10-708 Probabilistic Graphical Models

Homework 2

Due Mar 10, 7:00 PM

Rules:

1. Homework is due on the due date at 7:00 PM. The homework should be submitted via Gradescope. The solution to each problem should start on a *new* and marked appropriately on Gradescope. For policy on late submission, please see course website.
2. We recommend that you typeset your homework using appropriate software such as LaTeX. If you are writing, please make sure your homework is cleanly written up and legible. The TAs will not invest undue effort to decrypt bad handwriting.
3. **Code submission:** for programming questions, you must submit the complete source code of your implementation also via Gradescope. Remember to include a small README file and a script that would help us execute your code.
4. **Collaboration:** You are allowed to discuss the homework, but you should write up your own solution and code. Please indicate anyone you collaborated with in your submission.

---

1 EM Algorithm for Confounded Heterogeneous Data [50 pts] (Haohan)

In class, we have seen one of the most classical examples in the education of EM algorithm: a mixture of Gaussians, in which heterogeneous data X are believed to be generated from a collection of Gaussians. Here, let's try to extend this basic example into a more interesting problem.

First, let's say we have a linear relationship between labels and features:

$$y = X\beta + \epsilon$$

where X is heterogeneous, as in the mixture of Gaussians example, and ϵ just noise.

Now, to make it more interesting, we say that y is not simply determined by $X\beta$ (and not any generalized form in $f(X)$ either). Instead, y is an outcome of both X and the underlying distribution the corresponding data sample comes from. In other words, for the ith sample, y_i is determined by both X_i and the distribution X_i comes from. Figure 1 shows the graphical model of this problem, in which we use D to denote the underlying distribution. This is the setting we are interested in for this problem.

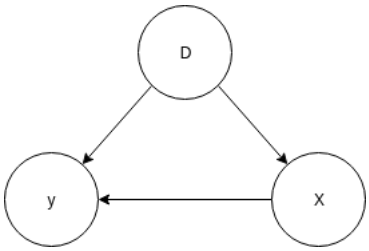

Figure 1: a graphical model for the problem

This setting has been extensively studied in the area of causal inference, where D is known as confounding factors. The existence of D makes it difficult to reliably estimate β. Now, we introduce a model that can do the job reasonably well.

One of the simplest models that is used to deal with this problem is:

$$y = X\beta + Zu + \epsilon$$

where Z is an indicator of the underlying distribution (For this problem, let's just assume we know Z ahead of time). We have X is of the size $n \times p$ and Z is of the size $n \times q$. y is a $n \times 1$ vector. The model has an equivalent form as following:

$$y \sim N(X\beta, K\sigma_u^2 + I\sigma_e^2)$$

where $K = ZZ^T$ and to make it simple, we just assume $u \sim N(0, I\sigma_u^2)$

Now let's play with this model.

1.1 EM algorithm [15 pts]

- What is the sufficient statistics for σ_u? [1pts]
- Derive the MLE for σ_u, σ_ϵ and β. [4pts]
- Derive the EM algorithm to estimate σ_u and β. [10pts]
  - ⋆ Hint: consider the joint distribution of (y, u) (as a $(n+q) \times 1$ vector), and u is the unobserved data.
  - ⋆ Feel free to assume the matrix you need to inverse is nonsingular.

1.2 A Playground [15 pts]

- Implement this model and the EM algorithm derived in the previous question. [10 pts]
- Download the data (X.csv, y.csv, K.csv) and feed it into your model, report the curves of MSE of estimated β and golden standard (beta.csv) over each iteration, as well as how the log likelihood changes over each iteration. [5 pts]

1.3 Let's Go Wild [20 pts]

This is an open-ended question that requires you:

- Suggest a data set that is suitable for this model and feed it to this model. Offer a discussion of why you suggested data set is suitable for this model. [5 pts]
  - ⋆ If you truly believe your data set is heterogeneous, but there is no information about K available, you can just run a K-means or whatever simple clustering algorithm to generate Z.
  - ⋆ Don't worry if the model does not work well on your data set.
- On your data set, compare this model to a vanilla linear regression $y = X\beta + \epsilon$ on a suitable classical evaluation metric that depends on your task. [5 pts]
  - ⋆ This model may not work well on prediction task, but should work well in tasks regarding estimation of β.
- Based on the comparison results you saw, suggest some improvements on this model or the basic EM algorithm. [10 pts]
  - ⋆ To fully get these 10 points, you need to offer some deep thoughts on this question. A good suggestion typically involves several references of state-of-the-art papers.
  - ⋆ For example, simply suggesting adding some weights over the data, or replacing $X\beta$ to a generalized form $f(X)$ will not be accredited as a 10 pts answer, even if your $f(\cdot)$ is a 1000 layer convolutional neural net.

