Resit Machine Learning for the Quantified Self
09. 08. 2017
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)**

   Steve is a student with serious health problems. He is obese due to a lack of movement and regularly has mental health problems (depression, anxiety). He is pretty addicted to his mobile phone which is equipped with all imaginable sensors.

   (a) **(4 pt)** Provide the definition of the quantified self that has been discussed during the lecture.

   "The quantified self is any individual engaged in the self-tracking of any kind of biological, physical, behavioral, or environmental information. The self- tracking is driven by a certain goal of the individual with a desire to act upon the collected information."

   (b) **(5 pt)** Identify a supervised machine learning task and an unsupervised machine learning task that could be useful for the case of Steve.

   An example of a supervised learning task could be the prediction of a state of depression or anxiety based on the sensor values measured. An unsupervised task could be to find clusters of locations where Steve suffers from anxiety attacks.

   (c) **(3 pt)** List three measurements that could be collected for the case of Steve and argue their relevance based on one or both tasks you have identified above.

   Examples of measurements could be accelerometer data to measure how active Steve is (could help with prediction the state of depression), the heart rate (could predict an anxiety attack), or the GPS location (from clustering the locations for an anxiety attack)

   (d) **(3 pt)** Provide the definition for reinforcement learning that has been treated during the lecture.

   "Reinforcement learning tries to find optimal actions in a given situation so as to maximize a numerical reward that does not immediately come with the action but later in time."

2. **Feature Engineering (20 pt)**

Consider the data shown in Table 1.

Table 1: Example dataset

Time point	*Activity level (0-100)*	*Activity type*	*Speed*	*Tired*
0	5	sitting	0	no
1	80	running	10	no
2	80	running	9	yes
3	50	walking	5	no
4	80	running	9	no
5	50	walking	5	no
6	80	running	9	no
7	80	running	8	yes

(a) **(4 pt)** Apply a transformation in the time domain for the attribute *Activity level* using a mean aggregation function. Use a window size $\lambda = 1$. Provide the values for the newly created attribute.

Table 2 shows the answer.

Table 2: Dataset extended with temporal feature

Time point	*Activity level (0-100)*	*Activity level temp (0-100)*	*Activity type*	*Speed*	*Tired*
0	5	-	sitting	0	no
1	80	42.5	running	10	no
2	80	80	running	9	yes
3	50	65	walking	5	no
4	80	65	running	9	no
5	50	65	walking	5	no
6	80	65	running	9	no
7	80	80	running	8	yes

(b) **(6 pt)** Apply the algorithm proposed by Batal *et al.* with a window size $\lambda = 1$ and a minimum support of $\theta = \frac{2}{7}$ on the attribute *Activity type*. List what new attributes result and show your calculations.

We start with single attribute-value pairs (i.e. 1-patterns). There are three activity types: sitting, running, and walking. The support for sitting is $\frac{1}{7}$, for running it is $\frac{7}{7}$ and for walking $\frac{4}{7}$. Hence, Activity type = running and Activity type = walking are added as features as those are the only ones reaching the threshold. Furthermore, for patterns of size two we only consider the before (b) construct as we do not have any co-occurrence since

we just focus on a single feature, obviously we just look at combinations of selected 1-patterns:

- ***Activity type = running (b) Activity type = running support:*** $\frac{2}{7}$
- ***Activity type = walking (b) Activity type = walking support:*** $\frac{0}{7}$
- ***Activity type = running (b) Activity type = walking support:*** $\frac{2}{7}$
- ***Activity type = walking (b) Activity type = running support:*** $\frac{2}{7}$

As a result the three 2-patterns that have a support of at least $\frac{2}{7}$ ***that are added as well.***

(c) **(4 pt)** Next to the time domain, which other domain is available for aggregation of temporal features? Discuss how features are created in that domain.

The frequency domain. We apply a Fourier transformation to a selected historical window in our data (for each data point) and decompose the data within that window into frequencies with an accompanying amplitude. These form the basis to compute the features such as the power spectral entropy.

(d) **(6 pt)** Sometimes we have unstructured data. Describe the pipeline you can use to identify features from natural language.

The pipeline contains four steps: (1) tokenization (identify sentences and words); (2) lower case; (3) stemming (map words to their stam), and (4) stop word removal.

3. **Clustering (20 pt)**

We have collected activity data for two quantified selves, see Table 3. We are going to apply clustering to this data.

Table 3: Two datasets

Time point	*Activity data*
	person 1
1	60
2	60
3	80
4	80
5	60
	person 2
1	60
2	80
3	80
4	60
5	60

(a) (**3 pt**) We are going to cluster on a person level using this data. Explain what a feature-based distance metric is on this person level.

The feature-based distance metric tries to extract features from a set of measurements to summarize the measurements (e.g. the mean value). Distance between persons is defined by the difference in values for those features.

(b) (**5 pt**) Compute the distance using the Euclidean distance function on a person level (an example of a so-called raw-based person level distance metric). Show your calculations.

We just pair the instances up and compute the Euclidean distance. For this we square the distance per instance, sum them over all instances, and take the square root. This results in $\sqrt{0^2 + 20^2 + 0^2 + 20^2 + 0^2} = \sqrt{800} \approx 28.28$***.***

(c) (**8 pt**) Let us now use dynamic time warping as a distance metric. Fill in Table 4 by using the dynamic time warping algorithm. Use the absolute difference between the values as distance metric. Show the steps you used in the calculations.

Table 4: answer table

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

The filled in table that results is shown below (note that the order of the time points has been flipped for patient 2 to make it easier to apply the algorithm). 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 5: filled in answer table

person 2	t=5 (60)	40	40	40	40	0
	t=4 (60)	40	40	20	20	0
	t=3 (80)	40	40	0	0	20
	t=2 (80)	20	20	0	0	20
	t=1 (60)	0	0	20	40	40
		t=1 (60)	t=2 (60)	t=3 (80)	t=4 (80)	t=5 (60)
		person 1				

(d) (**4 pt**) Given the characteristics and features of the simple dataset above: would subspace clustering be appropriate to use with this dataset? Argue why (not).

Subspace clustering is intended for datasets with a large number of features/attributes. In the current dataset, only a single features is present, therefore this does not fit the characteristics of subspace clustering.

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

This question concerns the temporal machine learning algorithms that have been discussed during the lectures.

(a) **(4 pt)** Explain what stationarity means in the domain of time series.

"We call a time series stationary if (i) trends and periodic variations are removed and (ii) if the variance of the remaining residuals is constant over time. This means that both the expected mean of a time series as well as its variance are constant."

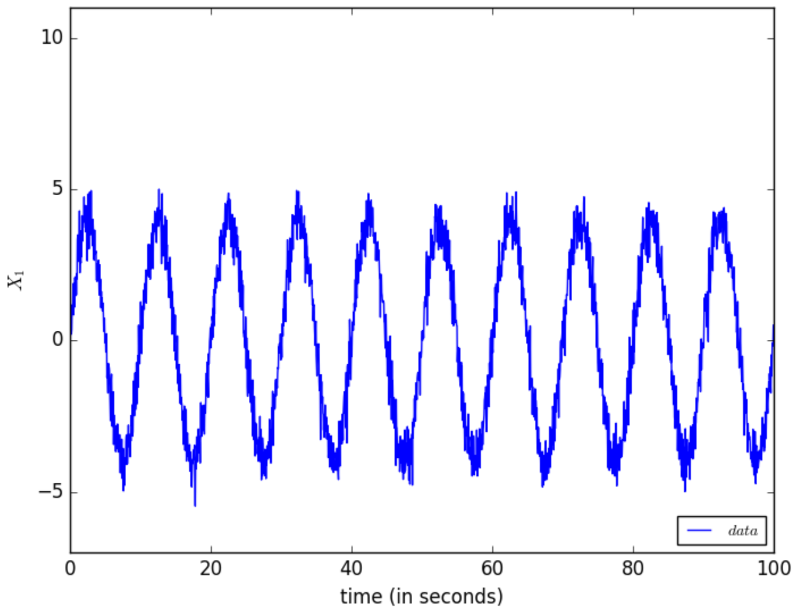

Figure 1: Example dataset

(b) **(4 pt)** Consider Figure 1, which shows a series of a measurement we are trying to predict. Which one of the three components that we normally break a time series down in would explain most of the pattern we see? Argue your choice.

The seasonality component as it seems that most of the values we measure are very periodic. Furthermore, we do not see an immediate trend nor huge other variations compared to this seasonality.

(c) **(5 pt)** We have collected data about three attributes, namely the x, y, and-z-axis of the accelerometer and aim to predict the current activity. We plan to use an echo state network for this purpose. Make a design for an echo state network for this problem and indicate the dimensions of the weight matrices.

