

文章编号:1001-9081(2006)10-2389-04

基于密度梯度的聚类算法研究

陈治平¹, 王雷¹, 李志成²

(1. 福建工程学院 计算机与信息科学系,福建 福州 350014;

2. 创智信息技术有限公司,广东 深圳 518057)

(Chenzhiping05@tsinghua.org.cn)

摘要:针对聚类中不规则形状数据点分布的处理难题,提出了一种基于密度梯度的聚类算法(CDG)。算法通过分析数据样本及其周边的点密度变化情况,选择沿密度变化大的方向寻找不动点,从而获取原始聚类中心,再利用类间边界点的分布情况对小类进行合并。实验结果表明,新算法较基于密度的带噪声数据应用的空间聚类方法(DBSCAN)具有更好的聚类性能。

关键词:聚类;模式分类;数据挖掘

中图分类号: TP181 **文献标识码:**A

Research of clustering algorithm based on density gradient

CHEN Zhi-ping¹, WANG Lei¹, LI Zhi-cheng²

(1. Department of Computer and Information Science, Fujian University of Technology, Fuzhou Fujian 350014, China;

2. Chuangzhi Information Technology Co. Ltd, Shenzhen Guangdong 518057, China)

Abstract: In order to solve difficult problems in clustering with irregularly distributed data set, a new clustering algorithm based on density gradient was provided. By analyzing the changing density of data sample and its neighbors, the algorithm searched points with the maximum density and took them as original centers of clusters. Then it combined some smaller clusters into larger ones according to the distribution of border points between clusters. Experimental results show that the new algorithm has better performance than Density Based Spatial Clustering of Applications with Noise(DBSCAN).

Key words: clustering; pattern classification; data mining

0 引言

随着电子信息的爆炸式增长,描述数据维度的增加,使人们无法对相似数据进行有效的区分并得到有效的判断依据,因此如何对目前海量信息进行有效地组织成为一项非常重要的课题。聚类方法通过建立数学模型,根据数据相似性将数据库划分为不同的部分,使得类内数据尽可能相似,类间数据差异尽可能大,从而成为数据挖掘与模式分类应用的重要工具,并得到广泛的研究与应用^[1,2]。由于目前常用的聚类方法采用欧氏距离、均值、方差概念,从而导致类的数据样本分布适合于超球形^[3],而对于螺旋线、椭圆甚至不具有任何形状的非规则分布数据集合则比较欠缺。

DBSCAN^[4]是一种基于密度的典型聚类方法,通过引入密度可达的概念,将 ε 邻域内所包含的对象数大于MinPs的点定义为核心点,相邻核心点相互直接可达,所有相互可达的点形成一个聚类,而不属于任一类的点视为噪声数据,从而有效地解决了非规则样本数据分布的问题,并得到较为广泛的研究。但由于算法采用全局变量 ε 和MinPs,导致不同的参数产生不同的聚类结果。同时 ε 和MinPs值的选择确定了样本的全局的密度分布,对于部分密度小的聚类可能被作为噪声数据进行处理;而处于两聚类边缘的点若存在该点的密度比较大的情况时容易造成单连通的情形。由于这些缺陷,制约了算法的发展。针对数据集合难以给出一个全局的密度参

数值,文献[5]提出了一种 OPTICS 算法,通过记录一系列的有序密度参数值实现对 DBSCAN 算法的扩展。在 DBSCAN 的基础上,文献[6]提出了能处理不同属性的 GDBSCAN 算法并应用在天文、生物、地球科学、几何等实际的问题处理中。针对 DBSACN 算法只能发现密度近似的簇,文献[7]提出了一种 PDBSCAN 算法,通过对数据空间预先进行分区,然后对不同分区采用不同的参数值进行聚类,从而在一定程度上解决了样本要求密度近似的问题。文献[8]针对输入参数敏感的缺点,提出了 DBSCANCC 的方法,通过利用簇间的记录信息纠正所有的错误分析结果。文献[9]结合模糊距离度量将 DBSACN 方法应用于模糊信息的处理。由于这些方法只是对某一具体问题进行了研究,没有从根本上脱离 DBSACN 方法的基本思想,因此该方法所存在的问题依然存在。

