Lecture 5: Optimal Architectures

Kimon Fountoulakis

UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

Outline

We will define a notion of local classifier

We will define a notion of optimal Bayes local classifier

We will compute the optimal classifier analytically

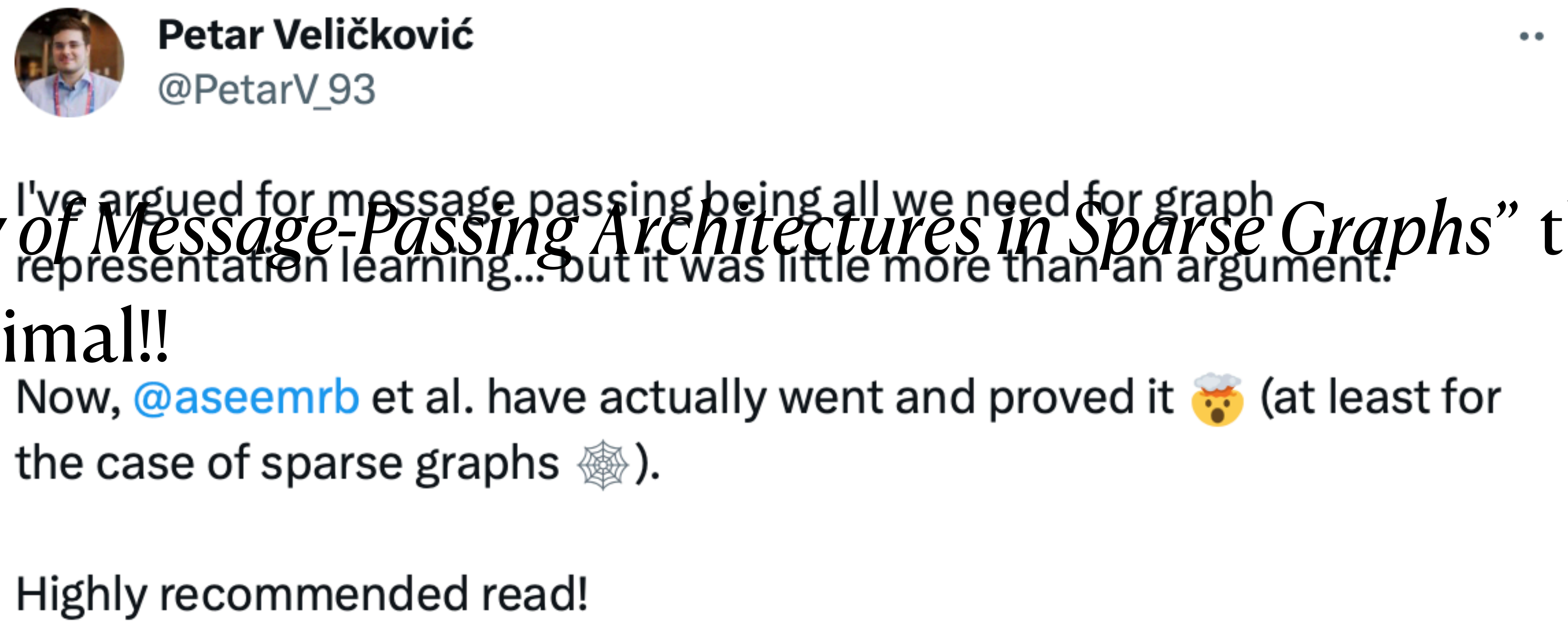
We will analyze its performance

Disclaimer

I have simplified the results.

The story behind the paper "Optimal Architectures"

- I taught CS 886 in 2023: Graph Neural Networks. We studied 37 papers!
- Half way through the course students felt bored. Why?
- All graph neural networks applied the same architecture!!! A message-passing architecture...
- What is happening?
- We prove in "*Optimality of Message-Passing Architectures in Sparse Graphs*" that message-passing is optimal!!



Problem setting

- Multi-class, number of classes $C \geq 2$.
- ***No assumption on the features. They can follow any distribution.***
- ***The graph does not have to be "dense enough".***

Assumptions

Gone

- Binary classification assumption

Gone

- Graph density assumption $p, q = \Omega\left(\dfrac{\log^2 n}{n}\right)$

Gone

- Sufficient gap between p, q.

Assumptions

Gone

- Features have ~~to be~~ Gaussian.

The data model: class membership

- The class membership of the nodes, denoted by y_u for node u, is uniform.

$$y_u \sim \mathrm{Uniform}(\{1, \ldots, C\})$$

- where C is the number of classes.

- So... balanced classes.

The data model: random graph

$$\Pr(\exists \text{ an edge between nodes } u \text{ and } v \mid y_u = i, y_v = j) = p_{ij}\,.$$

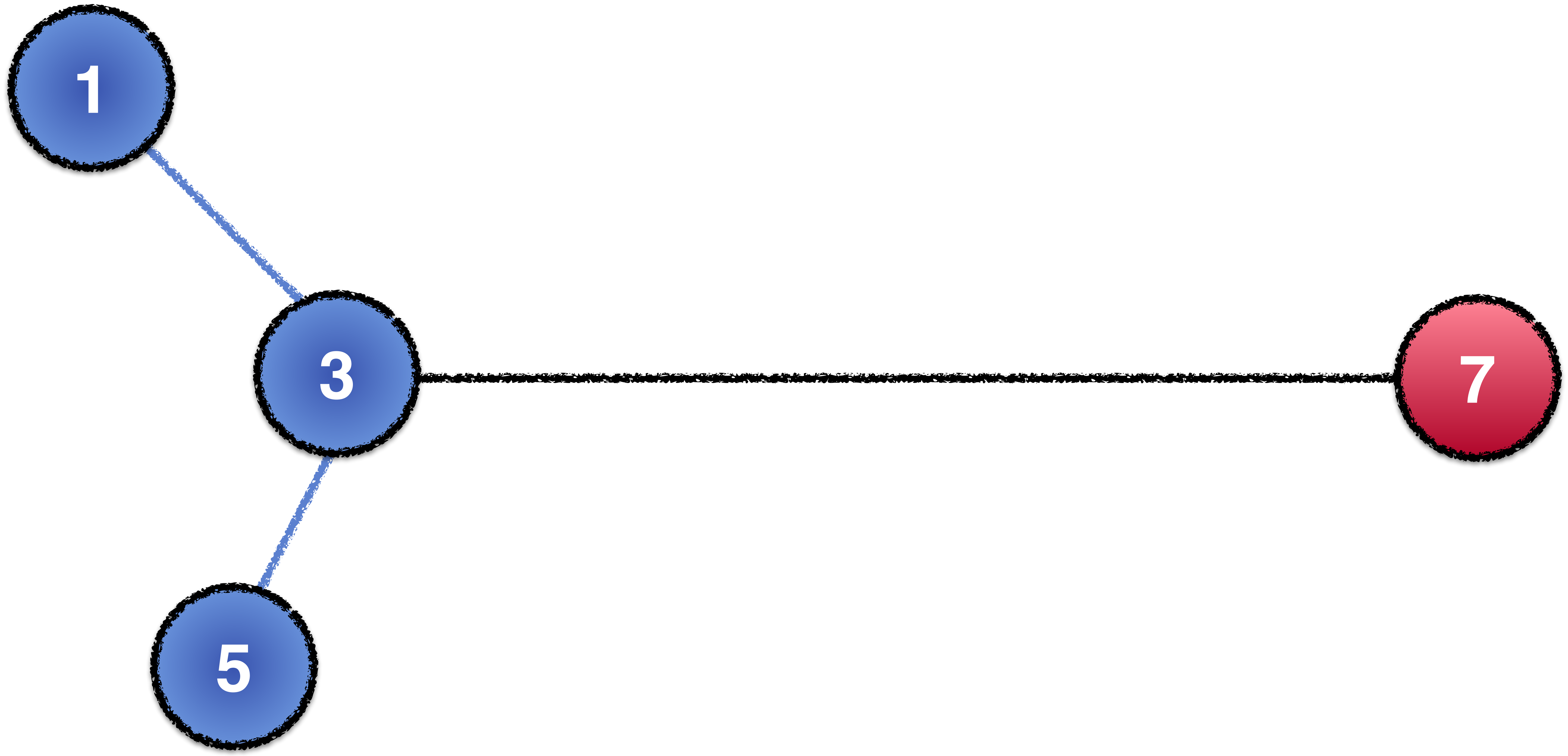
- where $p_{ij} \in [0,1]$.

The data model: features

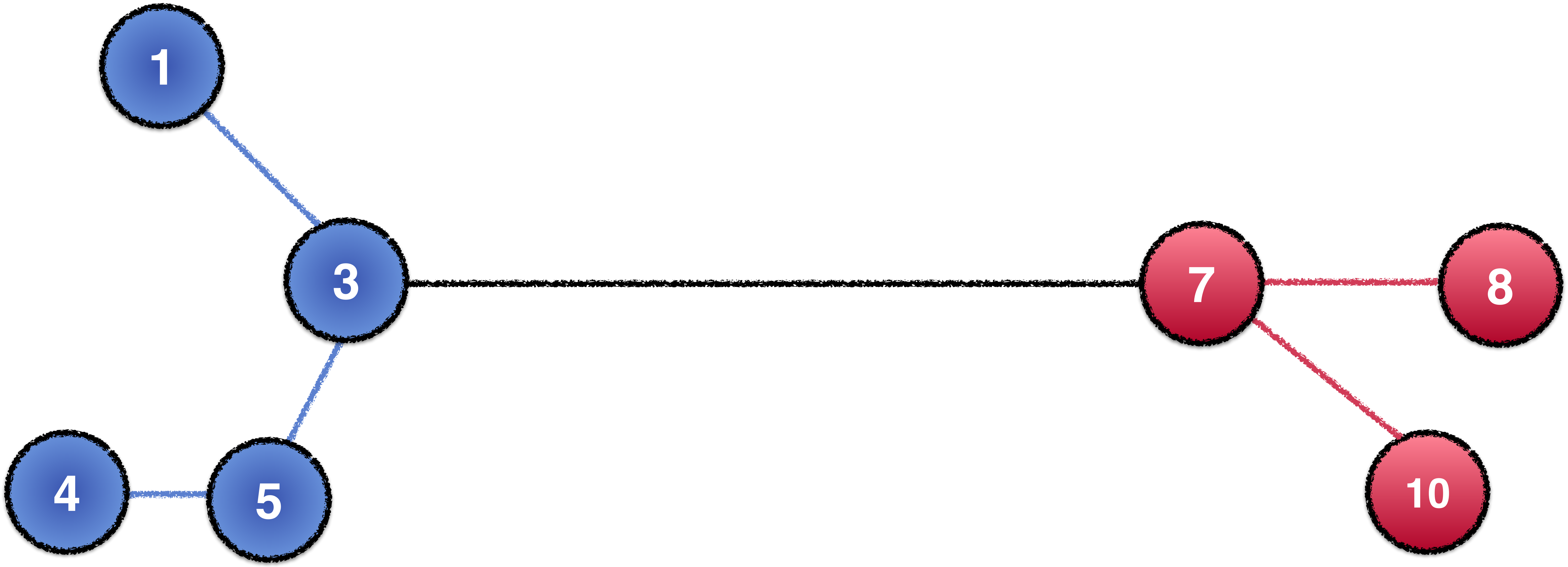
- They can have any distribution, discrete or continuous.

