E2 205 Error-Control Coding
Lecture 20

Rajat Chopra

October 21, 2019

1 Belief Propagation Continued

$\mathcal{X}$

Example 1 *Max-Likelihood decoding of [7,4,2] block code*

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$F_i(x_i) = \max_{\tilde{x}_i} \prod_{i=1}^{n} p(y_i|x_i)\mathcal{X}_{123}(x_1x_2x_3)\mathcal{X}_{346}(x_3x_4x_6)\mathcal{X}_{457}(x_4x_5x_7)$$

Thus, here we are operating in the max product semiring.

Example 2 *ML code-symbol decoding the same code*

$$p(x_i|\underline{y}) \propto p(x_i, \underline{y}) = \sum_{\tilde{x}_i} p(\underline{x}, \underline{y}) = \sum_{\tilde{x}_i} p(\underline{x})p(\underline{y}|x)$$

$$= \sum_{\tilde{x}_i} \prod_{i=1}^{n} p(y_i|x_i)\mathcal{X}_{123}(x_1x_2x_3)\mathcal{X}_{346}(x_3x_4x_6)\mathcal{X}_{457}(x_4x_5x_7)$$

this time we are operating in the sum-product semiring.

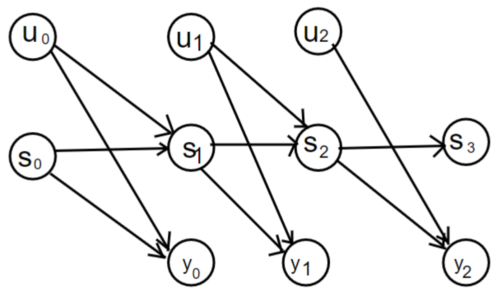

Figure 1:

1.1 An example of a bagesean network

see Figure 1

$$p(\{x_i\}_{i=0}^{2}, \{s_i\}_{i=0}^{3}, \{y_i\}_{i=0}^{2}) = p(s_o) \prod_{i=1}^{2} p(u_i) p(s_{i+1}|s_i u_i) p(y_i|s_i u_i)$$

The graph is called a DAG(Direct Acyclic Graph)

Example 3 *Let us pretend that the bagesean network to the left represents a convolutional code of rate k/n*
$s_i = state\ of\ time\ i$
$y_i = n - tuple\ output\ at\ time\ i$
$u_i = k - tuple\ input\ at\ time\ i$
$(see figure 2)$
the stable alphabet here is of size 4
ML codeword decoding of a convolutional code

$$\mathbb{F}_i(u_i) = \max_{\tilde{u}_i} p(\underline{y}|\underline{u}) \propto \max_{\tilde{u}_i} p(\underline{u}|\underline{y})$$

$$= \max_{\tilde{u}_i} p(\underline{y}|\underline{u})$$

$$= \max_{\tilde{u}_i} p(s_o) \prod_{i=0}^{2} p(u_i) p(s_{i+1}|s_i u_i) p(y_i|s_i u_i)$$

This is once again an instance of the MPF problem in the max product semirirng

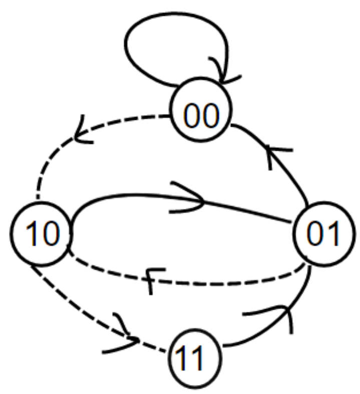

Figure 2:

2 Using the GDL to solve the MPF problem

2.1 Local Domain Graph

$$\mathbb{F}(x_4x_5x_6) = \sum_{x_1x_2x_3} (-1)^{x_1x_4+x_2x_5+x_3x_6} f(x_1x_2x_3)$$

