10-708: Probabilistic Graphical Models, Spring 2015

6 : Learning fully observed undirected GM

Lecturer: Yaoliang Yu *Scribes: Maria De Arteaga, Satwik Kottur, William Herlands*

1 Introduction

Undirected graphical models are useful for modeling systems where we understand which elements interact with one another, but we are unsure of the causal ordering. Additionally, domains where data is inherently unordered (such as image processing) are better suited to undirected models than directed ones. In this lecture we study the problem of learning or estimation of undirected graphical models. The lecture can be divided into two main parts. The first, detailed in Section 2 addresses structural learning by leveraging the Precision matrix of a multivariate Gaussian distribution. The second, detailed in Section 3, focuses on parameter estimation and approximations of the model's maximum likelihood.

2 ML Structural Learning via Neighborhood Selection for completely observed MRF

We begin by learning the structure of the undirected model. Our data are m instantiations of $X \in R^n$. For structure learning we focus on Gaussian Graphical Models. Such models assume that the n nodes follow a multivariate Gaussian distribution,

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{1}$$

Where Σ is the covariance matrix. Additionally, without loss of generality we assume $\mu = 0$, which may alternatively be achieved by simply subtracting out the mean.

2.1 Precision Matrix

The Precision Matrix, Q, can be substituted for the covariance matrix where,

$$Q = \Sigma^{-1} \tag{2}$$

To understand the difference between Σ and Q consider the case where $\Sigma_{i,j} = 0$

$$\Sigma_{i,j} = Cov(X_i, X_j) = 0 \tag{3}$$
$$\implies E[X_i X_j] - E[X_i]E[X_j] = 0 \tag{4}$$
$$\implies E[X_i X_j] = E[X_i]E[X_j] \tag{5}$$
$$\implies X_i \perp X_j \tag{6}$$
$$\tag{7}$$

This can be represented by the graphical concept of two separated nodes without any connections between themselves as shown in Figure 1.

Figure 1: Graphical representation of $\Sigma_{i,j} = 0$

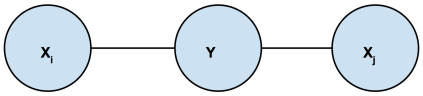

Figure 2: Graphical representation of $Q_{i,j} = 0$

However, if $Q_{i,j} = 0$ we have a different independence condition. In this case $X_i \perp X_j | X_{-i,j}$, the random variables are independent conditioned on all other random variables in the graph. This can be represented by the graphical concept of two nodes where all paths between the nodes pass through another node set. A simple diagram of such a system is shown in Figure 2.

The advantages of using the Precision matrix, Q, instead of the Covarience matrix, Σ is that in undirected models Q tends to contain more zeros that the corresponding Σ. Take for example the sample matrices below.

$$Q = \begin{pmatrix} 1 & 6 & 0 & 0 & 0 \\ 6 & 2 & 7 & 0 & 0 \\ 0 & 7 & 3 & 8 & 0 \\ 0 & 0 & 8 & 4 & 9 \\ 0 & 0 & 0 & 9 & 5 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.10 & 0.15 & -0.13 & -0.08 & 0.15 \\ 0.15 & -0.03 & 0.02 & 0.01 & -0.03 \\ -0.13 & 0.02 & 0.10 & 0.07 & -0.12 \\ -0.08 & 0.01 & 0.07 & -0.04 & 0.07 \\ 0.15 & -0.03 & -0.12 & 0.07 & 0.08 \end{pmatrix}$$

While Σ has no zero elements, Q has 12 zeros. This is important since we can derive the structure of the graphical model from the zero elements. For instance, in the above example, we see that the symmetric placement of zeros in Q corresponds to a set of conditional independences. These inter-dependencies can be represented by the undirected graph in figure 3

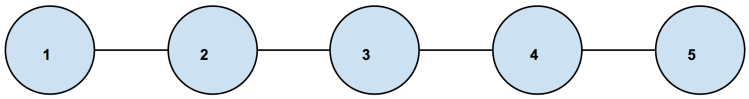

Figure 3: Undirected graph defined by $Q = \Sigma^{-1}$

One special case of note is when Q is a block matrix, such as shown below. This Precision matrix indicates a graphical model with two separated components, as shown in Figure 2.1. Note that under such circumstances we expect that Σ would also have some zero entries.

