Non-Iterative Methods for Classification, Forecasting and Visual Tracking

Part 5: Time Series Classification

Dr P. N. Suganthan epnsugan@ntu.edu.sg
School of EEE, NTU, Singapore

Some Software Resources Available from:
https://github.com/P-N-Suganthan

DeepLearn 2019
Warsaw, Poland
22 July – 26 July 2019

Time Series Classification (TSC) [1]

- Unlike traditional classification problems, TSC problems contains data with ordered attributes (Example: ordered by time)
  - May contain discriminative features depending on the order

- Many TSC algorithms had been proposed
  - Prior to 2003, at least 100 papers proposing TSC algorithms have published

- Classifiers are categorized into 6 groups
  - Whole series, Intervals, Shapelets, Dictionary, Combination and Model Based

Main Report:

[1] A. Bagnall, J. Lines, A. Bostrom, J. Large and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp 606-660, May 2017.

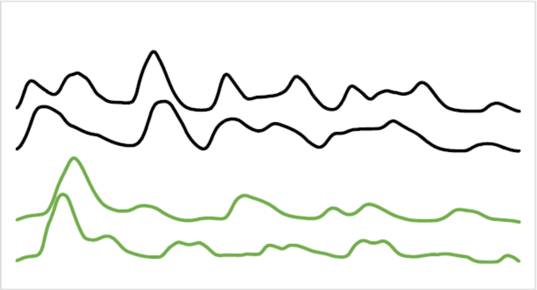
Whole Series

- In some time series, discriminative information are found in the entire series

- Series are compared with the following approaches :
  - Expressing series as a vector (in traditional classification)
  - Applying distance measure on the whole series (similarity)

- For Time Series Classification, distance measures are commonly used

Whole Series Similarity



- Distance measures are usually validated using 1 Nearest Neighbour Classifier (1-NN)

- However, data may contain some mis-alignments between series which reduces classification accuracy

- Hence, many research efforts aim to compensate for small mis-alignments (elastic distance measures)

- Dynamic Time Warping used as the standard benchmark elastic distance measure

Dynamic Time Warping (DTW)

- To measure the distance between 2 series of length m
  - **a** = (a_1, a_2, … , a_m) and **b** = (b_1, b_2, … , b_m)

- Create m×m pointwise distance matrix M(a, b)
  - $M_{i,j} = (a_i - b_j)^2$

- Find warping path P = ((e_1,f_1), (e_2,f_2), … , (e_s, f_s)) that has the minimum total distance
  - $D_P(a,b) = \sum_{i=1}^{s} M_{e_i f_i}$

Dynamic Time Warping (DTW)

- DWT can be a time consuming operation
  - Common solution: Add a (specified) upper limit r to the distance between any pair of indexes in a path (the Warping Window)
    - $|e_i - f_i| \le r$

- Many alternative methods had been proposed
  - Edit distance with Real Penalty (ERP) [2]
  - Longest Common Subsequence (LCSS) [3]

[2] L. Chen and R. Ng. "On the marriage of lp-norms and edit distance", in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, pp. 792-803.
[3] D. S. Hirschberg, "Algorithms for the longest common subsequence problem," *Journal of the ACM (JACM)*, vol. 24, no. 4, pp. 664-67, 1977.

Weighted Dynamic Time Warping (WDTW) [4]

- Smoothed alternative method to DTW with warping window

- Add multiplicative weight penalty based on distance between points in a path
  - Discourages large warping (Similar to DTW with warping window)

- Uses a logistic weight function
  - $\omega(a) = \frac{\omega_{max}}{1+e^{-g(a-m/2)}}$

[4] Y. S Jeong, M. K. Jeong and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, vol. 44, no. 9, pp. 2231-2240, 2011.

