基于查询聚类的排序学习算法*

花贵春 张 敏 刘奕群 马少平 茹立云

(智能技术与系统国家重点实验室 北京 100084)
(清华信息科学与技术国家实验室(筹) 北京 100084)
(清华大学 计算机科学与技术系 北京 100084)

摘 要 排序学习算法作为信息检索与机器学习的一个交叉领域,越来越受到人们的重视.然而,几乎没有排序学习算法考虑到查询差异的存在.文中查询被建模为多元高斯分布,KL 距离被用来度量查询之间的距离,利用谱聚类方法对查询进行聚类,为每个聚类类别训练一个排序函数.实验结果表明经过聚类得到的排序函数需要较少的训练样例,但是它的性能却和没有经过聚类得到的排序函数具有可比性,甚至优于后者.

关键词 排序学习, 排序函数, 谱聚类
中图法分类号 TP 391.3

Learning to Rank Based on Query Clustering

HUA Gui-Chun, ZHANG Min, LIU Yi-Qun, MA Shao-Ping, RU Li-Yun
(*State Key Laboratory of Intelligent Technology and Systems*, *Beijing* 100084)
(*Tsinghua National Laboratory for Information Science and Technology*, *Beijing* 100084)
(*Department of Computer Science and Technology*, *Tsinghua University*, *Beijing* 100084)

ABSTRACT

Learning to rank, the interdisciplinary field of information retrieval and machine learning, draws increasing attention and lots of models are designed to optimize the ranking functions. However, few methods take the differences among the queries into account. In this paper, the queries are modeled as multivariate Gaussian distributions and Kullback-Leibler divergence is adopted as distance measure. The spectral clustering is applied to cluster the queries into several clusters and a ranking function is learned for each cluster. The experimental results show that the ranking functions with clustering are trained with less data, but are comparable to or even outperform the ones without clustering.

Key Words Learning to Rank, Ranking Function, Spectral Clustering

* 国家自然科学基金(No. 60736044, 60903107, 61073071)、高等学校博士学科点专项科研基金(No. 20090002120005)资助项目
收稿日期:2010 - 01 - 14;修回日期:2010 - 08 - 23
作者简介 **花贵春**,男,1983 年生,博士研究生,主要研究方向为信息检索、机器学习. E-mail:huaguichun@ gmail. com. **张敏**,女,1977 年生,博士,副教授,主要研究方向为机器学习、信息检索. **刘奕群**,男,1981 年生,博士,讲师,主要研究方向为信息检索. **马少平**,男,1961 年生,教授,博士生导师,主要研究方向为知识工程、信息检索、汉字识别与后处理、中文古籍数字化. **茹立云**,男,1979 年生,博士研究生,主要研究方向为信息检索.

1 引　言

当用户需要从大规模数据集或互联网上查找某一信息的时候,往往需要求助于信息检索技术. 排序是信息检索技术中重要的环节之一. 目前,提出的用于构建排序函数的特征已有数百种,而且这些特征本身就可看作是一个排序函数,例如基于内容的特征,如 TFIDF、BM25;基于链接的特征,如 PageRank、HITS;基于点击数据的用户行为的特征. 传统的利用人工调整参数构建排序函数的方法,已经不适用于特征数为几百的情况. 因此,排序学习方法,作为信息检索与机器学习的一个交叉学科,越来越受到人们的重视. 排序学习方法一般是在优化一个损失函数的基础上,将多种特征通过融合的方式构建成一个排序函数,如线性组合. 现在排序学习主要的研究方向可归结为 3 种[1]:提出一种排序学习算法用于构建更有效率或更有效果的排序函数;构建一种优化函数,能够指导排序学习方法学习得到一个更优的排序函数;设计一种特征能够更好地描述某一数据集.

