基于稀疏 Parzen 窗密度估计的快速自适应相似度聚类方法

钱鹏江 [1] 王士同 [1,2] 邓赵红 [1]

摘 要 相似度聚类方法 (Similarity-based clustering method, SCM) 因其简单易实现和具有鲁棒性而广受关注. 但由于内含相似度聚类算法 (Similarity clustering algorithm, SCA) 的高时间复杂度和凝聚型层次聚类 (Agglomerative hierarchical clustering, AHC) 的高空间复杂度, SCM 不适用大数据集场合. 本文首先发现了 SCM 和核密度估计问题的本质联系, 并以此入手, 通过快速压缩集密度估计器 (Fast reduced set density estimator, FRSDE) 和基于图的松弛聚类 (Graph-based relaxed clustering, GRC) 算法提出了快速自适应相似度聚类方法 (Fast adaptive similarity-based clustering method, FASCM). 相比于原 SCM, 该方法的主要优点是: 1) 其总体渐近时间复杂度与样本容量呈线性关系; 2) 不依赖于人工经验的干预, 具有了自适应性. 由此, FASCM 适用于大数据集环境. 该方法的有效性在图像分割应用中进行了验证.

关键词 相似度聚类, 密度估计, 时间复杂度, 图像分割

DOI 10.3724/SP.J.1004.2011.00179

Fast Adaptive Similarity-based Clustering Using Sparse Parzen Window Density Estimation

QIAN Peng-Jiang[1] WANG Shi-Tong[1,2] DENG Zhao-Hong[1]

Abstract Similarity-based clustering method (SCM) has received much attention because it is robust and can be implemented simply and easily. However, because of its high time complexity of the embedded similarity clustering algorithm (SCA) and high space complexity of the embedded agglomerative hierarchical clustering (AHC), SCM is impractical for large data sets. In this paper, the relationship is revealed between SCM and the kernel density estimation of samples, a novel fast adaptive similarity-based clustering method (FASCM) is accordingly proposed by adopting fast reduced set density estimator (FRSDE) and graph-based relaxed clustering (GRC). The distinctive advantages of FMSSC over MSSC exist in: 1) its asymptotic linear time complexity with the data size; 2) independent on artificial experience and its adaptability. Thus, FASCM is practical for large datasets. Its effectiveness has also been demonstrated in image segmentation examples.

Key words Similarity-based clustering, density estimator, time complexity, image segmentation

Yang 等在文献 [1] 中提出了基于相似度的聚类方法 (Similarity-based clustering method, SCM), 它包含三个步骤: 1) 相关性比较算法 (Correlation comparison algorithm, CCA); 2) 相似度聚类算法 (Similarity clustering algorithm, SCA); 3) 凝聚型层次聚类 (Agglomerative hierarchical clustering, AHC). SCM 首先使用 CCA 确定幂参数 γ, 然后执行 SCA 定点迭代过程, 得到所有数据点的最终状态 (Final state), 最后基于这些最终状态使用 AHC 生成层次聚类树以确定最佳的类别数 c^* 和完成聚类. SCM 的简单性和具有一定的鲁棒性是被作者证明的优点, 但繁重的计算开销则是其很大的不足, 这点作者在文末也有所提及. 究其原因在于内含的 SCA 的时间复杂度和 AHC 的空间复杂度均达 $O(N^2)$, 这在面对大数据集时尤为明显. 此外, 我们认为 SCM 最后通过层次聚类树来确定最佳聚类数目并不直观易用, 也易受异常数据干扰. 克服和解决 SCM 的上述问题是本文的研究动机. 本文从新角度解析 SCM, 说明其相似度估计函数等价于一个 Parzen 窗 (Parzen window, PW) 密度估计算子, 基于这个发现, 采用我们的已有研究: 快速压缩集密度估计器 (Fast reduced set density estimator, FRSDE)[2−3] 和基于图的松弛聚类 (Graph-based relaxed clustering, GRC)[4] 算法, 本文就赋予了原 SCM 所不具有的低时间复杂度和自适应性, 我们把这种新方法命名为快速自适应相似度聚类方法 (Fast adaptive similarity-based clustering method, FASCM).

FASCM 也包含三个步骤: 1) 使用 FRSDE 快速得到数据集的具有稀疏权系数形式的 PW 密度估

收稿日期 2010-05-11 录用日期 2010-07-30
Manuscript received May 11, 2010; accepted July 30, 2010
国家自然科学基金 (60903100, 60975027, 60773206) 资助
Supported by National Natural Science Foundation of China (60903100, 60975027, 60773206)
1. 江南大学信息工程学院 无锡 214122 2. 江南大学数字媒体学院 无锡 214122
1. School of Information Technology, Jiangnan University, Wuxi 214122 2. School of Digital Media, Jiangnan University, Wuxi 214122

计函数[2, 5], 该密度函数本质上等价于 SCM 的相似度估计函数; 2) 执行基于该稀疏密度函数的固定点迭代过程, 求得每个数据点的最终状态, 把具有相同最终状态的数据点归属到同一分区 (Partition); 3) 构造分区相似度矩阵 W 并执行 GRC 完成最终聚类工作. 从 SCM 到 FASCM 后我们将从两方面获益: 1) FASCM 的总体渐近时间复杂度与样本容量 N 呈线性关系; 2) 融入 GRC 后, FASCM 具有了自适应性, 即无需预设类别数的前提下直观便捷地完成聚类任务. 这样我们就得到了一种适用于大数据集的有效聚类算法. 本文实验章节将验证该算法的快速性和有效性.

1 相似度聚类方法

设有数据集 $D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\} \in \mathbf{R}^d$, c 是其包含的类别数. 若用 $S(\boldsymbol{x}_j, \boldsymbol{z}_i)$ 表示数据点 $\boldsymbol{x}_j$ 与类中心 $\boldsymbol{z}_i$ 的相似程度, 则相似度聚类方法的一般目标函数可以表示为

$$J_S(Z) = \sum_{i=1}^{c}\sum_{j=1}^{N} f(S(\boldsymbol{x}_j, \boldsymbol{z}_i)) \tag{1}$$

其中, f 为一单调递增函数, $Z = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_c)$ 为类中心矩阵.

