Homework 1: Linear Regression

Due 23:59 on Monday, September 26, 2022

You will do this assignment individually and submit your answers as a PDF and code via the Blackboard course website. There is a mathematical component and a programming component to this homework.

1. MLE Estimate of the Bias Term (Bishop equation (3.19)) [10pts]

Let $\mathbf{\Phi}$ be our $N \times J$ design matrix, $\mathbf{t}$ our vector of N target values, $\mathbf{w}$ our vector of J parameters, and w_0 our bias parameter. As Bishop notes in (3.18), the sum-of-squares error function of $\mathbf{w}$ and w_0 can be written as follows

$$E(\mathbf{w}, w_0) = \frac{1}{2} \sum_{n=1}^{N} \left(t_n - w_0 - \sum_{j=1}^{J-1} w_j \cdot \phi_j(x_n) \right)^2 .$$

Show that the value of w_0 that minimizes E is

$$\begin{aligned} w_{0_{MLE}} &= \frac{1}{N} \sum_{n=1}^{N} t_n - \sum_{j=1}^{J-1} w_j \cdot \left(\frac{1}{N} \sum_{n=1}^{N} \phi_j(x_n) \right) \\ &= \bar{t} - \sum_{j=1}^{J-1} w_j \cdot \overline{\phi_j(x)} \qquad \text{[compare to Bishop eqn. (3.19)]} \end{aligned}$$

2. Non-Uniformly Weighted Data [10pts]

Consider a data set in which each data point t_n is associated with a weighting factor $r_n > 0$, so that the sum-of-squares error function becomes

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} r_n \{ t_n - \mathbf{w}^\top \mathbf{\Phi}(\mathbf{x}_n) \}^2.$$

Find an expression for the solution $\mathbf{w}^*$ that minimizes this error function.

3. Priors and Regularization [10pts]

Consider the Bayesian linear regression model given in Bishop 3.3.1. The prior is

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

where α is the precision parameter that controls the variance of the Gaussian prior. The likelihood can be written as

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid \mathbf{w}^\top \mathbf{\Phi}(\mathbf{x}_n), \beta^{-1}),$$

Using the fact that the posterior is the product of the prior and the likelihood (up to a normalization constant), show that maximizing the log posterior (i.e., $\ln p(\mathbf{w} \mid \mathbf{t}) = \ln p(\mathbf{w}|\alpha) + \ln p(\mathbf{t} \mid \mathbf{w})$) is equivalent to minimizing the regularized error term given by $E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$ with

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} (t_n - \mathbf{w}^\top \mathbf{\Phi}(\mathbf{x}_n))^2$$
$$E_W(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\top \mathbf{w}$$

Do this by writing $\ln p(\mathbf{w} \mid \mathbf{t})$ as a function of $E_D(\mathbf{w})$ and $E_W(\mathbf{w})$, dropping constant terms if necessary. Conclude that maximizing this posterior is equivalent to minimizing the regularized error term given by $E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$. (Hint: take $\lambda = \alpha/\beta$)

4. Modeling Motorcycle Helmet Forces [10pts]

The objective of this problem is to learn about linear regression with basis functions by modeling the g-forces associated with motorcycle helmet impacts. Download the file `motorcycle.csv` from the course website. It has two columns. The first one is the number of milliseconds since impact and the second is the g-force on the head. The data file looks like this:

```
"time since impact (ms)","g force"
2.4,0
2.6,-1.3
3.2,-2.7
3.6,0
4,-2.7
6.2,-2.7
6.6,-2.7
6.8,-1.3
...
```

and you can see a plot of the data in Figure 1.

Implement basis function regression with ordinary least squares.[1] Some sample Python

[1]Note that the data clearly don't have fixed variance! There is obviously less variance on the left of the plot. Modeling such *heteroscedastic* data is beyond the scope of the course.

