ML4QS: Assignment 2

Martin Banchero and Dennis Dekker

VU, Amsterdam, NL

1 Chapter 5

1.1 Pen and Paper 2

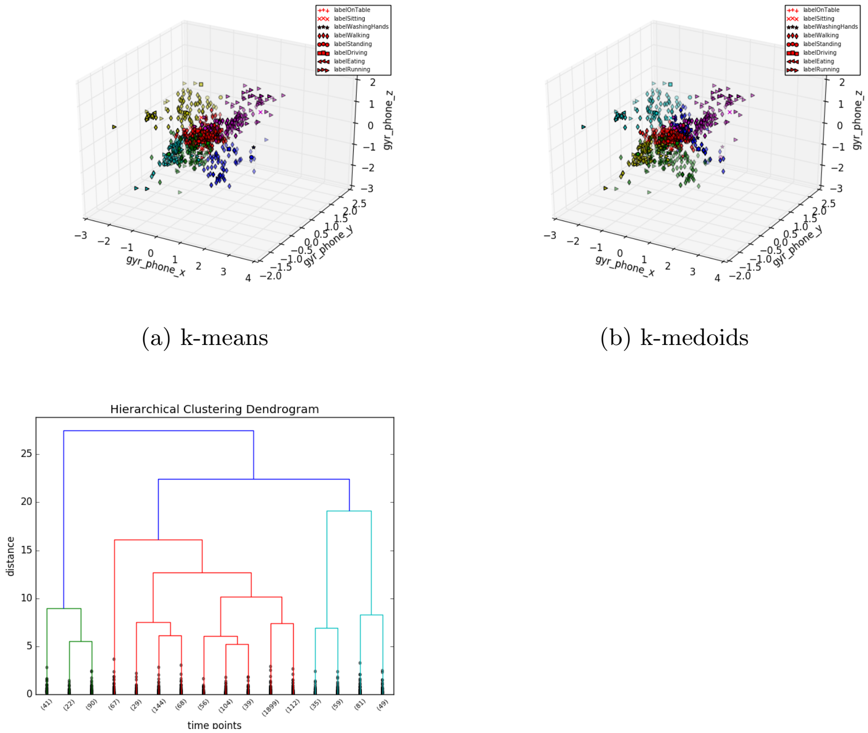
A short signal in a time series could align with a long signal, which should not always be logical/correct. An example of this is Therefore, in order to prevent long lagging periods, we could introduce a lagging penalty, which adds to the score whenever the dynamic time warping makes a 'jump' in time (go up or right instead of diagonal). By making the penalty small, the algorithm can still choose to make a lag, but will prefer alignments without long stretches of lag. This if for example the case when we want to align two radio signals with a different frequency. Here the lagging of a signal can be fine if the frequencies of the signal is similar, however when the frequencies are not similar we don't want to align the signal, as the signals can have two different meaning.

1.2 Pen and Paper 7

The subspace clustering is useful with high dimensional data, this is because some features are irrelevant and also the noise might mask some clusters [3]. An example of highly dimensional data is gene expression data. In this data, the genes are the features and the instances the patients/samples. The patients have different diseases and the goal is to find the disease of each patient based on the expression data. Subspace clustering could really help to find patients with similar diseases and then cluster them based on the gene expression. This could really help to decrease the computational time and give more insight into the important genes.

1.3 Coding 1

The clusters observed for accelerometer and gyroscope data seem to be different, as shown in figure 1 for the gyroscope and figure 2 for the accelerometer. It can clearly be seen that the accelerometer data is clustered more separately, and that the clusters contain more of the same labels. For the gyroscope the walking and running cluster together, and the labels *On table*, *Sitting* and *Standing* cluster together for both the gyroscope and the accelerometer.



(a) k-means

(b) k-medoids

(c) Agglomaritive clustering

Fig. 1. Different clustering methods applied to gyroscope data.