Local Classifiers

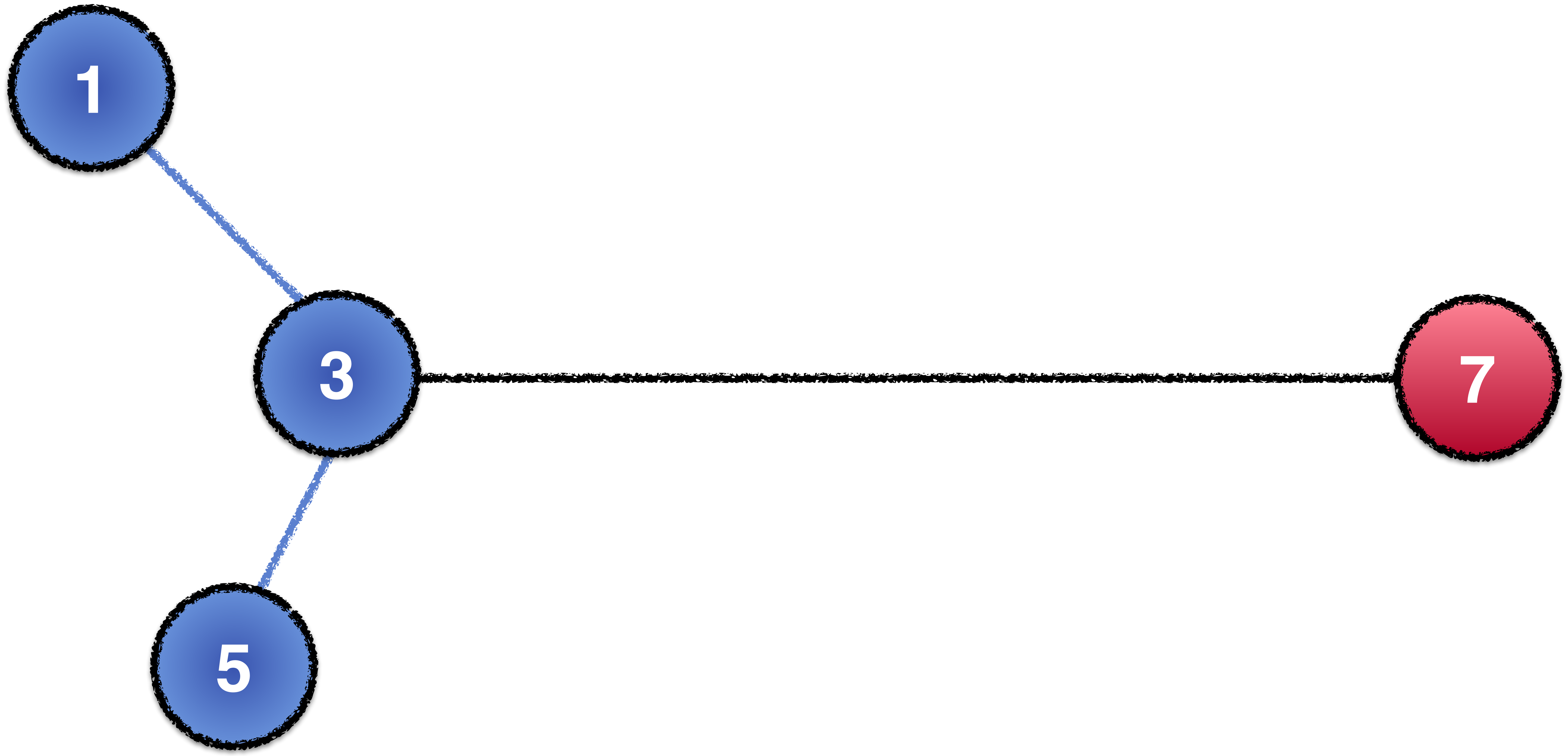
hop distance: nodes 1, 5 and 7 are 1 hop away from node 3.



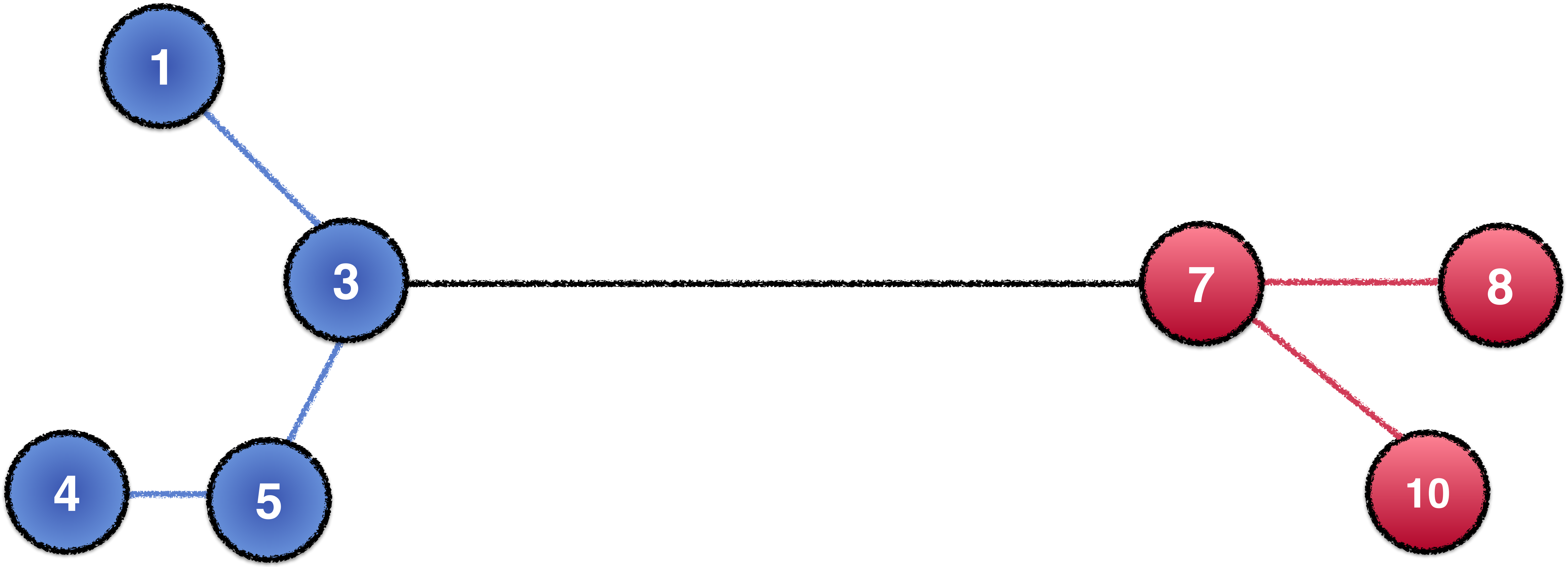
hop distance: nodes 4, 8 and 10 are 2 hops away from node 3.



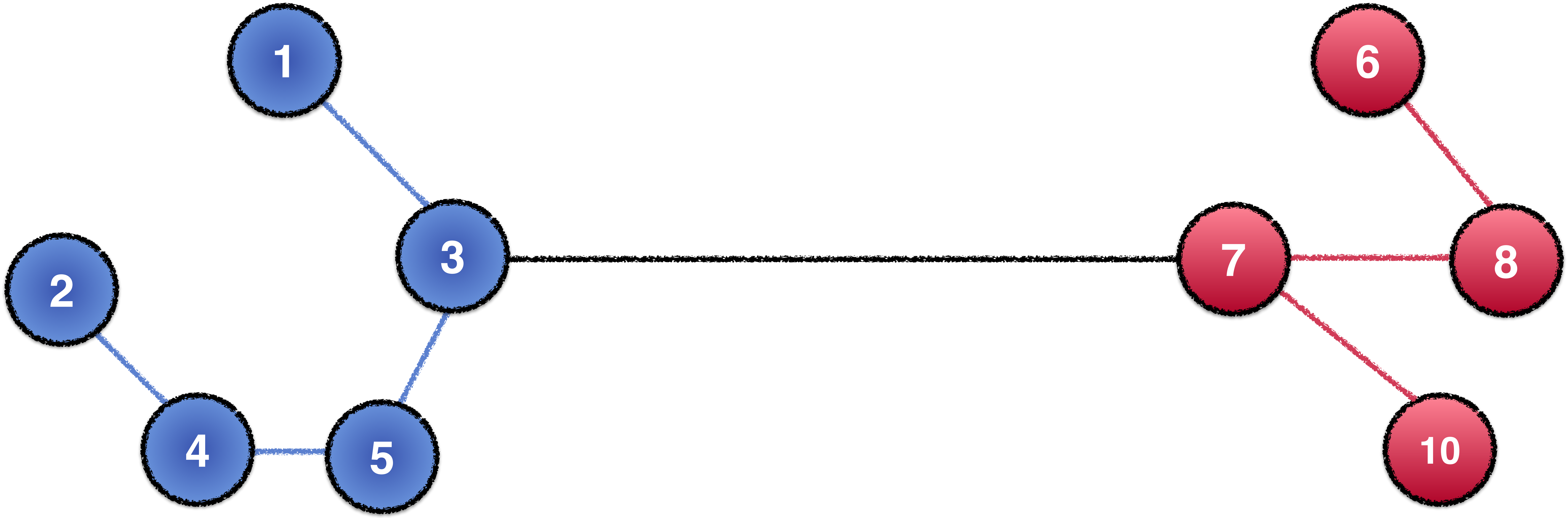
1-local classifier: classifies node 3 using data from node 3 and **1-hop** neighbours.



