

Probabilistic Graphical Models

CMSC 678

UMBC

Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \dots, X_N$

Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \dots, X_N$

Graph $G = (\text{vertices } V, \text{edges } E)$

Distribution $p(X_1, \dots, X_N)$

Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \dots, X_N$

Graph $G = (\text{vertices } V, \text{edges } E)$

Distribution $p(X_1, \dots, X_N)$

Vertices $\leftrightarrow$ random variables

Edges show dependencies among random variables

Probabilistic Graphical Models

A graph G that represents a probability distribution over random variables $X_1, \dots, X_N$

Graph $G = (\text{vertices } V, \text{edges } E)$

Distribution $p(X_1, \dots, X_N)$

Vertices $\leftrightarrow$ random variables

Edges show dependencies among random variables

Two main flavors: *directed* graphical models and *undirected* graphical models

Outline

Directed Graphical Models

Naïve Bayes

Undirected Graphical Models

Factor Graphs

Ising Model

Message Passing: Graphical Model Inference

Directed Graphical Models

A *directed* (acyclic) graph $G=(V,E)$ that represents a probability distribution over random variables

$$X_1, \dots, X_N$$

Joint probability factorizes into factors of X_i
conditioned on the parents of X_i

Directed Graphical Models

A *directed* (acyclic) graph $G=(V,E)$ that represents a probability distribution over random variables

$$X_1, \dots, X_N$$

Joint probability factorizes into factors of X_i conditioned on the parents of X_i

Benefit: read the independence properties are *transparent*

Directed Graphical Models

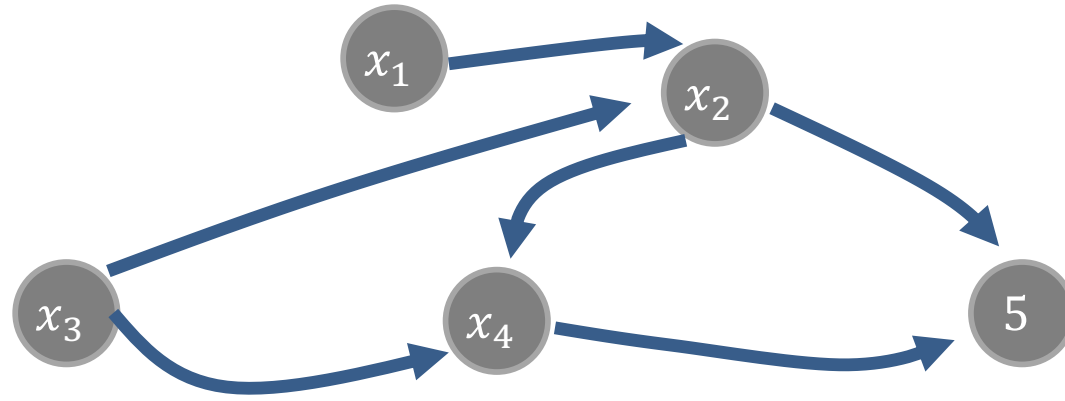
A *directed* (acyclic) graph $G=(V,E)$ that represents a probability distribution over random variables

$$X_1, \dots, X_N$$

Joint probability factorizes into factors of X_i conditioned on the parents of X_i

A graph/joint distribution that follows this is a **Bayesian network**

Bayesian Networks: Directed Acyclic Graphs

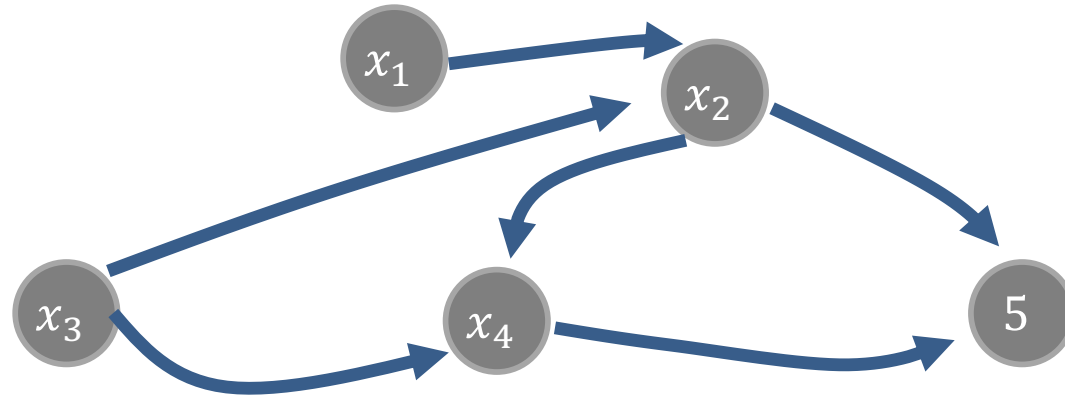


$$p(x_1, x_2, x_3, \dots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

topological
sort

“parents of”

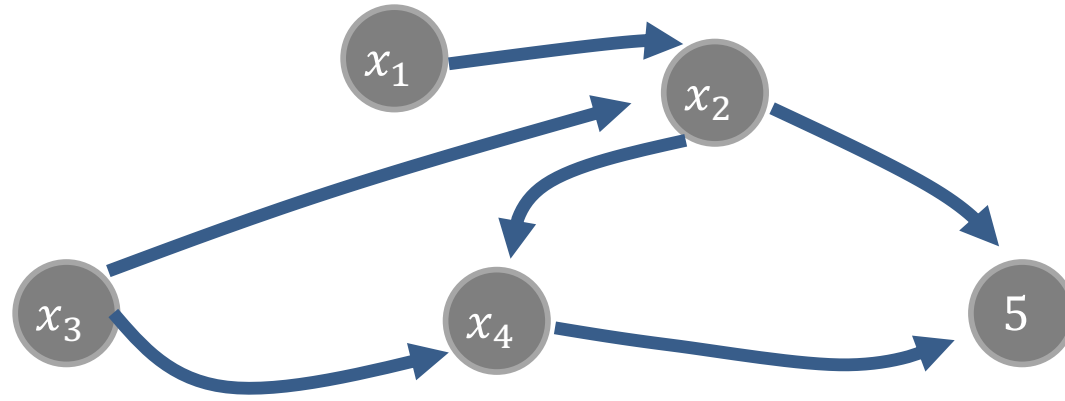
Bayesian Networks: Directed Acyclic Graphs



$$p(x_1, x_2, x_3, \dots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

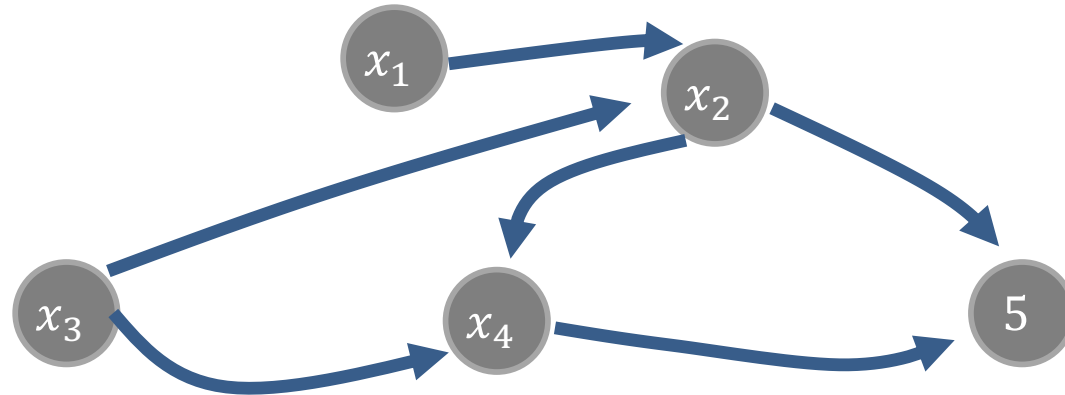
$$p(x_1, x_2, x_3, x_4, x_5) = ???$$

Bayesian Networks: Directed Acyclic Graphs



$$p(x_1, x_2, x_3, x_4, x_5) =$$
$$p(x_1)p(x_3)p(x_2|x_1, x_3)p(x_4|x_2, x_3)p(x_5|x_2, x_4)$$

Bayesian Networks: Directed Acyclic Graphs

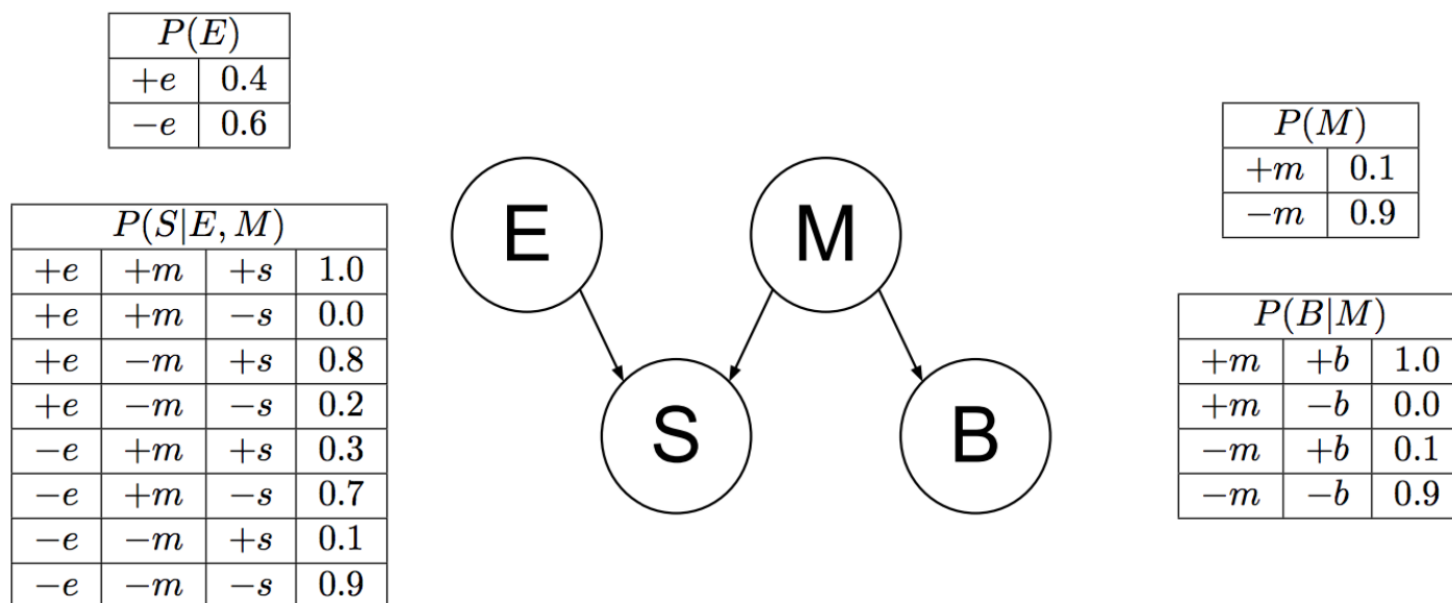


$$p(x_1, x_2, x_3, \dots, x_N) = \prod_i p(x_i \mid \pi(x_i))$$

exact inference in general DAGs is NP-hard

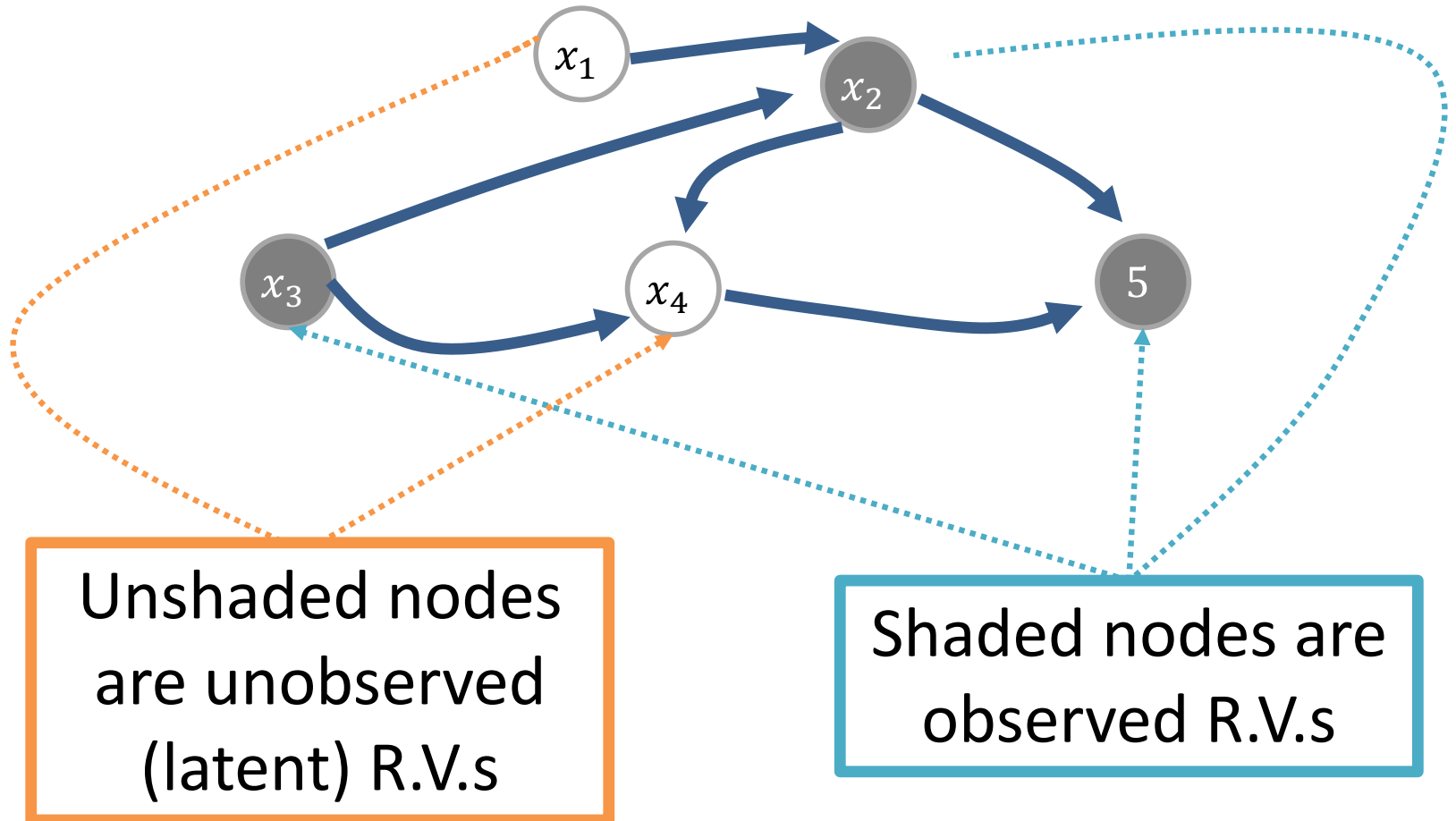
inference in trees can be exact

A smell of sulphur (S) can be caused either by rotten eggs (E) or as a sign of the doom brought by the Mayan Apocalypse (M). The Mayan Apocalypse also causes the oceans to boil (B). The Bayesian network and corresponding conditional probability tables for this situation are shown below. The notation $+x$ means that x is true, and $-x$ means that x is false. For each part, you should give either a numerical answer (e.g., 0.81) or an arithmetic expression in terms of numbers from the tables below (e.g., $0.9 * 0.9$). Note that the latter is easier and perfectly OK.



- (A) (X points) Compute the following entry from the joint distribution: $P(-e, -s, -m, -b)$
- (B) (X points) What is the probability that the oceans boil?

Directed Graphical Model Notation

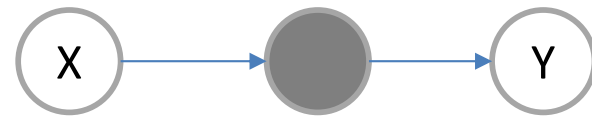


D-Separation: Testing for Conditional Independence

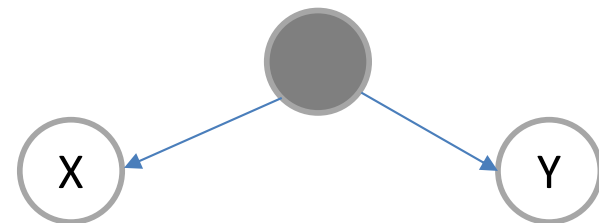
d-separation

X & Y are d-separated if for **all** paths P, one of the following is true:

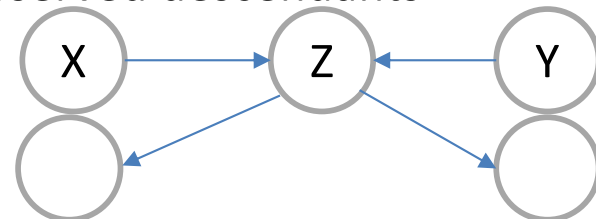
P has a chain with an observed middle node



P has a fork with an observed parent node



P includes a “v-structure” or “collider” with all unobserved descendants



Variables X & Y are conditionally independent given Z if all (undirected) paths from (any variable in) X to (any variable in) Y are **d-separated** by Z

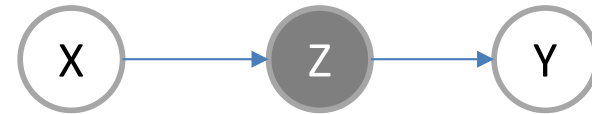
D-Separation: Testing for Conditional Independence

Variables X & Y are conditionally independent given Z if all (undirected) paths from (any variable in) X to (any variable in) Y are **d-separated** by Z

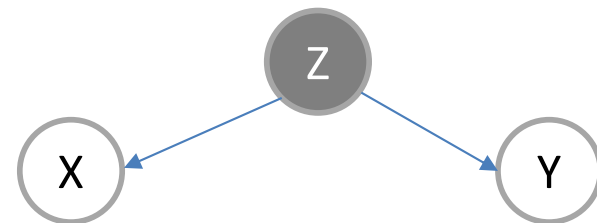
d-separation

X & Y are d-separated if for **all** paths P, one of the following is true:

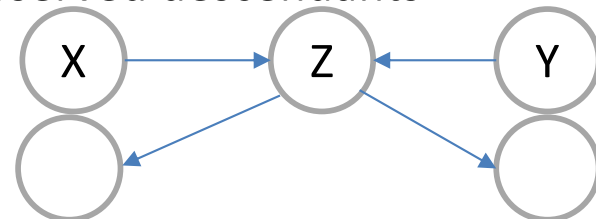
P has a chain with an observed middle node



P has a fork with an observed parent node



P includes a “v-structure” or “collider” with all unobserved descendants



*observing Z blocks
the path from X to Y*

*observing Z blocks
the path from X to Y*

*not observing Z blocks
the path from X to Y*

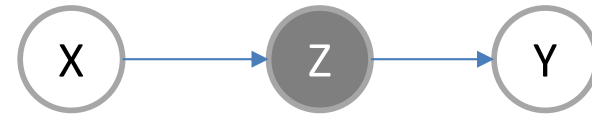
D-Separation: Testing for Conditional Independence

Variables X & Y are conditionally independent given Z if all (undirected) paths from (any variable in) X to (any variable in) Y are **d-separated** by Z

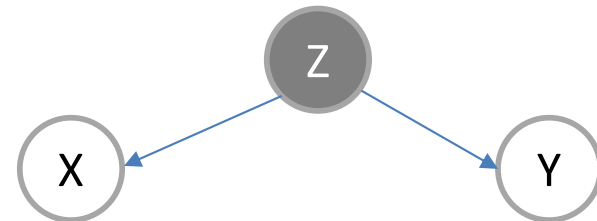
d-separation

X & Y are d-separated if for **all** paths P, one of the following is true:

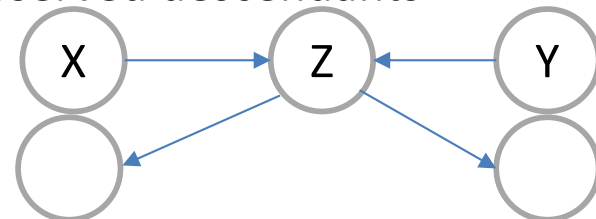
P has a chain with an observed middle node



P has a fork with an observed parent node



P includes a “v-structure” or “collider” with all unobserved descendants



observing Z blocks the path from X to Y

observing Z blocks the path from X to Y

not observing Z blocks the path from X to Y

$$p(x, y, z) = p(x)p(y)p(z|x, y)$$

$$p(x, y) = \sum_z p(x)p(y)p(z|x, y) = p(x)p(y)$$

Outline

Directed Graphical Models

Naïve Bayes

Undirected Graphical Models

Factor Graphs

Ising Model

Message Passing: Graphical Model Inference

Naïve Bayes

$$\operatorname{argmax}_Y p(Y | X)$$



Apply Bayes rule and take logs

$$\operatorname{argmax}_Y \log p(X | Y) + \log p(Y)$$

likelihood

prior

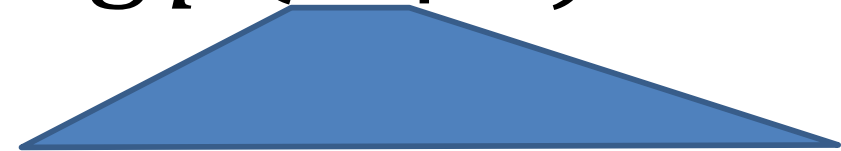
Naïve Bayes

$$\operatorname{argmax}_Y p(Y | X)$$



Apply Bayes rule and take logs

$$\operatorname{argmax}_Y \log p(X | Y) + \log p(Y)$$



Represent X is a D-dimensional
vector (of features):

$$X = (X_1, X_2, X_3, \dots, X_D)$$

Naïve Bayes

$$\operatorname{argmax}_Y p(Y | X)$$



Apply Bayes rule and take logs

$$\operatorname{argmax}_Y \log p(X | Y) + \log p(Y)$$



Naively generate each “feature”
of X , conditioned on Y

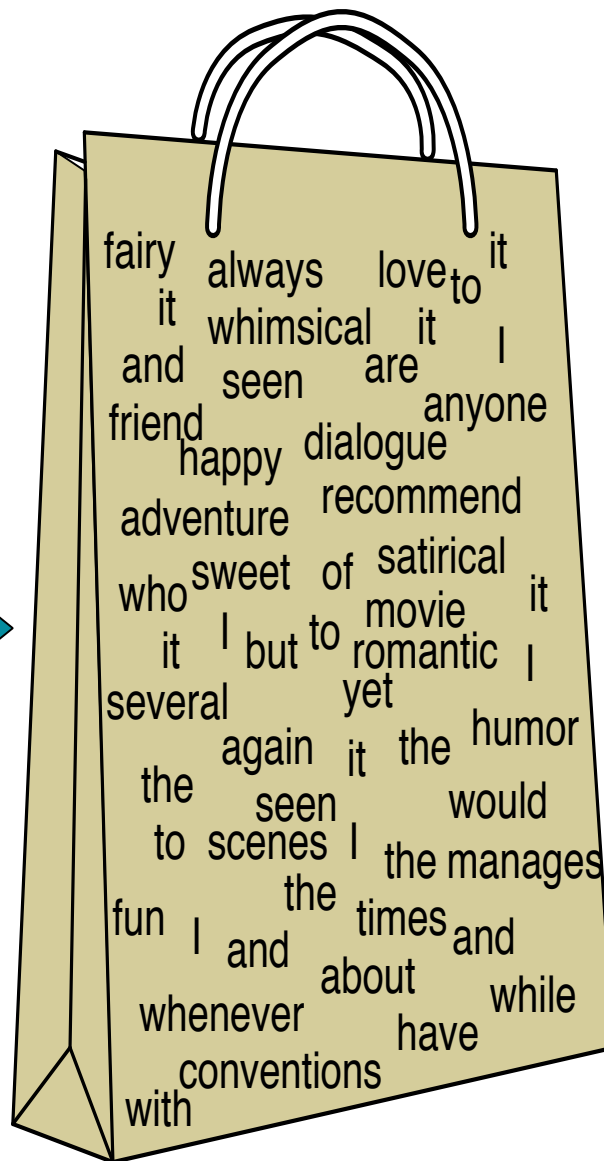
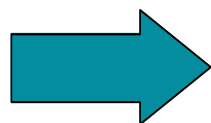
$$\operatorname{argmax}_Y \sum_{j=1}^D \log p(X_j | Y) + \log p(Y)$$

The Bag of Words Representation

I love this movie! It's sweet,
but with satirical humor. The
dialogue is great and the
adventure scenes are fun...
It manages to be whimsical
and romantic while laughing
at the conventions of the
fairy tale genre. I would
recommend it to just about
anyone. I've seen it several
times, and I'm always happy
to see it again whenever I
have a friend who hasn't
seen it yet!

