10-708: Probabilistic Graphical Models, Spring 2015

1 : Introduction

Lecturer: Yaoliang Yu *Scribes: Taiyuan Zhang, Prashant Sridhar*

1 Recap: Structural Learning for completely observed MRF

1.1 Gaussian Graphical Models

In multivariate Gaussian distribution, we let $\mu = 0, Q = \Sigma^{-1}$, we have

$$p(x_1, x_2, ..., x_p|\mu = 0, Q) = \frac{|Q|^{1/2}}{(2\pi)^{n/2}} \exp\{-\frac{1}{2}\sum_i q_{ii}x_i^2 - \sum_{i<j} q_{ij}x_i x_j\} \tag{1}$$

This can be viewed as continuous Markov Random Field with potentials defined on every node and edge.

1.2 The covariance and the precision matrices

Zero entry in covariance matrix Σ denotes independence

$$\Sigma_{i,j} = 0 \to X_i \perp X_j \tag{2}$$

Zero entry in precision matrix indicates conditional independence

$$Q_{i,j} = 0 \to X_i \perp X_j|X_{-(i,j)} \tag{3}$$

The sparsity of precision matrix doesn't lead to sparsity of the covariance matrix

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

1.3 Structural Learning

How to learn a MRF like this?

$$\begin{pmatrix} * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ 0 & * & * & * & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix} \tag{5}$$

And what if $p >> n$? MLE doesn't exist in general. Therefore we should enforce sparsity. And this can help us learn more interested structures.

1.3.1 Graph Lasso

Use Lasso to learn edges for each node iteratively.

Pros:

- Computationally convenient
- Strong theoretical guarantee

Cons:

- Asymmetry
- Not minimax optimal

1.3.2 Regularized MLE

This method solve for

$$\min_Q - \log\det Q + \mathrm{tr}(QS) + \lambda||Q||_1 \tag{6}$$

- S: sample covariance matrix, maybe singular
- $||Q||_1$: may exclude diagonal because we're not interested in the weight of self-circle edge
- $\log\det Q$: implicitly force Q to be PSD symmetric

Pros:

- Single step for estimating graph and inverse covariance
- Using MLE, thus with strong theory

Cons:

- Computationally challenging. Glasso method partly solve this issue

1.3.3 CLIME

If we take the derivative of the MLE formula and set to zero, we have

$$-Q^{-1} + S + \lambda \text{sign}(Q) = 0 \tag{7}$$

Which gives us

$$||Q^{-1} - S||_\infty \leq \lambda \tag{8}$$

So maybe we can solve

$$\min_Q = ||Q||_1 \ s.t \ ||Q^{-1} - S||_\infty \leq \lambda \tag{9}$$

The CLIME method make further relaxation based on that

$$\min_Q = ||Q||_1 \ s.t \ ||SQ - I||_\infty \leq \lambda \tag{10}$$

There're many advantanges of this method

- Both objective and constraint are element-wise separable
- It can be reformulated as LP
- It has strong theoretical guarantee. Variations are minimax-optimal

For large-scale problem, we can solve each column of Q independently in different cores/machines

$$\min_{q_i} = ||q_i||_1 \ s.t \ ||Sq_i - e_i||_\infty \leq \lambda \tag{11}$$

This is still a little troublesome if S is big. Therefore we need to consider first-order method.

2 Introduction to ADMM

The idea behind ADMM is to solve problems of the form

$$\begin{aligned} &min_{w,z} f(x) + g(w) s.t. \\ &\quad Aw + Bx = c \end{aligned}$$

That is we have uncoupled functions in the objective but couple fuctions in the contraint. This is hard to solve for traditional numerical analysis methods like interior point methods. Simply iterating between solving w and x does not work because they are coupled in the constrain. Fixing one constrains the other in an additional way.

The key insight is to try and uncouple the variables.

An exapmple of this problem is empirical risk minimization

$$min_w g(w) + \sum_i f_i(w).$$

Here $g(w)$ is a regularizer on w and $f_i(w)$ is the loss of w on the i^{th} data point. This loss might be squred loss or logistic loss depending on the application. The loss depends on the X and therefore couples w and x in the objective. This can be reformulated by coupling them in the constraint and converting to cannonical form through variable duplication.

For the example above we can define $v = [w_1, \ldots, w_n]$ and constrain $w_i = z$. The problem then becomes

$$\begin{aligned} min_{v,z} g(z) + f(v), s.t. \\ w_i = z, \forall i \end{aligned}$$

This decouples the objective but has the downside of introducing additional variables.

ADMM relies on an augmented laplacian which has a more complicated objective but removes the coupled constraints.

$$L_\mu = min_{w,z} max_\lambda f(w) + g(z) + \lambda^T(Aw + Bz - c) + \mu/2||Aw + Bz - c||_2^2$$

The quadratic term at the end gives better numerical stability and may make the problem strongly convex in w or z which leads to faster convergence. The larger numerical stability also allows larger step sizes. However, it is sometimes better to work with a standard laplacian.

The algorithm for ADMM is as follows:

1. Fix the dual λ and block descent on the primal variables

$$\begin{aligned} w^{t+1} = argmin_w L_\mu(w, z^t; \lambda^t) \\ z^{t+1} = argmin_z L_\mu(z, w^{t+1}; \lambda^t) \end{aligned}$$

2. Fix the primal variables and gradient ascent on the dual

$$\lambda^{t+1} = \lambda^t + \eta(Aw^{t+1} + Bz^{t+1} - c)$$

The step size can be as large as μ.

