高维数据中的相似性度量算法的改进

邵昌昇,楼 巍,严利民
(上海大学 机电工程与自动化学院,上海 200072)

摘 要:高维数据之间的相似性度量问题是高维空间数据挖掘中所面临的问题之一。为了有效解决高维效应给相似性度量带来的种种问题,首先分析传统相似性度量算法,得出其局限性。再通过对传统度量算法进行改进,提出新的 Close 函数,以弥补传统相似性度量算法应用在高维空间时的不足。提出 Close 函数后,将其与几种传统的相似性度量算法作比较,得出新算法在高维空间相似性度量方面的优越性。文中最后用 Matlab 对该函数做了定量分析,实验证明该函数在高维空间中能有效避免噪声和维灾效应的影响。

关键词:数据挖掘;高维数据;相似性度量
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2011)02-0001-04

Optimization of Algorithm of Similarity Measurement in High-Dimensional Data

SHAO Chang-sheng, LOU Wei, YAN Li-min
(School of Mechanical & Electronics Engineering and Automation, Shanghai University, Shanghai 200072, China)

Abstract: The problem of similarity measurement between high dimensional data is one of the problems high-dimensional data mining faces. In order to solve the problems of high-dimensional similarity measurement, analysis of traditional algorithms are made at first to obtain limitation. A new function Close( ) is presented based on the improvement of traditional algorithm to make up for the inadequate of traditional algorithm used in high-dimensional space. Advantages of the new function are obvious in high-dimensional similarity measurement after the comparison between Close( ) and tradition algorithms are made. Quantitative analysis of function Close( ) is made with Matlab and experiments prove that this function can avoid the affects of noise and the curse of high-dimension.

Key words: data mining; high-dimensional data; similarity-measurement

0 引 言

数据挖掘指的是从海量的数据中提取有价值的、事先未知的知识。随着信息化社会的飞速发展,在各个领域中,每天都有大量的数据会出现,如常见的文档词频数据,交易数据、Web 使用数据、用户评分数据、及多媒体数据等,并且这些数据也正从过去的低维属性向着高维属性转变。由于这种数据存在的普遍性,使得对高维数据挖掘的研究有着非常重要的意义[1]。

最近邻查询问题是高维数据挖掘中经常碰到的一个问题,其意是指在数据集中找到与查询点最相似的对象。而最近邻查询问题常会用到相似性度量函数来描述数据之间的相似程度[2]。在低维空间中,通常采用传统的基于空间距离的相似度来描述数据之间的差异,但由于高维空间存在着稀疏性,空空间现象等特性[3],使得这一传统方法使用在高维空间中时其效果将大大下降,导致最近邻查询结果变得不稳定有时甚至会变得毫无意义[4]。

为了解决高维数据的最近邻查询问题,确保在高维空间中也能像在低维空间中一样得出稳定的查询结果,文中在总结了现有的相似性度量方式的基础上,对原有的一些度量算法进行了改进,使其在高维空间中依然能描述数据间的相似度,得出稳定的查询结果。

1 高维空间的特殊性

通常高维空间具有数据维度高、数据量大、数据分布稀疏等特性,因此在高维空间中进行相似性度量会遇到低维空间中所不曾存在的问题。在高维空间进行相似性度量时一定要考虑到"维灾效应",度量算法也应能有效避免"维灾效应"的影响。

1.1 维灾效应

Bellman 首次提出了"维灾效应"。其本义原是指

收稿日期:2010-06-11;**修回日期**:2010-09-23
基金项目:上海市(科委)"科技创新行动计划"非政府间国际科技合作项目(09530708600)
作者简介:邵昌昇(1986-),男,硕士研究生,研究方向为数据挖掘;楼 巍,副教授,从事数据挖掘的研究。

在一个离散的多维网格上几乎不可能去用蛮力搜索去优化有着很多变量的函数[5]。原因是随着变量(也就是维度)的增加,网格的数目会呈指数级增加。久而久之,人们把在数据分析中遇到的由于变量(属性)过多而引起的所有问题都统称为"维灾效应"。这些问题通常表现为:(1)当维度增加至一定程度后,低维中常用的算法其时间复杂度将大大增加,变的不再适用,比如层次算法[6]。(2)Beyer 等人[3]指出,在高维空间中由于查询点到它的最近邻和最远邻在很多情况下几乎是等距离的,最近邻的概念常常会失去意义[7]。

