10708 Graphical Models: Homework 2

Due Monday, March 18, beginning of class

Feburary 27, 2013

Instructions: There are five questions (one for extra credit) on this assignment. There is a problem involves coding. You can program in whatever language you like, although we suggest MATLAB. Do *not* attach your code to the writeup. Instead, put your code in a directory called "andrewid-HW2" and tar it into a tgz named "andrewid-HW2". For example, epxing-HW2.tgz. Email your tgz file ONLY to gunhee@cs.cmu.edu, seunghak@cs.cmu.edu and kpuniyan@cs.cmu.edu. Refer to the web page for the policies regarding collaboration, due dates, extensions, and late days.

1 Learning Gaussian Graphical Models And Ising Models [35 points]

1. Consider a p-dimensional Gaussian graphical model $p \sim \mathcal{N}(0; \Sigma)$ defined on $\mathbf{x} = (x_1, \ldots, x_p)$. Let $\Omega = \Sigma^{-1}$ denote the precision matrix. In this problem, you will show that $\Omega_{ij} = 0$ iff x_i is conditionally independent of x_j given the remaining variables.

   (a) (5 pts) Suppose that we partition $\mathbf{x}$ into two subsets $\mathbf{x} = (\mathbf{x}_1; \mathbf{x}_2)$ where $\mathbf{x}_1$ is a subset of the p variables and $\mathbf{x}_2$ denotes the remaining variables. The joint Gaussian is

   $$p\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \middle| 0, \Sigma\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} \middle| 0, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right) \tag{1}$$

   Derive $p(\mathbf{x}_1|\mathbf{x}_2)$. (Hint: Use the form of inverse of a block matrix in terms of Schur complement).

   (b) (4 pts) Let us denote the precision matrix in block form $\Omega = \begin{bmatrix}\Omega_{11} & \Omega_{12}\\ \Omega_{21} & \Omega_{22}\end{bmatrix}$. Represent $var(\mathbf{x}_1|\mathbf{x}_2)$ in terms of Ω.

(c) (6 pts) Using the above two results, argue that $\Omega_{ij} = 0$ iff x_i is conditionally independent of x_j given the remaining variables.

2. (5 pts) The above results motivate the Graphical Lasso (Glasso) algorithm. Suppose that we have n multivariate normal observations of dimension p with mean 0 and covariance Σ. Let $\Theta = \Sigma^{-1}$ and S be the sample covariance $S = \sum_{i=1}^{n} \mathbf{x}^{(i)}\mathbf{x}^{(i)T}/n$. The Glasso performs the following optimization:

$$\Theta^* = \text{argmax}\,(\log\det\Theta - \text{tr}(S\Theta) - p||\Theta||_1) \tag{2}$$

The first two terms are the log-likelihood of Gaussian distribution, and the third term is l_1 penalty term: $||\Theta||_1 = \sum|\Theta_{ij}|$. Show that the log likelihood of the n multivariate Gaussian distribution that we maximize is identical to $\log\det\Theta - \text{tr}(S\Theta)$.

3. (15 pts) Implement the Meinshausen-Buhlmann algorithm and the Glasso algorithm discussed in the class. You can use any programming languages as you want (*e.g.*, Matlab, R, Python).

We generate 50 random vectors from p-dimensional multivariate normal distribution $N(0, \Sigma)$ with $p = 10$, and save it in `Xinput.mat`. Apply both MB algorithm and Glasso to estimate its precision matrices with different λ values: $\lambda = [0, 20, 30, 40]$ for MB algorithm and $\lambda = [0, 0.2, 0.5, 0.8]$ for Glasso.

Draw all the estimated precision matrices as 10×10 binary matrices using black (nonzero) and white (zero) colors. Discuss what happens as λ increases.

[References] (1) *High-dimensional graphs and variable selection with the Lasso.* Meinshausen and Buhlmann. Ann. Statist. 2006.
(2) *Sparse inverse covariance estimation with the graphical lasso.* J. Friedman, T. Hastie and R. Tibshirani. Biostat. 2008.

2 Hidden Markov Model with Mixture of Experts [30 marks]

In class, we saw the conditional mixture model, where we predict Y using a linear function of data X, but the prediction also depends on a latent variable Z. We will now extend this model to the case when $\mathbf{X}$ and $\mathbf{Y}$ are sequences of data. We can think of such a model as being a recurrent version of the conditional mixture of experts, and it has been shown to work well to learn Dynamic Audio/Visual Mapping of audio-visual data, to model EEG rhythms etc.

