STAT 221 - ASSIGNMENT 3

GREG TAM, STUDENT ID: 70908239

1. (a) By Adam's law, we have that

$$\begin{aligned}\mathbb{E}[y_n|\boldsymbol{x}_n] &= \mathbb{E}[\mathbb{E}[y_n|\lambda_n]\,|\boldsymbol{x}_n] \\ &= \mathbb{E}[h(\lambda_n)|\boldsymbol{x}_n] \\ &= \mathbb{E}[h(\boldsymbol{x}_n'\boldsymbol{\theta}^*)|\boldsymbol{x}_n] \\ &= h(\boldsymbol{x}_n'\boldsymbol{\theta}^*)\end{aligned}$$

Now consider

$$f(y_n|\lambda_n) = \exp\left\{\frac{\lambda_n y_n - b(\lambda_n)}{\phi}\right\}\cdot c(y_n,\phi)$$

Taking logs, we get

$$\ell(\lambda_n) = \log f(y_n|\lambda_n) = \frac{\lambda_n y_n - b(\lambda_n)}{\phi} + \log c(y_n,\phi)$$

Then differentiating, we get

$$\ell'(\lambda_n) = \frac{y_n - b'(\lambda_n)}{\phi}$$

Taking expectations on both sides and using the fact that $\mathbb{E}[\ell'(\lambda_n)] = 0$, we have

$$\mathbb{E}\left[\frac{y_n - b'(\lambda_n)}{\phi}\right] = 0$$

which implies

$$\mathbb{E}[y_n|\lambda_n] = b'(\lambda_n)$$

Using the same logic as above, we have

$$\begin{aligned}\mathbb{E}[y_n|\boldsymbol{x}_n] &= \mathbb{E}[\mathbb{E}[y_n|\lambda_n]\,|\boldsymbol{x}_n] \\ &= \mathbb{E}[b'(\lambda_n)|\boldsymbol{x}_n] \\ &= \mathbb{E}[b'(\boldsymbol{x}'\boldsymbol{\theta}^*)|x_n] \\ &= b'(\boldsymbol{x}'\boldsymbol{\theta}^*) \\ &= b'(\lambda_n)\end{aligned}$$

(b) For this part, we will use the the fact that under regularity conditions, we have

$$\mathbb{E}[-\ell''(\lambda_n)] = \mathbb{E}\big[\{\ell'(\lambda_n)\}^2\big]$$

Recall that

$$\ell'(\lambda_n) = \frac{y_n - b'(\lambda_n)}{\phi}$$

Differentiating once more, we get

$$\ell''(\lambda_n) = -\frac{b''(\lambda_n)}{\phi}$$

So this gives us

$$\begin{aligned}\mathbb{E}\left[\frac{b''(\lambda_n)}{\phi}\right] &= \mathbb{E}\left[\left(\frac{y_n - b'(\lambda_n)}{\phi}\right)^2\right] \\ &= \mathbb{E}\left[\frac{y_n^2 - 2y_n b'(\lambda_n) + \{b'(\lambda_n)\}^2}{\phi^2}\right]\end{aligned}$$

By linearity of expectation, we have

$$\frac{b''(\lambda_n)}{\phi} = \frac{\mathbb{E}\big[y_n^2|\lambda_n\big] - 2\mathbb{E}[y_n|\lambda_n]\,b'(\lambda_n) + \{b'(\lambda_n)\}^2}{\phi^2}$$

Now, using the fact that $\mathbb{E}[y_n|\lambda_n] = b'(\lambda_n)$, we get the middle term

$$2\mathbb{E}[y_n|\lambda_n]\,b'(\lambda_n) = \mathbb{E}[y_n|\lambda_n]^2 + \{b'(\lambda_n)\}^2$$

