# 基于约束传递的深度主动时序聚类方法

霍纬纲 朱 旭* 张 盼
(中国民航大学计算机科学与技术学院 天津 300300)

**摘 要**：已有的深度主动聚类方法未能通过标注样本推理生成必须链接(ML)约束或不能链接(CL)约束，标注成本较高。为此该文提出一种基于约束传递的深度主动时序聚类方法。该方法设置了标注类簇集合(ACS)及相应的辅助标注集合(AAS)。通过预训练时序自编码器得到时序样本的表示向量。在深度聚类的每个训练轮次过程中，采样并标注表示空间中离类簇中心最近的样本存入ACS，使每个ACS内的样本属同一类别而ACS集合间的样本属于不同类别，然后从包含样本数最小的ACS集合中随机选取时序样本，采样并标注与该样本不属于同一类簇且距其所在类簇中心最近的时序样本存入AAS，使ACS与相应的AAS中的样本为不同类别，由ACS及对应的AAS中的样本推理生成ML和CL约束。由基于t-分布的类簇分布与其生成的辅助分布间的KL散度以及使满足ML及CL约束的时序样本在表示空间距离分别变小和变大的约束损失更新时序自编码器中编码网络参数和聚类中心。在18个公开数据集上的实验结果表明，该方法聚类效果在较低标注预算下平均RI值比已有的典型基线模型均提升5%以上。

**关键词**：深度时序聚类；主动学习；约束传递
**中图分类号**：TN911.7; TP18 **文献标识码**：A **文章编号**：1009-5896(2025)04-1172-10
**DOI**: 10.11999/JEIT240855

## 1 引 言

物联网及传感器技术的快速发展使得海量时序数据被累积存储，然而这些数据缺乏精准标注，无法被充分利用。人工标注时序数据成本较高，这限制了有监督学习方法在时序数据集上的应用。因此研究人员考虑使用聚类方法对时序数据进行分析，时序聚类技术能揭示时序数据集的潜在规律，在气象学、生物科学和金融等领域中得到了广泛应用[1]。时序数据具有高维、时序关联等特点，使得传统聚类算法很难直接用于时序聚类任务。深度学习因其强大的数据表征能力已被应用于聚类任务[2–5]。但是深度聚类方法[6–10]缺乏标签指导，聚类结果易受噪声数据和异常值的影响[11]。为此研究者们提出半监督深度聚类方法[12]，旨在通过少量监督信息进一步提高聚类的准确性。在聚类任务中，监督信息通常表现为成对的必须链接(Must-Link, ML)约束和不能链接(Cannot-Link, CL)约束。ML约束表示两个样本属于同一类别，CL约束则表示两个样本属于不同类别。由于半监督聚类方法的性能依赖于预先给定的ML与CL约束，未考虑如何主动选择高质量ML和CL约束，因此对聚类效果提升有限[11]。主动学习通过选取少量具有代表性的标注数据，最大程度提高模型性能[13]。主动聚类方法通过人工标注获取ML与CL约束，并利用这些约束指导聚类过程[11,14]。考虑到时序数据标注的高成本，如何在时序聚类过程中选择ML与CL约束进行人工标注，并以最小标注代价指导聚类任务，成为当前深度主动时序聚类研究的关键环节。

## 2 相关工作

时序聚类的主要任务是以相似性度量为基础，将具有相同特征的时序样本划分同一类簇，不同特征的时序样本归为不同类簇[15]。半监督聚类方法通过给定的ML和CL约束监督提高聚类效果，主动聚类通过标注高价值ML与CL约束改进半监督聚类方法性能。下面将从传统时序聚类、基于深度学习的时序聚类、半监督及主动聚类3方面阐述与本文相关的研究工作。

### 2.1 传统时序聚类方法

传统的时序聚类方法可分为基于原始数据[16,17]和基于时序特征[18,19]的聚类方法。基于原始数据的方法通常采用更适用于时序分析的距离度量函数，以扩展和改善传统聚类算法的适用性。文献[16]使用动态时间弯曲方法作为时间序列的距离度量，基于K-means算法对时序数据进行聚类，实现了具有异步相似形态的不等长时序聚类。文献[17]提出了K-shape算法，该算法由时间序列的互相关性计算

收稿日期：2024-10-14；改回日期：2025-03-10；网络出版：2025-03-20
*通信作者： 朱旭 2022051001@cauc.edu.cn
基金项目：国家自然基金民航联合基金重点项目(U2033205)，国家自然科学基金(62173331)，天津市自然科学基金面上项目(24JCY-BJC00990)
Foundation Items: The Civil Aviation Joint Fund of The National Natural Science Foundation of China (U2033205), The National Natural Science Foundation of China (62173331), Tianjin Natural Science Foundation General Project (24JCYBJC00990)

时序样本间的距离，并考虑了时序数据的平移不变性和尺度不变性。这些基于原始数据的时序聚类方法易受噪声影响，计算成本高，不适合处理大规模含噪声的时序数据集。基于时序特征的聚类方法对原始时序数据进行特征提取，提高时间序列聚类算法的效率和质量。文献[18]提出了u-shapelets方法，通过提取时间序列具有辨识度的局部特征进行聚类，有较好的聚类可解释性。文献[18]改进了u-shapelets算法，通过选取最优子序列的方法得到了较好聚类效果，但基于u-shapelets的方法将特征提取作为预处理步骤，提取出的特征不一定适合聚类任务[5]。

### 2.2 基于深度学习的时序聚类方法

基于深度学习的时序聚类方法首先将原始时序数据转为低维特征嵌入，然后用传统的聚类方法如K-means聚类[20]、层次聚类[21]、谱聚类[22]等对特征嵌入进行聚类，与传统时序聚类算法相比，此类方法更能有效捕获时序样本特征。文献[6]提出了一种基于自编码器的深度聚类方法深度嵌入聚类(Deep Eebedding Clustering, DEC)。该方法由预训练的自编码器获得训练样本的表示向量，采用K-means计算表示空间中的初始聚类中心，根据t-分布计算每个表示向量隶属所有类簇的概率，通过优化表示向量隶属每个类簇的分布及由该分布生成的辅助分布间的KL散度更新自编器中编码网络参数。文献[7]提出了一种深度时序聚类方法深度时序聚类(Deep Temporal Clustering, DTC)，使用基于卷积神经网络(Convolutional Neural Network, CNN)与双向长短期记忆网络(Bidirectional Long Short-Term Memory, Bi-LSTM)的时序自编码器(Temporal Auto Encoder, TAE)提取时序数据全局与局部时序特征的表示向量，采用层次聚类生成表示空间中的初始类簇中心，由重构损失与文献[6]中的聚类损失更新TAE网络参数。文献[8]提出的深度时序聚类表示(Deep Temporal Clustering Representation, DTCR)由基于多层膨胀循环神经网络[23](Dilated Recurrent Neural Network, Dilated RNN)自编码器提取时序特征，将自编码器表示特征空间中的K-means迭代求解最佳簇分布过程转化为最大化表示向量Gram矩阵迹的过程，并以此作为聚类损失。此外该方法重新排列时序样本的部分时间步生成伪造样本，引入辅助分类任务区分真实样本与伪造样本，以提升编码器表示能力，使用重构损失、辅助分类损失及聚类损失优化网络参数。虽然这些深度时序聚类方法在聚类性能上优于传统时序聚类方法，但此类方法未使用人工标注的ML与CL约束对聚类指导聚类过程，因此聚类准确性仍有提升空间。

