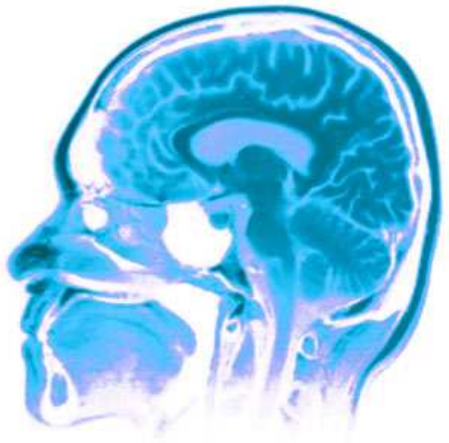
CPSC540



Undirected Graphical Models

Nando de Freitas

2011

KPM Book Sections: 23

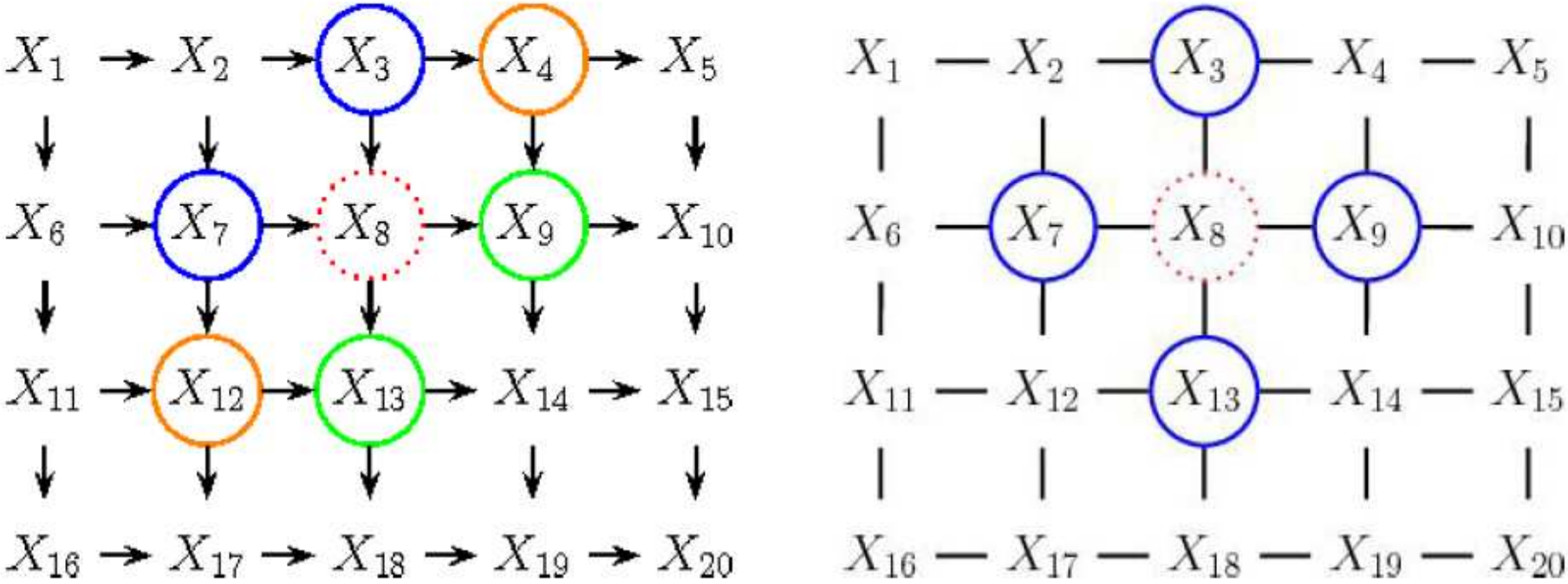
Markov Properties

- UGMs define Conditional Independence (CI) relationships via simple graph separation as follows: for sets of nodes A, B, and C, we say $\mathbf{x}_A \perp_G \mathbf{x}_B | \mathbf{x}_C$ iff C separates A from B in the graph G.
- This means that, when we remove all the nodes in C, if there are no paths connecting any node in A to any node in B, then the CI property holds.
- This is called the **global Markov property** for UGMs.
- The **local Markov property**, states that a node is independent of all the rest given its neighbors or, more formally,

