Xihan Zhang
xihanzhang@hsph.harvard.edu
CS181-S18

Assignment #1, v1.3
Due: 11:59pm February 2, 2018

Collaborators: Fangli Geng

Homework 1: Linear Regression

Introduction

This homework is on different forms of linear regression and focuses on loss functions, optimizers, and regularization. Linear regression will be one of the few models that we see that has an analytical solution. These problems focus on deriving these solutions and exploring their properties.

If you find that you are having trouble with the first couple problems, we recommend going over the fundamentals of linear algebra and matrix calculus. We also encourage you to first read the Bishop textbook, particularly: Section 2.3 (Properties of Gaussian Distributions), Section 3.1 (Linear Basis Regression), and Section 3.3 (Bayesian Linear Regression). (Note that our notation is slightly different but the underlying mathematics remains the same :).

Please type your solutions after the corresponding problems using this LaTeX template, and start each problem on a new page. You will submit your solution PDF and at most 1 `.py` file per problem (for those that involve a programming component) to Canvas.

Problem 1 (Priors and Regularization,15pts)

In this problem we consider a model of Bayesian linear regression. Define the prior on the parameters as,

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \,|\, \mathbf{0}, \tau_w^{-1}\mathbf{I}),$$

where τ_w is as scalar precision hyperparameter that controls the variance of the Gaussian prior. Define the likelihood as,

$$p(\mathbf{y} \,|\, \mathbf{X}, \mathbf{w}) = \prod_{i=1}^{n} \mathcal{N}(y_i \,|\, \mathbf{w}^\mathsf{T}\mathbf{x}_i, \tau_n^{-1}),$$

where τ_n is another fixed scalar defining the variance.

(a) Using the fact that the posterior is the product of the prior and the likelihood (up to a normalization constant), i.e.,

$$\arg\max_{\mathbf{w}} \ln p(\mathbf{w} \,|\, \mathbf{y}, \mathbf{X}) = \arg\max_{\mathbf{w}} \ln p(\mathbf{w}) + \ln p(\mathbf{y} \,|\, \mathbf{X}, \mathbf{w}).$$

Show that maximizing the log posterior is equivalent to minimizing a regularized loss function given by $\mathcal{L}(\mathbf{w}) + \lambda\mathcal{R}(\mathbf{w})$, where

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{w}^\mathsf{T}\mathbf{x}_i)^2$$

$$\mathcal{R}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w}$$

Do this by writing $\ln p(\mathbf{w} \,|\, \mathbf{y}, \mathbf{X})$ as a function of $\mathcal{L}(\mathbf{w})$ and $\mathcal{R}(\mathbf{w})$, dropping constant terms if necessary. Conclude that maximizing this posterior is equivalent to minimizing the regularized error term given by $\mathcal{L}(\mathbf{w}) + \lambda\mathcal{R}(\mathbf{w})$ for a λ expressed in terms of the problem's constants.

(b) Notice that the form of the posterior is the same as the form of the ridge regression loss

$$\mathcal{L}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda\mathbf{w}^\top\mathbf{w}.$$

Compute the gradient of the loss above with respect to $\mathbf{w}$. Simplify as much as you can for full credit. Make sure to give your answer in vector form.

(c) Suppose that $\lambda > 0$. Knowing that $\mathcal{L}$ is a convex function of its arguments, conclude that a global optimizer of $\mathcal{L}(\mathbf{w})$ is

$$\mathbf{w} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \tag{1}$$

For this part of the problem, assume that the data has been centered, that is, pre-processed such that $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$.

(d) What might happen if the number of weights in $\mathbf{w}$ is greater than the number of data points N? How does the regularization help ensure that the inverse in the solution above can be computed?

Solution

(a) Plug $p(\mathbf{w})$ and $p(\mathbf{y}\,|\,\mathbf{X},\mathbf{w})$ in $\ln p(\mathbf{w}\,|\,\mathbf{y},\mathbf{X})$, we get

$$\ln p(\mathbf{w}\,|\,\mathbf{y},\mathbf{X}) = \ln \mathcal{N}(\mathbf{w}\,|\,\mathbf{0},\tau_w^{-1}\mathbf{I}) + \ln \prod_{i=1}^{n} \mathcal{N}(y_i\,|\,\mathbf{w}^{\mathsf{T}}\mathbf{x}_i,\tau_n^{-1})$$

which is

$$\ln p(\mathbf{w}\,|\,\mathbf{y},\mathbf{X}) = -\frac{1}{2}\ln\det(2\pi\tau_w^{-1}) - \frac{1}{2}\tau_w\mathbf{w}^{\mathsf{T}}\mathbf{w} - \frac{1}{2}\sum_{i=1}^{n}\ln\det(2\pi\tau_n^{-1}) - \frac{1}{2}\tau_n\sum_{i=1}^{n}(y_i-\mathbf{w}^{\mathsf{T}}\mathbf{x}_i)^{\mathsf{T}}(y_i-\mathbf{w}^{\mathsf{T}}\mathbf{x}_i)$$

