October 17, 2017
第10回 NetSci 研究会

Inter-University Research Institute Corporation / Research Organization of Information and Systems
National Institute of Informatics

PRESTO SAKIGAKE

ネットワーク構造上の統計モデルと情報幾何的な解析

杉山 麿人 Mahito Sugiyama（NII, JST さきがけ）

Summary

- Introduce a log-linear model into a DAG structure (poset)
- Analysis of statistical manifold using information geometry
- Application to matrix/tensor balancing

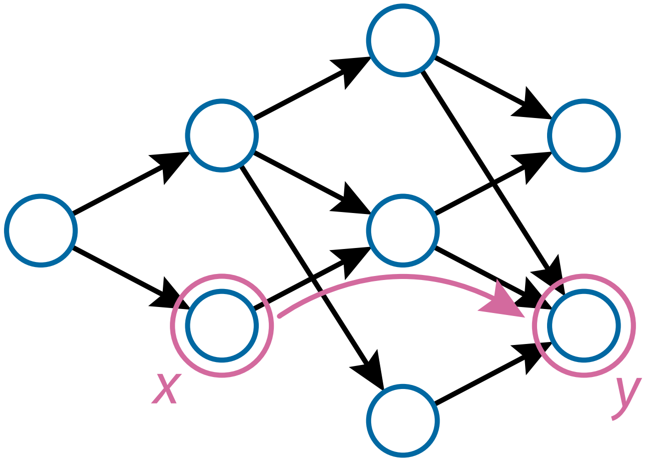
Source

- Sugiyama, M., Nakahara, H., Tsuda, K.:
  Tensor Balancing on Statistical Manifold, *ICML 2017*
  - arXiv:
    `https://arxiv.org/abs/1702.08142`
  - GitHub:
    `https://github.com/mahito-sugiyama/newton-balancing`
- Sugiyama, M., Nakahara, H., Tsuda, K.:
  Information Decomposition on Structured Space, *IEEE ISIT 2016*
  - arXiv:
    `http://arxiv.org/abs/1601.05533`

Outline

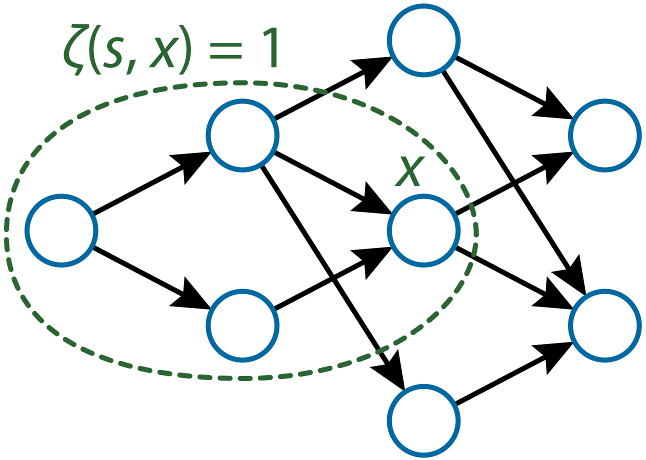
- Log-linear model on posets (networks)
- Dually flat Riemannian manifold
- Projection and its computation
- Application: Matrix balancing
- Conclusion

Partially Ordered Set



- Partially ordered set (poset) $(S, \leq)$
  (i) $x \leq x$ (reflexivity)
  (ii) $x \leq y, y \leq x \Rightarrow x = y$ (antisymmetry)
  (iii) $x \leq y, y \leq z \Rightarrow x \leq z$ (transitivity)
  - We assume that S is finite and includes the least element (bottom) $\bot \in S$
- Equivalent to a DAG
  - Each $x \in S$ is a node
  - $x \leq y \iff y$ is reachable from x

Möbius Function on Poset



- Zeta function $\zeta : S \times S \to \{0, 1\}$

$$\zeta(s, x) = \begin{cases} 1 & \text{if } s \le x, \\ 0 & \text{otherwise.} \end{cases}$$

- Möbius function $\mu : S \times S \to \mathbb{Z}$

$$\mu(x, y) = \begin{cases} 1 & \text{if } x = y, \\ -\sum_{x \le s < y} \mu(x, s) & \text{if } x < y, \\ 0 & \text{otherwise.} \end{cases}$$

  - We have $\zeta\mu = I$, i.e.,

$$\sum_{s \in S} \zeta(s, y)\mu(x, s) = \sum_{x \le s \le y} \mu(x, s) = \delta_{xy}$$

Möbius Function Is Generalization of Inclusion-Exclusion Principle

- For sets A, B, C,

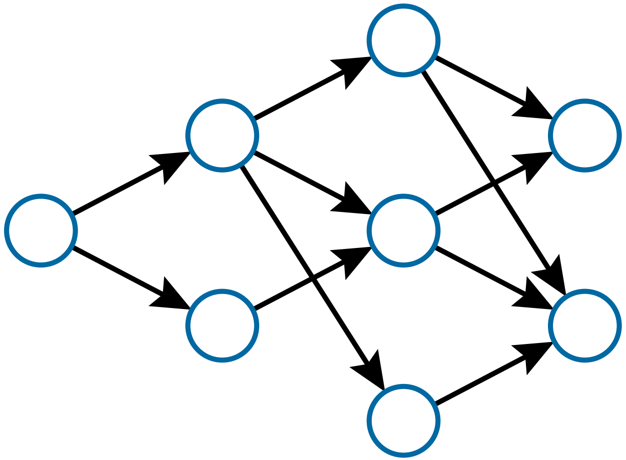
  $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$

