Lecture 7:

Fully Observed Trees

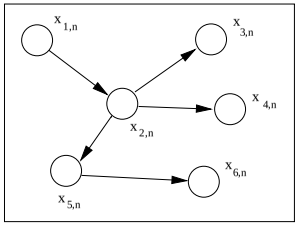
Sam Roweis

January 28, 2002

Directed Tree Models

- A tree model is a DAGM in which each node has exactly one parent except the "root" which has no parents. Notation:
  $\mathbf{y}_i \equiv$ a node $\mathbf{x}_i$ and its single parent $\mathbf{x}_{\pi_i}$.
  $\mathbf{V}_i \equiv$ set of joint configurations of node i and its parent

$$p(\mathbf{x}) = p(\mathbf{x}_{\text{root}}) \prod_{i \neq \text{root}} p(\mathbf{x}_i | \mathbf{x}_{\pi_i})$$



Outline

- Tree models
- ML parameters given structure
- Optimal structure
- MWST and Kruskal's algorithm
- ML Tree Learning
- Notes

Undirected Tree Models

- Trees have no "explaining-away".
  Therefore, d-separation and regular separation are equivalent.
- For fully observed models, undirected and directed models are equivalent and choice of root is arbitrary.
  The cliques are all pairs of connected nodes.

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \psi_i(\mathbf{x}_i, \mathbf{x}_{\pi_i}) = \frac{1}{Z} \prod_i \psi_i(\mathbf{y}_i)$$

  where we can make $Z = 1$ with the choice $\psi_i(\mathbf{y}_i) = p(\mathbf{x}_i | \mathbf{x}_{\pi_i})$
  except for one clique involving the root: $\psi_j(\mathbf{y}_j) = p(\mathbf{x}_r) p(\mathbf{x}_j | \mathbf{x}_{\pi_j})$

Likelihood function

- Trees are in the exponential family with $\mathbf{y}_i$ as sufficient statistics:

$$\begin{aligned}\ell(\theta;\mathcal{D}) &= \sum_n \log \prod_i \psi_i(\mathbf{y}_i^n) \\ &= \sum_n \sum_i \sum_{\mathbf{v}\in\mathbf{V}_i} [\mathbf{y}_i^n = \mathbf{v}] \log \psi_i(\mathbf{v}) \\ &= \sum_i \sum_{\mathbf{v}\in\mathbf{V}_i} N_i(\mathbf{v}) \log \psi_i(\mathbf{v})\end{aligned}$$

  where $N_i(\mathbf{v}) = \sum_n [\mathbf{y}_i^n = \mathbf{v}]$.

- Directed model:

$$\begin{aligned}\ell(\theta;\mathcal{D}) = &\sum_{i\neq r} \sum_{\mathbf{v}\in\mathbf{V}_i} N_i(\mathbf{v}) \log p(\mathbf{x}_i|\mathbf{x}_{\pi_i}) + \\ &\sum_{\mathbf{v}\in V_{\text{root}}} N_r(\mathbf{v}) \log p_r(\mathbf{v})\end{aligned}$$

Optimal Structure

- Let us rewrite the likelihood function:

$$\begin{aligned}\ell(\theta;\mathcal{D}) &= \sum_{\mathbf{x}\in\mathbf{V}_{\text{all}}} N(\mathbf{x}) \log p(\mathbf{x}) \\ &= \sum_{\mathbf{x}} N(\mathbf{x}) \left(\log p(\mathbf{x}_r) + \sum_{i\neq r} \log p(\mathbf{x}_i|\mathbf{x}_{\pi_i}) \right)\end{aligned}$$

- ML parameters, are equal to the observed frequency counts $q(\cdot)$:

$$\begin{aligned}\frac{\ell^*}{M} &= \sum_{\mathbf{x}\in\mathbf{V}_{\text{all}}} q(\mathbf{x}) \left(\log q(\mathbf{x}_r) + \sum_{i\neq r} \log q(\mathbf{x}_i|\mathbf{x}_{\pi_i}) \right) \\ &= \sum_{\mathbf{x}} q(\mathbf{x}) \left(\log q(\mathbf{x}_r) + \sum_{i\neq r} \log \frac{q(\mathbf{x}_i, \mathbf{x}_{\pi_i})}{q(\mathbf{x}_{\pi_i})} \right) \\ &= \sum_{\mathbf{x}} q(\mathbf{x}) \sum_{i\neq r} \log \frac{q(\mathbf{x}_i, \mathbf{x}_{\pi_i})}{q(\mathbf{x}_i) q(\mathbf{x}_{\pi_i})} + \sum_{\mathbf{x}} q(\mathbf{x}) \sum_i \log q(\mathbf{x}_i)\end{aligned}$$

- NB: second term does not depend on structure.

