**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE (BITS)-PILANI**


# A REPORT ON

# PREDICTIVE MODELING FOR MULTIVARIATE TIME SERIES DATA OF SPOKEN ARABIC DIGITS

Prepared for partial Fulfillment of Machine Learning Course No:

BITS F464

**Prepared by:**


| Name (s) of the Student (s) | ID. No (s) | Discipline (s) |
|---|---|---|
| **ANTIL SINGHAL** | 2013A7PS159P | **B.E. (Hons.) Computer Science** |
| **VANDIT THAKKAR** | 2013B4A7616P | **B.E. (Hons.) Computer Science M.Sc. (Hons.) Mathematics** |
| **MANVEER SINGH** | 2013A7PS196P | **B.E. (Hons.) Computer Science** |


At

BITS
Pilani

Pilani campus, Jhunjhunu, Rajasthan

**11/30/2015**

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI


## *Key Words:*

Multivariate Time Series Data, Two-dimensional singular value decomposition, Classification, 1NN algorithm


## *Areas:*

Machine learning, Speech Recognition.


## *Abstract:*

Spoken Arabic Digits Data is collected by the Laboratory of Automatic and Signals. It contains data points, consisting of successive measurements made over a time interval. Additionally, a sample for a digit doesn't have same number of frames. To deal with this Multivariate Time series data, a new approach using two-dimensional singular value decomposition (2dSVD) is proposed. 2dSVD is an extension of standard SVD; it captures explicitly the two-dimensional nature of MTS samples. The eigenvectors of row–row and column–column covariance matrices of MTS samples are computed for feature extraction. After the feature matrix is obtained for each MTS sample, one-nearest-neighbor classifier is used for MTS classification. Experimental results performed on spoken Arabic digits datasets demonstrate the effectiveness of our proposed approach.

## *Problem Statement:*

As part of application of Machine leaning techniques studied, select an appropriate method and apply it for predictive modeling of Multivariate time series data of spoken Arabic digits. Given a dataset of any number of frames for a digit, application should be able to classify it correctly.

## *Assumptions:*

In this project of spoken digit recognition, we take input from test data file and test it with application we have developed. We assume containment of at least 20 frames and at most 60 frames in test dataset; otherwise it has to extrapolate data based on assumption or loses valuable data in respective cases. There is no assumption taken at algorithmic level.

## *Data source and description:*

Training data for this application is taken from UCI machine learning repository. Root data source is the Laboratory of Automatic and Signals, University of Badji-Mokhtar ,Annaba, Algeria. For training purpose, Dataset from 6600(10 digits x 10 repetitions x 66 speakers) time series of 13 Frequency Cepstral Coefficients (MFCCs) had been taken from 44 males and 44 females Arabic native speakers between the ages 18 and 40 to represent ten spoken Arabic digit. Each line on the data base represents 13 MFCCs coefficients in the increasing order separated by comma. This corresponds to one analysis frame. The 13 Mel Frequency Cepstral Coefficients (MFCCs) are computed with the following conditions;

Sampling rate: 11025 Hz, 16 bits
Window applied: hamming
Filter pre-emphasized: 1-0.97Z^(-1)
There are no missing attribute values.

Frame Lines are organized into blocks, which are a set of 4-93 lines separated by blank lines and
corresponds to a single speech utterance of a spoken Arabic digit with 4-93 frames. Each spoken digit is a set of consecutive blocks. In Train_Arabic_Digit.csv there are 660 blocks for each spoken digit .The first 330 blocks represents male speakers
and the second 330 blocks represent the female speakers. Blocks 1-660 represent the spoken digit "0" (10 utterances of /0/ from 66 speakers), blocks 661-1320 represent
the spoken digit "1" (10 utterances of /1/ from the same 66 speakers 33 males and 33 females ), and so on up to digit 9.

In Test_Arabic_Digit.txt, digits 0 to 9 have 220 blocks for each one. The first 110 blocks represent male speakers and the second 110 blocks represent the female speakers. Therefore, blocks 1-220 represent digit "0" (10 utterances of /0/ from the 22 speakers ), blocks

221-440 represent digit "1" (10 utterances of /1/ from the same 22 speakers 11 males and 11 females ), and so on.

For example,

-0.81101,-7.2382,1.5429,-0.64774,1.4271,0.61356,0.36516,0.088906,0.47031,0.98844,0.044692,0.20817,0.5114
-0.37028,-7.1336,1.8856,-0.34316,0.96733,0.32763,0.42988,0.50479,0.41533,0.28804,0.086109,0.6269,0.78115
0.59659,-8.3059,1.6943,-0.66611,0.34967,-0.17425,0.82077,1.2611,0.41653,0.5005,0.57163,0.45316,0.64465
1.4585,-8.1957,1.8454,-1.1496,0.8266,-0.51313,0.067443,0.25637,0.115,-0.10915,0.085991,0.69064,0.33769
...
0.19664,-1.8318,0.94856,0.29956,0.13058,-0.94998,-0.8854,-0.91416,0.055284,-0.080658,-0.21335,-0.91618,0.017725
-0.53611,-2.4703,1.069,1.2785,0.47267,-0.72266,-0.51269,-0.78476,0.3939,-0.076555,-0.37365,-0.36757,-0.66361
-0.34113,-2.1089,0.89114,0.6747,0.3096,-1.0392,-1.4174,-0.84227,0.21493,-0.22037,-0.099087,0.10891,-0.29597
,,,,,,,,,,,,

Represents one sample, which contains **n** number of frames. Now, challenge is **n** varies over different datasets. So before applying dimension reduction techniques or classification we need to normalize data.

