Journal of Computer Applications
计算机应用,2017,37(11):3207 -3211

ISSN 1001-9081
CODEN JYIIDU

2017-11-10
http://www.joca.cn

文章编号:1001-9081(2017)11-3207-05

DOI:10.11772/j.issn.1001-9081.2017.11.3207

基于复杂学习分类系统的密度聚类方法

黄虹玮[1*], 葛笑天[2], 陈烜松[2]

(1. 计算机软件新技术国家重点实验室(南京大学),南京 210023; 2. 江苏省审计厅, 南京 210009)

(*通信作者电子邮箱 godenrainbowjade@gmail.com)

摘 要:提出一种基于复杂学习分类系统(XCS)的密度聚类方法,可以用于对任意形状且带有噪声的二维数据进行聚类分析。此方法称为 DXCSc,主要包括以下三个过程:1)基于学习分类系统,对输入数据生成规则种群,并对规则进行适当压缩;2)将已经生成的规则视为二维数据点,进而基于密度聚类思想对二维数据点进行聚类;3)对密度聚类后的规则种群进行适当聚合,生成最终的规则种群。在第一个过程中,采用学习分类系统框架生成规则种群并进行适当约减。第二个过程认为种群的各规则簇中心比它们的邻居规则具有更高的密度,并且与密度更高的规则间距离更大。在第三个过程中,采用图分割方法对相关重叠簇进行适当聚合。在实验中,将所提方法与 *K*-means、近邻传播聚类算法(AP)、Voting-XCSc 等算法进行了比较,实验结果表明,所提方法在精度方面优于对比算法。

关键词:学习分类系统;进化计算;强化学习;密度聚类;规则合并

中图分类号:TP181 **文献标志码**:A

Density clustering method based on complex learning classification system

HUANG Hongwei[1*], GE Xiaotian[2], CHEN Xuansong[2]

(1. *State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing Jiangsu* 210023, *China*;
2. *Jiangsu Provincial Audit Office, Nanjing Jiangsu* 210009, *China*)

Abstract: A density clustering method based on eXtended Classifier Systems (XCS) was proposed, which could be used to cluster the two-dimensional data sets with arbitrary shapes and noises. The proposed method was called Density XCS Clustering (DXCSc), which mainly included the following three processes: 1) Based on the learning classification system, regular population of input data was generated and compressed. 2) The generated rules were regarded as two-dimensional data points, and then the two-dimensional data points were clustered based on idea of density clustering. 3) The regular population after density clustering was properly aggregated to generate the final regular population. In the first process, the learning classifier system framework was used to generate and compact the regular population. In the second process, the rule cluster centers were characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. In the third process, the relevant clusters were properly merged using the graph segmentation method. In the experiments, the proposed DXCSc was compared with *K*-means, Affinity Propagation (AP) and Voting-XCSc on a number of challenging data sets. The experimental results show that the proposed approach outperforms *K*-means and Voting-XCSc in precision.

Key words: Learning Classifier System (LCS); evolutionary computing; reinforcement learning; density clustering; rule merging

www.joca.cn

0 引言

学习分类系统(Learning Classifier System, LCS)是一个动态感应环境、模拟认知的机器学习系统,它可根据环境反馈并通过强化学习[1]来评估种群中的分类规则,进而借助遗传算法[2-3]对种群进行进化,被认为是最常用的基于规则的学习工具之一。基于 LCS 框架,Wilson 提出了复杂学习分类系统(eXtended Classifier System, XCS)[4]和实值复杂学习分类系统(complex real Learning Classifier System, XCSR)[5],以处理样本连续实值属性问题。近年来,已有一些工作通过利用 XCS 进行聚类等无监督学习研究:Shi 等[6]提出了一种基于多规则表示的 XCS 聚类算法,每个数据点由规则对表示,并迭代合并包含共同数据点的规则;Shi 等[7]还提出了一种名为 XCSc(XCS clustering)的聚类方法,该方法采用基于多规则的表示法,通过随机生成几个规则来初始化每个数据点,同时,受 Chameleon 算法中使用的分层策略的启发,XCSc 采用基于图的规则合并方法,能够处理复杂结构数据集;Qian 等[8]提出一种基于多轮投票的 XCSc(Voting-XCSc)聚类方法,可以实现自动确定聚类簇数量。Voting-XCSc 算法在 XCSc[7]算法基础上演变而来,二者区别如图 1 所示。在 Voting-XCSc 中,当有新数据来自环境时,系统将自动检查整个规则种群,对于新数据,Voting-XCSc 会基于一定规则生成规则匹配集,并计算规则匹配集中每条规则的相应奖励,之后 Voting-XCSc 通过强化学习机制更新规则的相应参数。

