

Undirected Probabilistic Graphical Models

CS 760: Machine Learning
Spring 2018

Mark Craven and David Page

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

Goals for the lecture

you should understand the following concepts

- Markov networks
- Markov network syntax
- Markov network semantics
- Differences between Bayes nets and Markov nets
- Potential functions
- Partition function
- Loglinear formulation of MNs
- MN parameter learning by gradient ascent
- Markov chain Monte Carlo (MCMC) sampling
- Metropolis-Hastings
- Gibbs sampling
- Persistent contrastive divergence (PCD)
- Pseudo-likelihood
- Screening rules
- Graphical Lasso

Markov Networks

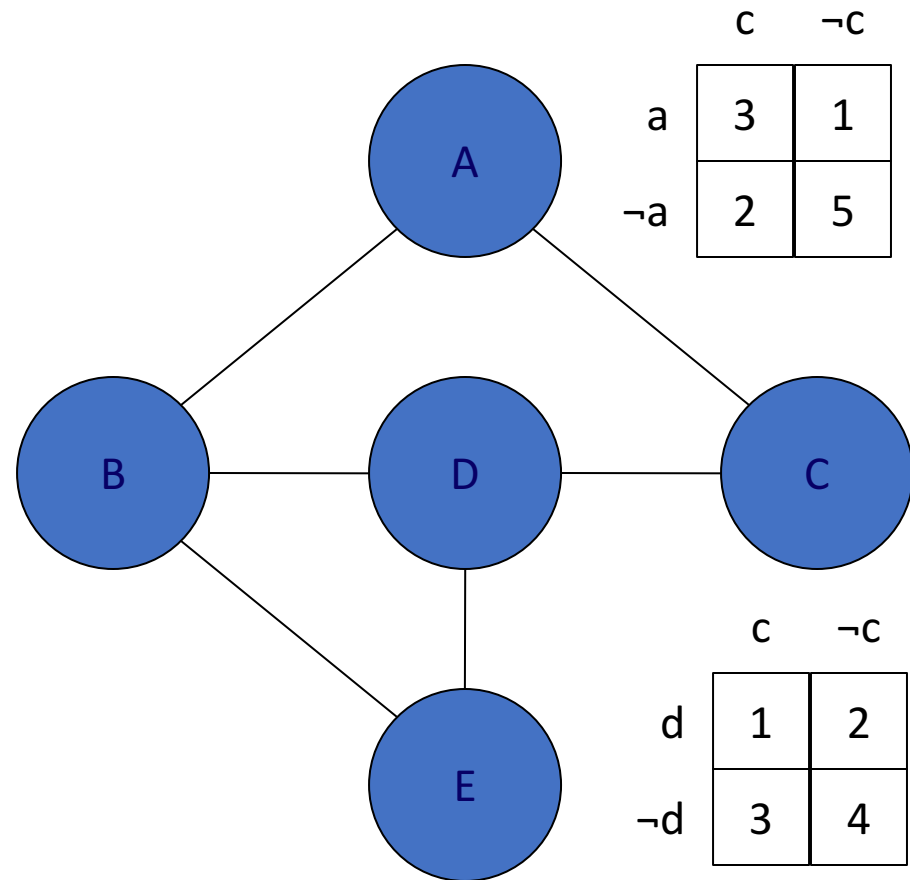
- Like Bayes Nets
 - Graphical model that describes joint probability distribution using tables (AKA potentials)
 - Nodes are random variables
 - Labels are outcomes over the variables

Markov Networks

- Unlike Bayes Nets
 - Undirected graph
 - No requirement that tables need not be conditional distributions
 - Table distributed over complete subgraph

