CSE 847: Statistical Machine Learning

Graphical Models

Jiayu Zhou
Computer Science & Engineering
Michigan State University

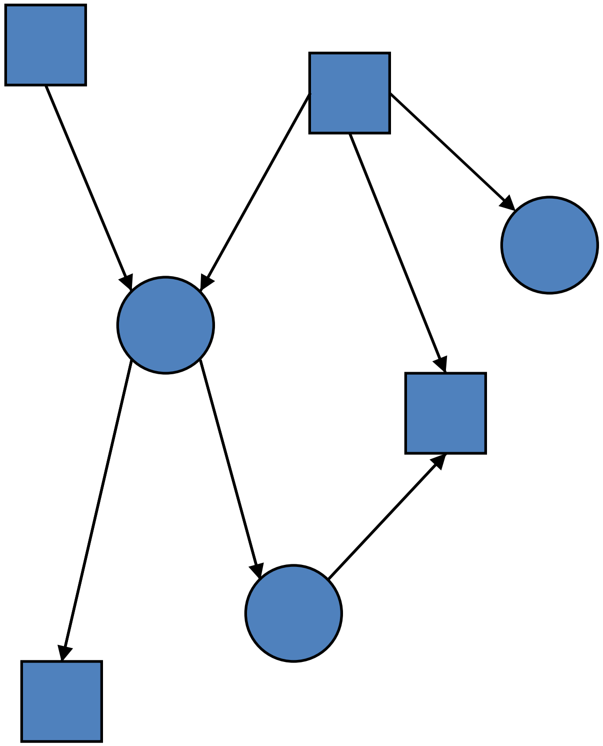
What is a graphical model ?

A graphical model is a way of representing probabilistic relationships between random variables.

Conditional (in)dependencies are represented by (missing) edges:

Undirected edges simply give correlations between variables (**Markov Random Field** or **Undirected Graphical model**):

Directed edges give causality relationships (**Bayesian Network** or **Directed Graphical Model**):



“Graphical models are a **marriage between probability theory and graph theory**.

They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – **uncertainty and complexity** –

and in particular they are playing an increasingly important role in the design and analysis of **machine learning algorithms**.

Fundamental to the idea of a graphical model is the **notion of modularity** – a complex system is built by combining simpler parts.

The graphical model framework provides a way to view all of these systems as instances of a **common underlying formalism**.

This view has many advantages -- in particular, **specialized techniques** that have been developed in one field can be **transferred between research communities** and exploited more widely.

Moreover, the graphical model formalism provides a **natural framework for the design of new systems**."

--- Michael Jordan, 1998.

Why can we do with graphical models?

- Graphs are an intuitive way of representing and visualising the relationships between many variables. (Examples: family trees, electric circuit diagrams, neural networks)

- Graphical models allow us to define general message-passing algorithms that implement probabilistic inference efficiently. Thus we can answer queries like "What is $P(A|C = c)$?" without enumerating all settings of all variables in the model.

- A graph allows us to abstract out the conditional independence relationships between the variables from the details of their parametric forms. Thus we can answer questions like: "Is A dependent on B given that we know the value of C ?" just by looking at the graph.

Applications of graphical models

- Information extraction
- Speech recognition
- Computer vision
- Modeling of gene regulatory networks
- Gene finding and diagnosis of diseases
- Graphical models for protein structure

Probability Distributions

- Let $X_1,\ldots,X_p$ be discrete random variables
- Let P be a joint distribution over $X_1,\ldots,X_p$

- If the variables are binary, then we need $O(2^p)$ parameters to describe P

- Can we do better?
  - **Key idea**: use properties of independence

Independent Random Variables

- Two variables X and Y are **independent** if

  $P(X = x|Y = y) = P(X = x)$ for all values x, y

  That is, learning the values of Y does not change prediction of X

- If X and Y are independent then

  $P(X,Y) = P(X|Y)P(Y) = P(X)P(Y)$

- In general, if $X_1,\ldots,X_p$ are independent, then

  $P(X_1,\ldots,X_p)= P(X_1)\ldots P(X_p)$

Conditional Independence

- Unfortunately, most of random variables of interest are not independent of each other
- A more suitable notion is that of **conditional independence**
- Two variables X and Y are **conditionally independent** given Z if

  $P(X = x|Y = y, Z=z) = P(X = x|Z=z)$ for all values x,y,z

  That is, learning the values of Y does not change prediction of X once we know the value of Z

  notation: $X \perp Y \mid Z$

Example: Naïve Bayesian Model

- A common model in early diagnosis:
  - Symptoms are conditionally independent given the disease (or fault)
- Thus, if
  - $X_1,\ldots,X_p$ denote whether the symptoms exhibited by the patient (headache, high-fever, etc.) and
  - H denotes the hypothesis about the patients health

then, $P(X_1,\ldots,X_p,H) = P(H)P(X_1|H)\ldots P(X_p|H)$,

- This **naïve Bayesian** model allows compact representation
  - It does embody strong independence assumptions

Probabilistic Graphical Models I

- Probabilities play a central role in modern pattern recognition.
- The probabilistic inference and learning may be complex.
- It is advantageous to augment the analysis using diagrammatic representations of probability distributions, called probabilistic graphical models.

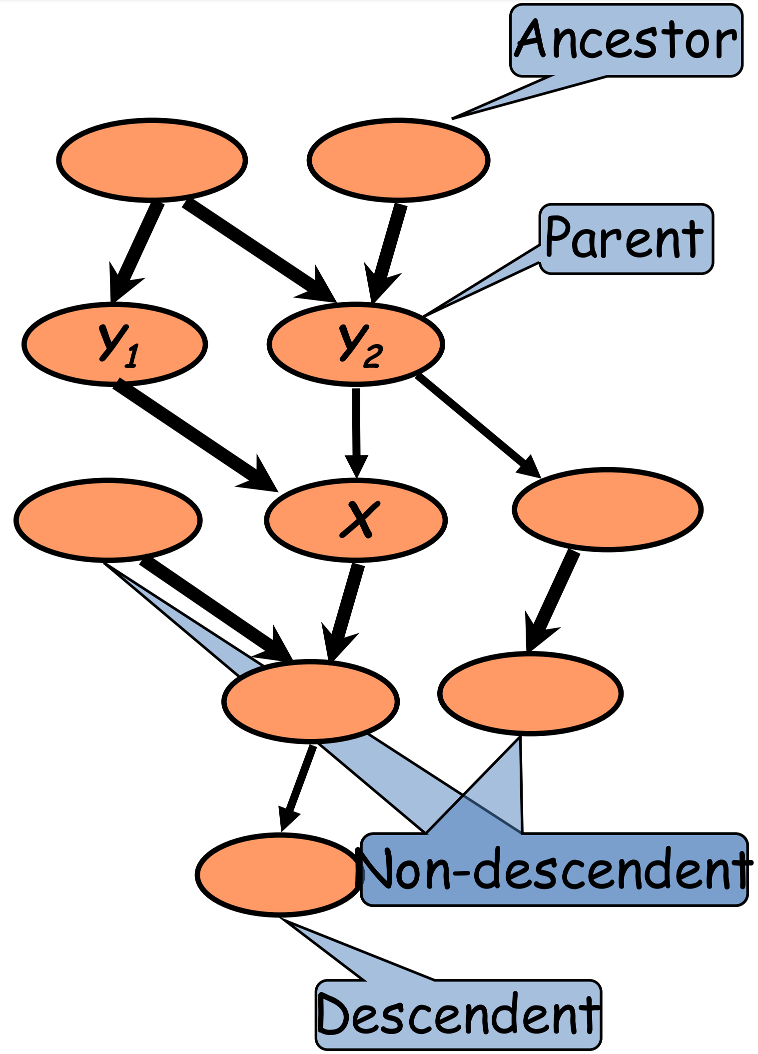
Probabilistic Graphical Models II

- Insights into the properties of the model, including **conditional independence properties**, can be obtained by inspection of the graph.

- Complex computations, required to perform inference and learning in sophisticated models, can be expressed in terms of graphical manipulations, in which underlying mathematical expressions are carried along implicitly.

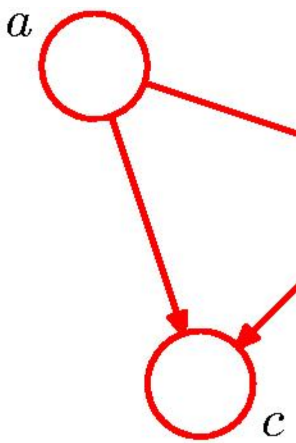
A Few Definitions

- Nodes (vertices) + links (arcs, edges)
  - Node: a random variable
  - Link: a probabilistic relationship
- Directed graphical models or Bayesian networks.
- Undirected graphical models or Markov random fields.
- Factor graphs convenient for solving inference problems



Bayesian Networks

Directed Acyclic Graph (DAG)



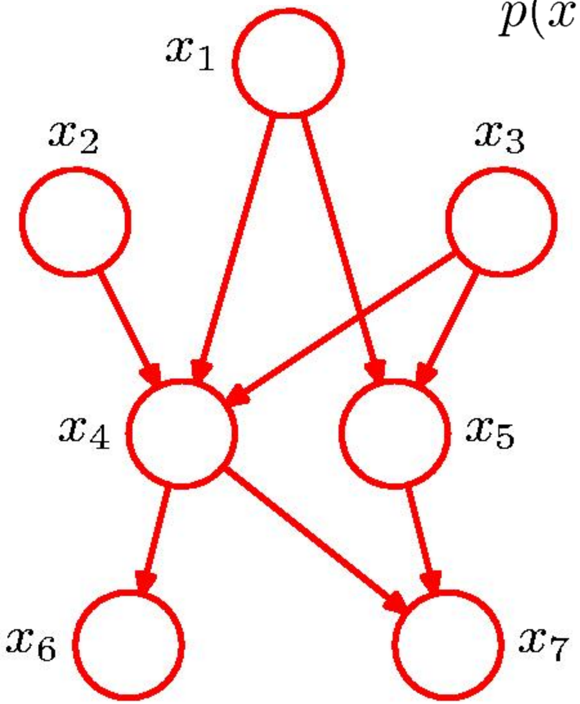
This graph is fully connected because there is a link between every pair of nodes.

$$p(a,b,c) = p(c|a,b)p(a,b) = p(c|a,b)p(b|a)p(a)$$

$$p(x_1,\ldots,x_K) = p(x_K|x_1,\ldots,x_{K-1})\ldots p(x_2|x_1)p(x_1)$$

Bayesian Networks

The absence of links conveys important information about the properties of the class of distributions that the graph represents.