- In general, for $A_1, A_2, \ldots, A_n$,

$$\left|\bigcup_i A_i\right| = \sum_{J \subseteq \{1,\ldots,n\}, J \neq \varnothing} (-1)^{|J|-1} \left|\bigcap_{j \in J} A_j\right|$$

- The Möbius function μ is the generalization of "$(-1)^{|J|-1}$"

Möbius Inversion



- The Möbius inversion formula [Rota (1964)]:

$$g(x) = \sum_{s \in S} \zeta(s,x) f(s) = \sum_{s \leq x} f(s)$$

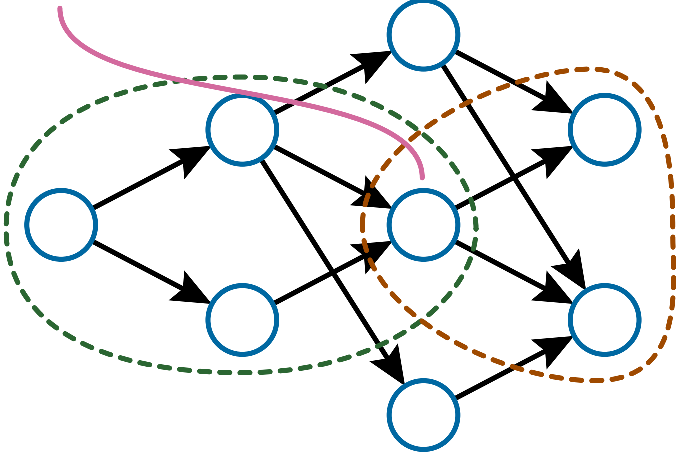
$$\Leftrightarrow f(x) = \sum_{s \in S} \mu(s,x) g(s),$$

$$h(x) = \sum_{s \in S} \zeta(x,s) f(s) = \sum_{s \geq x} f(s)$$

$$\Leftrightarrow f(x) = \sum_{s \in S} \mu(x,s) h(s)$$

Log-Linear Model on Poset (our contribution)

Each $x \in S$ has a triple:
$(p(x), \theta(x), \eta(x))$



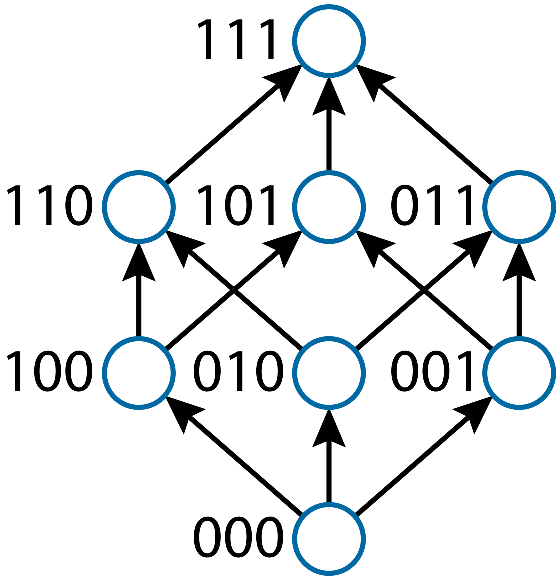
- A probability vector $p : S \to (0, 1)$ s.t. $\sum_{x \in S} p(x) = 1$
  - (Normalized) weight for each node
- We introduce $\theta : S \to \mathbb{R}$ and $\eta : S \to \mathbb{R}$ as

$$\theta(x) = \sum_{s \in S} \mu(s, x) \log p(s), \quad \eta(x) = \sum_{s \geq x} p(s)$$

- From the Möbius inversion formula:

$$\log p(x) = \sum_{s \leq x} \theta(s), \quad p(x) = \sum_{s \in S} \mu(x, s) \eta(s)$$

Our Model Includes Binary Case



- Our model:

$$\log p(x) = \sum_{s \le x} \theta(s), \quad \eta(x) = \sum_{s \ge x} p(s)$$

is generalization of the log-linear model on binary vectors with $\boldsymbol{x} \in \{0, 1\}^n = S$:

$$\log p(\boldsymbol{x}) = \sum_i \theta^i x^i + \sum_{i<j} \theta^{ij} x^i x^j + \dots$$
$$+ \theta^{1\dots n} x^1 x^2 \dots x^n - \psi,$$

$$\eta^i = \mathbf{E}[x^i] = \Pr(x^i = 1),$$
$$\eta^{ij} = \mathbf{E}[x^i x^j] = \Pr(x^i = x^j = 1), \dots$$

Outline

- Log-linear model on posets (networks)
- Dually flat Riemannian manifold
- Projection and its computation
- Application: Matrix balancing
- Conclusion

Dually Flat Structure

- Let $\psi(\theta) = -\theta(\bot) = -\log p(\bot)$

- The Legendre dual $\varphi(\eta)$ is obtained by the Legendre transformation:

  $$\varphi(\eta) = \max_{\theta'}\left(\theta'\eta - \psi(\theta')\right), \quad \theta'\eta = \sum\nolimits_{x \in S \setminus \{\bot\}} \theta'(x)\eta(x)$$

  - $\psi(\theta)$ and $\varphi(\eta)$ should be convex

- We can prove that its Legendre dual is the negative entropy:
  $\varphi(\eta) = \sum_{x \in S} p(x) \log p(x)$

- θ and η form a dual coordinate system: $\nabla\psi(\theta) = \eta, \ \nabla\varphi(\eta) = \theta$

