Julianne Lin

Math189R SP19
Homework 2
Monday, Feb 11, 2019

Please print out all the graphs generated by your own code and submit them together with the written part, and make sure you upload the code to your Github repository.

1 (Murphy 8.3) Gradient and Hessian of the log-likelihood for logistic regression.
(a) Let $\sigma(x) = \frac{1}{1+e^{-x}}$ be the sigmoid function. Show that

$$\sigma'(x) = \sigma(x)\left[1-\sigma(x)\right].$$

(b) Using the previous result and the chain rule of calculus, derive an expression for the gradient of the log likelihood for logistic regression.
(c) The Hessian can be written as $\mathbf{H} = \mathbf{X}^\top \mathbf{S}\mathbf{X}$ where $\mathbf{S} = \text{diag}(\mu_1(1-\mu_1), \ldots, \mu_n(1-\mu_n))$. Derive this and show that $\mathbf{H} \succeq 0$ ($A \succeq 0$ means that A is positive semidefinite).

Hint: Use the **negative** log-likelihood of logistic regression for this problem.

(a) $\sigma(x) = \frac{1}{1+e^{-x}} = (1+e^{-x})^{-1}$

$$\sigma'(x) = -\left(-\frac{e^{-x}}{(1+e^{-x})^2}\right) = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1-1+e^{-x}}{1+e^{-x}}\right)$$

$$= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{(1+e^{-x})-1}{1+e^{-x}}\right) = \left(\frac{1}{1+e^{-x}}\right)\left(1-\frac{1}{1+e^{-x}}\right) = \boxed{\sigma(x)\left[1-\sigma(x)\right]}$$

(b) From class, we know negative! (I checked the solution for this eq) log likelihood eq for logistic regression is:

$$n\ell(\theta) = -\sum_i y_i \log \sigma(\theta^T\vec{x}_i) + (1-y_i)\log(1-\sigma(\theta^T\vec{x}_i))$$

with the sigmoid fxn already plugged in

We take the gradient of this with respect to θ:

from $\frac{d}{dx}\log(x) = \frac{x'}{x}$

$$\nabla_\theta n\ell(\theta) = -\sum_i y_i \frac{1}{\sigma(\theta^T\vec{x}_i)}\sigma'(\theta^T\vec{x}_i) + (1-y_i)\frac{1}{1-\sigma(\theta^T\vec{x}_i)}(-\sigma'(\theta^T\vec{x}_i))$$

$$= -\sum_i y_i \frac{1}{\sigma(\theta^T\vec{x}_i)}\left(\sigma(\theta^T\vec{x}_i)\left[1-\sigma(\theta^T\vec{x}_i)\right]\right)\vec{x}_i + (1-y_i)\frac{1}{1-\sigma(\theta^T\vec{x}_i)}\left(-\sigma(\theta^T\vec{x}_i)\left[1-\sigma(\theta^T\vec{x}_i)\right]\right)\vec{x}_i$$

$$= -\sum_i y_i(1-\sigma(\theta^T\vec{x}_i))\vec{x}_i + (1-y_i)(-\sigma(\theta^T\vec{x}_i))\vec{x}_i$$

$$= -\sum_i y_i\vec{x}_i - \cancel{y_i\sigma(\theta^T\vec{x}_i)\vec{x}_i} - \sigma(\theta^T\vec{x}_i)\vec{x}_i + \cancel{y_i\sigma(\theta^T\vec{x}_i)\vec{x}_i}$$

$$= -\sum_i \vec{x}_i(y_i - \sigma(\theta^T\vec{x}_i))$$

$$= \sum_i (\sigma(\theta^T\vec{x}_i) - y_i)\vec{x}_i$$

because we know that $\mu_i = \sigma(\theta^T\vec{x}_i)$ ■

* We also know $\vec{x}_i$ is the transpose of the ith row in our design matrix X (ith column of X^T)

$$= \sum_i (\mu_i - y_i)\vec{x}_i$$

$$= \boxed{X^T(\vec{\mu}-\vec{y})}$$

1c) By definition of Hessian from multivar., $H_\theta = \nabla_\theta(\nabla_\theta n\ell(\theta))^T$ since it's the sq. matrix of 2. partial derivatives