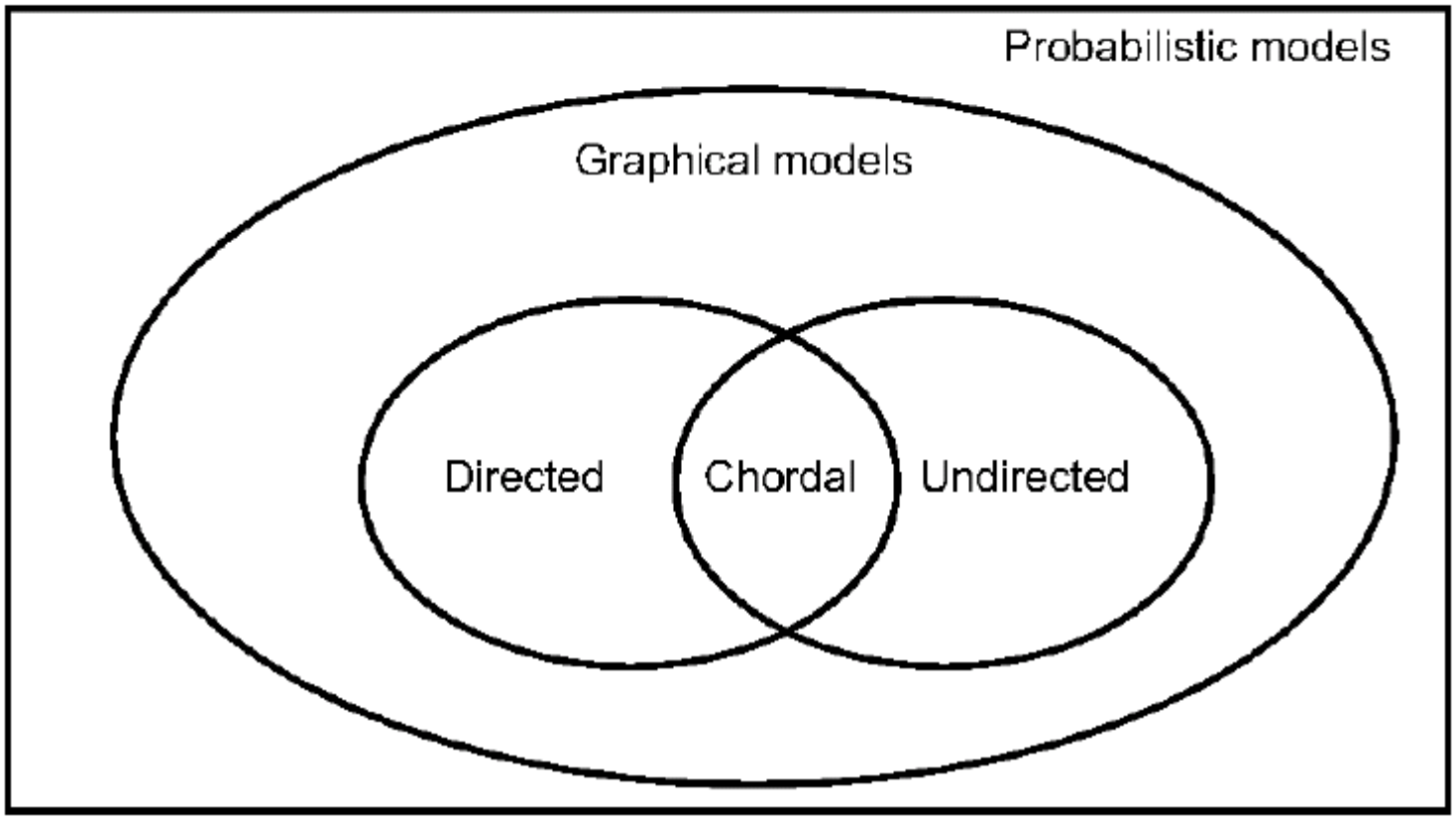
$$t \perp \mathcal{V} \setminus \mathrm{cl}(t) | \mathrm{mb}(t)$$

  where $\mathcal{V} = \{1, \ldots, D\}$ is the set of all the nodes, $\mathrm{mb}(t)$ is the Markov blanket of node t, and $\mathrm{cl}(t) = \mathrm{mb}(t) \cup \{t\}$ is the closure of node t.
- In the case of a UGM, the Markov blanket of a node in a UGM is just the node's neighbors: $\mathrm{mb}(t) = \mathrm{nbr}(t)$.