然而,现有的排序学习算法往往忽视一个问题:在许多方面,查询彼此之间差异很大. 例如,当用户使用搜索引擎的时候,根据用户查询意图的不同,查询可被划分为导航类查询,信息类查询和事务类查询;根据相关资源的多少,查询可分为热门类查询和长尾查询. 所以对所有的查询都采用一个统一的排序函数的方法是不合适的,例如,基于链接的特征 PageRank 对重要的和流行的查询非常有用,如查询'H1N1',它的相关页面非常的多. 然而,对于不是重要的和流行的查询来说,PageRank 的排序效果却不是很好,如,'SVMMAP',它的相关页面较少. 相反,基于内容的特征 BM25 对长尾查询却很有效.

针对这类问题,我们的解决办法是先利用多元高斯分布对查询进行建模,然后对查询进行聚类. 整个流程可分为线下和线上两部分. 线下工作:完成对查询进行聚类,然后利用排序学习算法为每个聚类类别学得一个排序函数;线上工作:先为新到来的查询分类,然后将这个类对应的排序函数应用到这个查询上. 实验结果显示,经过聚类后得到的排序函数需要较少的训练样例,而排序效果与未经聚类得到的排序函数具有可比性,甚至超过后者.

本文提出为不同类别的查询构建相应的排序函数的框架,利用谱聚类对经过建模后的查询进行聚类,为每一类别学得一个排序函数;对新来的查询进行分类,将这一类别对应的排序函数应用于新来的查询. 考虑到用户的体验,我们设计线下工作和线上工作两部分的工作模式,使得用户在使用搜索引擎的时候,响应时间在用户可接受的范围内.

2 相关工作介绍

有两方面的研究和本文的工作相关:查询分类和排序学习.

已有大量研究从不同的角度对查询进行分类. 文献[2]、[3]通过分析搜索引擎的用户行为,分别独立提出用户的查询意图可分为信息类的、导航类的和事务类的. 文献[4]根据查询对应的地理分布对查询进行分类,文献[5]根据查询对应主题的不同对查询进行分类,文献[6]、[7]根据用户意图的不同自动或手动地对查询进行分类,文献[8]、[9]和[10]的研究结果表明确定查询的类别可提高排序函数的性能. 但是这些分类方法都需要额外的数据,如 URL、点击数据等,因此考虑如何对一个新来的查询进行分类以及用户的响应时间,使得这些分类方法在实际的网络搜索引擎中不具有可应用性.

排序学习算法需要的数据是由 3 部分构成:查询,查询对应的文档以及人工标注的查询与文档之间的相关度. 现在提出的排序学习算法可根据训练样例的不同分为 3 类:Pointwise,Pairwise 和Listwise. Pointwise 方法,如 Pranking with Ranking[11],查询与一个文档之间的相关分数,是通过查询与这个文档对应的各个特征值计算后得到的. Pairwise 方法,如 Ranking SVM[12-13],把查询对应的文档对作为训练样例. Listwise 方法,如 SVMMAP[12,14],把查询对应的文档序列作为训练样例. 但是现在提出的排序学习算法,几乎都没有考虑查询之间的差异,对所有的查询都采用一个统一的排序函数,这种方法不适用于纷繁复杂的网络搜索环境.

在实际的网络搜索环境中存在着大量的查询,很难估计查询可分为多少类别以及多少查询属于同一类别. 这是我们设计这个模型的出发点,从而利用这种方法提高排序函数的性能.

3 系统结构及模型

首先介绍整个系统的结构,然后给出系统中用到的各个模型:查询表示以及查询距离的度量方法,对查询进行聚类的谱聚类算法,对每一类别进行排序学习的 Ranking SVM 算法,以及对新查询进行分类的查询分类算法. 分类算法,即比较新的查询与各个类别

类中心的距离,将最小距离对应的类别作为这个查询的类别. 由于篇幅限制,分类算法不做特殊说明.

3.1 系统结构

考虑到用户体验,系统可以划分为线上和线下两个工作模块,具体设计如图 1 所示.

