Markov Networks

www.biostat.wisc.edu/~dpage/cs760/

Goals for the lecture

you should understand the following concepts

- Markov network syntax
- Markov network semantics
- Potential functions
- Partition function
- MN parameter learning by gradient ascent

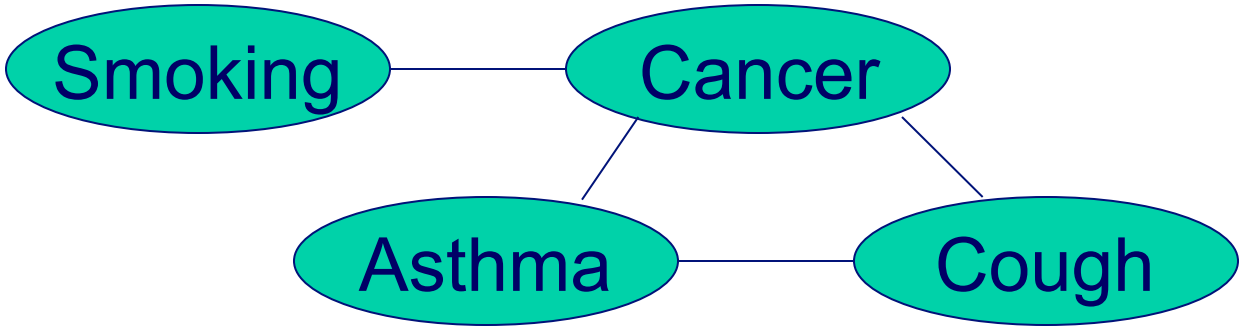
Goals for the lecture

you should understand the following concepts

- Markov network syntax
- Markov network semantics
- Potential functions
- Partition function
- MN parameter learning by gradient ascent (algorithm you should know, as presented for log-linear space)

Markov Networks

- **Undirected** graphical models



- Potential functions defined over cliques

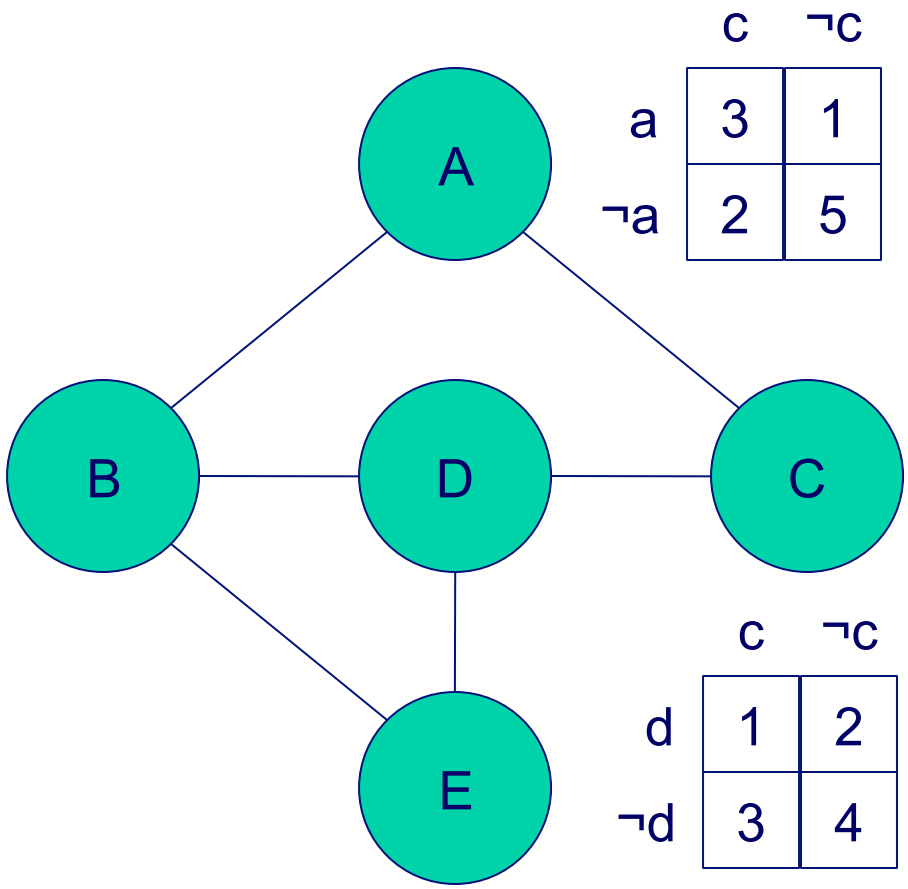
$$P(x) = \frac{1}{Z}\prod_c \Phi_c(x_c)$$

