ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.5, May 2006, pp.991−1000
DOI: 10.1360/jos170991

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于核矩阵学习的 XML 文档相似度量方法*

杨建武[1,2+], 陈晓鸥[1,2]

[1](北京大学 计算机研究所,北京 100871)

[2](北京大学 文字信息处理技术国家重点实验室,北京 100871)

Similarity Measures for XML Documents Based on Kernel Matrix Learning

YANG Jian-Wu[1,2+], CHEN Xiao-Ou[1,2]

[1](Institute of Computer Science and Technology, Peking University, Beijing 100871, China)

[2](National Key Laboratory for Text Processing, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-82529245, E-mail: yangjianwu@icst.pku.edu.cn, http://www.icst.pku.edu.cn

Yang JW, Chen XO. Similarity measures for XML documents based on kernel matrix learning. *Journal of Software*, 2006,17(5):991−1000. http://www.jos.org.cn/1000-9825/17/991.htm

Abstract: XML document as a new data model has been a hot research area. Similarity measure is a basic of analyses, management and text mining for XML documents. Structured Link Vector Model (SLVM) is a document model for the XML documents' similarity measure based on both the content and structure. The kernel matrix, which describes the relations between the structure units, plays an important role in the SLVM. In the paper, two algorithms are derived to learn the kernel matrix for capturing the relations between the structure units: one is based on the support vector machine and the other is based on matrix iterative analysis. For the performance evaluation, the proposed similarity measure is applied to similarity search. The experimental results show that the similarity measure based on kernel matrix learning outperform significantly the traditional measures. Furthermore, comparing with the kernel matrix leaning algorithm based on the support vector machine (SVM)'s regression, the kernel matrix leaning algorithms based on matrix iterative analysis not only acquires higher precision but also needs less training documents and cost.

Key words: XML document; similarity measure; kernel matrix learning; text mining

摘 要: XML 文档作为一种新的数据形式,成为当前的研究热点.XML 文档间相似度的计算是 XML 文档分析、管理及文本挖掘的基础.结构链接向量模型(structured link vector model,简称 SLVM)是一种综合考虑 XML 文档结构信息与内容信息进行 XML 文档相似度量的方法.体现 XML 文档结构单元关系的核矩阵在结构链接向量模型中扮演着重要角色.为自动捕获 XML 文档结构单元关系,提出了两种核矩阵的学习算法,分别是基于支持向量机(support vector machine,简称 SVM)的回归学习算法和基于矩阵迭代的学习算法.相似搜索实验对比结果表明,基于核矩阵学习方法的 XML 文档相似度量方法的准确性明显优于其他方法.进一步实验表明,基于矩阵迭代学习的核矩阵学习算法与基于支持向量机的回归学习算法相比,不仅具有更高的准确性,而且所需训练文档更少、计算代价更小.

关键词: XML 文档;相似度量;核矩阵学习;文本挖掘

* Received 2005-06-30; Accepted 2005-10-20

中图法分类号: TP181 文献标识码: A

由于具有结构化、可扩展性、跨平台性等特点,越来越多的数据标准采用 XML,如 MathML,NewsML,OWL,ebXML,cnXML 等.XML 逐渐成为信息存储与交换的主要形式.随着 XML 文档的迅速增多,如何有效地存储、管理、利用这些数据成为一个亟待解决的问题.XML 文档是文本内容信息与结构信息的综合体,XML 文档分析区别于传统的文本分析的关键在于结构信息的获取与利用.

近年来,国内外研究者对 XML 文档等半结构化数据的分析处理给予了越来越多的关注.他们有的侧重于半结构化数据的模型以及存储与查询,如 Stanford 大学的 Lore 项目[1];有的侧重于半结构化数据集成;还有的将半结构化文本与自然语言理解等技术相结合,实现对语义信息的理解.目前,在半结构化文本挖掘方面的研究成果相对较少.Yi 等人提出了一种用于半结构化文档分类的扩展向量模型.它采用嵌套定义的向量来描述文档元素,并在此模型基础上利用概率统计方法进行文档分类[2].Denoyer 等人深入研究了利用贝叶斯网络模型进行半结构化文档分类的方法[3].Zhang 等人提出了一种采用编辑距离进行 XML 文档的结构相似性计算的方法[4].Flesca 等人将结构信息看作时序关系,采用时序分析的方法进行 XML 文档的结构相似性计算[5].这些方法要么是针对特定的挖掘技术(如自动分类),缺乏可推广性和通用性;要么仅研究文档结构关系,而没有考虑作为 XML 文档主体的文本内容;有些方法虽然想法新颖、独特,但实际效果却很有限.

