一种近邻传播的层次优化算法

倪志伟 荆婷婷 倪丽萍

(合肥工业大学管理学院 合肥 230009) (过程优化与智能决策教育部重点实验室 合肥 230009)

摘 要 近邻传播算法是一种新的聚类算法,在许多领域有较好的应用。近邻传播算法倾向于生成多于真实数目的类,且先验值 P 对该算法结果优劣有很大影响。故提出了一种有效的近邻传播的层次优化算法——CAP算法。CAP算法利用CURE算法对近邻传播算法的结果进行优化,是一种半监督的聚类算法。在5个UCI数据集上进行了实验验证,结果显示该算法均取得比近邻传播算法更好的聚类结果质量且使得生成的类的个数更接近真实类个数;同时与K-means、Spectral、CURE算法进行比较,结果表明CAP算法能取得更优的结果。

关键词 近邻传播算法,CURE算法,层次优化,先验值

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.3.040

Affinity Propagation Hierarchical Optimization Algorithm

NI Zhi-wei JING Ting-ting NI Li-ping

(School of Management, Hefei University of Technology, Hefei 230009, China)

(Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, China)

Abstract Affinity propagation (AP) clustering algorithm is a new clustering algorithm, and it is used in many fields well. Affinity propagation clustering algorithm tends to generate more classes than the real data sets. P has a great influence on the result. So this paper proposed an effective affinity propagation clustering's hierarchical optimization algorithm called as CAP. CAP algorithm uses the CURE algorithm to optimize the result of AP algorithm, and CAP is a semi-supervised clustering algorithm. The result of experiment on five UCI data sets shows that CAP algorithm achieves higher quality than AP algorithm and the number of classes is much closer to the real number. At the same time, CAP also achieves much better clustering result than K-means, Spectral and CURE.

Keywords Affinity propagation algorithm, CURE algorithm, Hierarchical optimization, Priori value

1 引言

聚类作为一种有效的数据分析方法,被广泛应用于机器学习、数据挖掘、模式识别、图像分割和生物信息处理等领域[1]。聚类算法主要是将数据集聚为若干类,且使得类中的数据具有较大相似性而类与类之间具有较大差异性。迄今为止,研究人员已经提出了许多聚类算法,著名的有K-means、K-medoids、BIRCH、CURE、DBSCAN等[1]。2007年,Fray等人首次在science杂志上提出了一种新的聚类算法——近邻传播聚类算法(Affinity Propagation Clustering, AP)[10]。与传统K-means算法不同,AP算法将每个点看作潜在的聚类中心,不需要事先指定聚类数目,通过更新迭代每个点的吸引度和归属度来产生聚类中心[2]。相对于K-means而言,AP算法不受限于初始代表点的选择,且不易陷入局部最优。AP算法在蛋白质分析、图像分割方面有较好应用,且实验证明,在人脸图像、基因识别、手写字符识别等问题达到相同的聚类结果时,近邻传播算法比K-means算法花费更少的时间[3,5,19,22]。

有学者证明根据迭代机制,AP算法作为基于质心的聚类算法,针对结构较松散的数据倾向于产生较多的类,在处理具有复杂结构的数据集时,不能够得到合理的聚类效果[21]。同时,AP算法对于先验值 P 比较敏感,在初始时赋予每个点的先验值 P 的大小很大程度上决定了聚类数目的多少[6,7,21]。

CURE算法是一种层次聚类算法,它是基于质心和代表点的策略,对非球形的几何形状数据有很好的效果,且该算法对控制孤立点有良好的效果[8,20]。但是CURE算法仍然需要事先预知聚类数目。

因此本文针对AP算法聚类结果数目过多、对 P 初始值敏感两个缺点,利用CURE算法的优点,对AP算法的结果进行优化处理,目的是获得更高的聚类质量。在优化过程中克服CURE算法需要事先输入聚类数目的缺点,该优化算法能够自动确定聚类数目。

本文首先简要介绍近邻传播算法;其次简要介绍CURE

到稿日期:2014-04-24 返修日期:2014-07-18 本文受国家自然科学基金(71271071,71301041),国家"863"云制造主题项目(2011AA040501)资助。

倪志伟(1963—),男,博士,教授,主要研究方向为数据挖掘、机器学习等,E-mail:zhwnelson@163.com;荆婷婷(1990—),女,硕士生,主要研究方向为数据挖掘,E-mail:jingtingting@mail.hfut.edu.cn;倪丽萍(1981—),女,博士,副教授,主要研究方向为分形数据挖掘等,E-mail:niliping@hfut.edu.cn(通信作者)。

算法；再详细描述改进后的层次优化算法—CAP 算法；然后是实验验证；最后对本文工作进行总结并指出进一步的研究方向。

2 传统 AP 算法

假设数据集 D 有 n 个样本$\{x_1, x_2, \cdots, x_n\}$，AP 算法通过相似度来计算吸引度(responsibility)和归属度(availablity)，从吸引度和归属度两方面来衡量该点是否适合作为聚类中心。

2.1 相关定义

相似度矩阵 S：S 可以通过很多方法来度量，它可以是对称的也可以是不对称的，即 $s(i,j)$和 $s(j,i)$可以不相等[14,15]。

一般情况下，相似度可以通过距离来度量，为方便以后计算，一般选择欧氏距离的负值来衡量[18]，如式(1)所示：

$$s(i,j)=-\| x_i-x_j \|^2 \tag{1}$$

其中，当 $i=j$ 时 $s(i,i)$是一个很重要的参数，它反映了对点 i 作为聚类中心的偏好。所以这个值一般会被赋予成一个先验值 $P(i)$，这个先验值也叫做偏好值。

吸引度矩阵 R：吸引度 $r(i,k)$是由点 x_i 发送到候选聚类中心点 x_k 的消息，表示的是点 x_k 适合作为点 x_i 的聚类中心的程度。

归属度矩阵 A：归属度 $a(i,k)$是由候选聚类中心点 x_k 发送到点 x_i 的消息，表示的是点 x_i 选择点 x_k 作为其聚类中心的程度[4]。

$r(i,k)$越大，则表示 x_k 作为聚类中心的可能性越大；$a(i,k)$越大，则表示点 x_i 隶属于点 x_k 的可能性越大[6]。吸引度矩阵 $R=[r(i,k)]$以及归属度矩阵 $A=[a(i,k)]$的更新规则如下。

利用相似度矩阵 S 和归属度矩阵 A，更新吸引度矩阵 R：

$$r(i,k)\leftarrow s(i,k)-\max_{k'\neq k}\{a(i,k')+s(i,k')\},\ i\neq k \tag{2}$$

$$r(k,k)\leftarrow p(k)-\max_{k'\neq k}\{a(k,k')+s(k,k')\},\ i=k \tag{3}$$

利用吸引度矩阵 R 来更新归属度矩阵 A：

$$a(i,k)\leftarrow \min\{0, r(k,k)+\sum_{i'\notin\{i,k\}}\max\{0,r(i',k)\}\} \tag{4}$$

$$a(k,k)\leftarrow \sum_{i'\neq k}\max\{0,r(i',k)\} \tag{5}$$

AP 算法过程中很容易产生震荡，因此在 R 和 A 进行下一次迭代之前都需要根据阻尼因子 lam 对 R 和 A 进行缩放，缩放公式如下：

$$R=(1-lam)*R+lam*Rold \tag{6}$$

$$A=(1-lam)*A+lam*Aold \tag{7}$$

2.2 算法步骤及分析

对于拥有 n 个样本的数据集 D，AP 算法聚类过程如下：

Step1 初始化相似度矩阵 S，设定偏好值 P，初始化 R、隶属度矩阵 A，设置最大迭代次数 maxits、聚类中心稳定次数 convits。

Step2 当迭代次数小于 maxits 或者聚类中心稳定次数小于 convits 时，进行以下 3 步：

Step2.1 Rold=R；Aold=A；

Step2.2 根据式(2)—(5)计算 R 和 A；

Step2.3 对 R 和 A 进行缩放。

否则，进入 Step3。

Step3 当 r(i,i)+a(i,i)>0 时，选取该点作为聚类中心。

Step4 将数据点分配到距离最近的聚类中心，分配类标签。

从上述步骤可以看出 AP 算法的两个缺点：

1)对参数 P 很敏感。偏好值 P 的大小直接影响每个点自身可以作为聚类中心的可能性大小，P 值偏小则聚类数目偏少；反之则偏大。一般来说，$P(i)$取相似度的平均值得到的聚类数目较适中。

2)AP 算法容易产生较多类。假设针对有 n 个样本的数据集 D，采用欧氏距离负值作为相似度度量(见式(1))，则根据式(2)—(5)可以看出，当先验值 P 相同时，若样本点 x_i 处于相邻样本点的中心位置，则 x_i 与其他样本的吸引度较大，成为聚类中心的可能性也会越大，则 $E=\sum|d|$ 最小，其中 d 是每个点到聚类中心的距离。有学者提出 AP 算法聚类的原则是每类的 $\sum E$ 最小，以及整个数据集的 $\sum E$ 最小[21]。因此 AP 算法倾向于产生比真实类数更多的类。

3 CURE 算法

CURE 算法是针对大多数数据集而提出的一种层次聚类算法。采用 CURE 算法将每个点看作是独立的类，这一点与 AP 算法一致。该算法使用基于中心点和代表点的中间策略，一定数目的代表点可以有效地捕捉到任意形状簇；且通过收缩因子，代表点向中心点移动有利于控制孤立点。该算法对异常值处理分为两个阶段：第一阶段，当某个簇增长缓慢时，将该簇作为异常值去除；第二阶段，当临近聚类结束时，某个簇的样本数目明显少，将该簇作为异常值去除[9]。

假设对于数据集 D，CURE 算法将每个点看作一类，先计算类与类之间的距离，寻找到两个距离最近的类，进行类与类的融合，直到达到一定个数的类[16]。每个类均有一个中心点 $Cmean$ 和多个代表点 $Crep$，且每个类代表点最大个数一定。对于数据集 D，设目标聚类数目为 k，代表点最大个数为 n，CURE 算法步骤如下：

Step1 初始化：输入目标聚类数目 k，以及代表点最大个数 n。

Step2 进行迭代融合，当聚类数目>k 时，计算簇间距、新簇中心点、新簇代表点。

Step2.1 计算簇间距离：

dist(u,v)=min(dist(u. rep,v. rep))

u. rep 为簇 u 的代表点，v. rep 为簇 v 的代表点，当簇中只有一个点时，该点为代表点。

Step2.2 将距离最小的两个类合并，计算新簇的中心点 w. mean：

$$\text{w. mean}=\frac{|u|*\text{u. mean}+|v|*\text{v. mean}}{|u|+|v|}$$

|u|为簇 u 的数据点个数。

Step2.3 计算新簇的代表点 w. rep：

```
if |u. rep|+|v. rep|≤n
  for i=1:|u. rep|+|v. rep|
    w. rep=p+α*(w. mean-p)
  p 为簇 u. rep 和 v. rep 的代表点。
  end
else |u. rep|+|v. rep|>n
  w. rep(1)=max(d(w(i),w. mean))
  for i=2:n
  w. rep(i)=max(d(w(i),w. rep(i-1)))
    w(i)为新簇 w 的第 i 个点。
    end
  end
```

否则，进入 Step3。

Step3 异常值处理。

Step4　输出中心点和代表点。

Step5　分配类标签。

4　CAP 算法

由于 AP 算法倾向于通过生成多个聚类簇来达到好的聚类效果，有学者已经研究出根据调整 P 值的大小，来减小聚类数目。但是调整先验值 P 的大小比较困难，因为如果调整幅度过大，则容易跳过某个可能的聚类数目而减小聚类准确率；如果调整幅度过小，反而增加了计算代价[21]。此外有学者提出对 AP 算法的结果排序再进行后处理来提高准确率[19]；还有人提出和其他算法相结合改进相似度矩阵 S 来提高聚类准确率[22]。

4.1　CAP 算法思想

本文利用 CURE 算法能够处理任意形状数据集的优点，在 AP 算法的基础上提出 CAP 算法。假设有 N 个数据的数据集 D 内有样本 $\{x_1, x_2, x_3, \cdots, x_N\}$，CAP 算法分为 3 个阶段：第一阶段，先通过 AP 算法对数据集进行聚类，得到 m 个聚类中心 $\{center_1, center_2, \cdots, center_m\}$；第二阶段，将得到的聚类中心进行去异常值处理，得到 n 个高质量的聚类中心 $\{goodcenter_1, goodcenter_2, \cdots, goodcenter_n\}$；第三阶段，通过 CURE 算法对 $\{goodcenter_1, goodcenter_2, \cdots, goodcenter_n\}$ 进行进一步聚类，得到最好的聚类中心 *Cbest* 和代表点 *Crepbest*，再将样本按照 CURE 算法分配方式分配类标签。

CAP 算法是在 AP 算法之后，用 CURE 算法对其结果进行再次聚类。所以 CAP 算法中，在 AP 阶段应该将先验值 P 设置得稍大，将更多具有代表性的点作为 CURE 算法的输入。这样 CAP 算法就能够捕捉到任意形状的簇，获得更好的聚类效果。

但是增大 P 值，一些噪声点就会被 AP 算法选择出来，因此在进行 CURE 算法之前会进行异常值处理。设去噪声率 α，若 $center_i$ 为中心的簇中样本个数 $n_i < \alpha \times N$，则该 $center_i$ 不会被带入 CURE 算法过程。

为克服 CURE 算法需要预先设定聚类数目的缺点，在 CURE 算法阶段引入 Silhouette 指标对聚类结果进行引导，选择过程中最优的聚类结果输出。

4.2　Silhouette 指标

设一个有 N 个样本的数据集被划分为 k 个类 $\{c_1 \cdots c_k\}$，其中 n_i 表示 c_i 类中的样本个数。类 c_i 中存在一个点 p，则设 $a(p)$ 为点 p 到 c_i 类中其他样本的平均不相似度或者距离；$d(p, c_j)$ 为样本 p 到 c_j 类的所有样本的平均不相似度或者距离，则：

$$b(p) = \min\{d(p, c_j)\}, i = 1 \cdots k, i \neq j \tag{8}$$

于是，样本 p 的 Silhouette 指标为：

$$Sil(p) = [b(p) - a(p)] / \max\{a(p), b(p)\} \tag{9}$$

一个类中所有样本的 *Sil* 平均值反映了该类的紧密性和可分性，而整个数据集的所有样本的 *Sil* 平均值可反映整个聚类结果质量，平均 Silhouette 指标越大表示类内部越紧凑，类间差异越大，聚类质量越好，聚类数目越优。

4.3　CAP 算法步骤

Step1　初始化参数：相似度矩阵 S，先验值 P，去噪声率 α。

Step2　对 D 进行 AP 聚类，得到 m 个聚类中心 $center_1, center_2, \cdots, center_m$。

Step3　进行噪声点处理。

```
for i=1:m
  if 基于该聚类中心的簇中样本数<α×N
    将该聚类中心删除
  else
      将该聚类中心放入 goodcenter 中。
  end
end
```

得到较好的 n 个聚类中心即 $goodcenter_1, goodcenter_2, \cdots, goodcenter_n$。

Step4　对 n 个质量较好的聚类中心进行 CURE 算法聚类。

```
当类个数>2 时
  计算每一次迭代融合的 Silhouette 指标 Sil。
    if Silnew>Silold
    Silmax=Silnew;
    Cbest=Cnew;
    Crepbest=Crepnew;
    Kbest=K;
  else
  Silmax 保持不变，聚类中心 C 不变，每个类代表点 Crepbest 不变。
  end
否则，进入 Step5
```

Step5　得到 Kbest 个聚类中心，对剩余的数据集进行类标签分配。

5　实验结果及分析

5.1　评价指标

聚类算法的评价指标有很多，包括准确率、召回率、F-Measure 及熵等。F-Measure 将准确率和召回率结合在一起来评价聚类结果[12,13]，是一种较好的聚类评价指标。

假设在有 N 个数据的数据集 D 上将实验结果聚为 k 类：$C' = \{c_1', c_2', \cdots, c_k'\}$；正确聚类结果为 t 类：$C = \{c_1, c_2, c_3, \cdots, c_t\}$。假设 $c_i' \in C'$ 且 $c_j \in C$，则准确率为：

$$Precision(c_i', c_j) = \frac{|c_i' \cap c_j|}{|c_i'|} \tag{10}$$

召回率为：

$$Recall(c_i', c_j) = \frac{|c_i' \cap c_j|}{|c_j|} \tag{11}$$

聚类结果的 F-Measure 为：

$$F(C', C) = \frac{1}{N} \sum_{j=1}^{t} |c_j| \times \max_{1 \leqslant i \leqslant k} \left(\frac{2 \times Precision(c_i', c_j) \times Recall(c_i', c_j)}{Precision(c_i', c_j) + Recall(c_i', c_j)} \right) \tag{12}$$

化简之后 F-Measure 即为：

$$F(C', C) = \frac{1}{N} \sum_{j=1}^{t} |c_j| \times \max_{1 \leqslant i \leqslant k} \left(\frac{2 \times |c_i' \cap c_j|}{|c_i'| + |c_j|} \right) \tag{13}$$

5.2　实验环境

本文实验在 Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz 2.00GB 内存 PC 机上的 Matlab R2012a 版本上运行。

5.3　实验结果及分析

本实验主要从以下 3 个方面验证 CAP 算法的优劣性和稳定性。第一，对 UCI 上的 6 个数据集，分别用 K-means 算法[17]、Spectral 算法[4,11]、CURE 算法[8]、AP 算法[10] 和 CAP 算法进行聚类，从 F 测度评价 CAP 在松散数据集上的聚类质

量好坏。第二，由于 AP 算法对 P 值敏感，因此验证 CAP 算法和 AP 算法对不同 P 值的敏感性。第三，由于本算法引进了一个新的参数 α，因此将针对不同 α 来证明 CAP 算法的稳定性。在实验中，为方便验证聚类质量，均采用带标签的数据集。其中 Wine 和 Ionosphere 为小样本松散数据集[5]，Contraceptive 为中等样本数目数据集，page-blocks、waveform 为数据量稍大的数据集。表 1 统计的属性个数均为非标签属性个数。

表 1　实验数据集

数据名称	样本个数	非标签属性个数	聚类数目
Wine	178	13	3
Ionosphere	351	34	2
Contraceptive	1473	9	3
page-blocks	5473	10	5
waveform	5000	31	3

1)不同数据集不同算法之间聚类结果的比较

本文将 CAP 算法与 K-means 算法、Spectral 算法、CURE 算法以及 AP 算法进行聚类结果质量对比。在 K-means、Spectral 以及 CURE 算法中输入的目标聚类数均为数据集中真实聚类数。其中 CURE 算法将代表点个数设为 12，CAP 算法将代表点个数设为 10。在 CAP 算法中 P 值采用的是样本相似度均值 $Smid$，α 取值为 0.05。AP 算法中 P 值也采用的是相似度均值 $Smid$。在数据集 Wine、Ionosphere 以及 Contraceptive 上，由于数据集较小，均不采用抽样。在 page-blocks、waveform 数据集上随机抽取 15%作为训练集，其中对于 Spectral 算法，由于计算空间较大，电脑内存不够，因此没有给出结果。表 2 所列 F-measure 结果均为运行 20 次之后的平均值。

表 2　聚类结果的 F-measure 值比较

	K-means	Spectral	CURE	AP	CAP
Wine	0.6324	0.4634	0.7058	0.6257	0.7145
Ionosphere	0.6919	0.6417	0.6887	0.4551	0.7445
Contraceptive	0.3876	0.3929	0.4394	0.1576	0.4581
page-blocks	0.7517	—	0.8246	0.5514	0.8328
waveform	0.5179	—	0.4920	0.3757	0.6150

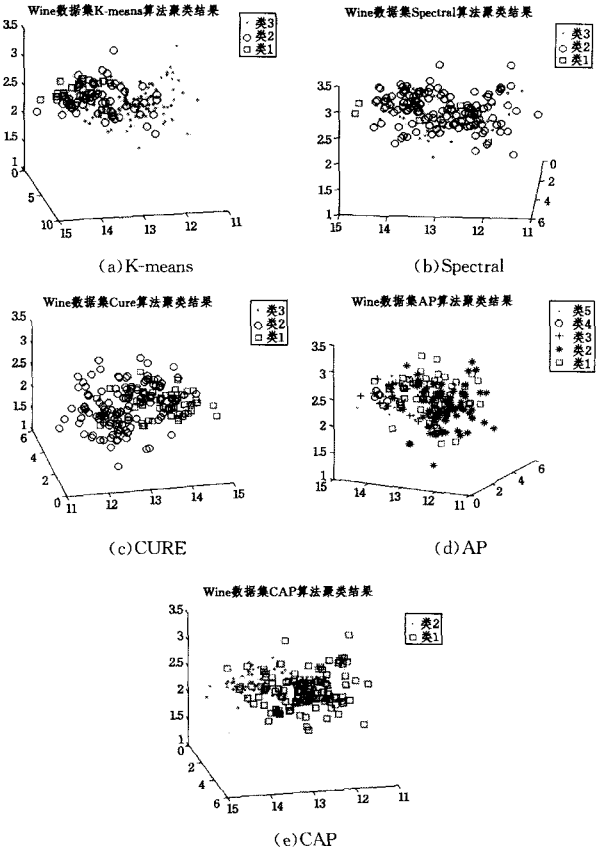
5 种算法在 5 个 UCI 数据集上的 F 测度比较，CAP 算法在松散的数据集上不仅仅比 AP 算法聚类质量高，比其他 4 个聚类算法质量也更好。与 CAP 相比较而言，K-means 在 5 个数据集上均表现出较差的聚类质量。特别是数据集 Contraceptive 上，CAP 算法的 F 测度达到了 0.4581，而 K-means 算法 F 测度只有 0.3876；在数据量稍大的 page-blocks、waveform 上，CAP 算法比 K-means 算法的聚类结果 F 测度高出 0.8 及以上。这是因为 K-means 算法受初始代表点影响较大，CAP 算法较为稳定。与 Spectral 算法比较，CAP 算法在松散的 Wine 数据集上，其 F 测度比 Spectral 算法高出了 0.25 左右；在 Contraceptive 和 Ionosphere 数据集上，CAP 算法 F 测度比 Spectral 算法高出 0.1 左右。与 CURE 算法比较，虽然 CURE 算法在松散的 Wine、Contraceptive 和大样本 page-blocks 数据集上能够得到较好效果，但是 CAP 算法在 5 个数据集上都得出了比 CURE 算法更高的 F 测度，特别是在 Ionosphere 和 waveform 上，证明了 CAP 算法在 AP 算法阶段已经将能代表数据集形状的点筛选出来，在降低 CURE 算法阶段计算代价的同时，又能够准确捕捉到数据集形状。与原始 AP 算法比较而言，CAP 算法在 5 个数据集上均取得可比 AP 算法更高的 F 测度。在 5 个数据集上，AP 算法均没有理想的聚类结果，特别是在数据集 Contraceptive 上，CAP 算法 F 测度为 0.4581，而 AP 算法 F 测度只有 0.1576，前者是后者的近 3 倍。在数据集稍大的 page-blocks、waveform 上，CAP 算法的 F 测度比 AP 算法要高出 0.25 以上。CAP 算法在松散数据集上能够取得比 AP 算法更好的聚类结果。

本文旨在提高聚类质量的基础上，降低聚类个数，使得最终聚类个数更贴近于真实类数。表 3 给出了原始 AP 算法和 CAP 算法的聚类个数与数据集中真实聚类个数的比较。

表 3　聚类类数比较

	真实聚类	CAP 聚类	AP 聚类
Wine	3	2	5
Ionosphere	2	3	36
Contraceptive	3	2	25
page-blocks	5	4	112
waveform	3	3	19

图 1 给出 5 种算法在 Wine 数据集上的聚类结果对比，其中三维图坐标选取数据集的前 3 个属性表达聚类结果。



(a)K-means　(b)Spectral　(c)CURE　(d)AP　(e)CAP

图 1　在 Wine 数据集上的聚类结果

在 Wine 数据集上，通过图 1 更加清楚地看到：K-means 算法的类 3，数据点过于分散，且类 2 与类 1 间距很小；Spectral 算法的聚类质量也很差，因为聚出来的每个类内部结构松散，类与类之间分隔不明显；CURE 算法在 Wine 数据集上有较好的聚类结果，但是其中类 3 与类 1 之间差异化不明显；AP 算法聚出 5 类，但是类 4 与类 5 类间距过小，且类 5 内部样本不够紧凑；CAP 算法在数据集 Wine 上虽然只有两类，但是得到的聚类结构类间差异明显，类内部结构紧凑，是比较好的聚类结果。

2)不同 P 值对 AP 算法和 CAP 算法稳定性的影响

在 AP 算法中 P 值很大程度上决定了聚类类数，因此通过 AP 算法和 CAP 算法中 P 值的调节，来验证不同 P 值对 AP 算法和 CAP 算法的影响程度，从一方面验证 CAP 算法的稳定性。在验证 P 值对聚类质量的影响的实验中，为了排除 α 的影响，α 统一取值为 0.05。

在 AP 算法值中有 3 个 P 值值得注意，相似度最大值 S_{max}、相似度平均值 S_{mid}、相似度最小值 S_{min}。一般情况下，P 值取所有相似度的平均值得到的聚类数目和聚类质量比较好。在有重复样本时，P 取最大值将导致算法崩溃，不能给出正确的聚类结果。若 P 值取 S_{min}，则容易导致聚类中心太少。

CAP 算法应该在第一阶段产生相对较多的聚类中心，将较多具有代表性的聚类中心作为第二阶段的输入。因此本文将 P 值在合理范围内取得偏大会使得效果较好，从而只考虑 P 取 $[S_{min}, S_{max})$ 之间的值。在下面的实验中通过式(14)调节 β，使得 $P\in[S_{mid}, S_{max})$：

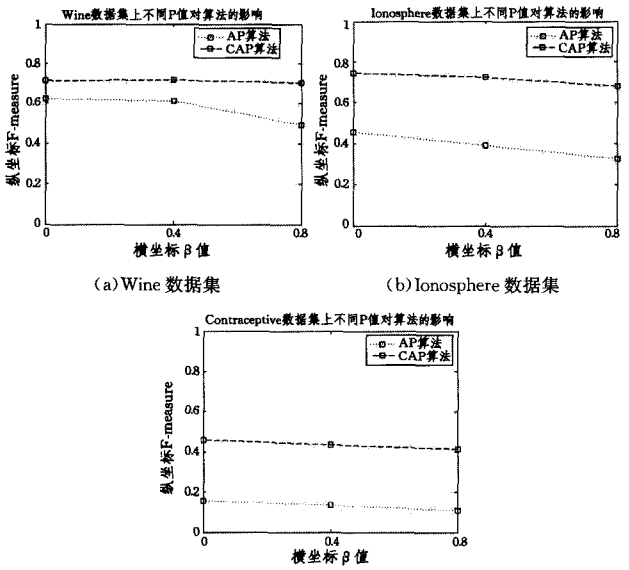
$$P=S_{mid}+\beta\times(S_{max}-S_{mid}),\beta\in[0,1) \tag{14}$$

不同 P 值的聚类结果如表 4 所列。

表 4　不同 P 值的聚类结果

	AP			CAP		
	β=0	β=0.4	β=0.8	β=0	β=0.4	β=0.8
Wine	0.6257	0.6122	0.4922	0.7145	0.7176	0.7204
Ionosphere	0.4551	0.3909	0.3257	0.7445	0.7226	0.6804
Contraceptive	0.1576	0.1353	0.1108	0.4581	0.4329	0.4123

为了更加直观地展示 P 值对 AP 算法和 CAP 的影响，下面给出 AP 算法和 CAP 算法对于 β 分别在 0、0.4、0.8 上的折线图，如图 2 所示。



(a)Wine 数据集　(b)Ionosphere 数据集

(c)Contraceptive 数据集

图 2　不同 P 对 CAP 算法和 AP 算法的影响

从上面实验可以看出，针对同一数据集，不同 P 值对 AP 算法和 CAP 算法的聚类结果均产生影响，但是 CAP 受 P 值影响明显比 AP 算法要小。在不同数据集上，因为受数据集形状和样本分布情况影响，每个数据集对应的最佳 P 值也不一样。但是从上面 3 个数据集实验情况看来，CAP 算法在不同数据集上对于 P 值的敏感性比 AP 算法要低。

