COMP 540 HW 1

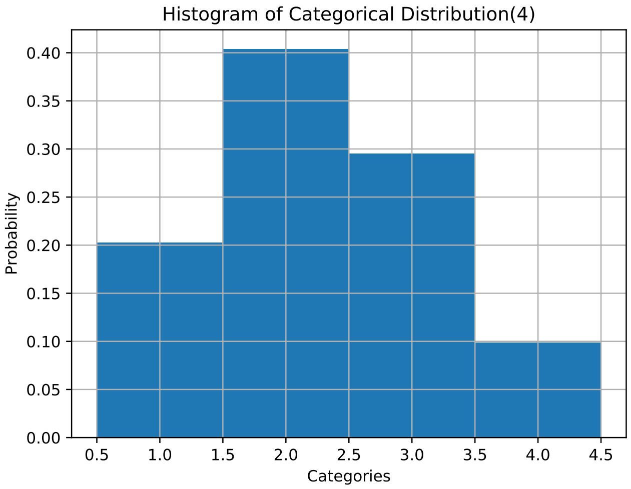
Peiguang Wang, Xinran Zhou

Due: 1/18/2018

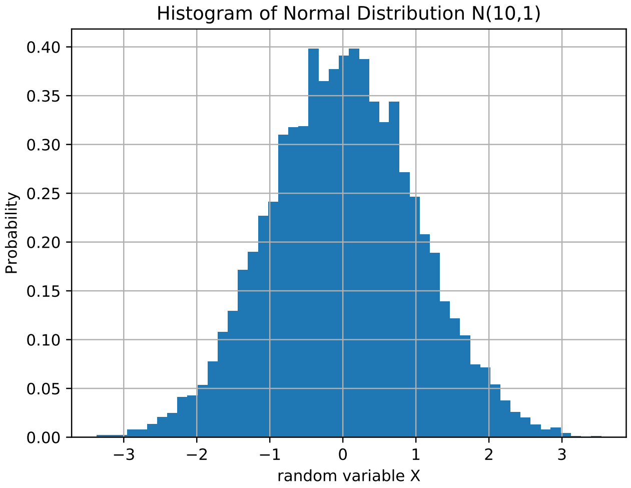
Part 0: Background refresher

1. Generate different distributions from uniform distribution:

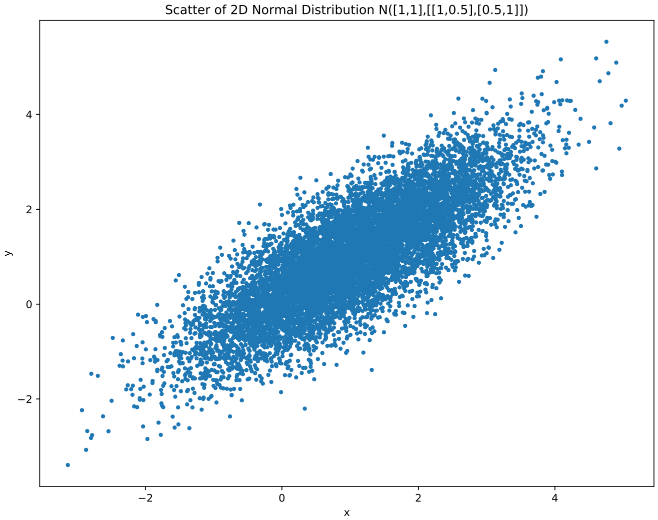
   (a) Plot the histogram of a categorical distribution with probabilities [0.2,0.4,0.3,0.1].



