

# Belief Propagation and Loopy Belief Propagation

## 1 Graph-Structured Factor Graphs

A graph-structured factor graph is a factor graph in which the factors involve at most two nodes. Graph structured factor graphs can be represented by a set of random variables (nodes)  $V$  and a set of edges  $E$  on  $V$  where, for  $X, Y \in V$ , we will write  $\{X, Y\} \in E$  if the edge  $\{X, Y\}$  is an edge in  $E$ . For convenience here we assume that for each  $X \in V$  the support of  $X$  (the set of values that the variable  $X$  can have) is finite and we denote the support of  $X$  by  $\mathcal{D}(X)$ . For each  $X \in V$  we assume a factor  $\Psi_X$  such that for each  $x \in \mathcal{D}(X)$  we have that  $\Psi_X(x)$  is a nonnegative real number. We also assume a factor  $\Psi_{X,Y}$  for each edge  $\{X, Y\} \in E$  such that for  $x \in \mathcal{D}(X)$  and  $y \in \mathcal{D}(Y)$  we have that  $\Psi_{X,Y}(x, y)$  is a nonnegative real number. The edges in  $E$  are undirected and we have  $\Psi_{X,Y}(x, y) = \Psi_{Y,X}(y, x)$ .

We let  $\sigma$  range over complete assignments of values to variables in  $V$  — for any  $X \in V$  we have  $\sigma(X) \in \mathcal{D}(X)$ . We then define the following unnormalized probability.

$$Z(\sigma) = \left( \prod_{X \in V} \Psi_X(\sigma(X)) \right) \left( \prod_{\{X, Y\} \in E} \Psi_{X, Y}(\sigma(X), \sigma(Y)) \right) \quad (1)$$

$$P(\sigma) = \frac{Z(\sigma)}{Z} \quad (2)$$

$$Z = \sum_{\sigma} Z(\sigma) \quad (3)$$

The quantity  $Z$  is called the partition function. If the factors  $\Psi_X$  and  $\Psi_{X,Y}$  involve parameters  $\Theta$  the one writes  $Z(\Theta)$  and the partition function is a function of the parameters defining the factors. Here we will not be concerned with parameters.

We let  $\rho$  range over partial assignments of values to variables —  $\rho$  is a finite set of pairs each of which specifies a value for one of the variables in  $V$ . We

write  $\sigma \sqsubseteq \rho$  if  $\sigma$  satisfies all of the assignments in  $\rho$ . We define  $Z(\rho)$  as follows.

$$Z(\rho) = \sum_{\sigma \sqsubseteq \rho} Z(\sigma) \quad (4)$$

$$P(\rho) = \sum_{\sigma \sqsubseteq \rho} P(\sigma) \quad (5)$$

$$= \frac{Z(\rho)}{Z} \quad (6)$$

If  $\rho'$  is a partial assignments that extends  $\rho$ , i.e.,  $\rho'$  contains all the pairs in  $\rho$  plus possibly additional pairs, written  $\rho' \sqsupseteq \rho$ , then we have the following.

$$P(\rho' | \rho) = \frac{P(\rho')}{P(\rho)} \quad (7)$$

$$= \frac{Z(\rho')}{Z(\rho)} \quad (8)$$

To compute conditional probabilities of the form  $P(\rho' | \rho)$  it therefore suffices to be able to compute values of the form  $Z(\rho)$ .

## 2 Belief Propagation for Trees

We now consider the case where the edges in  $E$  form a tree. In this case the junction tree algorithm can be formulated directly on the graph  $E$  with a message  $Z_{X \rightarrow Y}$  for  $\{X, Y\} \in E$ . The message  $Z_{X \rightarrow Y}$  is a function on  $\mathcal{D}(Y)$ . To define the semantics of the message we first define  $\mathcal{V}_{X \rightarrow Y}$  to be the set of  $U \in V$  such that the path in the tree  $E$  from  $U$  to  $Y$  includes  $X$ . Note that  $X \in \mathcal{V}_{X \rightarrow Y}$ . Intuitively, the subtree  $\mathcal{V}_{X \rightarrow Y}$  is the “input” to the message  $Z_{X \rightarrow Y}$ . The semantics of the message is defined as follows where  $\sigma$  ranges over assignments of values to the set  $\mathcal{V}_{X \rightarrow Y}$ .

