一种快速、鲁棒的有限高斯混合模型聚类算法

胡庆辉[1,2,3] 丁立新[1,2] 陆玉婧[3] 何进荣[1,2]
(武汉大学软件工程国家重点实验室 武汉 430072)[1] (武汉大学计算机学院 武汉 430072)[2]
(桂林航天工业学院信息工程系 桂林 541004)[3]

摘 要 有限混合模型聚类是一种基于概率模型的有效聚类方法。针对高斯混合模型的聚类算法,分别对模型的成分混合系数及样本所属成分的概率系数施加熵惩罚算子,实现对模型成分数的两级控制,快速消除无效成分,使算法能在很少的迭代次数内收敛到确定解。传统算法对初始值(成分数目 c 需事先指定)的设置非常敏感,容易导致 EM 算法陷入局部最优解或收敛到解空间的边界,而文中的算法对初始值的设定没有特殊的要求,实验证明其具有很好的鲁棒性。

关键词 高斯混合模型,聚类,信息熵,EM 算法
中图法分类号 TP301.6 **文献标识码** A

Rapid Robust Clustering Algorithm for Gaussian Finite Mixture Model

HU Qing-hui[1,2,3] DING Li-xin[1,2] LU Yu-jing[3] HE Jin-rong[1,2]
(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)[1]
(School of Computer, Wuhan University, Wuhan 430072, China)[2]
(School of Information Engineering, Guilin University of Aerospace Technology, Guilin 541004, China)[3]

Abstract Finite mixture model is an effective clustering method based on probability model. Aiming at the clustering algorithm of Gaussian mixture model. This paper imposed entropy penalized operators on the mixed coefficients of components and the labels of samples respectively, which brings to two levels controls for the number of components and rapid reduction of the illegitimate ones. Thus the algorithm converges to exact solutions with only a few iterations. Since the traditional algorithm is very sensitive to the initial values (for example, the number of components must be set in advance), which often leads to the EM algorithm to fall into local optima or converges to the boundary of the solution space, the new algorithm of this paper is very robust and has no special demands for the initializations, just testified by the experiments.

Keywords Gaussian finite mixture model, Clustering, Entropy, EM algorithm

1 引言

在很多科学与工程领域,均会涉及到大量的数据统计分析与数学建模。有限概率混合模型就是一种在统计数据分析中最常用也是最有效的数学方法之一。它是有限个独立概率模型的凸组合,通过使用多个成分(其中每一个成分即为一个独立的概率分布)来描述一个复杂的数据分布。不管数据分布的结构如何复杂,混合模型总是能通过增加成分的方式来描述数据分布的局部特性。这使得它成为有效的密度估计及聚类工具之一,在机器学习与数据挖掘领域中具有重要的地位。

有限概率混合模型首先要假定数据服从某个混合概率分布,常见的如高斯分布[1,7]、t 分布[2,3,9]、皮尔逊分布[4]等等。但实际的数据有时不一定满足前面的这些分布,为了使混合模型聚类算法有更强的适应性,一些研究者提出了若干其他分布模型,如文献[8]中提出了非中心 t 分布的混合鲁棒模型;文献[6,10]使用了基于 Dirichlet 分布的混合模型并将其成功应用于图像分割;文献[5]中,作者通过研究广义多元分析理论,提出了基于椭球等高分布混合模型的聚类算法,解决了有限混合分布模型的聚类分析中分量密度函数选择问题。

利用有限混合模型进行聚类时,样本属于不完全数据(缺失样本的类别标签),通常采用 EM(Expectation Maximization)算法求解最大似然值,而 EM 算法收敛速度慢,对初值(比如混合模型成分数的初始值)的设置很敏感,容易收敛到局部最优解或参数的边界[9]。一些文献专门针对 EM 算法的初始化问题[11]及混合模型成分的估计问题[12,13]进行了讨论,

到稿日期:2012-11-01 返修日期:2013-04-01 本文受国家自然科学基金(60975050),中央高校基本科研业务费专项基金(6081014),武汉大学研究生自主科研项目(2012211020209)资助。

