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

Machine Learning for Graphs and Sequential Data (Problem sheet)

Graded Exercise:	IN2323 / Retake	**Date:**	Thursday 14th October, 2021
Examiner:	Prof. Dr. Stephan Günnemann	**Time:**	14:15 – 15:30

Working instructions

- **DO NOT SUBMIT THIS SHEET! ONLY SUBMIT YOUR PERSONALIZED ANSWER SHEET THAT IS DISTRIBUTED THROUGH TUMEXAM!**
- Make sure that you solve the version of the problem stated on your personalized answer sheet (e.g., Problem 1 (Version B), Problem 2 (Version A), etc.)

Problem 1: Normalizing Flows (Version A) (6 credits)

0 1 2 3 4 5 6

We consider two transformations $f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}^3$ for use in normalizing flows.
Let

$$f_1(\mathbf{z}) = \begin{bmatrix} (1 + \max(0, z_2)) \cdot z_1 - \min(0, z_2) \cdot z_3 \\ (z_2)^3 \\ z_1 - z_3 \end{bmatrix} \qquad \text{and} \qquad f_2(\mathbf{z}) = \begin{bmatrix} (z_1)^3 \\ (z_3)^3 \cdot \exp(z_2) \\ z_1 \cdot |z_3| \end{bmatrix}.$$

Prove or disprove whether f_1 and/or f_2 are invertible.

Problem 1: Normalizing Flows (Version B) (6 credits)

We consider two transformations $f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}^3$ for use in normalizing flows.
Let

$$f_1(\mathbf{z}) = \begin{bmatrix} (z_2)^3 \\ (1 + \max(0, z_2)) \cdot z_1 - \min(0, z_2) \cdot z_3 \\ z_1 - z_3 \end{bmatrix} \qquad \text{and} \qquad f_2(\mathbf{z}) = \begin{bmatrix} (z_1)^3 \\ \ln(1 + |z_3|) \cdot \exp(z_2) \\ z_1 \cdot z_3 \end{bmatrix}.$$

Prove or disprove whether f_1 and/or f_2 are invertible.

0
1
2
3
4
5
6

Problem 1: Normalizing Flows (Version C) (6 credits)

0
1
2
3
4
5
6

We consider two transformations $f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}^3$ for use in normalizing flows.
Let

$$f_1(\mathbf{z}) = \begin{bmatrix} \min(0, z_2) \cdot z_3 + (1 + \max(0, z_2)) \cdot z_1 \\ (z_2)^3 \\ z_1 + 2 \cdot z_3 \end{bmatrix} \quad \text{and} \quad f_2(\mathbf{z}) = \begin{bmatrix} \ln(1 + |z_1|) \\ (z_3)^3 \cdot \exp(z_2) \\ z_1 \cdot z_3 \end{bmatrix}.$$

Prove or disprove whether f_1 and/or f_2 are invertible.

Problem 1: Normalizing Flows (Version D) (6 credits)

0
1
2
3
4
5
6

We consider two transformations $f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}^3$ for use in normalizing flows.
Let

$$f_1(\mathbf{z}) = \begin{bmatrix} (z_2)^3 \\ \min(0, z_2) \cdot z_3 + (1 + \max(0, z_2)) \cdot z_1 \\ z_1 + 2 \cdot z_3 \end{bmatrix} \quad \text{and} \quad f_2(\mathbf{z}) = \begin{bmatrix} (z_2)^3 \\ z_2 \cdot |z_1| \\ (z_1)^3 \cdot \exp(z_3) \end{bmatrix}.$$

Prove or disprove whether f_1 and/or f_2 are invertible.

Problem 2: Variational Inference (Version A) (7 credits)

Suppose we are given a latent variable model

$$p(z) = \mathcal{N}(z; 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$$
$$p_\theta(x|z) = \mathcal{N}(x; z + 5, \theta^2) = \frac{1}{\theta\sqrt{2\pi}} \exp\left(-\frac{(x - z - 5)^2}{2\theta^2}\right)$$

where $x, z \in \mathbb{R}$. We parametrize the variational distribution $q_\phi(z)$ as:

$$q_\phi(z) = \mathcal{N}(z; \phi, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(z - \phi)^2}{2}\right)$$

0 1 2 3 4

a) Derive the evidence lower bound (ELBO) for this particular parametrization. Simplify the parts depending on ϕ as far as possible.

Reminder: The ELBO for parameters θ and variational distribution q_ϕ is defined as

$$\mathcal{L}(\theta, q_\phi) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q_\phi}\left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\phi(\boldsymbol{z})\right].$$

Hint: Given a random variable X, the variance decomposition $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ can be rewritten as

$$\mathbb{E}[X^2] \;=\; \mathrm{Var}(X) \;+\; \mathbb{E}[X]^2.$$

0 1 2 3

b) Suppose θ is fixed. Derive the value of ϕ that maximizes the ELBO.

Problem 2: Variational Inference (Version B) (7 credits)

Suppose we are given a latent variable model

$$p(z) = \mathcal{N}(z; 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$$

$$p_\theta(x|z) = \mathcal{N}(x; 2z + 4, \theta^2) = \frac{1}{\theta\sqrt{2\pi}} \exp\left(-\frac{(x - 2z - 4)^2}{2\theta^2}\right)$$

where $x, z \in \mathbb{R}$. We parametrize the variational distribution $q_\mu(z)$ as:

$$q_\mu(z) = \mathcal{N}(z; \mu, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(z - \mu)^2}{2}\right)$$

0 1 2 3 4

a) Derive the evidence lower bound (ELBO) for this particular parametrization. Simplify the parts depending on μ as far as possible.

Reminder: The ELBO for parameters θ and variational distribution q_μ is defined as

$$\mathcal{L}(\theta, q_\mu) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q_\mu}\left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\mu(\boldsymbol{z})\right].$$

Hint: Given a random variable X, the variance decomposition $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ can be rewritten as

$$\mathbb{E}[X^2] \;=\; \mathrm{Var}(X) \;+\; \mathbb{E}[X]^2.$$

0 1 2 3

b) Suppose θ is fixed. Derive the value of μ that maximizes the ELBO.

Problem 2: Variational Inference (Version C) (7 credits)

Suppose we are given a latent variable model

$$p(z) = \mathcal{N}(z; 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$$
$$p_\theta(x|z) = \mathcal{N}(x; z + 3, \theta^2) = \frac{1}{\theta\sqrt{2\pi}} \exp\left(-\frac{(x - z - 3)^2}{2\theta^2}\right)$$

where $x, z \in \mathbb{R}$. We parametrize the variational distribution $q_\phi(z)$ as:

$$q_\phi(z) = \mathcal{N}(z; \phi, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(z - \phi)^2}{2}\right)$$

0 1 2 3 4

a) Derive the evidence lower bound (ELBO) for this particular parametrization. Simplify the parts depending on ϕ as far as possible.

