ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 38,No. 3,May 2023,pp. 525－538
DOI:10. 16337/j. 1004-9037. 2023. 03. 003

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

基于统计感知策略的高斯混合模型求解方法

陈佳琪[1]，何玉林[1,2]，黄哲学[1,2]，FOURNIER-VIGER Philippe[1]

(1. 深圳大学计算机与软件学院,深圳 518060;2. 人工智能与数字经济广东省实验室(深圳),深圳 518107)

摘　要：高斯混合模型(Gaussian mixture model,GMM)是一种经典的概率模型,常被用于无监督学习领域来确定无类别标记样本点的类别分布。作为求解GMM参数的重要技术,期望最大化(Expectation maximization,EM)算法通过计算GMM对应似然函数的最优解确定基模型自身参数以及基模型的混合系数。利用EM算法求解GMM存在如下两个缺陷:EM算法易于陷入局部最优解以及EM算法确定GMM基模型相关参数的不稳定,尤其是针对多维随机变量。本文提出了一种基于统计感知(Statistical-aware,SA)策略的GMM求解方法——SA-GMM方法。该方法从估计给定数据集的未知概率密度函数入手,建立了核密度估计(Kernel density estimation,KDE)与GMM之间的关联。为避免KDE对“过平滑”窗口的选取,设计了同时最小化KDE与GMM之间的经验风险和KDE窗口结构风险的目标函数,进而确定了GMM的最优参数。在11个标准概率分布上的实验证明了SA-GMM方法的可行性、合理性和有效性,同时结果也表明SA-GMM能够获得显著优于基于EM算法的GMM及其变体的概率密度函数估计表现。

关键词：高斯混合模型;概率密度函数估计;统计感知;经验风险;结构风险;粒子群优化

中图分类号：TN911.73　　**文献标志码**:A

Solution Method of Gaussian Mixture Model with Statistical-Aware Strategy

CHEN Jiaqi[1], HE Yulin[1,2], HUANG Zhexue[1,2], FOURNIER-VIGER Philippe[1]

(1. College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China;2. Guangdong Laboratory of Artificial Intelligence and Digital Economy (Shenzhen), Shenzhen 518107, China)

Abstract: Gaussian mixture model (GMM) is a classic probability model, which is usually used in the field of unsupervised learning to determine the class distribution of unlabeled samples. As an important method for solving GMM parameters, the expectation-maximization (EM) algorithm determines the parameters and component coefficients by calculating the optimal solution of the GMM likelihood function. The use of EM algorithm to solve GMM has the following two defects: EM algorithm is prone to getting stuck in a local optimal solution, and the relevant parameters of the GMM basic model determined by the EM algorithm are unstable, especially for high-dimensional data. For this reason, this paper proposes a GMM solution method based on statistical-aware (SA) strategy, i.e. SA-GMM method. Starting from the estimation of the unknown probability density function of a given data set, the method establishes the correlation between kernel density estimation (KDE) technology and GMM. To avoid the selection of

基金项目:国家自然科学基金面上项目(61972261);广东省自然科学基金面上项目(2023A1515011667);深圳市基础研究重点项目(JCYJ20220818100205012);深圳市基础研究面上项目(JCYJ20210324093609026)。
收稿日期:2022-06-30;**修订日期**:2022-12-11

KDE's over-smoothing bandwidth, the goal is to simultaneously minimize the empirical risk between KDE and GMM and the structural risk of KDE's bandwidth. The experiments on 11 standard probability distributions confirm the feasibility, rationality, and effectiveness of SA-GMM. And it is also shown that the proposed SA-GMM method can obtain the better performance on probability density function estimation than EM-based GMM and its variant.

Key words: Gaussian mixture model; probability density function estimation; statistical aware; empirical risk; structural risk; particle swarm optimization

引　言

高斯混合模型(Gaussian mixture model,GMM)是一种结构简单但却应用广泛的概率模型,具有较强的对未知数据分布的表达能力,在数学建模问题上有着很好的灵活性和适应性。GMM通过多个高斯分布或者正态分布的加权和来拟合数据集的真实概率分布。中心极限定理表明,在样本量足够大的情况下,独立同分布随机变量的均值近似服从于高斯分布,且高斯分布在已知均值及方差的连续分布中的信息熵最大,因此GMM通常选取高斯分布作为混合模型的主要构件进行数据建模。GMM在混合模型学习算法中训练速度相对较快,且GMM属于生成模型[1],可用于对任意随机变量概率分布进行拟合。此外,GMM能够容易地扩展到无监督学习邻域,这在无监督学习越来越重要的今天是一个很大的优势。当前,对GMM的研究主要集中在应用和求解两方面:GMM被应用于动/静态图像分割以及特征提取、语音分割和识别、密度估计、目标跟踪与识别、视觉信息处理等,以及寻求更加高效精确的GMM求解策略完成对未知分布数据集概率分布的拟合。

GMM应用的代表性工作:1995年,Reynolds等[2]对高斯混合模型作了详细介绍,并基于此模型实现了不依赖于文本的语音识别,并在后续的工作中运用基于GMM的方法在更大的公共语音数据库上进行评估,取得了不错的效果[3];2000年,Reynolds等[4]对基于GMM的语言识别系统进行描述,该系统由麻省理工大学林肯实验室研发并应用于多个NIST语言识别评估中取得良好表现,为语音识别从实验阶段走向应用阶段做出了重要贡献;2006年,Campbell等[5]研究了在支持向量机(Support vector machine,SVM)分类器中使用GMM超向量的思想,提出基于GMM模型间距离度量的SVM核。除了语音识别,GMM在计算机视觉领域也有很好的应用。2004年,Zivkovic[6]基于高斯混合模型研发了一种有效的用于背景提取的自适应算法,该方法实现了高斯混合模型中组件个数的自适应,通过递归不断更新参数,为每个像素确定适当数量的分量个数;2005年,Lee[7]提出一种有效的基于GMM的视频背景提取方法,在不影响模型稳定性的前提下提高收敛速度;2010年,Jian等[8]提出了一个统一的基于概率模型进行点云配准的框架,该框架的关键思想是使用高斯混合模型表示输入点集。除此之外,高斯混合模型在其他领域也有应用。例如,2010年,He等[9]介绍了一种基于流形结构的正则化概率模型进行数据聚类,该模型称为拉普拉斯正则化高斯混合模型;2015年,乔少杰等[10]提出了基于高斯混合模型来对轨道进行预测的方法;2022年,An等[11]提出了一种用于网络攻击异常检测的无监督集成自编码高斯混合模型。

GMM求解的代表性工作:在GMM的应用中,模型参数的确定非常关键。极大似然估计(Maximum likelihood estimation,MLE)方法是一种常用的模型参数估计方法,其基本思想是利用已知的样本信息来估计最可能出现样本结果的模型参数。直接利用MLE方法求解GMM无法求得解析解,目前常用的数值计算方法是期望最大化(Expectation-maximization,EM)算法,由Dempster等在1977年提出[12],解决含有隐变量(数据缺失)情况下的参数估计问题。除了EM算法外,其他常见的算法有智能优化算法[13]、信息熵方法[14]、ICE算法[15]、蒙特卡-洛马尔科夫算法[16]、基于正定矩阵黎曼几何的EM替

