基于近邻传播的限定簇数聚类方法研究

李海林[1,2]，魏 苗[1]

(1. 华侨大学工商管理学院 福建 泉州 362021；2. 华侨大学现代应用统计与大数据研究中心 福建 厦门 361021)

【摘要】针对传统近邻传播聚类算法不能进行限定类簇数目的聚类缺陷，提出一种三阶段的改进聚类方法。该方法通过近邻传播聚类从数据集中获得中心代表点集合，利用*K*-means算法对中心代表点集合进行指定类簇数目的聚类进而获得初始训练集，结合改进的*K*最近邻算法实现数据的聚类分析。采用人工仿真数据及UCI数据集进行对比实验，实验结果分析表明，与近邻传播聚类算法和传统限定类簇数目的聚类算法相比，新聚类算法具有更好的聚类效果。

关 键 词 近邻传播；聚类算法；类簇数目；数据挖掘；*K*均值聚类

中图分类号 TP301 **文献标志码** A doi:10.3969/j.issn.1001-0548.2018.05.015

Research on Clustering Method with Specified Cluster Number Based on Affinity Propagation

LI Hai-lin[1,2] and WEI Miao[1]

(1. School of Business Administration, Huaqiao University Quanzhou Fujian 362021;
2. Research Center of Applied Statistics and Big Data, Huaqiao University Xiamen Fujian 361021)

Abstract Due to disadvantage of the affinity propagation algorithm of which the number of clusters can not be pre-specified, an improved method including three phases is proposed in this paper. The proposed method uses affinity propagation algorithm to obtain the representation center points of the dataset. Then *K*-means is applied to the clustering of the center points and produces the initial training set. Moreover, the modified *K* nearest neighbor algorithm is applied to the procedure of clustering analysis. Artificial data and UCI datasets are used in experiment to compare the new algorithm with other clustering menthes. The results demonstrate that the new clustering algorithm is outperforms the affinity propagation algorithm and traditional clustering algorithms.

Key words affinity propagation; clustering algorithm; clusters number; data mining *K*-means clustering

聚类分析是一种在机器学习领域中对数据进行分析的有效方法，在数据挖掘与知识发现领域中也具有不可忽视的作用。通过聚类算法对数据进行分类分析，将一个数据集划分为若干个簇，使得每个簇中数据对象尽可能相似，而簇间数据对象尽可能相异。特别地，在大数据时代，海量数据通常不具有类标签，使得这种无监督的机器学习方法变得更加重要。另外，目前也存在适用于各类数据形态分布的聚类算法，如划分聚类、层次聚类、基于密度的聚类和基于模型的聚类等[1-2]，它们已经广泛应用在数据预测模式、人工智能、图像识别等相关领域中[3-5]。

在聚类分析中，经典的*k*-means和*k*-centers等方法通过多次迭代重新计算聚类中心得到最优聚类结果，但是在传统算法中初始聚类中心的选择对算法聚类结果和算法的迭代次数影响较大，不同的初始聚类中心经常会导致不同的聚类结果[6]。在过去的十几年中，部分学者提出了各种改进方法以提高聚类算法的效率与准确率[7]。特别地，文献[8]提出的基于图论的近邻传播算法(affinity propagation, AP)有效解决了初始聚类中心的选择对聚类结果产生的问题，是一种聚类质量和效率较好的聚类分析方法，已在多个领域得到了应用[9-11]，并有不少学者对它进行了研究[12-13]。

近邻传播与传统聚类算法相比，对数据间的相似性矩阵的输入没有特殊要求从而扩大了算法的适应性[14-15]。另外，近邻传播聚类不要求算法进行初始聚类中心的选择，避免了由初始中心产生的不利

收稿日期：2017－01－12；修回日期：2017－09－17
基金项目：国家自然科学基金(71771094, 61300139)；福建省社会科学规划基金(FJ2017B065)
作者简介：李海林(1982－)，男，博士，副教授，主要从事数据挖掘与决策支持方面的研究.