1.2 高维空间对相似性度量的影响

在高维空间中进行相似性度量,其困难不仅在于算法复杂度上,更重要的是必须要让所得的相似性结果有意义。由于传统的度量算法大多是基于空间距离定义的,因此在高维空间中所得到的结果经常是无意义的[8]。针对这一问题,许多文献中对高维空间中数据的相似性定义进行了修改或者重新定义[9]。例如定义子空间,使得数据在子空间中具有比较高的相似度,或者对高维空间进行维度简约,使得在简约后的高维空间中,可以使用一些传统算法进行相似性度量计算并能保证一定程度的有效性。此外,最近邻距离随着维数的增加呈增加趋势。在给定维数 d 的情况下,包含最近邻点的最近邻距离随着数据量的增加,其下降趋势非常缓慢,这就说明了对一个点进行最近邻查询,需要遍历较大的数据量才能使得最近邻距离相对较小一些[4]。

2 改进的高维相似性度量函数

2.1 传统的相似性度量算法

传统的距离函数有欧几里得距离、切比雪夫距离、曼哈顿距离、明氏距离、加权的明氏距离、马氏距离、余弦度量等[10]。通常根据应用领域的不同,选择不同的距离函数以达到较好的描述效果。而这些距离函数通常用于维数不高的场合,若将这些距离函数简单地运用于高维数据中,得出的结论经常是无意义的。具体如下:

$$\text{欧氏距离:}\ D^{(2)}(x,y)=\sqrt{\sum_{i=1}^{d}(x_i-y_i)^2} \tag{1}$$

欧氏距离是最常用的距离度量函数,在低维空间中有着良好的性能,具有空间旋转不变性的特点。

$$\text{切比雪夫距离:}\ D^{(\infty)}(x,y)=\max_{i=1}^{d}|x_i-y_i| \tag{2}$$

$$\text{曼哈顿距离:}\ D^{(1)}(x,y)=\sum_{i=1}^{d}|x_i-y_i| \tag{3}$$

$$\text{明氏距离:}\ D^{(p)}(x,y)=\left[\sum_{i=1}^{d}|x_i-y_i|^p\right]^{1/p} \tag{4}$$

明氏距离是欧氏距离、切比雪夫距离和曼哈顿距离的一种泛化表示,同时也是几何问题的标准度量。当 $p=1$ 时,就是曼哈顿距离,$p=2$ 时,是欧氏距离,$p=\infty$ 时,就变为切比雪夫距离。

加权的明氏距离:

$$D^{(p)}(x,y)=\left[\sum_{i=1}^{d}w_i|x_i-y_i|^p\right]^{1/p} \tag{5}$$

马氏距离:

$$D(x,y)=|\det\boldsymbol{C}|^{1/d}(x-y)^{T}\boldsymbol{C}^{-1}(x-y) \tag{6}$$

其中,$\boldsymbol{C}$ 为协方差矩阵,若 $\boldsymbol{C}$ 为单位阵 $\boldsymbol{I}$,那么马氏就退化为平方欧氏距离。

在信息检索领域,通常会用两个向量夹角的余弦作为相似性度量,也就是 Cosine 度量,其表达式如式(7)所示:

$$\cos(\boldsymbol{a},\boldsymbol{b})=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|_2\cdot\|\boldsymbol{b}\|_2} \tag{7}$$

Cosine 度量的主要特性是相似度不依赖于向量的幅值,即对于 $\lambda>0$,有:

$$\cos(\lambda\boldsymbol{a},\boldsymbol{b})=\cos(\boldsymbol{a},\boldsymbol{b}) \tag{8}$$

Cosine 度量的优点表现在对高维数据中空值现象的处理,而在高维数据中经常会存在大量的空值。在使用 Cosine 值进行度量时,并没有把所有维度全部考虑进去,当且仅当两个向量中都有非空值时才认为该维对相似度有贡献。若两个向量其中有一个向量为空值或两者都为空值的维,经过点乘以后分子都变成了0,也就不会对相似度造成影响。

