Message Passing Algorithms

Seungjin Choi

Department of Computer Science and Engineering
Pohang University of Science and Technology
77 Cheongam-ro, Nam-gu, Pohang 37673, Korea
seungjin@postech.ac.kr
Message Passing Algorithms

Several methods have been developed independently but all of them are a special instance of message passing:

- Forward-backward algorithm for HMM
- Pearl’s belief propagation
- Sum-product or max-product algorithms
- Kalman filter
- Viterbi algorithm
Outline

- **Sum-product algorithm**: Compute marginal probabilities.
- **Max-product algorithm**: Compute the modes.
- **Sum-product = belief propagation = message passing** is explained mainly for undirected graphical models.
- The same idea can be applied to directed graphs or factor graphs, although algorithms are slightly different.
Sum-Product Algorithm

▶ A method of probabilistic inference that only works on trees.
  ▶ Can compute the marginal probabilities of all nodes with the same time complexity as computing the marginal probability of just one node.
  ▶ A node-to-node message passing protocol that reuses messages between computing marginal probabilities.
▶ Works for both directed and undirected graphical models.
▶ In the case of directed graphical models, the algorithm can be used only if their moralized graph is a tree or polytree. That is, belief propagation works for singly-connected DAGs. A DAG is singly-connected if its underlying undirected graph is a tree.
An undirected tree: One and only one path between any pair of nodes.

A directed tree: A directed graph whose moralized graph is an undirected tree (at most one parent).

A ploytree: Its moralized graph has cycles.
Parameterization and Conditioning on $\mathcal{G}(\mathcal{V}, \mathcal{E})$

- **Parameterization**
  - **Undirected trees**
    \[
p(x) = \frac{1}{Z} \left[ \prod_{i \in \mathcal{V}} \psi(x_i) \prod_{(i,j) \in \mathcal{E}} \psi(x_i, x_j) \right]
    \]
  - **Directed trees**
    \[
p(x) = p(x_r) \prod_{(i,j) \in \mathcal{E}} p(x_j|x_i)
    \]

Given $p(x_r)$ and $p(x_j|x_i)$ of a directed graph, the potentials of the corresponding undirected graph can be parameterized as:

\[
\psi(x_i) = \begin{cases} 
p(x_r) & i = r \\
1 & i \neq r
\end{cases}, \quad \psi(x_i, x_j) = p(x_j|x_i).
\]

\[\Rightarrow Z = 1\]

- **Conditioning**

\[
\psi^E(x_i) \equiv \begin{cases} 
\psi(x_i)\delta(x_i, \bar{x}_i) = \delta(x_i, \bar{x}_i) & i \in \mathcal{E} \\
\psi(x_i) = 1 & i \notin \mathcal{E}
\end{cases}
\]

\[
p(x|\bar{x}_E) = \frac{1}{Z^E} \left[ \prod_{i \in \mathcal{V}} \psi^E(x_i) \prod_{(i,j) \in \mathcal{E}} \psi(x_i, x_j) \right]
\]
The parameterization of unconditional distributions on trees is given by

\[
p(x) = \frac{1}{Z} \left[ \prod_{i \in V} \psi(x_i) \prod_{(i,j) \in E} \psi(x_i, x_j) \right].
\]

The parameterization of conditional distributions on trees is given by

\[
p(x \mid \bar{x}_E) = \frac{1}{Z^E} \left[ \prod_{i \in V} \psi^E(x_i) \prod_{(i,j) \in E} \psi(x_i, x_j) \right].
\]

The parameterization of unconditional distributions and conditional distributions on trees, is formally identical, involving a product of potential functions associated with each node and each edge in the graph.
Elimination: A Review

1. Choose an elimination ordering \( I \) in which the query node \( f \) is the final node.

2. Place all potentials on an active list.

3. Eliminate a node \( i \) by removing all potentials referencing the node from the active list, taking the product, summing over \( x_i \), and placing the resulting intermediate factor back on the active list.

