基于用户相关反馈的排序学习算法研究*

蔡 飞,陈洪辉,舒 振
(国防科技大学 信息系统工程重点实验室,湖南 长沙 410073)

摘 要:在信息检索中,系统需要根据用户查询将文档按照相似度大小进行排序,吸引了众多信息检索和机器学习领域研究者的眼球,并形成了诸多排序算法模型。然而并未考虑到查询短语与文档构成的特征对与用户相关反馈之间存在的同质性。在机器学习算法基础上,通过提取训练样本的主要特征进行有效聚类,并结合用户的相关反馈获取各个类中相关度判断的置信值,形成相似度判定模型,应用该模型来对测试样本进行相关度排序。算法对 LETOR 数据集进行了测试,实验表明,信息检索性能指标比其他排序算法有了进一步提高,并且无需复杂的数据预处理工作和手动设定算法参数。

关键词:信息检索;排序学习;相关反馈;用户行为分析
中图分类号:TP311 **文献标志码**:A **文章编号**:1001-2486(2013)02-0132-05

Learning to rank based on user relevance feedback

CAI Fei, *CHEN Honghui*, *SHU Zhen*
(Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China)

Abstract: Many information retrieval applications have to present their results in the form of ranked lists, in which documents must be sorted in a descending order according to their relevance to a given query. This has led the interest of the information retrieval community in methods that automatically learn effective ranking models, and recently machine learning techniques have also been applied to model construction. Most of the existing methods do not take into consideration the fact that significant homogeneity exists between query-document pairs related to user's feedback. In this research, a novel method which clusters patterns in the training data with their relevance from the user, and then uses the discovered rules to rank documents at query-time. A systematic evaluation of the proposed method using the LETOR benchmark dataset is posposed. The experimental results show that the proposed method outperforms the state-of-the-art methods with no need of time-consuming and laborious pre-processing.

Key words: information retrieval; ranking; relevance feedback; user behavior analysis

近年来,随着互联网的普及以及信息媒体的多样化,信息量以指数级速度不断增长,如何有效获取信息,急需相应的理论和方法来研究。因此,学术界和工业界对信息检索也掀起了一个新的研究高潮[1-2],成为当前信息处理领域的一个研究热点。在此驱动下,形成了一些检索模型,其中影响较大的有:布尔模型[3]、向量空间模型[4]、语言模型[5]、BM25 模型[6]以及基于机器学习的检索算法[7],不断提高信息检索性能,推动着信息检索研究的发展。

同时机器学习算法被逐渐应用到信息检索领域,取得了良好的效果,成为当前信息检索研究的热点问题。2004 年,Nallapati[8] 提出了用支持向量机(Support Vector Machines, SVM)和最大熵(Maximum Entropy, ME)模型对文档排序;2005 年,Gao 等[9] 和 Burges 等[7] 根据相关文档和不相关文档构成的有序对(pair)训练排序模型,分别提出了基于感知机和神经网络模型优化算法;2008 年,Geng 等[10] 对查询短语进行特征表示,根据查询短语之间的差异性,应用 KNN 聚类算法将训练样本分类,学习形成各个类的排序模型,在测试阶段计算测试短语到各类距离,将测试查询分类,并应用相应类模型对文档进行排序;2010 年,Veloso 等[11] 结合统计学和机器学习原理,统计训练集文档单个或数个特征与相关度之间的关系,构造判断模型,在测试阶段根据样本的特征值大小选择相关度判断模型,计算与查询的相关度得分对文档进行排序,但计算复杂度较高,并且需手动设置算法参数。

同时,当用户构造一个好的查询有困难时,让

* **收稿日期**:2012-08-20
基金项目:国家自然科学基金资助项目(61070216)
作者简介:蔡飞(1984—),男,江苏南通人,博士研究生,E-mail:caifei@nudt.edu.cn;
陈洪辉(通信作者),男,教授,博士,博士生导师,chh0808@gmail.com