胡庆辉(1976-),男,博士生,副教授,主要研究方向为智能信息处理、数据挖掘等,E-mail:huqinghui2004@126.com;丁立新(1967-),男,博士,教授,博士生导师,主要研究方向为智能计算及其理论;陆玉婧(1973-),女,硕士,讲师,主要研究方向为汽车检测;何进荣(1984-),男,博士生,主要研究方向为半监督学习及降维。

但很少涉及到初始成分数的设置对 EM 算法鲁棒性的影响。文献[7]中作者提出了一个最小信息长度(MML)收敛准则，通过使用一个集成估计和模型选择准则的策略，一定程度上提高了 EM 算法的鲁棒性，但是由于初始成分数设置不同，该算法仍然可能会收敛到不同的解。文献[1]中，作者提出了另一种鲁棒性高斯混合模型聚类算法，成分数初值直接等于样本个数，对成分的混合系数施加熵惩罚算子，在迭代中根据一定准则自动减少成分数目。这使得 EM 的迭代不依赖于初始成分数，大量实验证明该算法具有很好的鲁棒性。但是成分数越大，计算的时间代价往往呈指数增加，因此，使算法在较少迭代次数内快速降低成分数目是一个有效提高该算法效率的办法。

本文针对文献[1]的算法，提出了同时对样本所属成分的概率系数施加熵惩罚算子，两级控制模型的成分数，来达到在确保算法收敛到正确解的同时进一步快速收敛的目的。实验表明，改进的算法较原有算法在收敛到确切解时，迭代次数明显减少，收敛速度明显提高。

2 高斯混合模型的基本形式

设$\{x_1,x_2,\cdots,x_n \mid x\in R^d\}$为随机变量 X 的 n 个随机样本值，则含有 c 个成分的 d 变量混合模型的概率密度函数表示为：

$$f(x;\omega,\theta)=\sum_{i=1}^{c}\omega_i f(x;\theta_i) \tag{1}$$

式中，$\omega_i>0$，表示混合模型中各成分的比例系数，并且满足 $\sum_{i=1}^{c}\omega_i=1$，$f(x;\theta_i)$表示第 i 成分的概率密度。当混合模型的各成分为高斯分布时，上式可以写成：

$$\begin{aligned}f(x;\omega,\mu,\Sigma)&=\sum_{i=1}^{c}\omega_i f(x;\mu_i,\Sigma_i)\\&=\sum_{i=1}^{c}\omega_i(2\pi)^{-(d/2)}|\Sigma_i|^{-1/2}e^{(-1/2)\delta(x,\mu_i,\Sigma_i)}\end{aligned} \tag{2}$$

式中，μ_i 和 Σ_i 分别为第 i 成分的均值向量和协方差，d 为样本的维数。$\delta(x,\mu;\Sigma)$表示 x 与 μ 的 Mahalanobis 距离。

设 $Z_1,Z_2,\cdots,Z_n$ 分别为样本 $x_1,x_2,\cdots,x_n$ 所属成分的概率向量，$z_{ij}=Z_j(i)$取值为 0 或 1，当 $z_{ij}=1$ 时，表示样本 x_j 属于第 i 个成分，否则不是。则最大似然法求解参数的 log 似然函数为：

$$L(\omega,\mu,\Sigma;x_1,x_2,\cdots,x_n)=\sum_{i=1}^{n}\sum_{k=1}^{c}z_{ki}\ln[\omega_k f(x_i;\theta_k)] \tag{3}$$

3 EM 算法估计未知参数

似然函数式(3)中，z_{ki} 属于缺失参数，通常用 EM 算法迭代求解。EM 算法分为以下两步。

E-Step：求解参数 z_{ki}：

$$\begin{aligned}z_{ki}&=\frac{\omega_k f(x_i;\theta_k)}{\sum_{s=1}^{c}\omega_s f(x_i;\theta_s)}\\&=\frac{\omega_k(2\pi)^{-d/2}|\Sigma_k|^{-(1/2)}e^{(-1/2)\delta(x_i,\mu_k,\Sigma_k)}}{\sum_{s=1}^{c}\omega_s(2\pi)^{-d/2}|\Sigma_s|^{-(1/2)}e^{(-1/2)\delta(x_i,\mu_s,\Sigma_s)}}\end{aligned} \tag{4}$$

