Assignment 2

Group 15: Xinyu Hu Andrei Udrişte Rui Pan

Vrije Universiteit Amsterdam

1 Theoretical part

1.1 Chapter 5

Q2. The dynamic time warping is a method which used Euclidean distance to find the similarity between to sequences. However, this algorithm would not work well in a situation that if two person spoke in different speeds. For example, if person A spoke ”A” sound very long, but person B did not do it. In this case, the DTW could not find the similarly in the random variables. The alternative approach is the nonlinear mixed-effects models. The Nonlinear MEM would work in the measurement of nonliear similarity. [1]

Q7. Subspace clustering approach is a method that finds certain clusters in higher dimensions. Considering a dataset which contains patient's gene data, in case of more than 20000 features. However, the Alzheimer could only be found in a subset of 100 genes. The common clustering approach could not deal with those higher dimension data. But subspace clustering approach could cover all features with their dense units. [2]

1.2 Chapter 6

Q1. The best fighting function can cause overfitting, because in a real world application we are not only focusing on the training data, The task should be make a prediction on a general case. This is where the good functions make sense, because it makes a prediction on the validation set, which the model did not go through before. This means that if the model overfitted, it would not work well on the validation set. In the case of avoiding the overffting problem, a ”good function” should be chosen, because it selected the smallest in-sample error for the validation data.

Q7. The value of Θ_{cut} is influenced by ROC curve. The optimal Θ_{cut} means that the ROC curve would be a straight line, which means no negative case would be treated as a positive one. If the minimum of the cost estimate lies at either 0 or 1, then this predicted true positive. This shows the optimal Θ_{cut} means a straight line in the ROS curve. If it were a 0, then the ROS curve would be a very curved line.

1.3 Chapter 7

Q3. Too many neurons could cause overfitting and very time consuming, but too little neurons would not have enough information to make a pretty model. The rule for choosing a right number of neurons is to treat hidden layers independently. Then, the number of hidden neurons should be at most $\frac{2}{3}$ of the size of the sum of input and output neurons. [3]

Q6.

1. Polynomial kernel: This method is not accurate and effective enough compared to other kernel methods. It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.
2. Gaussian radial basis function kernel: it adds the radial basis method to improve the transformation. It is usually chosen for non-linear data. It helps to make proper separation when there is no prior knowledge of data.
3. Sigmoid kernel: this function is equivalent to a two-layer perceptron model of the neural network, which is used as the activation function for artificial neurons. [4]

Q8. Firstly, since KNN is a distance-based algorithm in higher-dimensional space, the cost to calculate distance becomes expensive. It couldn't generalize data in advance. Hence, it impacts the performance. Second, KNN needs to re-scale input variables to get an accurate prediction, but the more features the harder it is to re-scale features.[5]

Q13. The feature selection algorithms, such as decision tree, would keep irrelevant attributes in the dataset, which overfitting took a place. Those feature selection algorithms would first split the features which were highly related to the prediction. In the deeper decision tree, those irrelevant features would lead to overfitting. To avoid it, it would be better to do the feature selection in the preprocessing step. [6]

1.4 Chapter 8

Q5. In short, the No Free Lunch Theorem claims that there is no such one superlative optimization method would perform better than any other optimization methods in any domain. For example, a linear regression model could be trained by by gradient descent and also by normal equations. However, the performance of these two methods are depended on the dataset itself. The high accuracy and precision would lead the algorithms better to perform, and it also works on the other way around. [7]

Q6. Assuming there is an setting that we use a time series as an input. If using a feed-forward network, we would come across a problem that transform the time series into feed-forward would be an issue. We need to add inputs from the previous to input layer. Nevertheless, the network can't support to learn a function that receive inputs that happen before. Therefore, we need RNN, which can solve the problem due to the hidden units containing historical information from previous states, but the feed-forward network only uses local states. In this setting, we can expect RNN perform better than feed-forward network.

Q8. All the algorithms of the dynamical systems model could not guarantee to find the exact global optimum, especially in a large and complex environment. However, these algorithms could find a local optimum for sure. In somehow the local optimum is the exact global optimum, but most of time it is not. We cannot guarantee that the local optimum is the global optimum, but the global optimum is one of the local optimum.

1.5 Chapter 9

Q2. A reinforcement learning problem is to maximize the future cumulative rewards. If in this case we considered the active as our ultimate goal, the goal in the rl form should be that getting more rewards for being more active. Therefore, the expected reward r should be the probability of being good-shape as the time changed. It means if the person were more active on exercise, this person would have a better-shape.

Q4. The satisfaction of quantified self for Markov property is really depending on how to define the states. For example, running is non Markov property. When the runner running, the runner would feel the winds. But when the runner stopped, the winds stopped too. The winds were created by the runner, which is not a Markov property. Because the states of the winds did not depend on the winds, the winds stopped does not relative to the previous states of the winds but the runner.

