Machine Learning — Repeat Exam — SOLUTION

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	$\sum$

Do not write anything above this line

Name:

Student ID: Signature:

- Only write on the sheets given to you by supervisors. If you need more paper, ask the supervisors.
- All sheets (including scratch paper) have to be returned at the end.
- **Do not unstaple the sheets!**
- Wherever answer boxes are provided, please write your answers in them.
- Please write your student ID (*Matrikelnummer*) on every sheet you hand in.
- **Only use a black or a blue pen (no pencils, red or green pens!).**
- You are allowed to use your A4 sheet of handwritten notes (two sides). **No other materials (e.g. books, cell phones, calculators) are allowed!**
- Exam duration - 120 minutes.
- This exam consists of 17 pages, 19 problems. You can earn 50 points.

Probability distributions

For your reference, we provide the following probability density functions.

- Normal distribution

$$\mathcal{N}(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- Gamma distribution

$$\text{Gamma}(x \mid \alpha, \beta) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} & \text{if } x \in (0, \infty), \\ 0 & \text{else} \end{cases}$$

where $\Gamma(\cdot)$ is the gamma function.

- Log-normal distribution

$$\text{Log-normal}(x \mid \mu, \tau) = \begin{cases} \frac{\sqrt{\tau}}{x\sqrt{2\pi}} \exp\left(-\frac{\tau}{2}(\log x - \mu)^2\right) & \text{if } x \in (0, \infty), \\ 0 & \text{else.} \end{cases}$$

1 Probability Theory

Problem 1 [1 point] Let Ω be the sample space. Let $A \subset \Omega$ and $B \subset \Omega$ be two independent events. Prove or disprove that their complements $A^c = \Omega \backslash A$ and $B^c = \Omega \backslash B$ are also independent.

$$\begin{aligned} P(A^c \cap B^c) &= 1 - P(A \cup B) \\ &= 1 - P(A) - P(B) + P(A \cap B) \quad \text{\# def. of union} \\ &= 1 - P(A) - P(B) + P(A)P(B) \quad \text{\# independece} \\ &= (1 - P(A))(1 - P(B)) \\ &= P(A^c)P(B^c). \end{aligned}$$

Problem 2 [2 points] There are two coins $C1$ and $C2$. $C1$ has an equal probability to land heads (H) and tails (T). The behavior of $C2$ depends on the outcome of flipping $C1$. If $C1$ lands heads, $C2$ will land heads with probability 0.7. If $C1$ lands tails, $C2$ will land heads with probability 0.5. $C1$ and $C2$ are tossed in sequence once, and exactly one of the two coins lands heads. What is the probability that $C1 = T$ and $C2 = H$?

Denote with A the event ($C_1 = T$ and $C_2 = H$) and denote with B the event ($C_1 = H$ and $C_2 = T$).

We are interested in $p(A \mid A \cup B)$, which by the Bayes formula is

$$\begin{aligned} p(A \mid A \cup B) &= \frac{p(A \cap (A \cup B))}{p(A \cup B)} \\ &= \frac{p(A)}{p(A) + p(B) + \underbrace{p(A \cap B)}_{=0}} \\ &= \frac{p(C1 = T \text{ and } C2 = H)}{p(C1 = T \text{ and } C2 = H) + p(C1 = H \text{ and } C2 = T)} \\ &= \frac{p(C2 = H | C1 = T)p(C1 = T)}{p(C2 = H | C1 = T)p(C1 = T) + p(C2 = T | C1 = H)p(C1 = H)} \\ &= \frac{0.5 \times 0.5}{0.5 \times 0.5 + (1 - 0.7) \times 0.5} \\ &= \frac{5}{8} \end{aligned}$$

2 Decision Trees

Problem 3 [3 points] Given is a dataset with 4 binary attributes and a binary label. You want to train a decision tree using the standard ID3 algorithm.

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Label
T	T	T	F	1
T	T	T	F	1
F	T	T	F	1
F	T	F	F	1
T	T	F	F	0
T	T	F	F	0
F	T	F	F	0
F	T	T	F	0

a) Which attribute will be selected as the first (root) node using entropy as a purity measure? Show your work!