The network is shown in Figure 2. It should contain three inputs, one output, and a reservoir which clearly contains a number of connected neurons. The dimensions of the weight matrices are shown in the figure as well.

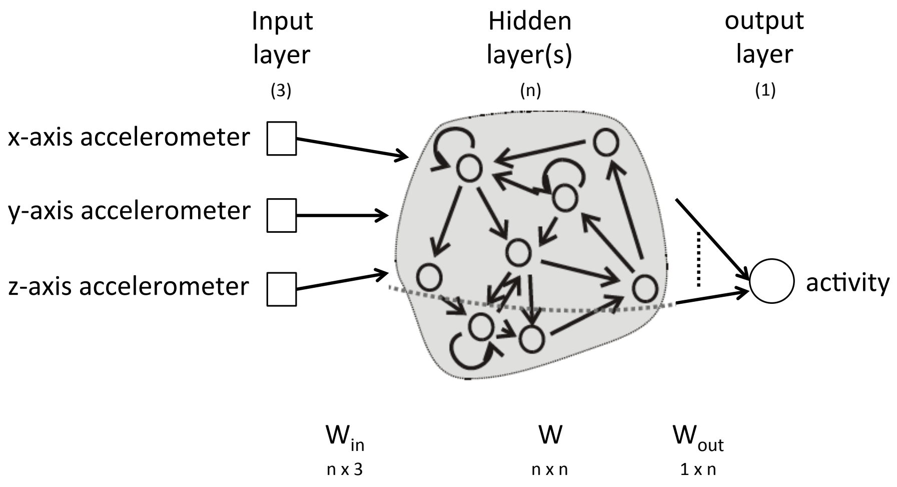

Figure 2: Example echo state network

(d) **(4 pt)** As an alternative to the echo state network, we also plan to use a regular recurrent neural network. Provide two differences between echo state networks and recurrent neural networks.

Some example differences are:

- ***within recurrent neural networks all weights are trained while in echo state networks only the weights to the output are trained.***
- ***echo state networks typically contain a lot more neurons compared to recurrent neural networks.***
- ***for recurrent neural networks we need to unfold the network to apply backpropagation, echo state networks are trained with a simpler approach.***

(e) **(3 pt)** Regular recurrent neural networks are known to suffer from several problems when learning time series. Provide one such problem and explain what causes this problem.

Regular recurrent neural networks have a problem with maintaining a long term memory. This is caused by the vanishing gradient as we propagate the error back through the network.

5. **Reinforcement Learning (15 pt)**

We are going to apply reinforcement learning to support a user in becoming more active. We measure the activity level and activity type of a person and want to provide suggestions to that person based on his measured state (examples of advices could be: do activity x, stop activity y, etc.).

(a) **(3 pt)** Explain what the Markov Property means (you can relate your explanation to this specific example or you can also explain it in general if you want).

The Markov property states that the probability of ending up in a next state with a certain reward given a current state and action can be determined

solely by the current state and action. There is no need to consider a history before the current state and action. Alternatively, you can say that the probability is equal to the case where you do consider the entire history.

(b) **(4 pt)** Explain how the one step Q-learning algorithm works.

Q-learning maintains so-called Q-values for state-action pairs. These indicate the expected reward for selecting an action in a given state. Actions are selected based on a certain selection approach. It updates the value of the state-action pair of the selected action (and the given state) by considering the reward obtained in the next state combined with the expected future reward based on selecting the action in the next state which maximizes the future reward.

(c) **(4 pt)** Some of the measurements we perform are continuous (specifically, the activity level is), would this be a problem for SARSA or Q-learning? Argue why (not).

Yes, this would be a problem as they maintain expected reward values for all state-action pairs (in their most rudimental form) and the number of states is infinite when defined based on the continuous measurement. Hence, we cannot store these. A solution (does not have to be part of your answer) is to use a dedicated approach that discretizes the continuous state space.

(d) **(4 pt)** We have the choice to either apply an ϵ-greedy approach or a softmax approach to select the actions. We know that the person we are supporting does not change at all in terms of responses to messages. Which one of the two approaches would be most suitable to use? Argue your choice.

Given that the user does not change its preferences, there is only a need to performing exploration in the beginning to find the most suitable messages and, once found, we can just perform exploitation without any need for exploration again. The ϵ-greedy approach does not change how much it explores or exploits, while the softmax does reduce the amount of exploration that takes place over time. Hence, the latter would be most suitable.