2-local classifier: classifies node 3 using data from node 3 and **2-hop** neighbours.



3-local classifier: classifies node 3 using data from node 3 and **3-hop** neighbours.



$\ell-$local classifiers

- ℓ is a hyper-parameter.

- An $\ell-$local classifier classifies a node u using all data, features + edges, within a neighbourhood of distance at most ℓ.

- distance = hop distance in the graph.

$\ell-$local optimal Bayes Classifier

The classifier that minimizes the probability of misclassifying a node among all $\ell-$local classifiers.

The Optimal Classifier for Our Data Model

The optimal architecture: general idea

- We will make a decision about some node u by considering data from ***all*** *nodes up to distance* ℓ.

- where ℓ is a hyper-parameter.

The optimal architecture: MLP component

- We are given the feature matrix X.

- Apply L layers of an MLP for X:

$$H^{(l)} = \sigma_l(H^{(l-1)}W^{(l)} + b^{(l)}) \text{ for } l \in [L]$$

$$H^{(0)} = X$$

- $H_{ui}^{(L)}$ represents the score of node u to be in class i.

The optimal architecture: learnable scores among classes

- Define additional ***learnable*** parameters:

$$Z_{ij} \in \mathbb{R}$$

- Z_{ij} represents a score for a node in class i to be connected to a node in class j.

The optimal architecture: learnable scores among classes

- Define additional ***learnable*** parameters:

$$Z \in R^{C \times C}$$

- where C is the number of classes (hyper-parameter, ideally should match ground-truth)

The optimal architecture: probability of an edge between classes

- Normalize Z

$$Q = \text{sigmoid}(Z)$$

- Q_{ij} represents the ***learned*** probability that a node in class i to be connected to a node in class j.

The optimal architecture: probability of paths between classes

- Raise to power k to get $Q^k = \underbrace{Q \cdots Q}_{k\ times}$

- Q_{ij}^k represents the learned probability that we get a distance k path between a node in class i and a node in class j.

The optimal architecture: messages

- For every triplet of (node v, class i, distance k) we collect the highest score ("message") among all classes j.

$$M_{v,i}^{(k)} = \max_{j \in [C]} H_{v,j}^{(L)} + \log(Q_{i,j}^{k}) \qquad \text{for } k \in [\ell], v \in [n], i \in [C]$$

The optimal architecture: the graph

- Define $\tilde{A}^{(k)}$ an $n \times n$ node matrix.
- $\tilde{A}^{(k)}_{uv} = 1$ if and only if node v is present in the distance k neighbourhood of node u but not within the distance $k-1$ neighbourhood.

The optimal architecture: messages

- Sum of messages $M_{v,i}^{(k)}$ of all distance k neighbours of node u.

$$\sum_{v \in [n]} \tilde{A}_{u,v}^{(k)} M_{v,i}^{(k)} = \sum_{v:\ k\text{-}hop\ neighbors\ of\ u} M_{v,i}^{(k)}$$

The optimal architecture: messages

- Sum of messages $M_{v,i}^{(k)}$ of neighbours of node u up distance ℓ.

$$\sum_{k=1}^{\ell} \sum_{v \in [n]} \tilde{A}_{u,v}^{(k)} M_{v,i}^{(k)} = \sum_{v:\ 1\text{-}hop\ neighbors\ of\ u} M_{v,i}^{(1)} + \sum_{v:\ 2\text{-}hop} M_{v,i}^{(2)} + \ldots + \sum_{v:\ \ell\text{-}hop} M_{v,i}^{(\ell)}$$

The optimal architecture: the optimal classifier

- The class of node u is computed using:

$$y_u = \mathrm{argmax}_{i\in[C]} \quad \boxed{H^{(L)}_{u,i}} + \boxed{\sum_{k=1}^{\ell} \sum_{v\in[n]} \tilde{A}^{(k)}_{u,v} M^{(k)}_{v,i}}$$

Score of node u for class i

sum of messages of neighbours of node u up to distance ℓ

The optimal architecture: the optimal classifier

- The classifier below

$$y_u = \mathrm{argmax}_{i\in[C]} \quad H_{u,i}^{(L)} + \sum_{k=1}^{\ell} \sum_{v\in[n]} \tilde{A}_{u,v}^{(k)} M_{v,i}^{(k)}$$

- is the Bayes optimal classifier among all $\ell-$local classifiers.

How to set the hyper-parameter ℓ

- Let's ignore computational complexity for now.
- Then, $\ell =$ diameter of the graph.

Def.: diameter is the maximum shortest path between two nodes.

- We need to make sure that each node utilizes information from the whole graph!

How to set the hyper-parameter ℓ

- What if we consider computational complexity?

I don't know!

- Project suggestion: ***how well does the optimal classifier perform in practice?***

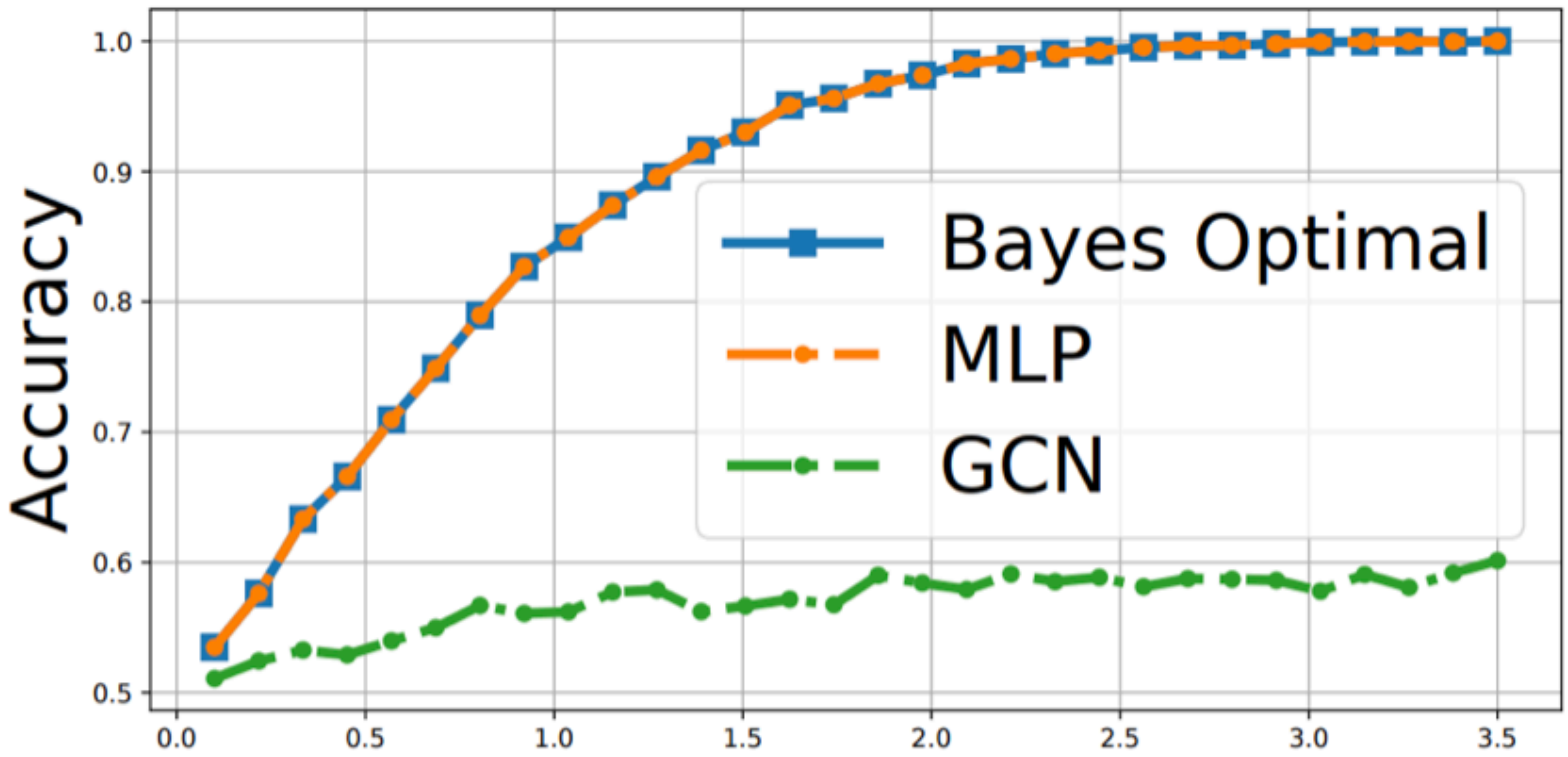
The optimal architecture: uninformative graph

- Let's assume that the graph has no information, i.e., edges are drawn with the same probability regardless of the class of the nodes.

- The optimal classifier completely ignores the graph.

$$y_u = \operatorname{argmax}_{i \in [C]} \quad H^{(L)}_{u,i}$$

Uninformative graph, graph signal-to-noise ratio = zero



$$\text{SNR of Gaussian Mixture} = \frac{\text{distance between means}}{\text{standard deviation}}$$