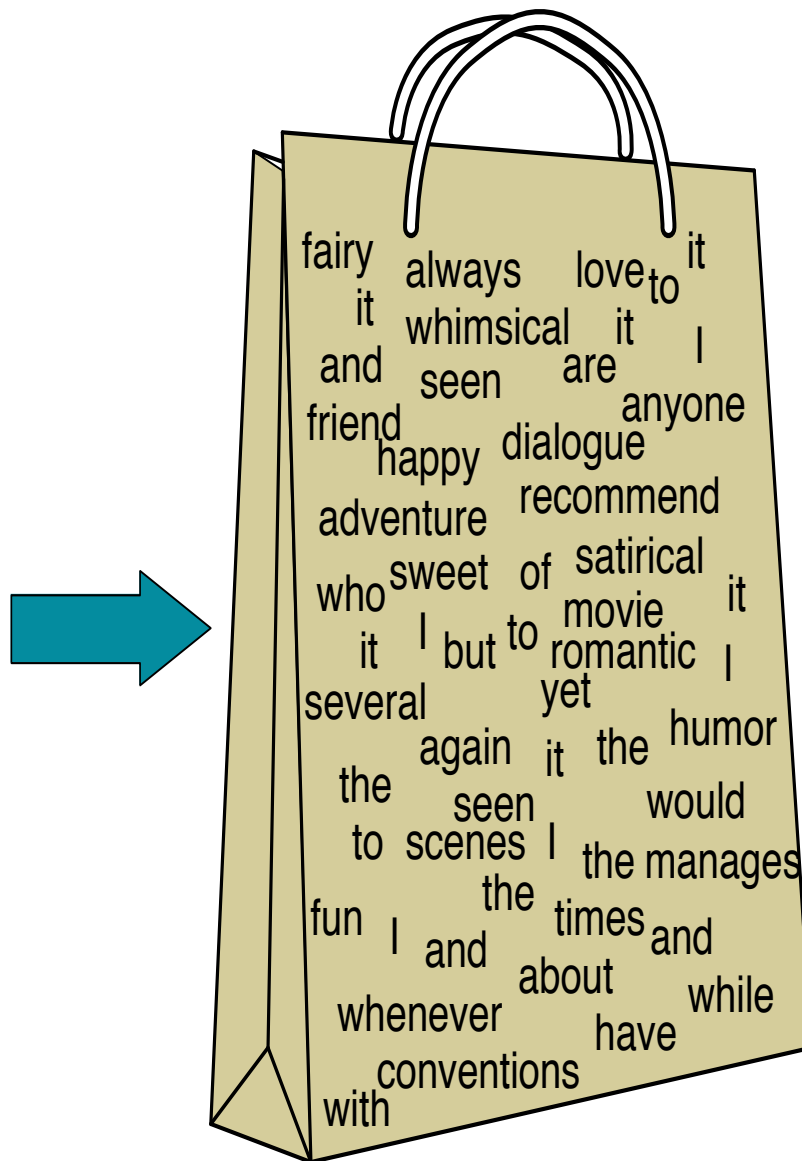
The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Bag of Words Representation



Naïve Bayes: A Generative Story

Generative Story

ϕ = distribution over K labels
for label $k = 1$ to K :

θ_k = generate parameters

$$p(y = k)$$

$$p(x_{ij}|y = k)$$

*global
parameters*

$$\sum_{j=1}^D \log p(X_{ij}|Y_i) + \log p(Y_i)$$

Naïve Bayes: A Generative Story

Generative Story

ϕ = distribution over K labels
for label $k = 1$ to K :



θ_k = generate parameters
for item $i = 1$ to N :

$$y_i \sim \text{Cat}(\phi)$$

Choose the label

$$\sum_{j=1}^D \log p(X_{ij}|Y_i) + \log p(Y_i)$$

Naïve Bayes: A Generative Story

Generative Story

ϕ = distribution over K labels
for label $k = 1$ to K :

θ_k = generate parameters

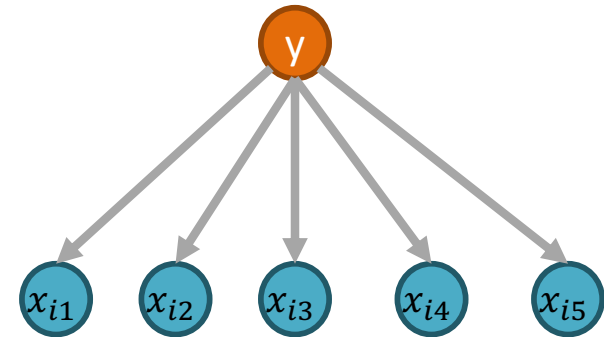
for item $i = 1$ to N :

$$y_i \sim \text{Cat}(\phi)$$

*local
variables*

for each feature j

$$x_{ij} \sim F_j(\theta_{y_i})$$



Generate each feature
based on the label

$$\sum_{j=1}^D \log p(X_{ij}|Y_i) + \log p(Y_i)$$

Naïve Bayes: A Generative Story

Generative Story

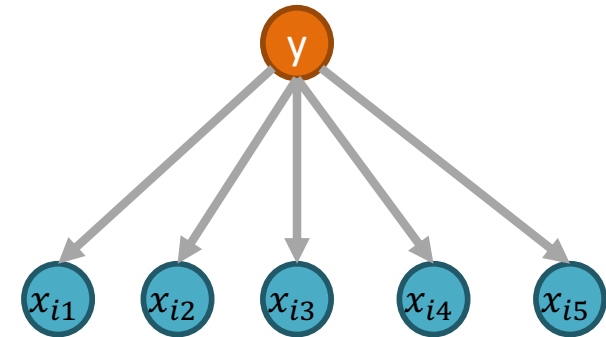
ϕ = distribution over K labels
for label $k = 1$ to K :

θ_k = generate parameters
for item $i = 1$ to N :

$$y_i \sim \text{Cat}(\phi)$$

for each feature j

$$x_{ij} \sim F_j(\theta_{y_i})$$



*each x_{ij} is conditionally
independent of one
another (given the label)*

$$\sum_{j=1}^D \log p(X_{ij}|Y_i) + \log p(Y_i)$$

Naïve Bayes: A Generative Story

Generative Story

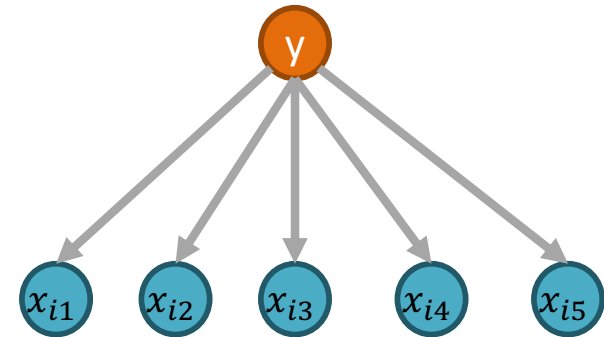
ϕ = distribution over K labels
for label $k = 1$ to K :

θ_k = generate parameters
for item $i = 1$ to N :

$$y_i \sim \text{Cat}(\phi)$$

for each feature j

$$x_{ij} \sim F_j(\theta_{y_i})$$



Maximize Log-likelihood

$$\mathcal{L}(\theta) = \sum_i \sum_j \log F_{y_i}(x_{ij}; \theta_{y_i}) + \sum_i \log \phi_{y_i} \quad \text{s. t.}$$
$$\sum_k \phi_k = 1 \quad \phi_k \geq 0 \quad \theta_k \text{ is valid for } F_j$$

Multinomial Naïve Bayes: A Generative Story

Generative Story

ϕ = distribution over K labels

for label $k = 1$ to K :

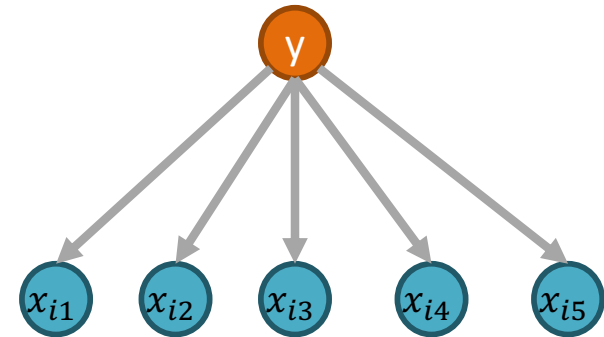
θ_k = distribution over J feature values

for item $i = 1$ to N :

$y_i \sim \text{Cat}(\phi)$

for each feature j

$x_{ij} \sim \text{Cat}(\theta_{y_i})$



Maximize Log-likelihood

$$\mathcal{L}(\theta) = \sum_i \sum_j \log \theta_{y_i, x_{i,j}} + \sum_i \log \phi_{y_i} \quad \text{s. t.}$$
$$\sum_k \phi_k = 1 \quad \phi_k \geq 0 \quad \sum_j \theta_{kj} = 1 \quad \forall k \quad \theta_{kj} \geq 0,$$

Multinomial Naïve Bayes: A Generative Story

Generative Story

ϕ = distribution over K labels

for label $k = 1$ to K :

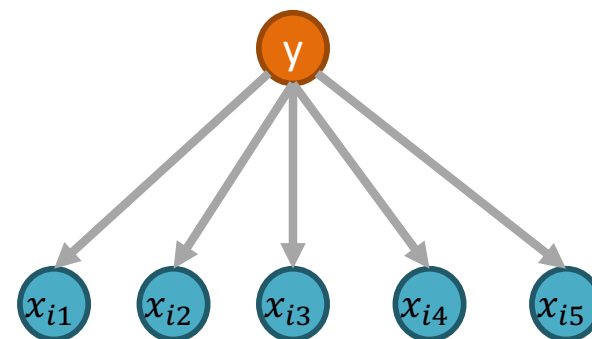
θ_k = distribution over J feature values

for item $i = 1$ to N :

$y_i \sim \text{Cat}(\phi)$

for each feature j

$x_{ij} \sim \text{Cat}(\theta_{y_i, j})$



Maximize Log-likelihood via Lagrange Multipliers (≥ 0 constraints not shown)

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_i \sum_j \log \theta_{y_i, x_{i,j}} + \sum_i \log \phi_{y_i} - \mu \left(\sum_k \phi_k - 1 \right) - \sum_k \lambda_k \left(\sum_j \theta_{kj} - 1 \right) \end{aligned}$$

Multinomial Naïve Bayes: Learning

Calculate *class priors*

For each k :

$items_k$ = all items with class = k

$$p(k) = \frac{|items_k|}{\# \text{ items}}$$

Calculate feature generation terms

For each k :

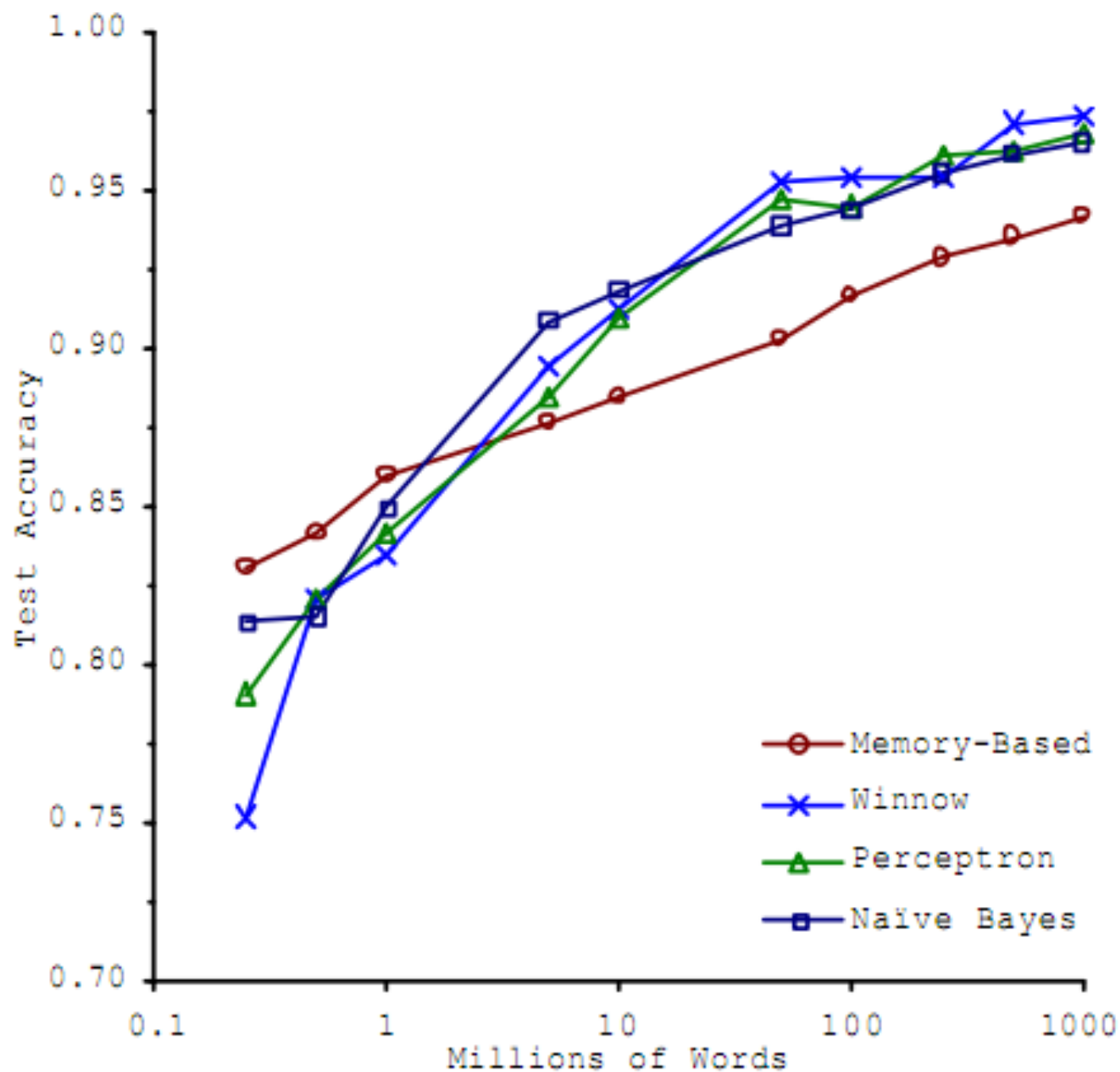
obs_k = single object containing all items labeled as k

For each feature j

n_{kj} = # of occurrences of j in obs_k

$$p(j|k) = \frac{n_{kj}}{\sum_{j'} n_{kj'}}$$

Brill and Banko (2001)
With enough data, the
classifier may not matter



Summary: Naïve Bayes is Not So Naïve, but not without issue

Pro

Very Fast, low storage requirements

Robust to Irrelevant Features

Very good in domains with many
equally important features

Optimal if the independence
assumptions hold

Dependable baseline for text
classification (but often not the best)

Con

Model the posterior in one go?
(e.g., use conditional maxent)

Are the features really
uncorrelated?

Are plain counts always
appropriate?

Are there “better” ways of
handling missing/noisy data?
(automated, more principled)

Outline

Directed Graphical Models

Naïve Bayes

Undirected Graphical Models

Factor Graphs

Ising Model

Message Passing: Graphical Model Inference

Undirected Graphical Models

An *undirected* graph $G=(V,E)$ that represents a probability distribution over random variables

$$X_1, \dots, X_N$$

Joint probability factorizes based on cliques in the graph

Undirected Graphical Models

An *undirected* graph $G=(V,E)$ that represents a probability distribution over random variables $X_1, \dots, X_N$

Joint probability factorizes based on cliques in the graph

Common name: **Markov Random Fields**

Undirected Graphical Models

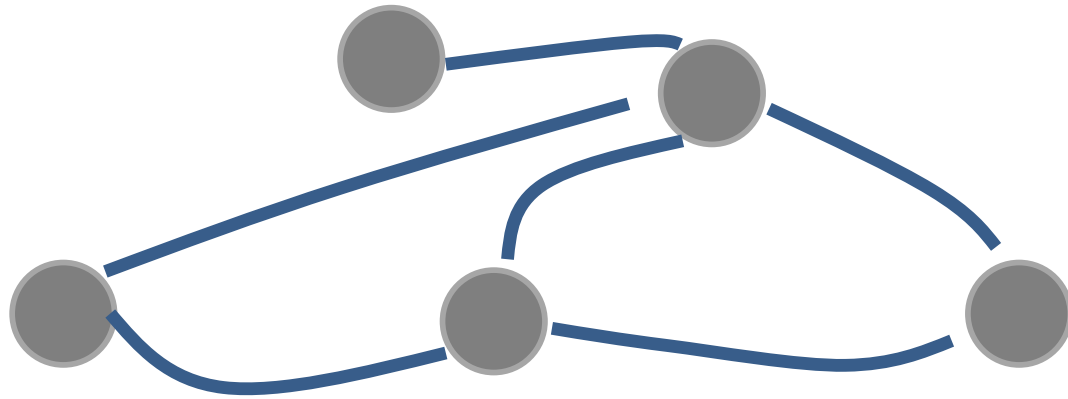
An *undirected* graph $G=(V,E)$ that represents a probability distribution over random variables $X_1, \dots, X_N$

Joint probability factorizes based on cliques in the graph

Common name: **Markov Random Fields**

Undirected graphs can have an alternative formulation as **Factor Graphs**

Markov Random Fields: Undirected Graphs

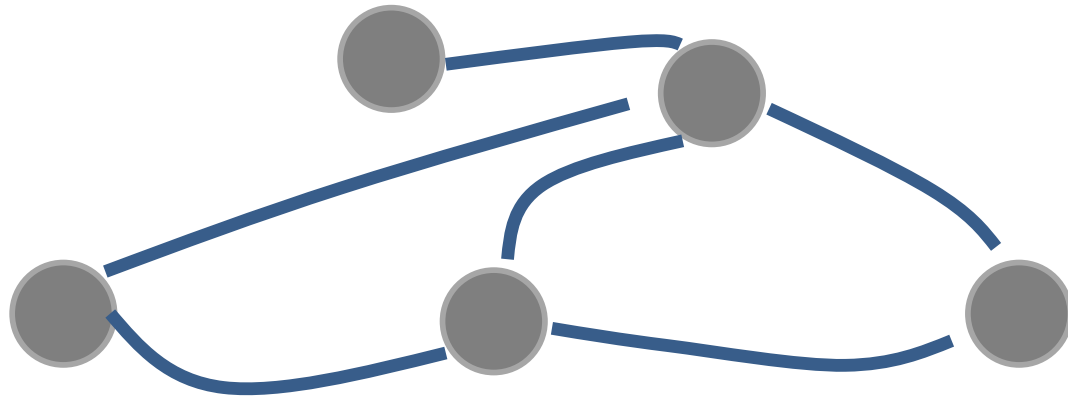


$$p(x_1, x_2, x_3, \dots, x_N)$$

Markov Random Fields: Undirected Graphs

clique: subset of nodes,
where nodes are
pairwise connected

maximal clique: a clique
that cannot add a node
and remain a clique

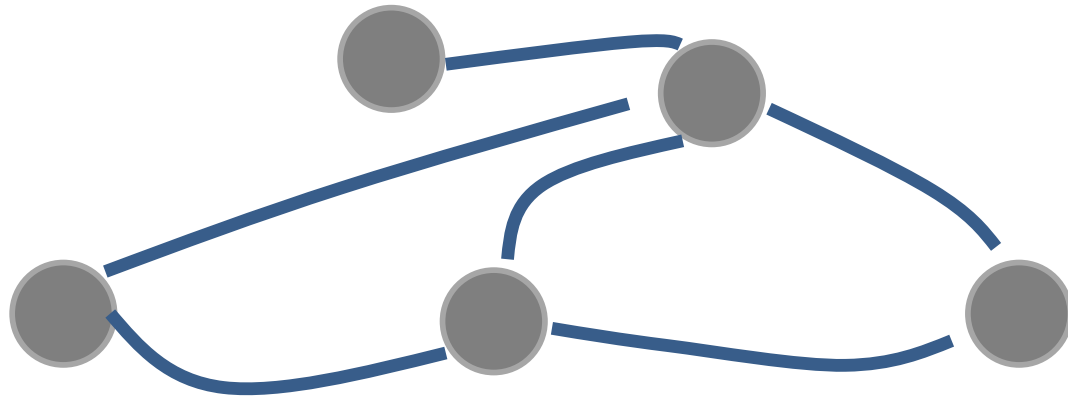


$$p(x_1, x_2, x_3, \dots, x_N)$$

Markov Random Fields: Undirected Graphs

clique: subset of nodes,
where nodes are
pairwise connected

maximal clique: a clique
that cannot add a node
and remain a clique



$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_C)$$

global
normalization

maximal
cliques

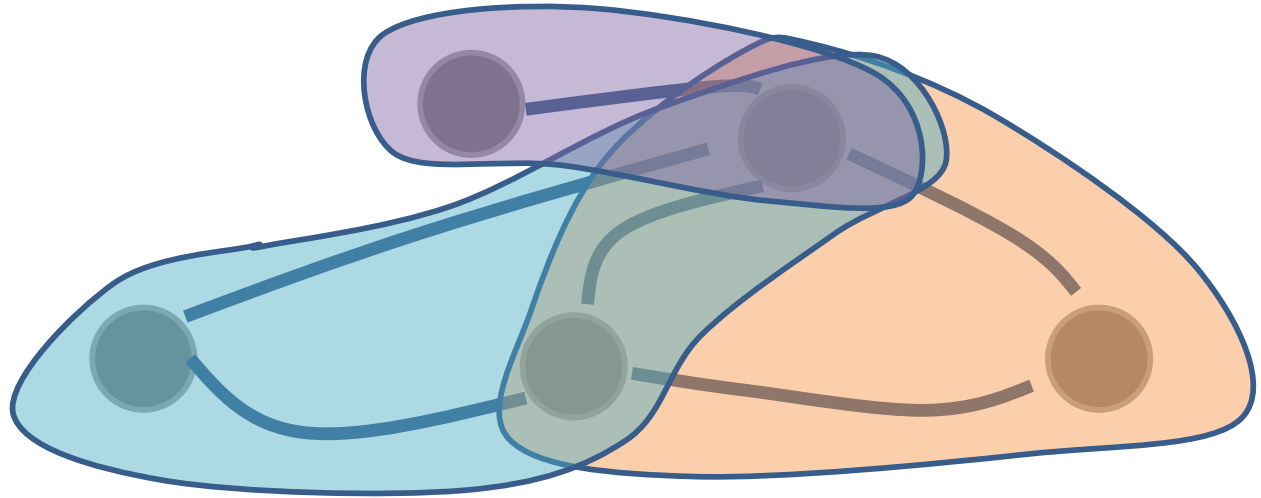
potential function (not
necessarily a probability!)

variables part
of the clique C

Markov Random Fields: Undirected Graphs

clique: subset of nodes,
where nodes are
pairwise connected

maximal clique: a clique
that cannot add a node
and remain a clique



$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_C)$$

global
normalization

maximal
cliques

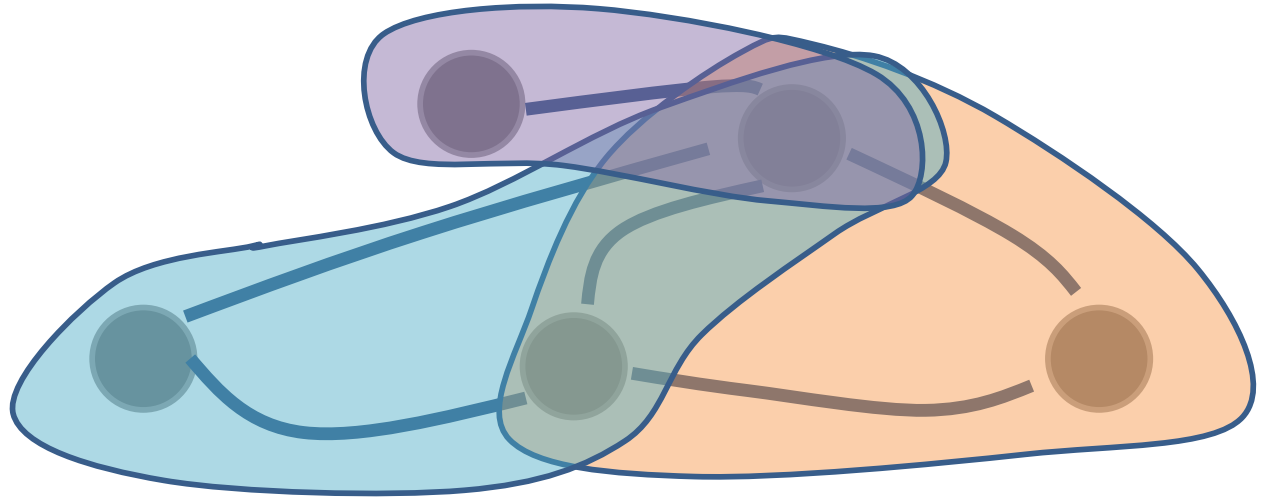
potential function (not
necessarily a probability!)

variables part
of the clique C

Markov Random Fields: Undirected Graphs

clique: subset of nodes,
where nodes are
pairwise connected

maximal clique: a clique
that cannot add a node
and remain a clique



$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_C)$$

Q: What restrictions should we
place on the potentials ψ_C ?

global
normalization

maximal
cliques

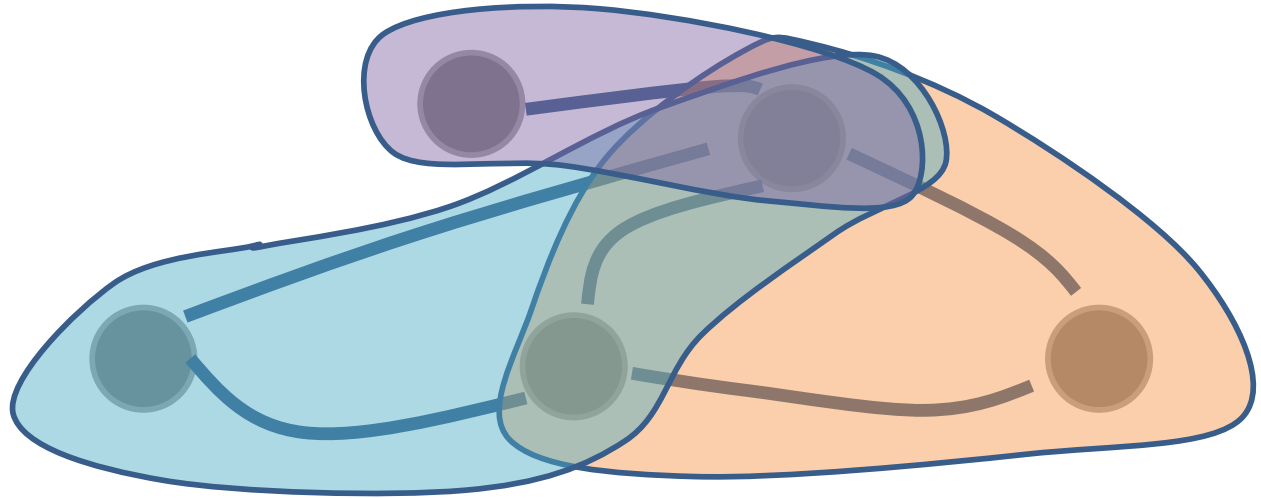
potential function (not
necessarily a probability!)

variables part
of the clique C

Markov Random Fields: Undirected Graphs

clique: subset of nodes,
where nodes are
pairwise connected

maximal clique: a clique
that cannot add a node
and remain a clique



$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_C \psi_C(x_C)$$

Q: What restrictions should we
place on the potentials ψ_C ?

