ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.2, February 2007, pp.311−322
DOI: 10.1360/jos180311

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

一种基于概念相似度的数据分类方法*

彭 京[1,2+], 唐常杰[1], 元昌安[1], 李 川[1], 胡建军[1]

[1](四川大学 计算机学院,四川 成都 610065)

[2](成都市公安局 科技处,四川 成都 610017)

A Data Classification Method Based on Concept Similarity

PENG Jing[1,2+], TANG Chang-Jie[1], YUAN Chang-An[1], LI Chuan[1], HU Jian-Jun[1]

[1](School of Computer Science, Sichuan University, Chengdu 610065, China)

[2](Science and Technology Department, Chengdu Public Security Bureau, Chengdu 610017, China)

+ Corresponding author: Phn: +86-28-85466105, E-mail: pj@pku.edu.cn, http://www.cs.scu.edu.cn/~tangchangjie

Peng J, Tang CJ, Yuan CA, Li C, Hu JJ. A data classification method based on concept similarity. *Journal of Software*, 2007,18(2):311−322. http://www.jos.org.cn/1000-9825/18/311.htm

Abstract: In this paper, a method of classification is proposed based on the similar information of data properties. The new method assumes that data properties are basic vectors of *m* dimensions, and each of the data is viewed as a sum vector of all the property-vectors. It suggests a novel distance algorithm to get the distance of every pair of the property based on similar information of the basic property vectors. An algorithm of data classification is also presented based on correlation computing formula composed of property vectors and projections of each other. Efficiency of the new method is proved by extensive experiments.

Key words: data mining; concept similarity; similar distance; property vector; classification

摘 要: 依据数据属性间的相似信息,提出了一种分类方法.该方法将属性矢量化,属性作为 *m* 维空间的基本矢量,数据记录作为属性矢量的和.利用属性间先验的概念相似信息,给出了求取任意属性矢量对的相似距离算法,并将数据间相关度计算转换为属性矢量及其相互投影的公式,从而得到任意两条数据的相关度;利用相关度,提出了一种分类算法.用详实的实验证明了该算法的有效性.

关键词: 数据挖掘;概念相似度;相似距离;属性矢量;分类

中图法分类号: TP311 文献标识码: A

数据挖掘(data mining)或数据库中的知识发现,是从大规模数据库的数据中抽取有效的、隐含的、有潜在使用价值的有用信息的过程,是当今众多学科领域,特别是数据库领域前沿的研究课题之一.在数据挖掘算法中,分类(classification)是具有广泛应用领域的最重要的问题之一,旨在发现属于同一类数据对象的共同特性,构

* Supported by the National Natural Science Foundation of China under Grant No.60473071 (国家自然科学基金); the China Postdoctoral Science Foundation under Grant No.20060400002 (中国博士后科学基金); the Major Science and Technology Project of Sichuan Province of China under Grant No.04SG1640 (四川省重点科技计划); the Sichuan Youth Science and Technology Foundation of China under Grant No.07ZQ026-055 (四川青年科技基金)

Received 2004-09-08; Accepted 2006-04-26

造分类器,对未知类别的样本进行类别的判断.

目前,已有若干种方法和技术用于构造分类模型,如决策树、决策表、神经网络、最近邻、贝叶斯方法以及支持向量机等.通常,分类算法需要 3 个要素,即已知的训练数据、测试数据和问题所在领域知识.领域知识没有直接保存在样本数据中,而是潜在地蕴涵在数据属性的关系上.如在文本分类问题中,词组集合作为文本属性往往成为分类的基础,而词组间实际上存在着潜在的相似关系.前述的分类模型中没有考虑这部分知识,因此在很多具体分类问题上,往往效果并不理想.

本文提出了一种新的基于概念相似度的数据分类方法 CCS(classification method based on concept similarity),其要点是从最近邻问题入手,在领域潜在知识的表示上作了一些探索.最近邻是在给定的具有 n 个元组的数据表 D 上,寻找与给定查询最接近或近似最接近的元组.利用一个尺度空间,最近邻方法可以很容易地将相似搜索或基于模式的分类问题转化为空间矢量的距离问题.在这方面最近的理论研究成果参见文献[1–3],应用研究成果参见文献[4–9].最近邻问题中的核心问题是距离计算,Kleinberg[2]提出了 d 维空间中沿着随机线性投影的思路,而 Dwork 提出了采用等级聚合(rank aggregation)的方法[10].这些方法均将数据表中的元组视为空间矢量,其坐标系为数据表的属性集合.

CCS 方法将数据表中的元组视为空间矢量,与已有的最近邻分类模型的不同点在于:不将数据表的属性集合视为坐标系,而将每个元组的属性看作基于一个未知维度空间的矢量,而元组表示为每个属性矢量的和矢量.同时,根据先验的领域相似信息,定义了标准属性矢量距离及投影方法.然后根据相关度公式,将距离计算问题转化为属性矢量及其相互投影的公式.实验表明,该方法具有较高的分类精度,与未引入领域相似信息相比,分类准确度有明显的提高.

本文第 1 节讨论如何将已获取的领域相似信息转换为标准属性矢量距离及投影算法.第 2 节给出元组矢量相关度的计算公式及分析.第 3 节提出分类算法框架.第 4 节是本文的实验部分.第 5 节总结全文,并提出下一步的研究方向.

1 属性矢量距离及投影算法

在已获取了属性间相似信息的前提下,我们需要寻找一种表达数据属性关系的方法,即如何统一刻度数据属性及其相互关系.只有这样,才能真正引入领域知识.本文的设想是:

- 将数据表的每个属性定义为基于一个多维空间的单位矢量;
- 将数据表中的每个元组视为这些单位矢量的加权和矢量,权重就是元组在每个属性上的取值;
- 根据相近程度,将已获取的属性间相似信息定义为单位矢量之间的距离.