Regardless of constant we can see

$$\ln p(\mathbf{w}\,|\,\mathbf{y},\mathbf{X}) \propto -\frac{1}{2}\sum_{i=1}^{n}(y_i-\mathbf{w}^{\mathsf{T}}\mathbf{x}_i)^2 - \frac{\tau_w}{\tau_n}\frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

which is

$$\ln p(\mathbf{w}\,|\,\mathbf{y},\mathbf{X}) \propto -\mathcal{L}(\mathbf{w}) - \frac{\tau_w}{\tau_n}\mathcal{R}(\mathbf{w})$$

So that maximizing the posterior is equivalent to minimizing the regularized error term given by $\mathcal{L}(\mathbf{w})+\lambda\mathcal{R}(\mathbf{w})$ for $\lambda=\frac{\tau_w}{\tau_n}$

(b) The loss can be written as:

$$\mathcal{L}(\mathbf{w}) = [\mathbf{y}^\top - (\mathbf{X}\mathbf{w})^\top](\mathbf{y}-\mathbf{X}\mathbf{w}) + \lambda\mathbf{w}^\top\mathbf{w}$$
$$\mathcal{L}(\mathbf{w}) = \mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top(\mathbf{X}\mathbf{w}) - (\mathbf{X}\mathbf{w})^\top\mathbf{y} + (\mathbf{X}\mathbf{w})^\top(\mathbf{X}\mathbf{w}) + \lambda\mathbf{w}^\top\mathbf{w}$$

Take the gradient of part we can get:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = -2\mathbf{X}^\top\mathbf{y} + 2\mathbf{X}^\top\mathbf{X}\mathbf{w} + 2\lambda\mathbf{w}$$

(c) Set the gradient $\mathcal{L}(\mathbf{w})=0$, we can get:

$$-2\mathbf{X}^\top\mathbf{y} + 2\mathbf{X}^\top\mathbf{X}\mathbf{w} + 2\lambda\mathbf{w} = 0$$
$$\mathbf{X}^\top\mathbf{y} - (\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})\mathbf{w} = 0$$

Since $\forall x \in \mathbf{R}^n$, $\mathbf{x}^\top(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})\mathbf{x} = \|\mathbf{X}\mathbf{x}\|_2^2 + \lambda\,\|x\|_2^2$ is positive in this case, so that $(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})$ is inversable. Then the equation can be written as:

$$\mathbf{w} = (\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

(d) In this way, our model would give more information than data has, so that the model would perfectly fit with data at the price of decreasing the capability of generalization. i.e. this would cause over-fitting problem. Regularization can help reduce the complexity of model by adding penalty term coordinated by parameter λ into loss function. The penalty term includes number of total weights. By increasing λ, loss function would be more sensitive to penalty term, so that the total number of weights would be reduced to decrease the whole loss function.

2. Modeling Changes in Congress

The objective of this problem is to learn about linear regression with basis functions by modeling the number of Republicans in the Senate. The file `data/year-sunspots-republicans.csv` contains the data you will use for this problem. It has three columns. The first one is an integer that indicates the year. The second is the number of sunspots. The third is the number of Republicans in the Senate. The data file looks like this:

```
Year,Sunspot_Count,Republican_Count
1960,112.3,36
1962,37.6,34
1964,10.2,32
1966,47.0,36
1968,105.9,43
1970,104.5,44
```

and you can see plots of the data in Figures 1 and 2.

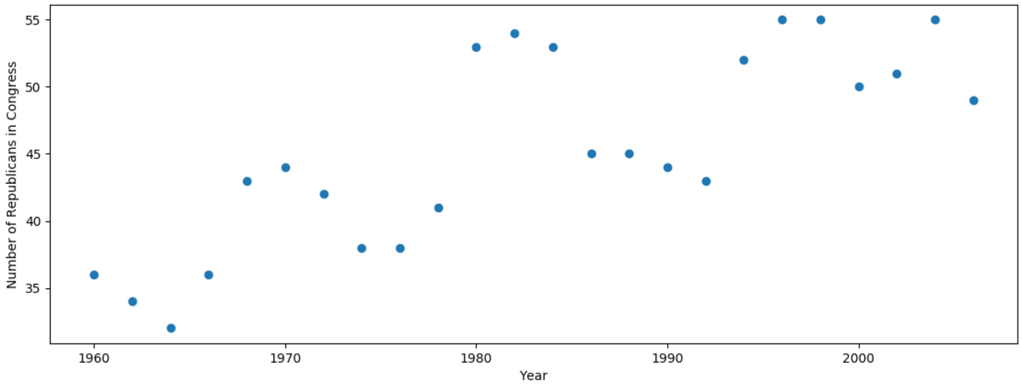


Figure 1: Number of Republicans in the Senate. The horizontal axis is the year, and the vertical axis is the number of Republicans.

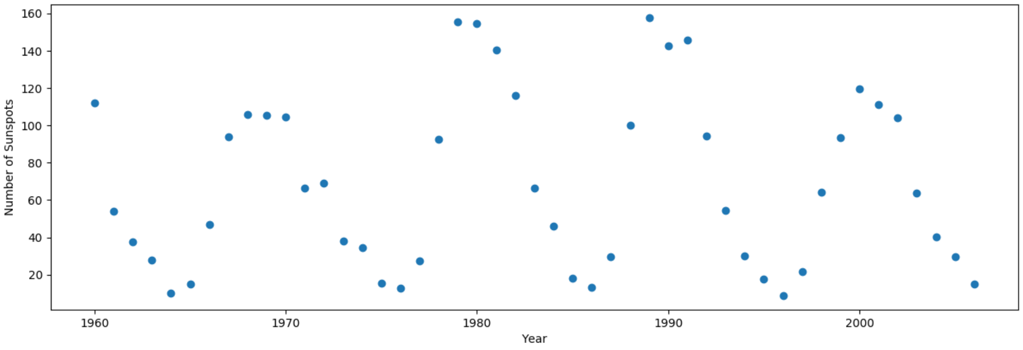


Figure 2: Number of sunspots by year. The horizontal axis is the year, and the vertical axis is the number of sunspots.

Data Source: http://www.realclimate.org/data/senators_sunspots.txt

Problem 2 (Modeling Changes in Republicans and Sunspots, 15pts)

Implement basis function regression with ordinary least squares for years vs. number of Republicans in the Senate. Some sample Python code is provided in `linreg.py`, which implements linear regression. Plot the data and regression lines for the simple linear case, and for each of the following sets of basis functions (only use (b) for years, skip for sunspots):

(a) $\phi_j(x) = x^j$ for $j = 1, \ldots, 5$

(b) $\phi_j(x) = \exp \frac{-(x-\mu_j)^2}{25}$ for $\mu_j = 1960, 1965, 1970, 1975, \ldots 2010$

(c) $\phi_j(x) = \cos(x/j)$ for $j = 1, \ldots, 5$

(d) $\phi_j(x) = \cos(x/j)$ for $j = 1, \ldots, 25$

In addition to the plots, provide one or two sentences for each with numerical support, explaining whether you think it is fitting well, overfitting or underfitting. If it does not fit well, provide a sentence explaining why. A good fit should capture the most important trends in the data.

Next, do the same for the number of sunspots vs. number of Republicans, using data only from before 1985. What bases provide the best fit? Given the quality of the fit, would you believe that the number of sunspots controls the number of Republicans in the senate?

Solution

(A) years vs. number of Republicans

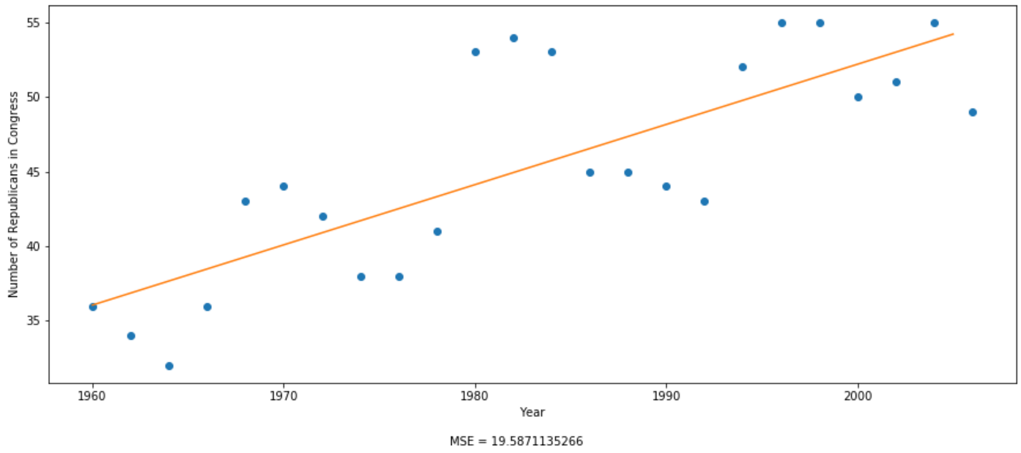


Figure 3: Data and regression for the simple linear case The model is too simple, so that the model is underfitting

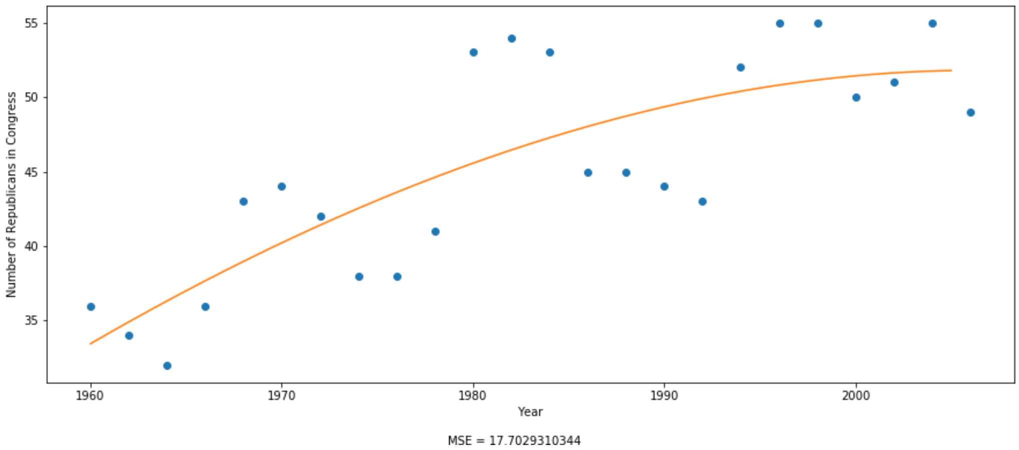


Figure 4: (a) $\phi_j(x) = x^j$ for $j = 1, \ldots, 5$. The number of polynomial features is too little to capture complexity of data, so that the model is underfitting

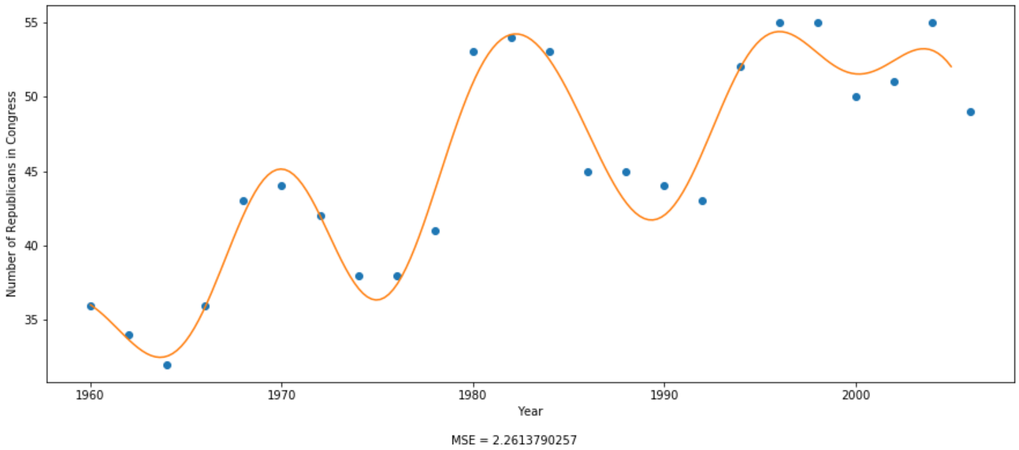


Figure 5: (b) $\phi_j(x) = \exp \frac{-(x-\mu_j)^2}{25}$ for $\mu_j = 1960, 1965, 1970, 1975, \ldots 2010$. The model sub-clusters it's variable by years, which coincide well with trend, so that it is well-fitting

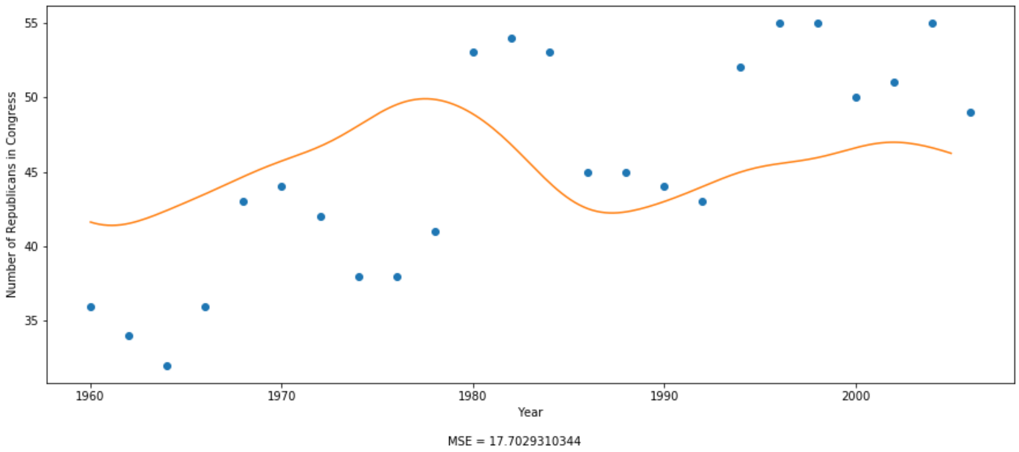


Figure 6: (c) $\phi_j(x) = \cos(x/j)$ for $j = 1, \ldots, 5$. The frequency of cos is too little to capture complexity of data, so that the model is underfitting

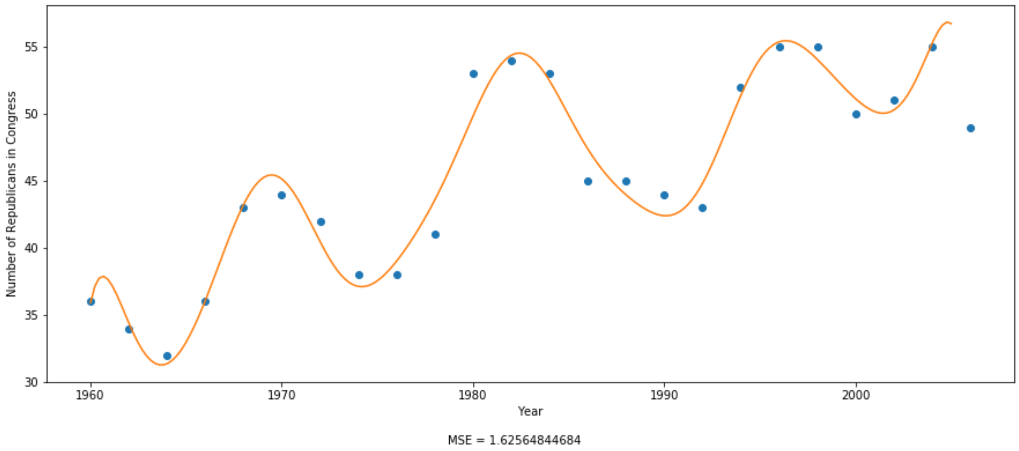
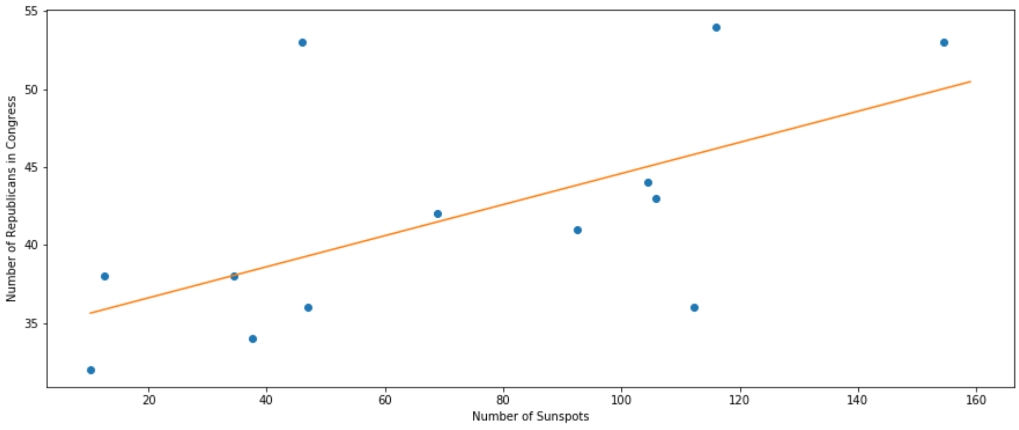