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

$$P(\vec{V}) = \frac{1}{Z} \prod_k \phi_k(\vec{V})$$

Normalization
constant

One potential

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

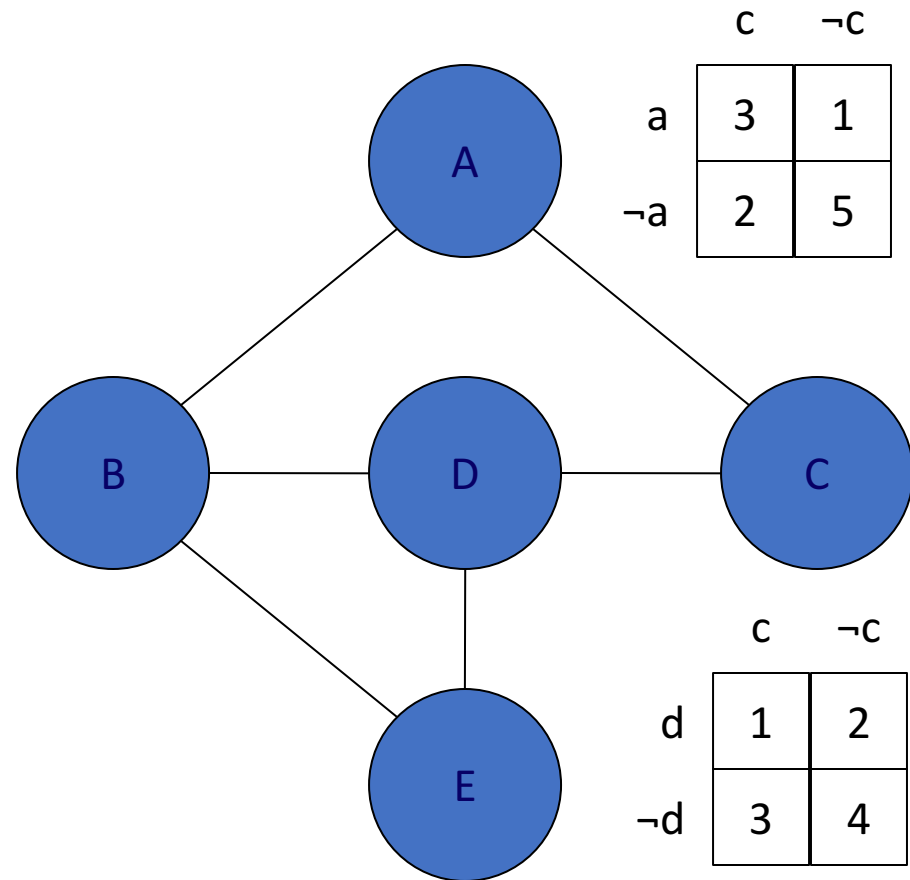
Calculating the Normalization Constant Z

$$Z = \sum_{\vec{v} \in \vec{V}} \prod_k \phi_k(\vec{v})$$

Using

$$P(\vec{V}) = \frac{1}{Z} \prod_k \phi_k(\vec{V})$$

- 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)



Hammersley-Clifford Theorem

- If Distribution is strictly positive ($P(x) > 0$)
- And Graph encodes conditional independences
- Then Distribution is product of potentials over cliques of graph
- Inverse is also true
- (“Markov network = Gibbs distribution”)

Markov Nets versus Bayes Nets

- Disadvantages of Markov Nets
 - Computationally intensive to compute probability of any complete setting of variables with Markov Net (NP-hard), easy for Bayes Net
 - Hard to learn Markov Net parameters in a straightforward way
 - Can't just use marginal frequencies from data as for Bayes nets
 - Gradient ascent requires inference (hard)

Markov Nets versus Bayes Nets

- Advantages of Markov Nets
 - Easier to reason about conditional independence
 - Markov Blanket is just set of neighbors
 - d-separation: conditional independence achieved iff all paths cut off by evidence
 - No need to select an arbitrary, potentially misleading direction for a dependency in cases where the direction is unclear
 - Learn structure just by learning parameters

Markov Nets vs. Bayes Nets

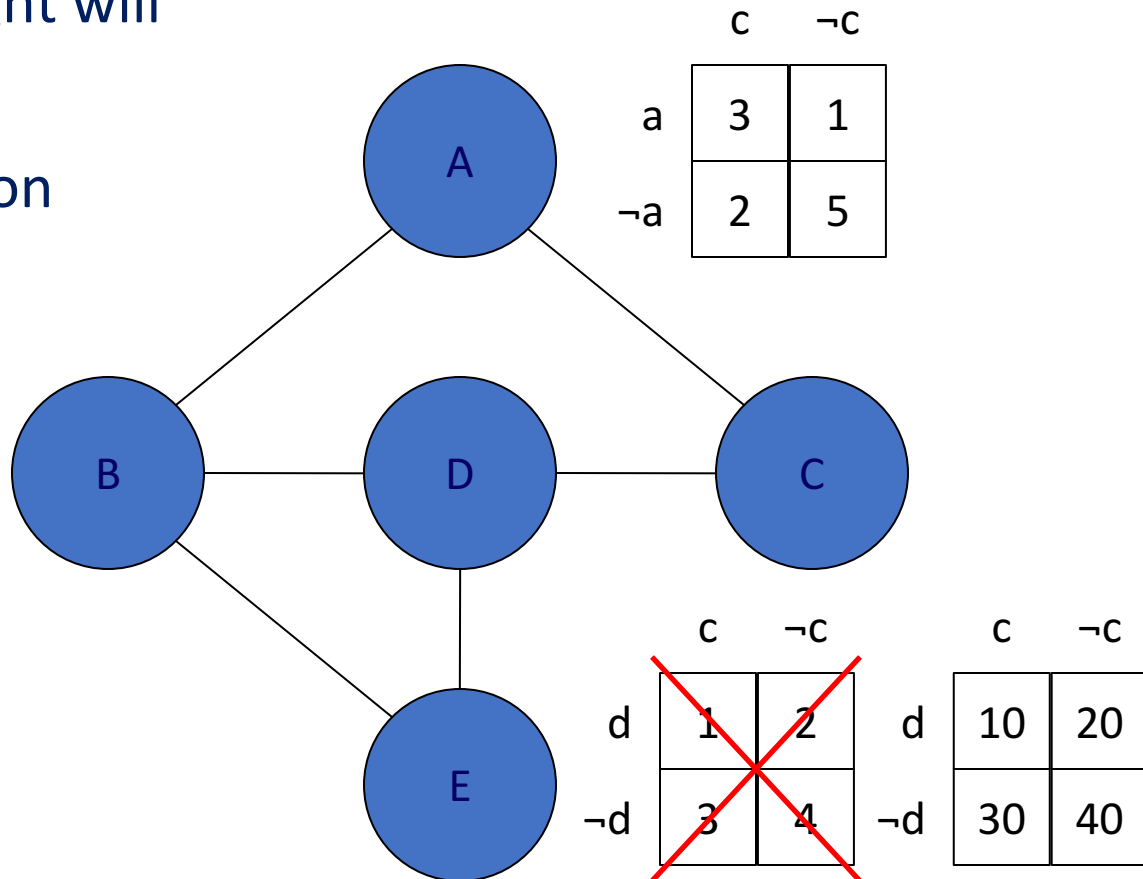
Property	Markov Nets	Bayes Nets
Form	Prod. potentials	Prod. potentials
Potentials	Arbitrary	Cond. probabilities
Cycles	Allowed	Forbidden
Partition func.	$Z = ?$	$Z = 1$
Indep. check	Graph separation	D-separation
Indep. props.	Some	Some
Inference	MCMC, BP, etc.	Convert to Markov

Constructing Markov Nets

- Just as in Bayes Nets, the decision of which tables to represent is based on background knowledge
- Although the model can be built from the data, it is often easier for people to leverage domain knowledge
- Although the model is undirected, it can still be helpful to think of directionality when constructing the Markov Net

Scale Invariance

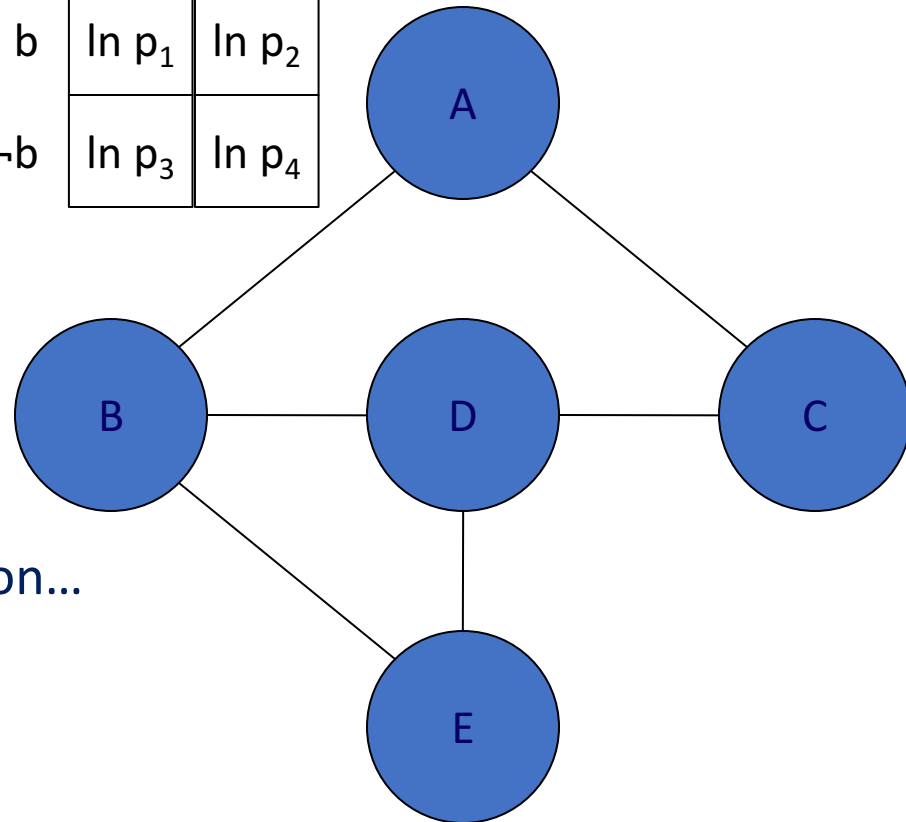
The change at the right will
not effect the joint
probability distribution



