Lecture 7:

Fully Observed Trees

Sam Roweis

January 27, 2003

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.

$$p(\mathbf{x}) = \frac{1}{Z}\prod_i \psi_i(x_i, x_{\pi_i})$$

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

- 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).

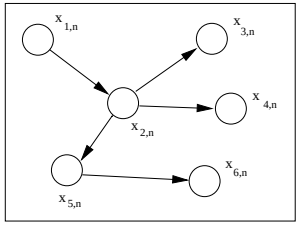
Directed Tree Graphical Models

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

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

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

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



Likelihood function

- Notation:
  $\mathbf{y}_i \equiv$ a node x_i and its single parent x_{π_i}.
  $\mathbf{V}_i \equiv$ set of joint configurations of node i and its parent x_{π_i}
  ($\mathbf{y}_{\text{root}} \equiv x_{\text{root}}$ and $\mathbf{V}_{\text{root}} \equiv \mathbf{v}_{\text{root}}$)
- Directed model likelihood:

$$\begin{aligned}
\ell(\theta; \mathcal{D}) &= \sum_n \log p(\mathbf{x}^n) = \sum_n \left[\log p_r(x_r^n) + \sum_{i \neq r} \log p({x_i}^n | {x_{\pi_i}}^n) \right] \\
&= \sum_n \sum_i \sum_{\mathbf{v} \in \mathbf{V}_i} [\mathbf{y}_i^n = \mathbf{y}] \log p_i(\mathbf{y}) \text{indicator trick} \\
&= \sum_i \sum_{\mathbf{v} \in \mathbf{V}_i} N_i(\mathbf{y}) \log p_i(\mathbf{y})
\end{aligned}$$

where $N_i(\mathbf{y}) = \sum_n [\mathbf{y}_i^n = \mathbf{y}]$ and $p_i(\mathbf{y}_i) = p(x_i|x_{\pi_i})$.

More on the Likelihood function

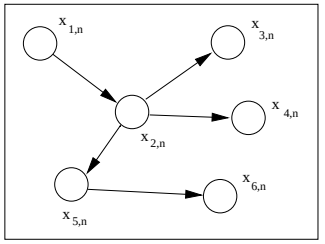
- Undirected model likelihood:

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

  where $N_i(\mathbf{y}) = \sum_n [\mathbf{y}_i^n = \mathbf{y}]$ and $\psi_i(\mathbf{y}_i) = p(x_i|x_{\pi_i})$.
  (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 $\mathbf{y}_i$ 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_i, each node stores a conditional probability table (CPT) over its values given its parent's value.
  The ML parameter estimates are just the empirical histograms of each node's values given its parent:

$$p^*(x_i = v_i|x_{\pi_i} = v_j) = \frac{N(x_i = v_i, x_{\pi_i} = v_j)}{\sum_{\mathbf{v}_i} N(x_i = v_i, x_{\pi_i} = v_j)} = \frac{N_i(\mathbf{y}_i)}{N_{\pi_i}(v_j)}$$

  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 $p_i(\mathbf{y}_i)$ 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:

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

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

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

- NB: second term does not depend on structure.

Edge Weights

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

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

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

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

- 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:

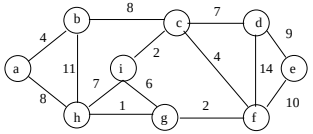
$$W(i;j) = \sum_{x_i, x_j} q(x_i, x_j) \log \frac{q(x_i, x_j)}{q(x_i) q(x_j)}$$

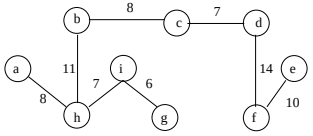
3. Find the maximum weight spanning tree A=MWST(W).
4. Using the undirected tree A chosen by MWST, pick a root arbitrarily and orient the edges away from the root. Set the conditional functions to the observed frequencies:

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

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 $\{A \mathrel{+}= e_k$ unless doing so creates a cycle$\}$

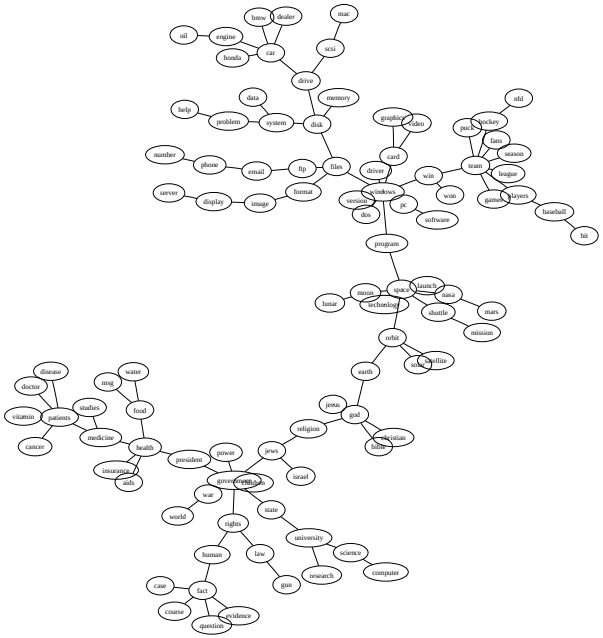




Notes

- *Any* directed tree consistent with the undirected tree found by the algorithm above will assign the same likelihood to any dataset.
- 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.

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


bmw
dealer
msrc
oil
engine
scsi
honda
car
drive
data
memory
help
problem
system
disk
graphics
video
ahl
hockey
pucks
nhl
card
fans
season
number
phone
email
ftp
files
driver
team
league
server
display
image
format
win
windows
won
games
players
baseball
version
pc
dos
software
ku
program
nasa
launch
moon
space
data
lunar
technology
shuttle
mars
mission
orbit
shuttle
earth
disease
doctor
water
msg
studies
food
vitamin
patients
medicine
cancer
health
insurance
aids
president
power
government
war
world
rights
human
law
case
fact
course
evidence
question
jesus
god
religion
bible
christian
jews
israel
state
university
science
research
computer
gun