10-708: Probabilistic Graphical Models 10-708, Spring 2014

7 : Learning in Fully Observed Bayesian Networks

Lecturer: Eric P. Xing *Scribes: Diyi Yang, Zichao Yang*

1 ML Structural Learning for Completely Observed GMs

The goal of learning GMs is, finding the best Bayesian Network given the set of independent samples (assignments of random variables). Here, learning represents the process of estimating the parameters or the topology of the network from data. Two classes of algorithms for guaranteed structure learning that only apply to certain families of graphs will be introduced below.

1.1 Information Theoretic Interpretation of ML

For the Maximum Likelihood Method, one can understand change the log likelihood from sum over data points to sum over count of variable states.

$$\begin{aligned}
l(\theta_G, G; D) &= log\ p(D|\theta_G, G) \qquad (Joint\ Likelihood) \\
&= log \prod_n (\prod_i p(x_{n,i}|\mathbf{x}_{n,\pi_{i(G)}}, \theta_{i|\pi_{i(G)}})) \qquad (BNFactorization\ Rule) \\
&= \sum_i (\sum_n log\ p(x_{n,i}|\mathbf{x}_{n,\pi_{i(G)}}, \theta_{i|\pi_{i(G)}})) \qquad (Exchange\ Order\ of\ n\ and\ i) \\
&= M \sum_i (\sum_{x_i, \mathbf{x}_{\pi_{i(G)}}} \frac{count(x_i, \mathbf{x}_{\pi_{i(G)}})}{M} log\ p(x_i|\mathbf{x}_{\pi_{i(G)}}, \theta_{i|\pi_{(G)}})) \\
&= M \sum_i (\sum_{x_i, \mathbf{x}_{\pi_{i(G)}}} \hat{p}(x_i, \mathbf{x}_{\pi_{i(G)}}) log\ p(x_i|\mathbf{x}_{\pi_{i(G)}}, \theta_{i|\pi_{(G)}}))
\end{aligned}$$

Here, M is the number of variable states, and the probability of $p(x_i)$ is replaced by using the count function. $(x_i, \mathbf{x}_{\pi_{i(G)}})$ takes the possible values of random variables.

$$\begin{aligned}
l(\theta_G, G; D) &= M \sum_i (\sum_{x_i, \mathbf{x}_{\pi_{i(G)}}} \hat{p}(x_i, \mathbf{x}_{\pi_{i(G)}}) log\ \hat{p}(x_i|\mathbf{x}_{\pi_{i(G)}}, \theta_{i|\pi_{(G)}})) \\
&= M \sum_i (\sum_{x_i, \mathbf{x}_{\pi_{i(G)}}} \hat{p}(x_i, \mathbf{x}_{\pi_{i(G)}}) log\ \frac{\hat{p}(x_i, \mathbf{x}_{\pi_{i(G)}}|\theta_{i|\pi_{(G)}})}{\hat{p}(\mathbf{x}_{\pi_{i(G)}})} \frac{\hat{p}(x_i)}{\hat{p}(x_i)}) \\
&= M \sum_i (\sum_{x_i, \mathbf{x}_{\pi_{i(G)}}} \hat{p}(x_i, \mathbf{x}_{\pi_{i(G)}}) log\ \frac{\hat{p}(x_i, \mathbf{x}_{\pi_{i(G)}}, \theta_{i|\pi_{(G)}})}{\hat{p}(\mathbf{x}_{\pi_{i(G)}})\hat{p}(x_i)}) - M \sum_i \sum_{x_i} -\hat{p}(x_i) log\ \hat{p}(x_i) \\
&= M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_{i(G)}}) - M \sum_i \hat{H}(x_i)
\end{aligned} \tag{1}$$

If we replace all the probabilities with empirical probabilities as above, we get the Equation 1. The maximum log likelihood function is divided into two parts, including the mutual information of all nodes as $M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_{i(G)}})$, and the entropy of each random variable. Thus, the conclusion is that, if we could have a structure that each node has only one parent node (tree structure), then Equation 1 will enable us to find an exact solution of an optimal tree under MLE. Two tricks here include MLE score decomposable to edge-related elements and each node has only one parent.

