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

Esolution
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

Exam:	IN2323 / Endterm	**Date:**	Friday 19th August, 2022
Examiner:	Prof. Dr. Stephan Günnemann	**Time:**	08:15 – 09:30

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9
I									

Working instructions

- This exam consists of **16 pages** with a total of **9 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 72 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - one A4 sheet of handwritten notes (two sides, not digitally written and printed).
- **No other material (e.g. books, cell phones, calculators) is allowed!**
- Physically turn off all electronic devices, put them into your bag and close the bag.
- There is scratch paper at the end of the exam (after problem 9).
- Write your answers only in the provided solution boxes or the scratch paper.
- If you solve a task on the scratch paper, clearly reference it in the main solution box.
- All sheets (including scratch paper) have to be returned at the end.
- **Only use a black or a blue pen (no pencils, red or greens pens!)**
- **For problems that say "Justify your answer" you only get points if you provide a valid explanation.**
- **For problems that say "Derive" you only get points if you provide a valid mathematical derivation.**
- **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", "Derive" or "Prove", it is sufficient to only provide the correct answer.

Left room from __________ to __________ / Early submission at __________

Problem 1 Generative models (6 credits)

Recall the variational autoencoder (VAE), which can be summarized by the following pseudocode

$$
\begin{aligned}
\mu, \sigma &= f_\theta(\mathbf{x}) \\
\epsilon &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
\mathbf{z} &= \epsilon * \sigma + \mu \\
\tilde{\mathbf{x}} &= g_\phi(\mathbf{z}),
\end{aligned}
$$

and is trained to model a distribution $p(\mathbf{x})$ via maximization of the evidence lower bound.
We now want to develop a VAE that can model a distribution of images conditioned on a label, i.e. $p(\mathbf{x} \mid y)$ where $\mathbf{x} \in \mathbb{R}^d$ is the image and y is the label, for example, "dog" or "cat".

0 1 2

a) Modify the above pseudocode for the VAE to condition the model on the label y. You can change the dimensions of functions' domains and codomains if necessary.

Only one line needs to be changed:

$$\tilde{\mathbf{x}} = g_\phi(\mathbf{z}, y)$$

Having two decoders is also fine, i.e., one Gaussian prior per class.

0 1 2 3 4

b) After training is completed we want to sample new images from our variational autoencoder. Write the pseudocode to generate an image given a label y. You should use the solution to the previous problem as a starting point.

$$
\begin{aligned}
\tilde{\mathbf{z}} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
\tilde{\mathbf{x}} &= g_\phi(\mathbf{z}, y)
\end{aligned}
$$

Problem 2 Robustness (10 credits)

0
1
2
3
4
5
6
7
8
9
10

We are interested in robustness certification for a model with discrete input data $\mathbf{x} \in \{0, 1, \dots, C\}^N$ and an adversary that changes exactly $\delta \in \mathbb{N}$ elements of $\mathbf{x}$.
The perturbation set can be expressed as

$$\mathcal{P}(\mathbf{x}) = \left\{ \tilde{\mathbf{x}} \in \{0, 1, \dots, C\}^N \middle| ||\mathbf{x} - \tilde{\mathbf{x}}||_0 = \delta \right\} \tag{2.1}$$

with $||\mathbf{x}||_0 = \sum_{n=1}^{N} \mathbb{I}[x_n \neq 0]$.
Specify a set of **linear constraints** on $\tilde{\mathbf{x}}$ to model the perturbation set in Eq. (2.1). You may introduce at most $\mathcal{O}(N)$ constraints and $\mathcal{O}(N)$ variables. You are allowed to use integer-valued variables.
Note: A linear constraint is an equality or inequality between two expressions that are **linear functions** of the variables.

