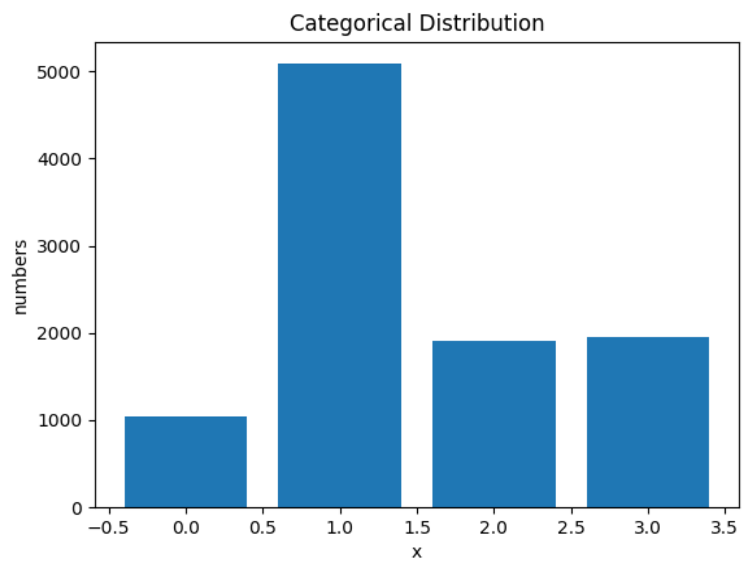
COMP540 HW1

Qichao Sun NetId: qs8
Yuyang Luo NetId:yl159

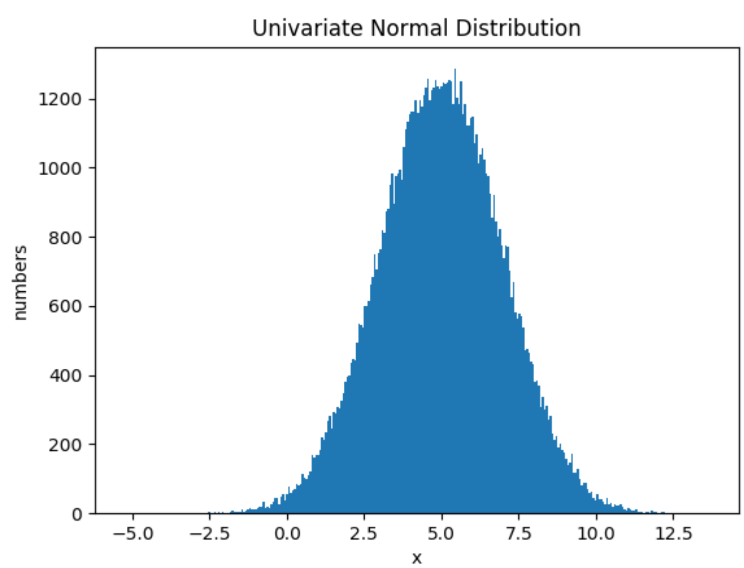
Problem 0

Question1: Write functions in Python to produce samples from four distributions: categorical, univariate Gaussian, multivariate Gaussian, and general mixture distributions.

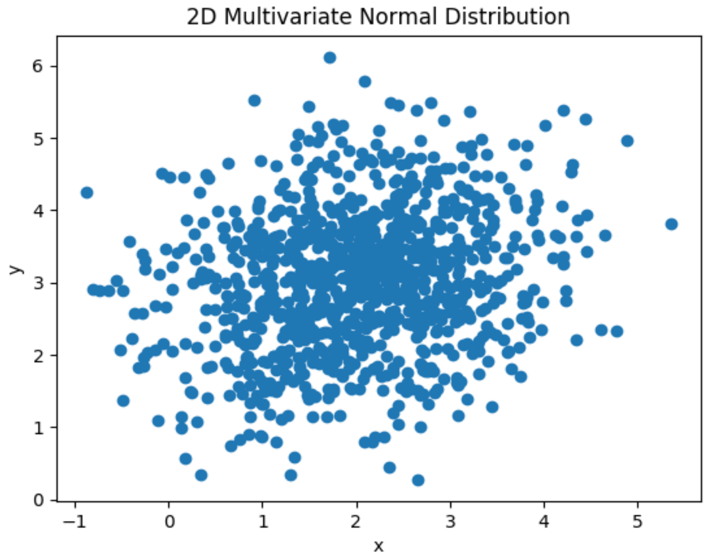
(1) Categorical Distribution



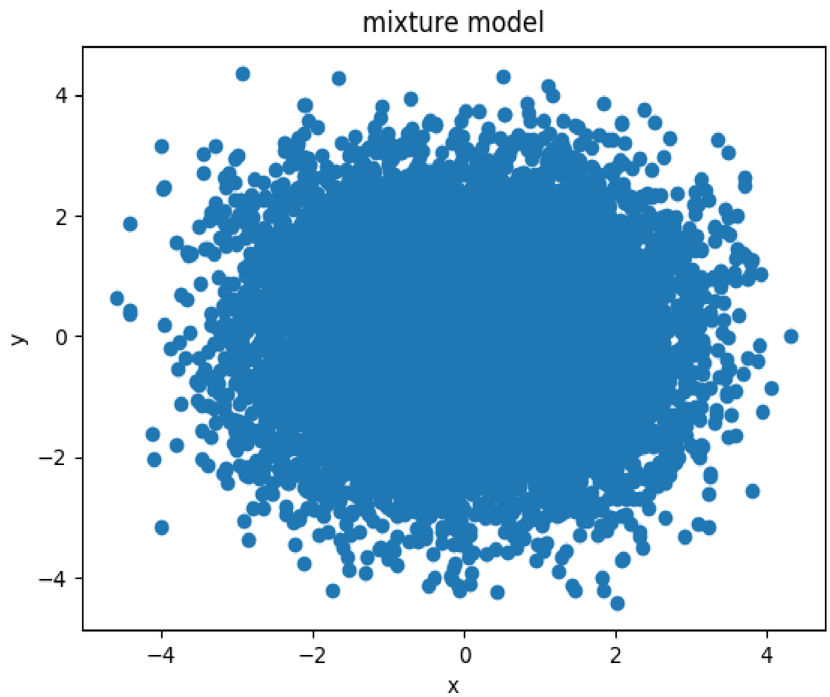
(2) Univariate Normal Distribution



(3) Multivariate Normal Distribution



(4) Mixture Distribution



Estimate the probability is 0.1756

Question2: Prove that the sum of two independent Poisson random variables is also a Poisson random variable.