Riemannian Manifold

- Let $\mathcal{S}$ be the set of probability vectors
- The manifold $(\mathcal{S}, g(\xi))$ is a Riemannian manifold with the Riemannian metric $g(\xi)$ s.t. $\forall x, y \in S \setminus \{\perp\}$

$$g_{xy}(\xi) = \begin{cases} \sum_{s \in S} \zeta(x,s)\zeta(y,s)p(s) - \eta(x)\eta(y) & \text{if } \xi = \theta, \\ \sum_{s \in S} \mu(s,x)\mu(s,y)p(s)^{-1} & \text{if } \xi = \eta \end{cases}$$

  - $g(\theta) = \nabla\nabla\psi(\theta)$, $g(\eta) = \nabla\nabla\varphi(\eta)$

Fisher Information Matrix and Orthogonality

- Since $g(\xi)$ coincides with the Fisher information matrix,

$$\mathbf{E}\left[\frac{\partial}{\partial\theta(x)}\log p(s)\frac{\partial}{\partial\theta(y)}\log p(s)\right] = g_{xy}(\theta),$$

$$\mathbf{E}\left[\frac{\partial}{\partial\eta(x)}\log p(s)\frac{\partial}{\partial\eta(y)}\log p(s)\right] = g_{xy}(\eta)$$

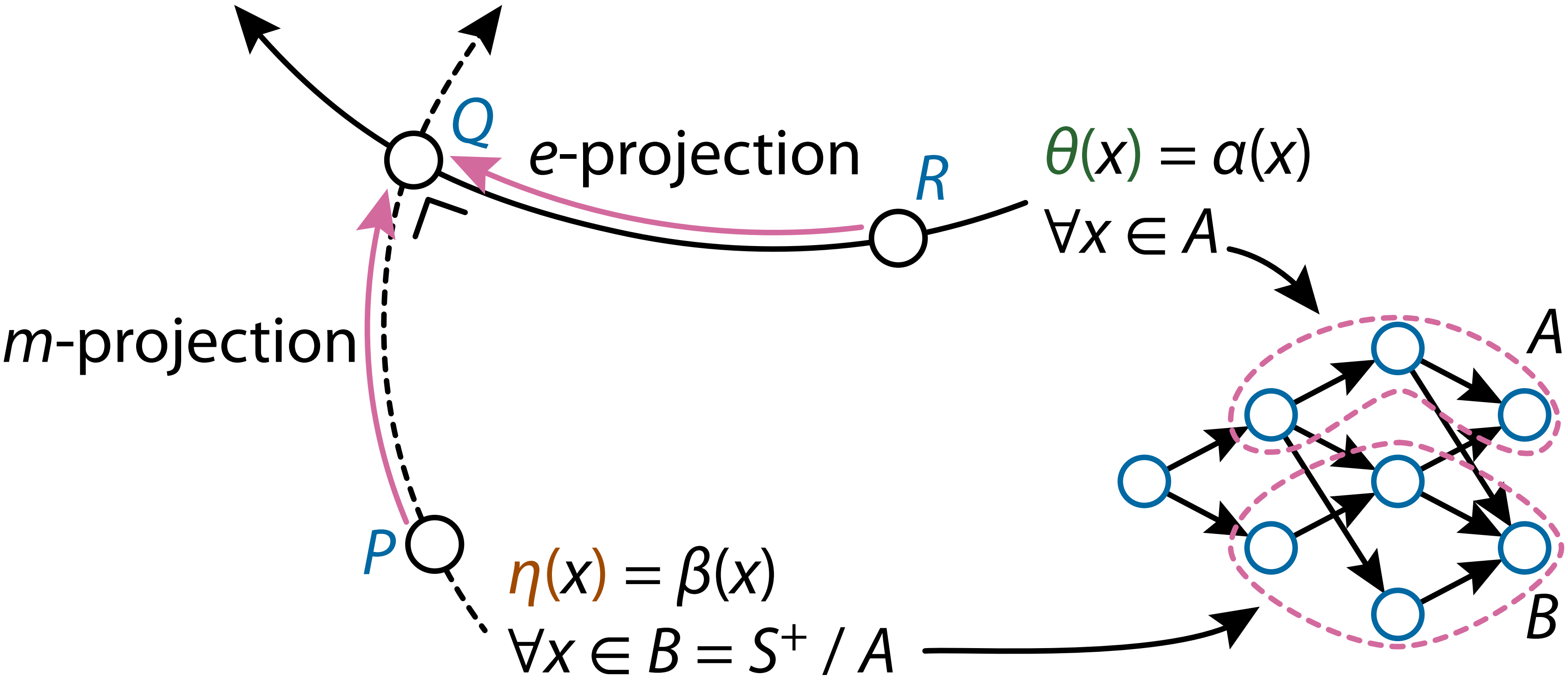
- θ and η are orthogonal, i.e.,

$$\mathbf{E}\left[\frac{\partial}{\partial\theta(x)}\log p(s)\frac{\partial}{\partial\eta(y)}\log p(s)\right] = \sum_{s\in S}\zeta(x,s)\mu(s,y) = \delta_{xy}$$