通过这样的假定,就可以很好地利用空间矢量的性质和计算公式来表达属性之间和元组之间的相关程度.其形式化描述如下.

1.1 定 义

定义 **1**(属性矢量). 设 $r_1,r_2,\ldots,r_m$ 是属性值域,$R^m=(r_1\times r_2\times\ldots\times r_m)$是 m 维矢量集合.

(1) 设 $f_1,f_2,...,f_d$ 是空间 R^m 中的矢量,d 为正整数,即 $f_i=(v_{i,1},v_{i,2},\ldots,v_{i,m}),v_{i,k}\in r_k$,如果$|f_i|=1,c$ 为非负实数,则称 f_i 为空间 R^m 中的单位属性矢量,称 $c\cdot f_i$ 为属性矢量,$i=1,\ldots,d$;

(2) 设值域 $A_i=Dom(f_i)$,以 $F=f_1,f_2,\ldots,f_d$ 为模式的最大关系记为 $D^d=(A_1\times A_2\times\ldots\times A_d)$.

例 1:设数据库 D^d 为某商场中所有的交易记录,则商品种类构成了 D^d 的属性集合.假设某顾客在商场的购物记录为:毛巾 1 条,啤酒 2 瓶,则毛巾和啤酒是数据库 D^d 的属性,即毛巾,啤酒$\in F$,而购买的数量作为属性的取值.看上去毛巾和啤酒作为属性没有关系,但实际上,它们又可以用多个共同特性来进行刻画,如长度、宽度、重量、价格等,长度、宽度这些特征用 $r_1,\ldots,r_m$ 来表示,而 R^m 就等于这些特性构成的空间,具体的取值则用 $v_{i,1},v_{i,2},\ldots,v_{i,m}$ 来表示.

定义 **2**. 在与定义 1 相同的环境和符号系统下:

(1) 对于$\forall v\in D^d$,设其对应的属性集合的取值为$(u_1,u_2,\ldots,u_d)$,其中,u_i 为非负实数,i=,1…,d,则称 $u=\sum_{i=1}^{d}u_i\cdot f_i$ 为 v 对应的元组矢量,其中 $u\in R^m$.

(2) 对于$\forall r_i,r_j\in R^m$,称从 r_i 到 r_j 的矢量投影为$\xi(r_i,r_j)$.于是有$\xi(r_i,r_j)=|r_i|\cdot\cos(r_i,r_j)$,其中,$\cos(r_i,r_j)$表示矢量 r_i 与 r_j 的夹角的余弦.

(3) 设 $u,v,w\in F,c_1,c_2$ 为非负实数,$\langle u,v,c_1\rangle$表示从 u 到 v 的一条路径,路径长度为 c_1.设 $path_0=\{\langle u,v,c_1\rangle,\langle v,w,c_2\rangle\}$ 为已知路径集合,则称路径$\langle u,w,c_1+c_2\rangle$为 $path_0$ 的派生路径.设 $P=\{\langle u,v,c\rangle|u,v\in F,c$ 为非负实数,表示属性 u,v 之间的路径长度$\}$是已知路径集合,则 P 中一切路径及其派生路径的集合称为路径闭包,记为 P^+.

(4) 设 u,v 为 R^m 中的属性矢量,P 是已知路径集合.则称 $d(u,v)=\mathrm{Min}\{h|\langle u,v,h\rangle \text{ in } P^+\}$为从 u 到 v 的属性矢量距离,即从 u 到 v 的最短路径的长度之和.

例 2:设数据库 D^d 具有属性(A,B,C),设 $P=\{\langle A,B,1\rangle,\langle B,C,3\rangle\}$,即属性 A,B 之间的距离为 1,属性 B,C 之间的距离为 3,则可推出 A,C 的属性矢量距离为 4,即经过路径 $A\rightarrow B\rightarrow C$ 的距离之和.

由以上定义可知:数据库 D^d 由多个属性构成,每个属性为基于一个多维空间的单位矢量,这些属性一起构成了属性集合;数据库的每条记录构成了元组,元组可以视为这些单位矢量的加权和矢量,权重就是元组在每个属性上的取值.属性之间的相近程度可以用距离来衡量,而距离定义为属性矢量之间的最短路径和.

1.2 最短属性距离算法(**shortest attribute distance algorithm**,简称为**SADA**)

本文首先将已知的数据属性之间的相似信息转换为路径集合 P,而属性相似程度用路径长度来表示.根据定义 1 和定义 2,本文构造了一种求取$\forall r_i,r_j\in F(i,j=1,\ldots,d)$之间属性矢量距离 $d(r_i,r_j)$的算法.该算法思路为:

- 首先将已知的属性矢量之间的路径集合 P,即已获取的属性间相似规则,进行排序;
- 建立$\forall r_i,r_j\in F(i,j=1,\ldots,d)$之间初始矢量距离矩阵 $M(d\times d)$,依次遍历每条已知路径集合 $p\in P$.判断加入此条路径后,其他属性矢量与属性矢量 $p.u$ 或 $p.v$ 之间是否存在更小的距离,如果有则替换,同时将其加入一个变更属性集合.
- 然后,根据刚才产生的属性集合,求彼此间在加入此条路径后是否有更小的距离,如果有则替换;直到所有的路径处理完成为止.

具体 SADA 算法伪码表示如下:

算法 **1**. SADA 算法.

Input: P, where all $p\in P$ has $p.u<p.v$.

Output: $M(d\times d)$.

