COMP9418: Advanced Topics in Statistical Machine Learning

Markov Networks

Instructor: Gustavo Batista

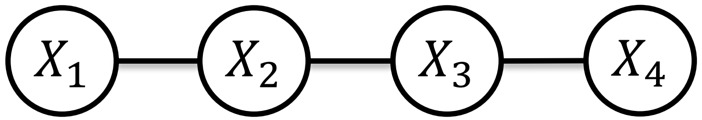
University of New South Wales

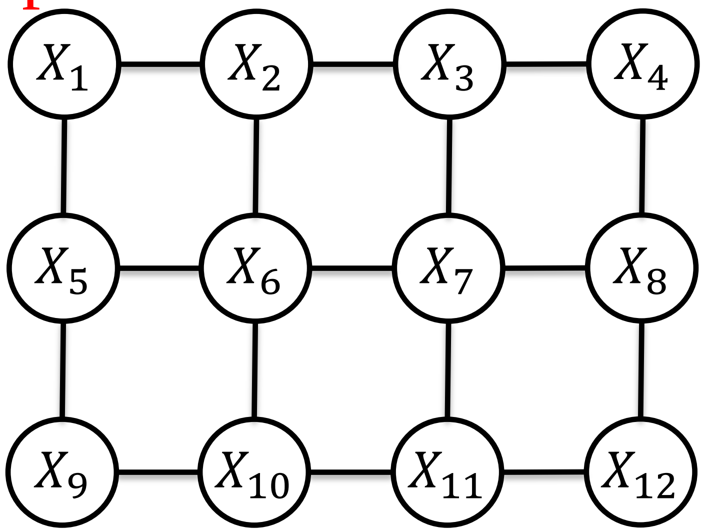
Introduction

- This lecture discusses Markov networks
  - These are undirected graphical models
  - They are frequently used to model symmetrical dependencies, as in case of pixels in an image
- Like Bayesian networks, Markov networks are used to model variable independencies
  - However, these representations are not redundant
  - There exist sets of independencies that can be expressed in a Markov network but not in a Bayesian network and vice-versa
- We will discuss the semantics of Markov networks
  - As well as some inference algorithms such as stochastic search and variable elimination

Introduction

- Several processes such as an sentence or image can be modelled as a series of states in a chain or grid
  - Each state can be influenced by the state of its neighbours
  - Such symmetry is modelled using undirected graphs called *Markov random fields* (MRFs) or *Markov networks* (MN)
- MNs were proposed to model ferromagnetic materials
  - In Physics, these models are known as *Ising* models
  - Each variable represents a dipole with two states + and –
  - The state of each dipole depends on an external field and its neighbours

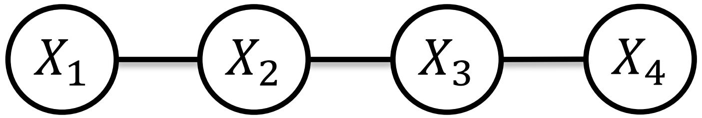


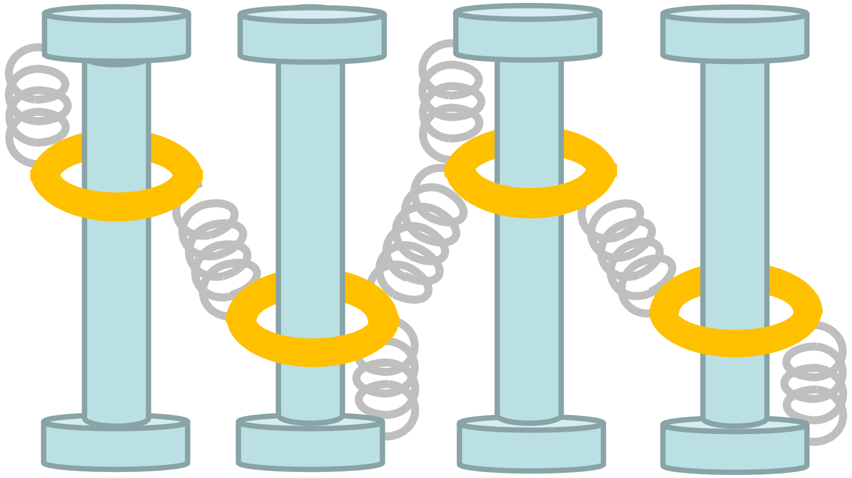


Introduction

- In an MN, a variable is independent of all other variables given its neighbours
  - For instance, in this figure, $X_1 \perp X_3, X_4 | X_2$
  - Therefore, $P(X_1|X_2, X_3, X_4) = P(X_1|X_2)$

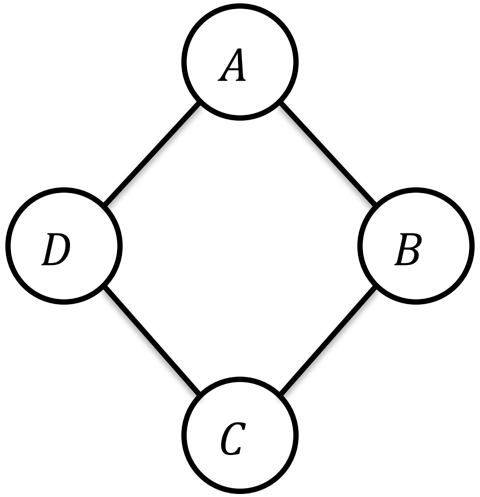
- A common query is to find the instantiation of maximum probability
  - MAP or MPE query
  - The probability of each instantiation depends on an *external* influence (prior) and the *internal* influence (likelihood)
  - MNs can be thought as a series of rings in poles, where each ring is a variable, and the height of a ring corresponds to its state





Voting Example

- Suppose that we are modeling voting preferences among four persons A, B, C, D
  - Let's say that $A - B$, $B - C$, $C - D$, and $D - A$ are friends
  - Friends can influence each other
  - These influences can be naturally represented by an undirected graph



- In this example, A does not interact directly with C. The same occurs with B and D
  - $A \perp C | B, D$ and $B \perp D | A, C$
  - We saw there is no Bayesian network that can represent *only* these independence assumption (Lecture 4 – Slide 33)

Voting Example

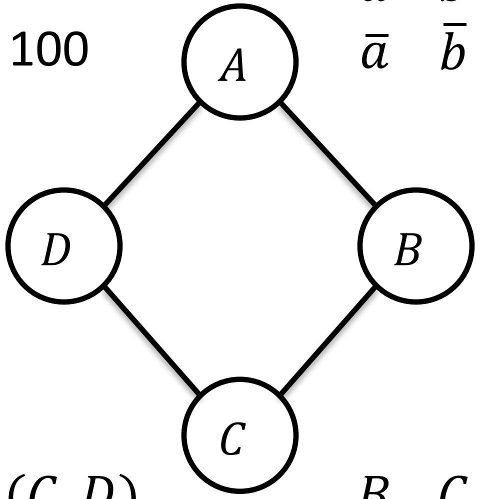
- Like Bayesian networks, Markov networks encode independence assumptions
  - Variables that are not independent must be in some *factor*
  - Factor is a generalization of a CPT. It does not need to store values in the range $0 - 1$
- In this example, we can factorise the joint distribution as

$$P(A,B,C,D) = \frac{1}{Z}\phi_1(A,B)\ \phi_2(B,C)\ \phi_3(C,D)\ \phi_4(D,A)$$

- Z is a normalizing constant known as the *partition function*
  - $Z = \sum_{A,B,C,D} \tilde{P}(A,B,C,D)$
  - $\tilde{P}(A,B,C,D) = \phi_1(A,B)\ \phi_2(B,C)\ \phi_3(C,D)\ \phi_4(D,A)$

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



C	D	$\phi_3(C,D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

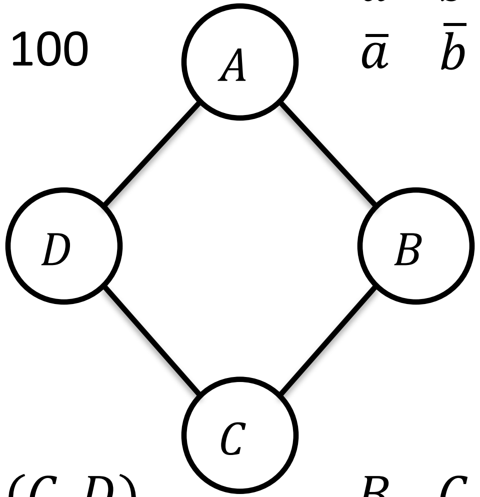
B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Voting Example

- We can view $\phi(A, B)$ as an interaction that pushes B's vote closer to that of A
  - The term $\phi(B, C)$ pushes B's vote closer to C, but C pushes D's vote away (and vice-versa).
  - The most likely vote will require reconciling these conflicting influences
- We simply indicate a level of coupling between dependent variables in the graph
  - This requires less prior knowledge than CPTs
  - It defines an energy landscape over the space of possible assignments
  - We convert this energy to a probability via the normalization constant

D	A	$\phi_4(D, A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A, B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



C	D	$\phi_3(C, D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

B	C	$\phi_2(B, C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

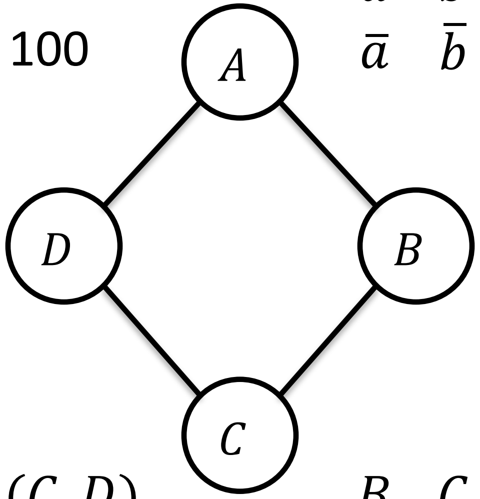
Voting Example

MPE assignment

Assignment				Unnormalized	Normalized
a	b	c	d	300,000	0.04
a	b	c	$\bar{d}$	300,000	0.04
a	b	$\bar{c}$	d	300,000	0.04
a	b	$\bar{c}$	$\bar{d}$	30	$4.1\ 10^{-6}$
a	$\bar{b}$	c	d	500	$6.9\ 10^{-5}$
a	$\bar{b}$	c	$\bar{d}$	500	$6.9\ 10^{-5}$
a	$\bar{b}$	$\bar{c}$	d	5,000,000	0.69
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	500	$6.9\ 10^{-5}$
$\bar{a}$	b	c	d	100	$1.4\ 10^{-5}$
$\bar{a}$	b	c	$\bar{d}$	1,000,000	0.14
$\bar{a}$	b	$\bar{c}$	d	100	$1.4\ 10^{-5}$
$\bar{a}$	b	$\bar{c}$	$\bar{d}$	100	$1.4\ 10^{-5}$
$\bar{a}$	$\bar{b}$	c	d	10	$1.4\ 10^{-6}$
$\bar{a}$	$\bar{b}$	c	$\bar{d}$	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	100,000	0.014

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



C	D	$\phi_3(C,D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Voting Example

- Although expensive, the joint probability can be used to answer probabilistic queries
  - Prior marginal queries, such as $P(A, B)$

A	B	$P(A,B)$
a	b	.13
a	$\bar{b}$	.69
$\bar{a}$	b	.14
$\bar{a}$	$\bar{b}$	.04

  - Probability of evidence, such as $P(\bar{b}) = 0.732$
  - Posterior marginal, such as $P(\bar{b}|c) = 0.06$

Assignment				Unnormalized	Normalized
a	b	c	d	300,000	0.04
a	b	c	$\bar{d}$	300,000	0.04
a	b	$\bar{c}$	d	300,000	0.04
a	b	$\bar{c}$	$\bar{d}$	30	$4.1\ 10^{-6}$
a	$\bar{b}$	c	d	500	$6.9\ 10^{-5}$
a	$\bar{b}$	c	$\bar{d}$	500	$6.9\ 10^{-5}$
a	$\bar{b}$	$\bar{c}$	d	5,000,000	0.69
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	500	$6.9\ 10^{-5}$
$\bar{a}$	b	c	d	100	$1.4\ 10^{-5}$
$\bar{a}$	b	c	$\bar{d}$	1,000,000	0.14
$\bar{a}$	b	$\bar{c}$	d	100	$1.4\ 10^{-5}$
$\bar{a}$	b	$\bar{c}$	$\bar{d}$	100	$1.4\ 10^{-5}$
$\bar{a}$	$\bar{b}$	c	d	10	$1.4\ 10^{-6}$
$\bar{a}$	$\bar{b}$	c	$\bar{d}$	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	100,000	0.014

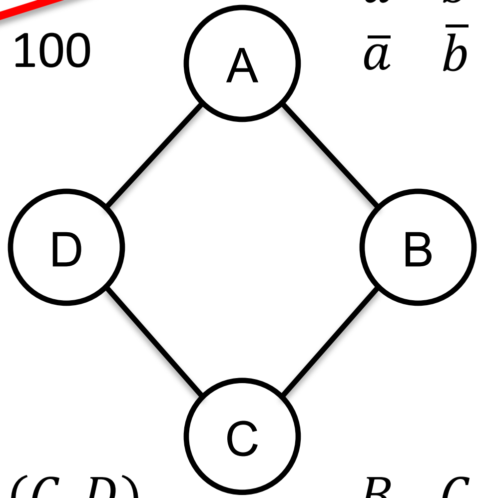
Voting Example: Bad News for Learning!