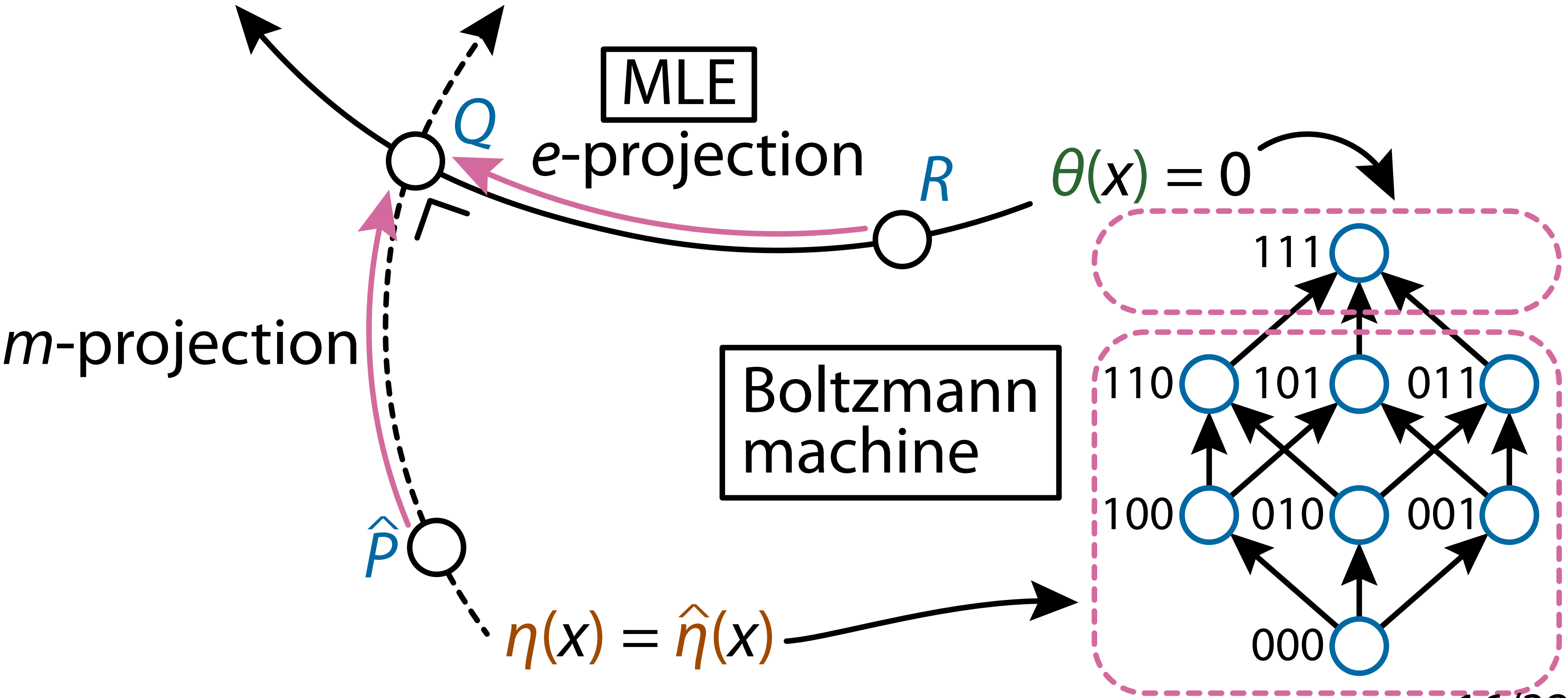
Outline

- Log-linear model on posets (networks)
- Dually flat Riemannian manifold
- Projection and its computation
- Application: Matrix balancing
- Conclusion

e-Projection and *m*-Projection



e-Projection and *m*-Projection



Computation of *e*-Projection

- Given P and β, we compute P_β such that

$$\begin{cases} \theta_{P_\beta}(x) = \theta_P(x) & \text{if } x \in (S \setminus \{\bot\}) \setminus \mathrm{dom}(\beta), \\ \eta_{P_\beta}(x) = \beta(x) & \text{if } x \in \mathrm{dom}(\beta) \end{cases}$$

- Initialize with $P_\beta^{(0)} = P$ and, at each step t, update $\eta_{P_\beta}^{(t)}(x)$ for $x \in \mathrm{dom}(\beta)$
  - Since θ and η are orthogonal, we can change $\eta_{P_\beta}^{(t)}(x)$ while fixing $\theta_{P_\beta}^{(t)}(y)$ for $y \notin \mathrm{dom}(\beta)$

Gradient

- We can use Newton's method as we can compute the derivatives $\partial\theta^{(t)}(x)/\partial\eta^{(t)}(y)$ and $\partial\eta^{(t)}(x)/\partial\theta^{(t)}(y)$, thanks to the Möbius inversion

- Gradient of θ and η is obtained as the Riemannian metric: $g(\theta) = \nabla\nabla\psi(\theta) = \nabla\eta$ and $g(\eta) = \nabla\nabla\varphi(\eta) = \nabla\theta$
$$\frac{\partial\eta(x)}{\partial\theta(y)} = \sum_{s\in S} \zeta(x,s)\zeta(y,s)p(s) - \eta(x)\eta(y),$$
$$\frac{\partial\theta(x)}{\partial\eta(y)} = \sum_{s\in S} \mu(s,x)\mu(s,y)p(s)^{-1}$$

Newton's Method (1/2)

- Each step of Newton's method:

$$\begin{bmatrix} \vdots \\ \eta_{P_\beta}^{(t)}(x) - \beta(x) \\ \vdots \\ \vdots \end{bmatrix} + J \begin{bmatrix} \vdots \\ \vdots \\ \theta_{P_\beta}^{(t+1)}(y) - \theta_{P_\beta}^{(t)}(y) \\ \vdots \end{bmatrix} = \mathbf{0},$$

