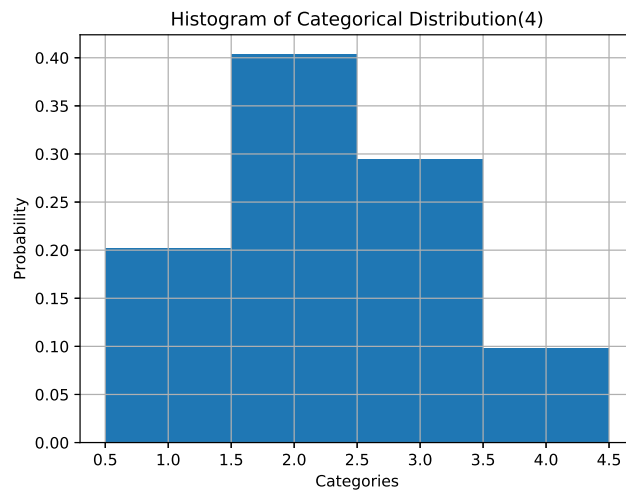


COMP 540      HW 1  
Peiguang Wang, Xinran Zhou  
Due: 1/18/2018

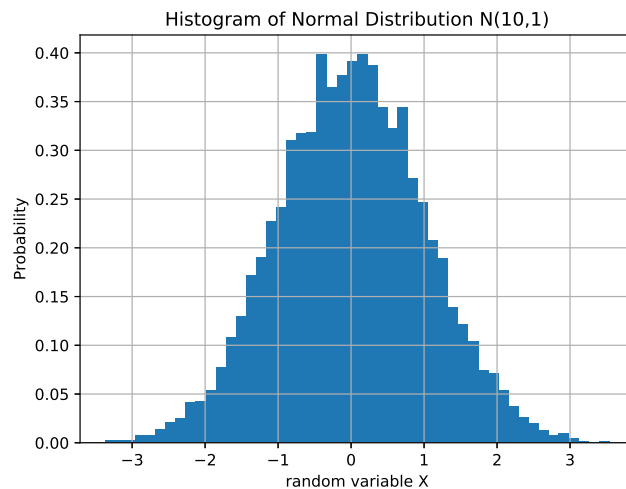
**Part 0: Background refresher**

1. Generate different distributions from uniform distribution:

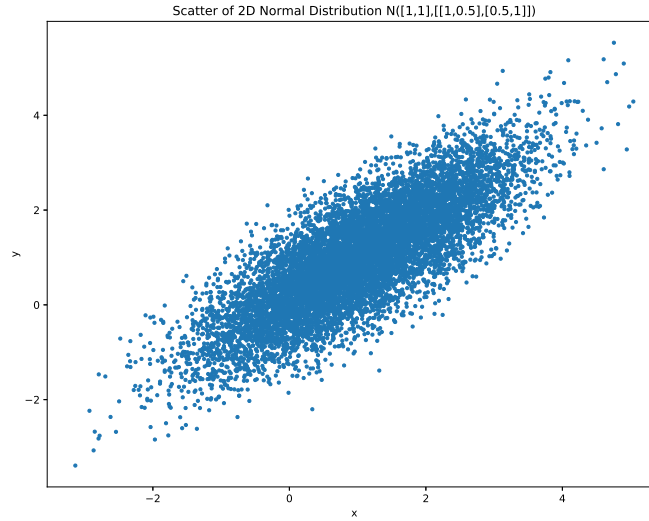
(a) Plot the histogram of a categorical distribution with probabilities  $[0.2, 0.4, 0.3, 0.1]$ .



(b) Plot the univariate normal distribution with mean of 10 and standard deviation of 1.



(c) Produce a scatter plot of the samples for a 2-D Gaussian with mean at  $[1,1]$  and a covariance matrix  $\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$



- (d) Test your mixture sampling code by writing a function that implements an equal weighted mixture of four Gaussians in 2 dimensions, centered at  $(\pm 1; \pm 1)$  and having covariance  $I$ . Estimate the probability that a sample from this distribution lies within the unit circle centered at  $(0.1, 0.2)$ .

