一种分层组合的半监督近邻传播聚类算法

张 震* 汪斌强 伊 鹏 兰巨龙
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：针对近邻传播(AP)聚类算法的计算复杂度和准确性，该文提出一种分层组合的半监督近邻传播聚类算法(SAP-SC)。算法引入“分层聚类”的思想，将一次AP聚类过程等分成若干层聚类，使得处理过程简单、易于实现；每层只关注聚类“困难”的数据点，并通过构造“成对点约束”和使用“子簇标签映射”进行半监督学习；基于“组合提升”的方法将各层聚类结果加权叠加，从而提升了算法的准确性能。理论分析和实验结果表明：算法在聚类准确性和计算复杂度方面有了较大改进。

关键词：半监督学习；近邻传播聚类；分层聚类；组合提升
中图分类号：TP181 **文献标识码**：A **文章编号**：1009-5896(2013)03-0645-07
DOI: 10.3724/SP.J.1146.2012.00673

Semi-supervised Affinity Propagation Clustering Algorithm Based on Stratified Combination

Zhang Zhen Wang Bin-qiang Yi Peng Lan Ju-long
(*National Digital Switching System Engineering and Technological Research Center*, *Zhengzhou* 450002, *China*)

Abstract: Considering the complexity and the accuracy, an improved affinity propagation clustering algorithm Semi-supervised Affinity Propagation clustering algorithm based on Stratified Combination (SAP-SC) is proposed. In order to make the operation simplified and easily-implemented, the proposed algorithm introduces a stratified clustering method which equally partitions the integrative clustering process into several smaller blocks. Focusing on the hard clustering data, every layer employs semi-supervised learning to conceive pair-wise constraints and maps each sub-cluster with the corresponding label. Also, assembled boosting method is utilized to weight together all layered results to improve the clustering performance. Finally, theoretical analysis and experimental results show that the algorithm can achieve both higher accuracy and better computational performance.

Key words: Semi-supervised learning; Affinity Propagation (AP) clustering; Stratified clustering; Assembled boosting

1 引言

随着信息技术的迅猛发展，数据聚类已广泛运用于模式识别、Web挖掘、生物医学以及多目标检测等领域，成为数据分析的必要手段[1−4]。尤其是近年来，Internet的飞速发展，使得互联网共享数据规模呈现几何速度增长，如何从海量数据中挖掘出有指导意义的信息变得更加迫切。借助于聚类技术，人们在对数据集几乎“一无所知”的情况下就能发现数据之间的内在联系和结构信息，使聚类技术显得尤为重要。目前，聚类技术在不同的应用领域，衍生出了许多新的优异算法，其中代表性的算法包括：基于划分的k-means、基于密度的DBSCAN、基于图的谱聚类等。

Frey等人[5]于2007年在《Science》上首次提出近邻传播(Affinity Propagation, AP)聚类算法，算法将每个样本点都视为网络中的一个节点，吸引力信息沿着节点连线递归传输，直到找到最优的类代表点集合。相比k-means, DBSCAN以及谱聚类，AP算法将所有的数据点作为候选类代表点，避免了聚类结果受限于初始类代表点的选择；聚类过程与数据的维数无关，对数据点的相似度矩阵没有对称性的要求。但是，AP算法是基于中心的聚类方法，在紧凑的超球形分布的数据集上具有较好的性能，并不适合复杂形状的聚类问题，聚类性能有待进一步提高；随着样本数的急剧增长，构建相似度矩阵需要消耗大量时间，计算复杂度需进一步降低。针对AP算法的缺陷，文献[6]提出了一种半监督的近邻传播聚类方法(Semi-supervised clustering based

2012-05-31收到，2012-12-31改回
国家973重点基础研究发展基金(2012CB312901, 2012CB312905)和国家863计划项目(2011AA01A103)资助课题
*通信作者：张震 zhangzhen2096@163.com

on Affinity Propagation algorithm, SAP)。SAP 引入数据点的成对点约束来调整相似度矩阵，改善了 AP 算法的聚类性能。但是，获得的用户先验信息通常是非常有限的，且当先验信息包含噪声时，反而可能误导聚类过程。文献[7]从数据流形的角度设计了一种可变相似性度量的近邻传播聚类(Affinity Propagation clustering based on Variable-Similarity Measure, AP-VSM), AP-VSM 通过缩短处于同一流形的数据间的距离、扩大不同流形区域的数据间的距离，来改进相似度矩阵，提高了聚类的精度。但是，该方法需要计算任意两个数据点之间的相似性度量，增加了算法的计算复杂度。

