

Undirected Graphical Models

Aaron Courville, Université de Montréal

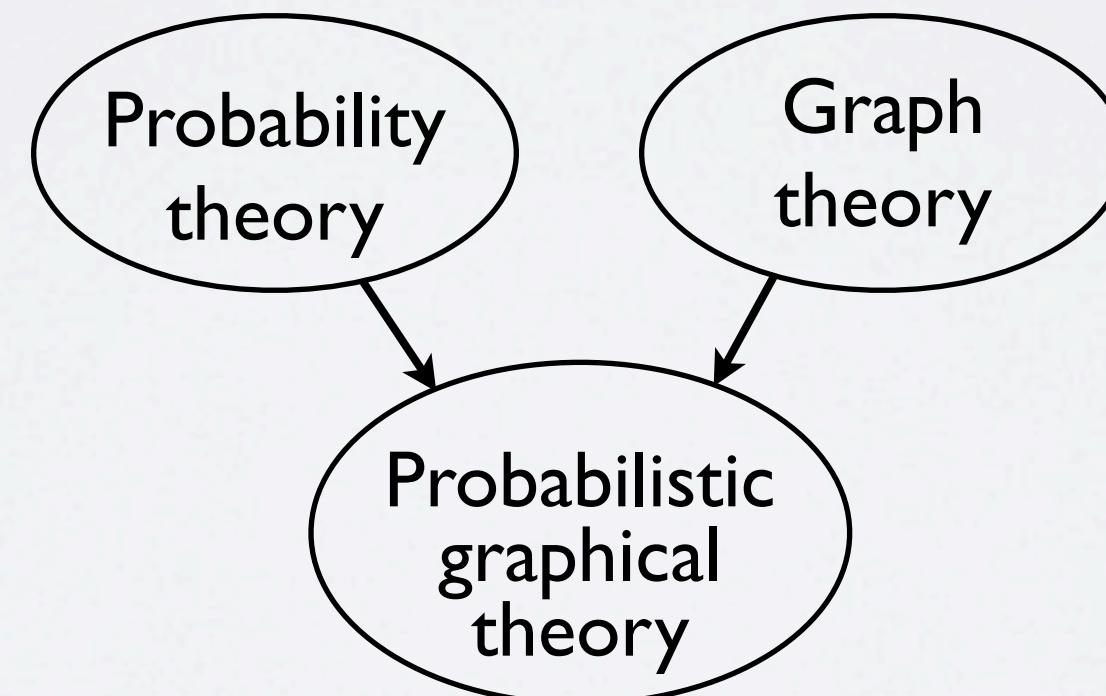
(UNDIRECTED) GRAPHICAL MODELS

Overview:

- Directed versus undirected graphical models
- Conditional independence
- Energy function formalism
- Maximum likelihood learning
- Restricted Boltzmann Machine
- Spike-and-slab RBM

Probabilistic Graphical Models

- Graphs endowed with a probability distribution
 - **Nodes** represent random variables and the **edges** encode conditional independence assumptions
- Graphical model express **sets of conditional independence** via graph structure (and conditional independence is useful)
- Graph structure plus associated parameters define joint probability distribution of the set of nodes/variables



Probabilistic Graphical Models

- Graphical models come in two main flavors:
 1. Directed graphical models (a.k.a Bayes Net, Belief Networks):
 - Consists of a set of nodes with arrows (directed edges) between some of the nodes
 - Arrows encode factorized conditional probability distributions
 2. Undirected graphical models (a.k.a Markov random fields):
 - Consists of a set of nodes with undirected edges between some of the nodes
 - Edges (or more accurately the lack of edges) encode conditional independence.
- Today, we will focus almost exclusively on undirected graphs.

PROBABILITY REVIEW: CONDITIONAL INDEPENDENCE

Definition: X is conditionally independent of Y given Z if the probability distribution governing X is independent of the value of Y , given the value of Z : for all (i, j, k)

$$\begin{aligned} P(X = x_i, Y = y_j \mid Z = z_k) &= P(X = x_i \mid Z = z_k)P(Y = y_j \mid Z = z_k) \\ P(X, Y \mid Z) &= P(X \mid Z)P(Y \mid Z) \end{aligned}$$

Or equivalently (by the product rule):

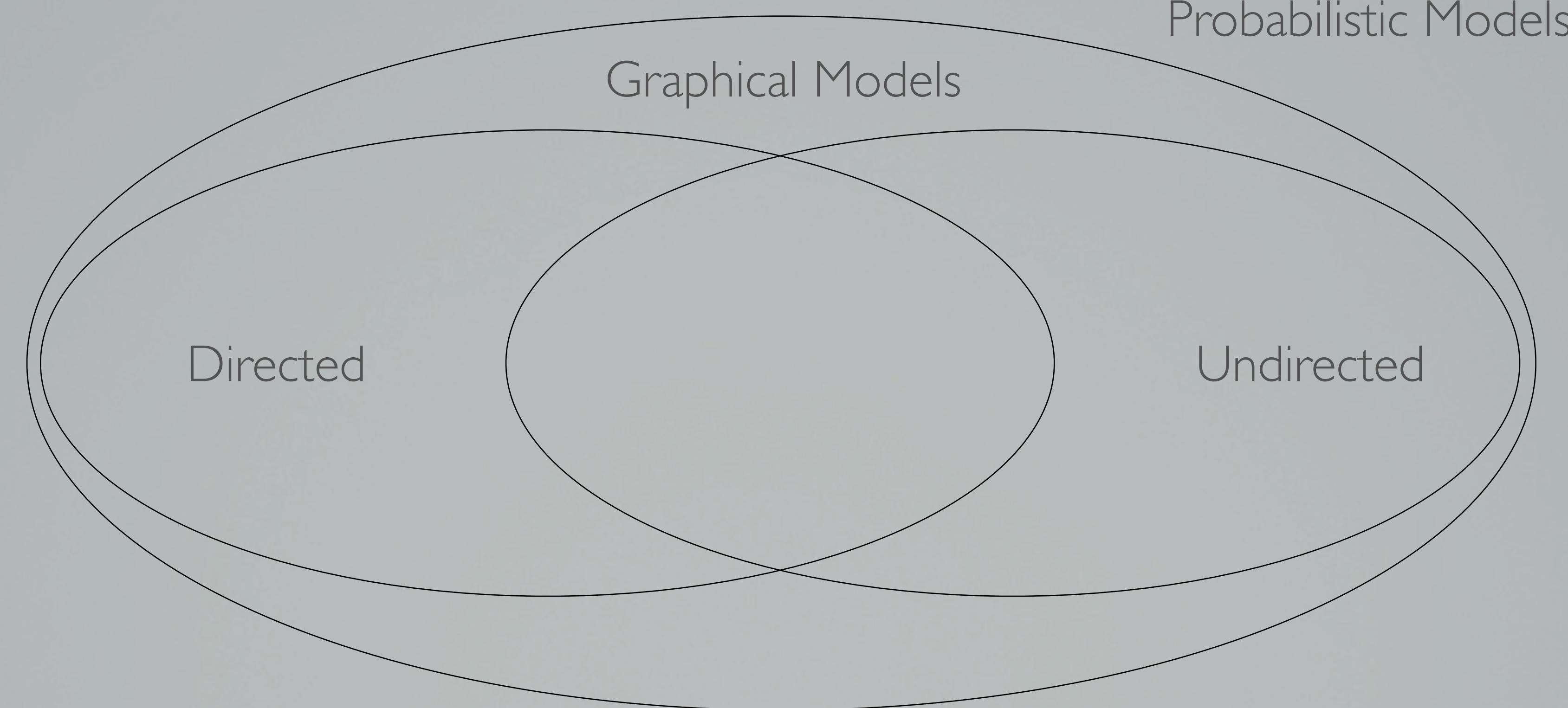
$$P(X \mid Y, Z) = P(X \mid Z) \quad P(Y \mid X, Z) = P(Y \mid Z)$$

Why? Recall from the probability product rule

$$P(X, Y, Z) = P(X \mid Y, Z)P(Y \mid Z)P(Z) = P(X \mid Z)P(Y \mid Z)P(Z)$$

Example: $P(\text{Thunder} \mid \text{Rain}, \text{Lightning}) = P(\text{Thunder} \mid \text{Lightning})$

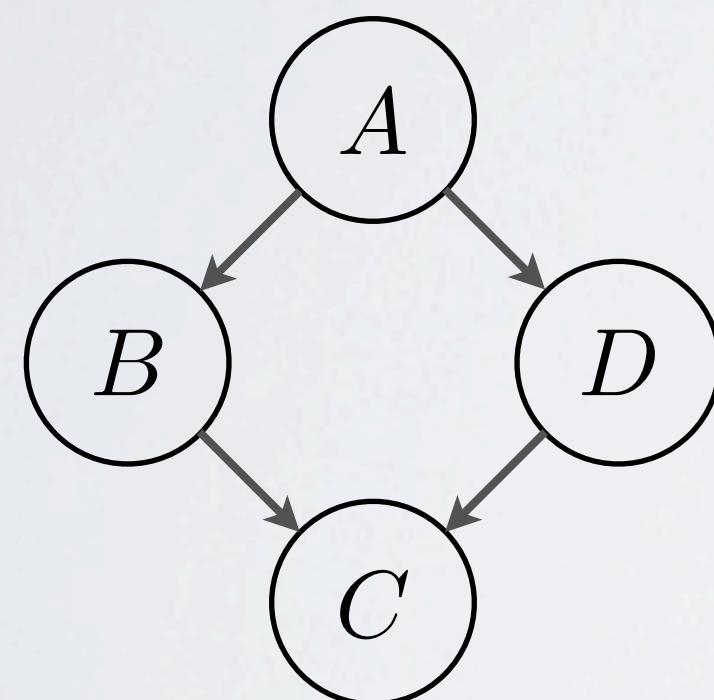
TYPES OF GRAPHICAL MODELS



REPRESENTING CONDITIONAL INDEPENDENCE

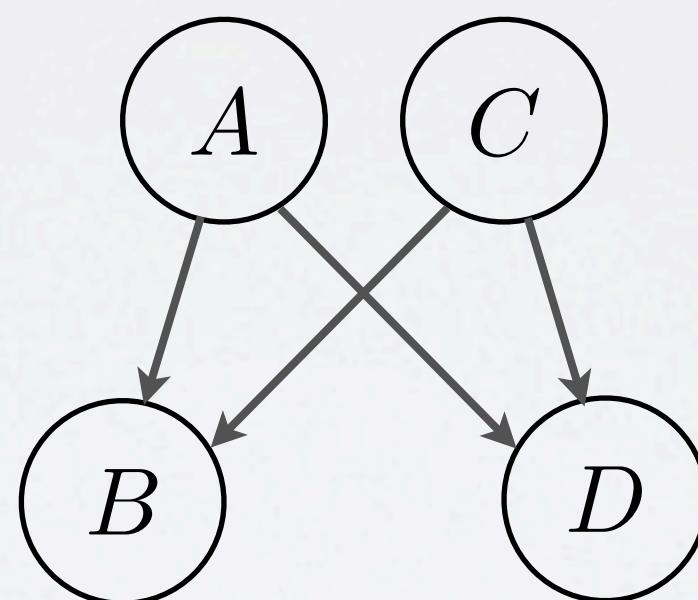
Some conditional independencies cannot be represented by directed graphical models:

- Consider 4 variables: A, B, C, D
- How do we represent the conditional independences: $(A \perp C | B, D)$
 $(B \perp D | A, C)$



$$(A \perp C | B, D)$$

$$(B \perp D | A, C)$$

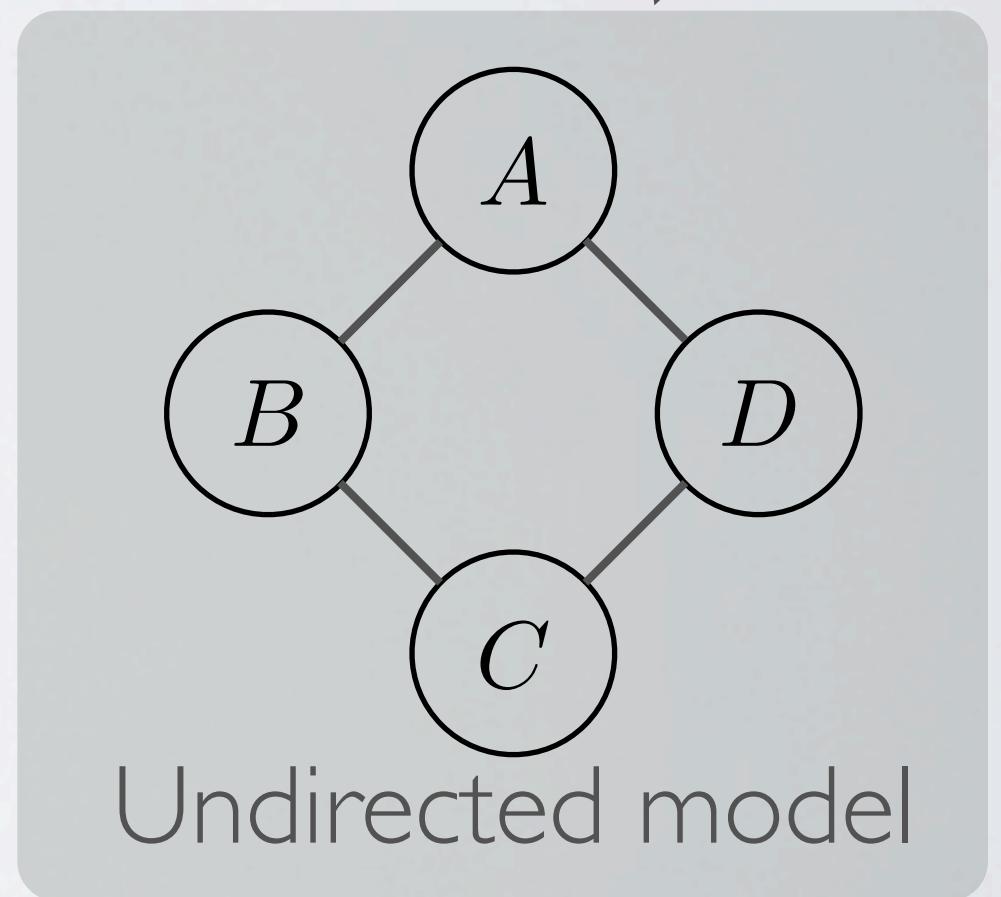


$$(A \perp C)$$

$$(B \perp D | A, C)$$

$$(A \perp C | B, D)$$

$$(B \perp D | A, C)$$



WHY UNDIRECTED GRAPHICAL MODELS?