2 Practical part

2.1 Chapter 5

Q1. After running the clustering algorithms using the crowdsignals dataset, more exactly we clustered the data measured using the Accelerometer we also wanted to see how the data measured a different sensor will behave. For this experiment we decided to use the data registered using the Gyroscope.

In Fig 1 we can observe the values obtained for the K-mean and Hierarchical silhouette scores for different k values. The right plots show the values obtained for the Accelerometer while the plots on the left depict the values for the Gyroscope sensor. If we look at the plots obtained using the K-mean algorithm we can see that there is a big difference between the two plots. For the Accelerometer data we can see that the silhouette score increases as k increases up until k is equal with 6, and after that it starts decreasing, this will result in 6 clusters. While if we observe the silhouette score obtained from the Gyroscope, we an see that the highest score is obtained for k is equal with 2 and after that the silhouette score starts slowly decreasing, this will result in 2 clusters.

If we look at the plots obtained using the Hierarchical algorithm we can see that the two datasets have completely different behaviours. The plot obtained using the Accelerometer data look similar to the one obtained using the K-mean algorithm, while the plot obtained using the Gyroscope data is completely different. More exactly almost all the silhouette values are negative, except the first value, which has a value of 0.012. One similarity between the two algorithms is that both of them will result in the same number of clusters as the ones obtained from K-mean

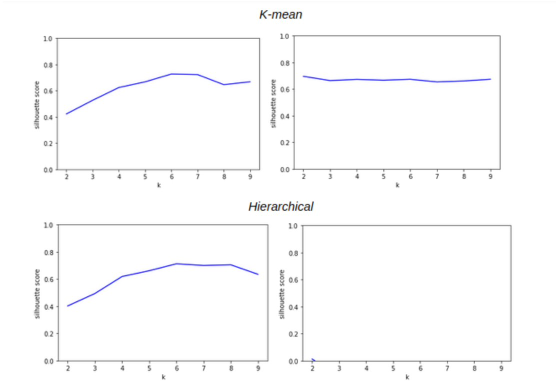

Fig. 1: K-mean and Hierarchical silhouette scores for different values for k. Accelerometer to the left and Gyroscope to the right

The last algorithm we are going to use for clustering the data is the k-medoids algorithm. In Fig 2 we can observe the plots and the clusters obtained using the k-medoids algorithm. The first observation that could be drawn here is that the plots for the silhouette score look similar with the one obtained using the k-mean algorithm. But there is a big difference, for the Gyroscope data, we obtained 5 clusters instead of 2. In the last two graphs we can observe the distribution of the clusters over a 3D space and how are they separated.

We can see that there are some similarities on how the data is clustered, mainly that there is almost the same number of clusters for both datasets, 6 for the Accelerometer and 5 for the Gyroscope. But this is only true when we use the k-medoids method, if we use the k-mean or the Hierarchical methods, the number of clusters for the Gyroscope data, drops to 2, which is completely different. Another observation would be the distribution of the data in the 3D space, mainly that the Accelerometer data has the shape of a torus, having most of the data alongside the edge of a circle, and almost no data in the center. While the Gyroscope data is much more concentrated in the middle and only has some extensions to the exterior, similar to a star.

Concerning how each cluster relates to the activity distribution, we could see that there are some big differences in there as well. Mainly that for the Accelerometer data the activities are distribute equally over each cluster, while for the Gyroscope data one cluster has the majority of the activities. To be more precise in cluster number 1 we have 6 activities that have a probability of over 90% to be situated in that cluster: *Table(99.64%), Sitting(97.92%), Washing Hands(89.84%), Standing(95.12%), Driving(98.08%), Eating(97.18%)*. One explanation why all those activities are in the same cluster could be that all of the require minimal movement, and most probably because of this, all of them are situated in the same cluster.

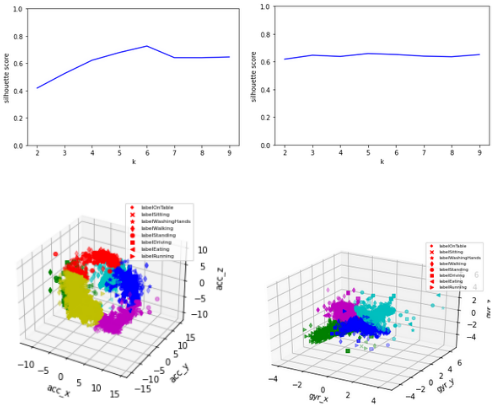

Fig. 2: k-medoids silhouette scores for different values for k and the obtained clusters. Accelerometer to the left and Gyroscope to the right

Q2. After analyzing the crowdsignals dataset we obtained some valuable insight in how the data will behave when we are using different methods for clustering. The next step is to analyze the data that we recorded using our own devices and see what kind of results we will obtain.