Prove:

Given two independent Poisson random variables X1~P(λ1)) and X2~P(λ2), we can see that

$$P\left(X_1=k\right) = \frac{e^{-\lambda_1}\lambda^k{}_1}{k!} \quad \text{and} \quad P\left(X_2=j\right) = \frac{e^{-\lambda_2}\lambda^j{}_2}{j!}$$

So for $X = X_1 + X_2,$

$$P(X=l) = \sum_{k=0}^{l}\left(P\left(X_1=k\right)P\left(X_2=l-k\right)\right)$$

$$= e^{-\left(\lambda_1+\lambda_2\right)}\sum_{k=0}^{l}\frac{\lambda^k{}_1}{k!}\frac{\lambda^{l-k}{}_2}{(l-k)!} = \frac{e^{-\left(\lambda_1+\lambda_2\right)}}{l!}\sum_{k=0}^{l}C_l^k\lambda_1^k\lambda_2^{l-k}$$

$$= \frac{e^{-\left(\lambda_1+\lambda_2\right)}}{l!}\left(\lambda_1+\lambda_2\right)^l$$

So X=(X1+X2)~P(λ1+λ2), X is also a Poisson random variable.

Question3: Write down expressions for these quantities in terms of α0, α, μ0, σ0 and σ.

Prove:

$$P(X_1=x_1, X_0=x_0) = P(X_1=x_1 \mid X_0=x_0)P(X_0=x_0)$$

According to marginal probability:

$$P(X_1=x_1) = \int P(X_1=x_1, X_0=x_0)\,dx_0$$

$$= a_0 a_1 \int e^{-\frac{1}{2}\left(\frac{(x_0-\mu_0)^2}{\sigma_0^2} + \frac{(x_1-x_0)^2}{\sigma^2}\right)} dx_0$$

$$= a_0 a_1 \int e^{-\frac{1}{2}\left(\frac{\sigma^2(x_0-\mu_0)^2 + \sigma_0^2(x_1-x_0)^2}{\sigma_0^2\sigma^2}\right)} dx_0$$

$$= \frac{a_0 a}{A} e^{\left(-\frac{1}{2}\frac{(x_1-\mu_0)^2}{\sigma^2+\sigma_0^2}\right)}.$$

where A is a constant.

So:

$$p(X_1=x_1) = a_1 e^{\left(-\frac{1}{2}\frac{(x_1-\mu_1)^2}{\sigma_1^2}\right)}$$

$\mu_1 = \mu_0.$

$\sigma_1^2 = \sigma^2 + \sigma_0^2$

$$a_1 = \frac{a_0 a}{A} = a_0 a \int e^{-\frac{1}{2}\left(\frac{x_0^2 - 2\left(\frac{\sigma^2\mu_0+\sigma_0^2 x_1}{\sigma^2+\sigma_0^2}\right)x_0 + \left(\frac{\sigma^2\mu_0+\sigma_0^2 x_1}{\sigma^2+\sigma_0^2}\right)^2}{\frac{\sigma_0^2\sigma^2}{\sigma^2+\sigma_0^2}}\right)} dx_0$$

Question4: Find the eigenvalues and eigenvectors of the following 2×2 matrix A.

The eigenvalue λ of the given matrix is

$$(0-\lambda)\cdot(-3-\lambda)-(-2\cdot 1) = 0$$

$$so\ the\ eigenvalue\ is\ -2\ and\ -1,$$

$$the\ eigenvector\ with\ -2\ is \begin{bmatrix} 1 \\ -2 \end{bmatrix} and\ the\ eigenvector\ with\ -1\ is \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Question5: Provide one example for each of the following cases, where A and B are 2 × 2 matrices.

$$for\quad (A+B)^2 != A^2 + 2AB + B^2,\ we\ have\ A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} and\ B = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}.$$

$$for\ AB = 0,\ A != 0,\ B != 0,\ we\ have\ A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} and\ B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Question6: Let u denote a real vector normalized to unit length. That is, uTu = 1. Show that A is orthogonal, i.e., ATA = 1.

Prove:

$$A^TA = (I - 2uu^T)^T(I - 2uu^T)$$

$$given\ u\ is\ a\ real\ vector,$$

$$(uu^T)^T = uu^T\ and,\ (I - 2uu^T)^T = (I - 2uu^T)$$

$$\begin{aligned} so\ A^TA &= I^2 - 4uu^T + 4uu^Tuu^T \\ &= I - 4uu^T + 4uu^T \\ &= I \end{aligned}$$

Question7：prove the following assertions.

Prove:

Question 7 ① Prove $f(x) = e^x$ is convex for $x \in \mathbb{R}$.

~~Question~~ 7: $(e^x)' = e^x > 0$, $(e^x)'' = e^x > 0$, so e^x is convex

② prove $f(x_1, x_2) = \max(x_1, x_2)$ is convex on $\mathbb{R}^2$

$\max(\lambda x_1 + (1-\lambda)y_1,\ \lambda x_2 + (1-\lambda)y_2) \le \max(\lambda x_1, \lambda x_2) + \max((1-\lambda)y_1, (1-\lambda)y_2)$

So $\max(x_1, x_2)$ is convex

③ f, g are convex, then $\max(f, g)$ is convex on S

prove: $f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y)$
$g(\lambda x + (1-\lambda)y) \le \lambda g(x) + (1-\lambda) g(y)$. for $x, y \in S$

$\max(f(\lambda x + (1-\lambda)y),\ g(\lambda x + (1-\lambda)y))$.