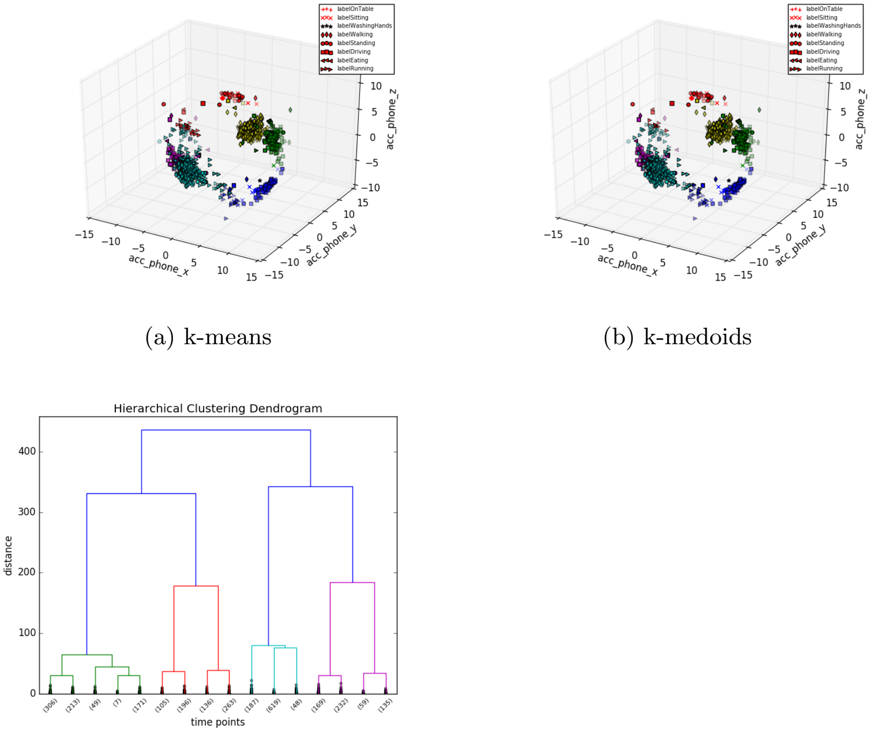
1.4 Coding 2

In this section the features for accelerometer (X, Y and Z direction) were chosen to cluster on. We used k-medoids and hierarchical clustering to see which of the two would cluster the data better. The results are displayed in figures 3a and 3b respectively.

2 Chapter 6

2.1 Pen and Paper 1

In this case the difference is based in in the sense that the best fitting function is not always a good function. This is because the best fitting model on the data could be fitting so good due to overfitting. When a model overfits it looses its generalizability of the real data and will therefore score worse than a not overfitted 'good' model, while its training score can be higher.



(a) k-means

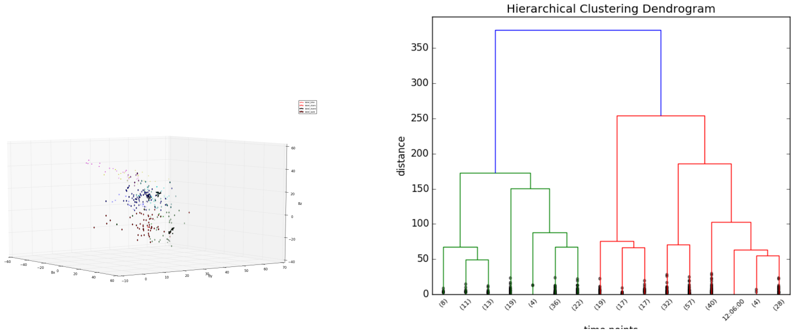
(b) k-medoids

(c) Agglomaritive clustering

Fig. 2. Different clustering methods applied to accelerometer data.

2.2 Pen and Paper 7

For every θcut the datapoints are classified differently. When the θcut decreases, the classifier will be less strict and will assign class 1 to more datapoint. Every point along the ROC curve can be associated with a new θcut value, as each point is a different classification, where moving to the right will assign more datapoints to class 1 and a lower θcut. For θcut equals zero all datapoint are predicted to be class 0. This leads to a false positive rate of 0, as no false positives can be predicted when there are no positives. Next to this, the True positive rate will also be 0, as there are no positively assigned (class 1) datapoints. On the other side, a θcut of 1 will lead to a False positive rate of 1, as all datapoints are classified positively, so all datapoint with a true class of 0 will be predicted to be 1. Similarly, the True positive rate will be 1 as all datapoint will be assigned class 1, so all datapoint with a true class of 1 will be assigned 1, which lead to a TPR of 1.



(a) k-medoids

(b) Hierarchical clusterig

Fig. 3. Different clustering methods applied to accelerometer data.

3 Chapter 7

3.1 Pen and Paper 3

An online search resulted in some guidelines on the number of neurons. The number of neurons can be restricted by the following rules of thumb [2]:

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.