影响，使得结果具有一定的确定性。然而，传统近邻传播聚类算法无法限定目标聚类结果的类别数目，需要人工多次调整近邻传播算法的偏向参数值来达到指定的分类数目[16]，限制了算法使用的灵活性和应用范围。鉴于近邻传播聚类算法无法实现限定分类数目的问题，本文提出一种基于近邻传播的新聚类算法，该方法通过AP聚类将原始数据集进行自适应中心代表点选择，再利用K均值将中心代表点聚类成指定的类数，最后提出改进后的近邻分类算法实现所有数据对象的聚类。该方法不仅能对AP聚类算法的结果进行确定类数分析，以便产生指定类数的聚类结果，还能提高AP聚类结果的质量。数值实验表明，相较对比算法，新方法具有更好的聚类效果。

1 近邻传播算法

近邻传播AP聚类算法的聚类基础是数据间的相似性度量矩阵 $s(i,k)$，其表示数据点 x_k 在多大程度上适合作为数据点 x_i 的类代表点。通常采用欧式距离作为相似性的测度指标，即任意两点之间的相似性定义为两点间距离平方的负数。如，数据点 x_i 和数据点 x_k 的相似性表示为 $s(i,k)=-\|x_i-x_k\|^2$。在相似性矩阵中，$s(k,k)$ 的值被称为preferences，即偏向参数，通常取对应行的中位数作为偏向参数值。

在迭代的过程中，代表度和有效性两种信息在数据中传递，两种信息代表了不同的竞争目的。代表度 $r(i,k)$ 是从数据点 x_i 传到候选代表点 x_k 的信息，反映在比较了其他点 x_i 的候选代表点之后点 x_k 作为点 x_i 的代表点的合适程度。有效性 $a(i,k)$ 是从候选代表点 x_k 传递到数据点 x_i 的信息，反映了考虑到其他点对点 x_k 的支持度后点 x_k 作为点 x_i 的代表点的有效程度。AP算法在迭代过程中，不断更新每个数据点的代表度和有效性的值，直到产生收敛的聚类结果。AP聚类算法的计算步骤如下：

近邻传播聚类算法：$\boldsymbol{C}=\mathrm{AP}(\boldsymbol{X})$

输入：原始数据集 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_n\}$

输出：聚类结果 $\boldsymbol{C}$

1) 初始化代表性矩阵和有效性矩阵为零矩阵；

2) 根据 $s(i,k)=-\|x_i-x_k\|^2$ 计算相似性矩阵 $\boldsymbol{S}$，其中 $s(k,k)=\mathrm{median}(\boldsymbol{S})$ 表示 $s(k,k)$ 的取值为相似矩阵 $\boldsymbol{S}$ 的中位数，有：

$$s(k,k)=\mathrm{median}(\boldsymbol{S})\quad k=1,2\cdots n \tag{1}$$

3) 更新代表性矩阵 $\boldsymbol{R}$，更新规则为：

$$\begin{cases} r(i,k)=s(i,k)-\max\limits_{k'\neq k}\{a(i,k')+s(i,k')\} & i\neq k \\ r(k,k)=s(k.k)-\max\limits_{k'\neq k}\{a(k,k')+s(k,k')\} & i=k \end{cases} \tag{2}$$

4) 更新有效性矩阵 $\boldsymbol{A}$，更新规则为：

$$\begin{cases} a(i,k)=\min\{0,r(k,k)+\sum\limits_{i'\notin\{i,k\}}\max\{0,r(i',k)\}\} & i\neq k \\ a(k,k)=\sum\limits_{i'\neq k}\max\{0,r(i',k)\} & i=k \end{cases} \tag{3}$$

5) 迭代步骤3)和步骤4)，当迭代次数超过最大迭代次数或者当产生收敛的聚类结果时停止计算；

6) 若 $r(k,k)+a(k,k)>0$，则数据点 x_k 为聚类中心；

7) 将剩余数据点分配到相应的聚类中心；

8) 聚类结束。

在某些特殊情况下AP聚类会发生数据震荡，即使进行了大量迭代计算也无法产生准确的聚类结果。为了解决数据震荡，AP聚类引入了阻尼因子(damping factor)。从第二次迭代开始，$r(i,k)$ 和 $a(i,k)$ 的值都是由当前迭代计算的新值和上一步迭代的值进行加权计算得到的。如当前迭代次数为 i，则加权公式为：

