

6.438 Fall 2020 Problem Set 4

Rylan Schaeffer

Problem 1

1a

4.1 |

$$x \sim N^{-1}(0, \Sigma) \quad q_{ij} \sim N\left(0, \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix}\right) \quad q_i(x_i) \sim N(0, \Sigma_{ii})$$

(a) $p \sim N(0, \Sigma)$

$$\begin{aligned}
 H(p) &= - \int p(x) \log p(x) dx \\
 &= - \int \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2}x^\top \Sigma^{-1} x\right) \log \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2}x^\top \Sigma^{-1} x\right) dx \\
 &= - \int \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2}x^\top \Sigma^{-1} x\right) \left[-\frac{1}{2}x^\top \Sigma^{-1} x - \frac{1}{2} \log 2\pi |\Sigma| \right] dx \\
 &= \frac{1}{2} \log 2\pi |\Sigma| + \int \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2}x^\top \Sigma^{-1} x\right) \left(-\frac{1}{2}x^\top \Sigma^{-1} x \right) dx \\
 &= \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\Sigma| + \underbrace{\int p(x) \left(-\frac{1}{2}x^\top \Sigma^{-1} x \right) dx}_{+ -\frac{1}{2} \text{Tr}[\mathbb{E}[x^\top \Sigma^{-1} x]]} \\
 &= \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\Sigma| + \frac{1}{2} \mathbb{E}[x^\top \Sigma^{-1} x] \\
 &\quad + -\frac{1}{2} \text{Tr}[\mathbb{E}[x^\top \Sigma^{-1} x]] \\
 &\quad + -\frac{1}{2} \text{Tr}[\Sigma^{-1} \cancel{\mathbb{E}[xx^\top]}] \\
 &\quad + -\frac{n}{2} \\
 &= \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\Sigma| + \frac{n}{2}
 \end{aligned}$$

1b

4.11

b) $p(x) \sim N^{-1}(0, \Sigma)$ $q_{ij}(x_i, x_j) \sim N\left(0, \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix}\right)$ $q_i(x_i) \sim N(0, \Sigma_{ii})$

For convenience, define $C_{ij} = \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix}$

$$\Phi_{\text{Bethe}}(q) = \sum_{i \in V} \left(\mathbb{E}_{q_i} [\tilde{\phi}_i(x_i)] + H[q_i] \right) + \sum_{(i,j) \in E} \left(\mathbb{E}_{q_{ij}} [\tilde{\psi}_{ij}] - I(q_{ij}) \right)$$

Going 1 piece at a time:

$$H[q_i] = \frac{1}{2} \log 2\pi + \frac{1}{2} + \frac{1}{2} \log |\Sigma_{ii}| \quad (\text{from 4.1(a)})$$

$$I(q_{ij}) = \mathbb{E}_{q_{ij}} \left[\log \frac{q_{ij}(x_i, x_j)}{q_i(x_i) q_j(x_j)} \right]$$

$$= \mathbb{E}_{q_{ij}} \left[\log q_{ij}(x_i, x_j) \right] - \mathbb{E}_{q_{ij}} \left[\log q_i(x_i) \right] - \mathbb{E}_{q_{ij}} \left[\log q_j(x_j) \right]$$

Focusing on just $-\mathbb{E}_{q_{ij}} [\log q_i(x_i)]$:

$$= - \int_{x_i x_j} q_{ij}(x_i, x_j) \left[-\frac{1}{2} x_i^\top \Sigma_{ii}^{-1} x_i - \frac{1}{2} \log 2\pi |\Sigma_{ii}| \right]$$

$$= - \int_{x_i} q_{ij}(x_i) \left[-\frac{1}{2} x_i^\top \Sigma_{ii}^{-1} x_i - \frac{1}{2} \log 2\pi |\Sigma_{ii}| \right] \int_{x_j} \overbrace{q_{ij}(x_j | x_i)}^1 dx_j dx_i$$

$$= -H[q_i]$$

By symmetry, $-\mathbb{E}_{q_{ij}} [\log q_j(x_j)] = H[q_j]$

$$\text{Then } \mathbb{E}_{q_{ij}} [\log q_{ij}(x_i, x_j)] = -H[q_{ij}]$$

$$\Rightarrow I(q_{ij}) = H[q_i] + H[q_j] - H[q_{ij}] = \frac{1}{2} \log |\Sigma_{ii}| + \frac{1}{2} \log |\Sigma_{jj}| - \frac{1}{2} \log |C_{ij}|$$

$$= \frac{1}{2} \log 2\pi + \frac{1}{2} + \frac{1}{2} \log |\Sigma_{ii}| + \frac{1}{2} \log |\Sigma_{jj}| - \frac{1}{2} \log 2\pi - \frac{1}{2} - \frac{1}{2} \log |C_{ij}|$$

1b Continued

4.1]

b cont) We also need to figure out node & edge potentials

$$p(\underline{x}) = \frac{1}{Z(\underline{x}, \underline{\beta})} \exp \left(\sum_{ij} -\frac{1}{2} J_{ij} x_i x_j + \sum_i x_i b_i^0 \right)$$

$$\Rightarrow \tilde{\Phi}_{ij} = -J_{ij} x_i x_j \quad \text{for } i \neq j$$

$$\tilde{\Phi}_i = -\frac{1}{2} J_{ii} x_i^2$$

Plugging in entropy, mutual info, node & edge potentials:

$$\varphi_{\text{Bethe}}(\underline{q}) = \sum_i \left(\mathbb{E}_{q_i} [\tilde{\Phi}_i(x_i)] + H(q_i) \right) + \sum_{(i,j) \in \mathcal{E}} \left(\mathbb{E}_{q_{ij}} [\tilde{\Phi}_{ij}(x_i, x_j)] - I(q_{ij}) \right)$$

$$\boxed{\varphi_{\text{Bethe}}(\underline{q}) = \sum_i \left(-\frac{1}{2} J_{ii} \Sigma_{ii} + \frac{1}{2} \log 2\pi + \frac{1}{2} + \frac{1}{2} \log \Sigma_{ii} \right) + \sum_{(i,j) \in \mathcal{E}} \left(-J_{ij} \Sigma_{ij} + \frac{1}{2} \log \left| \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix} \right| - \frac{1}{2} \log \Sigma_{ii} - \frac{1}{2} \log \Sigma_{jj} \right)}$$

