Dimensionality Selection of Hyperbolic Graph Embeddings using Decomposed Normalized Maximum Likelihood Code-Length

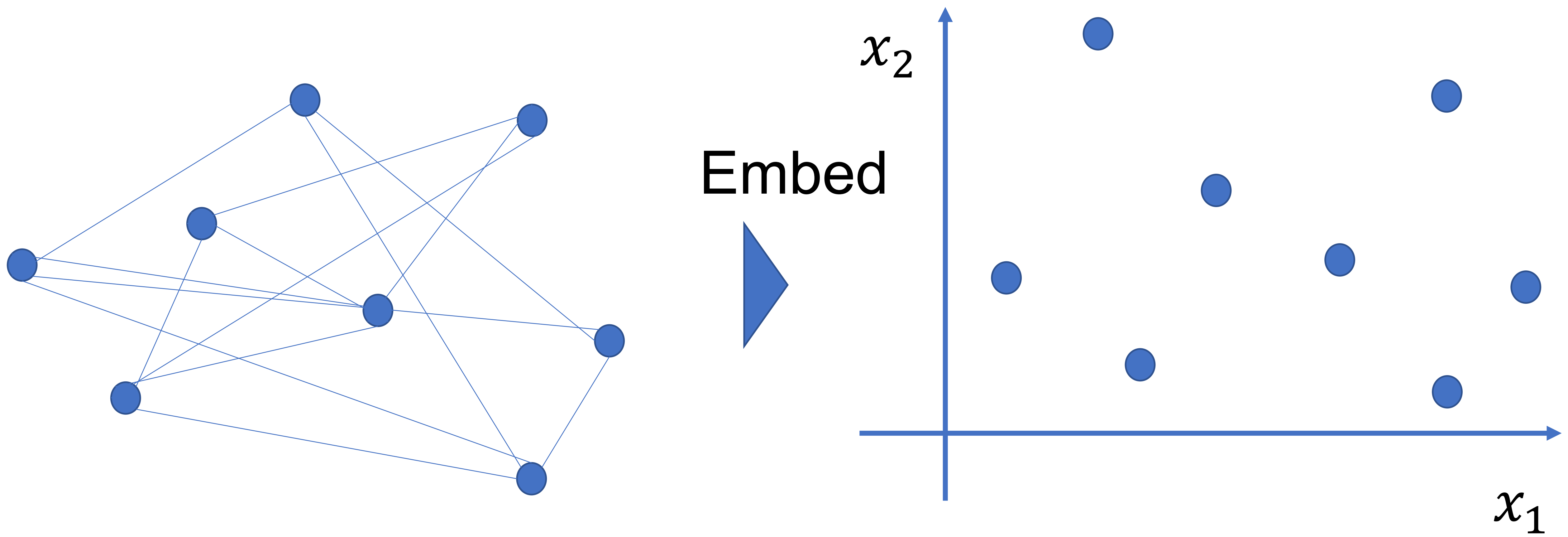
Ryo Yuki, Yuichi Ike, and Kenji Yamanishi
The University of Tokyo

Agenda

1. Background

Graph Embeddings and Dimensionality Selection

Convert discrete representation to continuous one.



- Node classification [1].
- Link prediction [2].
- Clustering [3].
- Visualization [4].

Dimensionality controls

1.performance (e.g., underfitting with low dimensionality and overfitting with high dimensionality).

2.time and space computational complexity.

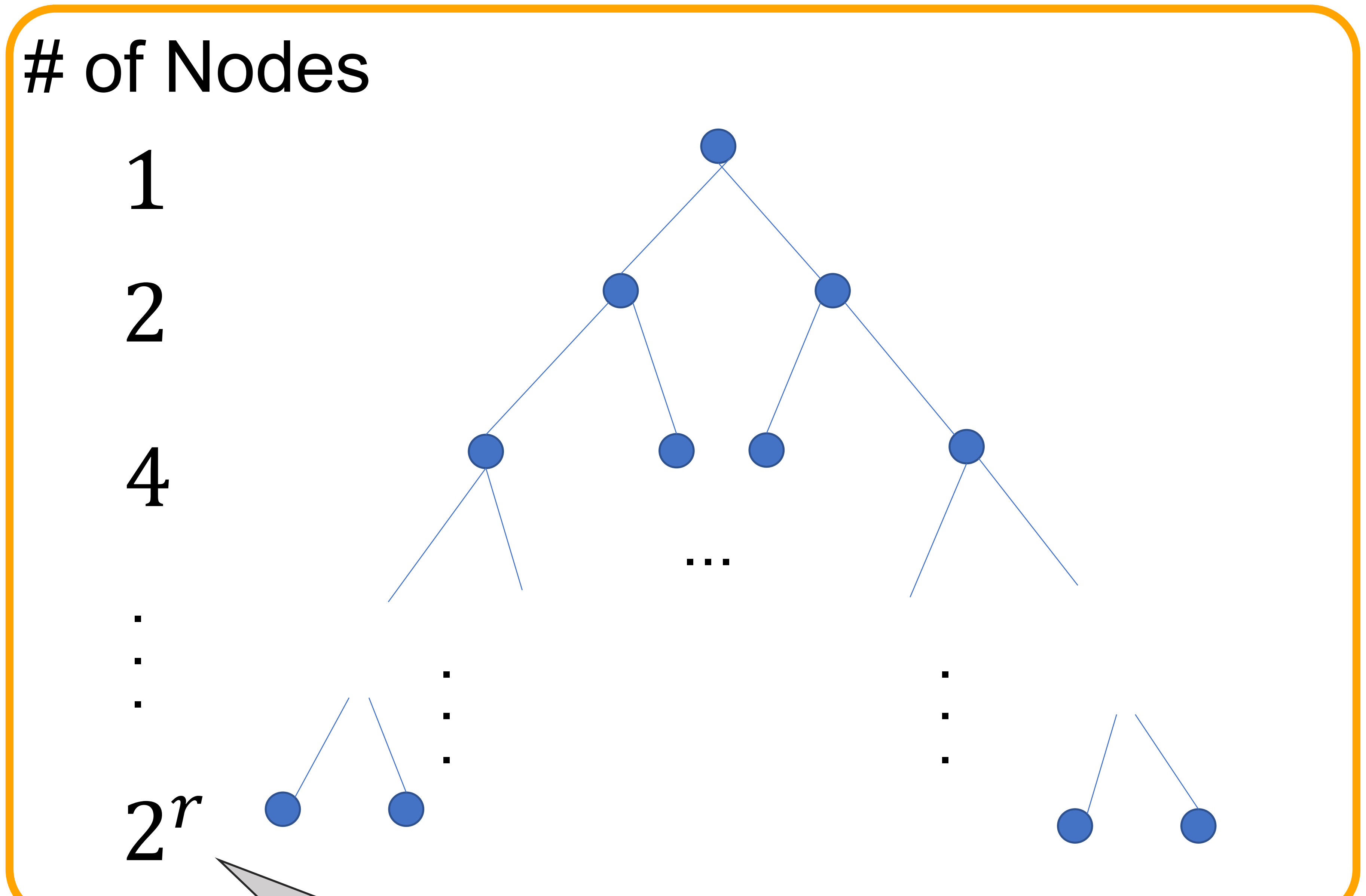
Our contribution: proposed dimensionality selection method for hyperbolic graph embeddings.

Hyperbolic Embeddings

Effective on hierarchical or tree-like structured graphs.

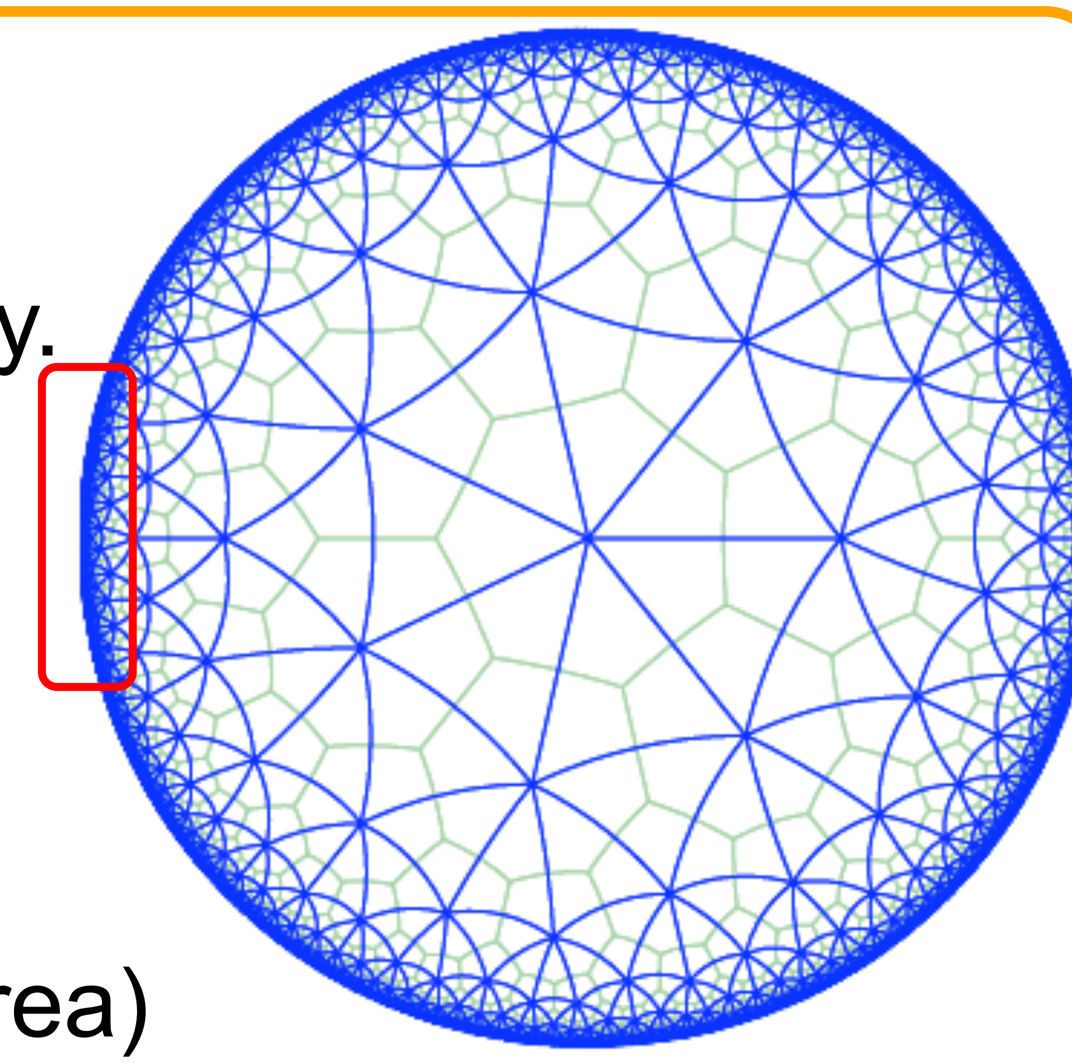
- The performance in **5-dimensional** hyperbolic space is better than that of **200-dimensional** Euclidean space for several graph mining tasks [8].

