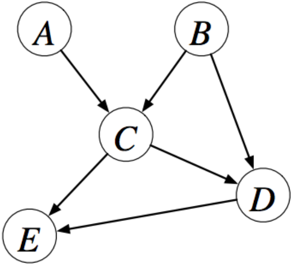
Lecture : Probabilistic Graphical Models I

Riashat Islam

Reasoning and Learning Lab
McGill University

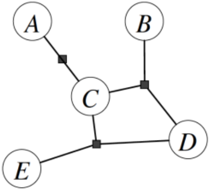
22nd October 2018

Representing Knowledge

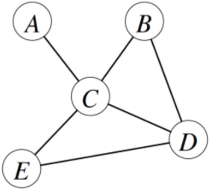


- Nodes correspond to random variables
- Edges represent statistical dependencies between variables

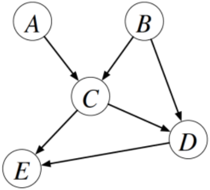
Graphical Models



factor graph



undirected graph



directed graph

- Open circles correspond to random variables
- Filled circles correspond to observed random variables
- Squared boxes show factor decompositions
- **Edges** represent statistical dependencies between variables
- Whole model represents a joint probability distribution

Motivation

- Graphs are an intuitive way of representing and visualising the relationships between many variables. (Examples: family trees, electric circuit diagrams, neural networks).
- A graph allows us to abstract out the conditional independencies between variables from the details of their parametric forms? We can answer questions like Is A dependent on B given that we know the value of C? just by looking at the graph.
- Graphical models allow us to define the 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 variables in the model

Graphical Models

GMs are graph based representations of various factorisation assumptions of distributions. These factorisations are typically equivalent to independence statements amongst (sets of) variables in the distribution.

Belief Network Each factor is a conditional distribution. Generative models, AI, statistics. Corresponds to a DAG.

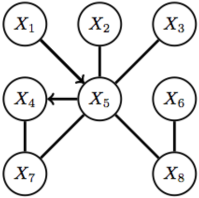
Markov Network Each factor corresponds to a potential (non negative function). Related to the strength of relationship between variables, but not directly related to dependence. Useful for collective phenomena such as image processing. Corresponds to an undirected graph.

Chain Graph A marriage of BNs and MNs. Contains both directed and undirected links.

Factor Graph A barebones representation of the factorisation of a distribution. Often used for efficient computation and deriving message passing algorithms.

The GM zoo There are many more kinds of GMs, each useful in its own right. We'll touch on some more when we consider inference.

Graphs



Definition

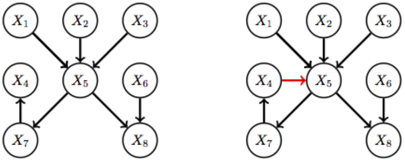
A graph consists of nodes (vertices) and undirected or directed links (edges) between nodes.

Path

A path from X_i to X_j is a sequence of connected nodes starting at X_i and ending at X_j.

Directed Graphs

All the edges are directed:



Directed Acyclic Graph

Directed Cyclic Graph

DAG

Directed Acyclic Graph: Graph in which by following the direction of the arrows a node will never be visited more than once.

Parents and Children:

X_i is a parent of X_j if there is a link from X_i to X_j. X_i is a child of X_j if there is a link from X_j to X_i.

Ancestors and Descendants:

The ancestors of a node X_i are the nodes with a directed path ending at X_i. The descendants of X_i are the nodes with a directed path beginning at X_i.

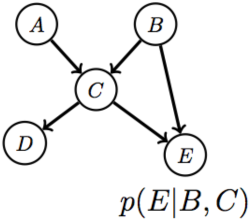
Directed Acyclic Graphical Models

A DAG Model / Bayesian network corresponds to a factorization of the joint probability distribution. They are also called Belief Networks.

A belief network is a directed acyclic graph in which each node has associated the conditional probability of the node given its parents.

