MS-SRALAT: Multi-granularity SubStructuRe-Aware Representation Learning Algorithm for Time-series

THAPANA BOONCHOO

DEPARTMENT OF COMPUTER SCIENCE

THAMMASAT UNIVERSITY

JUNE 22, 2023

Agenda

- Backgrounds
- Our proposed method: MS-SRALAT
- Experimental results
- Conclusion
- Q&A Session

Time-series

A time-series is a collection of observations measured in chronological order, and ubiquitous in almost human-related activities in various domains, for example, Finance, Environments (pollution monitoring), Genetics, Multimedia, etc.

https://www.equedia.com/how-to-read-stock-charts-trend-macd-crossovers/

https://knoow.net/ciencinformtelec/informatica/frame/

https://askthescientists.com/genetics/

Time-series Mining

- Time-series databases (Challenges)
  - Large
  - Noisy
  - High dimensional
- Mining Algorithms for Time-series Data
  - Clustering
  - Classification
  - Similarity search

Time-series Representations

- Time series representations aim to generate meaningful representations in a **lower-dimensional** space.
  - Explaining information of the time-series such as trends and shapes.
  - The similar time-series representations should be placed **nearby** in the space.
- **This is the key step in success of almost time-series mining tasks.**

Past methods

- Time-series mining tasks are typically performed on the higher representation or approximation of time-series instead of the original ones so that meaningful results can be obtained.

- Several methods were proposed to produce timeseries representation in **a lower dimension space**.
  - Discrete Fourier transform (DFT)
  - Discrete wavelet transform (DWT)
  - Piecewise aggregate approximation (PAA)
  - Adaptive piecewise constant approximation (APCA)
  - Singular value decomposition (SVD)
  - Etc…

Past methods (Cont’d)

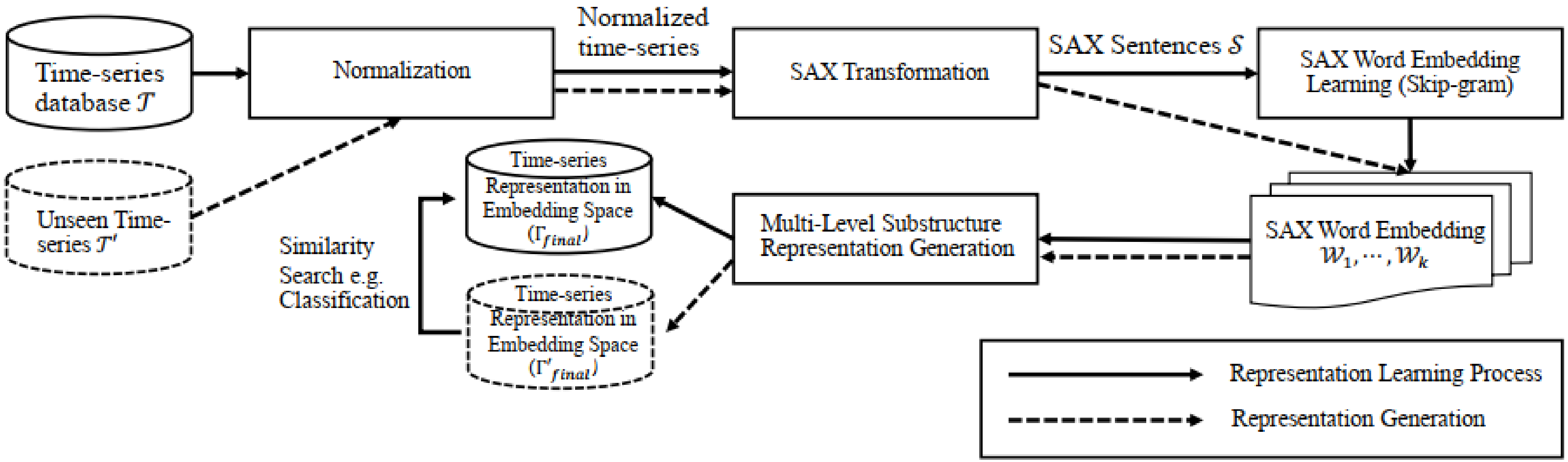
- *Symbolic representation methods* were proposed to reduce the dimensions of time-series data and use discrete symbols as the representation.
  - Symbolic aggregate approximation (SAX)
  - Bitmap representation
  - Bag-of-pattern representation (BoP)
  - Bag-of-SFA-Symbols (BOSS)

