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

Ecorrection
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.

Learning

Exam: IN2064 / Endterm **Date:** Saturday 11th July, 2020
Examiner: Prof. Dr. Stephan Günnemann **Time:** 10:45 – 12:45

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

Working instructions

- This exam consists of **16 pages** with a total of **12 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 55 credits.
- Allowed resources:
  - all materials that you will use on your own (lecture slides, calculator etc.)
  - **not allowed are any forms of collaboration between examinees and plagiarism**
- Only write on the provided sheets, **submitting your own additional sheets is not possible**.
- Last three pages can be used as scratch paper.
- All sheets (including scratch paper) have to be submitted to the upload queue. Missing pages will be cosidered empty.
- **Only use a black or blue color (no red or green)!**
- Write your answers only in the provided solution boxes or the scratch paper.
- **For problems that say "Justify your answer" you only get points if you provide a valid explanation.**
- **For problems that say "Prove" you only get points if you provide a valid mathematical proof.**
- If a problem does not say "Justify your answer" or "Prove" it's sufficient to only provide the correct answer.
- Exam duration - 120 minutes.

Left room from ____________ to ____________ / Early submission at ____________

Problem 1 KNN-Classification (4 credits)

0 1 2

a) Assume you use a KNN-classifier on the following training data, that contains at least 100 samples of each class.

PS	acceleration	max. velocity [km/h]	cylinder capacity [cm^3]	weight [kg]	class
150	12.5	[illegible]	1968	2001	van
600	3.6	[illegible]	3996	2150	car
113	3.5	[illegible]	937	227	motorcycle
...	...	[illegible]	...	...	...

You observe that the o[illegible]s bad on the test set. What might be the problem? Name at least two possible problems and [illegible]d solve them.

Problem: different range of the features ✓
⇒ features are equally important but have different impact on the model ✓
Standardize the data ✓ : $\boldsymbol{x}_i' = \frac{x_i - \mu_i}{\delta_i}$ ✓
Problem: bad hyperparameter k ✓
⇒ optimize hyperparameter k (gird-search)
Problem: shift between training and test set ✓
⇒ Choose training and test set such that they are from the same distribution

0 1 2

b) Would a decision tree have the same problems? Justify your answer.

Problem: different range of the reatures ⇒ No. ✓
Decision trees can handle features of different scale, because the splits/decision boundaries are computed based misclassification rate, entropy or Gini index. All these measures depend on the labels of the data-point and are computed based on distinguishing if the currently considered feature x is smaller or larger than a threshold. Only feature x influences these measure (for the considered split/test), the other ones don't. Thus, the scale of the features is not important.
Problem: bad hyperparameters ⇒ No, there is no hyperparameter k ✓
Problem: shift between training and test set ⇒ Yes. ✓

Problem 2 Overfitting (3 credits)

Explain overfitting. When does it occur? Why is overfitting unwanted? How can we spot overfitting? How can we avoid it?

☐ 0
☐ 1
☐ 2
☐ 3

Overfitting occurs when we t[illegible] data perfectly. ✓

Overfitting results in a poor g[illegible]del. ✓

Spot: Split the data-set into a [illegible]set and test-set an compare the error on the training- and validation-set (during training[illegible]ogether with a high validation error indicates over-fitting. ✓

Avoid: Track error on validati[illegible] stop training when the validation error stops decreasing. ✓ (possible alternatives: we[illegible]rs and full posterior inference)

Problem 3 Probabilistic inference (7 credits)

0 1 2 3 4 5 6 7

Consider the following probabilistic model

$$p(\lambda \mid a, b) = \mathrm{Gamma}(\lambda \mid a, b) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} \exp(-b\lambda)$$

$$\ldots \mathrm{Poisson}(x \mid \lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}$$

where $a \in (1, \infty)$ and ... observed a single data point $x \in \mathbb{N}$. Derive the maximum a posteriori (MAP) estimate of the ... ove probabilistic model. Show your work.