Yang 等在文献 [1] 设计 SCM 时令

$$S(\boldsymbol{x}_j, \boldsymbol{z}_i) = \exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{\beta}\right) \tag{2}$$

$$f(\cdot) = (\cdot)^\gamma, \ \gamma > 0 \tag{3}$$

这样, SCM 的目标可表示为

$$\max_{Z} \sum_{i=1}^{c}\sum_{j=1}^{N} \exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{\beta}\right)^\gamma, \ \gamma > 0 \tag{4}$$

其中, β 称为规格化项 (Normalized term). 但作者们指出鉴于幂参数 γ 可以代替 β 的作用, β 可以简单取值为样本方差

$$\beta = \frac{\sum\limits_{j=1}^{N}\|\boldsymbol{x}_j - \bar{\boldsymbol{x}}\|^2}{N}, \quad \bar{\boldsymbol{x}} = \frac{\sum\limits_{j=1}^{N}\boldsymbol{x}_j}{N} \tag{5}$$

最大化 $J_S(\boldsymbol{z})$ 就是寻找该目标函数峰值的过程. 由于数据集中包含的类别数 c 和各类中心很难预先确定, 因此文献 [1] 中用式 (6) 来替代式 (1):

$$\tilde{J}_S(\boldsymbol{x}_k) = \sum_{j=1}^{N} \exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{x}_k\|^2}{\beta}\right)^\gamma, \ k = 1, \cdots, N \tag{6}$$

即将类中心初始化为所有样本点.

SCM 首先需要确定幂参数 γ, 它给出了 CCA 算法. 文献 [1] 中, γ 一般取 5 的整数倍, 即 $\gamma = 5\,m$, $m = 1, 2, \cdots$. 于是式 (6) 可写为

$$\tilde{J}_S(\boldsymbol{x}_k)_m = \sum_{j=1}^{N}\left(\exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{x}_k\|^2}{\beta}\right)\right)^{5m}, \quad k = 1, \cdots, N; \ \ m = 1, 2, 3, \cdots \tag{7}$$

CCA 算法如下:

步骤 1. 预设参数 $m = 1$, 相关性阈值 $\varepsilon_1 = 0.97$.

步骤 2. 计算 $\tilde{J}_S(\boldsymbol{x}_k)_m$ 和 $\tilde{J}_S(\boldsymbol{x}_k)_{m+1}$ 的相关性.

步骤 3. 若相关性大于等于阈值 ε_1, 则选择当前 $5\,m$ 作为 γ 的估计值, 算法终止; 否则 $m = m + 1$, 转步骤 2.

估算出幂参数 γ 后, 下一步就是寻找 SCM 目标函数的峰值 (极大值) 点. $J_S(\boldsymbol{z})$ 对 $\boldsymbol{z}_i$ 求导:

$$\frac{\mathrm{d}J_S(Z)}{\mathrm{d}\boldsymbol{z}_i} = \sum_{j=1}^{N} 2\frac{\gamma}{\beta}(\boldsymbol{x}_j - \boldsymbol{z}_i)\left(\exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{\beta}\right)\right)^\gamma \tag{8}$$

并令式 (8) 等于 0, 则得最大化 $J_S(\boldsymbol{z})$ 的必要条件为

$$\boldsymbol{z}_i = \frac{\sum\limits_{j=1}^{N} x_j S_{ij}^\gamma}{\sum\limits_{j=1}^{N} S_{ij}^\gamma} \tag{9}$$

其中

$$S_{ij} = S(\boldsymbol{x}_j, \boldsymbol{z}_i) = \exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{\beta}\right) \tag{10}$$

显然式 (9) 不能直接求解, 需通过定点迭代 (Fixed-point iterative) 过程进行估算, 即 SCA 算法:

初始化 $\boldsymbol{z}_i^{(0)}$, $i = 1, \cdots, c$; 给定阈值 ε_2; 设迭代计数器 $\ell = 0$.

步骤 1. 由式 (10) 计算 $S_{ij}^{(\ell+1)}$.

步骤 2. 由式 (9) 计算 $\boldsymbol{z}_i^{(\ell+1)}$.

步骤 3. $\ell = \ell + 1$, 返回步骤 1, 直到 $\max_i \|\boldsymbol{z}_i^{(\ell+1)} - \boldsymbol{z}_i^{(\ell)}\| < \varepsilon_2$.

文献 [1] 指出, SCA 过程具有鲁棒性, 不论类中心如何初始化, 最终它们必将汇集到相似度目标函数的峰值点处[6−8]. 为了确保经过定点迭代过程后能同时找出目标函数的所有峰值点, 文献 [1] 初始化 $Z^{(0)} = (\boldsymbol{z}_1^{(0)}, \cdots, \boldsymbol{z}_N^{(0)}) = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$, 即所有数据点.

每个点经式 (9) 的定点迭代过程后都将得到各自的最终状态 (Final state), 容量为 N 的数据集经 SCA 后将得到 N 个最终状态, 只是有些点的最终

状态相同, 有些则不同. 因此 SCM 最后一步是使用 AHC 算法基于这些最终状态生成层次聚类树以完成聚类任务.

至此, SCM 方法的完整聚类步骤可以描述为:

步骤 1. 使用 CCA 估算幂参数 γ;

步骤 2. 执行 SCA 以得到目标函数 $J_S(Z)$ 的所有峰值点;

步骤 3. 对于所有数据点的最终状态执行 AHC;

步骤 4. 根据 AHC 生成的层次聚类树获得最佳聚类数 c^*;

步骤 5. 划分出 c^* 个类簇.

2　快速压缩集密度估计器

快速压缩集密度估计器 (Fast reduced set density estimator, FRSDE) 是在文献 [2] 中首次提出的, 并在文献 [3] 中延伸了其应用. FRSDE 算法的发现过程为: Girolami 等在文献 [5] 提出的压缩集密度估计 (Reduced set density estimator, RSDE) 算法可被视作一种特殊最小包含球 (Minimum enclosing ball, MEB) 问题[9−10], 即中心约束形最小包含球 (Center-constrained MEB, CCMEB). 限于篇幅, 这里仅列出 FRSDE 算法的实现过程, 具体细节可参见文献 [2]:

输入: 训练集 S, CCMEB 逼近精度 ε, 高斯核的窗宽 σ^2 等.

输出: 核心集 Q, 压缩集 S_r 和 FRSDE 对应密度函数 $\hat{p}(\boldsymbol{x};\sigma^2,\boldsymbol{\gamma})$ 的稀疏权向量 $\boldsymbol{\gamma}$.

训练步骤:

步骤 1. 初始化 Q_0, $\boldsymbol{c}_0$, r_0; 设置当前迭代步数 $t=1$.

步骤 2. 如果训练集中没用样本点 $\boldsymbol{x}$ 在扩展的特征空间中对应的点处于球体 $B(\boldsymbol{c}_t, (1+\varepsilon)r_t)$ 以外, 转步骤 6.

步骤 3. 在扩展的特征空间中选择距球心 $\boldsymbol{c}_t$ 最远的样本点 $\boldsymbol{x}$, 令 $Q_{t+1}=Q_t\cup\{\boldsymbol{x}\}$.

