CS 461: Machine Learning Principles

Class 24 Dec 2

Restricted **B**oltzmann **M**achine (RBM)

Instructor: Diana Kim

Outline

1. Review : Various ML methods Learning a Joint Probability Density

2. Undirected Graphical Probabilistic Modeling (aka Markov Random Fields: MRF)

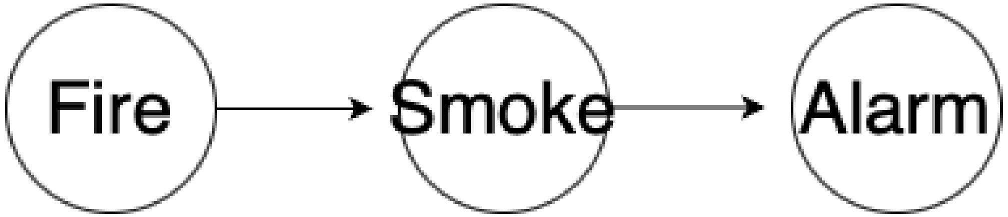
3. Energy Based Models (EBM)

4. Restricted Boltzmann Machine (RBM)
    - Special architecture based on MRF & EBM
      making Training/Prediction/ Sampling efficient

5. RBM applications: Netflix Recommendation System (Collaborative Filtering)

Review: We studied the various ML methods for learning a joint prob density. depending on our data and target problems, we choose a proper method.

[1] Bayesian Network (Directed Graphical Methods)

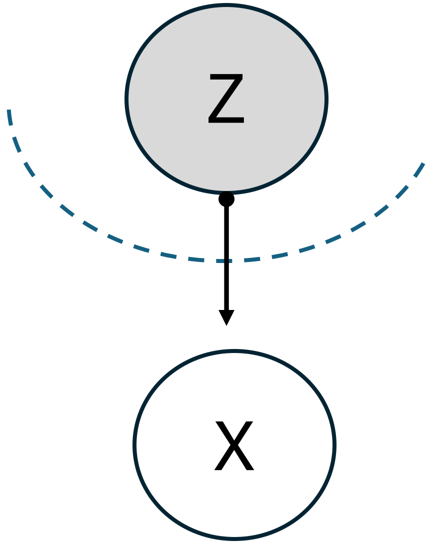


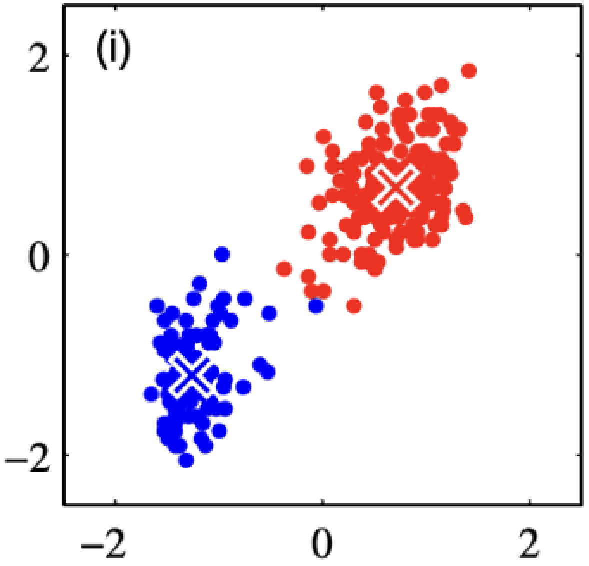
Q: P[Fire|Alarm]?

- useful and natural to representing a causal and effect relationship among R.Vs.
- The graphical structure encodes conditional independence (CI).
- by using the CI, the posterior queries can be computed efficiently.

[2] Gaussian Mixture Modeling (GMM)

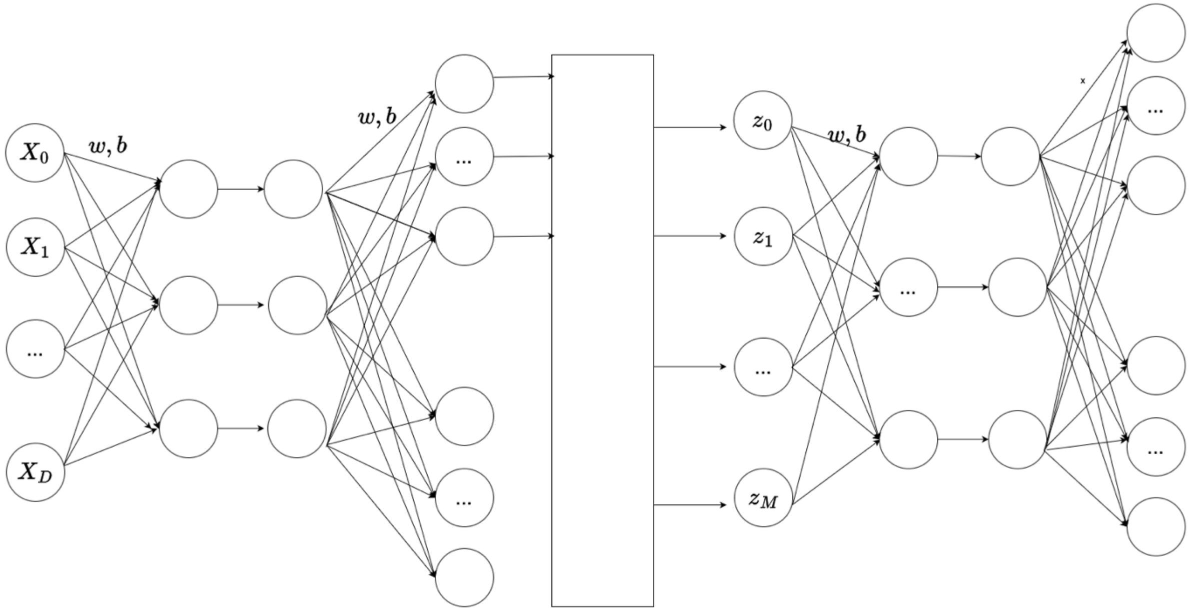
Bayesian Network adopting Latent Variables

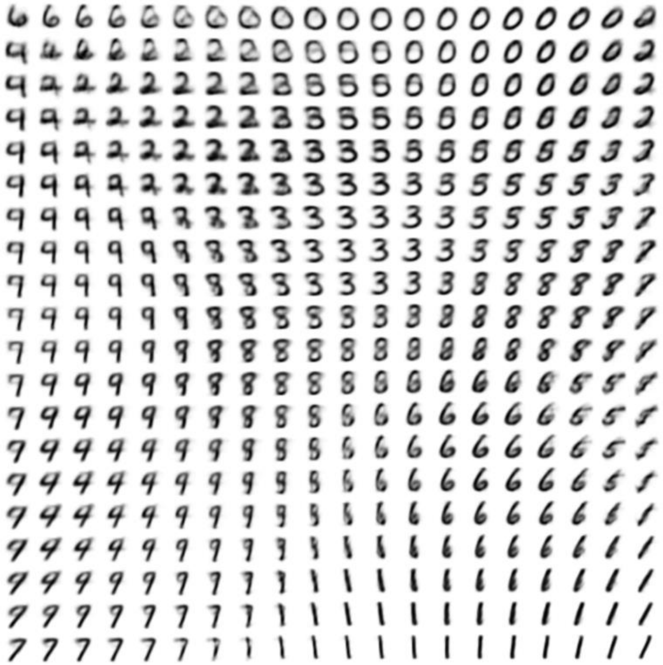




- useful for representing the density of continuous R.V with multiple modes.
- clustering: assigning a label to each data sample by using the posterior density $P[Z = k \mid x]$

[3] Variational Auto Encoder (VAE)





- useful to discover the continuous latent *Z* space
- (revelation of latent factors/ structures)
- data compression and new data sample generation from Z space.

Possible tasks with a joint probability density extends far beyond classification and regression, which are enough with discriminative models like $E[f(y|x)]$ or $P[C_k|x]$

What we can do with a joint probability density:

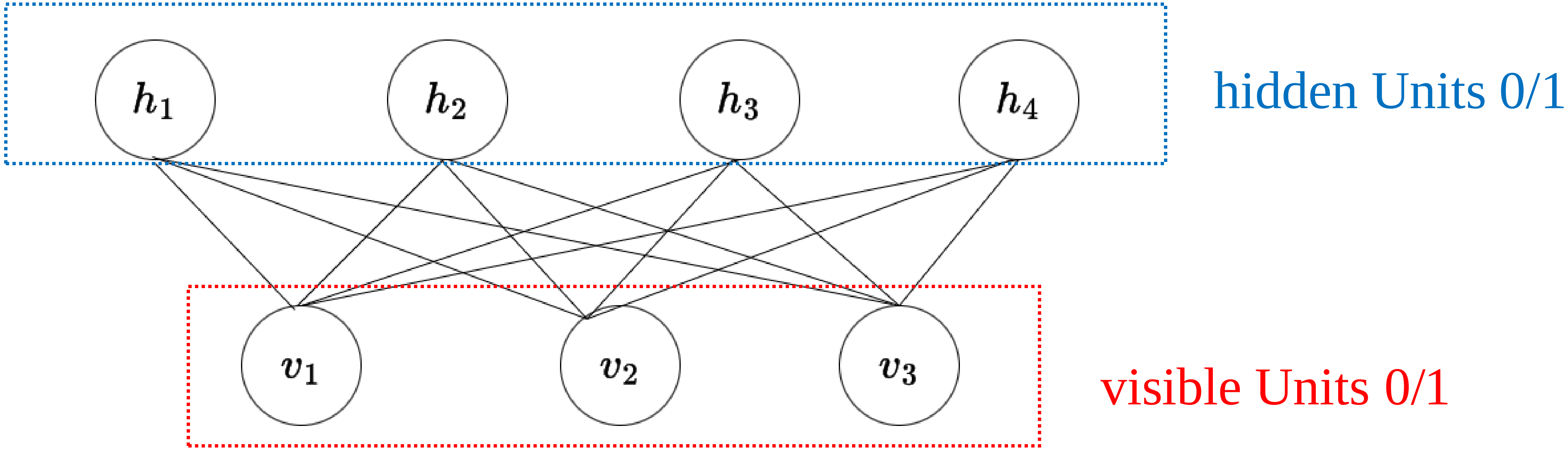
1. density estimation

2. new sample generation

3. missing value information: posterior

4. denoising

Today we are going to learn a new joint probability modeling
Which is called Restricted Boltzmann Machine (RBM)



[Example Structure of RBM]

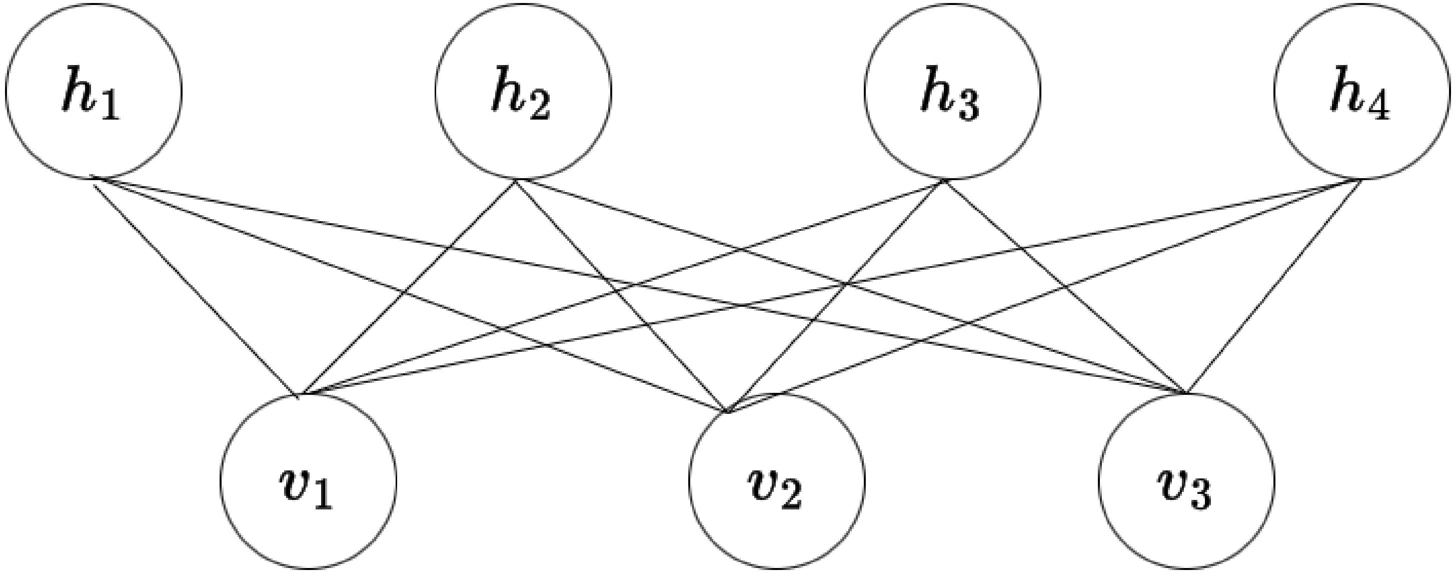
This is a generative ML for

(1) Learning Latent Space,

(2) New Sample Generation

(3) Efficient Posterior Computation (from visible to hidden and vice versa)

The structural constraints in RBM provide conditional properties that facilitate probabilistic ML tasks: Learning, Inference, Sampling



The conditional independence significantly simplifies the computation, making RBMs practical for tasks like sampling, inference, and learning.

$$P(h_1, h_2, h_3, h_4 | v_1, v_2, v_3) = \prod_{n=1}^{N=4} P(h_n | v_1, v_2, v_3)$$

$$P(v_1, v_2, v_3 | h_1, h_2, h_3, h_4) = \prod_{n=1}^{N=3} P(v_n | h_1, h_2, h_3, h_4)$$

Two Characteristic of RBM

(1) no DAG: RBM is a undirected graphical model; it is called **M**arkov **R**andom **F**ield

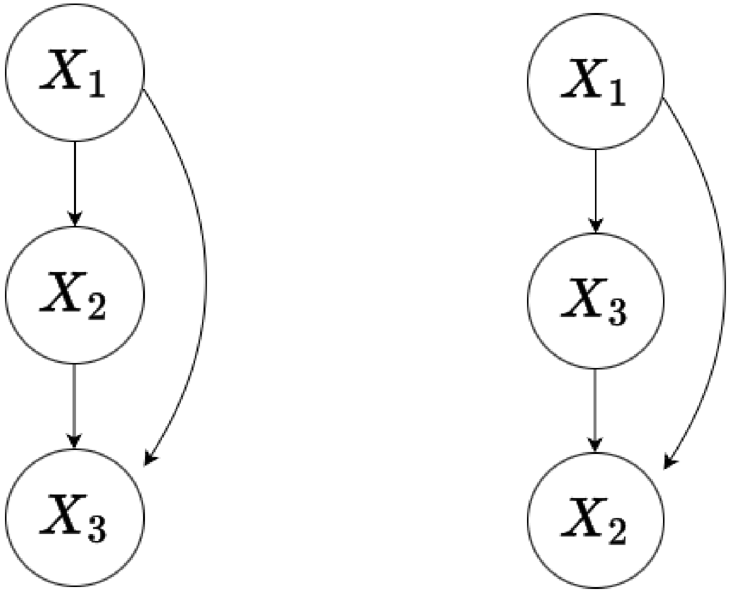
(2) Without a pre-defined structure/ specific parametric forms like Gaussian / Cauchy

: the density is free from, a function like $f(X,Y) = \frac{1}{Z} e^{AX+BY+C}$

it is called **E**nergy **B**ased **M**odel (EBM)

Undirected **G**raphical **M**odel (MRF)

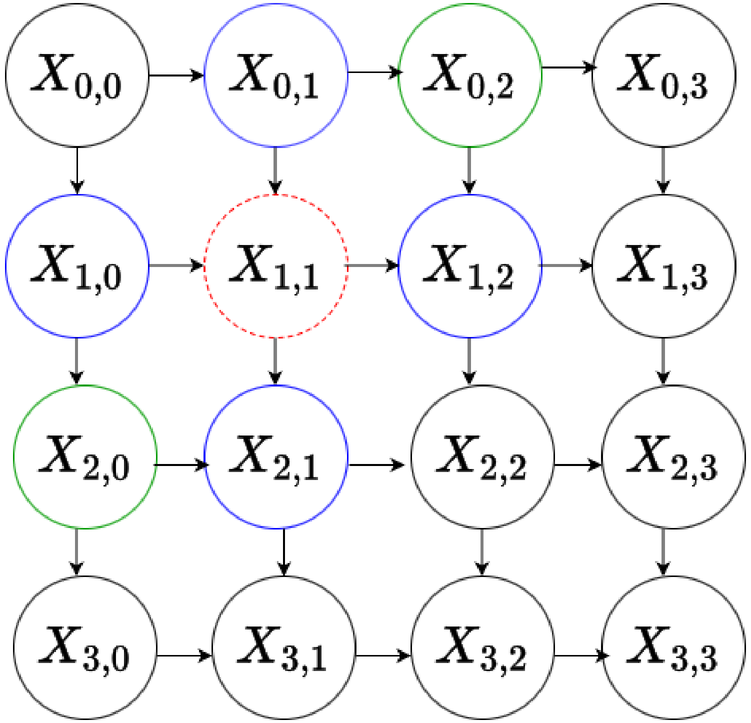
We know that a joint density can be represented by different Bayesian networks with varying structures and orders. However, if we know two R.Vs A and B have a causal relationship, it is natural to represent the structure with a specific direction.



$$\begin{aligned} P(X_1, X_2, X_3) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \\ &= P(X_1)P(X_3|X_1)P(X_2|X_1, X_3) \end{aligned}$$

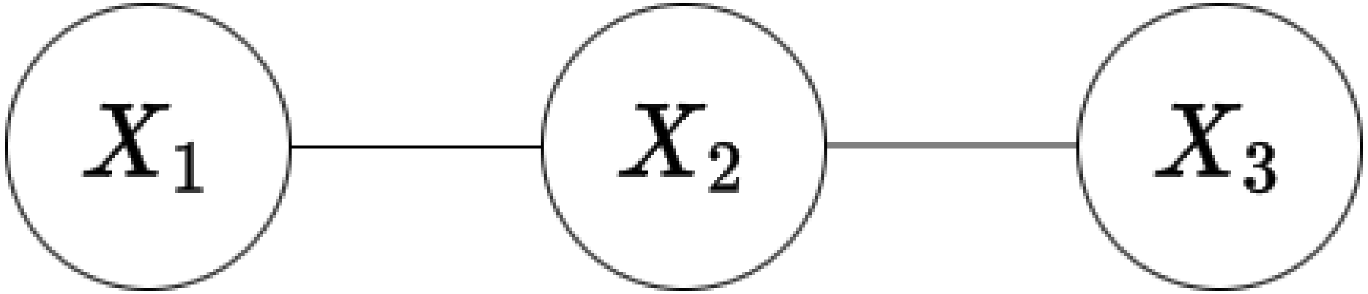
[Two different ways to represent the same density]

And what if we don’t have a clear direction to the interactions among the variables? How about pixels in images?



- no reason to follow a certain direction.

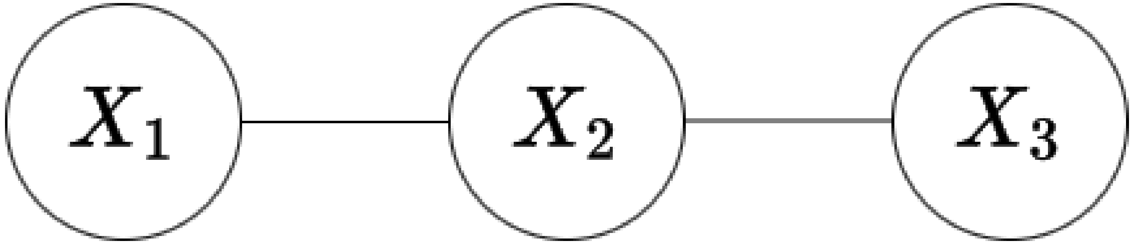
Undirected Graphical Model
Markov **R**andom **F**ield (MRF)



A direct edge between X_i and X_j

- $when\ X_i\ and\ X_j$ are <u>conditionally dependent</u> given all other variables.
- $\leftrightarrow X_i\ and\ X_j$ <u>are not conditionally independent</u> given all other variables.

Undirected Graphical Model
Markov **R**andom **F**ield (MRF)



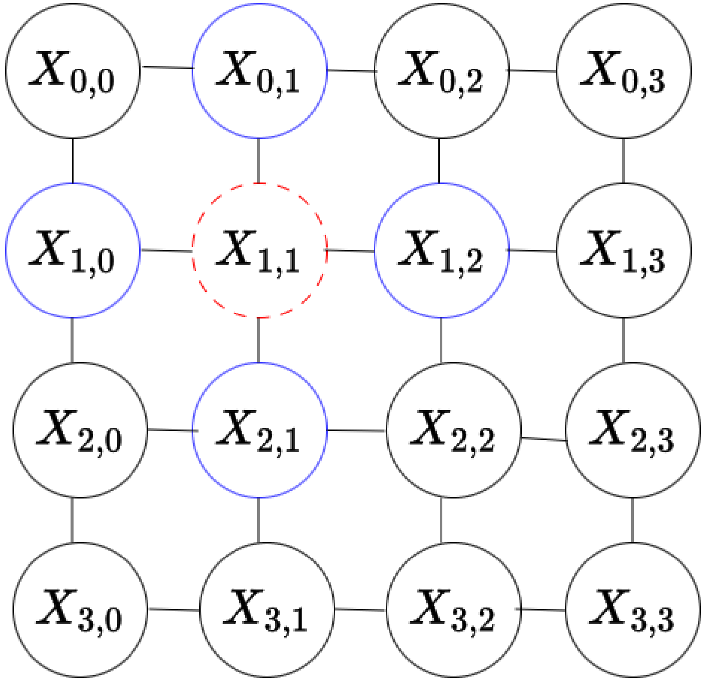
A direct edge between X_i and X_j

- $when\ X_i\ and\ X_j$ are conditionally dependent given all other variables.
- $\leftrightarrow X_i\ and\ X_j$ are not conditionally independent given all other variables.
- X_i's behavior cannot be described fully by other variables.

No direct edge between X_i and X_j

- $when\ X_i\ and\ X_j$ are conditionally indepdendent given all other variables.
- $No\ need\ of\ X_j\ to\ describe X_i$

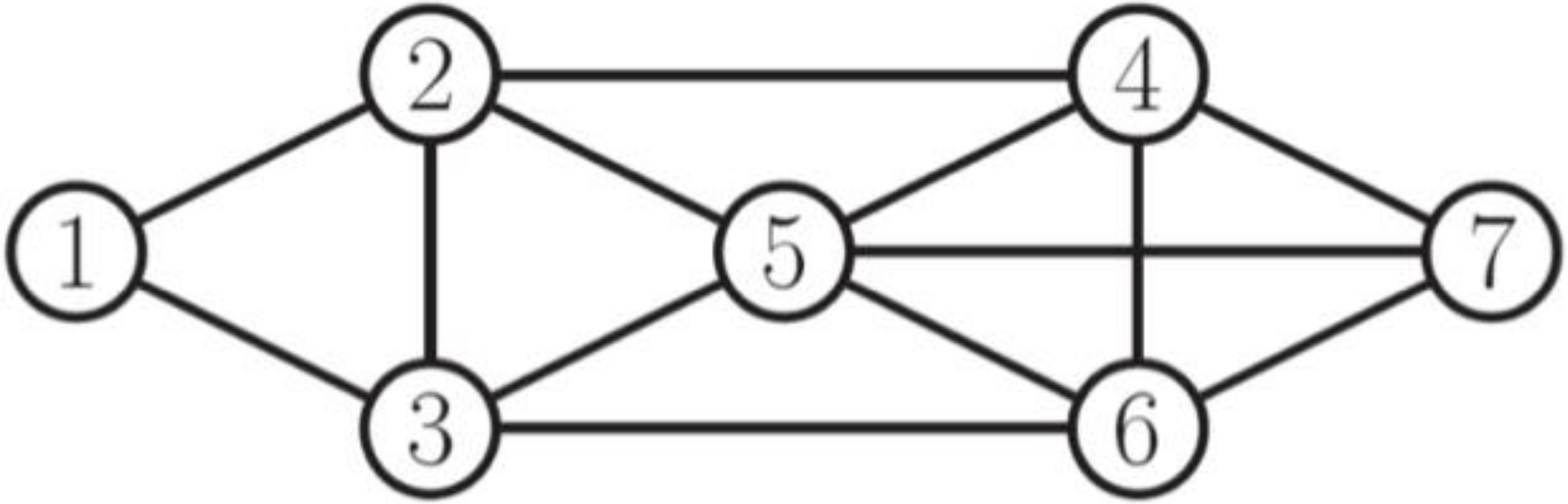
MRF Representation of Images



In this representation,
A pixel value/behavior is fully described by neighbors.
The adjacent nodes become Markov Blanket for a node.

General Markov Properties

$X_A \perp X_B \,| X_s$ iif the sets A and B are separated (no path exits) by S

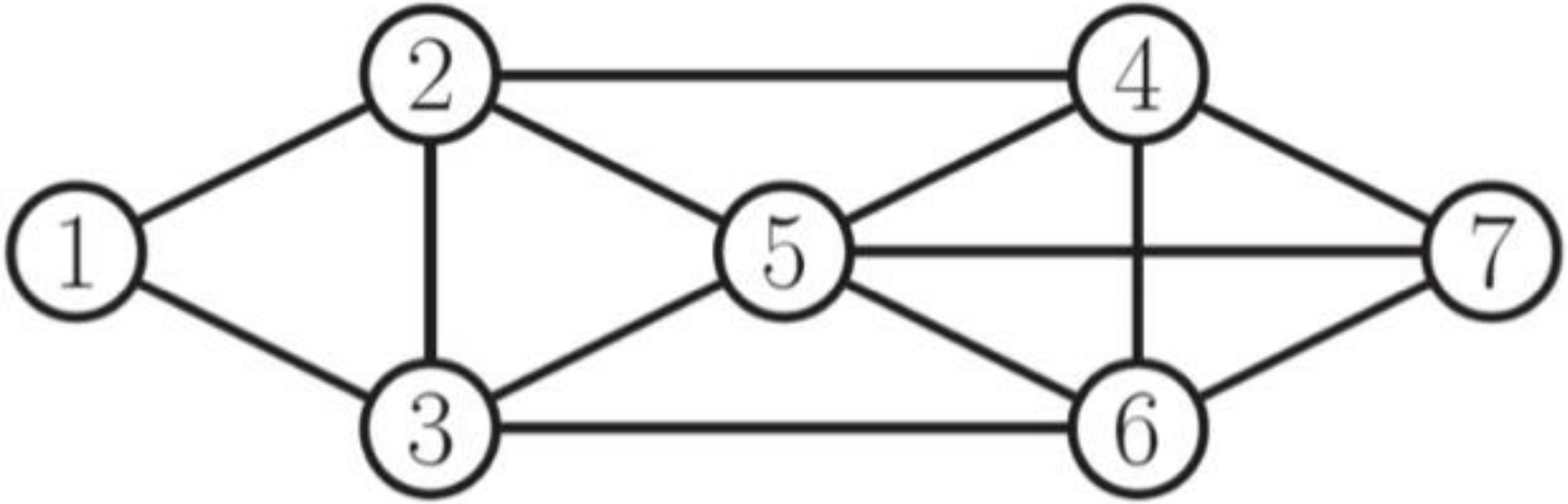


Reference:
https://erdogdu.github.io/csc412/slides/w03/sld03.pdf

Reference:
https://erdogdu.github.io/csc412/slides/w03/sld03.pdf

General Markov Properties

$X_A \perp X_B \,| X_s$ iif the sets A and B are separated (no path exits) by S



Q: $(X_1, X_2) \perp (X_6, X_7) \,|\, (X_3, X_4, X_5)$? True

Q: $X_1 \perp X_5 \,|\, (X_2, X_3)$? True

Hammersley–Clifford Theorem

Suppose a joint density satisfying conditional independence (CI) condition by the undirected graph G. Then *P* can be written as follow.

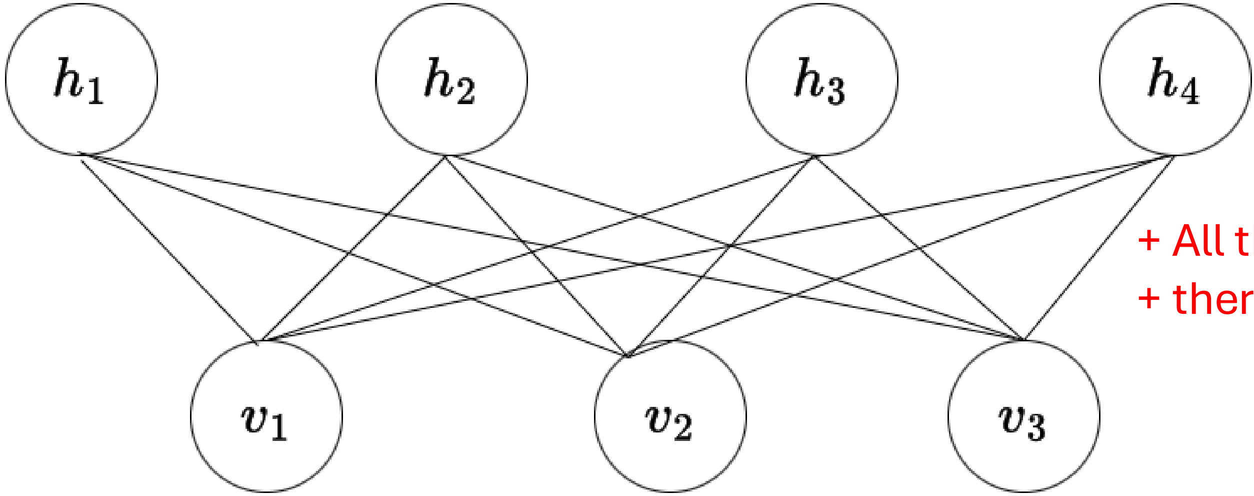
$$P(x|\theta) = \frac{1}{Z(\theta)} \prod_{c \in C} \psi(x_c|\theta_c)$$

where $Z(\theta) = \int \prod_{c \in C} \psi(x_c|\theta_c) dx$ **and** ψ **is a non-negative function**

and C be the set of all maximal cliques of G

- A clique C in an undirected graph $G = (V, E)$,
is a subset of vertex that every two distinct vertexes are adjacent.

- A maximal clique is a clique in a graph
that cannot be extended by adding another adjacent vertex.

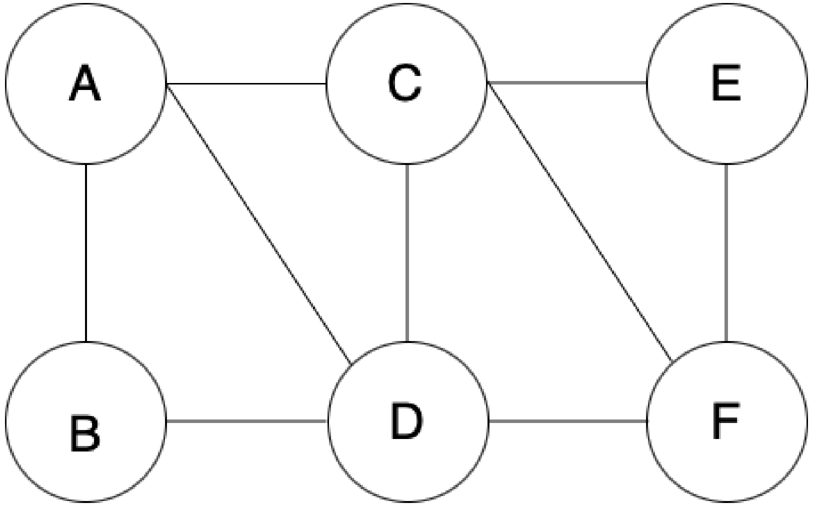


Q: What is the maximal cliques in this RBM?
Q: How many are they?

+ All the combinations of one hidden unit and one visible unit
+ there are 12 cliques.

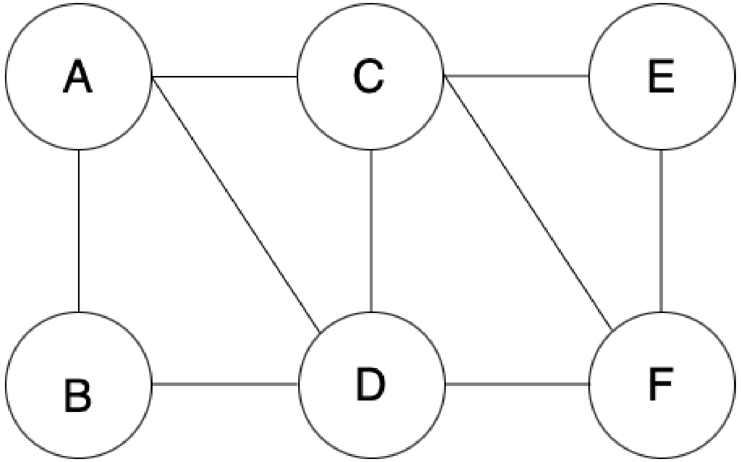
$$P(v,h) = \frac{1}{Z}\prod_{i,j}\psi(h_i, v_j)$$

Example of Hammersley–Clifford Theorem



$$P(A,B,C,D,E,F) \propto f_1(A,B,D) \cdot f_2(A,C,D) \cdot f_3(C,D,F) \cdot f_4(C,E,F)$$

Example of Hammersley–Clifford Theorem



$$P(A,B,C,D,E,F) \propto f_1(A,B,D) \cdot f_2(A,C,D) \cdot f_3(C,D,F) \cdot f_4(C,E,F)$$

$$\begin{aligned} P(A|e=B,C,D) &= \sum_{EF} \alpha f_1(A,\mathbf{B},\mathbf{D}) f_2(A,\mathbf{C},\mathbf{D}) f_3(\mathbf{C},\mathbf{D},F) f_4(\mathbf{C},E,F) \\ &= \alpha f_1(A,\mathbf{B},\mathbf{D}) f_2(A,\mathbf{C},\mathbf{D}) \sum_{EF} f_3(\mathbf{C},\mathbf{D},F) f_4(\mathbf{C},E,F) \\ &= \alpha f_1(A,\mathbf{B},\mathbf{D}) f_2(A,\mathbf{C},\mathbf{D}) \end{aligned}$$

Represented by two potential functions sharing A

Energy **B**ased **M**odel (EBM)

Energy Function $\mathcal{E}(\vec{x})$

It is a scalar valued function (no needs to be non-negative)
that measures the compatibility between the variables $(\vec{x})$.

+ A smaller value implies better compatibility
  For examples) -log $P(X)$

Inference & Learning using Energy Function

- **Inference**

  :finding unknown values minimizing the energy given observation

  $h *= \ argmin_h \, \mathcal{E}(v_1, v_2,,,, h)$

- **Learning**

  :finding an energy function

  that associates low energies to correct values,

  and higher energies to incorrect values.

One advantage of using the energy function is that
we use represent a density without a pre-defined structure/ specific parametric form.
And it will be expressive.

Energy function & probability density are conceptually aligned: $1/\mathcal{E}(x) \sim P(x)$.
Exponential family probabilistic models $P(x)$
can be represented by using using energy function $\mathcal{E}(x)$.
(we are going to focus on this today)

$$P(x) \; = \frac{1}{z} exp^{-\mathcal{E}(x)} \leftrightarrow \; \mathcal{E}(x) = -\ln p(x) + c$$

The exponential energy function
will be a reasonable choice for potential function in MRF probability modeling.
Whv?

$$P(x|\theta) = \frac{1}{Z(\theta)} \prod_{c \in C} \psi(x_c|\theta_c)$$

$$\frac{1}{Z(\theta)} \prod_{c \in C} \exp^{-\mathcal{E}(x_c|\theta_c)}$$

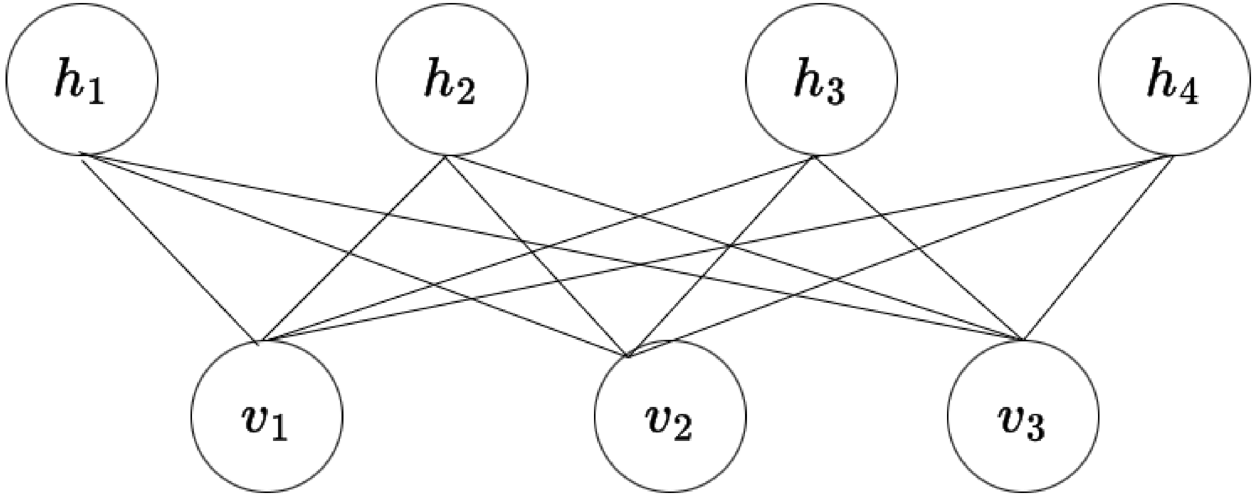
$$\frac{1}{Z(\theta)} \exp^{\sum_{c \in C} -\mathcal{E}(x_c|\theta_c)}$$

- Energy functions on local cliques
  defines the energy function
  of the global joint density as the sum of the locals

$$\mathcal{E}(x) = \sum_{c \in C} -\mathcal{E}(x_c|\theta_c)$$

Then how could we represent a joint density by using energy function in RBM?

defining the local energy → the global energy



Based on Hammersley–Clifford theorem,
The joint density by RBM is

$$P(v,h) = \frac{1}{Z}\prod_{i,j}\psi(h_i, v_j)$$

As the potential function is defined as an exponential energy function,
The joint density by RBM is

$$\begin{aligned} P(v,h) &= \frac{1}{Z}\prod_{i,j}\psi(v_i, h_j) \\ &= \alpha\prod_{i,j}\exp^{-\mathcal{E}(v_i,h_j)} \\ &= \alpha\exp^{-\sum_{i,j}\mathcal{E}(v_i,h_j)} \end{aligned}$$

+ Once we define the local energy relation between $v_i\ h_j$ it defines the global relations through the summation of exponents (energy functions).

Local & global Energy Function in RBM

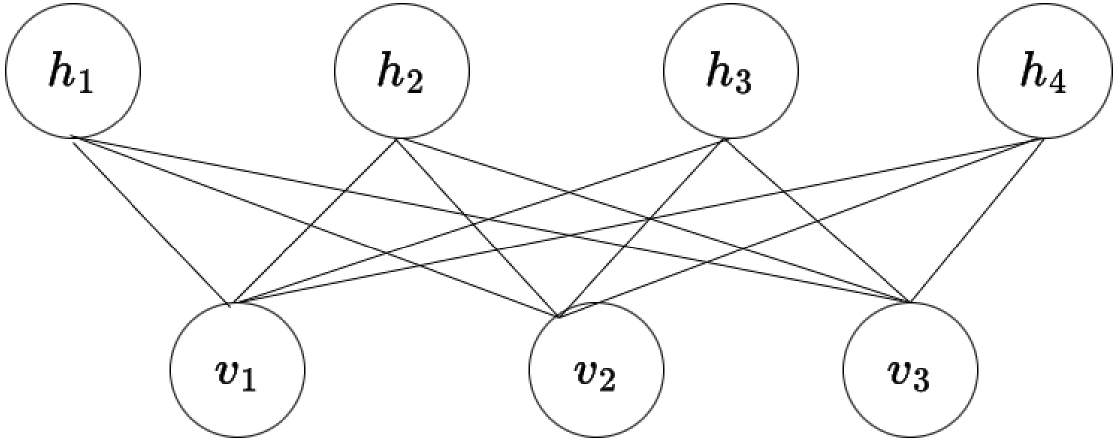
$$\mathcal{E}(v_i, h_j) = -w_{i,j} v_i h_j - b_i \cdot v_i - c_j \cdot h_j$$ [local]

$$\mathcal{E}(u, v) = -v^t W h - b^t v - c^t h$$ [global]

The energy function of local cliques for RBM is defined by the linear function. The total energy is represented by the matrix multiplication form.

How can we compute the posterior density using the joint density?

- $P(h_1|v_1, v_2, v_3)$
- $P(v_1|h_1, h_2, h_3, h_4)$



$$P(h_1|v_1, v_2, v_3) = \alpha P(h_1, v_1, v_2, v_3)$$
$$= \sum_{h_2,h_3,h_4} \alpha P(h_1, h_2, h_3, h_4, v_1, v_2, v_3)$$
$$= \alpha \exp^{v^t W[:,1]h_1 + c_1 h_1} \cdot \left(\sum_{h_2,h_3,h_4} \exp^{v^t W[:,i]h_i + c_i h_i} \right)$$
$$= \alpha \exp^{v^t W[:,1]h_1 + c_1 h_1}$$

Q : Why we did not consider the parameters b?

+ b is multiplied with the visible observation so the value is absorbed into \alpha

Normalization

$$\left\{\begin{aligned} P(h_1 = 1|v_1, v_2, v_3) &= \alpha \exp^{v^t W[:,1]+c_1} \\ P(h_1 = 0|v_1, v_2, v_3) &= \alpha \end{aligned}\right.$$

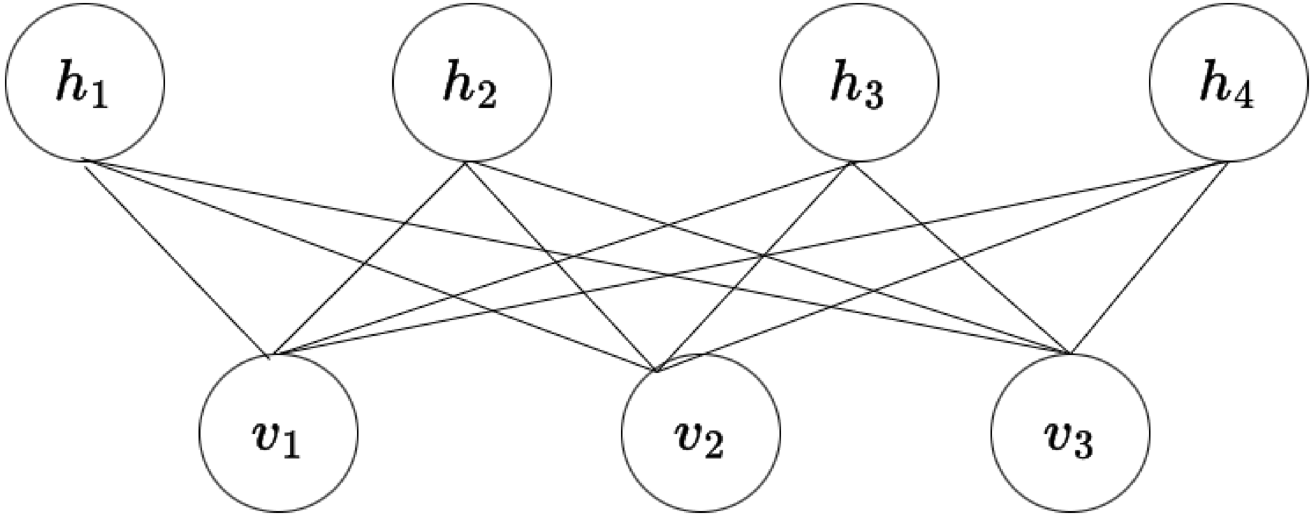
$$P(h_1 = 1|v_1, v_2, v_3) = \frac{\alpha \exp^{v^t W[:,1]+c_1}}{\alpha \exp^{v^t W[:,1]+c_1} + \alpha}$$

$$P(h_1 = 1|v_1, v_2, v_3) = \frac{1}{1 + \exp^{-v^t W[:,1]-c_1}} = \sigma(v^t W[:, 1] + c_1)$$

At any point in Training / Inference / Sampling

We can compute the posterior easily given the parameter (W, b, c)

This highlights the advantages of using RBM.



Q: how could we generate sample points (from the v and h layers) that follows the joint density encoded by RBM?

+ Start with an arbitrary v and iteratively generate samples from v to h, from h to v, from v to h, ...for infinite times.

Training RBM

Learning the parameters W, b, c

Q: how could learn the energy function?

[1] Minimization of energy function itself

$$L = \sum_{n=1}^{N} \mathcal{E}_\theta(x_i)$$

Q: Is this a good idea?

+ no, for model collapse

[2] Minimization of **N**egative **L**og **L**ikelihood (NLL)

$$
\begin{aligned}
-\log p_\theta(x) &= -\log \frac{\exp -\mathcal{E}_\theta(x)}{Z} \\
&= \mathcal{E}_\theta(x) + \log Z \\
-\nabla_\theta \log p_\theta(x) &= \nabla_\theta \mathcal{E}_\theta(x) + E_{p_\theta(x)}[\nabla_\theta\{-\mathcal{E}_\theta(x)\}] \\
&= \boxed{1/N \sum_{n=1}^{N} \nabla_\theta \mathcal{E}_\theta(x_n)} - \boxed{1/S \sum_{s=1}^{S} \nabla_\theta \mathcal{E}_\theta(x_s)}
\end{aligned}
$$

+ data

+ by sampling from the models

$$
\begin{aligned}
\nabla_\theta \log Z &= \nabla_\theta\{\log \int \exp -\mathcal{E}_\theta(x) dx\} \\
&= \{\int \exp -\mathcal{E}_\theta(x) dx\}^{-1} \nabla_\theta \int \exp -\mathcal{E}_\theta(x) dx \\
&= \int 1/Z \cdot \exp -\mathcal{E}_\theta(x) \nabla_\theta\{-\mathcal{E}_\theta(x)\} dx \\
&= E_{p_\theta(x)}[\nabla_\theta\{-\mathcal{E}_\theta(x)\}]
\end{aligned}
$$

[2] Minimization of **N**egative **L**og **L**ikelihood (NLL)

$$\begin{aligned} -\log p_\theta(x) &= -\log \frac{\exp -\mathcal{E}_\theta(x)}{Z} \\ &= \mathcal{E}_\theta(x) + \log Z \end{aligned}$$

$$\begin{aligned} \nabla_\theta \log Z &= \nabla_\theta\{\log \int \exp -\mathcal{E}_\theta(x)dx\} \\ &= \{\int \exp -\mathcal{E}_\theta(x)dx\}^{-1}\nabla_\theta \int \exp -\mathcal{E}_\theta(x)dx \\ &= \int 1/Z \cdot \exp -\mathcal{E}_\theta(x)\nabla_\theta\{-\mathcal{E}_\theta(x)\}dx \\ &\boxed{= E_{p_\theta(x)}[\nabla_\theta\{-\mathcal{E}_\theta(x)]} \end{aligned}$$

Q: how can we compute this?

