10417-617
Deep Learning: Fall 2019

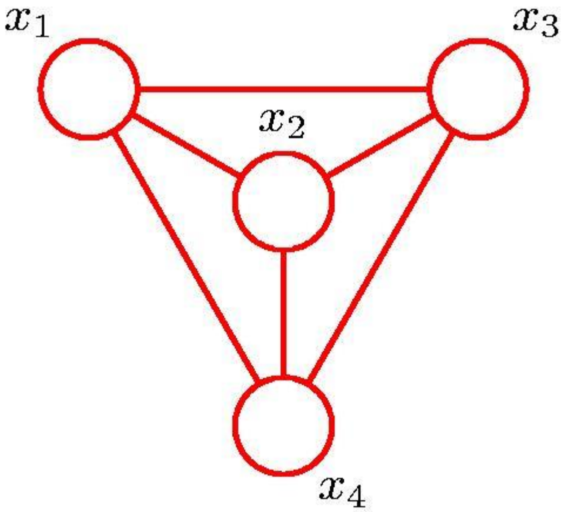
Andrej Risteski

Machine Learning Department

Lecture 12:
Markov Chains, applications to learning undirected models and RBMs

Graphical Models

Recall: graph contains a set of nodes connected by edges.



In a probabilistic graphical model, each node represents a random variable, links represent "probabilistic dependencies" between random variables.

Graph specifies how joint distribution over all random variables decomposes into a **product** of factors, each factor depending on a subset of the variables.

Two types of graphical models:

- **Bayesian networks**, also known as Directed Graphical Models (the links have a particular directionality indicated by the arrows)
- **Markov Random Fields**, also known as Undirected Graphical Models (the links do not carry arrows and have no directional significance).

Algorithmic pros/cons of latent-variable models (so far)

RBM's

- Hard to draw samples ✗

(In fact, #P-hard provably, even in Ising models)

- Easy to sample posterior distribution over latents ✓

Directed models

- Easy to draw samples ✓

- Hard to sample posterior distribution over latents ✗

(In fact, #P-hard even in mixtures)

Canonical tasks with graphical models

Inference

Given values for the parameters θ of the model, *sample/calculate* marginals (e.g. sample $p_\theta(x_1), p_\theta(x_4, x_5), p_\theta(z|x)$, etc.)

Learning

Find values for the parameters θ of the model, that give a *high likelihood* for the observed data. (e.g. canonical way is solving maximum likelihood optimization

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} \log p(x_i)$$

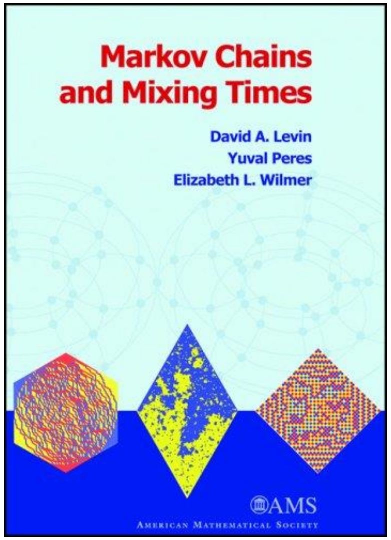
Other methods exist, e.g. method of moments (matching moments of model), but less used in deep learning practice.

Algorithmic approaches

When faced with a difficult to calculate probabilistic quantity (partition function, difficult posterior), there are two families of approaches:

MARKOV CHAIN MONTE CARLO

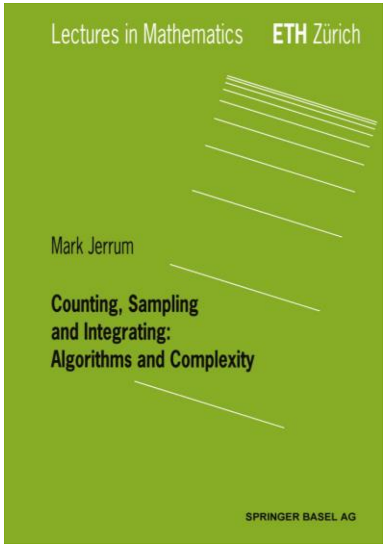
- Random walk w/ equilibrium distribution the one we are trying to sample from.



Markov Chains and Mixing Times

David A. Levin
Yuval Peres
Elizabeth L. Wilmer

AMS
American Mathematical Society



Lectures in Mathematics ETH Zürich

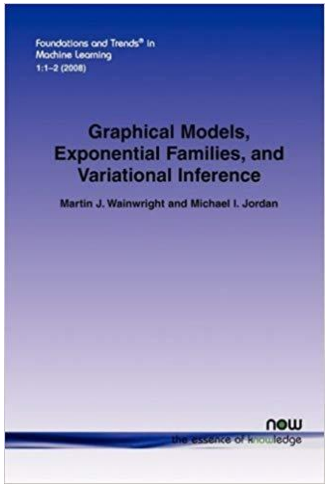
Mark Jerrum

Counting, Sampling and Integrating: Algorithms and Complexity

SPRINGER BASEL AG

VARIATIONAL METHODS

- Based on solving an optimization problem.



Foundations and Trends® in Machine Learning
1:1–2 (2008)

Graphical Models, Exponential Families, and Variational Inference

Martin J. Wainwright and Michael I. Jordan

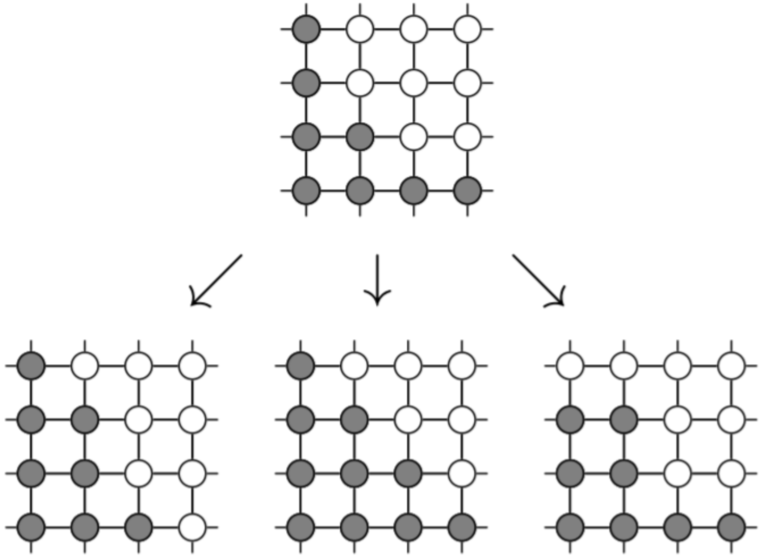
now
the essence of knowledge

Part I: Intro to Markov Chains

Sampling via random walks

Goal: Sample from distribution given up to constant of proportionality.

Idea: explore domain via *random, local* moves



Hope: enough moves ⇒ the random process "forgets" starting point, follows the distr. we are trying to sample.

Sampling via random walks

Goal: Sample from distribution given up to constant of proportionality.

A set of random variables $(X_1, X_2, \dots, X_T)$ is **Markov** if
$\forall t: P(X_t | X_{<t}) = P(X_t | X_{t-1})$
It is homogeneous if $P(X_t | X_{t-1})$ doesn't depend on t.

We can describe a homogeneous Markov process on a discrete domain $\mathcal{X}$ by a **transition matrix** $T \in \mathbb{R}_+^{|\mathcal{X}| \times |\mathcal{X}|}$: $T_{ij} = P(X_{t+1} = j \, | X_t = i)$

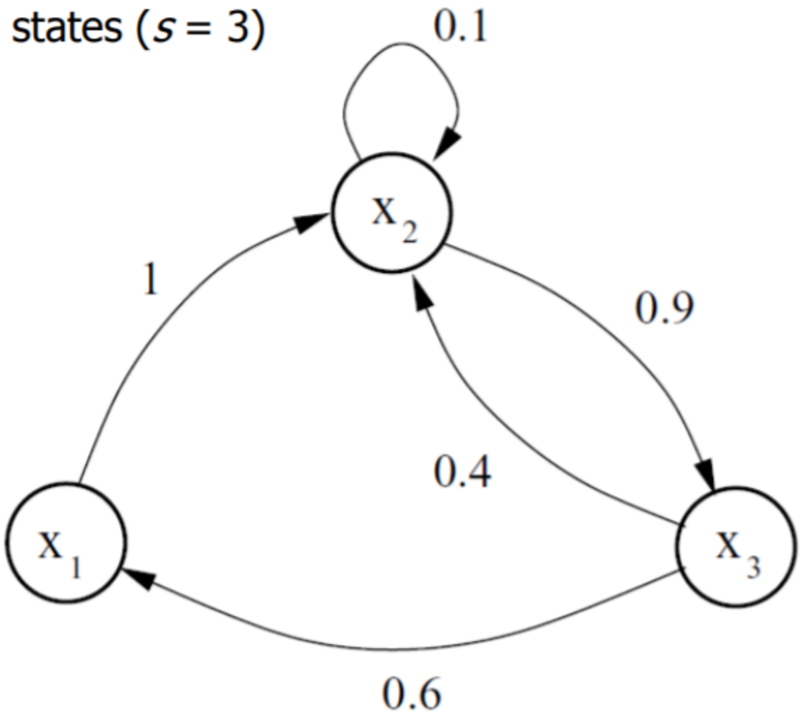
Clearly, $\forall i, \sum_j \; T_{ij} = 1$. We will also call such process a Markov Chain/ Markov random walk.

