UC Berkeley
Department of Electrical Engineering and Computer Science
Department of Statistics

EECS 281A / STAT 241A Statistical Learning Theory

Problem Set 2

Fall 2011

Issued: Wednesday, September 28, 2011 **Due:** Monday, October 10, 2011

Reading: For this problem set: Chapters 4, 16.

Problem 2.1

Conditioning:

Consider an undirected graph that is a single cycle (a "ring"). Let each of the nodes be discrete-valued, taking on one of K states.

(a) Note that if we condition on one of the nodes then we break the cycle into a chain. We can run the sum-product algorithm on the chain. Devise an algorithm for computing all single-node marginals in the cycle via multiple runs of the sum-product algorithm.

(b) Another approach to computing all single-node marginals in the cycle is to use the junction tree algorithm. What is size of biggest clique that arises during a sensible triangulation of the cycle? Compare the computational complexity of the junction tree algorithm using this triangulation to that of the conditioning-based algorithm in (a).

Problem 2.2

Sum-product algorithm from junction tree:

Consider an undirected tree, $G = (V, E)$, parametrized by pairwise potentials, $\psi_{st}(x_s, x_t)$, for $(s, t) \in E$. Outline the junction tree construction for this model, writing out the propagation rules explicitly, and derive the sum-product algorithm for this model from the junction tree propagation rules.

Problem 2.3

Sum-product algorithm: Consider the sum-product algorithm on an undirected tree with poential functions ψ_s and ψ_{st}. Consider any initialization of the messages such that $M_{t \to s}(x_s) > 0$ for all directions $t \to s$ and all states x_s.

(a) Prove by induction that:

(i) the sum-product algorithm, with the flooding schedule, converges in at most diameter of the graph iterations. (Diameter of the graph is the length of the longest path.)

(ii) the message fixed point M^* can be used to compute marginals

$$p(x_s) \quad \propto \quad \psi_s(x_s) \prod_{t \in N(s)} M^*_{t \to s}(x_s)$$

for every node of the tree.

(b) Implement the sum-product algorithm for an arbitrary tree-structured graph. Please document and hand in your code, including brief descriptions of the data structures that you define and steps involved. Run your algorithm on the tree in Figure 1, using the edge potential functions

$$\psi_{st}(x_s, x_t) = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

for all edges (s,t), and the singleton potential functions

$$\psi_s(x_s) = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \quad \text{for } s = 1, 3, 5, \text{ and} \quad \psi_s(x_s) = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} \quad \text{otherwise.}$$

Report the values of the single node marginals for all nodes.

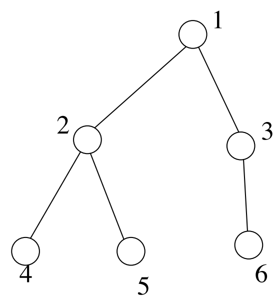

Figure 1: Tree for Problem 2.3(b).

Problem 2.4

Computing edge marginals in a tree. Consider an undirected tree $T = (V, E)$, where V are the nodes and E are the edges. Recall that we can use the sum-product algorithm to compute all the single node marginals $p(x_i)$.

(a) In this part you are to provide a modification of the sum-product algorithm that will yield edge marginals; i.e., a probability $p(x_i, x_j)$ for $(i,j) \in E$. What is the running time of the algorithm for computing all edge marginals?

(b) Now consider computing arbitrary pairwise marginals in a tree; a probability $p(x_i, x_j)$ for $(i,j) \notin E$. How can such a marginal be computed for a single pair (i,j)? What can you say about the running time of the algorithm?

Problem 2.5

Hammersley-Clifford and Gaussian models: Consider a zero-mean Gaussian random vector $(X_1, \ldots, X_N)$ with a strictly positive definite $N \times N$ covariance matrix $\Sigma \succ 0$. For a given undirected graph $G = (V, E)$ with N vertices, suppose that $(X_1, \ldots, X_N)$ obeys all the basic conditional independence properties of the graph G (i.e., one for each vertex cut set).

(a) Show the sparsity pattern of the inverse covariance $\Theta = (\Sigma)^{-1}$ must respect the graph structure (i.e., $\Theta_{ij} = 0$ for all indices i, j such that $(i,j) \notin E$.)

(b) Interpret this sparsity relation in terms of cut sets and conditional independence.