Local Domains
{1,4} {2,5} {3,6} {1,2,3} {4,5,6}

The local domain graph is a completely completed graph whose nodes are in 1-1 correspondence with the local(see figure 3)

domains and the edge connecting nodes
see figure 4

weight of node $s_i = |s_i|$

weight of an egde $=s_i \cap s_j$

step1: Organise the local domains so as to form if possible , a junction tree

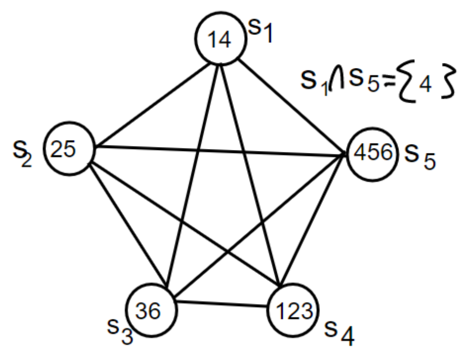

Figure 3:

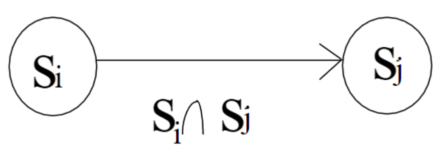

Figure 4:

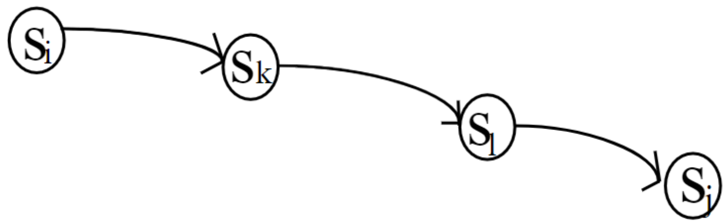

Figure 5:

2.2 Junction Tree

A junction tree (in the above context) is a tree whose nodes are in 1-1 correspondence with the local domains and whose edges are a subset of the edges of the local domain graph.

furthermore J is a juntion tree iff in the unique path leading from node i to node j , corresponding to local domain s_i, s_i,any intermediate vector(figure 5) $s_l \subseteq s_j$

$s_k \subseteq s_i \cap s_j$
$s_l \subseteq s_i \cap s_j$

it turns out that if it is possible to construct tree then the junction trace corresponds to a maximal-weight spanning tree of the local domain (bcd - domain) graph
Hence the junction tree can be constructed using an algorithm for constructing maximal-weight spanning tree.

Example 4 *of prism's greedy alogrithm for constructing a maximal-weight spanning tree (see figure 6)*

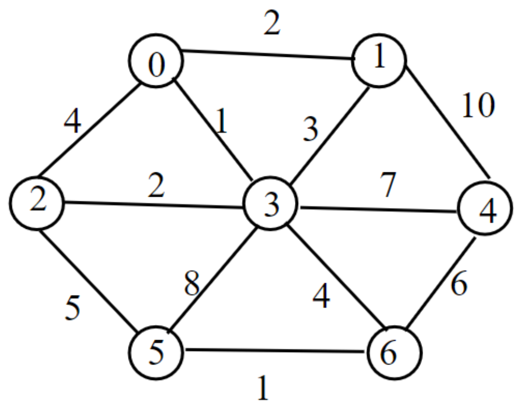

Figure 6:

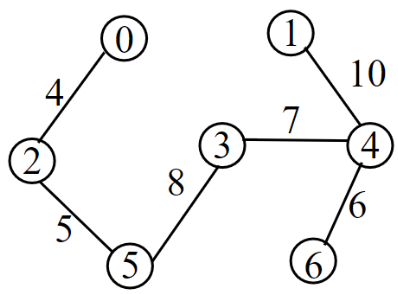

Figure 7:

2.3 Aside(tree)

A tree is a connected graph with no cycles (equvalently betwen any two nodes in the tree,there is a unique path),In every tree we have that number of nodes-number of edges=1 (can prove this by induction)(see figure7)

$$\prod_{i=1}^{7} p(y_i|x_i)\mathcal{X}_{124}(x_1x_2x_4)\mathcal{X}_{346}(x_3x_4x_6)\mathcal{X}_{456}(x_4x_5x_6)$$

