Journal of Computer Applications
计算机应用,2012,32(6):1654-1656,1669
ISSN 1001-9081
CODEN JYIIDU
2012-06-01
http://www.joca.cn

文章编号:1001-9081(2012)06-1654-03
doi:10.3724/SP.J.1087.2012.01654

分类数据的聚类边界检测技术

邱保志,王　波*
(郑州大学 信息工程学院,郑州 450001)
(*通信作者电子邮箱 wangsit@netease.com)

摘　要:随着分类属性数据集的应用越来越广泛,获取含有分类属性数据集的聚类边界的需求也越来越迫切。为了获取聚类的边界,在定义分类数据的边界度和聚类边界的基础上,提出了一种带分类属性数据的聚类边界检测算法——CBORDER。该算法首先利用随机分配初始聚类中心和边界度对类进行划分并获取记录边界点的证据,然后运用证据积累的思想多次执行该过程来获取聚类的边界。实验结果表明,CBORDER 算法能有效地检测出高维分类属性数据集中聚类的边界。

关键词:边界度;证据积累;聚类边界;分类数据
中图分类号:TP311.13　　**文献标志码**:A

Cluster boundary detection technology for categorical data

QIU Bao-zhi, WANG Bo*
(*School of Information Engineering, Zhengzhou University, Zhengzhou Henan* 450001, *China*)

Abstract: With the wide application of categorical-attribute dataset, the demand of obtaining the cluster boundary of categorical-attribute dataset becomes more and more urgent. In order to get cluster boundaries, a categorical-attribute data boundary detection algorithm: CBORDER (Categorical dataset BORDER detection algorithm) was proposed. In this algorithm, firstly, this paper initialized the center of cluster by using random allocation and utilizing boundary-degree to partition clusters; at the same time, the evidence of captured boundary records was got. Then, based on the evidence accumulation, the above procedure was executed repeatedly to acquire the boundaries of clusters at the end. The experimental results demonstrate that CBORDER can effectively detect the boundaries of the high-dimensional categorical data.

Key words: boundary-degree; evidence accumulation; cluster boundary; categorical data

聚类边界检测是对聚类研究领域的扩展,是近几年来数据挖掘领域中一个新的研究方向。聚类边界点具有多个聚类的特征,对它的研究有时比研究聚类更有价值、更重要。边界检测在信息工程、医学、生物学以及计算机科学等领域有着广泛的应用。例如:在医学领域中,可以通过对某种传染疾病已发病病人和病毒携带者之间的各类医学指标的对比分析来获得该疾病发病原因以及通过对病毒携带者和正常人的各类医学指标的对比分析来获得病毒的传播途径的信息,从而从根本上控制该疾病的传播和发病,这里病毒携带者就是具有该疾病特征的聚类边界点,通过单独对这些病毒携带者进行研究,可以获得病毒感染的途径和发病的条件。

1　相关工作

1996 年 Ester 等[1]在聚类算法 DBSCAN(Density-Based Spatial Clustering of Applications with Noise)中提出了聚类边界的概念,但是并没有给出具体的聚类边界检测算法。Xia 等[2]在 2006 年提出了聚类边界检测算法——BORDER(BOundaRy points DetectoR)。算法利用边界点的反向 k-近邻个数远小于聚类内部点的反向 k-近邻个数这一特性,将数据集中反向 k-近邻数目小于一定阈值的数据点作为聚类的边界点。在随后的几年中,Qiu 等在对聚类的边界检测算法的研究上表现较为活跃:分别在 2007 年提出了 BRIM(efficient BoundaRy points detecting algorIthM)算法[3],2008 年提出了 Greb(Grid-entropy-based boundary points detecting algorithm)[4]和 Green(Gravity-Based Boundary Points Detecting Algorithm)[5]等算法,以及最近在 2011 年提出 EDGE(Efficient boundary points Detecting alGorithm based on joint Entropy)[6]等算法。BRIM 算法将记录的邻域划分为正半邻域和负半邻域两部分,利用处于边界的记录正半邻域密度和负半邻域密度相差比较大而处于聚类中心的记录或孤立点的正半邻域密度和负半邻域密度相差较小的性质来获得聚类的边界。Greb 算法将网格技术同信息熵相结合来检测聚类边界。信息熵具有度量网格区域内的记录的分布均匀状况的特性,即信息熵值越小,区域内的记录分布越不均匀,而边界点位于这些分布不均匀的网格区域内,算法通过获取边界网格区域来获取聚类的边界。Green 算法通过计算记录所受到其他记录的引力的合力大小来判断该数据点是否为边界点。由于聚类边界部分分布的不均匀性,处于边界的记录受到的引力的合力要大于内部点所受到的合力。EDGE 算法将网格技术同联合熵结合在一起来检测聚类的边界,由于该算法对联合熵的计算具体到了一条记录中,所以比 Greb 算法具有更高的精度。