步骤 4. 获得新的 CCMEB, 也就是在扩展的特征空间得到相应的 CCMEB(Q_{t+1}), 且令 $\boldsymbol{c}_{t+1}=\boldsymbol{c}_{\mathrm{CCMEB}(Q_{t+1})}$, $r_{t+1}=r_{\mathrm{CCMEB}(Q_{t+1})}$.

步骤 5. 令 $t=t+1$, 转到步骤 2.

步骤 6. 结束训练过程并返回各输出量.

3　基于图的松弛聚类

2008 年, Lee 和 Zaïane 等在文献 [4] 中提出了求解 Normalized cuts[11−14] 的半正定规划松弛形式[15]:

$$\begin{aligned}&\max \quad -\boldsymbol{y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{y}\\&\text{s.t.} \quad \|\boldsymbol{y}\|=1,\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{y}=0\end{aligned}\tag{11}$$

其中 $\boldsymbol{e}^{\mathrm{T}}=(1,1,\cdots,1)$; $L=D-W$ 称为 Laplacian 矩阵, W 是图 $G=(V,E)$ 中各顶点的相似矩阵, $D=\mathrm{diag}\{d_1,d_2,\cdots,d_N\}$ 称为 W 的度矩阵, 其元素 $d_i=\sum_j w(i,j)$ 代表了从点 $\boldsymbol{x}_i$ 到其他点的连接度.

鉴于式 (11) 主要针对二分聚类问题, 不适合多类情况, Lee 等对式 (11) 的限制条件进行了进一步放宽, 即提出了文献 [4] 的基于图的松弛聚类 (Graph-based relaxed clustering, GRC) 方法:

$$\begin{aligned}&\max \quad -\boldsymbol{y}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{y}\\&\text{s.t.} \quad \boldsymbol{A}\boldsymbol{y}=\zeta,\quad \zeta\neq 0\end{aligned}\tag{12}$$

其中 $\boldsymbol{A}=\boldsymbol{e}^{\mathrm{T}}$. 式 (12) 的解称为 GRC 的聚类指示向量 (Clustering indicator, CI).

不同于层次聚类算法的贪婪性和趋于局部最优解等特点[16], 源于谱聚类的 GRC 收敛于全局最优解和适用于各种数据集 (凸形和非凸形)[13−14]. 此外 GRC 还具有一个显著的特性: 它的聚类指示向量 CI 图示后存在几条明显的横线就是存在几个明显的类, 同属一条横线的数据点属同一类[4]. 它比层次聚类方法常用的层次聚类树更直观方便和易用. 因此, GRC 具有便捷性 (一次性分出了所有类) 和自适应性 (算法不需要预设类别数).

4　快速自适应相似度聚类方法

总体而言, SCM 是一种简单易实现的聚类方法. 但正如本文引言中的介绍, 由于内含 SCA 的时间复杂度和 AHC 的空间复杂度均达 O(N^2), 原始 SCM 基本不适用于大数据集. 此外 AHC 对异常数据敏感、由层次聚类树确定最优聚类数目也并不直观方便, 这些都是它的主要问题. 本节将深入分析 SCM, 解析其本质, 并提出本文的快速自适应相似度聚类方法 (Fast adaptive similarity-based clustering method, FASCM).

4.1　SCM 和核密度估计问题之间的联系

前文已经介绍 SCM 的一般目标函数为

$$J_S(Z)=\sum_{i=1}^{c}\sum_{j=1}^{N}\exp\left(-\frac{\|\boldsymbol{x}_j-\boldsymbol{z}_i\|^2}{\beta}\right)^{\gamma},\ \gamma>0\tag{13}$$

首先对该目标函数进行化简和变形

$$J_S(Z)=\sum_{i=1}^{c}\sum_{j=1}^{N}\exp\left(-\frac{\|\boldsymbol{x}_j-\boldsymbol{z}_i\|^2}{\frac{\beta}{\gamma}}\right),\ \gamma>0\tag{14}$$

令 $\beta/\gamma = 2\sigma^2$, σ 为常数, 则可得到

$$J_S(Z) = \sum_{i=1}^{c}\sum_{j=1}^{N}\exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{2\sigma^2}\right) \tag{15}$$

即, SCM 的目标是

$$\max_Z(J_S) = \max_Z \sum_{i=1}^{c}\sum_{j=1}^{N}\exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{2\sigma^2}\right) \tag{16}$$

显然若我们给式 (16) 的指数表达式各增加常系数 $\frac{1}{(2\pi\sigma^2)^{d/2}}$, d 为常量, 并不影响目标解. 即

$$\max_Z \sum_{i=1}^{c}\sum_{j=1}^{N}\frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{2\sigma^2}\right) \tag{17}$$

再令 $G_{\sigma^2}(\boldsymbol{x}_j, \boldsymbol{z}_i) = \frac{1}{(2\pi\sigma^2)^{d/2}}\exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{z}_i\|^2}{2\sigma^2}\right)$, 则 SCM 的目标最终等价于

$$\max_Z \ N\sum_{i=1}^{c}\hat{p}(\boldsymbol{z}_i) \tag{18}$$

其中

$$\hat{p}(\boldsymbol{z}_i) = \frac{1}{N}\sum_{j=1}^{N}G_{\sigma^2}(\boldsymbol{x}_j, \boldsymbol{z}_i) \tag{19}$$

这时, 式 (19) 是个基于高斯核的 PW (Parzen window) 密度估计算子[17−19]. 即, 我们得到了一个重要结论: SCM 的相似度目标函数等价于一个 PW 核密度估计函数. 事实上可以把式 (18) 重新解读为: 寻找 PW 核密度估计函数的 c 个极大值点 (高密度点), 使得它们的密度之和最大. 正是这个认识为本文加速和优化原始 SCM 提供了可靠的理论基础.

4.2 FASCM

下面介绍本文 FASCM 的基本思想.

首先解释一个问题: 将 SCM 的相似度估计函数和核密度估计算子建立联系有何益处呢? 答案是可以从核密度估计问题入手, 找到降低原 SCM 算法计算开销的途径: 即利用第 3 节介绍的 FRSDE 算法. FRSDE 以与样本容量 N 呈线性关系的时间开销快速地生成形如式 (12) 具有稀疏权系数形式的密度估计算子, 且密度估计的精度不显著低于 PW. 因此完全可以用 FRSDE 替代 PW, 这在大数据集场合的作用尤显突出.

