Data Analytics and Machine Learning
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.
- This number is printed both next to the code and to the signature field in the attendance check list.

Maschinelles Lernen

Exam:	IN2064 / Endterm	**Date:**	Thursday 17$^{\text{th}}$ February, 2022
Examiner:	Prof. Dr. Stephan Günnemann	**Time:**	17:00 – 19:00

Working instructions

- This graded exercise consists of **52 pages** with a total of **11** problems and four versions of each problem.
  Please make sure now that you received a complete copy of the graded exercise.
- Use the problem versions specified in your personalized submission sheet on TUMExam. Different problems may have different versions: e.g. Problem 1 (Version A), Problem 5 (Version C), etc. If you solve the wrong version you get **zero** points.
- The total amount of achievable credits in this graded exercise is 96.
- 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
- You can ignore the “student sticker” box above.

Problem 1: Probabilistic inference (Version A) (10 credits)

Consider the the following probabilistic model:

$$\mathbb{P}(\theta \mid \lambda, \alpha) = \begin{cases} \frac{\alpha\lambda^\alpha}{\theta^{\alpha+1}} & \text{if } \lambda \le \theta \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{P}(x \mid \theta) = \begin{cases} \frac{1}{\theta} & \text{if } 0 \le x \le \theta \\ 0 & \text{otherwise} \end{cases}$$

with $\lambda > 0, \alpha > 0$ and a set of observations $\mathcal{D} = \{x_1, ..., x_N\}$ consisting of N samples $x_i \in \mathbb{R}_+$ generated from the above probabilistic model.

0
1
2
3
4
5
6
7
8
9
10

Derive the posterior distribution $\mathbb{P}(\theta \mid \mathcal{D}, \lambda, \alpha)$.

$$p(\theta|D,\lambda,\alpha) = \frac{p(D|\theta)\,p(\theta|\lambda,\alpha)}{p(D)}$$

$$\propto p(D|\theta)\,p(\theta|\lambda,\alpha)$$

$$\propto \prod_{i=1}^{N} p(x_i|\theta) \cdot p(\theta|\lambda,\alpha) = \left(\frac{1}{\theta}\right)^N \cdot \frac{\alpha\lambda^\alpha}{\theta^{\alpha+1}} = \frac{\alpha\lambda^\alpha}{\theta^{N+\alpha+1}}$$

$$-\ln p(\theta|D,\lambda,\alpha) \propto -\sum \ln p(x_i|\theta) + \ln p(\theta|\lambda,\alpha)$$

$$\propto -\sum \ln \frac{1}{\theta} \quad + \ln \frac{\alpha\lambda^\alpha}{\theta^{\alpha+1}}$$

$$\propto N \cdot \ln\theta - \left[\ln \alpha\lambda^\alpha - \ln \theta^{\alpha+1}\right]$$

$$\propto N\ln\theta - \alpha\ln\alpha\lambda - (\alpha+1)\ln\theta$$

$$\theta_{MLE} = \underset{\theta}{\text{minimize}} -\ln p(\theta|D,\lambda,\alpha) = \underset{\theta}{\text{argmin}}\; n\ln\theta - \alpha\ln\alpha\lambda - (\alpha+1)\ln\theta.$$

$$\frac{\partial -\ln p(\theta|D,\lambda,\alpha)}{\partial\theta} = \frac{N}{\theta} - \frac{\alpha+1}{\theta} \stackrel{!}{=} 0$$

$$N\theta = (\alpha+1)\theta$$

Problem 1: Probabilistic inference (Version B) **(10 credits)**

Consider the the following probabilistic model:

$$\mathbb{P}(\theta \mid \lambda, \alpha) = \begin{cases} \frac{\alpha\lambda^\alpha}{\theta^{\alpha+1}} & \text{if } \lambda \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{P}(x \mid \theta) = \begin{cases} \frac{1}{\theta} & \text{if } 0 \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

with $\lambda > 0, \alpha > 0$ and a set of observations $\mathcal{D} = \{x_1, ..., x_N\}$ consisting of N samples $x_i \in \mathbb{R}_+$ generated from the above probabilistic model.

Derive the posterior distribution $\mathbb{P}(\theta \mid \mathcal{D}, \lambda, \alpha)$.

0 1 2 3 4 5 6 7 8 9 10

Problem 1: Probabilistic inference (Version C) **(10 credits)**

Consider the the following probabilistic model:

$$\mathbb{P}(\theta \mid \lambda, \alpha) = \begin{cases} \frac{\alpha\lambda^\alpha}{\theta^{\alpha+1}} & \text{if } \lambda \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{P}(x \mid \theta) = \begin{cases} \frac{1}{\theta} & \text{if } 0 \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

with $\lambda > 0, \alpha > 0$ and a set of observations $\mathcal{D} = \{x_1, ..., x_N\}$ consisting of N samples $x_i \in \mathbb{R}_+$ generated from the above probabilistic model.

0 Derive the posterior distribution $\mathbb{P}(\theta \mid \mathcal{D}, \lambda, \alpha)$.

1
2
3
4
5
6
7
8
9
10

Problem 1: Probabilistic inference (Version D) **(10 credits)**

Consider the the following probabilistic model:

$$\mathbb{P}(\theta \mid \lambda, \alpha) = \begin{cases} \frac{\alpha \lambda^{\alpha}}{\theta^{\alpha+1}} & \text{if } \lambda \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{P}(x \mid \theta) = \begin{cases} \frac{1}{\theta} & \text{if } 0 \leq x \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

with $\lambda > 0, \alpha > 0$ and a set of observations $\mathcal{D} = \{x_1, ..., x_N\}$ consisting of N samples $x_i \in \mathbb{R}_+$ generated from the above probabilistic model.

Derive the posterior distribution $\mathbb{P}(\theta \mid \mathcal{D}, \lambda, \alpha)$.

0 1 2 3 4 5 6 7 8 9 10

Problem 2: Linear regression (Version A) (8 credits)

We want to perform regression on a dataset consisting of N samples $\mathbf{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \mathbb{R}^N$).
Assume that we have fitted a linear regression model and obtained the optimal weight vector $\mathbf{w}^* \in \mathbb{R}^D$ as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_i - y_i)^2.$$

Note that there is no bias term.
Now, assume that we normalize the target variables to have a variance of 1, i.e. $\mathbf{y}_{\text{new}} = \frac{1}{\sigma} \cdot \mathbf{y}$ with $\sigma = Var(\mathbf{y})$, where $Var(\mathbf{y})$ is the sample variance of $\mathbf{y}$.

0 1 2 3 4 5 6 7 8

Find the data matrix $\mathbf{X}_{\text{new}} \in \mathbb{R}^{N \times D}$ such that the solution to the new problem:

$$\mathbf{w}^*_{\text{new}} = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_{new,i} - y_{new,i})^2$$

will be the same as the solution to the previous problem i.e. $\mathbf{w}^*_{\text{new}} = \mathbf{w}^*$. Justify your answer.
Note: $\mathbf{x}_{new,i}$ is row i of $\mathbf{X}_{\text{new}}$, represented as a column vector.

If: $w^*_{new} = w^*$.

then $\cancel{\frac{1}{2}}(X'w - Y')^T(X'w - y') = \cancel{\frac{1}{2}}(Xw - y)^T(Xw - y)$

$w^T X'^T X' w - \frac{2}{\sigma} w^T X' \cdot y + \frac{1}{\sigma^2} y^T y = w^T X^T X w - 2 w^T X y + y^T y$

$y = XW = \sigma \cdot y_{new} = \sigma \cdot X_{new} \cdot W_{new}$

$X \cdot \cancel{W} = \sigma \cdot X_{new} \cdot \cancel{W_{new}}$

$\frac{1}{\sigma} X = X_{new}$

Problem 2: Linear regression (Version B) (8 credits)

We want to perform regression on a dataset consisting of N samples $\mathbf{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \mathbb{R}^N$).
Assume that we have fitted a linear regression model and obtained the optimal weight vector $\mathbf{w}^* \in \mathbb{R}^D$ as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_i - y_i)^2.$$

Note that there is no bias term.
Now, assume that we normalize the target variables to have a variance of 1, i.e. $\mathbf{y}_{\text{new}} = \frac{1}{\sigma} \cdot \mathbf{y}$ with $\sigma = Var(\mathbf{y})$, where $Var(\mathbf{y})$ is the sample variance of $\mathbf{y}$.

Find the data matrix $\mathbf{X}_{\text{new}} \in \mathbb{R}^{N \times D}$ such that the solution to the new problem:

$$\mathbf{w}^*_{\text{new}} = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_{new,i} - y_{new,i})^2$$

will be the same as the solution to the previous problem i.e. $\mathbf{w}^*_{\text{new}} = \mathbf{w}^*$. Justify your answer.
Note: $\mathbf{x}_{new,i}$ is row i of $\mathbf{X}_{\text{new}}$, represented as a column vector.

0 1 2 3 4 5 6 7 8

Problem 2: Linear regression (Version C) (8 credits)

We want to perform regression on a dataset consisting of N samples $\mathbf{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \mathbb{R}^N$).
Assume that we have fitted a linear regression model and obtained the optimal weight vector $\mathbf{w}^* \in \mathbb{R}^D$ as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_i - y_i)^2.$$

Note that there is no bias term.
Now, assume that we normalize the target variables to have a variance of 1, i.e. $\mathbf{y}_{\text{new}} = \frac{1}{\sigma} \cdot \mathbf{y}$ with $\sigma = Var(\mathbf{y})$, where $Var(\mathbf{y})$ is the sample variance of $\mathbf{y}$.

