对不同查询构建不同排序模型.此外,针对训练所得的多个超平面,本文提出一种加权表决(voting)方法对排序结果进行聚合.实验结果显示,与传统排序模型 BM25 相比,QDOMH 极大地改进了排序准确率;与其他基于支持向量机的排序学习方法相比,QDOMH 的模型训练时间显著降低,并取得了更好的排序性能;与传统表决方法相比,加权表决针对各超平面的排序精度进行权重分配更具优势.

本文第 1 节介绍排序学习方法并分析当前基于支持向量机的方法所存在的问题.第 2 节中给出查询依赖的有序多超平面排序模型构建方法.第 3 节中给出加权表决方法的详细过程.实验方法及实验数据在第 4 节中给出.第 5 节总结全文.

## 1 排序学习

信息检索中的排序模型通常是一个定义在查询和文档上的关于特征的函数.在文档检索领域,排序学习方法使用的数据主要包括查询、查询对应的文档集合、文档与该查询的相关度或权威度等级.在进行模型训练之前,将给定数据划分为 3 个部分:一些数据用做模型训练集,主要用于排序模型训练;一些数据用做参数校验集,用于模型的参数调整;再选取一些数据作为模型测试集,用于模型性能的评估.

### 1.1 排序学习过程

**定义 1(等级).** 设  $q$  表示某一用户查询, $D=\{d_1,d_2,\dots,d_n\}$  表示文档集合且文档数为  $n$ , $Y=\{y_1,y_2,\dots,y_t\}$  表示相关度等级集合且等级个数为  $t$ . $\succ$  为集合  $D$  上的线序关系, $d_i\succ d_j$  表示“ $d_i$  比  $d_j$  与查询  $q$  更相关”.设  $\succ'$  为集合  $Y$  上的线序关系, $f_q:D\rightarrow Y$  为文档等级标注函数, $f_q(d)$  表示文档与查询  $q$  的相关性程度.若有  $f_q(d_i)=r_i$  且  $f_q(d_j)=r_j$ ,则  $d_i\succ d_j$  且仅当  $r_i\succ' r_j$ .

当排序学习用于文档检索时,其主要任务就是为每一个文档  $d$  求解其等级标注  $y$ ,其流程如图 1 所示.训练阶段,给定查询集合  $Q=\{q_1,q_2,\dots,q_m\}$ 、每个查询的检索文档集合  $d_i = \{d_1^{(i)},d_2^{(i)},\dots,d_{n_i}^{(i)}\}$  以及这些文档的等级标注集合  $y_i = \{y_1^{(i)},y_2^{(i)},\dots,y_{n_i}^{(i)}\}$ ,其中,  $n_i$  为与查询  $q_i$  对应的文档数.因为每个文档针对一个查询  $q_i$  只有一个等级标注,因此,集合  $y_i$  的大小也是  $n_i$ .为每个查询-文档对构建一组特征矢量  $\bar{x}_{ij} \in X$ ,其中,  $i=1,2,\dots,m, j=1,2,\dots,n_i$ .这样,训练数据集可以表示为  $S = \{q_i, d_i, y_i\}_{i=1}^m$ .利用训练集  $S$  进行排序模型  $f: X \rightarrow R$  构建,然后计算排序模型  $f$  在校验集上的性能,如果  $f$  不满足损失函数的值最小,则对  $f$  进行参数调整;否则,将  $f$  作为最终的模型输出.测试阶段,利用获得的排序模型  $f$  对文档的等级进行标注,从而实现与查询  $q_i$  的相关度排序.

