基于 LLE 和高斯混合模型的时间序列聚类

杨秋颖,翁小清
(河北经贸大学 信息技术学院,河北 石家庄 050061)

摘 要:聚类分析是常见的数据挖掘方法,时间序列数据挖掘可以将海量时序信息转化成有组织的知识。由于时间序列具有高维度、非线性等特点,大多数聚类算法无法直接应用在原始时间序列数据上并取得令人满意的效果。研究如何在维数约简的同时尽可能多地保留数据的内蕴特征,识别代表知识的真正有趣的模式,具有重要意义。现有大多数时间序列聚类算法没有考虑数据集的局部结构,而数据集的局部结构对聚类性能有较大影响。提出一种基于局部线性嵌入(Locally Linear Embedding,LLE)和高斯混合模型(Gaussian Mixture Model,GMM)的时间序列聚类算法。首先从保留数据集局部结构的角度,使用 LLE 将每个高维时间序列样本表示为其 k 近邻的线性组合,并在低维空间进行重构,在保持数据集局部几何结构的同时实现维数约简;然后使用 GMM 从概率分布的角度进行聚类分析。与已有方法相比,该方法在单变量时间序列聚类上具有更优的效果。

关键词:局部线性嵌入;高斯混合模型;流形学习;时间序列聚类;深度学习
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2022)08-0033-09
doi:10.3969/j.issn.1673-629X.2022.08.006

Time Series Clustering Based on LLE and Gaussian Mixture Model

YANG Qiu-ying,WENG Xiao-qing
(School of Information Technology,Hebei University of Economics & Business,Shijiazhuang 050061,China)

Abstract:Cluster analysis is a common data mining method. Time series data mining can transform massive time series information into organized knowledge. In view of the high dimensionality,nonlinearity and other characteristics of time series,most clustering algorithms cannot be directly applied to the original time series data and achieve satisfactory results. It is important to study how to retain as many inherent features of the data as possible while reducing the dimensionality,and to identify interesting patterns that represent knowledge. Most of the existing nonlinear dimensionality reduction methods reduce the dimension from the perspective of preserving the global features and ignore the local linear features of the data set. A time series clustering algorithm based on LLE and GMM is proposed. Firstly,from the perspective of preserving local features,LLE is used to represent each sample of high-dimensional time series as a linear combination of its k-nearest neighbors and reconstruct it in the low-dimensional space,and dimension reduction is achieved while preserving the local geometric structure of data. Then,GMM is used to perform cluster analysis from the perspective of probability distribution. Compared with the existing methods,the proposed algorithm can obtain better clustering effect in univariate time series.

Key words:local linear embedding;Gaussian mixture model;manifold learning;time series clustering;deep learning

0 引 言

时间序列(TS)是从均匀的时间间隔和给定的采样率下测量收集的有序数据集,其研究遍及金融、医学、轨迹分析和人体动作分段等多个领域。时间序列聚类[1]是在没有任何先验知识的情况下分析大量时间序列数据的有效方法,其目的以某种方式将给定的数据集划分为一组不重叠的集群,从而揭示数据的底层结构。在进行聚类时合适的维数约简和相似性度量对聚类效果有重大影响[2],但由于时间序列高维,高冗余以及存在非线性结构等特点,将传统的聚类算法直接用于此类数据时往往无法取得满意的效果。

维数约简根据是否存在变换矩阵,可分为线性和非线性两种。多维尺度变换[3]、主成分分析[4]等线性方法默认先进行投影变换,然后找到一个使其目标最大化的低维空间;但现实中绝大部分时间序列是非线性的,线性方法在应用时存在局限性。非线性降维方

收稿日期:2021-09-13 **修回日期**:2022-01-14
基金项目:河北经贸大学科研基金(2019QR21)
作者简介:杨秋颖(1995-),女,硕士研究生,CCF 会员(E1915G),研究方向为数据科学与大数据计算;翁小清,博士,教授,研究方向为机器学习、数据挖掘。

法[5]有核方法、神经网络和流形学习等,局部线性嵌入(Locally Linear Embedding, LLE)[6]是一种重要的流形学习方法。流形学习认为采样数据是由低维流形映射到高维空间得到的,其本质是从原始的高维数据中寻找产生数据的内在流形,并求出相应的嵌入映射。LLE 假设采样数据分布在一个潜在的流形上,而流形的局部可以近似为欧氏空间,具有线性结构,故任意一点可以表示为其 k 近邻的线性组合,并能够在低维流形进行重构。LLE 将高维的非线性结构映射到低维空间的同时很好地保留了其内蕴特征。

针对时间序列非线性和维度高的特点,该文提出一种基于 LLE 和高斯混合模型(Gaussian Mixture Model, GMM)的时间序列聚类算法 LLE_GMM。首先从保留数据集局部结构的角度,使用 LLE 将每个高维时间序列样本表示为其 k 近邻的线性组合,并在低维空间进行重构,在保持数据集局部几何结构的同时实现维数约简;然后使用 GMM 从概率分布的角度进行聚类分析。将 LLE_GMM 算法与已有的非深度学习和深度学习算法进行了比较,在 36 个数据集上的实验结果表明,该方法对单变量时间序列具有更好的聚类效果。

1 背景和相关工作

狭义上的时间序列(TS)通常是一个过程的观察结果,其值是从均匀的时间间隔和给定的采样率下测量收集的,用 $\{X_i\}_{i=1}^{N}$, $X_i \in R^{m\times n}$ 表示,其中 N 为样本个数, m 、n 分别为观测值个数以及变量个数。当 $n=1$ 时,称为单变量时间序列;当 $n \geqslant 2$ 时,称为多变量时间序列[7]。该文所提方法针对单变量时间序列。