0 1 2 3 4 5 6 7 8

Find the data matrix $\mathbf{X}_{\text{new}} \in \mathbb{R}^{N \times D}$ such that the solution to the new problem:

$$\mathbf{w}^*_{\text{new}} = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_{new,i} - y_{new,i})^2$$

will be the same as the solution to the previous problem i.e. $\mathbf{w}^*_{\text{new}} = \mathbf{w}^*$. Justify your answer.
Note: $\mathbf{x}_{new,i}$ is row i of $\mathbf{X}_{\text{new}}$, represented as a column vector.

Problem 2: Linear regression (Version D) (8 credits)

We want to perform regression on a dataset consisting of N samples $\mathbf{x}_i \in \mathbb{R}^D$ with corresponding targets $y_i \in \mathbb{R}$ (represented compactly as $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \mathbb{R}^N$).
Assume that we have fitted a linear regression model and obtained the optimal weight vector $\mathbf{w}^* \in \mathbb{R}^D$ as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_i - y_i)^2.$$

Note that there is no bias term.
Now, assume that we normalize the target variables to have a variance of 1, i.e. $\mathbf{y}_{\text{new}} = \frac{1}{\sigma} \cdot \mathbf{y}$ with $\sigma = Var(\mathbf{y})$, where $Var(\mathbf{y})$ is the sample variance of $\mathbf{y}$.

Find the data matrix $\mathbf{X}_{\text{new}} \in \mathbb{R}^{N \times D}$ such that the solution to the new problem:

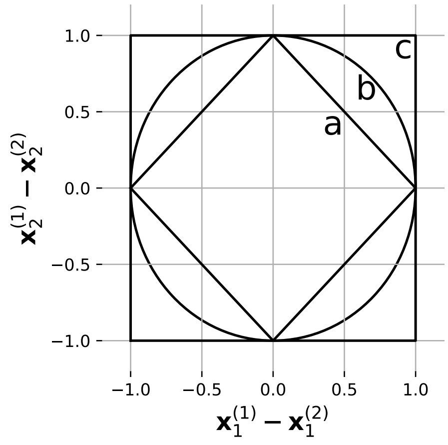
0 1 2 3 4 5 6 7 8

$$\mathbf{w}^*_{\text{new}} = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_{new,i} - y_{new,i})^2$$

will be the same as the solution to the previous problem i.e. $\mathbf{w}^*_{\text{new}} = \mathbf{w}^*$. Justify your answer.
Note: $\mathbf{x}_{new,i}$ is row i of $\mathbf{X}_{\text{new}}$, represented as a column vector.

Problem 3: k-nearest neighbors (Version A) **(3 credits)**

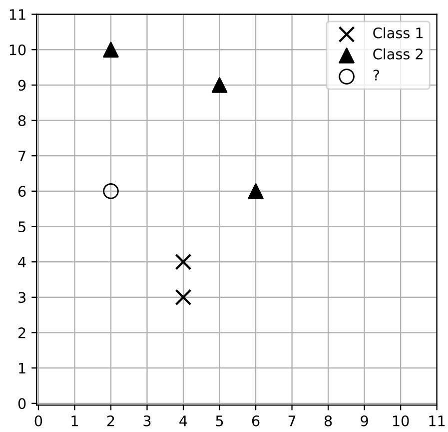
In the following figure we see the unit circles of three distance functions.



0 1 a) Assign each of the following three distance functions its corresponding unit circle (letter a-c) from the figure.

- L_2-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_2 = \sqrt{\sum_i \left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right)^2}$ b
- L_1-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_1 = \sum_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$ a
- L_∞-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty = \max_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$ c

In the following figure we see a two-dimensional dataset with two classes. We would like to classify the point $(2, 6)$ marked with a circle using k-nearest-neighbors with $k = 3$.



4 4 4 5 4 3 4 2 3

0 1 b) What is the predicted class of the point when using the L_1 distance?

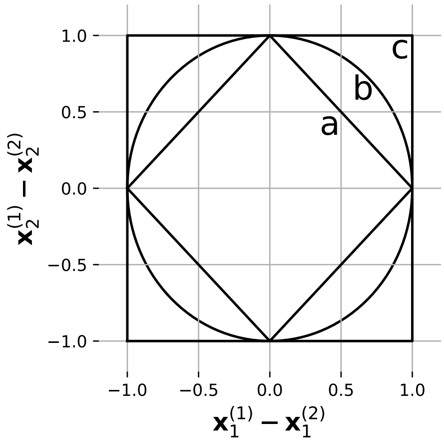
class 2

0 1 c) What is the predicted class of the point when using the L_∞ distance?

class 1

Problem 3: k-nearest neighbors (Version B) (3 credits)

In the following figure we see the unit circles of three distance functions.

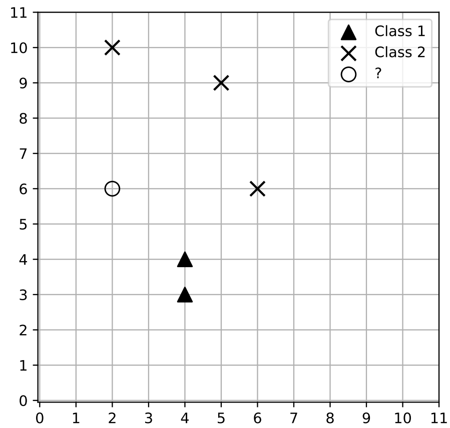


a) Assign each of the following three distance functions its corresponding unit circle (letter a-c) from the figure.

0
1

- L_2-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_2 = \sqrt{\sum_i \left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right)^2}$
- L_1-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_1 = \sum_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$
- L_∞-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty = \max_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$

In the following figure we see a two-dimensional dataset with two classes. We would like to classify the point $(2, 6)$ marked with a circle using k-nearest-neighbors with $k = 3$.



b) What is the predicted class of the point when using the L_1 distance?

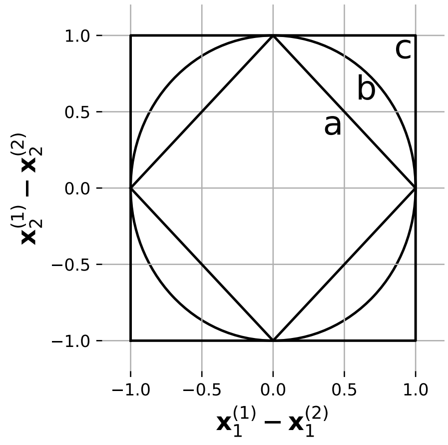
0
1

c) What is the predicted class of the point when using the L_∞ distance?

0
1

Problem 3: k-nearest neighbors (Version C) (3 credits)

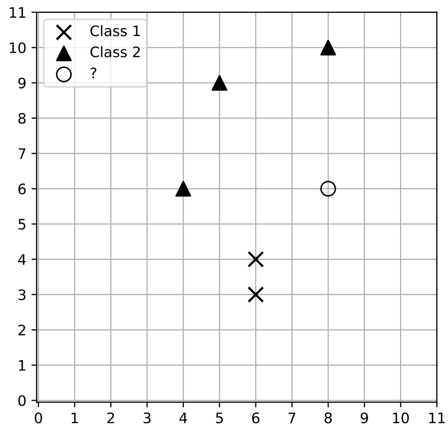
In the following figure we see the unit circles of three distance functions.



a) Assign each of the following three distance functions its corresponding unit circle (letter a-c) from the figure.

- L_2-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_2 = \sqrt{\sum_i \left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right)^2}$
- L_1-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_1 = \sum_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$
- L_∞-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty = \max_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$

In the following figure we see a two-dimensional dataset with two classes. We would like to classify the point $(8, 6)$ marked with a circle using k-nearest-neighbors with $k = 3$.

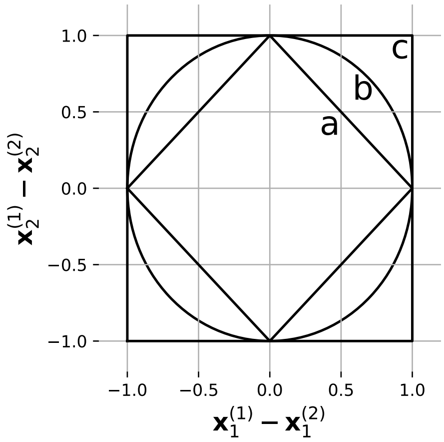


b) What is the predicted class of the point when using the L_1 distance?

c) What is the predicted class of the point when using the L_∞ distance?

Problem 3: k-nearest neighbors (Version D) (3 credits)

In the following figure we see the unit circles of three distance functions.

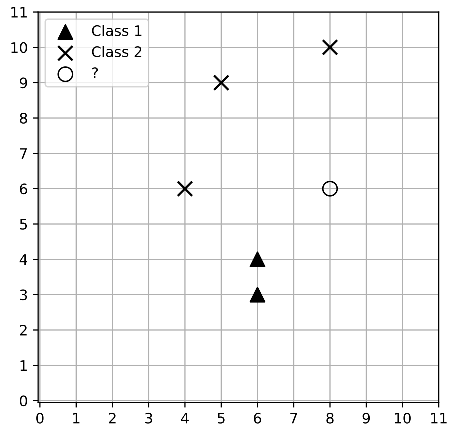


a) Assign each of the following three distance functions its corresponding unit circle (letter a-c) from the figure.