The optimal architecture: perfect graph

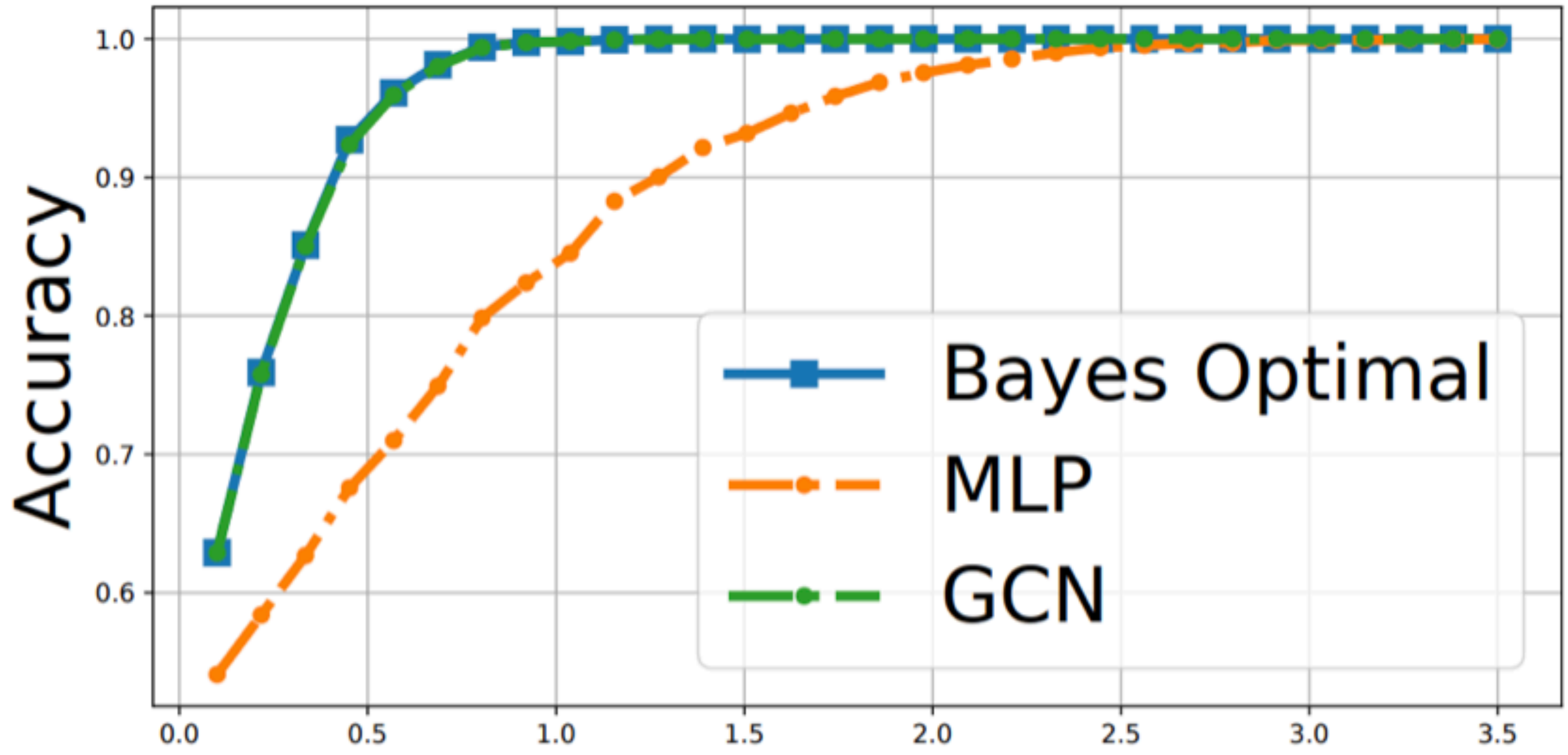
- Let's assume that the graph has ***no*** noisy edges (e.g., in previous lectures these where inter-edges if $p > q$).

- The optimal classifier reduces to vanilla convolution within each class:

$$y_u = \text{argmax}_{i \in [C]} \sum_{v \in \mathcal{N}(u)} H^{(L)}_{v,i}$$

- where $\mathcal{N}(u)$ are all neighbours of u of distance at most ℓ.

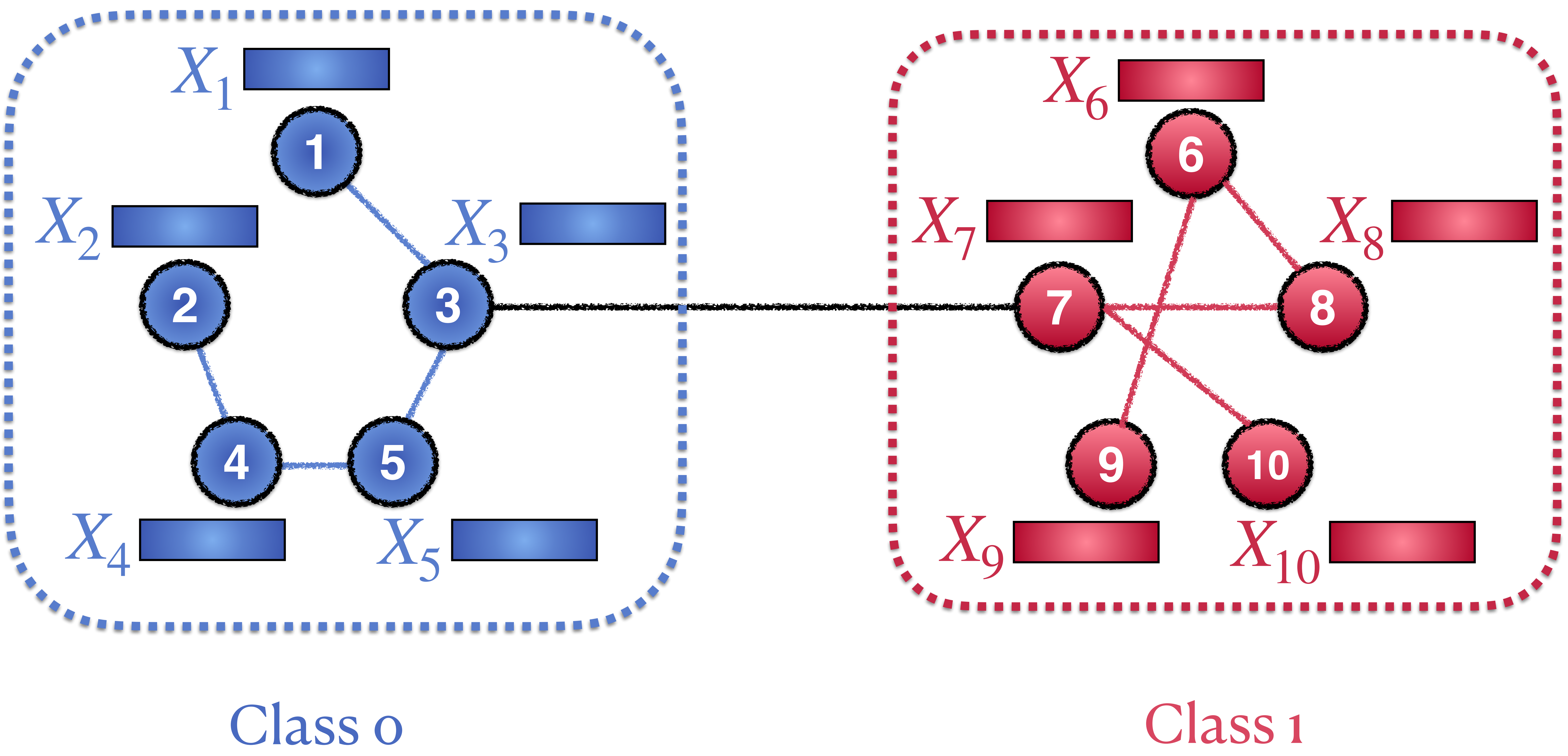
Perfect graph, graph signal-to-noise ratio = one



$$\text{SNR of Gaussian Mixture} = \frac{\text{distance between means}}{\text{standard deviation}}$$

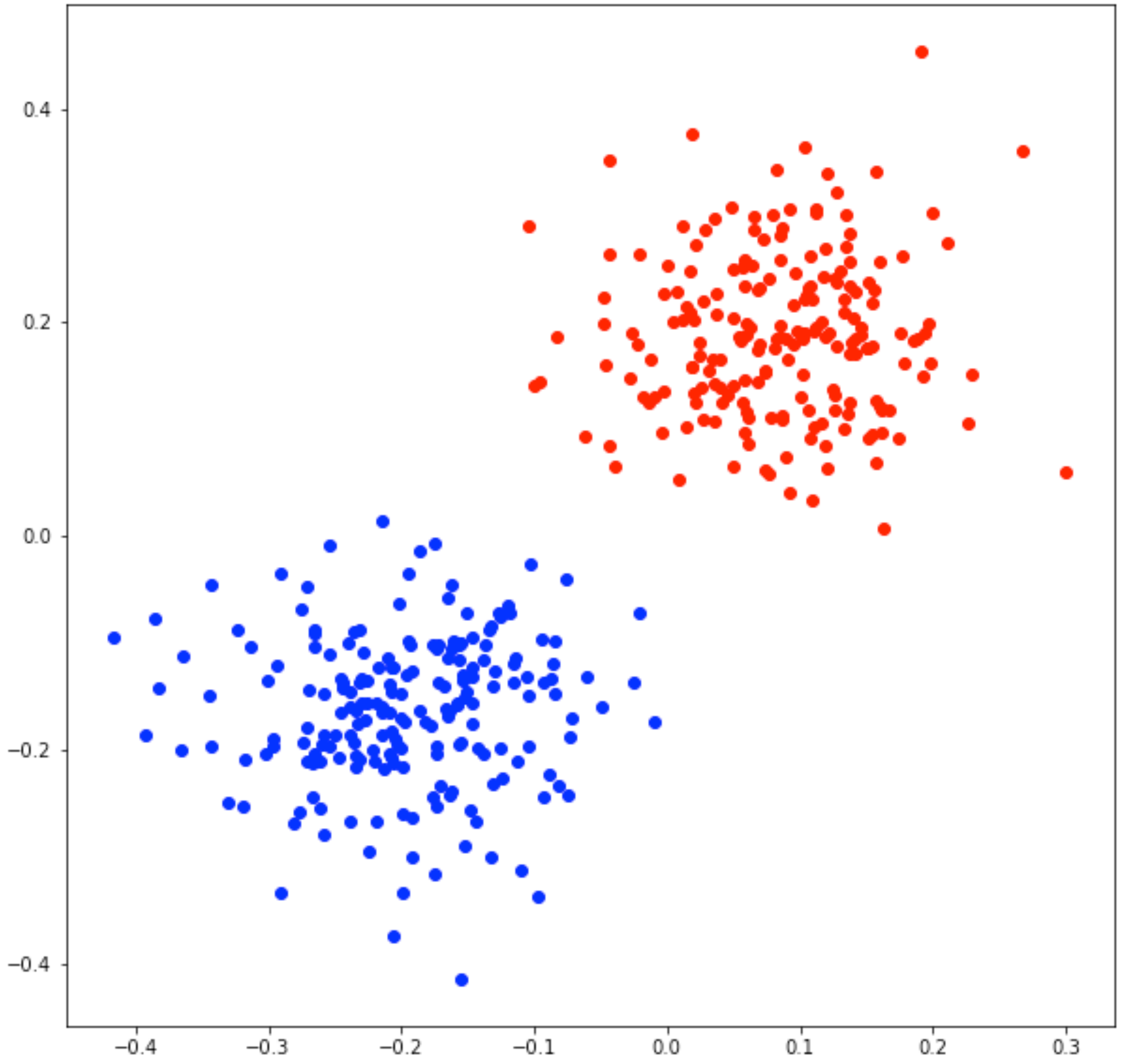
Reduction to Binary Classification and the Contextual Stochastic Block Model