**Reference:**

https://archive.ics.uci.edu/ml/datasets/Spoken+Arabic+Digit

.

## Scaling:

We need to scale down or scale up data from variable **n** frames to some standard **m** before applying dimension reduction and classification techniques.

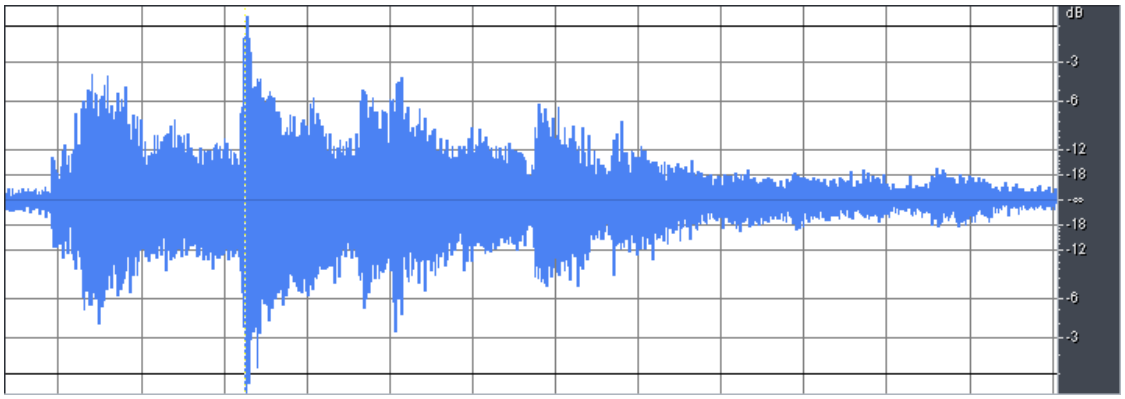


Figure 1.1

---------- **n** frames ----------

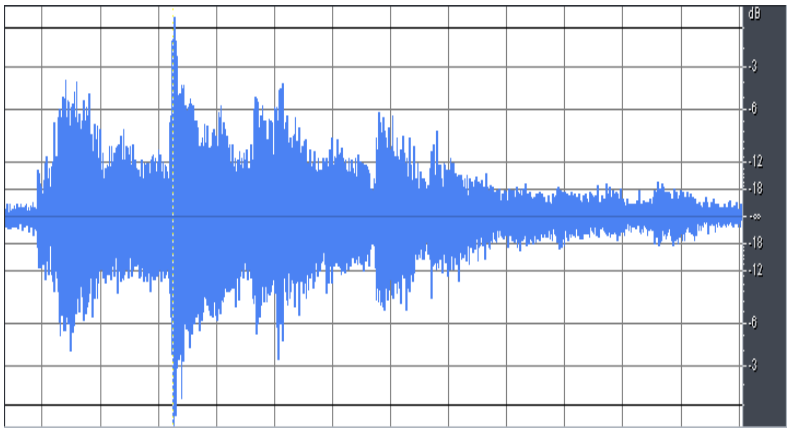


Figure 1.2

---------- **m** frames ----------

(**n** frames scaled down to **m** frames)

Image source: http://i.stack.imgur.com/

1. Iterating over all 6600 training datasets, we look upon how many frames does a particular dataset have. If it is greater than 35, then we call scaleDown, otherwise we call scaleUp.

2. We chose m=35 in order to minimize scaling so that we lose minimum data and add very less dummy frames under scaleUp assumption.

3. For scaling down, we calculate scaling factor and those rows which are selected for retaining in scaled down version of sample are put in new frame at appropriate index frame.

   Scaling factor = no_of _frames_in_sample (**n**) / no_of_frames_in_scaled_vesion(**m**)

4. For scaling up, we calculate scaling factor and those indexes of scaled up version which will get exactly those rows which we had in original version. Here we take 2 assumptions.
   (i) **n**<2*__m__
   (ii) Additional frames in scaled up version can be found using interpolation.

Thus, by choosing appropriate indexes and proper and tolerable interpolation techniques , we fill all m frames with data.

Scaling factor = no_of_frames_in_scaled_vesion(**m**)/ no_of _frames_in_sample(**n**)

At the end of scaling we have 6600 samples, each of which is 35 x 13 2d matrix. Now, we need apply dimension reduction technique as part of preprocessing technique.