**Solution.** The Probability that falls in unit circle with center at  $(0.1, 0.2)$  is 0.1815.

2. Prove that the sum of two independent Poisson random variables is also a Poisson random variable.

*Proof.* The characteristic function of a Poisson random variable is

$$\Phi_1(u) = e^{\lambda_1(e^{iu} - 1)}$$

Let  $X_1$  and  $X_2$  denote two independent Poisson random variables. Let  $X = X_1 + X_2$

Let  $\Phi_1(u)$  and  $\Phi_2(u)$  denote the characteristic functions of  $X_1$  and  $X_2$ :

$$\Phi_1(u) = e^{\lambda_1(e^{iu} - 1)}$$

$$\Phi_2(u) = e^{\lambda_2(e^{iu} - 1)}$$

Let  $\Phi(u)$  denote the characteristic functions of  $X$ . Since  $X = X_1 + X_2$ , we have:

$$\Phi(u) = \Phi_1(u)\Phi_2(u) = e^{\lambda_1(e^{iu} - 1)}e^{\lambda_2(e^{iu} - 1)}$$

Simplify the equation above,

$$\Phi(u) = e^{(\lambda_1 + \lambda_2)(\frac{\lambda_1}{\lambda_1 + \lambda_2}e^{iu} + \frac{\lambda_2}{\lambda_1 + \lambda_2}e^{iu}) - 1}.$$

That is

$$\Phi(u) = e^{(\lambda_1 + \lambda_2)(e^{iu} - 1)}.$$

Comparing with the characteristic function of Poisson distribution, we can see that  $X$  is also a Poisson random variable.  $\square$

3. Let  $X_0$  and  $X_1$  be continuous random variables. Show that if

$$P(X_0 = x_0) = \alpha_0 e^{-\frac{(x_0 - \mu_0)^2}{2\sigma_0^2}}$$

$$P(X_1 = x_1 | X_0 = x_0) = \alpha e^{-\frac{(x_1 - x_0)^2}{2\sigma^2}}$$

there exists  $\alpha_1$ ,  $\mu_1$  and  $\sigma_1$  such that

$$P(X_1 = x_1) = \alpha_1 e^{-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}}$$

Write down expressions for these quantities in terms of  $\alpha_0$ ,  $\alpha$ ,  $\mu_0$ ,  $\sigma_0$  and  $\sigma$ .

**Solution.** If  $X, Y$  are both Gaussian random variable, then

$$Y|X = x \sim N\left(\mu_Y + \rho \frac{\sigma_Y}{\sigma_X}(X - \mu_X), \sigma_Y^2(1 - \rho^2)\right)$$

where  $\mu_X$ ,  $\mu_Y$  are mean of  $X$  and  $Y$ ;  $\sigma_X^2$ ,  $\sigma_Y^2$  are variance of  $X$  and  $Y$ ;  $\rho$  is the correlation coefficient between  $X$  and  $Y$ .

According to the problem,  $X_0$ ,  $X_1$  and  $X_1|X_0$  are all Gaussian. So we have the following equations:

$$\begin{cases} \mu_1 + \rho \frac{\sigma_1}{\sigma_0}(x_0 - \mu_0) = x_0, \text{ for all } x_0 \\ \sigma_1^2(1 - \rho^2) = \sigma^2 \end{cases}$$

Solve the equation, then  $\sigma_1^2 = \sigma^2 + \sigma_0^2$ ,  $\mu_1 = -\mu_0$ . And since  $\alpha_1 = \frac{1}{\sqrt{2\pi\sigma_1}}$ , we have

$$\alpha_1 = \sqrt{\frac{1}{(1/\alpha)^2 + (1/\alpha_0)^2}}$$

.

4. Find the eigenvalues and eigenvectors of the following  $2 \times 2$  matrix  $A$ .

$$A = \begin{pmatrix} 13 & 5 \\ 2 & 4 \end{pmatrix}$$

**Solution.** Let  $\lambda$  and  $\mathbf{x}$  denote the eigenvalue and eigenvector of  $A$ . According to the definition of eigenvalue,

$$A\mathbf{x} = \lambda\mathbf{x}$$

Solve the equation to get eigenvalues

$$|A - \lambda I| = 0$$

That is,

$$\lambda^2 - 14\lambda + 42 = 0$$

$A$  has two eigenvalues:  $\lambda_1 = 14$ ,  $\lambda_2 = 3$ .

