November 24, 2017

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
- Probabilistic models on Posets
  - Relationship with pattern mining
- Relationship to the deep architecture
  - DBM (Deep Boltzmann Machines)

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 $\Phi \colon \{0, 1\}^n \to \mathbb{R}$ of a BM G is defined as

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

  - $\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$
  - $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n, \theta_{12}, \theta_{13}, \dots, \theta_{n-1n})$ 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 $\{i, j\} \notin E$

Boltzmann Distribution

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

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

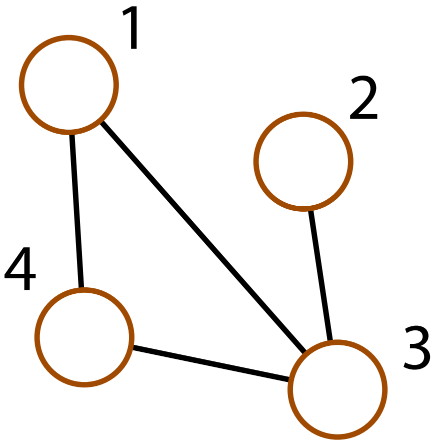
- $Z(\boldsymbol{\theta})$ is a partition function such that

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

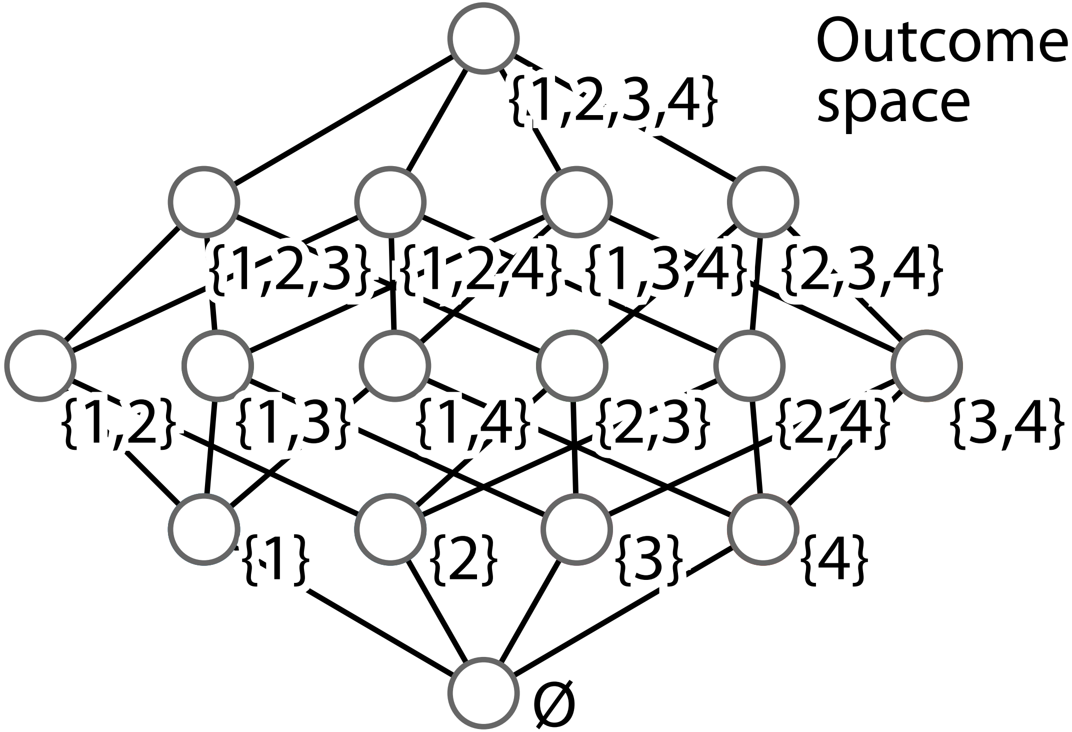
to ensure the condition $\sum_{\boldsymbol{x}\in\{0,1\}^n} p(\boldsymbol{x}) = 1$