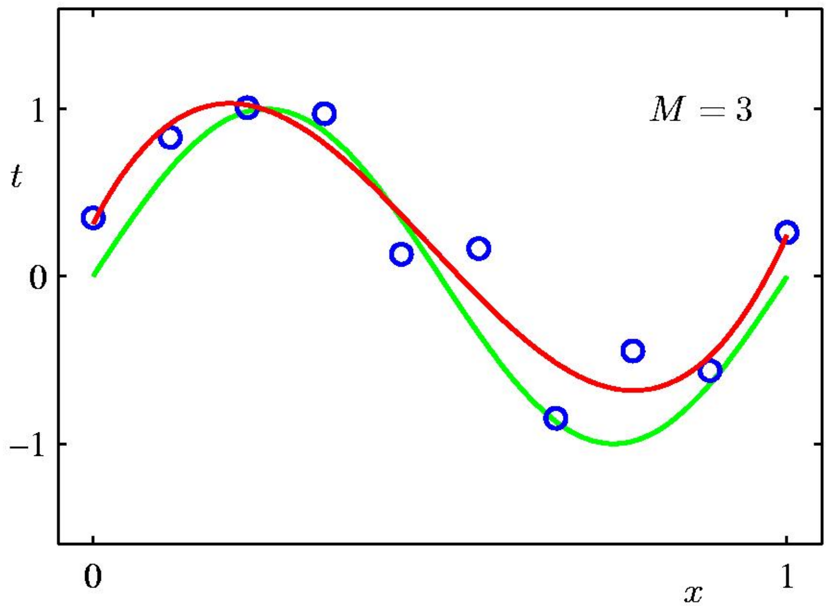
$$
\begin{aligned}
p(x_1, \ldots, x_7) \quad = \quad & p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\
& p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)
\end{aligned}
$$

General Factorization

$$
p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)
$$

Directed acyclic graphs, or DAGs

Bayesian Curve Fitting (1)



Polynomial

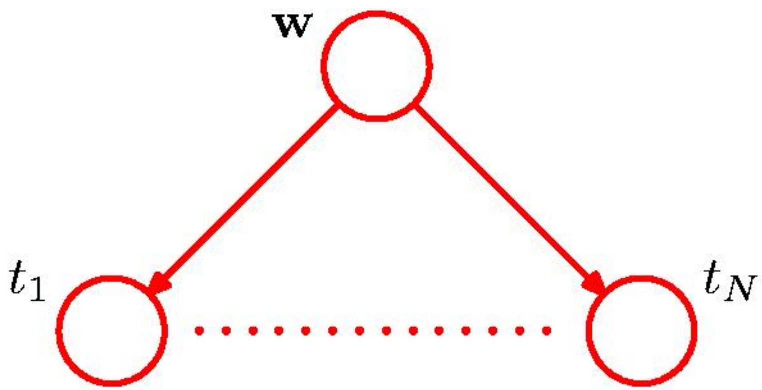
$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j x^j$$

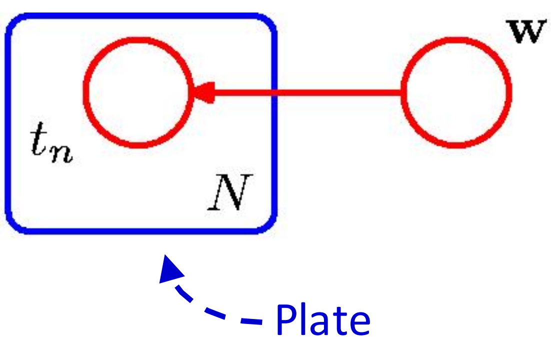
$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^{N} p(t_n | y(\mathbf{w}, x_n))$$

Random variables: polynomial coefficients **w** and the observed data **t**.

Bayesian Curve Fitting (2)

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^{N} p(t_n | y(\mathbf{w}, x_n))$$

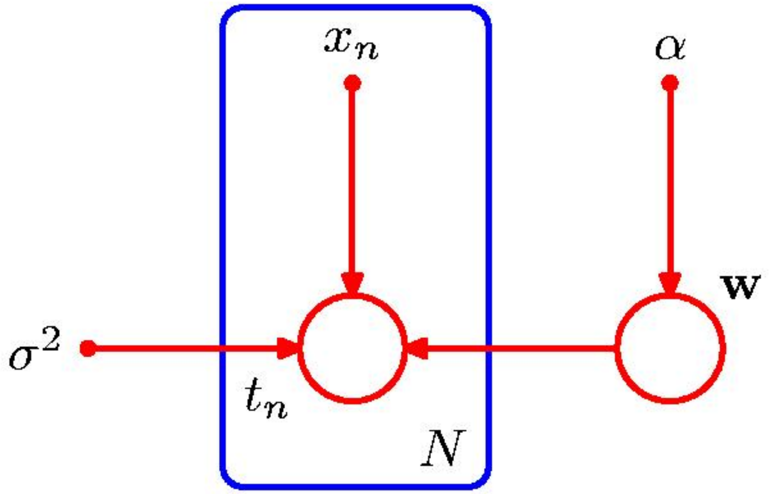




Bayesian Curve Fitting (3)

Input variables and explicit hyperparameters

$$p(\mathbf{t}, \mathbf{w}|\mathbf{x}, \alpha, \sigma^2) = p(\mathbf{w}|\alpha) \prod_{n=1}^{N} p(t_n|\mathbf{w}, x_n, \sigma^2).$$

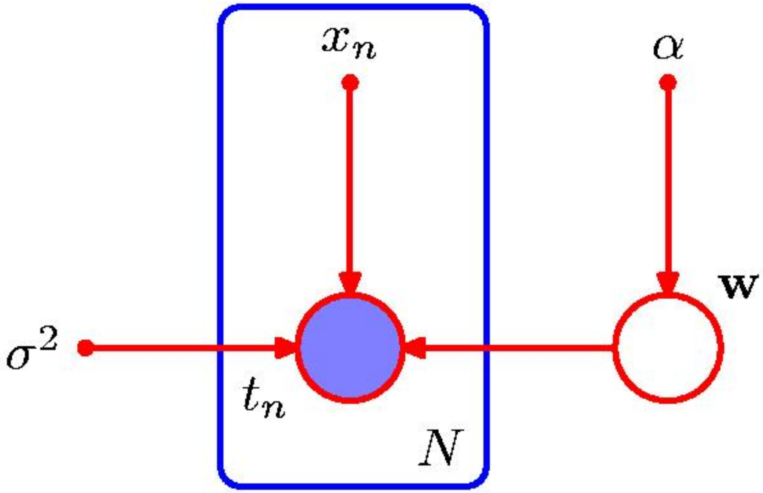


Deterministic parameters shown by small nodes

Bayesian Curve Fitting —Learning

Condition on data

$$p(\mathbf{w}|\mathbf{t}) \propto p(\mathbf{w}) \prod_{n=1}^{N} p(t_n|\mathbf{w})$$



latent or hidden

Shaded nodes are set to observed values

Discrete Variables (1)

General joint distribution: $\mathrm{K}^{2} \{ 1$ parameters

$$p(\mathbf{x}_1, \mathbf{x}_2 | \boldsymbol{\mu}) = \prod_{k=1}^{K} \prod_{l=1}^{K} \mu_{kl}^{x_{1k} x_{2l}}$$

Independent joint distribution: $2(\mathrm{K} \{ 1)$ parameters

$$\hat{p}(\mathbf{x}_1, \mathbf{x}_2 | \boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_{1k}^{x_{1k}} \prod_{l=1}^{K} \mu_{2l}^{x_{2l}}$$

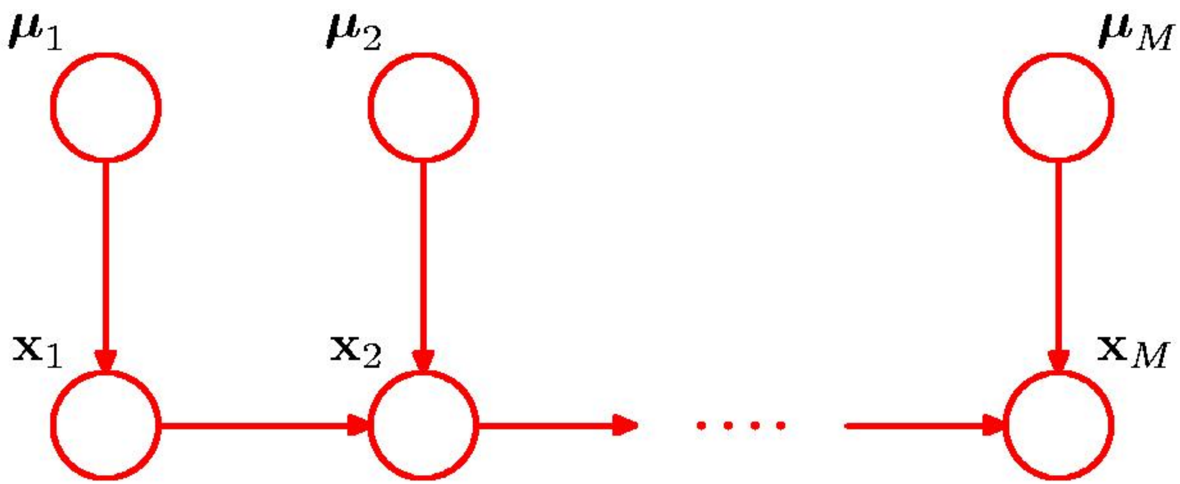
Discrete Variables (2)

General joint distribution over M variables:
  $K^M \{ 1$ parameters

M -node Markov chain: $K \{ 1 + (M \{ 1) K(K \{ 1)$
  parameters

Sparse connectivity results in fewer parameters.