代方案[17]以及黎曼牛顿信任域方法[18]。

EM算法在许多应用中都有良好的性能,因为其算法简单,且能有效找到最优收敛值而常作为混合模型参数求解方法。作为一种迭代优化策略,EM算法每一轮迭代由两个步骤组成:第1步是计算隐变量的期望(称为期望步,E步);第2步使用第1步的值求似然函数最大时的参数(称为极大步,M步),然后进入下一次迭代。M步得到的参数被用于下次迭代的E步计算中。但是EM算法对参数的初值敏感,初值的选择往往会影响收敛的效率以及是否易陷入局部最优。有许多文献对于EM算法存在的问题进行解决,例如,期望条件最大化(Expectation conditional maximization, ECM)算法[19]、确定性退火EM算法[20]、随机EM(Stochastic expectation maximization, SEM)算法[21]等。这些算法都是对EM算法的改进,没有脱离原来算法的框架。另外EM算法的稳定性不高,特别是在高维度情况下需要求解的参数个数很多,例如在D维下高斯混合模型的组件数为K,一共需要求解的参数个数为$K\times\left(1+D+D^2\right)$,极易导致对GMM参数确定的不稳定,即便是相同个数条件下的基模型参数差异性都很大。

为了解决EM算法求解GMM模型的上述两个缺陷,即EM算法易于陷入局部最优解以及EM算法确定GMM基模型相关参数的不稳定,尤其是针对多维随机变量,本文提出了一种基于统计感知策略的GMM(Statistical-aware based GMM,SA-GMM)求解方法。该方法试图通过最小化核密度估计(Kernel density estimation,KDE)确定的概率密度与GMM确定的概率密度之间的误差确定GMM的最优参数;为了避免KDE对“过平滑”窗口参数的选取,SA-GMM在最小化KDE与GMM误差,即经验风险最小化的同时还对KDE的窗口结构风险进行了最小化。在11个标准概率分布上对SA-GMM的性能进行了验证,实验结果表明SA-GMM能够获得显著优于基于EM算法的GMM(EM based GMM,EM-GMM)及其变体的概率密度函数估计表现,从而证实了SA-GMM的可行性、合理性和有效性。

1 EM-GMM的原理及分析

在基于EM算法的高斯混合模型EM-GMM中,EM算法通过每次迭代中的E步和M步来收敛到最优值,可以对包含隐变量或缺失数据的概率模型进行参数估计。EM算法的两个步骤为:

(1) E步(期望步):通过当前参数Θ^t来推断样本对应的隐变量的概率分布,即后验概率,然后计算对数似然$LL\left(\Theta\middle|X,Z\right)$关于后验概率的期望,公式表示为

$$Q\left(\Theta\middle|\Theta^t\right)=E_{Z|X,\Theta^t}LL\left(\Theta|X,Z\right)=\sum\nolimits_Z P\left(Z\middle|X,\Theta^t\right)\ln P\left(X,Z|\Theta\right) \tag{1}$$

式中:X表示已观测变量集;Z表示隐变量集;Θ表示模型参数;Θ^t表示第t次迭代的模型参数。

(2) M步(极大步):求使得Q函数最大化时的参数Θ^{t+1},即

$$\Theta^{t+1}=\arg\max_{\Theta} Q\left(\Theta\middle|\Theta^t\right) \tag{2}$$

EM算法的流程为:先初始化参数Θ^0;然后进行E步,通过当前参数Θ^t计算对数似然关于后验概率的期望;再进行M步,求得使得期望最大化时的参数Θ^{t+1}。如果当前满足收敛条件,则表明收敛到最优解,结束循环。否则M步得到的参数值被重新用于E步中,继续循环。EM算法的流程图如图1所示。

下面,通过两个具体的例子来对EM算法存在的缺陷进行直观地展示,进而引出本文设计的基于统计感知的高斯混合模型求解策略。

(1) EM算法易陷入局部最优解。该缺陷已经被众多学者在相关的研究工作中分析过,在此不再赘述,仅通过如下仿真实验展示。生成一组由2个一维高斯分量组成的GMM仿真数据,其中一个高斯分布的权重、均值、方差分别为$w_1=0.3,\mu_1=0,\sigma_1^2=1$;另一个

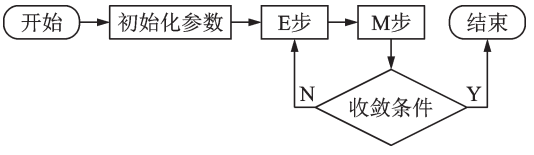


图1 EM算法流程图

Fig.1 Flow chart of EM algorithm

高斯分布的权重、均值、方差分别为$w_2=0.7$,$\mu_2=4$,$\sigma_2^2=1$,设置样本点个数$N=200$。随机设置EM算法的初始值,在仿真数据上求解GMM参数,重复实验50次。将这50次实验求得的参数进行可视化,第1个高斯构件的参数作为横坐标、第2个构件的参数为纵坐标,分别对权重、均值、方差作图,结果如图2所示。可以看到随机初始化的EM算法容易收敛到局部最优值附近,没有收敛到实际最优值。本文涉及到的仿真数据均可以通过公开链接下载:https://pan.baidu.com/s/1dimbklkCxZ57ZL0-LFYm-wQ,提取码:psc4。

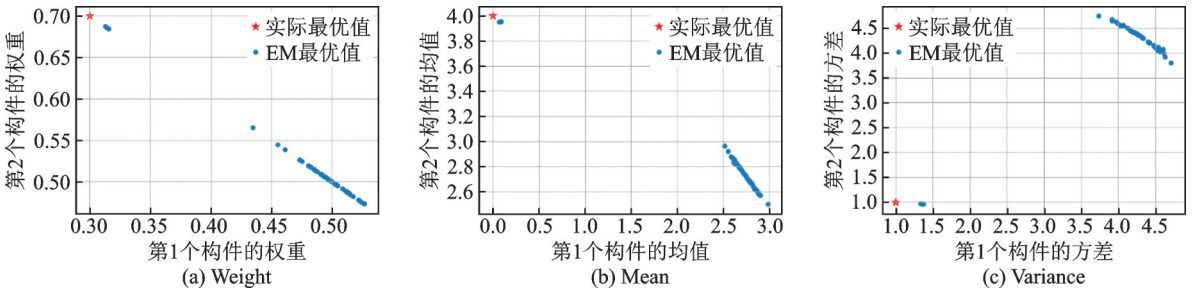


图2 EM算法的局部最优性

Fig.2 Local optimality of EM algorithm

(2) EM算法对GMM参数确定的不稳定。EM算法得到的结果受到参数初始值影响,容易导致参数确定的不稳定。这里在仿真的一维和二维数据集上验证不同初始值选取对GMM求解的影响。首先通过K-means算法和随机选取来设置初始值,之后利用EM算法来确定GMM的最优参数,最后得到如图3和图4所示的概率密度函数(Probability density function, PDF)估计结果。从图中可以观察到不同的初始值选取对于最终的GMM求解影响非常大,有针对性的初始值选取效果要好于随机的初始值选取,但同时带来了计算复杂度的显著增加。