### 2.3 半监督及主动聚类方法

半监督及主动聚类均借助ML和CL约束信息提高聚类效果。文献[12]由预先给定的ML与CL约束改进DEC[6]模型的样本嵌入学习过程，使得ML约束样本在表示空间中距离靠近，CL约束样本在表示空间中的距离拉远。但这类方法没有考虑ML与CL约束质量对于聚类性能的影响。文献[24]提出了包含探索和巩固两个阶段的主动聚类方法，探索阶段采样标注方法为：随机选则一个样本构建初始邻域，然后每次选择距上次所选样本距离最远的样本，标注该样本与已有邻域中样本关系，若为ML约束，则将样本放入该邻域，若该样本与已有邻域中的样本均为CL约束关系，则将该样本放入新建邻域，直至创建$k$个邻域；巩固阶段的目的是为每个邻域添加更多样本，其方法为：随机选择一个样本，若探索阶段生成的某个邻域中的样本与该随机选取的样本为ML约束，则将其存入该邻域，由此直至达到标注预算。使用$k$个邻域创建$k$个类簇，将每个邻域中所有样本的均值作为类簇中心，将训练样本分配到距其最近的簇中心以形成最终类簇。文献[11]中提出了面向图片的主动聚类方法主动深度聚类(Active Deep Clustering, ADC)，该方法采用K-means算法得到图片表示空间的初始聚类中心，在每个轮次训练中在每个类簇内选取距簇中心最近的一个样本及距中心最远的若干个样本组成样本对，然后标注这些样本对的ML或CL约束关系，采用文献[6]中的聚类损失与文献[12]中的ML及CL约束损失更新编码网络参数和类簇中心。

已有相关工作存在如下不足：

(1)深度时序聚类模型中尚未引入主动学习机制提高聚类效果。深度主动聚类方法中的主动学习策略只在表示空间聚类类簇内采样，没有考虑类簇间的差异生成ML和CL约束。

(2)未通过已标注的ML与CL约束推理生成更多的监督信息，人工标注的ML与CL约束未得到充分利用，从而使得标注成本较高。

为此本文提出了一种基于约束传递的深度主动时序聚类方法，主要创新点如下：

(1)设计了包含探索和巩固的两阶段主动聚类采样策略，其中探索阶段采样选取表示空间中离类簇中心最近的样本，巩固阶段选取与类簇中心最近但不属同一类簇的样本，实现了基于表示空间中类簇差异性的多样性主动采样。

(2)设置了与聚类类簇个数相同的标注类簇集

合(Annocation Cluster Set, ACS)与辅助标注集合(Auxiliary Annocation Set, AAS)，存储主动聚类采样标注结果。其中ACS内的样本属同一类别而ACS间的样本属于不同类别，ACS与相应的AAS中的样本为不同类别。由ACS与AAS中的样本推理生成新的ML约束与CL约束。

## 3 DATC-CT模型

### 3.1 问题定义

单变量时序数据集记为$X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$，其中$N$为时序样本数量。文中的目标为学习非线性映射$f_\theta:X\to Z$，$f_\theta$可同时得到$X$的特征表示$Z=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N\}$以及表示空间中的$G$个类簇$\pi_g\,(1<g<G)$，每个类簇的中心记为$\boldsymbol{\mu}_g(1<g<G)$。在训练$f_\theta$过程中，依据表示空间$Z$的聚类形态标注生成高质量的ML约束集合$M=\{\ (\boldsymbol{x}_i,\boldsymbol{x}_j)\,|\boldsymbol{x}_i,\boldsymbol{x}_j\in X\}$与CL约束集合$C=\{\,(\boldsymbol{x}_m,\boldsymbol{x}_n)\,|\boldsymbol{x}_m,\boldsymbol{x}_n\in X\}$，$f_\theta$使得集合$M$中$\boldsymbol{x}_i$的表示特征$\boldsymbol{z}_i$靠近$\boldsymbol{x}_j$的表示特征$\boldsymbol{z}_j$，集合$C$中$\boldsymbol{x}_m$的表示特征$\boldsymbol{z}_m$远离$\boldsymbol{x}_n$的表示特征$\boldsymbol{z}_n$。

### 3.2 模型结构

DATC-CT整体结构由图1所示。DATC-CT模型由时序自编码器(Temporal Auto Encoder, TAE)预训练模块、ML与CL约束生成模块和聚类模块组成。TAE预训练模块由基于1维卷积与双向长短时记忆网络的编码器E和解码器D，学习时序样本$\boldsymbol{x}_i$的特征表示$\boldsymbol{z}_i$。ML与CL约束生成模块设置$G$个标注类簇集合ACS与辅助标注集合AAS，分别记为$\{S_g\}_{g=1}^{G}$与$\{\bar{S}_g\}_{g=1}^{G}$，其中$G$为聚类数目，每个训练轮次中采样标注过程分为探索与巩固两个阶段，探索阶段采样并标注在表示空间中离类簇中心最近的样本存入集合$S_g(1<g<G)$，使集合$S_g$内的样本属同一类别而不同$S_g$的样本属于不同类别；巩固阶段从包含样本最少的ACS集合$S_k(1<k<G)$中随机选择一个样本，采样并标注与该样本不属于同一类簇且距其在表示空间所在类簇中心最近的时序样本存入$S_k$或$\bar{S}_k$，使$S_k$与$\bar{S}_k$中的样本为不同类别。由ACS集合$S_g\,(1<g<G)$与AAS集合$\bar{S}_g$中的样本推理生成ML与CL约束。聚类模块计算样本$x_i$的表示向量$\boldsymbol{z}_i$隶属于类簇中心$\boldsymbol{\mu}_g(1<g<G)$的概率，将每个样本分配给隶属概率最大的类簇中心形成类簇$\pi_g$。聚类损失$L_1$为表示向量隶属每个类簇的分布与提高样本所属类簇中心的置信度的辅助分布间的KL散度，约束损失$L_2$使满足ML与CL约束的时序样本在表示空间距离分别变小和变大。

### 3.3 TAE预训练模块

图1中的时序编码器E结构如图2所示，由3层1维卷积网络(One-Dimensional Convolution, Conv1D)与批量归一化层(Batch Normalization, BN)提取时序样本$\boldsymbol{x}_i$的局部特征，卷积核大小分别设置为4, 2和2，卷积核个数分别设置为32, 64, 128。由双向长短时记忆网络(Bidirectional Long Short-Term Memory, Bi-LSTM)提取样本$\boldsymbol{x}_i$的全局时序依赖特征。Bi-LSTM隐藏层的维度设为32。Conv1D与Bi-LSTM均采用Leaky ReLU激活函数。TAE的解码器结构与编码器对称，表示向量$\boldsymbol{z}_i$依次通过线性映射层、Bi-LSTM层和3层反卷积

