Probabilistic Graphical Models

Lecture 10 – Undirected Models

CS/CNS/EE 155

Andreas Krause

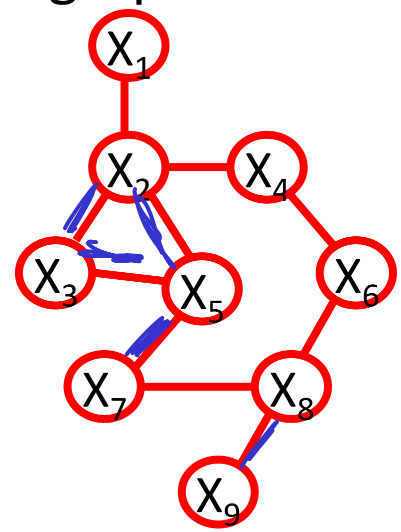
Announcements

- Homework 2 due this Wednesday (Nov 4) in class
- Project milestones due next Monday (Nov 9)
  - About half the work should be done
  - 4 pages of writeup, NIPS format
  - http://nips.cc/PaperInformation/StyleFiles

Markov Networks
(a.k.a., Markov Random Field, Gibbs Distribution, …)

- A Markov Network consists of
  - An undirected graph, where each node represents a RV
  - A collection of factors defined over cliques in the graph
- Joint probability

$$P(X) = \frac{1}{Z} \prod_i \psi_i(C_i)$$



P

- A distribution factorizes over undirected graph G if

$\exists$ factors $\psi_1 \cdots \psi_k$ over cliques of G s.t.

