# 6.438 Fall 2020 Problem Set 2

Rylan Schaeffer

## Problem 1

### 1a

I calculate the following KL Divergences:

1. 2.1(a)(i): [0.15220929]
2. 2.1(a)(ii)(a): [0.11645682]
3. 2.1(a)(ii)(b): [0.11645682]
4. 2.1(a)(ii)(c): [0.11644002]
5. 2.1(a)(ii)(d): [0.02275236]

Based on these results, I think the fourth distribution $p_x p_{y|x,z} p_z$ is the best approximation of the joint distribution $p_{x,y,z}$.

Code:

```
from itertools import product
import numpy as np
import pandas as pd


binary = (0, 1)

arrays = [
    [0, 0, 0, 0.25],
    [0, 0, 1, 0.05],
    [0, 1, 0, 0.05],
    [0, 1, 1, 0.20],
    [1, 0, 0, 0.05],
    [1, 0, 1, 0.15],
    [1, 1, 0, 0.10],
    [1, 1, 1, 0.15]]

joint = pd.DataFrame(arrays, columns=['x', 'y', 'z', 'p_xyz'])

# calculate marginals
p_x = joint.groupby(['x'])['p_xyz'].sum()
p_y = joint.groupby(['y'])['p_xyz'].sum()
p_z = joint.groupby(['z'])['p_xyz'].sum()

# calculate conditionals
p_x_given_y = joint.groupby(['y']).apply(
    lambda group: group.groupby(['x'])['p_xyz'].sum())
```

```
p_x_given_y = p_x_given_y / p_x_given_y.sum(axis=0)

p_y_given_x = joint.groupby(['x']).apply(
    lambda group: group.groupby(['y'])['p_xyz'].sum())
p_y_given_x = p_y_given_x / p_y_given_x.sum(axis=0)

p_y_given_z = joint.groupby(['z']).apply(
    lambda group: group.groupby(['y'])['p_xyz'].sum())
p_y_given_z = p_y_given_z / p_y_given_z.sum(axis=0)

p_z_given_y = joint.groupby(['y']).apply(
    lambda group: group.groupby(['z'])['p_xyz'].sum())
p_z_given_y = p_z_given_y / p_z_given_y.sum(axis=0)

p_y_given_xz = joint['p_xyz'].values.reshape(2, 2, 2).transpose(1, 0, 2)
p_y_given_xz = np.divide(
    p_y_given_xz,
    np.sum(p_y_given_xz, axis=0)[np.newaxis, :, :])

# 2.1 (a)(i)
total_kl = 0
for (x, y, z), outcome in joint.groupby(['x', 'y', 'z']):
    p_xyz = outcome['p_xyz'].values
    total_kl += p_xyz * np.log(p_xyz / (p_x[x] * p_y[y] * p_z[z]))
print('2.1(a)(i): ', total_kl)

# 2.1 (a)(ii)(a)
total_kl = 0
for x, y, z in product(binary, binary, binary):
    p_xyz = joint[(joint['x'] == x) & (joint['y'] == y) & (joint['z'] == z)]['p_xyz'].valu
    factorized = p_x[x] * p_y_given_x.loc[y, x] * p_z_given_y.loc[z, y]
    total_kl += p_xyz * np.log(p_xyz / factorized)
print('2.1(a)(ii)(a): ', total_kl)


# 2.1 (a)(ii)(b)
total_kl = 0
for x, y, z in product(binary, binary, binary):
    p_xyz = joint[(joint['x'] == x) & (joint['y'] == y) & (joint['z'] == z)]['p_xyz'].valu
    factorized = p_z[z] * p_y_given_z.loc[y, z] * p_x_given_y.loc[x, y]
    total_kl += p_xyz * np.log(p_xyz / factorized)
print('2.1(a)(ii)(b): ', total_kl)


# 2.1 (a)(ii)(c)
total_kl = 0
for x, y, z in product(binary, binary, binary):
    p_xyz = joint[(joint['x'] == x) & (joint['y'] == y) & (joint['z'] == z)]['p_xyz'].valu
    factorized = p_y[y] * p_x_given_y.loc[x, y] * p_z_given_y.loc[z, y]
    total_kl += p_xyz * np.log(p_xyz / factorized)
print('2.1(a)(ii)(c): ', total_kl)


# 2.1 (a)(ii)(d)
```

```
total_kl = 0
for x, y, z in product(binary, binary, binary):
    p_xyz = joint[(joint['x'] == x) & (joint['y'] == y) & (joint['z'] == z)]['p_xyz'].valu
    factorized = p_x[x] * p_z[z] * p_y_given_xz[y, x, z]
    total_kl += p_xyz * np.log(p_xyz / factorized)
print('2.1(a)(ii)(d): ', total_kl)
```

**1b**

2.1

b)

