Aug. 28, 2026
02901 Advanced Topics in Machine Learning:
Tensor Networks for Machine Learning

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

Information-Geometric Modeling for Tensors

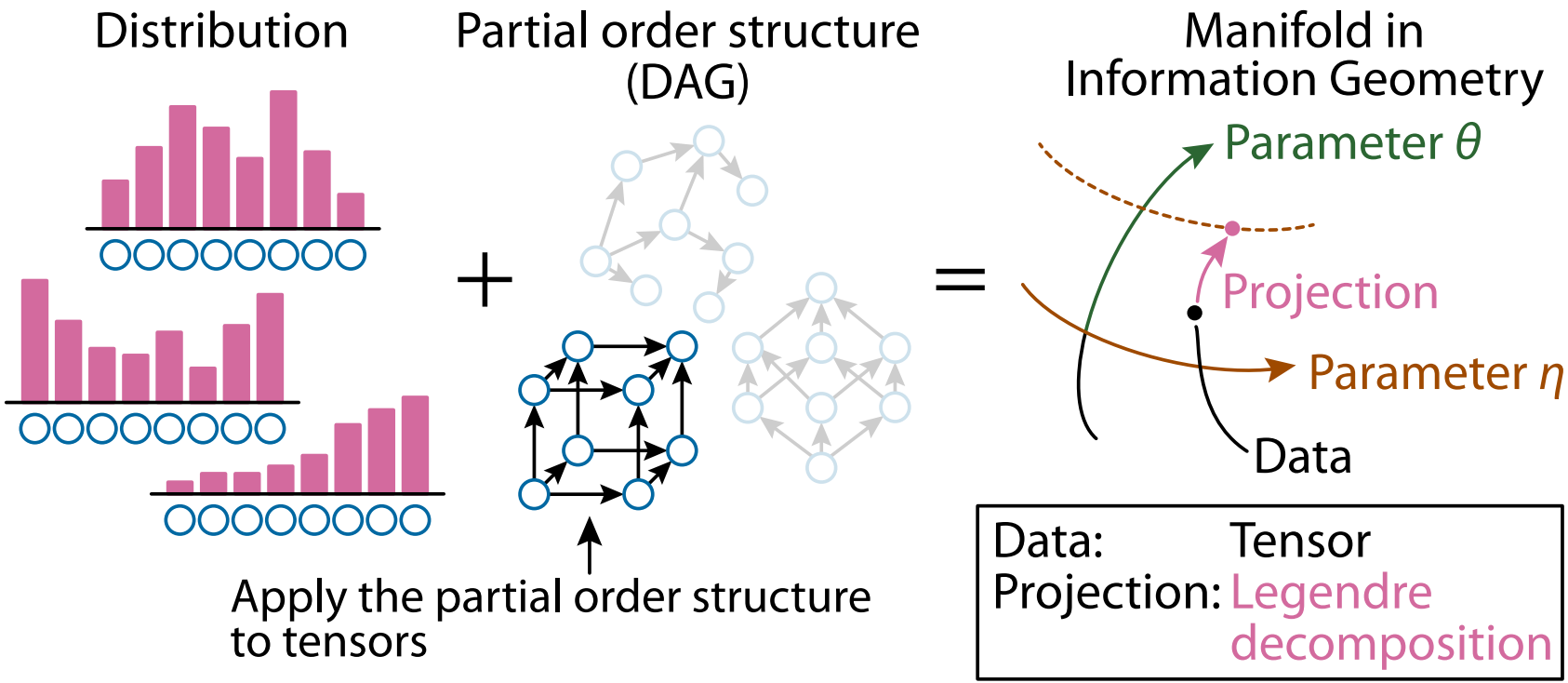
Mahito Sugiyama (NII) https://mahito.nii.ac.jp/

Self Introduction

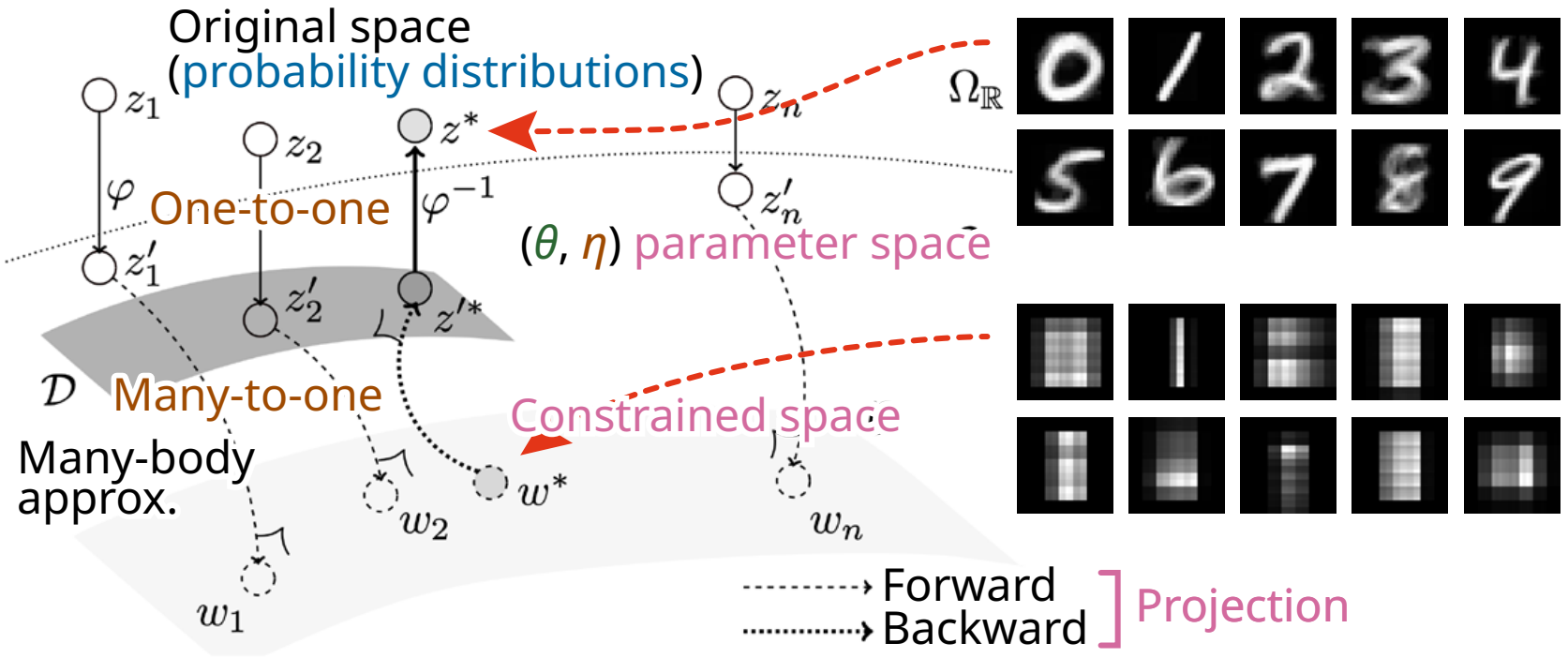
- Studying basic methodologies (theory and algorithms) of machine learning and data mining
  - We often focus on discrete structure such as graphs
- Ph.D. in Informatics at Kyoto Univ. (2012)
  → Postdoc of Karsten's group at MPI Tübingen (2012–2014)
  → Assist. Prof. at Osaka Univ. (2014–2016)
  → Assoc. Prof. at National Institute of Informatics (2017–now)
- Sugiyama lab. at NII (https://mahito.nii.ac.jp/):
  - One admin. staff
  - Two postdocs, eight Ph.D. students
  - (with regular internship students)

National Institute of Informatics (Tokyo)

Overview of IG-Based Approaches



Data Augmentation [Hu & Sugiyama, 2026]



Formulation as ML Problem

1. **Data:** Points (probability distributions, or tensors) on a statistical manifold
2. **Model:** Parameterization via (θ, η)-coordinates
   - The log-linear model on posets [Sugiyama et al., 2017]
     - Explicitly model any-order interactions between variables
     - Generalization of **higher-order Boltzmann machines** [Sejnowski,1986] and its **information geometry** [Amari, 2001]
3. **Learning:** (Forward) Projection onto constrained space
   - We use many-body approximation [Ghalamkari et al., 2023]
4. **Correctness:** Convex optimization

Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Example: Modeling of Binary Data

Dataset

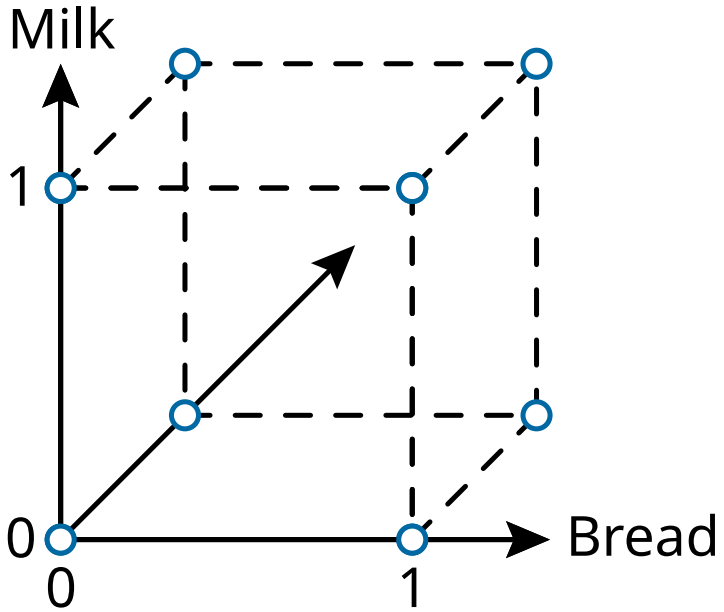
	Bread	Milk	Apple
ID 1	1	0	0
ID 2	1	1	0
ID 3	1	0	0
ID 4	0	1	1
ID 5	0	1	1
ID 6	1	1	0
ID 7	1	0	0
ID 8	1	1	0

Example: Modeling of Binary Data

Dataset

	Bread	Milk	Apple
ID 1	1	0	0
ID 2	1	1	0
ID 3	1	0	0
ID 4	0	1	1
ID 5	0	1	1
ID 6	1	1	0
ID 7	1	0	0
ID 8	1	1	0

variables
= # modes

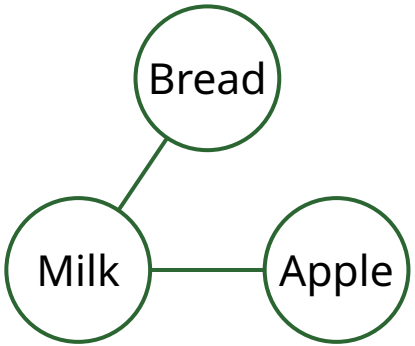


Example: Boltzmann Machines

log(prob.) = –10.41 + 9.43[Bread] + 8.52[Milk] – 9.84[Apple]
– 9.03[Bread&Milk] + 9.43[Milk&Apple]

MLE

Boltzmann machine



Bread	Milk	Apple	Probability from data	Learned probability
0	0	0	?	0.0000300109
1	0	0	0.375	0.3749599867
0	1	0	?	0.1499903954
0	0	1	?	0.0000000016
1	1	0	0.375	0.2250096042
1	0	1	?	0.0000200043
0	1	1	0.25	0.0999895960
1	1	1	?	0.1500004008

Probability Computation in BMs

- Boltzmann Machine: Undirected graph $G = (V, E)$
  - $V = \{\text{Bread}, \text{Milk}, \text{Apple}\}$
  - $E = \{\{\text{Bread}, \text{Milk}\}, \{\text{Milk}, \text{Apple}\}\}$
- Parameter: $\boldsymbol{\theta} = (\theta_{\text{Bread}}, \theta_{\text{Milk}}, \theta_{\text{Apple}}, \theta_{\text{Bread\&Milk}}, \theta_{\text{Milk\&Apple}})$
- Probability of model distribution:

$$p(\text{Bread}; \boldsymbol{\theta}) = \exp(\theta_{\text{Bread}}) / Z$$

$$p(\text{Bread\&Milk}; \boldsymbol{\theta}) = \exp(\theta_{\text{Bread}} + \theta_{\text{Milk}} + \theta_{\text{Bread\&Milk}}) / Z$$

$$Z = \exp(-\theta_{\emptyset}) \quad \text{(normalizing constant)}$$

Learning of Parameter θ: MLE

- For a dataset $X = \{\boldsymbol{x}_{\mathrm{ID1}}, \boldsymbol{x}_{\mathrm{ID2}}, \dots, \boldsymbol{x}_{\mathrm{ID8}}\}$, $\boldsymbol{x}_{\mathrm{ID}i} \in \{0,1\}^3$,
  find a parameter vector θ that maximizes likelihood L

  $L_X(\theta) = p(\boldsymbol{x}_{\mathrm{ID1}}; \theta) \cdot p(\boldsymbol{x}_{\mathrm{ID2}}; \theta) \cdot \dots \cdot p(\boldsymbol{x}_{\mathrm{ID8}}; \theta)$

  - This is **MLE** (Maximum Likelihood Estimation)
- Basic strategy is the gradient method:

  (i) Start with some θ

  (ii) Compute the direction (gradient) to the goal

  - Difference between
    frequencies of empirical and those of model distributions

  (iii) Move θ a bit toward the direction $\rightarrow$ (ii)

Formal Definition of Boltzmann Machines

- A Boltzmann machine (BM) is represented as an undirected graph $G = (V, E)$, with $V = \{1, 2, \dots, n\}$ and $E \subseteq \{\{i, j\} \mid i, j \in V\}$
  - Use indices $(1, 2, \dots, n)$ instead of names (Bread, Milk, Apple, ...)
- The energy function $\Phi{:}\{0,1\}^n \to \mathbb{R}$ of a BM G is defined as

$$\Phi(\boldsymbol{x}; \boldsymbol{\theta}) = -\sum_{i \in V} \theta_i x_i - \sum_{\{i,j\} \in E} \theta_{ij} x_i x_j$$

  - $\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \{0,1\}^n$
  - $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n, \theta_{12}, \theta_{13}, \dots, \theta_{(n-1)n})$ is a parameter vector for vertices (bias) $\theta_1, \dots, \theta_n$ and edges (weight) $\theta_{12}, \dots, \theta_{(n-1)n}$
  - $\theta_{ij} = 0$ if $\{i, j\} \notin E$

Gibbs Distribution

- Probability $p(\boldsymbol{x}; \boldsymbol{\theta})$ is obtained for each $\boldsymbol{x} \in \{0, 1\}^n$ as

  $$p(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{\exp(-\Phi(\boldsymbol{x}, \boldsymbol{\theta}))}{\psi(\boldsymbol{\theta})}$$

- $\psi(\boldsymbol{\theta})$ is a partition function such that

  $$\psi(\boldsymbol{\theta}) = \sum_{\boldsymbol{x} \in \{0,1\}^n} \exp(-\Phi(\boldsymbol{x}; \boldsymbol{\theta}))$$

  to ensure the condition $\sum_{\boldsymbol{x} \in \{0,1\}^n} p(\boldsymbol{x}; \boldsymbol{\theta}) = 1$

- The distribution P composed of $p(\boldsymbol{x}; \boldsymbol{\theta})$ is called Gibbs distribution

Learning of BM by MLE

- Given a dataset $D = \{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, \dots, \boldsymbol{x}_{(N)}\}$, the objective of "learning" Boltzmann machines is to maximize the (log-)likelihood (MLE):

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{k=1}^{N} p(\boldsymbol{x}_{(k)}; \boldsymbol{\theta}) \quad \left(= p(\boldsymbol{x}_{(1)}; \boldsymbol{\theta}) \cdot p(\boldsymbol{x}_{(2)}; \boldsymbol{\theta}) \cdot \dots \cdot p(\boldsymbol{x}_{(N)}; \boldsymbol{\theta}) \right)$$

  - The probability of generating the given dataset by a BM

- The log-likelihood is usually used (better for numerical stability):

$$L_D(\boldsymbol{\theta}) = \log \prod_{k=1}^{N} p(\boldsymbol{x}_{(k)}; \boldsymbol{\theta}) = \sum_{k=1}^{N} \log p(\boldsymbol{x}_{(k)}; \boldsymbol{\theta})$$

Gradient of θ

- Let us introduce the expectation as

$$\begin{cases} \eta_i := \mathbf{E}_\theta[x_i] = \Pr(x_i = 1) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}; \theta)\mathbf{1}[x_i = 1] \\ \eta_{ij} := \mathbf{E}_\theta[x_i x_j] = \Pr(x_i = x_j = 1) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}; \theta)\mathbf{1}[x_i = x_j = 1] \end{cases}$$

- The gradient of $L_D(\theta)$ w.r.t. θ_i and θ_{ij} is obtained as

$$\frac{\partial L_D(\theta)}{\partial \theta_i} = \frac{1}{N}|\{\boldsymbol{x}_{(k)} \in D \mid x_{(k)i} = 1\}| - \eta_i,$$

$$\frac{\partial L_D(\theta)}{\partial \theta_{ij}} = \underbrace{\frac{1}{N}|\{\boldsymbol{x}_{(k)} \in D \mid x_{(k)i} = x_{(k)j} = 1\}|}_{\text{data}} - \underbrace{\eta_{ij}}_{\text{model}}$$

Learning Equation of BM

- $L_D(\boldsymbol{\theta})$ is maximized when the gradient is zero
- The gradient is zero $\iff$ For all i and ij, we have

$$\begin{cases} \frac{1}{N}|\{\boldsymbol{x}_{(k)} \in D \mid x_{(k)i} = 1\}| = \eta_i \\ \frac{1}{N}|\{\boldsymbol{x}_{(k)} \in D \mid x_{(k)i} = x_{(k)j} = 1\}| = \eta_{ij} \end{cases}$$

- This is known as the learning equation of BM

Empirical Distribution

- For the empirical distribution $\hat{P}$ of $D = \{x_{(1)}, \dots, x_{(N)}\}$ with

  $$\hat{p}(\boldsymbol{x}) = \frac{1}{N} |\{\boldsymbol{x}_{(k)} \in D \mid \boldsymbol{x}_{(k)} = \boldsymbol{x}\}|,$$

  let us introduce the frequency as

  $$\hat{\eta}_i = \sum_{\boldsymbol{x}} \hat{p}(\boldsymbol{x}) \mathbf{1}[x_i = 1]$$

  $$\hat{\eta}_{ij} = \sum_{\boldsymbol{x}} \hat{p}(\boldsymbol{x}) \mathbf{1}[x_i = x_j = 1]$$

- The learning equation becomes

  $\hat{\eta}_i = \eta_i$ and $\hat{\eta}_{ij} = \eta_{ij}$

KL Divergence Minimization

- Given two distributions P, Q, the Kullback–Leibler (KL) divergence from P to Q is defined as:

  $$D_{\mathrm{KL}}(P, Q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

- Maximizing the (log)likelihood $L_D(\boldsymbol{\theta})$ is equivalent to minimizing the KL divergence:

  $$\min_{P \in \mathcal{S}} D_{\mathrm{KL}}(\hat{P}, P)$$

  - $\mathcal{S}$: the set of Gibbs distributions

- This optimization is guaranteed to be convex
  - Global optimum is guaranteed in the gradient method

Optimization: Gradient Descent

Algorithm 1: Learning of BM by gradient descent

Initialize $\boldsymbol{\theta}$ with some values; $t \leftarrow 0$;
repeat
  foreach $i \in V$ **do**
    $\theta_i^{(t+1)} \leftarrow \theta_i^{(t)} + \varepsilon\left(\hat{\eta}_i - \eta_i\right)$;
  foreach $\{i, j\} \in E$ **do**
    $\theta_{ij}^{(t+1)} \leftarrow \theta_{ij}^{(t)} + \varepsilon\left(\hat{\eta}_{ij} - \eta_{ij}\right)$;
  $t \leftarrow t + 1$
until $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t+1)}$;

Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Information Geometry [Amari, 2016]

- Information geometry considers the space of distributions
  - This space is often called a statistical manifold
  - Each point on the manifold represents a probability distribution
- By analyzing the structure and properties of the space, IG provides a geometric view of probability distributions
- IG is a powerful framework for understanding and analyzing probabilistic models
  - Many probabilistic operations can be interpreted as geometric operations on the manifold
  - For example, “learning” can be interpreted as a projection of the empirical distribution onto a submanifold specified by a model

Exponential Family

- The exponential family is a class of probability distributions that can be expressed in the form:

  $$p(x, \boldsymbol{\theta}) = \exp\left(\sum \theta_i k_i(x) + r(x) - \psi(\boldsymbol{\theta})\right)$$

  - x is a random variable and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)$ is a natural parameter, which determines the distribution
  - If we use a vector $\boldsymbol{x} = (x_1, \dots, x_n)$ with $x_i = k_i(x)$, it becomes

    $$p(\boldsymbol{x}, \boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{x} - \psi(\boldsymbol{\theta})\right)$$

- Many distributions are in the exponential family, including: Gamma, multinomial, beta, log-exponential, Poisson, and (multivariate) Gaussian distributions

Dual Coordinates in Exponential Family

- The function $\psi(\boldsymbol{\theta})$ is a convex function of $\boldsymbol{\theta}$
  - This is the normalizing constant, partition function, cumulant generating function in statistics, and free energy in physics
- The dually flat structure, the central concept in IG, is introduced in the set of distributions $\boldsymbol{S} = \{p(\boldsymbol{x}, \boldsymbol{\theta})\}$ using $\psi(\boldsymbol{\theta})$
- An affine coordinate system for $\boldsymbol{S}$ is the natural parameter $\boldsymbol{\theta}$
- Its dual coordinate system is introduce as

  $\boldsymbol{\eta} = \nabla\psi(\boldsymbol{\theta}), \quad \boldsymbol{\eta} = (\eta_1, \dots, \eta_n)$

  which coincides with the expectation of $\boldsymbol{x}$ and known as an expectation parameter in statistics

Gaussian Distribution (1/2)

- The Gaussian distribution is given as

$$p(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- Let us introduce $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$ as

$$x_1 = k_1(x) = x, \quad x_2 = k_2(x) = x^2$$

- The natural parameter $\boldsymbol{\theta} = (\theta_1, \theta_2)$ is given as

$$\theta_1 = \frac{\mu}{\sigma^2}, \quad \theta_2 = -\frac{1}{2\sigma^2}$$

Gaussian Distribution (2/2)

- The Gaussian distribution can be represented in the form of an exponential family as follows:

  $p(\boldsymbol{x}, \boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{x} - \psi(\boldsymbol{\theta})\right)$

- The partition function $\psi(\boldsymbol{\theta})$ is given as

  $$\psi(\boldsymbol{\theta}) = \frac{\mu^2}{2\sigma^2} + \log(\sqrt{2\pi}\sigma) = -\frac{(\theta_1)^2}{4\theta_2} - \frac{1}{2}\log(-\theta_2) + \frac{1}{2}\log\pi$$

- The dual affine coordinate $\boldsymbol{\eta} = (\eta_1, \eta_2)$ is given as

  $\eta_1 = \mathbf{E}[k_1(x)] = \mathbf{E}[x] = \mu,$

  $\eta_2 = \mathbf{E}[k_2(x)] = \mathbf{E}[x^2] = \mu^2 + \sigma^2$

Discrete Distribution (1/3)

- A discrete distribution of a random variable x over $S = \{0, 1, \dots, n\}$:

  $$p(x) = \sum_{i=0}^{n} p_i \mathbf{1}[x = i] \quad \text{where } p_i = \Pr(x = i)$$

  - Since $\sum p_i = 1$, we can identify it with $\boldsymbol{p} = (p_1, p_2, \dots, p_n)$
- Let us introduce $\boldsymbol{x} = (x_1, \dots, x_n) \in \{0, 1\}^n$ as

  $$x_i = k_i(x) = \mathbf{1}[x = i], \quad i = 1, \dots, n$$

- The natural parameter $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)$ is given as

  $$\theta_i = \log \frac{p_i}{p_0}, \quad i = 1, \dots, n$$

Discrete Distribution (2/3)

- Using

$$\mathbf{1}[x = 0] = 1 - \sum_{i=1}^{n} \mathbf{1}[x = i],$$

we have

$$\log p(x) = \sum_{i=1}^{n} \log p_i \mathbf{1}[x = i] + (\log p_0)\mathbf{1}[x = 0]$$

$$= \sum_{i=1}^{n} \log \frac{p_i}{p_0} \mathbf{1}[x = i] + \log p_0$$

Discrete Distribution (3/3)

- The discrete distribution can be represented in the form of an exponential family as follows:

  $p(x, \boldsymbol{\theta}) = \exp(\, \boldsymbol{\theta} \cdot \boldsymbol{x} + \log p_0 \,) \quad (= \exp(\theta_x + \theta_0))$

- The partition function $\psi(\boldsymbol{\theta})$ is given as

  $$\psi(\boldsymbol{\theta}) = -\log p_0 = \log\left(1 + \sum_{i=1}^{n} \exp \theta_i \right)$$

- The dual affine coordinate $\boldsymbol{\eta} = (\eta_1, \dots, \eta_n)$ is given as

  $\eta_i = \mathbf{E}[k_i(x)] = \mathbf{E}[\, \mathbf{1}[x = i] \,] = p_i, \quad i = 1, \dots, n$

Flatness: *e*-flat and *m*-flat

- Consider two distributions P_1, P_2 with $(\boldsymbol{\theta}_1, \boldsymbol{\eta}_1)$, $(\boldsymbol{\theta}_2, \boldsymbol{\eta}_2)$
- The straight line from $\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}_2$ with some parameter t:

  $\boldsymbol{\theta}(t) = (1-t)\boldsymbol{\theta}_1 + t\boldsymbol{\theta}_2$

  is called an e-geodesic
  - A submanifold defined by linear constraints in θ is e-flat
- When we use $\boldsymbol{\eta}$ instead, the straight line

  $\boldsymbol{\eta}(t) = (1-t)\boldsymbol{\eta}_1 + t\boldsymbol{\eta}_2$

  is called an m-geodesic
  - A submanifold defined by linear constraints in η is m-flat

Dually Flat Structure and Exponential Family

- The dually flat structure of e-flat and m-flat is the central concept in information geometry
- For a random variable $\boldsymbol{x}$, let its θ-coordinate be $\boldsymbol{\theta}(\boldsymbol{x})$
- The Bregman divergence is given as

  $D_\psi[\boldsymbol{\theta}, \boldsymbol{\theta}(\boldsymbol{x})] = \psi(\boldsymbol{\theta}) + \psi(\boldsymbol{\theta}(\boldsymbol{x})) - \boldsymbol{\eta} \cdot (\boldsymbol{\theta} - \boldsymbol{\theta}(\boldsymbol{x}))$
- It induces the following exponential family:

  $p(\boldsymbol{x}, \boldsymbol{\theta}) = \exp(-D_\psi[\boldsymbol{\theta}, \boldsymbol{\theta}(\boldsymbol{x})] + \psi(\boldsymbol{x})) = \exp(\boldsymbol{\theta} \cdot \boldsymbol{x} - \psi(\boldsymbol{\theta}))$
- Dually flat structure $\Rightarrow$ an exponential family
  - It is sufficient to consider only the exponential family

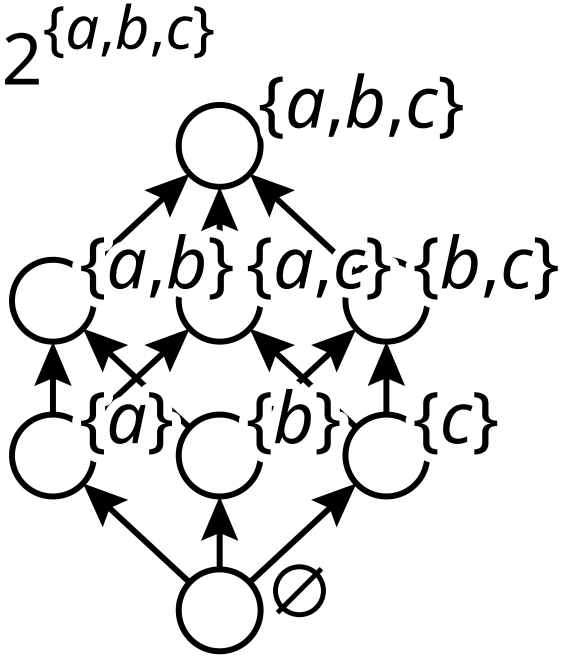
Outline

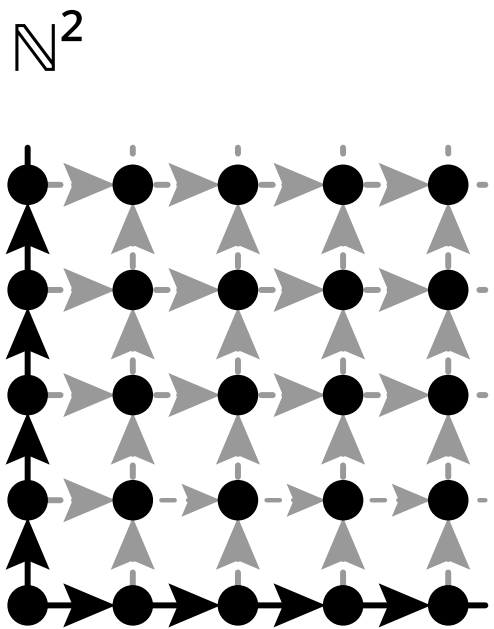
1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

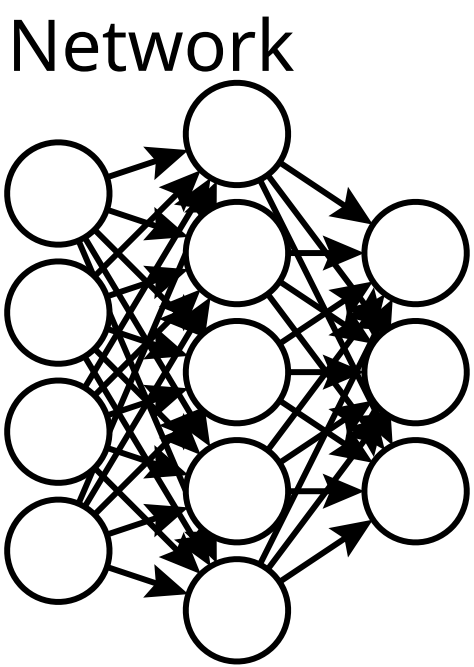
From Vectors to Matrices/Tensors

- In the standard discrete distribution, we do not assume any specific structure in the space $S = \{0, 1, \dots, n\}$ of possible values of the random variable x
- How can we introduce a richer structure, such as matrix or tensor, into S?
- We can use partial orders and their associated incidence algebras
- This will also include Boltzmann machines, where the hierarchy of interaction orders can be incorporated

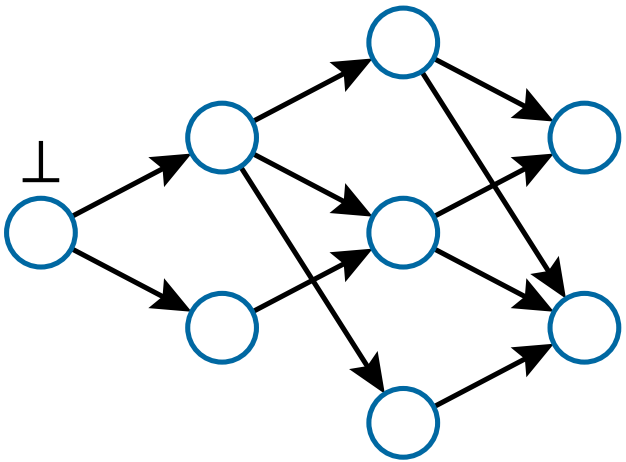
Examples of Partially Ordered Sets (Posets)





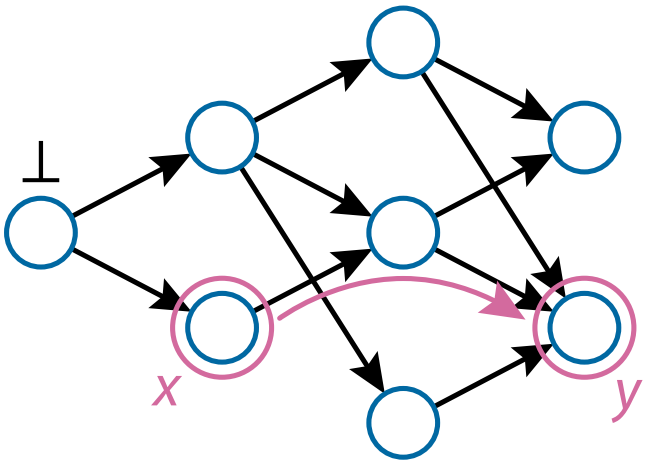


Partially Ordered Set (Poset)



- A 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) $\bot \in S$

Partially Ordered Set (Poset)



- A 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

Incidence Algebra [Rota, 1964]

- Incidence algebra is defined over a poset $(S, \leq)$
- We use (closed) intervals $[a, b] = \{s \in S \mid a \leq s \leq b\}$
- Members of the incidence algebra are functions $\alpha(a, b)$ from intervals $[a, b]$ to a scalar with

  $(\alpha + \beta)(a, b) = \alpha(a, b) + \beta(a, b)$

  $$(\alpha\beta)(a, b) = \sum_{a \leq x \leq b} \alpha(a, x)\beta(x, b) \quad \text{(convolution)}$$

Special Elements

- Delta function δ:

  $$\delta(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

  - The multiplicative identity element
- Zeta function ζ: (integral)

  $$\zeta(a, b) = \begin{cases} 1 & \text{if } a \leq b \\ 0 & \text{otherwise} \end{cases}$$

  - This gives the value 1 for every nonempty interval $[a, b]$
- Möbius function $\mu = \zeta^{-1}$: $\zeta\mu = \delta$ (differential)
  - The multiplicative inverse of the zeta function

(ζ, μ) Leads to Non-Singularity

- For a poset $(S, \leq)$, let $S = \{s_1, s_2, \ldots, s_n\}$
- Let us define the zeta matrix $\mathbf{Z} \in \{0, 1\}^{n \times n}$ as

  $z_{ij} = \zeta(s_i, s_j)$
- Relationship between ζ and μ guarantees that $\mathbf{Z}$ is always regular:

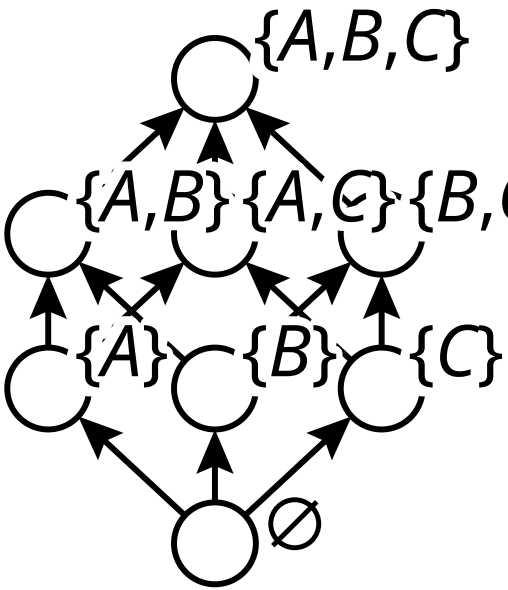
  $\mathbf{Z}^{-1} = \mathbf{M}$ such that $m_{ij} = \mu(s_i, s_j)$
  - $\mathbf{M} \in \mathbb{Z}^{n \times n}$

Möbius Inversion Formula

- Given a poset S, for any functions $f, g : S \to \mathbb{R}$, the Möbius inversion formula is given as follows:

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

Example: Inclusion-Exclusion Principle

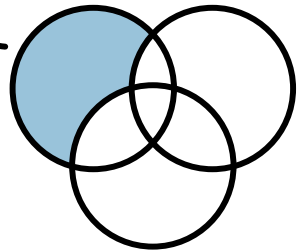


$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

$$f(\mathcal{X}) = \left|\bigcap_{X \in \mathcal{X}} X\right| \qquad g(\mathcal{X}) = \left|\bigcap_{X \in \mathcal{X}} X \setminus \bigcup_{Y \notin \mathcal{X}} Y\right|$$

$$\begin{cases} f(\mathcal{X}) = \sum_{\mathcal{Y} \geq \mathcal{X}} g(\mathcal{Y}) \\ g(\mathcal{X}) = \sum_{\mathcal{Y} \geq \mathcal{X}} \mu(\mathcal{X}, \mathcal{Y}) f(\mathcal{Y}) \end{cases}$$

$$(-1)^{|\mathcal{Y}| - |\mathcal{X}|}$$



Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Log-Linear Model on Posets [Sugiyama et al., 2017]

- For probability $p: S \to (0, 1)$ with $\sum_{x \in S} p(x) = 1$, introduce $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ as

  $$\theta_x = \sum_{s \in S} \mu(s, x) \log p(s),$$

  $$\eta_x = \sum_{s \in S} \zeta(x, s) p(s) = \sum_{s \geq x} p(s)$$

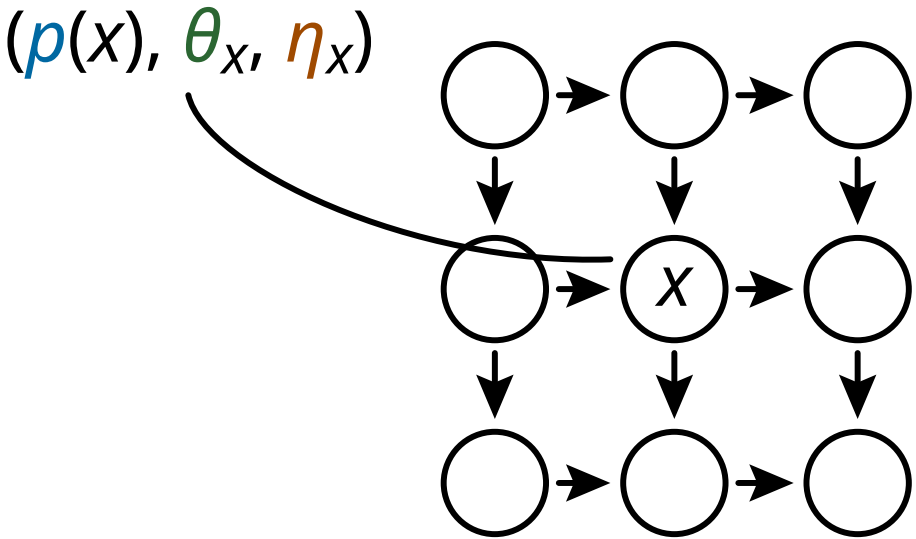
- From the Möbius inversion formula, log-linear model is:

  $$\log p(x) = \sum_{s \in S} \zeta(s, x) \theta_s = \sum_{s \leq x} \theta_s$$

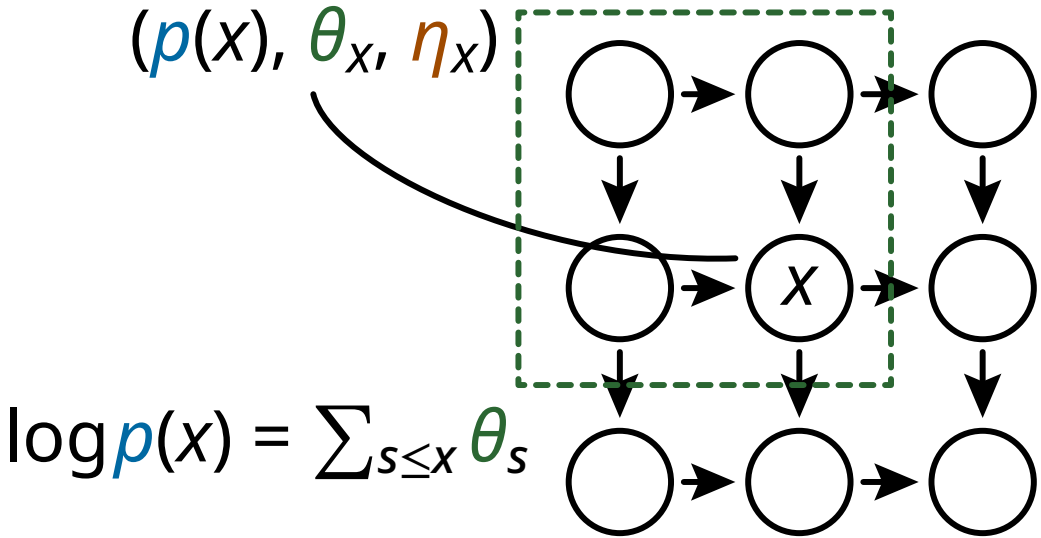
- In the matrix form, it can be written as