$$H_\theta = \nabla_\theta(\nabla_\theta n\ell(\theta))^T = \nabla_\theta[X^T(\vec{\mu}-\vec{y})]^T$$

$$= \nabla_\theta(\vec{\mu}^T X - \vec{y}^T X)$$

Note that we can drop the $\vec{y}$ because we're taking the gradient w/respect to θ & $\vec{y}$ has no θ dependence

$$= \nabla_\theta(\vec{\mu}^T X)$$

$$= \nabla_\theta\, \sigma(X\theta)^T X$$

$$= X^T \operatorname{diag}(\vec{\mu}(1-\vec{\mu}))X$$

I checked the solution here when I got stuck.

$$= X^T S X$$

We can see $S = \operatorname{diag}(\vec{\mu}(1-\vec{\mu})) = \operatorname{diag}(\mu_1(1-\mu_1), \dots, \mu_n(1-\mu_n))$.

To show $H_\theta \succeq 0$, we can just show $S \succeq 0$, aka. show S is positive semi-definite.

By def of positive semidefinite, we need to show S is a symmetric matrix w/non-negative eigenvals

By def of a diagonal matrix (S), its eigenvalues are its diagonal entries.

Thus we need to show that

$$\mu_i(1-\mu_i) \geq 0.$$

So: $\mu_i(1-\mu_i) = \sigma(\theta^T x_i)(1-\sigma(\theta^T x_i))$. By definition of the sigmoid fxn, for any $\theta^T x_i$, $0 < \sigma(\theta^T x_i) < 1$. Thus $0 < (1-\sigma(\theta^T x_i)) < 1$.

Thus it must be true that $\sigma(\theta^T x_i)(1-\sigma(\theta^T x_i)) \geq 0$.

Thus $\mu_i(1-\mu_i) \geq 0$.

We've shown then that S is positive semi-definite, so $H_\theta \succeq 0$. ∎

2 (**Murphy 2.11**) Derive the normalization constant (Z) for a one dimensional zero-mean Gaussian

$$\mathbb{P}(x;\sigma^2) = \frac{1}{Z}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

such that $\mathbb{P}(x;\sigma^2)$ becomes a valid density.

By definition, total probability is 1. Thus,

$$\int_{\mathbb{R}} \mathbb{P}(x;\sigma^2)\,dx = 1 \Rightarrow \int_{\mathbb{R}} \mathbb{P}(x;\sigma^2) = \int_{\mathbb{R}} \frac{1}{Z}\exp\left(-\frac{x^2}{2\sigma^2}\right)dx = \frac{1}{Z}\int_{\mathbb{R}} \exp\left(\frac{-x^2}{2\sigma^2}\right)dx = 1$$

Thus we have $Z = \int_{\mathbb{R}} \exp\left(-\frac{x^2}{2\sigma^2}\right)dx$. We know we want to derive $Z = \sqrt{2\pi\sigma^2}$.
$\left(Z^2 = 2\pi\sigma^2 \right)$

consider, $Z^2 = \int_{\mathbb{R}} \exp\left(-\frac{x^2}{2\sigma^2}\right)dx \int_{\mathbb{R}} \exp\left(-\frac{y^2}{2\sigma^2}\right)dy$ (* ... checked w/ Wolfram alpha)

$= \iint_{\mathbb{R}^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)dx\,dy$ "1st" has no y dep & vice versa

polar coord

$= \int_0^\infty \int_0^{2\pi} \exp\left(-\frac{r^2}{2\sigma^2}\right) r\,d\theta\,dr$

$= \int_0^\infty \left(\exp\left(-\frac{r^2}{2\sigma^2}\right)\theta r\right)\Big|_0^{2\pi} dr$

$= 2\pi \int_0^\infty \exp\left(-\frac{r^2}{2\sigma^2}\right) r\,dr$

$u = -\frac{r^2}{2\sigma^2}$

$du = -\frac{r}{\sigma^2}dr$

$= 2\pi(-\sigma^2)\int_0^\infty \exp\left(-\frac{r^2}{2\sigma^2}\right)\left(-\frac{r}{\sigma^2}\right)dr$

