

时间序列语义分割的综合研究的项目报告

古昊

Abstract—这份项目报告是对时间序列语义分割的综合研究。包括了论述语义分割的必要性、对不同算法性能的复现与应用场景的探究。

Index Terms—时间序列语义分割 (TSS)，转折点

I. 引言

近年来，低成本、高分辨率传感器的应用呈爆炸式增长，例如移动设备、制造监控或环境和医疗监控。这些传感器生成大量未标记的时间排序实值序列，也称为数据序列或时间序列 (TS)。这些数据背后蕴含着复杂的状态转换与语义信息，而“分割”作为解析数据结构的核心技术，已成为连接原始信号与智能决策的关键桥梁。因此，时间序列分离 (TSS) 试图事后在 TS 中发现此类变化，以推断数据生成过程中的变化。TSS 通常被视为一个无监督学习问题，旨在识别可通过某些统计特性区分的序列。因此，在这片报告中，我会充分的讨论有关于时间序列语义分割的必要性，以及不同算法之间运用的策略性问题。

II. 时间序列语义分割的必要性

时间序列分割 (TSS) 的问题是找到一个有意义的时间序列分割，以捕捉具有不同状态和转换的数据生成过程。此类信号捕获了过程的内在统计特性和时间模式。TS 中这些属性的突然偏移或变化点 (CP) 表示数据生成过程中的状态转换。这也就意味着我们可以充分提取数据集中的特征信息，理解数据内在结构，排除冗余的噪声，在医疗等方面起到重要的意义。这里引用 HanShuang Xie 的一篇报告作为举例¹。通常，心电图信号被单个或三个心跳分割，这会导致心跳之间的信息丢失。但是，如果 ECG 信号太长，则会出现数据冗余问题，从而导致计算过多。因此，在客观反映信号的基本心率，保留心跳的节律信息的基础上，选择对时间长度为 30 秒的心电信号进行分割比较合适。该例子充分说明，对像心电图这样的时间序列进行语义分割，不仅能对特征信息进行更正确的识别，还能提高运算效率。

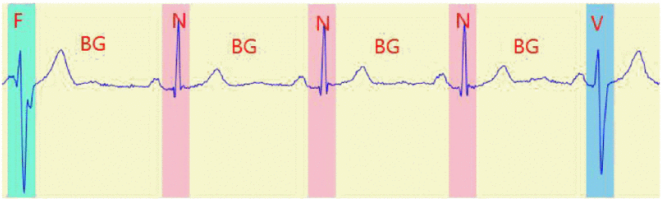


Fig. 1: 心电图分割示例

III. 数据库以及前期准备

A. 数据库

这里，我们采用了作者 Ermshaus 的时间序列数据库²进行比对测试。

B. 综合覆盖率分数

- *covering* 是数据库中包含的用来计算综合覆盖率分数的函数。
- 影响综合覆盖率分数的因数有：**预测变化点的数量、预测变化点的位置**

C. 基准综合覆盖率分数参考值

这里我们用空的变化点列表来测试每一组数据，得到一个平均的基准综合覆盖率分数参考值。这个参考值反映了不采取任何的语义分割操作能得到的基准综合覆盖率分数。这个参考值为：*0.4010*。

也就是说，如果语义分割算法得到的综合覆盖率分数如果小于等于这个参考值，那么该算法不能正确且有效的分割语义，甚至起到模糊数据特征等负面作用。

IV. 有超参数语义分割算法

有超参数时间序列语义分割算法需要人工设置关键参数（如窗口长度、变化点数量、距离度量阈值等）才能完成分割任务的算法。这类算法在特定场景下通过合理调参可实现高精度分割。这里，我们列举了两个经典的算法，同时也进行了相对应的分析。

A. FLUSS³

FLUSS (Fast Low-cost Unipotent Semantic Segmentation) 是 STUMPY 库中基于矩阵轮廓的时间序列语义分割算法，核心在于通过矩阵轮廓快速定位时间序列中的状态转换点，实现无监督的语义分段。在电商实时营销和工业物联网监控 具有优势。