  $$\log \boldsymbol{p} = \mathsf{Z} \boldsymbol{\theta}$$

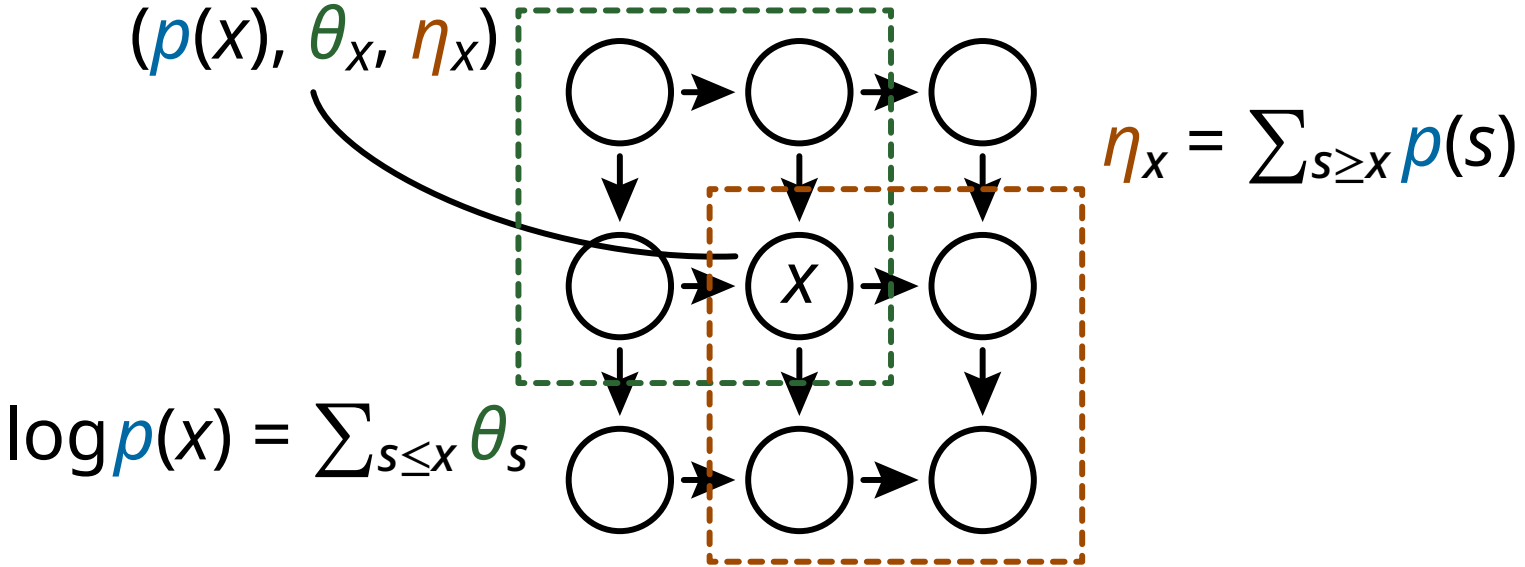
Log-Linear Model on Posets



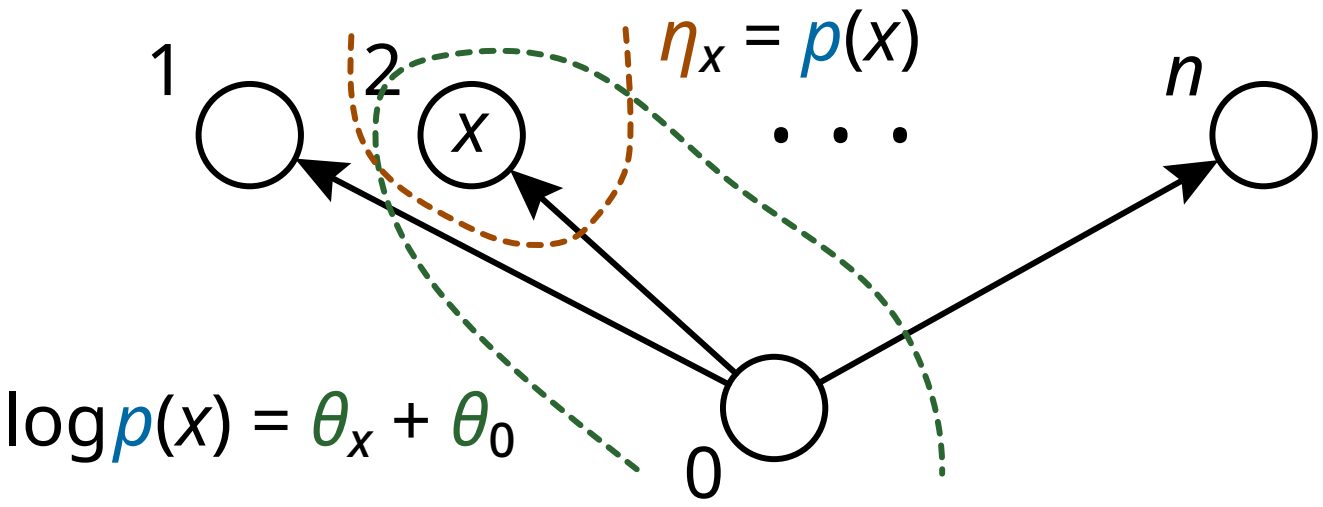
Log-Linear Model on Posets



Log-Linear Model on Posets



Discrete Distribution

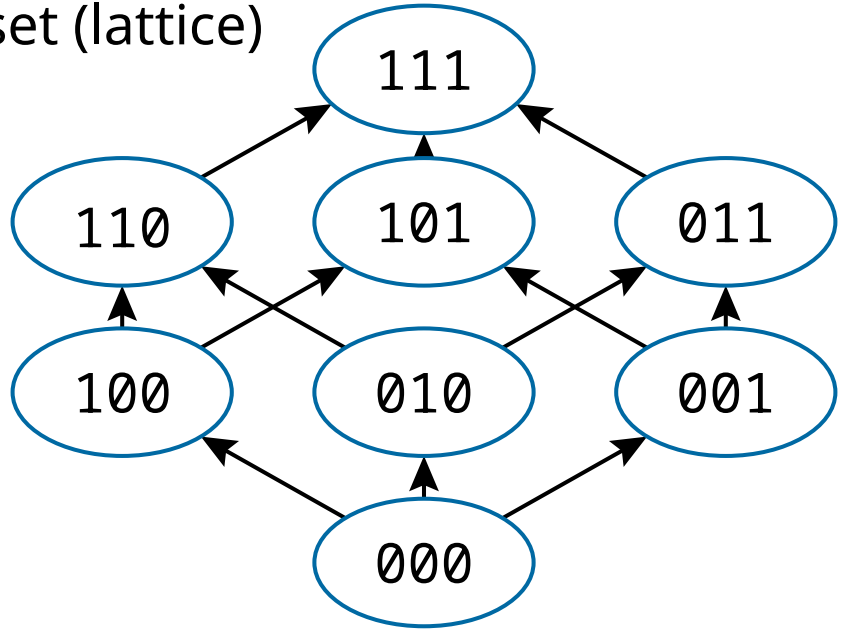


Boltzmann Machines

Dataset

	Bread	Milk	Apple
ID 1	1	0	0
ID 2	1	1	0
ID 3	1	0	0
ID 4	0	1	1
ID 5	0	1	1
ID 6	1	1	0
ID 7	1	0	0
ID 8	1	1	0

Poset (lattice)

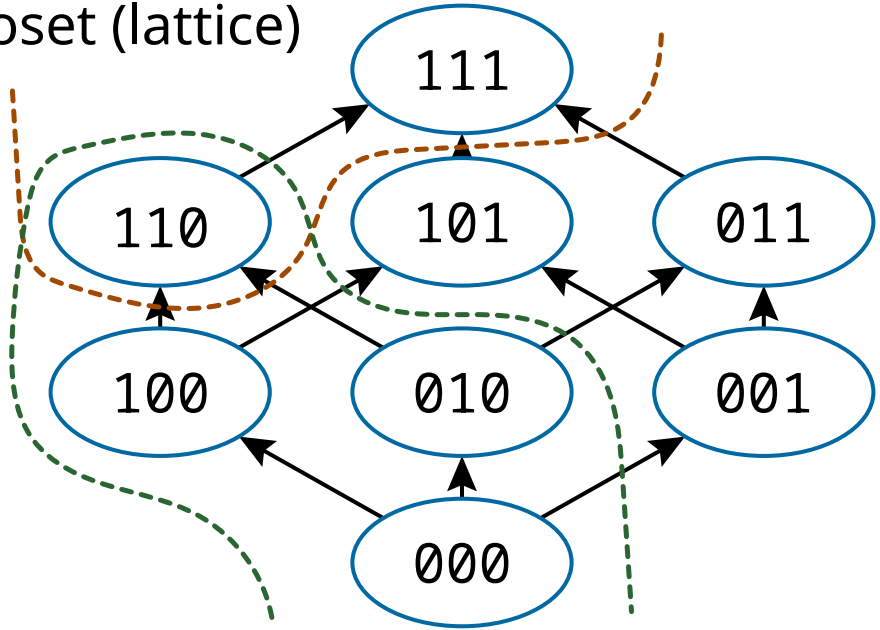


Boltzmann Machines

$$\eta_x = \sum_{s \geq x} p(s)$$
$$= \mathbb{E}[x_{i_1} \cdots x_{i_k}]$$
$$= \Pr(x_{i_1} = \cdots = x_{i_k} = 1)$$

$$\log p(x) = \sum_{s \leq x} \theta_s$$

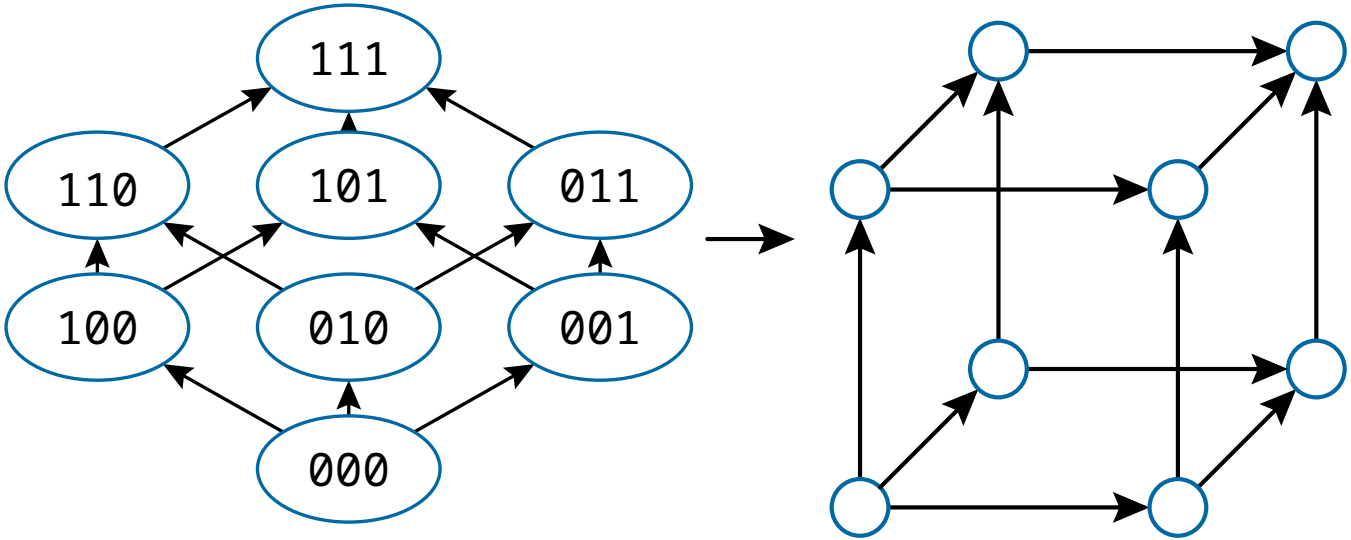
Poset (lattice)



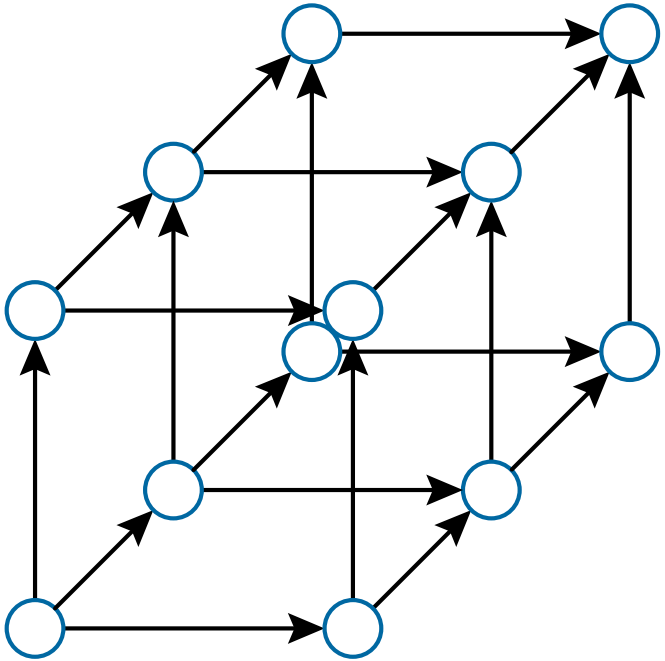
Treating Tensors as Posets

Poset (lattice)

variables = # modes

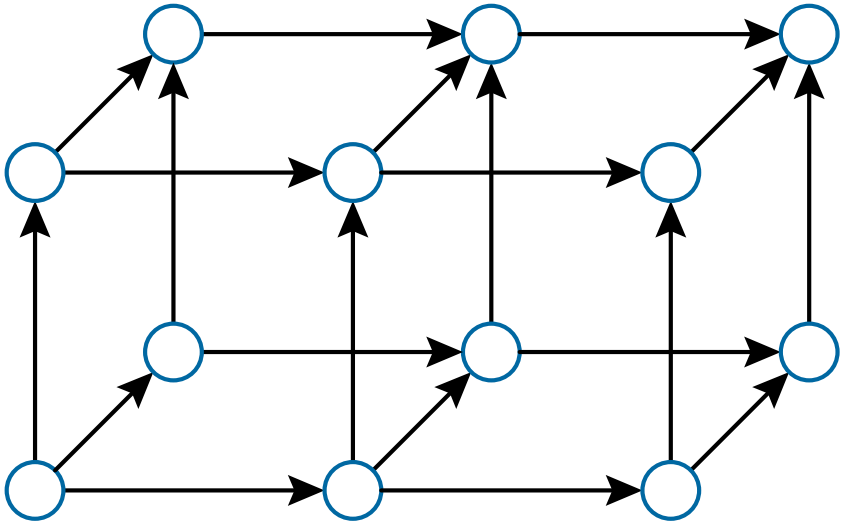


Treating Tensors as Posets



Treating Tensors as Posets

Any (positive) tensor can be treated as an instance of a poset



Relationship to Exponential Family

- In terms of the general form of the exponential family:

  $p(x, \boldsymbol{\theta}) = \exp\left(\sum \theta_i k_i(x) + r(x) - \psi(\boldsymbol{\theta})\right)$

  we can remove $r(x)$ and

  $$k_s(x) = \zeta(s, x) = \begin{cases} 1 & \text{if } s \leq x, \\ 0 & \text{otherwise,} \end{cases} \qquad \psi(\boldsymbol{\theta}) = -\theta_\perp = -\log p_\perp$$

  - If we replace $\zeta(s, x)$ with $\delta(s, x)$, it coincides with the standard discrete distribution

- $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ of the log-linear model on posets directly corresponds to natural and expectation parameters of the exponential family

