基于多示例学习的中文 Web 目录页面推荐*

黎 铭, 薛晓冰, 周志华+

(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

Chinese Web Index Page Recommendation Based on Multi-Instance Learning

LI Ming, XUE Xiao-Bing, ZHOU Zhi-Hua+

(National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

+ Corresponding author: Phn: +86-25-83686268, E-mail: zhouzh@nju.edu.cn, http://cs.nju.edu.cn/people/zhouzh/

Received 2004-02-12; Accepted 2004-05-09

Li M, Xue XB, Zhou ZH. Chinese Web index page recommendation based on multi-instance learning. *Journal of Software*, 2004,15(9):1328~1335.

http://www.jos.org.cn/1000-9825/15/1328.htm

Abstract: Multi-Instance learning provides a new way to the mining of Chinese web pages. In this paper, a particular web mining task, i.e. Chinese web index page recommendation, is presented and then addressed through transforming it to a multi-instance learning problem. Experiments on the real world dataset show that the proposed method is an effective solution to the Chinese web index page recommendation problem.

Key words: multi-instance learning; Web mining; machine learning; Chinese Web index page recommendation; prefix tree

摘 要: 多示例学习为中文 Web 挖掘提供了一种新的思路.提出中文 Web 目录页面推荐这种特殊的 Web 挖掘任务,并且将其转化为多示例学习问题来解决.在真实世界数据集上的实验结果显示,该方法能够有效地解决该问题.

关键词: 多示例学习;Web 挖掘;机器学习;中文 Web 目录页面推荐;前缀树

中图法分类号: TP183 **文献标识码**: A

互联网技术的不断成熟和发展,使得在全球范围内实现资源共享和信息交换成为可能.各种资源和信息以不同的格式、不同的存储方式分布在这个庞大分布式系统的各个结点上,并以一些固定的访问方式来提供用户使用.互联网上这些巨量的、无固定结构的信息,使用户从中有效找出自己感兴趣的部分变得越发困难.同时,从巨量信息中发掘出个性化信息和知识的要求也越来越大.因此,Web 挖掘技术[1]应运而生.

* Supported by the National Natural Science Foundation of China under Grant No.60105004 (国家自然科学基金); the National Outstanding Youth Foundation of China under Grant No. 60325207 (国家杰出青年科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312002 (国家重点基础研究发展规划(973))

作者简介: 黎铭(1980－),男,湖南长沙人,硕士生,主要研究领域为机器学习,数据挖掘;薛晓冰(1982－),男,硕士生,主要研究领域为机器学习,数据挖掘;周志华(1973－),男,博士,教授,博士生导师,主要研究领域为机器学习,数据挖掘,模式识别,信息检索,神经计算.

Web 挖掘技术大体上可以分成 3 大类[2]:Web 内容挖掘,用于发现 Web 文档数据中的知识;Web 结构挖掘,用于发现 Web 页面之间超链接相互指向的关系;Web 用户日志挖掘,用于预测用户在 Web 上交互式信息查询中的行为.其中,分析用户对于网上资源使用的偏好,能够更有效地为用户提供其可能感兴趣的信息,还可以帮助优化资源的分配,因此,Web 用户日志挖掘非常重要.

本文所要解决的中文 Web 目录页面推荐问题正是一种特殊的 Web 用户日志挖掘问题.通过对用户进行 Web 目录页面推荐,告诉用户当前浏览的 Web 目录页面中是否可能包含该用户感兴趣的内容页面,从而可以节省用户随着链接到达自己不感兴趣的 Web 页面的时间.此外,该技术还可以用于中间件服务器上,以有效地指导由客户端发起 Web 页面复本缓存,平衡网络负载.在进行中文 Web 页面推荐时,本文把目录页面及其相关联的内容页面看成一个个的包,从而将该问题转化为一个多示例学习(multi-instance learning)[3]问题.实验证明,通过多示例学习的方法能够有效地解决中文 Web 目录页面推荐问题,并且取得比传统方法更好的结果.