3.2 Pen and Paper 6

- Linear kernel: This is a simple kernel function used to separate data points using linear separation, this can be done using a line or hyperplane. In order to performed the classification a set of hyperplanes are selected between the decision boundaries and the training data. Within this margin a small subset of the data is selected as different support vectors ($w^T x + b$) to classify the data, for example for binary classification, as negative (-1) or positive$(+1)$ [1]. The size of the margin is calculated by using equation 1:

$$w^T x = ||w||||x||cos\alpha \tag{1}$$

  In this case the hyperparameter C or soft margin can be use to determined the size of the margin. For large values of C the margin is smaller leading to a lower number of missclassifying points.
- The kernel γRBF measures distance between two data points, it is represented in equation 2. In this equation, γ controls the width of the kernel.

$$k(a, b) = exp(-\gamma||a - b||) \tag{2}$$

3.3 Pen and Paper 8

This is mainly because the nearest neighbor will calculate the distance based on all features, and this can therefore easily be influenced by one feature with a divergent value. As the amount of features increase, the chance that one of the features has a divergent value, thus increasing the distance, will increase, what will lead to not clustering the datapoint to its 'true' cluster. This is not the case in model based approaches, because these models can apply weight to features based on importance and therefore filter out what is not relevant.

3.4 Pen and Paper 13

The feature selection step as a part of the preprocess data is very important, because sometimes many features are not relevant for the task that is being performed. It might happen that some features add noise or are highly correlated. Then, feature selection can help to solve this issues by leaving some unimportant features out. This leads to a model with more generalizability. Also, feature selection can help to prevent overfitting by not letting the model memorize training set related information in unimportant features, as this lowers the generalizability.

3.5 Coding 1

In figure 4 the results of the learning algorithm with normal and normal, unknown and double classes (figures 4a and 4b resp.). You can clearly see that there are a lot of points with unknown classes. Interestingly, introducing the unknown class seems to give the algorithm a way to annotate some data points as unknown if it is not really sure if it belongs to the true class. This can be seen in for example the first label *labelDriving*, where the algorithm predicts all but one point correctly, but when introducing the unknown label, performs worse by assigning three plus the previously wrong datapoint as unknown. Therefore, adding the unknown label to the algorithm has the disadvantage of misclassifying more datapoint, but the advantage of labeling most datapoint that are misclassified as unknown instead of a 'random' class.

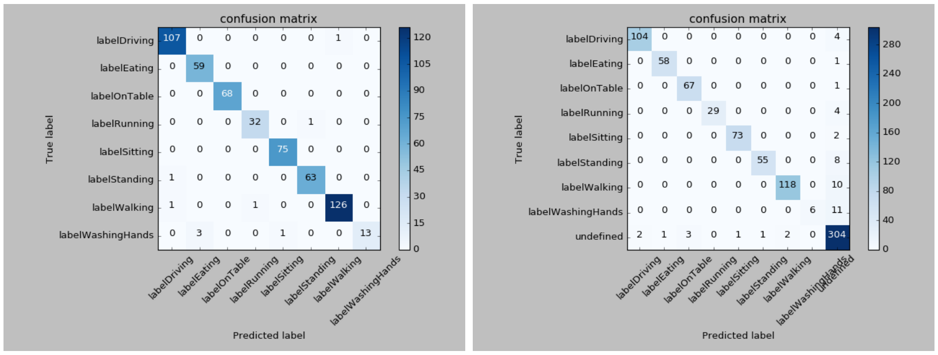
3.6 Coding 3

When compared figure 5a and 5b is possible to observed differences between both plots. In figure 5b is more evident that echo state network performs better than the rest while in 5a is not so clear.

4 Chapter 8

4.1 Pen and Paper 5

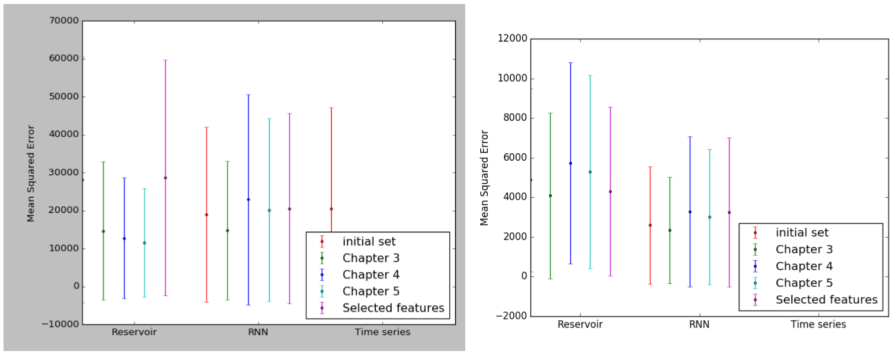
The no free lunch theorem states that there is no model capable to performed better than every other models, this because the asumptions for a model to solve



(a) Normal classes

(b) All classes

Fig. 4. Confusion matrix with normal classes and normal, unknown and double classes.



