COMP 540 HW 01

Lyu Pan (lp28), Yuhui Tong (yt30)

January 19, 2018

0. Background Refresher

0.1 Samplers

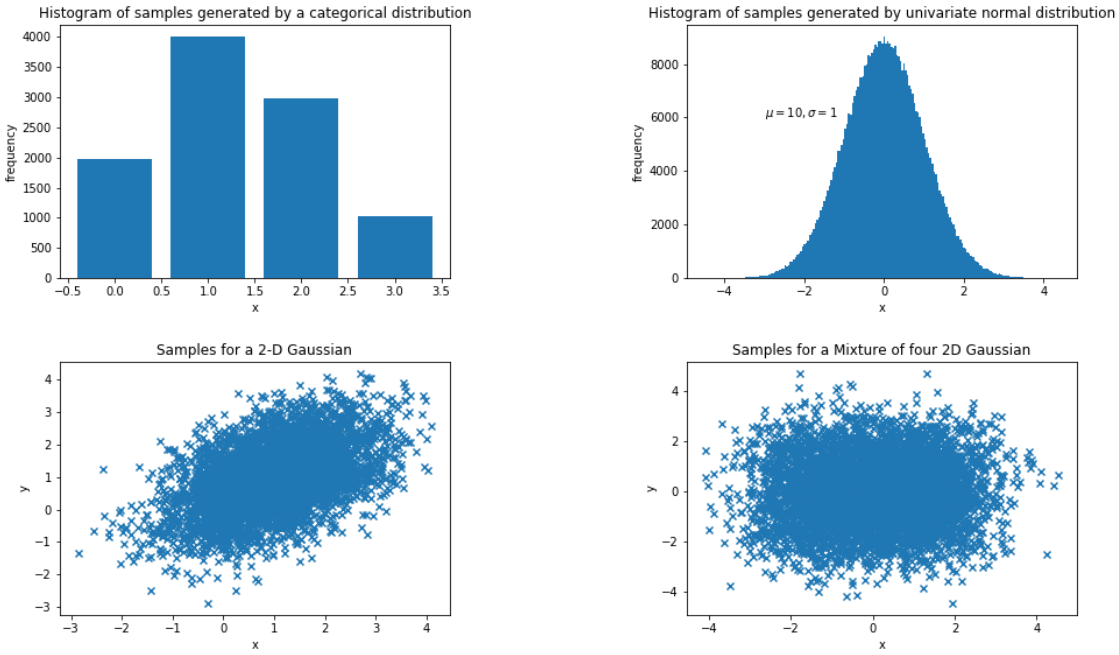


Figure 1: Visualization of four distributions

Figure 1 visualize four distributions. Specifically, for a mixture distribution of four 2D Gaussians, the probability that a sample from this distribution lies within the unit circle centered at $(0.1, 0.2)$ is $p = 0.1773$.

0.2 Prove two independent Poisson random variables are also Poisson variable.

Proof:

Given two independent random variables $X_1 \sim P(\lambda_1)$ and $X \sim P(\lambda_2)$. i.e.,

$$P(X_1 = m) = e^{-\lambda_1} \frac{\lambda_1^m}{m!}, \ m = 1, 2, ... \tag{1}$$

and

$$P(X_2 = n) = e^{-\lambda_2}\frac{\lambda_2^n}{n!},\ n = 0, 1, 2, ... \tag{2}$$

Denote sum of them are

$$X = X_1 + X_2 \tag{3}$$

then the probability distribution of X is:

$$\begin{aligned} P(X=k) &= \sum_{m+n=k} P(X_1 = m, X_2 = n) \\ &\overset{X_1, X_2 indep. R.V.}{=} \sum_{m+n=k} P(X_1 = m)P(X_2 = n) \\ &= \sum_{m+n=k} e^{-\lambda_1}\frac{\lambda_1^m}{m!}e^{-\lambda_2}\frac{\lambda_2^n}{n!} \\ &= \frac{1}{k!}e^{-(\lambda_1+\lambda_2)}\sum_{m+n=k}\frac{k!}{m!n!}\lambda_1^m\lambda_2^n \\ &= \frac{1}{k!}e^{-(\lambda_1+\lambda_2)}(\lambda_1+\lambda_2)^k \end{aligned} \tag{4}$$

that is, $X = X_1 + X_2 \sim P(\lambda_1 + \lambda_2)$. Q.E.D.