为了提高聚类的准确性和降低复杂度，本文提出一种分层组合的半监督近邻传播聚类方法(Semi-supervised Affinity Propagation clustering algorithm based on Stratified Combination, SAP-SC)。区别于 SAP 和 AP-VSM 方法，SAP-SC 具有以下特点：利用“半监督”机制指导聚类；采用“分层聚类”的思想，降低算法的计算复杂度；使用“Adaboost 组合提升”[8]方法提高聚类的精度。

2 近邻传播学习原理

设 $X=\{x_1,\cdots,x_N\}$ 为模式空间 R^d 的一个有限数据集，其中 $x_i(i=1,2,\cdots,N)$ 是由 d 维属性构成的向量空间中的数据点。所谓聚类，就是依据某种准则将数据集 X 划分成 m 个互不相交的非空子集 C_1, $C_2,\cdots,C_m$ 。

AP 算法的目标是找到最优的类代表点集合，使得所有数据点到最近的类代表点的相似度之和最大。首先将 N 个数据点都视为候选类代表点(即潜在的聚类中心)，并为每个点建立与其他数据点的吸引程度 $s(i,j)=-||x_i-x_j||^2(i\neq j)$ ，形成一个 $n\times n$ 的相似度矩阵 $\boldsymbol{S}_{n\times n}$ 。$s(i,j)=0$ 代表 x_i 和 x_j 有最大相似性，相反，$s(i,j)=-\infty$ 代表 x_i 和 x_j 属于不同类别；$s(i,i)$ 代表数据点 x_i 被选作类代表点的倾向性，$s(i,i)$ 的值越大，对应数据点 x_i 被选中作为类代表点的可能性越大。AP 算法初始设定所有 $s(i,i)$ 为相同值，并通过改变 $s(i,i)$ 值来寻找合适的聚类子簇数目。

为了进行数据点的相似度信息传播，AP 算法引入了两个重要的信息量参数，分别定义为“吸引度 $r(i,j)$ ”和“归属度 $a(i,j)$ ”。如图 1 所示， $r(i,j)$ 是从 x_i 指向 x_j ，表示 x_j 适合作为 x_i 的类代表点的程度； $a(i,j)$ 是从 x_j 指向 x_i ，表示 x_i 选择 x_j 作为其类代表点的合适程度。$r(i,j)$ 和 $a(i,j)$ 越大， x_j 作为 x_i 的类代表点的可能性越大。AP 算法信息量 $r(i,j)$ 和 $a(i,j)$ 的交替更新过程如下：

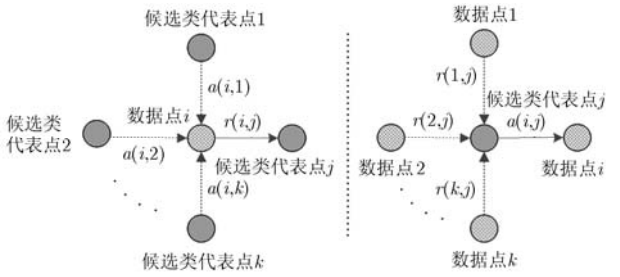

图 1 信息量传播示意图

$$r^{(t)}(i,j)\leftarrow\lambda\cdot r^{(t-1)}(i,j)+(1-\lambda)\cdot\left(s(i,j)-\max_{k\neq j}\left\{a^{(t-1)}(i,k)+s(i,k)\right\}\right) \tag{1}$$

$$a^{(t)}(i,j)\leftarrow\begin{cases}\lambda\cdot a^{(t-1)}(i,j)+(1-\lambda)\cdot\min\left\{0,\ r^{(t-1)}(j,j)+\sum\limits_{i'\neq i,i'\neq k}\max\left(0,r^{(t-1)}(i',j)\right)\right\}, & i\neq j\\ \lambda\cdot a^{(t-1)}(i,j)+(1-\lambda)\cdot\sum\limits_{i'\neq j}\max\left(0,r^{(t-1)}(i',j)\right), & i=j\end{cases} \tag{2}$$

其中 $0\leq\lambda<1$ 为阻尼因子，在每一次循环迭代中，$r(i,j)$ 和 $a(i,j)$ 的更新结果都是由当前迭代过程中更新的值和上一步迭代的结果加权得到的，目的是改进算法的收敛性，避免迭代过程中出现数值震荡。迭代终止的条件满足以下其中之一即可：超过某一迭代最大数目；信息改变量低于某一固定阈值；选择的类代表点在迭代过程中保持稳定。迭代完成后，对与任意 x_i ，计算满足条件 $\arg\max\limits_j\left(a^{(t)}(i,j)+r^{(t)}(i,j)\right)$ 的 x_j ，并将其作为 x_i 的类代表点。