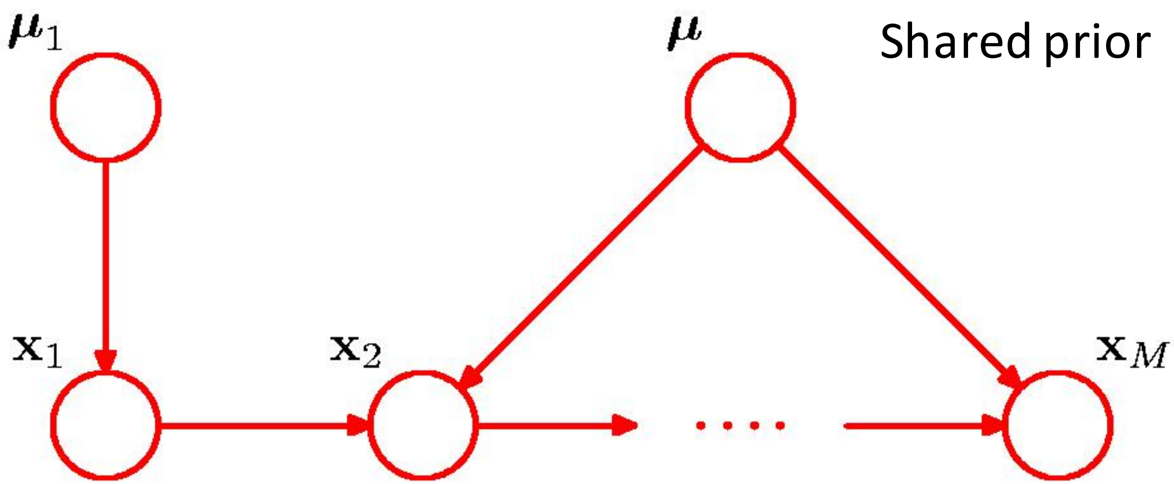
Discrete Variables: Bayesian Parameters (1)



$$p(\{\mathbf{x}_m, \boldsymbol{\mu}_m\}) = p(\mathbf{x}_1 | \boldsymbol{\mu}_1) \, p(\boldsymbol{\mu}_1) \prod_{m=2}^{M} p(\mathbf{x}_m | \mathbf{x}_{m-1}, \boldsymbol{\mu}_m) \, p(\boldsymbol{\mu}_m)$$

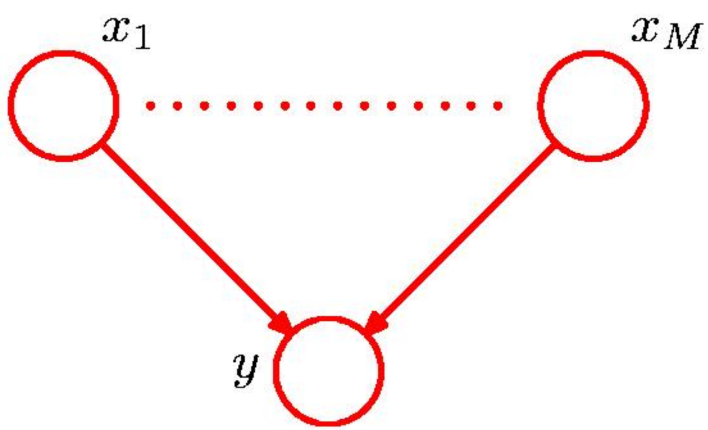
$$p(\boldsymbol{\mu}_m) = \mathrm{Dir}(\boldsymbol{\mu}_m | \boldsymbol{\alpha}_m)$$

Discrete Variables: Bayesian Parameters (2)



$$p\left(\{\mathbf{x}_m\}, \boldsymbol{\mu}_1, \boldsymbol{\mu}\right) = p\left(\mathbf{x}_1 | \boldsymbol{\mu}_1\right) p\left(\boldsymbol{\mu}_1\right) \prod_{m=2}^{M} p\left(\mathbf{x}_m | \mathbf{x}_{m-1}, \boldsymbol{\mu}\right) p\left(\boldsymbol{\mu}\right)$$

Parameterized Conditional Distributions



If $x_1, \ldots, x_M$ are discrete, K-state variables, $p(y = 1 | x_1, \ldots, x_M)$ in general has $O(K^M)$ parameters.

The parameterized form (more restricted form of conditional distribution)

$$p(y = 1 | x_1, \ldots, x_M) = \sigma\left(w_0 + \sum_{i=1}^{M} w_i x_i\right) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})$$

requires only $M + 1$ parameters

Linear-Gaussian Models

Directed Graph

$$p(x_i|\mathrm{pa}_i) = \mathcal{N}\left(x_i \,\middle|\, \sum_{j \in \mathrm{pa}_i} w_{ij} x_j + b_i, v_i\right)$$

Each node is Gaussian, the mean
is a linear function of the parents.

Vector-valued Gaussian Nodes

$$p(\mathbf{x}_i|\mathrm{pa}_i) = \mathcal{N}\left(\mathbf{x}_i \,\middle|\, \sum_{j \in \mathrm{pa}_i} \mathbf{W}_{ij} \mathbf{x}_j + \mathbf{b}_i, \mathbf{\Sigma}_i\right)$$

Conditional Independence

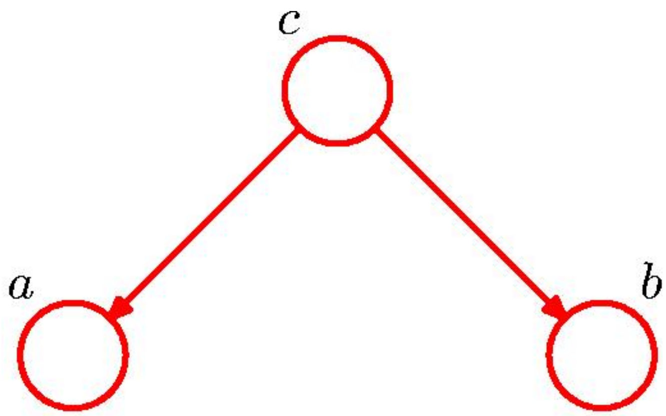
a is independent of b given c

$$p(a|b,c) = p(a|c)$$

Equivalently
$$\begin{aligned} p(a,b|c) &= p(a|b,c)p(b|c) \\ &= p(a|c)p(b|c) \end{aligned}$$

Notation
$$a \perp\!\!\!\perp b \mid c$$

Conditional Independence: Example 1

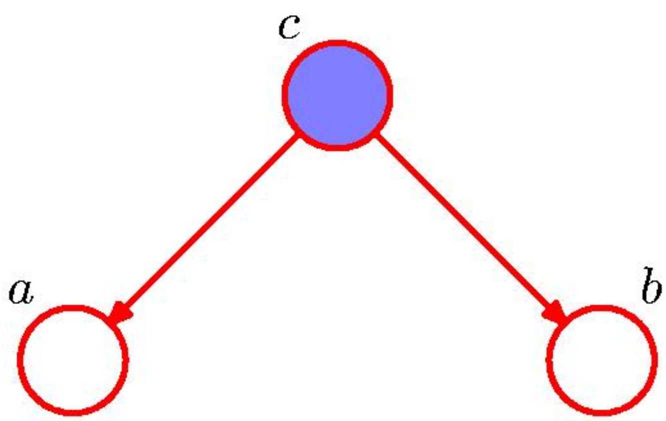


$$p(a,b,c) = p(a|c)p(b|c)p(c)$$

$$p(a,b) = \sum_c p(a|c)p(b|c)p(c)$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

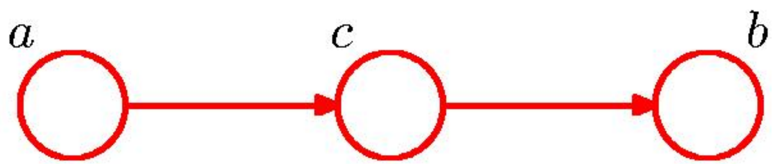
Conditional Independence: Example 1



$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = p(a|c)p(b|c)$$

$$a \perp\!\!\!\perp b \mid c$$

Conditional Independence: Example 2

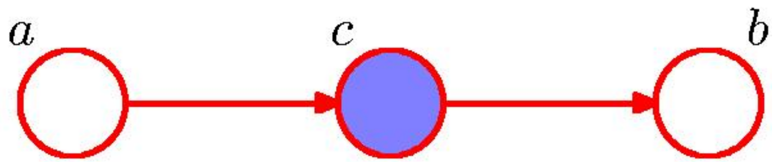


$$p(a, b, c) = p(a)p(c|a)p(b|c)$$

$$p(a, b) = p(a) \sum_c p(c|a)p(b|c) = p(a)p(b|a)$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

