Time Series Classification System for AALTD' 16 Challenge

Chin-Chia Michael Yeh and Eamonn Keogh
University of California, Riverside
myeh003@ucr.edu and eamonn@cs.ucr.edu

1 Overview

The AALTD'16 Challenge [2] is a multi-class classification problem with 6 classes. The dataset consists of 180 training samples and 180 testing samples. Each sample is a multi-variable time series with 24 channels and length of 51.

The submitted time series classification system is a two feature late fusion classifier. Specifically, we extract features using two different feature learning/transformation methods: *Sparse Coding Based Bag of Frame Model (BoFSC)* and *Random Shapelet Transformation (RandShape)*. Then, two linear Support Vector Machines (SVM) are trained independently for each feature, and the final estimated probability over all class are computed by aggregating probability outputted by each linear SVM (weighted by cross-validation accuracy). Figure 1 shows the overall diagram of the classification system.

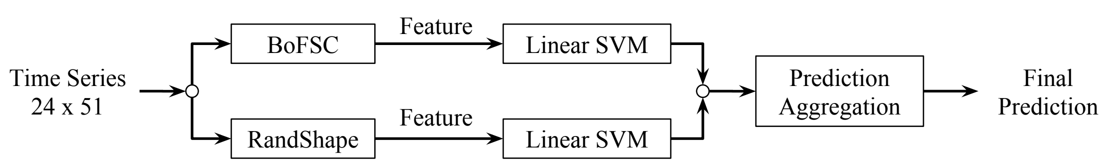

Figure 1: Overall system diagram. The input to the feature extraction pipeline is a 24×51 sized matrix, where 24 is the number of channels and 51 is the length. The output is a 6 sized vector where each element is the estimated probability of a class.

In the official leaderboard, the BoFSC and RandShape system on its own achieves an accuracy of 0.944 (ranked 5) and 0.883 (ranked 17) respectively. The combined system achieves an accuracy of 0.961 (ranked 1). The source code can be found at [1] The details of each feature extraction method are described in Section 2 and Section 3. The SVM implementation we used is LIBSVM [3]. The SVM library is capable of generating probability estimation and handle multi-class classification problems.

2 Sparse Coding Based Bag of Frame Model

BoFSC is effective in various music recognition tasks [7]; therefore, it is included as one of the two features in the classification system. As shown in [9], different bag of frame models can be created for different tasks by arranging various encoding, pooling, and normalization methods in specific orders. The particular pipeline used in this work consists the following building blocks: *Power Normalization*, *Consecutive Frame Concatenation*, *Unit Norm Normalization*, *Sparse Coding*, and *Mean Pooling*. The arrangement of these building blocks are shown in Figure 2.

2.1 Power Normalization

Power normalization is a technique which capable of suppressing various anomalies (e.g., non-additivity, non-normality, and heteroscedasticity) from data [7]. Given a input matrix x, the power normalization can be calculated as $sign(x)|x|^p$ where $p \in [0, 1]$ is a hyperparameter. It is set to $\frac{1}{3}$ in BoFSC system.

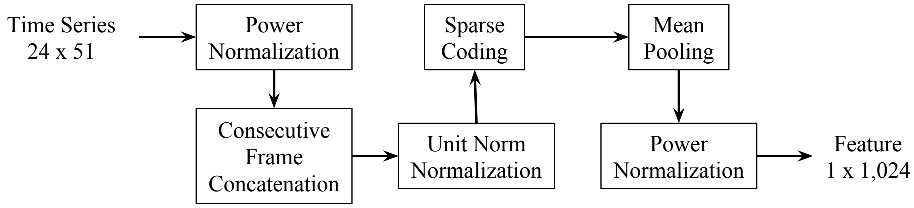

Figure 2: Sparse coding based bag of frame model adapted in this system. The input to the feature extraction pipeline is a 24×51 sized matrix, where 24 is the number of channels and 51 is the length. The output feature is a $1,024$ sized vector.

2.2 Consecutive Frame Concatenation

Consecutive frame concatenation is performed simply by concatenating successive n frames[1] within a sliding window into a single vector. The result vectors incorporate better temporal information comparing to the frames in the sample prior to this operation. The window and hop size of the sliding window are set to 16 and 1 respectively.

Given a 24×51 matrix, the operation outputs a 384×36 matrix using the $16/1$ sliding window.

2.3 Unit Norm Normalization

This operation is a common preprocess for sparse coding [10]. It is applied by normalizing each frame to unit vector (i.e., a vector with its 2-norm equals to 1).

