基于余弦相似度的文本空间索引方法研究*)

张振亚[1,2] 王 进[2] 程红梅[3] 王煦法[2]

(中国科学技术大学电子工程与信息科学系 合肥 230027)[1]

(中国科学技术大学计算机系 合肥 230027)[2] (安徽师范大学数学系 芜湖 241000)[3]

摘 要 基于相似度的数据空间索引在数据挖掘及数据可视化等方面有着重要的应用。本文以新闻的标题为研究对象,提出了以 CrossAVL 为基础的文本对象层次式聚类方法以及文本信息空间索引算法 FastMap-MDS,有效地保持了文本对象间的相似信息。实验表明,该方法具有较高的效率和精度。

关键词 相似度,空间索引,层次式聚类

An Approach for Spatial Index of Text Information Based on Cosine Similarity

ZHANG Zheng-Ya[1,2] WANG Jin[2] CHENG Hong-Mei[3] WANG Xu-Fa[2]

(Department of Electronical Engineering and Information Science, University of Sci. and Tech. of China, Hefei 230027)[1]

(Department of Computer Science, University of Sci. and Tech. of China, Hefei 230027)[2]

(Mathematics Department of Anhui Normal University, Wuhu 241000)[3]

Abstract Spatial index for data based on similarity can be employed by applications on data mining and data visualization widely. To build spatial index of news title, this paper implements hierarchical cluster algorithm for news titles with CrossAVL as data structure for the similarity matrix storing and presents an available and efficiency method named as FastMap-MDS. Experiment results show that this method can work efficiently while the similarity information are kept well.

Keywords Similarity, Spatial index, Hierarchical cluster

1 概述

数据对象的特征抽取/索引构造是信息检索系统必须进行的工作:复杂对象的检索需通过检索其特征/索引进行。文本对象的索引通常以文本中所含的字词为基础,这与对文本对象的检索时,需要用户提供检索文本的关键词要求一致。通过特定关键词的查询,信息检索系统推荐给用户特定的文本信息后,用户经常需要进一步查询"与特定的某条推荐信息内容接近"的信息(条件信息)。在基于关键词的文本信息检索框架下,这种查询请求可以通过由用户提供更多更精确的查询词来解决,但这种方法加重了用户的负担且检索精度依赖于用户新查询词的构造。相似度计算是另一种可行的方法:系统在用户提交条件信息后,计算文本集中各信息与条件信息的相似度,把相似度大的信息推荐给用户。这种方式,需要用户足够耐心。作为修正,信息检索系统可以静态地保存文本集中文本数据的相似度信息。这种修正方式的空间复杂度是 $O(n^2)$。对文本集较大的信息系统来说,即使拥有足够的存储设备,从 $O(n^2)$ 的存储信息中 k-近邻查询,效率低下。

为文本信息建立空间索引是快速实现这种查询请求的可行方法。数据的空间索引,是指把数据集映射到 k 维欧氏空间后,与数据对应的 k 维点的坐标。称建立数据的空间索引的过程或方法为空间索引映射或索引映射。基于数据之间距离的空间索引,是指利用索引映射获得数据的空间索引时尽可能地保持数据间的距离信息。构造数据的基于距离的空间索引,是数据挖掘及可视化应用中的一种重要的数据预处理手段。利用空间逼近法(SAMS, Spatial access methods),数据的空间索引被典型地应用于基于例子的查询、最佳匹配、最近邻查询等操作[2,3]。

已知数据间的距离信息,构造数据的空间索引的直观的解决方案是:首先将各数据的空间索引用 k 维欧氏空间中的随机点表达;然后通过反复调整诸点的相对位置,尽可能多地保持数据间的距离信息。"数据间距离信息被尽可能地保持"由偏差函数 *stress*[4,5] $stress=\sqrt{\frac{\sum_{i,j}(d'_{ij}-d_{ij})^2}{\sum_{i,j}d_{ij}^2}}$ 刻画,其中,d'_{ij} 为第 i、j 个数据的空间索引表示的点之间的距离,d_{ij} 为第 i、j 个数据之间的距离。所谓尽可能多地保留距离信息,是指 *stress* 取值尽可能地小。*MDS*[5] 是这类方法的典型代表。以 *stress*[1~7] 为标准,MDS 具有最小 *stress* 值[7]。由于对规模为 n 的数据集,优化的 MDS 算法的时间复杂度为 $O(n^2)$,MDS 仅适用于 n 较小时。