$\le \max(\lambda f(x) + (1-\lambda) f(y),\ \lambda g(x) + (1-\lambda) g(y))$

$\le \lambda \max(f(x), g(x)) + (1-\lambda) \max(f(y) + g(y))$

So $\max(f, g)$ is convex

④ f, g are convex, non-negative and same minimum point, the fg is convex.

prove: f, g are convex, then

$$(fg)'' = (f'g + fg')'$$
$$= f''g + f'g' + f'g' + fg''$$

we know that $f'' \geqslant 0$, $g'' \geqslant 0$. $f \geqslant 0$, $g \geqslant 0$

and then we know that f, g share same minimum, x_{min}

$f'(x_{min}) = 0$. $g'(x_{min}) = 0$. for $x \le x_{min}$ $f'(x) < 0$. $g'(x) < 0$

$x \geqslant x_{min}$ $f'(x) > 0$. $g'(x) > 0$

So. $f'g' \geqslant 0$.

$(fg)'' \geqslant 0$. So fg is convex

Question8: Using the method of Lagrange multipliers, find the categorical distribution that has the highest entropy.

Prove:

Question 8.

given entropy $H(p) = -\sum_{i=1}^{k} p_i \log(p_i)$, also $\sum_{i=1}^{k} p_i = 1$

constains $g(p) = \sum_{i=1}^{k} p_i - 1$

So in Lagrange Multipliers,

$$\varphi(p, \lambda) = H(p) + \lambda g(p)$$

$$\frac{\partial}{\partial p_i} \left(-\sum_{i=1}^{k} p_i \log(p_i) + \lambda \left(\sum_{i=1}^{k} p_i - 1 \right) \right) = 0$$

I don't know about this.. isn't it $\log_n p_i$? what is n?
~~to~~ I don't know what the base of this logarithm.
Set it as n, and n is a constant

$$-\left(\frac{1}{\ln n} + \log_n p_i \right) + \lambda = 0 \quad i = 1, 2, \dots, k$$

We can See that p_i is a constant and $p_1 = p_2 = p_3 \cdots = p_k$.

~~So~~ According to $\sum_{i=1}^{k} p_i = 1$, we can get that $p_i = \frac{1}{k}$,

So the uniform distribution has the ~~high~~ highest entropy.

Problem 1:

Question1: Show that $J(\theta)$ can be written in the form, for an appropriate diagonal matrix W, where X is the m×d input matrix and y is a m×1 vector denoting the associated outputs. State clearly what W is.

Problem 1.

$$\begin{bmatrix} -x^{(1)}- \\ -x^{(2)}- \\ \vdots \\ -x^{(m)}- \end{bmatrix}$$

Question 1.

X is a m×d matrix, and y is a m×1 matrix.

Set X as $\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & \cdots & & \vdots \\ \vdots & & & x_{md} \end{bmatrix}$ y as $\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$

Set W as $\begin{bmatrix} w_{11} & 0 & 0 & \cdots & 0 \\ 0 & w_{22} & \cdots & & 0 \\ \vdots & & \cdots & & \\ 0 & & \cdots & & w_{mm} \end{bmatrix}$, θ as $\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix}$

So

$$X\theta - y = \begin{bmatrix} \sum_{j=1}^{d} x_{1j}\theta_1 \\ \vdots \\ \sum_{j=1}^{d} x_{mj}\theta_j \end{bmatrix}$$

$x^{(i)} = [x_{i1}, x_{i2}, x_{i3} \cdots x_{id}]$

$y^{(i)} = y_i$

So $(X\theta - y)^T W (X\theta - y)$

$$= \sum_{i=1}^{m}\left[\sum_{j=1}^{d} x_{ij}\theta_i \cdot w_{ii} \sum_{j=1}^{d} x_{ij}\theta_i\right] = \sum_{i=1}^{m} w_{ii}(\theta^T x^{(i)} - y^{(i)})^2$$

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

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

So

W can be denoted as $\begin{bmatrix} \frac{1}{2}w^{(1)} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{2}w^{(2)} & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & 0 & \cdots & \frac{1}{2}w^{(m)} \end{bmatrix}$

Question2: By computing the derivative of the weighted J(θ) and setting it equal to zero, generalize the normal equation to the weighted setting and solve for θ in closed form in terms of W, X and y.
Prove:

$$If\ all\ w^{(i)}\ is\ 1,\ then,$$

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

$$J'(\theta) = 2X^TX\theta - 2X^Ty$$

$$set\ J'(\theta) = 0,$$

$$so\ X^TX\theta = X^Ty$$

$$for\ the\ normal\ equation:$$

$$J(\theta) = (X\theta - y)^TW(X\theta - y)$$

$$= \theta^TX^TWX\theta - 2\theta^TX^TWy + y^TWy$$

$$J'(\theta) = 2X^TWX\theta - 2X^TWy$$

$$set\ J'(\theta) = 0,$$

$$X^TWX\theta = X^TWy$$

$$\theta = (X^TWX)^{-1}X^TWy$$

Question3: 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?

$$\frac{\partial}{\partial\theta} = \frac{1}{m}\sum_{i=1}^{m} w^{(i)}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x^{(i)}$$

$$The\ batch\ gradient\ descent\ algorithm\ is:$$

$$initialize\ \theta\ randomly$$

$$while\ the\ loss\ is\ not\ coverage:$$

$$for\ every\ j:$$

$$\theta_j = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} w^{(i)} \left(h_\theta (x^{(i)}) - y^{(i)} \right) x^{(j)}$$

$$where\ \alpha\ is\ learning\ rate.$$

m-> n, random batch!