1.1 局部线性嵌入

局部线性嵌入(Locally Linear Embedding, LLE)[6]是重要的非线性维数约简方法,关注于降维时保持样本的局部拓扑结构。LLE 将每个高维时间序列 x_i 表示为其 k 近邻的线性加权组合 $x_i=\sum_j w_{ij}x_j$,求解出对应的权重系数向量 $\boldsymbol{W}_i$;权重系数和 k 近邻关系在投影前后不变,从而根据线性加权表示解得对应的低维嵌入 $y_i=\sum_j w_{ij}y_j$。

LLE 算法的具体步骤为:

(1)寻找每个样本点 x_i 的 k 近邻的集合。

(2)计算 x_i 的局部重建权值向量 $\boldsymbol{W}_i=\dfrac{\boldsymbol{Z}_i^{-1}\boldsymbol{1}_k}{\boldsymbol{1}_k^K\boldsymbol{Z}_i^{-1}\boldsymbol{1}_k}$,其中 $\boldsymbol{1}_k$ 是 k 维全 1 向量, $\boldsymbol{Z}_i=(x_i-x_j)^{\mathrm{T}}(x_i-x_j)$, x_j 是 x_i 的第 j 近邻; w_{ij} 描述流形局部的几何结构,LLE 认为 w_{ij} 在降维前后保持不变。

(3)求低维嵌入 Y。计算 x_i 在其低维空间的嵌入点 y_i,使其重构的代价函数 $\varphi(Y)$ 最小,即最小化式(1):

$$\min_Y \varphi(Y)=\min_Y \sum_i y_i-\sum_j w_{ij}y_j^2 \tag{1}$$

这一优化问题可以通过对式(2)进行特征值分解得到。

$$M=(I-W)(I-W)^{\mathrm{T}} \tag{2}$$

一般的, M 的第一个最小特征值为 0,不能反映数据特征,故选 M 的第 2 到 $d+1$ 个特征值对应的特征向量,即低维嵌入 $Y=\{y_2,\cdots,y_{d+1}\}$。

1.2 高斯混合模型

高斯混合模型(GMM)假设数据集是有限个高斯分布的线性混合,每个高斯分布对应一个类。具体地,给定类个数 C,对于给定的样本 y_i,GMM 的概率密度函数定义为:

$$p(y_i)=\sum_{c=1}^{C} w_c\cdot p(\boldsymbol{y}_i \mid \mu_c,\boldsymbol{\Sigma}_c) \tag{3}$$

式中, $p(\boldsymbol{y}_i \mid \mu,\boldsymbol{\Sigma})$ 是以 μ 为均值, $\boldsymbol{\Sigma}$ 为协方差矩阵的标准正态分布; w_c 为第 c 个多元高斯分布在混合模型中的权重,且有 $\sum_{c=1}^{C} w_c=1$。

用 EM(Expectation Maximization)算法估计 GMM 参数。其基本步骤如下:

(1)根据给定的 C 值,随机初始化每个簇的高斯分布参数(均值和方差)以及权重向量 w。

(2)E 步:计算数据点 x_i 对每个簇的隶属度 $E[Z_{ic}]$。隶属度越大,样本由该分模型生成的概率越大。隶属度公式如式(4)和式(5)所示:

$$E[Z_{ic}]=\frac{p(y_i \mid \mu_c,\Sigma_c)}{\sum_{c=1}^{C} p(y_i \mid \mu_c,\Sigma_c)} \tag{4}$$

$$p(\boldsymbol{y} \mid \mu_c,\boldsymbol{\Sigma}_c)=\frac{1}{\sqrt{2\pi}\boldsymbol{\Sigma}_c}\exp\left(-\frac{(\boldsymbol{y}-\mu_c)^2}{2\Sigma_c^2}\right) \tag{5}$$

(3)M 步:用第(2)步计算得到的所有点对每个分模型 Z_c 的隶属度更新模型参数,如式(6)~式(8)所示:

$$\mathrm{new}_{\mu c}=\frac{\sum_{i=1}^{m} E[Z_{ic}]y_i}{\sum_{i=1}^{m} E[Z_{ic}]} \tag{6}$$

$$\mathrm{new}_{\Sigma_c}=\frac{\sum_{i=1}^{m} E[Z_{ic}](y_i-\mathrm{new}_{\mu_c})(y_i-\mathrm{new}_{\mu_c})^{\mathrm{T}}}{\sum_{i=1}^{m} E[Z_{ic}]} \tag{7}$$

$$w_c=\frac{\sum_{i=1}^{m} E[Z_{ic}]}{m} \tag{8}$$

(4)循环执行(2)和(3)步,计算对数似然函数直

到收敛。

GMM 使用后验概率不断更新各个分模型的参数,最终得到 MTS 样本对各个类别的隶属度,从概率分布角度进行聚类分析。

1.3　相关工作

时间序列聚类大致可以分为基于实例、基于特征和基于模型的方法三种[8]。

基于实例的方法中,Azencott 等[9]将基于图的拉普拉斯谱聚类与模拟退火相结合研究时间序列间的互信息,自动生成最优的时间序列聚类,但该方法只是适用于等长的有限数据集。考虑时间序列的非线性以及滞后问题,张贝贝等[10]将 Copula 函数引入识别动态相关结构的相似性度量。Guo 等[11]推广了基于核的模糊 c 均值聚类算法,在动态时间对准核(DTAK)中嵌入非线性时间对准使得基于核的模糊 c 均值可以用于可变长度的序列。