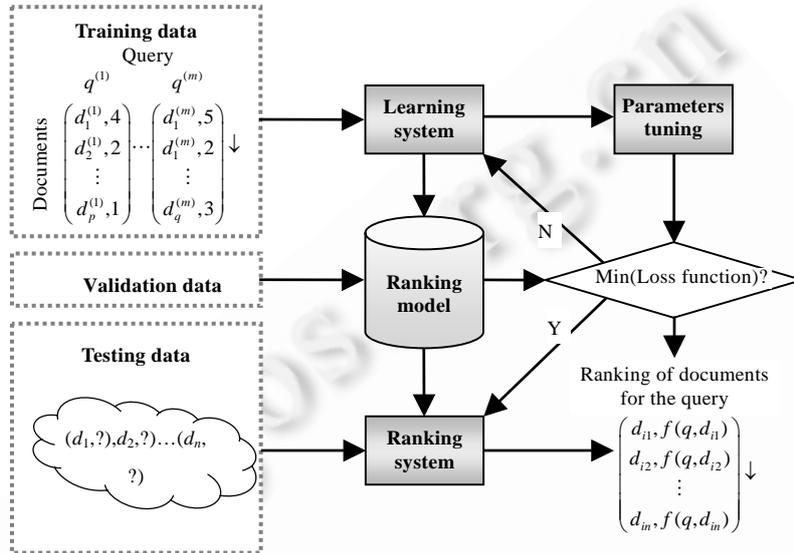


Fig.1 Learning to rank

图 1 排序学习

1.2 RankSVM原理与分析

RankSVM 是当前一种经典的排序学习模型.它利用支持向量机技术进行排序模型的构建,其工作原理简述如下:

在 RankSVM 中,排序函数  $f$  为如下形式的线性函数:

$$f(x)=\langle \omega,x \rangle \tag{1}$$

其中,  $\omega$  为权重向量,  $\langle \cdot, \cdot \rangle$  表示内积,  $f(x)=0$  即为输出空间中的一个超平面.根据定义 1 和公式(1),可得到等价关系:

$$x_i > x_j \Leftrightarrow \langle \omega, x_i - x_j \rangle > 0 \tag{2}$$

这样,利用实例对  $x_i, x_j$  可以创建一个新的实例  $x_i - x_j$ .如果  $x_i$  排在  $x_j$  之前,则将新实例标记为 +1;否则标记为 -1.按照这种方法得到一个新的数据集并利用它构建一个二值的分类模型,将排序问题转化为分类问题加以解决.构建排序模型等价于解决下面的二次优化问题:

$$\min_{\omega, \xi_{i,j}} \frac{1}{2} \|\omega\|^2 + C \sum \xi_{ij} \quad \text{s.t.} \quad \langle \omega, x_i - x_j \rangle > 1 - \xi_{ij}, \quad \forall x_i > x_j, \quad \xi_{ij} \geq 0 \tag{3}$$

其中,  $\|\omega\|^2$  为  $\ell_2$  范式,用于度量超平面的间隔(margin);  $\xi_{ij}$  为松散变量,用于线性不可分时使上式的解存在.设公式(3)得到的解为  $\omega^*$ ,则最终的排序函数可以表示为

$$f(x)=\langle \omega^*,x \rangle \tag{4}$$

RankSVM 模型能够利用复杂特征进行排序,取得了良好的排序效果.但其也有自身的局限性:其一,模型在处理现实中的复杂排序问题时显得过于简单;其二,模型的训练时间长,当训练数据集变大时,该问题愈加严重.

这主要是由于模型所使用的训练数据是实例对的形式,而实例对数几乎是数据集中实例数的平方;其三,模型利用一个排序超平面对所有的查询进行相关性排序,缺乏针对性而影响了排序性能.

为了更好地分析 RankSVM 方法,对标准数据集 LETOR OHSUMED 中的数据进行了主成分分析(principal component analysis,简称 PCA).如图 2 所示,其中,横坐标为实例的第一主元,纵坐标为实例的第二主元.LETOR OHSUMED 数据集是文档检索领域的标准数据集,其中的数据为查询-文档对的形式,且每个查询-文档对有一个相应的等级标签:irrelevant(记为  $R_1$ )、partially relevant(记为  $R_2$ )或 definitely relevant(记为  $R_3$ ).图 2(a)为对 Query 11 的主成分分析图,RankSVM 的解决方法是,构建一个超平面,将所有查询的不同等级实例分开.由于  $R_1 \sim R_3$  的实例对数较多,最终的超平面方向趋向于将  $R_1 \sim R_3$  分开的方向.图 2(b)是 Query 10 和 Query 11 的主成分分析图,由于 Query 11 的相关文档更多,则最终的超平面趋向于将 Query 11 的对应文档分开的方向.上述问题在训练数据的样本分布与真实数据的样本分布差距较大时愈加严重,将直接影响排序精度.

