6.438 Fall 2020 Problem Set 4

Rylan Schaeffer

Problem 1

1a

$\boxed{4.1}$

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

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

$$H(p) = - \int p(x) \log p(x) \, dx$$

$$= - \int \frac{1}{\sqrt{2\pi |\Sigma|}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) \log \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) dx$$

$$= \cancel{-\int} \frac{1}{\sqrt{2\pi|\Sigma|}} \exp(\quad) \left[-\frac{1}{2} x^T \Sigma^{-1} x - \frac{1}{2} \log 2\pi |\Sigma|\right]$$

$$= \frac{1}{2} \log 2\pi |\Sigma| + \int \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) \left(-\frac{1}{2} x^T \Sigma^{-1} x\right)$$

$$= \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\Sigma| + \underline{\int p(x) \left(-\frac{1}{2} x^T \Sigma^{-1} x\right) dx}$$

$$= \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\Sigma| + \frac{1}{2} \mathbb{E}\left[x^T \Sigma^{-1} x\right]$$

$$+ \frac{1}{2} \mathrm{Tr}\left[\mathbb{E}[x^T \Sigma^{-1} x]\right]$$

$$+ \frac{1}{2} \mathrm{Tr}\left[\Sigma^{-1} \overset{\Sigma}{\underset{I}{\cancel{\mathbb{E}[xx^T]}}}\right]$$

$$+ \frac{n}{2}$$

$$= \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\Sigma| + \frac{n}{2}$$

1b

4.1

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

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

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

Going 1 piece at a time:

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

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

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

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

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

$$= -\int_{x_i} q_i(x_i) \left[-\frac{1}{2} x_i \Sigma_{ii}^{-1} x_i - \frac{1}{2} \log 2\pi |\Sigma_{ii}| \right] \cancelto{1}{\int_{x_j} q_{j|i}(x_j | x_i)\, dx_j}\, dx_i$$

$$= H[q_i]$$

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

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

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

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

1b Continued

4.1

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

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

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

$$\check{\phi}_i = -\frac{1}{2} J_{ij} x_i^2$$

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

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

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

1c

4.1

c) $\partial_{\Sigma_{ij}} \varphi_\beta(q) = 0 = -J_{ij} + \frac{1}{2}\partial_{\Sigma_{ij}} \log \left| \begin{bmatrix} \hat\Sigma_{ii} & \hat\Sigma_{ij} \\ \hat\Sigma_{ji} & \hat\Sigma_{jj} \end{bmatrix} \right|$

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

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

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

1d

$\boxed{4.1}$

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

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

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

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

Note that from part (c), we had

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

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

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

$$= -1 + \frac{\hat{\Sigma}_{ii}\hat{\Sigma}_{jj}}{\hat{\Sigma}_{ii}\hat{\Sigma}_{jj} - \hat{\Sigma}_{ij}^2}$$

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

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

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

$$= -\frac{1}{2} \log 2\pi - \frac{1}{2} \frac{x_0^2}{1} + \sum_{i=1}^{N} \left(-\frac{1}{2} \log (2\pi) - \frac{1}{2} (x_i - \alpha x_{i-1})^2 \right)$$

$$+ \sum_{i=0}^{N} \left(-\frac{1}{2} \log (2\pi\sigma^2) - \frac{1}{2} \frac{(y_i - \beta x_i)^2}{\sigma^2} \right)$$

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

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

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

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

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)

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

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

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

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

Define $s = \sigma^{-2}$. Then

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

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

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

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

Problem 3

3a

4.3

a) Define 1 factor per node & 1 vertex per edge. For factor i, the factor potential is given by

$$f_i\left(N(i)\right) = \mathbb{1}\left\{\sum_{x_{ij} \in N(i)} x_{ij} \leq 1\right\} \prod_{x_{ij} \in N(i)} \exp\left(\frac{\beta}{2} W_{ij} x_{ij}\right)$$

3b

4.3

b) Given: Initialize $m^0_{i\to a}(x_i) = 1$ $\quad m^0_{a\to i}(x_i) = 1$

Update $m^{t+1}_{i\to a}(x_i) = \prod_{b\in N(i)\setminus\{a\}} m^t_{b\to i}(x_i)$ node to factor

