XCS229 PS1 Question 2

Siheon Lee

May 2025

1 2(a)

Since $h_\theta(x) = \theta^T \hat{x}$

$$J(\theta) = \frac{1}{2}\sum_{i=0}^{n}(h_\theta(x^{(i)}) - y^{(i)})^2 = \frac{1}{2}\sum_{i=0}^{n}(\theta^T \hat{x}^{(i)} - y^{(i)})^2 = \frac{1}{2}(\theta^T \hat{x} - y)(\theta^T \hat{x} - y)$$

Differentiating this objective, we get:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \sum_{i=0}^{n} \frac{\partial}{\partial \theta}\frac{1}{2}(\theta^T \hat{x}^{(i)} - y^{(i)})^2 \quad \text{by definition} \\
&= 2 \cdot \frac{1}{2}\sum_{i=0}^{n}(\theta^T \hat{x}^{(i)} - y^{(i)}) \cdot \frac{\partial}{\partial \theta}(\theta^T \hat{x}^{(i)} - y^{(i)}) \\
&= \sum_{i=0}^{n}(\theta^T \hat{x}^{(i)} - y^{(i)}) \cdot \hat{x}^{(i)} \\
&= (\theta^T \hat{x} - y)\hat{x}
\end{aligned}$$

The gradient descent update rule is

$$\theta := \theta - \lambda \nabla_\theta J(\theta)$$

which reduces to:

$$\theta := \theta - \lambda \sum_{i=0}^{n}(\theta^T \hat{x}^{(i)} - y^{(i)}) \cdot \hat{x}_j^{(i)}$$

2 2(c)

In the plot, the x-axis ranges from -2π to $+2\pi$ and represents the range of features from train_x. Train_y then represents the corresponding expected outputs, as represented by the blue dots. Now through differing kernels, we made

differing predictions based on plot_x data. While $k = 1$, which is $\begin{bmatrix} 1 \\ x \end{bmatrix}$ and $k = 2$'s fits are basically linear with negative slopes since low-degree polynomial features can only model so much, as we increasingly get larger kernels ($k = 5$, $k = 10$, or even $k = 20$, which is $\begin{bmatrix} 1 \\ x \\ \vdots \\ x^{20} \end{bmatrix}$, they start to follow a vaguely sinusoidal path along that of the original train_y data. I'd say that $k = 5$ and $k = 10$ seem to be the best fits with a balance between features and data while $k = 20$ with its bends suggest overfitting, being disturbed by even the slightest noise in the train_x data.

3 (2e)

Now that sine and polynomial features are both in the feature set, the create_sin implementation shows a great fit from $k = 1$ all the way through to $k = 20$. Polynomials naturally aren't great a capturing periodicity because they have a limited number of derivatives while the sin() function can be differentiated endlessly. However, $k = 20$ still shows the characteristic bends of overfitting due to the polynomial part of the feature set overpowering the sine.

4 (2g)

It is certainly true that all of these fits from $k = 1$ to $k = 20$ could potentially be accurate for the actual dataset. However, we can see how as k increases beyond a certain threshold (it seems to be 10 in this case), the model overfits the data, and we get these drastic curves that have varies wildly with the other regression attempts at lower kernel sizes. This happens because now even the slightest change in $x^{10}, x^{15}, \text{or} x^{20}$ could drastically affect the predicted outcome of the regression.

Request Penalty Waiver
Link to Full Solution:
https://drive.google.com/drive/folders/1TQqgASEPs3KTLf1dsdGD6ZhuvuWe1wBj?usp=sharing