同时针对某一特定应用领域的边界检测研究也很活跃。2008 年 Nosovskiy 等[7]针对图像边界提出的 ADACLUS(ADAptive CLUStering)算法,可以自动获取图像的边界。

收稿日期:2011-11-24;**修回日期**:2012-01-20。
基金项目:河南省重点科技攻关项目(112102310073);河南省教育厅自然科学研究计划项目(2009A520028)。
作者简介:邱保志(1964-),男,河南驻马店人,教授,博士,主要研究方向:数据挖掘;　王波(1979-),男,河南安阳人,硕士研究生,主要研究方向:数据挖掘。

ADACLUS 通过在各维度上进行整型编码来提高算法的效率,并且结合整体密度和局部密度来获取数据点的影响函数,最终通过各数据点的影响函数值来判定边界点,算法可以快速、精确地检测二维图像数据的边界。

以上的这些算法对于检测数值属性数据集的聚类边界具有很好的效果,但是这些算法不仅没有提出分类属性数据的边界概念,也没有提出相应的算法。针对这一问题,本文提出了分类属性数据聚类边界的概念,并在此基础上提出一种边界检测算法——CBORDER。

2　CBORDER 算法

D 是一个 m- 维数据集,其属性集为 $\{A_{d_i} \mid d_i = 1,2,\cdots,m\}$,$S_{d_i}$ 为各属性所对应的域,对于任意的 $r \in D$,则有 $r \in \prod_{d_i=1}^{m} S_{d_i}$。

2.1　相关概念

定义 1　集合 S 的补集。对于一个数据集 D,集合 S 为 D 的一个子集,即 $S \subseteq D$。集合 S 的补集 $\overline{S}$ 为:

$$\overline{S} = \{r_i \mid r_i \in D \wedge r_i \notin S\}$$

其中 $i \in \{1,2,\cdots,n\}$。

相似度是聚类中一个非常重要的概念,选择不同的相似性度量方法可以导致不同的聚类结果。本文需要一种既可以度量任意两条记录间的相似度又可以度量一条记录和一个数据集间相似度的相似性度量方法。虽然文献[8]中的使用熵的度量方法能够满足要求,但是使用熵度量相似度的方法在度量任意两条记录的相似度时偏差比较大。使用熵度量相似度对于两条极相似记录和两条极不相似记录结果可能相同,会造成在小数据集上的分类结果不正确。Jaccard 系数虽然可以很好地度量两条记录间的相似度,但是它仅仅适用于任意两条记录间的相似性的度量,为此将 Jaccard 系数相似度度量方法进行进一步的改进使其满足算法的要求。

对于一个 m- 维数据集 D 和 D 内的一条记录 $r = a_1 \times \cdots \times a_m$。$\sup(D,a_{d_ij})$ 表示数据集 D 内所有第 j 个属性值为 a_{d_ij} 的记录的个数,其中 a_{d_ij} 为记录 r 的第 j 个属性值。记录 r 相对于数据集 D 的相似度可定义为:

$$S(r,D) = \sum_{j=1}^{m} \frac{\sup(a_{d_ij})}{|D|} \tag{1}$$

其中:a_{d_ij} 为记录 r 的第 j 个属性值,$|D|$ 为数据集 D 中的记录个数。

由式(1)可以看出,当记录 r 的属性值在数据集上相同的个数越多,表示该记录同数据集越相似,$S(r,D)$ 的值越大;反之也成立。

定义 2　边界度。对于一个数据集 D,C 为数据集上的一个类($C \subseteq D$),$\overline{C}$ 为类 C 在数据集 D 上的补集,r 为数据集 D 上一条记录($r \in D$),则 $B(r,C)$ 为记录 r 在类 C 上的边界度。