When  $\lambda = 14$ ,

$$\begin{aligned} (A - \lambda I)\mathbf{x} &= \begin{pmatrix} -1 & 5 \\ 2 & -10 \end{pmatrix} \mathbf{x} = 0 \\ \mathbf{x} &= (5 \quad 1)^T \end{aligned}$$

When  $\lambda = 3$ ,

$$\begin{aligned} (A - \lambda I)\mathbf{x} &= \begin{pmatrix} 10 & 5 \\ 2 & 1 \end{pmatrix} \mathbf{x} = 0 \\ \mathbf{x} &= (1 \quad -2)^T \end{aligned}$$

In summary,  $A$  has two eigenvalues,  $\lambda_1 = 14$ ,  $\lambda_2 = 3$ . The corresponding eigenvectors are  $\mathbf{x}_1 = (5 \quad 1)^T$  and  $\mathbf{x}_2 = (1 \quad -2)^T$ .

5. Provide one example for each of the following cases, where A and B are 2 × 2 matrices.

(a)  $(A + B)^2 \neq A^2 + 2AB + B^2$

(b)  $AB = 0, A \neq 0, B \neq 0$

**Solution.** (a) one example that satisfies (a) is:

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Calculate left,

$$left = (A + B)^2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Calculate right,

$$right = A^2 + 2AB + B^2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \mathbf{0} + \mathbf{0} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

And  $left \neq right$

(b) one example that satisfies (b) is:

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

where  $A \neq 0$ , and  $B \neq 0$ . Calculate  $AB$ ,

$$AB = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \mathbf{0}$$

6. Let  $u$  denote a real vector normalized to unit length. That is,  $u^T u = 1$ . Show that

$$A = I - 2uu^T$$

is orthogonal, i.e.,  $A^T A = I$ .

*Proof.* Derive from left,

$$A^T A = (I - 2uu^T)^T (I - 2uu^T) = (I - 2uu^T)(I - 2uu^T) = I - 2uu^T - 2uu^T + 4uu^T = I$$

So  $left = right$ . □

## Part 1: Locally weighted linear regression

1. Show that  $J(\theta)$  can be written in the form

$$J(\theta) = (X\theta - y)^T W (X\theta - y)$$

for an appropriate diagonal matrix  $W$ , where  $X$  is the  $m \times d$  input matrix and  $y$  is a  $m \times 1$  vector denoting the associated outputs. State clearly what  $W$  is.

*Proof.* We know that  $J(\theta)$  can also be written as

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m w^{(i)} (\theta^T x^{(i)} - y^{(i)})^2$$

where  $x^{(i)}$  is  $d \times 1$  vector and  $\theta$  is a  $d \times 1$  vector. We consider each row of the matrix  $X$  as a  $1 \times d$  vector  $x^i$ , so we can write  $X = [x^1, x^2, \dots, x^m]^T$ . So

$$J(\theta) = (X\theta - y)^T W (X\theta - y) = [x^1\theta - y^1, x^2\theta - y^2, \dots, x^m\theta - y^m] W [x^1\theta - y^1, x^2\theta - y^2, \dots, x^m\theta - y^m]^T$$

So  $W$  is a  $m \times m$  diagonal matrix

$$W = \begin{pmatrix} 2w^{(1)} & 0 & 0 & 0 \\ 0 & 2w^{(2)} & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 2w^{(m)} \end{pmatrix}$$

□

2. If all the  $w^{(i)}$ 's are equal to 1, the normal equation to solve for the parameter  $\theta$  is:

$$X^T X \theta = X^T y$$

and the value of  $\theta$  that minimizes  $J(\theta)$  is  $(X^T X)^{-1} X^T y$ . By computing the derivative of the weighted  $J(\theta)$  and setting it equal to zero, generalized the normal equation to the weighted setting and solve for  $\theta$  in closed form in terms of  $W$ ,  $X$  and  $y$ .