Time Warp Edit (TWE) [5]

- Calculates distance based on amount of effort required to match the series (where **a** = (a_1, a_2, … , a_m) and **b** = (b_1, b_2, … , b_n))

- At each iteration (where i and j refers to index of series **a** and **b**), perform either 1 of 3 operations (Algorithm chooses the operation with the least effort required)
  - Deletes an element in series **a** (increment i)
  - Deletes an element in series **b** (increment j)
  - Match the element in series **a** and **b** ($a_i = b_j$, increment i and j)

- Amount of warping is controlled by stiffness parameter (similar to WDTW)

[5] P. F. Marteau, "Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 306-318, Feb. 2009.

Move-Split-Merge (MSM) [6]

- Similar to other edit distance-based approaches
  - Similarity calculated using a set of operations

- Consists of 3 operations
  - Move: Replace a value with another (substitute operation)
  - Split: Inserts a copy of the value immediately after itself
  - Merge: Deletes the value if it directly follows an identical value

[6] A. Stefan, V. Athitsos and G. Das, "The Move-Split-Merge Metric for Time Series," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1425-1438, June 2013.

Complexity Invariant Distance (CID) [7]

- When comparing time series, complex series tend to be more similar to simple series (which could cause classification errors)

- Hence, this method aims to compensate for differences in complexity between 2 series

- A simple complexity measure: sum of squares of first differences
  - $c = \sum_{i=1}^{m-1}(a_i - a_{i+1})^2$

- Can be used with Euclidean Distance / DTW

[7] G. E. Batista, E. J. Keogh, O. M. Tataw and V. M. De Souza, "CID: an efficient complexity-invariant distance for time series," *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 634-669, 2014.

Derivative Dynamic Time Warping (DD_{DTW}) [8]

- Applies distance measure on first-order differenced time series data in addition to the original data
  - Difference function: $a'_i = a_i - a_{i+1},\ \ i = 1, \ldots, m-1$
  - Creates 2 sets of distances

- Resultant distances are combined using a weighting parameter (tuneable using cross validation on training data)

[8] T. Górecki and M. Łuczak, "Using derivatives in time series classification," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 310-331, 2013.

Derivative Transform Distance (DTD_C) [9]

- Extension of DD_{DTW} algorithm
- Combines DD_{DTW} and distances on data with sine, cosine and Hilbert transform
  - Creates 3 sets of distances, 3 weighting parameters
- Cosine version is used in this paper
  - $c_i = \sum_{j=1}^{m} a_j \cos\left(\frac{\Pi}{2}\left(j - \frac{1}{2}\right)(i-1)\right), \quad i = 1, \dots, m$

[9] T. Górecki and M. Łuczak, "Non-isometric transforms in time series classification using DTW," *Knowledge-Based Systems*, vol. 61, pp. 98-108, 2014.

Elastic Ensemble (EE) [10]

- Collection of 11 1-Nearest Neighbour classifiers
  - Each classifier uses output from different distance measures
- Final output from Elastic Ensemble is based on weighted voting
  - Weights assigned set using cross-validation on training data

[10] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 565-592, 2015.

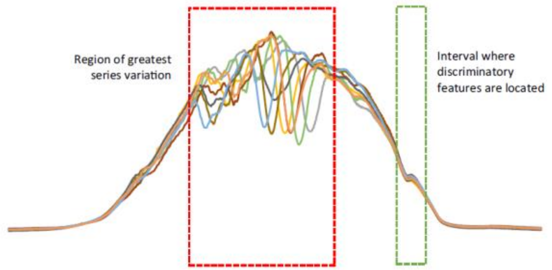
Elastic Ensemble (EE) [10]

Slide numbers refer to slides in Part 5.

- Collection of 11 distance measures
  - Euclidean Distance
  - DTW with full warping window (see slide 5)
  - DTW with warping window size set using cross validation
  - Derivative DTW with full warping window (see slide 11)
  - Derivative DTW with warping window size set using cross validation
  - Weighted DTW (see slide 7)
  - Longest Common Subsequence
  - Edit Distance with Real Penalty
  - Time Warp Edit Distance (see slide 8)
  - Move-Split-Merge distance metric (see slide 9)

[10] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 565-592, 2015.

