10-708: Probabilistic Graphical Models, Spring 2015

4 : Parameter Estimation in Fully Observed BNs

Lecturer: Eric P. Xing *Scribes: How Jing and Xiaoqiu Huang*

1 Learning Graphical Models

The goal of learning graphical models is to discover the best Bayesian Network given a set of independent samples. The learning process consists of learning the structure and the parameters of graphical model from data samples. These two learning tasks will be detailed discussed in the following sections.

2 ML Structural Learning for Fully Observed GMs

There are two optimal approaches which can guarantee to return a structure that maximizes the objectives. The first approach which assumes that the structure of graphical model is a tree is called the Chow-Liu algorithm. And we will only cover this algorithm in this section.

2.1 Information Theoretic Interpretation of ML

In order to maximize the likelihood function, we can change the interpretation from the sum over data points to the sum over count of variable states.

$$
\begin{aligned}
l(\theta_G, G; D) &= \log p(D|\theta_G, G) \quad ----- \ likelihood\ function \\
&= \log \prod_n (\prod_i p(x_{n,i}|\mathbf{x}_{\mathbf{n},\pi_\mathbf{i}(\mathbf{G})}, \theta_{\mathbf{i}|\pi_\mathbf{i}(\mathbf{G})})) \quad ---- \ factorization\ rule \\
&= \sum_i (\sum_n \log p(x_{n,i}|\mathbf{x}_{\mathbf{n},\pi_\mathbf{i}(\mathbf{G})}, \theta_{\mathbf{i}|\pi_\mathbf{i}(\mathbf{G})})) \\
&= M \sum_i (\sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \frac{count(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})}{M} \log p(x_i|\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{\mathbf{i}|\pi_\mathbf{i}(\mathbf{G})})) \quad -- \ counts\ of\ configurations \\
&= M \sum_i (\sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log \mathbf{p}(\mathbf{x_i}|\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G}), \theta_{\mathbf{i}|\pi_\mathbf{i}(\mathbf{G})}})) \quad ---- \ \textbf{empirical probability}
\end{aligned}
$$

Here, M denotes the number of data samples. $\pi_i(G)$ denotes the parents of node i in graph G. $count(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})$ denotes the number of certain configurations of node i and its parents. $\hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})$ denotes the empirical probability of certain configuration of node i and its parents.

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

After decomposing the function of graph structure, it is easy to observe that the likelihood function consists of the mutual information of nodes with their parents $\hat{I}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})$, and the entropy of each node $\hat{H}(x_i)$. Here, we use two tricks to find exact solution of the graphical model. First, we assume that each node in the graph has only one parent. Second, the MLE score is decomposed to edge-related elements.

2.2 Chow-Liu tree learning algorithm

The objective function in learning graphical model can be written as follows:

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

Since we only consider the structure of tree, the objective function can be reduced to the value of mutual information. Therefore, we only need to maximize following function.

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

Chow-Liu tree learning algorithm consists of mainly three steps:

- For each pair of variable x_i and x_j:
  - Compute empirical distribution: $\hat{p}(X_i, X_j) = \frac{count(x_i, x_j)}{M}$
  - Compute mutual information: $\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)}$
- Define a graph with node $x_1, \cdots, x_n$:
  - Edge (i, j) gets weight $\hat{I}(X_i, X_j)$
- For undirected graphic model, we can compute maximum weight spanning tree from the graph. For directed graphical model, we can pick any node as root and do breadth-first search to define directions.

It is noticed that if the nodes in the graph have at most $d (d \geq 2)$ parents, the problem of learning Bayesian Network structure is NP-hard.

3 ML Parameter Estimation for Completely Observed GMs of Given Structure

Let us now assume that the graph structure is given (likely designed by some domain expert), the goal then is to learn the parameters from a set of i.i.d. data $\{x_1, x_2, ..., x_N\}$ where each data point lies on some M dimensional space, that is, $x_i \in R^M$.