- Suppose we had learned $P(A, B)$ from data
  - By counting the occurrences of a and b
  - $P(A, B)$ is not a direct replacement for $\phi_1(A, B)$

D	A	$\phi_4(D, A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A, B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

A	B	$P(A, B)$
a	b	.13
a	$\bar{b}$	.69
$\bar{a}$	b	.14
$\bar{a}$	$\bar{b}$	.04



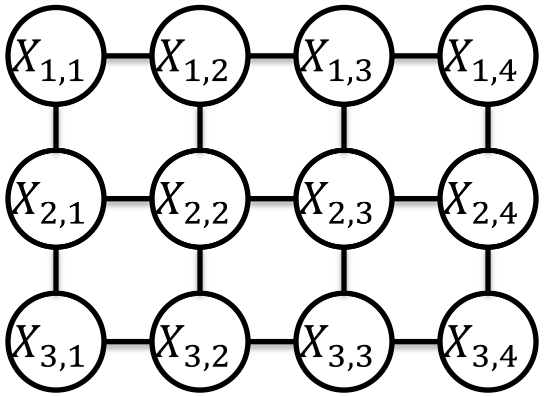
C	D	$\phi_3(C, D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

B	C	$\phi_2(B, C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Random Field

- A *random field* $\boldsymbol{X}$ is a set of random variables
  - It is common that each variable X_i to be associated with a *site*
  - This idea comes from areas such as image processing in which each variable is associated with a pixel or region
- We use a set $\boldsymbol{S}$ to index a set of n sites
  - The sites can be spatially *regular*, as in the case of a 2D image
  - Or *irregular*, if they do not present spatial regularity
- The sites in $\boldsymbol{S}$ are related to one another via a neighborhood system
  - A site is not neighboring to itself: $i \notin \boldsymbol{N}_i$
  - The neighboring relationship is mutual: $i \in \boldsymbol{N}_{i'}$ iff $i' \in \boldsymbol{N}_i$

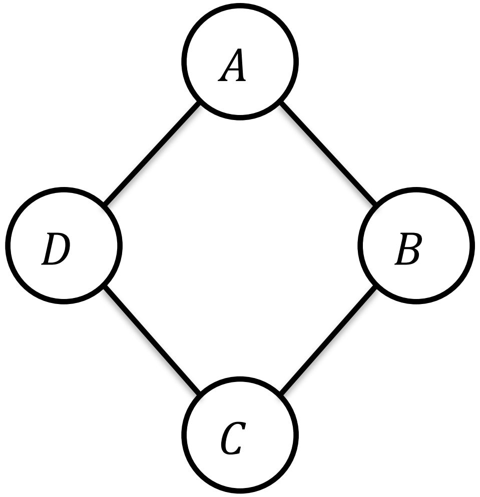
$$\boldsymbol{X} = \{X_1, \ldots, X_n\}$$



$\boldsymbol{N}_i$ is a set of sites neighboring i

$$\boldsymbol{N} = \{\boldsymbol{N}_i | \forall i \in \boldsymbol{S}\}$$

Markov Networks



- A random field $\boldsymbol{X}$ is a *Markov random field* (or *Markov network*) on $\boldsymbol{S}$ w.r.t. a neighbourhood system $\boldsymbol{N}$ if and only if
  - $P(X_1 = x_1, \dots, X_n = x_n) > 0, \forall \boldsymbol{x} \in \boldsymbol{X}$ (positivity)
  - $P\left(X_i \middle| \boldsymbol{X}_{\boldsymbol{S}\setminus\{i\}}\right) = P(X_i | \boldsymbol{X}_{\boldsymbol{N}_i})$ (Markovianity)

- Graphically, Markov networks (MN) are undirected graphical models
  - $G = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ consists of a set of random variables, and $\boldsymbol{E}$ a set of undirected edges
  - A set of variables $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ given $\boldsymbol{Z}$, if the variables in $\boldsymbol{Z}$ separate $\boldsymbol{X}$ and $\boldsymbol{Y}$ in the graph
  - Therefore, if we remove the nodes in $\boldsymbol{Z}$ from the graph, there will be no paths between $\boldsymbol{X}$ and $\boldsymbol{Y}$

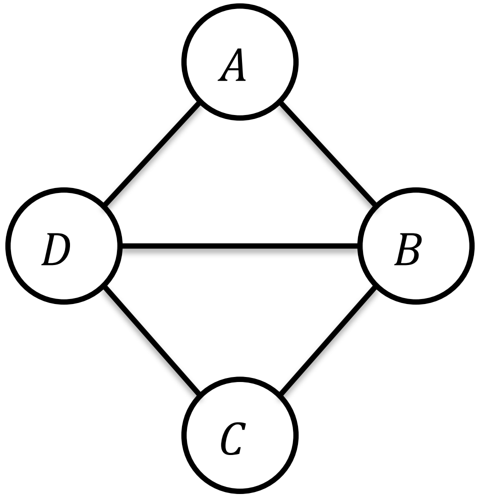
Markov Networks: Gibbs Distribution

- When the positivity condition is satisfied, the joint probability distribution is uniquely determined by the *Gibbs distribution*
  - This result is known as the *Hammersley-Clifford theorem*
  - Like in Bayesian networks, it allow us to factorise the full joint distribution into smaller factors
  - Therefore, we can efficiently answer probabilistic queries

$$P(\boldsymbol{X}) = \frac{1}{Z} \prod_{c \in cliques(G)} \phi_c(\boldsymbol{X}_c)$$

$$Z = \sum_{\boldsymbol{x}} \prod_{c \in cliques(G)} \phi_c(\boldsymbol{X}_c)$$

- Using the example, we have the following factorisation for maximal cliques
  - $P(A,B,C,D) = \frac{1}{Z}\phi_1(A,B,D)\phi_2(B,C,D)$
- In practice, we frequently use smaller cliques such as pairwise factors
  - $P(A,B,C,D) = \frac{1}{Z}\phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)\phi_5(D,B)$



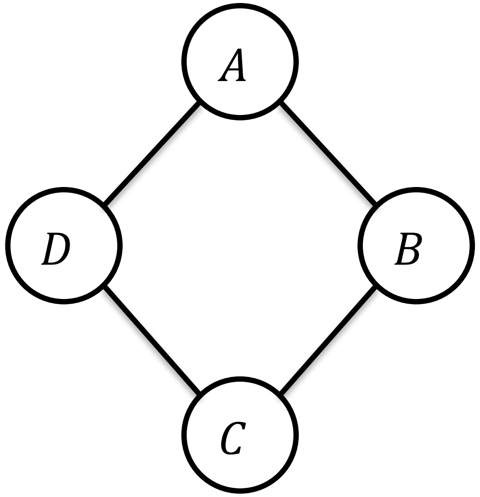
Markov Networks: Positivity

- This graph encodes the independencies
  - $A \perp C | B, D$ and $D \perp B | A, C$
  - Let us verify if this joint distribution has the same independence assumptions

B	D	A	$P(A\|B,D)$
b	d	a	.5
b	d	$\bar{a}$	.5
b	$\bar{d}$	a	1
b	$\bar{d}$	$\bar{a}$	0
$\bar{b}$	d	a	0
$\bar{b}$	d	$\bar{a}$	1
$\bar{b}$	$\bar{d}$	a	.5
$\bar{b}$	$\bar{d}$	$\bar{a}$	.5

B	D	C	$P(C\|B,D)$
b	d	c	1
b	d	$\bar{c}$	0
b	$\bar{d}$	c	.5
b	$\bar{d}$	$\bar{c}$	.5
$\bar{b}$	d	c	.5
$\bar{b}$	d	$\bar{c}$	.5
$\bar{b}$	$\bar{d}$	c	0
$\bar{b}$	$\bar{d}$	$\bar{c}$	1

B	D	A	C	$P(A,C\|B,D)$
b	d	a	c	.5
b	d	a	$\bar{c}$	0
b	d	$\bar{a}$	c	.5
b	d	$\bar{a}$	$\bar{c}$	0
b	$\bar{d}$	a	c	.5
b	$\bar{d}$	a	$\bar{c}$	.5
b	$\bar{d}$	$\bar{a}$	c	0
b	$\bar{d}$	$\bar{a}$	$\bar{c}$	0
$\bar{b}$	d	a	c	0
$\bar{b}$	d	a	$\bar{c}$	0
$\bar{b}$	d	$\bar{a}$	c	.5
$\bar{b}$	d	$\bar{a}$	$\bar{c}$	.5
$\bar{b}$	$\bar{d}$	a	c	0
$\bar{b}$	$\bar{d}$	a	$\bar{c}$	.5
$\bar{b}$	$\bar{d}$	$\bar{a}$	c	0
$\bar{b}$	$\bar{d}$	$\bar{a}$	$\bar{c}$	.5



A	B	C	D	$P(.)$
a	b	c	d	1/8
a	b	c	$\bar{d}$	1/8
a	b	$\bar{c}$	$\bar{d}$	1/8
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	1/8
$\bar{a}$	b	c	d	1/8
$\bar{a}$	$\bar{b}$	c	d	1/8
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	1/8
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	1/8

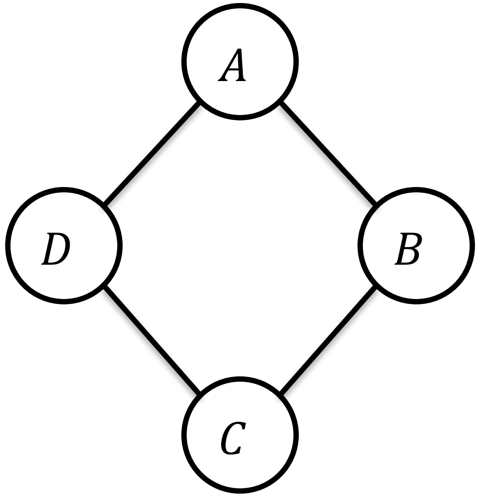
Markov Networks: Positivity

- This graph encodes the independencies
  - $A \perp C | B, D$ and $D \perp B | A, C$
  - Let us verify if this joint distribution has the same independence assumptions

A	C	B	$P(B\|A,C)$
a	c	b	1
a	c	$\bar{b}$	0
a	$\bar{c}$	b	.5
a	$\bar{c}$	$\bar{b}$	.5
$\bar{a}$	c	b	.5
$\bar{a}$	c	$\bar{b}$	.5
$\bar{a}$	$\bar{c}$	b	0
$\bar{a}$	$\bar{c}$	$\bar{b}$	1

A	C	D	$P(D\|A,C)$
a	c	d	.5
a	c	$\bar{d}$	.5
a	$\bar{c}$	d	0
a	$\bar{c}$	$\bar{d}$	1
$\bar{a}$	c	d	1
$\bar{a}$	c	$\bar{d}$	0
$\bar{a}$	$\bar{c}$	d	.5
$\bar{a}$	$\bar{c}$	$\bar{d}$	.5

A	C	B	D	$P(B,D\|A,C)$
a	c	b	d	.5
a	c	b	$\bar{d}$	.5
a	c	$\bar{b}$	d	0
a	c	$\bar{b}$	$\bar{d}$	0
a	$\bar{c}$	b	d	0
a	$\bar{c}$	b	$\bar{d}$	.5
a	$\bar{c}$	$\bar{b}$	d	0
a	$\bar{c}$	$\bar{b}$	$\bar{d}$	.5
$\bar{a}$	c	b	d	.5
$\bar{a}$	c	b	$\bar{d}$	0
$\bar{a}$	c	$\bar{b}$	d	.5
$\bar{a}$	c	$\bar{b}$	$\bar{d}$	0
$\bar{a}$	$\bar{c}$	b	d	0
$\bar{a}$	$\bar{c}$	b	$\bar{d}$	0
$\bar{a}$	$\bar{c}$	$\bar{b}$	d	.5
$\bar{a}$	$\bar{c}$	$\bar{b}$	$\bar{d}$	.5

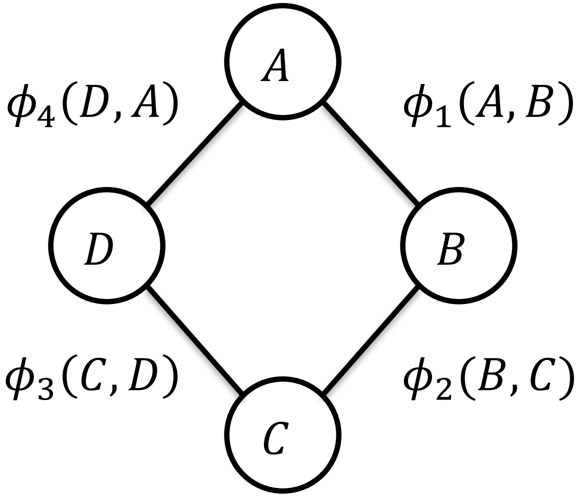


A	B	C	D	$P(.)$
a	b	c	d	1/8
a	b	c	$\bar{d}$	1/8
a	b	$\bar{c}$	$\bar{d}$	1/8
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	1/8
$\bar{a}$	b	c	d	1/8
$\bar{a}$	$\bar{b}$	c	d	1/8
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	1/8
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	1/8

Markov Networks: Positivity

- This graph encodes the independencies
  - $A \perp C | B, D$ and $D \perp B | A, C$
  - Let us verify if this joint distribution has the same independences assumptions

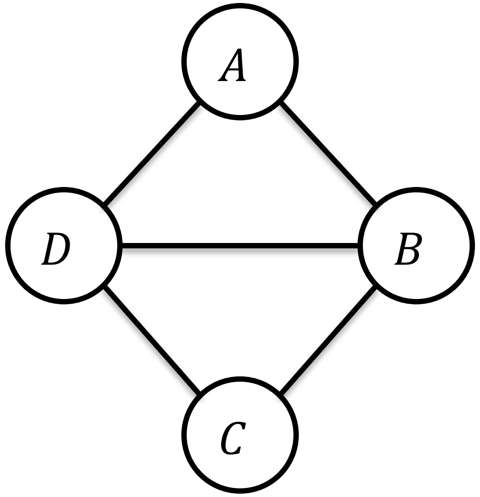
  - $P(\bar{a}, b, c, \bar{d}) = \phi_1(\bar{a}, b)\phi_2(b, c)\phi_3(c, \bar{d})\phi_4(\bar{d}, \bar{a}) = 0$
  - $P(\bar{a}, b, c, d) = \phi_1(\bar{a}, b)\phi_2(b, c)\phi_3(c, d)\phi_4(d, \bar{a}) = \frac{1}{8}$
  - $P(\bar{a}, \bar{b}, \bar{c}, \bar{d}) = \phi_1(\bar{a}, \bar{b})\phi_2(\bar{b}, \bar{c})\phi_3(\bar{c}, \bar{d})\phi_4(\bar{d}, \bar{a}) = \frac{1}{8}$
  - $P(a, b, c, \bar{d}) = \phi_1(a, b)\phi_2(b, c)\phi_3(c, \bar{d})\phi_4(\bar{d}, a) = \frac{1}{8}$