$= 2\pi(-\sigma^2)\left(\exp\left(-\frac{r^2}{2\sigma^2}\right)\right)\Big|_0^\infty$

$= -2\pi\sigma^2 \exp\left(-\frac{r^2}{2\sigma^2}\right)\Big|_0^\infty$

$= -2\pi\sigma^2\left(e^{-(\infty)} - e^{-(0)}\right)$

$= -2\pi\sigma^2(0-1)$

$= 2\pi\sigma^2$

Thus $Z^2 = 2\pi\sigma^2$ and $Z = \sqrt{2\pi\sigma^2}$.

Thus, we've derived Z for a one-dim. zero-mean Gaussian. ∎

3 (continued)

(d) **(math)** Consider regularized linear regression where we pull the basis term out of the feature vectors. That is, instead of computing $\hat{\mathbf{y}} = \boldsymbol{\theta}^\top \mathbf{x}$ with $\mathbf{x}_0 = 1$, we compute $\hat{\mathbf{y}} = \boldsymbol{\theta}^\top \mathbf{x} + b$. This corresponds to solving the optimization problem

$$\text{minimize: } ||A\mathbf{x} + b\mathbf{1} - \mathbf{y}||_2^2 + ||\Gamma \mathbf{x}||_2^2$$

Euclidean norm

Solve for the optimal $\mathbf{x}^*$ explicitly. Use this close form to compute the bias term for the previous problem (with the same regularization strategy). Make sure it is the same.

(e) **(implementation)** We can also compute the solution to the least squares problem using gradient descent. Consider the same bias-relocated objective

$$\text{minimize: } f = ||A\mathbf{x} + b\mathbf{1} - \mathbf{y}||_2^2 + ||\Gamma \mathbf{x}||_2^2.$$

Compute the gradients and run gradient descent. Plot the ℓ_2 norm between the optimal $(\mathbf{x}^*, b^*)$ vector you computed in closed form and the iterates generated by gradient descent. Hint: your plot should move down and to the left and approach zero as the number of iterations increases. If it doesn't, try decreasing the learning rate.

(a) Recall that $\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$. We're given:

$$\underset{\vec{w}}{\arg\max} \sum_{i=1}^{N} \log \mathcal{N}(y_i | w_0 + \vec{w}^T\vec{x}_i, \sigma^2) + \sum_{j=1}^{D} \log \mathcal{N}(w_j | 0, \tau^2)$$

apply $\mathcal{N}(x|\mu,\sigma)$

$$= \underset{\vec{w}}{\arg\max} \sum_{i=1}^{N} \log\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i - w_0 - \vec{w}^T\vec{x}_i)^2}{2\sigma^2}\right)\right] + \sum_{j=1}^{D}\log\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{w_j^2}{2\tau^2}\right)\right]$$

log rule $\log(xy) = \log x + \log y$

$$= \underset{\vec{w}}{\arg\max} \sum_{i=1}^{N} \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \log\left(\exp\left(-\frac{(y_i - w_0 - \vec{w}^T\vec{x}_i)^2}{2\sigma^2}\right)\right) + \sum_{j=1}^{D}\log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \log\left(\exp\left(-\frac{w_j^2}{2\tau^2}\right)\right)$$

$$= \underset{\vec{w}}{\arg\max} \sum_{i=1}^{N} -\log(\sqrt{2\pi}\sigma) - \frac{(y_i - w_0 - \vec{w}^T\vec{x}_i)^2}{2\sigma^2} + \sum_{j=1}^{D} -\log(\sqrt{2\pi}\sigma) - \left(\frac{w_j^2}{2\tau^2}\right)$$

$$= \underset{\vec{w}}{\arg\max} -\left((N+D)\log\sqrt{2\pi}\sigma + \sum_{i=1}^{N}\frac{(y_i - w_0 - \vec{w}^T\vec{x}_i)^2}{2\sigma^2} + \sum_{j=1}^{D}\frac{w_j^2}{2\tau^2}\right)$$