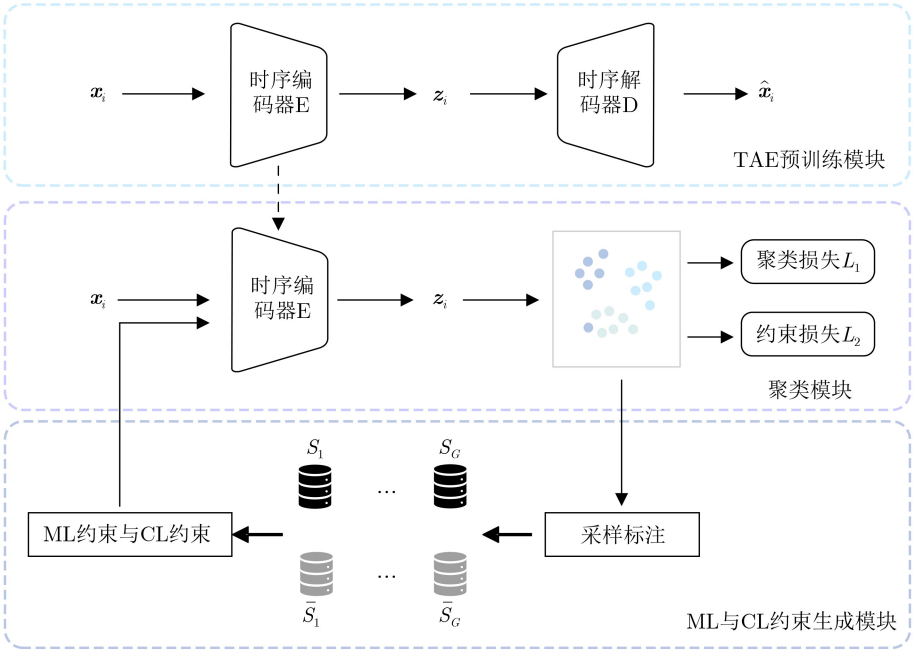



图 1 DATC-CT模型图

网络得到时序样本$\boldsymbol{x}_i$的重构$\hat{\boldsymbol{x}}_i$。TAE采用式(1)所示的重构误差为损失函数

$$L=\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i-\hat{\boldsymbol{x}}_i\|^2 \tag{1}$$

### 3.4 ML与CL约束生成模块

该模块根据每个训练轮次表示特征空间$Z$的聚类形态采样重要数据点并进行标注生成ML与CL约束。主动采样过程包括探索和巩固两个阶段。在探索阶段，从表示空间的每个类簇$\pi_g\,(1<g<G)$中选择距类簇中心$\boldsymbol{\mu}_g$最近的表示向量$\boldsymbol{z}_i$且$\boldsymbol{z}_i$对应的时序样本$\boldsymbol{x}_i$尚未被选择。然后遍历ACS集合$S_g\,(1<g<G)$，若$S_g$为空，将样本$\boldsymbol{x}_i$放入$S_g$，否则从$S_g$中随机选择样本$\boldsymbol{x}_j$，由领域专家标注$\boldsymbol{x}_i$与$\boldsymbol{x}_j$是否为同一类别。若$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为同一类别，将$\boldsymbol{x}_i$放入$S_g$，若$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为不同类别，则判断是否能将$\boldsymbol{x}_i$放入$S_{g+1}$，判断方法与能否放入$S_g$相同。由上述采样过程可知，ACS中存储了每个轮次训练过程中表示空间聚类类簇的代表样本，探索阶段具体采样过程见算法1。

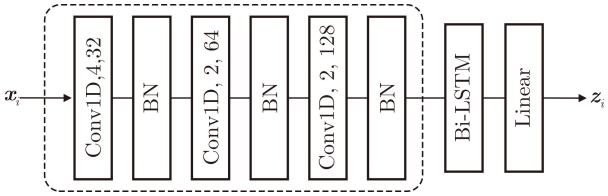



图 2 时序编码器$E$结构

**算法 1　探索阶段采样标注过程**

输入：已标注次数$b$，类簇$\{\pi_g\}_{g=1}^{G}$，类簇中心$\{\boldsymbol{\mu}_g\}_{g=1}^{G}$，ACS集合$\{S_g\}_{g=1}^{G}$

输出：更新后的ACS集合$\{S_g\}_{g=1}^{G}$

(1) **for** $i=1$ **to** $G$ **do**:
(2)　从$\pi_i$中选择尚未被选择且表示向量与$\boldsymbol{\mu}_i$距离最小的样本$\boldsymbol{x}_p$
(3)　**for** $j=1$ **to** $G$ **do**:
(4)　　**if** $S_j=\varnothing$ **then**:
(5)　　将样本$\boldsymbol{x}_p$放入集合$S_j$
(6)　　**break**
(7)　　**else**:
(8)　　从$S_j$中随机选择一个样本$\boldsymbol{x}_q$，标注样本$\boldsymbol{x}_p$与$\boldsymbol{x}_q$的关系，$b=b+1$
(9)　　**if** 样本$\boldsymbol{x}_p$与$\boldsymbol{x}_q$为ML约束 **then**:
(10)　　将样本$\boldsymbol{x}_p$放入集合$S_j$
(11)　　**break**
(12)　　**end if**
(13)　**end if**
(14)　**end for**
(15) **end for**

当每个ACS集合$S_g(1<g<G)$均至少包含一个时序样本时，深度主动聚类每个轮次的主动采样进入巩固阶段，不再进行探索阶段的采样过程。巩固阶段主要是为每个AAS集合$\bar{S}_g(1<g<G)$赋值，以推理生成CL约束。其采样过程为：从包含样本数量最少的ACS集合$S_g$中随机选择时序样本$\boldsymbol{x}_i$，$\boldsymbol{x}_i$的表示向量$\boldsymbol{z}_i$所属类簇及中心分别记为$\pi_g$和$\boldsymbol{\mu}_g$，从表示空间的其它类簇中采样表示向量与$\boldsymbol{\mu}_g$距离最小的样本$\boldsymbol{x}_j$，时序样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$是否同一类别，若$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为ML约束，将样本$\boldsymbol{x}_j$放入$S_g$，若$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为CL约束，将$\boldsymbol{x}_j$放入集合$\bar{S}_g$。巩固阶段过程如算法2所示。

由上述采样过程可知，ACS集合$\{S_g\}_{g=1}^{G}$与AAS集合$\{\bar{S}_g\}_{g=1}^{G}$满足以下性质：

(1)若$\boldsymbol{x}_i,\boldsymbol{x}_j\in S_g$，则样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为ML约束，其中$1<g<G$。