Outcome Space of BM

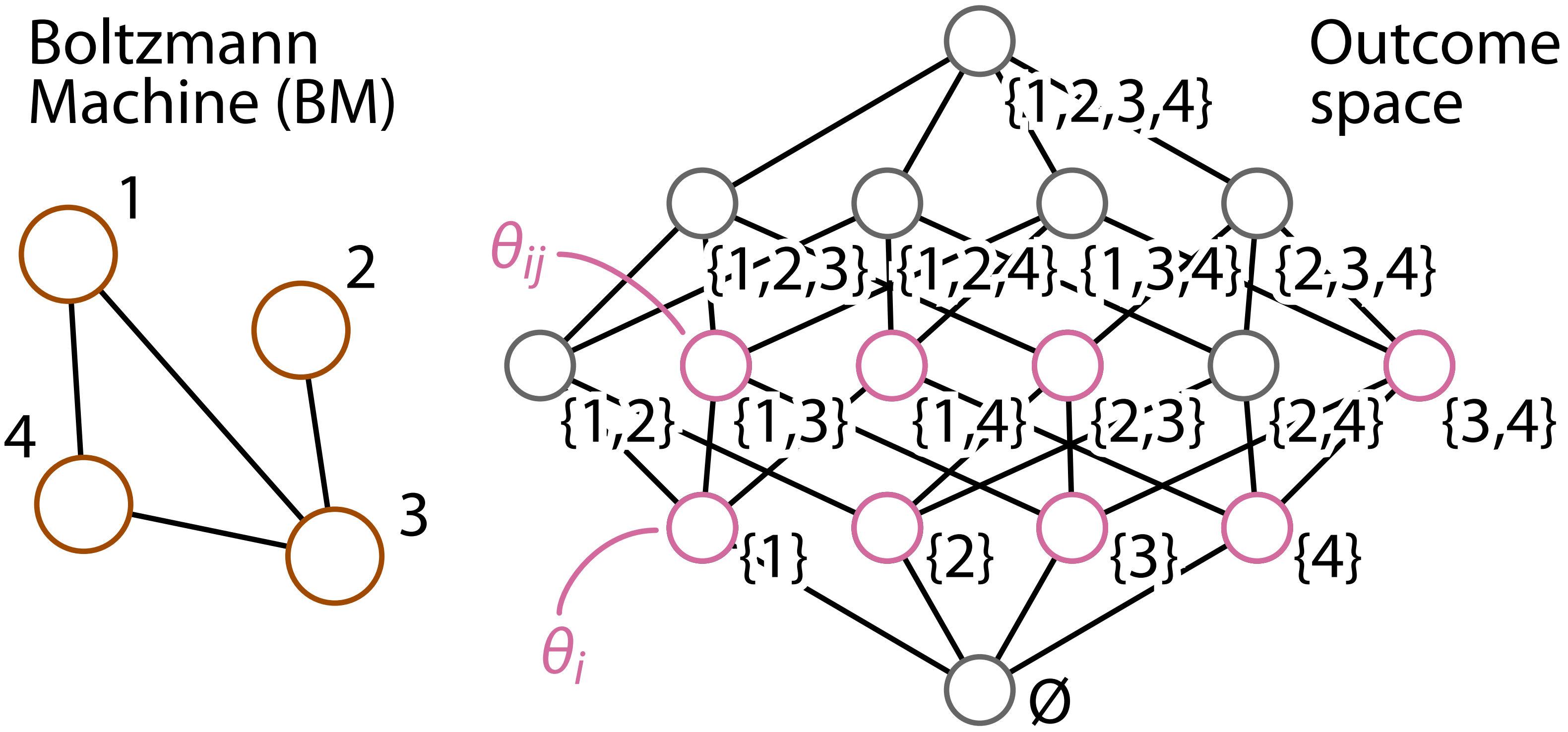
Boltzmann Machine (BM)



