

CS 3750 Machine Learning

Lecture 5

Graphical models

Inference

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

CS 3750 Advanced Machine Learning

Bayesian belief networks (BBNs)

Bayesian belief network:

- Represents the full joint distribution over the variables more compactly with a **smaller number of parameters**.
- Takes advantage of **conditional and marginal independences** among random variables

- **A and B are independent**

$$P(A, B) = P(A)P(B)$$

- **A and B are conditionally independent given C**

$$P(A, B | C) = P(A | C)P(B | C)$$

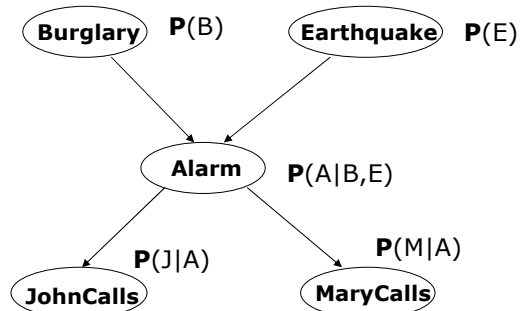
$$P(A | C, B) = P(A | C)$$

CS 3750 Advanced Machine Learning

Bayesian belief network

1. Directed acyclic graph

- **Nodes** = random variables
Burglary, Earthquake, Alarm, Mary calls and John calls
- **Links** = direct (causal) dependencies between variables.
The chance of Alarm being is influenced by Earthquake,
The chance of John calling is affected by the Alarm

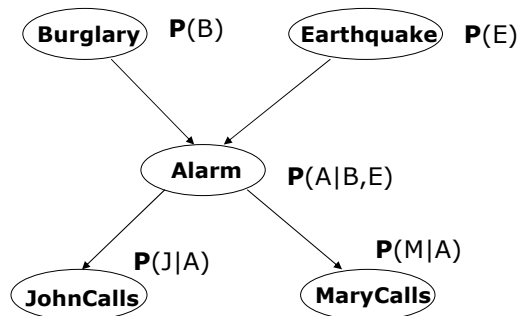


CS 3750 Advanced Machine Learning

Bayesian belief network

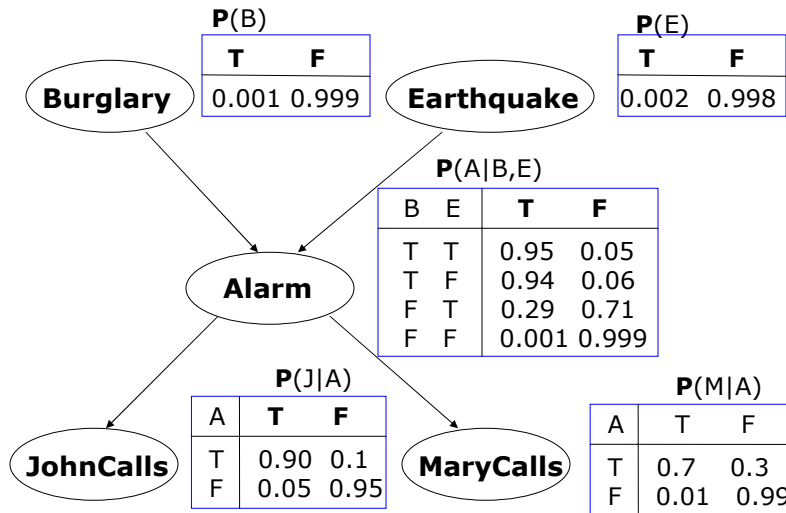
2. Local conditional distributions

- relate variables and their parents



CS 3750 Advanced Machine Learning

Bayesian belief network



CS 3750 Advanced Machine Learning

Full joint distribution in BBNs

Full joint distribution is defined in terms of local conditional distributions (obtained via the chain rule):

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} P(X_i \mid pa(X_i))$$

Example:

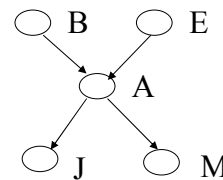
Assume the following assignment of values to random variables

$$B = T, E = T, A = T, J = T, M = F$$

Then its probability is:

$$P(B = T, E = T, A = T, J = T, M = F) =$$

$$P(B = T)P(E = T)P(A = T \mid B = T, E = T)P(J = T \mid A = T)P(M = F \mid A = T)$$



CS 3750 Advanced Machine Learning

Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?**

Alarm example: 5 binary (True, False) variables

of parameters of the full joint:

$$2^5 = 32$$

One parameter is for free:

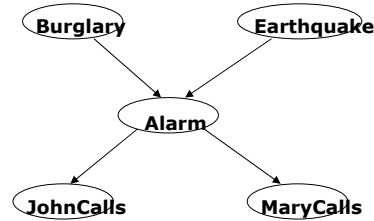
$$2^5 - 1 = 31$$

of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$

One parameter in every conditional is for free:

$$2^2 + 2(2) + 2(1) = 10$$



CS 3750 Advanced Machine Learning

Model acquisition problem

The structure of the BBN typically reflects causal relations

- BBNs are also sometime referred to as **causal networks**
- Causal structure is very intuitive in many applications domain and it is relatively easy to obtain from the domain expert

Probability parameters of BBN correspond to conditional distributions relating a random variable and its parents only

- Their complexity much smaller than the full joint
- Easier to come up (estimate) the probabilities from expert or automatically by learning from data

CS 3750 Advanced Machine Learning

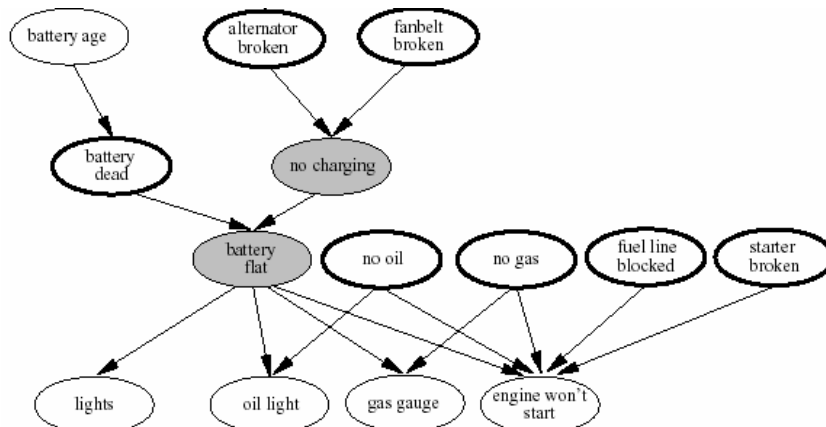
BBNs built in practice

- **In various areas:**
 - Intelligent user interfaces (Microsoft)
 - Troubleshooting, diagnosis of a technical device
 - Medical diagnosis:
 - Pathfinder (Intellipath)
 - CPSC
 - Munin
 - QMR-DT
 - Collaborative filtering
 - Military applications
 - Insurance, credit applications