Representation Learning Algorithm for Time-series

- Neural network-based models to learn the low-dimensional representations (Embedding).
  - Shallow neural network-based:
    - MAEAT
    - Signal2Vec
  - Deep learning-based:
    - T2Vec
    - NEUTRAJ

Proposed Method:
MS-SRALAT

Overall Algorithm

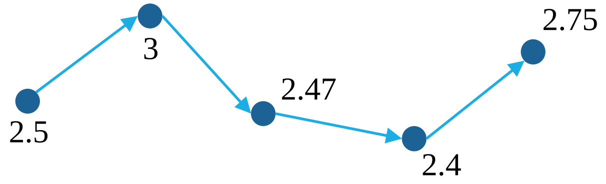


Definitions

Definition 1: (Time Series) A time-series $T_i \in \mathcal{T}$ is a sequence of real values, denoted by T=$\{x_1, x_2, \cdots, x_{|T|}\}$, where $x_j \in \mathbb{R}$, $1 \leq j \leq |T|$, and $|T|$ is the length of time series T.

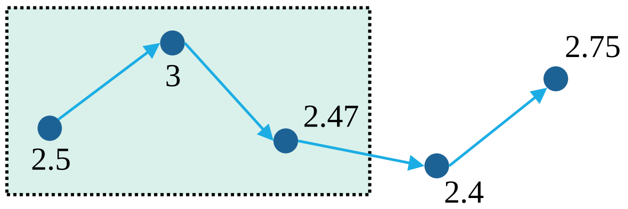
Definition 2: (Time-Series Subsequence) Given a time-series $T_i \in \mathcal{T}$, a time-series subsequence of time-series T_i is a sequence of consecutive values of length m defined as $\tau_{i,j}^{m} = \{x_j, x_{j+1}, \cdots, x_{j+m-1}\}$, where $m \leq |T_i|$, and $1 \leq j \leq |T_i| - m + 1$.

Time-series:



$\mathrm{T}_1 = \{2.5, 3\ , 2.47, 2.4, 2.75\}$

Time-series Subsequence:



$\tau_{1,1}^{3} = \{2.5, 3\ , 2.47\}$

Preprocessing

Definition 4: (Normalized Time Series) Given a time-series $T_i = \{x_1, x_2, \cdots, x_{|T|}\} \in \mathcal{T}$, the normalized time-series is denoted as $T_i' = \{x_1', x_2', \cdots, x_{|T|}'\}$, where $x_i' = \frac{x_i' - \mu_{T_i}}{\sigma_{T_i}}$, and μ_{T_i} and σ_{T_i} are the mean and standard deviation of T_i, respectively.

- **Normalization** using Z-normalization method such that the values of time-series has zero mean and unit-variance
- **SAX Representation**:
  - (1) Reducing the normalized time-series dimension using piecewise aggregate approximation (PAA) with "w" as the parameter (we regard this "w" as **the granularity level**)

$$x_i'' = \frac{w}{|T|} \sum_{j=\frac{|T|}{w}(i-1)+1}^{\frac{|T|}{w}i} x_j'$$

  - (2) Discretization procedure is carried out by determining the **breakpoints** that partition the area under the Gaussian curve N(0, 1) into equal-sized partitions: $B = \langle b_1, \cdots, b_{a-1} \rangle$,

where $b_0 = -\infty$ and $b_a = +\infty$.

