November 28, 2025

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

Boltzmann Machines

Data Mining 05 (データマイニング)

Mahito Sugiyama (杉山麿人)

Today's Outline

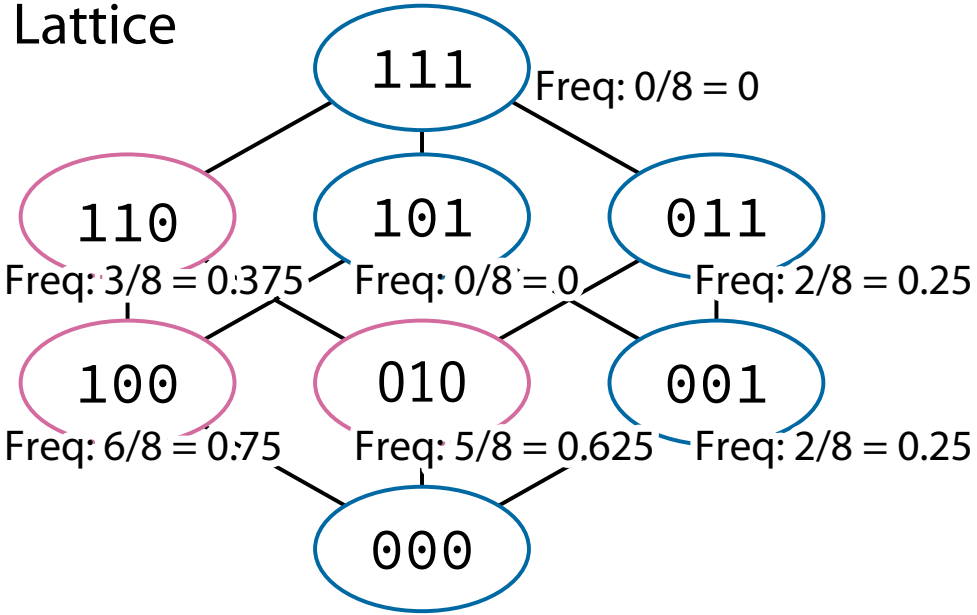
- Boltzmann Machines (Ising models):
  A fundamental probabilistic model of deep learning
  - Gibbs distribution
  - The learning equation
  - Gibbs sampling
- Relationship to the deep architecture
  - DBM (Deep Boltzmann Machines)

Learning Hierarchical Distribution (1/2)

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

Lattice

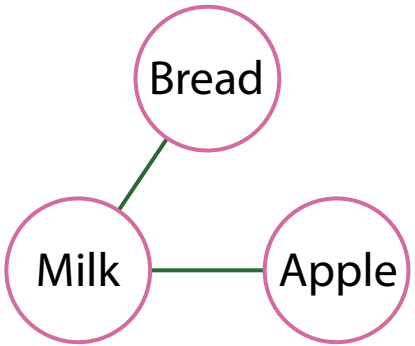


Learning Hierarchical Distribution (2/2)

MLE

$$\log(\text{prob.}) = -10.41 + 9.43[\text{Bread}] + 8.52[\text{Milk}] - 9.84[\text{Apple}] - 9.03[\text{Bread\&Milk}] + 9.43[\text{Milk\&Apple}]$$

Boltzmann machine



Bread	Milk	Apple	Prob. from data	Learned prob.
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

Boltzmann Machines

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

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

  - $x = (x_1, x_2, \dots, x_n) \in \{0,1\}^n$
  - $\theta = (\theta_1, \theta_2, \dots, \theta_n, \theta_{12}, \theta_{13}, \dots, \theta_{(n-1)n})$ 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 $p(x;\theta)$ is obtained for each $x \in \{0,1\}^n$ as

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

- $Z(\theta)$ is a partition function such that

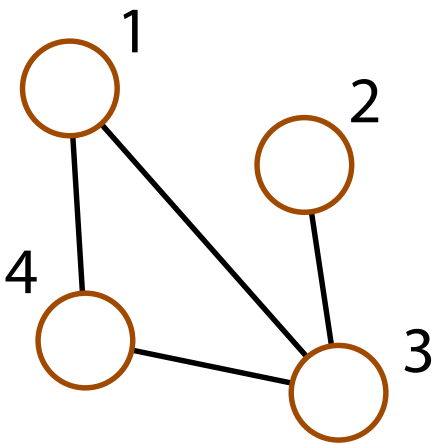
  $$Z(\theta) = \sum_{x \in \{0,1\}^n} \exp(-\Phi(x;\theta))$$