0
1

- L_2-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_2 = \sqrt{\sum_i \left(\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right)^2}$
- L_1-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_1 = \sum_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$
- L_∞-distance: $\|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\|_\infty = \max_i \left|\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}\right|$

In the following figure we see a two-dimensional dataset with two classes. We would like to classify the point $(8, 6)$ marked with a circle using k-nearest-neighbors with $k = 3$.



b) What is the predicted class of the point when using the L_1 distance?

0
1

c) What is the predicted class of the point when using the L_∞ distance?

0
1

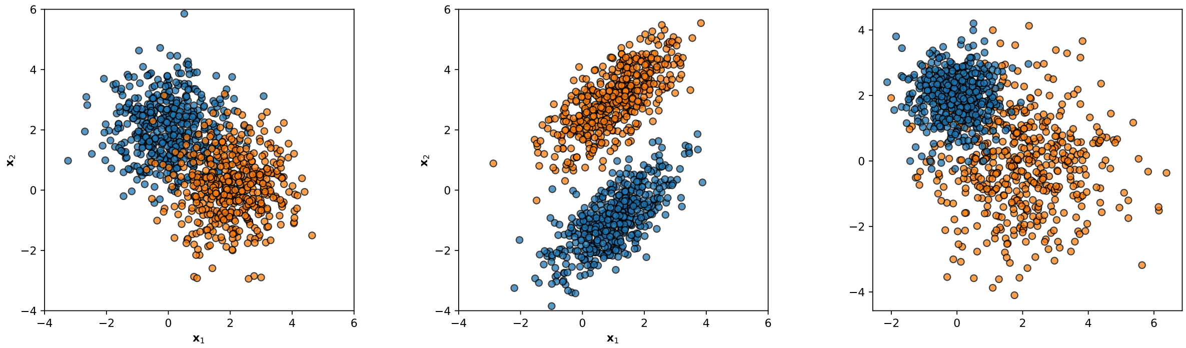
Problem 4: Classification (Version A) (6 credits)

You are given a balanced dataset with two classes, i.e. $p(y = 0) = p(y = 1)$. Assume that the ground truth class conditional distributions are bivariate Gaussian distributions, i.e. $p(\mathbf{x} \mid c) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ with mean $\boldsymbol{\mu}_c$ and covariance $\boldsymbol{\Sigma}_c$ for each class $c \in \{0, 1\}$.
Further assume that we can choose between two models to fit the data:

- Linear Discriminant Analysis with Gaussian class conditional distributions
- Naïve Bayes with Gaussian class conditional distributions

For each of the datasets shown below (a, b, c), choose one of the possible options (1,2,3) and justify your answer:

1. We should use Linear Discriminant Analysis.
2. We should use Naïve Bayes.
3. There is no clear reason to prefer one model over the other.



(a) (b) (c)

0 1 2 a) 2 3 ⇒ Quadratic decision boundary

0 1 2 b) 1 1 ⇒ Linear Decision boundary // same covariance // correlated

0 1 2 c) 3 2

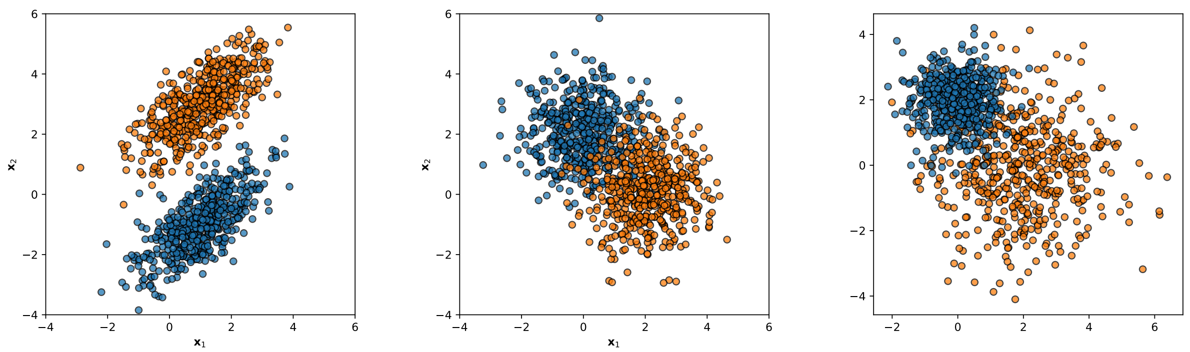
Problem 4: Classification (Version B) (6 credits)

You are given a balanced dataset with two classes, i.e. $p(y = 0) = p(y = 1)$. Assume that the ground truth class conditional distributions are bivariate Gaussian distributions, i.e. $p(\mathbf{x} \mid c) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ with mean $\boldsymbol{\mu}_c$ and covariance $\boldsymbol{\Sigma}_c$ for each class $c \in \{0, 1\}$.
Further assume that we can choose between two models to fit the data:

- Linear Discriminant Analysis with Gaussian class conditional distributions
- Naïve Bayes with Gaussian class conditional distributions

For each of the datasets shown below (a, b, c), choose one of the possible options (1,2,3) and justify your answer:

1. We should use Linear Discriminant Analysis.
2. We should use Naïve Bayes.
3. There is no clear reason to prefer one model over the other.



(a) (b) (c)

a)

0
1
2

b)

0
1
2

c)

0
1
2

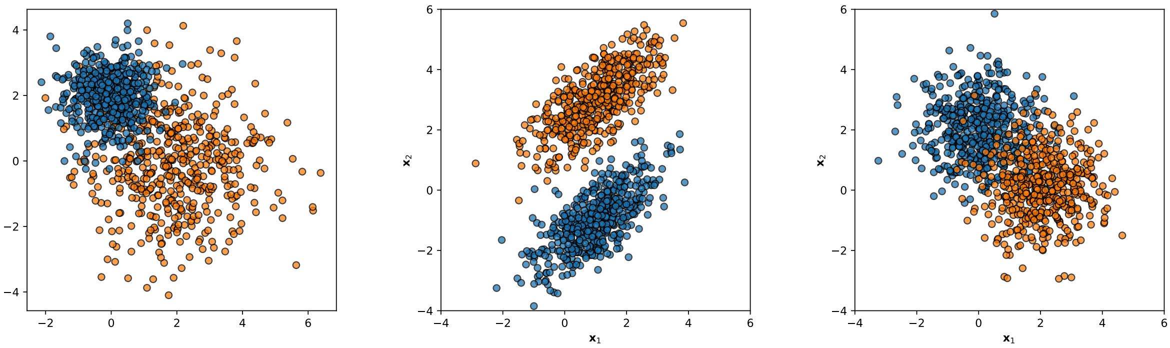
Problem 4: Classification (Version C) **(6 credits)**

You are given a balanced dataset with two classes, i.e. $p(y = 0) = p(y = 1)$. Assume that the ground truth class conditional distributions are bivariate Gaussian distributions, i.e. $p(\mathbf{x} \mid c) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ with mean $\boldsymbol{\mu}_c$ and covariance $\boldsymbol{\Sigma}_c$ for each class $c \in \{0, 1\}$.
Further assume that we can choose between two models to fit the data:

- Linear Discriminant Analysis with Gaussian class conditional distributions
- Naïve Bayes with Gaussian class conditional distributions

For each of the datasets shown below (a, b, c), choose one of the possible options (1,2,3) and justify your answer:

1. We should use Linear Discriminant Analysis.
2. We should use Naïve Bayes.
3. There is no clear reason to prefer one model over the other.



(a) (b) (c)

0 1 2 a)

0 1 2 b)

0 1 2 c)

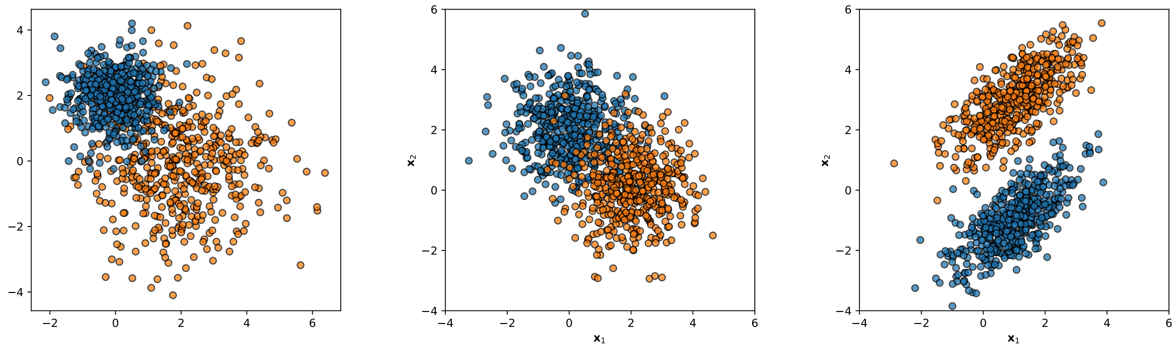
Problem 4: Classification (Version D) (6 credits)

You are given a balanced dataset with two classes, i.e. $p(y = 0) = p(y = 1)$. Assume that the ground truth class conditional distributions are bivariate Gaussian distributions, i.e. $p(\mathbf{x} \mid c) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ with mean $\boldsymbol{\mu}_c$ and covariance $\boldsymbol{\Sigma}_c$ for each class $c \in \{0, 1\}$.
Further assume that we can choose between two models to fit the data:

- Linear Discriminant Analysis with Gaussian class conditional distributions
- Naïve Bayes with Gaussian class conditional distributions

For each of the datasets shown below (a, b, c), choose one of the possible options (1,2,3) and justify your answer:

1. We should use Linear Discriminant Analysis.
2. We should use Naïve Bayes.
3. There is no clear reason to prefer one model over the other.