$$P(X) = \frac{1}{Z} \prod_i \psi_i(C_i)$$

Computing Joint Probabilities

- Computing joint probabilities in BNs

$$P(X_1, \ldots, X_n) = \prod_i P(X_i \mid Pa_i)$$

$P(X_1 \mid X_n)$

actually comp. $P(X_1, X_n)$

$$Z = \sum_x \prod_i \psi_i(C_i)$$

- Computing joint probabilities in Markov Nets

$$P(\underline{X_1 \ldots X_n}) = \frac{1}{\underline{Z}} \prod_i \psi_i(C_i)$$

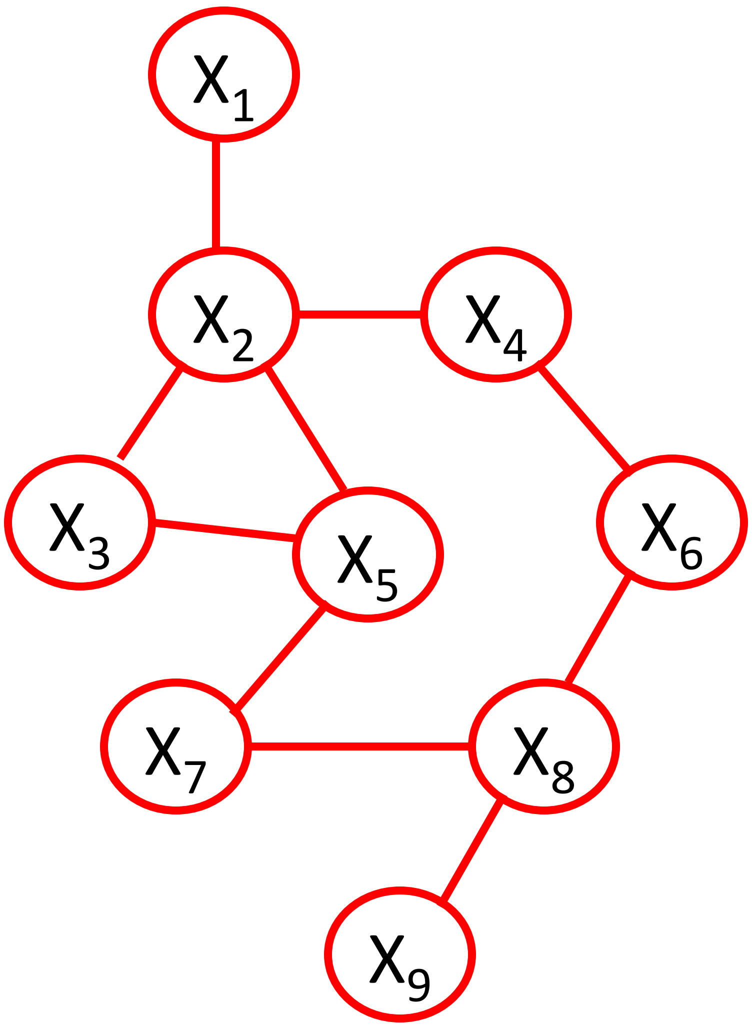
Can do VE

variable elimination

Need to know partition "function" Z

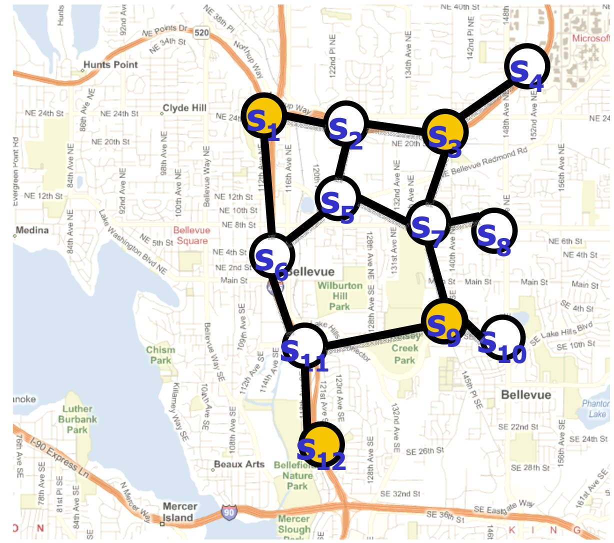
Can compute $\dfrac{P(x_1 \ldots x_n)}{P(x_1' \ldots x_n')} = \dfrac{\prod_i \psi_i(C_i)}{\prod_i \psi_i(C_i')}$

Local Markov Assumption for MN



- The **Markov Blanket MB(X)** of a node X is the set of neighbors of X
- Local Markov Assumption: $X \perp \text{EverythingElse} \mid MB(X)$
- $I_{loc}(G)$ = set of all local independences
- G is called an I-map of distribution P if $I_{loc}(G) \subseteq I(P)$

Factorization Theorem for Markov Nets "➔"



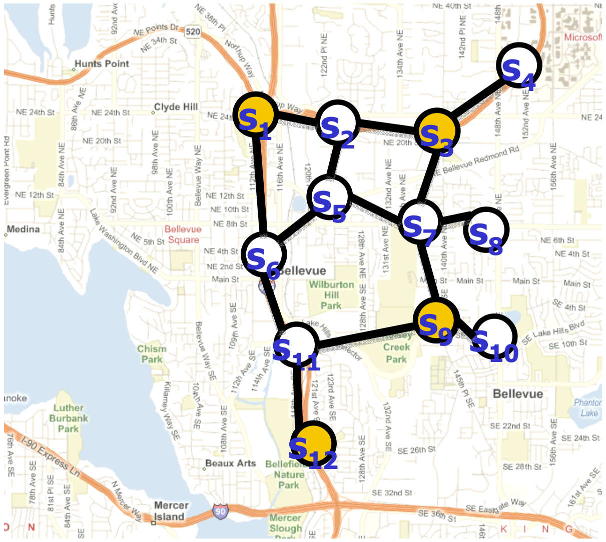
True distribution P
can be represented exactly as
a Markov net (G,P)

$$P(X_1, ..., X_n) = \frac{1}{Z} \prod_i \phi_i(\mathbf{C}_i)$$

$I_{loc}(G) \subseteq I(P)$

G is an **I-map** of P
(independence map)