Substituting this in, we get

$$\begin{aligned}\frac{b''(\lambda_n)}{\phi} &= \frac{\mathbb{E}\left[y_n^2|\lambda_n\right] - \{\mathbb{E}[y_n|\lambda_n]\}^2 - \{b'(\lambda_n)^2\} + \{b'(\lambda_n)\}^2}{\phi^2} \\ &= \frac{\mathbb{E}\left[y_n^2|\lambda_n\right] - \{\mathbb{E}[y_n|\lambda_n]\}^2}{\phi^2} \\ &= \frac{\mathrm{Var}(y_n|\lambda_n)}{\phi^2}\end{aligned}$$

and so

$$\mathrm{Var}(y_n|\lambda_n) = \phi \cdot b''(\lambda_n) = \phi \cdot h'(\lambda_n)$$

(c) We have that

$$\begin{aligned}f(y_n|\lambda_n) &= \exp\left\{\frac{\lambda_n y_n - b(\lambda_n)}{\phi}\right\} \cdot c(y_n, \phi) \\ &= \exp\left\{\frac{\boldsymbol{x}_n'\boldsymbol{\theta}^* y_n - b(\boldsymbol{x}_n'\boldsymbol{\theta}^*)}{\phi}\right\} \cdot c(y_n, \phi)\end{aligned}$$

Taking logs, we get

$$\ell(\theta; y_n, \boldsymbol{x}_n) = \frac{\boldsymbol{x}_n'\boldsymbol{\theta}^* y_n - b(\boldsymbol{x}_n'\boldsymbol{\theta}^*)}{\phi} + \log c(y_n, \phi)$$

This gives

$$\begin{aligned}\nabla\ell(\theta; y_n, \boldsymbol{x}_n) &= \frac{y_n\boldsymbol{x}_n - b'(\boldsymbol{x}_n'\boldsymbol{\theta}^*)\boldsymbol{x}_n}{\phi} \\ &= \frac{1}{\phi}\left\{y_n - h(\boldsymbol{x}_n'\boldsymbol{\theta}^*)\right\}\boldsymbol{x}_n\end{aligned}$$

(d) By definition, we have

$$\left(\mathcal{I}(\boldsymbol{\theta})\right)_{i,j} = -\mathbb{E}\left[\frac{\partial^2}{\partial\boldsymbol{\theta}_i\partial\boldsymbol{\theta}_j}\ell(\boldsymbol{\theta}; y_n, \boldsymbol{x}_n)\right]$$

for each index i, j. It follows trivially that

$$\mathcal{I}(\boldsymbol{\theta}) = -\mathbb{E}[\nabla\nabla\ell(\boldsymbol{\theta}; y_n, \boldsymbol{x}_n)]$$

Next, we have

$$\begin{aligned}\nabla\nabla\ell(\boldsymbol{\theta}; y_n, \boldsymbol{x}_n) &= \nabla\left\{\frac{1}{\phi}(y_n - h(\boldsymbol{x}_n'\boldsymbol{\theta}^*)\boldsymbol{x}_n\right\} \\ &= -\frac{\left\{0 + h'(\boldsymbol{x}_n'\boldsymbol{\theta}^*)\right\}\boldsymbol{x}_n\boldsymbol{x}_n'}{\phi} \\ &= -\frac{h'(\boldsymbol{x}_n'\boldsymbol{\theta}^*)\boldsymbol{x}_n\boldsymbol{x}_n'}{\phi}\end{aligned}$$

Multiplying by -1 and taking expectations yields our result:

$$\mathcal{I}(\boldsymbol{\theta}) = -\mathbb{E}[\nabla\nabla\ell(\boldsymbol{\theta}; y_n, \boldsymbol{x}_n)] = \frac{1}{\phi}\mathbb{E}[h'(\boldsymbol{x}_n'\boldsymbol{\theta}^*)\boldsymbol{x}_n\boldsymbol{x}_n']$$

(e) We can show that $h(\cdot)$ is non-decreasing by looking at $\mathrm{Var}(y_n|\lambda_n)$. Recall that $\mathrm{Var}(y_n|\lambda_n) = \phi \cdot h'(\lambda_n)$. We are given that $\phi > 0$ and we know

$$\mathrm{Var}(y_n|\lambda_n = \phi \cdot h'(\lambda_n) \geq 0$$

since it is a variance. hence, we have $h'(\lambda_n) \geq 0$, which implies that $h(\cdot)$ is non-decreasing.