基于特征的方法中,Chandereng 等[12]考虑时间的滞后性影响时间序列的相似性,提出了一种滞后惩罚加权相关(Lag Penalized Weighted Correlation,LPWC)的聚类相似度度量方法,用于对随着时间推移表现出密切相关行为的时间序列进行分组。针对长度比较短且存在相位差的时间序列,Yang 等[13]提出一种 Shape-Distance Ratio (SDR)的相似性度量方法并结合 k-Medoids (PAM)分区聚类算法实现时间序列聚类。Euan 等[14]将谱理论与层次聚类相结合,提出层次谱合并(HSM)时间序列聚类算法。Duan 等[15]用趋势滤波对时间序列进行最优分割和模糊信息粒化将原始数据转为粒状时间序列,提出基于线性模糊信息粒的动态时间扭曲(LFIG_DTW)距离的分层聚类方法,LFIG_DTW 算法不仅可以检测距离的增减趋势,还可以检测距离的变化周期和变化速率。Caiado 等[16]提出一种新的非参数的用于描述和比较长时间序列大集合的频域方法。Wang 等[17]针对不等长区间值时间序列的聚类问题提出 BRDTW 算法。

Wang 等[18]提出时间序列的稀疏子空间聚类算法(Sparse Subspace Clustering,SSC),利用稀疏表示构造相似度矩阵再进行光谱聚类,将其运用到电影票房研究问题。稀疏编码字典学习中数据样本与字典原子的长度不一致以及存在时间延迟的问题,Yazdi 等[19-20]提出基于非线性时间不变性 kSVD (twi-ksvd)的稀疏编码字典学习时间序列聚类算法。

为了提取时间序列的形状特征,Zhang 等[21]结合 shapelet 学习、shapelet 正则化、光谱分析和伪标记的优点,扩展了监督式 shapelet 学习模型来处理未标记的时间序列数据,提出了无监督显著子序列学习(Unsupervised Salient Subsequence Learning,USSL)。Xiao 等[22]结合时间特征网络和注意力 LSTM 网络提出一种鲁棒时序特征网络(RTFN),将基于残差网络和 multi-head 卷积神经网络的时间特征网络用于提取序列的时态特征,attentional LSTM 网络进一步提取时序中的 shapelets 特征,并将其用于分类和聚类。

在基于模型的方法中,Corduas 等[23]针对传统的 ARIMA 模型中 one-step-ahead 预测函数可能导致对模型的错误描述,提出 h-step-ahead 预测函数,用 h-step-ahead 预测误差的参数的欧氏距离平方和度量时间序列的相似性。

基于监督学习的深度学习算法可以学习数据的隐藏特征。但现实中的时间序列大部分没有标签信息,因此基于监督学习的深度学习算法无法直接用于时间序列聚类。Xie 等[24]提出 Deep Embedded Clustering 算法,以 self-learning 的方式定义聚类损失,同时更新网络和聚类中心的参数。然而聚类损失并不能保持局部结构,会导致嵌入空间的破坏。为此 Guo 等[25]使用 under-complete 的自动编码器来学习嵌入特征和保持数据生成分布的局部结构,提出了 Improved Deep Embedded Clustering 算法。

Sai 等[26]提出深度时间聚类(Deep Temporal Clustering,DTC),采用 CNN 自动编码器与 BI-LSTM 聚类层学习聚类表示。通过测量预测结果与目标分布之间的 KL 散度来设计聚类层;但直接转矩控制的性能很大程度上取决于编码器的能力,根据表示学习计算的预测分布在用来计算目标分布时存在不稳定性。为提高编码器能力,Ma 等[27]将时间重构和 K-Means 聚类集成到 seq2seq 模型中,提出了时间序列辅助分类任务的伪样本生成策略,提高了编码器的能力。此外,Fortuin 等[28]结合自组织映射(SOM)、变分自编码器和 Markov 模型,提出一种可解释离散表示学习。McConville 等[29]采用流形方法提取特征,对重嵌入空间进行浅聚类。Ding 等[2]将卷积神经网络在同一方向的输出变化次数转化为时间序列的相似性,通过优先收集少量的高相似度数据来创建标签,使用基于卷积神经网络的分类算法辅助聚类。

上述大多数方法或是未考虑时间序列的非线性结构,或是从保留全局特征的角度进行降维,没有考虑数据集的局部结构,而数据集的局部结构对聚类效果有较大影响;此外上述大多数方法从距离角度度量时间序列的相似性,该文在保留时间序列局部特征的基础上,使用 GMM 从概率分布角度进行聚类,提高了聚类性能。

2　基于 LLE 和 GMM 的聚类算法

基于 LLE 和 GMM 的聚类算法包括两步骤:首先

从保留数据集局部结构的角度,使用 LLE 将每个高维时间序列样本表示为其 k 近邻的线性组合,并在低维空间进行重构,在保持数据集局部几何结构的同时实现维数约简;然后使用 GMM 从概率分布的角度进行聚类分析。算法的主要步骤如下:

算法 1:LLE_GMM(X,C,k,d)。

输入:时间序列数据集。$X = \{x_1,x_2,\cdots,x_N,x_i \in R^m\}$,聚类个数 C ,近邻个数 k ,嵌入维数 d 。

输出:聚类结果。

Step1:对数据集 X 使用 PCA 算法去除噪声和冗余;

Step2:对任意 x_i 的 k 个最近邻点 x_j ,构造近邻集合;

Step3:计算 x_i 的局部重建权值向量 $\boldsymbol{W}_i = \dfrac{\boldsymbol{Z}_i^{-1}\mathbf{1}_k}{\mathbf{1}_k^K\boldsymbol{Z}_i^{-1}\mathbf{1}_k}$,组成权重系数矩阵 $\boldsymbol{W}$;

