Data Analytics and Machine Learning Group
Department of Informatics
Technical University of Munich

Eexam
Place student sticker here

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- ... both next to the code and to the signature field in the attendance check list.

...e Learning

Graded Exercise:	IN2064 / Endterm	**Date:**	Tuesday 16th February, 2021
Examiner:	Prof. Dr. Stephan Günnemann	**Time:**	11:00 – 13:00

Working instructions

- This graded exercise consists of **36 pages** with a total of **10 problems**.
  Please make sure now that you received a complete copy of the graded exercise.
- The total amount of achievable credits in this graded exercise is 48.
- This document is copyrighted and it is **illegal** for you to distribute it or upload it to any third-party websites.
- Do **not** submit the problem descriptions (this document) to TUMexam

Left room from ____________ to ____________ / Early submission at ____________

Problem 1: Probabilistic Inference (Version A) (4 credits)

Imagine a disease spreading across a small village. To make accurate forecasts for the necessary hospital beds, you want to estimate the severity of the disease with the following model. Let s be a measure for the severity of a disease. We assume a priori that s follows a standard normal distribution.

$$s \sim \mathcal{N}(0, 1) \propto \exp\left(-s^2\right)$$

The severity probabilis[illegible]equired days of hospital care t for a patient according to

$$[illegible]\mathrm{xp}(\lambda) = \lambda \exp(-\lambda t) \text{ where } \lambda = s^2.$$

After some time, you w[illegible] 5 data points. The respective patients left the hospital after 3, 7, 3, 2 and 4 days.

Derive an a-posteriori most likely value of the severity s considering these observations. Justify your answer.

Note: You can assume that $s \neq 0$, that is, you can safely divide by s if necessary.

Problem 1: Probabilistic Inference (Version B) (4 credits)

Imagine a disease spreading across a small village. To make accurate forecasts for the necessary hospital beds, you want to estimate the severity of the disease with the following model. Let s be a measure for the severity of a disease. We assume a priori that s follows a standard normal distribution.

$$s \sim \mathcal{N}(0, 1) \propto \exp\left(-s^2\right)$$

The severity probabilistically infl[illegible]ys of hospital care t for a patient according to

$$[illegible] \exp(-\lambda t) \text{ where } \lambda = s^2.$$

After some time, you were able [illegible] points. The respective patients left the hospital after 10, 11 and 5 days.

Derive an a-posteriori most likely value of the severity s considering these observations. Justify your answer.

Note: You can assume that $s \neq 0$, that is, you can safely divide by s if necessary.

0 1 2 3 4

Problem 1: Probabilistic Inference (Version C) (4 credits)

0 1 2 3 4

Imagine a disease spreading across a small village. To make accurate forecasts for the necessary hospital beds, you want to estimate the severity of the disease with the following model. Let s be a measure for the severity of a disease. We assume a priori that s follows a standard normal distribution.

$$s \sim \mathcal{N}(0, 1) \propto \exp\left(-s^2\right)$$

The severity probabilis [illegible] equired days of hospital care t for a patient according to

[illegible] $\mathrm{xp}(\lambda) = \lambda \exp(-\lambda t)$ where $\lambda = s^2$.

After some time, you w [illegible] 4 data points. The respective patients left the hospital after 8, 12, 9 and 6 days.

Derive an a-posteriori most likely value of the severity s considering these observations. Justify your answer.

Note: You can assume that $s \neq 0$, that is, you can safely divide by s if necessary.

Problem 2: k-nearest neighbors (Version A) (4 credits)

Consider a k-nearest neighbor classifier using Euclidean distance on a binary classification task. The prediction for an instance is the *majority* class of its k nearest neighbors. The training set is shown on Figure 4.1.

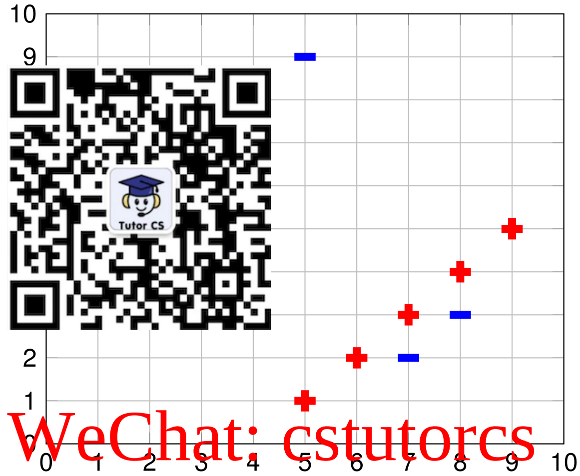


Figure 4.1: A red **+** denotes instances from class 1, and a blue **-** denotes instances from class 2.

a) Specify one value of k that minimizes the leave-one-out cross-validation (LOOCV) error. Please consider only *odd* values of k (e.g. $1, 3, 5, 7, \ldots$) to avoid ties. What is the resulting error, i.e. the number of misclassified data points?

0 1 2

b) Imagine that the training dataset contains one additional point with coordinates $(1, 2)$ and label **+**. Would this change the decision boundary for a 1-nearest neighbor classifier? Why or why not?

0 1

c) If your answer above was *no* what is the *shortest* distance that you need to move $(1, 2)$ such that it changes the decision boundary? If your answer above was *yes* what is the *shortest* distance that you need to move $(1, 2)$ so it does not change the decision boundary?

0 1

Write down the new coordinates after moving and specify the shortest distance.

Problem 2: k-nearest neighbors (Version B) (4 credits)

Consider a k-nearest neighbor classifier using Euclidean distance on a binary classification task. The prediction for an instance is the *majority* class of its k-nearest neighbors. The training set is shown on Figure 5.1.

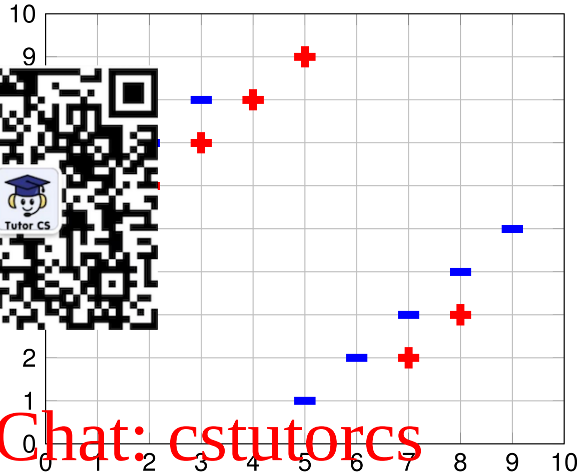


Figure 5.1: A red **+** denotes instances from class 1, and a blue **-** denotes instances from class 2.

0 1 2

a) Specify one value of k that minimizes the leave-one-out cross-validation (LOOCV) error. Please consider only *odd* values of k (e.g. $1, 3, 5, 7, \ldots$) to avoid ties. What is the resulting error, i.e. the number of misclassified data points?