The joint distribution is obtained by taking the product of the conditional probabilities

$$p(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|C)p(E|B, C) \tag{1}$$



Example

Sally's burglar **A**larm is sounding. Has she been **B**urgled, or was the alarm triggered by an **E**arthquake? She turns the car **R**adio on for news of earthquakes.

Choosing an ordering

Without loss of generality, we can write

$$\begin{aligned} p(A, R, E, B) &= p(A|R, E, B)p(R, E, B) \\ &= p(A|R, E, B)p(R|E, B)p(E, B) \\ &= p(A|R, E, B)p(R|E, B)p(E|B)p(B) \end{aligned}$$

Assumptions:

- The alarm is not directly influenced by any report on the radio, $p(A|R, E, B) = p(A|E, B)$
- The radio broadcast is not directly influenced by the burglar variable, $p(R|E, B) = p(R|E)$
- Burglaries don't directly 'cause' earthquakes, $p(E|B) = p(E)$

Therefore

$$p(A, R, E, B) = p(A|E, B)p(R|E)p(E)p(B)$$

Example : Inference

Initial Evidence: The alarm is sounding

$$p(B=1|A=1) = \frac{\sum_{E,R} p(B=1,E,A=1,R)}{\sum_{B,E,R} p(B,E,A=1,R)}$$
$$= \frac{\sum_{E,R} p(A=1|B=1,E)p(B=1)p(E)p(R|E)}{\sum_{B,E,R} p(A=1|B,E)p(B)p(E)p(R|E)} \approx 0.99$$

Additional Evidence: The radio broadcasts an earthquake warning:

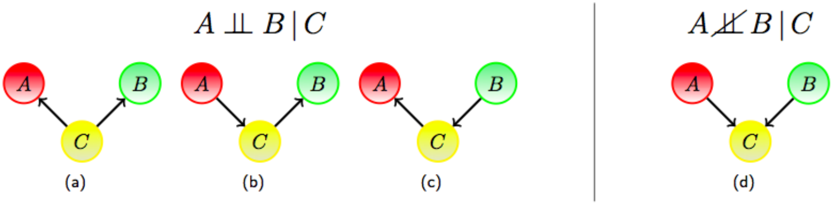
A similar calculation gives $p(B=1|A=1,R=1) \approx 0.01$.

Initially, because the alarm sounds, Sally thinks that she's been burgled. However, this probability drops dramatically when she hears that there has been an earthquake.

The earthquake 'explains away' to an extent the fact that the alarm is ringing.

Independence in Belief Networks

All belief networks with three nodes and two links:



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

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

(a) (b) (c) (d)

- In (a), (b) and (c), A, B are conditionally independent given C.

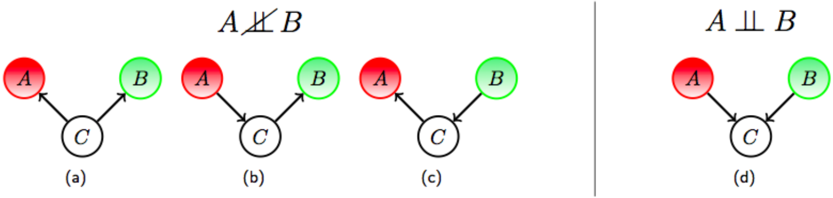
  (a) $p(A,B|C) = \frac{p(A,B,C)}{p(C)} = \frac{p(A|C)p(B|C)p(C)}{p(C)} = p(A|C)p(B|C)$

  (b) $p(A,B|C) = \frac{p(A)p(C|A)p(B|C)}{p(C)} = \frac{p(A,C)p(B|C)}{p(C)} = p(A|C)p(B|C)$

  (c) $p(A,B|C) = \frac{p(A|C)p(C|B)p(B)}{p(C)} = \frac{p(A|C)p(B,C)}{p(C)} = p(A|C)p(B|C)$

- In (d) the variables A, B are conditionally dependent given C, $p(A,B|C) \propto p(C|A,B)p(A)p(B)$.

