第 39 卷 第 12 期
Vol.39 No.12

计 算 机 工 程
Computer Engineering

2013 年 12 月
December 2013

• 人工智能及识别技术 •　文章编号：1000—3428(2013)12—0204—07　文献标识码：A　中图分类号：TP18

结合 mean-shift 与 MST 的 K-means 聚类算法

徐 沁，罗 斌

（安徽大学计算智能与信号处理教育部重点实验室，合肥 230039）

摘 要：针对初始点选择不当导致 K-means 陷入局部最小值问题，提出一种结合自适应 mean-shift 与最小生成树(MST)的 K-means 聚类算法。将数据对象投影到主成分分析(PCA)子空间，给出自适应 mean-shift 算法，并在 PCA 子空间内将数据向密度大的区域聚集，再利用 MST 与图连通分量算法，找出数据的类别数和类标签，据此计算原始空间的密度峰值，并将其作为 K-means 聚类的初始中心点。对 K-means 的目标函数、聚类精度和运行时间进行比较，结果表明，该算法在较短的运行时间内能给出较优的全局解。

关键词：聚类分析；K-means 算法；初始中心点；Mean-Shift 算法；主成分分析；最小生成树

K-means Clustering Algorithm Combined with mean-shift and Minimum Spanning Tree

XU Qin, LUO Bin

(Key Laboratory of Intelligence Computing and Signal Processing, Ministry of Education, Anhui University, Hefei 230039, China)

【Abstract】Given an inappropriate set of initial clustering centroids, K-means algorithm can get trapped in a local minimum. To remedy this, this paper proposes a K-means clustering algorithm combined with adaptive mean-shift and Minimum Spanning Tree(MST). The original data set is projected into Principal Component Analysis(PCA) subspace. An adaptive Mean-shift is proposed and run in the PCA subspace to let the data move to dense regions, and via the MST and graph connected component algorithm, it finds the number of clusters and the cluster indicators. According to the indicators, the density peaks are computed in the full space and taken as the initial centroids for K-means clustering. Experimental results show that the proposed algorithm can provide better global solution and higher clustering accuracy within a shorter period of execution time.

【Key words】clustering analysis; K-means algorithm; initial centroid; Mean-Shift algorithm; Principal Component Analysis(PCA); Minimum Spanning Tree(MST)

DOI: 10.3969/j.issn.1000-3428.2013.12.044

1 概述

聚类分析是将一个数据集分成若干个类(子集)，使得属于同一类的数据之间相似度最大，而属于不同类的数据之间相似度最小。K-means 算法简单且易于理解，能在并行与分布式计算机上执行以解决大规模数据集的实际问题，是目前应用最为广泛的一种聚类方法。但是 K-means 算法也存在一个根本的缺点——K-means 的函数形式是非凸的，在解决高维数据的问题时，会出现很多局部最小解，同时，求解 K-means 的 Lloyd 算法通常会从一系列不恰当的初始类中心点快速收敛到一个局部解。初始中心点的选择直接影响 K-means 聚类结果的优劣。本文针对初始点选择不当导致 K-means 陷入局部最小值的问题，提出一种结合自适应 mean-shift 与最小生成树(MST)的 K-means 聚类算法。

2 相关研究

从 K-means 算法提出至今，国内外诸多学者对该问题进行了研究。早在 1965 年 Forgy 提出在数据集中随机选择 k 个数据点作为初始的中心点，但是并不能保证被选择的点不会互相靠近(落于同一类中)，或靠近噪声点。之后一些具有代表性的研究方法有层次聚类方法以及 Kaufman 和 Katsavounidis 等人提出的 2 种依次寻找初始中心点的方法。层次聚类方法[1]是从大量的含有少数点的类开始，通过合并最近(相似)的 2 个类来依次减少类的数目直到类的数目达到目标值。如文献[2]提出的二元分裂(Binary Splitting)法，文献[3]提出的二元分裂的直接搜索方法(DSBS)，文献[4]采用 kd-树来寻找 k 个中心点。文献[5]提出的依次计算 k 个初始中心点的方法是首先取居于数据集中最中心的一点作为

基金项目：国家自然科学基金资助项目(61073116, 61211130309)

作者简介：徐 沁(1983－)，女，博士研究生，主研方向：机器视觉，模式识别；罗 斌，教授、博士

收稿日期：2012-10-29　**修回日期：**2012-12-24　**E-mail：**binluo@ahu.edu.cn

第 1 个中心点，然后选择周围有大量的数据点并且相互距离较远的数据点作为剩下的中心点。文献[6]将 Kaufman 方法与其他方法相比较，得出 Kaufman 方法表现优于其他方法。受 Kaufman 方法的启发，文献[7]研究设计了基于密度的相似性度量，并且提出了一种均衡化评价函数。KKZ 算法是文献[8]提出的另一种依次计算 k 个初始中心点的方法，它是取数据集中范数最大的点作为第 1 个初始中心点，给出了一种非中心点与中心点集的距离定义，并依据此来选择中心点。近期，文献[9]从图的角度，用 Prim 算法生成数据的最小生成树，利用 Prim 轨迹确定聚类的类别数和中心点。文献[10]采用四叉树来确定数据的初始类中心点，并控制阈值参数得到用户所需的类中心点。尽管此问题的研究方法层出不穷，但是当前运行 K-means 聚类的标准方法是重复执行 Forgy 随机方法并选取最好的结果，而此方法的缺点是耗时多。