Consider the model shown in figure 1, where we have a sequence of data $\mathbf{X} = (X_1, ...X_p)$, with "known" states $\mathbf{Y} = (Y_1, ...Y_p)$, but unknown latent variables $\mathbf{Z} = (Z_1, ...Z_p)$.

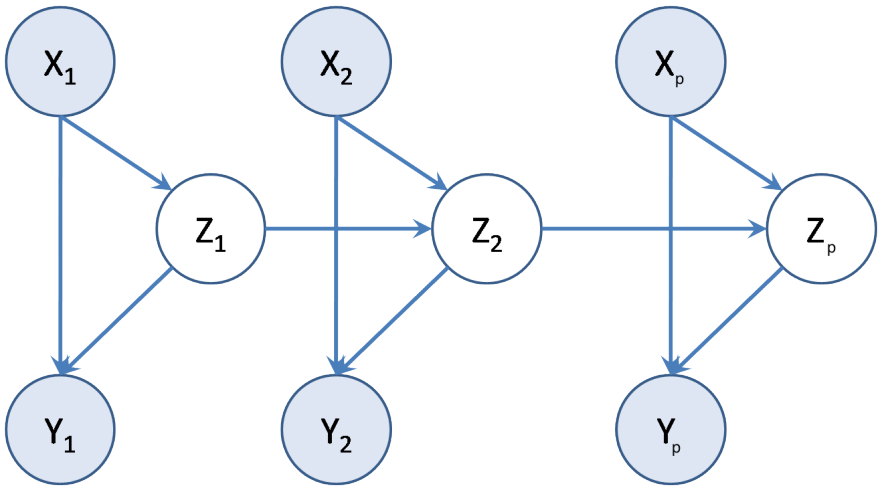

Figure 1: HMM with experts

X_i and Y_i are real-valued, but Z_i takes one of K possible values. Instead of representing each Z_i as a random variable that can take K values, we represent it as a vector with K binary variables eg. if $K = 4$ and $Z_i = 2$ for some i, then we represent it as $Z_i = (0, 1, 0, 0)$.

We assume our model has the following distribution. The observed variable Y_i depends on the observed variable X_i and the hidden variable Z_i as:

$$P(Y_i|x_i, z_{ik} = 1) = \mathcal{N}(y;\ \theta_k^T x_i,\ \sigma_k^2) \tag{3}$$

That is, Y_i is Gaussian with mean $\theta_k^T x_i$ and variance σ_k^2. The choice of the θ parameter used depends on which z_{ik} is 1.

To make our derivations simpler, we assume that

$$P(z_{ik} = 1|x_i, z_{i-1} = j) = P(z_{ik} = 1|x_i) \times P(z_{ik} = 1|z_{i-1} = j) \tag{4}$$

Note that this assumption is not true in a general Bayes Net, but has been assumed for this problem only to allow us to get easy updates!

Then, we need to define the transition function :

$$P(z_i = k|z_{i-1} = j) = \eta_{jk} \tag{5}$$

And finally, the dependence of Z_i on X_i:

$$P(z_{ik} = 1|x_i) \propto e^{\gamma_k^T x_i} \tag{6}$$

1. What are the parameters of this system, and what dimensions are they? (e.g. θ is a vector parameter of length K). What are the hidden variables of the system?

2. Given n i.i.d. data points, $(\mathbf{X}^{(1)}, \mathbf{Y}^{(1)}), ..., (\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})$, where each point $\mathbf{X}^{(j)} \in \mathbb{R}^p$ and $\mathbf{Y}^{(j)} \in \mathbb{R}^p$, write out the expected conditional log likelihood of the data, and derive its lower bound using Jensen's inequality.

3. Derive the update equations for the E and M steps for this model.

3 GLIMs and KL divergence [10 points]

Let $f_1(x), ..., f_k(x)$ denote k features of x, and let $P(x|\theta)$ and $P(x|\eta)$ denote two distributions in the exponential family over the features. Thus, $P(x|\theta) = exp(\sum_{i=1}^{k} \theta_i f_i(x) - A(\theta))$ and $P(x|\eta) = exp(\sum_{i=1}^{k} \eta_i f_i(x) - A(\eta))$.