CS 3750 Advanced Machine Learning

Diagnosis of car engine

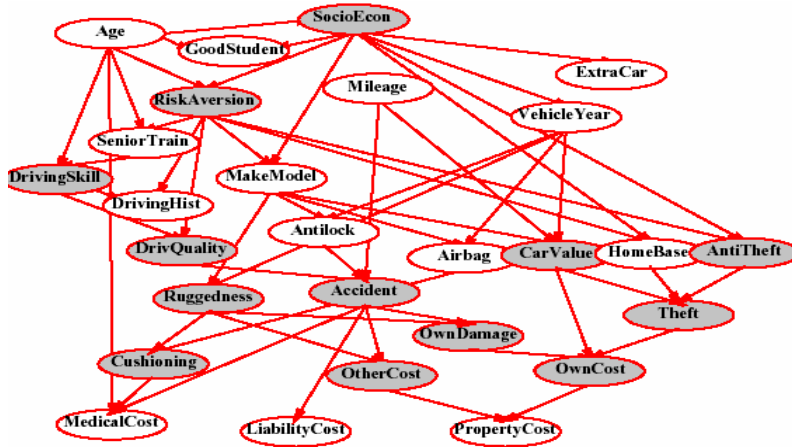
- Diagnose the engine start problem



CS 3750 Advanced Machine Learning

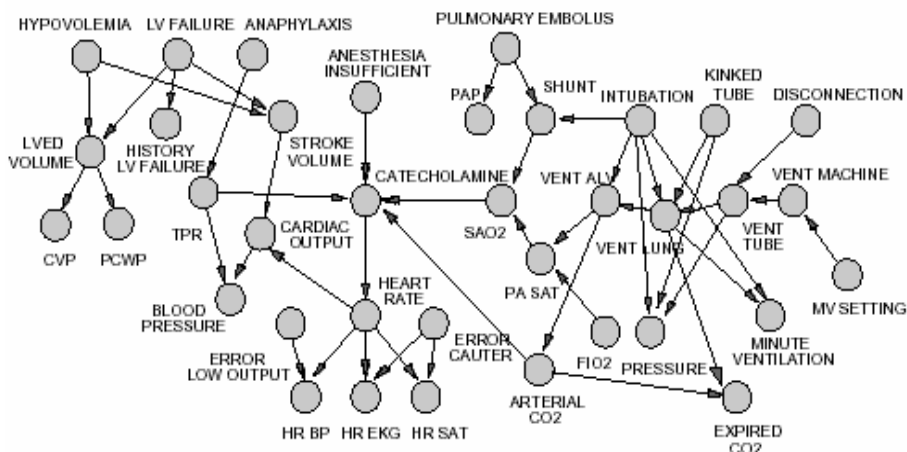
Car insurance example

- Predict claim costs (medical, liability) based on application data



CS 3750 Advanced Machine Learning

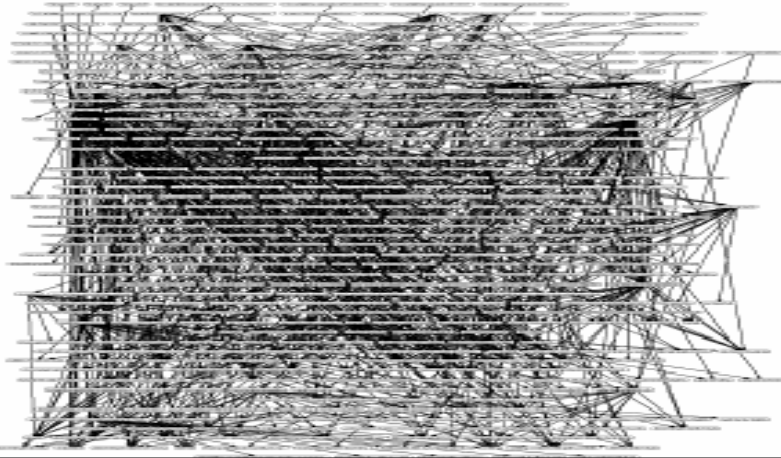
(ICU) Alarm network



CS 3750 Advanced Machine Learning

CPCS

- Computer-based Patient Case Simulation system (CPCS-PM) developed by Parker and Miller (at University of Pittsburgh)
- 422 nodes and 867 arcs

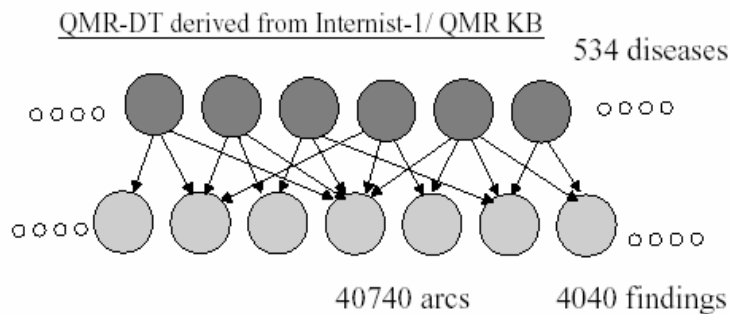


CS 3750 Advanced Machine Learning

QMR-DT

- **Medical diagnosis in internal medicine**

Bipartite network of disease/findings relations



CS 3750 Advanced Machine Learning

Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
 - Smaller number of parameters
- But we are interested in solving various **inference tasks**:
 - **Diagnostic task. (from effect to cause)**

$$P(\text{Burglary} \mid \text{JohnCalls} = T)$$

- **Prediction task. (from cause to effect)**

$$P(\text{JohnCalls} \mid \text{Burglary} = T)$$

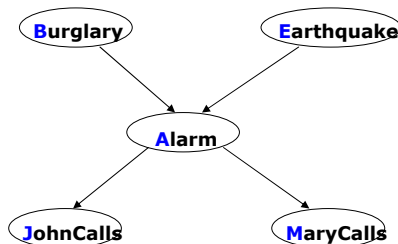
- **Other probabilistic queries** (queries on joint distributions).

$$P(\text{Alarm})$$

- **Question:** Can we take advantage of independences to construct special algorithms and speedup the inference?

Inference in Bayesian network

- **Bad news:**
 - Exact inference problem in BBNs is NP-hard (Cooper)
 - Approximate inference is NP-hard (Dagum, Luby)
- **But** very often we can achieve significant improvements
- Assume our Alarm network



- Assume we want to compute: $P(J = T)$

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$\begin{aligned}P(J = T) &= \\&= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m) \\&= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)\end{aligned}$$

Computational cost:

Number of additions: ?

Number of products: ?

CS 3750 Advanced Machine Learning

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$\begin{aligned}P(J = T) &= \\&= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m) \\&= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)\end{aligned}$$

Computational cost:

Number of additions: 15

Number of products: ?

CS 3750 Advanced Machine Learning

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$\begin{aligned}
 P(J = T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
 \end{aligned}$$

Computational cost:

Number of additions: 15

Number of products: $16 * 4 = 64$

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$\begin{aligned}
 P(J = T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \right] \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]
 \end{aligned}$$

Computational cost:

Number of additions: $1 + 2 * [1 + 1 + 2 * 1] = ?$

Number of products: $2 * [2 + 2 * (1 + 2 * 1)] = ?$

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$\begin{aligned}
 P(J=T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J=T | A=a) P(M=m | A=a) P(A=a | B=b, E=e) P(B=b) P(E=e) \\
 &= \sum_{b \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J=T | A=a) P(M=m | A=a) P(B=b) \left[\sum_{e \in T, F} P(A=a | B=b, E=e) P(E=e) \right] \\
 &= \sum_{a \in T, F} P(J=T | A=a) \left[\sum_{m \in T, F} P(M=m | A=a) \right] \left[\sum_{b \in T, F} P(B=b) \right] \left[\sum_{e \in T, F} P(A=a | B=b, E=e) P(E=e) \right]
 \end{aligned}$$

Computational cost:

Number of additions: $1+2*[1+1+2*1]=9$

Number of products: $2*[2+2*(1+2*1)]=?$

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way (multiplications by constants can be taken out of the sum)

$$\begin{aligned}
 P(J=T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J=T | A=a) P(M=m | A=a) P(A=a | B=b, E=e) P(B=b) P(E=e) \\
 &= \sum_{b \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J=T | A=a) P(M=m | A=a) P(B=b) \left[\sum_{e \in T, F} P(A=a | B=b, E=e) P(E=e) \right] \\
 &= \sum_{a \in T, F} P(J=T | A=a) \left[\sum_{m \in T, F} P(M=m | A=a) \right] \left[\sum_{b \in T, F} P(B=b) \right] \left[\sum_{e \in T, F} P(A=a | B=b, E=e) P(E=e) \right]
 \end{aligned}$$

Computational cost:

Number of additions: $1+2*[1+1+2*1]=9$

Number of products: $2*[2+2*(1+2*1)]=16$

Inference in Bayesian network

- **Exact inference algorithms:**
 - **Variable elimination**
 - Recursive decomposition (Cooper, Darwiche)
 - Symbolic inference (D'Ambrosio)
 - Belief propagation algorithm (Pearl)
 - Arc reversal (Olmsted, Schachter)
- **Approximate inference algorithms:**
 - **Monte Carlo methods:**
 - Forward sampling, Likelihood sampling
 - **Variational methods**

CS 3750 Advanced Machine Learning

Variable elimination

- **Idea:** interleave sum and products one variable at the time during the inference
 - Typically relies on a special structure (called **joint tree**) that groups together multiple variables
 - E.g. Query $P(J = T)$ requires to eliminate A,B,E,M and this can be done in different order

$$P(J = T) =$$
$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)$$

CS 3750 Advanced Machine Learning

Variable elimination

Assume order: M, E, B, A to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \quad 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \tau_1(A = a, B = b) \\
 &= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{b \in T, F} P(B = b) \tau_1(A = a, B = b) \right] \\
 &= \sum_{a \in T, F} P(J = T | A = a) \tau_2(A = a)
 \end{aligned}$$

CS 3750 Advanced Machine Learning

Factors

- **Factor:** is a function that maps value assignments for a subset of random variables to \mathfrak{R} (reals)
- **The scope of the factor:**
 - a set of variables defining the factor
- **Example:**
 - Assume discrete random variables x (with values a_1, a_2, a_3) and y (with values b_1 and b_2)
 - Factor:

$\phi(x, y) \longrightarrow$

a_1	b_1	0.5
a_1	b_2	0.2
a_2	b_1	0.1
a_2	b_2	0.3
a_3	b_1	0.2
a_3	b_2	0.4

- Scope of the factor:

$\{x, y\}$

CS 3750 Advanced Machine Learning

Factor Product

Variables: A,B,C

$$\phi(A, B, C) = \phi(B, C) \circ \phi(A, B)$$

$\phi(B, C)$

b1	c1	0.1
b1	c2	0.6
b2	c1	0.3
b2	c2	0.4

$\phi(A, B)$

a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

$\phi(A, B, C)$

a1	b1	c1	0.5*0.1
a1	b1	c2	0.5*0.6
a1	b2	c1	0.2*0.3
a1	b2	c2	0.2*0.4
a2	b1	c1	0.1*0.1
a2	b1	c2	0.1*0.6
a2	b2	c1	0.3*0.3
a2	b2	c2	0.3*0.4
a3	b1	c1	0.2*0.1
a3	b1	c2	0.2*0.6
a3	b2	c1	0.4*0.3
a3	b2	c2	0.4*0.4

Factor Marginalization

Variables: A,B,C

$$\phi(A, C) = \sum_B \phi(A, B, C)$$

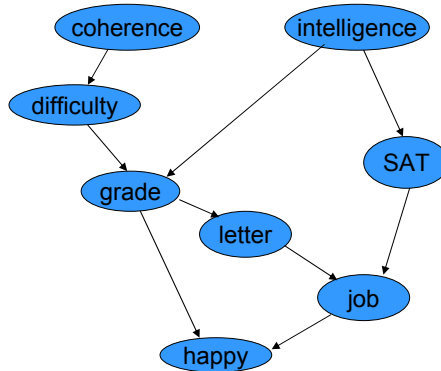
a1	b1	c1	0.2
a1	b1	c2	0.35
a1	b2	c1	0.4
a1	b2	c2	0.15
a2	b1	c1	0.5
a2	b1	c2	0.1
a2	b2	c1	0.3
a2	b2	c2	0.2
a3	b1	c1	0.25
a3	b1	c2	0.45
a3	b2	c1	0.15
a3	b2	c2	0.25

a1	c1	0.2+0.4=0.6
a1	c2	0.35+0.15=0.5
a2	c1	0.8
a2	c2	0.3
a3	c1	0.4
a3	c2	0.7

Variable elimination

The order in which variables are eliminated may effect the efficiency of the variable elimination process

Assume the following BBN and calculation of $P(\text{Job})$:

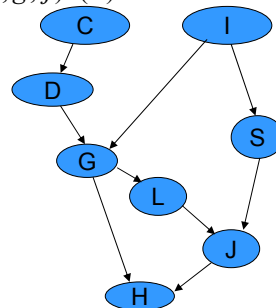


CS 3750 Advanced Machine Learning

Variable elimination

Calculations performed in terms of factors:

$$\begin{aligned}
 p(j) &= \sum_{L,S,G,H,I,D,C} \phi(c)\phi(i)\phi(d,c)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j) \\
 &= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j) \sum_C \phi(c)\phi(d,c) \\
 &= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j) \tau(d) \\
 &\dots \\
 &= \sum_{L,S} \phi(j,l,s) \sum_G \phi(l,g) \tau(s,g) \tau(g,j) \\
 &= \sum_{L,S} \phi(j,l,s) \tau(l,s,j) \\
 &= \sum_L \tau(l,j) \\
 &= \tau(j)
 \end{aligned}$$

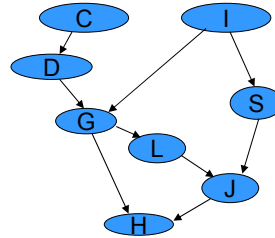


CS 3750 Advanced Machine Learning

Variable elimination

Trace 1:

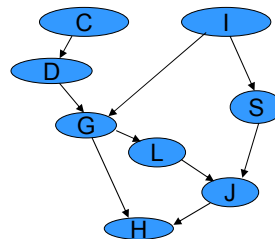
Step	Var	Factors Used	New Factor
1	C	$\phi_c(C), \phi_D(D, C)$	$\tau_1(D)$
2	D	$\phi_G(G, I, D), \tau_1(D)$	$\tau_2(G, I)$
3	I	$\phi_I(I), \phi_S(S, I), \tau_2(G, I)$	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	$\tau_4(G, J)$
5	G	$\tau_4(G, J), \tau_3(G, S), \phi_L(L, G)$	$\tau_5(J, L, S)$
6	S	$\tau_5(J, L, S), \phi_J(J, L, S)$	$\tau_6(J, L)$
7	L	$\tau_6(J, L)$	$\tau_7(J)$



Variable elimination

Trace 1:

Step	Var	Factors Used	New Factor
1	C	$\phi_c(C), \phi_D(D, C)$	$\tau_1(D)$
2	D	$\phi_G(G, I, D), \tau_1(D)$	$\tau_2(G, I)$
3	I	$\phi_I(I), \phi_S(S, I), \tau_2(G, I)$	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	$\tau_4(G, J)$
5	G	$\tau_4(G, J), \tau_3(G, S), \phi_L(L, G)$	$\tau_5(J, L, S)$
6	S	$\tau_5(J, L, S), \phi_J(J, L, S)$	$\tau_6(J, L)$
7	L	$\tau_6(J, L)$	$\tau_7(J)$

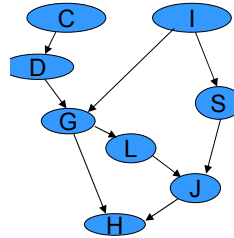


Complexity: 4 variables used – 1 summed away

Variable elimination

Trace 2:

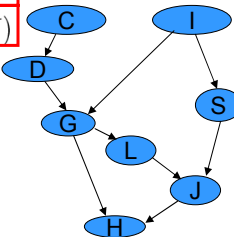
Step	Var	Factors Used	New Factor
1	G	$\phi_G(G, I, D), \phi_L(L, G)\phi_H(H, G, J)$	$\tau_1(I, D, L, J, H)$
2	I	$\phi_I(I), \phi_S(S, I)\tau_1(I, D, L, J, H)$	$\tau_2(D, L, S, J, H)$
3	S	$\phi_J(J, L, S), \tau_2(D, L, S, J, H)$	$\tau_3(D, L, J, H)$
4	L	$\tau_3(D, L, J, H)$	$\tau_4(D, J, H)$
5	H	$\tau_4(D, J, H)$	$\tau_5(D, J)$
6	C	$\tau_5(D, J), \phi_D(D, C)$	$\tau_6(D, J)$
7	D	$\tau_6(D, J)$	$\tau_7(J)$



Variable elimination

Trace 2:

Step	Var	Factors Used	New Factor
1	G	$\phi_G(G, I, D), \phi_L(L, G)\phi_H(H, G, J)$	$\tau_1(I, D, L, J, H)$
2	I	$\phi_I(I), \phi_S(S, I)\tau_1(I, D, L, J, H)$	$\tau_2(D, L, S, J, H)$
3	S	$\phi_J(J, L, S), \tau_2(D, L, S, J, H)$	$\tau_3(D, L, J, H)$
4	L	$\tau_3(D, L, J, H)$	$\tau_4(D, J, H)$
5	H	$\tau_4(D, J, H)$	$\tau_5(D, J)$
6	C	$\tau_5(D, J), \phi_D(D, C)$	$\tau_6(D, J)$
7	D	$\tau_6(D, J)$	$\tau_7(J)$



Complexity: 6 variables used – 1 summed out

Markov random fields

- **Probabilistic models with symmetric dependences.**
 - Typically models spatially varying quantities

$$P(x) \propto \prod_{c \in cl(x)} f_c(x_c)$$

$f_c(x_c)$ - A potential function (defined over factors)

$$P(x) = \frac{1}{Z} \exp\left(-\sum_{c \in cl(x)} \phi_c(x_c)\right)$$

- Gibbs (Boltzman) distribution

$$Z = \sum_{x \in \{x\}} \exp\left(-\sum_{c \in cl(x)} \phi_c(x_c)\right) \quad \text{- A partition function}$$

CS 3750 Advanced Machine Learning

Graphical representation of MRFs

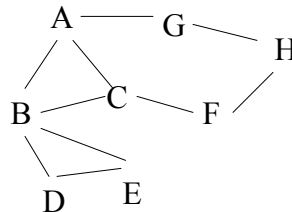
An undirected network (also called independence graph)

- $G = (S, E)$
 - $S = 1, 2, \dots, N$ correspond to random variables
 - $(i, j) \in E \Leftrightarrow \exists c : \{i, j\} \subset c$
or x_i and x_j appear within the same factor c

Example:

- variables A, B ..H
- Assume the full joint of MRF

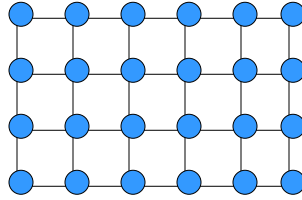
$$P(A, B, \dots, H) = \phi_1(A, B, C) \phi_2(B, D, E) \phi_3(A, G) \phi_4(C, F) \phi_5(G, H) \phi_6(F, H)$$



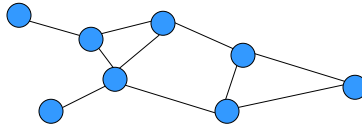
CS 3750 Advanced Machine Learning

Markov random fields

- regular lattice
(Ising model)



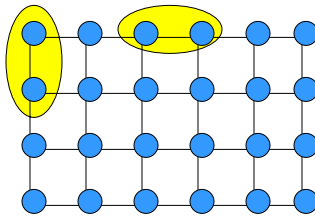
- Arbitrary graph



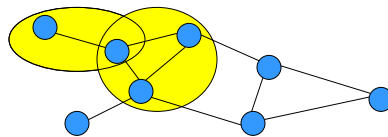
CS 3750 Advanced Machine Learning

Markov random fields

- regular lattice
(Ising model)



- Arbitrary graph



CS 3750 Advanced Machine Learning