(2)若$\boldsymbol{x}_i\in S_h,\boldsymbol{x}_j\in S_l$，则样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为CL约束关系，其中$1<h,l<G,h\neq l$。

(3)若$\boldsymbol{x}_i\in S_g,\boldsymbol{x}_j\in \bar{S}_g$，则样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为CL约束关系，其中$1<g<G$。

ML约束及ML与CL约束具有以下传递性：若样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$、样本$\boldsymbol{x}_j$与$\boldsymbol{x}_v$均为ML约束，则$\boldsymbol{x}_i$与$\boldsymbol{x}_v$必为ML约束；若样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$为ML约束，样本$\boldsymbol{x}_j$与$\boldsymbol{x}_v$为CL约束，则$\boldsymbol{x}_i$与$\boldsymbol{x}_v$必为CL约束。由此可使用ACS集合$S_g(1<g<G)$与AAS集合$\bar{S}_g(1<g<G)$中的样本推理生成新的约束。集合$S_g$中的两个样本$\boldsymbol{x}_i,\boldsymbol{x}_j$可作为ML约束添加至约束集合$M$，其中$\boldsymbol{x}_i,\boldsymbol{x}_j\in S_g$。集合$S_g$中的样本$\boldsymbol{x}_i$与集合$S_k(1<k<G)$或集合$\bar{S}_g$中的样本$\boldsymbol{x}_j$可作为CL约束添加至约束集合$C$，其中$\boldsymbol{x}_i\in S_g,\boldsymbol{x}_j\in S_k$或$\bar{S}_g$。ML约束集合$M$与CL约束集合$C$的生成过程如算法3所示。

### 3.5 聚类模块

聚类模块由聚类损失$L_1$和约束损失$L_2$实现时

**算法 2　巩固阶段采样标注过程**

输入：已标注次数$b$，类簇$\{\pi_g\}_{g=1}^{G}$，类簇中心$\{\boldsymbol{\mu}_g\}_{g=1}^{G}$，ACS集合$\{S_g\}_{g=1}^{G}$，AAS集合$\{S_g\}_{g=1}^{G}$

输出：更新后的ACS集合$\{S_g\}_{g=1}^{G}$与AAS集合$\{\bar{S}_g\}_{g=1}^{G}$

(1) 从样本数量最少的集合$S_g$中随机选择样本$\boldsymbol{x}_p$，从样本$\boldsymbol{x}_p$所属类簇$\pi_g$之外选择表示向量与$\boldsymbol{\mu}_g$距离最小的样本$\boldsymbol{x}_q$
(2) 标注样本$\boldsymbol{x}_p$与$\boldsymbol{x}_q$的关系，$b=b+1$
(3) **if** 样本$\boldsymbol{x}_p$与$\boldsymbol{x}_q$为ML约束关系 **then**:
(4)　将样本$\boldsymbol{x}_q$放入集合$S_g$
(5) **end if**
(6) **if** 样本$\boldsymbol{x}_p$与$\boldsymbol{x}_q$为CL约束关系 **then**:
(7)　将样本$\boldsymbol{x}_q$放入集合$\bar{S}_g$
(8) **end if**

**算法 3 生成ML约束集合$M$和CL约束集合$C$**

输入：ACS集合$\{S_g\}_{g=1}^{G}$与AAS集合$\{\bar{S}_g\}_{g=1}^{G}$
输出：ML约束集合$M$，CL约束集合$C$
(1) 初始化集合$M=\varnothing$, $C=\varnothing$
(2) **for** $g=1$ **to** $G$ **do**
(3) **for** $\boldsymbol{x}_i,\boldsymbol{x}_j\in S_g$ **do**
(4) 将样本对$(\boldsymbol{x}_i,\boldsymbol{x}_j)$添加至集合$M$
(5) **end for**
(6) **for** $\boldsymbol{x}_i\in S_g$, $\boldsymbol{x}_j\in \bar{S}_g$ **do**
(7) 将样本对$(\boldsymbol{x}_i,\boldsymbol{x}_j)$添加至集合$C$
(8) **end for**
(9) **for** $k=g+1$ **to** $G$ **do**
(10) **for** $\boldsymbol{x}_i\in S_g$, $\boldsymbol{x}_j\in S_k$ **do**
(11) 将样本对$(\boldsymbol{x}_i,\boldsymbol{x}_j)$添加至集合$C$
(12) **end for**
(13) **end for**
(14) **end for**

序样本表示空间的聚类目标。本文采用文献[6]中的聚类损失，具体计算方式如式(2)－式(4)所示，其中式(2)中的$q_{ig}$表示时序样本$\boldsymbol{x}_i$的表示向量$\boldsymbol{z}_i$隶属于类簇中心$\boldsymbol{\mu}_g(1<g<G)$的概率；式(3)中的$p_{ig}$用于增强时序样本$\boldsymbol{x}_i$归属类簇$\boldsymbol{\mu}_g$的置信度。式(2)中$\alpha$是t-分布的自由度，文中设置为1

$$q_{ig}=\frac{\left(1+\left\|\boldsymbol{f}(\boldsymbol{x}_i;\theta)-\boldsymbol{\mu}_g\right\|_2^2/\alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{g'=1}^{G}\left(1+\left\|\boldsymbol{f}(\boldsymbol{x}_i;\theta)-\boldsymbol{\mu}_{g'}\right\|_2^2/\alpha\right)^{-\frac{\alpha+1}{2}}} \tag{2}$$

$$p_{ig}=\frac{q_{ig}^2/\sum_{g=1}^{N}q_{ig}}{\sum_{g'=1}^{G}\left(q_{ig'}^2/\sum_{i=1}^{N}q_{ig'}\right)} \tag{3}$$

$$L_1=\sum_{i=1}^{N}\sum_{g=1}^{G}p_{ig}\log\frac{p_{ig}}{q_{ig}} \tag{4}$$

约束损失$L_2$使满足ML约束的样本表示向量在表示空间中的距离变小，满足CL约束的样本表示向量在表示空间中的距离变大。其具体计算如式(5)所示，$\delta$用于约束CL约束的两个样本在表示空间中的最小距离。模型总损失$L_3$如式(6)所示，其中$\sigma$为超参数。由$L_3$优化模型参数$\theta$与类簇中心$\boldsymbol{\mu}_g$

$$L_2=\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in M}\left\|\boldsymbol{f}(\boldsymbol{x}_i;\theta)-\boldsymbol{f}(\boldsymbol{x}_j;\theta)\right\|_2^2+\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in C}\max\left(\delta-\left\|\boldsymbol{f}(\boldsymbol{x}_i;\theta)-\boldsymbol{f}(\boldsymbol{x}_j;\theta)\right\|_2,0\right)^2 \tag{5}$$