0 1

b) Imagine that the training dataset contains one additional point with coordinates $(1, 2)$ and label **+**. Would this change the decision boundary for a 1-nearest neighbor classifier? Why or why not?

0 1

c) If your answer above was *no* what is the *shortest* distance that you need to move $(1, 2)$ such that it changes the decision boundary? If your answer above was *yes* what is the *shortest* distance that you need to move $(1, 2)$ so it does not change the decision boundary?

Write down the new coordinates after moving and specify the shortest distance.

Problem 2: k-nearest neighbors (Version C) (4 credits)

Consider a k-nearest neighbor classifier using Euclidean distance on a binary classification task. The prediction for an instance is the *majority* class of its k nearest neighbors. The training set is shown on Figure 6.1.

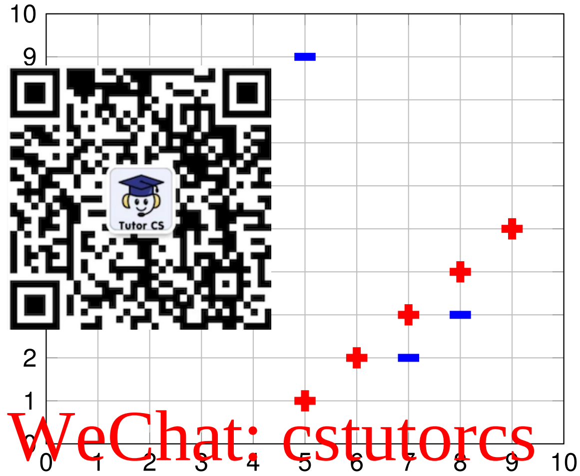


Figure 6.1: A red **+** denotes instances from class 1, and a blue **-** denotes instances from class 2.

a) Specify one value of k that minimizes the leave-one-out cross-validation (LOOCV) error. Please consider only *odd* values of k (e.g. $1, 3, 5, 7, \ldots$) to avoid ties. What is the resulting error, i.e. the number of misclassified data points?

0 1 2

b) Imagine that the training dataset contains one additional point with coordinates $(1, 2)$ and label **-**. Would this change the decision boundary for a 1-nearest neighbor classifier? Why or why not?

0 1

c) If your answer above was *no* what is the *shortest* distance that you need to move $(1, 2)$ such that it changes the decision boundary? If your answer above was *yes* what is the *shortest* distance that you need to move $(1, 2)$ so it does not change the decision boundary?

0 1

Write down the new coordinates after moving and specify the shortest distance.

Problem 2: k-nearest neighbors (Version D) (4 credits)

Consider a k-nearest neighbor classifier using Euclidean distance on a binary classification task. The prediction for an instance is the *majority* class of its k-nearest neighbors. The training set is shown on Figure 7.1.

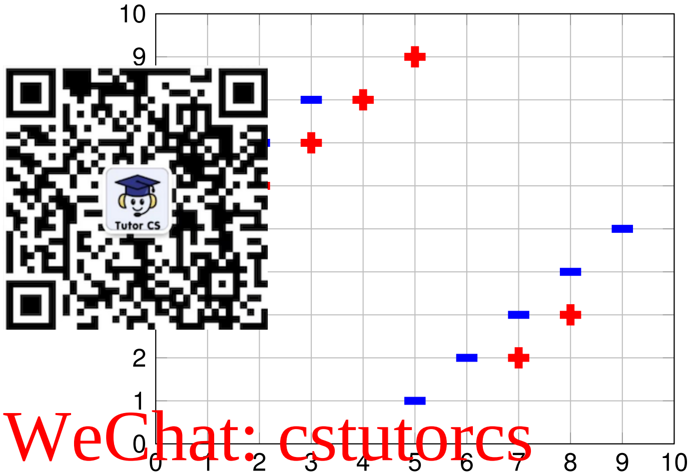


Figure 7.1: A red **+** denotes instances from class 1, and a blue **-** denotes instances from class 2.

0 1 2 a) Specify one value of k that minimizes the leave-one-out cross-validation (LOOCV) error. Please consider only *odd* values of k (e.g. $1, 3, 5, 7, \ldots$) to avoid ties. What is the resulting error, i.e. the number of misclassified data points?

0 1 b) Imagine that the training dataset contains one additional point with coordinates $(1, 2)$ and label **-**. Would this change the decision boundary for a 1-nearest neighbor classifier? Why or why not?

0 1 c) If your answer above was *no* what is the *shortest* distance that you need to move $(1, 2)$ such that it changes the decision boundary? If your answer above was *yes* what is the *shortest* distance that you need to move $(1, 2)$ so it does not change the decision boundary?

Write down the new coordinates after moving and specify the shortest distance.

Problem 3: Linear Regression (Version A) (6 credits)

Consider a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, with $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$, $y^{(i)} \in \mathbb{R}$ and centered features so that $\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)} = \boldsymbol{0}$. During preprocessing, we absorb the bias as a constant feature to produce the transformed dataset $\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}^{(i)}, \widetilde{y}^{(i)})\}_{i=1}^{N}$ where we map each $(\boldsymbol{x}^{(i)}, y^{(i)})$ to

$$\widetilde{\boldsymbol{x}}^{(i)} = \begin{pmatrix} \boldsymbol{x}^{(i)} \\ 1 \end{pmatrix} \in \mathbb{R}^{D+1} \quad \text{and} \quad \widetilde{y}^{(i)} = y^{(i)}.$$

We want to perform ridge regres[...]n strength λ to find the optimal weight vector $\widetilde{\boldsymbol{w}}^* \in \mathbb{R}^{D+1}$.

Reminder: In ridge regression w[...]ction

$$[...]\widetilde{\boldsymbol{x}}^{(i)} - y^{(i)}\big)^2 + \frac{\lambda}{2}\|\widetilde{\boldsymbol{w}}\|_2^2.$$

a) Derive the closed form solution for the last element of the weight vector $\widetilde{w}^*_{D+1}$ corresponding to the absorbed bias obtained from ridge regression on $\widetilde{\mathcal{D}}$.

0 1 2 3

b) Propose an alternative *preprocessing step*, i.e. define an alternative transformed dataset $\widehat{\mathcal{D}} = \{(\widehat{\boldsymbol{x}}^{(i)}, \widehat{y}^{(i)})\}_{i=1}^{N}$, such that the optimal ridge regression vector $\widehat{\boldsymbol{w}}^* \in \mathbb{R}^D$ on $\widehat{\mathcal{D}}$ is equivalent to $\widetilde{\boldsymbol{w}}^*$ obtained on $\widetilde{\mathcal{D}}$. Justify that ridge regression on your preprocessed data $\widehat{\mathcal{D}}$ finds an optimal $\widehat{\boldsymbol{w}}^* \in \mathbb{R}^D$ that is equal to $\widetilde{\boldsymbol{w}}^*$ in the first D elements, i.e. $\widehat{w}^*_i = \widetilde{w}^*_i$ for $i \in \{1, \dots, D\}$. You do not need to derive the closed-form solution for $\widehat{\boldsymbol{w}}^*$.

