1 Overview

In this lecture, we study the Belief propagation algorithm (BP) and the Max Product algorithm (MP). Last lecture reminds us of that in MRF, computing the marginal probabilities of random variables and Maximum A Posteriori (MAP) assignment is important. The Belief Propagation algorithm is a popular algorithm that is used to compute marginal probability of random variables. Max Product algorithm is a similar algorithm for computing MAP assignment. Today, we learn how Belief Propagation algorithm and Max Product algorithm operate.

2 From the Last Lecture

In the last lecture, we studied some examples of MRF such as Image denoising and Maximum Weight Independent Set (MWIS). In those examples, a probability distribution $X$ is a pair-wise MRF which is defined on a graph $G = (V, E)$ whose maximum clique size is 2. $X$ is expressed as following:

$$ P[X = x] = \frac{1}{Z} \prod_{i \in V} \psi_i(x_i) \prod_{(ij) \in E} \psi_{ij}(x_i, x_j) $$

where $X$ denotes the set of random variables and $x$ denotes the configuration of $X$. $\psi_i(x_i)$ is the potential function for a clique $i$ whose size is 1, a vertex. $\psi_{ij}(x_i, x_j)$ is the potential function for a clique $i, j$ whose size is 2, vertices which is connected by edge $ij$. And $Z$ is called the partition function which is used for a normalization.

Another example of MRF is Ising model that is a mathematical model to understand the structure of spinglass material in statistical physics. A discrete collection of random variables is called spins, which can take on the value 1 or -1. The spins interact with other spins and energy function $E(x)$ represents the relationship between them. The probability of spins in MRF is defined as following:

$$ P[(X = x)] \propto \exp(- \sum_{(u,v) \in E} J_{uv} x_u x_v) = \exp(E(x)) $$

where $x_v \in \{ -1, 1 \}$ and $X$ denotes the set of random variables about spins and $x$ denotes the configuration of values of spins. $J_{uv}$ is half the difference in energy between a pair of spin $u$ and $v$. $x_u$ and $x_v$ is the spin values for spin $u$ and $v$. Above expression is also a pair-wise MRF since only pairs of spins are considered. Finally, we can find the most probable assignment which is corresponding to
the assignment with minimum energy by finding MAP assignment. Figure 1 shows an example of *Ising model*.

![Ising model example](image)

Figure 1. An example of *Ising model*.

### 3 Belief Propagation algorithm

As mentioned in last lecture, computing MAP assignment and partition function (actually, the marginal probability) are very important in MRF. Many algorithms are introduced and the following properties are required for these algorithms:

- The algorithm has fast running time.
- The output of the algorithm is exact or close to the exact solution.
- (sometimes) The algorithm can be computed in a distributed way.

**Belief propagation algorithm** is an iterative algorithm which computes the marginal probability of each node and achieves these properties if a graph is a tree. It also can be extended to any form of graphs with approximate solution. And it is a time-slotted algorithm. Each node takes the following procedure at every time step until the values of beliefs are converged.

- Send messages to its neighbors.
- Calculate a belief based on messages passed from its neighbors.

Before watching the details of a message and a belief, the following example will give us the basic idea of *belief propagation algorithm*. As shown in Figure 2, a graph $G$ is a tree and a node $i$ is current root node. Let $Z_i(x_i)$, $Z_j(x_j)$ and $Z_k(x_k)$ be the marginal probability of MRF on the subtree rooted at $i$, $j$ and $k$ where $x_i \in \{0, 1\}$. Then, the marginal probability $Z_i(0)$ can be represented like the following:

$$Z_i(0) = \psi_i(0) \cdot \{\psi_{ij}(0, 0)Z_j(0) + \psi_{ij}(0, 1)Z_j(1)\} \cdot \{\psi_{ik}(0, 0)Z_k(0) + \psi_{ik}(0, 1)Z_k(1)\}$$
Since $Z_j(x_j)$ and $Z_k(x_k)$ are also the marginal probability of random variables which is appeared at each subtree, above expression is representing the exact marginal probability of $Z_i(0)$. In general, above expression can be represented as follows:

$$Z_i(x_i) = \psi_i(x_i) \prod_{j \in N(i)} \sum_{x_j} \psi_{ij}(x_i, x_j) Z_j(x_j)$$

where $N(i)$ denotes the set of neighbors of a node $i$.