0.3 Proof question 3

$$\begin{aligned} p(x_1, x_0) &= p(x_1|x_0)p(x_0) \\ &= \alpha e^{-\frac{(x_1-x_0)^2}{2\sigma^2}}\alpha_0 e^{-\frac{(x_0-\mu_0)^2}{2\sigma_0^2}} \end{aligned} \tag{5}$$

The probability of $X_1 = x_1$ is,

$$\begin{aligned} p(X_1 = x_1) &= \int p(X_1 = x_1, X_0 = x_0)\, dx_0 \\ &= \alpha_0\alpha\int exp\left(-\frac{1}{2}\left(\frac{(x_0-\mu_0)^2}{\sigma_0^2} + \frac{(x_1-x_0)^2}{\sigma^2}\right)\right)dx_0 \\ &= \frac{\alpha_0\alpha}{A}exp\left(-\frac{1}{2}\frac{(x_1-\mu_0)^2}{\sigma^2+\sigma_0^2}\right) \end{aligned}$$

Here A is a constant. So

$$p(X_1 = x_1) = \alpha_1 exp\left(-\frac{1}{2}\frac{(x_1-\mu_1)^2}{\sigma_1^2}\right)$$

And

$$\mu_1 = \mu_0$$

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

$$\alpha_1 = \frac{\alpha_0\alpha}{A} = \alpha_0\alpha\int exp\left(-\frac{1}{2}\frac{x_0^2 - 2\left(\frac{\sigma^2\mu_0+\sigma_0^2x_1}{\sigma^2+\sigma_0^2}\right)x_0 + \left(\frac{\sigma^2\mu_0+\sigma_0^2x_1}{\sigma^2+\sigma_0^2}\right)^2}{\sigma_0^2\sigma^2/\left(\sigma^2+\sigma_0^2\right)}\right)dx_0$$

0.4 Eigenvalues

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

Solving the eigen equations:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{X} &= \lambda \boldsymbol{X} \\ (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{X} &= 0 \\ \begin{bmatrix} 13-\lambda & 5 \\ 2 & 4-\lambda \end{bmatrix} \boldsymbol{X} &= 0 \end{aligned}$$

gives the eigen value and eigen vectors (not normalized):

$\lambda_1 = 3$, $\boldsymbol{X}_1 = \begin{pmatrix} -1/2 \\ 1 \end{pmatrix}$

$\lambda_2 = 14$, $\boldsymbol{X}_2 = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$

0.5 Matrix multiplication

$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$, we have $(A+B)^2 \neq A^2 + 2AB + B^2$

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

0.6 Question 6

$$\begin{aligned} A^T A &= (I - 2uu^T)^T (I - 2uu^T) \\ &= (I - 2uu^T)(I - 2uu^T) \\ &= I - 2uu^T - 2uu^T + 4u(u^T u)u^T \\ &= I \end{aligned} \tag{6}$$

0.7 Convex function

0.7.1 Triple

for $x \geq 0$

$$f(x) = 3x^3 \tag{7}$$

$$f^{'}(x) = 3x^2 \tag{8}$$

$$f^{''}(x) = 6x \geq 0 \tag{9}$$

so,$f(x) = x^3$ is convex.

0.7.2 Two dimension

For any $\lambda \in [0,1]$,
we have $\lambda \geq 0, 1-\lambda \geq 0$
For any $(x_1, y_1), (x_2, y_2) on R^2$

$$\begin{aligned} f(\lambda(x_1,y_1)+(1-\lambda)(x_2,y_2)) &= max(\lambda(x_1,y_1)+(1-\lambda)(x_2,y_2)) \\ &\leq max(\lambda(x_1,y_1))+max((1-\lambda)(x_2,y_2)) \\ &= \lambda max(x_1,y_1)+(1-\lambda)max(x_2,y_2) \\ &= \lambda f(x_1,y_1)+(1-\lambda)f(x_2,y_2) \end{aligned} \tag{10}$$

So, $f(x)$ is convex.

0.7.3 Plus

Because f is convex on S, for $\lambda \in [0,1]$ and all $x_1, x_2 \in S$, we have

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \tag{11}$$

Also,

$$g(\lambda x_1 + (1-\lambda)x_2) \leq \lambda g(x_1) + (1-\lambda) g(x_2) \tag{12}$$

Let $h = f + g$,

$$\begin{aligned} h(\lambda x_1 + (1-\lambda)x_2) &= f(\lambda x_1 + (1-\lambda)x_2) + g(\lambda x_1 + (1-\lambda)x_2) \\ &\leq \lambda f(x_1) + (1-\lambda) f(x_2) + \lambda g(x_1) + (1-\lambda) g(x_2) \\ &= \lambda(f(x_1)+g(x_1)) + (1-\lambda)(f(x_2)+g(x_2)) \\ &= \lambda h(x_1) + (1-\lambda) h(x_2) \end{aligned} \tag{13}$$

So, h is convex, i.e. $f+g$ is convex.

0.7.4 Multiply

Let $h = fg$

$$\begin{aligned} h^{''} &= (f^{'}g + fg^{'})^{'} \\ &= f^{''} + 2f^{'}g^{'} + fg^{''} \end{aligned} \tag{14}$$

Obviously, $f \geq 0$, $f^{''} \geq 0$, $g \geq 0$, $g^{''} \geq 0$ on S.
Let the minimum of both f and g be x_0, then when $x \leq x_0$, $f(x_0) \leq 0$ and $g(x_0) \leq 0$; when $x \geq x_0$, $f(x_0) \geq 0$ and $g(x_0) \geq 0$, i.e. $f^{'}g^{'} \geq 0$. So

$$h^{''} \geq 0 \tag{15}$$

h is convex, i.e. $f+g$ is convex.