M-Step：对式(3)的 μ_k、Σ_k 和 ω_k 求最大，有：

混合系数

$$\omega_k=\sum_{j=1}^{n}\hat{z}_{kj}/n \tag{5}$$

各成分的均值向量

$$\mu_k=\sum_{j=1}^{n}\hat{z}_{kj}u_{kj}x_j/\sum_{j=1}^{n}\hat{z}_{kj}u_{kj} \tag{6}$$

各成分的协方差矩阵

$$\Sigma_k=\sum_{j=1}^{n}\hat{z}_{kj}u_{kj}(x_j-\mu_k)(x_j-\mu_k)^{\mathrm{T}}/\sum_{j=1}^{n}\hat{z}_{kj}u_{kj} \tag{7}$$

4 一个鲁棒的高斯混合模型聚类算法

在多变量有限混合模型的概率密度函数中，混合系数 ω_k 代表第 k 成分存在的概率，ω_k 越大，表明该成分越重要，在下一次迭代中有更多的机会被保留；反之，当 ω_k 很小(小于某个预先定义的阈值 ξ，实验设置为 $1/n$)时，表明该成分应该被消除，从而达到减少成分数的目的。

用 $-\ln\omega_k$ 表示第 k 成分的信息量，$-\sum_{k=1}^{c}\omega_k\ln\omega_k$ 为平均信息量(即信息熵)，当 $\omega_k=1/c(\forall k\in 1,2,\cdots,c)$时，则认为没有关于 ω_k 的信息量。因此我们的目标是利用信息熵使目标函数达到最大值，以获得 ω_k 更多的信息，这可以通过最小化 $-\sum_{k=1}^{c}\omega_k\ln\omega_k$来实现。最小化 $-\sum_{k=1}^{c}\omega_k\ln\omega_k$ 等价于最大化 $\sum_{k=1}^{c}\omega_k\ln\omega_k$，于是我们在原始的目标函数(即式(3))基础之上添加了一个惩罚项 $\sum_{i=1}^{n}\sum_{k=1}^{c}\omega_k\ln\omega_k$，则似然函数式(3)变为：

$$J(\omega,\theta)=\sum_{i=1}^{n}\sum_{k=1}^{c}\hat{z}_{ki}\ln(\omega_k f(x_i;\theta_k)+\beta\sum_{i=1}^{n}\sum_{k=1}^{c}\omega_k\ln\omega_k \tag{8}$$

式中，β 为调节系数，取值在(0,1)之间。β 越大，则惩罚越强，在最开始迭代的时候，由于成分数目 c 较大，可以使用较大的值来加速成分的减少，随着迭代次数的增加，β 按一定比例减少。当成分数趋于稳定时设置 $\beta=0$，使算法按传统方式进行迭代。

在式(8)中，惩罚项始终为负值，能够对 EM 算法起到很好的调节作用。在等式约束 $\sum_{k}^{c}\omega_k=1$ 的条件下，利用拉格朗日乘子法，对式(8)的 ω_k 求一阶偏导，得到 ω_k 的迭代公式：

$$\omega_k^{t+1}=\sum_{i=1}^{n}\hat{z}_{ki}^{t}/n+\beta\omega_k^t(\ln\omega_k^t-\sum_{i=1}^{c}\omega_i^t\ln\omega_i^t) \tag{9}$$

式(9)表明，如果 $\ln\omega_k^t-\sum_{i=1}^{c}\omega_i^t\ln\omega_i^t<0$，则混合模型的第 k 成分在第 $t+1$ 次迭代的时候，其混合系数 ω_k^{t+1}将比没有施加惩罚算子时要小。ω_k^t 越小，则 ω_k^{t+1} 减小的趋势越大。当 ω_k^{t+1} 减小到阈值 ξ 时，可以认为第 k 成分是无效的，可以消除，同时 $c=c-1$。