使用式 (18) 的目标函数, 初始化 $Z^{(0)} = (\boldsymbol{z}_1^{(0)}, \cdots, \boldsymbol{z}_N^{(0)}) = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$, 执行 SCA 的时间开销将为 $\mathrm{O}(N^2)$. 显然在大数据集场合可能无法承受这种时间开销. 但采用 FRSDE 代替 PW 作为相似度函数, 则式 (18) 可写为

$$\max_Z \sum_{i=1}^{c}\sum_{k=1}^{m}\beta_k G_{\sigma^2}(\boldsymbol{x}_k, \boldsymbol{z}_i) \tag{20}$$

其中, $\boldsymbol{\beta} = \{\alpha_i > 0,\ i = 1, \cdots, N\}$, $\boldsymbol{x}_k \in S_{\boldsymbol{\beta}}$; $S_{\boldsymbol{\beta}} = \{\boldsymbol{x}_i | \alpha_i > 0,\ i = 1, \cdots, N\}$ 是 FRSDE 对应的压缩集; $m = |S_{\boldsymbol{\beta}}|$ 表示压缩集数据容量. 此时若使用式 (20) 的目标函数, 类中心向量只需初始化为 $Z^{(0)} = (\boldsymbol{z}_1^{(0)}, \cdots, \boldsymbol{z}_m^{(0)}) = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)$, 执行 SCA 的时间开销将降为 $\mathrm{O}(mN)$, 且 m 远小于 N, 这是一个近乎线性的时间复杂度. 通过这种途径就能大大降低原 SCM 方法 SCA 阶段的时间开销.

SCA 后将得到所有数据点的最终状态 (Final state), 也就是迭代终止点. 原始 SCM 是通过 AHC 将所有点的最终状态进行逐个合并生成层次聚类树, 以此为依据决定最佳聚类数 c^* 并最后完成聚类任务. 但鉴于 AHC 的 $\mathrm{O}(N^2)$ 空间开销和对异常数据敏感, 且从层次聚类树判断最佳聚类数目 c^* 也并不直观方便, 需要人工经验的干预. 因此在我们的 FASCM 新方法中将不再沿用 AHC. 我们把具有相同最终状态的数据点划分成一个分区 (Partition), 用分区为聚类元素代替原 SCM 中 AHC 直接以各数据点最终状态为元素的方式. 这将大大降低新算法的空间开销. 但由于各种不确定和误差等因素 (譬如, 有限样本空间无参密度估计的统计误差、一个类簇也可能存在多个高密度点等) 的存在, 我们不能期望每个分区就是一个独立完整的类, 不能直接以不同最终状态的数量来判定聚类数目 c^*. 应该计算分区间的相似度并进行聚类. 本文第 4 节介绍的 GRC 算法在这里将被采用. GRC 是一种新型的谱聚类算法, 具有了自适应性, 能够在不预设类别数的前提下直观便捷地完成聚类任务. 要执行 GRC, 首先需要构建分区相似度矩阵 W, 此时存在两种可能的情况: 1) SCA 后生成的不同所有最终状态多又分布密集时, 可以直接计算各不同最终状态间的相似度来近似估算各分区间的相似度; 2) 不同最终状态少又分布稀疏时, 可借助于柯西 – 许瓦尔兹分歧度 (Cauchy-Schwarz divergence, CSD)[20−21] 度量准则来估算各分区间的相似度.

综上所述, FASCM 的主要步骤可以归纳如下:

输入: 大数据集 S, CCMEB 逼近精度 ε, 高斯核窗宽 σ^2 等.

输出: GRC 聚类指示向量 $\boldsymbol{y}$, 最佳聚类数目 c^*, 各数据点的类标等.

主要步骤:

步骤 1. 使用 FRSDE 快速生成大数据集 S 的稀疏权系数密度估计函数 $\hat{p}(\boldsymbol{x}; \sigma^2, \boldsymbol{\beta})$, 把该密度函数视作 FASCM 的相似度函数;

步骤 2. 基于该相似度函数 $\hat{p}(\boldsymbol{x}; \sigma^2, \boldsymbol{\beta})$ 经 SCA

过程得到所有数据点的最终状态, 具有相同最终状态的数据点构成同一分区. 设最终所得的分区数目为 ℓ;

步骤 3. 构建 $\ell \times \ell$ 分区相似度矩阵 W;

步骤 4. 基于分区相似度矩阵 W, 执行 GRC 得其聚类指示向量 $\boldsymbol{y}$;

步骤 5. 依据指示向量 $\boldsymbol{y}$ 获知最佳聚类数目 c^* 并标记所有数据点的类标, 聚类完成.

4.3 FASCM 时间复杂度分析

本节分析 FASCM 的时间复杂度, 我们按它的具体步骤进行分析:

对于步骤 1: FRSDE 的渐近近间复杂度与样本容量 N 呈线性关系, 这在文献 [2] 中已经证明.

对于步骤 2: 设 FRSDE 后得到的压缩集容量 $|S_{\boldsymbol{\beta}}| = m$, 则基于该稀疏权系数密度函数的 SCA 的时间复杂度将降低为 $\mathrm{O}(mN)$, 且 $m << N$, 这是一个近似线性的时间复杂度.

对于步骤 3: 这里存在两种方法构建分区相似度矩阵 W: 计算不同最终状态间的相似度近似代替法和借助于 CSD 度量计算. 对于前者, 由于不同最终状态数目 ℓ 相对于大数据的容量 N, 往往微不足道, 故可以忽略这部分计算开销; 而对于后者则需要注意, 对于大数据集, SCA 后所得各分区的样本容量仍可能较大, 直接计算基于 CSD 的分区相似度矩阵可能会产生较大的计算开销. 因此我们的策略是从每个分区随机抽样 μ 个样本后进行估算. 这样, 这里的时间开销为 $\mathrm{O}(\mu^2\ell^2)$, 与原样本容量 N 无关. 若选择合适的 μ, 又 $\ell << N$, 我们也将可以忽略这部分开销.

对于步骤 4: 通常 GRC 方法的时间复杂度为 $\mathrm{O}(\ell^3)$, 但由于 $\ell << N$, 类似地我们忽略这部分开销.

算法进行到步骤 4 已经得出了具体的聚类效果, 步骤 5 是善后工作, 该步开销暂不计入时间复杂度分析.

综合上述分析, 对于大数据集, FASCM 算法总体渐近时间复杂度与样本容量 N 仍呈线性关系.

5 实验结果及分析

实验环境: Pentium Dual-Core 2.7 GHz CPU, 2 G 800 MHz RAM, Windows XP SP3, Matlab 7.8.

