Readings:
K&F: 9.1, 9.2, 9.3, 9.4, 4.1, 4.2, 4.3, 4.4

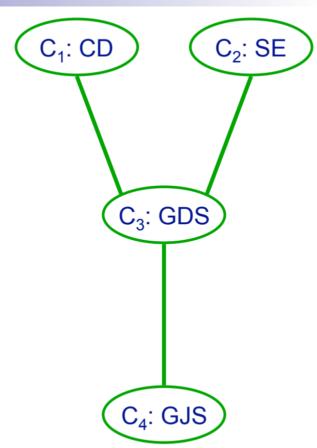
Junction Trees 3

Undirected Graphical Models

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University
October 27th, 2008

Introducing message passing with division

- Variable elimination (message passing with multiplication)
  - message:
  - belief:
- Message passing with division:
  - Belief:
  - Belief about separator:
  - message:



Factor division

- Let **X** and **Y** be disjoint set of variables
- Consider two factors: $\phi_1(\mathbf{X},\mathbf{Y})$ and $\phi_2(\mathbf{Y})$
- Factor $\psi=\phi_1/\phi_2$
  - 0/0=0

a^1	b^1	0.5
a^1	b^2	0.2
a^2	b^1	0
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.45

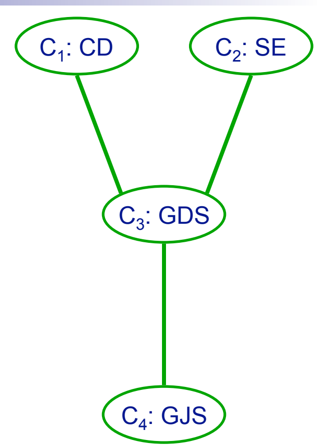
a^1	0.8
a^2	0
a^3	0.6

a^1	b^1	0.625
a^1	b^2	0.25
a^2	b^1	0
a^2	b^2	0
a^3	b^1	0.5
a^3	b^2	0.75

Lauritzen-Spiegelhalter Algorithm (a.k.a. belief propagation)

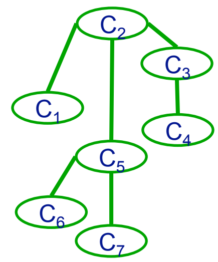
Simplified description see reading for details

- Separator potentials μ_{ij}
  - one per edge (same both directions)
  - holds "last message"
  - initialized to 1
- Message i→j
  - what does i think the separator potential should be?
    - $\sigma_{i\to j}$
  - update belief for j:
    - pushing j to what i thinks about separator
  - replace separator potential:



Convergence of Lauritzen-Spiegelhalter Algorithm

- Complexity: Linear in # cliques
  - for the "right" schedule over edges (leaves to root, then root to leaves)
- **Corollary**: At convergence, every clique has correct belief



VE versus BP in clique trees

- VE messages (the one that multiplies)
- BP messages (the one that divides)

Clique tree invariant

- **Clique tree potential**:
  - Product of clique potentials divided by separators potentials

- Clique tree **invariant**:
  - $P(\mathbf{X}) = \pi_T(\mathbf{X})$

Belief propagation and clique tree invariant

- **Theorem**: Invariant is maintained by BP algorithm!

- BP reparameterizes clique potentials and separator potentials
  - At convergence, potentials and messages are marginal distributions

Subtree correctness

- **Informed message** from i to j, if all messages into i (other than from j) are informed
  - Recursive definition (leaves always send informed messages)
- **Informed subtree**:
  - All incoming messages informed
- **Theorem**:
  - Potential of connected informed subtree *T'* is marginal over scope[*T'*]
- **Corollary**:
  - At convergence, clique tree is *calibrated*
    - π_i = P(scope[π_i])
    - μ_{ij} = P(scope[μ_{ij}])

Clique trees versus VE

- Clique tree advantages
  - Multi-query settings
  - Incremental updates
  - Pre-computation makes complexity explicit
- Clique tree disadvantages
  - Space requirements – no factors are "deleted"
  - Slower for single query
  - Local structure in factors may be lost when they are multiplied together into initial clique potential