0.8 Entropy of categorical distribution

The entropy of a categorical distribution on K values is

$$H(p) \quad = \quad -\sum_{i=1}^{K} p_i \log(p_i), \tag{16}$$

with constraint that

$$\sum_{i=1}^{K} p_i = 1. \tag{17}$$

Using Lagrange Multiplier, one can combine the above two equations into:

$$L(p, \lambda) = -\sum_{i=1}^{K} p_i \log(p_i) + \lambda(\sum_{i=1}^{K} p_i - 1). \tag{18}$$

Taking derivative of all unknown variables gives:

$$\frac{\partial L}{\partial p_i} = -(\log p_i + 1) + \lambda = 0, \ i = 1, 2, ... \tag{19}$$

Substituting p_i with λ yields

$$p_i = \frac{1}{K}, \ i = 1, 2, ..., K. \tag{20}$$

Q.E.D.

1. Locally weighted linear regression.

1.1 Expression of $J(\theta)$

Define $\boldsymbol{X}$, $\boldsymbol{W}$ in the following way:

$$\boldsymbol{X} = \begin{bmatrix} - - - - \left(x^{(1)}\right)^T - - - - \\ - - - - \left(x^{(2)}\right)^T - - - - \\ \vdots \\ - - - - \left(x^{(m)}\right)^T - - - - \end{bmatrix} \iff \boldsymbol{X}_{i,j} = (x^{(i)})_j \tag{21}$$

$$\boldsymbol{W} = \begin{bmatrix} w^{(1)} & & & \\ & w^{(2)} & & \\ & & \ddots & \\ & & & w^{(m)} \end{bmatrix} \iff \boldsymbol{W}_{i,j} = \delta_{i,j} w^{(i)} \tag{22}$$

Substituting $\boldsymbol{X}$, $\boldsymbol{W}$ into $(\boldsymbol{X}\theta - \boldsymbol{y})^T \boldsymbol{W} (\boldsymbol{X}\theta - \boldsymbol{y})$ yields

$$
\begin{aligned}
(\boldsymbol{X}\theta-\boldsymbol{y})^T\boldsymbol{W}(\boldsymbol{X}\theta-\boldsymbol{y}) &= \sum_{i,j}\left[(\boldsymbol{X}\theta-\boldsymbol{y})^T\right]_{1,i}\boldsymbol{W}_{i,j}(\boldsymbol{X}\theta-\boldsymbol{y})_{j,1} \\
&= \sum_{i,j}\delta_{i,j}w^{(i)}\left[(\boldsymbol{X}\theta-\boldsymbol{y})^T\right]_{1,i}(\boldsymbol{X}\theta-\boldsymbol{y})_{j,1} \\
&= \sum_i w^{(i)}\left[(\boldsymbol{X}\theta-\boldsymbol{y})_{i,1}\right]^2 \\
&= \sum_i w^{(i)}\left[\left(x^{(i)}\right)^T\theta-y^{(i)}\right]^2 \\
&= \sum_i w^{(i)}\left[\theta x^{(i)}-y^{(i)}\right]^2.
\end{aligned}
$$

So $J(\theta)=\frac{1}{2}(\boldsymbol{X}\theta-\boldsymbol{y})^T\boldsymbol{W}(\boldsymbol{X}\theta-\boldsymbol{y})$. Q.E.D.

1.2 Closed formed solution of θ

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial\theta} &= \frac{1}{2}\frac{\partial}{\partial\theta}\left[(\boldsymbol{X}\theta-\boldsymbol{y})^T\boldsymbol{W}(\boldsymbol{X}\theta-\boldsymbol{y})\right] \\
&= \frac{1}{2}\frac{\partial}{\partial\theta}tr\left[\boldsymbol{y}^T\boldsymbol{W}\boldsymbol{X}\boldsymbol{\theta}-\boldsymbol{y}^T\boldsymbol{W}\boldsymbol{y}-\boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}\boldsymbol{\theta}+\boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y}\right] \\
&= \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y}-\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}\boldsymbol{\theta}
\end{aligned}
$$

where in the third line we used the following two equations[1]:

$$\frac{\partial tr(\boldsymbol{A}^T\boldsymbol{B})}{\partial\boldsymbol{A}}=\boldsymbol{B}^T \tag{23}$$

and

$$\frac{\partial tr(\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})}{\partial\boldsymbol{A}}=\boldsymbol{B}\boldsymbol{A}+\boldsymbol{B}^T\boldsymbol{A} \tag{24}$$

Therefore the closed form solution for θ is

$$\theta=(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y}. \tag{25}$$

[1]Take the second equation for example:

$$
\begin{aligned}
\left[\frac{\partial tr(\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})}{\partial\boldsymbol{A}}\right]_{i,j} &= \frac{\partial A^T_{l,m}B_{m,n}A_{n,l}}{\partial A_{i,j}}=\frac{\partial A_{m,l}B_{m,n}A_{n,l}}{\partial A_{i,j}} \\
&= \delta_{m,i}\delta_{l,j}B_{m,n}A_{n,l}+A_{m,l}B_{m,n}\delta_{n,i}\delta_{l,j} \\
&= B_{i,n}A_{n,j}+A_{m,j}B_{m,i} \\
&= (\boldsymbol{B}\boldsymbol{A})_{i,j}+(\boldsymbol{B}^T\boldsymbol{A})_{i,j},
\end{aligned}
$$

hence leading to $\frac{\partial tr(\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})}{\partial\boldsymbol{A}}=\boldsymbol{B}\boldsymbol{A}+\boldsymbol{B}^T\boldsymbol{A}$