A: $\psi_C \geq 0$ (or $\psi_C > 0$)

global
normalization

maximal
cliques

potential function (not
necessarily a probability!)

variables part
of the clique C

Terminology: Potential Functions

$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_c \psi_c(x_c)$$

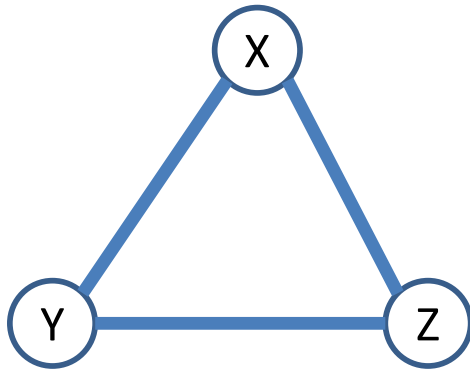
energy function (for clique C)

*(get the total energy of a
configuration by summing the
individual energy functions)*

$$\psi_c(x_c) = \underbrace{\exp -E(x_c)}$$

Boltzmann distribution

Ambiguity in Undirected Model Notation



$$p(x, y, z) \propto \psi(x, y, z)$$

$$p(x, y, z) \propto \psi_{1(x,y)}\psi_{2(y,z)}\psi_{3(x,z)}$$

Outline

Directed Graphical Models

Naïve Bayes

Undirected Graphical Models

Factor Graphs

Ising Model

Message Passing: Graphical Model Inference

MRFs as Factor Graphs

Undirected graphs: $G=(V,E)$ that represents $p(X_1, \dots, X_N)$

Factor graph of p : Bipartite graph of evidence nodes X , factor nodes F , and edges T

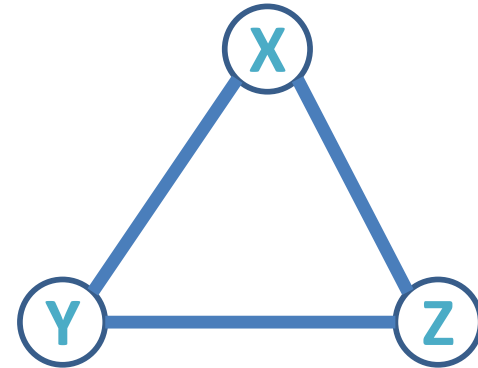
Evidence nodes X are the random variables

Factor nodes F take values associated with the *potential functions*

Edges show what variables are used in which factors

MRFs as Factor Graphs

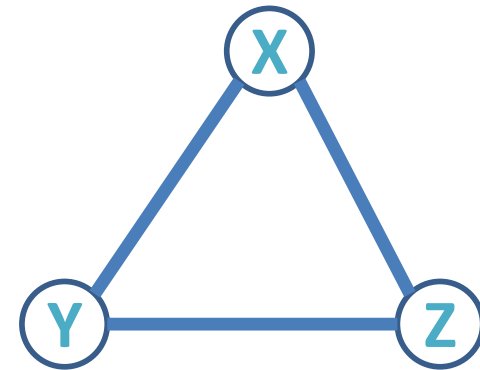
Undirected graphs:
 $G=(V,E)$ that
represents
 $p(X_1, \dots, X_N)$



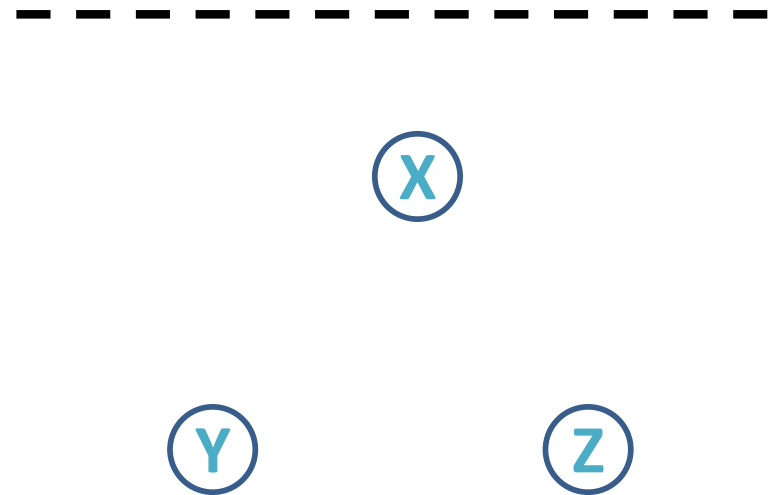
Factor graph of p :
Bipartite graph of
evidence nodes X ,
factor nodes F , and
edges T

MRFs as Factor Graphs

Undirected graphs:
 $G=(V,E)$ that represents
 $p(X_1, \dots, X_N)$



Factor graph of p :
Bipartite graph of
evidence nodes X ,
factor nodes F , and
edges T



Evidence nodes X are
the random variables

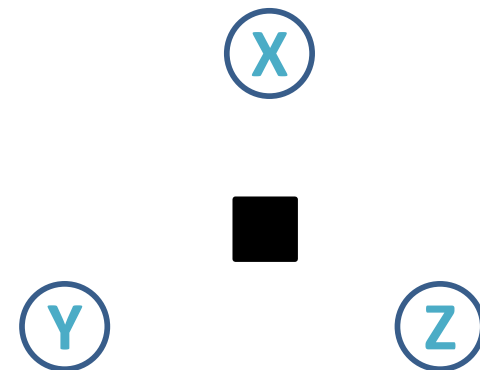
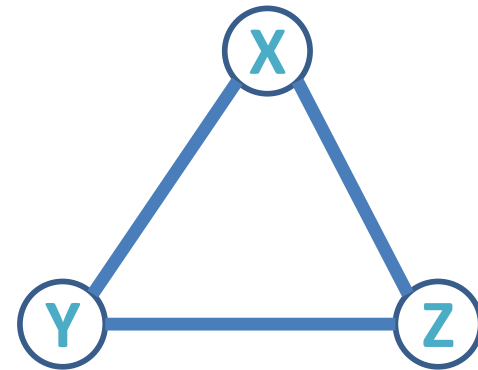
MRFs as Factor Graphs

Undirected graphs: $G=(V,E)$
that represents
 $p(X_1, \dots, X_N)$

Factor graph of p : Bipartite
graph of evidence nodes X ,
factor nodes F , and edges T

Evidence nodes X are the
random variables

Factor nodes F take values
associated with the
potential functions



MRFs as Factor Graphs

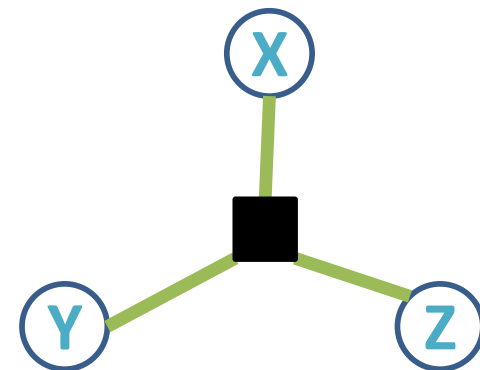
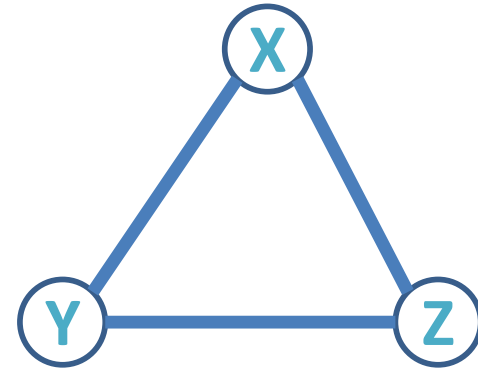
Undirected graphs: $G=(V,E)$ that represents $p(X_1, \dots, X_N)$

Factor graph of p : Bipartite graph of evidence nodes X , factor nodes F , and edges T

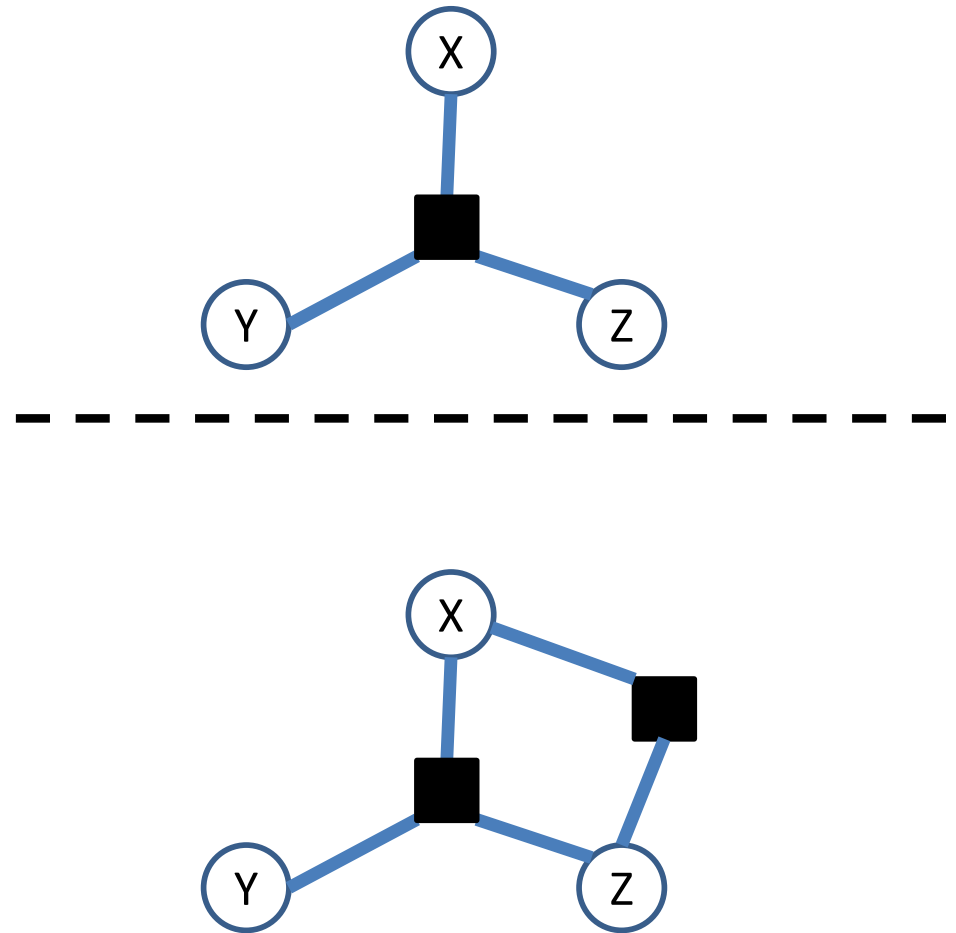
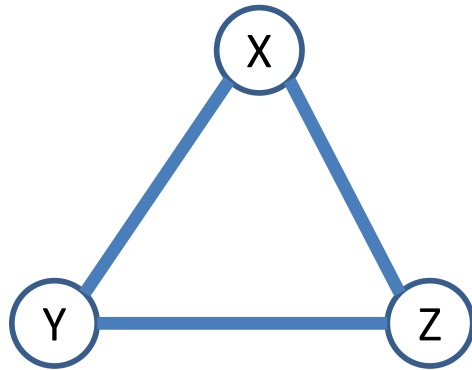
Evidence nodes X are the random variables

Factor nodes F take values associated with the *potential functions*

Edges show what variables are used in which factors

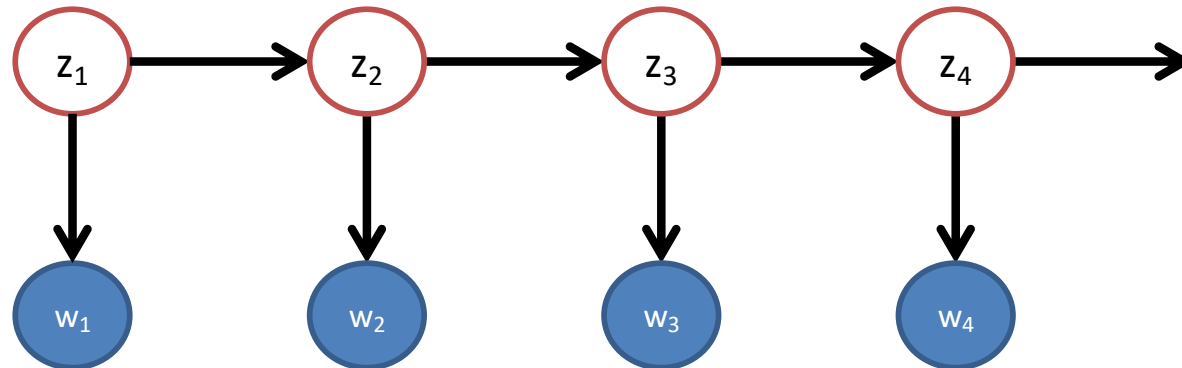


Different Factor Graph Notation for the Same Graph



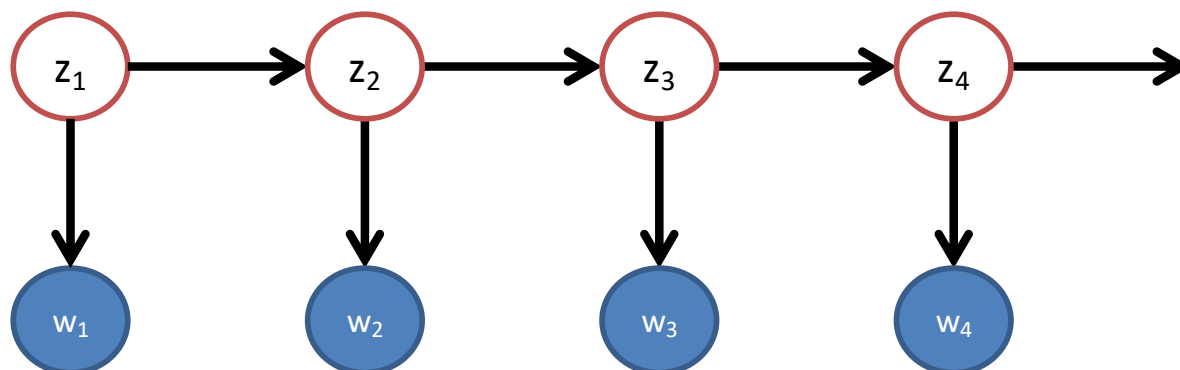
Example: Linear Chain

Directed (e.g.,
hidden Markov model
[HMM]; generative)

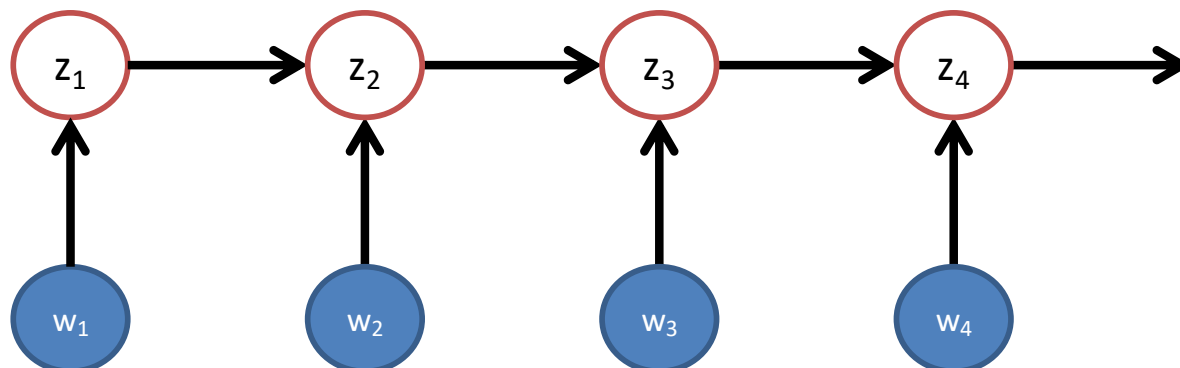


Example: Linear Chain

Directed (e.g.,
hidden Markov model
[HMM]; generative)

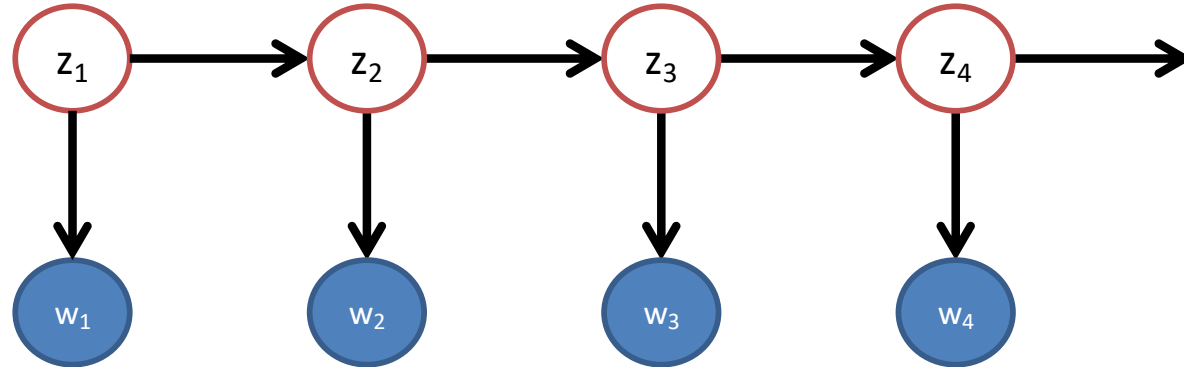


Directed (e.g.,
maximum entropy
Markov model
[MEMM]; conditional)

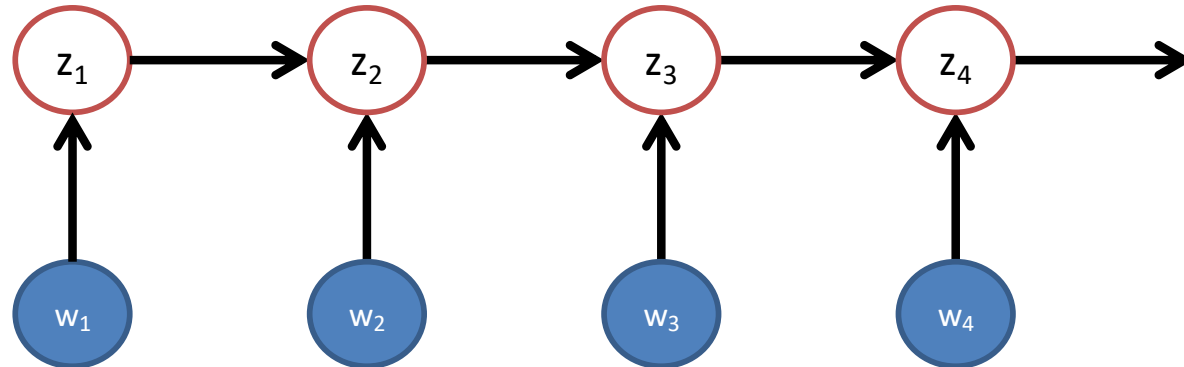


Example: Linear Chain

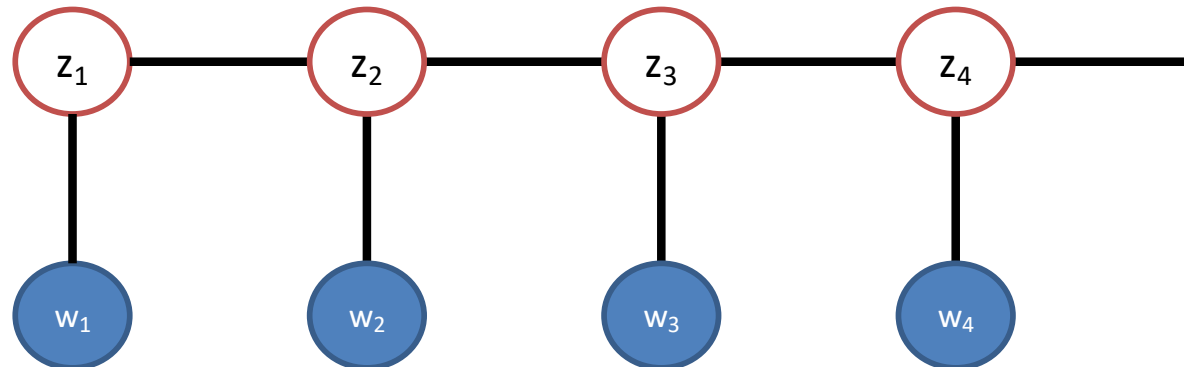
Directed (e.g.,
hidden Markov model
[HMM]; generative)



Directed (e.g.,
maximum entropy
Markov model
[MEMM]; conditional)

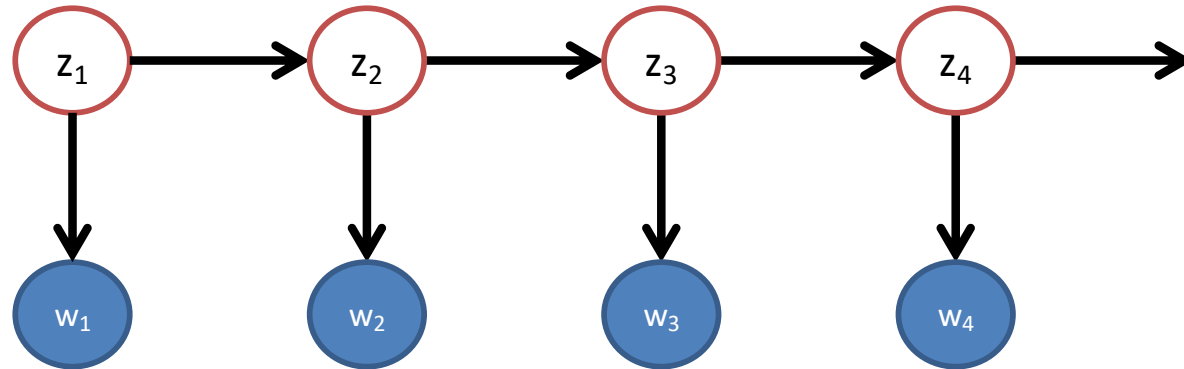


Undirected
(e.g., conditional
random field
[CRF])

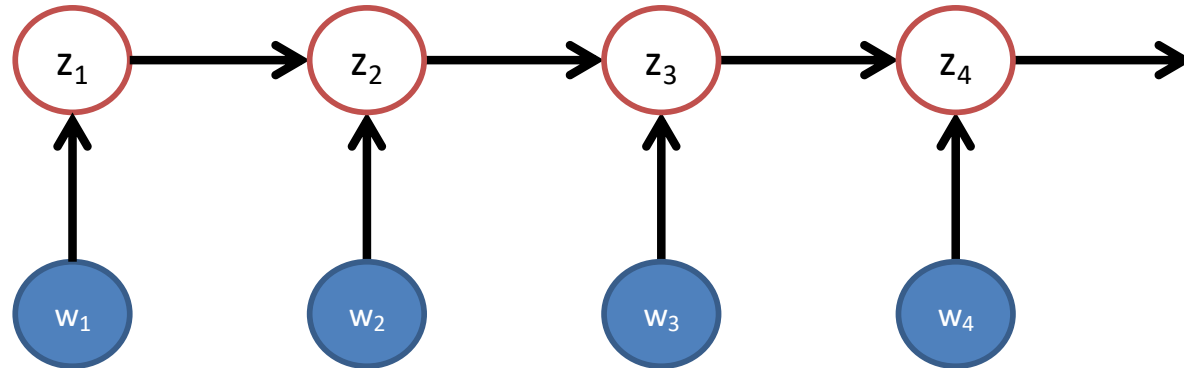


Example: Linear Chain

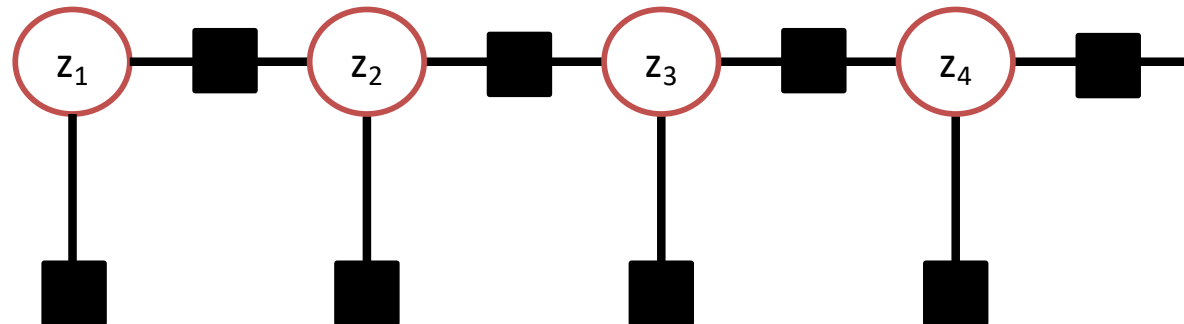
Directed (e.g.,
hidden Markov model
[HMM]; generative)



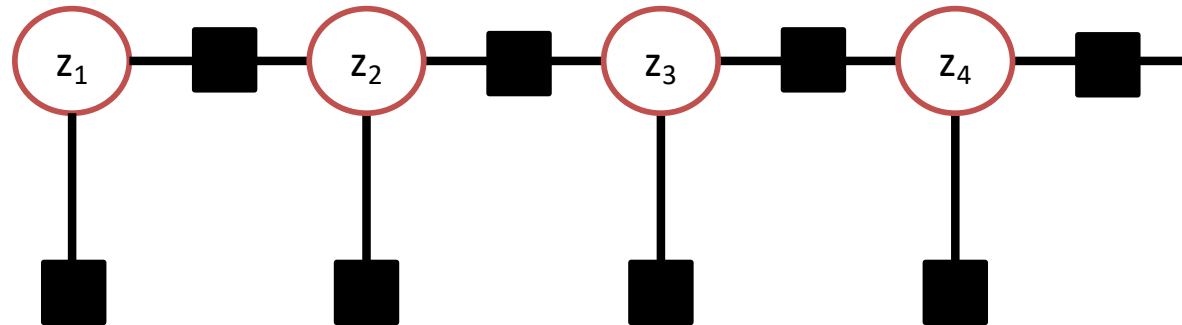
Directed (e.g.,
maximum entropy
Markov model
[MEMM]; conditional)



**Undirected as
factor graph**
(e.g., conditional
random field [CRF])



Example: Linear Chain Conditional Random Field

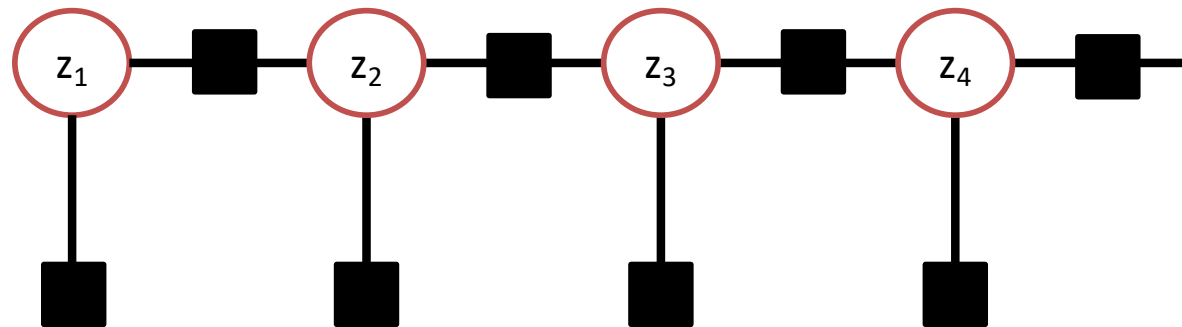


Widely used in applications like
part-of-speech tagging

Noun-Mod *Noun* *Verb* *Noun*

President Obama told Congress ...

Example: Linear Chain Conditional Random Field



Widely used in applications like
part-of-speech tagging

Noun-Mod Noun Verb Noun

President Obama told Congress ...

and *named entity recognition*

Person Person Other Org.

President Obama told Congress ...