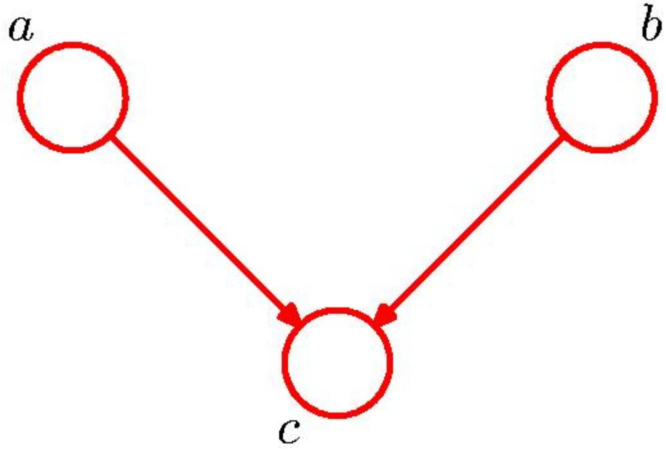
Conditional Independence: Example 2



$$
\begin{aligned}
p(a, b|c) &= \frac{p(a, b, c)}{p(c)} \\
&= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\
&= p(a|c)p(b|c)
\end{aligned}
$$

$$a \perp\!\!\!\perp b \mid c$$

Conditional Independence: Example 3



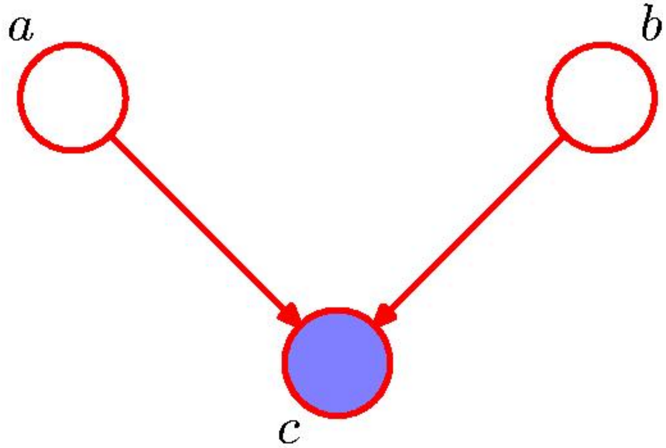
$$p(a, b, c) = p(a)p(b)p(c|a, b)$$

$$p(a, b) = p(a)p(b)$$

$$a \perp\!\!\!\perp b \mid \emptyset$$

Note: this is the opposite of Example 1, with c unobserved.

Conditional Independence: Example 3



$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(b)p(c|a, b)}{p(c)}$$

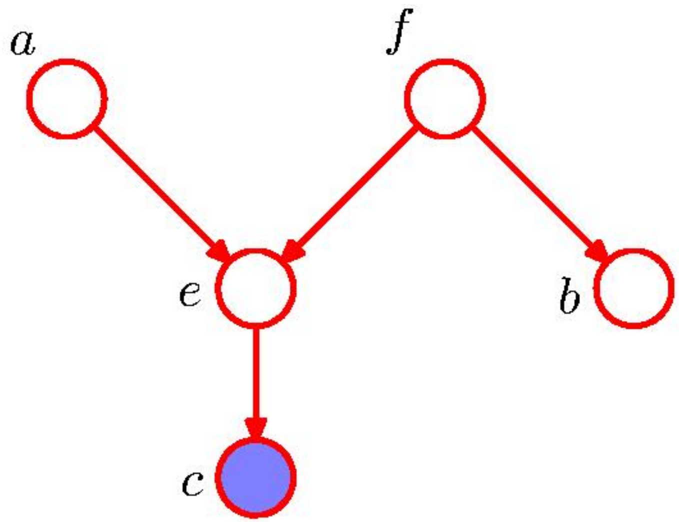
$$a \not\perp\!\!\!\perp b \mid c$$

Note: this is the opposite of Example 1, with c observed.

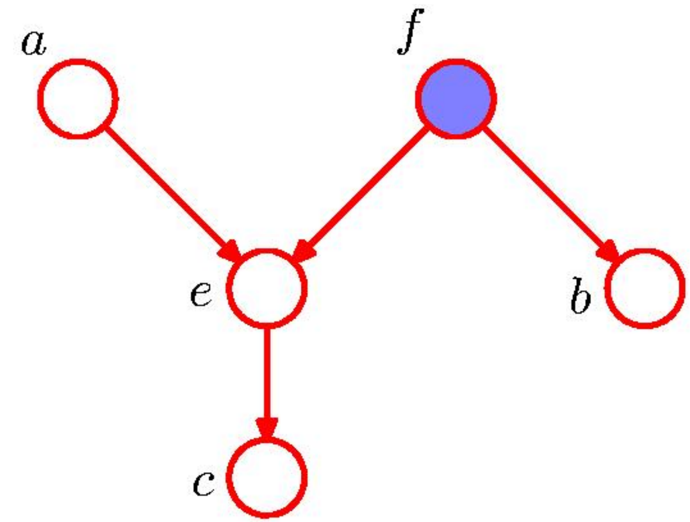
D-separation

- A, B, and C are non-intersecting subsets of nodes in a directed graph.
- A path from A to B is blocked if it contains a node such that either
  a) the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set C, or
  b) the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, are in the set C.
- If all paths from A to B are blocked, A is said to be d-separated from B by C.
- If A is d-separated from B by C, the joint distribution over all variables in the graph satisfies $A \perp\!\!\!\perp B \mid C$.

D-separation: Example

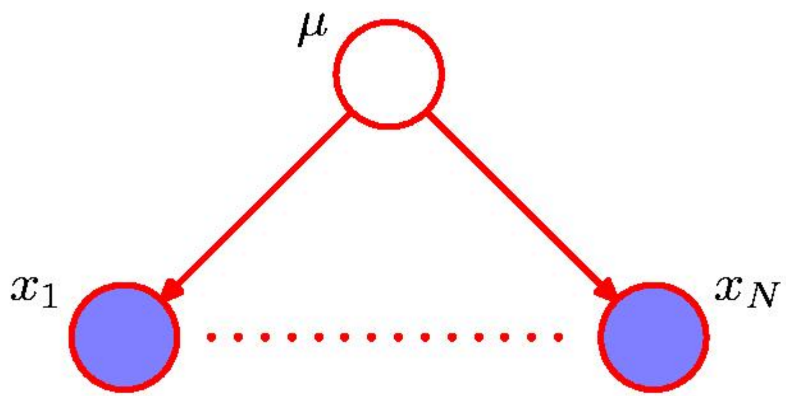


$a \not\perp\!\!\!\perp b \mid c$



$a \perp\!\!\!\perp b \mid f$

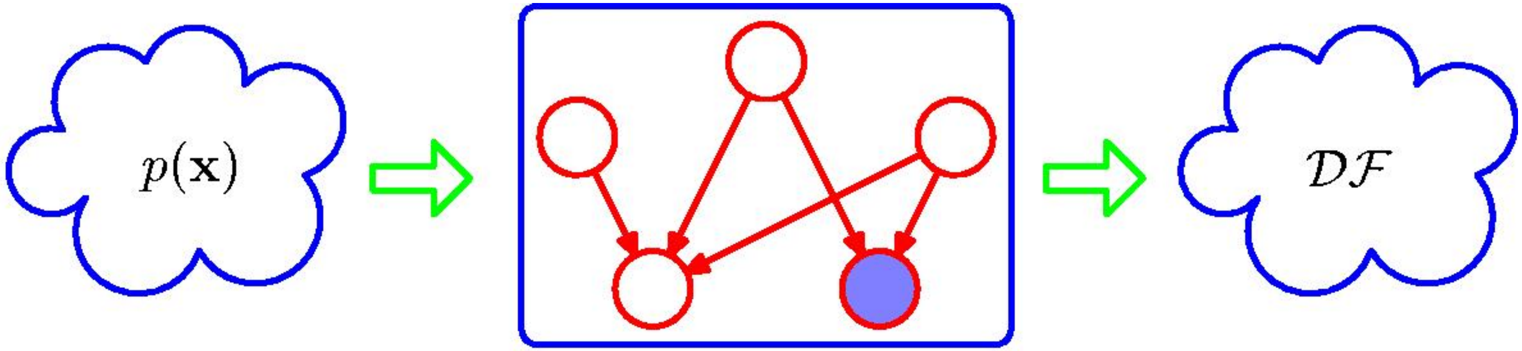
D-separation: I.I.D. Data



$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu)$$

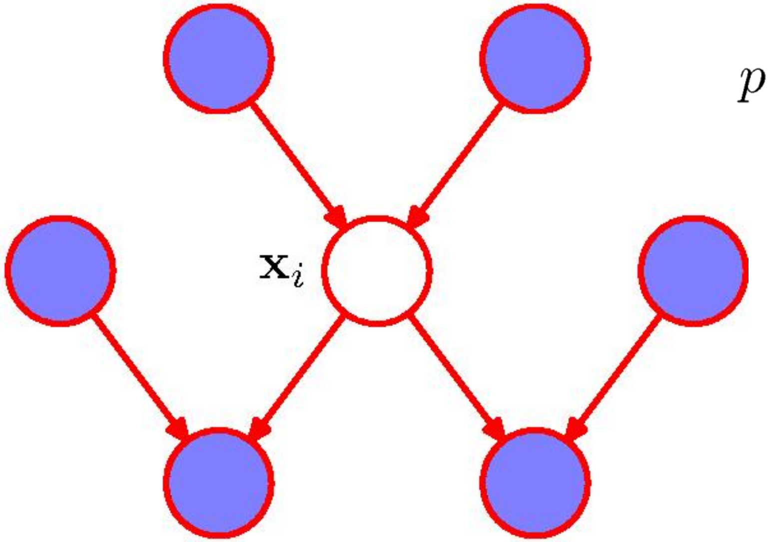
$$p(\mathcal{D}) = \int_{-\infty}^{\infty} p(\mathcal{D}|\mu)p(\mu)\,\mathrm{d}\mu \neq \prod_{n=1}^{N} p(x_n)$$

Directed Graphs as Distribution Filters



- We can view a graphical model as a filter in which a probability distribution p(x) is allowed through the filter if, and only if, it satisfies the directed factorization property .
    - The set of all possible probability distributions p(x) that pass through the filter is denoted DF.
- We can alternatively use the graph to filter distributions according to whether they respect all of the conditional independencies implied by the d-separation properties of the graph.