1.3 Batch gradient descent for locally weighted linear regression

The derivative of J_θ is:

$$\frac{\partial}{\partial\theta}J(\theta) = \sum_{i=1}^{m} w^{(i)}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}. \tag{26}$$

Therefore we have the following algorithm:

Algorithm 1: Batch gradient descent algorithm for locally weighted linear regression

Result: The estimated parameters θ for locally weighted linear regression

```
while θ does not converge do
    for every j do
        θ_j = θ_j − α Σ_{i=1}^{m} w^(i)(h_θ(x^(i)) − y^(i)) x_j^(i);
    end
end
```

The batch gradient descent algorithm for locally weighted linear regression is NON parametric model.

2. Properties of linear regression estimator.

2.1 Prove $E[\theta] = \theta^*$

Proof:

The following facts:

1. $y^{(i)} = \theta^{*T}x^{(i)} + \epsilon^{(i)}$,
2. $\epsilon^{(i)}, i = 1, 2, ..., m$ are *i.i.d.* of $N(0, \sigma^2)$,

indicate that:

given fixed arbitrary $x^{(i)}$ and fixed unknown parameter θ^*, $y^{(i)}$, $1 \le i \le m$ are *i.i.d. of* $N(\theta^{*T}x^{(i)}, \sigma^2)$

$$p(y^{(i)}|x^{(i)}, \theta^*) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y^{(i)}-\theta^{*T}x^{(i)})^2}{2\sigma^2}}. \tag{27}$$

The least-square estimate of θ^* is θ given by

$$\theta = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{28}$$

$$\overset{denote:(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\equiv\boldsymbol{A}}{=} \boldsymbol{A}\boldsymbol{y} \tag{29}$$

The expectation of θ is

$$E[\theta] = E[\boldsymbol{A}\boldsymbol{y}] \tag{30}$$

$$= E\begin{bmatrix} \sum_j A_{1,j}y^{(j)} \\ \sum_j A_{2,j}y^{(j)} \\ \vdots \\ \sum_j A_{d+1,j}y^{(j)} \end{bmatrix}. \tag{31}$$

Since expectations has the following property (regardless of the independence of Z_k):

$$E[Z_1 + Z_2 + ... + Z_l] = \sum_k Z_k, \tag{32}$$

we have

$$E[\theta] = E\begin{bmatrix} \sum_j A_{1,j}y^{(j)} \\ \sum_j A_{2,j}y^{(j)} \\ \vdots \\ \sum_j A_{d+1,j}y^{(j)} \end{bmatrix} = \begin{bmatrix} \sum_j A_{1,j}E[y^{(j)}] \\ \sum_j A_{2,j}E[y^{(j)}] \\ \vdots \\ \sum_j A_{d+1,j}E[y^{(j)}] \end{bmatrix} \tag{33}$$

$$= \boldsymbol{A}\begin{bmatrix} E[y^{(1)}] \\ E[y^{(2)}] \\ \vdots \\ E[y^{(m)}] \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} \theta^{*T}x^{(1)} \\ \theta^{*T}x^{(2)} \\ \vdots \\ \theta^{*T}x^{(m)} \end{bmatrix} \tag{34}$$

$$= \boldsymbol{A}\boldsymbol{X}\theta^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\theta^* \tag{35}$$

$$= \theta^* \tag{36}$$

where in the second step the following equation is used: $E[y^{(i)}] = \theta^{*T}x^{(i)}$ (trivial to obtain as $y^{(i)}$ observe normal distribution). Therefore $E[\theta] = \theta^*$, implying that the estimation θ is unbiased. Q.E.D

2.2 Prove $Var(\theta) = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\sigma^2$

Denote $\boldsymbol{A} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$, therefore $\theta = \boldsymbol{A}\boldsymbol{y}$.

$$Var(\theta) = Var(\boldsymbol{A}\boldsymbol{y}) \tag{37}$$

$$= Var(\begin{bmatrix} \sum_j A_{1,j}y^{(j)} \\ \sum_j A_{2,j}y^{(j)} \\ \vdots \\ \sum_j A_{m,j}y^{(j)} \end{bmatrix}) = \begin{bmatrix} \sum_j A_{1,j}Var(y^{(j)}) \\ \sum_j A_{2,j}Var(y^{(j)}) \\ \vdots \\ \sum_j A_{m,j}Var(y^{(j)}) \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} Var(y^{(1)}) \\ Var(y^{(2)}) \\ \vdots \\ Var(y^{(m)}) \end{bmatrix}, \tag{38}$$

where in the second line we made used of the property of variance:

$$Var(\sum_k Z_k) = \sum Var(Z_k),\ Z_k\ is\ independent\ from\ each\ other. \tag{39}$$

Substituting $Var(y^{(i)}) = \sigma^2\ i = 1, 2, ..., m$, we have $Var(\theta) = \boldsymbol{A}\sigma^2 = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\sigma^2$. Q.E.D.

