TICC算法介绍

对高维时间序列聚类方法-2017KDDbest paper

下一页 →

简介

Toeplitz Inverse Covariance-Based Clustering（TICC）。是KDD2017的最佳论文，该方法主要应用于现实生活中有大量的随时间变化的高维数据。

- 汽车驾驶：油门，刹车
- 支付软件: 用户登录，购买，转账等行为数据

在没有确定的标签的情况下，通过发掘这些时间序列数据中隐藏的信息，即将**输入的时间序列划分为若干可能的状态**,并且标记出每条序列的各段。

- 驾驶汽车：起步，上坡，超车，拥堵
- 支付软件：用户发放薪水，过节，购买股票

为了从时间序列数据中获得到有用的信息，我们需要**同时**对数据进行两种操作：

- 将各条数据进行**分割**
- 将分割后的各个结果**聚类**

传统的时间序列聚类方法是考察**各个维度之间的绝对值**，以此来确定不同序列之间的相似程度。TICC方法是通过**各个维度之间的相关性**来确定输入序列之间的相似度。

TICC方法的优势是可以同时的进行时间序列的分割和聚类。

原理简介

考虑输入的时间序列形状如下：

$$x_{\text{orig}} = \begin{bmatrix} | & | & | & & | \\ x_1 & x_2 & x_3 & \dots & x_T \\ | & | & | & & | \end{bmatrix}$$

其中$x_i \in \mathcal{R}^n$是第i个多元观察序列。TICC算法的目标是将T个观测序列聚为K个类。其中每一时刻的信号x_i为n维向量。

TICC的时间序列聚类的最小数据粒度不是单一的时间戳x_i,而是在时间序列中将每个时间戳放置在其前述背景下进行考虑（譬如在汽车驾驶的研究背景下，一次观察可能会显示汽车的当前驾驶状态，但一个短暂的窗口，即使只持续几分之一秒，也可以让我们更为全面的了解汽车的行驶状态）。因此TICC定义一个时间窗口，大小为$w << T$并将x_i之前相邻的w个时间戳拼接成一个向量，$\boldsymbol{X}_i = \left[x_{i-w+1}, \ldots, x_i\right]^T$，这样整个聚类输入的数据矩阵为：

$$X = \begin{bmatrix} | & | & & | & | \\ x_1 & x_2 & & x_{T-w} & x_{T-w+1} \\ | & | & & | & | \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ | & | & & | & | \\ x_w & x_{w+1} & & x_{T-1} & x_T \\ | & | & & | & | \end{bmatrix} \in \mathbb{R}^{nw \times (T-w+1)}$$

核心思想

TICC的核心思想是：**通过TICC聚类得到的每一个簇都代表原始多元时间序列中一种特定的“状态”，当原始多元时间序列处于这种特定的状态时会保留一定的时不变结构，该结构会在整个簇所涵盖的时间窗范围内持续存在，而与时间窗的起点无关。** 为了将上述核心假设，尤其是假设中的时不变结构显式化，文章引入了Toeplitz矩阵结构，用于构造能够描述各个时间序列之间相关性的逆协方差θ 。Toeplitz矩阵结构是这样的：

$$T = \begin{pmatrix} t_0 & t_1 & t_2 & \cdots & t_n \\ t_1 & t_0 & t_1 & \cdots & t_{n-1} \\ t_2 & t_1 & t_0 & \cdots & t_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_n & t_{n-1} & t_{n-2} & \cdots & t_0 \end{pmatrix}$$

在论文中描述各个序列之间相关性的逆协方差Θ_i就是Toeplitz矩阵，其公式如下：

$$\Theta_i = \begin{bmatrix} A^{(0)} & \left(A^{(1)}\right)^T & \left(A^{(2)}\right)^T & \cdots & \cdots & \left(A^{(w-1)}\right)^T \\ A^{(1)} & A^{(0)} & \left(A^{(1)}\right)^T & \ddots & \ddots & \vdots \\ A^{(2)} & A^{(1)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \left(A^{(1)}\right)^T & \left(A^{(2)}\right)^T \\ \vdots & \ddots & \ddots & A^{(1)} & A^{(0)} & \left(A^{(1)}\right)^T \\ A^{(w-1)} & \cdots & \cdots & A^{(2)} & A^{(1)} & A^{(0)} \end{bmatrix} \in \mathbb{R}^{nw \times nw}$$