$$r_i(i,k)=\lambda r_i+(1-\lambda)r_{i-1} \tag{4}$$

$$a_i(i,k)=\lambda a_i+(1-\lambda)a_{i-1} \tag{5}$$

在计算过程中迭代的次数受到阻尼因子 λ 的影响，当 λ 的值较小时，迭代次数较多；若 λ 取较大值，迭代次数也会减少。

2 限定簇类聚类算法

近邻传播算法的聚类结果不局限于初始代表点的选择，在数据聚类上具有较好的性能。然而，近邻传播在算法结束前无法得知结果中的聚类簇数，如需某个具有 k 个类簇的聚类结果就要通过调整偏向参数进行多次聚类直到产生较为理想的结果为止[17]。文献[18]设计出一种自动调整参数的adAp算法以自动产生最佳聚类结果，但是由于进行了多次AP聚类的迭代，adAP算法的时间代价过大，使得adAP算法在较大规模的数据集上无法取得很好应用效果。鉴于此类情况，本文提出在近邻传播算法的聚类结果的基础之上进行确定类数分析的聚类方法(affinity propagation with restricted number of clusters, AP-RNC)，其在保证AP聚类原有的高质量聚类结果上，通过再分类过程来实现聚类结果中类簇数目的限定，并且使得新算法以较小的时间代价取得更高质量的聚类结果。

2.1 簇中心代表点

AP-RNC算法首先对数据集 $\boldsymbol{X}$ 进行一次AP聚类，将聚类代表点记为 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$。由于AP聚类能够产生高质量的聚类结果，代表点集合 $\boldsymbol{C}$ 对原始数据的代表程度高，因此在减少算法下一阶段计算量的同时使用AP聚类算法提取数据中的代表点能将原始数据中的信息大幅度的保留下来。另外，提取AP聚类的聚类代表点作为算法基础的另一个目的是降低噪声数据对聚类结果的干扰。当噪声数据较多时，使用代表点集合 $\boldsymbol{C}$ 取代原始数据集能够减少噪声数据的数量，降低噪声数据对算法下一阶段的干扰。值得注意的是，为了保证算法在下一阶段的可行性，在进行AP聚类时需将AP 算法中的偏向参数设为一个较大的值(这里取相似性矩阵的70%分位数)以产生类簇较多的聚类结果。

AP-RNC算法的第二阶段是*K*-means聚类阶段。在*K*-means聚类阶段，AP-RNC算法将第一阶段产生的代表点集合 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$ 利用*K*-means聚类算法划分成指定的 k 类，并记录每个代表点所属的类别。*K*-means聚类阶段的目的是生成类簇数目为 k 的训练数据集(即使用*K*-means聚类后的代表点集合 $\boldsymbol{C}$)，将该训练数据集作为改进后的 k 最近邻算法的初始训练数据集。*K*-means聚类阶段的具体过程描述如下：

K-means聚类阶段： $\boldsymbol{F}=K\text{ means}(\boldsymbol{C},k)$

输入：AP 聚类阶段产生的代表点集合 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$，聚类结果中的类簇数目 k

输出：初始训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$

1) 对输入的代表点集合 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$ 进行一次限定类簇数目为 k 的 *K*means 聚类；

2) 将步骤 1)中的聚类结果记为初始训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$，实际上，$\boldsymbol{F}$ 即为带有分类标签的代表点集合 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$；

3) *K*-means 聚类阶段结束。

AP-RNC算法的前两个阶段通过对数据进行AP聚类与*K*-means聚类分析，提取出数据集中具有代表性的点，并将代表点分成 k 类，为后续算法提供分类依据。

2.2 改进后的*K*最近邻算法

经过AP聚类阶段和*K*-means聚类阶段，AP-RNC算法进入第三阶段：再分类阶段。AP-RNC算法通过再分类阶段完成对聚类结果的类簇数目限定，同时本阶段提出一种改进的 k 最近邻算法(improved *K*-nearest neighbor algorithm, IKNN)。

上述两阶段计算过程产生的训练数据集较为粗糙，如直接使用该数据集进行 k 最近邻聚类产生的聚类结果偏差较大，因此这里提出一种改进的 k 最近邻算法。在传统的 k 最近邻算法中，训练数据集作为先验知识在算法中起作用，未知类别数据的分类由训练数据集中距离自己最近的数据决定。在 k 最近邻算法的整个过程中，作为分类依据的训练数据集没有发生变化。与传统的 k 最近邻算法不同，改进后的 k 最近邻算法中训练数据集是不断扩大的，一旦一个未分类的点分类完毕后，这个数据就会作为已知分类的数据加入到训练数据集中，成为下一个数据点的分类依据。

IKNN聚类算法： $\boldsymbol{C}=\text{IKNN}\ (\boldsymbol{X},\boldsymbol{F},k)$

输入：原始数据集 $\boldsymbol{X}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n\}$，初始训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$,聚类结果中的类簇数目 k