Log Linear Models

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

	a	$\neg a$
b	$\ln p_1$	$\ln p_2$
$\neg b$	$\ln p_3$	$\ln p_4$



- Not scale-invariant to this change
- So must change definition of distribution...

Log Linear Models

- This change allows us to write the joint probability distribution or density as:

$$\Pr(\vec{V}) = \frac{1}{Z} \exp \sum_i w_i f_i(\vec{V})$$

$\exp(X) = e^X$

In potential values

Logical statements, either 1 or 0
Also known as *features* or *formulae*
For example,
 $f_1 = a \wedge b$
 $f_2 = \neg a \wedge b$

Inference

- Almost the same as in Bayes Nets (this is somewhat surprising considering all the other differences!)
- Possible approaches:
 - Gibbs sampling
 - Variable elimination
 - Belief propagation

Inference in Markov Networks

- Goal: Compute marginals & conditionals of

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

- Conditioning on Markov blanket of a proposition x is easy, because you only have to consider cliques (features) that involve x
- Gibbs sampling exploits this

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

Markov Chain Monte Carlo (MCMC)

- General algorithm: Metropolis-Hastings
 - Sample next state given current one according to transition probability
 - Reject new state with some probability to maintain detailed balance
- Simplest (and most popular) algorithm: Gibbs sampling
 - Sample one variable at a time given the rest
 - Requires that no settings have probability 0

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

MCMC: Gibbs Sampling

```
state  $\leftarrow$  random truth assignment  
for  $i \leftarrow 1$  to num-samples do  
  for each variable  $x$   
    sample  $x$  according to  $P(x | \textit{neighbors}(x))$   
    state  $\leftarrow$  state with new value of  $x$   
 $P(F) \leftarrow$  fraction of states in which  $F$  is true
```


Markov Chains: A 10-Slide Theoretical Detour

- A Markov chain includes
 - A set of states
 - A set of associated transition probabilities
 - For every pair of states s and s' (not necessarily distinct) we have an associated transition probability $T(s \rightarrow s')$ of moving from state s to state s'
 - For any time t , $T(s \rightarrow s')$ is the probability of the Markov process being in state s' at time $t+1$ given that it is in state s at time t

Some Properties of Markov Chains

- **Irreducible** chain: can get from any state to any other eventually (non-zero probability)
- **Periodic** state: state i is periodic with period k if all returns to i must occur in multiples of k
- **Ergodic** chain: irreducible and has an aperiodic state. Implies all states are aperiodic, so chain is aperiodic.
- Finite state space: can represent chain as matrix of transition probabilities... then *ergodic* = *regular*...
- **Regular** chain: some power of chain (transition matrix) has only positive elements
- **Reversible** chain: satisfies detailed balance (**later**)

Sufficient Condition for Regularity

- A Markov chain is regular if the following properties both hold:
 1. For any pair of states s, s' that each have nonzero probability there exists some path from s to s' with nonzero probability
 2. For all s with nonzero probability, the “self loop” probability $T(s \rightarrow s)$ is nonzero
- Gibbs sampling is regular if no zeroes in CPTs

