Exercises for Foundations of Data Science

University of Zurich[UZH]

Prof. Dan Olteanu,
Dr. Ahmet Kara, Dr. Nils Vortmeier,
Haozhe Zhang

DaST Data•(Systems+Theory)

Fall 2020/2021 | Sheet 2 | 25.09.2020

- The solutions will be discussed on Friday 09.10.2020, 14:00-15:45 on Zoom.
- Videos with solutions will be posted on OLAT after the exercise session.

Exercise 2.1 [Logical Gates Using Perceptron]
As introduced in the lecture, a perceptron with input features $x_1, \ldots, x_D$, weights $w_1, \ldots, w_D$, and bias w_0 outputs the value:

$$y = \begin{cases} 1, & \text{if } w_0 + \sum_{i=1}^{D} w_i x_i \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

(a) We consider perceptrons with at most two inputs. The inputs x_i take binary values, i.e., $x_i \in \{0, 1\}$. Construct three perceptrons that simulate an AND, an OR, and a NOT gate.

(b) Can you construct a perceptron simulating an XOR (exclusive OR) gate? If not, give reasons. Keep in mind that a perceptron can simulate only linearly separable functions.

(c) Construct a (multi-layer) perceptron simulating a SAME gate by combining the perceptrons constructed under (a). The truth table for SAME is as follows:

x_1	x_2	x_1 SAME x_2
0	0	1
0	1	0
1	0	0
1	1	1

(d) Instead of using a hard threshold we would like to use a continuous approximation. The hyperbolic tangent function is defined as $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. We consider perceptrons where the output of neurons is defined as follows:

$$y = \begin{cases} 1, & \text{if } \tanh\left(w_0 + \sum_{i=1}^{D} w_i x_i\right) > 0.99 \\ -1, & \text{if } \tanh\left(w_0 + \sum_{i=1}^{D} w_i x_i\right) < -0.99 \end{cases}$$

Show that similar constructions to the ones under (a) can still be used to construct logical gates.

Exercise 2.2 [Nearest Neighbour Classification]
We consider another approach to classify data: the nearest neighbour classifier (NN). Suppose we are given a natural number k and N data points $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $y_i \in \{0, 1\}$ are the labels of the points. Given a new point $\mathbf{x}'$, the k-NN approach does the following: find (according to some norm) the closest points $\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_k}$ to $\mathbf{x}'$ and output y' as the majority label from the set $\{y_{j_1}, \ldots, y_{j_k}\}$, i.e., the most commonly occurring label among the k-nearest neighbours.

(a) What advantage does the k-NN approach offer over a linear classifier like the perceptron?

(b) For D-dimensional data, the perceptron uses $D + 1$ parameters. How many parameters does the nearest neighbour model have? How much memory do you need to store the model? What is the computational cost of predicting the label y'?

Exercise 2.3 [Linear Regression and Basis Expansion]
Suppose we are given the following dataset $\mathbf{D}$ where the X-values are the input and the Y-values are the output values.

X	2	3	4	5	6	7
Y	5	6	5	9	7	10

(a) Assume we model the data using the following linear relationship:

$$y_i = w_0 + w_1 x_i, \quad \text{for } i \in \{1, \ldots, 6\},$$

where x_i and y_i are the X- and Y-values in the i^{th} column of the above table. Use the least squares estimator to determine $\mathbf{w} = [w_0, w_1]^\mathsf{T}$. Plot the data and the model in a two-dimensional coordinate system. Visualize the differences between the observed data and the predictions of the model. You do not need to do matrix arithmetic and plotting manually. You could use a programming language like Python.

(b) Now, assume that we model the data using a non-linear relationship:

$$y_i = w_0 + w_1 x_i + w_2 x_i^2, \quad \text{for } i \in \{1, \ldots, 6\}.$$

Again, use the least squares estimator to determine $\mathbf{w} = [w_0, w_1, w_2]^\mathsf{T}$. Plot the data and the model in a two-dimensional coordinate system. Visualize the differences between the observed data and the predictions of the model.

