计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2014.20130718
51(10):2277-2294,2014

MMCKDE:基于数据流的 m-混合聚类核概率密度估计

许 敏[1,2] 邓赵红[1] 王士同[1] 史荧中[1,2]
[1](江南大学数字媒体学院 江苏无锡 214122)
[2](无锡职业技术学院物联网技术学院 江苏无锡 214121)
(xum@wxit.edu.cn)

MMCKDE: m-Mixed Clustering Kernel Density Estimation over Data Streams

Xu Min[1,2], Deng Zhaohong[1], Wang Shitong[1], and Shi Yingzhong[1,2]
[1](*School of Digital Media, Jiangnan University, Wuxi, Jiangsu* 214122)
[2](*School of Internet of Things Technology, Wuxi Institute of Technology, Wuxi, Jiangsu* 214121)

Abstract In many data stream mining applications, traditional density estimation methods such as kernel density estimation and reduced set density estimation can not apply to the data stream density estimation because of their high computational burden and big storage space. In order to reduce the time and space complexities, a novel online data stream density estimation method by m-mixed clustering kernel is proposed. In the proposed method, MMCKDE nodes are created using a fixed number of mixed clustering kernels to get cluster information instead of all kernels obtained from other density estimation method. In order to further reduce the storage space, MMCKDE nodes can be merged by calculating KL divergence. Finally, the probability density functions over arbitrary time or the entire time can be estimated by the obtained model. We compared the MMCKDE algorithm with the SOMKE algorithm in terms of density estimation accuracy and running time for various stationary data sets. We also investigated the use of MMCKDE over evolving data streams. The experimental results illustrate the effectiveness and efficiency of the proposed method.

Key words m-mixed clustering kernel; kernel density estimation; probability density functions; Kullback Leibler (KL) divergence; streaming data mining

摘 要 数据流挖掘应用对时间、空间有着较高的要求,因而传统的密度估计方法,如核密度估计法、压缩集密度估计法等并不适用于数据流密度估计.提出一种新颖的面向在线数据流的 m-混合聚类核密度估计(m-mixed clustering kernel density estimation, MMCKDE)方法,该方法通过创建 MMCKDE 节点,用固定个数的混合聚类核获得聚类信息,以代替其他密度估计方法中的所有核.针对数据量不断增加的情况,通过计算 Kullback Leibler(KL)距离进行核合并,可进一步以更紧凑的形式表示概率密度估计信息.较之于其他一些方法只能估计整段数据流的密度,MMCKDE 方法最终获得的模型不仅适用于整段数据流,还适用于任意时间段上的密度估计.MMCKDE 算法同 SOMKE 算法在不同基准数据集及真实数据集上进行密度估计精度和运行时间的比较.实验结果表明,MMCKDE 算法具有更好的性能.

关键词 m-混合聚类核;核密度估计;概率密度函数;Kullback Leibler 距离;流数据挖掘

中图法分类号 TP391.4; TP18

收稿日期:2013-05-22;**修回日期**:2013-12-03

基金项目:国家自然科学基金项目(61272210,61170122);江苏省研究生创新工程项目(CXZZ12-0759)

随着网络技术的迅猛发展,在科技和商业应用中出现了以数据流形式存在的大量信息,如计算机网络流[1]、金融数据流[2]、环境监测数据流[3]和日志分析数据流等.这些数据的特点是产生速度快、数据量大、潜在无穷且数据分布随着时间的变化而改变.以近年来研究广泛、可应用于地理信息监测、生命信号检测和车辆运动跟踪等的传感器网络(sensor network)为例,在传感器网络中,大量传感器每时每刻不断感应出大量监测数据,并通过无线网络传输到基站,基站中的数据分析服务器必须及时地对这些数据进行分析和处理.几乎所有领域的研究者都面临怎样高速、有效地从数据流中获取有用信息的问题.文献[4]列出了一系列数据流算法设计准则:

1) 整个数据流只能遍历一次,没有时间再去访问旧数据;

2) 处理数据的时间必须简短,否则跟不上数据流生成的速度;

3) 整个数据处理过程所占的存储空间是有限的;

4) 算法应考虑到数据流的演化.

因真实数据概率密度分布不可知,故非参数核密度估计法(kernel density estimation, KDE)[5]是采用较广泛的密度估计法,估计出概率密度函数后,还可进一步应用于贝叶斯分类、异常点检测[1]和非参数回归等.但是 KDE 需要所有样本参与计算且需存储所有数据,故不适用于大规模数据流.压缩集概率密度估计[6]和快速压缩集概率密度估计法[7]被提出用于解决大规模数据集问题,该方法虽可对原数据集进行压缩,但压缩比率具有不确定性,很难满足整个数据流处理过程所占存储空间有限条件.上述传统的概率密度估计法均不适用于数据流概率密度估计.

基于此,本文针对数据流产生速度快、数据量大的特点,提出一种在线、实时且占用存储空间固定的概率密度估计法.为了达到此目的,本文首先提出 m-混合聚类核概率密度估计法(m-mixed cluster kernel density estimation, MMCKDE),该方法生成固定个数的 m-混合聚类核近似代替 KDE 中的所有核,在保证表示数据集信息完整的前提下,能大幅度节约存储空间.如图 1 所示,传统 KDE 存储空间与样本规模呈线性关系,本文提出的算法将样本点聚为 m 类,用 m 个混合聚类核近似表示 KDE 的所有核.图 1 假设 $m=5$.

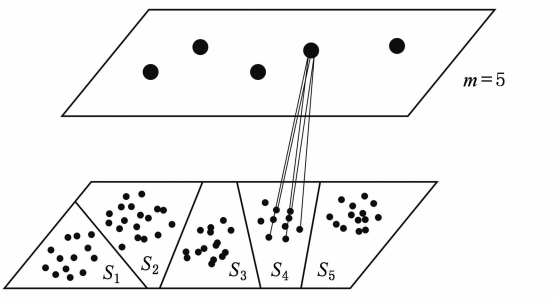


Fig. 1 An example of approximating original kernels with m kernels.

图 1 m 个混合核代替所有核示意图

下面将 MMCKDE 算法应用于数据流密度估计.如图 2 所示,将数据流按时间分为一个个时间窗,用 W_i 表示第 i 个时间窗,随着时间的推移,数据产生绵延不绝,将每个时间窗中的数据集聚类为 m 个混合聚类核,形成一个 MMCKDE 节点,用 SL_i 表示第 i 个节点,虽然可节约存储空间,但存储空间的增长与 m 呈线性关系,若数据流较大,仍不能满足存储空间有限的要求.

为了解决上述问题,本文进一步提出建立 MMCKDE 节点序列且与 Kullback Leibler(KL)距离相结合的方式,进一步进行核的合并,如图 3 所示.若 MMCKDE 节点小于规定的节点个数 n,则插入新节点;若 MMCKDE 节点超过规定的节点个数 n,则搜

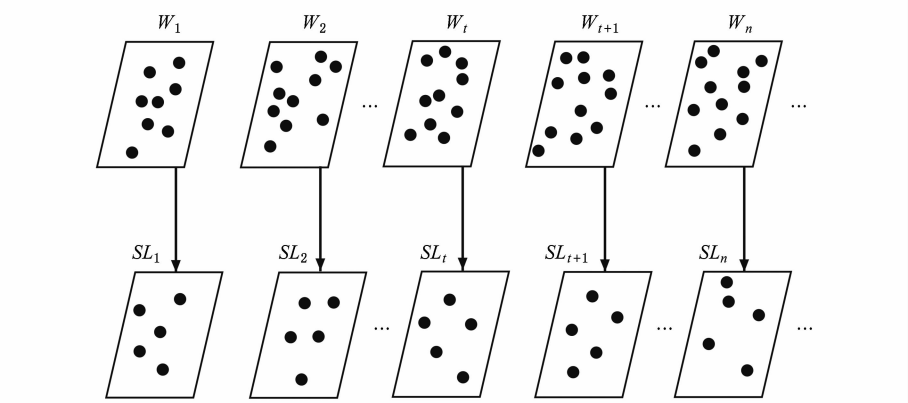


Fig. 2 An example of directly applying MMCKDE to a sequence of data windows.

图 2 MMCKDE 算法应用于数据流示意图

索 KL 距离最近的 2 个 MMCKDE 节点进行核合并,确保节点数仍为 n,达到存储空间固定的目的.此方式的另一优势在于,不同节点代表不同时间段的数据流密度估计信息,可对任意时间段数据流进行概率密度估计.图 3 假设 $n=4$,若产生第 5 个时间窗数据流,则搜索 KL 距离最近的 2 个节点,如节点 2 与节点 3,将这 2 个节点合并,故总节点数还是 4.

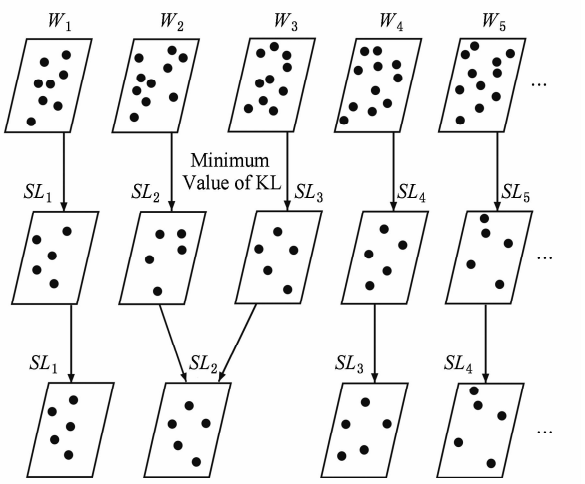


Fig. 3　An example of applying MMCKDE based on KL divergence to a sequence of data windows.

图 3　基于 KL 距离的 MMCKDE 算法应用于数据流示意图

1　m-混合聚类核概率密度估计算法

因真实样本概率密度分布不可知,故非参数核密度估计法应用广泛.但 KDE 法需要所有样本参与概率密度估计,时间空间复杂度高,故不适用数据流.一些快速算法如 RSDE(FRSDE)等虽能达到数据浓缩的目的,但其并不能固定样本浓缩比例且浓缩后的样本数仍随着数据窗的增加而增大,同样不能达到数据流概率密度估计要求存储空间有限的条件.本节下面将重点介绍使用固定个数 m 的混合聚类核近似表示 KDE 所有核的概率密度估计法.

设数据集 $X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_l\}$ 的核密度估计函数式为[8]

$$\hat{f}(\boldsymbol{x})=\frac{1}{lh^d}\sum_{i=1}^{l}K\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right)=\frac{1}{lh^d}\sum_{i=1}^{l}K_h(\boldsymbol{x}-\boldsymbol{x}_i),\tag{1}$$

其中,K 为核函数,h 为核宽,d 为数据集维数,l 为数据集规模.通常,核函数的选择有很多,且需满足式(1)(2):

$$K(\boldsymbol{u})\geqslant 0;\tag{2}$$

$$\int_{\mathbf{R}}K(\boldsymbol{u})\mathrm{d}\boldsymbol{u}=1.\tag{3}$$

本文算法采用最普遍的高斯核函数,定义如下:

$$K(\boldsymbol{u})=\frac{1}{(\sqrt{2\pi})^d}\exp\left(-\frac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}\right).\tag{4}$$

文献[9]指出核函数的选择对密度估计的影响远小于核宽,过小的核宽会导致密度估计的抖动幅度较大,过大的核宽会导致密度估计过于平滑,致使密度估计分辨率降低.可通过交叉验证等方法获得最优的核宽[10-11].一些文献也提供了简单而有效的核宽选择策略[8],例如 1 维数据集核宽选择:

$$h=1.06\hat{\sigma}l^{-\frac{1}{5}},\tag{5}$$

其中,$\hat{\sigma}$ 为输入数据集的标准差,l 为数据集规模.

对于式(1)核密度估计式,我们的工作是找到固定个数如 m 的混合聚类核,令 $g(\boldsymbol{x})=\sum_{j=1}^{m}\beta_jG_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)$,使 $g(\boldsymbol{x})$ 与 $f(\boldsymbol{x})$ 逼近.其中,$m<<n$,K,G 均为高斯核函数.

本文提出分块近似聚类法获得混合聚类核,使用一个新的聚类核代替原样本集中性质相近的若干核,即将原有的 n 个原始核划分成 m 个子块 $\{S_1,S_2,\cdots,S_m\}$,每个子块用一个混合聚类核来近似估计原始核中相近的 k 个核的总和.如图 1 所示,对第 j 个子类,使用混合核 $\beta_jG_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)$ 表示原始核中 $i\in S_j$ 的 k 个核的总和 $\sum_{i\in S_j}\alpha_iK_{h_i}(\boldsymbol{x},\boldsymbol{x}_i)$.