(xj -> xji)

So the locally weighted LR is a non-parametric method.

Problem 3

Problem 3.1.A

Problem 3.1.A2:

Figure 2: Linear Regression Model

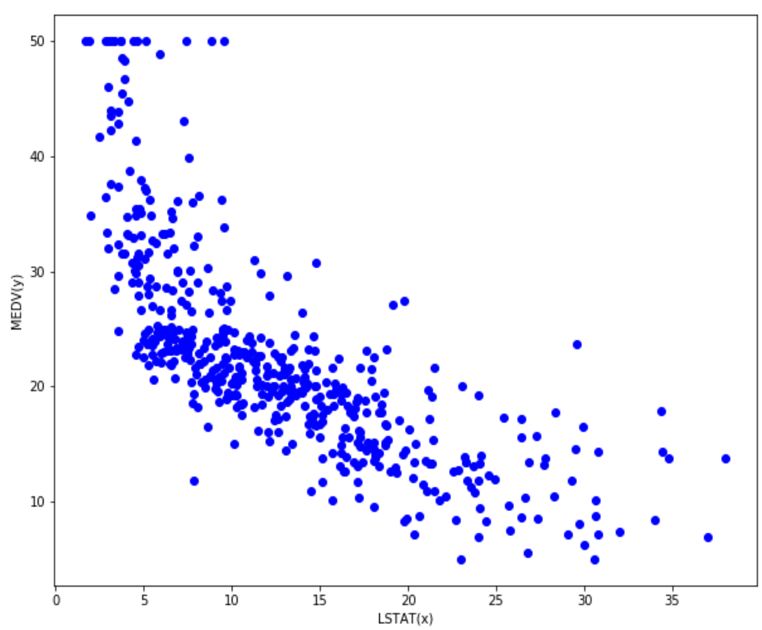
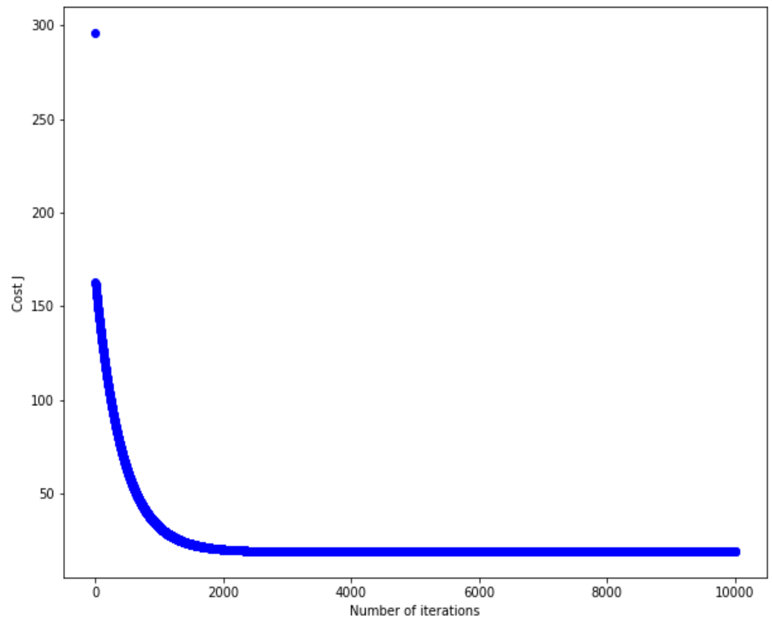


Figure 3: The plot of the J(θ) values during gradient descent



Training Loss:

```
iteration 0 / 10000: loss 296.073458
iteration 1000 / 10000: loss 32.190429
iteration 2000 / 10000: loss 20.410446
iteration 3000 / 10000: loss 19.347011
iteration 4000 / 10000: loss 19.251010
iteration 5000 / 10000: loss 19.242344
iteration 6000 / 10000: loss 19.241561
iteration 7000 / 10000: loss 19.241491
iteration 8000 / 10000: loss 19.241484
iteration 9000 / 10000: loss 19.241484
Theta found by gradient_descent: [34.55363411 -0.95003694]
```

Problem 3.1.A3:

For lower status percentage = 5, we predict a median home value of 298034.49
For lower status percentage = 50, we predict a median home value of -129482.13

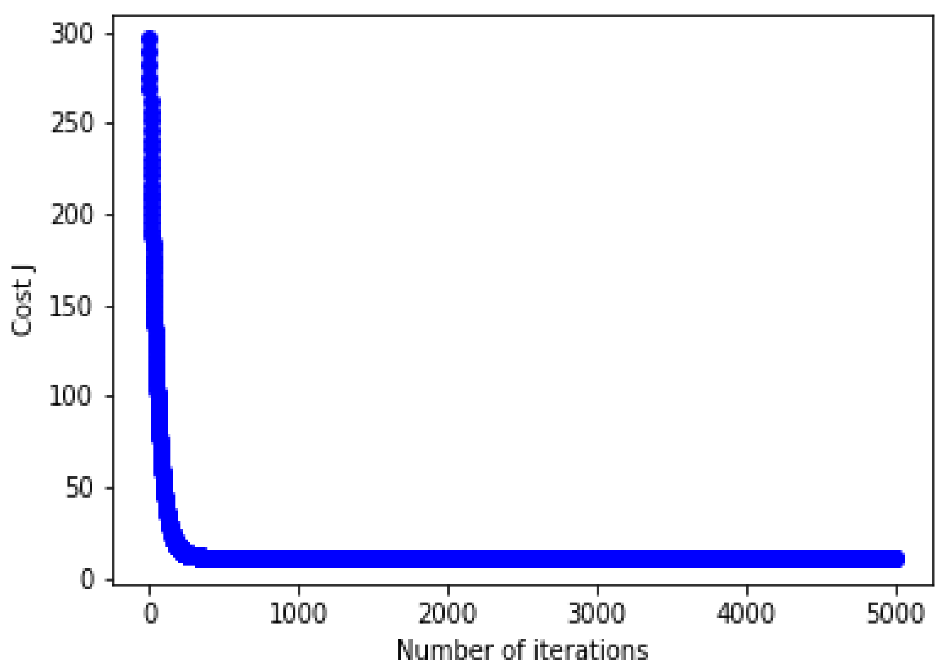
Assessing model quality

5 fold cross_validation MSE = 42.62
5 fold cross_validation r_squared = 0.30

Problem 3.1.B:

Problem 3.1.B2:

Figure 5: Convergence of gradient descent for linear regression with multiple variables (Boston housing data set)



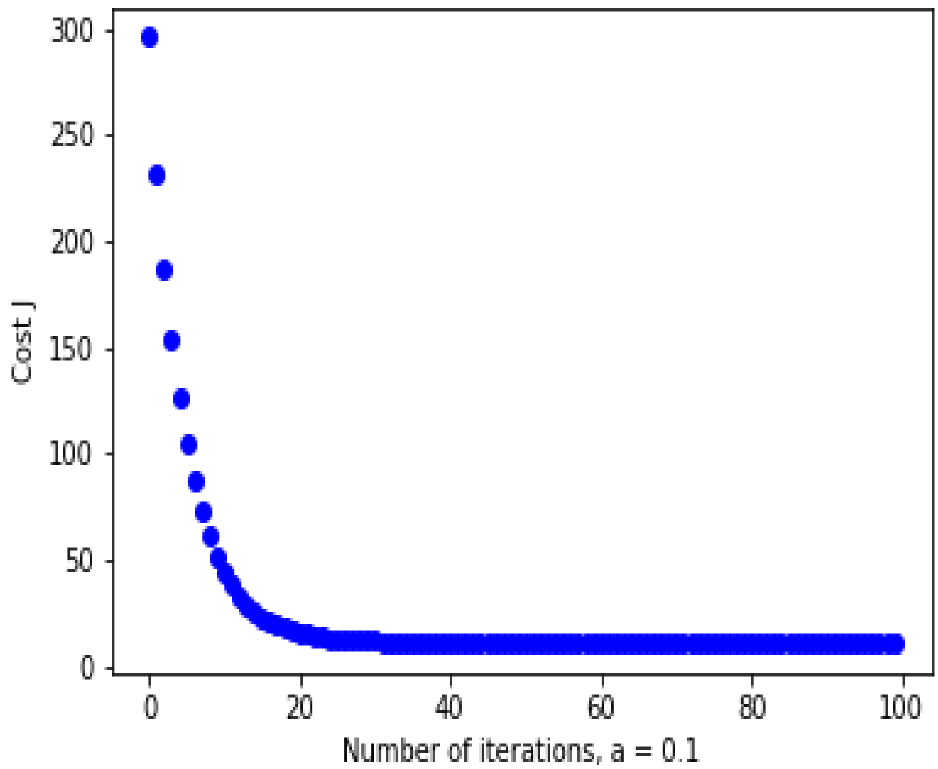
Problem 3.1.B3:

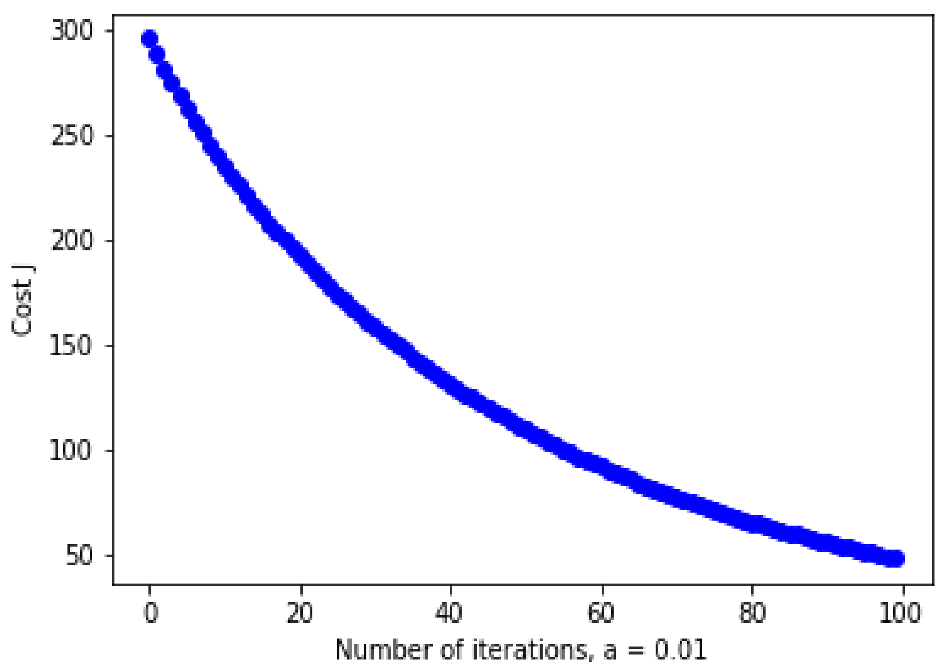
For average home in Boston suburbs, we predict a median home value of 225328.06