0 1 2 3

Problem 3: Linear Regression (Version B) (6 credits)

Consider a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, with $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$, $y^{(i)} \in \mathbb{R}$ and centered features so that $\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)} = \mathbf{0}$. During preprocessing, we absorb the bias as a constant feature to produce the transformed dataset $\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}^{(i)}, \widetilde{y}^{(i)})\}_{i=1}^{N}$ where we map each $(\boldsymbol{x}^{(i)}, y^{(i)})$ to

$$\widetilde{\boldsymbol{x}}^{(i)} = \begin{pmatrix} \boldsymbol{x}^{(i)} \\ 1 \end{pmatrix} \in \mathbb{R}^{D+1} \quad \text{and} \quad \widetilde{y}^{(i)} = y^{(i)}.$$

We want to perform ri[illegible]ularization strength λ to find the optimal weight vector $\widetilde{\boldsymbol{w}}^* \in \mathbb{R}^{D+1}$.

Reminder: In ridge re[illegible]e loss function

$$[illegible] \frac{1}{2}\sum_{i=1}^{N}\left(\widetilde{\boldsymbol{w}}^T\widetilde{\boldsymbol{x}}^{(i)} - y^{(i)}\right)^2 + \frac{\lambda}{2}\|\widetilde{\boldsymbol{w}}\|_2^2.$$

0 1 2 3

a) Derive the closed form solution for the last element of the weight vector $\widetilde{\boldsymbol{w}}^*_{D+1}$ corresponding to the absorbed bias obtained from ridge regression on $\widetilde{\mathcal{D}}$.

0 1 2 3

b) Propose an alternative *preprocessing step*, i.e. define an alternative transformed dataset $\widehat{\mathcal{D}} = \{(\widehat{\boldsymbol{x}}^{(i)}, \widehat{y}^{(i)})\}_{i=1}^{N}$, such that the optimal ridge regression vector $\widehat{\boldsymbol{w}}^* \in \mathbb{R}^D$ on $\widehat{\mathcal{D}}$ is equivalent to $\widetilde{\boldsymbol{w}}^*$ obtained on $\widetilde{\mathcal{D}}$. Justify that ridge regression on your preprocessed data $\widehat{\mathcal{D}}$ finds an optimal $\widehat{\boldsymbol{w}}^* \in \mathbb{R}^D$ that is equal to $\widetilde{\boldsymbol{w}}^*$ in the first D elements, i.e. $\widehat{\boldsymbol{w}}_i^* = \widetilde{\boldsymbol{w}}_i^*$ for $i \in \{1, \dots, D\}$. You do not need to derive the closed-form solution for $\widehat{\boldsymbol{w}}^*$.

Problem 4: Naive Bayes (Version A) (6 credits)

We have collected the dataset shown in the table below with the continuous feature x_1, and the discrete feature x_2 that can take either of the values yes or no. Each data point is labelled as one of three classes 1, 2 or 3 denoted by y.

Table 10.1: Naive Bayes Data (each column is one data point)

(4)	(5)	(6)	(7)
-1.0	3.0	4.0	6.0
yes	no	yes	yes
2	2	3	3

a) Set up a naive Bayes classifier (choose likelihoods and their parameterization) for the data in Table 10.1 using Normal distributions with fixed variance of 1 and Bernoulli distributions. Compute the maximum likelihood estimate of *all* parameters $\boldsymbol{\theta}$ required for naive Bayes.

b) You observe a new data point $\mathbf{x}^{(b)} = \begin{pmatrix}1 & \text{yes}\end{pmatrix}$. Compute the unnormalized posterior over classes $p(y^{(b)} \mid \mathbf{x}^{(b)}, \boldsymbol{\theta})$ for $\mathbf{x}^{(b)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

c) Next, you get another data point $\mathbf{x}^{(d)} = \begin{pmatrix}\text{n/a} & \text{n/a}\end{pmatrix}$, where all values are missing. Compute the unnormalized posterior over classes $p(y^{(d)} \mid \mathbf{x}^{(d)}, \boldsymbol{\theta})$ for $\mathbf{x}^{(d)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

d) Finally, you see a data point $\mathbf{x}^{(c)} = \begin{pmatrix}\text{n/a} & \text{no}\end{pmatrix}$, i.e. the features are only partially known. Compute the unnormalized posterior over classes $p(y^{(c)} \mid \mathbf{x}^{(c)}, \boldsymbol{\theta})$ for $\mathbf{x}^{(c)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

Problem 4: Naive Bayes (Version B) (6 credits)

We have collected the dataset shown in the table below with the continuous feature x_1, and the discrete feature x_2 that can take either of the values yes or no. Each data point is labeled as one of three classes 1, 2 or 3 denoted by y.

Table 11.1: Naive Bayes Data (each column is one data point)

[illegible]	[illegible]	(2)	(3)	(4)	(5)	(6)	(7)
[illegible]	[illegible]	2.0	-2.0	-1.0	3.0	4.0	6.0
[illegible]	[illegible]	no	no	yes	no	yes	yes
[illegible]	[illegible]	1	2	2	2	3	3

0 1 2 3 a) Set up a naive Bayes classifier (choose likelihoods and their parameterization) for the data in Table 11.1 using Normal distributions with fixed variance of 1 and Bernoulli distributions. Compute the maximum likelihood estimate of *all* parameters $\boldsymbol{\theta}$ required for naive Bayes.

0 1 b) You observe a new data point $\boldsymbol{x}^{(b)} = \begin{pmatrix} 2 & \text{yes} \end{pmatrix}$. Compute the unnormalized posterior over classes $\mathrm{p}(y^{(b)} \mid \boldsymbol{x}^{(b)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(b)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

0 1 c) Next, you get another data point $\boldsymbol{x}^{(d)} = \begin{pmatrix} \text{n/a} & \text{n/a} \end{pmatrix}$, where all values are missing. Compute the unnormalized posterior over classes $\mathrm{p}(y^{(d)} \mid \boldsymbol{x}^{(d)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(d)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

0 1 d) Finally, you see a data point $\boldsymbol{x}^{(c)} = \begin{pmatrix} \text{n/a} & \text{no} \end{pmatrix}$, i.e. the features are only partially known. Compute the unnormalized posterior over classes $\mathrm{p}(y^{(c)} \mid \boldsymbol{x}^{(c)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(c)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

Problem 4: Naive Bayes (Version C) (6 credits)

We have collected the dataset shown in the table below with the continuous feature x_1, and the discrete feature x_2 that can take either of the values yes or no. Each data point is labelled as one of three classes 1, 2 or 3 denoted by y.