FastMap[2] 通过欧氏空间中勾股定理和随机选择策略的运用,对规模为 n 的数据集,时间复杂度为 $O(n)$。与 MDS 方法相比,其 *stress* 较大,特别是 k 比较小时。

MDS-NN 方法[1] 试图通过从原始数据中选取少量代表数据,对代表数据使用 MDS 建立索引,再以代表数据及其索引训练 BP 神经网络;训练完成的 BP 网络用于计算数据的索引。这种方法,对代表数据的选取有着严格的要求:一是由于 MDS 以及训练 BP 网络的高时间复杂度原因,代表数据的数量要尽量地少,二是由于代表数据的索引是 BP 网络中的吸引子,要求代表数据与被代表数据要有高度的相似性。这两个要求,对较大规模的随机数据集,几乎不能被满足。

本文以新闻的文本标题为研究对象,通过层次式聚类对文本数据进行预处理,利用 FastMap 和 MDS 高效地为文本

*)中国博士后科学基金资助(2004036463)。张振亚 博士后,主要研究领域为信息检索、数据挖掘,机会/征兆发现;王 进 博士生,主要研究领域为信息检索、数据挖掘与半结构化数据;王煦法 教授,博导,主要研究领域为人工智能、进化计算。

数据建立空间索引，有效地保持了文本数据的余弦相似度信息。本文第 2 节介绍了十字索引平衡二叉树(Cross AVL)数据结构以及层次式聚类的合并策略；第 3 节介绍了 FastMap 与 MDS 联合构造数据空间索引的方案；第 4 节给出了相关的实验结果；第 5 节对未来的研究进行了展望。

2 层次式聚类算法的实现

文本对象 T 可形式地表达为 TF 向量 $TV=(tv_1, tv_2, \cdots, tv_n)$，其中 tv_i 的取值为字典中第 i 个词出现的频率，$i=1\cdots n$，n 为字典收录字词的数量。设 TV_1、TV_2 是文本对象 T_1、T_2 的 TF 向量表示，则 $cosine(TV_1, TV_2)=\frac{\sum_{i=1}^{n} tv_{1i}\times tv_{2i}}{\sqrt{\sum_{i=1}^{n} tv_{1i}^2}\sqrt{\sum_{i=1}^{n} tv_{2i}^2}}$ 表示了 T_1、T_2 的相似程度，称 $cosine(TV_1, TV_2)$ 为 T_1、T_2 的余弦相似度(相似度)。文本类间的相似度可类似计算。

以文本对象间的余弦相似度为度量，采用层次式聚类方法[6]，是对文本集进行聚类处理的通用方法之一。在类过程中，以文本对象类之间的相似度作为合并操作的依据。相似度的阈值定义为 *SimBound*，即若相似度不小于该阈值，类的合并操作可以执行。

文本对象间的余弦相似度，可以组织成一个实对称矩阵。一种有效的存储方式是不考虑文本对象自己之间的相似性(常数，1)，只存储该矩阵的上三角部分。设原始文本对象的数量为 n，则被存储的有效的相似度信息的个数为 $n(n-1)/2$。如果线性地组织这些信息(不排序)，从中查找最大的相似度的操作的时间复杂度为 $O(n^2)$。为降低该时间复杂度，可以采用树形的存储结构。若以平衡二叉树(AVL Tree)为存储的数据结构，将查找最大的相似度的操作的时间复杂度降低为 $O(\log(n))$。

一次合并操作完成后，需要重新计算、存储所有发生变化的相似度信息。一次合并操作合并两个类，合并前的相似度信息，只要与被合并的两个类无关就不发生变化；同时，合并操作产生一个新的类，需要计算、保存其它类与该类的相似度。