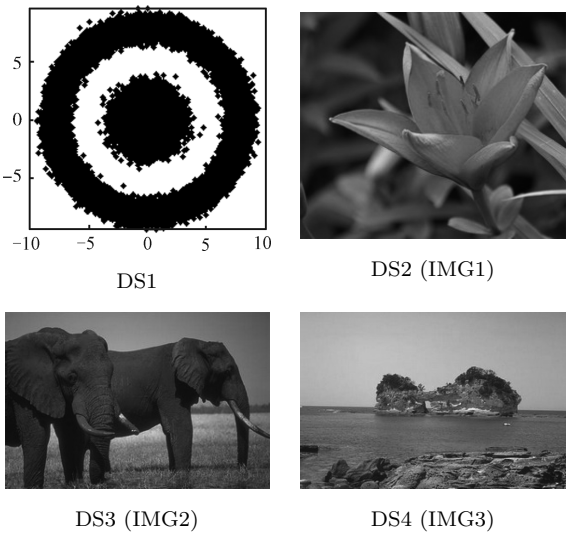
本文的所有实验, FASCM 中 FRSDE 均采用高斯密度核.

若采用以不同最终状态间的相似度近似估计分区相似度的策略时, 全部采用如下相似度度量:

$$w(i,j) = \exp\left(-\frac{\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2}{2\sigma^2}\right), \quad \sigma > 0 \qquad (21)$$

由于 FRSDE 存在迭代的二次规划 (Quadratic programming, QP) 计算, 又通常 QP 问题的时间复杂度与样本容量呈立方关系, 因此我们的实验中采用了文献 [22] 介绍的基于二阶信息的 SMO (Sequential minimal optimization) 方法以减少 QP 问题的计算开销.

我们选择了 4 个数据集进行实验研究, 如图 1 所示, 其中 DS1 是 2 维 (横坐标和纵坐标) 人造数据集, 另外 3 个数据集来自于 3 幅真实彩色图像 IMG1 ∼ IMG3, 分辨率分别为 270 像素 × 231 像素、285 像素 × 190 像素和 310 像素 × 207 像素, 其中 IMG2 和 IMG3 来自于著名的 Berkeley 图像分割数据库. 我们分别从这 3 幅图像提取特征形成实验数据集 DS2 ∼ DS4, 它们均为 5 维: 像素点所在横坐标和纵坐标、加 3 个 HSV 空间特征 (即 Hue, Saturation 和 Value). 四个数据集的样本容量分别为 111 780, 62 370, 54 150 和 64 170, 且均进行了归一化预处理.



DS1　　DS2 (IMG1)

DS3 (IMG2)　　DS4 (IMG3)

图 1　实验采用的大数据集 DS1 ∼ DS4 示意图

Fig. 1　Illustration of four large data sets DS1 ∼ DS4 adopted in the experiments

5.1 人造数据集上的实验

选用 DS1 人造数据集, 是出于两种需要: 1) 检验算法的时间复杂度; 2) 验证算法对于非凸形数据集的有效性.

本节实验, 我们从 DS1 随机抽取不同容量的子集执行 FASCM, 记录它包含的三个主要步骤 (FRSDE, SCA 和 GRC) 的执行时间, 汇总后得到总时间. 作为对比, 我们尝试同步执行 SCM 并记录它的执行时间. 这里说明一点: SCM 虽然包含 CCA, SCA 和 AHC 三个步骤, 但 CCA 作用是确定幂参数 γ, 可算作预处理过程, 因此本文实验中均不计入它的时间.

表 1 FASCM 和 SCM 在 DS1 不同抽样子集下的执行时间对照表 (秒)

Table 1 Comparison of running time of FASCM and SCM on different sampled subsets on DS1 (s)

抽样容量	FASCM				SCM		
	FRSDE	SCA	GRC	**总时间**	SCA	AHC	**总时间**
1 500	11.72	8.18	3.16	**23.06**	82.74	0.22	**82.96**
3 500	19.63	16.74	5.20	**41.57**	388.34	1.48	**389.82**
5 500	21.96	29.79	6.31	**58.06**	1050.55	7.63	**1 058.18**
7 500	26.62	53.31	7.49	**87.42**	1791.62	11.58	**1 803.20**
15 000	37.22	111.29	8.79	**157.30**	/	/	/
35 000	42.83	174.74	9.39	**226.96**	/	/	/
55 000	48.81	281.09	9.33	**339.23**	/	/	/
75 000	49.67	480.35	7.93	**537.95**	/	/	/
95 000	56.93	605.69	8.15	**670.77**	/	/	/
111 780	61.64	849.43	8.70	**919.77**	/	/	/

注: “/” 表示由于无法容忍的时间开销, SCM 算法在该抽样容量下未被执行

FASCM 和 SCM 关于 DS1 不同容量子集的执行时间细节见表 1, 该表中列出的是每种算法在每次抽样子集上执行 6 次后所得的平均时间. 按照表 1 的结果, 图 2 描绘了 SCM 和 FASCM 所需聚类时间与不同抽样容量的关系曲线, 图 3 进一步呈现了 FASCM 的三个主要步骤执行时间、总聚类时间与 DS1 不同抽样容量的关系曲线.

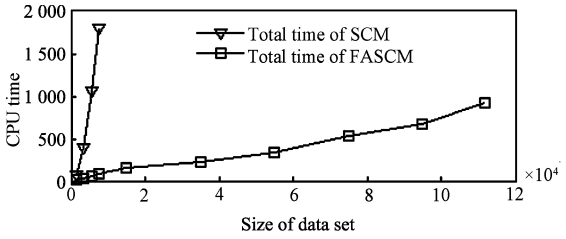


图 2 SCM 和 FASCM 所需聚类时间与 DS1 不同抽样容量关系曲线图

Fig. 2 Running time of SCM and FASCM on different sizes of samples from DS1

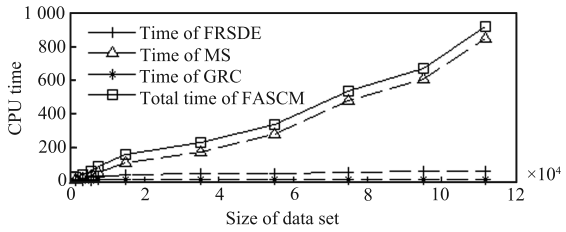


图 3 FASCM 的 3 个主要步骤执行时间及总体聚类时间与 DS1 不同抽样容量关系曲线图

Fig. 3 Running time of three main steps of FASCM on different sizes of samples from DS1

图 2 非常直观地反映了在同一抽样容量下 FASCM 所需聚类时间大大少于 SCM 这一事实, 且随着抽样容量的增大, 这一优势越加明显. 由于太大的、无法容忍的时间开销, 当抽样容量超过 7 500 后, 实验中我们没有继续同步执行 SCM 方法.