Step4:构造矩阵 $\boldsymbol{M} = (\boldsymbol{I}-\boldsymbol{W})(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}$,计算 $\boldsymbol{M}$ 的前 $d+1$ 个特征值和对应的特征向量,则低维嵌入为 $Y = \{y_2,\cdots,y_{d+1}\}$;

Step5:初始化高斯混合模型参数 (w,μ,Σ) 开始迭代;

Step6:E-step,求每个样本对每个类别的概率;

Step7:M-step,优化 E-step 的模型参数得到新的参数 (w,μ,Σ) ;

Step8:重复 E-step 和 M-step,直到参数收敛或是达到最大迭代次数;

Step9:用训练好的 GMM 模型进行聚类。

上述算法分为降维和模型训练两个部分。对于时间序列数据集 $X = \{x_1,x_2,\cdots,x_N,x_i \in R^m\}$, N 为样本总数, m 为输入样本维数。步骤 1 中使用 PCA 预处理的时间复杂度为 $O(Nm^2)$;步骤 2-5 为 LLE 降维,其中 k 近邻搜索的复杂度是 $O(mN^2)$,构造权重系数矩阵的时间复杂度是 $O(mNk^3)$,求解低维嵌入的时间复杂度是 $O(dN^2)$, d 为嵌入维数;步骤 5-9 是构建高斯混合模型聚类阶段,时间复杂度与迭代次数有关,每次迭代过程分为 E-step 和 M-step。E-step 计算样本的所属类别概率的时间复杂度为 $O(NC)$, C 为类别个数;M-step 更新参数 w,μ 的时间复杂度为 $O(k)$;计算协方差 Σ 的时间复杂度为 $O(NCd^2)$,故每次迭代的时间复杂度为 $O(NC(d^2+1)+C)$;当迭代次数为 h 时,算法整体时间复杂度为 $O(Nm^2+mN^2+mNk^3+dN^2+hNCd^2)$ 。

3 实验结果与分析

在 36 个来自 UCR[30] 数据库的时间序列数据集上用 Rand 指数对聚类性能进行评估。用 Matlab 2019b 编写了所有程序,并在方正计算机(内存 16 GB,CPU 3.30 GHz,Windows 7 操作系统)上实现。

3.1 数据集描述

采用来自 UCR 数据库的时间序列数据集,数据集都具有非随机结构且提供聚类基准,即标签信息。表 1 列出了 36 个数据集的主要特征,包括序号、样本集名称、样本总数、样本长度和类别个数。这些数据集涉及工业、图像识别、人体行为识别、医学和化学计量学等领域。

表 1 数据集概要情况

序号	数据集名称	样本总数	样本长度	类别个数
1	ArrowHead	211	251	3
2	Beef	60	470	5
3	BeetleFly	40	512	2
4	BirdChicken	40	512	2
5	Car	120	577	4
6	ChlorineConcentration	4 307	166	3
7	Coffee	56	286	2
8	DiatomSizeReduction	322	345	4
9	DistalPhalanxOutlineAgeGroup	539	80	3
10	DistalPhalanxOutlineCorrect	876	80	2
11	ECG200	200	96	2
12	ECGFiveDays	884	136	2
13	GunPoint	200	150	2
14	Ham	214	431	2
15	Herring	128	512	2
16	Lightning2	121	637	2
17	Meat	120	448	3
18	MiddlePhalanxOutlineAgeGroup	554	80	3
19	MiddlePhalanxOutlineCorrect	891	80	2
20	MiddlePhalanxTW	553	80	6
21	MoteStrain	1 272	84	2
22	OSULeaf	442	427	6
23	Plane	210	144	7
24	ProximalPhalanxOutlineAgeGroup	605	80	3
25	ProximalPhalanxTW	605	80	6
26	SonyAIBORobotSurface1	621	70	2
27	SonyAIBORobotSurface2	980	65	2
28	SwedishLeaf	1 125	128	15
29	Symbols	1 020	398	6
30	ToeSegmentation1	268	277	2
31	ToeSegmentation2	166	343	2
32	TwoPatterns	5 000	128	4
33	TwoLeadECG	1 162	82	2
34	Wafer	7 164	152	2
35	Wine	111	234	2
36	WordSynonyms	905	270	25

3.2 评价标准

为使文中算法与已有算法具有对比性,采用常见的外部方法 Rand 指数[31](RI)评价 LLE_GMM 的聚类效果。

$$\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{9}$$

式中,TP 表示属于同类的样本的预测标签相同,FN 表

示属于同类的样本的预测标签不同,FP 表示属于不同类的样本的预测标签相同,TN 表示不属于同一类的样本的预测标签也不同。Rand 指数取值为[0,1],是正向指标,当原有的标签信息与预测结果完全一致时,RI=1。

3.3　性能比较

为检验 LLE_GMM 算法性能,将其与 10 种已有算法进行 Rand 指数(RI)比较,10 种算法分为两个类型:基于非深度学习以及基于深度学习。其中非深度学习的分为基于实例和基于特征两种,基于特征的聚类算法又分为基于结构和基于形状两个方面。

表 2 给出了用 5 种基于非深度学习的方法以及 LLE_GMM 在 36 个数据集上进行聚类的 RI 值,六种方法的最高 RI 值在表 2 中加粗显示。表 2 中第 1 列的序号对应表 1 中的数据集,第 2 列至第 6 列分别为 KSC[32]、NDFS[33]、RSFS[34]、kshape[35]、USSL[21]的 RI 值;最后一列给出了 LLE_GMM 的 RI 值以及对应的近邻个数 k 和嵌入维数 d。

表 2 的倒数第 2 行 Avg 给出各种方法的平均 RI 值,可以看出 LLE_GMM 在 36 个数据集的平均 RI 为 0.802 0,在六种非深度学习算法中取得最优结果。表 2 的最后一行 Win 给出各算法在 36 个数据集上取得的最优 RI 的个数,可以看出 LLE_GMM 在 23 个数据集上取得最优结果。