   (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 [[1,0.5],[0.5,1]]



(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

$$\varPhi_1(u) = e^{\lambda_1(e^{iu}-1)}$$

Let X_1 and X_2 denote two independent Poisson random variables. Let $X = X_1 + X_2$
Let $\varPhi_1(u)$ and $\varPhi_2(u)$ denote the characteristic functions of X_1 and X_2:

$$\varPhi_1(u) = e^{\lambda_1(e^{iu}-1)}$$

$$\varPhi_2(u) = e^{\lambda_2(e^{iu}-1)}$$

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

$$\varPhi(u) = \varPhi_1(u)\varPhi_2(u) = e^{\lambda_1(e^{iu}-1)}e^{\lambda_2(e^{iu}-1)}$$

Simplify the equation above,

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

That is

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

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

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 α_1, μ_1 and σ_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 α_0, α, μ_0, σ_0 and σ.

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

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

where μ_X, μ_Y are mean of X and Y; σ_X^2, σ_Y^2 are variance of X and Y; ρ 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:

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

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 × 2 matrix A.

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

Solution. Let λ and $\boldsymbol{x}$ denote the eigenvalue and eigenvector of A. According to the definition of eigenvalue,

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

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 $\lambda = 14$,

$$(A - \lambda I)\boldsymbol{x} = \begin{pmatrix} -1 & 5 \\ 2 & -10 \end{pmatrix}\boldsymbol{x} = 0$$

$$\boldsymbol{x} = \begin{pmatrix} 5 & 1 \end{pmatrix}^T$$

When $\lambda = 3$,

$$(A - \lambda I)\boldsymbol{x} = \begin{pmatrix} 10 & 5 \\ 2 & 1 \end{pmatrix}\boldsymbol{x} = 0$$

$$\boldsymbol{x} = \begin{pmatrix} 1 & -2 \end{pmatrix}^T$$

In summary, A has two eigenvalues, $\lambda_1 = 14$, $\lambda_2 = 3$. The corresponding eigenvectors are $\boldsymbol{x_1} = \begin{pmatrix} 5 & 1 \end{pmatrix}^T$ and $\boldsymbol{x_2} = \begin{pmatrix} 1 & -2 \end{pmatrix}^T$.

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., $A^T A = 1$.

   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 $left = right$. □

7. Proof the following assertions.

   (a) $f(x) = x^3$ is convex for $x \geq 0$.

   Proof.

$$f^{'}(x) = 3x^2$$
$$f^{''}(x) = 6x$$

   for $x \geq 0$, $f^{''}(x) \geq 0$ always exists. So $f(x) = x^3$ is convex for $x \geq 0$ □

   (b) $f(x_1, x_2) = max(x_1, x_2)$ is convex on R^2.

   Proof. We suppose $a = max(x_1, x_2) = x_{max}$, $b = max(x_1^{'}, x_2^{'}) = x_{max}^{'}$, then

$$left = f(\lambda x_1 + (1-\lambda)x_1^{'}, \lambda x_2 + (1-\lambda)x_2^{'})$$

$$right = \lambda f(x_1, x_2) + (1-\lambda) f(x_1^{'}, x_2^{'}) = \lambda x_{max} + (1-\lambda)x_{max}^{'}$$

   therefore,

$$max(\lambda x_1 + (1-\lambda)x_1^{'}, \lambda x_2 + (1-\lambda)x_2^{'}) \leq \lambda x_{max} + (1-\lambda)x_{max}^{'}$$

   $f(x_1, x_2) = max(x_1, x_2)$ is convex on R^2. □

(c) If univariate functions f f and g are convex on S, then $f+g$ is convex on S.

Proof. We define function $u = f+g$, so we need to prove that

$$u(\lambda a + (1-\lambda)b) \leq \lambda u(a) + (1-\lambda)u(b)$$

$$left = f(\lambda a + (1-\lambda)b) + g(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda)f(b) + \lambda g(a) + (1-\lambda)g(b)$$

the right part of the previous equation can be written as the following

$$right = \lambda[f(a)+g(a)] + (1-\lambda)[f(b)+g(b)]$$

$$left \leq right$$

therefore function $f+g$ is convex. □

(d) If univariate functions f and g are convex and non-negative on S, and have their minimum within S at the same point, then fg is convex on S.

