MTH210: Lab 12

Cross-validation

The data is randomly split into K roughly equal-sized parts. For any kth split, the rest of the $K-1$ parts make up the training set and the model is fit to the training set. We then estimate the prediction error for each element in the kth part. Repeating this for all $k = 1, 2, \dots, K$ parts, we have an estimate of the prediction error.

Let $\kappa : \{1, \dots, N\} \mapsto \{1, \dots, K\}$ indicates the partition to which each ith observation belongs. Let $\hat{f}^{-\kappa(i)}(x)$ be the fitted function for the $\kappa(i)th$ partition removed. Then, the estimated prediction error is

$$\text{CV}_K(\hat{f}, \gamma) = \frac{1}{K}\sum_{k=1}^{K} \frac{1}{n/K} \sum_{i \in k^{\text{th}}\text{split}} L(y_i, \hat{f}^{-\kappa(i)}(x_i, \gamma)) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}^{-\kappa(i)}(x_i, \gamma))\,.$$

The chosen model is the one with γ such that

$$\theta_{\text{chosen}} = \arg\min_{\gamma} \left\{\text{CV}_K(\hat{f}, \gamma)\right\}$$

The final model is $\hat{f}(X, \gamma_{\text{chosen}})$ fit to all the data.

Points:

- For small K, the bias in estimating the true test error is large since each training data is quite different from the given dataset $\mathcal{D}$.
- The computational burden is lesser when K is small.

Usually, for large datasets, 10-fold or 5-fold CV is common. For small datasets, LOOCV is more common.

Now, let's try a few implementations.

1. Recall the regression model: $Y = X\beta + \epsilon$, where $\epsilon \sim N_n(0, \sigma^2 I_n)$. In this example subsection, we will focus our attention only on ridge regression estimators. Similar cross-validations can be done for bridge regression. Recall the ridge objective function for a given λ is

$$Q_\lambda(\beta) = \frac{(y - X\beta)^T(y - X\beta)}{2} + \frac{\lambda}{2}\beta^T\beta\,.$$

   Recall, the ridge estimator of β is

$$\hat{\beta}_\lambda = (X^T X + \lambda I_n)^{-1} X^T y \,.$$

Here, the solution for β depends on the value of λ. We want to choose λ so that prediction error is minimized. We choose the squared error loss function. To choose the best model in this case, we set a vector of $\lambda : \lambda_1, \dots, \lambda_m$.

Since closed-form solution for the ridge-regression estimator is available, cross-validation steps will be fairly fast. Thus we choose LOOCV. For each λ_i, we will implement LOOCV and estimate the prediction error. Whichever λ minimzies the prediction error, will be the chosen λ.

Consider the `mtcars` dataset in `R`. The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). The response is the miles per gallon of the 32 cars. Run a `?mtcars`} in `R` to learn more about the dataset.

We will choose

$$\lambda \in \{i \in (-8, 8) : 10^i\},$$

where i will increase in increments of .1. This makes it a large grid of λs.

```
###################################################
## Choosing lambda in ridge regression
## using LOOCV
## dataset is mtcars
###################################################
data(mtcars)
y <- mtcars$mpg # response

# adding intercept
X <- cbind(1, as.matrix(mtcars[ ,-1]))

n <- dim(X)[1]
p <- dim(X)[2]

# vector of lambdas
lam.vec <- c(10^(seq(-8, 8, by = .1)))
# Will store CV error in this.
CV.error <- numeric(length = length(lam.vec))
```

Now things are setup, to implement LOOCV.

```r
# for each lambda we will do ridge
for(l in 1:length(lam.vec))
{
  track.cv <- 0
  lam <- lam.vec[l]
  for(i in 1:n)
  {
    # Making training data
    X.train <- X[-i,] # removing ith X
    y.train <- y[-i]  #removing ith y

    # fitting model for training data
    beta.train <- ...
    # test error
    track.cv <- track.cv + ...
  }
  CV.error[l] <- track.cv/n
}
```

Now, we have calculated the CV errors for all λs. Now we can find the best one:

```r
(chosen.lam <- lam.vec[which.min(CV.error)] )
```

Which the chosen λ, λ_{chosen}, we should refit the model again on the whole data, to get the final β estimates.

```r
beta.final <- solve(t(X) %*% X + chosen.lam*diag(p)) %*% t(X) %*% y
```

2. Repeat the above process for 5-fold and 10-fold cross-validation. The following code will be useful. It splits the index set $\{1, 2, \dots, n\}$, into K different folds:

```r
permutation <- sample(1:n, replace = FALSE)
K <- 4

# Making a list of indices for each split
test.index <- split(permutation, rep(1:K, length = n, each = n/K))
```

3. For the above dataset, what is the best value of α and λ that minimizes test error for Bridge regression?

4. Recall the ridge logistic regression model you fit in Assignment 3. Find the optimal value of λ for the `titanic` dataset:

```r
titanic <- read.csv("https://dvats.github.io/assets/titanic.csv")
```