Outcome space

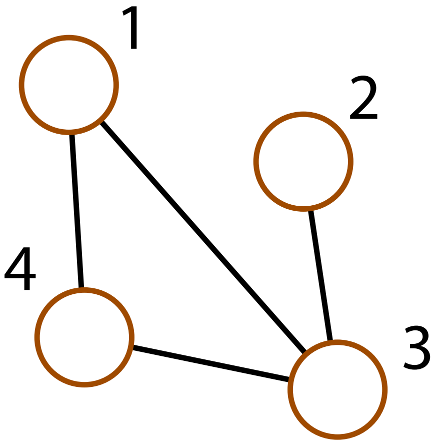


Parameters θ

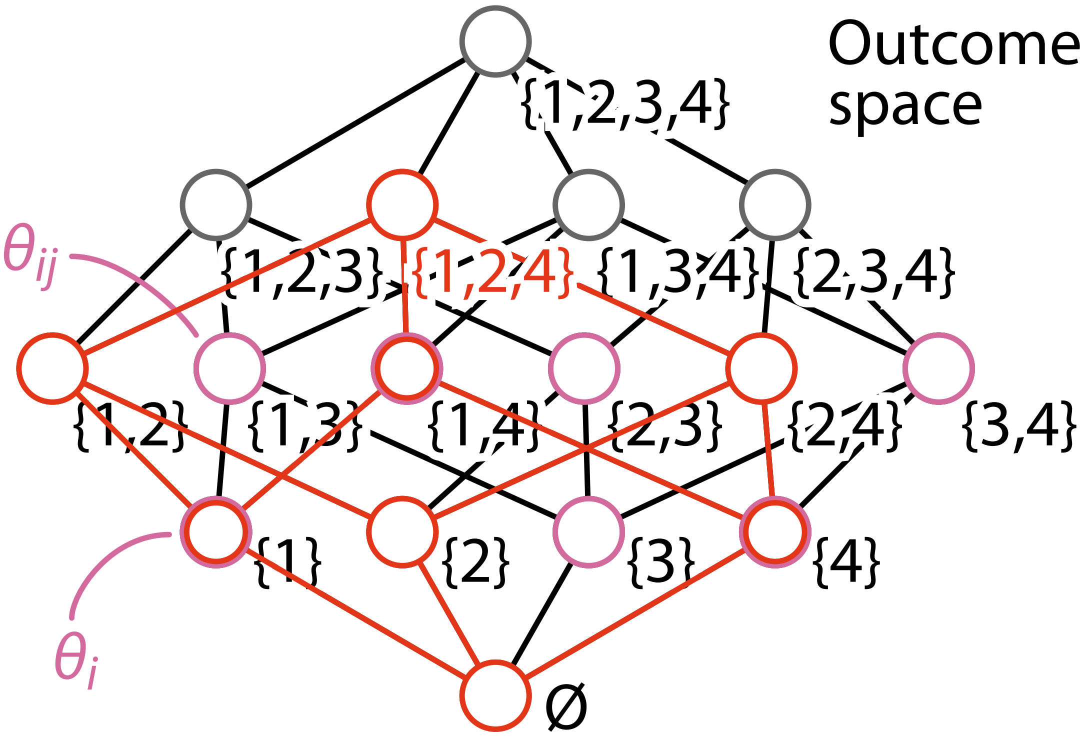


Probability Computation

Boltzmann
Machine (BM)



Outcome
space



Log-Linear Model

- BM is a special case of the log-linear model in statistics
- Let each set $x \in 2^V$ be the set of indices of "1" of $\boldsymbol{x} \in \{0, 1\}^n$
- The parameter set $B = \{\, x \in 2^V \setminus \{\varnothing\} \mid |x| = 1 \text{ or } x \in E \,\}$
- The Boltzmann distribution is described as
  $$\log p(x) = \sum_{s \subseteq x} \theta(s) - \psi(\theta), \qquad \psi(\theta) = -\theta(\varnothing) = \log Z(\boldsymbol{\theta})$$
  - Log of the probability is obtained by the linear combination of coefficients θ
  - $\theta(s) = 0$ if $s \notin B$

Learning of BM by MLE

- Given a dataset $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the objective of learning Boltzmann machines is to maximize the (log-)likelihood (Maximum Likelihood Estimation; MLE)

  $$\text{Find } \boldsymbol{\theta} \text{ that maximizes } \prod_{i=1}^{N} p(\boldsymbol{x}_i; \boldsymbol{\theta}) = p(\boldsymbol{x}_1; \boldsymbol{\theta}) \cdot p(\boldsymbol{x}_2; \boldsymbol{\theta}) \cdot \cdots \cdot p(\boldsymbol{x}_N; \boldsymbol{\theta})$$