Attribute 2 and Attribute 4 have the same value for all 8 instances (T and F respectively). Therefore, we can conclude that $\Delta_{iH} = 0$ for both of them.

Attribute 1 has the same class distribution (T, T, F, F) for both labels. Similarly we can again conclude that $\Delta_{iH} = 0$.

Since the class distribution for Attribute 3 is different for the different labels, we must have $\Delta_{iH} > 0$. Therefore, we will select Attribute 3 as the first (root) node.

b) Does there exist a decision tree that can achieve 100% accuracy on the training set? If yes, draw that decision tree, otherwise provide a simple explanation.

Since we have two instances that have the exact same attributes but different labels (instances 3 and 8), a decision tree that can achieve 100% accuracy on the training set does not exist.

Problem 4 [1 point] Suppose we are learning a decision tree for binary classification on a dataset that consists of N samples of dimensionality M (i.e., each sample has M features).

a) Can we conclude that the maximum depth of the decision tree must be less then $1 + \log_2(N)$? Why or why not?

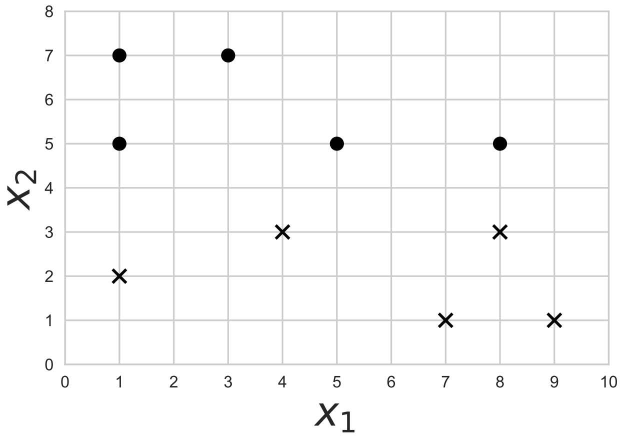
No. The tree may be unbalanced.

b) Can we alternatively conclude that the maximum depth must be less than $M + 1$? Why or why not?

No. We can use each feature multiple times.

3 K Nearest Neighbors

Problem 5 [3 points] Given is the data in the figure below, with binary labels marked as • and ×:



a) How many points get misclassified during the leave-one-out cross-validation (LOOCV) procedure when using a 1-NN classifier?

Five.

b) The original data can be represented as a matrix $\boldsymbol{X}_{orig} \in \mathbb{R}^{10\times 2}$, where samples are stored as rows. We can apply a linear transformation to the data as $\boldsymbol{X}_{new} = \boldsymbol{X}_{orig} \cdot \boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ is the transformation matrix.

Find a linear transformation matrix $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$, such that all points get classified correctly during the LOOCV procedure using a 1-NN classifier. Provide a short explanation.

The transformation matrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

projects the data onto the x_2 axis, effectively disregarding the x_1 attribute. Given such a projection we can see that all points get correctly classified using the LOOCV procedure.

4 Regression

Problem 6 [4 points] Given is a training set consisting of samples $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ with respective regression targets $\boldsymbol{y} = \{y_1, y_2, \dots, y_N\}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$.

Alice fits a linear regression model $f(\boldsymbol{x}_i) = \boldsymbol{w}^T\boldsymbol{x}_i$ to the dataset using the closed form solution for linear regression (normal equations).

Bob has heard that by transforming the inputs $\boldsymbol{x}_i$ with a vector-valued function $\boldsymbol{\phi}$, he can fit an alternative function, $g(\boldsymbol{x}_i) = \boldsymbol{v}^T\boldsymbol{\phi}(\boldsymbol{x_i})$, using the same procedure (solving the normal equations). He decides to use a linear transformation $\boldsymbol{\phi}(\boldsymbol{x_i}) = \boldsymbol{A}^T\boldsymbol{x}_i$, where $\boldsymbol{A} \in \mathbb{R}^{D\times D}$ has full rank.

a) Show that Bob's procedure will fit the same function as Alice's original procedure, that is $f(\boldsymbol{x}) = g(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^D$ (given that $\boldsymbol{w}$ and $\boldsymbol{v}$ minimize the training set error).