1c

4.11

$$c) \partial_{\Sigma_{ij}} \Psi_3(\hat{\boldsymbol{\theta}}) = 0 = -J_{ij} + \frac{1}{2} \partial_{\Sigma_{ij}} \log \left| \begin{bmatrix} \hat{\Sigma}_{ii} & \hat{\Sigma}_{ij} \\ \hat{\Sigma}_{ji} & \hat{\Sigma}_{jj} \end{bmatrix} \right|$$

$$\Rightarrow 0 = -J_{ij} + \frac{-\hat{\Sigma}_{ij}}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}^2}$$

$$\Rightarrow 0 = -J_{ij} \hat{\Sigma}_{ij} + J_{ij} \hat{\Sigma}_{ii} \hat{\Sigma}_{jj} + \Sigma_{ij}$$

$$\Rightarrow \hat{\Sigma}_{ij} = \frac{1 \pm \sqrt{J_{ij}^2 + 4 J_{ij}^2 \hat{\Sigma}_{ii} \hat{\Sigma}_{jj}}}{2 J_{ij}}$$

1d

4.1

$$d) \partial_{\hat{\Sigma}_{ii}} \varphi_{\text{Bethe}}(q) = -\frac{1}{2} J_{ii} + \frac{1}{2} \frac{1}{\hat{\Sigma}_{ii}} + \sum_{j \in N(i)} \left(\frac{1}{2} \partial_{\hat{\Sigma}_{ii}} \log \left| \begin{bmatrix} \hat{\Sigma}_{ii} & \hat{\Sigma}_{ij} \\ \hat{\Sigma}_{ji} & \hat{\Sigma}_{jj} \end{bmatrix} \right| - \frac{1}{2} \frac{1}{\hat{\Sigma}_{ii}} \right)$$

$$0 = -J_{ii} + \frac{1}{\hat{\Sigma}_{ii}} + \sum_{j \in N(i)} \left(\frac{\hat{\Sigma}_{jj}}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}} - \frac{1}{\hat{\Sigma}_{ii}} \right).$$

$$J_{ii} \hat{\Sigma}_{ii} = 1 + \sum_{j \in N(i)} \left(\frac{\hat{\Sigma}_{jj} \hat{\Sigma}_{ii}}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}} - 1 \right)$$

$$\hat{\Sigma}_{ii} = \frac{1}{J_{ii}} + \sum_{j \in N(i)} \frac{1}{J_{ii}} \left(\underbrace{\frac{\hat{\Sigma}_{jj} \hat{\Sigma}_{ii}}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}} - 1}_{\text{wanted: } = J_{ij} \hat{\Sigma}_{ij}} \right)$$

Note that from Part (c), we have

$$0 = -J_{ij} + \frac{\hat{\Sigma}_{ij}}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}^2}$$

$$= -J_{ij} \hat{\Sigma}_{ij} - \frac{\hat{\Sigma}_{ij}^2}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}^2}$$

$$\begin{aligned} J_{ij} \hat{\Sigma}_{ij} &= - \frac{\hat{\Sigma}_{ij}^2}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}^2} + \frac{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ii} \hat{\Sigma}_{jj}}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}^2} \\ &= -1 + \frac{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj}}{\hat{\Sigma}_{ii} \hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}^2} \end{aligned}$$

$$\Rightarrow \boxed{\hat{\Sigma}_{ii} = \frac{1}{J_{ii}} + \sum_{j \in N(i)} \frac{1}{J_{ii}} J_{ij} \hat{\Sigma}_{ij}}$$

1e

Alternate first computing the on-diagonal components $\hat{\Sigma}_{ii}$ using the off-diagonal components $\hat{\Sigma}_{ij}, i \neq j$, then computing the off-diagonal components using the on-diagonal components.

Problem 2

2b(i)

4.2 |

b) (i) $p(\underline{x}, \underline{y}; \theta) = p(x_0) \prod_{i=1}^N p(x_i | x_{i-1}) \prod_{i=0}^N p(y_i | x_i)$

$$\begin{aligned} \log p(\underline{x}, \underline{y}; \theta) &= \log p(x_0) + \sum_{i=1}^N \log p(x_i | x_{i-1}) + \sum_{i=0}^N \log p(y_i | x_i) \\ &= -\frac{1}{2} \log 2\pi - \frac{1}{2} \frac{x_0^2}{\sigma^2} + \sum_{i=1}^N \left(-\frac{1}{2} \log(2\pi) - \frac{1}{2} (x_i - \alpha x_{i-1})^2 \right) \\ &\quad + \sum_{i=0}^N \left(-\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2} \frac{(y_i - \beta x_i)^2}{\sigma^2} \right) \end{aligned}$$

$$\Rightarrow E[\ell] = \left[-\frac{1}{2} \sum_{i=0}^N x_{i-1}^2 \right] + \frac{-2N-2}{2} \log 2\pi - [N+1] \log \sigma + \left[\sum_{i=1}^N x_i x_{i-1} \right] \alpha +$$

$$\left[-\frac{1}{2} \sum_{i=1}^N x_{i-1}^2 \right] \alpha^2 + \left[-\sum_{i=0}^N y_i^2 + \beta 2 \sum_{i=0}^N y_i x_i - \beta^2 \sum_{i=1}^N x_i^2 \right] \frac{1}{2\sigma^2}$$

$$\Rightarrow E[\log p_{x,y}(\underline{x}, \underline{y}; \theta)] = \left[-\frac{1}{2} \sum_{i=0}^N \langle x_i^2 \rangle + \frac{-2N-2}{2} \log 2\pi \right] - [N+1] \log \sigma + \alpha \left[\sum_{i=1}^N \langle x_i x_{i-1} \rangle \right]$$

$$+ \alpha^2 \left[-\frac{1}{2} \sum_{i=1}^N \langle x_{i-1}^2 \rangle \right] + \left[-\sum_{i=0}^N y_i^2 + 2\beta \sum_{i=0}^N y_i \langle x_i \rangle - \beta^2 \sum_{i=1}^N \langle x_i^2 \rangle \right] \frac{1}{2\sigma^2}$$

2b(ii)

I accidentally dropped a sign for $\hat{\sigma}^2$; the numerator should be multiplied by -1. Too lazy to rewrite. I also thought that F and G were defined including β , but then realized they weren't, so that's why my F and G change mid-problem.