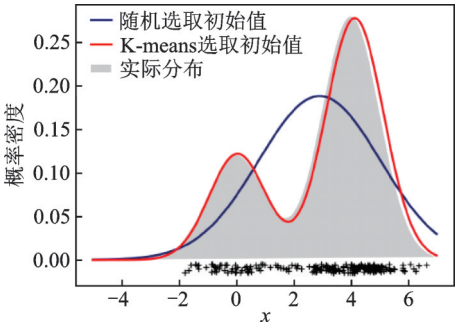


图3 一维下不同初始值下的EM算法

Fig.3 EM algorithm with different initial values for one-dimensional data

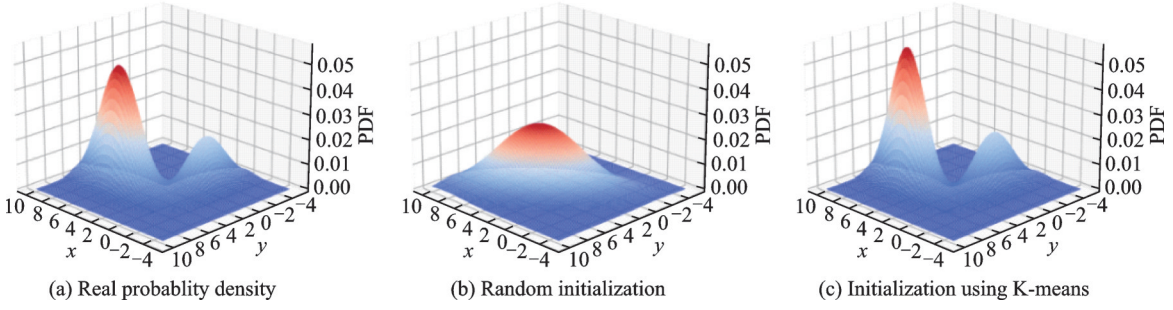


图4 二维下不同初始值下的EM算法

Fig.4 EM algorithm with different initial values for two-dimensional data

2 本文方法

针对上述EM-GMM算法的缺点,本文提出一种基于统计感知策略的GMM求解方法:SA-GMM方法。该方法将KDE策略引入到对GMM的求解中,通过构建KDE与GMM之间的关联进而确定

GMM的最优参数。KDE属于非参数统计方法,通常用来估计未知的PDF。SA-GMM算法以同时最小化KDE与GMM之间的经验风险和KDE窗口结构风险为目标,利用粒子群优化算法来求解目标参数,进而确定GMM最优参数。

2.1 KDE与GMM的等效性分析

为了描述的简便性,从一维数据集上进行讨论,多维数据集可看作是一维数据的简单推广。KDE是将每个样本点以及窗口参数作为核函数的参数,得到的N个核函数线性叠加就得到了KDE的概率密度函数。对于核函数为高斯核的KDE,得到的概率密度函数可以表示为

$$f_h(x)=\frac{1}{Nh}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{x-x_i}{h}\right)^2\right] \tag{3}$$

式中:N为样本的个数;h为核密度估计的窗口宽度参数;x_i为观测样本数据。GMM可以看成是多个高斯分布的加权叠加,对于由GMM表示的概率密度函数可以表示为

$$f(x)=\sum_{i=1}^{K}\frac{w_i}{\sqrt{2\pi}\ \sigma_i}\exp\left[-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2\right] \tag{4}$$

式中:K为高斯构件个数;w_i、μ_i、σ_i为每个高斯分布的混合权重、均值以及标准差。

对比式(3,4)可以观察到KDE和GMM在结构上的相似性。对于位于区间$[a,b]$内的样本数据,$a-b=\varepsilon$满足$\varepsilon\to 0$,取$N=K$,$\sigma=h$,$w=\frac{1}{N}$,$\mu=x_i$,此时$f_h(x)=f(x)$,即KDE与GMM间存在等效性。随机生成1组有三簇的仿真数据,分别通过GMM和KDE来拟合数据分布,通过得到的概率密度曲线来分析GMM与KDE间的联系,结果如图5所示。可以观测到KDE和GMM都得到3个峰值,都可以反映出样本有三簇的分布。

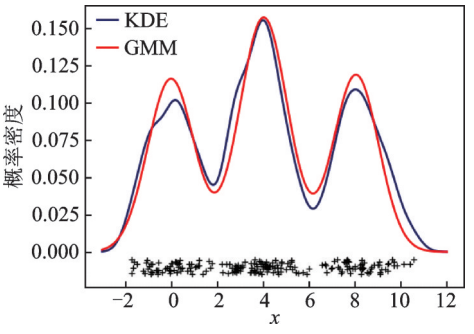


图5 KDE与GMM得到的概率密度函数

Fig.5 PDF estimated with KDE and GMM

2.2 SA-GMM目标函数的设计

通过核密度估计得到概率密度函数$\hat{f}(x)$,对于待估计的GMM概率密度函数$f(x)$,目标是让GMM密度函数$f(x)$去逼近KDE密度函数$\hat{f}(x)$,使得这两个概率密度更加接近。最优的GMM参数应使得GMM和KDE概率密度函数之间的误差最小化,即

$$\Theta=\arg\min\sum_{i=1}^{N}\left[f(x_i|\Theta)-\hat{f}(x_i)\right]^2 \tag{5}$$

KDE得到的概率密度函数受带宽h影响,对于h的取值可以通过交叉验证来确定。但是大部分情况下,交叉验证得到的带宽h_{best}会偏大,如图6所示。此时核密度估计得到的概率密度曲线相对平缓,得到的GMM概率密度曲线与真实概率密度曲线相比,峰值处的拟合效果不够好。

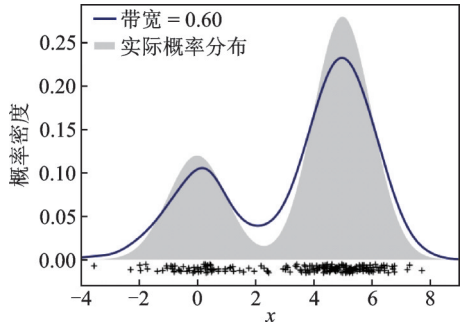


图6 KDE在交叉验证得到的带宽下估计的概率密度函数

Fig.6 Estimated PDF of KDE with the cross validation-based bandwidth

基于图6的仿真数据,给出了GMM与KDE概率密度函数之间的误差随带宽h的变化趋势,如图7所示。h在

区间$[h_{\text{best}}-0.2, h_{\text{best}}+0.2]$内变化，GMM概率密度估计通过EM算法求得。固定GMM概率密度，改变带宽h，使得KDE得到的概率密度发生改变，GMM与KDE间的误差也会发生改变。从图7可以看到，随着h的增大，GMM与KDE概率密度函数之间的误差先减小后增加，在最低点处的h对应最小误差，交叉验证得到的带宽比误差最小处的带宽大。

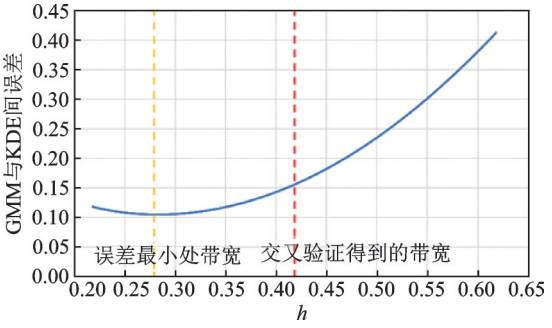


图7 GMM与KDE概率密度函数之间误差随h的变化趋势

Fig.7 Estimated error between GMM and KDE with different bandwidths

为了更好地拟合峰值处，使得KDE得到的概率密度更接近于真实的概率密度，对于h的取值，可以取比交叉验证得到的带宽偏小的值，所以将对h的约束纳入目标函数的设计中，即给目标函数添加一个惩罚项λh^2，则调整后的目标函数为

$$\Theta=\arg\min\left[\sum_{i=1}^{N}\left(f(x_i|\Theta)-\hat{f}(x_i)\right)^2+\lambda h^2\right]=\arg\min\left[\sum_{i=1}^{N}\left(\sum_{j=1}^{K}w_jN(x_i|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)-\hat{f}(x_i|h)\right)^2+\lambda h^2\right] \tag{6}$$

式中：K为GMM构件数；x_i表示第i个观测数据；w_j为第j个高斯分布组件的权重，满足$w_j\geqslant 0$，$\sum_{j=1}^{K}w_j=1$；$\boldsymbol{\mu}_j$、$\boldsymbol{\Sigma}_j$为第j个高斯分布的均值以及协方差。需要估计的参数Θ为w_j、μ_j、$\sigma_j^2\left(1\leqslant j\leqslant K\right)$以及$h$，目标是求得满足目标函数值最小的参数，使得取得的带宽偏小的同时GMM得到的概率密度更接近于核密度估计得到的概率密度，即GMM与KDE概率密度函数间误差更小。λ是大于0的系数，使得h在优化的过程中不会因为比前一项$\left[f(x)-\hat{f}(x)\right]^2$值小太多而被忽略，同时也可以调整目标函数取得最小值时的带宽大小。同样基于图6的仿真数据，固定GMM概率密度，观察随着带宽h变化，不同目标函数值的变化趋势以及在不同目标函数下取得极值时h的变化，结果如图8所示。从图8中可以看到随着h的增大，目标函数值呈现先减后增的变化趋势，对于不添加惩罚项的目标函数，取得极值时带宽比交叉验证得到的带宽小。加上惩罚项后，随着系数λ的增大，目标函数取得极值时的带宽呈减小趋势。由此可以得出结论，加入惩罚项可以调整带宽的大小，使得在取得较小带宽的同时，GMM与KDE的概率密度函数之间误差也尽可能地小。在求解最优参数的过程中，若h过小，核密度估计得到的概率密度函数$\hat{f}(x)$会更加曲折，波动较多，所以GMM的概率密度函数会更难拟合$\hat{f}(x)$，$\left[f(x)-\hat{f}(x)\right]^2$会增大，所以$h$的减小趋势会被约束，不会取到太小的值。

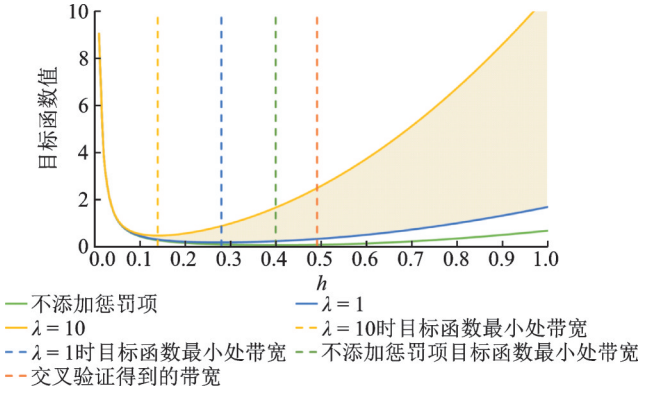


图8 随着带宽h增加不同目标函数值的变化趋势

Fig.8 Variation tendencies of different objective function values with the increase of bandwidth h

2.3 SA-GMM最优参数的求解

为了得到满足目标函数的参数，可以通过粒子群优化算法来求解。粒子群优化是一种进化计算技术，可以用来求解最优化问题。该算法的优势在于不需要调节过多的参数，算法容易实现。其基本思想是利用种

群中粒子的相互合作,将个体的信息进行共享来找到最优解。首先设置种群大小,对一群随机粒子进行初始化。每一次迭代中粒子通过个体信息(局部最优值)以及共享信息(全局最优值)来更新自己,不断通过迭代往全局最优靠近,最终找到最优解。

2.3.1 种群设置

在D维下高斯混合模型的组件个数为K,一共需要求解的参数个数为$K\times\left(1+D+D^2\right)+D$,包括每个高斯分布的混合权重$w$、均值$\mu$、协方差$\boldsymbol{\Sigma}$以及核密度估计带宽$h$。协方差矩阵是对称矩阵,所以可以只保存上三角或者下三角的数据,在需要计算的时候再利用保存的数据进行还原。参数个数为$K\times\left(1+D+\frac{D(D+1)}{2}\right)+D$。种群规模一般设置为20~40,对于较复杂的问题可以设置为100~200。这里设置种群规模为30。每个粒子$\left(w,\mu,\boldsymbol{\Sigma},h\right)$有速度和位置这两个属性,对于不同参数,由于搜索范围不同,可以设置不同的速度。

2.3.2 粒子初始化

在对粒子进行初始化。混合权重w可以都设置为$1/K$,速度设置为$[-0.1,0.1]$区间内随机数。对于均值μ、协方差$\boldsymbol{\Sigma}$中的元素,可以随机初始化,但是为了防止初始化得到的值过小或者过大,使得收敛比较慢,可以给定一个与样本相关的值。计算样本数据的均值以及协方差,取均值向量中最大元素mean以及协方差均值中最大元素cov。向量元素设置为$[-\text{mean},\text{mean}]$区间内的随机值,速度设置为$[-0.1\times\text{mean},0.1\times\text{mean}]$区间内随机数。协方差矩阵中对角线元素都是非负的,所以为了简便,将元素都设置为$[0,\text{cov}]$区间内的随机值,速度则在$[-0.1\times\text{cov},0.1\times\text{cov}]$内随机选取。这里需要保证协方差矩阵是半正定矩阵。带宽h取区间$(0,h_{\text{best}}]$中随机数。

2.3.3 迭代过程

在迭代过程中,对粒子的两个属性进行更新,如果有粒子找到了更优解,则更新局部最优以及全局最优。同时需要限制粒子的速度以及位置,防止粒子因速度太小而陷入局部搜索,或因速度过大而跳过最优值,还要维护协方差矩阵的半正定性质。

对粒子的位置以及速度进行更新。每个粒子根据自身信息(局部最优值)以及其他粒子共享的信息(全局最优解)来更新自己的速度以及位置,使得整个种群不断向全局最优位置移动。当一个粒子找到更优的值时,先更新个体信息,然后向整个种群共享,更新全局最优解。在更新速度后,需要限制粒子的速度,可以以粒子速度初始化时的区间作为粒子速度区间进行限制。防止速度过大而越过目标函数的极小值,以及速度过小导致的一直在局部区域进行搜索。更新位置后,需要限制粒子的位置,保证解的合理性。权重w必须为大于0的数,如果小于0则w设为$[0,1]$上的随机数。协方差$\boldsymbol{\Sigma}$的对角线上元素不能为负数,如果为负数则设为0。

更新完每个粒子的速度和位置后,根据适应值函数计算每个粒子的适应值。如果当前得到的适应值小于局部最优值的适应值,则更新局部最优值。如果适应值小于全局最优值的适应值,则更新全局最优值。每个粒子的局部最优值和种群共享的全局最优值更新完后,判断是否达到迭代次数或全局最优值的适应度是否达到设定的阈值。若不满足则继续进行迭代,不断更新粒子,搜索最优值。如果满足则跳出循环,得到的全局最优值即为所求参数。

协方差矩阵具有半正定性质。在迭代的过程中,要注意维护协方差矩阵的对称半正定性质。因为$\boldsymbol{\Sigma}$只保存了上三角的数据,所以对称性可以得到保证。在使用$\boldsymbol{\Sigma}$计算概率密度函数时,恢复其完整矩阵后需要保证$\boldsymbol{\Sigma}$是半正定的。为了判断得到的矩阵是否为半正定矩阵,可以使用Cholesky分解。如果矩阵可以被分解为一个下三角矩阵$\boldsymbol{L}$及其转置$\boldsymbol{L}^{\mathrm{T}}$的乘积,则该矩阵为半正定矩阵。在迭代过程中,粒子移动后协方差可能为非半正定矩阵,所以需要对矩阵进行调整,恢复其半正定性质。Higham[22]于1988