其来判断文档的相关性却是比较容易的,因此,利用相关反馈进行检索的反复迭代是非常有意义的,并且相关反馈对于跟踪用户信息需求的变化也是有效的。图像检索就是一个使用相关反馈的很好的例子,因为在图像检索中返回结果直观,而且用户不容易用词语来表达其需求,但很容易标记相关和不相关的图像结果。在结合用户反馈的信息检索中,He [20] 等将用户基本反馈信息作为一个特征,同时利用盲反馈信息来构建信息检索模型 FeedbackBoost,信息检索准确率有了较大的提高。Zhang[21] 等受电子商务启发,利用用户配置文件作为反馈信息,结合机器学习算法进行特征选择,对半结构化的文档信息进行检索,该方法同时适用于文档分类。

据此,本文结合机器学习相关研究,并基于如下假设:用户对查询与文档的相关度判断是正确的。在查询和文档的特征层次进行聚类分析,为了减少计算复杂度并保留重要信息,采用 PCA (Principal Component Analysis,主成分分析)获取低维特征表示,再结合用户的相关反馈,形成各个类的相关度判断模型,在测试阶段,获取离样本最近的三个类,利用这三个类的判断模型计算文档相关度得分,并最终给出排序,最后通过对标准测试数据集的实验验证本文算法的有效性和可行性。

1　信息检索中排序问题描述

在信息检索领域,排序算法模型的研究依然存在不少挑战。信息检索中的排序问题就是:在信息检索过程中,系统根据用户提交的查询短语,计算查询与文档库中文档的相关度,并由相关度大小给出排序列表,文档排序位置越前,表示与查询越相关。文档的相关度排序直接反映了检索系统的质量。在此过程中,用户可以对系统给出的若干文档进行相关性判断,系统利用该反馈信息进行再次检索,重新给出排序列表,直至用户检索满意为止,这就形成了基于用户反馈的信息检索过程,可用图 1 表示。

信息检索过程中的排序问题可进一步用如下模型描述:给出训练数据集 D,其组成元素为查询、文档以及对应相关度的组合 $pair\_1$,即 $pair\_1 = \langle q,d,r\rangle$,其中 q 代表查询短语,可进一步由一组单词 t_i 构成,$q = \{t_1,t_2,\cdots,t_n\}$,$d$ 表示文档,由一组特征值 f_j 构成,$d = \{f_1,f_2,\cdots,f_m\}$,$r$ 为两者的相关度,由一组离散值(比如 0,1,2 等)给出,值越大,表示查询与文档越相关。通过训练集数据,构建查询与文档相关度的判断模型 M。在测试阶段,给出测试数据集 T,其组成元素为查询和文档的组合 $pair\_2$,但查询和文档两者相关度未知,即 $pair\_2 = \langle q,d,?\rangle$,$q,d$ 已知。利用训练阶段产生的判断模型 M,计算各个文档和查询的相关性得分,由该估计值大小给出最终的文档排序。

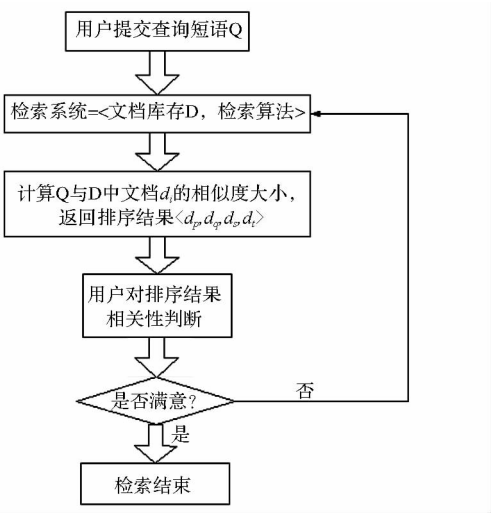

图 1　基于用户反馈的信息检索过程

Fig. 1　Information retrieval based on user feedback

目前,对信息检索算法的性能评估已有不少衡量指标,如查准率(Precision)、查全率(Recall)、F 值(F measure)等,较为通用的有平均准确率(Mean Average Precision,简称 MAP)、P 准确率(P-Precious,简称 P@k)和归一化折损累积增益(Normalized Discounted Cumulative Gain,简称 NDCG)等。本文将使用 MAP 和 P@k 对提出算法进行评估,并与相关的 Baselines 实验比较,分析方法存在的优劣。