针对非规则数据分布的聚类问题,本文提出了一种基于密度梯度的聚类算法(Clustering based on Density Gradient, CDG)。算法通过计算点的密度,利用邻近点的点密度变化趋势,寻找不动点,由不动点以及其相邻的点构成最小聚类,通过计算类间的边界点的分布情况,再对小类进行合并达到最终的聚类要求。由于算法利用点密度的变化情况,因此对于密度分布不均匀的样本集合等具有较好的应用效果,并且不受噪声数据的影响。实验结果表明新算法对数据的聚类效果明显优于 DBSACN 算法。

收稿日期:2006-04-25;修订日期:2006-06-12

基金项目:福建省自然科学基金资助项目(A0510024);福建省青年基金(2005J051);广东省关键领域重点突破项目(2005A10207003)

作者简介:陈治平(1971-),男,湖南安化人,副教授,博士,主要研究方向:机器学习; 王雷(1973-),男,湖南长沙人,副教授,博士,主要研究方向:计算机网络、机器学习; 李志成(1976-),男,湖南衡阳人,工程师,主要研究方向:数据挖掘。

1 DBSCAN 算法思想与分析

尽管 DBSCAN 可以对非规则的样本分布数据有效地进行聚类,但存在以下的一些缺陷:

1) 不同参数设置对聚类结果的影响大

由于算法使用 ε 和 MinPs 两个全局参数,不同的参数设置都将导致核心点集合发生变化,从而使得参数设置对聚类结果具有非常大的影响。而文献[4]中仅通过设置不同的数值获取聚类数目与参数的变化情况,以曲线拐点处得到的数据作为最终聚类结果。

2) 数据分布要求存在一个统一的密度分布

由于 DBSCAN 使用 ε 和 MinPs 两全局参数,实际上要求所有的聚类满足 ε 邻域内至少包含 MinPs 个点,从而使得所有的类的密度不小于该最小密度,而对于具有不同密度的聚类而言则难以区分。尽管 OPTICS 使用一系列的 ε 以记录不同聚类的密度情况,但仍然没有摆脱 DBSCAN 算法的影响。

3) 噪声数据处理不当

DBSCAN 算法将不满足密度分布的点全部视为噪声数据,以降低噪声数据对聚类的影响,但却使许多合法数据变为噪声数据而不能正确聚类。

除此以外,聚类过程中还存在聚类与聚类相互重叠、两聚类由于单连通而导致合并的现象等。因此针对聚类数据的分布情况,我们提出了一种基于密度梯度变化的聚类算法,利用样本数据分布中类数据分布具有密度的梯度变化情况(类中心的密度最大,而类边界的密度小)将数据划分为原始的局部聚类,然后通过计算类间的边界点的分布情况,再对小类进行合并达到最终的聚类要求,从而实现了对非规则样本分布数据的有效聚类。

2 基于密度梯度变化的聚类算法(CDG)

2.1 CDG 算法模型

定义 1 点邻域:以某一数据样本点 O 为中心,距离该点 O 的 d 个最近邻点的距离为半径所包含的点集。记为: $N(O, d)$ 。

定义 2 点密度:以某一数据样本点 O 为中心,距离该点 O 的 d 个最近邻点的平均距离作为该点的密度。记为 $D(O, d)$ 。

在密度的计算中,由于采用距离作为度量,因此,样本点 O 的密度越大,则 $D(O, d)$ 的数值越小,密度越小,则 $D(O, d)$ 的数值越大。在下文中除非特别说明,则采用密度的概念。

定义 3 不动点:某数据样本 O 的分布密度在以该点为中心,与距离 O 点的第 d 个最近邻点的距离为半径所辖的邻域范围内达到最大,称该点为不动点。记为: $Kernel(O, d)$ 。

引理 不动点的数目随邻域的增加而减少

显然易见,随着邻域的增加,密度分布较大的不动点所组成的聚类所包含的点数不断增加,而密度分布较小的类则逐渐减小,最终被大类所合并。特别地,当邻域达到某一上限时,所有的点构成一个类;当邻域等于 0 时,每个点都构成一个不动点,形成一个类。