*Proof.*

$$J(\theta) = (X\theta - y)^T W (X\theta - y) = \theta^T X^T W X \theta - \theta^T X^T W y - y^T W X \theta + y^T W y$$

Compute the derivative of  $J(\theta)$

$$\frac{\partial J(\theta)}{\partial(\theta)} = 2X^T W X \theta - X^T W y - X^T W^T y$$

Since  $W$  is a diagonal matrix  $W = W^T$ , the equation can be written as

$$\frac{\partial J(\theta)}{\partial(\theta)} = 2X^T W X \theta - 2X^T W y$$

By setting it equal to zero, we can find the value of  $\theta$  that minimizes  $J(\theta)$ , the equation is :

$$X^T W X \theta = X^T W y$$

So the value of  $\theta$  in form in terms of  $W$ ,  $X$  and  $y$  is  $(X^T W X)^{-1} X^T W y$ . □

3. To predict the target value for an input vector  $x$ , one choice for the weighting function  $w^{(i)}$  is:

$$w^{(i)} = \exp\left(-\frac{(x - x^{(i)})^T (x - x^{(i)})}{2\tau^2}\right)$$

Points near  $x$  are weighted more heavily than points far away from  $x$ . The parameter  $\tau$  is a band width defining the sphere of influence around  $x$ . Note how the weights are defined by the input  $x$ . Write down an algorithm for calculating  $\theta$  by batch gradient descent for locally weighted linear regression. Is locally weighted linear regression a parametric or a non-parametric method?

**Solution.** Algorithm to calculate  $\theta$  is shown below.

---

**Algorithm 1** Weighted linear regression using batch gradient descent

---

- 1: Calculate  $W$  using  $w^{(i)} = \exp\left(-\frac{(x - x^{(i)})^T (x - x^{(i)})}{2\tau^2}\right)$ ;
  - 2: Set learnign rate  $\alpha$ ;
  - 3: **for** enough iterations **do**
  - 4:     Update  $\theta$  where  $\theta_j = \theta_j - \alpha \sum_{i=1}^m w^{(i)} (\theta^T x^{(i)} - y^{(i)}) x_j$
  - 5: **end for**
  - 6: **return**  $\theta$ ;
- 

Because it uses data points when predicting, it is a non-parametric method.

## Part 2: Properties of the linear regression estimator

1. Show that  $E[\theta] = \theta^*$  for the least squares estimator.

*Proof.* In part 1 problem 2, we can get the value of  $\theta$  given the normal equation  $X^T X \theta = X^T y$  is

$$\theta = (X^T X)^{-1} X^T y$$

The data comes from the linear model:

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$$

the expectation of  $\theta$  is

$$\begin{aligned} E[\theta] &= E[(X^T X)^{-1} X^T y] \\ &= E[(X^T X)^{-1} X^T (X \theta^* + \epsilon)] \\ &= E[(X^T X)^{-1} (X^T X \theta^* + X^T \epsilon)] \\ &= E[(X^T X)^{-1} X^T X \theta^* + (X^T X)^{-1} X^T \epsilon] \\ &= E[\theta^*] + E[(X^T X)^{-1} X^T \epsilon] \end{aligned}$$

since each  $\epsilon^{(i)}$  is an independent random variable drawn from a normal distribution with zero mean and variance  $\sigma^2$  and  $\theta^*$  is a true parameter that has certain value. Then  $E[\theta]$  can be written as

$$E[\theta] = E[\theta^*] + 0 = \theta^*$$

□

2. Show that the variance of the least squares estimator is  $Var(\theta) = (X^T X)^{-1} \sigma^2$ .

*Proof.*

$$Var(\theta) = E[\theta^2] - (E[\theta])^2$$

since we already knew that  $E[\theta] = \theta^*$ . So in order to get  $Var(\theta)$ , all we need to do is to compute  $E[\theta^2]$ .

$$\begin{aligned} E[\theta^2] &= E[(X^T X)^{-1} X^T y] (X^T X)^{-1} X^T y^T \\ &= E[(\theta^* + (X^T X)^{-1} X^T \Sigma)(\theta^* + (X^T X)^{-1} X^T \Sigma)^T] \\ &= E[\theta^* \theta^{*T} + (X^T X)^{-1} X^T \Sigma \theta^{*T} + \theta^* \Sigma^T X (X^T X)^{-1} + (X^T X)^{-1} X^T \Sigma \Sigma^T X (X^T X)^{-1}] \end{aligned}$$

