Evaluation of Learning Models

Lecture by Shangsong Liang
Sun Yat-sen University
Thanks to Evgueni Smirnov

How to check if a model fit is good?

- The R^2 statistic has become the almost universally standard measure for model fit in linear models.
- What is R^2?

$$R^2 = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \bar{y})^2}$$

← Model error

← Variance in the dependent variable

- It is the ratio of error in a model over the total variance in the dependent variable.
- Hence the lower the error, the higher the R2 value.

How to check if a model fit is good?

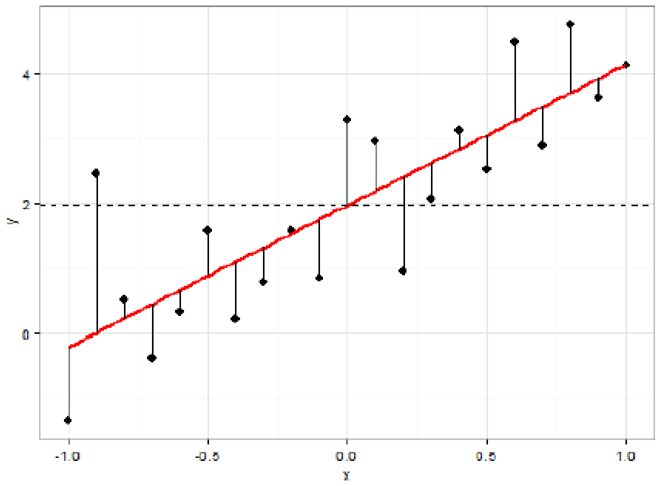
$\sum(y_i - f_i)^2$ = 18.568

$\sum(y_i - \bar{y})^2$ = 55.001

$R^2 = 1 - \frac{18.568}{55.001}$

$R^2 = 0.6624$

A decent model fit!



How to check if a model fit is good?

$\sum(y_i - f_i)^2 = 15.276$

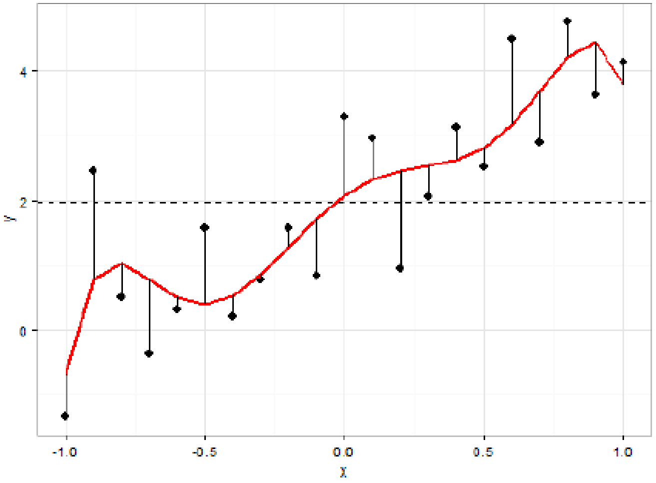
$\sum(y_i - \bar{y})^2 = 55.001$

$R^2 = 1 - \frac{15.276}{55.001}$

$R^2 = 0.72$

Is this a better model?

No, overfitting!



OVERFITTING

- Modeling techniques tend to overfit the data.
- Multiple regression:

✓ *Every* time you add a variable to the regression, the model's R^2 goes up.

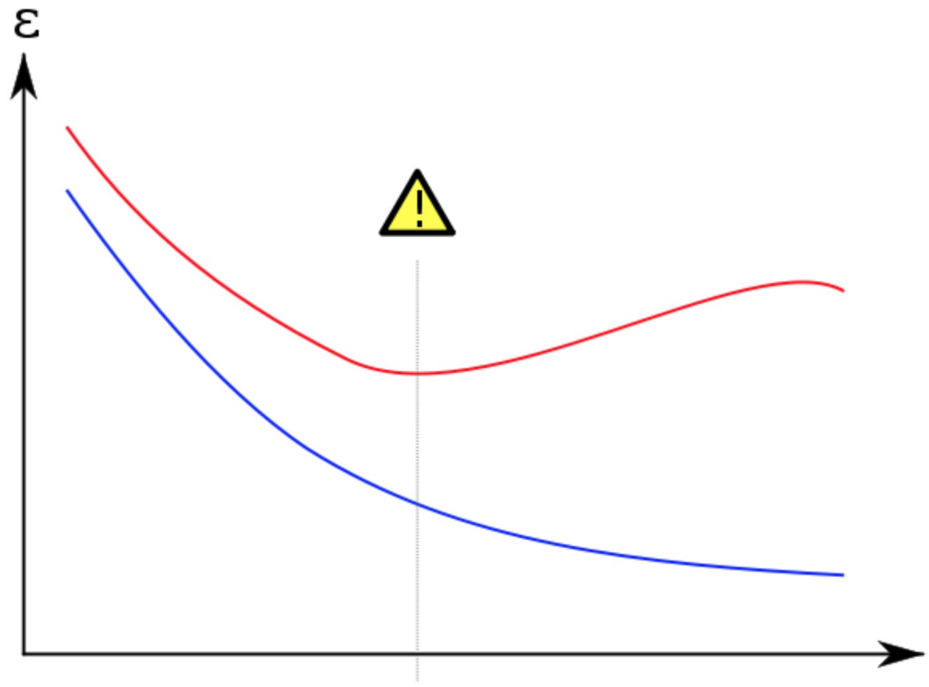
✓ Naïve interpretation: *every* additional predictive variable helps to explain yet more of the target's variance. But that can't be true!

