# Exam Machine Learning for the Quantified Self
19. 07. 2019
12:00 - 14:45

NOTES:

1. YOUR NAME MUST BE WRITTEN ON EACH SHEET IN CAPITALS.
2. Answer the questions in Dutch or English (English is preferred).
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)**

   (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" (4 pts).***

   Let us consider a specific application of the quantified self. We want to improve the adherence of medication intake. In order to do so, we build an app that learns how to coach a user in such a way that in the long run adherence is high and stable.

   (b) **(4 pt)** When considering the different types of learning problems in machine learning, what type would you consider the learning problem described above? Argue why.
   ***A reinforcement learning task since it is not immediately clear whether the coaching actions are suitable, only in the long run will these be clear (2 pts for right task, 2 pts for the explanation, also other types of learning problems as answer are acceptable provided that the right rational is provided).***

   (c) **(4 pt)** In the development of the aforementioned app, we collect a lot of data from different sensors. These are collected at very different sampling rates. Explain in a step-by-step way how we can create a suitable dataset from such data in order to apply machine learning.
   ***We consider a certain step size ($\Delta t$) and start with the time stamp of the earliest measurement. We continuously add the step size until we reach the last time point in the dataset. We create a table with the rows being***

***each distinct interval identified in this process and the columns being the types of measurements we do (i.e. sensors). For each of the intervals, we look up the relevant values for each sensor. In case there are none, the value of the column in the row is missing. In case there are multiple we use a function to combine the values into a single value (4 pts for the explanation).***

(d) **(3 pt)** Explain two advantages and one disadvantage of considering measurements at a lower level of granularity compared to a high granularity.

***Examples of advantages are:***

- ***Computation time***
- ***Less missing values***
- ***Less prone to noise***

***Examples of disadvantages are:***

- ***Might filter out interesting parts in the data***
- ***Less instances to learn from***

***(1+1 pts for correct advantages, 1 pt for correct disadvantage)***

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

(a) (**3 pt**) Provide the definition of an outlier as was discussed during the course.

***An outlier is an observation point that is distant from other observations (3 pts).***

(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.***

Let us consider one sensor $X_1$ for which we measure the following values:
$[1, 1, 2, 2, 3, \mathbf{10}, 20, 20, 21, 22, 22, 23, 34]$
Let us focus on the measured value **10**.

(c) (**9 pt**) For each the following three algorithms argue whether the value **10** would be considered an outlier given the measured values: *Chauvenet's criterion*; A *mixture model* with $K = 2$, and a *simple distance based* approach with $d_{min} = 5$ and $f_{min} = 0.9$. Provide an argumentation for each of the three answers you gave.

- ***Chavenet's criterion: not an outlier, a normal distribution will be fitted using the mean and standard deviation of the measured values. This means that 10 is very close to the mean, hence not an outlier.***

- ***Mixture model with $K = 2$: an outlier, two distributions will be fitted, one focusing on the low values, and one on the high values, hence, the probability of observing 10 will be very low according to those two distributions.***
- ***Simple distance based approach: an outlier, none of the points are within $d_{min} = 5$, and hence, the criterion $f_{min} = 0.9$ is exceeded.***

***(1 pt for correct answer per outlier detection algorithm, 2 pts per explanation).***

(d) (**3 pt**) We suffer from high frequency periodic noise in our dataset. What technique could we use to remove this noise? Argue you choice.

***A lowpass filter, this will filter out the high frequency period noise by means of a transfer function (1 pt for the name, 2 pts for the explanation).***

3. **Clustering (20 pt)**

We have collected heart rate data for two quantified selves namely Ali and Steven, see Table 1. We are going to apply clustering to this data.

Table 1: Two datasets

| *Time point* | *Heart rate* |
|---|---|
| *Ali* | |
| 1 | 80 |
| 2 | 80 |
| 3 | 100 |
| 4 | 80 |
| 5 | 80 |
| *Steven* | |
| 1 | 80 |
| 2 | 100 |
| 3 | 80 |
| 4 | 80 |
| 5 | 100 |

(a) (**4 pt**) Imagine we use a feature based approach to compute the distance between the time series using the mean. Compute the distance between the two time series using the Euclidean distance.

***We first take the mean of both series. For Ali this is 84, for Steven this is 88. We then compute the Euclidean distance:***
$\sqrt{(84-88)^2} = 4$ ***(2 pts for the step of computing the mean and 2 pts for the correct application of the Euclidean distance).***

(b) (**4 pt**) Next to the Euclidean distance, we can also use Gower's similarity as a distance metric. Explain how Gower's similarity is defined.