By fitting $g(\boldsymbol{x}) = \boldsymbol{v}^T \boldsymbol{\phi}(\boldsymbol{x})$, Bob can only set the function equal to linear combinations of the inputs $(\boldsymbol{v}^T \boldsymbol{A})\boldsymbol{x} = \boldsymbol{w}^T \boldsymbol{x}$, where $\boldsymbol{w} = \boldsymbol{A}^T \boldsymbol{v}$.

Moreover, just like Alice, all linear combinations are available: any function Alice fits can be matched by setting $\boldsymbol{v} = \boldsymbol{A}^{-1}\boldsymbol{w}$.

As both Alice and Bob are selecting the function that best matches the outputs from the same set of functions, and with the same cost function, they will select the same function.

b) Can Bob's procedure lead to a lower training set error than Alice's if the matrix $\boldsymbol{A}$ is not invertible? Explain your answer.

Since $\boldsymbol{A}$ is square and not invertible, then multiple input vectors are transformed to the same output vector. It's not possible for Bob to assign different function values to two such inputs, whereas Alice can. Bob can no longer fit all the same functions as Alice. As a result, Bob's training error might be worse than Alice's, but can't be better.

Problem 7 [2 points] Assume that we are given a dataset, where each sample x_i and regression target y_i is generated according to the following process

$$x_i \sim \text{Uniform}(-10, 10)$$
$$y_i = ax_i^3 + bx_i^2 + cx_i + d + \epsilon_i, \quad \text{where} \quad \epsilon_i \sim \mathcal{N}(0,1) \quad \text{and} \quad a, b, c, d \in \mathbb{R}.$$

The 4 regression algorithms below are applied to the given data. Your task is to say what the bias and variance of these models are (low or high). Provide a 1-2 sentence explanation to each of your answers.

a) Linear regression

Bias: Variance:

Bias: high. Variance: low.

A straight line cannot capture a degree 3 polynomial (thus underfitting the data).

b) Polynomial regression with degree 3

Bias: Variance:

Bias: low. Variance: low.

The model is same as the data generating process. We can achieve a good fit.

c) Polynomial regression with degree 10

Bias: Variance:

Bias: low. Variance: high.

Since we are using a polynomial regression with a degree much higher compared to the data generating process, the model will overfit the data.

d) K-NN regression with $K = 1$

Bias: Variance:

Bias: low. Variance: high.

1-NN regression will overfit the data (since every noisy sample will determine the target for its closest neighbor).

5 Classification

Problem 8 [5 points] We have a binary classification problem with 1-dimensional data. The training samples from class 1 are $\{-1, 1, 3, 6, 1\}$ and the samples from class 2 are $\{5.5, 7, 9, 9, 9.5\}$.

We model class priors as a categorical distribution with parameters $\{\pi_1, \pi_2\}$. We model the class-conditionals as Gaussian distributions $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$. Assume that the variances of class-conditionals are known to be $\sigma_1^2 = 2$ and $\sigma_2^2 = 1$.

a) Find the maximum likelihood estimates (MLE) of the prior class probabilities $\{\pi_1, \pi_2\}$, as well as of the means $\{\mu_1, \mu_2\}$ of the class conditional densities.

The MLE solution for the prior class probabilities is equal to the fraction of instances in each class:

$$\pi_1 = \pi_2 = \frac{5}{10} = 0.5$$

The MLE solution for the means is the the mean of the instances belonging to each class:

$$\mu_1 = \frac{-1+1+3+6+1}{5} = 2 \qquad \mu_2 = \frac{5.5+7+9+9+9.5}{5} = 8$$

b) What class will the point $x = 6$ be assigned to? Justify your answer. *Hint:* $\ln 1/\sqrt{2} \approx -0.35$

Ignoring constants we have:

$$p(y=1|x=6) = p(x=6|\mu=2, \sigma^2=2) \times \pi_1 = 0.5\frac{1}{\sqrt{2\pi 2}}e^{-\frac{(6-2)^2}{4}}$$

$$p(y=2|x=6) = p(x=6|\mu=8, \sigma^2=1) \times \pi_2 = 0.5\frac{1}{\sqrt{2\pi}}e^{-\frac{(6-8)^2}{2}}$$

Since this is a two class problem, it is convenient to calculate the log posterior probability ratio:

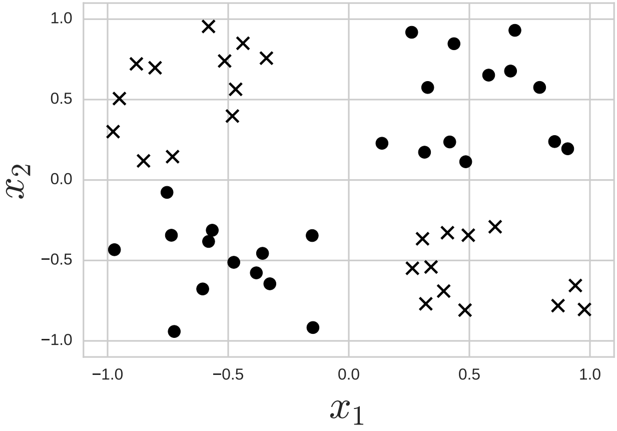
$$\ln\frac{p(y=1 \mid x=6)}{p(y=2 \mid x=6)} = \left(\ln(\frac{0.5}{\sqrt{2\pi}}) - \ln\frac{1}{\sqrt{2}} - \ln(\frac{0.5}{\sqrt{2\pi}}) + \ln\exp(-4+2)\right) \approx -2 \times 0.35 = -0.75$$

Thus, the point will be assigned to class 2.

c) How many points $x \in \mathbb{R}$ lie on the decision boundary? That is, for how many points $x \in \mathbb{R}$ does it hold $p(y=1 \mid x) = p(y=2 \mid x)$? Provide a mathematical justification for your answer.

Two. Quadratic.

Problem 9 [2 points] The goal is to perform binary classification on the dataset in the figure below. The two classes are denoted as •'s and ×'s.



Which of the following algorithms are able to achieve 100% accuracy on the training set (given an appropriate choice of hyperparameters)? For each of your answers provide a 1-sentence explanation.

a) Logistic regression.

No. The data is not linearly separable and logistic regression is a linear classifier.

b) SVM with a Gaussian kernel.

Yes. For a sufficiently small setting of the variance for the Gaussian kernel.

c) Decision tree of depth 2.

Yes. We can first split by $x_1 > 0$ then by $x_2 > 0$.

d) Feedforward neural network with 5 layers and no activation functions (i.e. $\sigma(x) = x$).

No. This is equivalent to a logistic regression since we don't have a non-linearity.

6 Convexity

Problem 10 [4 points] Consider the following function

$$F(x) = a(x) + b(\exp(-x)) - \min(c(x), d(x-1))$$

consisting of 4 component functions $a(\cdot)$, $b(\cdot)$, $c(\cdot)$ and $d(\cdot)$. Given are the following candidate functions:

where z is the argument fed into the function.

Assign the candidate functions to the component functions, such that $F(x)$ is convex on $\mathbb{R}_{>0}$ (set of positive real numbers). That is write the function name in the "component function" column in the table above. You can use each candidate and component function only once. One candidate function will remain unused.

Candidate functions	**Component function**
$(z-2)^3$	-
$-z^3$	c
$\exp(z)$	b
$-\frac{1}{z+2}$	d
$\exp(-z)$	a

Problem 11 [1 point] What are the advantages of having a convex objective function in optimization? (Explain in at most 2 sentences)

The function has no local optima. Every locally optimal solution is globally optimal. (Optionally: due to this property, more efficient algorithms can be constructed).

7 Gradient Descent

Problem 12 [2 points] Let $L(\mathbf{W})$ be a differentiable strictly convex loss function, where $\mathbf{W}$ is the set of model weights. Additionally, let $\mathbf{W}^{(0)}$ be the set of initial weights and $\mathbf{W}^{(1)}$ be the set of weights after performing one iteration of gradient descent.

