Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.438 Algorithms for Inference
Fall 2020

Problem Set 2

Issued: Tuesday, Sept. 22, 2020 **Due:** Thursday, Oct. 08, 2020

Problem 2.1
In this problem, we will take a closer look at the KL-divergence and M-projection.

(a) Consider three random variables $x, y, z \in \{0, 1\}$, and their joint probability mass function

$$\begin{aligned} p_{x,y,z}(0,0,0) &= 0.25,\ p_{x,y,z}(0,0,1) = 0.05, \\ p_{x,y,z}(0,1,0) &= 0.05,\ p_{x,y,z}(0,1,1) = 0.20, \\ p_{x,y,z}(1,0,0) &= 0.05,\ p_{x,y,z}(1,0,1) = 0.15, \\ p_{x,y,z}(1,1,0) &= 0.10,\ p_{x,y,z}(1,1,1) = 0.15. \end{aligned}$$

(i) Compute the marginal distributions p_x, p_y, p_z and the KL-divergence $D_{\mathrm{KL}}(p_{x,y,z} \parallel p_x p_y p_z)$. Use natural log to calculate the KL-divergence. Recall that $p_x p_y p_z$ is the M-projection of $p_{x,y,z}$ onto a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{x, y, z\}$ and $\mathcal{E} = \varnothing$.

(ii) Now consider the following four DAGs:

$$x \to y \to z \quad x \leftarrow y \to z \quad x \leftarrow y \leftarrow z \quad x \to y \leftarrow z.$$

Compute the KL-divergences between $p_{x,y,z}$ and the "factorizations" corresponding to these DAGs, i.e., compute

$$\begin{aligned} &D_{\mathrm{KL}}(p_{x,y,z} \parallel p_x p_{y|x} p_{z|y}), \quad D_{\mathrm{KL}}(p_{x,y,z} \parallel p_{x|y} p_y p_{z|y}), \\ &D_{\mathrm{KL}}(p_{x,y,z} \parallel p_{x|y} p_{y|z} p_z), \quad D_{\mathrm{KL}}(p_{x,y,z} \parallel p_x p_{y|x,z} p_z). \end{aligned}$$

Based on the KL-divergence values, which of four factorizations is the best approximation of the joint distribution $p_{x,y,z}$?

(b) Let p be a distribution on $x_1, \ldots, x_N$ and let $\mathcal{Q}$ be the set of distributions that factorize according to a DAG $\mathcal{G}$ on $x_1, \ldots, x_N$. Then, show that the M-projection of p onto the DAG $\mathcal{G}$ is

$$q^{\mathrm{M}} = \prod_{i=1}^{N} p_{x_i | x_{\pi_i}} = \arg\min_{q \in \mathcal{Q}} D_{\mathrm{KL}}(p \parallel q),$$

where π_i is the set of parent nodes of x_i in $\mathcal{G}$.

Problem 2.2

In this problem, we will investigate the MLE of the covariance matrix $\mathbf{\Lambda}$, or equivalently, the information matrix $\mathbf{J}$ of a multivariate Gaussian distribution represented by an undirected graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathbf{\Lambda}$ be the covariance matrix of the distribution, and $\mathbf{S}$ be the sample covariance matrix obtained out of i.i.d. observations from the distribution. We assume throughout that $\mathbf{\Lambda}$ and $\mathbf{S}$ are both positive definite. Given a matrix $\mathbf{A}$ and index sets $\mathcal{I}$ and $\mathcal{J}$, we use $[\mathbf{A}]_{\mathcal{I},\mathcal{J}}$ to denote a submatrix of $\mathbf{A}$ consisting of rows in $\mathcal{I}$ and columns in $\mathcal{J}$. Given any subset $\mathcal{I} \subset \mathcal{V}$, we let $\mathbf{\Lambda}_{\mathcal{I}} = [\mathbf{\Lambda}]_{\mathcal{I},\mathcal{I}}$ be the covariance matrix of the marginal distribution over $x_{\mathcal{I}}$. We also define $\mathbf{S}_{\mathcal{I}} = [\mathbf{S}]_{\mathcal{I},\mathcal{I}}$ in a similar way.