输出：聚类结果 $\boldsymbol{C}$

1) 取出一待分类的点 x_i，计算 x_i 与初始训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$ 中各点的距离；

2) 将 x_i 加入到初始训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$ 中与 x_i 距离最近的点 $\boldsymbol{F}_j$ 所属类簇中；

3) 将已分类的点 x_i 加入初始训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$ 中，使得 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w,\boldsymbol{F}_{w+1}\}$。

4) 重复进行步骤 1)，步骤 2)和步骤 3)，直到没有待分类数据为止。

改进后的 k 最近邻算法通过将分类完成的点加入到训练数据集中，解决了传统 k 最近邻算法在使用时由于训练样本过少而导致聚类结果的偏差，进一步减小噪声点的干扰能力，在提升聚类准确性方面做出了贡献。

2.3 AP-RNC算法

AP-RNC算法在第三个阶段将前两个阶段产生的分 k 个类簇的聚类代表点 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$ 视作训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$，将剩余的点视作待分类的点，使用改进后的 k 最近邻算法将待分类的点分成限定的 k 类。

AP-RNC算法的再分类阶段过程描述如下：取出任意一个待分类的点 x_i,计算点 x_i 与训练数据集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$ 之间的距离，将点 x_i 的分到与它距离最近的点 f_i 所属类簇中。点 x_i 的分类完成之后，将点 x_i 视作已分类的点加入到训练数据集

$\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$ 中，使 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w,\boldsymbol{F}_{w+1}\}$。以此类推，直到所有待分类的点都完成分类，加入到训练数据集 $\boldsymbol{F}$ 中，AP-RNC算法获得最终聚类结果 $\boldsymbol{C}$。

如图1所示，AP-RNC算法使用三阶段过程有效地对AP聚类产生的结果进行了指定类数分析，算法第一阶段对待分类的数据集进行AP聚类，产生聚类中心点集 $\boldsymbol{C}$。第二阶段使用*K*-means算法将聚类中心点集 $\boldsymbol{C}$ 分为指定的*k*类，并将聚类结果记为初始训练集 $\boldsymbol{F}$。最后以初始训练集 $\boldsymbol{F}$ 和待分类数据集中剩余的数据点集 $\overline{\boldsymbol{X}}$ 为输入参数，执行一次IKNN算法，得出聚类结果。

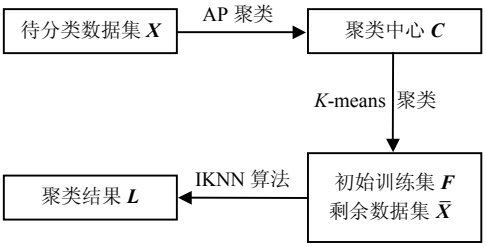

图1 算法流程图

AP-RNC聚类算法：$\boldsymbol{C}=\text{AP\_RNC}\ (\boldsymbol{X},k)$

输入：$\boldsymbol{X}$ 表示原始数据集，即 $\boldsymbol{X}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n\}$，聚类结果中的类簇数目 k

输出：聚类结果 $\boldsymbol{L}$

1) 将 AP 算法中的偏向参数设为相似性矩阵的70%分位数，对待分类数据集 $\boldsymbol{X}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n\}$ 进行一次 AP 聚类，记录产生的聚类中心 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$；

2) 对聚类中心 $\boldsymbol{C}=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_w\}$ 进行一次 *K*-means 聚类分析，获得初始训练集 $\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_w\}$；

3) 将剩余的数据点记为 $\overline{\boldsymbol{X}}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_{n-w}\}$ 执行一次 $\text{IKNN}(\overline{\boldsymbol{X}},\boldsymbol{F},k)$ 算法。

4) 聚类结束，获得聚类结果 $\boldsymbol{L}=\{\boldsymbol{L}_1,\boldsymbol{L}_2\cdots,\boldsymbol{L}_k\}$。