Goal: Show $D_{KL}(p \| q') - D_{KL}(p \| q^M) \geq 0 \qquad \forall q' \neq q^M$

$$= \int p(x) \log \frac{p(x)}{q'(x)}\, dx - \int p(x) \log \frac{p(x)}{q^M(x)}\, dx$$

$$= \mathbb{E}_{p(x)}\left[\log \frac{p(x)}{q'(x)}\right] - \mathbb{E}_{p(x)}\left[\log \frac{p(x)}{q^M(x)}\right] \qquad \text{defn of KL}$$

$$= -\mathbb{E}_{p(x)}\left[\log q'(x)\right] + \mathbb{E}_{p(x)}\left[\log q^M(x)\right] \qquad \text{entropies annihilate}$$

$$= \mathbb{E}_{p(x)}\left[\log \frac{q^M(x)}{q'(x)}\right]$$

$$= \mathbb{E}_{p(x)}\left[\log \prod_{i=1}^{N} \frac{q^M(x_i | x_{\pi_i})}{q'(x_i | x_{\pi_i})}\right] \qquad \text{Both } q' \text{ \& } q^M \text{ factorize according to graph}$$

$$= \sum_{i=1}^{N} \mathbb{E}_{p(x)}\left[\log \frac{p(x_i | x_{\pi_i})}{q'(x_i | x_{\pi_i})}\right] \qquad \text{Property that } q^M(x_i | x_{\pi_i}) = p(x_i | x_{\pi_i})$$

$$= \sum_{i=1}^{N} \mathbb{E}_{p(x_{\pi_i})}\Bigg[\underbrace{\mathbb{E}_{p(x_i | x_{\pi_i})}\left[\log \frac{p(x_i | x_{\pi_i})}{q'(x_i | x_{\pi_i})}\right]}_{D_{KL}\left(p(x_i | x_{\pi_i}) \,\|\, q'(x_i | x_{\pi_i})\right)}\Bigg] \qquad \text{Law of iterated expectation}$$

$\geq 0$ b/c all terms in integrals are KL divergences $\geq 0$

$\geq 0$

$\square$

# Problem 2

## 2a

2.2

a) (i) $p(x_1, x_2, x_3) = p(x_1, x_2)\, p(x_3 | x_2)$

$$= \frac{p(x_1, x_2)\, p(x_2, x_3)}{p(x_2)}$$

$$\propto \exp\left\{ -\frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^T \underbrace{\begin{bmatrix} \Lambda_A^{-1} & & 0 \\ & & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{J_A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^T \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \Lambda_B^{-1} & \\ 0 & & \end{bmatrix}}_{J_B} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^T \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & \Lambda_C^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{J_C} \right.$$

$$\Rightarrow J_{1,2,3} = \begin{bmatrix} \Lambda_A^{-1} & & 0 \\ & & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Lambda_B^{-1} & \\ 0 & & \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Lambda_C^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(ii) Given: $\left[\hat{J}^{-1}\right]_{JJ} = S_J$. Define $\hat{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix}$

From Wikipedia: $S_J = \left(\hat{J} \setminus J_{\bar{J}\bar{J}}\right)^{-1}$ i.e. Schur complement of non $J$ elements

$$\Rightarrow \left[\hat{J}^{-1}\right]_A = S_A = \left(\hat{J} \setminus J_{33}\right)^{-1} = \left( \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} - \begin{bmatrix} J_{13} \\ J_{23} \end{bmatrix} \left[J_{33}\right]^{-1} \begin{bmatrix} J_{13} \\ J_{23} \end{bmatrix}^T \right)^{-1}$$

$$\left[\hat{J}^{-1}\right]_B = S_B = \left(\hat{J} \setminus J_{11}\right)^{-1} = \left( \begin{bmatrix} J_{22} & J_{23} \\ J_{23} & J_{33} \end{bmatrix} - \begin{bmatrix} J_{21} \\ J_{31} \end{bmatrix} \left[J_{11}\right]^{-1} \begin{bmatrix} J_{12} \\ J_{13} \end{bmatrix}^T \right)^{-1}$$

$$\left[\hat{J}^{-1}\right]_C = S_C = \left(\hat{J} \setminus J_{\{1,3\}\{1,3\}}\right)^{-1} = \left( \left[J_{22}\right] - \left[J_{21}\ J_{23}\right] \begin{bmatrix} J_{11} & J_{13} \\ J_{31} & J_{33} \end{bmatrix}^{-1} \begin{bmatrix} J_{12} \\ J_{13} \end{bmatrix}^T \right)^{-1}$$

Invert both sides, pad w/ zeros:

$$\begin{bmatrix} (S_A)^{-1} & & 0 \\ & & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & (S_B)^{-1} & \\ 0 & & \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & (S_C)^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$ those 3 inverted ↑

$$\text{Simplifying} \Rightarrow \begin{bmatrix} (S_A)^{-1} & & 0 \\ & & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & (S_B)^{-1} & \\ 0 & & \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & (S_C)^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \overset{\text{def}}{=} \hat{J}$$