Hint: Throughout this problem, you can use the fact that for a given Gaussian graphical model $\mathcal{G}$, the MLE $\hat{\mathbf{J}}$ of its information matrix $\mathbf{J}$ is the unique solution to the following system of equations:

$$[\hat{\mathbf{J}}^{-1}]_{\mathcal{I},\mathcal{I}} = \mathbf{S}_{\mathcal{I}} \text{ for any maximal clique } \mathcal{I} \subset \mathcal{V} \text{ in } \mathcal{G},$$
$$[\hat{\mathbf{J}}]_{i,j} = 0 \text{ for any } (i,j) \notin \mathcal{E}.$$

(a) Consider a Gaussian graphcal model $\mathcal{G}_1$ in Fig. 1. Define index sets $\mathcal{A} = \{1,2\}$, $\mathcal{B} = \{2,3\}$, and $\mathcal{C} = \mathcal{A} \cap \mathcal{B} = \{2\}$.

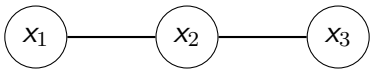


Figure 1: Undirected graph $\mathcal{G}_1$

(i) Show that the information matrix $\mathbf{J} = \mathbf{\Lambda}^{-1}$ of $\mathcal{G}_1$ satisfies

$$\mathbf{J} = \begin{bmatrix} \mathbf{\Lambda}_{\mathcal{A}}^{-1} & 0 \\ & 0 \\ 0 \quad 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & & \\ 0 & & \mathbf{\Lambda}_{\mathcal{B}}^{-1} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{\Lambda}_{\mathcal{C}}^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{1}$$

(ii) Show that the MLE $\hat{\mathbf{J}}$ for $\mathbf{J}$ satisfies

$$\hat{\mathbf{J}} = \begin{bmatrix} \mathbf{S}_{\mathcal{A}}^{-1} & 0 \\ & 0 \\ 0 \quad 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & & \\ 0 & & \mathbf{S}_{\mathcal{B}}^{-1} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{S}_{\mathcal{C}}^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

(b) Now consider another Gaussian graphical model $\mathcal{G}_2$ in Fig. 2. Define index sets $\mathcal{C}_1 = \{1,2\}, \mathcal{C}_2 = \{2,3\}, \mathcal{C}_3 = \{2,4\}, \mathcal{D} = \{2\}$.

(i) For any index set $\mathcal{I} \subset \mathcal{V}$, let $[\mathbf{\Lambda}_{\mathcal{I}}^{-1}]^{\text{fill}}$ denote a matrix $\mathbf{M}$ whose submatrix $[\mathbf{M}]_{\mathcal{I},\mathcal{I}} = \mathbf{\Lambda}_{\mathcal{I}}^{-1}$, while all the remaining entries are filled with zero. For example, the information matrix $\mathbf{J}$ of $\mathcal{G}_1$ (1) in Part (a).(i) can be written as $\mathbf{J} = [\mathbf{\Lambda}_{\mathcal{A}}^{-1}]^{\text{fill}} + [\mathbf{\Lambda}_{\mathcal{B}}^{-1}]^{\text{fill}} - [\mathbf{\Lambda}_{\mathcal{C}}^{-1}]^{\text{fill}}$.

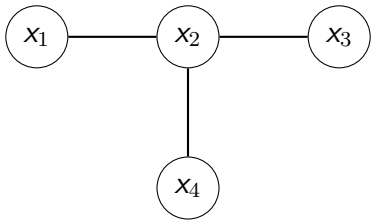


Figure 2: Undirected graph $\mathcal{G}_1$

Show that the information matrix $\mathbf{J}$ of $\mathcal{G}_2$ satisfies

$$\mathbf{J} = \sum_{j=1}^{3} [\mathbf{\Lambda}_{\mathcal{C}_j}^{-1}]^{\text{fill}} - 2[\mathbf{\Lambda}_{\mathcal{D}}^{-1}]^{\text{fill}}.$$