4.2

$$(b)(ii) \quad \partial_{\alpha} \mathbb{E}[\log p(x, y; \theta)] = 0 = 2 \hat{\alpha} \left[-\frac{1}{2} \sum_{i=1}^N \langle x_{i-1}^2 \rangle + \left[\sum_{i=1}^N \langle x_i x_{i-1} \rangle \right] \right]$$

$$\Rightarrow \hat{\alpha} = \frac{\sum_{i=1}^N \langle x_i x_{i-1} \rangle}{\sum_{i=1}^N \langle x_{i-1}^2 \rangle}$$

$$\partial_{\beta} \mathbb{E}[\log p(x, y; \theta)] = 0 = \frac{1}{2\hat{\sigma}^2} \left[2 \sum_{i=0}^N y_i \langle x_i \rangle \right] = \frac{2\hat{\beta}}{2\hat{\sigma}^2} \left[\sum_{i=0}^N \langle x_i^2 \rangle \right]$$

$$\Rightarrow \hat{\beta} = \frac{\sum_{i=0}^N y_i \langle x_i \rangle}{\sum_{i=0}^N \langle x_i^2 \rangle}$$

$$\text{Define } [\log s = \sigma^2] \text{ Then } = -\frac{(N+1)}{s} + \frac{-2}{2}$$

$$\partial_s \mathbb{E}[\log p(x, y; \theta)] = 0 = \frac{(N+1)}{2} \log s + \frac{1}{2} s [\mathbb{E}[E + F\beta + G\beta^2]]$$

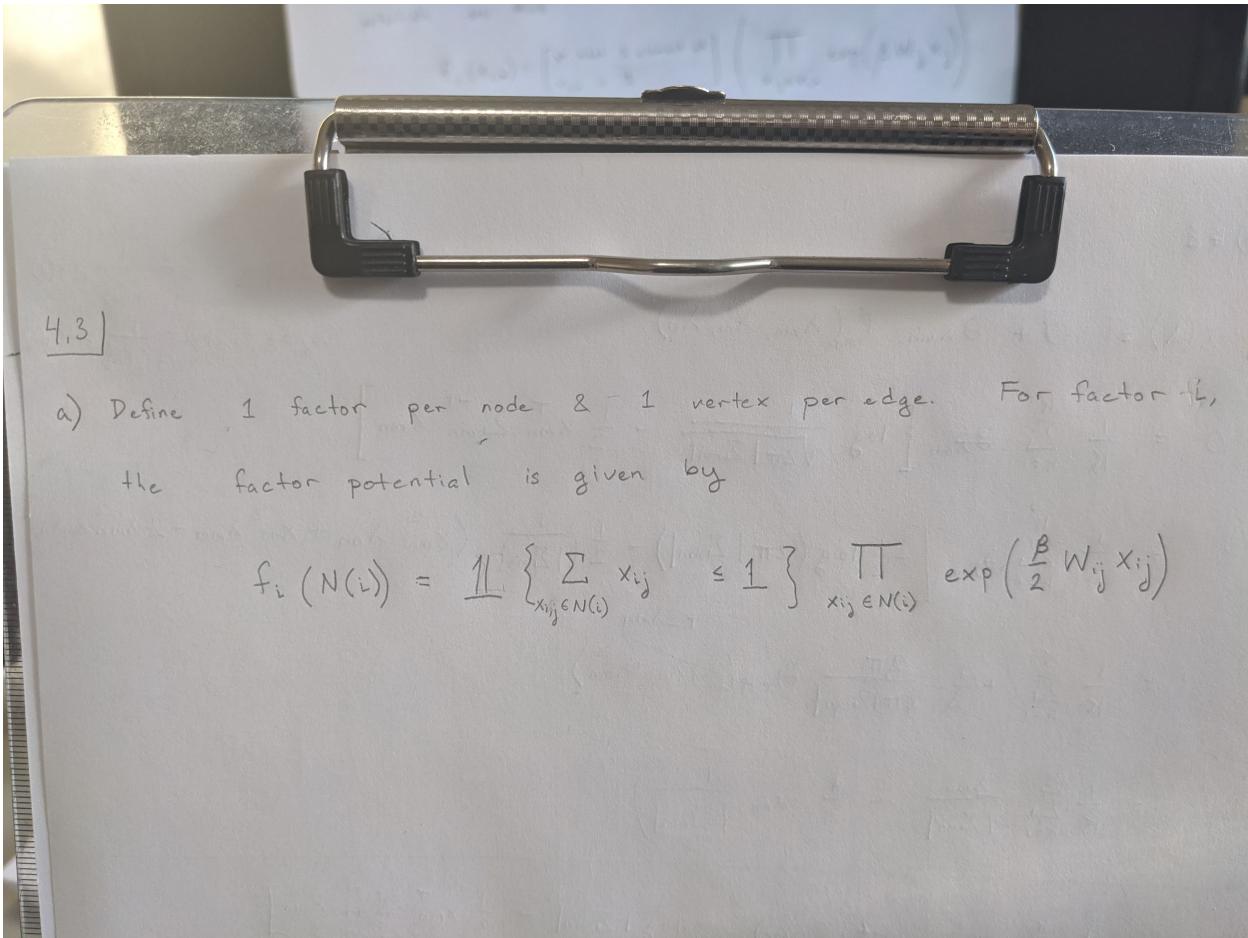
$$0 = \frac{(N+1)}{s} + \frac{1}{2} \mathbb{E}[E + F + G]$$

$$s = \frac{2(N+1)}{\mathbb{E}[E + F + G]} \Rightarrow \hat{\sigma}^2 = \frac{\mathbb{E}[E + F\beta + G\beta^2]}{-(N+1)}$$

$$\hat{\sigma}^2 = \frac{1}{N+1} \left[-\sum_{i=0}^N y_i^2 + 2\hat{\beta} \sum_{i=0}^N y_i \langle x_i \rangle - \hat{\beta}^2 \sum_{i=0}^N \langle x_i^2 \rangle \right]$$

Problem 3

3a



3b

4.3

b) Given: Initialize $m_{L \rightarrow a}^0(x_i) = 1$ $m_{a \rightarrow L}^0(x_i) = 1$

$$\text{Update } m_{L \rightarrow a}^{t+1}(x_L) = \prod_{b \in N(L) \setminus \{L\}} m_{b \rightarrow i}^t(x_i)$$

$a = \text{factor}$

$i = \text{node}$

$$m_{a \rightarrow L}^{t+1}(x_L) = \sum_{x_{N(a)} \setminus \{L\}} f_a(x_{N(a)}) \prod_{j \in N(a) \setminus \{L\}} m_{j \rightarrow a}^t(x_i)$$

node to factor

factor to node

Goal: Messages $m_{L \rightarrow j}^t(x_{ij})$ for each edge $(i, j) \in E$ in original graph

Approach: Consider factor $a \rightarrow$ node $(a, b) \rightarrow$ factor b e.g. 

$$m_{(a,b) \rightarrow b}^{t+1}(x_{ab}) = m_{a \rightarrow (a,b)}^t \quad \text{i.e., nodes forward}$$

X

messages from factors

$$m_{a \rightarrow (a,b)}^{t+1}(x_{ab}) = \sum_{x_{N(a)} \setminus \{(a,b)\}} f_a(x_{N(a)}) \prod_{(c,a) \in N(a) \setminus \{(a,b)\}} m_{(c,a) \rightarrow a}^t(x_{ca})$$

$$\begin{aligned} \text{if } x_{ab} = 1, \text{ then } f_a(x_{N(a)}) &= \exp\left(\frac{\beta}{2} W_{ab}\right) \text{ if all other } x_{ca} = 0 \\ &= \text{incoming msgs are for } x_{ca} = 0; \text{ otherwise} \\ &\text{at least two incoming edges} = 1 \Rightarrow \text{factor is 0} \end{aligned}$$

if $x_{ab} = 0$, then we split the sum into "active" and "non-active" incoming edges. If another factor is inactive, its factor potential is 1, so it forwards

$$\Rightarrow m_{a \rightarrow b}^{t+1} = m_{a \rightarrow b}^t \prod_{(c,a) \in N(a) \setminus \{(a,b)\}} m_{c \rightarrow a}^t \quad \text{If another factor is active, its factor potential times the other messages is } \exp\left(\frac{\beta}{2} W_{ca}\right) m_{c \rightarrow a}^t \prod_{d \in N(a) \setminus \{(a,b,c)\}} m_{d \rightarrow a}^t$$

since other edges must also be 0

$$\Rightarrow m_{a \rightarrow b}^{t+1}(x_{ab}) = \begin{cases} \exp\left(\frac{\beta}{2} W_{ab}\right) \prod_{(c,a) \in N(a) \setminus \{(a,b)\}} m_{c \rightarrow a}^t & \text{if } x_{ab} = 1 \\ \prod_{c \in N(a) \setminus b} m_{c \rightarrow a}^t + \sum_{c \in N(a) \setminus b} \exp\left(\frac{\beta}{2} W_{ca}\right) m_{c \rightarrow a}^t \prod_{d \in N(a) \setminus \{(a,b,c)\}} m_{d \rightarrow a}^t & \text{if } x_{ab} = 0 \end{cases}$$

3c & 3d

4.3

$$\begin{aligned}
 c) \quad \tilde{m}_{a \rightarrow b}^t &\stackrel{\text{def}}{=} \frac{1}{\beta} \log \left(\frac{m_{a \rightarrow b}^t(v)}{m_{a \rightarrow b}^t(0)} \right) \\
 &= \frac{1}{\beta} \log \left(\frac{\exp \left(\frac{\beta W_{ab}}{2} \right) \prod_{c \in N(a) \setminus \{b\}} m_{c \rightarrow a}^{t-1}(0)}{\prod_{c \in N(a) \setminus \{b\}} m_{c \rightarrow a}^{t-1}(0) + \sum_{c \in N(a) \setminus \{b\}} \exp \left(\frac{\beta W_{ca}}{2} \right) m_{c \rightarrow a}^{t-1}(1) \prod_{d \in N(c) \setminus \{b, c\}} m_{d \rightarrow c}^{t-1}(0)} \right) \\
 &= \frac{1}{\beta} \log \left(\frac{\exp \left(\frac{\beta W_{ab}}{2} \right)}{1 + \sum_{c \in N(a) \setminus \{b\}} \underbrace{\exp \left(\frac{\beta W_{ca}}{2} \right) m_{c \rightarrow a}^{t-1}(1) / m_{c \rightarrow a}^{t-1}(0)}_{\text{underbrace}}} \right) \\
 &= \exp \left(\beta \tilde{m}_{c \rightarrow a} \right)
 \end{aligned}$$

$$d) \quad \text{Using the fact that } \lim_{\beta \rightarrow \infty} \frac{1}{\beta} \log \left(\sum_i \exp(\beta a_i) \right) = \max_i a_i$$

$$\begin{aligned}
 \tilde{m}_{a \rightarrow b}^t &= \frac{1}{\beta} \log \left(\exp \left(\frac{\beta W_{ab}}{2} \right) \right) - \frac{1}{\beta} \log \left(1 + \sum_{c \in N(a) \setminus \{b\}} \underbrace{\exp \left(\frac{\beta W_{ca} + \beta \tilde{m}_{c \rightarrow a}}{2} \right)}_{\text{underbrace}} \right) \\
 \lim_{\beta \rightarrow \infty} \tilde{m}_{a \rightarrow b}^t &= \frac{W_{ab}}{2} - \max \left\{ 0, \max_{c \in N(a) \setminus \{b\}} \left\{ \tilde{m}_{c \rightarrow a} + \frac{W_{ca}}{2} \right\} \right\} \\
 &\text{b/c } \log(1 + \sum_i \exp) \geq 0 \quad \text{but } \max_{c \in N(a) \setminus \{b\}} \left\{ \tilde{m}_{c \rightarrow a} + \frac{W_{ca}}{2} \right\} \text{ can be negative}
 \end{aligned}$$

To compute the MPC, I'd iterate the $\tilde{m}_{a \rightarrow b}^t$ messages

in the above form (i.e. considering limit as $\beta \rightarrow \infty$)

Problem 4

4a & 4b First Half

a) In HW2 Prob 2.2, we showed mult covariance = sample covariance for maximal cliques. Since each pair of adjacent nodes in a tree is a maximal clique

$$\Rightarrow \hat{\lambda}_{ij} = \frac{1}{K} \sum_{k=1}^K (x_i^{(k)})^2$$

$$\hat{\lambda}_{ij} = \frac{1}{K} \sum_{k=1}^K x_i^{(k)} x_j^{(k)}$$

b)

$$\begin{aligned} \varrho_{ij}(\hat{\lambda}_{ij}, \hat{\lambda}_{ij}, \hat{\lambda}_{ij}) &\stackrel{\text{def}}{=} \frac{1}{K} \sum_{k=1}^K \log \frac{P(x_i^{(k)}, x_j^{(k)}; \hat{\lambda}_{ij}, \hat{\lambda}_{ij})}{P(x_i^{(k)}, \hat{\lambda}_{ij}; P(x_j^{(k)}; \hat{\lambda}_{ij}))} \\ &= \frac{1}{K} \sum_{k=1}^K -\frac{1}{2} \log ((2\pi)^2 |\hat{\lambda}_{ij}|) - \frac{1}{2} x_{ij}^{(k)} \sum_{j,j}^{-1} x_{ij}^{(k)} + \frac{1}{2} \log (2\pi \hat{\lambda}_{ij}) + \frac{1}{2} x_{ij}^{(k)} \hat{\lambda}_{ij}^{-1} x_{ij}^{(k)} \\ &\quad + \frac{1}{2} \log (2\pi \hat{\lambda}_{ij}) + \frac{1}{2} x_{ij}^{(k)} \hat{\lambda}_{ij}^{-1} x_{ij}^{(k)} \\ &= \frac{1}{K} \sum_{k=1}^K -\frac{1}{2} \log ((2\pi)^2 (\hat{\lambda}_{ij} \hat{\lambda}_{ij} - \hat{\lambda}_{ij}^{(k)} \hat{\lambda}_{ij}^{(k)})) - \frac{1}{2} \frac{1}{\hat{\lambda}_{ij} \hat{\lambda}_{ij} - \hat{\lambda}_{ij}^{(k)} \hat{\lambda}_{ij}^{(k)}} x_{ij}^{(k)} \begin{bmatrix} \hat{\lambda}_{ij} & -x_{ij}^{(k)} \\ -x_{ij}^{(k)} & \hat{\lambda}_{ij} \end{bmatrix} \times_{ij}^{(k)} \\ &\quad + \frac{1}{2} \log (2\pi \hat{\lambda}_{ij}) + \frac{1}{2} x_{ij}^{(k)} \hat{\lambda}_{ij}^{-1} x_{ij}^{(k)} + \frac{1}{2} \log (2\pi \hat{\lambda}_{ij}) + \frac{1}{2} x_{ij}^{(k)} \hat{\lambda}_{ij}^{-1} x_{ij}^{(k)} \end{aligned}$$

Group log terms:

$$\begin{aligned} &= \frac{1}{K} \sum_{k=1}^K -\frac{1}{2} \log \left(\frac{(2\pi)^2 (\hat{\lambda}_{ij} \hat{\lambda}_{ij} - \hat{\lambda}_{ij}^{(k)} \hat{\lambda}_{ij}^{(k)})}{2\pi \hat{\lambda}_{ij} \cdot 2\pi \hat{\lambda}_{ij}} \right) - \frac{1}{2} \frac{x_{ij}^{(k)} \hat{\lambda}_{ij} + x_{ij}^{(k)} \hat{\lambda}_{ij} - 2\hat{\lambda}_{ij} x_{ij} x_{ij}^{(k)}}{\hat{\lambda}_{ij} \hat{\lambda}_{ij} - \hat{\lambda}_{ij}^{(k)} \hat{\lambda}_{ij}^{(k)}} \\ &\quad + \frac{1}{2} \frac{x_{ij}^{(k)} \hat{\lambda}_{ij}^2}{\hat{\lambda}_{ij}} + \frac{1}{2} \frac{x_{ij}^{(k)} \hat{\lambda}_{ij}^2}{\hat{\lambda}_{ij}} \end{aligned}$$

4b Second Half & 4c

4.41

b cont)

$$\begin{aligned}
 \ell_{ij}(\hat{\lambda}_{ij}, \hat{\lambda}_i, \hat{\lambda}_j) &= \frac{1}{2} \log \frac{\hat{\lambda}_i \hat{\lambda}_j}{\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}} \\
 &= -\frac{1}{2} \left[\frac{\sum_{k=1}^K x_i^{(k)} \sum_{k=1}^K x_j^{(k)} + x_{ij}^{(k)} \frac{1}{K} \sum_{k=1}^K x_i^{(k)^2} - \sum_{k=1}^K x_{ij}^{(k)^2}}{\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}} \right] \\
 &\quad + \frac{1}{2} \left[\frac{\sum_{k=1}^K x_i^{(k)} \sum_{k=1}^K x_j^{(k)} - x_{ij}^{(k)} \frac{1}{K} \sum_{k=1}^K x_i^{(k)^2}}{\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}} \right] \\
 &= \frac{1}{2} \log \frac{\hat{\lambda}_i \hat{\lambda}_j}{\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}^2} - \frac{1}{2} \frac{2 (\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}^2)}{(\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}^2)} + 1
 \end{aligned}$$

$$\hat{\ell}_{ij}(\hat{\lambda}_{ij}, \hat{\lambda}_i, \hat{\lambda}_j) = \frac{1}{2} \log \frac{\hat{\lambda}_i \hat{\lambda}_j}{\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}^2}$$

c) Define

$$f = \hat{\ell}_{ij}(\hat{\lambda}_{ij}, \hat{\lambda}_i, \hat{\lambda}_j).$$

This says to add more weight to edges proportional to their mutual information from adding the edge.

Problem 6

6a & 6b & 6c

4.6

(a) Let edge $i \rightarrow j$ and $U \subseteq V \setminus \{i, j\}$.
 Recall that $x_i \perp\!\!\!\perp x_j \mid x_U$ if all paths connecting x_i and x_j are blocked.
 B/c edge $(i, j) \in \Sigma$, the path $x_i \rightarrow x_j$ is not blocked $\Rightarrow x_i \not\perp\!\!\!\perp x_j \mid x_U$.

(b) Let x_i, x_j be non-adjacent. WLOG, let i come before j in the topological ordering. Define $U = x_{\pi_j}$. Note that
 $|U| \leq d_{in}$ by construction.

Claim: $x_i \perp\!\!\!\perp x_j \mid x_{\pi_j}$. All paths

Proof: All edges $x_k \rightarrow x_j$ are blocked. There can be no immoralities at these $x_k \in x_{\pi_j}$ b/c edge $x_k \rightarrow x_j$ is outgoing.
 Only way to unblock is to condition on immorality or child of immorality, but this creates a cycle.

This recovers the true skeleton b/c any edge $x_k \rightarrow x_j$ s.t. $x_k \in x_{\pi_j}$ is preserved, whereas all other edges are excluded.

(c) No. Consider

Here, $d_{in} = 3 > d_{out} = 2$

However, to d-separate d and e , we need to condition on $\{a, b, c\}$ i.e. $d \perp\!\!\!\perp e \mid \{a, b, c\}$. Conditioning on any two parents (e.g. $\{a, b\}$) will fail to block the path to the third (e.g. $\{c\}$). So then d_{out} cannot bound d .

6d

4.6 |

(d) Suppose $x_i \rightarrow x_k \leftarrow x_j$. Recall from (b) that $U_{ij} = x_{\pi j}$.

Since x_k is not a parent of x_j , $x_k \notin U_{ij}$

Now, WLOG, let $i < j$ in topological ordering. Suppose

$x_k \notin U_{ij}$. Then x_k is not a parent of x_j , and we have
 $x_k \leftarrow x_j$. If $x_l \leftarrow x_k$, then $j < l$ in the topological
ordering, a contradiction. So $x_l \rightarrow x_k \rightarrow x_j$.

Computational

Comp a(i) & Comp a(ii)

Comp

(a) (i)

$$z_i \stackrel{\text{def}}{=} \sum_{t=m(i-1)}^{m_i-1} y_t$$

$$= \sum C x_t x_t w_t \quad (\text{defn of } y_t)$$

$$= C \sum x_t + \underbrace{\sum w_t}_{\text{sum of } m \text{ random } N(0, R) \text{ variables}}$$

$$= C \tilde{s}_i + N(0, mR) \quad \text{by linearity of variance}$$

$$= C \tilde{s}_i + r \quad \text{where } r \sim N(0, mR)$$

(a) (ii) Least squares estimates

Suppose we have I total (\tilde{s}_i, z_i) pairs. Then

$$\hat{C} = \frac{\sum_{i=1}^I z_i \tilde{s}_i}{\sum_{i=1}^I \tilde{s}_i}$$

and $\hat{R} = \frac{1}{m} \sum_{i=1}^I (z_i - C \tilde{s}_i)^2$

I recover $\hat{C} = 0.10100075$ and $\hat{R} = 0.5476069656312432$ using the following code:

```
import numpy as np

m = 10
with open('data.txt', 'r') as fp:
    y = fp.readline()
    y = np.array([float(substr) for substr in y.split()])
    s = fp.readline()
    s = np.array([float(substr) for substr in s.split()])

s_tilde = np.diff(s, prepend=0.)
z = np.reshape(y, newshape=(-1, m))
z = np.sum(z, axis=1)

def ols_estimator(X, Y):
    """
    X has dimensions (num observations, num input dimensions)
    Y has dimensions (num observations, num output dimensions)

    params has dimensions (num input dimensions, num output dimensions)
    """
    params = np.linalg.inv(X.T @ X) @ X.T @ Y
    errors = Y - X @ params

    return params, errors

# a(ii)
Chat, errors = ols_estimator(
    X=np.expand_dims(s_tilde, -1),
    Y=np.expand_dims(z, -1))

Rhat = np.var(errors) / m

print('Chat:', Chat)
print('Rhat:', Rhat)
```

Comp b(i)

Comp

(b) (i) Starting w/

$$\begin{aligned} p(x_0 \dots x_{T-1} | y_0 \dots y_{T-1}) &\propto p(x_0) \prod_{i=0}^{T-2} p(x_{i+1} | x_i) \prod_{i=0}^{T-1} p(y_i | x_i) \\ &\propto \exp\left(-\frac{1}{2} \frac{(x_0 - \mu_0)^2}{\lambda_0} + \mu_0 x_0\right) \prod_{i=0}^{T-2} \exp\left(-\frac{1}{2} \frac{(x_{i+1} - Ax_i)^2}{Q} + Ax_i x_{i+1}\right) \\ &\quad \prod_{i=0}^{T-1} \exp\left(-\frac{1}{2} \frac{(y_i - Cx_i)^2}{R_i} + Cx_i y_i\right) \end{aligned}$$

Rearranging & grouping

$$x_0^2 \text{ terms: } -\frac{1}{2} \frac{1}{\lambda_0} x_0^2 - \frac{1}{2} \frac{A^2}{Q} x_0^2 - \frac{1}{2} \frac{C^2}{R_0} x_0^2 \Rightarrow J_{00} = \underline{\lambda_0^{-1} + A^T Q^{-1} A + C^T R_0^{-1} C}$$

$$x_0 \text{ terms: } \frac{\mu_0}{\lambda_0} x_0 + Ax_1 x_0 + \frac{C}{R_0} x_0 y_0 \Rightarrow h_0 = \underline{\lambda_0^{-1} \mu_0 + C^T R_0^{-1} y_0}$$

$$x_{i \neq T-1}^2 \text{ terms: } \frac{1}{Q} + \frac{A^2}{Q} + \frac{C^2}{R_i} \Rightarrow J_{ii} = \underline{Q^{-1} + A^T Q^{-1} A + C^T R_i^{-1} C}$$

for $i \neq T-1$
 $i \neq 0$

$$x_{T-1} \text{ terms: } \frac{C}{R_{T-1}} y_{T-1} \Rightarrow h_{T-1} = \underline{C^T R_{T-1}^{-1} y_{T-1}}$$

$$x_{T-1}^2 \text{ terms} \quad \frac{1}{Q} + \frac{C^2}{R_{T-1}} \Rightarrow J_{TT} = \underline{Q^{-1} + C^T R_{T-1}^{-1} C}$$

$$x_i \text{ terms} \quad \frac{C}{R_i} y_i \Rightarrow h_i = \underline{C^T R_i^{-1} y_i}$$