我们在文献[6]中提出了结构链接向量模型,从而为综合利用结构信息与内容信息进行 XML 文档分析提供了一种有效的方法.该文提出采用核矩阵描述 XML 文档中结构单元之间的关系,但并没有深入讨论获得核矩阵的方法.本文提出两种核矩阵的学习算法,分别是基于支持向量机(support vector machine,简称 SVM)的回归学习算法和基于矩阵迭代的学习算法.实验表明,这两种学习算法能够有效地捕获 XML 文档结构单元关系,显著提高相似度量的准确性.进一步实验表明,基于矩阵迭代学习的核矩阵学习方法与基于支持向量机的回归学习方法相比,不仅具有更高的准确性,而且所需训练文档更少、计算代价更小.

1 相关工作

1.1 文档模型

向量空间模型(vector space model,简称 VSM)是一种常用的文档模型.它以词语构造一个高维空间,每个词语为该空间的一个维,文档被看作这个空间中的一个向量.

$$d_x=\langle d_{x(1)},d_{x(2)},\ldots,d_{x(n)}\rangle^T \tag{1}$$

其中,n 是文档集合中不同词语的个数.

TFIDF(term frequency inverse document frequency)是向量空间模型中一种常用的文档向量化方法,它综合考虑了词语在单个文档中出现的频度和该词语在文档集合中出现的频度.

$$d_{x(i)}=TF(w_i,doc_x)\cdot IDF(w_i) \tag{2}$$

其中:$TF(w_i,doc_x)$是词 w_i 在文档 doc_x 中出现的次数;$IDF(w_i)=\log(|D|/DF(w_i))$,$|D|$是文档集合中文档总数,$DF(w_i)$是包含词语 w_i 的文档个数;$IDF(w_i)$是词语 w_i 全局特性,用来体现词语 w_i 区分文档的能力.

向量空间模型能够有效地描述文档内容,在文本检索与文本挖掘中被广泛使用.但是,向量空间模型不能描述文档中的结构信息.

XML 等半结构化文档中包含丰富的结构信息.为了既能有效地描述 XML 文档中的内容信息又能描述结构信息,我们在文献[6]中对向量空间模型进行了扩展,提出了结构链接向量模型(structured link vector model,简称 SLVM).根据结构单元中的内容,将 XML 文档中的每个结构单元看作一个向量,类似于 VSM 模型的一个文档,整个 XML 文档则被量化为一组向量,以一个矩阵来表示,从而达到将半结构化文本的结构分析与文本内容分析相结合的目的.

在 SLVM 模型中,结构链接向量由 3 部分组成:结构向量、链出向量和链入向量.在该模型中,链出向量与链

入向量是以链出目标资源和链入起始资源的结构向量进行表示的.为简化论述、突出重点,本文仅考虑 SLVM 模型中的结构向量.

在 SLVM 模型中,文档 doc_x 被表示成一个矩阵 $d_x \in R^{n\times m}$,

$$d_x=\langle d_{x(1)},d_{x(2)},\ldots,d_{x(n)}\rangle^T \tag{3}$$

$$d_{x(i)}=\langle d_{x(i,1)},d_{x(i,2)},\ldots,d_{x(i,m)}\rangle \tag{4}$$

其中:n 是文档集合中不同词语的个数;m 是 XML 文档中不同结构单元(如元素)的个数.$d_{x(i,j)}$取决于在文档 doc_x 的结构单元 e_j 中出现词语 w_i 的情况.

$$d_{x(i,j)}=TF(w_i,doc_x\cdot e_j)\cdot IDF(w_i) \tag{5}$$

其中,$TF(w_i,doc_x\cdot e_j)$为词条 w_i 在文档 doc_x 的结构单元 e_j 中出现的频度;$IDF(w_i)=\log(|D|/DF(w_i))$,$|D|$是文档集合中文档总数,$DF(w_i)$是包含词语 w_i 的文档个数;$IDF(w_i)$是词语 w_i 的全局特性,用来体现词语 w_i 区分文档的能力.

1.2 文档相似性度量

在 VSM 模型中,通常采用文档向量夹角的余弦值来度量文档间的相似性.

$$sim(doc_x,doc_y)=\cos(\langle d_x,d_y\rangle)=d_x\cdot d_y=\sum_{i=1}^{n}d_{x(i)}\cdot d_{y(i)} \tag{6}$$

其中:n 是文档集合中不同词语的个数;d_x,d_y 分别是文档 doc_x,doc_y 经过单位化后的向量,即

$$\sum_{i=1}^{n}d_{x(i)}^2=1,\ \sum_{i=1}^{n}d_{y(i)}^2=1.$$

与 VSM 模型类似,在 SLVM 模型中采用如下方式度量两个 XML 文档间的相似性:

$$sim(doc_x,doc_y)=\sum_{i=1}^{n}d_{x(i)}^T\cdot M\cdot d_{y_{(i)}} \tag{7}$$

其中:n 是文档集合中不同词语的个数;d_x,d_y 分别是文档 doc_x,doc_y 在 SLVM 模型空间中的矩阵.为消除各文档结构单元中所含词语数量的差别,对矩阵中各列向量(同一结构单元的向量)进行单位化,即 $\sum_{k=1}^{n}d_{x(i,k)}^2=1$.$M$ 是一个 $m\times m$ 的矩阵,用来描述文档结构单元之间的相关性,以及它们对文档相似性度量的贡献程度.本文称其为核矩阵.