从 AP 算法的学习过程可以看出：AP 算法不受初始点选择的困扰，是一种连续优化的过程，各个数据点进行竞争而得到最终的聚类中心；只需要进行简单的局部计算，能够在更短的 CPU 运算时间里达到较好的聚类效果。目前该算法已经成功应用于最优航线设计、基因发现以及图像分割等领域[5,9,10]。但是，AP 算法也具有以下缺点：AP 是一种无监督的学习方法，没有考虑应用的先验信息和背景知识，分类性能有待提高；比较适合处理超球形结构的数据聚类问题，当数据集的结构比较松散时，却不能给出理想的聚类结果；构建相似度矩阵需要消耗大量时间，计算复杂度需进一步降低。

3 分层组合的聚类思想

3.1 半监督分层聚类

SAP-SC 算法采用“分层聚类”的方法，将 N

个数据点的一次 AP 聚类，等分成 N/M 次半监督 AP 聚类；每次聚类按照各数据点的权重抽取固定的 M 个样本。而“半监督”的思想则体现在：利用少量已知类别的数据点信息、构造成对点约束以及进行子簇标签映射。

(1)分层抽样处理　如图 2 所示，分层抽样就是每次聚类时都要根据各数据点的权重，从数据集 $X=\{x_1,x_2,\cdots,x_N\}$中进行不等概率的有放回抽样，抽取固定数量的 M 个样本；并且第 i+1 次聚类数据的抽样权重是由第 i 次聚类结束时更新得到的。抽样权重的更新原则是：针对抽中的数据点，若本次聚类结果与上次不同，则增大其下一次的抽样权重；相反，若两次聚类结果相同，则减小其抽样权重。不妨设每个数据点的抽样权重为 $w_i\,(i=1,2,\cdots,N)$，对应的抽样概率 $p_i=w_i/\sum w_i$。初始时，设定各个数据点的权重相等，抽样概率为 p_i=1/N。

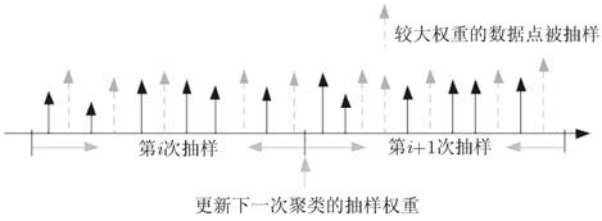

图2 分层抽样处理示意图

采用这种方法，主要具有以下优势：将大规模数据集进行分层抽样聚类，操作简单、易于实现；每层处理的数据点都是“困难”聚类数据，具有很好的代表性；聚类过程在每一层进行，再由各层进行加权组合投票，从而决定每个数据点所属子簇，能够提高聚类精度。

(2)构造成对点约束　设数据点的类别标签集合 $L=\{L_1,\cdots,L_m\}$，令 X^l，X^u 分别表示大小为 N_l，N_u 的已标记数据集合和未标记数据集合，其中 $N_l+N_u=N$，$X=X^l\cup X^u$。类似 SAP 算法，针对已标记数据集合 X^l，SAP-SC 将成对点约束分为两种[6]：若 $\forall x_i^l,x_j^l\in X^l$ 属于同一类别，表示为 $(x_i^l,x_j^l)\in$ Mustlink；若 x_i^l 和 x_j^l 不属于同一类别，表示为 $(x_i^l,x_j^l)\in$ Cannotlink。很显然，如果 $(x_i^l,x_j^l)\in$ Mustlink，则两个数据点的相似性应该最小；反之，若 $(x_i^l,x_j^l)\in$ Cannotlink 成立，则两个数据点的相似性最大。基于此，对 AP 聚类算法的相似度矩阵做如下调整：

(a)Must-link 约束：$(x_i^l,x_j^l)\in \text{Mustlink}\Rightarrow s(i,j)=0,\ s(j,i)=0$；

(b)Cannot-link 约束：$(x_i^l,x_j^l)\in \text{Cannotlink}\Rightarrow s(i,j)=-\infty, s(j,i)=-\infty$；

(c)Must-link 约束的传递：
$$\left.\begin{matrix}\left(x_i^l,x_j^l\right)\in \text{Mustlink}\\ \left(x_i^l,x_k^l\right)\in \text{Mustlink}\end{matrix}\right\}\Rightarrow\left(x_j^l,x_k^l\right)\in \text{Mustlink}\Rightarrow\begin{cases}s(j,k)=0\\ s(k,j)=0\end{cases};$$