2 Chow-Liu Algorithm

The objective function could be written as

$$l(\theta_G, G; D) = M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_{i(G)}}) - M \sum_i \hat{H}(x_i)$$

Considering that we only care about tree structure, we could make some reduction and have the following score function:

$$\begin{aligned} C(G) &= M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_{i(G)}}) \\ &= M \sum_i \hat{I}(x_i, x_j) \end{aligned}$$

This means we only need to compute the empirical distribution and mutual information of any pair of nodes. It can be pre-calculated, and empirical things can be directly counted from data. Therefore, for each pair of variable x_i and x_j, compute the empirical distribution and mutual information.

$$\begin{aligned} \hat{p}(X_i, X_j) &= \frac{count(x_i, x_j)}{M} \\ \hat{I}(X_i, X_j) &= \sum_{x_i, x_j} \hat{p}(x_i, x_j) log \ \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)} \end{aligned}$$

Then we can define a graph with nodes $x_1, x_2, x_3 ... x_n$ by assigning edge (i, j) to weight $\hat{I}(X_i, X_j)$. The Chow-Liu algorithm will compute maximum weight spanning tree. One can pick any node as root, and do breadth-first-search to define directions.

3 Parameter Learning for Completely Observed Graph of Given Structure

Assume graph G is known and fixed, our goal is to estimate from a dataset of N independent, identically distributed(iid) training cases $D = \{x_1, x_2, x_3 ..., x_N\}$. In general, each training case $x_n = x_{n,1}, x_{n,2} ... x_{n,M}$ is a vector of M values, one per node. To begin with, we firstly give several widely used distributions in parameter learning.

3.1 Multinomial Model Example

The observed N iid data could be represented as a unit basis vectors, $x_n = (x_{n,1}, x_{n,2}...x_{n,K})^T$. Here, $x_{n,k} = \{0, 1\}$ and $\sum_{k=1}^{K} x_{n,k} = 1$. The likelihood of dataset $D = x_1, x_2...x_N$ is:

$$L(\theta|D) = P(x_1, x_2...x_N|\theta) = \prod_{n=1}^{N} P(x_n|\theta) = \prod_k {\theta_k}^{n_k}$$
$$l(\theta|D) = log \prod_k {\theta_k}^{n_k} = \sum_k n_k log\ \theta_k$$

Maximize the log likelihood objective function $l(\theta|D)$ subject to the constrain $\sum_{k=1}^{K} x_{n,k} = 1$ by adding a Lagrange multiplier as follows:

$$\bar{l}(\theta|D) = \sum_k n_k log\ \theta_k + \lambda(1 - \sum_{k=1}^{K} x_{n,k})$$

Taking the derivatives wrt θ_k, we get

$$\widehat{\theta}_{k,MLE} = \frac{n_k}{N} \quad or$$
$$\widehat{\theta}_{k,MLE} = \frac{1}{N} \sum_n x_{n,k}$$

Besides, the counts, $\bar{n} = (n_1, n_2, ...n_K)$, $n_k = \sum_n x_{n,k}$ are sufficient statistics of data D.

3.2 Bayesian Parameter Estimation

Sometimes, one might have prior knowledge about those parameters. If such priors exist and if it could be represented as a probability distribution over the space of models, then one could use Bayesian Estimation to deal with that. One important characteristic of Bayesian model is the posterior distribution over models. This could be formulated by Bayes Rule. To begin with, we introduce two related priors.

3.2.1 Dirichlet Prior

Dirichlet prior is defined by using a set of hyper parameters $\alpha_1, \alpha_2...\alpha_N$. The Dirichlet distribution is

$$P(\theta) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\sum_k \alpha_k)} \prod_k {\theta_k}^{\alpha_k - 1} = C(\alpha) \prod_k {\theta_k}^{\alpha_k - 1}$$

Here, $C(\alpha)$ is the normalize constant. The posterior distribution can be written as follows:

$$P(\theta|x_1, x_2...x_N) = \frac{P(x_1, x_2...x_N|\theta)P(\theta)}{P(x_1, x_2...x_N)} \propto \prod_k {\theta_k}^{\alpha_k + n_k - 1}$$

Such a prior is called a conjugate prior for there is isomorphism of the posterior to the prior. This means, if we use a Dirichlet prior, then posterior is also a Dirichlet distribution. Based on this property, we can have

the sequential Bayesian updating algorithm. That is, start with Dirichlet prior $P(\overrightarrow{\theta}|\overrightarrow{\alpha}) = Dir(\overrightarrow{\theta}|\overrightarrow{\alpha})$, and the posterior becomes $P(\overrightarrow{\theta}|\overrightarrow{\alpha}, \overrightarrow{n'}) = Dir(\overrightarrow{\theta}|\overrightarrow{\alpha} + \overrightarrow{n'})$ after we observe N' samples with sufficient statistics $\overrightarrow{n'}$; The posterior changes to

$$P(\overrightarrow{\theta}|\overrightarrow{\alpha}, \overrightarrow{n'}, \overrightarrow{n''}) = Dir(\overrightarrow{\theta}|\overrightarrow{\alpha} + \overrightarrow{n'} + \overrightarrow{n''})$$

after observing another N'' data samples with sufficient statistics $\overrightarrow{n''}$. Thus sequentially absorbing data in any order is equivalent to batch update. One limitation of Dirichlet prior is that it has a center or focus on only one dimension. That is, it will fail if you need prior that could exhibit two dimensions, which leads to the logistic normal prior.