$\Sigma$  is the covariance matrix generated by  $\epsilon$  and each  $\epsilon^{(i)}$  is an independent random variable drawn from a normal distribution with zero mean and variance  $\sigma^2$ . Therefore the expectation of  $\Sigma$  is zero.  $\Sigma$  is also independent to  $X$  and  $\theta^*$ ,  $\Sigma = \sigma^2 I$ , where  $I$  is the identity matrix. Therefore

$$E[\theta^2] = (\theta^*)^2 + \sigma^2 I (X^T X)^{-1}$$

Then we have

$$\begin{aligned} Var(\theta) &= E[\theta^2] - (E[\theta])^2 \\ &= (\theta^*)^2 + \sigma^2 I (X^T X)^{-1} - (\theta^*)^2 \\ &= (X^T X)^{-1} \sigma^2 \end{aligned}$$

□

## Part 3: Implementing Linear Regression

### Problem 3.1.A1 Implementing gradient descent

Include Figure 1, Figure 2 and Figure 3

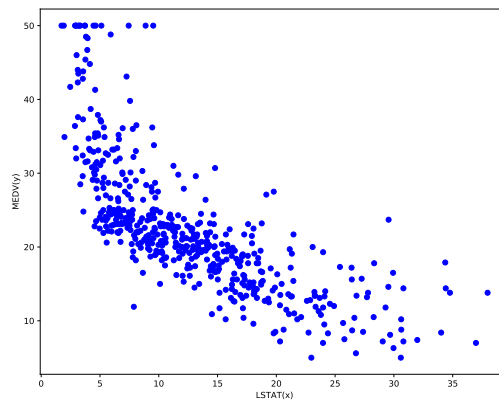


Figure 1: Scatter plot of training data

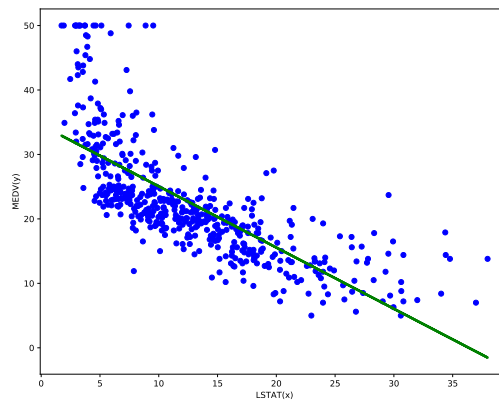


Figure 2: Fitting a linear model to the data in Figure 1

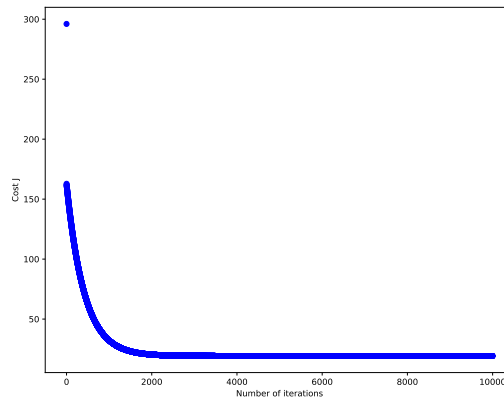


Figure 3: Convergence of gradient descent to fit the linear model in Figure 2

1. Qualitative analysis of the linear fit. What can you say about the quality of the linear fit for this data? In your assignment writeup.pdf, explain how you expect the model to perform at the low and high ends of values for LSTAT? How could we improve the quality of the fit?

**Solution.** As we can see in the Figure 2, the linear fit for this data is not that good, especially at the high and low ends. The regression should have some curve at the low and high ends of values for LSTAT, which we can replace  $x$  with some non-linear function to model non-linear relationship. Using polynomial regression.

### Problem 3.1.A3 Predicting on unseen data

1. Make predictions on median home values for census tracts where the percentage of the population of lower economic status is 5% and 50%.

**Solution.** For lower status percentage = 5, we predict a median home value of 298034.4941220727  
For lower status percentage = 50, we predict a median home value of -129482.12889798547

2. Comparing with sklearn's linear regression model.

**Solution.** We can see from the data below that the results using different methods are quite similar. The coefficients computed by sklearn: 34.5538408794 and -0.950049353758.  
The coefficients computed by gradient descent: 34.55363411 and -0.95003694.

### Problem 3.1.B2 Loss function and gradient descent

### Problem 3.1.B3 Predicting on unseen data

For average home in Boston suburbs, we predict a median home value of 225328.063241

### Problem 3.1.B4: Normal equations (5 points)

For average home in Boston suburbs, we predict a median home value of 225328.06324113606. The prediction matches.