3.1 Multinomial Model

We now show how to learn the parameters in a Multinomial model, which is widely used for problems involved discrete counts. Let the observed dataset be i.i.d. samples from some unknown multnomial distribution where θ_k is the probability that the observed data point belongs to the kth event, then we have likelihood function as follows,

$$\begin{aligned} L(X;\theta) &= \prod_{i=1}^{N}\prod_{k=1}^{M} \theta_k^{x_{i,k}} \\ &= \prod_{k=1}^{M} \theta_k^{\sum_{i=1}^{N} x_{i,k}} \\ &= \prod_{k=1}^{M} \theta_k^{n_k} \end{aligned}$$

where n_k is the total count for even k in the entire dataset. Now the define the objective function to be the log-likelihood of the data,

$$\begin{aligned} l(X;\theta) &= \sum_{k=1}^{M} \log \theta_k^{n_k} \\ &= \sum_{k=1}^{M} n_k \log \theta_k \end{aligned}$$

To maximize this function, we incorporate Lagrange multiplier to take into account the constraint that $\sum_k \theta_k = 1$. By adding the multiplier, we have the following objective function,

$$J(X;\theta,\lambda) = \sum_{k=1}^{M} n_k \log \theta_k + \lambda(1 - \sum_{k=1}^{M} \theta_k)$$

Take the gradient and set it equal to zero, we get the MLE of θ,

$$\begin{aligned} \frac{\partial J}{\partial \theta_k} = \frac{n_k}{\theta_k} - \lambda &:= 0 \\ \implies \theta_k &= \frac{n_k}{\lambda} \\ \implies \theta_k^* &= \frac{n_k}{N} \end{aligned}$$

where the last equation follows from solving $\frac{\sum_k n_k}{\lambda} = 1$ for λ. This result has a intuitive interpretation that the MLE is simply the empirical frequency of a particular event.

3.2 Bayesian Estimation

To adopt Bayesian framework for parameter estimation, we first need to put a prior distribution on the parameters we want to estimate. A natural choice for discrete count model is the Dirichlet distribution which has the following form,

$$p(\theta;\alpha) = \frac{\Gamma\Big(\sum_k \alpha_k\Big)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1}$$

where Γ is the gamma function: $\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx$ and a sample drawn from a k-dimensional Dirichlet distribution lies on a $k-1$-simplex. We can now compute the posterior distribution of θ given the dataset,

$$\begin{aligned} p(\theta|X) &= \frac{p(\theta;\alpha)p(x_1, x_2, ..., x_N|\theta)}{p(X)} \\ &\propto p(\theta;\alpha)p(x_1, x_2, ..., x_N|\theta) \\ &\propto \prod_k \theta_k^{\alpha_k - 1} \prod_k \theta_k^{n_k} \\ &= \prod_k \theta_k^{n_k + \alpha_k - 1} \end{aligned}$$

We observe that the posterior has the same form a the prior, and such prior is called a conjugate prior. Now we know that the posterior is simply Dirichlet with parameter $\alpha_k + n_k$, we can use the posterior mean as our estimation for the optimal parameters,

$$\begin{aligned} \theta_k &= \int \theta_k p(\theta|D)d\theta \\ &= \frac{\Gamma\Big(\sum_k \alpha_k + n_k\Big)}{\prod_k \Gamma(\alpha_k + n_k)} \int \theta_k \prod_k \theta_k^{\alpha_k + n_k - 1} d\theta \\ &= \frac{n_k + \alpha_k}{N + |\alpha|} \end{aligned}$$

We note that the Dirichlet prior plays a role as pseudo-counts that we have before we observe the data.

3.3 The Logistic Normal Prior

Despite its convenience of being conjugate to the Multinomial, Dirichlet prior ignores the covariance structure that might exist in the data. To capture the covariance structure, we can instead put a logistic normal prior over θ, where we first draw a sample from a multivariate Gaussian:

$$\gamma \sim \mathcal{N}(\mu, \Sigma)$$

then to force it to lie on the simplex, we renormalize:

$$\theta_k = \frac{\exp(\gamma_k)}{\sum_k \exp(\gamma_k)}$$

The advantage of using logistic normal is that we capture the covariance structure with covariance matrix of the multivariate normal. However this ruins the convenience of conjugacy and we no longer have a simple closed form expression of the posterior. To infer the posterior and to make prediction, we have to use other techniques that will be discussed later in the class.