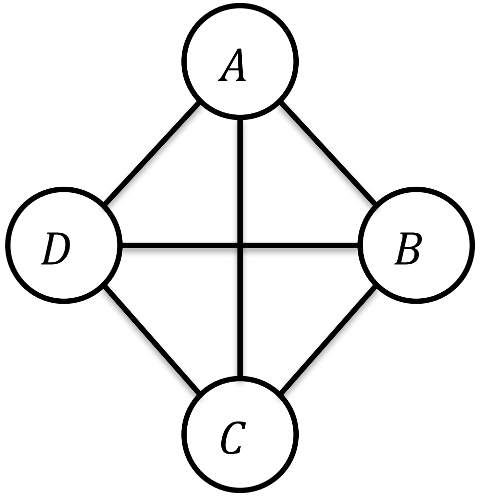
A	B	C	D	$P(.)$
a	b	c	d	1/8
a	b	c	$\bar{d}$	1/8
a	b	$\bar{c}$	$\bar{d}$	1/8
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	1/8
$\bar{a}$	b	c	d	1/8
$\bar{a}$	$\bar{b}$	c	d	1/8
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	1/8
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	1/8

Gibbs Distribution and Graph

- Different Gibbs distributions may induce a same undirected graph
  - $\phi_1(A,B,D)\phi_2(B,C,D)$
  - $\phi_1(A,B,D)\phi_2(B,D)\phi_3(B,C)\phi_4(C,D)$
  - $\phi_1(A,B)\phi_2(A,D)\phi_3(B,D)\phi_4(B,C)\phi_5(C,D)$



- Therefore, we cannot read the factorization from the graph
  - All these factorizations have the same independence assumptions
  - However, they do not have the same representational power
  - For example, for a fully connected graph, a maximal clique has $O(d^n)$ parameters, but a pairwise graph has only $O(n^2d^2)$ parameters



Potentials

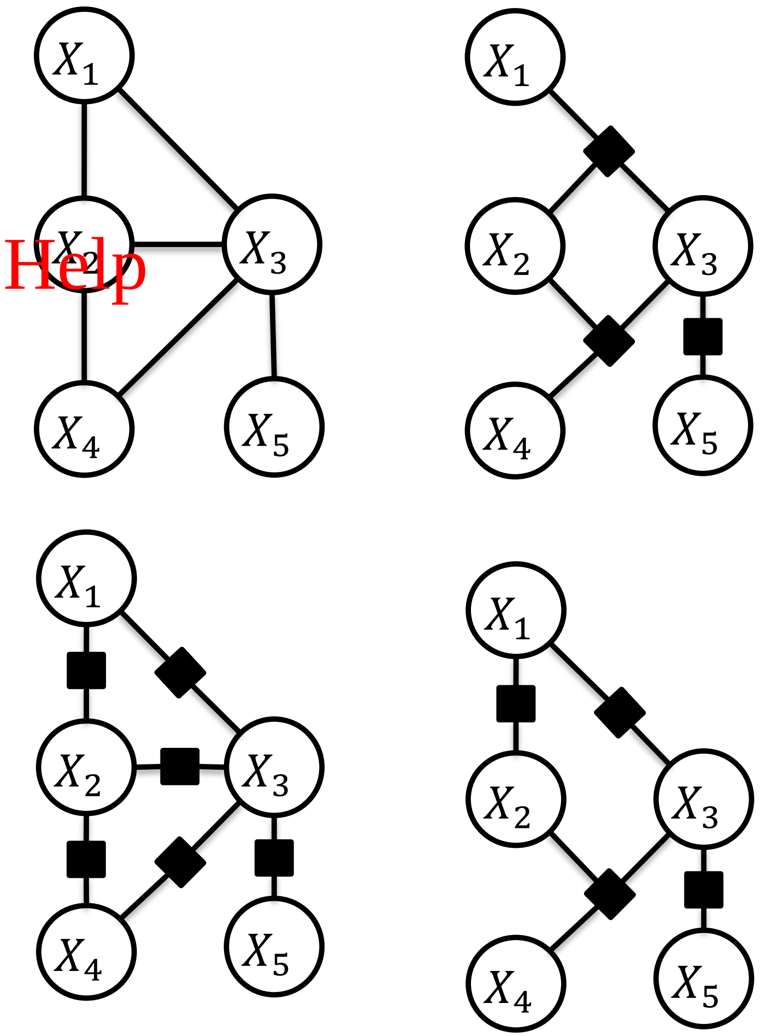
- Clique factors can be:
  - Single-node factors: specify an affinity for a particular candidate $\phi_A(a) = .8$
  - Pairwise-factors: enforce affinities between friends $\phi_{AB}(a, b) = 100 \text{ if } a = b$
  - Higher-order: important to specify relationships among sets of variables $\phi_{ABC}(a, b, c) = 100 \text{ if } a \oplus b \oplus c$

The normalization Z makes the factors scale invariant!

Factor Graphs

- A factor graph is a graph containing two types of nodes
  - Random variables
  - Factors over the sets of variables

- It allow us to derive the factorization without ambiguity
  - $P(X_1, X_2, X_3, X_4, X_5) = P(X_1, X_2, X_3) P(X_2, X_3, X_4) P(X_3, X_5)$
  - $P(X_1, X_2, X_3, X_4, X_5) = P(X_1, X_2) P(X_1, X_3) P(X_2, X_3) P(X_2, X_4) P(X_3, X_4) P(X_3, X_5)$
  - $P(X_1, X_2, X_3, X_4, X_5) = P(X_1, X_2) P(X_1, X_3) P(X_2, X_3, X_4) P(X_3, X_5)$



Energy Functions

- The joint probability in a MN is frequently expressed in terms of energy functions
  - $E(\boldsymbol{X})$ is the energy. Therefore, maximising $P(\boldsymbol{X})$ is equivalent to minimising $E(\boldsymbol{X})$
  - The energy function can be written in terms of local functions ψ_c known as *potentials*

$$P(\boldsymbol{X}) = \frac{1}{Z}\exp(-E(\boldsymbol{X}))$$

$$E(\boldsymbol{X}) = \sum_{c \in Cliques(G)} \psi_c(\boldsymbol{X}_c)$$

$$P(\boldsymbol{X}) = \frac{1}{Z}\exp\left(-\sum_{c \in Cliques(G)} \psi_c(\boldsymbol{X}_c)\right)$$

- Why?
  - Historical: statistical physics

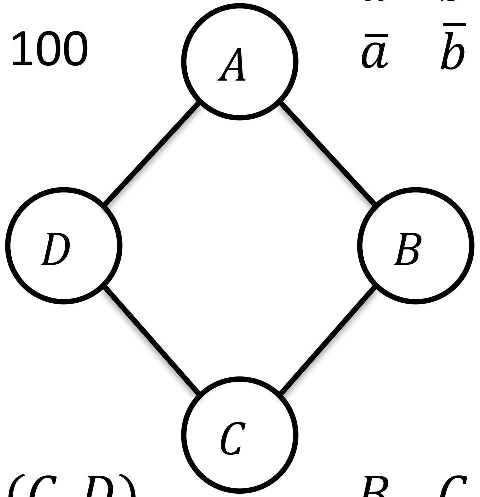
$$\psi(\boldsymbol{X}_c) = -\log \phi_c(\boldsymbol{X}_c)$$

Voting Example

Assignment				Unnormalized	Normalized
a	b	c	d	300,000	0.04
a	b	c	$\bar{d}$	300,000	0.04
a	b	$\bar{c}$	d	300,000	0.04
a	b	$\bar{c}$	$\bar{d}$	30	4.1 10^{-6}
a	$\bar{b}$	c	d	500	6.9 10^{-5}
a	$\bar{b}$	c	$\bar{d}$	500	6.9 10^{-5}
a	$\bar{b}$	$\bar{c}$	d	5,000,000	0.69
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	500	6.9 10^{-5}
$\bar{a}$	b	c	d	100	1.4 10^{-5}
$\bar{a}$	b	c	$\bar{d}$	1,000,000	0.14
$\bar{a}$	b	$\bar{c}$	d	100	1.4 10^{-5}
$\bar{a}$	b	$\bar{c}$	$\bar{d}$	100	1.4 10^{-5}
$\bar{a}$	$\bar{b}$	c	d	10	1.4 10^{-6}
$\bar{a}$	$\bar{b}$	c	$\bar{d}$	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	100,000	0.014

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



C	D	$\phi_3(C,D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

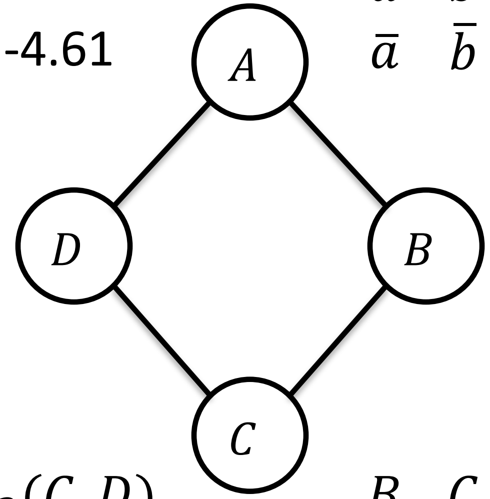
B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Voting Example

Assignment				Unnormalized	Normalized
a	b	c	d	300,000	0.04
a	b	c	$\bar{d}$	300,000	0.04
a	b	$\bar{c}$	d	300,000	0.04
a	b	$\bar{c}$	$\bar{d}$	30	$4.1\ 10^{-6}$
a	$\bar{b}$	c	d	500	$6.9\ 10^{-5}$
a	$\bar{b}$	c	$\bar{d}$	500	$6.9\ 10^{-5}$
a	$\bar{b}$	$\bar{c}$	d	5,000,000	0.69
a	$\bar{b}$	$\bar{c}$	$\bar{d}$	500	$6.9\ 10^{-5}$
$\bar{a}$	b	c	d	100	$1.4\ 10^{-5}$
$\bar{a}$	b	c	$\bar{d}$	1,000,000	0.14
$\bar{a}$	b	$\bar{c}$	d	100	$1.4\ 10^{-5}$
$\bar{a}$	b	$\bar{c}$	$\bar{d}$	100	$1.4\ 10^{-5}$
$\bar{a}$	$\bar{b}$	c	d	10	$1.4\ 10^{-6}$
$\bar{a}$	$\bar{b}$	c	$\bar{d}$	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	d	100,000	0.014
$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	100,000	0.014

D	A	$\psi_4(D,A)$
d	a	-4.61
d	$\bar{a}$	0
$\bar{d}$	a	0
$\bar{d}$	$\bar{a}$	-4.61

A	B	$\psi_1(A,B)$
a	b	-3.40
a	$\bar{b}$	-1.61
$\bar{a}$	b	0
$\bar{a}$	$\bar{b}$	-2.30

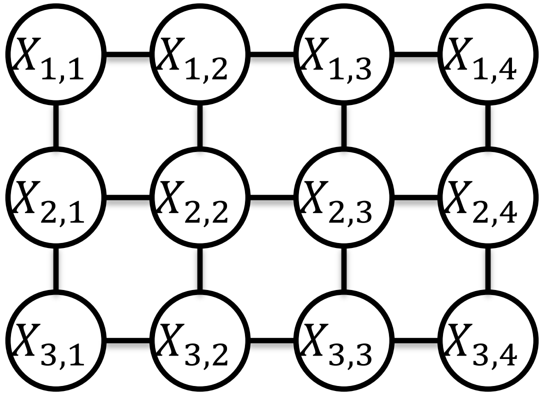


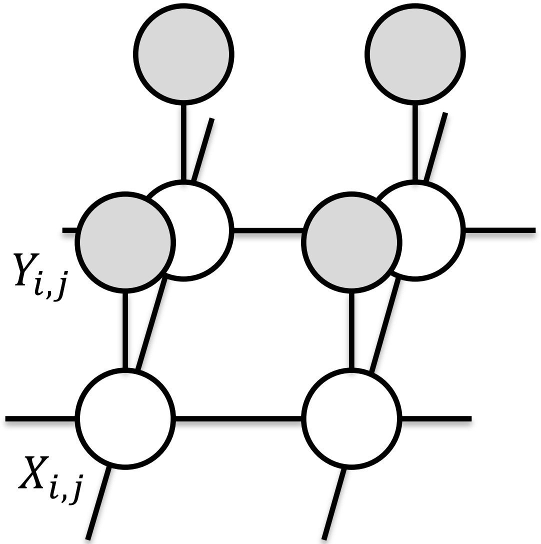
C	D	$\psi_3(C,D)$
c	d	0
c	$\bar{d}$	-4.61
$\bar{c}$	d	-4.61
$\bar{c}$	$\bar{d}$	0

B	C	$\psi_2(B,C)$
b	c	-4.61
b	$\bar{c}$	0
$\bar{b}$	c	0
$\bar{b}$	$\bar{c}$	-4.61

Pairwise Markov Networks

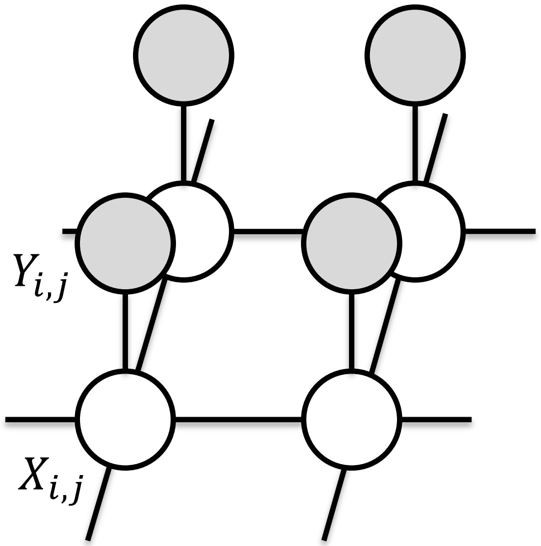
- Common subclass of Markov networks
  - All the factors are over single variables or pairs of variables
  - Node potentials: $\{\psi(X_i): i = 1, \ldots, n\}$
  - Edge potentials: $\{\psi(X_i, X_j): (X_i, X_j) \in H\}$
- Application: noise removal from binary images
  - Noisy image of pixel values, $Y_{i,j}$
  - Noise-free image of pixel values, $X_{i,j}$
  - Markov Net with
    - $\phi(X_{i,j}, X_{i',j'})$ potentials representing correlations between neighbouring pixels
    - $\phi(X_{i,j}, Y_{i,j})$ potentials describing correlations between same pixels in noise-free and noisy image