(ii) Show that the MLE $\hat{\mathbf{J}}$ of the information matrix $\mathbf{J}$ is

$$\hat{\mathbf{J}} = \sum_{j=1}^{3} [\mathbf{S}_{\mathcal{C}_j}^{-1}]^{\text{fill}} - 2[\mathbf{S}_{\mathcal{D}}^{-1}]^{\text{fill}}.$$

(iii) Briefly explain how you would extend Parts (b).(i) and (b).(ii) to general trees. What would $\mathbf{J}$ and $\hat{\mathbf{J}}$ look like if the graph is a tree? You don't have to prove it, but justify your guess.

The remaining two (optional) parts generalize the investigation above to chordal graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and show that the MLE of $\mathbf{J}$ in case of chordal graphs has a nice closed form solution.

(c) (optional) Suppose that $\mathcal{V} = \mathcal{A} \cup \mathcal{B}$, and define $\mathcal{C} = \mathcal{A} \cap \mathcal{B}$. Assume that the nodes in $\mathcal{C}$ form a clique, and there are no edges between $\mathcal{A} \setminus \mathcal{C}$ and $\mathcal{B} \setminus \mathcal{C}$. For any such sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, we say that $(\mathcal{A} \setminus \mathcal{C}, \mathcal{B} \setminus \mathcal{C}, \mathcal{C})$ form a decomposition of $\mathcal{G}$.

(i) Show that the information matrix $\mathbf{J} = \mathbf{\Lambda}^{-1}$ satisfies

$$\mathbf{J} = [\mathbf{\Lambda}_{\mathcal{A}}^{-1}]^{\text{fill}} + [\mathbf{\Lambda}_{\mathcal{B}}^{-1}]^{\text{fill}} - [\mathbf{\Lambda}_{\mathcal{C}}^{-1}]^{\text{fill}}.$$

(ii) Suppose that if we restrict our observations to $\mathcal{A}$, to $\mathcal{B}$, or to $\mathcal{C}$, the MLEs for $\mathbf{J}_{\mathcal{A}}$, $\mathbf{J}_{\mathcal{B}}$, and $\mathbf{J}_{\mathcal{C}}$ are $\hat{\mathbf{J}}_{\mathcal{A}}$, $\hat{\mathbf{J}}_{\mathcal{B}}$, and $\hat{\mathbf{J}}_{\mathcal{C}}$, respectively. Then, show that the MLE $\hat{\mathbf{J}}$ of the information matrix $\mathbf{J}$ is

$$\hat{\mathbf{J}} = [\hat{\mathbf{J}}_{\mathcal{A}}]^{\text{fill}} + [\hat{\mathbf{J}}_{\mathcal{B}}]^{\text{fill}} - [\hat{\mathbf{J}}_{\mathcal{C}}]^{\text{fill}}.$$

(d) (optional) Now suppose that $\mathcal{V} = \bigcup_{j=1}^{r} \mathcal{C}_j$ where $\mathcal{C}_1, \ldots, \mathcal{C}_r$ are cliques. Let $\mathcal{H}_j = \bigcup_{i=1}^{j} \mathcal{C}_i$ for $j = 1, \ldots, r$, and $\mathcal{S}_j = \mathcal{H}_j \cap \mathcal{C}_{j+1}$ for $j = 1, \ldots, r-1$. Assume that each $\mathcal{S}_j$ is nonempty, hence is also a clique. Assume that for $j = 1, \ldots, r-1$, $(\mathcal{H}_j \setminus \mathcal{S}_j, \mathcal{C}_{j+1} \setminus \mathcal{S}_j, \mathcal{S}_j)$ form a decomposition of $\mathcal{G}_{\mathcal{H}_{j+1}}$, the induced subgraph of $\mathcal{G}$ with vertices in $\mathcal{H}_{j+1}$.