图 3 则清晰地表明由于 FASCM 各主要步骤的时间开销与抽样容量基本呈线性关系, 因此 FASCM 的总体时间复杂度必然亦与样本容量呈线性关系, 这验证了本文 FASCM 算法的总体线性时间复杂度.

图 4 中给出了 FASCM 作用于完整 DS1 所得最终状态 (Final states, FS) 分布情况、聚类指示向量 (Clustering indicator, CI) 和最终聚类结果. 通过该 CI 我们能够很清楚地知道原数据集可归成两类, 因为 CI 中存在两条明显、清晰的水平指示线. 图 4 的结果表明 FASCM 方法对于形如 DS1 的非凸形数据集亦是有效的.

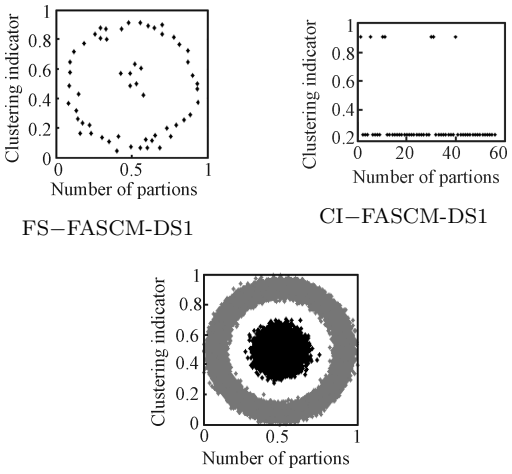


图 4 FASCM 在完整 DS1 上所得最终状态、聚类指示向量和聚类结果

Fig. 4 Final states (FS), clustering indicators (CI) and final clustering results yielded by FASCM on the whole dataset DS1

本节实验结果基于如下主要参数设置: SCM 中

设幂参数 $\gamma = 160$, 最佳聚类数 $c^* = 2$. FASCM 中, FRSDE 高斯核窗宽 $\sigma_1 = 0.0164$、逼近精度 $\varepsilon = 5.5\text{E}{-4}$, 从每个分区随机抽样 60 个数据点, 基于 CSD 构建分区相似度矩阵, CSD 参数 $\sigma_2 = 0.006$.

5.2　图像分割实验

为了方便比较, 除了 FASCM, 我们还执行两种已有算法: 模糊 C 均值聚类 (FCM) 和原 SCM. 实验将在 DS2～DS4 上进行.

由于计算开销太大, 本实验环境下, 我们无法在数据集 DS2～DS4 上直接执行 SCM. 我们采用从各数据集随机抽样一定容量子集的策略为 SCM 提供实验数据. 本节实验中我们从每个数据集随机抽样 3 500 和 5 500 两个容量的子集以运行 SCM.

表 2 列出了 FCM、SCM 和 FASCM 在 DS2～DS4 上的聚类时间. 表中关于 SCM 和 FASCM 列出的是它们的总体时间, 它们在各数据集上各步骤详细执行所需时间见表 3.

表 2　FCM、SCM 和 FASCM 关于 DS2～DS4 的聚类时间比较 (秒)

Table 2　Comparison of running time of FCM, SCM, and FASCM on DS2～DS4 (s)

算法		DS2	DS3	DS4
FCM		1.54	5.19	6.10
SCM	3 500	219.62	225.15	490.10
	5 500	834.80	773.15	1 120.46
FASCM		262.61	267.58	309.93

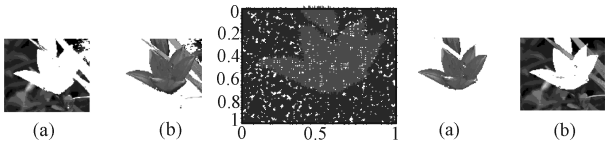
表 3 SCM 和 FASCM 在 DS2～DS4 上各步骤运行时间细节 (秒)

Table 3　More details on running time of SCM and FASCM on DS2～DS4 (s)

步骤	DS2		DS3		DS4	
	2 500	5 500	2 500	5 500	2 500	5 500
1-(1)	218.82	832.99	224.36	771.33	489.27	1 118.56
1-(2)	0.80	1.81	0.79	1.82	0.83	1.90
2-(1)	65.96		52.71		76.76	
2-(2)	196.63		214.85		233.14	
2-(3)	0.02		0.02		0.03	

注: 1-(1) 代表 SCM 的 SCA 步骤; 1-(2) 代表 SCM 的 AHC 步骤; 2-(1) 代表 FASCM 的 FRSDE 步骤; 2-(2) 代表 FASCM 的 SCA 步骤; 2-(3) 代表 FASCM 的 GRC 步骤.

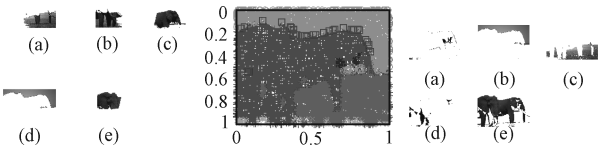
图 5～7 分别显示了 3 种算法对 IMG1～IMG3 的最终分割结果, 其中, 由于篇幅限制 SCM 仅列出了其作用于各数据集以 5 500 容量随机抽样子集的分割结果. 因为抽样数据点相对稀疏, 在以真实彩色效果还原分割结果时显示不清晰, 因此我们采用 Matlab 的 plot 函数以不同颜色和形状显示不同聚类区域的方法来代替. 图 8 描绘了 SCM 作用于 DS2～DS4 的数据子集生成的最终状态 (Final state, FS) 分布情况和 AHC 执行后所得层次聚类树 (AHC tree, AHCT). 图 9 则描绘 FASCM 在 DS2～DS4 上所得最终状态分布情况和 GRC 生成的最终聚类指示向量 (CI).