Factorization Theorem for Markov Nets “⇐” Hammersley-Clifford Theorem



True distribution P
can be represented exactly as

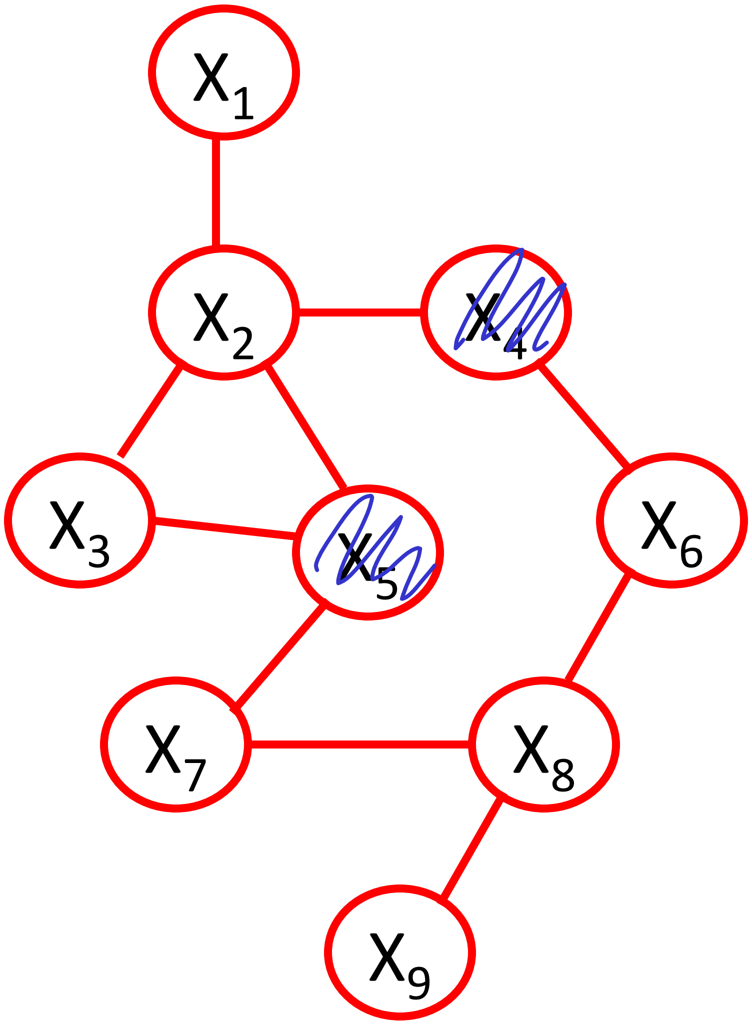
$I_{loc}(G) \subseteq I(P)$ ⇨

$$P(X_1, \ldots, X_n) = \prod_i P(X_i \mid \mathbf{Pa}_{X_i})$$

G is an **I-map** of P
(independence map)
and P>0

i.e., P can be represented as
a Markov net (G,P)

Global independencies



- A trail $X—X_1—\ldots—X_m—Y$ is called active for evidence E, if none of $X_1,\ldots,X_m \in \mathbf{E}$
- Variables X and Y are called **separated** by **E** if there is no active trail for **E** connecting X, Y
  Write sep(X,Y | **E**)
- $I(G) = \{X \perp Y \mid \mathbf{E}: sep(X,Y|\mathbf{E})\}$

Soundness of separation

- Know: For positive distributions P>0

  $I_{loc}(G) \subseteq I(P) \Leftrightarrow$ P factorizes over G

- **Theorem**: Soundness of separation

  For positive distributions P>0

  $I_{loc}(G) \subseteq I(P) \Leftrightarrow I(G) \subseteq I(P)$

- Hence, separation captures only true independences

- How about $I(G) = I(P)$?

