Kyle Jackson
CS 674
Prof. Lin
3/15/2017

HW 2

In this assignment, I tried three different strategies: (1) discovered the time series pair motif in the training data leveraging the brute force motif discovery algorithm, which I implemented myself, (2) leveraged the TSMining R package for subsequence motif discovery and did classification via 1-nearest neighbor, (3) converted each time series to SAX representation, segmented into words, and then performed classification using the bag of patterns approach. Experiment (1) should be viewed as the main work of this assignment, with (2) and (3) as secondary experiments, with (2) being incomplete. The results from HW1 and this assignment are in *Table 2.*

Approach

For experiment (1) the brute force algorithm for motif discovery (*Table 1*) was used to find time series motif pairs as defined in section 2, *Definition 3* of [1]. The distance metric was Euclidean distance. Specifically, for each dataset I took the train.txt file and split the data into n categories, where n represents the number of classes that exist in that dataset. Then I find the time series motif pairs which are a time series pair that are most similar among all possible pairs *within* that class. Then I used these pairs from each class to create a new train data set to be used for classification. This new training file consisted of $2n$ time series, since each class had a pair of time series motifs. Classification was then performed using 1-nearest neighbor. The results for this experiment is in *Table 2*, (1).

Table 1: Brute force motif discovery [1]

Algorithm		Brute Force Motif Discovery
Procedure		$[L_1, L_2]$=*BruteForce_Motif(D)*
In:	D	Database of Time Series
Out:	L_1, L_2	Indices for a Motif
1	best_so_far = 999999.0	
2	for i = 1 to m	
3	for $j = i+1$ to m	
4	if $d(D_i, D_j)$ < best_so_far	
5	best_so_far = $d(D_i, D_j)$	
6	$L_1=i, L_2=j$	

Experiment (2) leveraged the TSMining[1] R package which provides several functions for mining numeric data. I used the univariate motif discovery function, which required the conversion of the time series windows into a SAX representation. This was done since PAA and SAX have been thoroughly analyzed and determined very effective for time series approximation [2]. For this, I conducted the time series subsequence motif discovery in R with the default parameters for motif discovery in TSmining. These included a window size of 10, no overlap of windows, word size of 5, alphabet of 3, and mask size of 3 for random projection. Random projection has been shown to be effective for motif discovery, as described in [3]. The SAX word representations of each motif were then placed into a "sentence" which represented a time series. In other words, given a univariate time series of numeric values, the new time series consists of the motifs in the SAX representation. A sentence may look like t_i = "aabc cbaa abca" for time series t_i where each word in the string represents a motif. Each sentence is then converted to a bag of words (or patterns) representation. The representation of the entire

[1] Documentation here https://cran.r-project.org/web/packages/TSMining/TSMining.pdf and code here https://github.com/cran/TSMining

data matrix is a term frequency inverse document frequency (tf-idf) matrix. Scikit learn's TfidfVectorizer[2] module is used for this. Classification was performed using 1-nearest neighbor with Euclidean distance. Results for this are in *Table 2,* (2).

The approach for experiment (3) was to convert each time series into a SAX representation where the window is equivalent to the size of the time series. This was done using the SAX function built into TSMining. The key parameter for this conversion is alphabet size denoted by a, which can be seen in *Table 2.* Each time series are broken up into words of length w, which remained a constant value of $w = 4$ throughout the experiments discussed in this paper. Like experiment (2), these then formed sentences of SAX words which were then used for classification. The data matrices were then converted to the tf-idf representation. Finally, classification is done using 1-nearest neighbor and cosine similarity as the similarity metric. Results for this are in *Table 2,* (3).

The common theme across experiments (2) and (3) were the bag of patterns approach and tf-idf representation for classification. I decided to go with this method because it increases the value given to each unique word proportionally to the number of times it appears in the document, but reduces the value of the word as the frequency of the word in the corpus, or the collection of documents, increases. I chose this method over the word frequency representation, which is noisy and has been shown to provide poorer results than a tf-idf representation with text data. While cosine similarity, which has been shown more accurate than Euclidean distance on text data, was used for experiments (2) and (3), Euclidean distance was used for experiment (1).

Accuracy was calculated as follows:

$$Accuracy = (TP + TN)/(TP + FP + FN + TN)$$

Where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. This metric was used because of its widespread use and the fact that the data sets appeared to have relatively balanced label distributions.