### Problem 3.1.B5: Exploring convergence of gradient descent

Figure 5

1. Exploring convergence of gradient descent. What are good learning rates and number of iterations for this problem?



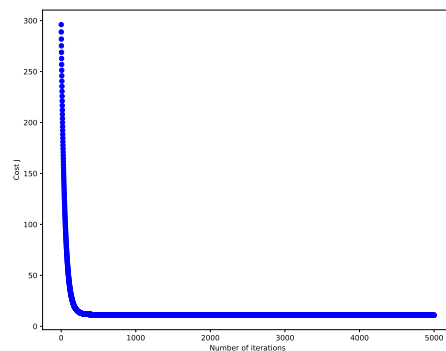


Figure 4: Number of iteration

### Problem 3.2 Visualizing the dataset

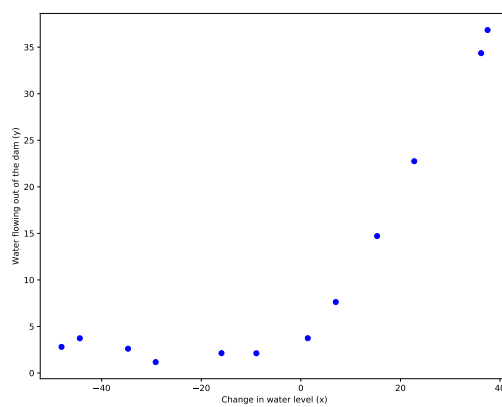


Figure 5: The training data for regularized linear regression

### Problem 3.2.A2 Regularized linear regression cost function

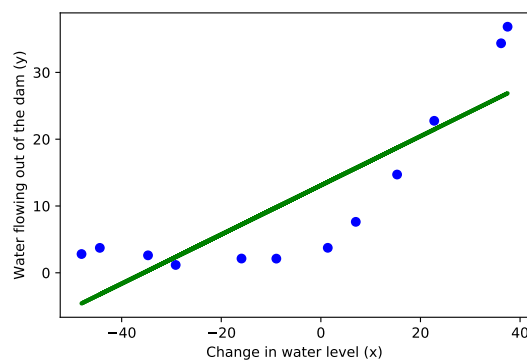


Figure 6: The best fit line for the training data

Problem 3.2.A3 Learning curves

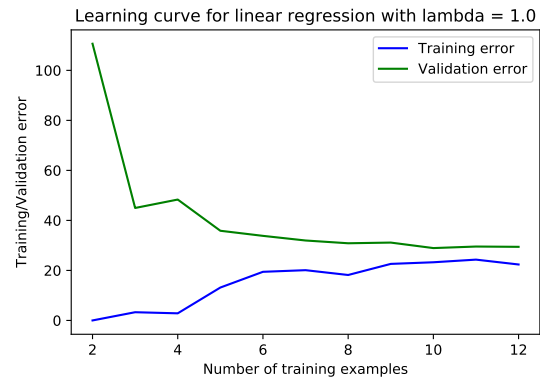


Figure 7: Learning curves

Problem 3.2 Learning polynomial regression models

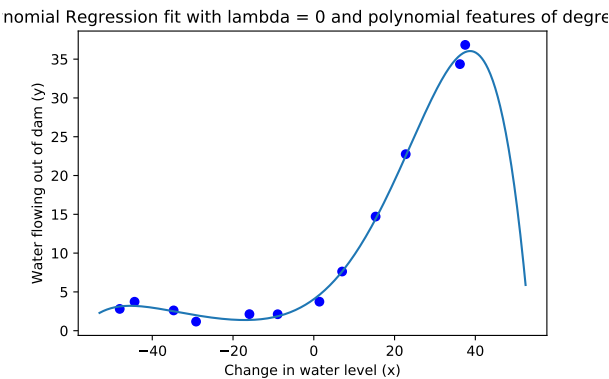


Figure 8: Polynomial fit for  $\lambda = 0$  with a  $p=6$  order model.

