The Very Basics of Graphical Models (Lecture 1)

Tibério Caetano

sml.nicta.com.au
Statistical Machine Learning Program
NICTA
Canberra, ACT 0200 Australia
Tiberio.Caetano@gmail.com

MLSS 2008, Kioloa, Australia

Material on Graphical Models

Many good books

- Chris Bishop's book‘“Pattern Recognition and Machine Learning” (Graphical Models chapter available from his webpage in pdf format, as well as all the figures – many used here in these slides!)
- Judea Pearl's “Probabilistic Reasoning in Intelligent Systems”
- Stephen Lauritzen's “Graphical Models”
- . . .

Unpublished material

- Michael Jordan's unpublished book “An Introduction to Probabilistic Graphical Models”
- Koller and Friedman's unpublished book “Structured Probabilistic Models”

Videos

- Sam Roweis' videos on videolectures.net (Excellent!)

Introduction

Graphical Models have been applied to

- Image Processing
- Speech Processing
- Natural Language Processing
- Document Processing
- Pattern Recognition
- Bioinformatics
- Computer Vision
- Economics
- Physics
- Social Sciences
- ...

Some Examples of Applications

- Physics: Given a set or particles that can assume different states, and given bonds of interactions between these particles (as well as the individual response to an external force field), what is the global state with minimal energy?

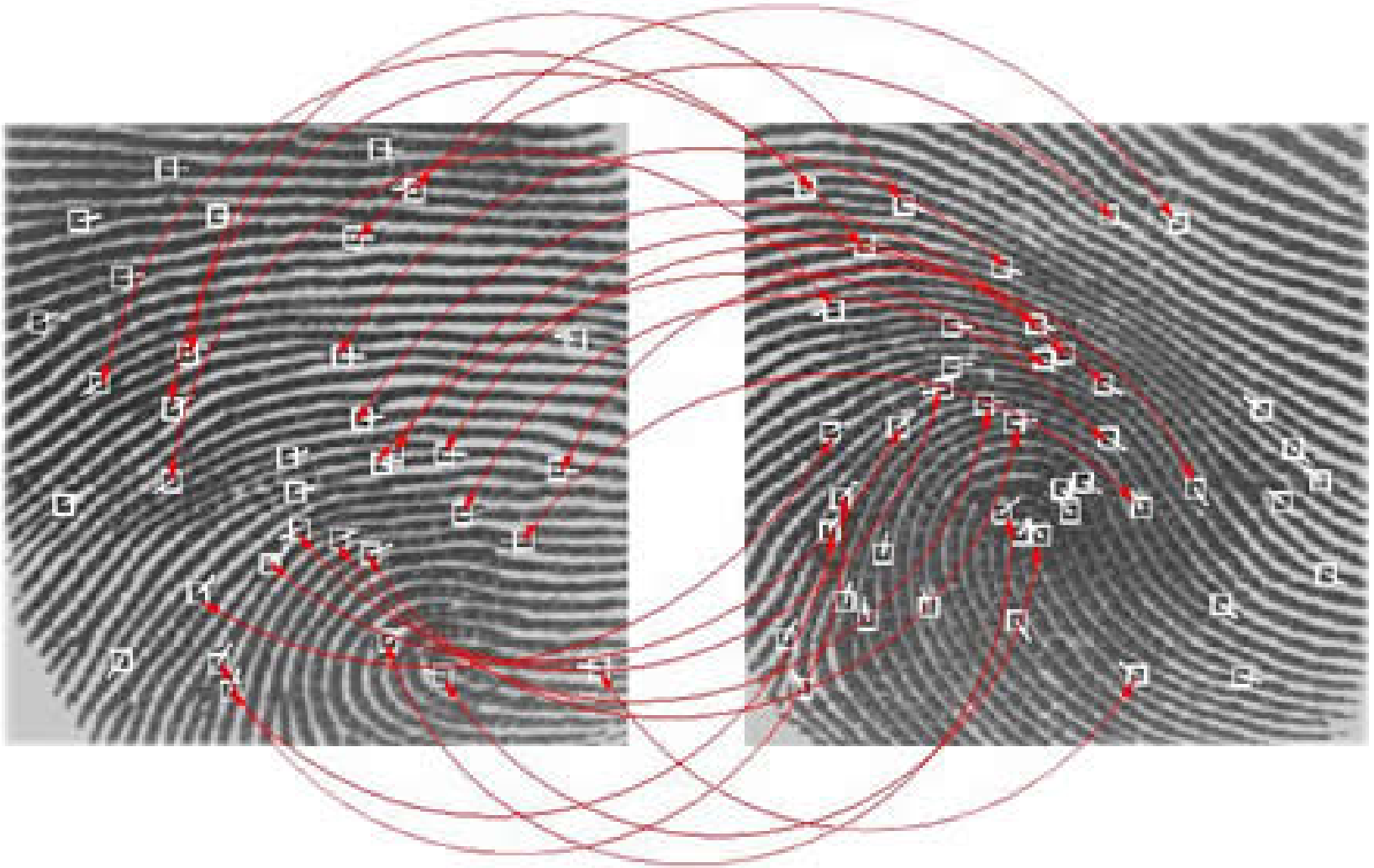
- Ecology: Given a set of organisms that can assume certain types of behaviour, and given interactions between these organisms (as well as their innate preference to have a certain behaviour), what is the set of behaviours that reaches equilibrium?

Some Examples of Applications

- Image processing: Given a set of pixels that can assume a set of labels (segmentation), and given continuity constraints that tell neighbouring pixels to have similar labels (as well as the individual preference depending on its location in the image), what is the best segmentation of the overall image ?

- Computer Vision: Given features of an object in a template image, which can match to any of a set of object features in a scene, and given that "neighbours should match to neighbours", what's the best overall match?

Feature Matching



Feature Matching

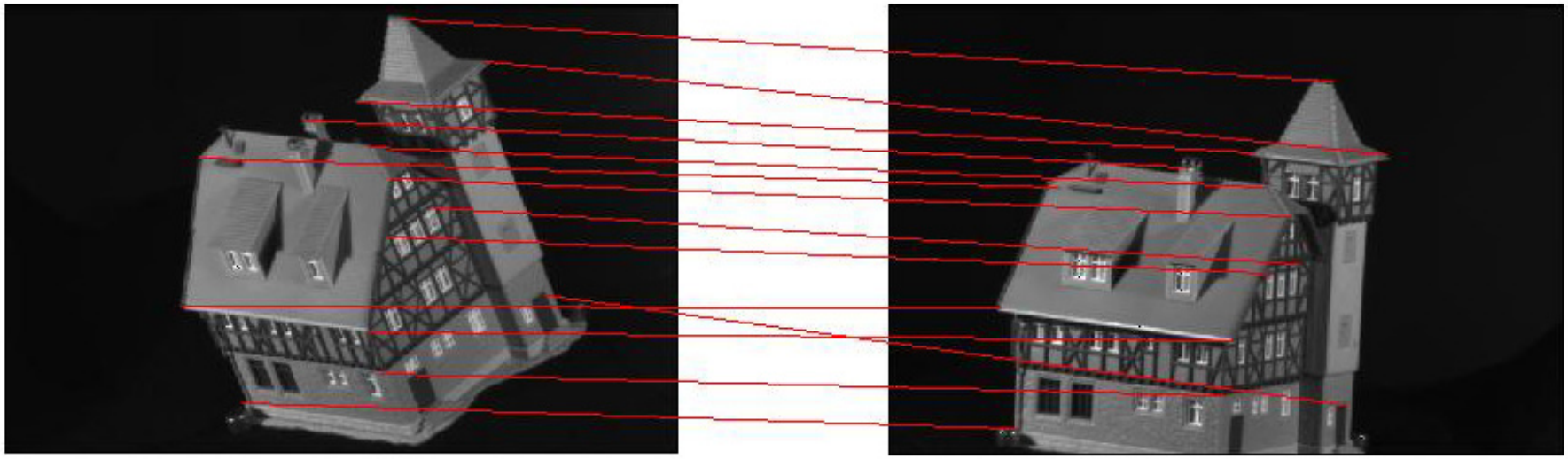


Image Inpainting

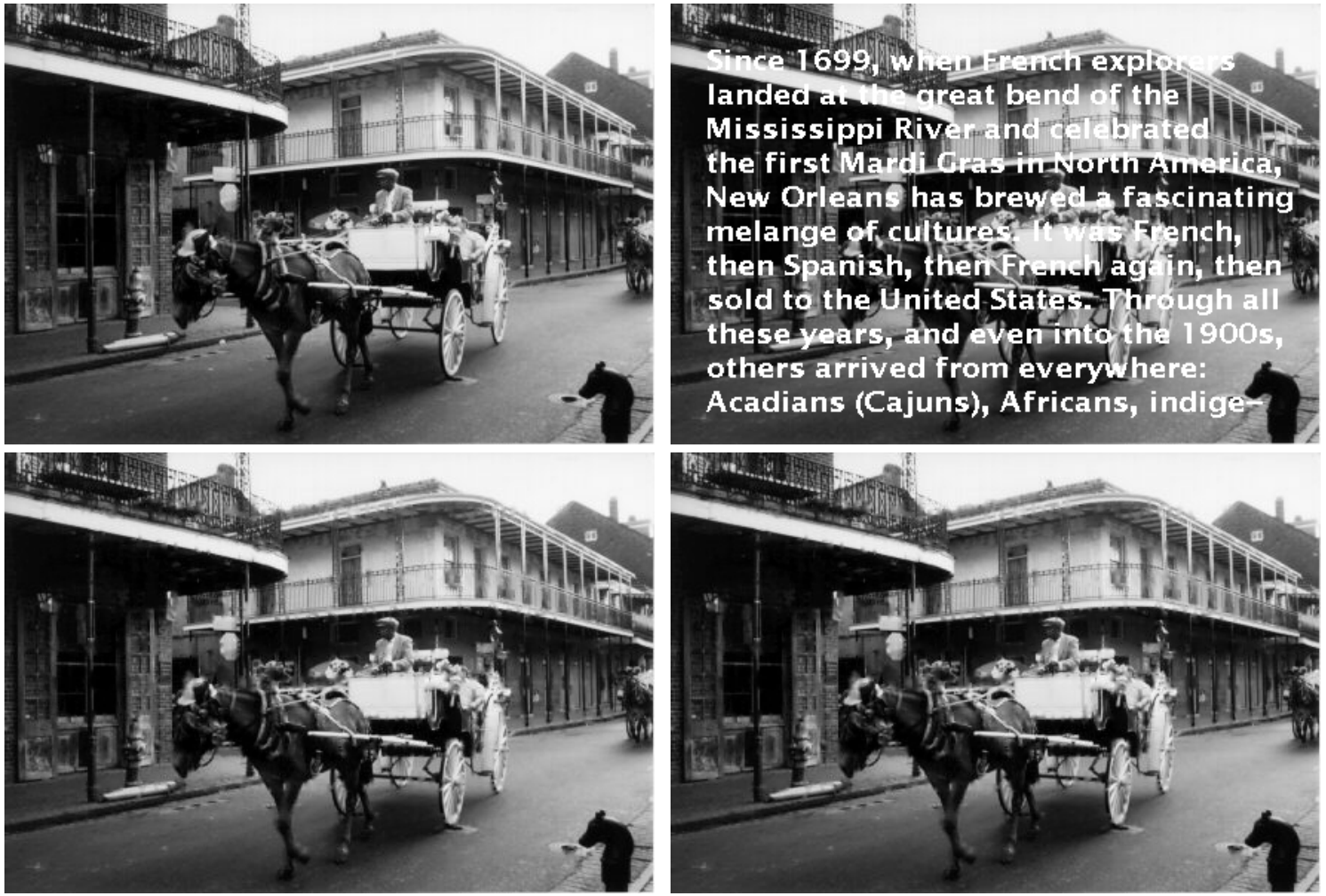
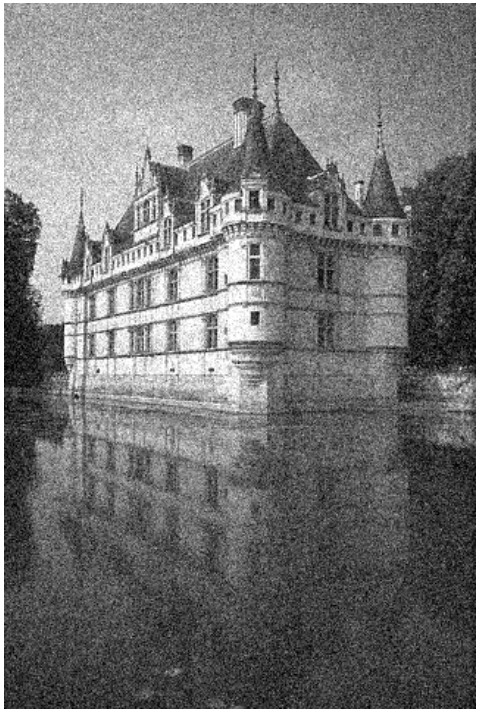


Image Denoising

Original



Noisy

Corrected

Introduction

Technically, Graphical Models are

Multivariate probabilistic models...

which are structured...

in terms of conditional independence statements

Informally

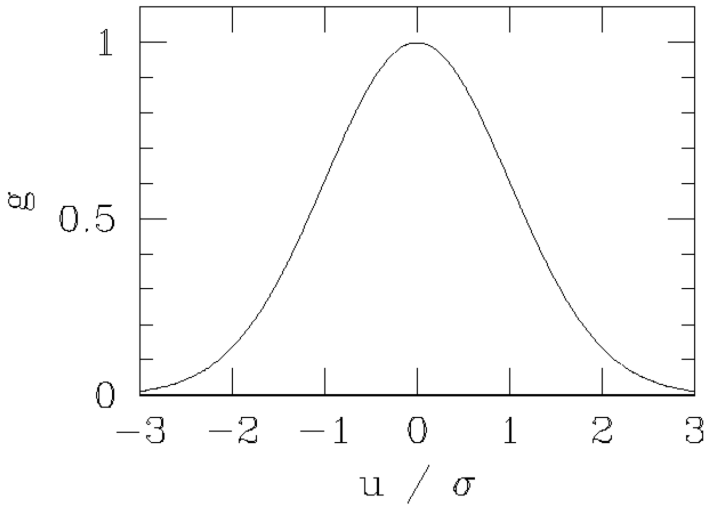
Models that represent a system by its parts and the possible relations among them in a probabilistic way

Introduction

Questions we want to ask about these models

- Estimating the parameters of the model given data
- Obtaining data samples from the model
- Computing probabilities of particular outcomes
- Finding most likely outcome

Univariate Example



$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{[-(x-\mu)^2/(2\sigma^2)]}$$

- Estimate μ and σ given $X = \{x^1, \ldots, x^n\}$
- Sample from $p(x)$
- Compute $P(\mu - \sigma \leq x \leq \mu + \sigma) := \int_{\mu-\sigma}^{\mu+\sigma} p(x)dx$
- Find $\text{argmax}_x\, p(x)$

Multivariate Case

We want to do the same things

- For multivariate distributions structured according to CI
- Efficiently
- Accurately

For example, given $p(x_1, \ldots, x_n; \theta)$

- Estimate θ given a sample X (and criterion, e.g. ML)
- Compute $p(x_A)$, $A \subseteq \{x_1, \ldots, x_n\}$ (marginal distributions)
- Find $\text{argmax}_{x_1,\ldots,x_n} p(x_1, \ldots, x_n; \theta)$ (MAP assignment)

Naively...