本文第 1 节简单介绍多示例学习以及 Web 目录页面推荐问题.第 2 节介绍对于中文 Web 页面的特征抽取和包生成方法.第 3 节叙述针对目录页面推荐问题提出的 Fretcit-*k*NN 算法.第 4 节给出实验结果.最后是结束语.

1 多示例学习和 Web 目录页面推荐

1.1 多示例学习

20 世纪 90 年代中期,Dietterich 等人[3]在对药物活性预测问题的研究中首先提出了多示例学习这个概念.在多示例学习问题中,训练集不再是由若干示例组成,而是由一组含有概念标记的包(bag)组成,每个包是若干没有概念标记的示例集合.如果一个包中至少存在一个正例,则该包被标记为正包;如果一个包中不含有任何正例,则该包为反包.学习系统通过对已经标定类别的包进行学习来建立模型,希望尽可能正确地预测不曾遇到过的包的概念标记.与监督学习相比,多示例学习中的示例是没有概念标记的,这与监督学习中所有示例都有概念标记不同.因此,多示例比监督学习更加困难.Dietterich 等人[3]发现,C4.5 决策树、BP 神经网络等常用的监督学习算法很难用于解决多示例问题.

然而,由于多示例学习具有独特的性质和广泛的应用前景,属于以往机器学习研究的一个盲区,因此在国际机器学习界引起了广泛的反响,被认为是和监督学习、非监督学习、强化学习并列的一种学习框架[4].研究者们已经提出了很多多示例学习的算法,例如:APR(axis-parallel rectangle)[3],Diverse Density[5],Citation-*k*NN[6],ID3-MI[7],RIPPER-MI[7],BP-MIP[8]等.Zhou 和 Zhang[9]将集成学习(ensemble learning)技术用于多示例学习,在基准测试上取得了迄今最好的结果,他们还指出,将传统的监督学习的着眼点从示例的层次上升至包的层次,就可以将传统的监督学习算法改造为多示例学习算法.此外,还有一些学者对 APR 算法的 PAC-可学习性进行了研究,并得到了一些有意义的结论[10,11].

1.2 Web目录页面推荐

在 World Wide Web 中富含信息的网页通过超链接相连,形成了一个庞大的网状结构.用户需要沿着超链接去寻找自己感兴趣的信息.这势必会经常出现用户通过看似相关的超链接而访问到与自己实际信息需求无关的网页.如果能够在用户访问某个页面时就告诉用户,该页面中是否包含了他可能感兴趣的内容,就可以帮助用户更有效地访问网上的信息资源.

虽然 World Wide Web 上信息组织方式因不同的网站而有所区别,但是大体上都符合一种隐含的二级层次结构,即把详细内容写在一个页面中,而把和该页面内容相关的词语或句子作为指向该页面的超链接放在另一个页面中,作为该页面的入口索引.本文中那些陈述内容的页面称为内容页面(content page),包含了大量指向内容页面的超链接的页面称为目录页面(index page).在目录页面中,一条超链接唯一地代表了与之相关联的内容页面.对于各大门户网站,例如:www.sina.com.cn,各栏目页面就是一个目录页面.图 1(a)是一个新浪国内新闻的栏目页面,它包含了若干指向具体新闻页面的目录,其中圆圈中的一条目录对应于图 1(b)中的内容页面.

(a) Index page (b) Content page

(a) 目录页面 (b) 内容页面

Fig.1 Index page and content page

图 1 目录页面和内容页面

通过分析用户所遇到的目录页面以及用户是否认为该目录页面包含自己感兴趣的内容,可以得到某个特定用户在信息需求上的偏好.当浏览新的目录页面时,用户能够得到根据自己偏好分析所得出的反馈,它指出当前目录页面是否可能包含用户感兴趣的信息.本文中称这个过程为 Web 目录页面推荐(Web index recommendation).