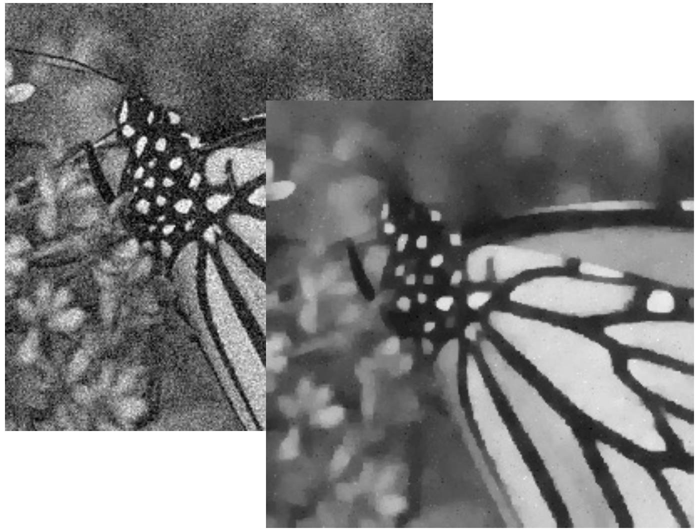




Example: Image Smoothing

- Many applications of Markov networks involve finding the MAP or MPE assignment
  - This is known as the MAP-MRF approach
  - Given the Gibbs distribution, it is equivalent to minimize the energy function
- The number of possible assignments is very large
  - It increases exponentially with the number of variables in the network
  - For instance, for a binary image of 100 x 100 pixels, there are $2^{10,000}$ possible assignments
- Finding the assignment of minimal energy is usually posed as a stochastic search
  - Start with a random value for each variable in the network
  - Improve this configuration via local operations
  - Until a configuration of (local) minimum energy is found





Stochastic Search Algorithm

Input: Markov network N with variables $\boldsymbol{X}$, energy function E

Output: an assignment $\boldsymbol{s}$ for $\boldsymbol{X}$ with minimum (local) energy

$\boldsymbol{s} \leftarrow$ initial assignment for every variable $X_i \in \boldsymbol{X}$

$\boldsymbol{s}_{prev} \leftarrow \boldsymbol{s}$

for $i = 1$ to I # I is maximum number of iterations

  $\boldsymbol{s}' \leftarrow \boldsymbol{s}$

  for each variable $X_i \in \boldsymbol{X}$ **do**

    $s_i{}' \leftarrow$ alternative value for variable X_i

    if $E(\boldsymbol{s}') < E(\boldsymbol{s})$ **or** random$(E(\boldsymbol{s}') - E(\boldsymbol{s})) < T$ **then**

      $\boldsymbol{s} \leftarrow \boldsymbol{s}'$ # T is threshold of accepting a change to a higher energy state

  if $\left|E(\boldsymbol{s}_{prev}) - E(\boldsymbol{s})\right| < \epsilon$ # ϵ is a convergence threshold

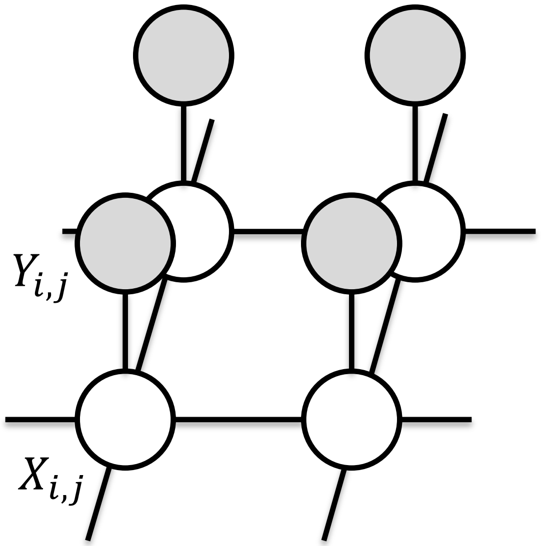
    break

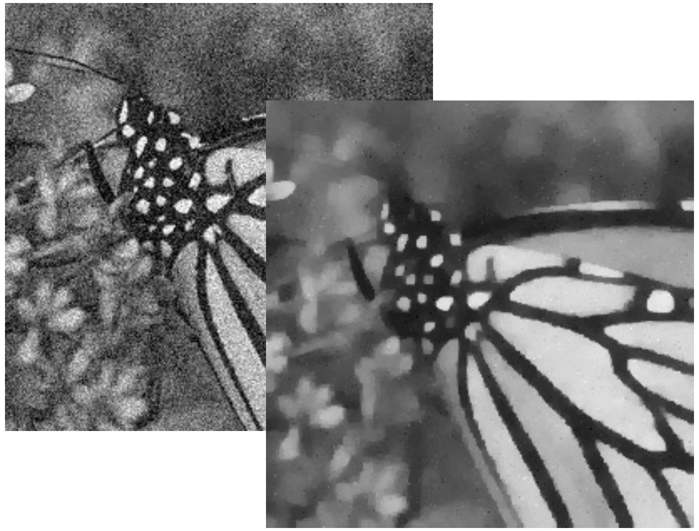
  $\boldsymbol{s}_{prev} \leftarrow \boldsymbol{s}$

return $\boldsymbol{s}$

Example: Image Smoothing

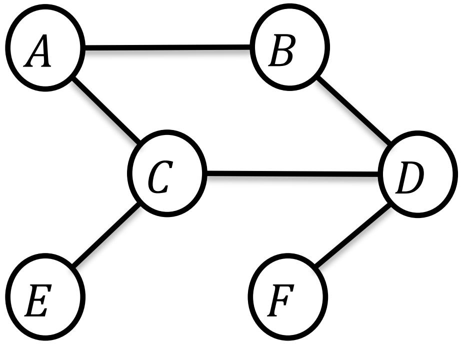
- This algorithm has three main variations
  - Iterative Conditional Modes (ICM): it always selects the assignment of minimum energy
  - Metropolis: with a fixed probability, p, it selects an assignment with higher energy
  - Simulated annealing (SA): with a variable probability, $P(T)$, it selects an assignment with higher energy. T is a parameter known as *temperature*. The probability of selecting a value with higher energy is determined by the expression $P(T) = e^{-\delta E/T}$ where δE is the energy difference. The value of T is reduced with each iteration

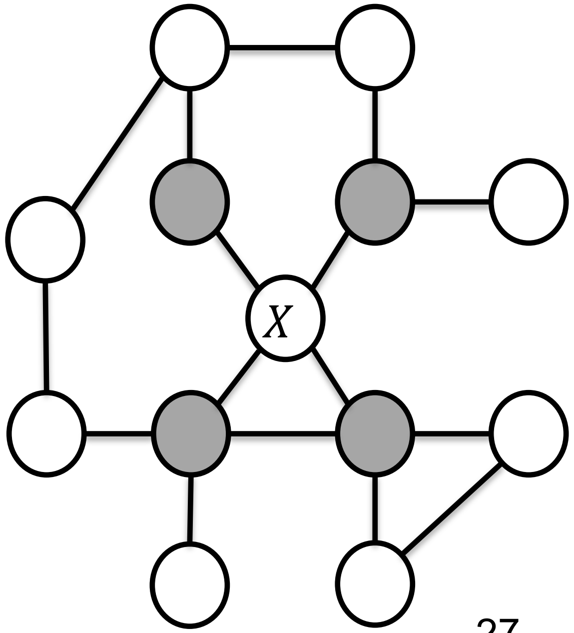




Local Independence

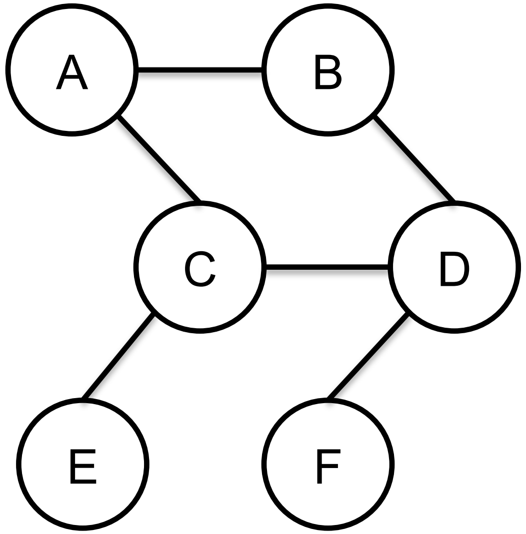
- In a Markov network the absence of edges imply in independence
  - Given an undirected graph $G = (\boldsymbol{V}, \boldsymbol{E})$
  - If the edge $X - Y \notin \boldsymbol{E}$ then $X \perp Y | \boldsymbol{V} \setminus \{X, Y\}$
  - These are known as *pairwise Markov independencies* of G
- Another local property of independence is the *Markov blanket*
  - As in the case of Bayesian networks, the Markov blanket U of a variable X is the set of nodes such that X is independent from the rest of the graph if U is observed
  - In the undirected case the Markov blanket turns out to be simply equal a node's neighborhood





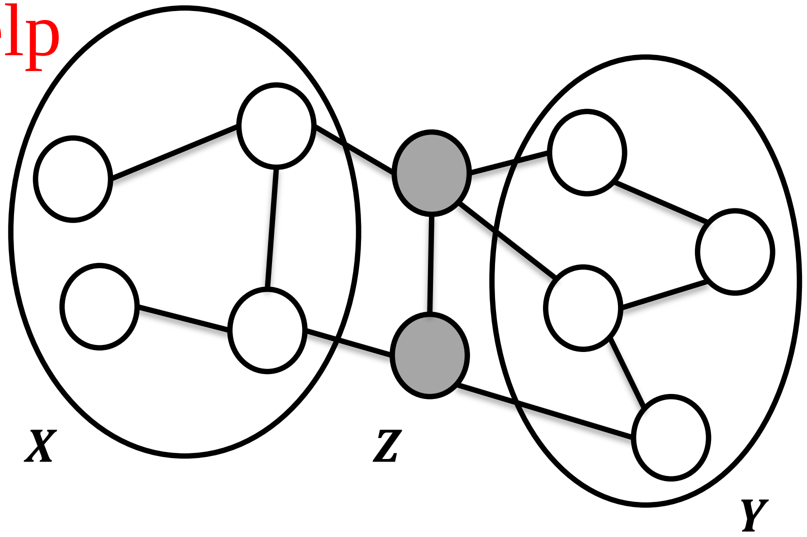
Global Independence: Separation

- A global interpretation of independence uses the idea of separation
  - Let $\boldsymbol{X}, \boldsymbol{Y}$, and $\boldsymbol{Z}$ be disjoint sets of nodes in a graph G. We say that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are separated by $\boldsymbol{Z}$, written $sep_G(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{Y})$, iff every path between a node in $\boldsymbol{X}$ and a node in $\boldsymbol{Y}$ is blocked by $\boldsymbol{Z}$
  - A path is blocked by $\boldsymbol{Z}$ iff at least one valve on the path is closed given $\boldsymbol{Z}$
  - Like Bayesian networks. But now, there is not the exception of convergent structures



Separation: Complexity

- The definition of separation considers all paths connecting a node in $\boldsymbol{X}$ with a node in $\boldsymbol{Y}$
  - In practice, this test is too inefficient
  - We can replace it by a *cut-set* test
- Two sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ of variables are separated by a set $\boldsymbol{Z}$ iff
  - There is no path from every node $X \in \boldsymbol{X}$ to every node $Y \in \boldsymbol{Y}$ after removing all nodes in $\boldsymbol{Z}$
  - $\boldsymbol{Z}$ is a *cut-set* between two parts of the graph

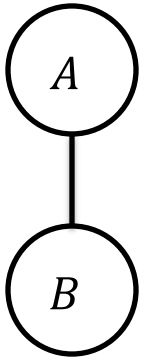


Separation: Soundness and Completeness

- Like d-separation, separation test is *sound*
  - If P is a probability distribution induced by a Markov network then $sep_G(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{Y})$ only if $\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z}$
  - We can safely use separation test to derive independence statements about the probability distributions induced by Markov networks

A	ϕ_A
a	5
$\bar{a}$	10

A	B	$\phi_{A,B}$
a	b	1
a	$\bar{b}$	1
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	1



B	ϕ_B
a	2
$\bar{a}$	20

- Like d-separation, separation test is not *complete*
  - The lack of separation does not imply into dependency
  - This is expected. As d-separation, separation only looks at the graph structure

Markov VS Bayesian Networks

Markov Nets

- Factors are easy to change (no normalization), but difficult to elicit
- Can be applied to problems with cycles or no natural directionality
- Difficult to read the factorization from the graph, but we can use factor graphs
- Z requires summing over all entries (NP-hard)

Bayes Nets

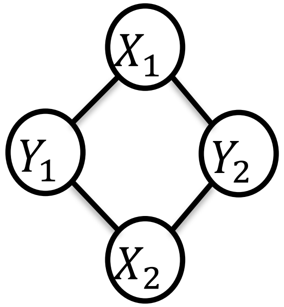
- Factors are easy to elicit from people
- Must have no cycles and edges are directed
- Graphs are easy to interpret particularly the causal ones
- Naturally normalized
- Easy to generate synthetic data from it (more about this later)