The Markov Blanket



$$p(\mathbf{x}_i|\mathbf{x}_{\{j\neq i\}}) = \frac{p(\mathbf{x}_1,\ldots,\mathbf{x}_M)}{\int p(\mathbf{x}_1,\ldots,\mathbf{x}_M)\,\mathrm{d}\mathbf{x}_i} = \frac{\prod_k p(\mathbf{x}_k|\mathrm{pa}_k)}{\int \prod_k p(\mathbf{x}_k|\mathrm{pa}_k)\,\mathrm{d}\mathbf{x}_i}$$

Factors independent of x_i cancel between numerator and denominator.

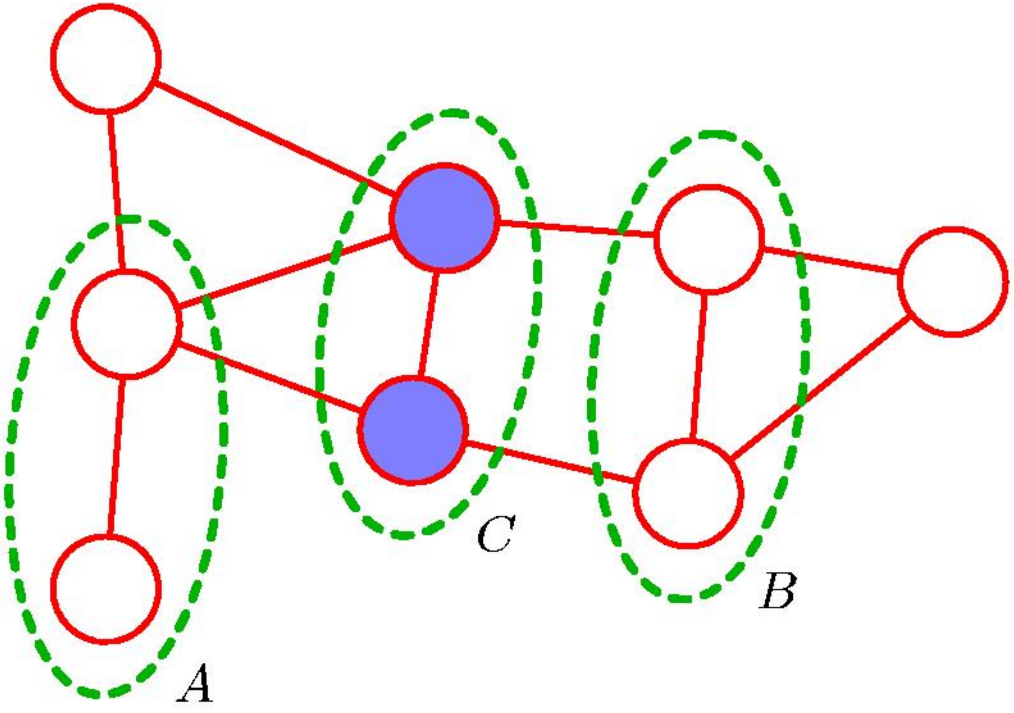
Conditional Independence

- Conditional independence properties simplify both the structure of a model and the computations needed to perform inference and learning under that model.

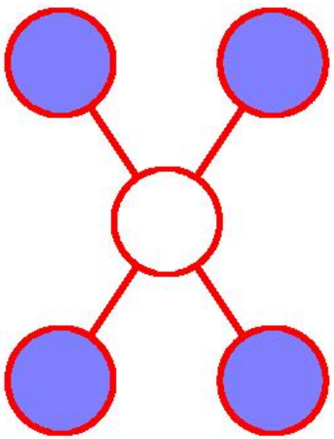
- Given an expression for the joint distribution, we can check conditional independence by repeated application of the sum and product rules of probability.

- An important and elegant feature of graphical models is that conditional independence properties of the joint distribution can be read directly from the graph.

Markov Random Fields



Markov Blanket



$A \perp\!\!\!\perp B \mid C$

Markov random field, also known as Markov network or undirected graphical model

Factorization (1)

- In directed graphical models, the joint distribution can be factored as the product of conditional distributions.
- Seek a factorization rule for undirected graphs that will correspond to the above conditional independence test.
  - Express the joint distribution p(x) as a product of functions defined over sets of variables that are local to the graph.

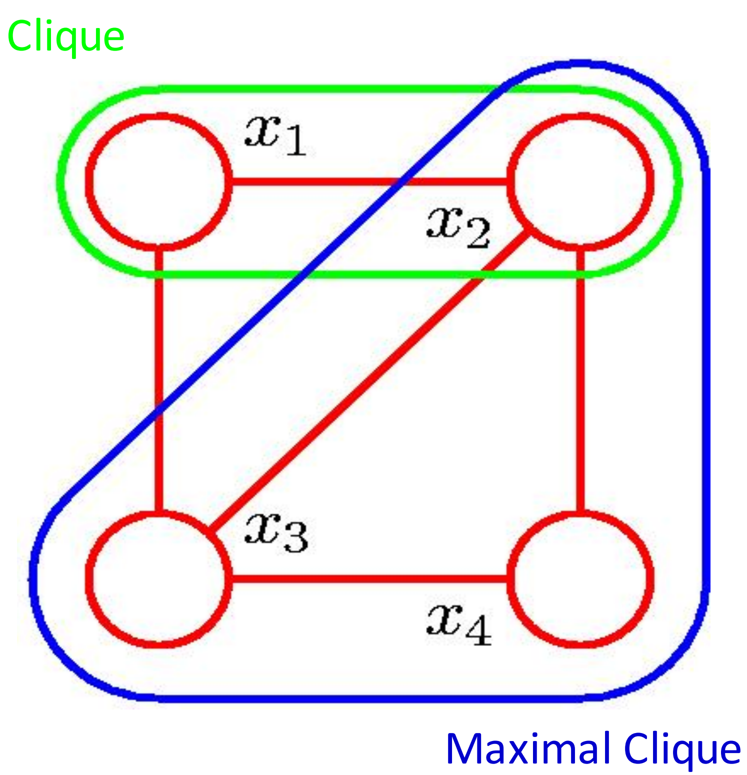
Factorization (2)

- Consider two nodes x_i and x_j that are not connected by a link:
  - They are conditionally independent given all other nodes in the graph.

$$p(x_i, x_j | \mathbf{x}_{\setminus \{i,j\}}) = p(x_i | \mathbf{x}_{\setminus \{i,j\}}) p(x_j | \mathbf{x}_{\setminus \{i,j\}})$$

- In the factorization of the joint distribution, x_i and x_j do not appear in the same factor.

Cliques and Maximal Cliques



- Clique is defined as a subset of the nodes in a graph such that there exists a link between all pairs of nodes in the subset.

- A maximal clique is a clique such that it is not possible to include any other nodes from the graph in the set without it ceasing to be a clique.

- Define the factors in the decomposition of the joint distribution to be functions of the variables in the cliques.

Joint Distribution (1)

- Express the joint distribution as $p(\mathbf{x}) = \frac{1}{Z}\prod_C \psi_C(\mathbf{x}_C)$

  where $\psi_C(\mathbf{x}_C)$ is the potential over clique C and the partition function Z:

$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

  is the normalization coefficient.

- We do not restrict the choice of potential functions to those that have a specific probabilistic interpretation as marginal or conditional distributions.

- One consequence of the generality of the potential functions is that their product will in general not be correctly normalized.
  - M K-state variables → K^M terms in Z.

Joint Distribution (2)

- We are restricted to potential functions which are strictly positive

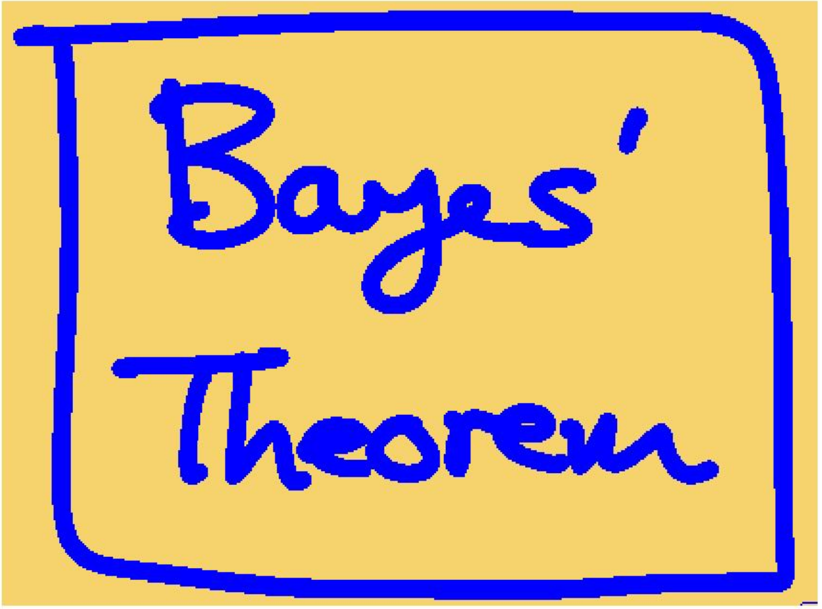
- Energies and the Boltzmann distribution

Energy function

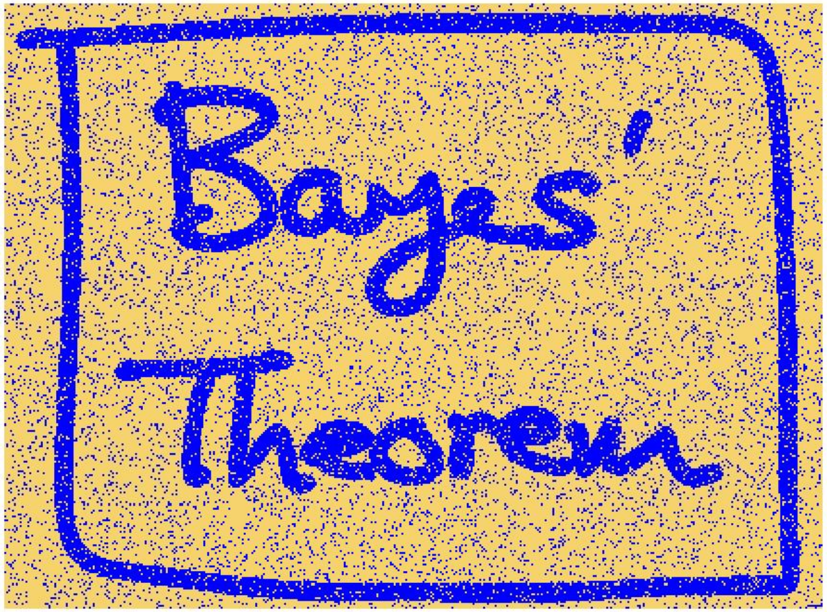
$$\psi_C(\mathbf{x}_C) = \exp\{-E(\mathbf{x}_C)\}$$

- The potentials in an undirected graph do not have a specific probabilistic interpretation. (greater flexibility)
- How to motivate a choice of potential function for a particular application?
  - Find a good balance in satisfying the (possibly conflicting) influences of the clique potentials.