(a) (b) (c)

a)

0
1
2

b)

0
1
2

c)

0
1
2

Problem 5: Optimization – Convexity (Version A) (10 credits)

Consider the two functions

$$f(\mathbf{x}) = \max_{i=1,\ldots,n} x_i - \min_{i=1,\ldots,n} x_i$$

$$g(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |x_i - \text{median}(\mathbf{x})|$$

with $\mathbf{x} \in \mathbb{R}^n$. You may assume that n is odd.

0 1 2 3 4

a)
Prove or disprove that $f(\mathbf{x})$ is convex in $\mathbf{x}$.

$\max x_i$ is convex rule 2

$-\min x_i = \max x_i$ rule 2.

$\max x_j + \max x_i$ rule 1. convex

0 1 2 3 4 5 6

b)
Prove or disprove that $g(\mathbf{x})$ is convex in $\mathbf{x}$.
Hint: $\text{median}(\mathbf{x}) = \arg\min_{t\in\mathbb{R}} \|\mathbf{x} - t\mathbf{1}\|_1$ *with* $\|\cdot\|_1$ *being the sum over* $\mathbf{x}$*'s elements' absolute values.*

$$\underbrace{|x_1 - t| + |x_i - t| \cdots}_{n}$$

① $Ix - t1 = Ax + b$

$\|Ix - t1\| = \|Ax + b\|$ convex rule 5

$\frac{1}{n} > 0$ rule 3.

$\|x - \text{median}(x)\|_1 = \min_{t\in\mathbb{R}} \|x - t1\|_1$

$g(x) = \min_{t\in\mathbb{R}} f(x, t)$ set $t_1 = \text{argmin } f(x_1, t)$

$t_2 = \text{argmin } f(x_2, t)$

$$\begin{aligned} g(\lambda x_1 + (1-\lambda)x_2) &= \min f(\lambda x_1 + (1-\lambda)x_2, t) \\ &\le f(\lambda x_1 + (1-\lambda)x_2, \lambda t_1 + (1-\lambda)t_2) \\ &\le \lambda f(x_1, t_1) + (1-\lambda) f(x_2, t_2) \\ &\le \lambda g(x_1) + (1-\lambda) g(x_2) \end{aligned}$$

②

Problem 5: Optimization – Convexity (Version B) (10 credits)

Consider the two functions

$$f(\mathbf{x}) = \max_{i=1,\ldots,n} x_i - \min_{i=1,\ldots,n} x_i$$

$$g(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |x_i - \text{median}(\mathbf{x})|$$

with $\mathbf{x} \in \mathbb{R}^n$. You may assume that n is odd.

a)
Prove or disprove that $f(\mathbf{x})$ is convex in $\mathbf{x}$.

b)
Prove or disprove that $g(\mathbf{x})$ is convex in $\mathbf{x}$.
Hint: $\text{median}(\mathbf{x}) = \arg\min_{t\in\mathbb{R}} \|\mathbf{x} - t\mathbf{1}\|_1$ *with* $\|\cdot\|_1$ *being the sum over* $\mathbf{x}$*'s elements' absolute values.*

Problem 5: Optimization – Convexity (Version C) **(10 credits)**

Consider the two functions

$$f(\mathbf{x}) = \max_{i=1,\dots,n} x_i - \min_{i=1,\dots,n} x_i$$

$$g(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |x_i - \mathrm{median}(\mathbf{x})|$$

with $\mathbf{x} \in \mathbb{R}^n$. You may assume that n is odd.

0 1 2 3 4

a)
Prove or disprove that $f(\mathbf{x})$ is convex in $\mathbf{x}$.

0 1 2 3 4 5 6

b)
Prove or disprove that $g(\mathbf{x})$ is convex in $\mathbf{x}$.
Hint: $\mathrm{median}(\mathbf{x}) = \arg\min_{t\in\mathbb{R}} \|\mathbf{x} - t\mathbf{1}\|_1$ *with* $\|\cdot\|_1$ *being the sum over* $\mathbf{x}$*'s elements' absolute values.*

Problem 5: Optimization – Convexity (Version D) (10 credits)

Consider the two functions

$$f(\mathbf{x}) = \max_{i=1,\ldots,n} x_i - \min_{i=1,\ldots,n} x_i$$

$$g(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |x_i - \operatorname{median}(\mathbf{x})|$$

with $\mathbf{x} \in \mathbb{R}^n$. You may assume that n is odd.

a)
Prove or disprove that $f(\mathbf{x})$ is convex in $\mathbf{x}$.

☐	0
☐	1
☐	2
☐	3
☐	4

b)
Prove or disprove that $g(\mathbf{x})$ is convex in $\mathbf{x}$.
Hint: $\operatorname{median}(\mathbf{x}) = \arg\min_{t\in\mathbb{R}} \|\mathbf{x} - t\mathbf{1}\|_1$ *with* $\|\cdot\|_1$ *being the sum over* $\mathbf{x}$*'s elements' absolute values.*

☐	0
☐	1
☐	2
☐	3
☐	4
☐	5
☐	6

Problem 6: Deep learning (Version A) (8 credits)

Suppose $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ are two vectors. We define the functions $f : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}$, and use them to compute

$$\mathbf{z} = f(\mathbf{x}, \mathbf{y})$$
$$t = g(\mathbf{z}).$$

The code below implements the computation of f and g, as well as its gradients using backpropagation. Your task is to complete the missing code fragments.

NOTE: The code is given in Python but you can write the solution in pseudocode as long as it is clear and unambiguous, making sure that the return values have correct shapes.

```
import numpy as np

class F:
    def forward(self, x, y):
        self.cache = (x, y)
        ##################################################
        # MISSING CODE FRAGMENT #1
        ##################################################
        return out

    def backward(self, d_out):
        # x, y are arrays of shape (N,)
        x, y = self.cache
        d_x = np.sin(y) * d_out
        d_y = x * np.cos(y) * d_out
        return d_x, d_y

class G:
    def forward(self, z):
        self.cache = z
        out = np.mean(z)
        return out

    def backward(self, d_out):
        # z is an array of shape (N,)
        z = self.cache
        ##################################################
        # MISSING CODE FRAGMENT #2
        ##################################################
        return d_z

# Example usage
f, g = F(), G()
x = np.array([1, 2, 3])
y = np.array([4, 5, 6])

z = f.forward(x, y)
t = g.forward(z)

d_z = g.backward(d_out=1.0)
d_x, d_y = f.backward(d_z)
```

out = $x \cdot \sin(y)$

out = $x \odot$ np.sin(y)

$\frac{1}{N}$ * d_out

a) Complete the `MISSING CODE FRAGMENT #1`.

0
1
2
3
4

b) Complete the `MISSING CODE FRAGMENT #2`.

0
1
2
3
4

Problem 6: Deep learning (Version B) (8 credits)

Suppose $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ are two vectors. We define the functions $f : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}$, and use them to compute

$$\mathbf{z} = f(\mathbf{x}, \mathbf{y})$$
$$t = g(\mathbf{z}).$$

The code below implements the computation of f and g, as well as its gradients using backpropagation. Your task is to complete the missing code fragments.

NOTE: The code is given in Python but you can write the solution in pseudocode as long as it is clear and unambiguous, making sure that the return values have correct shapes.

```python
import numpy as np

class F:
    def forward(self, x, y):
        self.cache = (x, y)
        ##################################################
        # MISSING CODE FRAGMENT #1
        ##################################################
        return out

    def backward(self, d_out):
        # x, y are arrays of shape (N,)
        x, y = self.cache
        d_x = np.exp(x) / np.exp(y) * d_out
        d_y = -d_x
        return d_x, d_y

class G:
    def forward(self, z):
        self.cache = z
        out = np.sum(z)
        return out

    def backward(self, d_out):
        # z is an array of shape (N,)
        z = self.cache
        ##################################################
        # MISSING CODE FRAGMENT #2
        ##################################################
        return d_z

# Example usage
f, g = F(), G()
x = np.array([1, 2, 3])
y = np.array([4, 5, 6])

z = f.forward(x, y)
t = g.forward(z)

d_z = g.backward(d_out=1.0)
d_x, d_y = f.backward(d_z)
```

a) Complete the `MISSING CODE FRAGMENT #1`.

0
1
2
3
4

b) Complete the `MISSING CODE FRAGMENT #2`.

0
1
2
3
4

Problem 6: Deep learning (Version C) (8 credits)

Suppose $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ are two vectors. We define the functions $f : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}$, and use them to compute

$$\mathbf{z} = f(\mathbf{x}, \mathbf{y})$$
$$t = g(\mathbf{z}).$$

The code below implements the computation of f and g, as well as its gradients using backpropagation. Your task is to complete the missing code fragments.

NOTE: The code is given in Python but you can write the solution in pseudocode as long as it is clear and unambiguous, making sure that the return values have correct shapes.

```
import numpy as np

class F:
    def forward(self, x, y):
        self.cache = (x, y)
        ##################################################
        # MISSING CODE FRAGMENT #1
        ##################################################
        return out

    def backward(self, d_out):
        # x, y are arrays of shape (N,)
        x, y = self.cache
        temp = np.cos(x * y) * d_out
        d_x = y * temp
        d_y = x * temp
        return d_x, d_y

class G:
    def forward(self, z):
        self.cache = z
        out = np.prod(z) # Product of array elements
        return out

    def backward(self, d_out):
        # z is an array of shape (N,)
        z = self.cache
        ##################################################
        # MISSING CODE FRAGMENT #2
        ##################################################
        return d_z

# Example usage
f, g = F(), G()
x = np.array([1, 2, 3])
y = np.array([4, 5, 6])

z = f.forward(x, y)
t = g.forward(z)

d_z = g.backward(d_out=1.0)
d_x, d_y = f.backward(d_z)
```

a) Complete the `MISSING CODE FRAGMENT #1`.