(d)Cannot-link 约束的传递：
$$\left.\begin{matrix}\left(x_i^l,x_j^l\right)\in \text{Cannotlink}\\ \left(x_i^l,x_k^l\right)\in \text{Mustlink}\end{matrix}\right\}\Rightarrow\left(x_j^l,x_k^l\right)\in \text{Cannotlink}\Rightarrow\begin{cases}s(j,k)=-\infty\\ s(k,j)=-\infty\end{cases}。$$

根据以上原则，对每层抽样获取的 M 个样本的相似度矩阵 $\boldsymbol{S}_{M\times M}$进行调整，并经过 AP 算法进行信息量迭代更新，使得同类数据点之间的吸引力最大化而被划归为同一类簇，不同类别数据点之间的吸引力最小化而被强制拆开，从而提高聚类算法的性能。

(3)子簇标签映射　子簇标签映射就是根据子簇中各数据点的聚类标签，来确定该子簇的所属类别，其主要分为两个步骤：

步骤 1　初始化 X^u 的类别标签。

应用少量的已标记数据集合 X^l，采用 K 最近邻(K Nearest Neighbors, KNN)方法，对大量的未标记数据点进行初始化标记。KNN 依据最近邻的 K 个样本点计算样本点的类别标签：对与任意未标记的数据点 x_i^u，找到离其最近的 K 个已标记数据点；如果此 K 个数据点中的大多数属于某一类别 L_j，则 x_i^u 也属于此类别 L_j。为了降低计算量，本文采用 1NN 的方法(令 K=1)，即将距离 x_i^u 最近的已标记数据点 x_j^l 的类别赋予 x_i^u。

步骤 2　确定子簇标签映射函数。

针对每层聚类的 M 个样本，SAP-SC 使用 SAP 算法得到 m' 个子簇 $C_1,C_2,\cdots,C_{m'}$，其中 $m'\le m$。子簇标签映射就是确定子簇 $C_1,C_2,\cdots,C_{m'}$ 和类别集合 $L=\{L_1,\cdots,L_m\}$ 的映射关系。SAP-SC 算法采用条件概率 $P(l=L_j|C_i)$ 来表示在子簇 C_i 中任意样本点属于类别标签 L_j 的概率，并且利用训练样本集对其进行估计，其中 $i\in[1,m'],\ j\in[1,m]$。

令 N_i^j 表示在子簇 C_i 中属于类别 L_j 的样本流总数，N_i 表示子簇 C_i 中所有的样本流总数，则根据最大似然估计，可得条件概率的估计表达式为：$\widehat{P}(l=L_j|C_i)=N_i^j/N_i$。基于此，可得到子簇的标签映射函数为

$$l(C_i)=\arg\max_{j=1,\cdots,m}\widehat{P}\left(l=L_j\middle|C_i\right)\qquad(3)$$

3.2 组合提升方法

机器学习中 AdaBoost 算法的目标是提高给定分类算法的准确率，算法的基本思想是[8]：自适应迭代训练样本的权重，使得基分类器聚焦在那些“困难”的数据样本上；利用分类能力一般的基分类器，通过一定的方法进行组合叠加，并最终生成一个强分类器。理论证明[8]：只要每个基分类器的分类能力比随机猜测好，当基分类器的个数趋向于无穷时，强分类器的错误率将趋于 0。

受 AdaBoost 分类算法的启发，SAP-SC 在半监督分层聚类的基础上，通过对各层聚类结果的加权组合来提升聚类结果的准确性。如图 3 所示，SAP-SC 算法主要包括 3 个步骤：数据分层抽样处理；对 $T=\lceil N/M \rceil$ 个子集依次进行半监督 AP 聚类；最后，进行组合投票确定各个数据点的聚类标签。

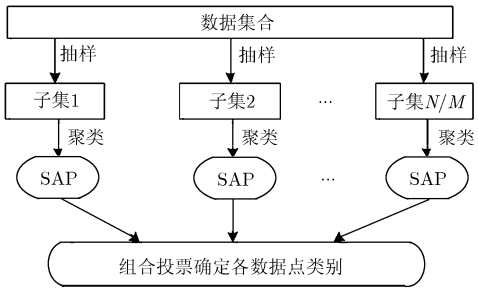

图3 SAP-SC算法逻辑流程图

4 分层组合的半监督 AP 算法

4.1 聚类算法详细流程

表1给出了SAP-SC算法的详细流程描述，SAP-SC 算法的核心思想体现在：利用 1NN 方法对未知标签集合进行初始化标记；采用“分而治之”的思想，将 N 个数据点的一次 AP 聚类，等分成 N/M 次半监督 AP 聚类；分层半监督 AP 聚类中，构造成对点约束改进相似度矩阵，并对各个子簇进行标签映射；在 AdaBoost 的基础上，使用“组合提升”方法提高聚类的精确度。