When trying to answer the relevant questions...

$$p(x_1) = \sum_{x_2,\ldots,x_N} p(x_1,\ldots,x_N)$$

$$O(|\mathcal{X}_1| \cdot |\mathcal{X}_2| \cdot \ldots \cdot |\mathcal{X}_N|)$$

and similarly for other questions: NOT GOOD

We need compact representations of multivariate distributions

When to Use Graphical Models

- When the compactness of the model arises from conditional independence statements involving its random variables.

- **CAUTION:** Graphical Models are useful in such cases. If the probability space is structured in different ways, Graphical Models may not (and in principle should not) be the right framework to represent and deal with the probability distributions involved.

The Very Basics of Graphical Models (Lecture 2)

Tibério Caetano

sml.nicta.com.au
Statistical Machine Learning Program
NICTA
Canberra, ACT 0200 Australia
Tiberio.Caetano@gmail.com

MLSS 2008, Kioloa, Australia

Notation

Basic definitions involving random quantities

- X - A random variable $X = (X_1, \ldots, X_N)$, $N \geq 1$
- x - A particular realization of X: $x = (x_1, \ldots, x_N)$
- $\mathcal{X}$ - Set of all realizations (sample space)
- X_A - A random vector of variables indexed by $A \subseteq \{1, \ldots, N\}$ (x_A for realizations)
- $X_{\tilde{A}}$ - The random vector comprised of all variables other than those in X_A ($A \cup \tilde{A} = \{1, \ldots, N\}$, $A \cap \tilde{A} = \emptyset$). $x_{\tilde{A}}$ for realizations
- $\mathcal{X}_A := \{x_A\}$, ($\mathcal{X}_A := \{x_A\}$)
- $p(x) :=$ probability that X assumes realization x

Basic properties of probabilities

- $0 \leq p(x) \leq 1, \forall x \in \mathcal{X}$
- $\sum_{x \in \mathcal{X}} p(x) = 1$

Conditioning and Marginalization

The two ‘rules’ you will always need

Conditioning

- $p(x_A, x_B) = p(x_A|x_B)p(x_B)$,
  for $p(x_B) > 0$

Marginalization

- $p(x_A) = \sum_{x_{\tilde{A}} \in \mathcal{X}_{\tilde{A}}} p(x_A, x_{\tilde{A}})$

Independence and Conditional Indep.

Independence

- $p(x_A, x_B) = p(x_A)p(x_B)$

Conditional Independence

- $p(x_A, x_B|x_C) = p(x_A|x_C)p(x_B|x_C)$, or equivalently
- $p(x_A|x_B, x_C) = p(x_A|x_C)$, or equivalently
- $p(x_B|x_A, x_C) = p(x_B|x_C)$

Notation: $X_A \perp\!\!\!\perp X_B \mid X_C$

Conditional Independence

Examples

- Weather tomorrow $\perp\!\!\!\perp$ Weather yesterday $\mid$ Weather today
- My Genome $\perp\!\!\!\perp$ my grandparents' Genome $\mid$ my parent's Genome
- My mood $\perp\!\!\!\perp$ my wife's boss mood $\mid$ my wife's mood
- A pixel's color $\perp\!\!\!\perp$ color of far away pixels $\mid$ color of surrounding pixels

Conditional Independence

The KEY Fact is

$$\underbrace{p(x_A, x_B|x_C)}_{f_1(\text{3 variables})} = \underbrace{p(x_A|x_C)}_{f_2(\text{2 variables})} \times \underbrace{p(x_B|x_C)}_{f_3(\text{2 variables})}$$

p factors as functions over proper subsets of variables

Conditional Independence

Therefore

$$\underbrace{p(x_A, x_B|x_C)}_{f_1(\text{3 variables})} = \underbrace{p(x_A|x_C)}_{f_2(\text{2 variables})} \times \underbrace{p(x_B|x_C)}_{f_3(\text{2 variables})}$$

- $p(x_A, x_B|x_C)$ *cannot* assume arbitrary values for arbitrary x_A, x_B, x_C
- If you vary x_A and x_B for fixed x_C, you can only realize probabilities that satisfy the above condition

What is a Graphical Model?

What is a Graphical Model?

Given a set of conditional independence statements for the random vector $X = (X_1, \ldots, X_N)$:

$$\{X_{A_i} \perp\!\!\!\perp X_{B_i} \mid X_{C_i}\}$$

Our object of study will be the family of probability distributions

$$p(x_1, \ldots, x_N)$$

where these statements hold

Questions to be addressed

Typical questions when we have a probabilistic model

- Estimate parameters of the model given data
- Compute probabilities of particular outcomes
- Find particularly interesting realizations (e.g. MAP assignment)

In order to manipulate the probabilistic model

- We need to know the mathematical structure of $p(x)$
- We need to find ways of computing efficiently (and accurately) in such structure

An Exercise

How does $p(x)$ look like?

Example:

$p(x_1, x_2, x_3)$ where $x_1 \perp\!\!\!\perp x_3 \mid x_2$

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

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

so,

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

An Exercise

However, $p(x_1, x_2, x_3) = p(x_1|x_2)p(x_3|x_2)p(x_2)$ **is also**

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

since $p(x_3|x_2)p(x_2) = p(x_2|x_3)p(x_3)$

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

since $p(x_1|x_2)p(x_2) = p(x_2|x_1)p(x_1)$

Cond. Indep. and Factorization

So CI seems to generate factorization of $p(x)$!

- Is this useful for the questions we want to ask?

Let's see an example of how expensive it is to compute $p(x_2)$

$p(x_2) = \sum_{x_1,x_3} p(x_1, x_2, x_3)$

Without factorization:

$p(x_2) = \sum_{x_1,x_3} p(x_1, x_2, x_3)$, $O(|\mathcal{X}_1||\mathcal{X}_2||\mathcal{X}_3|)$

With factorization:

$p(x_2) = \sum_{x_1,x_3} p(x_1, x_2, x_3) = \sum_{x_1,x_3} p(x_1|x_2)p(x_2|x_3)p(x_3)$

$p(x_2) = \sum_{x_3} p(x_2|x_3)p(x_3) \sum_{x_1} p(x_1|x_2)$, $O(|\mathcal{X}_2||\mathcal{X}_3|)$

Cond. Indep. and Factorization

Therefore

- Conditional Independence seems to induce a structure in $p(x)$ that allows us to exploit the distributive law in order to make computations more tractable

However, what about the general case $p(x_1, \ldots, x_N)$?

- What is the form that $p(x)$ will take in general, given a set of conditional independence statements?
- Will we be able to exploit the distributive law in this general case as well?

Re-Writing the Joint Distribution

A little exercise

$$p(x_1, \ldots, x_N) = p(x_1, \ldots, x_{N-1}) p(x_N | x_1, \ldots, x_{N-1})$$

$$p(x_1, \ldots, x_N) = p(x_1) p(x_2|x_1) p(x_3|x_1, x_2) \ldots p(x_N|x_1, \ldots, x_{N-1})$$

$$p(x) = \prod_{i=1}^{N} p(x_i | x_{<i})$$

where

"$< i$ ":= $\{j : j < i, j \in \mathbb{N}^+\}$

now denote by π a permutation of the labels $\{1, \ldots, N\}$ such that $\pi_j < \pi_i, \forall i, \forall j \in < i$. Above we have $\pi = \mathbf{1}$ (i.e. $\pi_i = i$)

So we can write

$$p(x) = \prod_{i=1}^{N} p(x_{\pi_i} | x_{<\pi_i}).$$

Re-Writing the Joint Distribution

So

Any $p(x)$ can be written as $p(x) = \prod_{i=1}^{N} p(x_{\pi_i}|x_{<\pi_i})$.

Now, assume that the following CI statements hold

$p(x_{\pi_i}|x_{<\pi_i}) = p(x_{\pi_i}|x_{pa_{\pi_i}}), \forall i$, where $pa_{\pi_i} \subset < \pi_i$.

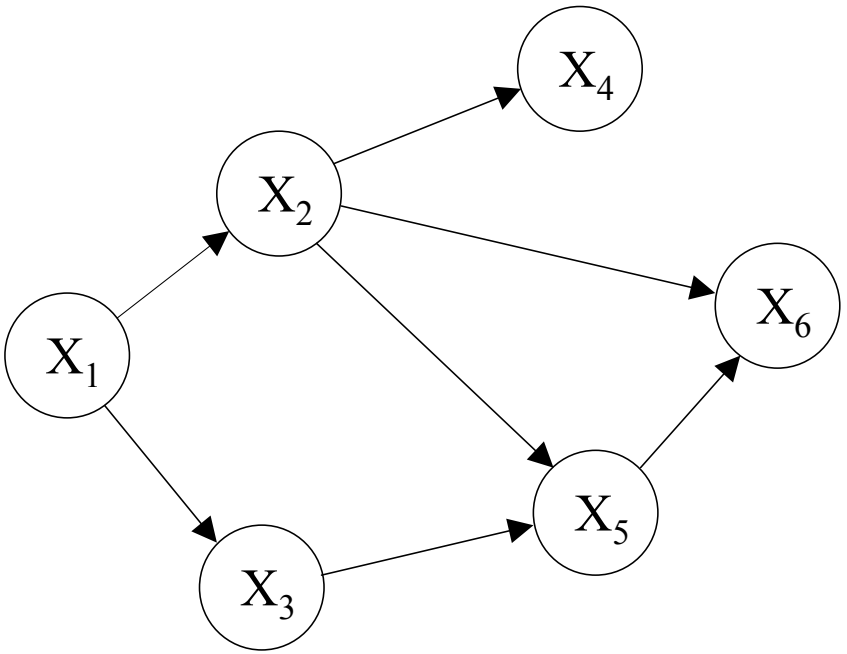
Then we immediately get

$p(x) = \prod_{i=1}^{N} p(x_{\pi_i}|x_{pa_{\pi_i}})$

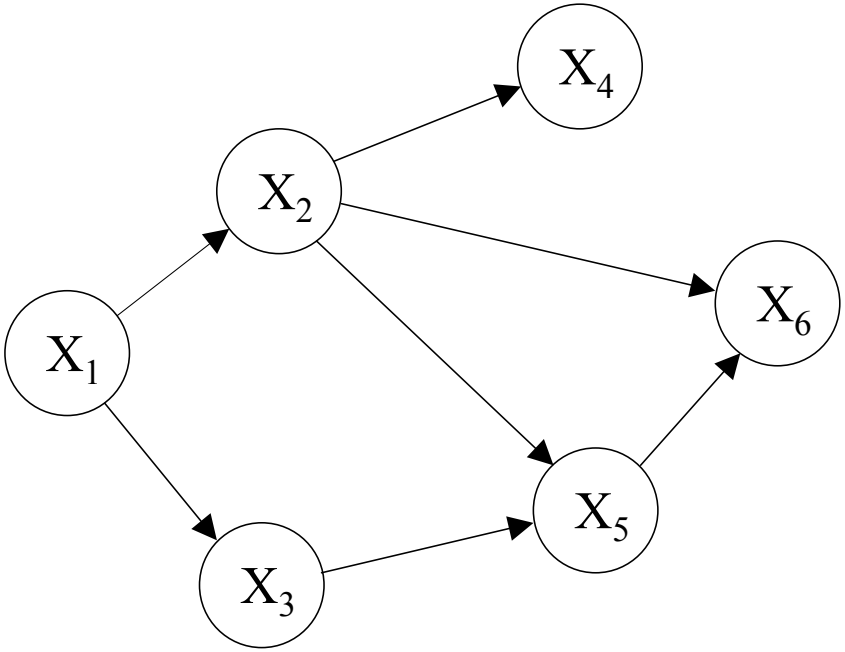
Computing in Graphical Models

Algebra is boring, so let's draw this

- Let's represent variables as circles
- Let's draw an arrow from j to i if $j \in pa_i$
- The resulting drawing will be a Directed Graph
- Moreover it will be Acyclic (no directed cycles) (Exercise:why?)



Computing in Graphical Models



$p(x) =$? (Exercise)

Bayesian Networks

This is why the name “Graphical Models”

Such Graphical Models with arrows are called

- Bayesian Networks
- Bayes Nets
- Bayes Belief Nets
- Belief Networks
- Or, more descriptively: Directed Graphical Models

Bayesian Networks

A Bayesian Network associated to a DAG is a set of probability distributions where each element $p(x)$ can be written as

$$p(x) = \prod_i p(x_i | x_{pa_i})$$

where random variable x_i is represented as a node in the DAG and $pa_i = \{x_j : \exists \text{ arrow } x_j \rightarrow x_i \text{ in the DAG } \}$. "*pa*" is for *parents*.

(Colloquially, we say the BN "is" the DAG)

Topological Sorts

A permutation π of the node labels which, for every node, makes each of its parents have a smaller index than that of the node is called a topological sort of the nodes in the DAG.

Theorem: Every DAG has at least one topological sort

(Exercise: Prove)

A Little Exercise Revisited

Remember?

$$p(x_1, \ldots, x_N) = p(x_1, \ldots, x_{N-1}) p(x_N | x_1, \ldots, x_{N-1})$$

$$p(x_1, \ldots, x_N) = p(x_1) p(x_2|x_1) p(x_3|x_1, x_2) \ldots p(x_N|x_1, \ldots, x_{N-1})$$

$$p(x) = \prod_{i=1}^{N} p(x_i | x_{<i})$$

where

"$< i$ ":= $\{j : j < i, j \in \mathbb{N}^+\}$

now denote by π a permutation of the labels $\{1, \ldots, N\}$ such that $\pi_j < \pi_i, \forall i, \forall j \in < i$. Above we have $\pi = \mathbf{1}$

So we can write

$$p(x) = \prod_{i=1}^{N} p(x_{\pi_i} | x_{<\pi_i}).$$

Exercises

Exercises:

- How many topological sorts has a BN where no CI statements hold?
- How many topological sorts has a BN where all CI statements hold?

The Very Basics of Graphical Models (Lecture 3)

Tibério Caetano

sml.nicta.com.au
Statistical Machine Learning Program
NICTA
Canberra, ACT 0200 Australia
Tiberio.Caetano@gmail.com

MLSS 2008, Kioloa, Australia

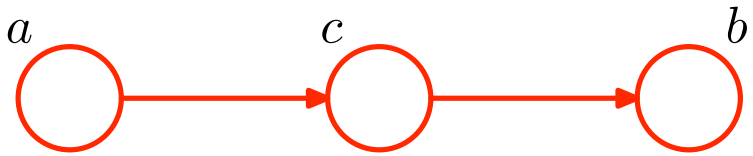
Hidden CI Statements?

We have obtained a BN by

- Introducing very "convenient" CI statements (namely those that shrink the factors of the expansion $p(x) = \prod_{i=1}^{N} p(x_{\pi_i}|x_{<\pi_i})$)
- By doing so, have we induced other CI statements?
- The answer is YES

Head-to-Tail Nodes (Independence)

Are a and b independent?



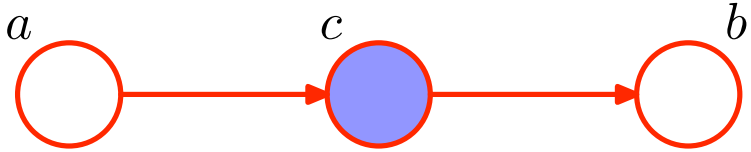
Does $a \perp\!\!\!\perp b$ hold?

Check whether $p(ab) = p(a)p(b)$

$$p(ab) = \sum_c p(abc) = \sum_c p(a)p(c|a)p(b|c) =$$
$$p(a) \sum_c p(b|c)p(c|a) = p(a)p(b|a) \neq p(a)p(b)$$

Head-to-Tail Nodes (Cond. Indep.)

