Probabilistic Graphical Models

Undirected Graphical Models

Probabilistic Graphical Models

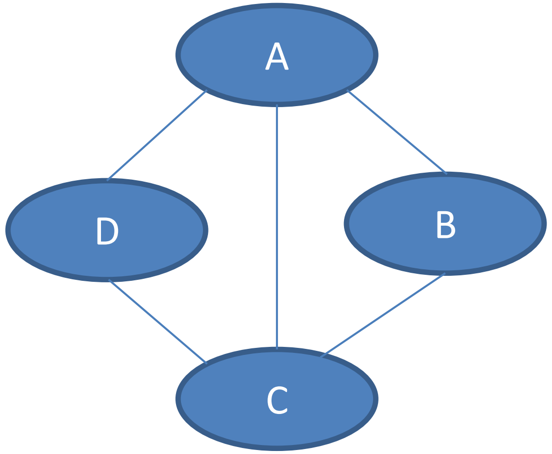
Directed graphical models

- Bayes Nets
- Conditional dependence

Undirected graphical models

- Markov random fields (MRFs)
- Factor graphs

General Markov Networks



$$\Phi = \{\phi_1(\boldsymbol{D}_1), \ldots, \phi_k(\boldsymbol{D}_k)\}$$

$$\tilde{P}_\Phi(X_1, \ldots, X_n) = \prod_i \phi_i(\boldsymbol{D}_\mathrm{i})$$

$$Z_\Phi = \sum_{X_1 \ldots, X_n} \tilde{P}_\Phi(X_1, \ldots, X_n)$$

$$P_\phi(X_1, \ldots, X_n) = \frac{1}{Z_\phi} \prod_i \phi_i(\boldsymbol{D}_\mathrm{i})$$

a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

Log-linear Representation

$$\tilde{P} = \prod_i \phi_i(\boldsymbol{D_i})$$

Original parameterization

$$\tilde{P} = \exp\left(-\sum_j w_j f_j(\boldsymbol{D_j})\right)$$

Log-linear parameterization

Features (f_j) are functions (like factors) without the non-negativity assumption.
Each feature has a single weight. (coefficient, w_j)
Different features can have the same scope.

Log-linear Representation

$$\phi(X_1, X_2) = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}$$

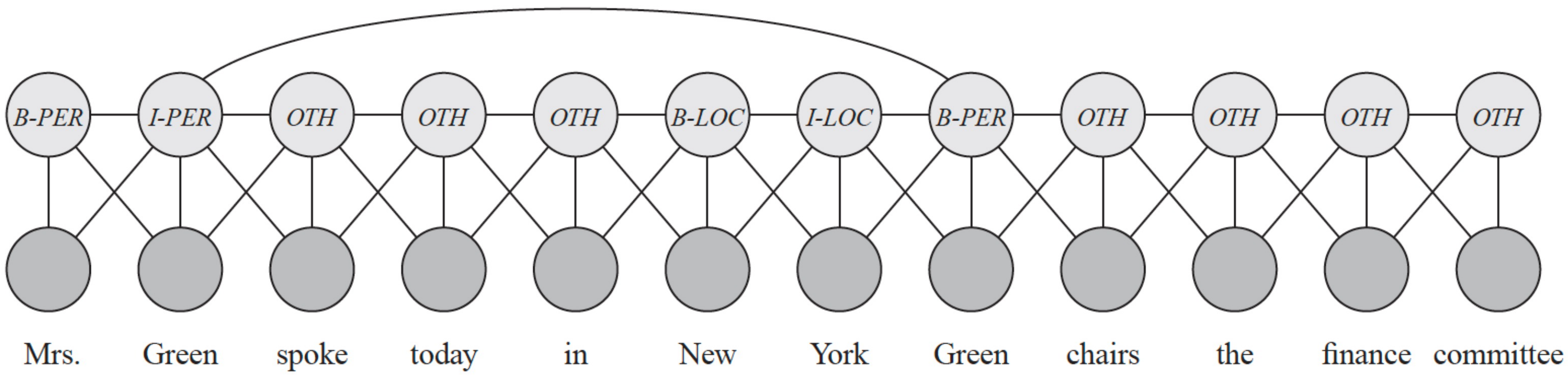
$$f_{12}^{ij} = I(X_1 = i \; and \; X_2 = j)$$

