

## STATS 665 Homework 2

Lecturer: Leon Lixing Yu

Scribe: Leon Lixing Yu

## 1 Problem 1

1.1 closed form of  $\hat{\beta}$  $\hat{\beta}$  is given by:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2} \|X\beta - y\|_2^2 + \lambda \|\beta\|_2^2$$

In order to find *argmin*, we need to take the derivative of the loss function and make it equal to 0. In this assignment, I will use the notation,  $x_j^{(i)}$ , to denote that for the given data  $(x^{(i)}, y^{(i)})_{i=1}^n$ ,  $x_j^{(i)}$  is the  $j$ th element of  $x^{(i)}$ , where  $x^{(i)} \in \mathbb{R}^d, y \in \mathbb{R}, X \in \mathbb{R}^{d \times n}, Y \in \mathbb{R}^n$ . Therefore, we have:

$$\begin{aligned} \frac{d}{d\beta_j} l(\beta|X, Y) &= \left( \sum_{i=1}^n \left( \sum_{j=1}^d x_j^{(i)} \beta_j \right) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^d \beta_j^2 \Big|' = 0 \\ &\Rightarrow 2X^T(X\beta - Y) + 2\lambda(I_d)\beta = 0 \\ &\Rightarrow \hat{\beta} = (X^T X + \lambda(I_d))^{-1} X^T Y \end{aligned}$$

1.2 Find a simple expression for  $\|\hat{\beta} - \beta^*\|$ 

FIXME

1.3 Find a closed form of  $\hat{f}$ 

Notation Note: Following the previous convention, I am using  $(x^{(i)}, y^{(i)})_{i=1}^n$  instead of  $(x_i, y_i)_{i=1}^n$  to represent the dataset. In this question, I use  $f(x^{(i)})$  and  $\langle f, \phi(x^{(i)}) \rangle$  interchangeably. We know that:

$$f \in \mathcal{H} \Rightarrow f(x^{(i)}) \in \mathcal{H} \Rightarrow \langle f, \phi(x^{(i)}) \rangle \in \mathcal{H}$$

With the notation given, we have equation:

$$\hat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^n (y^{(i)} - \langle f, \phi(x^{(i)}) \rangle_{\mathcal{H}})^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Using representer theorem, we have:

$$f = \sum_{i=1}^n \alpha^{(i)} \phi(x^{(i)}) = \sum_{i=1}^n \alpha^{(i)} k(x^{(i)}, \cdot)$$

The equation above means  $f$  is a linear combination of feature space, mapping of points. substitute the relation above into the original  $\hat{f}$  equation, we have:

$$\sum_{i=1}^n (y^{(i)} - \langle f, \phi(x^{(i)}) \rangle_{\mathcal{H}})^2 + \lambda \|f\|_{\mathcal{H}}^2 = \|Y - K\alpha\|^2 + \lambda \alpha^T K \alpha$$

Taking derivative over  $\alpha$  and make it equal to 0 to get argmin:

$$\frac{d}{d\alpha} (\|Y - K\alpha\|^2 + \lambda \alpha^T K \alpha) = 0$$

$$\Rightarrow 2K(Y - K\alpha) + 2\lambda \mathbf{I}_d K \alpha = 0$$

$$\Rightarrow (K + \lambda \mathbf{I}_d) \alpha = 2Y$$

$$\Rightarrow \hat{\alpha} = (K + \lambda \mathbf{I}_d)^{-1} Y$$

recall:

$$f = \sum_{i=1}^n \alpha^{(i)} \phi(x^{(i)}) = \sum_{i=1}^n \alpha^{(i)} k(x^{(i)}, \cdot)$$

our  $\hat{f}$  is then:

$$\hat{f} = K^T \hat{\alpha} = K^T (K + \lambda \mathbf{I}_d)^{-1} Y$$

In which  $K$  is the Kernel matrix W.R.T  $X$

## 1.4 implement the solution of $\hat{f}$ in matlab

The code for this part is attached in **Appendix A: problem 1 Code**, I use Gaussian Kernel because it is the first one I tried and it worked pretty well. I have attached a few graphs to show the differences between real label value and decision values got from  $\hat{f}$ .

To have a perfect fit, I adjust the values of  $\lambda$  and  $\sigma$ . Multiple attempts are shown below with descriptions.