Begin

```
1    Sort P.u, P.v by ascend through QuickSort Algorithm;
2    Initialize M(d×d); M(i,j)←Max_value, i,j∈1,…,d;
3    For Each p=⟨u,v,c⟩∈P Do
4        If p.c<M(u,v) Then
5            M(u,v)=p.c;M(v,u)=p.c;
6            H={∅};Q={∅};
7            For i=1 To d Do
8                If M(i,v)<M(u,v)+M(i,u) Then
9                    M(i,v)=M(u,v)+M(i,u);M(v,i)=M(i,v);
10                   H=H+{i};
11               End If
12           End For
13           For i=1 To d Do
```

```
14                  If M(i,u)<M(u,v)+M(i,v) Then
15                      M(i,u)=M(u,v)+M(i,v);M(u,i)=M(i,u);
16                      Q=Q+{i}
17              End If
18          End For
19          For Each of {h,q|h∈H,q∈Q,h<>q} Do
20            If M(h,q)<M(h,u)+M(u,v)+M(v,q) Then
21              M(h,q)=M(h,u)+M(u,v)+M(v,q);
22              M(h,q)=M(q,h);
23            End If
24          End For
25      End If
26  End For
End
```

命题 **1**. 对于$\forall r_i,r_j\in F(i,j=1,\ldots,d)$,如果存在一条从 r_i 到 r_j 的最短路径,且最短路径距离之和等于 k,则根据 SADA 算法得到距离矩阵,有 $M(i,j)=k$,即求出的矢量距离矩阵 M 代表了矢量之间的属性矢量距离;SADA 算法的复杂度为 $O(n\times d^2)$,其中,$n=|P|$.

证明:首先证明命题的第 1 部分.设对$\forall r_i,r_j\in F(1\le i<j\le d)$,从 r_i 到 r_j 的最短路径记为 $Path(i,j)=(h_1,\ldots,h_k)$,其中,$h_1=r_i,h_k=r_j$.

采用数学归纳法证明:当 $n=|P|=1$ 时,即相似规则数仅为一条;由 SADA 算法,命题自然成立.

假设当 $n=x$,命题成立;考察当 $n=x+1$ 时的情况.在没有应用第 $x+1$ 条路径规则之前,由假设可知,此时,M 得到一个所有属性矢量之间的最短路径距离(不考虑第 $x+1$ 条以后的路径规则).设第 $x+1$ 条路径规则为$\langle u,v,c\rangle$,则有 $u<v$,如果增加第 $x+1$ 条路径规则,由 SADA 算法第 7~18 行可知,可求得 u,v 两个矢量到任意矢量的最短距离.

如果增加第 $x+1$ 条路径规则后,存在一个从 r_i 到 $r_j(i<j$,且 $r_i,r_j<>u,v)$的最短路径发生了变化,则这条路径必然包含第 $x+1$ 条规则$\langle u,v,c\rangle$.

因此有 $Path(r_i,r_j)=\min(Path(r_i,u)\cup Path(v,r_j),Path(r_i,v)\cup Path(u,r_j))$. 假设 $Path(r_i,r_j)$ 满足 :$Path(r_i,r_j)=Path(r_i,u)\cup Path(v,r_j)$,则必然有 $Path(r_i,v)=Path(r_i,u)\cup v$.这是因为 $Path(r_i,r_j)$是从 r_i 到 r_j 的最短路径,则有从 r_i 到 $\forall y\in Path(r_i,r_j)$的最短路径,必定等于 $Path(r_i,r_j)$中 r_i 到 y 的序列.否则,我们就一定可以找到另一条路径,使得 r_i 到 r_j 的距离小于 $Path(r_i,r_j)$.

因为$\langle u,v\rangle$是新增的规则,且 $u,v\in Path(r_i,v)$,于是由 SADA 算法可知,$r_i\in H$;同理,根据 $Path(u,r_j)=u\cup Path(v,r_j)$有 $r_j\in Q$;反之,如果 $Path(r_i,r_j)=Path(r_i,v)\cup Path(u,r_j)$,则同理可知,$r_i\in Q,r_j\in H$.因此,无论 $Path(r_i,r_j)$满足哪种情况,均有 $r_i, r_j\in\{h,q|h\in H,q\in Q,h<>q\}$.根据算法第 19~23 行所示,必可求出 r_i,r_j 的最短距离.因此,命题在 $n=x+1$ 时成立.

综上所述,命题的第 1 部分得证.

现证明命题的第 2 部分.因为 SADA 算法首先完成对相似规则集合的快速排序,所以由快速排序算法可知,其时间复杂度为 $n\times\log(n)$;其次,算法依次读出每条相似规则,然后对每条相似规则与矩阵对应的其他行和列的值进行比较.故此部分时间复杂度小于等于 $O(n\times d)$,因为$\{h,q|h\in H,q\in Q,h<>q\}$最坏情况下个数为 d^2,于是,时间复杂度为小于等于 $O(d^2\times n)$,故有 SADA 算法的复杂度为 $O(n\times\log(n)+n\times d+d^2\times n)=O(d^2\times n)$(由 $n<d^2$ 可知). □

求取任意两点之间的最短路径是所有求解路径问题的基础,它有广泛的用途,如在网络寻优、道路交通等领域.问题的目标是找到从图中每个顶点 v 到其他任意顶点 u 的最短路径.利用 Dijkstra 算法,通过对每个顶点执行一次搜索,可以得到一种复杂度为 $O(d^3)$的求取任意两点之间最短路径算法.求取任意两点之间的最短路径,更为通用、简洁、高效的方法是 Floyd-Warshall 算法[11],它的时间复杂度同样为 $O(d^3)$.Han 在前期成果基础上提出了一种新的求取任意两点之间的最短路径算法[12],算法时间复杂度降为 $O(d^3(\log\log(d)/\log(d))^{5/7}$.但在实际

应用中,其性能与 Floyd-Warshall 差别不大[11].近期关于求取任意两点之间最短路径问题的研究成果可以参考文献[11−13].

由复杂度可知,SADA 算法与 Floyd 算法相差不大(Floyd 算法复杂度为 $O(d^3)$).但实际对比测试表明:当 n 数目较小时,SADA 算法速度要远远快于 Floyd 算法.具体测试情况见本文实验部分.

