Exam Machine Learning for the Quantified Self
29. 06. 2018
12:00 - 14:45

NOTES:

1. YOUR NAME MUST BE WRITTEN ON EACH SHEET IN CAPITALS.
2. Answer the questions in Dutch or English.
3. Points to be collected: 90, free gift: 10 points, maximum total: 100 points.
4. Grade: total number of points divided by 10.
5. This is a closed book exam (no materials are allowed).
6. You are allowed to use a SIMPLE calculator.

QUESTIONS

1. **Introduction (15 pt)**

   Cristiano is a sporty guy, and this is also his main occupation. In order to prepare well for an important tournament he is engaged in the quantified self. He tracks a variety of things, including all kinds of physiological measurements (such as heart rate and respiration), his movements (using the sensors of his mobile phone) as well as the training sessions he performs.

   (a) **(3 pt)** Choe et al distinguishes various purposes why someone would engage in the quantified self. Argue which purpose best fits Cristiano.

   "Improve health would be the most natural answer here (though others are accepted provided that a good rationale is provided). Cristiano tries to optimize his health state/execute a plan to get in the best possible shape."

   (b) **(4 pt)** Identify a supervised machine learning task and an unsupervised machine learning task that could be useful for the case of Cristiano.

   An example of a supervised learning task could be the prediction of his heart rate under different circumstances. An unsupervised task could be to find clusters of movements from the accelerometer data to see whether groups of activities can be found.

   (c) **(5 pt)** Explain for one of the two tasks you have identified above what the table **X** would look like (explain both the columns and the rows).

   For the prediction of the heart rate we could consider a number of different measurements (the columns), including the accelerometer data, the respiration, the training characteristics. The rows would be time points at which we perform measurements (i.e. examples in our dataset). The target is not part of X.

(d) **(3 pt)** Explain how we could apply reinforcement learning to the case of Cristiano.

"We could apply reinforcement learning to learn when to motivate Cristiano to really push his limits. For his, we could learn when to send motivating messages to him on his mobile phone such that he will in the end be in a better shape."

2. **Outlier detection (20 pt)**

Consider the data shown in Figure 1.

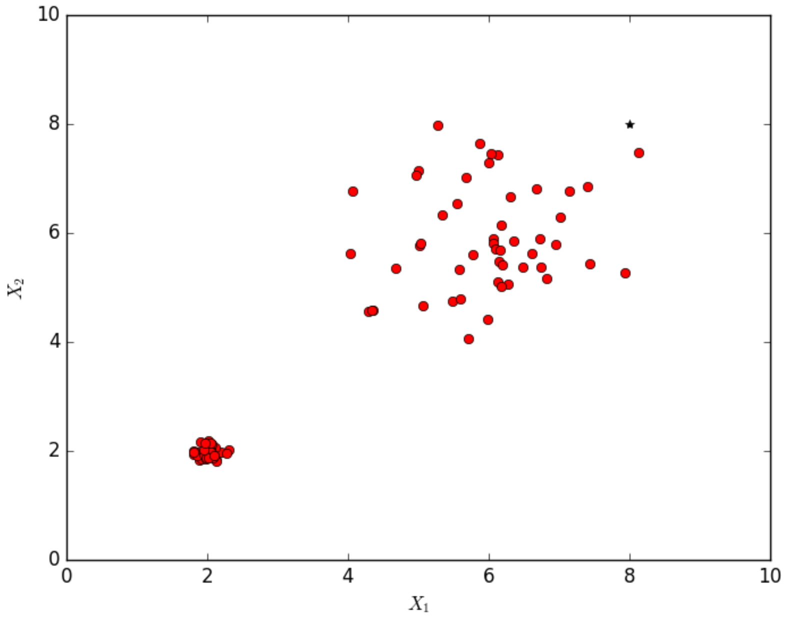

Figure 1: Example dataset

(a) **(4 pt)** We want to apply an outlier detection algorithm to this data. If we consider removing the outliers for the attribute X_1 alone, would you prefer to use Chauvenet's criterion or a mixture model? Argue your choice.

"The mixture model would be preferred since it seems to be very difficult to fit a single normal distribution on this data. Two normal distributions seem to fit quite well (one centred around 2 and one around 6. With a mixture model we could establish this."