We introduce the integer variables $y_i \in \{0, 1\}$ indicating if the i-th entry is flipped. The constraint set reads

$$\begin{aligned}
y_i &\in \{0, 1\} && \forall i = 1, \dots, N \\
\sum_{i=1}^{N} y_i &= \delta \\
\tilde{x}_i &\leq (1 - y_i)x_i + y_i C && \forall i = 1, \dots, N \\
\tilde{x}_i &\geq (1 - y_i)x_i && \forall i = 1, \dots, N
\end{aligned}$$

Alternative solution:

$$\begin{aligned}
y_i &\in \{0, 1\} && \forall i = 1, \dots, N \\
\sum_{i=1}^{N} y_i &= \delta \\
\tilde{x}_i - x_i &\leq y_i C && \forall i = 1, \dots, N \\
x_i - \tilde{x}_i &\leq y_i C && \forall i = 1, \dots, N
\end{aligned}$$

Problem 3 Autoregressive models (8 credits)

You are given an AR(3) model according to the formula

$$X_t = 17 + 4X_{t-1} + \frac{1}{4}X_{t-2} - X_{t-3} + \varepsilon_t\,,$$

with independently distributed noise variables $\varepsilon_t \sim \mathcal{N}(0, \sigma)$.

0 1 2 3

a) Write down the characteristic polynomial $\Phi(z)$ and show that it can be factorised according to $(2+z)\left(z^2 - \frac{9}{4}z + \frac{1}{2}\right)$.

The characteristic polynomial is given as:

$$\begin{aligned}\Phi(z) &= 1 - \varphi_1 z - \varphi_2 z^2 - \varphi_3 z^3\\ &= 1 - 4z - \frac{1}{4}z^2 + z^3\ [*]\end{aligned}$$

Expanding the given solution for $\Phi(z)$ yields:

$$\begin{aligned}(2+z)\left(z^2 - \frac{9}{4}z + \frac{1}{2}\right) &= \left(2z^2 - \frac{9}{2}z + 1\right) + \left(z^3 - \frac{9}{4}z^2 + \frac{1}{2}z\right)\\ &= 1 + \left(-\frac{9}{2} + \frac{1}{2}\right)z + \left(2 - \frac{9}{4}\right)z^2 + z^3\\ &= 1 - 4z - \frac{1}{4}z^2 + z^3\ [**]\\ \Rightarrow [*] &= [**]\end{aligned}$$

0 1 2 3 4 5

b) Decide if the process X_t is stationary. Justify your answer.

A process X_t is stationary iff the roots of the charactersitic polynomial lie outside the unit ball.

$$\begin{aligned}\Phi(z) &\overset{!}{=} 0\\ \Rightarrow 0 &= (2+z)\left(z^2 - \frac{9}{4}z + \frac{1}{2}\right)\\ \Rightarrow z_1 &= -2 \quad \text{, check quadratic term:}\\ \Rightarrow 0 &= z^2 - \frac{9}{4}z + \frac{1}{2}\\ 0 &= \left(z - \frac{9}{8}\right)^2 - \frac{9^2}{8^2} + \frac{32}{8^2}\\ 0 &= \left(z - \frac{9}{8}\right)^2 - \frac{7^2}{8^2}\\ \frac{7^2}{8^2} &= \left(z - \frac{9}{8}\right)^2\\ \Rightarrow z_{2,3} &= \frac{9}{8} \pm \frac{7}{8} = \left\{2, \frac{1}{4}\right\}\end{aligned}$$

The process is *not stationary* as $z_3 = 1/4$ lies inside the unit ball.

Problem 4 Hidden Markov Models (10 credits)

Consider a hidden Markov model with 2 states $\{1, 2\}$ and 6 possible observations $\{\mathrm{p}, \mathrm{a}, \mathrm{n}, \mathrm{e}, \mathrm{r}, \mathrm{t}\}$. The initial distribution π, transition probabilities $\mathbf{A}$ and emission probabilities $\mathbf{B}$ are

$$\pi = \begin{array}{c} 1 \\ 2 \end{array} \begin{pmatrix} 1/5 \\ 4/5 \end{pmatrix} \qquad \mathbf{A} = \begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 1/5 & 4/5 \\ 2 & 3/5 & 2/5 \end{array} \qquad \mathbf{B} = \begin{array}{c|cccccc} & \mathrm{p} & \mathrm{a} & \mathrm{n} & \mathrm{e} & \mathrm{r} & \mathrm{t} \\ \hline 1 & 0 & 1/5 & 0 & 2/5 & 0 & 2/5 \\ 2 & 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 \end{array},$$

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

a) You have observed the sequence $X = [\mathrm{pattern}]$. Specify all probability distributions $\mathbb{P}()$ that correspond to smoothing / offline inference on X.
Note: You do not need to perform any calculations or insert parameter values.