$$Q = \begin{pmatrix} 1 & 6 & 0 & 0 \\ 6 & 2 & 0 & 0 \\ 0 & 0 & 3 & 8 \\ 0 & 0 & 8 & 4 \end{pmatrix}$$

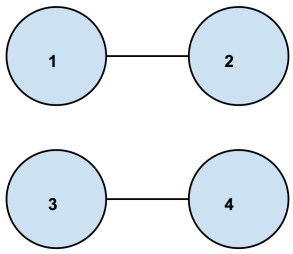

Figure 4: Undirected graph defined by block Q matrix

2.2 Sparsity in Graphs

Let n be the number of random variables and p be the number of observations for learning the graphical model. It is quite possible that we have $n \gg p$ and the co-variance matrix Σ calculated from empirical data is highly ill-conditioned. We, therefore, cannot obtain Q directly by inverting Σ. In such cases, we resort to constructing Q using optimizing methods which enforce sparsity. Note that the graphical model should faithfully represent as many conditional independences as possible present in the underlying distribution, without introducing additional ones; along with fitting the observed data well in terms of likelihood. Therefore, sparse graphical models are of interest to us.

From the above discussion, it is clear that we want a sparse Q such that maximizes the log-likelihood of the observed data. Regularization using L_1 norm (8), also called the lasso, achieves exactly this. We solve many such lasso problems for each variable to construct the precision matrix. For each variable x_i, we solve for i^{th} row of Q using the following equation:

$$\begin{aligned} \hat{\theta_i} \quad &= \quad \arg\min_{\theta_i} l(\theta_i) + \lambda_i |\theta_i|_1 \\ where \quad & l(\theta_i) = logP(y_i|x_i, \theta_i) \end{aligned} \tag{8}$$

Ising Models

A special type of discrete, pair-wise undirected graphical models where each edge has an associated potential along with node potentials. For a variable x_d in such a model, probability function can be written as:

$$P(x_d|\Theta) = exp\left(\sum_{i \in V} \theta_{ii}^t x_{d,i} + \sum_{(i,j) \in E} \theta_{ij} x_{d,i} x_{d,j} - A(\Theta)\right) \tag{9}$$

The probability takes the canonical form of exponential family, where $\{\theta_{ij}\}$ are the natural parameters and $\{x_{d,i}\}$ are the sufficient statistics. It can also been shown that we can use similar L_1 normalized logistic regression to obtain the sparse estimate of the neighborhood of each variable.

Consistency

As their is no direct way to show that the Q estimated by solving lasso problems is positive semi-definite, asymptotic bounds guarantee consistency, under specific verified conditions[1], as seen in eq. 10.

$$\mathcal{P}\left[\hat{G}(\lambda_n) \neq G\right] = \mathcal{O}(exp(-Cn^{\epsilon})) \to 0 \tag{10}$$

As $n \to \infty$, the probability that the graphs do not match goes down to zero.
Remark: It must be noted that the regularizer is not actually used to introduce an "artificial" sparsity bias, but a devise to ensure consistency under finite data and high dimension condition.

3 ML Parameter estimation for completely observed MRFs of given structure

3.1 MLE for BNs

In the case of fully observed directed graphs, the product form of the joint distribution can be used to decompose the log-likelihood function into a sum of local terms, one per node:

$$l(\theta; D) = \log p(D|\theta) = \log \prod_n \left(\prod_i p(x_{n,i}|x_{\pi_i}, \theta_i) \right) = \sum_i \left(\sum_n \log p(x_{n,i}|x_{\pi_i}, \theta_i) \right) \tag{11}$$

Once it is expressed as a sum, it is possible to find one parameter at a time.