Sometime its awkward to model phenomena with directed models

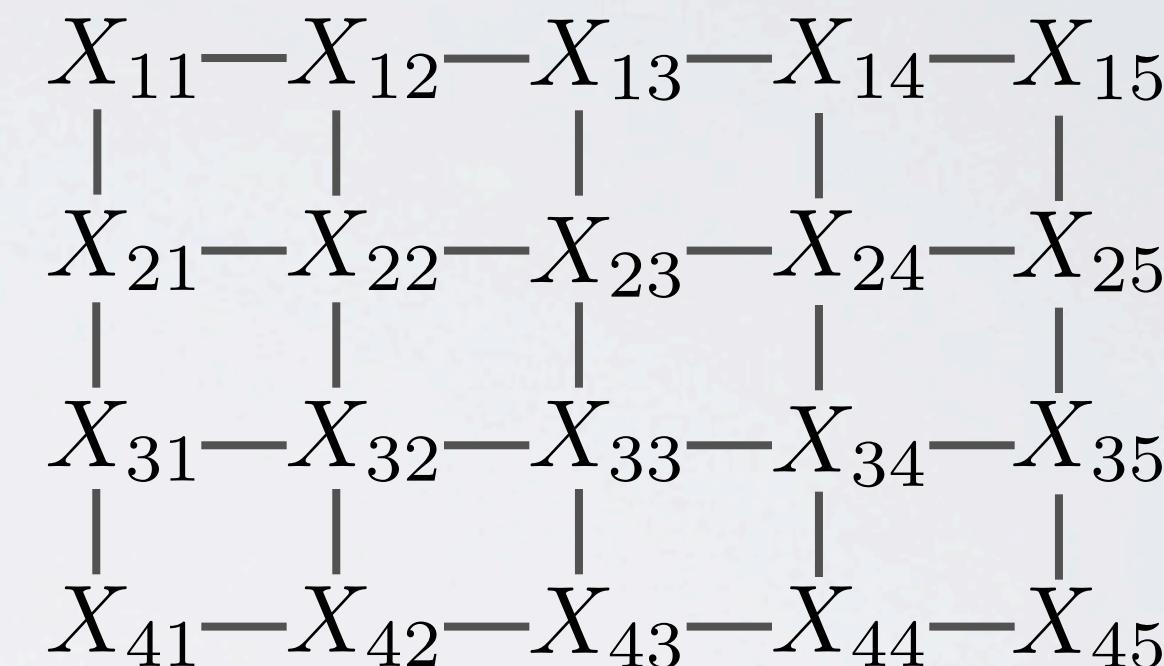
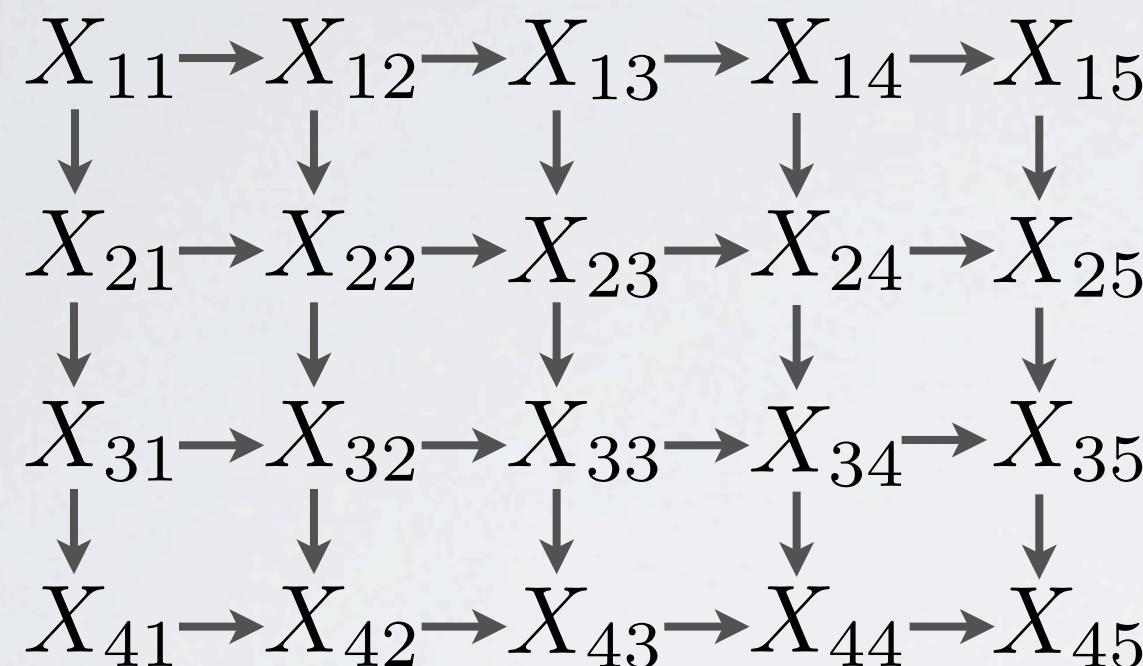
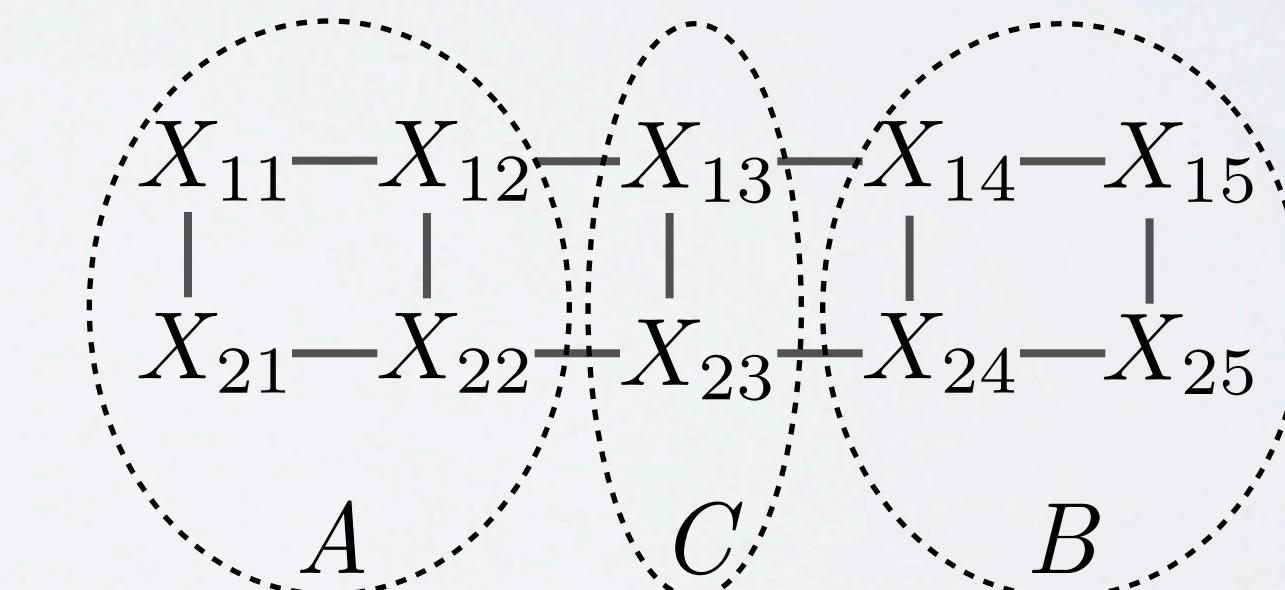


Image from “CRF as RNN Semantic Image Segmentation Live Demo” (<http://www.robots.ox.ac.uk/~szheng/crfasrnndemo/>)

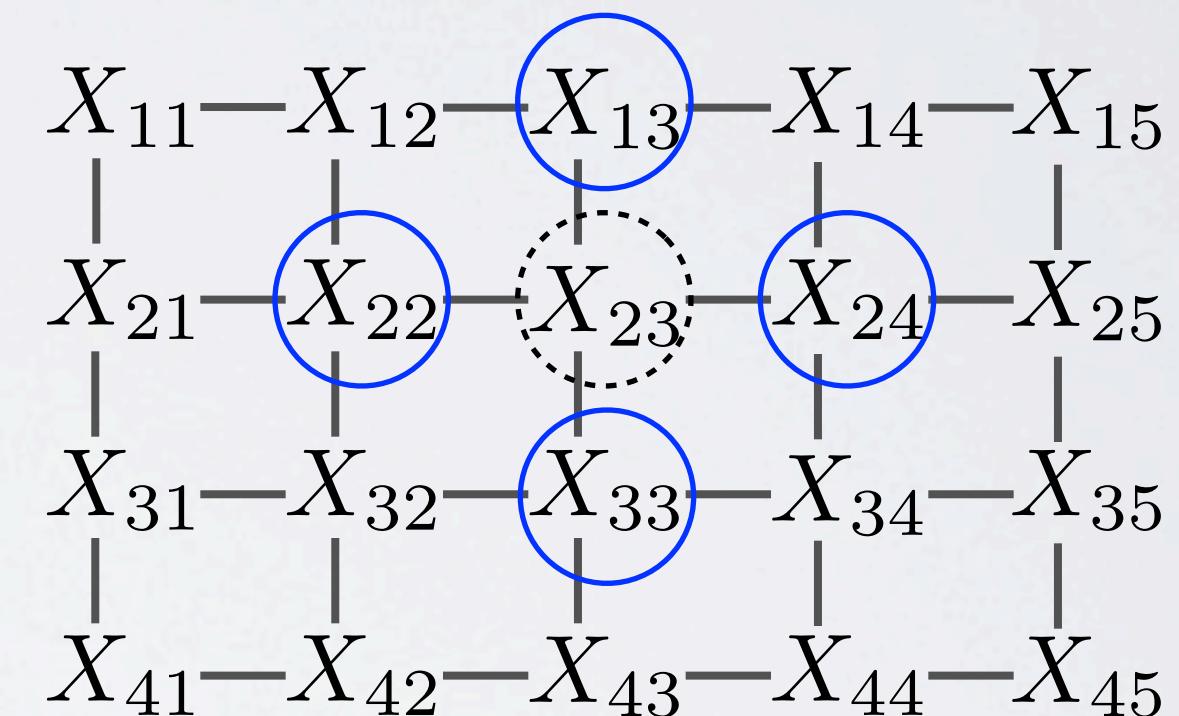
CONDITIONAL INDEPENDENCE PROPERTIES

- Undirected graphical models:
 - ▶ Conditional independence encoded by simple graph separation.
 - ▶ Formally, consider 3 sets of nodes: A , B and C , we say $\mathbf{x}_A \perp \mathbf{x}_B \mid \mathbf{x}_C$ iff C separates A and B in the graph.
 - ▶ C separates A and B in the graph: If we remove all nodes in C , there is no path from A to B in the graph.



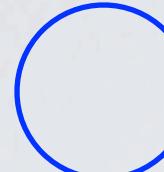
MARKOV BLANKET

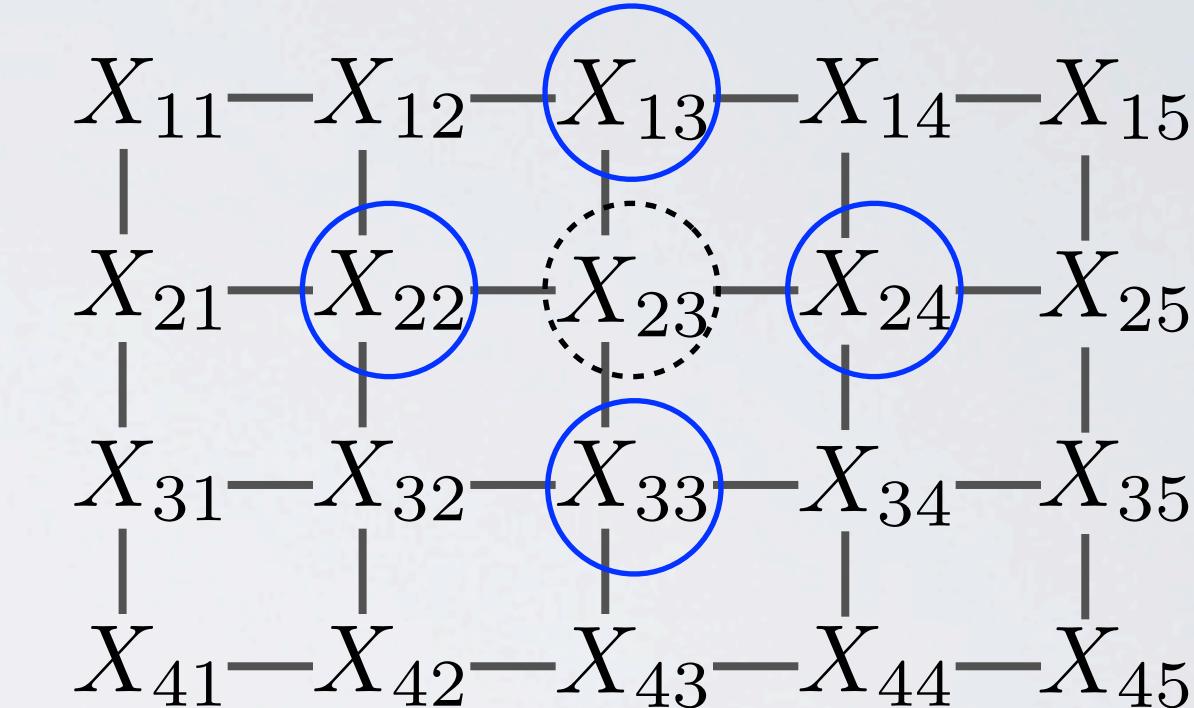
- Markov Blanket: For a given node x , the Markov Blanket is the smallest set of nodes which renders x conditionally independent of all other nodes in the graph.
- Markov blanket of the 2-d lattice MRF:



RELATING DIRECTED AND UNDIRECTED MODELS

- Markov blanket of the 2-d lattice MRF:

 — neighbours of X_{23}

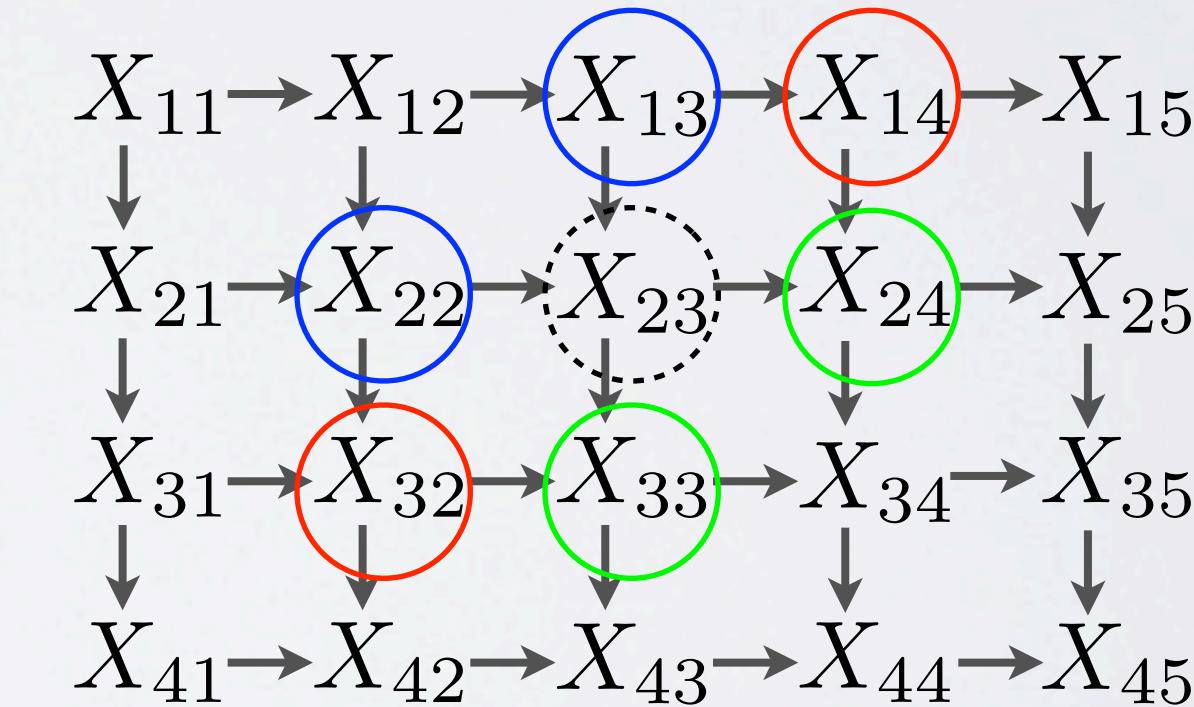


- Markov blanket of the 2-d causal MRF:

 — parents of X_{23}

 — children of X_{23}

 — parents of children of X_{23}

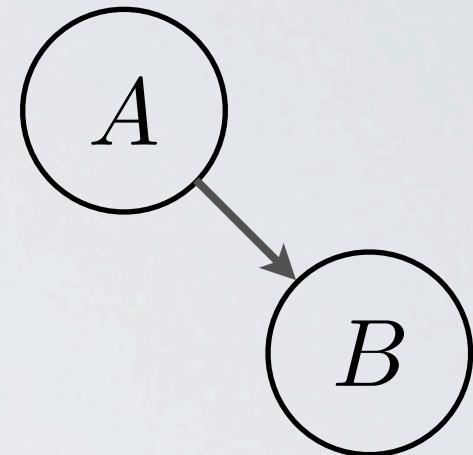


PARAMETERIZING DIRECTED GRAPHICAL MODELS

Directed graphical models:

- Parameterized by local conditional probability densities (CPDs)

$$P(A \mid B)$$



- Joint distributions are given as products of CPDs:

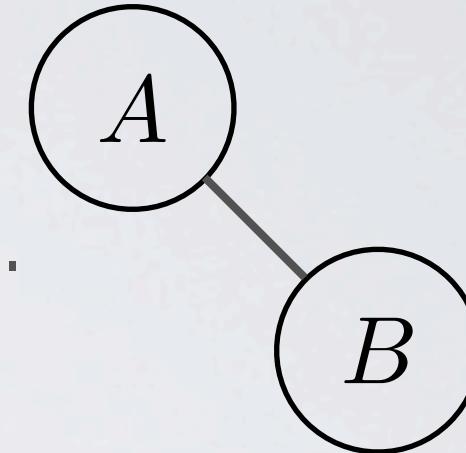
$$P(X_1, \dots, X_N) = \prod_{i=0}^N P(X_i \mid X_{\text{parents}(i)})$$

PARAMETERIZING MARKOV NETWORKS: FACTORS

Undirected graphical models:

- Parameterized by symmetric **factors** or **potential functions**.

$$\phi(A, B)$$



- Generalizes both the CPD and the joint distribution.
- Note: unlike the CPDs, the potential function are not required to normalize.
- **Definition:** Let \mathcal{C} be a set of cliques. For each $c \in \mathcal{C}$, we define a factor (also called potential function or clique potential) ϕ_c as a nonnegative function

$$\phi_c(\mathbf{x}_c) \rightarrow \mathbb{R}$$

where \mathbf{x}_c is the set of variables in clique c .