2.4 Sparse Coding

Sparse coding based system has been found more effective comparing to competing methods such as vector quantization methods for bag of frame model [7]. Given an input frame vector $\mathbf{z} \in \mathbb{R}^m$ and a pre-built dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$, it computes the sparse coding representation $\alpha^* \in \mathbb{R}^k$ (for each frame) via

$$\alpha^* = \arg\min_{\alpha} \; \|\mathbf{z} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1 \,. \tag{1}$$

As a result, α^* is a k-dimensional vector with a few non-zero terms that as a whole well approximates the input vector $\mathbf{z}$. The parameter λ is a trade-off between sparsity term and reconstruction error. We set $\lambda = m^{-1/2}$ following [6] and used the LARS-lasso algorithm [4] to solve Eq. 1. The dictionary $\mathbf{D}$ plays an important role in computing sparse coding representation, and can be obtain via solving the following optimization problem.

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} \; \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right) , \tag{2}$$

where n is the number of training data. This optimization problem can be efficiently solved by the online dictionary learning algorithm [6] adopted in this work.

We use SPAMS toolbox [6] for their LARS-lasso and online dictionary learning implementation. The size of the dictionary is set to $1,024$.

2.5 Mean Pooling

This operation is used to summarize the information contains in a multi-variable time series across time. It is applied by compute the mean of each channel across time.

Given a input multi-variable time series with $1,024$ channels and 36 in length, mean pooling yields a $1,024$ sized vector.

[1]frame is refer to one of the fifty-one 24×1 sized vectors in a sample.

3 Random Shapelet

Shapelet based time series classification system has been shown effective for solving the time series classification problems from various domains [5, 8]. Among different types of shaplet based system, we choose to built our system around shapelet transformation because it has allow us to combine the power of shapelet with any desired classifier [5]. We extract the shapelet based feature with the pipeline shown in Figure 3.

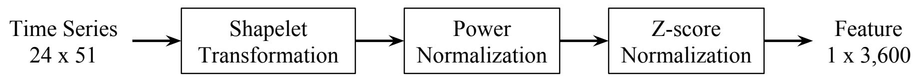

Figure 3: Random shapelet extraction system. The input to the feature extraction pipeline is a 24×51 sized matrix, where 24 is the number of channels and 51 is the length. The output feature is a $3,600$ sized vector.

3.1 Shapelet Transformation

Given a set of k shapelets S and a input signal x, shaplet transformation compute the distance between x with each of the shapelet in S, and use the k result distance measurement as the features [5]. The distance d_i between a input time series x with length n and a shaplet $s_i \in S$ with length m is computed by

$$d_i = \min_{i=0}^{n-m} ED(x[i : i+m], s_i) \tag{3}$$

where ED is a function that compute the z-normalized Euclidean distance between two input vectors. As there are total of k shaplets, the shapelet transformation of x is a vector $D = [d_1, d_2, d_3, ..., d_k]$ where d_i is the distance between x and s_i.

To construct S, we randomly selected 150 time series segment (with length of 25) from random channel in a randomly selected training sample. In other words, the S in our system is a set of 150 randomly selected single channel time series segment with length of 25. To extract the shapelet feature, we then apply shaplet transformation to each of the 24 channel independently with S, which yields 24 shaplet features (each is a sized 150 vector) for each time series sample. Lastly, we concatenate the 24 shaplet feature vectors to form the final feature vectors. The size of the output feature vector is $3,600$.

3.2 Power Normalization

See Section 2.1. The p is set to $\frac{1}{2}$ in RandShape system .

3.3 Z-score Normalization

Assuming each feature in the feature vector is normally distributed, z-score normalize convert each feature to z-score so each element in the feature vector is commeasure. The mean and standard deviation are estimated from training samples.

References

[1] AALTD'16 Submission. `https://github.com/mcyeh/aaltd16_fusion/`.

[2] Challenge — AALTD'16. `https://aaltd16.irisa.fr/challenge/`.

[3] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[4] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, pages 407–499, 2004.

[5] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall. Classification of time series by shapelet transformation. *Data Min. Knowl. Discov.*, 28(4):851–881, July 2014.

[6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM.

[7] L. Su, C.-C. M. Yeh, J.-Y. Liu, J.-C. Wang, and Y.-H. Yang. A systematic evaluation of the bag-of-frames representation for music information retrieval. *IEEE Transactions on Multimedia*, 16(5):1188–1200, Aug 2014.

[8] L. Ye and E. Keogh. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 947–956, New York, NY, USA, 2009. ACM.

[9] C.-C. M. Yeh, P.-K. Jao, and Y.-H. Yang. Awtoolbox: Characterizing audio information using audio words. In *ACM Multimedia*, 2014.

[10] C.-C. M. Yeh, L. Su, and Y.-H. Yang. Dual-layer bag-of-frames model for music genre classification. In *ICASSP*, 2013.