2　基于用户反馈的信息检索排序算法

本文假设用户的相关反馈,即对查询与文档的相关度判断是正确的。在此基础上,结合机器学习相关研究,提出基于用户反馈的信息检索排序算法,在查询和文档的特征层次进行聚类分析,同时为了减少计算复杂度,采用 PCA 对特征矩阵进行降维,再结合用户相关反馈信息,计算各个类中对不同查询文档相关度水平判断的可信值,如式(1),其中#表示统计数目,该过程即为训练阶段,其具体过程见图 2。

$$\hat{p}(r\mid cluster_j) = confidence\_ij = \frac{\#document\ of\ level\_i}{\#document\ in\ cluster_j} \tag{1}$$

在测试阶段,计算获取离样本最近的三个类,利用这三个类的判断模型计算文档相关度得分,如式(2)~(4),其中,式(2)中 w_i 反映某个类对排序总得分的权重,由样本离类中心距离决定,由

Training Algorithm: Training dataset → Cluster relevance estimation model

Input: Query-document feature pairs D with corresponding relevance R and cluster numbers K;

Output: Cluster relevance estimation model that is every relevance level with its confidence of each cluster

1: Loading query-document feature matrix D with m * n, row represents examples while column represents features;

2: Dimension reduction by PCA, producing matrix $D\_reduced$ with m * 2;

3: Clustering in lower space by FCM(fuzzy C-means), producing K clusters;

4: Computing each relevance confidence of each cluster;

5: for cluster i = 1 to K

6: for relevance level j = 0 to 2

7: $computing\ confidence\_0 \leftarrow \frac{\text{number of examples with relevance 0}}{\text{total number of examples in the cluster}}$

8: $computing\ confidence\_1 \leftarrow \frac{\text{number of examples with relevance 1}}{\text{total number of examples in the cluster}}$

9: $computing\ confidence\_2 \leftarrow \frac{\text{number of examples with relevance 2}}{\text{total number of examples in the cluster}}$

10: end for

11: end for

12: Return cluster relevance estimation model with confidences of each level.

图2 训练阶段模型生成算法

Fig. 2 Model formulation algorithm in training phase

式(4)而得,式(4)中0.5是归一化系数,使得权重之和为1,$cluster\_score_i$ 根据式(3)对不同类判断模型计算而得,最后计算排序文档的总得分 $rankscore_i$,如式(2),根据 $rankscore_i$ 得分大小给出文档排序,其具体算法详见图3。

Testing Algorithm: Testing dataset → Ranking list

Input: Query-document feature pairs T with no relevance R;

Output: A ranking list of documents sorted according to their relevance to a given query;

1: Loading query-document feature matrix T with m_2 * n;

2: $T\_reduced$ = PCA(T); $T\_reduced$ represent documents in a 2-D space;

3: Computing distance between every input example and each cluster center. Choosing three nearest clusters as relevance estimation model and take down their distance $d1,d2,d3$;

4: Scoring documents associated with given query using the distance as weight;

5: for document i = 1 to M

6: $rankscore_i = \sum_{i=1}^{3} w_i * cluster\_score_i$;

7: while $w_i = \frac{\sum_{j\in((1,2,3)\backslash i)} distance_j}{\sum_{i=1}^{3} distance_i}$

$cluster\_score_i = \sum_j relevance\_level_j * confidence_j$

8: end for

9: Return a ranking list of documents sorted according to their relevance estimation result to a given query

图3 测试阶段文档排序算法

Fig. 3 Proposed learning to rank algorithm in testing phase

$$rankscore_i = \sum_{i=1}^{3} w_i \times cluster\_score_i \quad (2)$$

$$cluster\_score_i = \mathrm{E}(r \mid cluster_j) = \sum_j relevancelevel_j \times \hat{p}(r \mid cluster_j) \quad (3)$$

$$w_i = 0.5 \times \frac{\sum_{j\in(\mid 1,2,3\mid \backslash i)} distance_j}{\sum_{i=1}^{3} distance_i} \quad (4)$$

3 实验结果与分析

3.1 数据集与 Baselines

亚洲微软研究院①为排序算法研究提供了免费的测试集 LETOR[12],包含三个子集:OHSUMED、TD2003、TD2004,同时列出了一些典型算法对 LETOR 的排序性能进行评估,如:Ranking SVM[13],RankBoost[14],AdaRank[15],FRank[16],ListNet[17] 和 MHR[18]。每个测试集中的数据为形如前文所述的查询、文档对形式 $pair\_1 = \langle q,d,r\rangle$,其中文档特征共有25个,如词频 TF、逆文档频率 IDF、TF × IDF、BM25 score 等。OHSUMED 测试集包含了106个查询,对于每个查询有几十或几百的文档与之关联,并给出相关度视为用户反馈信息。