\[
p(x_1, \bar{x}_6) = \frac{1}{2} \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} \sum_{x_6} \psi(x_1, x_2) \psi(x_1, x_3) \psi(x_2, x_4) \psi(x_3, x_5) \psi(x_2, x_5, x_6) \delta(x_6, \bar{x}_6)
\]

\[
= \frac{1}{2} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) \sum_{x_4} \psi(x_2, x_4) \sum_{x_5} \psi(x_3, x_5) \sum_{x_6} \psi(x_2, x_5, x_6) \delta(x_6, \bar{x}_6)
\]

\[
= \frac{1}{2} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) \sum_{x_4} \psi(x_2, x_4) \sum_{x_5} \psi(x_3, x_5) m_6(x_2, x_5)
\]

\[
= \frac{1}{2} \sum_{x_2} \psi(x_1, x_2) \sum_{x_3} \psi(x_1, x_3) m_5(x_2, x_3) \sum_{x_4} \psi(x_2, x_4)
\]

\[
= \frac{1}{2} \sum_{x_2} \psi(x_1, x_2) m_4(x_2) \sum_{x_3} \psi(x_1, x_3) m_5(x_2, x_3)
\]

\[
= \frac{1}{2} \sum_{x_2} \psi(x_1, x_2) m_4(x_2) m_3(x_1, x_2)
\]

\[
= \frac{1}{2} m_2(x_1)
\]
Special Features on Trees

▶ **Elimination Orderings:** depth-first traversal of the tree.
  ▶ Treat $f$ as a root and view the tree as a directed tree by directing all edges of the tree to point away from $f$.
  ▶ Any elimination ordering in which a node is eliminated only after all of its children in the directed version of the tree are eliminated.
  ▶ All subtrees with no evidence nodes can be ignored.

▶ **Elimination Step:** the intermediate factor that is created when a node is eliminated
  ▶ Which nodes appear in the potential created after summing over $j$?
    ▶ nothing in the subtree below $j$ (already eliminated)
    ▶ nothing outside the subtree, due to the assumption that the graph is tree
      (For a node $k$ in the subtree and a node $l$ outside of the subtree, there can be no potential $\psi(x_k, x_l)$ in the probability model.)
    ▶ only $i$ from $\psi(x_i, x_j)$
Message-Passing

- **Messages:** \( m_{ji}(x_i) = \sum_{x_j} \left[ \psi^E(x_j) \psi(x_i, x_j) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}(x_j) \right] \).

- **Final node** \( x_f \): \( p(x_f | \bar{x}_E) \propto \psi^E(x_f) \prod_{e \in \mathcal{N}(f)} m_{ef}(x_f) \).

- **Computational complexity:** \( O(m^2) \) for each message if \( x_j \) has \( m \) states and \( O(2n) \) for final node where \( n \) is the number of edges.

- **Message-passing protocol:** A node can send a message to a neighboring node when it has received messages from all of its other neighbors.
Message-Passing Protocol
Algorithm Outline: Sum-Product
(Belief Propagation)

1. Choose a root node (arbitrarily or as first query node).
2. If $j$ is an evidence node, $\psi^E(x_j) = \delta(x_j, \bar{x}_j)$, else $\psi^E(x_j) = 1$.
3. Pass messages from leaves up to root and then back down using

$$m_{ji}(x_i) = \sum_{x_j} \left[ \psi^E(x_i) \psi(x_i, x_j) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}(x_j) \right].$$

4. Given messages, compute the marginal of $x_f$ using

$$p(x_f | \bar{x}_E) \propto \psi^E(x_f) \prod_{k \in \mathcal{N}(f)} m_{kf}(x_f).$$
Key Insights

- Messages can be reused.
- Can obtain all marginals by simply doubling the amount of work required to compute a single marginal.
- The effect of computing over all possible elimination orderings (a huge number) by computing all possible messages (a small number).
Maximum a Posteriori (MAP) Configuration

The problem of MAP is to find the maximal probability that can be achieved by some set of random variables \((x_F)\), given a set of observation \((x_E)\), where \((E, F)\) is a partition of the indices, \(x = (x_1, \ldots, x_n)\).