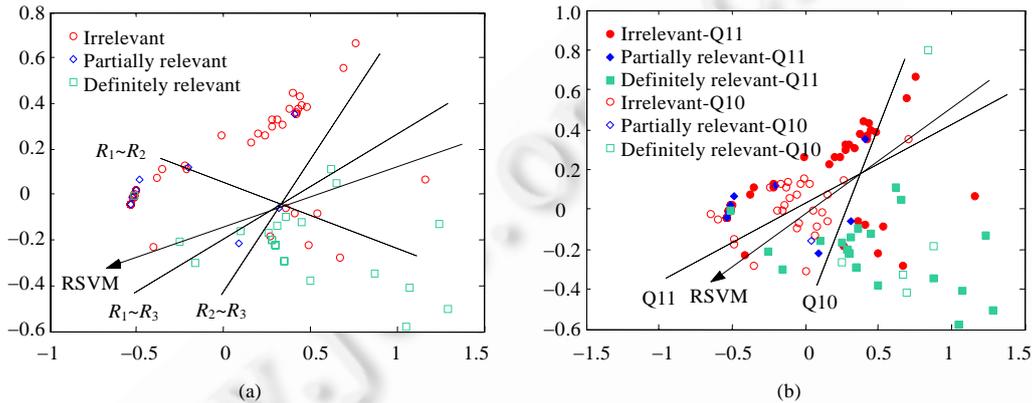


Fig.2 Instances distribution of Query 10 and Query 11 in LETOR OHSUMED dataset

图 2 LETOR OHSUMED 数据集中 Query 10 和 Query 11 的实例分布情况

## 2 查询依赖的有序多超平面排序模型

针对 RankSVM 模型存在的问题,本文提出了一种查询依赖的有序多超平面排序学习模型(query dependent ordered multiple hyperplanes,简称 QDOMH).QDOMH 方法包括两个步骤:首先进行模型构建,利用训练数据进行多超平面模型的训练和参数调整,得到排序模型;然后对多个超平面所产生的排序列表进行聚合,得到最终的排序结果.本节与第 3 节将分别对这两个步骤进行阐述.

### 2.1 查询依赖的多超平面构建

设训练数据集中的查询数为  $Q$ .首先根据查询将训练数据集分成  $Q$  个不同的子数据集,每个数据集中包含一个查询及其对应的所有文档,以及查询和文档的相关度等级.然后利用子数据集为每个查询构建一个超平面模型,这等同于求解下面的优化问题:

$$\min_{\omega, \xi_{q,i,j}} \frac{1}{2} \|\omega\|^2 + C \sum_{i,j} \xi_{q,i,j} \quad \text{s.t.} \quad \langle \omega, x_i - x_j \rangle \geq 1 - \xi_{q,i,j}, \quad \xi_{q,i,j} \geq 0 \quad (5)$$

其中,  $q$  为查询,  $\omega$  为超平面的参数矢量,  $x_i$  与  $x_j$  为训练实例,  $\xi_{q,i,j}$  为松散变量.

这样,为不同查询构建不同超平面,且单个超平面的训练样例数大幅度降低,所有超平面使用的实例对总数也有很大程度的缩减,这将直接缩短模型训练时间.

### 2.2 利用序关系进行模型优化

如上所述,排序学习方法中的训练实例与查询的相关性程度或权威程度由领域专家事先判定.例如,LETOR OHSUMED 数据集中的实例与查询的相关度分为 3 个等级:不相关(irrelevant)、部分相关(partially relevant)、

相关(definitely relevant).以往的排序学习方法往往简单地把它们看作 3 个类别,忽略了它们之间的序关系. QDOMH 方法利用等级之间的序关系进行超平面构建,主要利用序关系中的邻近关系,定义如下:

**定义 2(邻接关系).** 设  $Y=\{y_1,y_2,\dots,y_t\}$  表示相关度等级集合且等级个数为  $t$ ,其中, $y_m \in Y$ ,简记为  $m \triangleright$  是集合  $Y$  上的线序关系.若等级  $m,n \in Y$ ,且满足  $m \triangleright n \wedge \exists i(i \in Y \wedge m \triangleright i \triangleright n)$ ,则称等级  $n$  与等级  $m$  相互邻接.

