Lee Sutton
301145106

CMPT 726 – Assignment 1

1 Probabilistic Modelling

In lecture we went over an example of modeling coin tossing – estimating a parameter that is the probability the coin comes up heads. Consider instead the problem of modeling the outcome of the Canadian Federal election. To simplify matters, assume one party will win a majority (i.e. either the NDP, Liberals, Conservatives, or Green Party wins

1. (4 marks) What is the type of distribution that describes this situation? What are the parameters μ of this distribution? (See PRML Appendix B)

Instead of using a binomial distribution with 2 possible outcomes, this situation can best be modelled by a multinomial distributions with 4 outcomes. The requirements for this distribution is as follows:

- The experiment consists of n repeated trials. In our case, each riding can be considered a single trial.
- Each trial has a discrete number of possible outcomes. We will assume that all ridings are made up of one member from each party and no independent members will be running.
- On any given trial, the probability that a particular outcome will occur is constant. Meaning the probability is constant for each member running in each municipal riding.
- The trials are independent; that is, the outcome on one trial does not affect the outcome on other trials. This would indicate the outcome from each riding has no effect on the other outcomes.

Using the assumptions listed above, a multinomial distribution with 4 possible outcomes can be used to model this situation. There would be 4 parameters required for this distribution listed below:

- $\mu_N = \ Probability\ of\ the\ NDP\ winning\ a\ riding$
- $\mu_L = \ Probability\ of\ the\ liberals\ winning\ a\ riding$
- $\mu_C = \ Probability\ of\ the\ Conservatives\ winning\ a\ riding$
- $\mu_G = \ Probability\ of\ the\ Green\ party\ winning\ a\ riding$

2. (2 marks) What would be the value of the parameters μ for an election where the outcome is an equal chance of each party winning?

If the probability of each party winning is equal, the parameters would all take on the same value. Since all of the events are mutually exclusive and make up all the possible outcomes.

$$\mu_N = \mu_L = \mu_C = \mu_G$$

$$\mu_N + \mu_L + \mu_C + \mu_G = 1$$

$$\mu_N = \mu_L = \mu_C = \mu_G = 0.25$$

3. (2 marks) What would be the value of the parameters μ for an election that is completely "rigged"? E.g. the party currently in power is definitely going to win.

If the election was completely rigged, all of the parameters would be zero except for the parameter for the party that rigged the election. For the party that rigged the election, their parameter μ would take on a value of 1. For example, if the party currently in power completely rigged the election, the parameters would look as follows:

$$\mu_N = \mu_L = \mu_G = 0$$

$$\mu_C = 1$$

4. (4 marks) suppose my prior is that the Green Party has completely rigged the election. Assume I see a set of polls where the NDP has the largest share of the vote in each poll. What would be my posterior probability on the parameters μ?

If my prior is that the green party has completely rigged the election, my prior would assume $\mu_G=1$. The posterior is proportional to likelihood of the data given the parameters multiplied by my prior. This is shown below.

$$P(\mu|D) \propto P(D|\mu)P(\mu)$$

Posterior likelihood prior

If I were to see a set of polls where the NDP had the largest share of the vote my posterior would return 0 since the probability that $P(\mu_g = 1|D = majority\ NDP)$ would be = 0. You could then use the data given to maximize the likelihood estimation to get the best estimate of the parameters μ.

2 Regularized Least Squares Linear Regression

Solution on next page

2. SHOW THAT THE MINIMIZER FOR L_2 REGULARIZATION IS $\vec{\omega} = (\lambda I + \underline{\Phi}^T \underline{\Phi})^{-1} \underline{\Phi}^T t$

ERROR IS GIVEN BY:

$$\tilde{E}(\omega) = \frac{1}{2} \sum_{n=1}^{N} \{ y_n(x_n, \omega) - t_n \}^2 + \frac{\lambda}{2} \omega^T \omega$$

OR

$$\tilde{E}(\omega) = \left[\frac{1}{2} \sum_{n=1}^{N} (t_n - \bar{\omega}^T \phi(\bar{x}_n))^2 \right] + \left[\frac{\lambda}{2} \omega^T \omega \right]$$

