STA 414/2104: Statistical Methods in Machine Learning II

Week 3: Markov Random Fields/Exact Inference

Murat A. Erdoğdu and **Piotr Zwiernik**

University of Toronto

Today's lecture

Summary of the content:

- Markov Random Fields (MRFs).
- Exact inference on graphical models
- Variable elimination

Some announcements:

- Assignment 1 is released this week.
- TA office hours next week.

Are DAGMs always useful?

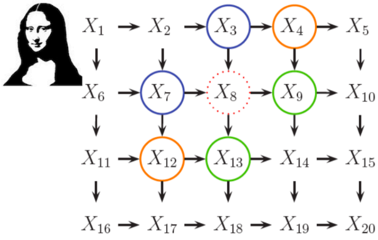


Figure : Causal MRF or a Markov mesh

- Each node is conditionally independent of its non-descendants given its parents

$$\{X_i \perp \text{non-desc}(X_i) \mid \text{parents}(X_i)\} \quad \forall i.$$

- For some problems, it is not clear how to choose the edge directions in DAGMs.

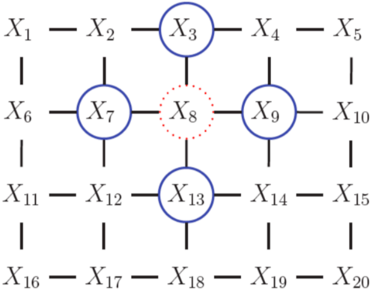
Markov blanket (mb): the set of nodes that makes X_i conditionally independent of all the other nodes.

In our example: $\text{mb}(X_8) = \{X_3, X_4, X_7, X_9, X_{12}, X_{13}\}$.

One would expect X_4 and X_{12} not to be in the Markov blanket $\text{mb}(X_8)$, especially given X_2 and X_{14} are not.

Markov Random Fields

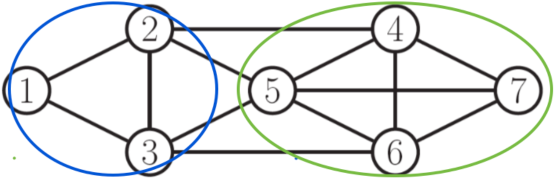
- Undirected graphical models (aka **Markov random fields (MRFs)**) are models with dependencies described by an undirected graph.
- The nodes in the graph represent random variables. However, in contrast to DAGMs, edges represent probabilistic interactions between neighbors (as opposed to conditional dependence).



Cliques

A **clique** is a subset of nodes such that every two vertices in the subset are connected by an edge.

A **maximal clique** is a clique that cannot be extended by including one more adjacent vertex.



Distributions Induced by MRFs

Let $X = (X_1, ..., X_m)$ be the set of all random variables in our graph G.

Let $\mathcal{C}$ be the set of all maximal cliques of G.

The distribution p of X factorizes with respect to G if

$$p(x) \propto \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

for some nonnegative *potential* functions ψ_C, where $x_C = (x_i)_{i \in C}$.

The MRF on G represents the distributions that factorize wrt G.

The factored structure of the distribution makes it possible to more efficiently do the sums/integrals and is a form of dimension reduction.

Hammersley-Clifford Theorem

If $p(x) > 0$ for all x, the following statements are equivalent:

- p factorizes according to G, that is,

$$p(x) \;\propto\; \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

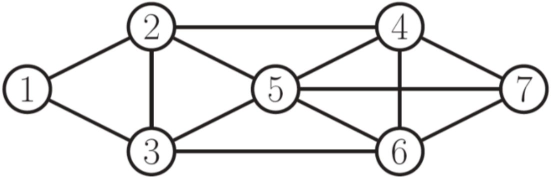
  for some nonnegative potential functions ψ_C.

- **Global Markov Properties**: $X_A \perp X_B | X_S$ if the sets A and B are *separated* by S in G (every path from A to B crosses S).

In particular,

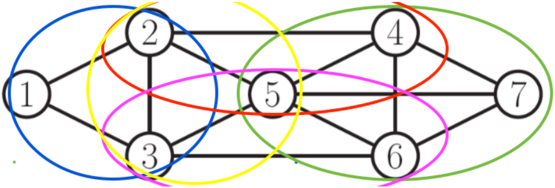
- If i, j are not connected by an edge then $X_i \perp X_j | X_{\text{rest}}$.
- The Markov blanket of X_i is given by its neighbors in G.