  – J is the $|\mathrm{dom}(\beta)| \times |\mathrm{dom}(\beta)|$ Jacobian matrix given as

$$J_{xy} = \frac{\partial \eta_{P_\beta}^{(t)}(x)}{\partial \theta_{P_\beta}^{(t)}(y)} = \sum_{s \in S} \zeta(x, s)\zeta(y, s) p_\beta{}^{(t)}(s) - \eta_{P_\beta}^{(t)}(x)\eta_{P_\beta}^{(t)}(y)$$

for each $x, y \in \mathrm{dom}(\beta)$

Newton's Method (2/2)

- Each update is

$$\begin{bmatrix} \vdots \\ \theta_{P_\beta}^{(t+1)}(x) \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \theta_{P_\beta}^{(t)}(x) \\ \vdots \\ \vdots \end{bmatrix} - J^{-1} \begin{bmatrix} \vdots \\ \vdots \\ \eta_{P_\beta}^{(t)}(y) - \beta(y) \\ \vdots \end{bmatrix}$$

  - J^{-1} is the inverse of J
  - J is the $|\mathrm{dom}(\beta)| \times |\mathrm{dom}(\beta)|$ Jacobian matrix given as

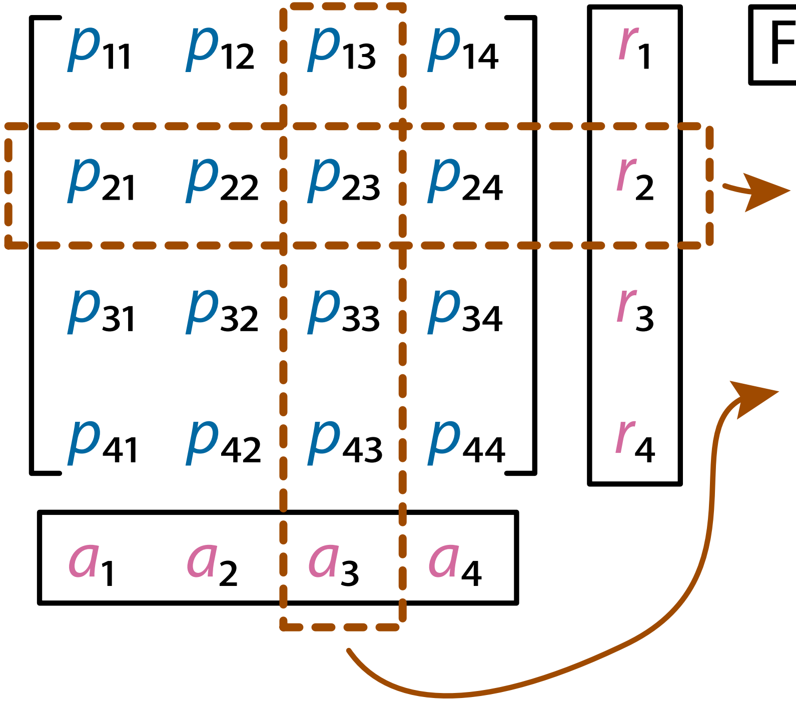
$$J_{xy} = \frac{\partial \eta_{P_\beta}^{(t)}(x)}{\partial \theta_{P_\beta}^{(t)}(y)} = \sum_{s \in S} \zeta(x,s)\zeta(y,s)p_\beta{}^{(t)}(s) - \eta_{P_\beta}^{(t)}(x)\eta_{P_\beta}^{(t)}(y)$$

for each $x, y \in \mathrm{dom}(\beta)$

Outline

- Log-linear model on posets (networks)
- Dually flat Riemannian manifold
- Projection and its computation
- Application: Matrix balancing
- Conclusion

