

November 24, 2023



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 (杉山磨人)

Today's Outline

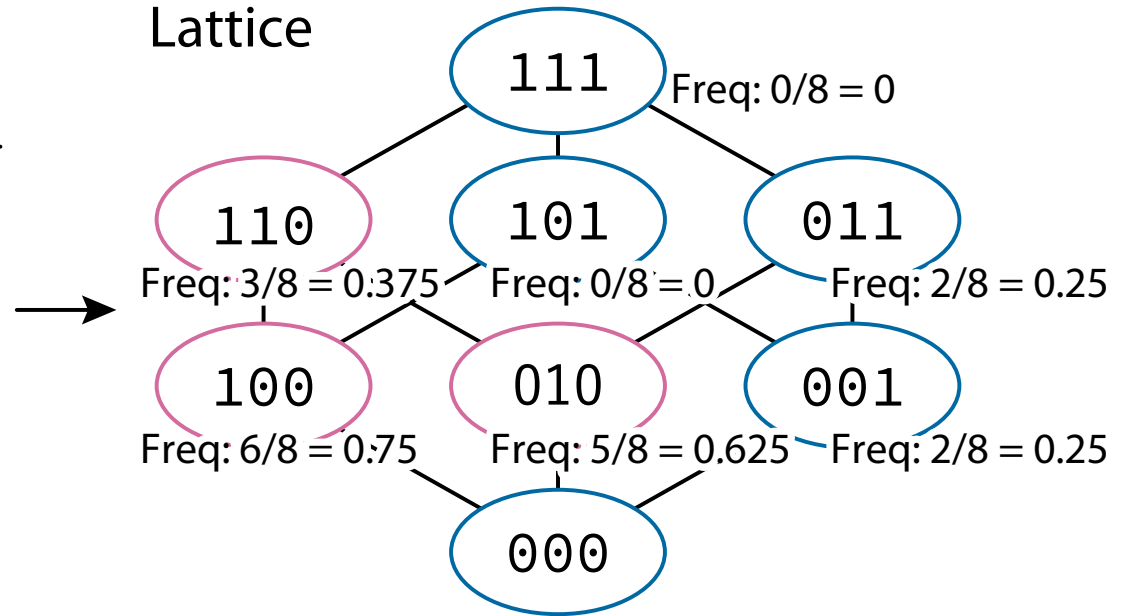
- Boltzmann Machines (Ising models):
A fundamental probabilistic model of deep learning
 - Gibbs distribution
 - The learning equation
 - Gibbs sampling
- Relationship to the deep architecture
 - DBM (Deep Boltzmann Machines)

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

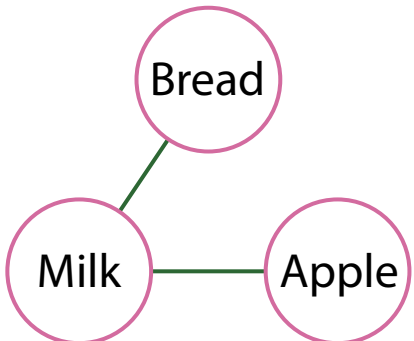


Learning Hierarchical Distribution (2/2)

MLE \rightarrow

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

Boltzmann machine



```
graph TD; Bread((Bread)) --- Milk((Milk)); Milk --- Apple((Apple));
```

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** $\Phi: \{0, 1\}^n \rightarrow \mathbb{R}$ of a BM G is defined as

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

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

Gibbs Distribution

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

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

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

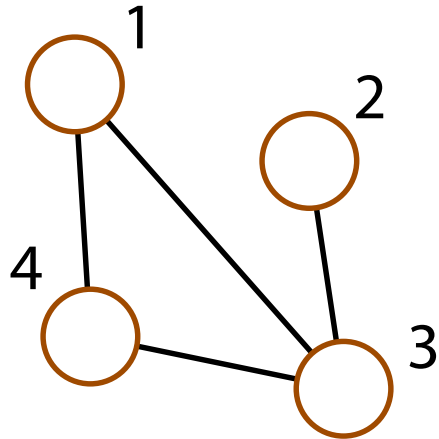
$$Z(\theta) = \sum_{\mathbf{x} \in \{0, 1\}^n} \exp(-\Phi(\mathbf{x}; \theta))$$