3.4 Continuous Distribution

We have discussed the MLE for discrete counts data using Multinomial models, and how Dirichlet prior can be placed on the top of it. Now we turn our attention into multivariate continuous distribution. To model such data, a natural choice would be to use a multivariate Gaussian distribution with the following form,

$$p(x;\mu,\Sigma) = \frac{1}{2\pi^{M/2}|\Sigma|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\Big)$$

Then we write down the log-likelihood of the data,

$$l(X;\mu,\Sigma) = \sum_{i=1}^{N} -\frac{M}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}\Big((x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\Big)$$

Take the gradient w.r.t. μ and set it equal to zero we have,

$$\begin{aligned}
&\frac{\partial \sum_{i=1}^{N} -\frac{1}{2}\Big(-2\mu^T\Sigma^{-1}x_i + \mu^T\Sigma^{-1}\mu\Big)}{\partial\mu} = 0 \\
&\implies \sum_{i=1}^{N} -\Sigma^{-1}x_i + \Sigma^{-1}\mu = 0 \\
&\implies \sum_{i=1}^{N} x_i = N\mu \\
&\implies \mu^* = \frac{\sum_i^N x_i}{N}
\end{aligned}$$

where we have used the identity of matrix derivative that $\frac{\partial x^T A x}{\partial x} = 2Ax$ for symmetric matrix A. Similarily, we can obtain MLE for the covariance matrix Σ,

$$\begin{aligned}
&\sum_{i=1}^{N} \frac{\partial \log|\Sigma|}{\partial\Sigma} + \frac{\partial x_i^T\Sigma^{-1}x_i - 2\mu^T\Sigma^{-1}x_i + \mu^T\Sigma^{-1}\mu}{\partial\Sigma} = 0 \\
&\implies \sum_{i=1}^{N} \Sigma^{-1} - \Sigma^{-1}x_i x_i^T\Sigma^{-1} + 2\Sigma^{-1}\mu x_i^T\Sigma^{-1} - \Sigma^{-1}\mu\mu^T\Sigma^{-1} = 0
\end{aligned}$$

where we have used the identity that $\frac{\partial \log|X|}{\partial X} = X^{-1}$ and $\frac{\partial a^T X^{-1} b}{\partial X} = -X^{-1}ab^TX^{-1}$ for symmetric matrix X. Now we multiply both side by Σ and obtain

$$\begin{aligned}
&\sum_{i=1}^{N} \Sigma - x_i x_i^T + 2\mu x_i^T - \mu\mu^T = 0 \\
&\implies N\Sigma = \sum_{i=1}^{N} x_i x_i^T - 2\mu x_i^T + \mu\mu^T \\
&\implies \Sigma^* = \frac{(x_i-\mu)(x_i-\mu)}{N}
\end{aligned}$$

where we can plug in μ^* for μ.

3.5 Bayesian parameter estimation for a Gaussian

There are various reasons to pursue a Bayesian approach for Gaussian.

- We would like to update our estimates sequentially over time.
- We may have prior knowledge about the expected magnitude of the parameters.
- The MLE for Σ may not be full rank if we dont have enough data.

In this section, we only discuss the condition where σ is known and μ is unknown. Since the likelihood of Guassian distribution has the form

$$p(x|\mu) = (2\pi\sigma^2)^{-N/2} \exp\{-\sum_{i=1}^{N}(x_i-\mu)^2/2\sigma^2\}$$

the conjugate prior has the form

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

Therefore, the posterior distribution has the form

$$p(\mu|x) \propto (2\pi\sigma^2)^{-N/2} \exp\{-\sum_{i=1}^{N}(x_i-\mu)^2/2\sigma^2\} \times (2\pi\tau^2)^{-1/2} \exp\{-(\mu-\mu_0)^2/2\tau^2\}$$

Since the product of two Gaussians is a Guassian, we can rewrite the posterior in the form

$$\begin{aligned} p(\mu|x) &\propto (2\pi\sigma^2)^{-N/2}(2\pi\tau^2)^{-1/2} \exp\{-\frac{\mu^2}{2}(\frac{1}{\tau^2}+\frac{n}{\sigma^2}) + \mu(\frac{\mu_0}{\tau}+\frac{\sum_{i=1}^{N} x_i}{\sigma^2}) - (\frac{\mu_0^2}{2\tau^2}+\frac{\sum_{i=1}^{N} x_i^2}{2\sigma^2})\} \\ &= (2\pi\widetilde{\sigma}^2)^{-1/2} \exp\{-(\mu^2 - 2\mu\widetilde{\mu}+\widetilde{\mu}^2)/2\widetilde{\sigma}^2\} = (2\pi\widetilde{\sigma}^2)^{-1/2} \exp\{-(\mu-\widetilde{\mu})^2/2\widetilde{\sigma}^2\} \end{aligned}$$

After matching coefficients of μ^2, we can find that

$$\begin{aligned} \frac{-\mu^2}{2\widetilde{\sigma}^2} &= -\frac{\mu^2}{2}(\frac{1}{\tau^2}+\frac{n}{\sigma^2}) \\ \widetilde{\sigma}^2 &= (\frac{N}{\sigma^2}+\frac{1}{\tau^2})^{-1} \end{aligned}$$