Linear Chain CRFs for Part of Speech Tagging

A linear chain CRF is a conditional probabilistic model of the sequence of tags $z_1, z_2, \dots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

Linear Chain CRFs for Part of Speech Tagging

$$p(\clubsuit | \diamond)$$

A linear chain CRF is a **conditional probabilistic model** of the sequence of tags $z_1, z_2, \dots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

Linear Chain CRFs for Part of Speech Tagging

$$p(z_1, z_2, \dots, z_N | \diamond)$$

A linear chain CRF is a conditional probabilistic model of the sequence of tags $z_1, z_2, \dots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

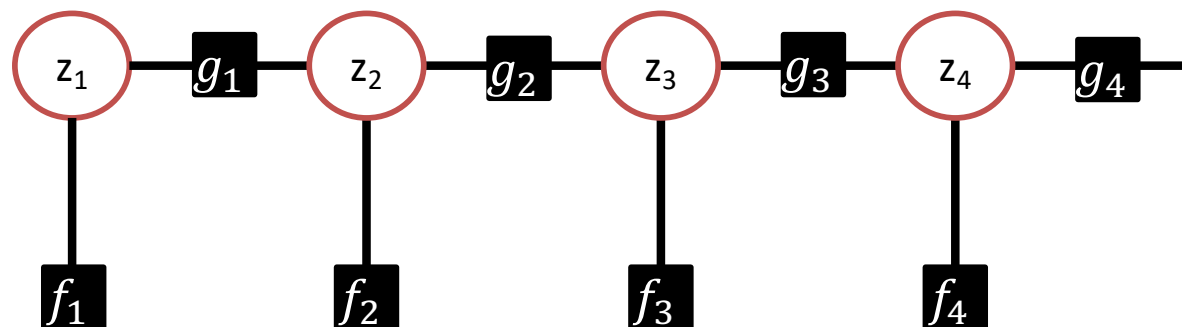
Linear Chain CRFs for Part of Speech Tagging

$$p(z_1, z_2, \dots, z_N | x_{1:N})$$

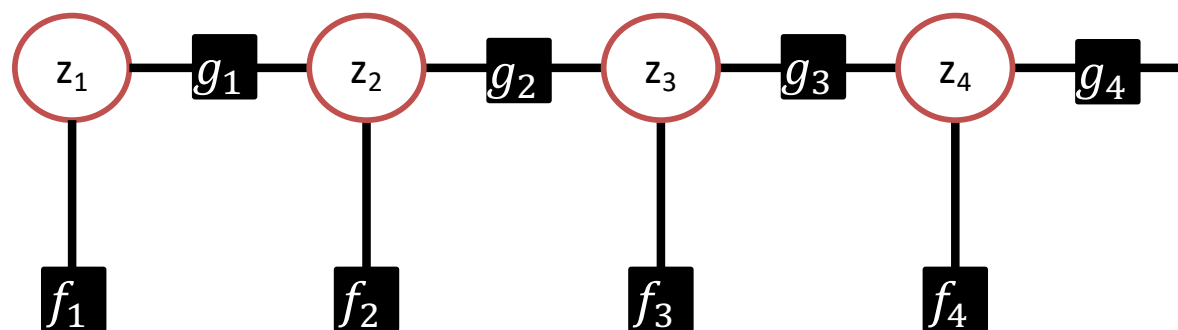
A linear chain CRF is a conditional probabilistic model of the sequence of tags $z_1, z_2, \dots, z_N$ conditioned on the *entire* input sequence $x_{1:N}$

Linear Chain CRFs for Part of Speech Tagging

$$p(z_1, z_2, \dots, z_N | x_{1:N})$$



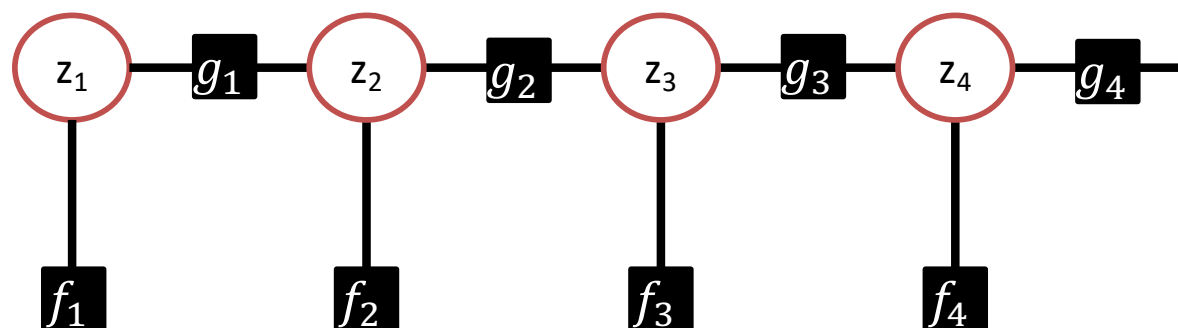
Linear Chain CRFs for Part of Speech Tagging



$$p(z_1, z_2, \dots, z_N | x_{1:N}) \propto \prod_{i=1}^N \exp(\langle \theta^{(f)}, f_i(z_i) \rangle + \langle \theta^{(g)}, g_i(z_i, z_{i+1}) \rangle)$$

Linear Chain CRFs for Part of Speech Tagging

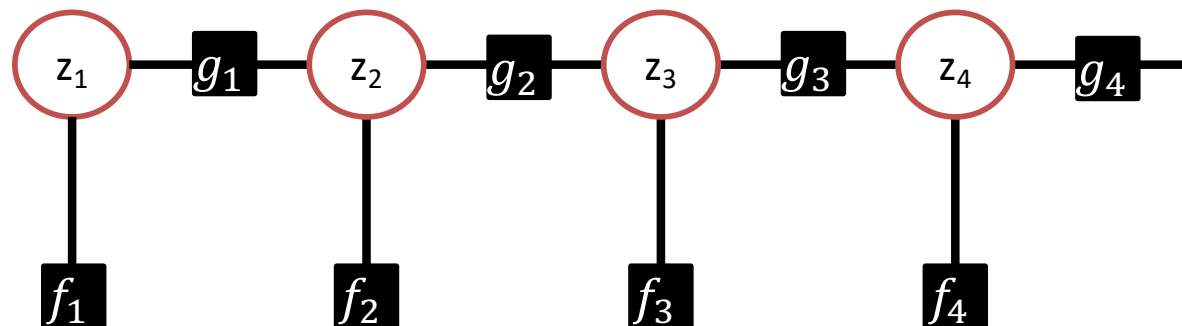
g_j : inter-tag features
(can depend on
any/all input words
 $x_{1:N}$)



Linear Chain CRFs for Part of Speech Tagging

g_j : inter-tag features
(can depend on
any/all input words
 $x_{1:N}$)

f_i : solo tag features
(can depend on
any/all input words
 $x_{1:N}$)



Linear Chain CRFs for Part of Speech Tagging

g_j : inter-tag features
(can depend on
any/all input words
 $x_{1:N}$)

f_i : solo tag features
(can depend on
any/all input words
 $x_{1:N}$)

Feature design, just
like in maxent
models!

Linear Chain CRFs for Part of Speech Tagging

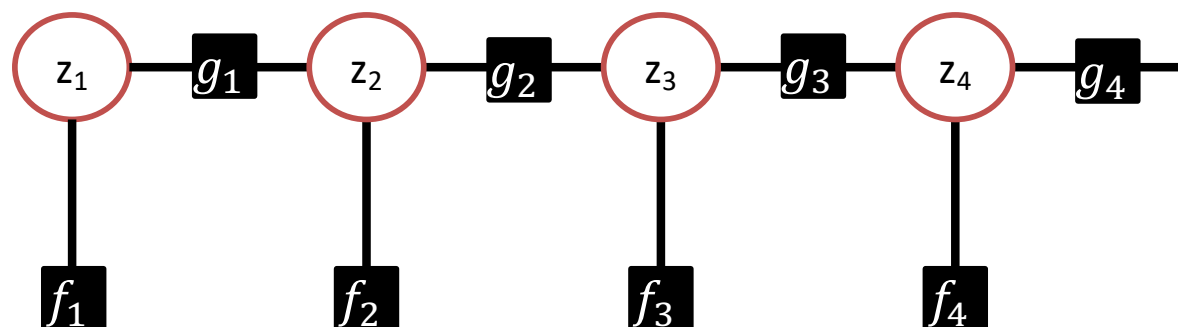
g_j : inter-tag features
(can depend on
any/all input words
 $x_{1:N}$)

f_i : solo tag features
(can depend on
any/all input words
 $x_{1:N}$)

Example:

$$g_{j,N \rightarrow V}(z_j, z_{j+1}) = 1 \text{ (if } z_j == N \text{ \& } z_{j+1} == V \text{) else } 0$$

$$g_{j,\text{told},N \rightarrow V}(z_j, z_{j+1}) = 1 \text{ (if } z_j == N \text{ \& } z_{j+1} == V \text{ \& } x_j == \text{told) else } 0$$



Outline

Directed Graphical Models

Naïve Bayes

Undirected Graphical Models

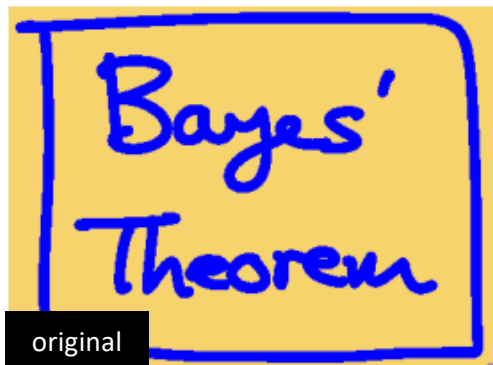
Factor Graphs

Ising Model

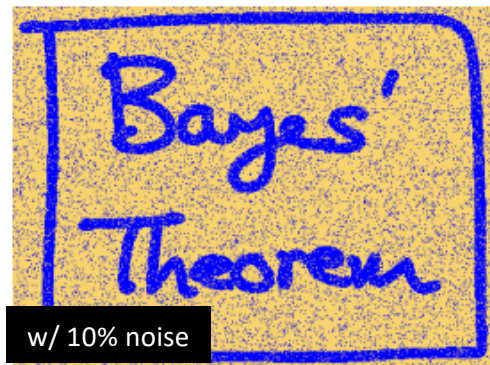
Message Passing: Graphical Model Inference

Example: Ising Model

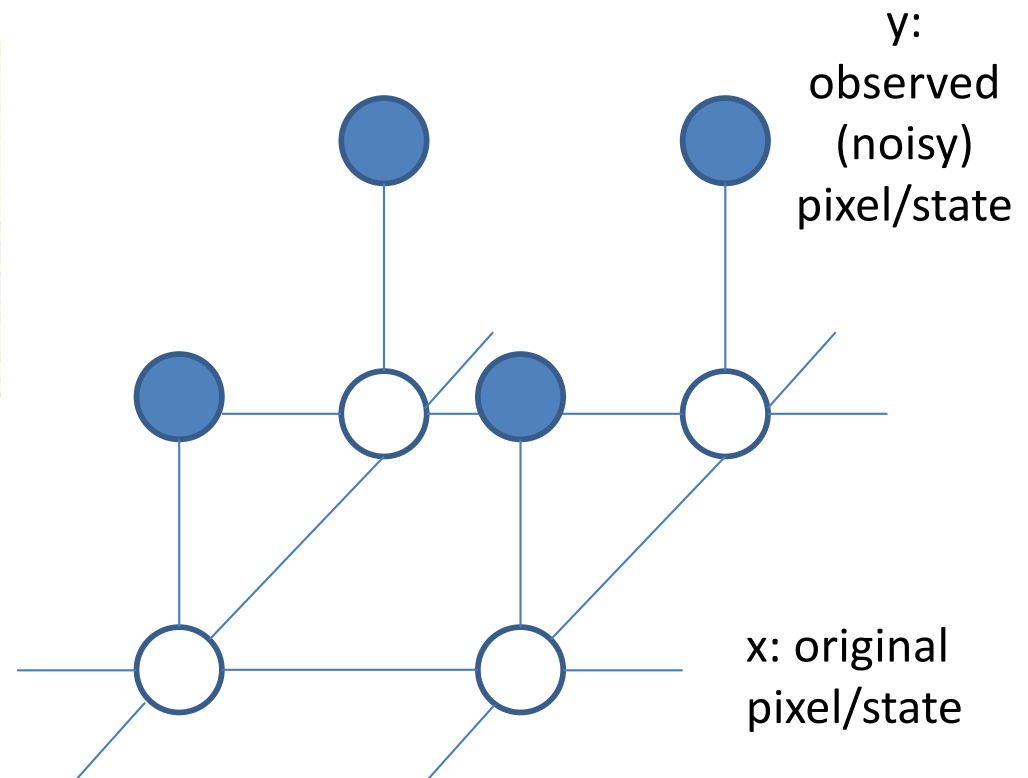
Image denoising (Bishop, 2006; Fig 8.30)



X

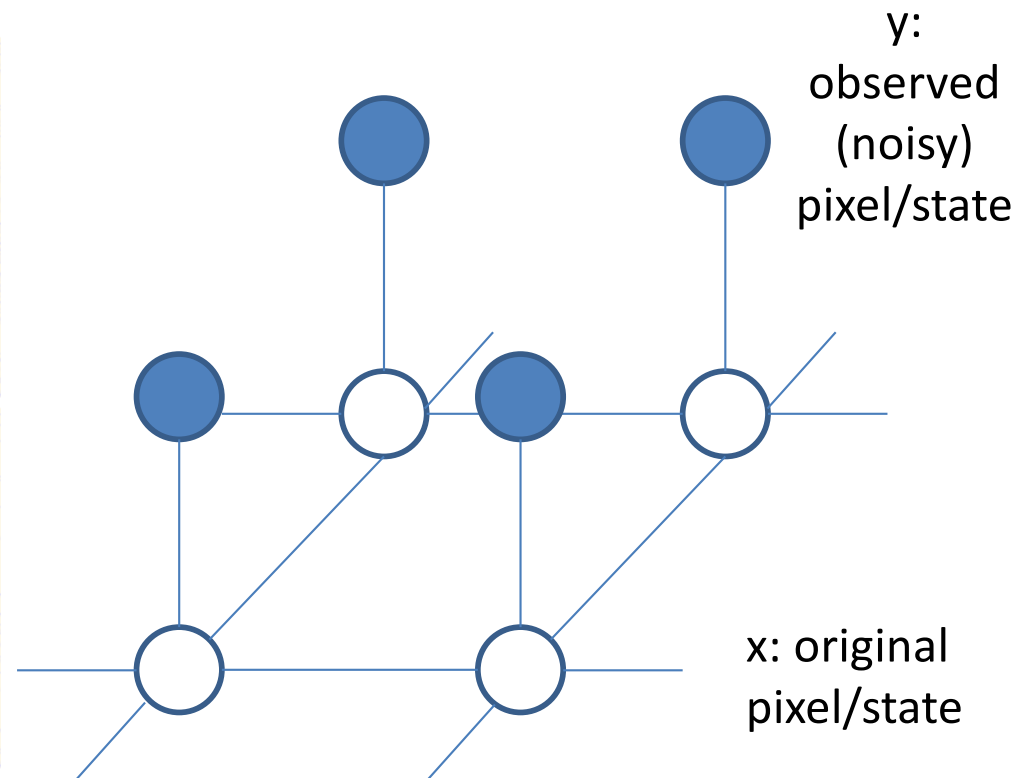
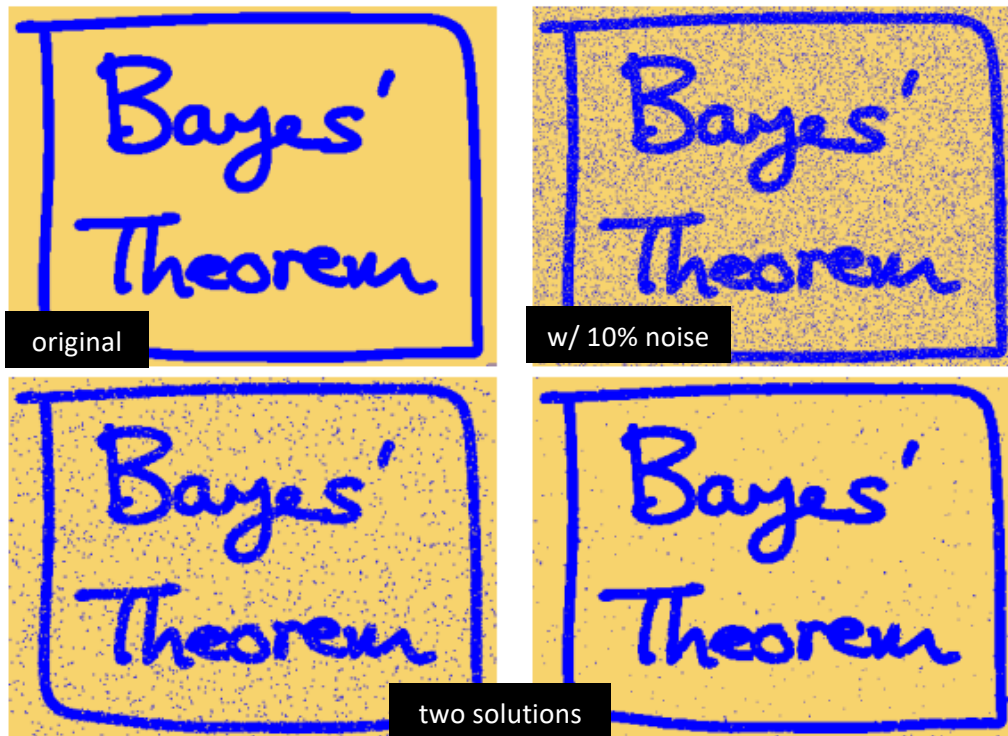


Y



Example: Ising Model

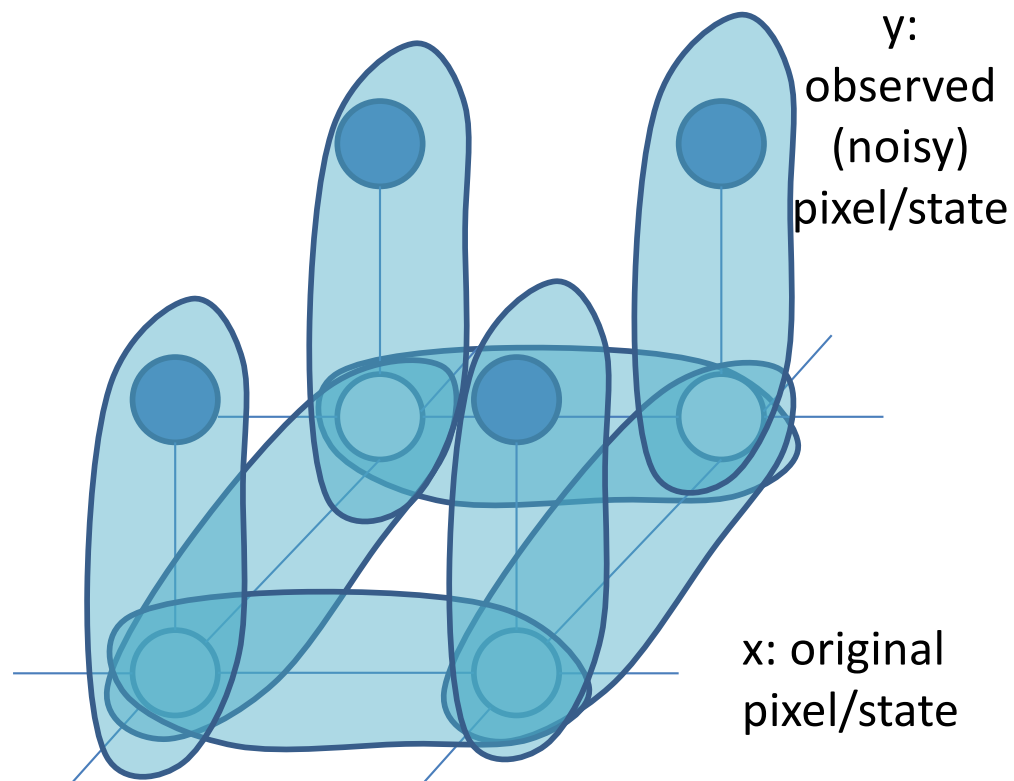
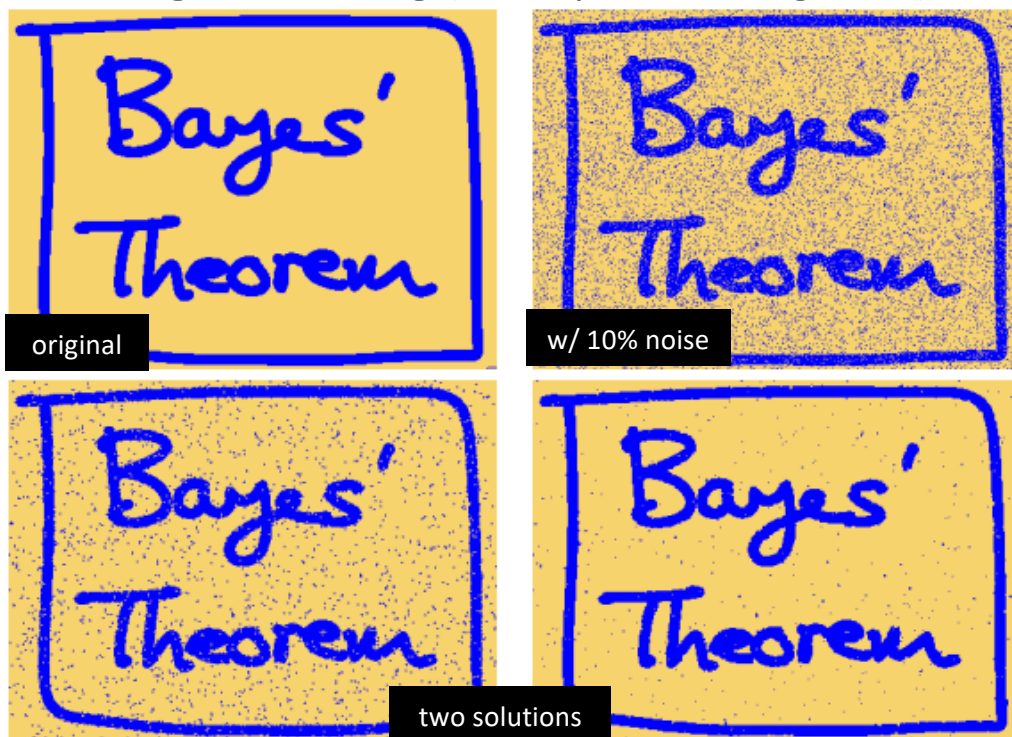
Image denoising (Bishop, 2006; Fig 8.30)



Q: What are the cliques?

Example: Ising Model

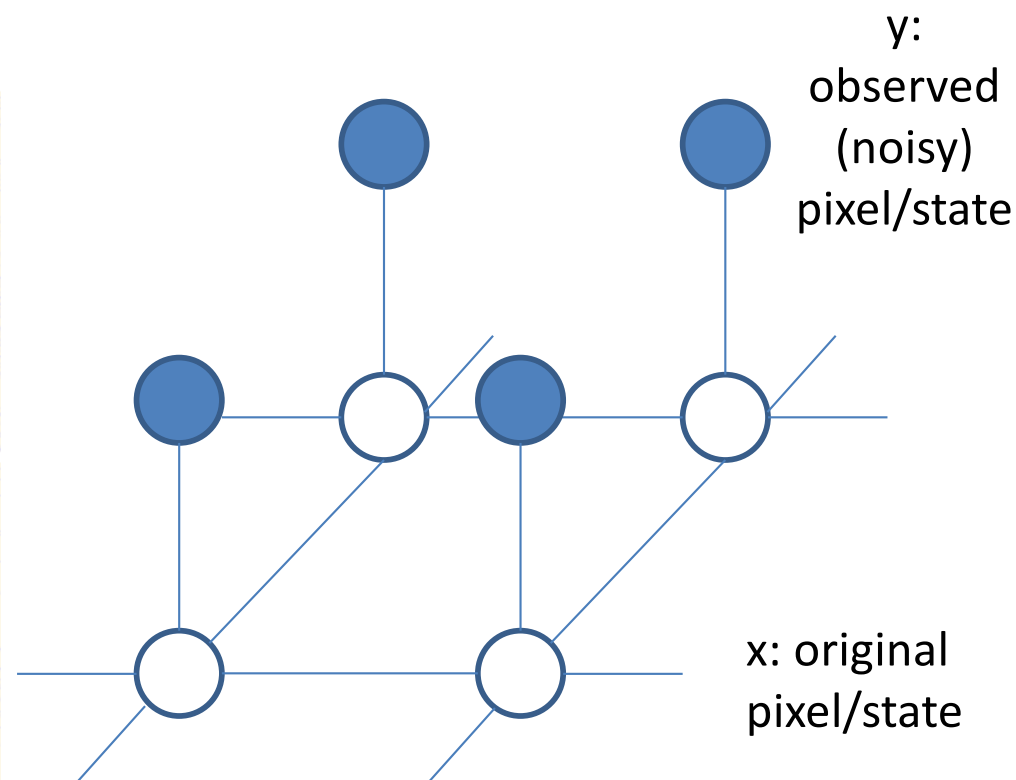
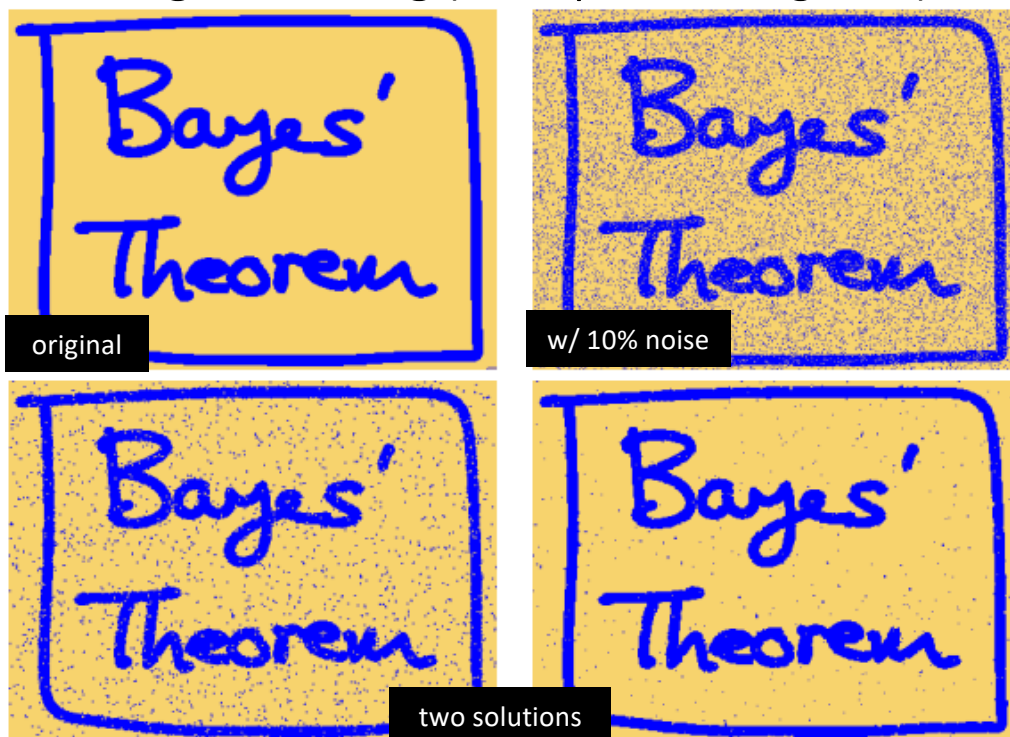
Image denoising (Bishop, 2006; Fig 8.30)



Q: What are the cliques?