Factorization $\Rightarrow$ CI ?



Does $p(abc) = p(a)p(c|a)p(b|c) \Rightarrow a \perp\!\!\!\perp b \mid c$?

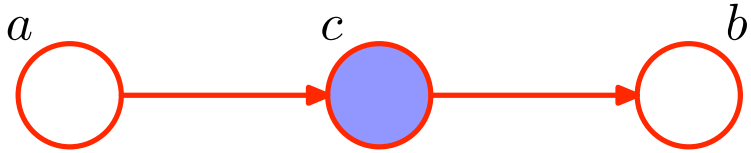
Assume $p(abc) = p(a)p(c|a)p(b|c)$ holds

Then

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

Head-to-Tail Nodes (Cond. Indep.)

CI $\Rightarrow$ Factorization ?



Does $a \perp\!\!\!\perp b \mid c \Rightarrow p(abc) = p(a)p(c|a)p(b|c)$?

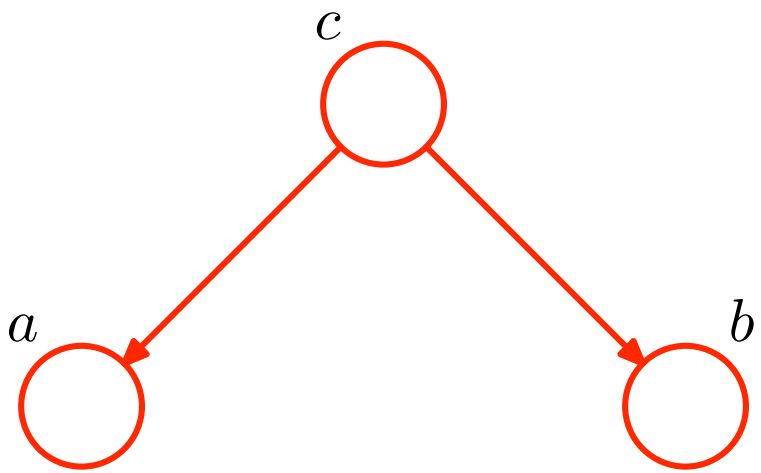
Assume $a \perp\!\!\!\perp b \mid c$, i.e. $p(ab|c) = p(a|c)p(b|c)$

Then

$p(abc) := p(ab|c)p(c) = p(a|c)p(b|c)p(c) \overset{\text{Bayes}}{=} p(a)p(c|a)p(b|c)$

Tail-to-Tail Nodes (Independence)

Are a and b independent?



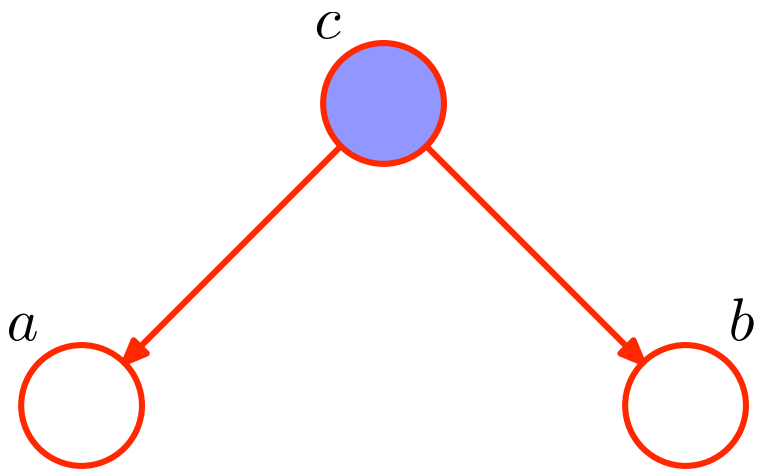
Does $a \perp\!\!\!\perp b$ hold?

Check whether $p(ab) = p(a)p(b)$

$p(ab) = \sum_c p(abc) = \sum_c p(c)p(a|c)p(b|c) =$
$\sum_c p(b)p(a|c)p(c|b) = p(b)p(a|b) \neq p(a)p(b)$, in general

Tail-to-Tail Nodes (Cond. Indep.)

Factorization $\Rightarrow$ CI ?



Does $p(abc) = p(c)p(a|c)p(b|c) \Rightarrow a \perp\!\!\!\perp b \mid c$?

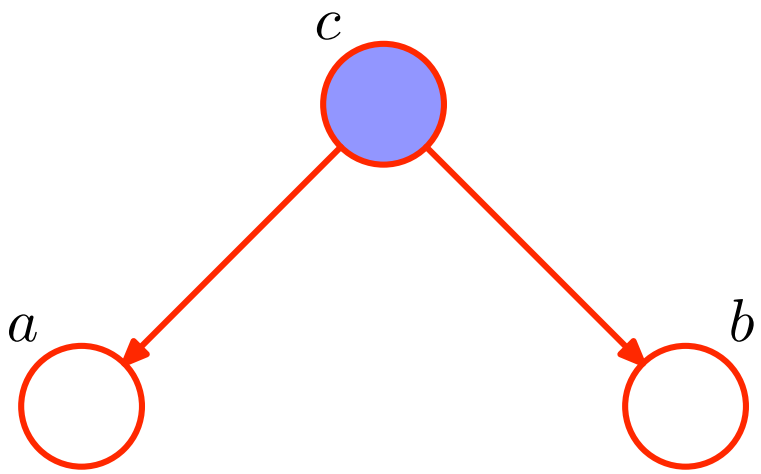
Assume $p(abc) = p(c)p(a|c)p(b|c)$.

Then

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

Tail-to-Tail Nodes (Cond. Indep.)

CI $\Rightarrow$ Factorization ?



Does $a \perp\!\!\!\perp b \mid c \Rightarrow p(abc) = p(c)p(a|c)p(b|c)$?

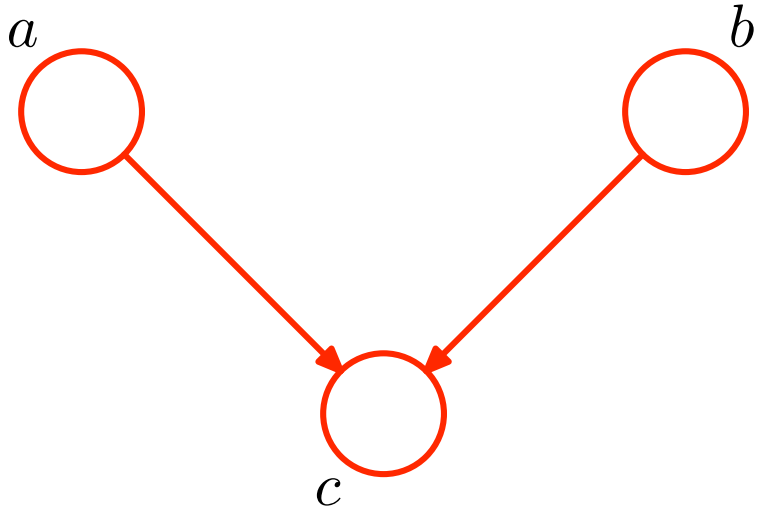
Assume $a \perp\!\!\!\perp b \mid c$, holds, i.e. $p(ab|c) = p(a|c)p(b|c)$ holds

Then

$$p(abc) = p(ab|c)p(c) = p(a|c)p(b|c)p(c)$$

Head-to-Head Nodes (Independence)

Are a and b independent?



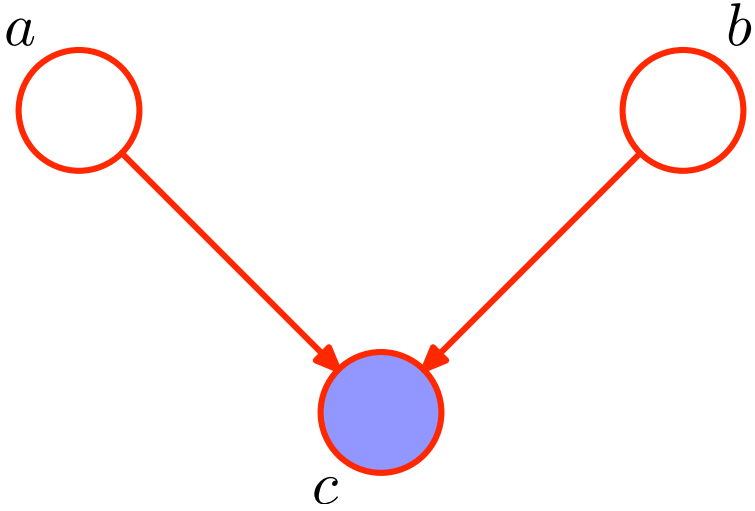
Does $a \perp\!\!\!\perp b$ hold?

Check whether $p(ab) = p(a)p(b)$

$$p(ab) = \sum_c p(abc) = \sum_c p(a)p(b)p(c|ab) = p(a)p(b)$$

Head-to-head Nodes (Cond. Indep.)

Factorization $\Rightarrow$ CI ?



Does $p(abc) = p(a)p(b)p(c|ab) \Rightarrow a \perp\!\!\!\perp b \mid c$?

Assume $p(abc) = p(a)p(b)p(c|ab)$ holds

Then

$$p(ab|c) = \frac{p(abc)}{p(c)} = \frac{p(a)p(b)p(c|ab)}{p(c)} \neq p(a|c)p(b|c) \text{ in general}$$

CI $\Leftrightarrow$ Factorization in 3-Node BNs

Therefore, we conclude that

- Conditional Independence and Factorization are equivalent for the "atomic" Bayesian Networks with only 3 nodes.

Question

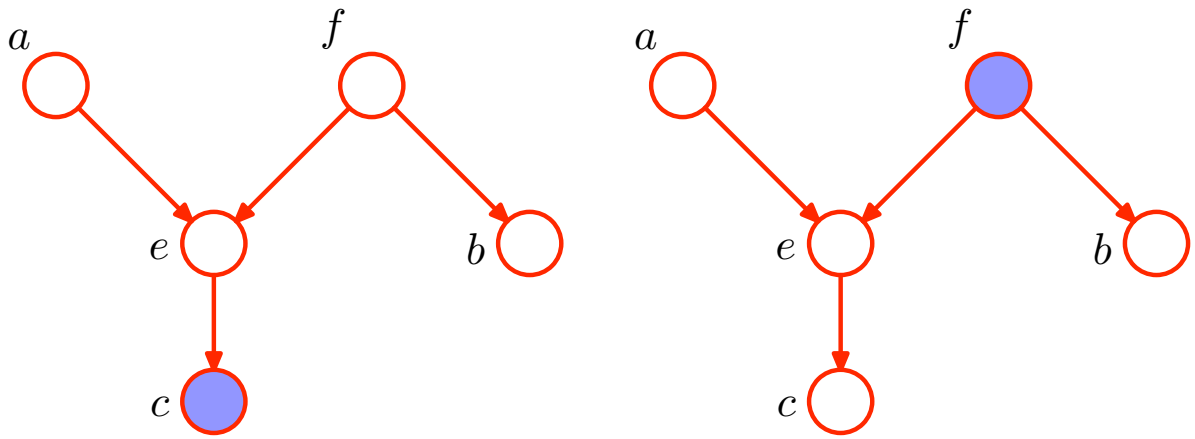
- Are they equivalent for any Bayesian Network?

- To answer we need to characterize which conditional independence statements hold for an arbitrary factorization and check whether a distribution that satisfies those statements will have such factorization.

Blocked Paths

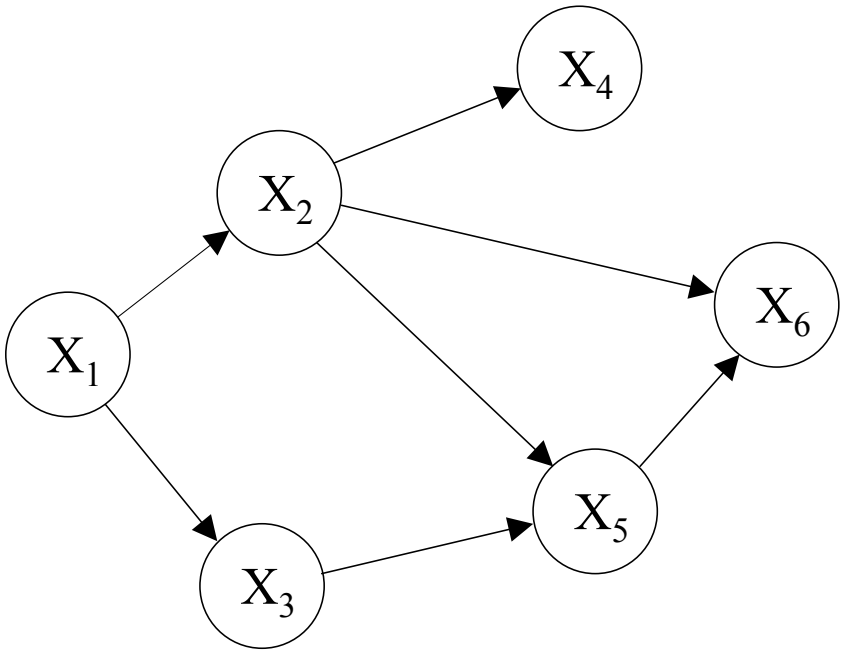
We start by defining a blocked path, which is one containing

- An observed HH or HT node, or
- A TT node which is not observed, nor any of its descendants is observed



D-Separation

- A set of nodes A is said to be d-separated from a set of nodes B from a set of nodes C if every path from A to B is blocked when C is in the conditioning set.



Exercise: Is X_3 d-separated from X_6 when the conditioning set is $\{X_1, X_5\}$?

CI $\Leftrightarrow$ Factorization for BNs

Theorem: Factorization $\Rightarrow$ CI

- If a probability distribution factorizes according to a directed acyclic graph, and if A, B and C are disjoint subsets of nodes such that A is d-separated from B by C in the graph, then the distribution satisfies $A \perp\!\!\!\perp B \mid C$.

Theorem: CI $\Rightarrow$ Factorization

- If a probability distribution satisfies the conditional independence statements implied by d-separation over a particular directed graph, then it also factorizes according to the graph.

Relevance of "CI $\Leftrightarrow$ Factorization for BNs"

- The domain expert has an idea of which CI statements hold in a system, because such knowledge is **local**. However, what the CI $\Rightarrow$ Factorization gives us is a recipe to built a **global** description of the system (i.e. the factorization of the joint distribution) which is implied by the CI statements provided by the expert. The part Factorization $\Rightarrow$ CI guarantees that **no other** CI statements are produced by the obtained factorization.
- Also, whether the knowledge usually comes in the CI side, operationally we need to play with the Factorization side, which consists of an explicit algebraic description of the object with which we want to perform computations (i.e. the joint probability distribution)

Changing the class of CI statements

We obtained BNs by assuming

- $p(x_{\pi_i}|x_{<\pi_i}) = p(x_{\pi_i}|x_{pa_{\pi_i}}), \forall i$, where $pa_{\pi_i} \subset < \pi_i$.
- We saw in general that such types of CI statements would produce others, and in general all CI statements can be read as d-separation in a DAG.
- However, there are sets of CI statements which cannot be satisfied by any BN.