收稿日期:2017-05-11;**修回日期**:2017-07-05。 **基金项目**:江苏省重点研发计划(产业前瞻与共性关键技术)项目(BE2015213)。

作者简介:黄虹玮(1986—),男,陕西汉中人,博士研究生,主要研究方向:在线聚类、学习分类系统; 葛笑天(1963—),男,江苏徐州人,高级审计师,硕士,CCF 会员,主要研究方向:政务大数据、审计信息化; 陈烜松(1971—),男,江苏南京人,高级审计师,硕士,主要研究方向:审计大数据应用、多维数据分析。

Voting-XCSc 属于无监督学习，一般在没有外部输入的情况下对未标记数据进行聚类分析。通过若干轮（需要设定最大学习次数）学习，Voting-XCSc 对规则进行合并整合，从而生成规则集群，最后为每个新数据确定相应的集群。Voting-XCSc 算法虽然可以自动确定聚类数目，但需要进行多轮学习，算法效率较低。

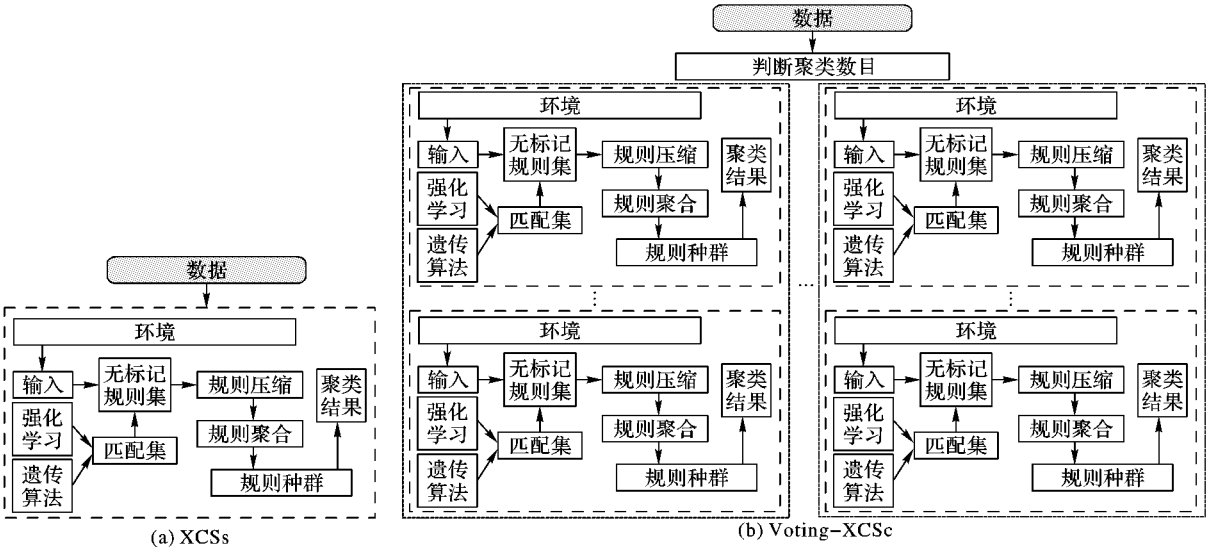

图 1 Voting-XCSc 算法与 XCSc 算法区别示意图

Fig. 1 Difference diagram between Voting-XCSc and XCSc

近邻传播聚类算法（Affinity Propagation, AP）[8] 将所有的数据点视为候选的簇代表点，避免聚类结果受限于初始簇代表点的选择，同时对于相似度矩阵的对称性没有要求。AP 算法较为成功的特点是能够自动产生合理的聚类簇数目，在数据比较充足的情况下，AP 能够准确地找出聚类代表点，其最终所获得的聚类效果往往也较为理想。然而，虽然 AP 算法同样解决了预先人为设定聚类簇数目的问题，但需要进行多轮迭代计算，同时由于 AP 算法是基于中心的聚类方法，它与其他基于中心的聚类方法一样，在紧凑的具有超球形分布的数据集上具有较好的聚类性能，但并不适合任意形状聚类问题。