(b) **(5 pt)** Explain the local outlier factor algorithm on a conceptual level.

Local outlier factor is a distance based outlier detection algorithm and considers the k closest neighbors around a point to determine whether it is an outlier. For those points it considers how far they are located from their closest neighbors and compares that to the distance of the current point to its neighbors. If the current point is much more distant from its neighbors compared to how distant its neighbors are to their neighbors the point is considered to be an outlier.

(c) **(4 pt)** Let us consider the point shown by means of the black star (at $X_1 = 8$ and $X_2 = 8$). Would it be more likely that this point would flagged as an outlier using the simple distance based outlier detection or using the local outlier factor? Argue why.

It is more likely to be flagged as outlier by the simple distance based outlier detection algorithm since it seems relatively far away from the other points (meaning that possibly to few points would fall in d_{min}). Local outlier factor would consider the fact that point are in general pretty distant from each other in that area.

(d) **(3 pt)** In outlier detection, the outlier detection algorithms have parameters to be set that eventually influence what is considered to be an outlier or not. Explain how appropriate parameter values can be found.

Through visual inspection, or by considering the number of points that are considered to be outliers.

(e) **(4 pt)** We want to apply a Principal Component Analysis to this data. Illustrate graphically what the principal component would look like. Argue why you have drawn it in that way.

This should be a diagonal line going from around $(0, 0)$ to $(8, 8)$. The reason for this is that it explain most variance in the data.

3. **Feature Enginering (15 pt)**

Consider the data shown in Table 1.

Table 1: Example dataset

Time point	*Heart rate*	*Intensity*	*Activity label*	*Tired*
0	60	low	sitting	no
1	60	low	sitting	no
2	70	low	walking	no
3	90	high	walking	yes
4	90	high	walking	yes

(a) (**3 pt**) Given this dataset, we are considering to aggregate the values of heart rate in the time domain by using the mean. A proposal is done to apply a window size of $\lambda = 3$. Given the size of the dataset shown, do you think this an appropriate choice? Argue why (not).

No, with such a window size only two datapoints would remain where we have a value for the feature from the time domain, that would be too limited.

(b) (**8 pt**) Apply the algorithm as proposed by Batal *et al.* to extract temporal features in the time domain on the combination of the features *Intensity* and *Activity label*. Consider a window size $\lambda = 1$ and a support threshold of $\Theta = 3/4$ (this is the minimum support needed). Explain what features result. Explain how you came to these features.

We start with the simple 1-patterns. Given the window size we see that:

- $Activitylabel = sitting$ ***has a support of 2/4***
- $Activitylabel = walking$ ***has a support of 3/4***
- $Intensity = low$ ***has a support of 3/4***
- $Intensity = high$ ***has a support of 2/4***

Hence, only $Activitylabel = walking$ ***and*** $Intensity = low$ ***meet the threshold. We can now make 2-patterns by combining these two patterns using the before (b) and co-occurs (c), for example we can consider*** $Activitylabel = walking(b)Intensity = low$ ***which has a support of 0/4. Note that all combinations can be made, also*** $Activitylabel = walking(b)Activitylabel = walking$***. Eventually, no 2-patterns meet the threshold.***

(c) (**4 pt**) Explain how we can derive temporal features in the time domain in case we have a combination of numerical and categorical features.

We create categories for the numerical features (e.g. normal, low, high, or increasing and decreasing) and apply the algorithm by Batal emphet al..

4. **Clustering (20 pt)**

We have collected heart rate data for two quantified selves, see Table 2. We are going to apply clustering to this data.

Table 2: Two datasets

Time point	*Heart rate*
person 1	
1	60
2	60
3	60
4	60
5	60
person 2	
1	100
2	120
3	80
4	60
5	60

(a) (**4 pt**) Imagine we use a raw-based approach to determine the distance between the two time series. First, let us consider the Euclidean distance by paring up each data point (i.e. we use time point 1 from person 1 and time point 1 from person 2, etc.). Calculate the distance between the two time series.

The answer is 120. We compute this as follows:

$$\sqrt{(60-100)^2 + (60-120)^2 + (60-80)^2 + (60-60)^2 + (60-60)^2 + (60-60)^2} = 74.8$$

(b) (**4 pt**) As an alternative we can apply Dynamic Time Warping (DTW). Explain what the boundary and monotonicity condition are in DTW.