Matrix Balancing



Find *r* and *a*

$$r_2a_1p_{21} + r_2a_2p_{22} + r_2a_3p_{23} + r_2a_4p_{24} = 1$$

$$r_1a_3p_{13} + r_2a_3p_{23} + r_3a_3p_{33} + r_4a_3p_{43} = 1$$

Matrix Balancing

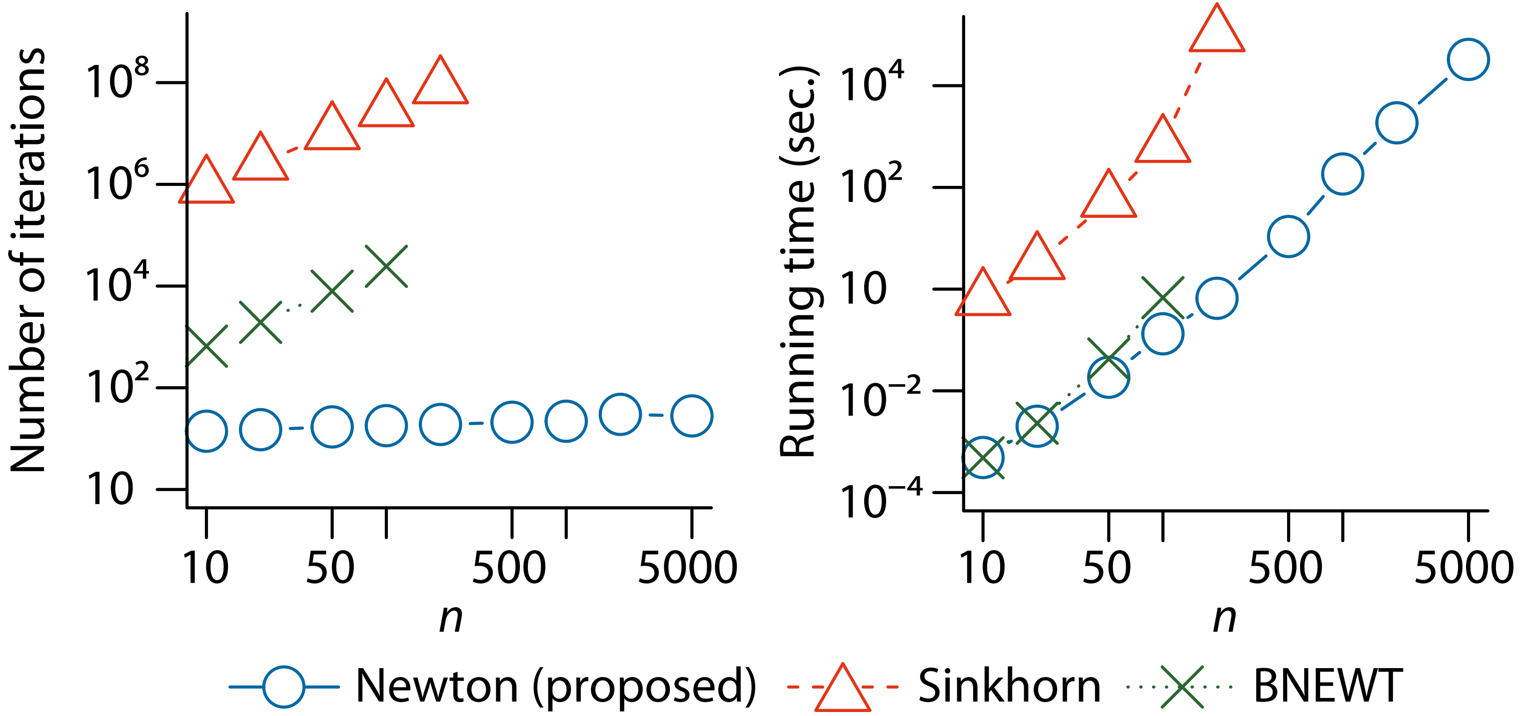
- Problem setting:
  Given a nonnegative matrix $P = (p_{ij}) \in \mathbb{R}_+^{n \times n}$, find $\boldsymbol{r}, \boldsymbol{s} \in \mathbb{R}^n$ s.t.

  $$(RPS)\mathbf{1} = \mathbf{1} \quad \text{and} \quad (RPS)^T\mathbf{1} = \mathbf{1}$$

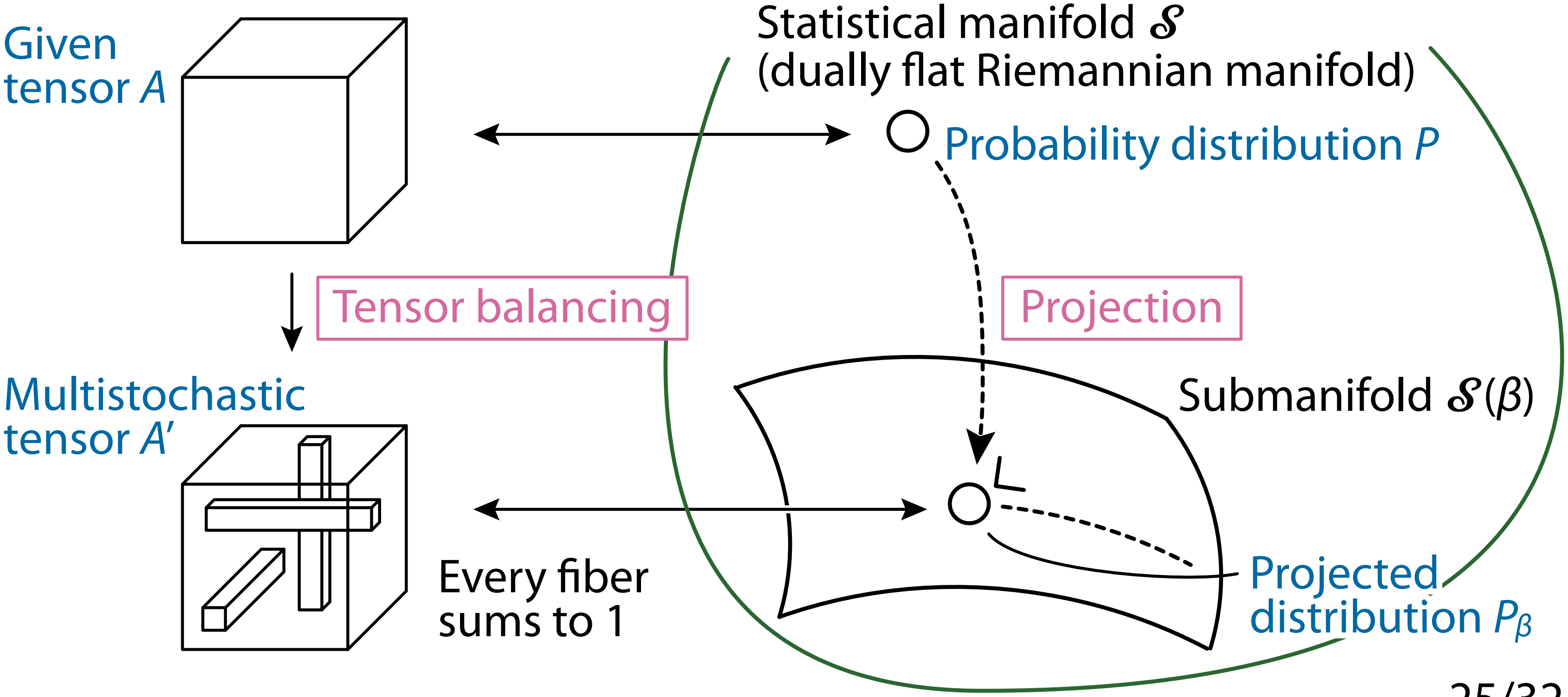
  - $R = \mathrm{diag}(\boldsymbol{r})$, $S = \mathrm{diag}(\boldsymbol{s})$
  - Each entry is given as $p'_{ij} = p_{ij} r_i s_j$