由于结构单元之间存在相关性,所以,SLVM 模型所构造的特征空间并非正交空间.举例来说,“计算机”这个词出现在 3 个文档中,分别在文档 A 和文档 B 的“标题”中以及文档 C 的“摘要”中.直观地看,在仅考虑“计算机”这个词的情况下,虽然该词出现在文档 A 和文档 C 的不同结构位置(元素)中,但它对文档 A、文档 C 间的相似度有一定的贡献.这种贡献要小于它对文档 A、文档 B 间相似度的贡献.文献[6]采用矩阵来描述结构单元之间的这种关系,但没有研究获得这种关系的具体方法.本文在数学变换的基础上提出通过学习算法来分析获取这种关系,并具体给出了基于支持向量机回归的学习算法和矩阵迭代的学习算法.这种 XML 结构分析方法不仅对 SLVM 模型是有意义的,而且对其他有关 XML 文档分析处理的研究具有通用性.

2 核矩阵学习方法分析结构单元间关系

如公式(7)所示,在 SLVM 模型中,核矩阵 M 在 XML 文档相似度量中扮演着重要角色,核矩阵的取值直接影响着 XML 文档间相似度的计算结果.但当我们在文献[6]中提出结构链接向量模型时,并没有深入讨论获得核矩阵的方法.

在 SLVM 模型中,核矩阵表示的是结构单元之间的关系及其对计算文档间相似度的贡献权重.结构单元之间的关系可以通过直接计算文档结构间相似性的方法进行计算,如 Zhang 等人在文献[4]中采用编辑距离进行 XML 文档结构相似度的计算.其基本想法是,将文档结构抽象为一个树结构,不同文档结构之间的相似性通过编辑距离方法计算对应树结构之间的相似性来获得.这种方法对一些文档相似度的计算是有效的,但它以一种固定的模式计算结构之间的关系,并不能真正代表结构单元之间的语义关系,不能随着文档结构关系的组织形

式的变化而变化.如果以不同的方式来组织相同的结构关系,则计算结果也可能会有很大的差别,其适用性受到很大限制.通过学习的方法进行这种关系的训练学习,可避免这种方法的限制.

2.1 基于支持向量机的回归学习方法

对公式(7)进行如下变换:

$$sim(doc_x,doc_y)=\sum_{i=1}^{n}d_{x(i)}^{T}\cdot M\cdot d_{y_{(i)}}=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m}(d_{x(i,j)}M_{(j,k)}d_{y(i,k)})=\sum_{j=1}^{m}\sum_{k=1}^{m}\left(M_{(j,k)}\cdot\sum_{i=1}^{n}d_{x(i,j)}d_{y(i,k)}\right) \tag{8}$$

记

$$f(doc_x,doc_y,j,k)=\sum_{i=1}^{n}d_{x(i,j)}d_{y(i,k)} \tag{9}$$

则有

$$sim(doc_x,doc_y)=\sum_{j=1}^{m}\sum_{k=1}^{m}(M_{(j,k)}\cdot f(doc_x,doc_y,j,k)) \tag{10}$$

函数 $f(doc_x,doc_y,j,k)$的意义是文档 doc_x 的第 j 个结构单元的内容(向量)与文档 doc_y 的第 k 个结构单元的内容(向量)的相似度.

更广义地来说,在保持函数 $f(doc_x,doc_y,j,k)$意义的前提下,该函数既可以是公式(6)所示的余弦算法,也可以是其他度量两个纯文本之间相似度的方法,如潜语义分析算法(latent semantic analysis,简称 LSA)[7]等.

分析公式(10)可以认为,这是一个线性回归问题,矩阵 M 的每个元素可以看作一个待学习的变量.具体来说,对给定的一组 XML 文档,人工标注文档两两之间是否相似.学习的目标则是求解如下优化问题:

$$\min\left\{\sum_{x=1}^{r}\sum_{y=1}^{r}\left(sim(doc_x,doc_y)-\left(\sum_{j=1}^{m}\sum_{k=1}^{m}(M_{(j,k)}\cdot f(doc_x,doc_y,j,k))\right)\right)\right\} \tag{11}$$

上式可看作

$$\min\left\{\sum_{l=1}^{r\times r}\left(y_l-\left(\sum_{u=1}^{m\times m}(w_{(u)}\cdot x_l(u))\right)\right)\right\} \tag{12}$$

函数式(12)可采用支持向量机的回归学习方法求解并计算 $w_{(u)}$,$w_{(u)}$作为矩阵 M 的元素,即可获得核矩阵 M.