PARAMETERIZING MARKOV NETWORKS: JOINT DISTRIBUTION

- Joint distribution given by a normalized product of factors:

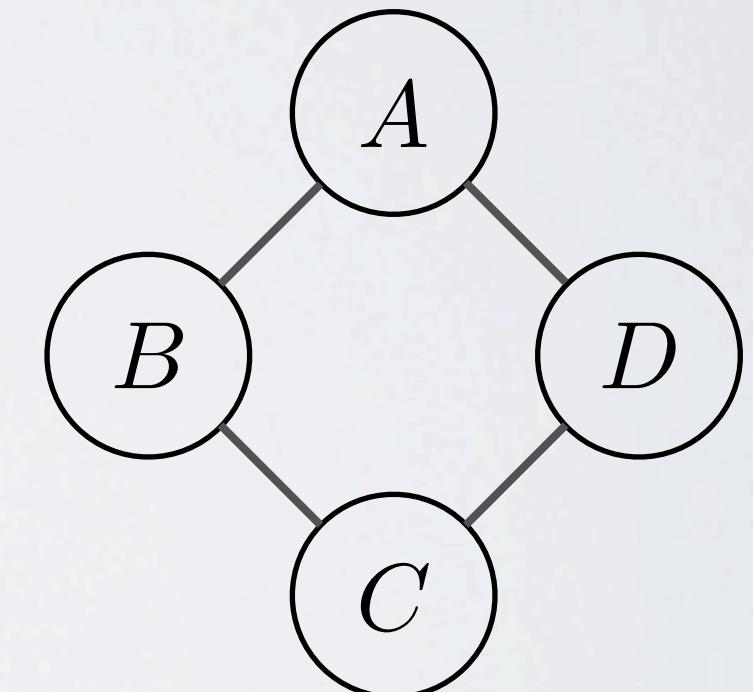
$$P(x_1, \dots, x_n) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c)$$

- Z is the **partition function**, it's the normalization constant: $Z = \sum_{x_1, \dots, x_n} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{x}_c)$

- Our 4 variable example:

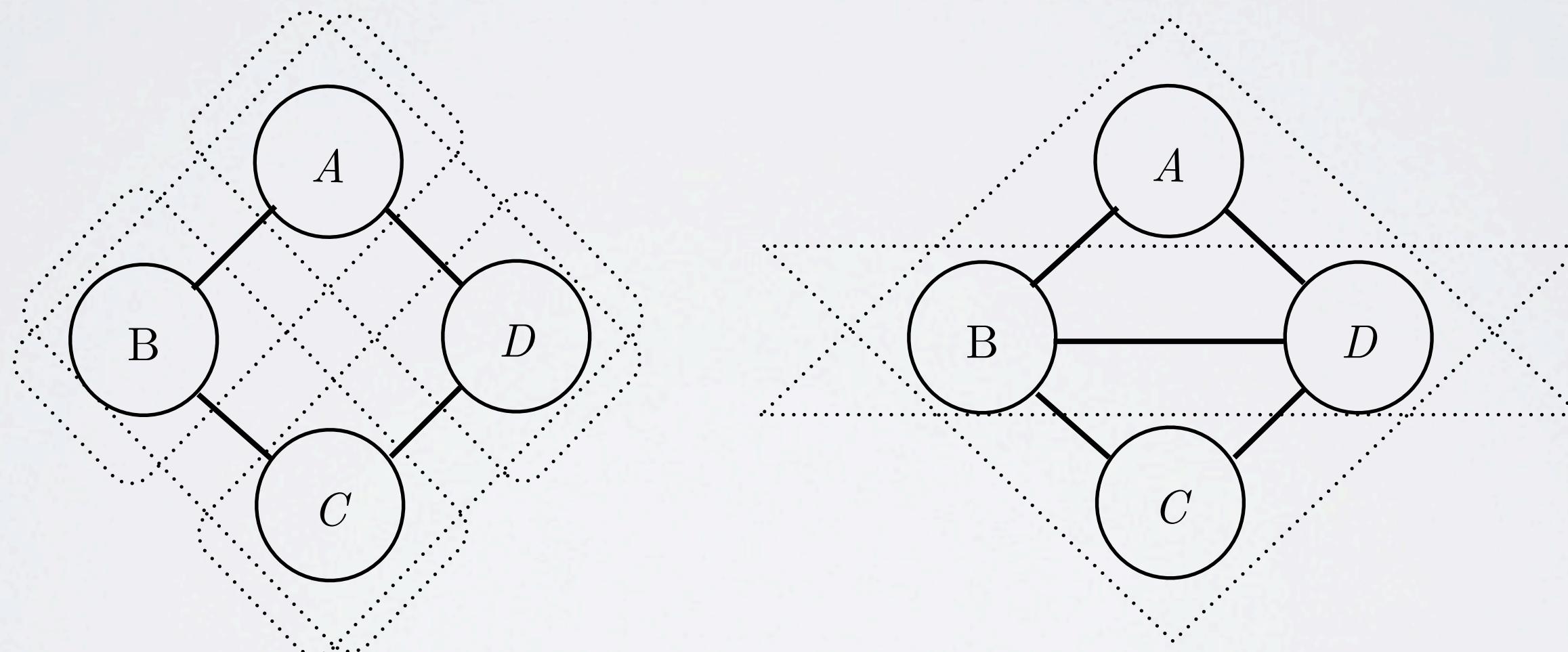
$$P(a, b, c, d) = \frac{1}{Z} \phi_1(a, b) \phi_2(b, c) \phi_3(c, d) \phi_4(d, a)$$

$$Z = \sum_{a, b, c, d} \phi_1(a, b) \phi_2(b, c) \phi_3(c, d) \phi_4(d, a)$$



CLIQUEs AND MAXIMAL CLIQUEs

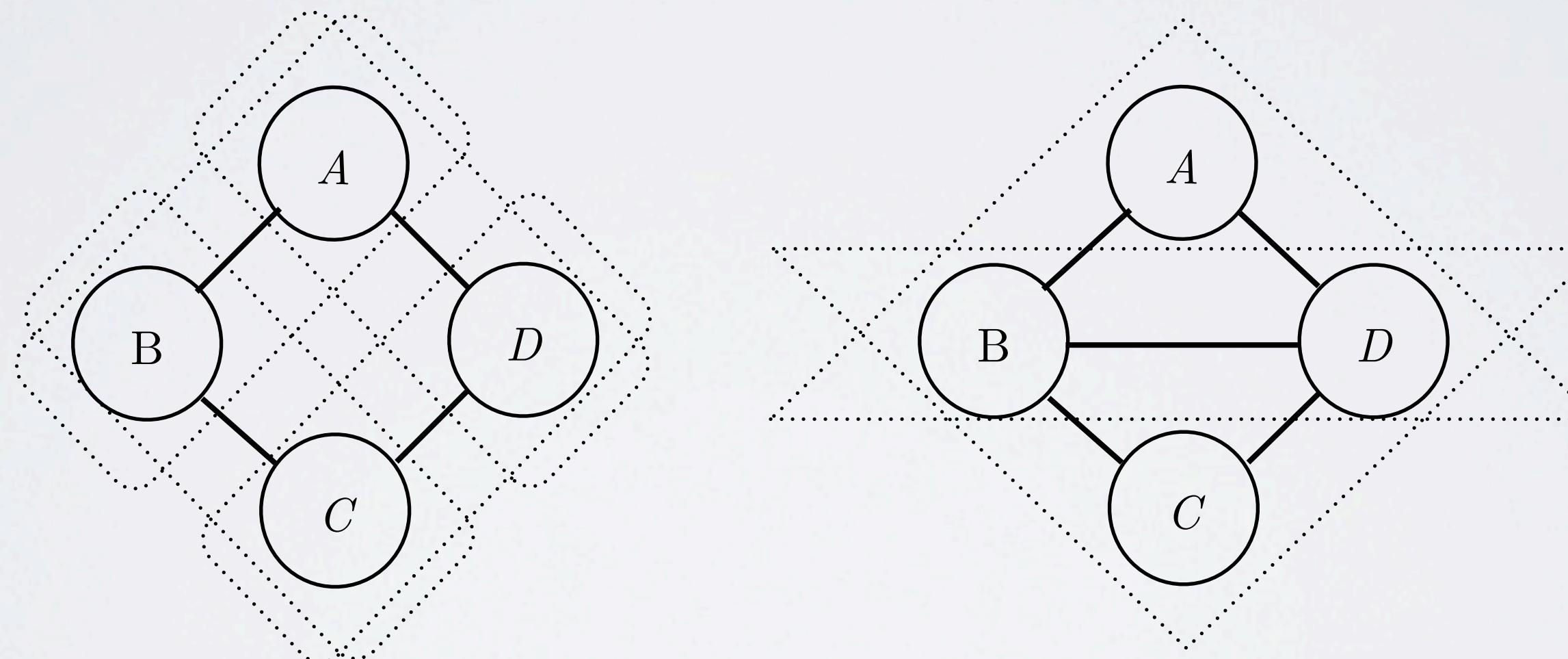
- What is a **clique**? A subset of nodes who's **induced subgraph** is **complete**
- A **maximal clique** is one where you cannot add any more nodes and remain a clique



Examples of maximal cliques.

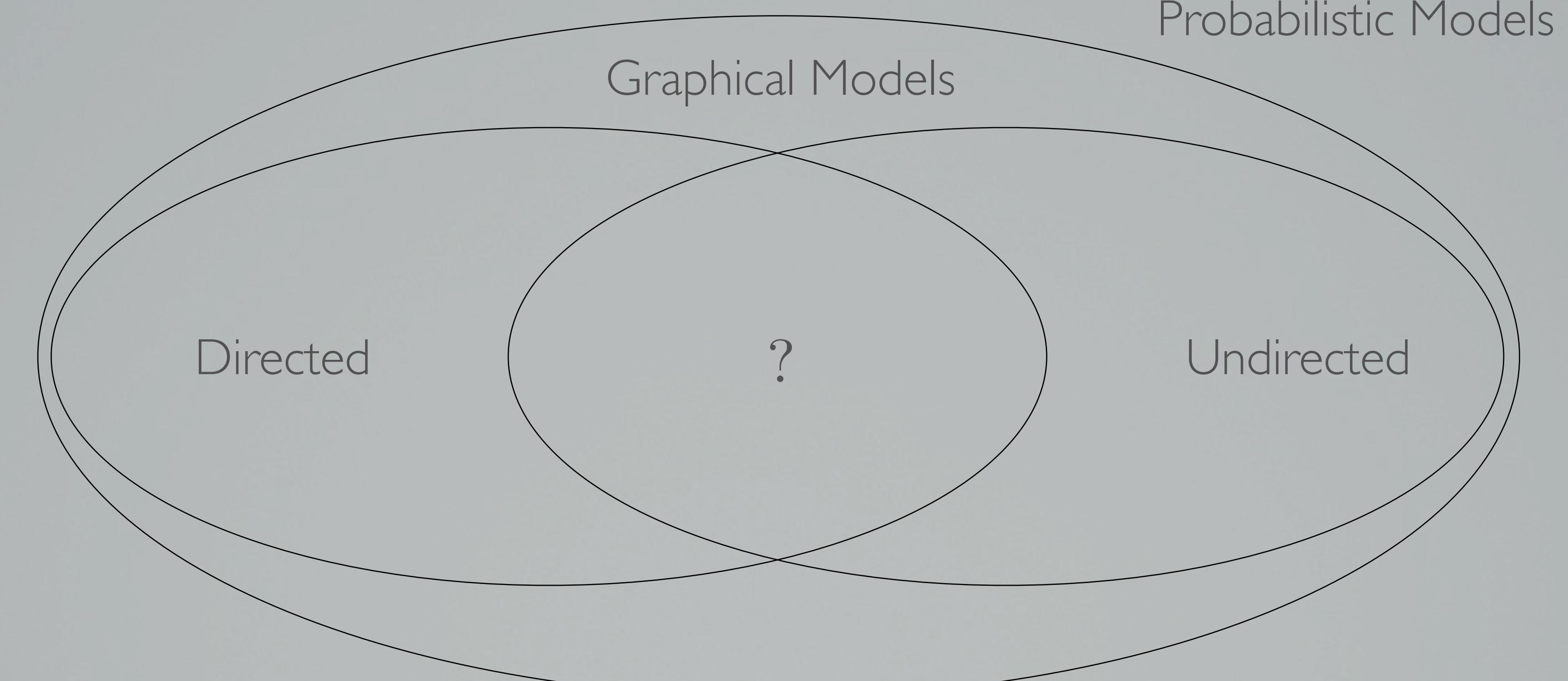
OF GRAPHS AND DISTRIBUTIONS

- Interesting fact: any positive distribution whose conditional independencies can be represented with an undirected graph can be parameterized by a product of factors (**Hammersley-Clifford theorem**).



Examples of maximal cliques.

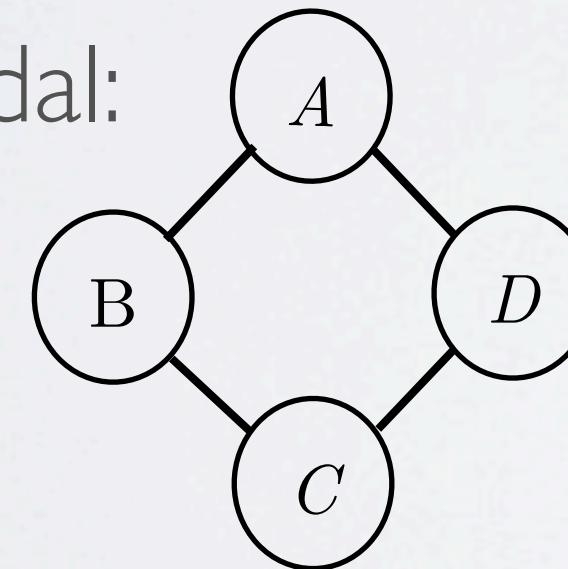
TYPES OF GRAPHICAL MODELS



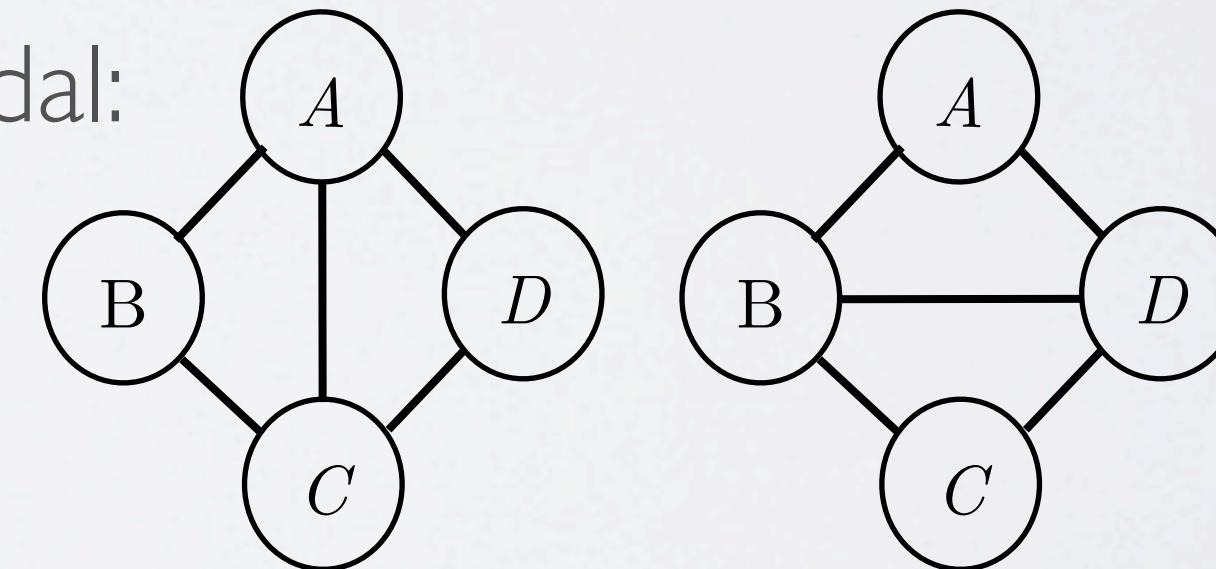
RELATING DIRECTED AND UNDIRECTED MODELS

- What kind of probability models can be encoded by both a directed and an undirected graphical model.
 - Answer: any probability mode whose cond. indep. relations are consistent with a chordal graph.
- Chordal graph: All undirected cycles of four or more vertices have a chord.
- Chord: Edge that is not part of the cycle but connects two vertices of the cycle.