FCM-DS2　　SCM-DS2-5500　　FASCM-DS2

图 5　FCM、SCM 和 FASCM 在 DS2 的聚类结果

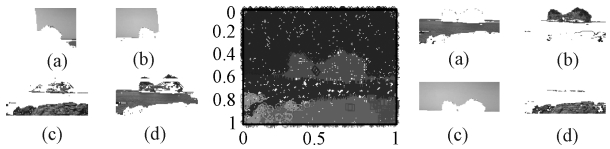
Fig. 5　Segmentation results of FCM, SCM, and FASCM on DS2



FCM-DS3　　SCM-DS3-5500　　FASCM-DS3

图 6　FCM、SCM 和 FASCM 在 DS3 的聚类结果

Fig. 6　Segmentation results of FCM, SCM, and FASCM on DS3



FCM-DS4　　SCM-DS4-5500　　FASCM-DS4

图 7　FCM、SCM 和 FASCM 在 DS4 的聚类结果

Fig. 7　Segmentation results of FCM, SCM, and FASCM on DS4

对照这些实验结果可知, FCM 虽然花费时间极少, 但分割效果相对较差, 且需要预设类别数. SCM 基本能在各数据集的小容量子集上执行并得到较好的分割效果, 但所需时间开销很大, 对于完整图像 IMG1～IMG3 是无法直接执行的. 且由于 AHC 对异常数据的敏感性, 常常导致聚类数目必须设较大值时才得到较好聚类结果. 如在 DS4 的子集上, 必须指定最佳聚类数达 6 至 7 时 SCM 才能得出相对较好的聚类效果. 而 FASCM 不仅在所有数据集上以较少时间顺利执行完毕, 而且依据 GRC 所得聚类指示向量, FASCM 具有了自适应性, 减少了人工经验的依赖. 如图 9 所示, 根据分属 GRC 的聚类指示向量, 我们能较直观地知道 DS2～DS4 可以粗略地分别分成 2 类、5 类和 4 类, 属于同一横线的分区就被合并成一个大类并给予相同类标, 聚类完成, 图像分割任务结束.

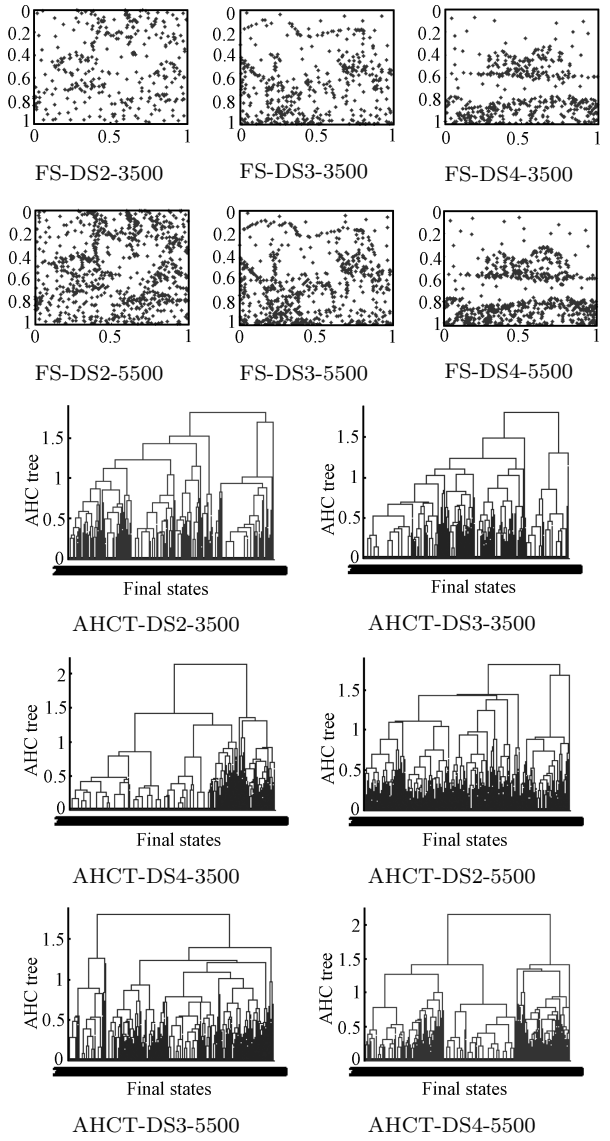


图 8 SCM 在 DS2 ∼ DS4 子集上所得最终状态 (FS) 和层次聚类树 (AHCT)

Fig. 8 Final states (FS) and AHC trees (AHCT) of SCM on 3 500 and 5 500 samples respectively from DS2 ∼ DS4

本节实验主要参数设置如下:

DS2 上, FCM 预设类别数为 2, 取系统默认参数设置. 3 500 子集上 SCM 参数 $\gamma = 110$, $\beta = 0.3298$, 最佳聚类数 $c^* = 2$; 5 500 子集上 SCM 参数 $\gamma = 210$, $\beta = 0.3301$, 最佳聚类数 $c^* = 2$. FASCM 中, FRSDE 高斯核窗宽 $\sigma_1 = 0.028$, 逼近精度 $\varepsilon =$ 5.5E−3, 采用式 (21) 以不同最终状态相似度近似估计分区相似度, 参数 $\sigma_2 = 0.072$.

DS3 上, FCM 预设类别数为 5, 取系统默认参数设置. 3 500 子集上 SCM 参数 $\gamma = 115$, $\beta = 0.2853$, 最佳聚类数 $c^* = 6$; 5 500 子集上 SCM 参数 $\gamma = 140$, $\beta = 0.2829$, 最佳聚类数 $c^* = 6$. FASCM 中, FRSDE 高斯核窗宽 $\sigma_1 = 0.03$、逼近精度 $\varepsilon =$ 6E−3, 采用式 (21) 以不同最终状态相似度近似估计分区相似度, 参数 $\sigma_2 = 0.029$.

DS4 上, FCM 预设类别数为 4, 取系统默认参数设置. 3 500 子集上 SCM 参数 $\gamma = 135$, $\beta = 0.3420$, 最佳聚类数 $c^* = 6$; 5 500 子集上 SCM 参数 $\gamma = 170$, $\beta = 0.3385$, 最佳聚类数 $c^* = 7$. FASCM 中, FRSDE 高斯核窗宽 $\sigma_1 = 0.025$、逼近精度 $\varepsilon =$ 7E−3, 采用式 (21) 以不同最终状态相似度近似估计分区相似度, 参数 $\sigma_2 = 0.0452$.