$$B(r,C) = S(r,\overline{C})/S(r,C) \tag{2}$$

其中 $S(r,C)$ 和 $S(r,\overline{C})$ 可以为任意满足前面要求的相似性度量函数。由式(2)计算获得的记录 r 在类 C 上的边界度,可以看出:$B(r,C)$ 的值越大表示记录 r 相对于 C 的相似性越小;反之则相似性越大。边界度不仅可以反映出记录相对于所属类的相似程度,而且还可以反映出它相对于其他类的相似接近程度。

定义 3　边界。对于一个数据集 D,C 为数据集上的一个类($C \subseteq D$),$\overline{C}$ 为类 C 在数据集 D 上的补集,r 为类 C 上一条记录($r \in D$),则

$$B(D,C) = \{r \mid r \in C \wedge B(r,C) > \delta\} \tag{3}$$

为类 C 在数据集 D 上的边界,其中 $\delta \in R$。

在定义 3 中 δ 为一个阈值,它一般取小于 1 的实数。由定义 3 可以看出,一个类的边界内的点是那些相对于本身类来说比较接近于其他类的记录。表 1 列出了文献[8]中所使用的一个购物篮数据集、COOLCAT 算法[8]获得的分类结果以及各记录所对应的相似度和边界度的值。从表 1 中可直观地发现:类 1 中的{1,2,3}、{1,2,4}、{1,2,5}三条记录比其他记录同类 2 中的记录更加相似;而类 2 中记录{1,2,6}、{1,2,7}相对于其他两条记录同类 1 中的记录更加相似。由于边界度既可以反映出一个记录同所属类的相似程度也可以反映出其同其他类的接近程度,所以边界度比相似度更加清楚地反映出我们直观的判定。

表 1　购物篮数据

记录	记录值	分类	相似度	边界度	记录	记录值	分类	相似度	边界度
1	{1,2,3}	1	0.42	0.92	8	{2,3,5}	1	0.42	0.74
2	{1,2,4}	1	0.42	0.92	9	{2,4,5}	1	0.42	0.74
3	{1,2,5}	1	0.42	0.92	10	{3,4,5}	1	0.42	0.56
4	{1,3,4}	1	0.42	0.74	11	{1,2,6}	2	0.45	0.73
5	{1,3,5}	1	0.42	0.74	12	{1,2,7}	2	0.45	0.73
6	{1,4,5}	1	0.42	0.74	13	{1,6,7}	2	0.45	0.52
7	{2,3,4}	1	0.42	0.74	14	{2,6,7}	2	0.45	0.52

2.2　算法描述

文献[9-11]都使用证据积累进行聚类,本文算法主要使用了证据积累的思想。利用定义 2 中的边界度来区分边界点,并通过 $\sqrt{n'}$ 次运行简化的类似 k-means 算法来实现证据的积累,最后依据各条记录的证据积累结果来获取数据集中的边界点。算法主要由记录合并、证据获取和边界判定三部分组成。

记录合并阶段主要用于将原始记录集中相同的数据合并为一条数据并且记录下相同数据的个数,本阶段可将记录个数减少为 $n'(n' \leqslant n)$。

证据获取阶段首先随机从合并后数据集内选取 k 条记录作为簇中心点,然后根据剩余记录相对于 k 个簇的边界度的值依次将各条记录分配到边界度最小的簇中,并且当边界度的值大于阈值 δ 时增加该记录的边界系数值。循环执行 $\lfloor \sqrt{n'} \rfloor$ 次。

边界判定阶段通过判定各个点证据获取阶段中积累的边界系数值是否大于 $beta \times \lfloor \sqrt{n'} \rfloor$ 来判定记录是否为边界点:当记录的边界系数值大于 $beta \times \lfloor \sqrt{n'} \rfloor$ 时为边界点;反之则不是边界点。

算法　CBORDER 算法。

输入　数据集 D,簇个数 k,边界度阈值 δ,边界阈值 $beta$。

输出　数据集 D 中的边界点。

1)数据合并。将相同的记录合并获得合并后的记录总数 n'。

2)证据获取。执行 $\lfloor \sqrt{n'} \rfloor$ 次循环:

2.1)随机选取 k 个中心点放入到 k 个簇集中。

2.2)从剩余记录中选取一条记录计算其相对于 k 个簇集的边界度的值,并将记录放入边界度最小的簇集中,更新簇集。

2.3)当剩余有未并入簇的记录时,转入 2.2),否则向下

执行。

2.4)计算各条记录相对于并入簇的边界度的值,当值大于δ时将该记录的边界度系数值增1。

3)获取边界点。当记录边界度系数值大于$beta \times \lfloor \sqrt{n'} \rfloor$时,记录为边界点将其输出,否则不输出。

算法中共有3个参数:簇个数参数k、边界度阈值δ和边界阈值$beta$。参数k为预测的簇的个数。边界度阈值δ用于确定需要将边界度为多大的记录认定为边界点,当δ的取值减小时相应的边界候选记录个数就会增加;当δ的值增大时边界候选记录个数就会减少。根据实验的结果δ取值在[0.65,0.85]内时会获得较好的效果。边界阈值$beta$用于控制边界点的多少,它表示在边界候选集中认定为边界的记录所占的比例。当$beta$增大时边界点数量增加,当$beta$减小时边界点数量减少,由于$beta$表示的是一个比例范围,所以$beta$的取值范围为[0,1]。

3 实验结果及分析

算法的实验环境:CUP为Intel Pentium Dual-CoreE5300 2.60 GHz,内存为2 GB,操作系统为Microsoft Windows XP professional,编译环境为Microsoft Visual Studio 2005。使用UCI[12]上的Congressional votes、Mushroom和Soybean三个真实数据集对算法的有效性进行了验证。Congressional votes数据集包含435条记录,每条数据记录包含有16个属性。Mushroom数据集包含8 124条记录,每条记录包含22个属性。Soybean数据集包含47条记录,每条记录包含25个属性。

这三个数据集的原始数据中都没有明确地指出聚类的边界。由前文所述可知使用边界度可以较好地区分出聚类的边界,表2~表4中分别列出了利用自然聚类的边界度获取的边界点的状况。这些边界点对应原始数据集有其特殊的含义,例如Mushroom数据集中的边界点可以认为是在物理特性上容易混淆有毒、无毒的蘑菇种类,日常生活中这样的蘑菇更加应该引起注意。本文使用检出率和正确率来验证算法的实验结果的有效性。

表2 Congressional votes数据统计

分类	原始分类	内部点	边界点
democrat	267	208	59
republican	168	150	18

表3 Mushroom数据统计

分类	原始分类	内部点	边界点
poisonous	3916	2861	1055
edible	4208	3187	1020

表4 Soybean数据统计

分类	原始分类	内部点	边界点
normal	20	10	10
irrelevant	27	24	3

$$检出率 = \frac{检出边界个数}{实际边界个数} \tag{4}$$

检出率反映出算法检出的能力,检出率越大说明算法的检出能力越强。

$$正确率 = \frac{检测正确边界个数}{检出边界个数} \tag{5}$$

正确率直接反映出算法的检出精度,正确率越高说明算法检出的精度越高。表5可以看出CBORDER算法的检出率和正确率处在一个较高的水平,可以有效检测出数据集的聚类边界。对于Congressional votes数据集参数为$k=2$、$\delta=0.70$、$beta=0.80$;Mushroom数据集$k=2$、$\delta=0.85$、$beta=0.25$;Soybean数据集$k=2$、$\delta=0.85$、$beta=0.05$。

表5 本文算法对三个数据集的边界检测结果

数据集	实际个数	检出个数	正确个数	检出率/%	正确率/%
Congressional votes	77	54	42	70.13	77.78
Mushroom	2075	1563	1263	75.32	80.81
Soybean	13	13	12	100.00	92.31

4 时间复杂度分析

算法的第1步为数据的预处理过程,当数据集中没有重复记录时这一步骤可以省略,此时的n和n'相等,该步骤的时间复杂度为$O(mn\log n)$;第2步是算法的主体,这一步骤中共包含两个循环分别是一个$\lfloor \sqrt{n'} \rfloor$次的外循环以及一个$n'$次的内循环并且在此内循环中有一个复杂度为$O(m)$的边界度计算过程,所以第2步的时间复杂度为$O(mn^{\frac{3}{2}})$;第3步是数据的输出,时间复杂度为$O(n)$。综上所述,算法总的时间复杂度为$O(mn^{\frac{3}{2}})$。