One feature for each i,j value

$$\phi(X_2, X_3) = \exp\left(-\sum_{kl} w_{kl} f_{12}^{kl}(X_1, X_2)\right)$$

$$w_{kl} = -\log(a_{kl})$$

Feature Example: Text



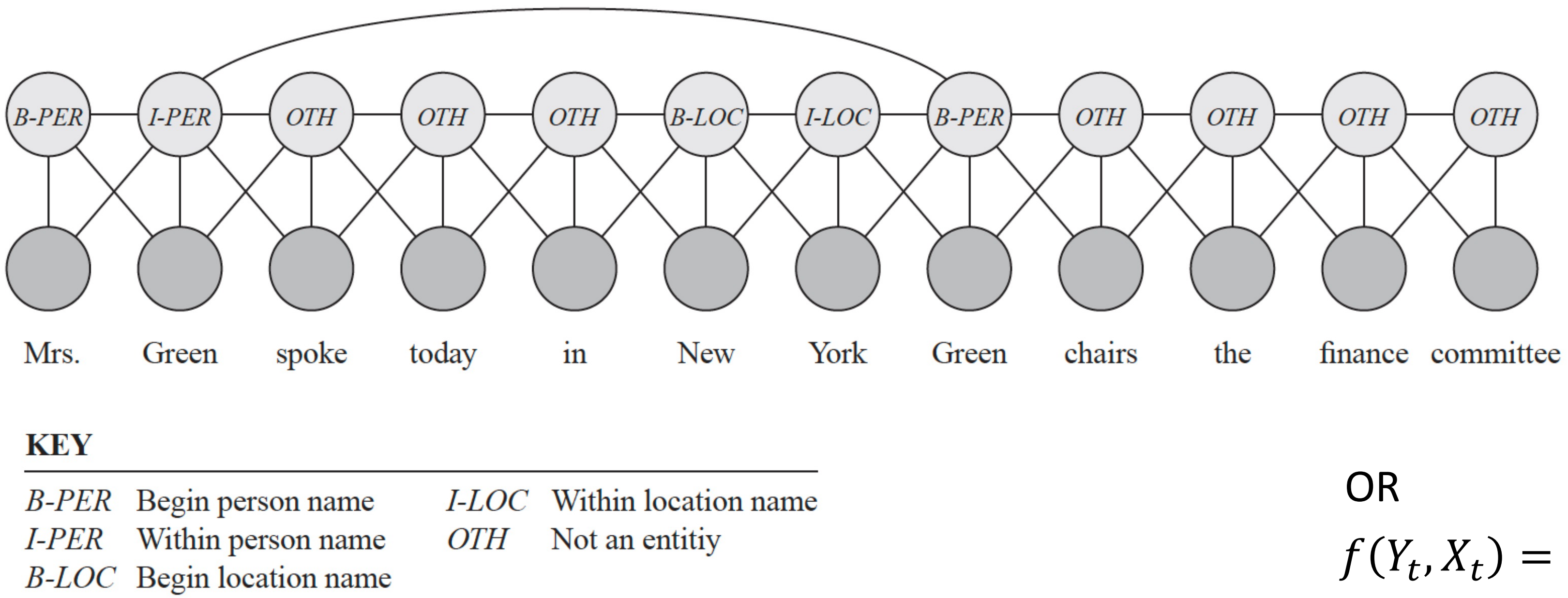
KEY

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

$\phi(Y_t, Y_{t+1})$
$\phi(Y_t | X_{t-1,}, X_t, X_{t+1})$

Problem: Extract entities from a word sequence
For each word: T, a target variable, Y_t, which indicates the entity type of the word.
Possible outcomes of Y_t :B-Person, I-Person, B-Location, I-Location, B-Organization, I-Organization, and Other..

Feature Example: Text



$$\phi(Y_t, Y_{t+1})$$
$$\phi(Y_t | X_{t-1,}, X_t, X_{t+1})$$

OR

$$f(Y_t, X_t) = 1\{Y_i = Person, X_i \text{ } is \text{ } capitalized\}$$

Problem: Extract entities from a word sequence

For each word: T, a target variable, Y_t, which indicates the entity type of the word.

Possible outcomes of Y_t :B-Person, I-Person, B-Location, I-Location, B-Organization, I-Organization, and Other..