Example

Markov chain with three states ($s = 3$)

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$

Transition matrix



Transition graph

Stationary distribution

Stationary distribution: a distribution $\pi = (\pi_1, \dots \pi_{|\mathcal{X}|})$ is stationary for a Markov walk if $\pi T = \pi$.

In other words: if we start with a sample of π and transition according to T, we end with a sample following π as well.

$$(0.22, 0.41, 0.37)\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix} = (0.22, 0.41, 0.37)$$

Stationary distribution need not be unique: e.g. T is the identity matrix.

Many Markov Chains have unique stationary distributions: after taking many steps, starting with any distribution, we get to the same distribution

$$\forall p_0, \lim_{t\to\infty} p_0 T^t = \pi$$

In other words, eventually, the chain "forgets" the starting point.

Stationary distribution

Stationary distribution: a distribution $\pi = (\pi_1, \dots \pi_{|\mathcal{X}|})$ is stationary for a Markov walk if $\pi T = \pi$.

Many Markov Chains have unique stationary distributions: after taking many steps, starting with any distribution, we get to the same distribution

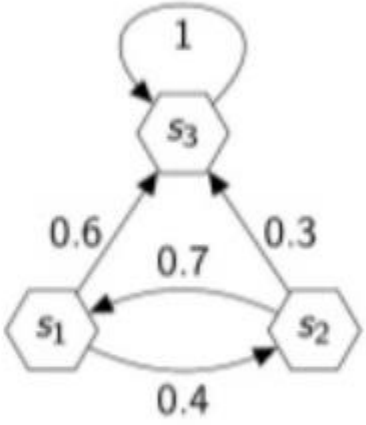
$$\forall p_0, \lim_{t \to \infty} p_0 T^t = \pi$$

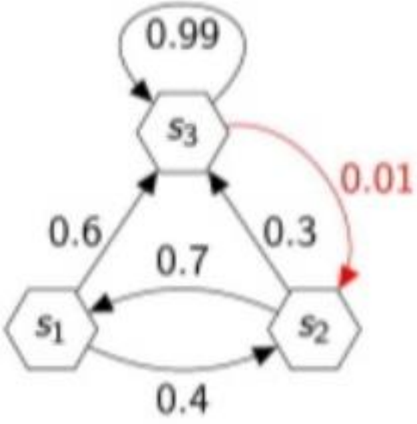
Name of the game: if we wish to sample from some π, design a Markov Chain which has π as stationary distribution.

If we run chain long enough (??), we can draw samples from something close to π

Conditions for having a unique stationary distribution

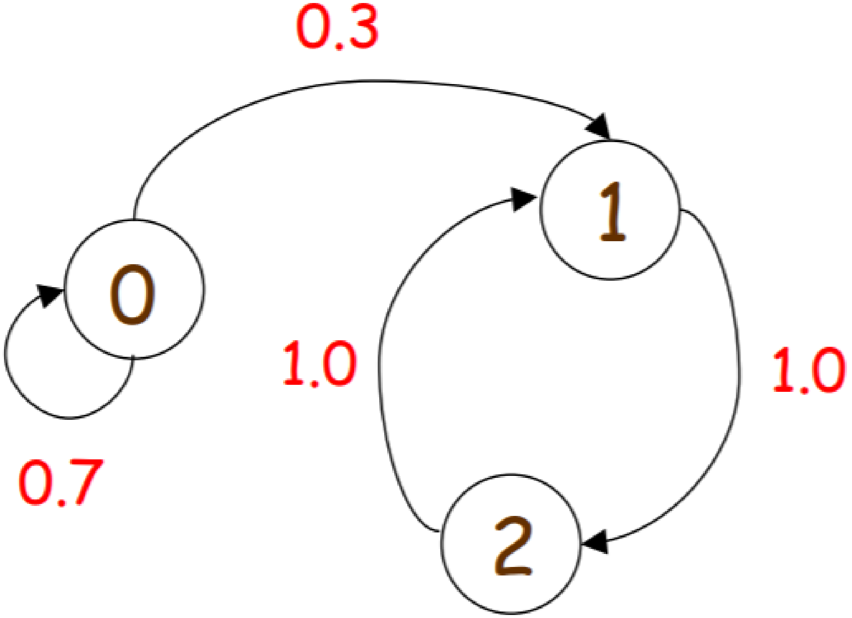
Potential problem: transition graph is not connected.





Conditions for having a unique stationary distribution

Potential problem: there are cycles in graph



Conditions for having a unique stationary distribution

These are all the possible problems!

Irreducibility: there is a path that transitions from any state to any other.
For each pairs of states (i,j), there is a positive probability, starting in state i, that the process will ever enter state j.

= Transition graph is connected;

Aperiodicity: random walk doesn't get trapped in cycles.
A state i is aperiodic if there exists n s.t., $\forall n' \geq n, P\left(X_{n'} = i \middle| X_0 = i\right) > 0$.
If all states are aperiodic, chain is called aperiodic.

Thm: for any *irreducible+aperiodic* Markov chain there is a unique π, s.t.

$$\forall p_0, \lim_{t \to \infty} p_0 T^t = \pi$$

Detailed balance

Useful sufficient condition for π to be a stationary distribution: detailed balance.

$$\pi_i T_{ij} = \pi_j T_{ji}, \forall (i,j)$$

Why?

$$(\pi T)_i = \sum_j \pi_j T_{ji} = \textstyle\sum_j \pi_i T_{ij}$$

$$= \pi_i \textstyle\sum_j T_{ij}$$

$$= \pi_i$$

Metropolis-Hastings

Suppose we are trying to sample from π defined over a domain of size m (think m is very large, like in Ising models), up to a constant of proportionality:

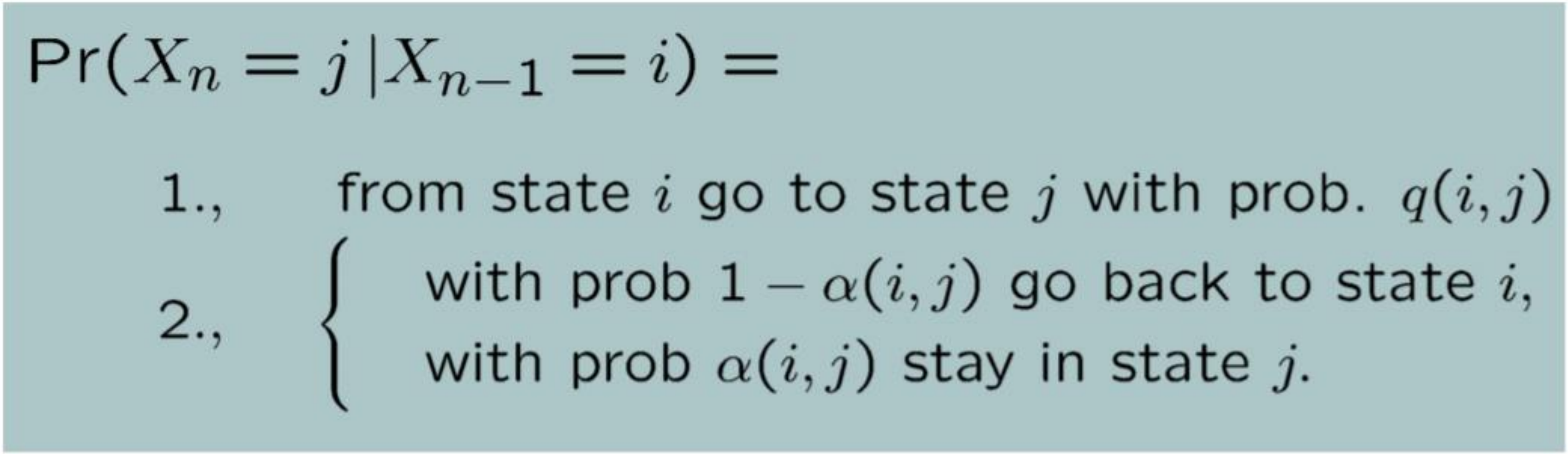
$$\pi_i = \frac{b(i)}{Z}, Z = \sum_{i=1}^{m} b(i)$$

Metropolis-Hastings: random walk assuming an “easy-to-sample from” transition kernel q(i,j), along with “corrections”.

Metropolis-Hastings

Suppose we have an easy to sample from “transition kernel” q(i,j).