在 Web 目录页面推荐过程中,用户只需指出某个目录页面中是否包含其感兴趣的内容,而无须指出感兴趣的具体链接,这样不仅便于用户使用,还可以简化人机界面的设计.但这样一来,与请求用户具体指出感兴趣的链接相比,推荐问题变得更加复杂.幸运的是,如果把每个内容页面看成一个示例,目录页面就是包含若干示例的包,那么 Web 目录页面推荐问题就映射为一个多示例学习问题,这样就可以利用多示例学习技术来解决这个问题.

2 特征抽取

2.1 Web页面的特征抽取

内容页面是用户感兴趣信息的主要载体,从内容页面中有效抽取最能表征用户感兴趣信息的特征是精确进行 Web 目录页面推荐的必要前提.一个内容页面中包含了图像、动画、音频、超链接等丰富的信息表达方式,但最主要的信息传递方式还是正文的文字信息.为了简单起见,本文只处理正文信息.

通常情况下,一篇文章中出现频率高的词汇都从某一个侧面反映了文章的主题.当然,对于那些诸如“如果”、“但是”、“而且”等无意义的虚词将不作考察.这些高频词是传递文章所包含信息的关键性词汇,因此可以用其作为属性值来表示整篇文章.

本文使用内容页面正文中出现频率最高的 p 个词汇,形成一个 p 维特征向量 $\boldsymbol{W}=[w_1,w_2,\ldots,w_p]^T$ 来代表内容页面,其中 $w_i(i=1,2,\ldots,p)$ 是 $\boldsymbol{W}$ 对应的内容页面正文中出现频率第 i 高的词汇.一个包含有 m 个内容页面的目录页面就可以表示成一个含有 m 个示例的包 $Bag=\{[w_{11},w_{12},\ldots,w_{1p}]^T,[w_{21},w_{22},\ldots,w_{2p}]^T,\ldots,[w_{m1},w_{m2},\ldots,w_{mp}]^T\}$.这样,对于中文 Web 页面特征抽取的关键就是如何准确地从中文 Web 页面中提取出高频词.

2.2 中文高频词提取

与英文不同,中文的词汇不像英语中的单词那样是自然分割的,而是词和词之间紧密连接成为句子.句子中的词汇需要人为地通过语境来切分,同一句话所表达的意思会因不同的切分方式而有所不同.

在中文分词问题上,主要有两大类解决办法,一是基于词典的分词,另一种是无词典的分词.由于 Web 页面推荐中所涉及到的词汇内容广泛,并且可能包括相当数量的专用名词,显然,基于词典的分词方法难以满足要求.同时,这些所需要的词汇必须在出现频率上满足一定的要求.因此,使用一些基于词汇出现频率统计的无词典的分词技术[12],可以较准确地提取出文本中的高频词.

本文在词频统计基础上引入前缀树结构,在不影响查找时间的同时有效地降低存储开销.前缀树是根据字

串的前缀组织而成的树型结构,如图 2 所示.前缀树第 i 层的每个结点中都包含了一个长度为 i 的字串以及该字串在文中出现的频率.父结点中的字串是子结点中字串的最大前缀.互为兄弟结点的字串仅最后一个汉字不同.

