EE 229A Homework 6

Shane Barratt (24297112)

November 23, 2016

1 Estimating a dependence tree

The goal of this problem was to demonstrate usage of the Chow-Liu [2] algorithm and the minimax estimators which were discussed in class. Given samples of a discrete multivariate probability density function, the Chow-Liu algorithm attempts to estimate dependences which factor its probability density function into the following form while minimizing the divergence between the 2nd-order pdf and the true pdf:

$$p(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} p(x_i | x_{j(i)})$$

1.1 The Dataset

This dataset is the '1984 United States Congressional Voting Records Database' and includes votes and abstains for 16 policy positions from each of the U.S. House of Representatives Congressmen as well as their political party (Democratic or Republican). It was retrieved from the UCI Machine Learning Data Repository. [1]

1.2 The Algorithm

The Chow-Liu Algorithm estimates a 2nd-order dependence tree from samples of a probability density function. Below is the cost function of a given 2nd-order dependence tree. I used the given MATLAB implementation. [4]

$$D(P||P_t) = -\sum_{i=1}^{n} I(x_i, x_{j(i)}) + \sum_{i=1}^{n} H(x_i) - H(\vec{x})$$

The Minimax and MLE Estimators estimate mutual information from samples of the probability density function, and the estimates are used as edge weights to the Chow-Liu Algorithm. [3] Figure 1 demonstrates some properties of the minimax estimator for entropy. I used the given MATLAB implementation for both estimators. The following is the equation for the MLE estimator:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i} + \sum_{i=1}^{n} q_i \log \frac{1}{q_i} - \sum_{i=1}^{n} z_i \log \frac{1}{z_i},$$

where p is the empirical distribution for x, q is the empirical distribution for y and z is the empirical distribution for (x, y).

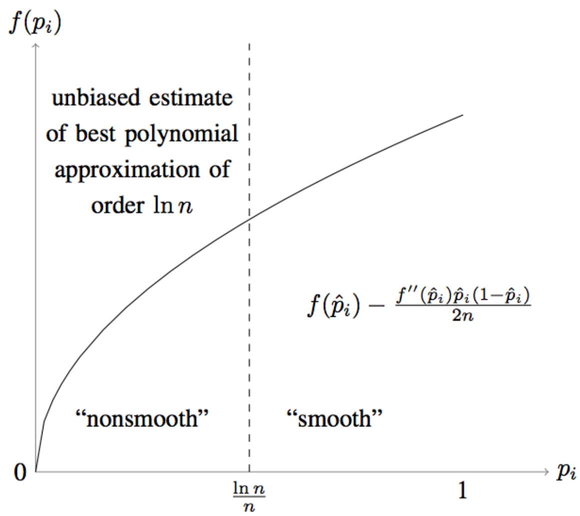

Figure 1: Illustration of the Minimax estimator for a function that has a similar form to entropy.

1.3 Results

The final sum of Mutual Informations (MI) of the MLE Dependence tree was 5.6518 and the final sum of MIs for the JVHW Dependence tree was 5.5753. The resulting dependence tree is displayed in Figure 2. The root of the tree is 0, which happens to be the party attribute (Democratic vs. Republican). Figure 3 is a plot of the fraction of edges in agreement between the MLE and JVHW Dependence Trees as a function of the number of samples used to estimate them. After around 200 samples, they essentially converge to the same graph. The benefit of deriving such a bayesian network is that this complex distribution can be described succintly. Also, inference and belief propagation algorithms can be applied to answer questions about the network and create intelligent systems.

It seems like nodes 4 and 5 are central to the rest of the network. These questions are "Physician Fee Freeze" and "Giving Aid to El Salvador". In other words, given a house rep's view on one of these two topics, their answer to questions 12/3/15/1 and 6/9/14/13/8 are independent. Also, the fact that 0 (dem/rep) is the root of the tree makes sense, as all of their views likely stem from this key dimension of the probability density function.