本文从概率分布的角度来处理数据聚类问题，假设特征空间中有足够的数据点并且它们的概率密度函数能被计算出来，则可按照数据的密度谷值将数据划分成若干类，识别出数据的类别。基于密度的方法有两方面优于传统的 K-means 算法。一是 K-means 只能找出数据集中数据似圆形或球形分布的团，而基于密度的方法适用于任何形状分布的数据。二是传统的聚类方法需要输入类的数目，而基于密度的方法能根据数据自身的密度分布找出团结构，自动学习到类的数目。近期，mean-shift 算法作为一种核密度估计方法已被应用到语音和图像的分析和处理中[11-12]。本文采用 mean-shift 算法识别数据对象的团结构。

然而正确估计数据的分布存在着一些难点。首先，在很多实际问题中会出现大量的高维数据，这给准确的密度估计带来难度。其次，正确的估计密度需要大量的数据对象，粗略估计每一维需要 5 个数据点，对于 100 维的数据，就需要 5^{100} 个数据点，但是在实际应用中往往没有足够多的数据对象。再次，即使有了密度函数之后，也很难通过计算密度谷值来实现数据的划分。针对这些难点，本文从以下面来解决：(1)根据文献[13]关于 K-means 聚类的研究成果，松弛的 K-means 聚类标签(忽略标签的非负性)等于主成分分析(PCA)的主成分，K-means 聚类的全局解在 PCA 子空间里。本文采用 PCA 对高维数据降维，在 PCA 子空间中进行密度估计。(2)数据的维数降低之后，所需数据对象的数量会比原先大大减少。(3)针对密度谷值难于计算的问题，本文提出搜索密度峰值以确定聚类初始中心点。

本文首先通过主成分分析将数据投影到 PCA 子空间中。其次将 mean-shift 的步长改进为自适应变化，提出一种自适应的 mean-shift 来准确估计数据的局部密度并且加快 mean-shift 算法的收敛，通过在 PCA 子空间里运行自适应的 mean-shift，数据快速向密度大的区域移动。最后利用最小生成树，找出数据对象的类标签和密度峰值，通过计算类的松散度，去除数据中的噪声，为 K-means 聚类提供了一系列具有代表性的初始中心点。实验部分将本文方法与经典的 Forgy 随机、层次分析聚类(HAC)、Kaufman、KKZ 以及近期出现的文献[9-10]中的方法相比较，从聚类的目标函数、聚类精度和运行时间等方面证明了本文方法在保证运行时间的同时，还能给出较高的聚类精度和较优的全局解。

3 自适应 mean-shift 算法

3.1 mean-shift 算法

令 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$ 表示 d 维空间的 n 个数据点。取其中任一点作为测试点，用 $\boldsymbol{y}$ 表示，它的核密度估计为：

$$f(\boldsymbol{y})=\frac{1}{nh^d}\sum_{i=1}^{n}K\left(\frac{\boldsymbol{y}-\boldsymbol{x}_i}{h}\right) \tag{1}$$

其中，核函数 $K(\cdot)$ 必须满足对称无偏性、一致性和均匀一致性[14]。常用的核函数有多元正态核函数：

$$K_N(\boldsymbol{x})=(2\pi)^{-d/2}\mathrm{e}^{-\frac{1}{2}\|\boldsymbol{x}\|^2} \tag{2}$$

和 Epanechnikov 核：

$$K_E(\boldsymbol{x})=\begin{cases}c(1-\|\boldsymbol{x}\|^2) & \|\boldsymbol{x}\|\leqslant 1\\ 0 & \|\boldsymbol{x}\|>1\end{cases} \tag{3}$$

其中，$\|\cdot\|$ 表示欧氏距离。由于多数核函数具有如式(2)和式(3)的形式 $K(\boldsymbol{x})=k(\|\boldsymbol{x}\|^2)$，那么密度估计函数式(1)可重写为：

$$f(\boldsymbol{y})=\frac{1}{nh^d}\sum_{i=1}^{n}k\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x}_i}{h}\right\|^2\right) \tag{4}$$

这是 mean-shift 的一个重要特点。则关于 $\boldsymbol{y}$ 点的密度梯度为：

$$\nabla f(\boldsymbol{y})=\frac{2}{nh^{d+2}}\sum_{i=1}^{n}(\boldsymbol{y}-\boldsymbol{x}_i)k'\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x}_i}{h}\right\|^2\right) \tag{5}$$

其中，$k'(\|\boldsymbol{x}\|^2)=\partial k(\|\boldsymbol{x}\|^2)/\partial\|\boldsymbol{x}\|^2$ 是核梯度。为搜索核密度函数的峰值，可以利用一系列的梯度下降步骤将数据点移动到密度较大的位置(密度峰值点)。令测试点 $\boldsymbol{y}$ 经过第 t 次迭代后的坐标为 $\boldsymbol{y}^t$，则经过新一次迭代后的坐标 $\boldsymbol{y}^{t+1}$ 为：

$$\boldsymbol{y}^{t+1}=\boldsymbol{y}^t+\eta\nabla f(\boldsymbol{y}^t)=\frac{\sum_{i=1}^{n}\boldsymbol{y}^t k'\left(\left\|\frac{\boldsymbol{y}^t-\boldsymbol{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n}k'\left(\left\|\frac{\boldsymbol{y}^t-\boldsymbol{x}_i}{h}\right\|^2\right)} \tag{6}$$

其中，步长 η 为：

$$\eta=\left[\frac{2}{nh^{d+2}}\sum_{i=1}^{n}k'\left(\left\|\frac{\boldsymbol{y}_t-\boldsymbol{x}_i}{h}\right\|^2\right)\right]^{-1} \tag{7}$$

由于该密度函数有上界，则能保证迭代收敛。文献[14]已证明该算法具有二次收敛性(quadratic convergence)。相比

于其他梯度下降方法，mean-shift 的一个明显优点是在给定了核函数的情况下，没有可调的参数。

如果要找出数据集所有的密度峰值点，则采取所有数据点作为测试点的方式。测试点的初始坐标也就是数据集 $\boldsymbol{X}$，然后按照式(6)对所有测试点做梯度下降迭代直到算法收敛。这样得到的每一个密度峰值点都是由原始的若干数据点的聚集(移动)得到。例如，令经过 t 次迭代后测试点的坐标为 $\boldsymbol{Y}^t=(y_1^t,y_2^t,\cdots,y_n^t)$，则目标就是计算：

$$\boldsymbol{Y}^0,\boldsymbol{Y}^1,\cdots,\boldsymbol{Y}^t,\boldsymbol{Y}^{t+1},\cdots \tag{8}$$

测试点的初始(位置)坐标就是原始数据 $\boldsymbol{Y}^0=\boldsymbol{X}$。

3.2 自适应 Mean-Shift

由于实际中的数据点通常不是均匀分布的，并且每一步 mean-shift 迭代引起数据点的位置变化会引起数据局部结构的变化，一个固定的核函数带宽(bandwith)并不适用于局部密度的估计，因此需要设计一个根据数据点位置的变化而自适应变化的核函数带宽，以便正确地估计数据的局部密度，找出密度大的区域。

本文利用每一个数据点的 k 近邻点来计算每一个数据点的带宽。对于数据点 $\boldsymbol{x}_j$，它的 k 个近邻点为 $N_j=\{x_{j_1},x_{j_2},\cdots,x_{j_k}\}$，则带宽为：

$$h_j=\eta\frac{\sum\limits_{\boldsymbol{x}_i\in N_j}\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|}{k} \tag{9}$$

其中，η 为常数，$\eta\approx1$，对于不同的数据集，η 需要做调整，通常地，$\eta=0.5\sim0.8$。在自适应 mean-shift 迭代的过程中，数据点的坐标是变化的，因此，对于第 $t+1$ 次迭代，数据点 $\boldsymbol{x}_j$ 的带宽就应为：

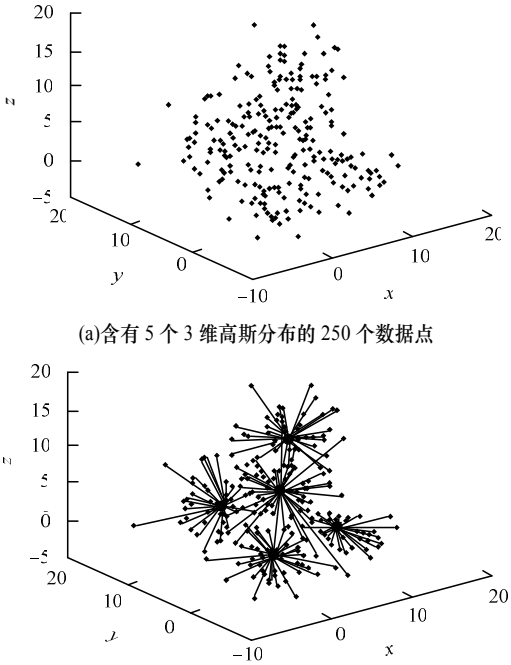
$$h_j^{t+1}=\eta\frac{\sum\limits_{\mathbf{x}_i\in N_j^t}\left\|\boldsymbol{x}_i^t-\boldsymbol{x}_j^t\right\|}{k} \tag{10}$$

其中，N_j^t 由第 t 次迭代后 $\boldsymbol{x}_j$ 的 k 个近邻点组成。则按照式(6)并采用多元正态核函数，经过 $t+1$ 次迭代后 $\boldsymbol{y}_j$ 的最新坐标为：

$$\boldsymbol{y}_j^{t+1}=\frac{\sum\limits_{i=1}^{n}\boldsymbol{y}_j^t\exp\left(-\frac{1}{2}\left\|\frac{\boldsymbol{y}_j^t-\boldsymbol{x}_i^t}{h_j^{t+1}}\right\|^2\right)}{\sum\limits_{i=1}^{n}\exp\left(-\frac{1}{2}\left\|\frac{\boldsymbol{y}_j^t-\boldsymbol{x}_i^t}{h_j^{t+1}}\right\|^2\right)} \tag{11}$$

其中，$\boldsymbol{x}_i^t$ 表示第 t 步迭代后数据点 $\boldsymbol{x}_i$ 的坐标。

图 1 给出自适应 mean-shift 的一个示例。图 1(a)是 5 个高斯分布的 3 维数据点，每一个高斯分布包含 50 个点。当自适应 mean-shift 算法收敛以后，所有数据点移动到它的密度最大的位置，每个点的原始位置和最终位置用线连接起来，如图 1(b)所示，由图中可清晰地分辨出 5 个类。然而，对于绝大多数数据集，经过几步自适应 mean-shift 迭代之后，数据点会快速聚集(移动)到它们各自密度大的区域内，数据团会逐渐显现出来。接下来将给出一种自动识别数据团的有效方法。



(a)含有 5 个 3 维高斯分布的 250 个数据点

(b)自适应 mean-shift 迭代结果

图 1 自适应 mean-shift 迭代示例

4 基于 MST 的密度峰值搜索

4.1 团结构

经过上述几步自适应 mean-shift 迭代之后，数据分布具有相似数据点之间远比不相似数据点之间的距离小的特点。因此，将每一个数据点看作图的一个节点，生成一个 MST，在这个 MST 上，近邻的点由短边连接起来形成了数据团，而不同的团之间由较长的边连接。则不同的 2 个数据团之间的距离为 2 个团元素之间的最小距离，即：

$$D(C_1,C_2)=\min_{\boldsymbol{x}_i\in C_1,\boldsymbol{x}_j\in C_2}D(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{12}$$

既然不同的数据团之间由 MST 上较长的边连接，因此定义一个阈值作为相似数据间距离的上界，这里定义：

$$\delta=0.1<D>\quad<D>=\frac{1}{n^2}\sum_{ij}D_{ij} \tag{13}$$

其中，D_{ij} 是数据点 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 的距离；$<D>$ 是任意两点之间的全局平均距离。

当 MST 上大于等于阈值的边被割断之后，MST 则被分割成若干个不连通的团。图 2~图 9 用二维数据演示了本文算法的主要步骤。图 2 是 5 个二维高斯分布的数据集。每个数据集有 50 个数据点。图 3~图 5 是分别经过 1 步、2 步、3 步自适应 mean-shift 迭代之后的所有数据点的位置。经过 4 步自适应迭代之后，在最新的数据点上生成一个 MST，如图 6~图 9 所示，MST 上有*号的边表示需要割断的长边，图中数据点旁边的数字表示该团的成员个数。由图 6 可以

看出，经过 4 步自适应 mean-shift 迭代之后，利用式(12)得到 8 个团，其中已经有 5 个分别拥有 47 个、47 个、40 个、71 个、34 个成员的团与真实的团相对应，其他 3 个团成员数较少，分别有 6 个、4 个、1 个成员。随着更多次数的自适应 mean-shift 的迭代，数据点继续向密度大的位置移动，数据团越渐明显。经过 8 步迭代之后，如图 8 所示，已经形成分别拥有 47 个、47 个、52 个、69 个、34 个成员的 5 个团，还有 1 个仅有 1 个成员的团。再经过一步自适应 mean-shift 迭代后，形成了与真实团对应的 5 个团，并且没有成员数少的团，如图 9 所示。此时，即使继续迭代(到收敛)，各团的结构也不会改变——算法收敛。

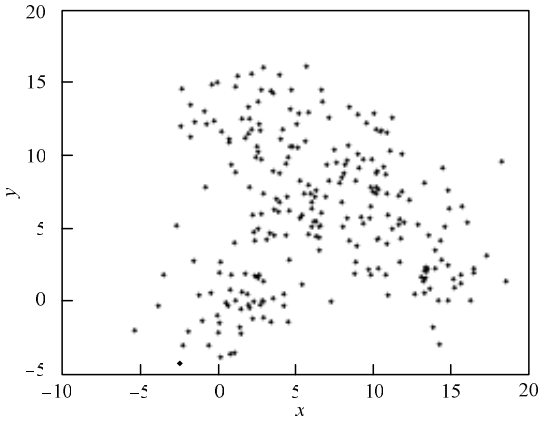


图 2　5 个均含有 50 个点的 2 维高斯分布数据

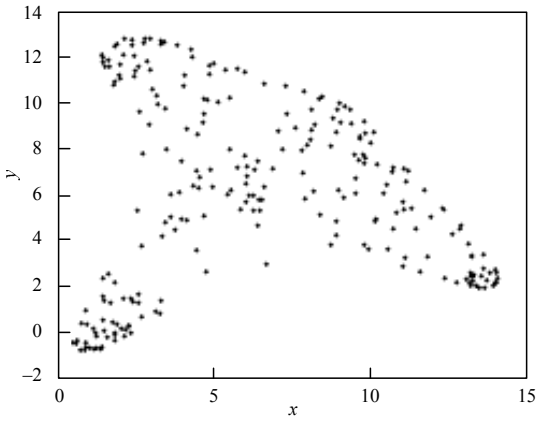


图 3　1 步自适应 mean-shift 迭代之后的数据点

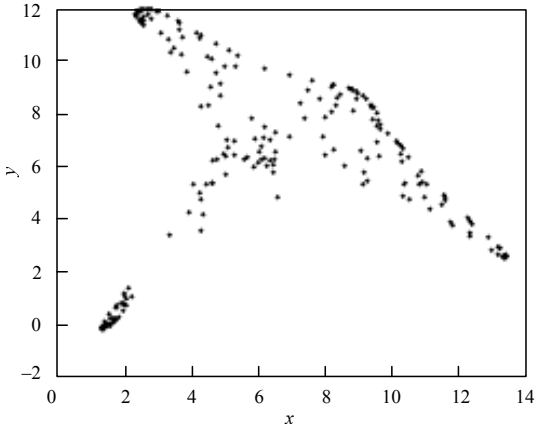


图 4　2 步自适应 mean-shift 迭代之后的数据点

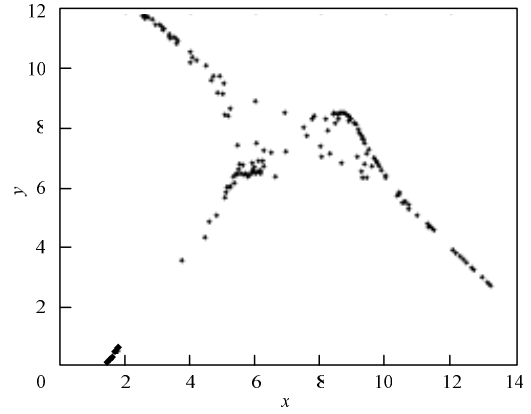


图 5　3 步自适应 mean-shift 迭代之后的数据点

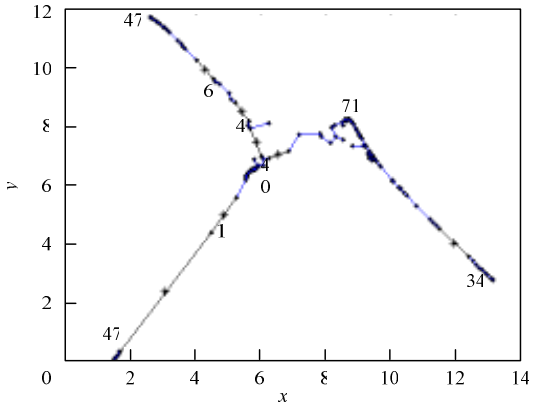


图 6　4 步自适应 mean-shift 迭代之后生成的 MST

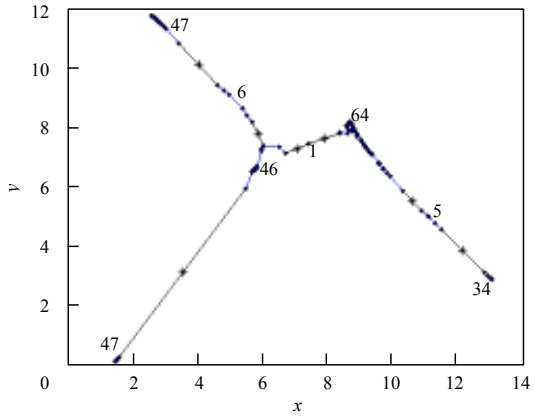


图 7　5 步自适应 mean-shift 迭代之后生成的 MST

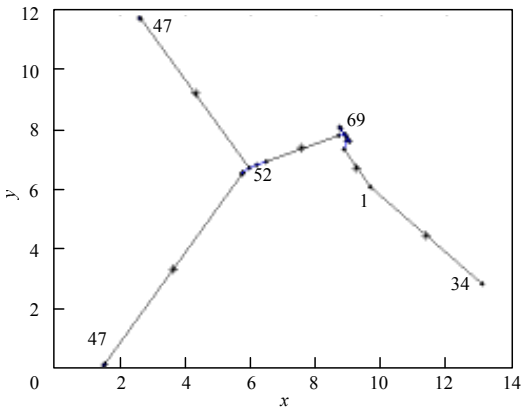


图 8　8 步自适应 mean-shift 迭代之后生成的 MST

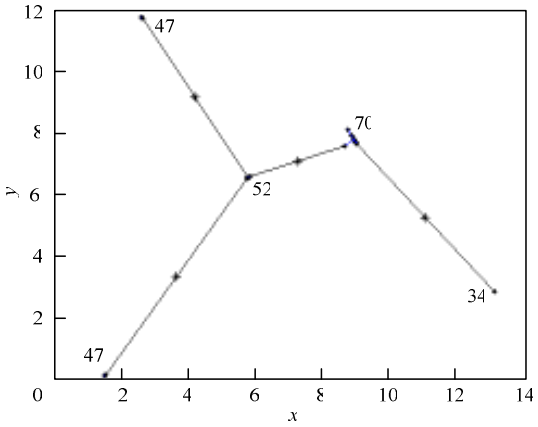


图 9 9 步自适应 mean-shift 迭代之后生成的 MST

4.2 初始中心点的确定

对于标准的 K-means 算法，类数 K 通常需要用户输入。现已有很多关于如何自动确定类数 K 的研究，但这仍是一个未解决的问题。在本文提出的基于 mean-shift 的方法中，不需要输入类数 K，相反，它能通过算法得到。

对于实际问题，用自适应 mean-shift 和图连通分量算法(graph connected component algorithm)得到的团往往会有含有少量数据点的小团。在做真正的无监督识别时，本文的初始中心点选择时仍然使用这些小的团。

但在做性能评价时，很多测试数据已经给定了类数 K，此时，由本文方法得到的类别数通常会与给定的 K 不相符，实际上会得到比 K 多的类，其中，有一些比较小的类(比如成员数为 2 个~5 个)。在这种情况下，只关心原始数据的 K 个数据团的中心点。而那些得到的小的数据团对于搜索团结构并无帮助，因此需要去除。但是什么样的数据团为小数据团？本文采用松散度来表征数据团的大小和松紧度，对于有 r 个成员的数据团 $C_k=\{x_1,x_2,\cdots,x_r\}$，其松散度定义为：

$$disp_k=\frac{\sum_{x_i\in C_k}\sum_{x_j\in C_j,j\neq i}D_{ij}}{r(r-1)/2} \tag{14}$$

如果数据团含有大量数据点同时点分布紧密，那么它的松散度就小。因此，应去除松散度大的团。

4.3 MS-MST 算法

以下将本文算法简称为 MS-MST 算法，具体如下：

输入 PCA 低维数据

输出 K 个初始中心点

步骤 1 将输入的 PCA 低维数据进行若干步的自适应 mean-shift 迭代；

步骤 2 以更新后的数据点作为图的顶点，两点之间的距离 D_{ij} 作为边的权值，建立最小生成树(MST)；

步骤 3 用阈值 δ 将 MST 上的长边进行切割，如 $D_{ij}\geqslant\delta$，则割边 (i,j)；

步骤 4 利用图连通分量算法得到数据的类别数和所有点的类标签；

步骤 5 如果得到的类数比给定的 K 大，计算每个类的松散度，取松散度最小的 K 个类的中心作为 K-means 聚类初始中心点。

5 实验结果与分析

为了验证本文算法的有效性，将本文算法与其他 6 种不同方法对真实图像数据集进行聚类和比较。进行比较实验的 6 种方法分别为：(1) Forgy 随机方法；(2) KKZ 方法；(3)Kaufman 方法；(4)HAC 方法；(5)文献[9]方法；(6)四叉树[10]方法。实验基于 Matlab 7.0 平台，采用 Pentium(R) Dual-Core CPU 3.00 GHz，内存 1.96 GB 的计算机完成。

5.1 数据描述

实验数据包括 4 个数据集：

AT&T 人脸图像库[15]是 40 个人的各 10 张不同的人脸图片，每张图片的大小为 92×112，256 灰度级。实验中采用像素特征，每幅人脸图像被转化成大小为 23×28 并被拉成 644 维向量

MNIST 手写数字图像库[16]是 0~9 的 8 bit 灰度级图像组成，每个(类)数字含有 6 000 幅训练图像。本文选出每类的前 50 幅图像并拉成 784 维的向量作为实验数据。

Binary Alphabet 手写字母图像数据库[16]由 26 个字母 A~Z 的二值图像组成，其中，每个(类)字母有 39 幅图像，实验中将每幅图像拉成 320 维的向量。

Coil-20 图像库[17]由 20 个物体的 1 440 幅图像组成，每个物体的 72 幅图像由不同光照条件和不同几何形变获得。该数据库有 2 个图像数据集。本文使用第 2 个大小为 1 440 的标准化数据库，并且取每类的前 50 个图像组成实验数据。

