Exercises for Foundations of Data Science

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

Fall 2020/2021 | Sheet 5 | 06.11.2020

- The solutions will be discussed on Friday 20.11.2020, 14:00-15:45 on Zoom.
- Videos with solutions will be posted on OLAT after the exercise session.

Exercise 5.1 [Logistic Regression]

(a) Recall the expression

$$\mathrm{NLL}(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = -\sum_{i=1}^{N}(y_i \log \sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i) + (1-y_i)\log(1-\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i))),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$, for the negative log-likelihood of observing the class labels $\mathbf{y}$ given the input $\mathbf{X}$ and the parameters $\mathbf{w}$ of a logistic regression model.

We write μ_i for the expression $\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i)$. Verify the equations

$$\nabla_\mathbf{w}\mathrm{NLL}(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \sum_{i=1}^{N}\mathbf{x}_i(\mu_i - y_i) = \mathbf{X}^\mathsf{T}(\boldsymbol{\mu} - \mathbf{y})$$
$$\mathbf{H} = \mathbf{X}^\mathsf{T}\mathbf{S}\mathbf{X}$$

for the gradient $\nabla_\mathbf{w}\mathrm{NLL}(\mathbf{y} \mid \mathbf{X}, \mathbf{w})$ and the Hessian $\mathbf{H}$ given in the lecture, where $\mathbf{S}$ is the diagonal matrix with $S_{ii} = \mu_i(1-\mu_i)$.

For this, recall from Exercise Sheet 1 that the derivative $\sigma'(z)$ of $\sigma(z)$ is given by $\sigma'(z) = \sigma(z)(1-\sigma(z))$.

Also show that the Hessian is positive semi-definite.

(b) Suppose we are given the following dataset $\mathcal{D}$ of 6 observations over features x_1 and x_2 with class label y.

x_1	x_2	y
8	2	1
5	5	1
7	7	1
9	8	0
3	8	0
4	5	0

Use Newton's method to estimate the parameters $\mathbf{w}$ of a logistic regression model for this data. For this, start with $\mathbf{w}_0 = [0, 0, 0]^\mathsf{T}$.

If you use Python or some other programming language: plot the data and the decision boundary in a two-dimensional coordinate system after every iteration. After how many iteration does the computation converge?

Hint: You should need less than 10 iterations.

(c) The dataset $\mathcal{D}$ from part (b) is not linearly separable, but the following slightly modified dataset is:

x_1	x_2	y
8	2	1
5	5	1
7	7	1
7	8	0
3	8	0
4	5	0

What happens if you use Newton's method on this dataset? How could the occurring problem be circumvented?

Hint: Suppose a logistic regression model with parameters $\mathbf{w}$ can correctly classify all observations of our dataset. What can you say for the parameter vector $\delta\mathbf{w}$, for an arbitrary $\delta > 1$?

(d) We obtain the optimal parameters $\mathbf{w}$ of a logistic regression model by minimizing the negative log-likelihood $\mathrm{NLL}(\mathbf{y} \mid \mathbf{X}, \mathbf{w})$. We could in principle also obtain the parameters of a linear classification model by minimizing the mean squared error

$$\mathrm{MSE}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(\sigma(\mathbf{w}^\mathsf{T}\mathbf{x}_i) - y_i)^2.$$

Discuss whether this is a good idea.

Exercise 5.2 [Logistic Regression vs. Naïve Bayes]

Logistic Regression is a discriminative model, and Naïve Bayes is a generative model. However, they are closely related. More precisely: if we fix some assumptions of a Naïve Bayes classifier, then the resulting model is equivalent to a logistic regression model.

The aim of this exercise is to prove this for a special case. Assume that our data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ is binary, so, $\mathbf{x}_i \in \{0,1\}^D$ for some D, and $y_i \in \{0,1\}$, for all i. Further, assume that $p(y = 1) = \pi$ and that $p(x_i = 1 \mid y = j) = \theta_{i,j}$, for some values $\pi, \theta_{i,j} \in [0,1]$.

Show that the Naïve Bayes conditional distribution

$$p_{\mathrm{NB}}(y = 1 \mid \mathbf{x}, \pi, \boldsymbol{\theta}) = \frac{p(y = 1 \mid \pi)p(\mathbf{x} \mid y = 1, \boldsymbol{\theta})}{\sum_{i=0}^{1} p(y = i \mid \pi)p(\mathbf{x} \mid y = i, \boldsymbol{\theta})}$$

can be translated into a Logistic Regression conditional distribution of the form

$$p_{\mathrm{LR}}(y = 1 \mid \mathbf{x}, \mathbf{w}, w_0) = \sigma(w_0 + \mathbf{w}^\mathsf{T}\mathbf{x}).$$

Hint: Start by dividing both numerator and denominator by the numerator. How can you introduce an exponential function?