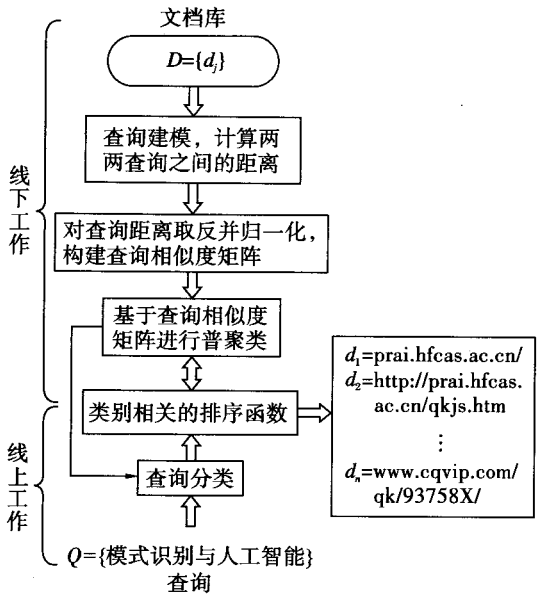


图 1 系统结构

Fig. 1 System structure

线下工作:首先利用多元高斯分布模型对查询进行建模,然后把多元高斯分布之间的 KL 距离作为两个查询之间的距离,从而得到所有查询上两两之间的距离. 将所有查询上两两之间的距离取反,并把归一化后得到的数值,作为两个查询之间的相似度,从而得到所有查询对应的相似度矩阵. 基于相似度矩阵,利用谱聚类算法对查询进行聚类,可以将查询聚为若干类别. 针对每个类别,利用 Ranking SVM 算法训练得到排序函数.

线上工作:对于一个新来的查询,利用分类算法将此查询分类为线下工作中得到的某一聚类类别,然后将这一类别对应的排序函数应用到这个查询对应的文档,最终得到排序结果.

这样当用户输入一个查询的时候,响应时间能够在一个可接受的范围内.

3.2 查询建模及查询之间的距离

表 1 中给出模型中用到的符号的定义以及说明. 存在一个查询集合 Q,它的第 i 个查询 q_i 对应着文档集合 $\boldsymbol{d}_i = \{d_i^1, d_i^2, \cdots, d_i^{n(q_i)}\}$,$n(q_i)$ 是 $\boldsymbol{d}_i$ 中文档的数目. 特征序列 $\boldsymbol{X}_i^j = [X_{i,1}^j, X_{i,2}^j, \cdots, X_{i,m(d_i^j)}^j]$ 对应着 $\boldsymbol{d}_i$ 中的第 j 个文档 d_i^j,其中 $m(d_i^j)$ 是 d_i^j 对应的特征的数目,$X_{i,k}^j$ 是 d_i^j 的第 k 个特征的值. 定义

$$\boldsymbol{X}_{i,k} = [X_{i,k}^1, X_{i,k}^2, \cdots, X_{i,k}^{n(q_i)}]^{\mathrm{T}},$$

从而得到矩阵

$$\boldsymbol{q}_i = [\boldsymbol{X}_{i,1}, \boldsymbol{X}_{i,2}, \cdots, \boldsymbol{X}_{i,m(d_i^j)}]$$

与一个查询 q_i 对应.

表 1 模型中应用到的符号的定义以及说明

Table 1 Notations and explanations used in model

定义	说明
$q_i \in Q$	第 i 个查询
$\boldsymbol{d}_i = \{d_i^1, d_i^2, \cdots, d_i^{n(q_i)}\}$	第 i 个查询 q_i 对应的文档序列
$\boldsymbol{X}_i^j = [X_{i,1}^j, X_{i,2}^j, \cdots, X_{i,m(d_i^j)}^j]$	查询 q_i 和文档 d_i^j 对应的特征向量
$\boldsymbol{X}_{i,k} = [X_{i,k}^1, X_{i,k}^2, \cdots, X_{i,k}^{n(q_i)}]^{\mathrm{T}}$	第 i 个查询 q_i 对应的文档序列中,第 k 个特征对应的列向量
$\boldsymbol{q}_i = [\boldsymbol{X}_{i,1}, \boldsymbol{X}_{i,2}, \cdots, \boldsymbol{X}_{i,m(d_i^j)}]$	第 i 个查询 q_i 对应的矩阵
$N(\mu, \Sigma)$	多元高斯分布
f	概率密度函数
KL	Kullback-Leibler 距离

如把 $\boldsymbol{X}_i^j$ 作为多元高斯分布的一个采样,则可得到对应关系:Φ: $q_i \leftrightarrow N(\mu, \Sigma)$.