Example:



- How many maximal cliques are there?
- What is the underlying factorization?
- What are the induced conditional independence statements?

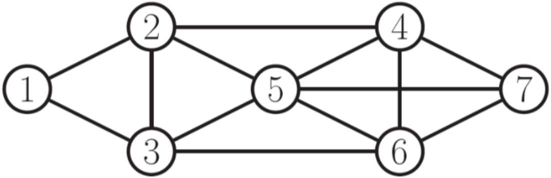
Example:



Lets see how to factorize the undirected graph of our running example:

$$p(x) \propto \psi_{1,2,3}(x_1, x_2, x_3)\psi_{2,3,5}(x_2, x_3, x_5)\psi_{2,4,5}(x_2, x_4, x_5) \times \psi_{3,5,6}(x_3, x_5, x_6)\psi_{4,5,6,7}(x_4, x_5, x_6, x_7)$$

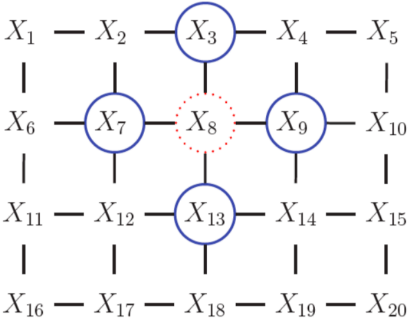
Example:



e.g. $$(X_1, X_2) \perp (X_6, X_7) \mid (X_3, X_4, X_5)$$

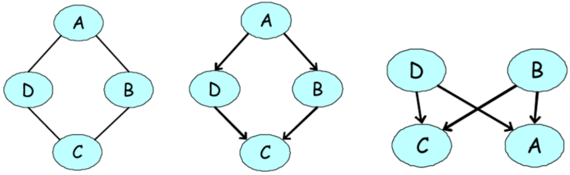
$$X_1 \perp X_5 \mid (X_2, X_3)$$

Image MRF


X_1 X_2 X_3 X_4 X_5
X_6 X_7 X_8 X_9 X_{10}
X_{11} X_{12} X_{13} X_{14} X_{15}
X_{16} X_{17} X_{18} X_{19} X_{20}

Not all MRFs can be represented as DAGMs

Take the following MRF for example (a) and our attempts at encoding this as a DAGM (b, c).

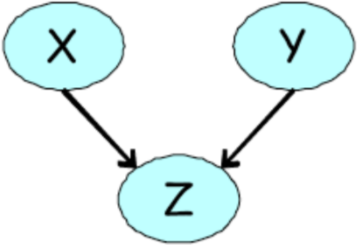


(a) (b) (c)

- Two conditional independencies in (a):
  - 1. $A \perp C | D, B$ 2. $B \perp D | A, C$
- In (b), we have the first independence, but not the second.
- In (c), we have the first independency, but not the second. Also, B and D are marginally independent.

Not all DAGMs can be represented as MRFs

Not all DAGMs can be represented as MRFs.
E.g. explaining away:



An undirected model is unable to capture the marginal independence, $X \perp Y$ that holds at the same time as $X \not\perp Y | Z$.

MRFs as Exponential Families

- Consider a parametric family of factorized distributions

$$p(x|\theta) \;=\; \frac{1}{Z(\theta)} \prod_{C \in \mathcal{C}} \psi_C(x_C|\theta_C), \qquad \theta = (\theta_C)_{C \in \mathcal{C}}.$$

- We can write this in an exponential form:

$$p(x|\theta) = \exp\left\{ \sum_{C \in \mathcal{C}} \log \psi_C(x_C|\theta_C) - \underbrace{\log Z(\theta)}_{=A(\theta)} \right\}$$

- Suppose the potentials have a log-linear form

$$\log \psi_C(x_C|\theta_C) = \theta_C^\top \phi_C(x_C)$$

we get the exponential family

$$p(x|\theta) = \exp\left\{ \sum_{C \in \mathcal{C}} \theta_C^\top \phi_C(x_C) - \underbrace{\log Z(\theta)}_{=A(\theta)} \right\}$$

MRFs as Exponential Families

Question: When $\log \psi_C(x_C|\theta_C) = \theta_C^\top \phi_C(x_C)$?

Finite discrete case:

- If X is finite discrete then x_C takes a finite number of values and so $\log \psi_C$ takes a finite number of values.
- Take θ_C as all these possible values, and let $\phi_C(x_C)$ is a vector 1 on the entry correspond to x_C and zeros otherwise.
- Then $\log \psi_C(x_C|\theta_C) = \theta_C^\top \phi_C(x_C)$ as required.