**2b**

2.2

b) (i) $$p(x_1, x_2, x_3, x_4) = p(x_1, x_2)\, p(x_4 | x_2)\, p(x_3 | x_2)$$

$$= \frac{p(x_1, x_2)\, p(x_4, x_2)\, p(x_3, x_2)}{p(x_2)\, p(x_2)}$$

Define $\underline{x} = [x_1 \; x_2 \; x_3 \; x_4]^T$

$$p(x_2) \propto \exp\left\{ -\frac{1}{2} x^T \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & (\Lambda_D)^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x \right\}$$

$$p(x_1, x_2) \propto \exp\left\{ -\frac{1}{2} x^T \begin{bmatrix} (\Lambda_A)^{-1} & & 0 & 0 \\ & & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x \right\}$$

$$p(x_3, x_2) \propto \exp\left\{ -\frac{1}{2} x^T \left[ (\Lambda_C)^{-1} \right]^{\text{fill}} x \right\}$$

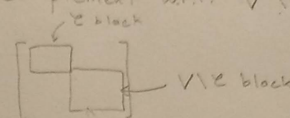
$$p(x_4, x_3) \propto \exp\left\{ -\frac{1}{2} x^T \left[ (\Lambda_D)^{-1} \right]^{\text{fill}} x \right\}$$

$$p(x_1 x_2 x_3 x_4) \propto \exp\left\{ -\frac{1}{2} x^T \underbrace{\left[ \sum_{j=1}^{3} \left[ (\Lambda_{C_j})^{-1} \right]^{\text{fill}} - 2 \left[ (\Lambda_D)^{-1} \right]^{\text{fill}} \right]}_{J} x \right\}$$

(ii) We use the same strategy as 2.2a(ii) i.e. define $\hat{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \\ J_{41} & J_{42} & J_{43} & J_{44} \end{bmatrix}$

$$\left[\hat{J}^{-1}\right]_{\mathcal{C}} = S_{\mathcal{C}} = \left(\hat{J} \setminus \overline{\mathcal{C}}\right)^{-1}$$ i.e. Schur Complement w.r.t. $V \setminus \mathcal{C}$ block

$$\Rightarrow \left(S_{\mathcal{C}}\right)^{-1} = \hat{J} \setminus \overline{\mathcal{C}}$$




$$\left(S_{e_1}\right)^{-1} = \hat{J} \setminus \overline{\mathcal{C}}_1 = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} - \begin{array}{c c} & \begin{array}{cc} 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} J_{13} & J_{14} \\ J_{23} & J_{24} \end{bmatrix} \end{array} \begin{bmatrix} J_{33} & J_{34} \\ J_{43} & J_{44} \end{bmatrix}^{-1} \begin{bmatrix} J_{13} & J_{23} \\ J_{14} & J_{24} \end{bmatrix}$$

$$\left(S_{e_2}\right)^{-1} = \hat{J} \setminus \overline{\mathcal{C}}_2 = \begin{bmatrix} J_{22} & J_{23} \\ J_{32} & J_{33} \end{bmatrix} - \begin{array}{c c} & \begin{array}{cc} 1 & 4 \end{array} \\ \begin{array}{c} 2 \\ 3 \end{array} & \begin{bmatrix} J_{21} & J_{24} \\ J_{31} & J_{34} \end{bmatrix} \end{array} \begin{bmatrix} J_{11} & J_{14} \\ J_{41} & J_{44} \end{bmatrix}^{-1} \begin{bmatrix} J_{21} & J_{31} \\ J_{24} & J_{34} \end{bmatrix}$$

$\left(S_{e_3}\right)^{-1} = \hat{J} \setminus \overline{\mathcal{C}}_3 =$ same but for $\mathcal{C} = \{2,4\}$ instead

$$\left(S_{e_4}\right)^{-1} = \hat{J} \setminus \overline{\mathcal{C}}_4 = \begin{bmatrix} J_{22} \end{bmatrix} - \begin{bmatrix} J_{21} & J_{23} & J_{24} \end{bmatrix} \begin{bmatrix} J_{11} & J_{13} & J_{14} \\ J_{31} & J_{33} & J_{34} \\ J_{41} & J_{43} & J_{44} \end{bmatrix}^{-1} \begin{bmatrix} J_{21} \\ J_{23} \\ J_{24} \end{bmatrix}$$

Write out $\sum_{j=1}^{3} \left[\left(S_{e_j}\right)^{-1}\right]^{fill} - 2\left[\left(S_{e_4}\right)^{-1}\right]^{fill}$

Simplifying yields $\hat{J}$

(iii) For each edge, $(i,j) \in \mathcal{E}$ we add $\left[\left(S_{A}\right)^{-1}\right]^{fill}$ for the MLE and $\left[\left(\Lambda_{A}\right)^{-1}\right]^{fill}$ for $J$, where $A = \{i,j\}$

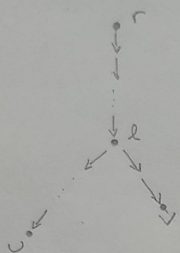
For each time node $i$ is counted beyond the first time, subtract $\left[\left(S_{\{i\}\times\{i\}}\right)^{-1}\right]^{fill}$