表 3 给出了用 5 种基于深度学习的方法以及 LLE_GMM 在 36 个数据集上进行聚类的 RI 值,这六种方法的最高 RI 值同样加粗显示。表 3 中第 1 列的序号对应表 1 中的数据集,第 2 列至第 6 列分别为 SOM-VAE[28]、N2D[29]、IDEC[25]、DTCR[27]和 TSC_CNN[2]的 RI 值;最后一列给出了 LLE_GMM 的 RI 值以及对应的近邻个数 k 和嵌入维数 d。

表 3 的倒数第 2 行 Avg 给出各种方法的平均 RI 值,LLE_GMM 在 36 个数据集的平均 RI 在六种算法中同样取得最优结果。表 3 的最后一行 Win 给出各算法在 36 个数据集上取得的最优 RI 的个数,可以看出 LLE_GMM 在 18 个数据集上取得最优结果。

深度学习算法在执行时会一定程度上受到算力的限制,LLE_GMM 在不依赖硬件设施的同时可以取得不差于深度学习算法的效果。

表 2　与非深度学习方法的 RI 比较

序号	KSC	NDFS	RSFS	Kshape	USSL	LLE_GMM(k, d)
1	0.725 4	0.738 1	0.710 8	0.725 4	0.715 9	**0.816 3**(**31**, **6**)
2	0.705 7	0.703 4	0.697 5	0.540 2	0.696 6	**0.811 3**(**25**, **15**)
3	0.605 3	0.557 9	0.651 6	0.605 3	0.810 5	**0.857 7**(**17**, **38**)
4	0.731 6	0.731 6	0.663 2	0.663 2	**0.810 5**	0.703 9(29, 9)
5	0.689 8	0.626 0	0.670 8	0.702 8	0.734 5	**0.760 1**(**6**, **9**)
6	0.525 6	0.522 5	0.531 6	0.411 1	0.499 7	**0.849 0**(**2**, **23**)
7	**1.000 0**	**1.000 0**	**1.000 0**	**1.000 0**	**1.000 0**	0.929 9(23, 5)
8	**1.000 0**	0.958 3	0.916 7	**1.000 0**	**1.000 0**	**1.000 0**(**10**, **4**)
9	0.653 5	0.623 9	0.653 9	0.602 0	0.665 0	**0.754 7**(**30**, **1**)
10	0.523 5	0.538 3	0.532 7	0.525 2	0.596 2	**0.673 1**(**36**, **21**)
11	0.631 5	0.631 5	0.691 6	0.701 8	0.728 5	**0.787 7**(**37**, **14**)
12	0.525 7	0.557 3	0.595 3	0.502 0	0.834 0	**0.861 7**(**12**, **4**)
13	0.497 1	0.510 2	0.499 4	0.627 8	**0.725 7**	0.678 4(16, 23)
14	0.536 2	0.536 2	0.512 7	0.531 1	**0.639 3**	0.625 6(72, 5)
15	0.494 0	0.516 4	0.515 1	0.496 5	**0.619 0**	0.555 6(84, 76)
16	0.626 3	0.537 3	0.526 9	0.654 8	**0.695 5**	0.607 7(6, 3)
17	0.672 3	0.663 5	0.665 7	0.657 5	0.774 0	**0.967 7**(**70**, **6**)
18	0.536 4	0.535 0	0.547 3	0.510 5	0.580 7	**0.746 3**(**95**, **39**)
19	0.501 4	0.504 7	0.514 9	0.511 4	**0.663 5**	0.577 1(31, 10)
20	0.818 7	0.191 9	0.806 2	0.621 3	0.792 0	**0.849 3**(**5**, **2**)
21	0.663 2	0.605 3	0.616 8	0.605 3	0.810 5	**0.820 1**(**5**, **36**)
22	0.571 4	0.562 2	0.566 5	0.553 8	0.655 1	**0.772 0**(**8**, **16**)
23	0.960 3	0.895 4	0.931 4	0.990 1	**1.000 0**	0.987 3(16, 37)

续表 2

序号	KSC	NDFS	RSFS	Kshape	USSL	LLE_GMM(k, d)
24	0.530 5	0.546 3	0.538 4	0.561 7	0.793 9	**0.831 3**(**100**, **3**)
25	0.605 3	0.605 3	0.521 1	0.521 1	0.728 2	**0.873 8**(**24**, **9**)
26	0.772 6	0.772 1	0.792 8	0.808 4	0.810 5	**0.946 7**(**31**, **19**)
27	**0.903 9**	0.886 5	0.894 8	0.561 7	0.857 5	0.750 5(21, 3)
28	0.492 3	0.550 0	0.503 8	0.533 3	0.854 7	**0.928 4**(**100**, **6**)
29	0.898 2	0.856 2	0.906 0	0.837 3	0.920 0	**0.957 4**(**20**, **8**)
30	0.500 0	0.587 3	0.496 8	0.614 3	**0.671 8**	0.651 2(91, 91)
31	0.525 7	0.596 8	0.582 6	0.525 7	0.677 8	**0.686 9**(**132**, **2**)
32	**0.858 5**	0.853 0	0.858 8	0.804 6	0.831 8	0.771 5(4, 22)
33	0.546 4	0.632 8	0.563 5	0.824 6	0.862 8	**0.977 9**(**8**, **7**)
34	0.492 5	0.526 3	0.492 5	0.492 5	0.824 6	**0.922 6**(**20**, **19**)
35	0.500 6	0.512 3	0.503 3	0.500 1	**0.898 5**	0.638 0(70, 14)
36	0.872 7	0.876 0	0.881 7	0.784 4	0.854 0	**0.903 4**(**6**, **5**)
Avg	0.658 2	0.640 2	0.654 3	0.641 9	0.767 6	0.802 0
Win	4	1	1	2	11	23

