HW2 — ISYE6416

Ashish Dhiman — ashish.dhiman9@gatech.edu

May 7, 2023

Contents

1 Question 1

1.1 part a: Soft Thresholding

We know

$$\hat{\beta}_{\text{Lasso}}(\lambda) \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \left\| y - X\beta \right\|_2^2 + \lambda \left\| \beta \right\|_1 \quad .$$

Given p =1 and X = I, we have

$$\begin{aligned}
\beta_{lasso} = \arg\min_{\beta} \frac{1}{2}(y-\beta)^2 + \lambda \left\| \beta \right\|_1 \\
Cost = F(\beta) + G(\beta) \\
\text{F is differentiable but G is not} \\
\text{By definition of sub differential and stationary} \\
\partial(Cost)|_{\beta_{lasso}} = 0 \quad (\partial \text{ is sub-differential.}) \\
\implies (\beta - y) + \lambda s = 0 \text{ where } s = \partial|G| = \begin{cases} sign(\beta), & \text{if } \beta \neq 0 \\ [-1,1], & \text{otherwise} \end{cases} \\
\text{If we simplify for } \beta \\
\beta_{lasso} = \begin{cases} y - \text{sign}(y)\lambda, & \text{if } |y| \geq \lambda \\ 0, & \text{otherwise} \end{cases} \\
= \text{sign}(y)\max(|y| - \lambda, 0)
\end{aligned} \tag{1}$$

The Soft thresholding looks as below:

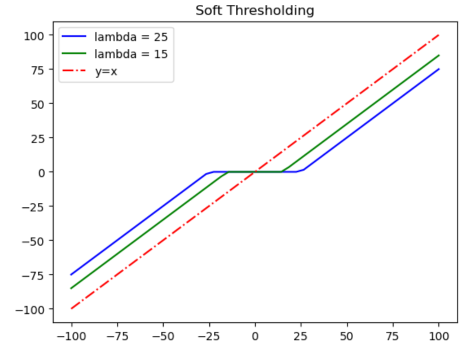

Figure 1: Loss (left) and Regression fits (right)

We know:

$$\beta_{ridge} = (X^T X + \lambda I)^{-1} X^T y$$

For Ridge Estimator in same single parameter setup we have:

$$\beta_{ridge} = (1+\lambda)^{-1} y$$

Differences between Lasso and Ridge

1. While Lasso explicitly does variable selection by making some beta coefficients 0. On the other hand Ridge shrinks beta coefficients towards 0 but not exactly 0.

2. There exists a close form solution for Beta in Ridge setup while Lasso can only be solved iteratively.

3. The penalty term in Lasso is L1 norm while in Ridge it's L2 norm. L1 is convex but L2 is convex and differentiable.

1.2 part b: Iterative algo for lasso

Proximal Gradient Descent

We need to simplify:

$$\begin{aligned}
x^{(k+1)} &= \arg\min_x f(x^{(k)}) + \langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \frac{L}{2}\left\| x - x^{(k)} \right\|^2 + \Omega(x) \\
&= \arg\min_x \langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \frac{L}{2}\left\| x - x^{(k)} \right\|^2 + \Omega(x) \\
&\text{We can add a constant to the minimisation:} \frac{1}{2L}\langle \nabla f(x^{(k)}), \nabla f(x^{(k)}) \rangle \\
x^{(k+1)} &= \arg\min_x \frac{1}{2L}\langle \nabla f(x^{(k)}), \nabla f(x^{(k)}) \rangle + \langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \frac{L}{2}\left\| x - x^{(k)} \right\|^2 + \Omega(x) \\
&\text{Divide by L} \\
x^{(k+1)} &= \arg\min_x \frac{1}{2L^2}\langle \nabla f(x^{(k)}), \nabla f(x^{(k)}) \rangle + \frac{1}{L}\langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \frac{1}{2}\left\| x - x^{(k)} \right\|^2 + \frac{1}{L}\Omega(x) \\
x^{(k+1)} &= \arg\min_x \frac{1}{2L^2}\left\| \nabla f(x^{(k)}) \right\|_2 + \frac{1}{L}\langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \frac{1}{2}\left\| x - x^{(k)} \right\|^2 + \frac{1}{L}\Omega(x) \\
&\text{now } \|x+y\|^2 = \|x\|^2 + \|y\|^2 + 2 < x, y > \\
x^{(k+1)} &= \arg\min_x \frac{1}{2}\left\| (x - x^{(k)}) + \frac{1}{L}\nabla f(x^{(k)}) \right\|^2 + \frac{1}{L}\Omega(x) \\
\implies x^{(k+1)} &= \arg\min_x \frac{1}{2}\left\| x - \left(x^{(k)} - \frac{1}{L}\nabla f(x^{(k)}) \right) \right\|^2 + \frac{1}{L}\Omega(x)
\end{aligned} \tag{2}$$