(a) Performance of rNN, Time series and Echo state network with our dataset

(b) rNN, Time series and Echo state network using crowdsignal dataset

Fig. 5. Performances comparing rNN, Time series and Echo state network

one problem might no be holded by other problems. In the particular case of the inizialitation of the random reservoir, the Echo state property has to be satisfied and there is not formal procedures to give a reservoir with the echo state property.

4.2 Pen and Paper 6

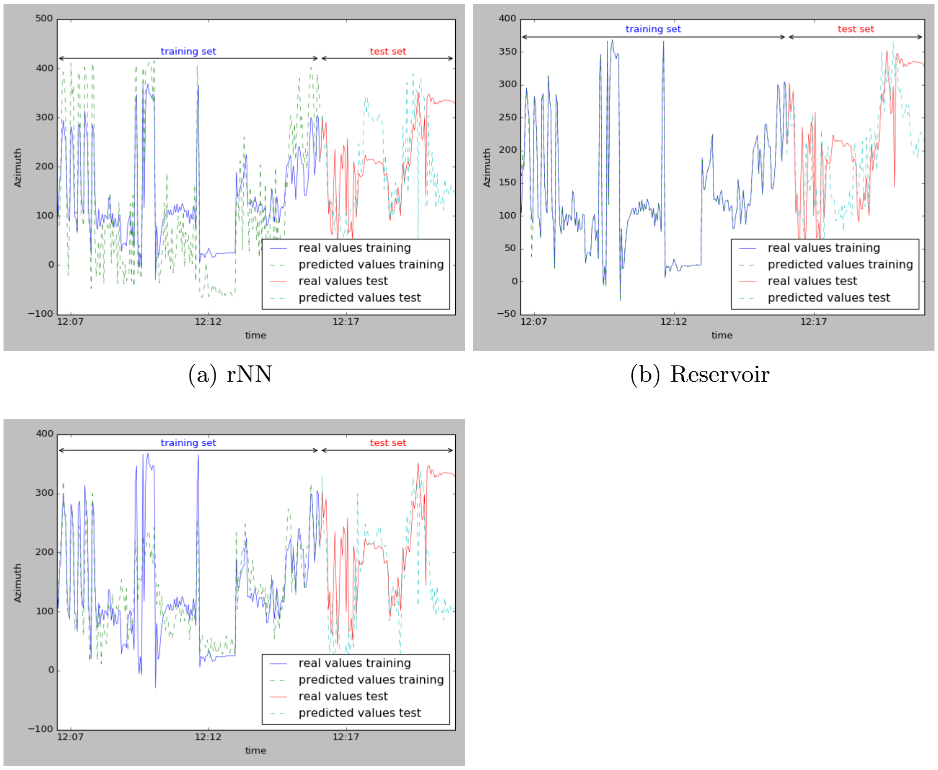
The recurrent neural networks can incorporate temporal data, in this sense the model that is produced is time-aware. In this example the goal is to predict the fitness condition of high performance athlete. Using rNN is possible to develop a model containing variable representing day, week or month to incorporate time-series together with past and current information of the fitness condition of the athlete. In this case we are aware that the rNN have problems to deal with large lags, in case to be need the long-short-term-memory(LSTM) can be used to avoid this problem.

4.3 Pen and Paper 8

The parameter optimization algorithms discussed are Simulated annealing and Genetic algorithms.

4.4 Coding 2

As is shown in 6 the best prediction is given by the Time series model, this is different to the prediction obtained with crowdsignal (result not shown).



(a) rNN

(b) Reservoir

(c) Time Series

Fig. 6. Different results for rNN, Time series and Echo state network.

5 Chapter 9

5.1 Pen and Paper 2

The user is more active if it does more high intensity activities, like cycling or running. These intensities can be extracted from the data and be used as a reward for the learning algorithm. A good definition could be amount of activities per day/week.

5.2 Pen and Paper 4

The Markov property holds for a process if the future state of the process is based only in the current state and not in previous states.

References

[1] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. *A Training Algorithm for Optimal Margin Classiiers.* Tech. rep. URL: http://www.svms.org/training/BOGV92.pdf.

[2] J Heaton. *Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks.* Artificial Intelligence for Humans Series. CreateSpace Independent Publishing Platform, 2015. ISBN: 9781505714340. URL: https://books.google.nl/books?id=q9mijgEACAAJ.

[3] Lance Parsons, Ehtesham Haque, and Huan Liu. *Subspace Clustering for High Dimensional Data: A Review *.* Tech. rep. URL: https://www.kdd.org/exploration_files/parsons.pdf.