Example: Ising Model

Image denoising (Bishop, 2006; Fig 8.30)



neighboring pixels
should be similar

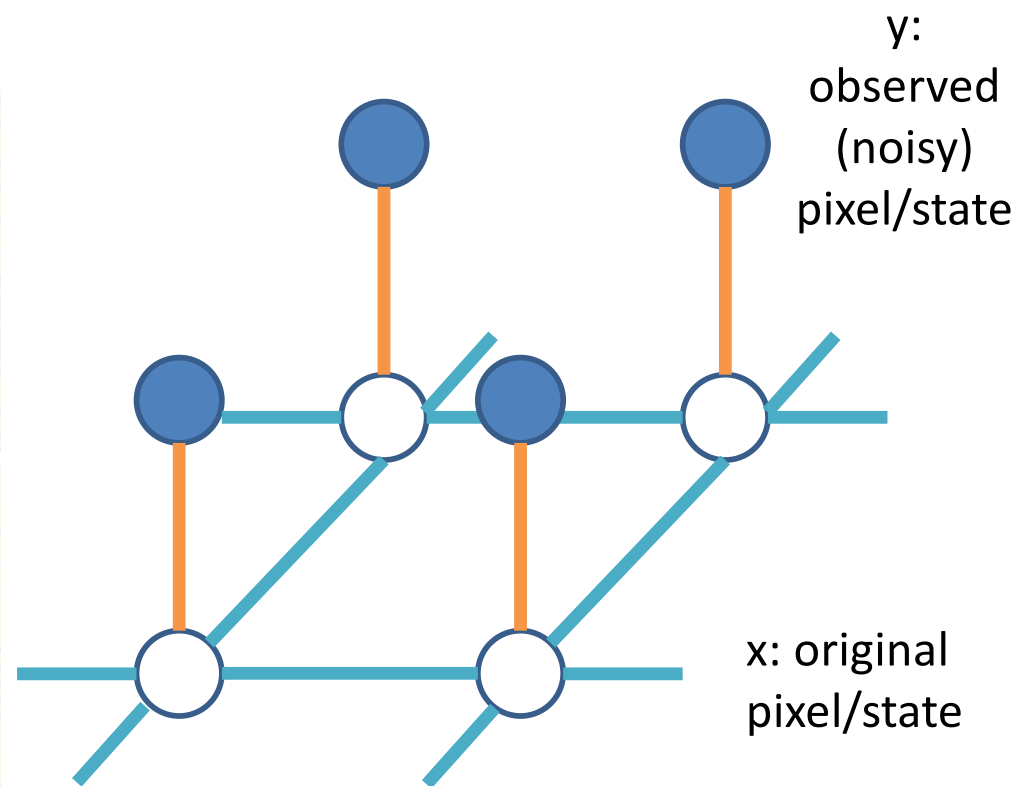
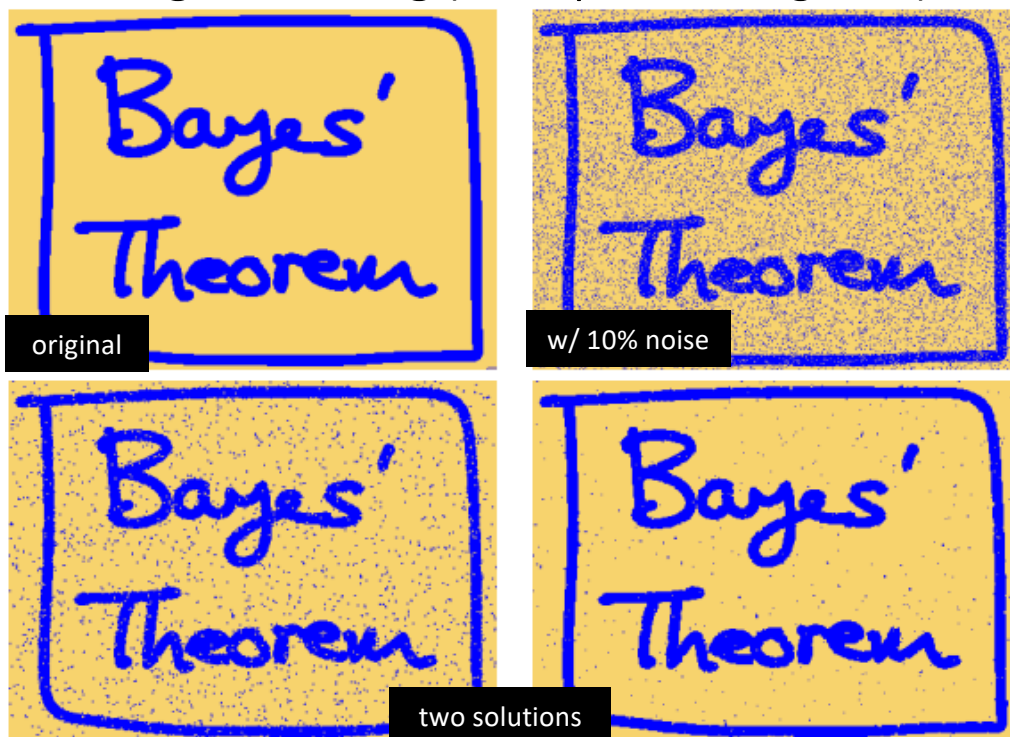
$$E(x, y) = h \sum_i x_i - \beta \sum_{ij} x_i x_j - \eta \sum_i x_i y_i$$

allow for a bias

x_i and y_i should
be correlated

Example: Ising Model

Image denoising (Bishop, 2006; Fig 8.30)



neighboring pixels
should be similar

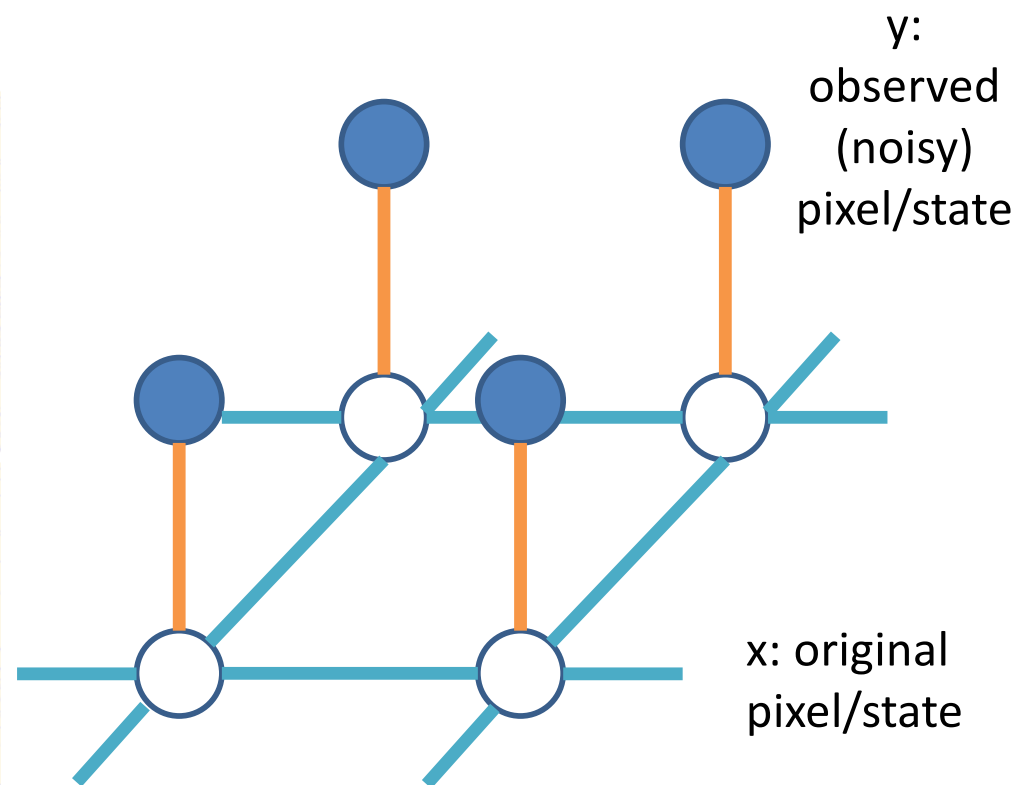
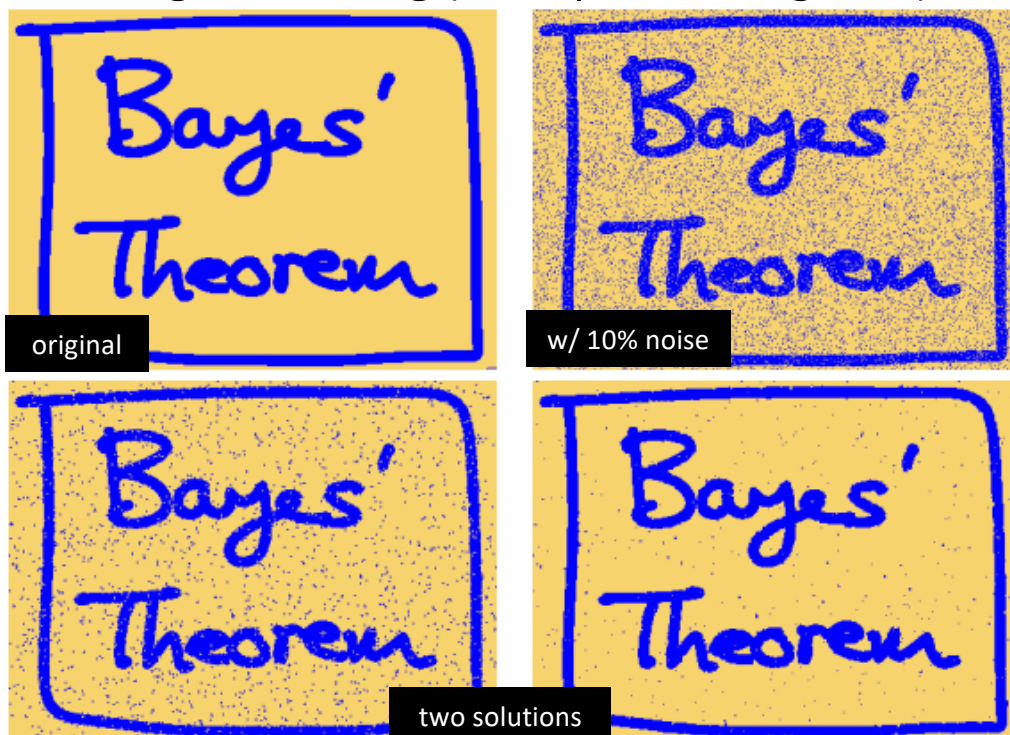
$$E(x, y) = h \sum_i x_i - \beta \sum_{ij} x_i x_j - \eta \sum_i x_i y_i$$

allow for a bias

x_i and y_i should
be correlated

Example: Ising Model

Image denoising (Bishop, 2006; Fig 8.30)



Q: Why subtract β and η ?

neighboring pixels should be similar

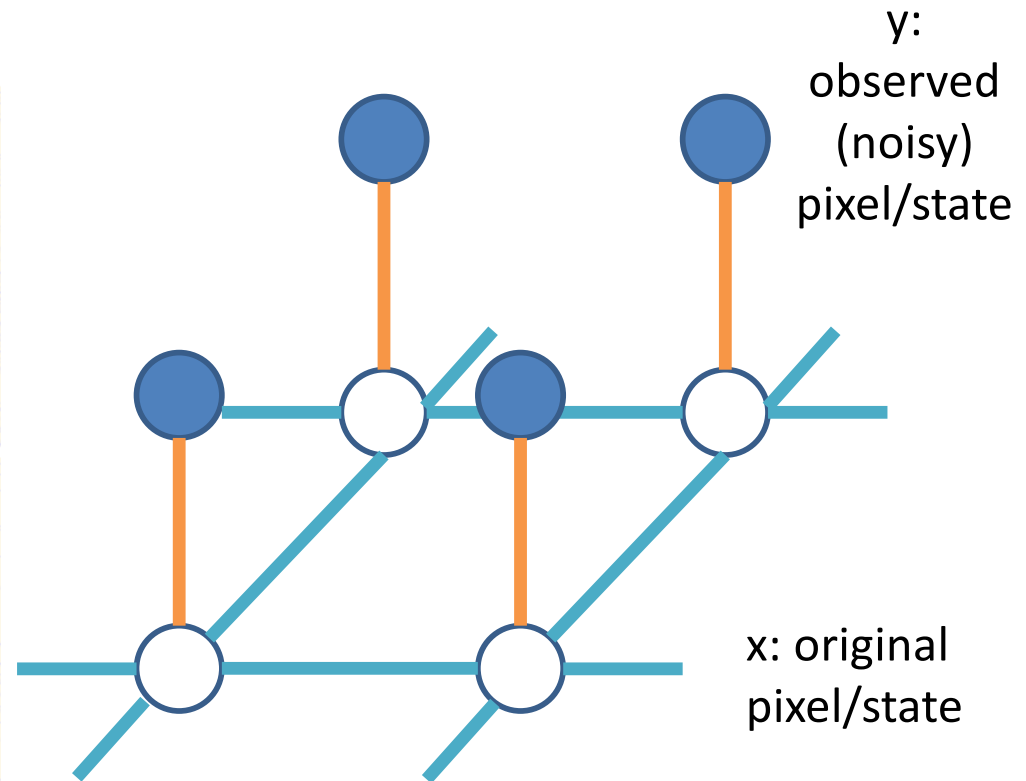
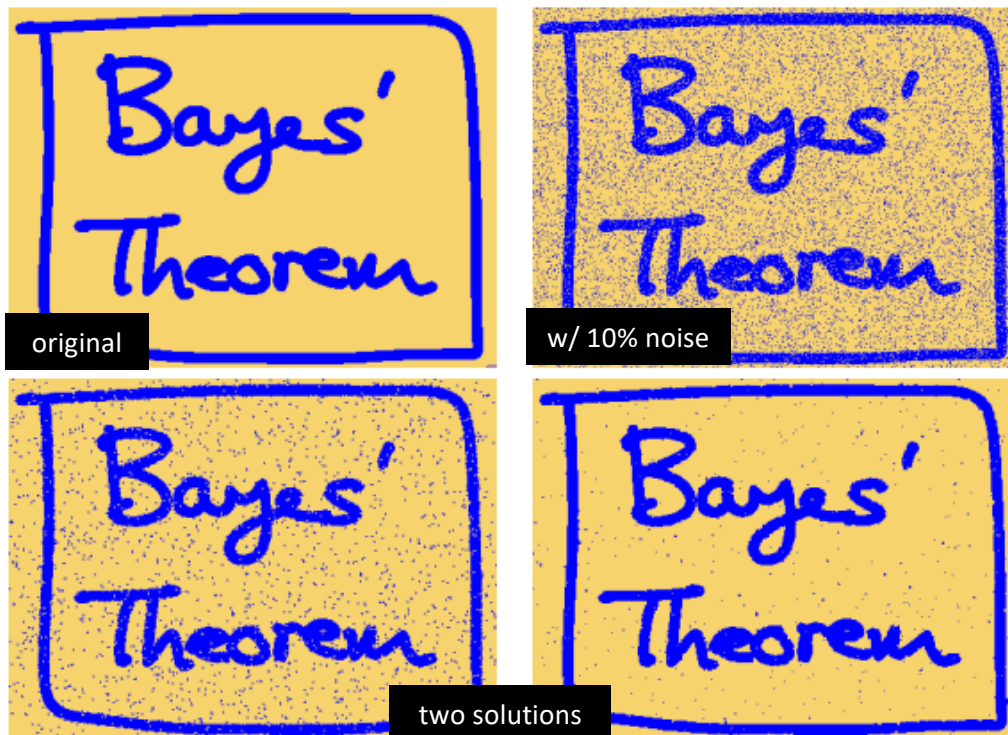
$$E(x, y) = h \sum_i x_i - \beta \sum_{ij} x_i x_j - \eta \sum_i x_i y_i$$

allow for a bias

x_i and y_i should be correlated

Example: Ising Model

Image denoising (Bishop, 2006; Fig 8.30)



Q: Why subtract β and η ?

A: Better states $\rightarrow$ lower energy (higher potential)
 $\psi_C(x_C) = \exp -E(x_C)$

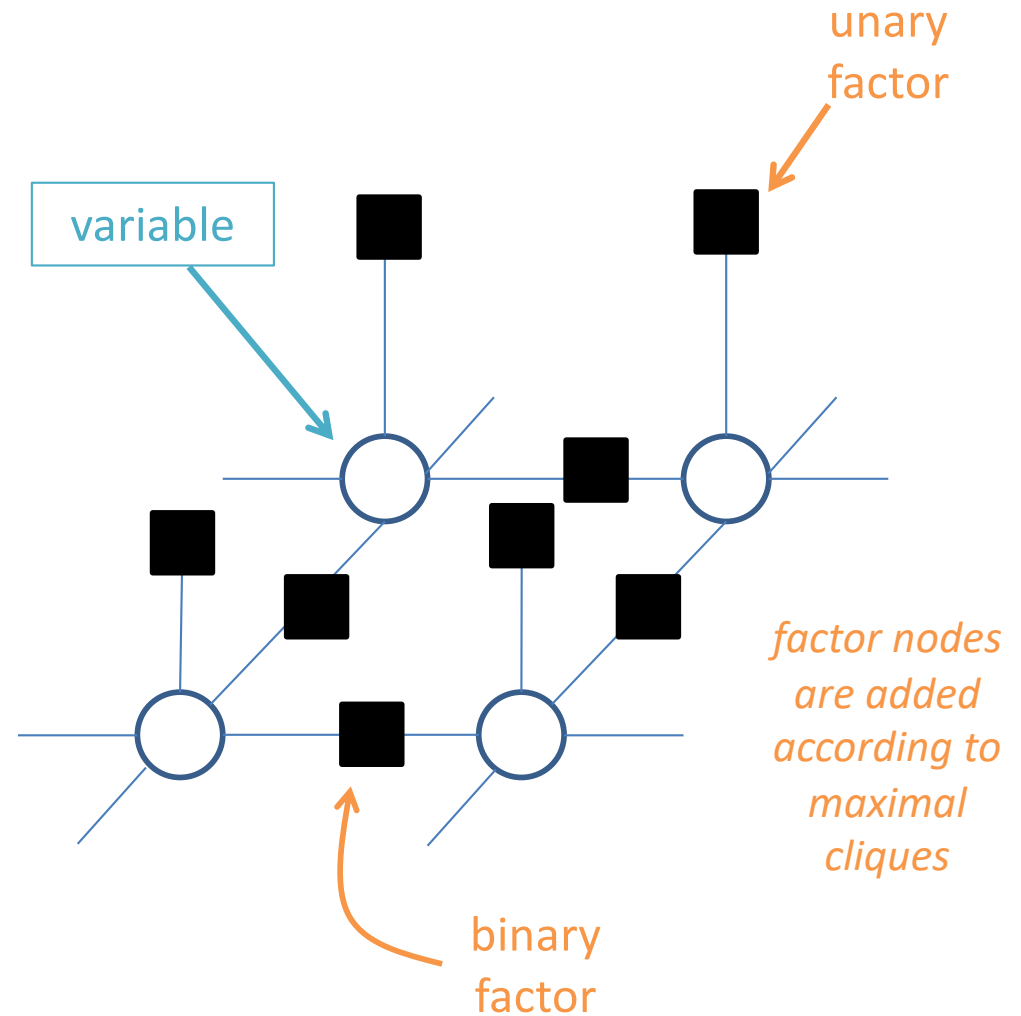
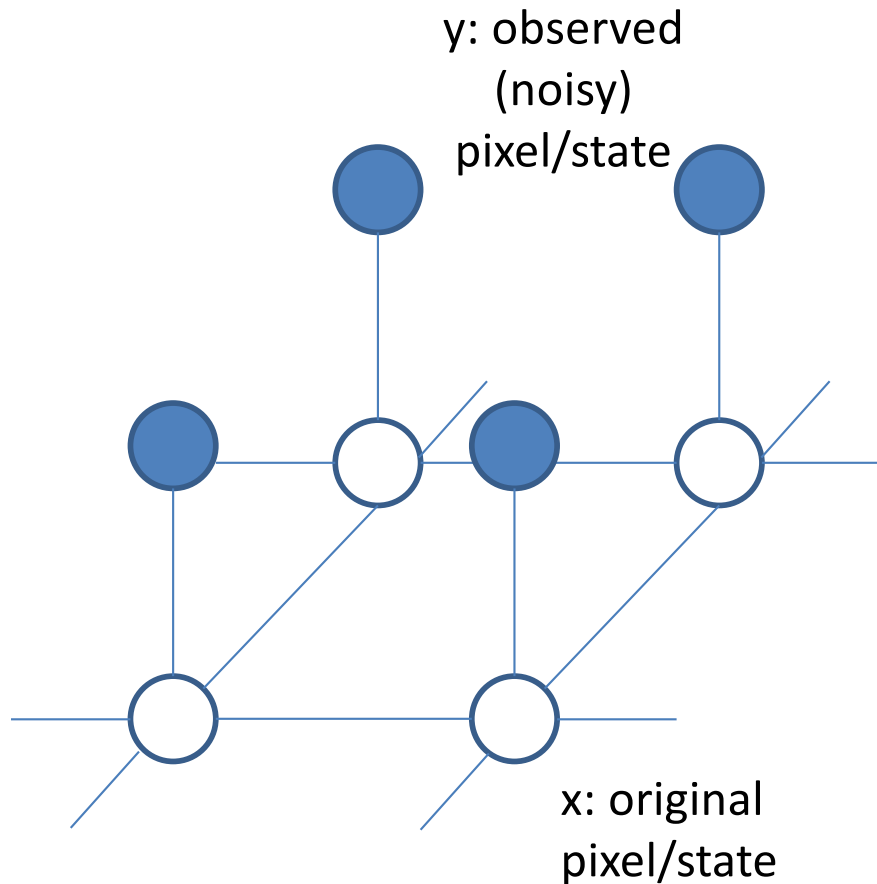
neighboring pixels should be similar

$$E(x, y) = h \sum_i x_i - \beta \sum_{ij} x_i x_j - \eta \sum_i x_i y_i$$

allow for a bias

x_i and y_i should be correlated

Markov Random Fields with Factor Graph Notation



factor graphs are **bipartite**

Outline

Directed Graphical Models

Naïve Bayes

Undirected Graphical Models

Factor Graphs

Ising Model

Message Passing: Graphical Model Inference

Two Problems for Undirected Models

$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_c \psi_c(x_c)$$

Finding the normalizer

Computing the marginals

Two Problems for Undirected Models

$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_c \psi_c(x_c)$$

Finding the normalizer

Computing the marginals

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

Two Problems for Undirected Models

$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_c \psi_c(x_c)$$

Finding the normalizer

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

Computing the marginals

Sum over all variable combinations, with the x_n coordinate fixed

$$Z_n(v) = \sum_{x: x_n=v} \prod_c \psi_c(x_c)$$

Example: 3 variables, fix the 2nd dimension

$$Z_2(v) = \sum_{x_1} \sum_{x_3} \prod_c \psi_c(x = (x_1, v, x_3))$$

Two Problems for Undirected Models

$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_c \psi_c(x_c)$$

Finding the normalizer

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

Q: Why are these difficult?

Computing the marginals

Sum over all variable combinations, with the x_n coordinate fixed

$$Z_n(v) = \sum_{x: x_n=v} \prod_c \psi_c(x_c)$$

Example: 3 variables, fix the 2nd dimension

$$Z_2(v) = \sum_{x_1} \sum_{x_3} \prod_c \psi_c(x = (x_1, v, x_3))$$

Two Problems for Undirected Models

$$p(x_1, x_2, x_3, \dots, x_N) = \frac{1}{Z} \prod_c \psi_c(x_c)$$

Finding the normalizer

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

Q: Why are these difficult?

A: Many different combinations

Computing the marginals

Sum over all variable combinations, with the x_n coordinate fixed

$$Z_n(v) = \sum_{x: x_n=v} \prod_c \psi_c(x_c)$$

Example: 3 variables, fix the 2nd dimension

$$Z_2(v) = \sum_{x_1} \sum_{x_3} \prod_c \psi_c(x = (x_1, v, x_3))$$

Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number 'one' to the soldier behind you.

If you are the rearmost soldier in the line, say the number 'one' to the soldier in front of you.

If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side

Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number 'one' to the soldier behind you.

If you are the rearmost soldier in the line, say the number 'one' to the soldier in front of you.

If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side

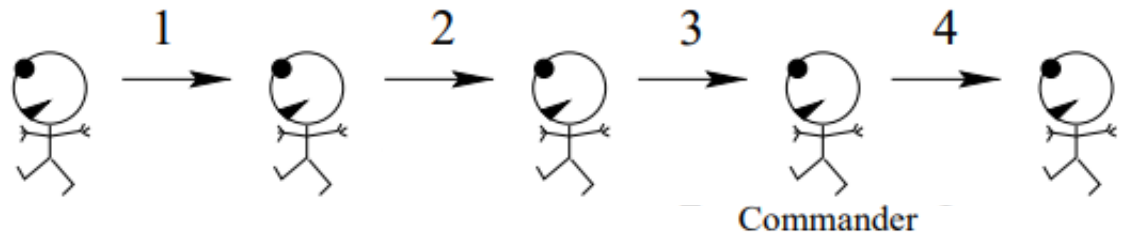


Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number 'one' to the soldier behind you.

If you are the rearmost soldier in the line, say the number 'one' to the soldier in front of you.

If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side

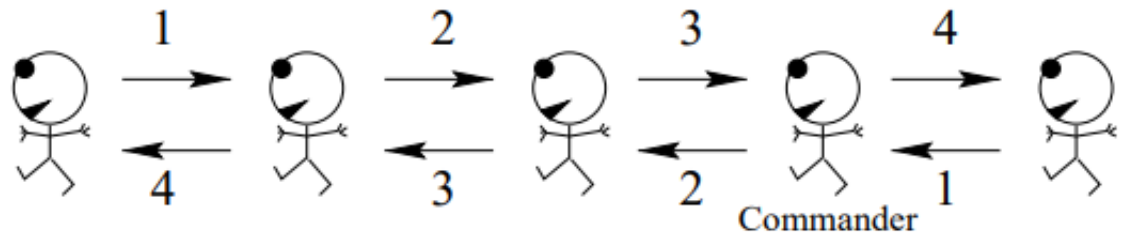


Message Passing: Count the Soldiers

If you are the front soldier in the line, say the number 'one' to the soldier behind you.

If you are the rearmost soldier in the line, say the number 'one' to the soldier in front of you.