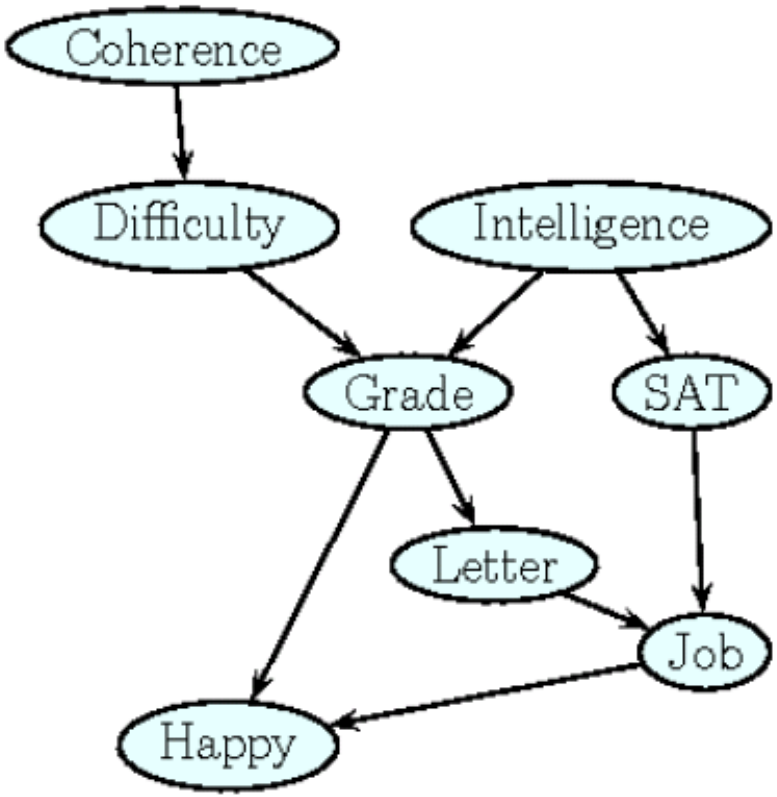
Conditional independence

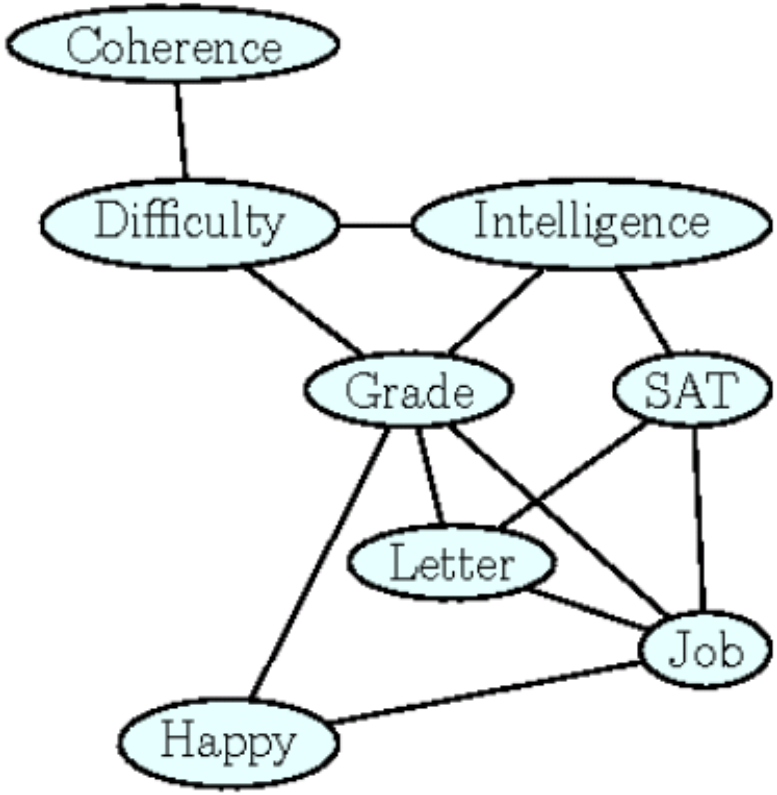


Directed or Undirected



DAG to undirected via moralization





Hammersley-Clifford Theorem

- A **clique** is a set of nodes which are all fully connected to each other. A **maximal clique** is one which cannot be made larger without losing the clique property.

- A **potential function** or **factor** associated with the variables in clique c, $\psi_c(\mathbf{x}_c|\boldsymbol{\theta}_c)$, is an arbitrary non-negative function of its arguments, i.e., $\psi_c(\mathbf{x}_c|\boldsymbol{\theta}_c) \geq 0$ for all $\mathbf{x}_c$.

  Theorem 0.1 (**Hammersley-Clifford**). *A positive distribution $p(\mathbf{x}) > 0$ satisfies the Markov properties of an undirected graph G iff p can be represented as a product of factors, one per maximal clique, i.e.,*

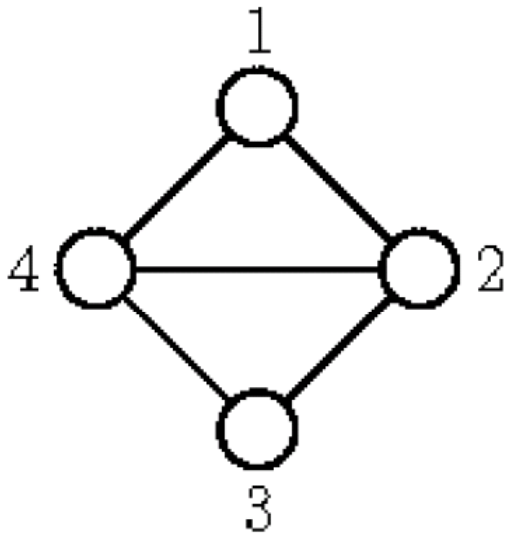
$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c|\boldsymbol{\theta}_c)$$

  where $\mathcal{C}$ is the set of all the (maximal) cliques of G, and $Z(\boldsymbol{\theta})$ is the **partition function** *given by*

$$Z(\boldsymbol{\theta}) := \sum_{\mathbf{x}} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c|\boldsymbol{\theta}_c)$$

Example

- $$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\psi_{124}(\mathbf{x}_{124})\psi_{234}(\mathbf{x}_{234})$$



- If the variables are discrete, we can represent the potential functions as tables of (non-negative) numbers, just as we did with CPTs. However, the potentials are not probabilities. Rather, they represent the relative "compatibility" between the different assignments to the potential.

Energy-based models

- A common way to ensure potentials are positive is to define

$$\psi_c(\mathbf{x}_c|\boldsymbol{\theta}_c) = \exp(-E(\mathbf{x}_c|\boldsymbol{\theta}_c))$$

  where $E(\mathbf{x}_c) \in \mathbb{R}$ is called an **energy**.

- The resulting joint distribution can then be written as a **Gibbs distribution**

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(-\sum_c E(\mathbf{x}_c|\boldsymbol{\theta}_c))$$

- We see that high probability states correspond to low energy configurations. Models of this form are known as **energy based models**.