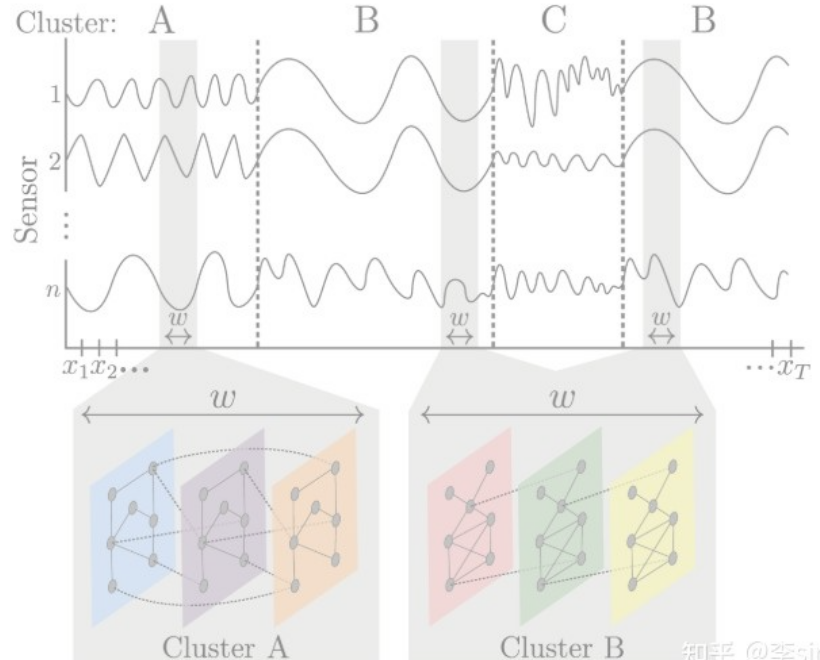
在上式中，$A^{(0)}, A^{(1)}, \ldots, A^{(w-1)} \in \mathbb{R}^{n \times n}$，$A_{ij}^{(0)}$，是$A_{ij}^{(0)}$矩阵中$i$行$j$列的元素，代表当前时刻序列$i$和序列$j$的依赖关系，依次类推,$A_{ij}^{(1)}$代表当前时刻序列$i$与后一时刻序列$j$之间的依赖关系。$A_{ij}^{(2)}$代表当前时刻序列$i$与后两个时刻序列$j$之间的依赖关系。总而言之，TICC方法将各个类的特征θ_i构造为Toeplitz矩阵，通过构造这样的矩阵可以保证只要是属于这一类的则固定顺序时间戳所包含的所有序列的依赖关系都保持不变。

算法核心

右图中A类中的三层依赖网络结构为例（由于图中绘制成三层网络结构，因此我们可认为$w=3$），首先：蓝色，紫色以及橙色三层，每层之间的图都相同，这对应了上述θ_i矩阵中主对角线元素都相同，为 $A^{(0)}$ 的事实；同时，蓝色与紫色层间边的位置和紫色与橙色层间边的位置相同，这对应了上述θ_i矩阵中一行二列以及二行三列元素相同，为 $\left(A^{(1)}\right)^T$ 的事实。

换句话说，任选cluster A中的任意一时间戳，其对应的所有序列与前置时间戳所对应的所有序列，包括与后置时间戳所对应的所有序列之间的依赖关系都是保持不变的，和cluster A何时开始以及合适结束没有任何关系，此种性质文中称之为时不变性。

TICC算法图



求解问题

本算法求解的最终问题是K个高斯逆协方差$\Theta = \{\Theta_1, \Theta_2, \ldots, \Theta_K\}$和最后被分配的集合$P = \{P_1, \ldots, P_K\}$。