年提出计算最近的对称半正定矩阵的方法，若得到的矩阵不是半正定的，则计算其“最近”的对称半正定矩阵替换原有的矩阵来作为协方差。基本流程为对于矩阵$\boldsymbol{\Sigma}$，先对其进行奇异值分解(Singular value decomposition, SVD)，即

$$\boldsymbol{\Sigma}=\boldsymbol{USV}^{\mathrm{T}} \tag{7}$$

然后定义一个矩阵$\boldsymbol{H}$为

$$\boldsymbol{H}=\boldsymbol{VSV}^{\mathrm{T}} \tag{8}$$

矩阵$\boldsymbol{\Sigma}$则可以被替换为

$$\boldsymbol{\Sigma}=\left(\boldsymbol{\Sigma}+\boldsymbol{\Sigma}^{\mathrm{T}}+\boldsymbol{H}+\boldsymbol{H}^{\mathrm{T}}\right)/4 \tag{9}$$

除了计算最近的半正定矩阵来替代原来的协方差矩阵，还可以通过协方差参数化的方法[23]来保证迭代过程中矩阵的对称半正定性质。

由于混合权重w_i满足$w_i>0$且$\sum_{i=1}^{K}w_i=1$，在迭代过程只需要限制$w_i>0$，最后得到全局最优解时，将权重w_i进行归一化处理，即

$$w_i=\frac{w_i}{\sum_{i=1}^{K}w_i} \tag{10}$$

则可以保证$\sum_{i=1}^{K}w_i=1$。

3 实验验证与分析

实验包括对SA-GMM算法的可行性、合理性和有效性进行验证。为了方便结果的可视化展示，在一维和二维数据集的基础上进行实验。需要设置的参数有惯性因子ω，学习因子c_1、c_2，迭代次数t，种群规模，目标函数的权重λ，具体如下：

(1) 惯性因子ω：设置$\omega_{\max}=0.5$，$\omega_{\min}=0.1$。通过线性微分递减策略来调整惯性因子。

(2) 学习因子c_1、c_2：设置为常数，$c_1=c_2=2$。

(3) 迭代次数t：设为200。

(4) 种群规模：种群规模一般设置为20～40，对于较复杂的问题可以设置为100～200。这里设置种群规模为30。

(5) 权重λ：λ可以取满足$1\leqslant\lambda h_{\mathrm{best}}{}^{2}\leqslant5$的值。

3.1 SA-GMM可行性验证

第1组仿真数据的分布由两个一维高斯分量组成的GMM，其概率密度函数为

$$f\left(x|\Theta\right)=\sum_{j=1}^{K}w_jN\left(x\middle|\mu_j,\ \sigma_j^2\right) \tag{11}$$

式中：一个高斯分量的参数为$w_1=0.3$，$\mu_1=0$，$\sigma_1^2=1$；另一个高斯分量参数为$w_2=0.7$，$\mu_2=7$，$\sigma_1^2=1$。设置样本点个数$N=200$，生成的仿真数据的分布以及真实概率密度曲线如图9所示。第2组仿真数据的分布由两个二维高斯分量组成的GMM，其概率密度函数为

$$f\left(x|\Theta\right)=\sum_{j=1}^{K}w_jN\left(x\middle|\boldsymbol{\mu}_j,\ \boldsymbol{\Sigma}_j\right) \tag{12}$$

式中：一个高斯分量的参数为$w_1=0.3$，$\boldsymbol{\mu}_1=[0,0]$，$\boldsymbol{\Sigma}_1=\begin{bmatrix}1&0\\0&5\end{bmatrix}$；另一个高斯分量参数为$w_2=0.7$，$\boldsymbol{\mu}_2=$

$[5,5]$,$\boldsymbol{\Sigma}_2=\begin{bmatrix}5 & 0\\ 0 & 1\end{bmatrix}$。设置样本点个数$N=500$。随机生成的仿真数据的分布如图10所示。

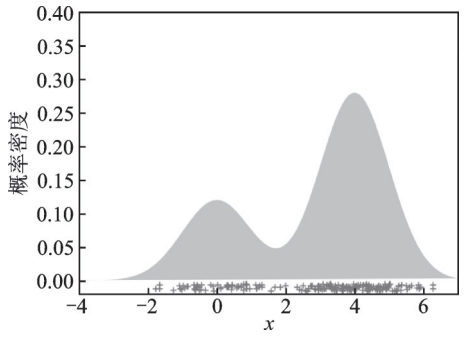


图9 一维下的仿真GMM数据

Fig.9 One-dimensional GMM simulation data

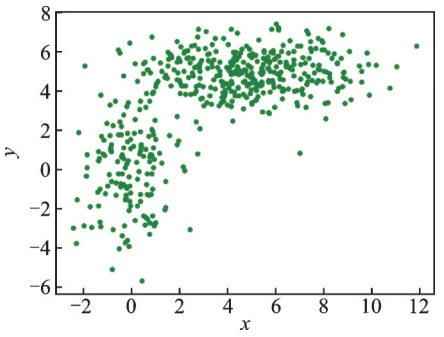


图10 二维下的仿真GMM数据

Fig.10 Two-dimensional GMM simulation data

生成仿真数据后,通过本文提出的方法SA-GMM来估计仿真数据集的概率密度函数,对得到的概率密度曲线进行可视化,与真实数据集的概率密度曲线进行比较来验证数据的可行性。通过SA-GMM估计上述随机仿真GMM数据的概率密度函数,得到的概率密度曲线以及真实概率密度曲线如图11(N=200个样本点)和图12所示。可以看到一维和二维的情况下,通过本文方法估计得到的概率密度函数接近于实际数据的概率密度函数。通过上述两个实验的结果图可以观察到本文提出的方法得到的概率密度曲线接近于真实的概率密度曲线,所以本文方法可以实现GMM参数的估计。由此可以得出结论:本文提出的SA-GMM对于GMM参数估计具有可行性。

