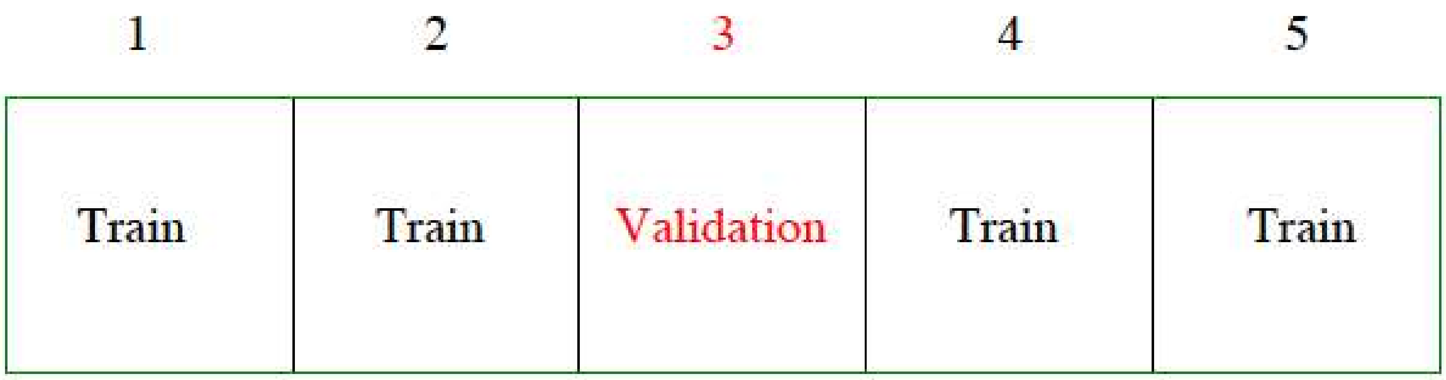
Reminder: cross-validation

Given training data (x_i, y_i), $i = 1, \ldots n$, we construct an estimator $\hat{f}$ of some unknown function f. Suppose that $\hat{f} = \hat{f}_\theta$ depends on a tuning parameter θ

How to choose a value of θ to optimize predictive accuracy of $\hat{f}_\theta$? Cross-validation offers one way. Basic idea is simple: divide up training data into K folds (here K is fixed, e.g., $K = 5$ or $K = 10$)

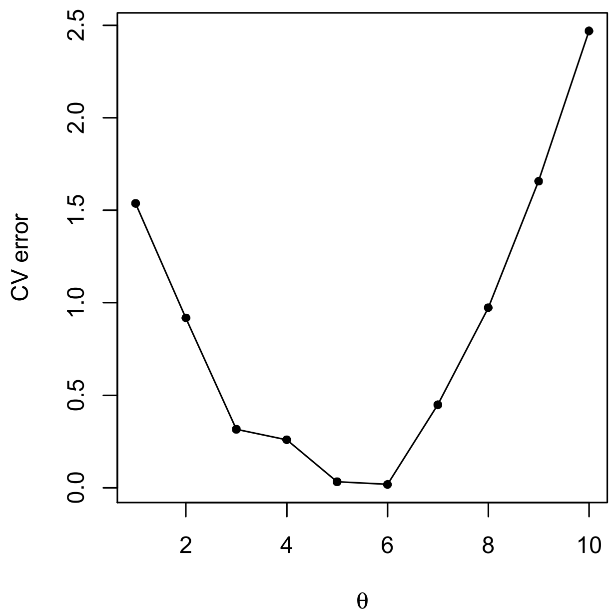


(Typically this is done at random)

Then, we hold out each fold one at a time, train on the remaining data, and predict the held out observations, for each value of the tuning parameter

I.e., for each value of the tuning parameter θ, the cross-validation error is

$$\mathrm{CV}(\theta) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} \big(y_i - \hat{f}_\theta^{-k}(x_i)\big)^2$$



We choose the value of tuning parameter that minimizes the CV error curve,

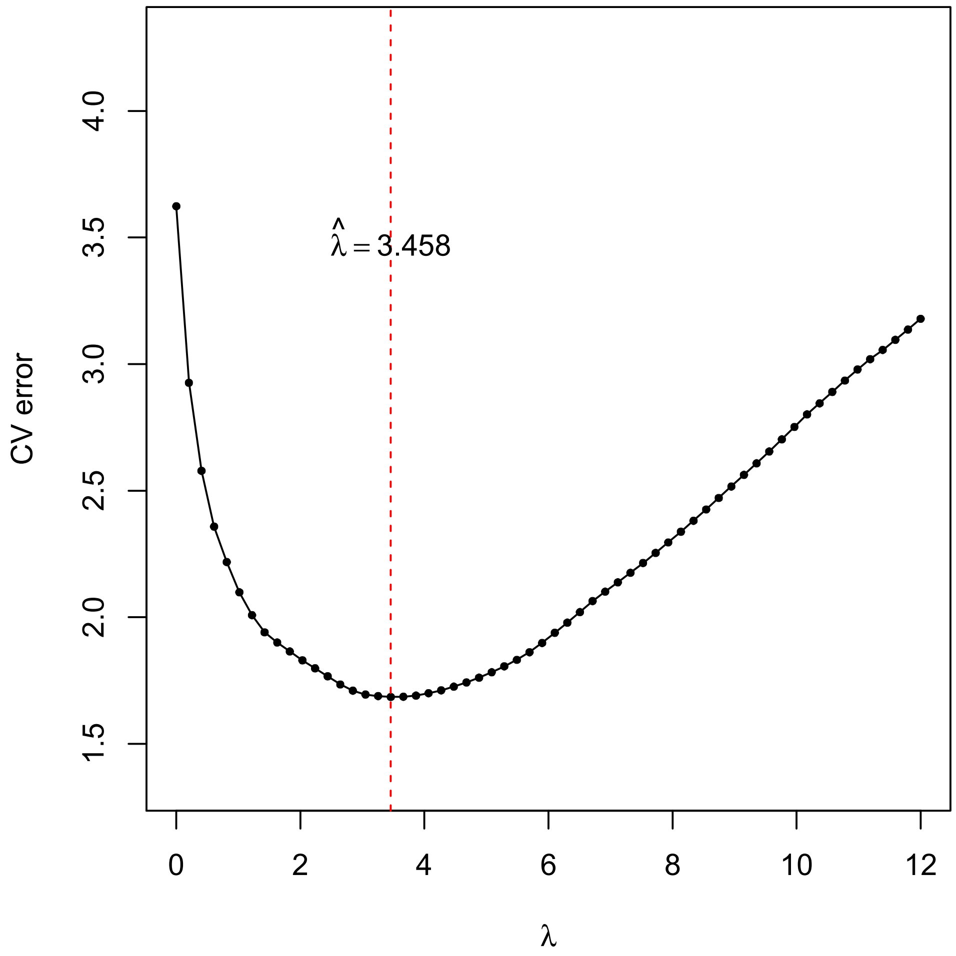
$$\hat{\theta} = \underset{\theta \in \{\theta_1, \ldots \theta_m\}}{\operatorname{argmin}} \; \mathrm{CV}(\theta)$$

Example: choosing λ for the lasso

Example from last time: $n = 50$, $p = 30$, and the true model is linear with 10 nonzero coefficients. We consider the lasso estimate

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

Performing 5-fold cross-validation, over 60 values of the tuning parameter between 0 and 12, we choose $\hat{\lambda} = 3.458$



What to do next?

What do we do next, after having used cross-validation to choose a value of the tuning parameter $\hat{\theta}$?

It may be an obvious point, but worth being clear: we now fit our estimator to the entire training set (x_i, y_i), $i = 1, \ldots n$, using the tuning parameter value $\hat{\theta}$

E.g., in the last lasso example, we resolve the lasso problem

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \, \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

on all of the training data, with $\hat{\lambda} = 3.458$

We can then use this estimator $\hat{\beta}^{\text{lasso}}$ to make future predictions

Reminder: standard errors

Recall that we can compute standard errors for the CV error curve at each tuning parameter value θ. First define, for $k = 1, \dots K$:

$$\mathrm{CV}_k(\theta) = \frac{1}{n_k} \sum_{i \in F_k} \big(y_i - \hat{f}_\theta^{-k}(x_i)\big)^2$$

where n_k is the number of points in the kth fold

Then we compute the sample standard deviation of $\mathrm{CV}_1(\theta), \dots \mathrm{CV}_K(\theta)$,

$$\mathrm{SD}(\theta) = \sqrt{\mathrm{var}\big(\mathrm{CV}_1(\theta), \dots \mathrm{CV}_K(\theta)\big)}$$

Finally we use

$$\mathrm{SE}(\theta) = \mathrm{SD}(\theta)/\sqrt{K}$$

for the standard error of $\mathrm{CV}(\theta)$

Reminder: one standard error rule

Recall that the one standard error rule is an alternative way of choosing θ from the CV curve. We start with the usual estimate

$$\hat{\theta} = \underset{\theta \in \{\theta_1, \ldots \theta_m\}}{\operatorname{argmin}} \; \mathrm{CV}(\theta)$$

and we move θ in the direction of increasing regularization until it ceases to be true that

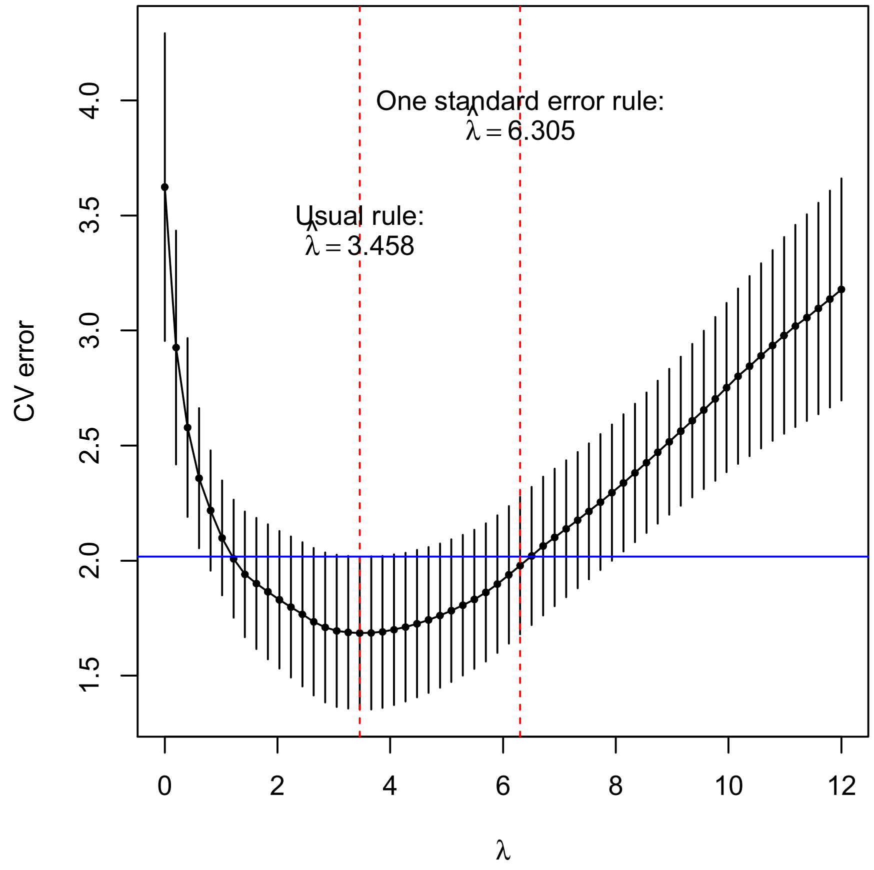
$$\mathrm{CV}(\theta) \leq \mathrm{CV}(\hat{\theta}) + \mathrm{SE}(\hat{\theta})$$

In words, we take the simplest (most regularized) model whose error is within one standard error of the minimal error

Example: choosing λ for the lasso

In the lasso criterion, larger λ means more regularization

For our last example, applying the one standard error rule has us increase the tuning parameter choice from $\hat{\lambda} = 3.458$ all the way up until $\hat{\lambda} = 6.305$



When fitting on the whole training data, this is a difference between a model with 19 and 16 nonzero coefficients

(Remember the true model had 10)