Statistical Machine Learning – Homework 3 Solution

Credit to Shuaiwen Wang

Problem 1

Proof. 1. Let $y = y_1 = -y_2$ and $x = x_{11} = x_{12} = -x_{21} = -x_{22}$, then we have the ridge regression equivalent to minimizing

$$R(\beta) = \Big(\big(y_1 - x_{11}\beta_1 - x_{12}\beta_2\big)^2 + \big(y_2 - x_{21}\beta_1 - x_{22}\beta_2\big)^2\Big) + \lambda\|\beta\|_2^2 = \big(y - x(\beta_1 + \beta_2)\big)^2 + \lambda(\beta_1^2 + \beta_2^2)$$

2. Let $(\hat\beta_1, \hat\beta_2)$ be a pair of solution of the above problem and WLOG assume $\hat\beta_1 < \hat\beta_2$. Pick $0 < \epsilon < \frac{\hat\beta_2 - \hat\beta_1}{2}$ and set $(\tilde\beta_1, \tilde\beta_2) = (\hat\beta_1 + \epsilon, \hat\beta_2 - \epsilon)$. then we have $\tilde\beta_1 + \tilde\beta_2 = \hat\beta_1 + \hat\beta_2$. However we have

$$R(\hat\beta) - R(\tilde\beta) = \hat\beta_1^2 + \hat\beta_2^2 - (\hat\beta_1 + \epsilon)^2 - (\hat\beta_2 - \epsilon)^2 = 2\epsilon\big(\hat\beta_2 - \hat\beta_1 - \epsilon\big) > 0$$

which is contradicted with the minimum of $\hat\beta$.

3. Lasso is to minimize

$$L(\beta) = \big(y - x(\beta_1 + \beta_2)\big)^2 + \lambda(|\beta_1| + |\beta_2|)$$

4. Let $(\hat\beta_1, \hat\beta_2)$ be a pair of solution of Lasso problem. Then $\hat\beta$ minimize $L(\beta)$.

- If $\text{sign}(\hat\beta_1) \neq \text{sign}(\hat\beta_2)$. WLOG, assume $\hat\beta_1 < 0 < \hat\beta_2$, then we can pick $\epsilon > 0$ small enough, such that $\hat\beta_1 + \epsilon < 0 < \hat\beta_2 - \epsilon$. Let $\tilde\beta = (\hat\beta_1 + \epsilon, \hat\beta_2 - \epsilon)$. Then we have

  $$L(\tilde\beta) = (y - x(\hat\beta_1 + \hat\beta_2))^2 + \lambda(\hat\beta_2 - \hat\beta_1 - 2\epsilon) < L(\hat\beta)$$

  This is contradicted with the minimum of $\hat\beta$. Thus this situation cannot happen.

- If it is not the above case, then we have $L(\hat\beta) = (y - x(\hat\beta_1 + \hat\beta)_2)^2 + \lambda|\hat\beta_1 + \hat\beta_2|$. Thus any other pair $\tilde\beta$ which does not belong to the first case, as long as $\tilde\beta_1 + \tilde\beta_2 = \hat\beta_1 + \hat\beta_2$, we have $L(\tilde\beta) = L(\hat\beta)$. Which means the solution is not unique. This will only be untrue when $\hat\beta_1 = \hat\beta_2 = 0$. In this case, the solution is unique.

$\square$

Problem 2

Proof. First we clarify one thing. To find $\hat{g}_1$, we need to search over $C^{(3)}(\mathbb{R})$, the set of functions which are three times differentiable. Similarly for $\hat{g}_2$, we need to search over $C^{(4)}$.

Follow a similar steps as Ex. 5.6 on the textbook, it is not hard to prove that $\hat{g}_1$ must be a order five(degree 4) natural spline. Similarly $\hat{g}_5$ must be an order six(degree 5) natural spline. Besides, we have

$$\min_{g\in C^(3)(\mathbb{R})}\Big(\sum_{i=1}^{n}(y_i-g(x_i))^2+\lambda\int\left[g^{(3)}(x)\right]^2dx\Big)\Leftrightarrow\min_{g\in C^(3)(\mathbb{R})}\sum_{i=1}^{n}(y_i-g(x_i))^2,\ \text{ subject to }\int\left[g^{(3)}(x)\right]^2dx\le t_\lambda$$

$$\min_{g\in C^(4)(\mathbb{R})}\Big(\sum_{i=1}^{n}(y_i-g(x_i))^2+\lambda\int\left[g^{(4)}(x)\right]^2dx\Big)\Leftrightarrow\min_{g\in C^(4)(\mathbb{R})}\sum_{i=1}^{n}(y_i-g(x_i))^2,\ \text{ subject to }\int\left[g^{(4)}(x)\right]^2dx\le t_\lambda$$