Note that $(N+D)\log\sqrt{2\pi}\sigma$ & $\frac{1}{2\sigma^2}$ are constants that don't affect $\vec{w}$

$$= \underset{\vec{w}}{\arg\max} -\left(\sum_{i=1}^{N}(y_i - w_0 - \vec{w}^T\vec{x}_i)^2 + \sum_{j=1}^{D}\frac{w_j^2}{2\tau^2}\right)$$

max of fxn equals min of neg fxn!

$$= \underset{\vec{w}}{\arg\min}\left(\sum_{i=1}^{N}(y_i - w_0 - \vec{w}^T\vec{x}_i)^2 + 2\sigma^2\sum_{j=1}^{D}\frac{w_j^2}{2\tau^2}\right)$$

$$= \underset{\vec{w}}{\arg\min}\left(\sum_{i=1}^{N}(y_i - w_0 - \vec{w}^T\vec{x}_i)^2 + \frac{\sigma^2}{\tau^2}\sum_{j=1}^{D} w_j^2\right)$$

I think I'm off by $\frac{1}{N}$!

Since we've defined: $\lambda = \frac{\sigma^2}{\tau^2}$, so

$$= \underset{\vec{w}}{\arg\min}\left(\sum_{i=1}^{N}(y_i - w_0 - \vec{w}^T\vec{x}_i)^2 + \lambda\sum_{j=1}^{D} w_j^2\right) = \underset{\vec{w}}{\arg\min} \sum_{i=1}^{N}(y_i - (w_0 + \vec{w}^T\vec{x}_i))^2 + \lambda||\vec{w}||_2^2$$

■

(b) Given $f = \|A\vec{x} - \vec{b}\|_2^2 + \|\Gamma \vec{x}\|_2^2$ → minimize f. (Set deriv to 0 & solve)

Euclidean norm

$$\nabla_{\vec{x}} f = \nabla_{\vec{x}} \left[(A\vec{x}-\vec{b})^T(A\vec{x}-\vec{b}) + (\Gamma\vec{x})^T(\Gamma\vec{x}) \right]$$

matrix rule for transpose on matrix mult.

$$= \nabla_{\vec{x}} \left[(\vec{x}^T A^T - \vec{b}^T)(A\vec{x}-\vec{b}) + x^T \Gamma^T \Gamma \vec{x} \right]$$

$$= \nabla_{\vec{x}} \left[\vec{x}^T A^T A \vec{x} - \vec{x}^T A^T \vec{b} - b^T A\vec{x} - \vec{b}^T\vec{b} + \vec{x}^T \Gamma^T \Gamma \vec{x} \right]$$

$(b^T A\vec{x} = \vec{x}^T A^T \vec{b})$

$$= \nabla_{\vec{x}} \left[\vec{x}^T A^T A \vec{x} - 2\vec{x}^T A^T \vec{b} - \vec{b}^T\vec{b} + \vec{x}^T \Gamma^T \Gamma \vec{x} \right]$$

$\left(\frac{\partial(\vec{x}^T\vec{x})}{\partial \vec{x}} = 2\vec{x}\right)$

$$= 2A^T A\vec{x} - 2A^T\vec{b} + 2\Gamma^T\Gamma\vec{x}$$

$$= 0$$

Thus $2A^T A\vec{x} - 2A^T\vec{b} + 2\Gamma^T\Gamma\vec{x} = 0$

$$(\cancel{2}A^T A + \cancel{2}\Gamma^T\Gamma)\vec{x} = \cancel{2}A^T\vec{b}$$

$$(A^T A + \Gamma^T\Gamma)\vec{x} = A^T\vec{b}$$

$$\vec{x} = (A^T A + \Gamma^T\Gamma)^{-1}A^T\vec{b}$$ → closed form solution, so x^*

* I checked solution here

To get rid of Γ, $\Gamma = \sqrt{\lambda} I$, so $\boxed{x^* = (A^T A + \lambda I)^{-1} A^T \vec{b}}$