Table 12.1: Naive Bayes Data (each column is one data point)

[illegible]	[illegible]	[illegible]	[illegible]	(4)	(5)	(6)	(7)
[illegible]	[illegible]	[illegible]	[illegible]	2.0	3.0	3.0	6.0
[illegible]	[illegible]	[illegible]	[illegible]	no	no	yes	yes
[illegible]	[illegible]	[illegible]	[illegible]	2	3	3	3

a) Set up a naive Bayes classifier (choose likelihoods and their parameterization) for the data in Table 12.1 using Normal distributions with fixed variance of 1 and Bernoulli distributions. Compute the maximum likelihood estimate of *all* parameters $\boldsymbol{\theta}$ required for naive Bayes.

0 1 2 3

b) You observe a new data point $\boldsymbol{x}^{(b)} = (\text{yes} \;\; 1)$. Compute the unnormalized posterior over classes $p(y^{(b)} \mid \boldsymbol{x}^{(b)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(b)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

0 1

c) Next, you get another data point $\boldsymbol{x}^{(d)} = (\text{n/a} \;\; \text{n/a})$, where all values are missing. Compute the unnormalized posterior over classes $p(y^{(d)} \mid \boldsymbol{x}^{(d)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(d)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

0 1

d) Finally, you see a data point $\boldsymbol{x}^{(c)} = (\text{n/a} \;\; \text{no})$, i.e. the features are only partially known. Compute the unnormalized posterior over classes $p(y^{(c)} \mid \boldsymbol{x}^{(c)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(c)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

0 1

Problem 4: Naive Bayes (Version D) (6 credits)

We have collected the dataset shown in the table below with the continuous feature x_1, and the discrete feature x_2 that can take either of the values yes or no. Each data point is labeled as one of three classes 1, 2 or 3 denoted by y.

Table 13.1: Naive Bayes Data (each column is one data point)

(2)	(3)	(4)	(5)	(6)	(7)
-1.0	2.0	2.0	3.0	3.0	6.0
no	no	no	no	yes	yes
1	2	2	3	3	3

0 1 2 3

a) Set up a naive Bayes classifier (choose likelihoods and their parameterization) for the data in Table 13.1 using Normal distributions with fixed variance of 1 and Bernoulli distributions. Compute the maximum likelihood estimate of *all* parameters $\boldsymbol{\theta}$ required for naive Bayes.

0 1

b) You observe a new data point $\boldsymbol{x}^{(b)} = (\text{yes} \quad 2)$. Compute the unnormalized posterior over classes $p(y^{(b)} \mid \boldsymbol{x}^{(b)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(b)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

0 1

c) Next, you get another data point $\boldsymbol{x}^{(d)} = (\text{n/a} \quad \text{n/a})$, where all values are missing. Compute the unnormalized posterior over classes $p(y^{(d)} \mid \boldsymbol{x}^{(d)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(d)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

0 1

d) Finally, you see a data point $\boldsymbol{x}^{(c)} = (\text{n/a} \quad \text{no})$, i.e. the features are only partially known. Compute the unnormalized posterior over classes $p(y^{(c)} \mid \boldsymbol{x}^{(c)}, \boldsymbol{\theta})$ for $\boldsymbol{x}^{(c)}$. Simplify as far as possible without evaluating exponential functions, square roots, logarithms, etc. Briefly justify how you arrive at your solution. What is the most likely class for this data point?

Problem 5: Optimization (Version A) (2 credits)

Suppose that $\boldsymbol{a} \in \mathbb{R}^d$ is some fixed vector. We define the function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$f(\boldsymbol{x}) = \exp(e^{\boldsymbol{a}^T \boldsymbol{x}} + e^{-\boldsymbol{a}^T \boldsymbol{x}}).$$

Prove or disprove that f is convex on $\mathbb{R}^d$.

0
1
2

Problem 5: Optimization (Version B) (2 credits)

0
1
2

Suppose that $\boldsymbol{a} \in \mathbb{R}^d$ is some fixed vector. We define the function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$f(\boldsymbol{x}) = \exp(e^{\boldsymbol{a}^T \boldsymbol{x}} + e^{-\boldsymbol{a}^T \boldsymbol{x}}).$$

Prove or disprove that f is convex on $\mathbb{R}^d$.

Problem 5: Optimization (Version C) (2 credits)

Suppose that $\boldsymbol{a} \in \mathbb{R}^d$ is some fixed vector. We define the function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$f(\boldsymbol{x}) = \exp(e^{\boldsymbol{a}^T \boldsymbol{x}} + e^{-\boldsymbol{a}^T \boldsymbol{x}}).$$

0
1
2

Prove or disprove that f is convex on $\mathbb{R}^d$.

Problem 5: Optimization (Version D) (2 credits)

0
1
2

Suppose that $\boldsymbol{a} \in \mathbb{R}^d$ is some fixed vector. We define the function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$f(\boldsymbol{x}) = \exp(e^{\boldsymbol{a}^T \boldsymbol{x}} + e^{-\boldsymbol{a}^T \boldsymbol{x}}).$$

Prove or disprove that f is convex on $\mathbb{R}^d$.

Problem 6: Convolutional Neural Networks (Version A) (3 credits)

Recall that a convolutional layer is defined by the following parameters:

- C_{in} — number of input channels
- C_{out} — number of output channels
- K — kernel (sliding window
- P — padding size
- S — stride

Suppose that $\boldsymbol{x}$ is an image with (in pixels) that has 3 channels, which we can represent as a tensor (three dimensional arra

We passed $\boldsymbol{x}$ through a neural r wo convolutional layers conv1 and conv2 (in this order). As output we obtained a tensor of shape $[16, 16, 8]$ (i.e., 16 channels, height 16 and width 8).

We know that the first convolutional layer conv1 has the following parameters

- $C_{\text{in}} = 3$
- $C_{\text{out}} = 32$
- $K = 3$
- $P = 1$
- $S = 4$

We also know that the second convolutional layer conv2 has kernel size $K = 3$.

What are the remaining parameters $C_{\text{in}}, C_{\text{out}}, P, S$ of the second convolutional layer conv2? Justify your answer.

0 1 2 3

Problem 6: Convolutional Neural Networks (Version B) (3 credits)

Recall that a convolutional layer is defined by the following parameters:

- C_{in} — number of input channels
- C_{out} — number of output channels
- K — kernel (slidi[illegible]
- P — padding siz[illegible]
- S — stride

Suppose that $\boldsymbol{x}$ is an i[illegible]d width 32 (in pixels) that has 3 channels, which we can represent as a tensor (three dimens[illegible]3, 64, 32].

We passed $\boldsymbol{x}$ through [illegible]isting of two convolutional layers conv1 and conv2 (in this order). As output we obtained a tensor of shape [16, 16, 8] (i.e., 16 channels, height 16 and width 8).