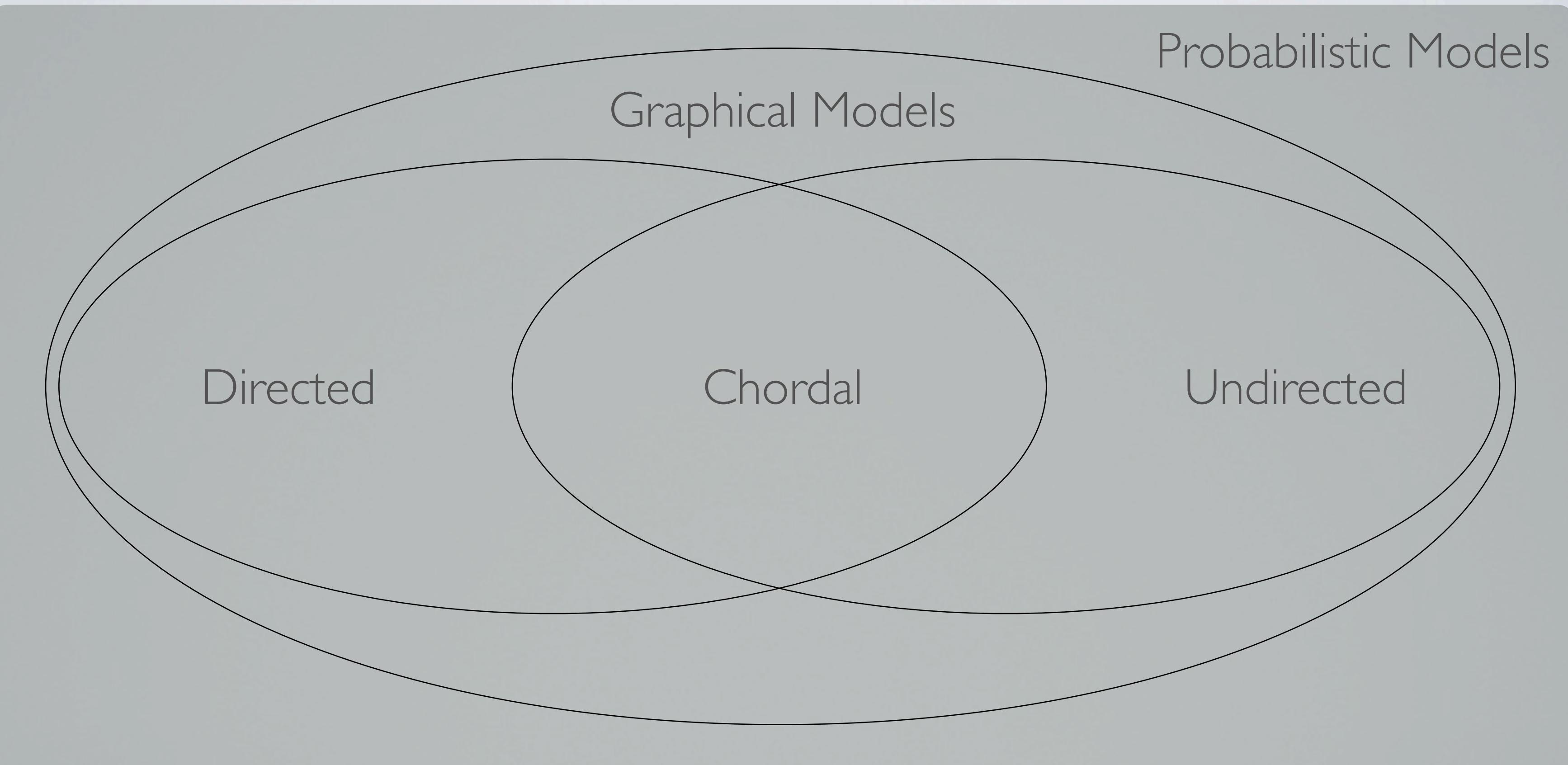
Not chordal:



Chordal:



TYPES OF GRAPHICAL MODELS



ENERGY-BASED MODELS

- The undirected models that most interest us are **energy-based models**.
 - We reformulate the factor $\phi(\mathbf{x}_c)$ in log-space: $\phi(\mathbf{x}_c) = \exp(-\epsilon(\mathbf{x}_c))$ or alternatively, $\epsilon(\mathbf{x}_c) = -\log \phi(\mathbf{x}_c)$, where $\epsilon(\mathbf{x}_c) \in \mathbb{R}$.
 - Energy-based formulation of joint dist: $P(x_1, \dots, x_n) = \frac{1}{Z} \exp(-E(x_1, \dots, x_n))$
- $E(x_1, \dots, x_n)$ is called the energy function.
- $$= \frac{1}{Z} \exp \left(- \sum_{c \in \mathcal{C}} \epsilon_c(\mathbf{x}_c) \right)$$

where $Z = \sum_{x_1} \cdots \sum_{x_n} \exp [-E(x_1, \dots, x_n)]$

LOG-LINEAR MODEL

- Log-linear models are a type of energy-based model with a particular, linear, parametrization.
- In log-linear models, for clique c , the corresponding element of the energy function $\epsilon_c(\mathbf{x}_c)$ is composed of:
 1. A parameter w_c
 2. A feature of the observed data $f_c(\mathbf{x}_c)$
- The joint distribution is given by $P(x_1, \dots, x_n) = \frac{1}{Z} \exp \left(- \sum_{c \in C} w_c f_c(\mathbf{x}_c) \right)$

MAXIMUM LIKELIHOOD LEARNING

- Maximum likelihood learning in the context of a fully observable MRF.

$$\begin{aligned}
 \boldsymbol{w}^{\text{ML}} &= \operatorname{argmax}_{\boldsymbol{w}} \log \prod_{i=1}^{\mathcal{D}} p(\boldsymbol{x}^{(i)}; \boldsymbol{w}) \\
 &= \operatorname{argmax}_{\boldsymbol{w}} \sum_{i=1}^{\mathcal{D}} \left(\sum_c \log \phi_c(\boldsymbol{x}_c^{(i)}; w_c) - \log Z(\boldsymbol{w}) \right) \\
 &= \operatorname{argmax}_{\boldsymbol{w}} \left[\left(\sum_{i=1}^{\mathcal{D}} \sum_c \log \phi_c(\boldsymbol{x}_c^{(i)}; w_c) \right) - |\mathcal{D}| \log Z(\boldsymbol{w}) \right] \\
 &= \operatorname{argmax}_{\boldsymbol{w}} \left[\left(\sum_{i=1}^{\mathcal{D}} \sum_c w_c f_c(\boldsymbol{x}_c^{(i)}) \right) - |\mathcal{D}| \log Z(\boldsymbol{w}) \right]
 \end{aligned}$$

log-linear model 

 decomposes over the cliques 

 does not decompose 

MAXIMUM LIKELIHOOD LEARNING

- In general, there is no closed form solution for the optimal parameters.

$$\log Z(\mathbf{w}) = \log \sum_{\mathbf{x}} \exp \left(\sum_c w_c f_c(\mathbf{x}_c) \right)$$

- We can compute a gradient of the partition function.

$$\begin{aligned} \frac{\partial}{\partial w_c} \log Z(\mathbf{w}) &= \frac{\partial}{\partial w_c} \log \left(\sum_{\mathbf{x}} \exp \left(\sum_{c'} w_{c'} f_{c'}(\mathbf{x}_{c'}) \right) \right) \\ &= \frac{\sum_{\mathbf{x}_c} \exp(w_c f_c(\mathbf{x}_c)) f_c(\mathbf{x}_c)}{\sum_{\mathbf{x}_c} \exp(\sum_c w_c f_c(\mathbf{x}_c))} \\ &= \mathbb{E}_{p(\mathbf{x}_c; w_c)} [f_c(\mathbf{x}_c)] \end{aligned}$$

MAXIMUM LIKELIHOOD LEARNING

- The gradient of the log-likelihood

$$\frac{\partial}{\partial w_c} \sum_{i=1}^{\mathcal{D}} \log p(\mathbf{x}^{(i)}; \mathbf{w}) = \frac{\partial}{\partial w_c} \left[\left(\sum_{i=1}^{\mathcal{D}} \sum_{c'} w_{c'} f_{c'}(\mathbf{x}_{c'}^{(i)}) \right) - \mathcal{D} \log Z(\mathbf{w}) \right]$$

$$= \left(\sum_{i=1}^{\mathcal{D}} f_c(\mathbf{x}_c^{(i)}) \right) - \mathcal{D} \frac{\partial}{\partial w_c} \log Z(\mathbf{w})$$

$$= \mathcal{D} \mathbb{E}_{p(\text{data})} [f_c(\mathbf{x}_c)] - \mathcal{D} \mathbb{E}_{p(\mathbf{x}_c; \mathbf{w}_c)} [f_c(\mathbf{x}_c)]$$



data term

often tractable
(e.g. fully observable \mathbf{x})



model term

often intractable
(e.g. fully observable \mathbf{x})

MAXIMUM LIKELIHOOD LEARNING

- How do we estimate the intractable expectation from the model term (due to the partition function contribution of the gradient)?

$$\frac{\partial}{\partial w_c} \log Z(\mathbf{w}) = \mathbb{E}_{p(\mathbf{x}_c; w_c)} [f_c(\mathbf{x}_c)]$$

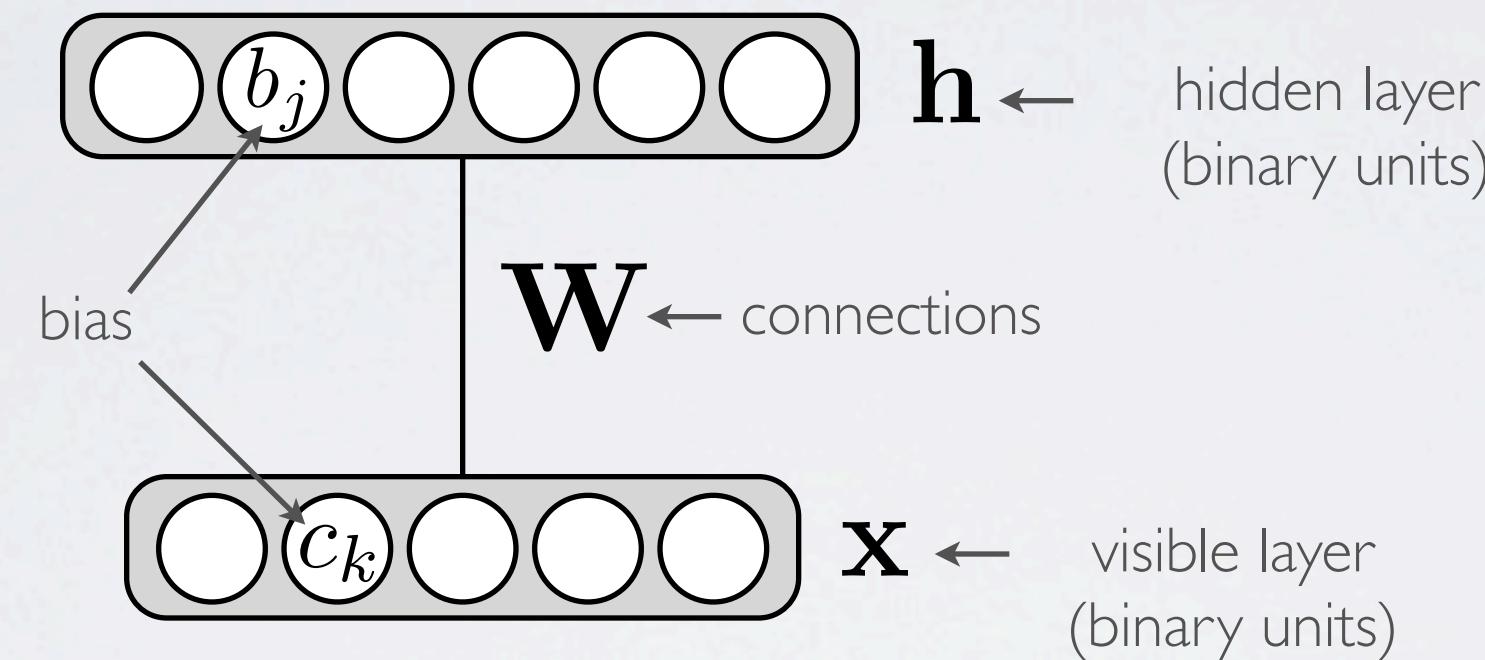
- We can sometimes use approximations methods such as pseudo-likelihood.
- More generally we can use **Monte Carlo (i.e. sampling)** methods to estimate this expectation.
 - This comes with some disadvantages, more on this when we discuss restricted Boltzmann machines.

Restricted Boltzmann Machines

An Introduction

RESTRICTED BOLTZMANN MACHINE

Topics: RBM, visible layer, hidden layer, energy function



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

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

partition function
(intractable)

MARKOV NETWORKVIEW

Topics: Markov network (with vector nodes)

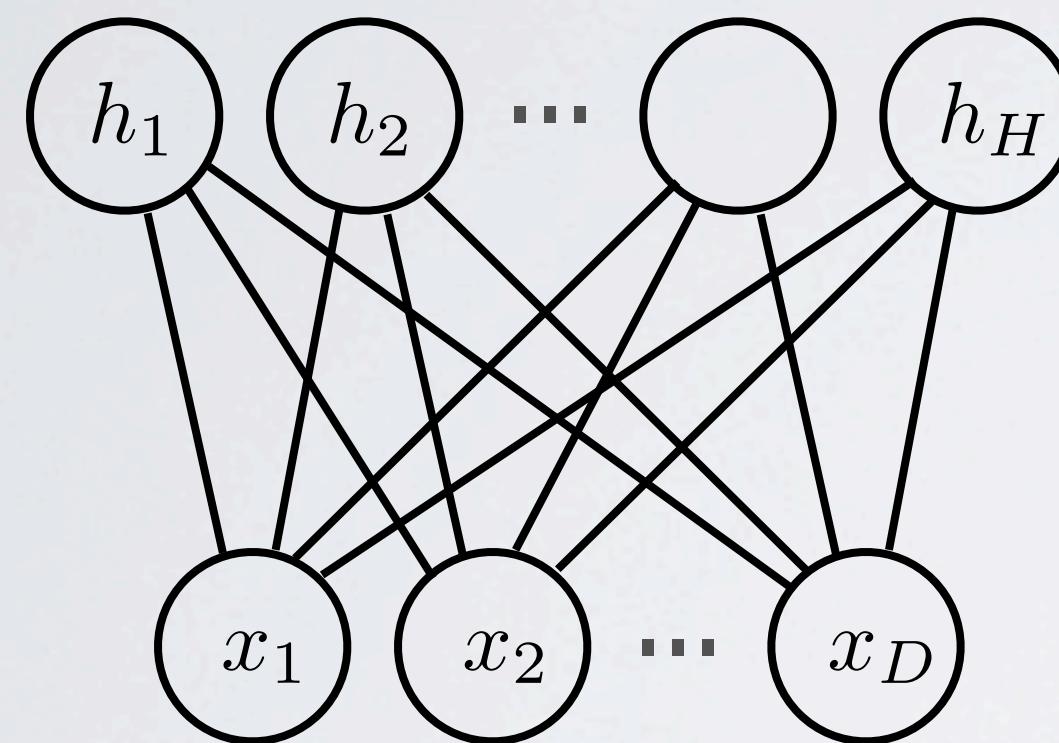


$$\begin{aligned}
 p(\mathbf{x}, \mathbf{h}) &= \exp(-E(\mathbf{x}, \mathbf{h}))/Z \\
 &= \exp(\mathbf{h}^\top \mathbf{W} \mathbf{x} + \mathbf{c}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{h})/Z \\
 &= \underbrace{\exp(\mathbf{h}^\top \mathbf{W} \mathbf{x}) \exp(\mathbf{c}^\top \mathbf{x}) \exp(\mathbf{b}^\top \mathbf{h})}_{\text{factors}}/Z
 \end{aligned}$$

- The notation based on an energy function is simply an alternative to the representation as the product of factors

MARKOV NETWORKVIEW

Topics: Markov network (with scalar nodes)