5 结语

针对现有的聚类边界检测算法仅仅适用于数值型数据,而无法检测分类数据聚类边界的这一问题,本文中使用边界度的概念有效地解决了分类数据聚类边界概念的定义问题,并给出一种分类数据聚类边界的形式化定义;同时,本文提出了一种分类数据聚类边界的检测算法,算法时间复杂度为$O(mn^{\frac{3}{2}})$,可用于对大型、高维分类数据集的聚类边界的检测。

参考文献:

[1] ESTER M, KRIEGEL H P, SANDER J. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Oregon, Portland: AAAI Press, 1996: 226-231.

[2] XIA CHENYI, HSU WYNNE, LEE MONGLI, *et al*. BORDER: efficient computation of boundary points[J]. Knowledge and Data Engineering, 2006, 18(3): 289-303.

[3] QIU BAOZHI, YUE FENG, SHEN JUNYI. BRIM: A efficient boundary points detecting algorithm[C]// Proceedings of Advances in Knowledge Discovery and Data Mining. Berlin: Springer-Verlag, 2007: 761-768.

[4] 邱保志,刘洋,陈本华.基于网格熵的边界点检测算法[J].计算机应用, 2008,28(3): 732-734.

[5] 邱保志,岳峰.基于引力的边界点检测算法[J].小型微型计算机, 2008, 29(2): 279-282.

[6] 邱保志,曹鹤玲.一种高效的基于联合熵的边界点检测算法[J].控制与决策, 2011, 26(1): 71-74.

[7] NOSOVSKIY G V, LIU DONGQUAN, SOURINA O. Automatic clustering and boundary detection algorithm based on adaptive influence function[J]. Pattern Recognition, 2008, 41(9): 2757-2776.

(下转第1669页)

测试者对同一关键词进行搜索,并对返回的结果进行判别以便找出符合自己查询要求的文档,查询结果以少数服从多数为原则。具体的结果如图 1 所示。

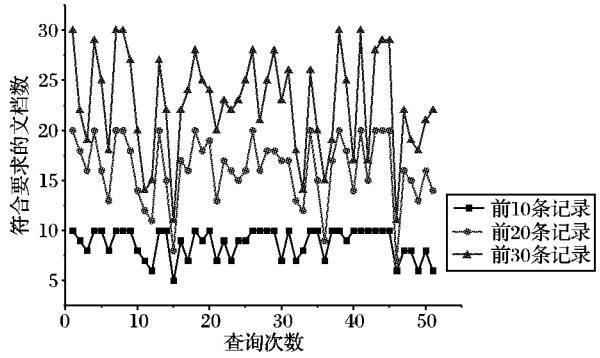

图 1　查询结果分布

从实验结果图 1 可以看出,当查看排序结果的前 10 条记录时,有 38 次符合要求的记录在 8 个以上,其平均查准率为 87.4%;前 20 条记录有 32 次符合要求的记录在 16 个以上,其平均查准率为 80.8%;前 30 条记录有 21 次符合要求的记录在 25 个以上,其平均查准率为 75.1%。这表明,查询结果的前 10 条记录,其查准率最高,最有可能满足用户的要求。

表 1 是 WFPR 算法与传统的 PageRank 算法、HITS 算法的前 10 条记录的查准率比较结果。测试时选取 50 个关键词,并获取每个关键词搜索结果的前 10 条记录,合并,然后交由 6 个人来分别评判。如果有 4 人认为符合自己的查询要求,则该网页视为相关网页;否则为不相关网页。

