TTIC 31230, Fundamentals of Deep Learning

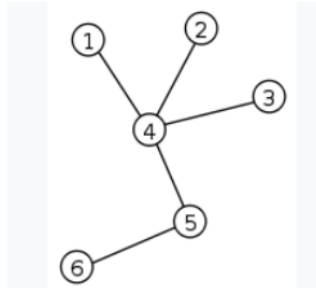
David McAllester, Winter 2020

Loopy Belief Propagation (Loopy BP)

Loopy Belief Propagation (Loopy BP)

We design an algorithm that is correct for tree graphs and use it on non-tree (loopy) graphs.

Belief Propagation on Trees



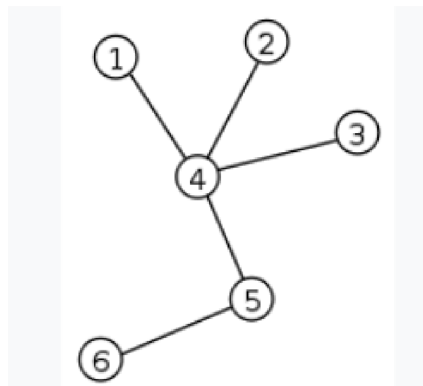
Belief Propagation is a message passing procedure (actually dynamic programming).

For each edge $\{i, j\}$ and possible value $\tilde{y}$ for node i we define $Z_{j \to i}[c]$ to be the partition function for the subtree attached to i through j and with $\hat{y}[i]$ restricted to c.

The function $Z_{j \to i}$ on the possible values of node i is called the **message** from j to i.

The reverse direction message $Z_{i \to j}$ is defined similarly.

Dynamic Programming Computes the Messages



$$Z_{j\to i}[c] = \sum_{c'} e^{s_n[j,c']+s_e[j,i,c',c]} \left(\prod_{k\in N(j),\, k\neq i} Z_{k\to j}[c'] \right)$$

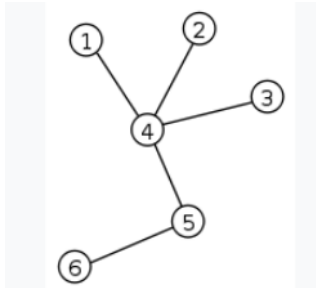
Loopy BP

In a Loopy Graph we can initializing all message $Z_{i\to j}[c] = 1$ and then repeating (until convergence) the updates

$$\tilde{Z}_{j\to i}[c] = \frac{1}{Z_{j\to i}}\, Z_{j\to i}[c] \qquad Z_{j\to i} = \sum_{c} Z_{j\to i}[c]$$

$$Z_{j\to i}[c] = \sum_{c'} e^{s_n[j,c'] + s_e[j,i,c',c]} \left(\prod_{k\in N(j),\; k\neq i} \tilde{Z}_{k\to j}[c'] \right)$$

Computing Node Marginals from Messages



$$Z_i(c) \doteq \sum_{\hat{y}:\ \hat{y}[i]=c} e^{s(\hat{y})}$$

$$= e^{s_i[c]} \left(\prod_{j \in N(i)} Z_{j \to i}[c] \right)$$

$$P_i(c) = Z_i(c)/Z, \qquad Z = \sum_c Z_i(c)$$

Computing Edge Marginals from Messages

$$Z_{i,j}(c, c') \doteq \sum_{\hat{y}:\ \hat{y}[i]=c,\ \hat{y}[j]=c'} e^{s(\hat{y})}$$

$$= e^{s_n[i,c]+s_n[j,c']+s_e[i,j,c,c']} \prod_{k\in N(i),\ k\neq j} Z_{k\to i}[c] \prod_{k\in N(j),\ k\neq i} Z_{k\to j}[c']$$

$$P_{i,j}(c, c') = Z_{i,j}(c, c')/Z \quad Z = \sum_{c,c'} Z_{i,j}(c, c')$$

END