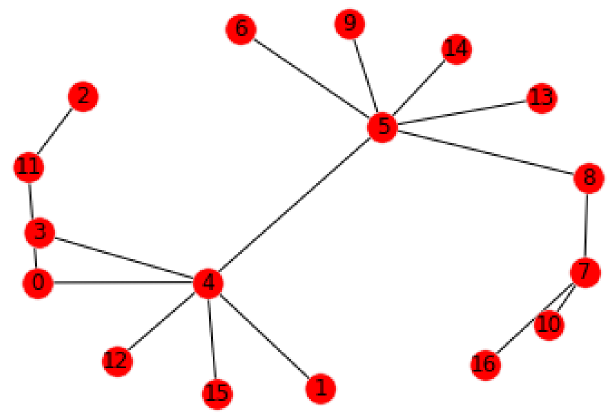

Figure 2: Estimated dependence tree for 1984 Congressional Voting Data

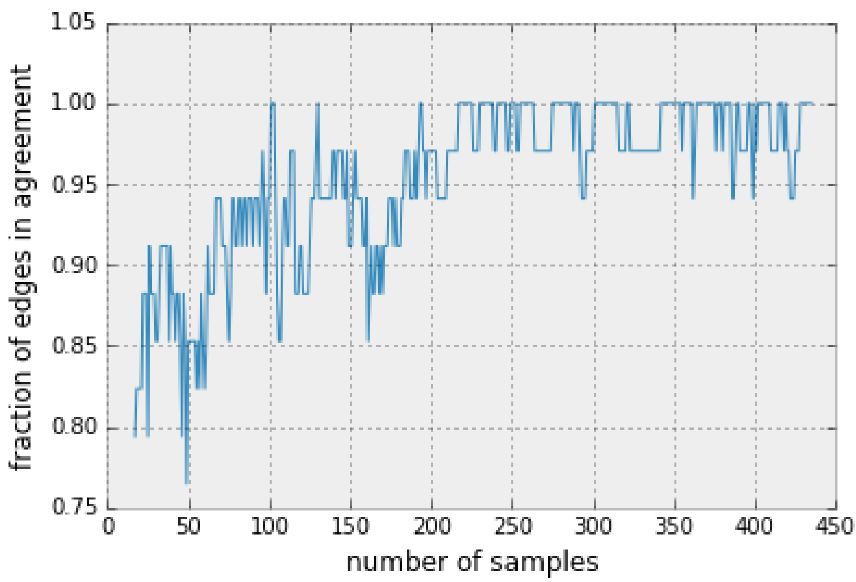

Figure 3: Fraction of edges in agreement between the resulting graphs of Chow-Liu's algorithm using MLE and Minimax Mutual Information Estimators as a function of the number of samples taken.

2 Computation of Channel Capacity

(a) Code:

The code is attached at the end of this document, as an IPython notebook exported as a PDF.

(b) Result:

$$Q_{x|y} = \begin{bmatrix} 0 & 0 & 0 \\ 0.906 & 0 & 0.177 \\ 0.094 & 1.00 & 0.823 \end{bmatrix}$$

$$r_x = \begin{bmatrix} 0 \\ 0.518 \\ 0.482 \end{bmatrix}$$

(c) Proof.

i) Choose r^1 arbitrarily. Then do ii and iii for $t = 1, 2, \cdots$

ii) Solve $C(t,t)$ and note $q^t(x|y)$

$$C(t,t) = \max_{q(x|y)} \sum_x \sum_y r^t(x) p(y|x) \log \frac{q(x|y)}{r^t(x)}$$

$$q^t(x|y) = \frac{r^t(x)p(y|x)}{\sum_y r^t(x)p(y|x)}$$

iii) Solve $C(t+1,t)$ and note $r^{t+1}(x)$

$$C(t+1,t) = \max_{r(x)} \sum_x \sum_y r(x) p(y|x) \log \frac{q^t(x|y)}{r(x)}$$

$$r^{t+1}(x) = \frac{\prod_y q^t(x|y)^{p(y|x)}}{\sum_{x'} \prod_y q^t(x'|y)^{p(y|x')}}$$

$$\sum_x r^0(x) \log \frac{r^{t+1}(x)}{r^t(x)} = \sum_x r^0(x) \log\{\frac{1}{r^t(x)} \frac{\prod_y q^t(x|y)^{p(y|x)}}{\sum_{x'} \prod_y q^t(x'|y)^{p(y|x')}}\}$$

$$= -C(t+1,t) + \sum_x r^0(x) \sum_y p(y|x) \log \frac{p(y|x)}{r^t(y)}$$

$$= -C(t+1,t) + \sum_x \sum_y r^0(x) p(y|x) [\log \frac{p(y|x) r^0(x)}{r^0(y) r^0(x)} + \log \frac{r^0(y)}{r^t(y)}]$$

$$= -C(t+1,t) + C(P) + \sum_y r^0(y) \log \frac{r^0(y)}{r^t(y)}$$

$$\geq C(P) - C(t+1,t)$$

Thus,

$$C(P) - C(t+1,t) \leq \sum_x r^0(x) \log \frac{r^{t+1}(x)}{r^t(x)}$$

$$\sum_{t=1}^{N} [C(P) - C(t+1,t)] \leq \sum_t \sum_x r^0(x) \log \frac{r^{t+1}(x)}{r^t(x)}$$

$$\leq \sum_x r^0(x) \log \frac{r^{N+1}(x)}{r^1(x)}$$

$$\leq \sum_x r^0(x) \log \frac{r^0(x)}{r^1(x)}$$

$$= D(r^0||r^1)$$

Because the sum of the error terms is finite and $C(t+1,t) \leq C(P)$ by definition, and thus C_k converges.

(d) Proof.

Assume each term $C(P) - C_k \geq \frac{Z}{k}$. Then the sum will diverge because $\sum \frac{1}{k}$ diverges. Thus, because $\sum_{k=1}^{N} C(P) - C_k$ converges to a finite value, each term must be $\leq \frac{Z}{k}$ for some Z. Thus, the k-th error term is proportional to k^{-1}.

References

[1] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Chow, C.K. and Liu, C.N. (1968). Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, 14, 462-467.

[3] Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. IEEE Transactions on Information Theory, 61(5):28352885, 2015.

[4] Li, Guangdi. "Maximum Weight Spanning Tree (Undirected) - File Exchange - MATLAB Central." File Exchange - MATLAB Central. N.p., 2009. Web. 22 Nov. 2016.