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

Smoking	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	2.7
True	True	4.5

More on Potentials

- Values are typically non-negative
- Values need not be probabilities
- Generally, one table associated with each clique



Calculating the Full Joint Probability Density

- Full Joint Probability Density is the normalized product of the event probabilities

$$P(x) = \frac{1}{Z}\prod_{c}\Phi_{c}(x_{c})$$

Normalization constant

One potential

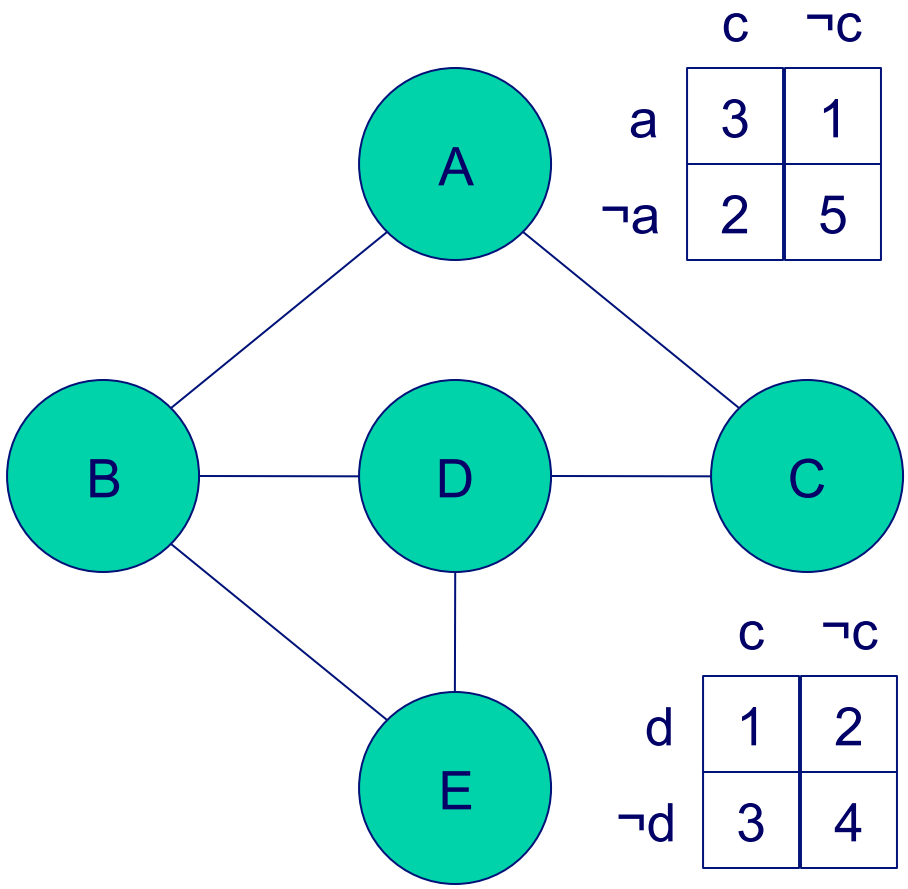
Feature vector (i.e. $\langle A, B, C, D, E \rangle$)