(c) See attached graphs :

RMSE on validation : 0.8840
RMSE on test : 0.8628
$\lambda^* = 8.5264$

(d) Minimize $\|A\vec{x} + b\vec{1} - \vec{y}\|_2^2 + \|\Gamma\vec{x}\|_2^2$ $(= f)$

$$= (A\vec{x} + b\vec{1} - \vec{y})^T(A\vec{x} + b\vec{1} - \vec{y}) + (\Gamma\vec{x})^T(\Gamma\vec{x})$$

remember $A^T B = B^T A$

$$= (\vec{x}^T A^T + b\vec{1}^T - \vec{y}^T)(A\vec{x} + b\vec{1} - \vec{y}) + (\vec{x}^T\Gamma^T)(\Gamma\vec{x})$$

$$= \vec{x}^T A^T A\vec{x} + 2b\vec{1}^T A\vec{x} - 2\vec{y}^T A\vec{x} - 2b\vec{1}^T\vec{y} + b^2 n + \vec{y}^T\vec{y} + \vec{x}^T\Gamma^T\Gamma\vec{x}$$

To minimize $(\nabla_x f = 0)$:

$$\left[\begin{array}{l} \nabla_x f = 2A^T A\vec{x} + 2bA^T\vec{1} - 2A^T\vec{y} + 2\Gamma^T\Gamma\vec{x} = 0 \\ \nabla_b f = 2\vec{1}^T A\vec{x} - 2\vec{1}^T\vec{y} + 2bn = 0 \\ b^* = \frac{\vec{1}^T(\vec{y} - A\vec{x})}{n} \end{array}\right]$$

Plug in!

Solve for b

(I checked solution for this)

divided out the 2!

$$\Rightarrow \nabla_x f = A^T A\vec{x} + \left(\frac{\vec{1}^T(\vec{y} - A\vec{x})}{n}\right)A^T\vec{1} - A^T\vec{y} = 0$$

[illegible] for $\vec{x}$:

$$\vec{x}^* = \left[A^T\left(\vec{I} - \frac{1}{n}\vec{1}\vec{1}^T\right)A + \Gamma^T\Gamma\right]^{-1} A^T\left(\vec{I} - \frac{1}{n}\vec{1}\vec{1}^T\right)\vec{y}$$