(i) Show that the graph $\mathcal{G}$ is chordal. In fact, any chordal graph can be decomposed into a tree of cliques that only overlap in cliques, so the converse of the statement is also true (you don't have to show this).

(ii) Show that the information matrix $\mathbf{J}$ satisfies

$$\mathbf{J} = \sum_{j=1}^{r} [\mathbf{\Lambda}_{\mathcal{C}_j}^{-1}]^{\text{fill}} - \sum_{j=1}^{r-1} [\mathbf{\Lambda}_{\mathcal{S}_j}^{-1}]^{\text{fill}}.$$

(iii) Show that the MLE $\hat{\mathbf{J}}$ of the information matrix $\mathbf{J}$ is

$$\hat{\mathbf{J}} = \sum_{j=1}^{r} [\mathbf{S}_{\mathcal{C}_j}^{-1}]^{\text{fill}} - \sum_{j=1}^{r-1} [\mathbf{S}_{\mathcal{S}_j}^{-1}]^{\text{fill}}.$$

Problem 2.3

A *Brownian Motion Tree Model* is a Gaussain Graphical Model defined on a DAG $\mathcal{G}$, under the condition that $\mathcal{G}$ is a *rooted tree*. A rooted tree on the vertex set $\mathcal{V}$ has a root r with no parents and exactly one child, and every other node in $\mathcal{V} \setminus \{r\}$ has exactly one parent. The leaves $\mathcal{L}$ are the set of nodes with no children. We assign to each directed edge e the weight $\alpha_e \geq 0$. We associate with each node a random variable such that $x_r = 0$ and for $i \in \mathcal{V} \setminus \{r\}$

$$x_i = x_{\pi_i} + \varepsilon_{(\pi_i, i)},$$

where $\varepsilon_e \sim \mathcal{N}(0, \alpha_e)$ are independent. We would like to understand the distribution of the leaves $p_{x_{\mathcal{L}}}(x_{\mathcal{L}})$.

(As a side note, the Brownian Motion Tree Model is used in phylogenetics to model continuous traits such as skull size. The leaves correspond to species which are alive, while the interior nodes are extinct species.)

(a) The *least common ancestor* of a pair of nodes u and v is the node ℓ such that u and v are both descendants of ℓ, and no descendant of ℓ also has both u and v as descendants.

Show that the covariance between two leaf nodes x_u and x_v can be expressed as

$$\mathbf{Cov}(x_u, x_v) = \sum_{i=1}^{K} \alpha_{e_i}$$

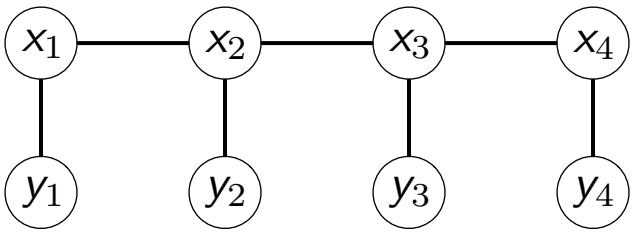
where $(e_1, e_2, \ldots, e_K)$ is the directed path from the root r to the least common ancestor ℓ of u and v (We can think of this quantity as the total time that species u and v have evolved together).

(b) Let $\{x_i\}_{i \in V}$ be a collection of random variables, and let $A, B \subset V$ such that $|A \cap B| = 1$ and $x_{A \setminus B} \perp\!\!\!\perp x_{B \setminus A} \mid x_{A \cap B}$. Show that x_A and x_B are MTP2 if and only if $x_{A \cup B}$ is MTP2. Intuitively, this lets us construct an MTP2 distribution by "gluing together" two MTP2 distributions which intersect at a single node.

(c) Use the previous part to show that in the Brownian Motion Tree Model, the joint distribution of all nodes $p_{x_V}(x_V)$ is MTP2. Conclude that the distribution of the leaves $p_{x_L}(x_L)$ is also MTP2.

Problem 2.4 (practice)
In this problem, we consider the elimination algorithm for inference and its computational complexity.