3.2 MLE for undirected graphical models

In the case of undirected graphical models, the normalization constant Z is a function of all the parameters, so a factorization like the one we did for directed graphical models is no longer possible. Therefore, it is necessary to do inference to solve the problem. Recall that in this case the joint probability takes the following form:

$$l(\theta; D) = \log p(x_1, x_2, ..., x_n|\theta) = \frac{1}{Z} \prod \Psi_c(x_c) \tag{12}$$

where $Z = \sum x_1, ..., x_n \prod_{c \in C} \Psi_c(x_c)$ and Ψ_c corresponds to the potentials.

[1] Omitted from the current discussion, for simplicity

Now, since we are working with empirical data, we can count the realizations of each vector, or the times a given configuration occurs. Here is an example to show how that would work in practice:

Let's assume we have (x_1, x_2), where $x_i \in \{0, 1\}$. The possible configurations we can have are: $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$. The number of times each of this appear in the data is:

$(0, 0) \to 5$

$(0, 1) \to 3$

$(1, 0) \to 10$

$(1, 1) \to 2$

Now, $m(x_1 = 0) = 8$, for example.

So, we can define the following:

1. Total counts $m(x) = \sum_n \delta(x, x_n)$

2. Clique counts $m(x_c) = \sum_{x_{V \setminus c}} m(X)$

In terms of the counts, the log likelihood is given by:

$$p(D|\theta) = \prod_n \prod_x p(x|\theta)^{\delta(x,x_n)} \tag{13}$$

$$\implies \log p(D|\theta) = \sum_n \sum_x \delta(x, x_n) \log p(x|\theta) = \sum_x \sum_n \delta(x, x_n) \log p(x|\theta) \tag{14}$$

$$\implies \qquad l = \sum_x m(x) \log \left(\frac{1}{Z} \prod_c \Psi_c(x_c) \right) \tag{15}$$

$$= \sum_c \sum_{x_c} m(x_c) \log \Psi_c(x_c) - N \log Z \tag{16}$$

Remember Z is a function of all the parameters, so we still have to figure out what to do with it.

Now, consider the double index parameter $\theta_{c,x(c)} = \log \Psi_c(x_c)$. Working with this natural parameter, the function is concave. Therefore, we can apply coordinate ascent.

$$\frac{\partial l}{\partial \Psi_c(x_c)} = \frac{m(x_c)}{\Psi_c(x_c)} - \frac{N}{Z} \frac{\partial Z}{\partial \Psi_c(x_c)} \tag{17}$$

Recall $Z = \sum_{\tilde{x}_c} \prod_c \Psi_c(\tilde{x}_c)$

Therefore,

$$\frac{\partial Z}{\partial \Psi_c(x_c)} = \sum_{\tilde{x}_c} \delta(\tilde{x}_c = x_c) \prod_{d \neq c} \Psi_d^{(t)}(\tilde{x}_d) \tag{18}$$

$$= \sum_{\tilde{x}_c} \delta(\tilde{x}_c = x_c) \prod_{d \neq c} \Psi_d^{(t)}(\tilde{x}_d) \frac{\Psi_c^{(t)}(x_c)}{\Psi_c^{(t)}(x_c)} \tag{19}$$

$$= \frac{1}{\Psi_c^{(t)}(x_c)} \sum_{\tilde{x}_c} \delta(\tilde{x}_c = x_c) \prod_{d \neq c} \Psi_d^{(t)}(\tilde{x}_d) \Psi_c^{(t)}(x_c) \tag{20}$$

$$\tag{21}$$

If we plug this back into equation 17, and taking into account:

- $Z^{(t)} = \sum_{\tilde{x}_c} \prod_c \Psi_c^{(t)}(\tilde{x}_c)$
- $p^{(t)}(x_c) = \delta(\tilde{x}_c = x_c) \prod_c \Psi_c^{(t)}(\tilde{x}_c)$

we obtain:

$$\implies \frac{\partial l}{\partial \Psi_c(x_c)} = \frac{m(x_c)}{\Psi_c(x_c)} - \frac{N}{Z} \frac{Z^{(t)}}{\Psi_c^{(t)}(x_c)} p^{(t)}(x_c) = 0 \tag{22}$$

Here, there are two ways of solving this. The first one, is to replace Z directly, which would involve more calculations. The second one, is to notice that in the equation

$$\Psi_c^{(t+1)}(x_c) = \Psi_c^{(t)}(x_c) \frac{m(x_c)}{N - m(x_c)} \frac{1 - p^{(t)}(x_c)}{p^{(t)}(x_c)} \tag{23}$$

the following holds: $N - m(x_c) = 1 - p^{(t)}(x_c)$.

