Machine Learning 1 — Final Exam

1 Preliminaries

- Please write your immatriculation number **but not your name** on *every* page you hand in.
- The exam is closed book. You may, however, use one A4 sheet of notes, handwritten before the exam.
- The exam is limited to 2 × 60 minutes.
- If a question says “Describe in 2–3 sentences” or “Show your work” or “Explain your answer” or so, these mean the same: give a succinct description or explanation.
- This exam consists of 4 pages, 12 problems. You can earn up to 24 points.

Problem 0 [$\sqrt{-4}$ points] We suffer from gender bias. It’s not clear which gender we favour in scoring, but it certainly—so science tells us—influences our grading skills. Help yourself by *not* writing your name on your sheets. Just fill in your immatriculation number on every sheet you hand in. Make sure it is easily readable.

2 Linear Algebra and Probability Theory

For the next two exercises, let X and Y be two random variables with the *joint cumulative distribution function* (cdf)

$$F_{X,Y} = \begin{cases} 1 - e^{-x} - e^{-y} + e^{-x-y} & x, y \geq 0 \\ 0 & \text{else} \end{cases} .$$

Problem 1 [3 points] Determine the *marginal probability density functions* (pdfs) f_X and f_Y.

Identify the marginal distributions.

We recall:

$$\begin{tikzcd}
f_X(x) \arrow[d, bend right, "\int_{-\infty}^{x} f_X(u)\,du"'] & f_{X,Y}(x,y) \arrow[l, "\int_{-\infty}^{\infty} f_{X,Y}(x,y)\,dy"'] \arrow[r, "\int_{-\infty}^{\infty} f_{X,Y}(x,y)\,dx"] \arrow[d, bend right, "\int_{-\infty}^{y}\int_{-\infty}^{x} f_{X,Y}(u,v)\,du\,dv"'] & f_Y(y) \arrow[d, bend right, "\int_{-\infty}^{y} f_Y(v)\,dv"'] \\
F_X(x) \arrow[u, bend right, "\frac{d}{dx} F_X(x)"'] & F_{X,Y}(x,y) \arrow[l, "\lim_{y\to\infty} F_{X,Y}(x,y)"] \arrow[r, "\lim_{x\to\infty} F_{X,Y}(x,y)"'] \arrow[u, bend right, "\frac{d}{dy}\frac{d}{dx} F_{X,Y}(x,y)"'] & F_Y(y) \arrow[u, bend right, "\frac{d}{dy} F_Y(y)"']
\end{tikzcd}$$

Two routes are possible (upwards-sidewards or sidewards-upwards in the illustration above).

imat:

For $x, y \geq 0$, the first route works as follow:

$$\begin{aligned} f_{X,Y}(x,y) &= \frac{\partial^2 F_{X,Y}}{\partial x \partial y}(x,y) \\ &= \frac{\partial}{\partial y}\left(e^{-x} - e^{-x-y}\right) \\ &= e^{-x-y} = e^{-x}e^{-y} \\ f_X(x) &= \int_{-\infty}^{\infty} f_{X,Y}(x,y)\,\mathrm{d}y \\ &= e^{-x}\int_0^{\infty} e^{-y}\,\mathrm{d}y \\ &= e^{-x}(0-(-1)) = e^{-x} \end{aligned}$$

The second route works as follows:

$$\begin{aligned} F_X(x) &= \lim_{y\to\infty} F_{X,Y}(x,y) \\ &= 1 - e^{-x} - \left(1 - e^{-x}\right)\lim_{y\to\infty} e^{-y} \\ &= 1 - e^{-x} \\ f_X(x) &= \frac{\mathrm{d}F_X}{\mathrm{d}x}(x) \\ &= e^{-x} \end{aligned}$$

We recognize the pdf of an exponential distribution with parameter $\lambda = 1$.

Furthermore, we observe that Y follows the same distribution: Due to the symmetry $F_{X,Y}(x,y) = F_{X,Y}(y,x)$, all calculations shown above work equivalently for Y.

Problem 2 [2 points] Are X and Y independent? Prove your claim.

One can show either $f_{X,Y} = f_X f_Y$ or $F_{X,Y} = F_X F_Y$, which shows independence of the two variables. For calculations, cf. the solution to the previous exercise.

3 kNN

Problem 3 [1 point] You are testing three new sensors. You would like to classify them using the nearest neighbour approach with Euclidean distance ($d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$).