其中$P_i \subset \{1, 2, \ldots, T\}$。整体目标函数定义如下：

$$\underset{\Theta \in \mathcal{T}, \mathrm{P}}{\operatorname{argmin}} \sum_{i=1}^{K} [\overbrace{\|\lambda \circ \Theta_i\|_1}^{\text{sparsity}} + \sum_{X_t \in P_i} (\overbrace{-\ell\ell\left(X_t, \Theta_i\right)}^{\text{log likelihood}} + \overbrace{\beta 1\left\{X_{t-1} \notin P_i\right\}}^{\text{temporal consistency}})]$$

在上式中,尽可能保证每个θ是稀疏的(即每个θ拥有尽可能多的0，仅有少部分元素不是0，这样可以增加可解释性),为了达到这个目的，加入了第一项惩罚项$\overbrace{\|\lambda \circ \Theta_i\|_1}^{\text{sparsity}}$，这里和$L_1$范数惩罚LASSO方法中的惩罚项一致。并且为了使相邻时间向量X_{i-1}以及X_i尽可能的聚为一类，加入了函数的最后一项$\overbrace{\beta 1\left\{X_{t-1} \notin P_i\right\}}^{\text{temporalconsistency}}$取最小值（其中$\beta$为参数）。

超参数设置

最后使用极大似然估计方法，保证函数中间一项$\sum_{X_t \in P_i} (\overbrace{-\ell\ell\left(X_t, \Theta_i\right)}^{\log\ \text{likelihood}})$取最小值，其中$\ell\ell\left(X_t, \Theta_i\right)$定义如下：

$$\ell\ell\left(X_t, \Theta_i\right) = -\frac{1}{2}\left(X_t - \mu_i\right)^T \Theta_i \left(X_t - \mu_i\right) + \frac{1}{2}\log\det\Theta_i - \frac{n}{2}\log(2\pi)$$

综上所述，TICC算法总共有四个超参数，正则化参数λ,β，时间窗大小w以及聚类数K。

数据实验

论文作者使用模拟数据和真实数据来验证模型的效果。将TICC模型与高斯混合模型（GMM），EEV模型，和基于距离的模型，DTW,Neural Gas,K-means。 通过设定假定的正确的类数K,将聚类问题的评估转化为多分类问题，使用`marco F1`来评估各个方法。

$$\mathrm{F1} = \frac{2\mathrm{Recall} \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}}$$

实验图表

		Temporal Sequence			
	Clustering Method	1,2,1	1,2,3,2,1	1,2,3,4,1,2,3,4	1,2,2,1,3,3,3,1
	TICC	**0.92**	**0.90**	**0.98**	**0.98**
Model-Based	TICC, $\beta = 0$	0.88	0.89	0.86	0.89
	GMM	0.68	0.55	0.83	0.62
	EEV	0.59	0.66	0.37	0.88
Distance-Based	DTW, GAK	0.64	0.33	0.26	0.27
	DTW, Euclidean	0.50	0.24	0.17	0.25
	Neural Gas	0.52	0.35	0.27	0.34
	K-means	0.59	0.34	0.24	0.34

Table 1: Macro-F_1 score of clustering accuracy for four different temporal sequences, comparing TICC with several alternative model and distance-based methods.