Example: Ising Models

$$E(x_1, \ldots, x_n) = -\sum_{i<j} w_{i,j} x_i x_j - \sum_i u_i x_i$$

$$x_i \in \{-1, 1\}$$

$$f_{i,j}(X_i, X_j) = X_i \cdot X_j$$

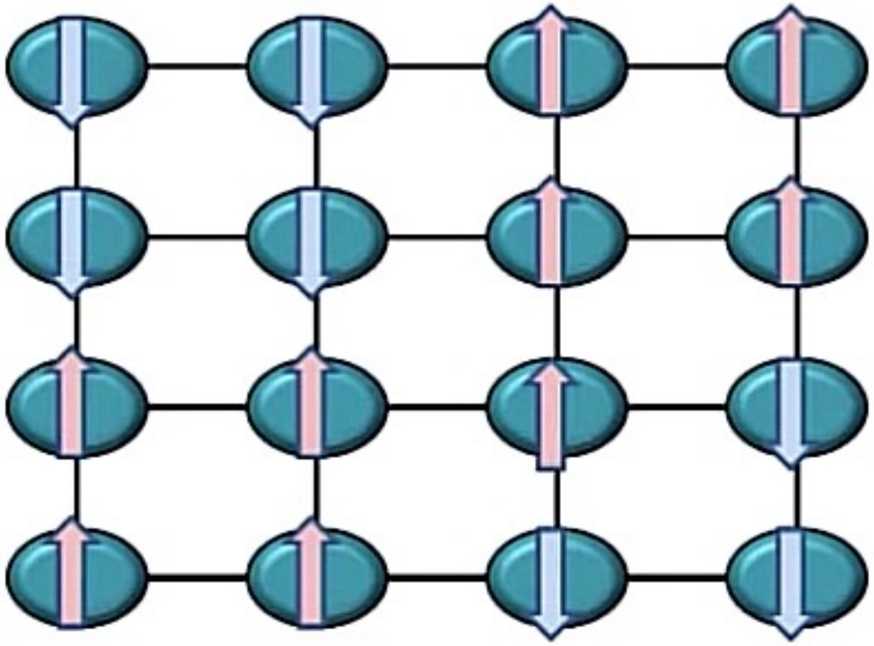
Example: Ising Models

$$E(x_1, \dots, x_n) = -\sum_{i<j} w_{i,j} x_i x_j - \sum_i u_i x_i$$

$$x_i \in \{-1, 1\}$$

$$f_{i,j}(X_i, X_j) = X_i \cdot X_j$$

$$P(\boldsymbol{X}) \propto e^{-\frac{1}{T}E(\boldsymbol{X})}$$

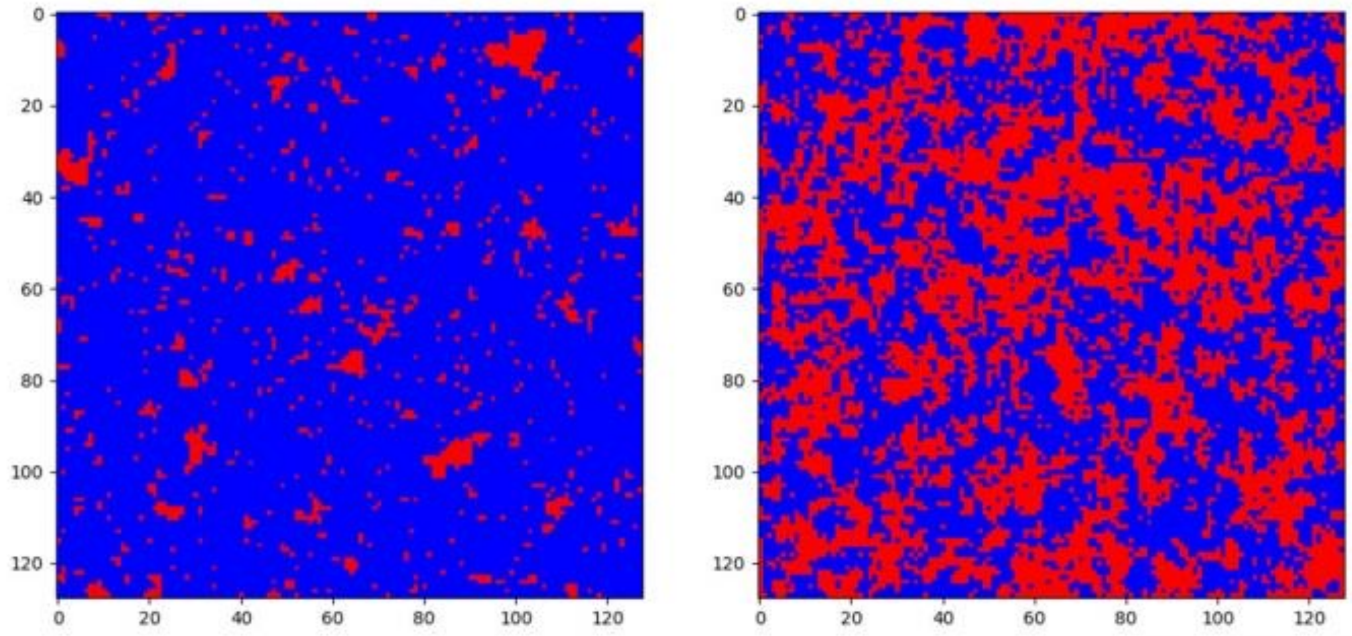


Example: Ising Models

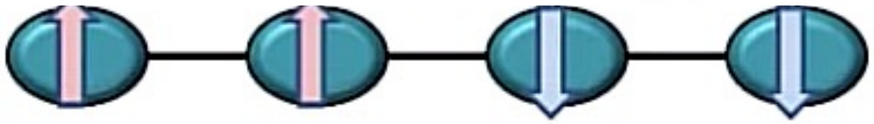
$$E(x_1, \ldots, x_n) = -\sum_{i<}$$

$$x_i \in \{$$

$$f_{i,j}(X_i, X_j$$



$$P(\boldsymbol{X}) \propto e^{-}$$

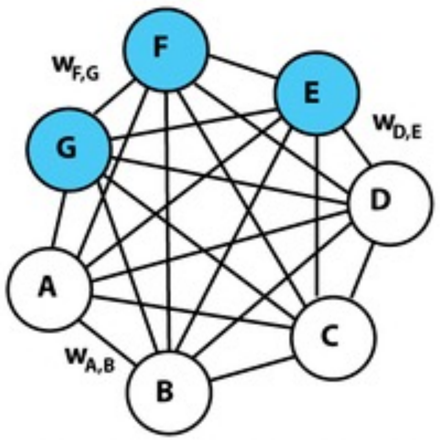


As T grows, w_{ij}'s become smaller

Example: Boltzman machine

$$E = -\sum_{i<j} w_{ij} s_i s_j + -\sum_i \theta_i s_i$$

$$s_i \in \{0, 1\}$$



- w_{ij} is the connection strength between unit j and unit i.
- s_i is the state, $s_i \in \{0,1\}$, of unit i.
- θ_i is the bias of unit i in the global energy function. ($-\theta_i$ is the activation threshold for the unit.)