对于查询  $q$ ,为其对应的任意两个满足邻接关系的等级对应的子数据集构建一个排序超平面.当等级  $m \triangleright n$  时,与等级  $m$  和  $n$  对应的超平面可以表示如下:

$$\min_{\omega_{m,n}, \xi_{q,i,j,m,n}} \frac{1}{2} \|\omega_{m,n}\|^2 + C \sum_{i,j} \xi_{q,i,j,m,n} \quad \text{s.t.} \quad \langle \omega_{m,n}, x_i - x_j \rangle \geq 1 - \xi_{q,i,j,m,n}, \xi_{q,i,j,m,n} \geq 0 \quad (6)$$

其中, $k$  为等级数, $\omega_{m,n}$  为等级  $m$  与等级  $n$  的超平面的参数矢量, $\xi_{q,i,j,m,n}$  为松散变量, $x_i$  表示等级  $m$  对应子数据集中的一个训练实例, $x_j$  表示等级  $n$  对应子数据集中的训练实例.

仍然利用 LETOR OHSUMED 数据集的 Query 11 举例说明 QDOMH 方法的超平面构建方式,如图 3 所示.根据图 2 中的符号表示和定义 2 得到: $R_2 \triangleright R_1, R_3 \triangleright R_2$ .利用等级  $R_1$  和  $R_2$  对应的子数据集训练得到超平面  $\omega'_{1,2}$ ,利用等级  $R_2$  和  $R_3$  对应的子数据集训练得到超平面  $\omega'_{2,3}$ .针对邻接等级创建不同的超平面,能够在较大程度上解决 RankSVM 中的训练数据分布不均所导致的超平面方向偏离问题.

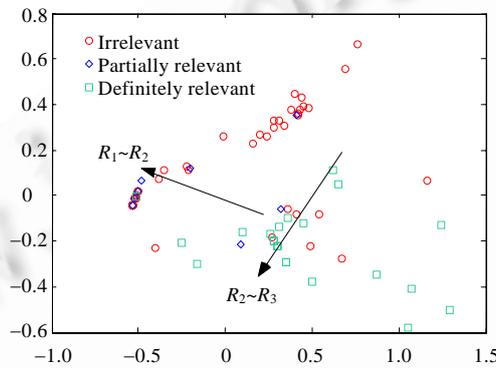


Fig.3 QDOMH hyperplane construction for Query 11 of LETOR OHSUMED dataset

图 3 LETOR OHSUMED 数据集中 Query 11 的 QDOMH 超平面构建

### 2.3 模型训练算法

模型构建是 QDOMH 方法的核心部分,其具体的算法描述如下:

**算法 1.** 查询依赖的有序多超平面排序模型构建.

输入:训练数据集  $T$ ,查询数  $Q$ ,等级数  $k$ .

输出:排序超平面.

- (1) 针对不同的查询将数据集  $T$  划分为  $Q$  个子数据集;
- (2) For  $i=1,2,\dots,Q$  do
- (3) 对查询对应实例的等级进行判定:如果  $\text{rank } j > \text{rank } l$ ,则根据公式(6),利用标记为等级  $j$  的实例和标记为  $l$  的所有实例训练排序超平面  $M_{ijl}$ ;
- (4) 利用校验集进行参数调整;
- (5) 输出排序超平面  $M_{ijl}$ ;
- (6) End For

### 3 加权表决排序聚合

利用已构建的多个超平面获得多个排序列表后,需要将这些列表聚合起来以获得最终的排序结果.设  $D$  表示实例集合, $n$  为  $D$  中的实例数.设  $\tau=\{\tau_1,\tau_2,\dots,\tau_p\}$  为利用多个超平面所获得的排序列表集合,其中, $p$  为超平面数.

这里提出一种加权表决方法对排序列表进行聚合, 设  $T = \{(x_i, y_i) | i=1, 2, \dots, N_T\}$  表示训练集, 其中,  $x_i$  为训练实例,  $y_i$  为  $x_i$  的等级标签,  $N_T$  为集合  $T$  中的训练实例数.  $T_k$  为  $T$  的一个子集, 设  $f_k(k=1, 2, \dots, K)$  为利用  $T_k$  训练得到的第  $k$  个排序超平面. 设  $E = \{e_j | j=1, 2, \dots, N_e\}$  表示测试集, 其中,  $e_j$  为测试实例,  $N_e$  为测试集中的实例数. 测试中, 利用训练得到的所有超平面  $f_k(k=1, 2, \dots, K)$  对测试实例进行评分. 设  $f_k(e_j)$  表示第  $k$  个排序超平面对实例  $e_j$  的评分, 该评分反映了超平面  $k$  对实例  $e_j$  与对应查询的相关程度或权威程度的判定.