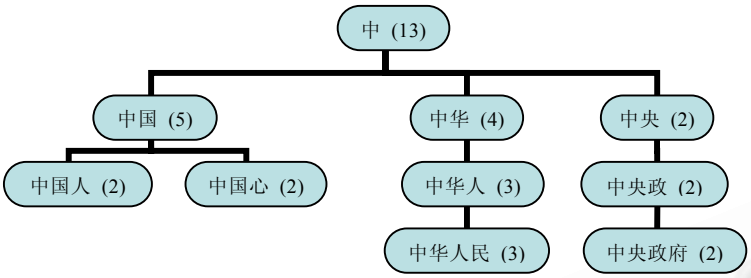

Fig.2 An instance of prefix tree
图 2 前缀树的一个实例

对于高频词的抽取可以分成预处理、字串形成和后处理 3 个阶段.在预处理阶段,首先利用绝对切分标记、条件切分标记以及特殊字符[12]将征文内容划分为若干字串,然后分别统计每个字的出现频率,对于大于某个阈值 f_{min} 的字记录其在文中的位置.在字串形成阶段,针对每一个字频高于 f_{min} 的汉字 C,构造一棵以 C 为根的前缀树,并把文中所有以 C 为第 1 个汉字并且 C 后面每个汉字出现频率都大于 f_{min} 的字串加入到前缀树中.如果有包含该字串的结点,则把该结点的频率加 1.最后从前缀树中导出候选高频词集合.具体算法如图 3 所示.在后处理阶段,为了得到最小的高频词集合,需根据预先设定好的阈值从候选高频字串集合中除去冗余的汉字模式[13].设字串 C_1 是字串 C_2 的前缀或后缀,w=Freq(C_1)/Freq(C_2),其中 Freq(C)表示字串 C 在文章中出现的频率.若 $w\le 0.3$,舍弃 C_1;若 $w\ge 0.9$,舍弃 C_2,这里 0.3 与 0.9 均为经验值[13].

Algorithm: Extract_FreqStr Extracting frequent string from given text
Input: the given text T, the minimum frequency required f_{min}
Output: the set of frequent string contained in the text, S
Process:
- Initialize $S\leftarrow\varnothing$
- For each character c in T with Freq(c)$\ge f_{min}$
  - Build a prefix tree *PreTree* rooted at c
  - For each appearance of c in T
    - Set *startPos* to the position of the current appearance of c in T
    - *tmpPos*$\leftarrow$(*startPos*+1)
    - While the character c' indexed by *tmpPos* is not a separator and Freq(c')$\ge f_{min}$
      - Form *tmpStr* begin with c and ended with c'
      - Add *tmpStr* to *PreTree*
      - *tmpPos*$\leftarrow$(*tmpPos*+1)
  - Traverse *PreTree* add frequent words to S

Fig.3 Algorithm for extracting high frequency candidate words
图 3 候选高频字串抽取算法

3 Fretcit-*k*NN 算法

k 近邻(k-nearest neighbor)算法是一种经典的惰性学习(lazy learning)[14]算法.由于用户在进行浏览时,其 Web 日志是不断积累的,为了更好地利用这些数据,本文使用 k 近邻算法来进行学习.k 近邻算法的关键是如何度量两个样本之间的距离,通常采用欧式距离来度量.在多示例问题中,样本是包含多个示例的包.为了使 k 近邻算法能够适用于多示例学习,就必须给出两个包之间距离的度量方法.Wang 和 Zucker 在他们提出的扩展 k 近邻算法[6]中引入 Hausdorff 距离[15]来度量包之间的距离.通俗地说,两个集合 A,B 之间的 Hausdorff 距离小于等于 d 当且仅当 A 中的每一个元素到至少一个 B 中的元素的距离不超过 d,同时,B 中的每一个元素到至少一个 A 中的元素的距离不超过 d.由于 Hausdorff 距离对于噪声比较敏感,他们建议在实际应用中使用最小 Hausdorff 距离(minimum Hausdorff distance):

$$\min H(A,B)=\max\{h(A,B),h(B,A)\} \tag{1}$$

其中, $h(A,B)=\min_{a\in A}\min_{b\in B}\|a-b\|$.

然而直接使用最小 Hausdorff 距离作为距离度量的 k 近邻算法有时候并不能很有效地解决多示例问题.概念标记为正的包中可能存在伪正例(false positive instance)[6],而这些伪正例会吸引反包.在使用最小 Hausdorff 距离计算一个含有一定数量伪正例的包的 k 个最近邻时,可能会因为反包的数量超过正包的数量,从而造成错误分类.为此,Wang 和 Zucker[6]提出了鲁棒性更好的 Citation-kNN 算法.该方法在使用多数投票对一个未知包 x 进行分类时,不仅要考虑 r 个离它最近的包的概念标记,同时还要考虑把 x 作为 c 近邻的所有包的概念标记,然后一并统计投票结果.如果投票相等,则 x 为反包.