→ TAKE $\nabla \tilde{E}$ AND SOLVE WHERE $\nabla \tilde{E} = 0$

$$\vec{0}^T = \nabla(\tilde{E}(\omega)) = \left[\frac{\partial}{\partial \omega_0}(\tilde{E}(\omega)), \frac{\partial}{\partial \omega_1}(\tilde{E}(\omega)), \frac{\partial}{\partial \omega_2}(\tilde{E}(\omega)) \ldots \right]$$

$$\vec{0}^T = \frac{2}{2} \sum_{n=1}^{N} (t_n - \bar{\omega}^T \phi(x_n))(-\phi(x_n)) + \lambda \omega^T$$

$$\vec{0}^T = \sum_{n=1}^{N} \left[(t_n - \bar{\omega}^T \phi(x_n))(-\phi(x_n)) \right] + \lambda \omega^T$$

DEFINE: (AS IN THE NOTES)

$$\underline{\Phi} = \begin{bmatrix} \phi_0(\bar{x}_1) & \phi_1(\bar{x}_1) & \cdots & \phi_{M-1}(\bar{x}_1) \\ \phi_0(\bar{x}_2) & \phi_1(\bar{x}_2) & \cdots & \phi_{M-1}(\bar{x}_2) \\ \vdots & \vdots & & \vdots \\ \phi_0(\bar{x}_N) & \phi_1(\bar{x}_N) & \cdots & \phi_{M-1}(\bar{x}_N) \end{bmatrix}$$

$$\vec{0}^T = -t^T \underline{\Phi} + \omega^T \underline{\Phi}^t \underline{\Phi} + \lambda \omega^T$$

$$\boxed{\vec{\omega} = \left(\underline{\Phi}^t \underline{\Phi} + \lambda I \right)^{-1} \underline{\Phi} t^T}$$

Lee Sutton
301145106

3 Training vs. Test Error

1. ***Suppose we perform unregularized regression on a dataset. Is the training error with a degree 10 polynomial always lower than or equal to that using a degree 9 polynomial? Explain***
   When using unregularized regression, the training error will always be less than or equal to the training error produced with a higher degree polynomial. The coefficients are based on minimizing the training error. By increasing the degree of the polynomial, there are more coefficients to minimize the error and allow the graph to pass through or closer to more points.

2. ***Suppose we perform unregularized regression on a dataset. Is the testing error with a degree 10 polynomial always lower than or equal to that using a degree 9 polynomial? Explain.***
   The testing error is not necessarily lower with a ten degree polynomial. When increasing the degree of the polynomial it is possible to over fit the curve. If this happens, the testing error will increase.

3. ***Suppose we perform unregularized regression on a dataset. Is the training error always lower than the testing error? Explain.***
   In most cases, the testing error will be higher than the training error. Since the curve is designed to minimize the error on the training points, the training error is likely lower than the testing error. However, it is possible for the testing error to be smaller than the training error. If the data set was not properly randomized initially or if the data set was small, it is possible that the test points happen to lie closer to the predicted curve than the training points.

4 Regression

4.1 Getting Started

1. ***Which country had the highest child mortality rate in 1990? What was the rate?***
   Niger had the highest under 5 mortality rate in 1990 with a mortality rate of 313.7 deaths per thousand live births.
2. ***Which country had the highest child mortality rate in 2011? What was the rate?***
   Sierra Leone had the highest under 5 mortality rate in 2011 with a mortality rate of 185.3 deaths per thousand live births.
3. ***Some countries are missing some features (see original .xlsx/.csv spreadsheet). How is this handled in the function loadUnicefData.m?***
   In the function loadUnicefData.m, the function reads data from excel and uses str2double to convert the data from strings into doubles. If the data to be converted is not a number, (including underscores) the value for that particular property is replaced with the median value for that property.

4.2 Regression

1. Plots of the training error and test error are shown below in Figure 1 and Figure 2. Figure 1 shows the results before the data was normalized. From the graph we can see that the training error is actually increasing with polynomial degree. This is due to the fact that the data was not normalized

before running the regression model. Figure 2 shows the results after the input data was normalized. After normalizing the data, the training error now decreases with an increasing polynomial degree.