3. Implementing linear regression and regularized linear regression

3.1.A1

No plot for this question.

3.1.A2

Figure 2 plots the linear fit using parameter obtained through training, while Fig. 3 shows the convergence of the loss function against iteration times.

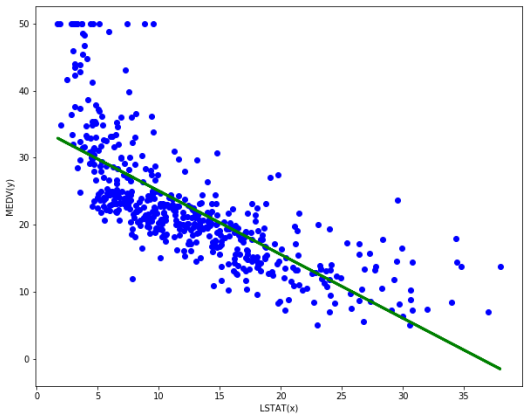


Figure 2: Fitting a linear model to the data

Figure 2 plots the linear fit using parameter obtained through training.

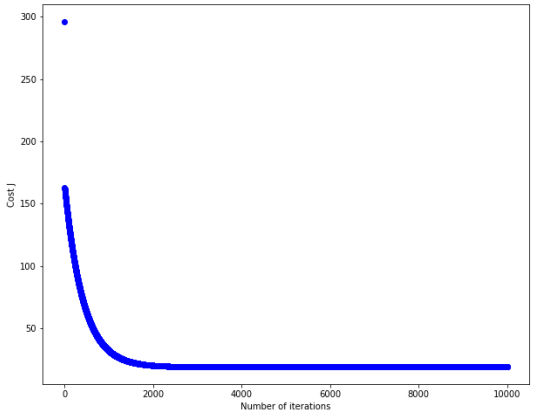


Figure 3: Convergence of Loss function against the number of iterations.

Problem 3.1.A3: Predicting on unseen data

For lower status percentage = 5, we predict a median home value of 298034.494122

For lower status percentage = 50, we predict a median home value of -129482.128898

3.1.B1: Feature Normalization

No plot for this question.

3.1.B2: Loss function and gradient descent

Figure 4 shows the loss function against the number of iterations.

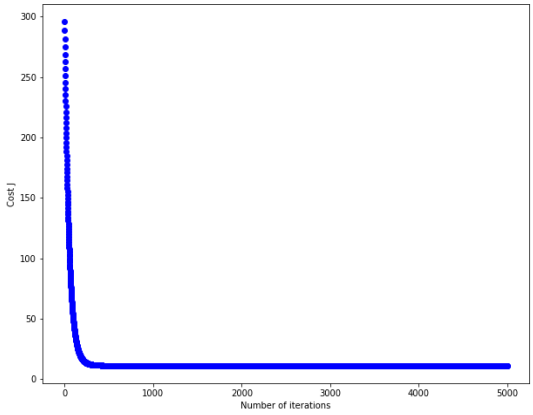


Figure 4: Fitting a linear model to the data

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?

- The linear model gives a rather rough estimation of the data. The issue with fitting clear is excessive bias / underfitting, i.e. too strong assumption on that Boston house price observes a sole linear relation with LSTAT.
- I expect the linear model performs awfully at the low and high ends of values for LSTAT. Using linear model with only one feature, one obviously assumes that the data should distreated rather evenly on both sides of the model, which in fact contradicts with the real data.
- The problem with current model is lack of variation, which can be treated with introducing more features (e,g. through feature engineering).

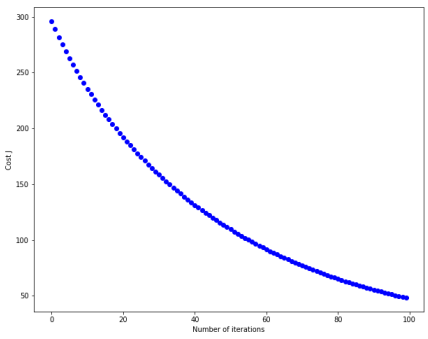
3.1.B3 Making predictions on unseen data

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

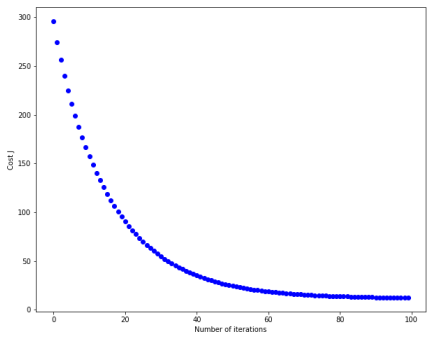
3.1.B4: Normal equations

For average home in Boston suburbs, we predict a median home value of 225328.063241, which is the same as we obtained in subsec. 3.1.B3.