Node classification



Assumptions: features

$$X_i \sim \mathcal{N}(\mu, \sigma^2 I) \text{ if } i \in C_0$$
$$X_i \sim \mathcal{N}(\nu, \sigma^2 I) \text{ if } i \in C_1$$



Assumptions: graph

Stochastic Block Model

$$\mathbb{P}(\text{edge } (i,j)) = \begin{cases} p & \text{if } i,j \text{ are in the same class} \\ q & \text{if } i,j \text{ are in the different class} \end{cases}$$

The Bayes optimal classifier for our data model

Intuitive interpretation of the $2-$local optimal Bayes Classifier

$\psi(X_5)$

5

2-hop neighbour

$\psi(X_1)$

$+$

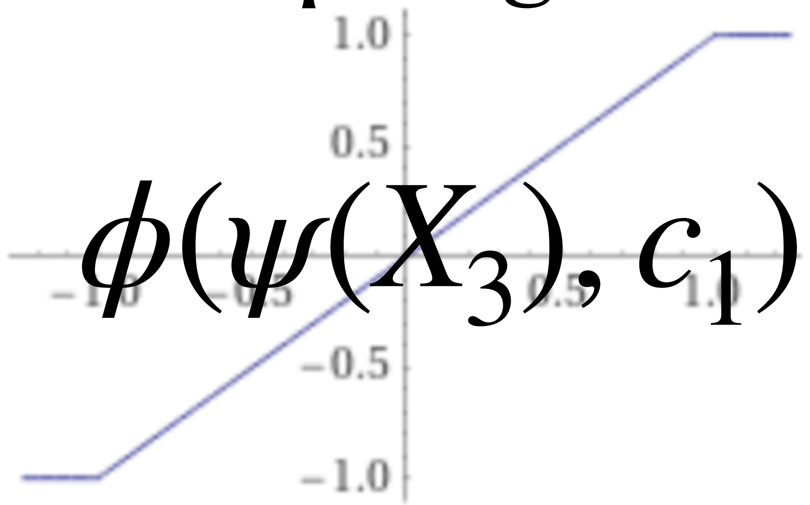
$+$

$+$

1

Clipping function ϕ

For 1-hop neighbours



$\phi(\psi(X_3), c_1)$

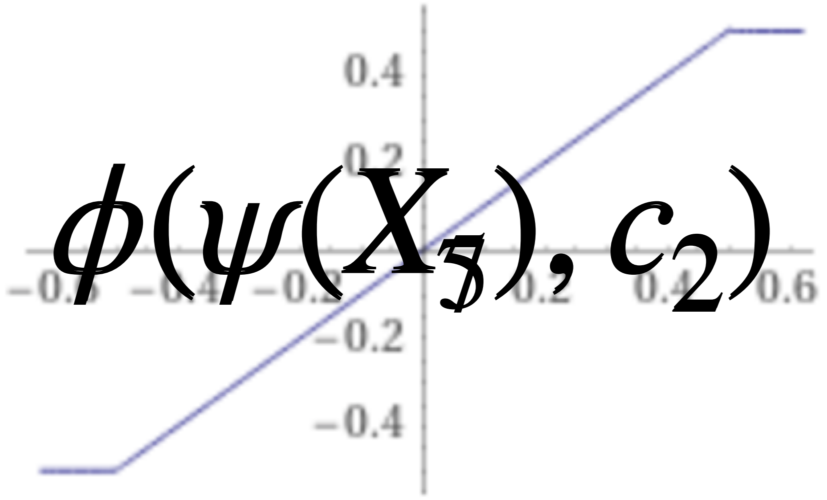
$\psi(X_3)$

3

1-hop neighbour

Clipping function ϕ

For 2-hop neighbours



$\phi(\psi(X_7), c_2)$

$$c_1 = \log \frac{1+\Gamma}{1-\Gamma}$$

$$\text{where } \Gamma = \frac{|p-q|}{p+q}$$

$$c_2 = \log \frac{1+\Gamma^2}{1-\Gamma^2}$$

$\psi(X_7)$

7

2-hop neighbour

$\psi(\mathbf{x}) = \log \dfrac{\rho_+(\mathbf{x})}{\rho_-(\mathbf{x})}$ is the log-likelihood ratio for the two classes.

$\ell-$local optimal Bayes Classifier

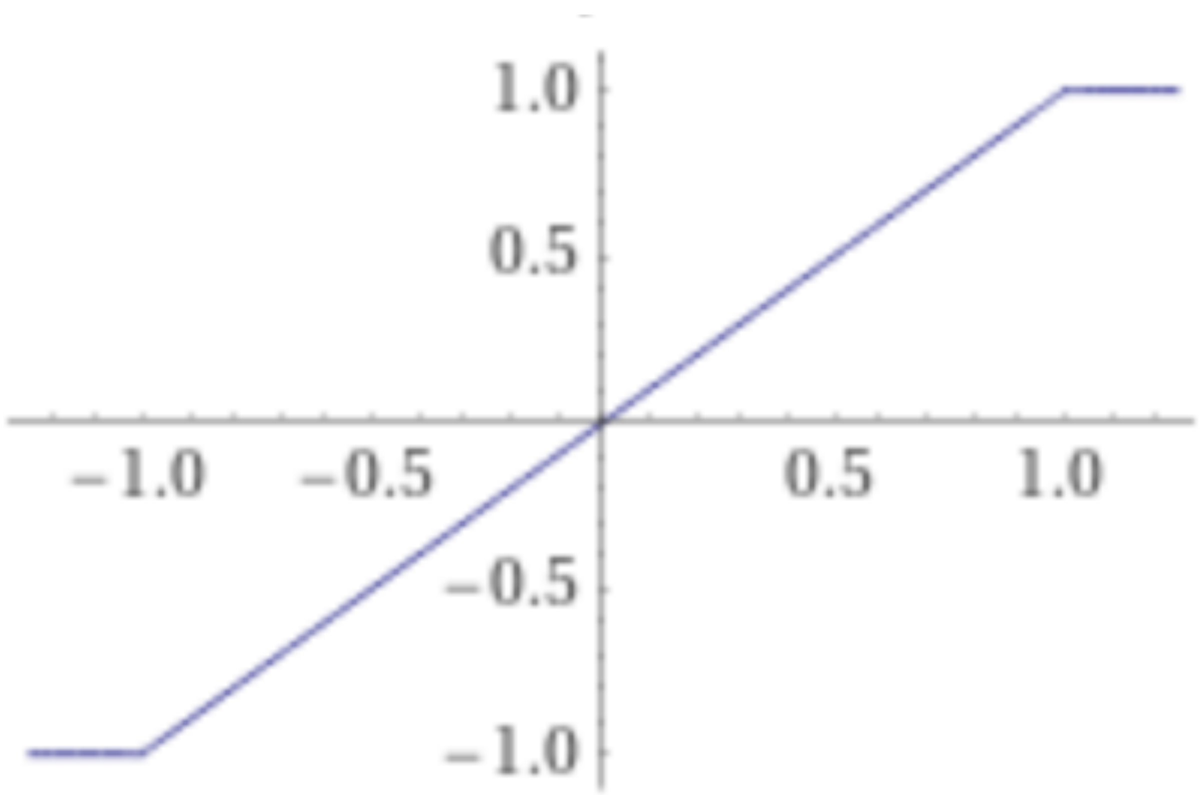
$$h_\ell^*(\mathbf{x}_v) = \text{sign}\left(\psi(\mathbf{x}_v) + \sum_{u:\ 1\text{-}hop\ neighbors\ of\ v} \mathcal{M}_1(\mathbf{x}_u) + \sum_{u:\ 2\text{-}hop} \mathcal{M}_2(\mathbf{x}_u) + \ldots + \sum_{u:\ \ell\text{-}hop} \mathcal{M}_\ell(\mathbf{x}_u)\right)$$

$$\psi(\mathbf{x}) = \log \frac{\rho_+(\mathbf{x})}{\rho_-(\mathbf{x})}$$

is the log-likelihood ratio
for the two classes.

$$\mathcal{M}_\ell(\mathbf{x}) = \phi(\psi(\mathbf{x}), c_\ell)$$

$$\phi(x) = min(max(x, -c), c)$$



$$c_\ell = \log \frac{1+\Gamma^\ell}{1-\Gamma^\ell} \text{ where } \Gamma = \frac{|p-q|}{p+q}$$

Performance Analysis

Assumptions: graph

- We will only assume that the graph is ***sparse.***

- Formally, this means that the expected number of neighbours per node is $O(1)$.

Gone

- ~~Degree concentration.~~

Assumptions: graph

Stochastic Block Model

$$\mathbb{P}(\text{edge } (i,j)) = \begin{cases} p & \text{if } i,j \text{ are in the same class} \\ q & \text{if } i,j \text{ are in the different class} \end{cases}$$

- $p, q = \mathcal{O}(1/n),\ n$ is the number of nodes in the graph.
- $\mathbb{E}(\text{number of neighbours per node}) = O(1)$

Assumptions: graph

Stochastic Block Model

$$\mathbb{P}(\text{edge } (i,j)) = \begin{cases} p & \text{if } i,j \text{ are in the same class} \\ q & \text{if } i,j \text{ are in the different class} \end{cases}$$

- $p, q = \mathcal{O}(1/n)$, n is the number of nodes in the graph.
- The graph is ***tree-like***