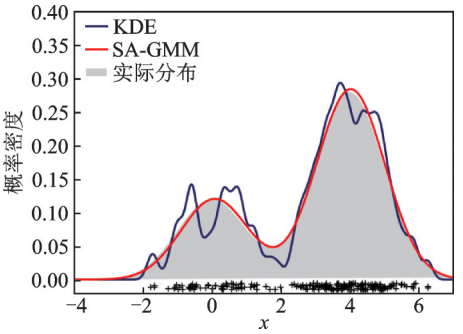


图11 SA-GMM估计的一维概率密度函数

Fig.11 Estimated PDF with SA-GMM for one-dimensional data

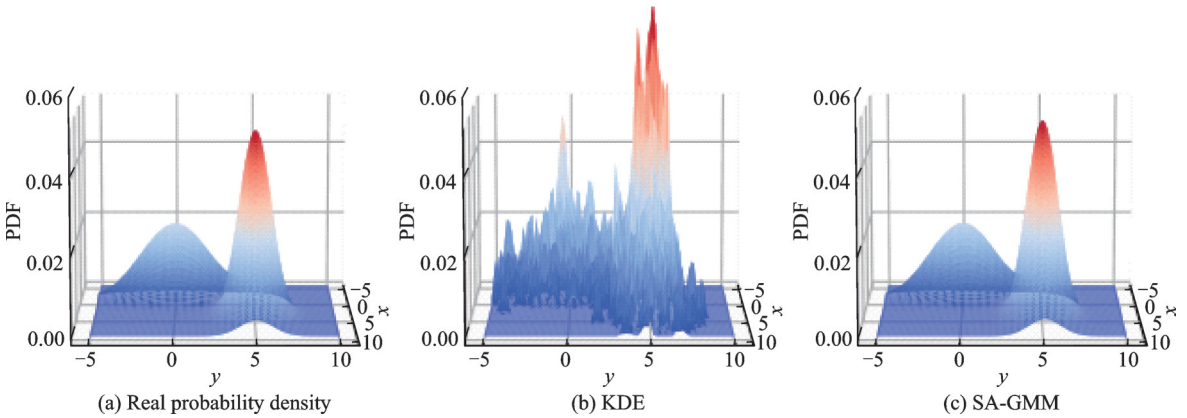


图12 SA-GMM估计的二维概率密度函数

Fig.12 Estimated PDF with SA-GMM for two-dimensional data

3.2 SA-GMM合理性验证

在SA-GMM算法合理性验证过程中,给出了随着迭代次数的增加,目标函数值(即粒子群优化过

程中的适应值)的变化,结果如图13、14所示。可以观测到随着迭代次数的增加,目标函数值先降低然后趋于平稳,呈收敛趋势。所以SA-GMM算法可以收敛到最优值,由此证实了算法的合理性。

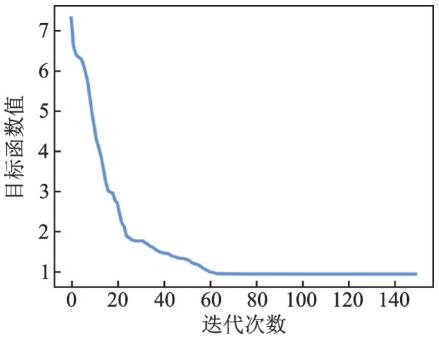


图13 一维数据集上SA-GMM的收敛性验证

Fig.13 Convergence of SA-GMM on one-dimensional GMM simulation data

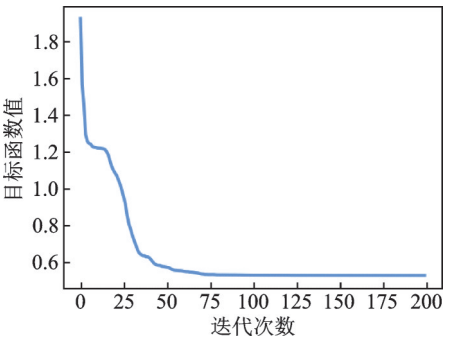


图14 二维数据集上SA-GMM的收敛性验证

Fig.14 Convergence of SA-GMM on two-dimensional GMM simulation data

3.3 SA-GMM有效性验证

在一维下生成8组服从不同分布的仿真数据。仿真数据分别服从均匀分布、指数分布、瑞利分布、伽马分布、卡方分布、F分布、Beta分布和t分布。设置数据集大小$N=200$,随机生成的数据实际分布如图15所示。在二维下随机生成两簇均匀分布在圆内的点。设置数据集大小$N=500$,实验随机生成的仿真数据集如图16所示,用于二维下的有效性验证实验。对于每组数据,通过EM-GMM、基于遗传算法期望最大化的高斯混合模型(Genetic algorithm expectation-maximization based Gaussian mixture model,GA-EM-GMM)[24]以及本文提出的SA-GMM来估计数据集的概率密度函数,与真实数据的概率密度进行可视化的比较分析。为了从数值上说明本文方法的有效性,避免算法结果的偶然性,进行了另一个数值实验。将不同分布的数据集随机生成10次,分别用EM-GMM、GA-EM-GMM和KDE-GMM来估计概率密度,采用均方误差(Mean square error, MSE)来计算估计的概率密度与实际概率密度的误差,基于这10次数据集的平均结果可以评价估计性能。MSE公式如下

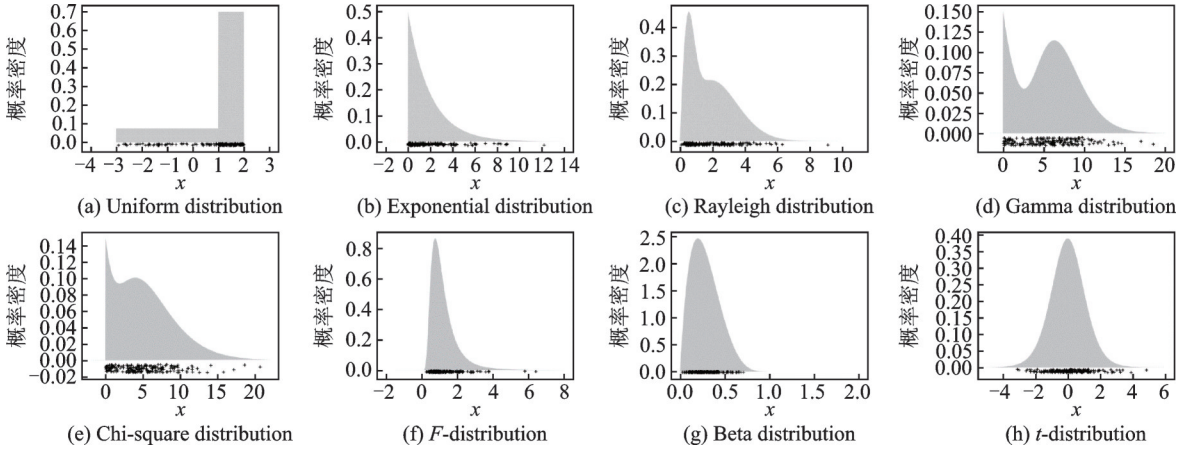


图15 用于验证SA-GMM有效性的一维仿真数据

Fig.15 One-dimensional simulation data for effectiveness validation of SA-GMM

$$\mathrm{MSE}=\frac{1}{N}\sum_{i=1}^{N}\left[f(x_i)-\hat{f}(x_i)\right]^2 \tag{13}$$

在一维下,对于不同分布的随机数据集,通过EM-GMM、GA-EM-GMM和SA-GMM得到的概率密度曲线如图17所示,与真实概率密度曲线作比较。从实验结果可以观察到:在一维情况下,对于服从均匀分布、指数分布、瑞利分布、卡方分布、伽马分布、F分布、Beta分布和t分布的数据,通过SA-GMM得到的概率密度曲线比EM-GMM以及GA-EM-GMM得到的概率密度曲线更加接近于真实数据,本算法相比于另外两个算法表现得更好,由此可以验证一维情况下本算法的有效性。虽然核密度估计得到的概率密度曲线很曲折,但是能够较好地拟合数据的整体分布,特别在峰值处有更好的体现。