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

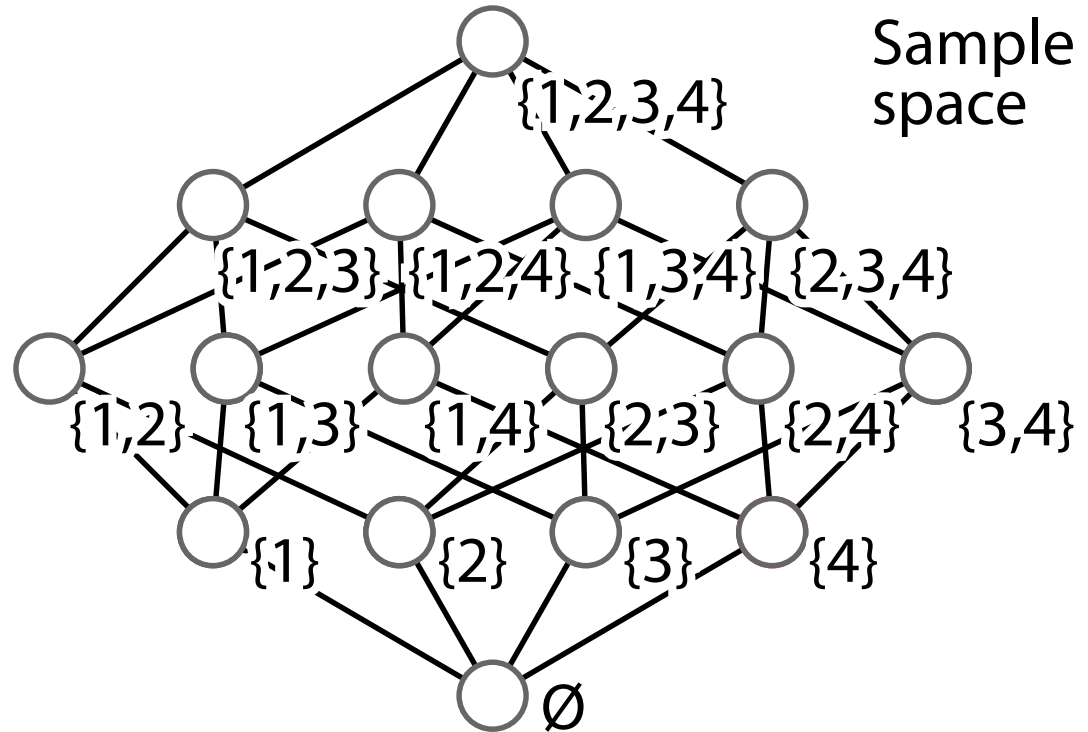
- The distribution P composed of $p(\mathbf{x})$ is called **Gibbs distribution**

Sample Space of BM

Boltzmann
Machine (BM)

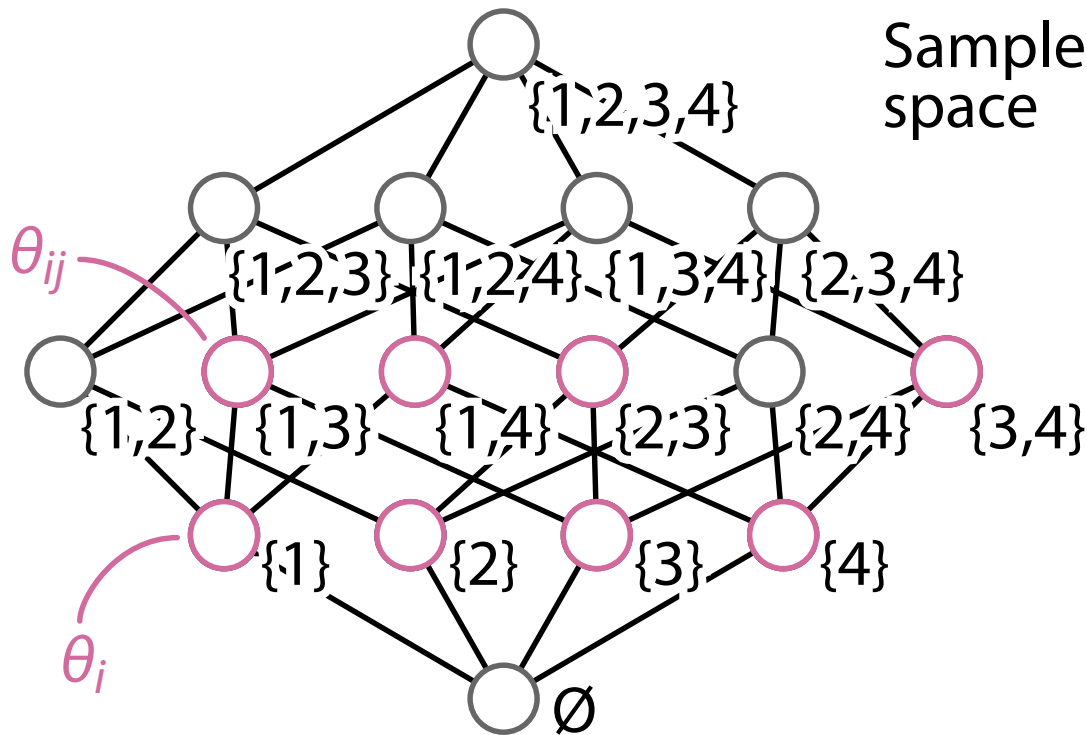
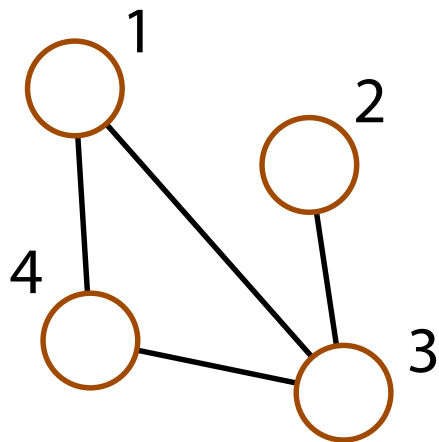