Example

- $p(x_0, x_1, x_2, x_3)$ where $x_i \perp\!\!\!\perp x_{i+2} \,|\{x_{i+1}, x_{i+3}\}, \forall i$ (mod 3 arithmetic)
  Exercise

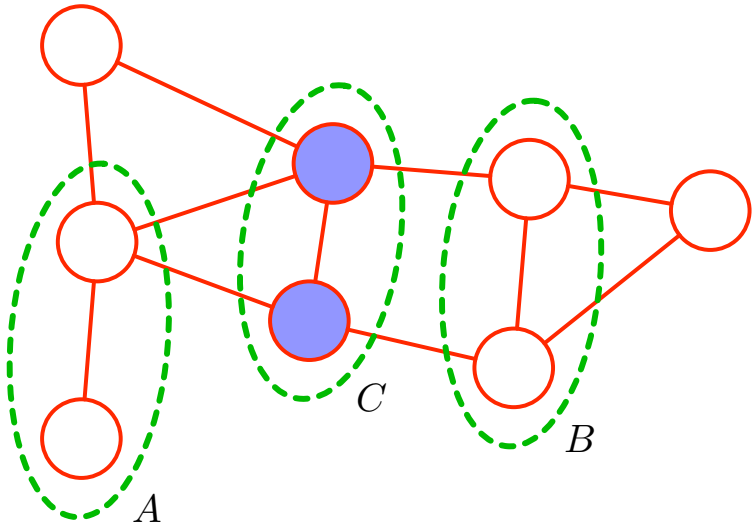
Markov Random Fields

- Ideally we would like to have more freedom
- There is another class of Graphical Models called Markov Random Fields (MRFs)
- MRFs allow for the specification of a different class of CI statements
- The class of CI statements for MRFs can be easily defined by graphical means in undirected graphs.

Graph Separation

Definition of Graph Separation

- In an undirected graph G, being A, B and C disjoint subsets of nodes, if every path from A to B includes at least one node from C, then C is said to separate A from B in G.



Graph Separation

Definition of Markov Random Field

- An MRF is a set of probability distributions $\{p(x) : p(x) > 0 \ \forall p, x\}$ such that there exists an undirected graph G with disjoint subsets of nodes A, B, C, in which whenever C separates A from B in G, $A \perp\!\!\!\perp B \mid C$ in $p(x)$, $\forall p(x)$

- Colloquially, we say that the MRF "is" such undirected graph. But in reality it is the set of all probability distributions whose conditional independency statements are precisely those given by graph separation in the graph.

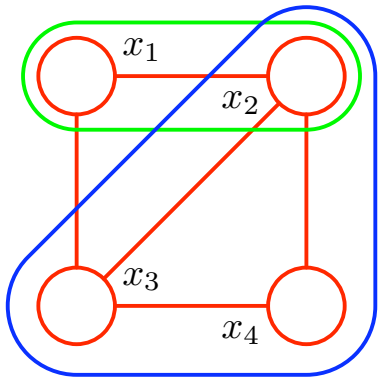
Structure of a Markov Random Field

- We saw that Bayesian Networks have a convenient factorized form that allows us to perform efficient computations.
- We saw that the factorization of a BN "arises" from the CI statements given by d-separation in a DAG.
- We saw that the converse is also true: factorization of a BN implies the CI statements given by d-separation in a DAG.
- We saw that this equivalence is important because it connects what we know ("local") with what we want ("global").
- Does something analogous happen with MRFs? Yes

Cliques and Maximal Cliques

Definitions concerning undirected graphs

- A clique of a graph is a complete subgraph of it (i.e. a subgraph where every pair of nodes is connected by an edge).
- A maximal clique of a graph is clique which is not a proper subset of another clique



$\{X_1, X_2\}$ form a clique and $\{X_2, X_3, X_4\}$ a maximal clique.

Factorization Property

Definition of factorization w.r.t. an undirected graph

A probability distribution $p(x)$ is said to factorize with respect to a given undirected graph if it can be written as

$$p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(x_c)$$

where $\mathcal{C}$ is the set of maximal cliques, c is a maximal clique, x_c is the domain of x restricted to c and $\psi_c(x_c)$ is an arbitrary non-negative real-valued function. Z ensures $\sum_x p(x) = 1$.

CI $\Leftrightarrow$ Factorization for positive MRFs

Theorem: Factorization $\Rightarrow$ CI

- If a probability distribution factorizes according to an undirected graph, and if A, B and C are disjoint subsets of nodes such that C separates A from B in the graph, then the distribution satisfies $A \perp\!\!\!\perp B \mid C$.

Theorem: CI $\Rightarrow$ Factorization (Hammersley-Clifford)

- If a strictly positive probability distribution ($p(x) > 0 \; \forall x$) satisfies the conditional independence statements implied by graph separation over a particular undirected graph, then it also factorizes according to the graph.

Relevance of CI $\Leftrightarrow$ Factorization for MRFs

Relevance is analogous to the BN case

- CI statements are usually what is known by the expert
- The expert needs the model $p(x)$ in order to compute things
- The CI $\Rightarrow$ Factorization part gives $p(x)$ from what is known (CI statements)
- The Factorization $\Rightarrow$ CI ensures that those are the only CI statements that hold in the resulting $p(x)$

Comparison BNs vs. MRFs

In both types of Graphical Models

- A relationship between the CI statements satisfied by a distribution and the associated simplified algebraic structure of the distribution is made in term of graphical objects.
- The CI statements are related to concepts of separation between variables in the graph.
- The simplified algebraic structure (factorization of $p(x)$ in this case) is related to "local pieces" of the graph (child + its parents in BNs, cliques in MRFs)

Comparison BNs vs. MRFs

Differences

- The set of probability distributions that can be represented as MRFs is different from the set that can be represented as BNs.
- Although both MRFs and BNs are expressed as a factorization of local functions on the graph, the MRF has a normalization constant $Z = \sum_x \prod_{c \in \mathcal{C}} \psi_c(x_c)$ that couples all factors, whereas the BN has not.
- The local "pieces" of the BN are probability distributions themselves, whereas in MRFs they need only be non-negative functions (i.e. they may not have range [0 1] as probabilities do).

Comparison BNs vs. MRFs

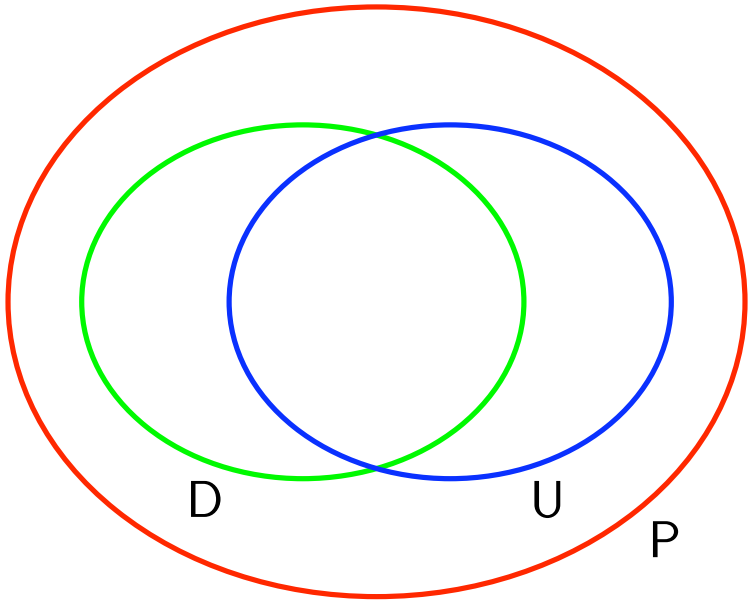
Exercises

- When are the CI statements of a BN and a MRF precisely the same?
- A graph has 3 nodes, A, B and C. We know that $A \perp\!\!\!\perp B$, but $C \perp\!\!\!\perp A$ and $C \perp\!\!\!\perp B$ both do not hold. Can this represent a BN? An MRF?

I-Maps, D-Maps and P-Maps

- A graph is said to be a D-map (for dependence map) of a distribution if every conditional independence statement satisfied by the distribution is reflected in the graph.
- A graph is said to be an I-map (for independence map) of a distribution if every conditional independence statement implied by the graph is satisfied in the distribution.
- A graph is said to be an P-map (for perfect map) of a distribution if it is both a D-map and an I-map for the distribution.

I-Maps, D-Maps and P-Maps

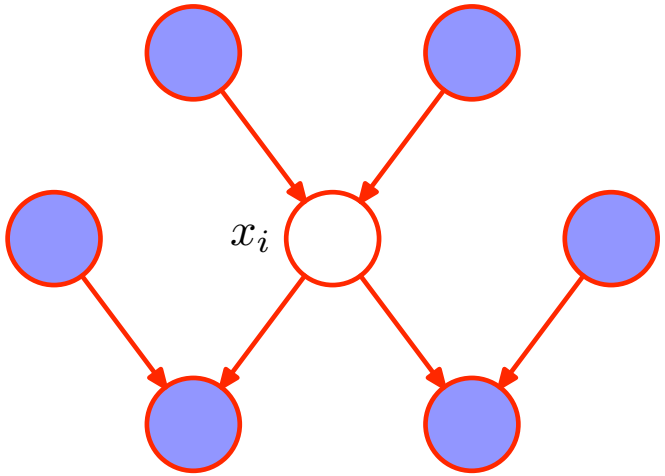


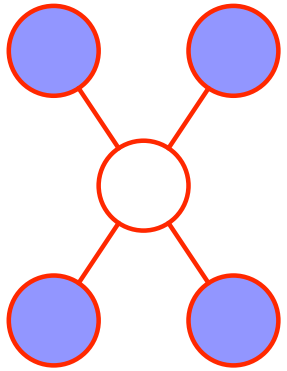
D: set of distributions on n variables that can be represented as a perfect map by a DAG

U: set of distributions on n variables that can be represented as a perfect map by an Undirected graph

P: set of all distributions on n variables

Markov Blankets





- The Markov Blanket of a node X_i in either a BN or an MRF is the smallest set of nodes A such that $p(x_i|x_{\tilde{i}}) = p(x_i|x_A)$
- BN: parents, children and co-parents of the node
- MRF: neighbors of the node

Exercises

Exercises

- Show that the Markov Blanket of a node x_i in a BN is given by it's children, parents and co-parents

- Show that the Markov Blanket of a node x_i in a MRF is given by its neighbors

The Very Basics of Graphical Models (Lecture 4)

Tibério Caetano

sml.nicta.com.au
Statistical Machine Learning Program
NICTA
Canberra, ACT 0200 Australia
Tiberio.Caetano@gmail.com

MLSS 2008, Kioloa, Australia

Factorized Distributions

Our $p(x)$ as a factorized form

For BNs, we have

$$p(x) = \prod_i p(x_i|pa_i)$$

for MRFs, we have

$$p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(x_c)$$

Will this enable us to answer the relevant questions in practice?

Key Concept: Distributive Law

Distributive Law

$$\underbrace{ab + ac}_{\text{3 operations}} = \underbrace{a(b + c)}_{\text{2 operations}}$$

I.e. if the same constant factor ('a' here) is present in every term, we can gain by "pulling it out"

Consider computing the marginal $p(x_1)$ for the MRF with factorization

$p(x) = \frac{1}{Z} \prod_{i=1}^{N-1} \psi(x_i, x_{i+1})$ (Exercise: which graph is this?)

$p(x_1) = \sum_{x_2,\ldots,x_N} \frac{1}{Z} \prod_{i=1}^{N-1} \psi(x_i, x_{i+1})$

$p(x_1) = \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_2, x_3) \cdots \sum_{x_N} \psi(x_{N-1}, x_N)$

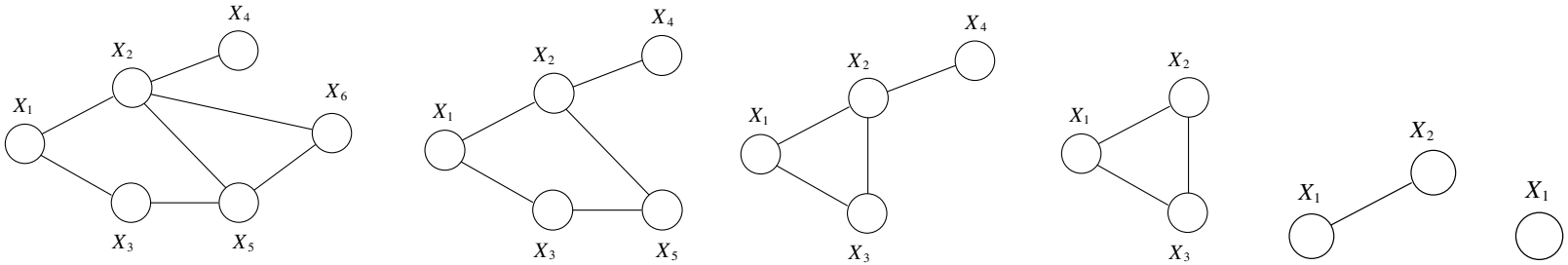
$O(\prod_{i=1}^{N} |\mathcal{X}_i|)$ vs. $O(|\mathcal{X}_1| \cdot (\sum_{i=2}^{N} |\mathcal{X}_i|))$

Elimination Algorithm

Distributive Law (DL) is the key to efficient inference in GMs

- The simplest algorithm using the DL is the Elimination Algorithm
- This algorithm is appropriate when we have a single query
- Just like in the previous example of computing $p(x_1)$ in a givel MRF
- This algorithm can be seen as successive elimination of nodes in the graph

Elimination Algorithm



Compute $p(x_1)$ with elimination order $(6, 5, 4, 3, 2)$

$$p(x_1) = Z^{-1} \sum_{x_2,\ldots,x_6} \psi(x_1, x_2)\psi(x_1, x_3)\psi(x_3, x_5)\psi(x_2, x_5, x_6)\psi(x_2, x_4)$$

$$p(x_1) = Z^{-1} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) \sum_{x_4} \psi(x_2, x_4) \underbrace{\sum_{x_5} \psi(x_3, x_5) \underbrace{\sum_{x_6} \psi(x_2, x_5, x_6)}_{m_6(x_2, x_5)}}_{m_5(x_2, x_3)}$$

$$p(x_1) = Z^{-1} sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) m_5(x_2, x_3) \underbrace{\sum_{x_4} \psi(x_2, x_4)}_{m_4(x_2)}$$

$$p(x_1) = Z^{-1} \underbrace{\sum_{x_2} \psi(x_1, x_2) m_4(x_2) \underbrace{\sum_{x_3} \psi(x_1, x_3) m_5(x_2, x_3)}_{m_3(x_1, x_2)}}_{m_2(x_1)}$$

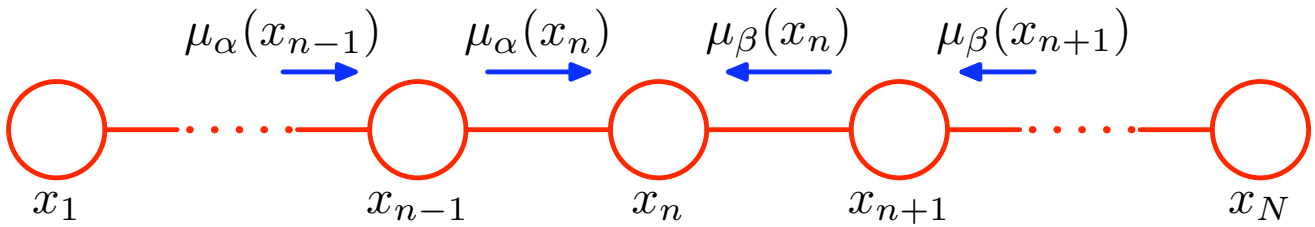
Belief Propagation

Belief Propagation Algorithm, also called

- Probability Propagation
- Sum-Product Algorithm

- Does not repeat computations
- Is specifically targeted at tree-structured graphs