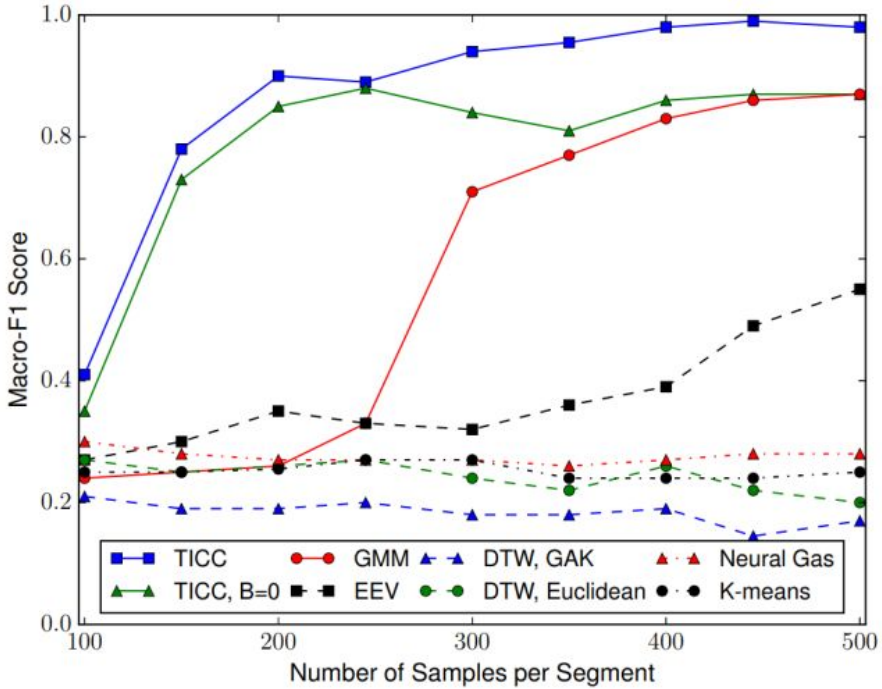


Figure 3: Plot of clustering accuracy macro-F_1 score vs. number of samples for TICC and several baselines. TICC needs significantly fewer samples than the other model-based methods to achieve similar performance, while the distance-based measures are unable to capture the true structure.

使用真实数据

TICC论文中使用汽车驾驶的真实数据，其数据来源有：

- 制动踏板位置
- X轴（前向）加速器
- Y轴（纵向）加速器
- 转向轮角度
- 车辆速度
- 引擎转速
- 油门踏板位置

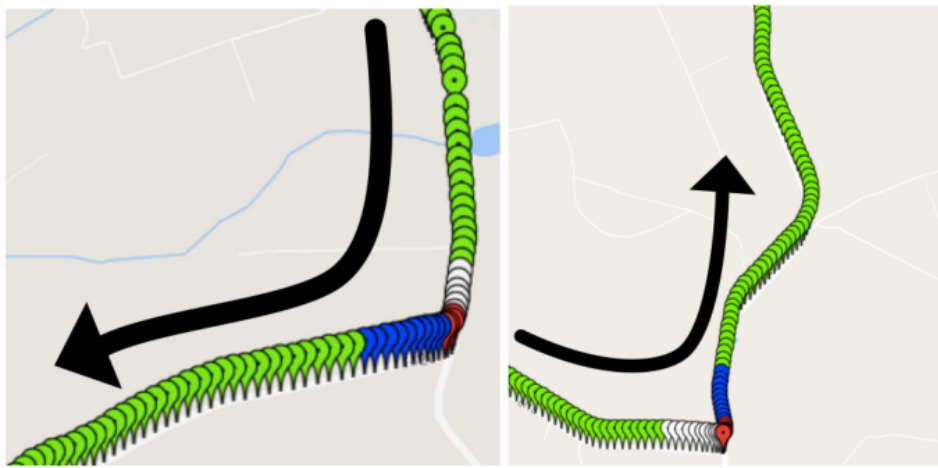
输入的时间序列有7个维度，根据BIC准则，最终选取聚类的类别为$K = 5$，也就是五种驾驶状态分别是：减速，转向，加速，直行，环路。

聚类结果

	Interpretation	刹车	前向加速器	纵向加速器	转向轮角度	车辆速度	引擎转速	油门位置
#1	减速	25.64	0	0	0	27.16	0	0
#2	转向	0	4.24	66.01	17.56	0	5.13	135.1
#3	加速	0	0	0	0	16.00	0	4.50
#4	直行	0	0	0	0	32.2	0	26.8
#5	环路	4.52	0	4.81	0	0	0	94.8

表中的得分可以看做是每个传感器对这一个类（形式状态）的相对重要性。

可视化结果



(a) (b)

Figure 5: Two real-world turns in the driving session. The pin color represents cluster assignment from our TICC algorithm (Green = Going Straight, White = Slowing Down, Red = Turning, Blue = Speeding up). Since we cluster based on structure, rather than distance, both a left and a right turn look very similar under the TICC clustering scheme.

总结

1. TICC方法提供了一种对高维时间序列数据进行聚类的方法，特点是同时进行分割和聚类，并且是基于模型的算法，不是基于距离的算法。
2. 与其他常用的方法相比TICC方法得到了更好的表现

Learn More

PDF · GitHub