Sample
space



Parameters θ

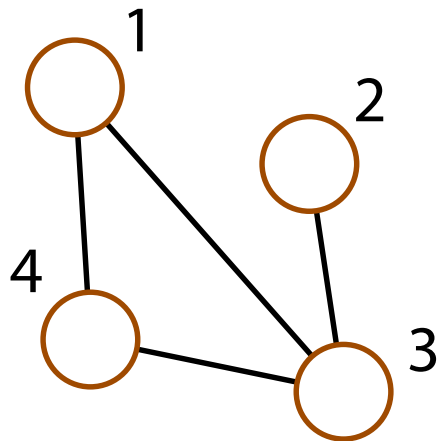
Boltzmann
Machine (BM)



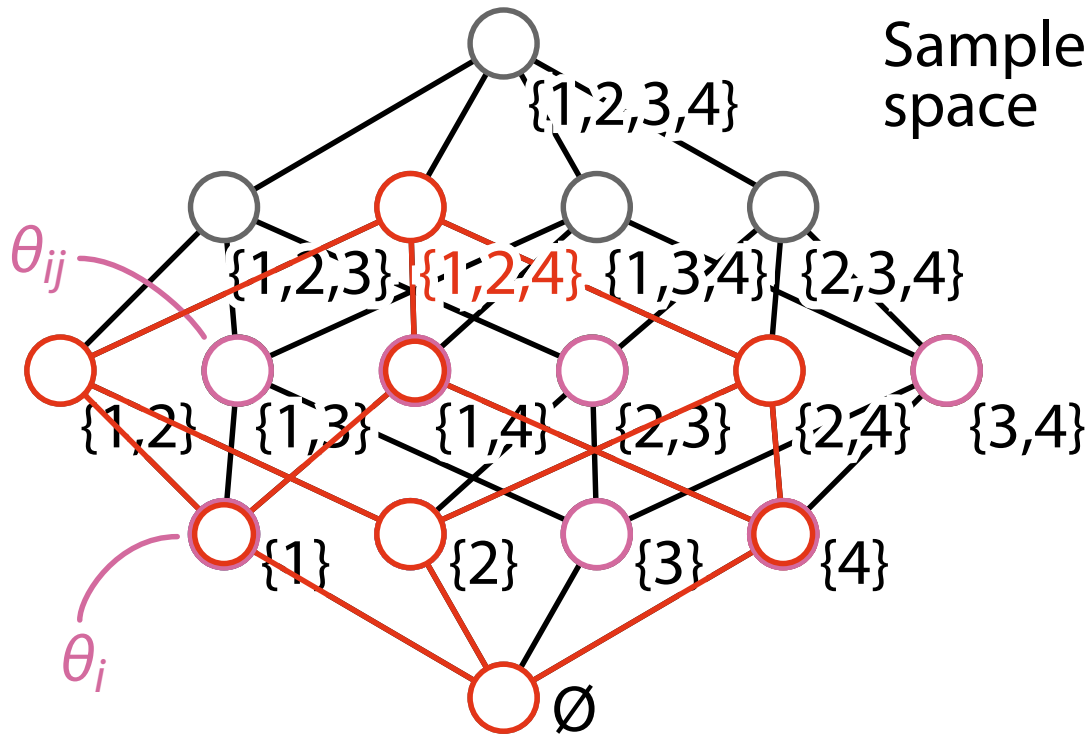
Sample
space

Probability Computation

Boltzmann Machine (BM)