为避免预先人为设定聚类簇数目，同时对任意形状的数据取得良好聚类效果，并保证一定的聚类算法效率，以适应未来流数据在线聚类需求，本文提出一种基于复杂学习分类系统（XCS）的密度聚类方法（Density XCS clustering, DXCSc），主要贡献在于可自动确定聚类数目，聚类精度优于 Voting-XCSc、AP 等算法，且算法复杂度低于 Voting-XCSc 和 AP 算法。本算法以学习分类系统算法 XCSc 为框架，运用规则种群概念代表数据点集合，有效降低了表示大规模数据的复杂度，同时运用 CFSFDP（Clustering by Fast Search and Find of Density Peaks）算法[10] 将各条规则重新视为二维数据点进行密度聚类，有效避免了 Voting-XCSc、AP 算法的多轮学习，取得良好的聚类效果。

1 密度聚类学习分类系统框架与学习机制

本文提出的 DXCSc 算法（如图 2 所示）与 Voting-XCSc 算法的主要不同在于：DXCSc 算法不需要预先设定最大聚类数目，且不用进行 k_{max} 作为限定运算次数下的多轮计算，算法复杂度低于 Voting-XCSc 算法。DXCSc 算法框图如图 2 所示。

在聚类算法中，一般假定样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$ 包含 m 个无标记样本，每个样本 $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{in}\}$ 是一个 n 维特征向量，则聚类算法将样本集 D 划分为 k 个不相交的簇 $\{C_l \mid l = 1, 2, \cdots, k\}$，其中 $C_{l'} \bigcap_{l' = l} C_l = \varnothing$ 且 $D = \bigcup_{l=1}^{k} C_l$。相应地，本文用 $\lambda_j \in \{1, 2, \cdots, k\}$ 表示样本 $\boldsymbol{x}_j$ 的“簇标记”（cluster label），即 $\boldsymbol{x}_j \in C_{\lambda_j}$。于是，聚类的结果可用包含 m 个元素的簇标记 $\chi(x)$ 表示。

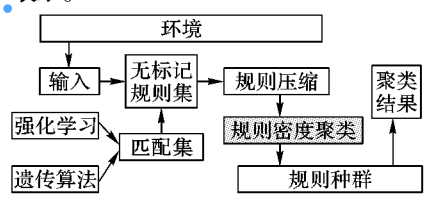

图 2 DXCSc 算法框图

Fig. 2 Diagram of DXCSc

本文在 XCSc 算法框架基础上，当新的数据来到并对应生成相关规则集合后，通过对重复冗余规则进行压缩，初步形成规则种群；之后本文创新性地提出，将规则种群中的规则再次视为二维数据点，用对规则点的聚类取代对原始数据点的聚类，一条规则可代表大量数据点，从而可大量减少聚类所需计算量，并后续存在将高维数据降至低维的可能。其中对于每一个规则数据点 i，计算 i 的局部密度 ρ_i 和与更高密度点之间的距离 δ_i，二者定义如下所示：

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{1}$$

$$\delta_i = \min_{j:\rho_i > \rho_j} (d_{ij}) \tag{2}$$

其中，$\chi < 0$ 时，$\chi(x) = 1$，当 $\chi \geqslant 0$ 时，$\chi(x) = 0$，其中 d_{ij} 是规则点 i 与规则点 j 之间的距离，d_c 是用户指定的一个距离参数；ρ_i 代表与规则点 i 距离小于 δ_i 的规则总数。式（2）中，δ_i 代表了距离规则点 i 最近且密度大于规则点 i 的规则点 j 与 i 之间的距离。