We know that the first convolutional layer conv1 has the following parameters

- $C_{\text{in}} = 3$
- $C_{\text{out}} = 32$
- $K = 3$
- $P = 1$
- $S = 1$

We also know that the second convolutional layer conv2 has kernel size $K = 3$.

0 1 2 3

What are the remaining parameters $C_{\text{in}}, C_{\text{out}}, P, S$ of the second convolutional layer conv2? Justify your answer.

Problem 6: Convolutional Neural Networks (Version C) (3 credits)

Recall that a convolutional layer is defined by the following parameters:

- C_{in} — number of input channels
- C_{out} — number of output channels
- K — kernel (sliding window [illegible]
- P — padding size
- S — stride

Suppose that $\boldsymbol{x}$ is an image with [illegible] (in pixels) that has 3 channels, which we can represent as a tensor (three dimensional arra[illegible]

We passed $\boldsymbol{x}$ through a neural [illegible]wo convolutional layers conv1 and conv2 (in this order). As output we obtained a tensor of shape $[16, 16, 8]$ (i.e., 16 channels, height 16 and width 8).

We know that the first convolutional layer conv1 has the following parameters

- $C_{\text{in}} = 3$
- $C_{\text{out}} = 8$
- $K = 3$
- $P = 1$
- $S = 4$

We also know that the second convolutional layer $\mathrm{conv2}$ has kernel size $K = 3$.

What are the remaining parameters $C_{\text{in}}, C_{\text{out}}, P, S$ of the second convolutional layer conv2? Justify your answer.

0 1 2 3

Problem 6: Convolutional Neural Networks (Version D) (3 credits)

Recall that a convolutional layer is defined by the following parameters:

- C_{in} — number of input channels
- C_{out} — number of output channels
- K — kernel (slidi [illegible]
- P — padding siz [illegible]
- S — stride

Suppose that $\boldsymbol{x}$ is an in [illegible] d width 32 (in pixels) that has 3 channels, which we can represent as a tensor (three dimens [illegible] 3, 64, 32].

We passed $\boldsymbol{x}$ through [illegible] isting of two convolutional layers conv1 and conv2 (in this order). As output we obtained a tensor of shape $[16, 16, 8]$ (i.e., 16 channels, height 16 and width 8).

We know that the first convolutional layer conv1 has the following parameters

- $C_{\text{in}} = 3$
- $C_{\text{out}} = 8$
- $K = 3$
- $P = 1$
- $S = 1$

We also know that the second convolutional layer conv2 has kernel size $K = 3$.

0 1 2 3

What are the remaining parameters $C_{\text{in}}, C_{\text{out}}, P, S$ of the second convolutional layer conv2? Justify your answer.

Problem 7: SVMs (Version A) (5 credits)

Consider a soft-margin SVM trained on a binary classification task with some fixed and finite penalty C, i.e. $0 < C < \infty$. Assume that the training set contains at least two instances from each class. After training you observe that the optimal slack variables are zero for all instances except for a *single* instance q. Specifically, $\xi_q > 2$ for q and $\xi_i = 0$ for all $i \neq q$. Let $m_{\text{soft}} = \frac{2}{||\boldsymbol{w}_{\text{soft}}||}$ be the optimal margin where $\boldsymbol{w}_{\text{soft}}$ are the optimal weights.

a) Now you *remove* the instance [illegible] you train a new *hard-margin* SVM. Let m_{hard} be the resulting optimal margin. Which one of th[illegible] $\leq m_{\text{soft}}$, $m_{\text{hard}} = m_{\text{soft}}$, $m_{\text{hard}} \geq m_{\text{soft}}$. Justify your answer.

0 1 2 3

b) Now instead of removing it, you *relabel* the instance q (if before it was class -1 you set it to class $+1$ and vice versa) and you train a new *hard-margin* SVM. Let m_{hard} be the resulting optimal margin. Which one of the following holds: $m_{\text{hard}} \leq m_{\text{soft}}$, $m_{\text{hard}} = m_{\text{soft}}$, $m_{\text{hard}} \geq m_{\text{soft}}$. Justify your answer.

0 1 2

Problem 7: SVMs (Version B) (5 credits)

Consider a soft-margin SVM trained on a binary classification task with some fixed and finite penalty C, i.e. $0 < C < \infty$. Assume that the training set contains at least two instances from each class. After training you observe that the optimal slack variables are zero for all instances except for a *single* instance q. Specifically, $\xi_q > 2$ for q and $\xi_i = 0$ for all $i \neq q$. Let $m_{\text{soft}} = \frac{2}{||\boldsymbol{w}_{\text{soft}}||}$ be the optimal margin where $\boldsymbol{w}_{\text{soft}}$ are the optimal weights.

a) Now you *remove* the [illegible]taset and you train a new *hard-margin* SVM. Let m_{hard} be the resulting optimal margin. Which [illegible]olds: $m_{\text{hard}} \leq m_{\text{soft}}$, $m_{\text{hard}} = m_{\text{soft}}$, $m_{\text{hard}} \geq m_{\text{soft}}$. Justify your answer.

b) Now instead of removing it, you *relabel* the instance q (if before it was class -1 you set it to class $+1$ and vice versa) and you train a new *hard-margin* SVM. Let m_{hard} be the resulting optimal margin. Which one of the following holds: $m_{\text{hard}} \leq m_{\text{soft}}$, $m_{\text{hard}} = m_{\text{soft}}$, $m_{\text{hard}} \geq m_{\text{soft}}$. Justify your answer.

Problem 8: Low-rank Approximation and Regression (Version A) (6 credits)

Consider a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, with $\mathbf{x}^{(i)} \in \mathbb{R}^D, y^{(i)} \in \mathbb{R}$. Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ be the matrix of features and $\mathbf{y} \in \mathbb{R}^N$ be the vector of regression targets.

We construct the matrix $\mathbf{M} = [\mathbf{X}, \ \mathbf{y}] \in \mathbb{R}^{N \times (D+1)}$ where $\mathbf{y}$ has been appended as an additional column to $\mathbf{X}$. Let $\mathbf{M}' \in \mathbb{R}^{N \times (D+1)}$ be the *best* rank K approximation of $\mathbf{M}$ for some chosen constant K. Define $\mathbf{M}' = [\mathbf{X}', \ \mathbf{y}']$ where $\mathbf{X}'$ is the matrix containing the first D columns of $\mathbf{M}'$ and $\mathbf{y}'$ is the last column of $\mathbf{M}'$.

We will fit a standard linear regr[illegible] as the feature matrix and $\mathbf{y}'$ as the target, i.e. we find the optimal weight vector $\mathbf{w}^* \in \mathbb{R}^D$ a[illegible]*areful: We do regression on* $\mathbf{X}'$ *and* $\mathbf{y}'$ *not on* $\mathbf{X}$ *and* $\mathbf{y}$*!*

a) Consider the case where D [illegible] features are one dimensional and $\mathbf{M}'$ is the the best rank 1 approximation of $\mathbf{M}$. Assumin[illegible]in at least two distinct elements, what is the *training error* achieved by the optimal $\mathbf{w}^*$ and [illegible]er.