3.2 性能评估指标

首先我们使用 MAP[19] 来衡量信息检索排序性能,如式(5)所示,$\mid Q \mid$ 为查询数目,$precision(R_{jk})/m_j$ 为对于某个查询,每个相关文档的理论位置与算法排序位置的比。

$$MAP(Q) = \frac{1}{\mid Q \mid} \sum_{j=1}^{\mid Q \mid} \frac{precision(R_{jk})}{m_j} \quad (5)$$

其次,据 iProspect Blended Search Results Study② 2008年统计,如图4,给出提问"当你在搜索引擎进行搜索时,你通常检查几个结果后点击一个链接?"的统计结果显示,大多数用户只会浏览排序靠前的数个网页,因此我们关注返回结果中靠前的准确率,采用 p@k[19] 对算法进行评估,如式(6),k 表示统计结果前 k 个文档,$relevant\_num_k$ 表示这 k 个文档中相关文档数目,该指标优点在于无需计算相关文档集合的数目。

$$p@k = \frac{relevant\_num_k}{k} \quad (6)$$

	2008	2006	2004	2002
Only a few	27%	23%	24%	16%
The first page	41%	39%	36%	32%
The first 2 pages	17%	19%	20%	23%
The first 3 pages	7%	9%	8%	10%
More than 3 pages	8%	10%	12%	19%

图4 近些年网络用户点击检索结果位置分布统计

Fig. 4 Where's the web-page clicked by user in the ranking list

① http://research.microsoft.com/users/LETOR/

② http://www.iprospect.com

3.3　实验评估与分析

本文基于 Matlab7.8 开发环境,实现了上述算法,实验中聚类个数设置为 5,便于将算法移植至其他相关度水平有 5 层的测试集,从 0 到 4 分别表示不相关至完全相关,使用该算法时只需设置类别数 $K > 3$ 即可。

实验中将 OHSUMED 测试集分为五部分:S1 ~ S5,将其中三部分数据作为训练,一部分数据作为测试样本,实验数据选择情况详见表 1。表 2 和表 3 给出了采用 MAP 和 P@k 指标对本文算法与其他排序算法的实验结果比较,粗字体表示本次实验中最佳方法结果。根据表 2,RankBoost 的总体实验结果最差(0.440),而本文提出算法结果最佳(0.451),是因为 OHSUMED 测试集提供的特征较少,并且特征主要基于文本信息提取,比如 TF、IDF、BM25 等,而没有加入文本以外特征信息,如用户浏览时间(Url Dwell Time)、查询文档对应点击数(Query-url Click Count)等,难以使算法性能显著提高,虽然本文算法无法在每次实验中均取得最佳效果,但依然比 Baseline 中最佳效果(ListNet 的 0.449)高出 0.45%,图 5 更直观地比较了各种算法性能,柱状图上方依次为各种算法结果和本文算法提高百分比。

表 1　实验数据选择分布情况

Tab. 1　Dataset partition

实验	Training	Testing
Trial 1	S1、S2、S3	S5
Trial 2	S2、S3、S4	S1
Trial 3	S3、S4、S5	S2
Trial 4	S4、S5、S1	S3
Trial 5	S5、S1、S2	S4

由 3.2 节内容可知,网络用户在信息检索过程中,对返回结果中位置靠前的文档点击率较高,因此本文又利用 P@k 对算法进行了实验比较,结果如表 3 所示,括号中数字表示本文算法与对应 Baseline 算法的比较结果,图 6 给出了各种方法准确率随统计结果数 k 的变化情况。随着 k 的增大,返回结果中不相关文档数目增加,这同时表明,返回结果中位置靠前的文档与查询短语相似度较大,满足网络用户信息检索需求。