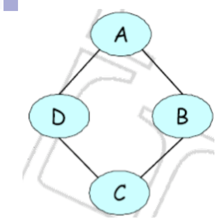
Clique tree summary

- Solve marginal queries for all variables in only twice the cost of query for one variable
- Cliques correspond to maximal cliques in induced graph
- Two message passing approaches
  - VE (the one that multiplies messages)
  - BP (the one that divides by old message)
- Clique tree invariant
  - Clique tree potential is always the same
  - We are only reparameterizing clique potentials
- Constructing clique tree for a BN
  - from elimination order
  - from triangulated (chordal) graph
- Running time (only) exponential in size of largest clique
  - Solve **exactly** problems with thousands (or millions, or more) of variables, and cliques with tens of nodes (or less)

Swinging Couples revisited

- This is no perfect map in BNs
- But, an undirected model will be a perfect map

Potentials (or Factors) in Swinging Couples



$\phi_1[A,B]$			$\phi_2[B,C]$			$\phi_3[C,D]$			$\phi_4[D,A]$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

Computing probabilities in Markov networks v. BNs

- In a BN, can compute prob. of an instantiation by multiplying CPTs
- In an Markov networks, can only compute ratio of probabilities directly

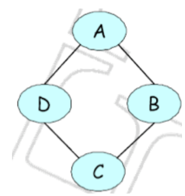
$\phi_1[A,B]$			$\phi_2[B,C]$			$\phi_3[C,D]$			$\phi_4[D,A]$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

Normalization for computing probabilities

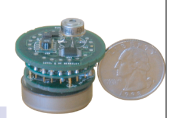
- To compute actual probabilities, must compute normalization constant (also called partition function)

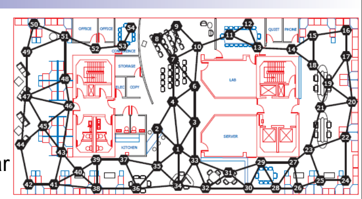
Assignment				Unnormalized	Normalized
a^0	b^0	c^0	d^0	300000	0.04
a^0	b^0	c^0	d^1	300000	0.04
a^0	b^0	c^1	d^0	300000	0.04
a^0	b^0	c^1	d^1	30	$4.1 \cdot 10^{-6}$
a^0	b^1	c^0	d^0	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^0	d^1	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^1	d^0	5000000	0.69
a^0	b^1	c^1	d^1	500	$6.9 \cdot 10^{-5}$
a^1	b^0	c^0	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^0	d^1	1000000	0.14
a^1	b^0	c^1	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^1	d^1	100	$1.4 \cdot 10^{-5}$
a^1	b^1	c^0	d^0	10	$1.4 \cdot 10^{-6}$
a^1	b^1	c^0	d^1	100000	0.014
a^1	b^1	c^1	d^0	100000	0.014
a^1	b^1	c^1	d^1	100000	0.014

- Computing partition function is hard! → Must sum over all possible assignments



Factorization in Markov networks

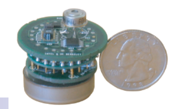




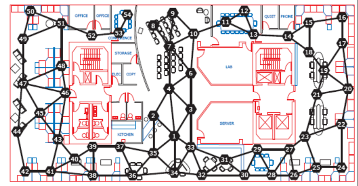
- Given an undirected graph H over variables $\mathbf{X}=\{X_1,\ldots,X_n\}$
- A distribution P **factorizes** over H if ∃
  - subsets of variables $\mathbf{D}_1 \subseteq \mathbf{X}, \ldots, \mathbf{D}_m \subseteq \mathbf{X}$, such that the $\mathbf{D}_i$ ar *fully connected* in H
  - *non-negative potentials* (or factors) $\phi_1(\mathbf{D}_1), \ldots, \phi_m(\mathbf{D}_m)$
    - also known as clique potentials
  - such that

- Also called Markov random field H, or Gibbs distribution over H

Global Markov assumption in Markov networks



- A path $X_1 - \ldots - X_k$ is **active** when set of variables **Z** are observed if none of $X_i \in \{X_1,\ldots,X_k\}$ are observed (are part of **Z**)