Multivariate Gaussian case will be covered later in the lecture.

We can find the expectation of the C-th feature

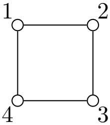
$$\frac{\partial \log Z(\theta)}{\partial \theta_C} = \mathbb{E}[\phi_C(X_C)].$$

Representing potentials

If the variables are finite discrete, we can represent the potential functions as tables of (non-negative) numbers.

e.f. consider a 4-cycle and binary random variables

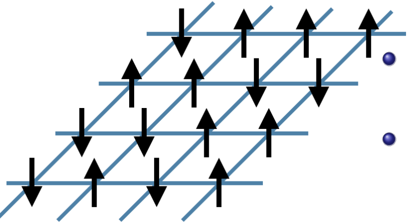
$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\psi_{3,4}(x_3, x_4)\psi_{1,4}(x_1, x_4)$$



$\psi_{1,2}(x_1, x_2)$			$\psi_{2,3}(x_2, x_3)$			$\psi_{3,4}(x_3, x_4)$			$\psi_{1,4}(x_1, x_4)$		
x_1	x_2		x_2	x_3		x_3	x_4		x_1	x_4	
0	0	30	0	0	100	0	0	1	0	0	100
0	1	5	0	1	1	0	1	100	0	1	1
1	0	1	1	0	1	1	0	100	1	0	1
1	1	10	1	1	100	1	1	1	1	1	100

These potentials are not probabilities. Even after normalization they will not, in general, correspond to marginal distributions.

Example: Ising model



- The Ising model is an MRF that is used to model magnets.
- The nodes variables are spins, i.e., we use $x_s \in \{-1, +1\}$.

- Define the pairwise clique potentials as

$$\psi_{st}(x_s, x_t) \;=\; e^{J_{st} x_s x_t}.$$

  where J_{st} is the coupling strength between nodes s and t.
- $\psi_{st}(-1,-1) = \psi_{st}(1,1) = e^{J_{st}}; \quad \psi_{st}(-1,1) = \psi_{st}(1,-1) = e^{-J_{st}}$
- If two nodes are not connected set $J_{st} = 0$.

Ising model

- We might want to add node potentials as well

$$\psi_s(x_s) = e^{b_s x_s}$$

- The overall distribution becomes

$$p(x) \propto \prod_{s\sim t} \psi_{st}(x_s, x_s) \prod_s \psi_s(x_s) = \exp\left\{ \sum_{s\sim t} J_{st} x_s x_t + \sum_s b_s x_s \right\}.$$

- Conditional log-odds ratio: $\log \frac{p(-1,-1,x_{\text{rest}})p(1,1,x_{\text{rest}})}{p(-1,1,x_{\text{rest}})p(1,-1,x_{\text{rest}})} = 4J_{st}$.
- If $J_{st} > 0$ the model promotes same spins on neighboring spins.
- Hammersley-Clifford theorem: $J_{ij} = 0$ then $X_i \perp X_j | X_{\text{rest}}$.

Multivariate Gaussian distribution

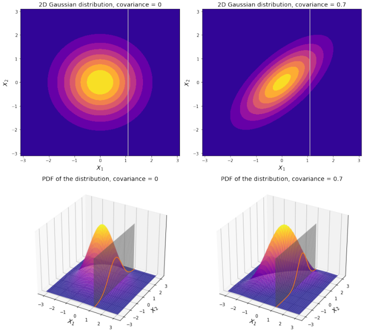
Univariate Gaussian: $f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}(x-\mu)^2)$.

Multivariate normal distribution, $X = (X_1, \ldots, X_m)$:

Let $\mu \in \mathbb{R}^m$ and Σ symmetric positive definite $m \times m$ matrix. We write $X \sim N_m(\mu, \Sigma)$ if the density of the vector X is

$$f(\boldsymbol{x};\mu,\Sigma) = \frac{1}{(2\pi)^{m/2}}(\det\Sigma)^{-1/2}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\mu)^T\Sigma^{-1}(\boldsymbol{x}-\mu)\right).$$

Positive definite: $\forall \boldsymbol{u} \neq \boldsymbol{0} \quad \boldsymbol{u}^\top\Sigma\boldsymbol{u} > 0$.



Moments:

- mean vector: $\mathbb{E}X = \mu$,
- covariance: $\mathrm{var}(X) = \Sigma$.