Hence proved

For Proximal Coordinate Gradient Descent

If we move in the direction $d = e_j(x_j - x_j^{(k)})$ with $e_j = (0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 is at the jth coordinate.

$$x_j^{(k+1)} = \arg\min_{x_j} f(x^{(k)}) + \langle e_j^\top \nabla f(x^{(k)}), x_j - x_j^{(k)} \rangle + \frac{L}{2} \left\| x_j - x_j^{(k)} \right\|^2 + \Omega_j(x_j)$$

Similar to proximal descent, let's drop $f(x_k)$ and add $\frac{1}{2L} \left\| e_j \nabla f(x^k) \right\|^2$

$$x^{(k+1)} = \arg\min_{x} \frac{1}{2L} \left\| e_j \nabla f(x^k) \right\|^2 + \langle e_j^\top \nabla f(x^{(k)}), x_j - x_j^{(k)} \rangle + \frac{L}{2} \left\| x_j - x_j^{(k)} \right\|^2 + \Omega_j(x_j)$$

Dividing by L and aggregating terms

$$= \arg\min_{x_j} \frac{1}{2} \left\| (x_j - x_j^{(k)}) + \frac{1}{L} e_j^\top \nabla f(x^{(k)}) \right\|^2 + \frac{1}{L}\Omega_j(x_j)$$

$$\implies x_j^{(k+1)} = \arg\min_{x_j} \frac{1}{2} \left\| x_j - \left(x_j^{(k)} - \frac{1}{L} e_j^\top \nabla f(x^{(k)}) \right) \right\|^2 + \frac{1}{L}\Omega_j(x_j) \tag{3}$$

Hence proved

Please don't mind the typos :)

Proximal GD for Lasso

We have derived Proximal GD as:

$$\implies x^{(k+1)} = \arg\min_{x} \frac{1}{2} \left\| x - \left(x^{(k)} - \frac{1}{L} \nabla f(x^{(k)}) \right) \right\|^2 + \frac{1}{L}\Omega(x)$$

For the lasso setup

$$x = \beta, f(\beta) = \frac{1}{2} \left\| y - X\beta \right\|^2, \Omega(\beta) = \lambda \sum_{j=1}^{p} |\beta_j|$$

$$\implies \beta^{(k+1)} = \arg\min_{\beta} \frac{1}{2} \left\| \beta - \left(\beta^{(k)} - \frac{1}{L} \nabla f(\beta^{(k)}) \right) \right\|^2 + \frac{1}{L} \lambda \sum_{j=1}^{p} |\beta_j|$$

$$\beta^{(k+1)} = \arg\min_{\beta} \frac{1}{2} \left\| \beta - \hat{\beta} \right\|^2 + \frac{1}{L} \lambda \sum_{j=1}^{p} |\beta_j| \quad (\text{where } \hat{\beta} = (\beta^{(k)} - \frac{1}{L} \nabla f(\beta^{(k)})))$$

$$\hat{\beta} = \beta^{(k)} - \frac{1}{L} \nabla f(\beta^{(k)}) = \beta^{(k)} - \frac{1}{L} X^T (X\beta^k - y)$$

Using the soft thresholding setup

$$\beta^{(k+1)} = \begin{cases} \hat{\beta} - \operatorname{sign}(\hat{\beta}) \frac{\lambda}{L}, & \text{if } |\hat{\beta}| \geq \frac{\lambda}{L} \\ 0, & \text{else} \end{cases}$$

$$\implies \beta^{(k+1)}(\lambda) = \operatorname{sign}(\hat{\beta}) \max(|\hat{\beta}| - \frac{\lambda}{L}, 0) \tag{4}$$

The above equation defines the iteration to be used for Proximnal GD. Also note that $\hat{\beta}$ is very similar to typical GD. Hence in essence we will do GD on beta and then add soft thresholding.