本文将具有最高局部密度的规则点 i 的 δ_i 设定为 $\max_j(d_{ij})$。本文提出的规则密度聚类方法主要有以下几步：1）计算规则种群中各规则点之间的距离；2）基于规则点之间的距离和用户设定的 d_c 计算每个规则点 i 的 ρ_i；3）计算每个规则点 i 的距离 δ_i。至此，参考 CFSFDP 算法[10]，通过决策图方法人为选择 ρ_i、δ_i 值均较高的点就可以逐步算出簇中心，但

www.joca.cn

是此方法存在的一个主要问题是,当数据分布存在较多密度极值点时,仅仅依靠规则密度搜索算法无法取得理想聚类效果。因此,本文采取 Fuzzy-CFSFDP[11] 提出的方法,进一步优化了簇中心选择方式。如式(3)、(4)所示,EC_i 代表预期的簇中心,$\sigma(\delta_i)$ 是由式(2) 计算所得的距离的标准差,$\mu(\rho_i)$ 是 ρ_i 的平均值。根据 CFSFDP 算法,各簇中心间距离较远,因此簇中心数据点间距离应满足式(3),与此同时,由于 CFSFDP 算法会产生具有较大距离 δ_i 和较小密度的噪声簇中心,因此,需要通过式(4) 来分离预期簇中心的噪声。所以,当规则点 i 的 ρ_i、δ_i 满足上述式(3)、(4) 时,方可认为该点为簇中心。

$$EC_i = (\delta_i) \geqslant 2\sigma(\delta_i) \tag{3}$$

$$LC_i = EC_i \geqslant \mu(\rho_i) \tag{4}$$

1.1　规则聚合(Merge)

在 1.1 节中,满足 EC_i、LC_i 条件而生成的初始子类数目较多,本文算法采用 Chameleon 算法[11] 对初始子类间距离进行了式(5)、(6)相关计算,主要进行类间相似性衡量比较,进而将相似性聚类子类进行合并得到最终聚类结果。Chameleon 算法在融合子类的过程中,既考虑两个类之间的相对紧密情况 RI_{ij}(类 i 与类 j 之间的相对互连性),又考虑两个类的内部连接情况 RC_{ij}(类 i 与类 j 之间的相对近似性)。

$$RI_{ij} = AI_{ij}/[(AI_i + AI_j)/2] \tag{5}$$

$$RC_{ij} = \frac{AC_{ij}}{\frac{|G_i|}{|G_i + G_j|}AC_i + \frac{|G_j|}{|G_i + G_j|}AC_j} \tag{6}$$

其中:AI_{ij}(绝对互连性) 是将图 G_i 中点连接到图 G_j 中点的边权重之和。AI_i 或 AI_j(绝对内部互连) 是属于图 G_i 或图 G_j 的最小切分平分线的边的权重之和。AC_{ij}(绝对近似性) 是将图 G_i 中的点连接到图 G_j 中点的边平均权重。AC_i 或 AC_j(绝对内部接近度) 是属于图 G_i 或图 G_j 最小切分平分线的边平均权重。

DXCSc 算法具体内容如下所示:

输入:$X = \{X_1, X_2, \cdots, X_n\}$ 无监督数据集;d_c 计算局部密度 ρ_i 所需半径;

输出:$C = \{C_1, C_2, \cdots, C_k\}$ 簇集合。

算法过程 1:基于学习分类系统框架,通过强化学习及遗传学习模块,对输入数据生成规则种群,并对规则进行适当压缩。

算法过程 2:对规则种群进行密度聚类。

2.1)计算各规则的局部密度 ρ 和 δ。

2.2)采用 Fuzzy-CFSFDP 方法确定簇中心。

2.3)将每个规则划归至与其距离最近且局部密度更高的规则种群中。

算法过程 3:对规则种群进行适当聚合。

3.1)对 2.3)步骤中生成的规则种群中的每一对规则 $\{C_i, C_j\}$,计算 $RI\{C_i, C_j\} * RC\{C_i, C_j\}^{\alpha}$。