0 1 2

b) For the same case as in (a) where $D = 1$ and $K = 1$, write down the optimal $\mathbf{w}^*$ and b^* in terms of the singular value decomposition of $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$, i.e. write down $\mathbf{w}^*$ and b^* as a function of the singular values and singular vectors $\mathbf{u}_i, \sigma_i, \mathbf{v}_i$.

0 1 2 3

c) Consider the general case of D-dimensional ($D > 1$) feature vectors and a rank K approximation of $\mathbf{M}$. Assuming that $\mathbf{X}'$ is full rank, what is the *training error* achieved by the optimal $\mathbf{w}^*$ and b^* as a function of K?

0 1

Problem 8: Low-rank Approximation and Regression (Version B) (6 credits)

Consider a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, with $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$, $y^{(i)} \in \mathbb{R}$. Let $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ be the matrix of features and $\boldsymbol{y} \in \mathbb{R}^N$ be the vector of regression targets.

We construct the matrix $\boldsymbol{M} = [\boldsymbol{X},\ \boldsymbol{y}] \in \mathbb{R}^{N \times (D+1)}$ where $\boldsymbol{y}$ has been appended as an additional column to $\boldsymbol{X}$. Let $\boldsymbol{M}' \in \mathbb{R}^{N \times (D+1)}$ be the *best* rank K approximation of $\boldsymbol{M}$ for some chosen constant K. Define $\boldsymbol{M}' = [\boldsymbol{X}',\ \boldsymbol{y}']$ where $\boldsymbol{X}'$ is the matrix containing the first D columns of $\boldsymbol{M}'$ and $\boldsymbol{y}'$ is the last column of $\boldsymbol{M}'$.

We will fit a standard l[illegible]l using $\boldsymbol{X}'$ as the feature matrix and $\boldsymbol{y}'$ as the target, i.e. we find the optimal weight vector [illegible]$^* \in \mathbb{R}$. *Careful: We do regression on $\boldsymbol{X}'$ and $\boldsymbol{y}'$ not on $\boldsymbol{X}$ and $\boldsymbol{y}$!*

a) Consider the case [illegible], i.e. our features are one dimensional and $\boldsymbol{M}'$ is the the best rank 1 approximation of $\boldsymbol{M}$. [illegible]s $\boldsymbol{X}'$ contain at least two distinct elements, what is the *training error* achieved by the optima[illegible]our answer.

b) For the same case as in (a) where $D = 1$ and $K = 1$, write down the optimal $\boldsymbol{w}^*$ and b^* in terms of the singular value decomposition of $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, i.e. write down $\boldsymbol{w}^*$ and b^* as a function of the singular values and singular vectors $\boldsymbol{u}_i, \sigma_i, \boldsymbol{v}_i$.

c) Consider the general case of D-dimensional ($D > 1$) feature vectors and a rank K approximation of $\boldsymbol{M}$. Assuming that $\boldsymbol{X}'$ is full rank, what is the *training error* achieved by the optimal $\boldsymbol{w}^*$ and b^* as a function of K?

Problem 8: Low-rank Approximation and Regression (Version C) (6 credits)

Consider a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, with $\boldsymbol{x}^{(i)} \in \mathbb{R}^D, y^{(i)} \in \mathbb{R}$. Let $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ be the matrix of features and $\boldsymbol{y} \in \mathbb{R}^N$ be the vector of regression targets.

We construct the matrix $\boldsymbol{M} = [\boldsymbol{X},\ \boldsymbol{y}] \in \mathbb{R}^{N \times (D+1)}$ where $\boldsymbol{y}$ has been appended as an additional column to $\boldsymbol{X}$. Let $\boldsymbol{M}' \in \mathbb{R}^{N \times (D+1)}$ be the *best* rank K approximation of $\boldsymbol{M}$ for some chosen constant K. Define $\boldsymbol{M}' = [\boldsymbol{X}',\ \boldsymbol{y}']$ where $\boldsymbol{X}'$ is the matrix containing the first D columns of $\boldsymbol{M}'$ and $\boldsymbol{y}'$ is the last column of $\boldsymbol{M}'$.

We will fit a standard linear regr[illegible] as the feature matrix and $\boldsymbol{y}'$ as the target, i.e. we find the optimal weight vector $\boldsymbol{w}^* \in \mathbb{R}^D$ a[illegible]*areful: We do regression on $\boldsymbol{X}'$ and $\boldsymbol{y}'$ not on $\boldsymbol{X}$ and $\boldsymbol{y}$!*

a) Consider the case where D [illegible] features are one dimensional and $\boldsymbol{M}'$ is the the best rank 1 approximation of $\boldsymbol{M}$. Assumin[illegible]in at least two distinct elements, what is the *training error* achieved by the optimal $\boldsymbol{w}^*$ and [illegible]er.

b) For the same case as in (a) where $D = 1$ and $K = 1$, write down the optimal $\boldsymbol{w}^*$ and b^* in terms of the singular value decomposition of $\boldsymbol{M} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$, i.e. write down $\boldsymbol{w}^*$ and b^* as a function of the singular values and singular vectors $\boldsymbol{u}_i, \sigma_i, \boldsymbol{v}_i$.

c) Consider the general case of D-dimensional ($D > 1$) feature vectors and a rank K approximation of $\boldsymbol{M}$. Assuming that $\boldsymbol{X}'$ is full rank, what is the *training error* achieved by the optimal $\boldsymbol{w}^*$ and b^* as a function of K?

Problem 8: Low-rank Approximation and Regression (Version D) (6 credits)

Consider a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, with $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$, $y^{(i)} \in \mathbb{R}$. Let $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ be the matrix of features and $\boldsymbol{y} \in \mathbb{R}^N$ be the vector of regression targets.

We construct the matrix $\boldsymbol{M} = [\boldsymbol{X},\ \boldsymbol{y}] \in \mathbb{R}^{N \times (D+1)}$ where $\boldsymbol{y}$ has been appended as an additional column to $\boldsymbol{X}$. Let $\boldsymbol{M}' \in \mathbb{R}^{N \times (D+1)}$ be the *best* rank K approximation of $\boldsymbol{M}$ for some chosen constant K. Define $\boldsymbol{M}' = [\boldsymbol{X}',\ \boldsymbol{y}']$ where $\boldsymbol{X}'$ is the matrix containing the first D columns of $\boldsymbol{M}'$ and $\boldsymbol{y}'$ is the last column of $\boldsymbol{M}'$.