If a soldier ahead of or behind you says a number to you, add one to it, and say the new number to the soldier on the other side



Sum-Product Algorithm

Main idea: message passing

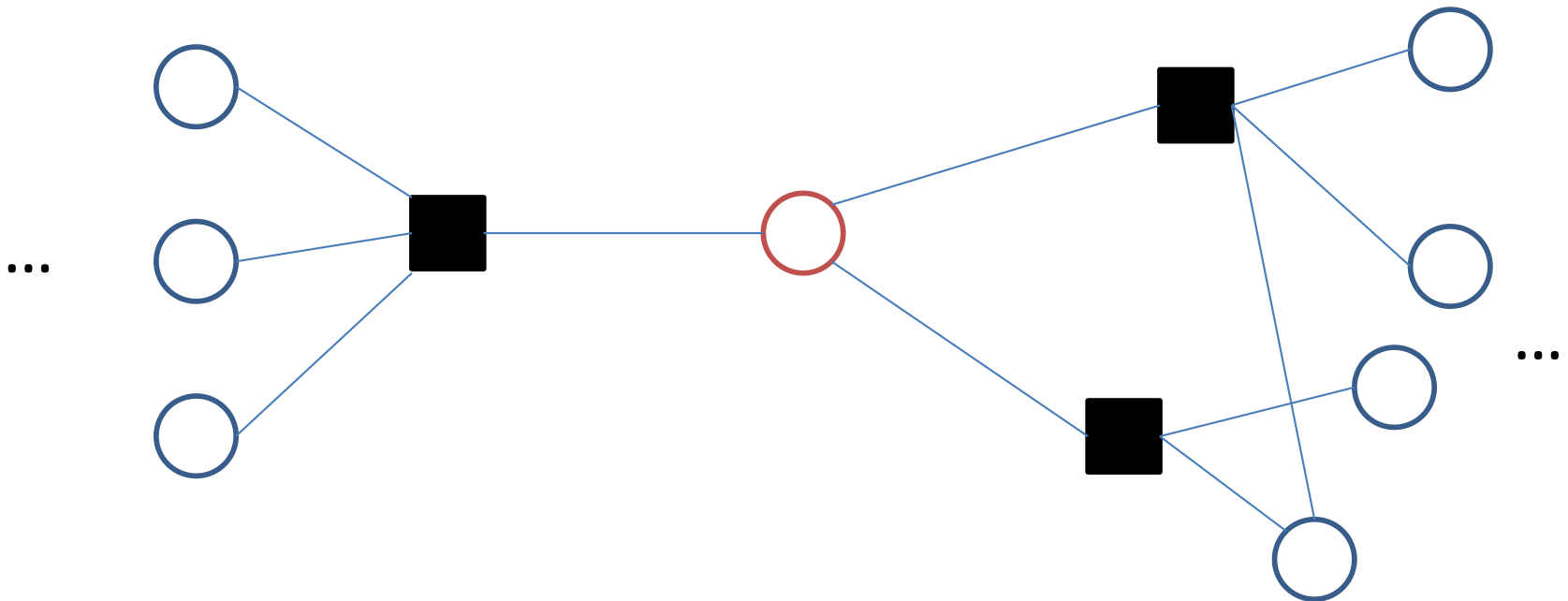
An exact inference algorithm for tree-like graphs

Belief propagation (forward-backward for HMMs) is a special case

Sum-Product

*definition of
marginal*

$$p(\textcolor{red}{x}_i = v) = \prod_{x: x_i = v} p(x_1, x_2, \dots, x_i, \dots, x_N)$$

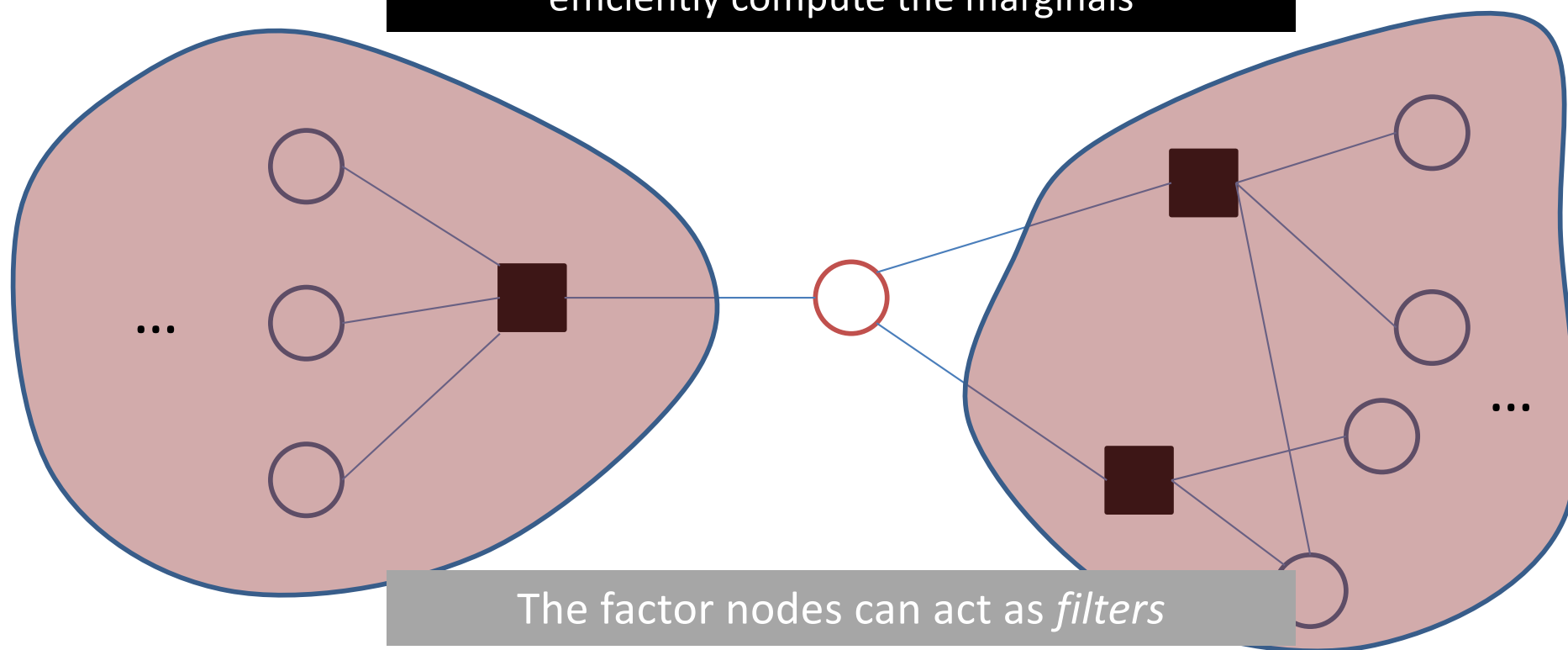


Sum-Product

*definition of
marginal*

$$p(\mathbf{x}_i = v) = \prod_{\mathbf{x}: \mathbf{x}_i = v} p(x_1, x_2, \dots, x_i, \dots, x_N)$$

main idea: use **bipartite** nature of graph to
efficiently compute the marginals

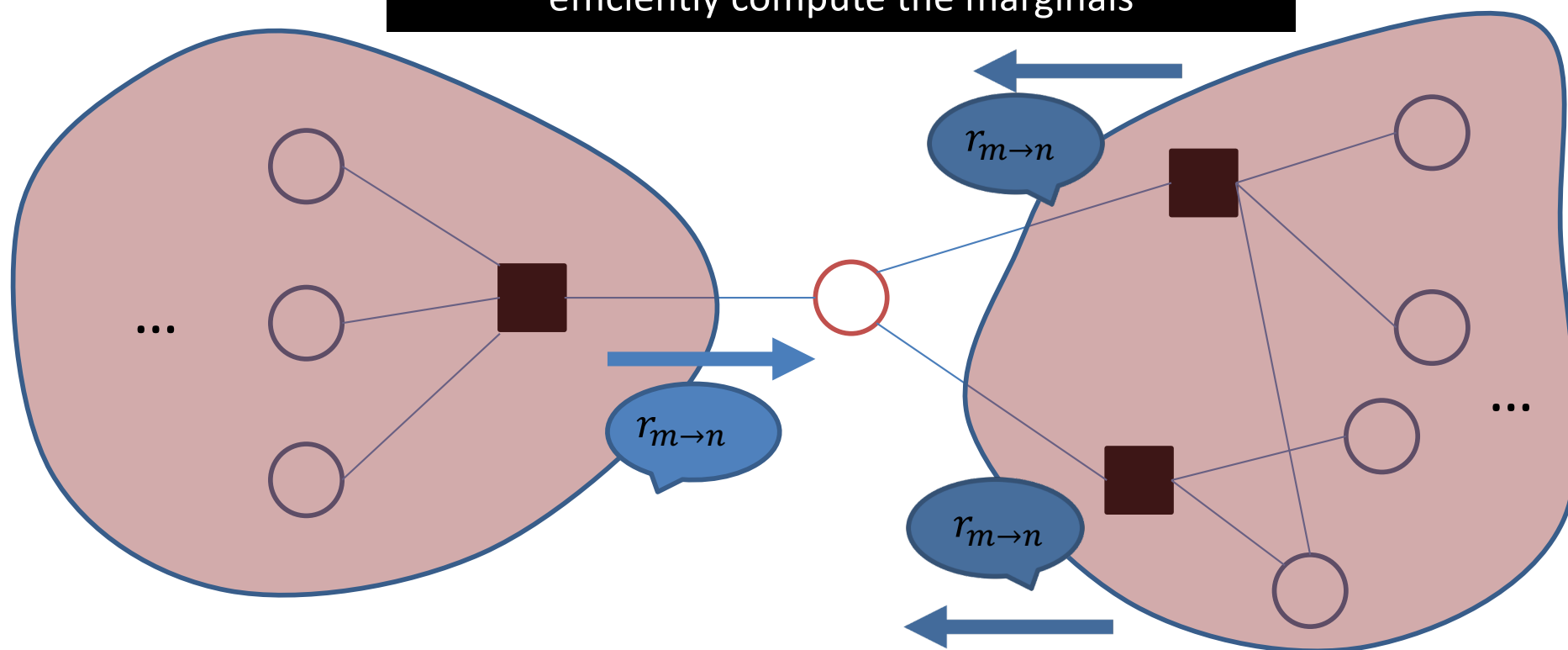


Sum-Product

*definition of
marginal*

$$p(\mathbf{x}_i = v) = \prod_{\mathbf{x}:\mathbf{x}_i=v} p(x_1, x_2, \dots, x_i, \dots, x_N)$$

main idea: use **bipartite** nature of graph to
efficiently compute the marginals

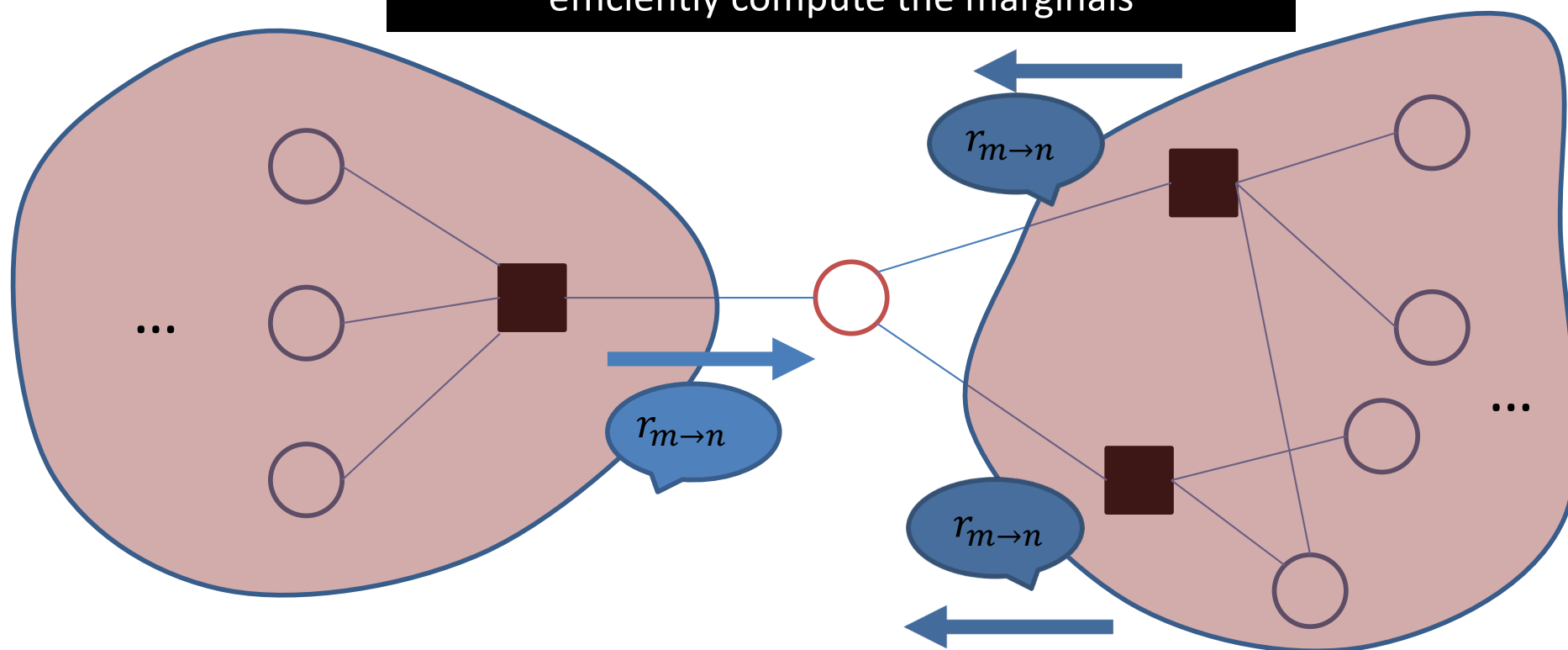


Sum-Product

*alternative
marginal
computation*

$$p(x_i = v) = \prod_f r_{f \rightarrow x_i}(x_i)$$

main idea: use **bipartite** nature of graph to efficiently compute the marginals



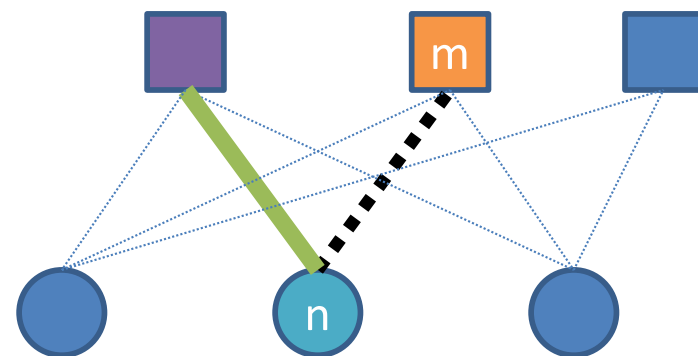
Sum-Product

From **variables** to **factors**

$$q_{\underline{n} \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

*set of factors in which
variable n participates*

*default value of 1 if
empty product*

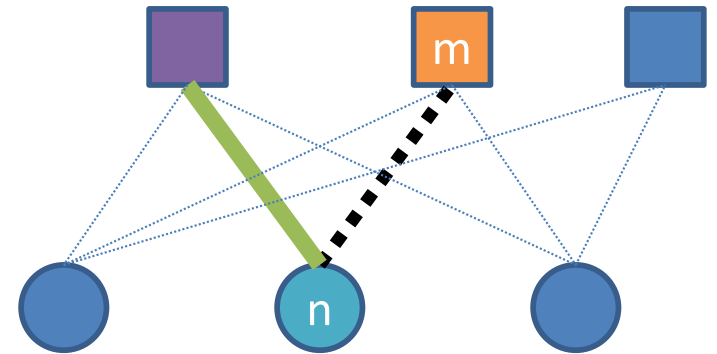


Sum-Product

From **variables** to **factors**

$$q_{\underline{n} \rightarrow \underline{m}}(x_n) = \prod_{\underline{m}' \in M(n) \setminus m} r_{\underline{m}' \rightarrow \underline{n}}(x_n)$$

set of factors in which
variable n participates



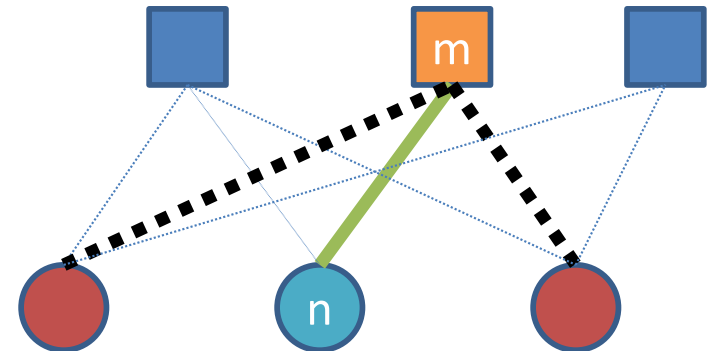
From **factors** to **variables**

$$r_{\underline{m} \rightarrow \underline{n}}(x_n) = \sum_{\underline{w}_m \setminus n} f_m(\underline{w}_m) \prod_{\underline{n}' \in N(m) \setminus n} q_{\underline{n}' \rightarrow \underline{m}}(x_{n'})$$

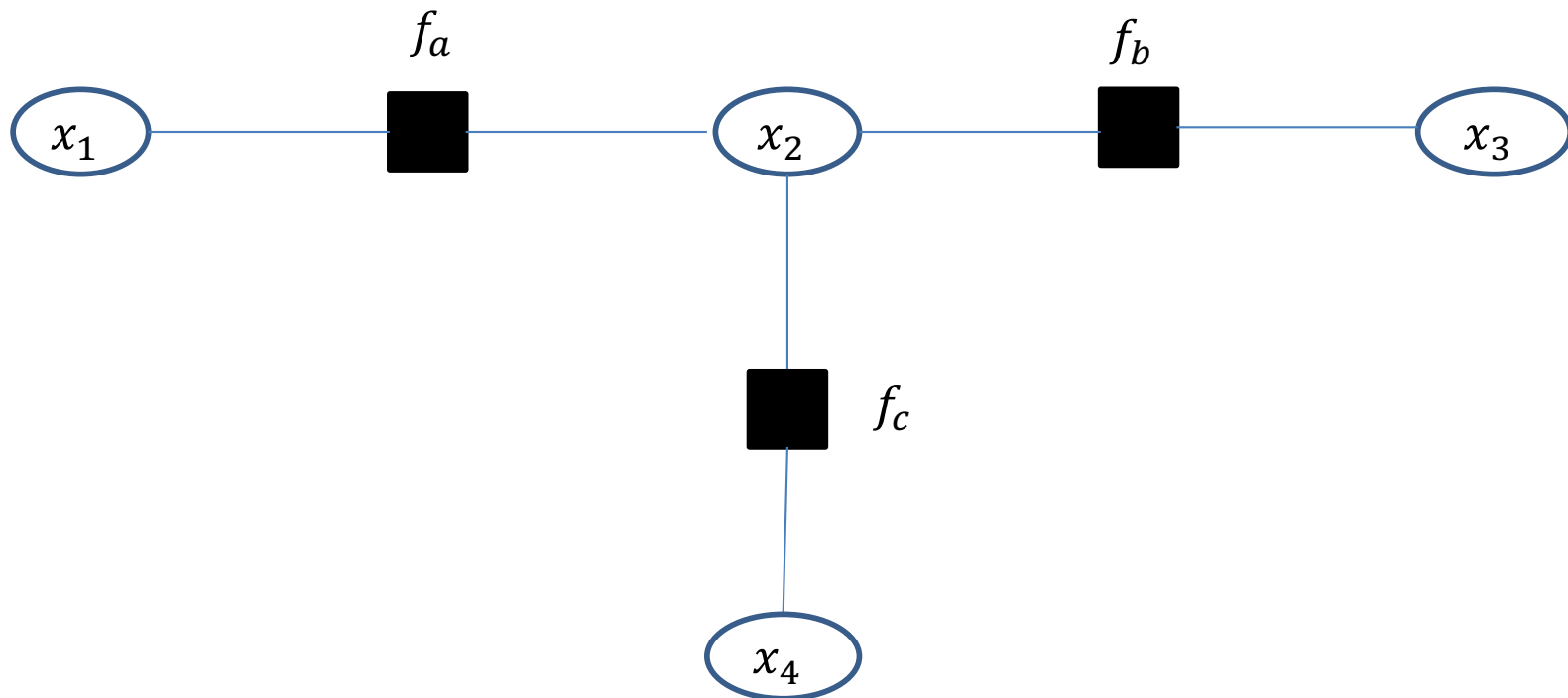
default value of 1 if
empty product

sum over configuration of
variables for the m^{th} factor,
with variable n fixed

set of variables that the
 m^{th} factor depends on

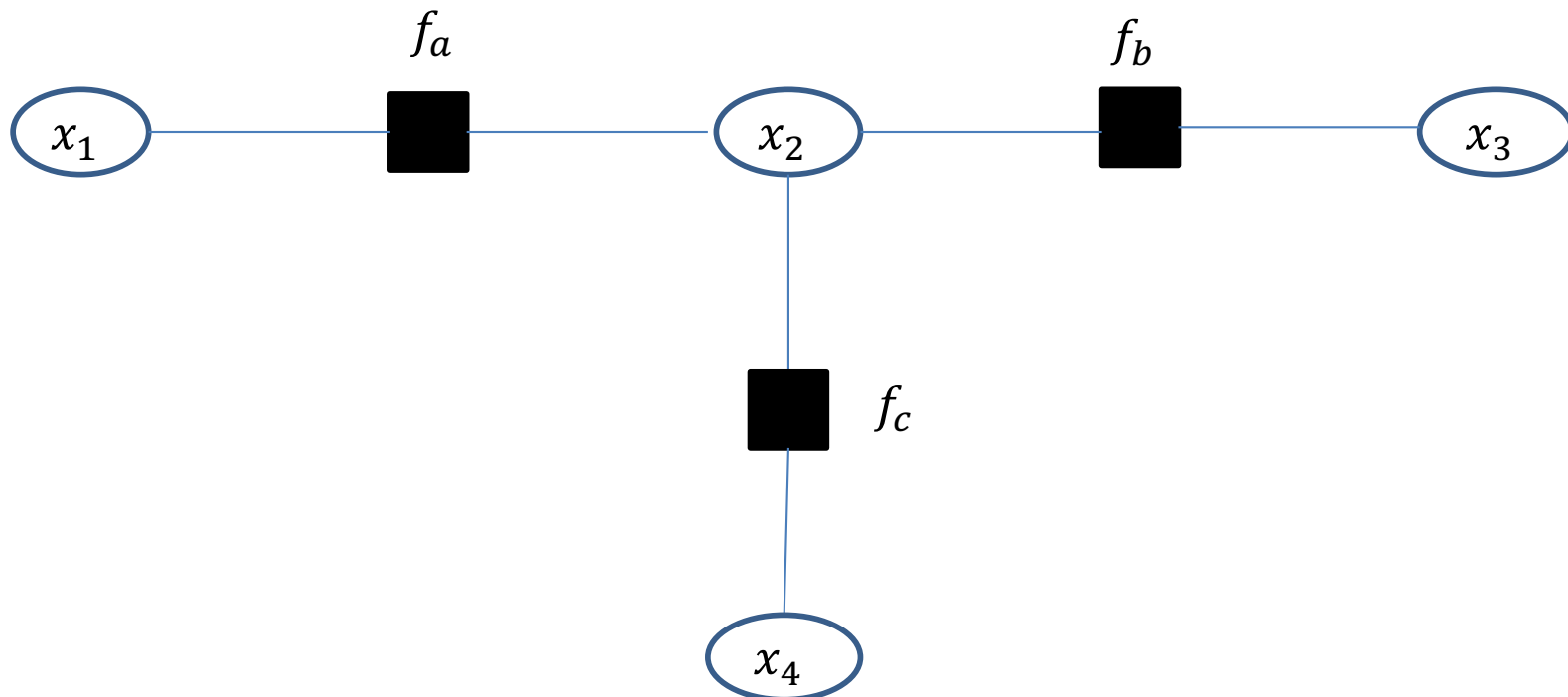


Example



Q: What are the variables?

Example

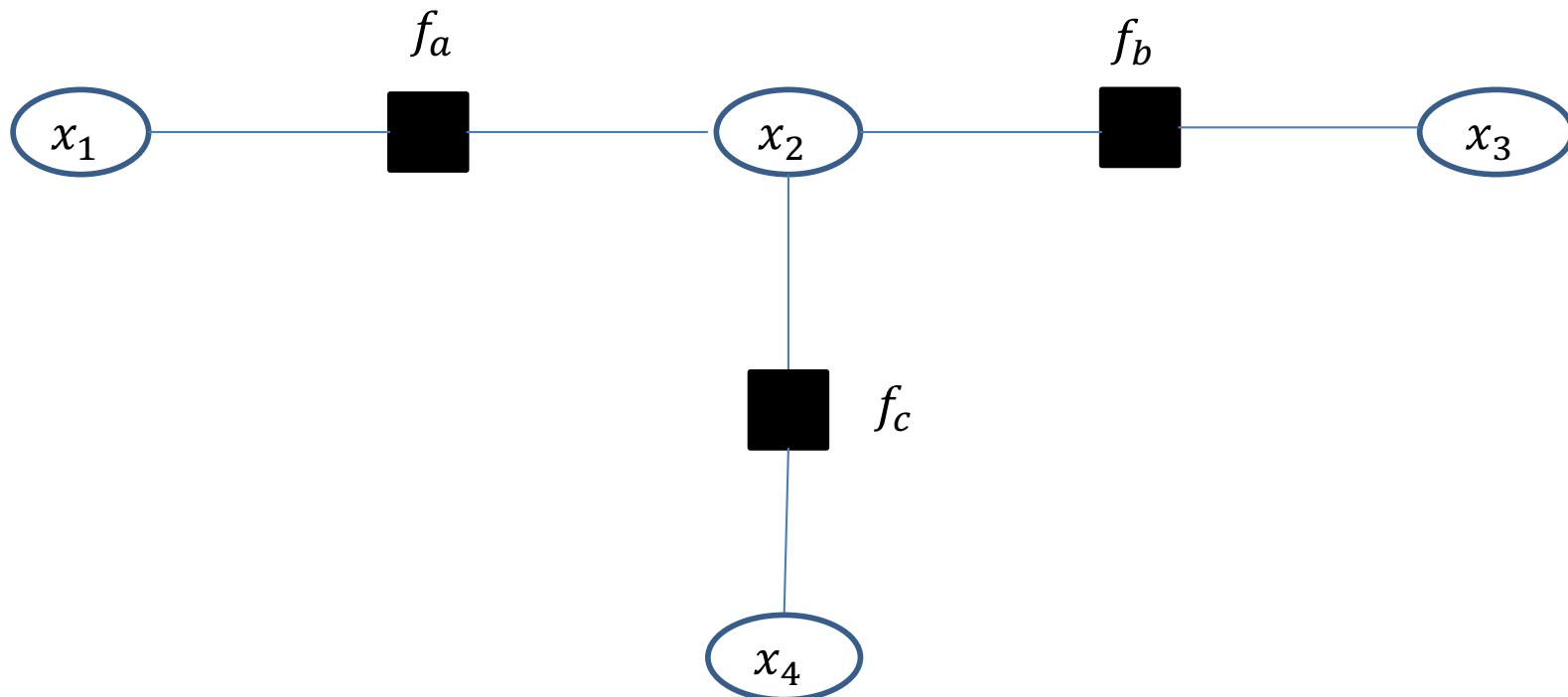


Q: What are the variables?

A:
 x_1, x_2, x_3, x_4

Q: What are the factors?

Example



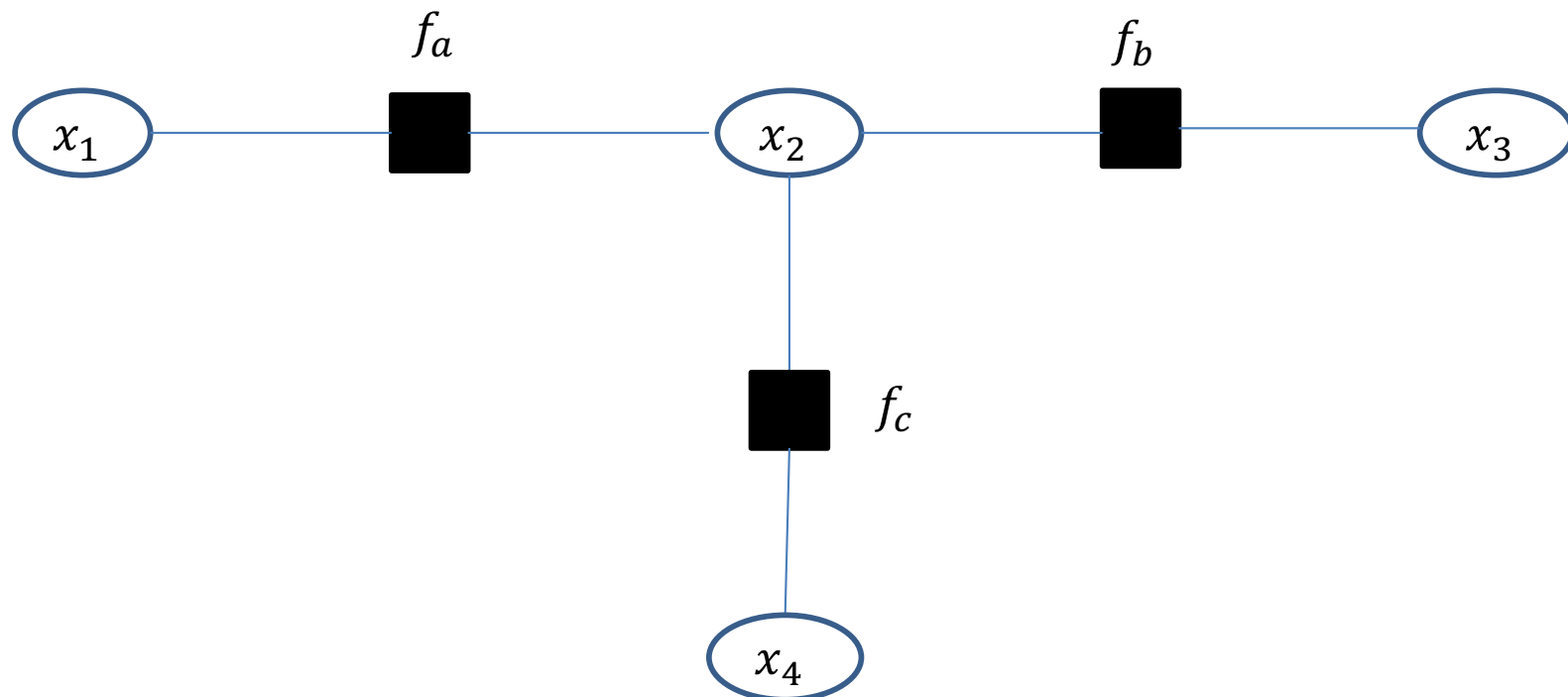
Q: What are the variables?

A:
 x_1, x_2, x_3, x_4

Q: What are the factors?

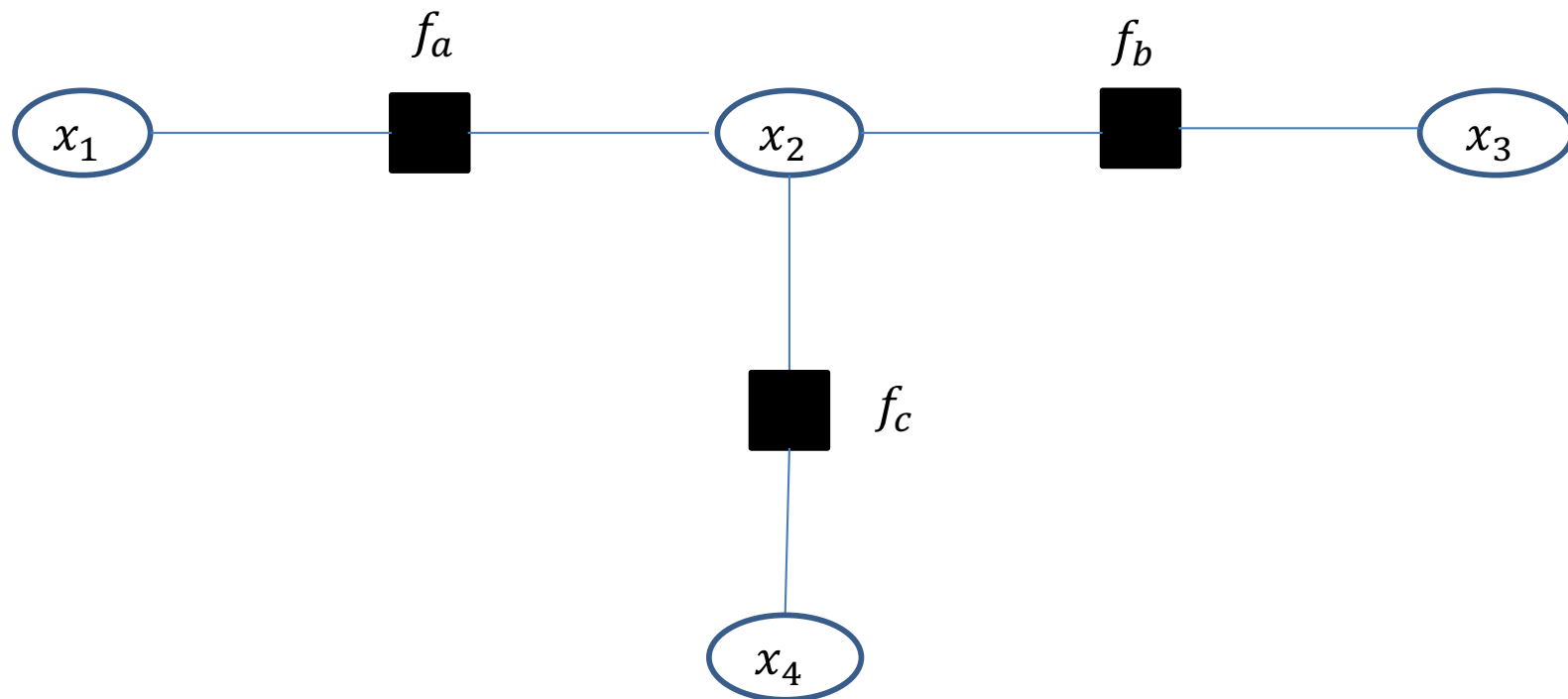
A: $f_a(x_1, x_2)$,
 $f_b(x_2, x_3)$,
 $f_c(x_2, x_4)$

Example



Q: What is the distribution we're modeling?

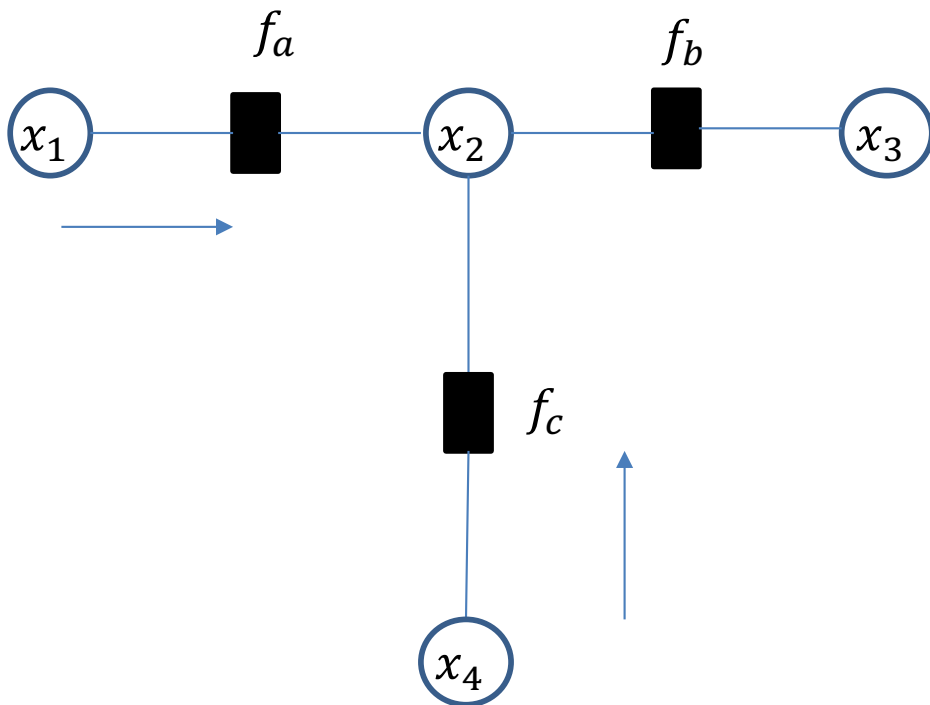
Example



Q: What is the distribution we're modeling?

A:
$$p(x_1, x_2, x_3, x_4) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4)$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root

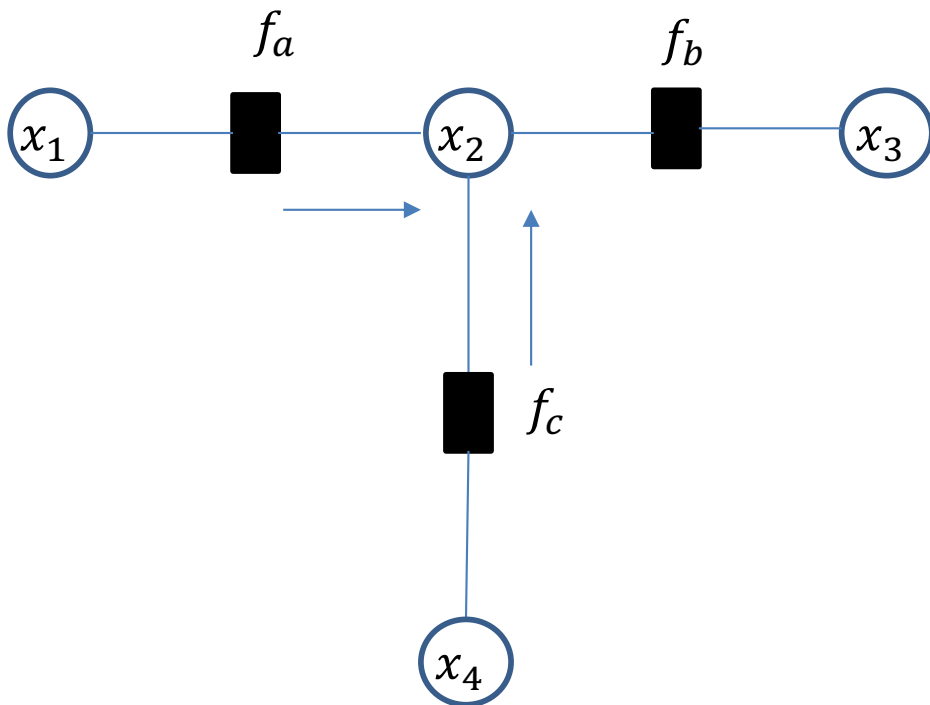
$$q_{x_1 \rightarrow f_a}(x_1) = 1$$

$$q_{x_4 \rightarrow f_c}(x_4) = 1$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)

1. Send messages from leaves to root

$$q_{x_1 \rightarrow f_a}(x_1) = 1$$

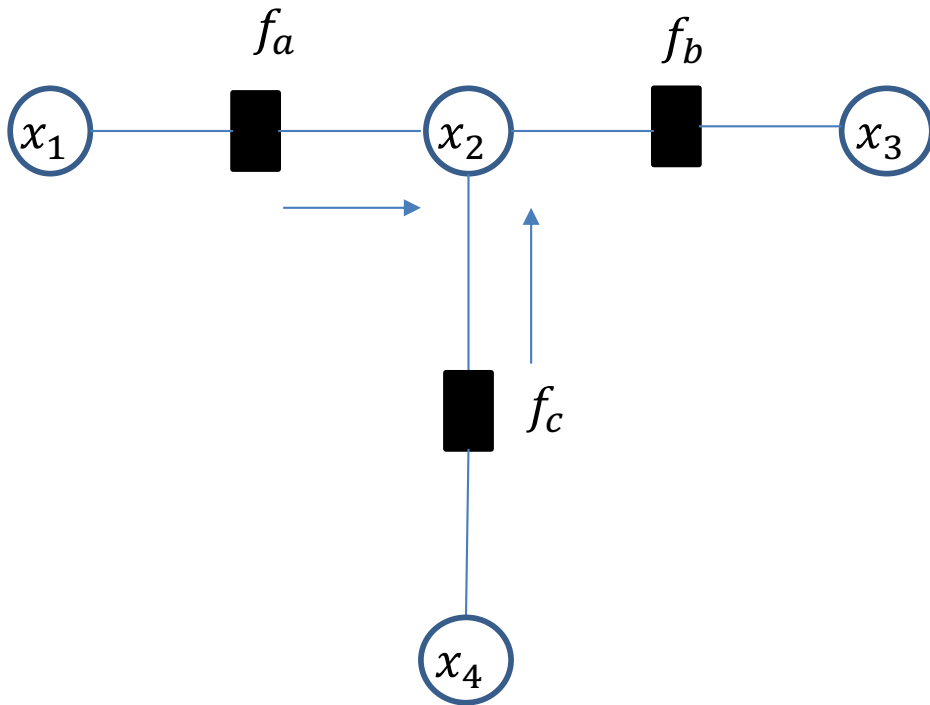
$$q_{x_4 \rightarrow f_c}(x_4) = 1$$

$$r_{f_a \rightarrow x_2}(x_2) = ???$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)

1. Send messages from leaves to root

$$q_{x_1 \rightarrow f_a}(x_1) = 1$$

$$q_{x_4 \rightarrow f_c}(x_4) = 1$$

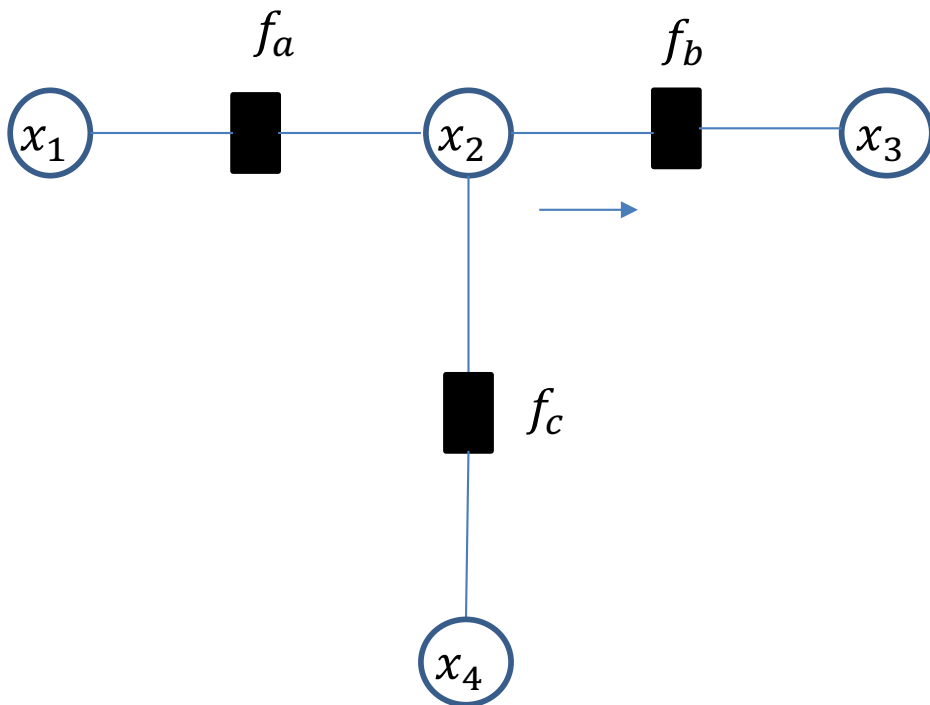
$$r_{f_a \rightarrow x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

$$r_{f_c \rightarrow x_2}(x_2) = \sum_k f_c(x_2, x_4 = k)$$

$$q_{\textcolor{teal}{n} \rightarrow \textcolor{brown}{m}}(x_n) = \prod_{\textcolor{violet}{m}' \in M(n) \setminus m} r_{\textcolor{green}{m}' \rightarrow \textcolor{green}{n}}(x_n)$$

$$r_{\textcolor{brown}{m} \rightarrow \textcolor{teal}{n}}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{\textcolor{red}{n}' \in N(m) \setminus n} q_{\textcolor{red}{n}' \rightarrow \textcolor{brown}{m}}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)

1. Send messages from leaves to root

$$q_{x_1 \rightarrow f_a}(x_1) = 1$$

$$q_{x_4 \rightarrow f_c}(x_4) = 1$$

$$r_{f_a \rightarrow x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

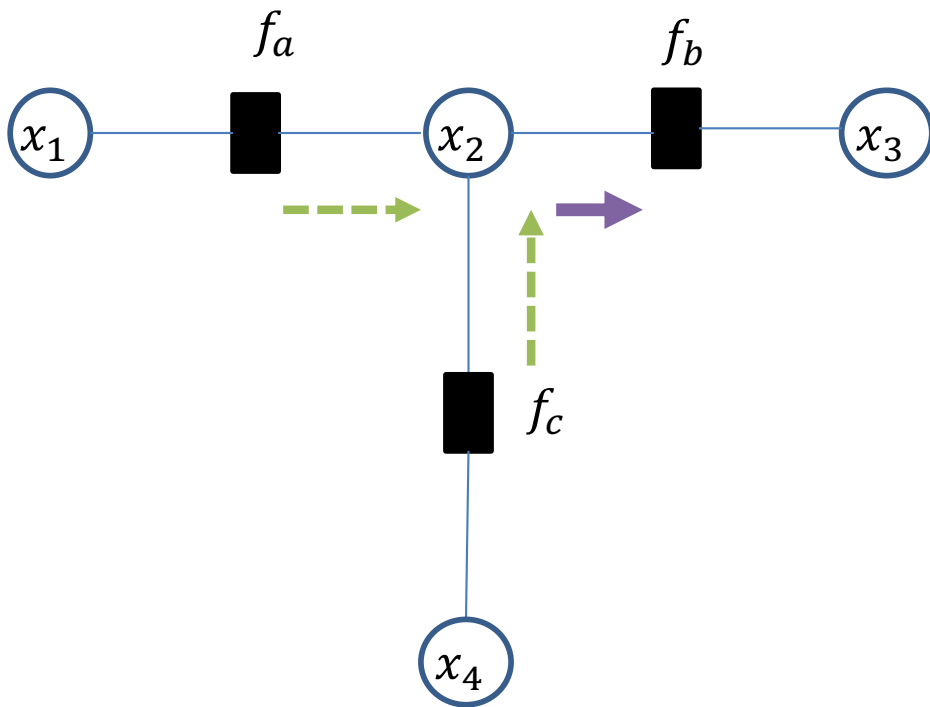
$$r_{f_c \rightarrow x_2}(x_2) = \sum_k f_c(x_2, x_4 = k)$$

$$q_{x_2 \rightarrow f_b}(x_2) = ???$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root

$$q_{x_1 \rightarrow f_a}(x_1) = 1$$

$$q_{x_4 \rightarrow f_c}(x_4) = 1$$

$$r_{f_a \rightarrow x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

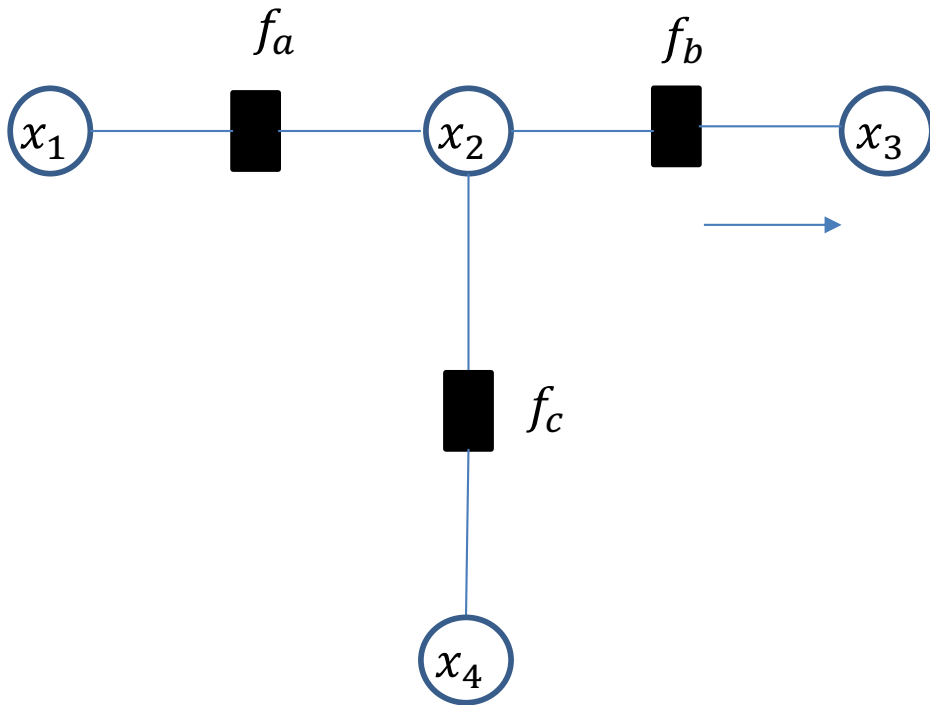
$$r_{f_c \rightarrow x_2}(x_2) = \sum_k f_c(x_2, x_4 = k)$$

$$q_{x_2 \rightarrow f_b}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root

$$q_{x_1 \rightarrow f_a}(x_1) = 1$$

$$q_{x_4 \rightarrow f_c}(x_4) = 1$$

$$r_{f_a \rightarrow x_2}(x_2) = \sum_k f_a(x_1 = k, x_2)$$

$$r_{f_c \rightarrow x_2}(x_2) = \sum_k f_c(x_2, x_4 = k)$$

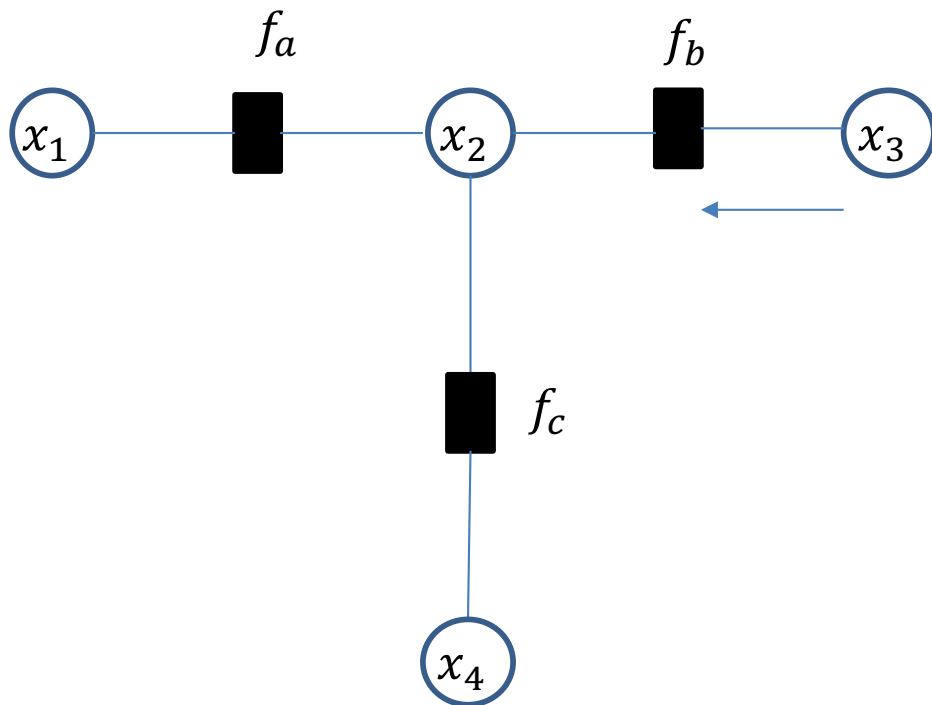
$$q_{x_2 \rightarrow f_b}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

$$r_{f_b \rightarrow x_3}(x_3) = \sum_k f_b(x_2 = k, x_3)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



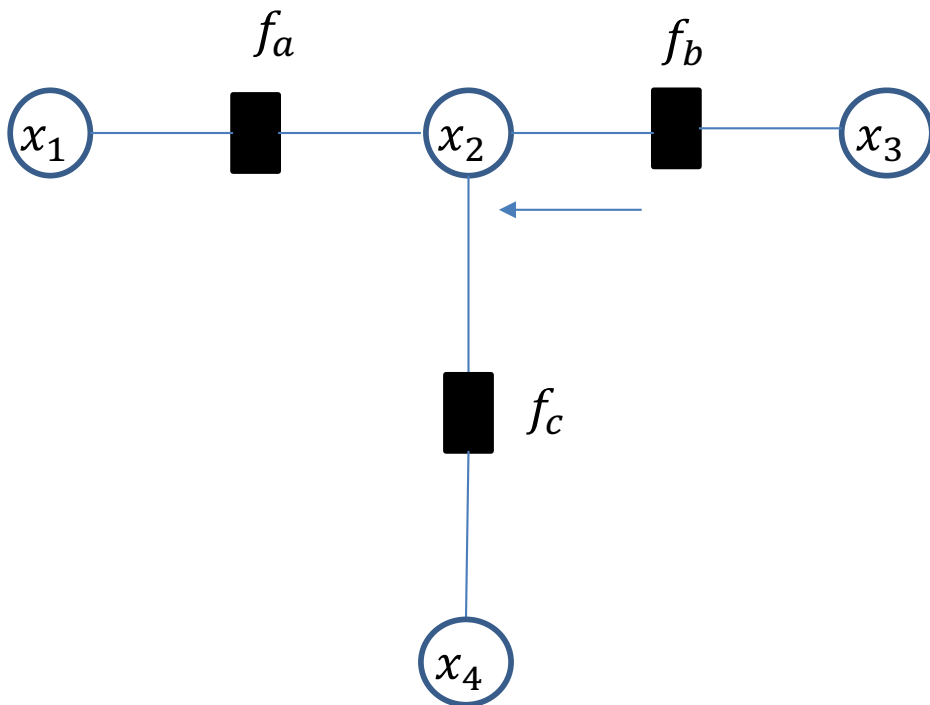
1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

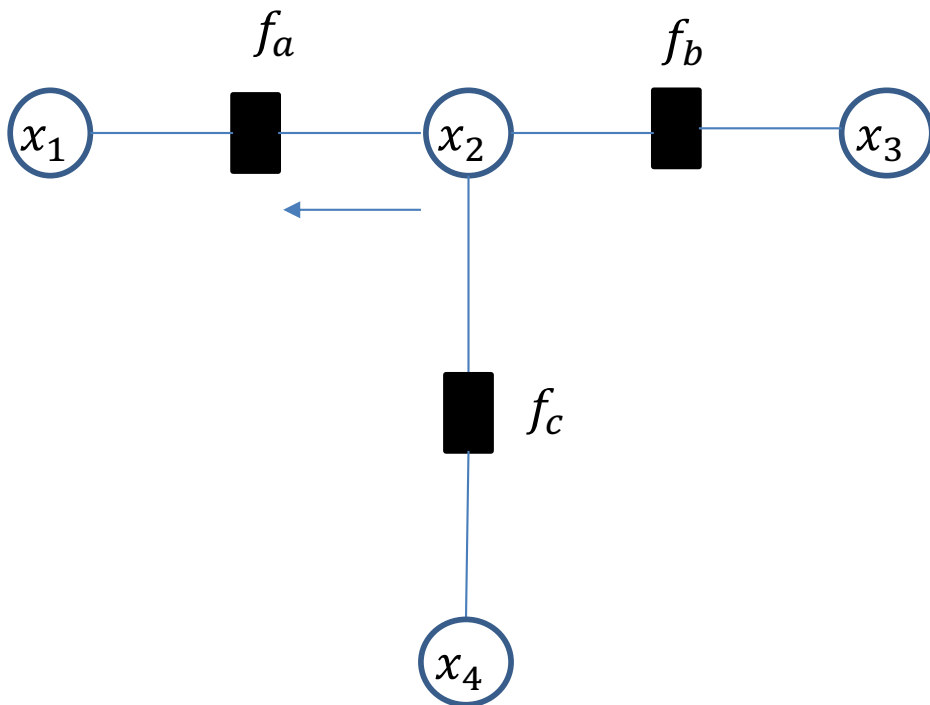
$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

$$r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

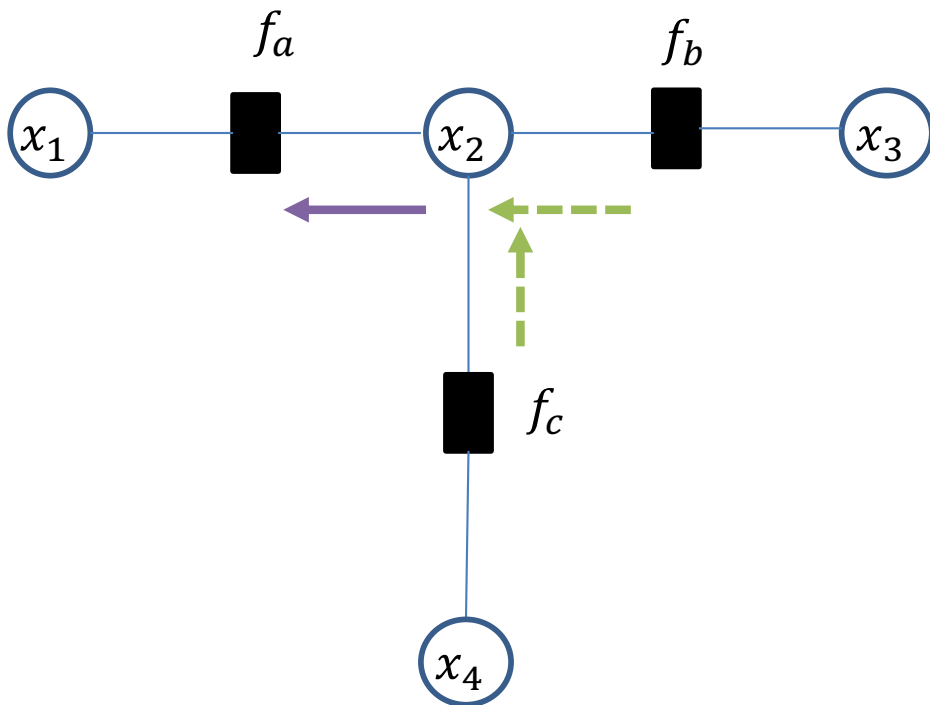
$$r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \rightarrow f_a}(x_2) = ???$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

$$r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

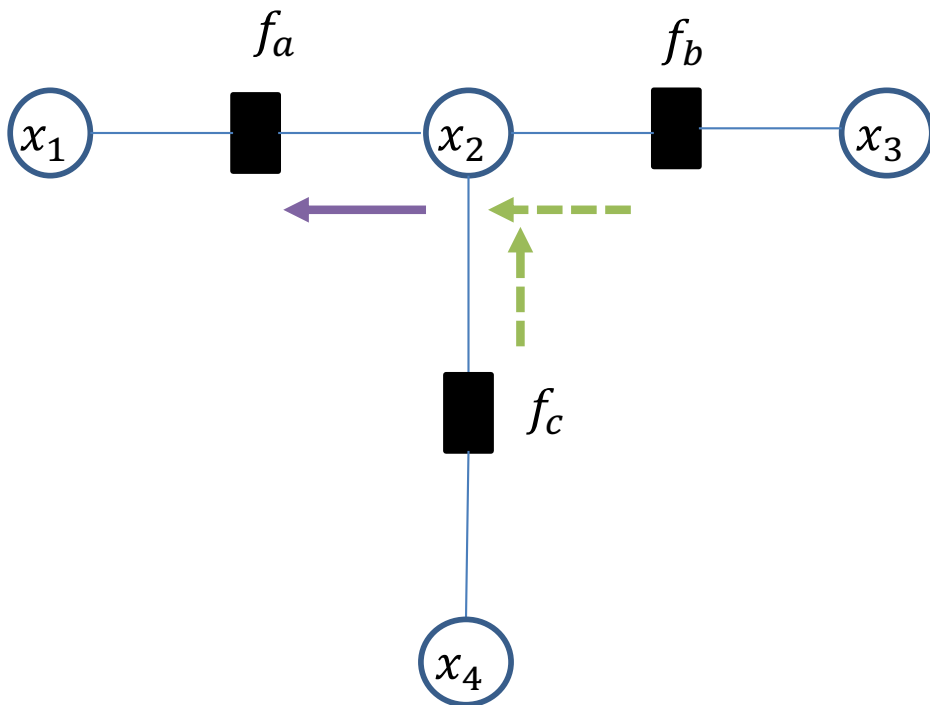
We just
computed this

Q: Where did we
compute this?

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

$$r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

We just
computed this

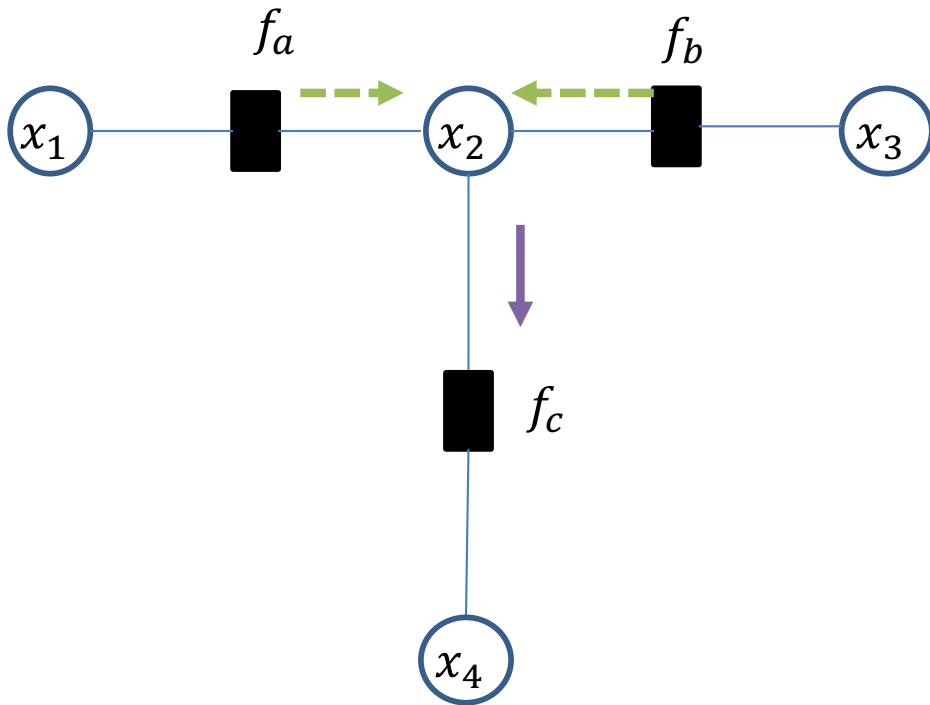
Q: Where did we
compute this?

A: In step 1
(leaves $\rightarrow$ root)

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

$$r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

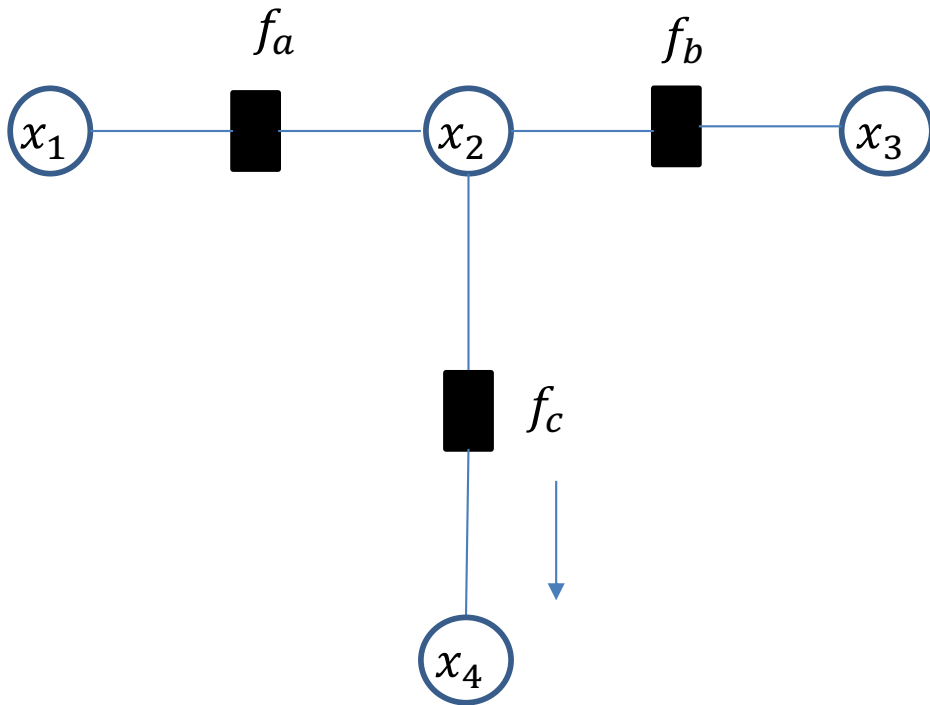
$$q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

$$q_{x_2 \rightarrow f_c}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_b \rightarrow x_2}(x_2)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

$$r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

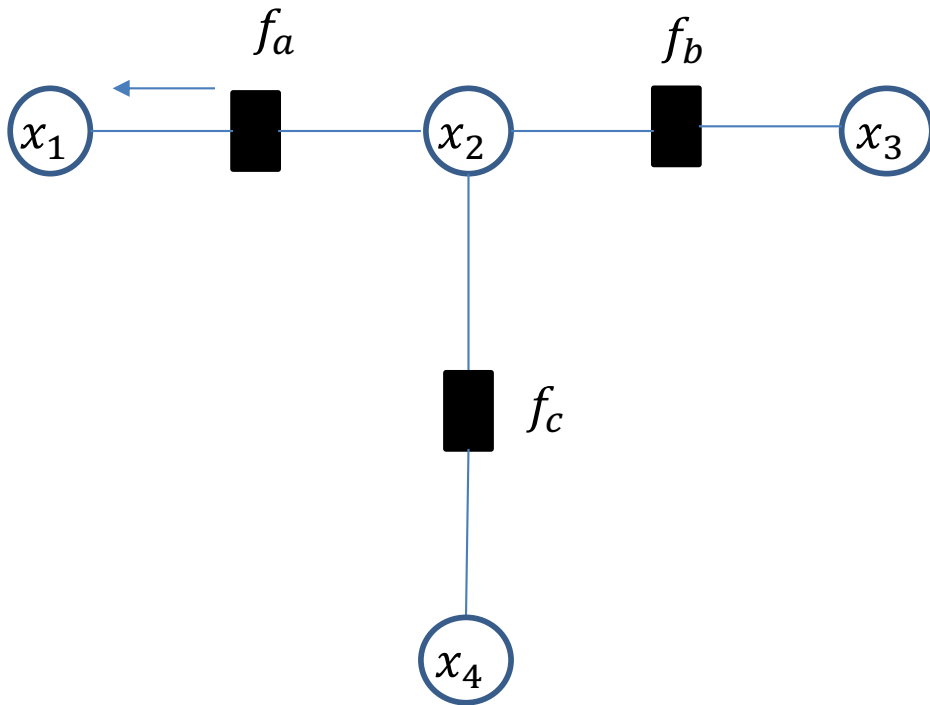
$$q_{x_2 \rightarrow f_c}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_b \rightarrow x_2}(x_2)$$

$$r_{f_c \rightarrow x_4}(x_4) = \sum_k f_c(x_2 = k, x_4)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves

$$q_{x_3 \rightarrow f_b}(x_3) = 1$$

$$r_{f_b \rightarrow x_2}(x_2) = \sum_k f_b(x_2, x_3 = k)$$

$$q_{x_2 \rightarrow f_a}(x_2) = r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

$$q_{x_2 \rightarrow f_c}(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_b \rightarrow x_2}(x_2)$$

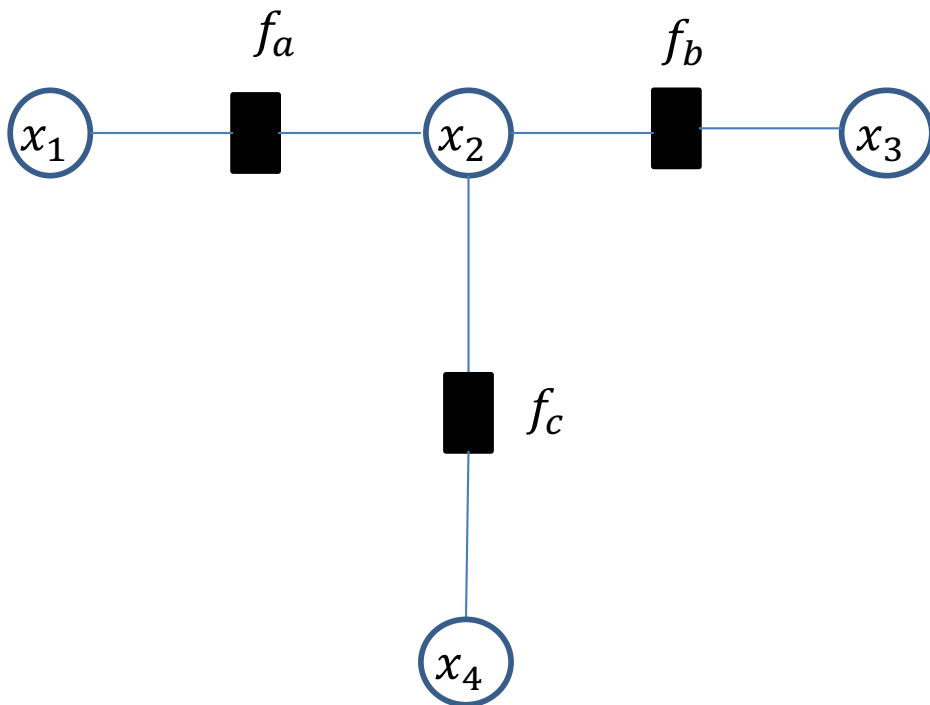
$$r_{f_c \rightarrow x_4}(x_4) = \sum_k f_c(x_2 = k, x_4)$$

$$r_{f_a \rightarrow x_1}(x_1) = \sum_k f_a(x_1, x_2 = k)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{w_m \setminus n} f_m(w_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



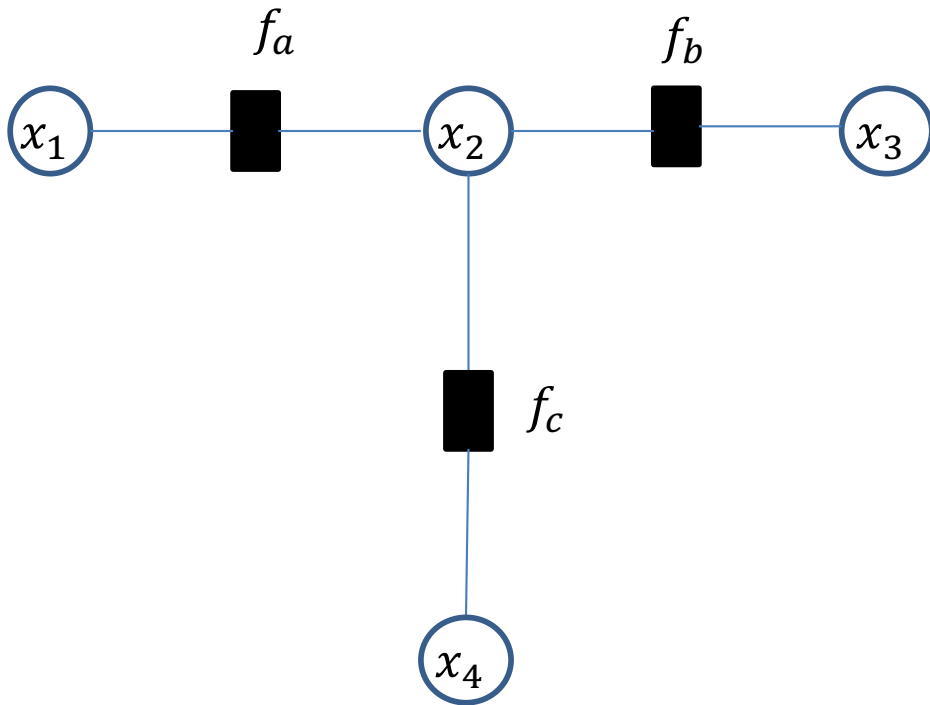
1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves
 3. Use messages to compute marginal probabilities

$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves
 3. Use messages to compute marginal probabilities

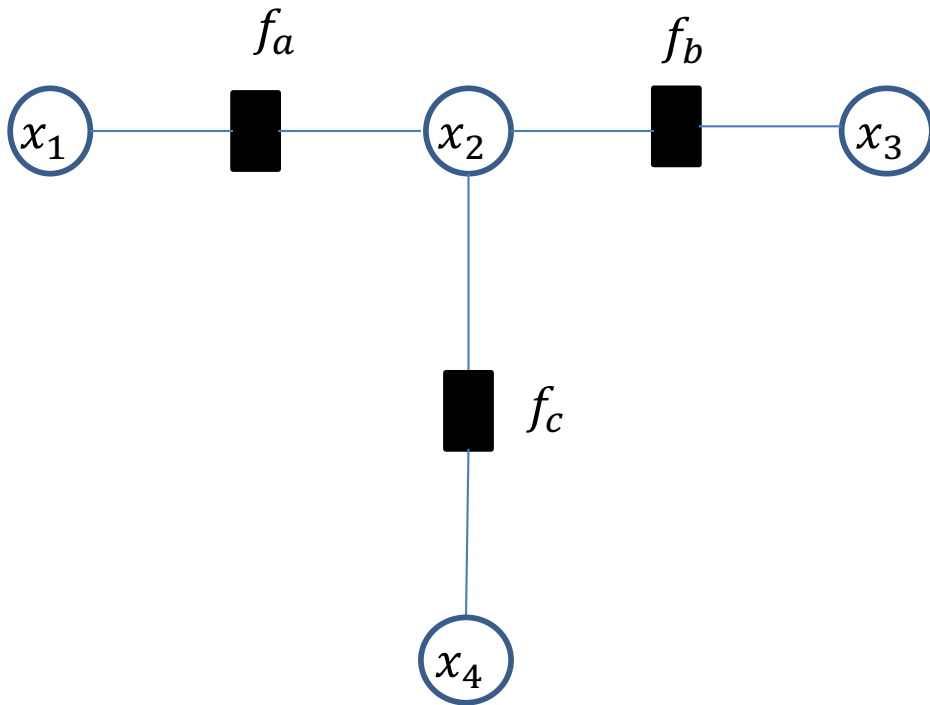
$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$p(x_1) = r_{f_a \rightarrow x_1}(x_1)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves
 3. Use messages to compute marginal probabilities

$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

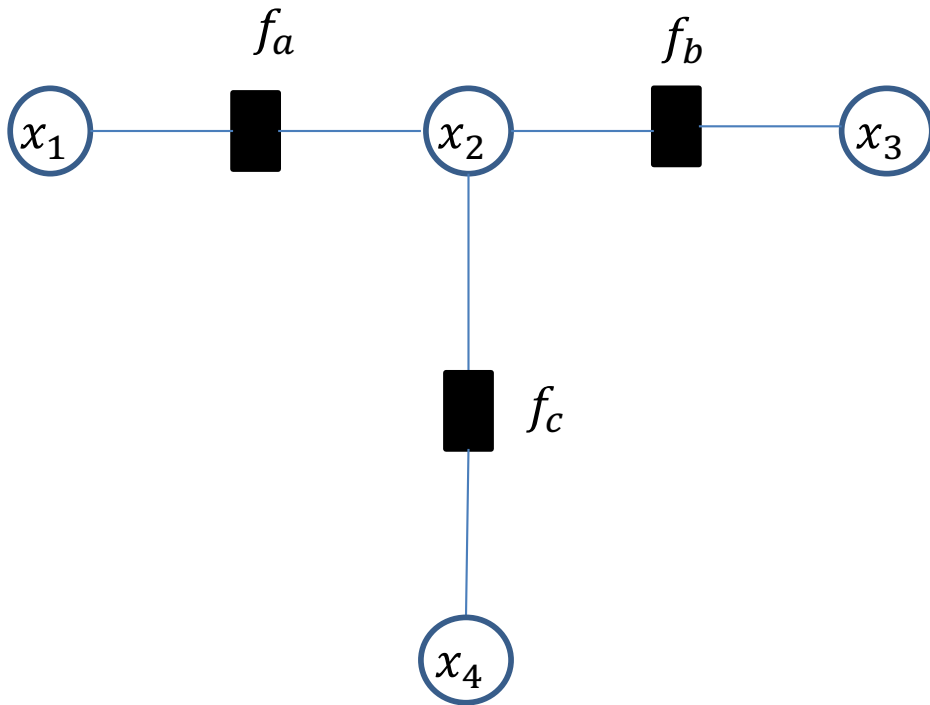
$$p(x_1) = r_{f_a \rightarrow x_1}(x_1)$$

$$p(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves
 3. Use messages to compute marginal probabilities

$$p(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$p(x_1) = r_{f_a \rightarrow x_1}(x_1)$$

$$p(x_2) = r_{f_a \rightarrow x_2}(x_2) r_{f_b \rightarrow x_2}(x_2) r_{f_c \rightarrow x_2}(x_2)$$

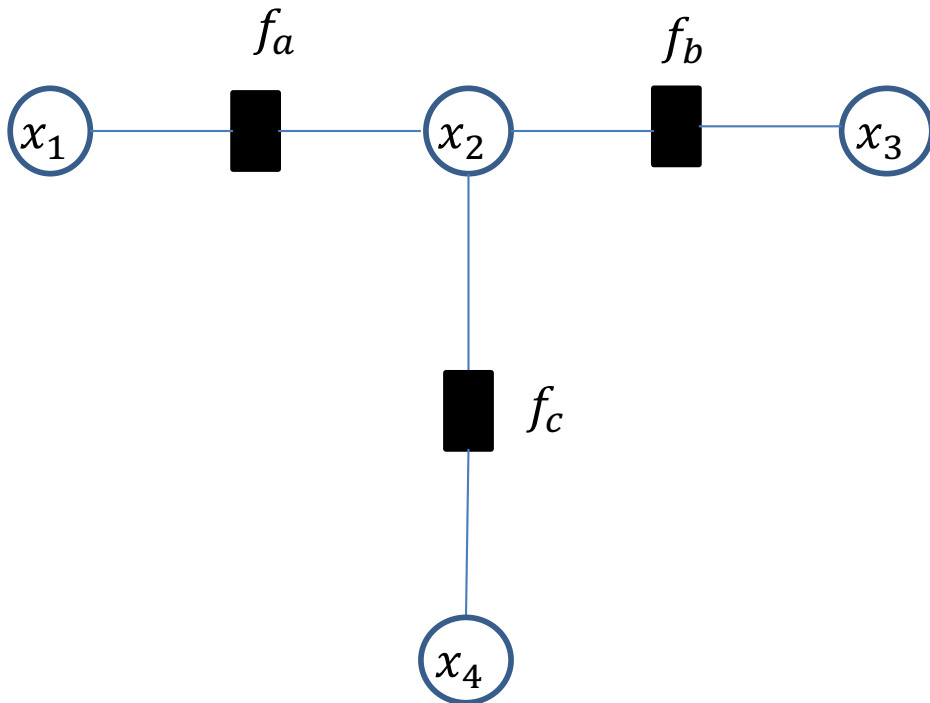
$$p(x_3) = r_{f_b \rightarrow x_3}(x_3)$$

$$p(x_4) = r_{f_c \rightarrow x_4}(x_4)$$

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example

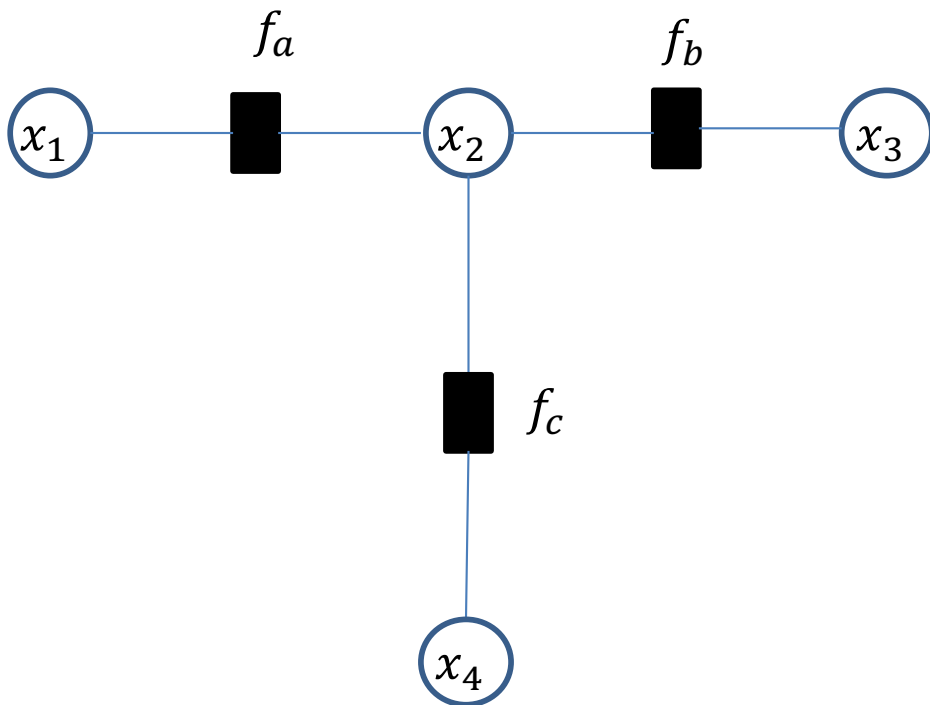


1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves
 3. Use messages to compute marginal probabilities
2. Are we done?
 1. If a tree structure, we've converged
 - 2.

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example

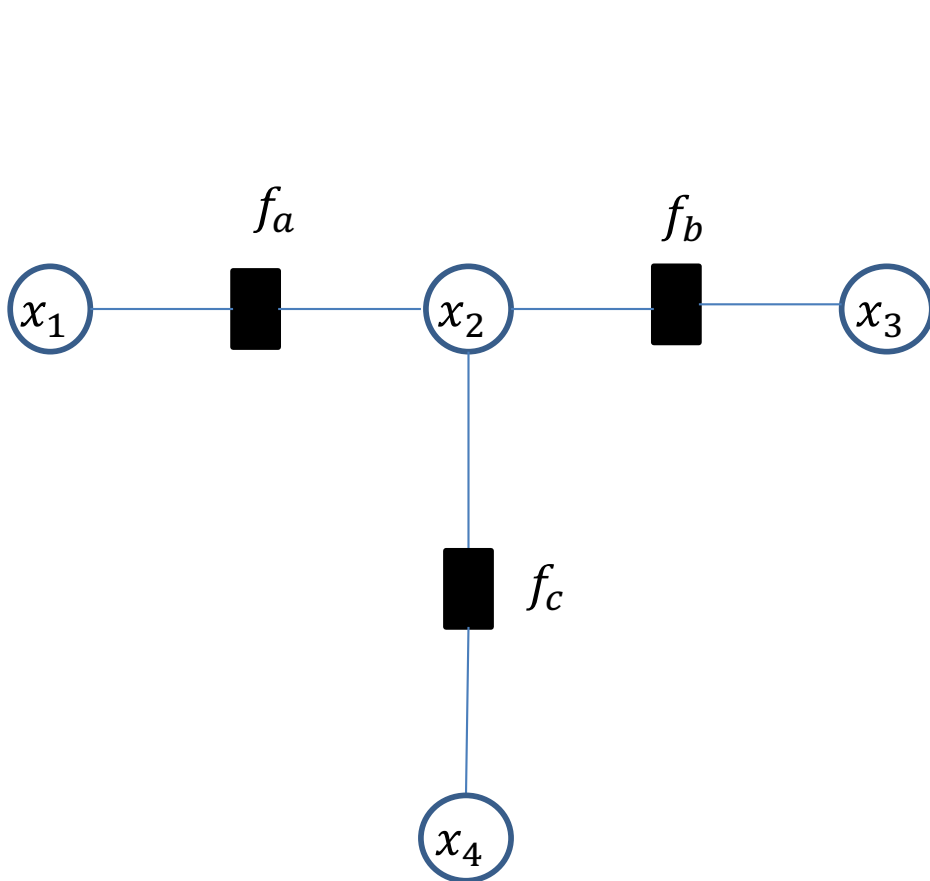


1. Select the root, or pick one if a tree (x_3)
 1. Send messages from leaves to root
 2. Send messages from root to leaves
 3. Use messages to compute marginal probabilities
2. Are we done?
 1. If a tree structure, we've converged
 2. If not:
 1. Either accept the partially converged result, or...
 - 2.

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Example



1. Select the root, or pick one if a tree
 1. Send messages from leaves to root
 2. Send messages from root to leaves
 3. Use messages to compute marginal probabilities
2. Are we done?
 1. If a tree structure, we've converged
 2. If not:
 1. Either accept the partially converged result, or...
 2. Go back to (1) and repeat

[Loopy BP]

$$q_{n \rightarrow m}(x_n) = \prod_{m' \in M(n) \setminus m} r_{m' \rightarrow n}(x_n)$$

$$r_{m \rightarrow n}(x_n) = \sum_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

Max-Product (Max-Sum)

Problem: how to find the most likely (best) setting of latent variables

Replace sum (+) with max in factor $\rightarrow$ variable computations

$$r_{m \rightarrow n}(x_n) = \max_{\mathbf{w}_m \setminus n} f_m(\mathbf{w}_m) \prod_{n' \in N(m) \setminus n} q_{n' \rightarrow m}(x_{n'})$$

(why max-sum? computationally,
implement with logs)

Loopy Belief Propagation

Sum-product algorithm is not exact for general graphs

Loopy Belief Propagation (Loopy BP): run sum-product algorithm *anyway* and hope for the best

Requires a **message passing schedule**

Outline

Directed Graphical Models

Naïve Bayes

Undirected Graphical Models

Factor Graphs

Ising Model

Message Passing: Graphical Model Inference