与传统的 AP 聚类相比 AP-RNC 算法所增加的两个用于限定聚类簇数的阶段所使用的方法分别为 *K*-means 聚类方法与 IKNN 分类方法。由上述 *K*-means 聚类方法的阐述容易得出 *K*-means 方法的时间复杂度为 $O(wkt)$，其中 w 表示 AP 聚类阶段结束后所产生的类代表点的个数，且 $w\leqslant n$，n 表示数据集 $\boldsymbol{X}$ 中的数据个数。k 表示限定的聚类簇数，t 代表 *K*-means 算法的迭代次数。而 IKNN 分类阶段，对每一个非代表点进行一次分类操作，由于训练集在计算的过程中不断扩大，因此每个待分类点的分类操作的时间复杂度都依次递增。由此可得 IKNN 分类阶段的时间复杂度为 $O(0.5\times(n+w-1)(n-w+1))$，由于一般情况下 $w\ll n$，因此 IKNN 的时间复杂度可简化为 $O(0.5\times n^2)$。由上述分析可得 AP-RNC 算法相对于传统 AP 聚类算法的时间复杂度增加为 $O(wkt+0.5n^2)$。

3 实验与结果分析

3.1 仿真实验

利用人工数据集对AP-RNC算法进行噪声点抗干扰性检测。构造人工数据集Test1和数据集Test2。Test1由两类聚类数据和3个噪声点组成。Test2由两个均匀分布的方形聚类数据和3个孤立噪声点构成。Test1和Test2数据集均能反映算法的抗噪性能。使用*K*-means和AP-RNC对测试数据集进行指定类数为3聚类分析，实验结果如图2与图3所示。

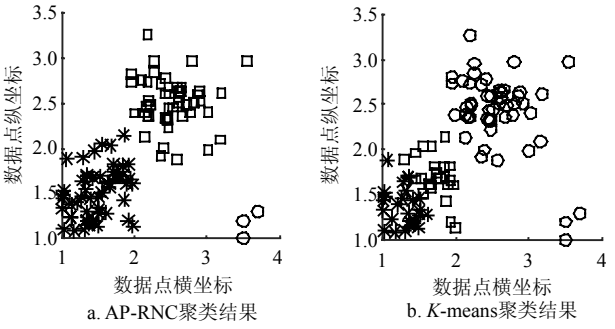

图2 AP-RNC算法与K-means算法在Test1的聚类结果

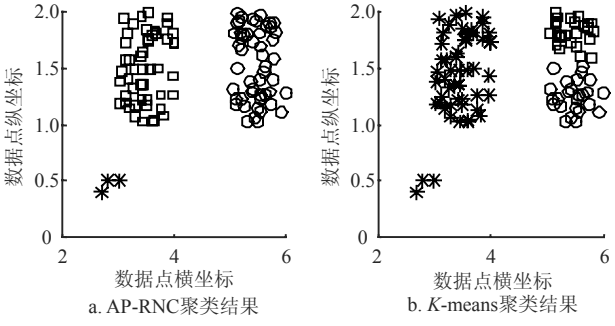

图3 AP-RNC算法与*K*-means算法在Test2的聚类结果

由实验结果可以看出，*K*-means算法的抗噪声性能较弱，当有噪声数据干扰时，*K*-means算法不能识别噪声数据，产生的聚类结果有明显偏差，而AP-RNC算法的聚类结果对噪声数据的敏感度更低，能够很好的识别噪声数据，并将噪声数据聚成单独的一类。

在试验过程中AP-RNC算法还表现出良好的抗偏向性能，当某一类中拥有多个相似数据时AP-RNC算法的聚类中心不会产生偏差从而影响聚类结果。

构造人工数据集Test3来进行算法的抗偏向性测试。Test3数据集由3个明显分类数据构成，第一类数据的左下角具有多个相似的数据点，即数据的局部密度较高。分别使用K-means算法和AP-RNC算法进行指定类数为3的聚类分析，实验结果如图4所示。

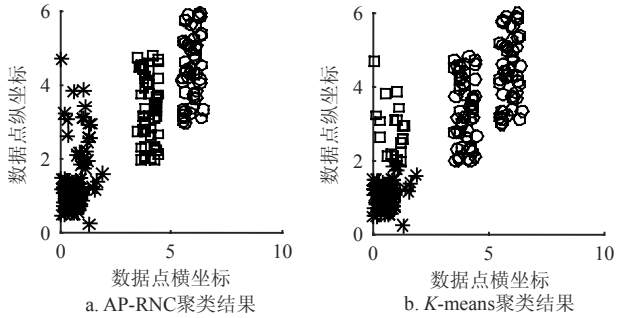

图4 AP-RNC算法与*K*-means算法在Test3的聚类结果

当数据集在某一类数据中拥有大量相似数据时，*K*-means算法有将大量相似数据分成一类且对该类周围较分散的数据不能准确聚类的倾向。在3次*K*-means算法聚类实验中，有1次会产生图5的聚类结果。AP-RNC算法产生的聚类结果不仅能对3类数据进行正确分类，在进行的5次算法测试中，没有产生偏差结果。