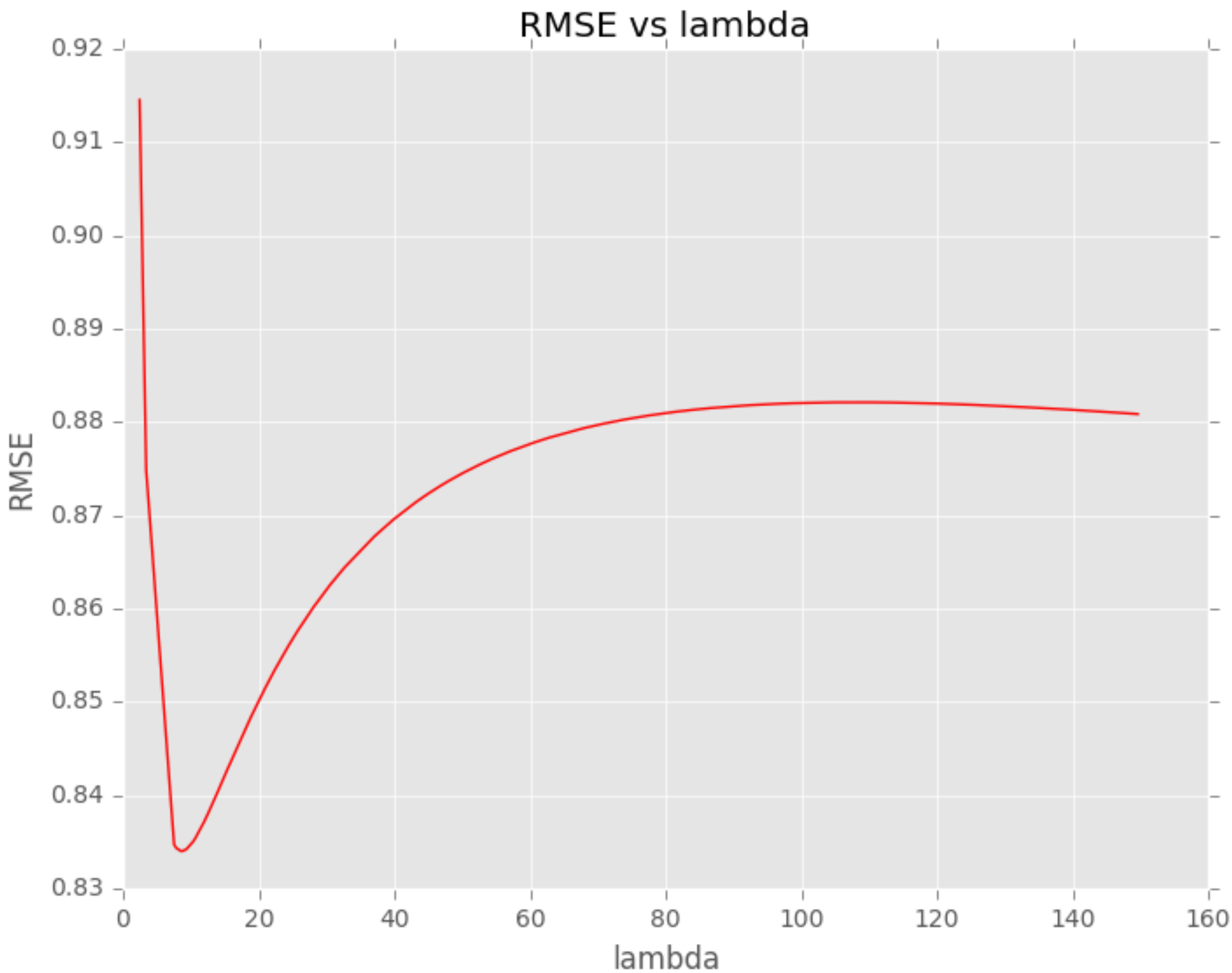
with identity $\vec{I}$ & $\vec{1}$ vector of ones.

By the code :