For the ERM example, ADMM is:

$$w^{t+1} = \sum_i f_i(w_i) + \mu/2||w_i - z^t + \lambda^t||^2$$

Which is completely decoupled and parallelizable. A closed form exists if f is realatively simple.

The demading step in each iteration is computing the quadratic term. If A is diagonal this reduces to a proximal map of w. If A is not diagonal then we must perform the messy loop.

We can reduce to diagonal A by using a single gradient step or linearizing the quadratic at A. The intuition being that we re compute w anyway and so errors do not matter.

We can define a proximal map as

$$P_f^\mu(w) = argmin_w 1/2\mu||z-w||_2^2 + f(z)$$

and a reflection map as

$$R_f^\mu(w) = 2P_f^\mu(w) - w$$

These are well defined for convex f. We can split up the lagrangian as

$$L(x,z;y) = min_x f(x) + y^T Ax + min_z g(z) + y^T(Bz-c)$$

The first part is denoted as $d_1(y)$ and the second as $d_2(y)$. ADMM is then the same as Douglas-Rachford splitting.

$$\begin{aligned} w = 1/2(w + R_{d_1}^\mu(R_{d_2}^\mu(w)));\\ y = P_d^\mu 2(w) \end{aligned}$$

This means that ADMM is fixed point iteration and that it converges. This also explains why the dual and not the primal converges.

3 Applying ADMM to CLIME

3.1 Update Rules

Recall that in CLIME, we want to solve

$$\min_Q = ||Q||_1 \ s.t \ ||SQ-E||_\infty \leq \lambda \tag{12}$$

Where Q now denotes a column, and E is the corresponding column in the identity matrix. Now, to solve the problem

Step 1: Reduce to ADMM canonical form

$$\min_{Q,Z} ||Q||_1 + [||Z-E||_\infty \leq \lambda] \ s.t. \ Z = SQ \tag{13}$$

Step 2: Write out augmented Lagrangian

$$L(Q, Z; Y) = ||Q||_1 + [||Z - E||\infty \leq] + \rho \text{tr}[(SQ - Z)Y] + \frac{\rho}{2}||SQ - Z||_F^2 \tag{14}$$

Step 3: Perform primal-dual updates

$$\begin{aligned} Q &\leftarrow \text{argmin}_Q ||Q||_1 + \frac{\rho}{2}||SQ - Z + Y||_F^2 \\ Z &\leftarrow \text{argmin}_Z [||Z - E||_\infty \leq \lambda] + \frac{\rho}{2}||SQ - Z + Y||_F^2 \\ &= \text{argmin}_{||Z-E||_\infty \leq \lambda} \frac{\rho}{2}||SQ - Z + Y||_F^2 \\ Y &\leftarrow Y + SQ - Z \end{aligned} \tag{15}$$

Step 4:

Solve the sub-problems.

Solve the update for Q As a lasso problem, the update equation for Q can be solved using existing Lasso techniques, but that would lead to a double-loop algorithm. It doesn't have a closed-form solution since S in the quadratic penalty makes Q coupled. We decouple Q by linearizing the quadratic penalty term and adding a proximal term as follows

$$Q = \text{argmin}_Q ||Q||_1 + \rho \text{tr}(Q^T S(Y + SQ_t - Z)) + \frac{\eta}{2}||Q - Q_t||_F^2 \tag{16}$$

which has the closed-form solution

$$Q = soft(Q - \frac{\rho}{\eta} S(Y + SQ_t - Z), \eta^{-1}) \tag{17}$$

Solve the update for Z

This is a box quadratic programming which has the closed-form solution as the following

$$Z = box(SQ + Y, E, \lambda) \tag{18}$$

where *box* denotes the projection onto the infinity norm constraint $||Z - E||_\infty \leq \lambda$

Solve the Update for Y

It's trivial :)

3.2 Exploring Structure

The matrix-matrix multiplication is the expensive step in ADMM-CLIME, such as $U = SQ$. For the case of SQ, because $S = AA^T$, we can do $A(A^TQ)$ instead if $p >> n$.

Also, we can change matrix-matrix multiplication into a for loop of matrix-vector multiplication

3.3 Parallization

- If A fits into memory, we can choose to use embarrassingly parallel algorithm which leverages multiple cores in the same machine
- If not, we can do it in a distributed way. We chop A into small blocks and distribute to multiple machines

The work from Wang (2013) introduce a block-cyclic method to deal with the imbalance problem in the distributed setting.

4 Nonparanormal extensions

Suppose we have

$$Z_i = f_i(X_i), i = 1, \ldots, p \tag{19}$$
$$(Z_1, \ldots, Z_p) \sim N(0, \Sigma) \tag{20}$$

where f_i are unknown monotonic functions. We only observe X, but not Z.

Note that independence preserved under transformation, so we have

$$X_i \perp X_j | X_{-i,j} \leftrightarrow Z_i \perp Z_j | Z_{-i,j} \leftrightarrow Q_{ij} = 0 \tag{21}$$

So we can estimate f_i first, then apply glasso on $f_i(X_i)$.

We want Σ, but since we don't observe Z, we need to use rank approximate Z-estimator R.

And we have

$$\tau_{ij} = \frac{1}{n(n-1)} \sum_{k,l} \text{sign}[(R_{i,k} - R_{i,l})(R_{j,k} - R_{j,l})] \tag{22}$$
$$\Sigma_{ij} = 2 \sin(\frac{\pi}{6} \mathbb{E}(\tau_{ij})) \tag{23}$$

We can use glasso algorithms after we have Σ.