Model for neural activation

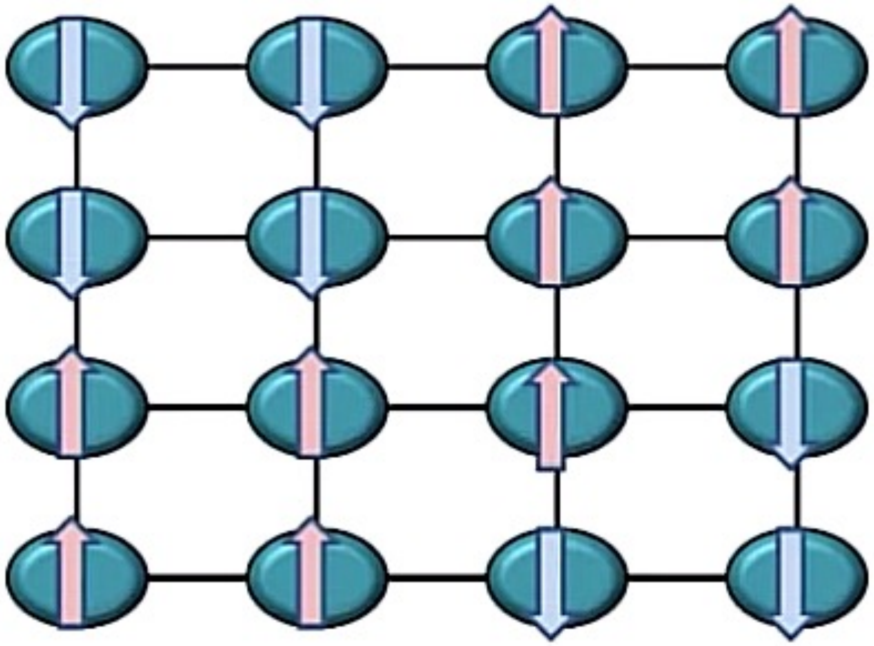
Example: Ising Models

$$E(x_1, \dots, x_n) = -\sum_{i<j} w_{i,j} x_i x_j - \sum_i u_i x_i$$

$$x_i \in \{-1, 1\}$$

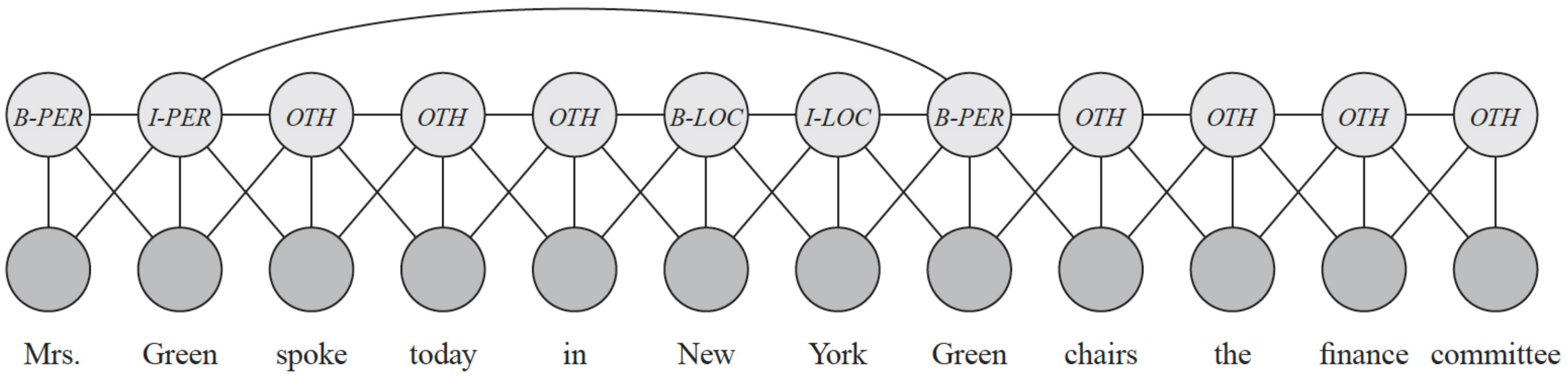
$$f_{i,j}(X_i, X_j) = X_i \cdot X_j$$

$$P(\boldsymbol{X}) \propto e^{-\frac{1}{T}E(\boldsymbol{X})}$$



$w_{i,j}$ will in general be the same for every pair i, j

Feature Example: Text



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		

$\phi(Y_t, Y_{t+1})$

$\phi(Y_t|X_{t-1,}, X_t, X_{t+1})$

OR

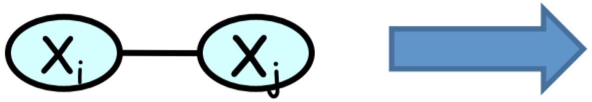
$f(Y_t, X_t) = 1\{Y_i = Person, X_i \text{ is capitalized}\}$

Same energy terms $w_k f_k(X_i, Y_i)$ repeat for all positions i in the sequence

Same energy terms $w_m f_m(Y_i, Y_{i+1})$ also repeat for all positions i

Metric MRFs

- All X_i take values in label space V



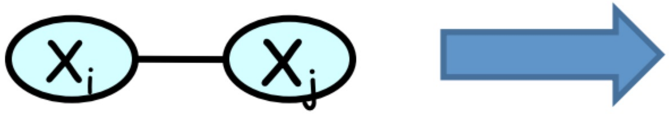
want X_i and X_j to take "similar" values

Distance function μ: V×V → R^+

- $\mu(v, v) = 0$ for all v
- Symmetry: $\mu(v_1, v_2) = \mu(v_2, v_1)$ for all v_1, v_2
- Triangle inequality: $\mu(v_1, v_2) \leq \mu(v_1, v_3) + \mu(v_3, v_2)$

Metric MRFs

- All X_i take values in label space V



want X_i and X_j to take "similar" values

- Distance function $\mu : V \times V \rightarrow R$

$$f_{i,j}(X_i, X_j) = \mu(X_i, X_j)$$

lower distance

higher

$$\exp\left(-w_{ij} f_{ij}(X_i, X_j)\right) \qquad w_{ij} > 0$$

higher probability

w_{ij}

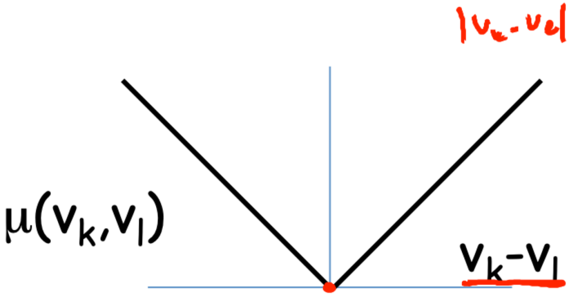
values of X_i and X_j far in μ ⇨ lower probability

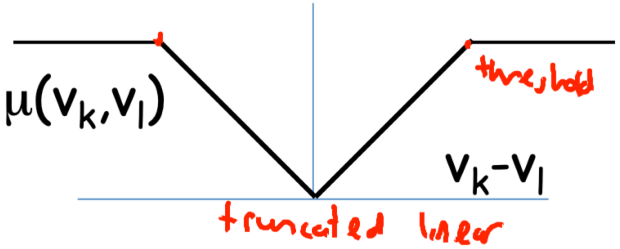
Daphne Koller

Example Metric MRFs

$$\mu(v_k,v_l) = \begin{cases} 0 & v_k=v_l \\ 1 & \text{otherwise} \end{cases}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



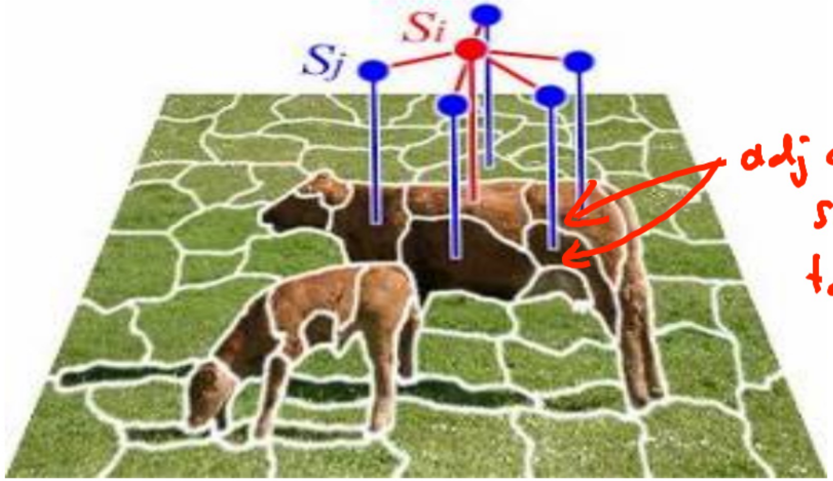


Daphne Koller

Example: Image Segmentation

$$\mu(v_k,v_l) = \begin{cases} 0 & v_k=v_l \\ 1 & \text{otherwise} \end{cases}$$

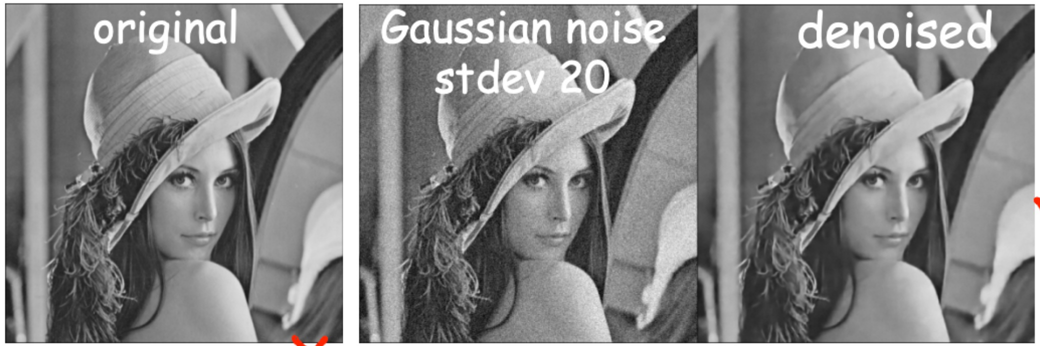
$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