Consider the following random walk, for some $\alpha(i,j)$ we will pick:



$$\Pr(X_n = j \,|X_{n-1} = i) =$$

$$\begin{array}{ll} 1., & \text{from state } i \text{ go to state } j \text{ with prob. } q(i,j) \\ 2., & \begin{cases} \text{with prob } 1-\alpha(i,j) \text{ go back to state } i, \\ \text{with prob } \alpha(i,j) \text{ stay in state } j. \end{cases} \end{array}$$

Then, we have:

$$\begin{aligned} P(X_{n+1} = j|X_n = i) &= q(i,j)\alpha(i,j) \quad \forall j \neq i \\ P(X_{n+1} = i|X_n = i) &= q(i,i) + \sum_{k \neq i} q(i,k)(1-\alpha(i,k)) \end{aligned}$$

Metropolis-Hastings

Observation

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \forall j \neq i \Leftrightarrow \pi_i q(i,j)\alpha(i,j) = \pi_j q(j,i)\alpha(j,i) \quad \forall j \neq i \quad (*)$$

$$P_{ij} = P(X_{n+1} = j | X_n = i) = q(i,j)\alpha(i,j) \ \forall j \neq i$$

Claim:

$$\text{If } \alpha(i,j) = \min\left(\frac{\pi_j q(j,i)}{\pi_i q(i,j)}, 1\right) = \min\left(\frac{b(j)q(j,i)}{b(i)q(i,j)}, 1\right)$$

$$\Rightarrow (\pi_1, \ldots \pi_m) \text{ stationary distribution}$$

Note, this only depends on unnormalized distribution (b(i) values)

$$\text{If } \alpha(i,j) = \frac{\pi_j q(j,i)}{\pi_i q(i,j)} \Leftrightarrow \alpha(j,i) = 1$$

=> Detailed balance (*) holds

Gibbs sampling

Consider sampling a distribution over n variables $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$, s.t. each of the conditional distributions $P(x_i | \boldsymbol{x}_{-i})$ is easy to sample. :

e.g. recall Ising models: $P_\theta(x_i = 1 | \mathbf{x}_{-i}) = \dfrac{1}{1 + \exp(-\theta_i - \sum_{ij \in E} x_j \theta_{ij})}$,

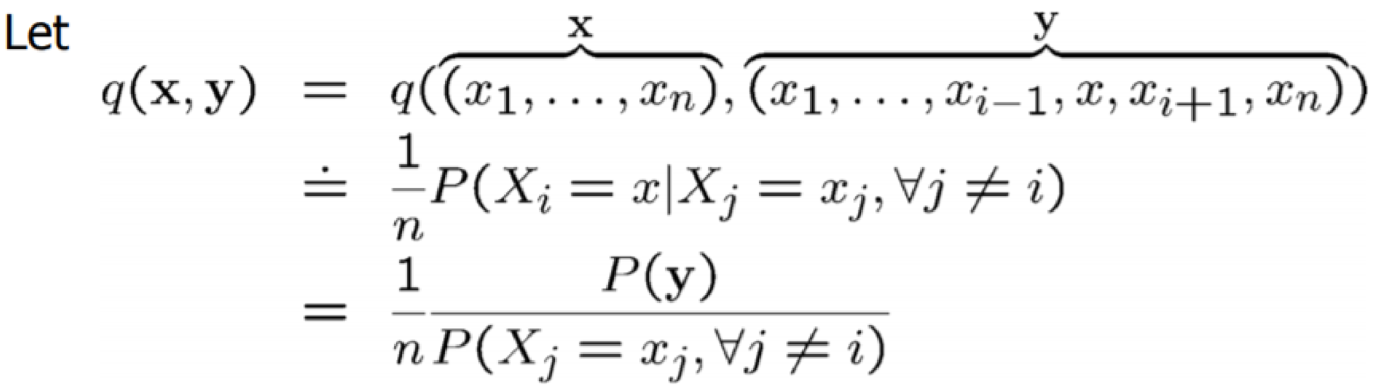
A common way to do this is using **Gibbs sampling**:

```
Repeat:
    Let current state be $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$
    Pick $i \in [n]$ uniformly at random.
    Sample x ~ $P(X_i = x | \boldsymbol{x}_{-i})$
    Update state to $\boldsymbol{y} = (x_1, x_2, \dots, x_{i-1}, x, x_{i+1}, \dots, x_n)$
```

Gibbs sampling

```
Repeat:
    Let current state be $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$
    Pick $i \in [n]$ uniformly at random.
    Sample x ~ $P(X_i = x | \boldsymbol{x}_{-i})$
    Update state to $\boldsymbol{y} = (x_1, x_2, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n)$
```

Why does it work? Metropolis-Hastings with appropriate kernel!



Let

$$\begin{aligned} q(\mathbf{x},\mathbf{y}) &= q(\overbrace{(x_1,\ldots,x_n)}^{\mathbf{x}},\overbrace{(x_1,\ldots,x_{i-1},x,x_{i+1},x_n)}^{\mathbf{y}}) \\ &\doteq \frac{1}{n}P(X_i = x | X_j = x_j, \forall j \neq i) \\ &= \frac{1}{n}\frac{P(\mathbf{y})}{P(X_j = x_j, \forall j \neq i)} \end{aligned}$$

Gibbs sampling

Why does it work? Metropolis-Hastings with appropriate kernel!

Let

$$\begin{aligned} q(\mathbf{x},\mathbf{y}) &= q(\overbrace{(x_1,\ldots,x_n)}^{\mathbf{x}},\overbrace{(x_1,\ldots,x_{i-1},x,x_{i+1},x_n)}^{\mathbf{y}}) \\ &\doteq \frac{1}{n}P(X_i = x \mid X_j = x_j, \forall j \neq i) \\ &= \frac{1}{n}\frac{P(\mathbf{y})}{P(X_j = x_j, \forall j \neq i)} \end{aligned}$$

Shouldn't we reject occasionally? **No**:

Claim:

If $\alpha(i,j) = \min\left(\frac{\pi_j q(j,i)}{\pi_i q(i,j)}, 1\right) = \min\left(\frac{b(j)q(j,i)}{b(i)q(i,j)}, 1\right)$

$\Rightarrow$ $(\pi_1, \ldots \pi_m)$ stationary distribution

$$\frac{p(\mathbf{y})q(\mathbf{y},\mathbf{x})}{p(\mathbf{x})q(\mathbf{x},\mathbf{y})} = \frac{p(\mathbf{y})\frac{1}{n}\frac{P(\mathbf{x})}{P(Y_j = y_j, j \neq i)}}{p(\mathbf{x})\frac{1}{n}\frac{P(\mathbf{y})}{P(X_j = x_j, j \neq i)}}$$

Gibbs sampling

Why does it work? Metropolis-Hastings with appropriate kernel!

Let

$$
\begin{aligned}
q(\mathbf{x},\mathbf{y}) &= q(\overbrace{(x_1,\ldots,x_n)}^{\mathbf{x}},\overbrace{(x_1,\ldots,x_{i-1},x,x_{i+1},x_n)}^{\mathbf{y}}) \\
&\doteq \frac{1}{n}P(X_i = x | X_j = x_j, \forall j \neq i) \\
&= \frac{1}{n}\frac{P(\mathbf{y})}{P(X_j = x_j, \forall j \neq i)}
\end{aligned}
$$

Shouldn’t we reject occasionally? **No**:

$$
\frac{p(\mathbf{y})q(\mathbf{y},\mathbf{x})}{p(\mathbf{x})q(\mathbf{x},\mathbf{y})} = \frac{p(\mathbf{y})\frac{1}{n}\frac{P(\mathbf{x})}{P(Y_j=y_j, j\neq i)}}{p(\mathbf{x})\frac{1}{n}\frac{P(\mathbf{y})}{P(X_j=x_j, j\neq i)}} \underset{\uparrow}{=} \frac{p(\mathbf{y})p(\mathbf{x})}{p(\mathbf{x})p(\mathbf{y})} = 1
$$

since $P(X_j = x_j, j \neq i) = P(Y_j = y_j, j \neq i)$

What governs "mixing time"

So far, we've only worried about designing chains s.t. $\forall p_0, \lim_{t\to\infty} p_0 T^t = \pi$

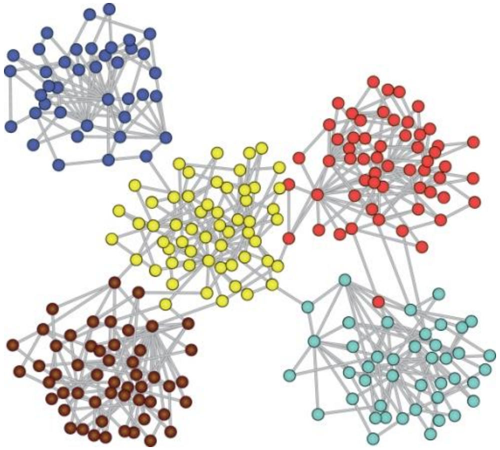
But, we're running this in practice, so want for sensible t, $\forall p_0, \; p_0 T^t \approx \pi$

(Appropriately formalized, this is called *mixing time*.)

There is no silver bullet for analyzing general transition T, but one common tool is *conductance*: which essentially says the transition graph doesn't have "bottlenecks".

The conductance of a subset S is defined as:

$$\phi(S) = \frac{\sum_{i\in S, j\notin S} T_{ij}}{\sum_{i\in S} \pi_i}$$



(e.g. how easy it is to leave S, given that we started in S)

(e.g. the colored sets have poor conductance)

What governs “mixing time”

So far, we’ve only worried about designing chains s.t. $\forall p_0, \lim_{t\to\infty} p_0 T^t = \pi$

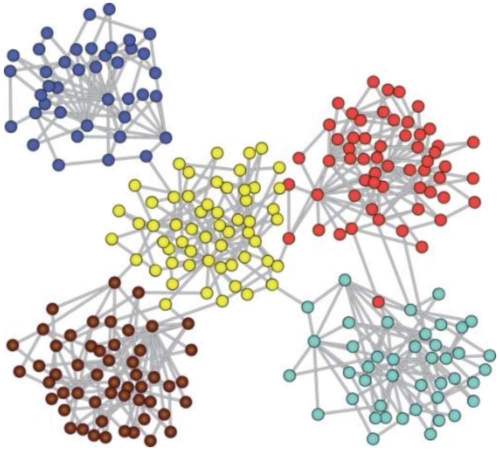
But, we’re running this in practice, so want for sensible t, $\forall p_0,\ p_0 T^t \approx \pi$

(Appropriately formalized, this is called *mixing time*.)