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

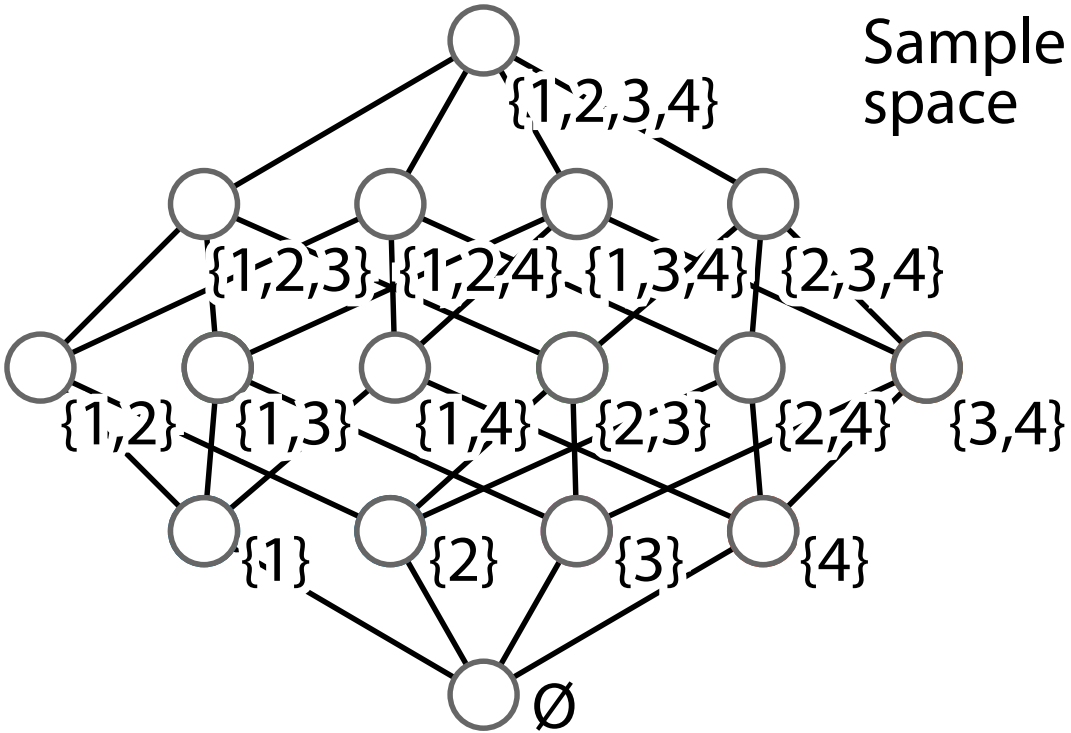
- The distribution P composed of $p(x)$ is called Gibbs distribution

Sample Space of BM

Boltzmann Machine (BM)