## Problem 3

### 3a

2.3.1

a) Let $\ell$ be the least common ancestor of $u, v$




$$X_u = X_{\pi(u)} + \varepsilon_u$$

$$= X_{\pi(\pi(u))} + \varepsilon_u + \varepsilon_{\pi(u)}$$

$$= X_r + \sum_{r \to \ell \to u} \varepsilon_i$$

Similarly, $X_v = X_r + \sum_{r \to \ell \to v} \varepsilon_j$

Both have mean 0: $\mathbb{E}[X_u] = \mathbb{E}[X_r] + \sum_{r \to \ell \to u} \mathbb{E}[\varepsilon_i]$

$$= 0 + \sum_{r \to \ell \to u} 0$$

$$= 0$$

$$\mathbb{E}[X_v] = \mathbb{E}[X_r] + \sum_{r \to \ell \to v} \mathbb{E}[\varepsilon_j]$$

$$= 0 + \sum 0$$

$$= 0$$

Covariance is then

$$\mathrm{Cov}[X_u, X_v] = \mathbb{E}[(X_u - \cancelto{0}{\mathbb{E}[X_u]})(X_v - \cancelto{0}{\mathbb{E}[X_v]})]$$

$$= \mathbb{E}[X_u X_v]$$

$$= \mathbb{E}\Big[\Big(X_r + \sum_{r \to \ell \to u} \varepsilon_i\Big)\Big(X_r + \sum_{r \to \ell \to v} \varepsilon_j\Big)\Big]$$

$$\mathbb{E}[\varepsilon_i \varepsilon_j] = \begin{cases} \alpha_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

$$= \mathbb{E}[X_r^2] + \sum_{r \to \ell} \mathbb{E}[\varepsilon_k^2]$$

$$= \alpha_0 + \sum_{r \to \ell} \alpha_k$$

**3b**

2.3

b) Note that $X_{A \cap B}$ is MTP2 by marginalization (either start w/ $X_A$ & marginalize $X_{A \setminus B}$ or similarly w/ $X_B$ then $X_{B \setminus A}$).

Note that $X_{A \setminus B} \mid X_{A \cap B}$ and $X_{B \setminus A} \mid X_{A \cap B}$ are MTP2 by conditioning

We want to show $X_A$ MTP2 & $X_B$ MTP2 $\Rightarrow$ $X_{A \cap B}$ MTP2

$$P(X_{A \cup B}) = P(X_{A \setminus B} \mid X_{A \cap B})\, P(X_{B \setminus A} \mid X_{A \cap B})\, P(X_{A \cap B})$$

All 3 RHS terms are MTP2. B/c the product of MTP2 distributions is MTP2, $X_{A \cup B}$ is MTP2

Other direction: we want to show $X_{A \cup B}$ is MTP2 $\Rightarrow$ $X_A$ MTP2, $X_B$ MTP2. Since MTP2 is closed under marginalization, just marginalize $X_{B \setminus A}$ to get the first result & $X_{A \setminus B}$ to get the second.

**3c**

$\boxed{2.3}$

c) Claim: Every node & its parent are jointly MTP-2

Proof: Consider $\{x_i, x_{\pi_i}\}$. We know $x_{\pi_i} \sim N$ w/ some variance $\alpha_{\pi_i}$ (mean 0 and)

By construction, $x_i = x_{\pi_i} + \varepsilon_i$ where $\varepsilon_i \sim N(0, \alpha_i)$

$$\Rightarrow \mathbb{V}[x_i] = \mathbb{V}[x_{\pi_i}] + \mathbb{V}[\varepsilon_i]$$

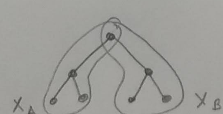
$$= \alpha_{\pi_i} + \alpha_i$$

Thus $\begin{bmatrix} x_i \\ x_{\pi_i} \end{bmatrix}$ has covariance $\Sigma_i = \begin{bmatrix} \alpha_{\pi_i} + \alpha_i & 0 \\ 0 & \alpha_i \end{bmatrix}$

$$\Rightarrow \Sigma_i^{-1} = \frac{1}{[\alpha_\pi + \alpha_i]\alpha_i} \begin{bmatrix} \alpha_i & 0 \\ 0 & \alpha_{\pi_i} + \alpha_i \end{bmatrix}$$

Consequently, b/c the joint $\begin{bmatrix} x_i \\ x_{\pi_i} \end{bmatrix}$ is Gaussian and its precision is a Stieltjes matrix, the joint is MTP2

Claim: Using 3(b), subtrees that overlap at 1 parent are MTP2




Per 3(b) $x_{A \cup B}$ is MTP2

Claim: The joint distr. of all nodes is MTP2

Proof: Use Claim 1 to merge subtree w/ parent. Use Claim 2 to merge 2 subtrees sharing parent. Recurse upward

Claim: The distr of all leaf nodes in MTP2

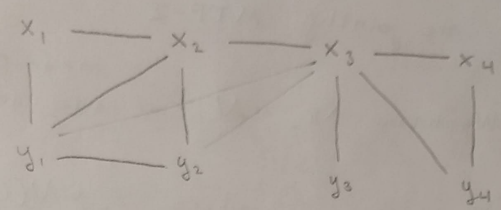
Proof: MTP2 is closed under marginalization. Marginalize non-leaf nodes

# Problem 4

### 4ab

2.4)

a)




b) Least computation:

Marginalization order: $(y_1 \dots y_n, x_1 \dots x_n)$

Time complexity: Marginalizing $y_i$ takes $O(|X||Y|) = O(k^2)$

Marginalizing $x_i$ takes $O(|X||X|) = O(k^2)$

Total complexity: $O(2nk^2) = \underline{O(nk^2)}$

Most computation:

Marginalization order: $(x_{\lfloor n/2 \rfloor}, x_{\lfloor n/2 \rfloor - 1}, x_{\lfloor n/2 \rfloor + 1}, \dots, x_1, x_n)$

Time complexity: Marginalizing $x_i$ takes $O(|Y|^i |X|^2) = O(k^{n+2})$

Marginalizing $y_i$ takes $O(|Y|^{n-i+1}) = O(k^n)$

Total complexity: $O\left(\sum_{i=1}^{n} k^{i+2} + \sum_{i=1}^{n} k^{n-i}\right)$

$= \underline{O(nk^n)}$

## Problem 5

### 5ab

2.5

(a) Claim: Eliminating node $i$ requires connecting all $j \in S_i$ pairwise

Claim: This creates a clique of size $\underbrace{|S_i|}_{\text{adjacent nodes}} + \underbrace{1}_{\text{node } i}$

Claim: Largest created clique is $\max_i |S_i| + 1$

Claim: A preexisting clique larger than $\max_i |S_i| + 1$ is not possible

Proof: By contradiction. Assume some big clique exists.

Elimination must eliminate all or all-but-one node

Eliminating first clique node has size $|S_i| + 1$

(b) Proof by contradiction. Assume $\exists$ cycle w/ length $N \geq 4$.

Elimination must eliminate all or all-but-one node.

Choose node $i$ & eliminate $\Rightarrow$ produces clique of size 3

& cycle of length $N-1$

Choose next node on cycle. Again, cycle length shrinks by 1.

All nodes but 1 must be eliminated $\Rightarrow$ cycle length continually decrements

# Problem 6

## 6abc

2.61

a) $$P_r^{do(t=A)}(r) = P(r \mid t=A)\, \mathbb{1}_{t=A}$$ ← or equivalently $P(r|t)\,\mathbb{1}_{t=A}$

$$= \frac{81+192}{81+6+192+71}$$

$$= 0.78$$

$$P_r^{do(t=B)}(r) = P(r \mid t=B)$$

$$= \frac{234+55}{234+36+55+25}$$

$$= 0.8257$$

b) $$P_r^{do(t=A)}(r=1) = \sum_{d\in\{S,L\}} P(r=1 \mid t=A, d)\, P(d)\, P(t=A)$$

$$= \frac{81}{81+6}\left(\frac{\sum \text{top 2 rows}}{\sum \text{all 4 rows}}\right) + \frac{192}{192+71}\left(\frac{\sum \text{bottom 2 rows}}{\sum \text{all 4 rows}}\right)$$

$$= 0.8325$$

$$P_r^{do(t=B)}(r=1) = \frac{234}{234+36}\left(\frac{\sum \text{top 2 rows}}{\sum \text{all 4 rows}}\right) + \frac{55}{55+25}\left(\frac{\sum \text{bottom 2 rows}}{\sum \text{all 4 rows}}\right)$$