Notation: Probabilities

- $\pi_t(\mathbf{y})$ = probability of being in state \mathbf{y} at time t
- Transition function $T(\mathbf{y} \rightarrow \mathbf{y}')$ = probability of moving from state \mathbf{y} to state \mathbf{y}'

How π Changes with Time in a Markov Chain

- $\pi_{t+1}(\mathbf{y}') = \sum_{\mathbf{y}} \pi_t(\mathbf{y}) T(\mathbf{y} \rightarrow \mathbf{y}')$
- A distribution π_t is stationary if $\pi_t = \pi_{t+1}$, that is, for all \mathbf{y} , $\pi_t(\mathbf{y}) = \pi_{t+1}(\mathbf{y})$

Detailed Balance

- A Markov chain satisfies detailed balance if there exists a unique distribution π such that for all states \mathbf{y}, \mathbf{y}' ,

$$\pi(\mathbf{y})T(\mathbf{y} \rightarrow \mathbf{y}') = \pi(\mathbf{y}')T(\mathbf{y}' \rightarrow \mathbf{y})$$

- If a regular Markov chain satisfies detailed balance with distribution π , then there exists t such that for any initial distribution π_0 , $\pi_t = \pi$
- Detailed balance with regularity implies convergence to unique stationary distribution

Gibbs Sampler satisfies Detailed Balance

A Gibbs sampler Markov chain defined by a Bayesian network with all CPT entries nonzero, representing probability distribution P , satisfies detailed balance with probability distribution $\pi(\mathbf{y})=P(\mathbf{y})$ for all states \mathbf{y}

Is special case of Metropolis-Hastings Algorithm, next

Using Other Samplers

- The Gibbs sampler only changes one random variable at a time
 - Slow convergence
 - High-probability states may not be reached because reaching them requires going through low-probability states
 - If zeroes in some CPTs, may fail to achieve detailed balance

Metropolis Sampler

- Propose a transition with probability $T^Q(\mathbf{y} \rightarrow \mathbf{y}')$
- Accept with probability $A = \min(1, P(\mathbf{y}')/P(\mathbf{y}))$
- If for all \mathbf{y}, \mathbf{y}' $T^Q(\mathbf{y} \rightarrow \mathbf{y}') = T^Q(\mathbf{y}' \rightarrow \mathbf{y})$ then the resulting Markov chain satisfies detailed balance

Metropolis-Hastings Sampler

- Propose a transition with probability $T^Q(\mathbf{y} \rightarrow \mathbf{y}')$
- Accept with probability
$$A = \min(1, P(\mathbf{y}')T^Q(\mathbf{y}' \rightarrow \mathbf{y}) / P(\mathbf{y})T^Q(\mathbf{y} \rightarrow \mathbf{y}'))$$
- Detailed balance satisfied
- Acceptance probability often easy to compute even though sampling according to P difficult

Learning in MNs: 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 the likelihood of this setting 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 changes 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 maximum likelihood function. We can then incrementally move each weight in direction of the gradient based on a learning parameter η
- The above approach amounts to differencing the expectation of priors and observed occurrences

Markov Net Learning, continued

- Assume that the dataset is composed of m data points. Consider the task of computing the expectation of priors and observed occurrences for $A \wedge B$
- Expectation of priors: $m \cdot Pr(A \wedge B)$
- Observed occurrences: Number of data points for which A and B hold
- Using this approach, it can be shown that gradient ascent converges