Recall: Marginal and conditional distributions

Split X into two blocks $X = (X_A, X_B)$. Denote

$$\mu = (\mu_A, \mu_B) \qquad \text{and} \qquad \Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}.$$

Marginal distribution

$X_A \sim N(\mu_A, \Sigma_{AA})$

Conditional distribution

$X_A | X_B = x_B \sim N\left(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B), \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}\right)$

- Note that the conditional covariance is constant.

Some other properties

Linear transformations:

$A \in \mathbb{R}^{m \times p}$ for $m \leq p$ and $X \sim N_p(\mu, \Sigma)$ then $AX \sim N_m(A\mu, A\Sigma A^T)$.

Conditional independence:

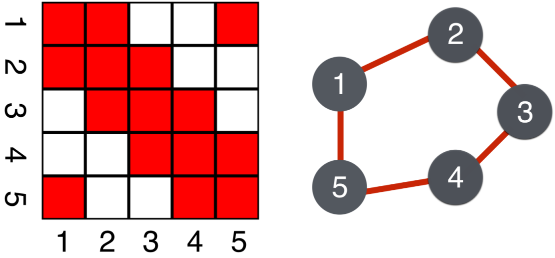
- $X_i \perp X_j$ if and only if $\Sigma_{ij} = 0$.
- $X_i \perp X_j | X_C$ if and only if $\Sigma_{ij} - \Sigma_{i,C}\Sigma_{C,C}^{-1}\Sigma_{C,j} = 0$
- Let $R = V \setminus \{i, j\}$. The following are equivalent:
  - $X_i \perp X_j | X_R$
  - $\Sigma_{ij} - \Sigma_{i,R}\Sigma_{R,R}^{-1}\Sigma_{R,j} = 0$
  - $(\Sigma^{-1})_{ij} = 0$

Gaussian Graphical models

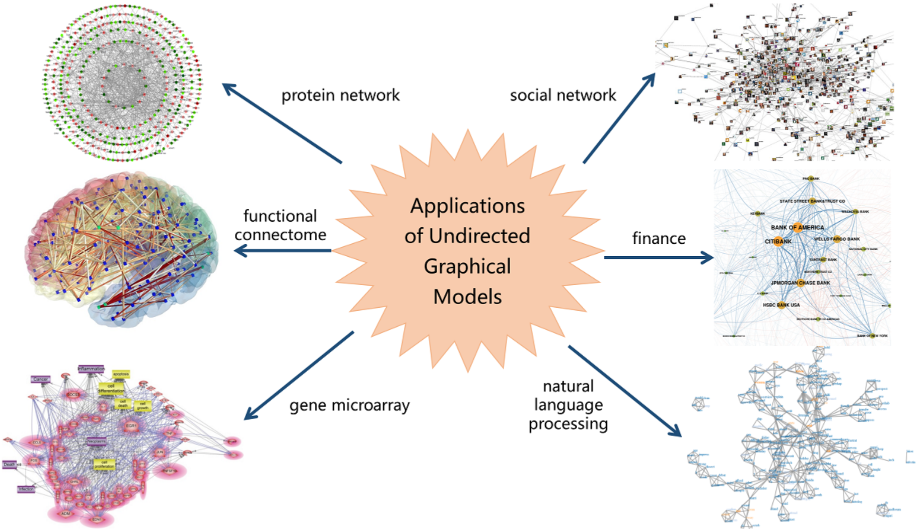
Denote $K = \Sigma^{-1}$ then

$$f(\boldsymbol{x}; \mu, \Sigma) \propto \prod_{s} e^{-\frac{1}{2} K_{ss}(x_s - \mu_s)^2} \prod_{s<t} e^{-K_{st}(x_s - \mu_s)(x_t - \mu_t)}.$$

Important interpretation: $K_{ij} = 0$ if and only if $X_i \perp X_j | X_{\text{rest}}$.



Show that this is an exponential family.


Applications of Undirected Graphical Models
protein network
social network
functional connectome
finance
gene microarray
natural language processing
BANK OF AMERICA
CITIBANK
WELLS FARGO BANK
JPMORGAN CHASE BANK
HSBC BANK USA
STATE STREET BANK&TRUST CO

Inference as Conditional Distribution

- We explore inference in probabilistic graphical models (PGMs).
  - x_E = The observed evidence
  - x_F = The unobserved variable we want to infer
  - $x_R = x - \{x_F, x_E\}$ = Remaining variables, extraneous to query.