Proximal Coordinate GD for Lasso

This will be very similar to Proximal GD step from above, only the iteration would update a single dimension of β at each iteration

$$\implies \beta_j^{(k+1)}(\lambda) = \operatorname{sign}(\hat{\beta}_j) \max(|\hat{\beta}_j| - \frac{\lambda}{L}, 0)$$

$$\hat{\beta}_j = \beta_j^{(k)} - \frac{e_j^T}{L} X^T (X\beta^k - y)$$

Please don't mind the typos :)

Iterates on synthetic data

I have chosen synthetic data with X as $[1, x, x^2], x = np.linspace(-1, 1, 1000)$ and $y = x^2 + Sin(x)$

If we fit lasso on this data with lambda = 1, with sklearn we get coef of 1 and x as 0 as expected but since y is linear in x^2 we get non zero coef there.

For the same settings, from my proximal GD implementation I get the following:

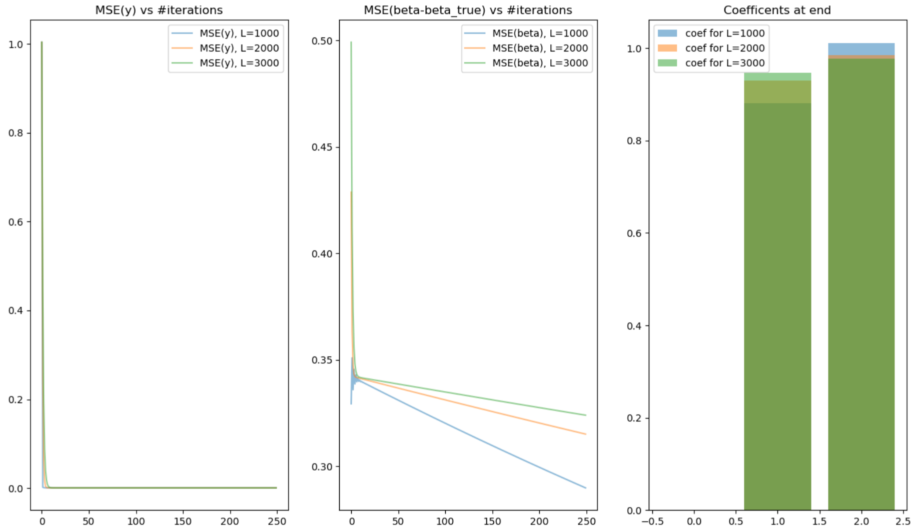

Figure 2: Loss (left) and Regression fits (right)

For the same setting but coordinate proximal descent we get:

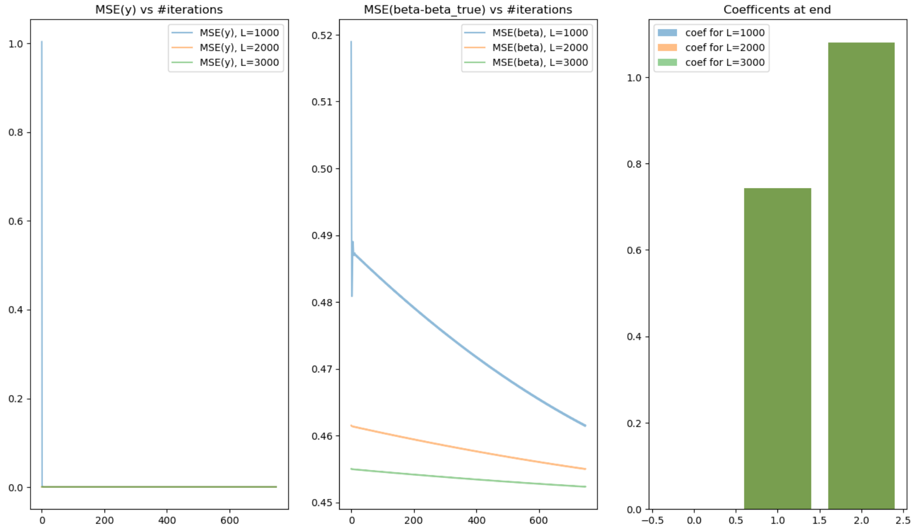

Figure 3: Loss (left) and Regression fits (right)

Bonus

1. if Ω is 0, the Lasso estimate is same as OLS, or the normal Gradient Descent
2. if Ω is $\|x\|_2$, the problem/ estimate is similar to Ridge regression, with a scaled λ
3. If Ω is indicator function for set C, the problem becomes a constrained optimisation problem, where we incentivise the solution in set C.