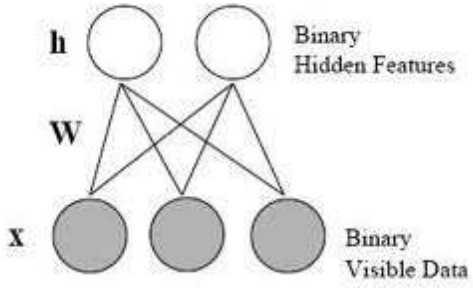
- If the energy is a linear function of the parameters, $E(\mathbf{x}_c|\boldsymbol{\theta}_c) = \boldsymbol{\phi}_c(\mathbf{x}_c)^T \boldsymbol{\theta}_c$, where $\boldsymbol{\phi}_c(\mathbf{x}_c)$ is a feature vector derived from the values of the variables $\mathbf{x}_c$, then the model is known as a **maximum entropy (maxent) model**.

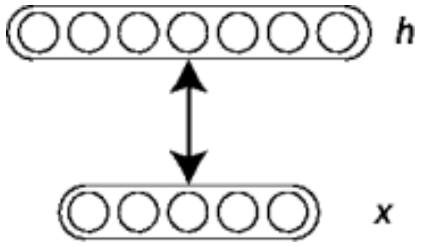
Example: Binary RBMs

- A binary RBM with D visible variables x_s and L hidden variables h_t can be defined as follows:

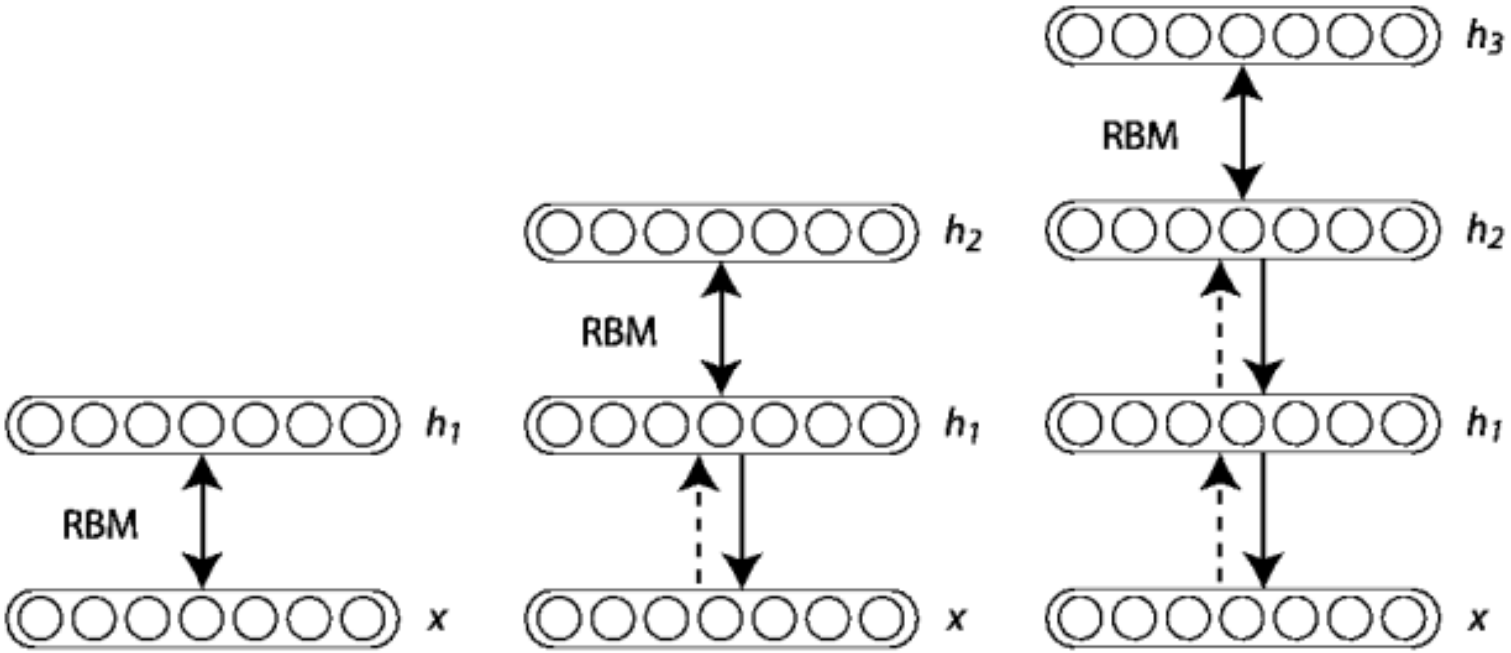
$$
\begin{aligned}
p(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta}) &= \frac{1}{Z(\boldsymbol{\theta})} \exp(-E(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta})) \\
E(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta}) &:= -\sum_{s=1}^{D}\sum_{t=1}^{L} x_s w_{st} h_t - \sum_{s=1}^{D} x_s b_s - \sum_{t=1}^{K} h_t c_t \\
&= -(\mathbf{x}^T \mathbf{W} \mathbf{h} + \mathbf{x}^T \mathbf{b} + \mathbf{h}^T \mathbf{c})
\end{aligned}
$$

where E is the energy function, $\mathbf{W}$ is a $D \times L$ weight matrix, $\mathbf{b}$ are the visible bias terms, $\mathbf{c}$ are the hidden bias terms, and $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{b}, \mathbf{c})$ are all the parameters. (We can absorb the bias terms into the weight matrix by clamping dummy units $x_0 = 1$ and $h_0 = 1$ and setting $\mathbf{w}_{0,t} = c_t$ and $\mathbf{w}_{s,0} = b_s$.)