1. Find the maximal probability:

\[
\max_{x_F} p(x_F | \bar{x}_E) = \max_{x_F} p(x_F, \bar{x}_E)
\]

\[
= \max_x p(x) \delta(x_E, \bar{x}_E)
\]

\[
= \max_x p^E(x).
\]

2. Find a configuration that achieves the maximal probability:

\[
x^* \in \arg \max_x p^E(x).
\]
MAP: An Example

\[
\max_x p(x) = \max_{x_1} \max_{x_2} \max_{x_3} \max_{x_4} \max_{x_5} p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_2)p(x_5|x_3)
\]

\[
= \max_{x_2} p(x_2|x_1) \max_{x_3} p(x_3|x_2) \max_{x_4} p(x_4|x_2) \max_{x_5} p(x_5|x_3).
\]
We define messages, $m_{ji}^{\text{max}}$, similarly as for the marginalization case, except that all sum operators are replaced by max operators:

$$m_{53}^{\text{max}}(x_3) = \max_{x_5} \psi^E(x_5)\psi(x_3, x_5),$$
$$m_{42}^{\text{max}}(x_2) = \max_{x_4} \psi^E(x_4)\psi(x_2, x_4),$$
$$m_{32}^{\text{max}}(x_2) = \max_{x_3} \psi^E(x_3)\psi(x_2, x_3) m_{53}^{\text{max}}(x_3).$$

Then the maximum probability is achieved by

$$\max_x p(x) = \max_{x_1} \psi^E(x_1) m_{21}^{\text{max}}(x_1).$$
Maximum Configuration

Given two nodes $i$ and $j$ such that $i$ is the parent of $j$, we need to record the maximizing values in a table $\delta_{ji}(x_i)$, which is computed as:

$$\delta_{ji}(x_i) \in \arg \max_{x_j} \psi^E(x_j)\psi(x_i, x_j) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}^{\max}(x_j),$$

which just becomes a table in the dimension of $x_i$ in the discrete case. The root node then becomes

$$x_f^* \in \arg \max_{x_f} \psi^E(x_f) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kf}^{\max}(x_f).$$

Once $x_f$ is computed, for neighbor $k$ of the root node, $x_k^* = \delta_{kf}(x_f^*)$. Starting this propagation at the root node and continuing until all nodes have been hit, will yield the maximum configuration.
Algorithm Outline: Max-Product

Max-Product Version of the Algorithm: inward pass (leaves $\rightarrow$ root)

$$m_{ji}^{\text{max}} = \max_{x_j} \left[ \psi^E(x_j) \psi(x_i, x_j) \prod_{k \in (\mathcal{N}(j)) \setminus i} m_{kj}^{\text{max}}(x_j) \right],$$

$$\max_p p^E(x) = \max_{x_i} \left[ \psi^E(x_i) \prod_{j \in (\mathcal{N}(i))} m_{ji}^{\text{max}}(x_i) \right].$$

MAP configurations inward pass and outward pass

$$x_f^* \in \arg\max_{x_f} \left[ \psi^E(x_f) \prod_{e \in \mathcal{N}(f)} m_{ef}^{\text{max}}(x_f) \right],$$

$$\delta_{ji}(x_i) \in \arg\max_{x_j} \left[ \psi^E(x_j) \psi(x_i, x_j) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}^{\text{max}}(x_j) \right].$$

It is often safer to work with the log of probabilities:

$$\log m_{ji}^{\text{max}} = \max_{x_j} \left[ \log \psi^E(x_j) + \log \psi(x_i, x_j) + \sum_{k \in (\mathcal{N}(j)) \setminus i} \log m_{kj}^{\text{max}}(x_j) \right].$$