表中 $\delta(l^t(x_i)\neq l^{t-1}(x_i))$ 表示：若 $l^t(x_i)\neq l^{t-1}(x_i)$ 成立，则其值等于 1，否则等于 0。可以看出，每个被抽中数据点的权重会不断得到调整：若上次聚类结果和本次聚类结果不同，认为其是“困难”样本，则增大其下一次的抽样权重；反之，则减小其下一次的抽样权重。最后，通过加权组合得到数据点的聚类标签。

4.2 算法准确性分析

根据文献[8]的定理 6, SAP-SC 算法的聚类错误率满足不等式：$\varepsilon \le 2^T \prod_{t=1}^{T} \sqrt{\varepsilon_t(1-\varepsilon_t)}$。其中 ε_t 为

表 1 SAP-SC 算法详细流程

SAP-SC 算法流程描述

(1) 输入：数据数据流 $X = X^l \cup X^u$，初始化参数 $s(i,i)$，$r(i,j)$，$a(i,j)$，λ，$T=\lceil N/M \rceil$，$w_i^1 = 1/N$

(2) 应用少量的已标记数据集合 X^l，使用 1NN 对 X^u 进行初始化标记；

(3) **for** t=1 to T **do**

(4) 根据 w_i^t，计算抽样概率 p_i^t，并抽样得到 M 个样本点集合 X_M；

(5) $\forall x_i, x_j \in X_M$，计算欧式距离 $d(i,j)=\sqrt{||x_i - x_j||^2}$，构造相似度矩阵 $\boldsymbol{S}_{M\times M}$；

(6) 由已标记数据点的成对点约束，调整 $\boldsymbol{S}_{M\times M}$；

(7) 根据式(1)，式(2)，应用 AP 进行聚类，得到 m' 个子簇；

(8) 根据式(3)进行子簇标签映射，确定每个被抽样本点的类别标签 $l^t(x_i)$；

(9) 计算聚类误差 $\varepsilon_t = \left[\sum_{i=1}^{M} p_i^t \delta\left(l^t(x_i) \neq l^{t-1}(x_i)\right)\right]$；

(10) **if** $\varepsilon_t > 0.5$ **then**

(11) 重置 $w_i = 1/N (i = 1,2,\cdots,N)$，返回步骤(4)；

(12) **end if**

(13) 设 $\alpha_t = \varepsilon_t/(1-\varepsilon_t)$，更新每个样本的权值 $w_i^{t+1} = w_i^t \alpha_t^{1-\delta(l^t(x_i)\neq l^{t-1}(x_i))}$；

(14) **end for**

(15) 聚类标签输出：$l(x_i) = \arg\max_{y\in L} \sum_{t=1}^{T} \left(\lg \frac{1}{\alpha_t}\right) \delta(l^t(x_i)=y)$。

第 t 步的聚类误差。如果每一步的聚类误差小于随机猜测的概率 0.5，则可得 $\varepsilon_t = 0.5 - h_t$，其中 h_t 度量了比随机猜测更准确的程度。基于此，可进一步得到聚类算法的错误概率满足：

$$\varepsilon \le \prod_{t=1}^{T} \sqrt{1-4h_t^2} \le \exp\left(-2\sum_t h_t^2\right) \tag{4}$$

如果存在某一猜测变量 h，满足条件 $\forall h_t < h$，则 $\varepsilon \le \exp(-2T\cdot h^2)$。如图 4、图 5 所示，每层的聚类准确性比随机猜测越好，聚类误差降低越快；聚类误差随迭代步骤的增大呈指数型递减，算法收敛也就越快。

4.3 算法复杂度分析

由 AP 聚类学习过程可知：AP 聚类的计算复杂度主要用于构建相似度矩阵和进行信息量迭代上。构建相似度矩阵的计算复杂度为 $O(N^2)$，信息量迭代所消耗的时间取决于算法的迭代次数。所以整个算法的时间复杂度不高于 AP 聚类最大迭代次数所消耗的时间，最低不小于 $O(N^2)$。一般来说，算法通常不会达到最大迭代次数，除非算法不收敛。因