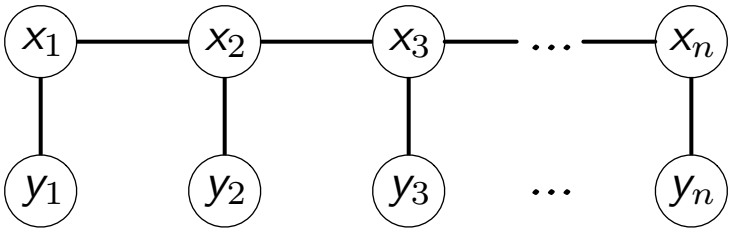
(a) Consider the following graph on 8 nodes.



Draw the reconstituted graph induced by the elimination ordering

$$(x_4, y_4, y_3, x_3, x_1, x_2, y_2, y_1).$$

(b) Now consider a graph on $2n$ nodes as drawn in the following figure, in which every random variable takes on values from a finite alphabet of size k. (That is, $\forall i \in \{1, \ldots, n\},\ x_i \in \mathcal{X},\ y_i \in \mathcal{Y}$ and $|\mathcal{X}| = |\mathcal{Y}| = k$.)



Describe an elimination ordering that requires the least computation and one that requires the most computation. Determine the asymptotic time complexity of the algorithm for each of these orderings with respect to k and n.

(c) Give an example of an undirected graph on n nodes such that the maximal clique size is constant with respect to n (i.e., the maximal clique size does *not* depend on n), but where the computation time required to perform elimination with any ordering is proportional to $k^{\alpha n}$, where k is the size of the alphabet. Specify the value of α for your example. Note that depending on the elimination ordering, your graph may have different values of α. However, your graph should be such that α is lower-bounded by some positive constant across all elimination orderings.

Problem 2.5

Let $\mathcal{G}$ be an undirected graph, and u a vertex of $\mathcal{G}$. Consider running the elimination algorithm on $\mathcal{G}$ to compute the marginal $p_u(\cdot)$. Let $\mathcal{S}_i$ be the set of all nodes (not including i and previously eliminated nodes) that share an active potential with node i, at the time of eliminating node i. Recall that the list of active potentials contains all potentials (including input potentials and intermediate computations) which have not yet been processed. Assume that there is one input potential for each maximal clique of $\mathcal{G}$. Let $\mathcal{H}$ be the resulting reconstituted graph.

(a) Prove that the largest clique size in $\mathcal{H}$ is $\max_i |\mathcal{S}_i| + 1$.

(b) Prove that $\mathcal{H}$ is chordal.

Problem 2.6

A 1986 medical study compared the efficacy of two treatments (denoted by t): open surgical procedures (Treatment A) and a minimally-invasive procedure called percutaneous nephrolithotomy (Treatment B) for alleviating kidney stones. The efficacy of each treatment was based on the recovery (r) of the patients, with $\mathsf{r} = 1$ denoting that the patients had no kidney stones 3 months after treatment and $\mathsf{r} = 0$ denoting that they still had kidney stones. As a doctor, we might be interested in the quantities $p_{\mathsf{r}|\mathrm{do}(\mathsf{t}=A)}(1)$ and $p_{\mathsf{r}|\mathrm{do}(\mathsf{t}=B)}(1)$ when deciding which of the two treatments to give a new patient. The patients are stratified according to the diameter d of their kidney stones (ℓ for large vs. s for small). The following table summarizes the number of patients belonging to the categories according to (d, t), and their results r:

(d, t)	$\mathsf{r} = 1$	$\mathsf{r} = 0$
(s, A)	81	6
(s, B)	234	36
(ℓ, A)	192	71
(ℓ, B)	55	25

(a) Consider the causal DAG $\mathcal{G}_1$ in Fig. 3. Note that we take a naive approach to the problem and ignore the effect of the diameter d. Based on the table, estimate $p_{r|\text{do}(t=A)}$ and $p_{r|\text{do}(t=B)}$ according to $\mathcal{G}_1$.

