COMP 540 Homework 1

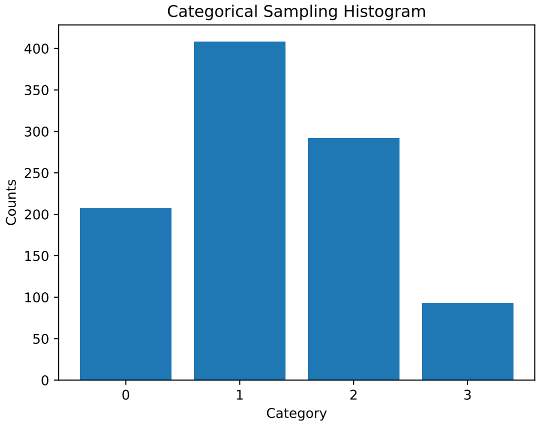
Tony Ren, Hanyang (Quentin) Li

January 21, 2018

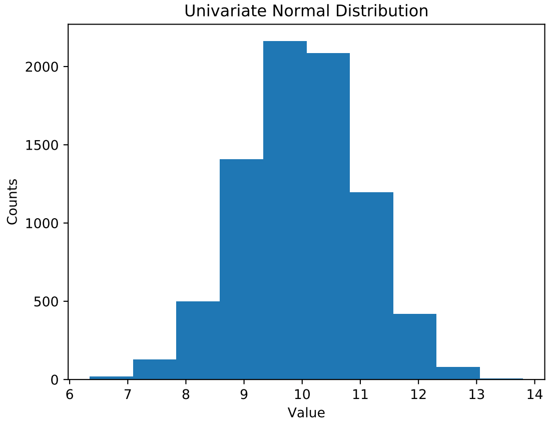
Problem 0

Problem 0 Part 1 -Sampler

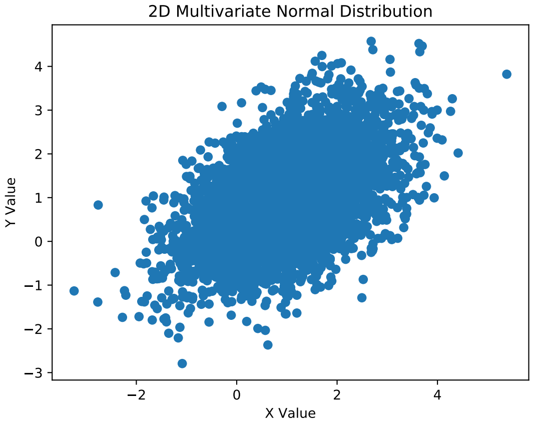
(1) Categorical Distribution



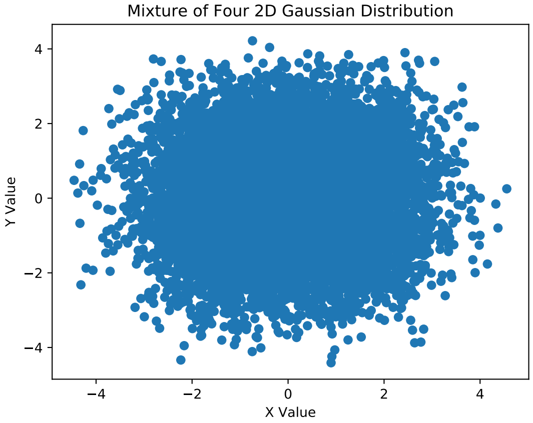
(2) Univariate Normal Distribution



(3) Multivariate Normal Distribution



(4)Mixture Distribution



The probability that a sample from this distribution lies within the unit circle centered at (0.1; 0.2) is approximately 0.1826, based on a sampling of 10000 data.

Answers to Problem 0 part 2 - part 8 are hand-written on paper. The scanned copies are attached on next page.