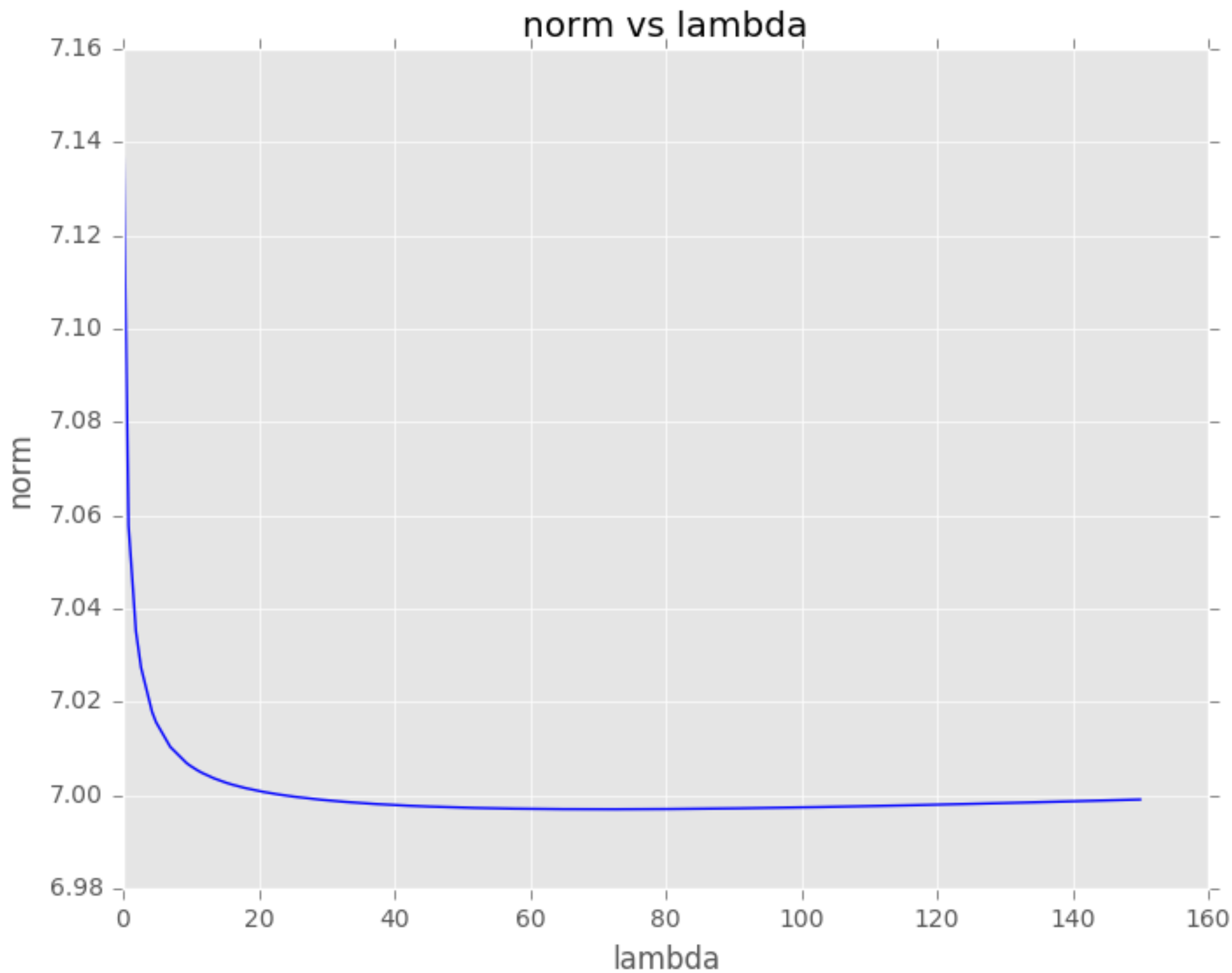
$$\boxed{\begin{array}{ll} \text{diff in bias} & : 4.3690 \times 10^{-10} \\ \text{diff in weights} : & 5.7736 \times 10^{-10} \end{array}}$$

(e) See attached :