✓✓✓✓ The M... fined as

$$\begin{aligned}
\lambda_{MAP} &= \arg\max_\lambda p(\lambda \mid x, a, b) \\
&= \arg\max_\lambda \log p(\lambda \mid x, a, b) \\
&= \arg\max_\lambda \left(\log p(x \mid \lambda) + \log p(\lambda \mid a, b)\right) \\
&= \arg\min_\lambda -\left(\log p(x \mid \lambda) + \log p(\lambda \mid a, b)\right) \\
&= \arg\min_\lambda \left(-x \log(\lambda) + \lambda + \log x! - a \log b + \log \Gamma(a) - (a-1) \log \lambda + b\lambda\right) \\
&= \arg\min_\lambda \left((1 - x - a) \log(\lambda) + (b+1)\lambda\right)
\end{aligned}$$

✓✓✓ This a convex function of λ. To minimize, compute the derivative, set it to zero and solve for λ.

$$\begin{aligned}
\frac{\partial}{\partial \lambda}\left((1 - x - a)\log(\lambda) + (b+1)\lambda\right) &= \frac{1 - x - a}{\lambda} + b + 1 \overset{!}{=} 0 \\
\iff \frac{x + a - 1}{\lambda} &= b + 1 \\
\iff \lambda &= \frac{x + a - 1}{b + 1} \checkmark
\end{aligned}$$

Therefore, $\lambda_{MAP} = \frac{x+a-1}{b+1}$.

Lecture 3, slide 26.

Problem 4 Regression (5 credits)

a) Assume you train a linear regression model on dataset $D = \{x_i, y_i\}_i$, $x_i \in \mathbb{R}^D$, $y_i \in \mathbb{R}$ with the mean-square-error as loss function. After training is finished, you compute the MSE on individual data-points of the training-set. You notice that for three points you obtain a high MSE (1000 times higher than for the other points).
Evaluation on the test-set shows that your regression model does not perform that well. What might be the reason for that? How would you improve [illegible] model? Justify your answer.

0 1 2 3

Reason: outliers in the data-[illegible] high MSE)
Due to training with MSE out[illegible]ence on the model
$\Rightarrow$ bad performance on the te[illegible]
Improve performance: Use a[illegible](L1)/ remove outliers and train again

b) You want to train another linear regression model and decide to use the log-cosh-loss:

0 1 2

$$E_{lc} = \sum_i \log\cosh(w^T x_i - y_i)$$

How do you learn the parameter w of your model? Describe in one or two sentences.
Hint: $\cosh(z) = 0.5(e^z + e^{-z})$

Log-cosh-loss is twice differentiable at each point. $\Rightarrow$ do gradient descent to minimize loss function f
Gradient points into steepest ascent direction and is a good local approximation of the objective function f
Gradient descent:
Initialize parameter w randomly
Update until stopping criterion is satisfied: $w = w - t\nabla f(w)$, where t is the learning rate

Problem 5 Classification (4 credits)

We would like to design a generative classification model for the following data.
Each data point is represented by a D-dimensional feature vector $\mathbf{x} = (x_1, ..., x_D)$ where each entry x_j is a real number between 0 and 1, that is $x_j \in [0, 1]$ for $j = 1, ..., D$. Each data point belongs to one of $K > 2$ possible classes, that is $y \in \{1, ..., K\}$.

Lecture 3, slide 23: Beta distribution. Lecture 5, slides 21-22: generative models for clasification. Exercise 5, problem 1.

a) Which of the following distributions is the most reasonable choice for the class prior $p(y)$?

- ☐ Bernoulli
- ☐ Normal
- ☐ Beta
- ☐ Exponential
- ☒ Categorical

b) We decide to model the class conditional distribution $p(\mathbf{x}|y)$ as $p(\mathbf{x}|y) = \prod_{j=1}^{D} p(x_j|y)$, that is, the features x_j are conditionally independent give the class label y.

Which of the following distributions is the most reasonable choice for $p(x_j|y)$?

- ☐ Categorical
- ☒ Beta
- ☐ Normal
- ☐ Exponential
- ☐ Bernoulli

0 ☐
1 ☐
2 ☐

c) What is the name of the posterior distribution $p(y|\mathbf{x})$ for the model that you specified in subtasks (a) and (b)? Justify your answer.