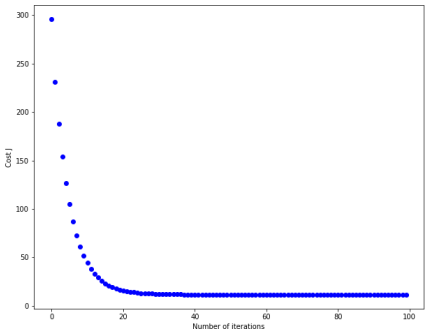
Problem 3.1.B5: Exploring convergence of gradient descent



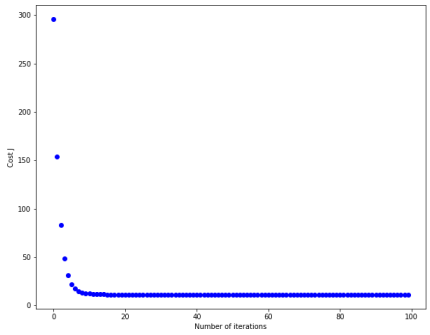
(a) learning rate $\alpha = 0.01$



(b) learning rate $\alpha = 0.03$



(c) learning rate $\alpha = 0.1$



(d) learning rate $\alpha = 0.3$

Figure 5: Convergence of gradient descent for linear regression with multiple variables using different learning rate.

$\alpha = 0.1, 0.3$ and $N_{iteration} = 80$ are good trade off between accuracy and efficiency. By observing Fig. 5d , one can easily find that small α (i.e. $\alpha = 0.01, 0.03$) leads to very slow convergence rate. $\alpha = 0.1, 0.3$ on the other hands, converges swiftly.

3.2.A1: Regularized linear regression cost function

No figures for this question.

3.2.A2: Gradient of the regularized linear regression cost function

Figure 6 shows the fitted curve of the linear model.

3.2.A3: Learning curves

Figure 7 shows the learning curve of the linear model.

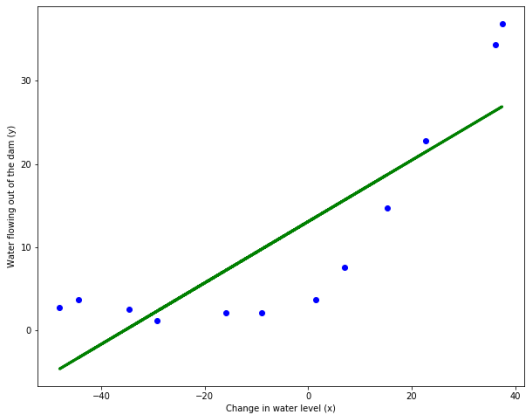


Figure 6: The fitted curve of the linear model.

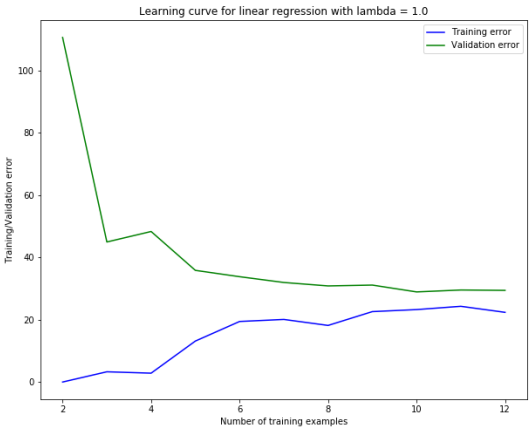
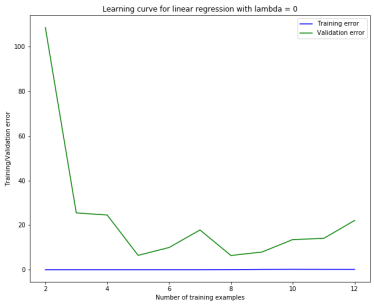


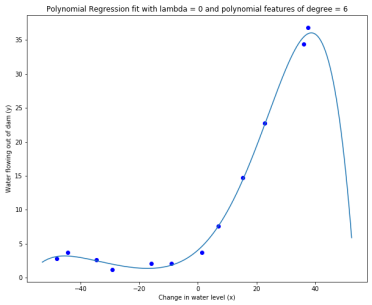
Figure 7: Learning curve of the linear model.

3.2.A4: Adjusting the regularization parameter

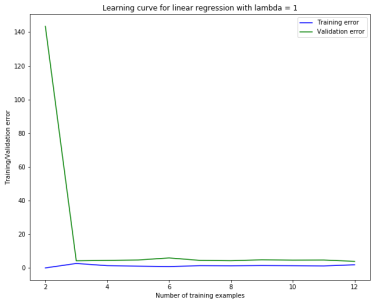
Figure 8 plots the polynomial fit and learning curves for each value of λ, from which we draw the following conclusions:



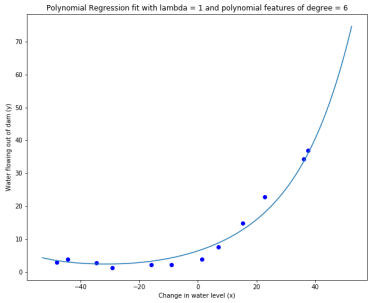
(a) learning rate $\lambda = 0$



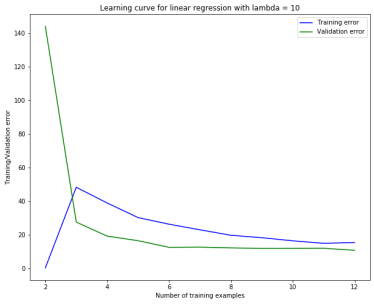
(b) Polynomial fit $\lambda = 0$



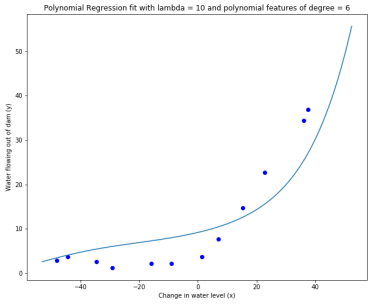
(c) learning rate $\lambda = 1$



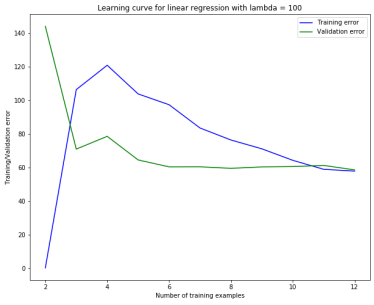
(d) Polynomial fit $\lambda = 1$



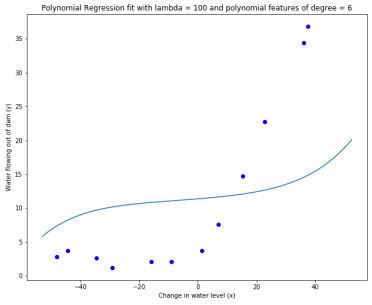
(e) learning rate $\lambda = 10$



(f) Polynomial fit $\lambda = 10$



(g) learning rate $\lambda = 100$



(h) Polynomial fit $\lambda = 100$

Figure 8: Convergence of gradient descent for linear regression with multiple variables using different learning rate.

If λ is too small, the effect of penalty term can be neglected so that the regulation will not be implemented effectively – the training model is still troubled by high-variation/over-fitting issue (as can be seen in Fig....). When λ is too large, the loss function is in fact dominated by the penalty term, which is a slightly similar to the biased issue that we have too strong assumptions on the model. Therefore, the training model turns to be under-fit. Only when λ takes appropriate value that the regulation can works effectively.

3.2.A5 Selecting λ using a validation set

Figure 9 shows the variation in trainng/validation error with respect of regulation parameter λ. For the current model, $\lambda = 1$ is approximately a good choice for the training. We list several reasons to justify our choice:

- The difference between training/validation error is too large when λ is significantly smaller than 1, implying an over-fitting issue – which is true because the penalty term characterized by λ is too small to regulate the effect of excessive features.
- The difference between training/validation error becomes relatively small, but the absolute training/validation error becomes too large when λ gets larger than 1. This indicates that the penalty term dominates the loss function and we actually are making a strong assumption of the target function (of similar form as that of penalty term), i.e., too much bias / not enough variation.
- $\lambda = 1$ makes a good balance to avoid either overfitting or underfitting, i.e. the validation/training error is small (the prediction performance is acceptable); the difference between validation/training error is small (the prediction model is reliable).

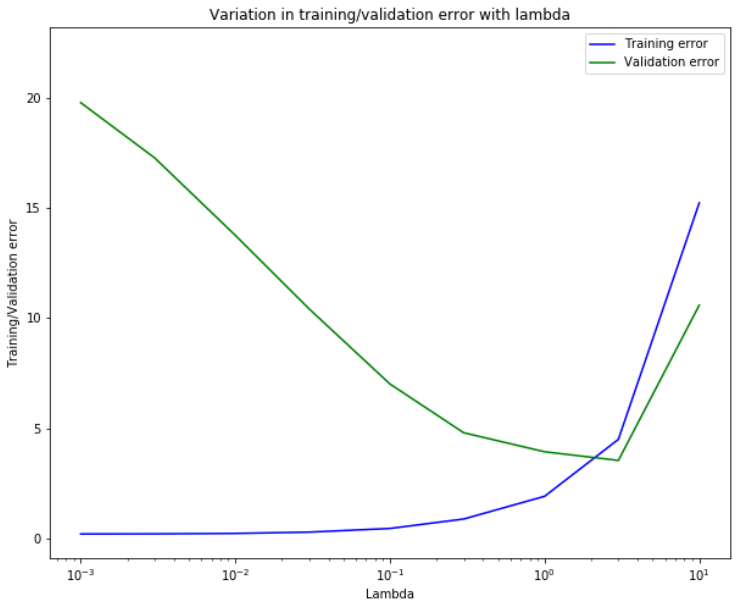


Figure 9: Variation in training/validation error with λ

3.2.A6 Computing test set error

Error when choosing best lamdba 1.0 is 30987.4826556 (USD).