概率密度函数为

$$f_i(\boldsymbol{X}_i^j) = \frac{1}{(2\pi)^{\frac{n_j}{2}} \cdot |\boldsymbol{\Sigma}^i|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{X}_i^j - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{X}_i^j - \boldsymbol{\mu}_i)\right).$$

假设存在两个查询:

$$\boldsymbol{q}_1 \leftrightarrow N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1),\ \boldsymbol{q}_2 \leftrightarrow N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2),$$

我们定义两个查询的差别 Difference$(\boldsymbol{q}_1, \boldsymbol{q}_2)$ 为 2 个多元高斯分布之间的距离 Distance(N_1, N_2),而多元高斯分布之间的距离这里采用KL距离进行计算,即

$$\begin{aligned}&\mathrm{Difference}(\boldsymbol{q}_1, \boldsymbol{q}_2)\\ &= \mathrm{KL}(N_1 \parallel N_2) + \mathrm{KL}(N_2 \parallel N_1)\\ &= \frac{1}{2}[trace(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2) +\\ &\quad (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^{\mathrm{T}}(\boldsymbol{\Sigma}_2^{-1} + \boldsymbol{\Sigma}_1^{-1})(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - 2N],\end{aligned}$$

其中

$$\mathrm{KL}(N_1 \parallel N_2) = \frac{1}{2}\left[\ln\left(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}\right) + trace(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - N\right].$$

3.3 谱聚类算法

谱聚类(Spectral Clustering) 的方法有很多,本

文采用的是文献[15]中的算法. 算法描述如下所示.

算法 1　归一化谱聚类

输入　相似矩阵 $S \in \mathbf{R}^{n\times n}$,需要聚类的类数 K

输出　聚类 $A_1, A_2, \cdots, A_K$, $A_i = \{j | y_i \in C_i\}$

step 1　利用 S 构建全联通相似图,W 为权重邻接矩阵;计算归一化的拉普拉斯 L_{sym},以及它的 K 个特征向量 $v_1, v_2 \cdots, v_K$.

step 2　矩阵 $V \in \mathbf{R}^{n\times K}$ 是由向量 $v_1, v_2, \cdots, v_K$ 作为列的矩阵,对 V 进行归一化得到矩阵 $U \in \mathbf{R}^{n\times K}$,使得

$$u_{ij} = \frac{v_{ij}}{\left(\sum_K v_{iK}^2\right)^{\frac{1}{2}}}.$$

step 3　$y_i \in \mathbf{R}^K$ 是 U 的第 i 行,$i = 1, 2, \cdots, n$;利用 k-means 算法将 $(y_i)_{i=1,\cdots,n}$ 聚类为 $C_1, C_2, \cdots, C_K$.

3.4　Ranking SVM 算法

Ranking SVM[12-13] 是一种有效的排序学习算法,它把文档偏序对作为训练样例,优化函数为

$$\min \frac{1}{2}\boldsymbol{\omega}\cdot\boldsymbol{\omega} + C\sum \varepsilon_{i,j,k},$$

$$\forall (d_k^i, d_k^j) \in \boldsymbol{d}_k \cdot \boldsymbol{d}_k : \boldsymbol{\omega}\Phi(q_k, d_k^i) > \boldsymbol{\omega}\Phi(\boldsymbol{q}_k, d_k^i) + 1 - \boldsymbol{\varepsilon}_{i,j,k},$$

其中,$\boldsymbol{\omega}$ 是在学习过程中需要逐步调整的权重向量,参数 C 标识模型复杂性与训练误差之间的折衷程度,$\varepsilon_{i,j,k}$ 是非零的松弛变量,$\Phi(\boldsymbol{q}_k, d_k^i)$ 是从查询 $\boldsymbol{q}_k$ 与文档 d_k^i 到其对应的特征向量 $\boldsymbol{X}_k^i$ 的一个映射.