对于已构建的多个超平面, 使用下面的权重设置方法为不同超平面赋予不同权重, 该权重值由超平面的排序准确率决定. 首先, 利用  $f_k(k=1, 2, \dots, K)$  对训练集  $T$  中实例进行评分, 其评分矩阵  $S$  可以表示为

$$S = \begin{bmatrix} f_1(x_1) & f_1(x_2) & \cdots & f_1(x_{N_T}) \\ f_2(x_1) & f_2(x_2) & \cdots & f_2(x_{N_T}) \\ \vdots & \vdots & \ddots & \vdots \\ f_K(x_1) & f_K(x_2) & \cdots & f_K(x_{N_T}) \end{bmatrix} \quad (7)$$

设集合  $T$  中实例的标签矢量为  $Y = (y_1, y_2, \dots, y_{N_T})$ , 则可以得到权重向量  $w$  如下:

$$w = (S^{-1}Y)^T \quad (8)$$

测试过程中, 在利用超平面获得实例的评分后, 任意实例的排序值通过一个表决过程决定. 设  $v_{ij}^k$  为第  $k$  个超平面对实例  $e_i$  和  $e_j$  的表决, 则  $v_{ij}^k$  定义如下:

$$v_{ij}^k = \begin{cases} 1, & \text{if } f_k(e_i) > f_k(e_j) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

其中,  $f_k(e_i)$  表示第  $k$  个超平面对实例  $e_i$  的评分. 由  $v_{ij}^k$  组成了超平面  $k$  的  $N_e \times N_e$  表决矩阵  $V^k$ , 则  $V^k$  中的第  $i$  行元素的和为排序值低于  $e_i$  的实例的个数.

$$v_i^k = \sum_{j=1}^{N_e} v_{ij}^k \quad (10)$$

于是, 可将  $v_i^k(k=1, 2, \dots, K)$  与其对应的权重系数乘积的平均值看作是对实例  $e_i$  的表决值  $v(e_i)$ , 然后根据  $v(e_i)$  值的大小对所有实例进行排序, 得到最终排序结果:

$$v(e_i) = \frac{1}{K} w \cdot (v_i^k)_{K \times 1} \quad (11)$$

**算法 2.** 加权表决排序聚合.

输入: 训练数据集  $T = \{(x_i, y_i) | i=1, 2, \dots, N_T\}$ , 训练集  $T$  中实例的标签集合  $Y = (y_1, y_2, \dots, y_{N_T})$ , 查询数  $Q$ , 测试数据集  $E = \{e_j | j=1, 2, \dots, N_e\}$ , 测试集中的实例数  $N_e$ , 排序超平面集合  $f_k(k=1, 2, \dots, K)$ .

输出: 排序列表.

- (1) 利用  $f_k(k=1, 2, \dots, K)$  获得对训练集  $T$  的评分矩阵  $S$ ;
- (2) 计算权重向量  $w = (S^{-1}Y)^T$ ;
- (3) If 在  $f_k$  的排序列表中测试实例  $e_i$  的位置高于  $e_j$  Then  $v_{ij}^k = 1$  Else  $v_{ij}^k = 0$ ;
- (4) For  $k=1, 2, \dots, K$  do
- (5) For  $j=1, 2, \dots, N_e$  do
- (6)  $v_i^k = v_i^k + v_{ij}^k$ ;
- (7) End For
- (8) End For
- (9) 根据权重向量  $w$  与  $v_i^k(k=1, 2, \dots, K)$  乘积的平均值大小对测试实例排序, 得到最终排序列表并输出.

#### 4 实验评价

这里主要将 QDOMH 方法与 BM25, RankSVM 和 MHR 方法进行了比较. 其中, BM25 方法为经典的无监督

排序方法,RankSVM 与 MHR 方法分别为经典的和当前精度最高的基于支持向量机的排序学习方法.

