

# STATS305A - Lecture 16

John Duchi  
Scribed by Michael Howes

11/16/21

## Contents

<b>1</b>	<b>Announcements</b>	<b>1</b>
<b>2</b>	<b>Validation</b>	<b>1</b>
2.1	Hold out set . . . . .	2
2.2	Cross validation . . . . .	2
2.3	Using cross validation for model selection . . . . .	3
2.4	Leave one out cross validation . . . . .	3
<b>3</b>	<b>Permuation testing</b>	<b>5</b>

## 1 Announcements

- HW3 due Friday.
- Etude 3 due Friday and corrections due Monday.

Today we will be discussing two randomized methods - cross validation and permutation tests.

## 2 Validation

Suppose we have a fitting method  $\hat{\beta}$ . We'd like to know how  $\hat{\beta}$  is going perform on future data. In a typical case we would have a hyperparameter  $\lambda$  and we want to pick the “best”  $\lambda$ . We have seen many examples of different things  $\lambda$  could represent, such as:

- We could have  $\lambda$  equal to the regularization parameter in ridge regression.
- We could also have  $\lambda$  equal to the number of coordinates in PC regression, forward stepwise regression or boosting.

Define

$$R(b) = \mathbb{E}[L(Y_{n+1}, X_{n+1}^T b)],$$

for some loss function  $L : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ . The quantity  $R(b)$  is the out-of-sample risk of the linear predictor  $\hat{y} = x^T b$ . We wish to estimate  $R(b)$  and choose  $\lambda$  which minimizes  $R(\hat{\beta})$ . There are two approaches that we will discuss, using a hold out set or performing cross validation.

## 2.1 Hold out set

Hold out a subset of our data, call it the *validation data*. Fit the model on *training data* (which is all the data apart from the validation set). We can then calculate the average loss on the independent validation set. There are two issues with this

- It can be a little high variance.
- We are not using all the data.

## 2.2 Cross validation

The idea behind cross validation is to split our data into  $k$  equal sized partitions which we call *folds*. For each fold we fit on the remaining  $k - 1$  folds and then evaluate the model on the held out fold.

More mathematically, let  $J(i)$  be the set of indices in fold  $i$ . Define

$$\hat{\beta}^{-J(i)} = \text{model fit on } (X, Y) \text{ but with the indices in } J(i) \text{ removed.}$$

The empirical risk of using a linear predictor  $b$  on fold  $i$  is

$$\hat{R}_i(b) = \frac{1}{|J(i)|} \sum_{j \in J(i)} L(y_j, x_j^T b),$$

where  $|J(i)|$  is the cardinality of  $J(i)$  which is typically  $\frac{n}{k}$  (this happens when all the folds are the exact same size). Then define the  $k$ -fold cross validation error to be

$$CV(k) = \frac{1}{k} \sum_{i=1}^k \hat{R}_i(\hat{\beta}^{-J(i)}).$$

Two natural questions are:

- What is  $CV(k)$  approximating?
- What do we use  $CV(k)$  for?

Let  $\mathcal{T}$  be a training set  $(X, Y) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$ . What we'd like to know is

$$\text{Err}(\mathcal{T}) = R(\hat{\beta}(\mathcal{T})),$$

where  $\hat{\beta}(\mathcal{T})$  is the model trained using our test set  $\mathcal{T}$ . We'd like to know the expected error of using  $\hat{\beta}(\mathcal{T})$  on new data. Our training set is considered to be random and thus we can define

$$\text{Err}_n = \mathbb{E}[\text{Err}(\mathcal{T})],$$

where the expectation is taken over all training sets of size  $n$ . If we assume that our data is i.i.d. and the  $k$  folds all have size  $\frac{n}{k}$ , then