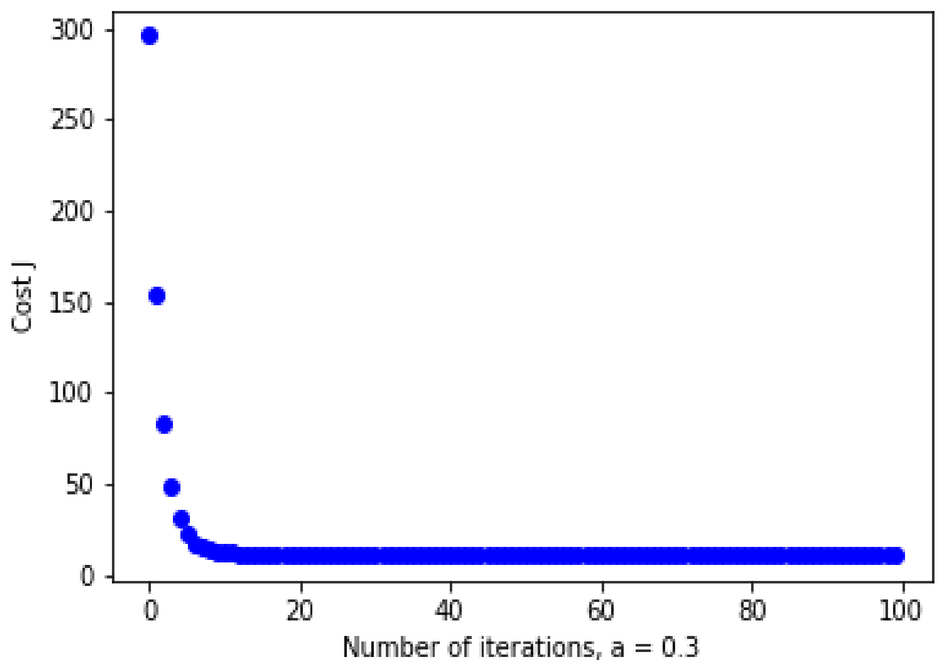
Problem 3.1.B4:

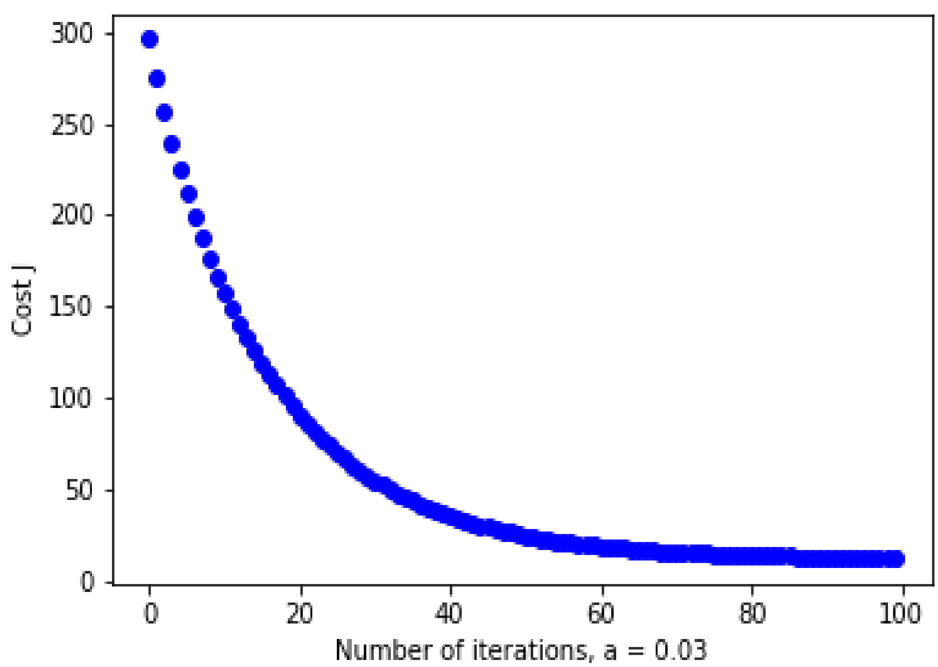
For average home in Boston suburbs, we predict a median home value using normal equation is 225328.06, which matches up with gradient descent method.

Problem 3.1.B5







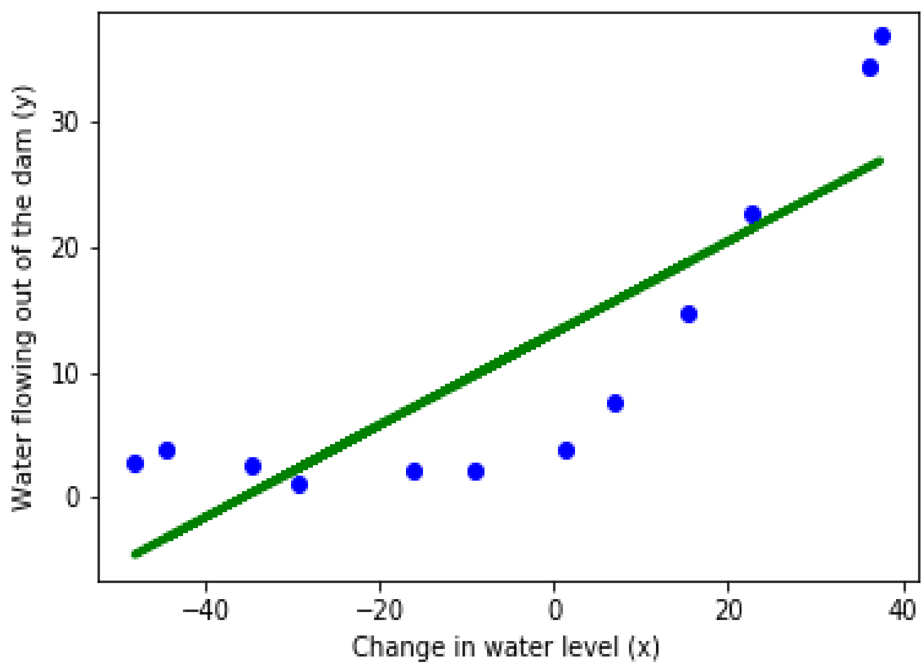


Loss converges slow when learning rate is 0.01 and 0.03. When learning rate increases to 0.3, loss function converges within 10 iterations, which is not we want.
So I suggest the learning rate should be less than 0.1 and greater than 0.03, the number of iterations should be 60 to 80.