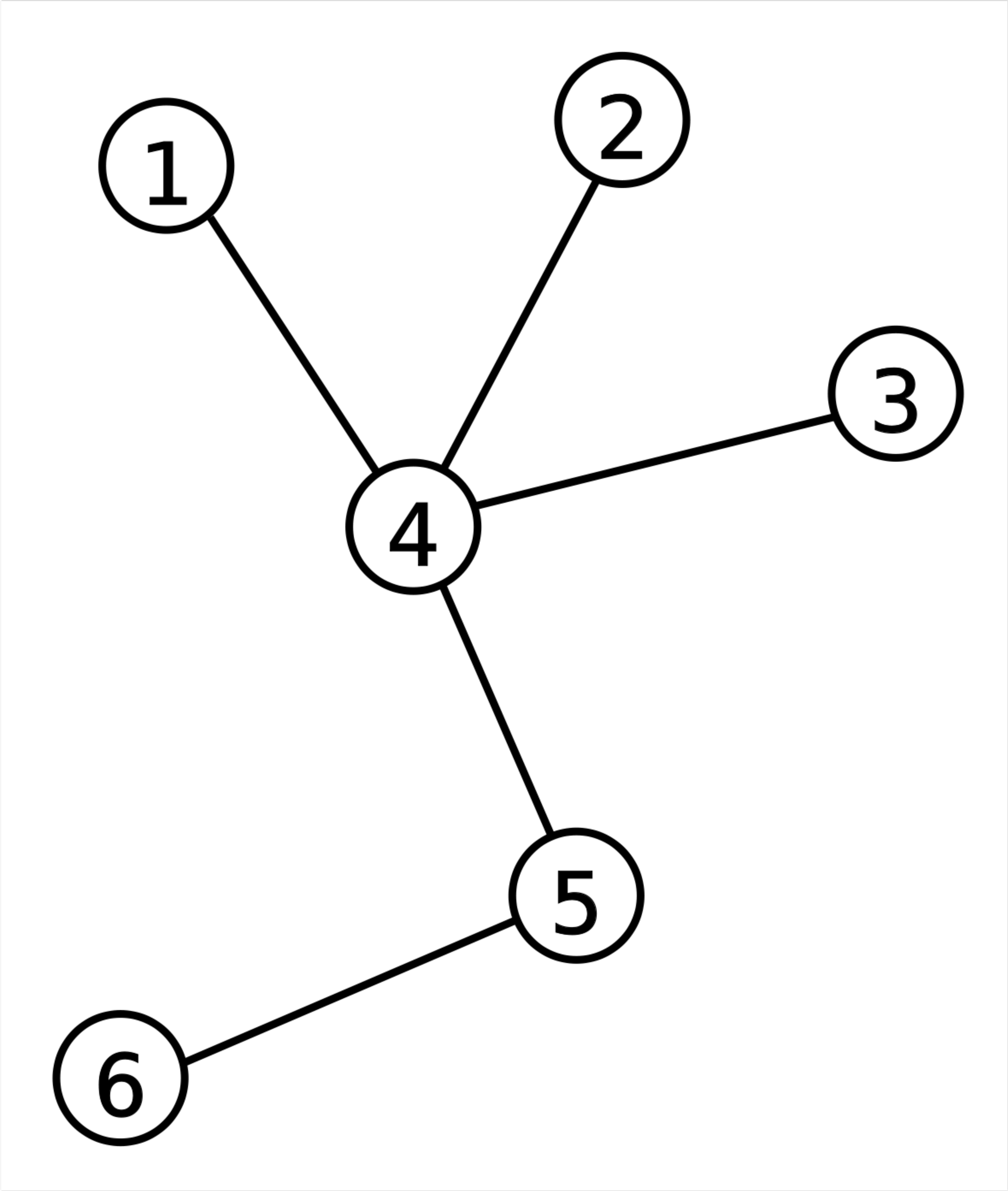
Let's focus on
this implication

Tree graphs

- Definition: the graph is connected and has no cycles.

Wikipedia

Example



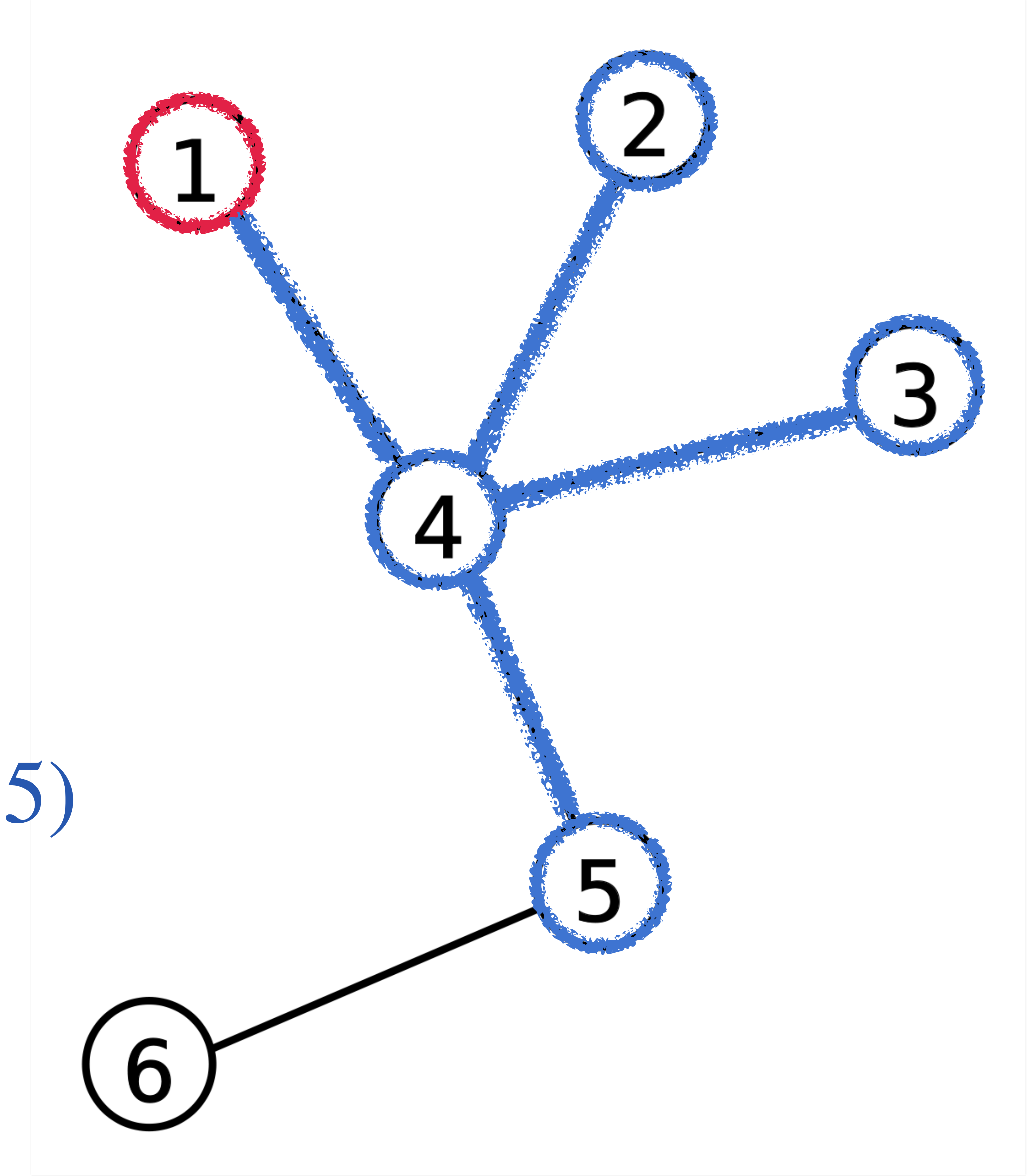
Local ℓ-sub-graph around node u

- Pick a node u from a given graph G.
- Pick a parameter $\ell > 0$.
- The local ℓ-sub-graph around u is defined as all the neighbours of node u up to distance k in the graph G and all edges among those nodes.

Local ℓ-sub-graph: example

- Pick a node 1

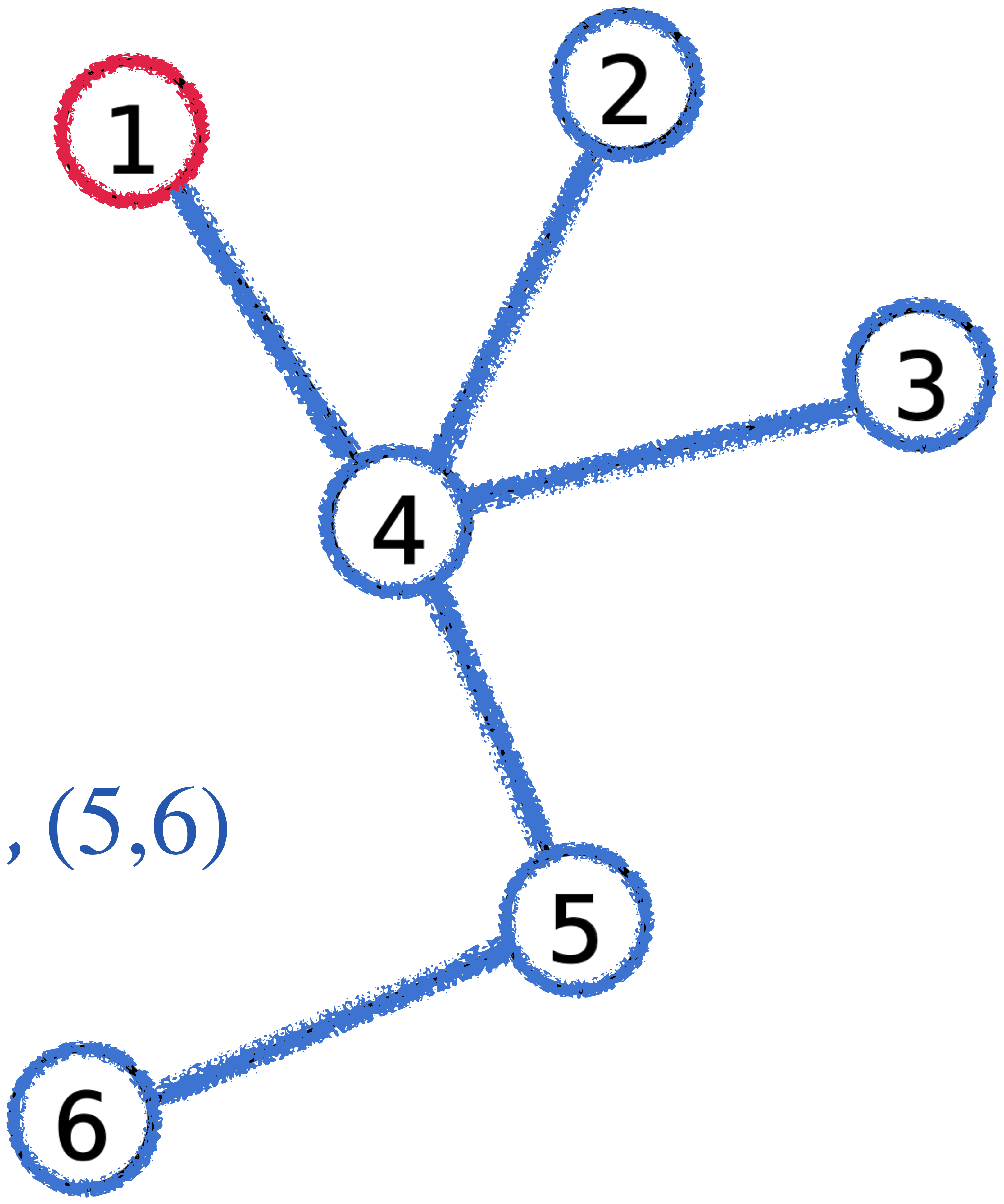
- Check the subgraph for $\ell = 2$
  - Nodes 1,2,3,4,5
  - Edges (1,4), (2,4), (3,4), (4,5)



Local ℓ-sub-graph: example

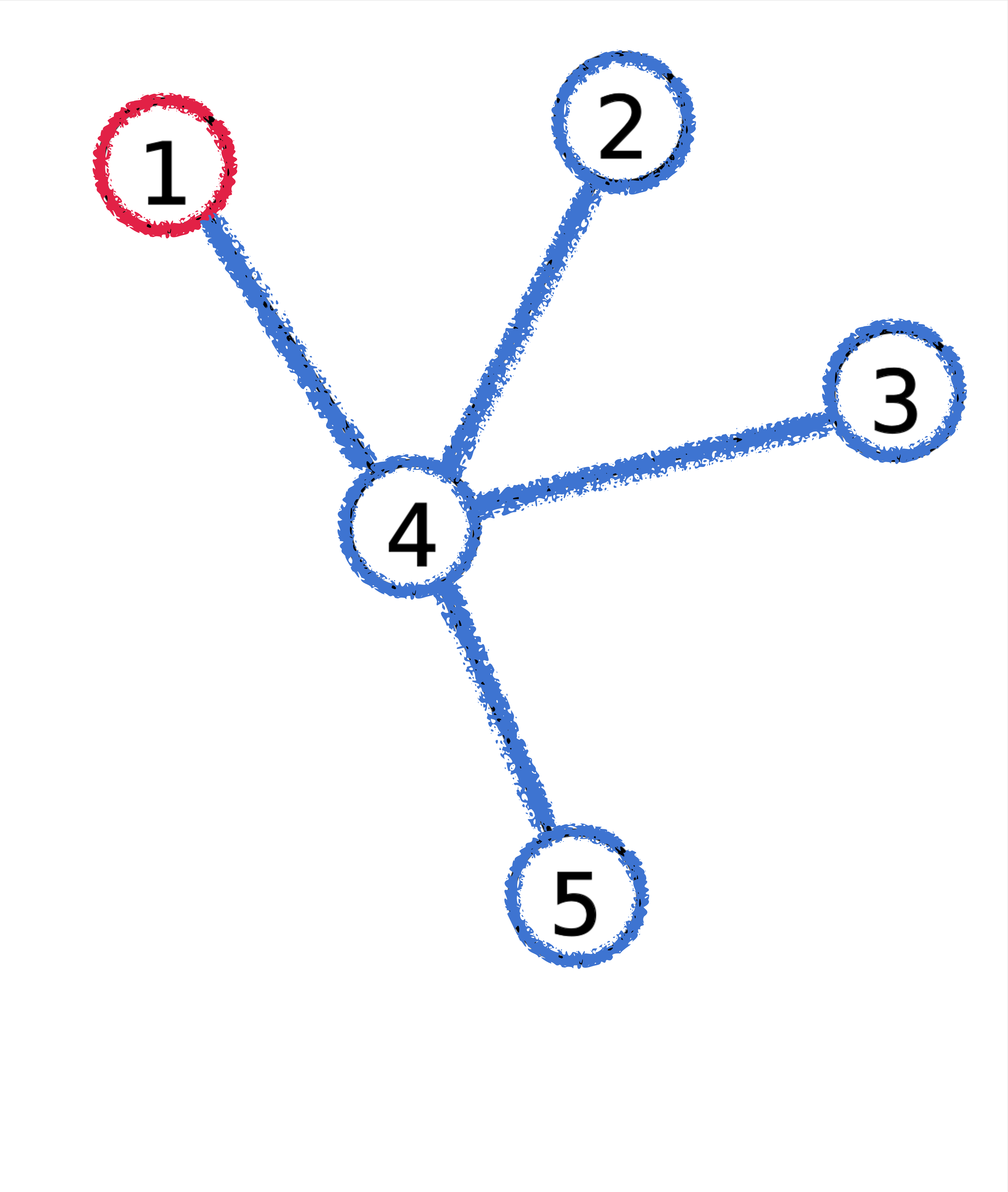
- Pick a node 1

- Check the subgraph for $\ell = 3$
  - Nodes 1,2,3,4,5,6
  - Edges (1,4), (2,4), (3,4), (4,5), (5,6)