Intervals



- In some cases, discriminative information lies only on a section of the time series

- Classifier selects one/some phase-dependent intervals of the series that contains discriminative information

- Feature extraction can be performed on each interval before performing classification
  - Examples: interval mean and standard deviation

Time Series Forests (TSF) [11]

- Employs a random forest approach that uses summary statistics of each interval as features

- To construct a tree:
  - Pick $\sqrt{m}$ intervals randomly
  - For each interval, calculate the mean, standard deviation and slope
  - Train decision tree using the resulting $3\sqrt{m}$ features

- Final output is based on majority voting of all the trees in the classifier

[11] H. Deng, G. Runger, E. Tuv and M. Vladimir, "A time series forest for classification and feature extraction," *Information Sciences*, vol. 239, pp. 142-153, 2013.

Time Series Bag of Features (TSBF) [12]

- An extension of TSF with 4 stages
  - Generate a subseries classification problem
  - Calculate class probability estimates for each subseries
  - Construct a bag of features for each original instance from these probabilities
  - Train classifier using the bag of features representation

- Random forest is used for classification

[12] M. G. Baydogan, G. Runger and E. Tuv, "A Bag-of-Features Framework to Classify Time Series," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2796-2802, Nov. 2013.

Learned Pattern Similarity (LPS) [13]

- Developed by the same research group as TSF and TSBF
- Differences:
  - Subseries become attributes instead of cases
  - LPS creates internal regression model instead of classification model in TSBF
- Consists of 2 stages:
  - Constructs an ensemble of regression trees
  - Form a count distribution over each tree's leaf node
- Classification of new cases is based on 1-NN on these count distribution

[13] M. G. Baydogan and G. Runger, "Time series representation and similarity based on local autopatterns," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 476-509, 2016.

Shapelets

- In some classification problems, classes can be identified based on absence/presence of patterns anywhere in the series.
- Classifier finds short patterns which describe the class (called shapelets) but can occur anywhere on the series
- Series are classified based on presence/absence of the shapelets in any part of the series
- Original algorithm finds the best shapelet (a subseries extracted from a series) and used as splitting criteria for decision tree [14]

[14] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," *Data mining and knowledge discovery*, vol. 22, no. 1-2, pp. 149-182, 2011.

Fast Shapelets (FS) [15]

- Extension of decision tree shapelet approach that speeds up shapelet discovery

- Approximates shapelets using Symbolic Aggregate Approximation (SAX)
  - Method for converting series to strings

- Creates a dictionary of SAX words for each shapelet length and evaluates them

- K Best SAX words are selected and convert back to shapelets

[15] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *proceedings of the 2013 SIAM International Conference on Data* Mining, 2013, pp. 668-676.

Shapelet Transform (ST) [16]

- Aims to find best K shapelets (instead of best shapelet at each node)

- Shapelets are used to transform the dataset
  - On each instance, each attribute is the (minimum) distance between the series and 1 shapelet

- Evaluation of shapelets is done in 2 stages
  - On each time series, find the minimum Euclidean distance between the shapelet and all possible subseries
  - Using the resultant distance vector of n observations, calculate the information gain

[16] J. Hills, J. Lines, E. Baranauskas, J. Mapp and A. Bagnall, “Classification of time series by shapelet transformation,” *Data Mining and Knowledge Discovery*, vol. 28, no. 4, pp. 851-881, 2014.

Improved Shapelet Transform (ST) [17]

- To improve performance on multi-class problems, the algorithm balances the number of shapelets for each class

- Evaluation of each shapelet is based on its capability to discriminate just 1 class

[17] A. Bostrom and A. Bagnall, "Binary shapelet transform for multiclass time series classification," In *International Conference on Big Data Analytics and Knowledge Discovery*, 2015, pp. 257-269.