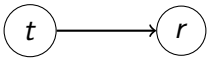


Figure 3: Causal DAG $\mathcal{G}_1$

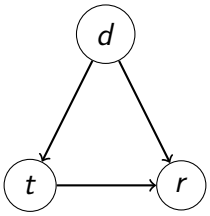


Figure 4: Causal DAG $\mathcal{G}_2$

(b) A closer look into the table reveals that the size of the kidney stones has an effect on the probability of recovery: patients with larger kidney stones are less likely to recover. Astutely, we might also notice that the size of the kidney stone seems to have an effect on which treatment a patient is given. If the doctors already expected Treatment A to perform better, perhaps they preferentially gave it to patients with large kidney stones, who required a better treatment. Thus, we may reasonably develop a new DAG $\mathcal{G}_2$ where d is a cause of both r and t, as in Fig. 4. Calculate the estimates for $p_{r|\text{do}(t=A)}$ and $p_{r|\text{do}(t=B)}$ according to $\mathcal{G}_2$.

(c) Compare the results in Parts (a) and (b). Which treatment would you prefer to use?

Computational Problem 2

Given N stocks, what is the optimal way to select a portfolio? Answering such a question often involves selecting some minimum variance portfolio, and requires estimating the covariance matrix $\mathbf{\Sigma}$ of the stock prices. Say that we have T days of returns from N stocks, where $\mathbf{x}_t = \begin{bmatrix} x_{1,t} & \cdots & x_{N,t} \end{bmatrix}^T$ is the vector of returns of each stock on day t. We assume that each day's returns is an independent draw from a Gaussian distribution

$$\begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^T \sim \mathcal{N}(\mu, \mathbf{\Sigma}),$$

for unknown $\mu, \mathbf{\Sigma}$. Since $\mathbf{x}$ is distributed as a Gaussian, it induces an underlying undirected Gaussian Graphical Model on the vertex set $\mathcal{V} = \{1, \ldots, N\}$. Our goal is to estimate $\mathbf{\Sigma}$ given the data.

In the attached "ret.csv" file are the returns (percentage increase/decrease from the previous day) of 50 different stocks over a 5 year span from 2011-2015. Each column is a different stock, while each row is a different day.

(a) In the last problem set, we showed that for a multivariate Gaussian, the maximum likelihood estimate for $\mathbf{\Sigma}$ is just the sample covariance

$$\mathbf{S} = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_t - \hat{\mu})(\mathbf{x}_t - \hat{\mu})^T,$$

where $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t$ is the sample mean.

(i) Compute the sample covariance matrix $\mathbf{S}$ of the data.

(ii) What does this tell you about the underlying Gaussian graphical model with covariance matrix $\mathbf{S}$?

(b) We may instead believe that the underlying Gaussian graphical model is *sparse*, i.e. has few edges. One method to encourage sparsity is *Graphical Lasso*, which minimizes the log likelihood but with an ℓ_1 penalty on the precision matrix to encourage sparsity. Use the scikit-learn package for Graphical Lasso to produce different estimates for the covariance matrix. Set the regularization parameter α equal to $10^{-5}.10^{-4}, 10^{-3}$, and in each case compute the number of edges in the underlying Gaussian graphical model. How does the sparsity change as you vary α?

(c) A stronger assumption we can impose on the data is that it is sampled from a Gaussian MTP2 distribution. Such an assumption makes a lot of sense for financial data, as the stocks are often strongly correlated with each other and move up and down together. Let us consider the problem of maximum likelihood estimation under this MTP2 constraint. In class we showed that computing the MLE is equivalent to finding the information matrix $\mathbf{J}$ which maximizes

$$\hat{\mathbf{J}} = \arg\max_{\mathbf{J}} \log|\mathbf{J}| - \text{trace}(\mathbf{JS})$$