By projection of a graph (that is derived from the local domain graph by deleting edges) on to variable x_i ,along with any ends connecting there ver-

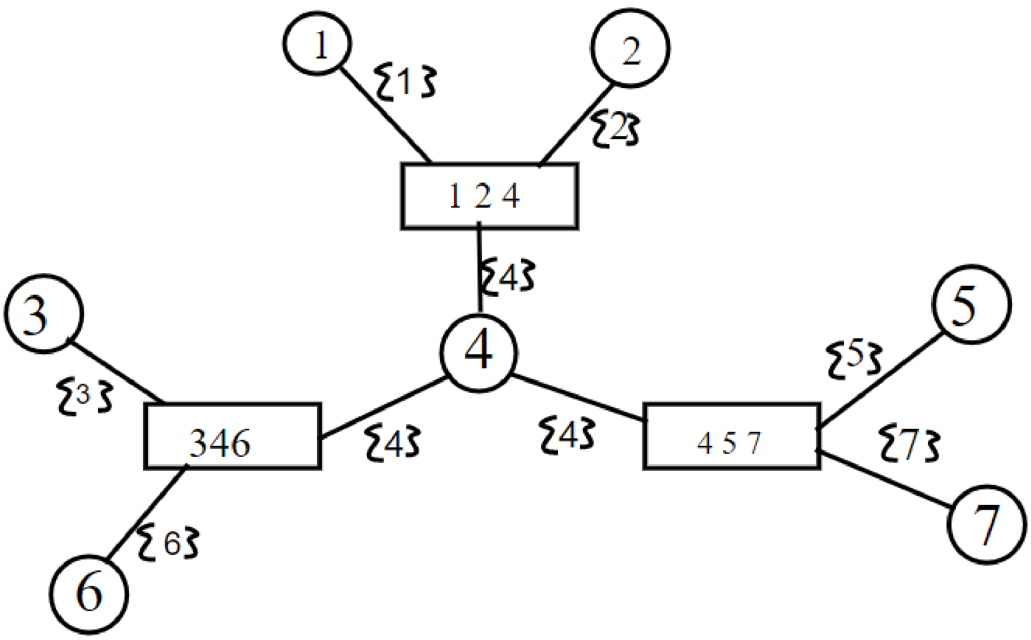

Figure 8:

tices we label each node in the projection by x_i,we also label each edge by x_i. A little thought well reved that J(derived from the local domain graph is a junction tree iff, all the n projections associated to the n variables $\{x_1, x_2........x_n\}$ are connected. The node weight NW(g) of a graph is defined to be the sum of the weights of the nodes in the graph.The edge weight g is similarly, the sum of the weights of the edges of g.

2.4 Note

1. if J is a junction tree, all its n projections are tree.Hence NW(J)-FW(J)=n

2. if J is a tree, but not a junction tree, then at least one of its projections will be a forest, rather than a tree. Hence in this case, we will have (NW(J)-EW(J)) greater than number of variables(n)

since this is the maximum possible edge weight,a maximal-weight spanning tree algorithm can be used to recover the junction tree, then it exists. This suggests the following approach to junction tree construction.

1. find a minimal-weight spanning tree for the local domain.

2. if this maximal -weight spanning tree satisfies $EW = NW - n$, then we have recovered a junction trees. if $EW! = NW - n$ then, it is not possible to construct a junction tree.

2.5 An additional consideration

when using prims greedy algorithm to construct a junction tree , one often runs into ties one reaches ties by selecting that additional node with minimize (figure 9)

$$q_{si} + q_{sj} - q_{si \cap si}$$

turns out that (1) is reflection of the cost of passing a messgae from node s_i to node s_j.

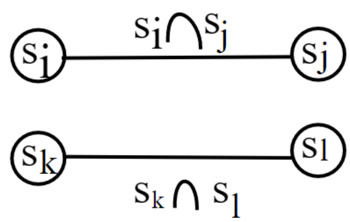

Figure 9: