一种AP算法的改进:M-AP聚类算法

甘月松 陈秀宏 陈晓晖
(江南大学数字媒体学院 无锡 214122)

摘 要 Affinity Propagation(AP)聚类算法将所有数据点作为潜在的聚类中心,在相似度矩阵的基础上通过消息传递进行聚类。与传统聚类方法相比,对于大规模数据集,AP是一种快速、有效的聚类方法。但是AP算法在聚类结构复杂的(非团状)数据集上得到的效果并不是很好。因此,在AP的基础上加入一个merge过程,将AP算法改进为M-AP算法,可以有效地解决这种问题。而当样本数目比较大时,将CVM压缩算法融入其中,可以有效地解决大样本问题。

关键词 聚类,Affinity propagation(AP算法),M-AP,合并过程,CVM压缩
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.1.051

Improved AP Algorithm:M-AP Clustering Algorithm

GAN Yue-song CHEN Xiu-hong CHEN Xiao-hui
(School of Digital Medium,Jiangnan University,Wuxi 214122,China)

Abstract Affinity propagation(AP) clustering simultaneously considers all data points as potential exemplars. It takes similarity between pairs of data points as input measures,and clusters gradually during the message-passing procedure. But the result of AP clustering algorithm in the data set of complex structure(non-group)is not very good. Therefore, we proposed a new clustering algorithm by adding a merge process on the basis of AP clustering algorithm,called M-AP algorithm which can effectively solve this kind of problem. When the number of samples is very large,the problem of large sample can be effectively solved by using CVM compression algorithm.

Keywords Clustering algorithm,Affinity propagation,Merge-AP,Merge process,CVM compress

1 引言

聚类(clustering)是指根据"物以类聚"的原理,将本身没有类别的样本聚集成不同的簇并对每一个这样的簇进行描述的过程,其目的是使得属于同一个簇的样本之间应该彼此相似,而不同簇的样本应该足够不相似。聚类算法是一种有效的数据分析方法,但与分类规则不同,聚类算法是在没有任何数据的先验信息下对数据进行聚类分析的。聚类是数据挖掘、模式识别等研究方向的重要研究内容之一,在识别数据的内在结构方面具有极其重要的作用,主要应用于模式识别中的语音识别、字符识别等领域。其中在机器学习领域,聚类算法被广泛应用于图像分割和机器视觉等方面;而在图像处理领域,聚类被很好地应用于数据压缩和信息检索中。聚类的另一个主要应用是数据挖掘(多关系数据挖掘)、时空数据库应用(GIS等)、序列和异类数据分析等。此外,聚类还应用于统计科学。值得一提的是,聚类分析对生物学、心理学、考古学、地质学、地理学以及市场营销等研究也都有重要作用。

本文提出的M-AP算法是对近邻传算法(affinity propagation,AP)[1,15,16,22,23]的改进。AP算法是由Science中的一篇文章[1]提出来的。与以往的聚类方法相比,此方法可以更快地处理大规模数据,得到较好的聚类效果。文献[1]中的实验结果表明,AP算法在很短的时间内就能得到K中心[3]算法花费很长时间才能达到的聚类效果。AP聚类的另一个优点是,它对数据形成的相似矩阵的对称性没有任何要求,这样也就扩大了它的应用范围。但是,对于一些本身具有复杂结构(非团状)的数据集,近邻传播聚类通常不能得到合理的聚类结果。

本文将原始AP算法与merge过程相结合,通过对AP聚类算法所得到类别的合并,达到了解决复杂结构的聚类问题的目的。实验结果表明,M-AP算法性能不仅在非团状数据集中有明显的提升,而且在团状数据集中与原始聚类算法性能相比也有一定的提高。对比实验结果也表明:从所有的实验数据集的整个实验结果来看,M-AP算法具有一定的优势。

2 几种聚类算法介绍

2.1 K-means算法

1967年,MacQueen首次提出了K均值聚类算法(K-means算法)[2,19]。迄今为止,很多聚类任务都选择该经典算法。该算法的核心思想是找出x个聚类中心$\Omega=\{\omega_1,\omega_2,\cdots,$

到稿日期:2014-03-10 返修日期:2014-06-16 本文受国家自然科学基金(61373055)资助。
甘月松(1992—),男,硕士生,主要研究方向为数字图像处理、模式识别,E-mail:1015415796@qq.com;陈秀宏(1964—),男,博士,教授,主要研究方向为数字图像处理、模式识别;陈晓晖(1989—),男,硕士生,主要研究方向为数字图像处理、模式识别。

$\omega_K\}$，使得每一个数据点 X 和与其最近的聚类中心 ω 的距离平方和最小(该平方距离和被称为偏差 D)。

K-means 算法的基本思想是：首先从 N 个数据对象中任意选择 k 个对象作为初始聚类中心，对其它数据则根据它们与这些聚类中心的相似度(距离)分别将它们分配给与其最相似的类(聚类中心所代表的)；然后再计算每个新聚类的聚类中心(该聚类中所有对象的均值)。重复以上过程，直到标准测度函数开始收敛为止。这里的标准测度函数为：

$$E=\sum_{i=1}^{k}\sum_{x\in\omega_i}|x-m_i|^2 \tag{1}$$

其中，m_i 是簇 ω_i 的均值。该准则是求每个对象到其簇中心距离的平方和，这样可使得所生成的 k 个结果簇尽可能紧凑和独立。

尽管 K-means 算法可以有效地解决很多种聚类问题，但是它也具有以下缺点：1)算法中的 k 需事先给定，而该值的选定是非常难以估计的；2)算法需要根据初始聚类中心来确定一个初始划分，然后对初始划分进行优化，且算法的初始聚类中心为虚拟的、不存在的点；3)该算法需要不断地进行样本分类调整，并计算调整后的新的聚类中心，因此当数据量非常大时，算法的时间开销非常大。

2.2 FCM 算法

利用均方逼近理论，构造类内平方误差和 WGSS(Within-in-Groups Sum of Squared Error)作为聚类目标函数，得到带约束的非线性规划问题，并以此来求解聚类问题。随着模糊划分概念的提出，Dunn 首先将其推广到加权 WGSS 函数，后来由 Bezdek 扩展到加权 WGSS 的无限族，获得了模糊 C-均值(FCM)聚类算法[4,18]。FCM 算法是利用隶属度来确定每个数据元素属于某个类别的程度(用值在 0 与 1 间的隶属函数表示)，把 N 个数据 x_n 分成 K 个模糊组，求每组的类中心，使得非相似性指标的价值函数：

$$J(U,V)=\sum_{n=1}^{N}\sum_{i=1}^{K}(u_{in})^m(d_{in})^2 \tag{2}$$

达到最小。当 $m=1$ 时，FCM 就退化为 HCM(硬聚类算法)。研究表明，m 的最佳选择范围是[1,2.5]，通常取 $m=2$。

FCM 聚类算法能从任意给定初始点开始而收敛到其目标函数 $J_m(U,V)$ 的局部极小点，但它对类的初值、类的形状、大小等都过于敏感，从而影响中间计算量以及最终聚类结果。

2.3 AP 算法

近邻传播(AP)算法由 Frey 等人提出以来，一直受到各个研究领域的广泛关注。AP 算法本质是一种基于因子图的信念传播和最大化算法。它根据 N 个数据点之间的相似度进行聚类，这些相似度可以是对称的(即两个数据点互相之间的相似度一样，如欧氏距离)，也可以是不对称的(即两个数据点互相之间的相似度不等)。AP 算法不需要事先指定聚类数目，而是将所有的数据点都作为潜在的聚类中心，称之为 exemplar。

将 N 个数据点之间的相似度组成 $N\times N$ 的相似度矩阵 S，并以对角线上的值 $S_{i,i}$ 作为第 i 个数据点能否成为聚类中心的评判标准，该值越大，表明该数据点成为聚类中心的可能性也就越大，称之为参考度 p(preference)。聚类的数量受到参考度 p 的影响，如果认为每个数据点都有可能作为聚类中心，那么 p 就应取相同的值；如果取相似度的均值作为 p 值，则聚类数量是中等的；如果取最小值，则得到类数较少的聚类。

AP 算法中传递两种类型的消息：从点 i 发送到候选聚类中心 k 的数值消息 $r_{i,k}$ 和从候选聚类中心 k 发送到 i 的数值消息 $a_{i,k}$，这里 $r_{i,k}$ 反映 k 点是否适合作为 i 点的聚类中心，而 $a_{i,k}$ 则反映 i 点是否选择 k 作为其聚类中心，因此它们分别称为吸引度和归属度。$r_{i,k}$ 与 $a_{i,k}$ 越强，则 k 点作为聚类中心的可能性越大，并且 i 点隶属于以 k 点为聚类中心的聚类的可能性也越大。AP 算法通过迭代而不断地更新每一个点的吸引度和归属度值，直到产生 m 个高质量的 exemplar，同时将其余的数据点分配到相应的聚类中。

AP 算法过程如下：先计算 N 个点之间的相似度值，并构作矩阵 S，选取 p 值(一般取 S 的均值)；设置一个最大迭代次数(随数据集而定，一般取 1000)，在迭代过程中，计算每一次迭代的 r 值和 a 值，根据 $r_{k,k}+a_{k,k}$ 值来判断是否为聚类中心(当 $r_{k,k}+a_{k,k}>0$ 时即认为是一个聚类中心)，当迭代次数超过最大迭代次数或者当聚类中心连续多少次迭代不发生改变(一般假设连续迭代 50 次而不发生改变)时终止计算。

AP 算法相对于其他聚类算法具有以下优势：1)AP 聚类不需要指定 K(经典的 K-means)或者是其他描述聚类个数的参数；2)一个聚类中聚类的中心点 exemplar 是原始数据中确切存在的数据点，而不是虚拟出的代表点；3)AP 算法是一个稳定的算法，多次执行的结果相同，不需要进行随机选取初值的步骤，即对初始化不敏感。

尽管 AP 算法对一些结构简单的数据其聚类效果很好，但是对一些结构比较复杂(如非团状)的数据集，它却往往得不到很好的聚类效果。

3 改进的 AP 算法——M-AP 算法

3.1 基于距离的 merge 过程

Merge 过程是将多个类别按照一定的计算方法合并成为较少类别的过程。已知其中 $\Omega=\{\omega_1,\omega_2,\cdots,\omega_K\}$ 是聚类的集合，ω_k 表示第 k 个聚类的集合，K 为总的聚类个数；$X=\{x_1,x_2,\cdots,x_n\}$ 是数据集合，x_n 表示第 n 个数据，N 表示数据总数，第 k 个类中数据点的个数为 N_k。则基于距离的 merge 过程如下：

①对第 $k(k=1,2,\cdots,K)$ 个类中的任意两个数据 x_i 和 x_j，求两点之间的距离 $d_{i,j}$：

$$d_{i,j}=|x_i-x_j| \tag{3}$$

②求第 $k(k=1,2,\cdots,K)$ 个类中的所有数据之间的平均距离 d_k：

$$d_k=\sum_{i,j=1}^{N_k}d_{i,j}\frac{1}{N_k(N_k-1)} \tag{4}$$

③求整个数据集所有点之间的平局距离 d：

$$d=\sum_{k=1}^{K}d_k\frac{1}{K} \tag{5}$$

④给定参数 T，对任意两个不同的类 $\omega_i(i=1,2,\cdots,k)$ 和 $\omega_j(j=1,2,\cdots,k)$，$\omega_i\neq\omega_j$，求得 ω_i 和 ω_j 中任意两点间的距离并取其最小值 $d_{\min}$；如果 $d_{\min}<T*d$，则将这两类合并；否则不合并；

⑤对所有的 m 类，按步骤④依次循环处理，直到结束为止。

3.2 CVM 压缩算法

2005 年，Tsang 提出了一种经典的 CVM[5] 分类算法，它

可以将单分类支持向量机的求解问题转换成一类特殊的最小包含球(MEB)问题,其核心思想是将大规模的数据进行压缩。

压缩过程如下:

①对数据集中的所有数据,随机取其中 3 个进行标记,由这 3 个数据点确定一个球,计算其球心 c,半径 r;

②随机取 59 个数据点(这 59 个点可以代表整个数据集,具体证明可以参考文献[5]),如果这 59 个点都在以 c 为圆心,r 为半径的球内,则算法结束;否则,计算 59 个点中距离圆心 c 最远的点标记,同时计算最远的点与点 c 的中点最新的球心,距离的一半作为新的半径。

③重复步骤②,直到所有的点全部包含在球内,算法结束。此时所有标记过的点可以代表整个数据集。

在聚类算法中,当数据集为大样本数据集时,算法的运算速度很慢,这时可将 CVM 压缩过程融入其中进行分类,从而有效地节省运算时间。同时,聚类结束后,计算原大样本数据集中所有点与压缩后已经分类出的点的距离,距离哪个点最近,就将其与这个点归为一类。

3.3 M-AP 算法

由于 AP 算法对聚类结构比较复杂(非团状)的数据集聚类效果不好,分出来类的数目远远地超过了聚类数据集本身的数目。本文将 merge 过程融入 AP 算法中,提出了一种改进的 AP 算法(称为 M-AP 算法),有效地解决了上述问题。

M-AP 算法基本过程如下:

第一步　初始数据处理:对初始数据集进行处理,当数据集为大样本时,用 CVM 算法对其进行压缩,得到新的数据集;

第二步　数据聚类:对第一步得到的数据集利用 AP 算法进行聚类,得到 m 类数据;

第三步　合并聚类:对聚类好的数据,调用基于距离的 merge 过程对其进行处理,得出实验结果。

在以上算法中,参数 T 的设置影响着算法的效果。当 T 很小时,M-AP 算法退化为 AP 算法;当 T 很大时,则容易把不同的类合并,从而会得到不好的效果。另外,为了减少距离的计算量,保证数据的不溢出,可以在第一步中对聚类数据集先进行归一化处理。研究表明,在正常情况下,T 的最佳取值范围是[0.5,2.0]。

4 算法优劣性指标

为衡量聚类算法的优劣,引入以下 4 个指标。

(1)芮氏指标(Rand Index,RI)[6,7]:

$$RI=\frac{2(f_{00}+f_{11})}{N(N-1)} \tag{6}$$

其中,f_{00} 表示数据点具有不同的类标签且属于不同类的配对点数目,f_{11} 则表示数据点具有相同的类标签且属于同一类的配对点数目,而 N 表示整个数据样本的总量。该指标把准确率和召回率看得同等重要。

(2)精度 ACC [8,9] 如下:

$$ACC=\frac{\sum_{i=1}^{N}\delta(y_i,map(c_i))}{N} \tag{7}$$

这里 N 是数据点的个数,y_i 和 c_i 分别表示真实数据标签和所获得的聚类标签。其中 $\delta(y,c)$ 是这样一个函数:当 $y=c$ 时,函数值为 1;否则为 0;而 $map(\cdot)$ 是一个排列函数,它将每一个聚类标签和类标进行匹配,最优匹配结果详见 Hungarian 算法[10]。

(3)标准化互信息 NMI[11]:

$$NMI=\frac{\sum_{i=1}^{K}\sum_{j=1}^{K}N_{i,j}\log\frac{N\cdot N_{i,j}}{N_i\cdot N_j}}{\sqrt{\sum_{i=1}^{K}N_i\log\frac{N_i}{N}\cdot\sum_{j=1}^{K}N_j\log\frac{N_j}{N}}} \tag{8}$$

式中,K 表示聚出类的个数,$N_{i,j}$ 表示聚类结果第 i 类中和真实标签 j 中共同数据的个数,N_i 表示在聚类 i 中数据的个数,N_j 表示在类 j 中数据的个数。N 表示整个数据集中数据的个数。NMI(NMI∈[0,1])的值越高,表明相应的聚类算法得到的聚类结果越合理。

(4)Purity 指标:

Purity 指标是只需计算正确聚类的数据个数占总数据数的比例:

$$purity(\Omega,X)=\frac{1}{N}\sum\max_j|\omega_k\cap x_j| \tag{9}$$

其中,$\Omega=\{\omega_1,\omega_2,\cdots,\omega_K\}$ 是聚类的集合,ω_k 表示第 k 个聚类的集合;$X=\{x_1,x_2,\cdots,x_n\}$ 是数据集合,x_j 表示第 j 个数据,N 表示数据总数。该指标的值在 0～1 之间,值为 0 时表示聚类完全错误,值为 1 时表示聚类完全正确。

Purity 方法是一种极为简单的聚类评价方法,其优势是方便计算;其缺点是无法对退化的聚类算法给出正确的评价,例如,假设某聚类算法把每篇文档单独聚成一类,那么算法认为所有文档都被正确分类,此时 Purity 值为 1,而这显然不是想要的结果。

5 实验与结果分析

5.1 人工数据集实验

构造非团状人工数据集 Test,如图 1 所示。数据集 Test 中有 30000 个数据点,对其用 CVM 压缩成 65 个点(见图 2)。AP 聚类算法的迭代曲线如图 3 所示,通过 AP 算法(p 取均值)与 M-AP 算法(取参数 $T=2$)得到的聚类结果如图 4 与图 5 所示,数据集所有点在采用 M-AP 聚类算法后恢复结果的聚类效果如图 6 所示。

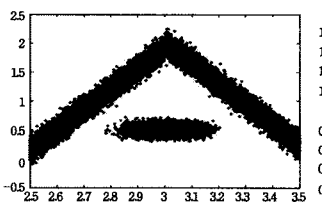

图 1　人工数据集 Test

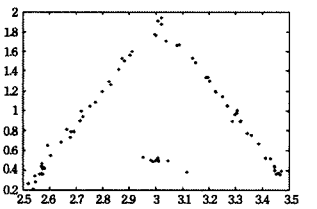

图 2　CVM 压缩后的数据集

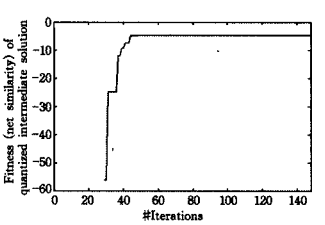

图 3　AP 聚类算法的迭代曲线

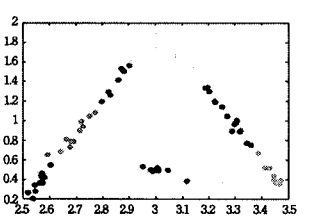

图 4　AP 算法聚类结果

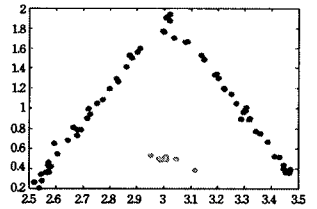

图 5 M-AP 算法聚类结果(参数 $T=2$)

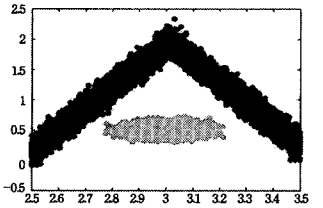

图 6 M-AP 聚类算法压缩后恢复结果

由图可知,对非团状人工数据集 Test,AP 聚类算法将数据集分为 7 类,这显然是错误的,而在参数 $T=2$ 的 M-AP 聚类算法中,数据集则被正确地分为两类。由此表明,M-AP 聚类算法可以有效地解决 AP 聚类算法在非团状数据集中聚类不准确的问题,对非团状数据集有很好的聚类效果。

5.2 常用数据集实验

本节考虑 4 个常用数据集的聚类,其中两个是非团状的,另两个是团状的。将本文 M-AP 聚类算法与 K-means 聚类算法、FMC 模糊聚类算法、基于密度的 DBSCAN 算法[20,21](代码来源于[24])及 AP 聚类算法进行比对,用上述 5 个聚类评价指标 RI(RandIndex)、ACC(accuracy)、NMI 及 Purity 来衡量算法的优劣性。

实验中,每个算法执行 20 次,求得 4 个指标的均值和标准差,其中标准差为 0.0000 表示标准差很小,但不是真正为 0;而标准差为 0 表示标准差为 0;粗体字表示所有对比算法中此指标的最大值。

由实验结果可以轻松看出,*K*-means 算法和 FCM 算法是不稳定的算法,而 AP 算法和 M-AP 算法是稳定的算法。

5.2.1 Optdigits 数据集[11]

Optdigits 是非团状数据集,包含 1263 个点,分为 10 类。5 种算法对该数据集的聚类性能如表 1 所列。

表 1 optdigits 数据集上 5 种算法的评价指标($T=0.8$,p 取均值)

	K-means (k=10)	FCM (k=10)	DBSCAN	AP	M-AP
RRI	0.9311± 0.0115	0.7698± 0.0288	0.0993± 0	0.9131± 0	0.9588± 0
ACC	0.7553± 0.0553	0.3728± 0.0333	0.1037± 0	0.2122± 0	0.8195± 0
NMI	0.7451± 0.0206	0.4031± 0.0260	0.0000± 0	0.6686± 0	0.8495± 0
Purity	0.8078± 0.0230	0.7099± 0.0395	1± 0	0.2118± 0	0.8856± 0

实验中,AP 算法将 optdigits 数据集分为 81 类,M-AP 算法较准确地分为 17 类,而 K-means 算法和 FCM 算法也需要预先设置好类别。另外,从表 1 可以看出,DBSCAN 算法在此数据集上表现很差,而 M-AP 算法相对于 K-means、FCM 及 AP 算法在所有指标均有很大的提升,这说明对非团状 optdigits 数据集,M-AP 算法有着非常好的聚类效果和性能。

5.2.2 Iris 数据集[12]

Iris 数据集是团状数据集,包含 150 个数据,并分为 3 类。5 种算法的聚类性能如表 2 所列。

表 2 Iris 数据集上 5 种算法的评价指标($T=0.5$,p 取均值)

	K-means(k=3)	FCM(k=2)	FCM(k=3)	FCM(k=4)	DBSCAN	AP	M-AP
RI	0.8394±0.0708	0.7637±0.0000	0.8797±0.0000	0.8410±0.0074	0.7724±0	0.8406±0	0.8650±0
ACC	0.8117±0.1451	0.6667±0.0000	0.8933±0.0000	0.7107±0.0098	0.6800±0	0.6600±0	0.8400±0
NMI	0.7150±0.0768	0.6565±0.0000	0.7496±0	0.7062±0.0089	0.6900±0	0.6751±0	0.7518±0
Purity	0.8817±0.0207	0.9800±0.0000	0.8933±0	0.7107±0.0098	0.9733±0	0.6600±0	0.8400±0

本实验中,AP 算法将 Iris 数据集分为 6 类,M-AP 算法分为 4 类,而 K-means 和 FCM 算法需要预先设置好类别。由表 2 可以看出,在聚类评价指标 RI、AC、NMI 及 Purity 方面,M-AP 算法相对于 AP 算法都有一定程度的提升,而仅次于 $k=3$ 时的 FCM 算法。但是 K-means 算法和 FCM 算法需要预先指定 k 的值,当 k 不为 3 时效果不好。而且由表 2 可以看出,K-means 算法很不稳定,效果浮动很大。DBSCAN 算法相对于 AP 算法在 ACC 和 Purity 指标上有一定提升,但是相对于 M-AP 算法却不占优势。上述实验结果说明,对团状数据集,M-AP 算法具有良好的聚类效果。

5.2.3 Clean 数据集[13]

Clean 也是非团状数据集,包含 476 个数据,并分为 2 类。5 种算法对该数据集的聚类性能如表 3 所列。

表 3 Clean 数据集上 5 种算法的评价指标($T=1.5$,p 取均值)

	K-means (k=2)	FCM (k=2)	DBSCAN	AP	M-AP
RI	0.5017± 0.0024	0.4995± 0.0000	0.5049± 0	0.5044± 0	**0.5064±** 0
ACC	0.5329± 0.0171	0.5168± 0.0000	0.5546± 0	0.0945± 0	**0.5610±** 0
NMI	0.0092± 0.0069	0.0031± 0.0000	0.0077± 0	**0.1255±** 0	0.0068± 0
Purity	0.6599± 0.0504	0.6155± 0.0000	0.9864± 0	0.0964± 0	**0.9963±** 0