Note that you need to provide the name of the distribution, not its probability density / mass functions. If the posterior distribution doesn't have a name, you should write "unknown distribution".

Since y can take K distinct values ✓, the only possible choice is **Categorical** ✓ distribution.

Problem 6 Alternative characterization of vertices (4 credits)

Consider a non-empty convex set $\mathcal{X} \subset \mathbb{R}^n$ and $\mathbf{x} \in \mathcal{X}$. Prove that if $\mathbf{x}$ is a vertex of $\mathcal{X}$ then $\mathcal{X} \setminus \{\mathbf{x}\}$ is convex.

0 1 2 3 4

Hint: additionally to the definition from the lecture you can use that $\mathbf{x} \in \mathcal{X}$ is a vertex of $\mathcal{X}$ if and only if for all $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}$ with $\mathbf{x}_0 \neq \mathbf{x}_1$ and all $\lambda \in (0,1)$ there holds that $\mathbf{x} \neq \mathbf{x}_\lambda$, where $\mathbf{x}_\lambda = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_0$ (i.e. $\mathbf{x}$ does not lie between two different points fro

We prove it by using the stand ity for sets. Take arbitrary $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X} \setminus \{\mathbf{x}\}$ and $\lambda \in [0,1]$, we show that $\mathbf{x}_\lambda \in \mathcal{X} \setminus \{\mathbf{x}\}$. W two cases ✓ .
Case 1: $\mathbf{x}_0 = \mathbf{x}_1$ or $\lambda \in \{0,1\}$, $\mathcal{X} \setminus \{\mathbf{x}\}$ or $\mathbf{x}_\lambda = \mathbf{x}_1 \in \mathcal{X} \setminus \{\mathbf{x}\}$ (or even both).✓
Case 2: $\mathbf{x}_0 \neq \mathbf{x}_1$ and $\lambda \in (0,1$ holds that $\mathbf{x}_\lambda \in \mathcal{X}$.✓ Additionally, all conditions from the definition of a vertex (from the therefore $\mathbf{x} \neq \mathbf{x}_\lambda$.✓ Therefore $\mathbf{x}_\lambda \in \mathcal{X} \setminus \{\mathbf{x}\}$ proving that $\mathcal{X} \setminus \{\mathbf{x}\}$ is convex.

Problem 7 Classification with Hinge loss and L_∞ penalty (7 credits)

For $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^D$ and $y_1, \dots, y_N \in \{-1, 1\}$ consider the following optimization problem with a fixed parameter $\lambda > 0$ and $\|\boldsymbol{w}\|_\infty = \max(|w_1|, \dots, |w_D|)$.

$$\text{minimize}_{\boldsymbol{w},b} \quad \sum_{i=1}^{N} \max\left(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\right) + \lambda\|\boldsymbol{w}\|_\infty \,. \tag{1}$$

a) In this task you have to choose the correct options. Problem (1) as formulated above is

- ☐ concave.
- ☒ not a quadratic problem.
- ☐ a linear problem.
- a quadratic problem.
- convex.
- a minimization problem.
- ☒ unconstrained.
- ☐ constrained.
- ☐ non-convex.

0 1 2 3 4 5

b) Reformulate problem (1) as an optimization problem with a linear objective and linear constraints. Justify your answer.

Hint: you can introduce new variables to the problem.

We know from the lecture that the non-linear Hinge loss terms can be removed by introducing new variables $\boldsymbol{\xi} \in \mathbb{R}^N$ and corresponding linear constraints. This way we can reformulate (1) as

✓

$$\text{minimize}_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \sum_{i=1}^{N} \xi_i + \lambda\|\boldsymbol{w}\|_\infty$$

✓ ✓

$$\text{subject to} \quad 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \le \xi_i \quad \text{and} \quad \xi_i \ge 0 \qquad \text{for all } i = 1, \dots, N\,.$$

Analogously to each ξ_i we introduce a new scalar variable $\zeta \in \mathbb{R}$ that should represent $\|\boldsymbol{w}\|_\infty$ (which is also the maximum of a finite number of values) and get the following problem.