0 1 2

Offline inference, also called smoothing, applies when data from the *past and future* is used. Hence, we obtain six different solutions:

$$\mathbb{P}(Z_i | X_{1:7} = [\mathrm{pattern}]) \quad \forall\, i = 1, 2, \dots, 6$$

b) Write down the MAP objective given the observed sequence $X = [\mathrm{pattern}]$.

0 1

The MAP objective refers to the computation the most probable sequence of latent variables $Z_{1:7}$ given $X = X_{1:7} = [\mathrm{pattern}]$. Hence, we obtain

$$\arg\max_{Z_{1:7}} \mathbb{P}(Z_{1:7} | X)$$

c) In another instance, you observe the sequence $X = [\mathrm{tea}]$. Given X, what is $\mathbb{P}(Z_3|X)$? [An unnormalised vector suffices]. Justify your answer. What is this type of inference called?

0 1 2 3 4 5 6 7

We have to apply the forward algorithm. First, compute $\alpha_1(k) = \mathbb{P}(Z_1 = k, X_1 = \mathrm{t})$. Note that all probabilities have a factor of $1/5$, therefore, we can ignore it.

$$\begin{aligned} \alpha_1 &= \pi \odot B_{:t} \\ &= \begin{pmatrix} 1 \\ 4 \end{pmatrix} \odot \begin{pmatrix} 2 \\ 0 \end{pmatrix} \\ &= c \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad , c = \text{const.} \end{aligned}$$

For the recursion on α, we have:

$$\alpha_{t+1} = B_{:x_{t+1}} \odot A^{\mathsf{T}} \alpha_t$$

Hence, we obtain:

$$\begin{aligned} \alpha_2 &= B_{:e} \odot A^{\mathsf{T}} \alpha_1 \\ &= \begin{pmatrix} 2 \\ 1 \end{pmatrix} \odot \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 2 \\ 1 \end{pmatrix} \odot \begin{pmatrix} 1 \\ 4 \end{pmatrix} \\ &= c \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad , c = \text{const.} \end{aligned} \qquad \begin{aligned} \alpha_3 &= B_{:a} \odot A^{\mathsf{T}} \alpha_2 \\ &= \begin{pmatrix} 1 \\ 1 \end{pmatrix} \odot \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 1 \end{pmatrix} \odot \begin{pmatrix} 7 \\ 8 \end{pmatrix} \\ &= c \begin{pmatrix} 7 \\ 8 \end{pmatrix} \qquad , c = \text{const.} \end{aligned}$$

This type of inference is called *filtering* or *online inference*.

Problem 5 Graph learning & Variational inference (10 credits)

Consider the following probabilistic model for generating a **directed, weighted** graph with N nodes, continuous adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ and two communities, represented by vector $\mathbf{z} \in \{0, 1\}^N$:

$$p_\lambda(\mathbf{A} \mid \mathbf{z}) = \prod_{n=1}^{N} \prod_{m=1}^{N} p_\lambda(A_{n,m} \mid z_n, z_m) \tag{5.1}$$

$$p_\theta(\mathbf{z}) = \prod_{n=1}^{N} \mathrm{Bern}(z_n \mid \theta) = \prod_{n=1}^{N} \theta^{z_n} \cdot (1-\theta)^{1-z_n} \tag{5.2}$$

with $\theta \in [0, 1]$. The conditional density $p_\lambda(A_{n,m} \mid z_n, z_m)$ will be specified later.
In the following, assume that we have observed a single graph $\mathbf{A} \in \mathbb{R}^{N \times N}$. We want to perform **mean-field variational inference** with variational family

$$q_\phi(\mathbf{z}) = \prod_{n=1}^{N} \mathrm{Bern}(z_n \mid \phi_n) = \prod_{n=1}^{N} \phi_n^{z_n} \cdot (1-\phi_n)^{1-z_n}. \tag{5.3}$$

Note that $\phi \in [0, 1]^N$, i.e. we have one parameter per node.