Dually Flat Structure

- Let us formally introduce the dually flat structure for the log-linear model on posets
- Let $\psi(\theta) = -\theta(\bot)$ (convex, partition function)

$$\psi(\theta) \xrightarrow{\text{Legendre transformation}} \phi(\eta) = \max_{\boldsymbol{\theta}}(\boldsymbol{\theta} \cdot \boldsymbol{\eta} - \psi(\boldsymbol{\theta})) = \sum_{x \in S} p(x) \log p(x)$$

- $(\psi(\theta), \phi(\eta))$ leads to dually flat coordinate system (θ, η):

$$\nabla\psi(\theta) = \eta, \quad \frac{\partial}{\partial\theta_x}\psi(\theta) = \eta_x$$

$$\nabla\phi(\eta) = \theta, \quad \frac{\partial}{\partial\eta_x}\phi(\eta) = \theta_x$$

Riemannian Metric (Fisher Information)

$$\frac{\partial}{\partial \theta_x}\frac{\partial}{\partial \theta_y}\psi(\theta) = \frac{\partial}{\partial \theta_x}\eta_y = \sum_{s \in S} \zeta(x,s)\zeta(y,s)p(s) - \eta_x \eta_y$$

$$\frac{\partial}{\partial \eta_x}\frac{\partial}{\partial \eta_y}\phi(\eta) = \frac{\partial}{\partial \eta_x}\theta_y = \sum_{s \in S} \mu(s,x)\mu(s,y)p(s)^{-1}$$

$$\mathbb{E}_s\left[\frac{\partial}{\partial \theta_x}\log p(s)\frac{\partial}{\partial \eta_y}\log p(s)\right] = \delta(x,y) \quad \text{(orthogonality)}$$

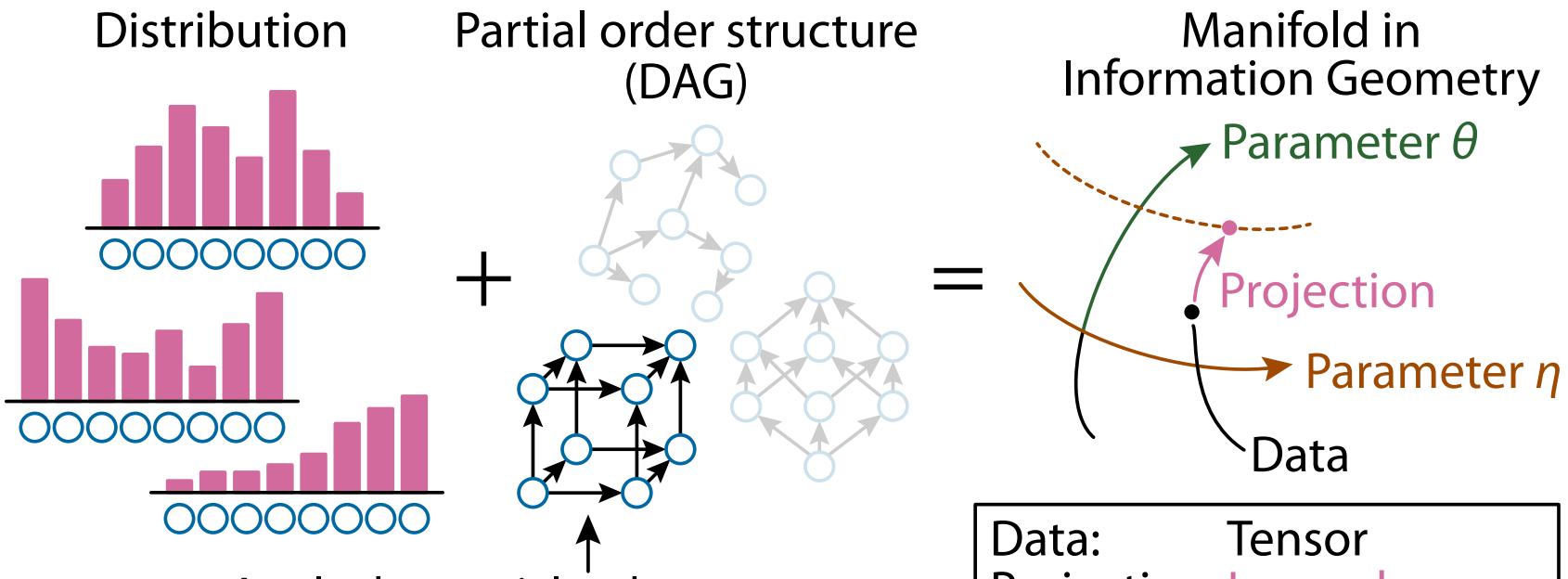
Riemannian Metric (Fisher Information)

$$\mathbb{E}_s\left[\frac{\partial}{\partial\theta_x}\log p(s)\frac{\partial}{\partial\theta_y}\log p(s)\right] = \sum_{s\in S}\zeta(x,s)\zeta(y,s)p(s) - \eta_x\eta_y$$

$$\mathbb{E}_s\left[\frac{\partial}{\partial\eta_x}\log p(s)\frac{\partial}{\partial\eta_y}\log p(s)\right] = \sum_{s\in S}\mu(s,x)\mu(s,y)p(s)^{-1}$$

$$\mathbb{E}_s\left[\frac{\partial}{\partial\theta_x}\log p(s)\frac{\partial}{\partial\eta_y}\log p(s)\right] = \delta(x,y) \quad \text{(orthogonality)}$$

Overview of Our Approach [Sugiyama et al., 2018]

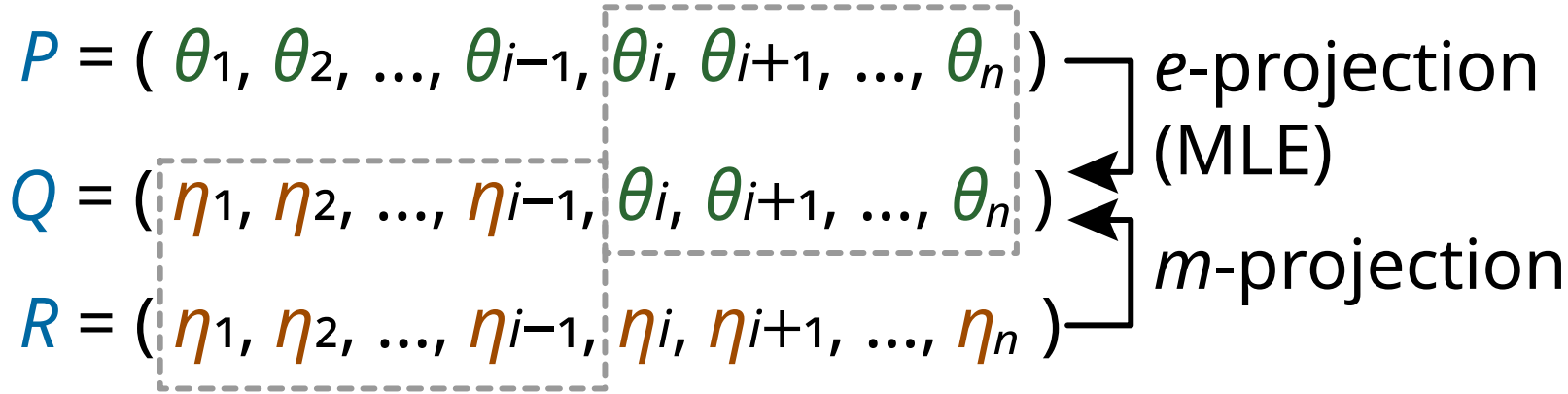


Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Mixed Coordinate System

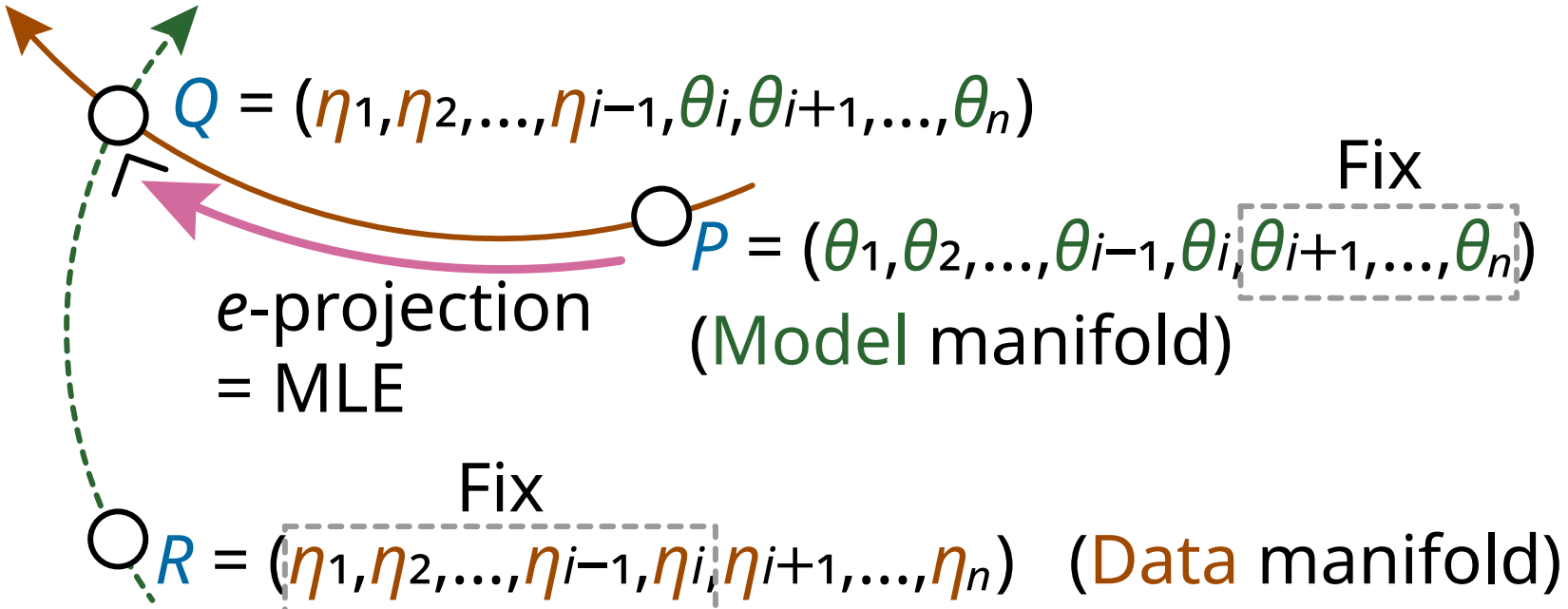
- Many problems are formulated as coordinate mixing



Pythagorean theorem:
KL(P, R) = KL(P, Q) + KL(Q, R)

(Q is always unique)

Two Submanifolds



Gradient methods for *e*-projection

- *e*-projection is convex optimization
- Gradient descent (first-order):

  $\theta_{\text{next}} \leftarrow \theta - \varepsilon(\eta - \hat{\eta}_{\text{target}})$
- Natural gradient (second-order)

  $\theta_{\text{next}} \leftarrow \theta - G^{-1}(\eta - \hat{\eta}_{\text{target}})$
  - G is Fisher information matrix w.r.t. θ
- Coordinate descent [Hayashi et al., 2020]

Mixed Coordinate System (Example)

- Many problems are formulated as coordinate mixing

$P = (0, 0, \ldots, 0, 0, 0, \ldots, 0)$ → Uniform dist.

$Q = (\hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_{i-1}, 0, 0, \ldots, 0)$

$R = (\hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_{i-1}, \hat{\eta}_i, \hat{\eta}_{i+1}, \ldots, \hat{\eta}_n)$ → Empirical dist.

Pythagorean theorem: (Q is always unique)

$\mathrm{KL}(P, R) = \mathrm{KL}(P, Q) + \mathrm{KL}(Q, R)$

Mixed Coordinate System (Example)

- Many problems are formulated as coordinate mixing

$$P_1 = (\ \theta_1^{(1)},\ \theta_2^{(1)},\ \ldots,\ \theta_k^{(1)},\ \boxed{0,\ 0,\ \ldots,\ 0}\ )$$ Initial state

$$\vdots \qquad\qquad \vdots$$

$$P_t = (\ \theta_1^{(t)},\ \theta_2^{(t)},\ \ldots,\ \theta_k^{(t)},\ 0,\ 0,\ \ldots,\ 0\ )$$ Converged

$$= (\ \eta_1^{(t)},\ \eta_2^{(t)},\ \ldots,\ \eta_k^{(t)},\ 0,\ 0,\ \ldots,\ 0\ )$$

$$\| \qquad \| \qquad\qquad \|$$

$$\hat{\eta}_1 \quad \hat{\eta}_2 \qquad \hat{\eta}_k$$

Fixed

Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Mathematical Modeling of MBA

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$$= H_0 + H_i^{(1)} + \cdots + H_l^{(4)} + H_{ij}^{(12)} + \cdots + H_{kl}^{(34)} + H_{ijk}^{(123)} + \cdots + H_{jkl}^{(234)} + H_{ijkl}^{(1234)}$$

One-Body Approximation

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$= H_0 + H_i^{(1)} + \cdots + H_l^{(4)} + 0 + \cdots + 0 + 0 + \cdots + 0 + 0$

$\sum \theta_{111l'}$

One body

- **Fix the unused parameters at zero**

Two-Body Approximation

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$= H_0 + H_i^{(1)} + \cdots + H_l^{(4)} + H_{ij}^{(12)} + \cdots + H_{kl}^{(34)} + 0 + \cdots + 0 + 0$

$\sum\sum \theta_{11k'l'}$

Two body

- **Fix the unused parameters at zero**

Three-Body Approximation

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$$= H_0 + H_i^{(1)} + \cdots + H_l^{(4)} + H_{ij}^{(12)} + \cdots + H_{kl}^{(34)} + H_{ijk}^{(123)} + \cdots + H_{jkl}^{(234)} + 0$$

$$\sum\sum\sum \theta_{1j'k'l'}$$

Three body

- **Fix the unused parameters at zero**

Many-Body Approximation for Tensors

- We explicitly model associations between features/modes using the log-linear model on posets

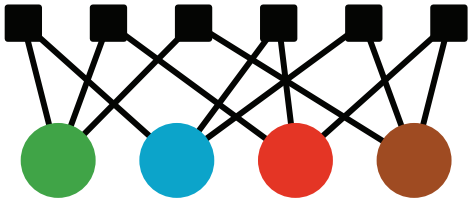
One-body approx.