My 1-nearest neighbor implementation from HW1 was used in this assignment across all experiments since it has been shown to be one of the most effective classification methods for time series. As noted before, the distance metric used was Euclidean distance.

Results

The results are in *Table 2*. 1-nearest neighbor with dynamic time warping (DTW) and Euclidean distance outperformed the motif based classification methods from this homework assignment. Experiment (1), which used the time series pair motifs to perform classification was one of the worst performing out of all the methods. Experiment (2) also did poorly, even though not all the data sets were able to be scored. The results from Experiment (3) were surprising, doing better on average than the other experiments performed in this homework. One thing to note is the increase in the alphabet size when converting an entire time series to a SAX representation. As the alphabet size a increased the results got better. This makes sense since it allowed more variability in the SAX representation of the time series. The surprising aspect was that splitting the time series SAX representation into chunks (or words) and using the bag of patters approach did relatively well, with data set 5 having an accuracy of 95.26%.

An interesting observation is that data set 5 consistently has the highest accuracy across all methods. Given that data set 5 has the largest number of instances, this lends to the conclusion that all these methods can perform well given a large enough training set. On the other hand, it could be that the time series from different classes in data set 5 are just highly distinct from one another.

[2] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Table 2: Results in accuracy

	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5
1-NN Euclidean Dist	85.22%	78.29%	51.65%	78.88%	**99.55%**
1-NN DTW without w	**99.67%**	**83.43%**	**59.09%**	**79.20%**	97.98%
1-NN DTW with w (20%)	32.89%	82.29%	9.91%	21.76%	91.23%
10-NN, euclidean dist.	65.22%	71.43%	45.04%	67.36%	98.65%
(1) TS pair motif	55.22%	40.00%	26.86%	36.16%	89.21%
(2) Subsequence motif	*	28.57%	23.14%	*	*
(3) BoP, a=5	52.11%	25.71%	40.08%	38.08%	95.01%
(3) BoP, a=15	53.11%	41.71%	46.28%	48.96%	94.21%
(3) BoP, a=20	49.44%	43.43%	41.73%	49.44%	95.26%

* Cells empty due to issues with the TSMining package not being able to process a few, but not all, of the data files. Regardless, my approach for using subsequence time series motifs proved ineffective, as shown on data sets 2 and 3.

Discussion

The experiments conducted in this homework were an excellent learning experience in motif discovery. While experiments (1) and (2) would be considered motif discovery, experiment (3) was tangentially related to motif discovery, but leveraged some of the same techniques, such as converting the time series to an approximation. The common theme in my work is that I converted these motifs into words which were used in the tf-idf matrix representation, except for (1). In (1), motifs are a pair of time series from each class which were used for classification.

Another lesson learned from this assignment are that motifs are patterns that likely occur primarily in the same class. These patterns can be used to classify unseen data by either comparing whole time series or subsequence matching. To select the most discriminative motifs, we look for either the most similar pair of possible time series in a database or the most similar pair (or a number >2) of subsequences within a time series.

The number of motifs in (1) were set to two per class in each data set since I was using the pair motif algorithm described in *Table 1.* This meant that for *n* classes, there were *2n* motif time series used for classification of unseen data. In (2), the number of motifs are dependent on the number of parameters used, particularly the sliding window size, how much overlap of the windows exist, the size of the alphabet, and the length of the SAX word. In (3), the number of motifs is dependent on the size of the window that broke the sequence of SAX letters, each a one to one mapping to the numerical values of the time series, into words. In this experiment, a value of 4 was used. Varying this parameter would impact the number of words and how well these words impact classification.

Conclusion

The experiments conducted as part of this experiment were exploratory and should not be considered anything but that. Several non-optimal choices were made, such as parameter values for subsequence motif discovery and the size of the word in (3). Future work would include a more thorough literature search of motif based classification of time series and ensuring that the implementations in this assignment are executing properly.

References

[1] Mueen, A., Keogh, E., Zhu, Q., Cash, S., & Westover, B. (2009, April). Exact discovery of time series motifs. In *Proceedings of the 2009 SIAM International Conference on Data Mining* (pp. 473-484). Society for Industrial and Applied Mathematics.
[2] Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, *15*(2), 107-144.
[3] Chiu, B., Keogh, E., & Lonardi, S. (2003, August). Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 493-498). ACM.