There is no silver bullet for analyzing general transition T, but one common tool is *conductance*: which essentially says the transition graph doesn’t have “bottlenecks”.

The conductance of a subset S is defined as:

$$\phi(S) = \frac{\sum_{i\in S, j\notin S} T_{ij}}{\sum_{i\in S} \pi_i}$$



It’s clear that sets of poor $\phi(S)$ impede mixing time:

If we start at S, even with the correct π, it’ll take us long to leave S.

What governs “mixing time”

So far, we’ve only worried about designing chains s.t. $\forall p_0, \lim_{t\to\infty} p_0 T^t = \pi$

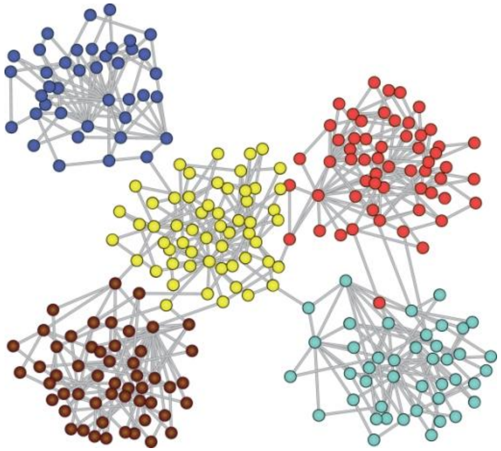
But, we’re running this in practice, so want for sensible t, $\forall p_0,\ p_0 T^t \approx \pi$

(Appropriately formalized, this is called *mixing time*.)

There is no silver bullet for analyzing general transition T, but one common tool is *conductance*: which essentially says the transition graph doesn’t have “bottlenecks”.

The conductance of a subset S is defined as:

$$\phi(S) = \frac{\sum_{i\in S, j\notin S} T_{ij}}{\sum_{i\in S} \pi_i}$$



It’s clear that sets of poor $\phi(S)$ impede mixing time:

The distribution is “multimodal”: has S’s that have large probability, but are difficult to transition between.

What governs "mixing time"

So far, we've only worried about designing chains s.t. $\forall p_0, \lim_{t\to\infty} p_0 T^t = \pi$

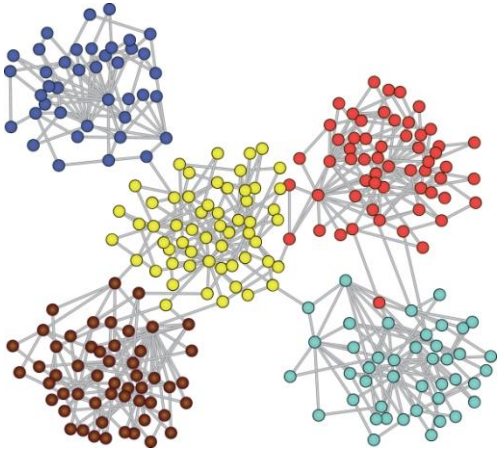
But, we're running this in practice, so want for sensible t, $\forall p_0,\ p_0 T^t \approx \pi$

(Appropriately formalized, this is called *mixing time*.)

There is no silver bullet for analyzing general transition T, but one common tool is *conductance*: which essentially says the transition graph doesn't have "bottlenecks".

The conductance of a subset S is defined as:

$$\phi(S) = \frac{\sum_{i\in S, j\notin S} T_{ij}}{\sum_{i\in S} \pi_i}$$



Conversely, if $\phi(S)$ is large for all S => mixing time is good!

What governs “mixing time”

So far, we’ve only worried about designing chains s.t. $\forall p_0, \lim_{t \to \infty} p_0 T^t = \pi$

But, we’re running this in practice, so want for sensible t, $\forall p_0,\ p_0 T^t \approx \pi$

(Appropriately formalized, this is called *mixing time*.)

Note common misconception: random walk must visit each state in domain to mix.

This is of course not true! (There does however need to be a reasonable **probability** that some set of moves gets us anywhere in the domain.)

(Otherwise, what would be the point of running a Markov Chain as opposed to brute force calculation of the partition function…)

Part II: Learning Undirected Models and Restricted Boltzmann Machines (RBMs)

Warmup: learning fully observed undirected models

Goal: Learn distribution given up to constant of proportionality

$$p_\theta(x) \propto \exp(-E_\theta(x))$$

Recall our basic approach: maximum likelihood

Given data $x_1, x_2, \ldots, x_n$, solve the optimization problem

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} \log p_\theta(x_i)$$

Expanding likelihoods: $\log\ p_\theta(x) = -E_\theta(x) - \log Z_\theta$

Our basic algorithm: gradient descent. Can we take gradients?

$\nabla_\theta E_\theta$ is typically easy (e.g. E_θ is an Ising model, neural network, etc.)

Warmup: learning fully observed undirected models

Goal: Learn distribution given up to constant of proportionality

$$p_\theta(x) \propto \exp(-E_\theta(x))$$

Recall our basic approach: maximum likelihood

Given data $x_1, x_2, \ldots, x_n$, solve the optimization problem

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} \log p_\theta(x_i)$$

$$\nabla_\theta \log Z_\theta = \frac{1}{Z_\theta} \nabla_\theta Z_\theta \;\; = \frac{1}{Z_\theta} \nabla_\theta \left(\int_x \exp(-E_\theta(x)) \right)$$

$$\frac{1}{Z_\theta} \int_x \exp(-E_\theta(x)) \nabla_\theta(-E_\theta(x)) = \mathbb{E}_{p_\theta}[-\nabla_\theta E_\theta(x)]$$

Warmup: learning fully observed undirected models

Goal: Learn distribution given up to constant of proportionality

$$p_\theta(x) \propto \exp(-E_\theta(x))$$

$$\nabla_\theta\left(\frac{1}{n}\sum_{i=1}^{n}\log p_\theta(x_i)\right) = \frac{1}{n}\left(\sum_i -\nabla_\theta E_\theta(x_i)\right) - \mathbb{E}_{p_\theta}[-\nabla_\theta E_\theta(x)]$$

$$\approx \underbrace{\mathbb{E}_{p_{data}}[-\nabla_\theta E_\theta(x)] - \mathbb{E}_{p_\theta}[-\nabla_\theta E_\theta(x)]}$$

Goal of the algorithm: *Try to make the expectation of the energy match*

Warmup: learning fully observed undirected models

Goal: Learn distribution given up to constant of proportionality

$$p_\theta(x) \propto \exp(-E_\theta(x))$$

$$\nabla_\theta\left(\frac{1}{n}\sum_{i=1}^{n}\log p_\theta(x_i)\right) = \frac{1}{n}\left(\sum_i -\nabla_\theta E_\theta(x_i)\right) - \mathbb{E}_{p_\theta}[-\nabla_\theta E_\theta(x)]$$

$$\approx \mathbb{E}_{p_{data}}[-\nabla_\theta E_\theta(x)] - \mathbb{E}_{p_\theta}[-\nabla_\theta E_\theta(x)]$$

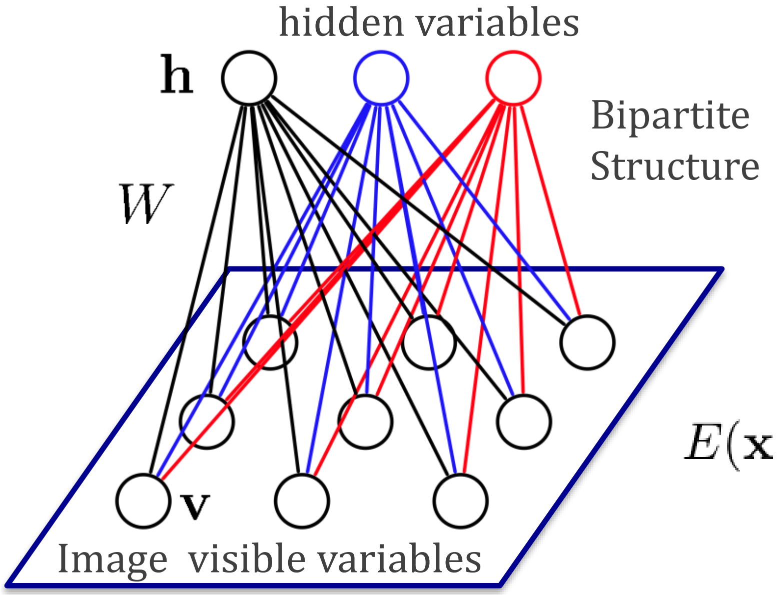
How does sampling come in: if we can sample from p_θ, and we draw samples $x_1, x_2, \ldots, x_m$ from p_θ, then:

$$\mathbb{E}_{p_\theta}[-\nabla_\theta E_\theta(x)] \approx \frac{1}{m}[-\nabla_\theta E_\theta(x_i)]$$

Restricted Boltzmann Machines

An **undirected** latent-variable model

We denote visible and hidden variables with vectors **v, h** respectively:



Visible variables $\mathbf{x} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{D}}$
are connected to hidden variables $\mathbf{h} \in \{0, 1\}^F$