- Focus on computing the conditional probability distribution

$$p(x_F|x_E) = \frac{p(x_F, x_E)}{p(x_E)} = \frac{p(x_F, x_E)}{\sum_{x_F} p(x_F, x_E)}$$

- for which, we marginalize out these extraneous variables, focussing on the joint distribution over evidence and subject of inference:

$$p(x_F, x_E) = \sum_{x_R} p(x_F, x_E, x_R)$$

Variable elimination

Order in which we marginalize affects the computational cost!

Our main tool is **variable elimination**:

- A simple and general **exact inference** algorithm in any probabilistic graphical model (DAGMs or MRFs).
- Computational complexity depends on the graph structure.
- Dynamic programming avoids enumerating all variable assignments.

Example: Simple chain

- Lets start with the example of a simple chain

$$A \to B \to C \to D$$

where we want to compute $p(D)$, with no evidence variables.

- We have

$$x_F = \{D\},\ x_E = \{\},\ x_R = \{A, B, C\}$$

- We saw last lecture that this graphical model describes the factorization of the joint distribution as:

$$p(A, B, C, D) = p(A)p(B|A)p(C|B)p(D|C)$$

- Assume each variable can take on k different values.

Example: Simple chain

- The goal is to compute the marginal $p(D)$:

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

- However, if we do this sum naively, cost is exponential $O(k^{n=4})$:

$$\begin{aligned} p(D) &= \sum_{A,B,C} p(A,B,C,D) \\ &= \sum_{C}\sum_{B}\sum_{A} p(A)p(B|A)p(C|B)p(D|C) \end{aligned}$$

- Instead, choose an **elimination ordering**:

$$\begin{aligned} p(D) &= \sum_{C,B,A} p(A,B,C,D) \\ &= \sum_{C} p(D|C)\left(\sum_{B} p(C|B)\Big(\sum_{A} p(A)p(B|A)\Big)\right). \end{aligned}$$

Example: Simple chain

- This reduces the complexity by first computing terms that appear across the other sums.

$$p(D) = \sum_C p(D|C) \sum_B p(C|B) \sum_A p(A)p(B|A)$$

- $$\begin{aligned} p(D) &= \sum_C p(D|C) \sum_B p(C|B) \sum_A p(A)p(B|A) \\ &= \sum_C p(D|C) \sum_B p(C|B)p(B) \\ &= \sum_C p(D|C)p(C) \end{aligned}$$

- The cost of performing inference on the chain in this manner is $\mathcal{O}(nk^2)$. In comparison, generating the full joint distribution and marginalizing over it has complexity $\mathcal{O}(k^n)$!

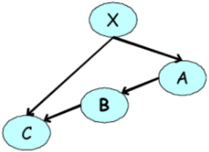
Best Elimination Ordering

- The complexity of variable elimination depends on the elimination ordering!
- Unfortunately, finding the best elimination ordering is NP-hard.

Intermediate Factors

The same algorithm both for DAGMs and MRFs:

- Introduce nonnegative **factors** ϕ (like for MRFs).
- e.g. in a simple DAG model:



$$\begin{aligned} p(A,B,C) &= \sum_X p(X)p(A|X)p(B|A)p(C|B,X) \\ &= \sum_X \phi_1(X)\phi_2(A,X)\phi_3(A,B)\phi_4(X,B,C) \\ &= \phi_3(A,B)\sum_X \phi_1(X)\phi_2(A,X)\phi_4(X,B,C) \\ &= \phi_3(A,B)\tau(A,B,C) \end{aligned}$$

- Marginalizing over X we introduce a new factor, denoted by τ.

Sum-Product Inference

- Abstractly, computing $p(x_F|x_E)$ is given by the **sum-product** algorithm:

$$p(x_F|x_E) \propto \tau(x_F, x_E) = \sum_{x_R} \prod_{C \in \mathcal{F}} \psi_C(x_C)$$

  where $\mathcal{F}$ is a set of potentials or factors.
- For DAGMs, $\mathcal{F}$ is given by the the sets of the form

$$\{i\} \cup \text{parents}(i) \qquad \text{for all nodes } i.$$

- For MRFs, $\mathcal{F}$ is given by the set of maximal cliques.