Reminder: The ELBO for parameters θ and variational distribution q_ϕ is defined as

$$\mathcal{L}(\theta, q_\phi) = \mathop{\mathbb{E}}_{\boldsymbol{z} \sim q_\phi} \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\phi(\boldsymbol{z})\right].$$

Hint: Given a random variable X, the variance decomposition $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ can be rewritten as

$$\mathbb{E}[X^2] = \mathrm{Var}(X) + \mathbb{E}[X]^2.$$

0 1 2 3

b) Suppose θ is fixed. Derive the value of ϕ that maximizes the ELBO.

Problem 2: Variational Inference (Version D) (7 credits)

Suppose we are given a latent variable model

$$p(z) = \mathcal{N}(z; 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$$
$$p_\theta(x|z) = \mathcal{N}(x; z + 7, \theta^2) = \frac{1}{\theta\sqrt{2\pi}} \exp\left(-\frac{(x - z - 7)^2}{2\theta^2}\right)$$

where $x, z \in \mathbb{R}$. We parametrize the variational distribution $q_\mu(z)$ as:

$$q_\mu(z) = \mathcal{N}(z; \mu, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(z - \mu)^2}{2}\right)$$

0 1 2 3 4

a) Derive the evidence lower bound (ELBO) for this particular parametrization. Simplify the parts depending on μ as far as possible.

Reminder: The ELBO for parameters θ and variational distribution q_μ is defined as

$$\mathcal{L}(\theta, q_\mu) = \mathop{\mathbb{E}}_{\boldsymbol{z} \sim q_\mu} \left[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\mu(\boldsymbol{z})\right].$$

Hint: Given a random variable X, the variance decomposition $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ can be rewritten as

$$\mathbb{E}[X^2] = \mathrm{Var}(X) + \mathbb{E}[X]^2.$$

0 1 2 3

b) Suppose θ is fixed. Derive the value of μ that maximizes the ELBO.

Problem 3: Variational Autoencoder (Version A) (2 credits)

0
1
2

We would like to define a variational autoencoder model for black-and-white images. Each image is represented as a binary vector $\boldsymbol{x} \in \{0, 1\}^N$. We define the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ as follows.

1. We obtain the distribution parameters as

$$\boldsymbol{\lambda} = \exp(f_{\boldsymbol{\theta}}(\boldsymbol{z})),$$

   where $\boldsymbol{z} \in \mathbb{R}^L$ is the latent variable and $f_{\boldsymbol{\theta}} : \mathbb{R}^L \to \mathbb{R}^N$ is the decoder neural network.

2. We obtain the conditional distribution as

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{N} \text{Exponential}(x_i|\lambda_i)$$

   where $\text{Exponential}(x|\lambda)$ is the exponential distribution with probability density function

$$\text{Exponential}(x_i|\lambda_i) = \begin{cases} \lambda_i e^{-\lambda_i x_i} & \text{if } x_i \geq 0, \\ 0 & \text{else.} \end{cases}$$

What is the main problem with the above definition of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$? Explain how we can modify the above definition to fix this problem. Justify your answer.

Problem 3: Variational Autoencoder (Version B) (2 credits)

We would like to define a variational autoencoder model for black-and-white images. Each image is represented as a binary vector $\boldsymbol{x} \in \{0, 1\}^N$. We define the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ as follows.

0
1
2

1. We obtain the distribution parameters as

$$\boldsymbol{\lambda} = \exp(f_{\boldsymbol{\theta}}(\boldsymbol{z})),$$

   where $\boldsymbol{z} \in \mathbb{R}^L$ is the latent variable and $f_{\boldsymbol{\theta}} : \mathbb{R}^L \to \mathbb{R}^N$ is the decoder neural network.

2. We obtain the conditional distribution as

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{N} \text{Exponential}(x_i|\lambda_i)$$

   where $\text{Exponential}(x|\lambda)$ is the exponential distribution with probability density function

$$\text{Exponential}(x_i|\lambda_i) = \begin{cases} \lambda_i e^{-\lambda_i x_i} & \text{if } x_i \geq 0, \\ 0 & \text{else.} \end{cases}$$

What is the main problem with the above definition of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$? Explain how we can modify the above definition to fix this problem. Justify your answer.

Problem 3: Variational Autoencoder (Version C) (2 credits)

0
1
2

We would like to define a variational autoencoder model for black-and-white images. Each image is represented as a binary vector $\boldsymbol{x} \in \{0, 1\}^N$. We define the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ as follows.

1. We obtain the distribution parameters as

$$\boldsymbol{\lambda} = \exp(f_{\boldsymbol{\theta}}(\boldsymbol{z})),$$

where $\boldsymbol{z} \in \mathbb{R}^L$ is the latent variable and $f_{\boldsymbol{\theta}} : \mathbb{R}^L \to \mathbb{R}^N$ is the decoder neural network.

2. We obtain the conditional distribution as

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{N} \text{Exponential}(x_i|\lambda_i)$$

where $\text{Exponential}(x|\lambda)$ is the exponential distribution with probability density function

$$\text{Exponential}(x_i|\lambda_i) = \begin{cases} \lambda_i e^{-\lambda_i x_i} & \text{if } x_i \geq 0, \\ 0 & \text{else.} \end{cases}$$

What is the main problem with the above definition of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$? Explain how we can modify the above definition to fix this problem. Justify your answer.

Problem 3: Variational Autoencoder (Version D) (2 credits)

0
1
2

We would like to define a variational autoencoder model for black-and-white images. Each image is represented as a binary vector $\boldsymbol{x} \in \{0, 1\}^N$. We define the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ as follows.

1. We obtain the distribution parameters as

$$\boldsymbol{\lambda} = \exp(f_{\boldsymbol{\theta}}(\boldsymbol{z})),$$

   where $\boldsymbol{z} \in \mathbb{R}^L$ is the latent variable and $f_{\boldsymbol{\theta}} : \mathbb{R}^L \to \mathbb{R}^N$ is the decoder neural network.

2. We obtain the conditional distribution as

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{N} \text{Exponential}(x_i|\lambda_i)$$

   where $\text{Exponential}(x|\lambda)$ is the exponential distribution with probability density function

$$\text{Exponential}(x_i|\lambda_i) = \begin{cases} \lambda_i e^{-\lambda_i x_i} & \text{if } x_i \geq 0, \\ 0 & \text{else.} \end{cases}$$

What is the main problem with the above definition of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$? Explain how we can modify the above definition to fix this problem. Justify your answer.

Problem 4: Robustness - Convex Relaxation (Version A) (7 credits)

0 1 2 3 4 5 6 7

In the lecture, we have derived a tight convex relaxation for the ReLU activation function. Now we want to generalize this result to the LeakyReLU activation function

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha x & \text{for } x < 0 \end{cases}$$

with $\alpha \in (0, 1)$.