- A fundamental process to analyze and compare matrices in a wide range of applications
  - Input-output analysis in economics, seat assignments in elections, Hi-C data analysis, Sudoku puzzle
  - Approximate Wasserstein distance

Results on Hessenberg Matrix

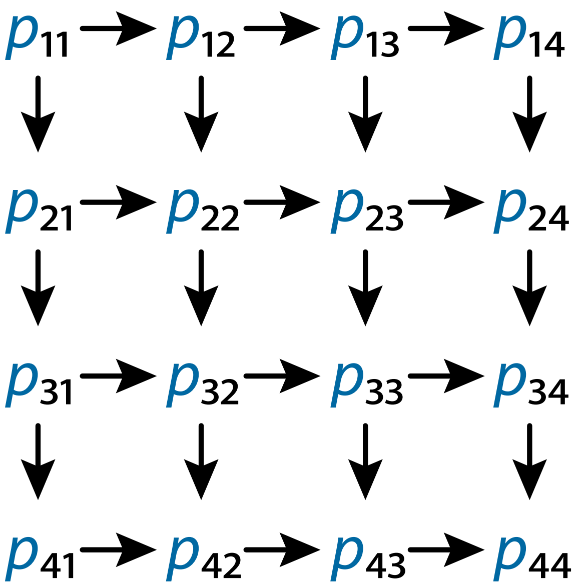


Overview of Our Approach



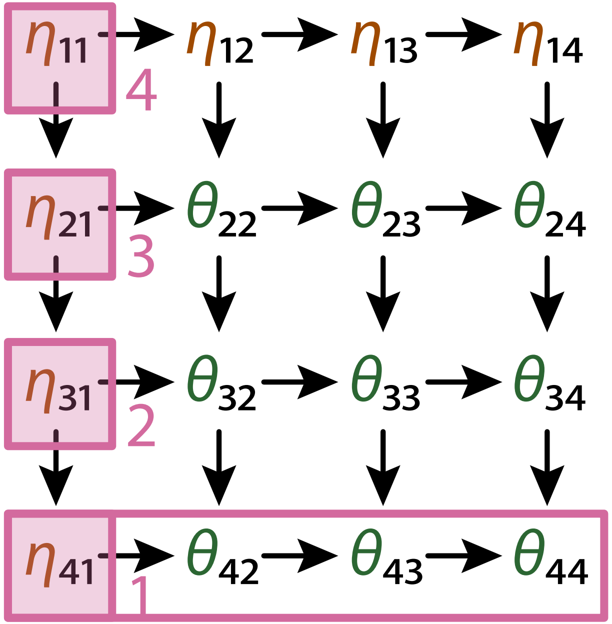
View Matrix as Poset

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$$



Introduce θ and η

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$$



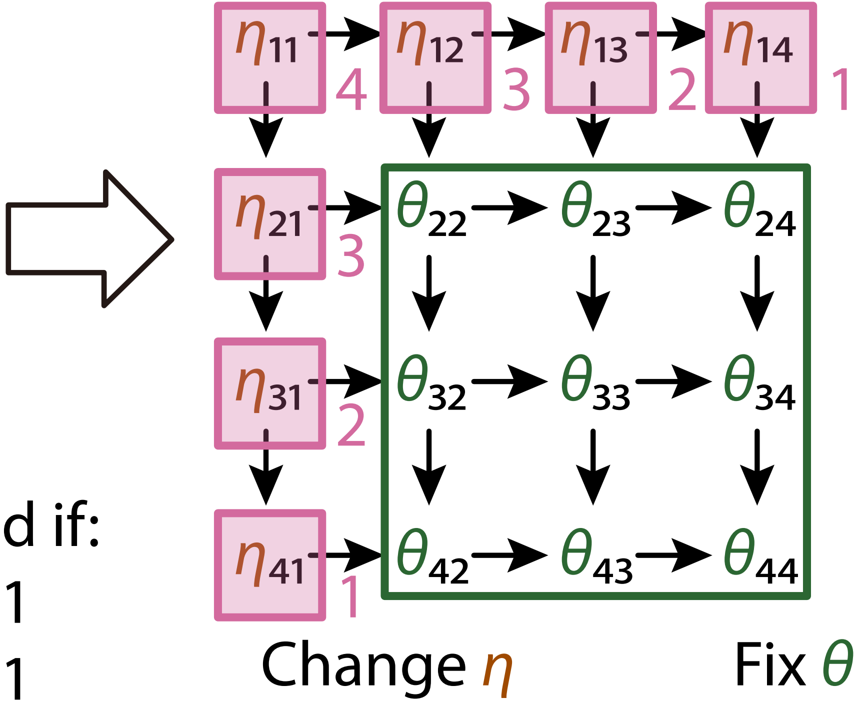
Matrix balancing is achieved if:

$\eta_{11} = 4, \eta_{21} = 3, \eta_{31} = 2, \eta_{41} = 1$

$\eta_{11} = 4, \eta_{12} = 3, \eta_{13} = 2, \eta_{14} = 1$

e-Projection = Balancing

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$$

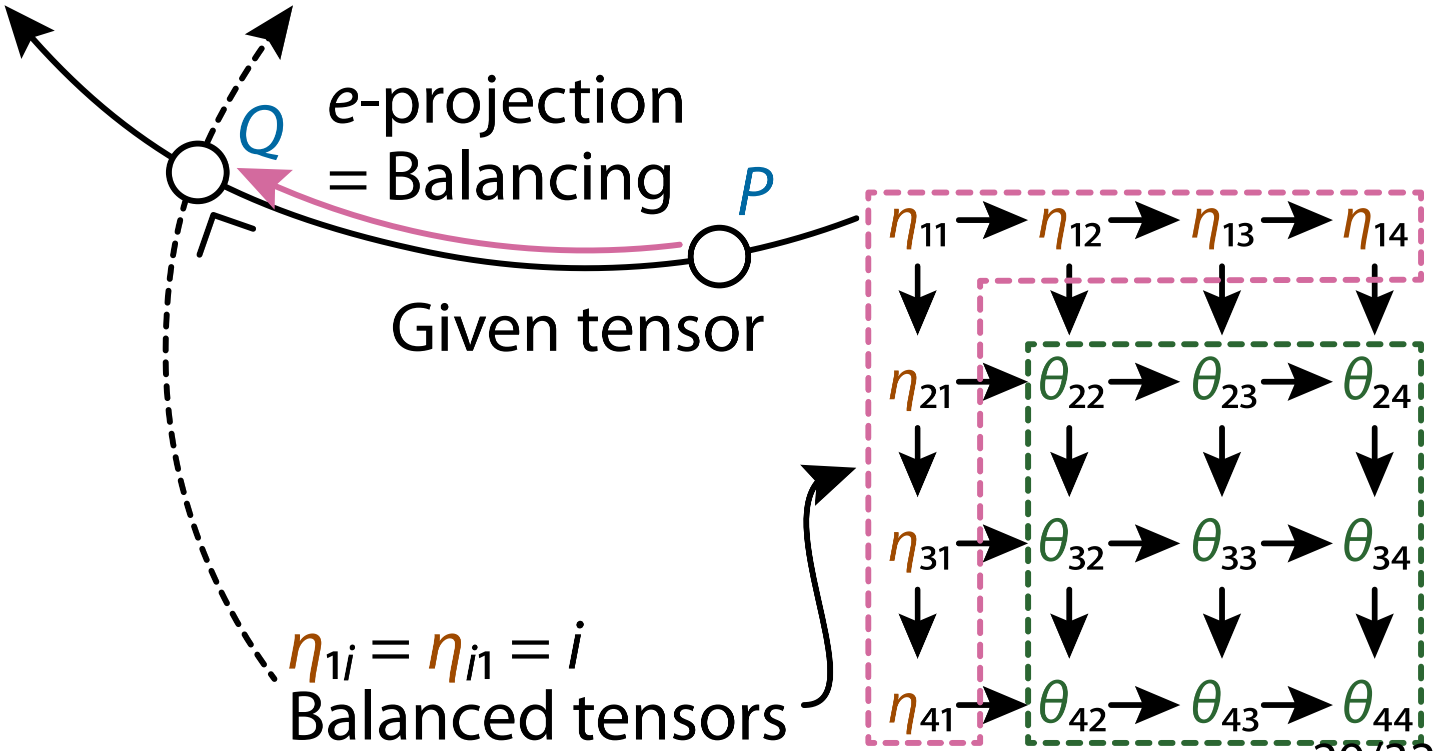


Matrix balancing is achieved if:

$\eta_{11} = 4, \eta_{21} = 3, \eta_{31} = 2, \eta_{41} = 1$

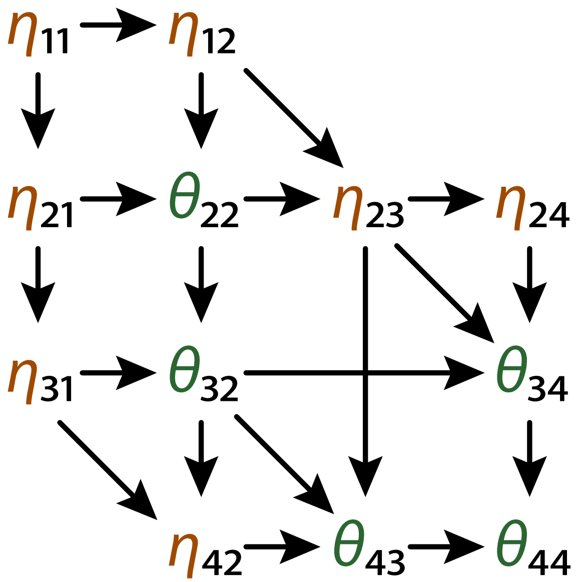
$\eta_{11} = 4, \eta_{12} = 3, \eta_{13} = 2, \eta_{14} = 1$

e-Projection = Balancing



Remove Zeros If Exists

$$\begin{bmatrix} p_{11} & p_{12} & 0 & 0 \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & 0 & p_{34} \\ 0 & p_{42} & p_{43} & p_{44} \end{bmatrix}$$



Matrix balancing is achieved if:

$\eta_{11} = 4, \eta_{21} = 3, \eta_{31} = 2, \eta_{41} = 1$

$\eta_{11} = 4, \eta_{12} = 3, \eta_{13} = 2, \eta_{14} = 1$

Outline

- Log-linear model on posets (networks)
- Dually flat Riemannian manifold
- Projection and its computation
- Application: Matrix balancing
- Conclusion

Conclusion

- The dually flat structure of the set of probability distributions (statistical manifold) of partially ordered outcome space
- We have achieved efficient matrix/tensor balancing with Newton's method
- **Discrete (network) structure + Information Geometry = original and significant data analysis methods!**