✓

$$\text{minimize}_{\boldsymbol{w},b,\boldsymbol{\xi},\zeta} \quad \sum_{i=1}^{N} \xi_i + \lambda\zeta$$

$$\text{subject to} \quad 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \le \xi_i \quad \text{and} \quad \xi_i \ge 0 \qquad \text{for all } i = 1, \dots, N$$

$$\zeta \ge |w_j| \qquad \text{for all } j = 1, \dots, d\,.$$

The only non-linear constraints are $\zeta \ge |w_j|$. Each of these can be equivalently replaced by two linear constraints $\zeta \ge w_j$ and $\zeta \ge -w_j$. ✓

This way we arrive at a reformulation of (1) which is a LP task.

✓ can be subtracted if
- explanation / reference to the lecture is missing,
- new variables are missing in minimize$_{\boldsymbol{w},b,\dots}$,
- objective function / constraints do not represent a reformulation of (1),
- final objective function / constraints are not linear.

Problem 8 Deep learning (4 credits)

0
1
2
3
4

We are using a fully-connected neural network with 2 hidden layers for binary classification of points in $\mathbb{R}^D$

$$f(\mathbf{x}, \mathbf{W}) = \sigma_2(\mathbf{W}_2\sigma_1(\mathbf{W}_1\sigma_0(\mathbf{W}_0\mathbf{x}))).$$

where $\mathbf{W} = \{\mathbf{W}_0, \mathbf{W}_1, \mathbf{W}_2\}$ with $\mathbf{W}_0 \in \mathbb{R}^{D_1\times D}$, $\mathbf{W}_1 \in \mathbb{R}^{D_2\times D_1}$ and $\mathbf{W}_2 \in \mathbb{R}^{1\times D_2}$ are the weights of the neural network. The neural network outputs pr[illegible]ive class, i.e. $p(y = 1|\mathbf{x}, \mathbf{W}) = f(\mathbf{x}, \mathbf{W})$, and is trained by minimizing the binary cross-entr[illegible]ollowing activation functions:

$$\sigma_0(t) = t\sqrt{69} \qquad \sigma_1(t) = [illegible]\frac{t}{54\pi} \qquad \sigma_2(t) = \frac{1}{1 + \exp(-67t)}$$

The neural network achieves 1[illegible]uracy on a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Which of the following statements is true? Justify your [illegible]

1. $\mathcal{D}$ is linearly separable.
2. $\mathcal{D}$ is NOT linearly separable.
3. There is not enough information to determine if $\mathcal{D}$ is linearly separable.

Lecture 5, slide 8: definition of a linearly separable datasets. Lecture 10, slide 10: NN with linear activations learns a linear decision boundary.

Activation functions σ_0 and σ_1 are linear. The prediction function can be rewritten as

$$f(\mathbf{x}, \mathbf{W}) = \sigma_2(\mathbf{W}_2(-\frac{1}{3\pi}\mathbf{I})\mathbf{W}_1(\sqrt{2}\mathbf{I})\mathbf{W}_0\mathbf{x})$$

This architecture is equivalent to a binary logistic regression model

$$g(\mathbf{x}, \mathbf{W}) = \sigma_2(\mathbf{V}\mathbf{x})$$

where $\mathbf{V} = \mathbf{W}_2(-\frac{1}{3\pi}\mathbf{I})\mathbf{W}_1(\sqrt{2}\mathbf{I})\mathbf{W}_0 = -\frac{\sqrt{2}}{3\pi}\mathbf{W}_2\mathbf{W}_1\mathbf{W}_0$.

This means that the decision boundary learned by the neural network above is linear. So, if the NN achieves 100% accuracy on the training set, the training set has to be linearly separable. That is, option (1) is correct.

1 point for stating that σ_0 and σ_1 are linear.
1 point for stating that σ_2 is a sigmoid or that we can combine the linear part into a single matrix.
1 point for using 100% accuracy (or something equivalent) as part of the argument.
1 point for the result: $\mathcal{D}$ is linearly separable.

