Machine Learning 1, WS 23/24

Exam 04.03.2024 (Ersttermin)

Lecture: Prof. Klaus-Robert Müller
Exercise: Jacob Kauffmann

(Solutions and grade distribution on the last pages)

1 Multiple Choice ($5 \times 4 = 20$ P)

Answer the following multiple choice questions. There is only one good answer per question. To mark an answer, put an x in box the next to it. For each question, no or false answer is zero points, correct answer is full points.

(a) Which of the following is True for the Expectation-Maximization (EM) algorithm?

- ☐ The EM algorithm is primarily used for supervised learning tasks.
- ☐ The EM algorithm always guarantees convergence to the global maximum likelihood.
- ☒ The EM algorithm iteratively estimates parameters for missing or hidden data.
- ☐ The EM algorithm requires labeled data to initialize its parameters.

(b) Which of the following is True for Principal Component Analysis (PCA)? The primary purpose of eigenvalue decomposition of the covariance matrix is...

- ☐ To classify the data into different clusters.
- ☒ To identify the directions that maximize the variance in the dataset.
- ☐ To normalize (or whiten) the data.
- ☐ To directly compute the mean and standard deviation of each variable.

(c) Which of the following is True in the context of bias-variance decomposition?

- ☒ High bias is indicative of underfitting, while high variance suggests overfitting.
- ☐ Higher bias always leads to lower variance, and vice versa.
- ☐ Bias measures the algorithm's flexibility, while variance measures its accuracy.
- ☐ Increasing the model complexity will generally decrease both bias and variance.

(d) Which of the following is True for a Bayes optimal classifier?

- ☐ It is synonymous with the Naive Bayes classifier, using independence assumptions.
- ☐ It refers to a classifier that relies solely on Bayesian probability theory.
- ☐ It is a specific algorithm that always outperforms other classifiers.
- ☒ It represents a theoretical framework that gives the lowest possible error rate.

(e) Which of the following is True: In the soft-margin SVM, the parameter C controls:

- ☐ The number of training points that are allowed to be misclassified.
- ☐ The number of test points that are allowed to be misclassified.
- ☒ By what amount the training points can lie not on the correct side of the margin.
- ☐ How nonlinear the margin is allowed to be.

2 Maximum Likelihood and Bayes ($5+5+5+5=20$ P)

The geometric distribution is given by

$$P(x \mid \theta) = \theta (1-\theta)^{x-1}$$

where $x \in \{1, 2, 3, \ldots\}$ and $0 < \theta < 1$ is a parameter. Let $D = \{x_1, x_2, \ldots, x_N\}$ be a dataset of independent draws from that distribution. We would like to learn the parameter θ from data.

$P(D|\theta) = \prod_k^N P(x_k|\theta) = \prod_k^N \theta(1-\theta)^{x_k-1}$

$= \theta^4(1-\theta)$

$4\log\theta + \log(1-\theta)$

$\frac{d}{d\theta} = \frac{4}{\theta} - \frac{1}{1-\theta} = 0$

$4 - 4\theta - \theta = 0$

$\theta = \frac{4}{5}$

(a) Write the likelihood function $P(D \mid \theta)$.

(b) Compute the maximum likelihood solution for θ given the dataset $D = \{1, 1, 2, 1\}$.

Now, we adopt a Bayesian viewpoint and assume that the parameter θ has prior probability

$$p(\theta) = \begin{cases} 1 & \text{if } 0 < \theta < 1 \\ 0 & \text{otherwise} \end{cases}$$

$p(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta}$

$= \frac{\theta^4(1-\theta)}{\int_0^1 \theta^4(1-\theta)d\theta} = \frac{\theta^4(1-\theta)}{\frac{1}{5}\theta^5 - \frac{1}{6}\theta^6\Big|_0^1} = 30\theta^4(1-\theta)$

(c) Show that the posterior distribution $p(\theta \mid D)$ after observing D as in (b) is given by $p(\theta \mid D) = 30\theta^4(1-\theta)$ on the interval $\theta \in [0, 1]$ and zero elsewhere.

(d) Evaluate under this posterior distribution the probability that x is larger than one, i. e. evaluate

$$\int P(x > 1 \mid \theta)\, p(\theta \mid D)\, d\theta.$$

$= 1 - \int P(x=1|\theta) P(\theta|D) d\theta$

$= 1 - \int \theta(1-\theta)^0 \cdot 30\theta^4(1-\theta) d\theta$

$= 1 - \int_0^1 30(\theta^5 - \theta^6) d\theta$

$= 1 - 30\left(\frac{1}{6}\theta^6 - \frac{1}{7}\theta^7\right)\Big|_0^1 = \frac{2}{7}$