2. (a) The function we wish to minimize is $\mathbb{E}_x\left[(\theta - x)'A(\theta - x)\right]$.

- To do this for SGD, we can take iterates

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \gamma_t\nabla(\boldsymbol{\theta}_t - \boldsymbol{x}_t)'A(\boldsymbol{\theta}_t - \boldsymbol{x}_t) \\ &= \boldsymbol{\theta}_t - \gamma_t 2A(\boldsymbol{\theta}_t - \boldsymbol{x}_t)\end{aligned}$$

- In the case of ASGD, do the same iteration, but instead of taking $\boldsymbol{\theta}_{t+1}$, we take $\bar{\boldsymbol{\theta}}_{t+1} = \frac{1}{t+1}\sum_{j=1}^{t+1}\boldsymbol{\theta}_j$, as our result.
- For the implicit case, we need to solve $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \gamma_t 2A(\boldsymbol{\theta}_{t+1} - \boldsymbol{x}_t)$. Moving the last term on the right to the left hand side, we get

$$\boldsymbol{\theta}_{t+1} + \gamma_t 2A(\boldsymbol{\theta}_{t+1} - x_t) = \boldsymbol{\theta}_t$$

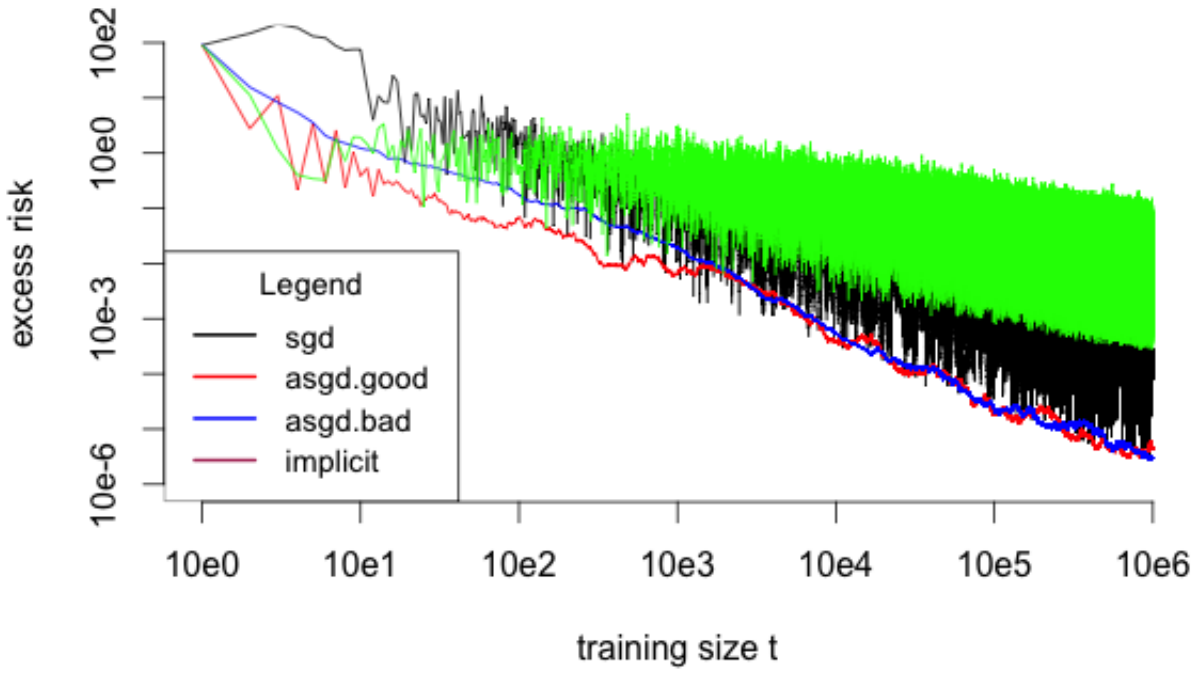
Then we can factor the $\boldsymbol{\theta}_{t+1}$ vector out.