Illustration: Image De-Noising (1)



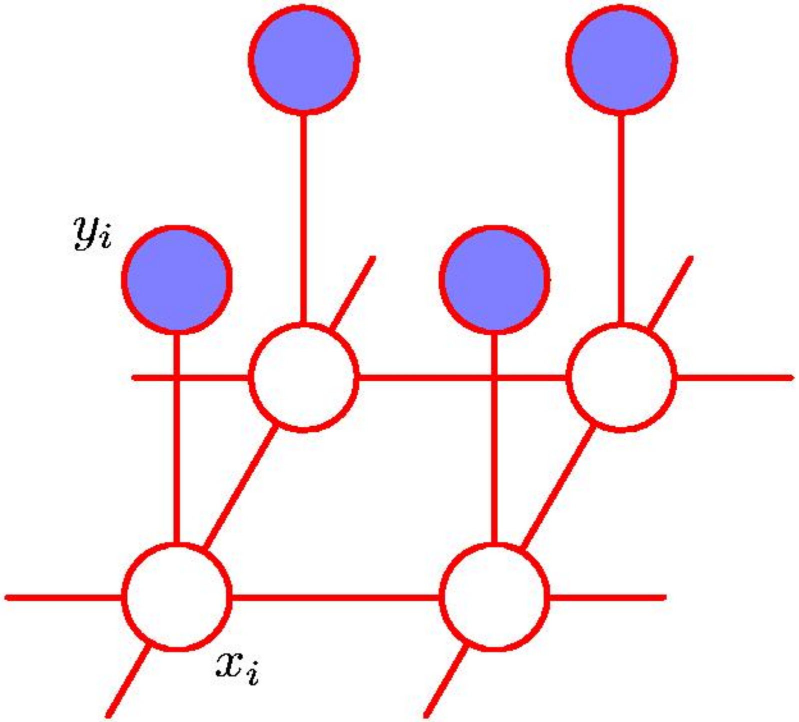
Original Image



Noisy Image

flipping the sign of the pixels with probability 10%

Illustration: Image De-Noising (2)

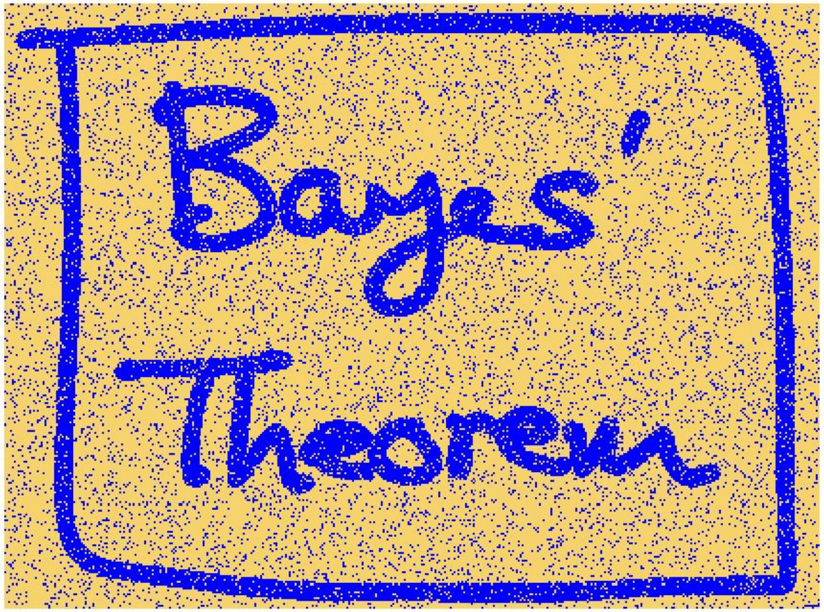


$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$

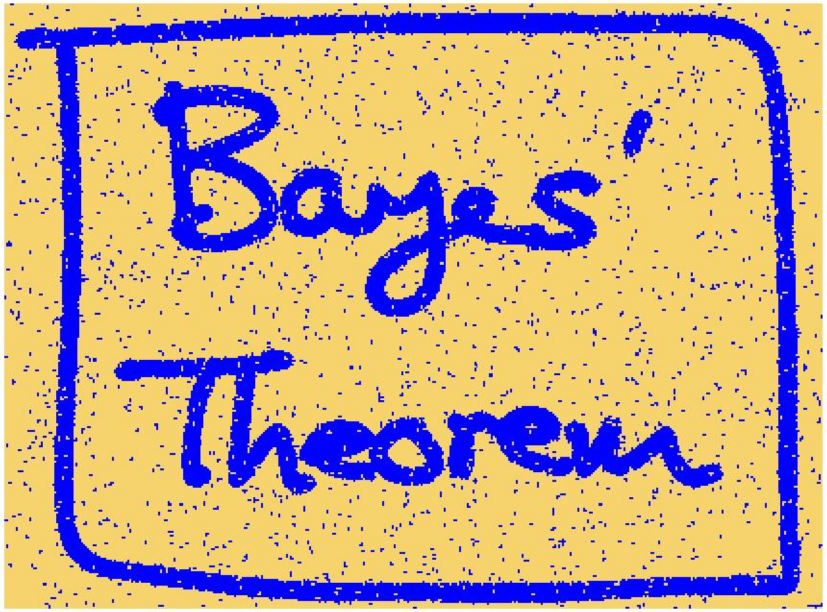
$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{x}, \mathbf{y})\}$$

Iterated conditional modes, or ICM: coordinate-wise gradient ascent

Illustration: Image De-Noising (3)



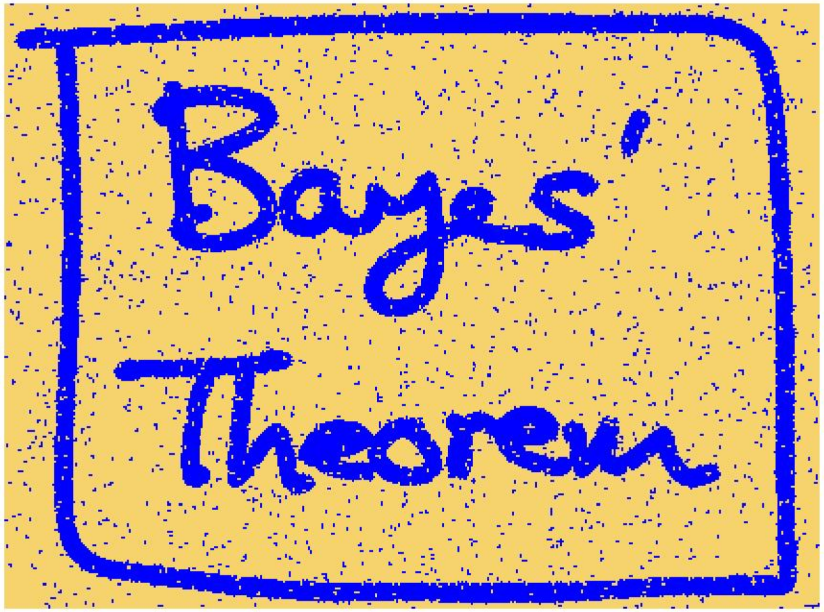
Noisy Image



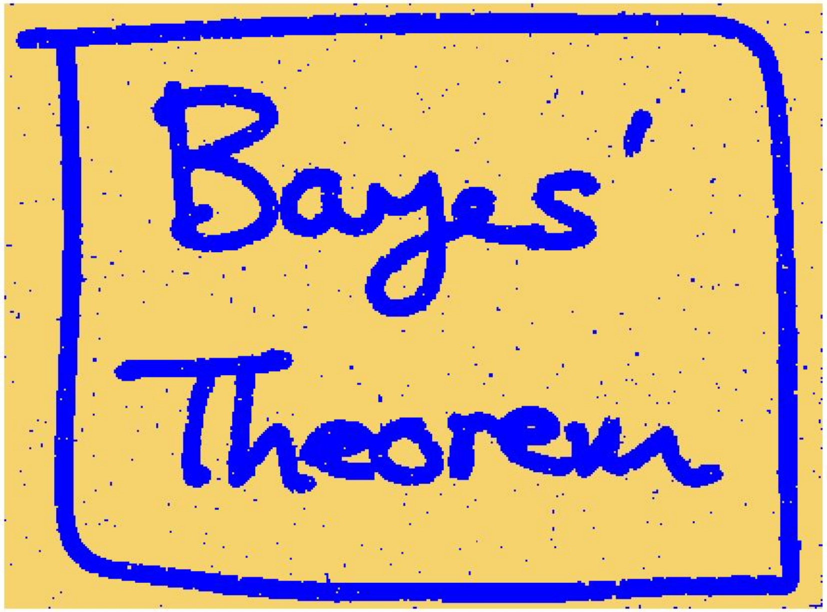
Restored Image (ICM)

$\beta = 1.0, \eta = 2.1, h = 0.$

Illustration: Image De-Noising (4)