Calculating the Normalization Constant Z

$$Z = \sum_{x} \prod_{c} \Phi_c(x_c)$$

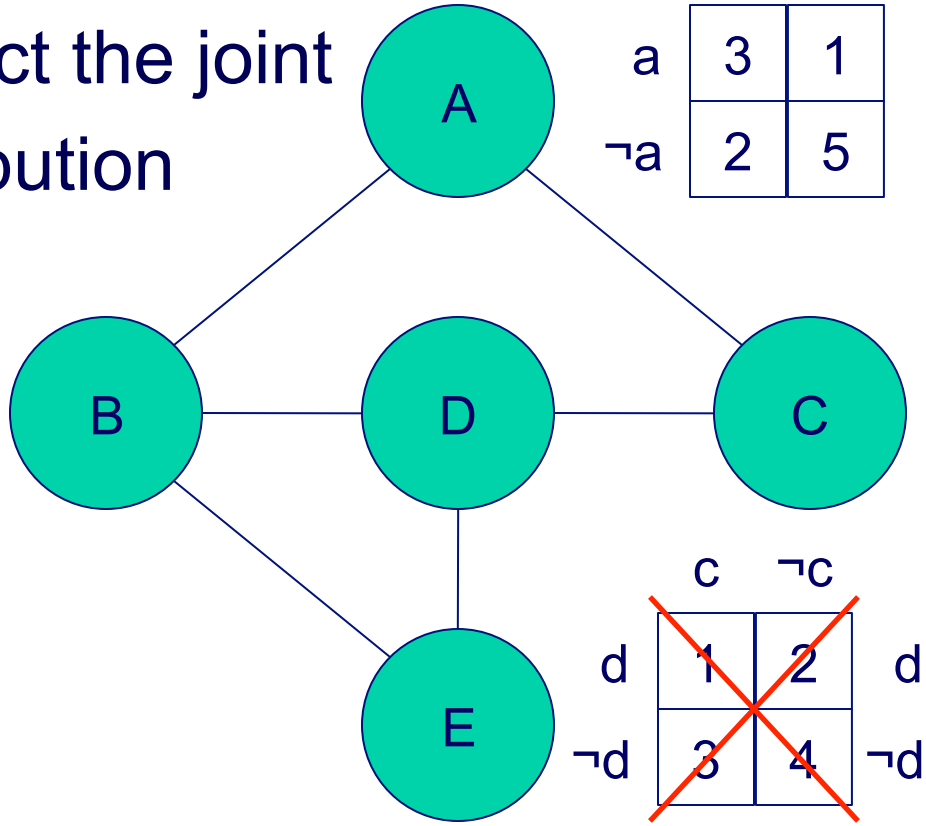
$$Z = \sum_{x} \prod_{c} \Phi_c(x_c)$$

- Get probability of A=1, B=0, C=1, D=0, E=0
- Only need potentials
- Multiply entries consistent with this setting: 3 x 3 = 9
- Normalize



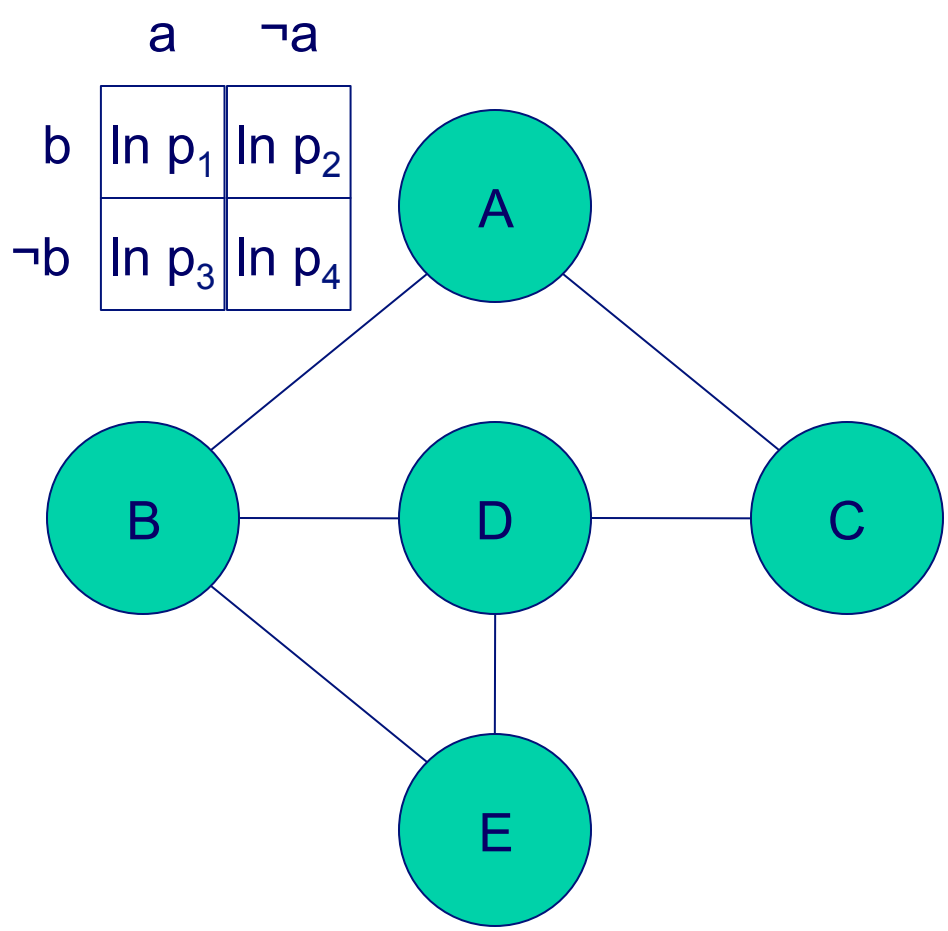
Scale Invariance

The change at the bottom right will not affect the joint probability distribution