(c) Which of the two above models is better on the dataset $\mathbf{D}$? Answer this question by considering the mean squared errors for both models. How could we design an even better model?

(d) Consider polynomial regression models of higher degrees for the dataset $\mathbf{D}$. Which model minimizes the mean squared error? Explain the implications of such a model.

Exercise 2.4 [Predicting Commute Times]
In the third lecture, we discussed about a linear regression model that predicts the commute time into the city centre given the distance to the city centre and the information whether it is weekend or weekday. Let us consider some extensions of that model.

(a) Assume we would like to distinguish not only between weekday and weekend but between all seven days of the week. How would you incorporate this distinction into the model?

(b) Assume now we want to make suggestions such as use the bus or the car. How would you extend the model to make such suggestions possible? How can you use your model to suggest bus or car?

(c) Now consider we have refined information on the commute time within each of the time frames 09:00 -12:00, 12:00 -15:00, 15:00 -18:00, and 18:00 -21:00 per day. How can we use this information to suggest which time frame within a given day would have the least commute time by bus or car?

Exercise 2.5 [Predicting Jogging Times]
Alice has recently started to jog in the evenings. She records her daily jogging times to see how the jogging time of a day is affected by the jogging times of the previous days. We want to help Alice by writing possible linear prediction models. Suppose that $x_1, x_2, \ldots$ denote the jogging times on day 1, day 2, and so on. When formulating the linear models, add a bias term if necessary.

(a) Write a linear model to predict x_{t+1} using the jogging times on the two previous days, i.e., x_t and x_{t-1}.

(b) Let $\triangle_{t+1} := x_{t+1} - x_t$ denote the change in the jogging time from day t to day $t+1$. Write a linear model to predict $\triangle_{t+1}$ using x_t and x_{t-1}.

(c) Write a linear model to predict $\triangle_{t+1}$ using $\triangle_t$.

(d) Write a linear model to predict $\triangle_{t+1}$ using $\triangle_t$ and x_t.

(e) Assume that Alice has recorded the jogging times only for T consecutive days. Explain how she should train the model under (d) using her data.

Exercise 2.6 [The Huber Loss in a Linear Regression Setting]
Given arbitrary but fixed parameters $\lambda, \mu \in \mathbb{R}$ with $\lambda, \mu > 0$, the Huber loss is given by the function $h_{\lambda,\mu} : \mathbb{R} \mapsto \mathbb{R}$ such that

$$h_{\lambda,\mu}(z) = \begin{cases} \lambda\big(|z| - \frac{\lambda}{4\mu}\big), & \text{if } |z| \geq \frac{\lambda}{2\mu} \\ \mu z^2, & \text{otherwise} \end{cases}$$

Consider also the absolute loss function $f(z) = |z|$ and the square loss function $g(z) = z^2$.
Given a training dataset $\mathbf{D}$ consisting of the data points $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$, we define the loss function (by ignoring the noise term)

$$\mathcal{L}(\mathbf{w}, \mathbf{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i - \mathbf{w}^\mathsf{T} \cdot \mathbf{x}_i),$$

where $\mathbf{w} \in \mathbb{R}^D$ is the vector of model parameters and ℓ is a placeholder for one of the functions $h_{\lambda,\mu}$, f, and g.

(a) Draw three graphs plotting the functions $h_{4,1}$, f, and g. What can you say about the differentiability of these functions?

(b) What can you say about the influence of outliers to $\mathcal{L}$ when we substitute ℓ in $\mathcal{L}$ by each of the functions $h_{4,1}$, f, and g?

(c) Compute $\nabla_{\mathbf{w}} \mathcal{L}$ for the case that ℓ in $\mathcal{L}$ is substituted by $h_{\lambda,\mu}$ for any $\lambda, \mu > 0$.

Hint: When dealing with absolute values, the sign function defined as follows is often helpful:

$$\text{sign}(z) = \begin{cases} 1, & \text{if } z > 0 \\ -1, & \text{otherwise} \end{cases}$$