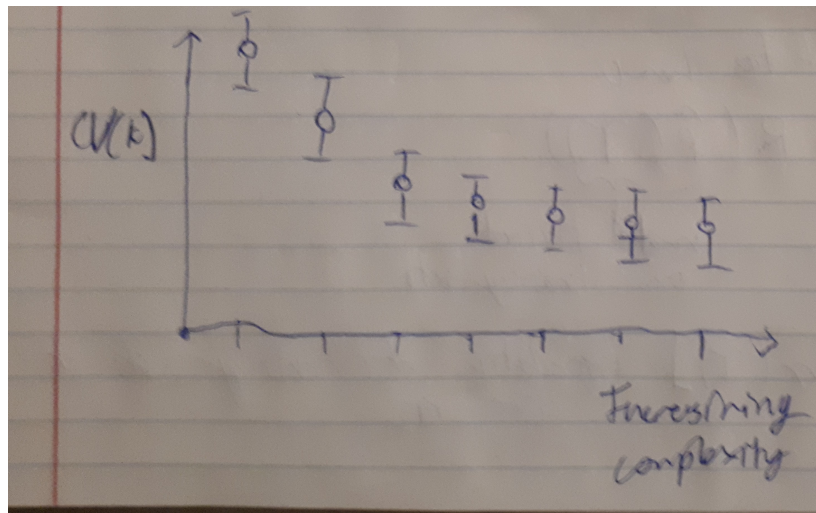
$$\begin{aligned} \mathbb{E}[CV(k)] &= \mathbb{E}[\hat{R}_k(\hat{\beta}^{-J(k)})] \\ &= \mathbb{E} \left[ \mathbb{E} \left[ \hat{R}_k(\hat{\beta}^{-J(k)}) \mid J(k) \right] \right] \\ &= \mathbb{E}[R(\hat{\beta}(\text{training set of size } n - k/n))] \\ &= \text{Err}_{\frac{(k-1)n}{k}} + \end{aligned}$$

Thus we have an issue  $\mathbb{E}[CV(k)]$  is not unbiased for  $\text{Err}_n$ . There are some “solutions”

- Just ignore the bias.
- Introduce correction terms. These can be a bit complicated and will depend on the loss function  $L$  and the choice of model fitting procedure. See John's notes for some details in special cases.
- If we take  $k$  large,  $\text{Err}_{\frac{(k-1)n}{k}}$  will be close to  $n$ . We will discuss this more in the leave one out section.

### 2.3 Using cross validation for model selection

In practice we calculate  $CV(k)$  for various values of  $\lambda$  and then compare  $CV(k)$  across these values. If we plot  $CV(k)$  we tend to see pictures that look like this:



As we increase complexity (ie increase  $\frac{1}{\lambda}$  in ridge regression or increase the number of components in PCA regression/forward stepwise),  $CV(k)$  decreases. We can put error bars on  $CV(k)$  by calculating the empirical standard deviation of  $CV(k)$ . This gives us the error bars in the plot. One may choose  $\lambda^*$  is to take the least complex model for which all the “more complex” models have  $CV(k)$  within one standard error. We then fit a model using the full data and this chosen value of  $\lambda^*$ .

### 2.4 Leave one out cross validation

Returning to the idea of taking  $k$  large, we can set  $k = n$ . This is nice because then  $CV(k)$  is unbiased for  $\text{Err}_{\frac{n-1}{n}}$  which should be close to  $\text{Err}_n$ . When  $k = n$  each fold is a single data point and so we set  $J(i) = \{i\}$ . There are two issues:

- This may be computationally challenging we have to fit  $n$  models for each model fitting procedure.
- The variance of  $CV(n)$  may be larger than when  $k = 5$  or  $k = 10$  (this is still disputed).

In ordinary least squares we have computational tricks that means that  $n$ -fold cross validation can be done quickly. Suppose that we are using the model  $y = X\beta + \varepsilon$  to fit  $\hat{\beta}$ . That is  $\hat{\beta} = (X^T X)^{-1} X^T y$  and

$$\hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y = Hy.$$

If we let  $H_{ii}$  be the leverage scores of the model (the diagonal entries of  $H$ ), then we have previously seen that

$$\hat{y}_i = H_{ii}y + (1 - H_{ii})\hat{y}_{-i},$$

where  $\hat{y}_{-i} = x_i^T \hat{\beta}^{-i}$  is the prediction for  $y_i$  given only  $(X_{-i}, y_{-i})$ . Thus we have

$$\hat{y}_{-i} = \frac{y_i}{1 - H_{ii}} \hat{y}_i - \frac{H_{ii}}{1 - H_{ii}} y_i.$$

And so

$$y_i - \hat{y}_{-i} = \frac{y_i - \hat{y}_i}{1 - H_{ii}}.$$

Thus for fitting  $\hat{\beta} = (X^T X)^{-1} X$  we have

$$CV(n) = \frac{1}{n} \sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{(1 - H_{ii})^2},$$

under square error loss. This method can be used when  $\lambda$  determines  $X$  and then  $\hat{\beta}$  is given by ordinary least square regression.