Completeness of separation

Theorem: Completeness of separation

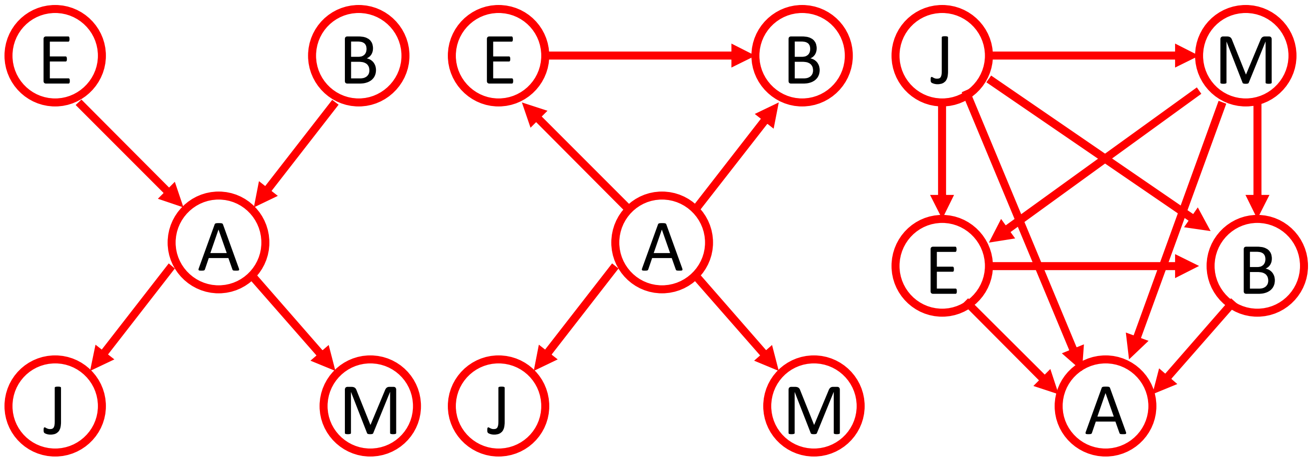
$$I(G) = I(P)$$

for "almost all" distributions P that factorize over G

"almost all": Except for of potential parameterizations of measure 0 (assuming no finite set have positive measure)

Minimal I-maps

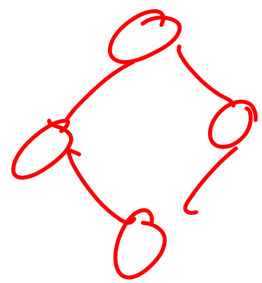
- For BNs: Minimal I-map not unique



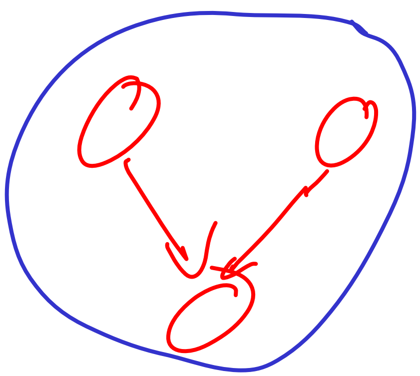
- For MNs: For positive P, minimal I-map is unique!!

P-maps

- Do P-maps always exist?
- For BNs: no



- How about Markov Nets?



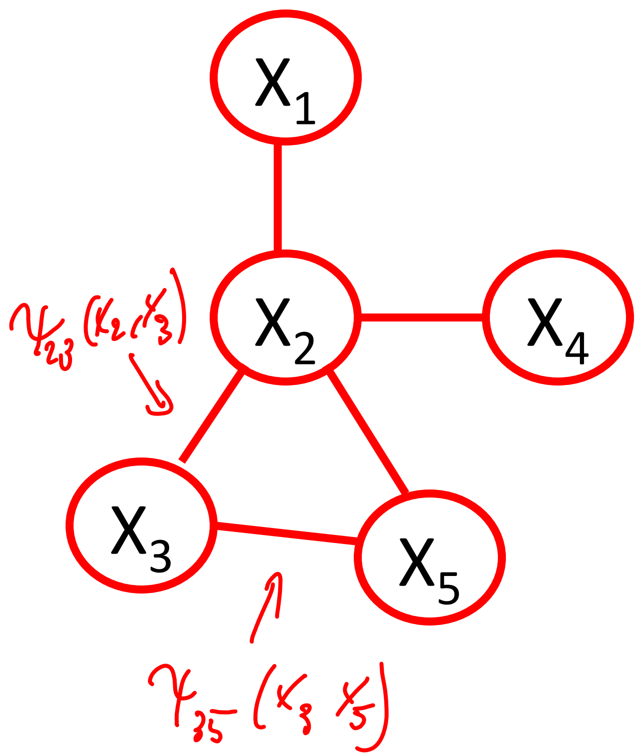
does not have
MN P-map!

Exact inference in MNs

- Variable elimination and junction tree inference work exactly the same way!
  - Need to construct junction trees by obtaining chordal graph through triangulation

Pairwise MNs

- A pairwise MN is a MN where all factors are defined over single variables or pairs of variables
- Can reduce any MN to pairwise MN!