Let $x, y \in \mathbb{R}$ be the variables we use to model the function's input and output, respectively. Assume we know that $l \leq x \leq u$ with $l, u \in \mathbb{R}$. Specify a set of **linear constraints** on $\begin{bmatrix} x & y \end{bmatrix}^T$ that model the **convex hull** of $\begin{bmatrix} x & \text{LeakyReLU}(x) \end{bmatrix}^T$, i.e. whose feasible region is

$$\left\{ \lambda \begin{bmatrix} x_1 \\ \text{LeakyReLU}(x_1) \end{bmatrix} + (1 - \lambda) \begin{bmatrix} x_2 \\ \text{LeakyReLU}(x_2) \end{bmatrix} \middle| \, x_1, x_2 \in [l, u] \wedge \lambda \in [0, 1] \right\}.$$

Reminder: A linear constraint is an inequality or equality relation between terms that are linear in x and y.
Hint: You will have to make a **case distinction** to account for different ranges of l and u.

Problem 4: Robustness - Convex Relaxation (Version B) (7 credits)

In the lecture, we have derived a tight convex relaxation for the ReLU activation function. Now we want to generalize this result to the LeakyReLU activation function

0
1
2
3
4
5
6
7

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha x & \text{for } x < 0 \end{cases}$$

with $\alpha \in (0, 1)$.

Let $x, y \in \mathbb{R}$ be the variables we use to model the function's input and output, respectively. Assume we know that $l \leq x \leq u$ with $l, u \in \mathbb{R}$. Specify a set of **linear constraints** on $\begin{bmatrix} x & y \end{bmatrix}^T$ that model the **convex hull** of $\begin{bmatrix} x & \text{LeakyReLU}(x) \end{bmatrix}^T$, i.e. whose feasible region is

$$\left\{ \lambda \begin{bmatrix} x_1 \\ \text{LeakyReLU}(x_1) \end{bmatrix} + (1 - \lambda) \begin{bmatrix} x_2 \\ \text{LeakyReLU}(x_2) \end{bmatrix} \,\middle|\, x_1, x_2 \in [l, u] \wedge \lambda \in [0, 1] \right\}.$$

Reminder: A linear constraint is an inequality or equality relation between terms that are linear in x and y.
Hint: You will have to make a **case distinction** to account for different ranges of l and u.

Problem 4: Robustness - Convex Relaxation (Version C) (7 credits)

0 1 2 3 4 5 6 7

In the lecture, we have derived a tight convex relaxation for the ReLU activation function. Now we want to generalize this result to the LeakyReLU activation function

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha x & \text{for } x < 0 \end{cases}$$

with $\alpha \in (0, 1)$.

Let $x, y \in \mathbb{R}$ be the variables we use to model the function's input and output, respectively. Assume we know that $l \leq x \leq u$ with $l, u \in \mathbb{R}$. Specify a set of **linear constraints** on $\begin{bmatrix} x & y \end{bmatrix}^T$ that model the **convex hull** of $\begin{bmatrix} x & \text{LeakyReLU}(x) \end{bmatrix}^T$, i.e. whose feasible region is

$$\left\{ \lambda \begin{bmatrix} x_1 \\ \text{LeakyReLU}(x_1) \end{bmatrix} + (1 - \lambda) \begin{bmatrix} x_2 \\ \text{LeakyReLU}(x_2) \end{bmatrix} \middle| \, x_1, x_2 \in [l, u] \wedge \lambda \in [0, 1] \right\}.$$

Reminder: A linear constraint is an inequality or equality relation between terms that are linear in x and y.
Hint: You will have to make a **case distinction** to account for different ranges of l and u.

Problem 4: Robustness - Convex Relaxation (Version D) (7 credits)

In the lecture, we have derived a tight convex relaxation for the ReLU activation function. Now we want to generalize this result to the LeakyReLU activation function

0
1
2
3
4
5
6
7

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha x & \text{for } x < 0 \end{cases}$$

with $\alpha \in (0, 1)$.

Let $x, y \in \mathbb{R}$ be the variables we use to model the function's input and output, respectively. Assume we know that $l \leq x \leq u$ with $l, u \in \mathbb{R}$. Specify a set of **linear constraints** on $\begin{bmatrix} x & y \end{bmatrix}^T$ that model the **convex hull** of $\begin{bmatrix} x & \text{LeakyReLU}(x) \end{bmatrix}^T$, i.e. whose feasible region is

$$\left\{ \lambda \begin{bmatrix} x_1 \\ \text{LeakyReLU}(x_1) \end{bmatrix} + (1 - \lambda) \begin{bmatrix} x_2 \\ \text{LeakyReLU}(x_2) \end{bmatrix} \middle| \, x_1, x_2 \in [l, u] \wedge \lambda \in [0, 1] \right\}.$$

Reminder: A linear constraint is an inequality or equality relation between terms that are linear in x and y.
Hint: You will have to make a **case distinction** to account for different ranges of l and u.

Problem 5: Markov Chain Language Model (Version A) (7 credits)

We want to use a Markov chain to model a very simple language consisting of the 4 words I, orange, like, eat. While the words are borrowed from the English language, our simple language is not bound to its grammatical rules. The words map to the Markov chain parameters as follows.

$$\boldsymbol{\pi} = \begin{array}{r} \text{I} \\ \text{orange} \\ \text{like} \\ \text{eat} \end{array} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{r} \\ \text{I} \\ \text{orange} \\ \text{like} \\ \text{eat} \end{array} \begin{array}{c} \begin{array}{cccc} \text{I} & \text{orange} & \text{like} & \text{eat} \end{array} \\ \begin{pmatrix} A_{11} & & \cdots & A_{14} \\ \vdots & \ddots & & \vdots \\ & & & \\ A_{41} & & \cdots & A_{44} \end{pmatrix} \end{array}$$

$\boldsymbol{A}_{ij}$ specifies the probability of transitioning from state i to state j.

0 1 2 a) Fit the Markov chain to the following dataset of example sentences by computing the most likely parameters.

- I like orange
- I eat orange
- orange eat orange
- I like I

For the remaining problems assume that you are given the following Markov chain parameters that were fit to a larger dataset.

$$\boldsymbol{\pi} = \begin{array}{r} \text{I} \\ \text{orange} \\ \text{like} \\ \text{eat} \end{array} \begin{pmatrix} 4/6 \\ 2/6 \\ 0 \\ 0 \end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{r} \\ \text{I} \\ \text{orange} \\ \text{like} \\ \text{eat} \end{array} \begin{array}{c} \begin{array}{cccc} \text{I} & \text{orange} & \text{like} & \text{eat} \end{array} \\ \begin{pmatrix} 0 & 1/6 & 3/6 & 2/6 \\ 0 & 0 & 2/6 & 4/6 \\ 1/6 & 3/6 & 1/6 & 1/6 \\ 1/6 & 5/6 & 0 & 0 \end{pmatrix} \end{array}$$

0 1 2 b) Which of the following two sentences is more likely according to the model? Justify your answer.