$$L_3=\sigma L_1+(1-\sigma)L_2 \tag{6}$$

### 3.6 模型训练过程

首先使用2.3节预训练的编码器E获取时序样本集$X$的表示向量集$Z$，由K-means算法对表示向量集$Z$进行聚类，得到初始类簇中心$\boldsymbol{\mu}_g(1<g<G)$。在训练的每个轮次根据式(2)将样本$\boldsymbol{x}_i$分配给类簇中心$\boldsymbol{\mu}_g$得到类簇$\pi_g(1<g<G)$。使用算法1从$\pi_g$中采样并标注距$\boldsymbol{\mu}_g$最近的样本存入ACS集合；当ACS集合$S_g(1<g<G)$中均至少包含一个时序样本时，将采样阶段标志SFlag设为False，由算法2从包含数量最少的ACS集合$S_g$中随机采样一个样本$\boldsymbol{x}_i$，并采样标注与$\boldsymbol{x}_i$不属于同一类簇且距$\boldsymbol{x}_i$所属类簇的中心最近的样本$\boldsymbol{x}_j$存入$S_g$或$\bar{S}_g$。采用算法3利用ACS集合与AAS集合中经人工采样标注的样本生成新的ML与CL约束，使用公式(6)计算模型总损失，优化编码器E的参数$\theta$与类簇中心$\boldsymbol{\mu}_g$，具体训练如算法4所示，其中$B$为深度主动聚类训练过程中总的标注次数，$b$为已训练轮次总共标注次数。

**算法 4 DATC-CT算法过程**

输入：数据集$X$，总预算$B$，当前预算$b$，训练轮次$T$，ACS集合$\{S_g\}_{g=1}^{G}$与AAS集合$\{\bar{S}_g\}_{g=1}^{G}$，主动采样阶段标志SFlag
输出：类簇中心$\{\boldsymbol{\mu}_g\}_{g=1}^{G}$
(1) 使用K-means方法对样本$\boldsymbol{x}_1,\boldsymbol{x}_2\cdots,\boldsymbol{x}_N$的表示向量$\boldsymbol{z}_1,\boldsymbol{z}_2\cdots,\boldsymbol{z}_N$聚类，得到初始类簇中心$\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_G$
(2) 初始化$S_1,S_2,\cdots,S_G=\varnothing$, $\bar{S}_1,\bar{S}_2,\cdots,\bar{S}_G=\varnothing$, SFlag=True, $b=0$
(3) **for** $t=1\ldots T$ **do**
(4) 使用TAE编码器获取样本$\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$嵌入$\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N$
(5) 使用式(2)计算样本类簇分布形成类簇$\pi_1,\pi_2,\cdots,\pi_G$
(6) **if** $b<B$ **then:**
(7) **if** SFlag **then:**
(8) 使用算法1更新$S_1,S_2,\cdots,S_G$
(9) **else:**
(10) 使用算法2更新$S_1,S_2,\cdots,S_G$与$\bar{S}_1,\bar{S}_2,\cdots,\bar{S}_G$
(11) **end if**
(12) **if** ACS集合$S_1,S_2,\cdots,S_G$中均至少包含1个样本 **then:**
(13) SFlag=False
(14) **end if**
(15) 使用算法3生成约束集合$M$与约束集合$C$
(16) 使用式(6)计算模型损失$L_3$
(17) 使用$L_3$更新TAE编码器参数与类簇中心$\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_G$
(18) **end for**

## 4 实验及结果分析

### 4.1 数据集与实验环境

实验所用数据集来自The UCR Time Series

Archive[25]，从中选取了18个具有代表性的单维时序数据集，各数据集样本数、类别数、序列长度信息如表1所示。实验的主要硬件和软件环境：Intel(R) Core(TM)i7-6700 CPU @ 3.40 GHz, NVIDIA Geforce GTX 4060，内存16 GB, Python版本为3.8，pytorch 2.0.0。

### 4.2 评价指标与参数设置

实验中使用兰德系数RI(Rand Index)评估聚类效果，其计算公式为

$$\mathrm{RI}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}} \tag{7}$$

其中，TP表示真实标签属于同类的两个样本在聚类结果中被分配到同一类簇中的数量，FP 表示真实标签属于不同类的两个样本在聚类结果中被分配到同一簇中的数量，FN表示真实标签属于同类的两个样本在聚类结果中被分配到不同类簇中的数量，TN表示真实标签属于不同类的两个样本被分配到不同类簇中的数量。

实验中每个数据集的类簇数目$G$与样本类别个数相同，式(6)超参数$\sigma$从{0.25, 0.5, 0.75}中选择最优兰德系数对应取值，最终$\sigma$设置为0.75。实验各个数据集中大部分样本经图2时序编码器生成的表示向量模取值在0～2，因此从{1, 1.5, 2}中选择最优兰德系数对应的式(5)中超参数$\delta$的取值，其中ECG200数据集上选1.5；在ECGFiveDays,GunPoint, MoteStrain和ToeSegmentation1数据集上选2；其余数据集上选择1。训练轮次$T$设置为300。实验选用的各数据集样本数量差异较大，未固定总预算取值，选取了各数据集样本总量的10%作为总预算$B$取值，采用查询样本真实标签完成人工标注过程，因此人工标注的聚类类别总是正确类别。

表 1　实验数据集描述

| 数据集 | 样本总数 | 类别数 | 序列长度 |
|---|---|---|---|
| ChlorineConcentration | 4307 | 3 | 166 |
| Dist.Phal.Outl.AgeGroup | 539 | 3 | 80 |
| Dist.Phal.Outl.Correct | 876 | 2 | 80 |
| ECG200 | 200 | 2 | 96 |
| ECGFiveDays | 884 | 2 | 136 |
| GunPoint | 200 | 2 | 150 |
| Ham | 214 | 2 | 431 |
| Herring | 128 | 2 | 512 |
| MoteStrain | 1272 | 2 | 84 |
| OSULeaf | 442 | 6 | 427 |
| Plane | 210 | 7 | 144 |
| Prox.phal.outl.AgeGroup | 605 | 3 | 80 |
| Prox.Phal.TW | 605 | 6 | 80 |
| SonyAIBORobotSurface1 | 621 | 2 | 70 |
| SonyAIBORobotSurface2 | 980 | 2 | 65 |
| SwedishLeaf | 1125 | 15 | 128 |
| ToeSegmentation1 | 268 | 2 | 277 |
| TwoLeadECG | 166 | 2 | 343 |

### 4.3 实验结果分析

#### 4.3.1 与已有深度时序聚类方法对比实验

选取DEC[6], DTC[7], DTCR[8] 3种无监督时序聚类方法进行对比。由于DEC模型中的编码器不适合处理时序数据，实验中将DEC模型编码器改为了与本文DATC-CT相同的编码网络。实验结果如表2所示，其中DTC与DTCR方法的实验结果来源于文献[8]，DATC-CT主动采样总预算为每个数据集样本总量的10%。由表2可知，DATC-CT平均RI值比DEC, DTC和DTCR方法分别提高了27.98%, 24.55%和5.91%，DATC-CT在14个实验数据集上均取得了较高的RI值这是因为DEC与DTC方法仅使用聚类损失作为优化目标与DTCR均未使用ML与CL约束将同类样本在表示空间中的距离拉近，不同类样本在表示空间中的距离拉远，因此聚类效果不佳。其中DEC与DTC方法仅使用聚类损失作为模型优化目标，DTCR模型除聚类损失外还