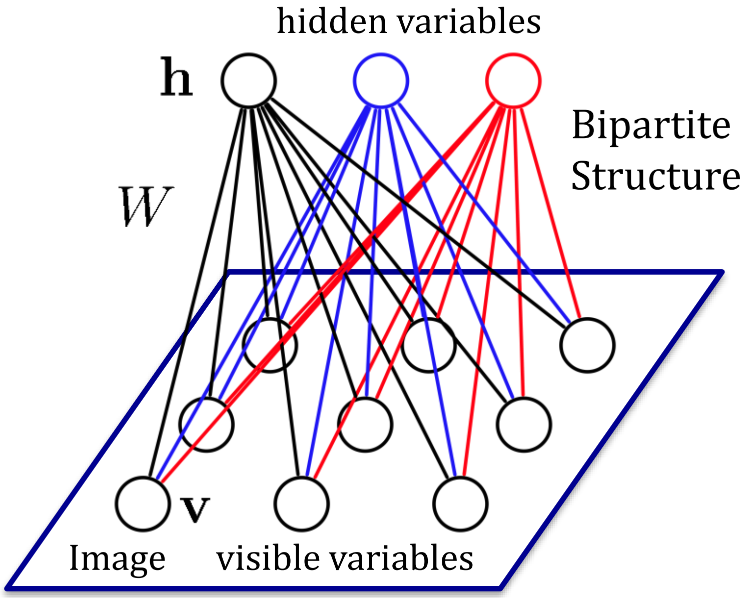
The energy of the joint configuration:

$$\begin{aligned} E(\mathbf{x}, \mathbf{h}) &= -\mathbf{h}^\top \mathbf{W} \mathbf{x} - \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \mathbf{h} \\ &= -\sum_j \sum_k W_{j,k} h_j x_k - \sum_k c_k x_k - \sum_j b_j h_j \end{aligned}$$

Probability of the joint configuration:

$$p(\mathbf{x}, \mathbf{h}) = \exp(-E(\mathbf{x}, \mathbf{h}))/Z$$

Restricted Boltzmann Machines



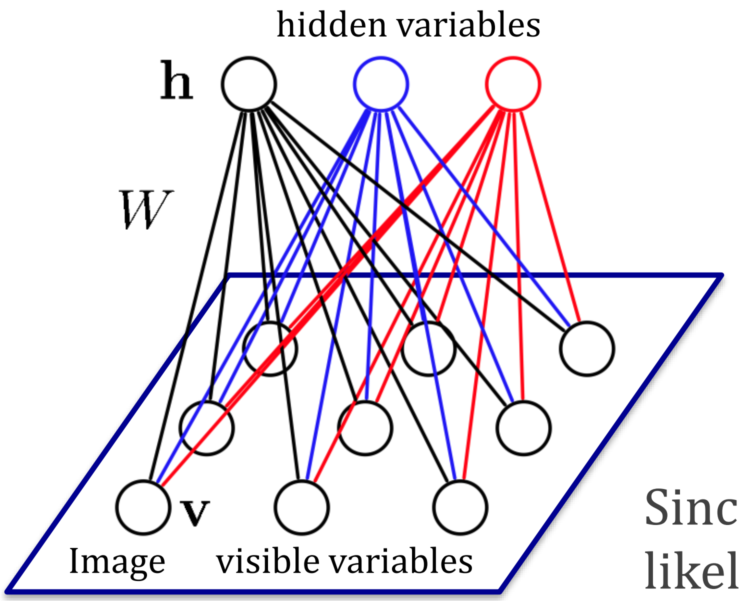
The **posterior** over the hidden variables is easy to sample from!
(Conditional independence!)

$$p(\mathbf{h}|\mathbf{x}) = \underbrace{\prod_j p(h_j|\mathbf{x})}_{\text{Factorizes}} \qquad p(h_j = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(b_j + \mathbf{W}_{j\cdot}\mathbf{x}))}$$

Similarly:

$$p(\mathbf{x}|\mathbf{h}) = \prod_k p(x_k|\mathbf{h}) \qquad p(x_k = 1|\mathbf{h}) = \frac{1}{1 + \exp(-(c_k + \mathbf{h}^\top \mathbf{W}_{\cdot k}))}$$

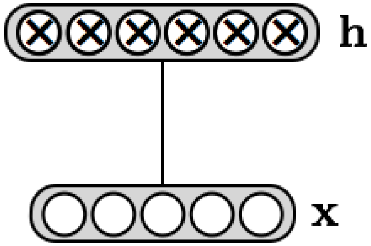
How to learn RBM's



Given data $x_1, x_2, \ldots, x_n$, solve

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} \log p_\theta(x_i)$$

Since we have latent variables, we need to express the likelihood when we marginalize out the latents:



How to learn RBM's

$$p(\mathbf{x}) = \sum_{\mathbf{h} \in \{0,1\}^H} \exp(\mathbf{h}^\top \mathbf{W} \mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h}) / Z$$

How to learn RBM’s

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{h}\in\{0,1\}^H} \exp(\mathbf{h}^\top \mathbf{W}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h})/Z \\ &= \exp(\mathbf{c}^\top \mathbf{x}) \sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp\left(\sum_j h_j \mathbf{W}_{j.}\mathbf{x} + b_j h_j\right)/Z \end{aligned}$$

How to learn RBM's

$$\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}\in\{0,1\}^H} \exp(\mathbf{h}^\top \mathbf{W}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h})/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp\left(\sum_j h_j \mathbf{W}_{j\cdot}\mathbf{x} + b_j h_j\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \left(\sum_{h_1\in\{0,1\}} \exp(h_1 \mathbf{W}_{1\cdot}\mathbf{x} + b_1 h_1)\right) \ldots \left(\sum_{h_H\in\{0,1\}} \exp(h_H \mathbf{W}_{H\cdot}\mathbf{x} + b_H h_H)\right)/Z
\end{aligned}$$

How to learn RBM's

$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}\in\{0,1\}^H} \exp(\mathbf{h}^\top \mathbf{W}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h})/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp\left(\sum_j h_j \mathbf{W}_{j\cdot}\mathbf{x} + b_j h_j\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \left(\sum_{h_1\in\{0,1\}} \exp(h_1 \mathbf{W}_{1\cdot}\mathbf{x} + b_1 h_1)\right) \dots \left(\sum_{h_H\in\{0,1\}} \exp(h_H \mathbf{W}_{H\cdot}\mathbf{x} + b_H h_H)\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x})\left(1 + \exp(b_1 + \mathbf{W}_{1\cdot}\mathbf{x})\right) \dots \left(1 + \exp(b_H + \mathbf{W}_{H\cdot}\mathbf{x})\right)/Z
\end{aligned}
$$

How to learn RBM's

$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}\in\{0,1\}^H} \exp(\mathbf{h}^\top \mathbf{W}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h})/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp\left(\sum_j h_j \mathbf{W}_{j\cdot}\mathbf{x} + b_j h_j\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \left(\sum_{h_1\in\{0,1\}} \exp(h_1 \mathbf{W}_{1\cdot}\mathbf{x} + b_1 h_1)\right) \dots \left(\sum_{h_H\in\{0,1\}} \exp(h_H \mathbf{W}_{H\cdot}\mathbf{x} + b_H h_H)\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \left(1 + \exp(b_1 + \mathbf{W}_{1\cdot}\mathbf{x})\right) \dots \left(1 + \exp(b_H + \mathbf{W}_{H\cdot}\mathbf{x})\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \exp(\log(1 + \exp(b_1 + \mathbf{W}_{1\cdot}\mathbf{x}))) \dots \exp(\log(1 + \exp(b_H + \mathbf{W}_{H\cdot}\mathbf{x})))/Z
\end{aligned}
$$

How to learn RBM’s

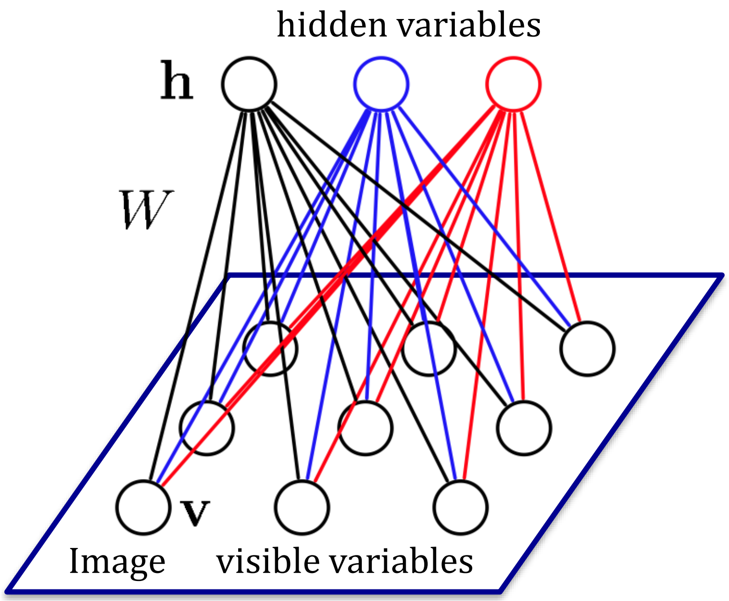
$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}\in\{0,1\}^H} \exp(\mathbf{h}^\top \mathbf{W}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h})/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp\left(\sum_j h_j \mathbf{W}_{j\cdot}\mathbf{x} + b_j h_j\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \left(\sum_{h_1\in\{0,1\}} \exp(h_1 \mathbf{W}_{1\cdot}\mathbf{x} + b_1 h_1)\right) \dots \left(\sum_{h_H\in\{0,1\}} \exp(h_H \mathbf{W}_{H\cdot}\mathbf{x} + b_H h_H)\right)/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x})\,(1+\exp(b_1+\mathbf{W}_{1\cdot}\mathbf{x}))\dots(1+\exp(b_H+\mathbf{W}_{H\cdot}\mathbf{x}))\,/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x})\exp(\log(1+\exp(b_1+\mathbf{W}_{1\cdot}\mathbf{x})))\dots\exp(\log(1+\exp(b_H+\mathbf{W}_{H\cdot}\mathbf{x})))/Z \\
&= \exp\left(\underbrace{\mathbf{c}^\top \mathbf{x} + \sum_{j=1}^{H} \log(1+\exp(b_j+\mathbf{W}_{j\cdot}\mathbf{x}))}_{=\,\mathrm{F}(\mathbf{x})}\right)/Z
\end{aligned}
$$