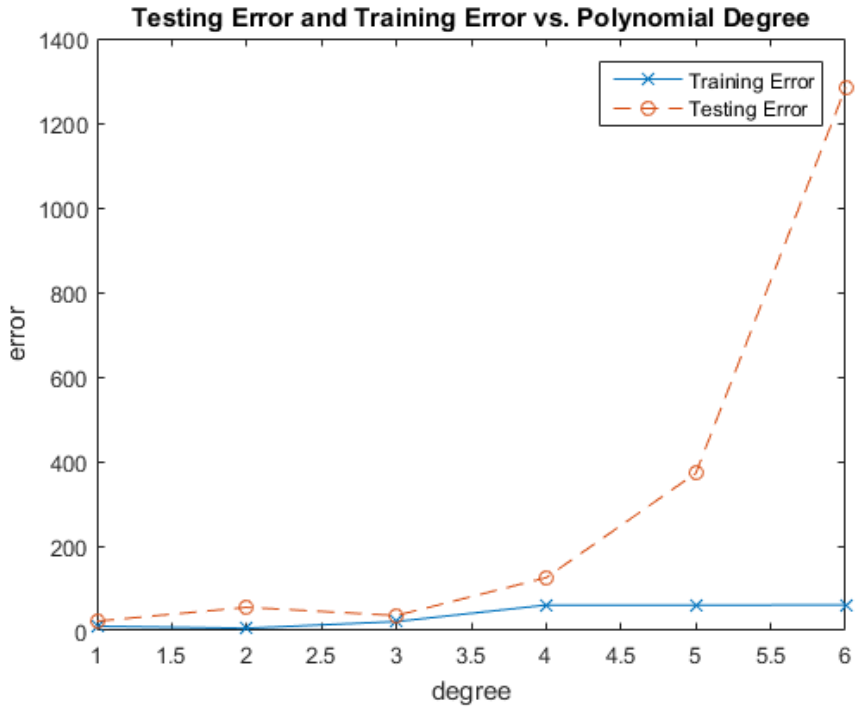

Figure 1 Testing error and training error vs polynomial degree before normalizing the input

Lee Sutton
301145106

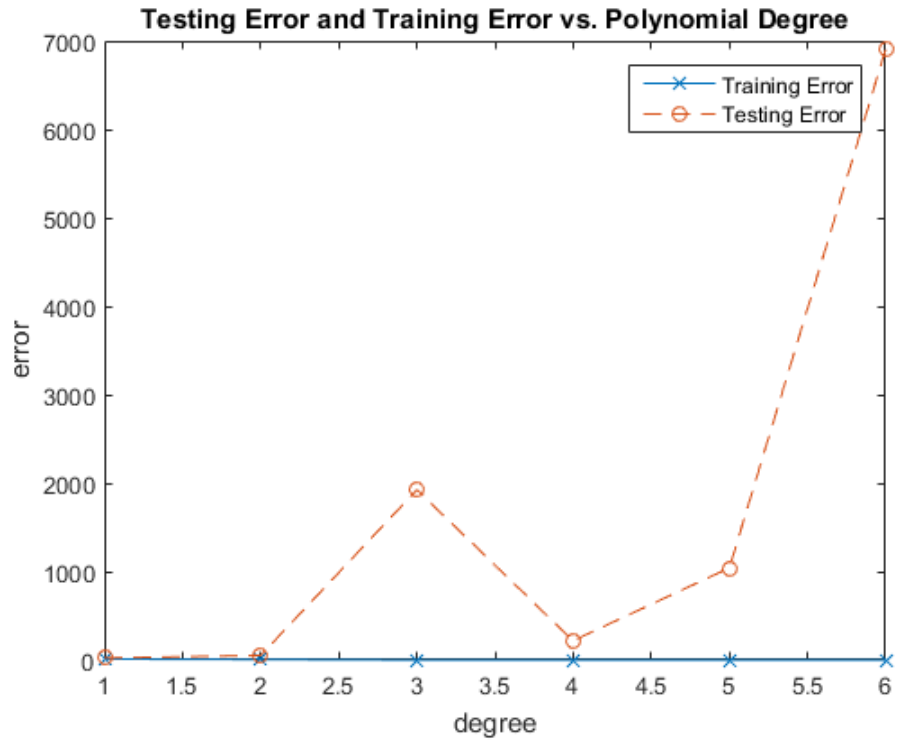

Figure 2 Testing error and training error vs polynomial degree after normalizing the input

2. For question 2, one dimensional polynomial regression was used. The results are shown below in a bar graph. Figure 3 shows the training error and test error for 8 input features Figure 4, Figure 5, and Figure 6 show the individual features with 3rd degree polynomial regression.