Logarithmic representation

- Can represent any positive distribution in log domain

$$P(X) = \frac{1}{Z} \prod_i \psi_i(C_i)$$

$$\log P(X) = \sum_i \underbrace{\log \psi_i(C_i)}_{\varphi_i(C_i)} - \log Z$$

$$P(X) = \frac{1}{Z} \exp\left(\sum_i \varphi_i(C_i) \right)$$

Log-linear models

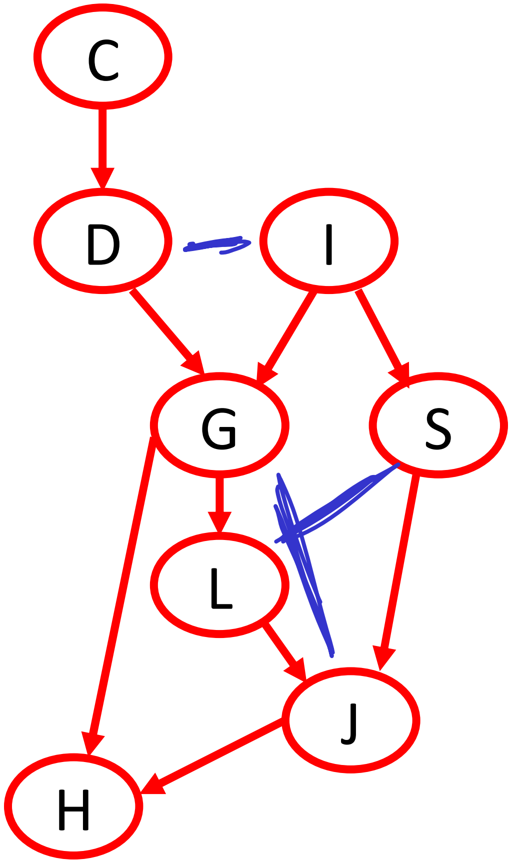
- Feature functions $\phi_i(D)$ defined over cliques

$$\phi_i(X_i, X_{i+1}) = \begin{cases} 1 & \text{if } X_i = X_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

- Log linear model over undirected graph G
  - Feature functions $\phi_1(D_1),\ldots,\phi_k(D_k)$
  - Domains D_i can overlap
  - Set of weights w_i learnt from data

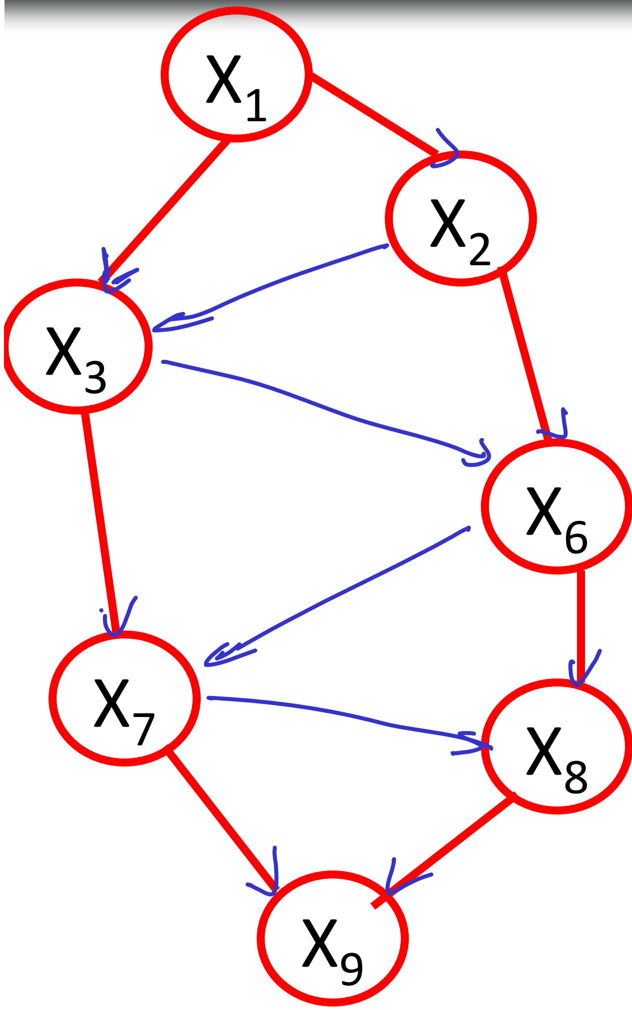
$$P(X) = \frac{1}{Z} \exp\left(\sum_i w_i^T \phi_i(C_i) \right)$$

Converting BNs to MNs



Theorem: Moralized Bayes net is minimal Markov I-map

Converting MNs to BNs



Resulting BN has far fewer cond. independencies than original MN

$I(G') \subseteq I(G)$

BN MN

Theorem: Minimal Bayes I-map for MN must be chordal

So far

- Markov Network **Representation**
  - Local/Global Markov assumptions; Separation
  - Soundness and completeness of separation
- Markov Network **Inference**
  - Variable elimination and Junction Tree inference work exactly as in Bayes Nets
- How about **Learning** Markov Nets?

