Lecture 7:
Fully Observed Trees

Sam Roweis

January 26, 2004

Directed Tree Graphical Models

- Directed trees are DAGMs in which each variable $x_i$ has exactly one
  other variable as its parent $x_{\pi_i}$ except the “root” $x_{\text{root}}$ which has
  no parents. Thus, the probability of a variable taking on a certain
  value depends only on the value of its parent:

\[
p(x) = p(x_{\text{root}}) \prod_{i \neq \text{root}} p(x_i|x_{\pi_i})
\]

- Trees are the next step up from assuming independence.
  Instead of considering variables in isolation, consider them in pairs.

\[
\begin{align*}
\text{Directed model likelihood:} & \quad \ell(\theta; D) = \sum_{n} \log p(x^n) = \sum_{n} \left[ \log p_r(x^n_r) + \sum_{i \neq \text{root}} \log p(x_i^n|x_{\pi_i}^n) \right] \\
& = \sum_{n} \sum_{i} \sum_{v \in V_i} [y_i^n = v] \log p_i(v) \quad \text{indicator trick} \\
& = \sum_{i} \sum_{v \in V_i} N_i(v) \log p_i(v)
\end{align*}
\]

where $N_i(v) = \sum_{n} [y_i^n = v]$ and $p_i(v_i) = p(x_i|x_{\pi_i})$.

Undirected Tree Graphical Models

- Undirected trees are connected, acyclic graphs with exactly (D-1)
  edges if there are D nodes (variables).

- For undirected trees, the cliques are all pairs of connected nodes.

\[
p(x) = \frac{1}{Z} \prod_{i} \psi_i(x_i, x_{\pi_i})
\]

where we can make $Z = 1$ with the choice $\psi_i = p(x_i|x_{\pi_i})$ except
for one clique involving the root: $\psi_j = p(x_j)p(x_{\pi_j})$

- Trees have no “explaining-away” (converging arrows).
  Therefore, d-separation and regular separation are equivalent.

- Directed and undirected trees are equivalent and the choice of root
  is arbitrary (for fully observed models).

- Another characterization of trees: there is exactly one path between
  any pair of nodes (without doubling back).

Likelihood function

- Notation:
  $y_i \equiv$ a node $x_i$ and its single parent $x_{\pi_i}$.
  $V_i \equiv$ set of joint configurations of node $i$ and its parent $x_{\pi_i}$
  $(y_{\text{root}} \equiv x_{\text{root}}$ and $V_{\text{root}} \equiv v_{\text{root}}$)

- Directed model likelihood:

\[
\begin{align*}
\ell(\theta; D) & = \sum_{n} \log p(x^n) = \sum_{n} \left[ \log p_r(x^n_r) + \sum_{i \neq \text{root}} \log p(x_i^n|x_{\pi_i}^n) \right] \\
& = \sum_{n} \sum_{i} \sum_{v \in V_i} [y_i^n = v] \log p_i(v) \\
& = \sum_{i} \sum_{v \in V_i} N_i(v) \log p_i(v)
\end{align*}
\]

where $N_i(v) = \sum_{n} [y_i^n = v]$ and $p_i(v_i) = p(x_i|x_{\pi_i})$.

NB: each node (except root) has exactly one parent, but nodes
may have more than one child.
More on the Likelihood function

- Undirected model likelihood:
  \[ \ell(\theta; D) = \sum_n \log \prod_i \psi_i(y_i^n) \]
  \[ = \sum_n \sum_{v \in V_i} N_i(v) \log \psi_i(v) \]
  where \( N_i(y) = \sum_n [y_i^n = y] \) and \( \psi_i(y_i) = p(x_i|\pi_i) \).
  (Except for one clique involving the root: \( \psi_j = p(x_r)p(x_j|x_{\pi_j}) \))

- Directed and undirected likelihoods are the same!
- Trees are in the exponential family with \( y_i \) as sufficient statistics.

Structure Learning

- What about the tree structure (links)?
  How do we know which nodes to make parents of which?

- Bold idea: how can we also learn the optimal structure?
  In principle, we could search all combinatorial structures, for each compute the ML parameters, and take the best one.
- But is there a better way? Yes. It turns out that structure learning in tree models can be converted to a good old computer science problem: maximum weight spanning tree.