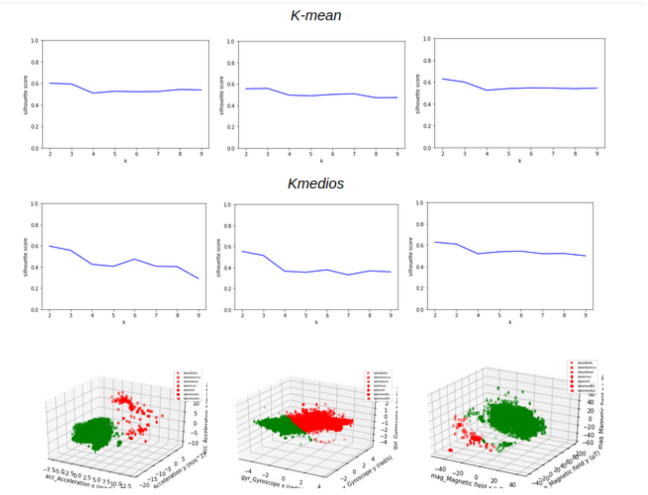

Fig. 3: k-mean and k-medoids silhouette scores for different values for k. Accelerometer to the left and Gyroscope in the middle and Magnetic to the right

For the examination of the data that was collected using our devices, we decided to focus on 3 fatures: Acceleration, Gyroscope and Magnetic. If we analyze Fig 3 we can observe that all of the selected features have the same behaviour, all the features have been separated in 2 cluster by the two algorithms that have been used (k-mean and k-medoids). All plots have the highest silhouette score at k=2, and after k=3 the score has a small drop and remains constant. The same observation could be made about the plots obtained using the k-medoids algorithm.

The last component to analyze is the clusters that have been obtained using the k-medoids algorithm. We can observe that the for the Accelerometer and the Magnetic sensor data, there is one big cluster of points that has been categorized as one cluster and the second cluster are the remaining points that have been scattered around the 3D space. While for the Gyroscope dataset the points are concentrated in one area, and one cluster is represented by half of the point and the other by the other half. In conclusion if we compare the clustering realised on the crowdsignals dataset and the one realised on our own dataset, we can see that there are some major differences between the obtained results.

2.2 Chapter 7

Q1. As we experiment with the dataset in case we would not throw two labels or unknown label cases out, we compare these two experiments. There are five comparative illustrations, the left plots derived from the experiment that filters unknown label cases, and the right plots are our experiment.

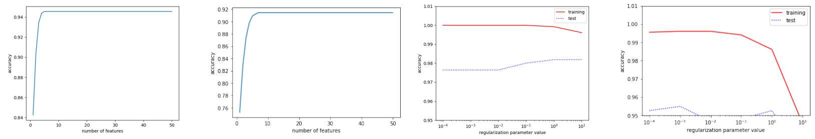

Fig. 4: number of features & regularization

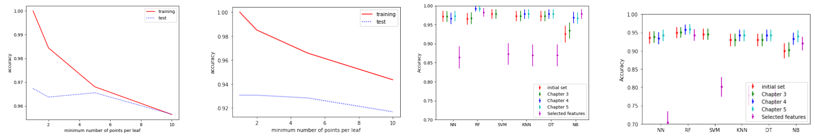

Fig. 5: points per leaf & generalization

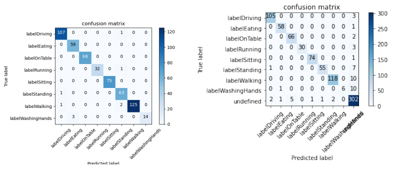

Fig. 6: confusion matrix

In figure 4, we observe that if we include undefined label cases, the highest accuracy of feature selection is only *0.92* when there are 10 selected features, which is slightly lower than *0.94* in normal experiment. The right plots show the influence of regularizer on objective functions, we can see there is a big difference. Both the training set and test set are below expectations, and the accuracy of the training set decreases dramatically when the value is 10, it reflects that the undefined cases indeed severely impact the performance.

In figure 5, we notice that with the growth of minimum number of points per leaf for a decision tree, the accuracy of the testing set is below *0.94* that is worsen than the original experiment. The right plots gives us a good indication of the generalizability of our models. Nevertheless, the performance of the neural network, KNN, and decision tree are worsened than the original experiment.

Finally, figure 6 gives us a comparison of confusion matrixes, we could notice the undefined labels have a great impact on prediction, such as undefined is classified as eating five times, washing hands is predicted as undefined ten times. Therefore, we conclude that the unrelated label cases have negative influence on performance.

Q3. By applying the learning algorithms to the dataset collected by ourselves, we wish to predict the activities and compare the performance with the crowdsignals dataset.