2 Question 2: Weighted Regression

2.1 part a: MAP estimate

We are given (x_i, y_i), $i = 1, \ldots, n$ of n independent examples, and $y_i = \theta^T x_i + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, {\sigma_i}^2)$ and let $\beta \sim \mathcal{N}(0, s^2 I)$

$$\begin{aligned}
p(y_i|\beta, x_i) \sim \mathcal{N}(\beta^T x_i, {\sigma_i}^2) &= \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{(y_i - \beta^T x_i)^2}{2\sigma_i^2}) \\
\implies \log p(y_i|\beta, x_i) &= const - \frac{(y_i - \beta^T x_i)^2}{2\sigma_i^2}\text{Also} \\
p(\beta) \sim \mathcal{N}(0, {\sigma_i}^2) &= \frac{1}{(2\pi)^{p/2} * det(S)^{1/2}} \exp(-\frac{1}{2}(\beta - 0)^T S^{-1} (\beta - 0)) \\
\text{where } S &= \begin{bmatrix} 1/s^2 & 0 & 0 \\ 0 & 1/s^2 & 0 \\ 0 & 0 & 1/s^2 \end{bmatrix}_{p*p} \\
\implies \log p(\beta) &= const - \lambda||\beta||_2^2 \quad (\lambda = \frac{1}{2s^2})
\end{aligned} \tag{5}$$

$$\begin{aligned}
\beta_{map} &= arg \max_{\beta} p(\beta|y_1, y_2, ..., y_n, x_1, x_2, ..., x_n) \\
&= \frac{p(y_1, y_2, ..., y_n|x_1, x_2, ..., x_n, \beta) * p(\beta)}{p(Y|X)} \\
&\propto (\prod_{i=1}^{n} p(y_i|\beta, x_i)) * p(\beta) \\
&\text{Taking log this becomes:} \\
\beta_{map} &= arg \max_{\beta} \sum_{i=1}^{n} \log p(y_i|\beta, x_i) + \log p(\beta) \\
&\text{From above} \\
\beta_{map} &= arg \max_{\beta} -[\sum_{i=1}^{n} \frac{(y_i - \beta^T x_i)^2}{2\sigma_i^2} + \lambda||\beta||_2^2] \\
&= arg \min_{\beta} [\sum_{i=1}^{n} \frac{(y_i - \beta^T x_i)^2}{\sigma_i^2} + \lambda||\beta||_2^2] \\
&= arg \min_{\beta} [(X\beta - y)^T W (X\beta - y) + \lambda||\beta||_2^2] \\
\text{where } X = [x_{ij}]_{n*p}, y = [y_i]_{n*1}, W &= \begin{bmatrix} 1/\sigma_i^2 & \\ 0 & 1/\sigma_i^2 \end{bmatrix}_{n*n}
\end{aligned} \tag{6}$$

2.2 part b: Closed form sol'n

Let

$$L = (X\beta - y)^T W(X\beta - y) + \lambda||\beta||_2^2$$

$$\begin{gathered}
\frac{\partial L}{\partial \beta} = 2X^T W(X\beta_{ridge} - y) + 2\lambda y = 0 \\
(X^T WX + \lambda I)\beta = X^T Wy \\
\implies \beta_{ridge} = (X^T WX + \lambda I)^{-1} X^T Wy \\
\text{Let } K = (X^T WX + \lambda I)^{-1} X^T W \\
\implies \beta_{ridge} = Ky = K(X\beta^* +) \\
\implies E(\beta_{ridge}) = E(KX\beta^*) + 0 = KX\beta^* \\
= (X^T WX + \lambda I)^{-1} X^T WX\beta^* \\
\text{Similarly we can get variance as} \\
Var(\beta) = KVar(Y)K^T \\
= (X^T WX + \lambda I)^{-1} X^T W[\sigma_i^2]_{n*n} W^T X(X^T WX + \lambda I)^{-1} \\
\text{The above moments define the distribution of } \beta_{ridge}
\end{gathered} \tag{7}$$

2.3 part c: expectation

$$\begin{gathered}
\text{For fixed test point x and lambda} \\
E[x^T \hat{\beta}(\lambda)] - x^T \beta^* = x^T [E(\hat{\beta}(\lambda)) - \beta^*] \\
= x^T ((X^T WX + \lambda I)^{-1} X^T WX\beta^* - \beta^*)
\end{gathered} \tag{8}$$