Shapelet Transform (ST)

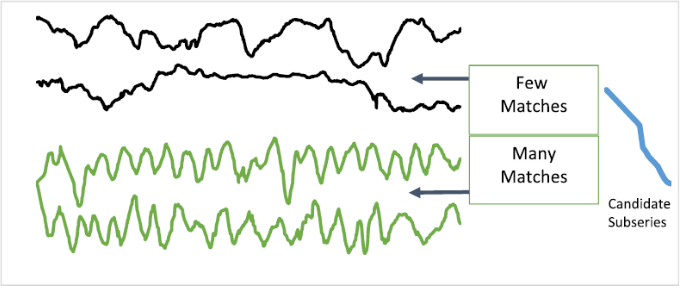
- Uses an ensemble of 8 classifiers on transformed dataset
  - K-Nearest Neighbour (K tuned using cross validation)
  - Naïve Bayes
  - C4.5 decision tree
  - Support Vector Machine with Linear Kernel
  - Support Vector Machine with Quadratic Kernel
  - Random Forest (500 trees)
  - Rotation Forest (50 trees)
  - Bayesian Network

Learned Shapelets (LS) [18]

- Adopts gradient descent shapelet search procedure
- K shapelets initialised using k-means clustering of candidates from training data
- Optimises logistic loss based on a logistic regression model for each class
- Learns the weights and shapelets to produce final logistic regression model

[18] J. Grabocka, N. Schilling, M. Wistuba and L. Schmidt-Thieme, "Learning time-series shapelets," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 392-401.

Dictionary Based



- In some problems, the relative frequency (instead of presence/absence) of patterns is important to distinguish the classes

- Methods creates frequency counts of recurring patterns and builds classifiers with the resulting information

- Dimensionality of series reduced by transforming series into representing words

- The distribution of words will be used to compare time series

Bag of Patterns (BOP) [19]

- Dictionary classifier built on SAX

- Transforms the data with following steps:
  - Extracts subseries of the same length (windowing)
  - Perform SAX: Quantize the time subsequence with SAX (create words)
  - Perform numerosity reduction
    - If consecutive windows produce the same word, only 1st run is recorded
  - Create a histogram of SAX words (which then fed to classifier)

- To classify new samples, same transformation is applied on new data

[19] J. Lin, R. Khade and Y. Li, "Rotation-invariant similarity in time series using bag-of-patterns representation," *Journal of Intelligent Information Systems*, vol. 39, no. 2, pp. 287-315, 2012.

Symbolic Aggregate Approximation-Vector Space Model (SAXVSM) [20]

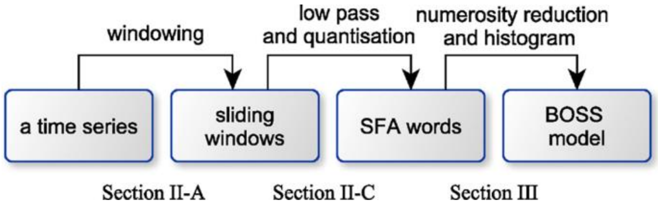
- Combination of SAX (used in BOP) and vector space model

- SAXVSM forms term frequencies over classes and weights by inverse document frequency (*tf* · *idf*)
  - *tf*: Number of times the word appears in a class
  - *idf*: Number of classes the word appears in

- Classification are made using 1-NN based on word frequency distribution and inverse document frequency of each class

[20] P. Senin and S. Malinchik, "SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model," *2013 IEEE 13th International Conference on Data Mining*, Dallas, TX, 2013, pp. 1175-1180.

Bag of SFA Symbols (BOSS) [21]



- BOSS creates the histogram of Symbolic Fourier Approximation (SFA) words (which is then fed to classifiers)
- Transforms the data with following steps (Similar to BOP but uses SFA):
  - Extracts subseries of the same length (windowing)
  - Perform SFA: Approximate each subsequence using discrete Fourier Transform and quantize the results (create SFA words)
  - Perform numerosity reduction
    - If consecutive windows produce the same word, only 1st run is recorded
  - Create a histogram of SFA words

[21] P. Schäfer, "The BOSS is concerned with time series classification in the presence of noise," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1505-1530, 2015.