Proof. We define a function $u = fg$, then

$$u(\lambda a+(1-\lambda)b) = f(\lambda a+(1-\lambda)b)g(\lambda a+(1-\lambda)b) \leq [\lambda f(a)+(1-\lambda)f(b)+\lambda g(a)][\lambda g(a)+(1-\lambda)g(b)] \leq \lambda^2 f(a)g(a)+(1-$$

$$left \leq \lambda f(a)g(a) + (1-\lambda)f(b)g(b) = \lambda u(a) + (1-\lambda)u(b)$$

therefore function fg is convex □

8. The entropy of a categorical distribution on K values is defined as

$$H(p) = -\sum p_i log(p_i)$$

Using the method use Lagrange multipliers, find the categorical distribution that has the highest entropy.

Proof. Since the sum of the probability for all events is 1, we can get constraint condition that $g(P) = 1$, where

$$g(p_1, p_2, p_3, ..., p_n) = \sum_{i=1}^{n} p_i = 1$$

use Lagrange multipliers, for i in range 1 to n,

$$\frac{\partial}{\partial p_k}(-\sum_{i=1}^{n} p_i log(p_i) + \lambda(\sum_{i=1}^{n} p_k - 1)) = 0$$

compute this equation, we get

$$-(\frac{1}{ln2} + log_2 p_k) + \lambda = 0$$

solving the above equation, we have the constraint condition $\sum p_k = 1$, and each p_i is identical.

$$p_k = \frac{1}{n}$$

therefore, we have proved that the Uniform distribution has the highest entropy. □

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.

Solution. 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 θ is a $d \times 1$ vector. We consider each row of the matrix X as as a $1 \times d$vector x^i, so we can write $X = [x^1, x^2, ..., x^m]^T$. So

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

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

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

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

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

and the value of θ 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 θ in closed form in terms of W, X and y.

Proof.

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

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 matirx $W = W^T$, 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 θ that minimizes $J(\theta)$, the equation is :

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

So the value of θ 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:

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

Points near x are weighted more heavily than points far away from x. The parameter τ 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 θ 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 θ is shown below.