Parameter Learning for Bayes nets

$$\log P(D \mid \theta) = \log \prod_{\ell} \prod_{i} P(X_i^{(\ell)} \mid Pa_i^{(\ell)}; \theta)$$

$$= \sum_{\ell} \sum_{i} \log P(X_i^{(\ell)} \mid Pa_i^{(\ell)}; \theta_{X_i \mid Pa_i})$$

Parameter independ.

$$\frac{\partial}{\partial \theta_{X_i \mid Pa_i}} \log P(D \mid \theta) = \sum_{j} \sum_{\ell} \frac{\partial}{\partial \theta_{X_i \mid Pa_i}} \log P(X_j^{(\ell)} \mid Pa_j^{(\ell)}; \theta_{X_j \mid Pa_j})$$

$$= \sum_{\ell} \frac{\partial}{\partial \theta_{X_i \mid Pa_i}} \log P(X_i^{(\ell)} \mid Pa_i, \theta_{X_i \mid Pa_i}) \stackrel{!}{=} 0$$

Problem breaks down into independent subproblems

Learn every CPD independent of others

Algorithm for BN MLE

Given BN structure G

For each variable X_i

learn $\hat{\theta}_{X_i|Pa_i} = \dfrac{\text{Count}(X_i, Pa_i)}{\text{Count}(Pa_i)}$

$\Rightarrow$ globally maximum likelihood estimate for fixed structure G

MLE for Markov Nets

- Log likelihood of the data

$$\log P(D \mid \theta) = \sum_{\ell} \log P(x^{(\ell)} \mid \theta)$$

$$= \sum_{\ell=1}^{m} \log \frac{1}{Z} \prod_i \psi_i(c_i^{(\ell)})$$

$$= \sum_{\ell} \sum_{i} \log \psi_i(c_i^{(\ell)}) - m \log Z$$

$$= m \sum_{i} \sum_{c_i} \hat{P}(c_i) \underbrace{\log \psi_i(c_i)}_{\text{In BN } \log P(X_i \mid Pa_i)} - \underbrace{m \log Z}_{\text{not in B}}$$

$$Z = Z(\theta) = \sum_{x} \prod_i \psi_i(c_i)$$

vary θ

$$\log Z = \log \sum_{x} \prod_i \psi_i(c_i)$$

Log-likelihood doesn't decompose

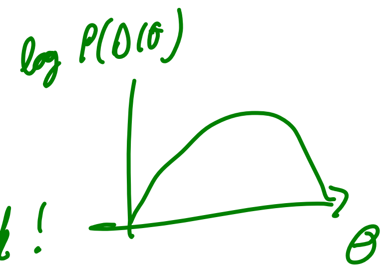
- Log likelihood

$$\log P(\mathcal{D} \mid \theta) = \underbrace{m \sum_i \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i)}_{\text{decomposes nicely}} - \underbrace{m \log Z(\theta)}_{\text{does not decompose}}$$

- l(D | θ) is concave function!

  No local optima!

  Gradient ascent won't get stuck!