3)不同 α 值对 CAP 算法的影响

因为本文在提出算法的过程中带入了一个新的变量去噪声率 α。当 P 越大时，选出的聚类越多，每个类就很小，总体来说单个簇的样本个数较少，此时 α 应当较小。在 CAP 算法中 α 取值在 0.02～0.06 之间均能有比 AP 算法更好的效果。因此通过 α 不同值的调节来验证 CAP 算法对 α 的敏感性。所以该实验阶段将 P 值统一设为相似度均值 S_{mid}。实验结果如表 5 和图 3 所示。

表 5　不同 α 的 CAP 算法的聚类结果

	AP	CAP				
		α=0.02	α=0.03	α=0.04	α=0.05	α=0.06
Wine	0.6257	0.7156	0.7156	0.7145	0.7145	0.7145
Ionosphere	0.4551	0.6570	0.6578	0.6638	0.7445	0.7445
Contraceptive	0.1576	0.4512	0.4501	0.4501	0.4499	0.4362

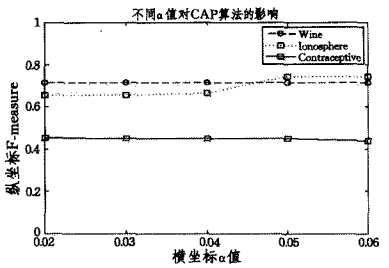


图 3　不同 α 值对 CAP 算法的影响

由图 3 可以看出，针对不同数据集，最优 α 值也不一样。在 Wine 数据集上，对于不同 α 值，CAP 算法 F 测度基本不变。在 Ionosphere 数据集上，当 $\alpha\leqslant$0.04 时，F 测度相对于 α=0.05 和 α=0.06 时普遍较小；而在 Contraceptive 数据集上，随着 α 值增大，CAP 算法质量在不断下降，因为每个数据集形状不一样，松散程度不同，导致去除异常点时应该设置的大小也不一样，但是当 α 值在 0.02 到 0.06 之间时均能够取得比 AP 算法更好的效果，且 F 值波动在合理范围之内。

结束语　本文主要针对 AP 算法的只适合于球形数据和对聚类数目过多两个缺点，提出了 CAP 算法。CAP 算法在对 AP 算法聚类准确率大幅提升的基础上，使得聚类数目更贴近于真实聚类数目；同时 CAP 算法很大程度上克服了 AP 算法对 P 值的敏感度。由于 CAP 算法是在 AP 算法结束之后再对其聚类中心进行一次 CURE 聚类，因此在运行时间上，CAP 算法比 AP 算法花费时间略长。因此下一步工作主要集中在通过 MapReduce 方式来缩短 CAP 算法运行时间。

参 考 文 献

[1] Jain A K. Data clustering:50 years beyond K-means[J]. Pattern Recognition Letters,2010,31(8):651-666

[2] Shang Fan-hua,Jiao L C,Shi Jia-rong,et al. Fast affinity propagation clustering: A multilevel approach[J]. Pattern Recognition,2012,45(2012):474-486

[3] Vlasblom J, Ahoshana J W. Markov clustering versus affinity propagation for thpartitioning of protein interaction graphs[J]. BMC Bioinformatics,2009(10):99-113

[4] Paccanaro A, Casbon J A, Saqi M A S. Spectral clustering of protein sequences[J]. Nucleic Acids Research, 2006, 34(5): 1571-1580

[5] Wang Kai-jun, Zhang Jun-ying, Li Dan, et al. Adaptive affinity propagation clustering[J]. Acta Automatica Sinica, 2007, 33(12):1242-1246

[6] Karen K. Affinity program slashes computing times[OL]. 2007-10-25. http://www.news.utoronto.ca/bin6/070215-2952.asp

[7] Bodenhofer U, Kothmeier A, Hochreiter S. APCluster: an R package for affinity propagation clustering[J]. Bioinformatics, 2011,27(17):2463-2464

[8] Guha S,Rastogi R,Shim K. CURE:an efficient clustering algorithm for large databases[J]. ACM SIGMOD Record, ACM, 1998,27(2):73-84

[9] Ertoz L, Steinbach M, Kumar V. A new shared nearest neighbor clustering algorithm and its applications[C]// Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining. 2002:105-115

[10] J Fray B J,Dueck D. Clustering by Passing Messages Between Data Points[J]. Science,2007,315(5814):972-976

[11] Hsu D, Kakade S M, Zhang Tong. A spectral algorithm for learning hidden Markov models[J]. Journal of Computer and System Sciences,2012,78(5):1460-1480

[12] Powers D M W. Evaluation:from precision,recall and F-measure to ROC,informedness,markedness & correlation[J]. Journal of Machine Learning Technologies,2011,2(1):37-63

[13] Rawashdeh M, Ralescu A. Crisp and fuzzy cluster validity: generalized intra-inter silhouette index[C]// 2012Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS). IEEE,2012:1-6

[14] Guan R,Shi X,Marchese M,et al. Text clustering with seeds affinity propagation[J]. IEEE Transactions on Knowledge and Data Engineering,2011,23(4):627-637

[15] Kazantseva A,Szpakowicz S. Linear text segmentation using affinity propagation[C]// Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics,2011:284-293