本文采用较常用的 L_2 准则来度量两者间的误差,以使误差达到最小,如式(6)所示:

$$\varepsilon=\int_{\mathbf{R}^d}(g(\boldsymbol{x})-f(\boldsymbol{x}))^2\mathrm{d}\boldsymbol{x}=\int_{\mathbf{R}^d}\left(\sum_{j=1}^{m}\beta_jG_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)-\sum_{i=1}^{n}\alpha_iK_{h_i}(\boldsymbol{x},\boldsymbol{x}_i)\right)^2\mathrm{d}\boldsymbol{x}=\int_{\mathbf{R}^d}\left(\sum_{j=1}^{m}\left(\beta_jG_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)-\sum_{i\in S_j}\alpha_iK_{h_i}(\boldsymbol{x},\boldsymbol{x}_i)\right)\right)^2\mathrm{d}\boldsymbol{x}.\tag{6}$$

通过使用柯西不等式 $\left(\sum_{i=1}^{m}a_i\right)^2\leqslant m\sum_{i=1}^{m}a_i$,可得:

$$\varepsilon\leqslant m\sum_{i=1}^{m}\int_{\mathbf{R}^d}\left(\beta_jG_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)-\sum_{i\in S_j}\alpha_iK_{h_i}(\boldsymbol{x},\boldsymbol{x}_i)\right)^2\mathrm{d}\boldsymbol{x}=\bar{\varepsilon}.\tag{7}$$

根据优化理论,对于式(6)的最优解可以通过求解其上界的界来逼近.进一步观察式(7)可知其表示聚类后的总误差的上界,由每个核聚类误差上界之和组成,即 $\bar{\varepsilon}=\bar{\varepsilon}_1+\cdots+\bar{\varepsilon}_m$.又由于各组成部分的解变量相互独立,因而优化上界可以转化为优化其各

组成部分,即我们的目标是使每个组成部分的误差 $\bar{\varepsilon}_j$ 达到最小:

$$\bar{\varepsilon}_j = \min_{\beta_j, G_{\sigma_j}(\cdot)} \int_{\mathbf{R}^d} \Big(\beta_j G_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j) - \sum_{i\in S_j}\alpha_i K_{h_i}(\boldsymbol{x},\boldsymbol{x}_i)\Big)^2 \mathrm{d}\boldsymbol{x}. \tag{8}$$

在此优化问题中,我们需求解的问题有系数 β_j、高斯核宽 σ_j、聚类点 $\boldsymbol{t}_j$,因聚类时不要求各个核相等,并将核宽也作为判断聚类核的依据,且聚类后的核宽不要求相同,故本文称之为混合聚类核概率密度估计法. 使用公式 $\int_{\mathbf{R}^d} k_{\sigma_1}(\boldsymbol{x},\boldsymbol{t}_1)k_{\sigma_2}(\boldsymbol{x},\boldsymbol{t}_2)\mathrm{d}x = k_{\sigma_1+\sigma_2}(\boldsymbol{t}_1-\boldsymbol{t}_2)$[6-7] 可对式(8)进行化简,附录中列出了式(8)的化简过程,经化简后可表示为

$$\bar{\varepsilon}_j = \frac{\beta_j^2}{(\sqrt{2\pi})^d(2\sigma_j^2)^{\frac{d}{2}}} - 2\sum_{i\in S_j}\frac{\beta_j\alpha_i}{(\sqrt{2\pi})^d(h_i^2+\sigma_j^2)^{\frac{d}{2}}}\times \exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2}{(h_i^2+\sigma_j^2)}\Big)\Big). \tag{9}$$

将 β_j 看作变量,σ_j,$\boldsymbol{t}_j$ 看作常量,$\frac{\partial\bar{\varepsilon}_j^{\min}}{\partial\beta_j}=0$,得出:

$$\beta_j = \sum_{i\in S_j}\frac{(2\sigma_j^2)^{\frac{d}{2}}\alpha_i}{(h_i^2+\sigma_j^2)^{\frac{d}{2}}}\times\exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2}{(h_i^2+\sigma_j^2)}\Big)\Big). \tag{10}$$

将式(10)代入式(9),得出:

$$\bar{\varepsilon}_j^{\min} = -\frac{(\sqrt{2})^d(\sigma_j^2)^{\frac{d}{2}}}{(\sqrt{2\pi})^d}\Big(\sum_{i\in S_j}\frac{\alpha_i}{(h_i^2+\sigma_j^2)^{\frac{d}{2}}}\times \exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2}{(h_i^2+\sigma_j^2)}\Big)\Big)\Big)^2. \tag{11}$$

令 $\frac{\partial\bar{\varepsilon}_j^{\min}}{\partial\boldsymbol{t}_j}=0$,则:

$$\boldsymbol{t}_j = \Big(\sum_{i\in S_j}\frac{\alpha_i}{(h_i^2+\sigma_j^2)^{\frac{d}{2}}(h_i^2+\sigma_j^2)}\times \Big(\exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2}{(h_i^2+\sigma_j^2)}\Big)\Big)\Big)'\boldsymbol{x}_i\Big)\Big/ \Big(\sum_{i\in S_j}\frac{\alpha_i}{(h_i^2+\sigma_j^2)^{\frac{d}{2}}(h_i^2+\sigma_j^2)}\times \Big(\exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2}{(h_i^2+\sigma_j^2)}\Big)\Big)\Big)'\Big). \tag{12}$$

令 $\frac{\partial\bar{\varepsilon}_j^{\min}}{\partial\sigma_j^2}=0$,则:

$$\sigma_j^2 = \Big(\sum_{i\in S_j}\frac{\alpha_i}{(h_i^2+\sigma_j^2)^{\frac{d}{2}}(h_i^2+\sigma_j^2)}\Big(d\times \exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2}{(h_i^2+\sigma_j^2)}\Big)\Big)h_i^2 - 4\times \Big(\exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2}{(h_i^2+\sigma_j^2)}\Big)\Big)\Big)'\times \|\boldsymbol{t}_j-\boldsymbol{x}_i\|^2\frac{1}{(h_i^2+\sigma_j^2)}\sigma_j^2\Big)\Big)\Big/ \Big(d\times\sum_{i\in S_j}\frac{\alpha_i}{(h_i^2+\sigma_j^2)^{\frac{d}{2}}(h_i^2+\sigma_j^2)}\Big). \tag{13}$$

本文核合并的思路是局部近似,即用一个聚类核近似代替原样本集中性质相近的若干核,需指出的是,在核合并时混合核不仅考虑了输入样本,还考虑到了不同核宽度对合并结果的影响,尤其是在静态数据流密度估计中,每个数据窗的 KDE 核宽近似但却不同,演进数据流不同核宽度数据窗的合并时,引入核宽度对聚类的贡献尤为重要.

本文所提方法误差公式如式(9)所示,但在计算误差时步骤较复杂,下面提出核距离最近聚类法,将当前核合并到与其核距离最近的分块中,且证明核距离误差本质上与分块局部近似聚类误差等价,故在聚类时可将样本聚类到核距离最近的类,且计算误差时可用核距离误差代替式(9).

$$Q = \sum_{j=1}^{m}\sum_{i\in S_j}\alpha_i\int_{\mathbf{R}^d}(F_j(\boldsymbol{x})-K_i(\boldsymbol{x}))^2\mathrm{d}\boldsymbol{x}, \tag{14}$$

其中,$F_j(\boldsymbol{x})$类似于 $\beta_j G_j(\boldsymbol{x},\boldsymbol{t}_j)$的形式. 令 $Q=\sum_{j=1}^{m}Q_j$,则:

$$Q_j = \sum_{i\in S_j}\alpha_i\int_{\mathbf{R}^d}(F_j(\boldsymbol{x})-K_i(\boldsymbol{x}))^2\mathrm{d}\boldsymbol{x} = \sum_{i\in S_j}\alpha_i\int_{\mathbf{R}^d}(K_i(\boldsymbol{x}))^2\mathrm{d}\boldsymbol{x} + \sum_{i\in S_j}\alpha_i\int_{\mathbf{R}^d}(F_j(\boldsymbol{x}))^2\mathrm{d}\boldsymbol{x} - 2\sum_{i\in S_j}\alpha_i\int_{\mathbf{R}^d}F_j(\boldsymbol{x})K_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{15}$$

式(15)第 1 项与 $F_j(\boldsymbol{x})$无关,故可以省略.

用 $\tilde{F}_j(\boldsymbol{x})$ 表示式(8)最优解 $\beta_j G_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)$,即 $\tilde{F}_j(\boldsymbol{x})=\beta_j G_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)$,$F_j(\boldsymbol{x})$表示式(15)最优解,可证明 $\tilde{F}_j(\boldsymbol{x})=Z_jF_j(\boldsymbol{x})$,其中 $Z_j=\sum_{i\in S_j}\alpha_i$.

定理 1. 对每一个 $j=1,2,\cdots,m$,若 $\tilde{F}_j(\boldsymbol{x})$为分块局部近似法最优解,$F_j(\boldsymbol{x})$为最近核距离法最优解,则 $\tilde{F}_j(\boldsymbol{x})=Z_jF_j(\boldsymbol{x})$,且 $Z_j=\sum_{i\in S_j}\alpha_i$.

定理证明详见附录 A. 定理 1 证明了分块局部近似最优解与核距离最优解之间只多了一项常数因子,故两者的目标是一致的. 换而言之,在使用混合聚类核代替所有核时,到某一混合聚类核距离最近的核就被聚为那一类,且核距离计算为

$$D^2(F_j,K_i) = \int_{\mathbf{R}^d}(F_j(\boldsymbol{x})-K_i(\boldsymbol{x}))^2\mathrm{d}\boldsymbol{x} = \int_{\mathbf{R}^d}\Big(\frac{\tilde{F}_j(\boldsymbol{x})}{Z_j}-K_i(\boldsymbol{x})\Big)^2\mathrm{d}\boldsymbol{x} = \Big(\frac{1}{(\sqrt{2\pi})^d}\Big)$$

$$\left(\frac{\beta_j^2}{Z_j^2}(2\sigma_j^2)^{(-d/2)}+(2h_i^2)^{(-d/2)}-\frac{\beta_j}{Z_j}(h_i^2+\sigma_j^2)^{(-d/2)}\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{t}_j\|^2}{2(h_i^2+\sigma_j^2)}\right)\right). \quad (16)$$

按式(16)将 K_i 归类到 F_j:

$$j=\arg\min_k D^2(F_k,K_i). \quad (17)$$

通过上述分析,可得到混合聚类核概率密度估计算法:

算法 1. CMMKDE 算法.

输入:$\{\alpha_i,\boldsymbol{x}_i,h_i\}_{i=1}^n$;

输出:$\{\beta_j,\boldsymbol{t}_j,\sigma_j\}_{j=1}^m$.

赋初值:混合核个数 $m\leqslant 30$;使用 K-means 算法对 n 个样本进行初始聚类;$\xi=10^{10}$;$\xi^{\text{new}}=\xi$.

do

$\xi=\xi^{\text{new}}$;

for $j=1,2,\cdots,m$ 个混合聚类核

步骤 1. 为 $\boldsymbol{t}_j,\sigma_j^2$ 赋初值:

$$\boldsymbol{t}_j=\frac{\sum_{i\in S_j}\alpha_i\boldsymbol{x}_i}{\sum_{i\in S_j}\alpha_i};$$

$$\sigma_j^2=\frac{1}{\sum_{i\in S_j}\alpha_i}\sum_{i\in S_j}\alpha_j(h_i^2+(\boldsymbol{t}_j-\boldsymbol{x}_i)(\boldsymbol{t}_j-\boldsymbol{x}_i)^{\mathrm{T}});$$

步骤 2. 对 $\boldsymbol{t}_j,\sigma_j^2$ 寻优:

$\boldsymbol{t}_j^{\text{new}}=\boldsymbol{t}_j$;$\sigma_j^{2\text{new}}=\sigma_j^2$;

do

$p=0$;

do

$\boldsymbol{t}_j=\boldsymbol{t}_j^{\text{new}}$;

根据式(12)计算 $\boldsymbol{t}_j^{\text{new}}$;

$p=p+1$;

until abs$(\boldsymbol{t}_j^{\text{new}}-\boldsymbol{t}_j)\leqslant 0.0001$;

$q=0$;

do

$\sigma_j^2=\sigma_j^{2\text{new}}$;

根据式(13)计算 $\sigma_j^{2\text{new}}$;

$q=q+1$;

until abs$(\sigma_j^{2\text{new}}-\sigma_j^2)\leqslant 0.0001$;

until$(p\leqslant 1$ && $q\leqslant 1)$;

步骤 3. 按式(10)计算 β_j 值;

end for

计算上述寻优操作所获得$\{\beta_j,\boldsymbol{t}_j,\sigma_j\}_{j=1}^m$的聚类误差:

for 每个输入组合$\{\alpha_i,\boldsymbol{x}_i,h_i\}_{i=1}^n$ for 每个聚类点$\{\beta_j,\boldsymbol{t}_j,\sigma_j\}_{j=1}^m$

按式(16)计算每个输入向量到 m 个聚类点之间的距离;

end for

按式(17)将 $\boldsymbol{x}_i$ 归类到距离最近的那一类,并记下误差 $\xi_i=\min D^2(F_j,K_i)$;

end for

$\xi^{\text{new}}=\xi_1+\cdots+\xi_n$;

until abs$(\xi^{\text{new}}-\xi)<0.001\xi^{\text{new}}$;

获得混合核模型:$G(\boldsymbol{x})=\sum_{j=1}^m\beta_jG_{\sigma_j}(\boldsymbol{x},\boldsymbol{t}_j)$.