Independence in Belief Networks



$A \not\perp\!\!\!\perp B$

$A \perp\!\!\!\perp B$

(a) (b) (c) (d)

- In (a), (b) and (c), the variables A, B are marginally dependent.

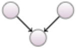
- In (d) the variables A, B are marginally independent.

$$p(A, B) = \sum_C p(A, B, C) = \sum_C p(A)p(B)p(C|A, B) = p(A)p(B)$$

General Rule of Independence in Belief Networks

Given three sets of nodes $\mathcal{X}, \mathcal{Y}, \mathcal{C}$, if all paths from any element of $\mathcal{X}$ to any element of $\mathcal{Y}$ are blocked by $\mathcal{C}$, then $\mathcal{X}$ and $\mathcal{Y}$ are conditionally independent given $\mathcal{C}$.

A path $\mathcal{P}$ is blocked by $\mathcal{C}$ if at least one of the following conditions is satisfied:

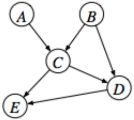
1. there is a collider  in the path $\mathcal{P}$ such that neither the collider nor any of its descendants is in the conditioning set $\mathcal{C}$.
2. there is a non-collider in the path $\mathcal{P}$ that is in the conditioning set $\mathcal{C}$.

---

d-connected/separated

We use the phrase 'd-connected' if there is a path from $\mathcal{X}$ to $\mathcal{Y}$ in the 'connection' graph – otherwise the variable sets are 'd-separated'. Note that d-separation implies that $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$, but d-connection does not necessarily imply conditional dependence.

Inference in Directed Graphical Models



Consider the following graph: which represents:

$$p(A,B,C,D,E) = p(A)p(B)p(C|A,B)p(D|B,C)p(E|C,D)$$

Inference: evaluate the probability distribution over some set of variables, given the values of another set of variables.

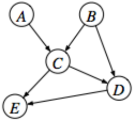
For example, how can we compute $p(A|C=c)$? Assume each variable is binary.

Naive method:

$$\begin{aligned} p(A, C=c) &= \sum_{B,D,E} p(A, B, C=c, D, E) && \text{[16 terms]} \\ p(C=c) &= \sum_{A} p(A, C=c) && \text{[2 terms]} \\ p(A|C=c) &= \frac{p(A, C=c)}{p(C=c)} && \text{[2 terms]} \end{aligned}$$

Total: 16+2+2 = 20 terms have to be computed and summed

Inference in Directed Graphical Models

Consider the following graph: 

which represents:

$$p(A,B,C,D,E) = p(A)p(B)p(C|A,B)p(D|B,C)p(E|C,D)$$

Computing $p(A|C=c)$.

More efficient method:

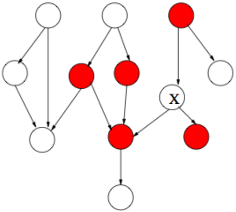
$$\begin{aligned}
p(A, C=c) &= \sum_{B,D,E} p(A)p(B)p(C=c|A,B)p(D|B,C=c)p(E|C=c,D) \\
&= \sum_{B} p(A)p(B)p(C=c|A,B) \sum_{D} p(D|B,C=c) \sum_{E} p(E|C=c,D) \\
&= \sum_{B} p(A)p(B)p(C=c|A,B) \qquad \text{[4 terms]}
\end{aligned}$$

Total: 4+2+2 = 8 terms

Belief propagation methods use the conditional independence relationships in a graph to do efficient inference (for singly connected graphs, exponential gains in efficiency!).

Markov Blanket of a Node

- Suppose we want the smallest set of nodes U such that X is independent of all other nodes in the network given U. What should U be?
- Clearly, at least X's parents and children should be in U
- But this is not enough if there are v-structures; U will also have to include X's "spouses" - i.e. the other parents of X's children
- The set U consisting of X's parents, children and other parents of its children is called the **Markov blanket** of X.