Combinations

- Ensembles are popular in recent TSC research and produces good results in general classification problems

- Classifiers under this category utilise more than one approach to improve classification accuracy
  - Example: DWT Features (Whole series + Dictionary based)

Dynamic Time Warping Features (DTW_F) [22]

- Combination of whole series and dictionary based approaches
- Transformed dataset is constructed based on 3 methods
  - Full window DWT
  - Optimal window DWT
  - Bag of Patterns
- Transformed dataset trained using Support Vector Machine with polynomial kernel

[22] R. J. Kate, "Using dynamic time warping distances as features for improved time series classification," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 283-312, 2016.

Collection Of Transformation Ensembles (COTE) [23]

- Collection of 35 classifiers from 4 major components
  - Elastic Ensemble (see slide 13)
  - Shapelet Transform (see slide 21)
  - Autocorrelation Function
    - Transform data using autocorrelation function before classifying
  - Power Spectrum
    - Transform data into power spectrum data before classifying

Classifiers used in Ensemble shown in slide 23 under Shapelet Transform

- Weights of the classifiers are assigned using 10-fold stratified cross validation on training data

[23] A. Bagnall, J. Lines, J. Hills and A. Bostrom, "Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522-2535, Sept. 1 2015.

Model Based

- Model based algorithms fit generative models to each series
- Measures similarity based on similarity of models
- Examples:
  - Fitting auto-regressive models
  - Hidden Markov models
  - Kernel models
- Commonly used for other tasks

Experiment Setup

- Classification performance of 18 Classifiers were assessed on 85 University of California, Riverside (UCR) datasets
- For each dataset, each classifier is evaluated on 100 stratified resampling folds
  - Size of training/testing sets and class distributions are kept constant
  - Average accuracy over 100 folds are recorded
  - For tuning, cross-validation on training set is used
- Experiments on all 18 classifiers are run using Java Language

Experiment Results

- Performance are compared with benchmark classifiers:
  - Dynamic Time Warping (**DTW**) and Rotation Forest (**RotF**)

- From the results of 18 classifiers,
  - **COTE** performed significantly better than other classifiers
  - 9 Classifiers (**COTE**, **MSM**, **LPS**, **TSBF**, **TSF**, **ST**, **BOSS**, **EE**, and **DTW_F**) performed significantly better than benchmark classifiers

- COTE is an ensemble of ensemble with about 35-40 classifiers inside.
- Codes and Detailed Results available:
  - Detailed Results: http://timeseriesclassification.com
  - Java Codes Repository: https://bitbucket.org/TonyBagnall/time-series-classification

Time Series Classification with Deep Neural Networks

- Use deep learning techniques to extract meaningful features instead of conventional feature extraction techniques.
- Multi-layer perceptron (MLP), Fully Convolutional Network (FCN) and ResNet based methods proposed in [24].
- LSTM Fully Convolutional Network in [25]. LSTM complements the FCN. Features from both pipelines are concatenated before final classification.
- Experiments on UCR datasets (non-vision tasks).
- Deep neural networks generally overfit since the UCR time series data is small [24]. To achieve a good performance, deep learning methods require large datasets. Different regularization techniques need to be used to achieve best results.
- Deep learning based methods show no significant improvement over non-deep learning techniques such as COTE, BOSS. Authors of [24] suggest to use COTE, BOSS if white box models are preferred.
- Github link for the codes of [24]: https://github.com/cauchyturing/UCR_Time_Series_Classification_Deep_Learning_Baseline

[24] Wang, Zhiguang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. *International Joint Conference on Neural Networks (IJCNN) IEEE*, 2017.
[25] Karim, Fazle, et al. LSTM fully convolutional networks for time series classification. *IEEE Access* 6 (2018): 1662-1669.