Lee Sutton
301145106

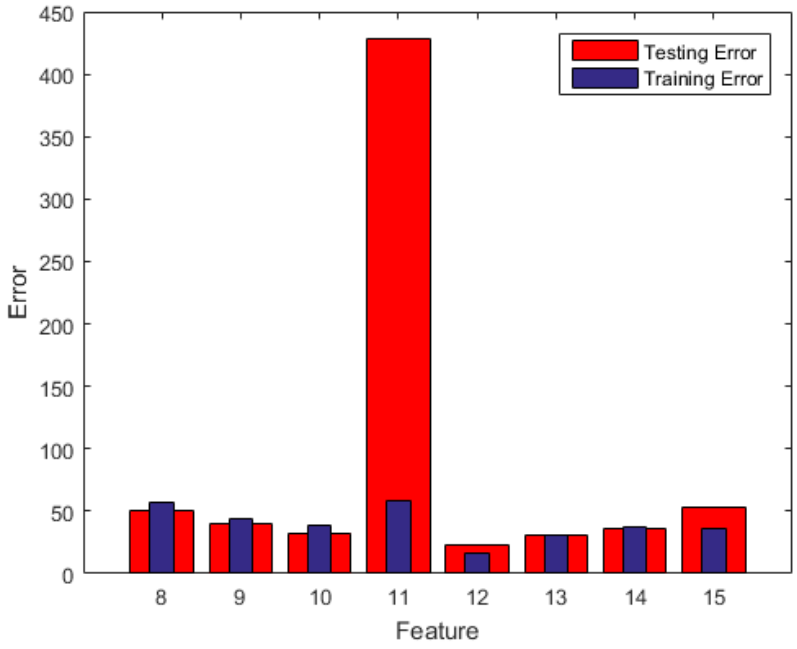

Figure 3 Bar graph showing the testing and training error of each individual feature

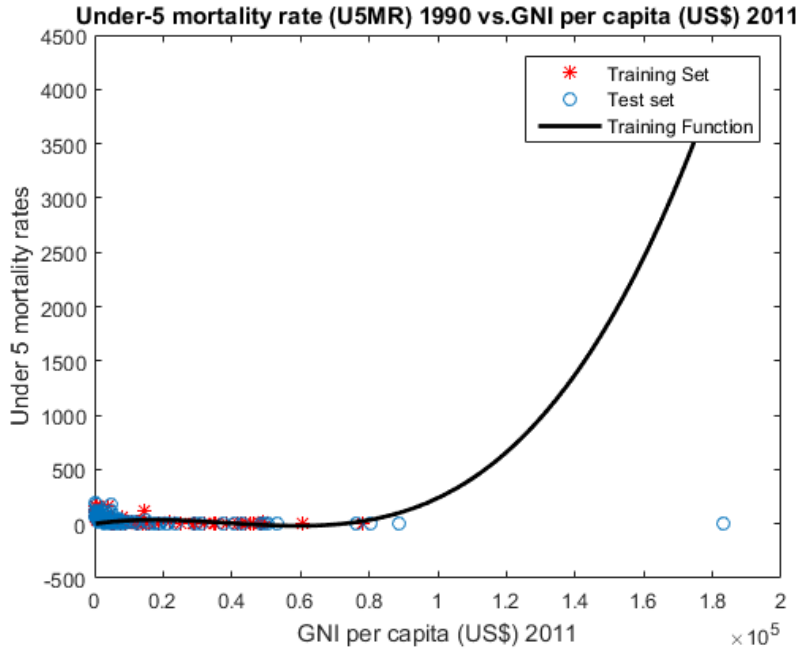

Figure 4 Plot of U5MR vs. GNI including the training set, test set, and the function obtained through regression