$$(I + \gamma_t 2A)\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \gamma_t 2A\boldsymbol{x}_t$$

Finally, we multiply both sides by $(I + \gamma_t 2A)^{-1}$.

$$\boldsymbol{\theta}_{t+1} = (I + \gamma_t 2A)^{-1}(\boldsymbol{\theta}_t + \gamma_t 2A\boldsymbol{x}_t)$$

When we run our code, we get the follow plot



(b) We have that $y_n|\lambda_n \sim \mathcal{N}(\lambda_n, \sigma^2)$, $\lambda_n = \boldsymbol{x}_n'\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}^* \in \mathbb{R}^p$. For this part, we are required to find $\min_{\boldsymbol{\theta}}(y - \boldsymbol{x}_n'\boldsymbol{\theta})^2$ as this determines the best $\boldsymbol{\theta}$ parameter because this is a linear regression. We have that $\ell(\boldsymbol{\theta}; y_t, \boldsymbol{x}_t) = (y_t - \boldsymbol{x}_t'\boldsymbol{\theta})^2$.

- Our SGD is defined by

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \gamma_t \nabla (y_t - \boldsymbol{x}_t'\boldsymbol{\theta}_t)^2 \\ &= \boldsymbol{\theta}_t - \gamma_t 2(y_t - \boldsymbol{x}_t'\boldsymbol{\theta}_t)\boldsymbol{x}_t \\ &= \boldsymbol{\theta}_t + \alpha_t(y_t - \boldsymbol{x}_t'\boldsymbol{\theta}_t)\boldsymbol{x}_t\end{aligned}$$

  where $\alpha_t = -2\gamma_t$.
- As before, we can get the ASGD by doing a simple modification to this by simply take $\bar{\boldsymbol{\theta}}_{t+1}$ instead of $\boldsymbol{\theta}_{t+1}$,
- For the implicit case, we need to solve

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t y_t \boldsymbol{x}_t - \alpha_t \boldsymbol{x}_t'\boldsymbol{\theta}_{t+1}\boldsymbol{x}_t$$

  We move the $\alpha_t \boldsymbol{x}_t'\boldsymbol{\theta}_{t+1}\boldsymbol{x}_t$ term to the left hand side, which gives us

$$\boldsymbol{\theta}_{t+1} + \alpha_t \boldsymbol{x}_t'\boldsymbol{\theta}_{t+1}\boldsymbol{x}_t = \boldsymbol{\theta}_t + \alpha_t y_t \boldsymbol{x}_t$$

  Since $\boldsymbol{x}_t'\boldsymbol{\theta}_{t+1}$ is simply a scalar quantity, we can move it to give

$$\boldsymbol{\theta}_{t+1} + \alpha_t \boldsymbol{x}_t\boldsymbol{x}_t'\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t y_t \boldsymbol{x}_t$$

  Factoring out the $\boldsymbol{\theta}_{t+1}$ vector gives

$$(I + \alpha_t \boldsymbol{x}_t\boldsymbol{x}_t')\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t y_t \boldsymbol{x}_t$$

  Then multiplying both sides by $(I + \alpha_t \boldsymbol{x}_t\boldsymbol{x}_t')^{-1}$ gives

$$\boldsymbol{\theta}_{t+1} = (I + \alpha_t \boldsymbol{x}_t\boldsymbol{x}_t')^{-1}(\boldsymbol{\theta}_t + \alpha_t y_t \boldsymbol{x}_t)$$