Sample space



Learning of BM by MLE

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

Find θ that maximizes $\prod_{k=1}^N p(\mathbf{x}_{(k)}; \theta) = p(\mathbf{x}_{(1)}; \theta) \cdot p(\mathbf{x}_{(2)}; \theta) \cdot \dots \cdot p(\mathbf{x}_{(N)}; \theta)$

- The probability of generating the given dataset by a BM
- The log-likelihood is usually used:

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

Gradient of θ

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

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

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

where

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

Learning Equation of BM

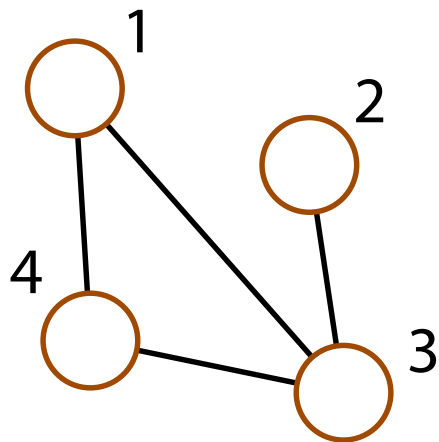
- $L_D(\theta)$ is maximized when the gradient is zero \iff For all θ_i, θ_{ij}

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

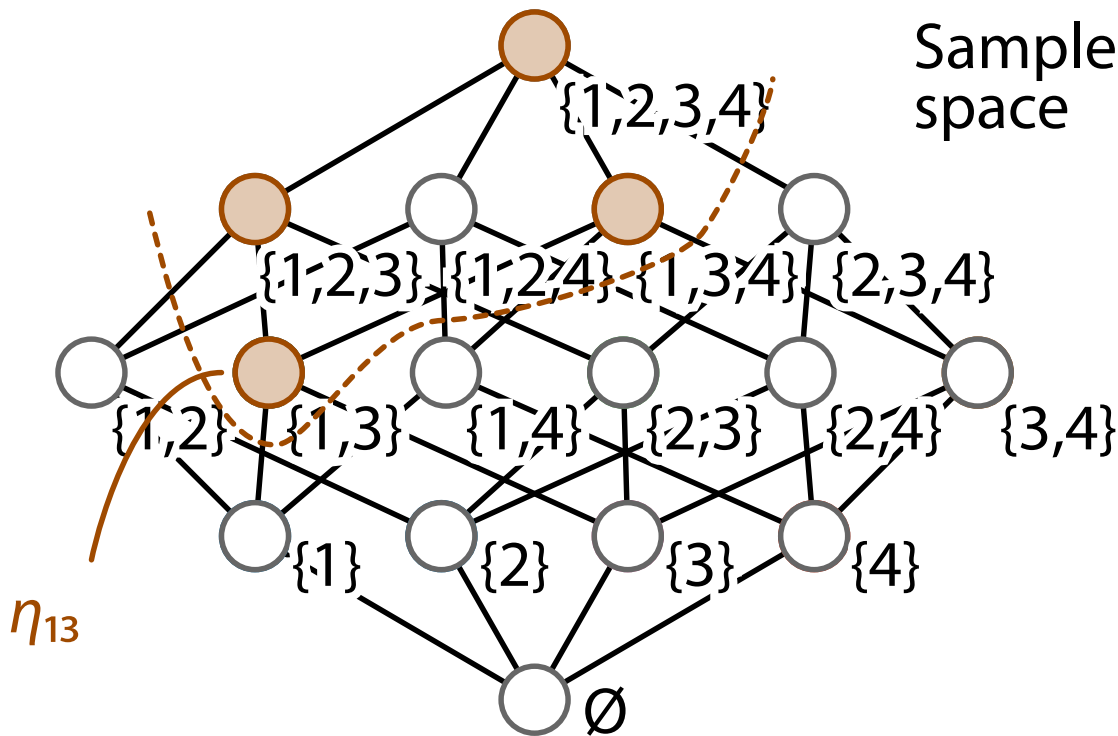
- This is known as **learning equation of BM**
- η coincides with the **frequency** used in itemset mining

Frequency η

Boltzmann
Machine (BM)