$\mathcal{P}_{ijkl} \simeq p_i^{(1)} p_j^{(2)} p_k^{(3)} p_l^{(4)}$

= Rank-1 approximation
(mean-field approximation)

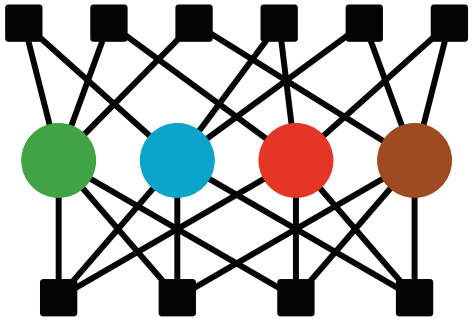
Two-body approx.

$\mathcal{P}_{ijkl} \simeq \chi_{ij}^{(12)} \chi_{ik}^{(13)} \cdots \chi_{kl}^{(34)}$



Three-body approx.

$\mathcal{P}_{ijkl} \simeq \chi_{ijk}^{(123)} \chi_{ijl}^{(124)} \chi_{ikl}^{(134)} \chi_{jkl}^{(234)}$



Larger capability

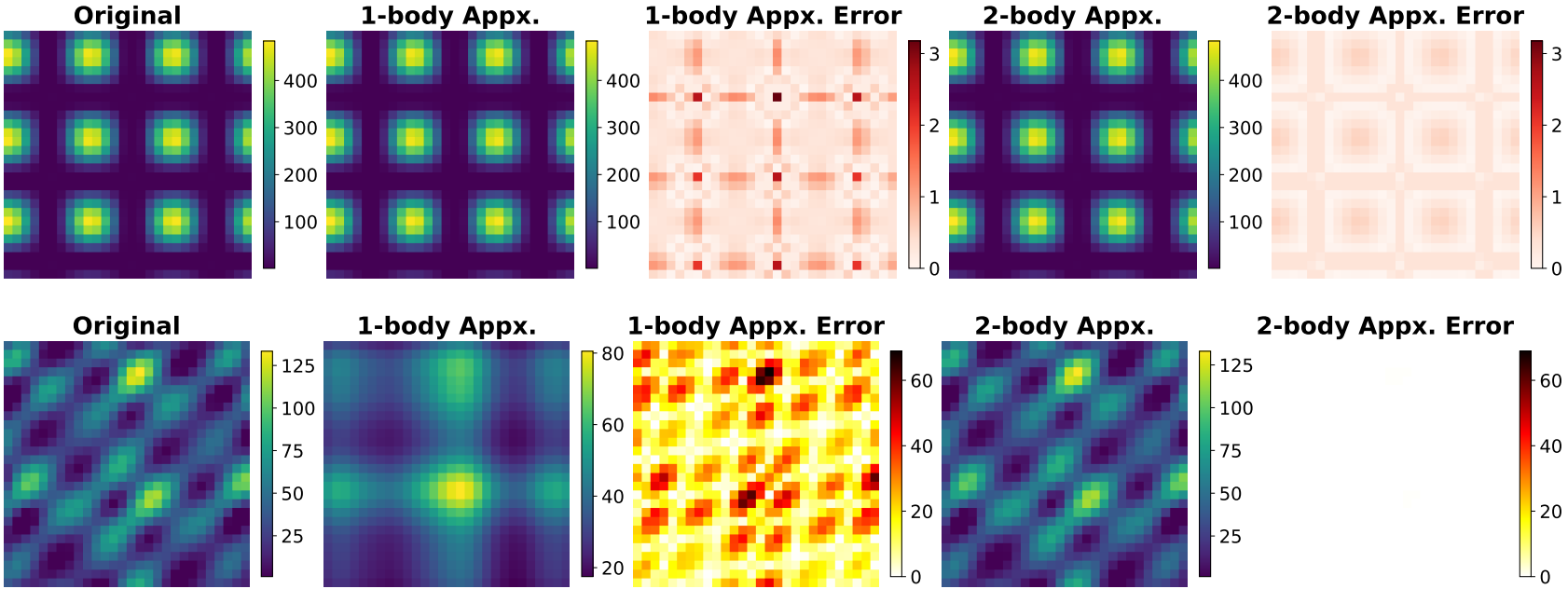
Refined Information [Enouen & Sugiyama, 2026]

- Given a tensor $\boldsymbol{P}$, let its k-body approximation be $\boldsymbol{P}_k$
  - $\boldsymbol{P}_0$ corresponds to the uniform distribution
  - When the number of modes of $\boldsymbol{P}$ is n, $\boldsymbol{P} = \boldsymbol{P}_n$
- We can decompose the KL-divergence as:

  $D_{KL}[\boldsymbol{P}, \boldsymbol{P}_0] = D_{KL}[\boldsymbol{P}, \boldsymbol{P}_{n-1}] + D_{KL}[\boldsymbol{P}_{n-1}, \boldsymbol{P}_{n-2}] + \cdots + D_{KL}[\boldsymbol{P}_1, \boldsymbol{P}_0]$

  and each $D_{KL}[\boldsymbol{P}_k, \boldsymbol{P}_{k-1}]$ is called the refined information
- The refined information intuitively shows the amount of information of the k-body interaction

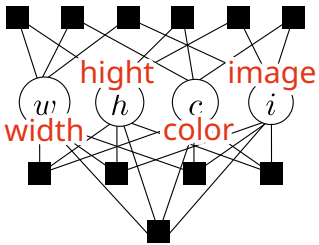
MBA on Synthetic Data (Images)



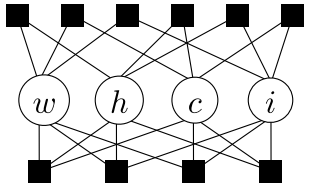
MBA on Real Data (Images)

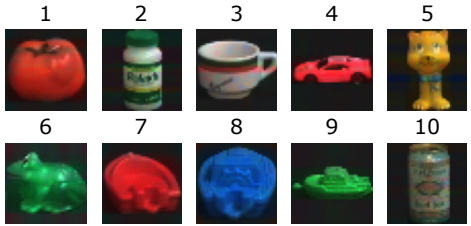
More flexible modeling by many-body approximation

a. Up to four-body

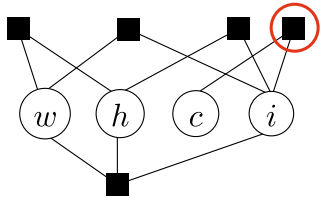


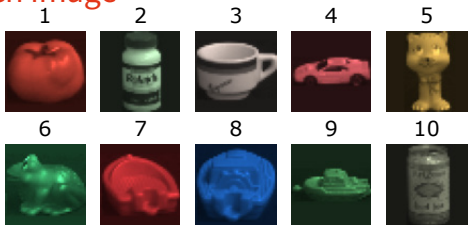
b. Up to three-body



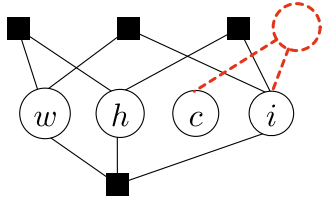


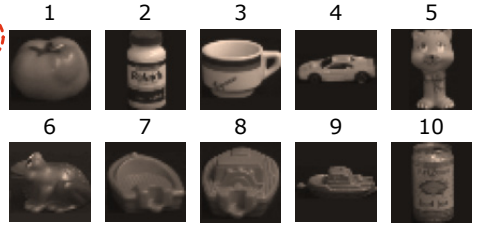
c. Color changes for each image



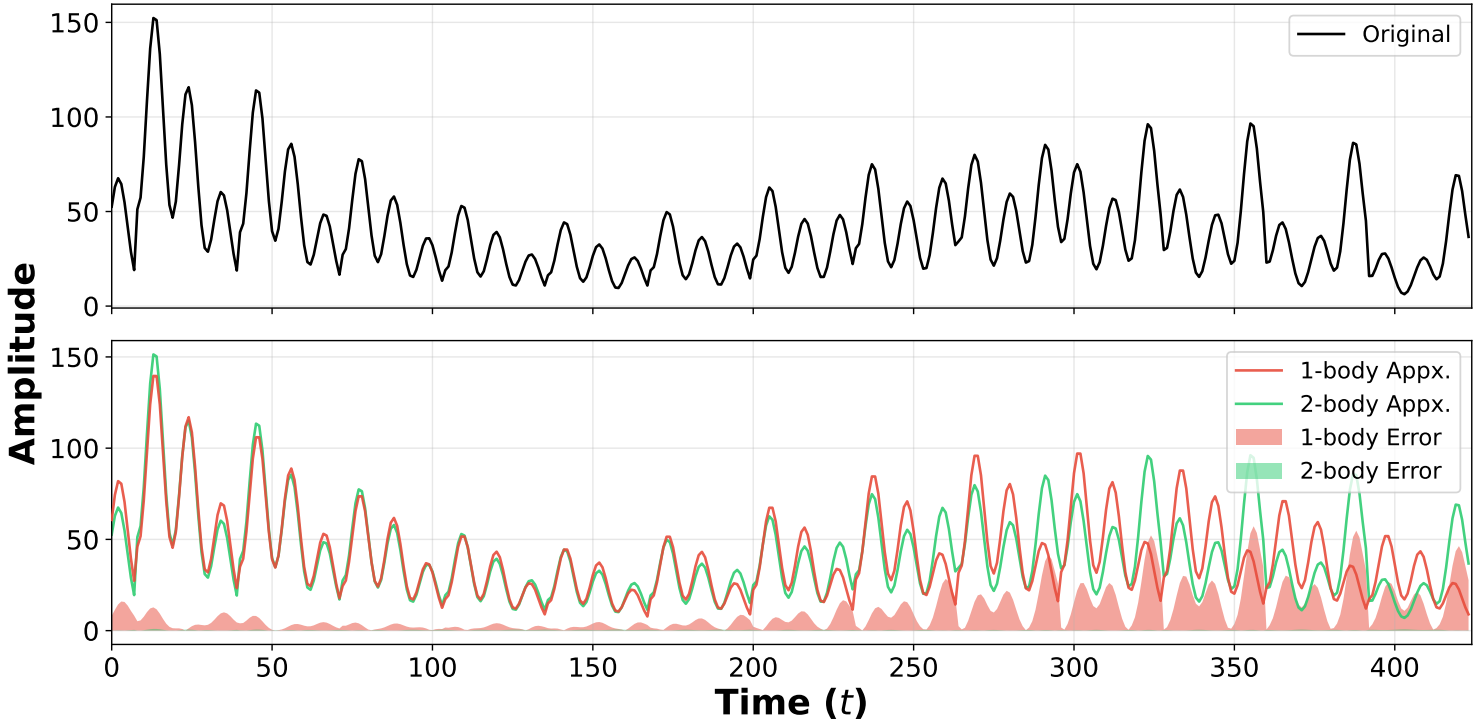


d. When the interaction between colors and image indices is removed...





MBA on Synthetic Data (Times Series)



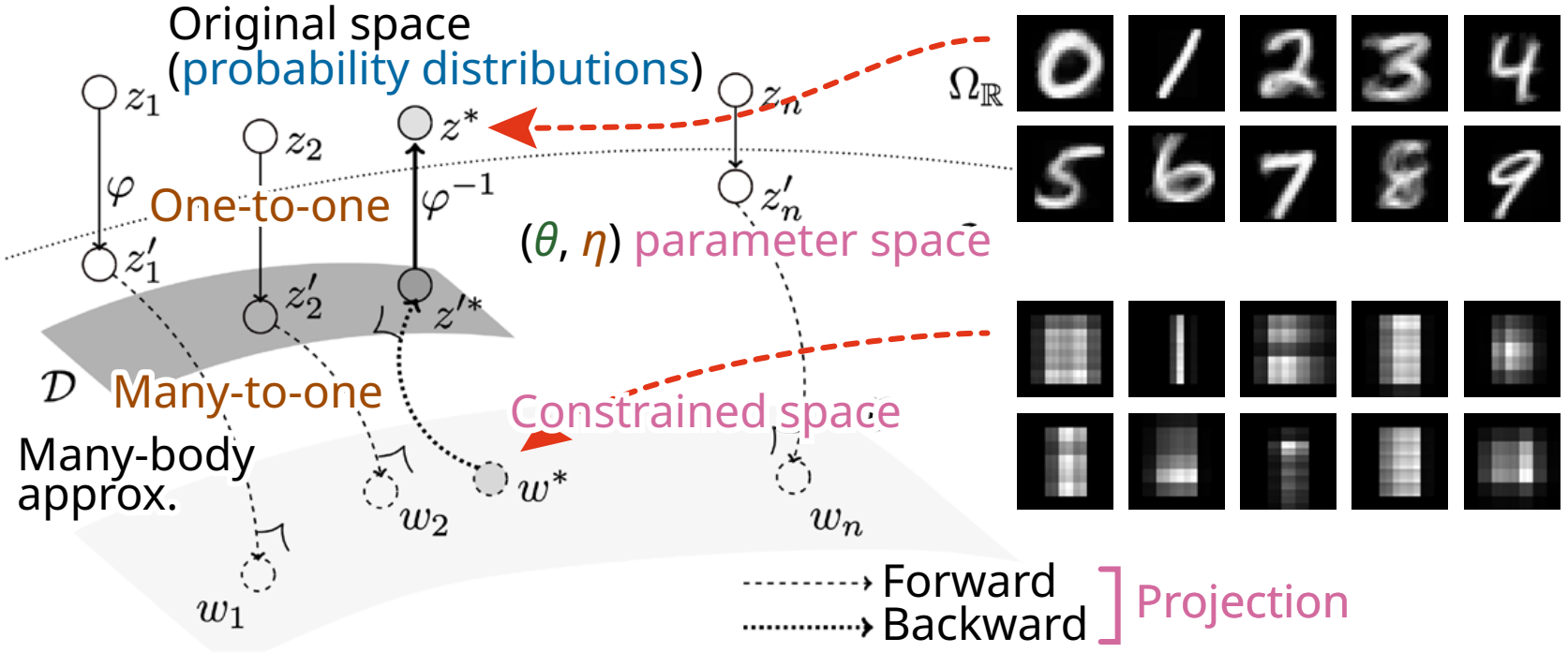
Recent Advances of MBA

- Generalization using alternative affine coordinates [Zhou & Sugiyama, 2026]
- Tensor-based discrete density estimation via the EM algorithm, where MBA is used as an M-step [Ghalamkari et al., 2026]
- Incorporating deformed decomposition [Ghalamkari et al., 2026]
- Tutorial at UAI 2026 by Ghalamkari and Mørup