[16] Patidar A K,Agrawal J,Mishra N. Analysis of Different Similarity Measure Functions and their Impacts on Shared Nearest Neighbor Clustering Approach[J]. International Journal of Computer Applications,2012,40(16):1-5

[17] Lee S S,Lin J C. An accelerated K-means clustering algorithm using selection and erasure rules[J]. Journal of Zhejiang University SCIENCE C,2012,13(10):761-768

[18] 鲁伟明,杜晨阳,魏宝刚,等. 基于 MapReduce 的分布式近邻传播聚类算法[J]. 计算机研究与发展,2012,49(8):1762-1772

[19] 唐东明,朱清新,杨凡,等. 一种有效的蛋白质序列聚类分析方法[J]. 软件学报,2011,22(8):1872-1837

[20] 沈洁,赵雷,杨季文,等. 一种基于划分的层次聚类算法[J]. 计算机工程与应用,2007,43(31):175-177

[21] 王开军,李健,张军英,等. 半监督的仿射传播聚类[J]. 计算机工程,2007,33(23):197-198

[22] 邢艳,周勇. 基于互近邻一致性的近邻传播算法[J]. 计算机应用研究,2012,29(7):2524-2526

(上接第 194 页)

于时间窗的疲劳状态判别方法,能够有效地从视频流图像中识别出疲劳状态。同时,为提高算法的个体适应性,在算法的识别阶段设计了基于时间窗和疲劳相关度的分层权重调整作为算法的反馈机制,让网络具有自进化和自增长功能。本文算法较好地解决了疲劳特征自动提取的问题,进一步提高了算法的个体适应性。实验结果表明,该算法对疲劳状态具有良好的识别率,且反馈机制能很好地使网络学习新的疲劳状态表示并调整已获得的疲劳状态表示,随着时间的推移,网络识别效果会越来越好。在实验过程中,我们发现当输入的视频流图像为大尺寸、高维度时,算法处理速度会明显变慢,而在线疲劳识别及预警对实时性要求很高,因此,如何利用 GPU 实现并行处理,以进一步提高算法效率,是我们今后研究的重点。

参 考 文 献

[1] Correa A G, Orosco L, Laciar E. Automatic detection of drowsiness in EEG records based on multimodal analysis[J]. Medical Engineering & Physics,2014,36(2):244-249

[2] Li G,Chung W Y. Detection of Driver Drowsiness Using Wavelet Analysis of Heart Rate Variability and a Support Vector Machine Classifier[J]. Sensors,2013,13(12):16494-16511

[3] 李伟,何其昌,范秀敏. 基于汽车操纵信号的驾驶人疲劳状态监测[J]. 上海交通大学学报,2010,44(2):292-295

[4] 张希波,成波,冯睿嘉. 基于转向盘操作的驾驶人疲劳状态实时检测方法[J]. 清华大学学报:自然科学版,2010,50(7):1072-1076

[5] 李绍文,王江波. 驾驶员疲劳检测系统研究[J]. 计算机工程与应用,2013,49(15):253-258

[6] Hinton G,Salakhutdinov R. Reducing the Dimensionality of Data with Neural Networks[J]. Science,2006,313(5786):504-507

[7] 余凯,贾磊,陈雨强,等. 深度学习的昨天、今天和明天[J]. 计算机研究与发展,2013,50(9):1799-1804

[8] Hinton G E,Osindero S. A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation,2006,18:1527-1554

[9] 刘建伟,刘媛,罗雄麟. 玻尔兹曼机研究进展[J]. 计算机研究与发展,2014,51(1):1-16

[10] Hinton G. Trainging Products of Experts by Minimizing Contrastive Divergence[J]. Neural Computation,2002,14(8):1771-1800

[11] Neal R M,Hinton G E. A View of the EM Algorithm that Justifies Incremental,Sparse and other Variants[M]// Learning in Graphical Models,1998,355-368

[12] 蒋斌,贾克斌,杨国胜. 人脸表情识别的研究进展[J]. 计算机科学,2011,38(4):25-31

[13] Dasgupta A,George A,Happy SL,et al. A Vision-Based System for Monitoring the Loss of Attention in Automotive Drivers[J]. IEEE Transactions on Intelligent Transportation Systems,2013,14(4):1825-1838

[14] 张伟,夏利民,罗大庸. 基于人脸运动信息和改进保局投影的疲劳识别[J]. 计算机科学,2010,37(11):265-267

[15] Cyganek B,Gruszczynski S. Hybrid computer vision system for drivers' eye recognition and fatigue monitoring[J]. Neural computation,2014,126(SI):78-94

[16] 孙树亮,林雪云. 基于记忆的 SVM 相关反馈算法[J]. 计算机科学,2011,38(10):256-258

[17] 陈云华,余永权,张灵,等. 基于面部视觉特征的精神疲劳可拓辨识模型[J]. 计算机科学,2013,40(2):284-288