Tony Ren
Hanyang (Quentin) Li

COMP540 HW1

Problem 0

2. Let X and Y be two independent Poisson.

$$\cancel{X=} \quad P(X=i) = \cancel{e}\frac{\mu^i}{i!}e^{-\mu} \qquad P(Y=j) = \frac{\lambda^j}{j!}e^{-\lambda}$$

$$P(X+Y=k) = \sum_{i=0}^{k} P(X+Y=k, X=i)$$

$$= \sum_{i=0}^{k} P(Y=k-i, X=i)$$

$$= \sum_{i=0}^{k} P(Y=k-i \times P(X=i))$$

$$= \sum_{i=0}^{k} e^{-\mu}\frac{\mu^{k-i}}{(k-i)!}e^{-\lambda}\frac{\lambda^i}{i!}$$

$$= e^{-(\mu+\lambda)}\frac{1}{k!}\sum_{i=0}^{k}\frac{k!}{i!(k-i)!}\mu^{k-i}\lambda^i$$

$$= e^{-(\mu+\lambda)}\frac{1}{k!}\sum_{i=0}^{k}\binom{k}{i}\mu^{k-i}\lambda^i$$

$$= \frac{(\mu+\lambda)^k}{k!}\cdot e^{-(\mu+\lambda)}$$

Therefore, $X+Y \sim P(\mu+\lambda)$ => a Poisson with rate $\mu+\lambda$

3. Convert exponentials of arbitrary quadratic forms into exponentials of the standard Gaussian form.

$$ax^2+bx+c = a(x^2+\frac{b}{a}x+\frac{c}{a}) = a[(x-\frac{-b}{2a})^2-\frac{b^2}{4a^2}+\frac{c}{a}].$$

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

$$= a_0a_1\int \exp\left(-\frac{1}{2}\frac{(x_0-\mu_0)^2}{\sigma_0^2}+\frac{(x_1-x_0)^2}{\sigma^2}\right)dx_0$$

$$= a_0a_1\int \exp\left(-\frac{1}{2}\frac{\sigma^2(x_0-\mu_0)^2+\sigma_0^2(x_1-x_0)^2}{\sigma_0^2\sigma^2}\right)dx_0$$

$$= a_0a_1\int \exp\left(-\frac{1}{2}\frac{(\sigma^2+\sigma_0^2)x_0^2-2(\sigma^2\mu_0+\sigma_0^2x_1)x_0+\sigma^2\mu_0^2+\sigma_0^2x_1^2}{\sigma_0^2\sigma^2}\right)dx_0$$

Continue

$\boxed{\text{Cont'd}}$

$$P(X_1=x_1) = a_0 a_1 \int \exp\left(-\frac{1}{2}\frac{x_0^2 - 2\left(\frac{\sigma^2\mu_0+\sigma_0^2 x_1}{\sigma^2+\sigma_0^2}\right)x_0 + \left(\frac{\sigma^2\mu_0+\sigma_0^2 x_1}{\sigma^2+\sigma_0^2}\right)^2}{\sigma_0^2\sigma^2/(\sigma^2+\sigma_0^2)}\right)$$

$$\cdot \exp\left(-\frac{1}{2}\frac{\sigma^2\mu_0^2+\sigma_0^2 x_1^2 - \frac{(\sigma^2\mu_0+\sigma_0^2 x_1)^2}{\sigma^2+\sigma_0^2}}{\sigma_0^2\cdot\sigma^2}\right)dx_0$$

~~$P = \frac{a_0 a_1}{a}$~~

Suppose the normalization constant is a'.

