Towards Similarity-Aware Time-Series Classification

Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, Xia Hu

Department of Computer Science, Rice University

Time-Series Classification (TSC)

- **General Goal.** Given a collection of time-series with the attached labels, TSC aims to train a classifier to classify unseen time-series.
- **Univariate Time-series.** A univariate time-series $\mathbf{x}$ of length T is represented as a vector $[x_1, x_2, ..., x_T]$.
- **Multivariate Time-Series.** An M-dimensional time-series $\mathbf{X}$ consists of M univariate time-series $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M]$. We regard a univariate time-series $\mathbf{x}$ as a special case of multivariate time-series, i.e., a 1-dimensional time-series $\mathbf{X} \in \mathbb{R}^{1 \times T}$.
- **Problem Formulation.** Given some testing time-series $\mathcal{X}^{\text{test}} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_{N^{\text{test}}}]$ and the labels $\mathbf{y}^{\text{test}} = [y_1, y_2, ..., y_{N^{\text{test}}}]$, where N^{test} is the number of testing time-series, we aim to train a classifier that can predict the labels based on $\mathcal{X}^{\text{test}}$ under one of the following settings:
  - **Supervised setting**: The classifier is trained based on a training time-series dataset $\mathcal{X}^{\text{train}} = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_{N^{\text{train}}}]$ and its corresponding labels $\mathbf{y}^{\text{train}} = [y_1, y_2, ..., y_{N^{\text{train}}}]$, where N^{train} is the number of training time-series.
  - **Inductive semi-supervised setting**: In addition to $\mathcal{X}^{\text{train}}$ and $\mathbf{y}^{\text{train}}$, the classifier can also access some unlabeled time-series $\mathcal{X}^{\text{unlabeled}}$, which does not overlap with $\mathcal{X}^{\text{test}}$.
  - **Transductive semi-supervised setting**: In addition to $\mathcal{X}^{\text{train}}$, $\mathbf{y}^{\text{train}}$ and $\mathcal{X}^{\text{unlabeled}}$, the classifier is exposed to testing time-series $\mathcal{X}^{\text{test}}$. Note that $\mathbf{y}^{\text{test}}$ is not accessible in training.

Existing Work

The existing work approaches TSC problem in two major directions:

- **Similarity-based methods:** Combine a k-NN classifier with a similarity measure for classification.
- **Deep Learning:** Perform end-to-end training on the raw time-series and learn the representations to do classification.

DTW vs ResNet

We compare DTW (a representative similarity-based method) and ResNet (a representative deep learning approach) on the full 128 UCR datasets (benchmarks in this domain). We report the average ranks. The lower the better.