表 2　与相关深度时序聚类方法对比结果

| 数据集 | DEC | DTC | DTCR | DATC-CT |
|---|---|---|---|---|
| ChlorineConcentration | 0.4994 | 0.5353 | 0.5357 | **0.5370** |
| Dist.Phal.Outl.AgeGroup | 0.7321 | 0.7812 | 0.7825 | **0.7918** |
| Dist.Phal.Outl.Correct | 0.4998 | 0.5010 | **0.6075** | 0.5824 |
| ECG200 | 0.6308 | 0.6018 | 0.6648 | **0.7345** |
| ECGFiveDays | 0.5002 | 0.5016 | 0.9638 | **1.0000** |
| GunPoint | 0.4975 | 0.5400 | 0.6398 | **0.8219** |
| Ham | 0.4987 | 0.5648 | 0.5362 | **0.5841** |
| Herring | 0.4971 | 0.5045 | **0.5759** | 0.5153 |
| MoteStrain | 0.7621 | 0.5062 | 0.7686 | **0.9228** |
| OSULeaf | 0.7283 | 0.7329 | 0.7739 | **0.7742** |
| Plane | 0.9283 | 0.9040 | 0.9549 | **0.9921** |
| Prox.phal.outl.AgeGroup | 0.3893 | 0.7430 | 0.8091 | **0.8140** |
| Prox.Phal.TW | 0.5151 | 0.8380 | **0.9023** | 0.8568 |
| SonyAIBORobotSurface1 | 0.8988 | 0.5563 | 0.8769 | **0.9967** |
| SonyAIBORobotSurface2 | 0.6509 | 0.7012 | 0.8354 | **0.8562** |
| SwedishLeaf | 0.8837 | 0.8871 | 0.9223 | **0.9241** |
| ToeSegmentation1 | 0.4995 | 0.5077 | **0.5659** | 0.5158 |
| TwoLeadECG | 0.5001 | 0.5116 | 0.7114 | **1.0000** |
| 最优个数 | 0 | 0 | **4** | **14** |
| 平均RI | 0.6173 | 0.6343 | 0.7459 | **0.7900** |

引入了分类损失及重构损失，分类损失区分真实样本与由重新排列部分时间步取值生成的伪造样本，提升了编码器的表示能力。这也使得DTCR在Dist.Phal.Outl.Correct, Herring, Prox.Phal.TW, ToeSegmentation1 4个类别数量较多，序列长度较长的复杂数据集数据集上的聚类效果较好。

#### 4.3.2 主动选择策略对比

将文中DATC-CT主动采样策略与随机选择策略(记为Rand)与ADC[11]方法中的主动选择策略进行了实验对比，实验结果如表3所示。实验过程中均采用与本文DATC-CT相同的编码网络结构。

表 3 与已有主动选择策略对比结果

| 数据集 | Rand | ADC | DATC-CT |
|---|---|---|---|
| ChlorineConcentration | 0.5294 | 0.5342 | **0.5370** |
| Dist.Phal.Outl.AgeGroup | 0.7313 | 0.7218 | **0.7918** |
| Dist.Phal.Outl.Correct | 0.5019 | 0.5059 | **0.5824** |
| ECG200 | 0.6882 | 0.6699 | **0.7345** |
| ECGFiveDays | 0.6747 | 0.8311 | **1.0000** |
| GunPoint | 0.5754 | 0.5348 | **0.8219** |
| Ham | 0.5302 | 0.5173 | **0.5841** |
| Herring | 0.5062 | 0.5022 | **0.5153** |
| MoteStrain | 0.8300 | 0.8915 | **0.9228** |
| OSULeaf | 0.7443 | 0.7257 | **0.7742** |
| Plane | 0.9506 | 0.9819 | **0.9921** |
| Prox.phal.outl.AgeGroup | 0.7656 | 0.8047 | **0.8140** |
| Prox.Phal.TW | 0.8315 | 0.8566 | **0.8568** |
| SonyAIBORobotSurface1 | 0.9526 | 0.9744 | **0.9967** |
| SonyAIBORobotSurface2 | 0.8033 | 0.7588 | **0.8562** |
| SwedishLeaf | 0.9051 | 0.9128 | **0.9241** |
| ToeSegmentation1 | 0.4982 | 0.5029 | **0.5158** |
| TwoLeadECG | 0.5011 | 0.9982 | **1.0000** |
| 最优个数 | 0 | 0 | **18** |
| 平均RI | 0.6955 | 0.7347 | **0.7900** |

Rand随机选择策略在每个训练轮次中随机选择一对样本对进行人工标注，生成ML约束或CL约束；ADC方法在每个训练轮次从每个类簇中分别选择距类簇中心最近与最远的样本进行人工标注，生成与类簇数目相同的ML约束或CL约束。Rand, ADC与DATC-CT的主动采样总预算均为每个数据集样本总量的10%。由表3可知，DATC-CT平均RI值比Rand和ADC方法分别提高了13.59%和7.53%，这是因为在相同的标注成本下，DATC-CT方法利用设置的标注类簇集合与辅助标注集合推理成新的ML与CL约束，并使用人工标注及其生成的大量ML与CL约束指导聚类过程。Rand与ADC方法仅使用人工标注的少量ML与CL约束改善样本在表示空间中的聚类形态，未充分利用人工标注推理生成更多的ML与CL约束。另一方面，Rand方法在表示空间中随机标注两个时序特征对应的样本，难以生成高质量的ML与CL约束，而ADC方法在仅表示空间的类簇内采样，使得生成CL约束的可能性降低。

图3展示了Rand, ADC和DATC-CT方法在MoteStrain, Prox.phal.outl.AgeGroup和SwedishLeaf数据集上，RI值随人工标注成本的变化情况。图3中横轴为人工标注的总预算，依次设置为40, 80, 120, 160和200，纵轴为上述3种方法的RI值，由图3可知，随着标注成本的增加，上述3种主动聚类方法所取得的RI值均有提高。相同标注预算下，DATC-CT方法在上述3个数据集上的RI值优于Rand与ADC方法，这是因为DATC-CT方法推理生成了大量ML与CL约束。图4展示了DATC-CT方法在MoteStrain, Prox.phal.outl.AgeGroup和SwedishLeaf数据集上推理生成的约束对个数与人工标注的约束对个数间的关系，图中横轴为人工标注的约束个数，纵轴为DATC-CT方法经推理生成新约束后约束的总个数。由图4可知，DATC-CT方法由少量人工标注的约束对生成了大量的ML与CL约束对。