Problem 9 Principal Component Analysis (4 credits)

Consider the data

$$\boldsymbol{X} = \begin{pmatrix} 0.37 & 0.05 & 0.73 & 0.60 \\ 0.16 & 0.16 & 0.06 & 0.87 \\ 0.60 & 0.71 & 0.02 & 0.97 \\ 0.83 & 0.21 & 0.18 & 0.18 \\ 0.30 & 0.52 & 0.43 & 0.29 \\ 0.61 & 0.14 & 0.29 & 0.37 \\ 0.46 & 0.79 & 0.20 & 0.51 \\ 0.59 & 0.05 & 0.61 & 0.17 \end{pmatrix},$$

where each row of $\boldsymbol{X}$ r[illegible]

0 1 2 3 4

In each of the followin[illegible]st row of $\boldsymbol{\Gamma}$ corresponds to the first principal component (associated with the first variance), the second row to the second, etc. Only one of these solutions is correct. Which one is it? For each wrong solution give a reason for why it is wrong!

Variances	Principal component matrix $\boldsymbol{\Gamma}$	Answer
$\begin{pmatrix} 0.16 \\ 0.10 \\ 0.05 \end{pmatrix}$	$\begin{pmatrix} 0.25 & -0.72 & 0.14 \\ 0.01 & 0.54 & 0.71 \\ -0.81 & -0.37 & 0.41 \\ -0.52 & 0.23 & -0.56 \end{pmatrix}$	Wrong. Principal components should be from $\mathbb{R}^4$. ✓
$\begin{pmatrix} 0.16 \\ 0.10 \\ 0.05 \end{pmatrix}$	$\begin{pmatrix} 0.25 & -0.72 & 0.14 & -0.63 \\ 0.01 & 0.54 & 0.71 & -0.46 \\ -0.81 & -0.37 & 0.41 & 0.18 \end{pmatrix}$	Correct.
$\begin{pmatrix} 0.16 \\ -0.10 \\ 0.05 \\ 0.01 \end{pmatrix}$	$\begin{pmatrix} 0.25 & -0.72 & 0.14 & -0.63 \\ 0.01 & 0.54 & 0.71 & -0.46 \\ -0.81 & -0.37 & 0.41 & 0.18 \\ -0.52 & 0.23 & -0.56 & -0.60 \end{pmatrix}$	Wrong. Variances must be positive. ✓
$\begin{pmatrix} 0.16 \\ 0.05 \\ 0.10 \\ 0.01 \end{pmatrix}$	$\begin{pmatrix} 0.25 & -0.72 & 0.14 & -0.63 \\ -0.81 & -0.37 & 0.41 & 0.18 \\ 0.01 & 0.54 & 0.71 & -0.46 \\ -0.52 & 0.23 & -0.56 & -0.60 \end{pmatrix}$	Wrong. Variances are not monotonically decreasing. ✓
$\begin{pmatrix} 0.16 \\ 0.10 \\ 0.05 \end{pmatrix}$	$\begin{pmatrix} 0.25 & -0.72 & 0.14 & -0.63 \\ 0.01 & 0.54 & 0.71 & -0.46 \\ 0.50 & -0.72 & 0.14 & -0.63 \end{pmatrix}$	Wrong. Principal components are not orthogonal (1st and 3rd vectors). ✓

Problem 10 Mixture Models (1 credit)

Let $z \sim \mathrm{Cat}(\pi)$ be a random variable with categorical distribution on $\{1, \dots, K\}$ with probabilities $p(z = k) = \pi_k$ for $k \in \{1, \dots, K\}$. Furthermore, let x be a random variable dependent on z with an arbitrary likelihood, i.e. $p(x \mid z)$ can be any probability distribution. Which of the following is the general form of $p(z = k \mid x)$?