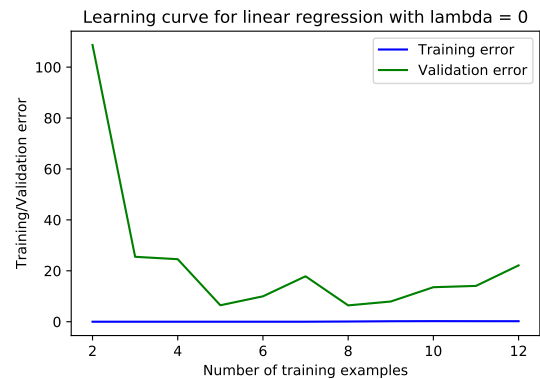


Figure 9: Learning curve for  $\lambda = 0$ .

Problem 3.2.A4: Adjusting the regularization parameter

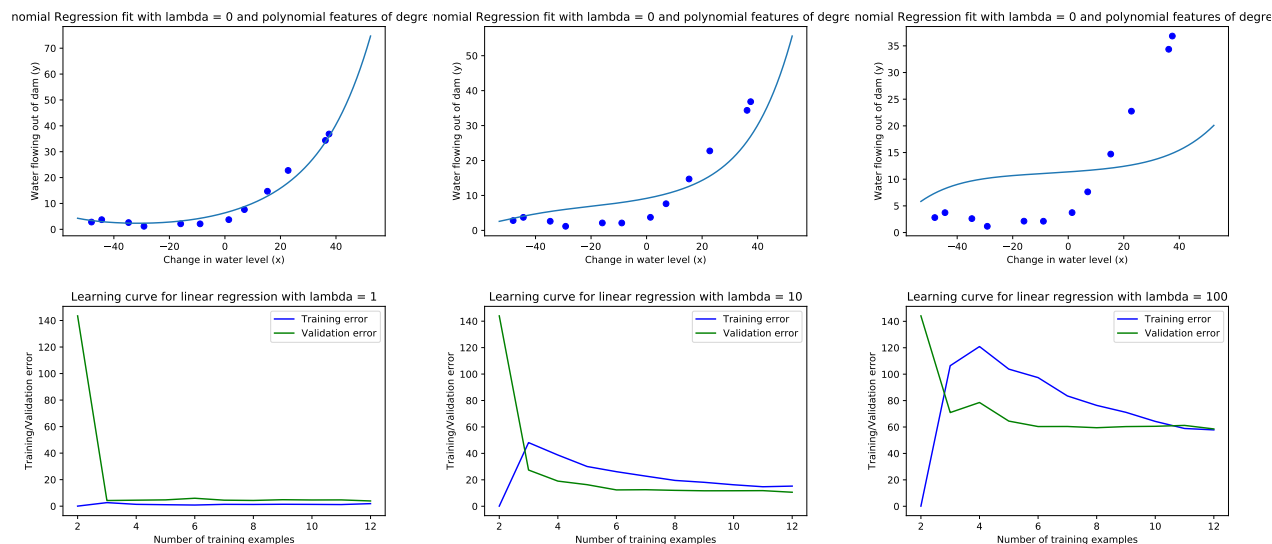


Figure 10: Adjusting the regularization parameter

Increasing  $\lambda$  results in less overfitting but also greater bias. The training error and testing error increase as long as the  $\lambda$  increases.

### Problem 3.2.A5: Selecting using a validation set

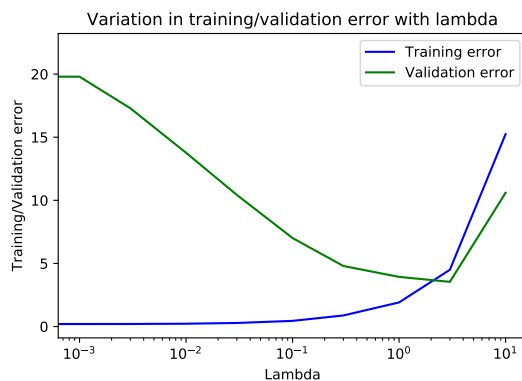


Figure 11: training and validation error on different lambda

The best model is  $\lambda = 0.3$ . When  $\lambda = 0.3$ , the validation error is the smallest.