Belief Propagation in a Chain



$$p(x_n) = \sum_{x_{<n}, x_{>n}} \frac{1}{Z} \prod_{i=1}^{N-1} \psi(x_i, x_{i+1})$$

$$p(x_n) = \frac{1}{Z} \sum_{x_{<n}, x_{>n}} \prod_{i=1}^{n-1} \psi(x_i, x_{i+1}) \prod_{i=n}^{N-1} \psi(x_i, x_{i+1})$$

$$p(x_n) = \frac{1}{Z} \left[\sum_{x_{<n}} \prod_{i=1}^{n-1} \psi(x_i, x_{i+1}) \right] \cdot \left[\sum_{x_{>n}} \prod_{i=n}^{N-1} \psi(x_i, x_{i+1}) \right]$$

$$p(x_n) = \frac{1}{Z} \underbrace{\left[\sum_{x_{<n}} \prod_{i=1}^{n-1} \psi(x_i, x_{i+1}) \right]}_{\mu_\alpha(x_n) = O(|\mathcal{X}_1| \cdot (\sum_{i=2}^{n-1} |\mathcal{X}_i|))} \cdot \underbrace{\left[\sum_{x_{>n}} \prod_{i=n}^{N-1} \psi(x_i, x_{i+1}) \right]}_{\mu_\beta(x_n) = O(|\mathcal{X}_1| \cdot (\sum_{i=n}^{N} |\mathcal{X}_i|))}$$

Belief Propagation in a Chain

- So, in order to compute $p(x_n)$, we only need the "incoming messages" to x_n
- But n is arbitrary, so in order to answer an arbitrary query, we need an arbitrary pair of "incoming messages"
- So we need all messages
- To compute a message to the right (left), we need all previous messages coming from the left (right)
- So the protocol should be: start from the leaves up to x_n, then go back towards the leaves
- Chain with N nodes $\Rightarrow 2(N-1)$ messages to be computed

Computing Messages

Defining trivial messages:

$$m_0(x_1) := 1, \quad m_{N+1}(x_N) := 1$$

For $i = 2$ to N compute

$$m_{i-1}(x_i) = \sum_{x_{i-1}} \psi(x_{i-1}, x_i) m_{i-2}(x_{i-1})$$

For $i = N - 1$ back to 1 compute

$$m_{i+1}(x_i) = \sum_{x_{i+1}} \psi(x_i, x_{i+1}) m_{i+2}(x + 1)$$

Belief Propagation in a Tree

- The reason why things are so nice in the chain is that every node can be seen as a leaf after it has received the message from one side (i.e. after the nodes from which the message come have been “eliminated”)
- “Original Leaves” give us the right place to start the computations, and from there the adjacent nodes “become leaves” as well
- However, this property also holds in a tree

Belief Propagation in a Tree

Message Passing Equation

$$m_j(x_i) = \sum_{x_j} \psi(x_j, x_i) \prod_{k:k\sim j, k\neq i} m_k(x_j)$$

$$\left(\prod_{k:k\sim j, k\neq i} m_k(x_j) := 1 \text{ whenever } j \text{ is a leaf}\right)$$

Computing Marginals

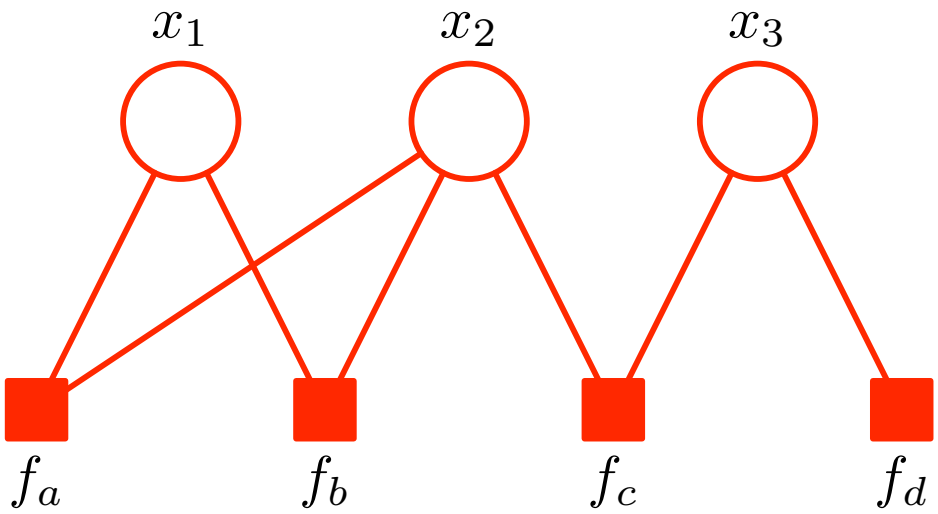
$$p(x_i) = \prod_{j:j\sim i} m_j(x_i)$$

Factor Graphs

Make explicit every factor of the joint distribution

- Formalism that allows explicit representation of all the factors present in a given factorization of a joint probability distribution.
- Circles denote variables, as in BNs and MRFs
- Squares denote functions
- An edge between a circle and a square states that the corresponding function has the corresponding variable as an argument.
- The joint distribution is the product of all functions (so, the functions are factors).

Factor Graphs



$$p(x_1, x_2, x_3) = f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3)$$

In general,

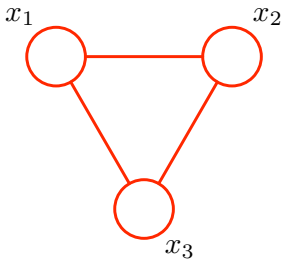
$$p(x) = \prod_s f_s(x_s)$$

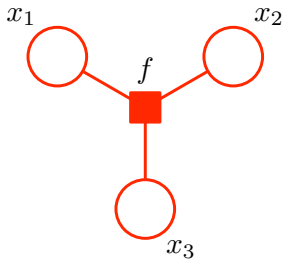
Factor Graphs for BNs and MRFs

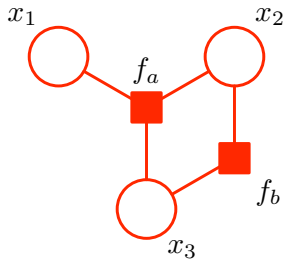
BNs and MRFs can be represented by Factor Graphs

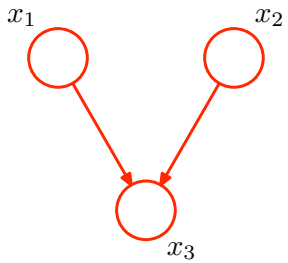
- BNs: $\{s\} =$ set of children and its parents $\{i \cup pa_i\}$, $f_s(x_s) = p(x_i|pa_i)$
- MRFs: $\{s\} =$ set of cliques $\mathcal{C}$, $f_s(x_s) = \psi_c(x_c)$ ($1/Z$ can be viewed as a factor over the empty set of variables)

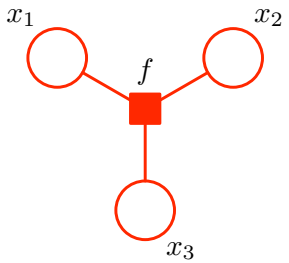
However, we can give more fine-grained description:

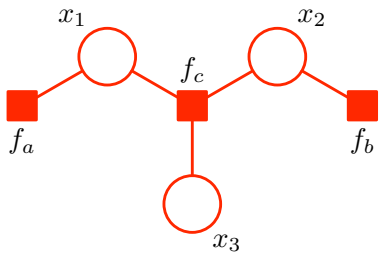










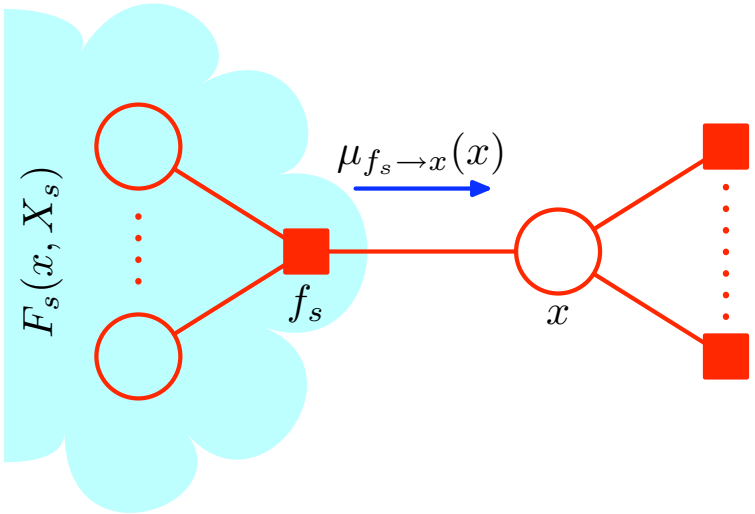


Message-Passing in Factor Graphs

Let's compute the marginal $p(x)$ for a factor graph (x denotes a particular variable and $\mathbf{x}$ the vector with all variables):

$p(x) = \sum_{\tilde{x}} p(\mathbf{x}) = \sum_{\tilde{x}} \prod_s f_s(x_s)$, which can be written as

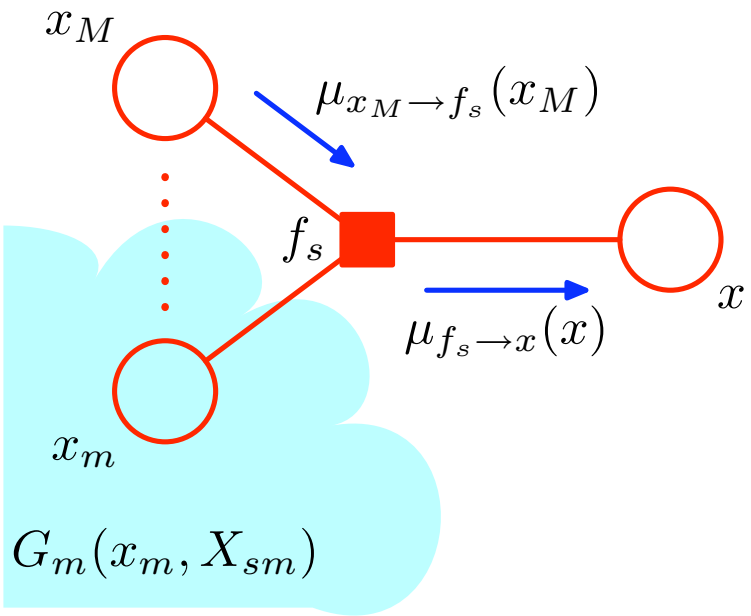
$$p(x) = \sum_{\tilde{x}} \prod_{ne(x)} F_s(x, X_s) = \prod_{ne(x)} \left[\sum_{X_s} F_s(x, X_s)\right] = \prod_{ne(x)} \mu_{f_s \to x}(x)$$



Message-Passing in Factor Graphs

$F_s(x, X_s)$ is itself factorized and can be written as

$$F_s(x, X_s) = f_s(x, x_1, \ldots, x_M) G_1(x_1, X_{s1}) \ldots G_M(x_M, X_{xM})$$

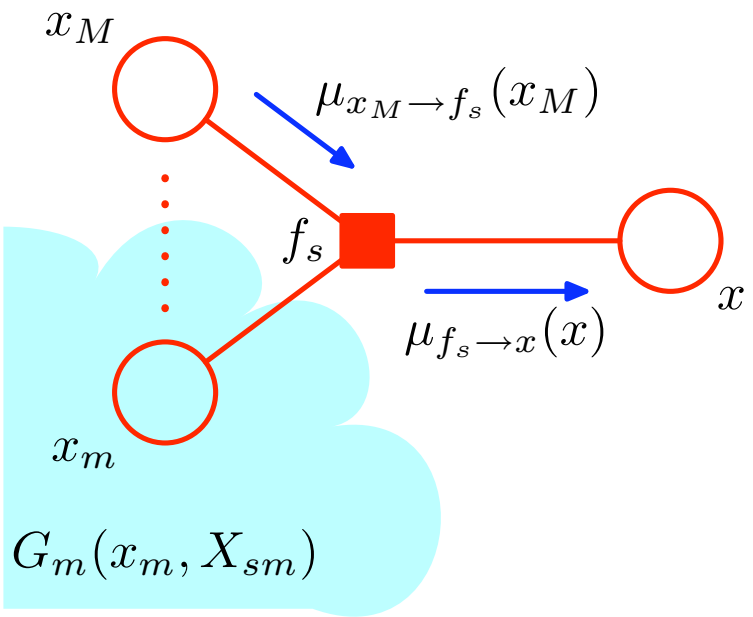


Message-Passing in Factor Graphs

Substituting this into $\mu_{f_s \to x}(x) := \sum_{X_s} F_s(x, X_s)$ gives

$$\mu_{f_s \to x}(x) = \sum_{x_1,\ldots,x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{xm \to f_s}(x_m)$$

where $\mu_{xm \to f_s}(x_m) := \sum_{X_{xm}} G_m(x_m, X_{sm})$



Message-Passing in Factor Graphs

Therefore we have two types of messages

Messages from factor nodes to variable nodes:

$$\mu_{f \to x}(x)$$

and messages from variable nodes to factor nodes:

$$\mu_{x \to f}(x)$$

Message-Passing in Factor Graphs

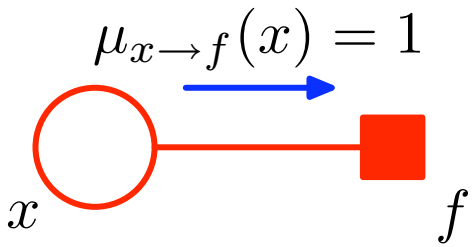
Finally, each G is a product of F's

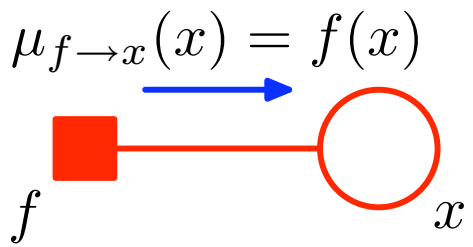
$$G_m(x_m, X_{xm}) = \prod_{l \in \text{ne}(x_m) \setminus f_s} F_l(x_m, X_{ml})$$

and we finally obtain

$$\mu_{x_m \to f_s} = \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

For the end-nodes, the messages are





Max-Product Algorithm

There are important queries other than computing marginals. For example, we may want to compute the most likely assignment:

$$x^* = \text{argmax}_x \, p(x)$$

as well as its probability

$$p(x^*)$$

one possibility would be to compute $p(x_i) = \sum_{x_{\tilde{i}}} p(x)$ for all i, then $x_i^* = \text{argmax}_{x_i} \, p(x_i)$ and then simply

$$x^* = (x_1^*, x_2^*, \ldots, x_N^*)$$

What's the problem with this?

Exercises

Exercise

- Construct $p(x_1, x_2)$, with $x_1, x_2 \in \{0, 1, 2\}$, such that $p(x_1^*, x_2^*) = 0$ (where $x_i^* = \text{argmax}_{x_i} p(x_i)$)

Max-Product (and Max-Sum) Algorithms

Instead we need to compute directly

$$x^* = \text{argmax}_{x_1,\ldots,x_N} p(x_1, \ldots, x_N)$$

We can use the distributive law again, since

$$\max(ab, ac) = a \max(b, c)$$

for $a > 0$

Exactly the same algorithm applies here with 'max' instead of $\sum$: max-product algorithm.

To avoid underflow we compute x^* via

$$\log(\text{argmax}_x p(x)) = \text{argmax}_x \log p(x) = \text{argmax}_x \sum_s \log f_s(x_s)$$

since log is a monotonic function. We can still use the distributive law since $(\max, +)$ is also a commutative semiring, i.e.

$$\max(a + b, a + c) = a + \max(b, c)$$

A Detail in Max-Sum

After computing the max-marginal for the root x:

$$p_i^* = max_{x_i} \sum_{s \sim x} \mu_{f_s \to x}(x)$$

and its maximizer

$$x_i^* = \text{argmax}_{x_i} p_i^*$$

It's not a good idea simply to pass back the messages to the leaves and then terminate (Why?)

In such cases it is safer to store the maximizing configurations of previous variables with respect to the next variables and then simply backtrack to restore the maximizing path.

In the particular case of a chain, this is called Viterbi algorithm, an instance of dynamic programming.

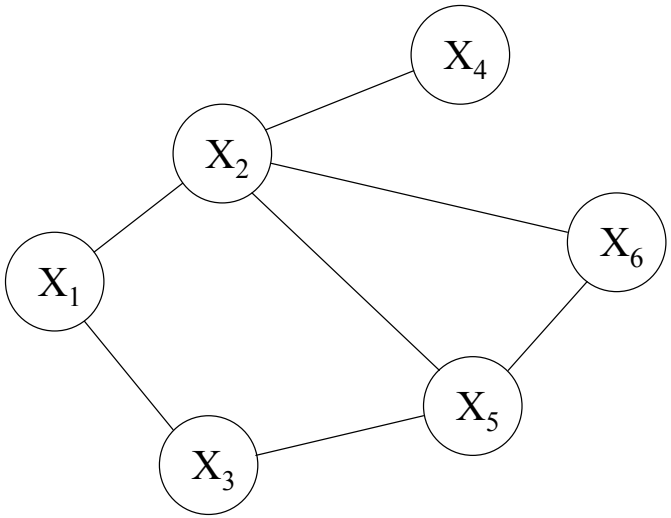
The Very Basics of Graphical Models (Lecture 5)

Tibério Caetano

sml.nicta.com.au
Statistical Machine Learning Program
NICTA
Canberra, ACT 0200 Australia
Tiberio.Caetano@gmail.com

MLSS 2008, Kioloa, Australia

Arbitrary Graphs

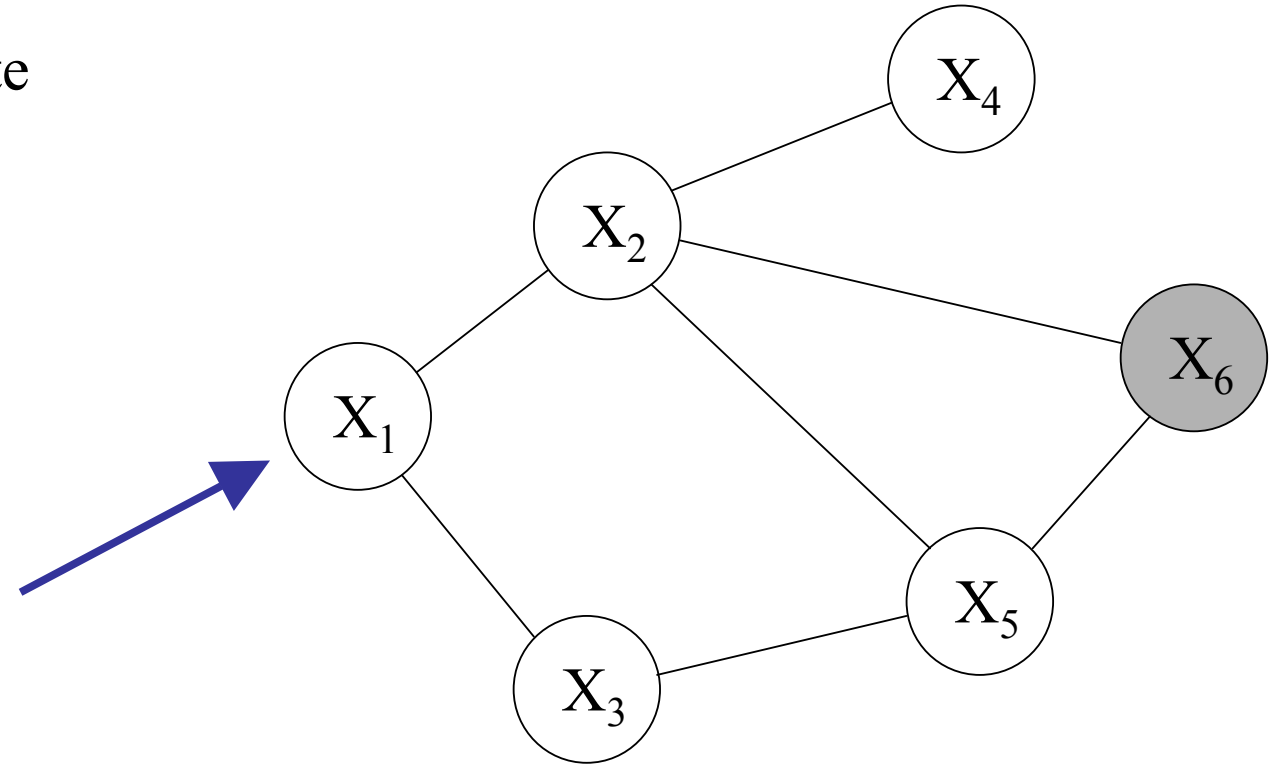


- Elimination algorithm is needed to compute marginals

A Problem with the Elimination Algorithm

How to compute

$$p(x_1 \mid \bar{x}_6)$$



A Problem with the Elimination Algorithm

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} \sum_{x_6} \psi(x_1, x_2) \psi(x_1, x_3) \psi(x_2, x_4) \psi(x_3, x_5) \psi(x_2, x_5, x_6) \delta(x_6, \bar{x}_6)$$

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) \sum_{x_4} \psi(x_2, x_4) \sum_{x_5} \psi(x_3, x_5) \sum_{x_6} \psi(x_2, x_5, x_6) \delta(x_6, \bar{x}_6)$$

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) \sum_{x_4} \psi(x_2, x_4) \sum_{x_5} \psi(x_3, x_5) m_6(x_2, x_5)$$

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) m_5(x_2, x_3) \sum_{x_4} \psi(x_2, x_4)$$

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) m_4(x_2) \sum_{x_3} \psi(x_1, x_3) m_5(x_2, x_3)$$

$$p(x_1, \bar{x}_6) = \frac{1}{Z} \sum_{x_2} \psi(x_1, x_2) m_4(x_2) m_3(x_1, x_2)$$

$$p(x_1, \bar{x}_6) = \frac{1}{Z} m_2(x_1)$$

A Problem with the Elimination Algorithm

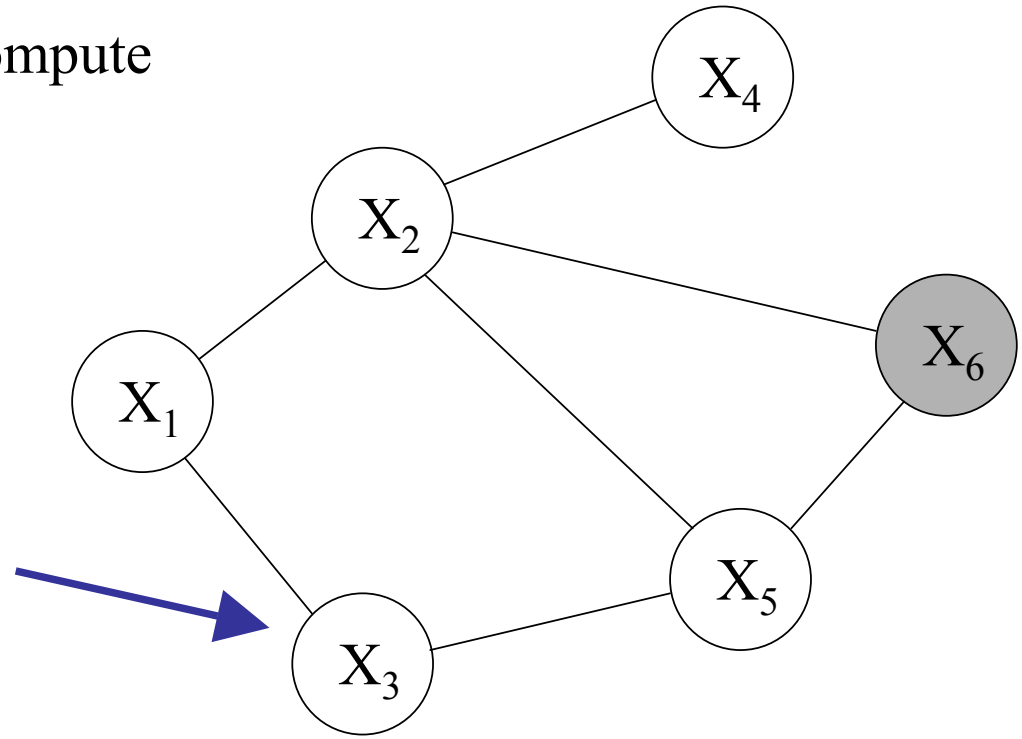
$$p(\bar{x}_6) = \frac{1}{Z} \sum_{x_1} m_2(x_1) \qquad p(x_1, \bar{x}_6) = \frac{1}{Z} m_2(x_1)$$

$$p(x_1 \mid \bar{x}_6) = \frac{m_2(x_1)}{\sum_{x_1} m_2(x_1)}$$

A Problem with the Elimination Algorithm

What if now we want to compute

$$p(x_3 \mid \bar{x}_6)$$



A Problem with the Elimination Algorithm

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\sum_{x_2}\sum_{x_4}\sum_{x_5}\sum_{x_6}\psi(x_1,x_2)\psi(x_1,x_3)\psi(x_2,x_4)\psi(x_3,x_5)\psi(x_2,x_5,x_6)\delta(x_6,\bar{x}_6)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)\sum_{x_4}\psi(x_2,x_4)\sum_{x_5}\psi(x_3,x_5)\sum_{x_6}\psi(x_2,x_5,x_6)\delta(x_6,\bar{x}_6)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)\sum_{x_4}\psi(x_2,x_4)\sum_{x_5}\psi(x_3,x_5)m_6(x_2,x_5)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)m_5(x_2,x_3)\sum_{x_4}\psi(x_2,x_4)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)m_4(x_2)m_5(x_2,x_3)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}m_2(x_1,x_3)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}m_1(x_3)$$

A Problem with the Elimination Algorithm

Repeated Computations!!

$p(x_1 \mid \bar{x}_6)$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\sum_{x_2}\sum_{x_4}\sum_{x_5}\sum_{x_6}\psi(x_1,x_2)\psi(x_1,x_3)\psi(x_2,x_4)\psi(x_3,x_5)\psi(x_2,x_5,x_6)\delta(x_6,\bar{x}_6)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)\sum_{x_4}\psi(x_2,x_4)\sum_{x_5}\psi(x_3,x_5)\sum_{x_6}\psi(x_2,x_5,x_6)\delta(x_6,\bar{x}_6)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)\sum_{x_4}\psi(x_2,x_4)\sum_{x_5}\psi(x_3,x_5)m_6(x_2,x_5)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)m_5(x_2,x_3)\sum_{x_4}\psi(x_2,x_4)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}\psi(x_1,x_3)\sum_{x_2}\psi(x_1,x_2)m_4(x_2)m_5(x_2,x_3)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}\sum_{x_1}m_2(x_1,x_3)$$

$$p(x_3,\bar{x}_6)=\frac{1}{Z}m_1(x_3)$$

$p(x_3 \mid \bar{x}_6)$

$$p(x_1,\bar{x}_6)=\frac{1}{Z}\sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5}\sum_{x_6}\psi(x_1,x_2)\psi(x_1,x_3)\psi(x_2,x_4)\psi(x_3,x_5)\psi(x_2,x_5,x_6)\delta(x_6,\bar{x}_6)$$

$$p(x_1,\bar{x}_6)=\frac{1}{Z}\sum_{x_2}\psi(x_1,x_2)\sum_{x_3}\psi(x_1,x_3)\sum_{x_4}\psi(x_2,x_4)\sum_{x_5}\psi(x_3,x_5)\sum_{x_6}\psi(x_2,x_5,x_6)\delta(x_6,\bar{x}_6)$$

$$p(x_1,\bar{x}_6)=\frac{1}{Z}\sum_{x_2}\psi(x_1,x_2)\sum_{x_3}\psi(x_1,x_3)\sum_{x_4}\psi(x_2,x_4)\sum_{x_5}\psi(x_3,x_5)m_6(x_2,x_5)$$

$$p(x_1,\bar{x}_6)=\frac{1}{Z}\sum_{x_2}\psi(x_1,x_2)\sum_{x_3}\psi(x_1,x_3)m_5(x_2,x_3)\sum_{x_4}\psi(x_2,x_4)$$

$$p(x_1,\bar{x}_6)=\frac{1}{Z}\sum_{x_2}\psi(x_1,x_2)m_4(x_2)\sum_{x_3}\psi(x_1,x_3)m_5(x_2,x_3)$$

$$p(x_1,\bar{x}_6)=\frac{1}{Z}\sum_{x_2}\psi(x_1,x_2)m_4(x_2)m_3(x_1,x_2)$$

$$p(x_1,\bar{x}_6)=\frac{1}{Z}m_2(x_1)$$

How to avoid that?