3.2.2 Logistic Normal Prior

The logistic normal prior gives richer distribution definitions than Dirichlet, and is defined as follows.

$$\begin{aligned} \theta &\sim LN_K(\mu, \Sigma) \\ \gamma &\sim N_{K-1}(\mu, \Sigma) \qquad \gamma_K = 0 \\ \theta_i &\sim \left\{ \gamma_i - log(1 + \sum_{i=1}^{K-1} e^{\gamma_i}) \right\} \end{aligned}$$

$$C(\gamma) = log(1 + \sum_{i=1}^{K-1} e^{\gamma_i}) \qquad Log\ Partition\ Function$$

The pro of this logistic normal prior is that it could better capture the co-variance structure, and the con is it is non-conjugate.

3.2.3 Parameter estimation for Multivariate Gaussian

The pdf of a multivariate Gaussian distribution is given by

$$p(X; \mu, \Sigma) = \frac{1}{\sqrt{2\pi}^{n/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\}.$$

It can be shown that the MLE for μ and Σ is

$$\begin{aligned} \mu_{MLE} &= \frac{1}{N}\sum_n x_n \\ \Sigma_{MLE} &= \frac{1}{N}\sum_n (x_n - \mu_{MLE})(x_n - \mu_{MLE})^T \end{aligned}$$

Note that Σ_{MLE} may not be full rank and hence is not invertible. This is one reason why we want to pursue a Bayesian approach in estimating the parameters. There are many other as well, for example, we would like to update the estimates sequentially or we may have some prior knowledge about parameters.

Specifically, we restrict our attention to conjugate priors and consider the case of unknow μ and known σ. The norm prior of μ is given by

$$p(\mu) = (2\pi\tau^2)^{-1/2} \exp\{-(\mu - \mu_0)^2/2\tau^2\}.$$

The joint distribution is

$$P(x,\mu) = (2\pi\tau^2)^{-1/2}\exp\{-(\mu-\mu_0)^2/2\tau^2\} * (2\pi\sigma^2)^{-N/2}\exp\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\}$$

The posterior is

$$P(\mu|X) = (2\pi\tilde{\sigma})^{-1/2}\exp\{-(\mu-\tilde{\mu})^2/2\tilde{\sigma}^2\},$$

where $\tilde{\mu} = \frac{N/\sigma^2}{N/\sigma^2+1/\tau}\bar{x} + \frac{1/\tau}{N/\sigma^2+1/\tau}\mu_0$ and $\tilde{\sigma}^2 = (\frac{N}{\sigma^2}+\frac{1}{\tau^2})^{-1}$.

The posterior mean is the convex combination of the prior and the MLE, with weights proportional to the relative noise levels. The precision of the posterior $1/\tilde{\sigma}^2$ is the precision of the prior $1/\sigma_0^2$ plus one contribution of data precision $1/\sigma^2$ for each observed data point.

3.3 MLE for general Bayesian Networks

If we assume that the parameters for each CPD are globally independent, and all nodes are fully observed, then the log-likelihood function decomposes into a sum of local terms, one per node:

$$l(\theta;D) = \log p(D|\theta) = \sum_i(\sum_n \log p(x_{n,i}|x_{n,\pi_i},\theta_i)).$$

A plate is a "macro" that allows subgraphs to be replicated. For i.i.d data, the likelihood is

$$p(D|\theta) = \Pi_n p(x_n|\theta).$$

We can represent a plate a Bayes net with N nodes.

3.3.1 MLE for BN with tabular CPDs

Assume each CPD is represented as a table where $\theta_{ijk} = p(X_i = j|X_{\pi_i} = k)$, then sufficient statistics are counts of family configurations $n_{ijk} = \sum_n x_{n,i}^j x_{n,\pi_i}^k$. The log-likelihood is

$$l(\theta;D) = log\Pi_{i,j,k}\theta_{ijk}^{n_{ijk}} = \sum_{i,j,k} n_{i,j,k}\log\theta_{i,j,k}$$