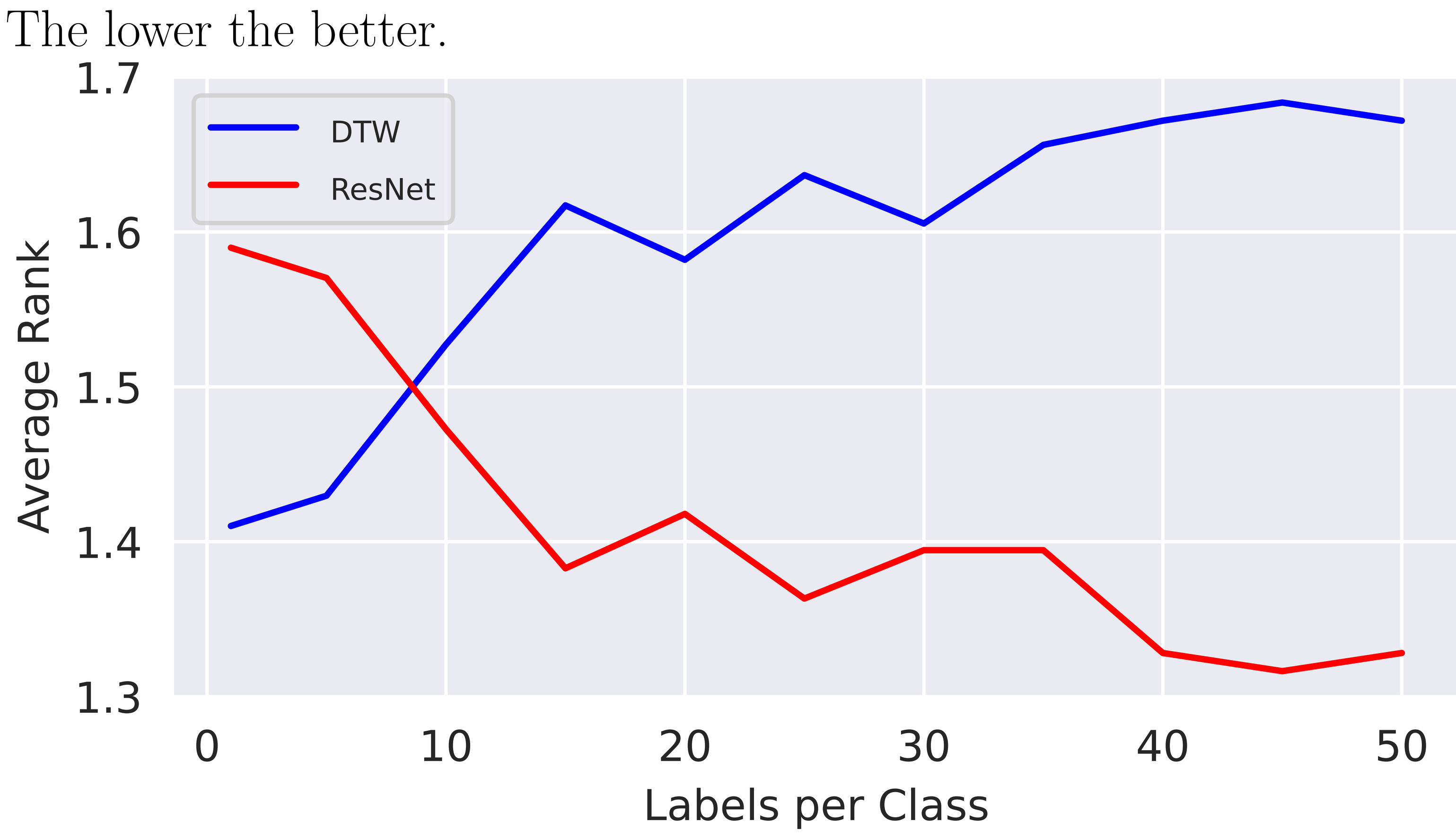


Figure: Average ranks (↓) of ResNet and DTW on the full 128 UCR datasets, where different numbers of labels per class is given.

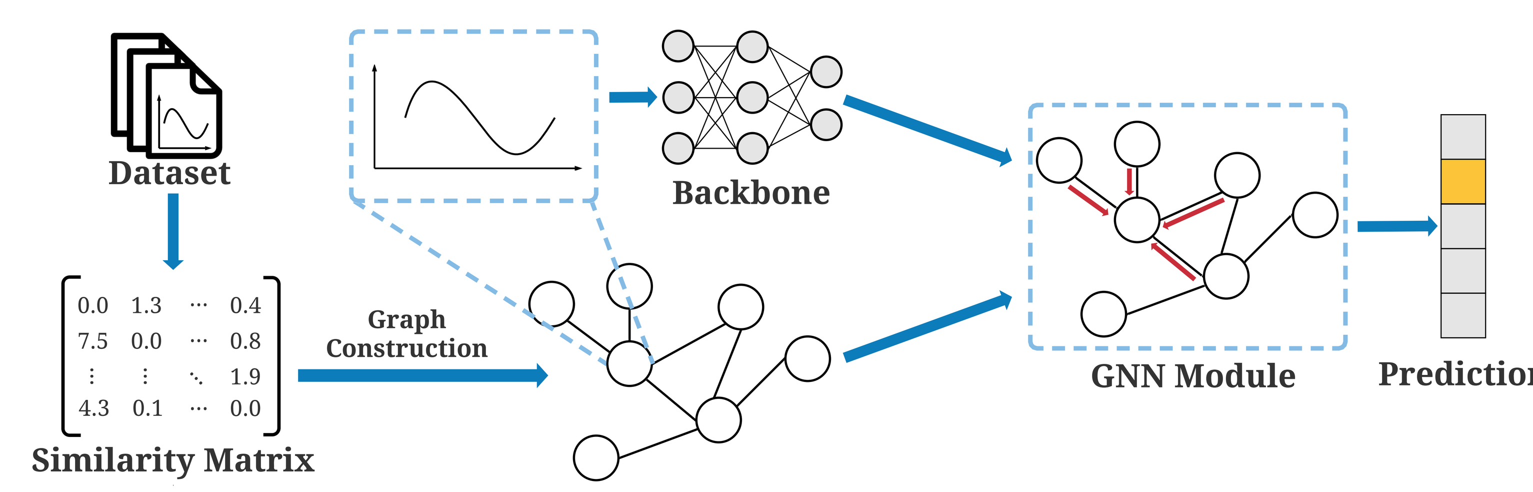
- **Observation 1:** ResNet dominates DTW when we have enough labels.
- **Observation 2:** DTW achieves better performance when we have very few labels by reasoning with pair-wise similarities.
- **Research Question:** Can we connect them in such a way as to jointly model time-series similarities and learn the representations?