2.4 part d: variance

We know

$$E[X^2] = Var(X) + (E[X])^2$$

, using this we

$$\begin{gathered}
\text{For fixed test point x and given lambda} \\
x^T \hat{\beta}(\lambda) = x^T (X^T WX + \lambda I)^{-1} X^T Wy = x^T (X^T WX + \lambda I)^{-1} X^T W(X\beta^* + \epsilon) \\
= K\beta^* + KX^{-1}\epsilon \\
E[x^T \hat{\beta}(\lambda)] = x^T ((X^T WX + \lambda I)^{-1} X^T WX\beta^* = K\beta^* \\
\implies X = x^T \hat{\beta}(\lambda) - E[x^T \hat{\beta}(\lambda)] = KX^{-1}\epsilon = Q\epsilon \\
E(X^2) = Var(Q\epsilon) + Q(E(\epsilon)) = Q^T [\sigma_i^2]_{n*n} Q \\
\text{where } Q = x^T (X^T WX + \lambda I)^{-1} X^T W
\end{gathered} \tag{9}$$

Please don't mind the typos :)

2.5 part e: MSE

For the setup given,

$$y = X\beta, W = \begin{bmatrix} 1/2 & \\ 0 & 1/1 \end{bmatrix}$$

The plot of MSE is given below for $x_test = (1, 0.5)$

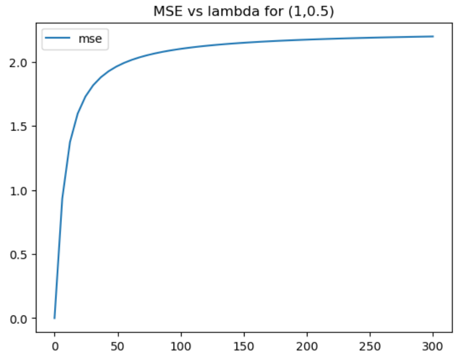

Figure 4: Loss (left) and Regression fits (right)

2.6 part f: MSE

Similarly the plot of MSE is given below for $x_test = (1, 2)$

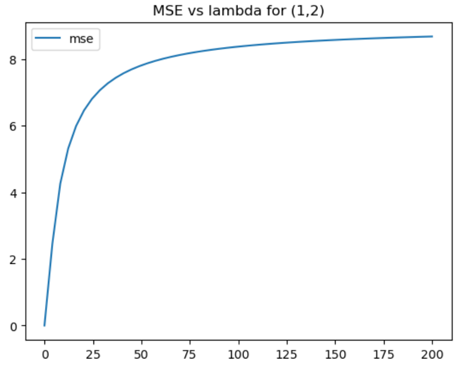

Figure 5: Loss (left) and Regression fits (right)

The MSE levels in this part are much higher at 8 vs 2 in the previous part. This is because while both 0.5 and 2 are equally away from training data of 1 and 1.5. However, there is much higher weight on 1 (since it has lower sigma), and consequently test point 2 shows higher MSE since it is farther away from it vis-a-vis test point 0.5

3 Question 3: Housing Dataset

For the following question I have done the below preprocessing steps:

1. One Hot Encode the Status column and only keep 2 categories to prevent dependent columns
2. Scale the rest X columns namely Bedroom, Bathroom, Size, Price/Sq ft

Before proceeding with Ridge and Lasso Regression, we have also fit OLS to get a benchmark. The plot of y_{ols} vas y_{true} is given below:

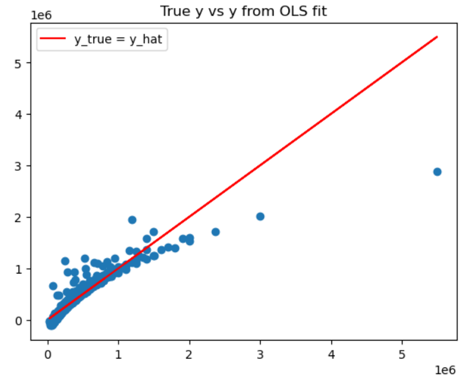

Figure 6: Loss (left) and Regression fits (right)

From above figure we see that there are some outliers, and the y variable might also need some transformation. The SSR from OLS model is $1.6279 * 10^{13}$.