Have fun.

2 Estimation of Precision Matrix [40 pts] (Haohan)

Let's visit some relevant works on estimating a graph by estimating the precision matrix, In this question, precision matrix is denoted as $P = \Sigma^{-1}$, where Σ is the covariance matrix of Gaussian data.

2.1 Precision Matrix [15pts]

- Many relevant pieces of literature state that "it is **well known** that the zero coefficients in P correspond to conditional independence of corresponding variables", but they barely show why. Let's try to do it here: show that the coefficient in precision matrix $P_{i,j}$ encodes partial correlation relationship of corresponding variables $\rho_{i,j}$.
- Let's take a step further about partial correlation: show that partial correlation of two variables $\rho_{i,j}$ is proportional to linear dependency coefficient $\phi_{i,j}$ where $X_j = X_i\phi_{i,j}$.
- Show that estimating the graph is effectively optimizing the following equation:

$$\arg\min_{P} - \log\det P + \mathbf{tr}\left(\tilde{\Sigma}P\right)$$

  where $\tilde{\Sigma}$ is the empirical covariance matrix.

2.2 Graphical Lasso [15 pts]

Graphical Lasso [**?**] is one of the most famous methods for estimating precision matrix, it appends the traditional L_1 regularizer onto the cost function in the previous question, yielding:

$$\arg\min_P -\log\det P + \mathbf{tr}\left(\tilde{\Sigma}P\right) + \lambda||P||_1$$

where $\tilde{\Sigma}$ is the empirical covariance matrix.

- Implement it.
- Use your implementation to estimate a graph from the data in *graph.csv*, for $\lambda = \{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$, report the performance in precision and recall compared to the golden standard in Figure 2

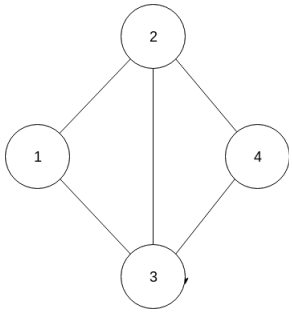

Figure 2: golden standard for graph estimation

- You probably find that your graphical lasso cannot effectively estimate the graph even if we have 1 million data points for just 4 nodes, and there is no noise in the data at all. Now let's see what's happening.

  It is known that graphical lasso can only estimate the graph consistently when the following condition is met:

$$\max_{e\in S^C} ||(\Sigma\otimes\Sigma)_{e,S}(\Sigma\otimes\Sigma)^{-1}_{S,S}||_1 < 1$$

  where $\otimes$ stands for Kronecker product, S stands for the support set (edges in the graph), S^C stands for the complementary set of S.

  Our *graph.csv* is generated according to this covariance matrix:

$$\Sigma = \begin{pmatrix} 1 & c & c & 2c^2 \\ c & 1 & 0 & c \\ c & 0 & 1 & c \\ 2c^2 & c & c & 1 \end{pmatrix}$$

  where c is a positive scalar. We must have $c \in (0, 2^{-\frac{1}{2}}]$ to make sure P is p.s.d. Now, since graphical lasso cannot estimate this graph, find a tighter bound of c.

2.3 More about Graphical Lasso [10 pts]

Now, let's see an alternative cost function to estimate the precision matrix.

$$\arg\min_P \frac{1}{2}\mathbf{tr}\left(P^2\tilde{\Sigma}^T\right) - \mathbf{tr}\left(P\right) + \lambda||P||_1$$

- Show that this new cost function is convex and has a unique minimizer at $P = \Sigma^{-1}$. You may need assume no regularization strength ($\lambda = 0$) to get this.
- The corresponding condition for this cost function is that:

$$\max_{e \in S^C} ||\Gamma_{e,S}\Gamma_{S,S}^{-1}||_1 < 1$$

where $\Gamma = \frac{1}{2}(\Sigma \oplus \Sigma)$ and $\oplus$ stands for Kronecker sum. If the data in *graph.csv* can be estimated with this cost function (but not with graphical lasso), find a tighter bound of c.

