Time Series Classification Utility
User Manual

Hüseyin Kaya

April 17, 2013

Contents

1 Introduction

Time Series Classification Utility (TSCU) is a simple MATLAB program that you can use it to classify time series by choosing a couple of alignment methods including Dynamic Time Warping (DTW), Constrained DTW (CDTW) and Signal Alignment via Genetic Algorithm (SAGA).

I decided to prepare TSCU during my PhD which is about creating a new time series alignment algorithm and its application to various real-world problems. There were (and there are still) a bunch of useful tools for time series alignment, but none of them seem to provide a general framework for classification [1]. I also wanted to create a useful website so that people searching for a time series classification task will find all crucial information quickly.

2 Installation

TSCU is freely available from GitHub. Majority of the code lies in just one MATLAB script: tscu.m. However it is recommended to fetch the whole repository (which is about 1Mb) in order to obtain a few dependent files. You can use the following command to download the repository:

```
git clone https://github.com/hkayabilisim/TSCU.git
```

This will checkout the repository into a new directory named TSCU. After downloading the source code, you should follow the following steps.

- In order to use the alignment methods DTW and constrained DTW, you should compile the mex file dtw.c by issuing the following command on MATLAB:

  ```
  mex dtw.c
  ```

- If everyting goes well, you will have a new executable file with extension name begins with mex. In my Macbook Pro, its name is dtw.mexmaci64. If you have problems in compiling the mex file (you are very likely to face such problems, by the way), then you can look for precompiled binaries on TSCU repository.

- If you want to test the utility with the University of California, Riverise (UCR) time series repository, then you should request the dataset from Eamonn Keogh personally because I don't have permission to provide the dataset [1]. I strongly suggest you to have a copy of this large and diverse dataset if you want to do detailed analysis on alignment techniques.

- If you want to test the alignment algorithm Signal Alignment via Genetic Algorithm (SAGA), you should have Genetic Algorithm Toolbox of MATLAB. SAGA is optional, so if you don't have this toolbox, don't bother. You can still use DTW or constained DTW for alignment.

3 Running

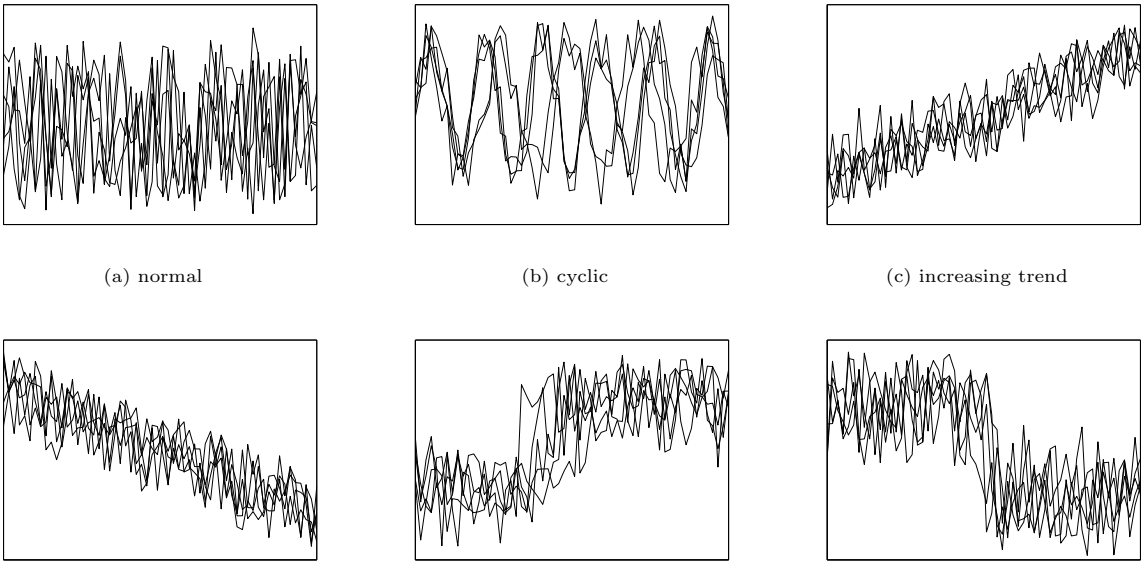
A straightforward way yo to test the installation is to run a few tasks. Let's start with the Synthetic Control dataset from University of California-Riverside (UCR) time series repository. This dataset is freely available on UCR time repository web page[1]. There are 6 different classes of time series each has length 60. There are totally 600 time series half of it is reversed for training. Some examples of the dataset are displayed in Figure 1.