✓ Left to its own devices, Multiple Regression will fit *too many* patterns.

✓ A reason why modeling requires subject-matter expertise.

OVERFITTING

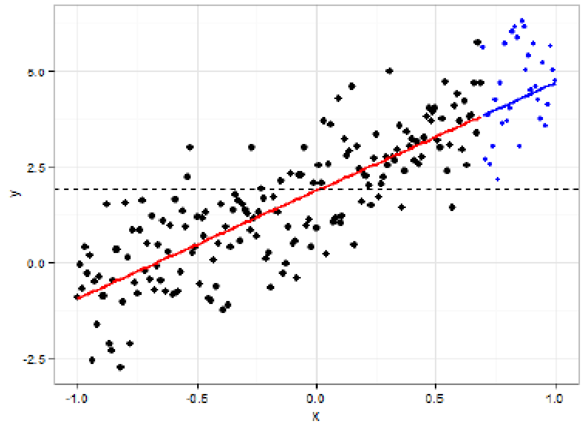
- Error on the dataset used to *fit* the model can be misleading
  › Doesn't predict future performance.
- Too much complexity can diminish model's accuracy on future data.
  › Sometimes called the Bias-Variance Tradeoff.

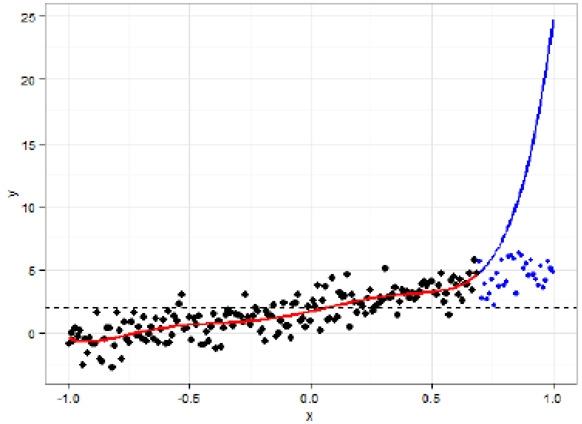


OVERFITTING

- What are the consequences of overfitting?

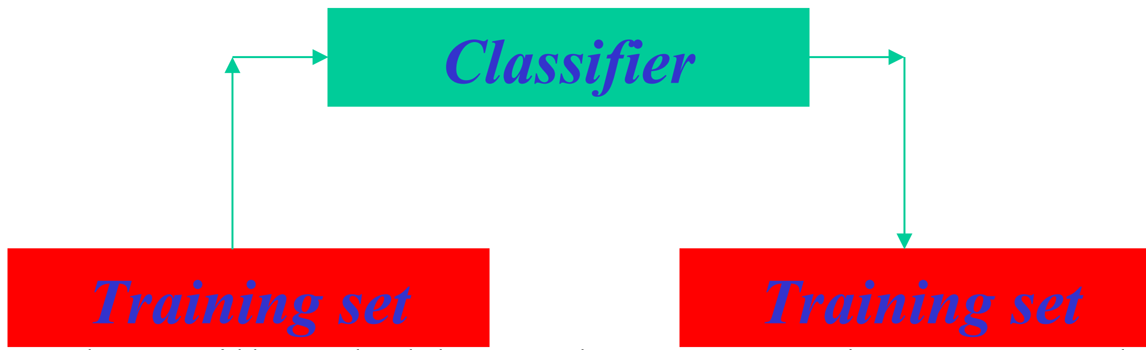
›"*Overfitted models will have high R^2 values, but will perform poorly in predicting out-of-sample cases*"