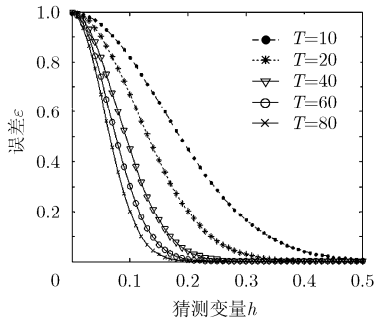

图 4 聚类误差与每层精度的关系曲线

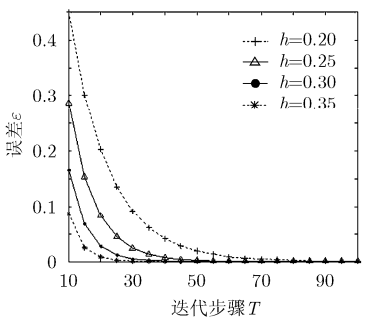

图 5 聚类误差与迭代步骤的关系曲线

此，不妨设定 AP 算法的计算复杂度为 $O(\rho N^2)$，其中 $1<\rho\ll N$。

SAP 和 AP-VSM 都没有改变 AP 算法的数据处理框架，只是在 AP 的基础上增加了额外的操作。所以，SAP 和 AP-VSM 的计算复杂度都高于 AP。而 SAP-SC 将数据进行了分层处理，将一次大规模数据的聚类转换为多次小规模的分层迭代聚类。由于要进行 N/M 步的迭代聚类，每一步聚类数据样本的个数等于 M，所以 SAP-SC 的计算复杂度为 $O(\rho M^2\cdot N/M)=O(\rho MN)<O(\rho N^2)$，其中 $M\ll N$。如果设定 $M=\sqrt{N}$，则 SAP-SC 的计算复杂度等于 $O(\rho N^{3/2})$。

5　实验结果及分析

5.1 算法评价指标

实验中采用以下两种评价指标：

定义 1 整体准确率　对于任意聚簇 $C_i\in C=\{C_1,\cdots,C_m\}$，假设被正确聚类为 C_i 的数据样本总数为 $N_i^{'}$，聚类为 C_i 的数据样本总数为 N_i，则检测准确率为：$P_i=N_i^{'}/N_i$，算法的整体准确率：$P_{\text{all}}=\sum_{i=1}^{m}N_i^{'}\Big/\sum_{i=1}^{m}N_i$。

定义 2 计算复杂度　针对某一数据集，聚类算法收敛所需要的时间。

5.2 实验数据说明

实验采用文献[11]中 18 种不同类的 UCI 数据集，其中数据集的样本总数 N=50612。在每个算法的仿真实验中，初始化信息量 $r(i,j)=a(i,j)=0$，阻尼系数 $\lambda=0.9$，样本初始权重 $w_i^1=1/N$，每层抽样样本数 $M=\left\lfloor\sqrt{N}\right\rfloor$。设定偏向参数 $s(i,i)$ 等于所有数据点的相似度的平均值，因此，AP, SAP, AP-VSM 的偏向参数为 $S(i,i)=\sum_{j=1}^{N}\sum_{k=1}^{N}S(j,k)\Big/N^2$；同理，对于算法 SAP-SC，每一步迭代中的偏向参数设置为 $S(i,i)=\sum_{j=1}^{M}\sum_{k=1}^{M}S(j,k)\Big/M^2$。

5.3 算法仿真比较

(1)整体准确率仿真比较　针对 UCI 数据集，分别使用 AP, SAP, AP-VSM 和 SAP-SC 算法进行聚类。根据定义 1 计算整体准确率。表 2 给出了 4 种算法的整体准确率的对比，其中半监督的 SAP 和 SAP-SC 算法均使用了 10%的标签率(即已标记样本占总样本的比重等于 10%)。并且针对每一种数据集和各个算法，进行了 10 次实验仿真，计算相应的平均值和方差。通过表 2 可以看出：SAP-SC 算法的准确率比 AP 有较高提升，特别是在 optdigits, pendigits, sick, breast-cancer 数据集上准确率提升较为明显；由于 SAP 和 AP-VSM 对 AP 的相似度矩阵进行了改进，一定程度上改善了聚类的性能；SAP-SC 算法使用了组合提升的分层方法，使得聚类误差呈指数下降，其聚类效果表现最好。

针对 letter, anneal, car, dermatology 等 4 种数据集，图 6–图 9 分别仿真了算法的整体准确率随标签率的变化规律。可以看出：无监督的 AP 和 AP-VSM 整体准确率随标签率的变化有小幅震荡，但没有增长的趋势；而半监督的 SAP 和 SAP-SC 算法准确率随标签率的变大明显增大。这主要是由于 SAP 和 SAP-SC 算法均使用了少量的已标记数据来指导聚类，标记数据点越多，聚类学习越能贴近实际。