1) I like orange

2) orange eat I

c) Given that the 3rd word X_3 of a sentence is orange, compute the (unnormalized) probability distribution over the previous word X_2. Justify your answer.

☐ 0
☐ 1
☐ 2
☐ 3

Problem 5: Markov Chain Language Model (Version B) (7 credits)

We want to use a Markov chain to model a very simple language consisting of the 4 words I, orange, see, like. While the words are borrowed from the English language, our simple language is not bound to its grammatical rules. The words map to the Markov chain parameters as follows.

$$\boldsymbol{\pi} = \begin{array}{r} \text{I} \\ \text{orange} \\ \text{see} \\ \text{like} \end{array} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{r|cccc} & \text{I} & \text{orange} & \text{see} & \text{like} \\ \text{I} & A_{11} & & \cdots & A_{14} \\ \text{orange} & \vdots & \ddots & & \vdots \\ \text{see} & & & & \\ \text{like} & A_{41} & & \cdots & A_{44} \end{array}$$

$\boldsymbol{A}_{ij}$ specifies the probability of transitioning from state i to state j.

0
1
2

a) Fit the Markov chain to the following dataset of example sentences by computing the most likely parameters.

- I see I
- I like orange
- orange like orange
- I see orange

For the remaining problems assume that you are given the following Markov chain parameters that were fit to a larger dataset.

$$\boldsymbol{\pi} = \begin{array}{r} \text{I} \\ \text{orange} \\ \text{see} \\ \text{like} \end{array} \begin{pmatrix} 4/6 \\ 2/6 \\ 0 \\ 0 \end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{r} \\ \text{I} \\ \text{orange} \\ \text{see} \\ \text{like} \end{array} \begin{array}{c} \begin{array}{cccc} \text{I} & \text{orange} & \text{see} & \text{like} \end{array} \\ \begin{pmatrix} 0 & 1/6 & 3/6 & 2/6 \\ 0 & 0 & 2/6 & 4/6 \\ 1/6 & 3/6 & 1/6 & 1/6 \\ 1/6 & 5/6 & 0 & 0 \end{pmatrix} \end{array}$$

0
1
2

b) Which of the following two sentences is more likely according to the model? Justify your answer.

1) I see orange

2) orange like I

c) Given that the 3rd word X_3 of a sentence is orange, compute the (unnormalized) probability distribution over the previous word X_2. Justify your answer.

0
1
2
3

Problem 5: Markov Chain Language Model (Version C) (7 credits)

We want to use a Markov chain to model a very simple language consisting of the 4 words you, apple, see, like. While the words are borrowed from the English language, our simple language is not bound to its grammatical rules. The words map to the Markov chain parameters as follows.

$$\boldsymbol{\pi} = \begin{array}{r}\text{you}\\ \text{apple}\\ \text{see}\\ \text{like}\end{array}\begin{pmatrix}\pi_1\\ \pi_2\\ \pi_3\\ \pi_4\end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{r}\\ \text{you}\\ \text{apple}\\ \text{see}\\ \text{like}\end{array}\begin{array}{c}\begin{array}{cccc}\text{you} & \text{apple} & \text{see} & \text{like}\end{array}\\ \begin{pmatrix}A_{11} & & \cdots & A_{14}\\ \vdots & \ddots & & \vdots\\ & & & \\ A_{41} & & \cdots & A_{44}\end{pmatrix}\end{array}$$

$\boldsymbol{A}_{ij}$ specifies the probability of transitioning from state i to state j.

0
1
2

a) Fit the Markov chain to the following dataset of example sentences by computing the most likely parameters.

- you see apple
- you like apple
- apple like apple
- you see you

For the remaining problems assume that you are given the following Markov chain parameters that were fit to a larger dataset.

$$\boldsymbol{\pi} = \begin{array}{r}\text{you}\\ \text{apple}\\ \text{see}\\ \text{like}\end{array}\begin{pmatrix}^4/_6\\ ^2/_6\\ 0\\ 0\end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{r}\\ \text{you}\\ \text{apple}\\ \text{see}\\ \text{like}\end{array}\begin{array}{c}\begin{array}{cccc}\text{you} & \text{apple} & \text{see} & \text{like}\end{array}\\ \begin{pmatrix}0 & ^1/_6 & ^3/_6 & ^2/_6\\ 0 & 0 & ^2/_6 & ^4/_6\\ ^1/_6 & ^3/_6 & ^1/_6 & ^1/_6\\ ^1/_6 & ^5/_6 & 0 & 0\end{pmatrix}\end{array}$$

0
1
2

b) Which of the following two sentences is more likely according to the model? Justify your answer.

1) you see apple

2) apple like you

c) Given that the 3rd word X_3 of a sentence is apple, compute the (unnormalized) probability distribution over the previous word X_2. Justify your answer.

0
1
2
3

Problem 5: Markov Chain Language Model (Version D) (7 credits)

We want to use a Markov chain to model a very simple language consisting of the 4 words they, apple, like, eat. While the words are borrowed from the English language, our simple language is not bound to its grammatical rules. The words map to the Markov chain parameters as follows.

$$\boldsymbol{\pi} = \begin{array}{r} \text{they} \\ \text{apple} \\ \text{like} \\ \text{eat} \end{array} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{rc} & \begin{array}{cccc} \text{they} & \text{apple} & \text{like} & \text{eat} \end{array} \\ \begin{array}{r} \text{they} \\ \text{apple} \\ \text{like} \\ \text{eat} \end{array} & \begin{pmatrix} A_{11} & & \cdots & A_{14} \\ \vdots & \ddots & & \vdots \\ & & & \\ A_{41} & & \cdots & A_{44} \end{pmatrix} \end{array}$$

$\boldsymbol{A}_{ij}$ specifies the probability of transitioning from state i to state j.

0 1 2

a) Fit the Markov chain to the following dataset of example sentences by computing the most likely parameters.

- they like they
- they eat apple
- apple eat apple
- they like apple

For the remaining problems assume that you are given the following Markov chain parameters that were fit to a larger dataset.

$$\boldsymbol{\pi} = \begin{array}{r} \text{they} \\ \text{apple} \\ \text{like} \\ \text{eat} \end{array} \begin{pmatrix} 4/6 \\ 2/6 \\ 0 \\ 0 \end{pmatrix} \qquad \boldsymbol{A} = \begin{array}{rc} & \begin{array}{cccc} \text{they} & \text{apple} & \text{like} & \text{eat} \end{array} \\ \begin{array}{r} \text{they} \\ \text{apple} \\ \text{like} \\ \text{eat} \end{array} & \begin{pmatrix} 0 & 1/6 & 3/6 & 2/6 \\ 0 & 0 & 2/6 & 4/6 \\ 1/6 & 3/6 & 1/6 & 1/6 \\ 1/6 & 5/6 & 0 & 0 \end{pmatrix} \end{array}$$

0 1 2

b) Which of the following two sentences is more likely according to the model? Justify your answer.

