**CS 4850: Mathematical Foundations for the Information Age** April 30

Lecture 39

*Lecturer: Prof. Hopcroft* *Scribe: Jiaqi Zhai (jz392)*

# 1 Belief Propagation

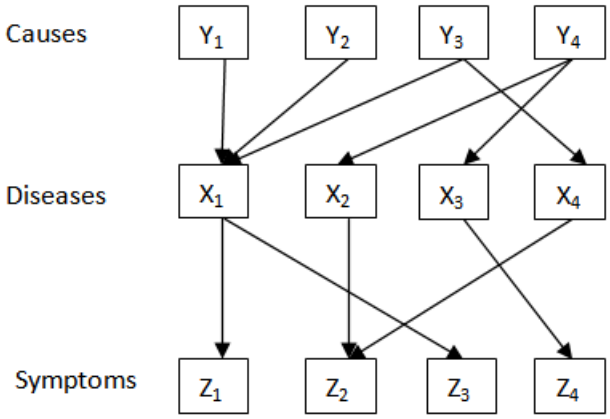
## 1.1 Terminology

**Graphical Model** A probabilistic model represented as a graph. We associate random variables with the vertices, and use edges to express relationships between variables.

**Types** Directed and undirected.

**Directed (acyclic) graph** This kind of graphical model is also called Bayesian network or belief network.

**A doctor's problem** Consider the following graph:




Each variable in the graph can take on one of the two values, true and false.
We can express the variables on the first line, causes, as unconditional probabilities, and the other variables, diseases and symptoms, as conditional probabilities.
We might want to know:

- What is the marginal probability $P(X_1) = \sum_{y1,y2,\ldots} P(\cdots)$ ?

- Given information about $y$ and $z$, what is the most likely disease? i.e., assign 0 or 1 to each $X_i$ such that the probability is maximized. *This can be solved with maximum a posteriori (MAP).*

We will not go into details here.

**Undirected graph** Also known as pairwise Markov Random field. Here, $P(x_1, x_2, \cdots, x_n)$ can be written in a special form:

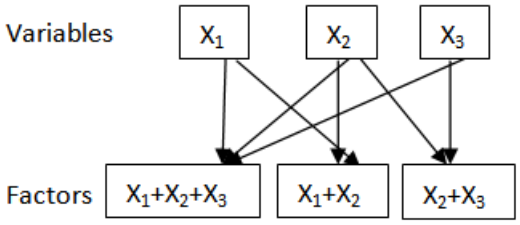
$$P(x_1, x_2, \cdots, x_n) = \frac{1}{Z} \prod (i, j) \in E \Psi(x_i, x_j)$$

where $E$ is the set of edges and $Z$ is the normalizing factor.
If we don't require the Markov random field to correspond to a graph, we can obtain a more general form of the above equation, in which the function $\Psi$ can take more than two variables as its parameters.

**Factor graph** Example:

$$f(x_1, x_2, x_3) = \frac{1}{Z}(x_1 + x_2 + x_3)(x_1 + x_2)(x_1 + x_3)(x_2 + x_3)$$

We can construct a bipartite graph, in which one side are the variables and the other side are the factors. Add an edge between nodes $x$ and $y$ if variable $x$ is included in the factor $y$.




**Special case: tree**

Consider the special case that the factor graph is a tree. It would be easy to calculate properties such as marginal probabilities, etc.
Take a simple formula:

$$f = x_1(x_1 + x_2 + x_3)(x_3 + x_4 + x_5)x_4x_5$$

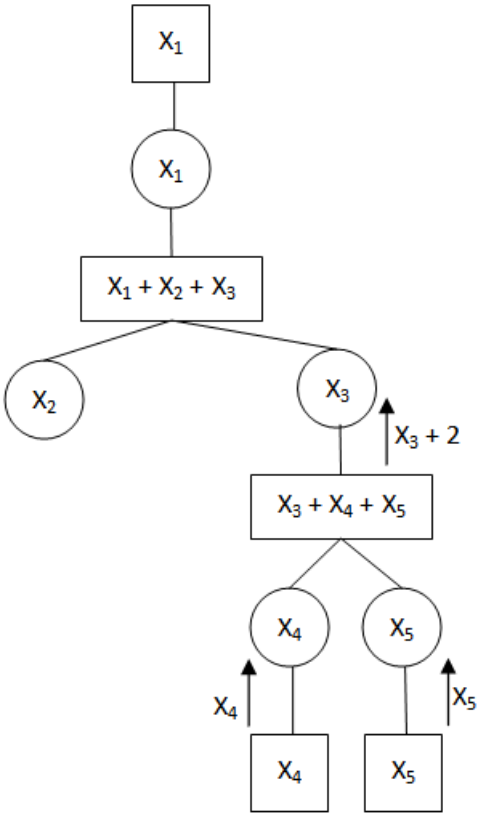
Note that

$$f(x_1) = \sum_{x_2,x_3,x_4,x_5} f = \sum_{x2,x_3} x_1(x_1 + x_2 + x_3) \sum_{x_4,x_5} (x_3 + x_4 + x_5)x_4x_5$$

for the term $\sum_{x \neq x_5}(x_3 + x_4 + x_5)x_4x_5$, we could sum over $x_4$ and $x_5$ to obtain $x_3 + 2$ and so

$$f(x_1) = \sum_{x2,x_3} x_1(x_1 + x_2 + x_3)(x_3 + 2)$$

This is the general idea used in message passing. An illustration of the tree and the messages passed between nodes is as follows:




We could apply this to compute the marginal probabilities for all $x_i$s and reduce the time complexity from $O(n^2)$ to $O(n)$.

Rules for message passing:

(a) At leaf nodes:

(i) If the node is a variable node, sum over all possible values, and send the sum to its parent.

For example, the variable node $(x_2)$ sends message 1 to the factor node $[x_1 + x_2 + x_3]$.

(ii) If the node is a factor node, then it can have only one variable. We send the one-variable function up directly.

For example, the factor node $[x_4]$ sends message $x_4$ to the variable node $(x_4)$.

(b) At interior nodes:

Suppose $y$ is a neighbor of $x$. We cannot pass the message from $x$ to $y$ until $x$ has received a message from every neighbor of $x$ except $y$.

(i) If $x$ is a variable node, compute the product of all incoming messages except the one from $y$, and sum this product over all variables except the variable at the node $x$.

For example, at the variable node $(x_5)$, we send $x_5$ up to the factor node $[x_3 + x_4 + x_5]$.

(ii) If $x$ is a factor node, compute the product of $x's$ factor function and all incoming messages except the one from $y$, then sum the resulting function over all variables except the variable at the node $y$.

For example, at the factor node $[x_3+x_4+x_5]$, the product is $(x_3+x_4+x_5)x_4x_5$, and summing over $x_4$ and $x_5$, we obtain $x_3 + 2$. This is the message to be sent up to the variable node $(x_3)$.

We could also modify the above procedure to solve maximization problems, etc.

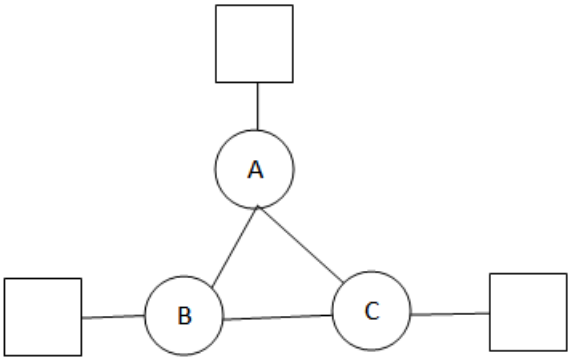
**General case**

The message passing algorithm seems to be a complicated way of computing marginal probabilities, but we can extend it onto general graphs. Obviously, we have to iteratively update the messages (and normalize the values accordingly). Two questions arise:

(a) Does this procedure converge?

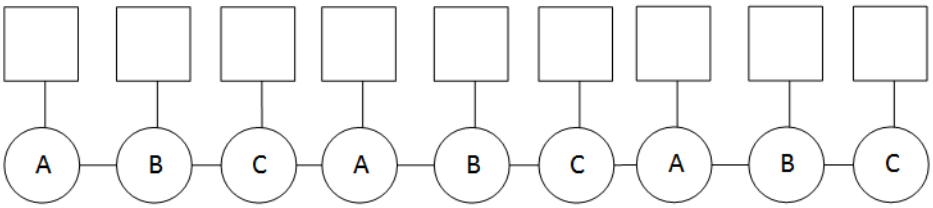
(b) What does it converge to?

People have shown that for certain problems, this procedure is guaranteed to converge, e.g., graphs with only one loop.




Proof Idea:

Unroll the graph (by replicating the vertices). We have a graph like the following (note that $A, B, C$ can be repeated infinitely many times):




Performing belief propagation in the loopy network is then equivalent to performing belief propagation in the unwrapped network. Moreover, since the new unwrapped network is a chain, belief propagation is guaranteed to give us the correct result.