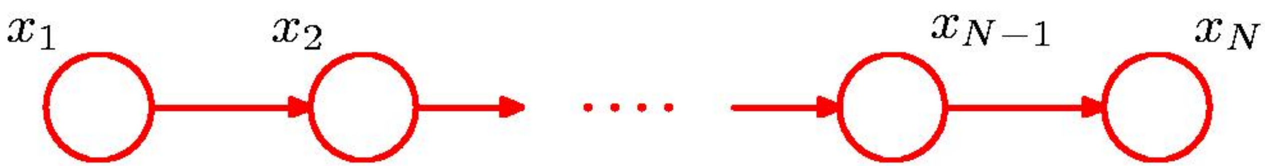


Restored Image (ICM)



Restored Image (Graph cuts)

Converting Directed to Undirected Graphs (1)

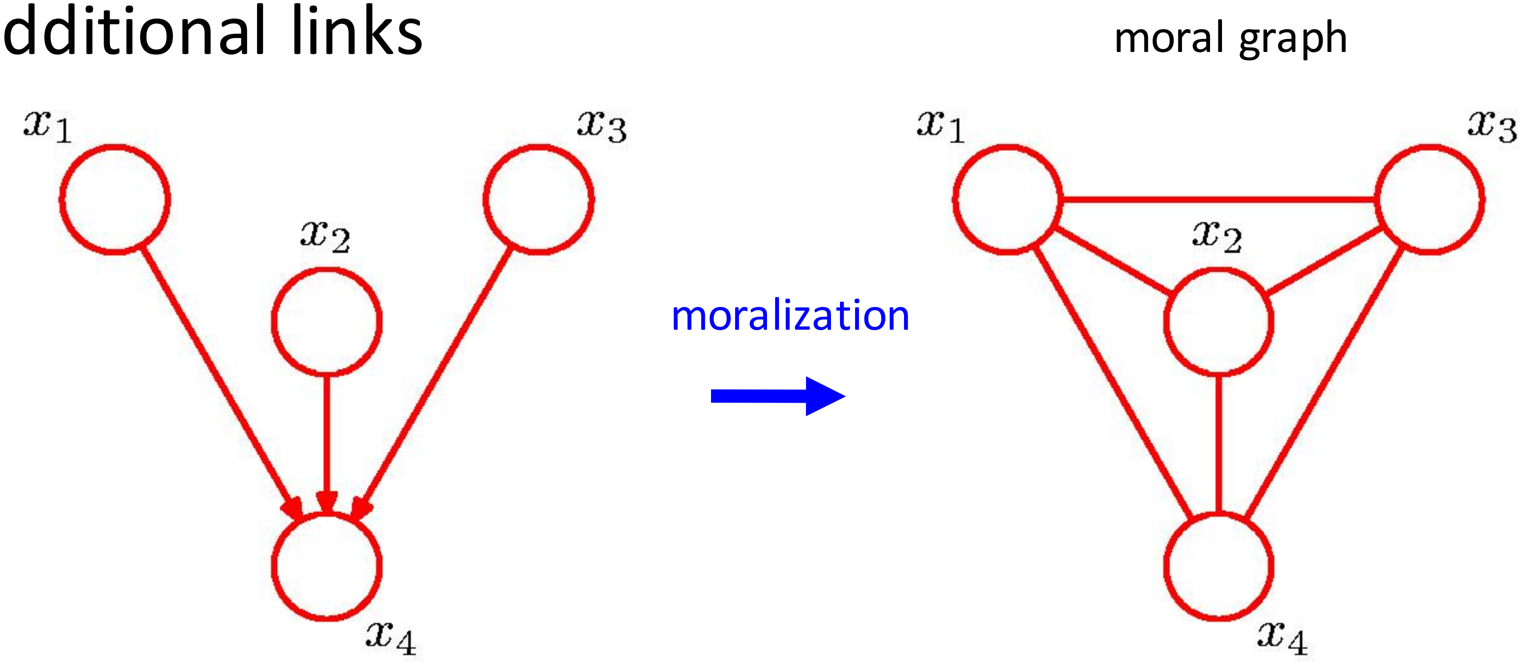


$$p(\mathbf{x}) = \underbrace{p(x_1)p(x_2|x_1)}\; p(x_3|x_2) \cdots p(x_N|x_{N-1})$$

$$p(\mathbf{x}) = \frac{1}{Z}\, \psi_{1,2}(x_1, x_2)\; \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N)$$

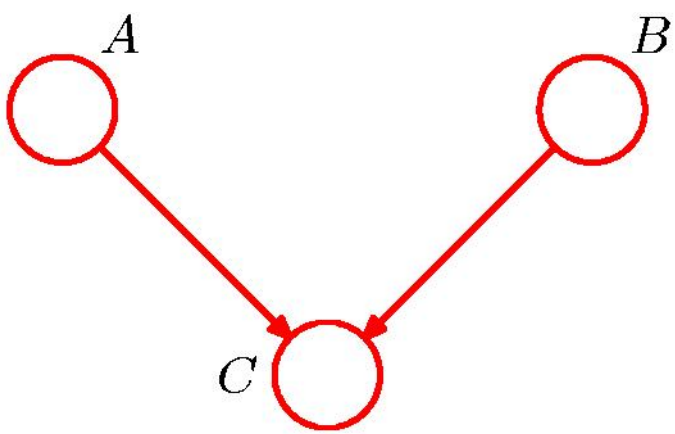
Converting Directed to Undirected Graphs (2)

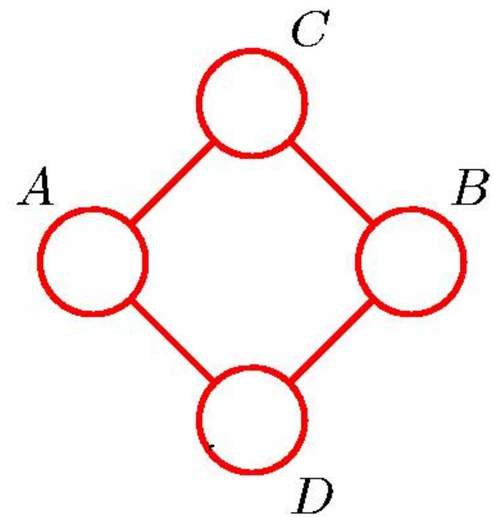
Additional links



$$
\begin{aligned}
p(\mathbf{x}) &= p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\
&= \frac{1}{Z}\psi_A(x_1, x_2, x_3)\psi_B(x_2, x_3, x_4)\psi_C(x_1, x_2, x_4)
\end{aligned}
$$

Directed vs. Undirected Graphs





$$A \perp\!\!\!\perp B \mid \emptyset$$

$$A \not\!\perp\!\!\!\perp B \mid C$$

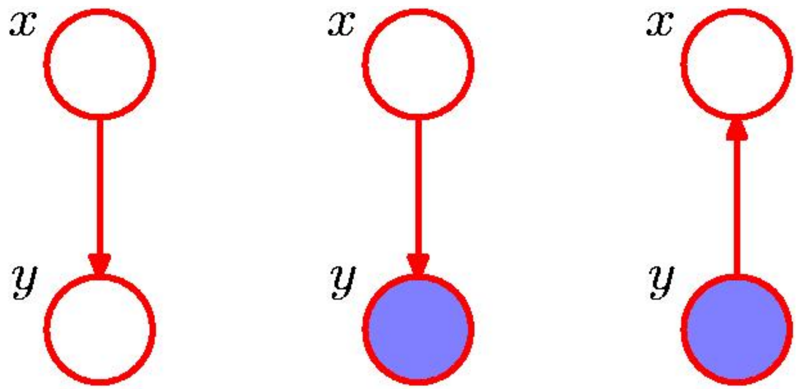
$$A \not\!\perp\!\!\!\perp B \mid \emptyset$$

$$A \perp\!\!\!\perp B \mid C \cup D$$

$$C \perp\!\!\!\perp D \mid A \cup B$$

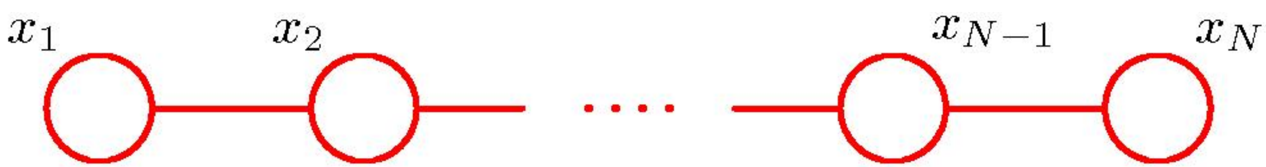
Inference in Graphical Models

- Inference in graphical models
  - Some of the nodes in a graph are clamped to observed values, and we wish to compute the posterior distributions of one or more subsets of other nodes.



$$p(y) = \sum_{x'} p(y|x')p(x') \qquad\qquad p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Inference on a Chain



$$p(\mathbf{x}) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\cdots\psi_{N-1,N}(x_{N-1}, x_N)$$

$$p(x_n) = \sum_{x_1}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_N} p(\mathbf{x})$$

- In a naive implementation, we would first evaluate the joint distribution and then perform the summations explicitly.
  - N variables each with K states => there are K^N values for **x**

Inference on a Chain

- Obtain a much more efficient algorithm by exploiting the conditional independence properties of the graphical model.

- Key Idea: Rearrange the order of the summations and the multiplications to allow the required marginal to be evaluated much more efficiently.