Example

- This describes a factorization:

$$p(C,D,I,G,S,L,H,J) = p(C)p(D|C)p(I)$$
$$\times p(G|D,I)p(L|G)p(S|I)p(J|S,L)p(H|J,G)$$

We have

$$\mathcal{F} = \big\{\{C\},\{C,D\},\{I\},\{G,D,I\},\{L,G\},\{S,I\},\{J,S,L\},\{H,J,G)\}\big\}$$

We are interested in the probability of getting a job, $p(J)$.

We perform exact inference as follows.

Example $\left(\mathcal{F} = \{\{C\},\{C,D\},\{I\},\{G,D,I\},\{L,G\},\{S,I\},\{J,S,L\},\{H,J,G)\}\}\right)$

Elimination Ordering $\prec \{C, D, I, H, G, S, L\}$

$$
\begin{aligned}
p(J) &= \sum_L \sum_S \psi(J,L,S) \sum_G \psi(L,G) \sum_H \psi(H,G,J) \sum_I \psi(S,I)\psi(I) \sum_D \psi(G,D,I) \underbrace{\sum_C \psi(C)\psi(C,D)}_{\tau(D)} \\
&= \sum_L \sum_S \psi(J,L,S) \sum_G \psi(L,G) \sum_H \psi(H,G,J) \sum_I \psi(S,I)\psi(I) \underbrace{\sum_D \psi(G,D,I)\tau(D)}_{\tau(G,I)} \\
&= \sum_L \sum_S \psi(J,L,S) \sum_G \psi(L,G) \sum_H \psi(H,G,J) \underbrace{\sum_I \psi(S,I)\psi(I)\tau(G,I)}_{\tau(S,G)} \\
&= \sum_L \sum_S \psi(J,L,S) \sum_G \psi(L,G)\tau(S,G) \underbrace{\sum_H \psi(H,G,J)}_{\tau(G,J)} \\
&= \sum_L \sum_S \psi(J,L,S) \underbrace{\sum_G \psi(L,G)\tau(S,G)\tau(G,J)}_{\tau(J,L,S)} \\
&= \sum_L \underbrace{\sum_S \psi(J,L,S)\tau(J,L,S)}_{\tau(J,L)} \\
&= \underbrace{\sum_L \tau(J,L)}_{\tau(J)} \\
&= \tau(J)
\end{aligned}
$$

Do we need to normalize the final τ?

Complexity of Variable Elimination Ordering

- We discussed previously that variable elimination ordering determines the computational complexity. This is due to how many variables appear inside each sum.
- Different elimination orderings will involve different number of variables appearing inside each sum.
- The complexity of the VE algorithm is

$$O(mk^{N_{\max}})$$

  where
  - m is the number of initial factors.
  - k is the number of states each random variable takes (assumed to be equal here).
  - N_i is the number of random variables inside each sum $\sum_i$.
  - $N_{\max} = \max_i N_i$ is the number of variables inside the largest sum.

Example

Elimination Ordering $\prec \{C, D, I, H, G, S, L\}$

- Here are all the initial factors:

$$\mathcal{F} = \big\{\{C\}, \{C, D\}, \{I\}, \{G, D, I\}, \{L, G\}, \{S, I\}, \{J, S, L\}, \{H, J, G)\}\big\}$$

$$\implies m = |\Phi| = 8$$

- Here are the sums, and the number of variables that appear in them

$$\underbrace{\sum_C \psi(C)\psi(C,D)}_{N_C=2} \quad \underbrace{\sum_D \psi(G,D,I)\tau(D)}_{N_D=3} \quad \underbrace{\sum_I \psi(S,I)\psi(I)\tau(G,I)}_{N_I=3}$$

$$\underbrace{\sum_H \psi(H,G,J)}_{N_H=3} \quad \underbrace{\sum_G \psi(L,G)\tau(S,G)\tau(G,J)}_{N_G=4} \quad \underbrace{\sum_S \psi(J,L,S)\tau(J,L,S)}_{N_S=3}$$

$$\underbrace{\sum_L \tau(J,L)}_{N_L=2} \implies \text{the largest sum is } N_G = 4.$$

- For simplicity, assume all variables take on k states. So the complexity of the variable elimination under this ordering is $O(8 \cdot k^4)$.

Summary

Undirected graphical models:

- MRFs are useful if there is no topological ordering in the graph.
- Cliques are key to parametrizing distributions induced by MRFs.
- Ising model and Gaussian graphical models are important example.

Variable elimination:

- Variable elimination can be used for exact inference in PGMs.
- The ordering in variable elimination can significantly reduce the computational complexity.
- The overall complexity of the variable elimination algorithm can be computed.