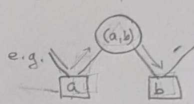
a = factor

i = node

$m^{t+1}_{a\to i}(x_i) = \sum_{x_{N(a)}\setminus\{i\}} f_a(x_{N(a)}) \prod_{j\in N(a)\setminus\{i\}} m^t_{j\to a}(x_i)$ factor to node

Goal: Messages $m^t_{i\to j}(x_{ij})$ for each edge $(i,j)\in\mathcal{E}$ in original graph

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



$m^{t+1}_{(a,b)\to b} = m^t_{a\to(a,b)}$ i.e. nodes forward messages from factors

$$m^{t+1}_{a\to(a,b)}(x_{(a,b)}) = \sum_{x_{N(a)}\setminus\{(a,b)\}} f_a(x_{N(a)}) \prod_{(c,a)\in N(a)\setminus\{(a,b)\}} m^t_{(c,a)\to a}(x_{(c,a)})$$

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

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 $\prod_{(c,a)\in N(a)\setminus\{a,b\}} m^t_{c\to a}$. If another factor is active, its factor potential times the other messages is $\exp\left(\frac{\beta W_{c\to a}}{2}\right) m_{c\to a}(1) \prod_{d\in N(a)\setminus\{b,c\}} m^t_{d\to a}(0)$ since other edges must also be 0

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

3c & 3d

4.3

c) $$\tilde{m}^t_{a\to b} \overset{\text{def}}{=} \frac{1}{\beta}\log\left(\frac{m^t_{a\to b}(1)}{m^t_{a\to b}(0)}\right)$$

$$= \frac{1}{\beta}\log\left(\frac{\exp\left(\frac{\beta W_{ab}}{2}\right)\prod_{c\in N(a)\setminus\{b\}} m^{t-1}_{c\to a}(0)}{\prod_{c\in N(a)\setminus\{b\}} m^{t-1}_{c\to a}(0) + \sum_{c\in N(a)\setminus\{b\}} \exp\left(\frac{\beta W_{ca}}{2}\right) m^{t-1}_{c\to a}(1)\prod_{d\in N(a)\setminus\{b,c\}} m^{t-1}_{d\to a}(0)}\right)$$

$$= \frac{1}{\beta}\log\left(\frac{\exp\left(\frac{\beta W_{ab}}{2}\right)}{1+\sum_{c\in N(a)\setminus\{b\}} \exp\left(\frac{\beta W_{ca}}{2}\right) \underbrace{m^{t-1}_{c\to a}(1)/m^{t-1}_{c\to a}(0)}_{=\exp(\beta \tilde{m}_{c\to a})}}\right)$$

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

$$\tilde{m}^t_{a\to b} = \frac{1}{\beta}\log\left(\exp\left(\frac{\beta W_{ab}}{2}\right)\right) - \frac{1}{\beta}\log\left(1+\sum_{c\in N(a)\setminus\{b\}}\exp\left(\frac{\beta W_{ca}}{2}+\beta\tilde{m}_{c\to a}\right)\right)$$

$$\lim_{\beta\to\infty}\tilde{m}^t_{a\to b} = \frac{W_{ab}}{2} - \max\left\{0, \max_{c\in N(a)\setminus\{b\}}\left\{\tilde{m}_{c\to a}+\frac{W_{ca}}{2}\right\}\right\}$$

b/c $\log(1+\sum\exp) \geq 0$ but $\max_{c\in N(a)\setminus\{b\}}\left\{\tilde{m}_{c\to a}+\frac{W_{ca}}{2}\right\}$ can be negative

To compute the MPC, I'd iterate the $\tilde{m}^t_{a\to b}$ messages in the above form (i.e. considering limit as $\beta\to\infty$)

Problem 4

4a & 4b First Half

4.4

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

$$\Longrightarrow \quad \hat{\lambda}_i = \frac{1}{K} \sum_{k=1}^{K} \left(x_i^{(k)}\right)^2$$