2.2 改进的高维空间相似性度量函数

如文献[3]所述,在高维空间中采用传统的基于距离的相似性度量会得到最近点与最远点几乎等距的结论,这种情况下的相似性度量问题得出的结果是没有意义的。Hinneburg 等提出了投影最近邻的概念[11],并认为重新对高维空间中最近邻的含义进行定义是解决这一问题的方法。采用这种方法时,将不再平等地看待所有的维,而是使用一个质量准则,根据查询点选择一些相关的维度,并且只对这些被选择的维进行最近邻查询。这种方法的优点是仍旧可以采用原来的传统距离度量函数,不过结果是将原来的在全空间中的查询变为了在子空间中的查询。该方法的缺点在于难以确定维度选择的质量准则。

而文中从另一角度出发,针对传统距离度量在高维空间中的缺陷,修改原来的距离度量函数,来缓解和消除高维带来的影响。

设 $X=(x_1,x_2,\cdots,x_d)$ 和 $Y=(y_1,y_2,\cdots,y_d)$ 是 d 维空间中的两个点,定义接近函数如公式(9)所示:

$$\text{Close}(X,Y)=\frac{\sum_{i=1}^{d}e^{-|x_i-y_i|}}{d} \tag{9}$$

这是一个表示两数据之间相似度的函数,有如下

性质:

(1)函数最小值为0,表示 X 和 Y 在每个维度上的差都接近于无穷大,此时的 X 和 Y 相似度最小。

(2)函数最大值为1,表示 X 和 Y 在所有维度的值都相等,此时 X 和 Y 在 d 维空间上是相重合的,因此相似度最大。

Close 函数与传统的基于距离的函数相比,在该函数中占主要地位的是那些值相近的维度,而那些值相差很远的维度几乎被略去,两个数据中,彼此相接近的维度越多,相似性越大,这与判定事物是否相似的逻辑习惯相吻合。

在实际情况中,高维空间中经常会存在两个非常相似的数据,它们在大多数维度上的值都非常接近,只是在少数维上,这两个数据的值相差很大。这些值相差很大的维度在传统距离相似性度量函数中占了主导作用。因此,两个数据的相似信息就被淹没在这些少数维中,而这些维又多数是噪声信息[12]。文中提出的接近函数就可以有效避免这些噪声的干扰。

为了更好地适应不同的数据集,类似于传统距离函数的加权算子,可以对接近函数稍加修改,使其成为加权的接近函数:

$$\text{Close}(X,Y)=\frac{\sum_{i=1}^{d} w_i e^{-|x_i-y_i|}}{d} \tag{10}$$

这里 w_i 的取值为0到1之间,w_i 越大,说明第 i 维上的数据越重要,对数据相似度的贡献也越大。在高维数据集中,也常常会出现属性值为空值的情况,对两个 d 维对象进行比较,如果两个数据在某个属性 i 上至少有一个对象为空值,则可以认为 $|x_i-y_i|$ 为无穷大,此时该维度对相似度的贡献为0,即 w_i 取0。

3 改进的相似性度量函数与传统相似性度量函数的比较

3.1 与 Minkowsky 距离的比较

Minkowsky 距离是几何问题的标准度量,也是低维空间中最常用的相似性度量。通过与 Minkowsky 距离的比较,就可以看出 Close 函数相对于传统距离函数的优点。然而这些传统距离函数的值都是在0到无穷大之间,其值越小说明两个数据越相似。为了将其与 Close 函数做比较,可以通过一个单调递减的函数将距离函数转化为一种相似性度量,使其值在0到1之间,值越大说明数据越相似。

令 Minkowsky 距离 $D^{(p)}(x,y)=\left[\sum_{i=1}^{d}|x_i-y_i|^p\right]^{1/p}$ 中的 $p=2$,即欧氏距离,取其倒数,可在分母上加1从而避免分母为0的情况,即:

$$S(x,y)=\frac{1}{1+D^{(2)}(x,y)} \tag{11}$$

这种先计算欧氏距离,然后再将所得的欧式距离转换成相似性度量是传统相似性度量简单的扩展到高维空间后的结果,因此将它难以克服高维度带来的问题,即由于受到高维特性的影响,两个数据的相似性信息会被少数几个值差别较大的维所淹没。例如:设有 a、b、c 三个10维空间中的点,其每个维的值都列于表1中。