  To get this inverse, we apply the Sherman-Morrison formula which states that

$$(A + uv')^{-1} = A^{-1} - \frac{A^{-1}uv'A^{-1}}{1 + v'A^{-1}u}$$

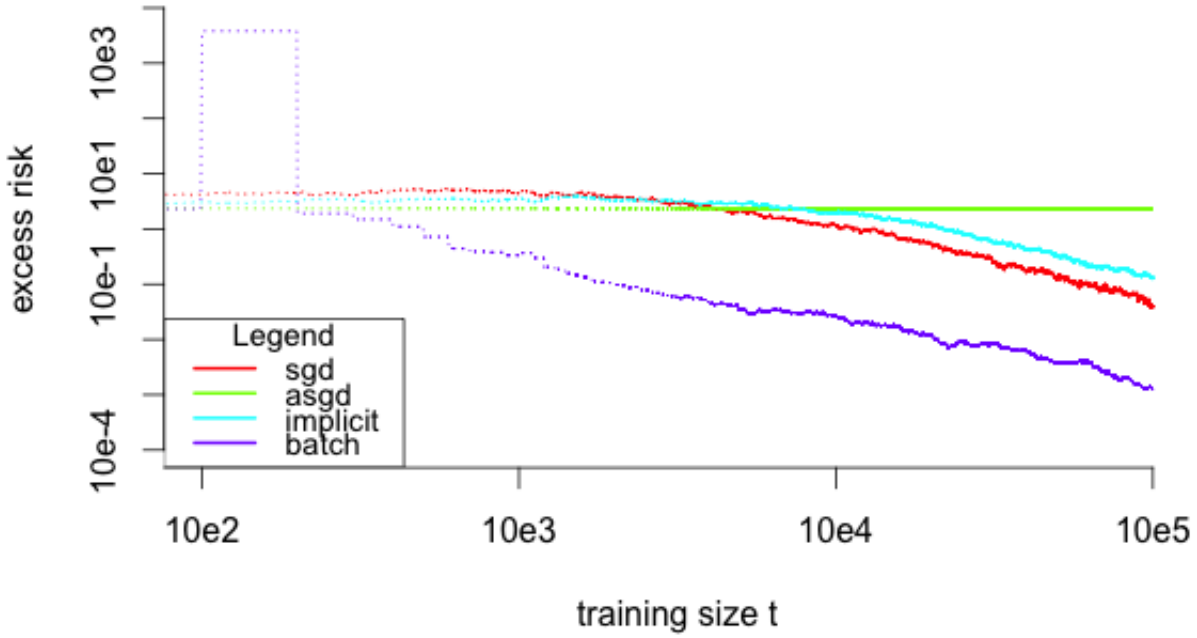
Now, we have

$$\begin{aligned}(I+\alpha_t \boldsymbol{x}_t \boldsymbol{x}_t')^{-1} &= \frac{1}{\alpha_t}\left(\frac{1}{\alpha_t} I + \boldsymbol{x}_t \boldsymbol{x}_t'\right)^{-1} \\ &= \frac{1}{\alpha_t}\left(\alpha_t I - \frac{\alpha_t I \boldsymbol{x}_t \boldsymbol{x}_t' \alpha_t I}{1+\boldsymbol{x}_t' \alpha_t I \boldsymbol{x}_t}\right) \qquad \text{by Sherman-Morrison} \\ &= \frac{1}{\alpha_t}\left(\alpha_t I - \frac{\alpha_t^2 \boldsymbol{x}_t \boldsymbol{x}_t'}{1+\alpha_t \boldsymbol{x}_t' \boldsymbol{x}_t}\right) \\ &= I - \frac{\alpha_t \boldsymbol{x}_t \boldsymbol{x}_t'}{1+\alpha_t \|\boldsymbol{x}_t\|^2}\end{aligned}$$

Using this, we have

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \left(I - \frac{\alpha_t \boldsymbol{x}_t \boldsymbol{x}_t'}{1+\alpha_t \|\boldsymbol{x}_t\|^2}\right)(\boldsymbol{\theta}_t + \alpha_t y_t \boldsymbol{x}_t) \\ &= \boldsymbol{\theta}_t + \alpha_t y_t \boldsymbol{x}_t - \frac{\alpha_t (\boldsymbol{x}_t' \boldsymbol{\theta}_t)\boldsymbol{x}_t}{1+\alpha_t \|\boldsymbol{x}_t\|^2} - \frac{\alpha_t^2 y_t \|\boldsymbol{x}_t\|^2 \boldsymbol{x}_t}{1+\alpha_t \|\boldsymbol{x}_t\|^2}\end{aligned}$$

Our plots are shown below:



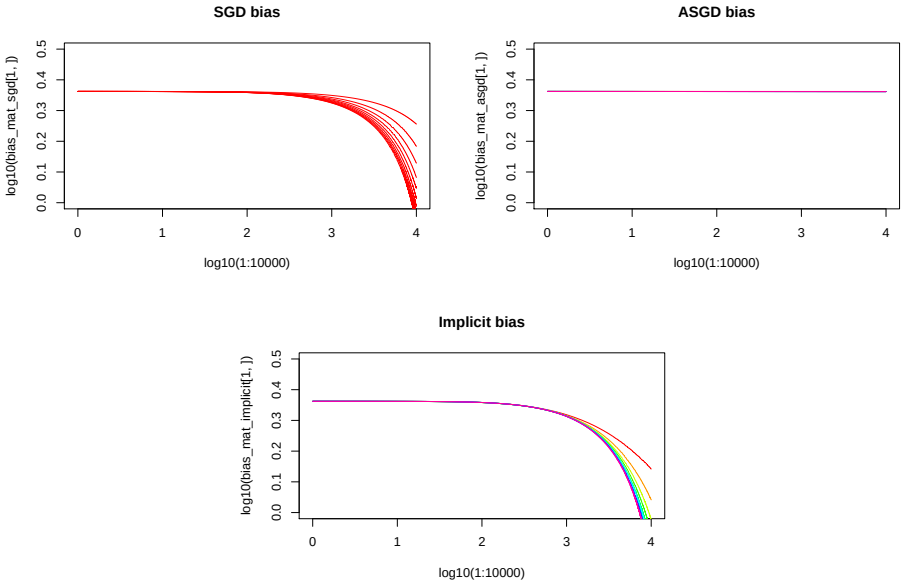
Note: I had some strange issues with the ASGD, which I couldn't figure out how to fix. The ASGD is a simple modification of the SGD. Once we calculate θ_{t+1} given θ_t, instead of returning θ_{t+1}, we return the average of $\theta_1, \ldots, \theta_{t+1}$. That is,

$$\theta_{t+1} = \frac{t\bar{\theta}_t + \theta_{t+1}}{t+1}$$

(c) The plots of the bias and the variances for the methods SGD, ASGD, and Implicit are as follows: For the bias term, I used the metric

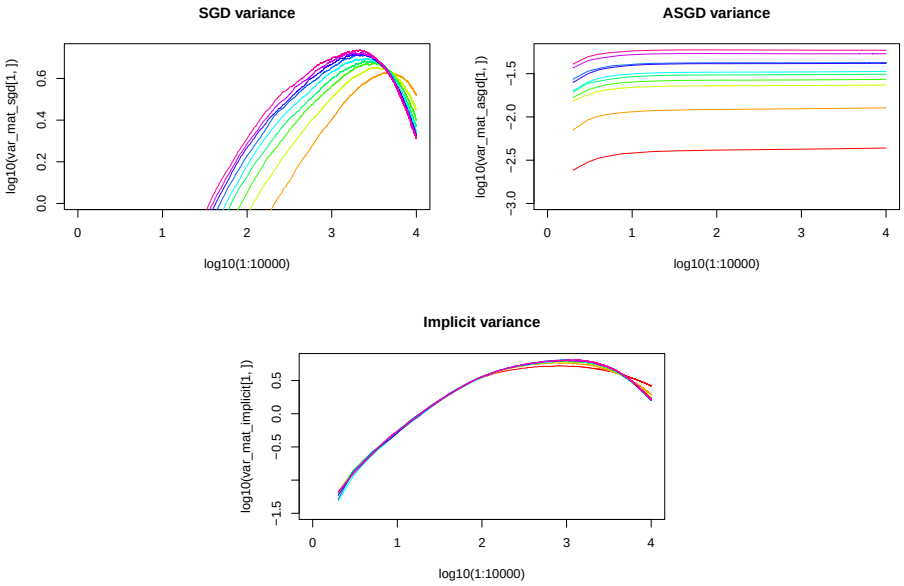
$$\|\theta_n - \theta^*\| = \log((\theta_n - \theta^*)'(\theta_n - \theta^*))$$

where $\theta* = \mathbf{1}$, that is, it is a vector of ones.