上述使用固定个数为 m-混合聚类核代替所有核的核进行概率密度估计,主要可用于以下 2 种场合:

1) 使用固定个数的混合聚类核表示每个时间窗数据流的概率密度估计式,如 KDE 密度估计,输入 $\alpha_i=\frac{1}{n}$,h_i 按交叉验证或式(5)获得,$\boldsymbol{x}_i$ 为样本输入值;

2) 随着时间的推移,数据流窗口个数增加且混合聚类核密度估计式也呈线性增长,为了控制存储空间的使用,按约定规则(如 KL 距离最小)将 2 个混合聚类核密度估计式合并,值得指出的是,此处的核合并涉及到不同核带宽的核合并.

下面重点讨论混合聚类核密度估计在数据流概率密度估计中的应用.

2　基于 MMCKDE 的数据流概率密度估计法

本节首先对文中涉及到的流数据模型、Kullback-Leilber(KL)度量等进行简单介绍,然后介绍基于 MMCKDE 的数据流概率密度估计法.

2.1　数据流模型

本节介绍我们所研究的数据流模型,其定义如文献[12].设 d 维数据流$\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n,\cdots\}$,其中 $\boldsymbol{x}_k=(x_{k1},x_{k2},\cdots,x_{kd})\in\mathbb{R}^d$ 且 k 与 d 均为整数,假设数据流按时间宽度 $T(i)$分割成窗口 W_i 的序列,每个 W_i 窗格数据独立同分布.故数据流可以由一系列窗格构成$\{W_1,W_2,\cdots,W_n,\cdots\}$,其中 $W_i=\{\boldsymbol{x}_i^1,\boldsymbol{x}_i^2,\cdots,\boldsymbol{x}_i^{T(i)}\}$且所有的 $\boldsymbol{x}_i^k\in W_i$ 均服从相同的分布 p_i.基于此数据流模型,我们使用固定个数为 m-混合核代替 KDE 或其他概率密度估计法中的核用来表示概率密度函数,通过这种方式减少密度估计函数的所需项,极大提高了数据流概率密度估计的效率.

2.2 Kullback-Leilber 距离

KL 距离用于度量 2 个随机变量的距离[13]. 设 P 和 Q 分别来自 2 个度量空间 (Ω, F), KL 距离可用式(18)表示:

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{\Omega} \mathrm{d}P \ln \frac{\mathrm{d}P}{\mathrm{d}Q} = \int_{\mathbf{R}^d} p(\boldsymbol{x}) \ln \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}, \tag{18}$$

其中, $p(\boldsymbol{x})$ 和 $q(\boldsymbol{x})$ 分别为 P 和 Q 的概率密度估计函数.

因本文实验真实数据集概率密度函数不可知, 故采用近似估计式 $\hat{p}(\boldsymbol{x})$ 和 $\hat{q}(\boldsymbol{x})$ 来计算 KL 距离:

$$\hat{D}_{\mathrm{KL}}(P \parallel Q) = \int_{\mathbf{R}^d} \hat{p}(\boldsymbol{x}) \ln \frac{\hat{p}(\boldsymbol{x})}{\hat{q}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}. \tag{19}$$

本文实验为 1 维数据集, 我们使用下面近似式:

$$\hat{D}_{\mathrm{KL}}(P \parallel Q) = \sum_{i=1}^{T} f\left(\frac{\boldsymbol{x}_i + \boldsymbol{x}_{i-1}}{2}\right)(\boldsymbol{x}_i - \boldsymbol{x}_{i-1}), \tag{20}$$

其中, $f(\boldsymbol{x}) = \hat{p}(\boldsymbol{x}) \ln \dfrac{\hat{p}(\boldsymbol{x})}{\hat{q}(\boldsymbol{x})}$, $\boldsymbol{x}_0 < \boldsymbol{x}_1 < \cdots < \boldsymbol{x}_T$ 为 T 等分点, $[\boldsymbol{x}_0, \boldsymbol{x}_T] \in \mathbb{R}^d$, $\hat{p}(\boldsymbol{x})$, $\hat{q}(\boldsymbol{x})$ 为相邻 2 时间窗密度估计。我们在下面算法中, 说明在建立 MMCKDE 序列和合并 MMCKDE 序列时 KL 距离的具体计算方法.

2.3 基于 MMCKDE 的数据流概率密度估计法

基于 MMCKDE 的数据流概率密度估计法主要包含 3 个操作: 1)建立混合聚类核序列; 2)合并混合聚类核序列; 3)概率密度估计.

2.3.1 建立 MMCKDE 序列

由图 2 所示, 每个时间窗数据流均使用 MMCKDE 算法使用固定个数为 m 的混合聚类核进行概率密度估计. 设 W_t 是第 t 个时间窗数据流, 我们用 MMCKDE 节点 $\langle MMCKDE_i, c_i, kl_i, r_i \rangle$ 来表示第 t 个时间窗数据流 W_t 的概率密度估计信息, 其中, $MMCKDE_i$ 为使用固定核个数为 m 的混合聚类核概率密度估计式参数, 由聚类中心点 $\boldsymbol{t}$、核宽 σ、系数 β 构成; c_i 表示 W_t 时间窗数据流规模; kl_i 表示当前时间窗数据流与前一时间窗数据流的 KL 距离, 其中 $\hat{p}(\boldsymbol{x})$ 为当前时间窗密度估计, 由其相应的 MMCKDE 节点中 $MMCKDE_i$ 项存储的 m-混合聚类核概率密度估计式参数 $\boldsymbol{t}$, σ 和 β 获得, 即 $\hat{p}(x) = \sum_{j=1}^{m} \beta_i^j G_{\sigma_i^j}(\boldsymbol{x}, \boldsymbol{t}_i^j)$; $\hat{q}(\boldsymbol{x})$ 为前一时间窗密度估计, 由其相应的 MMCKDE 节点中 $MMCKDE_{i-1}$ 项存储的 m-混合聚类核概率密度估计式参数 $\boldsymbol{t}$, σ 和 β 获得, 即 $\hat{q}(\boldsymbol{x}) = \sum_{j=1}^{m} \beta_{i-1}^j G_{\sigma_{i-1}^j}(\boldsymbol{x}, \boldsymbol{t}_{i-1}^j)$; r_i 表示当前 $MMCKDE_i$ 估计式所表示的时间窗下标起始值与终止值, 例如 $r_i = [a, b]$ 表示概率密度估计式对第 a 个时间窗数据流至第 b 个时间窗数据流间所有数据进行密度估计, 此例仅对第 t 个时间窗数据流进行概率密度估计, 故时间窗下标起始值与终止值均为 t, 即 $r_i = [t, t]$.

前面引言已经提到, 为了进一步节约存储空间, 当满足一定条件时会涉及到核的合并, 即 $\langle MMCKDE_i, c_i, kl_i, r_i \rangle$ 也可以表示为若干时间窗的数据流概率密度估计信息. 当 W_t 节点中 KL 距离最小时, 表明 W_{t-1} 与 W_t 密度最相近, 可将 W_{t-1} 的 $\langle MMCKDE_{t-1}, c_{t-1}, kl_{t-1}, r_{t-1} \rangle$ 节点与 W_t 的 $\langle MMCKDE_t, c_t, kl_t, r_t \rangle$ 节点合并, 合并后的 $MMCKDE_{\mathrm{new}}$ 节点仍为由 m 个聚类核构成的密度估计函数, c_{new} 表示合并后 2 个节点 c_i 项的总和, kl_{new} 表示合并后的节点与前一节点数据集间的 KL 距离, r_{new} 表示当前 $MMCKDE_{\mathrm{new}}$ 概率密度估计式所表示的时间窗下标初始值与终止值, 为 $[t-1, t]$. 图 3 示出了 W_2 与 W_3 的合并过程.

令 $SL_i = \langle MMCKDE_i, c_i, kl_i, r_i \rangle$ 表示 MMCKDE 节点, $SL = \{SL_1, SL_2, \cdots, SL_i\}$ 即构成 MMCKDE 序列. 其中因涉及到 MMCKDE 节点的合并, 故节点个数小于时间窗个数, 即 $i < t$. 若设 MMCKDE 序列规模固定, 且远小于数据流规模, 则与原数据流相比所需存储空间大大降低. 下面介绍 MMCKDE 序列合并规则.

2.3.2 合并 MMCKDE 序列

1) 合并方式

为了进一步节约存储空间, 根据数据流的不同类型, 采用 3 种方式进行 MMCKDE 节点合并.

(1) 方式 1. 固定存储空间策略.

当存储 MMCKDE 序列的存储空间固定时, 意味着 MMCKDE 节点的个数固定, 当有新的数据窗到达, 生成一个新的 MMCKDE 节点 SL_i, 如果节点数超过规定数 n, 则意味着超出了所分配内存, 需在序列中找到 KL 距离最接近的 2 个 MMCKDE 节点进行合并. 即 $|kl_j| = \min(kl_2, kl_3, \cdots, kl_{n+1})$, $j \in \mathbb{Z}$, $2 \leqslant j \leqslant n+1$, SL_{j-1} 与 SL_j 合并. MMCKDE 概率密度估计的特点是可以使用较少的混合聚类核, 来表示某段数据流的概率密度估计式, 使用固定规模为 m 的混合聚类核代替所有核固然可以减少存储空间, 但每一次核的合并均会产生一定的误差, 而这种

误差在不断的核合并中不断累加.故在固定MMCKDE序列核总数即存储数据流的存储空间固定的情况下,混合聚类核个数应取小一些,节点应尽可能地取多一些.例如存储空间固定为200,我们将其分成10个MMCKDE节点,每个节点包含20个混合聚类核.

(2) 方式2. 固定阈值策略.

该方式采用固定阈值的方式来控制每个MMCKDE节点所表示的时间窗个数,主要用于演进数据流密度估计.设阈值为 θ,当2个临近的MMCKDE节点的KL距离小于阈值 θ,则合并.例如,如果 $kl_j<\theta$,则 SL_{j-1} 与 SL_j 合并.通过这种方式可以将相似的数据合并,不同分布的时间窗数据流在不同MMCKDE节点上表示,可用于演进数据流的概率密度估计.

(3) 方式3. 固定存储空间策略与固定阈值策略相结合.

当进行演进数据流概率密度估计时,采用固定阈值方式可使不同分布的数据流用不同的MMCKDE节点表示,但这种方式可能会导致节点数目不可控,为了解决这个问题,可采用固定存储空间策略与固定阈值策略相结合的方式,若节点数超过规定的数目,则将KL距离最近的2个节点合并.