3.2　算法聚类比较

为了进一步分析新方法与传统方法的聚类效果，首先将AP-RNC算法与经典聚类算法进行对比。实验采用ICU数据集Iris、Wine、Soybean、Zoo和leuk72_3k共5个小规模数据集和Wine Quality、contraceptive、waveform、page-block共4个较大规模的数据集作为实验数据集实现聚类结果的分析与比较，以更加全面地验证算法性能，具体信息如表1所示。

表1　数据集信息

数据集	样本数	属性数	样本分类数
Iris	150	4	3
Wine	178	13	3
soybean	47	35	4
Zoo	101	16	7
leuk72_3k	72	39	3
Wine Quality	1 600	12	6
contraceptive	1 473	10	3
waveform	5 000	31	3
page-block	5 473	11	5

为了测试算法的准确性，引入两个衡量算法精确性指标：芮氏指标 (rand index, RI)[19] 和Folkes_Mallows指标(FM)[20]。当对两个不相关的聚类使用FM指标进行评价时，所参加评价的数据点越多，所得的FM值越接近于0，而RI指标在相同数据上的值迅速地接近1，这表明FM指标在对两个不相关的数据集上的准确性更高。此外当数据集中出现噪声点时FM指标的准确性也高于RI指标。

K-means算法是聚类分析中的经典算法，fuzzy-cmeans算法是基于模糊集理论提出的模糊聚类算法，clara是具有良好伸缩性的一种基于*K*-中心值的算法。在实验过程中，将新方法与*K*-means算法，fuzzy-cmeans算法和clara算法等传统方法进行比较，并且这4种算法都可以预先限定类簇数目，实验结果如表2所示。

表2　传统算法与AP-RNC算法的聚类结果比较

数据集	RI				FM			
	AP-RNC	*K*-means	fuzzy	clara	AP-RNC	*K*-means	fuzzy	clara
Iris	0.892	0.88	0.891	0.886	0.841	0.821	0.841	0.829
Wine	0.725	0.685	0.717	0.73	0.592	0.586	0.581	0.581
Soybean	0.843	0.763	0.774	0.85	0.686	0.557	0.673	0.696
leuk72_3k	0.964	0.707	0.766	0.898	0.945	0.616	0.729	0.845
Zoo	0.925	0.851	0.794	0.845	0.844	0.638	0.536	0.625
Wine Quality	0.558	0.587	0.596	0.581	0.318	0.268	0.258	0.271
Contraceptive	0.586	0.558	0.561	0.559	0.372	0.366	0.364	0.364
Waveform	0.667	0.667	0.535	0.666	0.509	0.504	0.529	0.502
page-block	0.764	0.608	0.352	0.395	0.841	0.739	0.464	0.522
均值	0.770	0.701	0.665	0.712	0.661	0.566	0.553	0.582
胜出率/%	66.67	0	11.11	22.22	77.78	0	11.11	11.11

由表2中的实验数据可以看出AP-RNC算法无论是在RI指标与FM指标上都能够在大多数数据集上获得优势聚类结果。前5个数据集为小型数据集，虽然AP-RNC算法在wine和soybean数据集上以微弱的差距输于clara算法，但是在剩余的3个数据集上AP-RNC算法均明显优于对比算法。后4个数据集中

的数据点较多，由于所使用的对比的算法均为限定类簇数目的聚类算法，因此RI指标和FM指标的准确性不受到类簇数目的影响。可以看出AP-RNC算法在所使用的两个指标上都表现出明显优势。综合算法的均值和胜出率来看，AP-RNC算法能够取得比传统限定聚类簇数算法质量更高的聚类结果。

在算法设计阶段也提出过一种两阶段分类算法AP_Kmeans算法，这种算法与AP-RNC算法类似，直接省去算法的第三阶段。将AP聚类产生的聚类代表点进行一次*K*-means聚类后，将剩余未分类的点根据AP聚类结果中所属类别的代表点分类，即各个类中的点的分类与各自分类的代表点相同，AP_Kmeans的优势在于能较迅速的对AP聚类的结果进行分类。为此，实验针对AP_Kmeans算法进行了相关的对比实验。