3 State Space Model [10 pts] (Haohan)

In the class, we have covered the basic State Space Model, which is:

$$X_t = AX_{t-1} + w_t$$
$$Y_t = CX_t + v_t$$

where X_t are states, Y_t are observations, A is the transition matrix, C is the output matrix, w and v are Gaussian noises with zero mean and covariance matrix Q, R respectively. Therefore, we have:

$$p(Y_t, X_t) = P(X_1)P(Y_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(Y_t|X_t)$$
$$p(Y_t|X_t) = (2\pi)^{\frac{d}{2}}|R|^{-\frac{1}{2}}\exp(-\frac{1}{2}(Y_t - CX_t)^T R^{-1}(Y_t - CX_t))$$

Now, let's make some improvements upon it. Instead of assuming there is only one hidden state sequence $X_1, X_2, ...X_t$ governing the observed variables, we have M state sequences, $X_1^i, X_2^i, ...X_t^i$ and $i \in [1, M]$, correspondingly, we have $\{A_1, A_2, ...A_M\}$ and $\{C_1, C_2, ...C_M\}$ (where $M \ll T$). Then, the next question is which X_t^i, A_i and C_i we need to use at t. To account for this question, let's put an HMM on top of this SSM model, resulting

$$P(Y_t, X_1, X_2, ..., X_M, S_t) = P(S_1)\prod_{t=1}^{T} P(S_t|S_{t-1})\prod_{m=1}^{M} P(X_1^m)\prod_{t=2}^{T} P(X_t^m|X_{t-1}^m)\prod_{t=1}^{T} p(Y_t|X_t^1, X_t^2, ..., X_t^M, S)$$
$$p(Y_t|X_1, X_2, ..., X_M, S = m) = (2\pi)^{\frac{d}{2}}|R|^{-\frac{1}{2}}\exp(-\frac{1}{2}(Y_t - C_m X_t^m)^T R^{-1}(Y_t - C_m X_t^m))$$

where S is a multinomial variable that can take on M values, that determines which set of A_i, C_i or X_t^i to use. One can think of S as hidden states of an HMM on top of this SSM, with initial probability $P(S_0)$ and transition probability $P(S_{t+1}|S_t)$. This is called switching state space model.

3.1 EM algorithm [2 pts]

- Argue that the exact inference is intractable with EM algorithm.

3.2 Mean Field Method [8 pts]

Now we introduce mean field assumption to make EM tractable. For this question, we use the following posterior:

$$Q(\{S_t, X_t\}) = \frac{1}{Z_Q}[\psi(S_1)\prod_{t=2}^{T}\psi(S_{t-1}, S_t)]\prod_{m=1}^{M}\psi(X_1^m)\prod_{t=2}^{T}\psi(X_{t-1}^m, X_t^m)$$

where Z is the normalized constant and $\psi(\cdot)$ are potentials that are defined as following:

$$\begin{aligned}
\psi(S_1 = m) &= P(S_1 = m)q_1^m \\
\psi(S_{t-1}, S_t = m) &= P(S_t = m|S_{t-1})q_t^m \\
\psi(X_1^m) &= P(X_1^m)[P(Y_1|X_1^m, S_1 = m)]^{h_1^m} \\
\psi(X_{t-1}^m, X_t^m) &= P(X_t^m|X_{t-1}^m)[P(Y_t|X_t^m, S_t = m)]^{h_t^m}
\end{aligned}$$

where q_t^m and h_t^m are variational parameters.

Show that, minimizing the KL divergence $KL(Q||P)$ will lead to:

$$\begin{aligned}
h_t^m &= Q(S_t = m) \\
q_t^m &= \exp(-\frac{1}{2}(Y_t - C_m X_t^m)^T R^{-1}(Y_t - C_m X_t^m))
\end{aligned}$$

- See Theorem 1 in [**?**] for a convenient way of minimizing KL.