采用上述方法,可以较好地解决典型的多示例学习问题,例如药物活性预测问题[3].然而对于本文中提出的 Web 目录页面推荐问题来说,所有示例的属性都是非数值可枚举型的,不能像对待数值型属性那样直接使用欧氏距离进行计算.如何计算两个示例的特征向量之间的距离成为重要的问题.本文在第 2 节提到,每一个内容页面使用一个 p 维特征向量来表示,第 i 维的属性值为对应的内容页面中的第 i 频繁词.从直观上说,如果两个内容页面所传达的信息内容越相近,它们所对应的特征向量中高频词相同的概率也应该越大,因此,如果两个特征向量包含相同的词汇越多,距离就应该越小.根据这个启发式原则,本文定义如下距离计算方法.

设两个示例 $\boldsymbol{a}=[x_1,x_2,\ldots,x_p]^T$,$\boldsymbol{b}=[y_1,y_2,\ldots,y_p]^T$ 是 p 维特征向量,则 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 之间的距离为

$$\|\boldsymbol{a}-\boldsymbol{b}\|=1-\frac{1}{p}\sum_{i,j=1}^{p}\delta(x_i,y_j) \tag{2}$$

其中,$\delta(x,y)=1$ iff $x=y$.

这样,把式(1)中计算两个示例之间距离的部分用式(2)来代替,就得到了频繁项最小 Hausdorff 距离(frequent term minimum Hausdorff distance).将其作为 Citation-kNN 的距离度量,便得到了适合于解决 Web 目录页面推荐问题的方法 FREquent Term CITation-kNN,简记为 Fretcit-kNN.

在使用 Fretcit-kNN 对一个未知包 x 的概念标记进行预测时,首先需要计算 x 和训练集中其他包之间的频繁项最小 Hausdorff 距离,从而找出 x 的 r 个最近邻以及所有把 x 作为 c 近邻的包.然后根据这些包的多数投票结果来确定 x 的概念标记.

4 实验及结果

本文采用来自于真实世界的数据来检验 Fretcit-kNN 算法在解决中文 Web 目录页面推荐问题上的有效性.实验数据由 117 个目录页面及其所有相关的内容页面产生,这些页面分别来自新浪、搜狐等知名中文门户网站.整个数据集未经过压缩的大小为 854MB,每个目录页面最多包含 247 个链接,最少包含 26 个.8 个志愿者分别在浏览了每个目录页面及其相关的内容页面之后,按照如下规则给每一个目录页面添加一个概念标记:对于当前浏览的目录页面,如果能够通过其包含的某条超链接访问到自己所感兴趣的内容页面,则标记由该目录页面所生成的包为正包;如果该目录页面中所有目录所相关的内容页面自己都不感兴趣,则标记由该目录页面所生成的包为反包.把这 8 组不同的概念标记分别与从这 117 个目录页面生成的包相结合,就得到了 8 个不同的中文 Web 目录页面推荐问题的数据集,其分别包含的正例数和反例数见表 1.针对上述每一个数据集,随机挑选 66%的数据(77 个样本)作为训练集,余下的 34%的数据(40 个样本)用作测试.其中,训练集和测试集中正例与反例的分布与原数据集相同.

本文使用两种常用的文本分类算法和 Fretcit-kNN 算法作比较.一种是 TFIDF 算法[16],它把每个文本表示成为一个特征向量,并分别导出一个表示正例的特征向量和一个表示反例的特征向量,然后计算待分类文本对应向量和这两个向量夹角的余弦值,待分类文档的概念标记被置成余弦值最大的向量的概念标记.另一种算法是普通的 k 近邻算法,它使用式(2)来计算两个示例之间的距离.本文中称其为 Txt-kNN 算法.对于上述两种非多示例学习算法,目录页面及其所有相关内容页面的正文连接成的一段文字就看成一个示例,从中提取高频词作为特征向量.