![Figure 2. A graph $G$ for above example.](image)

Actually belief propagation algorithm is based on this basic idea. If we have the marginal probability of MRF at subtrees, the marginal probability of the parent node of them can be calculated by using only local information. Since we assumed that the graph is a tree and any node at a tree can decompose it into 2 or more subtrees and take a role as a root, above expression makes sense. This example shows us that what kind of values should be passed among nodes and how the marginal probability is calculated.

The details of messages and beliefs are the following:

- $m^t_{j \rightarrow i}(x_i)$ is a message that a node $j$ sends to a node $i$ about the value $x_i$ of a random variable $X_i$ at a time step $t$. And each node generates next messages to each neighbor for a time step $t+1$ as the following:

$$m^{t+1}_{j \rightarrow i}(x_i) = \sum_{x_j} \psi_{ji}(x_j, x_i) \psi_j(x_j) \prod_{k \in N(j) \setminus i} m^t_{j \rightarrow i}(x_j)$$

Initially ($t = 0$), since there is no messages from neighbors, each node sends first message by using its own potential.

- $b^t_i(x_i)$ is a belief at a node $i$ and actually it represents the marginal probability that a random variable $X_i$ has $x_i$ at a time step $t$. It is based on messages from its neighbors at a time step $t$, so
a belief can be calculated after messages passed.

\[ b^t_i(x_i) = \psi_i(x_i) \prod_{j \in N(i)} m^t_{j \rightarrow i}(x_i) \]

Above tasks of belief propagation algorithm at one time step is shown at Figure 3.

Figure 3. Left : the procedure of passing a message. Right : the procedure of calculating a belief.

This procedure is repeated until beliefs of all nodes are converged. It is clear that belief propagation algorithm needs at least \( d \) time steps for the exact marginal probability at every node, since messages are passed to only neighbors at each time step where \( d \) denotes the diameter of \( G \). It can be proved by mathematical induction on the number of vertices of \( G \) using the definition of a message and a belief.

In short, when a graph \( G \) is a tree and message passing is done through more than \( d \) iterations, we can get the exact marginal probability of all random variables. Even if we do not have global structure information or a diameter \( d \) of a graph \( G \), belief propagation algorithm can calculate exact or close to the exact marginal probability with only local information.

4 Max Product algorithm

Max product algorithm is a very similar algorithm with belief propagation algorithm. While belief propagation algorithm gives us the marginal probability, max product algorithm is to find MAP assignment. MAP assignment is the maximum probability configuration of the random variables on the entire graph.
Max product algorithm has the same procedure with belief propagation algorithm, but details of calculating messages is slightly different. The following tasks are repeated until beliefs of all nodes are converged like belief propagation algorithm.

- $m^t_{j \rightarrow i}(x_i)$ is a message from a node $j$ to a node $i$ at a time step $t$. And the next message is defined as the following:

$$m^{t+1}_{j \rightarrow i}(x_i) = \max_{x_j} [\psi_i(x_j, x_i) \psi_j(x_j) \prod_{k \in N(j) \setminus i} m^t_{k \rightarrow j}(x_j)]$$

- $b^t_i(x_i)$ is a belief of a node $i$ at a time step $t$. It is the same with belief propagation algorithm and also represents the marginal probability of $X_i$.

$$b^t_i(x_i) = \psi_i(x_i) \prod_{j \in N(i)} m^t_{j \rightarrow i}(x_i)$$

But what we want to know is the configuration, in other words, the set of values which gives the highest marginal probability to each random variable. Therefore, MAP assignment of $X_i$ can be given by:

$$X_i = \arg \max_{x_i} b^t_i(x_i)$$

It is so much clear that each maximum value comprises the maximum probability configuration, since the probability distribution of $X$ is the product of them. As with belief propagation algorithm, the exact MAP assignment can be calculated after $d$ iterations where $d$ denotes the diameter of $G$. But in some cases, there may exists multiple MAP assignments. Some perturbation(small random vertex potentials) may help to find a unique MAP assignment.