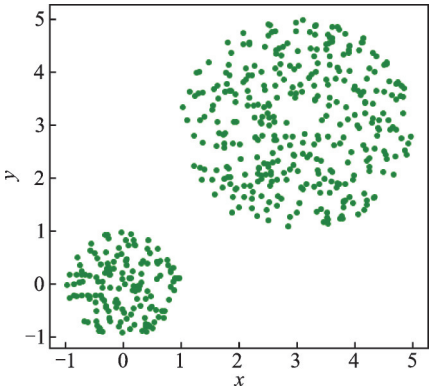


图16 用于验证SA-GMM有效性的二维仿真数据

Fig.16 Two-dimensional simulation data for effectiveness validation of SA-GMM

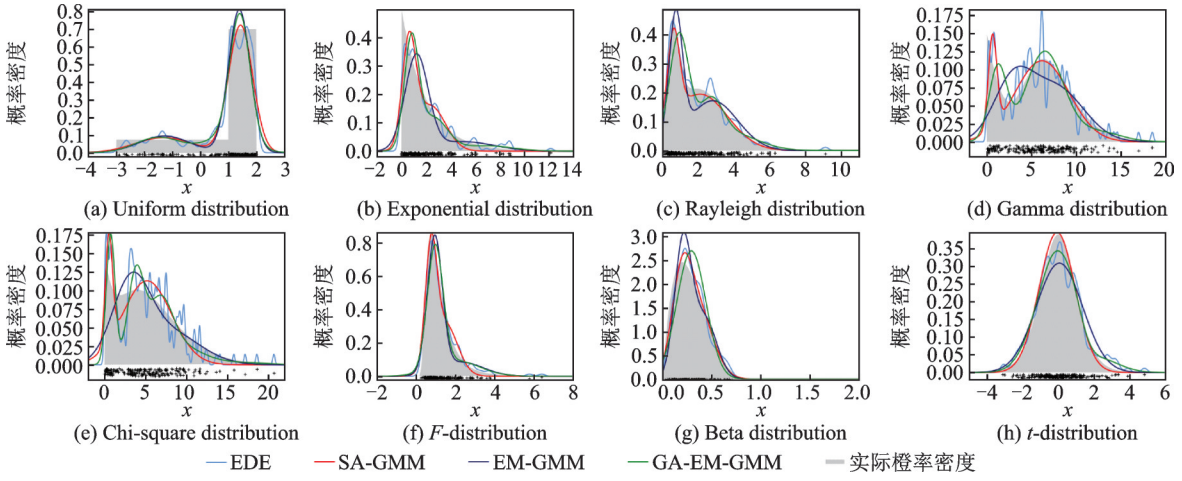


图17 一维仿真数据上EM-GMM、GA-EM-GMM和SA-GMM的概率密度函数估计表现

Fig.17 PDF estimation performances of EM-GMM, GA-EM-GMM and SA-GMM on one-dimensional simulation data

将上述实验进行10次,对于不同分布,用EM-GMM、GA-EM-GMM和SA-GMM来估计概率密度函数,计算得到的概率密度与真实数据概率密度间的MSE,结果如表1以及图18所示。从表中数据以

表1 一维数据集上EM-GMM、GA-EM-GMM和SA-GMM估计的概率密度函数对应的MSE

Table 1 MSE of PDF estimated with EM-GMM, GA-EM-GMM and SA-GMM on one-dimensional simulation data

分布	EM-GMM	GA-EM-GMM	SA-GMM
指数分布	0.002 700 7	0.002 267 7	0.001 872 4
F分布	0.004 461 0	0.004 334 0	0.001 701 6
均匀分布	0.009 253 0	0.008 928 5	0.007 512 7
瑞利分布	0.001 927 9	0.001 779 5	0.000 689 7
伽马分布	0.000 544 2	0.000 486 3	0.000 329 2
卡方分布	0.000 386 1	0.000 376 4	0.000 273 2
Beta分布	0.038 548 5	0.038 548 5	0.018 908 9
t分布	0.000 249 9	0.000 163 4	0.000 117 0

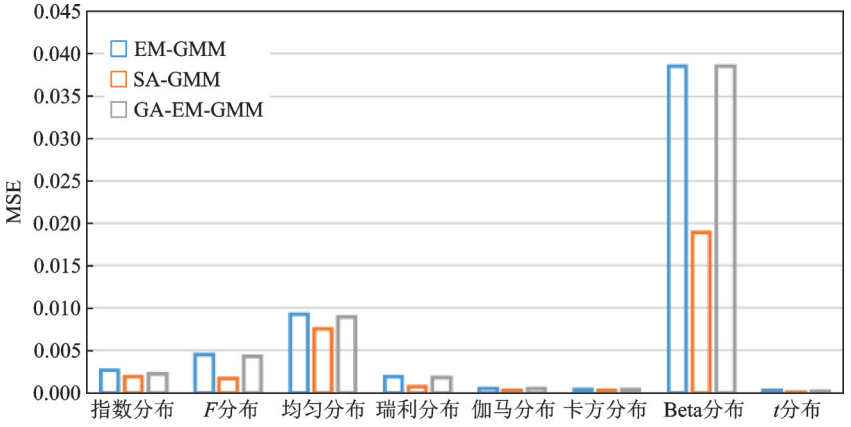


图 18 表 1 实验结果的图形化展示

Fig.18 Graphical illustration of experimental results in Table 1

及柱状图可以总结到:在一维下,对于不同分布,EM-GMM 以及 GA-EM-GMM 得到的 MSE 大于 SA-GMM 得到的 MSE,说明 EM 算法以及基于遗传算法的 EM 算法得到的概率密度与真实概率密度误差比较大,本算法相较于这两个算法有更好的表现,从而可以从数值上证明本算法的有效性。

在二维随机均匀分布数据下进行实验,实验结果如图 19 所示。可以观察到 SA-GMM 在峰值上拟合更好,相比于 EM-GMM 于 GA-EM-GMM,SA-GMM 得到的概率密度曲线更接近于真实数据的分布,下面进行二维下的数值实验。对于 10 次重复实验,记录的 MSE 如表 2 以及图 20 所示。可以看出在

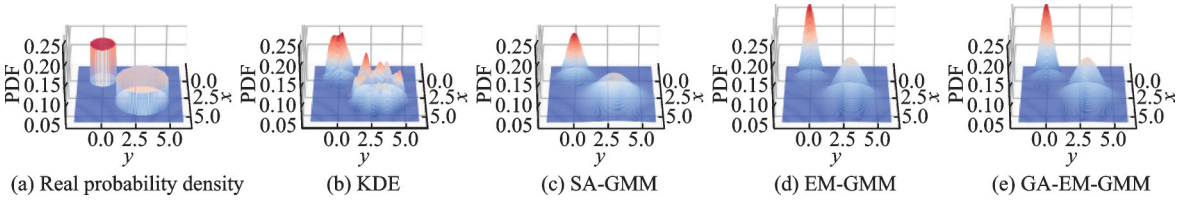


图 19 二维仿真数据上 EM-GMM、GA-EM-GMM 和 SA-GMM 的概率密度函数估计表现

Fig.19 PDF estimation performances of EM-GMM, GA-EM-GMM and SA-GMM on two-dimensional simulation data

表 2 二维数据集上 EM-GMM、GA-EM-GMM 和 SA-GMM 估计的概率密度函数对应的 MSE

Table 2 MSE of PDF estimated with EM-GMM, GA-EM-GMM and SA-GMM on two-dimensional simulation data

次数	EM-GMM	GA-EM-GMM	SA-GMM
1	0.000 290 3	0.000 289 3	0.000 212 4
2	0.000 268 2	0.000 268 2	0.000 216 8
3	0.000 280 7	0.000 280 6	0.000 223 1
4	0.000 281 9	0.000 281 2	0.000 214 1
5	0.000 291 0	0.000 242 3	0.000 212 8
6	0.000 270 9	0.000 270 9	0.000 216 9
7	0.000 300 3	0.000 300 3	0.000 212 7
8	0.000 295 1	0.000 236 7	0.000 210 1
9	0.000 269 5	0.000 269 5	0.000 223 9
10	0.000 292 5	0.000 292 5	0.000 209 5
平均值	0.000 284 0	0.000 273 1	0.000 215 2

二维数据下,SA-GMM得到的MSE更小,得到的概率密度曲线更接近于真实数据的概率密度曲线,从而可以验证在二维仿真数据下本算法的有效性。

通过上述实验结果图以及表中数据可以从直观上以及数值上观察到,相比于EM-GMM以及GA-EM-GMM,本文提出的SA-GMM得到的概率密度曲线更接近于真实数据的概率密度曲线,即SA-GMM比EM-GMM和GA-EM-GMM在一定条件下有更好的表现,由此可以验证本算法的有效性。

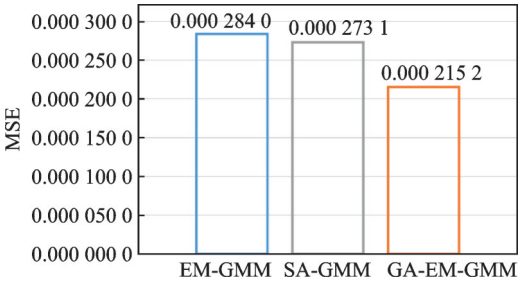


图20 表2实验结果的图形化展示

Fig.20 Graphical illustration of experimental results in Table 2

4 结束语

本文提出了一种基于统计感知的高斯混合模型求解方法SA-GMM。首先,将KDE纳入目标函数中,建立GMM与KDE间的关联。然后,通过粒子群优化算法来求解目标函数的最优值,使得GMM概率密度函数$f(x)$在接近$\hat{f}(x)$的同时也接近于真实数据的概率密度,即最小化GMM与KDE间的经验风险。并且将带宽h的值纳入目标函数的构建中,在最小化KDE窗口结构风险的同时使得GMM与KDE之间的误差更小。最后,通过设计的实验对本文提出的SA-GMM方法的可行性、合理性以及有效性进行了验证。然而,本文方法还存在一些问题需要改善,在今后可以对以下工作展开进一步的研究。(1) 高斯混合模型中高斯构件数K的精准确认。在低维度情况下通常可以通过目测来划分簇从而确定K值,但是随着维度的增大,这种人为给定K值的方法显然不适用,所以在未来的工作中计划实现对K值的精准和自动化确认。(2) 对目标函数中参数λ值选取的探究:参数λ的选取会影响到h的取值,在后续的工作中可以对λ值的选取进行进一步探究,用以选取更为准确的KDE窗口宽度。(3) 对SA-GMM算法的实际应用:尽管本文已经对算法的可行性、合理性以及有效性进行了验证,但是还没有将其应用到具体的实际应用中,在后续工作中可以考虑利用本文方法来解决真实场景中的概率密度函数估计问题。

参考文献:

[1] NG A Y, JORDAN M I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes[C]//Proceedings of Advances in Neural Information Processing Systems. [S.l.]: The MIT Press, 2002: 841-848.

[2] REYNOLDS D A, ROSE R C. Robust text-independent speaker identification using Gaussian mixture speaker models[J]. IEEE Transactions on Speech and Audio Processing, 1995, 3(1): 72-83.

[3] REYNOLDS D A. Speaker identification and verification using Gaussian mixture speaker models[J]. Speech Communication, 1995, 17(1/2): 91-108.

[4] REYNOLDS D A, QUATIERI T F, DUNN R B. Speaker verification using adapted Gaussian mixture models[J]. Digital Signal Processing, 2000, 10(1/2/3): 19-41.

[5] CAMPBELL W M, STURIM D E, REYNOLDS D A. Support vector machines using GMM supervectors for speaker verification[J]. IEEE Signal Processing Letters, 2006, 13(5): 308-311.

[6] ZIVKOVIC Z. Improved adaptive Gaussian mixture model for background subtraction[C]//Proceedings of the 17th International Conference on Pattern Recognition. [S.l.]: IEEE, 2004: 28-31.

[7] LEE D S. Effective Gaussian mixture learning for video background subtraction[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 827-832.

[8] JIAN B, VEMURI B C. Robust point set registration using Gaussian mixture models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 33(8): 1633-1645.

[9] HE X, CAI D, SHAO Y, et al. Laplacian regularized Gaussian mixture model for data clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 23(9): 1406-1418.

[10] 乔少杰, 金琨, 韩楠, 等. 一种基于高斯混合模型的轨迹预测算法[J]. 软件学报, 2015(5): 1048-1063.

QIAO Shaojie, JIN Kun, HAN Nan, et al. A trajectory prediction algorithm based on Gaussian mixture model[J]. Journal of Software, 2015(5): 1048-1063.

[11] AN P, WANG Z, ZHANG C. Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection[J]. Information Processing & Management, 2022, 59(2): 102844.

[12] DEMPSTER A P, LAIRD N M, RUBIN D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1977, 39(1): 1-22.

[13] ACI M, INAN C, AVCI M. A hybrid classification method of *k* nearest neighbor, bayesian methods and genetic algorithm[J]. Expert Systems with Applications, 2010, 37(7): 5061-5067.

[14] 马继涌,高文.基于最大交叉熵估计高斯混合模型参数的方法[J].软件学报,1999(9): 974-978.
MA Jiyong, GAO Wen. Method of estimating Gaussian mixture model parameters based on maximum cross entropy[J]. Journal of Software, 1999(9): 974-978.

[15] IMAGE G. Statistical image segmentation[J]. Machine Graphics and Vision, 1992, 1(1/2): 261-268.

[16] MARJORAM P, MOLITOR J, PLAGNOL V, et al. Markov chain Monte Carlo without likelihoods[J]. Proceedings of the National Academy of Sciences, 2003, 100(26): 15324-15328.

[17] HOSSEINI R, SRA S. An alternative to EM for Gaussian mixture models: Batch and stochastic Riemannian optimization[J]. Mathematical Programming, 2020, 181(1): 187-223.

[18] SEMBACH L, BURGARD J P, SCHULZ V. A Riemannian Newton trust-region method for fitting Gaussian mixture models [J]. Statistics and Computing, 2022, 32(1): 1-20.

[19] MENG X L, RUBIN D B. Maximum likelihood estimation via the ECM algorithm: A general framework[J]. Biometrika, 1993, 80(2): 267-278.

[20] FIGUEIREDO M A T, JAIN A K. Unsupervised learning of finite mixture models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(3): 381-396.

[21] MENG X L, RUBIN D B. Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm[J]. Journal of the American Statistical Association, 1991, 86(416): 899-909.

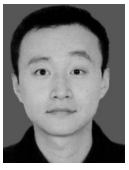
[22] HIGHAM N J. Computing a nearest symmetric positive semidefinite matrix[J]. Linear Algebra and Its Applications, 1988, 103: 103-118.

[23] KIRANYAZ S, INCE T, YILDIRIM A, et al. Fractional particle swarm optimization in multidimensional search space[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2009, 40(2): 298-319.

[24] PERNKOPF F, BOUCHAFFRA D. Genetic-based EM algorithm for learning Gaussian mixture models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(8): 1344-1348.

作者简介:

陈佳琪(1999-),女,硕士研究生,研究方向:数据挖掘、机器学习,E-mail: chenjq0725@qq.com。



何玉林(1982-),**通信作者**,男,博士,副研究员,研究方向:数据挖掘、机器学习、大数据系统计算技术,E-mail:yulinhe@gml.ac.cn。

黄哲学(1959-),男,博士,特聘教授,研究方向:大数据系统计算技术,E-mail: zx.huang@szu.edu.cn。

FOURNIER-VIGER Philippe(1980-),男,博士,特聘教授,研究方向:数据挖掘、人工智能、知识表示和推理、认知模型建构等,E-mail:philfv@szu.edu.cn。

(编辑:刘彦东)