随着成分数目 c 的减少，为保持约束条件 $\sum_{i=1}^{c}\omega_i=1$ 和 $\sum_{k=1}^{c}\hat{z}_{kj}=1$，需要对其做如下调整：

$$\omega_k^{t+1}=\omega_k^{t+1}/\sum_{s=1}^{c}\omega_s^{t+1} \tag{10}$$

$$\hat{z}_{ki}^{t+1}=\hat{z}_{ki}^{t+1}/\sum_{s=1}^{c}\hat{z}_{si}^{t+1} \tag{11}$$

式(9)代替式(5)后，迭代算法具有很好的鲁棒性。

5 加快算法的收敛速度

似然函数式(3)中，z_{ki} 表示样本 x_i 属于第 k 个成分的概率，其取值满足条件 $\sum_{k=1}^{c}z_{ki}=1$。z_{ki} 越大，表明样本 x_i 属于成分 k 的可能性越高，在下一次迭代时有更多的机会属于这个成分，反之亦然。

类似于对 ω_k 的调节方法，用 $-\ln z_{ki}$ 表示样本 x_i 属于第 k 成分的信息量，则 $-\sum_{k=1}^{c} z_{ki}\ln(z_{ki})$ 为 x_i 属于各成分的平均信息量，当 $z_{ki}=1/c$（即 x_i 属于任何一个成分的概率相同）时，则认为信息量为 0。按照此观点，迭代时必须使信息量达到最大。于是利用信息熵最大化原理对 z_{ki} 施加惩罚算子，将 z_{ki} 的迭代公式变为：

$$z_{ki}^{t+1}=\frac{\omega_k^{t+1}f(x_i;\theta_k^{t+1})}{\sum_{s=1}^{c}\omega_s^{t+1}f(x_s;\theta_k^{t+1})}+\alpha_i z_{ki}^t\{\ln z_{ki}^t-\sum_{s=1}^{c}z_{si}^t\ln z_{si}^t\} \quad (12)$$

式中，α_i 为调节系数，考虑到相关的约束条件，其取值由下式计算：

$$\alpha_i=\min\{\frac{1-\max(z_{ki}^t)}{-\max(z_{ki}^{t-1})\sum_{k=1}^{c}z_{ki}^{t-1}\ln z_{ki}^{t-1}},\frac{(\sum_{k=1}^{c}e^{(-n|z_{ki}^t-z_{ki}^{t-1}|)})}{c}\} \quad (13)$$

从式(12)可以看出，$\sum_{s=1}^{c}z_{si}^t\ln z_{si}^t$ 是 $\ln z_{ki}^t$ 关于 $z_{1i}^t,z_{2i}^t,\cdots,z_{ci}^t$ 的加权平均，当 $\ln z_{ki}^t$ 小于 $\sum_{s=1}^{c}z_{si}^t\ln z_{si}^t$ 时，z_{ki}^{t+1} 将小于没有施加惩罚算子时的计算值，这会使得样本加速向某些成分靠拢，从而促进成分数 c 的减少。

求出 z_{ki}^{t+1} 后，利用式(3)调整 $\omega_k'^{t+1}$（注意不使用式(9)，目的是使两次计算相互独立），并再次消除不合理的成分。

当成分数趋于稳定时，设置调节系数 $\alpha_i=0$，不再对 z_{ki} 施加惩罚算子，以保持收敛的稳定性。

改进后算法的基本步骤如下：

Step1　初始化，需要做以下工作：

(1)设定成分个数 $c=n$，混合参数惩罚系数 $\beta=1$ 及样本所属成分概率惩罚系数 $\alpha=1$；

(2)初始化每个成分的协方差矩阵 Σ_k 和均值向量 $\mu_k=x_k$；

(3)初始化 z_{ki}。

Step2　利用式(9)替换式(5)来计算混合系数 ω_k，如果 $\omega_k<1/n$，则删除成分 k，同时执行式(10)、式(11)。

Step3　利用式(7)更新成分的协方差矩阵 Σ_k。

Step4　利用式(12)替换式(4)计算 z_{ki}。

Step5　利用式(5)计算混合系数 ω_k，如果 $\omega_k<1/n$，则消除成分 k，同时执行式(10)、式(11)。

Step6　利用式(13)调整 α_i。

Step7　利用式(6)计算均值向量 μ_k。

Step8　如果 $\max_{1\leqslant k\leqslant c}\|\mu_k^{t+1}-\mu_k^t\|<\in$，则退出循环，否则执行 Step2。