4　实验及结果分析

4.1　实验设置

4.1.1　实验数据

数据集包含399个查询,每个查询对应着1 000篇网页,查询和网页全部是从商业搜索引擎中采样得到的. 每篇网页(本文中文档和网页表示同一概念)分为关键域和正文两部分,关键域包括网页标题、锚文本等信息. 我们为每个查询、文档对抽取基于链接的特征 PageRank,为关键域和正文分别抽取基于内容的特征:BM25,TFIDF,语言模型对应的特征等,一共23个特征. 查询与文档之间的相关度是由商业搜索引擎的专业标注人员进行标注的,从0到5共6个级别,0代表查询与文档不相关,数值越大,表示查询与文档越相关,5代表查询与文档最相关.

4.1.2　实验设计

实验采用5份交叉验证的方式,数据被平分为5份,3份用于训练,1份用于验证,1份用于测试.

经过对 Ranking SVM 中参数的调整,我们发现当 $C = 0.01$ 的时候,在大多数情况下,训练得到的排序函数都是最优的,所以在对比实验中,C 的值固定为 0.01. 在训练阶段,首先对训练集进行聚类,这里为了验证方法的有效性,我们仅将训练集聚为两类,然后分别对这两类采用 Ranking SVM 算法进行训练,得到两个排序函数 *Cluster_1* 和 *Cluster_2*,在未经聚类的整个训练集上训练得到的排序函数为 *All*. 在测试阶段,如果一个查询经过分类后属于第一类,则采用 *Cluster_1* 和 *All_1*(应用于第一类查询的 *All* 称为 *All_1*). 如果为第二类,则采用 *Cluster_2* 和 *All_2*(应用于第二类查询的 *All* 称为 *All_2*).

然后分别对比 *Cluster_1* 和 *All_1* 的性能,*Cluster_2* 和 *All_2* 的性能. 因为相比 *All* 而言,*Cluster_1* 和 *Cluster_2* 采用较少的训练样例,如果他们的性能能够分别和 *All_1*、*All_2* 具有可比性,甚至超过后者,则证明本文方法是有效的.

4.1.3　实验评测函数

实验中采用 *P@n*,*NDCG@n* 和 *MAP* 这3种评测函数,用于评测排序函数的性能.

P@n 是前 n 个结果中的准确率,它的定义如下:

$$P@n = \frac{1}{n}\sum_{i=1}^{n} rel(d_i),$$

其中

$$rel(d_i) = \begin{cases} 1, & \text{如果 } d_i \text{ 和查询是相关的} \\ 0, & \text{其他情况} \end{cases}$$

$n = 1, 2, \cdots, 10$.

在返回的结果序列中,如果考虑处于不同位置的文档与查询之间相关度的不同,对排序函数性能的影响,那么 *NDCG@n* 是一个非常有用的评测函数[16-17],广泛应用于网络搜索与其他相关的任务中,它的定义如下:

$$NDCG@n = \frac{1}{Z_n}DCG@n = \frac{1}{Z_n}\sum_{i=1}^{n}\frac{2^{rel(d_i)} - 1}{\log(1+i)},$$

其中,Z_n 是在结果序列是最优的情况下的 *NDCG@n* 的值,$n = 1, 2, \cdots, 10$.

MAP 是综合考虑到准确率与召回率的一个评测函数,它的定义如下:

$$MAP = \frac{1}{m}\sum_{j=1}^{m} AP_j$$

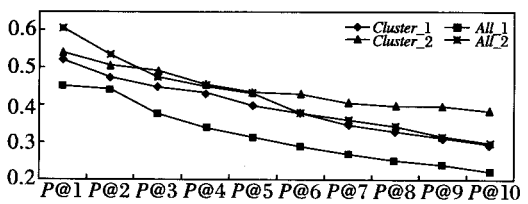
$$= \frac{1}{m}\sum_{j=1}^{m}\frac{1}{R_j}\left(\sum_{i=1}^{k} rel_j(d_i)\cdot(P@i)_j\right),$$