Q: what is the effect of adding this?

+ we can generate the samples following a model by using the method in the slide page 35.

+ we can pull up the energy function for the data set which is not following the true distribution.

Monte Carlo Estimates of Expectation (From the slide Nov. 25)

$$
\begin{aligned}
E_{f(x)}[Q(x)] &= \int Q(x)f(x)dx \\
&\simeq 1/L \sum_{l=1}^{L} Q(x_i)
\end{aligned}
$$

[3] NLL Loss in RBM

$$
\begin{aligned}
-\log p_\theta(x) &= -\log \frac{\exp -\mathcal{E}_\theta(x)}{Z} \\
&= \mathcal{E}_\theta(x) + \log Z \\
-\nabla_\theta \log p_\theta(x) &= \nabla_\theta \mathcal{E}_\theta(x) + E_{p_\theta(x)}[\nabla_\theta\{-\mathcal{E}_\theta(x)] \\
&= 1/N \sum_{n=1}^{N} \nabla_\theta \mathcal{E}_\theta(x_n) - 1/S \sum_{s=1}^{S} \nabla_\theta \mathcal{E}_\theta(x_s)
\end{aligned}
$$

- Derivatives for RBM are straightforward to compute!

$$
\nabla \mathcal{E}(W,b,c) = \begin{bmatrix} \dfrac{\partial \mathcal{E}(W,b,c)}{\partial W_{i,j}} = -v_i h_j \\ \dfrac{\partial \mathcal{E}(W,b,c)}{\partial b_i} = -v_i \\ \dfrac{\partial \mathcal{E}(W,b,c)}{\partial c_j} = -h_j \end{bmatrix}
$$

$$
W,b,c(t+1) = W,b,c(t) - \eta(\nabla \mathcal{E}_{data}(W,b,c) - \nabla \mathcal{E}_{model}(W,b,c))
$$

Q: how could we collect the samples to follow the current model ?

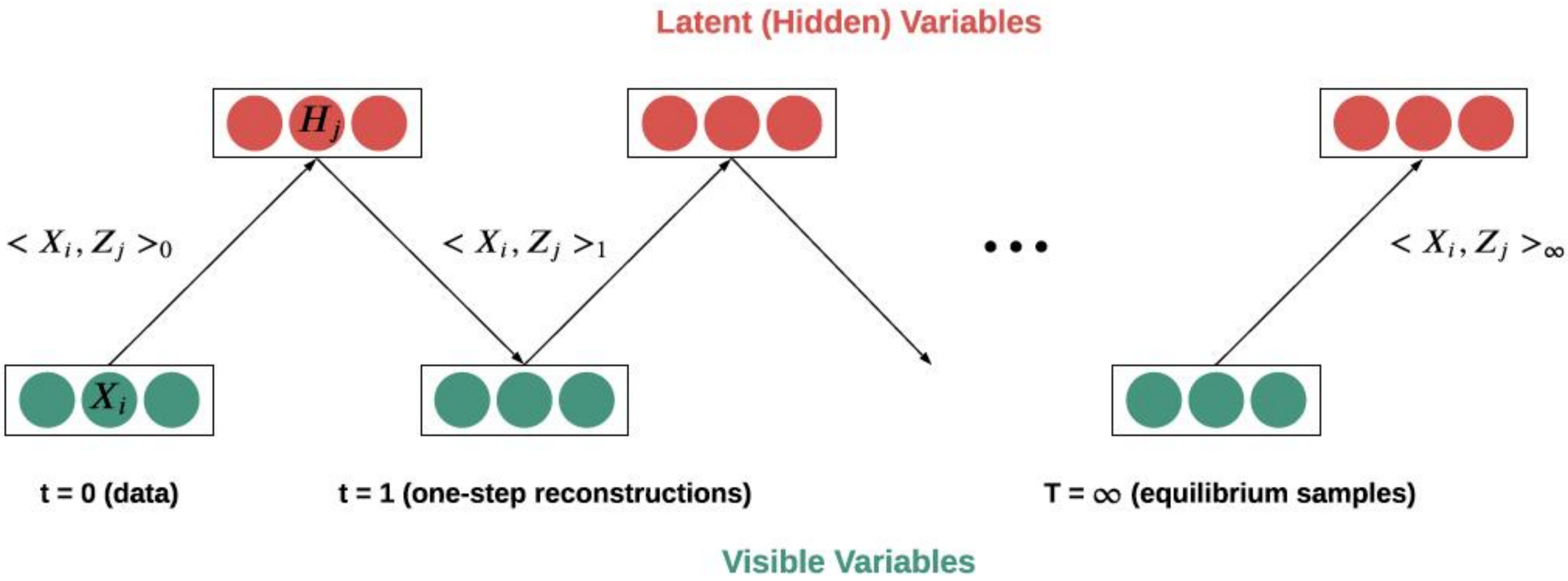
we can generate the samples following a model by using the method in the slide page 35.

Contrastive Divergence (Geoffrey Hinton 2002)

enables us to compute approximate MLE. (approximate model sampling)

https://www.cs.toronto.edu/~fritz/absps/tr00-004.pdf

Illustration of Contrastive Divergence Sampling for RBM



Theoretically, the reconstruction samples from CD_∞ will compute the correct exact gradient. However, CD_1 works well enoguh in many RBM applications!

The Learning Direction of RBM:

In RBM, learning is to update energy function (internal model) toward the direction of minimizing the energy difference between training and simulated data at the current model parameters. Hence, once current model is well aligned with training data (converged), no more update.

$$\begin{aligned}\nabla_{NLL} &= \nabla_\theta \mathcal{E}_\theta(x) + E_{p_\theta(x)}[\nabla_\theta\{-\mathcal{E}_\theta(x)\}] \\ &= \underbrace{\nabla_\theta(\mathcal{E}_{data} - \mathcal{E}_{model})}\end{aligned}$$

In computational brain science.
RBM learning to minimize the energy difference
is related to reduce the free energy!
is to reduce surprise by the gap between internal model and external data.
Our brain modify/update the internal model about world through the new data.

$$
\begin{aligned}
\nabla_{NLL} &= \nabla_\theta \mathcal{E}_\theta(x) + E_{p_\theta(x)}[\nabla_\theta\{-\mathcal{E}_\theta(x)] \\
&= \nabla_\theta \underbrace{(\mathcal{E}_{data} - \mathcal{E}_{model})}_{}
\end{aligned}
$$