Binary RBMs and deep nets



Binary RBMs

- The principal advantage of the RBM over general Boltzmann machines is that one can perform efficient inference of the hidden states.
- In particular, we can compute the posterior over the hidden variables in parallel as follows:

$$\begin{aligned} p(\mathbf{h}|\mathbf{x}, \boldsymbol{\theta}) &= \prod_{t=1}^{L} p(h_t|\mathbf{x}, \boldsymbol{\theta}) \\ p(h_t = 1|\mathbf{x}, \boldsymbol{\theta}) &= \text{sigm}(\mathbf{w}_{:,t}^T \mathbf{x} + c_t) \end{aligned}$$

- By symmetry, one can show that we can generate data given the hidden variables as follows:

$$p(x_s = 1|\mathbf{h}, \boldsymbol{\theta}) = \text{sigm}(\mathbf{w}_{s,:}^T \mathbf{h} + b_s)$$

Pairwise MRFs

- $$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{s \in \mathcal{V}} \psi_s(x_s|\boldsymbol{\theta}_s) \prod_{(s,t) \in \mathcal{E}} \psi_{s,t}(x_s, x_t|\boldsymbol{\theta}_{st})$$

  where $\mathcal{V}$ are the nodes and $\mathcal{E}$ are the edges. This is known as a **pairwise MRF**.

- Frequently we represent the potential functions as 1d vectors and 2d arrays of numbers:

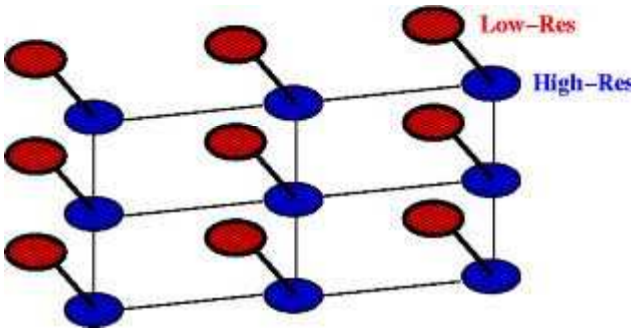
  $$\psi_s(x_s) = e^{\theta_s(x_s)}, \quad \psi_{st}(x_s, x_t) = e^{\theta_{st}(x_s, x_t)}$$

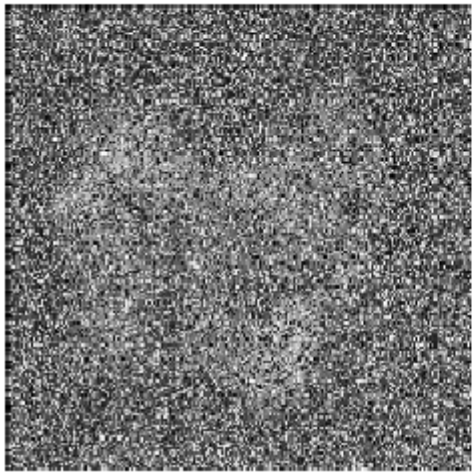
  from which we get

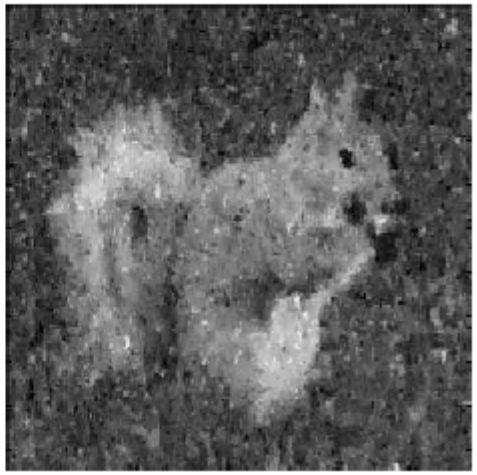
  $$\log p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{s \in \mathcal{V}} \theta_s(x_s) + \sum_{(s,t) \in \mathcal{E}} \theta_{st}(x_s, x_t) - \log Z(\boldsymbol{\theta})$$

  This is called the **standard overcomplete representation**. It is called "overcomplete" because it contains more parameters than are strictly necessary.