Are the following statements true or false? Provide a clear explanation (at most 2-3 sentences), stating all the assumptions that you make.

a) The loss always improves, i.e., $L(\mathbf{W}^{(1)}) < L(\mathbf{W}^{(0)})$ for any choice of $\mathbf{W}^{(0)}$.

No, not necessarily. If the learning rate is too high for example, we may overshoot the minimum and have higher loss then before.

b) For any choice of $\mathbf{W}^{(0)}$ it is possible to choose a step size such that the loss improves after 1 iteration of gradient descent, i.e., $L(\mathbf{W}^{(1)}) < L(\mathbf{W}^{(0)})$.

No, not necessarily. If $\mathbf{W}^{(0)}$ is initialized to be at the global minimum, the gradient w.r.t. the loss is zero and gradient descent will not update the weights. Thus, the loss will not improve for any choice of the step size.

8 SVM & Constrained optimization

Problem 13 [5 points] The goal is to minimize the function $f : \mathbb{R} \to \mathbb{R}$ with $f(x) = 4x^2$ subject to $f_1(x) = -2x + 1 \leq 0$ using constrained optimization methods.

a) Write down the Lagrangian $L(x, \alpha)$ for this constrained optimization problem. Denote the necessary Lagrange multiplier as α.

$$L(x, \alpha) = f(x) + \alpha f_1(x) = 4x^2 + \alpha(1 - 2x) = 4x^2 - 2\alpha x + \alpha \,.$$

b) Obtain the Lagrange dual function $g(\alpha)$ from the Lagrangian $L(x, \alpha)$.

The Lagrange dual function is obtained by minimizing $L(x, \alpha)$ w.r.t. x. This is done by calculating the corresponding partial derivative,

$$\frac{\partial L}{\partial x} = 8x - 2\alpha \,,$$

setting it to zero and solving for x,

$$x = \frac{\alpha}{4} \,.$$

By substituting this back into $L(x, \alpha)$ we obtain

$$g(\alpha) = L(\alpha/4, \alpha) = -\frac{1}{4}\alpha^2 + \alpha \,.$$

c) State the dual problem explicitly and solve it to obtain the value for the Lagrange multiplier α.

The Lagrange dual problem is

$$\text{maximize } g(\alpha) = -\frac{1}{4}\alpha^2 + \alpha$$
$$\text{s.t. } \alpha \geq 0$$

The solution is given by calculating the derivative,

$$g'(\alpha) = -\frac{\alpha}{2} + 1 \,,$$

and setting it to zero. Solving for α gives

$$\alpha^* = 2 \,.$$

d) What is the duality gap in this problem? Justify your answer.

Since f is convex and f_1 is affine Slater's theorem applies and the duality gap is zero.

e) Obtain the solution to the original problem from the solution of the dual problem.

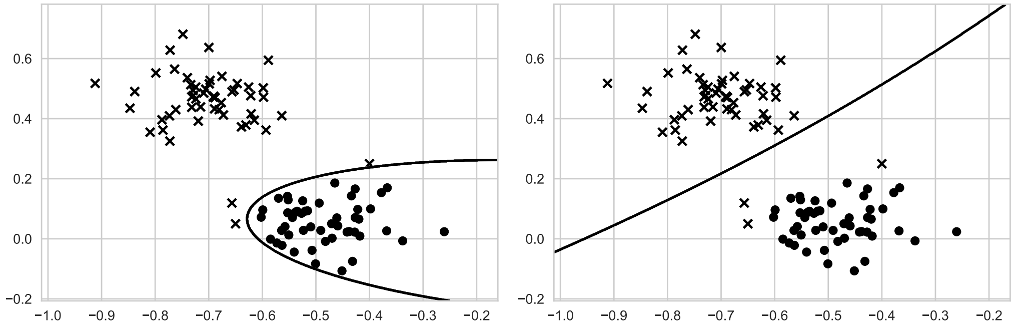
The minimizer x^* of the original optimization problem is given by

$$x^* = \frac{\alpha^*}{4} = \frac{2}{4} = \frac{1}{2}$$

and the minimum is

$$f(x^*) = 1\,.$$

Problem 14 [1 point] Sketch the decision boundary of an SVM with a quadratic kernel (polynomial with degree 2) for the data in the figure below, for two specified values of the penalty parameter C. (The two classes are denoted as $\bullet$'s and $\times$'s.)