Hence, we obtain that the derivative of the log-likelihood is

$$\frac{\partial l}{\partial \Psi_c(x_c)} = \frac{m(x_c)}{\Psi_c(x_c)} - N \frac{p(x_c)}{\Psi_c(x_c)} \tag{24}$$

From this, it can be derived that for the maximum likelihood parameters the following holds:

$$p^{\star}_{MLE}(x_c) = \frac{m(x_c)}{N} = \tilde{p}(x_c) \tag{25}$$

This is, for each clique, the model marginals must be equal to the empirical marginals. This, however, is not enough to estimate the parameters, but barely a property that we know must hold when we find them.

There is a special type of UGM for which estimating parameters is easier: decomposable graphs.

Figure 5: In the left, an example of a graph that is *not* decomposable. In the right, one that is decomposable.

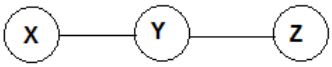

Figure 6: En example of decomposable graph. Cliques are (X, Y) and (Y, Z) and the separator is Y.

A decomposable graph is one in which all clique potentials are defined on maximal cliques. Figure 3.2 shows an example of a graph that is *not* decomposable in the left, and one that is in the right. For this type of graphs, the MLE of clique potentials equate to the empirical marginals of the corresponding clique, which means we can solve the problem of estimating the parameters of the MLE by inspection.

The potential based representation is

$$p(x) = \frac{\prod_c \Psi_c(x_c)}{\prod_s \Psi_s(x_s)} \tag{26}$$

where c refers to the cliques and s to the separator. Let's see an example of this.

In Figure 3.2 the cliques are (X, Y) and (Y, Z) and the separator is Y. Knowing that the empirical marginals must equal the model marginals, we can take a 'guess' and then verify such guess satisfies this condition.

Our guess will be the following:

$$\hat{p}_{MLE}(x, y, z) = \frac{\tilde{p}(x, y)\tilde{p}(y, z)}{\tilde{p}(y)} \tag{27}$$

Now, we can see the condition over empirical marginals and model marginals holds:

$$\hat{p}_{MLE}(x, y) = \sum_z \hat{p}_{MLE}(x, y, z) = \tilde{p}(x|y) \sum_z \tilde{p}(y, z) = \tilde{p}(x, y) \tag{28}$$

$$\hat{p}_{MLE}(x, y) = \tilde{p}(x, y) \tag{29}$$

With an analogous reasoning, we can conclude $\hat{p}_{MLE}(y, z) = \tilde{p}(y, z)$.

Having this, we can obtain the clique potentials:

$$\hat{\Psi}_{xy}^{MLE}(x,y) = \tilde{p}(x,y) \tag{30}$$

$$\hat{\Psi}_{yz}^{MLE}(y,z) = \frac{\tilde{p}(y,z)}{\tilde{p}(y)} = \tilde{p}(y|z) \tag{31}$$

$$\tag{32}$$

In general, we can compute the clique potentials by equating them to the empirical marginals or conditionals.

3.3 Iterative Proportional Fitting (IPF)

In general, learning of parameters for undirected graphs is difficult, as compared to directed Bayesian networks, due to the presence of the partition function that combines all the potentials. In the previous discussion, we have seen that for special kinds of undirected graphs i.e. *decomposable graphs*, we can decouple the learning problems locally. In fact, such graphs allow us to write the solution just by the method of inspection. On the other hand, graphs which are decomposable, *non-decomposable graphs*, do not have this luxury of decoupling. We develop methods which work with such general graphs using fixed-point iterations.