How to learn RBM’s

$$\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{h}\in\{0,1\}^H} \exp(\mathbf{h}^\top \mathbf{W}\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h})/Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp\left(\sum_j h_j \mathbf{W}_{j\cdot}\mathbf{x} + b_j h_j\right) /Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \left(\sum_{h_1\in\{0,1\}} \exp(h_1 \mathbf{W}_{1\cdot}\mathbf{x} + b_1 h_1)\right) \ldots \left(\sum_{h_H\in\{0,1\}} \exp(h_H \mathbf{W}_{H\cdot}\mathbf{x} + b_H h_H)\right) /Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \left(1 + \exp(b_1 + \mathbf{W}_{1\cdot}\mathbf{x})\right) \ldots \left(1 + \exp(b_H + \mathbf{W}_{H\cdot}\mathbf{x})\right) /Z \\
&= \exp(\mathbf{c}^\top \mathbf{x}) \exp(\log(1 + \exp(b_1 + \mathbf{W}_{1\cdot}\mathbf{x}))) \ldots \exp(\log(1 + \exp(b_H + \mathbf{W}_{H\cdot}\mathbf{x})))/Z \\
&= \exp\left(\underbrace{\mathbf{c}^\top \mathbf{x} + \sum_{j=1}^{H} \log(1 + \exp(b_j + \mathbf{W}_{j\cdot}\mathbf{x}))}_{= \mathrm{F}(\mathbf{x})}\right) /Z
\end{aligned}$$

$$= \exp\big(\mathrm{F}(\mathbf{x})\big) /\mathrm{Z}$$

How to learn RBM's



Given data $x_1, x_2, \dots, x_n$, solve

$$\max_{\theta\in\Theta} \sum_{i=1}^{n} \log p_\theta(x_i)$$

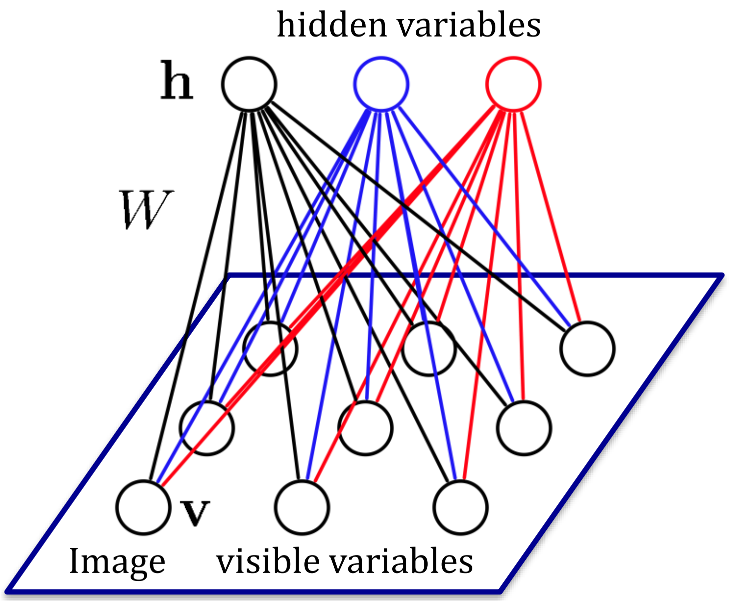
With this reduction, the undirected model calculations imply:

$$\nabla_\theta\left(\frac{1}{n}\sum_{i=1}^{n}\log p_\theta(x_i)\right) = \frac{1}{n}\left(\sum_i -\nabla_\theta F_\theta(x_i)\right) - \mathbb{E}_{p_\theta}[-\nabla_\theta F_\theta(x)]$$

$$\nabla_{\mathrm{w}_{\mathrm{ij}}}\mathrm{F}_\theta(\mathbf{x}) = \nabla_{\mathrm{w}_{\mathrm{ij}}}(\mathbf{c}^{\mathrm{T}}\mathbf{x} + \textstyle\sum_{\mathrm{j}=1}^{\mathrm{H}}\log\left(1+\exp\left(\mathrm{b}_{\mathrm{j}}+\mathbf{W}_{\mathrm{j}}.\mathbf{x}\right)\right) = \frac{\exp\left(\mathrm{b}_{\mathrm{j}}+\mathbf{W}_{\mathrm{j}}.\mathbf{x}\right)}{1+\exp\left(\mathrm{b}_{\mathrm{j}}+\mathbf{W}_{\mathrm{j}}.\mathbf{x}\right)}\mathbf{x}_{\mathrm{i}}$$

$$= \frac{1}{1+\exp\left(-\left(\mathrm{b}_{\mathrm{j}}+\mathbf{W}_{\mathrm{j}}.\mathbf{x}\right)\right)}\mathbf{x}_{\mathrm{i}} = \mathrm{P}\left(\mathbf{h}_{\mathrm{j}} = 1\middle|\mathbf{x}\right)\mathbf{x}_{\mathrm{i}}$$

How to learn RBM's



Given data $x_1, x_2, \dots, x_n$, solve

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} \log p_\theta(x_i)$$

With this reduction, the undirected model calculations imply:

$$\nabla_\theta \left(\frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i) \right) = \frac{1}{n} \left(\sum_i -\nabla_\theta F_\theta(x_i) \right) - \mathbb{E}_{p_\theta}[-\nabla_\theta F_\theta(x)]$$

$\nabla_{\mathrm{W_{ij}}} \mathrm{F}_\theta(\mathbf{x}) = \mathrm{P}(\mathbf{h}_\mathrm{j} = 1|\mathbf{x})\ \mathbf{x}$_i $\Rightarrow$ $\nabla_\mathbf{W} \mathrm{F}_\theta(\mathbf{x}) = \mathbf{h}(\mathbf{x})\ \mathbf{x}^\mathrm{T}$

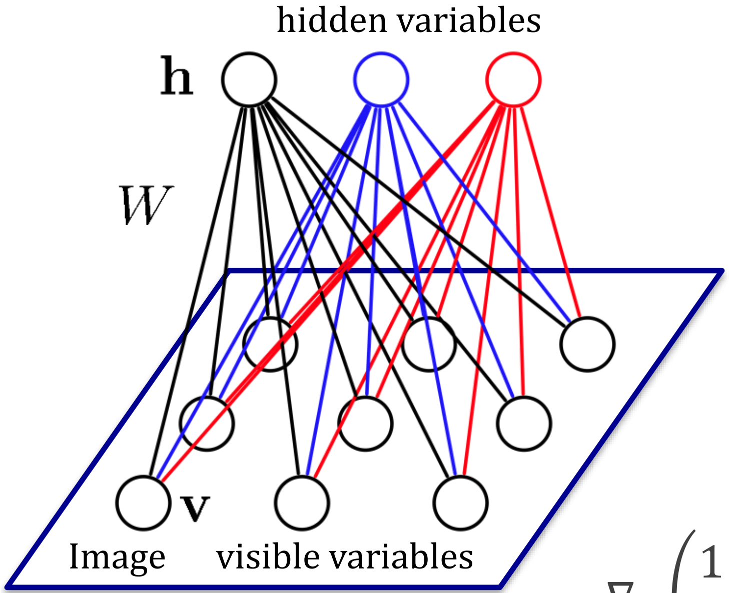
$\nabla_{\boldsymbol{b}} \mathrm{F}_\theta(\mathbf{x}) = \mathbf{h}(\mathbf{x})$

$\nabla_\mathrm{c} \mathrm{F}_\theta(\mathbf{x}) = \mathbf{x}$

$$\mathbf{h}(\mathbf{x}) \overset{\text{def}}{=} \begin{pmatrix} p(h_1 = 1|\mathbf{x}) \\ \dots \\ p(h_H = 1|\mathbf{x}) \end{pmatrix}$$

$$= \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})$$

How to learn RBM's



Given data $x_1, x_2, \dots, x_n$, solve

$$\max_{\theta \in \Theta} \sum_{i=1}^{n} \log p_\theta(x_i)$$

$$\nabla_\theta \left(\frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i) \right) = \frac{1}{n} \left(\sum_{i} -\nabla_\theta F_\theta(x_i) \right) - \mathbb{E}_{p_\theta}[-\nabla_\theta F_\theta(x)]$$

The hard term is again: $\mathbb{E}_{p_\theta}[-\nabla_\theta E_\theta(x)]$ --- we need to draw samples from p_θ

We will draw samples using a Markov random walk: **Gibbs sampler**!

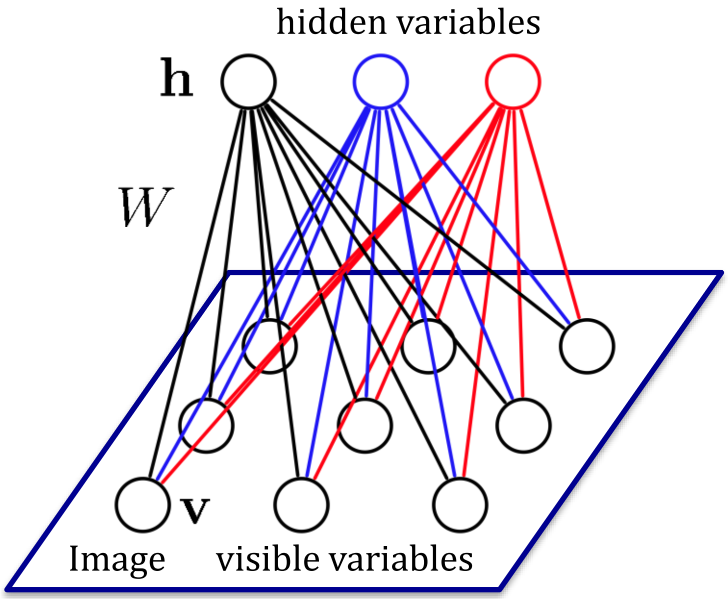
Gibbs sampling

Consider sampling a distribution over n variables $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$, s.t. each of the conditional distributions $P(x_i|\boldsymbol{x}_{-i})$ is easy to sample. :

A common way to do this is using **Gibbs sampling**:

```
Repeat:
    Let current state be $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$
    Pick $i \in [n]$ uniformly at random.
    Sample x $\sim P(X_i = x|\boldsymbol{x}_{-i})$
    Update state to $\boldsymbol{y} = (x_1, x_2, \dots, x_{i-1}, x, x_{i+1}, \dots, x_n)$
```

Gibbs sampling for RBM's

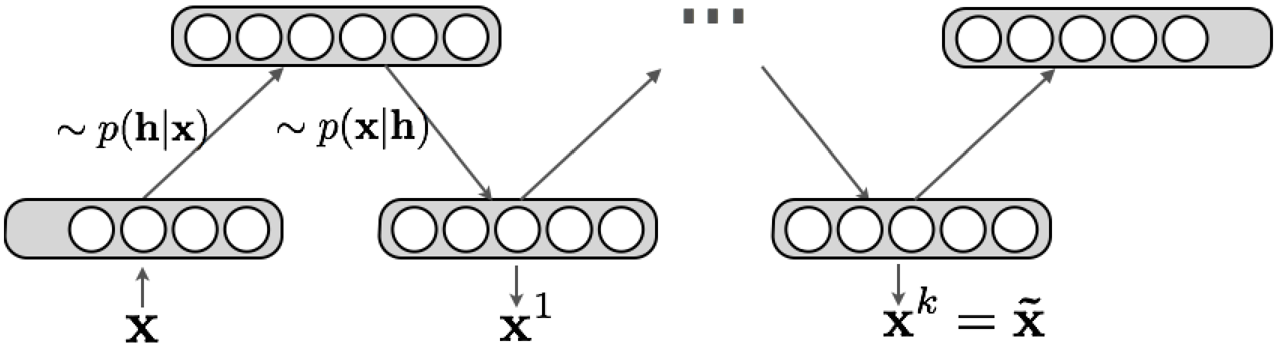


Repeat:

Sample $\mathbf{h} \sim P(\boldsymbol{h}|\boldsymbol{v})$

Sample $\mathbf{v} \sim P(\boldsymbol{v}|\boldsymbol{h})$

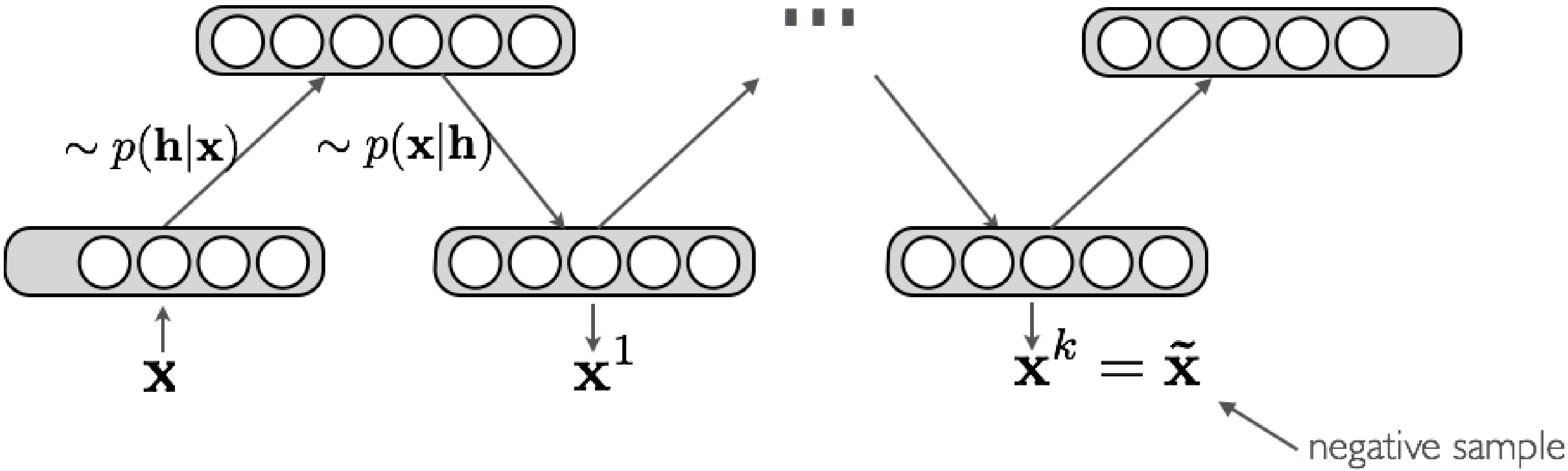
Pictorially:


$\sim p(\mathbf{h}|\mathbf{x})$
$\sim p(\mathbf{x}|\mathbf{h})$
...
$\mathbf{x}$
$\mathbf{x}^1$
$\mathbf{x}^k = \tilde{\mathbf{x}}$

Contrastive Divergence

Key idea behind Contrastive Divergence:

- Replace the expectation by a point estimate at $\tilde{\mathbf{x}}$
- Obtain the point $\tilde{\mathbf{x}}$ by Gibbs sampling
- Start sampling chain at $\mathbf{x}$


$\sim p(\mathbf{h}|\mathbf{x})$
$\sim p(\mathbf{x}|\mathbf{h})$
...
$\mathbf{x}$
$\mathbf{x}^1$
$\mathbf{x}^k = \tilde{\mathbf{x}}$
negative sample

k is often taken to be just 1.

Hinton, Neural Computation, 2002

CD-k Algorithm

For each training example $\mathbf{x}$

- Generate a negative sample $\tilde{\mathbf{x}}$ using k steps of Gibbs sampling, starting at the data point $\mathbf{x}$
- Update model parameters:

$$\left.\begin{aligned}\mathbf{W} &\Longleftarrow \mathbf{W}+\alpha\left(\mathbf{h}(\mathbf{x}\quad)\,\mathbf{x}\quad^{\top}-\mathbf{h}(\tilde{\mathbf{x}})\,\tilde{\mathbf{x}}^{\top}\right)\\ \mathbf{b} &\Longleftarrow \mathbf{b}+\alpha\left(\mathbf{h}(\mathbf{x}\quad)-\mathbf{h}(\tilde{\mathbf{x}})\right)\\ \mathbf{c} &\Longleftarrow \mathbf{c}+\alpha\left(\mathbf{x}\quad-\tilde{\mathbf{x}}\right)\end{aligned}\right\}$$