从以上步骤可以看出，新算法每次迭代时执行了两次成分的消除操作，分别是 Step2 和 Step5，这使得算法的迭代次数大大减少，提高了算法的收敛速度。

算法的时间复杂度分析：算法的计算复杂性与迭代次数有关，每次迭代的时间花费包括两部分：E-Step 和 M-Step。在 E-Step 中，计算 z_{ki} 需要花费的时间复杂度为 $O(nc)$；在 M-Step 中，计算 ω_k 和 μ_k 的时间复杂度都是 $O(c)$，计算协方差 Σ_k 的时间复杂度都是 $O(nd^2c)$。于是每次迭代为 $O(nc(d^2+1)+c)$这和原始 EM 算法及文献[1]中的算法在单次迭代的时间复杂度基本相同。而迭代次数与成分数 c 有关，加速 c 的减少有利于迭代次数的减少，本文中的方法通过分别对混合模型的成分系数及样本所属成分的概率施加熵惩罚，加快成分数的减少，从而大大减少了迭代次数。

6　实验分析

本文的实验针对 4 个算法进行了比较，算法 CA1 是对混合系数和样本所属成分的概率系数都没有施加惩罚算子的情况（即传统的 EM 聚类算法），算法 CA2 表示只对混合系数施加惩罚算子的情况（即文献[1]中的算法），算法 CA3 是同时对混合系数和样本所属成分的概率系数施加惩罚算子的情况（即本文中的算法），CA4 为文献[7]中的算法，初始成分数都设置为样本个数。比较以下几方面的内容：

(1)算法的迭代次数与成分数变化的趋势；

(2)算法的收敛性能，即算法结束后成分数收敛到正确解的概率；

(3)算法平均收敛时间；

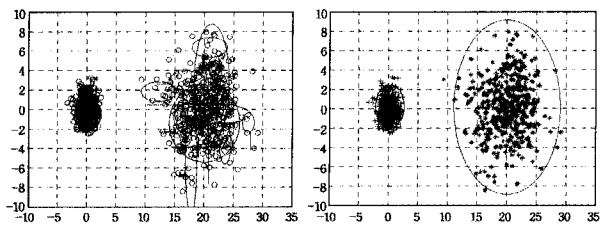
(4)成分数在得到正确解的情况下数据分类的正确率。

实验中首先选用了文献[1]中两组有代表性的样本数据进行比较；然后选用了文献[14]中的一组真实数据进行比较。

6.1　第 1 组比较

样本数 $n=800$，数据维数 $d=2$，样本随机来源于两个不同的高斯分布（产生概率相同），其中：$\mu_1=(0\quad 0)^T$，$\mu_2=(20\quad 0)^T$，$\Sigma_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$，$\Sigma_2=\begin{pmatrix}9&0\\0&9\end{pmatrix}$，比较的结果如下。

如图 2 所示，传统 EM 算法 CA1 进行聚类时收敛速度非常慢，这是由于该算法一直保持在比较高的成分数目，导致计算量很大，并且当成分数初始值设置为样本个数时收敛不到正确解，最终成分数目最好的情况是 158，如图 1(a)所示（图中的椭圆是按照 $\frac{(x_1-\mu_1)^2}{\sigma_1^2}+\frac{(x_2-\mu_2)^2}{\sigma_2^2}=1$ 的形式画出，椭圆的中心点在均值 μ 处，σ_1、σ_2 为协方差矩阵的对角元素，σ 和 μ 均由计算所得）。从图 1(b)可以看出，CA2 和 CA3 两个算法都可以收敛到正确解，得到各成分的均值和协方差基本一致。图 2 表明，CA2 和 CA3 在开始迭代初期，成分数的下降明显快于传统的算法，其中 CA2 迭代了 25 次左右得到了最终成分数 2，CA4 经过了 50 多次迭代，而 CA3 基本上经过 10 次左右的迭代就达到了相同的结果，总的迭代次数减少非常明显。最终 CA2、CA3 和 CA4 都能收敛到正确的结果，并且数据的分类完全正确。由于算法每次迭代的时间复杂度与成分数呈指数关系，因此，在迭代初期快速降低成分数可以大大加快算法的执行速度。