Challenges

- How can we incorporate similarity information into representation learning?
- How can we balance similarity information and the original representation learning?

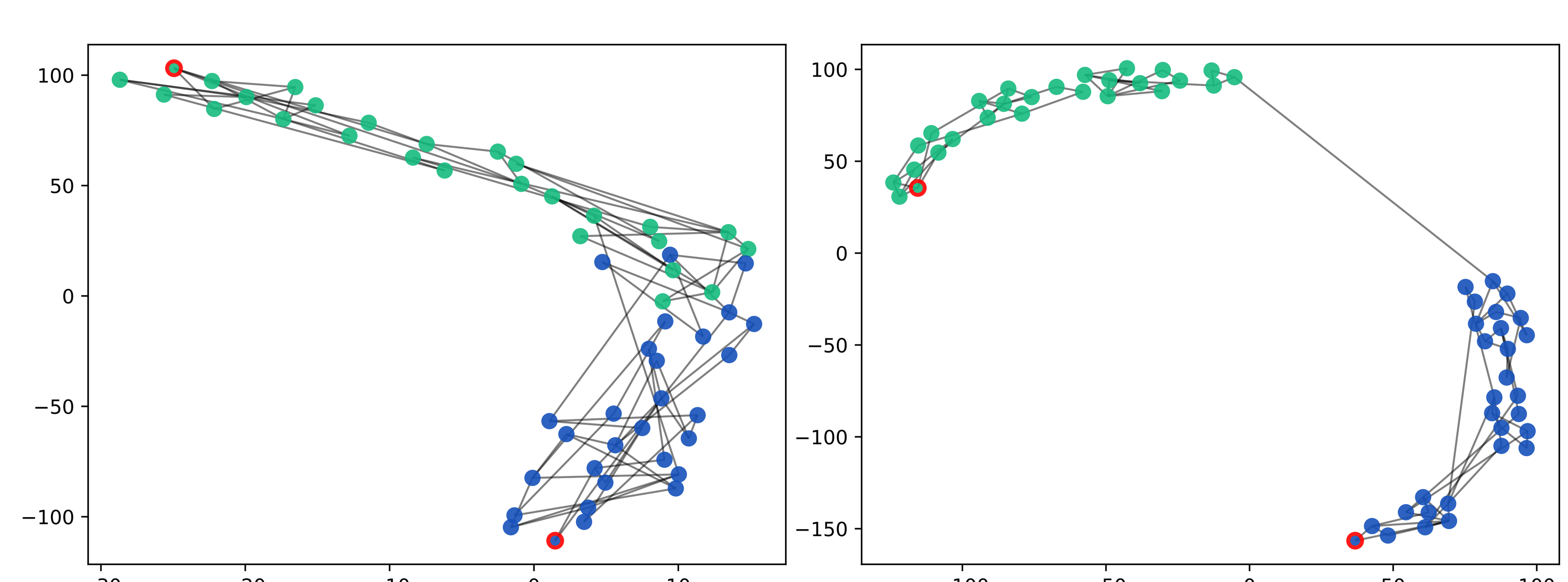
SimTSC framework

To address the challenges, we propose **Sim**ilarity-Aware **T**ime-**S**eries **C**lassification (SimTSC), a conceptually simple and general framework for incorporating similarity information into deep learning models.



- **Step 1: Graph Construction.** We treat each time-series as a node in the graph and treat the pair-wise similarity (e.g., DTW) as the edge weight.
- **Step 2: Backbone.** Use a backbone (e.g., ResNet) to extract time-series features.
- **Step 3: Aggregation with GNN.** Use Graph Neural Networks (GNNs) to aggregate the representations based on the constructed graph.
- **Step 4: Classification.** A classification head will make the final predictions.

Visualization of the Learned Representations



(a) ResNet (83% accuracy) (b) SimTSC (100% accuracy)

Figure: Learned representations of ResNet and SimTSC on Coffee with 56 time-series, two classes marked in blue and green, respectively, and only one time-series labeled in each class (circled in red).

Read More

For more experimental results, please read our paper.

Paper Code