FLUSS 具有以下特点。

- 轻量级与高效性：计算复杂度为 $O(n)$
- 领域无关的通用性：不假设时间序列的分布或变化类型（如突变 / 渐变），通过数据驱动的弧曲线自适应识别语义段

1) 准确率与局部分析：对于整个数据集，该算法有平均综合覆盖率分数：0.6859

- 优势领域：数据集具有低噪声和周期性或重复性。同时，长时间序列提供了足够的上下文信息，便于算法检测变化点。

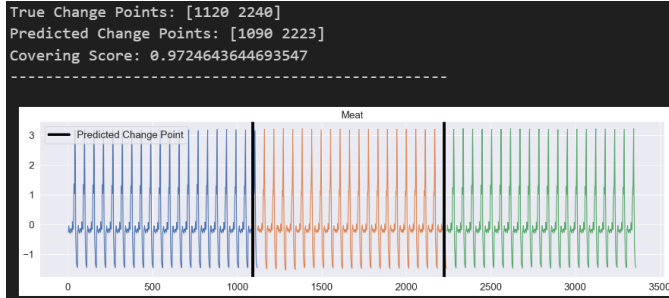


Fig. 2: FLUSS 的优势领域

总的来说，FLUSS 作为 STUMPY 库的核心模块，通过矩阵轮廓与弧曲线的轻量级计算，实现了高效的时间序列语义分割。其优势在于计算速度快、跨领域通用性强，适合对实时性和解释性要求高 同时，在特定的条件下，既是用默认的超参数进行调试也能取得非常高的正确率。

B. BINSEG⁴

BINSEG (Binary Segmentation) 是离线变点检测中最经典的算法之一，属于递归二分法框架。其核心逻辑是通过迭代将时间序列划分为子序列，每次寻找最优分割点，直到满足停止条件。该算法在时间序列异常检测、趋势分析等场景中广泛应用，例如在金融市场波动检测、工业设备故障预警等领域。

BINSEG 具有以下特点。

- 计算效率高：计算复杂度为 $O(n \log n)$

- 灵活性强：可自定义成本函数，适配不同分布的数据。

1) 准确率与局部分析：以 BINSEG 算法为基础，分别对 Binseg、BottomUp 和 PELT 模型进行对比测试。以下分别展示不同模型下的平均综合覆盖率分数：

- Binseg: 0.4033
- BottomUp: 0.5277
- PELT: 0.2384

根据上面数据可知，PELT 的平均综合覆盖率分数过低，足以证明：在该模型下，语义分割能力极差，甚至起到负面作用，甚至分割效果远低于不采取任何分割措施。对于默认超参数情况下的 Binseg 和 PELT 模型，无法得到有效的语义分割，需要进一步调整如窗口大小等超参数。

- 优势领域：数据集具有显著的模式变化和线性或平稳特性。变化点前后具有显著的模式差异，例如均值、方差、频率等特征的变化能够放大成本函数。平稳特性，便于 L2 范数捕捉段内一致性。

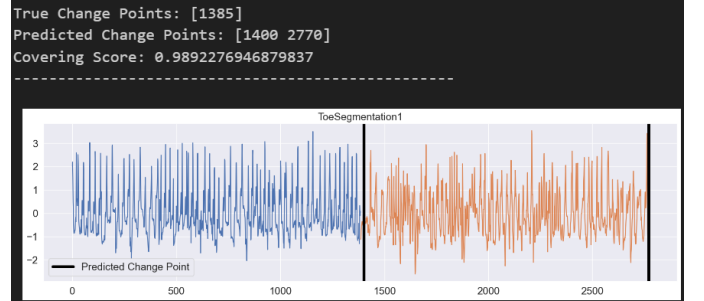


Fig. 3: BINSEG (bottomup 方法) 的优势领域

2) 数学模型：通过定义一个能量化“异质性”的成本函数，算法试图找到一组变点，使得所有分段的总成本最小，从而实现序列的最优分割。这里用 L2 范数为例：

$$\text{cost}(S_1, S_2) = \sum_{x \in S_1} (x - \mu_1)^2 + \sum_{x \in S_2} (x - \mu_2)^2 \quad (1)$$

符号解析：

- (S_1, S_2) : 时间序列被分割后的两个子段（如 $S_1 = [x_1, x_2, \dots, x_k]$, $S_2 = [x_{k+1}, \dots, x_n]$ ）。
- (μ_1, μ_2) : 对应子段的均值，即 $\mu_1 = \frac{1}{|S_1|} \sum_{x \in S_1} x$, $\mu_2 = \frac{1}{|S_2|} \sum_{x \in S_2} x$ 。
- $(x - \mu)^2$: 单个数据点与均值的平方误差，求和后反映整个子段数据的离散程度。

总的来说, BINSEG 作为离线变点检测的经典算法, 凭借高效的递归二分逻辑和灵活的成本函数设计, 在时间序列分析中占据重要地位。尽管存在参数依赖性和局部最优问题, 但其在长序列处理和实时性要求较高的场景中仍不可替代。

C. 总结

虽然 FLUSS、BINSEG 等有超参数算法在计算效率和数学模型可解释性上有巨大的优势, 但默认下的超参数往往只能对部分有特定特征的数据结构起到高效的语义分割。因此, 有超参数算法通常需要结合全局搜索、自适应参数调整和多成本函数融合。