$$P(X_1=x_1) = \frac{a_0 a_1}{a'}\exp\left(-\frac{1}{2}\frac{(\sigma^2\mu_0^2+\sigma_0^2 x_1^2)(\sigma^2+\sigma_0^2) - (\sigma^2\mu_0+\sigma_0^2 x_1)^2}{\sigma_0^2\sigma^2(\sigma^2+\sigma_0^2)}\right)$$

$$= \frac{a_0 a_1}{a'}\exp\left(-\frac{1}{2}\frac{|\sigma_0^2(\sigma^2+\sigma_0^2)-\sigma_0^4|x_1^2 - 2\sigma^2\sigma_0^2\mu_0 x_1 + \sigma^2\mu_0^2(\sigma^2+\sigma_0^2)-\sigma^4\mu_0^2}{\sigma_0^2\sigma^2(\sigma^2+\sigma_0^2)}\right)$$

$$= \frac{a_0 a_1}{a'}\exp\left(-\frac{1}{2}\frac{x_1^2-2\mu_0 x_1+\mu_0^2}{\sigma^2+\sigma_0^2}\right)$$

$$= \frac{a_0 a_1}{a'}\exp\left(-\frac{1}{2}\cdot\frac{(x_1-\mu_0)^2}{\sigma^2+\sigma_0^2}\right)$$

$\boxed{\mu_1 = \mu_0}$ $\boxed{\sigma_1 = \sqrt{\sigma^2+\sigma_0^2}}$

$\frac{a_0 a_1}{a'} = a \iff \boxed{a_1 = \frac{a' a}{a_0}}$

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

Let $x = \begin{pmatrix} a \\ b \end{pmatrix}$ be A's eigen vector, and λ be A's eigen value.

Then, $Ax = \lambda x$

$\Rightarrow \begin{pmatrix} 13 & 5 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \lambda\begin{pmatrix} a \\ b \end{pmatrix}$

$$\begin{cases} 13a+5b = \lambda a & ① \\ 2a+4b = \lambda b & ② \end{cases}$$

From ②, we obtain $a = \frac{(\lambda-4)b}{2}$

Plug into ①, we obtain $\lambda^2 - 17\lambda + 42 = 0$

$(\lambda-3)(\lambda-14) = 0$

Therefore, $\boxed{\lambda_1 = 3}$, $a = -\frac{1}{2}b \Rightarrow \boxed{x_1 = \begin{pmatrix} \sqrt{\frac{1}{5}} \\ -\sqrt{\frac{4}{5}} \end{pmatrix}}$

$\boxed{\lambda_2 = 14}$, $a = 5b \Rightarrow \boxed{x_2 = \left(\sqrt{\frac{25}{26}}, -\sqrt{\frac{1}{26}}\right)^T}$

5. (1) $(A+B)^2 \neq A^2+2AB+B^2$

example: $\boxed{A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}}$

$$(A+B)^2 = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}$$

$$A^2+2AB+B^2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix}$$

(2) $AB = 0, \quad A \neq 0, \quad B \neq 0$

example: $\boxed{A = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}}$

6. $u^T u = 1$ show $A = I - 2uu^T$ is orthogonal.

$$\begin{aligned} A^T A &= (I-2uu^T)^T(I-2uu^T) \\ &= (I-2uu^T)(I-2uu^T) \\ &= I - 4uu^T + 4(uu^T)(uu^T) \\ &= I - 4uu^T + 4u(u^Tu)u^T \\ &= I - 4uu^T + 4uu^T \\ &= I \end{aligned}$$

Thus, A is orthogonal

7.

(1) $f(x) = x^3$

$f''(x) = 6x \geq 0$ for $x \geq 0$

Therefore, $\boxed{f(x) = x^3 \text{ is convex for } x \geq 0.}$

(2) $\vec{x} = (x_1, x_2)$, $\vec{y} = (y_1, y_2)$

$$\begin{aligned} f(\lambda \vec{x} + (1-\lambda)\vec{y}) &= f(\lambda x_1 + (1-\lambda)y_1, \lambda x_2 + (1-\lambda)y_2) \\ &= \max(\lambda x_1 + (1-\lambda)y_1, \lambda x_2 + (1-\lambda)y_2) \\ &\leq \max(\lambda x_1, \lambda x_2) + \max((1-\lambda)y_1, (1-\lambda)y_2) \\ &= \lambda \max(x_1, x_2) + (1-\lambda)\max(y_1, y_2) \\ &= \lambda \cdot f(\vec{x}) + (1-\lambda) \cdot f(\vec{y}) \end{aligned}$$