表 3 与深度学习方法的 RI 比较

序号	SOM-VAE	N2D	IDEC	DTCR	TSC_CNN	LLE_GMM(k, d)
1	0.648 7	0.645 2	0.621 0	0.686 8	0.734 9	**0.816 3**(**31**, **6**)
2	0.671 1	0.678 5	0.627 6	0.804 6	0.712 6	**0.811 3**(**25**, **15**)
3	—	—	0.605 3	**0.900 0**	—	0.857 7(17, 38)
4	0.519 2	0.533 3	0.478 9	0.810 5	**1.000 0**	0.703 9(29, 9)
5	0.648 8	0.706 4	0.687 0	0.750 1	**0.794 3**	0.760 1(6, 9)
6	0.529 6	0.534 2	0.535 0	0.535 7	0.534 0	**0.849 0**(**2**, **23**)
7	0.725 3	0.864 9	0.576 7	0.926 8	0.928 6	**0.929 9**(**23**, **5**)
8	0.863 7	0.943 1	0.734 7	0.968 2	0.979 2	**1.000 0**(**10**, **4**)
9	0.660 9	0.606 7	0.778 6	0.782 5	**0.783 2**	0.754 7(30, 1)
10	0.499 5	0.499 8	0.533 0	0.607 5	0.603 1	**0.673 1**(**36**, **21**)
11	0.623 1	0.638 6	0.623 3	0.664 8	0.664 8	**0.787 7**(**37**, **14**)
12	—	—	0.511 4	**0.963 8**	—	0.861 7(12, 4)
13	0.497 6	0.497 4	0.497 4	0.639 8	0.572 2	**0.678 4**(**16**, **23**)
14	0.513 4	0.513 4	0.495 6	0.536 2	0.516 0	**0.625 6**(**72**, **5**)
15	0.497 1	0.496 0	0.509 9	**0.575 9**	0.514 2	0.555 6(84, 76)
16	0.503 5	0.495 8	0.551 9	0.591 3	**0.820 0**	0.607 7(6, 3)
17	0.733 4	0.750 7	0.622 0	**0.976 3**	0.797 0	0.967 7(70, 6)
18	0.733 0	0.731 0	0.680 0	**0.798 2**	0.546 3	0.746 3(95, 39)
19	0.499 4	0.501 3	0.542 3	0.561 7	**0.856 1**	0.577 1(31, 10)
20	0.814 6	0.757 9	0.862 6	**0.863 8**	0.815 8	0.849 3(5, 2)
21	0.697 5	0.669 8	0.732 4	0.768 6	**0.821 1**	0.820 1(5, 36)
22	0.732 3	0.761 4	0.760 7	0.773 9	**1.000 0**	0.772 0(8, 16)
23	0.948 9	0.977 2	0.944 7	0.954 9	0.817 0	**0.987 3**(**16**, **37**)
24	0.776 8	0.698 0	0.809 1	0.809 1	**0.903 6**	0.831 3(100, 3)

续表 3

序号	SOM-VAE	N2D	IDEC	DTCR	TSC_CNN	LLE_GMM(k, d)
25	0.786 7	0.750 8	**0.903 0**	0.902 3	0.810 5	0.873 8(24, 9)
26	0.591 7	0.609 1	0.690 0	0.876 9	0.925 9	**0.946 7(31, 19)**
27	0.654 8	0.679 6	0.657 2	0.835 4	**0.927 2**	0.750 5(21, 3)
28	0.874 2	0.893 4	0.889 3	0.922 3	0.533 3	**0.928 4(100, 6)**
29	—	—	0.885 7	0.916 8	—	**0.957 4(20, 8)**
30	—	—	0.501 7	0.565 9	—	**0.651 2(91, 91)**
31	0.498 7	0.505 8	0.499 1	0.828 6	**0.892 0**	0.686 9(132, 2)
32	0.498 2	0.628 3	0.633 8	0.698 4	**0.871 5**	0.771 5(4, 22)
33	0.502 3	0.500 1	0.501 6	0.711 4	0.589 9	**0.977 9(8, 7)**
34	0.535 5	0.511 5	0.559 7	0.733 8	0.824 7	**0.922 6(20, 19)**
35	0.495 4	0.495 8	0.515 7	0.627 1	0.547 6	**0.638 0(70, 14)**
36	0.895 7	0.889 0	0.894 7	0.898 4	0.883 5	**0.903 4(6, 5)**
Avg	0.646 0	0.655 1	0.651 5	0.771 3	0.766 2	0.802 0
Win	0	0	1	6	11	18

3.4　消融实验

LLE_GMM 算法有 LLE 和 GMM 两个模块,为验证两个模块的有效性,分别设置 GMM 和 LLE_Kmeans 两个对照实验,实验结果如表 4 中第 2 和第 3 列所示。仅使用 GMM 模块,平均 RI 指数为 0.715 6,相较于 LLE_GMM 下降了 8.64%;LLE_Kmeans 的平均 RI 指数为 0.773 8,相较于 LLE_GMM 下降了 2.82%。实验证明,GMM 相较于 Kmeans 可以更好地拟合复杂的数据分布,发现椭圆形簇,提升聚类效果。加入 LLE 模块的 GMM 通过维数约简有效降低了数据冗余,更好地表达非线性数据的内蕴特征,提升了聚类效果。