由于 d 的大小决定初始聚类后类数的多少,因此选择 d 值的变化范围成为 DBSCAN 算法中求解 ε 的另一对应的问题,但与之不同的是 d 越小,对应的原始聚类数目就越多,并且这些原始聚类的中心包含了 d 取大值时所划分的原始聚类的类中心,因此聚类情况受 d 的影响没有 ε 那么敏感。在实验

中, d 的取值范围为 $(\sqrt{N/c} \sim \sqrt{N/2})$ 可以达到最佳效果, N 为数据点的数目, c 为用户期望的聚类数。

定义 4 边界点:若某类 C_i 中的数据点 O 的 d 近邻域内的点分属于两个或两个以上的类,则这样的点称为类 C_i 的边界点。同属于 C_i 类的所有边界点所构成的集合称为 C_i 类的边界点集,记为 $Boader(C_i)$ 。

定义 5 两聚类的合并程度函数 $H(C_1, C_2)$:定义函数作为判断两类进行合并的依据。在实验中我们采用边界点与类中所包含点数的熵值进行计算,具体公式如下所示:

$$H(C_1, C_2) = \frac{c_1}{c} \log \frac{c_1}{c} + \frac{c_2}{c} \log \frac{c_2}{c}$$

其中 c_1 为类 C_1 所包含的点的数目, c_2 为类 C_2 所包含的点的数目, c 为 C_1 与 C_2 所包含的边界点数,若边界点数为 0, 则定义其合并程度函数值为 $+\infty$ 。

2.2 基于密度梯度变化的聚类算法

根据上述的算法模型,可以得到相应的聚类算法如下:

(1) 初始化,计算密度分布

计算各点的 d 近邻,以其 d 近邻的平均距离作为该点的密度。

(2) 获取不动点,获得原始聚类

随机选择某一未分类的点 O , 获取其 d 近邻中密度最大的点 P , 根据若该点 P 的密度与点 O 的密度的比较分两种情况比较:

$<$: P 点密度小于点 O 的密度,或 $D(P, d)$ 的数值大于 $D(O, d)$ 的数值,则点 O 为一不动点,并赋予一个新的类别号;

\geq : 则根据点 P 的聚类情况进行判断,若该点的类号已确定,则将该聚类号作为该点 O 的聚类号;否则计算以点 P 为始点的不动点。

当找到相应的不动点后给该路径下的所有点赋予该不动点的类别号;继续循环直到所有的点都被分类。

(3) 调整边界点

选择边界点的 d 个近邻点的类别号中采用多数表决的方式作为该边界点的类别号:

$$\arg \max_{C_i \in |C|} \sum_{i=1}^d \delta(c_i, f(x_i)) \rightarrow f(x)$$

$$\delta(a, b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$$

其中, $f(x)$ 为 x 所属的聚类号。

(4) 计算边界点对各聚类的分布情况进行合并

计算各边界点的分布情况:统计每类中的边界点的分布情况。

若 $\exists p \in C_j, O \in Boader(C_i)$, 且 $p \in N(O, d)$, 则:

$Num(NearCluster(C_i, C_j)) + 1 \rightarrow Num(NearCluster(C_i, C_j))$

其中, $Num(NearCluster(C_i, C_j))$ 代表 C_i, C_j 类间边界点的数目。

合并的终止条件:

1) 若指定聚类数

按照合并程度函数的大小排序,选择最小函数值的两类进行合并,合并后当满足要求或不存在任何两类具有共同边界点时合并终止。

2) 若未指定聚类数

按照合并程度函数的大小排序,当最小合并函数值大于给定的 Threshold 时,算法终止,否则选择最小函数值的两类

进行合并。

在合并过程中,对于边界点的处理:

- $\forall p, p \in \{C_i\}$, 若 $Boader(p, C_{old1}) \neq \phi$ 或 $Boader(p, C_{old2}) \neq \phi$, 且 $p \neq C_{old1}, p \neq C_{old2}$, 则:
- 若 $Boader(p, C_{old1}) \neq \phi$ 且 $Boader(p, C_{old2}) \neq \phi$, 则 $Boader(p, C_{old1}) + Boader(p, C_{old2}) \rightarrow Boader(p, C_{new})$;
 - 若 $Boader(p, C_{old1}) = \phi$ 且 $Boader(p, C_{old2}) \neq \phi$, 则 $Boader(p, C_{old2}) \rightarrow Boader(p, C_{new})$;
 - 若 $Boader(p, C_{old1}) \neq \phi$ 且 $Boader(p, C_{old2}) = \phi$, 则 $Boader(p, C_{old1}) \rightarrow Boader(p, C_{new})$;

同时重新计算与其他相关的近邻类的合并程度函数,并重复该步骤直至算法终止。

3 算法复杂度分析

由于需要采用距离计算获取点的近邻,因此,在这一过程中的时间复杂度为 $N * (N - 1)$ (若采用文献[4]所述的 R* 树,则时间复杂度为 $N \sqrt{N}$);而在获取不动点的过程中只需要扫描点的密度,因此,所花费的时间为 $O(N)$;在边界点的调整过程中,由于点的数目小于 \sqrt{N} ,因此其时间开销为 $O(\sqrt{N})$,在对不动点所代表的小类(设为 K 个)进行合并过程中,由于子类的合并不涉及到点的计算,而只与类数相关,因此其时间复杂度为 $O(K * (K - 1) + N)$,其中 N 为建立 K 阶距阵时需要对点进行的一次扫描。因此,总的时间复杂度为 $O(N * (N - 1) + N + \sqrt{N} + K * (K - 1) + N)$ 。

而 DBSCAN 算法的时间复杂度因为要计算邻域内的点,首先必须计算任意两点之间的距离,因此在这个过程中所花费的时间代价同样为 $O(N * (N - 1))$;在得到点的距离值后计算每点邻域内的点的密度所花费的时间开销为 $O(N)$,在聚类的结果产生过程中所花费的时间开销为 $O(N)$ 。因此总的时间开销为 $O(N * (N - 1) + N + N)$ 。

从以上的时间分析可以看出,由于时间的计算开销主要在近邻的计算过程中,因此新算法的时间复杂度只是略高于 DBSCAN 算法。

若结合参数的选择过程来看,由于 CDG 算法的 d(d 近邻)比较容易获得,而要达到给定类数的情况则只是一个类的合并过程,因此增加的时间复杂度开销只是 $O(K)$;然而对于 DBSCAN 算法而言,由于每一次试探都对应着整个算法的重复过程,因此每次的时间开销均为 $O(N * (N - 1) + N + N)$ 。因此,从整体来看,CDG 的时间开销比 DBSCAN 的时间开销要小得多。

4 实验分析

实验分两步进行,首先利用 2D 的 CheckBoader 数据集跟踪新算法的分类的准确性并与 DBSCAN 进行比较,然后是针对 4D 的 Iris 数据集进行性能上的实验比较。

4.1 CheckBoader 数据集^[10]

CheckBoader 数据集共包含两种类型的数据(1 000 个点),其中我们选择其第一个属性为 1 的点(图 1 所示,共为 487 个点)作为实验数据的样本点进行实验。从图 1 中可以人为地将数据划分为 8 个类作为判断结果。

首先利用 CDG 算法对该数据集进行实验。根据已知的数据样本数, d 的取值范围为($\sqrt{N/c} \sim \sqrt{N/2}$)。实验中设置

不同的 d 值所得到的不动点及其临近点所构成的初始聚类数如表 1 所示。

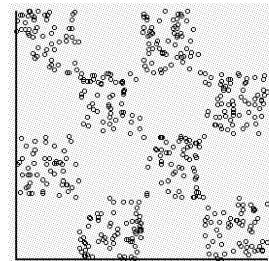


图 1 CheckBoader 数据点所对应的分布图

表 1 CDG 算法使用不同参数所得到的实验结果

d 值	不动点数	
	< 6	> 41
7 ~ 16	33 ~ 15	
17 ~ 27		15 ~ 8
28 ~ 55		8
> 56		< 8

从表 1 中可以看到,初始的聚类数随 d 取值由小到大变化而由多到少进行变化。

在 CDG 的实验中由于选取聚类数 8 作为最终结果,因此在 CDG 的实验过程中采用类的合并条件为第 2 种方法,直接指定聚类数 8 作为算法的终止条件,因此算法的终止与 Threshold 的选取无关。在实验中当 d = 7 ~ 55 时,所得到的聚类情况如图 2(a)所示(d 为 28 ~ 55 时直接得到图 2(a)的结果,图(b)中未被线条所包含的圆点对应为噪声数据)。