- Log Partition function log Z(θ) doesn't decompose

Derivative of log-likelihood

$$\log P(\mathcal{D} \mid \theta) = m \sum_i \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i) - m \log Z(\theta)$$

$$\frac{\partial \log P(\mathcal{D} \mid \theta)}{\partial \psi_i(c_i)} = \underbrace{m \sum_j \sum_{c_j} \hat{P}(c_j) \frac{\partial}{\partial \psi_i(c_i)} \log \psi_j(c_j)} - m \frac{\partial}{\partial \psi_i(c_i)} \log Z(\theta)$$

$$= m \quad \hat{P}(c_i) \frac{1}{\psi_i(c_i)} \quad - m \frac{\partial}{\partial \psi_i(c_i)} \log Z(\theta)$$

Derivative of log-likelihood

$\psi_i =$

A	B	$\psi_i(A,B)$
0	0	$\psi_i(0,0)$
0	1	$\psi_i(0,1)$
1	0	
1	1	

$$\frac{\partial \log P(\mathcal{D} \mid \theta)}{\partial \psi_i(\mathbf{c}_i)} = m \frac{\hat{P}(\mathbf{c}_i)}{\psi_i(\mathbf{c}_i)} - m \frac{\partial \log Z(\theta)}{\partial \psi(\mathbf{c}_i)}$$

$$\frac{\partial \log Z(\theta)}{\partial \psi_i(c_i)} = \frac{\frac{\partial}{\partial \psi_i(c_i)} Z(\theta)}{Z(\theta)} = \frac{\cancel{Z}\, P(c_i \mid \theta)}{\cancel{Z}\, \psi_i(c_i)} = \frac{P(c_i \mid \theta)}{\psi_i(c_i)}$$

$$\frac{\partial Z(\theta)}{\partial \psi_i(c_i)} = \frac{\partial}{\partial \psi_i} \underbrace{\sum_x \prod_j \psi_j(c_j)}_{Z(\theta)} = \sum_x \underbrace{\frac{\partial}{\partial \psi_i} \prod_j \psi_j(c_j)}_{\substack{0 \text{ if } x \nsim c_i \\ x \text{ inconsistent w } c_i}}$$

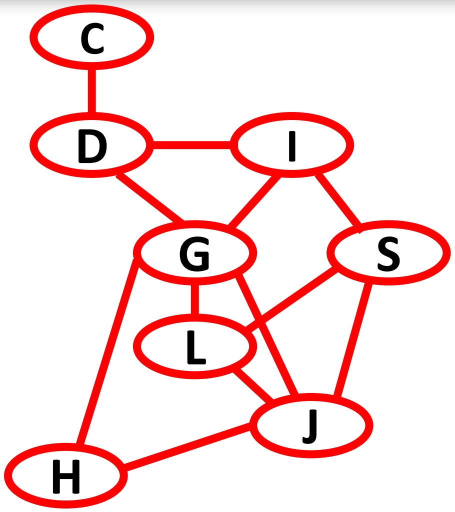
$$= \sum_{x \sim c_i} \prod_{j \neq i} \psi_j(c_j) \frac{\psi_i(c_i)}{\psi_i(c_i)}$$

$$= \frac{Z\, P(c_i \mid \theta)}{\psi_i(c_i)}$$

Computing the derivative

- Derivative

$$\frac{\partial \log P(\mathcal{D} \mid \theta)}{\partial \psi_i(\mathbf{c}_i)} = m\frac{\hat{P}(\mathbf{c}_i)}{\psi_i(\mathbf{c}_i)} - m\frac{P(\mathbf{c}_i \mid \theta)}{\psi_i(\mathbf{c}_i)}$$



$$\frac{\partial \log P(D \mid \theta)}{\partial \psi_G(G=1, D=0, I=1)} = m\frac{\hat{P}(1,0,1)}{\psi_i(1,0,1)} - m\frac{P(1,0,1 \mid \theta)}{\psi_i(1,0,1)}$$

Can do this using VE Junction tree, ...

- Computing P(c_i | θ) requires inference!

- Can optimize using conjugate gradient etc.

Alternative approach: Iterative Proportional Fitting (IPF)

- At optimum, it must hold that

$$\frac{\partial \log P(\mathcal{D} \mid \theta)}{\partial \psi_i(\mathbf{c}_i)} = m\frac{\hat{P}(\mathbf{c}_i)}{\psi_i(\mathbf{c}_i)} - m\frac{P(\mathbf{c}_i \mid \theta)}{\psi_i(\mathbf{c}_i)} = 0$$

At opt.: $\frac{\hat{P}(c_i)}{\psi_i(c_i)} = \frac{P(c_i \mid \theta)}{\psi_i(c_i)}$ "Data agrees with model on marginals"

➔ Solve fixed point equation $\psi_i^{(0)}(c_i) = 1$

$$\psi_i^{(t+1)}(c_i) = \psi_i^{(t)}(c_i) \cdot \frac{\hat{P}(c_i)}{P(c_i \mid \theta)}$$