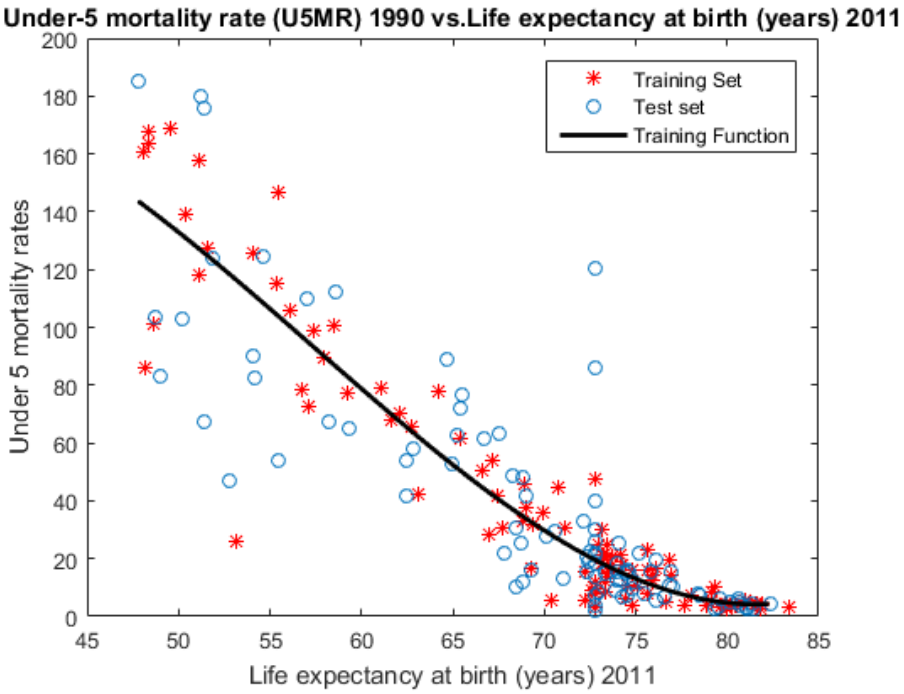

Figure 5 Plot of U5MR vs. Life Expectancy at birth including the training set, test set, and the function obtained through regression

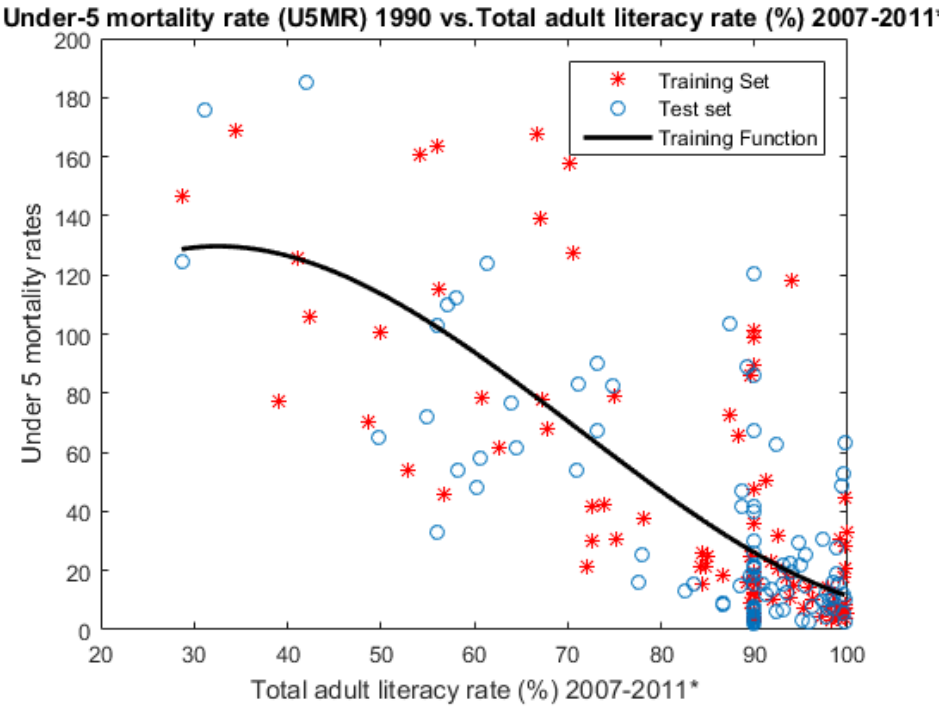

Figure 6 Plot of U5MR vs. adult literacy rate including the training set, test set, and the function obtained through regression