Markov VS Bayesian: Representation

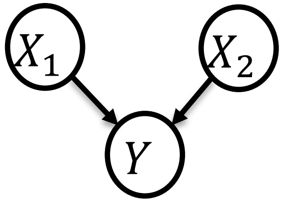
- Bayesian and Markov networks can be understood as languages to represent independencies
  - These languages can represent different sets of independencies
  - Therefore, these representations are not redundant
- For example, there is no directed graph that is a perfect map for the top case
  - Conversely, there is no undirected graph that is a perfect map for the bottom case
- In several circumstances, we need to find a Markov network that is an I-MAP for a Bayesian network
  - This is achievable through moralisation
  - We connect the parents of unmarried child nodes
  - We lose the marginal independence of parents

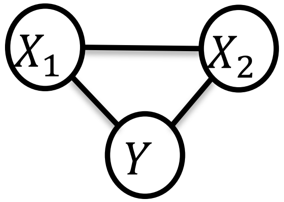
$X_1 \perp X_2 \mid Y_1, Y_2$

$Y_1 \perp Y_2 \mid X_1, X_2$



$X_1 \perp X_2$





Variable Elimination

- Let us now consider if Variable Elimination (VE) works for Markov networks
  - The idea of VE is to anticipate the elimination of variables
  - Using the network example, suppose we want to compute $P(A, B)$
- We start with the Gibbs distribution

$$P(A,B) = \sum_C \sum_D P(A,B,C,D)$$

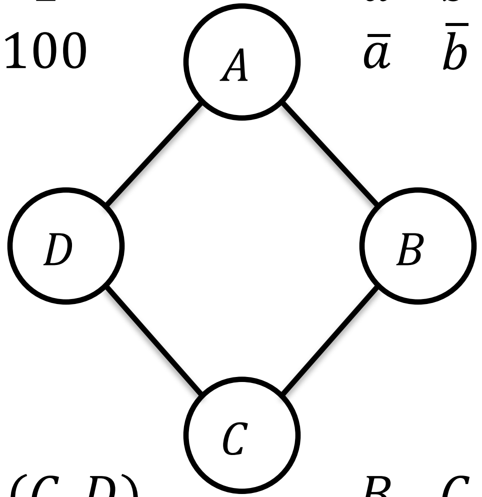
$$= \sum_C \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$

$$\propto \sum_C \sum_D \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$

$$= \phi_1(A,B) \sum_C \phi_2(B,C) \sum_D \phi_3(C,D)\phi_4(D,A)$$

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



C	D	$\phi_3(C,D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

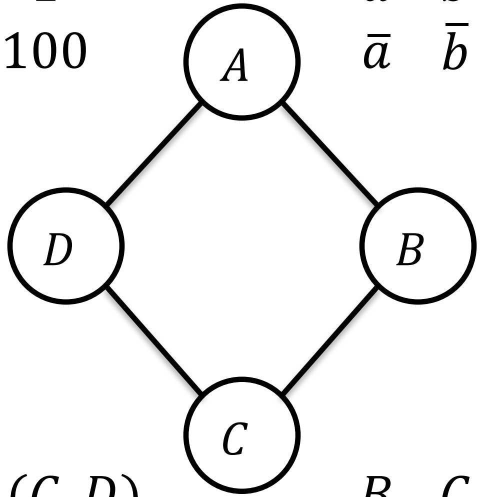
B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Variable Elimination

C	D	A	Φ_1
c	d	a	100
c	d	$\bar{a}$	1
c	$\bar{d}$	a	100
c	$\bar{d}$	$\bar{a}$	10000
$\bar{c}$	d	a	10000
$\bar{c}$	d	$\bar{a}$	100
$\bar{c}$	$\bar{d}$	a	1
$\bar{c}$	$\bar{d}$	$\bar{a}$	100

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



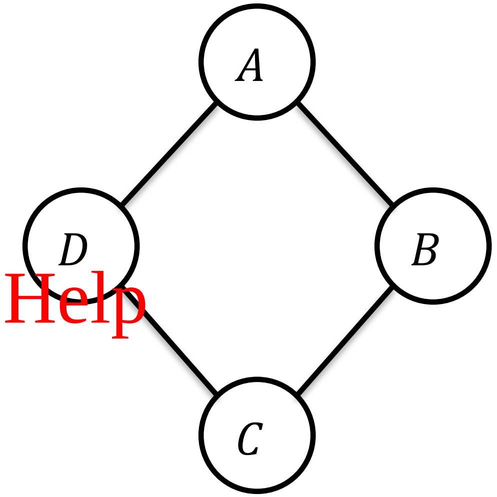
C	D	$\phi_3(C,D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Variable Elimination

- Let us now consider if Variable Elimination (VE) works for Markov networks
  - The idea of VE is to anticipate the elimination of variables
  - Using the network example, suppose we want to compute $P(A,B)$
- We start with the Gibbs distribution

$$P(A,B) = \sum_C \sum_D P(A,B,C,D)$$
$$= \sum_C \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$
$$\propto \sum_C \sum_D \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C) \sum_D \phi_3(C,D)\phi_4(D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C) \sum_D \Phi_1(C,D,A)$$



A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

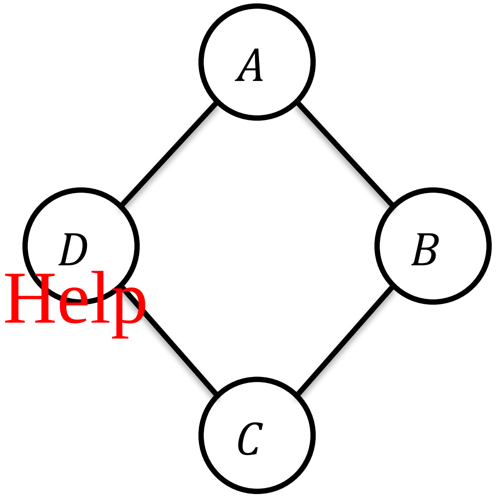
C	D	A	Φ_1
c	d	a	100
c	d	$\bar{a}$	1
c	$\bar{d}$	a	100
c	$\bar{d}$	$\bar{a}$	10000
$\bar{c}$	d	a	10000
$\bar{c}$	d	$\bar{a}$	100
$\bar{c}$	$\bar{d}$	a	1
$\bar{c}$	$\bar{d}$	$\bar{a}$	100

B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Variable Elimination

- Let us now consider if Variable Elimination (VE) works for Markov networks
  - The idea of VE is to anticipate the elimination of variables
  - Using the network example, suppose we want to compute $P(A,B)$
- We start with the Gibbs distribution

$$P(A,B) = \sum_C \sum_D P(A,B,C,D)$$
$$= \sum_C \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$
$$\propto \sum_C \sum_D \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C) \sum_D \phi_3(C,D)\phi_4(D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C) \sum_D \Phi_1(C,D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C)\tau_1(C,A)$$



A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

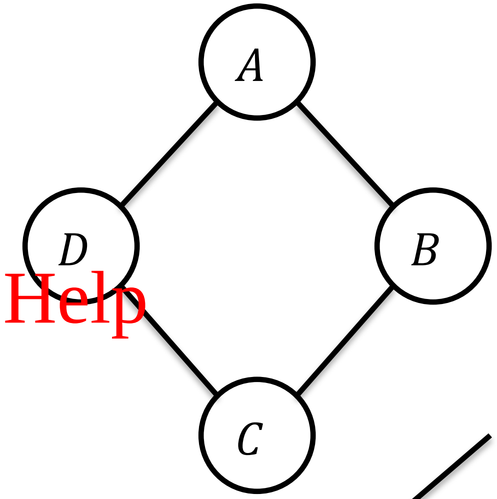
C	A	$\tau_1(C,A)$
c	a	200
c	$\bar{a}$	10001
$\bar{c}$	a	10001
$\bar{c}$	$\bar{a}$	200

B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

Variable Elimination

- We start with the Gibbs distribution

$$
\begin{aligned}
P(A,B) &= \textstyle\sum_C \sum_D P(A,B,C,D) \\
&= \textstyle\sum_C \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A) \\
&\propto \textstyle\sum_C \sum_D \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A) \\
&= \phi_1(A,B) \textstyle\sum_C \phi_2(B,C) \sum_D \phi_3(C,D)\phi_4(D,A) \\
&= \phi_1(A,B) \textstyle\sum_C \phi_2(B,C) \sum_D \Phi_1(C,D,A) \\
&= \phi_1(A,B) \textstyle\sum_C \phi_2(B,C) \tau_1(C,A) \\
&= \phi_1(A,B) \textstyle\sum_C \Phi_2(C,A,B)
\end{aligned}
$$



A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

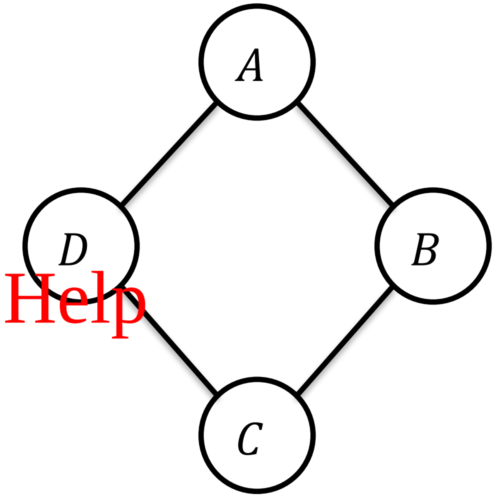
C	A	B	$\Phi_2(C,A,B)$
c	a	b	20000
c	a	$\bar{b}$	200
c	$\bar{a}$	b	1000100
c	$\bar{a}$	$\bar{b}$	10001
$\bar{c}$	a	b	10001
$\bar{c}$	a	$\bar{b}$	1000100
$\bar{c}$	$\bar{a}$	b	200
$\bar{c}$	$\bar{a}$	$\bar{b}$	20000

C	A	$\tau_1(A,C)$
c	a	200
c	$\bar{a}$	10001
$\bar{c}$	a	10001
$\bar{c}$	$\bar{a}$	200

Variable Elimination

- We start with the Gibbs distribution

$$
\begin{aligned}
P(A,B) &= \textstyle\sum_C \sum_D P(A,B,C,D) \\
&= \textstyle\sum_C \sum_D \frac{1}{Z}\phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A) \\
&\propto \textstyle\sum_C \sum_D \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A) \\
&= \phi_1(A,B)\textstyle\sum_C \phi_2(B,C)\sum_D \phi_3(C,D)\phi_4(D,A) \\
&= \phi_1(A,B)\textstyle\sum_C \phi_2(B,C)\sum_D \Phi_1(C,D,A) \\
&= \phi_1(A,B)\textstyle\sum_C \phi_2(B,C)\tau_1(C,A) \\
&= \phi_1(A,B)\textstyle\sum_C \Phi_2(C,A,B) \\
&= \phi_1(A,B)\tau_2(A,B)
\end{aligned}
$$



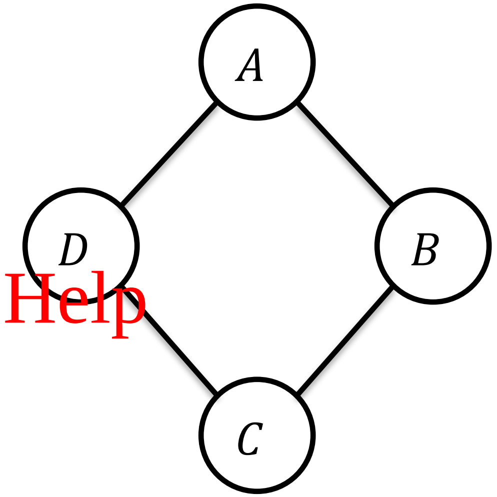
A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

A	B	$\tau_2(A,B)$
a	b	30001
a	$\bar{b}$	1000300
$\bar{a}$	b	1000300
$\bar{a}$	$\bar{b}$	30001

Variable Elimination

- We start with the Gibbs distribution

$$P(A,B) = \sum_C \sum_D P(A,B,C,D)$$
$$= \sum_C \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$
$$\propto \sum_C \sum_D \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C) \sum_D \phi_3(C,D)\phi_4(D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C) \sum_D \Phi_1(C,D,A)$$
$$= \phi_1(A,B) \sum_C \phi_2(B,C)\tau_1(C,A)$$
$$= \phi_1(A,B) \sum_C \Phi_2(C,A,B)$$
$$= \phi_1(A,B)\tau_2(A,B)$$
$$= \Phi_3(A,B)$$

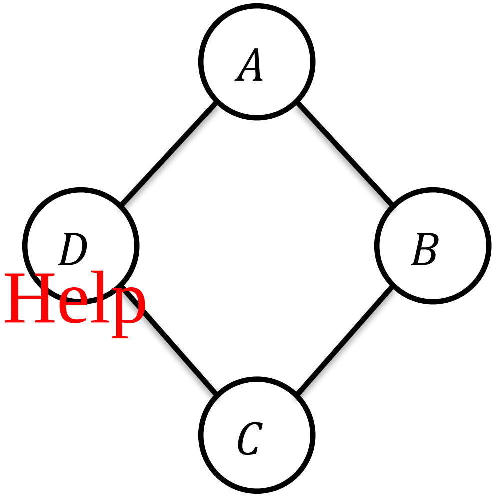


A	B	$\Phi_3(A,B)$
a	b	900030
a	$\bar{b}$	5001500
$\bar{a}$	b	1000300
$\bar{a}$	$\bar{b}$	300010

Variable Elimination

- We start with the Gibbs distribution

$$
\begin{aligned}
P(A,B) &= \textstyle\sum_C \sum_D P(A,B,C,D) \\
&= \textstyle\sum_C \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A) \\
&\propto \textstyle\sum_C \sum_D \phi_1(A,B)\phi_2(B,C)\phi_3(C,D)\phi_4(D,A) \\
&= \phi_1(A,B) \textstyle\sum_C \phi_2(B,C) \sum_D \phi_3(C,D)\phi_4(D,A) \\
&= \phi_1(A,B) \textstyle\sum_C \phi_2(B,C) \sum_D \Phi_1(C,D,A) \\
&= \phi_1(A,B) \textstyle\sum_C \phi_2(B,C)\tau_1(C,A) \\
&= \phi_1(A,B) \textstyle\sum_C \Phi_2(C,A,B) \\
&= \phi_1(A,B)\tau_2(A,B) \\
&= \Phi_3(A,B)
\end{aligned}
$$