Table 1 Distribution of positive and negative instances in the data sets

表 1 数据集中正例和反例分布情况

Dataset	Positive	Negative
U1	47	70
U2	43	74
U3	96	21
U4	84	33
U5	56	61
U6	39	78
U7	79	38
U8	39	78

Table 2 Experimental results of TFIDF, Txt-*k*NN and Frecit-*k*NN with 5-dimention feature vector

表 2 TFIDF,Txt-*k*NN 和 Fretcit-*k*NN 在 5 维特征向量下的比较结果

Data	TFIDF			Txt-*k*NN			Fretcit-*k*NN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
U1	.769	.400	1.00	.692	.667	.588	.641	.400	.545
U2	.718	.214	1.00	.872	.643	1.00	.795	.714	.714
U3	.974	1.00	.970	.949	1.00	.941	.974	1.00	.970
U4	.769	.857	.828	.692	.821	.767	.718	.893	.758
U5	.821	.947	.750	.872	.842	.889	.846	.895	.810
U6	.590	.538	.412	.615	.462	.429	.795	.538	.778
U7	.949	1.00	.929	.974	.962	1.00	1.00	1.00	1.00
U8	.872	.769	.833	.872	.923	.750	.949	.923	.923
Avg	.808	.716	.840	.817	.790	.796	.840	.795	.812

Table 3 Experimental results of TFIDF, Txt-*k*NN and Frecit-*k*NN with 7-dimention feature vector

表 3 TFIDF,Txt-*k*NN 和 Fretcit-*k*NN 在 7 维特征向量下的比较结果

Data	TFIDF			Txt-*k*NN			Fretcit-*k*NN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
U1	.769	.400	1.00	.615	.533	.500	.718	.533	.667
U2	.718	.214	1.00	.769	.500	.778	.795	.714	.714
U3	.974	1.00	.970	.949	.969	.969	.923	1.00	.914
U4	.692	.750	.808	.641	.821	.719	.744	.821	.821
U5	.821	1.00	.731	.872	.895	.850	.846	.947	.783
U6	.590	.462	.400	.769	.462	.750	.641	.308	.444
U7	.923	.962	.926	.949	.962	.962	1.00	1.00	1.00
U8	.923	.846	.917	.846	.923	.706	.949	.923	.923
Avg	.801	.704	.844	.801	.758	.779	.827	.781	.783

Table 4 Experimental results of TFIDF, Txt-*k*NN and Frecit-*k*NN with 10-dimention feature vector

表 4 TFIDF,Txt-*k*NN 和 Fretcit-*k*NN 在 10 维特征向量下的比较结果

Data	TFIDF			Txt-*k*NN			Fretcit-*k*NN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
U1	.769	.400	1.00	.667	.600	.563	.692	.533	.615
U2	.718	.214	1.00	.846	.643	.900	.871	.929	.765
U3	.923	.969	.939	.897	.969	.912	.974	1.00	.970
U4	.59	.607	.773	.718	.893	.758	.846	.964	.844
U5	.821	1.00	.731	.923	.947	.900	.923	1.00	.864
U6	.641	.538	.467	.692	.385	.556	.718	.385	.625
U7	.974	1.00	.963	.949	.923	1.00	1.00	1.00	1.00
U8	.923	.846	.917	.846	.923	.706	.974	1.00	.929
Avg	.795	.697	.849	.817	.785	.787	.875	.851	.827