adjacent superpixels take same class

Daphne Koller

Example: Denoising

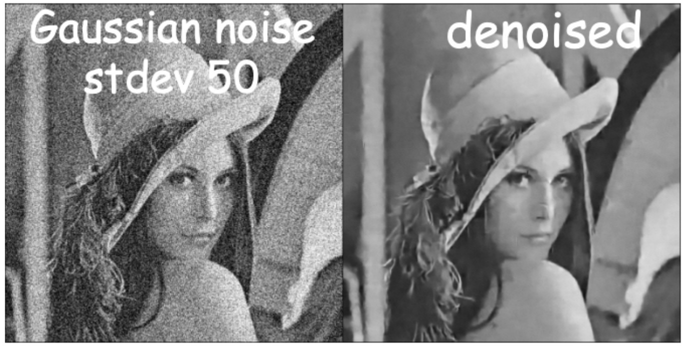


X
noisy pixels

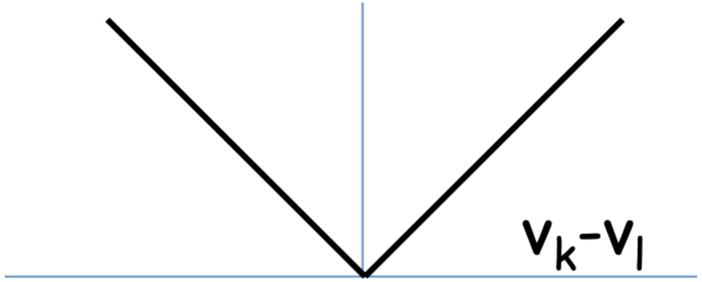
Y clean pixels

Y_i close to X_i

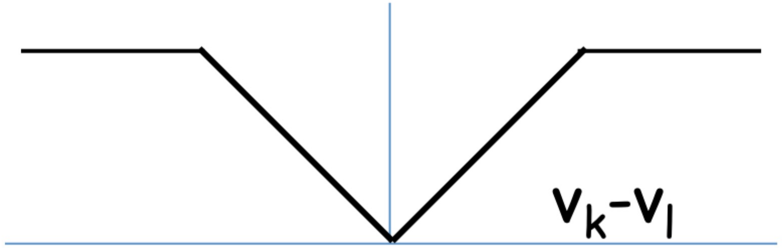
Y_i close to its neighbors



$$\mu(v_k,v_l) = |v_k - v_l|$$



$$\mu(v_k,v_l) = \min(|v_k - v_l|, d)$$



Similar idea for stereo reconstruction

Daphne Koller

Repeated Features

- Need to specify for each feature f_k a set of scopes $Scopes[f_k]$
- For each $\boldsymbol{D}_k \in Scopes\ [f_k]$, we have a term $w_k f_k(\boldsymbol{D}_k)$ in the energy function

- $w_k \sum_{\boldsymbol{D_k}} f(\boldsymbol{D_k})$

- Parameters and structure are reused within an MN and across different MNs

Pt 2: Inference

Queries on PGMs

Conditional Probability Queries

- Evidence: $\mathrm{E} = \boldsymbol{e}$
- Query: a subset of variables $\boldsymbol{Y}$
- Task: compute $P(Y|\boldsymbol{E} = \boldsymbol{e})$

Applications

- Medical/fault diagnosis
- Pedigree analysis

NP-hardness

Exact Inference is NP hard:

Approximate Inference is also NP hard

Queries on PGMs

Conditional Probability Queries

- Evidence: $\mathrm{E} = \boldsymbol{e}$
- Query: a subset of variables $\boldsymbol{Y}$
- Task: compute $P(Y|\boldsymbol{E} = \boldsymbol{e})$

Applications

- Medical/fault diagnosis
- Pedigree analysis

NP-hardness

Exact Inference is NP hard:

Approximate Inference is also NP hard

Why is the expression $\sum_{\bar{W}} P(\bar{Y}, \bar{W}, \bar{e})$ hard to compute in general?

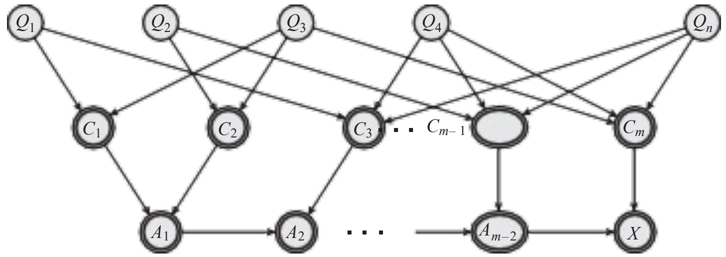
It may be intractable to sum over all the different values that $\bar{W}$ can take.

The summation over all values of $\bar{W}$ is exponential. If $\bar{W}$ has 100 binary variables, then summing will take 2^{100} operations.

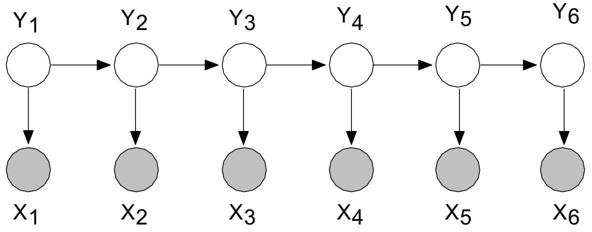
$P(\bar{Y}, \bar{W}, \bar{e})$ is always easy to compute because it is just the product of all CPDs.