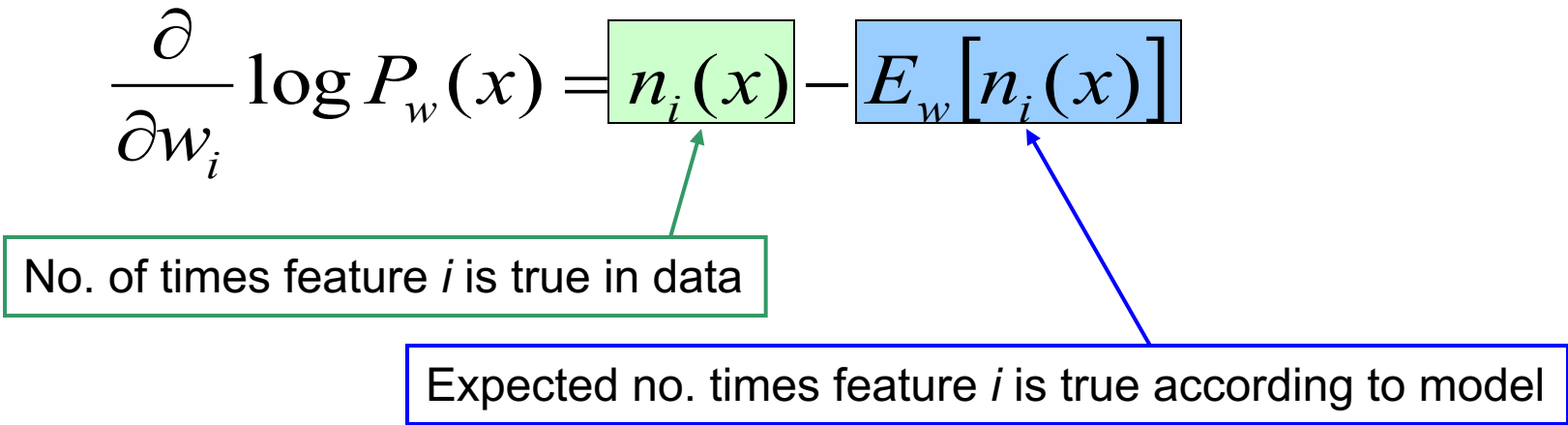
Analyzing

$$\Pr(\vec{V}) = \frac{1}{Z} \exp \sum_i w_i f_i(\vec{V})$$

- In this formulation, the w 's are just weights and the f 's are just features
- As such, we can throw the graph out if we want – we have everything we need in the w_i s and f_i s
- In this view, parameter learning is just weight learning

Weight Learning

- Maximize likelihood or posterior probability
- Numerical optimization (gradient or 2nd order)
- Negative likelihood is *convex*: No local maxima

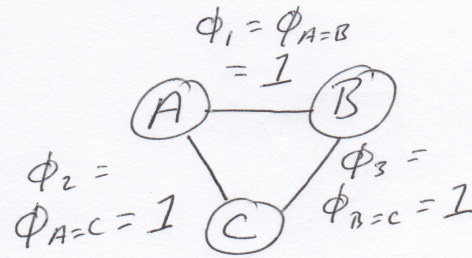
$$\frac{\partial}{\partial w_i} \log P_w(x) = \boxed{n_i(x)} - \boxed{E_w[n_i(x)]}$$


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!)

Example:



$$Z = \sum_{\vec{v}} e^{\sum_i \phi_i(\vec{v})}$$

$$\approx 56.2$$

\vec{v}			$e^{\sum_i \phi_i(\vec{v})}$	$P_{\vec{\phi}}(\vec{v})$
A	B	C		
0	0	0	$e^3 \approx 20$.35
0	0	1	$e \approx 2.7$.05
0	1	0	.	.
0	1	1	.	.
1	0	0	.	.
1	0	1	.	.
1	1	0	.	.
1	1	1	$e^3 \approx 20$.35

$$\#_{A=B} = 3$$

$$\#_{B=C} = 3$$

$$\#_{A=C} = 3$$

$$E_{\vec{\phi}, 101=5} [\#_{A=B}] = .8(5) = 4$$

same for $\#_{A=C}$, $\#_{B=C}$

$$\text{Gradient: } \langle -1, -1, -1 \rangle$$

$$\text{New } \vec{\phi} : \langle 1, 1, 1 \rangle + \eta \langle -1, -1, -1 \rangle$$

$$(\text{e.g., } .05) = \langle .95, .95, .95 \rangle$$

For real-world problems, Z takes too long to compute

- Combine gradient descent and MCMC inference: *Persistent Contrastive Divergence (PCD)*
- Use *Pseudo-likelihood* approximation to Likelihood: just do many logistic regressions, to get parents (Markov blanket) of each node
- Often include Lasso penalty... *screening rules* can eliminate many features by guaranteeing their weights will be 0 for a given λ
- Can avoid problem altogether if use continuous variables instead of discrete, and use Gaussian graphical model: *Graphical Lasso*

PCD- k

- Initialize $m=100$ MCMC (Gibbs) chains
- After every k steps of Gibbs sampling, take a gradient step in weight space, based on expected feature counts from the last state of each of the m chains
- Don't restart Gibbs chains after weight update: continue same chains
- Common to use $k=1$
- Justification:
 - Weight updates are small so...
 - After any weight update the model has changed very little
 - Weight update only needs right gradient direction, not exact expectations