1) they like apple

2) apple eat they

c) Given that the 3rd word X_3 of a sentence is apple, compute the (unnormalized) probability distribution over the previous word X_2. Justify your answer.

0
1
2
3

Problem 6: Neural Sequence Models (Version A) (4 credits)

0
1
2
3
4

We want to find out the limitations of our neural models for sequential data. To do that, we construct a dataset where the inputs are multiple sequences of $n > 10$ numbers $[x_1, x_2, \dots, x_n]$, $x_i \in \mathbb{R}$, where the corresponding target for each sequence is $y = x_1 + x_n$. We use four different encoders:

1. RNN with positional encoding
2. Transformer with positional encoding
3. Transformer without positional encoding
4. Dilated causal convolution with 2 hidden layers. We set dilation size to 2.

After processing the sequence with the above described encoders, we have access to hidden states $\boldsymbol{h}_i \in \mathbb{R}^h$, corresponding to the i-th place in the sequence. We use the last hidden state $\boldsymbol{h}_n$ to make the prediction. For each of the four encoders, write down if they can learn the given task *in theory*. Justify your answer. For those encoders that can learn it, what issues might you encounter in practice?

Problem 6: Neural Sequence Models (Version B) (4 credits)

0
1
2
3
4

We want to find out the limitations of our neural models for sequential data. To do that, we construct a dataset where the inputs are multiple sequences of $n > 10$ numbers $[x_1, x_2, \dots, x_n]$, $x_i \in \mathbb{R}$, where the corresponding target for each sequence is $y = x_1 + x_n$. We use four different encoders:

1. Recurrent neural network
2. Transformer without positional encoding
3. Transformer with positional encoding
4. Sliding window neural network that takes $[x_{i-k}, \dots, x_{i-1}, x_i]$ and outputs $\boldsymbol{h}_i \in \mathbb{R}^h$, for each i. We set $k = 5$

After processing the sequence with the above described encoders, we have access to hidden states $\boldsymbol{h}_i \in \mathbb{R}^h$, corresponding to the i-th place in the sequence. We use the last hidden state $\boldsymbol{h}_n$ to make the prediction. For each of the four encoders, write down if they can learn the given task *in theory*. Justify your answer. For those encoders that can learn it, what issues might you encounter in practice?

Problem 6: Neural Sequence Models (Version C) (4 credits)

0
1
2
3
4

We want to find out the limitations of our neural models for sequential data. To do that, we construct a dataset where the inputs are multiple sequences of $n > 10$ numbers $[x_1, x_2, \dots, x_n]$, $x_i \in \mathbb{R}$, where the corresponding target for each sequence is $y = x_1 + x_n$. We use four different encoders:

1. Transformer <u>with</u> positional encoding
2. Transformer <u>without</u> positional encoding
3. Multilayer neural network that takes vector in $\mathbb{R}^n$ as input (all numbers concatenated) and outputs $\mathbb{R}^{n \times h}$
4. Recurrent neural network

After processing the sequence with the above described encoders, we have access to hidden states $\boldsymbol{h}_i \in \mathbb{R}^h$, corresponding to the i-th place in the sequence. We use the last hidden state $\boldsymbol{h}_n$ to make the prediction. For each of the four encoders, write down if they can learn the given task *in theory*. Justify your answer. For those encoders that can learn it, what issues might you encounter in practice?

Problem 6: Neural Sequence Models (Version D) (4 credits)

0
1
2
3
4

We want to find out the limitations of our neural models for sequential data. To do that, we construct a dataset where the inputs are multiple sequences of $n > 10$ numbers $[x_1, x_2, \dots, x_n]$, $x_i \in \mathbb{R}$, where the corresponding target for each sequence is $y = x_1 + x_n$. We use four different encoders:

1. Recurrent neural network
2. Dilated causal convolution with 2 hidden layers. We set dilation size to 2.
3. Transformer <u>with</u> positional encoding
4. Transformer <u>without</u> positional encoding

After processing the sequence with the above described encoders, we have access to hidden states $\boldsymbol{h}_i \in \mathbb{R}^h$, corresponding to the i-th place in the sequence. We use the last hidden state $\boldsymbol{h}_n$ to make the prediction. For each of the four encoders, write down if they can learn the given task *in theory*. Justify your answer. For those encoders that can learn it, what issues might you encounter in practice?

Problem 7: Temporal Point Process (Version A) (6 credits)

We fit a homogeneous Poisson process with intensity parameter μ to model event occurrences in a time interval $[0, T]$. We have observed a single sequence that contains n points $\{t_1, t_2, \ldots, t_n\}$, $t_i \in [0, T]$.

0 1 2 3 4

a) Derive the maximum likelihood estimate of the parameter μ.

0 1 2

b) Suppose we install a sensor next to a busy road that records the times when cars drive by. We model the times as described above, using the events from the whole day as one sequence. We estimate μ using data we collected in one year. Our task is to find the least busy 2 hour interval in each day to close down the road for maintenance. Can we use the homogeneous Poisson process to achieve this? If not, can you suggest an alternative model? Justify your answer.

Problem 7: Temporal Point Process (Version B) (6 credits)

We fit a homogeneous Poisson process with intensity parameter μ to model event occurrences in a time interval $[0, 5]$. We have observed a single sequence $\{0.7, 0.8, 1.5, 2.3, 4.7\}$.

a) Derive the maximum likelihood estimate of the parameter μ.

0
1
2
3
4

b) Suppose we install a sensor next to a busy road that records the times when cars drive by. We model the times as described above, using the events from the whole day as one sequence. For each day of the week we estimate the parameter μ using data we collected in one year. That means we have $\mu_{\text{Mon}}, \mu_{\text{Tue}}, \dots, \mu_{\text{Sun}}$, each μ corresponding to one day of the week. Our task is to find the least busy day of the week to close down the road for maintenance. Can we use the homogeneous Poisson process to achieve this? If not, can you suggest an alternative model? Justify your answer.

0
1
2

Problem 7: Temporal Point Process (Version C) (6 credits)

We fit a homogeneous Poisson process with intensity parameter μ to model event occurrences in a time interval $[0, 2]$. We have observed a single sequence $\{0.1, 0.8, 1.3, 1.5, 1.7, 1.9\}$.

0
1
2
3
4

a) Derive the maximum likelihood estimate of the parameter μ.

0
1
2

b) Suppose we install a sensor next to a busy road that records the times when cars drive by. We model the times as described above, using the events from the whole day as one sequence. Using our model, we want to estimate the probability that less than 100 cars will pass our sensor in a day. Can we use the homogeneous Poisson process to achieve this? If not, can you suggest an alternative model? Justify your answer.

Problem 7: Temporal Point Process (Version D) (6 credits)

We fit a homogeneous Poisson process with intensity parameter μ to model event occurrences in a time interval $[3, 13]$. We have observed a single sequence $\{3.5, 4.3, 4.5, 7.1, 8.3\}$.

a) Derive the maximum likelihood estimate of the parameter μ.