Tree



Hyperbolic Space

Vaster near the boundary.



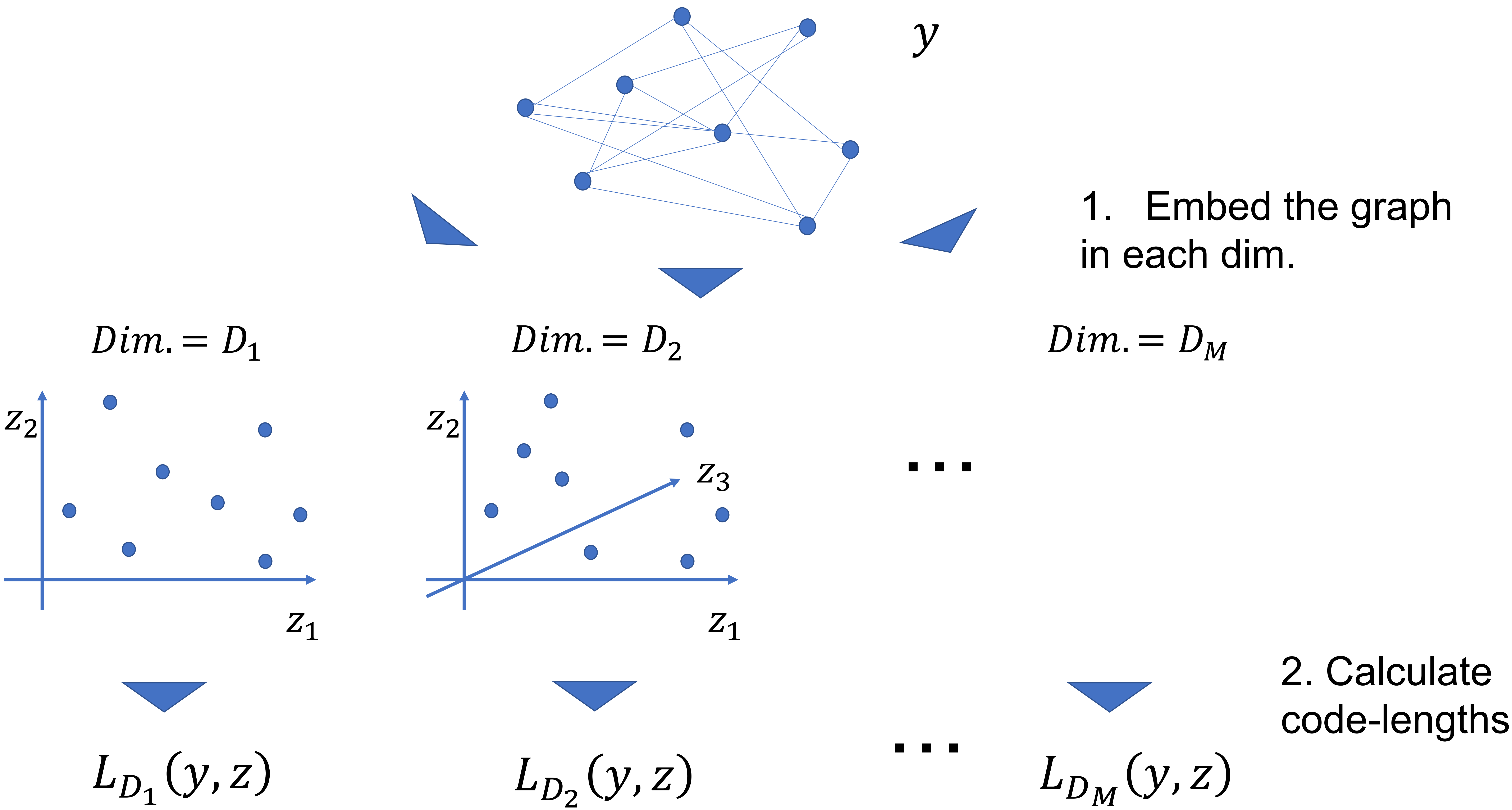
$\mathcal{H}^d$: (surface area) $\propto e^{(d-1)r}$

Fig.1 Triangles in a hyperbolic space (from [9]).

Exponential $\leftarrow$ **Analogy !** $\rightarrow$ **Exponential**

But the surface area in $\mathcal{R}^d$ is proportional to r^{d-1}. $\rightarrow$ **Polynomial !**

Proposed Method



→ Select the dimensionality that minimizes $L_{D_k}(y,z)$.

Agenda

2. Dimensionality Selection using DNML Code-Length

MDL Principle [10]

Select the model that minimizes the code-length.

- One of the information criteria (e.g., AIC [11], BIC [12], etc).
- Theoretical properties such as consistency in model selection [10], etc.

- To apply the MDL principle, we need to do the following:

1. Formulate hyperbolic embeddings as a probabilistic model.

2. Derive $L_D(y, z)$: the encoding function associated with dimensionality $D \in \mathcal{M}$.

1. Latent Variable Model (LVM)

Use LVM with hyperbolic geometric graphs (HGGs).

- Latent variable model (hyperbolic geometric graph, HGG) [9, 13]:

$p(y, z) = p(y|z)p(z)$.

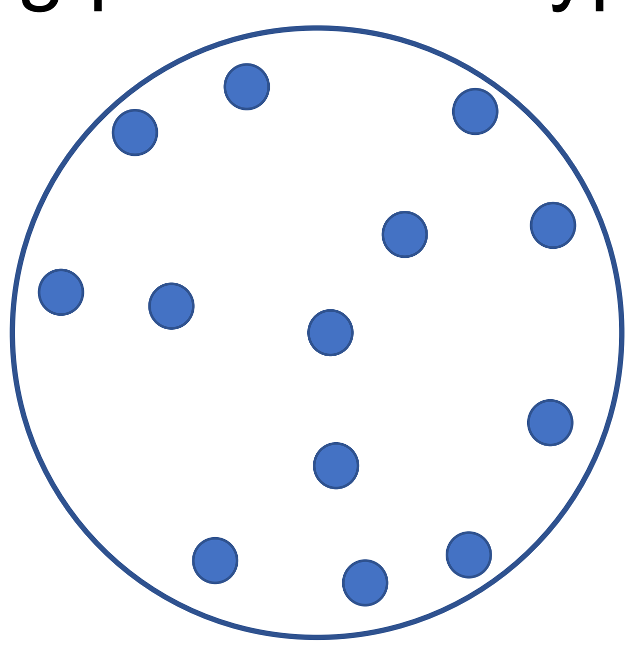
$z = \{z_i\}_{i \in [n]}$: embedding as latent variables.

$y = \{y_{ij}\}_{(i,j) \in \Lambda_{[n]}}$: observed edges.

Properties

- Power law of degree distribution [14].
- High clustering coefficient [14].

1. Sampling points in a hyperbolic space.



$$p(z; \sigma, R) = \prod_{i \in [n]} p(z_i; \sigma, R),$$

$$p(z_i; \sigma, R) = p(r_i; \sigma, R) \prod_{j=1}^{D-1} p(\theta_{ij}),$$

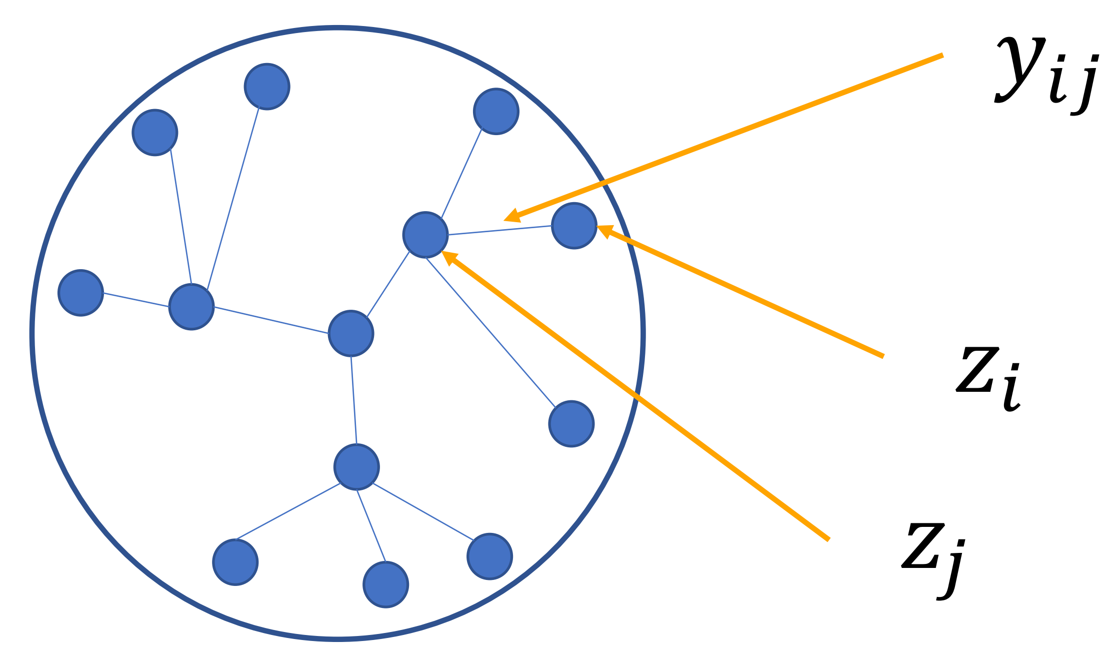
$$p(\theta_{ij}) = \begin{cases} \frac{\sin^{D-1-j} \theta_{ij}}{I_{D,j}} & (j \in [D-2]), \\ \frac{1}{2\pi} & (j = D-1), \end{cases}$$

$$p(r_i; \sigma, R) = \frac{\sinh^{D-1}(\sigma r_i)}{C_D(\sigma)},$$

where $I_{D,j} = \int_0^\pi \sin^{D-1-j} \theta d\theta$ and $C_D(\sigma) = \int_0^R \sinh^{D-1}(\sigma r) dr$.