Sample
space



Empirical Frequency

- For the empirical distribution \hat{P} with

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

define

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

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

- The learning equation becomes

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

KL Divergence Minimization

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

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

- Maximizing the (log)likelihood is equivalent to minimizing the KL divergence: $\min_{P \in \mathcal{S}} D_{\text{KL}}(\hat{P}, P)$
 - \mathcal{S} : the set of Gibbs distributions

Optimization: Gradient Ascent

Algorithm 1: Learning of BM by gradient ascent

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

Combinatorial Explosion

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

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

is $O(2^n)$ and it is impossible to evaluate

- This is required to get the gradient $\hat{\eta}(x) - \eta(x)$
- 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 $S = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_M\}$ by Gibbs sampling, η_i can be approximated as

$$\eta_i \approx \frac{1}{M} |\{\mathbf{s} \in S \mid s_i = 1\}|,$$

$$\eta_{ij} \approx \frac{1}{M} |\{\mathbf{s} \in S \mid s_i = s_j = 1\}|$$

Gibbs Sampling (2/2)

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

$$p_i = \frac{p(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)}{p(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) + p(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)}$$
$$= \frac{\exp(\lambda_i x_i)}{1 + \exp(\lambda_i)},$$

$$\lambda_i = \theta_i + \sum_{j \neq i} \theta_{ij} x_j$$

Algorithm 2: Gibbs Sampling

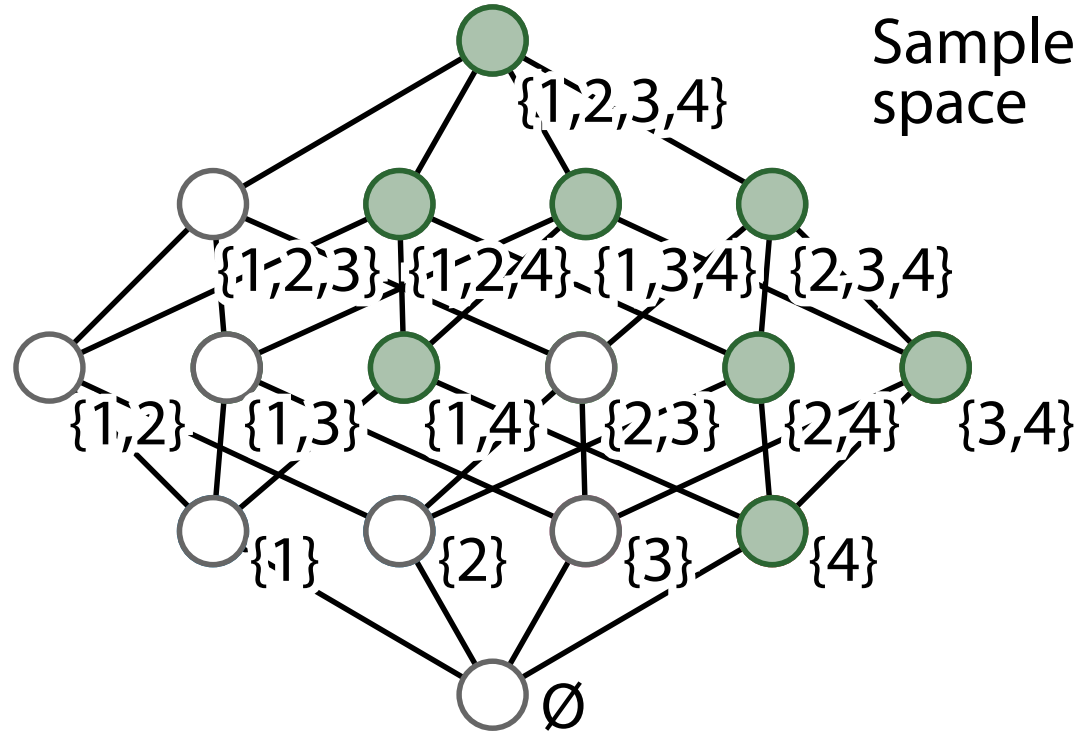
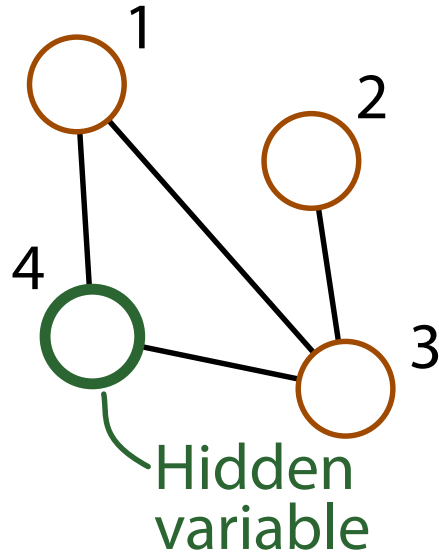
```
1 Initialize  $x$  with some values;  
2 repeat  
3   foreach  $i \in \{1, 2, \dots, n\}$  do  
4     if  $p_i \geq \text{random value } u \in [0, 1]$  then  
5        $x_i \leftarrow 1$   
6     else  
7        $x_i \leftarrow 0$   
8   Output  $x$  and use it for the next initial vector  
9 until 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

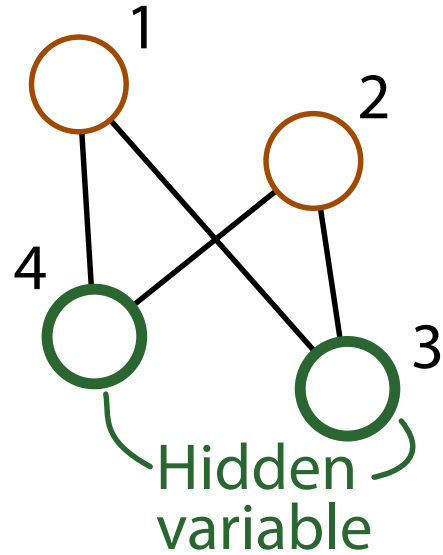
Outcome Space with Hidden Variable

Boltzmann Machine (BM)

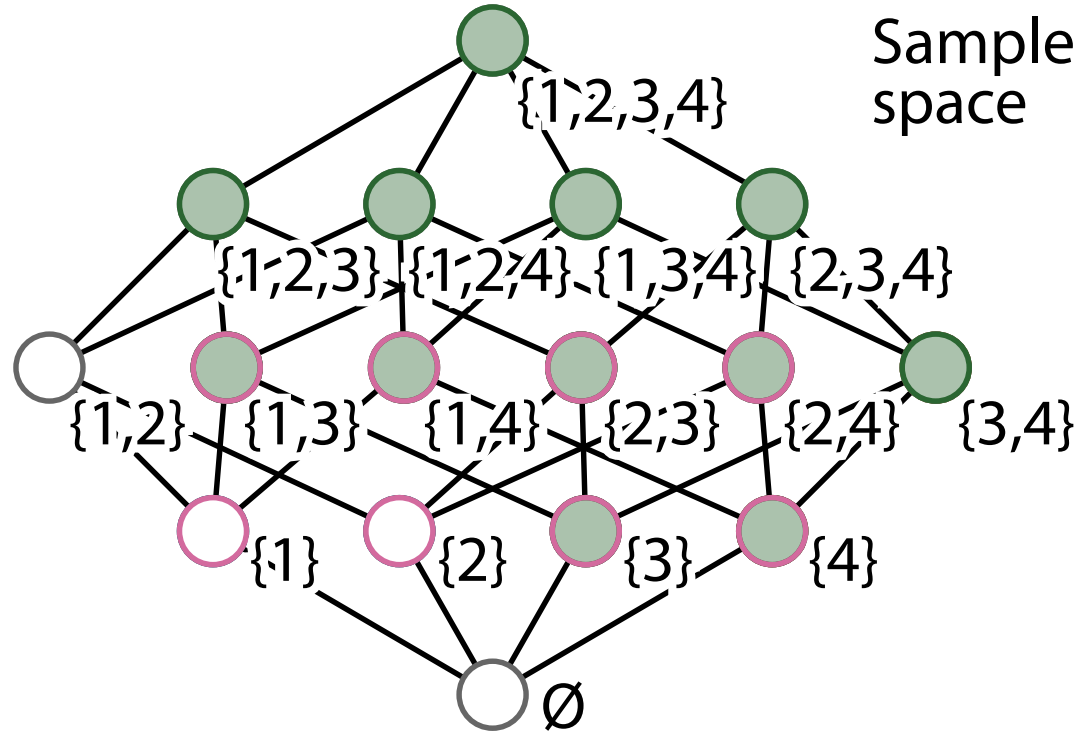


Restricted Boltzmann Machines (RBMs)

Boltzmann Machine (BM)



Sample space



Learning of RBMs

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

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

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

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

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

Deep Boltzmann Machines (DBMs)