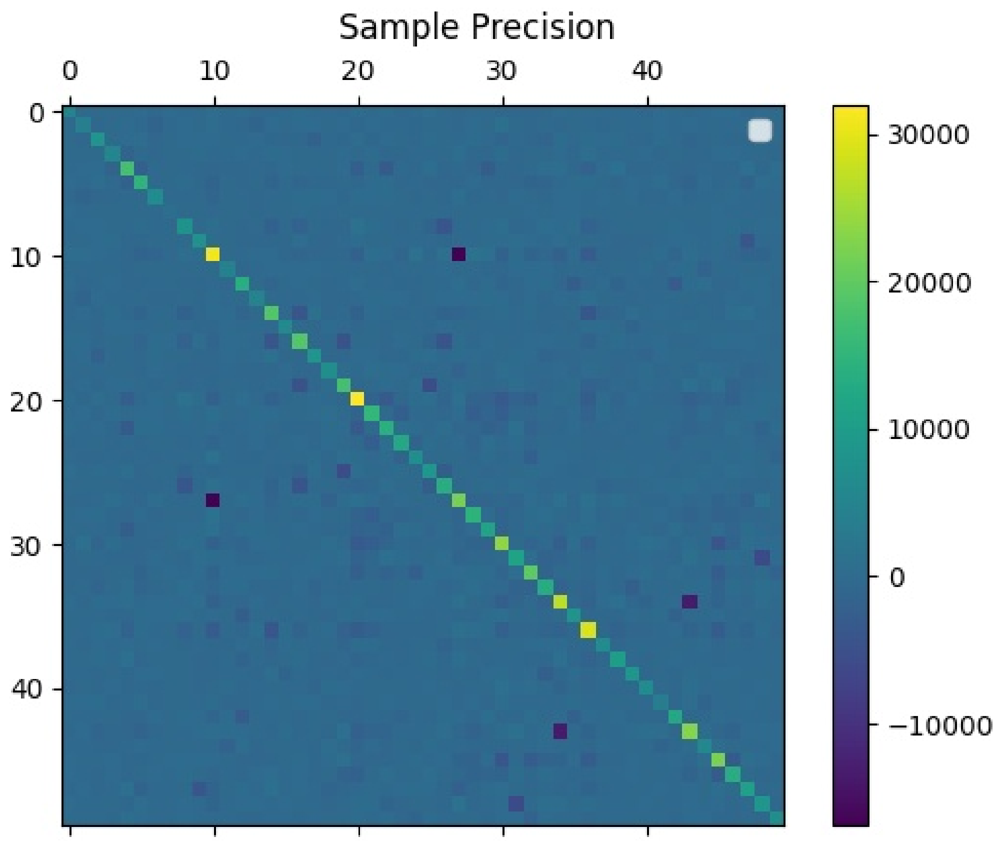
$$= 0.7789$$

c) I would prefer Treatment A b/c prob. of recovery higher taking into account how doctors refer patients

## Computational

### Comp a

All covariances are positive, with smallest value 3.63e-05. Many pairwise precisions are near 0, but not exactly 0, which suggests that the underlying undirected graphical model is not sparse.



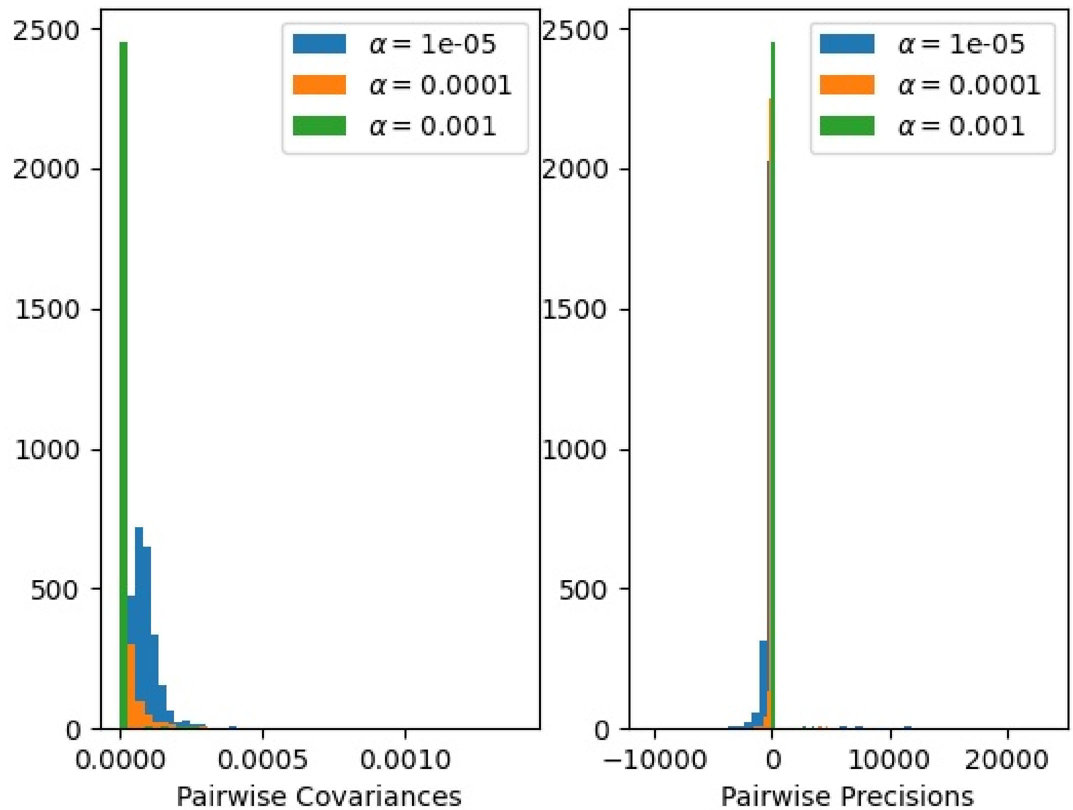

```
returns = pd.read_csv('returns.csv', header=None)

ksample_means = returns.mean().values
sample_cov = returns.cov().values
sample_prec = np.linalg.inv(sample_cov)
fig, ax = plt.subplots(nrows=1, ncols=1)
fig.title('Sample Precision')
cax = ax.matshow(sample_prec)
fig.colorbar(cax)
ax.legend()
plt.show()
plt.savefig('compa.jpg')
```

### Comp b

As alpha increases in size, the covariance elements shrink towards 0 and the sparsity of the graph increases. We can see that by how more precision values approach 0.