1.3 属性距离与矢量投影的转换

利用上一节得到的任意属性矢量之间的距离,本文构造了一个在 m 维空间中属性矢量投影的计算公式.根据定义 2 有,矢量投影 $\xi(r_i,r_j)=|r_i|\cdot\cos(r_i,r_j)$.因为 $\cos(r_i,r_j)$无法直接获知,于是,本文利用已求得的矢量之间的距离来近似表示,

$$\cos(r_i,r_j)=\begin{cases}1, & \text{if } d(r_i,r_j)=0\\ (\varphi-d(r_i,r_j))/\varphi, & \text{if } 0<d(r_i,r_j)<\varphi\\ 0, & \text{if } d(r_i,r_j)\geq\varphi\end{cases} \tag{1}$$

矢量距离与矢量夹角余弦的对应关系如图 1 中的粗线所示.在具体应用中,也可以将函数关系用曲线表示,如图 1 中的细线所示.与公式(1)相比,可以降低距离较近的矢量夹角的余弦值,同时,距离较远的矢量对的余弦值仍然可以大于 0.

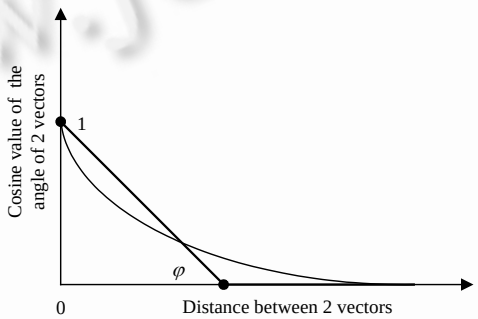

Fig.1 Function relationship between vectors' distance and vector-projection

图 1 矢量距离与投影之间的函数关系

类似于公式(1),对于任意属性矢量,采用如下公式求取属性矢量投影:

$$\xi(f_i,f_j)=|f_i|\cdot\cos(f_i,f_j)=\begin{cases}|f_i|, & \text{if } d(f_i,f_j)=0\\ |f_i|\cdot(\varphi-d(f_i,f_j))/\varphi, & \text{if } 0<d(f_i,f_j)<\varphi\\ 0, & \text{if } d(f_i,f_j)\geq\varphi\end{cases} \tag{2}$$

2 元组矢量相关度计算

在上一节中,我们利用 SADA 算法获取了属性矢量间的距离,并根据属性矢量的距离,得到了任意属性矢量之间的投影公式.本节利用这个结果,推导元组矢量相关度的计算公式.

由定义 2 可知,元组矢量 $u=\sum_{i=1}^{d}u_i\cdot f_i$ 且 $u\in R^m$.由此推出元组矢量 u 的长度为

$$|u|=\sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d}(\cos(f_i,f_j)\cdot u_i\cdot|f_i|\cdot u_j\cdot|f_j|)}=\sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d}(\cos(f_i,f_j)\cdot u_i\cdot u_j)} \tag{3}$$

根据矢量性质,元组矢量 u 到任意属性矢量 $f_k\in F(k=1,\ldots,d)$的投影等于各分矢量投影到 f_i 之和,即

$$\xi(u,f_k)=\sum_{j=1}^{d}\xi(u_j\cdot f_j,f_k)=\sum_{j=1}^{d}u_j\cdot\xi(f_j,f_k)=\sum_{j=1}^{d}u_j\cdot\cos(f_j,f_k) \tag{4}$$

由式(3)、式(4),可以得到元组矢量 u 与任意属性矢量 $f_k\in F(k=1,\ldots,d)$的夹角余弦,

$$\cos(u, f_k) = \frac{\xi(u, f_k)}{|u|} = \frac{\sum_{j=1}^{d} u_j \cdot \cos(f_j, f_k)}{\sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d}(\cos(f_i, f_j)\cdot u_i \cdot u_j)}} \tag{5}$$

通过式(5),可以得到元组矢量 u 与每个属性矢量 $f_k \in F(k=1,\ldots,d)$ 的夹角余弦值.当每个属性矢量 f_k 彼此正交,且 $i<>j$ 时,有 $\cos(f_i,f_j)=0$,当 $i=j$ 时,有 $\cos(f_i,f_j)=1$,其中,$f_i,f_j \in F$.根据式(3),推出元组矢量 u 在此条件下的长度为

$$|u| = \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d}(\cos(f_i, f_j)\cdot u_i \cdot u_j)} = \sqrt{\sum_{i=1}^{d} u_i^2} \tag{6}$$

而式(4)变为 $\xi(u,f_k)=u_k$,代入式(5),有

$$\cos(u, f_k) = \frac{\xi(u, f_k)}{|u|} = \frac{u_k}{\sqrt{\sum_{i=1}^{d} u_i^2}} \tag{7}$$

可以看出,式(6)是 d 维空间中矢量长度公式,式(7)为空间矢量与各个维度的夹角余弦的公式.因此,式(3)~式(5)实际上是对 m 维空间矢量的长度、投影以及与各个维度的夹角余弦公式的推广.

另外,当每个属性矢量 f_k 完全重合时,有 $\cos(f_i,f_j)=1$,其中,$f_i,f_j \in F$.此时,根据式(3)有

$$|u| = \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d}(\cos(f_i, f_j)\cdot u_i \cdot u_j)} = \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d} u_i \cdot u_j} = \sqrt{\left(\sum_{i=1}^{d} u_i\right)^2} = \sum_{i=1}^{d} u_i.$$

根据式(3)~式(5),我们定义 $\forall r,s \in D^d$ 的相关度公式为

$$\lambda(r,s) = \frac{\sqrt{2}}{2}\cdot\sqrt{\sum_{k=1}^{d}\left(\frac{r_i}{|r|}+\frac{s_i}{|s|}\right)\left|\cos(r, f_k)-\cos(s, f_k)\right|} \tag{8}$$

其中,$f_k \in F(k=1,\ldots,d)$.