2) 核合并过程

假设 SL_{j-1} 与 SL_j 合并,合并后的密度估计式为

$$\hat{p}(\boldsymbol{x})=\gamma_1\hat{p}_{j-1}(\boldsymbol{x})+\gamma_2\hat{p}_j(\boldsymbol{x}), \tag{21}$$

其中,$\gamma_1=\frac{c_{j-1}}{c_{j-1}+c_j}$,$\gamma_2=\frac{c_j}{c_{j-1}+c_j}$,$\hat{p}_j(\boldsymbol{x})=\sum_{i=1}^{m}\beta_j^i G_{\sigma_j^i}(\boldsymbol{x},\boldsymbol{t}_j^i)$.故:

$$\hat{p}(\boldsymbol{x})=\frac{c_{j-1}}{c_{j-1}+c_j}\sum_{i=1}^{m}\beta_{j-1}^i G_{\sigma_{j-1}^i}(\boldsymbol{x},\boldsymbol{t}_{j-1}^i)+\frac{c_j}{c_{j-1}+c_j}\sum_{i=1}^{m}\beta_j^i G_{\sigma_j^i}(\boldsymbol{x},\boldsymbol{t}_j^i). \tag{22}$$

图4显示了核合并的过程.将 SL_{j-1} 和 SL_j 节点进行合并时,所有节点中已有核作为输入数据,其中:$X=\{\boldsymbol{t}_{j-1}^1,\boldsymbol{t}_{j-1}^2,\cdots,\boldsymbol{t}_{j-1}^m,\boldsymbol{t}_j^1,\boldsymbol{t}_j^2,\cdots,\boldsymbol{t}_j^m\}$,$h=\{\sigma_{j-1}^1,\sigma_{j-1}^2,\cdots,\sigma_{j-1}^m,\sigma_j^1,\sigma_j^2,\cdots,\sigma_j^m\}$,$\alpha=\{\beta_{j-1}^1\gamma_1,\beta_{j-1}^2\gamma_1,\cdots,\beta_{j-1}^m\gamma_1,\beta_j^1\gamma_2,\beta_j^2\gamma_2,\cdots,\beta_j^m\gamma_2\}$.需指出的是,本文提出的合并算法在核合并时考虑到了核宽的影响,故在不同核带宽间的核合并精度上有一定的优势.聚类新的 SL_{new} 节点 $T_{\text{new}}=\{\boldsymbol{t}_{\text{new}}^1,\boldsymbol{t}_{\text{new}}^2,\cdots,\boldsymbol{t}_{\text{new}}^m\}$,$\beta_{\text{new}}=\{\beta_{\text{new}}^1,\beta_{\text{new}}^2,\cdots,\beta_{\text{new}}^m\}$,$\sigma_{\text{new}}=\{\sigma_{\text{new}}^1,\sigma_{\text{new}}^2,\cdots,\sigma_{\text{new}}^m\}$.$c_{\text{new}}=c_{j-1}+c_j$ 表示参与 SL_{new} 概率密度估计样本总数.r_{new} 表示 SL_{new} 概率密度估计所表示到的时间窗下标范围,若 SL_{j-1} 的 $r_{j-1}=[j-1,j-1]$,SL_j 的 $r_j=[j,j]$,则 $r_{\text{new}}=[j-1,j]$.kl_{new} 表示 SL_{j-2} 与 SL_{new} 之间的KL距离,此外,还要更新 SL_{j+1} 节点中的 kl_{j+1} 值,替换为 SL_{j+1} 和 SL_{new} 节点间的KL距离.kl_{new} 计算方法为使用 SL_{new} 节点中 $\boldsymbol{t}_{\text{new}}$,$\beta_{\text{new}}$ 和 σ_{new} 计算 $\hat{p}(x)=\sum_{i=1}^{m}\beta_{\text{new}}^i G_{\sigma_{\text{new}}^i}(\boldsymbol{x},\boldsymbol{t}_{\text{new}}^i)$,使用 $MMCKDE_{j-2}$ 节点中 $\boldsymbol{t}_{j-2}$,β_{j-2} 和 σ_{j-2} 计算 $\hat{q}(x)=\sum_{i=1}^{m}\beta_{j-2}^i G_{\sigma_{j-2}^i}(\boldsymbol{x},\boldsymbol{t}_{j-2}^i)$,$kl_{j+1}$ 值计算方法相同.需指出的是,这种合并可将任意长度的2个节点进行合并,并不要求2个节点规模一致.

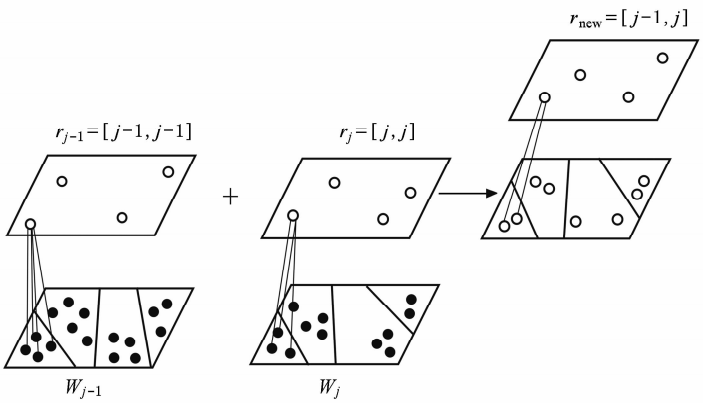


Fig. 4　Merging procedure over entries.

图4　MMCKDE节点合并示意图

全面性和重要性是时间序列很重要的2个方面.时间序列数据流处理既要学习新的更重要的新信息,同时不能丢弃旧的模型.到目前为止,我们主要讨论了存储整个数据流的信息,即数据流的全面性.若要考虑数据的重要性,则新数据比旧数据更重要,在数据流中体现为当其他条件相同的前提下,旧数据先被合并,为了达到这个目的,我们引入了新的KL距离估计式:

$$kl'_j = g(kl_j, \bar{r}_j), \tag{23}$$

其中，$g(\cdot)$是$\mathbb{R}\times\mathbb{R}^+\to\mathbb{R}$的映射，$\bar{r}_j$ 是 SL_j 数据流时间窗的均值，即 $\bar{r}_j=\frac{1}{2}(r_j(1)+r_j(2))$，如式(23)所示，修改后的 KL 距离不仅依赖于原 kl_j 值，也与时间窗跨度有关. 当 kl_j 固定时，新窗口具有较大的 $\bar{r}_j$，意味着较大的 kl'_j. 本文定义 kl'_j：

$$kl'_j = g(kl_j, \bar{r}_j) = kl_j \times \exp(-\lambda(t-\bar{r}_j)), \tag{24}$$

其中，t 为当前时间窗下标，λ 为控制下降速率的非负参数. 若 λ 设置为 0，则 $\exp(-\lambda(t-\bar{r}_j))$为 1，表示旧数据和新数据一样重要. 增大 λ 值，若 kl_j 值相同，则旧 SL 数据被合并. 若新进来的窗口 $r_j=[t,t]$，则 $kl'_j=kl_j$. 若旧数据比新数据更重要，可设置 $\lambda<0$.

2.3.3 最终概率密度估计

最后一步工作是将所有的 MMCKDE 序列用于概率密度估计. 主要有 2 种估计方式：

1) 按时间窗进行概率密度估计

如图 5 所示，假设需要进行密度估计的时间窗下标范围为$[t,T]$，找到两 MMCKDE 序列 SL_i 和 SL_{i+h}，其中 $r_i(1)\leqslant t\leqslant r_i(2)$ 且 $r_{i+h}(1)\leqslant T\leqslant r_{i+h}(2)$：

$$\hat{p}(\boldsymbol{x}) = \gamma_i\hat{p}_i(\boldsymbol{x}) + \cdots + \gamma_{i+h}\hat{p}_{i+h}(\boldsymbol{x}), \tag{25}$$

其中，$\gamma_j = c_j \Big/ \sum_{k=i}^{i+h} c_k$，$\hat{p}_j(x) = \sum_{k=1}^{m} \beta_j^k G_{\sigma_j^k}(\boldsymbol{x}, \boldsymbol{t}_j^k)$，$i\leqslant j\leqslant i+h$.

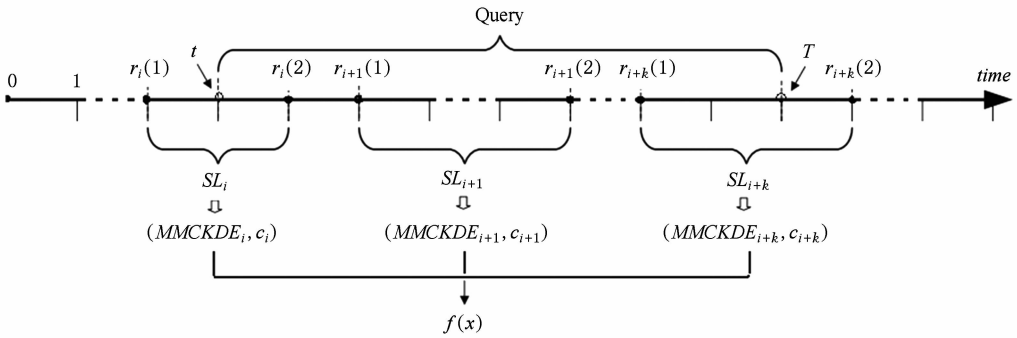


Fig. 5 Density estimation over arbitrary time period.

图 5 任意段密度估计示意图

2) 整个数据流概率密度估计

若式(25)中 $i=1$，$i+h$ 取最后一个时间窗下标，式(25)可用于计算整个数据流的概率密度估计信息.

算法 2. 数据流密度估计 *DataStreamDensityEstimate*(X).

输入：数据流 $X=\{W_1, W_2, \cdots, W_N, \cdots\}$，其中 $W_i=\{\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \cdots, \boldsymbol{x}_i^{T(i)}\}$，MMCKDE 节点个数：$n=10$，每个节点混合核个数 $m=(5,10,15,20,25,30)$，数据窗宽 $n_t=500$；

输出：数据流 X 的密度估计式 $\hat{f}(\boldsymbol{x})$.

过程：

MMCKDEList.setNodeSize(n)；

/* 设置 MMCKDE 节点个数为 n */

for $i=1,2,\cdots,N$

读入数据窗 W_i；

求 KDE 核宽：

方法 1. 交叉验证；

方法 2. 近似算法：$h_i=1.06std(W_i)n_t^{-\frac{1}{5}}$，

std 用于求样本的标准方差；

CurMMCKDE$(\boldsymbol{t},\sigma,\beta)$=*MMCKDE*$(W_i,h_i,\alpha)$；

计算新 MMCKDE 节点与前一 MMCKDE 节点的 KL 距离；

MMCKDEList.add$(CurMMCKDE)$；

/* 将当前节点加入 MMCKDE 序列 */

if MMCKDE 序列已满

找到最小 KL 值的节点；

将该节点与前一节点合并；

重新计算相关节点各参数值；

end if

end for

for 所有 MMCKDE 序列中的节点

$\hat{f}(\boldsymbol{x})+=\gamma_i\hat{p}_i(\boldsymbol{x})$，其中：

$\gamma_i = c_i \Big/ \sum_{k=1}^{MMCKDEList.Count()} c_k$，

$\hat{p}_i(x) = \sum_{l=1}^{m} \beta_i^l G_{\sigma_i^l}(\boldsymbol{x}, \boldsymbol{t}_i^l)$；

end for

3　实验与分析

本节实验分为 3 部分:1)验证 MMCKDE 算法(算法 1)的优势;2)静态数据流实验;3)演进数据流实验.3 部分实验均与 SOMKE 算法在数据流密度估计精度及运行时间上进行对比.

为了测试估计密度 $\hat{f}$ 的性能,本文选用均积分平方误差(mean-integrated squared error, *MISE*)进行性能比较.对于真实数据流,因实际应用场景中密度未知,故本文用整个数据流的核密度估计作为近似真实密度.*MISE* 计算如下:

$$MISE(\hat{f}) = E\left[\int(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2 \mathrm{d}\boldsymbol{x}\right] \approx \frac{1}{N}\sum_{i=1}^{N}(\hat{f}(\boldsymbol{x}_i) - f(\boldsymbol{x}_i))^2 \Delta\boldsymbol{x},$$

其中,测试集 $X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$,且 $\boldsymbol{x}$ 间等步长 $\Delta\boldsymbol{x}$.

SOMKE 算法参数设置如文献[14],其中,迭代次数 $n=3\,000$;初始学习率 $\eta_0=3$;初始领域宽度 $\sigma_0=25$;时间常量 $\tau_1=1\,000/\ln\sigma_0$ 和 $\tau_2=1\,000$.