#### 4.1 评价标准与数据集

本文采用信息检索中通常使用的 3 个指标——Precision at  $n$ ( $P@n$ ), Mean Average Precision(MAP)<sup>[2]</sup>, Normalized Discounted Cumulative Gain(NDCG)<sup>[13]</sup>对提出的排序模型进行实验评价. $P@n$  用于衡量一个查询的前  $n$  个排序结果的准确率,如公式(12)所示.例如,当某个查询的前 6 文档的相关度分别为{相关,相关,不相关,不相关,相关,相关}时,其  $P@1, P@2, \dots, P@6$  值分别为{1/1, 2/2, 2/3, 2/4, 3/5, 4/6}.

$$P@n = \frac{\text{number of relevant instances in top } n}{n} \quad (12)$$

给定一个查询  $q$ , 其 Average Precision(AP)被定义为

$$AP = \sum_{n=1}^N \frac{P@n \times pos(n)}{\text{number of relevant instances}} \quad (13)$$

其中,  $N$  为检索到的文档数;  $pos(n)$  是一个二值函数, 当位置  $n$  处的文档为相关的其值为 1, 否则为 0.  $MAP$  为一个查询集合中所有查询  $AP$  的平均值.

给定查询  $q$ , 其位置  $n$  处的  $NDCG$  值被定义为

$$NDCG@n = Z_n \sum_{j=1}^n (2^{R(j)} - 1) / \log(1 + j) \quad (14)$$

其中,  $R(j)$  为第  $j$  个文档的等级;  $Z_n$  为规范化常量, 用于保证当结果为最佳排序时其  $NDCG@n$  值为 1.

这里使用排序学习领域的标准数据集 LETOR OHSUMED<sup>[14]</sup>进行实验测试, 其中包含 106 个查询, 348 566 个文档, 16 140 个带有相关度的查询-文档对, 其相关度共有 3 种: 相关、部分相关、不相关, 由领域专家事先给定. 数据集中的文档由一组特征向量表示, 其中包括内容、链接和混合这 3 种类型的特征. 内容特征又分为低层特征和高层特征, 其中, 低层内容特征包括词频(tf)、倒排文档频率(idf)、文档长度(dl), 以及它们的组合(tf\*idf)等, 高层内容特征由 BM25 和 LMIR 等算法计算得到. 链接特征包括 PageRank, HITS 以及它们的改进算法. 最后, 由内容和链接信息共同得到两个混合特征.

#### 4.2 实验结果

实验在主频为 3.0GHz 的 P4 CPU, 1GB RAM 内存, 以及 Window XP 系统的 PC 机上完成. 使用的支持向量机工具是 LibSVM<sup>[15]</sup>, 实验中采用交叉验证(cross validation)的方式进行模型训练与测试, 将 LETOR OHSUMED 数据集划分为 5 个子集, 每次使用其中的 3 个子集进行模型训练, 一个子集进行模型参数调整, 另一个作为测试集. 总共进行 5 次训练与测试, 将所有测试得到实验数据的平均值作为模型的性能指标.

表 1 中给出了 BM25, RankSVM, MHR 和 QDOMH 几种方法在 LETOR OHSUMED 数据集上的  $P@n$  值, 其中,  $n=1, 2, \dots, 10$ .  $P@n$  值反映了每种方法前 10 位排序结果的精度. 此外, 为了检验文中提出的加权表决方法的先进性, 实验中根据聚合方法的不同, 将 QDOMH 分为利用传统表决方法进行排序聚合的 QDOMH-V 方法和使用加权表决进行聚合的 QDOMH-WV 方法进行实验测试, 并分别给出实验结果.

由图 4 可以看出, 所有 3 种基于支持向量机的排序学习方法的  $MAP$  值比 BM25 方法均有较大提高, 这也证明了排序学习方法利用更多复杂特征提升排序准确率的能力. 在基于支持向量机的所有方法中, QDOMH-WV 的效果最佳, 其  $MAP$  值比 RankSVM 方法提升了将近 3%, 比 MHR 方法提高了将近 2%; QDOMH-V 方法的  $MAP$  也比 RankSVM 和 MHR 方法有所提升, 这证实了 QDOMH 方法在改善平均排序准确率方面的有效性. 根据  $NDCG$  值所遵循的原则——高度相关的文档比一般相关的文档价值更高; 排序值越低的文档, 被用户查看的概率越小, 其对用户的价值也越低, 这里给出了每种方法前 10 位的  $NDCG$  值. 从图中可以看出, QDOMH-WV 方法在  $NDCG@1$  上比其他方法有明显优势; 在其他位置上, 该方法的  $NDCG$  值均超过了其他几种方法, 比 BM25 方法高出了 6% 以上, 比 RankSVM 方法高出了 3%, 比 MHR 整体上也有一定程度的提高. QDOMH-V 方法也在绝大部分位置上与其他方法持平或有提高. 这也反映了 QDOMH 方法在提升前几位排序结果准确率方面的有效性.