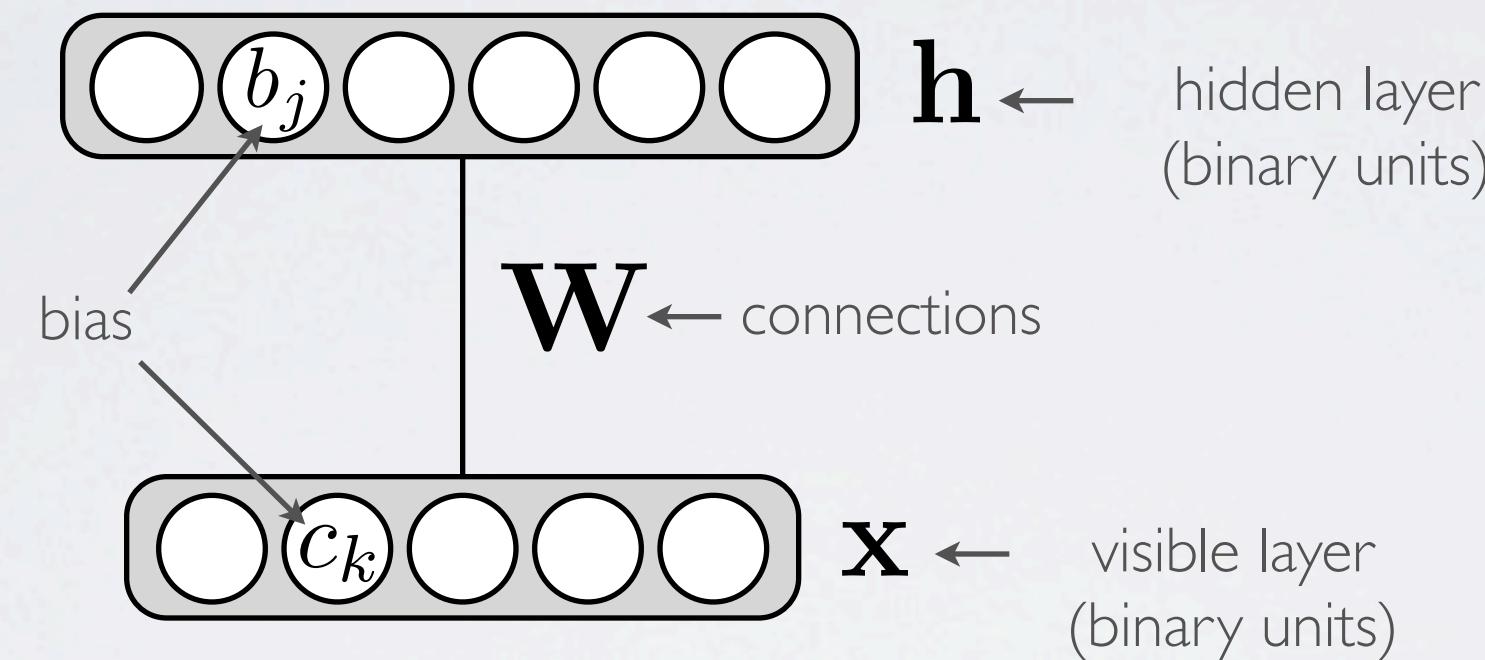


$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \overbrace{\prod_j \prod_k \exp(W_{j,k} h_j x_k)}^{\text{pair-wise factors}} \left\{ \begin{array}{l} \prod_k \exp(c_k x_k) \\ \prod_j \exp(b_j h_j) \end{array} \right\}^{\text{unary factors}}$$

- The scalar visualization is more informative of the structure within the vectors

RESTRICTED BOLTZMANN MACHINE

Topics: RBM, visible layer, hidden layer, energy function



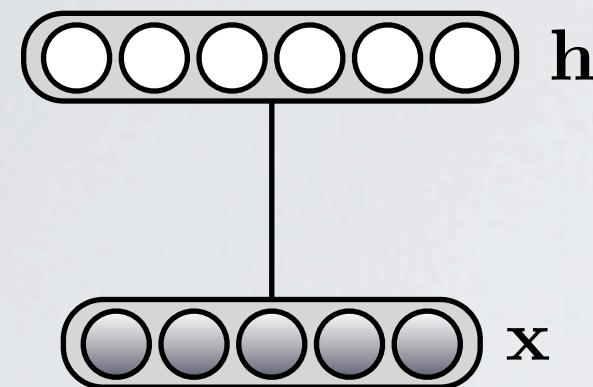
$$\begin{aligned}\text{Energy function: } E(\mathbf{x}, \mathbf{h}) &= -\mathbf{h}^\top \mathbf{W} \mathbf{x} - \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \mathbf{h} \\ &= -\sum_j \sum_k W_{j,k} h_j x_k - \sum_k c_k x_k - \sum_j b_j h_j\end{aligned}$$

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

partition function
(intractable)

INFERENCE

Topics: conditional distributions

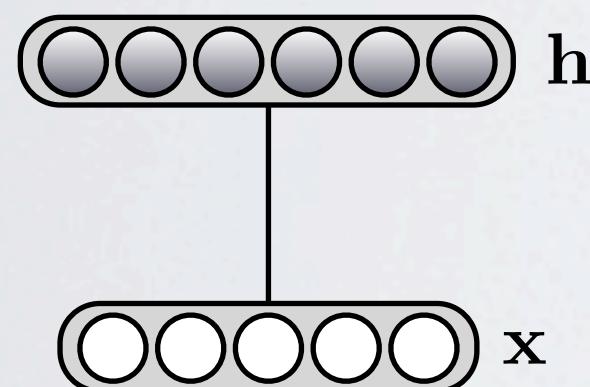


$$p(\mathbf{h}|\mathbf{x}) = \prod_j p(h_j|\mathbf{x})$$

$$p(h_j = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(b_j + \mathbf{W}_{j \cdot} \cdot \mathbf{x}))}$$

$$= \text{sigm}(b_j + \mathbf{W}_{j \cdot} \cdot \mathbf{x})$$

$\nearrow j^{\text{th}} \text{ row of } \mathbf{W}$



$$p(\mathbf{x}|\mathbf{h}) = \prod_k p(x_k|\mathbf{h})$$

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

$$= \text{sigm}(c_k + \mathbf{h}^\top \mathbf{W}_{\cdot k})$$

$\nearrow k^{\text{th}} \text{ column of } \mathbf{W}$

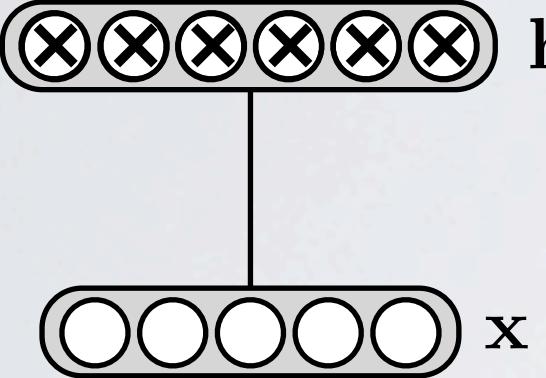
$$p(\mathbf{h}|\mathbf{x})$$

$$p(h_j = 1 | \mathbf{x})$$

FREE ENERGY

Topics: free energy

- What about $p(\mathbf{x})$?



$$\begin{aligned}
 p(\mathbf{x}) &= \sum_{\mathbf{h} \in \{0,1\}^H} p(\mathbf{x}, \mathbf{h}) = \sum_{\mathbf{h} \in \{0,1\}^H} \exp(-E(\mathbf{x}, \mathbf{h}))/Z \\
 &= \exp \left(\mathbf{c}^\top \mathbf{x} + \sum_{j=1}^H \log(1 + \exp(b_j + \mathbf{W}_{j \cdot} \mathbf{x})) \right) / Z \\
 &= \exp(-F(\mathbf{x}))/Z
 \end{aligned}$$

free energy

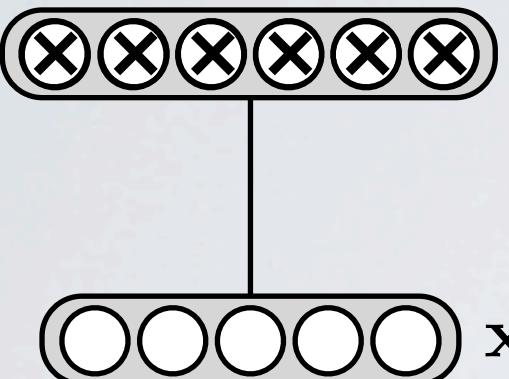
$$p(\mathbf{x})$$

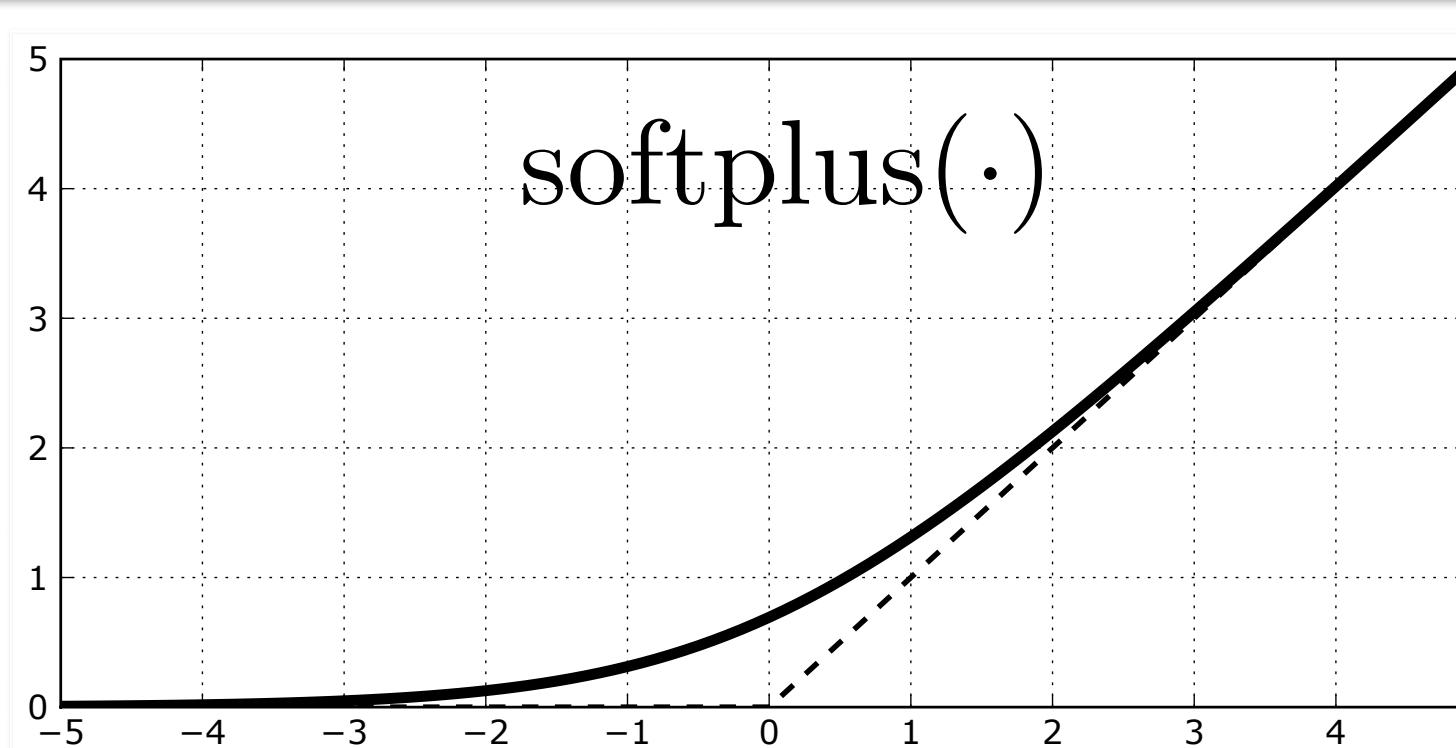
RESTRICTED BOLTZMANN MACHINE

Topics: free energy

$$\begin{aligned}
 \text{h} \quad p(\mathbf{x}) &= \exp \left(\mathbf{c}^\top \mathbf{x} + \sum_{j=1}^H \log(1 + \exp(b_j + \mathbf{W}_{j \cdot} \mathbf{x})) \right) / Z \\
 &= \exp \left(\mathbf{c}^\top \mathbf{x} + \sum_{j=1}^H \text{softplus}(b_j + \mathbf{W}_{j \cdot} \mathbf{x}) \right) / Z
 \end{aligned}$$

“feature” expected in \mathbf{x}
 bias of each feature
 bias the prob of each x_i





MAXIMUM LIKELIHOOD TRAINING

Topics: training objective

- To train an RBM, we'd like to minimize the average negative log-likelihood (NLL)

$$\frac{1}{T} \sum_t l(f(\mathbf{x}^{(t)})) = \frac{1}{T} \sum_t -\log p(\mathbf{x}^{(t)})$$

- We'd like to proceed by stochastic gradient descent

$$\frac{\partial -\log p(\mathbf{x}^{(t)})}{\partial \theta} = \underbrace{\mathbb{E}_{\mathbf{h}} \left[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{h})}{\partial \theta} \mid \mathbf{x}^{(t)} \right]}_{\text{positive phase}} - \boxed{\underbrace{\mathbb{E}_{\mathbf{x}, \mathbf{h}} \left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta} \right]}_{\text{negative phase}}}$$

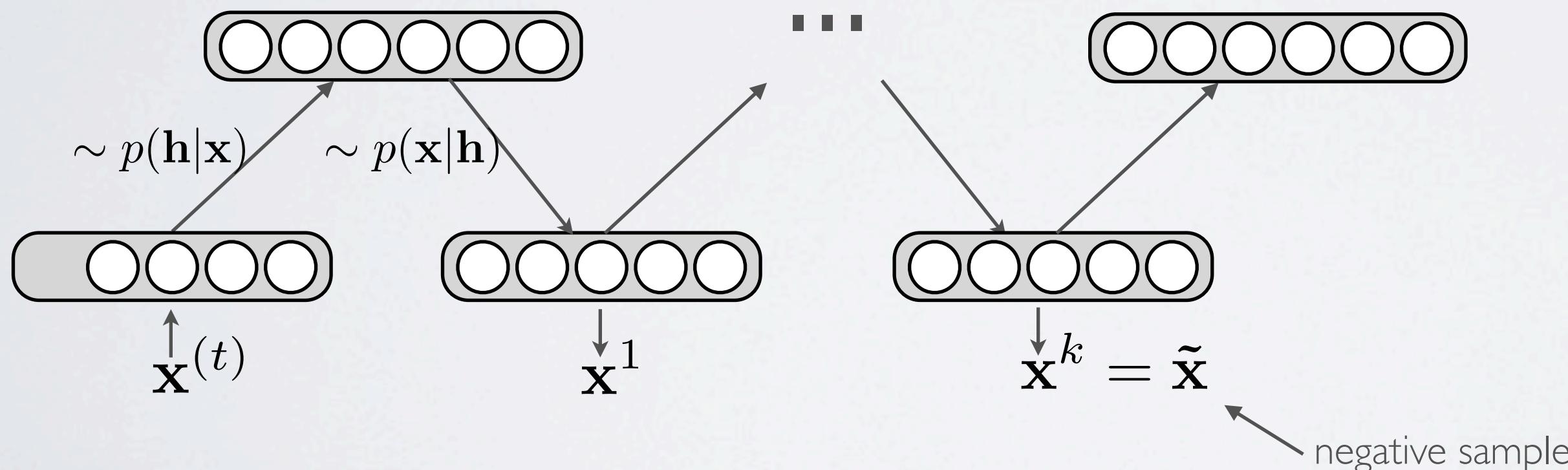
hard to compute

CONTRASTIVE DIVERGENCE (CD)

(HINTON, NEURAL COMPUTATION, 2002)

Topics: contrastive divergence, negative sample

- Idea:
 1. replace the expectation by a point estimate at $\tilde{\mathbf{x}}$
 2. obtain the point $\tilde{\mathbf{x}}$ by Gibbs sampling
 3. start sampling chain at $\mathbf{x}^{(t)}$