0
1
2
3
4

b) Suppose we install a sensor next to a busy road that records the times when cars drive by. We model the times as described above, using the events from the whole day as one sequence. Using our model, we want to answer whether fast vehicles get stuck behind slower vehicles. That is, we want to see if observing one vehicle leads to a few more following behind it. Can we use the homogeneous Poisson process to achieve this? If not, can you suggest an alternative model? Justify your answer.

0
1
2

Problem 8: Clustering (Version A) (6 credits)

0
1
2
3
4
5
6

We consider the graph $G = (E, V)$ with adjacency matrix $\boldsymbol{A}$ where the nodes are separated into two clusters, C and $\bar{C}$. We define the associated random walk $\Pr(X_{t+1} = j | X_t = i) = \frac{A_{ij}}{d_i}$ where $d_i = \sum_j A_{ij}$ is the degree of node i and $\Pr(X_0 = i) = \frac{d_i}{\mathrm{vol}(V)}$ is the starting distribution where $\mathrm{vol}(V) = \sum_{i \in V} d_i$ is the volume of the set of nodes V. We define the probability to transition from cluster C to cluster $\bar{C}$ in the first random walk step as $\Pr(\bar{C} \mid C) = \Pr(X_1 \in \bar{C} \mid X_0 \in C)$ and vice versa. Show that the normalized cut satisfies the equation

$$\mathrm{Ncut}(C, \bar{C}) = \Pr(\bar{C} \mid C) + \Pr(C \mid \bar{C}).$$

Reminder: The normalized cut of an undirected graph is defined as

$$\mathrm{Ncut}(C, \bar{C}) = \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(C)} + \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(\bar{C})}.$$

Problem 8: Clustering (Version B) (6 credits)

We consider the graph $G = (E, V)$ with adjacency matrix $\boldsymbol{A}$ where the nodes are separated into two clusters, C and $\bar{C}$. We define the associated random walk $\Pr(X_{t+1} = j | X_t = i) = \frac{A_{ij}}{d_i}$ where $d_i = \sum_j A_{ij}$ is the degree of node i and $\Pr(X_0 = i) = \frac{d_i}{\text{vol}(V)}$ is the starting distribution where $\text{vol}(V) = \sum_{i \in V} d_i$ is the volume of the set of nodes V. We define the probability to transition from cluster C to cluster $\bar{C}$ in the first random walk step as $\Pr(\bar{C} \mid C) = \Pr(X_1 \in \bar{C} \mid X_0 \in C)$ and vice versa. Show that the normalized cut satisfies the equation

0
1
2
3
4
5
6

$$\text{Ncut}(C, \bar{C}) = \Pr(\bar{C} \mid C) + \Pr(C \mid \bar{C}).$$

Reminder: The normalized cut of an undirected graph is defined as

$$\text{Ncut}(C, \bar{C}) = \frac{\text{cut}(C, \bar{C})}{\text{vol}(C)} + \frac{\text{cut}(C, \bar{C})}{\text{vol}(\bar{C})}.$$

Problem 8: Clustering (Version C) (6 credits)

0
1
2
3
4
5
6

We consider the graph $G = (E, V)$ with adjacency matrix $\boldsymbol{A}$ where the nodes are separated into two clusters, C and $\bar{C}$. We define the associated random walk $\Pr(X_{t+1} = j | X_t = i) = \frac{A_{ij}}{d_i}$ where $d_i = \sum_j A_{ij}$ is the degree of node i and $\Pr(X_0 = i) = \frac{d_i}{\mathrm{vol}(V)}$ is the starting distribution where $\mathrm{vol}(V) = \sum_{i \in V} d_i$ is the volume of the set of nodes V. We define the probability to transition from cluster C to cluster $\bar{C}$ in the first random walk step as $\Pr(\bar{C} \mid C) = \Pr(X_1 \in \bar{C} \mid X_0 \in C)$ and vice versa. Show that the normalized cut satisfies the equation

$$\mathrm{Ncut}(C, \bar{C}) = \Pr(\bar{C} \mid C) + \Pr(C \mid \bar{C}).$$

Reminder: The normalized cut of an undirected graph is defined as

$$\mathrm{Ncut}(C, \bar{C}) = \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(C)} + \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(\bar{C})}.$$

Problem 8: Clustering (Version D) (6 credits)

We consider the graph $G = (E, V)$ with adjacency matrix $\boldsymbol{A}$ where the nodes are separated into two clusters, C and $\bar{C}$. We define the associated random walk $\Pr(X_{t+1} = j | X_t = i) = \frac{A_{ij}}{d_i}$ where $d_i = \sum_j A_{ij}$ is the degree of node i and $\Pr(X_0 = i) = \frac{d_i}{\text{vol}(V)}$ is the starting distribution where $\text{vol}(V) = \sum_{i \in V} d_i$ is the volume of the set of nodes V. We define the probability to transition from cluster C to cluster $\bar{C}$ in the first random walk step as $\Pr(\bar{C} \mid C) = \Pr(X_1 \in \bar{C} \mid X_0 \in C)$ and vice versa. Show that the normalized cut satisfies the equation

0
1
2
3
4
5
6

$$\text{Ncut}(C, \bar{C}) = \Pr(\bar{C} \mid C) + \Pr(C \mid \bar{C}).$$

Reminder: The normalized cut of an undirected graph is defined as

$$\text{Ncut}(C, \bar{C}) = \frac{\text{cut}(C, \bar{C})}{\text{vol}(C)} + \frac{\text{cut}(C, \bar{C})}{\text{vol}(\bar{C})}.$$

Problem 9: Embeddings & Ranking (Version A) (6 credits)

We consider a graph $G = (V, E)$ with adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ where $A_{ij} = \mathbb{1}_{(i,j) \in E}$ indicates if an edge exist between node i and node j in the graph G. The node features are represented by the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times D}$. We consider the three following models $M_k, k \in \{1, 2, 3\}$ which produce node embeddings $\boldsymbol{E}_k = M_k(G, \boldsymbol{X}) \in \mathbb{R}^{n \times D'}$. The vector $\boldsymbol{E}_k[i, :] \in \mathbb{R}^{D'}$ denotes the embedding of node i for model M_k:

- M_1: Node2Vec.

- M_2: Mean of Graph2Gauss i.e. $\boldsymbol{E}_2[i, :] = \boldsymbol{\mu}_i$ where the Graph2Gauss mapping transforms node i into the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, diag(\boldsymbol{\sigma}_i))$.

- M_3: Spectral embedding with k smallest eigenvectors.

0 1 2 3

a) We modify the attributed graph such that all nodes have the same features, i.e. the adjacency matrix is $\boldsymbol{A}' = \boldsymbol{A}$ and the new node attributes are $\boldsymbol{X}'$ such that $\boldsymbol{X}'[i, :] = \boldsymbol{X}'[j, :]$ for all (i, j). For which model will the new node embeddings $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graphs $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify your answer.