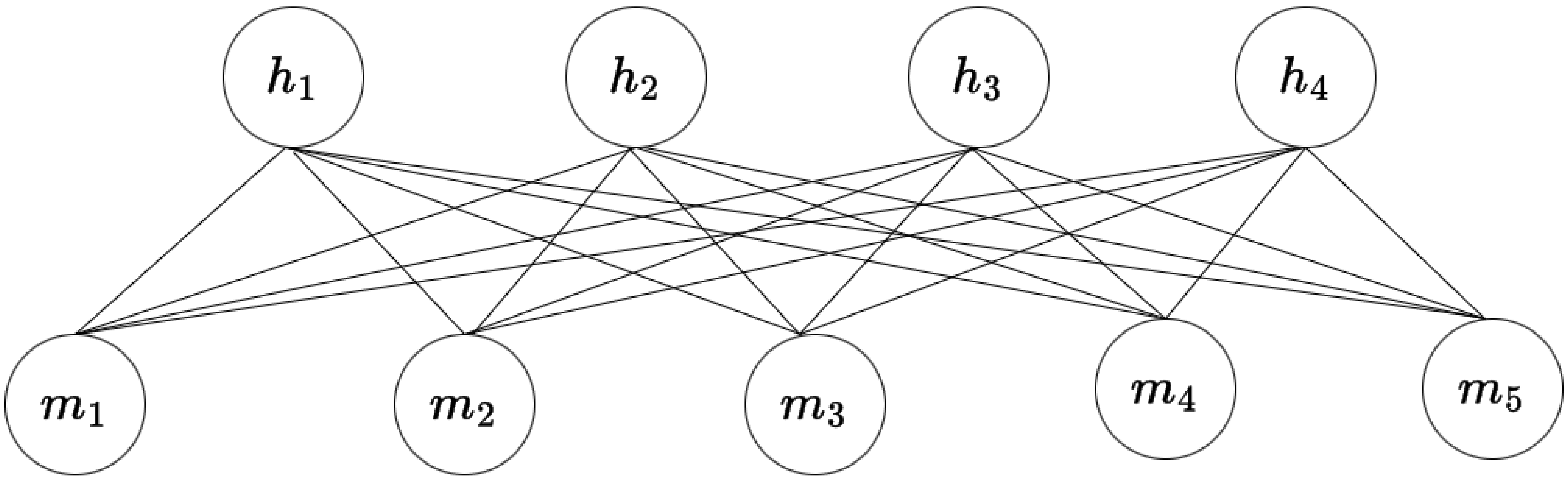
- https://www.fil.ion.ucl.ac.uk/~karl/Learning%20and%20inference%20in%20the%20brain.pdf (Free Energy)
- http://www.cs.utoronto.ca/~hinton/helmholtz.html (Helmholtz Machine)

Netflix Recommendation System (Collaborative Filtering)

http://www.cs.utoronto.ca/~hinton/absps/netflixICML.pdf

RBM Movie Recommendation System with Binary Hidden and Visible units

Hidden State to Encode Movie Preference



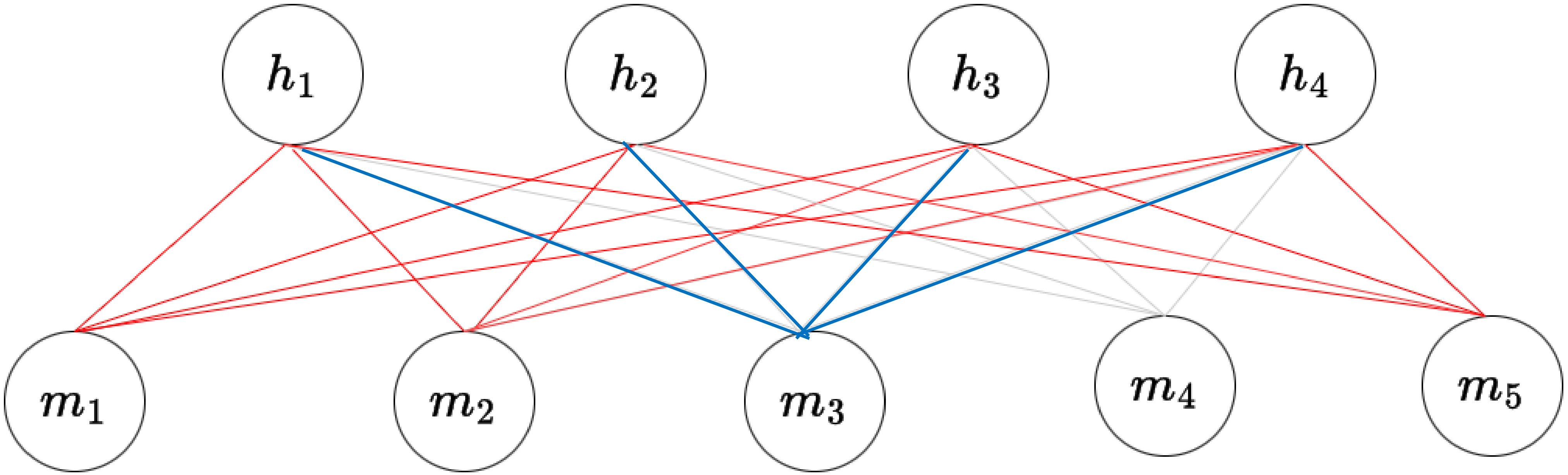
Movie Preference Encoded by 0 /1

[1] When all N users rated the same set of Movies,
we train one RBM.

[2] When many of the rating are missing (this is a common scenario),
we use a different RBM for each user.

When many of the rating are missing (this is a common scenario), we will train different RBMs for each user.

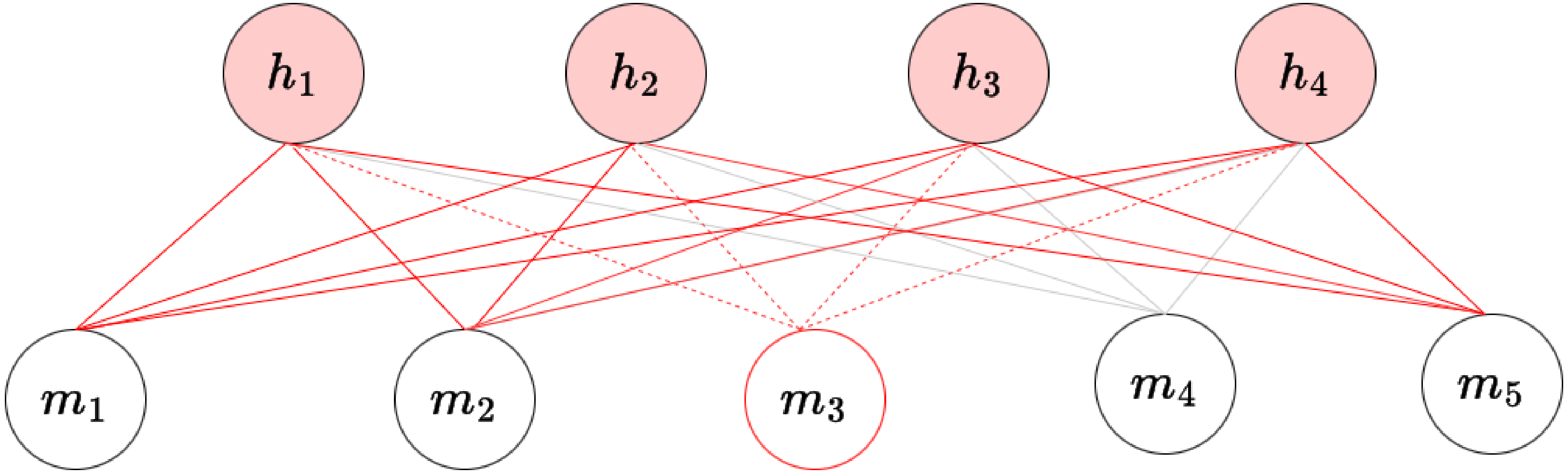
However, the weights will be shared among the user!!!



Q: Draw the RBM connection when a user has the preference data for m3 only?

+ blue-line connection.

[Prediction of Preference M3]



Regardless of the presence of missing preferences,
A user ‘s preference can be encoded by using $P[h|m]$.
Then we can reconstruct the preference information for a target movie m_k
by using $p[m_k|\ h]$