MRFs







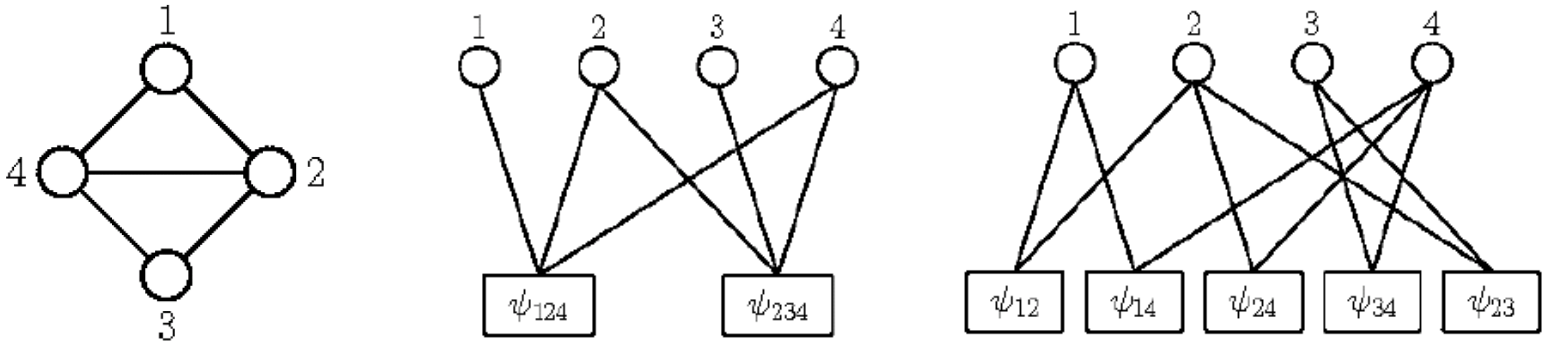
Factor graphs

- When defining UGMs, there can be ambiguity about whether we assume that there is one potential function per clique or if the potential functions are only defined on subsets of nodes in a clique, such as just on the edges.
- In the example below: if we assume one potential per maximal clique we get

$$p(\mathbf{x}_{1:4}) = \frac{1}{Z}\psi_{124}(\mathbf{x}_{124})\psi_{234}(\mathbf{x}_{234})$$

But if we assume one potential per edge, we get

$$p(\mathbf{x}_{1:4}) = \frac{1}{Z}\psi_{12}(\mathbf{x}_{12})\psi_{14}(\mathbf{x}_{14})\psi_{23}(\mathbf{x}_{23})\psi_{24}(\mathbf{x}_{24})\psi_{34}(\mathbf{x}_{34})$$



- # Gaussian graphical models

$$
\begin{aligned}
p(\mathbf{x}|\boldsymbol{\theta}) &\propto \prod_{s\sim t} \psi_{st}(x_s, x_t) \prod_t \phi_t(x_t) \\
\psi_{st}(x_s, x_t) &= \exp(-\frac{1}{2} x_s \Lambda_{st} x_t) \\
\phi_t(x_t) &= \exp(-\frac{1}{2}\Lambda_{tt} x_t^2 + \eta_t x_t)
\end{aligned}
$$

The joint distribution can be written as follows:

$$
p(\mathbf{x}|\boldsymbol{\theta}) \propto \exp[\boldsymbol{\eta}^T \mathbf{x} - \frac{1}{2}\mathbf{x}^T \boldsymbol{\Lambda} \mathbf{x}]
$$

We recognize this as a multivariate Gaussian in information form, where $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$. Hence this is called a **Gaussian MRF**, also known as a **Gaussian graphical model** or **GGM**.

- If $\Lambda_{st} = 0$, then there is no pairwise term connecting s and t, so by the factorization theorem, we conclude that

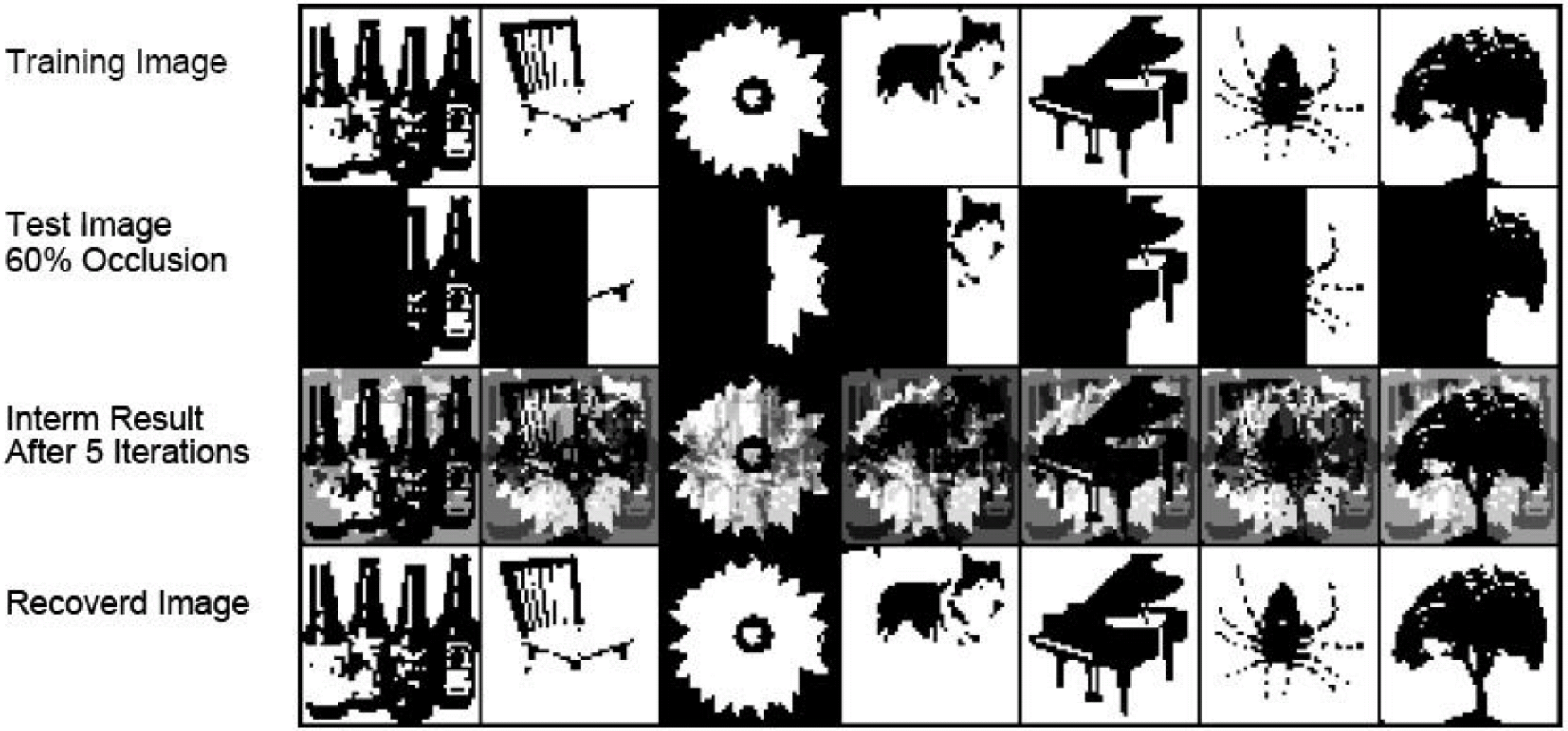
$$
x_s \perp x_t | \mathbf{x}_{-(st)} \iff \Lambda_{st} = 0
$$