2.2 基于矩阵迭代的学习方法

2.2.1 依赖性假设

在现实问题中,很多情况下所面临的对象空间是非正交的空间,非正交空间中各维之间存在一定的相关性.这种关系是分析该空间对象之间关系的基础,正被不同领域的研究者所关注.一种比较有效的方法是:假设对象实体本身之间的相似性与对象特征之间的相似性是互相依赖的[8,9].其基本形式是

$$S_o=B^TS_fB \text{ and } S_f=BS_oB^T \tag{13}$$

其中:B 是一组对象向量组成的矩阵;S_o 两两对象之间相似度形成的矩阵;S_f 是两两特征之间相似度形成的矩阵.

为避免矩阵求逆所带来的巨大计算量,可采用迭代的方法对上式求解特征间相似矩阵 S_f.在文献[9]中,采用如下递归形式进行特征间相似矩阵 S_f 的求解:

$$S_o^{k+1}=\lambda_1B^TS_f^kB+L_1^k \tag{14}$$

$$S_f^{k+1}=\lambda_2BS_o^kB^T+L_2^k \tag{15}$$

其中,λ_1 和 λ_2 分别是满足 $\lambda_1\le 1/\|B\|_\infty$ 和 $\lambda_2\le 1/\|B\|_1$ 的两个正实数常量,

$$L_1^k=I-diag(\lambda_1B^TS_f^kB),\quad L_2^k=I-diag(\lambda_2BS_o^kB^T).$$

2.2.2 SLVM 中矩阵迭代算法

根据依赖性假设,SLVM 模型中有式(16)、式(17)成立.

$$S=\sum_{i=1}^{n}B_{(i)}^{T}\cdot M\cdot B_{(i)} \tag{16}$$

$$M=\sum_{i=1}^{n}B_{(i)}\cdot S\cdot B_{(i)}^{T} \tag{17}$$

其中:S 是一组 XML 文档中两两文档之间相似度形成的矩阵;$B_{(i)}$是一组 XML 文档关于第 i 个词语在各结构单元上的向量所组成的矩阵;M 是两两结构特征之间相似度矩阵,即核矩阵.

对式(16)、式(17)进行变换:

$$S_{(j,k)}=\sum_{i=1}^{n}\sum_{u=1}^{m}\sum_{v=1}^{m}(B_{(i)(u,j)}M_{(u,v)}B_{(i)(v,k)})=\sum_{u=1}^{m}\sum_{v=1}^{m}\left(M_{(u,v)}\cdot\sum_{i=1}^{n}(B_{(i)(u,j)}B_{(i)(v,k)})\right) \tag{18}$$

$$M_{(u,v)}=\sum_{i=1}^{n}\sum_{j=1}^{r}\sum_{k=1}^{r}(B_{(i)(u,j)}S_{(j,k)}B_{(i)(v,k)})=\sum_{j=1}^{r}\sum_{k=1}^{r}\left(S_{(j,k)}\cdot\sum_{i=1}^{n}(B_{(i)(u,j)}B_{(i)(v,k)})\right) \tag{19}$$

记

$$f(j,k,u,v)=\sum_{i=1}^{n}B(i)(u,j)B(i)(v,k) \tag{20}$$

则有

$$S_{(j,k)}=\sum_{u=1}^{m}\sum_{v=1}^{m}(M_{(u,v)}\cdot f(j,k,u,v)) \tag{21}$$

$$M_{(u,v)}=\sum_{j=1}^{r}\sum_{k=1}^{r}(S_{(j,k)}\cdot f(j,k,u,v)) \tag{22}$$

与式(9)类似,函数 $f(j,k,u,v)$的意义是文档 j 第 u 个结构单元的内容(向量)与文档 k 第 v 个结构单元的内容(向量)的相似度.在保持函数 $f(j,k,u,v)$意义的前提下,该函数可以是度量纯文本之间相似度的其他任何方法.

采用如下迭代形式进行核矩阵(结构单元相关矩阵)的求解:

$$S_{(j,k)}^{g+1}=\begin{cases}1, & \text{if } j==k\\ \lambda\cdot\sum_{u=1}^{m}\sum_{v=1}^{m}(M_{(u,v)}^{g}\cdot f(j,k,u,v)), & \text{if } j\langle\rangle k\end{cases} \tag{23}$$

$$M_{(u,v)}^{g+1}=\lambda\cdot\sum_{j=1}^{r}\sum_{k=1}^{r}(S_{(j,k)}^{g}\cdot f(j,k,u,v)) \tag{24}$$

其中

$$\lambda\le 1/\max\left\{\max_{j,k}\left\{\sum_{u=1}^{m}\sum_{v=1}^{m}f(j,k,u,v)\right\},\max_{u,v}\left\{\sum_{j=1}^{r}\sum_{k=1}^{r}f(j,k,u,v)\right\}\right\} \tag{25}$$

如果 S 的初始值为对训练集的人工标注值,本文称其为有指导的学习.

$$S^{0}=S^{H} \tag{26}$$

其中 S^{H} 为对训练集合中两两文档之间的人工标注值所组成的矩阵.