(a) $C = 10^{10}$ (b) $C = 10^{-10}$

Explain the reasoning behind your sketch of the decision boundary for both cases (one sentence for each plot).

a) With such a large penalty SVM will try to correctly classify **all** of the instances in the training set.

b) Given the small penalty, we can allow few misclassified instances, and obtain a larger margin between the two classes. The decision boundary looks linear.

9 Deep Learning

Problem 15 [4 points] Consider the following classification problem. There are two real-valued features x_1 and x_2, and a binary class label. The ground truth class labels are generated according to the following rule:

$$y = \begin{cases} 1 & \text{if } x_2 \geq |x_1|, \\ 0 & \text{else} \end{cases}$$

a) Can this function be perfectly represented by a feed-forward neural network with no hidden layers and 1 softmax output layer? Why or why not?

No. The function is non-linear.

b) Design a two layer feed-forward network (that is, one hidden layer followed by an output layer, two weight matrices in total) that represents this function. You are allowed to use the hard thresholding activation function $\sigma(x)$ as the elementwise nonlinearity.

$$\sigma(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{else} \end{cases}$$

Specify the number of neurons and values of weights in each layer.

Let the first hidden neuron of the first layer compute the following output:

$$h_1 = \sigma(1.0 \times x_2 - 1.0 \times x_1 + 0.0) = \begin{cases} 1 & x_2 > x_1 \\ 0 & \text{else} \end{cases}$$

and the second hidden neuron compute the following output:

$$h_2 = \sigma(1.0 \times x_2 + 1.0 \times x_1 + 0.0) = \begin{cases} 1 & x_2 > -x_1 \\ 0 & \text{else} \end{cases}$$

In the second layer we combine them to produce the final output:

$$y = 1.0 \times h_1 + 1.0 \times h_2 - 1 = \begin{cases} 1 & x_2 > x_1 \quad \text{and} \quad x_2 > -x_1 \\ 0 & \text{else} \end{cases} = \begin{cases} 1 & x_2 > |x_1| \\ 0 & \text{else} \end{cases}$$

Here we've used the bias term -1 to construct what is effectively an AND gate.

Problem 16 [2 points] We apply an autoencoder to D-dimensional data. The autoencoder has a single K-dimensional hidden layer, there are no biases, and all activation functions are identity ($\sigma(x) = x$). Why is it usually impossible to get zero reconstruction error in this setting if $K < D$? Under which conditions is this possible?

We have $f(\boldsymbol{x}) = \boldsymbol{X}\boldsymbol{W}_1\boldsymbol{W}_2$ where $\boldsymbol{X}$ is the data matrix and the dimensions of the weight matrices are $D \times K$ for $\boldsymbol{W}_1$ and $K \times D$ for $\boldsymbol{W}_2$.

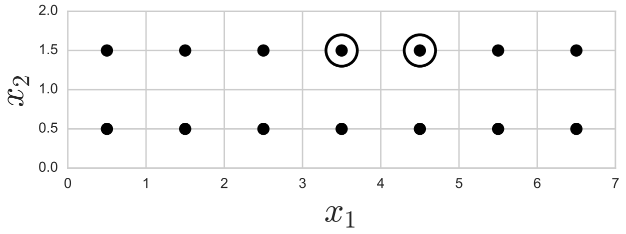
The final multiplication $\boldsymbol{W}_2$ brings points from K-dimensions up into D-dimensions but the points will still all be in a K-dimensional linear subspace. Unless the data happen to lie exactly in a K-dimensional linear subspace, they can't be exactly fitted.

10 Clustering

Problem 17 [3 points]

a) Given is the dataset displayed in the figure below. Apply the K-means algorithm to this data using $K = 2$ and using the circled points as initial centroids.