0
1
2
3
4

b) Complete the `MISSING CODE FRAGMENT #2`.

0
1
2
3
4

Problem 6: Deep learning (Version D) (8 credits)

Suppose $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^N$ are two vectors. We define the functions $f : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}$, and use them to compute

$$\mathbf{z} = f(\mathbf{x}, \mathbf{y})$$
$$t = g(\mathbf{z}).$$

The code below implements the computation of f and g, as well as its gradients using backpropagation. Your task is to complete the missing code fragments.

NOTE: The code is given in Python but you can write the solution in pseudocode as long as it is clear and unambiguous, making sure that the return values have correct shapes.

```
import numpy as np

class F:
    def forward(self, x, y):
        self.cache = (x, y)
        ##################################################
        # MISSING CODE FRAGMENT #1
        ##################################################
        return out

    def backward(self, d_out):
        # x, y are arrays of shape (N,)
        x, y = self.cache
        d_x = (1 + y) * d_out
        d_y = x * d_out
        return d_x, d_y

class G:
    def forward(self, z):
        self.cache = z
        out = np.dot(z, z) # Dot product
        return out

    def backward(self, d_out):
        # z is an array of shape (N,)
        z = self.cache
        ##################################################
        # MISSING CODE FRAGMENT #2
        ##################################################
        return d_z

# Example usage
f, g = F(), G()
x = np.array([1, 2, 3])
y = np.array([4, 5, 6])

z = f.forward(x, y)
t = g.forward(z)

d_z = g.backward(d_out=1.0)
d_x, d_y = f.backward(d_z)
```

a) Complete the `MISSING CODE FRAGMENT #1`.

☐ 0
☐ 1
☐ 2
☐ 3
☐ 4

b) Complete the `MISSING CODE FRAGMENT #2`.

☐ 0
☐ 1
☐ 2
☐ 3
☐ 4

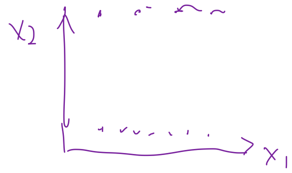
Problem 7: Dimensionality reduction (Version A) (12 credits)

We would like to perform binary classification on a dataset $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \{0, 1\}^N$.
Assume that we first reduce the dimensionality of $\mathbf{X}$ via PCA to obtain the matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times K}$ (where $K < D$).

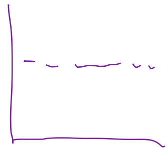
0 1 2 3 4 5 6

a) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is linearly separable. Is the dataset $(\tilde{\mathbf{X}}, \mathbf{y})$ also guaranteed to be linearly separable? If yes, prove it. If not, provide a counterexample.

举反例



linear separable



unlinearseparable.

0 1 2 3 4 5 6

b) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is *NOT* linearly separable. Is the dataset $(\tilde{\mathbf{X}}, \mathbf{y})$ guaranteed to *NOT* be linearly separable either?

- If yes (i.e. $(\tilde{\mathbf{X}}, \mathbf{y})$ is *NOT* linearly separable), prove it.
- If no (i.e. $(\tilde{\mathbf{X}}, \mathbf{y})$ may be linearly separable), provide a counterexample.

original dataset (X, y) is not linearly separable

$(\tilde{X}, y)$ is linearly separable

So exits $w^T \tilde{X} + b \geq 0$.

$\tilde{X} = \Gamma^T X \cdot$

$\cancel{\Gamma^{-1} \tilde{X} = X}$

$w^T X \cdot \Gamma + b \geq 0$.

Problem 7: Dimensionality reduction (Version B) (12 credits)

We would like to perform binary classification on a dataset $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \{0, 1\}^N$.
Assume that we first reduce the dimensionality of $\mathbf{X}$ via PCA to obtain the matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times K}$ (where $K < D$).

a) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is linearly separable. Is the dataset $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ also guaranteed to be linearly separable? If yes, prove it. If not, provide a counterexample.

0 1 2 3 4 5 6

b) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is *NOT* linearly separable. Is the dataset $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ guaranteed to *NOT* be linearly separable either?

0 1 2 3 4 5 6

- If yes (i.e. $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ is *NOT* linearly separable), prove it.
- If no (i.e. $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ may be linearly separable), provide a counterexample.

Problem 7: Dimensionality reduction (Version C) **(12 credits)**

We would like to perform binary classification on a dataset $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X} \in \mathbb{R}^{N\times D}$ and $\mathbf{y} \in \{0, 1\}^N$.
Assume that we first reduce the dimensionality of $\mathbf{X}$ via PCA to obtain the matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{N\times K}$ (where $K < D$).

0 1 2 3 4 5 6

a) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is linearly separable. Is the dataset $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ also guaranteed to be linearly separable? If yes, prove it. If not, provide a counterexample.

0 1 2 3 4 5 6

b) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is *NOT* linearly separable. Is the dataset $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ guaranteed to *NOT* be linearly separable either?

- If yes (i.e. $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ is *NOT* linearly separable), prove it.
- If no (i.e. $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ may be linearly separable), provide a counterexample.

Problem 7: Dimensionality reduction (Version D) (12 credits)

We would like to perform binary classification on a dataset $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \in \{0, 1\}^N$.
Assume that we first reduce the dimensionality of $\mathbf{X}$ via PCA to obtain the matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times K}$ (where $K < D$).

a) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is linearly separable. Is the dataset $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ also guaranteed to be linearly separable? If yes, prove it. If not, provide a counterexample.

0
1
2
3
4
5
6

b) Suppose the original dataset $(\mathbf{X}, \mathbf{y})$ is *NOT* linearly separable. Is the dataset $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ guaranteed to *NOT* be linearly separable either?

- If yes (i.e. $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ is *NOT* linearly separable), prove it.
- If no (i.e. $\left(\tilde{\mathbf{X}}, \mathbf{y}\right)$ may be linearly separable), provide a counterexample.

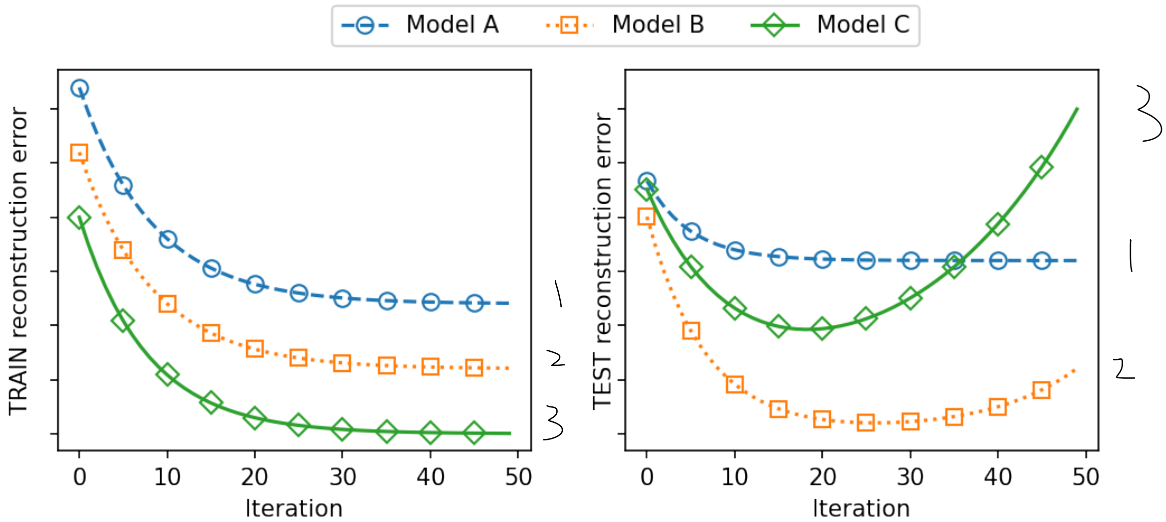
0
1
2
3
4
5
6

Problem 8: Matrix factorization (Version A) (6 credits)

We would like to perform recommendation using matrix factorization. We have trained 3 latent factor models on the same dataset with gradient descent. These models are identical, except using a different value of k (number of latent factors):

- Model 1: $k = 5$
- Model 2: $k = 20$
- Model 3: $k = 50$

The figure below shows the reconstruction error for different models at each optimization step.



0 1 2 3 4 5 6

Your task is to assign the different models (1, 2, 3) to the loss curves in the figure above (A, B, C). Justify your answer.