Maximum Likelihood Parameters Given Structure

- Trees are just a special case of fully observed graphical models.
- For discrete data \( x_i \) with values \( v_j \), each node stores a conditional probability table (CPT) over its values given its parent.
  The ML parameter estimates are just the empirical histograms of each node’s values given its parent:
  \[ p^*(x_i = v_i|x_{\pi_i} = v_j) = \frac{N(x_i = v_i, x_{\pi_i} = v_j)}{\sum_{v_i} N(x_i = v_i, x_{\pi_i} = v_j)} = \frac{N_i(y_i)}{N_{\pi_i}(v_j)} \]
  except for the root which uses marginal counts \( N_r(v_r)/N \).
- For continuous data, the most common model is a two-dimensional Gaussian at each node. The ML parameters are just to set the mean of \( p_i(y_i) \) to be the sample mean of \( [x_i; x_{\pi_i}] \) and the covariance matrix to the sample covariance.
- In practice we should use some kind of smoothing/regularization.

Optimal Structure

- Let us rewrite the likelihood function:
  \[ \ell(\theta; D) = \sum_{x \in V_{all}} N(x) \log p(x) \]
  \[ = \sum_x N(x) \left( \log p(x_r) + \sum_{i \neq r} \log p(x_i|x_{\pi_i}) \right) \]
  - ML parameters, are equal to the observed frequency counts \( q(\cdot) \):
    \[ \ell^* = \sum_{x \in V_{all}} q(x) \left( \log q(x_r) + \sum_{i \neq r} \log \frac{q(x_i, x_{\pi_i})}{q(x_{\pi_i})} \right) \]
    \[ = \sum_x q(x) \sum_{i \neq r} \log \frac{q(x_i, x_{\pi_i})}{q(x_{\pi_i})} + \sum_x q(x) \sum_i \log q(x_i) \]
  - NB: second term does not depend on structure.
**Edge Weights**

- Each term in sum $i \neq r$ corresponds to an edge from $i$ to its parent.
  
  $$
  \ell^r = \sum_x q(x) \sum_{i \neq r} \log \frac{q(x_i, x_r)}{q(x_i)q(x_r)} + C
  = \sum_{i \neq r} \sum_{x_i, x_r} q(x_i, x_r) \log \frac{q(x_i, x_r)}{q(x_i)q(x_r)} + C
  = \sum_{i \neq r} \sum_{y_i} q(y_i) \log \frac{q(y_i)}{q(x_i)q(x_r)} + C
  = \sum_{i \neq r} W(i; \pi_i) + C
  $$

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

  $$
  W(i; j) = \sum_{x_i, x_j} q(x_i, x_j) \log \frac{q(x_i, x_j)}{q(x_i)q(x_j)}
  $$

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

**Maximum Likelihood Trees**

We can now completely solve the tree learning problem:

1. Compute the marginal counts $q(x_i)$ for each node
   and pairwise counts $q(x_i, x_j)$ for all pairs of nodes.
2. Set the weights to the mutual informations:

   $$
   W(i; j) = \sum_{x_i, x_j} q(x_i, x_j) \log \frac{q(x_i, x_j)}{q(x_i)q(x_j)}
   $$

3. Find the maximum weight spanning tree $A = \text{MWST}(W)$.
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:

   $$
   p(x_i | x_{\pi_i}) = \frac{q(x_i, x_{\pi_i})}{\sum_{x_i} q(x_i, x_{\pi_i})} = \frac{q(x_i, x_{\pi_i})}{q(x_{\pi_i})}
   $$

**Kruskal’s algorithm (Greedy Search)**

- To find the maximum weight spanning tree $A$ on a graph with
  nodes $U$ and weighted edges $E$:

  1. $A \leftarrow$ empty
  2. Sort edges $E$ by nonincreasing weight: $e_1, e_2, \ldots, e_K$.
  3. for $k = 1$ to $K$ { $A \leftarrow A + e_k$ 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.

- Amazingly, as far as likelihood goes, 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
  (cross-entropy) between a distribution and the product of its
  marginals. Measures how far from independent the joint
  distribution is.

  $$
  W(i; j) = I[x_i; x_j] = \text{KL}[q(x_i, x_j) || q(x_i)q(x_j)]
  $$