0 1 2

a) Why is evaluating the ELBO $\mathcal{L}((\lambda, \theta), \phi) = \mathbb{E}_{\mathbf{z} \sim q_\phi}\left[\log p_{\lambda,\theta}(\mathbf{A}, \mathbf{z}) - \log q_\phi(\mathbf{z})\right]$ not tractable for large graphs (e.g. $N > 1000$)?

For any function f, we have $\mathbb{E}_{\mathbf{z} \sim q_\phi}[f(\mathbf{z})] = \sum_{\mathbf{z} \in [0,1]^N} f(\mathbf{z}) q_\phi(\mathbf{z})$.
This means that we have to sum over exponentially many values, which is not tractable.

0 1 2 3 4

b) Assume that we approximate the ELBO with a single Monte Carlo sample $\mathbf{z} \in \{0, 1\}^N$, i.e.

$$\mathcal{L}((\lambda, \theta), \phi) \approx \log p_{\lambda,\theta}(\mathbf{A}, \mathbf{z}) - \log q_\phi(\mathbf{z}). \tag{5.4}$$

Let

$$p_\lambda(A_{n,m} \mid z_n, z_m) = \begin{cases} \lambda_1 \exp(-\lambda_1 A_{n,m}) & \text{if } A_{n,m} \geq 0 \wedge z_n = z_m, \\ \lambda_2 \exp(-\lambda_2 A_{n,m}) & \text{if } A_{n,m} \geq 0 \wedge z_n \neq z_m, \\ 0 & \text{else.} \end{cases}$$

with $\lambda_1, \lambda_2 > 0$. Assume that λ_2, θ and ϕ are fixed.
Prove that the optimal value of λ_1, i.e. the value that maximizes $\log p_{\lambda,\theta}(\mathbf{A}, \mathbf{z}) - \log q_\phi(\mathbf{z})$ is

$$\lambda_1^* = \frac{|\{n, m \mid z_n = z_m\}|}{\sum_{n,m \mid z_n = z_m} A_{n,m}}.$$

Note: You may also write on the next page.

By definition, we have

$$\log p_{\lambda,\theta}(\mathbf{A},\mathbf{z}) - \log q_\phi(\mathbf{z}) = \sum_{n=1}^{N}\sum_{m=1}^{N} \log p_\lambda(A_{n,m} \mid z_n, z_m) + c,$$

where c are terms that are constant in λ.
By definition of $p_\lambda(A_{n,m} \mid z_n, z_m)$, all terms for which $z_n \neq z_m$ are also constant in λ_1, meaning

$$\log p_{\lambda,\theta}(\mathbf{A},\mathbf{z}) - \log q_\phi(\mathbf{z}) = \sum_{n,m \mid z_n = z_m} \log p_\lambda(A_{n,m} \mid z_n, z_m) + c'.$$

We can find the optimal λ_1 by setting the derivative to zero:

$$\begin{aligned}
&\frac{\partial}{\partial \lambda_1} \sum_{n,m \mid z_n = z_m} \log p_\lambda(A_{n,m} \mid z_n, z_m) \overset{!}{=} 0 \\
\iff &\frac{\partial}{\partial \lambda_1} \sum_{n,m \mid z_n = z_m} \log \lambda_1 - \lambda_1 A_{n,m} = 0 \\
\iff &\sum_{n,m \mid z_n = z_m} \frac{1}{\lambda_1} - A_{n,m} = 0 \\
\iff &\frac{1}{\lambda_1} \left|\{n, m \mid z_n = z_m\}\right| = \sum_{n,m \mid z_n = z_m} A_{n,m} \\
\iff &\lambda_1^* = \frac{\left|\{n, m \mid z_n = z_m\}\right|}{\sum_{n,m \mid z_n = z_m} A_{n,m}}
\end{aligned}$$

c) To allow optimization w.r.t. ϕ, we want to apply the reparameterization trick. Specify a base distribution $b(\epsilon)$ and a transformation $T(\epsilon, \phi)$ such that

0 1 2 3 4

$$\mathbf{E}_{\mathbf{z} \sim q_\phi}\left[\log p_{\lambda,\theta}(\mathbf{A},\mathbf{z}) - \log q_\phi(\mathbf{z})\right] = \mathbf{E}_{\epsilon \sim b}\left[\log p_{\lambda,\theta}(\mathbf{A}, T(\epsilon,\phi)) - \log q_\phi(T(\epsilon,\phi))\right]. \tag{5.5}$$