Exercise 5.3 [Support Vector Machines I]

(a) Let us look at support vector machines (without kernels) and assume that the data is linearly separable. In order to maximize the margin, a more natural formulation would be the following: Fix $||\mathbf{w}||_2 = 1$, so the distance of $\mathbf{x}$ from the hyperplane defined by $(\mathbf{w}, w_0)$ is exactly $|\mathbf{x}\cdot\mathbf{w}+w_0|$. Then, we can define the optimization problem:

$$\begin{aligned} \text{maximize} \quad & \alpha \\ \text{subject to} \quad & y_i(\mathbf{x}_i \cdot \mathbf{w} + w_0) \geq \alpha \quad \text{for } i = 1, \ldots, N \\ & ||\mathbf{w}||_2 = 1 \end{aligned}$$

Unfortunately, the condition $||\mathbf{w}||_2 = 1$ implies that the set of admissible $\mathbf{w}$ do not form a convex set. Argue that relaxing the constraint to be $||\mathbf{w}||_2 \leq 1$ does not change the optimal solution of the above program. Then show that this formulation is equivalent to the one we considered in the lectures:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to} \quad & y_i(\mathbf{x}_i \cdot \mathbf{w} + w_0) \geq 1 \quad \text{for } i = 1, \ldots, N \end{aligned}$$

So, show that an optimal solution for one optimization problem can be used to obtain an optimal solution for the other one.

(b) Suppose we use the SVM formulation for separable data, and that the data indeed is linearly separable. Recall that in this case, support vectors are those points $\mathbf{x}_i$ in the dataset for which $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + w_0^*) = 1$, where $\mathbf{w}^*, w_0^*$ is the max-margin hyperplane. If your dataset consists of N points in a D-dimensional space, what is the maximum number of support vectors possible? What is the minimum number?

(c) Suppose you use the primal SVM formulation for the non-separable case, i.e., with slack variables ζ_i, but your data is actually linearly separable. Do you always recover the "true" max-margin separating hyperplane?

(d) Given a training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, prove that in the primal SVM formulation the sum of slacks $\sum_{1\leq i\leq N} \zeta_i$ of an optimal solution in the non-separable case gives an upper bound on the number of misclassified training examples.

Exercise 5.4 [Support Vector Machines II]

Suppose we are given the following dataset $\mathcal{D}$ of observations with feature x and class label y.

x	y
-3	1
-2	1
-1	-1
0	-1
1	-1
3	1

(a) Is the dataset linearly separable (in the current feature space)?

(b) Consider the map $\phi(x) = [x, x^2]^\mathsf{T}$. Is the dataset linearly separable in the feature space induced by ϕ? If so, give the hyperplane with maximum margin that separates the dataset, and compute the margin.

(c) Which decision boundary for the original one-dimensional feature space does your solution for Part (b) imply?

Exercise 5.5 [Kernels]

(a) Which of the following are Mercer kernels?

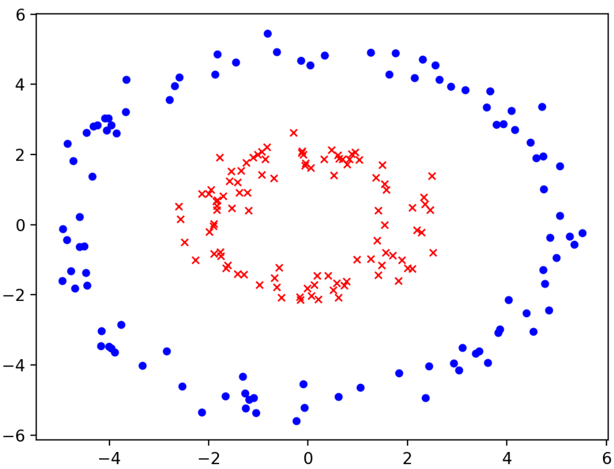
(i) $f(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^\mathsf{T}\mathbf{x}_2)^2 + (1 - \mathbf{x}_1^\mathsf{T}\mathbf{x}_2)^2$

(ii) $f(\mathbf{x}_1, \mathbf{x}_2) = (1 - \mathbf{x}_1^\mathsf{T}\mathbf{x}_2)^2$

(b) We recall the nearest-neighbour classifier as presented in Sheet 2. In its (maybe) easiest form, a nearest-neighbour classifier assigns a new input vector $\mathbf{x}$ to the same class as that of the nearest input vector $\mathbf{x}_n$ from the training set, where the distance is defined by the Euclidean metric $||\mathbf{x} - \mathbf{x}_n||^2$.

By expressing this rule in terms of scalar products and then making use of kernel substitution, formulate the nearest-neighbour classifier for a general nonlinear kernel.

(c) Consider the two-dimensional dataset with a binary class label that is given by the following plot.



Propose a map ϕ such that the dataset becomes linearly separable in the feature space induced by that map. Give the corresponding kernel function.