- [ ] $\pi_k \prod_{i=1}^{K} \log \left(p(x \mid z = i\right.$ … g $p(x)$
- [ ] $p(x \mid z = k) \cdot \frac{\pi_k}{\sum_{i=1}^{K} \pi_i}$ Th…
- [ ] $p(z = k) \cdot \mathbb{E}_z \left[p\left(x \mid z\right)\right.$ …
- [x] $p(x \mid z = k)\, \pi_k \left(\sum_{i=1}^{K}\right.$ …

Problem 11 EM Algorithm (10 credits)

Consider a one-dimensional mixture of exponential distributions with K components and a uniform prior over components, i.e.

$$p(z_i = k) = \frac{1}{K} \qquad p(x \mid \lambda_k, z_i = k) = \lambda_k \exp\left(-\lambda_k x\right) \qquad \text{where } \lambda_k > 0.$$

We have observed N values $x_i \in \mathbb{R}_{\geq 0}$ ($i = 1, \dots, N$) and want to fit this mixture model with the EM algorithm.

a) Derive the M-step, i.e. the responsibilites respectively the posterior $\gamma(z_i = k) = p(z_i = k \mid x_i)$.

0 1 2 3

The application of Bayes' theorem gets us

$$p(z_i = k \mid x_i, \boldsymbol{\lambda}) \propto p(z_i = k) \cdot p(x_i \mid \lambda_k, z_i = k) = \frac{1}{K} \cdot \lambda_k \exp\left(-\lambda_k x_i\right)$$

and with the sum-to-1 constraint on discrete distributions we conclude

$$\gamma(z_i = k) = p(z_i = k \mid x_i, \boldsymbol{\lambda}) = \frac{\frac{1}{K} \cdot \lambda_k \exp\left(-\lambda_k x_i\right)}{\sum_{k'=1}^{K} \frac{1}{K} \lambda_{k'} \exp\left(-\lambda_{k'} x_i\right)} = \frac{\lambda_k \exp\left(-\lambda_k x_i\right)}{\sum_{k'=1}^{K} \lambda_{k'} \exp\left(-\lambda_{k'} x_i\right)}.$$

✓ ✓ ✓ for deriving the E-step correctly

0 1 2 3 4 5

b) Derive the E-step, i.e. find $\arg\max_{\lambda} \mathbb{E}_{\mathbf{Z}\sim\gamma}\left[\log p\left(\mathbf{Z}, \mathbf{X} \mid \boldsymbol{\lambda}\right)\right]$. Here $\mathbf{Z}$ represents all z_i and $\mathbf{X}$ all x_i ($i = 1 \dots N$).

First we take a closer look at the expected data log-likelihood term

$$
\begin{aligned}
\underset{\mathbf{Z}\sim\gamma}{\mathbb{E}}\left[\log p\left(\mathbf{Z}, \mathbf{X} \mid \boldsymbol{\lambda}\right)\right] &= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma(z_i = k) \log p\left(\mathbf{Z}, \mathbf{X} \mid \boldsymbol{\lambda}\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma(z_i = k) \log \frac{1}{K} \lambda_k \exp\left(-\lambda_k x_i\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma(z_i = k) \left(\log\left(\lambda_k\right) - \lambda_k x_i\right) + C
\end{aligned}
$$

where all terms cor are collected into C. Now compute the derivative with respect to λ_k

$$
\frac{\partial}{\partial \lambda_k} \underset{\mathbf{Z}\sim\gamma}{\mathbb{E}}\left[\log p\left(\mathbf{Z}, \mathbf{X} \mid \boldsymbol{\lambda}\right)\right] = \sum_{i=1}^{N} \gamma(z_i = k)\left(\frac{1}{\lambda_k} - x_i\right)
$$

and find the root

$$
\sum_{i=1}^{N} \gamma(z_i = k)\left(\frac{1}{\lambda_k} - x_i\right) = 0 \Leftrightarrow \lambda_k = \frac{\sum_{i=1}^{N} \gamma(z_i = k)}{\sum_{i=1}^{N} \gamma(z_i = k) x_i}
$$

which constitutes the update step.