Show that the KL distance can be expressed as

$$KL(P(x|\theta); P(x|\eta)) = \sum_{i=1}^{k} (\theta_i - \eta_i) \frac{\partial A(\theta)}{\partial \theta_i} - A(\theta) + A(\eta) \tag{7}$$

4 Iterative Proportional Fitting [20 points]

In this problem, we will have insight of Iterative Proportional Fitting (IPF) by showing that it is related to the joint probability of a graphical model.

Consider an undirected graphical model distribution,

$$p(x) = \frac{1}{Z} \prod_{C} \Psi_C(x_C).$$

Given the empirical marginal, $\tilde{p}(x_C)$, IPF update rule for estimating MLE of a graphical model is:

$$\Psi_C(x_C)^{(t+1)} = \Psi_C(x_C)^{(t)} \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}.$$

Let us assume that Z is constant across iterations.

Now prove the following: The above IPF update rule implies that the joint probabilities are updated as follows:

$$p^{(t+1)}(x_U) = p^{(t)}(x_{U \setminus C}|x_C)\tilde{p}(x_C),$$

where U is the set of all nodes in the graph.

5 (Extra credit) Module network learning [20 points]

[Exercise 18.22, Daphne Koller and Nir Friedman]

In this problem, we will consider the task of learning a generalized type of Bayesian networks that involves shared structure and parameters. Let χ be a set of variables, which we assume are all binary-valued. A *module network* over χ partitions the variables χ into K disjoint clusters, for $K << n = |\chi|$. All of the variables assigned to the same cluster have precisely the same parents and CPD. More precisely, such a network defines:

- An assignment function $\mathcal{A}$, which defines for each variable X, a cluster assignment $\mathcal{A}(X) \in \{C_1, ..., C_K\}$.
- For each cluster $C_k (k = 1, ..., K)$, a graph $\mathcal{G}$ which defines a set of parents $\mathbf{Pa}_{C_k} = \mathbf{U_k} \subset \chi$ and a CPD $P_k(X|\mathbf{U_k})$.

The cluster network structure defines a ground Bayesian network where, for each variable X, we have the parents $\mathbf{U_k}$ for $k = \mathcal{A}(X)$ and the CPD $P_k(X|\mathbf{U_k})$. Figure 2 shows an example of such a network.

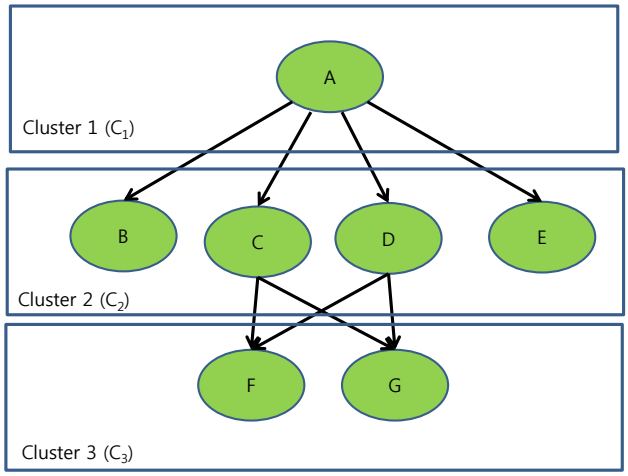

Figure 2: An example of module network

Assume that our goal is to learn a cluster network that maximizes the BIC score given a data set $\mathcal{D}$, where we need to learn both the assignment of variables to clusters and the graph structure.

1. (5 pts) Define an appropriate set of parameters and an appropriate notion of sufficient statistics for this class of models, and write down a precise formula for the likelihood function of a pair $(\mathcal{A}, \mathcal{G})$ in terms of the parameters and sufficient statistics.

2. (15 pts) We use greedy local search to learn the structure of the cluster network. We will use the following types of operators (each operation should remain the graph acyclic):

   - **Add** operators that add a parent for a cluster;
   - **Delete** operators that delete a parent for a cluster;
   - **Node-Move** operators $o_{k \to k'}(X)$ that change from $\mathcal{A}(X) = k$ to $\mathcal{A}(X) = k'$. (If $X \in \mathbf{Pa}_{C_{k'}}$, moving X to k' is not allowed.)

   As usual, we want to reduce the computational cost by caching our evaluations of operators and reusing them from step to step.

   (a) Why did we not include edge reversal in our set of operators?

   (b) Describe an efficient implementation for the update associated with the Node-Move operator.

   (c) For each type of operator, specify which other operators need to be reevaluated once the operator has been taken. Briefly justify your response. (Suppose that we cache and update evaluations of operators and reuse them to save the computation.)