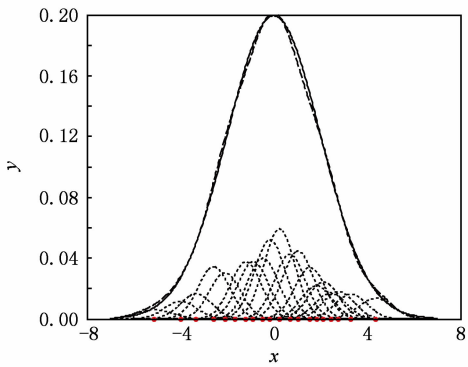
3.1　MMCKDE 算法优势分析

从第 2 节分析可知,MMCKDE 算法的优势主要有如下 2 点:

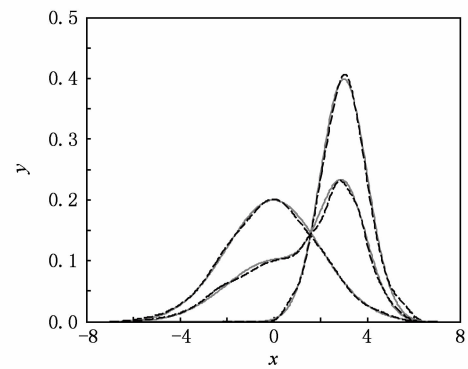
1) 可使用较少的核表示单个数据窗密度估计;

2) 支持不同核宽的窗口数据合并,这一特性有利于数据流中节点的进一步合并.

为了验证上述两大优势,分别生成包含 500 个元素的均值为 0、方差为 2、记为 $N(0,2)$,均值为 3、方差为 1、记为 $N(3,1)$的数据集,实验分 2 部分进行,首先设置 m 值(SOM 算法为神经元个数,MMCKDE 算法为混合核个数)为 5,10,15,20,25,30,使用 MMCKDE 算法与 SOM 算法用较少的聚类核代替所有核进行概率密度估计,比较 2 种算法的精度;接着,将所获得的聚类核进一步进行合并,即使用 m 个聚类核表示 2 个数据集合并后的概率密度估计.图 6(a)显示了使用 20 个混合核代替 500 个样本数据进行概率密度估计的示意图,图 6(b)显示了 MMCKDE 算法将不同核宽数据窗合并后的效果图,由图 6 可知,由于 MMCKDE 算法在进行核聚类时考虑到了核宽的影响,故聚类性能较好.2 种算法比较结果如表 1 所示.



(a) m cluster kernels



(b) Chart after merging at different kernel width

Fig. 6　An example of MMCKDE algorithm with m being 20.

图 6　MMCKDE 算法($m=20$)

Table 1　*MISE* Comparison on Different m Values

表 1　不同 m 值时 *MISE* 性能比较

m	$N(0,2)$		$N(3,1)$		Merge	
	MMCKDE	SOM	MMCKDE	SOM	MMCKDE	SOM
5	6.1660E−007	2.8939E−006	1.5960E−006	1.0063E−005	1.1559E−005	2.2636E−004
10	1.9380E−007	2.7776E−007	7.9840E−007	1.2639E−006	2.8800E−006	4.4399E−005
15	1.6690E−007	2.0699E−007	7.3520E−007	8.6946E−007	9.5810E−007	4.5422E−006
20	1.6360E−007	1.7419E−007	7.2130E−007	8.2394E−007	6.2630E−007	1.9107E−006
25	1.5960E−007	1.7883E−007	7.1900E−007	7.7572E−007	5.5120E−007	1.1304E−006
30	1.6140E−007	1.5526E−007	7.1360E−007	7.3593E−007	4.4610E−007	9.0318E−007

由表1和图6可知,当 m 取较小值时,MMCKDE算法的 $MISE$ 优于SOM算法,随着 m 增大,2种算法 $MISE$ 值逐渐接近,即将MMCKDE算法应用于数据流概率密度估计,其优势在于当 m 取较小值时,其概率密度估计精度较SOMKE算法更高.由表1还可知,当需合并的2个数据窗核宽不相等时,MMCKDE算法的精度明显优于SOM算法,其原因在于核宽对概率密度估计有重要的影响,而MMCKDE算法在进行核聚类时考虑到了这种影响.事实上,m 取较小值和不同核宽数据窗合并这2个特性,可使所得概率密度估计以更紧凑的形式表示,从而便于实际应用时存储空间有限与更高实时处理的需要.下面实验将在数据流中验证实验1所得的结论.

3.2 静态数据流密度估计

本节实验选用数据规模为100 000的2个模拟一维数据集和1个二维数据集进行实验,并与文献[14]提出的SOMKE算法进行比较.设计2个一维模拟数据集分别为一维高斯分布、一维高斯混合分布,其中,一维高斯分布均值方差均设置为1;高斯混合型模拟数据集一半数据来自均值4、方差2,一半数据来自均值−4、方差2;此外,本文按文献对二维数据benchmark数据流进行实验,该数据集均值为[2,2],协方差矩阵为 $\begin{pmatrix}2 & 0\\ 0 & 2\end{pmatrix}$,数据规模为50 000.

1) 一维数据流实验

为了验证实验1所得结论,本节实验同样将 m 值设为5,10,15,20,25,30,节点总数设为10,采用固定存储空间策略进行实验.表2列出了 m 值不同时2种算法的 $MISE$ 及运行时间情况,图7显示了 $m=30$ 时概率密度估计图.

由表2和图7可知,当 m 取较小值时,虽然SOMKE算法在运行时间上优于MMCKDE算法,但MMCKDE算法的 $MISE$ 明显优于SOMKE算法.当 $m=15$ 时,MMCKDE算法的 $MISE$ 均达到 10^{-6} 数量级,而SOMKE算法的 $MISE$ 还处于 10^{-5} 数量级,处于明显劣势;当 $m\geqslant 15$ 时,MMCKDE算法的 $MISE$ 及运行时间均明显优于SOMKE算法.

Table 2 *MISE* and Running Time at Different m on 1-dimensional Artificial Data Streams

表2 不同 m 值时一维模拟数据流 *MISE* 及运行时间比较

m	Gaussian				Gaussian Mixture			
	MMCKDE		SOMKE		MMCKDE		SOMKE	
	MISE	Running Time/s	*MISE*	Running Time/s	*MISE*	Running Time/s	*MISE*	Running Time/s
5	3.1731E−005	180.395 3	3.9176E−005	99.328 1	1.1826E−005	248.303 1	4.9932E−005	99.715 6
10	1.4159E−005	148.775 0	2.7488E−005	159.520 3	1.0485E−005	182.056 3	1.5123E−005	159.360 9
15	8.2026E−006	200.409 4	1.1807E−005	228.312 5	6.5862E−006	208.861 0	1.0302E−005	221.064 1
20	4.0402E−006	262.637 5	6.9414E−006	297.834 4	4.1419E−006	266.056 3	6.5582E−006	278.328 1
25	2.4721E−006	306.253 1	5.1742E−006	363.140 6	3.5127E−006	307.429 7	4.2490E−006	350.937 5
30	1.8596E−006	366.346 9	4.3250E−006	429.061 0	2.6976E−006	361.321 9	4.0424E−006	412.354 7

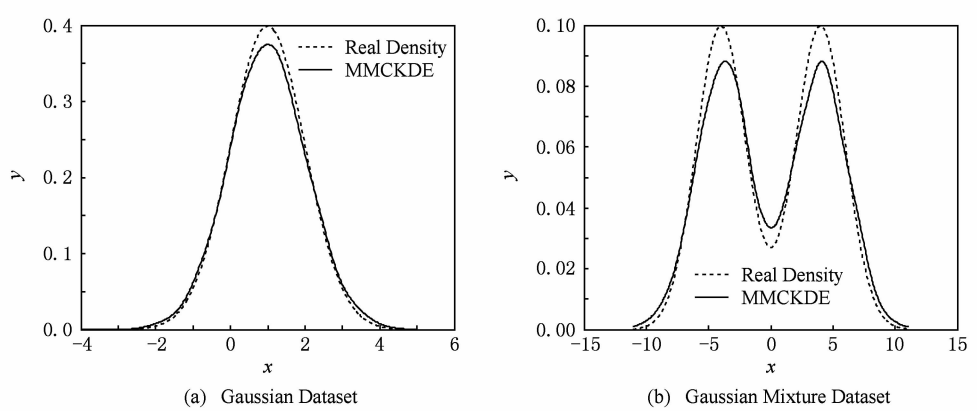


Fig. 7 Off-line density estimation over one dimensional artificial data with m being 30.

图7 1维数据流概率密度估计示意图($m=30$)

通过一维数据流实验进一步验证了当 m 取较小值时 MMCKDE 算法的性能优势,且在核聚类时考虑到了核宽的影响,故性能优于 SOMKE 算法.

2) 二维数据流实验

本实验将 m 值设为 20,节点总数为 10,采用固定存储空间策略进行实验.实验效果如图 8 所示.

图 8 显示了二维高斯数据流 MMCKDE 概率密度估计结果,可以看出,用本文算法进行密度估计所得到的密度估计图能较好的与真实密度分布基本保持一致.图 8(a)(c)显示真实密度分布;图 8(b)(d)显示 MMCKDE 估计结果.图 8(a)(b)显示密度分布,图 8(c)(d)显示密度分布等高线.需指出的是本文算法使用混合聚类核代替高斯核,可对多维数据流进行概率密度估计.

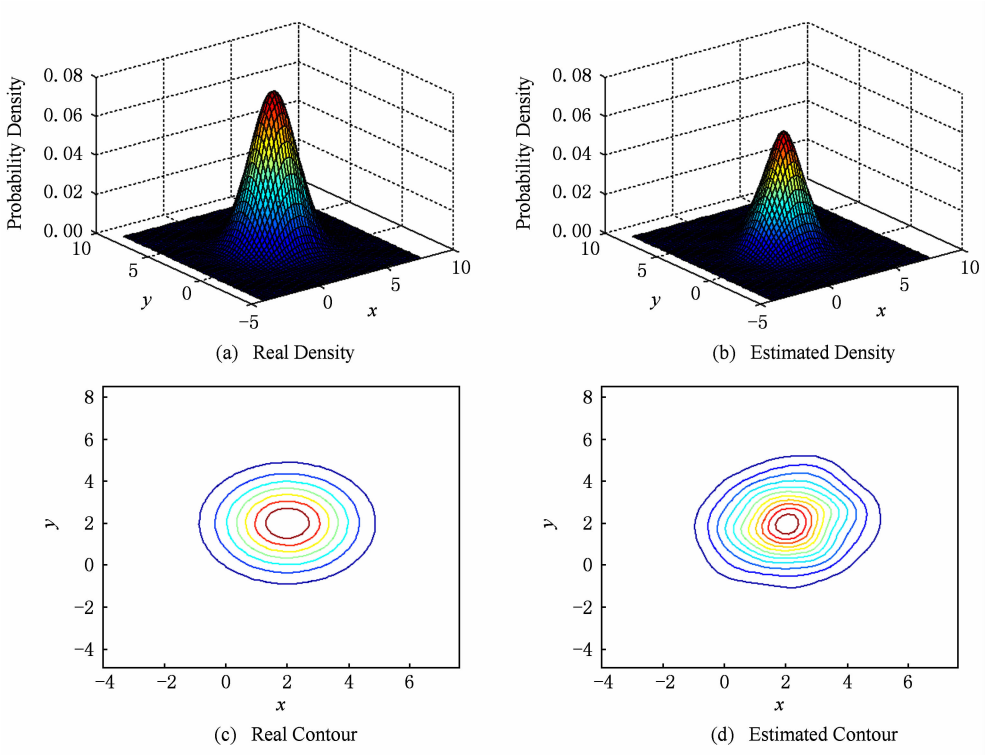


Fig. 8　Density functions of the 2-dimensional data stream with m being 20.

图 8　2 维数据流实验效果($m=20$)

3.3　演进数据集实验

文献[15]指出,真实数据流中静态数据流较少,大多数据流的密度分布随着时间的变化会发生改变,故本节将从基准数据集 benchmark 及真实数据流 2 方面对演进数据流进行实验.

1) benchmark 数据集实验

benchmark 数据集[14],数据流规模为 30000,取窗宽为 500,则共 60 个窗.数据流分布如下:

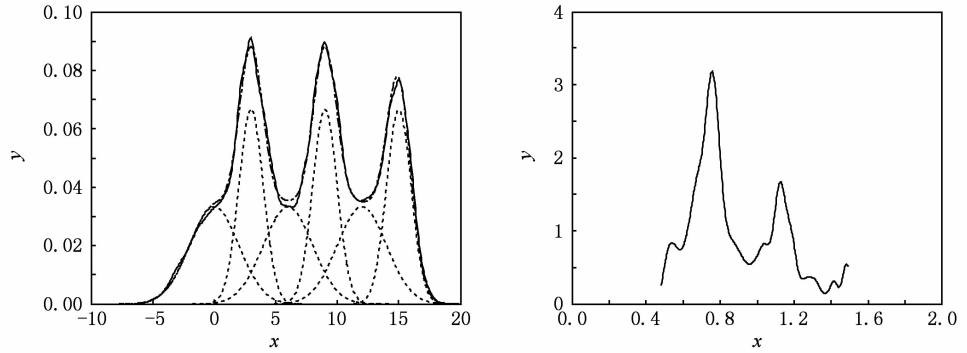
$$W_t=\begin{cases}N(0,2):t\in[1,10],\\ N(3,1):t\in[11,20],\\ N(6,2):t\in[21,30],\\ N(9,1):t\in[31,40],\\ N(12,2):t\in[41,50],\\ N(15,1):t\in[51,60],\end{cases}$$

其中 $t\in\mathbb{Z}$.图 9(a)显示了上述分布分段及整个数据流密度分布情况,其中点线表示分段数据流密度分布,虚线表示整个数据流密度分布,实线表示 $m=30$ 时概率密度估计曲线.