(2)计算复杂度仿真比较　针对前 4 种数据量相对较大的数据集，图 10 进行了计算复杂度的比较，实验的计算机环境为：处理器为 Core2 1.6 GHz，内存 4 GB，硬盘 250 G，编程平台为 matlab2008b。仿真表明：SAP 和 AP-VSM 的算法复杂度高于 AP, SAP-SC 算法收敛时间最短。其主要原因：SAP 和 AP-VSM 通过引入成对点约束和流形空间的概念来调整相似度矩阵，势必会增加额外的操作；SAP-SC 算法利用了“数据分层”的思想，每层只对“困难”的聚类数据进行处理，降低了算法的计算复杂度。

6　结束语

本文分析了原始的 AP 聚类算法及其改进的 SAP 和 AP-VSM 算法的优缺点，从提高聚类的准确性和降低算法计算复杂度的角度出发，提出了一

表 2 算法整体准确率仿真(%)

数据集	AP	SAP	AP-VSM	SAP-SC
audiology	42.55±1.63	52.18±0.98	60.92±2.27	74.36±1.22
letter	54.20±1.87	70.10±1.35	74.74±3.55	81.17±0.65
pendigits	74.39±1.57	85.25±1.51	89.78±1.35	96.81±0.13
optdigits	68.42 ±2.23	76.35±3.65	87.13±0.65	92.30±0.57
sick	90.00±1.03	94.26±0.70	93.91±0.26	95.34±1.69
segment	83.12±1.63	82.88±3.05	87.10±1.34	91.50±0.33
car	61.31±0.76	65.64±1.89	70.02±0.93	80.39±3.87
anneal	69.65±0.01	71.84±0.14	72.85±1.64	80.75±2.05
vehicle	49.63±3.96	56.17±2.60	58.80±2.09	70.23±1.50
pima_diabetes	65.03±5.15	70.22±4.39	78.19±2.96	86.60±1.00
breast-cancer	82.54±2.98	89.00±2.19	93.77±1.08	97.79±0.23
soybean	46.76±0.88	59.64±2.98	74.42±2.04	80.35±2.41
vote	63.51±0.64	73.21±0.15	90.35±5.07	91.32±4.11
horse-colic	66.08±0.29	75.39±2.09	73.08±0.26	81.00±0.56
dermatology	50.64±1.36	51.07±0.54	73.07±2.21	75.07±0.20
ionosphere	65.05±0.28	69.00±1.24	86.39±3.44	83.02±3.56
ecoli	46.17±0.13	60.85±0.99	78.11±3.50	85.46±3.96
glass	42.53±0.49	56.26±0.15	64.31±3.64	65.05±4.38