Pseudo-uniform dist. of hyperbolic

2. Connect points with some probability.



$$p(\boldsymbol{y}; \boldsymbol{\phi}, \beta) = \prod_{(i,j) \in \Lambda_{[n]}} p(y_{ij}; \phi_i, \phi_j, \beta),$$

$$p(y_{ij}; \phi_i, \phi_j, \beta) = \begin{cases} \frac{1}{1+\exp(\beta(d_{\phi_i \phi_j} - R))} & (y_{ij} = 1), \\ 1 - \frac{1}{1+\exp(\beta(d_{\phi_i \phi_j} - R))} & (y_{ij} = 0). \end{cases}$$

Sigmoid function.

2. Decomposed Normalized Maximum Likelihood (DNML) Code-Length [15, 16]

Use DNML code-length for LVM.

- DNML is easy to calculate for latent variable models:

$$L_{\mathrm{DNML}}(\boldsymbol{y}, z) := L_{\mathrm{NML}}(\boldsymbol{y}|z) + L_{\mathrm{NML}}(z),$$

where NML code-lengths [17] are given by

$$L_{\mathrm{NML}}(\boldsymbol{y}|z) := \boxed{-\log p(\boldsymbol{y}|z; \hat{\beta}(\boldsymbol{y}, z))} + \boxed{\log \sum_{\boldsymbol{y}'} p(\boldsymbol{y}'|z; \hat{\beta}(\boldsymbol{y}', z))},$$

Negative logarithm of the maximum likelihood

Penalty term (parametric complexity).

$$L_{\mathrm{NML}}(z) := \boxed{-\log p(z; \hat{\sigma}(z))} + \boxed{\log \int dz' p(z'; \hat{\sigma}(z'))}.$$

Our contribution: derived the explicit form of the approximation of each penalty term.

Approximation of DNML #1

NML is approximated with Fisher information [17, 18].

$$L_{\mathrm{NML}}(\boldsymbol{y}|\boldsymbol{z}) \approx -\log p(\boldsymbol{y}|\boldsymbol{z};\hat{\beta}(\boldsymbol{y},\boldsymbol{z})) + \frac{1}{2}\log\frac{n(n-1)}{4\pi} + \underline{\log\int_{\beta_{\min}}^{\beta_{\max}}\sqrt{|I_n(\beta)|}\,d\beta},$$

Integral over the parameter domain.

$$L_{\mathrm{NML}}(\boldsymbol{z}) \approx -\log p(\boldsymbol{z};\hat{\sigma}(\boldsymbol{z})) + \frac{1}{2}\log\frac{n}{2\pi} + \underline{\log\int_{\sigma_{\min}}^{\sigma_{\max}}\sqrt{|I(\sigma)|}\,d\sigma},$$

Integral over the parameter domain.

$$I(\sigma) := \lim_{n\to\infty}\frac{1}{n}E_{\sigma}\left[-\frac{\partial^2\log p(z;\sigma)}{\partial\sigma^2}\right] = E_{\sigma}\left[-\frac{\partial^2\log p(z;\sigma)}{\partial\sigma^2}\right].$$

$$I_n(\beta) := E_{\beta}\left[-\frac{2}{n(n-1)}\frac{\partial^2}{\partial\beta^2}\sum_{(i,j)\in\Lambda_{[n]}}\log p(y_{ij}|z_i,z_j;\beta)\right].$$

Fisher information

Approximation of DNML #2

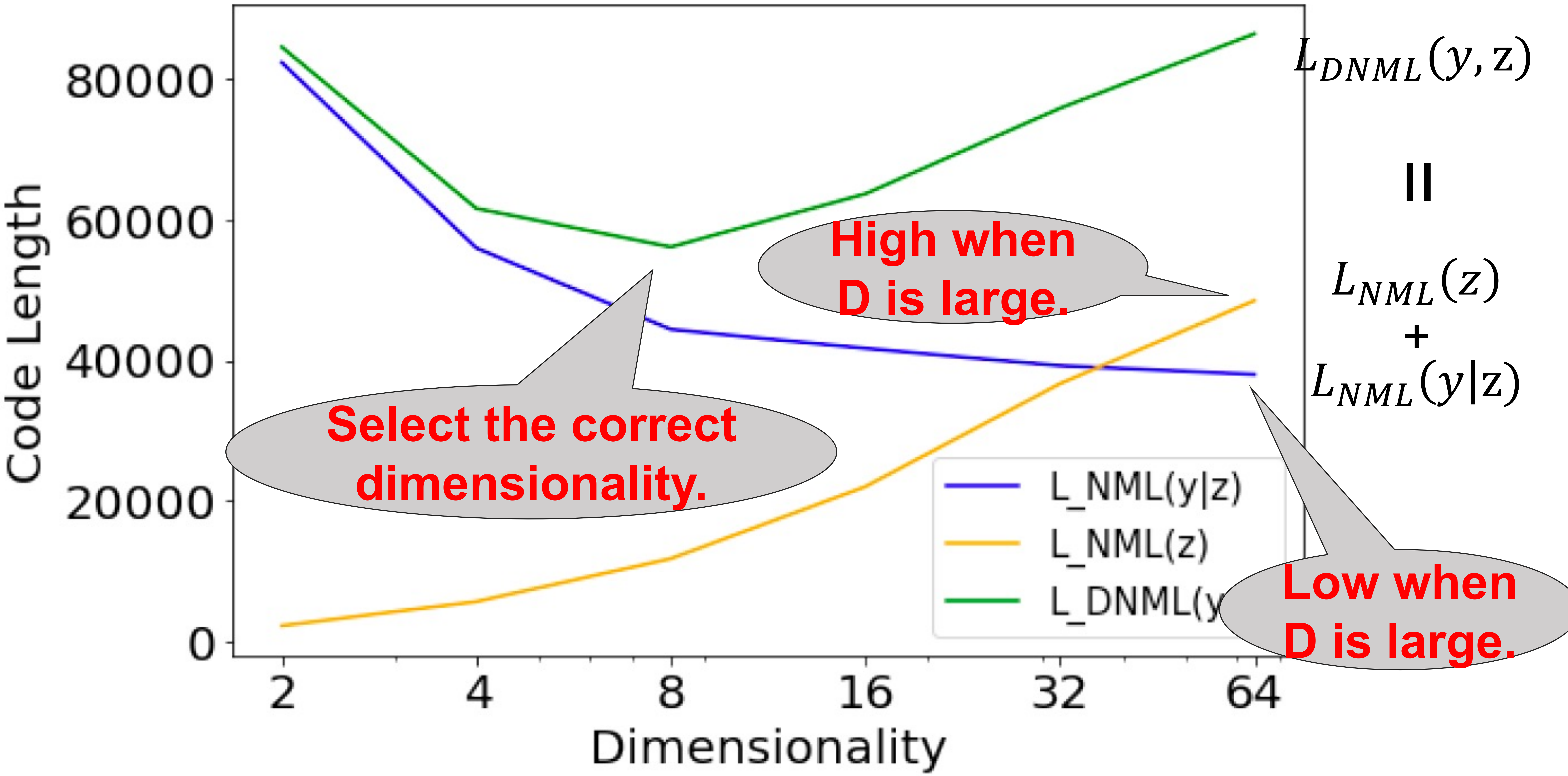
- With some calculation, the Fisher information are given by:

$$I_n(\beta) = \frac{2}{n(n-1)} \sum_{(i,j)\in\Lambda_{[n]}} \frac{(R - d_{z_i z_j})^2 \exp(-\beta(R - d_{z_i z_j}))}{\left(1 + \exp(-\beta(R - d_{z_i z_j}))\right)^2},$$

$$I(\sigma) = (D-1)^2 \frac{\int_0^R r^2 \cosh^2(\sigma r) \sinh^{D-3}(\sigma r) dr}{C_D(\sigma)} + \left\{ \frac{\int_0^R (D-1) r' \cosh(\sigma r') \sinh^{D-2}(\sigma r') dr'}{C_D(\sigma)} \right\}^2.$$

- Numerical integration with the Gaussian quadrature [19].

Typical Behavior of DNML



$$L_{DNML}(y, z) = L_{NML}(z) + L_{NML}(y|z)$$

Agenda

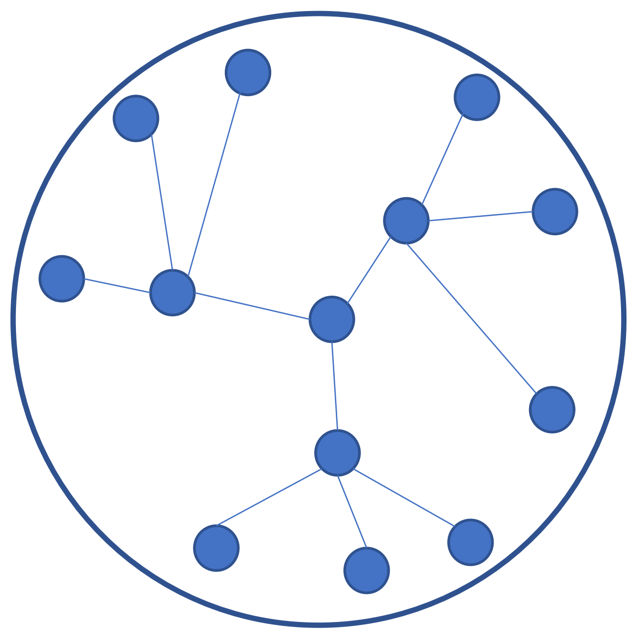
3. Experimental Results

1. Artificial Dataset

DNML-HGG selected the correct dimensionality with enough data.