Comp b(ii)

```

# b(ii)
from collections import defaultdict

def compute_potentials(y, A, C, Q, Rs, mu_0, Lambda_0):
    # compute potentials
    hs = dict()
    Js = defaultdict(dict)
    for i in range(len(y)):
        J_ii = C.T @ inv(Rs[i, :, :]) @ C
        if i == 0:
            J_ii += inv(Lambda_0)
        if i != 0:
            J_ii += inv(Q)
        if i != (len(y) - 1):
            J_ii += A.T @ inv(Q) @ A
        Js[i][i] = np.copy(J_ii)

        h_i = C.T @ inv(Rs[i]) @ y[i]
        if i == 0:
            h_i += inv(Lambda_0) @ mu_0
        hs[i] = np.copy(h_i)

        J_ij = -A.T @ inv(Q)
        if i == (len(y) - 1):
            J_ij *= 1
        Js[i][i + 1] = np.copy(J_ij)

    return hs, Js

def forward_pass(hs, Js):
    hs_forward = defaultdict(dict)
    Js_forward = defaultdict(dict)
    for i in range(len(hs) - 1):
        if i == 0:
            hs_forward[i][i + 1] = -Js[i][i + 1].T @ inv(Js[i][i]) @ hs[i]
            Js_forward[i][i + 1] = -Js[i][i + 1].T @ inv(Js[i][i]) @ Js[i][i + 1]
        else:
            hs_forward[i][i + 1] = -Js[i][i + 1].T @ inv(Js[i][i] + Js_forward[i - 1][i])
            hs[i] + hs_forward[i - 1][i]
            Js_forward[i][i + 1] = -Js[i][i + 1].T @ inv(Js[i][i] + Js_forward[i - 1][i])

    return hs_forward, Js_forward

def backward_pass(hs, Js):
    hs_backward = defaultdict(dict)
    Js_backward = defaultdict(dict)
    for i in range(len(hs) - 2, -1, -1):
        if i == (len(hs) - 2):

```

```

hs_backward[i + 1][i] = - Js[i][i + 1] @ inv(Js[i + 1][i + 1]) @ hs[i + 1]
Js_backward[i + 1][i] = - Js[i][i + 1] @ inv(Js[i + 1][i + 1]) @ Js[i][i + 1].T
else:
    hs_backward[i + 1][i] = - Js[i][i + 1] @ inv(Js[i + 1][i + 1]) + Js_backward[i - 1][i]
    hs[i + 1] + hs_backward[i + 2][i + 1])
    Js_backward[i + 1][i] = - Js[i][i + 1] @ inv(Js[i + 1][i + 1]) + Js_backward[i - 1][i]

return hs_backward, Js_backward

def compute_posteriors(hs, hs_forward, hs_backward, Js, Js_forward, Js_backward):

    hs_hat, Js_hat = dict(), dict()
    for i in range(len(hs)):
        if i == 0:
            hs_hat[i] = hs[i] + hs_backward[i+1][i]
            Js_hat[i] = Js[i][i][0, 0] + Js_backward[i+1][i]
        elif i == (len(hs) - 1):
            hs_hat[i] = hs[i] + hs_forward[i-1][i]
            Js_hat[i] = Js[i][i] + Js_forward[i-1][i]
        else:
            hs_hat[i] = hs[i] + hs_forward[i-1][i] + hs_backward[i+1][i]
            Js_hat[i] = Js[i][i] + Js_forward[i-1][i] + Js_backward[i + 1][i]

    return hs_hat, Js_hat

def gaussian_belief_prop(y, A, C, Q, Rs, mu_0, Lambda_0):
    """
    :param n: dimension of
    :param A:
    :param C:
    :param Q:
    :param R:
    :return:
    """

# correct
hs, Js = compute_potentials(y=y, A=A, C=C, Q=Q, Rs=Rs, mu_0=mu_0, Lambda_0=Lambda_0)

hs_forward, Js_forward = forward_pass(hs=hs, Js=Js)
hs_backward, Js_backward = backward_pass(hs=hs, Js=Js)

hs_hat, Js_hat = compute_posteriors(
    hs=hs, hs_forward=hs_forward, hs_backward=hs_backward,
    Js=Js, Js_forward=Js_forward, Js_backward=Js_backward)

times = np.arange(len(y))
vars = np.array([inv(Js_hat[time])[0, 0] for time in times])
means = np.array([(inv(Js_hat[time]) @ hs_hat[time])[0] for time in times])

p = plt.plot(times, means)
plt.fill_between(times,

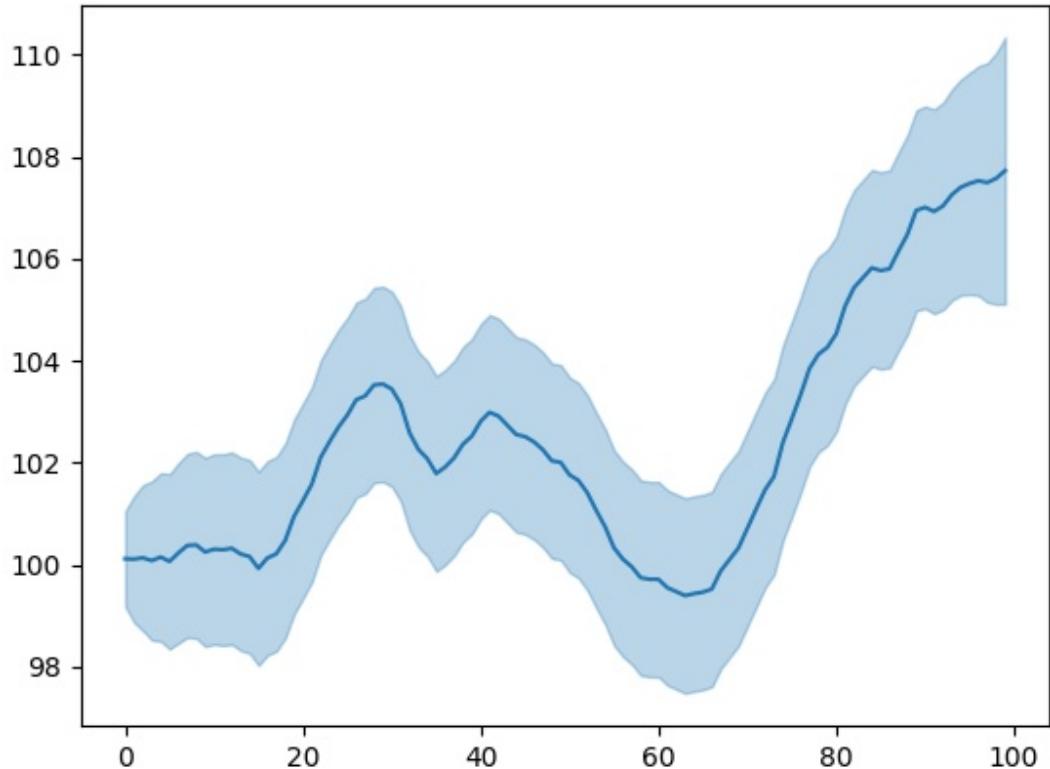
```

```

means - np.sqrt(vars),
means + np.sqrt(vars),
color=p[-1].get_color(),
alpha=0.3)