(a) CA1 聚类结果(不收敛)　　(b) CA2、CA3 和 CA4 正确的聚类结果

图 1　4 种算法的聚类结果图

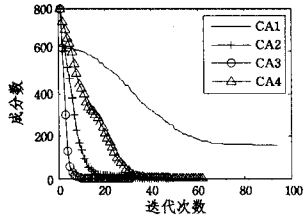


图 2　第 1 组数据的收敛趋势比较

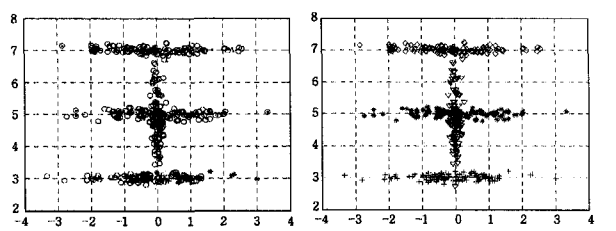
表 1 是随机执行 100 次的平均比较结果。

表 1　第 1 组数据执行 100 次的比较结果

算法	迭代次数	耗时(s)	成分数正确率	数据分类正确率
CA1	136	912.73	—	—
CA2	63	107.45	100%	100%
CA3	42	66.32	100%	100%
CA4	75	92.47	100%	100%

6.2　第 2 组比较

样本来源于 4 个不同的高斯分布，$n=400$，$d=2$，$\mu_1=(0\quad 3)^T$，$\mu_2=(0\quad 5)^T$，$\mu_3=(0\quad 7)^T$，$\mu_4=(0\quad 5)^T$，$\Sigma_1=\Sigma_2=\Sigma_3=\begin{pmatrix}1.2 & 0\\ 0 & 0.01\end{pmatrix}$，$\Sigma_4=\begin{pmatrix}0.01 & 0\\ 0 & 0.8\end{pmatrix}$，每组样本的产生概率相等，该组数据的特点是第 4 个成分和第 2 个成分成十字交叉。比较结果如下。



(a) CA1 聚类结果　　(b) CA2、CA3 和 CA4 正确的聚类结果

图 3　4 种算法的聚类结果

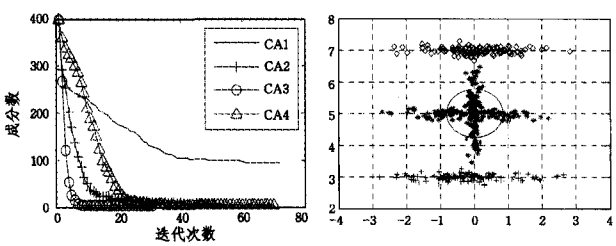


图 4　第 2 组数据的收敛趋势比较　　图 5　CA3 收敛到 $c=3$ 的情况

第 2 组数据中，CA1 最好的收敛结果是 $c=86$，CA2、CA3 和 CA4 基本能收敛到正确解 $c=4$，图 3(b)中的椭圆变成一条直线，是因为均值 μ 趋近于 0。图 4 表明，本文的算法在收敛速度上比 CA1、CA2、CA4 都有明显的改善。

表 2 是第 2 组数据随机执行 100 次的平均比较结果。其中 CA2 有 6 次收敛的成分数为 5；而 CA3 有 4 次收敛的成分数是 3(将成分 2 和成分 4 收敛到一起，见图 5)，3 次收敛的成分数为 5，CA4 的收敛速度比 CA2 和 CA3 要慢，同时成分正确率和数据分类正确率也相对低一些。CA2 和 CA3 数据分类的正确率都达到了 97%以上。

表 2　第 2 组数据执行 100 次的比较结果

算法	迭代次数	耗时(s)	成分数正确率	数据分类正确率
CA1	78	206.47	0%	—
CA2	65	48.56	94%	97.3%
CA3	41	20.63	93%	97.5%
CA4	72	65.46	86%	95.8%