Probabilistic inference in practice

- NP-hardness simply says that there **exist** difficult inference problems
- Real-world inference problems are not necessarily as hard as these worst-case instances
- The reduction from SAT created a very complex Bayesian network:



- Some graphs are **easy** to do inference in! For example, inference in hidden Markov models



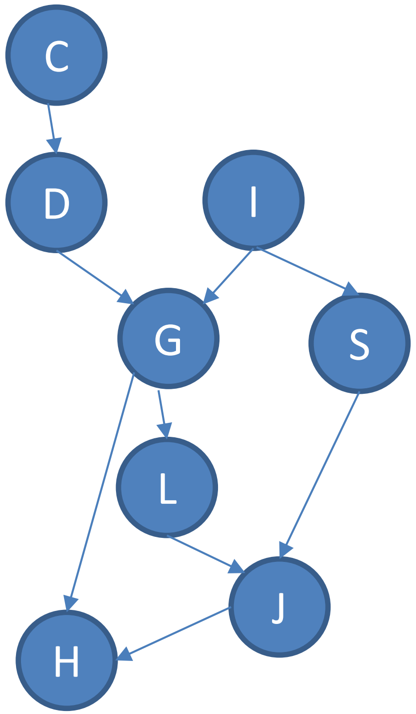
and other tree-structured graphs can be performed in **linear time**

Slides by David Sontag

Sum-product Inference

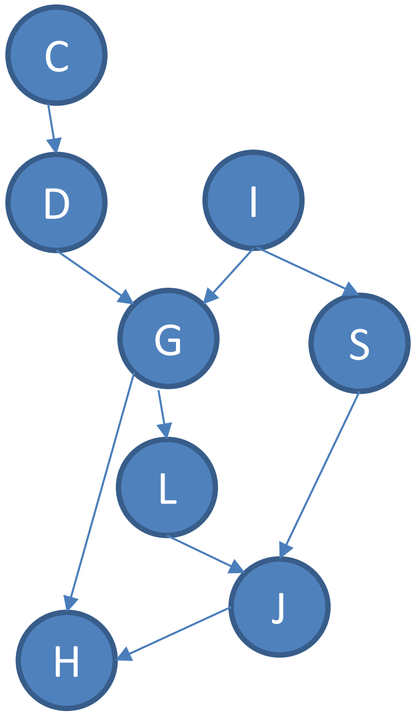
$$\phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)$$
$$\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,G,J)$$

Compute $P(J)$



Sum-product Inference

$$\phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)$$
$$\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,G,J)$$



Compute $P(J)$

$$\sum_{C,D,I,G,S,L,H} \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,G,J)$$

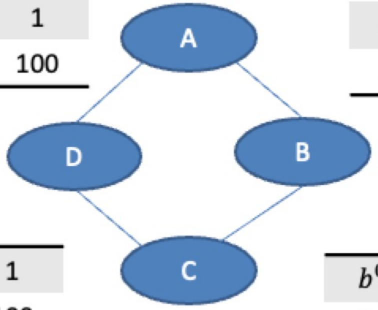
Sum-product Inference for MNs

$$\tilde{P}(D) = \sum_{A,B,C} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$

What about the normalization constant?

d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	100



c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

Introducing Evidence

$$P(\boldsymbol{Y} \mid \boldsymbol{E} = \boldsymbol{e}) = \frac{P(\boldsymbol{Y}, \boldsymbol{E} = \boldsymbol{e})}{P(\boldsymbol{E} = \boldsymbol{e})}$$

$$P(\boldsymbol{Y}, \boldsymbol{E} = \boldsymbol{e}) = \sum_{\boldsymbol{W}} P(\boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{E} = \boldsymbol{e})$$

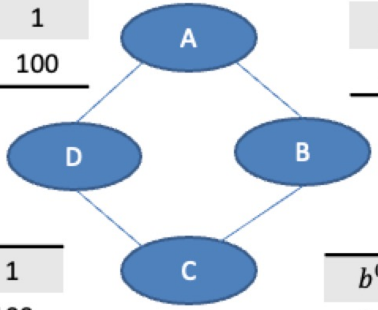
$$\boldsymbol{W} = \boldsymbol{V} \setminus \boldsymbol{E} \cup \boldsymbol{Y}$$

Use the reduced factors:

Example: A=0

d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	100



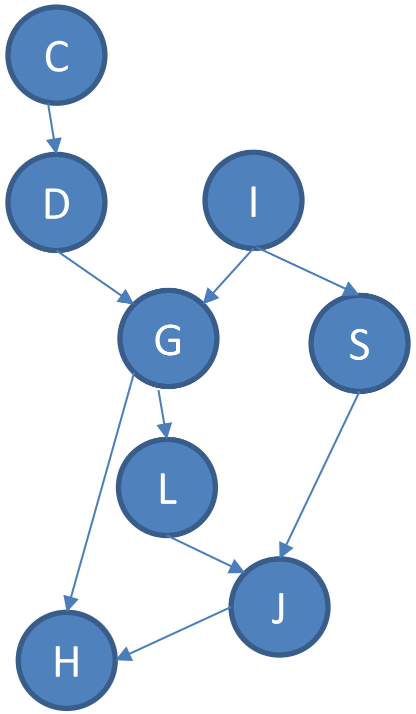
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

Sum-product Inference

$$\phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)$$
$$\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,G,J)$$

Compute $P(J,i,h)$



$$\sum_{C,D,I,G,S,L,H} \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,G,J)$$

Algorithms: Conditional Probability

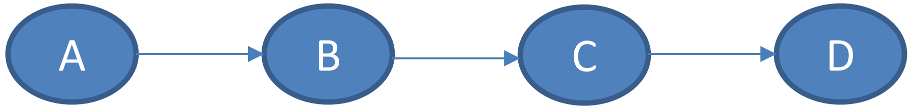
Push summations into factor product

- Variable elimination (dynamic programming)

Message passing over a graph

- Belief propagation (exact)
- Variational approximations
- Random sampling instantiations
- Markov chain Monte Carlo (MCMC)
- Importance sampling

Inference in Chains



- We want to compute *p*(*D*)
- $p(D)$ is a **set** of values, $\{p(D = d), d \in Val(D)\}$.
- Algorithm computes sets of values at a time – an entire distribution

By the chain rule and conditional independence, the joint distribution factors as

$$P(A,B,C,D) = P(A)P(B \mid A)P(C \mid B)P(D \mid C)$$

In order to compute *p*(*D*), we have to marginalize over *A, B, C*:

$$P(A,B,C,D) = \sum_{A,B,C} P(A)P(B \mid A)P(C \mid B)P(D \mid C)$$

Let's be a bit more explicit...

$$
\begin{array}{rllll}
 & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\
+ & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{rllll}
 & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\
+ & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^2 \mid c^2) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^2 \mid c^2)
\end{array}
$$

- There is structure to the summation, e.g., repeated $P(a^1)P(b^1|a^1)+P(a^2)P(b^1|a^2)$
- Let's modify the computation to first compute

$$P(a^1)P(b^1|a^1) + P(a^2)P(b^1|a^2)$$

Let's be a bit more explicit...