Setting

1. Generate artificial graphs with $D_{true} = 16$.



2. Estimate $\widehat{D}$ with the proposed method and the following competitive methods:

- AIC [11]
- BIC [12]
- MinGE [20]
  (Euclidean dimensionality selection method)

Results

Table 1 Average benefits on HGG-16.

# of nodes	DNML-HGG	AIC	BIC	MinGE
400	**0.042**	0.000	0.000	0.000
800	**0.250**	0.000	0.000	0.000
1600	**0.042**	0.000	0.000	0.000
3200	0.000	**0.250**	0.000	0.000
6400	**1.000**	0.375	0.000	0.000
12800	**1.000**	0.542	0.000	0.000

Best !

$$b(\hat{D}, D_{\mathrm{true}}) = \max\left\{0, 1 - \frac{|\log_2 \hat{D} - \log_2 D_{\mathrm{true}}|}{T_{\mathrm{gap}}}\right\},$$

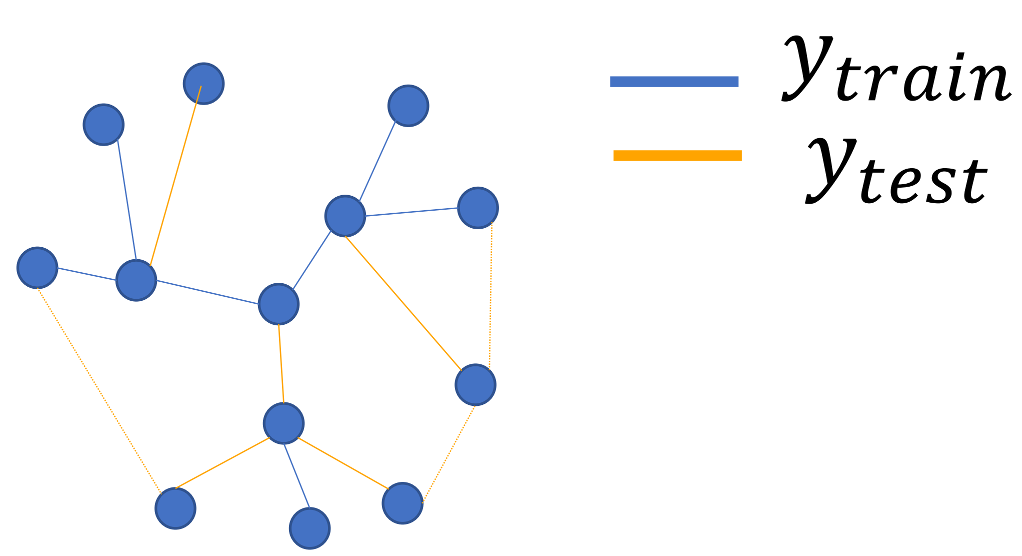
where $T_{gap} = 2, D_{true} = 16.$

2. Scientific Collaboration Networks

The selected dimensionality suppresses the computational resource, whereas keeping the AUC.

Setting

1. Separate y into y_{train} and y_{test}.



2. Train with y_{train} and estimate $\hat{D}$.
Then, perform link prediction on y_{test}.

Results

Selected Dim. Max. Performance

AUC (Four graphs)

Network	2	4	8	16	32	64
AstroPh	0.549	0.813	0.842	0.846	0.849	**0.850**
CondMat	0.522	0.754	**0.767**	0.764	0.766	**0.767**
GrQc	0.526	0.728	0.744	**0.756**	0.755	**0.756**
HepPh	0.559	0.847	0.880	0.880	0.883	**0.885**

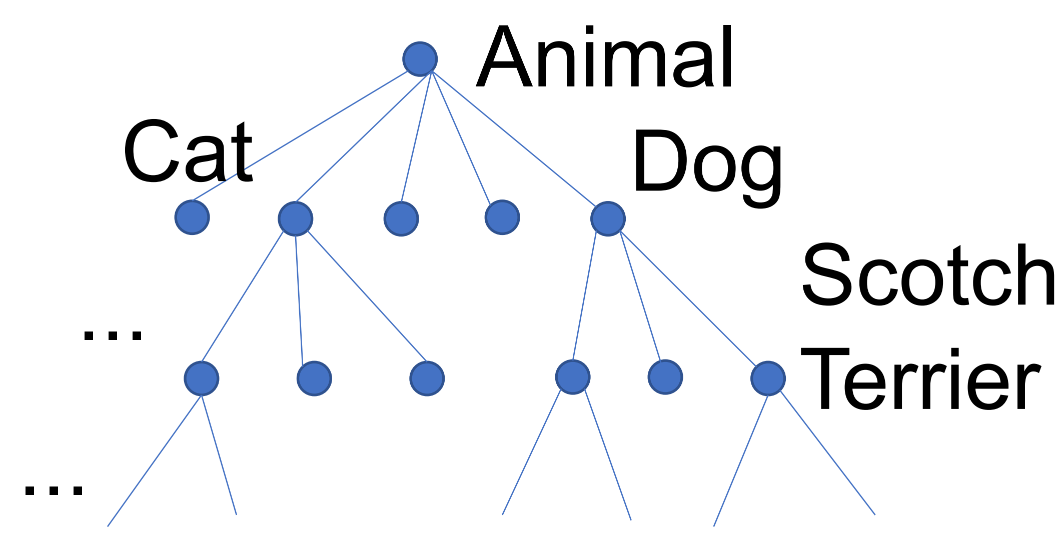
≥(Maximum)-0.01

3. WordNet[21] Datasets

DNML-HGG selected the dimensionality that preserves the hierarchy of the graphs.

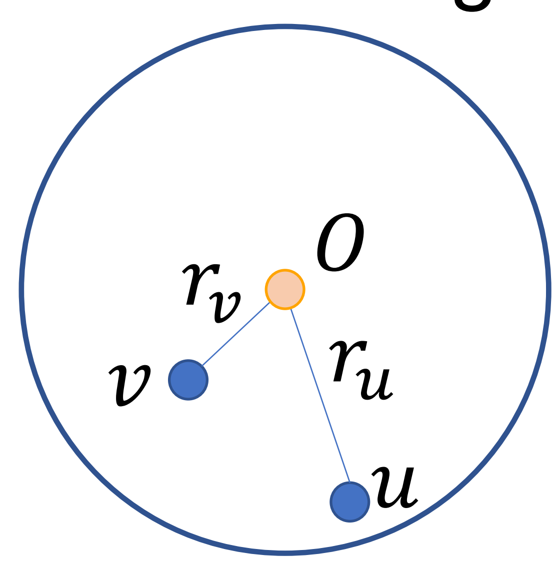
Setting

1. Generate graphs with hierarchical structure using is-a relation.



2. Calculate is-a-scores.

Embedding



$$\text{is-a-score}(\boldsymbol{u}, \boldsymbol{v}) = (\alpha(r_u - r_v) - 1) d_{\boldsymbol{uv}},$$

→High when "u is a v".

Results

Benefit
(Average of six graphs)

DNML-HGG	AIC	BIC	MinGE
0.714	0.286	0.500	0.000

Best !

$$b(\hat{D}, D_{\text{true}}) = \max\left\{0, 1 - \frac{|\log_2 \hat{D} - \log_2 D_{\text{true}}|}{T_{\text{gap}}}\right\},$$

where $T_{gap} = 2, D_{true} = \max_D is - a - score.$

Agenda

4. Summary
&
Future Perspectives

Summary & Future Perspective

- **Summary**
  - Contribution #1: proposed dimensionality selection method for hyperbolic graph embeddings.
    - Few studies for hyperbolic embeddings [22].
  - Contribution #2: derived the explicit form of the approximation of DNML.
  - Experimentally showed the effectiveness of the proposed method.

- **Future Perspective**
  - Extension for other Riemannian manifolds.
    - Euclidean spaces, spherical spaces, etc…

Reference #1

[1] Bhagat, Smriti, Graham Cormode, and S. Muthukrishnan. "Node classification in social networks." *Social network data analytics*. Springer, Boston, MA, 2011. 115-148.

[2] Liben-Nowell, David, and Jon Kleinberg. "The link-prediction problem for social networks." *Journal of the American society for information science and technology* 58.7 (2007): 1019-1031.

[3] Ding, Chris HQ, et al. "A min-max cut algorithm for graph partitioning and data clustering." *Proceedings 2001 IEEE international conference on data mining*. IEEE, 2001.

[4] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of machine learning research* 9.11 (2008).

[5] Yin, Zi, and Yuanyuan Shen. "On the dimensionality of word embedding." *arXiv preprint arXiv:1812.04224* (2018).

[6] Luo, Gongxu, et al. "Graph entropy guided node embedding dimension selection for graph neural networks." *arXiv preprint arXiv:2105.03178* (2021).

[7] Hung, Pham Thuc, and Kenji Yamanishi. "Word2vec skip-gram dimensionality selection via sequential normalized maximum likelihood." *Entropy* 23.8 (2021): 997.

[8] Nickel, Maximillian, and Douwe Kiela. "Poincaré embeddings for learning hierarchical representations." *Advances in neural information processing systems* 30 (2017): 6338-6347.

[9] Krioukov, Dmitri, et al. "Hyperbolic geometry of complex networks." *Physical Review E* 82.3 (2010): 036106.

[10] Rissanen, Jorma. *Optimal estimation of parameters*. Cambridge University Press, 2012.

Reference #2

[11] Akaike, Hirotogu. "Information theory and an extension of the maximum likelihood principle." *Selected papers of hirotugu akaike*. Springer, New York, NY, 1998. 199-213.

[12] Schwarz, Gideon. "Estimating the dimension of a model." *The annals of statistics* (1978): 461-464.

[13] Yang, Weihua, and David Rideout. "High Dimensional Hyperbolic Geometry of Complex Networks." *Mathematics* 8.11 (2020): 1861.

