Accelerating k-Shape Time Series Clustering Algorithm Using GPU

Ritu Patel, Sarthak Siddhpura, Moin Vinchhi, Vrunda Patel, Vishv Boda
Ahmedabad University, Ahmedabad, Gujarat 380009, India

I. Research Contributions by Paper

- Proposed a novel GPU-based parallel algorithm, Times-C, for accelerating the k-Shape time series clustering algorithm, achieving a performance improvement of one to two orders of magnitude over the multi-core CPU version.
- Introduced a parallel data hash sorting method to enhance data aggregation efficiency, optimizing memory access and improving data locality for high-dimensional time series data.
- Developed a two-level parallel structure (intra-class and inter-class parallelism) using CUDA streams to compute class centroids efficiently, targeting eigenvectors corresponding to maximum eigenvalues.
- Designed a parallel frequency domain similarity calculation method utilizing three-dimensional thread parallelism to compute cross-correlation measures rapidly.
- Conducted extensive experiments on benchmark datasets (e.g., UCR, UAE, urbansound8k), demonstrating significant speedups over existing implementations like tslearn and k-Shape-GPU, particularly on large datasets.
- Provided open-source implementations of Times-C in both GPU (C++/CUDA) and CPU (Python) versions, enhancing accessibility for further research and application.

II. Key Insights From Paper

The paper by Wang et al. presents a significant advancement in time series clustering through the Times-C algorithm, which leverages GPU parallelism to accelerate the k-Shape algorithm, a method known for its effectiveness in shape-based clustering. This is particularly relevant to our research direction of identifying abnormal driving behavior using trajectory datasets, where spatial and temporal features must be analyzed efficiently. The assumption in our project—that drivers behave similarly on the same road patch but anomalies exist—aligns well with the k-Shape algorithm's ability to cluster time series based on shape similarity while being robust to temporal misalignments. Times-C's parallel data aggregation and alignment phase, which uses a hash sorting method to group similar trajectories, could be adapted to cluster vehicle trajectories from a road segment, enabling the identification of normal driving patterns. By representing trajectories as time series (e.g., sequences of position, speed, or acceleration), we can apply Times-C to group similar behaviors efficiently, providing a foundation for a binary classifier to distinguish normal from abnormal trajectories based on deviations from cluster centroids.

A key insight from the paper is the use of GPU parallelism to handle large-scale, high-dimensional datasets, which is critical for processing real-world trajectory data that often involves thousands of vehicles over extended periods. The two-level parallel structure for centroid computation—splitting tasks into intra-class and inter-class parallelism—offers a scalable approach to manage the computational complexity of clustering trajectories with varying lengths and sampling rates. In our context, this could translate to clustering trajectories from different drivers on a road patch, where intra-class parallelism handles the alignment and feature extraction within a single driver's data, and inter-class parallelism compares across drivers to define typical behavior. The paper's emphasis on frequency domain similarity calculation using Fast Fourier Transforms (FFT) further enhances efficiency by reducing the time complexity from O(m^2) to O($m \log m$), where m is the sequence length. For trajectory analysis, this means faster computation of similarity between a given trajectory and the cluster centroid, crucial for real-time anomaly detection in driving behavior. The experimental results showing speedups on datasets like InsectSound (139,883 time series) suggest that Times-C can handle the volume of trajectory data expected in urban settings, making it a promising tool for our probabilistic and statistical modeling goals.

Moreover, the paper's focus on optimizing memory access and SM occupancy on GPUs provides practical insights for implementing our binary classifier. By adapting Times-C's hash sorting for aggregation, we could preprocess trajectory data to ensure spatial and temporal coherence, enhancing the classifier's ability to detect outliers—such as sudden lane changes or erratic speeds—that deviate from the norm. The stability of Times-C across varying numbers of clusters (k) is another valuable insight, as it implies robustness in defining normal behavior even when the number of distinct driving patterns is uncertain. This flexibility is essential in our research, where road conditions or driver demographics might influence the number of clusters. While the paper does not directly address binary classification, its clustering framework can serve as a feature extraction step, feeding into a statistical model (e.g., logistic regression or SVM) to classify trajectories. The open-source availability of Times-C also facilitates experimentation with trajectory datasets, allowing us to test its efficacy in capturing spatial-temporal anomalies and potentially integrating it with probabilistic methods to quantify

the likelihood of abnormal behavior, aligning with our prob-stat focus.

III. Unaddressed Issues and Assumptions

- **Unaddressed Issues:**
  - The paper does not explore the integration of Times-C with supervised learning techniques, such as binary classification, limiting its direct applicability to anomaly detection tasks like ours.
  - Scalability to real-time processing is not evaluated, which is critical for detecting abnormal driving behavior in dynamic environments.
  - The impact of noisy or incomplete time series data (e.g., missing GPS points in trajectories) on clustering accuracy and performance is not addressed.
  - The paper lacks discussion on handling multi-modal data (e.g., combining trajectory with contextual data like weather or traffic), which could enhance anomaly detection in driving behavior.
- **Assumptions Made:**
  - Assumes time series data is relatively clean and complete, which may not hold for trajectory datasets with GPS noise or interruptions.
  - Assumes that shape-based similarity (via cross-correlation) is sufficient for clustering, potentially overlooking other relevant features like absolute speed or direction changes in driving trajectories.
  - Assumes uniform computational resources (e.g., NVIDIA A100 GPU), which may not be available in all research or deployment settings.
  - Assumes the number of clusters (k) is predefined or easily determined, whereas in driving behavior analysis, the optimal k may vary by road segment or context.

IV. Motivation Behind Choosing This Paper

The motivation for selecting this paper stems from its innovative approach to accelerating time series clustering, a critical step in our research to identify abnormal driving behavior using trajectory datasets. The Times-C algorithm's ability to efficiently cluster large-scale, high-dimensional time series data using GPU parallelism directly addresses the computational challenges we face in processing extensive vehicle trajectory data. Given our prob-stat focus, the paper's enhancement of the k-Shape algorithm—known for its statistical robustness in capturing shape-based patterns—provides a scalable foundation for defining normal driving behavior on road patches, enabling subsequent anomaly detection. The open-source availability of Times-C further motivates its selection, as it allows us to adapt and extend the algorithm to our specific needs, such as integrating spatial-temporal features or interfacing with a binary classifier. Moreover, the paper's demonstrated performance gains on datasets analogous to our expected trajectory volumes (e.g., urbanSound8k, MosquitoSound) suggest its potential to handle the scale and complexity of real-world driving data, making it a compelling choice to advance our research objectives.

References