k-Fold Cross-Validation

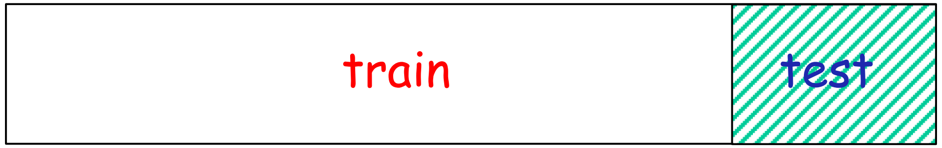
Maria-Florina Balcan

Tuning Hyperparameters

Suppose you want to determine a good value for some hyperparameter (like number of nodes in a decision tree or the right level of complexity in a hierarchy or regularization parameter for SVM)

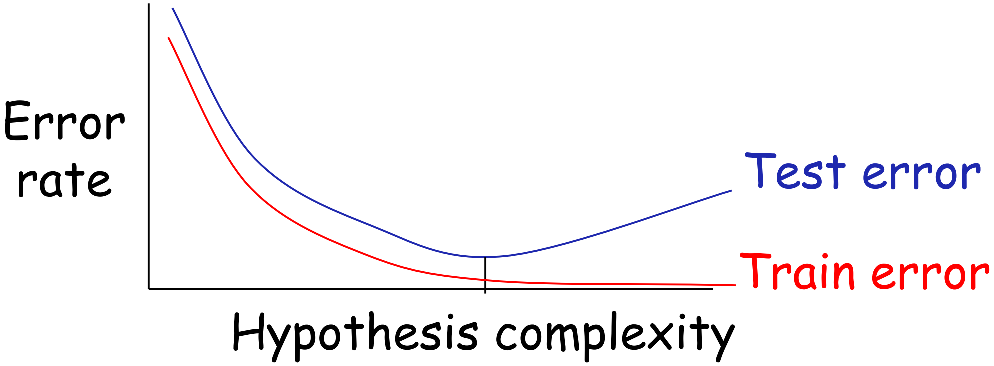
One good approach: use a holdout set (the train and test method)

- Partition available data into train and test set



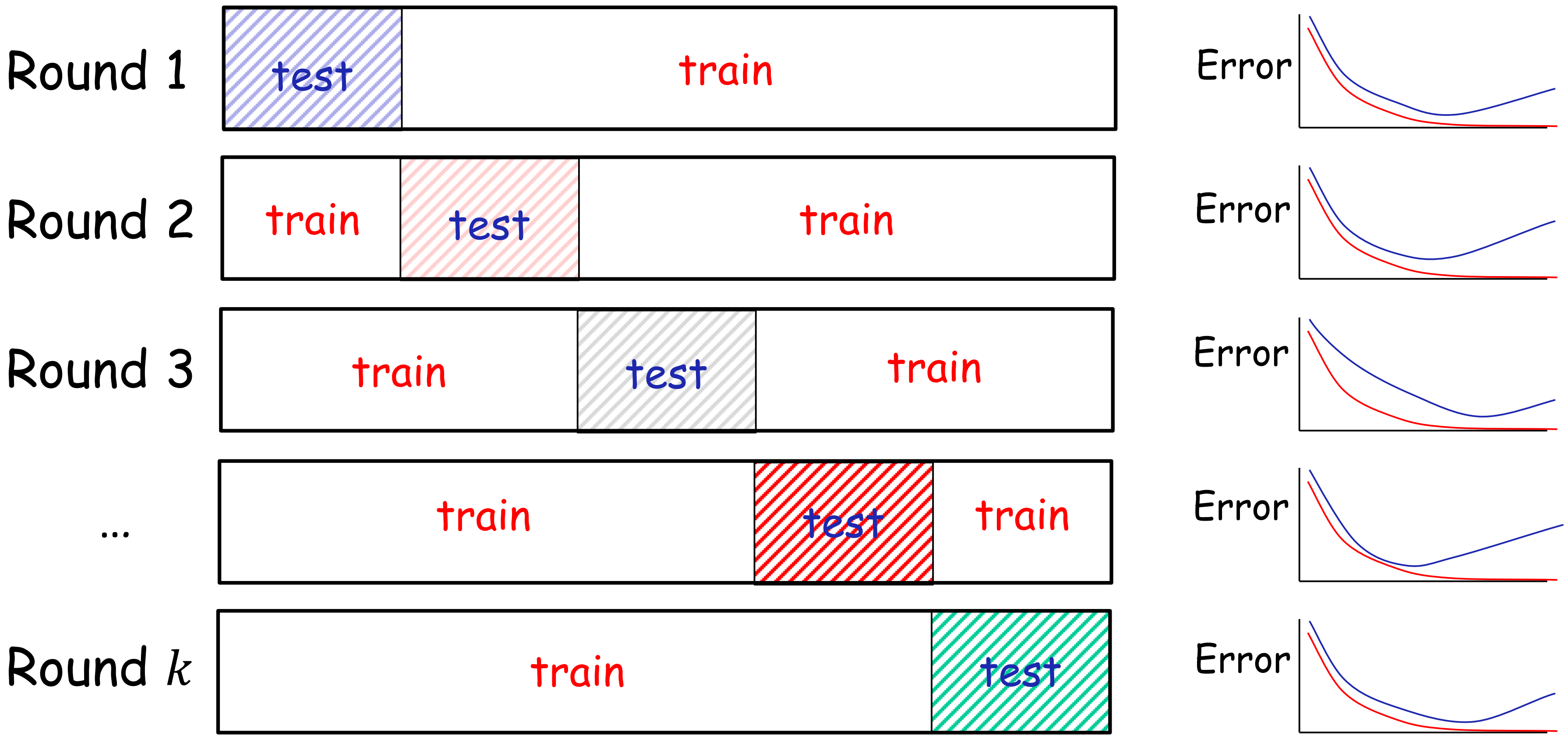
- For each hyperparameter value i, run learning algorithm on training set and evaluate on test set.

- Plot and take best

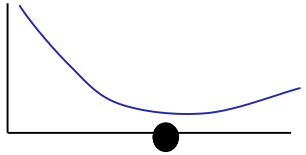


k-Fold Cross-Validation

Idea: partition data into k equal-size pieces. Repeat the holdout process k times, where in round j, use piece j as the test set.



Now average the test curves and take the best value



Intuition

For a given hyperparameter value i, our estimate of its quality is the average of k experiments

Averaging k independent copies of a random variable has lower variance than a single copy.

If the hypotheses produced in the different experiments were always the same, these would be k independent estimates of its error.

Of course, they might not be. So this is just intuition not a theorem.

Note: in practice, once best i is determined, then train on entire set.

Theoretical Guarantees

[BKL'99]: k-fold CV, for $2 < k < n$ is always (a little) better than a single holdout (except in degenerate cases where both are perfect).

- Suppose you use k-fold CV to produce k hypotheses $h_1, \ldots, h_k$, with true errors $err(h_1), \ldots, err(h_k)$, and error estimates $\widehat{err}(h_1), \ldots, \widehat{err}(h_k)$.

- Define h to be the function that randomly chooses among $h_1, \ldots, h_k$. So, $err(h) = \frac{err(h_1)+\cdots+err(h_k)}{k}$ and our estimate is $\widehat{err}(h) = \frac{\widehat{err}(h_1)+\cdots+\widehat{err}(h_k)}{k}$.

- Then, for any power $p \geq 2$,

$$E[|\widehat{err}(h) - err(h)|^p] < E[|\widehat{err}(h_1) - err(h_1)|^p]$$

(so long as RHS $\neq 0$).

[KKV'11, KLVV'13]: significant variance reduction for learning algorithms that satisfy stability properties.