表 4　消融实验结果

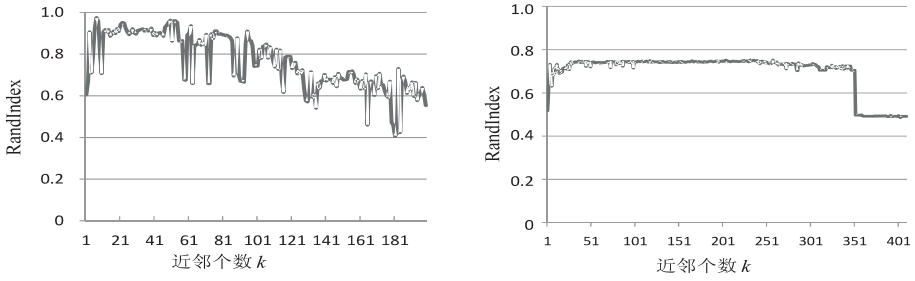
算法	平均 RI 指数
GMM	0.715 6
LLE_Kmeans	0.773 8
LLE_GMM	0.802 0

3.5　参数对算法性能的影响

LLE_GMM 算法有两个参数 k、d,分别表示近邻个数以及嵌入维数。

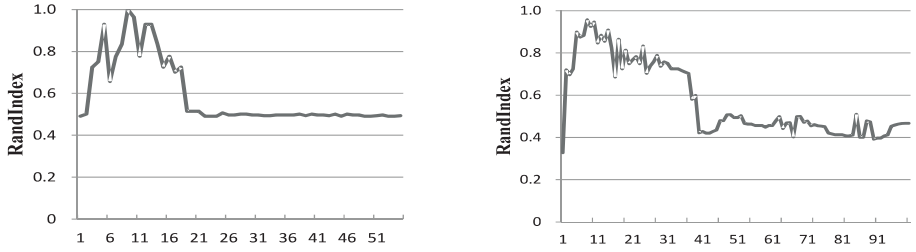
图 1 给出了 $d=35$ 在 DiatomSizeReduction 数据集上,以及 $d=16$ 在 DistalPhalanxOutlineAgeGroup 数据集上,算法的 RI 值随近邻个数 k 的变化情况。从图 1 中可以看出,当 k 的取值过小时,RI 值较小,考虑可能是过小的近邻个数无法保证时间序列样本在低维空间的拓扑结构;随着 k 的增大,RI 值逐渐增大达到最大值,然后在一定范围内波动;但是当 k 值过大时,RI 值呈现下降趋势,考虑近邻个数过大时无法体现数据集的局部特性。因此,LLE_GMM 算法需要根据应用场景选择合适的 k 值。

图 2 给出了 $k=15$ 时在 coffee 和 Meat 数据集上,算法的 RI 值随嵌入维数 d 的变化情况。从图 2 中可以看出,当 d 的取值过小时,RI 值较小,考虑可能是过小的嵌入维数导致不同样本在嵌入空间相互交叠;随着 d 逐步增大,RI 值快速增大达到最大值;随后当 d 值过大时,RI 值呈现下降趋势并最终稳定在一定范围内,考虑信息保留过多影响对原始数据的特征表达,使得效果下降。所以 LLE_GMM 算法并不需要很高的嵌入维数就可以获得很好的聚类效果。



(a)DiatomSizeReduction 数据　　(b)DistalPhalanxOutlineAgeGroup 数据

图 1　LLE_GMM 算法 RI 值随近邻个数 k 的变化



(a)coffee 数据 (b)Meat 数据

图 2 LLE_GMM 算法 RI 值随嵌入维数 d 的变化

4 结束语

提出了一种基于 LLE 和 GMM 的时间序列聚类算法。首先从保留数据集局部结构的角度,使用 LLE 将每个高维时间序列样本表示为其 k 近邻的线性组合,并在低维空间进行重构,在保持数据集局部几何结构的同时实现维数约简;然后使用 GMM 从概率分布的角度进行聚类分析。在 36 个数据集上分别与基于深度学习和基于非深度学习的算法进行对比,结果表明 LLE_GMM 的聚类性能好于已有算法。该文所提算法有两个参数 k 和 d,人工选取参数耗时且可能无法获得全局最优,因此如何自适应地选择最优参数值有待进一步研究;同时 GMM 限制样本个数不得小于维数,如何在小样本高维数据上改进聚类效果仍需进一步探索。

参考文献:

[1] SOHEILA M, MOHAMMAD R K. A comparative study on weighting - based clustering techniques: time series data [C]//Proc of 8th conference of AI & robotics and 10th RoboCup Iranopen international symposium (IRANOPEN). Qazvin: IEEE, 2018: 65-72.

[2] DING Xin, HAO Kuangrong, CAI Xin, et al. A novel similarity measurement and clustering framework for time series based on convolution neural networks [J]. IEEE Access, 2020, 8: 173158-173168.

[3] SHANG Du, SHANG Pengjian, LIU Liu. Multidimensional scaling method for complex time series feature classification based on generalized complexity-invariant distance [J]. Nonlinear Dynamics, 2019, 95(4): 2875-2892.

[4] HOTELLING H. Analysis of a complex of statistical variables into principal compnents [J]. Journal of Educational Psychology, 1993, 24: 417-441.

[5] WANG J Z. Geometric structure of high-dimensional data and dimensionality reduction [M]. Berlin: Springer, 2011.

[6] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000, 290 (5500): 2323-2326.

[7] 武天鸿,翁小清,单中南. 基于 LDA 符号表示的时间序列分类算法[J]. 计算机应用与软件,2020,37(2):259-265.

[8] HE H, TAN Y. Unsupervised classification of multivariate time series using VPCA and fuzzy clustering with spatial weighted matrix distance [J]. IEEE Transactions on Cybernetics, 2020, 50(3): 1096-1105.