如果不对训练集进行标注,则初始时认为各文档只与其自身相似而与其他文档都不相似,即

$$S_{(j,k)}^{0}=\begin{cases}1, & \text{if } j==k\\ 0, & \text{if } j\langle\rangle k\end{cases} \tag{27}$$

这种情况下,本文称其为无指导的学习.

显然,在λ满足式(25)的条件下,迭代计算过程中矩阵 M 和 S 的任何元素的取值都在 0,1 之间,并可以证明该迭代计算过程是收敛的.由于篇幅所限,证明过程略.

3 实 验

3.1 数据集与实验设计

实验使用 Intel P4 2.0GHz CPU,512M 内存的个人计算机,在 Windows 2000 Server 操作系统上,以 Visual C++ 5.0 作为开发环境.

分别选用英文的 ACMSIGMOD[10]和中文的 CEDB[11]两个 XML 数据集进行实验分析.ACMSIGMOD 数据集是由数百篇 XML 格式的 ACMSIGMOD 论文组成的,其文档结构如图 1 所示.排除那些没有分类信息的文档后,实验中使用了 461 个 XML 文档.CEDB 是中国百科术语数据库的部分 XML 文档.该数据库将数百万的百科术语词条采用 XML 格式进行描述,是中国最早采用 SGML/XML 的大型国家项目.实验随机选用了其中 960 个文档.实验中使用的 4 组数据见表 1.

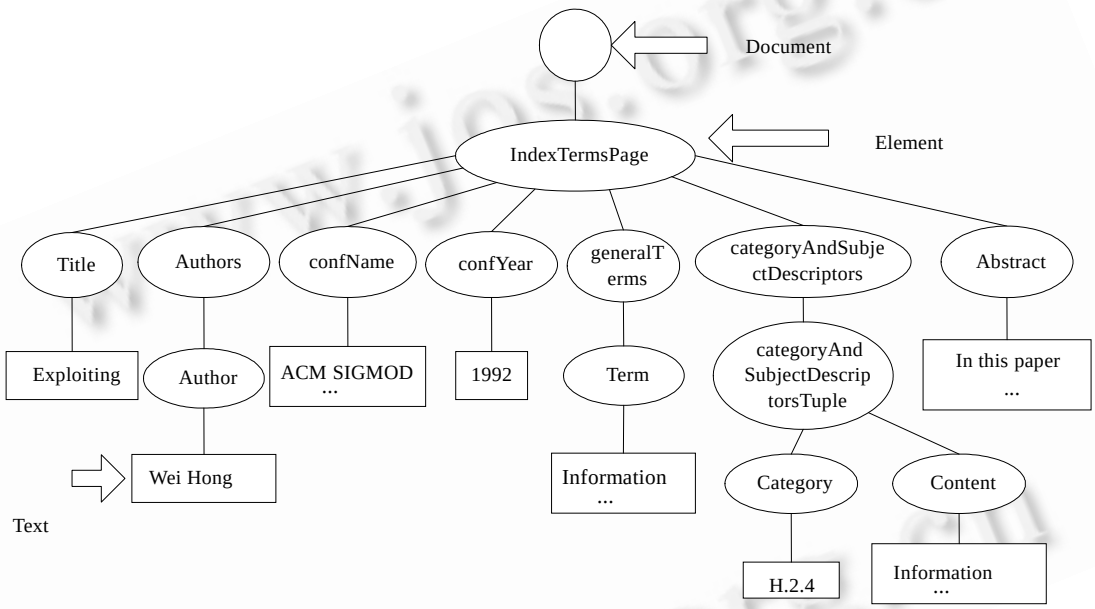

Fig.1 The DOM tree of an XML document extracted from ACMSIGMOD dataset

图 1 ACMSIGMOD 数据集合中的一个 XML 文档的 DOM 树

Table 1 Data subsets used in our experiments

表 1 实验中使用的数据子集

Datasets	Sources	Total num. of documents
ACM-1	ACM SIGMOD	96
ACM-2	ACM SIGMOD	461
CEDB-1	CEDB	320
CEDB-2	CEDB	960

实验采用 KNN(*K*-nearest neighbors)相似搜索进行实验评价.

这两个数据集都有分类信息:在 CEDB 中,每个文档的分类是唯一的;ACMSIGMOD 文档中,每个文档通常是属于多个分类的.我们认为,属于同一个分类的 XML 文档具有较强的相似性.在实验中,利用分类信息来替代对 XML 文档两两间相似度的人工标注.

具体来说,我们采用式(28)替代人工对 XML 文档 i 和 j 之间的相似度进行标注.

$$sim(i,j)=\frac{2\cdot|C_i\cap C_j|}{|C_i|+|C_j|} \tag{28}$$

其中,C_i 和 C_j 分别是文档 i 和 j 所属类别的集合.