***In Gower's similarity, for different types of variables, different similarity scores are defined. For dichotomous, categorical, and numerical values.***

***Hereby, for dichotomous a value of 1 is assigned in case both are present (otherwise 0), for categorical values, a value of 1 is assigned when the category of both is the same (and otherwise 0) and for numerical 1 minus the absolute difference is taken, whereby the difference is normalized based on the range. Similarities between values are only computed when both are known. Then the similarities that have been computed are summed and divided by the number of comparisons that could be made.***

(c) (**8 pt**) As an alternative, we can apply Dynamic Time Warping (DTW). Fill in Table 2 (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 2: answer table

| | | | | | | |
|---|---|---|---|---|---|---|
| *Steven* | t=5 | | | | | |
| | t=4 | | | | | |
| | t=3 | | | | | |
| | t=2 | | | | | |
| | t=1 | | | | | |
| | | t=1 | t=2 | t=3 | t=4 | t=5 |
| | | *Ali* | | | | |

***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 (1 pt deduction per wrong value in table).***

Table 3: filled in answer table

| | | | | | | |
|---|---|---|---|---|---|---|
| *Steven* | t=5 (100) | 40 | 40 | 20 | 20 | 20 |
| | t=4 (80) | 20 | 20 | 40 | 0 | 0 |
| | t=3 (80) | 20 | 20 | 20 | 0 | 0 |
| | t=2 (100) | 20 | 20 | 0 | 20 | 40 |
| | t=1 (80) | 0 | 0 | 20 | 20 | 20 |
| | | t=1 (80) | t=2 (80) | t=3 (100) | t=4 (80) | t=5 (80) |
| | | *Ali* | | | | |

(d) (**4 pt**) Explain the difference between agglomerative and divisive clustering. Furthermore, explain in this context what a dendrogram is.

***In agglomerative clustering, each point starts in its own cluster, and clusters are merged based on some similarity metric. In divisive clustering, all points are put into a single cluster and then divided into smaller clusters. The clusters to be split are selected based on some metric, as are the points to be moved out of the cluster. A dendrogram shows the results of clustering on different levels, going from one cluster at the top to the most refined clusters at the bottom. This allows one to select the level of***

***clustering which is appropriate for the domain (2 pts for the difference, 2pts for the explanation of the dendrogram).***

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

In this question, we are going to focus on supervised learning based on Quantified Self data. We want to apply a supervised learning approach on sensory data (specifically accelerometer data) to recognize activities. We consider both algorithms that take the notion of time into account and those that do not.

(a) (**4 pt**) Explain for which of the two (machine learning algorithms that take time into account explicitly or not) the temporal feature engineering step is most important.

***The algorithms which take time into account can learn to recognize patterns over time themselves while the other class of algorithms are unaware of this. Hence, the feature engineering is more important for the learning algorithms that do not take time into account explicitly (2 pt for correct answer, 2 pts for explanation, other answers are accepted provided that the right arguments are given).***

(b) (**4 pt**) Name the two domains we can identify temporal features in, and explain the difference between the two domains.

***The time and the frequency domain. The time domain summarizes values values within a historical window by considering the measure values and applying some aggregation function over it (e.g. mean, slope). The frequency domain considers the periodicity in the historical values and decomposes the signal into sinusoid functions with different frequencies, each having their own amplitude (1+1 pts for correct terms, 2 pts for difference).***

(c) (**2 pt**) When considering temporal features using a window combined with non-temporal machine learning approaches, we mostly remove instances of which the windows overlap too much (e.g. we allow for a maximum of 50% overlapping windows). Give one advantage and one disadvantage of allowing more overlap between windows.

***An example advantage is that less instance will be thrown away, an example disadvantage is that more overlap means that the learning algorithm is more prone to overfitting on the specific data (1 pt for advantage, 1 pt for disadvantage).***

(d) (**3 pt**) We need to decide on the algorithm to use. Explain based on the supervised learning theory how the size of the dataset and the complexity of the hypothesis space resulting from the choice of our machine learning algorithm are related.

***PAC learnability allows one to give some form of guarantees on the difference between the error on the training and the test set. Only when the difference is small enough is the problem learnable. The more complex the hypothesis space, the more data is needed to guarantee this difference is sufficiently small (3 pts for explanation).***

(e) (**4 pt**) List two approaches that can be used to avoid overfitting of the data. For both approaches, explain how these approaches contribute to less overfitted models.