Number of Non-Zero Edges (alpha = 1e-05): 1132 Number of Non-Zero Edges (alpha = 0.0001): 538 Number of Non-Zero Edges (alpha = 0.001): 50




```
from sklearn.covariance import graphical_lasso


penalized_covs, penalized_precs = {}, {}
for alpha in [1e-5, 1e-4, 1e-3]:
    penalized_cov, penalized_prec = graphical_lasso(
        emp_cov=sample_cov,
        alpha=alpha,
        max_iter=1000)
    penalized_covs[alpha] = penalized_cov
    penalized_precs[alpha] = penalized_prec

fig, axes = plt.subplots(nrows=1, ncols=2)
axes[0].set_xlabel('Pairwise Covariances')
axes[1].set_xlabel('Pairwise Precisions')
for alpha, penalized_cov in penalized_covs.items():
    axes[0].hist(penalized_cov.flatten(),
                 label=r'$\alpha = $' + str(alpha),
                 bins=50)
    axes[1].hist(penalized_precs[alpha].flatten(),
                 label=r'$\alpha = $' + str(alpha),
                 bins=50)
```

```
axes[0].legend()
axes[1].legend()
plt.savefig('compb.jpg')
plt.show()
```

**Comp c**

Computational 2

c) (i) We want to maximize $\log|J| - Tr(JS)$ wrt $J_{AA}$

$$\log|J| - Tr|JS| = \log\left|J_{AA} - J_{AB}J_{BB}^{-1}J_{BA}\right| - Tr\left(J_{AA}S_{AA}\right) + \text{other terms that don't depend } J_{AA}$$

$$\nabla_{J_{AA}} \log|J| - Tr(JS) = 0 = (J_{AA} - J_{AB}J_{BB}^{-1}J_{BA})^{-T} - S_{AA}^{T}$$

$$(S_{AA})^{-1} = J_{AA} - J_{AB}J_{BB}^{-1}J_{BA}$$

$$\boxed{(S_{AA})^{-1} + J_{AB}J_{BB}^{-1}J_{BA} = J_{AA}}$$

This update keeps $J$ positive definite b/c $\forall x$, we have

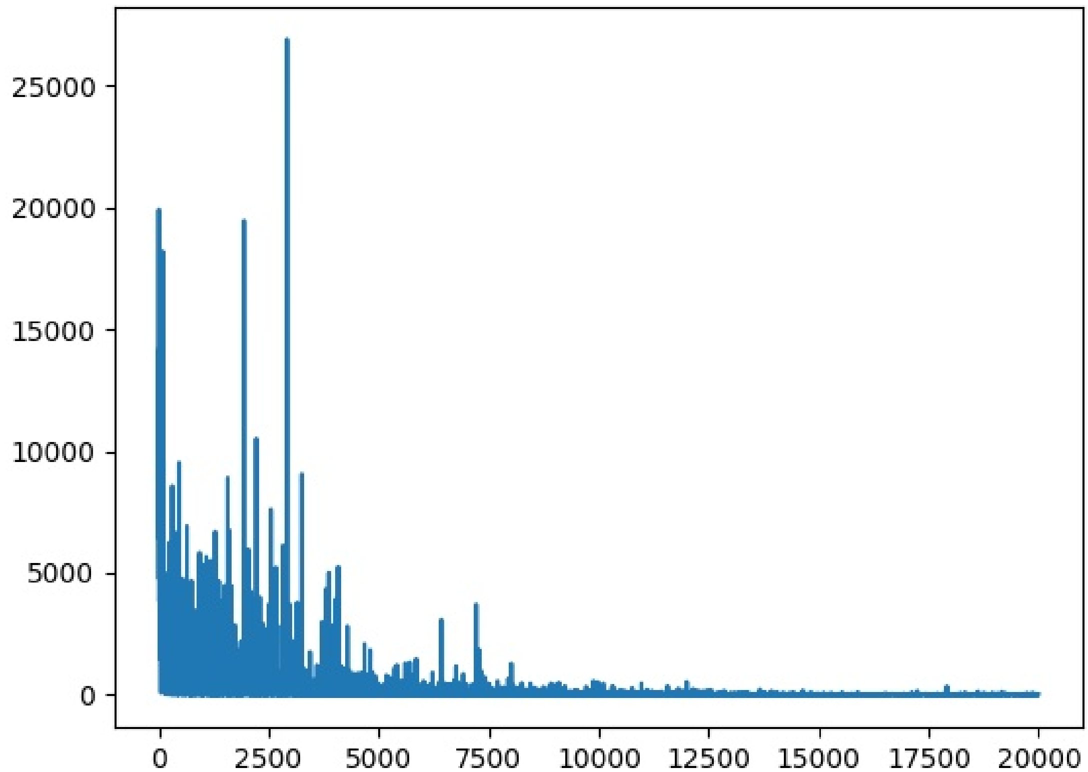
$$x^T J_{AA} x = \underbrace{x^T(S_{AA})^{-1}x}_{PD} + \underbrace{x^T J_{AB}}_{y^T}\underbrace{J_{BB}^{-1}}_{P.D}\underbrace{J_{BA}x}_{y = J_{BA}x} \geq 0$$