6.3　真实数据集上的比较

本组比较使用了来自于文献[14]的真实数据集，包含了 145 个糖尿病人的观测数据，每个病人包含 3 个测量值，分别是血糖对口服葡萄糖的反应、血糖胰岛素水平对口服葡萄糖的反应以及胰岛素耐受性程度。临床诊断结果包括正常，化学性糖尿病及显性糖尿病 3 种情况。

随机执行 100 次后，比较结果如表 3 所列。

表 3　真实数据集的比较结果

算法	迭代次数	耗时(s)	成分数正确率	数据分类正确率
CA2	123	67.5	89%	87.5%
CA3	72	34.6	89%	87.2%
CA4	154	65.2	83%	82.6%

以上 3 组实验表明，本文中的算法和文献[1]中的算法在收敛性能和数据分类正确率等方面都很接近，大大优于传统的 EM 算法和文献[7]中的算法，而本文的算法所花费的时间明显要少许多，因此在具有很好的鲁棒性的同时还具有更快的收敛速度。

结束语　本文在文献[1]中提出的鲁棒性高斯模型聚类算法之上，提出了一个快速、鲁棒的聚类算法，其利用信息熵最大化原理，对混合成分系数及样本所属成分的概率系数同时施加惩罚因子，加快样本向各成分集中，从而进一步加快无效成分的消除，使得算法只需很少的迭代次数就能收敛到确定解，大大提高了算法执行的速度。算法对初始设置的成分数不敏感，因此具有很好的鲁棒性。

需要说明的是，高斯混合模型对噪声很敏感，因此本文实验没有测试带有噪声的样本数据。混合 t 模型[2,3]和混合 Pearson-VII 模型[4]属于重尾分布，可以通过自由度参数 v 调节噪声的干扰，本文中所采用的方法只需要做很小的改动，就可以移植到这两个模型中，从而实现噪声环境下的鲁棒性改善。

参 考 文 献

[1] Yang M-S, Lai C-Y, Lin C-Y. A robust EM clustering algorithm for Gaussian mixture models[J]. Pattern Recognition, 2012(5): 3950-3961

[2] Andrews J L, McNicholas P D, Subedi S. Model-based classification via mixtures of multivariate t-distributions[J]. Computational Statistics and Data Analysis, 2010(6): 520-529

[3] Peel D, McLachlan G J. Robust Mixture Modeling using the t Distribution[J]. Statistics and Computing, 2000(10): 339-348

[4] Sun Jian-yong, Garibaldi J M. Robust mixture clustering using Pearson type VII distribution[J]. Pattern Recognition Letters, 2010(7): 1-8

[5] 朱峰，宋余庆，陈健美. 基于椭球等高分布混合模型的聚类方法[J]. 江苏大学学报：自然科学版，2011(6)：701-705

[6] Bouguila N, ElGuebaly W. Discrete data clustering using finite mixture models[J]. Pattern Recognition, 2009(1): 33-42

[7] Figueiredo M A T, Jain A K. Unsupervised learning of finite Mixture models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002(24): 381-396

[8] Lin T I. Robust mixture modeling using multivaritate skew t distribution[J]. Statistics and Computing, 2010(20): 343-356

[9] 余成文,郭雷. 基于有限混合多变量 t 分布的鲁棒聚类算法[J]. 计算机科学, 2007(5): 190-193

[10] Bouguila N, Ziou D, Vaillancourt J. Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application[J]. IEEE Transactions on Image Processing, 2004 (11): 1533-1543

[11] Biernacki C, Celeux G, Govaert G. Choosing starting values for the EM algorithm for getting the highest likelihood inmultivariate Gaussian mixture models[J]. Computational Statistic&Data Analysis, 2003(41): 561-575

[12] Reddy K, Chiang H D, Rajaratnam B. TRUST-TECH-based expectation maximization for learning finite mixture models[J]. IEEE Transactionson Pattern Analysis and Machine Intelligence, 2008(30): 1146-1157

[13] Richardson P, Green J. On Bayesian analysis of mixtures with an unknown number of components[J]. Journal of the Royal Statistical Society-SeriesB, 1997(30): 731-758