Again, I ran into issues with the ASGD. I ran this across 10^4 iterations locally and it had more risk that the other methods. I then tried to run it anyway for more iterations to see if it would lower in the long term, but for some reason something bizarre happened and it did not perform like it should.

(d) Our variance plots are as follows. In order to plot this, a one dimensional summary of each variance-covariance matrix was used. In this case, we used $\text{tr}(\text{Var}(\theta_n))$.



Again, we have the same caveat that something went wrong in the ASGD.

(e) We have that

$$y_n | x_n, \theta^* \sim \mathcal{N}(x_n' \theta^*, \sigma^2)$$

This has density

$$f_{Y_n}(y_n) = \frac{1}{\sqrt{2\pi\sigma^2}} e - \frac{(y_N - x_n'\theta^*)^2}{2\sigma^2}$$

Taking logs, we get

$$\log f_{Y_n}(y_n) = \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{(y_n - x_n'\theta^*)^2}{2\sigma^2}$$

Taking the partial derivative with respect to θ, we get

$$\begin{aligned}\frac{\partial}{\partial\theta}\log f_{Y_n}(y_n) &= \frac{2(y_n - x_n'\theta^*)x_n}{2\sigma^2} \\ &= \frac{(y_n - x_n'\theta^*)x_n}{\sigma^2}\end{aligned}$$

and so the Fisher Information matrix is

$$\begin{aligned}
\mathcal{I}(\theta^*) &= \mathbb{E}\left[\left(\frac{\partial}{\partial\theta}\log f_{Y_n}(y_n)\right)\left(\frac{\partial}{\partial\theta}\log f_{Y_n}(y_n)\right)'\right] \\
&= \mathbb{E}\left[\frac{(y_n - x_n'\theta)^2}{\sigma^4}x_n x_n'\right] \\
&= \frac{x_n x_n'}{\sigma^4}\mathbb{E}\big[(y_n - x_n'\theta)^2\big] \\
&= \frac{x_n x_n'}{\sigma^4}\sigma^2 \\
&= \frac{x_n x_n'}{\sigma^2}
\end{aligned}$$

3. (a) The elastic net is an algorithm which encompasses both lasso and ridge regression. It is a more general form which solves the problem

$$\min_{(\beta_0,\beta)\,\sin\,\mathbb{R}^{p+1}} R_\lambda(\beta_0,\beta) = \min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}}\left[\frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - x_i'\beta)^2 + \lambda P_\alpha(\beta)\right]$$

where

$$\begin{aligned}
P_\alpha(\beta) &= (1-\alpha)\frac{1}{2}\|\beta\|_{\ell_2}^2 + \alpha\|\beta\|_{\ell_1} \\
&= \sum_{j=1}^{p}\left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right]
\end{aligned}$$

The algorithm `glmnet` exploits sparcity through the use of P_α. This penalty term is a compromise between the ridge-regression penalty, when $\alpha = 0$, and the lasso penalty, when $\alpha = 1$. Increasing α increases the number of coefficients equal to zero, and hence the sparcity increases, making calculations run faster.

(b) Using Panos' `glmnet.R` code and then running

```
times = matrix(,nrow=0,ncol=6)
params = cbind(N=c(1000,5000,100,100,100,100), p=c(100,100,1000,5000,20000,50000))
for(i in 1:6)
{
  N = params[i,"N"]
  p = params[i,"p"]
  ans = run.glmnet(N, p,type="naive")
  ans2 = run.glmnet(N, p,type="covariance")
  times = rbind(times,tapply(ans[,"time"],c(rep(1,6),rep(2,6),rep(3,6)), mean))
  times = rbind(times,tapply(ans2[,"time"],c(rep(1,6),rep(2,6),rep(3,6)), mean))
}
times
```

we get the table:

Linear regression - Dense features

	Correlation 0	0.1	0.2	0.5	0.9	0.95
	$N = 1000, p = 100$					
`glmnet (type="naive")`	0.09383333	0.07866667	0.07500000	0.09383333	0.07866667	0.07500000
`glmnet (type="cov")`	0.01300000	0.01433333	0.01350000	0.01300000	0.01433333	0.01350000
	$N = 5000, p = 100$					
`glmnet (type="naive")`	0.28583333	0.32300000	0.34333333	0.28583333	0.32300000	0.34333333
`glmnet (type="cov")`	0.03900000	0.03833333	0.03633333	0.03900000	0.03833333	0.03633333
	$N = 100, p = 1000$					
`glmnet (type="naive")`	0.03366667	0.03633333	0.03300000	0.03366667	0.03633333	0.03300000
`glmnet (type="cov")`	0.07016667	0.07216667	0.06050000	0.07016667	0.07216667	0.06050000
	$N = 100, p = 5000$					
`glmnet (type="naive")`	0.14050000	0.09366667	0.09700000	0.14050000	0.09366667	0.09700000
`glmnet (type="cov")`	0.37366667	0.32083333	0.33233333	0.37366667	0.32083333	0.33233333
	$N = 100, p = 20000$					
`glmnet (type="naive")`	0.32933333	0.34200000	0.38166667	0.32933333	0.34200000	0.38166667
`glmnet (type="cov")`	1.49150000	1.43166667	1.45516667	1.49150000	1.43166667	1.45516667
	$N = 100, p = 50000$					
`glmnet (type="naive")`	0.74233333	0.76333333	0.88183333	0.74233333	0.76333333	0.88183333
`glmnet (type="cov")`	3.31166667	3.83350000	4.49833333	3.31166667	3.83350000	4.49833333

(c) After running the SGD for each set of N and p values, we get the following results.

Linear regression - Dense features

	Correlation 0	0.1	0.2	0.5	0.9	0.95
	$N = 1000, p = 100$					
`SGD`	0.31100000	0.2986667	0.30866667	0.30900000	0.31500000	0.33233333
	$N = 5000, p = 100$					
`SGD`	7.09300000	7.2303333	7.16200000	7.29366667	7.32633333	7.67033333
	$N = 100, p = 1000$					
`SGD`	0.03933333	0.0370000	0.03666667	0.03666667	0.04033333	0.03766667
	$N = 100, p = 5000$					
`SGD`	0.37600000	0.3633333	0.39733333	0.29366667	1.28133333	1.38033333
	$N = 100, p = 20000$					
`SGD`	2.45833333	1.8180000	1.51933333	1.52100000	1.56833333	1.58300000
	$N = 100, p = 50000$					
`SGD`	3.48600000	3.4553333	3.49833333	3.55466667	3.54500000	3.63233333

We can see that SGD is a lot slower than the naive case of `glmnet`. When $N = 100$, the cov case of `glmnet` runs quicker, but SGD is a slower when there are more iterations as shown in the $N = 1000$ and $N = 5000$ cases.

(d) When I run this part, I do a very slight modiciation fo part (c).

```
times = matrix(,nrow=0,ncol=6)

start_time = Sys.time()
task.id = as.numeric(Sys.getenv("SLURM_ARRAY_TASK_ID"))
job.id = as.numeric(Sys.getenv("SLURM_ARRAY_JOB_ID"))
```

```
print(paste("task.id ", task.id,"   ---  job.id  ",job.id))
N = 50000
p = 10e3
ans = run.glmnet(N, p)
times = rbind(times,tapply(ans[,"time"],c(rep(1,3),rep(2,3),rep(3,3),rep(4,3),rep(5,3),rep(6,3)), mean))
end_time = Sys.time()
total_time = as.numeric(end_time-start_time,units="mins")

save(list=c("times","start_time","end_time","total_time"),file=sprintf("odyssey/pset3/3d_job%d_task%d.rda", job.id, task.id))
```

Unfortunately, due to time constraints, my file wasn't able to complete. It had run at $N = 50000$ for over 9 hours. Every iteration increases the amount of computational time required exponentially in an SGD. As to expect, the times for each of the SGDs with different correlations are expected to be large.