ML parameters given structure

- Since trees are just a special case of directed models, the ML estimates of the parameters are just the empirical counts of each node given its parents:

$$p^*(\mathbf{x}_i = \mathbf{v}_i|\mathbf{x}_{\pi_i} = \mathbf{v}_j) = \frac{N(\mathbf{x}_i = \mathbf{v}_i, \mathbf{x}_{\pi_i} = \mathbf{v}_j)}{\sum_{\mathbf{v}_i} N(\mathbf{x}_i = \mathbf{v}_i, \mathbf{x}_{\pi_i} = \mathbf{v}_j)}$$

  except for the root which uses marginal counts $N(\mathbf{x}_r)/M$.

- What about the tree structure (links)? Can we also learn that? In principle, we could search all combinatorial structures, for each compute the ML params and take the best one.
- But is there a better way? Yes. Turns out can be converted to a maximum weight spanning tree problem.

Edge Weights

- Each node in sum $i \neq r$ corresponds to an edge from i to its parent.

$$\begin{aligned}\frac{\ell^*}{M} &= \sum_{\mathbf{x}} q(\mathbf{x}) \sum_{i\neq r} \log \frac{q(\mathbf{x}_i, \mathbf{x}_{\pi_i})}{q(\mathbf{x}_i) q(\mathbf{x}_{\pi_i})} + C \\ &= \sum_{i\neq r} \sum_{\mathbf{x}_i, \mathbf{x}_{\pi_i}} q(\mathbf{x}_i, \mathbf{x}_{\pi_i}) \log \frac{q(\mathbf{x}_i, \mathbf{x}_{\pi_i})}{q(\mathbf{x}_i) q(\mathbf{x}_{\pi_i})} + C \\ &= \sum_{i\neq r} \sum_{\mathbf{y}_i} q(\mathbf{y}_i) \log \frac{q(\mathbf{y}_i)}{q(\mathbf{x}_i) q(\mathbf{x}_{\pi_i})} + C \\ &= \sum_{i\neq r} W(i;\pi_i) + C\end{aligned}$$

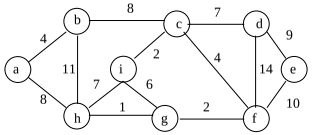
  where the edge weights W are defined by *mutual information*:

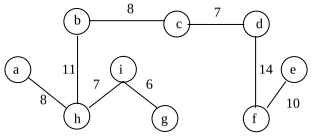
$$W(i;j) = \sum_{\mathbf{x}_i, \mathbf{x}_j} q(\mathbf{x}_i, \mathbf{x}_j) \log \frac{q(\mathbf{x}_i, \mathbf{x}_j)}{q(\mathbf{x}_i) q(\mathbf{x}_j)}$$

- So overall likelihood is sum of weights on edges that we use. We need the maximum weight spanning tree.

Kruskal's algorithm (Greedy Search)

- To find the maximum weight spanning tree A on a graph with nodes U and weighted edges E:
  1. $A \leftarrow \text{empty}$
  2. Sort edges E by nonincreasing weight: $e_1, e_2, \ldots, e_K$.
  3. for $k = 1$ to K $\{A \mathrel{+}= e_k$ unless doing so creates a cycle$\}$





Notes

- *Any* directed tree consistent with the undirected tree found by the algorithm above will assign the same likelihood to any dataset.
- Thus, the root is arbitrary.
  We can just pick one node and orient the edges away from it.
  Or we can work with undirected models.
- For continuous nodes (e.g. Gaussian), the situation is similar, except that computing the mutual information requires an integral.
- Mutual information is the *Kullback-Leibler* divergence between a distribution and the product of its marginals.

Maximum Likelihood Trees

We can now completely solve the tree learning problem:

1. Compute the marginal counts $q(\mathbf{x}_i)$ for each node and pairwise counts $q(\mathbf{x}_i, \mathbf{x}_j)$ for all pairs of nodes.
2. Set the weights to the mutual informations:

$$W(i;j) = \sum_{\mathbf{x}_i, \mathbf{x}_j} q(\mathbf{x}_i, \mathbf{x}_j) \log \frac{q(\mathbf{x}_i, \mathbf{x}_j)}{q(\mathbf{x}_i) q(\mathbf{x}_j)}$$

3. Find the maximum weight spanning tree A=MWST(W).
4. Using the undirected tree A chosen by MWST, pick a root arbitrarily and orient the edges away from the root.
   Set the conditional functions to the observed frequencies:

$$p(\mathbf{x}_i | \mathbf{x}_{\pi_i}) = \frac{q(\mathbf{x}_i, \mathbf{x}_{\pi_i})}{\sum_{\mathbf{x}_i} q(\mathbf{x}_i, \mathbf{x}_{\pi_i})} = \frac{q(\mathbf{x}_i, \mathbf{x}_{\pi_i})}{q(\mathbf{x}_{\pi_i})}$$