本文 2.3.2 节提到,修改 KL 距离公式可将新旧数据重要性作为节点合并的依据,故本节按新、旧数据同等重要($\lambda=0$)及新数据比旧数据重要($\lambda=0.085$)两种方式进行实验.

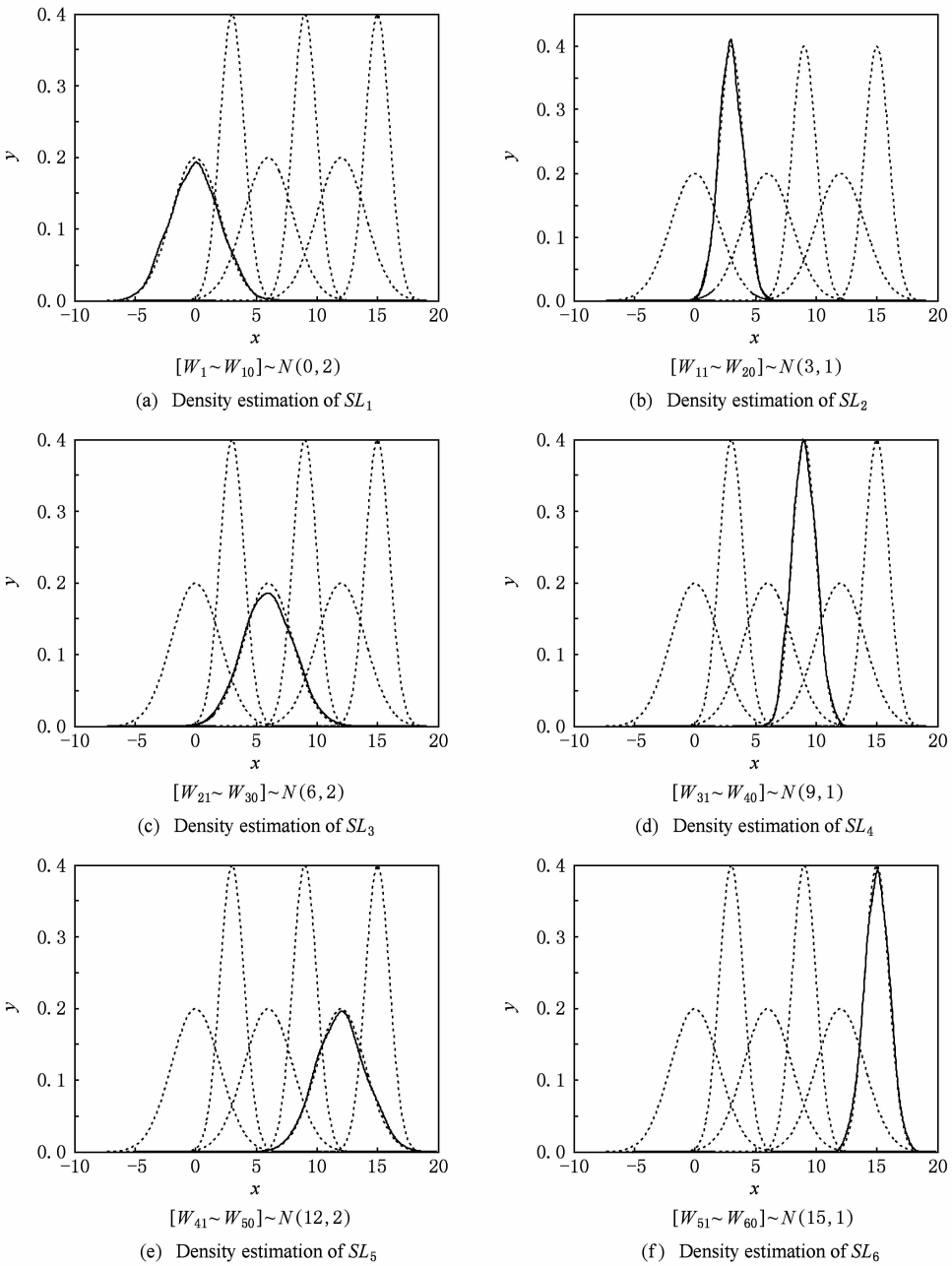
① 新、旧数据同等重要($\lambda=0$)实验

实验设置不同 m 值,且选用阈值法,即 MMCKDE 节点间的 KL 距离 $kl_j<1$ 则合并的方式,比较 SOMKE 及 MMCKDE 算法对整个数据流的概率密度估计精度.其中,KL 距离按式(24)计算,故具体节点数目由演进数据流密度变化决定.表 3 列出了不同 m 值时,2 种算法 *MISE* 及运行时间的比较.图 10 显示了 $m=30$ 时演进数据流各段概率密度估计曲线.



(a) Benchmark Dataset—Evolving Gaussian (b) Real Dataset—Financial Data

Fig. 9 Density estimation of evolutionary data streams.



$[W_1 \sim W_{10}] \sim N(0,2)$

(a) Density estimation of SL_1

$[W_{11} \sim W_{20}] \sim N(3,1)$

(b) Density estimation of SL_2

$[W_{21} \sim W_{30}] \sim N(6,2)$

(c) Density estimation of SL_3

$[W_{31} \sim W_{40}] \sim N(9,1)$

(d) Density estimation of SL_4

$[W_{41} \sim W_{50}] \sim N(12,2)$

(e) Density estimation of SL_5

$[W_{51} \sim W_{60}] \sim N(15,1)$

(f) Density estimation of SL_6

Fig. 10 Estimated probability density functions with each entry in the MMCKDE sequence with λ being 0.

图 10 演进数据流 MMCKDE 密度估计($\lambda=0$)

从图 10 及表 3 可知,当 $\lambda=0$ 时,MMCKDE 序列包含 6 个节点,每个节点用于表示分布相同的数据流窗的概率密度估计,例如 SL_1 表示数据窗 1～10 分布为 $N(0,1)$的数据流概率密度.2 种算法均可应用于演进数据流概率密度估计,且 MMCKDE 算法性能略优于 SOMKE 算法.

Table 3　*MISE* Comparison on Different m Values

表 3　不同 m 值时 2 种算法精度时间比较

m	Evolutionary Data Stream($L=6$)			
	MMCKDE		SOMKE	
	MISE	Running Time/s	*MISE*	Running Time/s
5	3.3152E−006	59.5656	6.3917E−006	30.9375
10	1.3620E−006	46.2531	3.5808E−006	48.6922
15	7.8254E−007	60.7781	1.3556E−006	69.0219
20	4.4964E−007	76.2641	7.2203E−007	88.8156
25	4.0273E−007	92.3125	5.4058E−007	109.4297
30	2.1525E−007	105.6078	4.0014E−007	133.1672

但采用阈值法进行概率密度估计,节点的个数取决于数据流变化的频率,若数据流密度变化频繁,可能导致节点过多,存储空间不可控的问题.为了解决这一问题,有 2 种方式:1)前面提到的修改 KL 距离公式,将新旧数据重要性作为节点合并的依据;2)将阈值法与固定存储空间法相结合.故本文下面按新数据比旧数据重要方式进行实验.

② 新数据比旧数据重要($\lambda=0.085$)实验

新、旧数据重要性判定通过式(24)λ 值进行判定.实验 1 中 $\lambda=0$,即旧数据与新数据一样重要,若将 λ 值设置为 0.085,降低旧数据的 KL 值,可让旧数据先合并.表 4 列出了不同 λ 值合并窗口范围 r_j,原 KL 距离 kl_j 和改进后 KL 距离 kl'_j 值.表 5 列出了 λ 值设置为 0.085 时,2 种算法各节点及总概率密度估计 *MISE* 值.图 11 显示了 λ 值设置为 0.085 时 MMCKDE 节点所代表的每段数据流概率密度分布.

Table 4　MMCKDE Sequence Entries over Evolving Data Stream

表 4　演进数据流各 MMCKDE 节点

SL_j	$\lambda=0$			$\lambda=0.085$		
	r_j	kl_j	kl'_j	r_j	kl_j	kl'_j
SL_1	[1,10]			[1,20]		
SL_2	[11,20]	1.3083	1.3083	[21,40]	8.2800	3.3917
SL_3	[21,30]	5.7985	5.7985	[41,50]	8.9294	6.0913
SL_4	[31,40]	1.3778	1.3778	[51,60]	1.5170	1.0349
SL_5	[41,50]	9.6303	9.6303			
SL_6	[51,60]	1.5358	1.5358			

Table 5　*MISE* Comparison of Two Algorithms When λ is Equal to 0.085

表 5　$\lambda=0.085$ 时 2 种算法 *MISE* 比较

Algorithm	Total Data Stream	SL_1	SL_2	SL_3	SL_4
MMCKDE	6.5722E−007	3.0453E−006	1.8783E−006	1.0428E−006	2.4194E−006
SOMKE	1.2449E−006	5.1529E−006	4.2388E−006	1.5723E−006	4.8594E−006

当 $\lambda=0.085$ 时,新信息比旧信息更重要,新信息存储更详细,而旧 MMCKDE 序列更有可能被合并.λ 值的增大决定了旧信息被合并的概率增大.本文实验获得了 4 个 MMCKDE 节点如图 11 所示,与 $\lambda=0$ 相比,前面 4 个 MMCKDE 节点被合并成 2 个 MMCKDE 节点.虽然前面 40 个窗口被合并成 2 个 MMCKDE 节点,但仍能较好地体现历史数据的概率密度分布,在节约存储空间的同时,并没有影响整个数据流的概率密度估计.

从表 5 实验数据进一步可知,由于涉及到不同核宽的数据窗的合并,故 MMCKDE 算法的 MISE 值明显优于 SOMKE 算法.

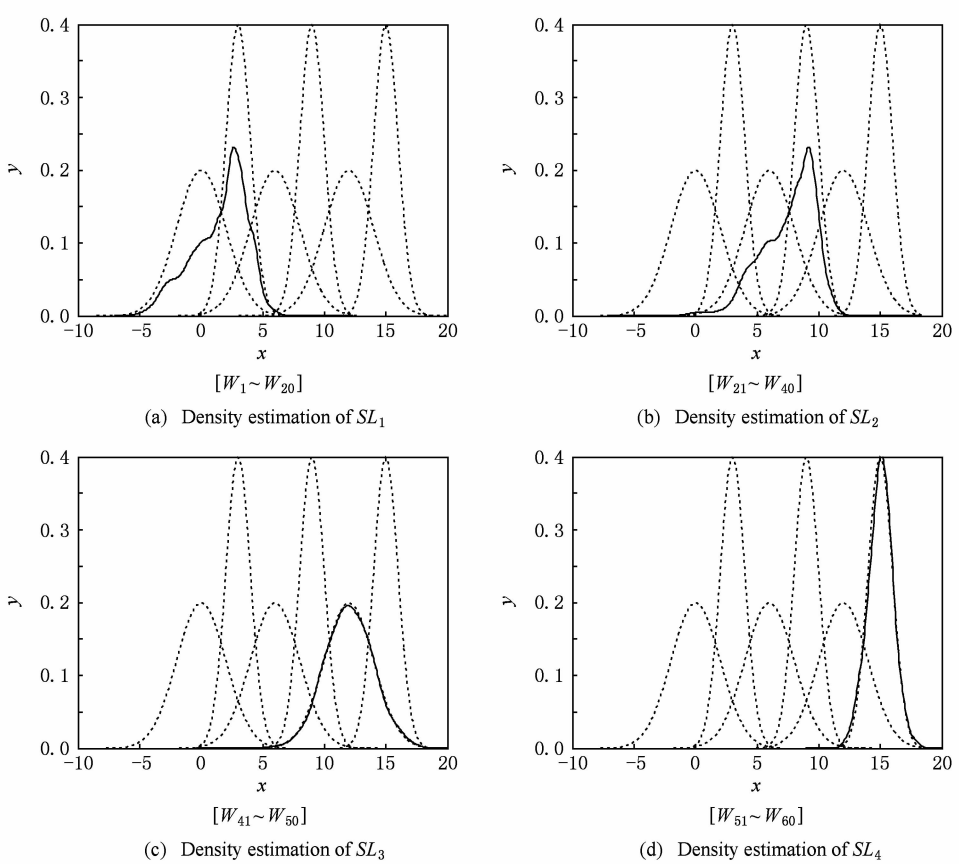


Fig. 11 Estimated probability density functions with each entry in the MMCKDE sequence with λ being 0.085.

图 11 演进数据流 MMCKDE 密度估计(λ=0.085)

③ 实验小结

从 benchmark 数据集实验分析可知,当 MMCKDE 算法用于演进数据流密度估计时,不但可表示某段数据流的概率密度估计,还可表示整段数据流概率密度分布,若采用新的 KL 计算公式,通过设置 λ 值,还可调整新旧数据的重要程度,根据需要将一些数据合并,在节约存储空间的同时,又不影响整个数据流的概率密度估计.且在合并不同核宽的数据窗时,MMCKDE 算法的性能明显优于 SOMKE 算法.因真实数据流大多为演进数据流,故下面对真实数据流进行实验.

2) 真实数据集实验

无参数估计在金融领域应用较多,故本文演进真实数据集①,数据流规模为 10535,每年采集数据样本约 250 个,本文将每 2 年作为一个时间窗,数据流分为 21 个窗,因真实数据集概率密度分布不可知,故本文采用全部样本进行 KDE 估计作为该数据流的数据真实分布,如图 9(b)所示.数据流每个时间窗的概率密度分布如图 12 所示.

首先,用阈值法对演进数据流进行实验,每个 MMCKDE 节点包含 30 个混合核.图 13 表示 λ=0、阈值 θ=1 时,各 MMCKDE 节点及整个数据流的概率密度估计图.表 6 列出了真实数据流各 MMCKDE 节点的 r_j 及 kl_j 值.表 7 为真实演进数据流前 5 个时间窗数据的最大值、最小值、均值及方差.

Table 6 r_j and kl_j in the MMCKDE Sequence Entries over True Evolving Data Stream

表 6 真实演进数据流各 MMCKDE 节点的 r_j 及 kl_j 值

SL_j	r_j	kl_j
SL_1	[1,1]	
SL_2	[2,2]	26.9034
SL_3	[3,3]	5.8810
SL_4	[4,5]	2.1264
SL_5	[6,6]	7.9957
SL_6	[7,7]	1.6379
SL_7	[8,14]	4.6534
SL_8	[15,15]	1.3308
SL_9	[16,16]	1.1829
SL_{10}	[17,18]	36.2910
SL_{11}	[19,20]	2.6798
SL_{12}	[21,21]	1.8262

① http://datamarket.com/data/set/1fo6/公布的 1971—2012 年共 42 年的美国/澳大利亚外汇汇率数据集.

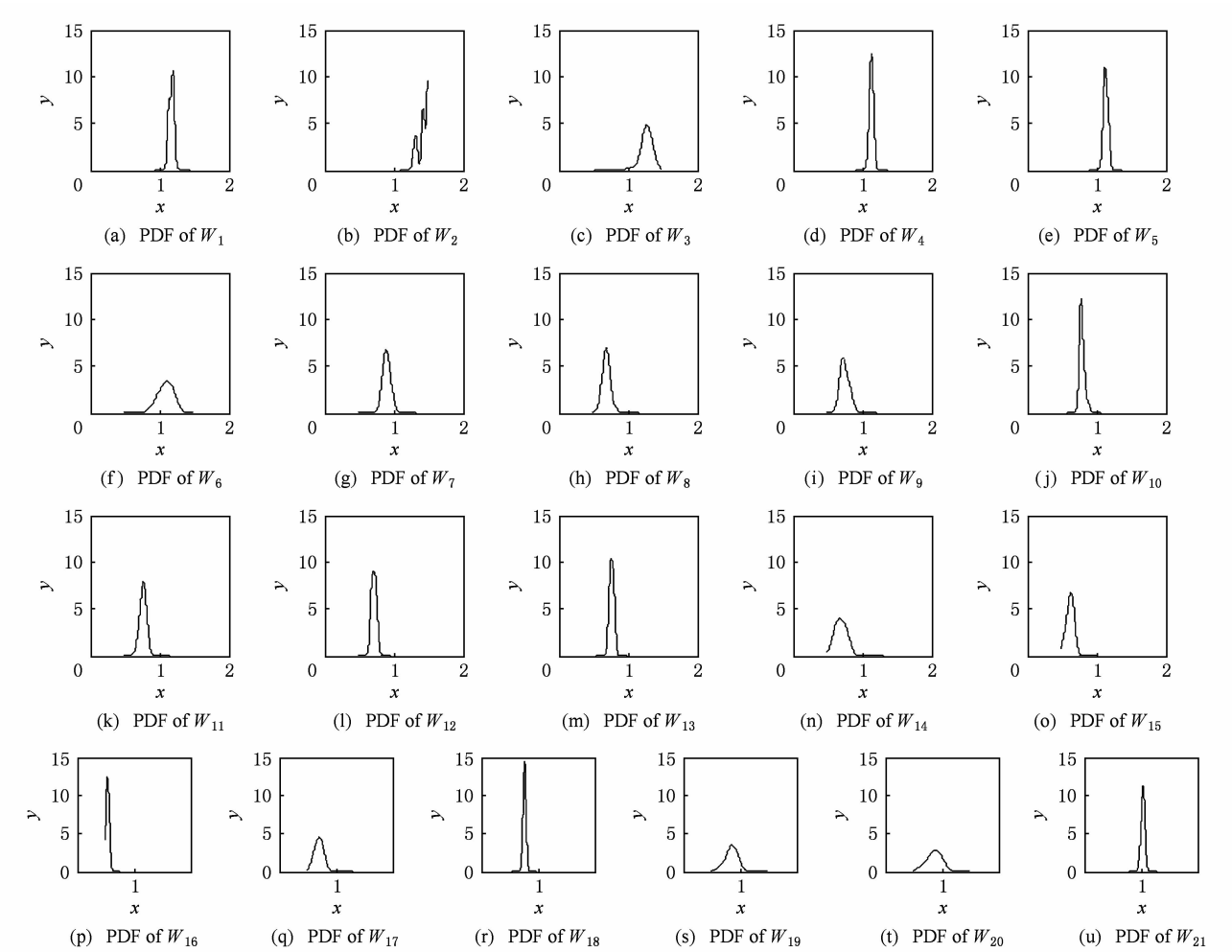


Fig. 12　KDE estimates of 21 windows.

图 12　真实数据流 21 个 MMCKDE 节点概率密度估计图

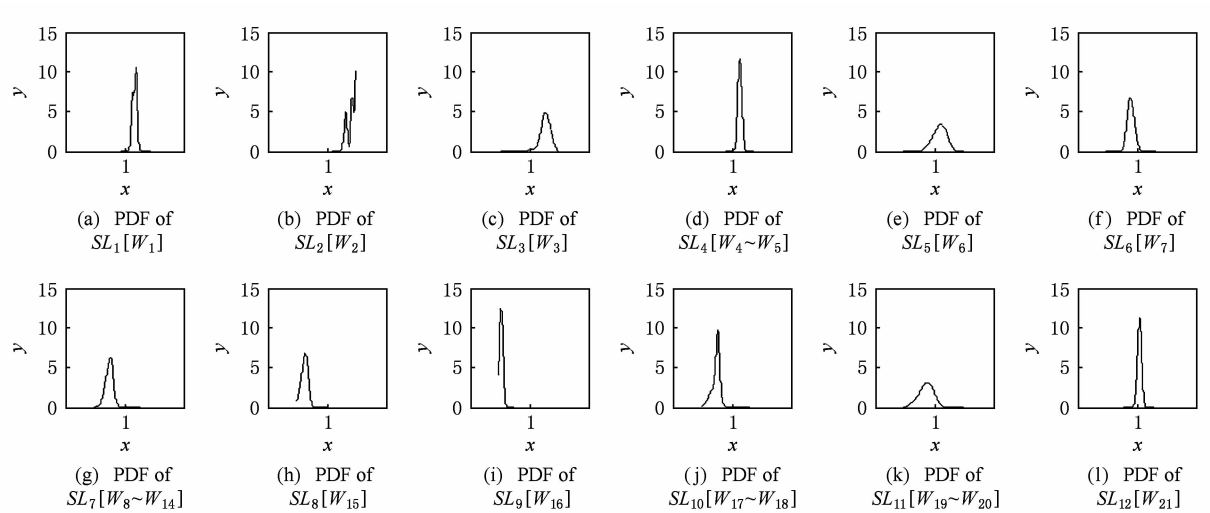


Fig. 13　PDF estimates of 12 entries in the MMCKDE sequence with θ being 1.

图 13　$\theta=1$ 时数据流概率密度估计图

比较图 12 与图 13,以 SL_1 至 SL_4 为例进行分析,图 13 中 SL_4 是时间窗 W_4 与 W_5 的合并,结合表 7 可知,节点合并的年份、汇率均值、方差相近,新节点的产生表示概率密度发生变化,由此可以判断

出汇率稳定及汇率发生明显变化的年份,如 1974—1975 年汇率相对稳定,1971—1973 年间汇率发生明显变化,上述分析可作为数据进一步处理的依据.

用阈值法对演进数据流进行实验后,节点由 21 个降为 12 个,若对存储空间有要求,可采用固定阈值法和固定存储空间法相结合对数据流进行处理.若新节点与前一节点间 KL 距离小于阈值则进行节点合并,若总节点数大于规定的节点数,则距离最近的两节点合并.固定节点为 5,合并后各节点及总概率密度估计示意图如图 14 所示.

Table 7 Maximum, Minimum, Mean, and Standard Deviation of the First 5 Windows

表 7 前 5 个时间窗数据的最大值、最小值、均值及方差

W_j	Maximum	Minimum	Mean	Variance
W_1	1.2732	1.1124	1.1643	0.0319
W_2	1.4885	1.2710	1.4290	0.0695
W_3	1.3655	1.0054	1.2645	0.0655
W_4	1.1860	1.0822	1.1262	0.0222
W_5	1.1814	1.0670	1.1286	0.0269

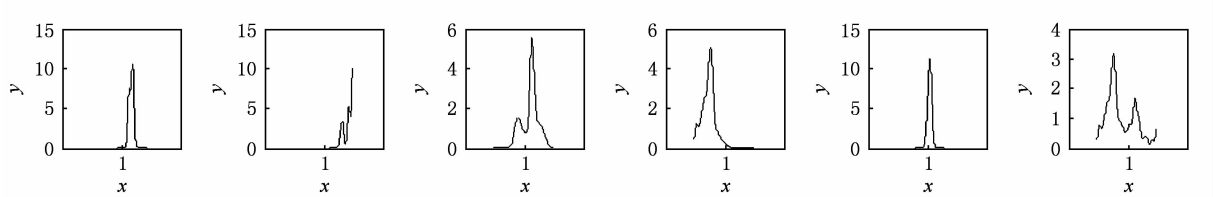


Fig. 14 PDF estimates of 5 entries in the MMCKDE sequence and the whole data stream.

图 14 固定存储空间数据流各节点概率密度估计图

3.4 时间及空间复杂度分析

本文对固定存储空间策略 MMCKDE 算法的时间复杂度和空间复杂度进行分析,因 MMCKDE 算法主要涉及到核合并,故首先对核合并算法进行时间复杂度分析.

3.4.1 核合并算法时间复杂度分析

设输入核规模 n,混合核个数 m,第 i 个聚类中所代表的原始核个数 n_i,T 是算法外层总循环次数,p 是为 $\boldsymbol{t}_j$ 寻优循环次数,q 是为 σ_j 寻优循环次数,S 表示 p,q 外层循环,故时间总运行时间为 $T\left(\sum_{i=1}^{m} n_i S(p+q)+m\times n\right)$.整理后的 $Tn(S(p+q)+m)$.通过实验可知,迭代次数 T,S,p,q 的数值均很小,T 一般在 15 左右,故混合核合并算法的时间复杂度为 $O(n(S(p+q)+m))$,令 $\tilde{m}=S(p+q)+m$,故时间复杂度可简化为 $O(n\tilde{m})$.

3.4.2 数据流密度估计时间复杂度分析

设 l 为 MMCKDE 节点数,N 为整个数据流规模,M 为测试集规模,N_{ec} 为数据窗数据规模.数据流的时间复杂度分为训练时间和预测时间.在训练部分,为每个数据窗训练获得 MMCKDE 节点,训练一个数据窗的时间复杂度为 $O(N_{ec}\tilde{m})$,训练整个数据流的时间复杂度为 $O(N\tilde{m})$.将 N/N_{ec} 个 MMCKDE 节点合并成 k 个 MMCKDE 节点的时间复杂度为 $O((N/N_{ec}-k)\times m\tilde{m})$,因为 $k<<N/N_{ec}$,故时间复杂度可近似为 $O(N/N_{ec}\times m\tilde{m})$,又因为 $m/N_{ec}<<1$,故训练阶段时间复杂度为 $O(N\tilde{m})$.预测阶段时间复杂度为 $O(M\times(lm))$,故整个时间复杂度为 $O(N\tilde{m}+M\times(lm))$.传统的核密度估计法的时间复杂度为 $O(MN)$,因为 $\tilde{m}<<M$,$lm<<N$,故 MMCKDE 算法比传统的核密度估计法速度大大加快.

3.4.3 数据流密度估计空间复杂度分析

MMCKDE 算法所需存储空间为存储 MMCKDE 节点的空间,为 $O(km)$,而传统的核密度估计法所需存储空间为 $O(N)$,故 MMCKDE 算法可以大大节省存储空间.

4 结 论

本文基于混合聚类核与 KL 距离提出了一种新颖的数据流密度估计法 MMCKDE.该方法主要包含 3 方面内容:1)建立 MMCKDE 序列;2)合并 MMCKDE 序列;3)最终概率密度估计.第 1 步操作主要用于使用数目较少的混合聚类核代替数据窗中所有核进行概率密度估计;第 2 步操作主要基于 KL 距离将相似节点进行合并,合并策略有固定存储空间法及固定阈值法 2 种,进一步节省存储空间.最终的数据流概率密度估计,既可估计某一段数据流密度,也可用于估计整个数据流密度.本文使用

Benchmark 数据集及真实数据集对所提出算法进行评估,且与 SOMKE 算法进行比较.实验证明 MMCKDE 算法具有 m 取较小值和不同核宽数据窗合并的特性,使所得概率密度估计能以更紧凑的形式表示,从而适合于实际应用时存储空间有限与更高实时处理的需要.

参 考 文 献

[1] Wei Xiaotao, Huang Houkuan, Tian Shengfeng. An online adaptive network anomaly detection system model and algorithm [J]. Journal of Computer Research and Development, 2010, 47(3): 485-492 (in Chinese)
(魏小涛, 黄厚宽, 田盛丰. 在线自适应网络异常检测系统模型与算法[J]. 计算机研究与发展, 2010, 47(3): 485-492)

[2] DiNardo J, Tobias J L. Nonparametric density and regression estimation [J]. The Journal of Economic Perspectives, 2001, 15(4): 11-28

[3] Chang Jianlong, Cao Feng, Zhou Aoying. Clustering evolving data streams over sliding windows [J]. Journal of Software, 2007, 18(4): 905-918 (in Chinese)
(常建龙, 曹锋, 周傲英. 基于滑动窗口的进化数据流聚类[J]. 软件学报, 2007, 18(4): 905-918)

[4] Domingos P, Hulten G. Catching up with the data: Research issues in mining data streams [C/OL] //Proc of Workshop on Research Issues in Data Mining and Knowledge Discovery. 2001. [2013-01-30]. http://www.cs.cornell.edu/johannes/dmkd2001.htm

[5] Parzen E. On estimation of a probability density function and mode [J]. The Annals of Mathematical Statistics, 1962, 33(3): 1065-1076

[6] Girolami M, He C. Probability density estimation from optimally condensed data samples [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2003, 25(10): 1253-1264

[7] Deng Z H, Chung F L, Wang S T. FRSDE: Fast reduced set density estimator using minimal enclosing ball approximation [J]. Pattern Recognition, 2008, 41(4): 1363-1372

[8] Silverman B W. Density Estimation for Statistics and Data Analysis [M]. London: Chapman & Hall: 1986

[9] Kollios G, Gunopulos D. Efficient biased sampling for approximate clustering and outlier detection in large datasets [J]. IEEE Trans on Knowledge and Data Engineering, 2003, 15(5): 1170-1187

[10] Jones M C, Marron J S, Sheather S J. A brief survey of bandwidth selection for density estimation [J]. Journal of the American Statistical Association, 1996, 91(433): 401-407

[11] Raykar V C, Duraiswami R. Fast optimal bandwidth selection for kernel density estimation [C] //Proc of the 6th SIAM Int Conf on Data Mining. Philadelphia: SIAM, 2006: 524-528

[12] He H, Chen S, Li K, et al. Incremental learning from stream data [J]. IEEE Trans on Neural Network, 2011, 22(12): 1901-1914

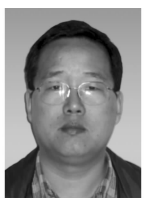
[13] Kullback S, Leibler R A. On information and sufficiency [J]. The Annals of Mathematical Statistics, 1951, 22(1): 79-86

[14] Cao Y, He H B, Man H. SOMKE: Kernel density estimation over data streams by sequences of self-organizing maps [J]. IEEE Trans on Neural Network and Learning Systems, 2012, 23(8): 1254-1268

[15] Blohsfeld B, Heinz C, Seeger B. Maintaining nonparametric estimators over data streams [C] //Proc of the German Conf on Datenbanksysteme in Business, Technologie und Web. Karlsruhe, Baden-Württemberg, German: GI, 2005: 385-404

Xu Min, born in 1980. PhD. Her research interest covers pattern recognition, intelligent computation.

Deng Zhaohong, born in 1981. PhD. Associate professor at the School of Digital Media, Jiangnan University. His research interest covers fuzzy modeling and intelligent computation (dzh666828@yahoo.com.cn).



Wang Shitong, born in 1964. Professor at the School of Digital Media, Jiangnan University. His research interest covers artificial intelligence, pattern recognition and bioinformatics (wxwangst@yahoo.com.cn).

Shi Yingzhong, born in 1970. PhD candidate at the School of Digital Media, Jiangnan University. His research interest covers pattern recognition, intelligent computation (shiyz@wxit.edu.cn).

附录 A.

1) $\bar{\varepsilon}_j$ 推导.

扩展正文中式(8),可得:

$$\bar{\varepsilon}_j = \min_{\beta_j, G_{\sigma_j}(\cdot)} \int_{\mathbf{R}^d} (\beta_j G_{\sigma_j}(\boldsymbol{x}, \boldsymbol{t}_j))^2 \mathrm{d}\boldsymbol{x} - 2\int_{\mathbf{R}^d} (\beta_j G_{\sigma_j}(\boldsymbol{x}, \boldsymbol{t}_j)) \Big(\sum_{i \in S_j} \alpha_i K_{h_i}(\boldsymbol{x}, \boldsymbol{x}_i) \Big) \mathrm{d}\boldsymbol{x} + \int_{\mathbf{R}^d} \Big(\sum_{i \in S_j} \alpha_i K_{h_i}(\boldsymbol{x}, \boldsymbol{x}_i) \Big)^2 \mathrm{d}\boldsymbol{x}, \quad \text{(A1)}$$

式(A1)等号右边第 3 项与所求目标无关,故舍弃,本附录中推导只求前 2 项.

① 求式(A1)等号右边第 1 项:

因本文采用高斯核,其核函数式为 $K(\boldsymbol{u}) = \frac{1}{(\sqrt{2\pi})^d} \exp\Big(-\frac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}\Big)$,且 $\int_{\mathbf{R}^d \mathbf{R}} K(\boldsymbol{u}) \mathrm{d}\boldsymbol{u} = 1$,通过应用式 $\int_{\mathbf{R}^d} k_{\sigma_1}(\boldsymbol{x}, \boldsymbol{t}_1) k_{\sigma_2}(\boldsymbol{x}, \boldsymbol{t}_2) \mathrm{d}\boldsymbol{x} = k_{\sigma_1 + \sigma_2}(\boldsymbol{t}_1 - \boldsymbol{t}_2)$[6-7],可进一步简化式(A1)等号右边第 1 项.

故:

$$\int_{\mathbf{R}^d} (\beta_j G_{\sigma_j}(\boldsymbol{x}, \boldsymbol{t}_j))^2 \mathrm{d}\boldsymbol{x} = \int_{\mathbf{R}^d} \beta_j^2 \frac{1}{(\sqrt{2\pi}\sigma_j)^d} \exp\Big(-\frac{\|\boldsymbol{x} - \boldsymbol{t}_j\|^2}{2\sigma_j^2}\Big) \times \frac{1}{(\sqrt{2\pi}\sigma_j)^d} \exp\Big(-\frac{\|\boldsymbol{x} - \boldsymbol{t}_j\|^2}{2\sigma_j^2}\Big) \mathrm{d}\boldsymbol{x} = \beta_j^2 \frac{1}{(\sqrt{2\pi})^d (2\sigma_j^2)^{\frac{d}{2}}}. \quad \text{(A2)}$$

② 求式(A1)等号右边第 2 项:

$$2\int_{\mathbf{R}^d} (\beta_j G_{\sigma_j}(\boldsymbol{x}, \boldsymbol{t}_j)) \Big(\sum_{i \in S_j} \alpha_i K_{h_i}(\boldsymbol{x}, \boldsymbol{x}_i) \Big) \mathrm{d}\boldsymbol{x} = 2\sum_{i \in S_j} \int_{\mathbf{R}^d} \beta_j \alpha_i \frac{1}{(\sqrt{2\pi}\sigma_j)^d} \exp\Big(-\frac{\|\boldsymbol{x} - \boldsymbol{t}_j\|^2}{2\sigma_j^2}\Big) \times \frac{1}{(\sqrt{2\pi}h_i)^d} \exp\Big(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2h_i^2}\Big) \mathrm{d}\boldsymbol{x} = 2\sum_{i \in S_j} \frac{\beta_j \alpha_i}{(\sqrt{2\pi})^d (h_i^2 + \sigma_j^2)^{\frac{d}{2}}} \times \exp\Big(-\frac{1}{2}\Big(\frac{\|\boldsymbol{t}_j - \boldsymbol{x}_i\|^2}{(h_i^2 + \sigma_j^2)}\Big)\Big). \quad \text{(A3)}$$

将式(A2)、式(A3)代入式(A1),可得正文中优化式(9).

2) 定理 1 证明.

证明. 定义新的目标式 $\chi = \sum_{i=1}^{m} \chi_i$,其中:

$$\chi_j = \int_{\mathbf{R}^d} \Big(F_j(\boldsymbol{x}) - \frac{1}{Z_j} \Big(\sum_{i \in S_j} \alpha_i K(\boldsymbol{x}) \Big) \Big)^2 \mathrm{d}\boldsymbol{x}; \quad \text{(A4)}$$

$$Z_j \chi_j = \frac{1}{Z_j} \int_{\mathbf{R}^d} \Big(\sum_{i \in S_j} \alpha_i K(\boldsymbol{x}) \Big)^2 \mathrm{d}\boldsymbol{x} + Z_j \int_{\mathbf{R}^d} (F_j(\boldsymbol{x}))^2 \mathrm{d}\boldsymbol{x} - 2\int_{\mathbf{R}^d} F_j(\boldsymbol{x}) \sum_{i \in S_j} \alpha_i K(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \frac{1}{Z_j} \int_{\mathbf{R}^d} \Big(\sum_{i \in S_j} \alpha_i K(\boldsymbol{x}) \Big)^2 \mathrm{d}\boldsymbol{x} + \sum_{i \in S_j} \alpha_i \int_{\mathbf{R}^d} (F_j(\boldsymbol{x}))^2 \mathrm{d}\boldsymbol{x} - 2\int_{\mathbf{R}^d} F_j(\boldsymbol{x}) \sum_{i \in S_j} \alpha_i K(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \quad \text{(A5)}$$

比较式(A5)与正文中式(15),除第 1 项与 $F_j(\boldsymbol{x})$无关的常数项外,其余项均相同,故式(A4)、式(A5)和正文中式(15)的 $F_j(\boldsymbol{x})$最优值相同.

比较正文中式(8)与式(A4),得出结论:

$$\tilde{F}_j(\boldsymbol{x}) = Z_j F_j(\boldsymbol{x}).$$

证毕.