3.2)将具有最高 $RI\{C_i, C_j\} * RC\{C_i, C_j\}^{\alpha}$ 值的规则对 $\{C_i, C_j\}$ 合并。

3.3)重复 3.1)、3.2)步,直至终止条件达到。

2　实验结果和分析

2.1　实验设置

对于实验中的参数设置,本文遵循 Voting-XCSc[8] 中对应参数的相同值进行公平比较。种群最大规则数设为 1 000,学习率设为 0.2,遗传算法中突变和交叉的概率分别为 0.001 和 0.8,用于计算精度的参数 α 和 v 分别设置为 0.1 和 5。此外,根据 CFSFDP[9] 中分析,通过选择适当的参数 d_c 使得平均邻居数量在整个数据集中占比约 1% 至 2% 即可,因为在决策图中选择簇中心主要取决于数据点密度和高密度距离的相对关系,而不是绝对值,因此 d_c 的选择对最终聚类效果影响不大。实验计算机环境为:处理器为 2.3 GHz Intel Core i5,内存为 8 GB 1 600 MHz DDR3,硬盘为 320 GB,操作系统为 Mac OS X EI Capitan,编程语言为 Java、C++ 和 Matlab。

2.2　典型数据集实验分析

本文首先使用 Flame[13]、Aggregation[14]、D31[15]、R15[15] 等典型数据集,测试并对比本文提出的 DXCSc 算法聚类效果。Flame 数据集包含 240 个二维数据点,可分为 2 簇;Aggregation 数据集包含 788 个二维数据点,可分为 7 簇;D31 数据集包含 3 100 个二维数据点,可分为 31 簇;R15 数据集包含 600 个二维数据点,可分为 15 簇。

如图 3 所示,DXCSc 算法可以在无预先簇数目输入的情况下对任意形状的二维数据 Flame Data 进行有效聚类。

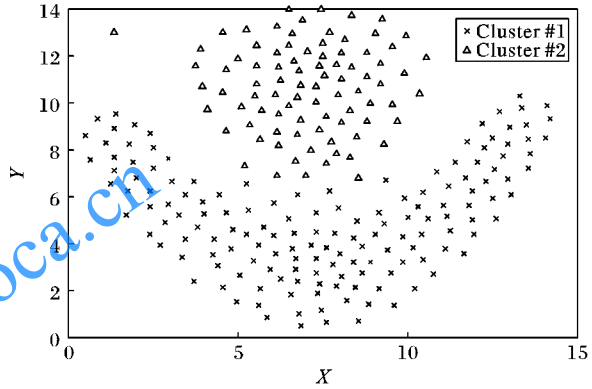

图 3　DXCSc 算法在 Flame[13] 数据集上聚类效果

Fig. 3　DXCSc clustering effects on dataset Flame[13]

如图 4 所示,K-means 算法在预先输入簇数目 ($k_{max} = 2$) 情况下,对任意形状的二维数据 Flame Data 聚类效果不是很好。

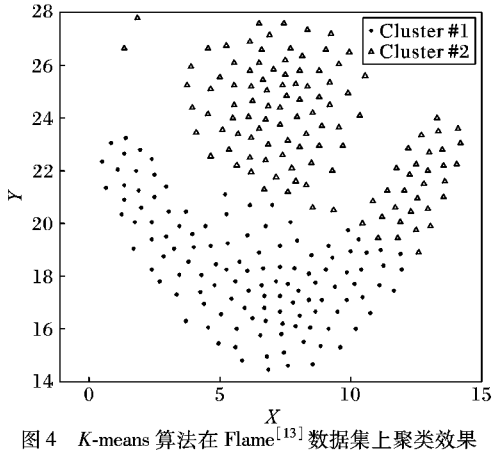

图 4　K-means 算法在 Flame[13] 数据集上聚类效果

Fig. 4　K-means clustering effects on dataset Flame[13]