0 1 2 3

b) We modify the attributed graph such that the graph is a clique, i.e. the new adjacency matrix is $\boldsymbol{A}' = \boldsymbol{1} - \boldsymbol{I}$ where $\boldsymbol{1}$ is the all-ones matrix, and the node attributes are $\boldsymbol{X}' = \boldsymbol{X}$. For which model will the new node embeddings $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graph $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify your answer.

Problem 9: Embeddings & Ranking (Version B) (6 credits)

We consider a graph $G = (V, E)$ with adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ where $A_{ij} = \mathbb{1}_{(i,j) \in E}$ indicates if an edge exist between node i and node j in the graph G. The node features are represented by the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times D}$. We consider the three following models $M_k, k \in \{1, 2, 3\}$ which produce node embeddings $\boldsymbol{E}_k = M_k(G, \boldsymbol{X}) \in \mathbb{R}^{n \times D'}$. The vector $\boldsymbol{E}_k[i, :] \in \mathbb{R}^{D'}$ denotes the embedding of node i for model M_k:

- M_1: Node2Vec.
- M_2: Mean of Graph2Gauss i.e. $\boldsymbol{E}_2[i, :] = \boldsymbol{\mu}_i$ where the Graph2Gauss mapping transforms node i into the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, diag(\boldsymbol{\sigma}_i))$.
- M_3: Spectral embedding with k smallest eigenvectors.

0 1 2 3

a) We modify the attributed graph such that all nodes have the same features, i.e. the adjacency matrix is $\boldsymbol{A}' = \boldsymbol{A}$ and the new node attributes are $\boldsymbol{X}'$ such that $\boldsymbol{X}'[i, :] = \boldsymbol{X}'[j, :]$ for all (i, j). For which model will the new node embeddings $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graphs $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify your answer.

0 1 2 3

b) We modify the attributed graph such that the graph is a clique, i.e. the new adjacency matrix is $\boldsymbol{A}' = \boldsymbol{1} - \boldsymbol{I}$ where $\boldsymbol{1}$ is the all-ones matrix, and the node attributes are $\boldsymbol{X}' = \boldsymbol{X}$. For which model will the new node embeddings $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graph $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify your answer.

Problem 9: Embeddings & Ranking (Version C) (6 credits)

We consider a graph $G = (V, E)$ with adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ where $A_{ij} = \mathbb{1}_{(i,j) \in E}$ indicates if an edge exist between node i and node j in the graph G. The node features are represented by the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times D}$. We consider the three following models $M_k, k \in \{1, 2, 3\}$ which produce node embeddings $\boldsymbol{E}_k = M_k(G, \boldsymbol{X}) \in \mathbb{R}^{n \times D'}$. The vector $\boldsymbol{E}_k[i, :] \in \mathbb{R}^{D'}$ denotes the embedding of node i for model M_k:

- M_1: Node2Vec.

- M_2: Mean of Graph2Gauss i.e. $\boldsymbol{E}_2[i, :] = \boldsymbol{\mu}_i$ where the Graph2Gauss mapping transforms node i into the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, diag(\boldsymbol{\sigma}_i))$.

- M_3: Spectral embedding with k largest eigenvectors.

0 a) We modify the attributed graph such that all nodes have the same features, i.e. the adjacency matrix is $\boldsymbol{A}' = \boldsymbol{A}$ and
1 the new node attributes are $\boldsymbol{X}'$ such that $\boldsymbol{X}'[i, :] = \boldsymbol{X}'[j, :]$ for all (i, j). For which model will the new node embeddings
2 $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graphs $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify
3 your answer.

0 b) We modify the attributed graph such that the graph is a clique, i.e. the new adjacency matrix is $\boldsymbol{A}' = \boldsymbol{1} - \boldsymbol{I}$
1 where $\boldsymbol{1}$ is the all-ones matrix, and the node attributes are $\boldsymbol{X}' = \boldsymbol{X}$. For which model will the new node embeddings
2 $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graph $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify
3 your answer.

Problem 9: Embeddings & Ranking (Version D) (6 credits)

We consider a graph $G = (V, E)$ with adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ where $A_{ij} = \mathbb{1}_{(i,j) \in E}$ indicates if an edge exist between node i and node j in the graph G. The node features are represented by the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times D}$. We consider the three following models $M_k, k \in \{1, 2, 3\}$ which produce node embeddings $\boldsymbol{E}_k = M_k(G, \boldsymbol{X}) \in \mathbb{R}^{n \times D'}$. The vector $\boldsymbol{E}_k[i, :] \in \mathbb{R}^{D'}$ denotes the embedding of node i for model M_k:

- M_1: Node2Vec.
- M_2: Mean of Graph2Gauss i.e. $\boldsymbol{E}_2[i, :] = \boldsymbol{\mu}_i$ where the Graph2Gauss mapping transforms node i into the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, diag(\boldsymbol{\sigma}_i))$.
- M_3: Spectral embedding with k largest eigenvectors.

0 1 2 3

a) We modify the attributed graph such that all nodes have the same features, i.e. the adjacency matrix is $\boldsymbol{A}' = \boldsymbol{A}$ and the new node attributes are $\boldsymbol{X}'$ such that $\boldsymbol{X}'[i, :] = \boldsymbol{X}'[j, :]$ for all (i, j). For which model will the new node embeddings $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graphs $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify your answer.

0 1 2 3

b) We modify the attributed graph such that the graph is a clique, i.e. the new adjacency matrix is $\boldsymbol{A}' = \boldsymbol{1} - \boldsymbol{I}$ where $\boldsymbol{1}$ is the all-ones matrix, and the node attributes are $\boldsymbol{X}' = \boldsymbol{X}$. For which model will the new node embeddings $\boldsymbol{E}'_k = M_k(G', \boldsymbol{X}')$ be different from the embeddings obtained with the original attributed graph $\boldsymbol{E}_k = M_k(G, \boldsymbol{X})$? Justify your answer.

Problem 10: Semi-Supervised Learning (Version A) (6 credits)

In this problem, we consider a Stochastic Block Model with two ground-truth communities C_1 and C_2. The SBM has community proportions $\boldsymbol{\pi}$ and edge probability $\boldsymbol{\nu}$ given as

$$\boldsymbol{\pi} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\nu} = \begin{bmatrix} 0.2 & 0.9 \\ 0.9 & 0.2 \end{bmatrix}.$$

We consider a sampled graph G with n nodes from the SBM defined as above where the node labels are defined as the ground-truth communities of the SBM. The task is now to predict the labels of all nodes of the graphs where only a fraction of the node labels is available for training.