在实际应用中,很多情况下,高维空间中元组矢量的大部分属性矢量取值为空.如在文本分类中,一个具体文本中的词组集合只是语言所有词组集合的极小一部分.我们接着讨论在此条件下的相关度计算公式.设 $r \in D^d, r=\sum_{i=1}^{x} r_i \cdot f_i^r, f_i^r \in F$ 表示 r 中所有取值非空的标准属性矢量;同理,设 $s \in D^d, s=\sum_{i=1}^{y} s_i \cdot f_i^s, f_i^s \in F$ 表示 s 中所有取值非空的标准属性矢量.将式(8)中所有为 0 的项目去掉,则有

$$\lambda(r,s) = \frac{\sqrt{2}}{2}\cdot\sqrt{\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r, f_i^r)-\cos(s, f_i^r)\right| + \sum_{i=1}^{y}\frac{s_i}{|s|}\cdot\left|\cos(s, f_i^s)-\cos(r, f_i^s)\right|} \tag{9}$$

考察式(9),我们可以得到如下命题:

命题 **2**. 设 r,s 是元组矢量,$r,s \in D^d$,λ为式(9)中定义的相关度公式,则当 r,s 相同时,有 $\lambda(r,s)=0$.

证明:当 $r=s$ 时,有 $\left|\cos(r, f_i^r)-\cos(s, f_i^r)\right| = \left|\cos(r, f_i^r)-\cos(r, f_i^r)\right| = 0$,$i=1,\ldots,x$,即式(9)左边的求和项中每项均为 0;同理,式(9)右边的求和项也为 0,故有 $\lambda(r,s)=0$,命题得证. □

命题 **3**. 设 r,s 是元组矢量,$r,s \in D^d$,λ为式(9)中定义的相关度公式,则当 r,s 夹角为 0 时,有 $\lambda(r,s)=0, r,s \in D^d$.

证明:当 r 与 s 的夹角为 0 时,根据矢量特点有元组矢量 r,s 与任意矢量 $t \in R^m$ 的夹角均相等,则有 $|\cos(r,f_i)-\cos(s,f_i)|=0, f_i \in F(i=1,\ldots,d)$.根据式(9),有 $\lambda(r,s)=0$,命题得证. □

命题 **4**. 设 r,s 是元组矢量,$r,s \in D^d$,λ为式(9)中定义的相关度公式,则当 r,s 中所有属性矢量彼此均正交,且 r,s 没有包含相同的标准属性矢量时,有 $\lambda(r,s)=1$.

证明:当所有的属性矢量彼此均正交时,必然有 $\forall f_i,f_j \in F$,当 $i<>j$ 时,有 $\cos(f_i,f_j)=0$;当 $i=j$ 时,有 $\cos(f_i,f_j)=1$.根据式(7), $\cos(r, f_k^r) = \frac{r_k}{\sqrt{\sum_{i=1}^{x} r_i^2}}$;根据题设,r,s 没有包含相同的标准属性矢量;结合式(5),于是有 $\cos(s,r_k)=0$. □

再根据式(6),于是有式(9)中左边的求和值:

$$\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r,f_i^r)-\cos(s,f_i^r)\right|=\sum_{i=1}^{x}\frac{r_i}{\sqrt{\sum_{j=1}^{x}r_j^{\,2}}}\cdot\frac{r_i}{\sqrt{\sum_{j=1}^{x}r_j^{\,2}}}=\sum_{i=1}^{x}\frac{r_i^{\,2}}{\sum_{j=1}^{x}r_j^{\,2}}=1.$$

同理,式(9)右边的求和值也等于 1.故有 $\lambda(r,s)=\frac{\sqrt{2}}{2}\cdot\sqrt{1+1}=1$.命题得证. □

命题 **5**. 设 r,s 是元组矢量,$r,s\in D^d$,λ为式(9)中定义的相关度公式.则当 r 包含的每个属性矢量$(r_1,\ldots,r_u)$均与 s 中任意属性矢量$(s_1,\ldots,s_v)$正交;反之也是一样,且 r,s 没有包含相同的标准属性矢量,则同样有$\lambda(r,s)=1$.

此命题实际上是对命题 4 的推广,根据式(3)、式(5),有式(9)中左边的求和值

$$\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r,f_i^r)-\cos(s,f_i^r)\right|=\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r,f_i^r)\right|=\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r,f_i^r)\right|,$$

$$\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r,f_i^r)-\cos(s,f_i^r)\right|=\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r,f_i^r)\right|=\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\left|\cos(r,f_i^r)\right|$$

$$=\sum_{i=1}^{x}\frac{r_i}{|r|}\cdot\frac{\sum_{m=1}^{x}r_m\cdot\cos(f_m^r,f_i^r)}{\sqrt{\sum_{m=1}^{x}\sum_{n=1}^{x}(\cos(f_m^r,f_n^r)\cdot r_m\cdot r_n)}}=\frac{\sum_{i=1}^{x}(r_i\cdot\sum_{m=1}^{x}r_m\cdot\cos(f_m^r,f_i^r))}{\sum_{m=1}^{x}\sum_{n=1}^{x}(\cos(f_m^r,f_n^r)\cdot r_m\cdot r_n)}=1.$$

同理,式(9)中右边的求和值也等于 1.故有 $\lambda(r,s)=\frac{\sqrt{2}}{2}\cdot\sqrt{1+1}=1$.命题得证. □

命题 **6**. 设 r,s 是元组矢量,$r,s\in D^d$,λ为式(9)中定义的相关度公式,令 $w_i=\left|\cos(r,f_i^r)-\cos(s,f_i^r)\right|,i=1,\ldots,x$,则有 $\frac{\partial(\lambda(r,s)}{\partial w_i}\geq 0$;同样,设 $E_i=\left|\cos(r,f_i^s)-\cos(s,f_i^s)\right|,i=1,\ldots,y$,有 $\frac{\partial(\lambda(r,s)}{\partial E_i}\geq 0$.