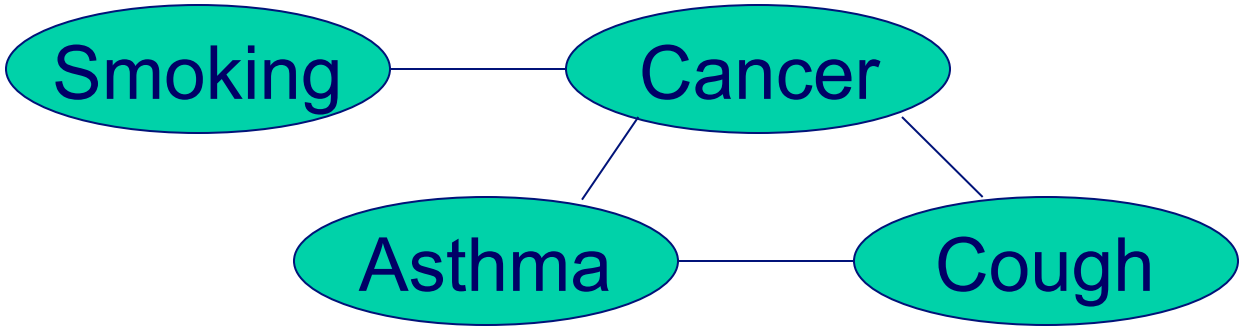
Log Linear Models

- Equivalent to Markov Nets (though they look different)
- Take the natural log of each parameter



Markov Networks

- **Undirected** graphical models



- Log-linear model:

$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(x) \right)$$

Weight of Feature *i*

Feature *i*

$$f_1(\text{Smoking, Cancer}) = \begin{cases} 1 & \text{if } \neg \text{ Smoking } \vee \text{ Cancer} \\ 0 & \text{otherwise} \end{cases}$$

$$w_1 = 1.5$$

Markov nets vs. Bayes nets

Property	Markov nets	Bayes nets
Graph	undirected	directed
Distribution form	product of potentials	product of potentials
Potentials	arbitrary	conditional probabilities
Cycles	allowed	forbidden
Partition function	$Z = ?$	$Z = 1$

Inference: Markov Chain Monte Carlo (MCMC)

- General algorithm: **Metropolis-Hastings**
- Simplest (and most popular) algorithm: **Gibbs sampling**
  - Sample one variable at a time given the rest

$$P(x \mid MB(x)) = \frac{\exp\left(\sum_i w_i f_i(x)\right)}{\exp\left(\sum_i w_i f_i(x=0)\right) + \exp\left(\sum_i w_i f_i(x=1)\right)}$$

Gibbs Sampling

state ← random truth assignment
for *i* ← 1 **to** *num-samples* **do**
    for each variable *x*
        sample *x* according to P(*x*|*neighbors*(*x*))
        state ← *state* with new value of *x*
P(*F*) ← fraction of states in which *F* is true

Parameter Learning: Recall the Bayes Net approach

- In Bayes Nets, we go through each variable one at a time, row by row in the CPT adjusting weights

- One way to think of this approach is that we look at the prior setting and ask what the probability of this setting is based on what we see in the data, then adjust the CPT to be consistent with the data

Can we use this approach on Markov Nets?

- No! Consider changing a single table value.
  - This change the partition function, Z.
  - Thus, a local change to one table effects other tables; local changes have global effects!

Markov Net Learning

- We want to get the derivative of the likelihood function with respect to weights or cells of potentials. Then move each weight in direction of the gradient based on learning parameter η

- In log space the above approach amounts to taking difference in expectations (given current parameters) and observed values (in data set), computed as on the next slide

Weight Learning

- Maximize likelihood or posterior probability
- Numerical optimization (gradient or 2nd order)
- No local maxima

$$\frac{\partial}{\partial w_i} \log P_w(x) = n_i(x) - E_w[n_i(x)]$$

No. of times feature *i* is true in data

Expected no. times feature *i* is true according to model

- Requires inference at each step (slow!)

Comments on Markov network learning

- Markov nets (a.k.a., Markov random fields, MNs, MRFs) are an undirected variation on Bayes nets
- Some things stay roughly the same as BNs, most notably *inference*
- Some things get easier than for BNs: d-separation, Markov blanket, determining directions on arcs in a learned structure (don't have to)
- Some things got much harder: determining the probability of a particular example (complete setting of variables), learning parameters
- Widely used variations: conditional random fields (CRFs), Markov logic networks (MLNs) for statistical relational learning, next…