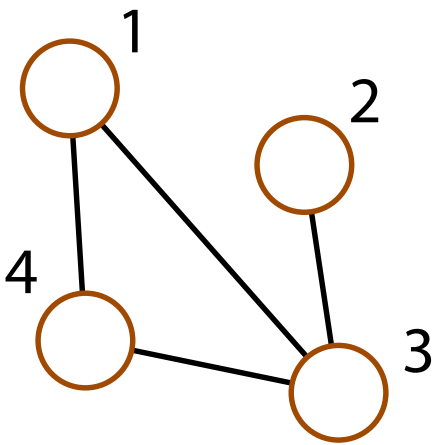


Sample space

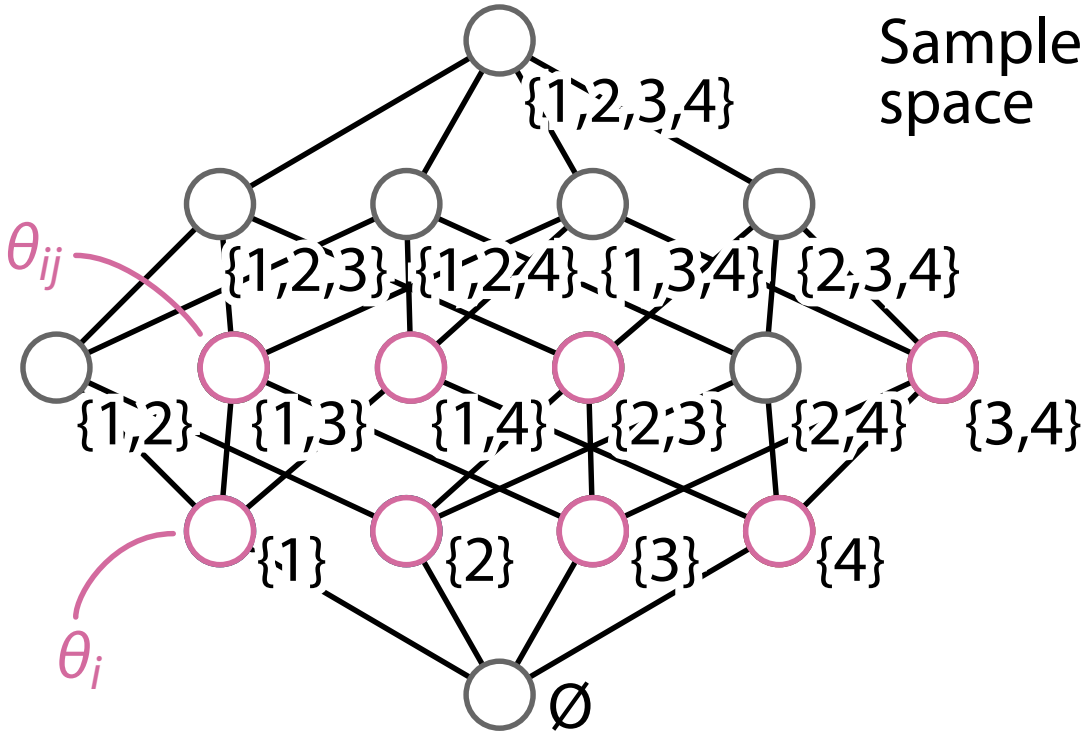


Parameters θ

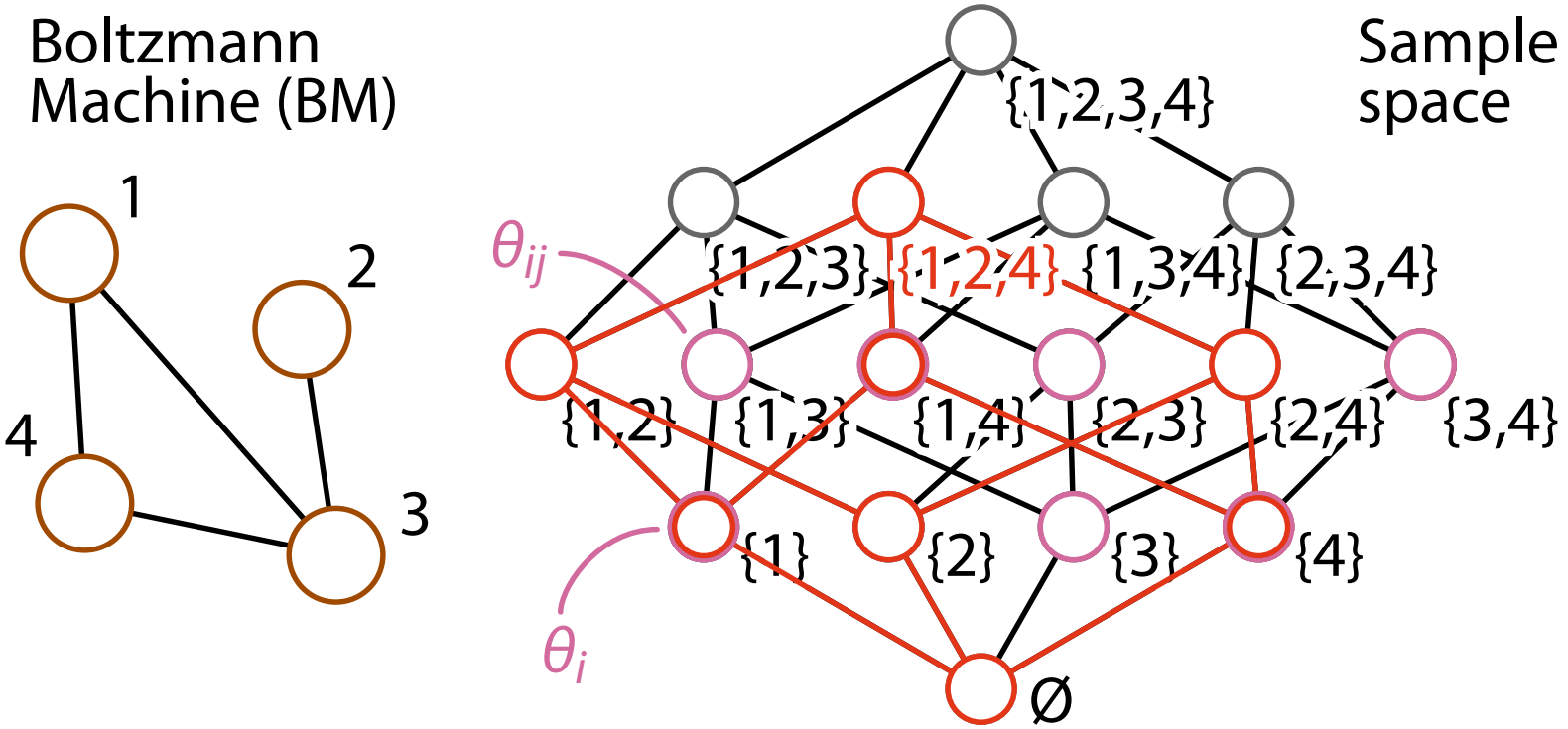
Boltzmann Machine (BM)



Sample space



Probability Computation



Learning of BM by MLE

- Given a dataset $D = \{x_{(1)}, x_{(2)}, \dots, x_{(N)}\}$, the objective of "learning" Boltzmann machines is to maximize the (log-)likelihood (Maximum Likelihood Estimation; MLE):

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

  - The probability of generating the given dataset by a BM

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

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

Gradient of θ

- Let us define

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

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

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

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

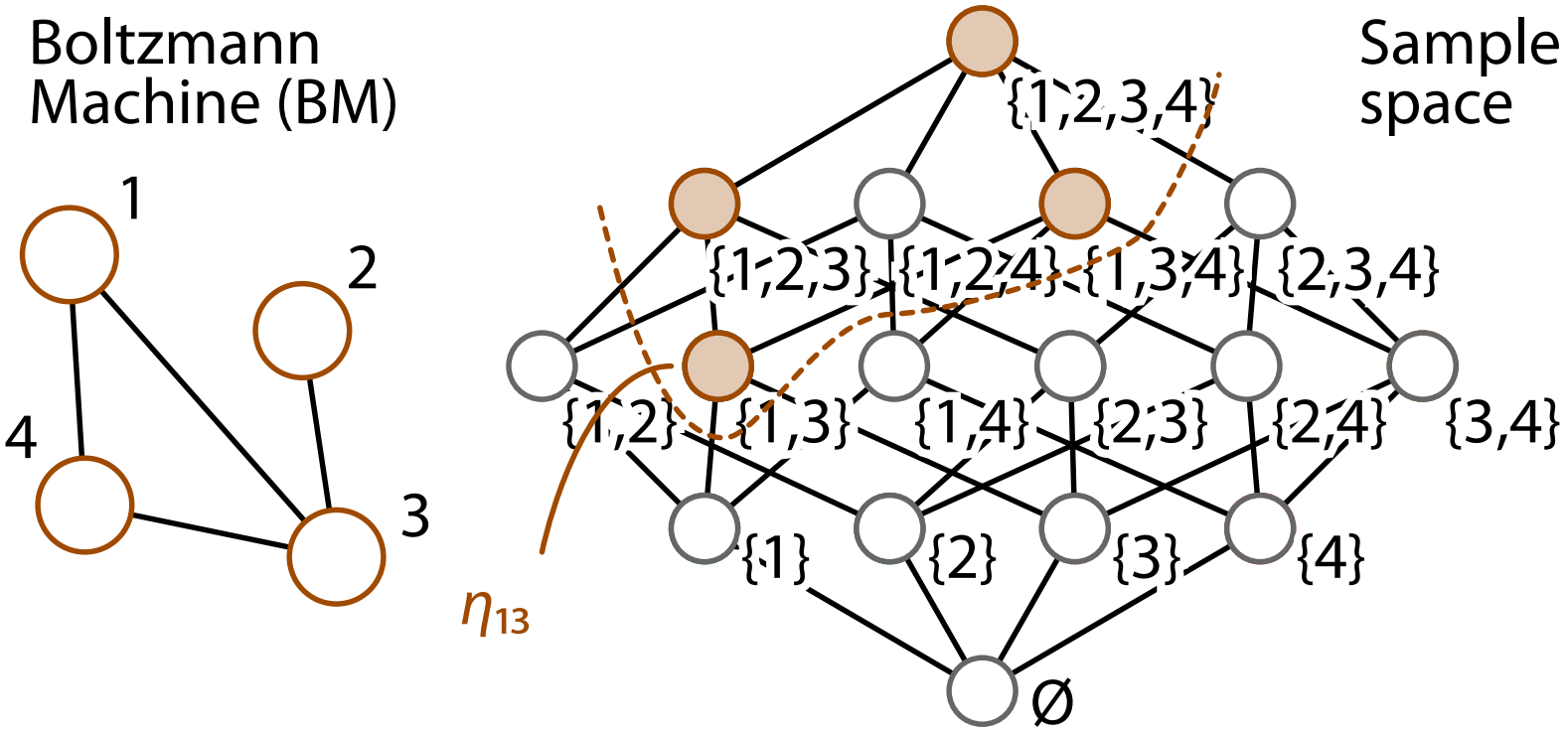
Learning Equation of BM

- $L_D(\theta)$ is maximized when the gradient is zero
- The gradient is zero $\iff$ For all θ_i, θ_{ij}

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

- This is known as the learning equation of BM

Frequency η



Empirical Frequency

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

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

  introduce the frequency used in pattern mining as

  $$\hat{\eta}_i = \frac{1}{N}|\{x_{(k)} \in D \mid x_{(k)i} = 1\}|$$

  $$\hat{\eta}_{ij} = \frac{1}{N}|\{x_{(k)} \in D \mid x_{(k)i} = x_{(k)j} = 1\}|$$

- 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 $L_D(\theta)$ 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

Optimization: Gradient Ascent

Algorithm 1: Learning of BM by gradient ascent

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

Combinatorial Explosion

- The serious problem of learning BMs: combinatorial explosion!!
- The time complexity of computing η_i:

  $$\eta_i = \sum_{x \in \{0,1\}^n} p(x;\theta)\mathbf{1}[x_i = 1]$$

  is $O(2^n)$, which is impossible to evaluate in practice
  - This is required to get the gradient $\hat{\eta}_i - \eta_i$
- **Solution:** Approximate it by Gibbs sampling

Gibbs Sampling (1/2)

- A Markov chain Monte Carlo (MCMC) algorithm
- We can generate samples from the current Gibbs distribution
  - n variables are dependent with each other
  - The partition function is not needed
- After obtaining enough sample $S = \{s_{(1)}, s_{(2)}, \dots, s_{(M)}\}$ by Gibbs sampling, each η can be approximated as

$$\eta_i \approx \frac{1}{M} \left| \left\{ s_{(k)} \in S \mid s_{(k)i} = 1 \right\} \right|,$$

$$\eta_{ij} \approx \frac{1}{M} \left| \left\{ s_{(k)} \in S \mid s_{(k)i} = s_{(k)j} = 1 \right\} \right|$$

Gibbs Sampling (2/2)

- For $x = (x_1, x_2, \dots, x_n)$, the conditional probability of the ith variable being x_i with fixing others is

$$
\begin{aligned}
p_i &= \frac{p(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)}{p(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) + p(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)} \\
&= \frac{\exp(\lambda_i x_i)}{1 + \exp(\lambda_i)}, \\
\lambda_i &= \theta_i + \sum_{j \neq i} \theta_{ij} x_j
\end{aligned}
$$