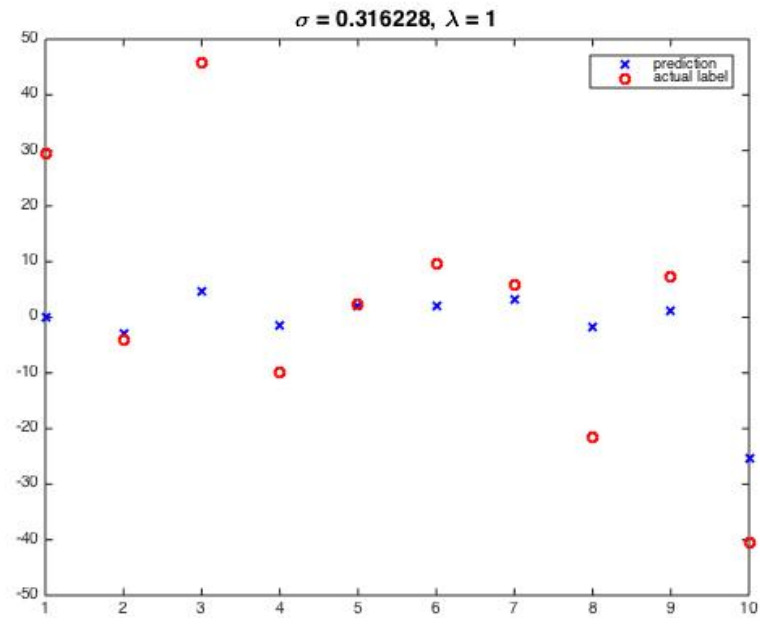


Figure 1: Red circle indicates labels, blue cross is the  $\hat{f}$  value.  $n = 10, \lambda = 1, \sigma^2 = 0.1$  in this test case. I see that when  $x^{(i)}$  is close to decision boundary, 0, it is more accurate in this setting. It is an underfit case.

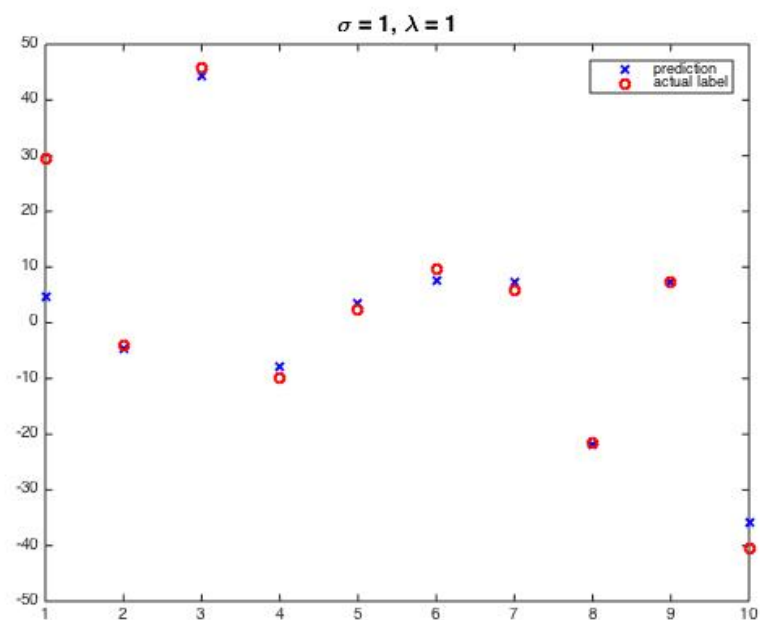


Figure 2:  $\lambda = 1, \sigma = 1$  in this test case. I see that most of the  $x^{(i)}$  can be predicted accurately with two exceptions at the end of the decision boundary