We can choose N arbitrary, independent distributions $b_1, \dots, b_N$ over $\mathbb{R}$ and define $b(\epsilon) = \prod_{n=1}^{N} b_n(\epsilon_n)$.
Let F_n be the cumulative distribution function of b_n. We can then define $T : \mathbb{R}^N \times [0,1] \to [0,1]$ via.

$$T(\epsilon, \phi) = \begin{bmatrix} \mathbb{I}\left[F_1(\epsilon_1) \le \phi_1\right] \\ \mathbb{I}\left[F_2(\epsilon_2) \le \phi_2\right] \\ \vdots \\ \mathbb{I}\left[F_N(\epsilon_N) \le \phi_N\right] \end{bmatrix}$$

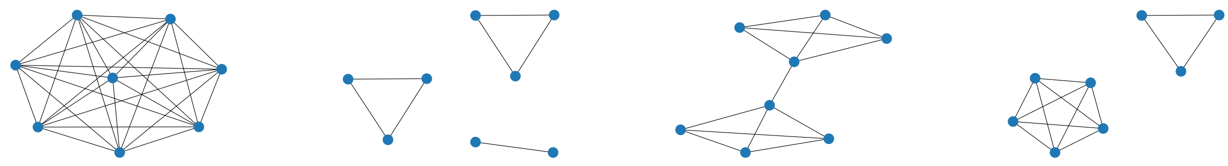
One simple example is choosing $b_n = \mathrm{Uniform}(0,1)$ and

$$T(\epsilon, \phi) = \begin{bmatrix} \mathbb{I}\left[\epsilon_1 \le \phi_1\right] \\ \mathbb{I}\left[\epsilon_2 \le \phi_2\right] \\ \vdots \\ \mathbb{I}\left[\epsilon_N \le \phi_N\right] \end{bmatrix}$$

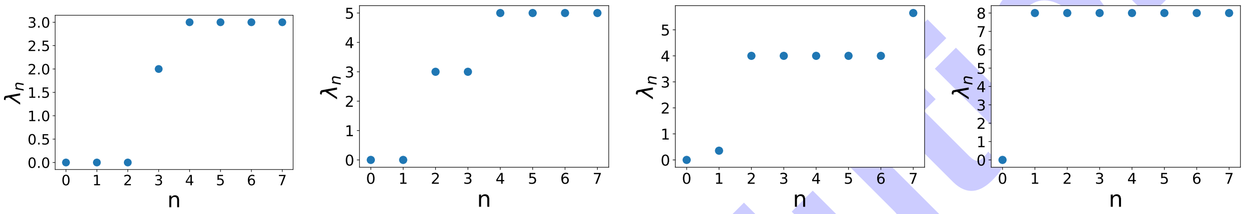
Problem 6 Graphs – Laws & patterns (8 credits)

0 1 2 3 4 5 6 7 8

You are given four graphs (a-d), each consisting of eight nodes. You are further given four eigenspectra (1-4), i.e. eigenvalues of the graph Laplacian ordered in ascending order. Assign each of the graphs (a-d) to an eigenspectrum (1-4). Justify your answer.



(a) (b) (c) (d)



(1) (2) (3) (4)

(b) - (1), because the graph has three components and three eigenvalues are 0.
(d) - (2), because the graph has two components and two eigenvalues are 0.
Both *a* and *c* are connected graphs, meaning we need to find another criterion to distinguish them.
(c) - (3) because we observe an eigengap at eigenvalue λ_2 .
(a) - (4) by the exclusion principle .

Problem 7 Page Rank (8 credits)

The PageRank scores (without teleports) of the graphs a-d have been computed with power iteration. Match the graphs a-d with the results 1-4. Justify your answer.

0 1 2 3 4 5 6 7 8

1. Does not converge.
2. Does not converge.
3. Converges to $r_A = 0.167$, $r_B = 0.167$, $r_C = 0.167$, $r_D = 0.5$.
4. Converges to $r_A = 0.125$, $r_B = 0.375$, $r_C = 0.25$, $r_D = 0.25$.