AP聚类算法自提出以来已被广泛的应用在各个领域，有不少学者对AP聚类算法进行了改进，本文选取发表于2015年的一种改进的AP聚类算法SAP算法[3]与AP-RNC算法进行对比，具体实验结果如表3所示。由表3可以看出在前5个小型数据集中AP-RNC算法在其中的4个数据集上都能获得优势聚类结果，而AP-Kmeans算法只在iris数据集上可以取得与AP-RNC持平的准确性。相比之下AP算法和SAP算法在经过多次对偏向参数的调节后得到的符合类簇数目的聚类结果后的聚类质量也相对较差。

在较大规模的数据集中，虽然AP聚类在RI指标上的数据占优势，但是AP算法往往不能得到准确的聚类簇数，要通过多次调整算法*p*值才能取得与实际数据集相近的类簇数目，较多的聚类簇数也在一定程度上使得RI指标更加接近于1。另外，由于数据集规模较大，AP算法在每次调整*p*值时所花费的时间代价较大。在FM指标上，AP聚类的聚类质量明显弱于AP-RNC算法，这也说明了AP-RNC算法的聚类结果更加接近真实数据集。AP-Kmeans和SAP算法在后4个数据集的表现不够突出，除了SAP算法在个别数据集上聚类效果较好，在剩余的数据集上取得的聚类结果相较对比算法并没有突出优势。从准确率的均值上看，虽然AP-RNC算法在RI指标上的结果略低于AP聚类，但其具有较高的胜出率，说明在大部分数据集中本文新方法AP-RNC算法能取得较好的聚类结果。与此同时，AP-RNC的FM评价指标明显优于AP_K-means , AP和SAP算法，这说明新聚类方法中的各阶段为提高聚类准确性提供了积极的作用。

表3 基于AP的聚类方法结果比较

数据集	RI				FM			
	AP-RNC	AP_Kmeans	AP	SAP	AP-RNC	AP_Kmeans	AP	SAP
iris	0.892	0.892	0.886	0.868	0.841	0.841	0.832	0.788
wine	0.725	0.663	0.720	0.717	0.592	0.539	0.586	0.581
soybean	0.843	0.852	0.850	0.500	0.686	0.657	0.695	0.409
leuk72_3k	0.964	0.964	0.898	0.875	0.945	0.945	0.845	0.793
zoo	0.925	0.869	0.885	0.855	0.844	0.710	0.737	0.680
Wine Quality	0.554	0.583	0.629	0.626	0.332	0.271	0.130	0.149
Contraceptive	0.556	0.610	0.649	0.634	0.372	0.237	0.234	0.146
Waveform	0.668	0.676	0.675	0.682	0.509	0.185	0.185	0.240
page-block	0.734	0.473	0.756	0.225	0.841	0.611	0.879	0.217
均值	0.762	0.729	0.772	0.665	0.662	0.555	0.569	0.598
胜出率/%	44.5	11.11	33.33	11.11	77.78	11.11	22.22	0

4 结 束 语

本文提出了一种对近邻传播算法进行确定类数分析的方法AP-RNC，该方法分别经过AP聚类阶段，*K*-means阶段和再分类阶段的计算，使得AP聚类产生的结果能够得到准确率较高的限定类数聚类结果。在保留了AP聚类产生的高质量聚类结果的基础上，用较小代价将聚类结果进行再分类，使得限定类簇的聚类准确性得到提高。新方法对异常值的敏感性较低，当局部数据点较密集时，不易产生聚类中心偏移，具有提高对边缘零散数据聚类准确率的优势。实验结果与分析表明，新方法具有更好的聚类质量。然而，由于近邻传播算法需要消耗较多计算时间，如何加快近邻传播聚类的速度以提高新算法的计算效率成为将来需要进一步研究的内容。

参 考 文 献

[1] XU Rui, DONALD W. Survey of clustering algorithms[J]. IEEE Transactions on Neural Networks, 2005, 16(3): 645-678.

[2] 陈黎飞, 姜青山, 王声瑞. 基于层次划分的最佳聚类数确定方法[J]. 软件学报, 2008, 9(1): 62-72.
CHEN Li-fei, JIANG Qing-shan, WANG Sheng-rui. A hierarchical method for determining the number of clusters[J]. Journal of Software, 2008, 9(1): 62-72.