3.2.A7 Plotting learning curves with randomly selected examples

Figure 10 plot the averaged learning curve for $\lambda = 1$.

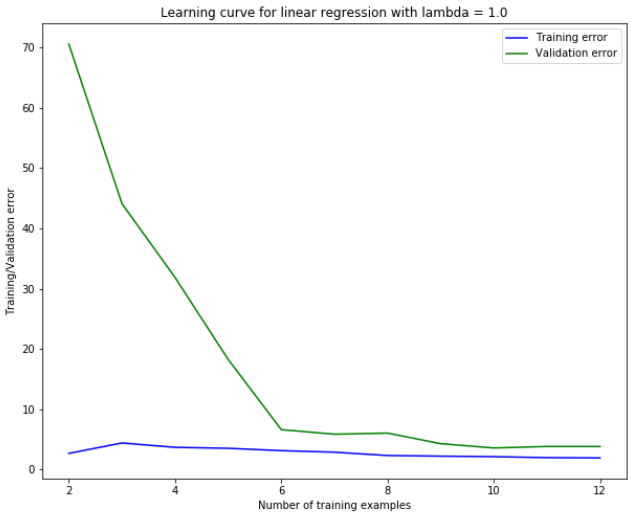


Figure 10: Averaged learning curve for $\lambda = 1$

Bonus question:

Please see **bostonexp.ipynb** for our detailed solution.

4.1 Regularized Linear Regression

Figure 11 plots the variation in training/testing error with λ for regularized linear model.

- Best $\lambda = 10$:
- training error: 11.605155
- test error: 12.6566689644

4.2 Selecting λ with quadratic features

Figure 12 plots the variation in training/testing error with λ for regularized linear model with quadratic features.

- Best $\lambda = 0.3$:

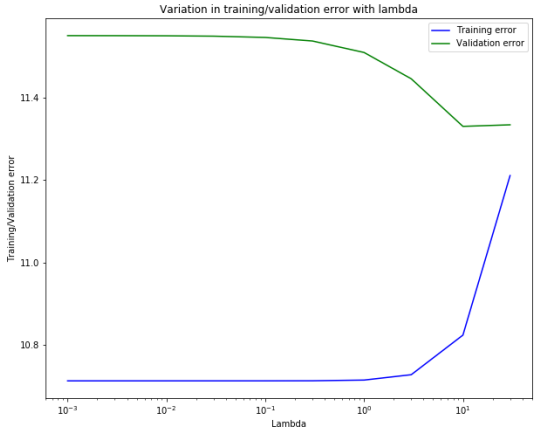


Figure 11: Variation in training/testing error with λ for regularized linear model

- training error: 4.256794
- test error: 4.82815820863

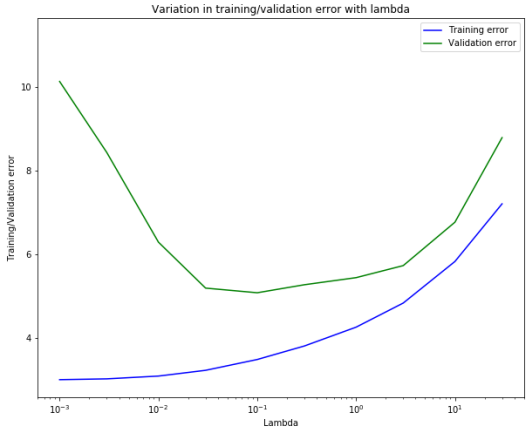


Figure 12: Variation in training/testing error with λ for regularized linear model with quadratic features

4.3 Selecting λ with cubic features

Figure 13 plots the variation in training/testing error with λ for regularized linear model with quadratic features.

- Best $\lambda = 3$:
- training error: 3.645624
- test error: 4.73217608015

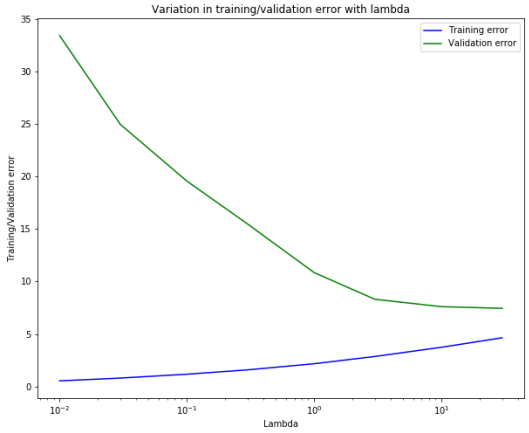


Figure 13: Variation in training/testing error with λ for regularized linear model with quadratic features

Summary

In above sections, we trained linear regression model with linear/quadratic/cubic features to predict Boston house price. We can observe that all three models we built embodied the power of regulation in that appropriate value of λ gives the "sweet region" — and even though the number of features grow dramatically as we square and even cube the features for more variation, the regulation term in the loss function successfully constrains the model from overfitting. The fact that models with linear, quadratic, cubic features are successively one better than the other shows that: given more features (variations) and proper control of overfitting, the more delicate model has better performance in prediction.