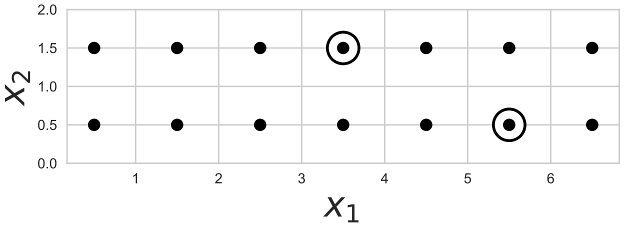
What are the clusters after K-Means converges? Draw your solution in the figure above, i.e. mark the location of the centroids with ×'s and show the clusters by drawing two bounding boxes around the points assigned to each cluster.



How many iterations did it take for K-Means to converge in the above problem?

One.

b) Provide a different initialization, for which the algorithm will take **more** iterations to converge to the **same** solution. Make sure that your initialization does not lead to ties. Draw your initialization in the figure below.



11 Variational inference

Problem 18 [3 points] You are given a dataset consisting of N positive samples $\boldsymbol{x} = \{x_1, x_2, \dots, x_N\}$, $x_i \in \mathbb{R}^+$. You model the data-generating distribution (i.e., the likelihood) using a log-normal distribution, that is

$$p(x_i \mid \mu, \tau) = \text{Log-normal}(x_i \mid \mu, \tau),$$

where μ is the *known* and *fixed* mean parameter, and τ is the *unknown* precision parameter. You choose a Gamma distribution as the prior for τ, that is $p(\tau \mid a, b) = \text{Gamma}(\tau \mid a, b)$.

You would like to approximate the posterior $p(\tau \mid \boldsymbol{x}, \mu, a, b)$ using variational inference. Which of the following families of variational distributions $q(\tau)$ will yield the best approximation (in terms of KL-divergence)?

a) $q(\tau) = \text{Log-normal}(\tau \mid \nu, \beta) = \frac{\sqrt{\beta}}{\tau\sqrt{2\pi}} \exp\left(-\frac{\beta}{2}(\ln\tau - \nu)^2\right)$

b) $q(\tau) = \text{Normal}(\tau \mid \nu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(\tau - \nu)^2\right)$

c) $q(\tau) = \text{Gamma}(\tau \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-1}\exp(-\beta\tau)$

d) $q(\tau) = \text{Inverse-gamma}(\tau \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{-\alpha-1}\exp\left(-\frac{\beta}{\tau}\right)$

Show your work! Just stating a), b), c) or d) is not enough!

Short answer:

For a Log-normal distribution with a known mean, Gamma is a conjugate prior for the precision. Therefore, the posterior of the precision is also a Gamma distribution. Thus, c) will yield the best approximation.

Long answer:

$$\begin{aligned} p(\tau \mid \boldsymbol{x}, \mu, a, b) &\propto \prod_i p(x_i \mid \mu, \tau) p(\tau \mid a, b) \\ &= \prod_i \Big[\tau^{\frac{1}{2}} \exp\big(-\frac{\tau}{2}(\ln x_i - \mu)^2\big)\Big] \tau^{a-1} \exp\left(-b\tau\right) \\ &= \tau^{\frac{N}{2}+a-1} \exp\Big(-\big[\sum_i \frac{(\ln x_i - \mu)^2}{2} + b\big]\tau\Big) \end{aligned}$$

We can see that the posterior matches a Gamma distribution up to the normalization constant:

$$p(\tau \mid \boldsymbol{x}, \mu, a, b) = \text{Gamma}\big(\tau \mid \frac{N}{2} + a, \sum_i \frac{(\ln x_i - \mu)^2}{2} + b\big)$$

Thus, c) will yield the best approximation.

Problem 19 [2 points] Prove that ELBO, defined as $\mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})}\left[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z})\right]$, is a lower bound to the model evidence $\log p(\boldsymbol{x})$.

$$\begin{aligned} \log p(x) &= \log \int_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}) \\ &= \log \int_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}) \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})} \\ &= \log\left(\mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})}\left[\frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right]\right) \\ &\geq \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right] \\ &= \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})}\left[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z})\right] \end{aligned}$$