According to the left figure 7, we notice that the accuracy increases with several features, which is as same as the crowd signals dataset except that the accuracy of our own dataset is slightly higher than crowdsignals. The right plots show the effect of regularization parameter on objective functions, we observe a trend that the performance on the training set drops in our dataset when the value is 10ˆ1. Moreover, the accuracy of our test set is far below 0.95, which is worse than the crowdsignals dataset.

In figure 8, we can see that the accuracy of our test set decreases rapidly than the crowdsignals dataset when the minimum number of points per leaf increases. And in the right figure, the overall performance of models in our dataset is also worsened than crowdsigmal dataset, especially SVM algorithm performs pretty bad with Chapter 4 and 5 datasets.

Finally, in figure 9, we observe that the prediction of our dataset is pretty good, because a lot of high numbers on the diagonal in the confusion matrix, indicating high accuracy of our dataset.

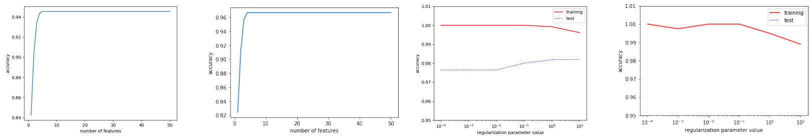

Fig. 7: number of features & regularization

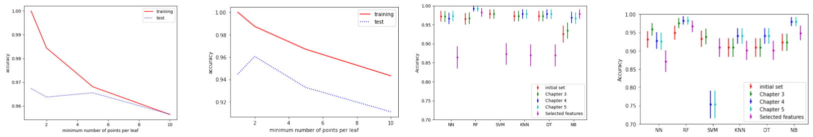

Fig. 8: points per leaf & generalization

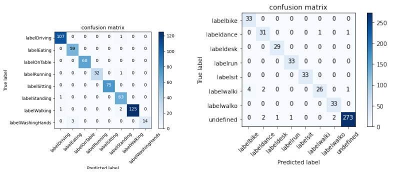

Fig. 9: confusion matrix

2.3 Chapter 8

Q3. For predicting the heart rate by developed a dynamical systems model, it could be considered as a regression problem. Thus, the *nsga_2* function should be used here. To setup for this algroithm, it needs to split the dataset for both train and test sets. Furthermore, the $nsga_2$ function passed the columns with ['self.acc_watch_x', 'self.acc_watch_y', 'self.hr_watch_rate']. This is because the heart rate is highly related to the acceleration of a watch. And for the equation, the *nsga_2* used simplest linear equations with heart rate * watch_x, heart rate * b and watch_y * b. The reason why used them is because of several random tests. And for sure, the target is heart rate itself. The result shows that whether the training or testing lines are all predicting well. The predicted values fit to the real values.

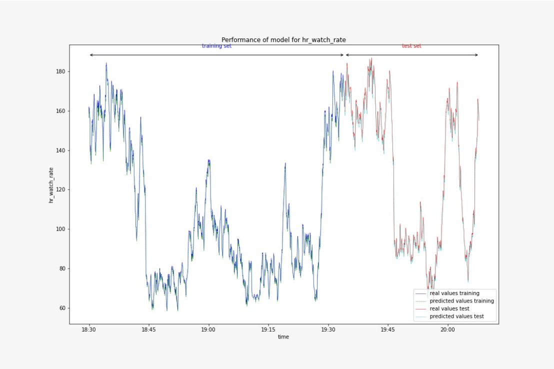

Fig. 10: Performance of model for hr_watch_rate

References

1. Raket, Lars Lau and Sommer, Stefan and Markussen, Bo, F.: A nonlinear mixed-effects model for simultaneous smoothing and registration of functional data. Pattern Recognition Letters **2**(5), 1–7 (2014)
2. Subspace clustering, https://towardsdatascience.com/subspace-clustering-7b884e8fff73. Last accessed Apr 7, 2019
3. The Number of Hidden Layers, https://www.heatonresearch.com/2017/06/01/hidden-layers.html. Last accessed Jun 1, 2017
4. Seven Most Popular SVM Kernels, https://dataaspirant.com/svm-kernels/. Last accessed Dec 17, 2020
5. Is kNN best for classification?, https://stats.stackexchange.com/questions/118268/is-knn-best-for-classification. Last accessed Oct 8, 2014
6. Feature Selection to Improve Accuracy and Decrease Training Time, https://machinelearningmastery.com/feature-selection-to-improve-accuracy-and-decrease-training-time/. Last accessed Aug 8, 2020
7. No Free Lunch Theorem for Machine Learning , https://chemicalstatistician.wordpress.com/2014/01/24/machine-learning-lesson-of-the-day-the-no-free-lunch-theorem/comment-page-1/. Last accessed Jan 24, 2014