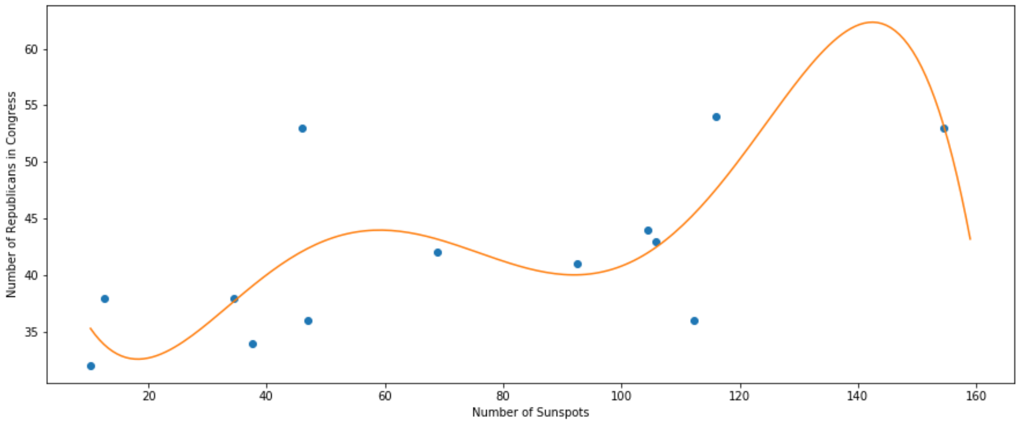
Figure 7: (d) $\phi_j(x) = \cos(x/j)$ for $j = 1, \ldots, 25$. The frequency of cos is too large, so that the model is overfitting

(B) Number of Sunspots vs. Number of Republicans



MSE = 19.5871135266

Figure 8: Simple linear case. There's no obvious pattern of data, and simple linear model is too simple to capture the complexity of data, so that the model is underfitting



MSE = 27.0175335211

Figure 9: (a) $\phi_j(x) = x^j$ for $j = 1, \ldots, 5$. There's no obvious pattern of data that follows polynomial models, so that the model is underfitting

We don't fit the model (b), because it's designed to capture the data trend towards year, which makes no sense in the relationship between number of sunspots and number of republicans.

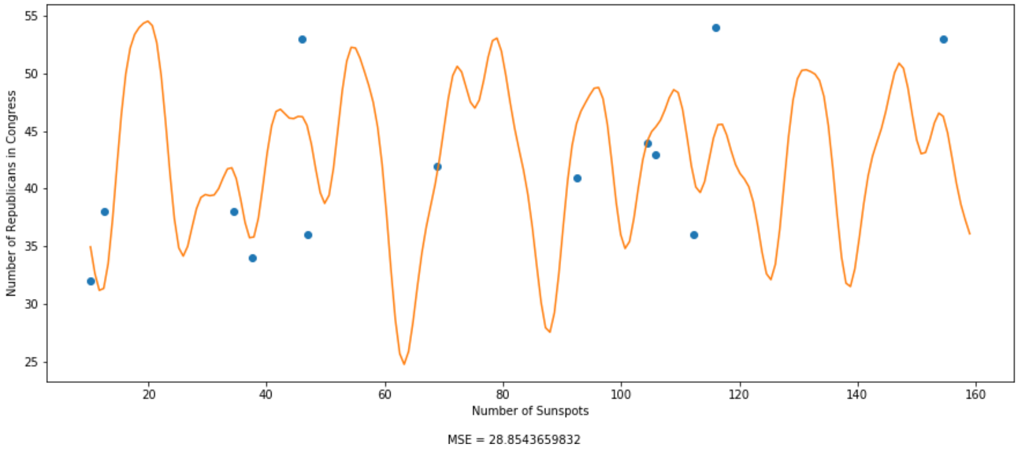


Figure 10: (c) $\phi_j(x) = \cos(x/j)$ for $j = 1, \ldots, 5$. There's no obvious pattern of data that follows cos models, and the frequency of cos is too little to capture every point of data, so that even though the model seems too complex to data, it is still underfitting