表 2　MAP 评估 OHSUMED 数据集结果

Tab. 2　MAP for OHSUMED dataset

Methods	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Avg	**Improvement**
Ranking SVM	0.333	0.451	0.459	0.511	0.479	0.446	+1.12%
RankBoost	0.339	0.446	0.445	0.505	0.463	0.440	+2.50%
FRank	0.344	**0.460**	0.448	**0.541**	**0.462**	0.446	+1.12%
ListNet	0.345	0.449	0.466	0.517	0.468	0.449	+0.45%
AdaRank　MAP	0.341	0.448	0.457	0.507	0.454	0.441	+2.27%
NDCG	0.348	0.449	0.457	0.509	0.447	0.442	+2.04%
MHR	0.329	0.442	0.456	0.501	0.470	0.440	+2.50%
PROPOSED	**0.358**	0.453	**0.474**	0.519	0.449	**0.451**	——

表 3　P@k 评估 OHSUMED 数据集结果

Tab. 3　Average P@k of trials for OHSUMED dataset

Methods	P@1	P@2	P@3	P@4	P@5	P@10
Ranking SVM	0.633 (+6.16%)	0.619 (+2.91%)	0.592 (+5.24%)	0.578 (+3.98%)	0.576 (+0.00%)	0.507 (+3.35%)
RankBoost	0.604 (+11.2%)	0.595 (+7.06%)	0.586 (+6.31%)	0.562 (+6.94%)	0.544 (+5.88%)	0.495 (+5.86%)
FRank	0.670 (+0.30%)	0.618 (+3.07%)	0.617 (+0.97%)	0.581 (+3.44%)	0.559 (+3.04%)	0.485 (+8.04%)
ListNet	0.642 (+4.87%)	0.628 (+1.43%)	0.602 (+3.49%)	0.576 (+4.34%)	0.574 (+0.35%)	0.509 (+2.95%)
AdaRank　MAP	0.661 (+1.66%)	0.604 (+5.46%)	0.583 (+6.86%)	0.567 (+6.00%)	0.537 (+7.26%)	0.490 (+6.94%)
NDCG	0.633 (+6.16%)	0.604 (+5.46%)	0.570 (+9.30%)	0.562 (+6.94%)	0.533 (+8.07%)	0.490 (+6.94%)
MHR	0.652 (+3.07%)	0.614 (+3.75%)	0.611 (+1.96%)	0.590 (+1.86%)	0.565 (+1.95%)	0.502 (+4.38%)
PROPOSED	0.672	0.637	0.623	0.601	0.576	0.524

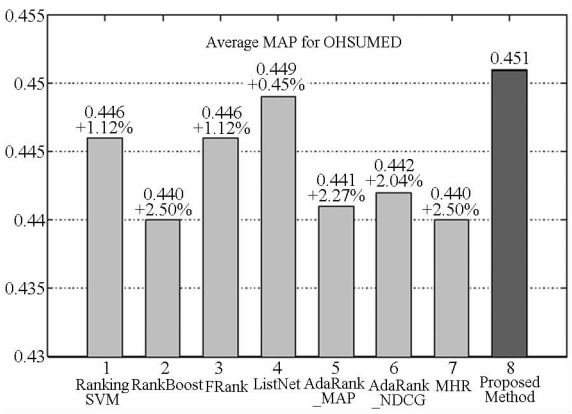

图5 OHSUMED 数据集 MAP 平均值

Fig. 5 Average MAP for OHSUMED dataset

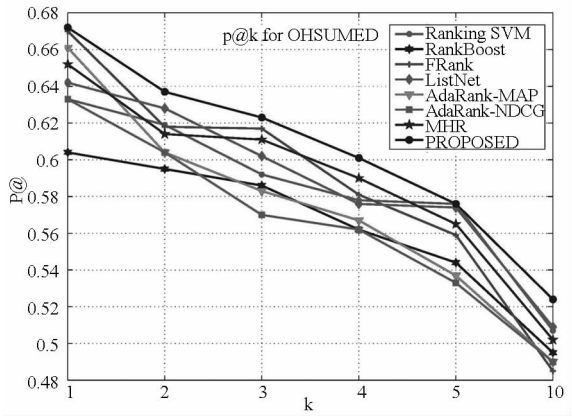

图6 OHSUMED 数据集 P@k 指标

Fig. 6 P@k for OHSUMED dataset

4 结束语

本文对基于用户反馈的信息检索问题进行了描述,结合机器学习研究成果,提出了一种基于模式聚类和用户相关反馈相结合的排序算法。该算法利用训练集数据学习产生查询短语与文档的相似度判断模型,并利用该模型对测试数据中查询与文档的相关度进行估计,给出文档的排序列表,实验结果显示,所提出的算法具有更高的信息检索性能,并无需复杂的数据预处理和参数调整工作,从而具有很好的实用性。

未来进一步研究可在以下方面展开:(1)本文利用主成分分析将高维特征数据映射至低维空间,必然会丢失部分信息,可利用回归思想,拟合相关度与特征变量的函数曲线,结合灵敏度分析,从高维特征中选择重要特征,从而获取低维空间的特征表示。(2)进一步分析用户的间接反馈信息,如鼠标点击记录、浏览时间等,研究查询与文档的判断模型,提高信息检索性能。

参考文献(References)

[1] Lv Y H, Zhai C X, Chen W. A boosting approach to improving pseudo-relevance feedback[C]//Proceedings of ACM SIGIR, 2011:165 - 174.

[2] Wang L D, Lin J, Metzler D. A cascade ranking model for efficient ranked retrieval[C] //Proceedings . of ACM SIGIR, 2011:105 - 114.

[3] Ricardo B Y, Berthier R N. Modern information retrieval[M]. Addison Wesley, 1999.

[4] Salton G. The SMART retrieval system-experiments in automatic document processing[R]. Prentice-Hall, NJ, USA, 1971.

[5] Lafferty J, Zhai C X. Document language models, query models, and risk minimization for information retrieval[C]// Proceedings of ACM SIGIR, 2001:111 - 119.

[6] Robertson S E. Overview of the okapi projects[J]. Journal of Documentation, 1998: 275 - 281.

[7] Burges C, Shaked T, Renshaw E, et al. Learning to rank using gradient descent[C] //Proceedings of ACM ICML: 2005:89 -96.

[8] Nallapati R. Discriminative models for information retrieval[C] //Proceedings of ACM SIGIR, 2004:64 -71.

[9] Gao J F, Qi H , Xia X, et al. Linear discriminant model for information retrieval[C] //Proceedings of ACM SIGIR, 2005: 290 -297.

[10] Geng X B, Liu T Y, Qin T, et al. Query dependent ranking using k-nearest neighbor[C] //Proceedings of ACM SIGIR, 2008:115 -122.

[11] Veloso A, Goncalves M, Meira W, Jr, et al. Learning to rank using query-level rules[J]. Journal of Information and Data Management, 2010, 1(3): 567 - 581.

[12] Liu T Y, Xu J, Qin T, et al. LETOR:Benchmark dataset for research on learning to rank for information retrieval[C] // Proceedings of ACM SIGIR, Workshop, 2007.

[13] Herbrich R, Graepel T, Obermayer K. Large margin rank boundaries for ordinal regression[M]. MIT Press, 2000.

[14] Freund Y, Iyer R, Schapire R, et al. An efficient boosting algorithm for combining preferences[J]. Journal of Machine Learning Research, 2003(4):933 - 969.

[15] Xu J, Li H. Adarank: A boosting algorithm for information retrieval [C] //Proceedings of ACM SIGIR, 2007: 391 - 398.

[16] Tsai M F, Liu T Y, Qin T, et al. FRank:A ranking method with fidelity Loss[C] //Proceedings of ACM SIGIR, 2007: 383 -390.

[17] Cao Z, Qin T, Liu T Y, et al. Learning to rank: From pair-wise approach to list-wise approach [C] //Proceedings of ACM SIGIR, 2007:129 - 136.

[18] Qin T, Zhang X D, Wang D S, et al. Ranking with multiple hyperplanes[C]//Proceedings of ACM SIGIR, 2007: 279 -286.

[19] Manning C D, Raghavan P, Schutze H. An introduction to information retrieval[M]. Cambridge University Press, 2009.

[20] He J, Zhao W X, Shu B H, et al. Efficiently collecting relevance information from clickthroughs for web retrieval system evaluation[C]//Proceedings of ACM SIGIR, 2011: 275 -284.

[21] Zhang L B, Zhang Y, Xing Q L. Filtering semi-structured documents based on faceted feedback[C] //Proceedings of ACM SIGIR, 2011:645 -654.