Gradients we derived before

- Go back to 1 until stopping criteria

Step size

CD-k Algorithm

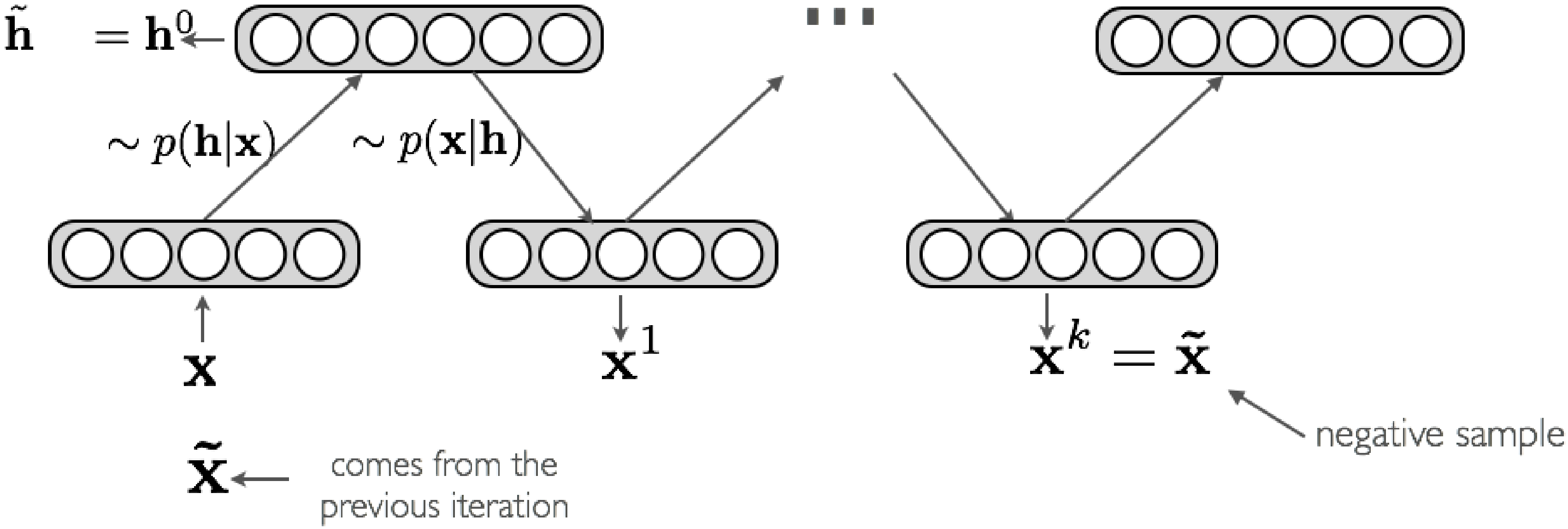
- CD-k: contrastive divergence with k iterations of Gibbs sampling

- In general, the bigger k is, the less biased the estimate of the gradient will be

- In practice, k=1 works well for learning good features and for pre-training

Persistent CD

Idea: instead of initializing the chain to $\mathbf{x}$, initialize the chain to the negative sample of the last iteration

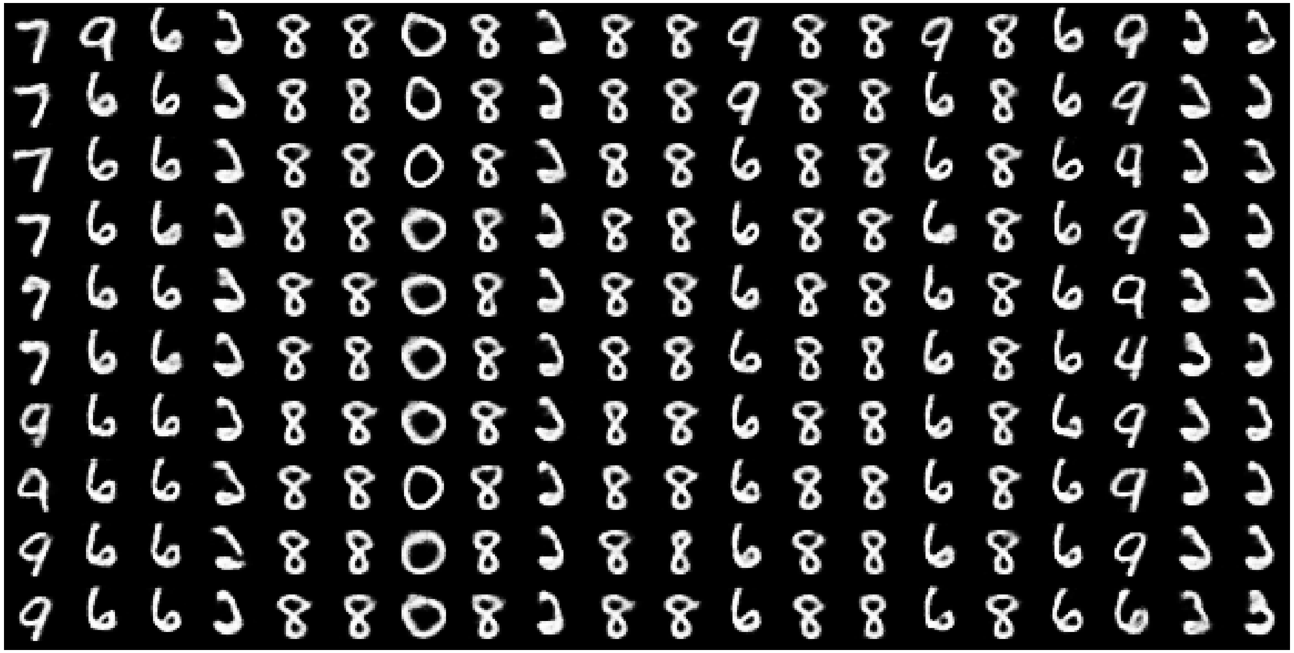


$\tilde{\mathbf{h}} = \mathbf{h}^0$
$\sim p(\mathbf{h}|\mathbf{x})$
$\sim p(\mathbf{x}|\mathbf{h})$
...
$\mathbf{x}$
$\mathbf{x}^1$
$\mathbf{x}^k = \tilde{\mathbf{x}}$
negative sample
$\tilde{\mathbf{x}}$
comes from the previous iteration

Tieleman, ICML, 2008

Example: MNIST

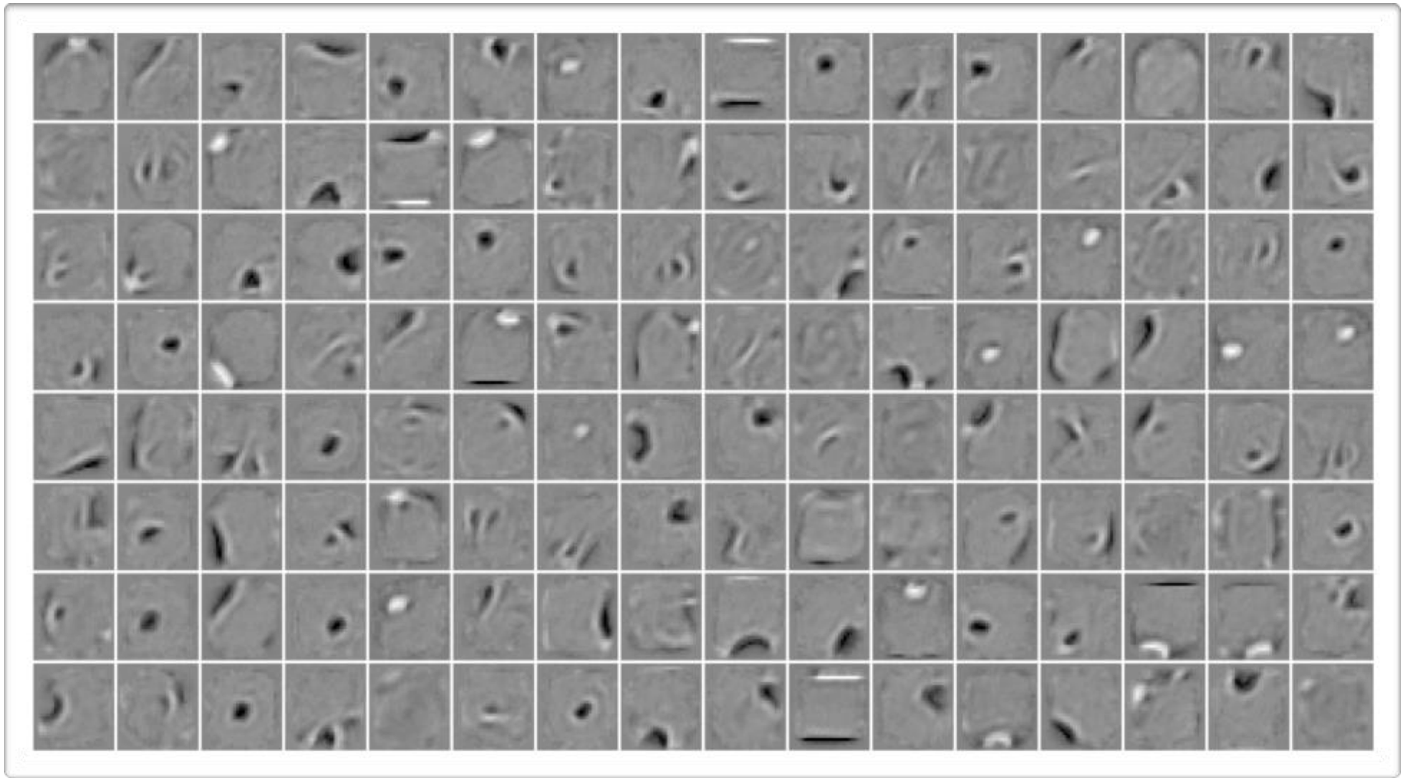
MNIST dataset:



Each row is small set of "initial points", after which next row is gotten by running 1000 Gibbs steps.

Learned Features

MNIST dataset:



(Larochelle et al., JMLR 2009)

Tricks and Debugging

Unfortunately, it is not easy to debug training RBMs (e.g. using gradient checks)

We instead rely on approximate “tricks”

- ➢ we plot the average stochastic reconstruction $||\mathbf{x}^{(t)} - \tilde{\mathbf{x}}||^2$ and see if it tends to decrease
- ➢ for inputs that correspond to image, we visualize the connection coming into each hidden unit as if it was an image
- ➢ gives an idea of the type of visual feature each hidden unit detects
- ➢ we can also try to approximate the partition function Z and see whether the (approximated) NLL decreases

(Salakhutdinov, Murray, ICML 2008)