In general, computing  $\hat{\beta}^{-i}$  can sometimes be easy and sometimes very hard. Suppose that instead of squared error we fit a model with the more general loss

$$L_n(b) = \frac{1}{n} \sum_{i=1}^n l(y_i - x_i^T b).$$

A very effective strategy is to approximate  $L_n$  with a quadratic at  $\hat{\beta} = \operatorname{argmin}_b L_n(b)$ , then remove the  $i^{th}$  term and choose  $\hat{\beta}_{\text{quad}}^{-i}$  to be the value which minimizes the approximation. More specifically we have

$$L_n(b) \approx L_n(\hat{\beta}) - \frac{1}{n} \sum_{i=1}^n l'(y_i - x_i^T \hat{\beta}_i) x_i^T (b - \hat{\beta}) + \frac{1}{2n} \sum_{i=1}^n (b - \hat{\beta})^T l''(y_i - x_i^T \hat{\beta}_i) x_i x_i^T (b - \hat{\beta}).$$

Define  $\hat{\varepsilon}_i = y_i - x_i^T \hat{\beta}$ , then

$$\begin{aligned} L_{-i}(b) &= \frac{1}{n-1} \sum_{j \neq i} L_j(y_j - x_j^T b) \\ &\approx \frac{1}{n-1} \sum_{j \neq i} L_j(\hat{\varepsilon}_j) - \frac{1}{n-1} \sum_{j \neq i} l'(\hat{\varepsilon}_j) x_j^T (b - \hat{\beta}) + \frac{1}{2(n-1)} \sum_{j \neq i} (b - \hat{\beta})^T l''(\hat{\varepsilon}_j) x_j x_j^T (b - \hat{\beta}) \\ &= c + g_{-i}^T b + \frac{1}{2} b^T A_{-i} b, \end{aligned}$$

where

$$g_{-i} = -\frac{1}{n-1} \sum_{j \neq i} \left( l'(\hat{\varepsilon}_j) x_j + l''(\hat{\varepsilon}_j) x_j x_j^T \hat{\beta} \right),$$

and

$$A_{-i} = \frac{1}{n-1} \sum_{j \neq i} l''(\hat{\varepsilon}_j) x_j x_j^T.$$

The minimizer of the quadratic approximation is

$$\hat{\beta}_{\text{quad}}^{-i} = -A_{-i}^{-1} g_{-i}.$$

Note that the matrix  $A_{-i}$  differs from the Hessian of  $L_n(\hat{\beta})$  by a rank one update. Thus inverting  $A_{-i}$  can be done quickly provided that we have stored the inverse Hessian.

### 3 Permutation testing

We will leverage the fact that if two random variables have no relationship (ie  $X_i$  and  $Y_i$  are independent), then

$$(X_i, Y_i)_{i=1}^n \stackrel{\text{dist}}{=} (X_i, Y_{\pi(i)})_{i=1}^n,$$

where  $\pi : [n] \rightarrow [n]$  is a permutation of  $[n] = \{1, 2, \dots, n\}$ . Thus under independence, any statistic  $T_n = T((X_i, Y_i)_{i=1}^n)$  should have the same distribution under permutations i.e.

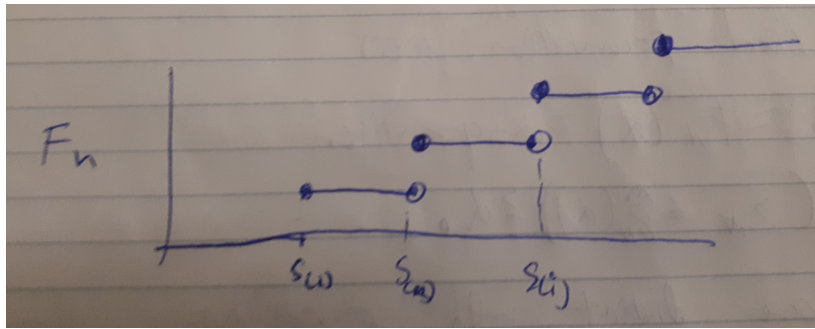
$$T_n \stackrel{\text{dist}}{=} T((X_i, Y_{\pi(i)})_{i=1}^n).$$

**Definition 1.** Let  $S_1, \dots, S_N$  be real values statistics and define the *empirical CDF* of  $s_i$  to be

$$F_N(t) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(S_i \leq t),$$

where  $\mathbb{I}(S_i \leq t)$  is 1 if  $S_i \leq t$  and 0 otherwise.

The function  $F_N$  is a step function that looks something like this:

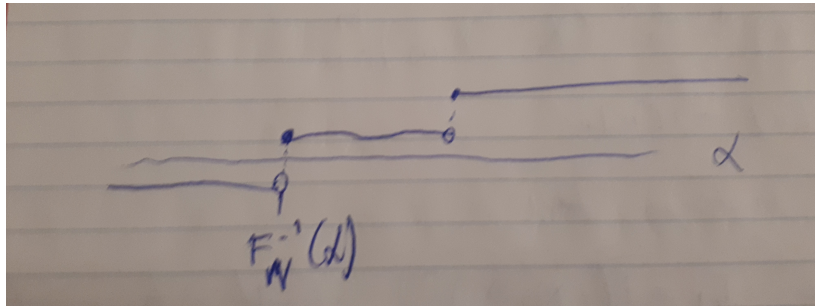


Note that  $F_N(t)$  is right continuous.