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

b) $$\ell_{ij}(\hat{\lambda}_{ij}, \hat{\lambda}_i, \hat{\lambda}_j) \stackrel{\text{def}}{=} \frac{1}{K} \sum_{k=1}^{K} \log \frac{p(x_i^{(k)}, x_j^{(k)}; \hat{\lambda}_{ij}, \hat{\lambda}_i, \hat{\lambda}_j)}{p(x_i^{(k)}; \hat{\lambda}_i)\, p(x_j^{(k)}; \hat{\lambda}_j)}$$

define $\hat{\Sigma}_{ij} = \begin{bmatrix} \hat{\lambda}_i & \hat{\lambda}_{ij} \\ \hat{\lambda}_{ji} & \hat{\lambda}_j \end{bmatrix}$

$$= \frac{1}{K} \sum_{k=1}^{K} -\frac{1}{2} \log\left((2\pi)^2 |\hat{\Sigma}_{ij}|\right) - \frac{1}{2} x_{ij}^{T(k)} \hat{\Sigma}_{ij}^{-1} x_{ij}^{(k)} + \frac{1}{2} \log(2\pi \hat{\lambda}_i) + \frac{1}{2} x_i^{(k)} \hat{\lambda}_i^{-1} x_i^{(k)}$$

$$+ \frac{1}{2} \log(2\pi \hat{\lambda}_j) + \frac{1}{2} x_j^{(k)} \hat{\lambda}_j^{-1} x_j$$

$$= \frac{1}{K} \sum_{k=1}^{K} -\frac{1}{2} \log\left((2\pi)^2 (\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}^2)\right) - \frac{1}{2} \frac{1}{\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}^2} x_{ij}^{(k)T} \begin{bmatrix} \lambda_j & -\lambda_{ji} \\ -\lambda_{ij} & \lambda_i \end{bmatrix} x_{ij}^{(k)}$$

$$+ \frac{1}{2} \log(2\pi \hat{\lambda}_i) + \frac{1}{2} x_i^{(k)} \hat{\lambda}_i^{-1} x_i^{(k)} + \frac{1}{2} \log(2\pi \hat{\lambda}_j) + \frac{1}{2} x_j^{(k)} \lambda_j^{-1} x_j^{(k)}$$

Group log terms:

$$= \frac{1}{K} \sum_{k=1}^{K} -\frac{1}{2} \log\left(\frac{\cancel{(2\pi)}^2 (\hat{\lambda}_i \hat{\lambda}_j - \lambda_{ij}^2)}{\cancel{2\pi} \hat{\lambda}_i \cdot \cancel{2\pi} \hat{\lambda}_j}\right) - \frac{1}{2} \frac{x_i^{(k)2} \hat{\lambda}_j + x_j^{(k)2} \hat{\lambda}_i - 2\hat{\lambda}_{ji} x_i x_j}{\hat{\lambda}_i \hat{\lambda}_j - \hat{\lambda}_{ij}^2}$$

$$+ \frac{1}{2} \frac{x_i^{(k)2}}{\lambda_i} + \frac{1}{2} \frac{x_j^{(k)2}}{\lambda_j}$$

4b Second Half & 4c

4.4

b cont)

$$-2\frac{1}{K}\sum x_i^{k^2} x_j^2$$

$$\ell_{ij}(\hat{\lambda}_{ij}, \hat{\lambda}_i, \hat{\lambda}_j) = \frac{1}{2}\log\frac{\hat{\lambda}_i\hat{\lambda}_j}{\hat{\lambda}_i\hat{\lambda}_j - \hat{\lambda}_{ij}} - \frac{1}{2}\frac{1}{K}\sum_{k=1}^{K}\frac{x_i^{(k)^2}\frac{1}{K}\sum x_j^{(k)^2} + x_j^{(k)^2}\frac{1}{K}\sum x_i^{(k)^2} - \ldots}{\hat{\lambda}_i\hat{\lambda}_j - \hat{\lambda}_{ij}}$$

$$+ \cancel{\frac{1}{2}\frac{1}{K}\sum_{k}\frac{x_i^{(k)2}}{\frac{1}{K}\sum x_i^{(k)2}}} \frac{1}{2} + \cancel{\frac{1}{2}\frac{1}{K}\sum_{k}\frac{x_j^{(k)2}}{\frac{1}{K}\sum_{k} x_j^{(k)2}}} \frac{1}{2}$$

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

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

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 the mutual information from adding the edge.