✓ ✓ for expanding the expected data log-likelihood

✓ ✓ ✓ for solving for the optimal λ

c) Is the EM algorithm guranteed to converge to a global optimum in general? If yes, justify why. If no, how to avoid getting stuck in local optima or saddle points?

0
1
2

Local optima can only be reached from very specific initial values. So this can for all intents and purposes be avoided with random initialization of the model parameters. In general, one can repeat the model fitting multiple times with different parameters and keep the configuration that achieved the highest likelihood.

✓ for saying no
✓ for suggesting multiple ra

Problem 12 Differential Privacy (2 credits)

0
1
2

Let $\mathcal{M}_f : \mathbb{R}^D \to \mathbb{R}^D$ be an $\epsilon-$DP mechanism with a privacy parameter ϵ applied to the function $f : \mathbb{R}^D \to \mathbb{R}^D$. Similarly, let $\mathcal{N}_g : \mathbb{R}^D \to \mathbb{R}^D$ be a $\sigma-$DP mechanism with a privacy parameter σ applied to the function g.

Let $h_1 : \mathbb{R}^D \to \mathbb{R}^D$ and $h_2 : \mathbb{R}^D \to \mathbb{R}^D$ be arbitrary functions and $\boldsymbol{X} \in \mathbb{R}^D$. Can we provide differential privacy guarantees for the following mappings? If yes, what is their respective privacy parameter? If no, why not?

a) $\boldsymbol{X} \mapsto (\mathcal{M}_f(\boldsymbol{X}), \mathcal{N}_g(\boldsymbol{X}))$

b) $\boldsymbol{X} \mapsto h_1(\mathcal{N}_g(h_2(\boldsymbol{X})))$

c) $\boldsymbol{X} \mapsto h_2(\mathcal{M}_f(\boldsymbol{X}))$

d) $\boldsymbol{X} \mapsto (\mathcal{M}_f(h_1(\boldsymbol{X})), \mathcal{N}_g(\boldsymbol{X}))$

You randomly received four of the following mappings. Here is the correct solution for all of them.

- $\boldsymbol{X} \to (\mathcal{M}_f(\boldsymbol{X}), \mathcal{N}_g(\boldsymbol{X}))$: Yes, from the composition property the privacy parameter is $\epsilon + \sigma$
- $\boldsymbol{X} \to h_1(\mathcal{N}_g(h_2(\boldsymbol{X})))$: No, h_2 is arbitrary and is applied before $\mathcal{N}_g$
- $\boldsymbol{X} \to h_2(\mathcal{M}_f(\boldsymbol{X}))$: Yes, from the robustness to postprocessing property the privacy parameter is ϵ
- $\boldsymbol{X} \to (\mathcal{M}_f(h_1(\boldsymbol{X})), \mathcal{N}_g(\boldsymbol{X}))$: No, h_1 is arbitrary and is applied before $\mathcal{M}_f$
- $\boldsymbol{X} \to h_2(h_1(\boldsymbol{X}))$: No, h_1 and h_2 are arbitrary
- $\boldsymbol{X} \to \mathcal{M}_f(h_1(\boldsymbol{X}))$: No, h_1 is arbitrary and is applied before $\mathcal{M}_f$
- $\boldsymbol{X} \to h_2(h_1(\mathcal{N}_g\boldsymbol{X})))$: Yes, from the robustness to postprocessing property the privacy parameter is σ
- $\boldsymbol{X} \to \mathcal{M}_f(h_2(h_1(\boldsymbol{X})))$: No, h_1 and h_2 are arbitrary and are applied before $\mathcal{M}_f$
- $\boldsymbol{X} \to (h_2(\mathcal{M}_f(\boldsymbol{X})), h_1(\mathcal{N}_g(\boldsymbol{X})))$: Yes, from the robustness to postprocessing property and the composition property the privacy parameter is $\epsilon + \sigma$

Answers that only say "no" or "yes" without explaining why give 0 points.
If 1/4 is answered correctly you get 1 point.
If **all** 4/4 are answered correctly you get 2 points.
Note that if only 2/4 or 3/4 are correct you still get only 1 point.

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