We will fit a standard l[illegible]l using $\boldsymbol{X}'$ as the feature matrix and $\boldsymbol{y}'$ as the target, i.e. we find the optimal weight vector [illegible]$^* \in \mathbb{R}$. *Careful: We do regression on $\boldsymbol{X}'$ and $\boldsymbol{y}'$ not on $\boldsymbol{X}$ and $\boldsymbol{y}$!*

0 a) Consider the case [illegible], i.e. our features are one dimensional and $\boldsymbol{M}'$ is the the best rank
1 1 approximation of $\boldsymbol{M}$. [illegible]s $\boldsymbol{X}'$ contain at least two distinct elements, what is the *training error*
achieved by the optim[illegible]our answer.
2

0 b) For the same case as in (a) where $D = 1$ and $K = 1$, write down the optimal $\boldsymbol{w}^*$ and b^* in terms of the singular
1 value decomposition of $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, i.e. write down $\boldsymbol{w}^*$ and b^* as a function of the singular values and singular vectors $\boldsymbol{u}_i, \sigma_i, \boldsymbol{v}_i$.
2
3

0 c) Consider the general case of D-dimensional ($D > 1$) feature vectors and a rank K approximation of $\boldsymbol{M}$. Assuming
1 that $\boldsymbol{X}'$ is full rank, what is the *training error* achieved by the optimal $\boldsymbol{w}^*$ and b^* as a function of K?

Problem 9: K-means (Version A) (6 credits)

Consider a variant of K-means that uses the (squared) Mahalanobis distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \tag{1}$$

with the covariance matrix $\mathbf{\Sigma}$, instead of the L_2 distance.

Hint: $\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{a}} = (\mathbf{X} + \mathbf{X}^T)\mathbf{a}^T$

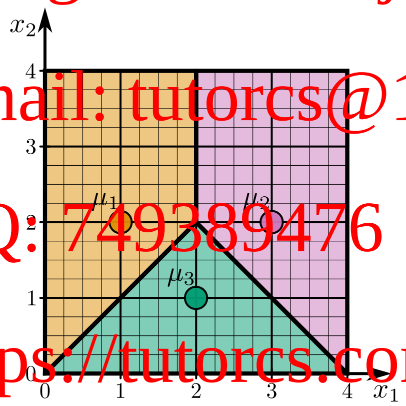
a) Derive the K-means cluster a[illegible]d updates for the (squared) Mahalanobis distance.

0 1 2

b) Instead of considering data samples we will now consider how the cluster centroids $\boldsymbol{\mu}_i$ (which are given and fixed) partition the space into K parts. Every part is defined as the space of all points $\mathbf{x} \in \mathbb{R}^d$ that would be assigned to the corresponding cluster k using the (squared) Mahalanobis distance.

0 1 2

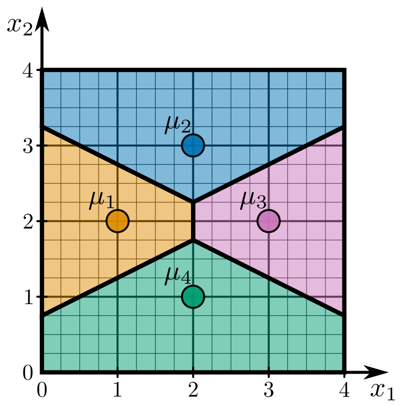
Consider the following result using the (squared) Mahalanobis distance with $\mathbf{\Sigma}^{-1}$, where the parts are denoted by the colors orange, pink, and green. Specify a possible $\mathbf{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.



Hint: Consider the boundary between two partitions. Furthermore, the inverse of a symmetric matrix is also symmetric, and the inverse of a positive/negative (semi-)definite matrix is also positive/negative (semi-)definite.

c) Consider the following result using the (squared) Mahalanobis distance with $\mathbf{\Sigma}^{-1}$, where the parts are denoted by the colors blue, orange, pink, and green. Specify a possible $\mathbf{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.

0 1 2



Problem 9: K-means (Version B) (6 credits)

Consider a variant of K-means that uses the (squared) Mahalanobis distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \tag{2}$$

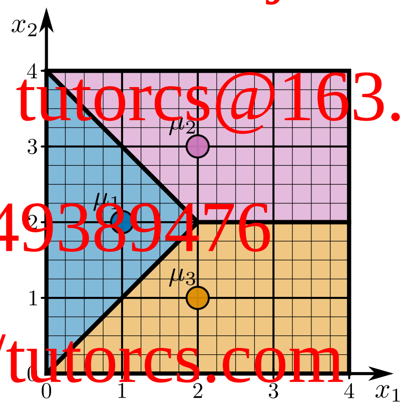
with the covariance matrix $\boldsymbol{\Sigma}$, instead of the L_2 distance.

Hint: $\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{a}} = (\mathbf{X} + \mathbf{X}^T)\mathbf{a}$

a) Derive the K-means [illegible]nd centroid updates for the (squared) Mahalanobis distance.

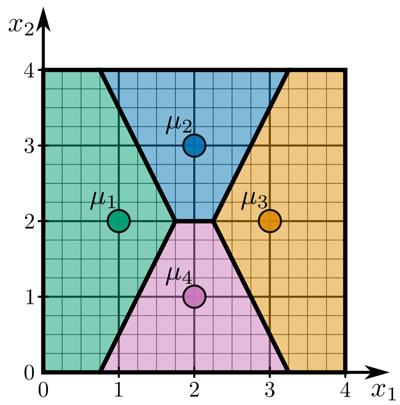
b) Instead of considering data samples we will now consider how the cluster centroids $\boldsymbol{\mu}_i$ (which are given and fixed) partition the space into K parts. Every part is defined as the space of all points $\mathbf{x} \in \mathbb{R}^d$ that would be assigned to the corresponding cluster k using the (squared) Mahalanobis distance.

Consider the following result using the (squared) Mahalanobis distance with $\boldsymbol{\Sigma}^{-1}$, where the parts are denoted by the colors blue, orange, and pink. Specify a possible $\boldsymbol{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.



Hint: Consider the boundary between two partitions. Furthermore, the inverse of a symmetric matrix is also symmetric, and the inverse of a positive/negative (semi-)definite matrix is also positive/negative (semi-)definite.

c) Consider the following result using the (squared) Mahalanobis distance with $\boldsymbol{\Sigma}^{-1}$, where the parts are denoted by the colors blue, orange, pink, and green. Specify a possible $\boldsymbol{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.



Problem 9: K-means (Version C) (6 credits)

Consider a variant of K-means that uses the (squared) Mahalanobis distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \tag{3}$$

with the covariance matrix $\boldsymbol{\Sigma}$, instead of the L_2 distance.