Sensor	Output	Nearest Neighbour (current)	Nearest Neighbour (expected)
sensor1	$[1, 150]$	sensor2	sensor3
sensor2	$[2, 110]$	sensor3	sensor1 or sensor3
sensor3	$[1, 100]$	sensor2	sensor1

Your sensors output two parameters each. This output is in the above table. Since this is a controlled test of the sensors, you know that sensor1 and sensor3 belong to the same group and that sensor2 belongs to a different group. We can see that sensor2 is causing trouble for the classification of the other sensors. We would expect the output to look like the fourth column in the above table. How can you fix this problem without changing your distance measure from Euclidean?

Normalise the values of the parameters. (Whether L1, L2, or infinity norm, normalisation will fix this problem)

Problem 4 [1 point] In general, especially with data which might have many uninformative parameters, why can the use of Euclidean distance as the distance measure be problematic? (Even after the problem from above has been fixed?)

Euclidean distance treats each parameter with the same weight. Thus, uninformative parameters can overpower predictive parameters.

4 Gaussian Processes

We have a data set $x \in \mathbb{R}^1$. You are given Gaussian processes $f \sim GP$ with mean function $m(x) = 0$, covariance function $k(x, x')$, and a noisy observation $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$.

Problem 5 [2 points] Assume the covariance function is $k(x, x') = (xx' + 1)^2$, and the observation data $x_1 = -\frac{1}{2}, x_2 = 2$ is given, write down the distribution of $p(f(x_1), f(x_2))$. What is the relationship of $f(x_1)$ and $f(x_2)$? Describe your reasoning.

$$p(f(x_1), f(x_2)) = \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \frac{25}{16} + \sigma_y^2 & 0 \\ 0 & 25 + \sigma_y^2 \end{pmatrix}\right)$$

(1 point)

The covariance function is diagonal, so they are independent. (1 point)

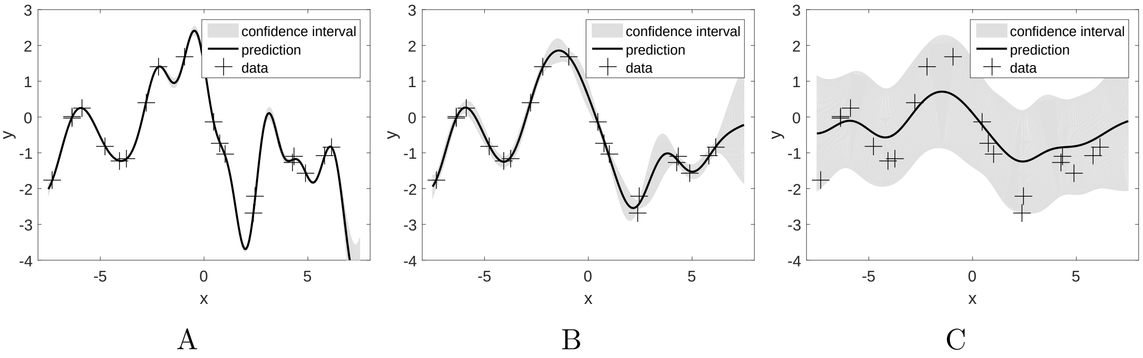


Figure 1: Gaussian Processes

Problem 6 [2 points] We have a squared exponential (SE) kernel. With different values of σ_y^2, the GP models are shown in Fig. 1. Which model is best? What causes the other two to be not good? Explain your answer.

B is best. A noise variance σ_y^2 is too small, and C is too large.

5 Neural networks

Problem 7 [3 points] A neural network with activation functions tanh(·) in the hidden units is initialised with all parameters (weights, biases) set to 0. Can it learn? Explain your answer.

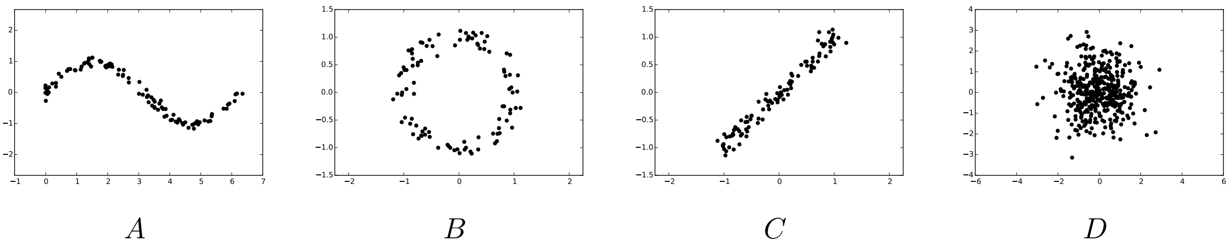
It's a bit of a trick question, really, but most people should get at least 2 points out of it.