The boundary condition states that the first and last time point of the two series should be matched while the monotonicity condition states that you can cannot move back in time when making a new pair.

(c) (**8 pt**) Let us now apply DTW. Fill in Table 3 (next page) by using the dynamic time warping algorithm. Use the absolute difference between the values as distance between two points. Show the steps you used in the calculations.

Table 3: answer table

person 2	t=5					
	t=4					
	t=3					
	t=2					
	t=1					
		t=1	t=2	t=3	t=4	t=5
		person 1				

The filled in table that results is shown below. Each position is calculated by considering the distance between the matched points and the cheapest path to get there. Note that you can only move up, to the right, or diagonal to the upper right.

Table 4: filled in answer table

person 2	t=5 (60)	120	120	120	120	120
	t=4 (60)	120	120	120	120	120
	t=3 (80)	120	120	120	140	160
	t=2 (120)	100	100	140	180	220
	t=1 (100)	40	80	120	160	200
		t=1 (60)	t=2 (60)	t=3 (60)	t=4 (60)	t=5 (60)
		person 1				

(d) (**4 pt**) Explain the difference between k-means and k-medoids clustering. Which of these two algorithms would be best to combine with DTW as distance metric? Explain why.

k-means clustering uses the average of the data points in a cluster as center, while k-medoids selects a real datapoint that acts as a center. Often k-medoids is preferred over k-means when focusing on the person level (for which you use DTW as distance metric) as having a real person as center is more insightful. Research has also shown that the overall performance is better when combing DTW with k-medoids.

5. **Supervised Learning (20 pt)**

This question concerns the several supervised learning algorithms as well as the theory underlying supervised learning. We assume a learning problem where sensory data ($p = 100$ features) has been collected for $N = 1000$ time points (i.e. learning instances) and we want to create a predictive model for the activity type (similar to our CrowdSignals data).

(a) **(3 pt)** We decide to apply a convolutional neural network with in total 2000 hidden neurons. We nicely split the data into a training and test set. Even though we vary the parameter settings of the network a lot (though not the number of neurons) we experience a lot of overfitting. Explain why this is not surprising. Support your answer by means of the theories that have been treated during the course.

The convolutional neural network is a very complex network that can represent very complex patterns. In order to train this complex network properly and make it generalizable (i.e. make sure the difference between the training and test error is small) a lot of data is required (think of the PAC learnability results). In this case we do not have sufficient data available.

(b) **(5 pt)** We want to select a number of features before we apply a machine learning problem to come to a model. Explain how forward selection works to select features.

You select a learning algorithm (e.g. a decision tree learning algorithm), and apply this learning algorithm on all possible single features. We select the feature for which the algorithm performs best. In the next iteration we combine that feature with another feature and select the set of two features that result in the best performance. This continues until we have a set containing all possible features. We then select the number of features at which the performance no longer seems to increase.

(c) **(3 pt)** Explain the concept of regularization and explain how it can be used to avoid overfitting.

Regularization is used to punish complex models. It is a term that can be added to the error function to keep models simple. Simpler models are less prone to overfitting.

(d) **(3 pt)** We are considering the application of one of the temporal learning algorithms that have been discussed in the book to the case at hand (i.e. predict the activity type, a categorical feature). Based on your knowledge of the algorithms, would it be better to apply a *time series algorithm* or a *recurrent neural network*? Argue why.

A recurrent neural network would be preferable as a time series algorithm is aimed at prediction of numerical values. Alternative answers are accepted given that the argumentation is valid.

(e) **(6 pt)** Another way to predict based on temporal patterns involves dynamical systems models. Provide three algorithms that have been treated during the course to learn the best parameter values for such models and briefly explain how they work.

- ***Simulated Annealing: make random moves in the parameter space, always accept moves that result in a lower error, and accept "bad" moves with a certain probability that reduces over time.***

- ***Simple GA: represent the parameter values as bit strings, create a random population of such bit strings and evolve the population based on the fitness of the individuals. Perform selection (using the fitness), crossover and mutation to move to the next generation.***
- ***NSGA-II: variant of the Simple GA which focuses on multiple objectives. Tries to find parameter vectors which provide a nice coverage of the Pareto front.***