其中,m 是查询的总数,R_j 是第 j 个查询的相关文档的

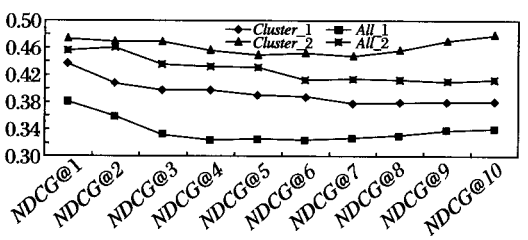
数目,k 是系统为这个查询返回的结果中文档的总数,$rel_j(d_i)$ 和 $(P@i)_j$ 分别是 $rel(d_i)$ 和 $P@i$ 的值.

4.2 实验结果

相比 *All*(即 *All_1*,*All_2*)而言,*Cluster_1* 和 *Cluster_2* 采用的训练样例较少,它们的性能应该分别比 *All_1* 和 *All_2* 差. 如果 *Cluster_1* 和 *Cluster_2* 的性能分别与 *All_1* 和 *All_2* 的性能具有可比性,甚至超过后者,这就证明我们通过聚类后,利用排序学习算法训练排序函数的方法是有效的. 排序函数性能的对比结果,如图 2 所示. *Cluster_1*、*Cluster_2*、*All_1* 和 *All_2* 在 *MAP* 上的结果分别为 0.339 5,0.434 6,0.308 3,0.358 3.



(a) $P@n$



(b) $NDCG@n$

图 2 4 种函数在 $P@n$ 和 $NDGG@n$ 上结果对比

Fig. 2 Performance Comparison of 4 functions on $P@n$ and $NDGG@n$

从对比结果中可以看出,*Cluster_1* 在所有的评测函数上都优于 *All_1*. *Cluster_2* 在 *NDCG@n* 和 *MAP* 两个评测函数上都优于 *All_2*,在评测函数 $P@n$ 上,除了 $P@1$,$P@2$ 两个指标上低于后者外,其余的指标也都优于 *All_2*.

表 2 给出 *Cluster_1* 和 *Cluster_2* 分别相对于 *All_1* 和 *All_2*,在各个评测函数上性能的提升. 如在 *NDCG@1*, *Cluster_1* 相对于 *All_1* 提升 13.2%, *Cluster_2* 相对于 *All_2* 提升 3.6%. 从表 2 可以看出,相比于 *All_1* 和 *All_2*,*Cluster_1* 和 *Cluster_2* 几乎在所有的评测函数上都有显著提高.

4.3 结果分析

从 4.2 节的实验结果中可以看出,*Cluster_1* 和 *Cluster_2* 分别优于 *All_1* 和 *All_2*,但是 *Cluster_1* 和 *Cluster_2* 需要的训练样例只是 *All* 需要的训练样例的一部分,*Cluster_1* 的训练样例占 *All* 的训练样例的比例为 52%,*Cluster_2* 占 48%,两个类别需要的训练样例都只是 *All* 的一半左右,但是性能却优于后者. 这就验证通过聚类后,利用排序学习算法训练排序函数的方式是有效的.

表 2 *Cluster_k* 相对于 *All_k* 在 3 种评测函数上性能的提升 (k = 1,2)

Table 2 Performance improvement of *Cluster_k* to *All_k* on 3 evaluation measures (k = 1,2)

%

	Cluster_1 vs All_1	*Cluster_2 vs All_2*
NDCG@1	+13.2	+3.6
NDCG@5	+16.5	+4.2
NDCG@10	+10.5	+13.9
P@1	+13.3	−12.6
P@5	+20.8	+1.0
P@10	+23.9	+22.4
MAP	+9.2	+17.6

5 结束语

本文提出在对查询进行区分的基础上,对查询进行聚类,从而为每一类别训练不同的排序函数,并采用分类的方法为每一个新来的查询选择一个最适合的排序函数.

本文主要利用聚类的方法区分不同的查询,为不同聚类类别的查询生成不同的排序函数,从而为不同的查询提供不同的排序函数,提高排序函数的性能. 提出线上、线下的工作模式,尽可能地降低用户的响应时间,在不影响用户体验的情况下,提高系统的性能.