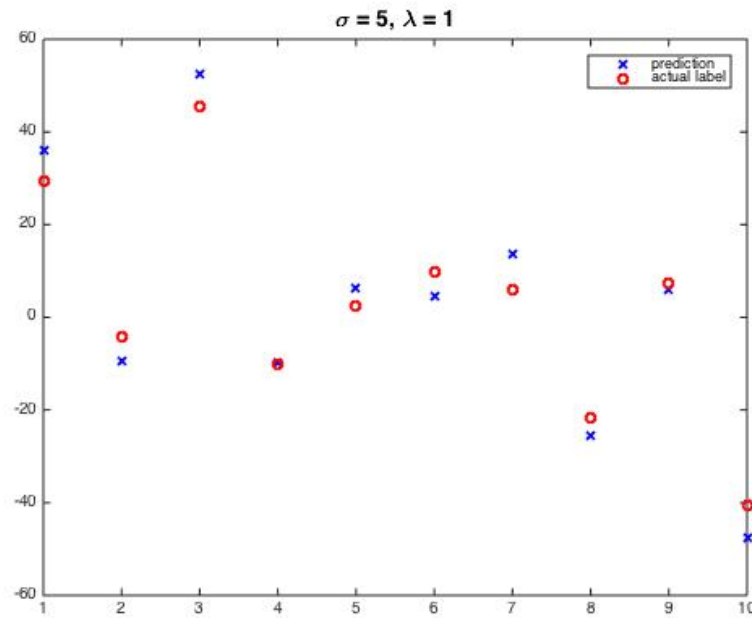


Figure 3:  $\lambda = 1, \sigma = 5$  in this test case. Though the  $x^{(i)}$  at both ends are better predicted, the average accuracy has dropped.

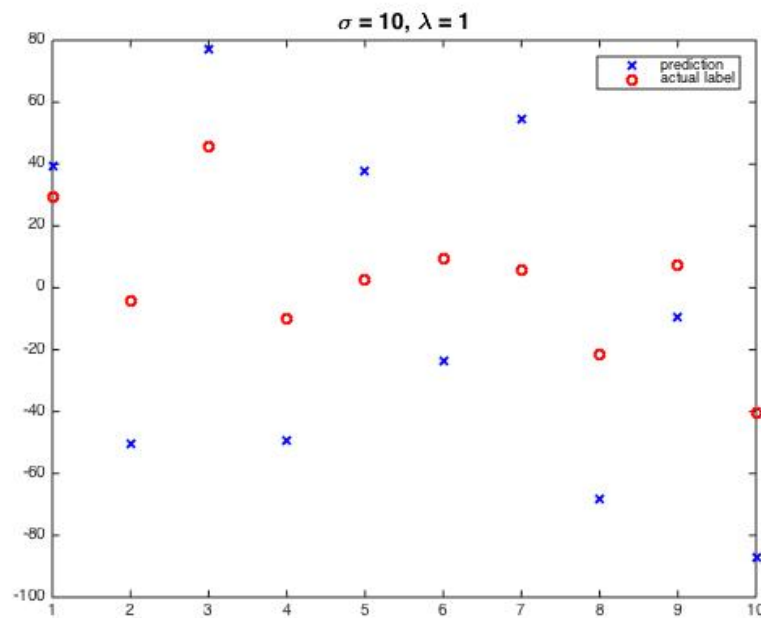


Figure 4:  $\lambda = 1, \sigma = 10$ . All  $x^{(i)}$  are fitted badly. It is clearly a case of over fitting.

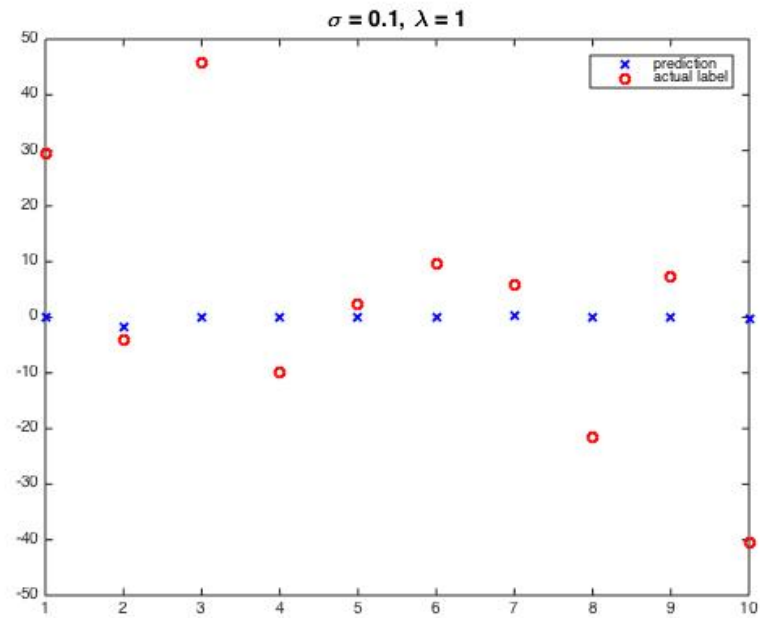


Figure 5:  $\lambda = 1, \sigma = 0.1$ . It is clearly a case of under fit. It is way under fit so that the decision boundary looks like a straight line.