为只处理发生变化的相似度信息，更新操作在 AVL 树中进行。为快速定位需要删除的 AVL 节点，每个类有水平、垂直两个双向索引链表。一个类的水平索引链表记录表示该类与其它类相似度值的 AVL 树的节点；垂直索引链表记录表示其它类与该类相似度值的 AVL 树的节点。这两种链表是对类之间相似度矩阵的行、列值的十字链表的表示。称这种表示类间相似度矩阵的方法为十字索引平衡二叉树(Cross AVL 树，简记 CAVL)。在这种方式下，相关的层次式聚类算法的形式描述如算法 1。其中，为有效地使用 MDS 为同一类的文本对象建立空间索引，限定同一类中的文本对象的数量不超过阈值 *InnerBound*，即若被合并的两个类中，如果某个类中的文本对象的数大于 *InnerBound* 或者合并操作产生的新类中的文本对象数大于 *InnerBound*，则合并操作不被执行。

算法 1：基于 Cross—AVL 数据结构的层次式聚类算法

输入：相似度的阈值 *SimBound*，类内文本对象数量阈值 *InnerBound*、文本对象的 TF 表示
输出：对文本对象的聚类结果
1. 初始化各类的描述子以及 CrossAVL 下的相似度矩阵；
2. Stop=FALSE;
3. While not Stop
4. 在 CrossAVL 化的相似度矩阵中查找具有最大相似度值的节点 Node;
5. If Node 中的最大相似度值小于 SimBound then Stop=TRUE;
6. Else
7. If Node 指示的两个类各自包含文本对象的数量或者之和大 InnerBound
Then Node 赋值为 Node 的直接前驱，转 5;
8. 设需要对第 i、j 类进行合并，构造新类的类别信息以及与其它类的相似度；
9. 在 CrossAVL 树中，从第 i 类的水平、垂直索引表获得需要被删除的树的节点列表，删除；
10. 在 CrossAVL 树中，从第 j 类的水平、垂直索引表获得需要被删除的树的节点列表，删除；
11. 构造新类的水平、垂直索引表；
12. 构造新类与其它类的相似度节点，将新节点插入树以及相关类的水平、垂直索引表；
13. End if
14. End while

算法 1 的特色在于文本对象类之间的相似度矩阵的组织。CrossAVL 结构的运用，使得 4 中一次插入两个类的相似度信息操作的时间复杂度为 $O(\log(n))$，一次删除两个类的相似度信息操作的时间复杂度为 $O(\log(n))$，近而一次 9、10 合并操作的时间复杂度为 $O(n\log(n))$。同时，阈值 *InnerBound* 的引入，使得某一类中文本对象的数量不至于太多，尽可能地避免了同一类文本有多个近似但有差异主题的现象的发生。

3 空间索引的 FastMap-MDS 的方法

虽然 MDS 方法建立文本对象的空间索引能保持很高的精度，但 MDS 的时间复杂度限制了可处理文本对象的数量。为了利用 MDS 的高精度特点，可在对文本对象进行聚类处理后，同类内的数据采用 MDS 方法建立空间索引。某类包含的文本对象的空间索引以该类的空间索引为几何中心。对大量的随机文本对象，经过高 SimBound、低 InnterBound 的约束完成聚类处理后，类别繁多。如何高效地将这些类别映射为目标空间中的点，成为快速建立文本对象空间索引的关键。研究中使用 FastMap 方法线性地确定各文本类的空间索引。

FastMap 和 MDS 都是根据数据对象间的距离建立数据的空间索引，而文本的余弦相似度，是文本的内容相似程度的度量。为构造文本的空间索引，需要定义文本对象间的距离。文本对象的距离，必须满足非负性、对称性和三角不等式这三个距离定义的充要条件。由于余弦相似度表达的是两个文本的 TF 向量的夹角的余弦值，若文本对象的 TF 向量被模 1 处理，可按照命题 1 规定两个文本对象的距离。利用 FastMap、MDS 建立文本对象空间索引的算法 FastMap-MDS 如算法 2。

命题 1：P、Q 为模为 1 的文本对象的 TF 向量，其余弦相似度为 $\cos\theta$，则 PQ 的距离为 $\sqrt{2(1-\cos\theta)}$。

