CS 229: Machine Learning
Problem Set 1

William Ma

July 28, 2017

Question 1

1.a

We can calculate the Hessian, H, of the average empirical loss for logistic growth,

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\log\Big(1 + e^{y^{(i)}\theta^T x^{(i)}}\Big).$$

First, we let $h_\theta(x) = g(\theta^T x)$, where $g(z) = 1/(1+e^{-z})$, and calculate the partial derivative of h_θ.

$$\begin{aligned}
\frac{\partial h_\theta(x)}{\partial \theta_k} &= \frac{\partial}{\partial \theta_k}\log\Big(1 + e^{-x^T\theta}\Big) \\
&= \frac{1}{1 + e^{-yx^T\theta}}(-x^T e^{-x^T\theta}) \\
&= \frac{1}{1 + e^{yx^T\theta}}(-yx^T) \\
&= -h_\theta(-x^T)x_k.
\end{aligned}$$

With this, we can calculate the second partial derivative of each term.

$$\begin{aligned}
\frac{\partial^2 J(\theta)}{\partial\theta_k\partial\theta_l} &= \frac{\partial^2}{\partial\theta_k\partial\theta_l}\frac{1}{m}\sum_{i=1}^{m}\log\Big(1 + e^{y^{(i)}\theta^T x^{(i)}}\Big) \\
&= \frac{\partial}{\partial\theta_l}\frac{-1}{m}\sum_{i=1}^{m}h_\theta(-y^{(i)}x^{(i)})y^{(i)}x_k^{(i)} \\
&= \frac{1}{m}\sum_{i=1}^{m}h_\theta(-y^{(i)}x^{(i)})(1 - h_\theta(-y^{(i)}x^{(i)}))y^{(i)}x_k^{(i)}y^{(i)}x_l^{(i)}
\end{aligned}$$

Since $y^{(i)}$ is either 1 or -1, $(y^{(i)})^2 = 1$. Also, since

$$\begin{aligned}
h_\theta(-y^{(i)}x^{(i)})(1 - h_\theta(-y^{(i)}x^{(i)})) &= \frac{1}{1 + e^{-yx^T\theta^T}}\frac{e^{-yx^T\theta^T}}{1 + e^{-yx^T\theta^T}} \\
&= \frac{1}{1 + e^{-yx^T\theta^T}}\frac{1}{1 + e^{yx^T\theta^T}},
\end{aligned}$$

$h_\theta(-y^{(i)}x^{(i)})(1 - h_\theta(-y^{(i)}x^{(i)})) = h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))$. Thus,

$$\frac{\partial^2 J(\theta)}{\partial\theta_k\partial\theta_l} = \frac{1}{m}\sum_{i=1}^{m}h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))x_k^{(i)}x_l^{(i)}$$

Summing over k and l,

$$H = \nabla^2 J(\theta) = \frac{1}{m}\sum_{i=1}^{m}h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))x^{(i)}(x^{(i)})^T$$

To show that $H \in \mathbb{S}_{+}^{m \times m}$,

$$
\begin{aligned}
z^T H z &= \frac{1}{m} \sum_{i=1}^{m} h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})) z^T x^{(i)} (x^{(i)})^T z \\
&= \frac{1}{m} \sum_{i=1}^{m} h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))(z^T x^{(i)})^2
\end{aligned}
$$

Thus, $z^T H z \geq 0$, which implies that $H \in \mathbb{S}_{+}^{m \times m}$.

1.b

Using the following implementation of Newton's method in MATLAB,

```
close all; clear all; clc;

% Read in data
X = load('logistic_x.txt');
Y = load('logistic_y.txt');

% Prepare for fitting
X = [ones(size(X, 1), 1) X];
theta = log_reg(X ,Y, 20);

% Plot
figure; hold on;
plot(X(Y < 0, 2), X(Y < 0, 3), 'rx', 'linewidth', 2);
plot(X(Y > 0, 2), X(Y > 0, 3), 'go', 'linewidth', 2);
x1 = min(X(:,2)):.01:max(X(:,2));
x2 = -(theta(1) / theta(3)) - (theta(2) / theta(3)) * x1;
plot(x1,x2, 'linewidth', 2);
xlabel('x1');
ylabel('x2');


% Logistic regression fitting function
function f = log_reg(X, Y, maxiter)
m = size(X, 1);
n = size(X, 2);
theta = zeros(n, 1);

for i = 1 : maxiter
    expon = Y .* (X * thttps://www.overleaf.com/10385422rtmhvrzjdrzv#heta);
    h_theta = 1 ./ (1+exp(expon));
    grad = -(1/m) * (X' * (h_theta .* Y));
```

```
    H = (1/m) * (X' * diag(h_theta .* (1-h_theta)) * X);
    theta = theta - H \ grad;
end
f = theta;
end
```

we get $\theta = \begin{bmatrix} -2.62051159718020 \\ 0.760371535897677 \\ 1.17194674156714 \end{bmatrix}$.

1.c

The following is the plot of the training data and decision boundary from 1.b.

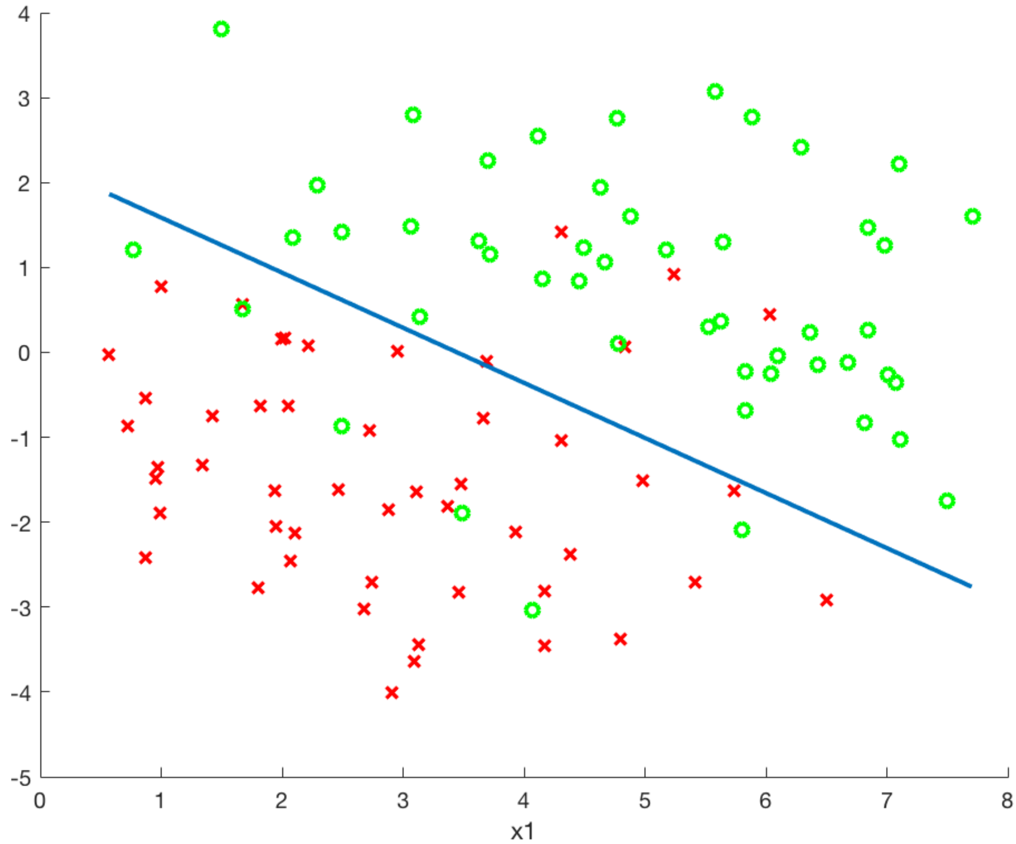

Figure 1: The green dots are where $y^{(i)} = 1$ and the red X's are where $y^{(i)} = -1$.

Question 2

2.a

We can demonstrate that the Poisson distribution is a member of the exponential family.

$$\begin{aligned} p(y;\lambda) &= \frac{e^{-\lambda}\lambda^y}{y!} \\ &= \frac{1}{y!}\exp\Big(\log e^{-\lambda}\lambda^y\Big) \\ &= \frac{1}{y!}\exp(y\log\lambda - \lambda) \end{aligned}$$

Thus, $b(y) = \frac{1}{y!}$, $T(y) = y$, $\eta = \log\lambda$, and $a(\eta) = \lambda$.

2.b

Since the Poisson distribution is in the exponential family, we can perform regression using GLM with it.

$$\begin{aligned} h_\theta(x) &= E[y|x;\theta] \\ &= \lambda \\ &= e^\eta \\ &= e^{\theta^T x} \end{aligned}$$

Thus, the canonical response function of the Poisson distribution is $h(x) = e^\eta = e^{\theta^T x}$.

2.c

Given a training set $\{(x^{(i)}, y^{(i)}); i = 1\ldots m\}$, we can calculate the stochastic gradient ascent for a GLM with Poisson response y and the canonical response function $h(x)$. First, we calculate the conditional probability

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{y^{(i)}!}\exp\Big(y^{(i)}\theta^T x^{(i)} - e^{\theta^T x^{(i)}}\Big).$$

Then, we can calculate the derivative of the log-likelihood with respect to θ_j.

$$\begin{aligned} \frac{\partial}{\partial\theta_j}\ell(\theta) &= \sum_{i=1}^{n}\log\left(\frac{1}{y^{(i)}}\exp\Big(y^{(i)}\theta^T x^{(i)} - e^{\theta^T x^{(i)}}\Big)\right) \\ &= \sum_{i=1}^{n} y^{(i)}x_j^{(i)} - x_j^{(i)}e^{\theta^T x^{(i)}} \end{aligned}$$

Thus, the stochastic gradient ascent update rule for a GLM with a Poisson response would be $\theta_j \coloneqq \theta_j - \alpha(h(x) - y)x_j$

2.d

Proof. Given a GLM with a response variable from any of the exponential family in which $T(y) = y$ and a canonical response $h(x)$, we have that $p(y;\eta) = b(y)\exp\big(\eta^T T(y) - a(\eta)\big)$, where $a(x) = h(x)$ due to the moment property of exponential family. First, we let $\eta_i = \theta^T x_i$ and calculate the derivative of the log-likelihood with respect to θ_i.

$$\begin{aligned}\frac{\partial}{\partial\theta_i}\ell(y|x;\theta) &= \frac{\partial}{\partial\theta_i} y\theta^T x - a(\theta^T x) + \log(b(y)) \\ &= yx_i - \frac{\partial a(\theta^T x)}{\partial\theta_i} \\ &= yx_i - h(x)x_i\end{aligned}$$

Thus, the stochastic gradient ascent update rule would be $\theta_i := \theta_i - \alpha(h(x) - y)x_i$. $\square$

Question 3

3.a

We consider the case $y = 1$ to calculate the posterior. Given the Bernoulli likelihood and multivariate Gaussian prior, we can calculate the posterior as follows.

$$\begin{aligned}p(y|x;\phi,\Sigma,\mu_1,\mu_{-1}) &= \frac{p(x|y)p(y)}{p(x)} \\ &= \frac{p(x|y=1)p(y)}{p(x|y=1)p(y=1) + p(x|y=-1)p(y=-1)}\end{aligned}$$

To simplify the algebra, we let $\sigma = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}$ and $\mu'_y = \frac{-1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)$.

$$\begin{aligned}p(y|x;\phi,\Sigma,\mu_1,\mu_{-1}) &= \frac{\sigma e^{\mu_1}\phi}{\sigma e^{\mu_1}\phi + \sigma e^{\mu_{-1}}(1-\phi)} \\ &= \frac{1}{1 + \frac{1-\phi}{\phi}e^{\mu_{-1}-\mu_1}} \\ &= \frac{1}{1 + \exp\Big(\log\frac{1-\phi}{\phi} - \frac{1}{2}(x-\mu_{-1})^T\Big)\Sigma^{-1}(x-\mu_{-1}) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)} \\ &= \frac{1}{1 + \exp\Big(\log\frac{1-\phi}{\phi} + \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 - \frac{1}{2}\mu_{-1}^T\Sigma^{-1}\mu_{-1} + (\mu_{-1}-\mu_1)^T\Sigma^{-1}x\Big)} \\ &= \frac{1}{1 + \exp(-y(\theta^T x + \theta_0))}\end{aligned}$$

where $\theta^T = (\mu_{-1}-\mu_1)^T\Sigma^{-1}x$ and $\theta_0 = \log\frac{1-\phi}{\phi} + \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 - \frac{1}{2}\mu_{-1}^T\Sigma^{-1}\mu_{-1}$.

3.b

See 3.c but reduced down to one dimension.

3.c

We can calculate the maximum likelihood estimates given the log-likelihood

$$\begin{aligned}
\ell(\phi, \Sigma, \mu_1, \mu_{-1}) &= \log \prod_{i=1}^{m} p(x^{(i)}|y^{(i)}; \Sigma, \mu_1, \mu_{-1}) p(y^{(i)}; \phi) \\
&= \sum_{i=1}^{m} \log p(x^{(i)}|y = y^{(i)}; \Sigma, \mu_1, \mu_{-1}) + \log p(y^{(i)}; \phi) \\
&= \sum_{i=1}^{m} - \log\Big((2\pi)^{n/2} |\Sigma|^{1/2}\Big) + \frac{-1}{2}(x^{(i)} - \mu_{y^{(i)}})^T \Sigma^{-1} (x^{(i)} - \mu_{y^{(i)}}) \\
&\qquad + 1\{y^{(i)} = 1\} \log \phi + (1 - 1\{y^{(i)} = 1\}) \log(1 - \phi)
\end{aligned}$$

To determine the maximum likelihood of each parameter, we take the derivative of each parameter individually and set the derivative equal to zero as follows.

$$\begin{aligned}
\frac{\partial \ell(\phi, \Sigma, \mu_1, \mu_{-1})}{\partial \phi} &= 0 \\
\sum_{i=1}^{m} \frac{1\{y^{(i)} = 1\}}{\phi} - \frac{1 - 1\{y^{(i)} = 1\}}{1 - \phi} &= 0 \\
\sum_{i=1}^{m} \frac{1\{y^{(i)} = 1\}}{\phi} &= \sum_{i=1}^{m} \frac{1 - 1\{y^{(i)} = 1\}}{1 - \phi} \\
(1 - \phi) \sum_{i=1}^{m} 1\{y^{(i)} = 1\} &= \phi \sum_{i=1}^{m} 1 - 1\{y^{(i)} = 1\} \\
\sum_{i=1}^{m} 1\{y^{(i)} = 1\} - \phi \sum_{i=1}^{m} 1\{y^{(i)} = 1\} &= m\phi - \phi \sum_{i=1}^{m} 1\{y^{(i)} = 1\} \\
\phi &= \frac{1}{m} \sum_{i=1}^{m} 1\{y^{(i)} = 1\}
\end{aligned}$$

Thus, the maximum likelihood for ϕ is given by $\phi = \frac{1}{m}\sum_{i=1}^{m}\{y^{(i)} = 1\}$.

$$\frac{\partial \ell(\phi, \Sigma, \mu_1, \mu_{-1})}{\partial \mu_{y^{(i)}}} = 0$$

$$\sum_{i=1}^{m} \frac{-1}{2}(-\Sigma^{-1}x^{(i)} - \Sigma^{-1}x^{(i)} + \Sigma^{-1}\mu_{y^{(i)}} + \Sigma^{-1}\mu_{y^{(i)}})1\{y^{(i)} = 1\} = 0$$

$$\sum_{i=1}^{m}(\Sigma^{-1}x^{(i)} - \Sigma^{-1}\mu_{y^{(i)}})1\{y^{(i)} = 1\} = 0$$

$$\sum_{i=1}^{m}(\Sigma^{-1}x^{(i)})1\{y^{(i)} = 1\} = \sum_{i=1}^{m}(\Sigma^{-1}\mu_{y^{(i)}})1\{y^{(i)} = 1\}$$

$$\mu_{y^{(i)}} = \frac{\sum_{i=1}^{m} x^{(i)}1\{y^{(i)} = 1\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

Thus, the maximum likelihood for $\mu_{y^{(i)}}$ is given by $\mu_{y^{(i)}} = \frac{\sum_{i=1}^{m} x^{(i)}1\{y^{(i)}=1\}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}}$.
To calculate the maximum likelihood for Σ, we let $S = \Sigma^{-1}$ to simplify the algebra.

$$\frac{\partial \ell(\phi, \Sigma, \mu_1, \mu_{-1})}{\partial S} = 0$$

$$\sum_{i=1}^{m} \frac{(2\pi)^{n/2}}{|S^{-1}|^{1/2}} \frac{1}{2(2\pi)^{n/2}|S^{-1}|^{1/2}} \nabla_S |S|$$

$$-\nabla_S \frac{-1}{2}(x^{(i)} - \mu_{y^{(i)}})^T S (x^{(i)} - \mu_{y^{(i)}}) = 0$$

$$\sum_{i=1}^{m} \frac{1}{2}(S^{-1})^T - \frac{-1}{2}(x^{(i)} - \mu_{y^{(i)}})^T(x^{(i)} - \mu_{y^{(i)}}) = 0$$

$$(S^{-1})^T m = \sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})^T(x^{(i)} - \mu_{y^{(i)}})$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})^T(x^{(i)} - \mu_{y^{(i)}})$$

Thus, the maximum likelihood for Σ is $\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})^T(x^{(i)} - \mu_{y^{(i)}})$.

Question 4

4.a

Proof. Given a matrix $A \in \mathbb{R}^{n \times n}$, vectors $x, z \in \mathbb{R}^n$, where $x = Az$ and $x^{(0)} = \vec{0}$, and the function $g(z) = f(Az)$, we first need to find the gradient and Hessian of $g(z)$.

$$\begin{aligned}
\nabla g(z) &= \sum_i \frac{\partial g(z)}{\partial z_i} \\
&= \sum_i \frac{\partial f(Az)}{\partial z_i} \\
&= \sum_i A_i \nabla f(Az) \\
&= A^T \nabla f(Az)
\end{aligned}$$

Thus, $\nabla g(z) = A^T \nabla f(Az)$.

$$\begin{aligned}
\nabla^2 g(z) &= \sum_i \sum j \frac{\partial^2 g(z)}{\partial z_i \partial z_j} \\
&= \sum_i \sum j \frac{\partial^2 f(Az)}{\partial z_i \partial z_j} \\
&= \sum_i \sum j \frac{\partial}{\partial z_j} A_i \nabla f(Az) \\
&= \sum_i \sum j A_i A_j \nabla^2 f(Az) \\
&= A^T A \nabla^2 f(Az)
\end{aligned}$$

Thus, $\nabla^2 g(z) = A^T A \nabla^2 f(Az)$.
We can now show that Newton's method is invariant to linear reparametrization as follows.

$$\begin{aligned}
z^{(i+1)} &:= z^{(i)} - (\nabla^2 g(z^{(i)}))^{-1} \bullet (\nabla g(z^{(i)})) \\
&= z^{(i)} - (A^T A \nabla^2 f(Az^{(i)}))^{-1} \bullet (A^T \nabla f(Az^{(i)})) \\
&= z^{(i)} - A^{-1} (\nabla^2 f(Az^{(i)}))^{-1} \bullet (\nabla f(Az^{(i)})) \\
Az^{(i+1)} &:= Az^{(i)} - (\nabla^2 f(Az^{(i)}))^{-1} \bullet (\nabla f(Az^{(i)})) \\
x^{(i+1)} &:= x^{(i)} - (\nabla^2 f(x^{(i)}))^{-1} \bullet (\nabla f(x^{(i)}))
\end{aligned}$$

Since, when $x^{(i)} = Az^{(i)}$, $x^{(i+1)} = Az^{(i+1)}$ and it is obvious that $x^{(0)} = \vec{0} = z^{(0)}$, Newton's method is invariant to linear reparametrization. $\square$

4.b

Using the same assumptions as in problem 4.a, we can show that gradient descent is not invariant to linear reparamtarization.

$$\begin{aligned}
z^{(i+1)} &:= z^{(i)} - \alpha \nabla g(z^{(i)}) \\
&= z^{(i)} - \alpha A^T \nabla f(Az^{(i)}) \\
(A^T)^{-1} z^{(i+1)} &:= (A^T)^{-1} z^{(i)} - \alpha f(Az^{(i)})
\end{aligned}$$

Since it is obvious that $x^{(i+1)} \neq (A^T)^{-1} z^{(i+1)}$, we are at an impasse. Thus, gradient descent is not invariant to linear reparametrization.

Question 5

Part a

5.a.i

Given that X, the matrix of input vectors, $\vec{y}$, the output vector, and W is a diagonal matrix with the diagonal elements are $\frac{1}{2} w^{(i)}$, where $w^{(i)}$ is the weight of the i-th element, are the proper dimensions,

$$\begin{aligned}
J(\theta) &= (X\theta - \vec{y})^T W (X\theta - \vec{y}) \\
&= \sum_i (x^{(i)}\theta - y^{(i)})^2 \frac{1}{2} w_{ii} \\
&= \frac{1}{2} \sum_i w^{(i)} (\theta^T x^{(i)} - y^{(i)})^2
\end{aligned}$$

Thus, $(X\theta - \vec{y})^T W (X\theta - \vec{y}) = \frac{1}{2} \sum_i w^{(i)} (\theta^T x^{(i)} - y^{(i)})^2$.

5.a.ii

We can also extend the normal equation to include the weight matrix W.

$$\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta} &= 0 \\
\frac{\partial}{\partial \theta} (X\theta - \vec{y})^T W (X\theta - \vec{y}) &= 0 \\
X^T W X \theta + (\theta^T X^T W X)^T - X^T W y^T - (yWX)^T &= 0 \\
X^T W X \theta + X^T W X \theta - X^T W y^T - X^T W y^T &= 0 \\
X^T W X \theta &= X^T W y^T \\
\theta &= (X^T W X)^{-1} X^T W y^T
\end{aligned}$$

Thus, the normal equation including the weights is $\theta = (X^T W X)^{-1} X^T W y^T$.

5.a.ii

For the training set $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$ and given that

$$p(y^{(i)}|x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma^{(i)}} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2(\sigma^{(i)})^2}\right),$$

the maximum likelihood is simply solving a weighted linear regression as follows.

$$\begin{aligned}
\arg\max_\theta \ell(\theta) &= \log \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \theta) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma^{(i)}} - \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2(\sigma^{(i)})^2} \\
&= \sum_{i=1}^{m} \frac{-1}{2(\sigma^{(i)})^2}(y^{(i)} - \theta^T x^{(i)})^2 \\
\arg\min_\theta \ell(\theta) &= \frac{1}{2}\sum_{i=1}^{m} \frac{1}{\sigma^2}(y^{(i)} - \theta^T x^{(i)})^2
\end{aligned}$$

Thus, fitting θ for a normally distributed set is essentially solving a weighted linear regression with $w^{(i)} = \frac{1}{(\sigma^{(i)})^2}$.

Part b

5.b.i

We can fit an unweighted least squares regression to the first training example using the following code.

```
% Load in data
run('load_quasar_data.m');

% Non-weighted model fitted with the first training example
X = lambdas;
Y = train_qso(1,:)';
theta = inv(X' * X) * X' * Y;

% Plot non-weighted model and corresponding points
figure; hold on;
plot(X, Y, 'go', 'linewidth', 2);
x1 = min(X):1:max(X);
x2 = theta * x1;
plot(x1, x2, 'linewidth', 2);
xlabel('lambda');
ylabel('flux');
```

With this code, we get the following plot.

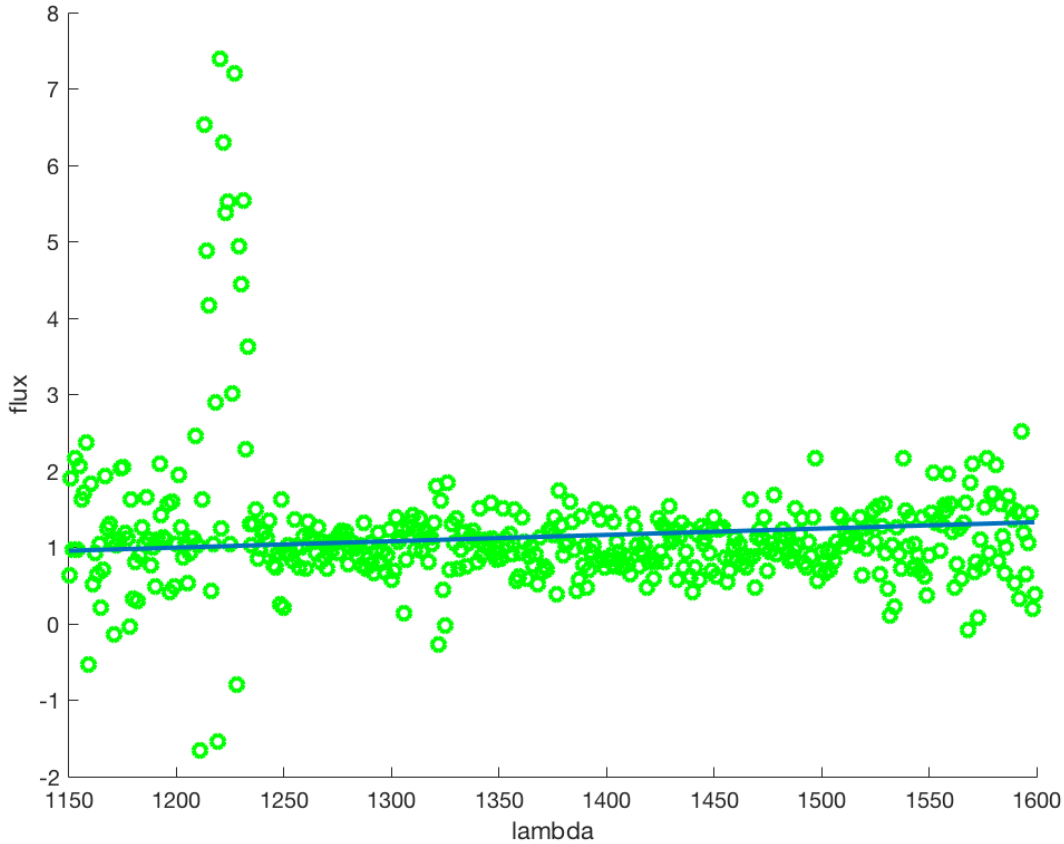

Figure 2: The green dots are where the true values and the line is the predicted values.

5.b.ii

We can also fit a weighted least squares regression to the first training example using the following code.

```
% Load in data
run('load_quasar_data.m');

% Weighted model fitted with the first training example
X = lambdas;
Y = train_qso(1,:)';
taus = [5];

% Make plot for weighted model
figure; hold on;
```

```
plot(lambdas, Y, 'kx', 'linewidth', 2);
x1 = min(lambdas):1:max(lambdas);
xlabel('lambda');
ylabel('flux');

% Fit weighted model for each value of tau
for tau = taus
    y = [];
    for xi = lambdas'
        w = exp(-(xi - X).^2/(2*tau^2));
        W = diag(w, 0);
        theta = inv(X' * W * X) * X' * W * Y;
        y = [y theta*xi];
    end
x2 = y';
plot(x1, x2, 'linewidth', 2);
end
```

With this code, we get the following plot.

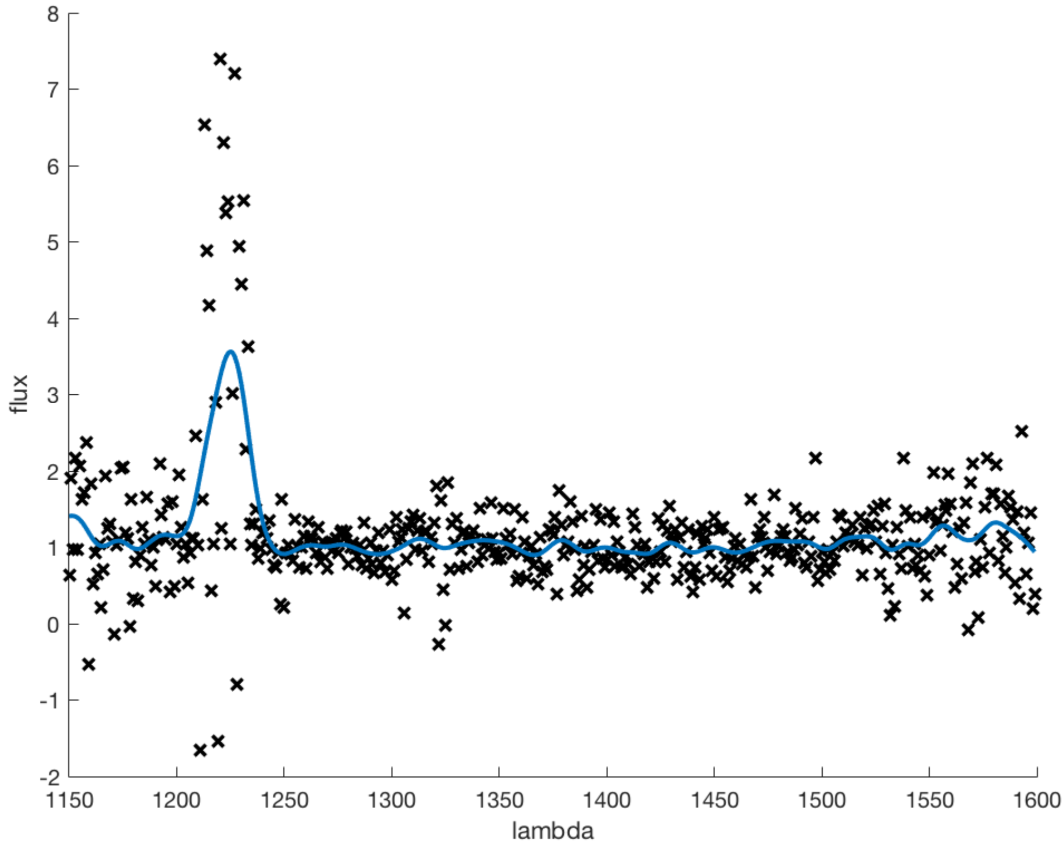

Figure 3: The black x's are where the true values and the line is the predicted values.

5.b.iii

We can also modify the code to be the following to explore the effect of τ.

```
% Load in data
run('load_quasar_data.m');

% Weighted model fitted with the first training example
X = lambdas;
Y = train_qso(1,:)';
% taus = [5];
taus = [1, 10, 100, 1000];

% Make plot for weighted model
figure; hold on;
```

```
plot(lambdas, Y, 'kx', 'linewidth', 2);
x1 = min(lambdas):1:max(lambdas);
xlabel('lambda');
ylabel('flux');

% Fit weighted model for each value of tau
for tau = taus
    y = [];
    for xi = lambdas'
        w = exp(-(xi - X).^2/(2*tau^2));
        W = diag(w, 0);
        theta = inv(X' * W * X) * X' * W * Y;
        y = [y theta*xi];
    end
x2 = y';
plot(x1, x2, 'linewidth', 2);
end
```

With this code, we get the following plot.

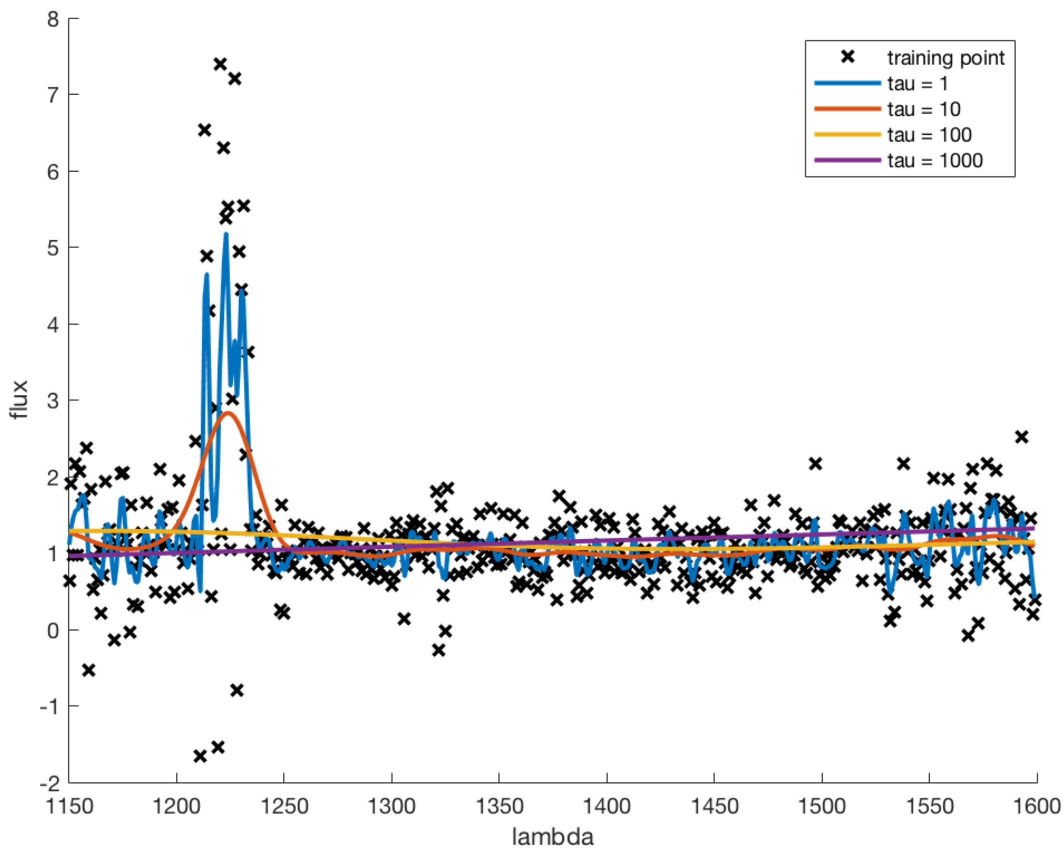

Figure 4: The black x's are where the true values and the lines are the different predicted values correlating to different values of τ.

We notice that the larger the value of τ, the "smoother" the line becomes. This is due to the fact that larger values of τ reduce the weight of points. Thus, the larger the τ the smaller effect the weights will have.