对 Clean 数据集,AP 算法将其分为 28 类,M-AP 算法可准确地分为 2 类,而 K-means 算法和 FCM 算法需要预先设置好类别,M-AP 算法聚类结果准确。

从表 3 可以看出,与 AP 算法相比,M-AP 算法在评价指标 AC 和 Purity 上均有很大的提升,在指标 RI 上提升不明显,而在指标 NMI 上要差于 AP 算法。与 K-means 算法、FCM 算法和 DBSCAN 算法相比,M-AP 算法的聚类性能也占一定的优势。这进一步说明对于非团状数据集,M-AP 算法聚类结果和性能要好于其他方法。

5.2.4 Soybean 数据集[14]

Soybean 是团状数据集,包含 47 个点,并分为 4 类。5 种算法对该数据集的聚类性能如表 4 所列。

表 4 Soybean 数据集上 5 种算法的评价指标($T=0.5$,p 取均值)

	K-means (k=4)	FCM (k=4)	DBSCAN	AP	M-AP
RI	0.8329± 0.0444	0.8316± 0.0000	0.32507± 0	**0.8816±** 0	0.8594± 0
ACC	0.7138± 0.0948	0.7234± 0.0000	0.3617± 0	0.7447± 0	**0.8293±** 0
NMI	0.7269± 0.0802	0.7158± 0.0000	0.0000± 0	**0.8111±** 0	0.8099± 0
Purity	0.7838± 0.0683	0.7574± 0.0000	1± 0	0.7824± 0	**0.8824±** 0

(下转第 267 页)

similarity between words using multiple information sources [J]. IEEE Transaction on Knowledge and Data Engineering, 2003,15(4):871-882

[20] Turney P D. Features of similarity[J]. Psychological Review, 1997,84(4):327-352

[21] Chen H, Lin M, Wei Y. Novel association measures using web search with double checking[C]// Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. 2006:1009-1016

[22] Sahami M, Heilman D. A Web-based kernel function for measuring the similarity of short text snippets[C]// Proceedings of the 15th International World Wide Web Conference. ACM Press: New York, NY, 2006:377-386

[23] Islam A, Inkpen D. Second order co-occurrence PMI for determining the semantic similarity of words[C]// Proceedings of the International Conference on Language Resources and Evaluation. 2006:1033-1038

[24] Bollegala D, Matsuo Y, Ishizuka M. Measuring semantic similarity between words using web search engines[C]// Proceedings of 16th International World Wide Web Conference. ACM Press: New York, NY, 2007:757-766

[25] Firth R. A synopsis of linguistic theory 1930-1955[D]. Studies in Linguistic Analysis, Philological Society: Oxford, 1957

[26] Bayardo R J, Ma Y, Srikant R. Scaling up all pairs similarity search[C]// Proceedings of 16th International World Wide Web Conference. ACM Press: New York, NY, 2007:131-140

[27] Rubenstein H, Goodenough B. Contextual correlates of synonymy[J]. Communications of the ACM, 1965, 8(10):627-633

[28] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[C]// Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. Vol. 22, ACM Press: New York, NY, 1993:207-216

[29] Church W, Hanks P. Word association norms, mutual information and lexicography[C]// Proceedings of the 27th Annual Conference of the Association of Computational Linguistics. 1989:76-83

(上接第 235 页)

实验中，AP 算法将 Soybean 数据集分为 5 类，M-AP 算法将其准确地分为 4 类，而 K-means 算法和 FCM 算法也需要预先设置好类别。另外，从表 4 可以看出，DBSCAN 算法在此数据集中表现很差，而 M-AP 算法相对于 AP 算法在指标 RI 和 NMI 上略有下降，在指标 AC 和 Purity 上均有很大的提升。而 M-AP 算法的所有指标均明显优于 K-means 算法和 FCM 算法。

由上述 4 个实验可以看出，M-AP 算法在整体上相对于 AP 算法、K-means 算法、FCM 算法以及 DBSCAN 算法都有一定的优势，有效地解决了某些算法（AP 等）并不适用于非团状数据集的问题，并可以得到每个聚类的聚类中心代表点（DBSCAN 等无法得到），具有相当高的实用价值。