Lee Sutton
301145106

4.3 Sigmoid Basis Functions

1. Using sigmoid regression, a curve was generated for feature 11. The results are shown below.

Training Error	28.4137
Test Error	31.2946

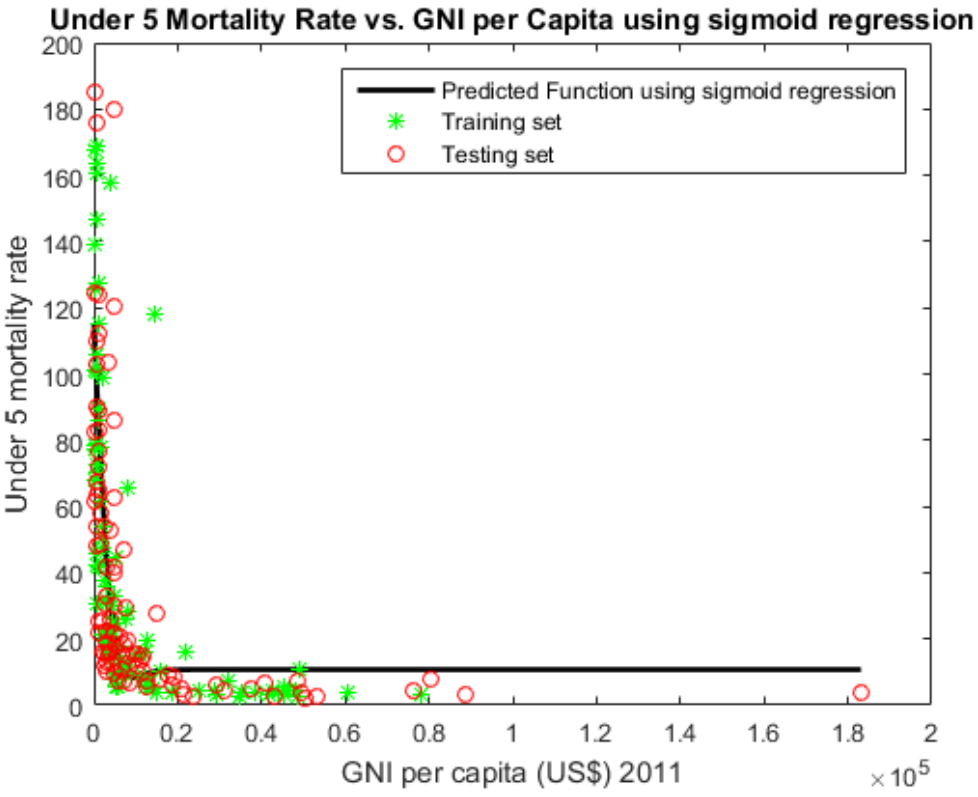

Figure 7 U5MR vs. GNI per capita fitted using sigmoid regression

Lee Sutton
301145106

4.4 Regularized Polynomial Regression

Using L2 regularization, a 2 degree polynomial was fit to the function. Cross validation was used to test the values of lambda. The results are shown below in Figure 8. Note that the validation error for lambda = 0 could not be plotted since the x –axis used a log scale. **The validation error for lambda = 0 was found to be 69.66**. When examining the graph it appears that lambda = 100 or 1000 would give the best results. Upon further inspection the validation error was slightly lower with **lambda = 1000**.

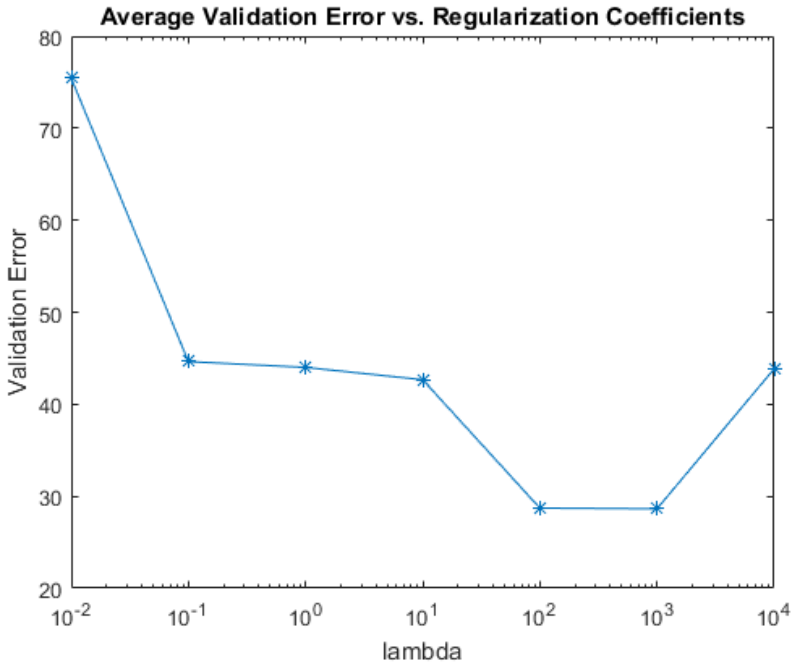

Figure 8 Average validation error for different regularization coefficients