[3] GAN Guo-jun, MICHAEL K. Subspace clustering using affinity propagation[J]. Pattern Recognition, 2015, 48(4): 1455-1464.

[4] 相洁, 赵冬琴. 改进谱聚类算法在MCI患者检测中的应用研究[J]. 通信学报, 2015, 36(4): 27-34.
XIANG Jie, ZHAO Dong-qin. Improved spectral clustering algorithm and its application in MCI detection[J]. Journal on Communications, 2015, 36(4): 27-34.

[5] SAGHABOZORGI S, SHIRKHORSHIDI A S, WAH T Y. Time-series clustering – a decade review[J]. Information Systems, 2015, 53(C):16-38.

[6] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1): 48-61.
SUN Ji-gui, LIU Jie, ZHAO Lian-yu. Clustering algorithms research[J]. Journal of Software, 19(1): 48-61.

[7] 周涛, 陆惠玲. 数据挖掘中聚类算法研究进展[J]. 计算机工程与应用, 2012, 48(12): 100-111.
ZHOU Tao, LU Hui-lin. Clustering algorithm research advances on data mining[J]. Computer Engineering and Applications, 2012, 48(12): 100-111.

[8] BREND F J, DELBERT D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.

[9] 李海林, 万校基, 林春培. 基于关键词重要性和近邻传播聚类的主题分析研究[J]. 情报学报, 2018, 37(5): 533-542.
LI Hai-lin, WAN Xiao-ji, LIN Chun-pei. Theme analysis based on keyword importance and affinity propagation clustering[J]. Journal of the China Society for Scientific and Technical Information, 2018, 37(5): 533-542.

[10] ARZENO N M, VIKALO H. Semi-supervised affinity propagation with soft instance-level constraints[J]. IEEE Transactions on Pattern Analysis and Machine Itelligence, 2015, 37(5): 1041-1052.

[11] HANG Wen-long, CHUANG Fu-lai, WANG Shi-tong. Transfer affinity propagation-based clustering[J]. Information Sciences, 2016, 34(8): 337-356.

[12] 李海林, 魏苗. 自适应属性加权近邻传播聚类算法[J]. 电子科技大学学报, 2018, 47(2): 247-255.
LI Hai-lin, WEI Miao. Affinity propagation clustering algorithm based on adaptive feature weight[J]. Journal of University of Electronic Science and Technology of China, 2018, 47(2): 247-255.

[13] 张震, 汪斌强, 伊鹏, 等. 一种分层组合的半监督近邻传播聚类算法[J]. 电子与信息学报, 2013, 35(3): 645-651.
ZHANG Zhen, WANG Bing-qiang, YI Peng, et al. Semi-supervised affinity propagation clustering algorithm based on stratified combination[J]. Journal Of Electronics & Information Technology, 2013, 35(3): 645-651.

[14] ZHANG Tao, WU Ren-biao. Affinity propagation clustering of measurements for multiple extended target tracking[J]. Sensors, 2015, 15(9): 22646-22659.

[15] ZHAO Xiu-li, XU Wei-xiang. An extended affinity propagation clustering method based on different data density types[J]. Computational Intelligence and Neuroscience, 2015, 1: 1-12.

[16] FUJITA A, TAKAHASHI D Y, PATRIOTA A G. A non-parametric method to estimate the number of clusters[J]. Computational Statistics & Data Analysis, 2014, 73(2): 27-39.

[17] 周世兵, 徐振源, 唐旭清. 一种基于近邻传播算法的最佳聚类数确定方法[J]. 控制与决策, 2011, 26(8): 1147-1152.
ZHOU Shi-bing, XUN Zhen-yuan, TANG Xu-qing. Method for determining optimal number of cluster based on affinity propagation clustering[J]. Control and Decision, 26(8): 1147-1152.

[18] 王开军, 张军英, 李丹, 等. 自适应仿射传播聚类[J]. 自动化学报, 2007, 33(12): 1242-1246.
WANG Kai-Jun, ZHANG Jun-ying, LI Dan, et al. Adaptive affinity propagation clustering[J]. Acta Automatica Sinica, 2007, 33(12): 1242-1246.

[19] RAND W M. Objective criteria for the evaluation of clustering methods[J]. Publications of the American Statistical Association, 1971, 66(336): 846-850.

[20] FOWLKES E B, MALLOWS C L. A method for comparing two hierarchical clusterings[J]. Publications of the American Statistical Association, 1983, 78(383): 553-569.

编 辑 蒋 晓