- Strategy: Perform sampling for each variable one by one using the above probability

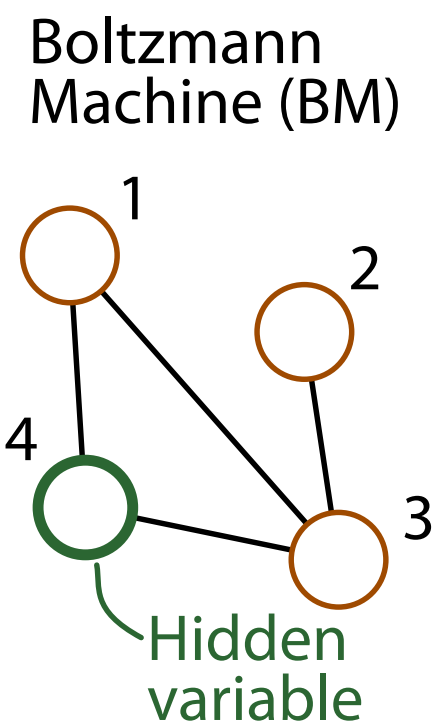
Algorithm 2: Gibbs Sampling

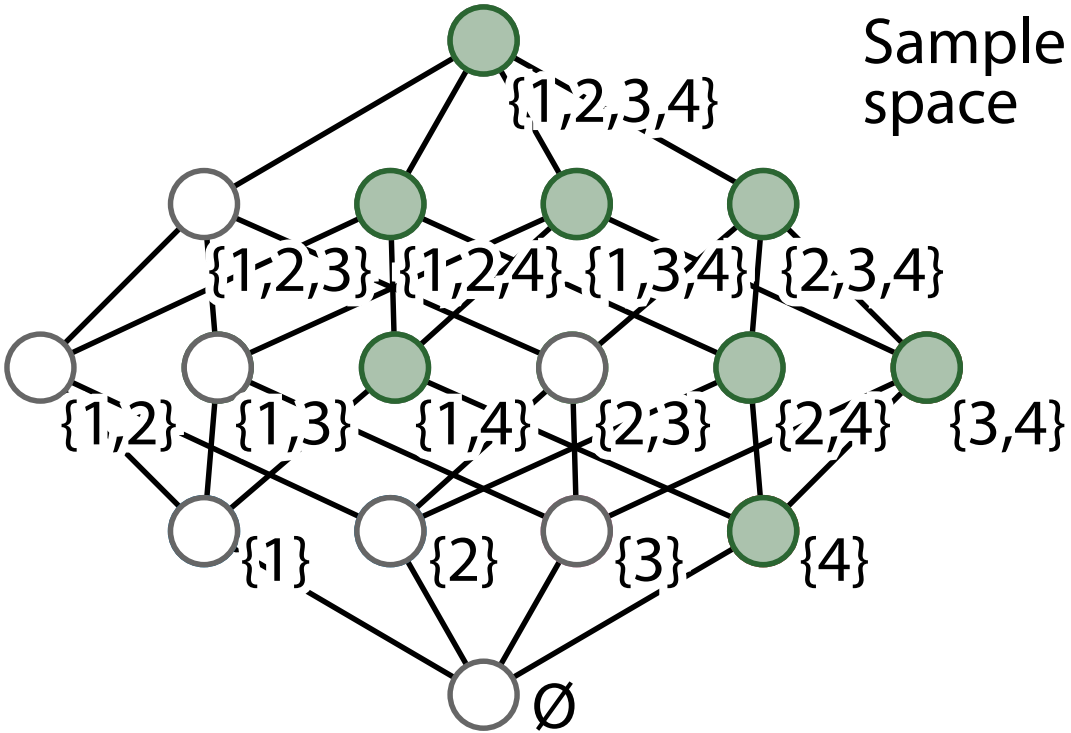
Initialize x with some values;
repeat
    foreach $i \in \{1, 2, \ldots, n\}$ **do**
        if $p_i \geq$ *random value* $u \in [0, 1]$ **then**
            $x_i \leftarrow 1$
        else
            $x_i \leftarrow 0$
    Output x and use it for the next initial vector
until *getting enough sample*;