- Must recompute parameters every iteration

$P(c_i \mid \theta)$

Parameter learning for log-linear models

- Feature functions $\phi_i(C_i)$ defined over cliques
- Log linear model over undirected graph G
  - Feature functions $\phi_1(C_1),\ldots,\phi_k(C_k)$
  - Domains C_i can overlap
- Joint distribution

$$P(X_1,\ldots,X_n)=\frac{1}{Z}\underline{\exp\left(\sum_i w_i^T\phi_i(C_i)\right)}$$

- How do we get weights w_i?

Derivative of Log-likelihood 1

$$\frac{\partial \log P(\mathcal{D} \mid \theta)}{\partial w_i} = m \underbrace{\sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \frac{\partial w_i^T \phi_i(\mathbf{c}_i)}{\partial w_i}} - m \frac{\partial \log Z(w)}{\partial w_i}$$

$$= m \underbrace{\sum_{c_i} \hat{P}(c_i)\, \phi_i(c_i)}_{\hat{E}[\phi_i]} - m \frac{\partial \log Z(w)}{\partial w_i}$$

If $\phi_i(x_i, x_{i+1}) = \begin{cases} 1 & \text{if } x_i = x_{i+1} \\ 0 & \text{otherwise} \end{cases}$, $\hat{E}[\phi_i] = \frac{\text{Count}(x_i = x_{i+1})}{m}$

Derivative of Log-likelihood 2

$$\frac{\partial \log P(\mathcal{D} \mid \theta)}{\partial w_i} = m \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i)\phi_i(\mathbf{c}_i) - m \frac{\partial \log Z(w)}{\partial w_i}$$

$$\frac{\partial}{\partial w_i} \log Z(w) = \frac{1}{Z(w)} \frac{\partial}{\partial w_i} \sum_{x} \exp\Big(\sum_i w_i^T \phi_i(c_i)\Big)$$

$$= \frac{1}{Z(w)} \sum_{x} \phi_i(c_i) \exp\Big(\sum_i w_i^T \phi_i(c_i)\Big)$$

$$= \sum_{x} \phi_i(c_i) \underbrace{\frac{1}{Z} \exp\Big(\sum_i w_i^T \phi_i(c_i)\Big)}_{P(x)}$$

$$= \sum_{c_i} \phi_i(c_i) P(c_i \mid w)$$

$$= \mathbb{E}_w(\phi_i)$$

Optimizing parameters

- Gradient of log-likelihood

$$\frac{\partial \log P(\mathcal{D} \mid w)}{\partial w_i} = m \underbrace{\sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i)\phi_i(\mathbf{c}_i)}_{\hat{\mathbb{E}}(\phi_i)} - m \underbrace{\sum_{\mathbf{c}_i} P(\mathbf{c}_i \mid w)\phi_i(\mathbf{c}_i)}_{\mathbb{E}_w(\phi_i)}$$

- Thus, w is MLE ⇔ $\hat{\mathbb{E}}[\phi_i] = \mathbb{E}_w[\phi_i]$

Regularization of parameters

- Put prior on parameters w $P(w)$

$$\frac{\partial \log P(\mathcal{D} \mid w) P(w)}{\partial w_i} = \underbrace{m \sum_{\mathbf{c}_i} \hat{P}(\mathbf{c}_i) \phi_i(\mathbf{c}_i) - m \sum_{\mathbf{c}_i} P(\mathbf{c}_i \mid w) \phi_i(\mathbf{c}_i)}_{\text{last slide}} + \underbrace{\frac{\partial \log P(w)}{\partial w_i}}_{(*)}$$

Prior: $P(w) = \mathcal{N}(w; 0, I) \propto \exp\left(-\sum_i w_i^2\right)$

$(*)\quad \log P(w) = -\sum_i w_i^2$

$\frac{\partial}{\partial w_i} \log P(w) = -2 w_i$

Summary: Parameter learning in MN

- MLE in BN is easy (score decomposes)
- MLE in MN requires inference (score doesn’t decompose)
- Can optimize using gradient ascent or IPF

Tasks

- Read Koller & Friedman Chapters 20.1-20.3, 4.6.1