We also showed in class that for a Gaussian distribution, the MTP2 constraint is equivalent to the constraint that the information matrix is an M-matrix, i.e. that $\mathbf{J}_{ij} \leq 0$ whenever $i \neq j$. Thus computing the MLE under the MTP2 constraint is equivalent to the following constrained optimization problem:

$$\begin{aligned} &\text{maximize } \log|\mathbf{J}| - \text{trace}(\mathbf{JS}) \\ &\text{subject to } \mathbf{J}_{ij} \leq 0 \end{aligned}$$

One way to solve this problem is via *coordinate ascent*. In each step of coordinate ascent, we choose a pair of variables $u, v \in \mathcal{V}$, and solve the constrained optimization problem such that all entries of $\mathbf{J}$ besides the submatrix

$$\begin{bmatrix} \mathbf{J}_{uu} & \mathbf{J}_{uv} \\ \mathbf{J}_{vu} & \mathbf{J}_{vv} \end{bmatrix},$$

are held constant. In other words, the coordinate ascent algorithm is as follows:

- Repeat until convergence:
  - Loop through all pairs of entries u, v:
    - Compute $\begin{bmatrix} \mathbf{J}_{uu} & \mathbf{J}_{uv} \\ \mathbf{J}_{vu} & \mathbf{J}_{vv} \end{bmatrix}$ which maximizes $\log|\mathbf{J}| - \text{trace}(\mathbf{JS})$, subject to $\mathbf{J}_{uv} \leq 0$

For the remainder of this problem, assume that $\mathbf{S}$ is positive definite.

(i) Let us first compute the coordinate ascent update rule for the unconstrained optimization problem. Say that we have selected $A = \{u, v\}$, and want to compute $\mathbf{J}_{AA} = \begin{bmatrix} \mathbf{J}_{uu} & \mathbf{J}_{uv} \\ \mathbf{J}_{vu} & \mathbf{J}_{vv} \end{bmatrix}$ which maximizes $\log|\mathbf{J}| - \text{trace}(\mathbf{JS})$. Show that the update is given by

$$\mathbf{J}_{AA} \leftarrow (\mathbf{S})^{-1}_{AA} + \mathbf{J}_{AB}\mathbf{J}^{-1}_{BB}\mathbf{J}_{BA},$$

where $B = \mathcal{V} \setminus A$. Additionally, show that this update keeps $\mathbf{J}$ positive definite.

(Hint: write the objective in terms of the Schur complement of B, and optimize with respect to this Schur complement)

(ii) In the constrained setting, one can show that the update is given by the following:

$$\mathbf{J}_{AA} \leftarrow (\mathbf{S})^{-1}_{AA} + \mathbf{J}_{AB}\mathbf{J}^{-1}_{BB}\mathbf{J}_{BA}$$

if $[(\mathbf{S})^{-1}_{AA} + \mathbf{J}_{AB}\mathbf{J}^{-1}_{BB}\mathbf{J}_{BA}]_{12} \leq 0$, and

$$\mathbf{J}_{AA} \leftarrow \begin{bmatrix} \mathbf{L}_{11} + \frac{1+\sqrt{1+4\mathbf{S}_{uu}\mathbf{S}_{vv}\mathbf{L}_{12}^2}}{2\mathbf{S}_{uu}} & 0 \\ 0 & \mathbf{L}_{22} + \frac{1+\sqrt{1+4\mathbf{S}_{uu}\mathbf{S}_{vv}\mathbf{L}_{12}^2}}{2\mathbf{S}_{vv}} \end{bmatrix}$$

otherwise, where $\mathbf{L} = \mathbf{J}_{AB}\mathbf{J}^{-1}_{BB}\mathbf{J}_{BA}$. (As a fun exercise, if you know some convex optimization you can try to prove this formula!)

Implement this coordinate ascent update rule, and use it to find the MLE under the MTP2 constraint. Analyze the underlying graph - how many edges does it have? How does it compare to the graphs you computed using Graphical Lasso?

(Hint: To make sure your code is correct, start of with $\mathbf{S}$ being the covariance matrix for an MTP2 Gaussian. The algorithm should then converge to $\mathbf{J} = \mathbf{S}^{-1}$, since in this case the unconstrained optimum is MTP2.)