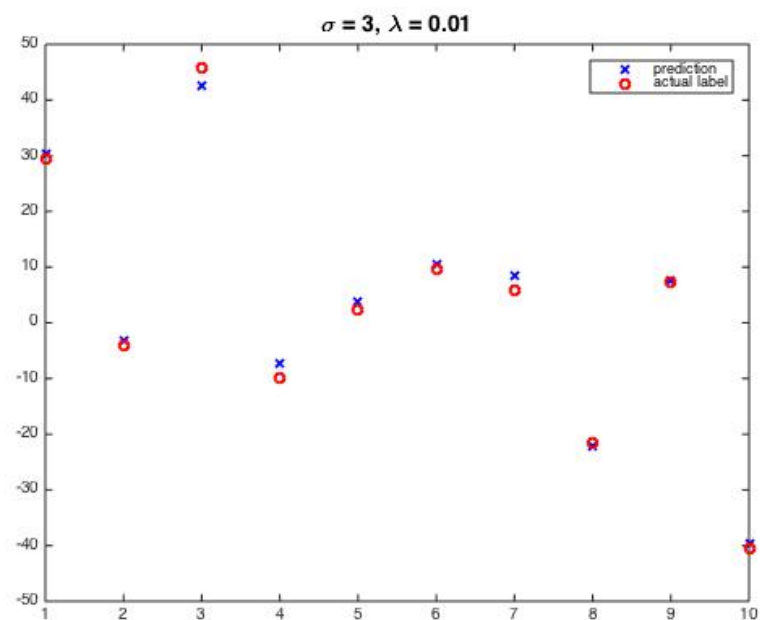


Figure 6:  $\lambda = 1, \sigma = 0.1$ . It is clearly a case of under fit. It is way under fit so that the decision boundary looks like a straight line.

## 2 Problem 2

## 3 Problem 3

### 3.1 Show $k(x, y)$ is a valid kernel

We know that kernel is a function that maps  $\chi \times \chi \rightarrow \mathbb{R}$ . We also know that kernel is valid if and only if for  $x_{i=1}^n \in \chi$ ,  $K_{ij} = k(x_i, x_j)$  is PSD. We need to prove these two statements.

Proving  $k(x, y)$  maps  $\chi \times \chi \rightarrow \mathbb{R}$ :

Assuming:

$$k_1(x, y) = \langle \Phi_1(x), \Phi_1(y) \rangle$$

$$k_2(x, y) = \langle \Phi_2(x), \Phi_2(y) \rangle$$

Since  $k_1(x, y)$  and  $k_2(x, y)$  are valid kernel, they both maps  $\chi \times \chi \rightarrow \mathbb{R}$ . we now have

$$\begin{aligned} k(x, y) &= \langle \Phi_1(x), \Phi_1(y) \rangle + \langle \Phi_2(x), \Phi_2(y) \rangle \\ \Rightarrow k(x, y) &= (\Phi_1(x)^T \Phi_1(y)) + (\Phi_2(x)^T \Phi_2(y)) \\ &\Rightarrow k(x, y) \in \mathbb{R} \end{aligned}$$

We also know that  $(x, y) \in \mathbb{R}^x$ , so  $k(x, y)$  maps  $\chi \times \chi \rightarrow \mathbb{R}$ .

Proving  $k(x, y)$  is PSD:

$$K_{ij} = k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle = \Phi(x_i)^T \Phi(x_j)$$

$$\begin{aligned} v^T K v &= \sum_i^n \sum_j^n v_i K_{ij} v_j \\ &= \sum_i^n \sum_j^n v_i \Phi(x_i)^T \Phi(x_j) v_j \\ &= \sum_i^n \sum_j^n v_i \sum_l^n \phi_l(x_i)^T \phi_l(x_j) v_j \\ &= \sum_l^n \sum_i^n \sum_j^n v_i \phi_l(x_i)^T \phi_l(x_j) v_j \\ &= \sum_l^n \left( \sum_i^n v_i \phi_l(x_i) \right)^2 \geq 0 \end{aligned}$$

end of the story