- Variables **X** are **separated** from **Y** given **Z** in graph *H*, $sep_H(\mathbf{X};\mathbf{Y}|\mathbf{Z})$, if there is no active path between any $X \in \mathbf{X}$ and any $Y \in \mathbf{Y}$ given **Z**

- The **global Markov assumption** for a Markov network *H* is

The BN Representation Theorem

If conditional independencies in BN are subset of conditional independencies in *P*

Obtain

Joint probability distribution:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

Important because:
Independencies are sufficient to obtain BN structure *G*

If joint probability distribution:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

Obtain

Then conditional independencies in BN are subset of conditional independencies in *P*

Important because:
Read independencies of *P* from BN structure *G*

Markov networks representation Theorem 1

If joint probability distribution P: $P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\phi_i(\mathbf{D}_i)$	Then	H is an I-map for P

- If you can write distribution as a normalized product of factors $\Rightarrow$ Can read independencies from graph

What about the other direction for Markov networks ?

If H is an I-map for P	Then	joint probability distribution P: $P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\phi_i(\mathbf{D}_i)$

- Counter-example: $X_1,\ldots,X_4$ are binary, and only eight assignments have positive probability: (0,0,0,0) (1,0,0,0) (1,1,0,0) (1,1,1,0) (0,0,0,1) (0,0,1,1) (0,1,1,1) (1,1,1,1)
- For example, $X_1 \perp X_3 | X_2, X_4$:
  - E.g., $P(X_1=0|X_2=0, X_4=0)$
- But distribution doesn't factorize!!!

Markov networks representation Theorem 2 (Hammersley-Clifford Theorem)

If H is an I-map for P and P is a positive distribution	Then	joint probability distribution P: $P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\phi_i(\mathbf{D}_i)$

- Positive distribution and independencies $\Rightarrow$ P factorizes over graph

Representation Theorem for Markov Networks

If joint probability distribution P: $P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\phi_i(\mathbf{D}_i)$	Then	H is an I-map for P

If H is an I-map for P and P is a positive distribution	Then	joint probability distribution P: $P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\phi_i(\mathbf{D}_i)$

Completeness of separation in Markov networks

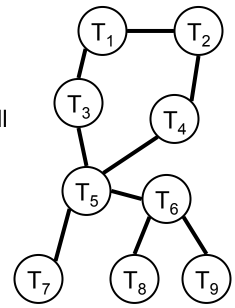
- **Theorem: Completeness of separation**
  - For "almost all" distributions that *P* factorize over Markov network *H*, we have that I(*H*) = I(*P*)
  - *"almost all" distributions*: except for a set of measure zero of parameterizations of the Potentials (assuming no finite set of parameterizations has positive measure)
- **Analogous to BNs**

What are the "local" independence assumptions for a Markov network?

- In a BN *G*:
  - local Markov assumption: variable independent of non-descendants given parents
  - d-separation defines global independence
  - Soundness: For all distributions:
- In a Markov net *H*:
  - **Separation** defines global independencies
  - What are the notions of local independencies?

Local independence assumptions for a Markov network

- **Separation** defines global independencies



- **Pairwise Markov Independence**:
  - Pairs of non-adjacent variables A,B are independent given all others
- **Markov Blanket**:
  - Variable A independent of rest given its neighbors

Equivalence of independencies in Markov networks

- **Soundness Theorem**: For all positive distributions *P*, the following three statements are equivalent:
  - *P* entails the global Markov assumptions
  - *P* entails the pairwise Markov assumptions
  - *P* entails the local Markov assumptions (Markov blanket)

Minimal I-maps and Markov Networks

- A fully connected graph is an I-map
- Remember minimal I-maps?
  - A “simplest” I-map → Deleting an edge makes it no longer an I-map
- In a BN, there is no unique minimal I-map
- Theorem: For positive distributions & **Markov network, minimal I-map is unique!!**
- Many ways to find minimal I-map, e.g.,
  - Take pairwise Markov assumption:
  - If *P* doesn't entail it, add edge:

How about a perfect map?

- Remember perfect maps?
  - independencies in the graph are exactly the same as those in *P*
- For BNs, doesn't always exist
  - counter example: Swinging Couples
- How about for Markov networks?

Unifying properties of BNs and MNs

- BNs:
  - give you: V-structures, CPTs are conditional probabilities, can directly compute probability of full instantiation
  - but: require acyclicity, and thus no perfect map for swinging couples
- MNs:
  - give you: cycles, and perfect maps for swinging couples
  - but: don't have V-structures, cannot interpret potentials as probabilities, requires partition function
- Remember PDAGS???
  - skeleton + immoralities
  - provides a (somewhat) unified representation
  - see book for details

What you need to know so far about Markov networks

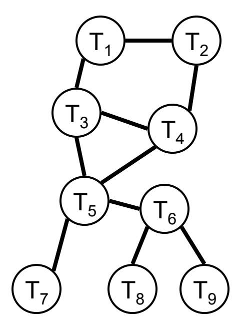
- Markov network representation:
  - undirected graph
  - potentials over cliques (or sub-cliques)
  - normalize to obtain probabilities
  - need partition function
- Representation Theorem for Markov networks
  - if P factorizes, then it's an I-map
  - if P is an I-map, only factorizes for positive distributions
- Independence in Markov nets:
  - active paths and separation
  - pairwise Markov and Markov blanket assumptions
  - equivalence for positive distributions
- Minimal I-maps in MNs are unique
- Perfect maps don't always exist

Some common Markov networks and generalizations

- Pairwise Markov networks
- A very simple application in computer vision
- Logarithmic representation
- Log-linear models
- Factor graphs

Pairwise Markov Networks

- All factors are over single variables or pairs of variables:
  - Node potentials
  - Edge potentials
- Factorization:



- Note that there may be bigger cliques in the graph, but only consider pairwise potentials

A very simple vision application

- Image segmentation: separate foreground from background
- Graph structure:
  - pairwise Markov net
  - grid with one node per pixel
- Node potential:
  - "background color" v. "foreground color"
- Edge potential:
  - neighbors like to be of the same class

Logarithmic representation

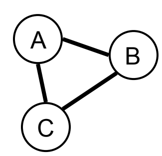
- Standard model: $$P(X_1, \ldots, X_n) = \frac{1}{Z} \prod_{i=1}^{m} \phi_i(\mathbf{D}_i)$$
- Log representation of potential (assuming positive potential):
  - also called the energy function
- Log representation of Markov net:

Log-linear Markov network (most common representation)

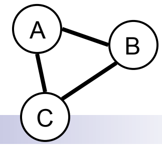
- **Feature** is some function ϕ[**D**] for some subset of variables **D**
  - e.g., indicator function
- **Log-linear model** over a Markov network *H*:
  - a set of features $\phi_1[\mathbf{D}_1], \ldots, \phi_k[\mathbf{D}_k]$
    - each $\mathbf{D}_i$ is a subset of a clique in *H*
    - two ϕ's can be over the same variables
  - a set of weights $w_1, \ldots, w_k$
    - usually learned from data
  - $P(X_1, \ldots, X_n) = \frac{1}{Z} \exp \left[\sum_{i=1}^{k} w_i \phi_i (\mathbf{D}_i) \right]$

Structure in cliques

- Possible potentials for this graph:



Factor graphs



- Very useful for approximate inference
  - Make factor dependency explicit
- Bipartite graph:
  - variable nodes (ovals) for $X_1,\ldots,X_n$
  - factor nodes (squares) for $\phi_1,\ldots,\phi_m$
  - edge $X_i - \phi_j$ if $X_i \in \mathrm{Scope}[\phi_j]$

Summary of types of Markov nets

- Pairwise Markov networks
  - very common
  - potentials over nodes and edges
- Log-linear models
  - log representation of potentials
  - linear coefficients learned from data
  - most common for learning MNs
- Factor graphs
  - explicit representation of factors
    - you know exactly what factors you have
  - very useful for approximate inference