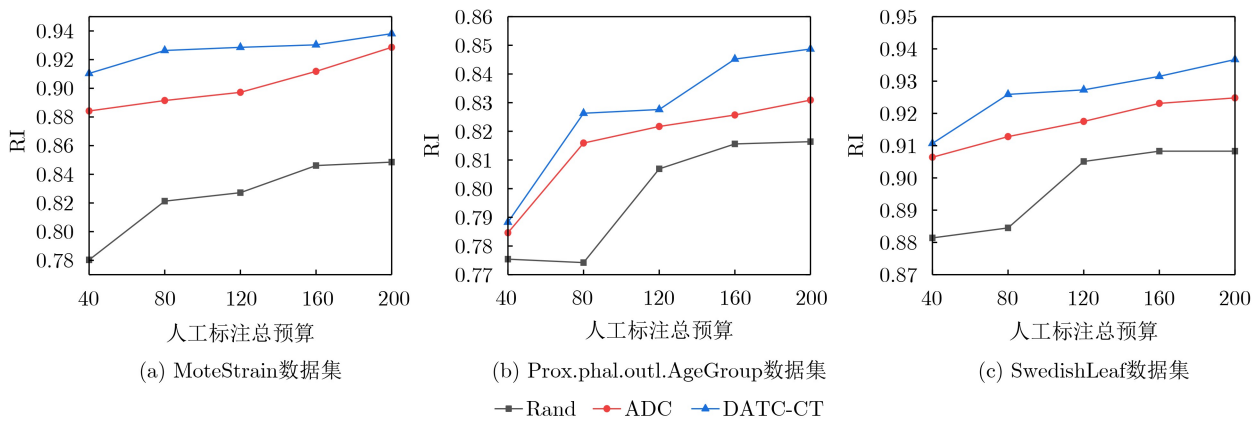
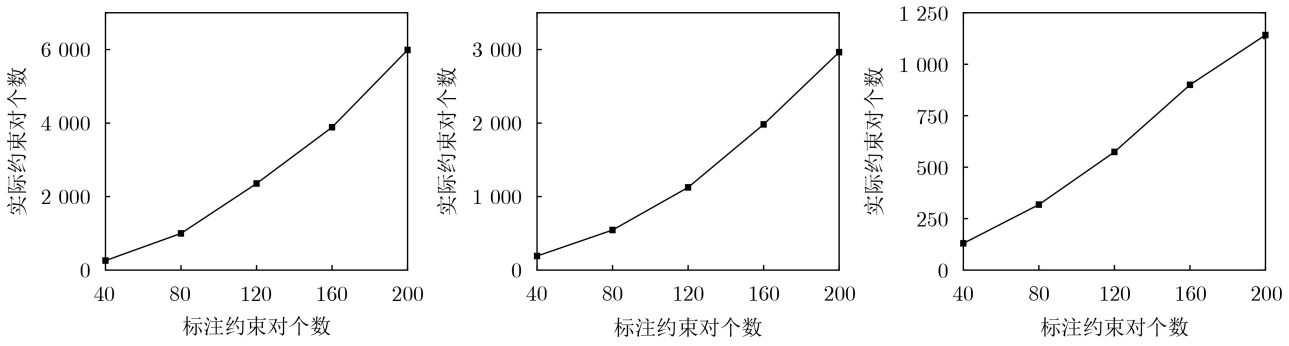



图 3 不同标注成本下主动聚类方法的RI值




(a) MoteStrain数据集　(b) Prox.phal.outl.AgeGroup数据集　(c) SwedishLeaf数据集

图 4 DATC-CT方法推理生成的ML与CL约束对的数量

## 5　结论

本文提出了一种深度主动时序聚类模型，本模型使用基于一维卷积与双向长短时记忆网络的自编码器学习时序表示特征，设置了与聚类数目相同的标注类簇集合与辅助标注集合，用于存放利于聚类任务的人工标注样本，在每个训练轮次中从表示空间的每个类簇中选择距类簇中心最近的样本存入标注类簇集合；从包含样本数量最少的标注类簇集合中随机选择一个样本，采样并标注与该样本不属于同一类簇且距其在表示空间所在类簇中心最近的时序样本存入辅助标注集合，由标注类簇集合与辅助标注集合中的样本推理生成ML约束与CL约束。在18个公开数据集上的实验表明，文中的模型能够在较低的标注预算下提高深度主动时序聚类效果。

## 参 考 文 献

[1] AIGNER W, MIKSCH S, SCHUMANN H, *et al*. Visualization of Time-Oriented Data[M]. London: Springer, 2011: 153–196.

[2] LEE S, CHOI C, and SON Y. Deep time-series clustering via latent representation alignment[J]. *Knowledge-Based Systems*, 2024, 303: 112434. doi: 10.1016/j.knosys.2024.112434.

[3] CAI Jinyu, FAN Jicong, GUO Wenzhong, *et al*. Efficient deep embedded subspace clustering[C]. The 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, USA, 2022: 21–30. doi: 10.1109/cvpr52688.2022.00012.

[4] ZHONG Ying, HUANG Dong, and WANG Changdong. Deep temporal contrastive clustering[J]. *Neural Processing Letters*, 2023, 55(6): 7869–7885. doi: 10.1007/s11063-023-11287-0.

[5] ROS F, RIAD R, and GUILLAUME S. Deep clustering framework review using multicriteria evaluation[J]. *Knowledge-Based Systems*, 2024, 285: 111315. doi: 10.1016/j.knosys.2023.111315.

[6] XIE Junyuan, GIRSHICK R, and FARHADI A. Unsupervised deep embedding for clustering analysis[C]. The 33rd International Conference on International Conference on Machine Learning, New York, USA, 2016: 478–487.

[7] MADIRAJU N S. Deep temporal clustering: Fully unsupervised learning of time-domain features[D]. [Ph. D. dissertation], Arizona State University, 2018.

[8] MA Qianli, ZHENG Jiawei, LI Sen, *et al*. Learning representations for time series clustering[C]. The 33rd International Conference on Neural Information Processing Systems, Red Hook, USA, 2019: 3781–3791.

[9] PENG Furong, LUO Jiachen, LU Xuan, *et al*. Cross-domain contrastive learning for time series clustering[C]. The AAAI Conference on Artificial Intelligence, Vancouver, Canada, 2024: 8921–8929. doi: 10.1609/aaai.v38i8.28740.

[10] LI Xiaosheng, XI Wenjie, and LIN J. Randomnet: Clustering time series using untrained deep neural networks[J]. *Data Mining and Knowledge Discovery*, 2024, 38(6): 3473–3502. doi: 10.1007/s10618-024-01048-5.

[11] SUN Bicheng, ZHOU Peng, DU Liang, *et al*. Active deep image clustering[J]. *Knowledge-Based Systems*, 2022, 252: 109346. doi: 10.1016/j.knosys.2022.109346.

[12] REN Yazhou, HU Kangrong, DAI Xinyi, *et al*. Semi-supervised deep embedded clustering[J]. *Neurocomputing*, 2019, 325: 121–130. doi: 10.1016/j.neucom.2018.10.016.