证明:当 w_i 增大时,式(9)左边求和值也相应增加,于是就有$\lambda(r,s)$增大,因为 $w_i\geq 0$,推出$\lambda(r,s)\geq 0$,故有 $\frac{\partial(\lambda(r,s)}{\partial w_i}\geq 0$;同理,有 $\frac{\partial(\lambda(r,s)}{\partial E_i}\geq 0$. □

根据命题 6 可知,如果矢量 r,s 与任意属性矢量的夹角余弦差越大,则 r,s 越不相似,而$\lambda(r,s)$的值也就越大.

命题 2~命题 6 中对式(9)的讨论同样适用于式(8).只需简单地设置 $x=y=d$ 就可以看到,此时,式(9)与式(8)完全相同.因此,命题 2~命题 6 在式(8)下仍然满足.

通过上述讨论可以发现,元组矢量间的相关度公式$\lambda(r,s)$很好地表示了矢量之间的相似程度.它基于这样一个归纳偏置:数据库中的元组可以很好地用一个 m 维空间的矢量来表示,尽管可能我们不知道这个 m 维空间的明确定义和 m 的大小.观察式(8)、式(9),所有的计算都可以最终由属性矢量及相互之间的夹角来表示,而属性之间的夹角可以通过根据预先得到的矢量距离来求出.因此,式(8)、式(9)中每一项均是可计算的,不存在未知变量.

比较式(8)、式(9)可以发现,利用式(9)可以高效地解决在属性维度过高而实际元组矢量中仅包含少量属性矢量的情况下,如何计算元组相关度的问题.

3 基于概念相似度的分类算法

基于上述讨论,本文构造了一种基于概念相似度的分类算法:CCS(classification method based on concept similarity).算法目标是对新的数据求出与之相关度最小的元组矢量,以该元组矢量所属的分类来对新数据进行划分.如果分类允许重复,则对于任意 $r\in R^m$,对应的分类 $C(r)$的集合为

$$C_r=\{Classify(p)|p\in D^d,d(p,r)\leq\theta,d(p,r)=\min\{d(q_j,r)|q_j\in D^d\}\},$$

否则,有

$$C_r=\{Classify(p_i)|p_i\in D^d,d(p_i,r)\leq\theta,i=\min\{k|d(q_k,r)=\min\{d(q_j,r)|q_j\in D^d\}\}\},$$

其中,$Classify(p)$表示元组矢量所属的类别;D^d 表示样本数据表.

算法分为两部分:

1) 预处理过程,调用 SADA 算法,完成从先验的属性相似数据得到属性之间的距离;

2) 根据相关度计算公式,对每个待分类的数据求取它与每个样本数据的相关度,求得满足取值最小(值越小越相似)的元组矢量 r,如相关度小于 θ,则将数据分类到 r 对应的分类中,否则作为离群点.

算法的伪码表示如下:

算法 **2**. CCS 算法.

Input: $u=(u_1,u_2,\ldots,u_d)$.

Output: Classification of u.

Begin

Pre-processing:

1 Call SADA algorithm;

Classify-processing:

2 $d_{\min}$=Max_value;

3 $v_{\min}=\varnothing$;

4 **For** Each $v\in D^d$ **do**

5 **if** $d_{\min}>\lambda(u,v)$ **then**

6 $d_{\min}=\lambda(u,v)$;

7 $v_{\min}=v$;

8 **End if**

9 **End For**

10 **If** $d_{\min}<\theta$ **then**

11 **Return** *Classify*($v_{\min}$);

12 **Else**

13 **Return** $\varnothing$;

End

其中,$\lambda(u,v)$表示矢量 u,v 的相关度公式,具体定义参见式(8)、式(9).如不考虑预处理过程,CCS 算法对每个元组分类的时间复杂度为 $O(n\times d)$,其中,n 为训练集的大小;d 为属性矢量的个数.

4 实验与性能分析

本文成功地将 CCS 算法应用到中药方剂功效的自动归纳和依据方剂功效的证状分类研究[14]当中.实验主要比较了 CCS 算法和在不引入属性相似信息时的 CCS 算法(此时,所有属性矢量全部正交,不同矢量的投影全部为 0)以及与标准欧式空间距离在分类精度和性能上的差别.

实验的硬件环境为 PIII533M,512M 内存,40G 硬盘;开发工具为 DELPHI7.0;数据库使用的是微软的 Access 2000.

样本数据为中药脾胃类方库,含脾胃类中药方剂表、相似功效表、基本药物表等信息.其中,中药方剂表中除了方剂基本信息以外(如方名、症状、病因、病机等),主要包括了方剂所含药物的信息,如中药方剂中的化滞调中汤的组成药物见表 1.

在中药方剂中,每种药物还包含了性、味、归经以及分类、功效等信息,且每种药物可以具有多种不同的功效.以药物中的生姜为例,其属性见表 2.

本文的实验过程是:首先得到方剂的组成药物,然后根据组成药物的功效归纳出方剂的功效,最后根据方剂的功效判断方剂所属的类别.具体实验结果如下:

Table 1 Example of Chinese traditional medicine prescription

表 1 中药方剂的组成示例

	Formal name	Original dosage	Dosage-g	Processing
1	Shengjiang	3 pcs.	6	
2	Laifuzi	Lack of dosage	8	Fried
3	Baizhu	1 Qian 5 Fen	5.595	
4	Renshen	1 Qian	3.73	
5	Fuling	1 Qian	3.73	
6	Jupi	1 Qian	3.73	
7	Houpo	1 Qian	3.73	Processed with ginger
8	Shanzha	1 Qian	3.73	
9	Banxia	1 Qian	3.73	
10	Shenqu	8 Fen	2.984	Fried
11	Maiya	8 Fen	2.984	Fried
12	Sharen	7 Fen	2.611	

Table 2 Medicament attribute example

表 2 药物属性示例

	Property	Value
1	Category	Diaphoretics Chinese medical
2	Sub-Category	Exterior cold syndrome relieving Chinese medical
3	Medicine name	Shengjiang
4	Byname	
5	Action	Diaphoresis, expelling cold, eliminating phlegm, subdue qi warming middle-jiao, arrest vomiting
6	Flavors	Pungent
7	Meridian tropism	Lung, spleen and stomach meridians
8	Four natures	Slightly warm
9	Toxicity	
10	Minimum dosage (g)	3
11	Maximum dosage (g)	9
12	Remark	

实验 1. 利用任意属性矢量之间的距离,本文依据式(9)对每个元组矢量与单位属性矢量的夹角余弦进行预处理,即权重 $\cos(r, f_i^r)$,求得脾胃类 1 060 个中药方剂的功效权重.以桂枝汤为例,其部分功效权重见表 3;功效权重曲线如图 2 所示,其中,纵坐标为权重值,横坐标为功效编号.