After downloading training and testing set, you can classify the time series in the testing set by using the following command:

```
>> trn=load('synthetic_control_TRAIN');
>> tst=load('synthetic_control_TEST');
>> tscu(trn,tst)

Size of training set.....................: 300
Size of testing set......................: 300
Time series length.......................: 60
Classification method....................: 1-NN
Alignment method.........................: None
Overall Accuracy.........................: 0.880
Overall Error............................: 0.120
Producer Accuracy........................: 0.440   1.000   0.980   1.000   0.940   0.920
User Accuracy............................: 1.000   0.833   0.891   0.862   0.887   0.885
Kappa....................................: 0.856
Z-value..................................: 5.439
Confusion matrix
          1     2     3     4     5     6    UA    TO
    1    22     0     0     0     0     0 1.000    22
    2    10    50     0     0     0     0 0.833    60
    3     3     0    49     0     3     0 0.891    55
    4     4     0     0    50     0     4 0.862    58
    5     5     0     1     0    47     0 0.887    53
    6     6     0     0     0     0    46 0.885    52
   PA 0.440 1.000 0.980 1.000 0.940 0.920
   TO    50    50    50    50    50    50         300
Time elapsed (sec).......................: 1.08
```

[1] http://www.cs.ucr.edu/ eamonn/time_series_data/.



(a) normal (b) cyclic (c) increasing trend

(d) decreasing trend (e) upward shift (f) downward shift

Figure 1: Some examples from 6 different control charts used in the synthetic control dataset.

Output of TSCU is pretty self-explanatory. As you see from the output, TSCU does not use an alignment algorithm in its default form. Overall error in this case is 0.12 which is the same as the published error on UCR web site. TSCU also outputs confusion matrix which can sometimes be usefull for further analysis.

If you want to use Dynamic Time Warping (DTW) as the alignmnt method for the same dataset, then you can append the following options:

```
>> trn=load('synthetic_control_TRAIN');
>> tst=load('synthetic_control_TEST');
>> tscu(trn,tst,'alignment','DTW')

Size of training set.....................: 300
Size of testing set......................: 300
Time series length.......................: 60
Classification method....................: 1-NN
Alignment method.........................: DTW
Overall Accuracy.........................: 0.993
Overall Error............................: 0.007
Producer Accuracy........................: 0.960   1.000   1.000   1.000   1.000   1.000
User Accuracy............................: 1.000   0.980   1.000   1.000   0.980   1.000
Kappa....................................: 0.992
Z-value..................................: 24.884
Confusion matrix
          1     2     3     4     5     6    UA    TO
    1    48     0     0     0     0     0 1.000    48
    2     1    50     0     0     0     0 0.980    51
    3     0     0    50     0     0     0 1.000    50
    4     0     0     0    50     0     0 1.000    50
    5     1     0     0     0    50     0 0.980    51
    6     0     0     0     0     0    50 1.000    50
   PA 0.960 1.000 1.000 1.000 1.000 1.000
   TO    50    50    50    50    50    50         300
Time elapsed (sec).......................: 5.77
```

4 Options

There are varios options that you may want to use. Each option should be given with a key-value pair like `tscu(trn,tst,'Option1','value1','Option2','value2')`. Available options can be listed by running `help tscu` or `doc tscu` on MATLAB assuming that `tscu.m` is on the working directory or in the path.

4.1 'Classifier'

This option sets the classifier that is used to classify the instances in training and testing sets. Currently you can only set `'K-NN'` classifier. `'K-NN'` is also the default classfier with $K = 1$.

4.2 'Alignment'

This option specifies the alignment algorithm used in distance calculation between any two time series. The following values are available.

`'None'` *(default)* In this case no alignment takes place and usual Euclidean distance between two time series is taken as the distance.

`'DTW'` Standard Dynamic Time Warping is used in its original simple form without any lower bounding or bands. The implementation is based on The UCR Suite[2]. The code is written as a MATLAB MEX file to gain some speed. However one should compile the mex file *dtw.c* to be able to use it in TSCU.

`'CDTW'` Constrained Dynamic Time Warping in which the path is constrained in Sakoe-Chiba band. It is implemented again in the same mex file *dtw.c* however one should use the additional option `'DTWbandwidth'` to set the width of the band.

`'SAGA'` Signal Alignment via Genetic Algorithm. It uses smooth monotone functions suggested by Ramsay[2], but solves the best parameter set by using Genetic Algorithm[3]. It is more accurate but slower than others.

4.3 'DTWbandwidth'

6 *default* This parameter is used when one choose `'CDTW'` as the alignment method. It is the width of the Sakoe-Chiba band defined in percentage. Setting it to 100 is the same effect as running DTW.

4.4 'SAGACostFunction'

This options specifies the cost function which plays an important role in SAGA. Speed and performance of SAGA is directly affected with this choice. This option is closely related with SAGA. Therefore if you set this option but not choose SAGA, it will be silently ignored. All cost functions calculates the same distance below but with slightly different ways

$$d = ||x - y(w(t))||$$

where x and y are the time series and $w(t)$ is the warping path determined by the alignment algorithm. Warping function w is obtained by solving an ODE suggested by Ramsay [2]. The weight vector of the ODE is the free variable of the cost function.