Introducing Hidden Variables

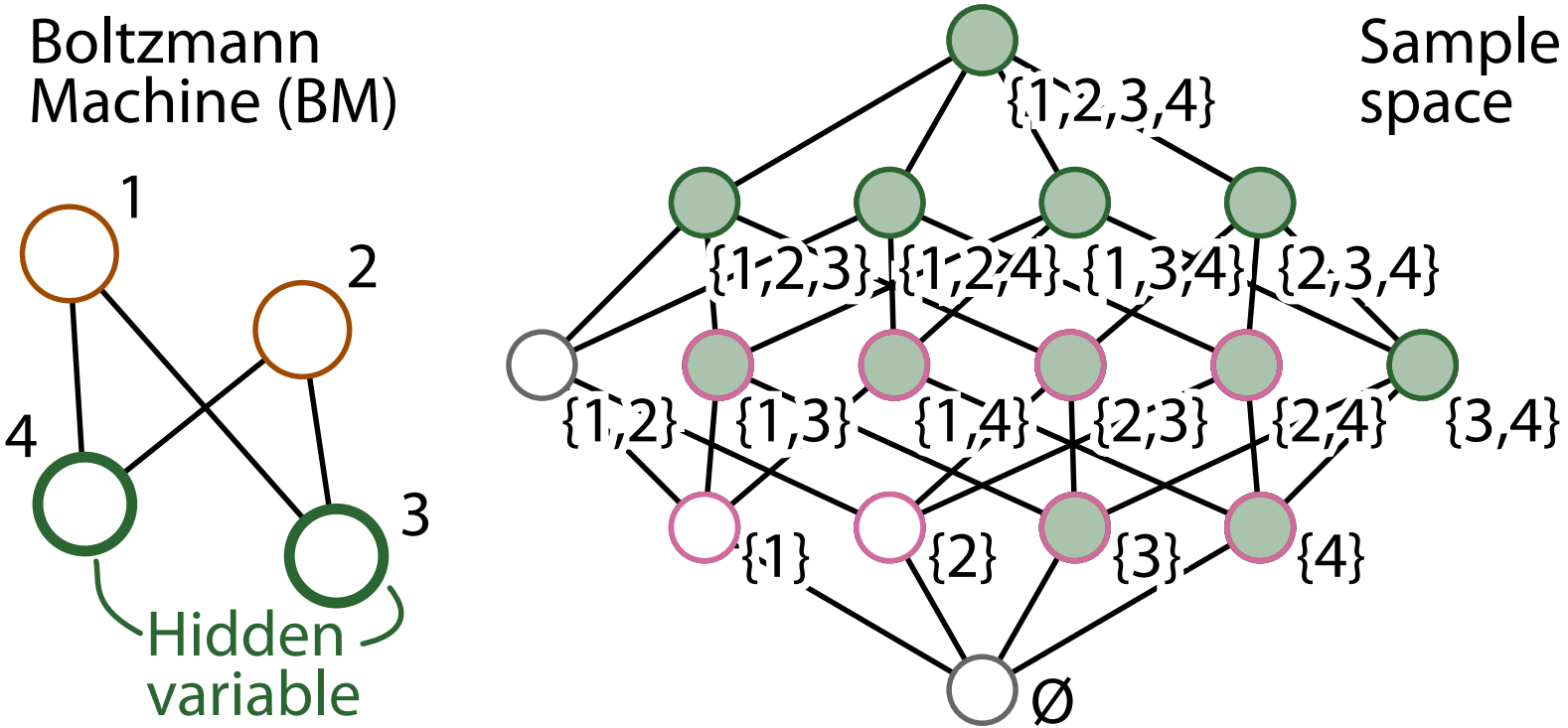
- To increase the representation power of BMs, we can introduce hidden variables
- When there are hidden nodes, the (log-)likelihood is maximized with respect to the distribution in which the hidden variables are marginalized out
- Let V and H be visible and hidden nodes

Outcome Space with Hidden Variable





Restricted Boltzmann Machines (RBMs)