A	B	$\Phi_3(A,B)$
a	b	900030
a	$\bar{b}$	5001500
$\bar{a}$	b	1000300
$\bar{a}$	$\bar{b}$	300010

A	B	$P(A,B)$
a	b	.13
a	$\bar{b}$	.69
$\bar{a}$	b	.14
$\bar{a}$	$\bar{b}$	.04

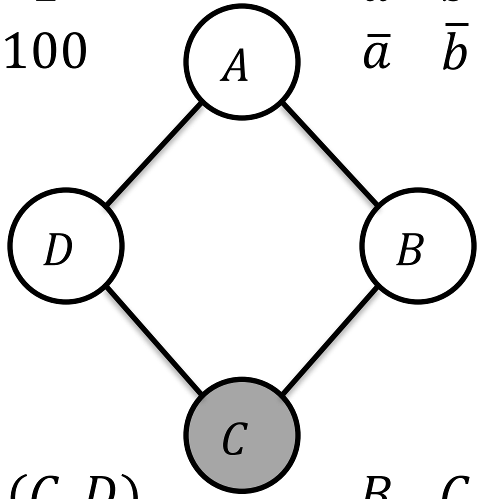
- We need to normalise the results to get $P(A,B)$
- Differently from BN, MN are not naturally normalised

Variable Elimination with Evidence

- Let us now consider computing a query with evidence such as $P(B|c = true)$ using VE
  - We start by setting evidence by eliminating the rows that do not match the evidence

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



C	D	$\phi_3(C,D)$
c	d	1
c	$\bar{d}$	100
$\bar{c}$	d	100
$\bar{c}$	$\bar{d}$	1

B	C	$\phi_2(B,C)$
b	c	100
b	$\bar{c}$	1
$\bar{b}$	c	1
$\bar{b}$	$\bar{c}$	100

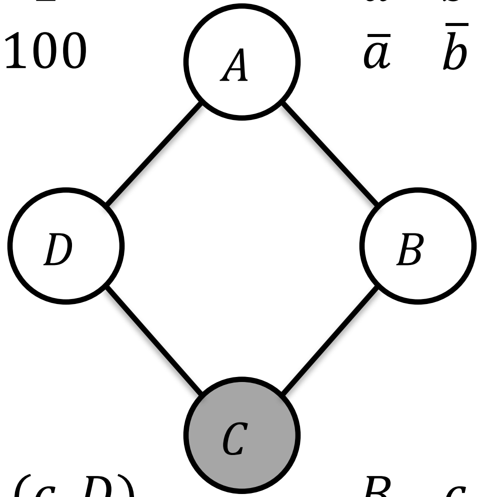
Variable Elimination with Evidence

- Let us now consider computing a query with evidence such as $P(B|c = true)$ using VE
  - We start by setting evidence by eliminating the rows that do not match the evidence
- Again, we start with the Gibbs distribution

$$
\begin{aligned}
P(B,c) &= \textstyle\sum_A \sum_D P(A,B,c,D) \\
&= \textstyle\sum_A \sum_D \frac{1}{Z}\phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&\propto \textstyle\sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c)\textstyle\sum_A \phi_1(A,B)\sum_D \phi_3(c,D)\phi_4(D,A)
\end{aligned}
$$

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10



c	D	$\phi_3(c,D)$
c	d	1
c	$\bar{d}$	100

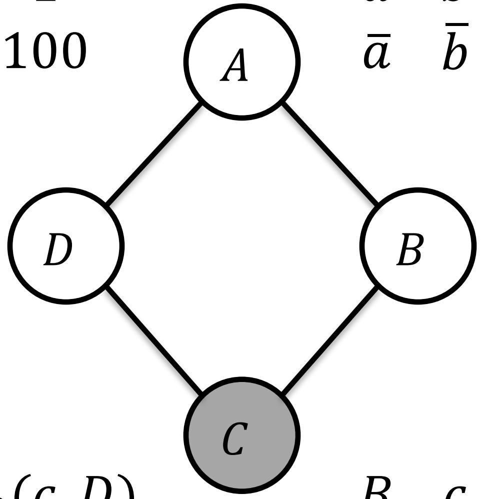
B	c	$\phi_2(B,c)$
b	c	100
$\bar{b}$	c	1

Variable Elimination with Evidence

D	A	$\phi_4(D,A)$
d	a	100
d	$\bar{a}$	1
$\bar{d}$	a	1
$\bar{d}$	$\bar{a}$	100

A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

c	D	A	$\Phi_1(c,D,A)$
c	d	a	100
c	d	$\bar{a}$	1
c	$\bar{d}$	a	100
c	$\bar{d}$	$\bar{a}$	10000



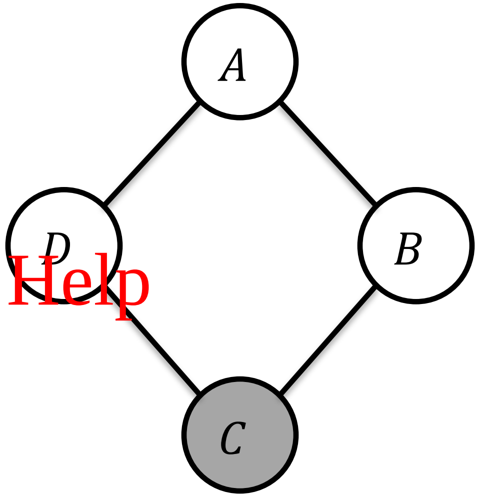
c	D	$\phi_3(c,D)$
c	d	1
c	$\bar{d}$	100

B	c	$\phi_2(B,c)$
b	c	100
$\bar{b}$	c	1

Variable Elimination with Evidence

- Let us now consider computing a query with evidence such as $P(B|c = true)$ using VE
  - We start by setting evidence by eliminating the rows that do not match the evidence
- Again, we start with the Gibbs distribution

$$P(B,c) = \sum_A \sum_D P(A,B,c,D)$$
$$= \sum_A \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A)$$
$$\propto \sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A)$$
$$= \phi_2(B,c) \sum_A \phi_1(A,B) \sum_D \phi_3(c,D)\phi_4(D,A)$$
$$= \phi_2(B,c) \sum_A \phi_1(A,B) \sum_D \Phi_1(c,D,A)$$



A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

c	D	A	$\Phi_1(c,D,A)$
c	d	a	100
c	d	$\bar{a}$	1
c	$\bar{d}$	a	100
c	$\bar{d}$	$\bar{a}$	10000

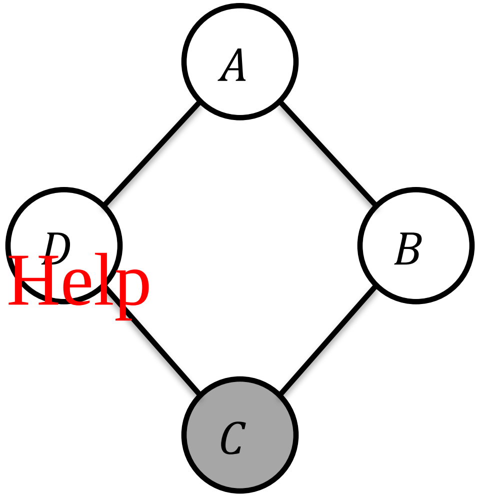
B	c	$\phi_2(B,c)$
b	c	100
$\bar{b}$	c	1

Variable Elimination with Evidence

- Let us now consider computing a query with evidence such as $P(B|c = true)$ using VE
  - We start by setting evidence by eliminating the rows that do not match the evidence

- Again, we start with the Gibbs distribution

$$
\begin{aligned}
P(B,c) &= \textstyle\sum_A \sum_D P(A,B,c,D) \\
&= \textstyle\sum_A \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&\propto \textstyle\sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \Phi_1(c,D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B)\tau_1(c,A)
\end{aligned}
$$



A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

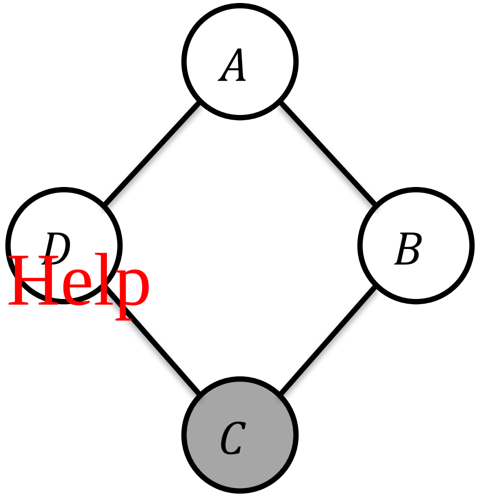
c	A	$\tau_1(c,A)$
c	a	200
c	$\bar{a}$	10001

B	c	$\phi_2(B,c)$
b	c	100
$\bar{b}$	c	1

Variable Elimination with Evidence

- Again, we start with the Gibbs distribution

$$
\begin{aligned}
P(B,c) &= \textstyle\sum_A \sum_D P(A,B,c,D) \\
&= \textstyle\sum_A \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&\propto \textstyle\sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \Phi_1(c,D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B)\tau_1(c,A) \\
&= \phi_2(B,c) \textstyle\sum_A \Phi_2(c,A,B)
\end{aligned}
$$



A	B	$\phi_1(A,B)$
a	b	30
a	$\bar{b}$	5
$\bar{a}$	b	1
$\bar{a}$	$\bar{b}$	10

c	A	B	$\Phi_2(c,A,B)$
c	a	b	6000
c	a	$\bar{b}$	1000
c	$\bar{a}$	b	10001
c	$\bar{a}$	$\bar{b}$	100010

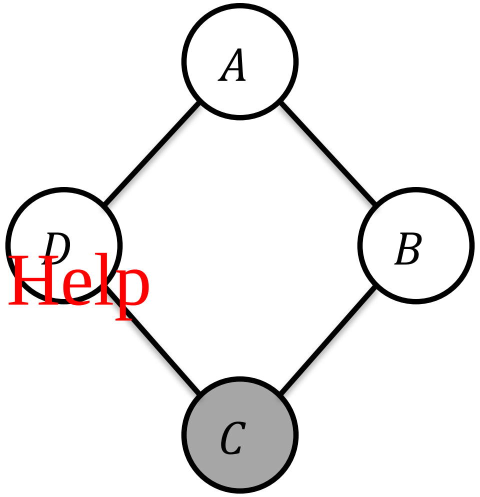
c	A	$\tau_1(c,A)$
c	a	200
c	$\bar{a}$	10001

B	c	$\phi_2(B,c)$
b	c	100
$\bar{b}$	c	1

Variable Elimination with Evidence

- Again, we start with the Gibbs distribution

$$
\begin{aligned}
P(B,c) &= \textstyle\sum_A \sum_D P(A,B,c,D) \\
&= \textstyle\sum_A \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&\propto \textstyle\sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \Phi_1(c,D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B)\tau_1(c,A) \\
&= \phi_2(B,c) \textstyle\sum_A \Phi_2(c,A,B)
\end{aligned}
$$



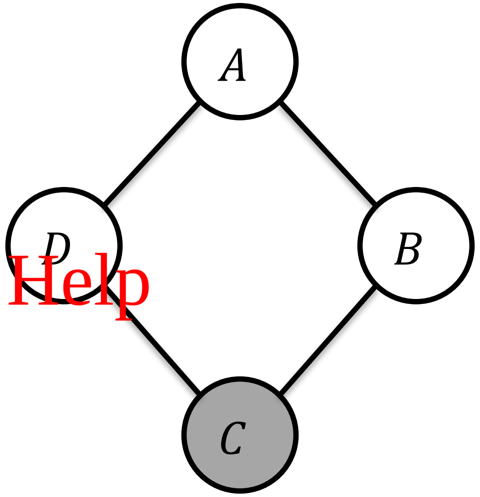
c	A	B	$\Phi_2(c,A,B)$
c	a	b	6000
c	a	$\bar{b}$	1000
c	$\bar{a}$	b	10001
c	$\bar{a}$	$\bar{b}$	100010

B	c	$\phi_2(B,c)$
b	c	100
$\bar{b}$	c	1

Variable Elimination with Evidence

- Again, we start with the Gibbs distribution

$$
\begin{aligned}
P(B,c) &= \textstyle\sum_A \sum_D P(A,B,c,D) \\
&= \textstyle\sum_A \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&\propto \textstyle\sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \Phi_1(c,D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B)\tau_1(c,A) \\
&= \phi_2(B,c) \textstyle\sum_A \Phi_2(c,A,B) \\
&= \phi_2(B,c)\tau_2(c,B)
\end{aligned}
$$



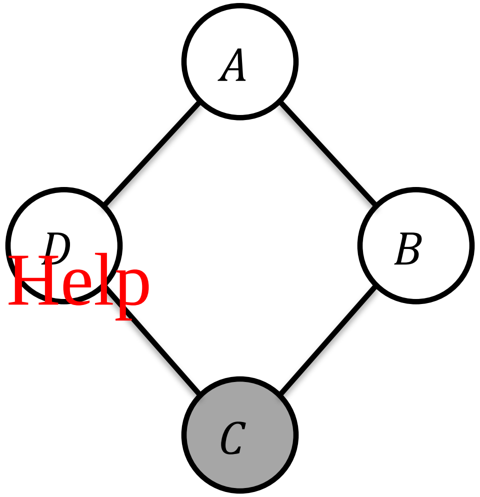
c	B	$\tau_2(c,B)$
c	b	16001
c	$\bar{b}$	101010

B	c	$\phi_2(B,c)$
b	c	100
$\bar{b}$	c	1

Variable Elimination with Evidence

- Again, we start with the Gibbs distribution

$$
\begin{aligned}
P(B,c) &= \textstyle\sum_A \sum_D P(A,B,c,D) \\
&= \textstyle\sum_A \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&\propto \textstyle\sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B) \sum_D \Phi_1(c,D,A) \\
&= \phi_2(B,c) \textstyle\sum_A \phi_1(A,B)\tau_1(c,A) \\
&= \phi_2(B,c) \textstyle\sum_A \Phi_2(c,A,B) \\
&= \phi_2(B,c)\tau_2(c,B) \\
&= \Phi_3(c,B)
\end{aligned}
$$