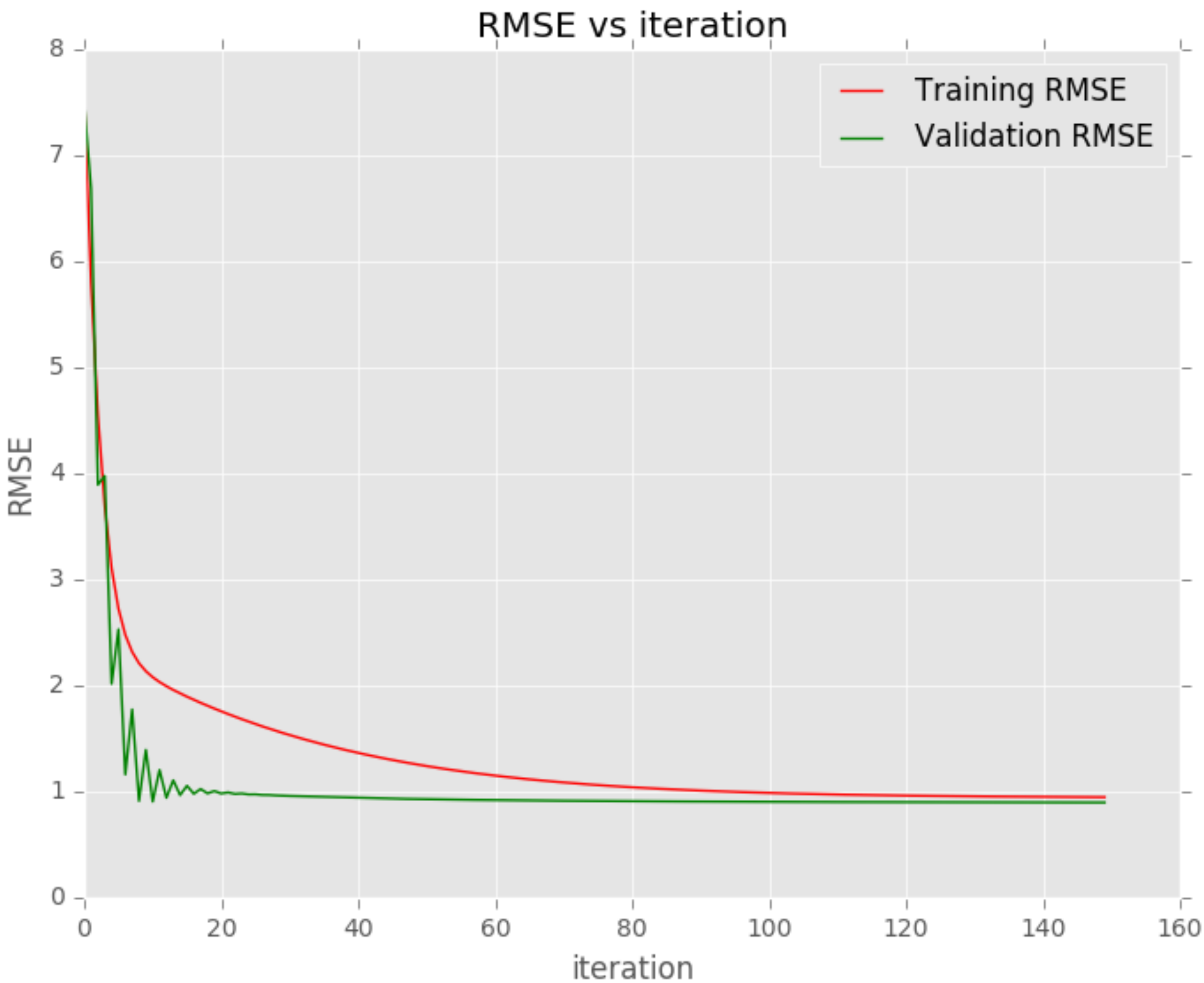
$$\boxed{\begin{array}{ll} \text{Diff bias} : & 1.5387 \times 10^{-1} \\ \text{Diff weights} : & 8.0108 \times 10^{-1} \end{array}}$$



RMSE vs lambda
0.92
0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
0.83
RMSE
0
20
40
60
80
100
120
140
160
lambda



norm vs lambda
7.16
7.14
7.12
7.10
7.08
7.06
7.04
7.02
7.00
6.98
norm
0
20
40
60
80
100
120
140
160
lambda


RMSE vs iteration
Training RMSE
Validation RMSE
RMSE
iteration
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160

```
In [7]: %run hw2pr3.py
==> Loading data...
==> Step 1: RMSE vs lambda...
==> Plotting completed.
==> The optimal regularization parameter is  8.5264.
==> The RMSE on the validation set with the optimal regularization parameter is
0.8340.
==> The RMSE on the test set with the optimal regularization parameter is
0.8628.

==> Step 2: Norm vs lambda...
==> Plotting completed.

==> Step 3: Linear regression without bias...
--Time elapsed for training: 25.19 seconds
==> Difference in bias is  4.3690E-10
==> Difference in weights is  5.7736E-10

==> Step 4: Gradient descent
==> Running gradient descent...
-- Iteration25 - training rmse  1.6604 - gradient norm  3.6435E+04
-- Iteration50 - training rmse  1.2519 - gradient norm  2.4237E+04
-- Iteration75 - training rmse  1.0664 - gradient norm  1.5202E+04
-- Iteration100 - training rmse  0.9896 - gradient norm  9.7337E+03
-- Iteration125 - training rmse  0.9596 - gradient norm  6.3025E+03
-- Iteration150 - training rmse  0.9481 - gradient norm  4.1295E+03
--Time elapsed for training: 87.69 seconds
==> Plotting completed.
==> Difference in bias is  1.5387E-01
==> Difference in weights is  8.0108E-01
```