表 2 中给出了几种方法的模型训练时间. 由于空间所限, QDOMH 方法中仅给出所有超平面的总训练时间,

各超平面的训练时间未逐一列出.从表中可以看出,QDOMH 方法的模型训练时间仅为 MHR 方法的 1/12,为 RankSVM 方法的 1/30.QDOMH 方法极大地降低了模型训练时间,更具实际应用价值.

**Table 1**  $P@n$  value of BM25, RankSVM, MHR and QDOMH on the LETOR OHSUMED dataset

表 1 BM25 与 RankSVM, MHR 和 QDOMH 方法在 LETOR OHSUMED 数据集上的  $P@n$  值

	$P@1$	$P@2$	$P@3$	$P@4$	$P@5$	$P@6$	$P@7$	$P@8$	$P@9$	$P@10$
BM25	0.453	0.442	0.438	0.411	0.391	0.372	0.367	0.352	0.331	0.310
RankSVM	0.613	0.572	0.536	0.498	0.487	0.461	0.458	0.445	0.440	0.426
MHR	0.644	0.600	0.574	0.560	0.539	0.533	0.530	0.522	0.501	0.482
QDOMH-V	0.651	0.644	0.630	0.601	0.580	0.572	0.557	0.533	0.517	0.486
QDOMH-WV	0.667	0.653	0.645	0.615	0.600	0.585	0.571	0.564	0.543	0.512

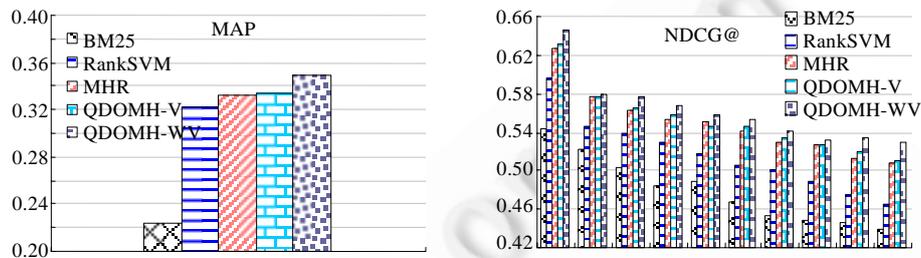


Fig.4 MAP and NDCG values of BM25, RankSVM, MHR and QDOMH on the LETOR OHSUMED dataset

图 4 BM25 与 RankSVM, MHR 和 QDOMH 方法在 LETOR OHSUMED 数据集上的 MAP 与 NDCG 值

**Table 2** Training time of model on the LETOR OHSUMED dataset

表 2 在 LETOR OHSUMED 数据集上的模型训练时间

Minutes	RankSVM	MHR				QDOMH
		$\omega_{1,2}$	$\omega_{2,3}$	$\omega_{1,3}$	Sum	
Trial 1	1 728.64	306.16	51.64	228.43	586.23	53.330 3
Trial 2	2 013.57	456.41	38.72	283.57	778.7	62.838
Trial 3	1 845.36	329.31	35.44	260.85	625.60	46.711 5
Trial 4	906.53	157.74	17.23	139.27	314.24	29.544 3
Trial 5	1 113.72	202.01	14.63	163.51	380.15	38.690 3
Average	1 521.56	290.33	31.53	215.13	536.98	46.222 88

## 5 结束语

本文提出了一种查询依赖的有序多超平面排序学习模型,利用等级间的序关系使模型创建更具针对性.另外,本文提出一种依据排序精度赋权的加权表决方法.实验结果显示:与传统排序模型 BM25 相比,排序学习方法具有明显优势;与其他基于支持向量机的排序学习方法相比,QDOMH 方法的模型训练时间显著降低,并取得了更好的排序性能,这使得该模型具有良好的应用前景;与传统表决方法相比,加权表决方法获得了更佳的聚合效果.如何分析和利用待排实例间的关系,进一步提高排序学习效率及准确率,是一个值得后续深入研究的课题.