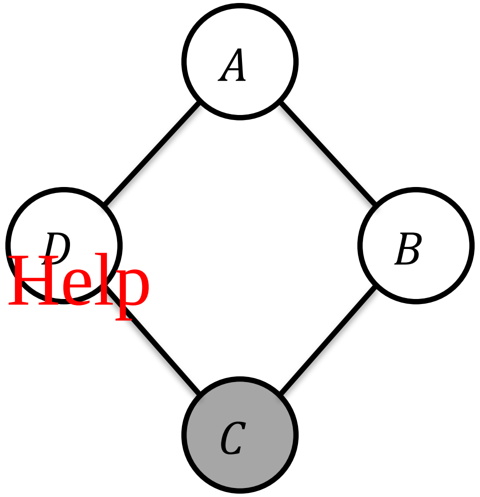


c	B	$\Phi_3(c,B)$
c	b	16001
c	$\bar{b}$	101010

Variable Elimination with Evidence

- Again, we start with the Gibbs distribution

$$
\begin{aligned}
P(B,c) &= \sum_A \sum_D P(A,B,c,D) \\
&= \sum_A \sum_D \frac{1}{Z} \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&\propto \sum_A \sum_D \phi_1(A,B)\phi_2(B,c)\phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \sum_A \phi_1(A,B) \sum_D \phi_3(c,D)\phi_4(D,A) \\
&= \phi_2(B,c) \sum_A \phi_1(A,B) \sum_D \Phi_1(c,D,A) \\
&= \phi_2(B,c) \sum_A \phi_1(A,B) \tau_1(c,A) \\
&= \phi_2(B,c) \sum_A \Phi_2(c,A,B) \\
&= \phi_2(B,c) \tau_2(c,B) \\
&= \Phi_3(c,B)
\end{aligned}
$$



c	B	$\Phi_3(c,B)$
c	b	1600100
c	$\bar{b}$	101010

c	B	$P(B\|c)$
c	b	.94
c	$\bar{b}$	.06

- After normalisation, we get

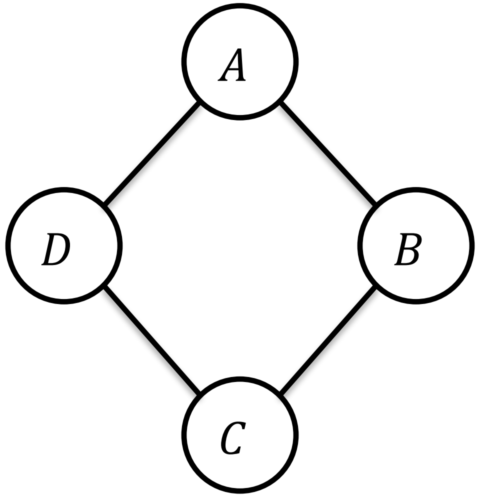
Variable Elimination with Energy Functions

- We start with the Gibbs distribution

$$P(A,B) = \sum_C \sum_D P(A,B,C,D)$$
$$= \sum_C \sum_D \frac{1}{Z} \exp(-(\psi_1(A,B) + \psi_2(B,C) + \psi_3(C,D) + \psi_4(D,A)))$$
$$\propto \sum_C \sum_D \exp(-(\psi_1(A,B) + \psi_2(B,C) + \psi_3(C,D) + \psi_4(D,A)))$$
$$= \sum_C \sum_D \exp(-\psi_1(A,B))\exp(-\psi_2(B,C))\exp(-\psi_3(C,D))\exp(-\psi_4(D,A))$$
$$= \phi_1 \exp\big(-\psi_1(A,B)\big) \sum_C \exp(-\psi_2(B,C)) \sum_D \exp(-\psi_3(C,D))\exp(-\psi_4(D,A))$$

A	B	$\psi_1(A,B)$
a	b	-3.40
a	$\bar{b}$	-1.61
$\bar{a}$	b	0
$\bar{a}$	$\bar{b}$	-2.30

B	C	$\psi_2(B,C)$
b	c	-4.61
b	$\bar{c}$	0
$\bar{b}$	c	0
$\bar{b}$	$\bar{c}$	-4.61

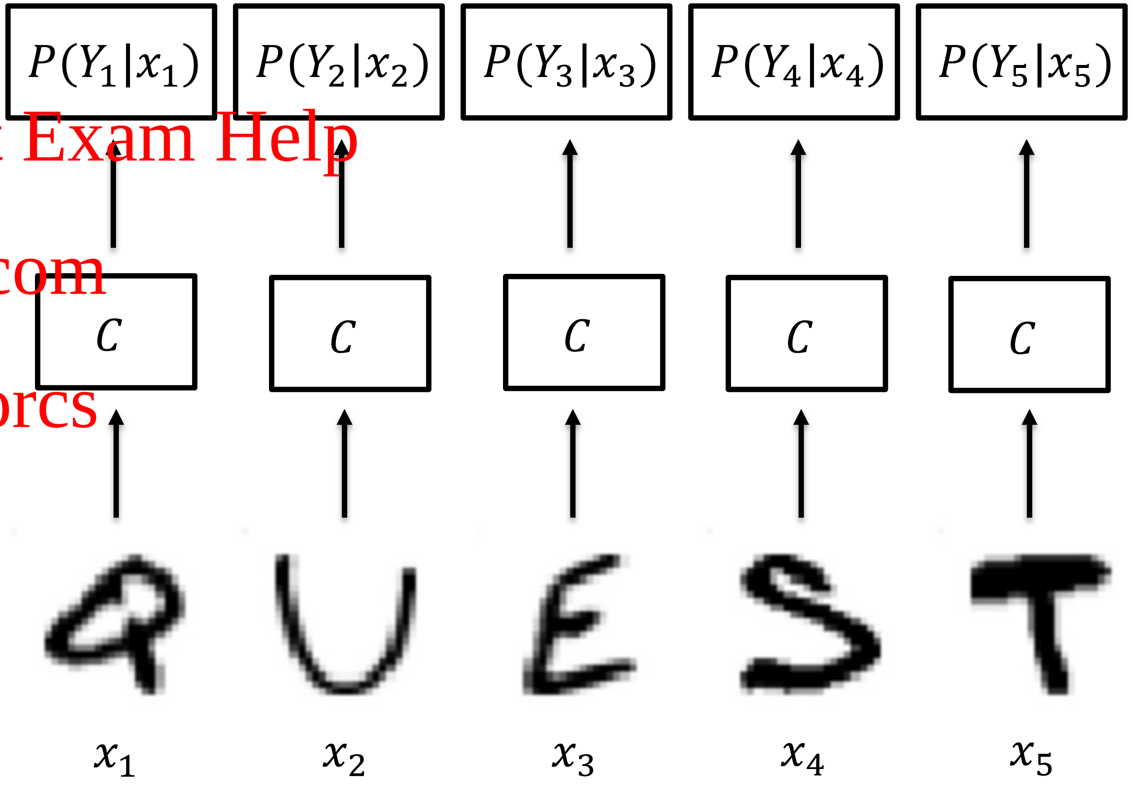


C	D	$\psi_3(C,D)$
c	d	0
c	$\bar{d}$	-4.61
$\bar{c}$	d	-4.61
$\bar{c}$	$\bar{d}$	0

D	A	$\psi_4(D,A)$
d	a	-4.61
d	$\bar{a}$	0
$\bar{d}$	a	0
$\bar{d}$	$\bar{a}$	-4.61

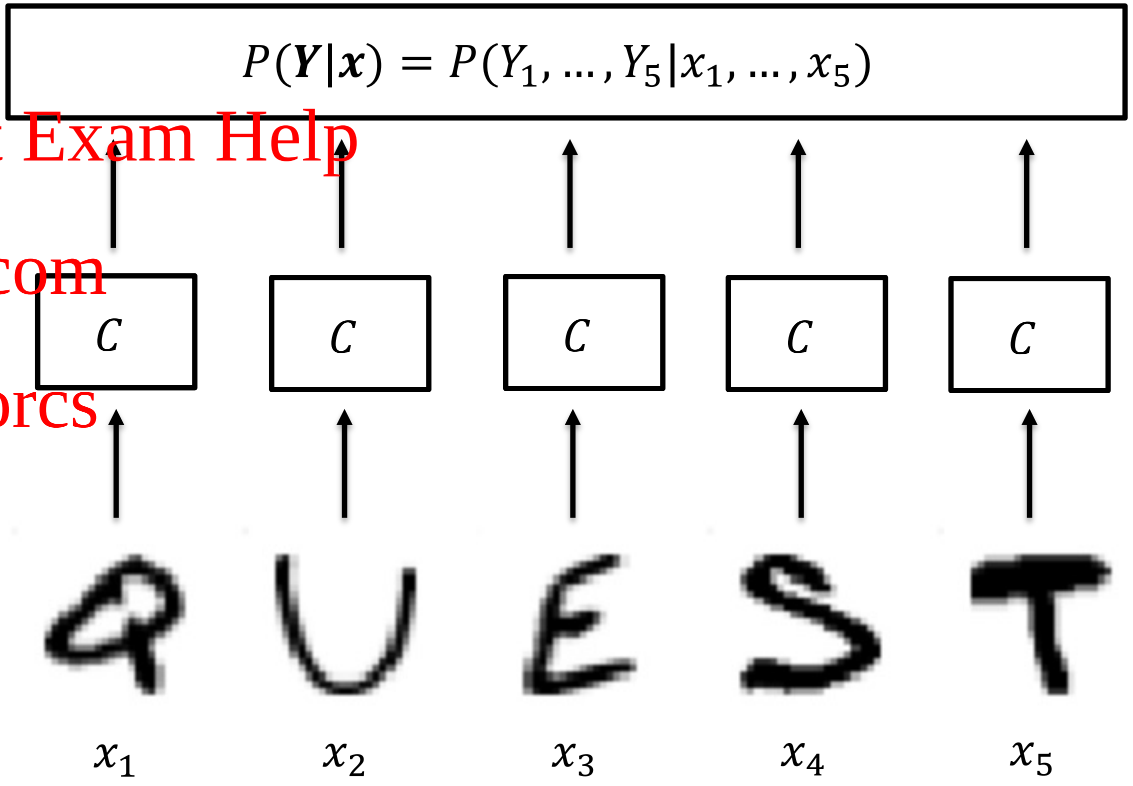
Conditional Random Fields (CRFs)

- Suppose we want to use Machine Learning classifiers to recognise handwritten words
  - We can train a classifier C that takes as input an image of a single letter x
  - C outputs a class probability $P(Y|x)$ or a score that is proportional to the classifier confidence
- Given an input sequence (word) $x_1, \ldots, x_n$
  - We can call the classifier C n times and obtain n independent predictions $P(Y_i|x_i)$
  - However, this approach does use the information that some sequences of letters may be very unlikely
  - For instance, we expect that "QU" is much more common than "QV"



Conditional Random Fields (CRFs)

- A *conditional random field* (CRF) is a discriminative model
  - In this example, it will directly approximate $P(\boldsymbol{Y}|\boldsymbol{x}) = P(Y_1, \dots, Y_n | x_1, \dots, x_n)$
  - So far, we have only studied *generative models* (more about this later)

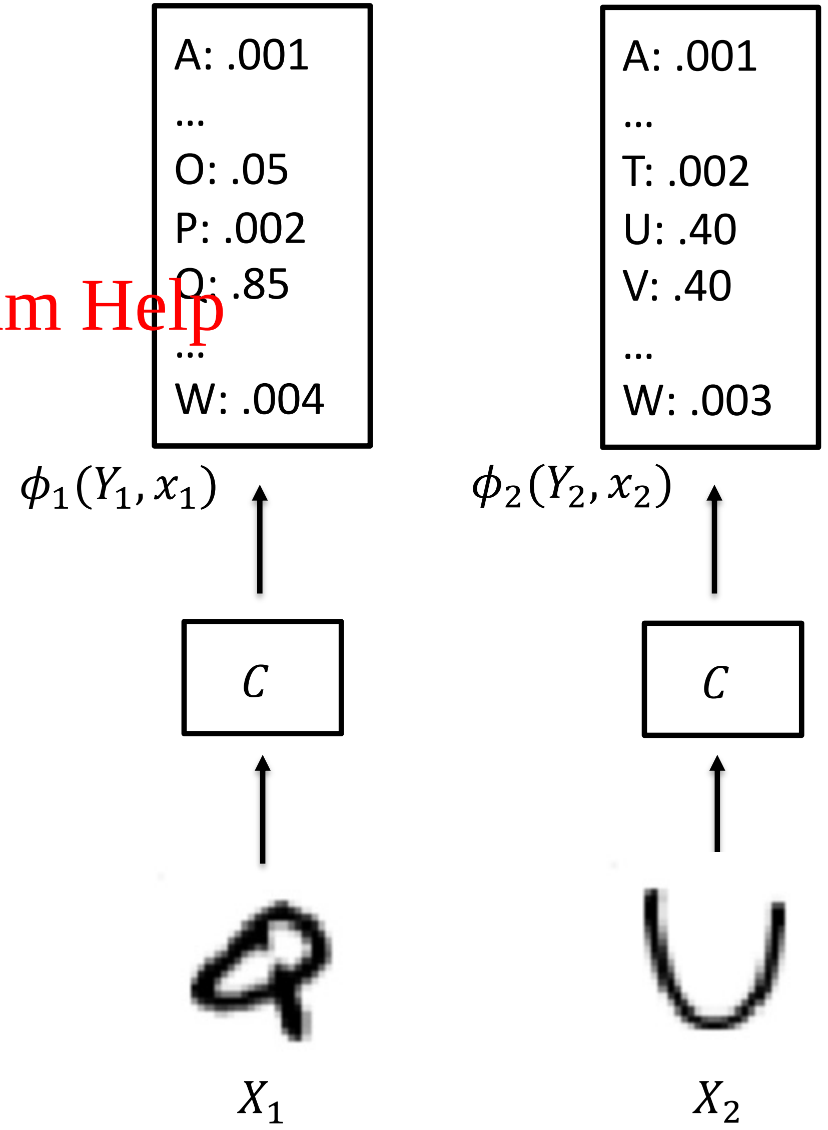


- With independent classifiers, the probability of classifying a given input $\boldsymbol{x}$ with n letters is simply