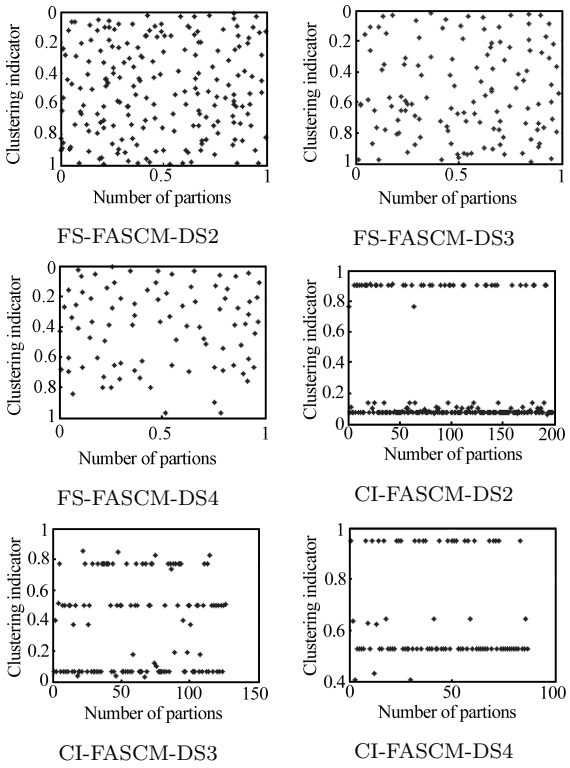


图 9 前景边缘点检测结果

Fig. 9 Foreground edge detection results

6 结论

本文 FASCM 方法通过 FRSDE 快速生成数据集的具有稀疏权系数形式的核密度估计函数, 并以此密度估计函数替代原始 SCM 相似度估计函数, 大大降低了后继 SCA 的时间开销, 从而使基于相似度的聚类算法在大数据场合具有了适用性. 此外通过融入 GRC 算法, FASCM 又具有了自适应性, 减少了人工经验的干预.

总之, 本文的主要研究贡献可以归纳为: 1) 建立了相似度聚类问题和核密度估计问题之间的联系; 2) 解决了原 SCM 的高时间开销问题, 新提出的 FASCM 方法时间复杂度与样本容量总体呈线性关系. 从而使相似度聚类方法成功运用于大数据环境.

应当指出, 当数据容量极大时, FASCM 仍面临进一步提高实用性的挑战. 根据实验结果, 若仅从聚

类速度角度评估, FASCM 显然无法与其他快速聚类算法相比, 这将是我们下一步研究的焦点.

References

1 Yang M S, Wu K L. A similarity-based robust clustering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, **26**(4): 434−448

2 Deng Z H, Chung F L, Wang S T. FRSDE: fast reduced set density estimator using minimal enclosing ball approximation. *Pattern Recognition*, 2008, **41**(4): 1363−1372

3 Chung F L, Deng Z H, Wang S T. From minimum enclosing ball to fast fuzzy inference system training on large datasets. *IEEE Transactions on Fuzzy Systems*, 2009, **17**(1): 173−184

4 Lee C H, Zaïane O, Park H H, Huang J Y, Greiner R. Clustering high dimensional data: a graph-based relaxed optimization approach. *Information Sciences*, 2008, **178**(23): 4501−4511

5 Girolami M, Chao H. Probability density estimation from optimally condensed data samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, **25**(10): 1253−1264

6 Frigui H, Krishnapuram R. Clustering by competitive agglomeration. *Pattern Recognition*, 1997, **30**(7): 1109−1119

7 Frigui H, Krishnapuram R. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, **21**(5): 450−465

8 Krishnapuram R, Frigui H, Nasraoui O. Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. *IEEE Transactions on Fuzzy Systems*, 1995, **3**(1): 29−43

9 Tsang I W H, Kwok J T Y, Zurada J A. Generalized core vector machines. *IEEE Transactions on Neural Networks*, 2006, **17**(5): 1126−1140

10 Tsang I W, Kwok J T, Cheung P M. Core vector machines: fast SVM training on very large data sets. *The Journal of Machine Learning Research*, 2005, **6**(12): 363−392

11 Shi J B, Malik J. Normalized cuts and image segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Juan, Argentina: IEEE, 1997. 731−737

12 Shi J B, Malik J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, **22**(8): 888−905

13 Higham D J, Kibble M. A Unified View of Spectral Clustering, Technical Report 02, Department of Mathematics, University of Strathclyde, UK, 2004

14 Luxburg U. A tutorial on spectral clustering. *Statistics and Computing*, 2007, **17**(4): 395−416

15 Heiler M, Keuchel J, Schnorr C. Semidefinite clustering for image segmentation with a priori knowledge. *Lecture Notes in Computer Science*, Berlin: Springer, 2005. 309−317

16 Ning H Z, Liu M, Tang H, Huang T S. A spectral clustering approach to speaker diarization. In: Proceedings of the 9th International Conference on Spoken Language Processing. Pittsburgh, USA: ISCA, 2006. 2178−2181

17 Freedman D, Kisilev P. Fast data reduction via KDE approximation. In: Proceedings of the Data Compression Conference. Snowbird, USA: IEEE, 2009. 445−445

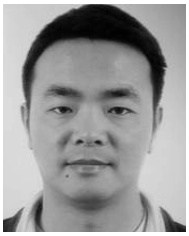
18 Chao H, Girolami M. Novelty detection employing an L_2 optimal non-parametric density estimator. *Pattern Recognition Letters*, 2004, **25**(12): 1389−1397

19 Li Cun-Hua, Sun Zhi-Hui, Chen Geng, Hu Yun. Kernel density estimation and its application to clustering algorithm construction. *Journal of Computer Research and Development*, 2004, **41**(10): 1712−1719
(李存华, 孙志挥, 陈耿, 胡云. 核密度估计及其在聚类算法构造中的应用. 计算机研究与发展, 2004, **41**(10): 1712−1719)

20 Rao S, Sanchez J C, Han S, Principe J C. Spike sorting using non parametric clustering via Cauchy Schwartz PDF divergence. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Toulouse, France: IEEE, 2006. 881−884

21 Jenssen R, Principe J C, Erdogmus D, Eltoft T. The Cauchy-Schwarz divergence and parzen windowing: connections to graph theory and mercer kernels. *Journal of the Franklin Institute*, 2006, **343**(6): 614−629

22 Fan R E, Chen P H, Lin C J. Working set selection using second order information for training support vector machines. *The Journal of Machine Learning Research*, 2005, **6**: 1889−1918



钱鹏江 江南大学信息工程学院讲师, 博士研究生. 主要研究方向为模式识别和智能计算及其应用. 本文通信作者.
E-mail: qianpjiang@126.com
(**QIAN Peng-Jiang** Lecturer and Ph. D. candidate at the School of Information and Technology, Jiangnan University. His research interest covers pattern recognition, intelligent computation and their applications. Corresponding author of this paper.)

王士同 江南大学数字媒体学院教授. 主要研究方向为人工智能、模式识别和生物信息.
E-mail: wxwangst@yahoo.com.cn
(**WANG Shi-Tong** Professor at the School of Digital Media, Jiangnan University. His research interest covers artificial intelligence, pattern recognition, and bioinformatics.)

邓赵红 江南大学信息工程学院副教授, 博士. 主要研究方向为模糊建模和智能计算.
E-mail: dzh666828@yahoo.com.cn
(**DENG Zhao-Hong** Assistant professor and Ph. D. at the School of Information and Technology, Jiangnan University. His research interest covers fuzzy modeling and intelligent computation.)