Problem 6

6a & 6b & 6c

4.6

(a) Let edge $i \to j$ 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 $(i,j) \in \mathcal{E}$, the path $x_i \to x_j$ is not blocked $\Longrightarrow x_i \not\!\perp\!\!\!\perp x_j \mid x_U$

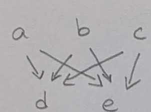
(b) Let x_i, x_j be non-adjacent. WLOG, let i come before j in the topological ordering. Define $U = x_{\pi_j}$. Note that $|U| \leq d_{in}$ by construction.

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

Proof: All edges $x_k \to x_j$ are blocked. There can be no immoralities at these $x_k \in x_{\pi_j}$ b/c edge $x_k \to x_j$ 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 $x_k \to x_j$ 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

$\boxed{4.6}$

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

Computational

Comp a(i) & Comp a(ii)

Comp

(a) (i)

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

$$= \sum C x_t \cancel{x_t} + w_t \qquad (\text{defn of } y_t)$$

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

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

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

(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 1

(b) (i) Starting w/

$$p(x_0 \ldots x_{T-1} \mid y_0 \ldots y_{T-1}) \propto p(x_0) \prod_{i=0}^{T-2} p(x_{i+1} \mid x_i) \prod_{i=0}^{T-1} p(y_i \mid x_i)$$

$$\propto \exp\left(-\frac{1}{2}\frac{(x_0-\mu_0)^2}{\Lambda_0} + \mu_0 x_0\right) \prod_{i=0}^{T-2} \exp\left(-\frac{1}{2}\frac{(x_{i+1}-Ax_i)^2}{Q} + Ax_i x_{i+1}\right)$$

$$\prod_{i=0}^{T-1} \exp\left(-\frac{1}{2}\frac{(y_i - Cx_i)^2}{R_i} + Cx_i y_i\right)$$

Rearranging & grouping

x_0^2 terms: $-\frac{1}{2}\frac{1}{\Lambda_0}x_0^2 - \frac{1}{2}\frac{A^2}{Q}x_0^2 - \frac{1}{2}\frac{C^2}{R_0}x_0^2 \Rightarrow \underline{J_{00} = \Lambda_0^{-1} + A^T Q^{-1} A + C^T R_0^{-1} C}$

x_0 terms: $\frac{\mu_0}{\Lambda_0}x_0 + \frac{C}{R_0}x_0 y_0 \Rightarrow \underline{h_0 = \Lambda_0^{-1}\mu_0 + C^T R_0 y_0}$

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

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

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

x_i^2 terms: $\frac{1}{Q} + \frac{C^2}{R_i} \Rightarrow \underline{J_{ii} = Q^{-1} + C^T R_i^{-1} C}$

for $i = T-1$

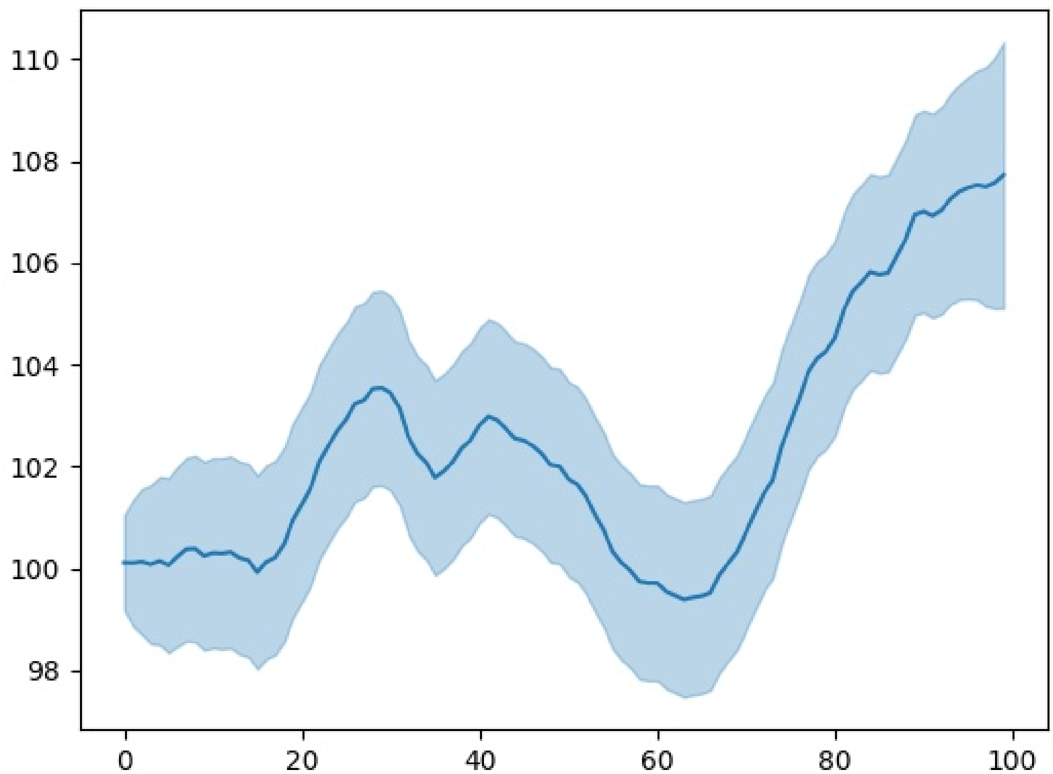
x_i terms: $\frac{C}{R_i}y_i \Rightarrow \underline{h_i = C^T R_i^{-1} y_i}$