## References:

- [1] Robertson SE, Hull DA. The TREC-9 filtering track final report. In: Voorhees EM, Harman DK, eds. Proc. of the 9th Text Retrieval Conf. National Institute of Standards and Technology (NIST). 2000. 25–40.
- [2] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval. Addison Wesley, 1999.
- [3] Joachims T. Optimizing search engines using clickthrough data. In: Zaki MJ, Wang JTL, Toivonen H, eds. Proc. of the 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2002. 133–142. [doi: 10.1145/775047.775067]
- [4] Cao YB, Xu J, Liu TY, Li H, Huang YL, Hon HW. Adapting ranking SVM to document retrieval. In: Efthimiadis EN, Dumais ST, Hawking D, Järvelin K, eds. Proc. of the 29th Annual Int'l ACM SIGIR Conf. on Research and Development in Information

- Retrieval. New York: ACM Press, 2006. 186–193. [doi: 10.1145/1148170.1148205]
- [5] Qin T, Zhang XD, Wang DS, Liu TY, Lai W, Li H. Ranking with multiple hyperplanes. In: Kraaij W, de Vries AP, Clarke CLA, Fuhr N, Kando N, eds. Proc. of the 30th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2007. 279–286. [doi: 10.1145/1277741.1277791]
- [6] Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G. Learning to rank using gradient descent. In: Raedt L, Wrobel S, eds. Proc. of the 22nd Annual Int'l Conf. on Machine Learning. New York: ACM Press, 2005. 89–96. [doi: 10.1145/1102351.1102363]
- [7] Cao Z, Qin T, Liu TY, Tsai MF, Li H. Learning to rank: from pairwise approach to listwise approach. In: Ghahramani Z, ed. Proc. of the 24th Annual Int'l Conf. on Machine Learning. New York: ACM Press, 2007. 129–136. [doi: 10.1145/1273496.1273513]
- [8] Freund Y, Iyer RD, Schapire RE, Singer Y. An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research, 2003,4(11):933–969.
- [9] Xu J, Li H. AdaRank: A boosting algorithm for information retrieval. In: Kraaij W, de Vries AP, Clarke CLA, Fuhr N, Kando N, eds. Proc. of the 30th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2007. 391–398. [doi: 10.1145/1277741.1277809]
- [10] Burges CJC, Ragno R, Le QV. Learning to rank with nonsmooth cost functions. In: Platt JC, Koller D, Singer Y, Roweis ST, eds. Proc. of the 21 Annual Conf. on Neural Information Processing Systems. Cambridge: MIT Press, 2007. 193–200.
- [11] Qin T, Liu TY, Zhang XD, Wang DS, Xiong WY, Li H. Learning to rank relational objects and its application to Web search. In: Huai JP, Chen R, Hon HW, Liu YH, Ma WY, Tomkins A, Zhang XD, eds. Proc. of the 17th Int'l Conf. on World Wide Web. New York: ACM Press, 2008. 407–416. [doi: 10.1145/1367497.1367553]
- [12] Yue YS, Finley T, Radlinski F, Joachims T. A support vector method for optimizing average precision. In: Kraaij W, de Vries AP, Clarke CLA, Fuhr N, Kando N, eds. Proc. of the 30th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2007. 271–278. [doi: 10.1145/1277741.1277790]
- [13] Jarvelin K, Kekalainen J. IR evaluation methods for retrieving highly relevant documents. In: Belkin NJ, Ingwersen P, Leong M, eds. Proc. of the 23rd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2000. 41–48. [doi: 10.1145/345508.345545]
- [14] Liu TY, Xu J, Qin T, Xiong WY, Li H. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In: Joachims T, Li H, Liu TY, Zhai CX, eds. Proc. of the SIGIR 2007 Int'l Workshop on Learning to Rank for Information Retrieval. New York: ACM Press, 2007.
- [15] Chang CC, Lin CJ. LIBSVM: A library for support vector machines. 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>



孙鹤立(1983—),女,辽宁铁岭人,博士,讲师,CCF 会员,主要研究领域为信息检索,机器学习.



赵志勤(1982—),男,助理工程师,主要研究领域为信息检索.



黄健斌(1975—),男,博士,副教授,CCF 会员,主要研究领域为数据挖掘与知识发现,网络信息检索.



刘均(1973—),男,博士,教授,主要研究领域为数据/文本挖掘,CSCW.



冯博琴(1942—),男,教授,博士生导师,主要研究领域为智能网络,信息检索.



郑庆华(1969—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为智能网络学习理论,计算机网络应用技术.