3.1 part a: Ridge Regression

After the above steps, we get the following CV curve for lambda:

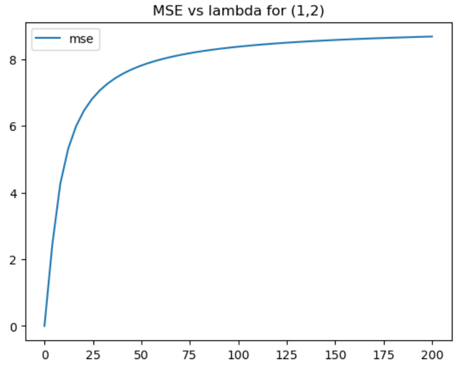

Figure 7: Loss (left) and Regression fits (right)

Thus we see that as we increase the regularisation the beta values shrink to 0 and the error values stabilise to around 8.

$$\implies \lambda_{best} = 80$$

For the selected lambda, model coefficients are:

Interccept	390293
Bedrooms	-2916
Bathrooms	34505.8
Size	169675
Price/SQ.Ft	194113
Status_Regular	25566.7
Status_Short Sale	-15644

Table 1: Model Coefficients for Ridge Regression with lambda = 80

The SSR for the above model is $1.6975 * 10^{13}$

3.2 part b: Lasso Regression

With the same setup as above, we get the following Lasso CV plot:

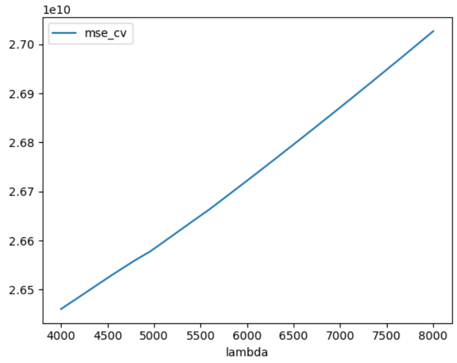

Figure 8: Loss (left) and Regression fits (right)

From the above plot the best lambda value is 4000.
The lasso path of coefficients is

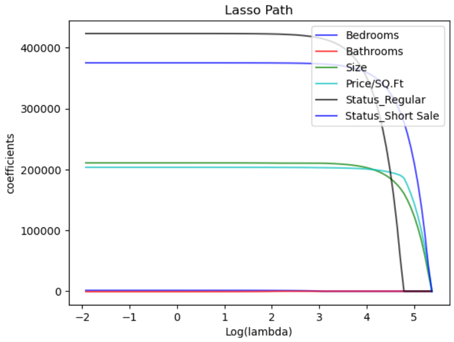

Figure 9: Loss (left) and Regression fits (right)

Thus we see that as we increase lambda, all the coefficients become 0. Also from the plot it might seem that the coefficients are only positive, however the lowest line lies below zero, and the scale of the axis blurs this.

For the selected lambda, model coefficients are:

Interccept	385600
Bedrooms	0
Bathrooms	7058.2
Size	201378
Price/SQ.Ft	213482
Status_Regular	0
Status_Short Sale	-3437

Table 2: Model Coefficients for Lambda Regression with lambda = 4000

The SSR for the above model is $1.6439 * 10^{13}$

Hence lasso does variable selection, since we have dropped Status.Regular and Bedroom variable.

4 Question 4: Multivariate Gaussian

4.1 part a: MLE Estimate

We are given $x_i \sim \mathcal{N}(\mu, \Sigma)$, where:

$$x_i \in \mathcal{R}^d, \mu \in \mathcal{R}^d, \Sigma \in \mathcal{R}^d$$

To find MLE estimate of μ, Σ, we need to maximise the likelihood, i.e.

$$\mu_{mle}, \Sigma_{mle} = arg \max_{(\mu,\Sigma)} = Lik(x_1, x_2 ... x_n | \mu, \Sigma)$$

$$\begin{aligned}
Lik &= p(x_1, x_2 ... x_n | \mu, \Sigma) \\
&= p(x_1|\mu,\Sigma) * p(x_2|\mu,\Sigma)...p(x_n|\mu,\Sigma) = \prod_{i=1}^{n} p(x_i|\mu,\Sigma) \\
&= \prod_{i=1}^{n} \frac{1}{2\pi^{d/2} det(\Sigma)^{1/2}} * \exp(\frac{-1}{2}(x_i - \mu)^T \Sigma^{-1}(x_i - \mu)) \quad \text{(From pdf)} \\
&= \frac{1}{2\pi^{nd/2} det(\Sigma)^{n/2}} * \exp \frac{-1}{2} \sum_{i=1}^{n} ((x_i - \mu)^T \Sigma^{-1} (x_i - \mu)) \qquad (10) \\
&\text{Taking log and dropping constant terms, we get} \\
\log Lik &\propto -\frac{n}{2} log(det(\Sigma)) \frac{-1}{2} \sum_{i=1}^{n} ((x_i - \mu)^T \Sigma^{-1} (x_i - \mu)) \\
&\propto -n \log(det(\Sigma)) - \sum_{i=1}^{n} ((x_i - \mu)^T \Sigma^{-1} (x_i - \mu))
\end{aligned}$$