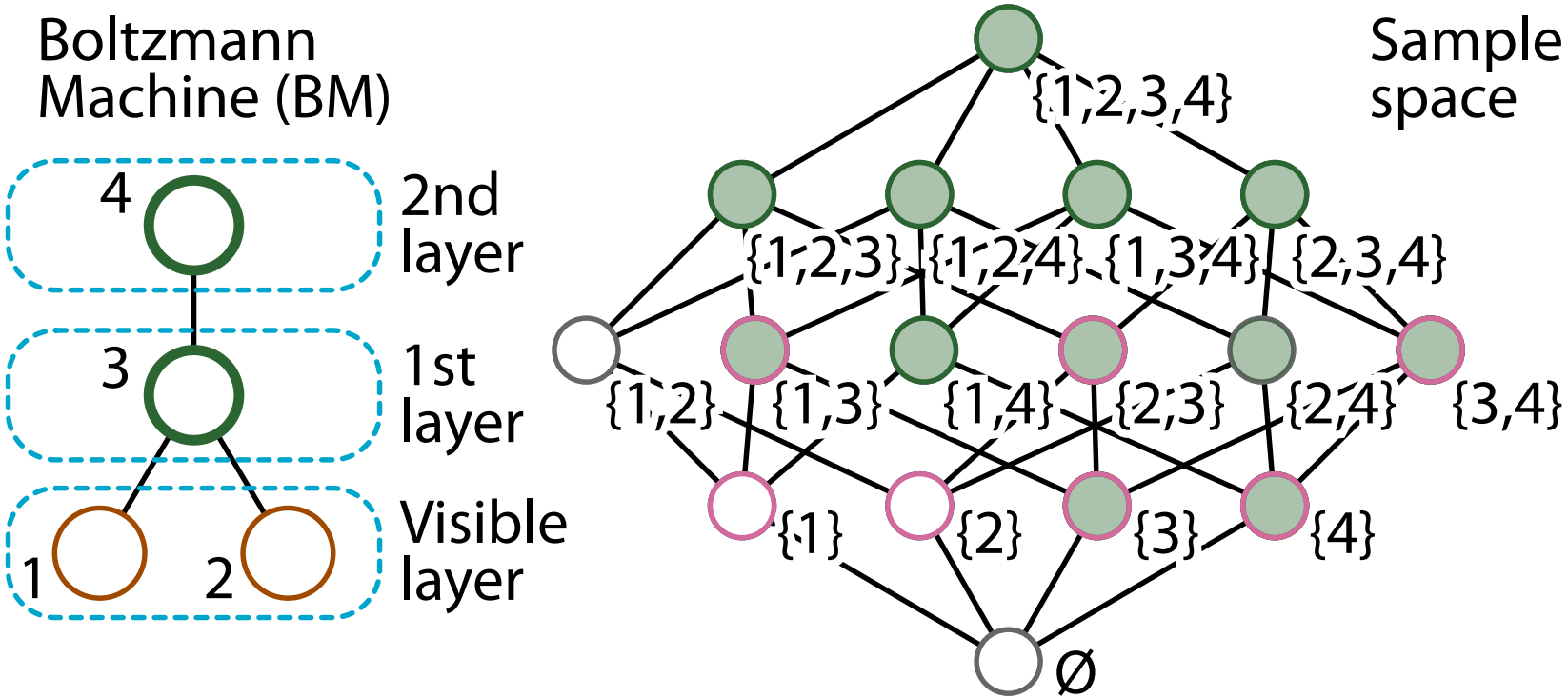
Learning of RBMs

- Given a dataset $D = \{v_{(1)}, \dots, v_{(N)}\}$, learning equations in RBMs are

$$\frac{1}{N}\sum_{k=1}^{N} v_{(k)i} = \eta_i \quad \text{(visible, } i \in V\text{)}$$

$$\frac{1}{N}\sum_{k=1}^{N} \mathrm{sig}(\lambda_{(k)j}) = \eta_j \quad \text{(hidden, } j \in H\text{)}$$

$$\frac{1}{N}\sum_{k=1}^{N} v_{(k)i}\mathrm{sig}(\lambda_{(k)j}) = \eta_{ij} \quad \text{(visible–hidden)}$$

$$\mathrm{sig}(\lambda_{(k)j}) = \frac{\exp(\lambda_{(k)j})}{1+\exp(\lambda_{(k)j})}, \quad \lambda_{(k)j} = \theta_j + \sum_i \theta_{ij} v_{(k)i}$$

Deep Boltzmann Machines (DBMs)