Undirected Graphical Models

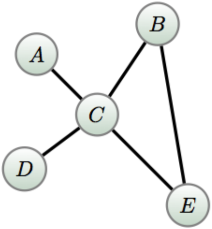
- So far we have used directed graphs as the underlying structure of a Bayes net
- Why not use **undirected graphs** as well?
- Example : variables might not be in a causality relation, but they can still be correlated, like the pixels in a neighbourhood in an image
- An undirected graph over a set of random variables $X_1, X_2, ...X_n$ is called a **undirected graphical model** or **Markov Random Field (MRF)** or **Markov Network**

Markov Network

Clique: Fully connected subset of nodes.

Maximal Clique: Clique which is not a subset of a larger clique.

A Markov Network is an undirected graph in which there is a potential (non-negative function) ψ defined on each maximal clique.



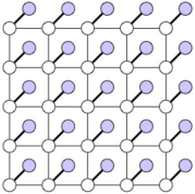
The joint distribution is proportional to the product of all clique potentials.

$$p(A,B,C,D,E) = \frac{1}{Z}\psi(A,C)\psi(C,D)\psi(B,C,E)$$

$$Z = \sum_{A,B,C,D,E} \psi(A,C)\psi(C,D)\psi(B,C,E)$$

Application of Markov Network

Problem: We want to recover a binary image from the observation of a corrupted version of it.



$X = \{X_i, i = 1, \ldots, D\}$ $X_i \in \{-1, 1\}$: clean pixel

$Y = \{Y_i, i = 1, \ldots, D\}$ $Y_i \in \{-1, 1\}$: corrupted pixel

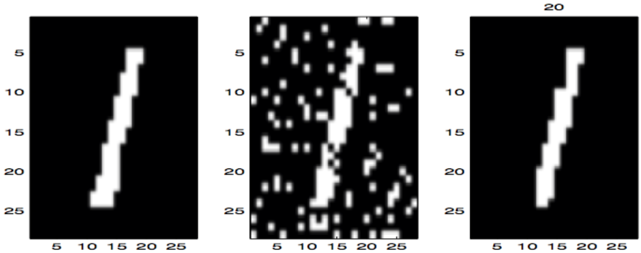
$\phi(Y_i, X_i) = e^{\gamma X_i Y_i}$ encourage Y_i and X_i to be similar

$\psi(X_i, X_j) = e^{\beta X_i X_j}$ encourage the image to be smooth

$$p(X, Y) \propto \left[\prod_{i=1}^{D} \phi(Y_i, X_i)\right] \left[\prod_{i \sim j} \psi(X_i, X_j)\right]$$

Finding the most likely X given Y is not easy (since the graph is not singly-connected), but approximate algorithms often work well.

Application of Markov Network

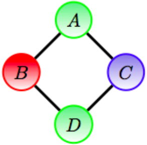


left Original clean image

middle Observed (corrupted) image

right Most likely clean image $\operatorname*{argmax}_{X} p(X|Y)$

Independence in Markov Networks

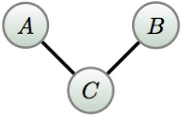


$$B \perp\!\!\!\perp C \mid A, D?$$

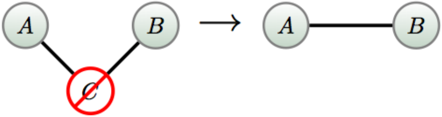
$$p(B|A,D,C) = p(B|A,D)?$$

$$\begin{aligned}
p(B|A,D,C) &= \frac{p(A,B,C,D)}{p(A,C,D)} \\
&= \frac{p(A,B,C,D)}{\sum_B p(A,B,C,D)} \\
&= \frac{\psi(A,B)\cancel{\psi(A,C)}\psi(B,D)\cancel{\psi(C,D)}}{\sum_B \psi(A,B)\cancel{\psi(A,C)}\psi(B,D)\cancel{\psi(C,D)}} \\
&= p(B|A,D)
\end{aligned}$$