$$Z_{X \rightarrow Y}(y) = \sum_{\sigma} \begin{cases} \Psi_{X,Y}(\sigma(X), y) \\ \prod_{U \in \mathcal{V}_{X \rightarrow Y}} \Psi_U(\sigma(U)) \\ \prod_{\{U,W\} \in E, U,W \in \mathcal{V}_{X \rightarrow Y}} \Psi_{U,W}(\sigma(U), \sigma(W)) \end{cases} \quad (9)$$

In words, the message  $Z_{X \rightarrow Y}(y)$  is the partition function for the factor graph consisting of the input subtree  $\mathcal{V}_{X \rightarrow Y}$  plus the edge from  $X$  to  $Y$  but with the value of  $Y$  fixed at  $y$ . The messages can be efficiently computed using the

following equations which can be proved as a lemma from the definition of the message.

$$Z_{X \rightarrow Y}(y) = \sum_{x \in \mathcal{D}(X)} \Psi_X(x) \left( \prod_{\{U, X\} \in E; U \neq Y} Z_{U \rightarrow X}(x) \right) \Psi_{X, Y}(x, y) \quad (10)$$

Equation (10) defines the belief propagation algorithm. As noted above, this is equivalent to the junction tree algorithm but has a simpler form due to the simple form of a tree structured factor graph. In the junction tree algorithm there would be an additional node representing each edge in  $E$  (so that the cover property holds), and we get a message into the node for the edge  $X \rightarrow Y$  which is defined on  $\mathcal{D}(X)$  as well as a message out of the node for the edge  $X \rightarrow Y$  which is defined on  $\mathcal{D}(Y)$ . In (10) we have only the outgoing messages. Note that the recursion in (10) terminates because the recursive call always involves computing the partition function of a smaller subtree.

We can compute  $Z(\rho)$  for any  $\rho$  by restricting the domain of each variables assigned in  $\rho$  to consist of the single value assigned in  $\rho$  and then applying the belief propagation algorithm under this restriction of domains. For any variable  $X$  we also have the following.

$$Z(X = x) = \Psi_X(x) \left( \prod_{\{Y, X\} \in E} Z_{Y \rightarrow X}(x) \right) \quad (11)$$

### 3 Loopy Belief Propagation

We now consider a graph structured factor graph but where the graph  $E$  is not a tree (it is “loopy”). Loopy belief propagation (loopy BP) uses a form of equation (10) even though the equations no longer have a clear semantics. The intuition is that if the loops are long then the effect of the loops fades out as the messages propagate and the resulting answer is accurate in any case. Loopy BP has proved very effective in many applications. In loopy BP we assume a message  $m_{X \rightarrow Y}^t$  for every edge  $\{X, Y\} \in E$  and compute a new system of messages as follows.

$$m_{X \rightarrow Y}^{t+1}(y) = \frac{1}{Z} \sum_{x \in \mathcal{D}(X)} \Psi_X(x) \left( \prod_{\{U, X\} \in E; U \neq Y} m_{U \rightarrow X}^t(x) \right) \Psi_{X, Y}(x, y) \quad (12)$$

In (12) the constant  $Z$  is selected so that the messages are normalized, i.e.,  $\sum_{y \in \mathcal{D}(Y)} m_{X \rightarrow Y}(y) = 1$ . It is useful to compare (12) with (10). We have replaced  $Z_{X \rightarrow Y}$  by  $m_{X \rightarrow Y}$  because in the loopy case we no longer have a semantics

for the messages as partition function values. The messages  $m_{X \rightarrow Y}^0$  are typically initialized to be uniform. Normalization is needed in the loopy case to prevent the message values from diverging exponentially as  $t$  increases. Equation (12) can be computed more efficiently as follows.

$$B_X^t(x) = \Psi_X(x) \prod_{\{X,Y\} \in E} m_{Y \rightarrow X}^t(x) \quad (13)$$

$$m_{X \rightarrow Y}^{t+1}(y) = \frac{1}{Z_{X \rightarrow Y}^{t+1}} \sum_{x \in \mathcal{D}(X)} \frac{B_X^t(x)}{m_{Y \rightarrow X}^t(x)} \Psi_{X,Y}(x, y) \quad (14)$$