证明：结合图 1 的示意，命题显然成立。

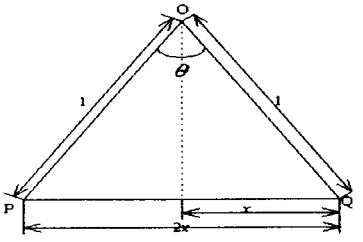

图 1 P、Q 的距离计算

算法 2：基于层次式聚类的文本对象空间索引算法

FasMDS

输入：相似度的阈值 SimBound，类内文本对象数量阈值 InnerBound，原始文本对象的 *TF* 表示，目标空间维数 k

输出：原始文本对象的 k 空间索引

1. 使用算法 1 对原始文本对象进行聚类处理；
2. 根据类间的相似度，表示出类间的距离矩阵；
3. 使用 FasrMap 方法计算各类的 k 维空间索引；
4. 计算各类的有效半径；
5. 对各类内的文本对象，根据相似度信息，构造距离矩阵，利用 MDS 方法建立各对象的空间索引；
6. 将各类中文本对象的索引映射到以该类的空间索引为几何中心，有效半径内的空间；

其中，在算法 2 步骤 4 中，某类的有效半径定义为其它类距离该类最小的距离与某一个界于 0～1 之间常数的乘积，可根据应用的不同而改变。利用算法 2 对文本对象建立空间索引，可以较高程度地保持文本对象间的相似度信息。

4 实验结果

为测试 FastMap-MDS 的性能，从相关研究的演示系统的数据库中获得了 2003 年 7 月 3 日 10:08 到 2003 年 7 月 3 日 11:29 时间段内采集自 http://news.sina.com.cn 的 1000 条新闻的标题作为被处理的文本对象。首先计算出全部文本对象的余弦相似度；近而，以获得 2 维空间索引为目标，分别用 MDS、FastMap 和 FastMap-MDS 进行处理。为比较这三种方法的 stress 值以及相应的变化，采集了目标空间的维数从 1 到 20 变化时各方法的 stress 值序列。最后，采集了顺序为这 1000 个文本对象中的前 100、200、…、1000 个文本对象建立 2 空间索引时各方法所需要的时间。

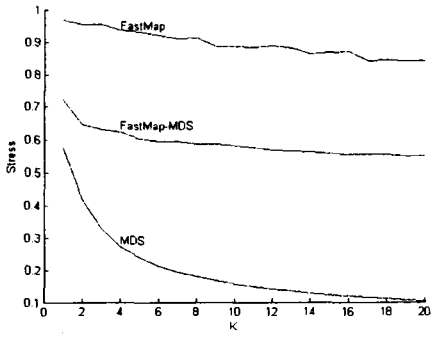

图 2　3 种方法的 Stress 对比曲线

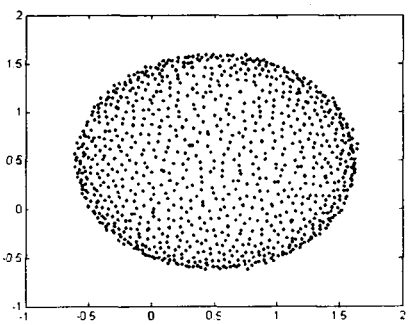

图 3　MDS 方法下文本对象的 2 维空间分布

图 2 给出在利用这三种方法建立 k 空间索引时 *stress* 的变化对比曲线，$k=1\cdots20$。图中，横坐标表示空间的维数，纵坐标表示 *stress* 的取值。显然，MDS 方法具有最小的 *stress* 值，FastMap-MDS 次之，而 FastMap 的最大。因此，FastMap-MDS 建立文本对象 k 空间索引的精度在 MDS 和 FastMap 两者之间。