结束语 针对 AP 聚类算法无法正确地处理非团状数据集，提出了一个基于 AP 聚类算法的新的聚类 M-AP 算法。该方法将 merge 过程拓展至 AP 聚类算法中，解决了 AP 算法对非团状数据聚类效果不好的问题，而对团状数据仍有着较好的支持。对大规模数据的聚类，M-AP 算法可以先采用压缩算法对数据集进行压缩，然后再进行聚类，来取得很好的效果，从而为数据的聚类提供了一个可靠且有效的解决方案。

参 考 文 献

[1] Frey B J, Dueck D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814):972-976

[2] Pollard D. Strong consistency of Kmeans clustering[J]. Ailnals of Statistics, 1981, 9(1):135-140

[3] Zhang T, Ramakrishnan R, Livny M. BIRCH. An efficient data clustering method for very large databases[J]. Montreal, 1996, 6(96):103-114

[4] Pal N R, Bezdek J C. On cluster validity for the fuzzy c-means model[J]. IEEE Transactions on Fuzzy Systems, 1995, 3(3):370-379

[5] Tsang I W, Kwok J T, Cheung P M. Core vector machines: fast SVM training on very large data sets[J]. Journal of Machine Learning Research, 2005, 8(6):363-392

[6] Deng Zhao-hong, Choi K S, Chung F L, et al. Enhanced soft subspace clustering integrating within cluster and between bluster-Information[J]. Pattern Recognition, 2010, 43(3):767-781

[7] Liu Jun, Mohammed J, Carter J, et al. Distance based clustering of CGH data[J]. Bioinformatics, 2006, 22(16):1971 -1978

[8] Chen W Y, Song Y, Bai H, et al. Parallel spectral clustering in distributed systems[J]. IEEE Transactions onPattern Analysis and Machine Intelligence, 2011, 33(3):568-586

[9] Wu M, Schölkopf B. A local learning approach for clustering[J]. Proc. Conf. Neural Information Processing Systems, 2007(1):1529-1536

[10] Papadimitriou C H, Steiglitz K. Combinatorial Optimization: Algorithms and Complexity[M]. Dover Publications, 1998

[11] Optdigits数据集[OL]. https://archive.ics.uci.edu/ml/machine-learning-databases/optdigits/

[12] Iris 数据集[OL]. http://archive.ics.uci.edu/ml/datasets/Iris

[13] Clean 数据集[OL]. http://archive.ics.uci.edu/ml/machine-learning-databases/musk/

[14] Soybean 数据集[OL]. http://archive.ics.uci.edu/ml/datasets/Soybean+(Small)

[15] 孙吉贵，刘杰，赵连宇. 聚类算法研究[J]. Journal of Software, 2008, 19(1):48-61

[16] 肖宇，于剑. 基于近邻传播算法的半监督聚类[J]. Journal of Software, 2008, 19(11):2803-2813

[17] 牟廉明，詹德川，黎铭，等. 基于结构相似性和压缩变换的聚类方法[J]. Pattern Recognition and Artificial Intelligence, 2011, 24(5):637-644

[18] 于剑，程乾生. 模糊聚类方法中的最佳聚类数的搜索范围[J]. 中国科学 E 辑，2002, 32(2):274-280

[19] 杨善林，李永森，胡笑旋，等. K-means 算法中的 k 值优化问题研究[J]. 系统工程理论与实践，2006, 26(2):97-101

[20] 周水庚，周傲英，曹晶. 基于数据分区的 DBSCAN 算法[J]. 计算机研究与发展，2000(10):1153-1159

[21] Ester M. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, Portland: AAAI Press. 1996:226-239

[22] 计华，张化祥，孙晓燕. 基于最近邻原则的半监督聚类算法 [J]. 计算机工程与设计，2011, 32(7):2455-2459

[23] 李昆仑，曹铮，曹丽苹. 半监督聚类的若干新进展[J]. 模式识别与人工智能，2009(5):735-742

[24] DBSCAN 算法代码[OL]. http://wenku.baidu.com/view/47a26ebbba0d4a7302763a9c.html