  - The probability of generating the given dataset by a BM

- The log-likelihood is usually treated

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

Gradient of θ

- The log-likelihood is

$$L_D(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{s \subseteq x_i} \theta(s) - N\psi(\theta)$$

- The gradient of $\theta(s)$ for each $s \in B$ is obtained as

$$\frac{\partial L_D(\boldsymbol{\theta})}{\partial \theta(s)} = |\{x_i \in D \mid s \supseteq x_i\}| - N\,\eta(s),$$

$$\eta(s) = \sum_{x \supseteq s} p(x) = \begin{cases} \mathbf{E}_{\boldsymbol{\theta}}[s^i] = \sum_{\boldsymbol{s}} s^i p(\boldsymbol{s}; \boldsymbol{\theta}) = \Pr(s^i = 1) \\ \mathbf{E}_{\boldsymbol{\theta}}[s^i s^j] = \sum_{\boldsymbol{s}} s^i s^j p(\boldsymbol{s}; \boldsymbol{\theta}) = \Pr(s^i = 1 \text{ and } s^j = 1) \end{cases}$$

Learning Equation of BM

- $L_D(\boldsymbol{\theta})$ is maximized when the gradient is zero $\iff$

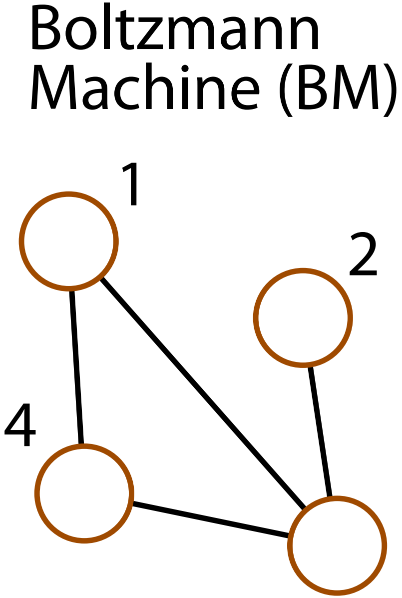
  $$\frac{1}{N}|\{x_i \in D \mid s \supseteq x_i\}| = \eta(s)$$