Inference on a Chain

$$p(\mathbf{x}) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\cdots\psi_{N-1,N}(x_{N-1}, x_N)$$

$$p(x_n) = \sum_{x_1}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_N} p(\mathbf{x})$$

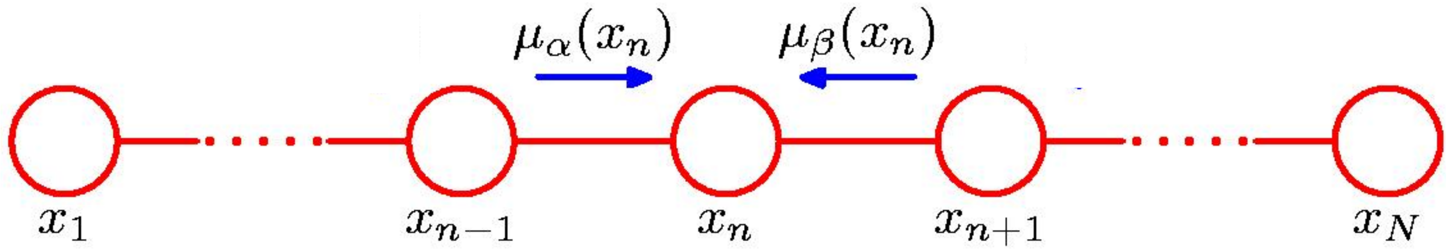
Consider for instance the summation over *x_N:*

$$\mu_\beta(x_{N-1}) \Leftarrow \sum_{x_N}\psi_{N-1,N}(x_{N-1}, x_N)$$ Only one that depends on x_N

$$\mu_\beta(x_{N-2}) \Leftarrow \sum_{x_{N-1}}\psi_{N-2,N-1}(x_{N-2}, x_{N-1})\mu_\beta(x_{N-1})$$

Each summation effectively removes a variable from the distribution. This can be viewed as the removal of a node from the graph.

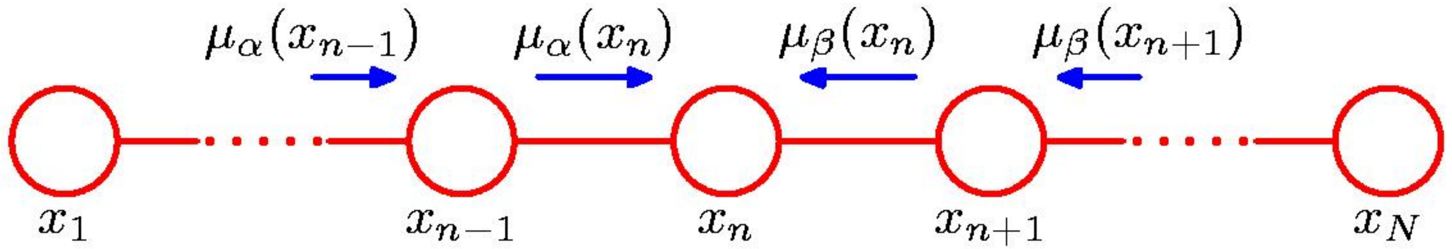
Inference on a Chain



$$p(x_n) = \frac{1}{Z} \underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_1} \psi_{1,2}(x_1, x_2)\right] \cdots \right]}_{\mu_\alpha(x_n)}$$

$$\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)\right] \cdots \right]}_{\mu_\beta(x_n)}$$

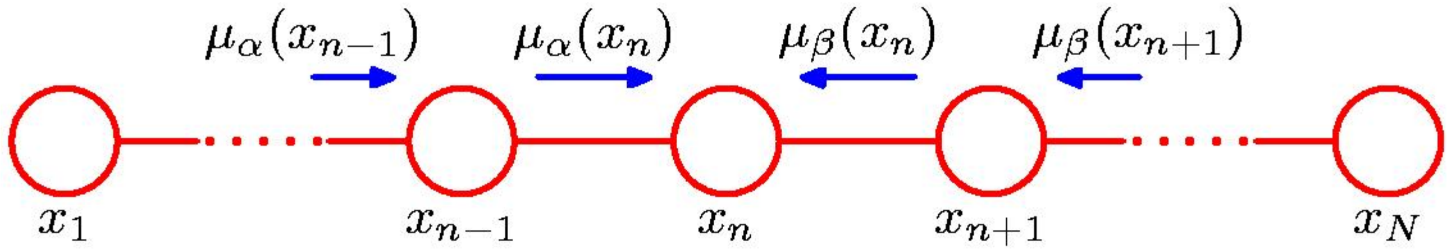
Inference on a Chain



$$\begin{aligned}
\mu_\alpha(x_n) &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \left[\sum_{x_{n-2}} \cdots \right] \\
&= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1}).
\end{aligned}$$

$$\begin{aligned}
\mu_\beta(x_n) &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \left[\sum_{x_{n+2}} \cdots \right] \\
&= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \mu_\beta(x_{n+1}).
\end{aligned}$$

Inference on a Chain



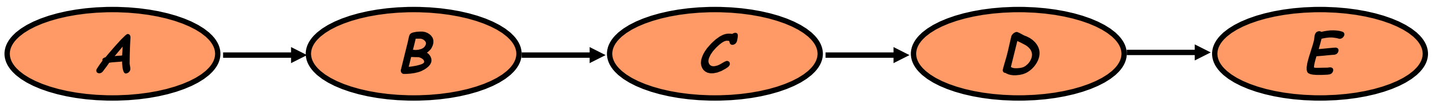
$$\mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \qquad \mu_\beta(x_{N-1}) = \sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)$$

$$Z = \sum_{x_n} \mu_\alpha(x_n)\mu_\beta(x_n)$$

Total time complexity: $O(NK^2)$. This is linear in the length of the chain, in contrast to the exponential cost of a naive approach.

Elimination in Chains

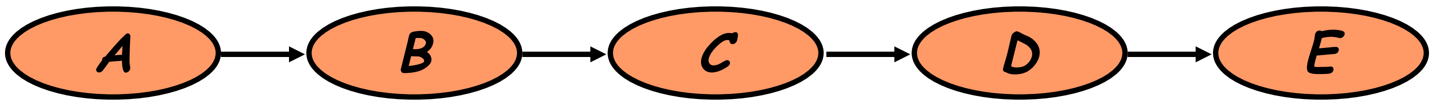
We now try to understand the simple chain example using first-order principles



Using definition of probability, we have

$$P(e) = \sum_d \sum_c \sum_b \sum_a P(a,b,c,d,e)$$

Elimination in Chains

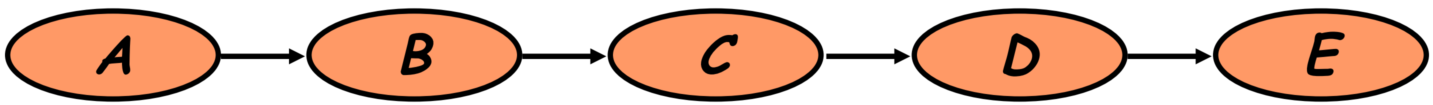


By chain decomposition, we get

$$P(e) = \sum_d \sum_c \sum_b \sum_a P(a,b,c,d,e)$$

$$= \sum_d \sum_c \sum_b \sum_a P(a)P(b \mid a)P(c \mid b)P(d \mid c)P(e \mid d)$$

Elimination in Chains

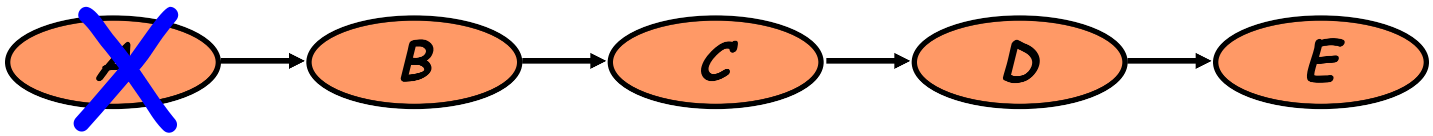


Rearranging terms ...

$$P(e) = \sum_d \sum_c \sum_b \sum_a P(a)P(b \mid a)\overbrace{P(c \mid b)P(d \mid c)P(e \mid d)}$$

$$= \sum_d \sum_c \sum_b P(c \mid b)P(d \mid c)P(e \mid d)\sum_a P(a)P(b \mid a)$$

Elimination in Chains



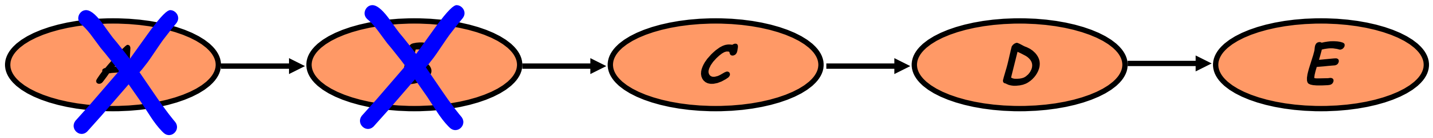
Now we can perform innermost summation

$$P(e) = \sum_d \sum_c \sum_b P(c \mid b) P(d \mid c) P(e \mid d) \sum_a P(a) P(b \mid a)$$

$$= \sum_d \sum_c \sum_b P(c \mid b) P(d \mid c) P(e \mid d) p(b)$$

This summation, is exactly the first step in the forward iteration we describe before

Elimination in Chains



Rearranging and then summing again, we get

$$P(e) = \sum_d \sum_c \sum_b P(c \mid b) P(d \mid c) P(e \mid d) p(b)$$

$$= \sum_d \sum_c P(d \mid c) P(e \mid d) \sum_b P(c \mid b) p(b)$$

$$= \sum_d \sum_c P(d \mid c) P(e \mid d) p(c)$$

Inference on a Chain

To compute local marginals:

- Compute and store all forward messages, $\mu_\alpha(x_n)$.
- Compute and store all backward messages, $\mu_\beta(x_n)$.
- Compute Z at any node x_m
- Compute

$$p(x_n) = \frac{1}{Z}\mu_\alpha(x_n)\mu_\beta(x_n)$$

for all variables required.

Inference on a Chain: Summary

- Exact inference on a graph comprising a chain of nodes can be performed efficiently
  - Message passing along the chain

- This can’t be extended to an arbitrary graph.

- Inference can be performed efficiently using local message passing on a broader class of graphs called trees.
  - Sum-product algorithm