在实验中,利用该相似度作为人工标注的相似度来进行核矩阵的学习及 KNN 搜索结果的评价.

在实验中,我们随机抽取部分文档作为训练文档,把其他文档作为测试文档.通过对训练文档的学习获得核矩阵(文档的结构单元之间的关系).利用该核矩阵,以每个文档为测试文档,分别找到与之最相似的 k 个(k 分别取 10,20,30 等)文档,并且通过判断这些文档与人工标注的最相似的 k 个文档进行比较来评价相似搜索的准确性.具体来说,我们采用如下公式进行结果的评价:

$$p(k)=\frac{1}{r}\sum_{i=1}^{r}\frac{|Q_{(i,k)}\cap R_{(i,k)}|}{k} \tag{29}$$

其中:k 是指定搜索最接近的文档个数;r 是集合中文档总数;$Q_{(i,k)}$是基于相似度计算方法所获得的与第 i 个文档最相似的 k 个文档;$R_{(i,k)}$是基于人工标注信息所获得的与第 i 个文档最相似的 k 个文档.

在实验中,我们分别采用如下 4 种方法进行对比分析:

- 传统的向量空间模型 TFIDF(不考虑结构信息),本文以 TFIDF 对其标识;
- 采用编辑距离计算核矩阵的方法,本文以 SLVM-Edit 对其标识;
- 采用支持向量机的回归学习方法进行核矩阵学习的方法,本文以 SLVM-SVM 对其标识;
- 采用矩阵迭代学习进行核矩阵学习的方法,本文以 SLVM-KM 对其标识,并分别用 SLVM-KM-Supervised 和 SLVM-KM-UnSupervised 区分有指导学习和无指导学习.

结构单元的选择方法也是一个重要的研究课题.由于不是本文的研究重点,本实验中采用比较简单的方式选择结构单元,即以结构树路径作为结构单元,又由于实验数据集的每个元素在文档集合中只有唯一的路径,所以,实验中我们以元素名称来标识结构单元.

另外,实验中 $\lambda=0.9/\max\left\{\max_{j,k}\left\{\sum_{u=1}^{m}\sum_{v=1}^{m}f(j,k,u,v)\right\},\max_{u,v}\left\{\sum_{j=1}^{r}\sum_{k=1}^{r}f(j,k,u,v)\right\}\right\}$.

训练文档是在文档集合中随机抽取指定数量的文档作为训练集合,并重复随机选取 5 次,采用各次结果的平均值.

3.2 实验结果分析

我们对表 1 中的数据集做对比实验.实验中分别选择 20%的文档作为训练文档,实验结果如图 2~图 5 所示.