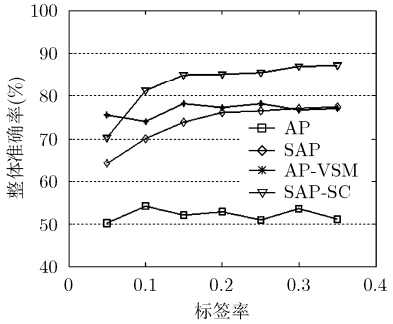

图 6 letter 数据集的准确率

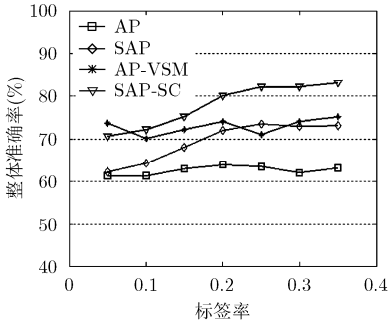

图 7 car 数据集的准确率

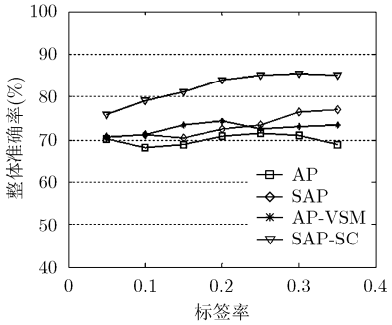

图 8 anneal 数据集的准确率

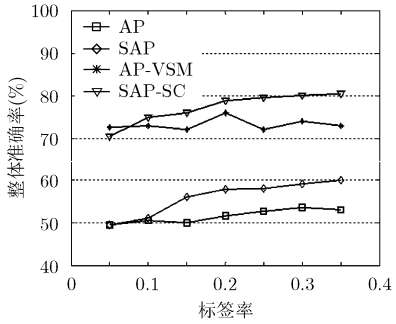

图 9 dermatology 数据集的准确率

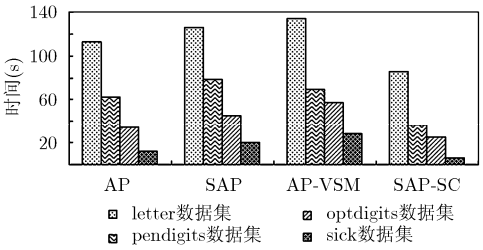

图 10 计算复杂度仿真比较

种分层组合的半监督近邻传播聚类方法 SAP-SC。SAP-SC 扩展了 SAP 的“半监督”的思想，不仅引入了“成对点约束”指导聚类，还利用“子簇标签映射”的方法来估计聚簇类型。最后，本文还对算法的准确率和计算复杂度进行了理论分析和实验仿真。论文下一步将结合最新的“半监督”方法[12,13]，从海量的数据背后提取丰富的先验信息，对近邻传播聚类算法进行不断改进。

参 考 文 献

[1] Thedoridis S and Koutroumbas K. Pattern Recognition[M]. 北京:电子工业出版社, 2010: 1–7.

[2] 宗瑜, 金萍, 陈恩红, 等. 面向 Weblog 的模糊协同聚类算法[J]. 电子与信息学报, 2012, 34(3): 543–548.
Zong Y, Jin P, Chen E H, *et al*.. Fuzzy co-clustering algorithm for Weblog[J]. *Journal of Electronics & Information Technology*, 2012, 34(3): 543–548.

[3] 刘若辰, 沈正春, 贾建, 等. 基于免疫优势的克隆选择聚类算法[J]. 电子学报, 2010, 38(4): 960–965.
Liu R C, Shen Z C, Jia J, *et al*.. Immunodomaince based on clonal selection clustering algorithm[J]. *Acta Electronica Sinica,* 2010, 38(4): 960–965.

[4] 叶有时, 唐林波, 赵保军. 一种基于聚类的深空红外多目标快速检测算法[J]. 电子与信息学报, 2011, 33(1): 77–84.
Ye Y S, Tang L B, and Zhao B J. A fast deep-space infrared multi-target detection algorithm based on clustering[J]. *Journal of Electronics & Information Technology*, 2011, 33(1): 77–84.

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

[6] 肖宇, 于剑. 基于近邻传播算法的半监督聚类[J]. 软件学报, 2008, 19(11): 2803–2813.
Xiao Y and Yu J. Semi-supervised clustering based on affinity propagation algorithm[J]. *Journal of Software*, 2008, 19(11): 2803–2813.

[7] 董俊, 王锁萍, 熊范纶. 可变相似性度量的近邻传播聚类[J]. 电子与信息学报, 2010, 32(3): 509–514.
Dong J, Wang S P, and Xiong F L. Affinity propagation clustering based on variable similarity measure[J]. *Journal of Electronics & Information Technology*, 2010, 32(3): 509–514.

[8] Freund Y and Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. *Journal of Computer and System Science*s, 1997, 55(1): 119–139.

[9] Leone M, Sumedha, and Weigt M. Clustering by soft-constraint affinity propagation: applications to gene expression data[J]. *Bioinformatics*, 2007, 23(20): 2708–2715.

[10] Jia Sen, Qian Yun-tao, and Ji Zhen. Band selection for hyper-spectral imagery using affinity propagation[C]. Proceedings of Digital Image Computing: Techniques and Applications, Canberra, 2008: l37–l41.

[11] Frank A and Asuncion A. UCI machine learning repository [EB/OL]. http://archive.ics.uci.edu/ml, 2010.

[12] Jiang He, Ren Zhi-lei, Xuan Ji-feng, *et al*.. Extracting elite pairwise constraints for clustering[J]. *Neurocomputing*, 2012, 99(1): 124–133.

[13] 宗瑜, 李明楚, 江贺. 近似骨架导向的归约聚类算法[J]. 电子与信息学报, 2009, 31(12): 2953–2957.
Zong Y, Li M C, and Jiang H. Approximate backbone guided reduction algorithm for clustering[J]. *Journal of Electronics & Information Technology*, 2009, 31(12): 2953–2957.

张　震：男，1985 年生，博士生，研究方向为数据挖掘、网络测量.

汪斌强：男，1963 年生，教授，博士生导师，研究方向为宽带信息网络.

伊　鹏：男，1977 年生，副教授，硕士生导师，研究方向为路由交换技术.

兰巨龙：男，1962 年生，教授，博士生导师，研究方向为网络体系结构.