3 ⇒ too much features, learn fastest but learn early ⇒ overfit

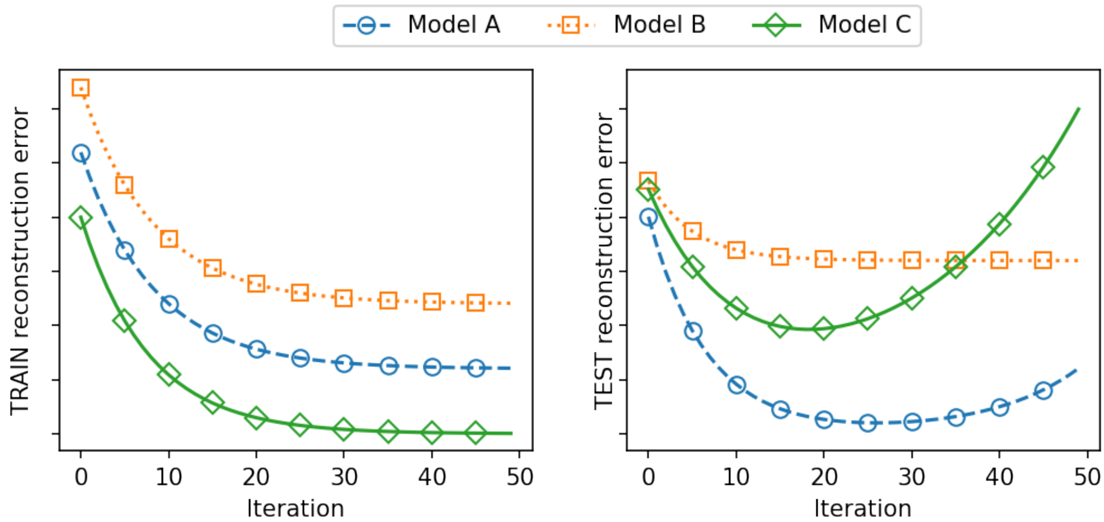
1 ⇒ too little features, learn slowest ~~but~~ can later fitting
underfitting

Problem 8: Matrix factorization (Version B) (6 credits)

We would like to perform recommendation using matrix factorization. We have trained 3 latent factor models on the same dataset with gradient descent. These models are identical, except using a different value of k (number of latent factors):

- Model 1: $k = 5$
- Model 2: $k = 20$
- Model 3: $k = 50$

The figure below shows the reconstruction error for different models at each optimization step.



Your task is to assign the different models (1, 2, 3) to the loss curves in the figure above (A, B, C). Justify your answer.

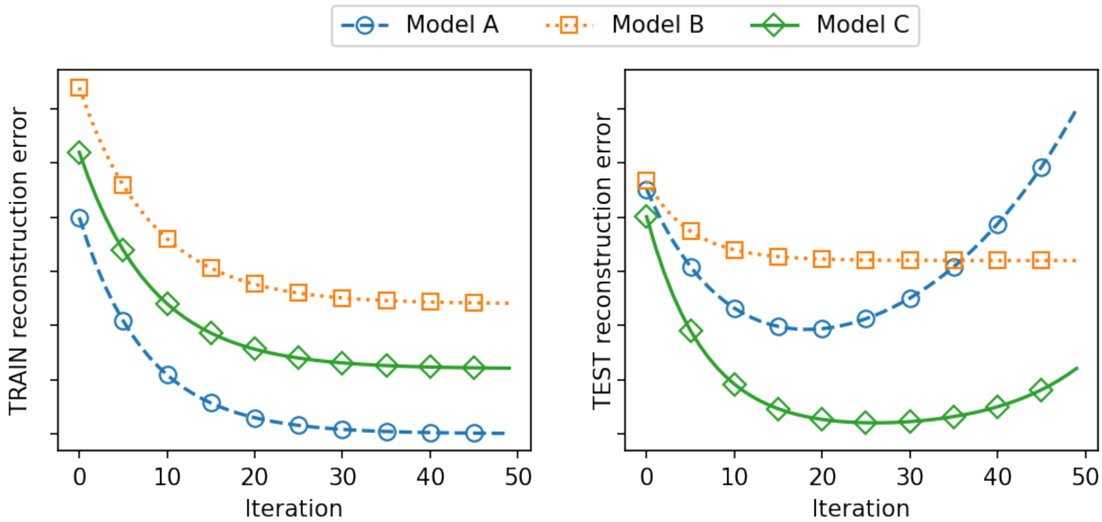
0
1
2
3
4
5
6

Problem 8: Matrix factorization (Version C) (6 credits)

We would like to perform recommendation using matrix factorization. We have trained 3 latent factor models on the same dataset with gradient descent. These models are identical, except using a different value of k (number of latent factors):

- Model 1: $k = 5$
- Model 2: $k = 20$
- Model 3: $k = 50$

The figure below shows the reconstruction error for different models at each optimization step.



0 1 2 3 4 5 6

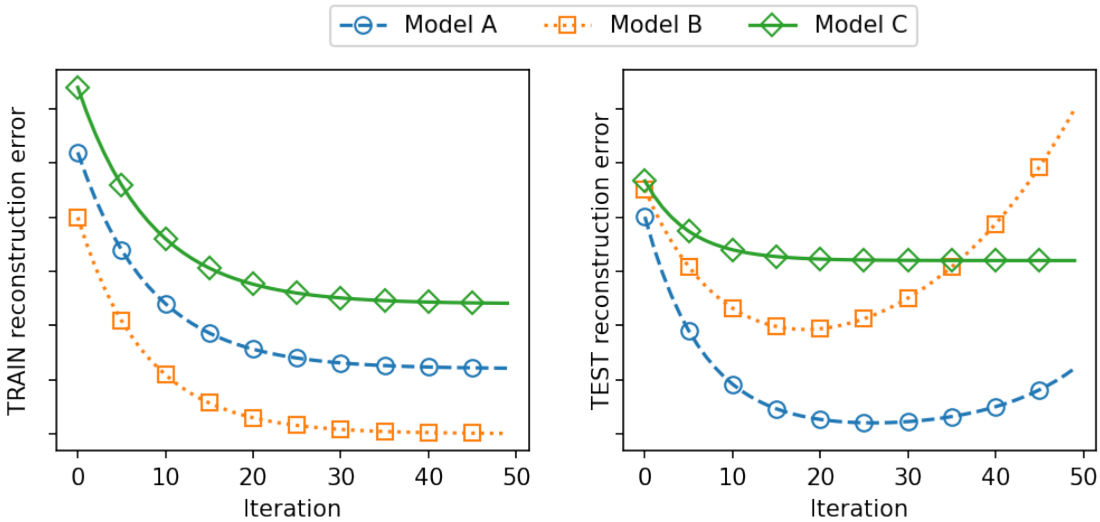
Your task is to assign the different models (1, 2, 3) to the loss curves in the figure above (A, B, C). Justify your answer.

Problem 8: Matrix factorization (Version D) (6 credits)

We would like to perform recommendation using matrix factorization. We have trained 3 latent factor models on the same dataset with gradient descent. These models are identical, except using a different value of k (number of latent factors):

- Model 1: $k = 5$
- Model 2: $k = 20$
- Model 3: $k = 50$

The figure below shows the reconstruction error for different models at each optimization step.



Your task is to assign the different models (1, 2, 3) to the loss curves in the figure above (A, B, C). Justify your answer.

0
1
2
3
4
5
6

Problem 9: Clustering (Version A) (12 credits)

0 1 2 3 4 5 6 7 8 9 10 11 12

Consider the following mixture model with K components and a uniform prior over the components:

$$p(z_i = k) = \frac{1}{K} \qquad\qquad p(\mathbf{x}_i | z_i = k, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K) = \prod_{d=1}^{D} \frac{(\mu_{kd} x_{id})^{(x_{id}-1)} \exp(-\mu_{kd} x_{id})}{x_{id}!},$$

with parameters $\boldsymbol{\mu}_k = (\mu_{k1}, ..., \mu_{kD})^T \in [0,1]^D$ for $k \in \{1, ..., K\}$.

Suppose we are given a dataset consisting of N data points $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, where each data point is represented by a D-dimensional vector of positive natural number, that is $\mathbf{x}_i = (x_{i1}, ..., x_{iD})^T \in \{1, 2, 3, ...\}^D$.

Derive the M-step of the EM algorithm for the above mixture model, assuming that the responsibilites $\gamma_t(z_{ik})$ are given.

$$E \log p(X, Z | \pi, \mu, \Sigma) = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_t(z_i = k) \cdot \ln p(x_i | z_i = k, \mu) \cdot p(z_i = k)$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_t(z_i = k) \left[\left(\sum_{d=1}^{D} \ln \frac{(\mu_{kd} x_{id})^{(x_{id}-1)} \exp(-\mu_{kd} x_{id})}{x_{id}!} \right) + \ln \frac{1}{K} \right]$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_t(z_i = k) \cdot \ln \frac{1}{K} + \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_t(z_i = k) \left[\sum_{d=1}^{D} (x_{id} - 1) \ln(\mu_{kd} x_{id}) - \mu_{kd} \cdot x_{id} - \ln x_{id}! \right]$$

$$\frac{\partial (\cdot)}{\partial \mu_k} = \sum_{i=1}^{N} \gamma_t(z_i = k) \left[(x_{id} - 1) \cdot \frac{1}{\mu_{kd}} - x_{id} \right] \overset{!}{=} 0$$

$$\sum_{i=1}^{N} \gamma_t(z_i = k) \cdot \frac{x_{id} - 1}{\mu_{kd}} = \sum_{i=1}^{N} \gamma_t(z_i = k) x_{id}$$

$$\frac{\sum_{i=1}^{N} \gamma_t(z_i = k)(x_{id} - 1)}{\sum_{i=1}^{N} \gamma_t(z_i = k) x_{id}} = \mu_{kd}.$$

Problem 9: Clustering (Version B) **(12 credits)**

Consider the following mixture model with K components and a uniform prior over the components:

$$p(z_i = k) = \frac{1}{K} \qquad\qquad p(\mathbf{x}_i | z_i = k, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K) = \prod_{d=1}^{D} \frac{(\mu_{kd} x_{id})^{(x_{id}-1)} \exp(-\mu_{kd} x_{id})}{x_{id}!},$$

with parameters $\boldsymbol{\mu}_k = (\mu_{k1}, ..., \mu_{kD})^T \in [0, 1]^D$ for $k \in \{1, ..., K\}$.