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 +
                        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 +

    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)

    vars = np.array([inv(Js_hat[time]) for time in times])
    means = np.array([(inv(Js_hat[time]) @ hs_hat[time]) for time in times])
    return means, vars
```

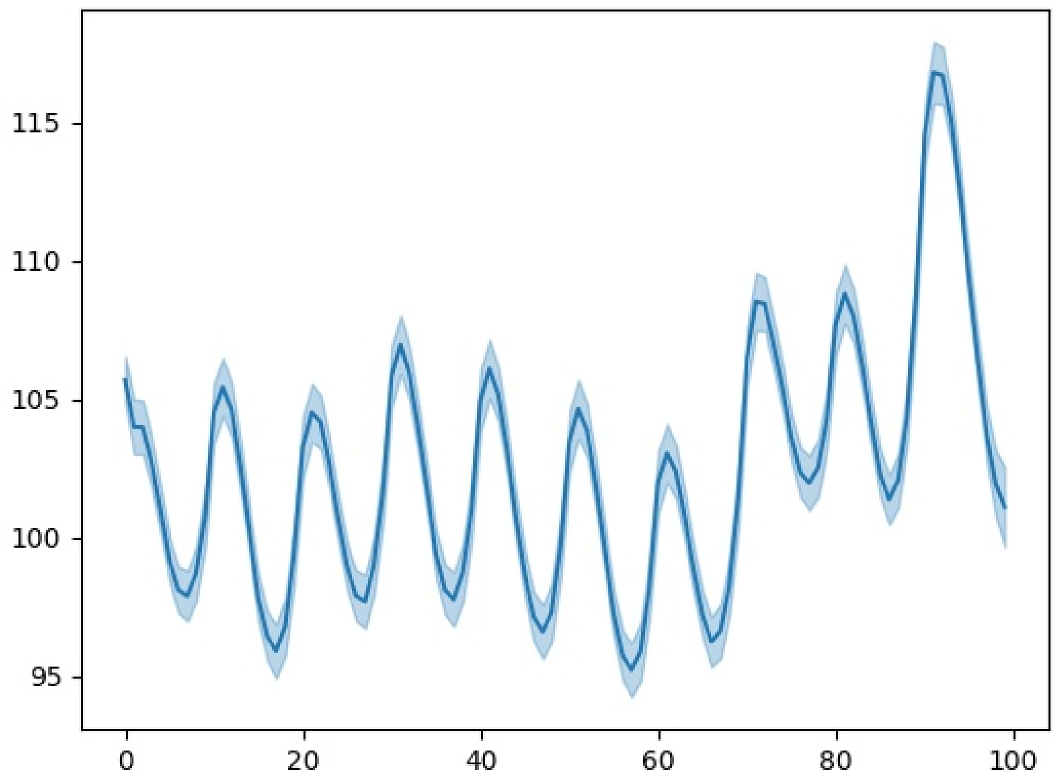
Comp b(iii)

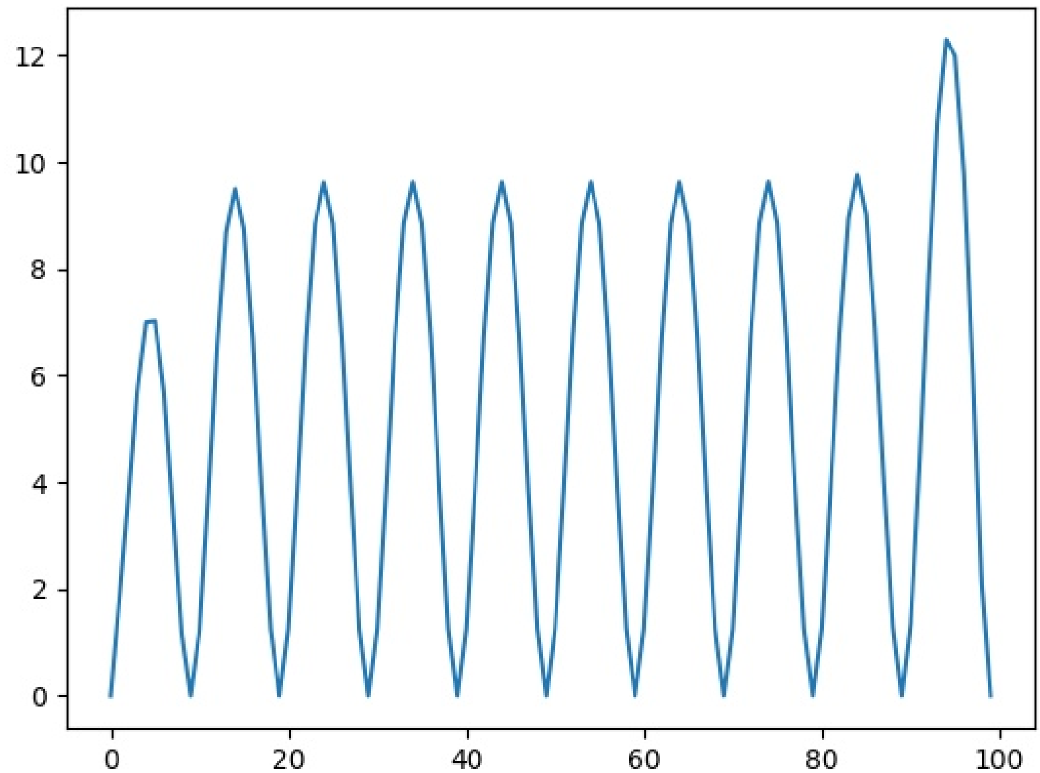