未来的工作可以从以下 2 个方面继续进行:1)在每个类别上区分不同特征的适用性,为不同的类别采用不同的特征;2)尝试其他的排序学习算法,为不同的类别采用最合适的排序学习算法.

参 考 文 献

[1] Duh K, Kirchhoff K. Learning to Rank with Partially-Labeled Data // Proc of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Singapore, Singapore, 2008: 251-258

[2] Broder A. A Taxonomy of Web Search. ACM SIGIR Forum, 2002, 36(2): 3-10

[3] Rose D E, Levinson D. Understanding User Goals in Web Search // Proc of the 13th International Conference on World Wide Web. New York, USA, 2004: 13 - 19

[4] Gravano L, Hatzivassiloglou V, Lichtenstein R. Categorizing Web Queries According to Geographical Locality // Proc of the 20th International Conference on Information and Knowledge Management. New Orleans, USA, 2003: 325 - 333

[5] Shen Dou, Sun Jiantao, Yang Qiang, *et al.* Building Bridges for Web Query Classification // Proc of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, USA, 2006: 131 - 138

[6] Liu Yiqun, Zhang Min, Ru Liyun, *et al.* Automatic Query Type Identification Based on Click through Information // Proc of the 3rd Asia Information Retrieval Symposium. Singapore, Singapore, 2006: 593 - 600

[7] Lee U, Liu Zhenyu, Cho J. Automatic Identification of User Goals in Web Search // Proc of the 14th International Conference on World Wide Web. Chiba, Japan, 2005: 391 - 400

[8] Craswell N, Hawking D, Robertson S. Effective Site Finding Using Link Anchor Information // Proc of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New Orleans, USA, 2001: 250 - 257

[9] Westerveld T, Kraaij W, Hiemstra D. Retrieving Web Pages Using Content, Links, URLs and Anchors // Proc of the 10th Text Retrieval Conference. Gaithersburg, USA, 2001: 663 - 672

[10] Kang I, Kim G. Query Type Classification for Web Document Retrieval // Proc of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Toronto, Canada, 2003: 64 - 71

[11] Crammer K, Singer Y. Pranking with Ranking // Proc of the Conference on Neural Information Processing Systems. Whistler, Canada, 2002: 641 - 647

[12] Herbrich R, Graepel T, Obermayer K. Large Margin Rank Boundaries for Ordinal Regression. Advances in Large Margin Classifiers, 2000, 88(2): 115 - 132

[13] Joachims T. Optimizing Search Engines Using Click through Data // Proc of the 8th ACM Conference on Knowledge Discovery and Data Mining. Edmonton, Canada, 2002: 133 - 142

[14] Yue Yisong, Finley T, Radlinski F, *et al.* A Support Vector Method for Optimizing Average Precision // Proc of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Amsterdam, Netherlands, 2007: 271 - 278

[15] Jordan M, Weiss Y. On Spectral Clustering: Analysis and an Algorithm // Dietterich T, Becker S, Ghahramani Z, eds. Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, XIV: 849 - 856

[16] Jarvelin K, Kekalainen J. IR Evaluation Methods for Retrieving Highly Relevant Documents // Proc of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Athens, Greece, 2000: 41 - 48

[17] Jarvelin K, Kekalainen J. Cumulated Gain-Based Evaluation of IR Techniques. ACM Trans on Information Systems, 2002, 20(4): 422 - 446

基于查询聚类的排序学习算法

作者： 花贵春， 张敏， 刘奕群， 马少平， 茹立云， HUA Gui-Chun， ZHANG Min， LIU Yi-Qun， MA Shao-Ping， RU Li-Yun

作者单位： 智能技术与系统国家重点实验室 北京100084;清华信息科学与技术国家实验室(筹) 北京 100084;清华大学计算机科学与技术系 北京100084

刊名： 模式识别与人工智能 ISTIC EI PKU

英文刊名： Pattern Recognition and Artificial Intelligence

年，卷(期)： 2012,25(1)