Preprocessing (Cont'd)

Definition 3: (SAX Sentence) Given a time-series $T_i \in \mathcal{T}$, a SAX sentence contains a sequence of discrete words corresponding to T_i, denoted by $s^{T_i} = \{W_1^{T_i}, W_2^{T_i}, \cdots, W_k^{T_i}\}$, where $W_j^{T_i}$ is a SAX word at j^{th} position of s^{T_i}, k is the sentence length and $k \leq |T_i|$. We will give the details of how to convert a time-series to its corresponding sentence in the subsequent sections.

- **SAX Representation (cont'd)**:
  - (2) Discretization procedure:
    - The breakpoints B will partition the Gaussian curve into $\boldsymbol{a}$ equal-sized areas of 1/$\boldsymbol{a}$.
  - (3) English alphabet mapping (SAX mapping)
    - Once the breakpoint list B has been constructed and the normalized time-series has been transformed into the PAA representation, each $x_i'' \in T''$ will be mapped to the English alphabet **A** if the value x_i'' less than the break point b_1 and mapped to **B** if $b_1 \leq x_i'' < b_2$ and so forth.
    - We can regard these English alphabets as SAX symbols; These symbols can be used to construct a SAX word (described in IV-A2). The SAX words can then be used to further construct a SAX sentence as defined in Definition 3.

Preprocessing (Cont'd)

> *Definition 5:* (SAX transformation) Given a time-series T, the SAX transformation is a function $f^w_{SAX} : \mathcal{T} \rightarrow \Sigma^w$ mapping a time-series T to a sequence of symbols of length w which is regarded as a SAX word, where Σ^w is the set of all possible SAX words whose length is w.

- **Word Extraction and Sentence Construction**:
  - Given a time-series $T_i \in T$, we can extract its possible subsequences of length m denoted by $SS_{T_i,m} = \langle \tau^m_{i,1}, \tau^m_{i,2}, \cdots, \tau^m_{i,|T_i|-m+1} \rangle$ where each $\tau^m_{i,j}$ will correspond to a SAX word by the SAX transformation.
  - We can then convert the timeseries T_i to its corresponding SAX sentence (see Definition 3) $S_{T_i} = \langle W_1^{T_i}, W_2^{T_i}, \cdots, W_k^{T_i} \rangle$ by first extracting all subsequences from T_i and then transforming all the subsequences $SS_{Ti,m}$ into the sequence of SAX words (SAX sentence)

SAX Word Embedding Learning

- We learn substructure-aware latent representations of each SAX word using the Skip-gram model.
- The training set for learning SAX word embedding can be obtained by converting the raw timeseries database $T = \{T_1, T_2, \cdots, T_N\}$ into the corresponding database of SAX sentences denoted as $S = \{s^{T_1}, s^{T_2}, \cdots, s^{T_N}\}$.
- Given that a SAX sentence s^{T_i} comprises the sequence of SAX words $\left\langle W_1^{T_i}, W_2^{T_i}, \cdots, W_k^{T_i} \right\rangle$, we define the context of word $W_1^{T_i}$ by the set of its surrounding words denoted by $\mathbf{C}_\phi(W_j^{T_i}) = (\bigcup_{l=j-\phi}^{j+\phi} W_l^{T_i}) \setminus W_j^{T_i}$, ϕ is the window size.