The basic idea is to express log likelihood as the function of clique potentials (eq. **??**), find derivative (eq. 33) with a particular clique potential and set it to zero. As the likelihood turns out to be concave, the zero of derivative gives the global maximum.

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N\frac{p(x_C)}{\psi_C(x_C)} \tag{33}$$

Defining the empirical distribution, $\tilde{p}(x_C) \triangleq m(x_C)/N$, we have:

$$\frac{\tilde{p}(x_C)}{\psi_C(x_C)} = \frac{p(x_C)}{\psi_C(x_C)} \tag{34}$$

This is a fixed-point equation in $\psi_C(x_C)$. Eq. 34 results in a set of nonlinear equations which involve the clique potentials x_C implicitly. We adopt an iterative procedure where we hold the value of x_C on right side constant, updating it for the current iteration using eq. 35. By holding constant, we mean use the value obtained in the previous iteration, denoted by the superscript for clarity. Therefore, eq. 35 provides the update rule for iterative proportional fitting approach.

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \tag{35}$$

In order to compute $p^{(t)}(x_C)$, we have to run inference after each iteration. This involves the computation of the partition function $Z^{(t)}$ for every step, which can be computationally expensive. However, it can be shown that $Z^{(t)} = Z^{(t+1)}$ alleviating this problem. The method of iterative proportional fitting can also be interpreted as a co-ordinate descent and KL divergence minimization, each giving further insights.

Coordinate Descent View

Treating the clique potential for each assignment of variables as independent coordinate for the log likelihood function, we once again obtain eq. 35 as the update equation for each step. This approach can be used to explain the increase in the convex log likelihood at each iteration, resulting in a global maximum.

KL Divergence Minimization View

There exists an equivalence between maximizing the log-likelihood of the observed data and minimizing the KL divergence of empirical distribution of the data and model distribution.

$$\arg\max_{\Theta} \; l(D|\Theta) \Leftrightarrow \arg\min_{\Theta} \; KL(\tilde{p}(x) \;||\; p(x|\Theta)) = \sum_{x} \tilde{p}(x) log \left(\frac{\tilde{p}(x)}{p(x|\Theta)} \right) \tag{36}$$

Using conditional KL Divergence (see Appendix A), we get eq. 37. It can be shown that changing the clique potential ψ_C does not have any effect on the conditional distribution, therefore the second term remains unchanged. The minimization reduces to just the first term. In this context, we can interpret IPF as retaining the 'old' conditional probability $p^{(t)}(x_{-C}|x_C)$ and replacing the 'old' marginal probability $p^{(t)}(x_C)$ with observed marginal probability $\tilde{p}(x_C)$.

$$KL(\tilde{p}(x) \;||\; p(x|\Theta)) = KL(\tilde{p}(x_C) \;||\; p(x_C|\Theta)) + \sum_{x_C} \tilde{p}(x_C) \; KL(\tilde{p}(x_{-C}|x_C) \;||\; p(x_{-C}|x_C)) \tag{37}$$

Appendix

A Conditional KL Divergence

Consider two distributions $p(x)$ and $q(x)$. Splitting x into two non-overlapping set of variables x_A, x_B such that $p(x) = p(x_A)p(x_B|x_A)$, we have:

$$\begin{aligned} KL(q(x_A, x_B) \;||\; p(x_A, x_B)) &= \sum_{x_A, x_B} q(x_A)q(x_B|x_A) log \frac{q(x_A)q(x_B|x_A)}{p(x_A)p(x_B|x_A)} && (38) \\ &= \sum_{x_A, x_B} q(x_A)q(x_B|x_A) log \frac{q(x_A)}{p(x_A)} + q(x_A)q(x_B|x_A) log \frac{q(x_B|x_A)}{p(x_B|x_A)} && (39) \\ &= KL(q(x_A) \;||\; p(x_A)) + \sum_{x_A} q(x_A) \; KL(q(x_B|x_A) \;||\; p(x_B|x_A)) && (40) \end{aligned}$$

Now let $q(x) = \tilde{p}(x)$ and $x_A = x_C, x_B = x_{-C}$ to get eq. 37.