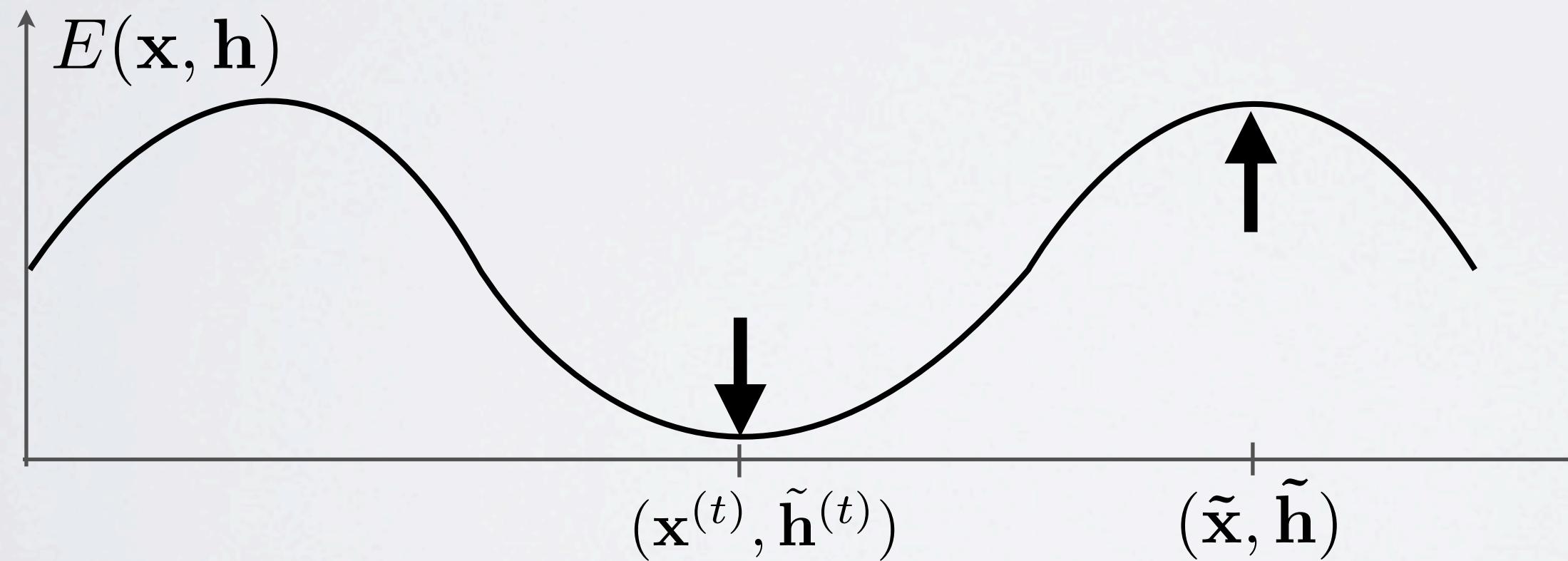
CONTRASTIVE DIVERGENCE (CD)

(HINTON, NEURAL COMPUTATION, 2002)

Topics: contrastive divergence, negative sample

$$\mathbb{E}_{\mathbf{h}} \left[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{h})}{\partial \theta} \mid \mathbf{x}^{(t)} \right] \approx \frac{\partial E(\mathbf{x}^{(t)}, \tilde{\mathbf{h}}^{(t)})}{\partial \theta}$$

$$\mathbb{E}_{\mathbf{x}, \mathbf{h}} \left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta} \right] \approx \frac{\partial E(\tilde{\mathbf{x}}, \tilde{\mathbf{h}})}{\partial \theta}$$



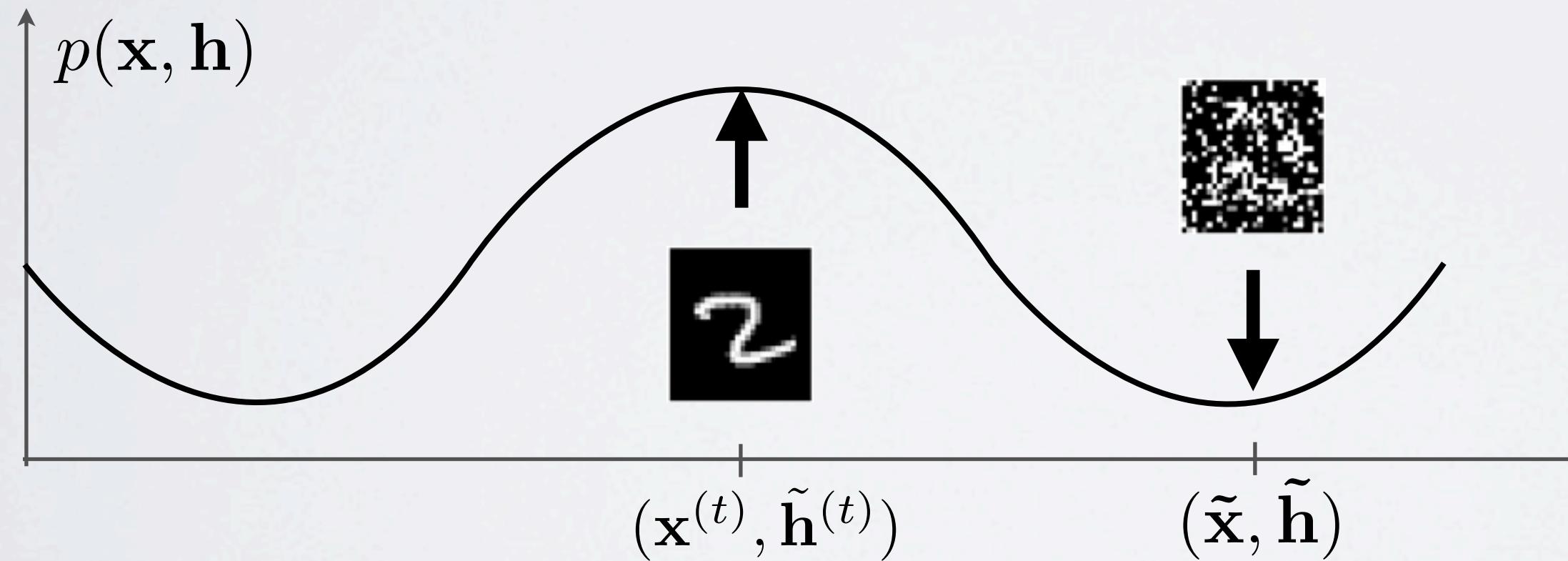
CONTRASTIVE DIVERGENCE (CD)

(HINTON, NEURAL COMPUTATION, 2002)

Topics: contrastive divergence, negative sample

$$\mathbb{E}_{\mathbf{h}} \left[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{h})}{\partial \theta} \mid \mathbf{x}^{(t)} \right] \approx \frac{\partial E(\mathbf{x}^{(t)}, \tilde{\mathbf{h}}^{(t)})}{\partial \theta}$$

$$\mathbb{E}_{\mathbf{x}, \mathbf{h}} \left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta} \right] \approx \frac{\partial E(\tilde{\mathbf{x}}, \tilde{\mathbf{h}})}{\partial \theta}$$



TRAINING

Topics: training objective

- To train an RBM, we'd like to minimize the average negative log-likelihood (NLL)

$$\frac{1}{T} \sum_t l(f(\mathbf{x}^{(t)})) = \frac{1}{T} \sum_t -\log p(\mathbf{x}^{(t)})$$

- We'd like to proceed by stochastic gradient descent

$$\frac{\partial -\log p(\mathbf{x}^{(t)})}{\partial \theta} = \underbrace{\mathbb{E}_{\mathbf{h}} \left[\frac{\partial E(\mathbf{x}^{(t)}, \mathbf{h})}{\partial \theta} \mid \mathbf{x}^{(t)} \right]}_{\text{positive phase}} - \boxed{\underbrace{\mathbb{E}_{\mathbf{x}, \mathbf{h}} \left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta} \right]}_{\text{negative phase}}}$$

hard to compute

DERIVATION OF THE LEARNING RULE

Topics: contrastive divergence

- Derivation of $\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta}$ for $\theta = W_{jk}$

$$\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial W_{jk}} = \frac{\partial}{\partial W_{jk}} \left(-\sum_{jk} W_{jk} h_j x_k - \sum_k c_k x_k - \sum_j b_j h_j \right)$$

$$= -\frac{\partial}{\partial W_{jk}} \sum_{jk} W_{jk} h_j x_k$$

$$= -h_j x_k$$

$$\nabla_{\mathbf{W}} E(\mathbf{x}, \mathbf{h}) = -\mathbf{h} \mathbf{x}^\top$$

DERIVATION OF THE LEARNING RULE

Topics: contrastive divergence

- Derivation of $\mathbb{E}_{\mathbf{h}} \left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial \theta} \middle| \mathbf{x} \right]$ for $\theta = W_{jk}$

$$\begin{aligned} \mathbb{E}_{\mathbf{h}} \left[\frac{\partial E(\mathbf{x}, \mathbf{h})}{\partial W_{jk}} \middle| \mathbf{x} \right] &= \mathbb{E}_{\mathbf{h}} \left[-h_j x_k \middle| \mathbf{x} \right] = \sum_{h_j \in \{0,1\}} -h_j x_k p(h_j | \mathbf{x}) \\ &= -x_k p(h_j = 1 | \mathbf{x}) \end{aligned}$$

$$\mathbb{E}_{\mathbf{h}} [\nabla_{\mathbf{W}} E(\mathbf{x}, \mathbf{h}) | \mathbf{x}] = -\mathbf{h}(\mathbf{x}) \mathbf{x}^\top$$

$$\begin{aligned} \mathbf{h}(\mathbf{x}) &\stackrel{\text{def}}{=} \begin{pmatrix} p(h_1=1|\mathbf{x}) \\ \vdots \\ p(h_H=1|\mathbf{x}) \end{pmatrix} \\ &= \text{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x}) \end{aligned}$$

DERIVATION OF THE LEARNING RULE

Topics: contrastive divergence

- Given $\mathbf{x}^{(t)}$ and $\tilde{\mathbf{x}}$ the learning rule for $\theta = \mathbf{W}$ becomes

$$\begin{aligned}
 \mathbf{W} &\leftarrow \mathbf{W} - \alpha \left(\nabla_{\mathbf{W}} - \log p(\mathbf{x}^{(t)}) \right) \\
 &\leftarrow \mathbf{W} - \alpha \left(\mathbb{E}_{\mathbf{h}} \left[\nabla_{\mathbf{W}} E(\mathbf{x}^{(t)}, \mathbf{h}) \mid \mathbf{x}^{(t)} \right] - \mathbb{E}_{\mathbf{x}, \mathbf{h}} [\nabla_{\mathbf{W}} E(\mathbf{x}, \mathbf{h})] \right) \\
 &\leftarrow \mathbf{W} - \alpha \left(\mathbb{E}_{\mathbf{h}} \left[\nabla_{\mathbf{W}} E(\mathbf{x}^{(t)}, \mathbf{h}) \mid \mathbf{x}^{(t)} \right] - \mathbb{E}_{\mathbf{h}} [\nabla_{\mathbf{W}} E(\tilde{\mathbf{x}}, \mathbf{h}) \mid \tilde{\mathbf{x}}] \right) \\
 &\leftarrow \mathbf{W} + \alpha \left(\mathbf{h}(\mathbf{x}^{(t)}) \mathbf{x}^{(t)\top} - \mathbf{h}(\tilde{\mathbf{x}}) \tilde{\mathbf{x}}^\top \right)
 \end{aligned}$$

CD-K: PSEUDOCODE

Topics: contrastive divergence

- I. For each training example $\mathbf{x}^{(t)}$
 - i. generate a negative sample $\tilde{\mathbf{x}}$ using k steps of Gibbs sampling, starting at $\mathbf{x}^{(t)}$
 - ii. update parameters

$$\mathbf{W} \leftarrow \mathbf{W} + \alpha \left(\mathbf{h}(\mathbf{x}^{(t)}) \mathbf{x}^{(t)\top} - \mathbf{h}(\tilde{\mathbf{x}}) \tilde{\mathbf{x}}^\top \right)$$

$$\mathbf{b} \leftarrow \mathbf{b} + \alpha \left(\mathbf{h}(\mathbf{x}^{(t)}) - \mathbf{h}(\tilde{\mathbf{x}}) \right)$$

$$\mathbf{c} \leftarrow \mathbf{c} + \alpha \left(\mathbf{x}^{(t)} - \tilde{\mathbf{x}} \right)$$

2. Go back to I until stopping criteria

CONTRASTIVE DIVERGENCE (CD)

(HINTON, NEURAL COMPUTATION, 2002)

Topics: contrastive divergence

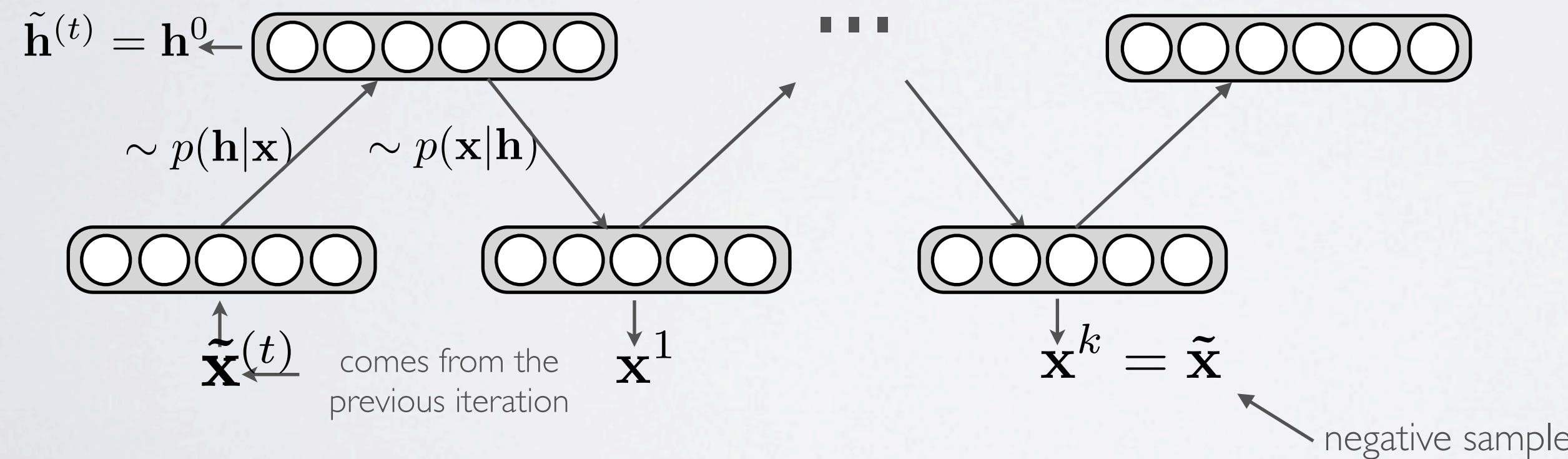
- CD- k : contrastive divergence with k iterations of Gibbs sampling
- In general, the bigger k is, the less **biased** the estimate of the gradient will be
- In practice, $k=1$ works well for pre-training

PERSISTENT CD (PCD)