5.2 K-means 聚类

由于 Forgy 随机方法的结果不稳定，因此实验将随机方法执行 1 000 次，记录每一次 K-means 聚类的目标函数值：

$$J_K=\sum_{k=1}^{K}\sum_{i\in C_k}(x_i-m_k)^2 \tag{15}$$

画出 1 000 次结果的概率分布，如图 10~图 13 所示，本文方法 MS-MST 的结果是固定的。

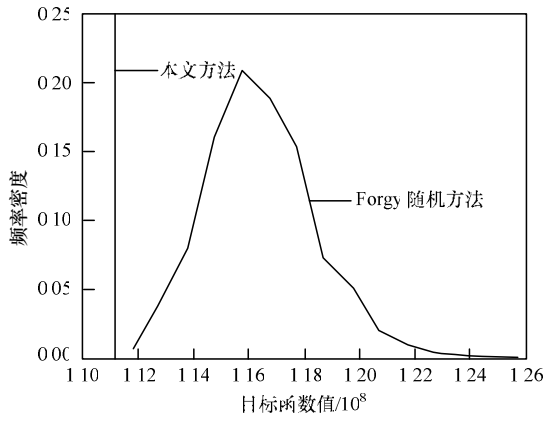


图 10 AT&T 数据集的 K-means 目标函数值

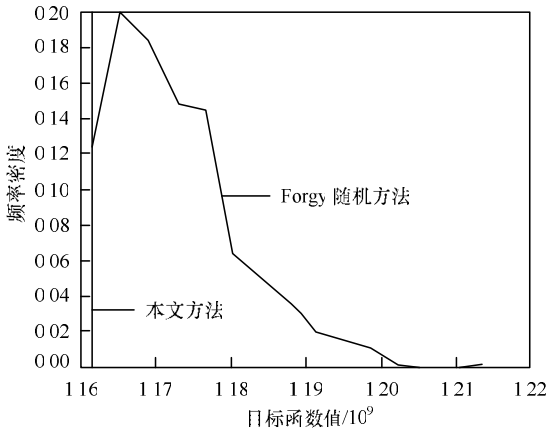


图 11　MNIST 数据集的 K-means 目标函数值

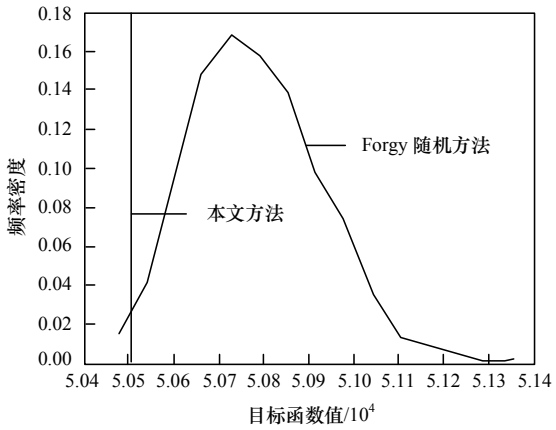


图 12　Binary Alphabet 数据集的 K-means 目标函数值

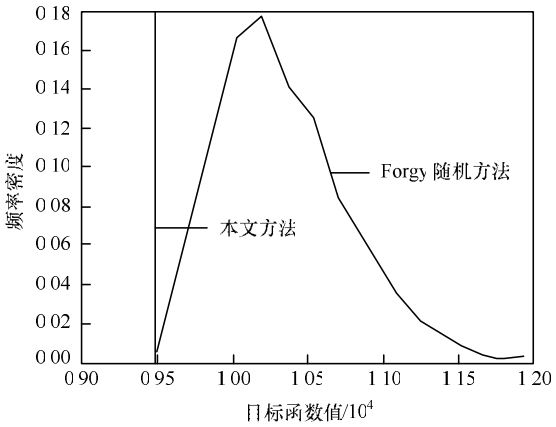


图 13　Coil-20 数据集的 K-means 目标函数值

由图中可看出，由本文方法得到的目标函数值远远比 Forgy 随机方法的要低，更接近全局解。而对于 Binary Alphabet 数据集，由 1 000 次 Forgy 随机方法得到的结果中有 11 次比 MS-MST 结果好，那么由 Forgy 随机方法得到的结果以 11/1 000 = 0.011 的概率比 MS-MST 好，显然，随机方法比 MS-MST 好的概率很小。另一方面，对于其他结果固定的方法，除了 K-means 的目标函数值，还计算了聚类精度，即正确聚类的比率，它是由匈牙利算法调整混淆矩阵的行和列，使混淆矩阵的对角线上的元素和最大得到，如表 1~表 4 所示。

表 1　AT&T 数据集上的 K-means 聚类结果

方法	执行时间/s	目标函数值	聚类精度
KKZ	2.51	1.158 3e+08	0.735 0
HAC	532.17	1.115 3e+08	0.740 0
Kaufman	373.71	1.112 2e+08	0.745 0
文献[9]方法	125.21	1.127 9e+08	0.723 1
四叉树	91.26	1.114 3e+08	0.732 4
MS-MST	32.08	1.111 6e+08	0.747 5

表 2　MNIST 数据集上的 K-means 聚类结果

方法	执行时间/s	目标函数值	聚类精度
KKZ	2.80	1.164 2e+09	0.542 0
HAC	1 237.99	1.167 3e+09	0.560 0
Kaufman	54.15	1.166 7e+09	0.554 0
文献[9]方法	73.91	1.165 9e+09	0.567 8
四叉树	56.28	1.162 7e+09	0.591 0
MS-MST	51.52	1.161 4e+09	0.604 0

表 3　Binary Alphabet 数据集上 K-means 聚类结果

方法	执行时间/s	目标函数值	聚类精度
KKZ	6.43	5.063 6e+04	0.481 2
HAC	7 932.17	5.055 2e+04	0.501 5
Kaufman	805.93	5.053 4e+04	0.476 3
文献[9]方法	652.10	5.052 9e+04	0.478 9
四叉树	510.39	5.051 3e+04	0.493 1
MS-MST	326.76	5.050 3e+04	0.508 9

表 4　Coil-20 数据集上的 K-means 聚类结果

方法	执行时间/s	目标函数值	聚类精度
KKZ	2.92	1 0019.9	0.689 5
HAC	1 388.47	9 505.2	0.692 0
Kaufman	205.53	9 492.2	0.718 0
文献[9]方法	165.21	9 564.8	0.692 8
四叉树	120.95	9 510.7	0.715 2
MS-MST	51.58	9 484.1	0.726 0

表 1~表 4 给出不同方法的运行时间、目标函数值及聚类精度。由表中可看出，虽然 KKZ 方法的运行时间比 MS-MST 短，但是由 KKZ 方法获得的聚类精度并没有 MS-MST 高，目标函数的结果也比 MS-MST 差。相比于其他方法，本文算法均能在较短的时间内能得到更低的目标函数值和较高的聚类精度。

6　结束语

本文从密度估计的角度考虑数据聚类问题，提出一种结合自适应 mean-shift 和最小生成树的 K-means 聚类算法。首先利用自适应 mean-shift 将数据在 PCA 子空间中进行聚类，再生成最小生成树，并运用图连通分量算法找出类别数和数据标签，最后在原始空间中计算出 K-means 聚类的中心点。实验结果证明，该方法有助于 K-means 聚类找到全局最优解。

参考文献

[1] Banfield J, Raftery A. Model-based Gaussian and Non-gaussian Clustering[J]. Biometrics, 1993, 49(1): 803-821.