Hint: $\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{a}} = (\mathbf{X} + \mathbf{X}^T)\mathbf{a}^T$

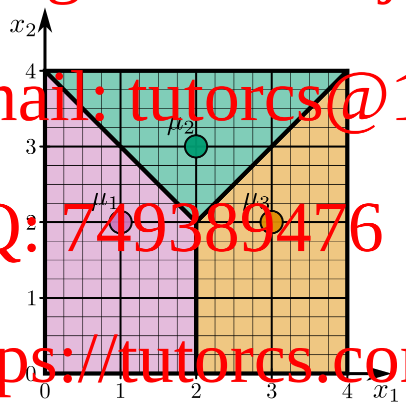
a) Derive the K-means cluster a[illegible]d updates for the (squared) Mahalanobis distance.

0 1 2

b) Instead of considering data samples we will now consider how the cluster centroids $\boldsymbol{\mu}_i$ (which are given and fixed) partition the space into K parts. Every part is defined as the space of all points $\mathbf{x} \in \mathbb{R}^d$ that would be assigned to the corresponding cluster k using the (squared) Mahalanobis distance.

0 1 2

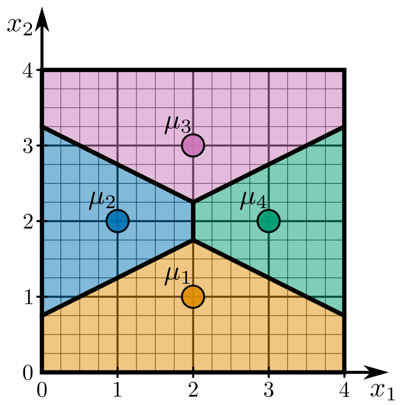
Consider the following result using the (squared) Mahalanobis distance with $\boldsymbol{\Sigma}^{-1}$, where the parts are denoted by the colors green, orange, and pink. Specify a possible $\boldsymbol{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.



Hint: Consider the boundary between two partitions. Furthermore, the inverse of a symmetric matrix is also symmetric, and the inverse of a positive/negative (semi-)definite matrix is also positive/negative (semi-)definite.

c) Consider the following result using the (squared) Mahalanobis distance with $\boldsymbol{\Sigma}^{-1}$, where the parts are denoted by the colors blue, orange, pink, and green. Specify a possible $\boldsymbol{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.

0 1 2



Problem 9: K-means (Version D) (6 credits)

Consider a variant of K-means that uses the (squared) Mahalanobis distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \tag{4}$$

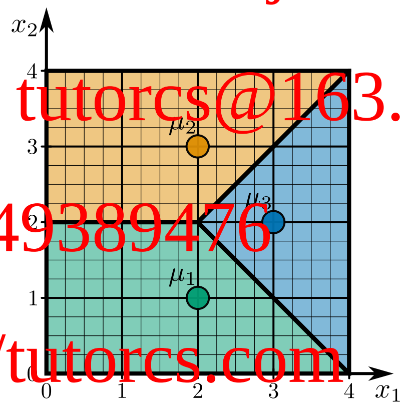
with the covariance matrix $\mathbf{\Sigma}$, instead of the L_2 distance.

Hint: $\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{a}} = (\mathbf{X} + \mathbf{X}^T)$[illegible]

a) Derive the K-means [illegible] nd centroid updates for the (squared) Mahalanobis distance.

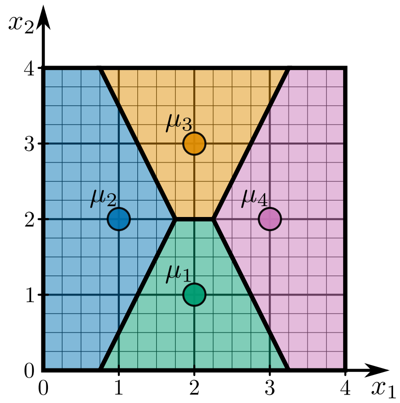
b) Instead of considering data samples we will now consider how the cluster centroids $\boldsymbol{\mu}_i$ (which are given and fixed) partition the space into K parts. Every part is defined as the space of all points $\mathbf{x} \in \mathbb{R}^d$ that would be assigned to the corresponding cluster k using the (squared) Mahalanobis distance.

Consider the following result using the (squared) Mahalanobis distance with $\mathbf{\Sigma}^{-1}$, where the parts are denoted by the colors orange, blue, and green. Specify a possible $\mathbf{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.



Hint: Consider the boundary between two partitions. Furthermore, the inverse of a symmetric matrix is also symmetric, and the inverse of a positive/negative (semi-)definite matrix is also positive/negative (semi-)definite.

c) Consider the following result using the (squared) Mahalanobis distance with $\mathbf{\Sigma}^{-1}$, where the parts are denoted by the colors blue, orange, pink, and green. Specify a possible $\mathbf{\Sigma}^{-1}$ that leads to this partitioning. Justify your answer.



Problem 10: Differential Privacy (Version A) (6 credits)

You are given a dataset with n instances $\{x_1, \dots, x_n\}$, with $x_i \in \mathcal{X}$. The instances are randomly split into disjoint groups $G_1, G_2, \dots G_m$, each of size $\frac{n}{m}$ (assume that m divides n, i.e. $\frac{n}{m}$ is an integer).

a) First you apply an *arbitrary* function $f : \mathcal{X}^{\frac{n}{m}} \to [a, b]$ (where a and b are given constants) to each of the groups, i.e. you compute $g_1 = f(G_1), g_2 = f(G$ [illegible] you compute the final output by aggregating the per-group outputs by computing mean$(g_1,$ [illegible] al Δ_1 sensitivity of the function $f' :=$ mean$(f(G_1), \dots, f(G_m))$?

0 1 2 3

b) How does the global sensitivity of f' change if we increase n keeping m fixed? How does the global sensitivity of f' change if we increase m keeping n fixed?

0 1 2

c) Can you make the function f' differentially private for any function f? If yes, specify the noise distribution from which we have to sample to obtain an ϵ-DP private mechanism. If no, why not?

0 1

Problem 10: Differential Privacy (Version B) (6 credits)

You are given a dataset with n instances $\{x_1, \dots, x_n\}$, with $x_i \in \mathcal{X}$. The instances are randomly split into disjoint groups $G_1, G_2, \dots G_m$, each of size $\frac{n}{m}$ (assume that m divides n, i.e. $\frac{n}{m}$ is an integer).

0 1 2 3

a) First you apply an *arbitrary* function $f : \mathcal{X}^{\frac{n}{m}} \rightarrow [a, b]$ (where a and b are given constants) to each of the groups, i.e. you comp[illegible]), ..., $g_m = f(G_m)$. Then you compute the final output by aggregating the per-group outputs [illegible]n$(g_1, \dots, g_m)$. Derive the global Δ_1 sensitivity of the function $f' :=$ median$(f(G_1), \dots, f(G_m))$[illegible]

0 1 2

b) How does the global sensitivity of f' change if we increase n keeping m fixed? How does the global sensitivity of f' change if we increase m keeping n fixed?

0 1

c) Can you make the function f' differentially private for any function f? If yes, specify the noise distribution from which we have to sample to obtain an ϵ-DP private mechanism. If no, why not?

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.