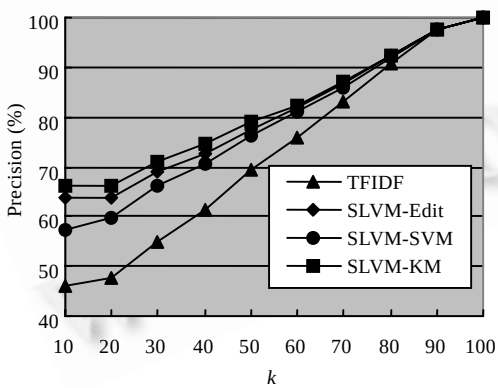

Fig.2 Precision of similarity queries on ACM-1

图 2 在 ACM-1 数据集上相似检索的准确度

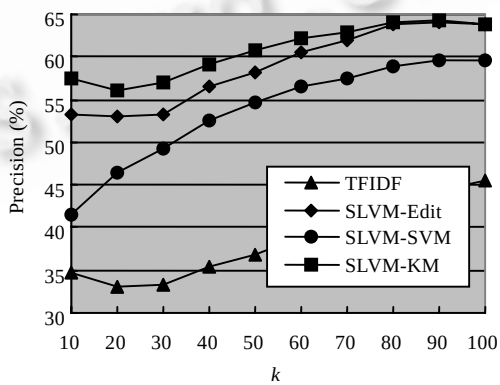

Fig.3 Precision of similarity queries on ACM-2

图 3 在 ACM-2 数据集上相似检索的准确度

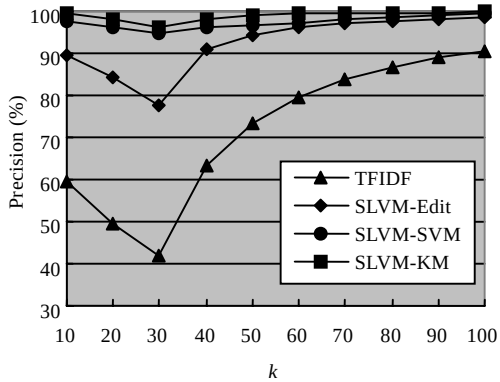

Fig.4 Precision of similarity queries on CEDB-1

图 4 在 CEDB-1 数据集上相似检索的准确度

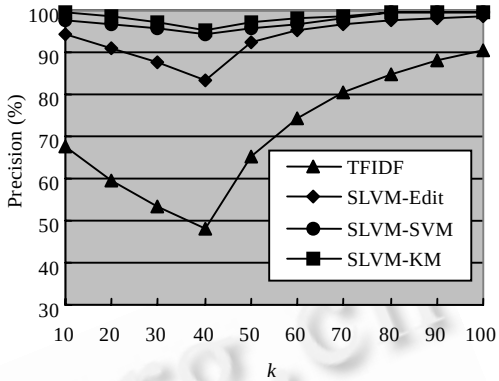

Fig.5 Precision of similarity queries on CEDB-2

图 5 在 CEDB-2 数据集上相似检索的准确度

实验结果数据表明,相对于基于传统的向量空间模型的方法,基于 SLVM 的方法由于利用了文档的结构信息,实验结果准确性明显提高,提高量普遍在 10%~30%.同时,实验结果也表明,基于编辑距离的固定计算方法在不同数据集上的表现有较大差异:在 ACM 数据集上,其效果明显介于两种基于学习的方法之间;而在 CEDB 数据集上,其效果明显比两种基于学习的方法要差(约 5%).其原因在于,基于编辑距离的方法是以一种固定的方式描述结构单元之间的相互关系;而基于学习的方法能根据不同数据集的结构单元之间的实际语义关系动态捕获结构单元之间的相互关系.同样是学习算法,基于矩阵迭代的方法相对于基于 SVM 回归学习的方法在两种数据集上均有更高的准确性(高 2%~10%).

在实验中我们发现,训练文档数量对学习效果的影响较小.在 4 个数据集上分别用 10~100 个文档作为训练文档进行实验对比,结果表明:随着训练文档数的增加,SLVM-KM 方法的准确率基本上保持不变;而 SLVM-SVM 方法准确率略有提高.在 CEDB-1 数据集上,SLVM-KM 方法中训练文档数在 20 个左右就可以获得较好的效果;SLVM-SVM 方法中训练文档数在 40 个左右才可以获得较好的效果.另外,对于 SLVM-KM 方法,有指导的学习比无指导的学习具有更好的效果,但差别不是很大.由于篇幅所限,本文只给出 CEDB-1 数据集上(K=30)的测试结果,如图 6 所示.其他数据集上的实验结果类似.

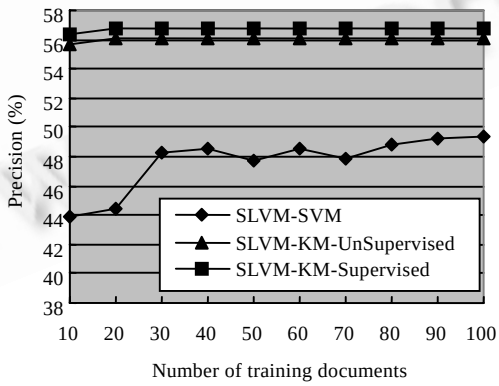

Fig.6 Precision of similarity queries on CEDB-1 with different number of training documents

图 6 不同数量训练文档条件下,CEDB-1 数据集上相似检索的准确度

为进行时间开销的评价,在 4 个数据集上分别用 20%,30%和 50%的文档作为训练文档进行实验对比.实验结果见表 2,其所列时间是一轮相似搜索评测的总时间开销.结果表明,SLVM-SVM 方法明显慢于 SLVM-KM 方法,其主要原因在于,SLVM-SVM 中的 SVM 回归学习的计算代价要大.

Table 2 The time cost in different dataset
表 2 不同数据集评测的时间开销

	SLVM-KM-Unsupervised (s)	SLVM-KM-Supervised (s)	SLVM-SVM (s)
ACM-1	15	13	39
ACM-2	191	167	1 022
CEDB-1	69	71	642
CEDB-2	1 825	1 856	N/A*

注:在 CEDB-2 数据集上进行 SLVM-SVM 方法计算时,若选择 50%的文档作为训练集进行学习,由于所需内存太大,计算中需要进行大量的虚拟内存与物理内存的交换,所以总耗时非常大.

作为本文方法的一个副产品,核矩阵提供了结构单元(如元素)之间的语义关系.如图 7 所示是 ACM SIGMOD 数据集上矩阵迭代学习获得的核矩阵,其中,X 轴和 Y 轴的数值是元素的索引号(见表 3).结合表 3 可知,元素对{term,content},{abstract,term},{term, content},{title,term},{title,content}相对于其他元素对之间有更大的相关性,对文档相似度计算具有更大的贡献.同时可以看出,元素 abstract,authors,和 categories' content 等具有较高的重要性,而 conference year,initial page 等具有较低的重要性.这些都是与我们的直观认识相一致的.