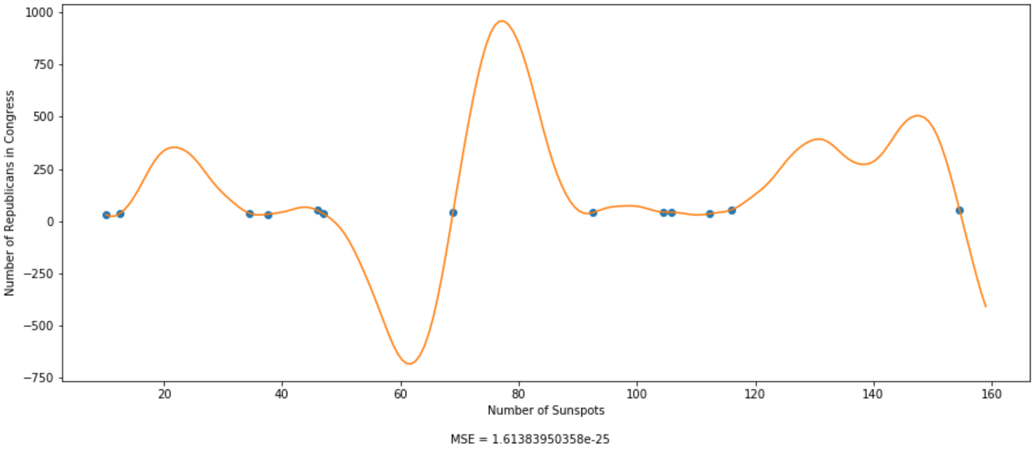


Figure 11: (d) $\phi_j(x) = \cos(x/j)$ for $j = 1, \ldots, 25$. There's no obvious pattern of data that follows cos models, and high frequency drives cos model crazy to fit with every data, so that the model is overfitting

From the MSE of all the models above, the simple linear regression fits best, because it is the simplest among all the models, but the error is still too large. From the quality of fits, we have no evidence to believe that number of sunspots control number of republicans.

Problem 3 (BIC, 15pts)

Adapted from *Biophysics : Searching for Principles* by William Bialek.

Consider data $\mathcal{D} = \{(x_i, y_i)\}$ where we know that

$$y_i = f(x_i; \mathbf{a}) + \epsilon_i$$

where $f(x_i; \mathbf{a})$ is a function with parameters $\mathbf{a}$, and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is an additive noise term. We assume that f is a polynomial with coefficients $\mathbf{a}$. Consider the class of all polynomials of degree K. Each of these polynomials can be viewed as a generative model of our data, and we seek to choose a model that both explains our current data and is able to generalize to new data. This problem explores the use of the Bayesian Information Criterion (BIC) for model selection. Define the χ^2 (*chi-squared*) error term for each model of the class as:

$$\chi^2 = \frac{1}{\sigma^2} \sum_{i=1}^{N} \left(y_i - \sum_{j=0}^{K} a_j x_i^j \right)^2$$

Using this notation, a formulation of the BIC can be written as:

$$-\ln P(x_i, y_i | \text{model class}) \approx \frac{N}{2} \ln(2\pi\sigma^2) - \sum_{i=1}^{N} \ln p(x_i) + \frac{1}{2}\chi^2_{min} + \frac{K}{2} \ln N$$

where $\chi^2_{min}(K)$ denote the minimum value of the χ^2 over the set of polynomial models with K parameters. Finally, assume that $x_i \sim Unif(-5, 5)$ and that each $a_j \sim Unif(-1, 1)$. Let $K_{true} = 10$.

(a) Write code that generates N data points in the following way:

1. Generate a polynomial $f(x) = \sum_{j=0}^{K_{true}} a_j x^j$
2. Sample N points x_i
3. Compute $y_i = f(x_i) + \epsilon_i$ where ϵ is sampled from $\mathcal{N}(0, \sigma^2 = \frac{\max_i f(x_i) - \min_i f(x_i)}{10})$.

(b) For a set of y_i generated above and a given K, write a function that minimizes χ^2 for a polynomial of degree K by solving for $\mathbf{a}$ using numpy `polyfit`. Check for $N = 20$ that $\chi^2_{min}(K)$ is a decreasing function of K.

(c) For $N = 20$ samples, run 500 trials. This involves generating a new polynomial for each trial, then from that polynomial, 20 sample data points $\{(x_i, y_i)\}$. For each trial, we can calculate the optimal K by minimizing BIC. Compute the mean and variance of the optimal K over 500 trials.

