Linear Regression Algorithms Variation Analysis

Mohamad Lakkis

March, 2024

This PDF is to explain the theoretical questions !

Problem 1

First, we need to address why we are centering the predictors what is the use of it in our proof?
And that is because the OLS estimates stay the same under centering (as shown in notes 2), the only thing that differs is the β_0, which in the case of centering we get $\hat{\beta}_0 = \bar{Y}$, as also shown in notes 2. (The idea of the proof simple is using to our advantage that $E(X_i) = 0$, when X_i is centered, and the assumption that $E(\epsilon) = 0$).
Thus we can drop W.L.O.G the β_0 term and then we can just consider $\hat{\beta}$ is p x 1, and the X is n x p, rather than n x (p+1).(since we don't need now to estimate the β_0, we know its estimation so it is just a constant)
And thus the RSS, becomes:

$$(Y - X\beta)^T(Y - X\beta) =$$

$$= (Y^T - \beta^T X^T)(Y - X\beta)$$

$$= Y^T Y - Y^T X\beta - \beta^T X^T Y + \beta X^T X\beta$$

Now, let's take $Y^T X\beta$ as A, notice that also $\beta^T X^T Y = A^T$. But notice that A is a scaler (i.e. 1 x 1 matrix)so $A^T = A$. And thus we get,

$$= Y^T Y - 2\beta^T X^T Y + \beta^T X^T X\beta$$

Now notice that $X^T Y = X^T X\hat{\beta}_{OLS}$(Assuming that X is invertible), and so we get,

$$= Y^T Y - 2\beta^T X^T X\hat{\beta}_{OLS} + \beta^T X^T X\beta \qquad (1)$$

Now let's go and try to develop the given equation:

$$(\beta - \hat{\beta}_{OLS})^T X^T X(\beta - \hat{\beta}_{OLS})$$

$$= \beta^T X^T X\beta - \beta^T X^T X\hat{\beta}_{OLS} - \hat{\beta}_{OLS}^T X^T X\beta + \hat{\beta}_{OLS}^T X^T X\hat{\beta}_{OLS}$$

Similarly as before, take $\beta^T X^T X \hat{\beta}_{OLS} = A$, and notice that A is a scalar so $A = A^T$, thus,

$$= \beta^T X^T X \beta - 2\beta^T X^T X \hat{\beta}_{OLS} + \hat{\beta}_{OLS}^T X^T X \hat{\beta}_{OLS} \qquad (2)$$

Now since the last term $\hat{\beta}_{OLS}^T X^T X \hat{\beta}_{OLS}$, is independent of β, we can remove it when finding the optimal β. Similarly, we can see that in eq (1), we have $Y^T Y$ which is also independent of β and we can remove it as well !!!

Thus, we can see that equation (1) is the same as (2), and we get finally,

$$RSS(\beta) \approx (\beta - \hat{\beta}_{OLS})^T X^T X (\beta - \hat{\beta}_{OLS})$$

By $\approx$ we mean that there is some additional constant (i.e. independent of β) terms as explained previously !

Meaning of this Result

This is an amazing result, this shows how the $RSS(\beta)$ deviates when the β moves away from the OLS estimates. Notice that this deviation is the function of ellipsoid centered at $\hat{\beta}_{OLS}$, and so when we add a penalty term (for example lasso or ridge), we will be taking the intersection of these ellipsoids with the constraints that we have, as shown in the figure 3.11 from the ESL book page 71, which I added it to this report. Now notice that the constraints of the ridge regression are of the form $\sum_{j=1}^{p} \beta_j^2 \leq t$, which is a sphere centered at 0. And as for the lasso constraints we have $\sum_{j=1}^{p} |\beta_j| \leq t$, which is a diamond shape, also centered at 0. And since we are taking the intersection of the ellipsoids with these constraints we see why the ridge and the lasso estimates pull the β towards 0. And because of the shape of these constraints we see why the lasso can have exactly 0 coefficients where ridge can't because of the sphere shape !

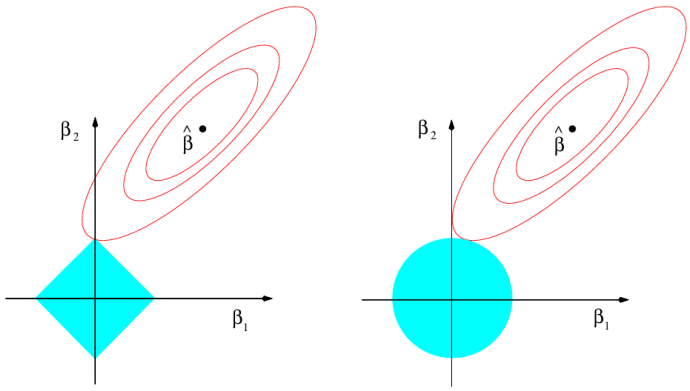


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Problem 2

We know that we have the $MSE = Bias^2 + Var$, let's now calculate each term!

Expectation

Let's start by computing the Expected of β^{RR} (since we know$\beta^{OLS} = \beta$)

$$E^{RR}[\hat{\beta}] = E[(X^T X + \lambda I_p)^{-1} X^T Y]$$

$$= E[(X^T X + \lambda I_p)^{-1} X^T (X\beta + \epsilon)]$$

$$= E[(X^T X + \lambda I_p)^{-1} X^T (X\beta)]$$

$$= (X^T X + \lambda I_p)^{-1} X^T X\beta$$

$$= \sum_{i=1}^{p} \frac{d_i^2}{d_i^2 + \lambda} \beta_i$$

So notice that, if $\lambda = 0$, this leads us to the OLS estimates which is unbiased we get $E(\hat{\beta}) = \beta$!

Variance

Now let's compute the variance with respect to OLS at a new point $\boldsymbol{x}$

$$Var^{OLS}(\hat{f}(x)|X = x)$$

$$= x^T Var(\hat{\beta}^{OLS}) x$$

$$= x^T Var((X^T X)^{-1} X^T Y) x$$

$$= x^T Var((X^T X)^{-1} X^T (X\beta + \epsilon)) x$$

$$= \sigma^2 x^T Var((X^T X)^{-1} X^T) x$$

$$= x^T x \sum_{i=1}^{p} \sigma^2 \frac{d_i^2}{d_i^4}$$

$$= x^T x \sum_{i=1}^{p} \sigma^2 \frac{1}{d_i^2}$$

Now let's compute the variance with respect to RR at a new point $\boldsymbol{x}$,

$$Var^{RR}(\hat{f}(x)|X = x)$$

$$= x^T Var(\hat{\beta}^{RR})x$$
$$= x^T Var((X^T X + \lambda I_p)^{-1} X^T Y)$$
$$= x^T Var((X^T X + \lambda I_p)^{-1} X^T (X\beta + \epsilon))$$
$$= x^T x \sum_{i=1}^{p} \sigma^2 \frac{d_i^2}{(d_i^2 + \lambda)^2}$$

Notice how if we plug into Var^{RR}, $\lambda = 0$, we get the same variance as in OLS (evident result since in RR ($\lambda = 0$) $\equiv$ OLS)

Bias

We know that the Bias of OLS estimates is 0.
Now let's compute the Bias of Ridge Regression, at a new point $\boldsymbol{x}$

$$Bias^{RR}(\hat{f}(x)|X = x) = E(\hat{f}(x)) - f(x)$$
$$= x^T E(\hat{\beta}) - x\beta$$
$$= x^T \sum_{i=1}^{p} \frac{d_i^2}{d_i^2 + \lambda} \beta_i - x\beta$$
$$= \sum_{i=1}^{p} \beta_i x_i (\frac{d_i^2}{d_i^2 + \lambda} - 1)$$

Now let's get the Bias squared,

$$Bias^2(\hat{f}(x)|X = x)$$
$$= \sum_{i=1}^{p} \beta_i^2 x_i^2 (\frac{d_i^2}{d_i^2 + \lambda} - 1)^2$$

And notice that here if we plug $\lambda = 0$, we will get the 0 $Bias$ which is the Bias of OLS

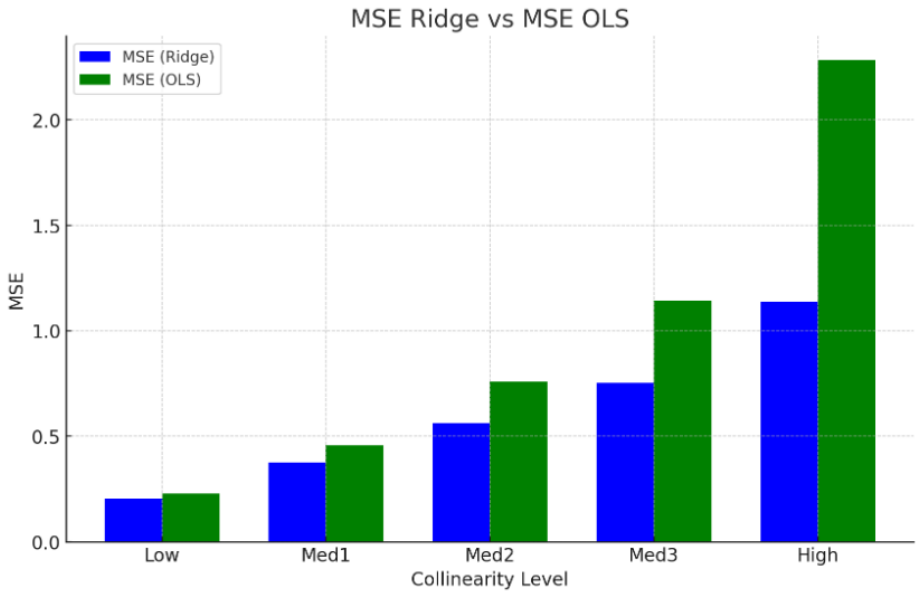
Interpretation

We need to understand two things, the first is how when we are increasing the bias a little we are seeing huge decrease in the variance in the case of the RR, the second is how the MSE will behave while we make the data more and more collinear. Let's draw some graphs to understand more (we will fix the new point x to ones for simplicity)!

Bias-Variance Tradeoff

- We can see that initially a small increase in the bias can lead to major decrease in the variance, but after a threshold it doesn't become that big of a change so we need to stop at some point when we find the minimum MSE, which we can estimate by the CV. Notice that this curve depends also on the collinearity of the data, so for each data we need to choose a different λ by the use of CV for example.

High Collinear Situation: OLS vs RR



- Notice how as the collinearity is getting larger and larger (d_i^2 getting smaller and smaller), the MSE for the RR model is better than the OLS, which is what we expect.
- Note that while computing the MSE^{RR}, we are not necessarily choosing the best λ value for each level of collinearity, we could have done so using cross validation, we are only interested in any λ that get us lower MSE^{RR} than MSE^{OLS}, since this proves the point that for high level of collinearity we can find a λ that leads to RR being better than OLS estimates !

Conclusion

Keep in mind that when the data becomes more and more collinear, the d_i^2, becomes smaller and smaller which will lead to a bigger variance in OLS, since it is divided by d_i^2 without a scaling factor, which makes it a lot more vulnerable compared to the $RidgeRegression$, which adds a penalty term before scaling the variance.
Lastly, under the assumptions of the Gauss-Markov theorem, and especially in the presence of very low collinearity, OLS is preferable from the blue theorem.

Problem 3

a)

We are trying to minimize with respect to β, the following

$$Ridge(\lambda) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

We can ignore β_0 and can expand the sums since there's only two terms.

$$Ridge(\lambda) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Let's focus now on the RSS terms since it will be common for both ridge regression and lasso !

$$RSS = \sum_{i=1}^{n} (y_i - \beta_1 x_{i1} - \beta_2 x_{i2})^2$$

$= \sum_{i=1}^{n} (y_i - x_{i1}(\beta_1 + \beta_2))^2, x_{i1} = x_{i2}$
$= \sum_{i=1}^{n} (y_i^2 + x_{i1}^2(\beta_1 + \beta_2)^2 - 2y_i x_{i1}(\beta_1 + \beta_2))$
$= y_1^2 + y_2^2 + x_{11}^2(\beta_1 + \beta_2)^2 + 2y_1 x_{11}(\beta_1 + \beta_2)$
$+ x_{21}^2(\beta_1 + \beta_2)^2 + 2y_2 x_{21}(\beta_1 + \beta_2), x_{21} = -x_{11}$
$= y_1^2 + y_2^2 + 2x_{11}^2(\beta_1^2 + \beta_2^2 + 2\beta_1\beta_2) + 2y_1 x_{11}(\beta_1 + \beta_2) + -2y_2 x_{11}(\beta_1 + \beta_2)$
$= y_1^2 + y_2^2 + 2x_{11}^2\beta_1^2 + 2x_{11}^2\beta_2^2 + 4x_{11}^2\beta_1\beta_2 + 2y_1 x_{11}\beta_1 + 2y_1 x_{11}\beta_2 - 2y_2 x_{11}\beta_1 - 2y_2 x_{11}\beta_2$

b)

And so if we take the partial derivative with respect to each β,

$$\frac{\partial Ridge(\lambda)}{\partial \beta_1} = 4x_{11}^2\beta_1 + 4x_{11}^2\beta_2 + 2y_1 x11 - 2y_2 x11 + 2\lambda\beta_1 \qquad (1)$$

$$\frac{\partial Ridge(\lambda)}{\partial \beta_2} = 4x_{11}^2\beta_2 + 4x_{11}^2\beta_1 + 2y_1 x11 - 2y_2 x11 + 2\lambda\beta_2 \qquad (2)$$

So we get a system of two equations,

$$(1) = (2)$$

$$2\lambda\beta_1 = 2\lambda\beta_2$$

Therefore, $\hat{\beta}_1^{RR} = \hat{\beta}_2^{RR}$

c)

We are trying to minimize with respect to β, the following

$$Lasso(\lambda) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Similarly, as before we have,

$$Lasso(\lambda) = y_1^2+y_2^2+2x_{11}^2\beta_1^2+2x_{11}^2\beta_2^2+4x_{11}^2\beta_1\beta_2+2y_1x_{11}\beta_1+2y_1x_{11}\beta_2-2y_2x_{11}\beta_1-2y_2x_{11}\beta_2+\lambda|\beta_1|+\lambda|\beta_2|$$

d)

And so if we take the partial derivative with respect to each β,

$$\frac{\partial Ridge(\lambda)}{\partial \beta_1} = 4x_{11}^2\beta_1 + 4x_{11}^2\beta_2 + 2y_1x11 - 2y_2x11 + \lambda \qquad (1)$$

$$\frac{\partial Ridge(\lambda)}{\partial \beta_2} = 4x_{11}^2\beta_2 + 4x_{11}^2\beta_1 + 2y_1x11 - 2y_2x11 + \lambda \qquad (2)$$

if we set (1) = (2), we get $0 = 0$, which shows us that we have many solutions for the lasso, (not like the ridge case).
In addition to that, notice that here we have the *exact* same *equation*, which is with two variables, and so this doesn't give necessity for β_1 to be equal to β_2, it can but this is not a requirement as in part b !

Problem 4

We know that from Problem 1, that $RSS(\beta) \approx (\beta - \hat{\beta}_{OLS})^T X^T X (\beta - \hat{\beta}_{OLS})$, and since in this case $X^T X = I_p$, we get $RSS \approx (\beta - \hat{\beta}_{OLS})^T (\beta - \hat{\beta}_{OLS})$. The function that we are trying to minimize in the case of lasso is the following, $J(\beta) = (\beta - \hat{\beta}_{OLS})^T (\beta - \hat{\beta}_{OLS}) + \lambda|\beta|$. And so, $J(\beta_i) = (\beta_i - \hat{\beta}_i^{OLS})^2 + \lambda|\beta_i|$, so we need to minimize each β_i (notes 1). Notice that in $J(\beta_i)$ all terms are positive except $-2\beta_i\hat{\beta}_i^{OLS}$, which could be positive or negative depending on the values of the betas and so we need to make sure that this term is always negative(or at least 0, in the case where $\hat{\beta}_i^{OLS} = 0$) to make the $J(\beta_i)$ smaller. Thus we have two cases,

$\hat{\beta}_i^{OLS} \geq 0$

We have in this case two options for the β_i, we need to choose the option that minimizes $J(\beta_i)$:

0.0.1 $\beta_i = 0$

$$J(\beta_i) = \hat{\beta}_{i,OLS}^2$$

0.0.2 $\beta_i > 0$

By taking the derivative with respect to β_i we get, $\beta_i = \hat{\beta}_{i,OLS} - \frac{\lambda}{2}$, and so by rescaling λ we get, $\beta_i = \hat{\beta}_{i,OLS} - \lambda$. And since $\beta_i > 0$, this means that $\hat{\beta}_{i,OLS} > \lambda$

Best option in this case?

In order to find the best option let's find the $J(\beta_i)$ for the second case (since the first case is already calculated)
So by plugging $\beta_i = \hat{\beta}_{i,OLS} - \lambda$ in the $J(\beta_i)$, we get with some basic simplifications:

$$J(\beta_i) = \lambda\hat{\beta}_{i,OLS}$$

But notice that since $\lambda < \hat{\beta}_{i,OLS}$ we have that $\lambda\hat{\beta}_{i,OLS} < \hat{\beta}^2_{i,OLS}$.
This means that when we have the option(i.e. $\hat{\beta}_{i,OLS} > \lambda$) to take $\beta_i = \hat{\beta}_{i,OLS} - \lambda$, we should take it since it will minimize the desired quantity !

And so we can write for this case,

$$\hat{\beta}_i^{lasso} = sign(\hat{\beta}_i^{OLS})max\{|\hat{\beta}_i^{OLS}| - \lambda, 0\} = sign(\hat{\beta}_i^{OLS})(|\hat{\beta}_i^{OLS}| - \lambda)_+$$

The max value indicates that when Option 2 is available (i.e. $\hat{\beta}_i^{OLS} > \lambda$) we should always take it, when this is not the case we should take it as zero

$\hat{\beta}_i^{OLS} < 0$

We will proceed similarly as the first case.
We have in this case two options for the β_i, we need to choose the option that minimizes $J(\beta_i)$:

0.0.3 $\beta_i = 0$

$$J(\beta_i) = \hat{\beta}^2_{i,OLS}$$

0.0.4 $\beta_i < 0$

By taking the derivative with respect to β_i (and by rescaling as in the first case), we get $\beta_i = \hat{\beta}_i^{OLS} + \lambda$. And since $\beta_i < 0$ we have that $-\hat{\beta}_i^{OLS} > \lambda$, so if that is the case we have this option available, but is it like in the first case preferable? Let's find out !

Best option in this case?

So similarly as before with some basic simplifications and by plugging $\beta_i = \hat{\beta}_i^{OLS} + \lambda$, we get

$$J(\beta_i) = -\hat{\beta}_i^{OLS}\lambda$$

And so since this option is only available when $-\hat{\beta}_i^{OLS} > \lambda$, we have $-\hat{\beta}_i^{OLS}\lambda < \hat{\beta}_{i,OLS}^2$. So when it is available we should take it. And so we can write this solution by:

$$\hat{\beta}_i^{lasso} = sign(\hat{\beta}_i^{OLS})max\{|\hat{\beta}_i^{OLS}| - \lambda, 0\} = sign(\hat{\beta}_i^{OLS})(|\hat{\beta}_i^{OLS}| - \lambda)_+$$

Conclusion

Thus, we can now say that in this orthogonal case we have

$$\hat{\beta}_i^{lasso} = sign(\hat{\beta}_i^{OLS})max\{|\hat{\beta}_i^{OLS}| - \lambda, 0\} = sign(\hat{\beta}_i^{OLS})(|\hat{\beta}_i^{OLS}| - \lambda)_+$$

Which show us why the *Lasso* coefficients shrink the OLS based on their absolute value, so if the coefficients is so small (i.e. $|\beta_i^{OLS}| \leq \lambda$) they will be shrunk to zero. This also shows why we will never get negative coefficients in the lasso case, and how exactly the coefficients are being shrunk! Very interesting !

Problem 5

Part a

We know that in the case of LDA, we model the class density as a multivariate gaussian distribution:

$$f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)}$$

But we know that in LDA we assume that $\Sigma_k = \Sigma, \forall k$:

$$f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)} \tag{1.1}$$

The probability that we are interested in calculating is the posterior probability, given by the bias theorem:

$$P(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{\ell=1}^K f_\ell(x)\pi_\ell}$$

And in our case we have only two classes so this reduces to

$$P(G = 2|X = x) = \frac{f_2(x)\pi_2}{f_2(x)\pi_2 + f_1(x)\pi_1}$$

$$P(G = 1|X = x) = \frac{f_1(x)\pi_1}{f_2(x)\pi_2 + f_1(x)\pi_1}$$

And thus, in order to know which class we need to classify is by looking at the log ratio, and it should be greater than zero in this case since we want to classify the observation as class 2

$$\log\frac{P(G = 2|X = x)}{P(G = 1|X = x)} > 0$$

Let's now try and simplify the ratio:

$$\log\frac{P(G=2|X=x)}{P(G=1|X=x)} = \log\frac{f_2(x)\pi_2}{f_1(x)\pi_1} = \log\frac{f_2(x)}{f_1(x)} + \log\frac{\pi_2}{\pi_1}$$

$$= \log(e^{-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)+\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)}) + \log\frac{\pi_2}{\pi_1}$$

Now using the fact that $\hat{\pi}_1 = \frac{N_1}{N}$ and $\hat{\pi}_2 = \frac{N_2}{N}$, we get:

$$= \log(e^{-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)+\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)}) + \log\frac{N_2}{N_1}$$

$$\approx \log\frac{N2}{N1} - \frac{1}{2}(\hat{\mu_2}+\hat{\mu}_1)^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu}_1) + x^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu_1})$$

And so by setting the following quantity > 0, we get the desired formula to classify an observation to class 2, given by:

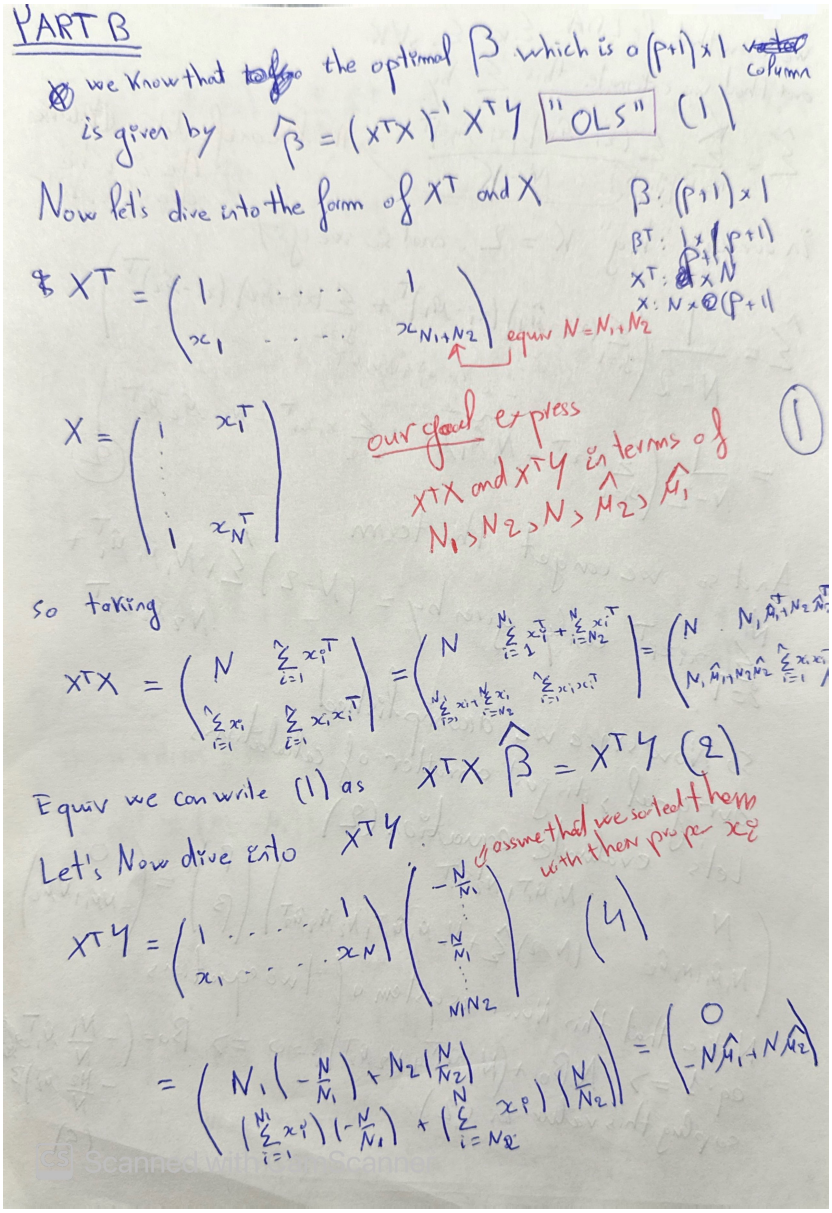
$$\log\frac{N2}{N1} - \frac{1}{2}(\hat{\mu_2}+\hat{\mu}_1)^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu}_1) + x^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu_1}) > 0$$

$$x^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu_1}) > \frac{1}{2}(\hat{\mu_2}+\hat{\mu}_1)^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu}_1) - \log\frac{N2}{N1}$$

And so if the classification criterion is the following:

$$\begin{cases} 2 & \text{if } x^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu_1}) > \frac{1}{2}(\hat{\mu_2}+\hat{\mu}_1)^T\hat{\Sigma}^{-1}(\hat{\mu}_2-\hat{\mu}_1) - \log\frac{N2}{N1} \\ 1 & \text{otherwise} \end{cases}$$

PARTS b,c,d,e


PART B
we know that the optimal β which is a (p+1)×1 column
is given by β̂ = (XᵀX)⁻¹ Xᵀy "OLS" (1)
Now let's dive into the form of Xᵀ and X
β: (p+1)×1
βᵀ: 1×(p+1)
Xᵀ: (p+1)×N
X: N×(p+1)
Xᵀ = (1 ... 1 ; x1 ... xN1+N2) equiv N = N1+N2
X = (1 x1ᵀ ; ... ; 1 xNᵀ)
our goal express XᵀX and Xᵀy in terms of N1, N2, N, μ̂2, μ̂1
(1)
So taking
XᵀX = (N Σ xiᵀ ; Σ xi Σ xixiᵀ) = (N N1μ̂1ᵀ + N2μ̂2ᵀ ; N1μ̂1 + N2μ̂2 Σ xixiᵀ)
Equiv we can write (1) as XᵀX β̂ = Xᵀy (2)
Let's Now dive into Xᵀy
assume that we sorted them with them pw per xi
Xᵀy = (1 ... 1 ; x1 ... xN) (−N/N1 ; ... ; −N/N1 ; ... ; N1N2) (4)
= (N1(−N/N1) + N2(N/N2) ; (Σ xi)(−N/N1) + (Σ xi)(N/N2)) = (0 ; −Nμ̂1 + Nμ̂2)

We know that for LDA $\Sigma_K = \Sigma, \forall K$
and that we estimate this Σ by

$$\hat{\Sigma} = \sum_{K=1}^{K} \frac{\sum_{g_i=K} (x_i - \hat{\mu}_K)(x_i - \hat{\mu}_K)^T}{N-K}$$

this like an if conditions to take the x_i whe class of x_i is K

In our case "big" $K = 2$, and so we get

$$\hat{\Sigma} = \frac{1}{N-2}\left(\sum_{g_i=1}(x_i - \hat{\mu}_1)(x_i - \hat{\mu}_1)^T + \sum_{g_i=2}(x_i - \hat{\mu}_2)(x_i - \hat{\mu}_2)^T\right)$$

$$= \frac{1}{N-2}\left(\sum_{g_i=1} x_i x_i^T - N_1 \hat{\mu}_1 \hat{\mu}_1^T + \sum_{g_i=2} x_i x_i^T - N_2 \hat{\mu}_2 \hat{\mu}_2^T\right) \quad (2)$$

$-2N\mu_1\mu_1^T + N\mu_1\mu_1^T$

And so we can get the term $\sum_{i=1}^{N} x_i x_i^T$ easily given by $= (N-2)\hat{\Sigma} + N_1 \hat{\mu}_1 \hat{\mu}_1^T + N_2 \hat{\mu}_2 \hat{\mu}_2^T$

So Now since we accomplished our goal, it is just a matter of calculations, Let's evaluate equation (2)

$$\begin{pmatrix} N & N_1\hat{\mu}_1^T + N_2\hat{\mu}_2^T \\ N_1\hat{\mu}_1 + N_2\hat{\mu}_2 & (N-2)\hat{\Sigma} + N_1\hat{\mu}_1\hat{\mu}_1^T + N_2\hat{\mu}_2\hat{\mu}_2^T \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ -N\hat{\mu}_1 + N\hat{\mu}_2 \end{pmatrix}$$

Notice that this Now is a system of two equations

eq: 1 $\Rightarrow N\beta_0 + (N_1\hat{\mu}_1^T + N_2\hat{\mu}_2^T)\beta = 0 \Rightarrow \beta_0 = \left(-\frac{N_1}{N}\hat{\mu}_1^T - \frac{N_2}{N}\hat{\mu}_2^T\right)\beta$ (5)

so plug this value in (2)

Thus,

$$(N_1\hat{\mu}_1 + N_2\hat{\mu}_2)\left(-\frac{N_1}{N}\hat{\mu}_1^T - \frac{N_2}{N}\hat{\mu}_2^T\right)\beta + \left((N-2)\hat{\Sigma} + N_1\hat{\mu}_1\hat{\mu}_1^T + N_2\hat{\mu}_2\hat{\mu}_2^T\right)\beta = N(\hat{\mu}_2 - \hat{\mu}_1)$$

with some long computations,

we get

$$\left((N-2)\hat{\Sigma} + \frac{N_1N_2}{N}\underbrace{(\hat{\mu}_2-\hat{\mu}_1)(\hat{\mu}_2-\hat{\mu}_1)^T}_{\hat{\Sigma}_B}\right)\beta = N(\hat{\mu}_2-\hat{\mu}_1)$$

Thus,

$$\left[(N-2)\hat{\Sigma} + \frac{N_1N_2}{N}\hat{\Sigma}_B\right]\beta = N(\hat{\mu}_2-\hat{\mu}_1) \quad (*)$$

PART C

This is a direct follow up from part B

$$\hat{\Sigma}_B\beta = (\mu_2-\mu_1)\underbrace{(\mu_2-\mu_1)^T}_{1\times p}\underbrace{\beta}_{p\times 1}$$

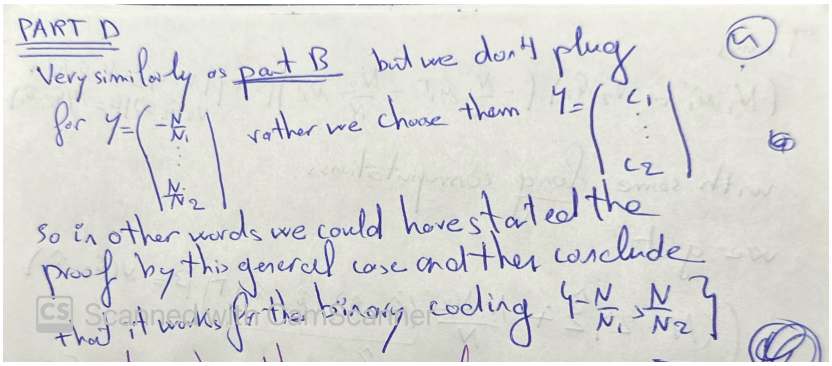
1×1 (i.e scalar)

so this means the vector direction of $\hat{\Sigma}_B\beta$ is $\mu_2-\mu_1$

and from $(*)$ we have $\frac{N_1N_2}{N}\hat{\Sigma}_B$ some size also $\frac{N_1N_2}{N}$ just a constant

and since $N(\mu_2-\mu_1)$ is also in the direction of $\mu_2-\mu_1$ Thus

and since $(N-2)\hat{\Sigma}$ is a constant given the training data, $\hat{\beta}$ is also in the direction of $\mu_2-\mu_1$ with $\beta \propto \Sigma^{-1}(\hat{\mu}_2-\hat{\mu}_1)$


PART D
Very similarly as part B but we don't plug
for $y=\begin{pmatrix} -\frac{N}{N_1} \\ \vdots \\ \frac{N}{N_2} \end{pmatrix}$ rather we choose them $y=\begin{pmatrix} c_1 \\ \vdots \\ c_2 \end{pmatrix}$
so in other words we could have stated the
proof by this general case and then concluded
that it works for the binary coding $y=\{\frac{N}{N_1}, \frac{N}{N_2}\}$

PART E (5)

* we have from part B eq 5

$$\hat{\beta}_0 = \left(-\frac{N_1}{N}\hat{\mu}_1^T - \frac{N_2}{N}\hat{\mu}_2^T\right)\hat{\beta}$$

and as a direct follow up

$$\hat{f}(x) = \hat{\beta}_0 + x^T\hat{\beta} = \left(-\frac{N_1}{N}\hat{\mu}_1^T - \frac{N_2}{N}\hat{\mu}_2^T\right)\hat{\beta} + x^T\hat{\beta}$$

$$= \left(-\frac{N_1}{N}\hat{\mu}_1^T - \frac{N_2}{N}\hat{\mu}_2^T + x^T\right)\hat{\beta}$$

* we know that $\hat{\beta} = C\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$ as shown in part C for some C. Assume $C > 0$ if $C < 0$ it is as choosing $\mathcal{L}$ instead of $\mathcal{L}$ but these are just dummy numbers.

$$\hat{f}(x) = \left(-\frac{N_1}{N}\hat{\mu}_1^T - \frac{N_2}{N}\hat{\mu}_2^T + x^T\right) C\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$$

Let's now evaluate the NEW classification rule $\begin{cases} 2 \text{ if } \hat{f}(x) > 0 \\ 1 \text{ otherwise} \end{cases}$

$\hat{f}(x) > 0$

~~Since C > 0 we can just remove it to obtain~~ since C is > 0 we can ignore it (6)

$$\left(-\frac{1}{N}\left(N_1\hat{\mu}_1^T + N_2\hat{\mu}_2^T\right) + x^T\right)\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > 0 \quad \text{(NEW)}$$

from <u>part a</u>

$$x^T\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2}(\hat{\mu}_2 + \hat{\mu}_1)^T\Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1) - \log\left(\frac{N_2}{N_1}\right) \quad \text{(LDA)}$$

In the case $N_2 = N_1$

we get

(LDA): $$x^T\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2}(\hat{\mu}_2 + \hat{\mu}_1)^T\Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$$

(NEW): $$\left(-\frac{1}{N}\left(N_1\hat{\mu}_1^T + N_1\hat{\mu}_2^T\right) + x^T\right)\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > 0$$

$$\left(-\frac{N_1}{N}\left(\hat{\mu}_1^T + \hat{\mu}_2^T\right) + x^T\right)\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > 0$$

but $N_1 = \frac{N}{2}$ since $N_1 + N_2 = N$ and $N_1 = N_2$

$$\left(-\frac{1}{2}\left(\hat{\mu}_1^T + \hat{\mu}_2^T\right) + x^T\right)\hat{\Sigma}^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > 0$$

$$-\frac{1}{2}\Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1) + x^T\Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > 0$$

$$x^T\Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2}\Sigma^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$$

equivalent to the LDA rule when $N_1 = N_2$ and $\neq$ otherwise

What does it mean when these two classification rules are not the same?

Problem 6

We will be using in the following the sigmoid function that maps any real number to [0,1]. Given by: $\sigma(\zeta) = \frac{1}{1+e^{-\zeta}}$, where $\zeta \in \mathbb{R}$
For simplicity we will use the Binary Class (0,1).
Logistic Regression models the probability that an observation X^* belongs to a class say "1". Defined as follows:

$$P(Y = 1|X) = \sigma(X\beta)$$

$$P(Y = 0|X) = 1 - \sigma(X\beta)$$

And thus, we find the MLE (which we can obtain from the binomial distribution when n = 1 i.e. thinking that each data point is one trial):

$$MLE(\beta) = \prod_{i=1}^{n} P(y_i|X_i;\beta)^{y_i}(1 - P(y_i|X_i;\beta))^{1-y_i}$$

$$MLE(\beta) = \prod_{i=1}^{n} \sigma(X_i\beta)^{y_i}(1 - \sigma(X_i\beta))^{1-y_i}$$

So to find the best β, we need to maximize the $MLE(\beta)$, or equivalently, minimizing the negative log-likelihood, which will make it easier on us to work with. And so we find:

$$\text{-log}(MLE(\beta)) = -\sum_{i=1}^{n}[y_i\text{log}(\sigma(X_i\beta)) + (1 - y_i)\text{log}(1 - \sigma(X_i\beta))]$$

And so to find our betas:

$$\hat{\beta} = \arg\min_{\beta} -(y\text{log}(\sigma(X\beta)) + (1 - y)\text{log}(1 - \sigma(X\beta)))$$

And so for each beta,

$$\hat{\beta} = \arg\min_{\beta} -(y_i\text{log}(\sigma(X_i\beta)) + (1 - y_i)\text{log}(1 - \sigma(X_i\beta)))$$

As mentioned before we will take $Y \sim Ber$, the same applies for any binary class.

Case 1: $y_i = 1$

we care now about minimizing the -log($\sigma(X_i\beta)$), and since the σ maps any value to a value between 0 and 1, and since we applied the log this means that we need to make $\sigma(X_i\beta)$ as close to one as possible(and we know that a β exists since

they are seperable so once we start with a β that does the job keep on increasing it to make the probability closer to one) (since any other value will result in a positive number (because of the -1)). Because if they weren't seprable it is not always a good option to send β to ∞ because this might lead to a decrease in the MLE (discussed at the end).
So simply said: When the data is linearly separable, there exists a β such that $sigma(X_i\beta)$ is very close to one $\forall i$ where $y_i = 1$. And so as β increases the probability become more and more closer to 1, which will result in the $MLE(\beta)$ becoming larger which is our goal. And so $\lim_{X_i\beta\to\infty} \sigma(X_i\beta) = 1$ and so $\lim_{X_i\beta\to\infty} \log(\sigma(X_i\beta)) = 0$.

Case 2: $y_i = 0$

Very similar analysis as in case 1 and thus we find: $\lim_{X_i\beta\to-\infty} \sigma(X_i\beta) = 1$ and so $\lim_{X_i\beta\to-\infty} \log(1 - \sigma(X_i\beta)) = 0$.

Conclusion

Thus, for the linearly separable case it is ALWAYS a better option to make $|\hat{\beta}| \to \infty$, because we won't have overlapping points on the sigmoid function(which if that is the case the choice of β is not always ∞). Which we will now show it by an example.

One question one might ask, why this process doesn't apply for non-separable data, can we just take β to ∞?

Well, for non-separable data, the logistic function allows for probabilities that are neither 0 nor 1(so probabilities in the middle). When data is not perfectly separable, in these cases, moving β towards infinity would actually decrease the likelihood because it would be assigning a higher probability to the wrong class. Thus, the maximizing process will find values of β that balance correctly classifying as many instances as possible with some misclassification errors.

For perfectly separable data, each class can be perfectly predicted with no errors. In this case, for $y_i = 1$, you can keep increasing β to make $\sigma(X_i\beta)$ even closer to 1, and for $y_i = 0$, you can make $\sigma(X_i\beta)$ even closer to 0 by making $X_i\beta$ more negative. Since there are no misclassified points to penalize the likelihood, the likelihood increases as $|\beta| \to \infty$.

We can see clearly, that if the class are separable making the sigmoid function very close to step function is the goal, since we won't have probabilities in the middle, because we are in the separable case(we can of course stretch it to make adaptable to observation which might have one or two crossing points, but this is not the case here), but making the non-separable case would result in lowering the MLE, because there no line that would separate the data perfectly !

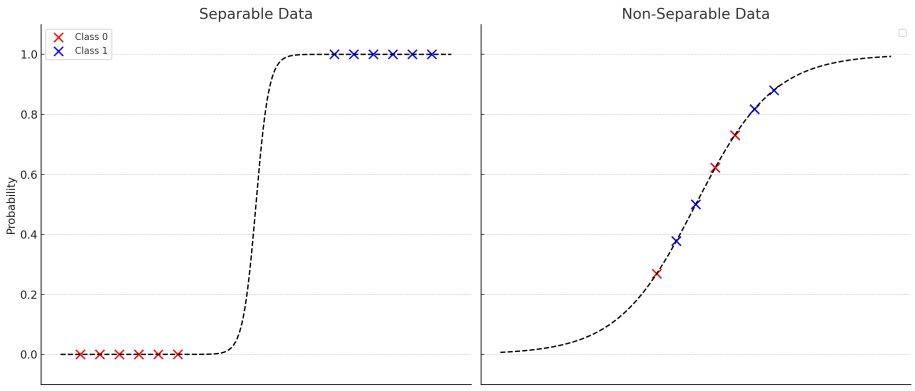


Figure 1: Comparison of the sigmoid function between separable and non-separable data

Problem 7

Part 1

We have that $Y|X \sim N(\mu, \sigma^2)$ and we also have that $\theta = m(f(x))$. But since in this part m is the identity function and $f(x) = E(Y|X = x)$, and so we get that $\mu(x) = f(x) = E(Y|X = x)$, so what we care about estimating is μ. So the idea of the MLE in general is to use many estimation of μ and use the one that result with the maximum likelihood. In this problem, we don't care about finding the best estimation for μ we just care about evaluating one estimate of it which we will call θ, so $Y|X \sim N(\theta(X), \sigma^2)$. Now let's start by calculating the log-likelihood of θ. One important note: θ is a function of x. So for each training $X = x_i$ we have a particular θ

$$\ell(\theta) = \sum_{i=1}^{n} \log(f_i(y_i|X; \theta))$$

Notice that here by f we are designating the pdf of $Y|X$, and NOT our function that we which to estimate.

$$\ell(\theta) = \sum_{i=1}^{n} \log(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - \theta(x_i))^2}{2\sigma^2}})$$

$$= \sum_{i=1}^{n} -\frac{(y_i - \theta(x_i))^2}{2\sigma^2} - \log(\sigma\sqrt{2\pi})$$

Now let's calculate the deviance:

$$D(y, f(x)) = -2\ell(\theta)$$

$$= \sum_{i=1}^{n} \frac{(y_i - \theta(x_i))^2}{\sigma^2} + 2\log(\sigma\sqrt{2\pi})$$

But notice that we don't care about the second term $2\log(\sigma\sqrt{2\pi})$, since we are calculating the likelihood of θ. (since the likelihood alone doesn't make sense unless compared with other likelihood of the estimator (i.e. it is relative, so for example if I get a value of 7 that doesn't mean that this function θ is a good estimator of the true function μ, we need to compare with the likelihood of other θ to do the analysis)), and similarly we can rescale it by multiplying by σ, thus we get

$$D(y, f(x)) = -2\ell(\theta) = \sum_{i=1}^{n} (y_i - \theta(x_i))^2$$

But notice that in this case $\theta(x) = f(x)$, and here by by this f we mean our desired function to be estimated.

$$D(y, f(x)) = \sum_{i=1}^{n} (y_i - f(x_i))^2 = L_2 loss$$

Part 2

Note that in the following when we say f_i, θ_i, p_i we mean respectively $f(x_i), \theta(x_i), p(x_i)$.
We have $Y|X \sim Ber(p(X))$ coded as $\{1, -1\}$, and so what we care is $\theta(x) = p(Y = 1|X = x) = \frac{1}{1+e^{-f(x)}}$, and so easily we can get $f(x) = \log(\frac{p_i}{1-p_i})$. Note that for each x_i we have $p(x_i)$, $f(x_i)$ and $\theta(x_i)$.
One additional note before starting the derivations, is that in this case θ is not the $E(Y|X = x)$ since $E(Y|X = x)$ in this case is $= p(Y = 1|X = x) - p(Y = -1|X = x)$, but in the case of 0-1 classes, the $E(Y|X = x) = p(Y = 1|X = x)$
Now let's us start deriving the $\ell(\theta)$:

$$\ell(\theta) = \sum_{i=1}^{n} \log(f_i(y_i|X; \theta))$$

And since we are dealing with an classes $\{1, -1\}$, we get:

$$\ell(\theta) = \sum_{i=1}^{n} \log(p_i^{\frac{y_i+1}{2}} (1 - p_i)^{\frac{1-y_i}{2}})$$

$$= \sum_{i=1}^{n} \frac{y_i + 1}{2} \log(p_i) + \frac{1 - y_i}{2} \log(1 - p_i)$$

$$= \sum_{i=1}^{n} -(\frac{y_i + 1}{2}) \log(1 + e^{-f_i}) - f_i \frac{1 - y_i}{2} - (\frac{1 - y_i}{2}) \log(1 + e^{-f_i})$$

Now let's multiply it by -2:

$$-2\ell(\theta) = \sum_{i=1}^{n}(y_i + 1)\log(1 + e^{-f_i}) + fi - y_i f_i + (1 - y_i)\log(1 + e^{-f_i})$$

$$= y_i\log(1 + e^{-f_i}) + \log(1 + e^{-f_i}) + f_i - y_i f_i + \log(1 + e^{-f_i}) - y_i\log(1 + e^{-f_i})$$

$$= \sum_{i=1}^{n} 2\log(1 + e^{-f_i}) + f_i - y_i f_i$$

Unfortunately I wasn't able to identify any loss out of this one from HW1. I actually re-calculated it many times to see if I did a calculation mistake, and each time I was getting the same answer.