V. 无超参数语义分割

无超参数时间序列语义分割算法无需人工设置关键参数 (如窗口长度、变化点数量等), 能自动从数据中学习最优分割策略的算法。相较于上面的有超参数的语义分割算法, 无超参数极大的提高了算法的部署效率和对抗复杂极端数据集的强鲁棒性。

A. ClaSP 算法⁵

这里我们选用 ClaSP 算法作为测试对象, 这是一种新颖的、高精度的、超参数的、与域无关的 TSS 方法²。ClaSP 按层次结构将 TS 拆分为两部分。通过为每个可能的分割点训练一个二进制 TS 分类器并选择最擅长识别来自任一分区的子序列的分割来确定变化点。

ClaSP 具有以下特点。

- 计算复杂度为 $O(C \cdot n^2)$, 在预计算 k-NN 轮廓后, 每次分割点评估时间复杂度为 $O(n)$,
- 强抗噪声能力
- 超参数自学习机制

1) 算法逻辑:

- 1) 通过滑动窗口生成子序列, 构建自监督二分类数据集。
- 2) 用 k-NN 分类器对 k 近邻轮廓预计算。
- 3) 交叉验证并计算分类得分
- 4) 窗口大小自学习和变化点显著性检验

2) 准确率与局部分析:

- 优势领域: 具有显著差异的“状态段”组成, 且状态间存在明确的转换点的多模态时间序列。或

者数据整体不满足平稳性假设, 包含趋势、突变、周期嵌套等复杂模式。

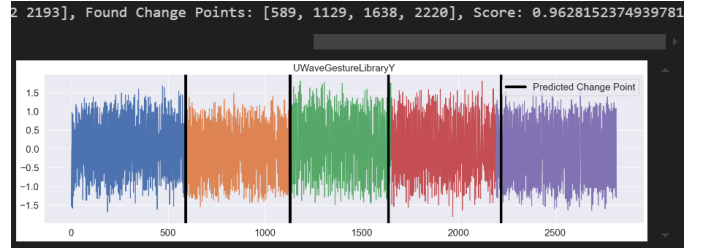


Fig. 4: ClaSP 的优势领域

- 鲁棒性: 同时我们可以注意到, 相较于上面的有超参数算法在面对不同特征的数据结构时存在较大的语义分割能力差异, ClaSP 算法得到的综合覆盖率分数具有较强的一致性, 体现在不同数据组的覆盖率分数方差: **0.0358**, 远低于 FLUSS 算法与 BINSEG 算法得到的覆盖率分数方差, 分别为: **0.0525**、**0.0528**、**0.0423**。足以证明, ClaSP 算法有很强的鲁棒性。

3) 总结: ClaSP 算法通过将时间序列分割转化为分类问题, 利用自监督学习和统计检验实现无超参数的语义分割, 适用于医疗、工业、金融等领域的时间序列状态分析。当然也存在一些局限性, 包括: 长序列计算成本高, 对百万级数据点的序列处理效率较低; 对渐变趋势识别不足。

VI. 结论

时间序列语义分割的核心目标是定位数据中的“状态转换点”, 将序列划分为具有同质特征的语义段。算法适用的领域和效果是多样的, BINSEG 和 FLUSS 凭借高效性在特定领域不可替代, 而 ClaSP 通过无超参数设计引领跨领域泛化。未来, 算法将更注重效率与鲁棒性的平衡, 做到多策略融合和技术创新。

REFERENCES

- [1] H. Xie, M. Zheng, H. Zhu, F. Wu and Y. Pan, "Arrhythmia Detection Based on Semantic Segmentation for Multi-lead ECG," 2022 Computing in Cardiology (CinC), Tampere, Finland, 2022, pp. 1-4, doi: 10.22489/CinC.2022.173. keywords: Training;Sensitivity;Heart beat;Databases;Semantic segmentation;Arrhythmia;Electrocardiography
- [2] A. Ermshaus, "time-series-segmentation-benchmark" [Source code], GitHub, 2023. [Online]. Available: <https://github.com/ermshaus/time-series-segmentation-benchmark.git>. Accessed: Aug. 1, 2023.

- [3] S. M. Law et al., "STUMPY: A Python Library for Time Series Analysis," 2023. [Online]. Available: <https://stumpy.readthedocs.io>. [Accessed: 2025-07-06].
- [4] TechRando, "A Brief Introduction to Change Point Detection Using Python," TechRando, Aug. 14, 2019. [Online]. Available: <https://techrando.com/2019/08/14/a-brief-introduction-to-change-point-detection-using-python/>. [Accessed: 2025-07-06].
- [5] A. Ermshaus, P. Schäfer, and U. Leser, "*ClaSP: Parameter-Free Time Series Segmentation*," *Data Mining and Knowledge Discovery*, vol. 37, no. 2, pp. 1001 – 1020, 2023.