Algorithm 1 Weighted linear regression using batch gradient descent

```
Calculate $W$ using $w^{(i)} = \exp(-\frac{(x-x^{(i)})^T(x-x^{(i)})}{2\tau^2})$;
Set learning rate $\alpha$;
for enough iterations do
    Update $\theta$ where $\theta_j = \theta_j - \alpha \sum_{i=1}^{m} w^{(i)}(\theta^T x^{(i)} - y^{(i)})x_j$
end for
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 thee value of θ given the normal equation $X^TX\theta = X^Ty$ is

$$\theta = (X^TX)^{-1}X^Ty$$

The date comes from the linear model:

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

the expectation of θ is

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

since each $\epsilon^{(i)}$is an independent random variable drawn from a normal distribution with zero mean and variance σ^2 and θ^* 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 $Var(\theta) = (X^TX)^{-1}\sigma^2$.

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 $E[\theta^2]$.

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

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

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

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

1. First we visualize the data by plotting the scatter plot of the training data, as shown in figure 1.

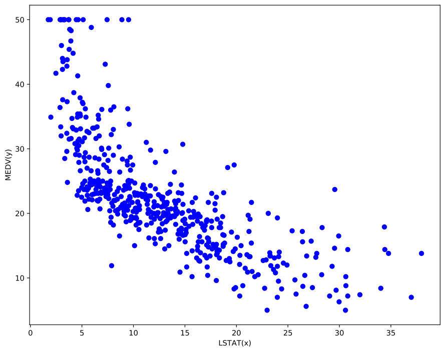


Figure 1: Scatter plot of training data

2. Fit the data using linear regression. The result is shown in 2.

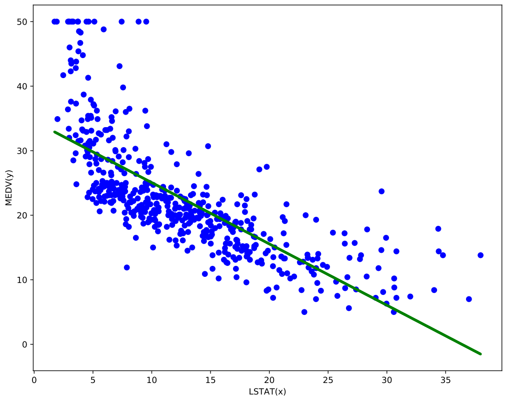


Figure 2: Fitting a linear model to the data in Figure 1

3. Visualize the cost to see the convergence of gradient descent. The result is shown in figure 3.

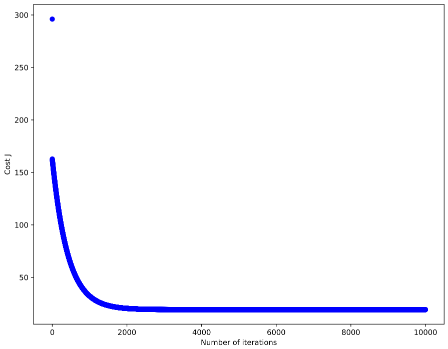


Figure 3: Convergence of gradient descent to fit the linear model in Figure 2

4. 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

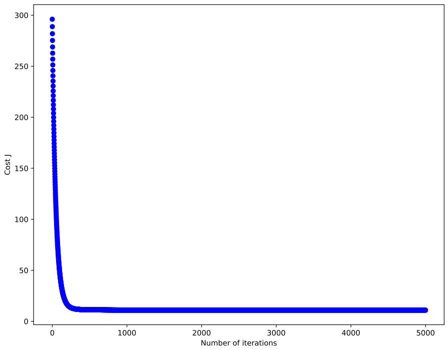


Figure 4: Number of iteration

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)

Predict unseen data using normal equation. For average home in Boston suburbs, we predict a median home value of 225328.06324113606. The prediction matches with the result using gradient descent.

Problem 3.1.B5: Exploring convergence of gradient descent

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

   Solution. We plotted the convergence figure with different learning rates. We used 0.01, 0.03, 0.1, and 0.3 as our different learning rates. The results are shown in figure 5. The up left figure is the convergence figure with learning 0.01. And up right for 0.03, bottom left for 0.1, bottom right for 0.3. We choose 0.3 as the best learning rate. The curve converges and it uses the least number of iterations to converge.

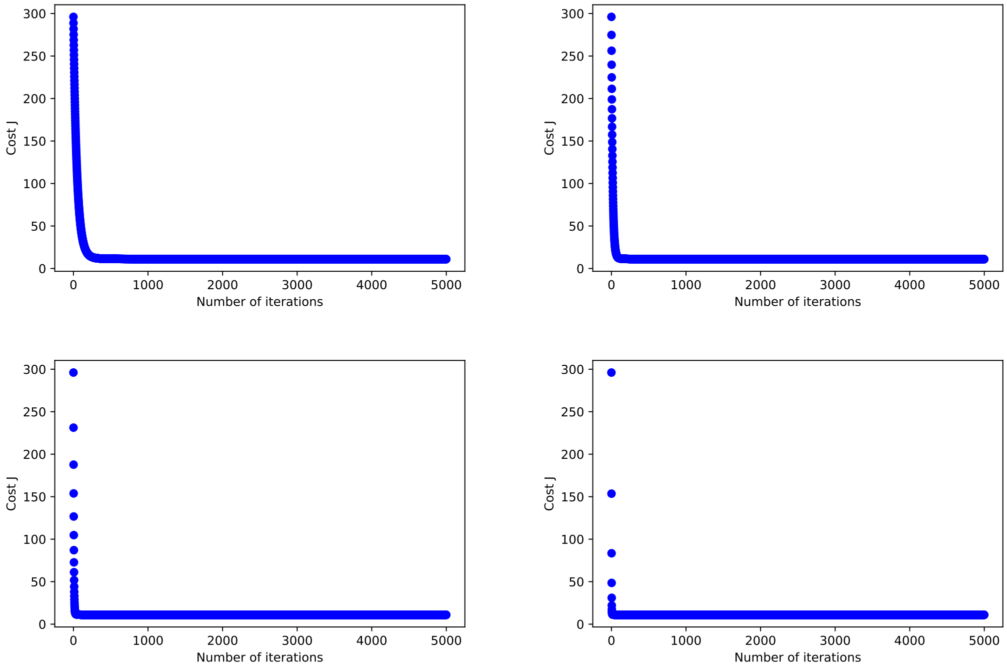


Figure 5: Adjusting the regularization parameter. The up left figure is the convergence figure with learning 0.01. And up right for 0.03, bottom left for 0.1, bottom right for 0.3.

Problem 3.2 Visualizing the dataset

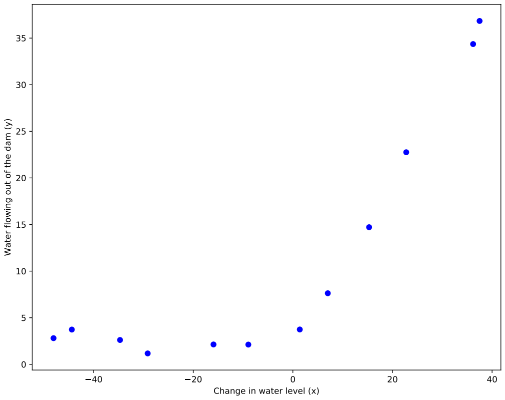


Figure 6: The training data for regularized linear regression

Problem 3.2.A2 Regularized linear regression cost function