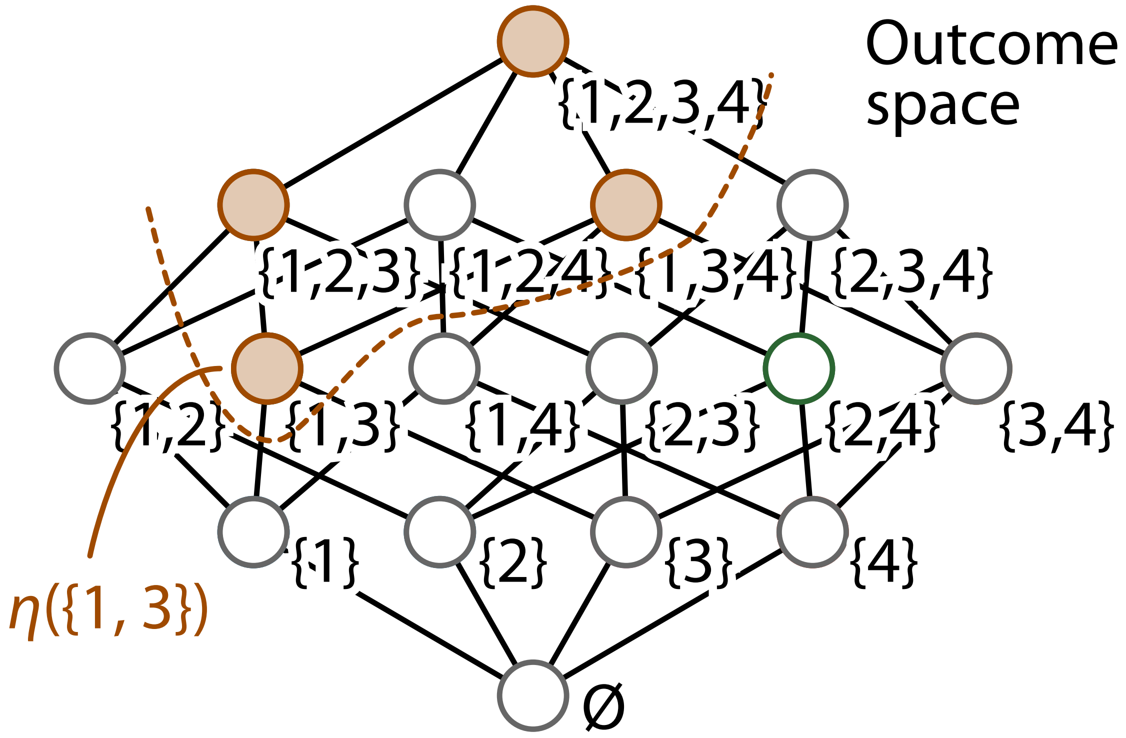
  $$\hat{\eta}(s) = \eta(s)$$

  for all $s \in B$

  - This is known as learning equation of BM
  - $\hat{\eta}(s)$ coincides with the frequency of a pattern s used in itemset mining

Frequency η





KL Divergence Minimization

- Given two distributions P, Q, the KL divergence from P to Q:

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

- Given a dataset $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the empirical distribution $\hat{P}$ is

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

- Maximizing the (log)likelihood is equivalent to minimizing the Kullback–Leibler (KL) divergence: $\min_{P \in \mathcal{S}(B)} D_{\mathrm{KL}}(\hat{P}, P)$
  - $\mathcal{S}(B)$: the set of Boltzmann distributions

Optimization: Gradient Ascent

Algorithm 1: Learning of BM by gradient ascent

Initialize $\boldsymbol{\theta}$ with some values;
$t \leftarrow 0$;
repeat
    foreach $s \in B$ **do**
        $\theta^{(t+1)}(s) \leftarrow \theta^{(t)}(s) + \varepsilon\big(\hat{\eta}(s) - \eta(s)\big)$;
    $t \leftarrow t + 1$
until $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t+1)}$;

Combinatorial Explosion

- The serious problem of learning BMs: combinatorial explosion!!
- The time complexity of computation of $\eta(x)$:
  $$\eta(x) = \sum_{s \supseteq x} p(s),$$
  is $2^{O(n)}$ 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 Boltzmann distribution
  - n variables are dependent with each other
  - The partition function is not needed

- After obtaining an enough sample $S = \{s_1, s_2, \ldots, s_M\}$ by Gibbs sampling, $\eta(s)$ can be approximated as

$$\eta(s) \approx \frac{1}{M}|\{s_i \in S \mid s \supseteq s_i\}|$$

Gibbs Sampling (2/2)

- For $\boldsymbol{x} = (x^1, x^2, \ldots, x^n)$, the conditional probability of the $i$the variable being x^i with fixing others is

$$p_i = \frac{p(x^1, \ldots, x^{i-1}, x^i, x^{i+1}, \ldots, x^n)}{p(x^1, \ldots, x^{i-1}, 0, x^{i+1}, \ldots, x^n) + p(x^1, \ldots, x^{i-1}, 1, x^{i+1}, \ldots, x^n)}$$

$$= \frac{\exp(\lambda_i x_i)}{1 + \exp(\lambda_i)},$$

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

Algorithm 2: Gibbs Sampling

```
Initialize x with some values;
repeat
    foreach i ∈ {1, 2, . . . , n} do
        if p_i ≥ a random value u ∈ [0, 1] then
            x^i ← 1
        else
            x^i ← 0
        Output x and use it for the next initial vector
until getting enough sample;
```

Introducing Hidden Variables

- To increase the representation power of the BM, we can introduce hidden variables