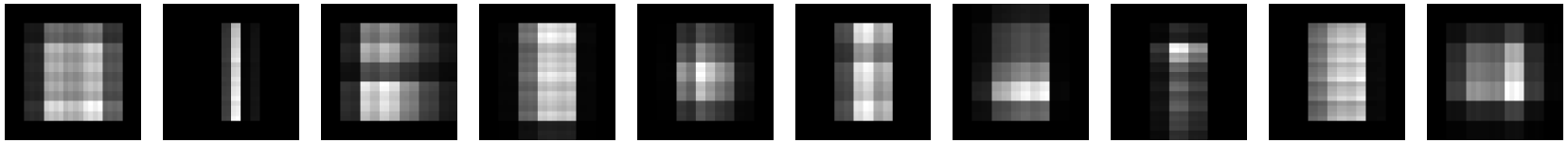
Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Data Augmentation (Revisited)

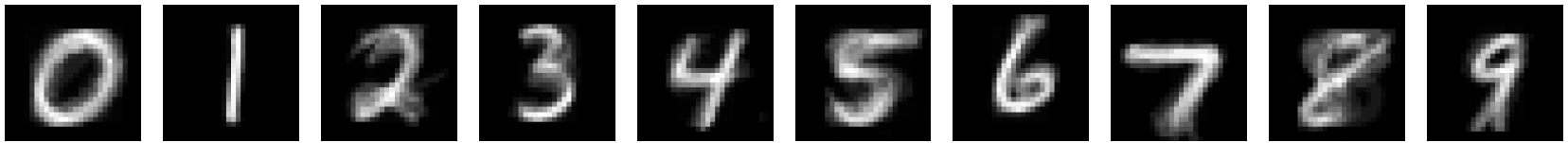


Data Augmentation on MNIST



Forward projection (MBA) using 17 θ-parameters

$$(\theta_0, \underbrace{\theta_1, \dots, \theta_{17}}_{\text{17 parameters are available}}, \underbrace{0, 0, 0, \dots, 0, 0}_{\text{767 parameters are fixed}})$$



Backward projection using 767 θ-parameters

$$(\theta_0, \underbrace{\bar{\theta}_1, \dots, \bar{\theta}_{17}}_{\text{17 parameters are fixed to mean of neighbor } \theta}, \underbrace{\theta_{18}, \theta_{19}, \theta_{20}, \dots, \theta_{783}, \theta_{784}}_{\text{767 parameters are available}})$$

Data Augmentation on CIFAR-10

Forward projection (MBA) using 1,410 θ-parameters

$$(\theta_0, \underbrace{\theta_1, \dots, \theta_{1410}}_{\text{1410 parameters are available}}, \underbrace{0, \dots, 0}_{\text{1662 parameters are fixed}})$$

Backward projection using 2,334 θ-parameters

$$(\theta_0, \underbrace{\bar{\theta}_1, \dots, \bar{\theta}_{738}}_{\text{738 parameters are fixed to mean of neighbor } \theta}, \underbrace{\theta_{739}, \dots, \theta_{3072}}_{\text{2334 parameters are available}})$$

Test Accuracy on Real Data (1/2)

Training Set	Dataset		
	MNIST	CIFAR-10	Speech Commands
OG	$97.98 \pm 0.19\%$	$88.57 \pm 0.57\%$	$84.48 \pm 0.50\%$
OG$^{\text{STD}}$	$97.98 \pm 0.24\%$	$89.89 \pm 0.44\%$	$82.98 \pm 0.50\%$
OG$^{\text{AE}}$	$97.97 \pm 0.25\%$	$88.36 \pm 0.46\%$	$83.13 \pm 0.32\%$
OG$^{\text{MU}}$	$96.45 \pm 0.23\%$	$86.60 \pm 0.49\%$	$81.85 \pm 0.61\%$
OG$^{\text{MMU}}$	$97.52 \pm 0.30\%$	$88.02 \pm 0.39\%$	$83.06 \pm 0.54\%$
OG$^{\text{PROPOSAL}}$	$97.91 \pm 0.21\%$	$88.07 \pm 0.46\%$	$84.35 \pm 0.37\%$

STD: Standard techniques, **AE**: AutoEncoder, **MU**: MixUp,
MMU: Manifold MixUp, **PROPOSAL**: Our proposal

Test Accuracy on Real Data (2/2)

Training Set	Dataset		
	Connect. Bench	Taiwanese Bank.	Wine Quality
OG	$88.10 \pm 8.58\%$	$96.54 \pm 0.56\%$	$55.00 \pm 1.69\%$
OG$^{\text{STD}}$	$85.24 \pm 7.66\%$	$96.17 \pm 0.57\%$	$57.85 \pm 1.81\%$
OG$^{\text{AE}}$	$82.86 \pm 7.59\%$	$95.92 \pm 0.62\%$	$57.23 \pm 1.67\%$
OG$^{\text{MU}}$	$89.29 \pm 4.97\%$	$96.55 \pm 0.68\%$	$57.76 \pm 1.67\%$
OG$^{\text{MMU}}$	$91.19 \pm 5.06\%$	$96.44 \pm 0.53\%$	$58.70 \pm 1.74\%$
OG$^{\text{PROPOSAL}}$	$93.81 \pm 4.54\%$	$96.53 \pm 0.47\%$	$59.03 \pm 1.74\%$

STD: Standard techniques, **AE**: AutoEncoder, **MU**: MixUp, **MMU**: Manifold MixUp, **PROPOSAL**: Our proposal

Outline

1. Boltzmann machines
2. Information geometry
3. Partially ordered sets
4. Log-linear model on posets
5. Projection via Mixed coordinates
6. Many-body approximation
7. Data augmentation with many-body approximation
8. Matrix balancing

Matrix Balancing

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

Matrix Balancing

Find r and s:

(Make doubly stochastic matrix)

$$\begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$$

$$= \begin{bmatrix} r_1 s_1 p_{11} & r_1 s_2 p_{12} \\ r_2 s_1 p_{21} & r_2 s_2 p_{22} \end{bmatrix} \begin{matrix} \rightarrow \sum_j r_1 s_j p_{1j} = 1 \\ \rightarrow \sum_j r_2 s_j p_{2j} = 1 \end{matrix}$$

$$\downarrow \qquad \downarrow$$

$$\sum_i r_i s_1 p_{i1} = 1 \qquad \sum_i r_i s_2 p_{i2} = 1$$

Sinkhorn-Knopp Algorithm

- Alternately rescale all rows and columns of a matrix P to sum to 1
- Commonly used to compute entropy-regularized Optimal transport (Wasserstein distance)
  - [Cuturi, 2013]

Example of Sinkhorn–Knopp Algorithm

$$\begin{bmatrix} 0.8 & 0.1 & 0.7 \\ 0.2 & 0.1 & 0.5 \\ 0.4 & 0.2 & 0.3 \end{bmatrix}$$

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

0.8	0.1	0.7	×1/1.6	1.6
0.2	0.1	0.5	×1/0.8	0.8
0.4	0.2	0.3	×1/0.9	0.9
1.4	0.4	1.5		

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

$$\begin{bmatrix} 0.500 & 0.062 & 0.437 \\ 0.250 & 0.125 & 0.625 \\ 0.444 & 0.222 & 0.333 \end{bmatrix} \begin{matrix} \rightarrow 1.0 \\ \rightarrow 1.0 \\ \rightarrow 1.0 \end{matrix}$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$1.194 \quad 0.409 \quad 1.395$$

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

$$\begin{bmatrix} 0.500 & 0.062 & 0.437 \\ 0.250 & 0.125 & 0.625 \\ 0.444 & 0.222 & 0.333 \end{bmatrix} \quad \begin{matrix} 1.0 \\ 1.0 \\ 1.0 \end{matrix}$$

$\times 1/1.194$ $\times 1/0.409$ $\times 1/1.395$

1.194 0.409 1.395

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

$$\begin{bmatrix} 0.419 & 0.153 & 0.313 \\ 0.209 & 0.305 & 0.448 \\ 0.372 & 0.542 & 0.239 \end{bmatrix} \begin{matrix} \rightarrow 0.885 \\ \rightarrow 0.962 \\ \rightarrow 1.153 \end{matrix}$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$1.0 \qquad 1.0 \qquad 1.0$$

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

$$\begin{bmatrix} 0.473 & 0.172 & 0.354 \\ 0.218 & 0.317 & 0.465 \\ 0.323 & 0.470 & 0.207 \end{bmatrix} \begin{matrix} \rightarrow 1.0 \\ \rightarrow 1.0 \\ \rightarrow 1.0 \end{matrix}$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$1.013 \quad 0.960 \quad 1.027$$

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

$$\begin{bmatrix} 0.467 & 0.18 & 0.345 \\ 0.215 & 0.33 & 0.453 \\ 0.318 & 0.49 & 0.202 \end{bmatrix} \begin{matrix} \rightarrow 0.992 \\ \rightarrow 0.998 \\ \rightarrow 1.010 \end{matrix}$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

$$1.0 \qquad 1.0 \qquad 1.0$$

Repeat many times...

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed

$$\begin{bmatrix} 0.471 & 0.182 & 0.348 \\ 0.215 & 0.332 & 0.453 \\ 0.315 & 0.486 & 0.199 \end{bmatrix} \begin{matrix} \rightarrow 1.0 \\ \rightarrow 1.0 \\ \rightarrow 1.0 \end{matrix}$$

$$\downarrow \qquad \downarrow \qquad \downarrow$$

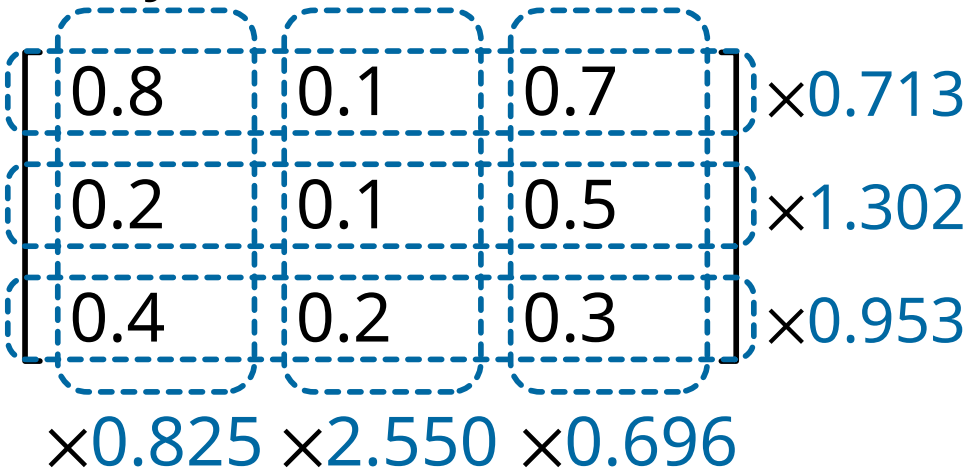
$$1.0 \qquad 1.0 \qquad 1.0$$

This is the solution

Example of Sinkhorn–Knopp Algorithm

Objective: Normalize each row and column so that their sums are 1

Rule: Only row- and column-wise scaling by constants is allowed



Sinkhorn–Knopp algorithm (optimal transport)

Revisit Matrix Balancing [Sugiyama et al, 2017]