Problem 3.2:

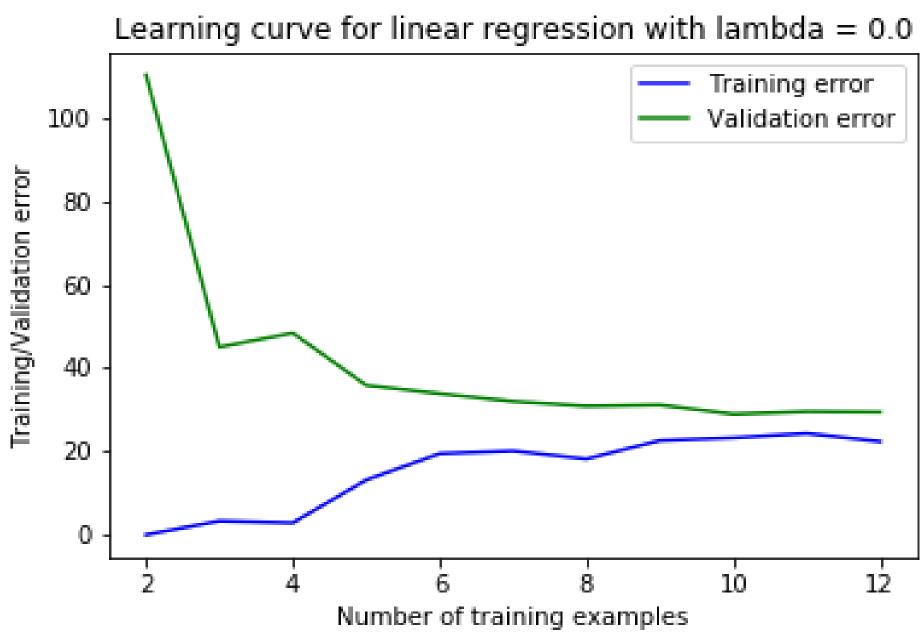
Problem 3.2.A2

The best fit line for the training data



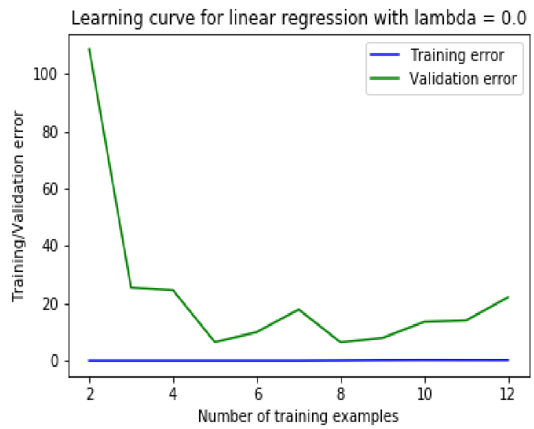
Problem 3.2.A3:

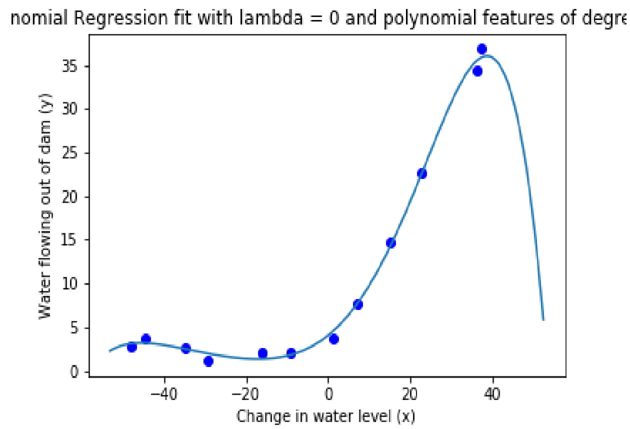
Learning Curve:



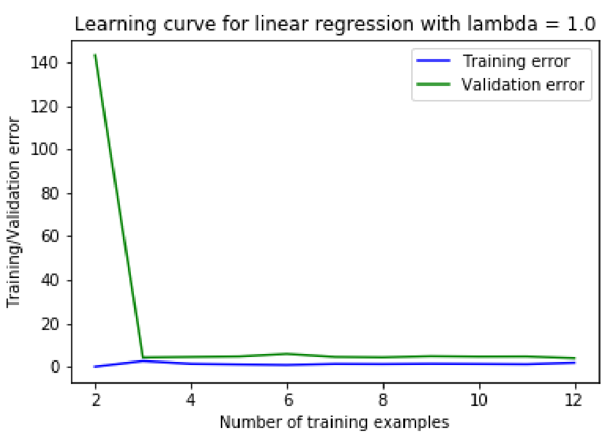
Problem 3.2.A4:

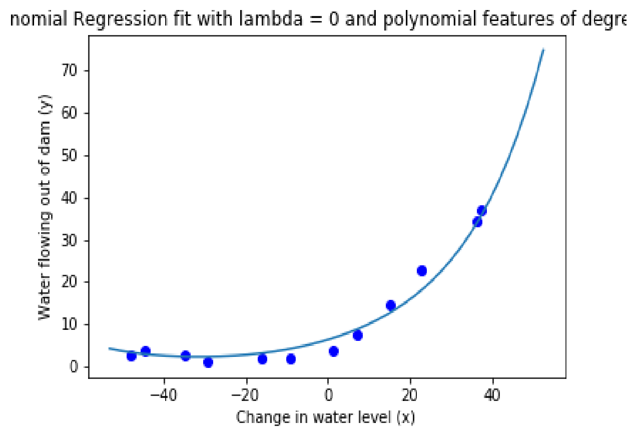
Lambda = 0



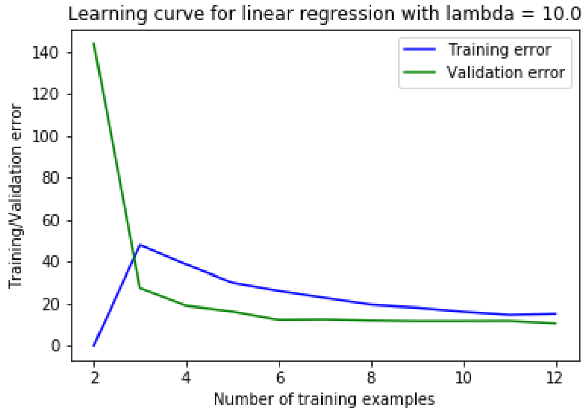


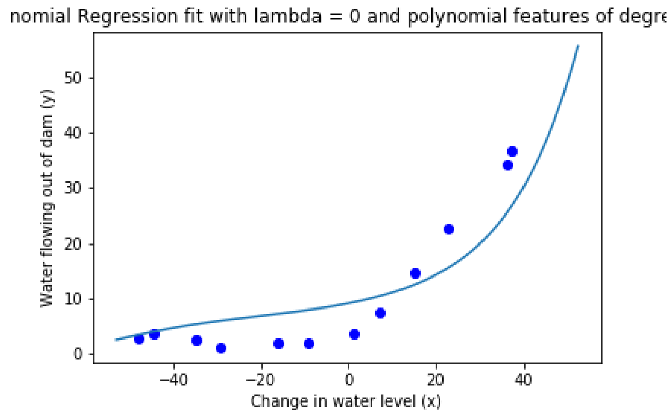
Lambda = 1



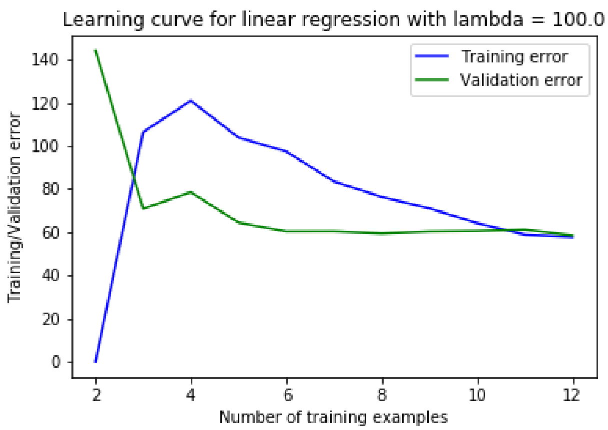


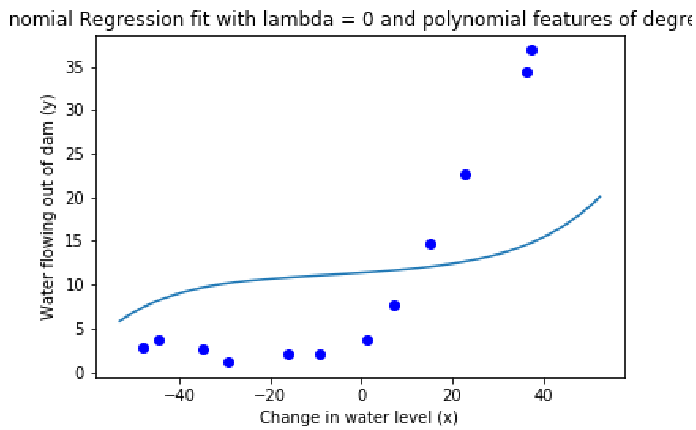
Lambda = 10





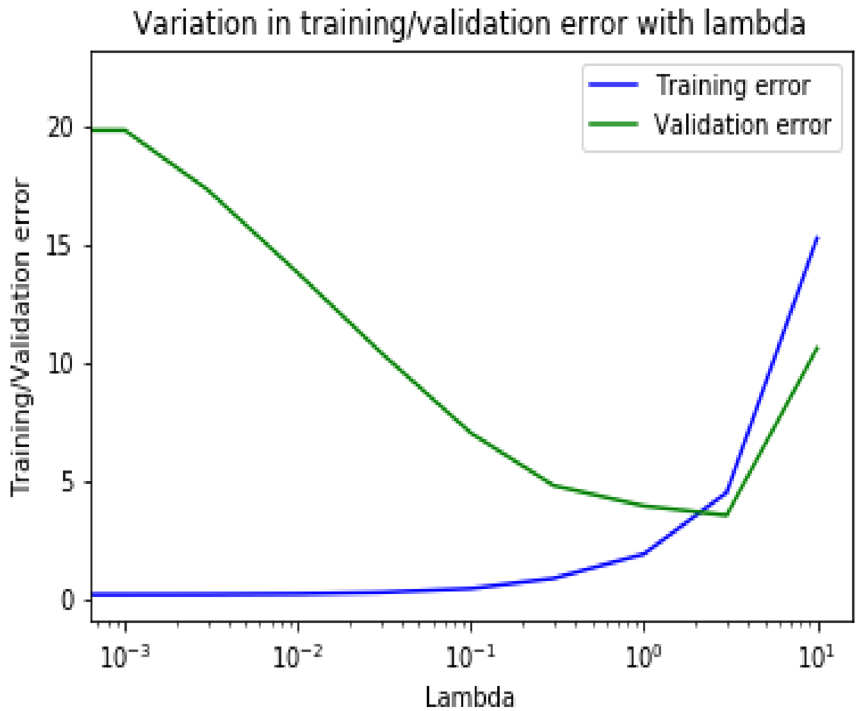
Lambda = 100





When lambda is too small, the penalty is not obvious thus the model is still over-fitting on validation process, When lambda is to large, the effect of penalty is overwhelming and theta is close to 0 therefore the model is under-fitting

Problem 3.2.A5:



The best value of lambda is 1, since there is a good balance between validation error and training error, neither over-fitting nor under-fitting.

Problem 3.2.A6

When lambda is 1, the best test error is 3.098748265552584

Problem 3.2.A7