[13] REN Pengzhen, XIAO Yun, CHANG Xiaojun, *et al*. A survey of deep active learning[J]. *ACM Computing Surveys (CSUR)*, 2022, 54(9): 180. doi: 10.1145/3472291.

[14] DENG Xun, LIU Junlong, ZHONG Han, *et al*. A3S: A general active clustering method with pairwise constraints[C]. The 41st International Conference on Machine Learning, Vienna, Austria, 2024: 10488–10505.

[15] 李海林, 张丽萍. 时间序列数据挖掘中的聚类研究综述[J]. 电子科技大学学报, 2022, 51(3): 416–424. doi: 10.12178/1001-0548.2022055.

LI Hailin and ZHANG Liping. Summary of clustering

research in time series data mining[J]. *Journal of University of Electronic Science and Technology of China*, 2022, 51(3): 416–424. doi: 10.12178/1001-0548.2022055.

[16] PETITJEAN F, KETTERLIN A, and GANÇARSKI P. A global averaging method for dynamic time warping, with applications to clustering[J]. *Pattern Recognition*, 2011, 44(3): 678–693. doi: 10.1016/j.patcog.2010.09.013.

[17] PAPARRIZOS J and GRAVANO L. Fast and accurate time-series clustering[J]. *ACM Transactions on Database Systems (TODS)*, 2017, 42(2): 8. doi: 10.1145/3044711.

[18] ZAKARIA J, MUEEN A, and KEOGH E. Clustering time series using unsupervised-shapelets[C]. 2012 IEEE 12th International Conference on Data Mining, Brussels, Belgium, 2012: 785–794. doi: 10.1109/icdm.2012.26.

[19] 余思琴, 闫秋艳, 闫欣鸣. 基于最佳u-shapelets的时间序列聚类算法[J]. 计算机应用, 2017, 37(8): 2349–2356. doi: 10.11772/j.issn.1001-9081.2017.08.2349.
YU Siqin, YAN Qiuyan, and YAN Xinming. Clustering algorithm of time series with optimal u-shapelets[J]. *Journal of Computer Applications*, 2017, 37(8): 2349–2356. doi: 10.11772/j.issn.1001-9081.2017.08.2349.

[20] MACQUEEN J B. Some methods for classification and analysis of multivariate observations[C]. The 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, USA, 1967: 281–297.

[21] JOHNSON S C. Hierarchical clustering schemes[J]. *Psychometrika*, 1967, 32(3): 241–254. doi: 10.1007/bf02289588.

[22] NG A Y, JORDAN M I, and WEISS Y. On spectral clustering: Analysis and an algorithm[C]. The 15th International Conference on Neural Information Processing Systems: Natural and Synthetic, Cambridge, USA, 2001: 849–856.

[23] CHANG Shiyu, ZHANG Yang, HAN Wei, *et al*. Dilated recurrent neural networks[C]. The 31st International Conference on Neural Information Processing Systems, Red Hook, USA, 2017: 76–86.

[24] BASU S, BANERJEE A, and MOONEY R J. Active semi-supervision for pairwise constrained clustering[C]. The 2004 SIAM International Conference on Data Mining, Lake Buena Vista, USA, 2004: 333–344. doi: 10.1137/1.9781611972740.31.

[25] DAU H A, BAGNALL A, KAMGAR K, *et al*. The UCR time series archive[J]. *IEEE/CAA Journal of Automatica Sinica*, 2019, 6(6): 1293–1305. doi: 10.1109/jas.2019.1911747.

霍纬纲：男，教授，研究方向为时序大数据分析、数据挖掘.
朱 旭：男，硕士生，研究方向为时间序列分析.
张 盼：女，硕士生，研究方向为时间序列分析.

责任编辑：余 蓉

# Deep Active Time-series Clustering Based on Constraint Transitivity

HUO Weigang ZHU Xu ZHANG Pan

(*School of Computer Science and Technology, Civil Aviation University of China, Tianjin* 300300, *China*)

**Abstract**:

**Objective** The rapid advancement of the Internet of Things and sensor technology has led to the accumulation of vast amounts of unlabeled time-series data, making deep time-series clustering a key analytical approach. However, existing deep clustering methods lack supervised constraint information and label guidance, making them susceptible to noise and outliers. Deep semi-supervised clustering methods rely on predefined Must-Link (ML) and Cannot-Link (CL) constraints, limiting improvements in clustering performance. Existing active clustering approaches sample only within clusters in the representation space, overlooking pairwise annotations from different clusters. This results in lower-quality ML and CL constraints and prevents further extrapolation from manually annotated pairs, increasing annotation costs. To address these limitations, this paper proposes Deep Active Time-series Clustering based on Constraint Transitivity (DATC-CT), which improves clustering performance while reducing annotation costs.

**Methods** DATC-CT defines an Annotation Cluster Set (ACS) and an Auxiliary Annotation Set (AAS) and obtains the representation vector of time-series samples using a pre-trained autoencoder. In each clustering epoch, samples closest to cluster centers in the representation space are selected, labeled, and stored in ACS. This ensures that all samples within an ACS belong to the same category, while those in different ACSs belong to different categories. Next, a time-series sample is randomly chosen from the ACS with the fewest samples. Another sample, which does not belong to the same cluster but is nearest to the selected sample's center, is

then sampled, labeled, and stored in either AAS or ACS. Samples in ACS and AAS belong to different categories. ML and CL constraints are inferred from these samples. The encoder's network parameters and cluster centers are updated using KL divergence between the cluster distribution (modeled by a t-distribution) and an auxiliary distribution generated from it. Additionally, a constraint loss function is applied to increase the distance between ML-constrained samples while reducing the distance between CL-constrained samples in the representation space.

**Results and Discussions** Experimental results on 18 public datasets show that the proposed method improves the average Rand Index (RI) by more than 5% compared to existing deep time-series clustering methods (Table 2). With the same labeling budget, it achieves an RI improvement of over 7% compared to existing active clustering methods (Table 3). These findings confirm the effectiveness of the active sampling strategy and constraint reasoning mechanism. Additionally, the method infers a large number of ML and CL constraints from a small set of manually annotated constraints (Fig. 4), significantly reducing annotation costs.

**Conclusions** This paper proposes a deep active time-series clustering model based on constraint transitivity, incorporating a two-phase active sampling strategy: exploration and consolidation. In the exploration phase, the model selects the sample closest to each cluster center in the representation space and stores it in ACS. During consolidation, a sample is randomly chosen from the ACS with the fewest samples. Another sample, which does not belong to the same cluster but is nearest to the selected sample's center, is then sampled, labeled, and stored in either AAS or ACS. The number of ACS and AAS matches the number of clusters. ML and CL constraints are inferred from ACS and AAS samples. Experiments on public datasets demonstrate that inferring new clustering constraints reduces annotation costs and improves deep time-series clustering performance.

**Key words**: Deep time series clustering; Active learning; Constraint transitivity