Cross Validation

- can use to choose a model, M, or to simply estimate the generalization error of a model: $E_{\mathcal{D}\sim P^m,\, X,Y\sim P}\left[\ell(Y, f_M^{(\mathcal{D})}(X))\right]$

- given our data, we can construct a pretty good estimator $\hat{\varepsilon}_M$, of this error w/ the following algo:

(equivalent to GridSearchCV in SciKit)

For M in $\mathcal{M}$: can be a grid of params

For $k=1$ to K:

- Fit model w/ M and w/o k

$\hat{\varepsilon}_M^{(k)} = MSE_M^{(k)}$ or whatever other loss you want

should be reindented

$\Rightarrow \hat{\varepsilon}_M = \frac{1}{K}\sum_{k=1}^{K} \hat{\varepsilon}_M^{(k)}$

$M^* = \underset{M}{\arg\min}\left(\hat{\varepsilon}_M\right)$

• how to choose K?

2 extremes:

LOOCV

- low bias
  - training uses lots of data, so models are good and ~~have~~ do ~~not~~ not have ~~realistic~~ error biased to the high end

- high variance
  - all models use essentially the same data and are thus highly correlated. Since we average the errors in the end, the correlations adds a lot to the variance

2-Fold

- high bias
  - traing uses few data, so models aren't great and have error biased on the high end.

- low variance
  - low for opposite reason

A medium K (5 → 10) is usually preferred

- to hyperparameter tune <u>and</u> evaluate performance, we could do CV for the tuning, and test on a test set that wasn't used in tuning

- <u>or</u> for more robustness we could use nested CV (i.e., CV for hyperparam tuning, and for ~~the~~ generalization error)

For $l = 1$ to L:

$\mathcal{D}' = \mathcal{D} - \{l\}$

for M in $\mathcal{M}$:

for $k = 1$ to K:

$\mathcal{D}'' = \mathcal{D}' - \{k\}$

- train model w/ $\mathcal{D}''$

$\hat{\varepsilon}_M^{(l,k)} = MSE_M^{(l,k)}$

$\hat{\varepsilon}_M^{(l)} = \frac{1}{K} \sum_{k=1}^{K} \hat{\varepsilon}_M^{(l,k)}$

$\hat{\varepsilon}^{(l)} = \min_M (\hat{\varepsilon}_M^{(l)})$

$\hat{\varepsilon} = \frac{1}{L} \sum_{l=1}^{L} \hat{\varepsilon}^{(l)}$

- outer loop tests error
  - inner loop does CV hyperparam tuning.

- After this, we use normal CV hyperparam tuning to pick tuned params.

- then we train on all data and deploy

Bias / Variance for Supervised learning (i.e., for function estimation)

- Assuming in the real world that Y is generated from a deterministic function by adding noise: $Y = f(X) + \varepsilon$, it is not difficult to show that:

$$\underbrace{E_{\mathcal{D}\sim p^m,\, \varepsilon\sim E}\left[\left(Y - \hat{f}_{\mathcal{D}}(X)\right)^2 \,\middle|\, X=x\right]}_{\text{MSE}} = \text{Bias}^2\left(\hat{f}_{\mathcal{D}}(X)\right) + \text{Var}\left[\hat{f}_{\mathcal{D}}(X)\right] + \text{Var}[\varepsilon]$$

our estimator of Y: $\hat{f}_{\mathcal{D}}(X)$

w/ $\text{Bias}^2\left(\hat{f}_{\mathcal{D}}(X)\right) = \left(E\left[\hat{f}_{\mathcal{D}}(X)\right] - f(x)\right)^2$; $\text{Var}\left[\hat{f}_{\mathcal{D}}(x)\right] = E\left[\left(\hat{f}_{\mathcal{D}}(x) - E\left[\hat{f}_{\mathcal{D}}(x)\right]\right)^2\right]$