Table 3 Instance of action weight

表 3 功效权重示例

	Action	Value	Weight
1	Buqi	69.5	0.143 787
2	Jianpi	62.55	0.172 842
3	Zhitong	55	0.230 964
4	Shufeng	41.25	0.119 249
5	Fahan	41.25	0.107 37
6	Sanhan	37.125	0.116 029
7	Jiebiao	37.125	0.115 163
8	Lianyin	37.125	0.035 069

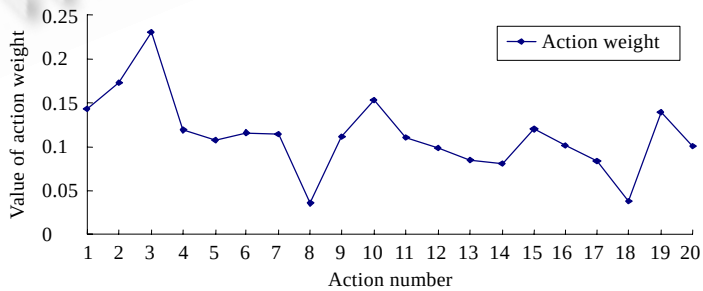

Fig.2 Curve of Chinese traditional medicine action weight

图 2 中药功效权重曲线

实验 2. 根据方剂功效,判断方剂的证状分类.本文分别实现了 3 种方式的分类算法.实验中,以 1 060 条已分类方剂功效数据作为样本,以 3 组数据(每组包含 400 条数据)测试 3 种方法的分类精度,实验结果见表 4、表 5.

Table 4 Comparison of the three methods' accuracy rate (%)

表 **4** 3 种方法分类准确率比较 (%)

	CCS	CCS (without similar information)	Euclidean distance
Experiment 1	92.8	68.0	37.0
Experiment 2	85.0	65.0	48.3
Experiment 3	89.0	70.3	55.3
Average	88.9	67.8	46.8

Table 5 Comparison of 3 methods' time consumption (ms)

表 **5** 3 种方法时间花费比较 (ms)

	CCS	CCS (without similar information)	Euclidean distance
Experiment 1	24 940	24 611	23 442
Experiment 2	24 903	24 917	23 775
Experiment 3	27 097	25 515	23 916

通过实验可以看到:在时间上,3 种算法相差不大;而引入属性相似信息后的 CCS 算法分类精度明显高于采用欧式空间距离的方法,平均准确率提高了 42.1%.

实验 3. 测试在不同样本数下,CCS 算法与采用欧式距离的分类算法在性能上的差别.实验中,我们将样本数设置为 100~1000,测试数据数目设置为 80,具体实验结果如图 3 所示,其中:横坐标为样本数;纵坐标为耗费时间;单位为 ms.

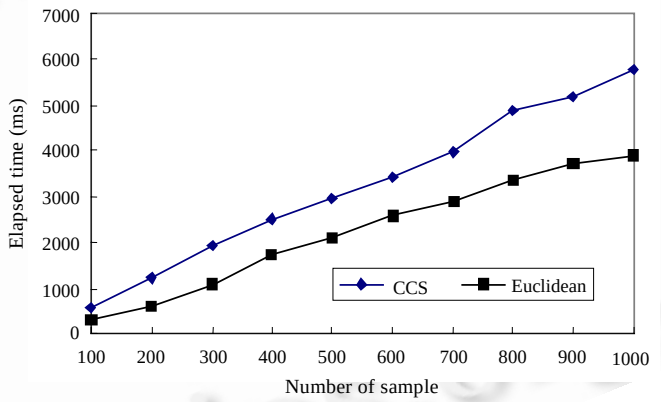

Fig.3 Capability comparison of CACS algorithm and classification algorithm based on Euclidean distance

图 3 CCS 算法与基于欧式距离的分类算法性能比较

由图 3 可知,与欧式距离一样,CCS 算法的性能与样本数基本呈线性关系,且二者在时间花费上相差很小.

实验 4. 比较 SADA 算法与 Floyd 算法在求任意两点最短路径距离上的性能差别.在实验中,相似规则数(即两点的路径)分别设置为 400~3200 条,矩阵的大小为 400×400,实验结果见表 6.

Table 6 Comparison of SADA and Floyd algorithm

表 **6** SADA 算法与 Floyd 算法比较

Order	Rules	SADA algorithm time (ms)	Floyd algorithm time (ms)
1	400	070	4 426
2	800	270	4 476
3	1 200	331	4 416
4	1 600	340	4 427
5	2 000	391	4 486
6	2 400	400	4 437
7	2 800	431	4 446
8	3 200	441	4 446

通过实验发现:当规则数较小时,SADA 算法与 Floyd 算法相比具有较大优势,速度为其 10~63 倍.而 Floyd 算法花费时间与规则数的多少基本上没有关系,这主要与算法本身的设计有关.

SADA 算法性能与规则数的关系如图 4 所示,其中:横坐标为规则数;纵坐标为 SADA 算法耗费时间;单位为 ms.