在实验中,Fretcit-*k*NN 的参数 r 和 c 分别被设置为 3 和 5,Txt-*k*NN 中的 k 值设置为 3.由于特征向量中的高频词的个数直接影响到特征向量对示例的表征能力,为此,实验需要在高频词个数不同的情况下来比较上述 3 种算法.在高频词个数一定的情况下,首先得到 TFIDF,Txt-*k*NN,Fretcit-*k*NN 在每一个数据集上的正确率(accuracy)、查准率(precision)和查全率(recall),然后针对每种算法求出对应的上述 3 项指标在 8 个数据集上面的平均值.表 2~表 4 分别对应于高频词个数为 5,7,10 时的结果.正确率、查准率、查全率的计算方式由式(3)~式(5)给出,其中设测试集中包含 P 个正例和 N 个反例,正例中包含被分类器正确分类的 P_a 个样本以及分类错误的 P_r 个样本,反例中包含被分类器误认为是正例的 N_a 个样本以及正确识别为反例的 N_r 个样本.

$$accuarcy = \frac{P_a + N_r}{P + N} \tag{3}$$

$$precision = \frac{P_a}{P_a + N_a} \tag{4}$$

$$recall = \frac{P_a}{P} \tag{5}$$

从表 2~表 4 可以看出,在特征向量中各高频词数量不同的情况下,Fretcit-*k*NN 在所有数据集上的平均正确率都明显优于其他两种非多示例学习算法,如图 4 所示.其中 Fretcit-*k*NN 在特征向量中包含 10 个高频词时性能较好,只有 12.5%的错误率,分别是 TFIDF 和 Txt-*k*NN 错误率的 61.0%((1−0.875)/(1−0.795)=0.610)和 68.3%((1−0.875)/(1−0.817)=0.683).值得注意的是,当特征向量包含高频词较少的时候(5 个高频词),Fretcit-*k*NN 也只有 16%的错误率.由此看出,Fretcit-*k*NN 在只考虑 Web 页面中少量高频词的时候仍然有效.

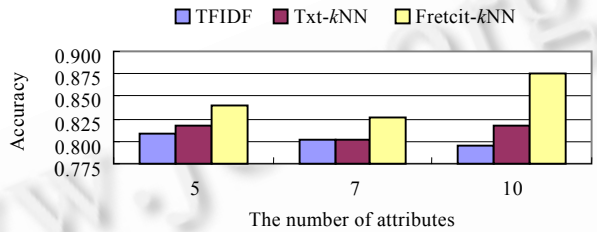

Fig.4 Comparison of accuracy under various number of attributes between TFIDF, Txt-*k*NN and Fretcit-*k*NN

图 4 TFIDF,Txt-*k*NN 和 Fretcit-*k*NN 不同属性值个数下的正确率对比

对于查全率来说,Fretcit-*k*NN 在 3 种不同的高频词个数的情况下,整体的表现也优于 TFIDF 和 Txt-*k*NN.从表 2~表 4 中的数据可以计算出 Fretcit-*k*NN 查全率的平均值为 80.9%,而 TFIDF 和 Txt-*k*NN 的 3 种情况下查全率的平均值分别只有 70.6%和 77.8%.因此 Fretcit-*k*NN 较其他两种算法能够发现更多用户感兴趣的页面.

对于查准率来说,Fretcit-*k*NN 在表 2~表 4 这 3 种不同情况下的平均值为 80.7%,优于 Txt-*k*NN 的 78.7%,但是差于 TFIDF 的 84.4%.这说明在进行 Web 目录页面推荐时,Fretcit-*k*NN 算法可能会向用户推荐比 TFIDF 更多的不感兴趣的内容.

虽然 TFIDF 的查准率比 Fretcit-*k*NN 高 3.7%,但其查全率比 Fretcit-*k*NN 低 10.3%.因此综合 3 项指标来看,采用多示例学习的 Fretcit-*k*NN 算法明显地优于 TFIDF 和 Txt-*k*NN 这两种非多示例学习算法.这表明,使用多示例学习能够较好地解决 Web 目录页面推荐问题.