Using a Lagrange multiplier to enforce $\sum_j \theta_{ijk} = 1$, we get

$$\theta_{ijk}^{ML} = \frac{n_{ijk}}{\sum_{j'} n_{ij'k}}.$$

3.4 Parameter Priors For Bayesian estimation

How to define parameter prior for a bayesian network? Recall that $p(X=x) = \Pi_{i=1}^M p(x_i|x_{\pi_i})$ and the local distribution is given by $p(x_i^k|x_{\pi_i}^j) = \theta_{x_i^k|x_{\pi_i}^j}$. There are two kinds of independence properties that make the parameters estimation easy: global independence and local independence. Global independence: $p(\theta_m|G) = \Pi_{i=1}^M p(\theta_i|G)$. Local independence: $p(\theta_i|G) = \Pi_{j=1}^{q_i} p(\theta_{x_i^k|x_{\pi_i}^j}|G)$. By assuming the two kinds of independence, we can perform Bayesian update each parameter independently provided all variables are observed in all cases.

There are two types of distribution that satisfy the independence properties:

- Discrete DAG models with $x_i|\pi_{x_i}^j \sim \text{Multi}(\theta)$ with Dirichlet prior $p(\theta) = C(\alpha)\Pi_k \theta_k^{\alpha_k - 1}$
- Gaussian DAG models with $x_i|\pi_{x_i}^j \sim \text{Normal}(\mu, \Sigma)$. The Normal-Wishart prior is given by

$$\begin{aligned} p(\mu|v, \alpha_\mu, W) &= \text{Normal}(v, (\alpha_\mu W)^{-1}) \\ p(W|\alpha_w, T) &= c(n, \alpha_w)|T|^{\alpha_w/2}|W|^{(\alpha_w - n - 1)/2} \exp\{\frac{1}{2}\text{tr}\{TW\}\}, \end{aligned}$$

  where $W = \Sigma^{-1}$.

3.5 Learning a Markov Chain Transition Matrix

Consider a time-invariant 1-order Markov model, the initial state probability vectoris $\pi_k = p(X_1^k = 1)$ and the state transition probability matrix $A_{ij} = p(X_t^j = 1|X_{t-1}^i = 1)$. Then the joint probability is given by

$$p(X_{1:T}|\theta) = p(x_1|\pi)\Pi_{t=2}^T \Pi_{t=2} p(X_t|X_{t-1})$$

The log-likelihood function is given by

$$l(\theta; D) = \sum_n \log p(x_{n,1}|\pi) + \sum_n \sum_{t=2}^T \log p(x_{n,t}|x_{n,t-1,A})$$

A is a stochastic matrix $\sum_j A_{ij} = 1$, so the MLE of A_{ij} is the fraction of transition from i to j

$$A_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=2}^T x_{n,t-1}^i x_{n,t}^j}{\sum_n \sum_{t=2}^T x_{n,t-1}^i}$$

There is a sparse data problem with the above method: if $i \to j$ did not occur, we will have $A_{ij} = 0$, then any future sequence with word pair $i \to j$ will have zero probability. A standard hack is to use backoff smoothing or deleted interpolation

$$\tilde{A}_{i \to \bullet} = \lambda \eta_t + (1 - \lambda) A_{i \to \bullet}^{ML}$$

3.6 Example: HMM

There are two types of learning for HMM: supervised learning and unsupervised learning. Supervised learning means we do the estimation when the right answer is known.

Recall that for HMM,

- The transition probabilities between any two states is $p(y_t^j = 1|y_{t-1}^i = 1) = a_{i,j}$ or $p(y_t|y_{t-1} = 1) \sim \text{Multinomial}(a_{i,1}, a_{i,2}, ..., a_{i,M})$.
- Start probabilities is given by $p(y_1) \sim \text{Multinomial}(\pi_1, \pi_2, ..., \pi_M)$.
- Emission probabilities associated with each state $p(x_t|y_t^i = 1) \sim \text{Multinomial}(b_{i,1}, b_{i,2}, ..., b_{i,K})$

Given $x = x_1...x_N$ for which the true state path $y = y_1...y_N$ is known, define

$$A_{ij} = \#\text{times state transition } i \to j \text{ occurs in } y$$

$$B_{ik} = \#\text{times state } i \text{ in } y \text{ emits } k \text{ in } x$$

We can show that the maximum likelihood parameters θ are:

$$a_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{A_{ij}}{\sum_j A_{ij}}$$

$$b_{ik}^{ML} = \frac{\#(i \to k)}{\#(i \to \bullet)} = \frac{B_{ij}}{\sum_k B_{ik}}$$

Solution for small training sets

$$A_{ij} = \#\text{times state transition } i \to j \text{ occurs in } y + R_{ij}$$

$$B_{ik} = \#\text{times state } i \text{ in } y \text{ emits } k \text{ in } x + S_{ik}$$

R_{ij}, S_{ij} are pseudocounts representing our prior belief.