Hopfield networks

- A **Hopfield network** is just another name for an **Ising model**.
- What makes these models different from the physics case is that a Hopfield network is usually fully connected, so all the entries of w_{st} are free, modulo the symmetry constraint $\mathbf{W} = \mathbf{W}^T$.
- Also, the weights w_{st} are learned from training data using (approximate) maximum likelihood, rather than being set based on some physical principle.

Hopfield networks

- One application for Hopfield models is **pattern completion**. Below, the network has been trained on 7 binary images. At test time, a partially observed image is presented, and inference imputes the missing components. This can be thought of as retrieving an example from memory based on the example itself; this is known as an **associative memory**.



Conditional random fields (CRFs)

- UGMs/MRFs define unconditional joint probability distributions, $p(\mathbf{x}|\boldsymbol{\theta})$. We can define conditional joint distributions, $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ too:

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{x}, \boldsymbol{\theta})} \prod_c \psi_c(\mathbf{y}_c|\mathbf{x}, \boldsymbol{\theta})$$

  This is called a **conditional random field** or **CRF** (or sometimes a **discriminative random field**).

- CRFs can be used to solve **structured output classification** problems.

Conditional random fields (CRFs)

- With a CRF, we don't "waste resources" modeling things that we always observe (e.g., the image or the words). Instead we can focus our attention on modeling what we care about, namely the distribution of labels given the data.
- Another important advantage of CRFs is that we can make the potentials (or factors) of the model be data-dependendent.
- e.g., in natural language processing (NLP) problems, such as POS tagging, we can make the POS tags depend on global properties of the sentence, such as which language it is written in. It is hard to incorporate global features into generative models, since they will not be independent of the local features.
- The disadvantage of CRFs over MRFs is that they require labeled training data,

Noun phrase chunking

- One common NLP task is **noun phrase chunking**, which refers to the task of segmenting a sentence into its distinct noun phrases (NPs). This is a simple example of a technique known as **shallow parsing**.
- In more detail, we tag each word in the sentence with B (meaning beginning of a new NP), I (meaning inside a NP), or O (meaning outside an NP). This is called **BIO** notation. For example, in the following sentence, the NPs are marked with brackets:

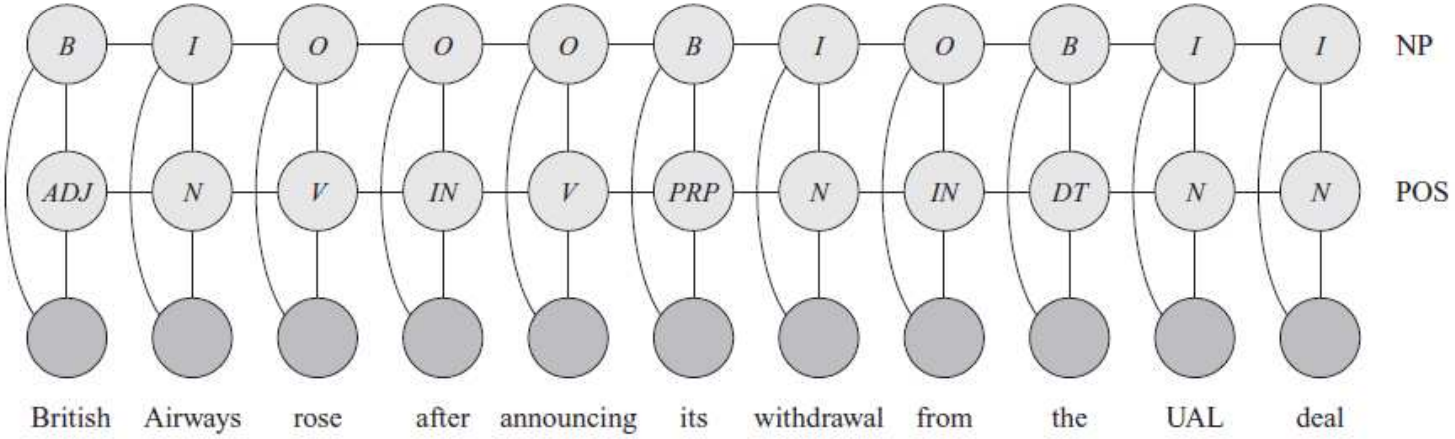
```
   B       I         O     O       O          B    I          O    B   I   I
(British Airways) rose after announcing (its withdrawal) from (the UAI deal)
```

(We need the B symbol so that we can distinguish `I I`, meaning two words within a single NP, from `B B`, meaning two separate NPs.)