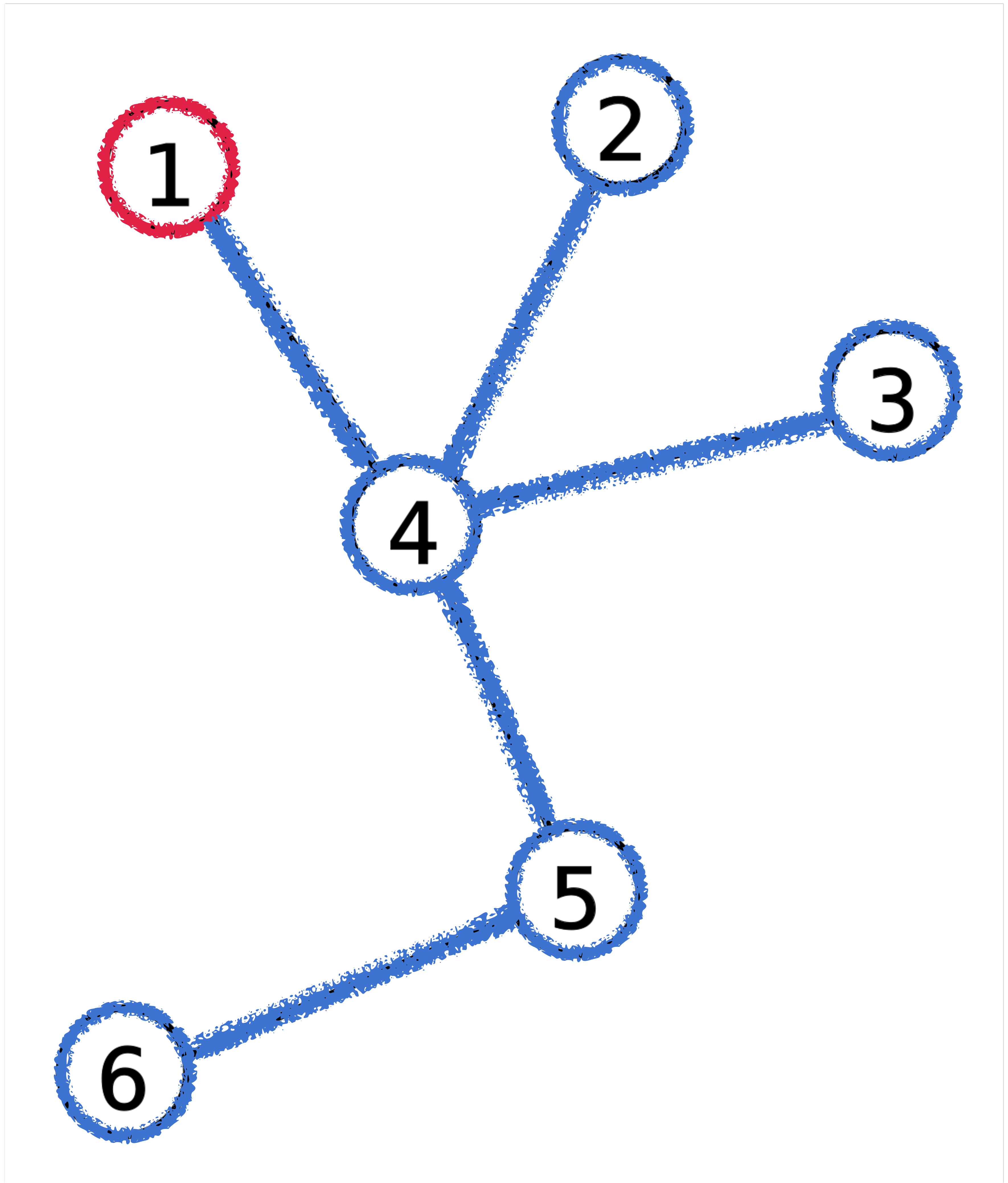


Local ***tree*** ℓ-sub-graphs

$\ell = 2$



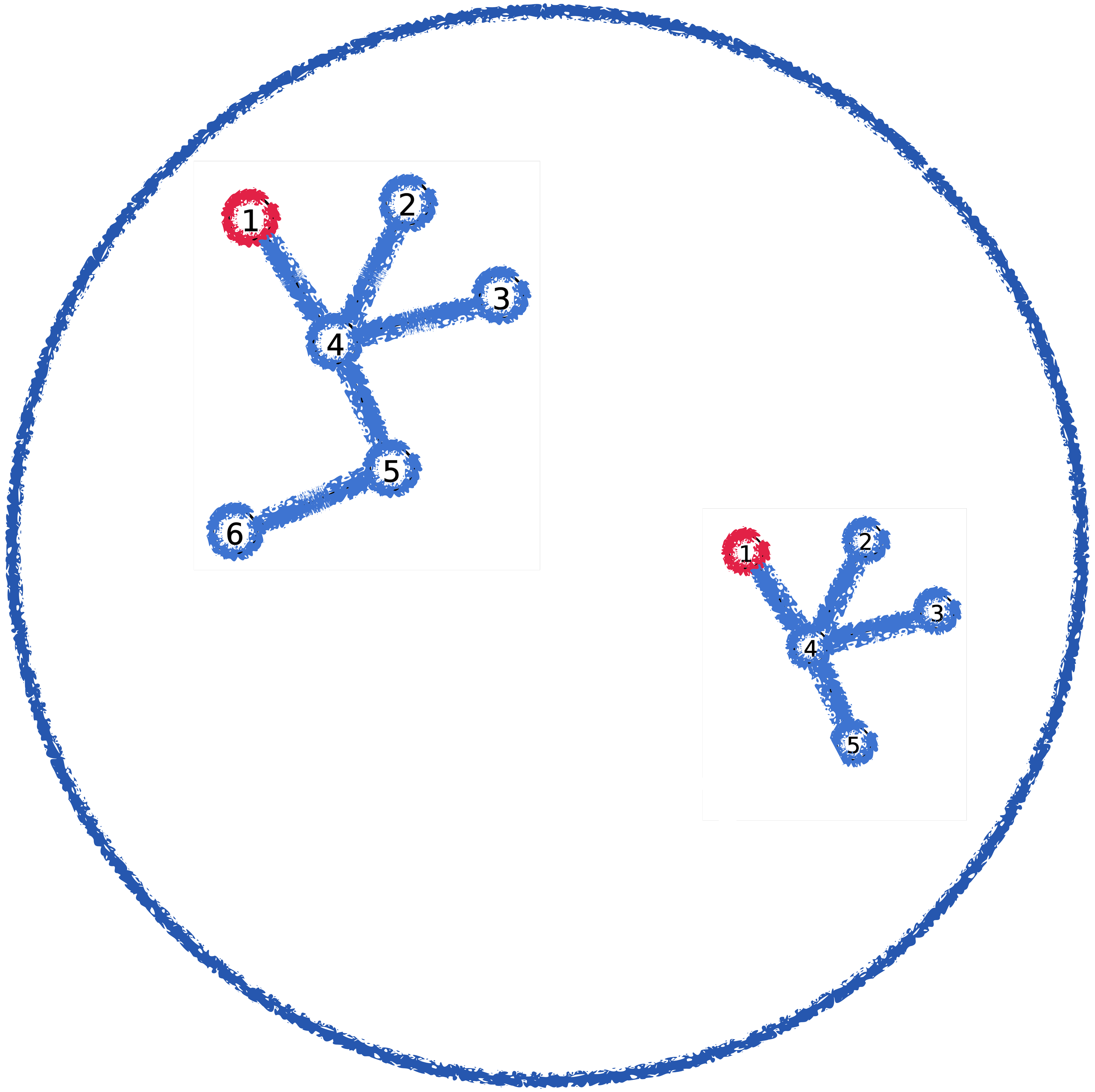
$\ell = 3$



Tree-like graphs

- ***The given graph does not need to be a tree.***

- ***But, up to some ℓ, the local subgraphs have to be a tree.***



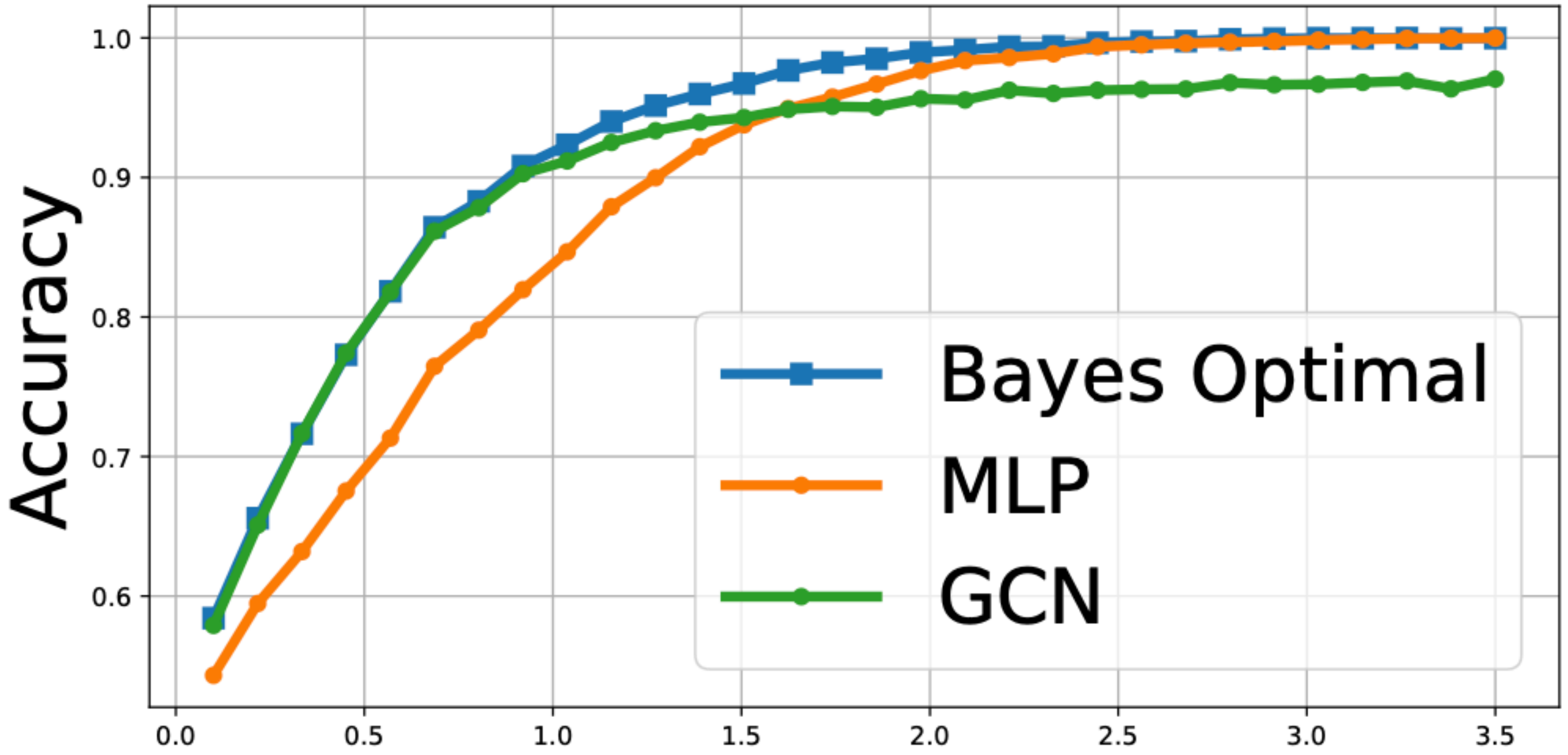
Assumptions: graph

Stochastic Block Model

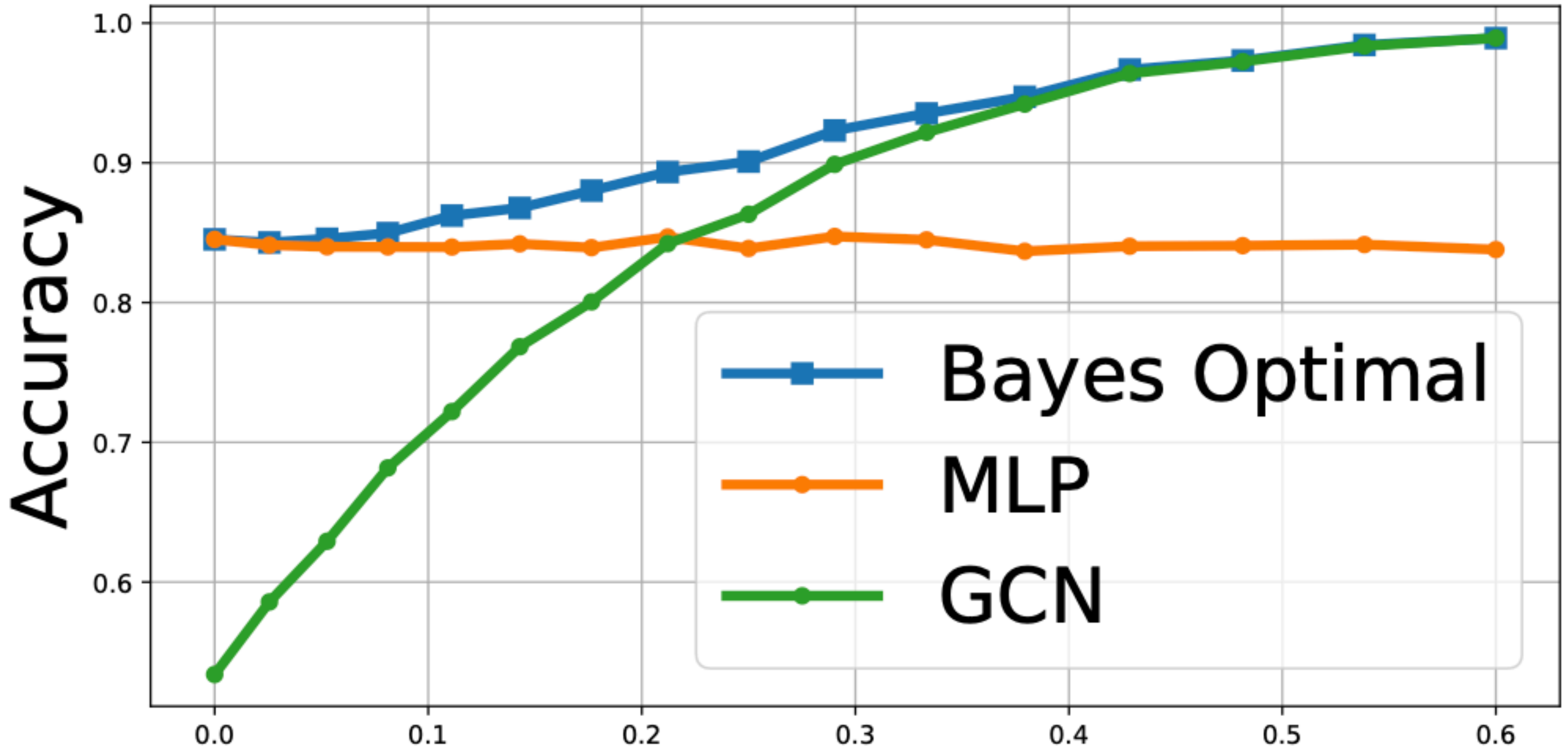
Set the class membership of the nodes (fixed or random)

$$\mathbb{P}(\text{edge } (i,j)) = \begin{cases} p & \text{if } i,j \text{ are in the same class} \\ q & \text{if } i,j \text{ are in the different class} \end{cases}$$

- $p, q = \mathcal{O}(1/n)$, n is the number of nodes in the graph.
- For $\ell \leq O(\log n)$, the ***majority***, i.e.,$1 - o_n(1)$, of the local ℓ-sub-graphs are trees.



SNR of Gaussian Mixture $= \dfrac{\text{distance between means}}{\text{standard deviation}}$



SNR of Stochastic Block Model $= \Gamma = \dfrac{|p-q|}{p+q}$

- We prove that
  - SNR of the SBM $\to 1$, then Bayes Optimal $\to$ Graph Convolution Neural Network (GCN).
  - SNR of the SBM $\to 0$, then Bayes Optimal $\to$ MLP.
  - It interpolates between MLP and GCN in-between.

Thank you!