[2] http://www.cs.ucr.edu/%7Eeamonn/UCRsuite.html

`'Jcost0'` This cost function first discretize the unit interval and obtains a time vector whose length is equal to the length of time series. Then, it solves the ODE by using the weight vector in order to find the warping path. One of the time series is warped by using the warping function. Warping is achieved by using the linear interpolation. Finally the Euclidean distance between the warped time series and the unwarped one is calculated. The related MATLAB excerpt is shown below. Solution of ODE, interpolation and Euclidean distance are all conducted in MATLAB without any MEX. So it is rather slow but portable i.e. you don't need to compile a bit, everything is in the same MATLAB file.

```
t=linspace(0,1,length(x));
J = @(s) norm(interp1(t,y,ramsay3(t,s))-x);
```

`'Jcost1'` This cost function is equivalent of `'Jcost0'` but it is rewritten as a MEX file (`'Jcost1.c'`) in order to gain some speed. The MEX file itself uses a LAPACK call (`dgesv`) to solve a linear system of equation. In normal circumstances, it should be faster than `'Jcost0'`. However, it is generally not.

`'SAGAOptimizationMethod'`

The alignment method SAGA relies on minimization of the cost function defined in `'SAGACostFunction'`. By default, minimization of the cost function is achieved by using Genetic Algorithm. However some other optimization techniques may also be used.

`'GA'` *(default)* By default genetic algorithm is used to find the minimum point of the cost function.

`'Simplex'` By choosing this value, Nelder-Mead Simplex method is used as a minimization routing [4]. It requires an initial point which can be specified with `'InitialSolution'` option.

4.5 `'SAGAInitialSolution'`

Some of the optimization algorithms specified in `'SAGAOptimizationMethod'` option may need an initial solution. For instance `'Simplex'` requires a starting point. By default it is set to zero vector with length `'SAGABaseLength'`.

4.6 `'SAGABaseLength'`

It is the number of spline bases used in ODE proposed by Ramsay. Default value is 8.

4.7 `'LogLevel'`

There are eight log levels whose range starts from "absolute silence" to "display everything": `'Emergency'`, `'Alert'`, `'Critical'`, `'Error'`, `'Warning'`, `'Notice'`, `'Info'` (*default*),and `'Debug'`.

4.8 `'MATLABPool'`

If your MATLAB distribution includes Parallel Computing Toolbox, then one can feed the name of parallel pool into TSCU so that any suitable loop is run parallel. If your computer has a multi-core CPU and you have this toolbox, setting this option to `'local'` will enable MATLAB to distribute the workload to the available cores. Currently, only the "for" loops related with K-NN classifier have been converted to "parfor" counterparts. If you choose K-NN and enable this option, it is high likely to get a speed-up of factor 4 by using a quad-core processor.

4.9 `'DisplayInputData'`

If it is set to `'yes'`, training and testing data will be displayed before classification.

4.10 'trainingRatio'

If the data is not already divided, it is divided into training and testing set

5 Examples

In order to classify synthetic control dataset with default options you can use the following commands provided that you first downloaded the dataset:

```
trn=load('synthetic_control_TRAIN');
tst=load('synthetic_control_TEST');

tscu(trn,tst)
```

You can specify the alignment algorithm by using the option `'alignment'`. For instance:

```
tscu(trn,tst,'alignment','DTW')
```

choose good old Dynamic Time Warping method. In order use constrained DTW, you can use `'CDTW'` together with `'DTWbandwidth'`.

```
tscu(trn,tst,'alignment','CDTW','DTWbandwidth',6)
```

6 Known issues

- TSCU can not use multi-channel time series. However, one can first reduce the number of channels to a single channel by using straightforward technique of summing the channels or by using some other dimensional reduction algorithms. Choosing the right approach really depends on the application, so I didn't implement such methods in TSCU. Channel reduction is left to the responsibility of the user.
- `'Jcost1'` is slower than `'Jcost0'` although the former is written in MEX.

References

[1] Keogh E., Zhu Q., Hu B., Hao Y., Xi X., Wei L., and Ratanamahatana C. A. The ucr time series classification/clustering homepage:. Last checked on 4 Feb 2013.

[2] J. O. Ramsay and X. C. Li. Curve registration. *Journal of the Royal Statistical Society Series B-statistical Methodology*, 60(Part 2):351–363, 1998.

[3] Hüseyin Kaya and Şule Gündüz-Öğüdücü. Saga: A novel signal alignment method based on genetic algorithm. *Information Sciences*, 228(0):113 – 130, 2013.

[4] JC Lagarias, JA Reeds, MH Wright, and PE Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM JOURNAL ON OPTIMIZATION*, 9(1):112–147, DEC 21 1998.