Thus, $f(\lambda \vec{x} + (1-\lambda)\vec{y}) \leq \lambda f(\vec{x}) + (1-\lambda) f(\vec{y})$

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

(3) $(f+g)''(x) = f''(x) + g''(x)$

Since f and g are both convex on S and univariate,

~~$f''(x) = g''(x)$~~

$f''(x) \geq 0$, $g''(x) \geq 0$

Thus, $(f+g)''(x) = f''(x) + g''(x) \geq 0$

$\boxed{f+g \text{ is convex on } S}$.

(4) $(f \cdot g)''(x) = (f' \cdot g + g'f)'(x) = (f'' \cdot g + \overset{2f'g'}{\cancel{2f''g''}} + fg'')(x)$

Since f and g are convex and non-negative,

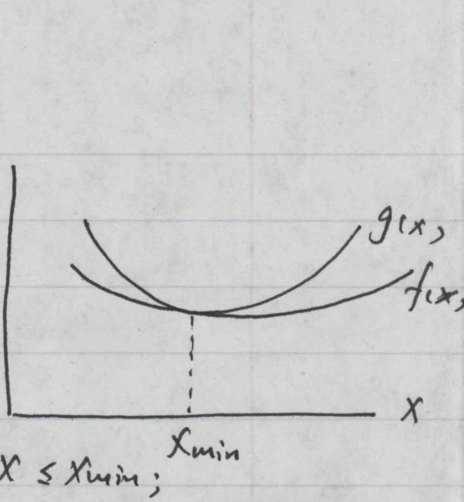
$f(x) \geq 0$, $g(x) \geq 0$, $f''(x) \geq 0$, $g''(x) \geq 0$

Therefore, $(f''g)(x) \geq 0$, $(fg'')(x) \geq 0$

$\boxed{\text{Continue}}$

Cont'd

f and g are convex.



Let the common minimum point of f and g occurs at $x = X_{min}$.

Since f and g are convex,

$f'(x)$ and $g'(x)$ both ≤ 0 when $x \le X_{min}$;

Similarly, $f'(x)$ and $g'(x)$ both ≥ 0 when $x \ge X_{min}$.

Thus, $f'g'(x) \ge 0$ on the given set.

Thus, $(fg)''(x) = (f''g + 2f'g' + fg'')(x) \ge 0$

$\iff$ $\boxed{fg \text{ is convex on the same set.}}$

8. $H(X) = -\sum_{i=1}^{K} p_i \log(p_i)$

$Pr(X = x_i) = p_i$ for $i = 1, 2, \cdots, K$

Constraint of the optimization of entropy h:

$$\sum_{i=1}^{K} p_i = 1 \iff \sum_{i=1}^{K} p_i - 1 = 0 \quad ①$$

$$p_i \ge 0 \quad ②$$

Form Lagrangian:

$$J(p) = -\sum_{i=1}^{K} p_i \log(p_i) + \lambda\left(\sum_{i=1}^{K} p_i - 1\right)$$

$$\frac{\partial J}{\partial p_i} = -\log p_i - 1 + \lambda = 0$$

$$p_i^* = e^{\lambda - 1}$$

$\Rightarrow$ All pi's for $i = 1, 2, \cdots, k$ are equal.