3 Kernels $(5 + 15 = 20$ **P**$)$

A kernel $k\colon X \times X \to \mathbb{R}$ is positive semi-definite (psd) if

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j k\left(x_i, x_j\right) \geq 0$$

for all collections of inputs $x_1, \ldots, x_N \in X$ and real numbers $c_1, \ldots, c_N \in \mathbb{R}$. If a kernel is psd, then there exists a feature map $\phi\colon X \to F$ such that

$$\forall x, x' \in X\colon k\left(x, x'\right) = \langle \phi\left(x\right), \phi\left(x'\right)\rangle.$$

$\sum_i \sum_j c_i c_j k_f(x_i, x_j)$
$= \sum_i \sum_j c_i c_j f(x_i) k(x_i, x_j) f(x_j)$
$= \sum_i \sum_j c_i c_j f(x_i) \left(\sum_k \phi_k(x_i)\phi_k(x_j)\right) f(x_j)$
$= \sum_k \left(\sum_i c_i f(x_i)\phi_k(x_i)\right)^2 \geq 0$

(a) Show that if $k\left(x, x'\right)$ is a psd kernel, then

$$k_f\left(x, x'\right) = f\left(x\right) k\left(x, x'\right) f\left(x'\right)$$

is also a psd kernel for $f\colon X \to \mathbb{R}$.

(b) We now consider the real input space $X = \mathbb{R}^d$ and the psd kernel $k\left(x, x'\right) = \exp\left(\gamma\langle x, x'\rangle\right)$. Show that the Gaussian kernel,

$$k_f\left(x, x'\right) = \exp\left(-\gamma \cdot \frac{1}{2}||x - x'||^2\right)$$

is psd, and find the function f. Hint: Combine the fact that $k\left(x, x'\right) = \exp\left(\gamma\langle x, x'\rangle\right)$ is psd with the result from (a).

$||x-x'||^2 = (x-x')^T(x-x')$
$= x^Tx - 2x^Tx' + x'^Tx'$
$= \langle x, x\rangle - 2\langle x, x'\rangle + \langle x', x'\rangle$

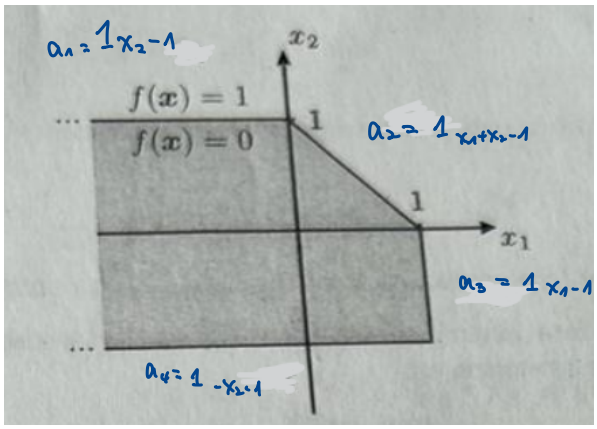
$\sum_i \sum_j c_i c_j k_f(x, x') = \sum_i \sum_j c_i c_j$
$= \sum_i \sum_j c_i c_j$
$= \sum_i \sum_j c_i c_j$

$\exp\left(-\gamma \sum_i ||x-x'||^2\right) = \exp\left[-\gamma \sum_i \cdot (\langle x, x\rangle - 2\langle x, x'\rangle + \langle x', x'\rangle)\right)$

$= \underbrace{\exp\left(-\gamma \sum_i \langle x, x\rangle\right)}_{f(x)} \cdot \underbrace{\exp(\gamma\langle x, x'\rangle)}_{\text{psd kernel}} \cdot \underbrace{\exp\left(-\gamma \sum_i \langle x', x'\rangle\right)}_{f(x)}$

4 Neural Networks $(15 + 5 = 20$ **P**$)$

The diagram below shows the decision boundary implemented by the function $f(x)$ for classifying data points $x \in \mathbb{R}^2$.



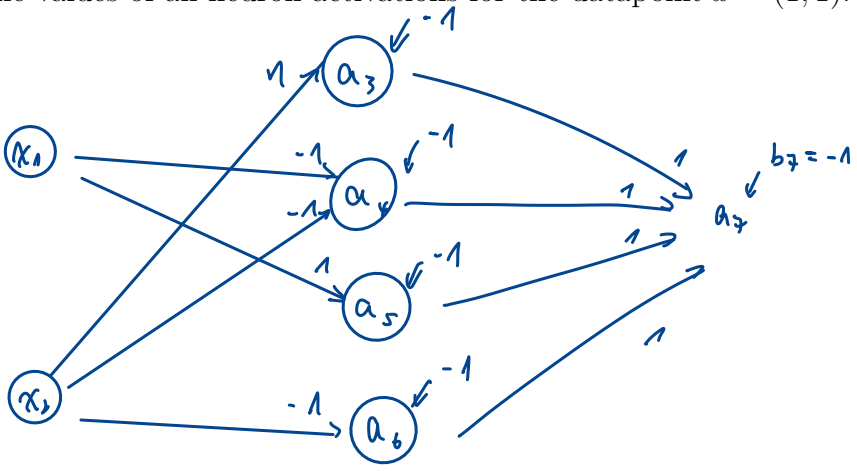
We consider the neurons

$$a_j = 1_{\{\sum_i w_{ij} a_i + b_j \geq 0\}}$$

where a_i are the activations of incoming neurons and w_{ij} and b_j are the real-valued parameters of the neuron.

(a) Construct a neural network composed exclusively of the elementary neurons above, that implements the function $f(x)$. Annotate your drawing with the relevant variable names, and the values of the neurons's parameters.

(b) Write the values of all neuron activations for the datapoint $x = (1, 1)$.



(b) $a_3 = 1 \quad a_4 = 0 \quad a_5 = 1 \quad a_6 = 0$

$a_7 = 1$

5 Using a Quadratic Solver ($5 + 15 = 20$ P)

Given a labeled dataset $((x_1, y_1), \ldots, (x_N, y_N))$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we consider the regularized regression problem

$$\min_w \sum_{n=1}^{N} \left(y_n - w^T x_n\right)^2,$$

$= \min_w \sum_{n=1}^{N} - 2\, y_n w^T x_n + w^T x_n w^T x_n$

$\Rightarrow \min_w - 2 y^T X w + w^T X^T X w$

where $w \in \mathbb{R}^d$ is a trainable patameter subject to the constraints

$$\forall i \in \{1, \ldots, d\} : 0 \le w_i \le C \text{ and } \sum_{i=1}^{d} w_i \le D.$$

$w_i \le C$

$-w_i \le 0$

$\Rightarrow \begin{bmatrix} I \\ -I \end{bmatrix} w \le \begin{bmatrix} C \\ 0 \end{bmatrix}$

$[1\ 1\ \ldots\ 1]\, w \le [D]$

The scalars C and D are hyperparameters of the model that control the amount of regularization.

(a) Denotng by X the data matrix of size $(N \times d)$, and by y the vector of targets of size N, show that the objective can be rewritten as

$$\min_w [w^T X^T X w - 2 y^T X w]$$

subject to the same constraints as before.

$Q = X^T X \qquad l = -2 X^T y$

(b) You have at your disposal a quadratic solver that solves the generic problem

$$\min_v v^T Q v + l^T v \text{ s.t. } Av \le b.$$

Input and output of this function are numpy arrays. Write down the code that builds the numpy arrays Q, l, A, B from the data X, y also given as numpy arrays. (A function template was given.)

```
Q = X.T @ X
l = -2 * X.T @ y
N,d = X.shape
A = np.concatenate([ np.eye(d) , -np.eye(d), np.ones((1,d))] )
b = np.concatenate([C*np.ones(d) , np.zeros(d), np.array([D])
```

6 Solutions

6.1 Multiple Choice

(a) 3 The EM algorithm iteratively estimates parameters for missing or hidden data.

(b) 2 To identify the directions that maximize the variance in the dataset.

(c) 1 High bias is indicative of underfitting, while high variance suggests overfitting.

(d) 4 It represents a theoretical framework that gives the lowest possible error rate.

(e) 3 By what amount the training points can lie not on the correct side of the margin.

6.2 Maximum Likelihood and Bayes

(a) $P(D \mid \theta) =$ whatever form you like

(b) $\theta = \frac{4}{5}$

(c) Use Bayes rule. Maybe take care that in the prior we have $(0, 1)$ and in the posterior we have $[0, 1]$.

(d) $\mathbb{E}(x > 1 \mid D) = \frac{2}{7}$

6.3 Kernels

(a) Standard proof.

(b) Choose $f(x) = \exp\left(-\gamma \cdot \frac{1}{2}||x||^2\right)$ and use $||x - x'||^2 = ||x||^2 + ||x'||^2 - 2\langle x, x' \rangle$.

6.4 Neural Networks

(a) One can draw a neural network with 4 neurons in the first layer, which represent the four parts of the boundary and one neuron as output neuron in the second layer. After asking in the exam, they clarified that the value on the boundary itself is not important.
Even though there are larger networks doing the same thing and there isn't written anything about the optimality of the solution in the task, they wanted to have this simple solution. A bigger network was leading to only getting 10 or even 5 points.

(b) Just write $a_1 = \dots$ with the right values.

6.5 Using a Quadratic Solver

(a) Calculate.

(b)

$$
\begin{aligned}
Q &= X^T X \\
l &= -2X^T y \\
A &= \begin{pmatrix} -I \\ I \\ \mathbf{1}^T \end{pmatrix} \\
b &= \begin{pmatrix} \mathbf{0} \\ C \cdot \mathbf{1} \\ D \end{pmatrix}.
\end{aligned}
$$

One can use the numpy functions *np.eye*, *np.zeros* and *np.ones* to build the matrices. Concatenation can be done using *np.concatenate*, *np.hstack* or *np.block*.

7 Grade Distribution

The maximum number of points is 100. The grade distribution is as follows:
1.0 for at least 95 points, then every note is 5 points less. This means one needs 50 points to pass.