Towards Similarity-Aware Time-Series Classification

Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, Xia Hu
Rice University
Time-Series Classification (TSC) Problem

• Problem setting
  • Given a collection of time-series with the attached labels, TSC aims to train a classifier to classify unseen time-series.

• Main challenge
  • How to model and incorporate the temporal information in the classification?

Source: https://www.cvphysiology.com/uploads/images/CAD012%20ECG%20ST%20depression.png
Existing Solutions

- Existing studies approach TSC in two major directions
  - **Similarity-based**: 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.

Dynamic Time Warping Matching

Source: https://commons.wikimedia.org/wiki/File:Euclidean_vs_DTW.jpg

ResNet with 1-D convolution

Source: https://arxiv.org/pdf/1809.04356.pdf
Motivation

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

Average ranks of ResNet and DTW on the full 128 UCR datasets, where different numbers of labels per class is given.
Research Question and Challenges

• Our research question
  • Can we connect the two research lines 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?
  • Even though we can enable similarity in deep learning models, how can we balance similarity information and the original representation learning?
SimTSC Framework

- **Our simple yet effective solution**
  - We propose Similarity-Aware Time-Series Classification (SimTSC) framework based on Graph Neural Networks.
  - Time-series -> node
  - Similarity of time-series -> edge
  - TSC -> node classification

Each time-series is first processed by a backbone, and enhanced by GNN with aggregation.
Our Instantiation

- **Backbone**
  - We use **ResNet** as the backbone since it has strong performance.

- **Similarity Measure**
  - We use **DTW** as the similarity measure because it is the most popular one.

- **Graph Neural Networks**
  - We use **Graph Convolutional Networks (GCN)** because it is the most basic one.
  - We only use 1-layer GCN. We find that it delivers the best performance.

- **Other Tricks**
  - We use **negative sampling** to sample a half batch of labeled time-series and a half batch of unlabeled time-series.
Results on Univariate Time-Series

• Experimental Setting
  • We compare SimTSC with the existing similarity-based and deep learning methods on the full 128 UCR datasets. We report average rank and Wilcoxon signed rank test (p < 0.05) for the significance test.

| Algorithm       | Labels 1 | 5     | 10    | 15    | 20    | 25    | 30    | 35    | 40    | 45    | 50    |
|-----------------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| DTW             | 3.776    | 4.163 | 4.465 | ▲      | 4.738 | ▲      | 4.824 | † ▲    | 5.048 | † ▲    | 5.160 | † ▲    | 5.309 | † ▲    | 5.199 | † ▲    | 5.211 | † ▲    |
| MLP             | 5.504 † ▲ | 5.496 † ▲ | 5.438 | † ▲    | 5.309 | † ▲    | 5.316 | † ▲    | 5.256 | † ▲    | 5.367 | † ▲    | 5.477 | † ▲    | 5.195 | † ▲    | 5.402 | † ▲    | 5.348 | † ▲    |
| FCN             | 4.630 | ▲   | 4.310 | ▲     | 4.583 | ▲     | 4.723 | † ▲    | 4.803 | † ▲    | 4.699 | † ▲    | 4.910 | † ▲    | 4.773 | † ▲    | 4.883 | † ▲    | 4.852 | ▲     |
| ResNet          | 4.846 † ▲ | 4.857 † ▲ | 4.617 | † ▲    | 4.047 | 4.449 | † ▲    | 4.039 | 4.102 | 4.090 | 4.086 | 3.840 | 3.895 |
| InceptionTime   | 5.484 † ▲ | 5.302 † ▲ | 5.438 | † ▲    | 5.434 | † ▲    | 5.215 | † ▲    | 5.145 | † ▲    | 5.168 | † ▲    | 4.914 | † ▲    | 4.941 | † ▲    | 5.066 | † ▲    | 5.039 | ▲     |
| SimTSC-S        | 4.224 † ▲ | 4.278 | ▲ ▼  | 4.074 | ▲ ▼  | 4.277 | ▲ ▼  | 4.141 | ▲ ▼  | 4.044 | ▲ ▼  | 4.148 | ▲ ▼  | 3.988 | ▲ ▼  | 3.887 | ▲ ▼  | 3.918 | ▲ ▼  | 4.047 |
| SimTSC-I        | 3.724 | 3.817 | 3.793 | ▲ ▼  | 3.836 | ▲ ▼  | 3.746 | 4.031 | ▲ ▼  | 3.762 | ▲ ▼  | 3.734 | ▲ ▼  | 3.852 | ▲ ▼  | 3.867 | ▲ ▼  | 3.797 |
| SimTSC-T        | 3.811 | ▲ ▼  | 3.778 | ▲ ▼  | 3.781 | ▲ ▼  | 3.852 | ▲ ▼  | 3.586 | ▲ ▼  | 3.632 | ▲ ▼  | 3.727 | ▲ ▼  | 3.957 | ▲ ▼  | 3.824 | ▲ ▼  | 3.812 |
Results on Multivariate Time-Series

**Experimental Setting**

- We conduct experiments on 4 multivariate time-series classification tasks, including **Character Trajectories**, **ECG**, **KickVsPunch**, and **NetFlow**.

### Experimental Setting

| Dataset        | Algorithm       | Labels         | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
|----------------|-----------------|----------------|---|----|----|----|----|----|----|----|----|----|
| **Character Trajectories** | DTW             |                | 0.847 ± 0.014 | 0.881 ± 0.005 | 0.895 ± 0.009 | 0.900 ± 0.014 | 0.908 ± 0.013 | 0.907 ± 0.010 | 0.900 ± 0.007 | 0.909 ± 0.010 | 0.913 ± 0.008 |
|                | ResNet          |                | 0.853 ± 0.024 | 0.898 ± 0.017 | 0.920 ± 0.010 | 0.972 ± 0.010 | 0.930 ± 0.008 | 0.941 ± 0.009 | 0.949 ± 0.008 | 0.956 ± 0.007 | 0.958 ± 0.007 |
|                | InceptionTime   |                | 0.883 ± 0.010 | 0.939 ± 0.007 | 0.947 ± 0.006 | 0.968 ± 0.006 | 0.964 ± 0.003 | 0.974 ± 0.005 | 0.979 ± 0.003 | 0.978 ± 0.005 | 0.979 ± 0.001 | 0.986 ± 0.003 |
|                | TapNet          |                | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |
| **ECG**        | DTW             |                | 0.605 ± 0.124 | 0.670 ± 0.086 | 0.740 ± 0.112 | 0.755 ± 0.103 | 0.805 ± 0.043 | 0.825 ± 0.050 | 0.805 ± 0.053 | 0.800 ± 0.057 | -              | -              |
|                | ResNet          |                | 0.745 ± 0.048 | 0.895 ± 0.037 | 0.805 ± 0.058 | 0.800 ± 0.079 | 0.860 ± 0.030 | 0.855 ± 0.048 | 0.850 ± 0.052 | 0.855 ± 0.029 | 0.830 ± 0.037 | 0.870 ± 0.029 |
|                | InceptionTime   |                | 0.750 ± 0.045 | 0.805 ± 0.033 | 0.785 ± 0.020 | 0.800 ± 0.037 | 0.830 ± 0.037 | 0.830 ± 0.043 | 0.825 ± 0.016 | 0.850 ± 0.027 | 0.855 ± 0.015 | 0.850 ± 0.016 |
|                | TapNet          |                | 0.770 ± 0.043 | 0.780 ± 0.012 | 0.755 ± 0.025 | 0.795 ± 0.048 | 0.810 ± 0.037 | 0.705 ± 0.029 | 0.785 ± 0.025 | 0.815 ± 0.037 | 0.830 ± 0.019 | 0.845 ± 0.024 |
|                | InceptionTime   |                | 0.795 ± 0.043 | 0.810 ± 0.020 | 0.855 ± 0.040 | 0.840 ± 0.051 | 0.830 ± 0.064 | 0.840 ± 0.020 | 0.860 ± 0.041 | 0.825 ± 0.047 | 0.830 ± 0.071 | 0.860 ± 0.025 |
|                | TapNet          |                | 0.790 ± 0.062 | 0.765 ± 0.072 | 0.830 ± 0.070 | 0.730 ± 0.159 | 0.740 ± 0.087 | 0.800 ± 0.091 | 0.830 ± 0.048 | 0.750 ± 0.052 | 0.790 ± 0.108 | 0.735 ± 0.108 |
| **NetFlow**    | DTW             |                | 0.810 ± 0.418 | 0.815 ± 0.046 | 0.770 ± 0.108 | 0.815 ± 0.115 | 0.730 ± 0.118 | 0.745 ± 0.075 | 0.745 ± 0.099 | 0.760 ± 0.051 | 0.770 ± 0.071 | 0.710 ± 0.101 |
|                | ResNet          |                | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |
|                | InceptionTime   |                | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |

Similarity-based: DTW, ResNet, InceptionTime, TapNet
Deep Learning: Ours
Impact of the Number of GCN Layers

• **Experimental Setting**
  • We vary the number of GCN layers when we have 10 or 20 labels

![Graph showing average rank vs. number of layers for 10 and 20 labels]

• **Observation**
  • One GCN layer achieves the best performance.
  • Possibly because the graph is dense and more layers lead to over-smoothing.
Learned representations of ResNet and SimTSC on Coffee with 56 time-series. The two classes are marked in blue and green. Only one time-series is labeled.
• Takeaways
  • SimTSC is a conceptually simple yet effective framework to join the research efforts of similarity-based and deep learning methods for time-series classification.
  • We demonstrated the effectiveness of graph neural networks in time-series classification.

• Future Work
  • Larger dataset, sparse graph, other tasks in time-series.
  • Differentiable similarity learning.

• Acknowledgement
  • This work is supported by National Science Foundation (NSF).