Now max -x is same as min x. Therefore let's find gradient wrt to μ, Σ

$$\begin{aligned}
\nabla_\mu \log Lik = 0 + \sum_{i=1}^{n} 2\Sigma^{-1}(x_i - \mu) &= 0 \\
\sum_{i=1}^{n} \mu &= \sum_{i=1}^{n} (x_i) \qquad (11) \\
&\text{Thus we have:} \\
\implies \mu_{mle} &= \frac{1}{n} \sum_{i=1}^{n} (x_i)
\end{aligned}$$

Similarly taking gradient wrt Σ^{-1}

$$\begin{aligned}
\nabla_{\Sigma^{-1}} \log Lik = \nabla_{\Sigma^{-1}}(-n \log(det(\Sigma))) &+ \sum_{i=1}^{n} ((x_i - \mu)^T (x_i - \mu)) = 0 \\
&\text{From } det(A) = 1/det(A^{-1}) \\
\nabla_{\Sigma^{-1}}(n \log(det(\Sigma))) = n(\Sigma^{-1})^{-1T} &\quad \text{From } \nabla_A log(det(A) = (A^{-1})^T \qquad (12) \\
\implies \nabla_{\Sigma^{-1}} \log Lik = -n\Sigma^T &+ (\sum_{i=1}^{n} ((x_i - \mu)(x_i - \mu)^T))^T = 0 \\
\implies \Sigma_{mle} &= \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T
\end{aligned}$$

4.2 part b: Biased/Unbiased ?

Bias check for Σ_{mle}

$$\begin{aligned}
\implies E[\mu_{mle}] &= E[\frac{1}{n}\sum_{i=1}^{n}(x_i)] \\
&= \frac{1}{n}\sum_{i=1}^{n} *E[x_i] \\
&= \frac{1}{n} * n\mu = \mu \\
&\text{Hence MLE estimate for mean is unbiased}
\end{aligned} \tag{13}$$

Bias check for μ_{mle}

$$\begin{aligned}
\implies E[\Sigma_{mle}] &= E[\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})(x_i - \hat{\mu})^T] \\
&= \frac{1}{n}\sum_{i=1}^{n} E[x_i x_i^T - x_i\hat{\mu}^T - \hat{\mu}x_i^T + \hat{\mu}\hat{\mu}^T] \\
&= \frac{1}{n}\sum_{i=1}^{n}(E[x_i x_i^T] - E[\hat{\mu}\hat{\mu}^T]) \\
\implies E[x_i x_i^T] &= (\mu\mu^T + \Sigma) \quad (\Sigma = E[x_i x_i^T]) \\
\implies E[\hat{\mu}\hat{\mu}^T] &= E[(\frac{1}{n}\sum_{i=1}^{n}(x_i))(\frac{1}{n}\sum_{j=1}^{n}(x_j))^T] \\
&= \frac{1}{n^2}(n^2\mu\mu^T + n * \Sigma) \\
\implies E[\Sigma_{mle}] &= \frac{1}{n}\sum_{i=1}^{n}((\mu\mu^T + \Sigma) - (\mu\mu^T + \frac{\Sigma}{n}) \\
&\neq \Sigma \\
&\text{Hence MLE estimate for cov is biased}
\end{aligned} \tag{14}$$

5 References:

1. https://ashim95.github.io/docs/notes/mle_for_mv_gaussian.pdf
2. https://arxiv.org/pdf/1509.09169.pdf
3. https://www.stat.cmu.edu/~larry/=stat401/lecture-24.pdf
4. https://www.kaggle.com/code/residentmario/soft-thresholding-with-lasso-regression/notebook
5. https://towardsdatascience.com/unboxing-lasso-regularization-with-proximal-gradient-me
6. Collaborators: Rakesh, Dipendra