## *Preprocessing involved:*

- **Need of dimension reduction :**

The classifier we need to implement here, 1-NN classifier suffers with curse if dimensionality. So, we can't apply it directly, as we have 6600 samples to be classified in 10 classes, each of 35 frames with 13 dimensions. So we apply dimension reduction technique, 2dSVD, to reduce each sample to **r** frames of **s** dimensions.

- **Implementation:**

- First the average of all the frames is calculated. This is a 2D matrix of 35x13.

- This is used to calculate the deviation for all frames. The deviation for each frame is also 35x13 matrix. These deviations are used to calculate covariance of rows and column. We get a row co-variance matrix **R** of 35x35 and a column co-variance matrix **C** of 13x13.

- Eigen vectors of matrices **R** and **C** are calculated and the top **r** and **s** vectors are taken respectively. These vectors form the columns of matrices **Ur** and **Vs**. Since **R** is matrix of 35x35, the **r** Eigen vectors form **Ur** of dimensions 35xr. Similarly, dimensions of **Vs** are 13xs.

- Each sample is then transformed using Us and Vs according to the formula:

$$(Ur')_{rx35} \; X \; (Sample_i)_{35x13} \; X \; (Vs)_{13xs}$$

This gives the reduced sample of **rxs**. Hence, samples are reduced from 35x13 to **rxs**.

## *Classification algorithms used:*

**1-nearest neighbor classifier** is used for this project for predicting spoken digits application.

1. After preprocessing techniques applied, we have 6600 examples with **r** rows and **s** dimensions.

2. So we have training data ready in preferable form we can apply 1-NN algorithm to. We find distance of test example

3. We calculate the distance of the test input from each sample. Here we calculate $||\;||_1$ (L2-norm), which is the sum of mod values of each column of the difference matrix obtained by subtracting transformed test input and sample.

4. We have obtained 6600 distances from step 3 (one for each sample). Now minimum of these distances leads that the digit corresponding to that class is most likely to get matched with test input. So, that digit is final output of this method.

## *Results:*

We tested this application using 2200 examples we obtained from Test_Arabic_Digit.csv. Each data sample was matched with all training examples, and it was labeled on basis of nearest neighbor. In our results, we found that 2130 examples were correctly classified. Thus, we got 96.81% accuracy for this predictive modeling application we prepared.

We always have an option of choosing parameters r and **s** of 2Dsvd on basis of which we can enhance accuracy level of this application. We tested it with different **r** and **s,** for given **m.** As we have included scaling as part of preprocessing here, we have tolerance range of parameter **m** also.

In this confined level of application**, Time taken** and **CPU usage** is not of our interest perhaps. But when we extend this application to real world and proper speech recognition instead of only digits recognition, it becomes a significant factor.

| Test | **m** (standard rows per frame scaled to) | **(r,s)** (pair of parameters in 2dSVD ) | **Accuracy** (Percentage of correctly classified samples) | **Total Time taken** (seconds) | **Classification Time** (seconds) |
|---|---|---|---|---|---|
| 1 | 35 | (10,10) | 97.27% | 27.50 | 26.258 |
| 2 | 35 | (6,6) | 96.22% | 17.59 | 16.98 |
| 3 | 35 | (20,7) | 96.95% | 49.36 | 48.07 |
| 4 | 35 | (3,7) | 93.5% | 11.28 | 10.65 |
| 5 | 40 | (20,7) | 96.54% | 49.13 | 47.75 |
| 6 | 40 | (10,10) | 96.81 | 28.08 | 26.75 |
| 7 | 40 | (6,13) | 96.31% | 19.91 | 18.58 |
| 8 | 40 | (6,10) | 95.86% | 18.50 | 17.18 |
| 9 | 30 | (20,7) | 97.04% | 48.89 | 47.72 |
| 10 | 30 | (10,10) | 97.36% | 28.19 | 27.05 |
| 11 | 30 | (15,10) | 97.22% | 39.33 | 38.12 |
| 12 | 30 | (5,10) | 96.04% | 16.13 | 14.98 |

## *Conclusions:*

We tested 2200 test examples in trained application of predictive modeling for spoken digits data. We got an average of about **96% accuracy**. We can conclude here that even if we have large MTS data, we can apply appropriate dimension reduction techniques and classification algorithms for better results. In this example, we have timestamp-ordered collection of data with unequal number of frames as no two persons can speak a digit in same time (at scale of milliseconds). So we applied tolerable scaling algorithms so that minimum data is lost. There is always trade-off between different requirements of whole process. In this case, we chose 1-NN as better and safer option for classification, but it has limitations, so we needed to scale data and then reduce dimensions. If we could choose a classification algorithm in which scaled data is not required, we would have used it and removed scaling part but we didn't want to compromise with accuracy. So, simply we can choose different methods for sub processes and parameters within them on basis of requirement of situation in terms of accuracy needed, time limitation and resources constraint.

---