***Examples are:***

- ***regularization, punishes more complex models using a regularization term during the learning process (e.g. supresses high weights)***
- ***hyperparameter settings, you can select the hyperparameters of the approaches such that they avoid overfitting for instance by making less complex models (e.g. a lower max tree depth in decision trees)***

***Other answers are also accepted (1+1 pts for the correct term, 1+1 pts for explanation).***

(f) (**3 pt**) What is the main disadvantage of regular recurrent neural networks when it comes to learning from temporal data? What is the cause of this problem?

***They have trouble learning long term dependencies due to the vanishing gradients problem (1 pt for the correct answers, 2 pts for the explanation).***

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

We are going to focus on a reinforcement learning case. We focus on a case where we want to support people that are hopelessly out of shape to regain their shape again by sending them messages to coach them in their work out endeavors. The MDP which includes the states, actions, and rewards is shown in Figure 1. We see two actions, namely an advice to *work out* and an advise to *remember: start slow*.

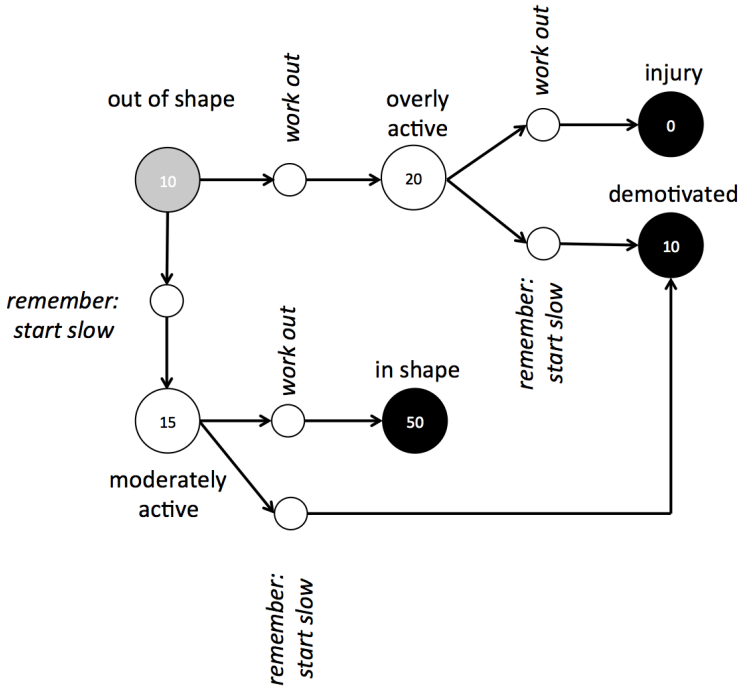


Figure 1: MDP example

(a) **(5 pt)** Assume that we use $\gamma = 1$ to compute the value function. Compute the final Q-values for each state-action pair for the non-terminal states (i.e. you do not have to compute the values for the black nodes). Explain how you came to your answer.

***For each state-value pair we compute the value of the action, meaning we sum up all rewards we obtain in the future, assuming that we take the action with the highest future reward. We assume $\gamma = 1$. We can then see that:***

- ***Q(out_of_shape, work_out) = 30***
- ***Q(out_of_shape, remember:_start_slow) = 65***
- ***Q(moderately_active, work_out) = 50***
- ***Q(moderately_active, remember:_start_slow) = 10***
- ***Q(overly_active, work_out) = 0***
- ***Q(overly_active, remember:_start_slow) = 10***

(b) **(4 pt)** Explain the difference between the two reinforcement learning algorithms *Q-learning* and *SARSA*.

***In Q-learning, an action is selected using a certain action selection approach. To determine the value of a selected action, the value of the resulting state is taken, plus the value of the action with the highest value from that resulting state (while our action selection approach does not necessarily take that action with the highest reward). This is called off-policy. For SARSA, the same action selection selection mechanism is used, also to compute the action to select in the next state. This is called on-policy.***

(c) **(3 pt)** Explain the concept of eligibility traces.

***Eligibility traces can be used to put credit on states and actions that have been seen more often in the past. Using these concepts, the estimated values of state-actions pairs are updated more rigorously when these are seen more often in the past.***

(d) **(3 pt)** What is the purpose of the U-tree algorithm? Explain how the algorithm works.

***The U-tree algorithm can be used to discretize the action space in case of a continuous actions space. It considers all observed values for the features included in the state and orders them per feature. It tries to look at all possible splits in this ordered list and checks whether the rewards accompanying the state before and after the split are significantly different using a Kolmogorov Smirnov test. The split which is most different (provided that the p-value is lower than 0.05) is taken and used to create a tree to discretize the state space. This process continues as more observations are performed.***