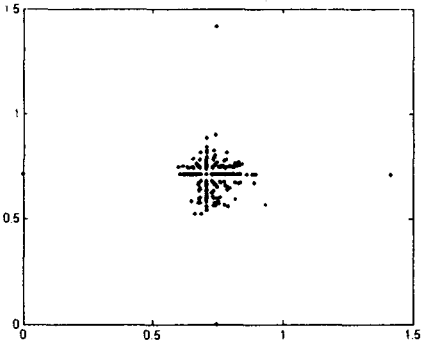

图 4　FastMap 方法下文本对象的 2 维空间分布

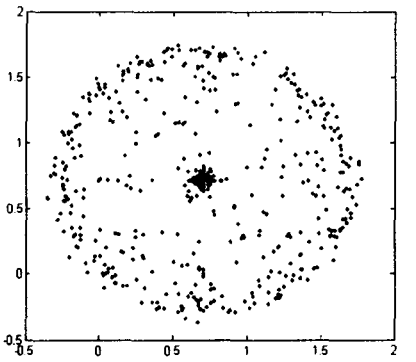

图 5　FastMap-MDS 方法下文本对象的 2 维空间分布

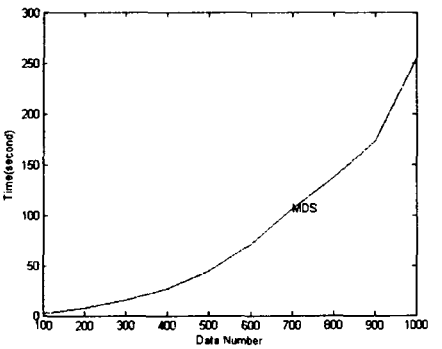

图 6　MDS 方法的时间曲线

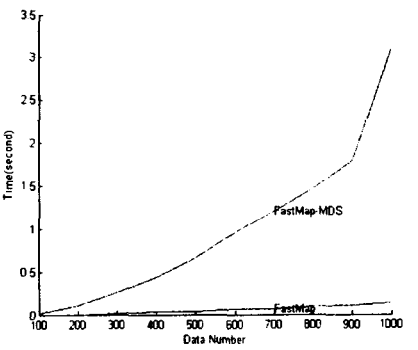

图 7　FastMap、FastMap-MDS 方法的时间曲线

2 空间索引与平面中的点对应。3 种方法所建立的文本对象的 2 空间索引在平面内的分布如图 3～图 5。由于 MDS 具有最高的索引精度，以图 3 为标准，图 5 所示的 FastMap-MDS 建立的 2 索引要比图 4 示意的 FastMap 建立的 2 索引精确得多。

图6、图7给出了顺序为这1000条新闻中的前100、200…1000条新闻建立2空间索引时3种方法所使用的时间。横坐标表示数据量，纵坐标表示时间。数据处理任务包括相似度、距离信息的构造和建立2空间索引两部分。显然，FastMap-MDS方法所使用的时间界于MDS和FastMap之间，如对1000个文本对象建立2空间索引，MDS用时254.718秒，FastMap-MDS用时3.11秒，而FastMap仅用时0.141秒。由于FastMap建立空间索引的精度较差，而在研究中，由于需要10分钟以内完成全部新闻及其相关特征描述的更新，而FastMap-MDS使用的时间在可接受范围之内，故FastMap-MDS是FastMap和MDS在精度和效率之间的有效平衡。

实验使用Intel P4 1.7G CPU，1G DDR266内存，数据库系统为SQL Server 2000企业版。编程环境为Visual C++ 6.0，使用ADO技术操作存储在数据库中的数据。

研究展望 数据的空间索引，在数据的压缩存储、可视化处理、信息查询以及其它领域有着重要的应用。本文以文本数据为研究对象，在聚类处理的基础之上，进行了有益的尝试，获得了肯定的结论。只要在对数据间相似性度量、距离良定义的前提下，这类方法可应用到其它研究和生产领域。在其它领域中的有效运用，是未来的研究之一。

在对数据对象进行聚类处理后，需要定义各类的有效半径。各类的有效半径决定了各类中数据的空间索引在空间中的分布区域。如何有效地定义各类的有效半径或空间拓扑，是提高可视化效果的关键问题之一。同时，不同领域内的数据有着自己的特点，如何根据这些特点，结合实际应用需求，设计有效的空间索引方法，也是值得关注的问题。

致谢 本文受“面向21世纪教育振兴行动计划”资助。

参考文献

1 陈恩红，等. 基于神经网络的增量式数据索引机制研究. 小型微型计算机系统，2003(10)：1783～1786

2 Faloutsos C. FastMap: A Fast Algorithm for indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In: Proc. of ACM SIGMOD, 1995. 163～174

3 Jagadish H V. A retrieval technique for similar shapes. In: Proc. ACM SIGMOD Conf, May 1990. 208～217

4 Torgerson S. Multidimensional scaling: I. theory and method. Psychometrika, 1952, 17: 401～419

5 Kruskal J B, Wish M. Multidimensional scaling. SAGE publications, Beverly Hills, 1978

6 Ding C. Cluster merging and splitting in hierarchical clustering algorithms. In: IEEE Intl. Conf. on Data Mining (ICDM'02), Dec. 2002. 139～146