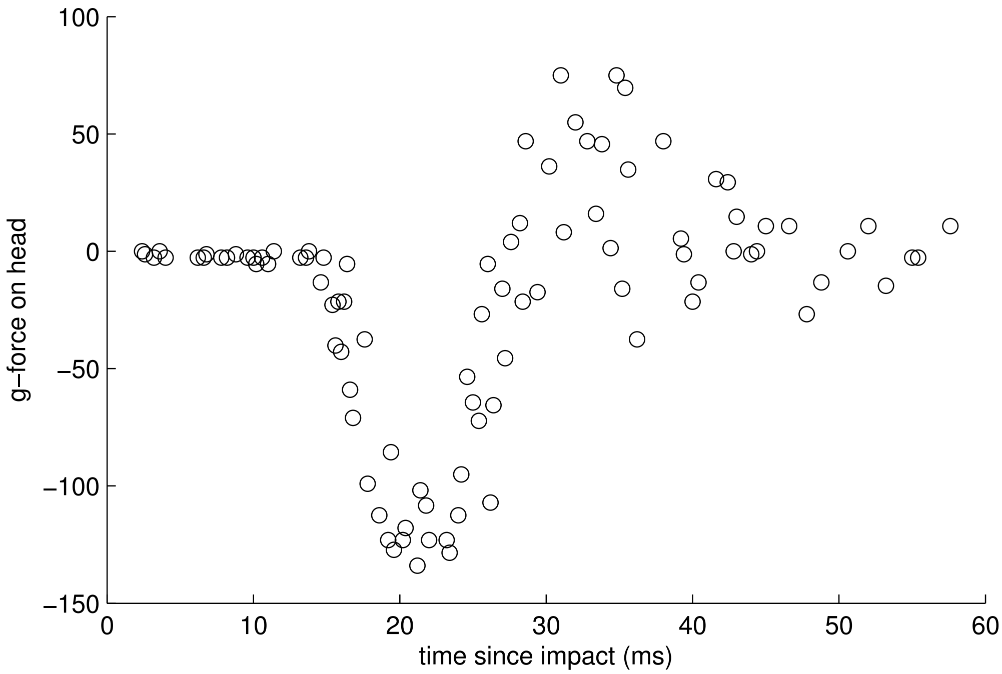

Figure 1: Motorcycle crash helmet data. The horizontal axis is time since impact and the vertical axis is force on the head.

code is provided in `linreg.py`. Plot the data and regression lines for the simple linear case, and for each of the following sets of basis functions:

(a) $\phi_j(x) = \exp\{-((x-10j)/5)^2\}$ for $j = 0, \ldots, 6$

(b) $\phi_j(x) = \exp\{-((x-10j)/10)^2\}$ for $j = 0, \ldots, 6$

(c) $\phi_j(x) = \exp\{-((x-10j)/25)^2\}$ for $j = 0, \ldots, 6$

(d) $\phi_j(x) = x^j$ for $j = 0, \ldots, 10$

(e) $\phi_j(x) = \sin\{x/j\}$ for $j = 1, \ldots, 20$

In addition to the plots, provide one or two sentences for each, explaining whether you think it is fitting well, overfitting or underfitting. If it does not fit well, provide a sentence explaining why.

5. Explore Google Colab and Scikit-learn [10pts]

The objective of this problem is to gain some experience with Google Colaboratory and Scikit-learn.

1. Navigate to https://colab.research.google.com/. Complete the Welcome To Colaboratory tutorial. Nothing to submit for this. Just do the rest of the problem in Colab and submit the notebook.

2. In a new Colab notebook, load the diabetes regression dataset, see https://scikit-learn.org/stable/datasets/toy_dataset.html. Split the dataset into a training set, a validation set, and a test set.

3. Train linear regression https://scikit-learn.org/stable/modules/linear_model.html and k-nearest neighbors https://scikit-learn.org/stable/modules/neighbors.html models. Select the k-nearest neighbors k parameter using the validation set.

4. Report metrics that summarize algorithm performance on the test set https://scikit-learn.org/stable/modules/model_evaluation.html. R^2 and mean squared error are good metrics to include.

6. Calibration [0pt]

Approximately how long did this homework take you to complete?