- Let's modify the computation to first compute

$$P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)$$

and

$$P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)$$

- Then, we get

$$\begin{array}{ll}
 & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) \quad P(c^1 \mid b^1) \quad P(d^1 \mid c^1) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) \quad P(c^1 \mid b^2) \quad P(d^1 \mid c^1) \\
+ & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) \quad P(c^2 \mid b^1) \quad P(d^1 \mid c^2) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) \quad P(c^2 \mid b^2) \quad P(d^1 \mid c^2)
\end{array}$$

$$\begin{array}{ll}
 & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) \quad P(c^1 \mid b^1) \quad P(d^2 \mid c^1) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) \quad P(c^1 \mid b^2) \quad P(d^2 \mid c^1) \\
+ & (P(a^1)P(b^1 \mid a^1) + P(a^2)P(b^1 \mid a^2)) \quad P(c^2 \mid b^1) \quad P(d^2 \mid c^2) \\
+ & (P(a^1)P(b^2 \mid a^1) + P(a^2)P(b^2 \mid a^2)) \quad P(c^2 \mid b^2) \quad P(d^2 \mid c^2)
\end{array}$$

- We define $\tau_1 : Val(B) \rightarrow R$, $\tau_1(b^i) = P(a^1)P(b^i \mid a^1) + P(a^2)P(b^i \mid a^2)$

Let's be a bit more explicit...

- We now have

$$
\begin{array}{rlll}
 & \tau_1(b^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\
+ & \tau_1(b^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\
+ & \tau_1(b^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\
+ & \tau_1(b^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{rlll}
 & \tau_1(b^1) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\
+ & \tau_1(b^2) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\
+ & \tau_1(b^1) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\
+ & \tau_1(b^2) & P(c^2 \mid b^2) & P(d^2 \mid c^2)
\end{array}
$$

- We can once more reverse the order of the product and the sum and get

$$
\begin{array}{rll}
 & (\tau_1(b^1)P(c^1 \mid b^1) + \tau_1(b^2)P(c^1 \mid b^2)) & P(d^1 \mid c^1) \\
+ & (\tau_1(b^1)P(c^2 \mid b^1) + \tau_1(b^2)P(c^2 \mid b^2)) & P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{rll}
 & (\tau_1(b^1)P(c^1 \mid b^1) + \tau_1(b^2)P(c^1 \mid b^2)) & P(d^2 \mid c^1) \\
+ & (\tau_1(b^1)P(c^2 \mid b^1) + \tau_1(b^2)P(c^2 \mid b^2)) & P(d^2 \mid c^2)
\end{array}
$$

- There are still other repeated computations!

Let's be a bit more explicit...

- We define $\tau_2 : \mathrm{Val}(C) \longrightarrow R$, with

$$\begin{aligned}
\tau_2(c^1) &= \tau_1(b^1)P(c^1|b^1) + \tau_1(b^2)P(c^1|b^2) \\
\tau_2(c^2) &= \tau_1(b^1)P(c^2|b^1) + \tau_1(b^2)P(c^2|b^2)
\end{aligned}$$

- Now we can compute the marginal $p(D)$ as

$$\begin{array}{ll}
& \tau_2(c^1) \quad P(d^1 \mid c^1) \\
+ & \tau_2(c^2) \quad P(d^1 \mid c^2)
\end{array}$$

$$\begin{array}{ll}
& \tau_2(c^1) \quad P(d^2 \mid c^1) \\
+ & \tau_2(c^2) \quad P(d^2 \mid c^2)
\end{array}$$

What Did We Do?

$$P(D) = \sum_C \sum_B \sum_A P(A)P(B \mid A)P(C \mid B)P(D \mid C)$$

Push in the summation of A

$$P(D) = \sum_C \sum_B P(C \mid B)P(D \mid C) \sum_A P(A)P(B \mid A)$$

Push in the summation of B

$$P(D) = \sum_C P(D \mid C) \sum_B P(C \mid B)\ \mathrm{P(B)}$$

Push in the summation of C

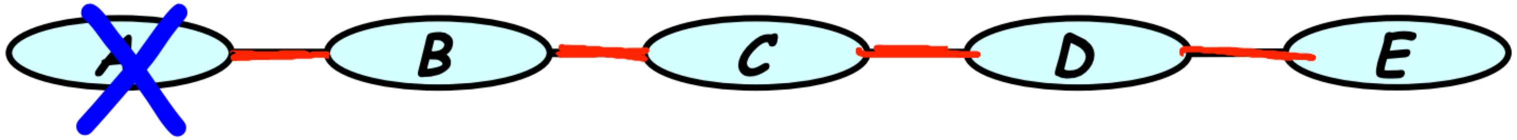
$$P(D) = \sum_C P(D \mid C)\mathrm{P(C)}$$

Rule for Sum-Product VE

If $X \notin Scope[\phi_1]$, then

$$\sum_X (\phi_1 \cdot \phi_2) = \phi_1 \cdot \sum_X \phi_2$$

Elimination In Chains: MNs



$$P(E) \propto \sum_D \sum_C \sum_B \sum_A \tilde{P}(A,B,C,D,E)$$

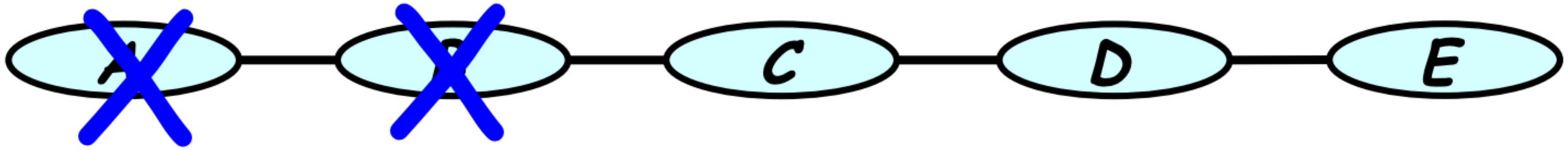
$$= \sum_D \sum_C \sum_B \sum_A \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,E)$$

$$= \sum_D \sum_C \sum_B \phi_2(B,C)\phi_3(C,D)\phi_4(D,E)\underbrace{\sum_A \phi_1(A,B)}_{\tau_1(B)}$$