7 张振亚. 基于文本信息检索的知识发现技术研究：[中国科学技术大学博士学位论文]. 中国科学技术大学档案馆，2004

(上接第133页)

观的了解。它只是将搜集到的资源信息转换成prolog格式，然后利用prolog scheduler来进行时间安排。而且它的资源信息提供者是采用硬编码方式，是固定的。而我们的工作利用语义Web和语义Web服务技术可以动态地发现资源信息，更适用于动态开放的网络环境。

结论 在本文中我们将语义Web服务和约束满足技术结合起来，构造了一个旅游活动规划系统。我们利用了语义Web服务在动态环境中发现相应资源信息提供者的能力和约束满足技术在处理涉及大量资源信息规划问题的优势。而且我们用对象变量来标识要解决的问题，这样可以从更较高层次更清晰地将要解决的问题模型为约束满足问题，而且这恰好和用语义Web标注语言标识的结构资源信息吻合。我们构造了一个独立于问题的约束满足求解Agent，其通过Web服务和外部交互，只要将要解决的问题描述为如图3所示格式，此Agent就会用高效算法将问题解出，并按照一定格式返回。

更复杂的一种活动规划涉及到多个人之间的协商问题，例如，有多个人参加一个会议，不同的人有不同的日程安排，会议地点可以选择多个地方，不同的人又有不同的偏好，我们将来考虑这种更复杂的规划活动。

传统上，约束满足一直被应用于closed-world环境下，即变量取值域和约束在开始就是固定的。而在Internet等open-world环境中，变量取值域和约束需要从动态的信息源中发现，例如关于航班信息，不同的航空公司提供不同的信息源，而且这些信息量可能很大，在开始时将某些变量的值域全部列出是不现实的，这样很多高效的约束满足算法的基础constraint propagate不再成立。为此Boi Faltings等人在文[10]中提出了open constraint satisfaction问题，并提出了一个将信息收集和约束解决交互进行的完备算法。我们将来打算将语义Web服务资源信息发现能力和open constraint satisfaction结合起来，探讨一下如何将约束满足应用于大量的语义Web标注的资源信息环境中，并且如何用动态和模糊约束满足算法来找到优化解，从而使约束满足技术可以真正应用到语义Web时代。

参考文献

1 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001, 284(5): 34～43

2 The OWL Services Coalition. OWL-S: Semantic Markup for Web Services. Available at: http://www.daml.org/services/owl-s/1.0/owl-s.html. accessed on Jan. 2004

3 The RuleML Coalition. The Rule Markup Initiative. Available at: http://www.ruleml.org. accessed on Mar. 2004

4 Tsang E. Foundations of Constraint Satisfaction. London, UK: Academic Press, 1993

5 Paolucci M, Kawmura T, Payne T, et al. Semantic Matching of Web Services Capabilities. In: The First Int. Semantic Web Conf., Sardinia, Italy, 2002

6 Willmott S, Calisti M, Faltings B, et al. CCL: Expressions of Choice in Agent Communication. In: The Fourth Intl. Conf. on MultiAgent Systems (ICMAS-2000), Boston, USA, 2000

7 Torrens M, Weigel R, Faltings B. Java Constraint Library: bringing constraints technology on the Internet using the Java language: [In Working Notes of the Workshop on Constraints and Agents, Tehnical Report WS-97-05, AAAI-97]. Rhode Island, USA: Marc Torrens, 1997

8 Macho S, Torrens M, Faltings B. A Multi-Agent Recommender System for Planning Meetings. Available at: http://liawww.epfl.ch/Publications/Archive/Macho-Gonzaleza.pdf, accessed on Mar, 2004

9 Grimnes G AA, Chalmers S, Edwards P, et al. GraniteNights-A Multi-Agent Visit Scheduler Utilising Semantic Web Technology. Available at: http://www.csd.abdn.ac.uk/~ggrimnes/pubs/GraniteNights.pdf, accessed on Mar. 2004

10 Faltings B, Macho-Gonzalez S. Open Constraint Optimization. CP, 2003, 243(4): 303～317