Thus, the Categorical Distribution with maximum entropy has K pi's, where $\boxed{\text{all } p_i\text{'s} = \frac{1}{k}}$

Problem 1

Problem 1 Part 1

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

For matrix W =

$$\begin{bmatrix} 1/2w^1 & 0 & \dots & 0 \\ 0 & 1/2w^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/2w^i \end{bmatrix}$$

and X is the $m \times d$ input matrix, y is a $m \times 1$ output vector, and θ is a $d \times 1$ model variable vector

$(X\theta - y) =$

$$\begin{bmatrix} (\theta_1 x_1^{(1)} + \theta_2 x_2^{(1)} + \dots + \theta_D x_D^{(1)}) - y^{(1)} \\ (\theta_1 x_1^{(2)} + \theta_2 x_2^{(2)} + \dots + \theta_D x_D^{(2)}) - y^{(2)} \\ \vdots \\ (\theta_1 x_1^{(m)} + \theta_2 x_2^{(m)} + \dots + \theta_D x_D^{(m)}) - y^{(m)} \end{bmatrix}$$

$W(X\theta - y) =$

$$\begin{bmatrix} 1/2w^{(1)} \times ((\theta_1 x_1^{(1)} + \theta_2 x_2^{(1)} + \dots + \theta_D x_D^{(1)}) - y^{(1)}) \\ 1/2w^{(2)} \times ((\theta_1 x_1^{(2)} + \theta_2 x_2^{(2)} + \dots + \theta_D x_D^{(2)}) - y^{(2)}) \\ \vdots \\ 1/2w^{(m)} \times ((\theta_1 x_1^{(m)} + \theta_2 x_2^{(m)} + \dots + \theta_D x_D^{(m)}) - y^{(m)}) \end{bmatrix}$$

$(X\theta - y)^T W(X\theta - y) =$
$1/2w^{(1)} \times ((\theta_1 x_1^{(1)} + \theta_2 x_2^{(1)} + \dots + \theta_D x_D^{(1)}) - y^{(1)})^2$
$+ 1/2w^{(2)} \times ((\theta_1 x_1^{(2)} + \theta_2 x_2^{(2)} + \dots + \theta_D x_D^{(2)}) - y^{(2)})^2$
$+ \dots$
$+ 1/2w^{(m)} \times ((\theta_1 x_1^{(m)} + \theta_2 x_2^{(m)} + \dots + \theta_D x_D^{(m)}) - y^{(m)})^2$

Convsersely, collecting all of the sums over the ith term for $J(\theta)$ results in this

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

$=$
$1/2w^{(1)} \times ((\theta_1 x_1^{(1)} + \theta_2 x_2^{(1)} + \dots + \theta_D x_D^{(1)}) - y^{(1)})^2$
$+ 1/2w^{(2)} \times ((\theta_1 x_1^{(2)} + \theta_2 x_2^{(2)} + \dots + \theta_D x_D^{(2)}) - y^{(2)})^2$
$+ \dots$
$+ 1/2w^{(m)} \times ((\theta_1 x_1^{(m)} + \theta_2 x_2^{(m)} + \dots + \theta_D x_D^{(m)}) - y^{(m)})^2$

These two are equivalent, therefore

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

When W =

$$\begin{bmatrix} 1/2w^1 & 0 & 0 & \dots & 0 \\ 0 & 1/2w^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1/2w^i \end{bmatrix}$$

Problem 1 Part 2

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

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

$= (X\theta)^T WX\theta - y^T WX\theta - (X\theta)^T Wy + y^T Wy$

$= \theta^T X^T WX\theta - 2(X\theta)^T Wy + y^T Wy$

$\Delta_\theta J(\theta) = 2X^T WX\theta - 2X^T Wy = 0$
$2X^T WX\theta = 2X^T Wy$

$X^T WX\theta = X^T Wy$

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

Problem 1 Part 3

Algorithm for batch gradient descent (calculating predicted y for each x)
1. Load training data and set numIters for each gradient descent

for i in range(1,numNewDataPoints):

2. Calculate weights for one data point utilizing formula $w^{(i)} = exp(-(x - x^{(i)})^T (x - x^{(i)})/2\tau^2)$

3. Initialize guess of θ

for j in range(1,numIters):