[14] Reaven G M, Miller R G. An attempt to define the nature of chemical diabetes using a multidimensional analysis[J]. Diabetologia, 1979(16): 17-24

(上接第 150 页)

K_{ack1}和K_{ack2}，改进后的协议根据比特流版本号确定密钥K_{TAG}'，K_{TAG+1}'和K_{TAG+2}'，每轮比特流更新的密钥均不相同，这种一次一密的加密方式具有更强的安全性。

5.2 性能比较

由于改进后的协议在每轮比特流更新时均使用新的请求和确认密钥，密钥管理中对这 3 个密钥的定期更新就不再需要，仅需对 K_B 进行定期更新，与 Devic 协议相比，减轻了协议密钥管理的负担。

在多 FPGA 系统中，Devic 协议要求每个 FPGA 都存储 3 个请求和确认密钥，且各 FPGA 的密钥均不同，若 SD 管理 n 个 FPGA，则 SD 需要存储 $3n$ 个密钥。改进后的协议中 FPGA 只需要存储一个密钥链种子 K_0，SD 中也仅需保存与 n 个 FPGA 对应的 n 个 K_0。比特流的版本号和密钥 K_B 在两个协议中均须保存，不改变协议参与方的存储负担。因此，与 Devic 协议相比，改进后的协议减轻了协议参与方的存储负担。

表 1 给出了在具有 n 个 FPGA 的系统中，经过 m 次密钥定期更新后，Devic 协议与改进协议在密钥存储空间、密钥分发次数、密钥更新次数等方面的比较。

表 1 协议性能比较表

	Devic 协议	改进协议
SD 密钥存储空间	4n	2n
FPGA 密钥存储空间	4	2
初始化密钥分发次数	4	2
密钥更新次数	4m	m

结束语 本文首先分析了 Devic 提出的防重放攻击的远程比特流更新协议，发现 Devic 协议在密钥存储和管理方面效率不高，然后，基于双密钥链结构，对协议进行改进。改进协议在不改变原协议具有的机密性、完整性和防重放攻击特性的前提下，采用一次一密的方式加密 TAG，具有更好的安全性。此外，改进协议能提高协议密钥管理的效率，并降低协议参与各方的存储负担。

进一步的研究中，我们将考虑将群加密策略引入多 FPGA 系统的比特流更新协议，以期提高协议的运行效率。

参 考 文 献

[1] Lysaght P, Stockwood J. A simulation tool for dynamically reconfigurable field programmable gate arrays[J]. IEEE Transactions on Very Large Scale Integration(VLSI)Systems, 1996, 4(3): 381-390

[2] Upegui A, Peña-Reyes C A, Sanchez E. An FPGA platform for on-line topology exploration of spiking neural networks[J]. Microprocessors and microsystems, 2005, 29(5): 211-223

[3] Upegui A, Peña-Reyes C A, Sanchez E. A methodology for evolving spiking neural-network topologies on line using partial dynamic reconfiguration[C]// International Conference on Computational Intelligenc. Medellin, Colombia, 2003

[4] Devic F, Torres L, Badrignans B. Secure protocol implementation for remote bitstream update preventing replay attacks on FPGA[C]// 2010 International Conference on Field Programmable Logic and Applications(FPL). IEEE, 2010: 179-182

[5] Actel. ProASIC®3 Handbook. 2008[OL]. www. actel. com/documents/PA3_HB. pdf

[6] Badrignans B, Elbaz R, Torres L. Secure FPGA configuration technique preventing system downgrade[C]// Proceedings of the 18th International Conference on Field Programmable Logic and Applications(FPL'08). 2008

[7] Drimer S. Volatile FPGA design security-a survey [M]. IEEE Computer Society Annual Volume, 2008: 292-297

[8] Lamport L. Password authentication with insecure communication[J]. Communications of the ACM, 1981, 24(11): 770-772

[9] Cederquist J, Dashti M T, Mauw S. A certified email protocol using key chains[C]// Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on. IEEE, 2007: 525-530

[10] 李磊,谭新莲,王育民. 密钥链多方非否认协议[J]. 计算机科学, 2009, 36(010): 89-90