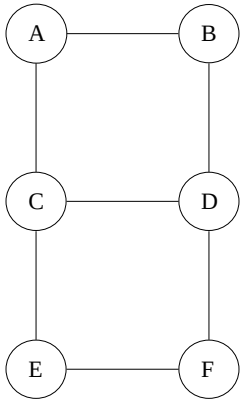
The Junction Tree Algorithm

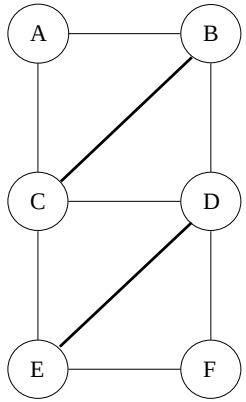
- The Junction Tree Algorithm is a generalization of the belief propagation algorithm for arbitrary graphs
- In theory, it can be applied to any graph (DAG or undirected)
- However, it will be efficient only for certain classes of graphs

Chordal Graphs

Chordal Graphs (also called triangulated graphs)

- The JT algorithm runs on chordal graphs
- A chord in a cycle is an edge connecting two nodes in the cycle but which does not belong to the cycle (i.e. a shortcut in the cycle)
- A graph is chordal if every cycle of length greater than 3 has a chord.

not chordal 



chordal

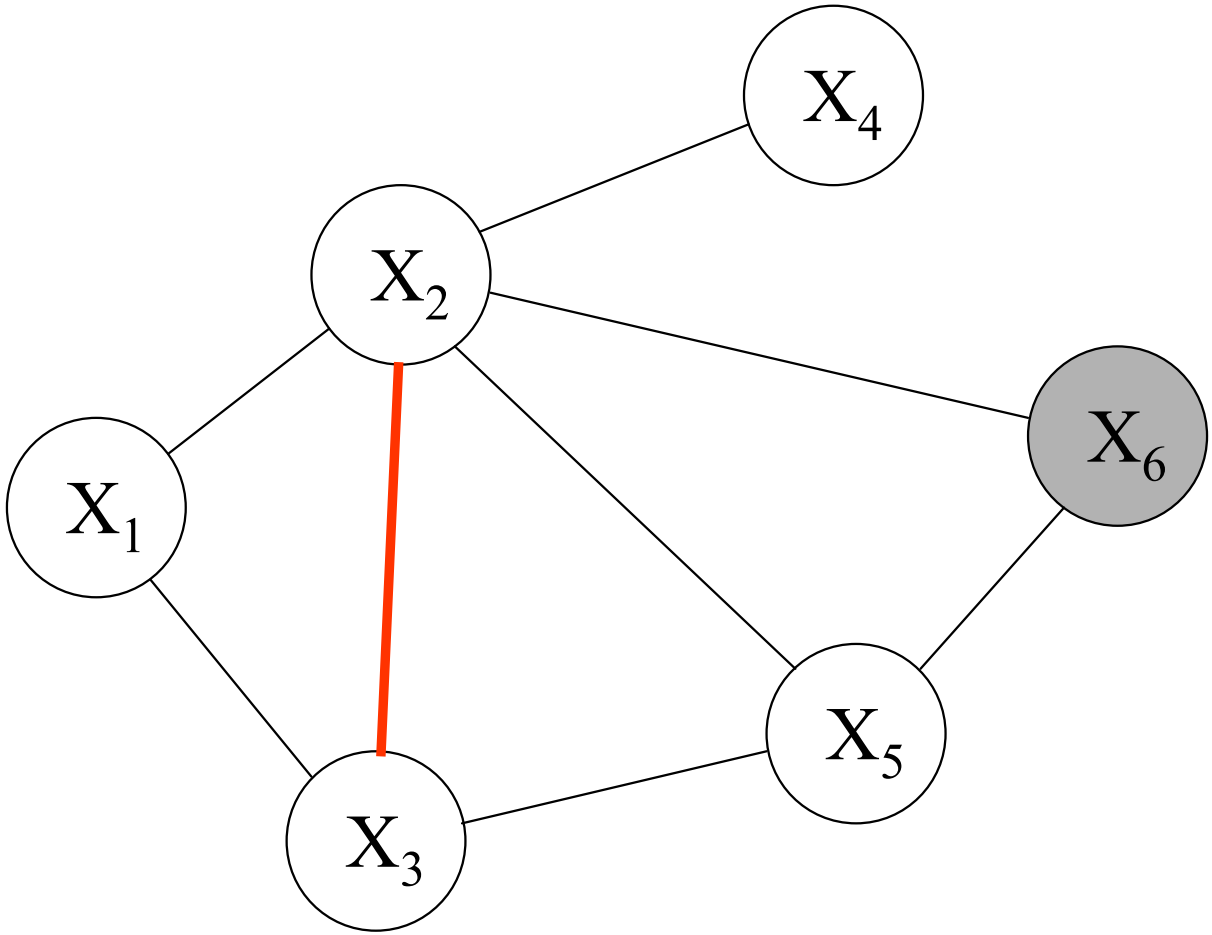
Chordal Graphs

What if a graph is not chordal?

- Add edges until it becomes chordal
- This will change the graph
- Exercise: Why is this not a problem?

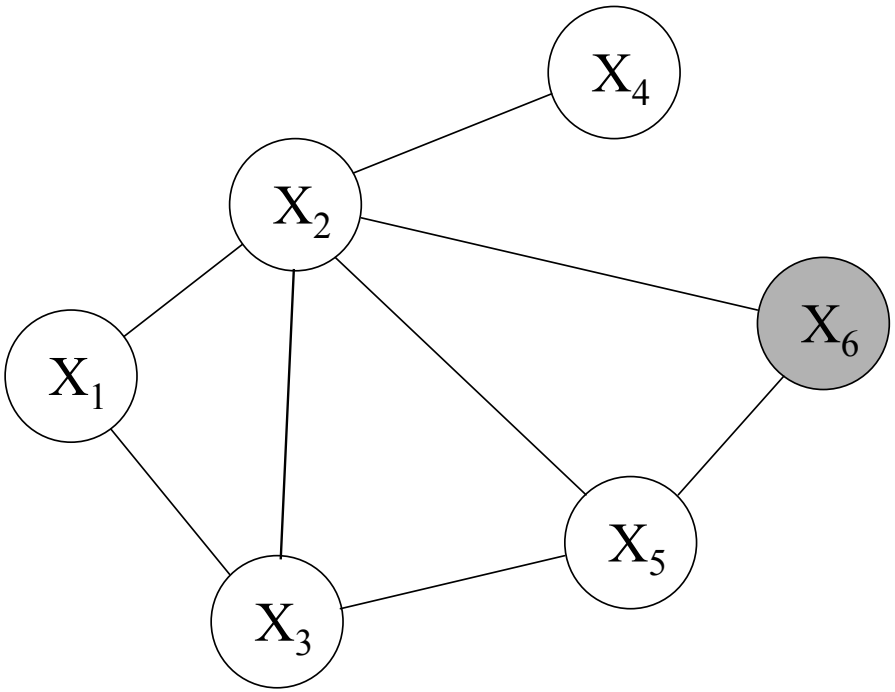
Triangulation Step

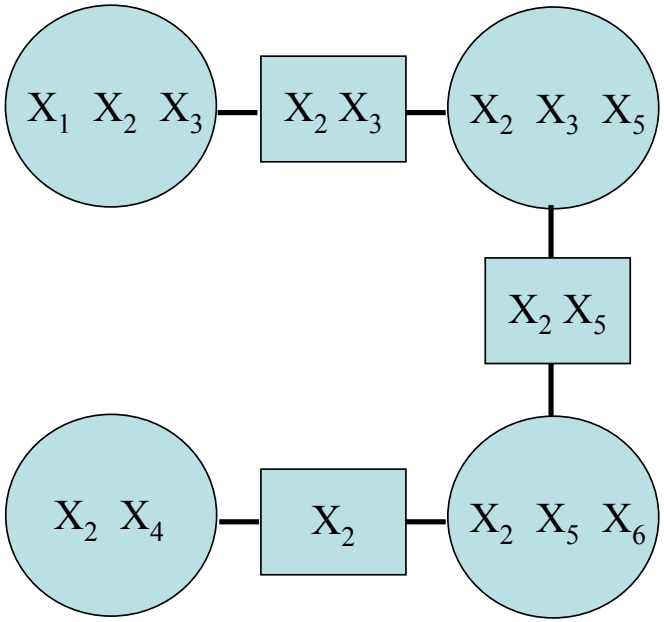
(1) *Triangulate* the graph (if it's not triangulated)



Junction Tree Construction

(2) Create a Junction Tree



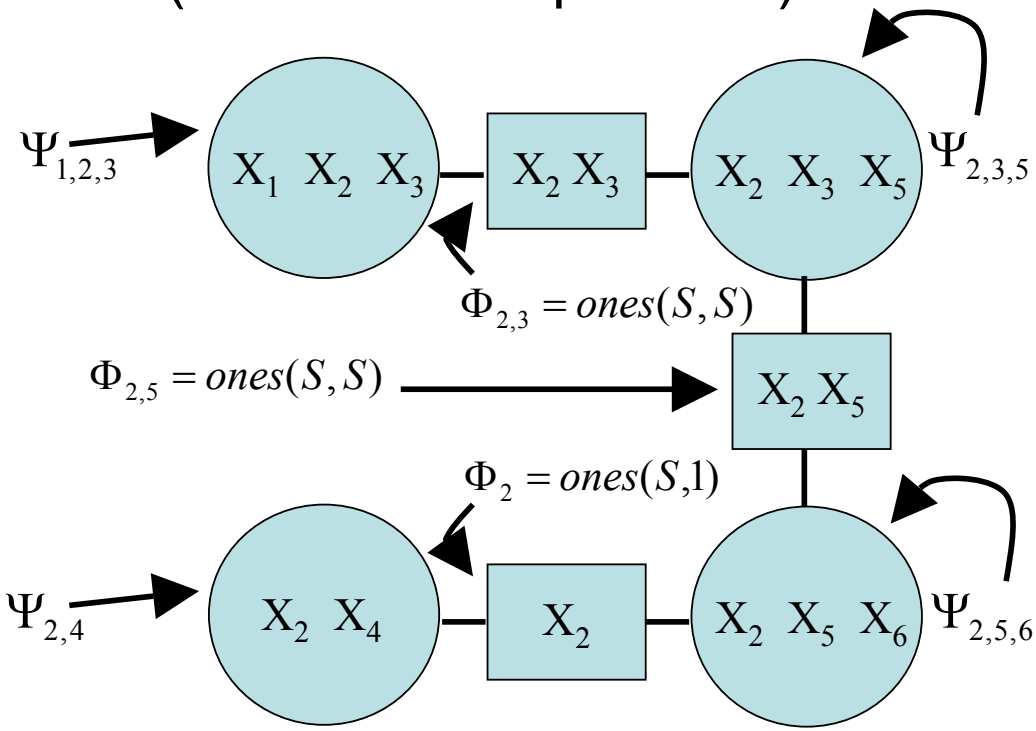


Initialization

(3) Initialize clique potentials (nodes and separators)

Ψ_c Directly introduced

Φ_s Initialized to 1



Propagation

(4) Message passing

$$\Psi^{**}_{1,2,3} = \frac{\Phi^{**}_{2,3}}{\Phi^{*}_{2,3}} \Psi_{1,2,3} \qquad \Phi^{**}_{2,3} = \sum_{x_5} \Psi^{**}_{2,3,5}$$

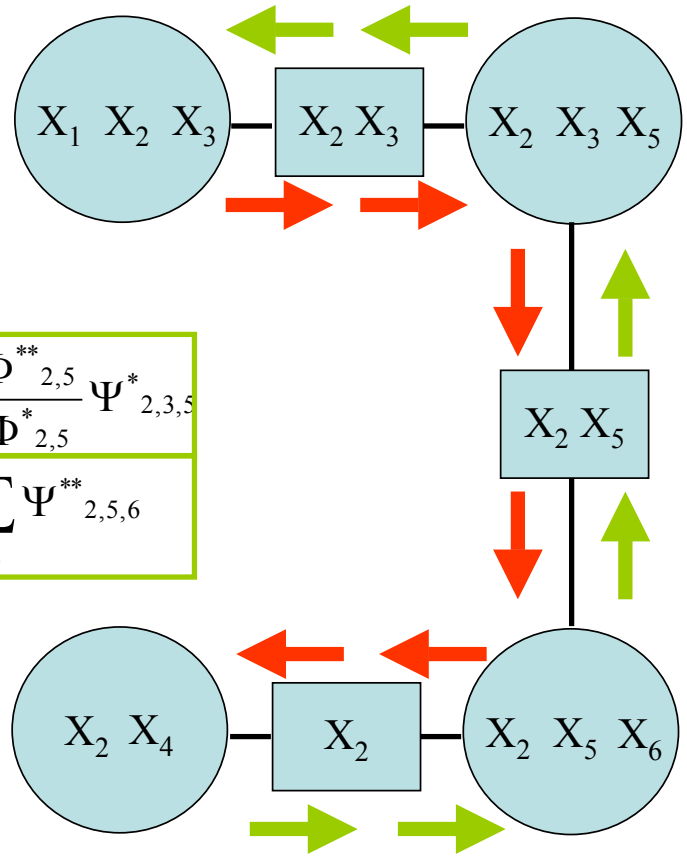
$$\Phi^{*}_{2,3} = \sum_{x_1} \Psi_{1,2,3} \qquad \Psi^{*}_{2,3,5} = \frac{\Phi^{*}_{2,3}}{\Phi_{2,3}} \Psi_{2,3,5}$$

$$\Phi^{*}_{2,5} = \sum_{x_3} \Psi^{*}_{2,3,5} \qquad \Psi^{**}_{2,3,5} = \frac{\Phi^{**}_{2,5}}{\Phi^{*}_{2,5}} \Psi^{*}_{2,3,5}$$

$$\Psi^{*}_{2,5,6} = \frac{\Phi^{*}_{2,5}}{\Phi_{2,5}} \Psi_{2,5,6} \qquad \Phi^{**}_{2,5} = \sum_{x_6} \Psi^{**}_{2,5,6}$$

$$\Psi^{*}_{2,4} = \frac{\Phi^{*}_{2}}{\Phi_{2}} \Psi_{2,4} \qquad \Phi^{*}_{2} = \sum_{x_5 x_6} \Psi^{*}_{2,5,6}$$

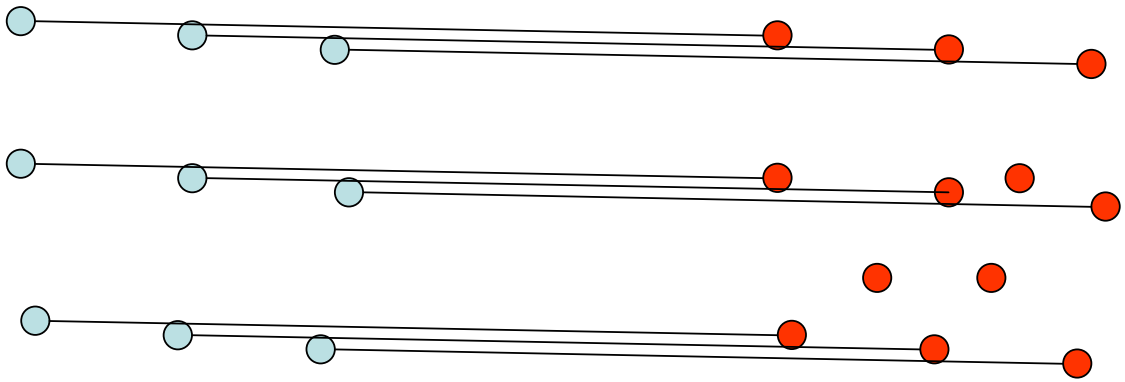
$$\Phi^{**}_{2} = \sum_{x_4} \Psi^{*}_{2,4} \qquad \Psi^{*}_{2,5,6} = \frac{\Phi^{**}_{2}}{\Phi^{*}_{2}} \Psi_{2,5,6}$$