Suppose we are given a dataset consisting of N data points $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, where each data point is represented by a D-dimensional vector of positive natural number, that is $\mathbf{x}_i = (x_{i1}, ..., x_{iD})^T \in \{1, 2, 3, ...\}^D$.

Derive the M-step of the EM algorithm for the above mixture model, assuming that the responsibilites $\gamma_t(z_{ik})$ are given.

Problem 9: Clustering (Version C) **(12 credits)**

0 1 2 3 4 5 6 7 8 9 10 11 12

Consider the following mixture model with K components and a uniform prior over the components:

$$p(z_i = k) = \frac{1}{K} \qquad\qquad p(\mathbf{x}_i | z_i = k, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K) = \prod_{d=1}^{D} \frac{(\mu_{kd} x_{id})^{(x_{id}-1)} \exp(-\mu_{kd} x_{id})}{x_{id}!},$$

with parameters $\boldsymbol{\mu}_k = (\mu_{k1}, ..., \mu_{kD})^T \in [0, 1]^D$ for $k \in \{1, ..., K\}$.

Suppose we are given a dataset consisting of N data points $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, where each data point is represented by a D-dimensional vector of positive natural number, that is $\mathbf{x}_i = (x_{i1}, ..., x_{iD})^T \in \{1, 2, 3, ...\}^D$.

Derive the M-step of the EM algorithm for the above mixture model, assuming that the responsibilites $\gamma_t(z_{ik})$ are given.

Problem 9: Clustering (Version D) (12 credits)

Consider the following mixture model with K components and a uniform prior over the components:

0
1
2
3
4
5
6
7
8
9
10
11
12

$$p(z_i = k) = \frac{1}{K} \qquad\qquad p(\mathbf{x}_i | z_i = k, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K) = \prod_{d=1}^{D} \frac{(\mu_{kd} x_{id})^{(x_{id}-1)} \exp(-\mu_{kd} x_{id})}{x_{id}!},$$

with parameters $\boldsymbol{\mu}_k = (\mu_{k1}, ..., \mu_{kD})^T \in [0, 1]^D$ for $k \in \{1, ..., K\}$.

Suppose we are given a dataset consisting of N data points $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, where each data point is represented by a D-dimensional vector of positive natural number, that is $\mathbf{x}_i = (x_{i1}, ..., x_{iD})^T \in \{1, 2, 3, ...\}^D$.

Derive the M-step of the EM algorithm for the above mixture model, assuming that the responsibilites $\gamma_t(z_{ik})$ are given.

Problem 10: Differential privacy (Version A) **(10 credits)**

In the following, we want to ensure that an affine function $f : \mathbb{R}^4 \to \mathbb{R}^4$ with

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$$

and $\mathbf{A} \in \mathbb{R}^{4\times 4}, \mathbf{b} \in \mathbb{R}^4$ does not leak private information about $\mathbf{x} \in \mathbb{R}^4$.

0 1 2 3 4 5 6

a)
Assume that

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 4 & 3 \\ 7 & 2 & 6 & 8 \\ 3 & 8 & 7 & 0 \\ 4 & 1 & 7 & 7 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 2 \\ 7 \\ 1 \\ 8 \end{bmatrix}.$$

Determine the Δ_1 sensitivity of f w.r.t. "$\simeq$", where $\mathbf{x} \simeq \mathbf{x}' \iff \left\{\exists d \ : \ \left(|x_d - x'_d| \leq 1 \land \forall d' \neq d : x_{d'} = x'_{d'}\right)\right\}$.
That is: $\mathbf{x}$ and $\mathbf{x}'$ are considered indistinguishable if they only differ in one component and this component changes by at most 1. Justify your answer.

0 1

b)
To ensure privacy, we construct the Laplace mechanism $\mathcal{M}_{f,\mathrm{Lap}} = f(\mathbf{x}) + \mathbf{z}$ with $\mathbf{z}$ following a 4-dimensional isotropic Laplace distribution, i.e. $\mathbf{z} \sim \mathrm{Lap}\,(0, \sigma)^4$.
Which value must be chosen for σ to ensure $\frac{1}{2}$-differential privacy?

0 1 2 3

c)
Now, we want to ensure differential privacy w.r.t. to the l_∞-norm, i.e. differential privacy w.r.t "$\simeq_\infty$", where $\mathbf{x} \simeq_\infty \mathbf{x}' \iff ||\mathbf{x} - \mathbf{x}'||_\infty \leq 1$.
Prove that the $\frac{1}{2}$-DP mechanism we derived in the previous subproblem is 2-DP w.r.t. "$\simeq_\infty$".
Note: Recall that $||\mathbf{v}||_\infty = \max_d |v_d|$.

Problem 10: Differential privacy (Version B) (10 credits)

In the following, we want to ensure that an affine function $f : \mathbb{R}^4 \to \mathbb{R}^4$ with

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$$

and $\mathbf{A} \in \mathbb{R}^{4\times 4}, \mathbf{b} \in \mathbb{R}^4$ does not leak private information about $\mathbf{x} \in \mathbb{R}^4$.

a)
Assume that

$$\mathbf{A} = \begin{bmatrix} 6 & 6 & 3 & 6 \\ 3 & 5 & 7 & 5 \\ 0 & 2 & 2 & 8 \\ 9 & 6 & 0 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 9 \\ 2 \end{bmatrix}.$$

Determine the Δ_1 sensitivity of f w.r.t. "$\simeq$", where $\mathbf{x} \simeq \mathbf{x}' \iff \left\{\exists d \ : \ \left(|x_d - x'_d| \leq 1 \land \forall d' \neq d : x_{d'} = x'_{d'}\right)\right\}$.
That is: $\mathbf{x}$ and $\mathbf{x}'$ are considered indistinguishable if they only differ in one component and this component changes by at most 1. Justify your answer.

b)
To ensure privacy, we construct the Laplace mechanism $\mathcal{M}_{f,\text{Lap}} = f(\mathbf{x}) + \mathbf{z}$ with $\mathbf{z}$ following a 4-dimensional isotropic Laplace distribution, i.e. $\mathbf{z} \sim \text{Lap}(0, \sigma)^4$.
Which value must be chosen for σ to ensure $\frac{1}{2}$-differential privacy?

c)
Now, we want to ensure differential privacy w.r.t. to the l_∞-norm, i.e. differential privacy w.r.t "$\simeq_\infty$", where $\mathbf{x} \simeq_\infty \mathbf{x}' \iff ||\mathbf{x} - \mathbf{x}'||_\infty \leq 1$.
Prove that the $\frac{1}{2}$-DP mechanism we derived in the previous subproblem is 2-DP w.r.t. "$\simeq_\infty$".
Note: Recall that $||\mathbf{v}||_\infty = \max_d |v_d|$.

Problem 10: Differential privacy (Version C) (10 credits)

In the following, we want to ensure that an affine function $f : \mathbb{R}^4 \to \mathbb{R}^4$ with

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$$

and $\mathbf{A} \in \mathbb{R}^{4\times 4}, \mathbf{b} \in \mathbb{R}^4$ does not leak private information about $\mathbf{x} \in \mathbb{R}^4$.

0 1 2 3 4 5 6

a)
Assume that

$$\mathbf{A} = \begin{bmatrix} 5 & 2 & 1 & 0 \\ 5 & 1 & 4 & 3 \\ 1 & 7 & 2 & 5 \\ 7 & 6 & 0 & 3 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 4 \\ 4 \end{bmatrix}.$$

Determine the Δ_1 sensitivity of f w.r.t. "$\simeq$", where $\mathbf{x} \simeq \mathbf{x}' \iff \left\{\exists d \ : \ \left(|x_d - x'_d| \leq 1 \wedge \forall d' \neq d : x_{d'} = x'_{d'}\right)\right\}$.
That is: $\mathbf{x}$ and $\mathbf{x}'$ are considered indistinguishable if they only differ in one component and this component changes by at most 1. Justify your answer.

0 1

b)
To ensure privacy, we construct the Laplace mechanism $\mathcal{M}_{f,\text{Lap}} = f(\mathbf{x}) + \mathbf{z}$ with $\mathbf{z}$ following a 4-dimensional isotropic Laplace distribution, i.e. $\mathbf{z} \sim \text{Lap}(0, \sigma)^4$.
Which value must be chosen for σ to ensure $\frac{1}{2}$-differential privacy?

0 1 2 3

c)
Now, we want to ensure differential privacy w.r.t. to the l_∞-norm, i.e. differential privacy w.r.t "$\simeq_\infty$", where $\mathbf{x} \simeq_\infty \mathbf{x}' \iff ||\mathbf{x} - \mathbf{x}'||_\infty \leq 1$.
Prove that the $\frac{1}{2}$-DP mechanism we derived in the previous subproblem is 2-DP w.r.t. "$\simeq_\infty$".
Note: Recall that $||\mathbf{v}||_\infty = \max_d |v_d|$.

Problem 10: Differential privacy (Version D) (10 credits)

In the following, we want to ensure that an affine function $f : \mathbb{R}^4 \to \mathbb{R}^4$ with

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$$

and $\mathbf{A} \in \mathbb{R}^{4\times4}, \mathbf{b} \in \mathbb{R}^4$ does not leak private information about $\mathbf{x} \in \mathbb{R}^4$.

a)
Assume that

$$\mathbf{A} = \begin{bmatrix} 8 & 9 & 8 & 0 \\ 7 & 7 & 5 & 8 \\ 6 & 8 & 0 & 5 \\ 9 & 8 & 7 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 3 \\ 3 \\ 2 \\ 9 \end{bmatrix}.$$

Determine the Δ_1 sensitivity of f w.r.t. "$\simeq$", where $\mathbf{x} \simeq \mathbf{x}' \iff \left\{\exists d \ : \ \left(|x_d - x'_d| \leq 1 \wedge \forall d' \neq d : x_{d'} = x'_{d'}\right)\right\}$.
That is: $\mathbf{x}$ and $\mathbf{x}'$ are considered indistinguishable if they only differ in one component and this component changes by at most 1. Justify your answer.

0 1 2 3 4 5 6

b)
To ensure privacy, we construct the Laplace mechanism $\mathcal{M}_{f,\text{Lap}} = f(\mathbf{x}) + \mathbf{z}$ with $\mathbf{z}$ following a 4-dimensional isotropic Laplace distribution, i.e. $\mathbf{z} \sim \text{Lap}(0, \sigma)^4$.
Which value must be chosen for σ to ensure $\frac{1}{2}$-differential privacy?

0 1

c)
Now, we want to ensure differential privacy w.r.t. to the l_∞-norm, i.e. differential privacy w.r.t "$\simeq_\infty$", where $\mathbf{x} \simeq_\infty \mathbf{x}' \iff ||\mathbf{x} - \mathbf{x}'||_\infty \leq 1$.
Prove that the $\frac{1}{2}$-DP mechanism we derived in the previous subproblem is 2-DP w.r.t. "$\simeq_\infty$".
Note: Recall that $||\mathbf{v}||_\infty = \max_d |v_d|$.

0 1 2 3

Problem 11: Fairness (Version A) **(11 credits)**

You are given data as shown in Table 41.1 where $X_1, X_2 \in \mathbb{R}$ denote the non-sensitive features, $A \in \{a, b\}$ denotes the sensitive feature, and $Y \in \{0, 1\}$ denotes the ground-truth label.

Table 41.1: Fairness data (each column is one data point)

ID	1	2	3	4	5	6
X_1	-4	-1	1	2	1	-3
X_2	2	-5	-3	-5	-1	-2
A	a	a	a	b	b	b
Y	0	0	1	0	1	1

You classify the data using the decision tree $r(X_1, X_2)$ shown in Figure 41.1:

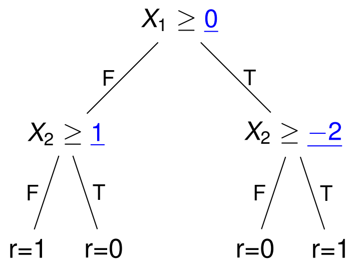


Figure 41.1: Decision tree $r(X_1, X_2)$

a) Compute the prediction of decision tree r for each of the six datapoints.

In the following, we want to modify the decision tree so that different formal fairness criteria are fulfilled. For all fairness criteria, assume that we use the percentages / relative frequences in our dataset in place of probabilities. For instance: $\Pr(Y = 1) \simeq \frac{|\{\text{ID}|Y(\text{ID})=1\}|}{6}$ and $\Pr(Y = 1|A = a) \simeq \frac{|\{\text{ID}|Y(\text{ID})=1 \wedge A(\text{ID})=a\}|}{|\{\text{ID}|A(\text{ID})=a\}|}$.

b) Ensure that the decision tree fulfills the *independence* fairness criterion on the given dataset by modifying *exactly one* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

c) Ensure that the decision tree fulfills the *separation* fairness criterion on the given dataset by modifying *at most two* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

d) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *equality of opportunity* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

e) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *sufficiency* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

Problem 11: Fairness (Version B) **(11 credits)**

You are given data as shown in Table 42.1 where $X_1, X_2 \in \mathbb{R}$ denote the non-sensitive features, $A \in \{a, b\}$ denotes the sensitive feature, and $Y \in \{0, 1\}$ denotes the ground-truth label.

Table 42.1: Fairness data (each column is one data point)

ID	1	2	3	4	5	6
X_1	1	3	-2	-1	1	4
X_2	-3	2	4	1	2	-1
A	a	a	a	b	b	b
Y	0	1	1	0	0	1

You classify the data using the decision tree $r(X_1, X_2)$ shown in Figure 42.1:

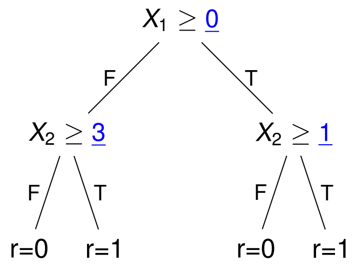


Figure 42.1: Decision tree $r(X_1, X_2)$

a) Compute the prediction of decision tree r for each of the six datapoints.

In the following, we want to modify the decision tree so that different formal fairness criteria are fulfilled. For all fairness criteria, assume that we use the percentages / relative frequences in our dataset in place of probabilities. For instance: $\Pr(Y = 1) \simeq \frac{|\{\mathrm{ID}|Y(\mathrm{ID})=1\}|}{6}$ and $\Pr(Y = 1|A = a) \simeq \frac{|\{\mathrm{ID}|Y(\mathrm{ID})=1 \wedge A(\mathrm{ID})=a\}|}{|\{\mathrm{ID}|A(\mathrm{ID})=a\}|}$.

b) Ensure that the decision tree fulfills the *independence* fairness criterion on the given dataset by modifying *exactly one* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

c) Ensure that the decision tree fulfills the *separation* fairness criterion on the given dataset by modifying *at most two* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

d) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *equality of opportunity* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

e) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *sufficiency* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

Problem 11: Fairness (Version C) **(11 credits)**

You are given data as shown in Table 43.1 where $X_1, X_2 \in \mathbb{R}$ denote the non-sensitive features, $A \in \{a, b\}$ denotes the sensitive feature, and $Y \in \{0, 1\}$ denotes the ground-truth label.

Table 43.1: Fairness data (each column is one data point)

ID	1	2	3	4	5	6
X_1	3	-4	-2	-5	0	0
X_2	5	2	0	-2	0	4
A	a	a	a	b	b	b
Y	0	0	1	0	1	1

You classify the data using the decision tree $r(X_1, X_2)$ shown in Figure 43.1:

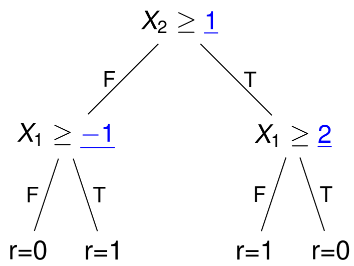


Figure 43.1: Decision tree $r(X_1, X_2)$

a) Compute the prediction of decision tree r for each of the six datapoints.

In the following, we want to modify the decision tree so that different formal fairness criteria are fulfilled. For all fairness criteria, assume that we use the percentages / relative frequences in our dataset in place of probabilities. For instance: $\Pr(Y = 1) \simeq \frac{|\{\text{ID}|Y(\text{ID})=1\}|}{6}$ and $\Pr(Y = 1|A = a) \simeq \frac{|\{\text{ID}|Y(\text{ID})=1 \wedge A(\text{ID})=a\}|}{|\{\text{ID}|A(\text{ID})=a\}|}$.

b) Ensure that the decision tree fulfills the *independence* fairness criterion on the given dataset by modifying *exactly one* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

c) Ensure that the decision tree fulfills the *separation* fairness criterion on the given dataset by modifying *at most two* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

d) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *equality of opportunity* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

e) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *sufficiency* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

Problem 11: Fairness (Version D) (11 credits)

You are given data as shown in Table 44.1 where $X_1, X_2 \in \mathbb{R}$ denote the non-sensitive features, $A \in \{a, b\}$ denotes the sensitive feature, and $Y \in \{0, 1\}$ denotes the ground-truth label.

Table 44.1: Fairness data (each column is one data point)

ID	1	2	3	4	5	6
X_1	4	-1	-3	0	-1	2
X_2	1	3	-1	-3	1	4
A	a	a	a	b	b	b
Y	0	1	1	0	0	1

You classify the data using the decision tree $r(X_1, X_2)$ shown in Figure 44.1:

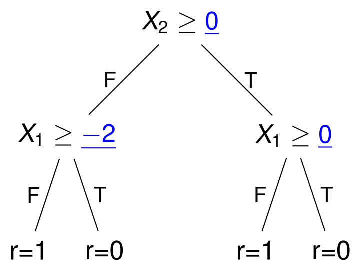


Figure 44.1: Decision tree $r(X_1, X_2)$

a) Compute the prediction of decision tree r for each of the six datapoints.

In the following, we want to modify the decision tree so that different formal fairness criteria are fulfilled. For all fairness criteria, assume that we use the percentages / relative frequences in our dataset in place of probabilities. For instance: $\Pr(Y=1) \simeq \frac{|\{\mathrm{ID}|Y(\mathrm{ID})=1\}|}{6}$ and $\Pr(Y=1|A=a) \simeq \frac{|\{\mathrm{ID}|Y(\mathrm{ID})=1 \wedge A(\mathrm{ID})=a\}|}{|\{\mathrm{ID}|A(\mathrm{ID})=a\}|}$.

b) Ensure that the decision tree fulfills the *independence* fairness criterion on the given dataset by modifying *exactly one* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

c) Ensure that the decision tree fulfills the *separation* fairness criterion on the given dataset by modifying *at most two* of the the three decision thresholds (underlined and highlighted in blue).
Draw the modified decision tree.
Note: You are *not* allowed to change the "$\geq$" or the "$X_{1/2}$" in the decision nodes.

d) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *equality of opportunity* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

e) Is it possible to construct an arbitrary decision tree that simultaneously fulfills *independence* and *sufficiency* on the given dataset? If yes, draw such a decision tree. If no, justify your answer.

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