[14] Barabási, Albert-László. "Network science." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1987 (2013): 20120375.

[15] Yamanishi, K., Wu, T., Sugawara, S., & Okada, M. (2019). The decomposed normalized maximum likelihood code-length criterion for selecting hierarchical latent variable models. *Data Mining and Knowledge Discovery*, *33*(4), 1017-1058.

[16] Wu, T., Sugawara, S., & Yamanishi, K. (2017, August). Decomposed normalized maximum likelihood codelength criterion for selecting hierarchical latent variable models. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1165-1174).

[17] Shtar'kov, Yurii Mikhailovich. "Universal sequential coding of single messages." *Problemy Peredachi Informatsii* 23.3 (1987): 3-17.

[18] Grünwald, P. D., Myung, I. J., & Pitt, M. A. (Eds.). (2005). *Advances in minimum description length: Theory and applications*. MIT press.

[19] Vetterling, W. T., Vetterling, W. T., Press, W. H., Press, W. H., Teukolsky, S. A., Flannery, B. P., & Flannery, B. P. (1992). *Numerical recipes: example book C*. Cambridge University Press.

[20] Luo, Gongxu, et al. "Graph entropy guided node embedding dimension selection for graph neural networks." *arXiv preprint arXiv:2105.03178* (2021).

Reference #3

[21] Miller, George A. *WordNet: An electronic lexical database*. MIT press, 1998.

[22] Almagro, Pedro, Marián Boguñá, and M. Serrano. "Detecting the ultra low dimensionality of real networks." *Nature communications* 13.1 (2022): 1-10.

[23] Nickel, Maximillian, and Douwe Kiela. "Learning continuous hierarchies in the lorentz model of hyperbolic geometry." *International Conference on Machine Learning*. PMLR, 2018.

[24] Amari, S. I. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, *5*(4-5), 185-196.

[25] Bonnabel, Silvere. "Stochastic gradient descent on Riemannian manifolds." *IEEE Transactions on Automatic Control* 58.9 (2013): 2217-2229.

[24] Enokida, Y., Suzuki, A., & Yamanishi, K. (2018). Stable geodesic update on hyperbolic space and its application to poincare embeddings. *arXiv preprint arXiv:1805.10487*.

[26] Gongxu Luo, Jianxin Li, Hao Peng, Carl Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2021. Graph Entropy Guided Node Embedding Dimension Selection for Graph Neural Networks. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2767–2774. MainTrack

Appendix: Hyperboloid Model [23]

Hyperbolic Space:

$$\mathbb{H}^D := \{\boldsymbol{x} = (x_0, x_1, \ldots, x_D)^\top \mid \boldsymbol{x} \in \mathbb{R}^{D+1}, < x, x >_{\mathcal{L}} = -1, x_0 > 0\}$$

where $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{L}} = \boldsymbol{u}^\top g_D \boldsymbol{v}.$ $g_D = \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(D+1)\times(D+1)},$

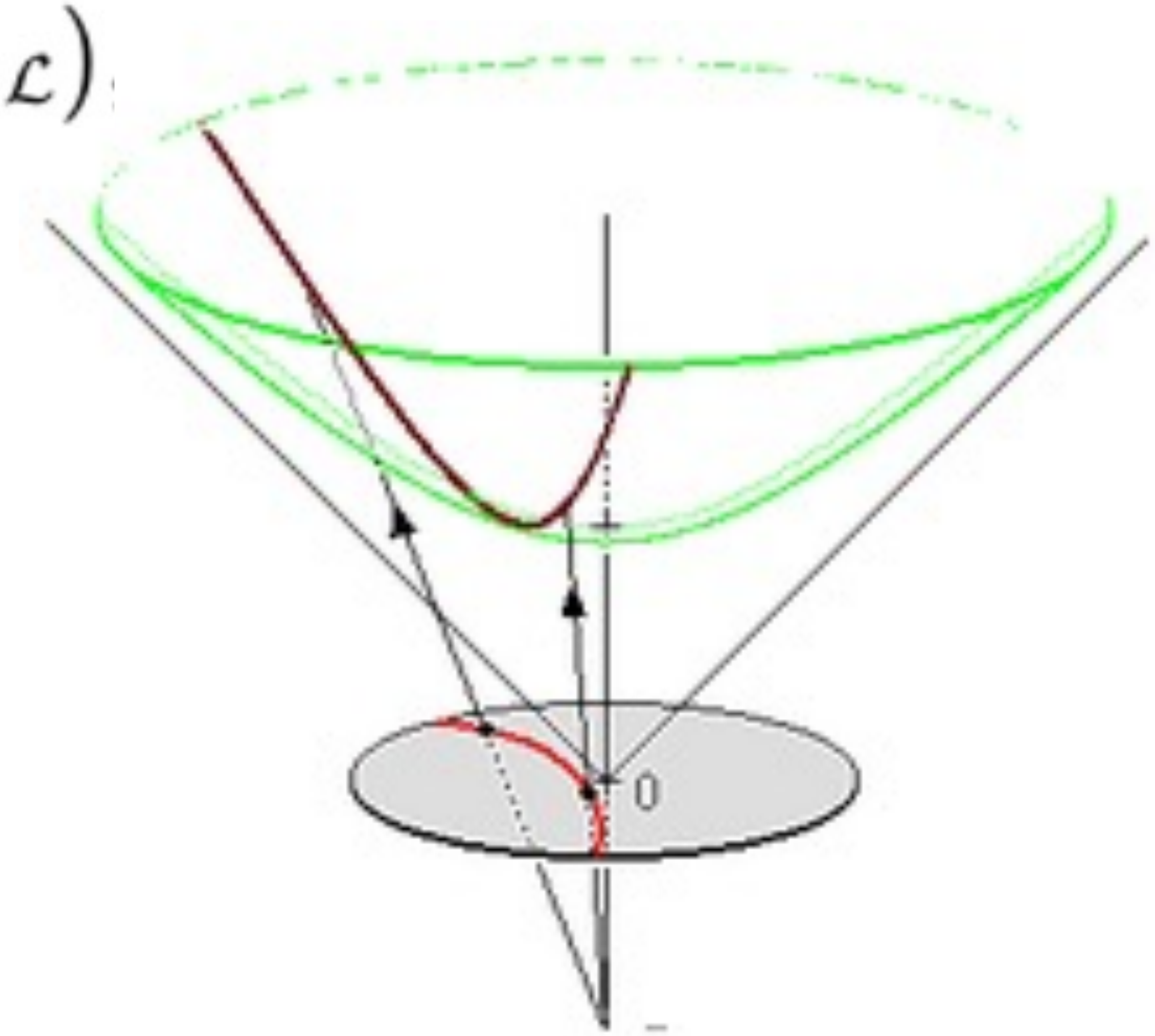
Distance: $d_{\boldsymbol{uv}} = \operatorname{arcosh}(- < \boldsymbol{u}, \boldsymbol{v} >_{\mathcal{L}})$

Exponential Map:

$$\operatorname{Exp}_{\boldsymbol{z}}(\boldsymbol{u}) := \cosh(\sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle_{\mathcal{L}}})\boldsymbol{z} + \sinh(\sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle_{\mathcal{L}}})\frac{\boldsymbol{v}}{\sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle_{\mathcal{L}}}},$$

Merits

- Numerically stable.
- Exponential map is easy to implement.



*https://en.wikipedia.org/wiki/Hyperboloid_model

Appendix: Loss Function

Optimize $-\log p(\boldsymbol{y}, \boldsymbol{z}; \boldsymbol{\beta}, \boldsymbol{\sigma})$ through stochastic framework [24].

$$
\begin{aligned}
L(z, \beta, \sigma) &= -\log p(y|z;\beta) - \log p(z;\sigma) \\
&= \underbrace{\sum_{(i,j)\in\Lambda_{[n]}} -\log p(y_{ij}|z_i, z_j;\beta)}_{\propto n^2} + \underbrace{\sum_{i\in[n]} -\log p(z_i;\sigma)}_{\propto n}
\end{aligned}
$$

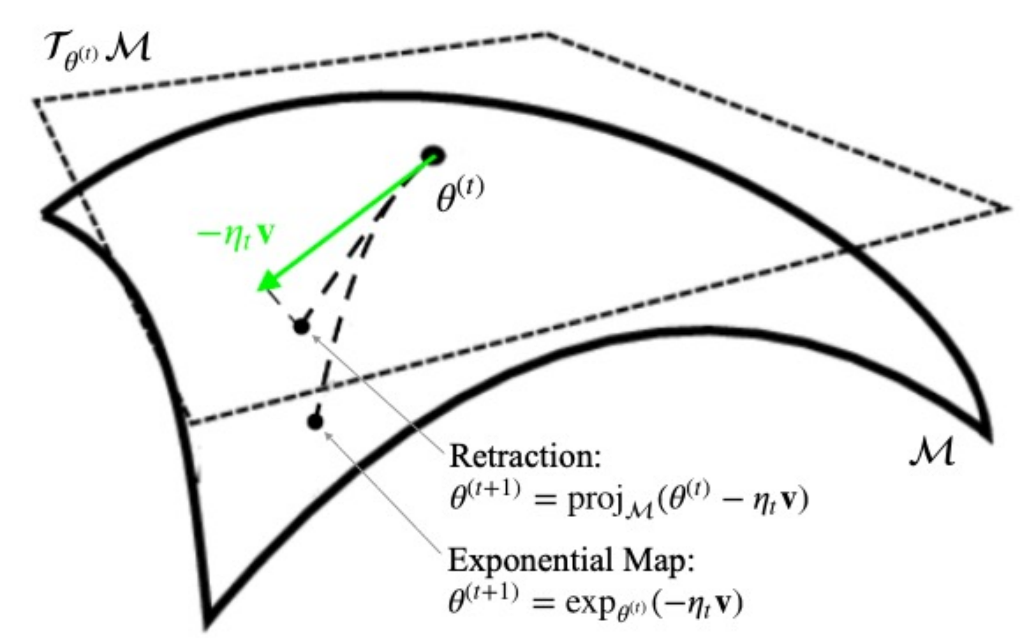
→Different sample size.

$$
= \sum_{(i,j)\in\Lambda_{[n]}} \left\{ -\log p(y_{ij}|z_i, z_j;\beta) - \frac{1}{n-1}\log p(z_i;\sigma) - \frac{1}{n-1}\log p(z_j;\sigma) \right\}.
$$

Summation over all possible pairs.

Division by n-1.

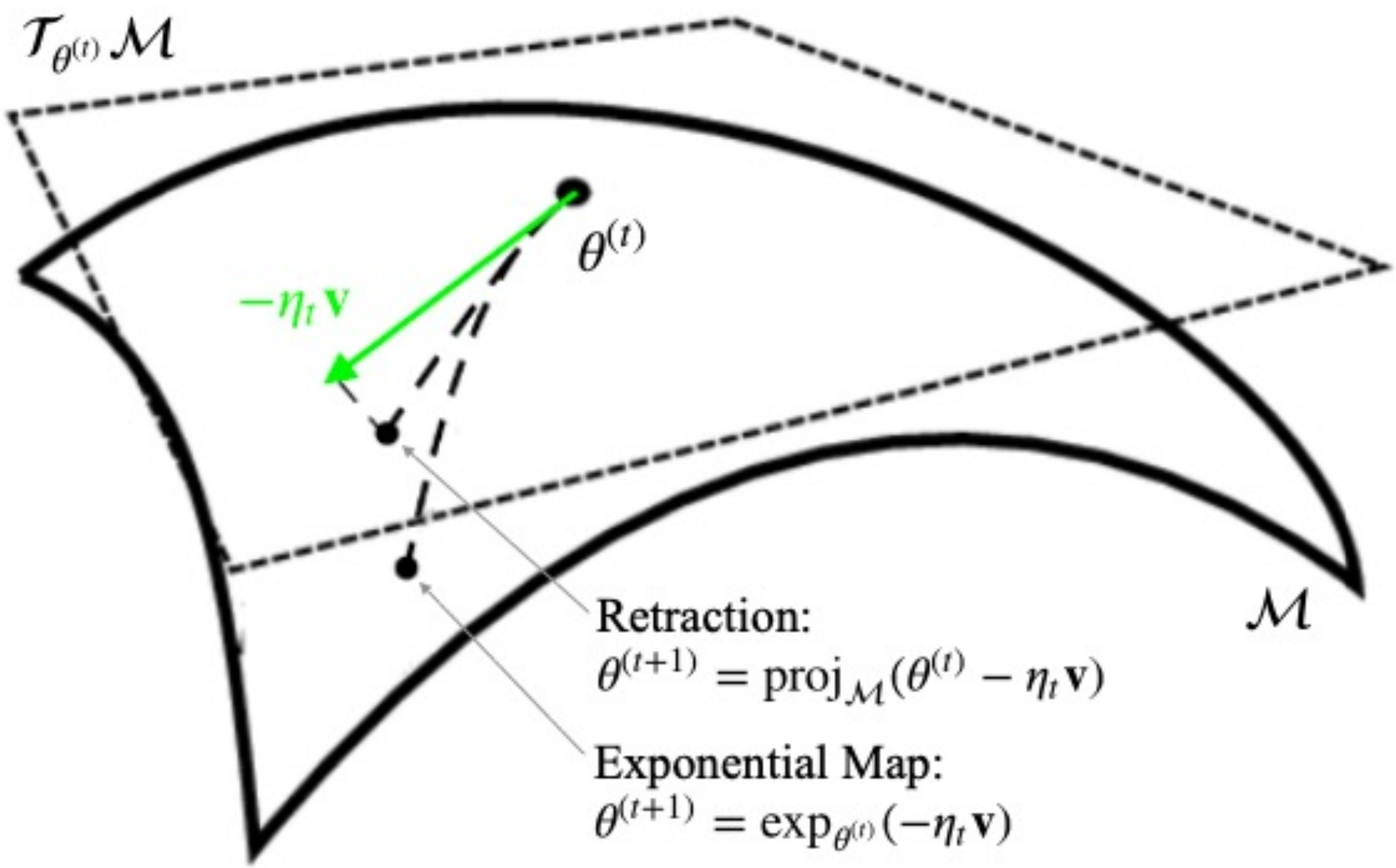
→Uniformly sample S ⊂(possible pairs).
Riemannian Gradient Descent [25].



*https://andbloch.github.io/Stochastic-Gradient-Descent-on-Riemannian-Manifolds/

Appendix: Riemannian Gradient Descent

Use Riemannian Gradient Descent [25].



*https://andbloch.github.io/Stochastic-Gradient-Descent-on-Riemannian-Manifolds/

Appendix: Derivation of DNML #1

For $x_{ij} := R - d_{z_i z_j} \in [-R, R]$, we reformulate $p(y_{ij}|z_i, z_j; \beta)$ as follows:

$$
\begin{aligned}
p(y_{ij}|z_i, z_j; \beta) &= p(y_{ij}|x_{ij}; \beta), \\
&= p_{ij}^{y_{ij}} (1 - p_{ij})^{(1-y_{ij})},
\end{aligned}
$$

where $p_{ij} = 1/(1+\exp(-\beta x_{ij}))$. This form is the logistic regression with the constraint $\beta \in [0, \beta_{\max}]$. Then, the negative logarithm of the likelihood $L(\beta)$ is

$$
\begin{aligned}
L(\beta) &:= - \sum_{(i,j)\in\Lambda_{[n]}} \log p(y_{ij}|x_{ij}; \beta), \\
&= - \sum_{(i,j)\in\Lambda_{[n]}} \left\{ y_{ij} \log \frac{p_{ij}}{1 - p_{ij}} + \log(1 - p_{ij}) \right\}, \\
&= - \sum_{(i,j)\in\Lambda_{[n]}} \left\{ y_{ij} \beta x_{ij} + \log(1 + \exp(\beta x_{ij})) \right\}.
\end{aligned}
$$

Appendix: Derivation of DNML #2

Hence, we obtain

$$\frac{\partial L(\beta)}{\partial \beta} = -\sum_{(i,j)\in\Lambda_{[n]}} \left\{ y_{ij}x_{ij} - \frac{x_{ij}}{1+\exp(-\beta x_{ij})} \right\},$$

$$\frac{\partial^2 L(\beta)}{\partial \beta^2} = \sum_{(i,j)\in\Lambda_{[n]}} \frac{x_{ij}^2 \exp(-\beta x_{ij})}{\left(1+\exp(-\beta x_{ij})\right)^2}.$$

Since $\partial^2 L(\beta)/\partial\beta^2$ is independent of y_{ij}, we derive

$$I_n(\beta) = \frac{2}{n(n-1)} \sum_{(i,j)\in\Lambda_{[n]}} \frac{x_{ij}^2 \exp(-\beta x_{ij})}{\left(1+\exp(-\beta x_{ij})\right)^2}.$$

Appendix: Derivation of DNML #3

The negative logarithm of the likelihood for $z = (r, \theta_1, \ldots, \theta_{D-1})$ is

$$L(\sigma) := -\log \frac{\sinh^{D-1}(\sigma r)}{C_D(\sigma)} - \sum_{j=1}^{D-2} \log \frac{\sin^{D-1-j}\theta_j}{I_{D,j}} - \log \frac{1}{2\pi}.$$

Interchanging the derivative and the integral, we obtain

$$\frac{\partial L(\sigma)}{\partial \sigma}$$
$$= -(D-1)\frac{r\cosh \sigma r}{\sinh \sigma r} + \frac{\int_0^R (D-1) r' \cosh(\sigma r') \sinh^{D-2}(\sigma r') dr'}{C_D(\sigma)}.$$

Similarly, we get

$$\frac{\partial^2 L(\sigma)}{\partial \sigma^2} = (D-1)\frac{r^2}{\sinh^2(\sigma r)}$$
$$+(D-1)\frac{\int_0^R r'^2 \sinh^{D-1}(\sigma r') + (D-2) r'^2 \cosh^2(\sigma r') \sinh^{D-3}(\sigma r') dr'}{C_D(\sigma)}$$
$$-\left\{\frac{\int_0^R (D-1) r' \cosh(\sigma r') \sinh^{D-2}(\sigma r') dr'}{C_D(\sigma)}\right\}^2,$$

Appendix: Derivation of DNML #4

Note that the second and third terms are independent of r. The expectation of the first term with respect to r is calculated as

$$(D-1)\int_0^R \frac{\sinh^{D-1}(\sigma r)}{C_D(\sigma)} \cdot \frac{r^2}{\sinh^2(\sigma r)} dr = \frac{D-1}{C_D(\sigma)} \int_0^R r^2 \sinh^{D-3}(\sigma r) dr.$$

Finally, we derive the following:

$$I(\sigma) = (D-1)^2 \frac{\int_0^R r^2 \cosh^2(\sigma r) \sinh^{D-3}(\sigma r) dr}{C_D(\sigma)} - \left\{ \frac{\int_0^R (D-1) r' \cosh(\sigma r') \sinh^{D-2}(\sigma r') dr'}{C_D(\sigma)} \right\}^2.$$

Appendix: Competitive Methods

- Three methods were chosen:

$$\mathrm{AIC}(\boldsymbol{y}, z; D) := -\log p(\boldsymbol{y}|z; \hat{\beta}(\boldsymbol{y}, z), \hat{\sigma}(z)) + (nD + 1),$$

$$\mathrm{BIC}(\boldsymbol{y}, z; D) := -\log p(\boldsymbol{y}|z; \hat{\beta}(\boldsymbol{y}, z), \hat{\sigma}(z)) + \frac{nD+1}{2} \log \frac{n(n-1)}{2}.$$

- AIC and BIC do not guarantee their rationales.
- Minimum graph entropy (MinGE [20]) is the dimensionality selection method of Euclidean embeddings.

Appendix: Non-identifiability Problem

- Lack of one-to-one correspondence between parameter and probability distribution.

- Conventional information criteria such as Akaike's information criterion (AIC) [11], Bayesian information criterion (BIC) [12], etc… do not guarantee their rationales because their derivation depend on the central limit theorem.

$$AIC = -\log p\left(x; \hat{\theta}\right) + k,$$

$$BIC = -\log p\left(x; \hat{\theta}\right) + \frac{k}{2} \log n,$$

where k is the number of free parameters, and n is the number of data.

Appendix: Non-identifiability Problem of Hyperbolic Graph Embeddings

$\phi_i \coloneqq (r_i, \theta_i)$: embeddings, $y \in \{0,1\}$: edges.

LEMMA 1. *Assume that $r_i \neq 0$ for some $i \in [n]$. For $\alpha \in (0, 2\pi)$, we define $\phi'_i \coloneqq (r_i, \theta_i + \alpha)$ for $i \in [n]$.* Rotation.

Then, $\phi'_i \in \mathbb{H}^D_R$ and $\boldsymbol{\phi} \neq \boldsymbol{\phi}' = \{\phi'_i\}_{i \in [n]}$. Moreover, the following equation holds:

$$p(\boldsymbol{y}; \boldsymbol{\phi}, \beta) = p(\boldsymbol{y}; \boldsymbol{\phi}', \beta).$$

The same at
two different parameters !

Therefore, the probability distribution of a Poincaré embedding is non-identifiable.

$$p(\boldsymbol{y}; \boldsymbol{\phi}, \beta) = \prod_{(i,j) \in \Lambda_{[n]}} p(y_{ij}; \phi_i, \phi_j, \beta),$$

$$p(y_{ij}; \phi_i, \phi_j, \beta) = \begin{cases} \frac{1}{1+\exp(\beta(d_{\phi_i \phi_j} - R))} & (y_{ij} = 1), \\ 1 - \frac{1}{1+\exp(\beta(d_{\phi_i \phi_j} - R))} & (y_{ij} = 0). \end{cases}$$

Sigmoid function.

Appendix: Training Detail

- All embeddings were initialized uniformly at random over $[-0.001, 0.001]^D$.
- Chose the following parameters:
  - $\sigma_{max} = 1.0, \sigma_{min} = 0.001,$
  - $\beta_{max} = 10.0, \beta_{min} = 1.0,$
  - $\beta^{(0)} = 1.0, \sigma^{(0)} = 1.0,$
  - $R = \log n$.
- When making mini-batches, 10 negative samples were sampled per a positive sample.
- The learning rate was 0.1 for the first 10 epochs, and 34.375 for the remaining 790 epochs.
- The number of epochs was 800.
- The likelihood and the Fisher information were approximated as follows:

$$-\log p(\boldsymbol{y}|z;\hat{\beta}(\boldsymbol{y},z)) \approx -\frac{|\boldsymbol{y}|}{|\boldsymbol{y}'_{\mathrm{train}}|}\log p(\boldsymbol{y}'_{\mathrm{train}}|z;\hat{\beta}(\boldsymbol{y}_{\mathrm{train}},z)),$$

$$I_n(\beta) \approx \frac{2}{n'(n'-1)}\sum_{(i,j)\in\Lambda'_S}\frac{(R-d_{z_iz_j})^2\exp(-\beta(R-d_{z_iz_j}))}{(1+\exp(-\beta(R-d_{z_iz_j})))^2},$$

where $S \subset [n]$ was sampled uniformly at random over $[n]$, and $\Lambda'_S := \{(i,j)|(i,j)\in S\times S, i<j\}$.

Appendix: Selected Dimensionalities on Artificial Datasets

# of Nodes	DNML-HGG	AIC	BIC	MinGE
400	4.3±1.1	4.0±0.0	3.5±0.87	64.0±0.0
800	6.0±2.0	4.0±0.0	4.0±0.0	64.0±0.0
1600	4.3±1.1	4.0±0.0	4.0±0.0	64.0±0.0
3200	4.0±0.0	6.0±2.0	4.0±0.0	64.0±0.0
6400	16.0±0.0	7.0±1.7	4.0±0.0	64.0±0.0
12800	16.0±0.0	8.7±2.2	4.0±0.0	64.0±0.0

Correct dim. with enough nodes

Underestimate

Overestimate

Appendix: Selected Dimensionalities on Scientific Collaboration Networks

Table 5 Selected dimensionalities of each method.

Network	DNML-HGG	AIC	BIC	MinGE
AstroPh	16	16	4	64
CondMat	16	16	4	64
GrQc	16	8	4	64
HepPh	16	8	4	64

Appendix: Consiceness

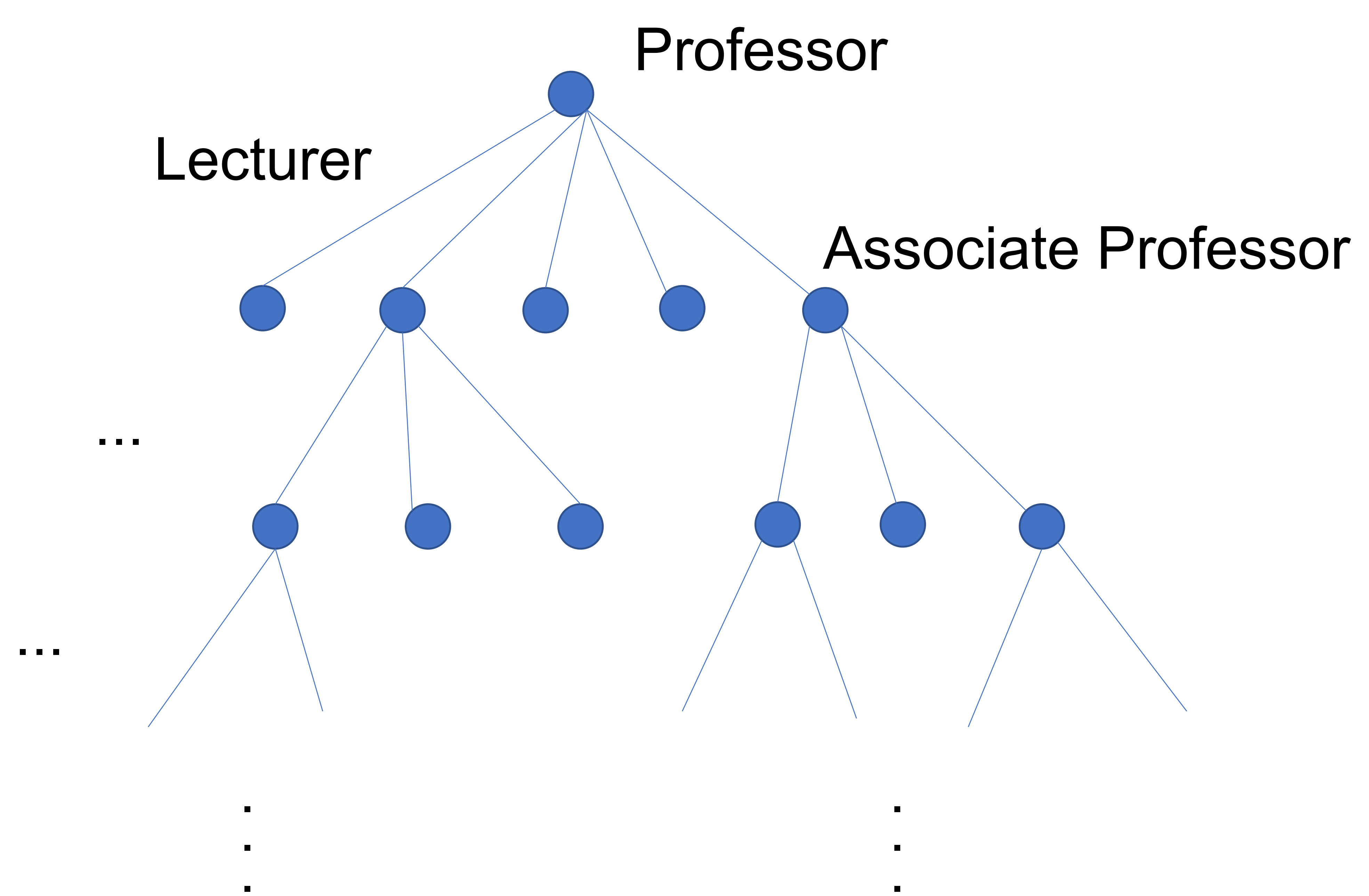
Table 6 Average conciseness of each method with $\epsilon_{\max} = 0.01$.

Network	DNML-HGG	AIC	BIC	MinGE
AstroPh	**0.501**	0.378	0.000	0.000
CondMat	**0.451**	**0.473**	0.000	0.000
GrQc	**0.944**	**0.948**	0.080	0.000
HepPh	**0.367**	0.176	0.000	0.000

Appendix: Hierarchical Structure of Scientific Collaboration Networks

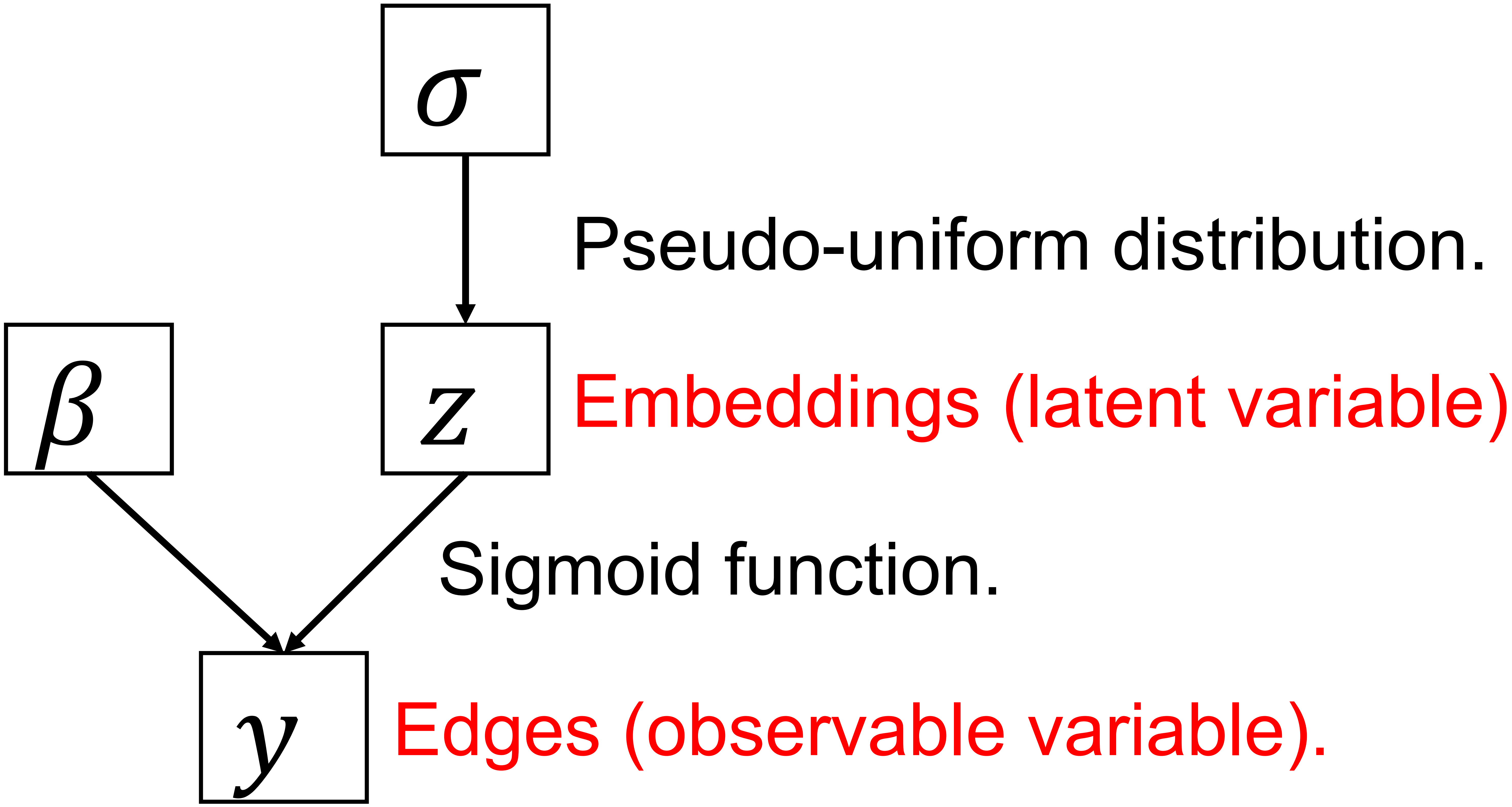
Real-world complex networks tend to have hierarchical structures.

- Ex: networks of co-authorship, paper citation, web pages, etc.



- We can approximate such structures as trees.

Appendix: Graphical Model



Appendix: Power law of degree distributions

- $P(k) \propto k^{-\gamma}$: degree distribution (k is the degree).
  - The majority of papers are cited infrequently, while a small number of papers are cited frequently.

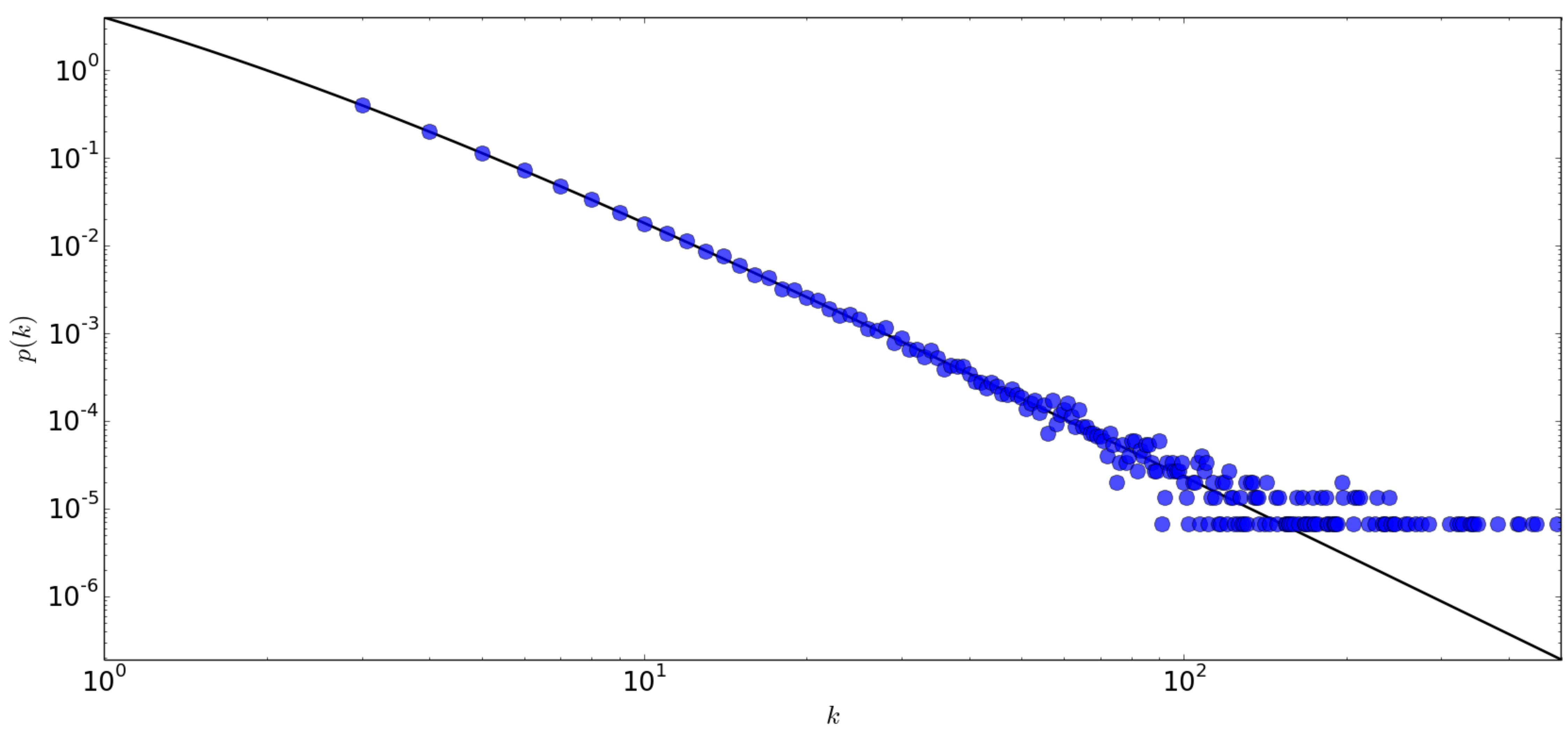
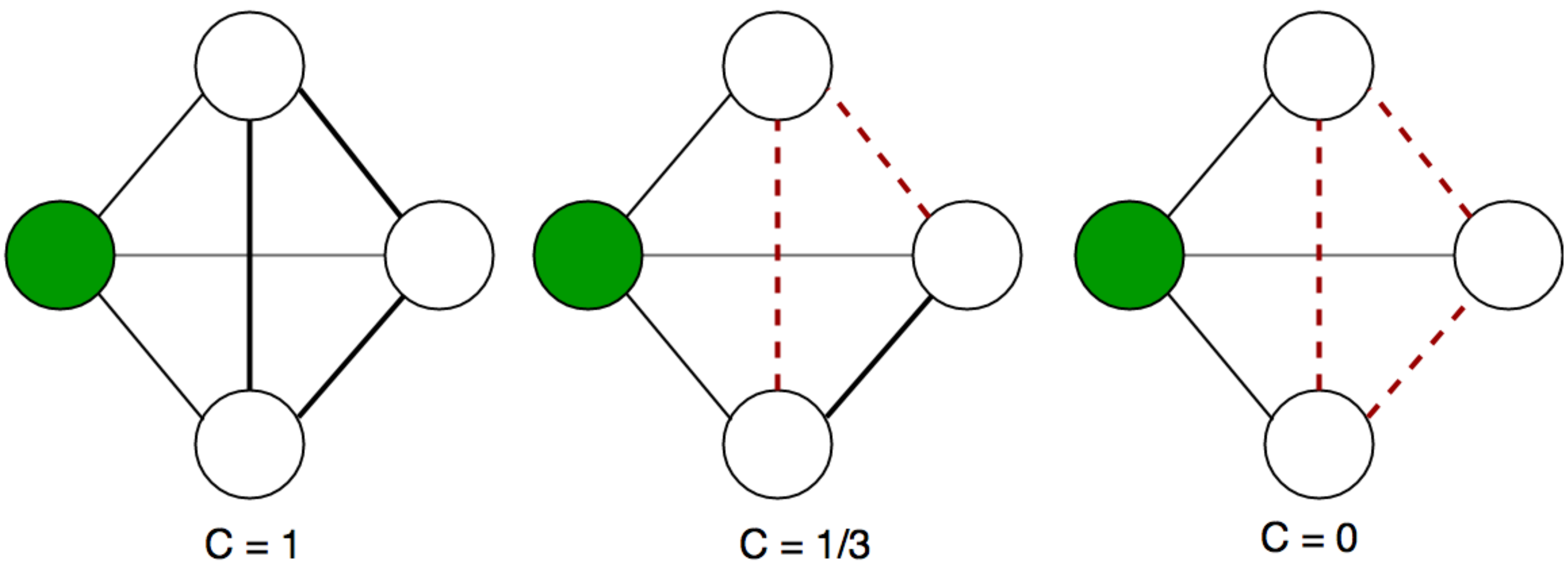


FIg2. degree distribution in a graph (https://en.wikipedia.org/wiki/Scale-free_network).

Appendix: High Clustering Coefficient

High when the graph contains many triangles.



*https://www.geeksforgeeks.org/clustering-coefficient-graph-theory/

Appendix:Ordinal MDL Principle

- Normalized maximum likelihood (NML, [10]) code length is defined as:
- $L_{NML}(x^n) = -\log p_{NML}(x^n)$
$$= \underbrace{-\log p\left(x^n; \hat{\theta}(x^n)\right)}_{} + \underbrace{\log \sum_{x'^n \in \mathcal{X}^n} p\left(x'^n; \hat{\theta}(x'^n)\right)}_{}$$

Low when the model is complex. High when the model is complex.

where $\mathcal{X}$ is the data domain and $\hat{\theta}(x)$ is the maximum likelihood estimate.