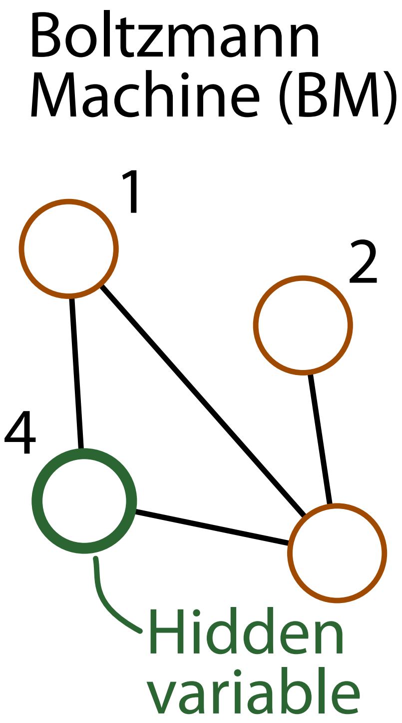
- When there are hidden nodes, the (log-)likelihood is maximized with respect to the distribution in which the hidden variables are marginalized out

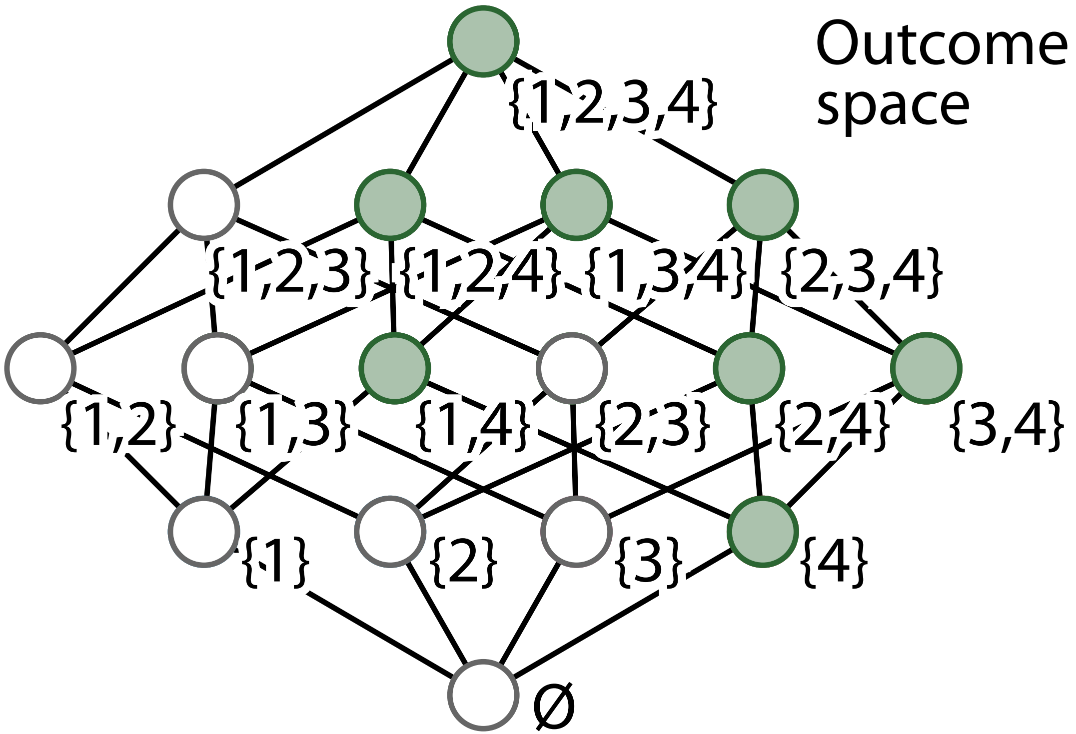
- Let V and H be visible and hidden nodes

- The learning equation for $x = v \cup h$ with $v \subseteq V$ and $h \subseteq H$ is

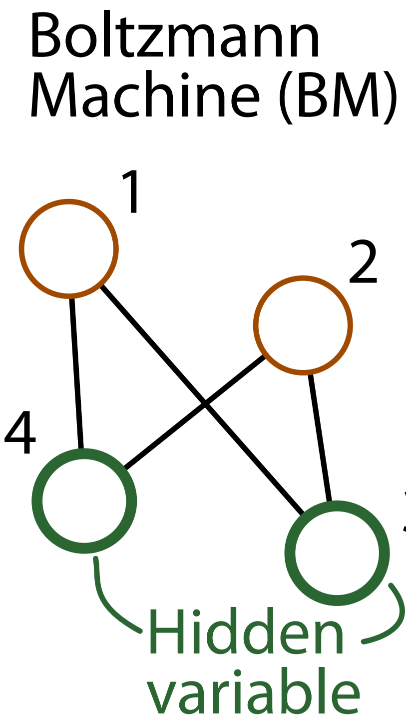
$$\sum_{s \supseteq v, h} p(h \mid s)\hat{p}(s) = \eta(x)$$