(+2 points) No, not with a gradient-based rule. If all parameters are 0, then the output is independent of the input. So all weights would get the same gradient. Biases can generally be initialised to zero but weights need to be initialised carefully to break the symmetry between hidden units of the same layer. Because different output units receive different gradient signals, this symmetry breaking issue does not concern the output weights (into the output units), which can therefore also be set to zero.

(+1 point) But yes, if you use a non-gradient based learning method. Like random search.

Problem 8 [1 point] A neural network with activation functions tanh(·) in the hidden units is initialised with all parameters (weights, biases) set to 1. Can it learn? Explain your answer.

It might not work *well*, but it would work as long as the neurons do not saturate.



A *B* *C* *D*

Figure 2: Four datasets

6 Unsupervised learning

Problem 9 [3 points] PCA was performed on dataset A, B and C (Fig.: 2). The three results are:

$$\begin{array}{rccc} & \text{Result 1} & \text{Result 2} & \text{Result 3} \\ \text{normalised eigenvalues} = & [0.5, 0.5] & [0.95, 0.05] & [0.99, 0.01] \\ \text{normalised eigenvectors} = & \left[\begin{pmatrix}0.78\\0.63\end{pmatrix}, \begin{pmatrix}0.63\\-0.78\end{pmatrix}\right] & \left[\begin{pmatrix}0.96\\-0.27\end{pmatrix}, \begin{pmatrix}0.27\\0.96\end{pmatrix}\right] & \left[\begin{pmatrix}0.71\\0.71\end{pmatrix}, \begin{pmatrix}0.71\\-0.71\end{pmatrix}\right] \end{array}$$

Unfortunately the results got mixed up. Which result corresponds to which dataset (A, B, C)? Explain your answer.

(+1.5 points) $A2$, $B1$, $C3$

(+1.5 points) for the explanations.

Problem 10 [1 point] What would the result on dataset D look like (Fig.: 2)? Use the same notation as in Problem 9.

(+1 point) $\begin{array}{c}[0.5, 0.5] \\ \left[\begin{pmatrix}a\\b\end{pmatrix}, \begin{pmatrix}b\\-a\end{pmatrix}\right]\end{array}$

7 Kernels

Consider the following algorithm.

Algorithm 1: Counting something

```
input  : Character string $x$ of length $m$ (one based indexing)
input  : Character string $y$ of length $n$ (one based indexing)
output: A number $s \in \mathbb{R}$

$s \leftarrow 0$;
for $i \leftarrow 1$ to $m$ do
    for $j \leftarrow 1$ to $n$ do
        if $x[i] == x[j]$ then
            $s \leftarrow s + 1$;
```

Problem 11 [1 point] Explain, in no more than two sentences, what the above algorithm is doing.

The algorithm sums how many times each character c from string x appears in string y (or vice versa).

Problem 12 [4 points] Let $\mathcal{S}$ denote the set of strings over a finite alphabet of size v. Define a function $K : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ as the output of running algorithm 1 on a pair of strings x, y. Show that $K(x, y)$ is a valid kernel.

Algorithm 1 can be rewritten as follows

```
$s \leftarrow 0$;
for $i \leftarrow 1$ to $m$ do
    $s \leftarrow s +$ (number of occurrences of $x[i]$ in $y$);
```

and furthermore

```
$s \leftarrow 0$;
for $c \leftarrow 1$ to $v$ do
    $s \leftarrow s +$ (number of occurrences of $c$ in $x$) $\times$ (number of occurrences of $c$ in $y$);
```

Thus defining the feature map $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^v$ where $\boldsymbol{\phi}(x)_c$ is the number of occurrences of character c in string x, we see that the above algorithm computes $K(x, y) = \boldsymbol{\phi}(x) \cdot \boldsymbol{\phi}(y)$. Since we constructed an explicit feature map $\boldsymbol{\phi}$, K is a valid kernel.