Noun phrase chunking

- In the CRF, the connections between adjacent labels encode the probability of transitioning between the B, I and O states, and can enforce constraints such as the fact that B must preceed I.

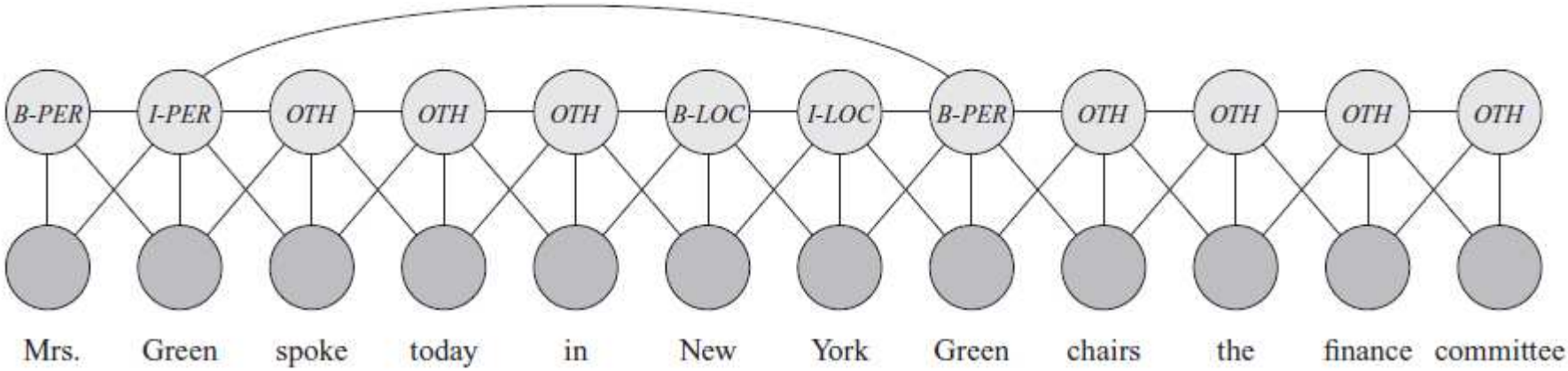
- The features are usually hand engineered and include things like: is the POS tag for this word "noun", does this word begin with a capital letter, is this word followed by a full stop, etc. Typically there are $\sim 1,000 - 10,000$ features per node.



KEY

B	Begin noun phrase	*V*	Verb
I	Within noun phrase	*IN*	Preposition
O	Not a noun phrase	*PRP*	Possesive pronoun
N	Noun	*DT*	Determiner (e.g., a, an, the)
ADJ	Adjective		

CRFs for entity extraction



KEY

B-PER	Begin person name	*I-LOC*	Within location name
I-PER	Within person name	*OTH*	Not an entitiy
B-LOC	Begin location name		

CRFs for computational biology

- An interesting analog to the skip-chain model arises in the problem of predicting the structure of protein side chains. Each residue in the side chain has 4 dihedral angles, which are usually discretized into 3 values called rotamers. The goal is to predict this discrete sequence of angles, $\mathbf{y}$, from the discrete sequence of amino acids, $\mathbf{x}$.

- We can define an energy function $E(\mathbf{x}, \mathbf{y})$, where we include various pairwise interaction terms between nearby residues (elements of the $\mathbf{y}$ vector). This energy is usually defined as a weighted sum of individual energy terms, $E(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = \sum_{j=1}^{D} \theta_k E_j(\mathbf{x}, \mathbf{y})$, where the E_j are energy contribution due to various electrostatic charges, hydrogen bonding potentials, etc, and $\boldsymbol{\theta}$ are the parameters of the model.

- Given the model, we can compute the most probable side chain configuration using $\mathbf{y}^* = \arg\min E(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$.

Photo montage

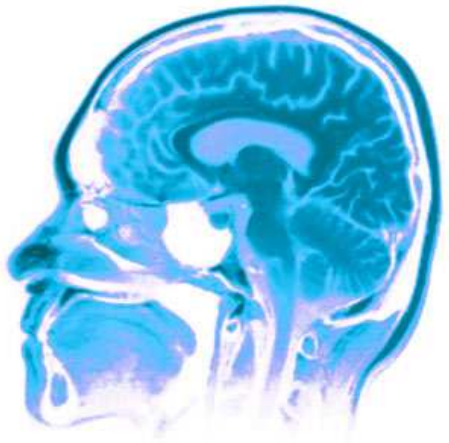
(a)

(b)

(c)

(d)

Next class



Stochastic Algorithms for Inference and Learning

Nando de Freitas

2011

KPM Book Sections: *12*