1. As $\lambda\to\infty$, it is equivalent to set $t_\lambda\to 0$. As a result, we will have $g^{(3)}(x)$, $g^{(4)}(x)=0$ almost everywhere. If we use $\mathcal{P}_k$ to denote the collection of degree k polynomials. Then

$$\hat{g}_1=\operatorname*{argmin}_{g\in\mathcal{P}_2}\sum_{i=1}^{n}\big(y_i-g(x_i)\big)^2$$

$$\hat{g}_2=\operatorname*{argmin}_{g\in\mathcal{P}_3}\sum_{i=1}^{n}\big(y_i-g(x_i)\big)^2$$

Because $\mathcal{P}_2\subset\mathcal{P}_3$, $\hat{g}_2$ has a smaller training error;

2. When it goes to test error, it depends on the true model. If the true model is quadratic, of course $\hat{g}_1$ will be better; if the true model is a cubic polynomial, $\hat{g}_2$ will be better. In general, $\hat{g}_2$ can be easier to overfit compared with $\hat{g}_1$;

3. Here essentially we need to consider the case $\lambda\to 0$ since $\lambda=0$ is trivial.

As $\lambda\to 0$, $\hat{g}_1$ will become the degree 4 piece polynomial which pass through all y_i and have up to three order continuous derivatives at all the knots. Similarly $\hat{g}_2$ will be the degree-5 fit. Thus the training error for both are 0. When it goes to test error, it depends on the true model. On average 5 degree polynomial fit tends to be more spiky if the distance between two consecutive knots are large. Thus it *may* present large variance.

□

Problem 3

Proof. Notice that the parameter `cost` in the function `svm()`, `tune.svm()` corresponds to the margin parameter in SVM. The `gamma` controls the radius of RBF kernel. The tuning results are shown in Figure

1. The optimal cost for linear- SVM is $C = 0.029$; the optimal cost and radius for RBF-SVM is $C = 3, \gamma = 0.016$.

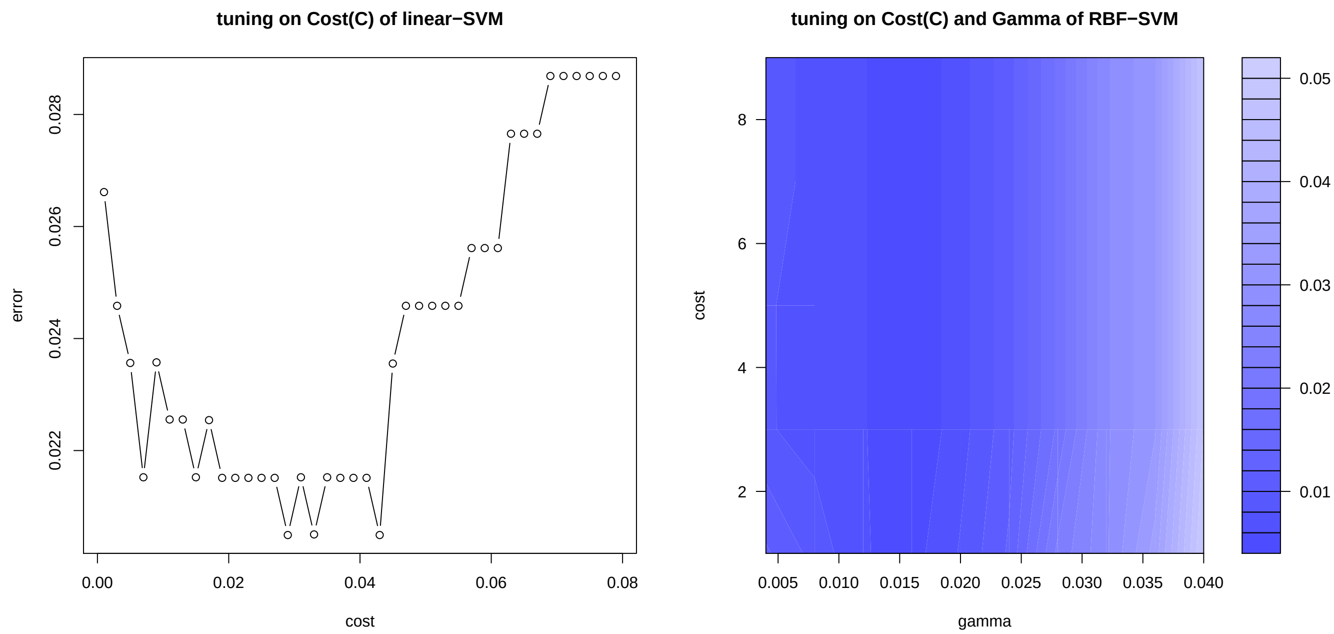

Figure 1: 10-fold CV run by `tune.svm()`. The optimal parameter for linear-SVM is $C = 0.029$, the optimal parameter for RBF-SVM is $C = 3$, $\gamma = 0.016$.

After fitting the best model using their own best tuning, we have the classification errors, linear-SVM: 0.0122, RBF-SVM: 0.0041. □

Obviously the non-linear one is better. Notice that the fitting of RBF-SVM is not sensitive to the tuning of Cost parameter.