表 1　查准率比较结果

算法	查准率/%	算法	查准率%
WFPR	87.4	HITS	36.7
PageRank	47.1		

对比 PageRank、HITS 和 WFPR,WFPR 比前二者具有明显优势。由于 PageRank 只是单纯考虑了网页间的链接关系,使得旧的、大量不相关的网页获得提升,从而产生主题漂移和对旧网页的过分偏爱,进而影响了排序结果的质量;而 HITS 在对初次检索结果的根集进行扩充,使得根集中包含部分相关或完全不相关的网页,从而产生主题漂移。WFPR 的优势在于利用网站与网页间的互增强关系为网站内的网页赋予权重,并改变了传统的相关度算法中以单词为单位进行匹配的做法,而是以句子为单位进行匹配,使得搜索结果更为相关;同时利用新网页与用户对网页的反应来对新网页及用户感兴趣的网页进行一定程序的排名提升,使得网页的相关信息得到充分利用,从而较好的排序效果。

总的来说,WFPR 算法在查准率上比其他算法有较大提高。这表明,通过有机结合网页的更新率和查询关键词与网页的相关性和刻画网站与网页之间的互相增强关系的互增强模型,能够有效地提高网页排序结果的质量。

4　结语

本文研究了基于更新时间、网页权威性和用户对网页的反映因素的网页排序算法,以改进网页检索性性能。利用网站与网页之间的相互增强关系,将网站的权威值按各网页对网站的贡献度进行分配;通过记录以往网页的更新时间来预测下一次网页可能的更新时间;将用户的转载、回复次数和其他权威网站对该网页的引用作为用户对网页的评价;利用查询词在摘要和标题中的位置不同、查询词是否在一起等来调整查询相关度的计算。最后,结合 WFPR 查询模型和互增强模型计算网页迭代计算网页评分。实验结果表明,该算法比目前的代表性算法在性能上有较大提高,在专业搜索方面效果更好。

参考文献:

[1] NARAYAN B L, MURTHY C A, PAL S K. Topic continuity for Web document categorization and ranking[C]// Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence. Washington, DC: IEEE Computer Society, 2003: 310 - 315.

[2] RICHARDSON M, DOMINGOS P. The intelligent surfer: Probabilistic combinatino of link and content information in PageRank[C]// Advances in Neural Information Processing System. New York: ACM, 2002: 673 - 680.

[3] HAVELIWALA T H. Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4): 784 - 796.

[4] INGONGNGAM P, RUNGSAWANG A. Topic-centric algorithm: a novel approach to Web link analysis[C]// Proceedings of the 18th AINA Conference. Washington, DC: IEEE Computer Society Press, 2004: 299.

[5] KURLAND O, LEE L. PageRank without hyperlinks: Structural reranking using links induced by language models[J]. ACM Transactions on Information Systems, 2010, 28(4): 18.

[6] ZHANG BENYU, LI HUA, LIU YI, *et al*. Improving Web search results using affinity graph[C]// Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2005: 504 - 511.

[7] RICHARDSON M, PRAKASH A, BRILL E. Beyond PageRank: Machine learning for static ranking[C]// Proceedings of the 15th International Conference on World Wide Web. New York: ACM, 2006: 707 - 715.

[8] 杨伟杰,戴汝为,崔霞. 一种基于信息检索技术的网络新闻影响力分析方法[J]. 软件学报, 2009, 20(9): 2397 - 2406.

[9] The Java search engine[EB/OL]. [2011-10-18]. http://lucene.apache.org/nutch.

[10] 搜狐 IT. 绝大多数搜索引擎用户其实只看第一页搜索结果[EB/OL]. [2011-10-20]. http://www.techw-eb.com.cn/news/2006-05-19/57577.shtml.

(上接第 1656 页)

[8] BARBARA D, COUTO J, LI YI. COOLCAT: an entropy-based algorithm for categorical clustering[C]// Proceedings of the eleventh International Conference on Information and Knowledge Management. New York: ACM Press, 2002: 4 - 9.

[9] FRED A L N, JAIN A K. Data clustering using evidence accumulation[C]// Proceedings of the 16th International Conference on Pattern Recognition. Washington, DC: IEEE Computer Society, 2002: 276 - 280.

[10] FRED A L N, JAIN A K. Evidence accumulation clustering based on the k-means algorithm[C]// Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition. London: Springer-Verlag, 2002: 303 - 333.

[11] FRED A L N, JAIN A K. Combining multiple clusterings using evidence accumulation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(6): 835 - 850.

[12] Online UCI machine learning repository[DB/OL]. [2011-09-20]. http://www.ics.uci.edu/mlearn/MLRepository.