0
1
2
3

a) Do you expect label propagation with the optimization problem

$$\min \sum_{i,j} w_{ij}(\boldsymbol{y}_i - \boldsymbol{y}_j)^T(\boldsymbol{y}_i - \boldsymbol{y}_j)$$

to work well for this task? If not, propose a modification of the optimization problem which would solve the problem. Justify your answer.

0
1
2
3

b) The nodes are now assigned node features sampled as

$$\boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_1 \qquad \text{and} \qquad \boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_2.$$

We define $N(v)$ as the 1-hop neighborhood of node v. Do you expect a one-layer GNN with the message passing step $\boldsymbol{m}_v^{(1)}(\boldsymbol{h}_1^{(0)}, \dots, \boldsymbol{h}_n^{(0)}) = \frac{1}{|N(v)|}\sum_{u \in N(v)}\left(\boldsymbol{W}\boldsymbol{h}_u^{(0)} + \boldsymbol{b}\right)$ and the update step $\boldsymbol{h}_v^{(1)} = \mathrm{ReLU}(\boldsymbol{Q}\boldsymbol{h}_v^{(0)} + \boldsymbol{p} + \boldsymbol{m}_v^{(1)})$ to work well for this task? If not, propose a modification to the message passing and/or update step that would solve the problem. Justify your answer.

Problem 10: Semi-Supervised Learning (Version B) (6 credits)

In this problem, we consider a Stochastic Block Model with two ground-truth communities C_1 and C_2. The SBM has community proportions π and edge probability ν given as

$$\pi = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \nu = \begin{bmatrix} 0.2 & 0.9 \\ 0.9 & 0.2 \end{bmatrix}.$$

We consider a sampled graph G with n nodes from the SBM defined as above where the node labels are defined as the ground-truth communities of the SBM. The task is now to predict the labels of all nodes of the graphs where only a fraction of the node labels is available for training.

a) Do you expect label propagation with the optimization problem

0 1 2 3

$$\min \sum_{i,j} w_{ij}(\boldsymbol{y}_i - \boldsymbol{y}_j)^T(\boldsymbol{y}_i - \boldsymbol{y}_j)$$

to work well for this task? If not, propose a modification of the optimization problem which would solve the problem. Justify your answer.

b) The nodes are now assigned node features sampled as

0 1 2 3

$$\boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_1 \qquad \text{and} \qquad \boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_2.$$

We define $N(v)$ as the 1-hop neighborhood of node v. Do you expect a one-layer GNN with the message passing step $\boldsymbol{m}_v^{(1)}(\boldsymbol{h}_1^{(0)}, \dots, \boldsymbol{h}_n^{(0)}) = \frac{1}{|N(v)|} \sum_{u \in N(v)} \left(\boldsymbol{W}\boldsymbol{h}_u^{(0)} + \boldsymbol{b}\right)$ and the update step $\boldsymbol{h}_v^{(1)} = \text{ReLU}(\boldsymbol{Q}\boldsymbol{h}_v^{(0)} + \boldsymbol{p} + \boldsymbol{m}_v^{(1)})$ to work well for this task? If not, propose a modification to the message passing and/or update step that would solve the problem. Justify your answer.

Problem 10: Semi-Supervised Learning (Version C) (6 credits)

In this problem, we consider a Stochastic Block Model with two ground-truth communities C_1 and C_2. The SBM has community proportions $\boldsymbol{\pi}$ and edge probability $\boldsymbol{\nu}$ given as

$$\boldsymbol{\pi} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\nu} = \begin{bmatrix} 0.1 & 0.8 \\ 0.8 & 0.1 \end{bmatrix}.$$

We consider a sampled graph G with n nodes from the SBM defined as above where the node labels are defined as the ground-truth communities of the SBM. The task is now to predict the labels of all nodes of the graphs where only a fraction of the node labels is available for training.

0 1 2 3

a) Do you expect label propagation with the optimization problem

$$\min \sum_{i,j} w_{ij} (\boldsymbol{y}_i - \boldsymbol{y}_j)^T (\boldsymbol{y}_i - \boldsymbol{y}_j)$$

to work well for this task? If not, propose a modification of the optimization problem which would solve the problem. Justify your answer.

0 1 2 3

b) The nodes are now assigned node features sampled as

$$\boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_1 \qquad \text{and} \qquad \boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_2.$$

We define $N(v)$ as the 1-hop neighborhood of node v. Do you expect a one-layer GNN with the message passing step $\boldsymbol{m}_v^{(1)}(\boldsymbol{h}_1^{(0)}, \dots, \boldsymbol{h}_n^{(0)}) = \frac{1}{|N(v)|} \sum_{u \in N(v)} \left(\boldsymbol{W}\boldsymbol{h}_u^{(0)} + \boldsymbol{b}\right)$ and the update step $\boldsymbol{h}_v^{(1)} = \text{ReLU}(\boldsymbol{Q}\boldsymbol{h}_v^{(0)} + \boldsymbol{p} + \boldsymbol{m}_v^{(1)})$ to work well for this task? If not, propose a modification to the message passing and/or update step that would solve the problem. Justify your answer.

Problem 10: Semi-Supervised Learning (Version D) (6 credits)

In this problem, we consider a Stochastic Block Model with two ground-truth communities C_1 and C_2. The SBM has community proportions π and edge probability ν given as

$$\pi = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad \text{and} \quad \nu = \begin{bmatrix} 0.1 & 0.8 \\ 0.8 & 0.1 \end{bmatrix}.$$

We consider a sampled graph G with n nodes from the SBM defined as above where the node labels are defined as the ground-truth communities of the SBM. The task is now to predict the labels of all nodes of the graphs where only a fraction of the node labels is available for training.

a) Do you expect label propagation with the optimization problem

0 1 2 3

$$\min \sum_{i,j} w_{ij}(\boldsymbol{y}_i - \boldsymbol{y}_j)^T(\boldsymbol{y}_i - \boldsymbol{y}_j)$$

to work well for this task? If not, propose a modification of the optimization problem which would solve the problem. Justify your answer.

b) The nodes are now assigned node features sampled as

0 1 2 3

$$\boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_1 \qquad \text{and} \qquad \boldsymbol{h}_v^{(0)} \sim \mathcal{N}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{ for } v \in C_2.$$

We define $N(v)$ as the 1-hop neighborhood of node v. Do you expect a one-layer GNN with the message passing step $\boldsymbol{m}_v^{(1)}(\boldsymbol{h}_1^{(0)}, \dots, \boldsymbol{h}_n^{(0)}) = \frac{1}{|N(v)|} \sum_{u \in N(v)} \left(\boldsymbol{W}\boldsymbol{h}_u^{(0)} + \boldsymbol{b}\right)$ and the update step $\boldsymbol{h}_v^{(1)} = \text{ReLU}(\boldsymbol{Q}\boldsymbol{h}_v^{(0)} + \boldsymbol{p} + \boldsymbol{m}_v^{(1)})$ to work well for this task? If not, propose a modification to the message passing and/or update step that would solve the problem. Justify your answer.