SAX Word Embedding Learning

- The objective to train the Skip-gram model is to *maximize* the **average log likelihood** of context words $C_\phi(W_j^{T_i})$ given the word $W_j^{T_i}$.
- Considering the SAX sentence database S, the objective function $J(W)$ can be computed by an average negative log likelihood which is equivalent to maximizing the average log likelihood as follows.

$$J(\mathcal{W}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{W_c \in \mathbf{C}_\phi(W_j^{T_i})} \log P(W_c|W_j^{T_i};\mathcal{W})$$

$$\boxed{P(W_c|W_j^{T_i};\mathcal{W}) = \frac{exp(\mathbf{v}_{W_c} \bullet \mathbf{v}_{W_j^{T_i}})}{\sum_{W_x \in \mathcal{V}} exp(\mathbf{v}_{W_x} \bullet \mathbf{v}_{W_j^{T_i}})}}$$

Negative sampling

$$\boxed{P(W_c|W_j^{T_i};\mathcal{W}) = \log \sigma(W_c \bullet W_j^{T_i}) + \sum_{m=1}^{k} \log \sigma(-W_m \bullet W_j^{T_i})}$$

Substructure-Aware Time-series Representation

- Once we have obtained the SAX word embeddings learned from the preceding process, we can simply construct a representation for a raw time-series $T_i \in T$ as the following two steps.
  - First, we convert the time-series Ti to a SAX sentence (sequence of SAX word)
  - Second, for each SAX word $\mathrm{W}_j^{T_i} \in V$ in the sentence s^{T_i}, we look up its corresponding SAX word embedding $v_{\mathrm{W}_j^{T_i}}$ in the learned W, then combining these SAX word embeddings by finding the average over all embedding vectors as follows.

$$\Gamma^{T_i} = 1/|T_i| \sum_{l=1}^{k} \mathbf{v}_{W_l^{T_i}}$$

Substructure-Aware Time-series Representation (Cont'd)

- In this paper, we proposed to exploit the substructure of time-series and encode the discretization process in different levels of substructures.
  - Specifically, we used multiple SAX transformation functions

$$\mathcal{F} = \{f_{SAX}^{w_1}, \cdots, f_{SAX}^{w_L}\}$$

  - we can learn the set of representation parameters $\mathcal{P} = \{\mathcal{W}^{w_1}, \cdots, \mathcal{W}^{w_L}\}$
  - Given a time-series T_i, we can obtain the set of embedding vectors $\{\Gamma_{w_1}^{T_i}, \cdots, \Gamma_{w_L}^{T_i}\}$, and the final representation for the time-series T_i is obtained by the concatenation of these vectors

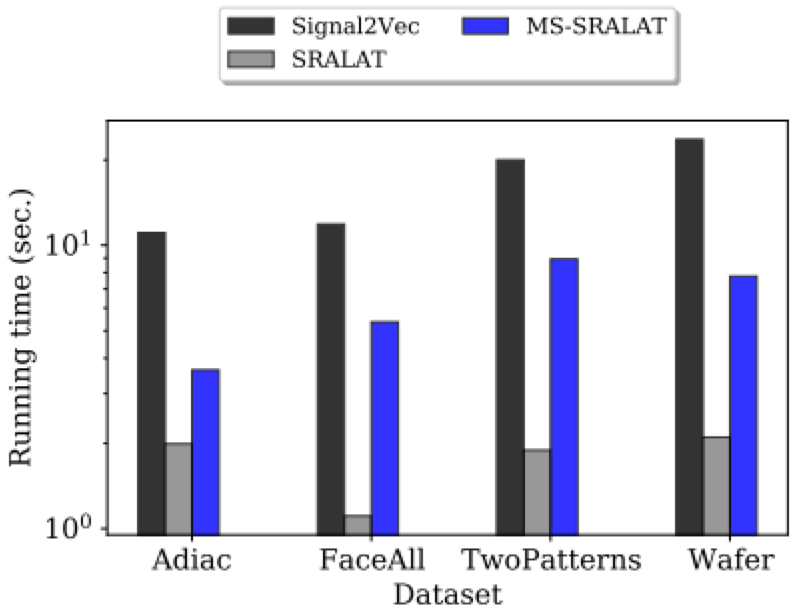
$$\Gamma_{final}^{T_i} = [\Gamma_{w_1}^{T_i}; \cdots; \Gamma_{w_L}^{T_i}]$$

Experimental results

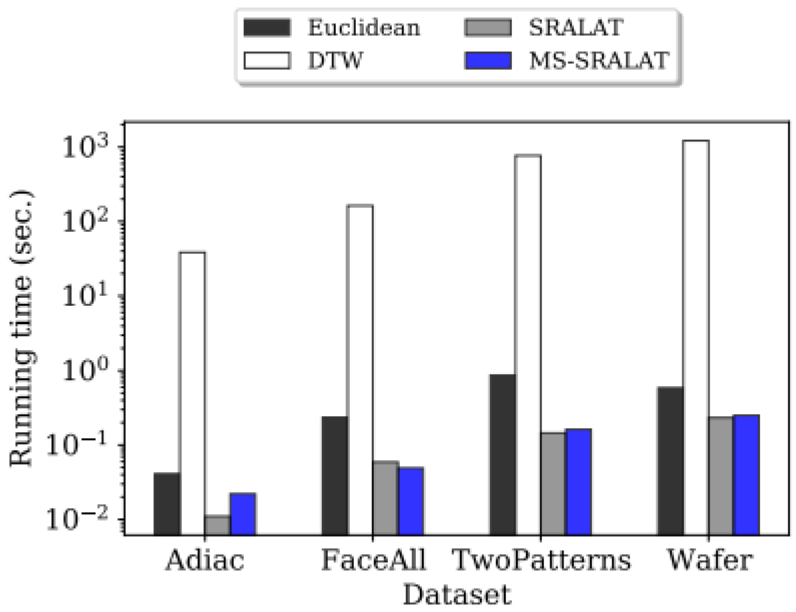
CLASSIFIER ERROR RATES OF DIFFERENT COMPARED METHODS ON VARIOUS DATASETS

Dataset	#Train	#Test	#Classes	1-NN ED	1-NN DTW	Signal2Vec	Single-level SRALAT	Multi-granularity SRALAT
Adiac	390	391	37	**0.389**	0.391	0.698	0.752	0.621
Beef	30	30	5	0.467	0.467	0.633	0.467	**0.333**
CBF	30	900	3	0.148	0.003	0.453	0.003	**0.000**
Coffee	28	28	2	0.25	0.18	0.429	0.179	**0.107**
ECG200	100	100	2	**0.12**	0.23	0.350	0.140	0.200
FaceAll	560	1690	14	0.286	**0.192**	0.825	0.475	0.246
FaceFour	24	88	4	0.216	0.17	0.534	0.182	**0.023**
Fish	175	175	7	0.217	**0.167**	0.549	0.320	0.229
Gun-Point	50	150	2	0.087	0.093	0.167	0.200	**0.060**
Lightning2	60	61	2	0.246	**0.131**	0.344	0.213	**0.131**
Lightning7	70	73	7	0.425	**0.274**	0.671	0.384	0.356
OliveOil	30	30	4	**0.133**	**0.133**	0.200	0.333	0.167
OSULeaf	200	242	6	0.483	**0.409**	0.690	0.607	0.492
SynControl	300	300	6	0.12	**0.007**	0.727	0.077	0.030
SwedLeaf	500	625	15	0.213	**0.21**	0.566	0.355	0.272
Trace	100	100	4	0.24	**0**	0.220	0.140	0.040
TwoPatterns	1000	4000	4	0.09	**0**	0.742	0.186	0.030
Wafer	1000	6164	2	**0.005**	0.02	0.084	0.009	0.010

The running time comparison



(a) Discretization + Training



(b) Similarity Search

Future Work

- Study more on the parameters (e.g., granularity level, number of substructures, …) used in training models since different sets of parameters make the model to produce different representations in terms of semantics.

- Produce substructures of time-series in an adaptive manner as opposed to a static manner so that the representation might capture more clearly the characteristic of each individual time-series data.

Q&A

Thank you.