**Definition 2.** With  $S_i$  as before define the *quantile function* to be

$$\hat{q}_N(\alpha) = F_N^{-1}(\alpha) = \inf\{t \in \mathbb{R} : F_N(t) \geq \alpha\}.$$

The quantile function looks like this:



Note that  $F_N(F_N^{-1}(\alpha)) \geq \alpha$  for all  $\alpha$  and that  $F_N^{-1}(i/N) = S_{(i)}$  if

$$S_{(1)} < S_{(2)} < \dots < S_{(N)}.$$

**Theorem 1.** Suppose that the statistics  $S_1, \dots, S_N$  are exchangeable and so that for all permutations  $\pi$

$$(S_1, \dots, S_N) \stackrel{\text{dist}}{=} (S_{\pi(1)}, \dots, S_{\pi(N)}).$$

Let  $\hat{q}_N = F_N^{-1}$  be the quantile function. Then

$$\mathbb{P}(S_N \leq \hat{q}_N(\alpha)) \geq \alpha,$$

for all  $\alpha$ . If  $S_1, \dots, S_N$  are all distinct with probability one, then

$$\mathbb{P}(S_N \leq \hat{q}_N(\alpha)) \leq \alpha + \frac{1}{N}.$$

*Proof.* Note that for all  $i$ ,  $\mathbb{P}(S_i \leq \hat{q}_N(\alpha)) = \mathbb{P}(S_N \leq \hat{q}_N(\alpha))$  by exchangeability. Thus

$$\begin{aligned} \mathbb{E}[F_N(\hat{q}_N(\alpha))] &= \frac{1}{N} \sum_{i=1}^N \mathbb{P}(S_i \leq \hat{q}_N(\alpha)) \\ &= \mathbb{P}(S_N \leq \hat{q}_N(\alpha)). \end{aligned}$$

We also know that  $F_N(\hat{q}_N(\alpha)) \geq \alpha$  and so

$$\mathbb{P}(S_N \leq \hat{q}_N(\alpha)) = \mathbb{E}[F_N(\hat{q}_N(\alpha))] \geq \alpha.$$

And if  $S_i$  are distinct then the size of the jumps in  $F_N$  are exactly  $\frac{1}{N}$  and so

$$F_N(\hat{q}_N(\alpha)) \leq \alpha + \frac{1}{N}.$$

This thus implies

$$\mathbb{P}(S_N \leq \hat{q}_N(\alpha)) = \mathbb{E}[F_N(\hat{q}_N(\alpha))] \leq \alpha + \frac{1}{N}.$$

□

The upshot is that we should think of  $\mathbb{P}(S_N \leq \hat{q}_N(\alpha))$  as being equal to  $\alpha$  but there is some error due to discretization. The error has size  $\leq \frac{1}{N}$ .

Note that as a consequence we have  $\mathbb{P}(S_N > \hat{q}_N(\alpha)) \leq \alpha$  since

$$\mathbb{P}(S_N > \hat{q}_N(\alpha)) = 1 - \mathbb{P}(S_N \leq \hat{q}_N(\alpha)) \leq 1 - (1 - \alpha) = \alpha.$$

**Example 1.** Say we have i.i.d data  $(X_i, Y_i)$  and we want to test the null  $H_0 : X_i \perp\!\!\!\perp Y_i$ . Under the null,  $(X_i, Y_i) \stackrel{\text{dist}}{=} (X_i, Y_{\pi(i)})$ . We can fit  $\hat{\beta} = (X^T X)^{-1} X^T Y$  and  $\hat{\beta}_\pi = (X^T X)^{-1} X^T Y_\pi$  where

$$Y_\pi = \begin{bmatrix} Y_{\pi(1)} \\ \vdots \\ Y_{\pi(n)} \end{bmatrix}.$$

Suppose we do this for  $m - 1$  random permutations. Set  $S_m = \|\hat{\beta}\|_2^2$  (although we could use any function of  $\hat{\beta}$ ) and  $S_i = \|\hat{\beta}_\pi\|_2^2$ . We can then reject  $H_0$  if

$$S_m \geq \text{Quantile}_{1-\alpha}(S_{\pi_i}, S_m).$$

This test will have level  $\alpha$ .