4. Calculate cost and gradient using the data, weights, and θ

5. Use gradient and learning rate to adjust all θ values at once

6. Repeat these steps 2-4 until numIters is up or cost is sufficiently small

7. Use the θ calculated to create linear model and give an output prediction y for the current x point of interest

8. Repeat for all points in the new dataset(numNewDataPoints)

The locally weighted linear regression is still a non-parametric method because it has a flexible number of parameters that grow with the size of the training data due to the local weight formula. The regression must keep the training data to use it flexibly and calculate the weights each time a prediction is made with the algorithm. It has a parametric method embedded within(the linear regression) but as it performs linear regression locally for each data point, it should be considered a non-parametric method

Problem 2

Problem 2 Part 1

$E[\theta] = \theta^*$
$\theta = (X^TX)^{-1}X^Ty$ from the normal equation
$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)}$
The true parameter vector θ^* is θ^T when we estimate the values of y with x $\theta = (X^TX)^{-1}X^T(\theta^*X + \epsilon)$
$\theta = (X^TX)^{-1}(X^TX\theta^* + X^T\epsilon)$
$\theta = \theta^* + (X^TX)^{-1}X^T\epsilon$
$E(\theta) = \theta^* + (X^TX)^{-1}X^TE(\epsilon)$
as ϵ is the only other independent variable
Since the mean of $\epsilon = 0$, $E(\epsilon) = 0$
Therefore $E(\theta) = \theta^*$

Problem 2 Part 2

$Var(\theta) = (X^TX)^{-1}\sigma^2$
$\theta = \theta^* + (X^TX)^{-1}X^T\epsilon$ from part 1
As $Var(AB) = AVar(B)A^T$, we can take the above variance the same way
$Var(\theta) = (X^TX)^{-1}X^TVar(\epsilon)X(X^TX)^{-1}$
As $Var(\epsilon) = \sigma^2$
$Var(\theta) = (X^TX)^{-1}X^T\sigma^2X(X^TX)^{-1}$
$Var(\theta) = (X^TX)^{-1}\sigma^2X^TX(X^TX)^{-1}$
Cancelling terms
$Var(\theta) = (X^TX)^{-1}\sigma^2$

Problem 3

Note: all figures for problems 3 are saved as PDF files in HW1 zip file.

Problem 3.1.A2: Implementing gradient descent

Theta found by gradient descent: [34.55363411 -0.95003694]

The quality of the linear fit for the data is relatively good in the middle part, where the value of LSTAT is from 5 to 25. However, at the low and high ends, the model tends to give predictions lower than the real median home values. Fitting the model with more data would improve the quality of the fit. In addition, since the shape of the data is not very linear, using a model of higher order may also result in a better fit.

Problem 3.1.A3: Predicting on unseen data

Prediction on unseen data
For lower status percentage = 5, we predict a median home value of $ 298034.49412207
For lower status percentage = 50, we predict a median home value of $ -129482.12889799
Comparing with sklearn's linear regression model
The coefficients computed by sklearn: 34.5538408794 and -0.950049353758
The model we implemented using gradient descent matches the one that sklearn's linear regression model learns on the same data.
Assessing model quality
5 fold cross validation MSE = 42.61890333
5 fold cross validation r squared = 0.297106799977

Problem 3.1.B3: Making predictions on unseen data

Prediction on unseen data
For average home in Boston suburbs, we predict a median home value of $ 225328.063241