[9] AZENCOTT R, MURAVINA V, HEKMATI R, et al. Automatic clustering in large sets of time series [J]. Contributions to Partial Differential Equations and Applications, Computational Methods in Applied Sciences, 2019, 47: 65-75.

[10] 张贝贝,安百国,张宝学. 基于 Copula 函数的非线性时间序列聚类[J]. 数理统计与管理,2019,38(3):450-459.

[11] GUO Hongyue, WANG Lidong, LIU Xiaodong. Dynamic time alignment kernel based fuzzy clustering of non-equal length vector time series [J]. International Journal of Machine Learning and Cybernetics, 2019, 10(11): 3167-3179.

[12] CHANDERENG T, GITTER A. Lag penalized weighted correlation for time series clustering [J]. BMC Bioinformatics, 2020, 21(1): 1-15.

[13] YANG Huahui, MENG Chen, WANG Cheng, et al. SDR: a novel similarity measure using curve fitting method for time series data clustering [C]//Proc of 9th international conference on information science and technology (ICIST). Hulunbuir: IEEE, 2019: 464-469.

[14] EUAN C, OMBAO H, ORTEGA J. The hierarchical spectral merger algorithm: a new time series clustering procedure [J]. Journal of Classification, 2018, 35(1): 71-99.

[15] DUAN Lingzi, YU Fusheng, PEDRYCZ W, et al. Time-series clustering based on linear fuzzy information granules [J]. Applied Soft Computing Journal, 2018, 73(1): 1053-1067.

[16] CAIADO J, CRATO N, PONCELA P. A fragmented-periodogram approach for clustering big data time series [J]. Advances in Data Analysis and Classification, 2020, 14(1): 117-146.

[17] WANG Xiao, YU Fusheng, PEDRYCZ W, et al. Clustering of interval-valued time series of unequal length based on improved dynamic time warping [J]. Expert Systems with Applications: 2019, 125: 293-304.

[18] WANG Yan, RU Yunian, CHAI Jianping. Time series clustering based on sparse subspace clustering algorithm and its application to daily box-office data analysis [J]. Neural Com-

puting and Applications,2019,31(9):4809-4818.

[19] YAZDI S V,DOUZAL-CHOUAKRIA A,GALLINARI P, et al. Time warp invariant dictionary learning for time series clustering: application to music data stream analysis[J]. Machine Learning and Knowledge Discovery in Databases, 2019,11051:356-372.

[20] YAZDI S V,DOUZAL-CHOUAKRIA A. Time warp invariant kSVD:sparse coding and dictionary learning for time series under time warp[J]. Pattern Recognition Letters,2018, 112:1-8.

[21] ZHANG Qin, WU Jia, ZHANG Peng, et al. Salient subsequence learning for time series clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2019,41 (9):2193-2207.

[22] XIAO Zhiwen,XU Xin. RTFN:a robust temporal feature network for time series classification[J]. Information Sciences, 2020,571(4):65-86.

[23] CORDUAS M, RAGOZINI G. Comparing multistep ahead forecasting functions for time series clustering[C]//Proc of classification, (Big) data analysis and statistical learning, studies in classification,data analysis,and knowledge organization. [s. l.]:Springer,2018:191-199.

[24] XIE Junyuan, GIRSHICK R, FARHADI A. Unsupervised deep embedding for clustering analysis[C]//Proc of the 33rd international conference on machine learning. New York:JMLR,2016:478-487.

[25] GUO Xifeng, GAO Long, LIU Xinwang, et al. Improved deep embedded clustering with local structure preservation [C]//Pros of the 26th international joint conference on artificial intelligence (IJCAI-17). Melbourne: AAAI Press, 2017:1753-1759.

[26] MADIRAJU N S,SADAT S M,FISHER D,et al. Deep temporal clustering: fully unsupervised learning of timedomain features[J]. arXiv:1802.01059,2018:1-11.

[27] MA Qianli,ZHENG Jiawei,SEN Li,et al. Cottrell. Learning representations for time series clustering[C]//Proc of 33rd conference on neural information processing systems. Vancouver:NeurIPS,2019:3781-3791

[28] FORTUIN V,HÜSER M,LOCATELLO F,et al. SOMVAE: interpretable discrete representation learning on time series [C]//Proc of the seventh international conference on learning representations (ICLR). [s. l.]:[s. n.],2019:1-18.

[29] MCCONVILLE R,SANTOS-RODRIGUEZ R,PIECHOCKI R J,et al. N2d:(Not too) deep clustering via clustering the local manifold of an autoencoded embedding[C]//Proc of 25th international conference on pattern recognition (ICPR). Milan:IEEE,2020.

[30] CHEN Y,KEOGH E,HU B,et al. The UCR time series classification archive [EB/OL]. 2015. www. cs. ucr. edu/~eamonn/time_series_data/

[31] RAND W M. Objective criteria for the evaluation of clustering methods[J]. Publications of the American Statistical Association,1971,66(336):846-850.

[32] YANG J,LESKOVEC J. Patterns of temporal variation in online media[C]//Proc of the forth international conference on web search and web data mining,WSDM 2011. Hong Kong: ACM,2011:9-12.

[33] LI Zechao,YANG Yi,LIU Jing,et al. Unsupervised feature selection using nonnegative spectral analysis[C]//Proc of twenty-sixth AAAI conference on artificial intelligence. Canada:AAAI,2012:1026-1032.

[34] SHI Lei,DU Liang,SHEN Yidong. Robust spectral learning for unsupervised feature selection[C]//Proc of IEEE international conference on data mining. Shenzhen:IEEE,2014: 977-982.

[35] PAPARRIZOS J,GRAVANO L. K-shape:efficient and accurate clustering of time series[J]. SIGMOD Record,2016, 45(1):69-76.