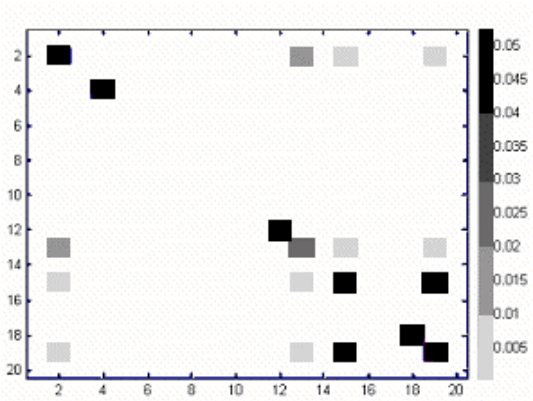

Fig.7 The kernel matrix learned using the ACMSIGMOD dataset
图 7 ACMSIGMOD 数据集学习获得的核矩阵

Table 3 Index to elements of the ACMSIGMOD dataset (see Fig.1)
表 3 ACMSIGMOD 数据集中元素索引号(参见图 1)

No.	Element Name	No.	Element Name
1	IndexTermsPage	11	fullTextURL
2	**title**	**12**	**Size**
3	authors	**13**	**abstract**
4	**author**	14	generalTerms
5	confName	**15**	**Term**
6	confYear	16	categoryAndSubjectDescriptors
7	volume	17	categoryAndSubjectDescriptorsTuple
8	number	**18**	**category**
9	initPage	**19**	**content**
10	endPage		

注:表 3 中的字体加粗项是在图 7 中具有高权重的元素.

4 结束语

本文在我们先前的研究成果 SLVM 模型的基础上,为自动捕获 XML 文档结构单元关系提出了两种核矩阵的学习算法.KNN 相似搜索实验对比结果表明:基于 SLVM 的方法由于利用了文档的结构信息,相对于基于传统的向量空间模型的方法,其实验结果的准确性普遍提高 10%~30%.由于基于编辑距离的方法是以一种固定的方式描述结构单元之间的相互关系,它在不同数据集上的表现有较大差异;而基于学习的方法能够根据不同数据

集的结构单元之间的实际语义关系动态地捕获结构单元之间的相互关系.同样是学习算法,基于矩阵迭代的方法相对于基于 SVM 回归学习的方法在两种数据集上均有更高的准确性(高 2%~10%),而且所需训练文档更少、计算代价更小.本文对 XML 结构分析的方法不仅对 SLVM 模型有重要意义,而且该方法可被用于其他有关 XML 文档分析处理的研究之中.

References:

[1] http://www-db.stanford.edu/lore/home/

[2] Yi J, Sundaresan N. A classifier for semi-structured documents. In: Ramakrishnan R, Stolfo S, Pregibon D, eds. Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2000). New York: ACM Press, 2000. 340−344.

[3] Denoyer L, Gallinari P. Bayesian network model for semi-structured document classification. Information Processing and Management, 2004,40(5):807−827.

[4] Zhang ZP, Li R, Cao SL, Zhu YY. Similarity metric for XML documents. In: Ralph B, Martin S, eds. Proc. of the 2003 Workshop on Knowledge and Experience Management (FGWM 2003). Karlsruhe, 2003. 255−261. http://km.aifb.uni-karlsruhe.de/ws/LLWA/fgwm/Resources/FGWM03_13_Zhongping_Zhang.pdf

[5] Flesca S, Manco G, Masciari E, Pontieri L, Pugliese A. Detecting structural similarities between XML documents. In: Fernandez MF, Papakonstantinou Y, eds. Proc. of the Int'l Workshop on the Web and Databases (WebDB). 2002. 55−60. http://www.db.ucsd.edu/webdb2002/papers/19.pdf

[6] Yang JW, Chen XO. A semi-structured document model for text mining. Journal of Computer Science and Technology, 2002,17(5): 603−610.

[7] Deerwester S, Dumais ST, Landauer TK, Furnas GW, Harshman RA. Indexing by latent semantic analysis. Journal of the Society for Information Science, 1990,41(6):391−407.

[8] Kandola J, Shawe-Taylor J, Cristianini N. Learning semantic similarity. In: Becker S, Thrun S, Obermayer K, eds. Proc. of the Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2002. 657−664.

[9] Liu N, Zhang BY, Yan J, Yang Q, Yan SC, Chen Z, Ma WY. Learning similarity measures in the non-orthogonal space. In: Grossman D, Gravano L, Zhai C, Herzog O, Evans D, eds. Proc. of the 13th Conf. on Information and Knowledge Management (CIKM 2004). New York: ACM Press, 2004. 334−341.

[10] 2001. http://www.acm.org/sigs/sigmod/record/xml/XMLSigmodRecordMarch1999.zip

[11] http://www.cndbk.com.cn/

杨建武(1973－),男,江西南城人,博士,副研究员,主要研究领域为文本挖掘与信息检索,SGML/XML.

陈晓鸥(1960－),男,研究员,主要研究领域为内容管理与知识管理,XML 数据交换与表现.