如图 5 所示,AP 算法虽然同样不需要预先设定簇数目,但对任意形状的二维数据 Flame Data 聚类效果也不好。此外,本文将上述算法分别在 Aggregation[14]、D31[15]、R15[15] 数据集上进行了测试比对,实验结果表明:DXCSc 算法在任意形状分布数据集 Flame 和超球形分布数据集 D31、R15 上都具有较好的聚类性能;而 K-means、AP 算法只在超球形分布数据集上具有较好聚类性能,对任意形状数据集聚类效果不好。

www.joca.cn

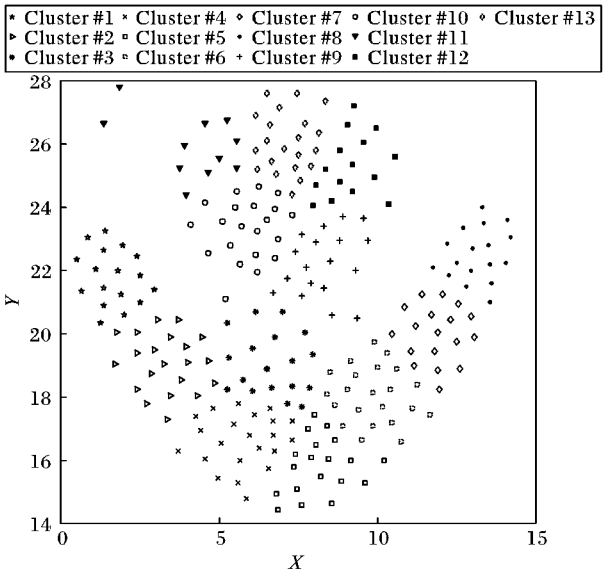

图 5 AP 算法在 Flame[13] 数据集上聚类效果

Fig. 5 AP clustering effects on dataset Flame[13]

如表 1 所示，本文对 DXCSc、*K*-means、AP 以及 Voting-XCSc 算法在上述数据集上的聚类精度(Fitness)进行了比较，如表中结果所示，DXCSc 在任意形状的二维数据集上可以取得良好的聚类效果。

表 1 聚类精度比较(典型数据集)

Tab. 1 Comparison of clustering fitness on typical data sets

算法	数据集			
	Aggregation	D31	R15	Flame
K-means (需预先指定簇数量 *k*)	0.81	0.93	0.93	0.86
AP (不需预先指定簇数量 *k*)	0.41	0.97	0.99	0.18
Voting-XCSc (无需预先指定簇数量 *k*)	0.94	0.45	0.97	0.91
DXCSc (无需预先指定簇数量 *k*)	**0.95**	**0.97**	**0.98**	**0.93**

与此同时，为确保评价指标的科学性，本文采用 NMI (Normalized Mutual Information)评价指标对 DXCSc、*K*-means、AP 以及 Voting-XCSc 算法在上述数据集上的聚类效果进行了比较(如表 2 所示)，该指标可用于衡量两个数据分布的吻合程度，取值范围为[0,1]，该值越大表明聚类结果与真实情况越吻合。

表 2 聚类 NMI 比较(典型数据集)

Tab. 2 Comparison of NMI on typical data sets

算法	数据集			
	Aggregation	D31	R15	Flame
K-means	0.89	0.92	0.95	0.42
AP	0.77	0.85	0.98	0.47
Voting-XCSc	0.66	0.95	0.98	0.39
DXCSc	**0.99**	**0.96**	**0.98**	**0.93**

2.3 真实数据集实验分析

本文采用 UCI 中的 Iris、Wine、Ionosphere、Letter[16] 等真实数据集对上述算法进行了进一步验证。其中，Iris 为鸢尾花卉数据集(数据实例数 $N=150$，数据维度 $D=4$，聚类簇数 $K=3$)，是一类多重变量分析的数据集，可通过花萼长度、花萼宽度、花瓣长度、花瓣宽度 4 个属性预测鸢尾花卉属于 Setosa、Versicolour、Virginica 三个种类中的哪一类。Wine 数据集($N=178$，$D=13$，$K=3$)包含来自 3 种不同起源的葡萄酒的共 178 条记录，数据集中 13 个属性是葡萄酒的 13 种化学成分，通过化学分析可以来推断葡萄酒的起源。Ionosphere 数据集($N=351$，$D=34$，$K=2$)根据给定的电离层中的自由电子的雷达回波预测大气结构。Letter Recognition 数据集($N=20\,000$，$D=16$，$K=2$)将大量的黑白矩形像素显示的每一个字符标识为英文字母表中的 26 个大写字母之一。

如表 3 所示，*K*-means 算法在事先指定聚类簇数量 *k* 的情况下，聚类精度一般优于 AP 和 DXCSc 算法。在不指定聚类簇数量 *k* 的条件下，DXCSc 算法精度优于 AP 算法。上述算法在未进行大量参数调整实验情况下，对真实数据的聚类效果并不是很好。

表 3 聚类精度比较(真实数据集)

Tab. 3 Comparison of clustering fitness on real data sets

数据集	*K*-means	AP	DXCSc
Iris	0.98	0.50	0.59
Wine	0.70	0.39	0.57
Ionosphere	0.19	0.27	0.55
Letter	0.37	0.15	0.10

2.4 算法复杂度分析

在 DXCSc 算法过程 1 中，当有新数据输入时，学习分类系统遍历整个种群并更新相关规则参数，算法复杂度为 $O(n)$，其中 n 为数据点数；在算法过程 2 中，需要 $O(n^2)$ 的时间来建立距离矩阵，得出任意两点间的距离，并计算局部密度 ρ 和 δ；在算法过程 3 中，算法运行时间主要消耗在簇与簇之间的相似度计算和簇的融合更新之中，此部分算法复杂度为 $O(n\log n)$。综合分析，DXCSc 的算法复杂度为 $O(n^2+n\log n)$。*K*-means算法优点是简单实用，确定的 *k* 个划分到达平方误差最小，该算法当聚类是密集的，且类与类之间区别明显时，效果较好。对于处理大数据集，*K*-means 算法相对可伸缩和高效的，计算的复杂度为 $O(nkt)$(其中 n 是数据对象的数目，t 是迭代的次数，k 为簇数目)；虽然 *K*-means 算法复杂度相对较低，但需要用户预先设定聚类簇数目 *k*，存在一定局限性。AP 算法复杂度较高，为 $O(n^2\log n)$。

基于上述分析，本文对实验所耗时间进行了比较分析，结果如表 4 所示。AP 算法复杂度较高，耗时较长，*K*-means 算法与 DXCSc 算法耗时较为接近。

表 4 聚类时间比较 s

Tab. 4 Comparison of clustering time s

数据集	*K*-means	AP	DXCSc
Aggregation	0.23	11.5500	0.27
D31	0.31	237.4600	0.34
R15	2.07	13.9200	2.04
Flame	0.19	2.2100	0.22
Iris	0.21	2.6188	0.20
Wine	0.19	2.7961	0.21
Ionosphere	1.15	4.1100	1.02
Letter	6.36	491.5600	3.41

2.5 讨论

本文提出的 DXCSc 算法在任意形状的二维分布数据集上

www.joca.cn

能够取得良好的聚类效果,主要取决于以下几个因素:首先,DXCSc 采用学习分类系统框架后用规则代替原始二维数据点的表示方法,有效降低了数据表示的复杂度。具体如下,在 DXCSc 中,使用矩形作为规则的表示形式以处理连续值。规则可表示为$(C_i,S_i)(i=1,2,\cdots,d)$,其中 C_i 为数据点的中心,S_i 为拓展半径。当一个数据点 I_i 满足 $C_i-S_i\leqslant I_i\leqslant C_i+S_i$ 时,则该点可由规则(C_i,S_i)表示。通过上述处理,海量的二维数据点就可以通过合并约减后的规则种群来表示,如图 6 所示。

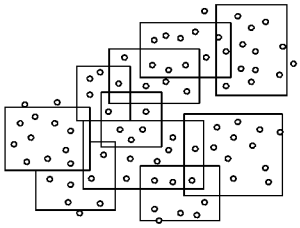

图 6　规则种群降低数据表示复杂度示意图

Fig. 6　Diagram of reducting data representation complexity by rule population

其次,DXCSc 将合并约减后的规则再次表示为二维数据点(如图 7 所示),并应用密度聚类思想将上述数据点进行快速聚类,从而可以将 Voting-XCSc 的多轮学习转化为了单轮学习,避免了对原始数据的多次遍历。

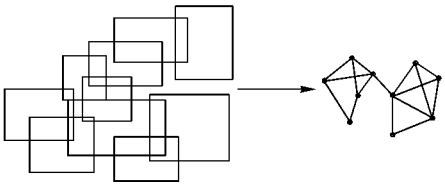

图 7　规则种群密度聚类示意图

Fig. 7　Diagram of clustering rule population density

3　结语

本文提出了一个新的基于学习分类系统的密度聚类方法(DXCSc),用于研究学习分类系统在无监督环境下的聚类效果。DXCSc 聚类方法的主要流程包含 3 个阶段:第一,DXCSc 基于学习分类系统框架,通过强化学习及遗传学习模块,对输入数据生成规则种群,并对规则进行适当压缩;第二,DXCSc 对规则种群进行密度聚类;第三,对规则种群进行适当聚合,生成最终的规则种群。本文通过一些数据集的无监督聚类实验说明了 DXCSc 的学习过程与效果。实验结果表明,DXCSc 能够在不提前设定簇数量 k 的前提下,对无监督的典型二维数据取得较好的聚类效果,聚类精度优于 AP、Voting-XCSc 等算法。

下一步的工作有:首先,基于学习分类系统框架具有适应在线学习环境的特点,尝试把 DXCS 运用到二维流数据在线聚类中,并与一些更为复杂的流数据聚类算法比较性能。其次,设计面向流数据的 Online Voting-XCSc 算法。此外,可进一步将 DXCSc 算法扩展为面向高维数据以及高维流数据的聚类算法。

参考文献(References)

[1] SUTTON R S, BARTO A G. Reinforcement Learning: an Introduction [M]. Cambridge: MIT Press, 1998: 1.

[2] HOLLAND J H. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control and Artificial Intelligence[M]. Cambridge: MIT Press, 1992: 90 – 118.

[3] GOLDBERG D E. Genetic algorithm in search, optimization, and machine learning [EB/OL]. [2016-11-20]. http://www.openisbn.com/download/0201157675.pdf.

[4] WILSON S W. Classifier fitness based on accuracy [J]. Evolutionary Computation, 1995, 3(2): 149 – 175.

[5] WILSON S W. Get real! XCS with continuous-valued inputs[C]// Learning Classifier Systems, From Foundations to Applications. London: Springer-Verlag, 2000: 209 – 222.

[6] SHI L, SHI Y, GAO Y. Clustering with XCS and agglomerative rule merging[C]// Proceedings of the 10th International Conference on Intelligent Data Engineering and Automated Learning. Berlin: Springer-Verlag, 2009: 242 – 250.

[7] SHI L, SHI Y, GAO Y, et al. XCSc: a novel approach to clustering with extended classifier system [J]. International Journal of Neural Systems, 2011, 21(1): 79 – 93.

[8] QIAN L, SHI Y, GAO Y, et al. Voting-XCSc: a consensus clustering method via learning classifier system[C]// Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning. Berlin: Springer, 2013: 603 – 610.

[9] FREY B J, DUECK D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972 – 976.

[10] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492 – 1496.

[11] MEHMOOD R, BIE R, DAWOOD H, et al. Fuzzy clustering by fast search and find of density peaks[C]// Proceedings of the 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things. Piscataway, NJ: IEEE, 2015: 258 – 261.

[12] KARYPIS G, HAN E H, KUMAR V. Chameleon: hierarchical clustering using dynamic modeling[J]. Computer, 2002, 32(8): 68 – 75.

[13] FU L, MEDICO E. FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data [J]. BMC Bioinformatics, 2007, 8(1): 3.

[14] GIONIS A, MANNILA H, TSAPARAS P. Clustering aggregation [J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 1 – 30.

[15] VEENMAN C J, REINDERS M J T, BACKER E. A maximum variance cluster algorithm[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(9): 1273 – 1280.

[16] BLAKE C L, KEOGH E, MERZ C J. UCI repository of machine learning databases[DB/OL]. [2016-11-20]. https://archive.ics.uci.edu/ml/datasets.html.

This work is partially supported by the Key Research and Development Program (Industry Forward and Common Key Technology) Project of Jiangsu Province (BE2015213).

HUANG Hongwei, born in 1986, Ph. D. candidate. His research interests include online clustering, learning classifier system.

GE Xiaotian, born in 1963, M. S., senior auditors. His research interests include government affairs data, audit information.

CHEN Xuansong, born in 1971, M. S., senior auditors. His research interests include audit big data applications, multidimensional data analysis.

www.joca.cn