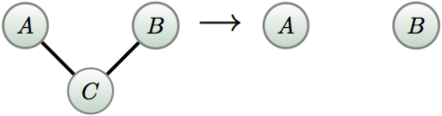
Properties of Markov Networks



$$p(A,B,C) = \phi_{AC}(A,C)\phi_{BC}(B,C)/Z$$

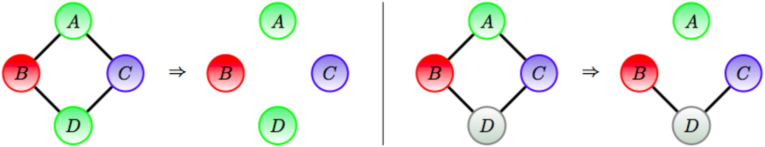


Marginalising over C makes A and B (graphically) dependent. In general $p(A,B) \neq p(A)p(B)$.



Conditioning on C makes A and B independent: $p(A,B|C) = p(A|C)p(B|C)$.

General Rule of Independence in Markov Networks

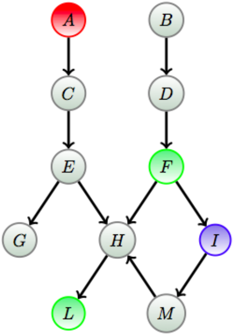


- Remove all links neighbouring the variables in the conditioning set $\mathcal{Z}$.
- If there is no path from any member of $\mathcal{X}$ to any member of $\mathcal{Y}$, then $\mathcal{X}$ and $\mathcal{Y}$ are conditionally independent given $\mathcal{Z}$.

Alternative Rule for Independence in Belief Networks

$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

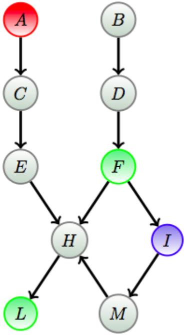
- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.
- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.
- Separation: Remove all links from $\mathcal{Z}$.
- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$.



Alternative Rule for Independence in Belief Networks

$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

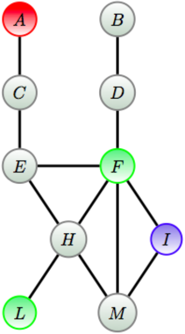
- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.
- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.
- Separation: Remove all links from $\mathcal{Z}$.
- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$.



Alternative Rule for Independence in Belief Networks

$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.
- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.
- Separation: Remove all links from $\mathcal{Z}$.
- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$.



Alternative Rule for Independence in Belief Networks

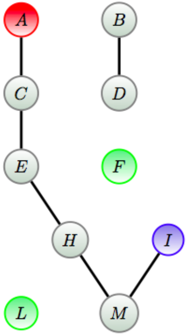
$\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$?

- Ancestral Graph: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.

- Moralisation: Add a line between any two nodes which have a common child. Remove arrowheads.

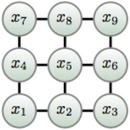
- Separation: Remove all links from $\mathcal{Z}$.

- Independence: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$.



$A \top\!\!\!\top I | F, L$

The Ising Model



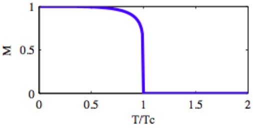
$x_i \in \{+1, -1\}$:

$$p(x_1, \ldots, x_9) = \frac{1}{Z} \prod_{i \sim j} \phi_{ij}(x_i, x_j)$$

$$\phi_{ij}(x_i, x_j) = e^{-\frac{1}{2T}(x_i - x_j)^2}$$

$i \sim j$ denotes the set of indices where i and j are neighbours in the graph. The potential encourages neighbours to be in the same state.

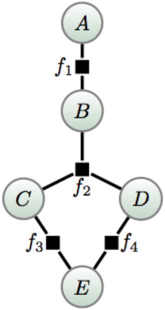
Spontaneous global behaviour



$M = |\sum_{i=1}^{N} x_i|/N$. As the temperature T decreases towards the critical temperature T_c a phase transition occurs in which a large fraction of the variables become aligned in the same state. Even though we only 'softly' encourage neighbours to be in the same state, for a low but finite T, the variables are all in the same state. Paradigm for 'emergent behaviour'.