[2] Linde Y, Buzo A, Gray R M. An Algorithm for Vector Quantizer Design[J]. IEEE Transactions on Communications, 1980, 28(1): 84-95.

[3] Huang C M, Harris R W. A Comparison of Several Vector Quantization Codebook Generation Approaches[J]. IEEE Transactions on Image Processing, 1993, 2(1): 108-112.

[4] Redmond S J, Heneghan C. A Method for Initialising the K-means Clustering Algorithm Using Kd-trees[J]. Pattern Recognition Letters, 2007, 28(8): 965-973.

[5] Kaufman L, Rousseeuw P J. Finding Groups in Data: An Introduction to Cluster Analysis[M]. New Jersey, USA: Wiley Interscience, 1990.

[6] Peña J M, Lozano J A, Larrañaga P. An Empirical Comparison of Four Initialization Methods for the K-means Algorithm[J]. Pattern Recognition Letters, 1999, 20(10): 1027-1040.

[7] 汪 中, 刘贵全, 陈恩红. 一种优化初始中心点的 K-means 算法[J]. 模式识别与人工智能, 2009, 22(2): 299-304.

[8] Katsavounidis I, Kuo C C J, Zhang Z. A New Initialization Technique for Generalized Lloyd Iteration[J]. IEEE Signal Processing Letters, 1994, 10(1): 144-146.

[9] Galluccio L, Michel O, Comon P, et al, Graph Based K-means Clustering[J]. Signal Processing, 2012, 92(9): 1970-1984.

[10] Bishnu P S, Bhattacherjee V. Software Fault Prediction Using Quad Tree-based K-means Clustering Algorithm[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(6): 1146-1150.

[11] 杨 烜. Mean Shift 的渐进无偏变换图像配准[J]. 电子与信息学报, 2012, 34(2): 393-397.

[12] Ayllón D, Gil-Pita R, Amores P J, et al. Speech Source Separation Using a Generalized Mean Shift Algorithm[J]. Signal Processing, 2012, 92(9): 2248-2252.

[13] Ding C, He Xiaofeng. K-means Clustering via Principal Component Analysis[C]//Proceedings of the 21th International Conference on Machine Learning. Banff Alberta, Canada: [s. n.], 2004: 225- 232.

[14] Cheng Yizong. Mean Shift, Mode Seeking, and Clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(8): 790-799.

[15] Cambridge University Computer Laboratory: The Database of Faces[EB/OL]. (2002-10-20). http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[16] Roweis S. Data for MATLAB hackers[EB/OL]. (2010-10-20). http://www.cs.nyu.edu/~roweis/.

[17] Keysers D. COIL-RWTH[EB/OL]. (2004-05-20). http://www-i6.informatik.rwth-aachen.de/~keysers/COIL-RWTH/.

编辑 索书志

(上接第 203 页)

参考文献

[1] Gibson D, Punera K, Tomkins A. The Volume and Evolution of Web Page Templates[C]//Proc. of the 14th International Conference on World Wide Web. New York, USA: ACM Press, 2005.

[2] Rahman A, Alam H, Hartono R. Content Extraction from HTML Documents[C]//Proc. of the 1st International Workshop on Web Document Analysis. New York, USA: ACM Press, 2001.

[3] Wang Jiying, Lochovsky F H. Data-rich Section Extraction from HTML Pages[C]//Proc. of the 3rd International Conference on Web Information Systems Engineering. Washington D. C., USA: IEEE Computer Society, 2002.

[4] 欧健文, 董守斌, 蔡 斌. 模板化网页主题信息的提取方法[J]. 清华大学学报: 自然科学版, 2005, 45(S1): 1743-1747.

[5] Sun Fei, Song Dandan, Liao Lejian. Dom Based Content Extraction via Text Density[C]//Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM Press, 2011.

[6] Weninger T, Hsu W H, Han J. CETR: Content Extraction via Tag Ratios[C]//Proc. of the 19th International Conference on World Wide Web. New York, USA: ACM Press, 2010.

[7] Abdul P, Qureshi R, Memon N. Hybrid Model of Content Extraction[J]. Journal of Computer and System Sciences, 2012, 78(4): 1248-1257.

[8] Cai Deng, Yu Shipeng, Wen Jirong, et al. VIPS: A Vision Based Page Segmentation Algorithm[EB/OL]. (2003-10-20). http://research.microsoft.com/apps/pubs/default.aspx?id=70027.

[9] Song Mingqiu, WU Xintao. Content Extraction from Web Pages Based on Chinese Punctuation Number[C]//Proc. of International Conference on Wireless Communications, Networking and Mobile Computing. [S. 1.]: IEEE Press, 2007.

[10] 张志刚, 陈 静, 李晓明. 一种 HTML 网页净化方法[J]. 情报学报, 2004, 23(4): 387-393.

[11] 陈竹敏. 面向垂直搜索引擎的主题爬行技术研究[D]. 济南: 山东大学, 2008.

[12] 聂 卉, 张津华. 分块布局下的主题型网页的内容抽取[J]. 情报学报, 2012, 31(1): 31-39.

编辑 索书志