(d) For N ranging from 3 to $3 \cdot 10^4$ on a log scale (you can use the function $3 * np.logspace(0, 4, 40)$ as your values of N), compute the mean and variance of the optimal K over 500 trials for each N. Plot your results, where the x-axis is the number of samples (N) on a log-scale, and the y-axis is the mean value of the optimal K with error bars indicating the variance over 500 trials. Verify that minimizing the BIC controls the complexity of the fit, selecting a nontrivial optimal K. You should observe that the optimal K is smaller than K_{true} for small data sets, and approaches K_{true} as you analyze larger data sets.

Solution

(b) By plotting $\chi^2_{min}(K)$ with K=1, 2, 3, ..., 15, we can see the function is decreasing:

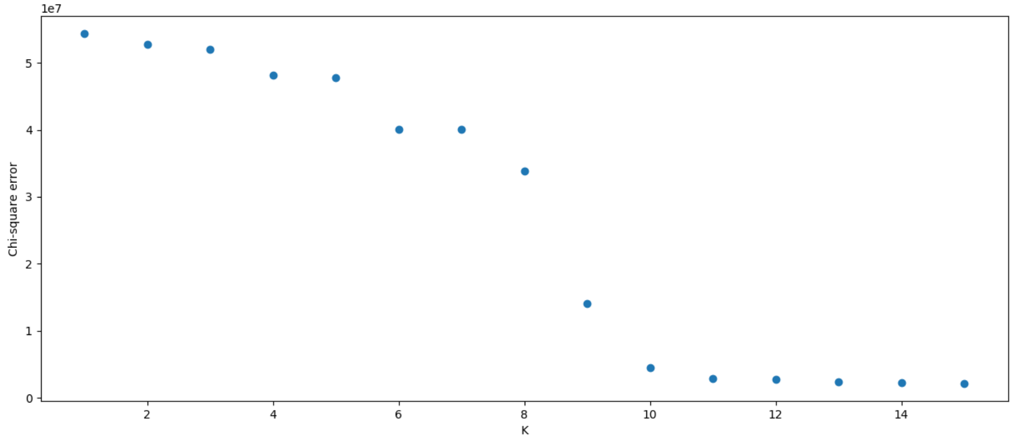


Figure 12: Plot of $\chi^2_{min}(K)$

(c) After simulation we get the following results:

The mean of optimal K over 500 trails is 9.99. The varriance of optimal K over 500 trails is 0.0139

(d) The plot result is as follow:

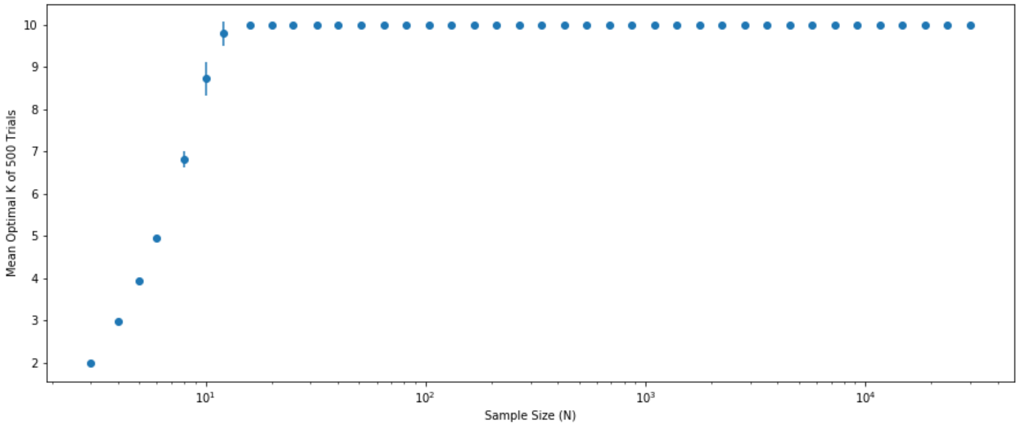


Figure 13: x-axis is the number of samples (N) on a log-scale, and the y-axis is the mean value of the optimal K with error bars indicating the variance over 500 trials

Problem 4 (Calibration, 1pt)

Approximately how long did this homework take you to complete?

Answer: 6 hours on reading books and notes; 15 hours on coding; 4 hours on Latex the result. Thus, it is 25 hours.