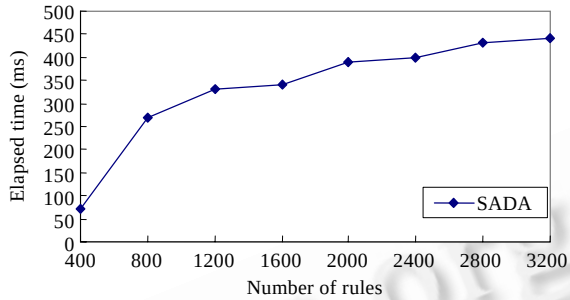

Fig.4 The relationship between capability of SADA algorithm and numbers of rules

图 4 SADA 算法性能与规则数的关系

由图 4 可以看出,SADA 算法耗费时间相对于规则数增长较为缓慢.通过以上实验可以看出 SADA 算法在计算任意两点距离上的优势.

5 结论及下一步工作

利用属性之间先验的概念相似信息,本文采用 SADA 算法求得任意属性矢量之间的距离,并根据矢量之间的距离推导出任意元组矢量相关度的计算公式.通过相关分析,说明该公式很好地表达了元组矢量之间的相似程度.利用这个公式,本文相应地提出了一种基于相关度的分类算法:CCS.通过实验对比采用欧氏距离的分类方法,证明了该算法具有较高的分类精度.

目前,该方法仅考虑属性值为正和概念相似为正相关的情况.下一步,我们拟对该方法在属性值为负及存在负相关的情况进行扩展;另一方面,为了进一步提高分类精度,拟通过遗传算法来优化属性权重及投影公式的参数.

本文提出了一种计算任意元组矢量相关度的公式.将相关度看作元组矢量的距离,可以很自然地将本文的工作用于数据聚类或相似查询.

References:

[1] Indyk P, Motwani R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In: Jeffrey V, ed. Proc. of the 30th Annual ACM Symp. on Theory of Computing. New York: ACM Press, 1998. 604−613.

[2] Kleinberg J. Two algorithms for nearest-neighbor search in high dimensions. In: Leighton FT, Borodin A, eds. Proc. of the 27th Annual ACM Symp. on Theory of Computing. New York: ACM Press, 1997. 599−608.

[3] Kushilevitz E, Ostrovsky R, Rabani Y. Efficient search for approximate nearest neighbor in high dimensional spaces. SIAM Journal on Computing, 2000,30(2):451−474.

[4] Aggarwal C. Hierarchical subspace sampling: A unified framework for high dimensional data reduction, selectivity estimation, and nearest neighbor search. In: Michael J, ed. Proc. of the ACM SIGMOD Conf. New York: ACM Press, 2002. 452−463.

[5] Berchtold S, Keim D, Kriegel HP. The X-tree: An index structure for high dimensional data. In: Vijayaraman TM, Buchmann AP, Mohan C, Sarda NL, eds. Proc. of the 22nd Int'l Conf. on Very Large Databases. San Francisco: ACM Press, 1996, 28−39.

[6] Beyer K, Goldstein J, Ramakrishnan R, Shaft U. When is nearest neighbors meaningful? In: Beeri C, Buneman P, eds. Proc. of the 7th Int'l Conf. on Database Theory. Jerusalem: Springer-Verlag, 1999. 217−235.

[7] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing. In: Atkinson MP, Orlowska ME, Valduriez P, Zdonik SB, Brodie ML, eds. Proc. of the 25th Int'l Conf. on Very Large Databases. San Francisco: ACM Press, 1999. 518−529.

[8] Goldstein J, Ramakrishnan R. Contrast plots and P-sphere trees: Space vs. time in nearest neighbour searches. In: Abbadi AE,

Brodie ML, Chakravarthy S, Dayal U, Kamel N, Schlageter G, Whang KY, eds. Proc. of the 26th Int'l Conf. on Very Large Databases. San Francisco: ACM Press, 2000. 429−440.

[9] White D, Jain R. Similarity indexing with the SS-tree. In: Su SYW, ed. Proc. of the 12th Int'l Conf. on Data Engineering. New Orleans: IEEE Computer Society, 1996. 516−523.

[10] Dwork C, Kumar R, Naor M, Sivakumar D. Rank aggregation methods for the web. In: Shen VY, Saito N, Lyu MR, Zurko ME, eds. Proc. of the 10th Int'l World Wide Web Conf. New York: ACM Press, 2001. 613−622.

[11] Pettie S, Ramachandran V. A shortest path algorithm for real-weighted undirected graphs. SIAM Journal on Computing, 2005, 34(6):1398−1431.

[12] Han Y. Improved algorithm for all pairs shortest paths. Information Processing Letters, 2004,91(5):245−250.

[13] Pettie S, Ramachandran V, Sridhar S. Experimental evaluation of a new shortest path algorithm. In: Mount D, Stein C, eds. Proc. of the 4th ALENEX. London: Springer-Verlag, 2002. 126−142.

[14] Peng J, Tang CJ, Zeng T, Qiao SJ, Yong XJ. A Chinese traditional medicine prescription effect reduction algorithm based on artificial neural network and property distance matrix. Journal of Sichuan University (Engineering Science Edition), 2006,38(1): 92−97 (in Chinese with English abstract).

附中文参考文献:

[14] 彭京,唐常杰,曾涛,乔少杰,雍小嘉.基于神经网络和属性距离矩阵的中药方剂功效归约算法.四川大学学报(工程科学版),2006, 38(1):92−97.

彭京(1973－),男,四川成都人,博士,高级工程师,主要研究领域为数据挖掘,进化计算.

李川(1977－),男,博士,主要研究领域为数据库,数据挖掘.

唐常杰(1946－),男,教授,博士生导师,CCF高级会员,主要研究领域为数据库,数据挖掘.

胡建军(1970－),男,博士,主要研究领域为数据库,数据挖掘.

元昌安(1964－),男,博士,教授,主要研究领域为数据库,数据挖掘.