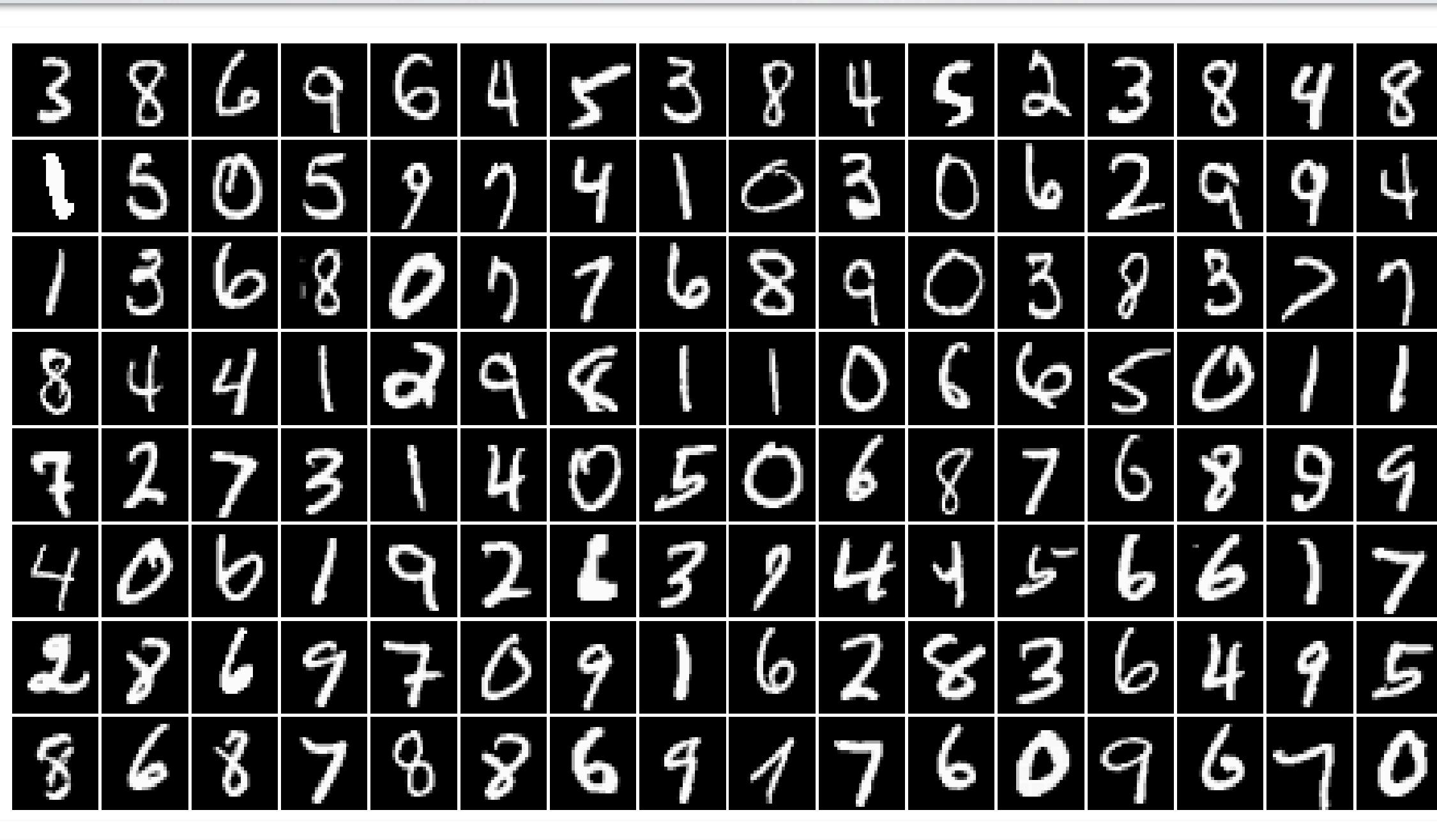
(TIELEMAN, ICML 2008)

Topics: persistent contrastive divergence

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

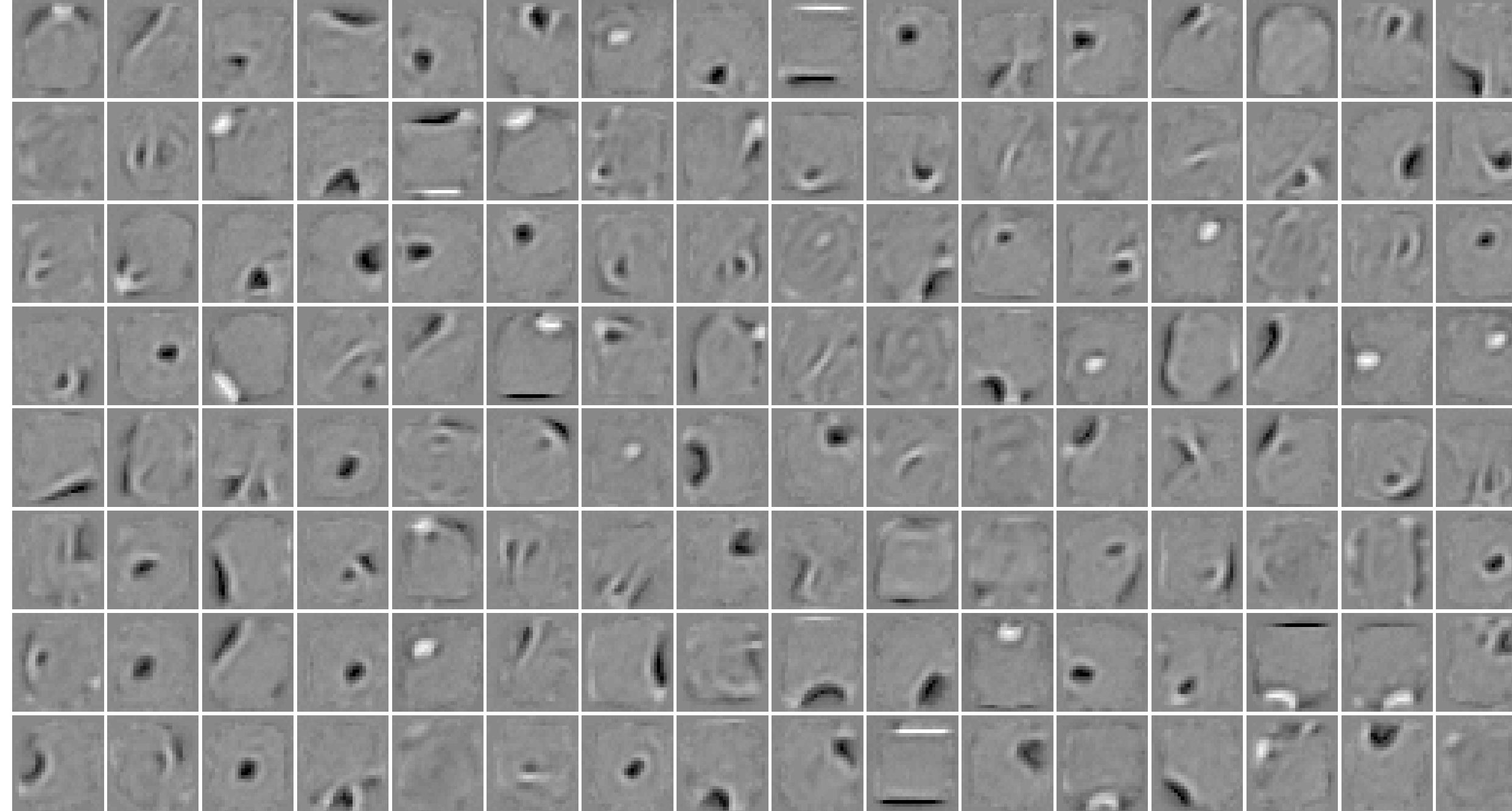


EXAMPLE OF DATA SET: MNIST



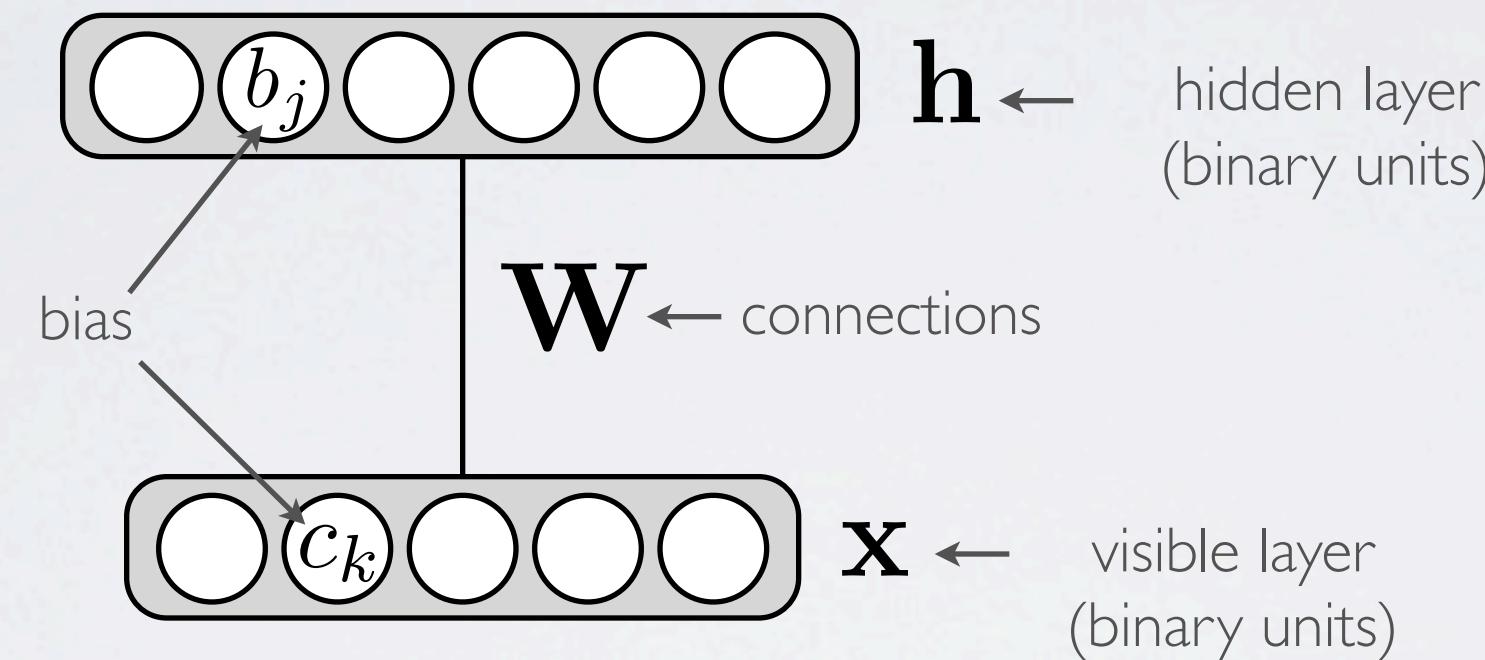
FILTERS

(LAROCHELLE ET AL., JMLR2009)



RESTRICTED BOLTZMANN MACHINE

Topics: RBM, visible layer, hidden layer, energy function



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

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

partition function
(intractable)

GAUSSIAN-BERNOULLI RBM

Topics: Gaussian-Bernoulli RBM

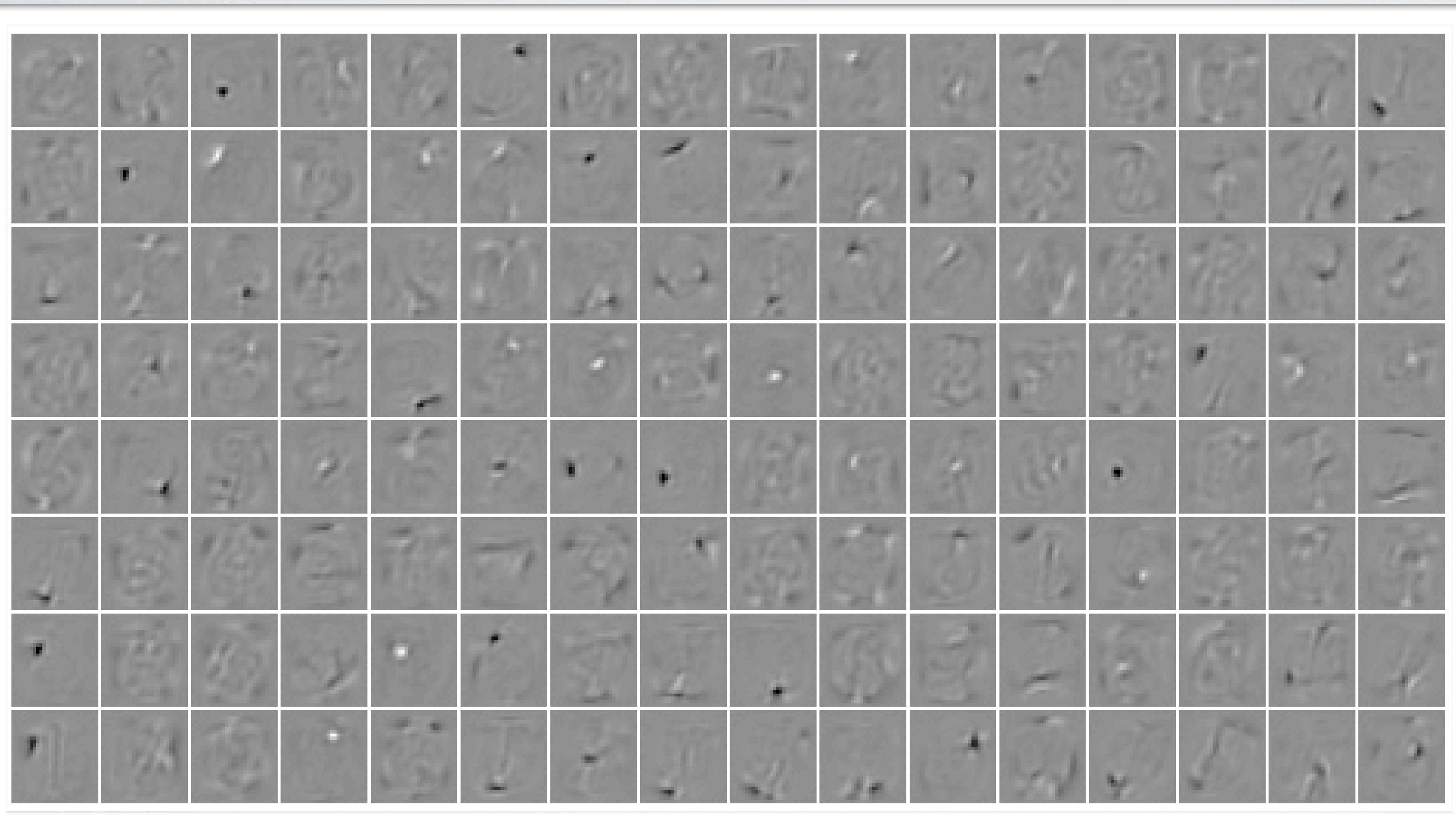
- Inputs \mathbf{x} are unbounded reals
 - ▶ add a quadratic term to the energy function

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{h}^\top \mathbf{W} \mathbf{x} - \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \mathbf{h} + \frac{1}{2} \mathbf{x}^\top \mathbf{x}$$

- ▶ only thing that changes is that $p(\mathbf{x}|\mathbf{h})$ is now a Gaussian distribution with mean $\boldsymbol{\mu} = \mathbf{c} + \mathbf{W}^\top \mathbf{h}$ and identity covariance matrix
- ▶ recommended to normalize the training set by
 - subtracting the mean of each input
 - dividing each input x_k by the training set standard deviation
- ▶ should use a smaller learning rate than in the regular RBM

FILTERS

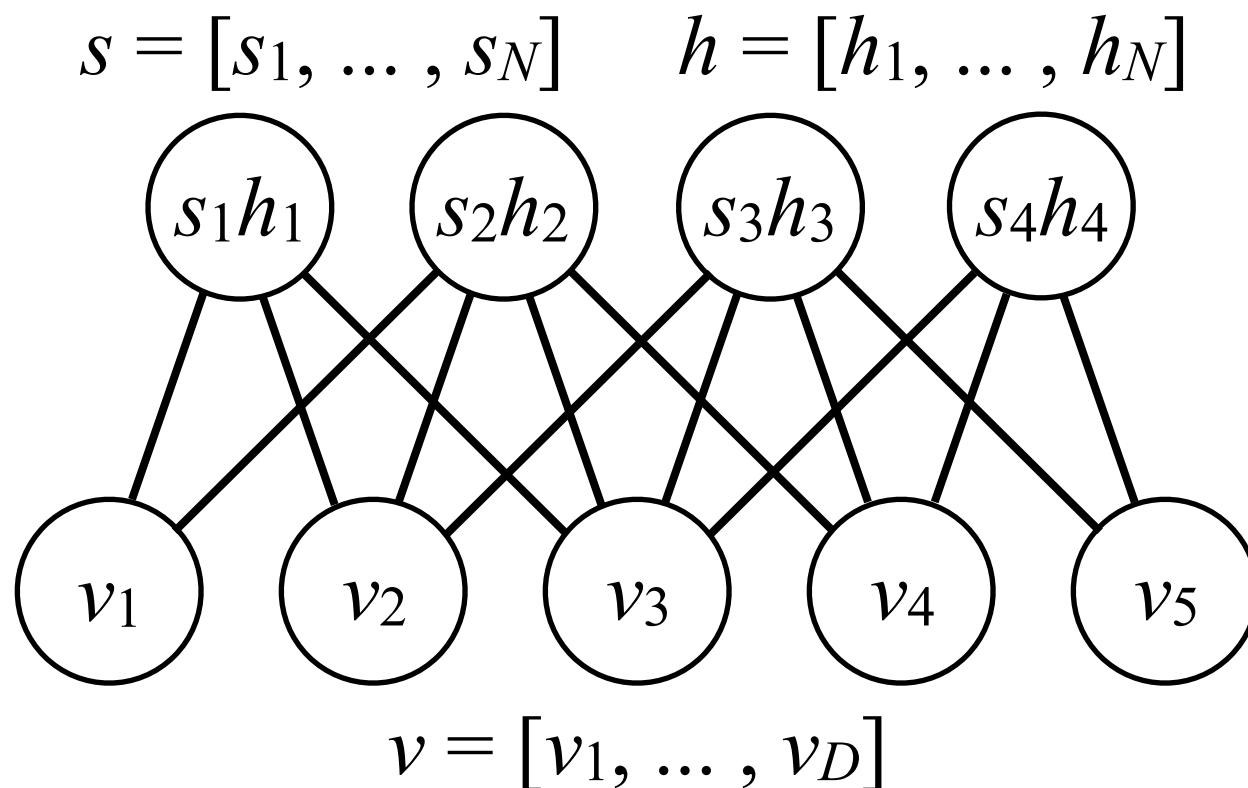
(LAROCHELLE ET AL., JMLR2009)