$$= \sum_D \sum_C \sum_B \phi_2(B,C)\phi_3(C,D)\phi_4(D,E)\tau_1(B)$$

Daphne Koller

Elimination In Chains: MNs

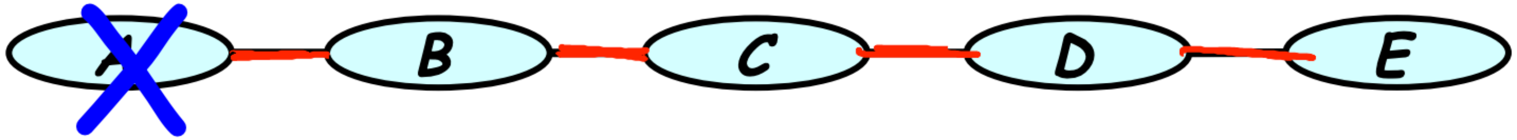


$$P(E) \propto \sum_D \sum_C \sum_B \phi_2(B,C)\phi_3(C,D)\phi_4(D,E)\tau_1(B)$$

$$= \sum_D \sum_C \phi_3(C,D)\phi_4(D,E)\overbrace{\left(\sum_B \phi_2(B,C)\tau_1(B)\right)}^{\tau_2(c)}$$

$$= \sum_D \sum_C \phi_3(C,D)\phi_4(D,E)\tau_2(C)$$

Elimination In Chains: MNs



$$P(E) \propto \sum_D \sum_C \sum_B \sum_A \tilde{P}(A,B,C,D,E)$$

$$= \sum_D \sum_C \sum_B \sum_A \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,E)$$

$$= \sum_D \sum_C \sum_B \phi_2(B,C)\phi_3(C,D)\phi_4(D,E) \underbrace{\sum_A \phi_1(A,B)}_{\tau_1(B)}$$

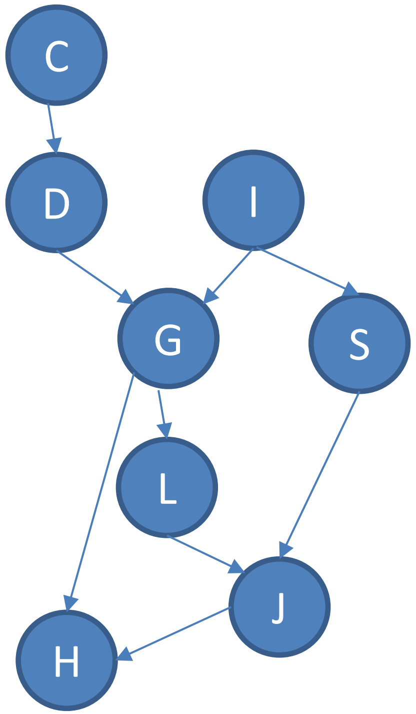
$$= \sum_D \sum_C \sum_B \phi_2(B,C)\phi_3(C,D)\phi_4(D,E)\tau_1(B)$$

Daphne Koller

Variable Elimination

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L

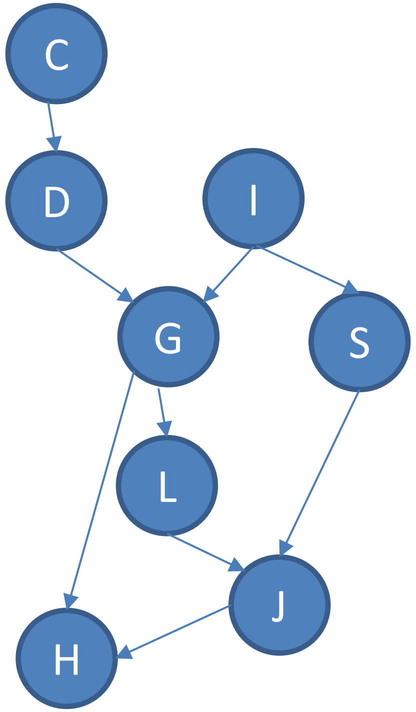
$$\sum_{C,D,I,G,S,L,H} \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,G,J)$$



Variable Elimination with evidence

- Goal: P(J, I =i, H=h)
- Eliminate: C,D, G,S,L

$$\sum_{C,D,I,G,S,L,H} \phi_C(C)\phi_D(C,D)\phi_I(I)\phi_G(G,I,D)\phi_S(S,I)\phi_L(L,G)\phi_J(J,L,S)\phi_H(H,G,J)$$



Variable Elimination in MNs

- Goal: P(D)
- Eliminate: A,B,C

$$\sum_{A,B,C} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(A,D)$$

$$\sum_{B,C} \phi_2(B,C)\phi_3(C,D)\underbrace{\overbrace{\sum_{A} \phi_1(A,B)\phi_4(A,D)}^{A,B,D}}_{\tau_1(B,D)}$$

$$\sum_{B,C} \phi_2(B,C)\phi_3(C,D)\tau_1(B,D)$$

At the end of elimination get $\tau_3(D)$ $= \tilde{P}(D)$ $\propto P(D)$

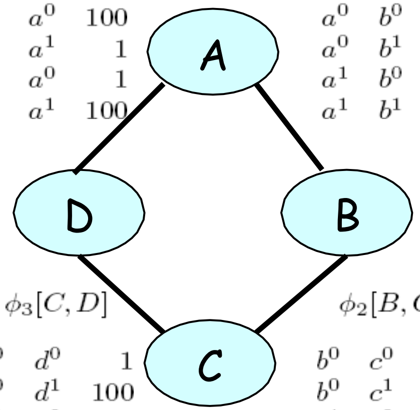
renormalize

$\phi_4[D, A]$

d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

$\phi_1[A, B]$

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10



$\phi_3[C, D]$

c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

$\phi_2[B, C]$

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

- Reduce all factors by evidence
  - Get a set of factors Φ
- For each non-query variable Z
  - Eliminate-Var Z from Φ:

$$
\begin{aligned}
&\Phi' = \{\phi_i \in \Phi \ : \ Z \in Scope[\phi_i]\} \\
&\psi = \prod_{\phi_i \in \Phi'} \phi_i \\
&\tau = \sum_Z \psi \\
&\Phi := \Phi - \Phi' \cup \{\tau\}
\end{aligned}
$$

- Multiply all remaining factors
- Renormalize to get distribution

Summary

- Simple algorithm
- Works for both BNs and MNs
- Factor product and summation steps can be done in any order, subject to:
  - when Z is eliminated, all factors involving Z have been multiplied in