Match options as:
Graph a and Option 3. Nodes a, b, and c are symmetric and should have the same PageRank score. Node a should have 3 times the PageRank score than the other nodes.
Graph b and Option 1/2. The graph is periodic and thus, the calculation of the PageRank score of the graph via Power Iteration will not converge.
Graph c and Option 4. Nodes c and d are symmetric and should have the same PageRank score, Node b should have the highest PageRank score.
Graph d and Option 1/2. The graph is reducible and thus, the calculation of the PageRank score of the graph via Power Iteration will not converge.

Problem 8 Graph Neural Networks (6 credits)

Below, you can find three different types of Graph Neural Network modules. The node embedding $h_u^{(t+1)}$ of node u at layer $t+1$ is calculated with:

- Network Propagation (NP): $h_u^{(t+1)} = \sum_{v \in N(u) \cup \{u\}} h_v^{(t)}$
- Graph Convolution (GCN): $h_u^{(t+1)} = \phi_{gcn}(h_u^{(t)}, \oplus_{v \in N(u)} \psi_{gcn}(h_v^{(t)}))$
- Message Passing (MP): $h_u^{(t+1)} = \phi_{mp}(h_u^{(t)}, \oplus_{v \in N(u)} \psi_{mp}(h_v^{(t)}, h_u^{(t)}))$

where $\oplus$ is some permutation invariant function without learnable parameters, the functions ψ_{gcn}, ψ_{mp} transform hidden features, functions ϕ_{gcn}, ϕ_{mp} are update functions and $N(u)$ is the neighbourhood of node u.

0 ☐
1 ☐
2 ☐
3 ☐

a) Prove that network propagation is a special case of graph convolution.
Hint: You can do this by providing specific realizations of $\oplus$, ψ_{gcn} and ϕ_{gcn}.

$$\oplus_{v \in N(u)} = \sum_{v \in N(u)} \tag{8.1}$$

$$\psi_{gcn}(h_v^{(t)}) = h_v^{(t)} \tag{8.2}$$

$$\phi_{gcn}(h_u^{(t)}, a) = h_u^{(t)} + a \tag{8.3}$$

0 ☐
1 ☐
2 ☐
3 ☐

b) Prove that graph convolution is a special case of message passing.
Hint: You can do this by providing specific realizations of ψ_{mp} and ϕ_{mp}.

$$\phi_{mp} = \phi_{gcn} \tag{8.4}$$

$$\psi_{mp}(h_v^{(t)}, h_u^{(t)}) = \psi_{gcn}(h_v^{(t)}) \tag{8.5}$$

Problem 9 Limitations of Graph Neural Networks (6 credits)

a) Briefly explain two challenges when attacking GNNs using adversarial attacks.

- Optimization over discrete variables (the graph structure). Perturbations are measured via non-convex L_0 norm.
- Relational dependencies between nodes: cannot view samples (nodes) in isolation
- $(\mathbf{A}', \mathbf{X}') \approx (\mathbf{A}, \mathbf{X})$: What is a sensible measure of perturbations that do not change the semantics for (attributed) graphs?
- Transductive setting: unlabeled data is used during training; most realistic scenario is a poisoning attack, wehre the attacker modifies the training data, which corresponds to a challenging bilevel optimization problem

b) We model the absence or presence of an edge in a graph with N nodes using a binary vector $\mathbf{x} \in \{0, 1\}^{N^2}$. Now, we want to use randomized smoothing to certify that a smoothed classifer using GNNs as base-classifers is robust against attacks on the graph structure.

Recall that a smoothed classifier $g(\mathbf{x})_c$ returns the probability that the base classifier f classifies a smoothed sample $\tilde{\mathbf{x}} \sim \phi(\mathbf{x})$ as class c, i.e. $g(\mathbf{x})_c := \mathbb{P}(f(\phi(\mathbf{x})) = c)$ with a randomization scheme $\phi(\mathbf{x})$.

What is the problem when we want to use Gaussian noise as our randomization scheme? How could that problem be solved?

The problem with using Gaussian noise as $\phi(\mathbf{x})$ is that we generate perturbed $\tilde{\mathbf{x}}$ in $\mathbb{R}$. As a solution, one can use Bernoulli random variables to model whether a particular edge should be flipped or not.

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