Spike-and-Slab RBM

Basic Idea \Rightarrow Each hidden unit i possesses:

1. A binary-valued latent **spike** $h_i \in [0,1]$,
2. A real-valued latent **slab** $s_i \in \mathcal{R}$.



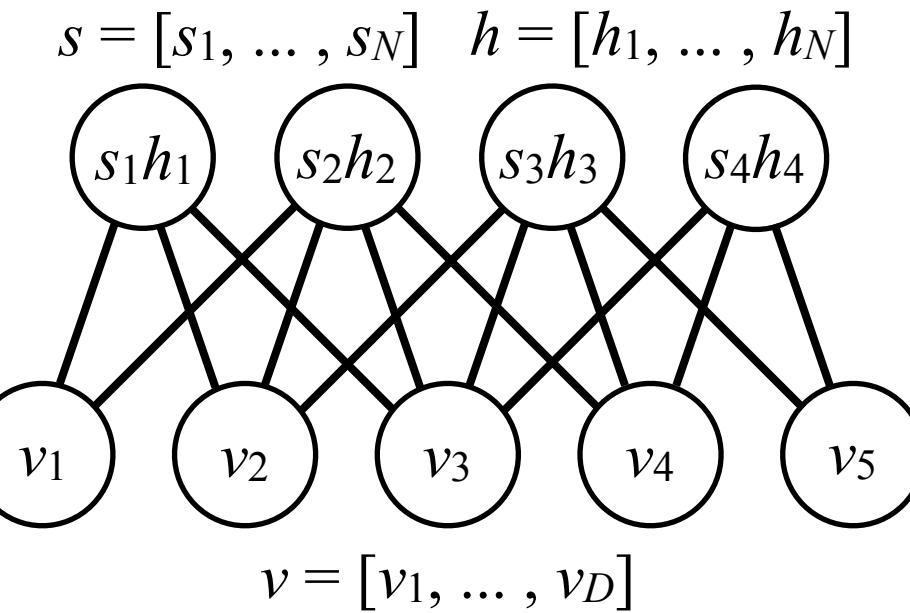
Spike-and-Slab RBM

- ssRBM energy function:

$$E(v, s, h) = - \sum_{i=1}^N v^T W_i s_i h_i + \frac{1}{2} v^T \Lambda v + \frac{1}{2} \sum_{i=1}^N \alpha_i s_i^2 - \sum_{i=1}^N \alpha_i \mu_i s_i h_i + \sum_{i=1}^N \alpha_i \mu_i^2 h_i - \sum_{i=1}^N b_i h_i$$

- ssRBM joint probability density:

$$p(v, s, h) = \frac{1}{Z} \exp \{-E(v, s, h)\}$$



ssRBM Conditional $p(v \mid h)$

Conditional of visible variables v given h :

$$p(v \mid h) = \frac{1}{P(h)} \frac{1}{Z} \int \exp \{-E(v, s, h)\} \ ds = \mathcal{N} \left(C_{v|h} \sum_{i=1}^N W_i \mu_i h_i, C_{v|h} \right)$$

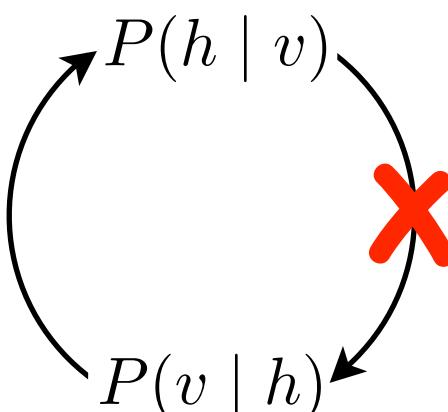
where $C_{v|h} = \left(\Lambda - \sum_{i=1}^N \alpha_i^{-1} h_i W_i W_i^T \right)^{-1}$

↑
Non-diagonal ☹

☺ Models both mean and covariance of the conditional $p(v \mid h)$.

☹ Cannot perform efficient block Gibbs sampling:

$$v \sim p(v \mid h) \neq \prod_j p(v_j \mid h)$$



Conditionals II: $p(v|s,h)$ & $p(s|v,h)$

Conditional dist. of the **visibles** v given s and h :

$$p(v | s, h) = \frac{1}{p(s, h)} \frac{1}{Z} \exp \{-E(v, s, h)\} = \mathcal{N}\left(\left(\Lambda + \sum_{i=1}^N \Phi_i h_i\right)^{-1} \sum_{i=1}^N W_i s_i h_i, \left(\Lambda + \sum_{i=1}^N \Phi_i h_i\right)^{-1}\right)$$

Diagonal Covariance 

- While $p(v | h) \neq \prod_d p(v_d | h)$ given s : $p(v | s, h) = \prod_d p(v_d | s, h)$.

Conditional dist. of the **slabs** s given **visibles** v and **spikes** h :

$$p(s | v, h) = \prod_{i=1}^N p(s_i | v, h) = \prod_{i=1}^N \mathcal{N}\left((\alpha_i^{-1} v^T W_i + \mu_i) h_i, \alpha_i^{-1}\right).$$

Sampling from both $p(v|s,h)$ and $p(s|v,h)$ is simple and efficient.

Conditionals III: $p(h | v)$

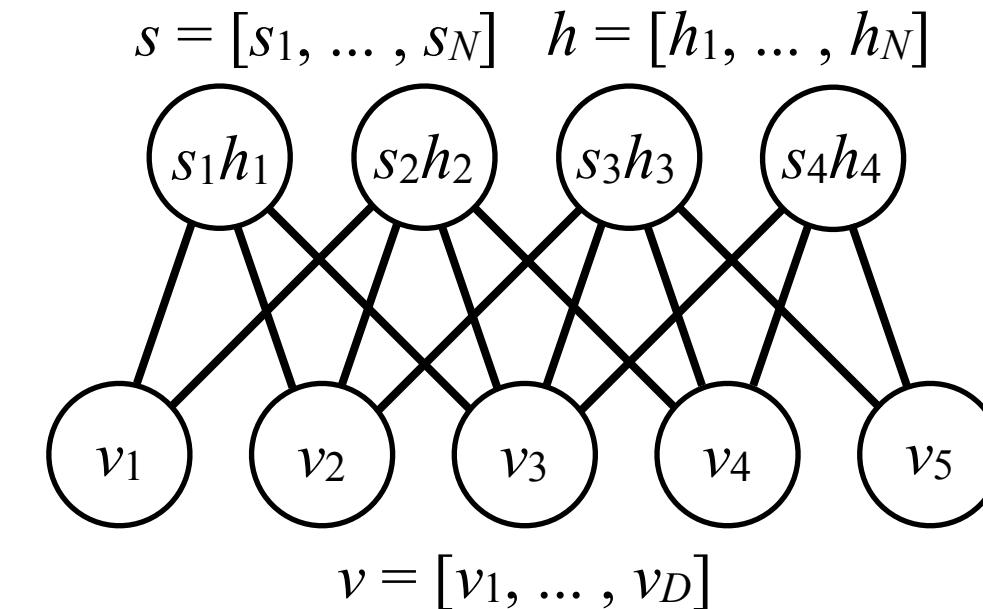
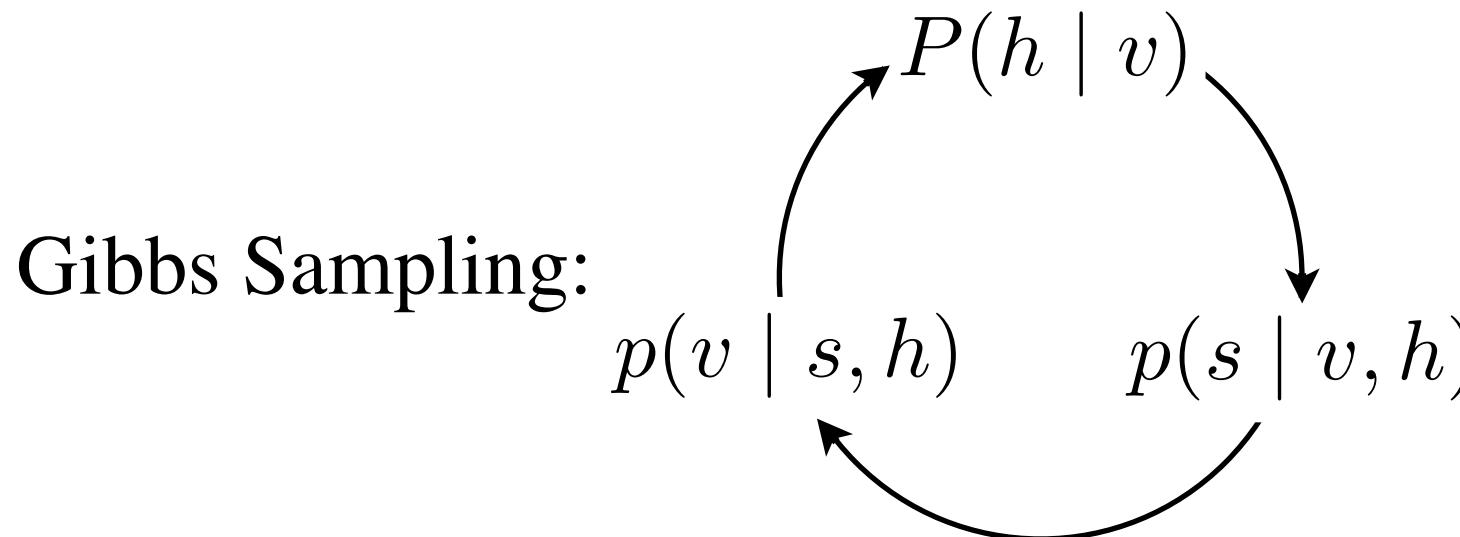
Conditional of the spike variables h given v : $P(h | v) = \prod_i P(h_i | v)$

$$P(h_i = 1 | v) = \text{sigmoid} \left(\frac{1}{2} \alpha_i^{-1} (v^T W_i)^2 - \frac{1}{2} v^T \Phi_i v + v^T W_i \mu_i + b_i \right)$$

quadratic in v **linear in v**

- Activation of **each** spike is controlled by **both** mean and covariance info.
- Compare this to the analogous mcRBM conditionals:
 - Covariance units: $P(h_i^c = 1 | v) = \text{sigmoid} \left(-\frac{1}{2} (v^T W_i^c)^2 - b_i^c \right)$,
 - Mean units: $P(h_j^m = 1 | v) = \text{sigmoid} (v^T W_j^m + b_j^m)$

ssRBM Inference and Learning



- By sampling s , we define a **3-phase block Gibbs sampler**

1.

$$P(h | v) = \prod_{i=1}^N \text{sigmoid} \left(\frac{1}{2} \alpha_i^{-1} (v^T W_i)^2 - \frac{1}{2} v^T \Phi_i v + v^T W_i \mu_i + b_i \right)$$

2.

$$p(s | v, h) = \prod_{i=1}^N \mathcal{N} \left((\alpha_i^{-1} v^T W_i + \mu_i) h_i , \alpha_i^{-1} \right).$$

3.

$$p(v | s, h) = \mathcal{N} \left(\left(\Lambda + \sum_{i=1}^N \Phi_i h_i \right)^{-1} \sum_{i=1}^N W_i s_i h_i , \left(\Lambda + \sum_{i=1}^N \Phi_i h_i \right)^{-1} \right)$$

- Learning via stochastic maximum likelihood.

Sampling from the Convolutional ssRBM

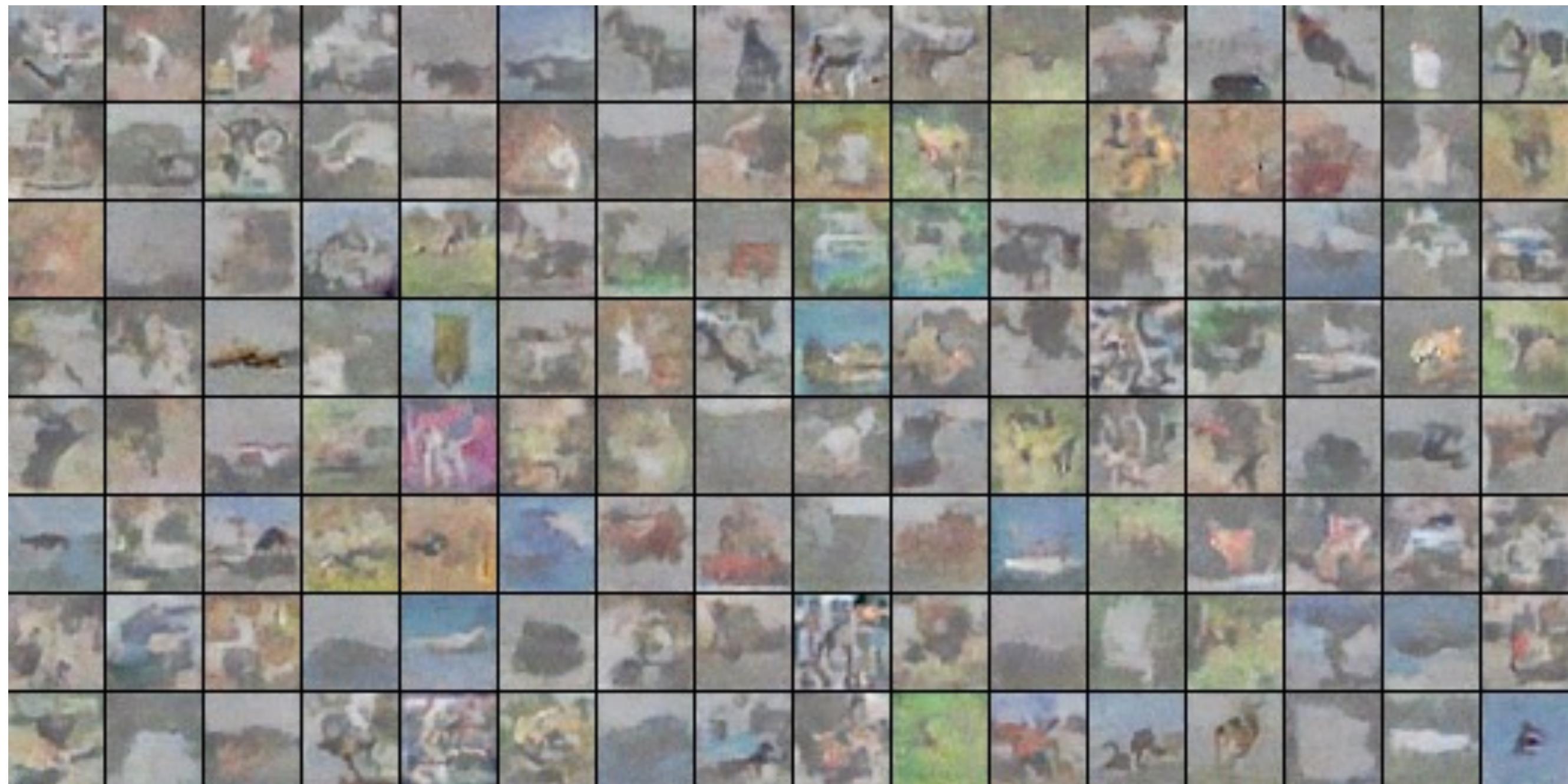
Used the convolutional setup of Krizhevsky (2010)

- Combines both (9x9) convolutional and (32x32) global weight vectors



Sampling from the Convolutional ssRBM

Samples from the Spike-and-slab RBM:



OTHER TYPES OF OBSERVATIONS

Topics: extensions to other observations

- Extensions support other types:
 - ▶ real-valued: Gaussian-Bernoulli RBM
 - ▶ Binomial observations:
 - Rate-coded Restricted Boltzmann Machines for Face Recognition.
Yee Whye Teh and Geoffrey Hinton, 2001
 - ▶ Multinomial observations:
 - Replicated Softmax: an Undirected Topic Model.
Ruslan Salakhutdinov and Geoffrey Hinton, 2009
 - Training Restricted Boltzmann Machines on Word Observations.
George Dahl, Ryan Adam and Hugo Larochelle, 2012
 - ▶ and more (see course website)