```
# b(iii)
A = np.array([[0.9999]])
C = np.copy(Chat)
Q = np.array([[1.]])
Rs = np.full(shape=(len(y), 1, 1), fill_value=Rhat)
mu_0 = np.array([100.])
Lambda_0 = np.array([[1.]])
times = np.arange(len(y))
means, vars = gaussian_belief_prop(y=np.expand_dims(y, axis=-1),
                        A=A, C=C, Q=Q, Rs=Rs, mu_0=mu_0, Lambda_0=Lambda_0)

p = plt.plot(times, means[:, 0])
plt.fill_between(times,
                 means[:, 0] - np.sqrt(vars[:, 0, 0]),
                 means[:, 0] + np.sqrt(vars[:, 0, 0]),
                 color=p[-1].get_color(),
                 alpha=0.3)
plt.savefig('Comp_b(iii).jpg')
plt.show()
```

Comp b(iv)

I used $\epsilon = 1e^{-}3$ and $\sigma = 1e^{3}$.






```
# b(iv)
A = np.array([[0.9999, 0.], [0.9999, 1.]])
C = np.array([[Chat[0, 0], 0.], [0., 1.]])
epsilon = 1e-3
sigma = 1e3
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.]])
means, vars = gaussian_belief_prop(y=y_aug, A=A, C=C, Q=Q, Rs=Rs, mu_0=mu_0, Lambda_0=Lamb

p = plt.plot(times, means[:, 0])
plt.fill_between(times,
                 means[:, 0] - np.sqrt(vars[:, 0, 0]),
                 means[:, 0] + np.sqrt(vars[:, 0, 0]),
                 color=p[-1].get_color(),
                 alpha=0.3)
plt.savefig('Comp_b(iv)-x.jpg')
```

```
plt.show()

p = plt.plot(times, vars[:, 1, 1])
plt.savefig('Comp_b(iv)-s.jpg')
plt.show()
```