Applications

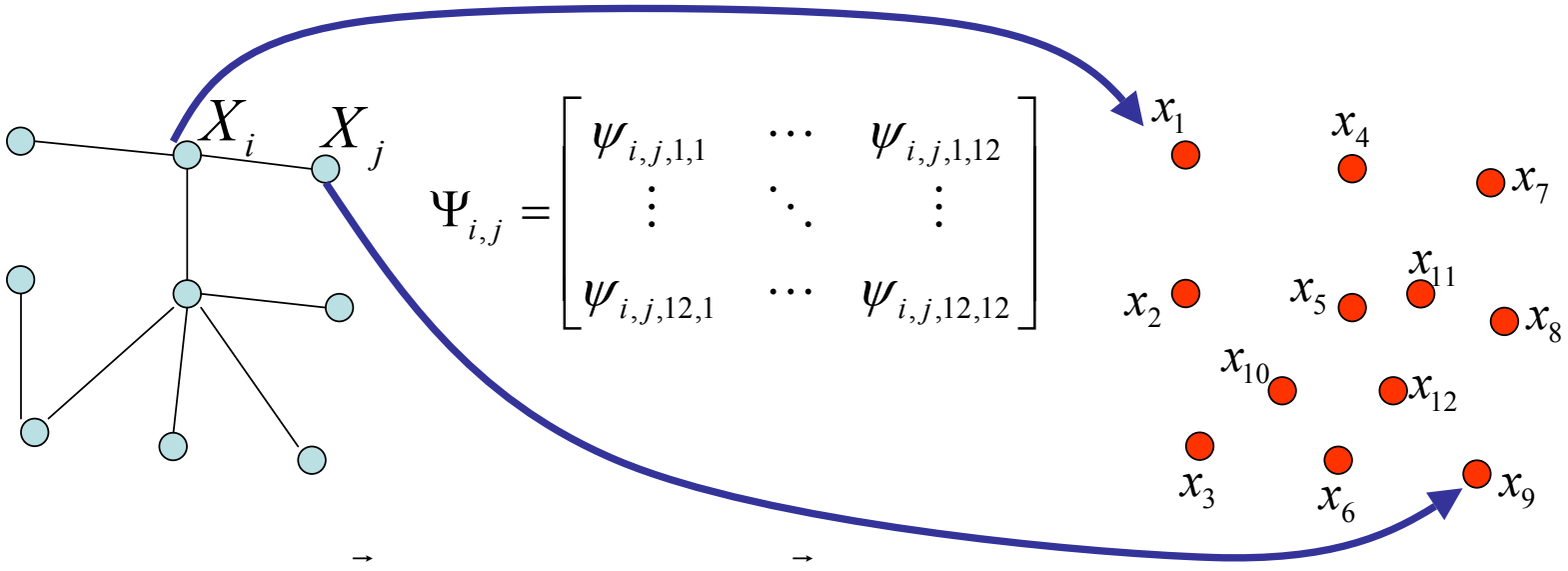
Matching

One can solve invariant matching problems using appropriate Graphical Models



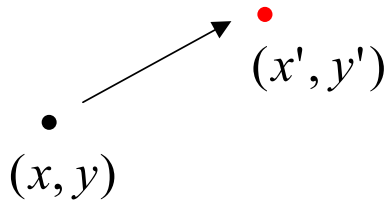
Applications

Rigid Matching



Applications

Affine Matching



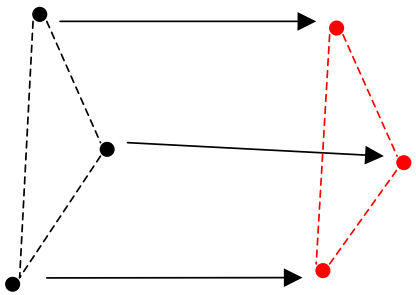
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = a_{11}x + a_{12}y + a_{13}$$
$$y' = a_{21}x + a_{22}y + a_{23}$$

2 equations,

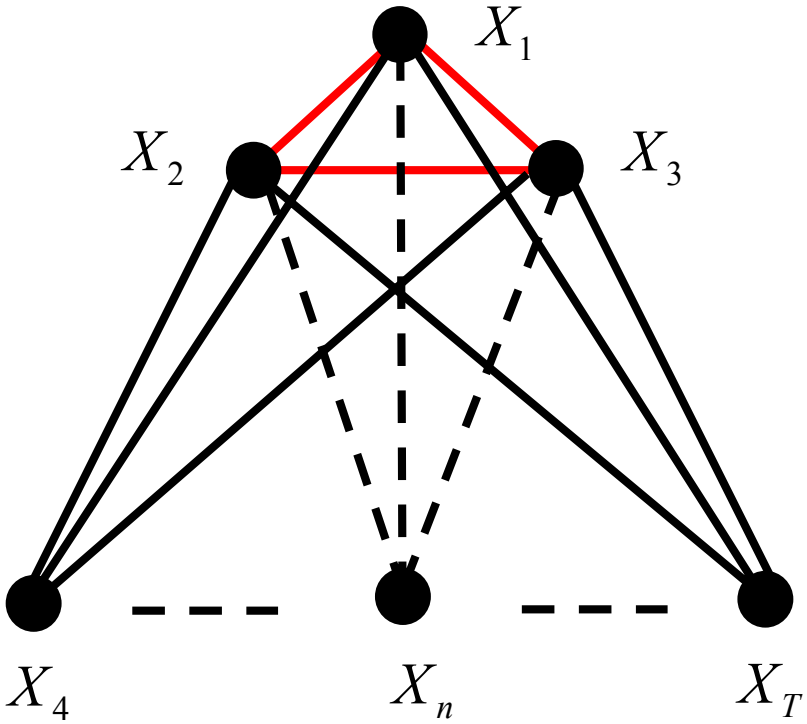
6 unknowns

Needs 3 correspondences to uniquely specify affine transformation

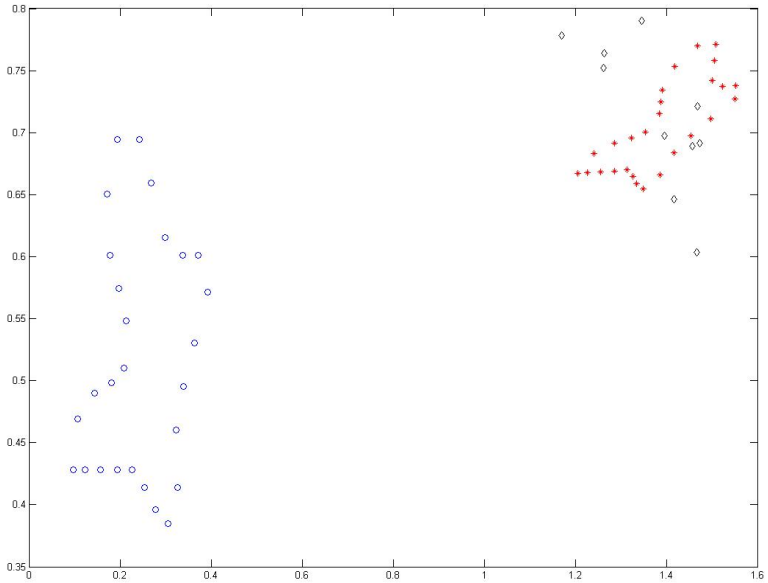


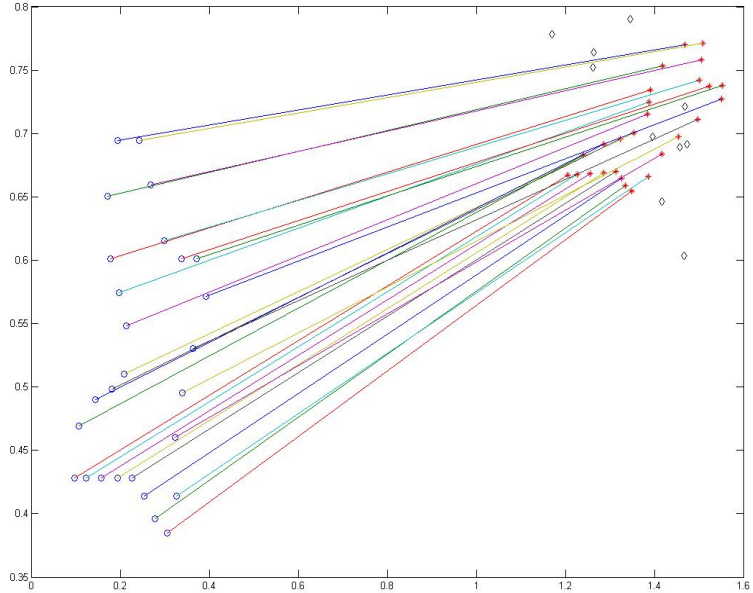
Applications

Affine matching

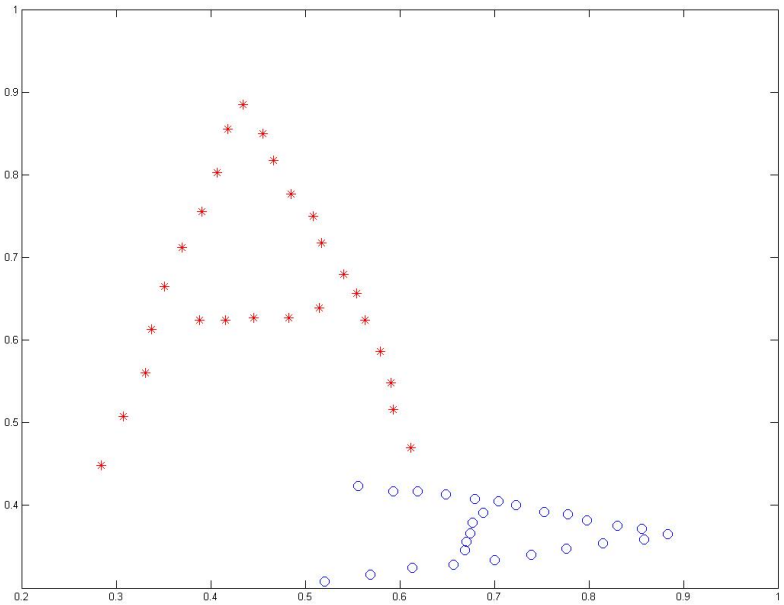


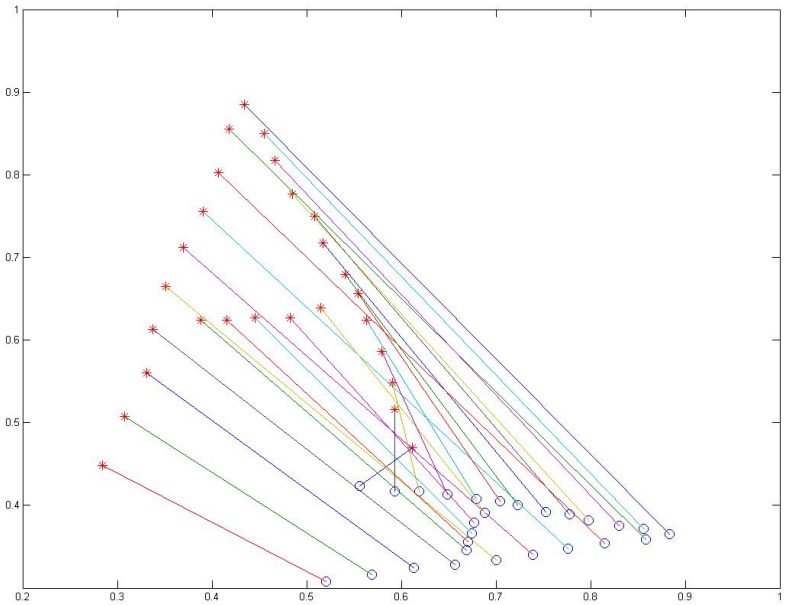
Applications





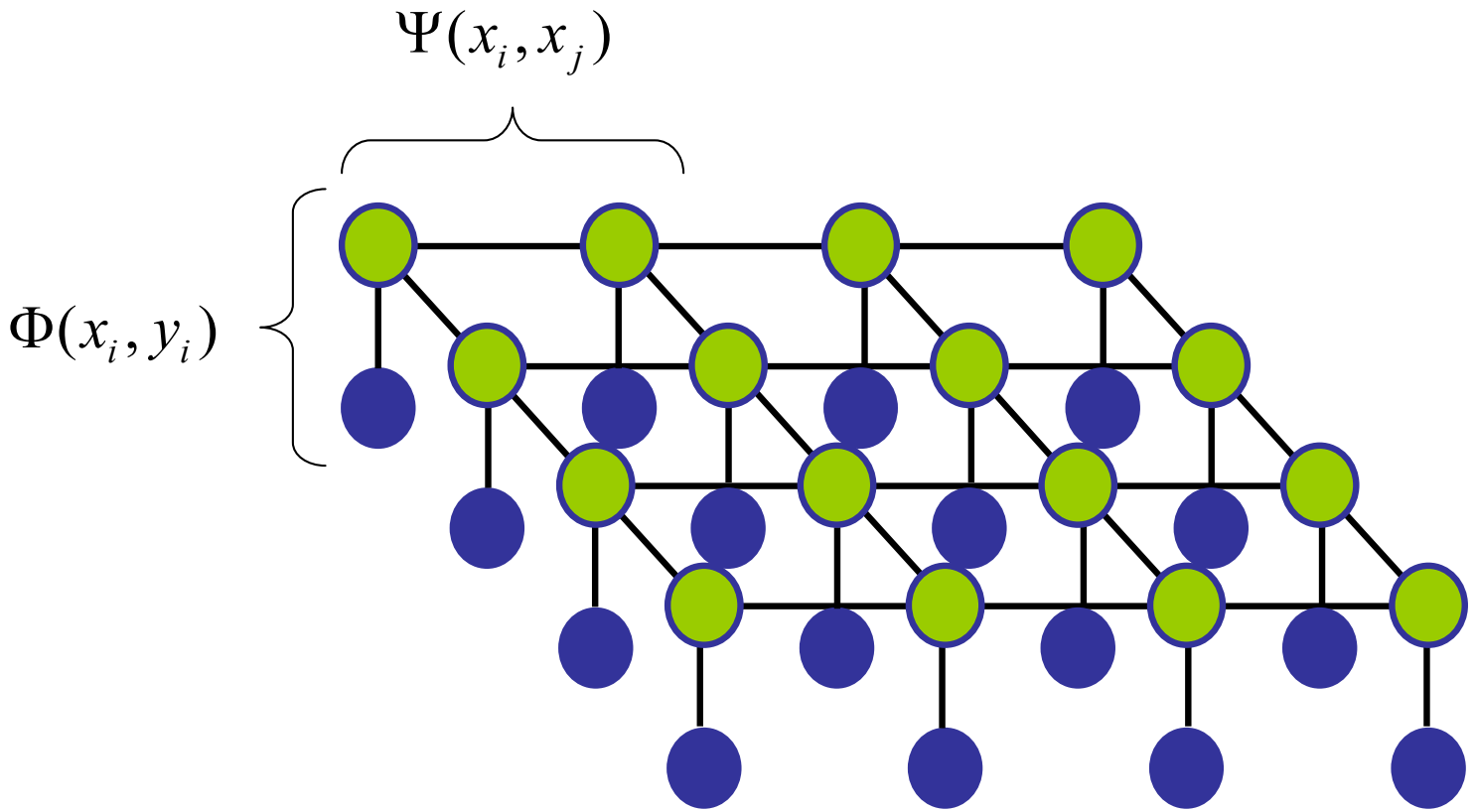
Applications





Applications

Denoising



Applications

Denoising

$$p(x,y) = \frac{1}{Z}\prod_{i,j}\Psi(x_i,x_j)\prod_{i}\Phi(x_i,y_i)$$

$$p(x \mid y) \propto \frac{1}{Z}\prod_{i,j}\Psi(x_i,x_j)\prod_{i}\Phi(x_i,y_i)$$

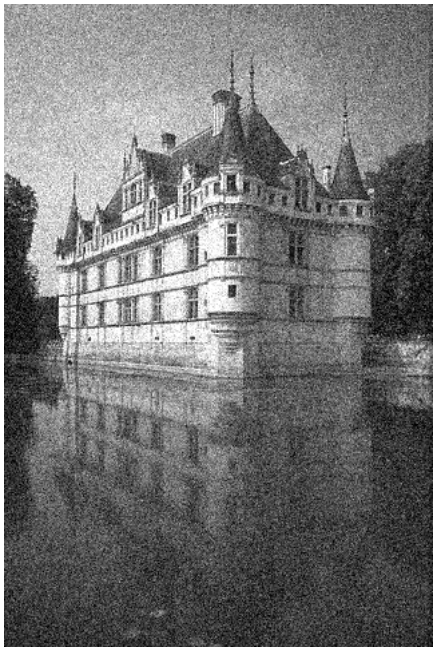
$$\Psi(x_i,x_j) = \exp(-(x_i - x_j)^2 / 2\sigma)$$

$$\Phi(x_i,y_i) = \exp(-(x_i - y_i)^2 / 2\sigma_N)$$

Applications

Denoising

Original



Noisy

Corrected