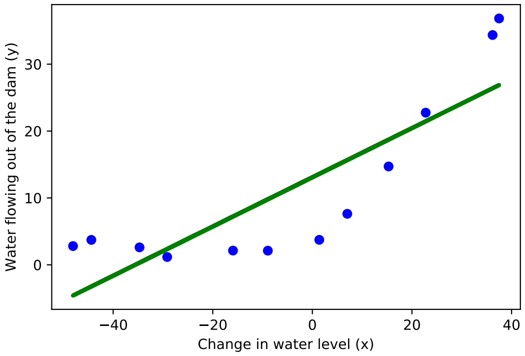


Figure 7: The best fit line for the training data

Problem 3.2.A3 Learning curves

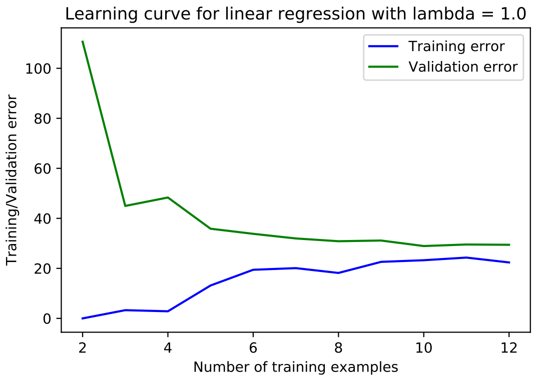


Figure 8: Learning curves when $\lambda = 1$

Problem 3.2 Learning polynomial regression models

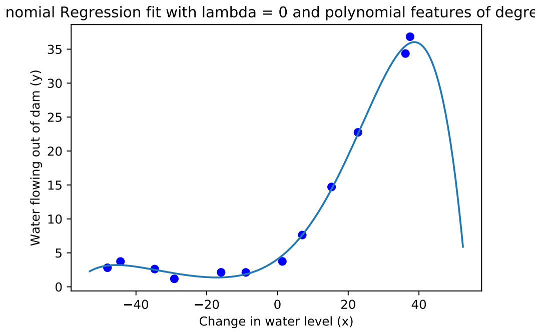


Figure 9: Polynomial fit for $\lambda = 0$ with a $p = 6$ order model.

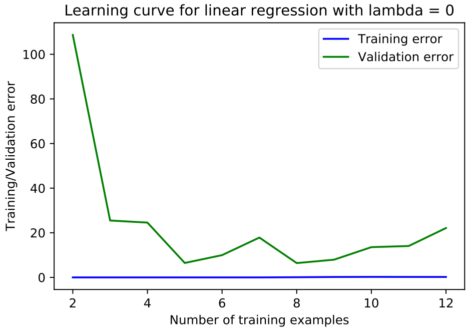


Figure 10: Learning curve for $\lambda = 0$.

Problem 3.2.A4: Adjusting the regularization parameter

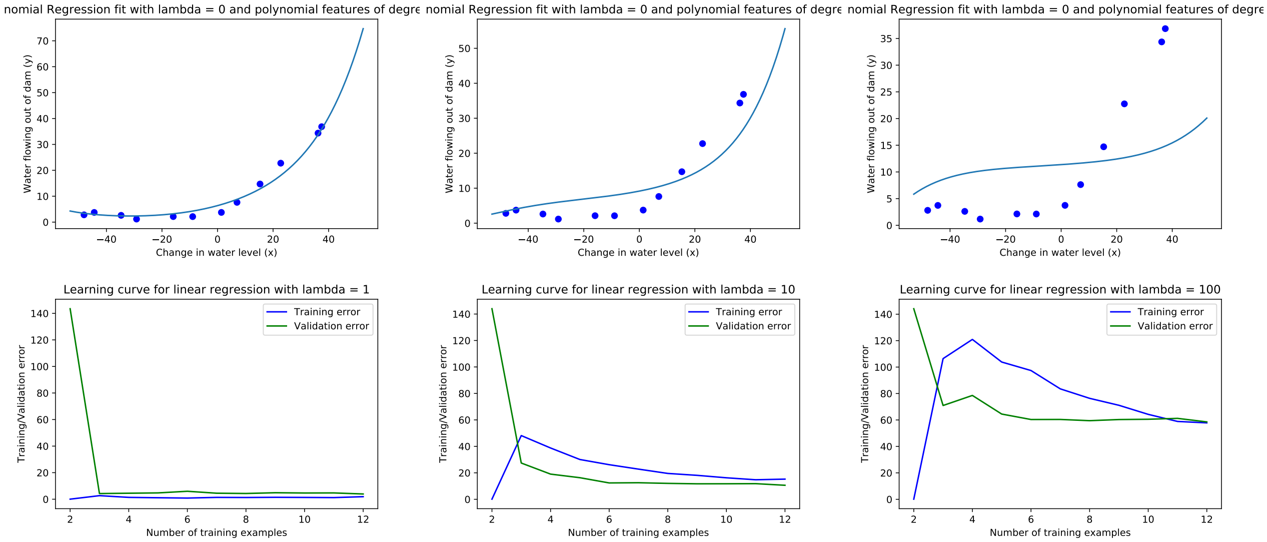


Figure 11: Polynomial figures and learning curves for different regularization parameters. The left column is for $lambda = 1$. The middle column is for $lambda = 10$. The right column is for $lambda = 100$.

We use different values of λ to see the effect of the regularization term. We set λ to 1, 10 and 100 and plot the polynomial and learning curves. The results are shown in figure 11. The left column is for $lambda = 1$. The middle column is for $lambda = 10$. The right column is for $lambda = 100$. As we can see from the figure, 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 λ using a validation set

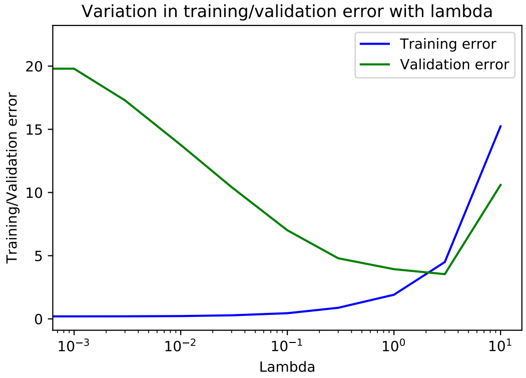


Figure 12: training and validation error on different λ

The best model is lambda = 3. When lambda = 3, the validation error is the smallest.

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

When lambda = 1, the model has the smallest validation error. The test error is: 5.944297830862095