Factor Graphs

A square node represents a factor (non negative function) of its neighbouring variables.

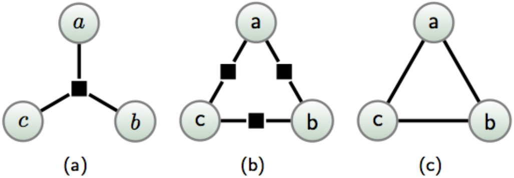


The joint function is the product of all factors:

$$f(A,B,C,D,E) = f_1(A,B)f_2(B,C,D)f_3(C,E)f_4(D,E)$$

Factor graphs are useful for performing efficient computations (not just for probability).

Factor Graphs vs Markov Networks



a $\phi(a, b, c)$

b $\phi(a, b)\phi(b, c)\phi(c, a)$

c $\phi(a, b, c)$

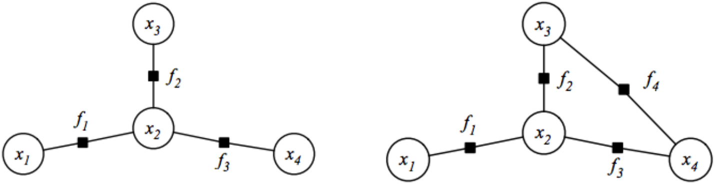
- Both (a) and (b) have the same Markov network (c).
- Whilst (b) contains the same (lack of) independence statements as (a), it expresses more constraints on the form of the potential.

Factor Graph Propagation

Algorithmically and implementationally, it's often easier to convert directed and undirected graphs into factor graphs, and run *factor graph propagation*.

$$\begin{aligned} p(\mathbf{x}) &= p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_2) \\ &\equiv f_1(x_1, x_2)f_2(x_2, x_3)f_3(x_2, x_4) \end{aligned}$$

Singly connected vs Multiply connected factor graphs:



Propagation in Factor Graphs

Let $\mathrm{n}(x)$ denote the set of factor nodes that are neighbors of x.
Let $\mathrm{n}(f)$ denote the set of variable nodes that are neighbors of f.

We can compute probabilities in a factor graph by propagating messages from variable nodes to factor nodes and viceversa.

message from variable x to factor f:

$$\mu_{x \to f}(x) = \prod_{h \in \mathrm{n}(x) \setminus \{f\}} \mu_{h \to x}(x)$$

message from factor f to variable x:

$$\mu_{f \to x}(x) = \sum_{\mathbf{x} \setminus x} \left(f(\mathbf{x}) \prod_{y \in \mathrm{n}(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

where $\mathbf{x}$ are the variables that factor f depends on, and $\sum_{\mathbf{x} \setminus x}$ is a sum over all variables neighboring factor f except x.

Propagation in Factor Graphs

$\mathrm{n}(x)$ denotes the set of factor nodes that are neighbors of x.
$\mathrm{n}(f)$ denotes the set of variable nodes that are neighbors of f.

message from variable x to factor f:

$$\mu_{x \to f}(x) = \prod_{h \in \mathrm{n}(x) \setminus \{f\}} \mu_{h \to x}(x)$$

message from factor f to variable x:

$$\mu_{f \to x}(x) = \sum_{\mathbf{x} \setminus x} \left(f(\mathbf{x}) \prod_{y \in \mathrm{n}(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

If a variable has only one factor as a neighbor, it can initiate message propagation.

Once a variable has received all messages from its neighboring factor nodes, one can compute the probability of that variable by multiplying all the messages and renormalising:

$$p(x) \propto \prod_{h \in \mathrm{n}(x)} \mu_{h \to x}(x)$$

Summary

We have introduced...

- Directed and Undirected Graphical Models
- Independence in Graphs

Next : Exact Inference in Graphical Models

- Variable Elimination
- Belief Propagation