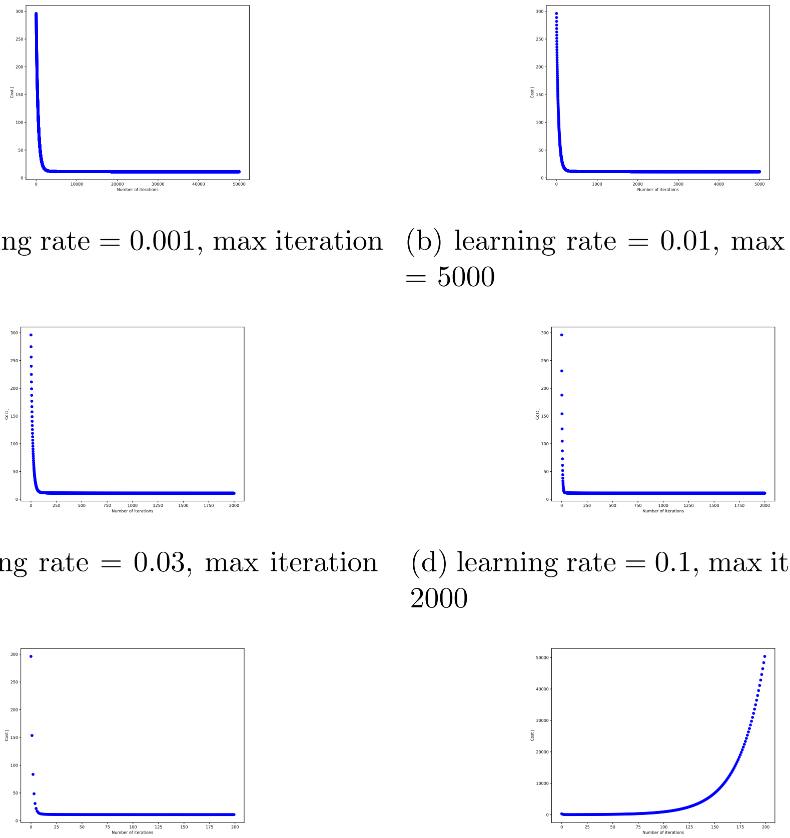
Problem 3.1.B4: Normal equations

Prediction on unseen data using normal equation
For average home in Boston suburbs, we predict a median home value of $ 225328.063241
The predictions of the model using gradient descent and the model using normal equation match up. Normal equation allows us to find the closed form of the optimal parameters θ's. If gradient descent method converges, the optimal parameters it finds should be the same as the closed form solution. However, when the number of examples is large, gradient descent would be much faster than normal equation to determine parameters.

Problem 3.1.B5: Exploring convergence of gradient descent

The plots of loss function vs. iteration numbers are shown below. In general, a good learning rate should be relatively small, so that the gradient descent method can converge.

In this case, the proper learning rate should be smaller than 0.33. As shown in Figure 1(f), when learning is 0.33, the gradient descent diverges. Figures 1(a) - (e) also show that as learning rate decreases, the number of iterations required for convergence increases. When the learning rate is 0.001 (Figure 1(a)), it takes more than 2000 iterations to converge, while when the learning rate is 0.3 (Figure 1(e)), the required number of iteration is only about 150. Therefore, the best learning rate should be relatively small so that gradient descent can converge, but large enough so that convergence will occur fast. In this specific case, 0.3 is a good learning rate.



(a) learning rate = 0.001, max iteration = 50000

(b) learning rate = 0.01, max iteration = 5000

(c) learning rate = 0.03, max iteration = 2000

(d) learning rate = 0.1, max iteration = 2000

(e) learning rate =0.3, max iteration = 200

(f) learning rate = 0.33, max iteration = 200

Figure 1: Exploring convergence of gradient descent with various learning rates

Problem 3.2.A4: Adjusting the regularization parameters

Adjusting λ(the regularization parameter) can have a very large impact on the quality of the learned model. When lambda is too small, the model tends to be overfitted with high variance as seen by the large validation error and small training error. When lambda is too large, the model becomes underfitted with high training and validation error, showing high bias in situations where the model fitted doesn't seem to match the data.

Problem 3.2.A5: Selecting λ using a validation set

The best λ value we found was 3.

0.1 Problem 3.2.A6: Calculating test error on the best model

The best error is 8.38.