

## IE 582 Statistical Learning for Data Mining

### Homework 3, due December 6<sup>th</sup>, 2021

Instructions: Please solve the following exercises using R (<http://www.r-project.org/>) or Python (<https://www.python.org/>). You are expected to use GitHub Classroom and present your work as an html file (i.e. web page) on your progress journals. There are alternative ways to generate an html page for you work:

- A Jupyter Notebook including your codes and comments. This works for R and Python, to enable using R scripts in notebooks, please check:
  - <https://docs.anaconda.com/anaconda/navigator/tutorials/r-lang/>
  - <https://medium.com/@kyleake/how-to-install-r-in-jupyter-with-irkernel-in-3-steps-917519326e41>

Things are little easier if you install Anaconda (<https://www.anaconda.com/>). Please export your work to an html file. Please provide your \*.ipynb file in your repository and a link to this file in your html report will help us a lot.

- A Markdown html document. This can be created using RMarkdown for R and Python Markdown for Python

Note that html pages are just to describe how you approach to the exercises in the homework. They should include your codes. You are also required to provide your R/Python codes separately in the repository so that anybody can run it with minimal change in the code. This can be presented as the script file itself or your notebook file (the one with \*.ipynb file extension).

The last and the most important thing to mention is that academic integrity is expected! Do not share your code (except the one in your progress journals). You are always free to discuss about tasks but your work must be implemented by yourself. As a fundamental principle for any educational institution, academic integrity is highly valued and seriously regarded at Boğaziçi University.

#### Task 1 – On the use of distance information for UwaveGesture Recognition Task

Homework 2 introduced the description of the gesture recognition task with the accelerometer data. This task will involve evaluation of alternative strategies using both training and test data provided on <https://drive.google.com/drive/u/1/folders/13553neknu7U8why55KM1WrjgkA9IJKm> (same link as in Homework 2).

Use the UWave data provided in the second homework for this task. In the second homework, you were given only the training data. This task requires performing classification task on the test data and evaluate the performance of certain classifiers. Test data is uploaded as \*.zip files for each axis (i.e. X, Y and Z). It has the same format as the training data.

- a) Suppose we decided to apply a nearest-neighbor (NN) classifier to find the labels of test instances. You can use the strategy you employed when you apply PCA to this data in Homework 2 (i.e. concatenation of the axes). Propose **two distance measures for computing similarity between two time series**. The distance calculation on the concatenated time series implicitly weights the distances of each axis in an equal way. For each distance measure

alternative, use the training data to identify the ideal value of  $k$  which minimizes the error of a 10-fold cross-validation.

- b) Using the value of  $k$  (identified for each distance measure) in part (a) and evaluate your final performance on the test data and present your results in a (8-by-8) confusion matrix, showing the counts for actual and predicted labels. In addition, quote the runtime and accuracy for your results.
- c) The observations from different axes are weighted equally if we compute the distance over each axis and sum them to obtain a final similarity measure. Is this reasonable? For example, we can compute the distance as below:









$$finalDist = w_1 Dist_x + w_2 Dist_y + w_3 Dist_z$$

where  $Dist_x$  is the distance based on the acceleration only on X axis,  $Dist_y$  is for Y axis and so on. Do you think weighting the distances over different axes to obtain a final similarity measure makes sense for classification? Why?

## Task 2 – Linear models on alternative representations of the data

Recall that most of the machine learning problems assume that we have a nice and informative representation of the data (after data selection, preprocessing and transformation steps). In part (a), we have preprocessed the information from each axis to obtain our feature vector (of 3T length which keeps the information about X, Y and Z axis).

Suppose we are willing to perform a binary classification task to identify if a test time series is from Class 3. As a reminder you can find the class definitions in Figure 1.

1	2	3	4
			
5	6	7	8
			

**Figure 1:** Gesture vocabulary considered by Liu et al. (2009). The dot denotes the start and the arrow the end

- a) Train a logistic regression model on the training data and use the model to make a prediction on the test data. Note that you will obtain probabilistic predictions (i.e. probability of a time series being from Class 3 if you encoded Class 3 as 1 in binary classification setting). This will require you to select a threshold since 0.5 as a threshold may not work well under this imbalanced class setting. To make

things easier, use the ratio of Class 3 instances in the training data as threshold. Use the learned model to predict the class for test data. Present your results in a (2-by-2) confusion matrix.

**b)** An advantage of logistic regression is related to the interpretability however when we have large number of features together with a method without penalization, it is harder to interpret the results. Therefore, an alternative way is to train a logistic regression model with lasso penalties. This will require you setting of penalization term (namely lambda). Use 10-fold cross-validation to determine your ideal lambda level based on binomial deviance (Note that we have used accuracy as primary metric to determine the lambda in class, however this strategy may not work well for the imbalanced data). You can check [http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/MLSC\\_Lec4.pdf](http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/MLSC_Lec4.pdf) for details of binomial deviance. This is also referred to as logistic loss. If you are using “glmnet” package in R, “type.measure” can be set to “deviance” which is the default value. If you are Python user, sklearn module has “LogisticRegressionCV” function in which you can provide the scorer as “metrics.log\_loss”.

Once you determine your best lambda value using 10-fold cross-validation, perform classification on test data similar to part a and compare your results. Comment on the regression coefficients. Is there any interesting information? Try to interpret the model.

**c)** An alternative way to represent the feature matrix on a new space to introduce nonlinear relations is to use distance matrix as a feature matrix. For example, we have 896 training instances and the observations over time are used as features in the previous tasks (i.e. we worked on  $N$  by  $3T$  matrix). Recall that multidimensional scaling also works on distance matrices and we have mentioned that it can handle nonlinear relations (Homework 2 also aims at revealing such an information). This non-linearity stems from the use of Euclidean distances. Use of Euclidean distance as input to a learning algorithm allows for handling nonlinear relations\*. In other words, your features keep the nonlinear information.

\*We will have further discussion on this behavior when we cover support vector machines. This type of transformations are discussed under “distance-based kernels” which is out of scope for now. Additional information is provided in case you are willing to perform research on distance based transformations.

Given this information, you are expected to transform your training data to distance information (i.e.  $N$  by  $N$  matrix). Note that you need to perform a similar transformation to your test data. In other words, you need to calculate the distance of each test instance to training instance to obtain a distance based representation for your test data. This will be an  $N_{\text{test}}$  by  $N$  matrix ( $N_{\text{test}}$  refers to the number of test instances) where each entry  $(i,j)$  refers to the distance of test time series  $i$  to the training time series  $j$ . You can use Euclidean distance as your distance measure.

Perform the same training and test strategy as in part b but use the distances as your new feature matrices. Comment on the regression coefficients. What do they imply under this new representation setting?

**d)** Provide an overall comparison on the results you obtain for each part (over all tasks). You can compare test accuracy of each alternative method you developed.

**Reference**

J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. Pervasive Computing and Communications, IEEE International Conference on, 0:1-9, 2009.  
(link: <http://www.ruf.rice.edu/~mobile/publications/liu09percom.pdf>)