$$p_{11} \quad p_{12} \quad p_{13}$$

$$p_{21} \quad p_{22} \quad p_{23}$$

$$p_{31} \quad p_{32} \quad p_{33}$$

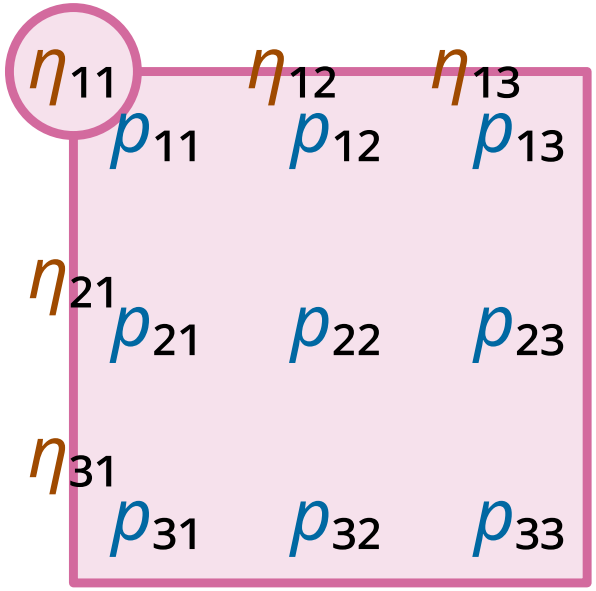
Introduce η

η_{11} η_{12} η_{13}
p_{11} p_{12} p_{13}

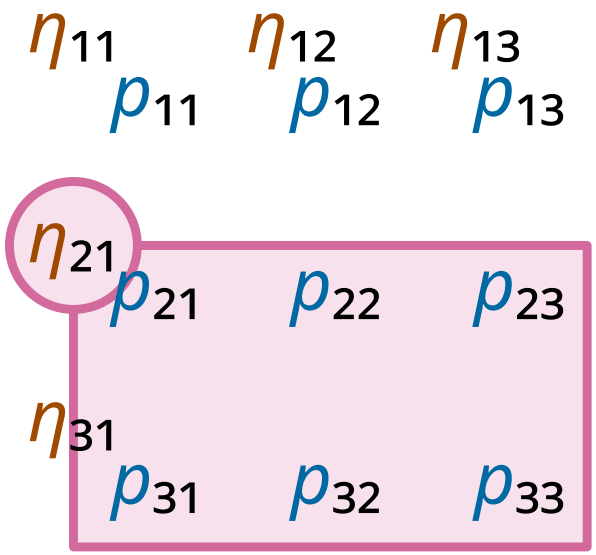
η_{21}
p_{21} p_{22} p_{23}

η_{31}
p_{31} p_{32} p_{33}

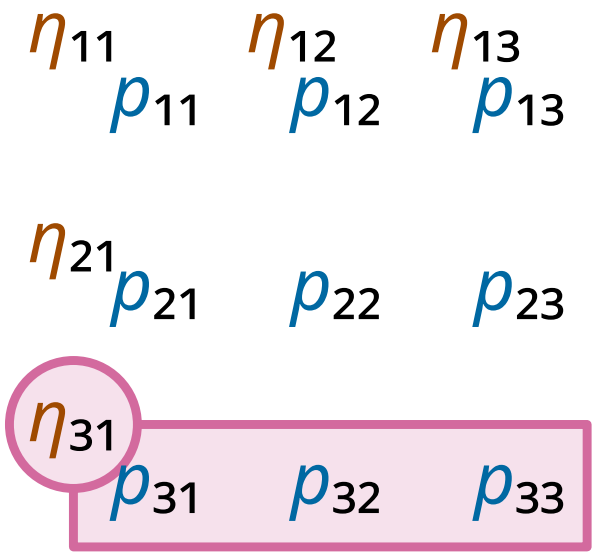
Introduce η



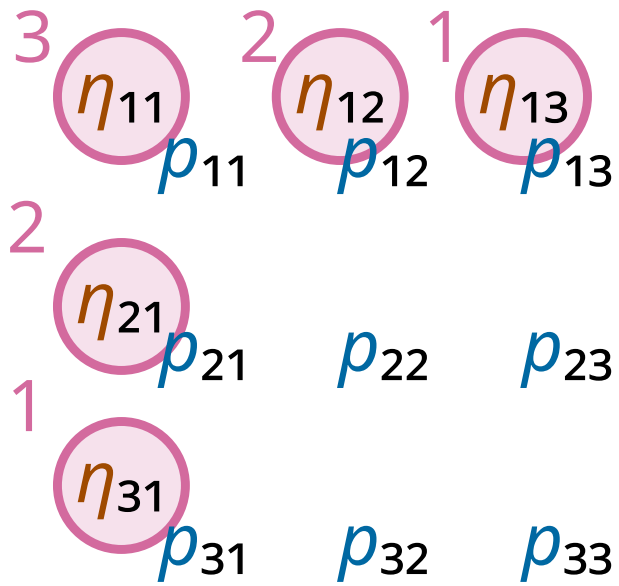
Introduce η



Introduce η



Introduce η

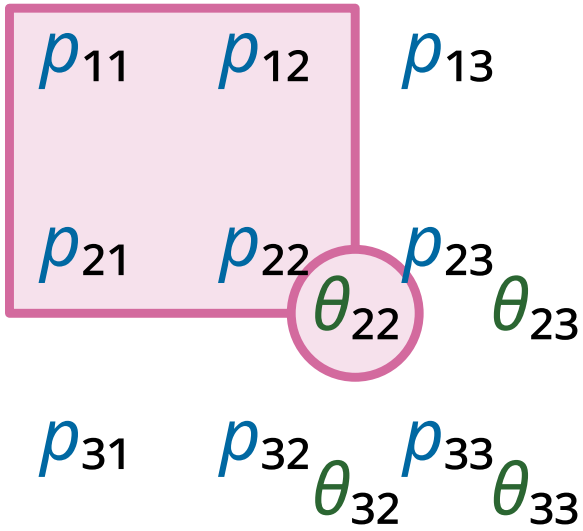


Introduce θ

$$\begin{array}{lll} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22}\ \theta_{22} & p_{23}\ \theta_{23} \\ p_{31} & p_{32}\ \theta_{32} & p_{33}\ \theta_{33} \end{array}$$

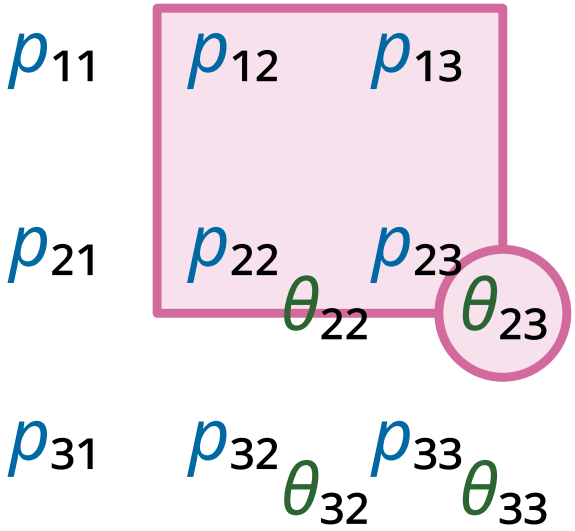
$$\theta_{ij} = \log p_{ij} - \log p_{i-1j} - \log p_{ij-1} + \log p_{i-1j-1}$$

Introduce θ



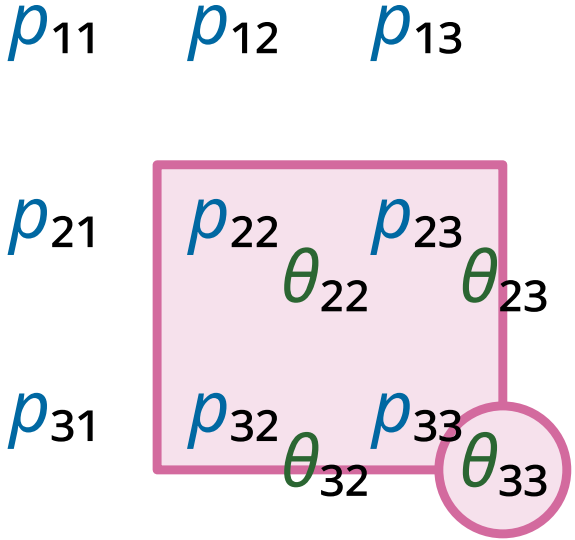
$$\theta_{ij} = \log p_{ij} - \log p_{i-1j} - \log p_{ij-1} + \log p_{i-1j-1}$$

Introduce θ



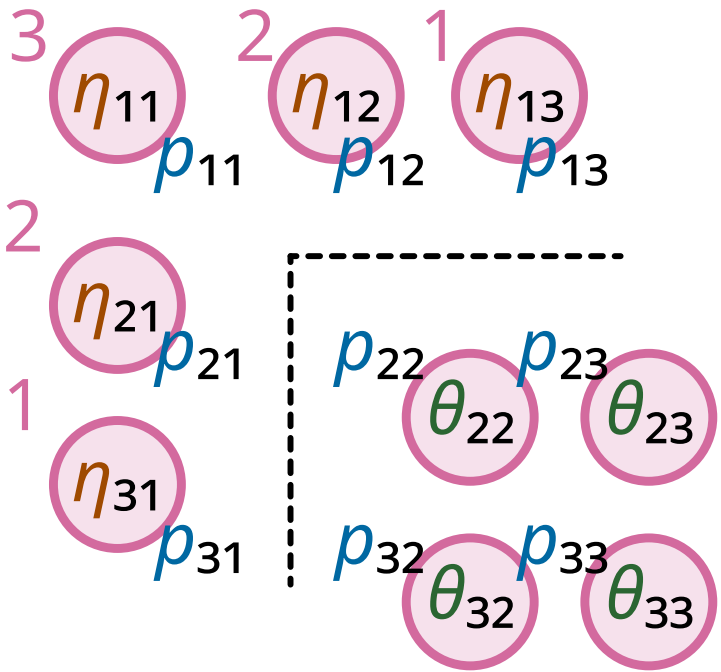
$$\theta_{ij} = \log p_{ij} - \log p_{i-1j} - \log p_{ij-1} + \log p_{i-1j-1}$$

Introduce θ



$$\theta_{ij} = \log p_{ij} - \log p_{i-1j} - \log p_{ij-1} + \log p_{i-1j-1}$$

Balancing as Constraints on η and θ



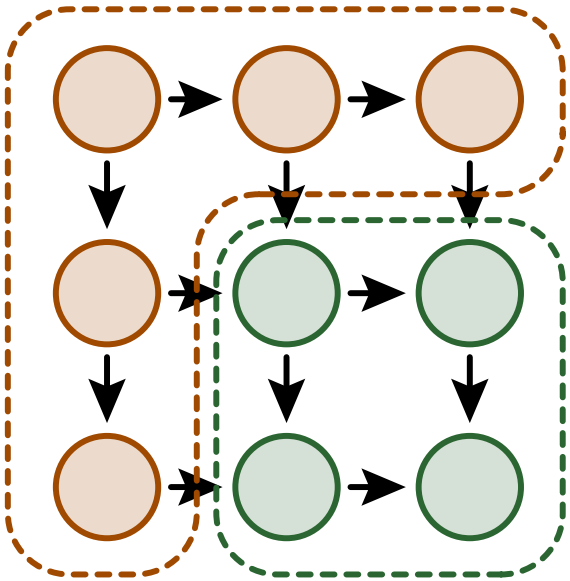
$$\theta_{ij} = \log p_{ij} - \log p_{i-1j} - \log p_{ij-1} + \log p_{i-1j-1}$$

Matrix balancing $\Leftrightarrow$

Satisfy $\eta_{i1} = \eta_{1i} = 3 - i + 1$

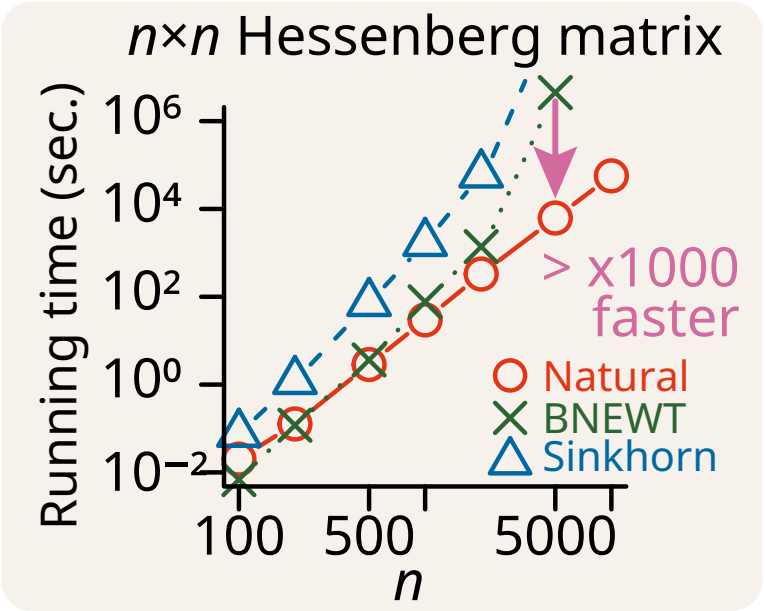
with keeping all θ_{ij}

Results of Natural Gradient

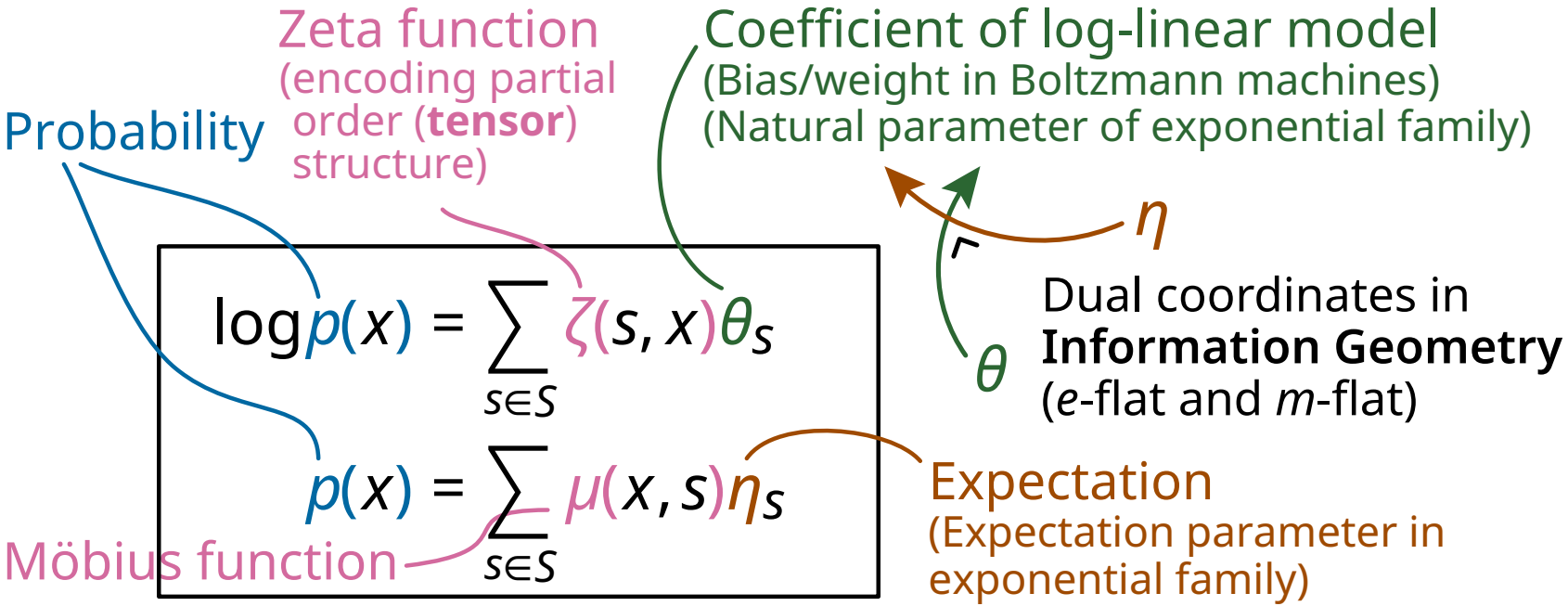


$\eta = 3 - i + 1$

$\theta = \hat{\theta}$



Summary



Slide: https://mahito.nii.ac.jp/pdf/DTU2026.pdf