5 结束语

本文使用多示例学习技术来解决中文 Web 目录页面推荐问题.本文把每个内容页面看成一个示例,包含若干指向内容页面的超链接的目录页面就可以看成包含多个示例的包.从而将中文 Web 目录页面推荐问题转化成一个多示例学习问题.本文中利用基于词频统计的中文分词技术提取高频词作为示例的属性值,并定义了频繁项最小 Hausdorff 距离,从而得到多示例学习算法 Fretcit-*k*NN.在真实数据集上的实验证明,以多示例学习方法来解决中文 Web 目录页面推荐问题是有效的.在将来的工作中,寻找一种方法来标定目录页面中那些用户真正感兴趣的超链接将更加方便用户迅速找到所需信息.此外,由于本文使用的基于 k 近邻的算法需要保存所有的访问记录,如何通过只挑选一部分重要的记录并加以保存以减少存储和计算开销,从而使其具有能够处理更大规模数据的能力也是一个十分值得研究的问题.

References:

[1] Etzioni O. The world wide web: Quagmire or gold mine. Communications of the ACM, 1996,39(11):65~68.

[2] Kosala R, Blockeel H. Web mining research: A survey. ACM SIGKDD Explorations, 2000,2(1):1~15.

[3] Dietterich TG, Lathrop RH, Lozano-Pérez T. Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence, 1997,89(1-2):31~71.

[4] Maron O. Learning from ambiguity [Ph.D. Thesis]. Cambridge: Massachusetts Institute of Technology, 1998.

[5] Maron O, Lozano-Pérez T. A framework for multiple-instance learning. In: Jordan MI, Kearns MJ, Solla SA, eds. Advances in Neural Information Processing Systems 10. Cambridge: MIT Press, 1998. 570~576.

[6] Wang J, Zucker JD. Solving the multiple-instance problem: A lazy learning approach. In: Langley P, ed. Proc. of 17th Int'l Conf. on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 2000. 1119~1125.

[7] Chevaleyre Y, Zucker JD. Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem. In: Stroulia E, Matwin S, eds. Lecture Notes in Artificial Intelligence 2056, Berlin: Springer-Verlag, 2001. 204~214.

[8] Zhou ZH, Zhang ML. Solving the multi-instance problem with neural networks. Technical Report, Nanjing: AI Laboratory, Department of Computer Science and Technology, Nanjing University, 2002.

[9] Zhou ZH, Zhang ML. Ensembles of multi-instance learners. In: Lavrac N, Gamberger D, Blockeel H, Todorovski L, eds. Lecture Notes in Artificial Intelligence 2837, Berlin: Springer-Verlag, 2003. 492~502.

[10] Long PM, Tan L. PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. Machine Learning, 1998,30(1):7~21.

[11] Auer P, Long PM, Srinivasan A. Approximating hyper-rectangles: Learning and pseudo-random sets. Journal of Computer and System Sciences, 1998,57(3):376~388.

[12] Han KS, Wang YC, Chen GL. Research on fast high-frequency strings extracting and statistics algorithm with no thesaurus. Journal of Chinese Information Processing, 2001,15(2):23~30 (in Chinese with English abstract).

[13] Jin XY, Sun ZX, Zhang FY. A domain-independent dictionary-free lexical acquisition model for Chinese document. Journal of Chinese Information Processing, 2001,15(6):33~39 (in Chinese with English abstract).

[14] Aha DW. Lazy learning: Special issue editorial. Artificial Intelligence Review, 1997,11(1-5):7~10.

[15] Edgar GA. Measure, Topology, and Fractal Geometry. Berlin: Springer-Verlag, 1990.

[16] Joachims T. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: Fisher D, ed. Proc. of the 14th Int'l Conf. on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1997. 143~151.

附中文参考文献:

[12] 韩客松,王永成,陈桂林.无词典高频字串快速提取和统计算法研究.中文信息学报,2001,15(2):23~30.

[13] 金翔宇,孙正兴,张福炎.一种非受限中文文档抽词方法.中文信息学报,2001,15(6):33~39.