After matching coefficients of μ, we can get that

$$\begin{aligned} \frac{2\mu\widetilde{\mu}}{2\widetilde{\sigma}^2} &= \mu(\frac{\mu_0}{\tau}+\frac{\sum_{n=1}^{N} x_n}{\sigma^2}) \\ \widetilde{\mu} &= \frac{N/\sigma^2}{N/\sigma^2+1/\tau^2}\bar{x} + \frac{1/\tau^2}{N/\sigma^2+1/\tau^2}\mu_0 \end{aligned}$$

It is noticed that the posterior mean is a convex combination of the prior and the MLE, with weights proportional to the relative noise levels. Utilizing the posterior mean, we can derive the equation that can sequentially update the mean value. When we get the first input, the mean value can be updated as follows:

$$\mu_1 = \mu_0 + (x-\mu_0)\frac{\sigma_0^2}{\sigma^2+\sigma_0^2} = x - (x-\mu_0)\frac{\sigma_0^2}{\sigma^2+\sigma_0^2}$$

4 ML Parameter Estimation for General BNs

4.1 Decomposability of the Likelihood

If we assume that nodes in a BN are fully observed and the parameters for each CPD are globally independent, the (log) likelihood decomposes into several independent terms and can be learned separately:

$$\log p(X;\theta) = \sum_{i=1}^{N}\sum_{k=1}^{M} \log p(x_{i,k}|\pi(x_k);\theta_k)$$

For the discrete count, it can be shown that the MLE has a intuitive interpretation that the optimal estimate is the counts of the joint configuration of the child and its parents, divide by the marginal counts of the parents.

4.2 Example: Learning Markov Chain Transition Matrix

A Markov transition matrix A is a stochastic matrix is a matrix where each row is a Multinomial distribution, i.e. $\sum_j A_{ij} = 1$ with applications in, for instance, estimating a bi-gram language model. The MLE then follows from the general rule we just derived derived:

$$A^*_{ij} = \frac{n_{i\to j}}{n_{i\to *}}$$

that is, the counts that we transit from i to j devides by the counts from i to any state. However, if the data we have is very sparse, we might not have seen every possible state we want to estimate. In this case, the probability becomes zero, then any future sequence which has such transition will have zero probability. To mitigate this problem, a standard hack is to use backoff smoothing or deleted interpolation,

$$\hat{A}_{i\to *} = \lambda\eta + (1-\lambda)A^{ML}_{i\to *}$$

4.3 Example: Learning Hidden Markov Model

There are two scenarios of HMM: ones is supervised learning where we know the true state that the observed data comes from, and the other is that these states are hidden and not given. Since we are discussing MLE for fully observed BNs, here we only consider the case where we are given the true states and the goal is to estimate the transition probabilities. Reall that for HMM, we have a observed sequence $x_1, x_2, ..., x_T$ where each data point comes from a state, so we also have a state sequences $y_1, y_2, ..., y_T$. We assume that there are temporal relations between states of which we try to capture via a transition matrix a, that is,

$$p(y^j_t = 1|y^i_{t-1} = 1) = a_{i,j}$$

or

$$p(y^j_t|y^i_{t-1} = 1) = Multinomial(a_{i,1}, a_{i,2}, ..., a_{i,M})$$

Also, we have a start probabilities for the first state,

$$p(y_1) = Multinomial(\pi_1, \pi_2, ..., \pi_M)$$

and finally we have the emission probabilities that generates the data given a state,

$$p(x_t|y^i_t = 1) = Multinomial(b_{i,1}, ..., b_{i,K})$$

for discrete data with K different possible events.

Now we define A_{ij} be the counts that transition from i to j occurs in the state sequence; B_{ik} be the counts that state i in y emits k in x. Then we can show that the maximum likelihood estimates for these two parameters are,

$$a_{ij}^{ML} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}$$
$$b_{ik}^{ML} = \frac{B_{ik}}{\sum_{k'} B_{ik'}}$$

5 Summary of Learning Fully Observed BNs

In this note, we first described maximum likelihood learning for BNs. Then we consider a structure learning algorithm that aims to learn a tree structure for the fully observed scenario that is optimal in the information theoretic sense. Afterwards, we focus on how to learn the parameters for discrete counts model of a BN where decomposability plays an important role. Last but not least, we also discuss about how under some circumstance (e.g. conjugacy), Bayesian estimation can be easily carried out without having much more effort than MLE.