$$P(Y_1, \dots, Y_n | \boldsymbol{x}) = \prod_i P(Y_i | x_i)$$

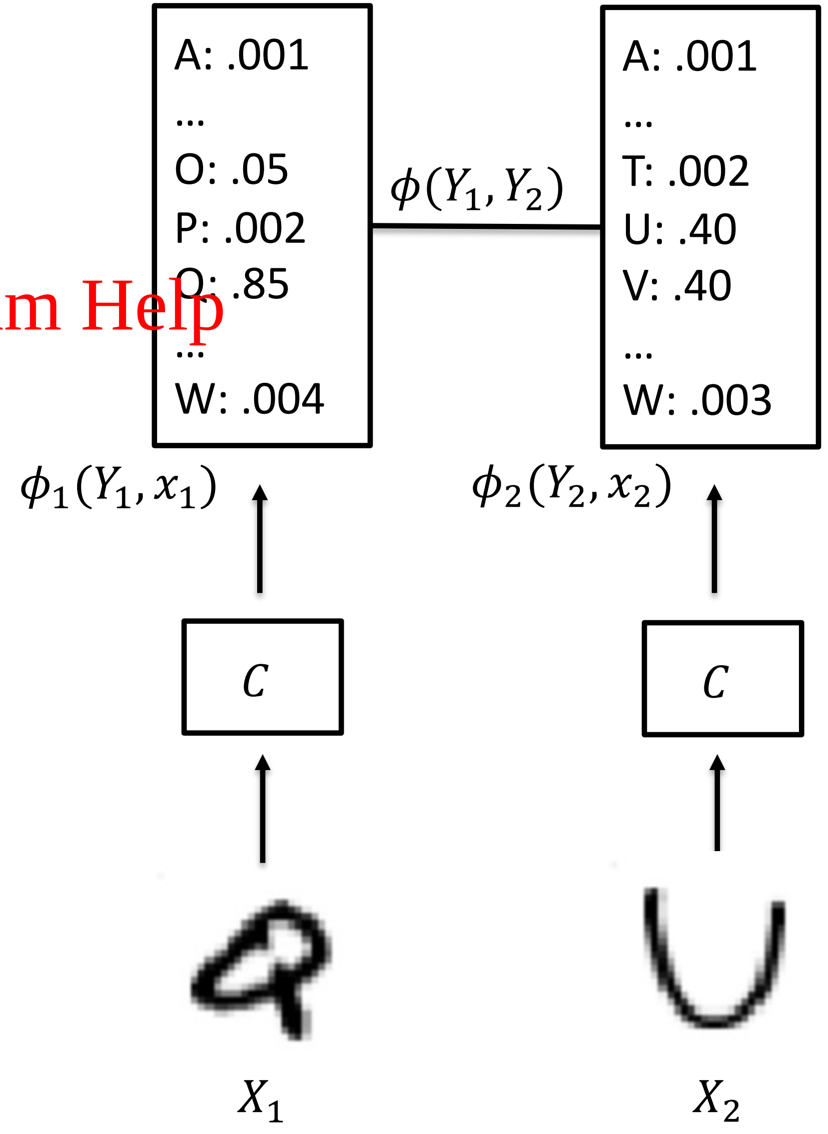
Conditional Random Fields (CRFs)

- We can see the output of the classifiers as factors
  - $\phi_i(Y_i, x_i)$ is the score of the classifier
  - It assigns higher values to y_i's that are consistent with the input x_i



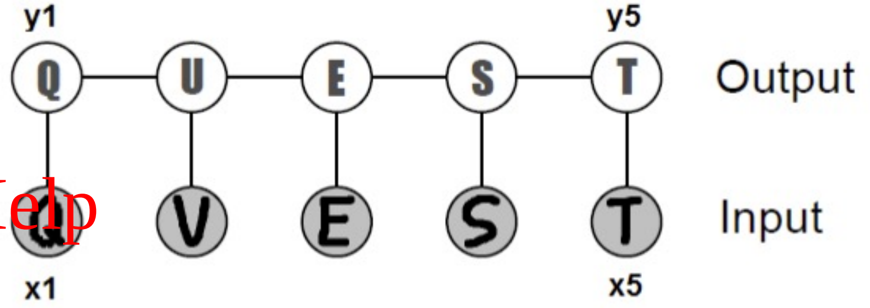
Conditional Random Fields (CRFs)

- We can see the output of the classifiers as factors
  - $\phi_i(Y_i, x_i)$ is the score of the classifier
  - It assigns higher values to y_i's that are consistent with the input x_i
- We can add a new pairwise factor for consecutive letters
  - $\phi(Y_i, Y_{i+1})$ is a measure of co-occurrence of consecutive letters
  - It measures the affinity between y values



Conditional Random Fields (CRFs)

- Therefore, this problem can be modelled by the graph shown on the right
  - It is known as the *linear chain CRF*
  - It is an undirected version of the Hidden Markov Model

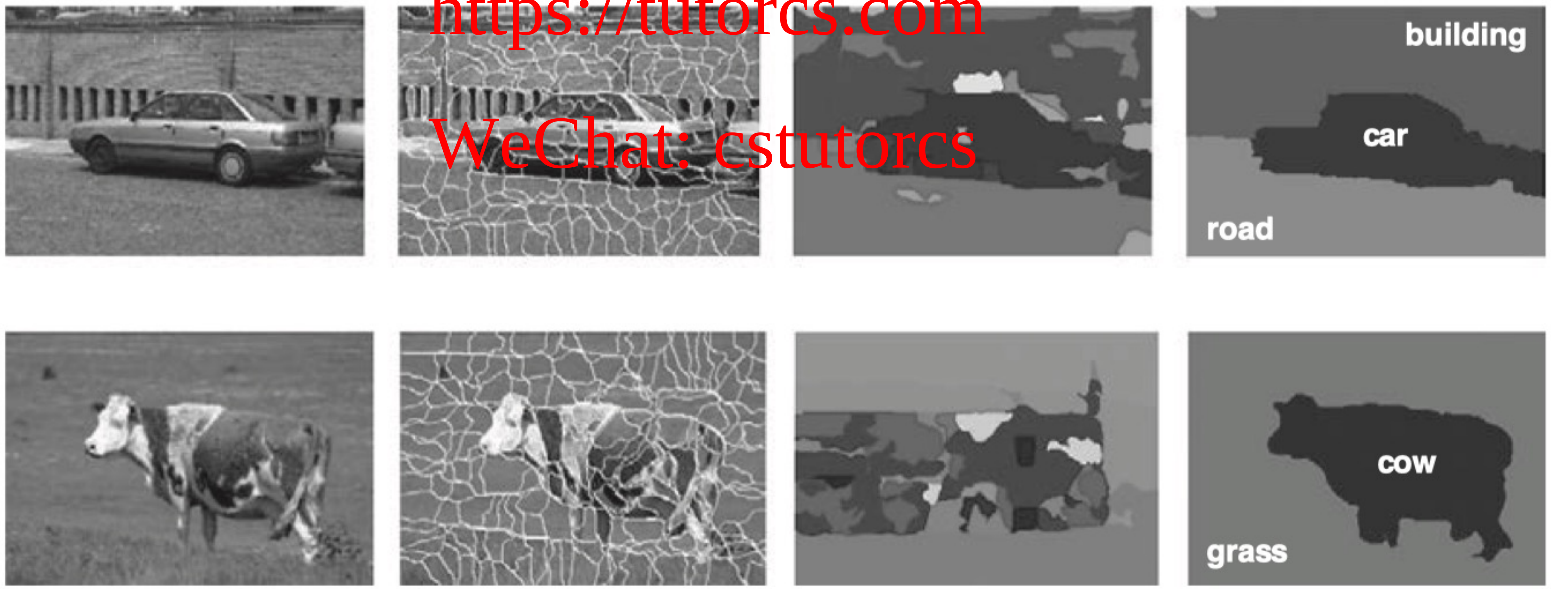


- In this application, we want to know the most probable instantiation
  - MAP or MPE query
  - The output is a sequence of letters that corresponds to the assignment with the highest probability
  - The answer is efficiently computed by the Viterbi algorithm

$$P(\boldsymbol{Y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \phi_1(Y_1, x_1) \prod_{i=2} \phi_i(Y_i, x_i) \phi(Y_{i-1}, Y_i)$$

Conditional Random Fields (CRFs)

- Structured (output) learning
  - Techniques that involves predicting structured objects, rather than scalar discrete or real values
  - CRF graph can be as complex as necessary



Original | Segmented | Independent classifiers | CRFs

Generative and Discriminative Models

- In this course, we have discussed several generative models
  - Markov chains, Hidden Markov models, Bayesian networks, Markov networks are examples of generative models
  - They model $P(\boldsymbol{X})$ being $\boldsymbol{X}$ a set of variables that correspond to graph nodes
  - As these models estimate $P(\boldsymbol{X})$, they can be used to answer any queries that involve variables in $\boldsymbol{X}$
- However, most of the Machine Learning algorithms are discriminative
  - Discriminative models approximate $P(Y|\boldsymbol{X})$
  - These models can only answer queries that involve estimating the probability of Y given $\boldsymbol{X}$, such as in the case of classification
- Generative models can be used in classification tasks
  - We pick one variable as class attribute (Y) and compute $P(Y|\boldsymbol{X})$ from $P(Y, \boldsymbol{X})$
  - But, in this case, which model is better? Generative or discriminative?

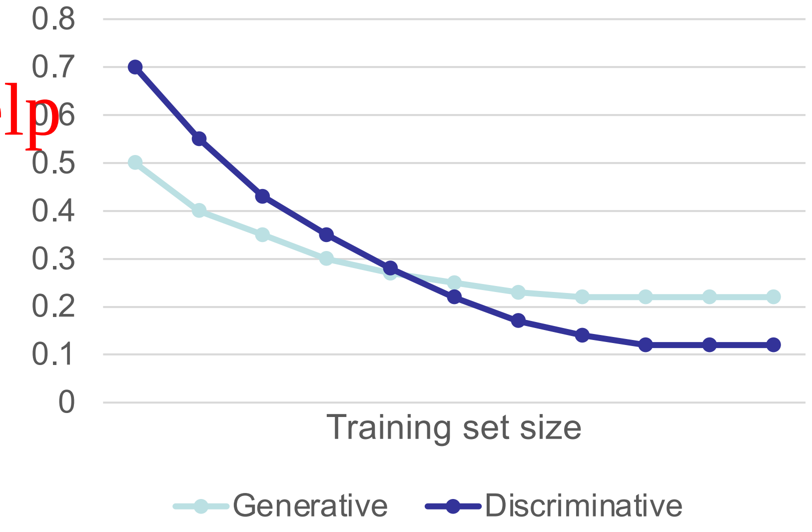
Generative and Discriminative Models

- Generative models are particularly useful when missing data is present
  - We can leave the attributes with missing data as unobserved as run inference
  - Several discriminative models require complete data

- However, the prevailing consensus is that discriminative models are preferred for classification tasks
  - “Discriminative models have lower generalization error”
  - “Discriminative models need less data to train”

Generative and Discriminative Models

- This paper compares a generative-discriminative pair
  - Naïve Bayes and logistic regression
  - The generative model has indeed a higher asymptotic error as the training set grows
  - However, it approaches its asymptotic error much faster than the discriminative model



- Therefore, we can observe two regimes of performance
  - For smaller datasets, the generative model has already approached its asymptotic error and is performing better
  - For larger datasets, the discriminative model approaches its lower asymptotic error and performs better

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (pp. 841-848).

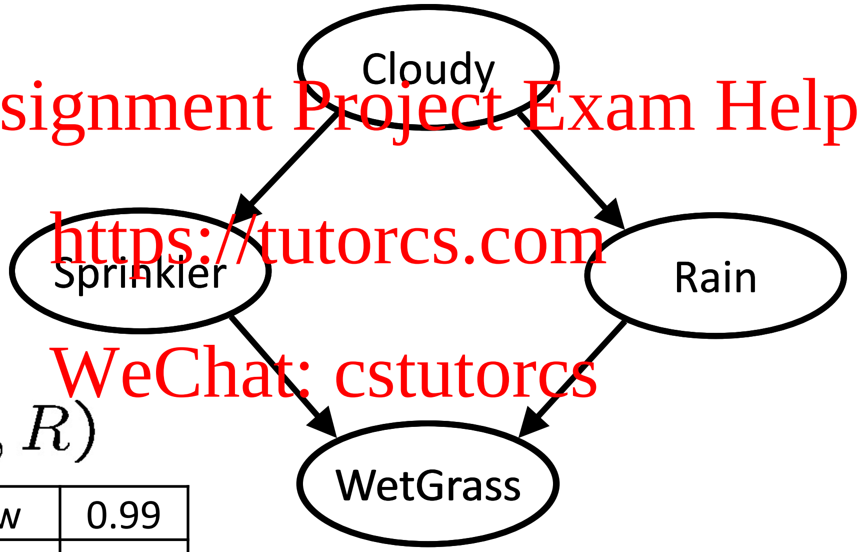
Generative Models and Synthetic Data

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Conclusion

- Markov networks are undirected probabilistic graphical models
  - These models are widespread in areas such as image and language processing
  - The dependency between variables do not have an intrinsic direction

- Several tasks in image processing involve the computation of a MAP or MPE assignment
  - It is known as the MAP-MRF approach

  - As images involve a large number of variables and have large treewidth. This task requires specialised approximate inference methods

- Variable elimination works for Markov networks
  - Most of the algorithms were designed for MN and involve transforming the BN to an MN
  - VE is one case, the interaction graph is an MN
- CRFs are popular discriminative approaches
  - Frequently used in structured output prediction tasks