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

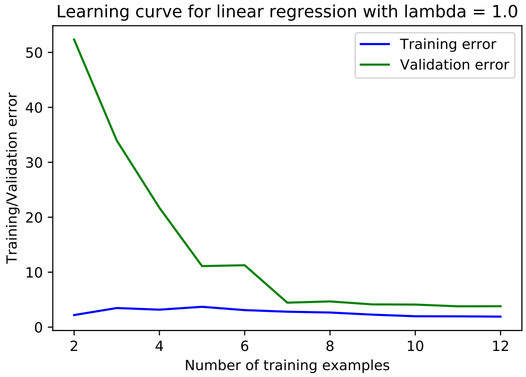


Figure 13: Averaged Learning curve for $\lambda = 1$

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 λ =0 ?

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

3. Select the best value for λ and report the test set error with the best λ.

   Solution. I select the value of λ using validation set. The result is shown in figure 14.

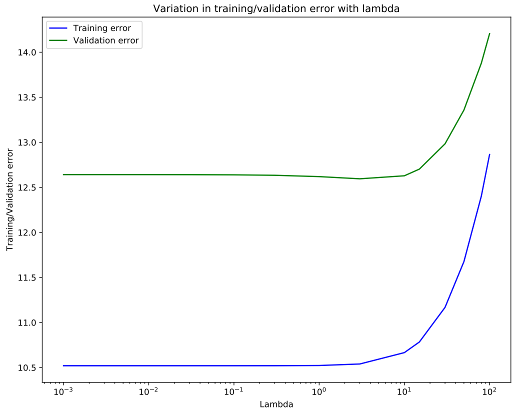


Figure 14: Select value for λ using validation set

   We can see that when $\lambda = 3$, the validation error is the lowest. However, I think the regularization term is not helpful here. We can see from the figure that the validation error almost does not change when λ changes from 10^{-4} to 1. Then it increases. It means the model underfits. In fact, the data has 13 features while there are only 250 training data points. I think no matter how we choose λ, it does not help to improve the model. Since the data points are not enough.

4. Polynomial regression: quadratic features.

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

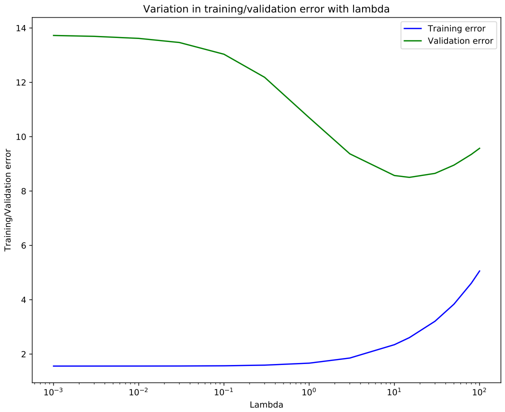


Figure 15: Select value for λ using validation set for quadratic features

   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 λ selection using validation set. The λ selection result is shown in figure 16

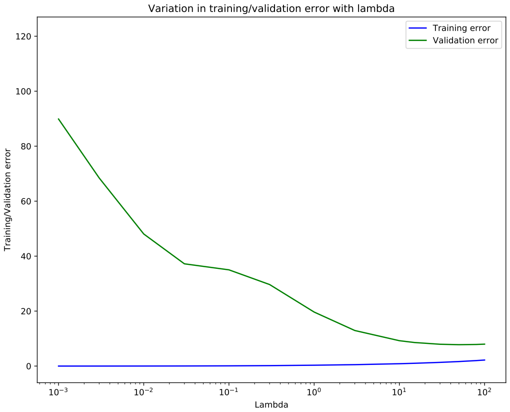


Figure 16: Select value for λ using validation set for cubic features

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