### Problem 3.2.A6: Computing test set error on the best model

When  $\lambda = 0.3$ , the model has the smallest validation error. The test error is: 5.857077821089781

### Problem 3.2.A7: Plotting learning curves with randomly selected examples

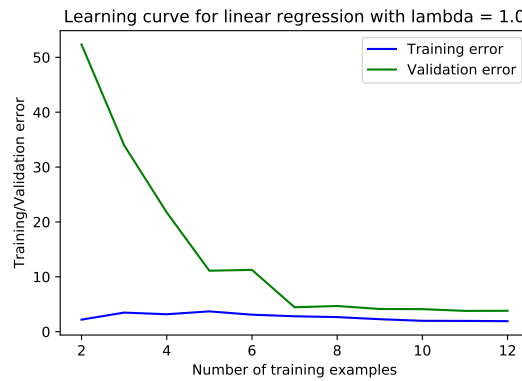


Figure 12: Averaged Learning curve for  $\lambda = 1$

### Extra Credit: Building regularized models for Boston data set

1. Use sklearn's built-in functions to split the data into training, validation and test sets.

**Solution.** I divide the data set into 3 different parts. The training set comes from 50% of the data, the validation set comes from 25% of the data, and the test set comes from 25% of the data.

2. What is the lowest achievable error on the test set with  $\lambda = 0$  ?

**Solution.** When  $\lambda = 0$ , the error of on the test set is 12.3122226194.

3. Select the best value for  $\lambda$  and report the test set error with the best  $\lambda$ .

**Solution.** I select the value of  $\lambda$  using validation set. The result is shown in figure 13.

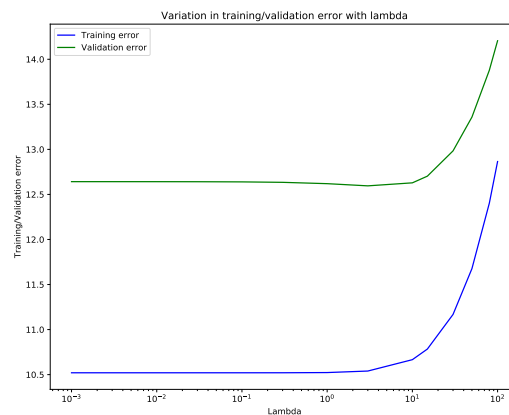


Figure 13: Select value for  $\lambda$  using validation set

We can see that when  $\lambda = 3$ , the validation error is the lowest.

4. Polynomial regression: quadratic features.

**Solution.** After transform data into polynomial, I repeated the steps above. At first I do  $\lambda$  selection using validation set. The  $\lambda$  selection result is shown in figure 14

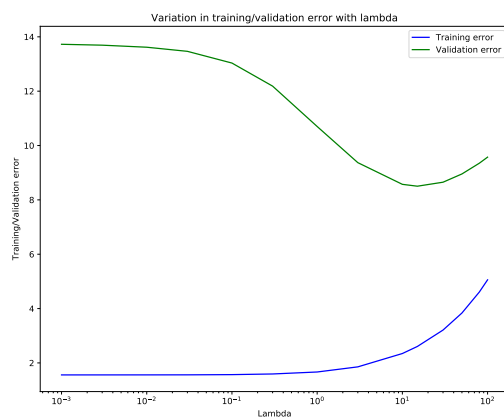


Figure 14: Select value for  $\lambda$  using validation set for quadratic features

We select  $\lambda = 10$ . Then we train the model with  $\lambda = 10$ , the error is: 7.09036691533.

5. Polynomial regression: cubic features.

**Solution.** After transform data into polynomial, I repeated the steps above. At first I do  $\lambda$  selection using validation set. The  $\lambda$  selection result is shown in figure 15

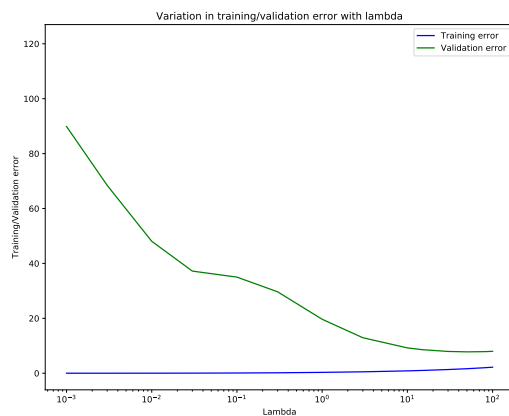


Figure 15: Select value for  $\lambda$  using validation set for cubic features

We select  $\lambda = 10$ . Then we train the model with  $\lambda = 10$ , the error is: 7.09036691533.