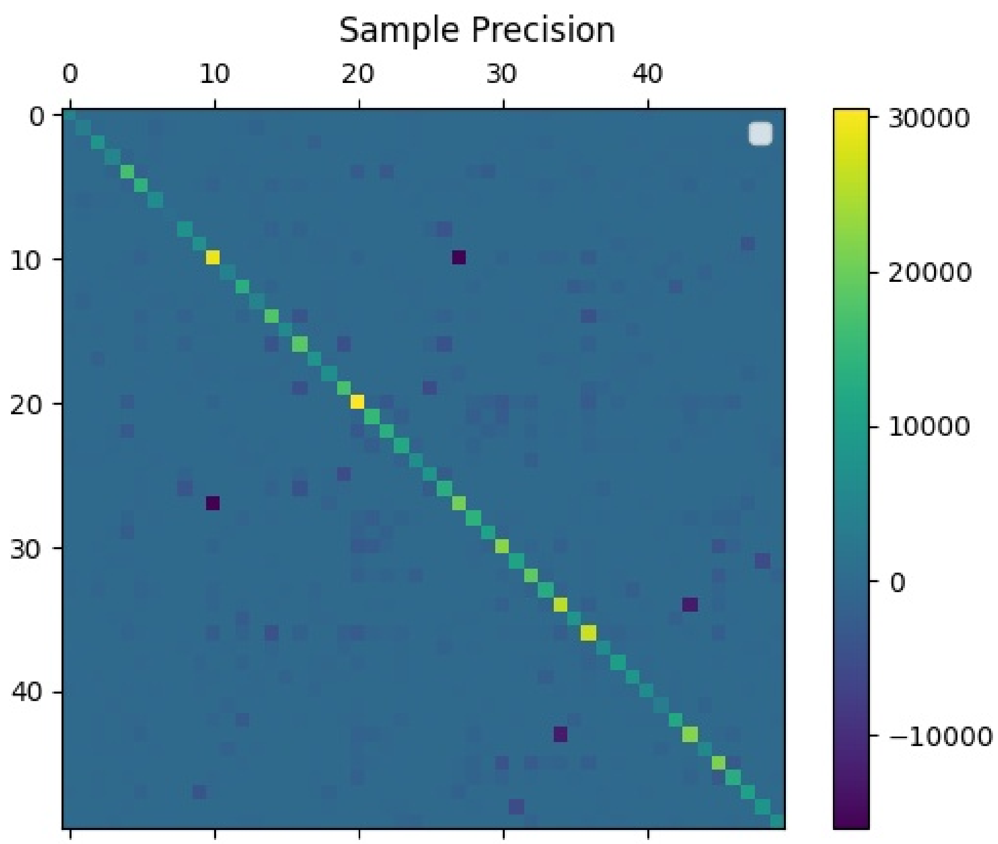
The sum of 2 PD matrices matrices is itself P.D.

The Schur complement of $J_{AA}$ is also P.D. by construction (initialization + update)

$J$ is P.D. b/c $J_{AA}$ is P.D. & $J \setminus J_{AA}$ is P.D.

Number of Non-Zero Edges: 1030
This number of non-zero edges is closest to $\alpha$ = 1e-05.






```python
J = np.eye(len(sample_cov))
n_iters = 20000
diffs = np.zeros(n_iters)
for n_iter in range(n_iters):

    assert np.all(np.diag(J) > 0)

    rand_indices = np.random.choice(
        np.arange(len(sample_cov)),
        size=2,
        replace=False)

    A_mask = np.full(len(sample_cov), fill_value=False)
    A_mask[rand_indices] = True
    B_mask = np.logical_not(A_mask)

    Saa_inv = np.linalg.inv(sample_cov[np.ix_(A_mask, A_mask)])
    Jaa = J[np.ix_(A_mask, A_mask)]
    Jab = J[np.ix_(A_mask, B_mask)]
    Jba = J[np.ix_(B_mask, A_mask)]
    Jbb = J[np.ix_(B_mask, B_mask)]
    L = Jab @ np.linalg.inv(Jbb) @ Jba
    Saa_inv_plus_L = Saa_inv + L
    if Saa_inv_plus_L[0, 1] <= 0:
```

```python
            block_replacement = Saa_inv_plus_L
        else:
            L11, L22 = L[0, 0], L[1, 1]
            Suu = sample_cov[np.ix_(A_mask, A_mask)][0, 0]
            Svv = sample_cov[np.ix_(A_mask, A_mask)][1, 1]
            weird_numerator = (1 + np.sqrt(1 + 4 * Suu * Svv * L[0, 1] * L[1, 0]))
            block_replacement = np.array([
                [L11 + (weird_numerator / (2 * Suu)), 0.],
                [0., L22 + (weird_numerator / (2 * Svv))]])

        diff = np.linalg.norm(block_replacement - Jaa)
        print(f'Iteration {n_iter}: {diff}')
        diffs[n_iter] = diff
        J[np.ix_(A_mask, A_mask)] = block_replacement

plt.plot(np.arange(n_iters), diffs)
plt.savefig('compcii_convergence.jpg')
plt.show()


plt.savefig('compcii_precision.jpg')
fig, ax = plt.subplots(nrows=1, ncols=1)
fig.suptitle('Sample Precision')
cax = ax.matshow(J)
fig.colorbar(cax)
ax.legend()
plt.savefig('compcii_precision.jpg')
plt.show()

print('Number of Non-Zero Edges: ', np.sum(J != 0.))
```