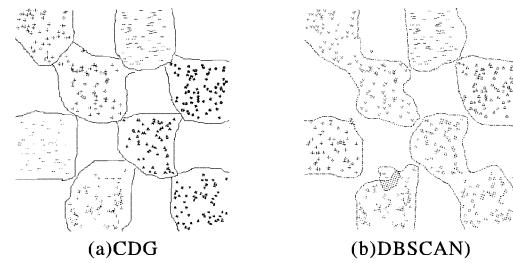


图 2 获得的最终聚类结果

利用 DBSCAN 进行对比实验,由于无法预知最佳的 MinPs 与 ε ,因此通过设定不同的值得到相应的聚类结果如图 3 所示(其中 MinPs 取值为 3、4、5、6)。

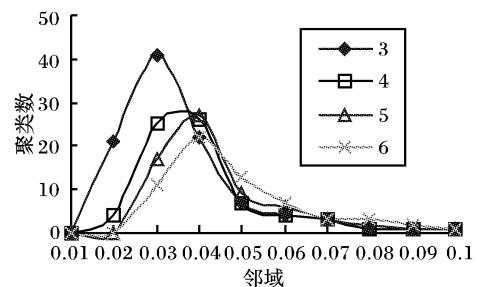


图 3 利用 DBSCAN 算法使用不同参数所得到的实验结果

从图 3 可以看出,当 ε 由 0.01 到 0.1 的变化过程中,聚类数的变化从 0 到大,再由大到小,最终为 1。通过分析,当 ε 小时,所有的点都不能达到所要求的聚类密度要求,因此全部被当成噪声数据处理,因此得到的聚类数为 0,当 ε 为 0.02 时,部分噪声数据变为聚类,随 ε 的增大,噪声数据也不断变为聚类,当达到某一取值时聚类数获得最大;尔后随 ε 的继续增大,相应的聚类逐渐合并从而不断减少,当 ε 达到一定时,

所有数据点合并为一个聚类。

设定 $\text{MinPs} = 3, \varepsilon = 0.05$ 时得到 7 类(8 类的取值难以获得),其聚类结果如图 2(b)所示。在图 2(b)中,由于两类中存在部分数据的密度比较大,导致原本可分为两大类的数据合并成为一类,从而在图 2(b)的实验结果中产生两个单连通的现象;另外由于 DBSCAN 采用密度可达的概念,使实验数据中某些非核心点与多个不同类的核心点存在密度可达的情况从而导致两类交叉重叠的现象产生,如图 2(b)中的阴影部分为两类的重叠区。

4.2 Iris 数据集^[11]

Iris 数据集是 UCL 提供的专用于分类的数据集合。数据集合包含 3 个类(Setosa, Versicolour 与 Virginica),每类 50 个数据样本,其中 Setosa 类是线性可分的,而另外两类是线性不可分的,由于其数据集的异样性,从而被许多聚类算法用来测试聚类的性能。

为了评价聚类的效果,实验中采用 F1 的评价指标体系。

$$F(Cr, Si) = 2 * R(Cr, Si) * P(Cr, Si) / (R(Cr, Si) +$$

$$P(Cr, Si))$$

$$F(Cr) = \max(F(Cr, Si))$$

$$F\text{Score} = \sum_{r=1}^c \frac{n_r}{n} F(C_r)$$

其中 Cr 表示第 r 个聚类; Si 表示第 i 个原始类; $R(Cr, Si)$ 为召回率: Si 类被正确划分的数据与 Si 类数据的比例; $P(Cr, Si)$ 为精度: Cr 聚类结果中包含的 Si 类数据的比例; N_r 表示第 r 个聚类中所包含的样本数。

设定新算法中的 $d = 7$, $\text{Threshold} = 20$; DBSCAN 算法中的 $\text{MinPs} = 3, \varepsilon = 0.1$; 分别得到 3 种类别的聚类结果如表 2 所示。

表 2 利用新算法所得到的分类结果

	类号	Setosa	Versi-colour	Virgi-nica	F(Cr)	F-Score
CDG	1	49	0	0	0.989 899	
	2	1	39	20	0.709 091	0.787 22
	3	0	11	30	0.659 341	
	1	49	0	0	0.989 899	
DBS-CAN	2	0	45	46	0.652 482	0.727 39
	3	1	5	1(3)	0.175 439	

(说明: 括号中的数据为被算法误认为噪声的数据)

从表 2 中可以看出,由于 Setosa 具有线性可分的特性,CDG 算法与 DBSCAN 算法基本上都能正确将该类与其他两类区分开来,召回率达到 98%,错分类数为一个;而对于另两类 Versicolour 与 Virginica 由于其线性不可分,导致其数据存在交叉,因此聚类的效果没有 Setosa 的高,从两种算法的聚类比较可以看出,CDG 所划分的 Cluster2 中包含 Versicolour 的样本数为 39 个,其对应的召回率为 78%,而 Cluster3 中包含 Virginica 30 个,其召回率为 60%;相反,对于 DBSCAN 算法基本上将 Versicolour 与 Virginica 的样本合并为一个大类了,而 Cluster3 的样本数只有 7 个,且每个所包含的样本分属于 3 个原始类,因此聚类的结果不符合实际的要求。其相应的 FScore 小于 CDG 新算法所得到的 FScore 值。同时还存在 Virginica 类中的 3 个样本被错误地判定为噪声数据,从而导致 DBSCAN 算法的聚类性能更差。

同时在参数的设置过程中,CDG 方法的 d 值取值范围为 $\sqrt{N/c} \sim \sqrt{N/2}$,而数据的样本数可以预先得知(150 个),因此 d 的取值范围为 $\sqrt{150/3} \sim \sqrt{150/2}$ 之间,即 7 ~ 9 之间,在

实验中对应 d 值分别为 7,8,9 所得到的原始分类结果一致,而对于 Threshold 取值则需要观察不同的取值以获取最佳的聚类效果,但由于在设定不同的 Threshold 值时只与原始聚类后的 K 阶矩阵的运算相关,因此其时间复杂度为 $O(K^2)$,由于 $K \ll N$,因此每次运算所消耗的时间有限;而对于 DBSCAN 算法,由于不同的 ε 与 MinPs 都会导致整个算法的重新计算,每次调整的时间开销为 $O(N^2)$,因此在获得最佳的参数过程中新算法的额外时间开销要小于 DBSCAN 的时间开销。

结合上述的实验结果与分析表明,CDG 算法的聚类性能要优于 DBSCAN 算法,并能更好地对非规则型数据集进行聚类处理。

5 结语

基于密度的变化情况,本文提出了一种沿密度最大梯度变化的方向寻找聚类中心的聚类算法。算法通过分析点的密度变化情况,获取点附近的最大密度点作为不动点的寻找方向,使算法在最快的时间内获取不动点的信息,并将由该不动点所包含的邻域内的点构成一个最原始的聚类。由于算法不需要全局的密度分布情况,因此对于样本分布不均匀的数据集的聚类过程不会受到密度分布不均的影响,总能正确找到相应的聚类,同时由于算法依赖密度变化情况,因此大量的类内数据的聚类判别不会受小密度的边沿数据或噪声数据的影响,从而可以避免正确数据被错判为噪声数据。实验结果也表明新算法具有较好的聚类性能。

由于新算法的原始聚类数目比较大,可能超过期望的聚类数量,因此算法提供了一种聚类合并的方法,通过调整相应的 Threshold 进行相邻类的合并以达到所需要的聚类结果。

但对于 Threshold 的设置只能通过聚类过程中的随时调整以满足用户的最终的期望,因此如何有效地将最原始的聚类进行有效的自动合并达到最佳的聚类效果将是下一步工作的重点。

参考文献:

- XU R, DONALD WI. Survey of Clustering Algorithm [J]. IEEE Transactions on Neural Networks, 2005, 16(3): 645 - 678.
- NG RT, HAN JW. CLARANS: A Method for Clustering Objects for Spatial Data Mining[J]. IEEE transactions on knowledge and data engineering, 2002, 14 (5): 1003 - 1016.
- KHAN SS, AHMAD A. Cluster center initialization algorithm for K-means clustering[J]. Pattern Recognition Letters, 2004(25): 1293 - 1302.
- ESTER M, KRIEGEL H-P, SANDER J, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[A]. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining[C]. Portland: ACM Press. 1996. 226 - 231.
- ANKERST M, BREUNIG MM, KRIEGEL H-P, et al. OPTICS: Ordering Points to identify the clustering structure[A]. In: Proceedings of the ACM SIGMOD Conference. Philadelphia [C]: ACM Press. 1999. 49 - 60.
- SANDER J, ESTER M, KRIEGEL H-P, et al. Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications[J]. Data Mining and Knowledge Discovery, 1998, 2(2): 169 - 194.

(下转第 2404 页)

事实上,可以由中心极限定理来确定这个独立成分,就像上面提到的,两个独立随机变量的和比其中任何一个随机变量更加接近于高斯分布,也就是说上述表达式蕴含着随机变量,比任何一个独立成分更加靠近高斯分布,只有当这个随机变量恰好等于其中一个独立成分时,y 离高斯分布最远。因此,我们可以求最优值 b 使得 $b^T x$ 非高斯性最大化,则 y 就是一个独立成分。作为衡量随机变量 y 非高斯性的一个传统方法,峰度 $kurt(y)$ 在统计学上表示为:

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2$$

另外一种度量非高斯性的方法是负熵,定义为:

$$J(y) = H(y_{gauss}) - H(y)$$

其中, y_{gauss} 表示服从高斯分布的随机变量,且与 y 具有相同的方差^[13,14]。

2) 极大似然估计方法

极大似然估计方法也是实现独立成分分析的一个很有效的方法,是统计学习的一个最基本的方法,它的应用最为广泛和实用,其基本思想就是,求模型中的参数使得样本的概率分布达到最大。

对于模型 $X = As$ 中 X 的概率分布,可以表示为:

$$P_x(X) = |\det(W)| p_s(s) = |\det(W)| \prod_i p_i(s_i)$$

其中, $W = A^{-1}$ 表示分离矩阵, p_i 表示各个独立成分的概率密度函数。则上式表示为 $W = (w_1, w_2, \dots, w_n)^T$ 和 x 的函数:

$$P_x(X) = |\det(W)| p_s(s) = |\det(W)| \prod_i p_i(w_i^T x)$$

现在假设 T 个观察样本点表示为: $X(1), \dots, X(T)$,且样本点是相互独立的,则得到样本的似然函数:

$$\begin{aligned} P(X) &= \prod_{t=1}^T p(X(t)) \\ &= \prod_{t=1}^T \{|\det(W)| \prod_i p_i(w_i^T X(t))\} \end{aligned}$$

最大化样本的似然函数等价于最大化它的对数似然函数^[14,15]:

$$\begin{aligned} L(W) &= \log\{p(X)\} \\ &= T \log |\det(W)| + \sum_{t=1}^T \sum_{i=1}^n \log\{p_i(w_i^T X(t))\} \end{aligned}$$

3) 基于互信息最小的方法

基于互信息的独立成分分析是很重要的一个内容,这个方法不仅仅局限于独立成分分析中,也在其他的领域中有着广泛的应用。随机向量 $y = (y_1, \dots, y_n)^T$ 的各个元素之间的互信息表示为:

$$I(y_1, \dots, y_n) = \sum_{i=1}^n H(y_i) - H(y)$$

互信息有一个重要的性质: $I(y_1, \dots, y_n) \geq 0$,当且仅当随机向量 y 的各个分量之间相互独立时取零^[13,14]。

(上接第 2392 页)

- [7] ZHOU AY, ZHOU SG, CAO J, et al. approaches for scaling DBSCAN algorithm to large spatial database[J]. Journal of computer science and technology, 2000, 15(06): 509–526.
- [8] 蔡颖琨, 谢昆青, 马修军. 屏蔽了输入参数敏感性的 DBSCAN 改进算法[J]. 北京大学学报(自然科学版), 2004, 40(3): 480–486.
- [9] KRIEGL H-P, PFEIFLE M. Density-based clustering of uncertain data[A]. Proc. 11th Int. Conf. on Knowledge Discovery and Data

3 结语

以上介绍的主要是线性的维规约方法,但现实中非线性情形更具适用性。对于非线性的维规约方法目前也有很多研究和发展,如:主曲线、多维标度法、神经网络、基因遗传算法等^[4],具体领域使用的维规约技术的比较、改进的或综合的维规约算法、新的维规约算法、维规约算法的效率等仍将是研究的热点。

本文借鉴国外已有关于维规约的综述^[3,15],结合国内对各种相关维规约技术研究的成果,从维规约技术的两个方面(属性选择和维变换)概述了常用的维规约技术思想和原理。

参考文献:

- [1] 刘明吉, 王秀峰, 黄亚楼. 数据挖掘中的数据预处理[J]. 计算机科学, 2000, 27(4): 54–57.
- [2] HAN JW, KAMBER M. 数据挖掘概念与技术[M]. 北京: 机械工业出版社, 2001.
- [3] FODOR IK. A Survey of Dimension Reduction Techniques [DB/OL]. Center for Applied Scientific Computing. <http://www.llnl.gov/CASC/sapphire/pubs/148494.pdf>, 2002.
- [4] YE J-P, LI Q, XIONG H, et al. IDR/QR: An Incremental Dimension Reduction Algorithm via QR Decomposition [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(9): 1208–1222.
- [5] 李喆, 王仁超, 褚春超. 数据挖掘中的数据预处理与维度优化[J]. 东北林业大学学报, 2003, 31(3): 70–72.
- [6] 于之虹, 郭忠志. 改进主成分分析法用于暂态稳定评估的输入特征选择[J]. 电力自动化设备, 2003, 23(8): 17–21.
- [7] 万星火, 檀亦丽. 主成分分析原始数据的预处理问题[J]. 中国卫生统计, 2005, 22(5): 327–329.
- [8] 刘直芳, 游志胜, 王运琼. 基于概率主成分分析的人脸识别[J]. 红外与激光工程, 2004, 33(1): 71–75.
- [9] TIPPING ME, BISHOP CM. Probabilistic Principal Component Analysis [Z], 1999.
- [10] 赵广社, 张希仁. 基于主成分分析的支持向量机分类方法研究[J]. 计算机工程与应用, 2004: 37–38.
- [11] 余锦华, 杨维权. 多元统计分析与应用[M]. 广州: 中山大学出版社, 2005.
- [12] 高建芸, 宋德众, 林秀芳. 影响福建热带气旋年季频数的投影寻踪回归模型[J]. 热带气象学报, 2004, 20(4): 443–448.
- [13] 杨竹青, 李勇, 胡德文. 独立成分分析方法综述[J]. 自动化学报, 2002, 28(5): 762–772.
- [14] 钟明军. 独立成分分析算法研究及其在功能核磁共振成像中的应用[DB/OL]. <http://218.69.114.37/wf/~CDDBN/Y665703/PDF/index.htm>, 2006.
- [15] BURA E. Dimension Reduction Techniques: A Review [DB/OL]. The George Washington University. <http://srccs.snu.ac.kr/Workshop/04Statistics/7.pdf>, 2006.

Mining [C]. Chicago, IL, 2005. 672–677.

- [10] HO TK, KLEINBERG EM. Checkerboard dataset[EB/OL]. <http://www.cs.wisc.edu/math-prog/mpml.html>, 1996.
- [11] NEWMAN DJ, HETTICH S, BLAKE CL, et al. UCI Repository of machine learning databases[EB/OL]. <http://www.ics.uci.edu/~mlearn/MLRepository.html>. Irvine, CA: University of California, Department of Information and Computer Science, 1998.