表1　10维空间中的三条数据记录

点	维1	维2	维3	维4	维5	维6	维7	维8	维9	维10
a	1	1	1	1	1	1	1	1	1	1
b	1	1	1	1	1	1	1	1	1	1000
c	5	5	5	5	5	5	5	5	5	5

如表1所示,a 和 b 在前9个维上的值是完全一样的,但是在第10维上却有着极大的差异,而 a 和 c 在每个维度上都有着5倍的差距。主观上通常会认为 a 和 b 更为接近。现用2种方法计算的相似度如下:

$S(a,b)=0.001$　$\text{Close}(a,b)\approx 0.9$

$S(a,c)\approx 0.073$　$\text{Close}(a,c)\approx 0.183$

由此可见,$S(a,b)$ 没有反映出 a,b 之间存在的相似性,而 Close 函数与我们的主观认识是一致的。

3.2 与 Cosine 度量的比较

文中提出的 Close 函数虽然不具备 Cosine 度量的不依赖于幅值的特性,但是在处理空值问题上两者有着相似之处。在计算 Close 函数时,只有那些两个向量中都是非空值的维才被考虑,若其中有一维是空值,则 w_i 可以取0,因为此维对相似度的贡献为0。Close 函数与 Cosine 度量的不同之处在于这两种度量方式所反映的相似性本质不同。Cosine 相似度反映的是两个向量的方向差异,若向量 $\boldsymbol{a}$ 在各个维上的值与向量 $\boldsymbol{b}$ 均相等或者是向量 $\boldsymbol{b}$ 的 n 倍,则向量 $\boldsymbol{a}$ 与向量 $\boldsymbol{b}$ 方向相同,因此也最相似。而 Close 函数的值表示的是两个向量在各个维度上的差异程度,差值越小,则两个向量越相似。举例如下(见表2),其中"0"表示空值。

表2　10维空间中的4条数据记录

向量	维1	维2	维3	维4	维5	维6	维7	维8	维9	维10
$\boldsymbol{a}$	1	2	3	2	1	3	5	4	1	3
$\boldsymbol{b}$	1	2	0	2	0	0	0	4	0	3
$\boldsymbol{c}$	1	0	3	0	1	3	5	0	1	3
$\boldsymbol{d}$	5	0	15	0	5	15	25	0	5	15

计算的结果如表3所示。可以看到 Cosine 度量和 Close 函数在对一个向量进行自身比较时,相似度值都为1,表示最相似。如 $\text{Cosine}(\boldsymbol{a},\boldsymbol{a})$,$\text{Cosine}(\boldsymbol{b},\boldsymbol{b})$,$\text{Close}(\boldsymbol{a},\boldsymbol{a})$,$\text{Close}(\mathbf{b},\mathbf{b})$。当 $\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}$ 之间进行相互

比较时，Close 度量与 Cosine 度量得到的值虽然不同，但基本能表现出相同的相似度变化趋势。而差异较大的是比较 c、d 和 a、d 时产生的结果。由于 d 的幅值是 a 的5倍，故 a 与 d 是同方向的向量，因此 Cosine(c,d) = 1，Cosine(a,c) = Cosine(a,d) = 0.85。而 Close(c,d) = Close(a,d) = 0.008。由此可见，两者 Close 函数依赖于向量的幅值，而 Cosine 度量依赖于向量的方向。其实如果把 d 的幅值缩小到原来的 1/5，Close 函数与 Cosine 度量所反映的相似性程度还是基本一致的。

表3 Close 函数与 Cosine 度量的结果比较

	a,a	b,b	a,b	a,c	b,c	c,d	a,d
Cosine	1	1	0.67	0.85	0.04	1	0.85
Close	1	1	0.5	0.7	0.2	0.008	0.008

3.3 Close 函数在高维空间中的稳定性

文献[10]中将 $v = \frac{D_{maxd} - D_{mind}}{D_{mind}}$，即最远距离和最近距离之差与最近距离的比值作为衡量距离函数是否有意义的标准，若 v 值随着 d 的增加以常数概率趋近于0，那么说明该函数随着维数的增加，其最远与最近距离变得几乎相同，而这时的最近邻查询是不稳定的。文中也将以此值作为评判标准，利用 Matlab 随机生成300条50维，100维，150维，200维，300维的数据，得出文中的 Close 函数相对应的 v 值随着维度 d 变化的曲线图，见图1。

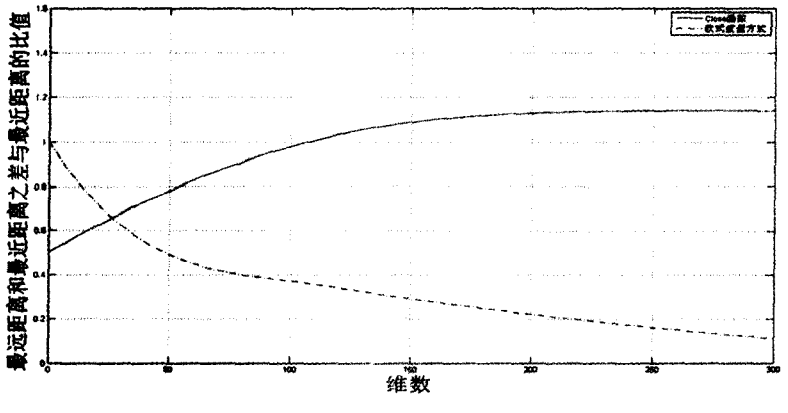

图1 最远距离和最近距离之差与最近距离的比值随维数变化曲线

图1表明，在运用文中提出的度量方法计算 Matlab 随机产生的数据所得到的 v 值时，其结果并没有随着数据维度的增高而降低，而简单的将欧氏距离公式运用于高维数据相似性计算时，其所得的 v 值将随着数据维数的增高急剧减小。因此，从区分最近邻与最远邻的能力上来讲，文中提出的高维度量方式要明显优于传统度量方式，它从一定程度上减小了“高维”所带来的影响，同时也保证了高维空间中数据的相似性度量以及最近邻查询的稳定性。

4 结束语

高维空间中数据之间的相似性度量已经被越来越多的领域所关注与重视，其研究价值也越来越大。而造成“高维效应”的主要原因一方面在于高维空间的特性以及由此产生的大量噪声数据，另一方面是由于将传统的距离度量由低维简单扩充到高维所引起的不足。

针对这一问题，文中根据原有的距离度量函数进行改进，提出了新的接近函数用来描述高维空间中两数据点之间的相似程度。经实验表明，该函数可以有效避免噪声影响，运用于高维数据空间中。

参考文献：

[1] 陈慧萍，王 煜，王建东. 高维数据挖掘算法的研究与进展[J]. 计算机工程与应用，2006(24)：170-173.

[2] 杨 莉. 高维数据的可视化和快速聚类算法[J]. 计算机科学，2006，33(11)：132-138.

[3] Beyer K，Goldstein J，Rishnan R R. When is nearest neighbor meaningful[C]//ICDT Conferenee Proceedings. Jerusalem，Israel：[s. n.]，1999：217-235.

[4] 汪祖媛，庄镇泉，王煦法. 逐维聚类的相似度索引算法[J]. 计算机研究与发展，2004，41(6)：1003-1009.

[5] Bellman R. Adaptive Control Processes：A Guided Tour[M]. Princetcn：Princeton University Press，1961.

[6] 项响琴，汪彩梅. 基于聚类高维空间算法的离群数据挖掘技术研究[J]. 计算机技术与发展，2010，20(1)：124-127.

[7] Berchtold S，Ertl B，Keim D A，et al. Fast Nearest Neighbor Search in High-Dimensional Space[C]//14th International Conference on Data Engineering(ICDE'98). [s. l.]：[s. n.]，1998：209-215.

[8] Ferhatosmanoglu H，Tuncel E，Agrawal D，et al. High dimensional nearest neighbor searching[J]. Information Systems，2006(31)：512-540.

[9] 余肖生，周 宁，张芳芳. 高维数据可视化方法研究[J]. 情报科学，2007，25(1)：117-120.

[10] 贺 玲，吴玲达，蔡益朝. 高维空间中数据的相似性度量[J]. 数学的实践与认识，2006，36(9)：189-194.

[11] Hinneburg A，Aggarwal C C，Keim D. What's the Nearest Neighbor in High Dimensional Spaces[C]//Proc. of the 26th International Conference on VLDB. Cairo，EgyPt：[s. n.]，2000.

[12] 杨风召. 高维数据挖掘中若干关键问题的研究[D]. 上海：复旦大学，2003.