```

Comp b(iii)



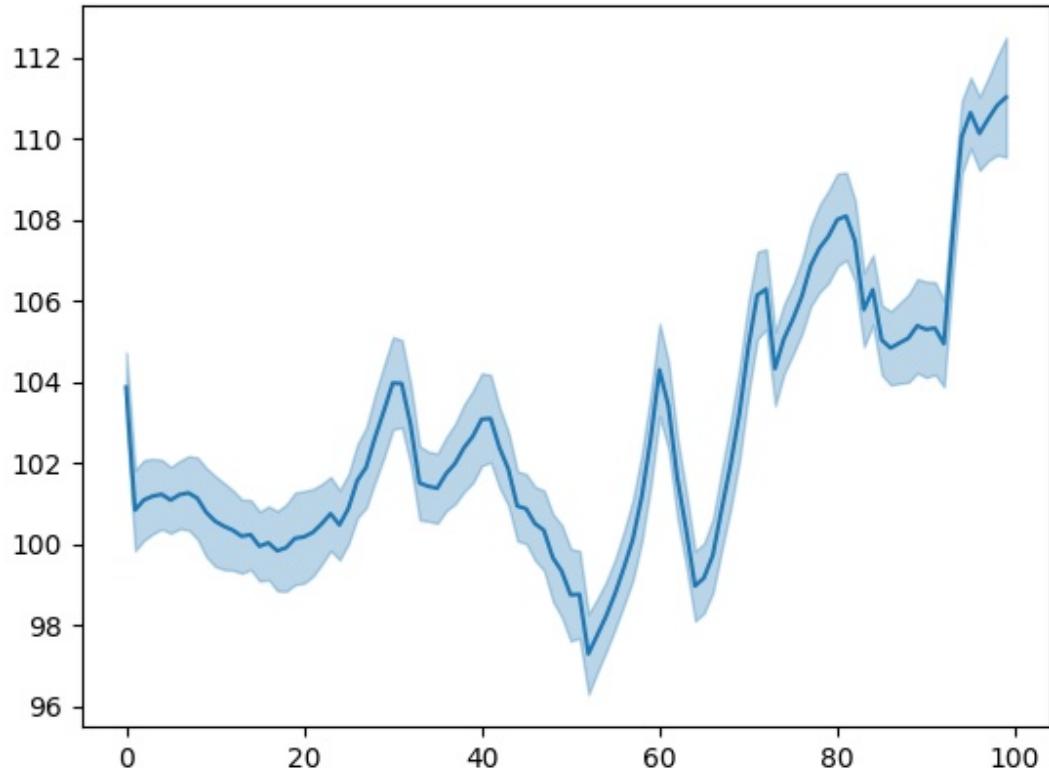
```

# b( iii )
A = np.array([[0.9999]])
C = np.copy(Chat)
Q = np.array([[1.]])
Rs = np.full(shape=(len(y), 1, 1), fill_value=Rhat)
mu_0 = np.array([100.])
Lambda_0 = np.array([[1.]])
gaussian_belief_prop(y=np.expand_dims(y, axis=-1),
                      A=A, C=C, Q=Q, Rs=Rs, mu_0=mu_0, Lambda_0=Lambda_0)
plt.savefig('Comp_b( iii ).jpg')
plt.show()

```

Comp b(iv)

I used $\epsilon = 1e^{-6}$ and $\sigma = 1e^6$.



```

# b(iv)
A = np.array([[0.9999, 0.], [0.9999, 1.]])
C = np.array([[Chat[0, 0], 0.], [0., 1.]])
epsilon = 1e-6
sigma = 1e6
Q = np.array([[1., 1.], [1., 1. + epsilon]])
Rs = np.stack([np.array([[Rhat, 0.], [0., 0.]]) for _ in range(len(y))])
observed_indices = np.array([True if (i + 1) % m == 0 else False for i in range(len(y))])
Rs[observed_indices, 1, 1] = epsilon
Rs[np.logical_not(observed_indices), 1, 1] = sigma

s_rep = np.array([s[i//m] for i in range(len(y))])
y_aug = np.stack([np.squeeze(y), s_rep]).transpose((1, 0))
mu_0 = np.array([100., 100.])
Lambda_0 = np.array([[1., 0.], [0., 1.]])
gaussian_belief_prop(y=y_aug, A=A, C=C, Q=Q, Rs=Rs, mu_0=mu_0, Lambda_0=Lambda_0)
plt.savefig('Comp_b(iv).jpg')
plt.show()

```