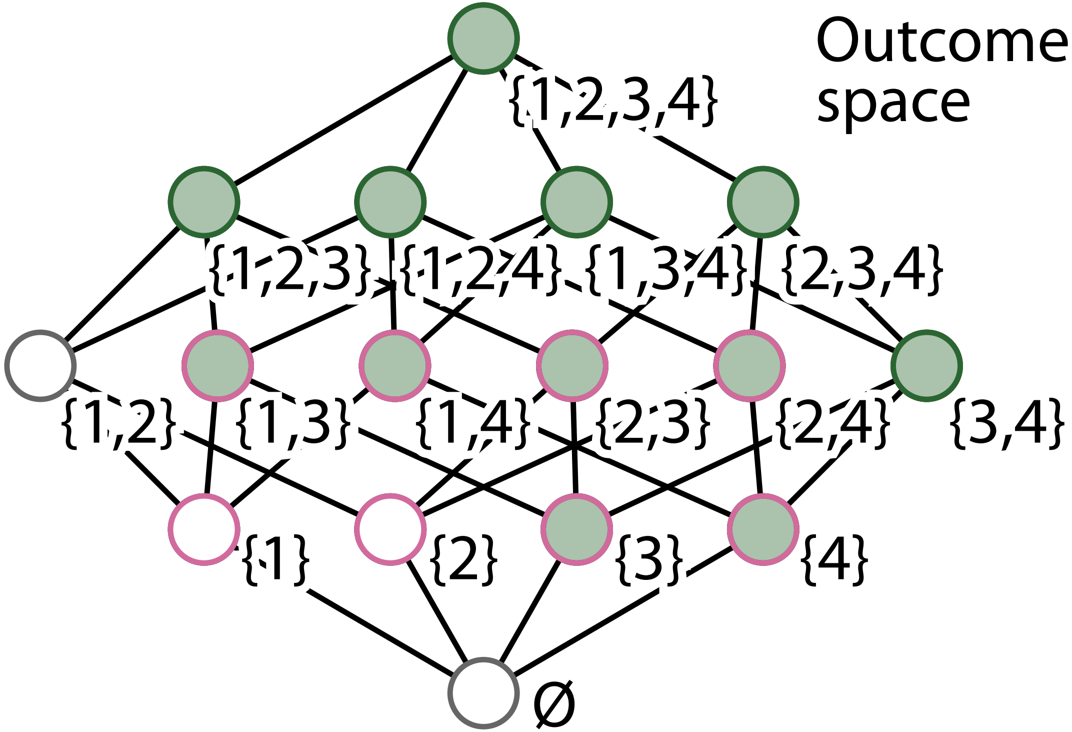
Outcome Space with Hidden Variable





Restricted Boltzmann Machines (RBMs)





Learning of RBMs

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

$$\hat{p}(v) = \eta(v)$$

$$\frac{1}{N}\sum_{\mu=1}^{N} \mathrm{sig}(\lambda^{j}_{(\mu)}) = \eta(h)$$

$$\frac{1}{N}\sum_{\mu=1}^{N} v^{i}_{(\mu)}\mathrm{sig}(\lambda^{j}_{(\mu)}) = \eta(v \cup h), \quad \text{where}$$

$$\mathrm{sig}(\lambda^{j}_{(\mu)}) = \frac{\exp(\lambda^{j}_{(\mu)})}{1 + \exp(\lambda^{j}_{(\mu)})}, \quad \lambda^{j}_{(\mu)} = \theta_j + \sum_{i \in V} \theta_{ij} v^{i}_{(\mu)}$$

Deep Boltzmann Machines (DBMs)