Estimation with Training Data

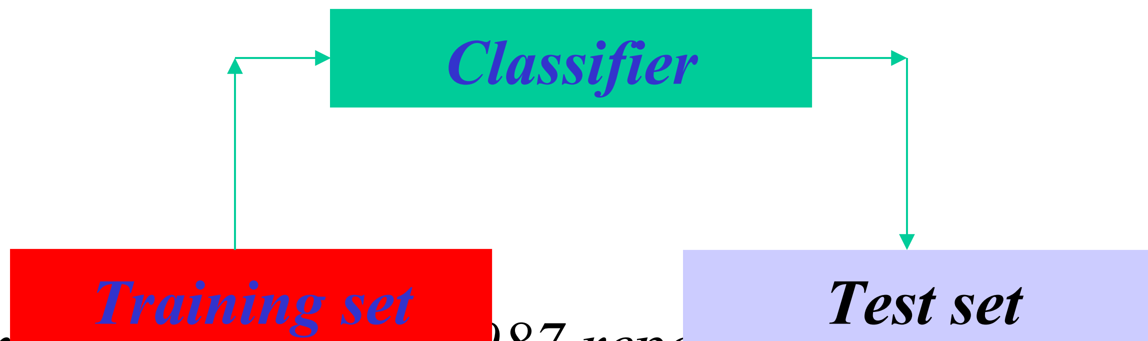
- The accuracy/error estimates on the training data are *not* good indicators of performance on future data.



  - **Q: Why?**
  - **A:** Because new data will probably not be **exactly** the same as the training data!
- The accuracy/error estimates on the training data measure the degree of classifier's overfitting.

Estimation with Independent Test Data

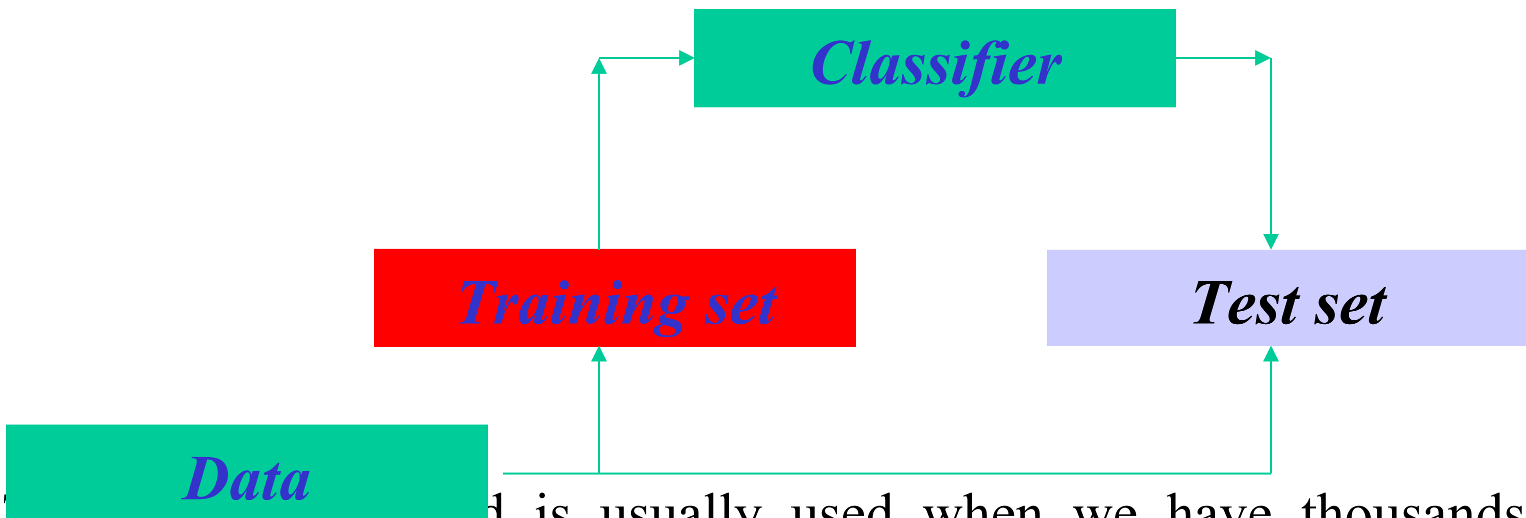
- Estimation with independent test data is used when we have plenty of data and there is a natural way to forming training and test data.



- *For example: Quinlan in 1987 reported experiments in a medical domain for which the classifiers were trained on data from 1985 and tested on data from 1986.*

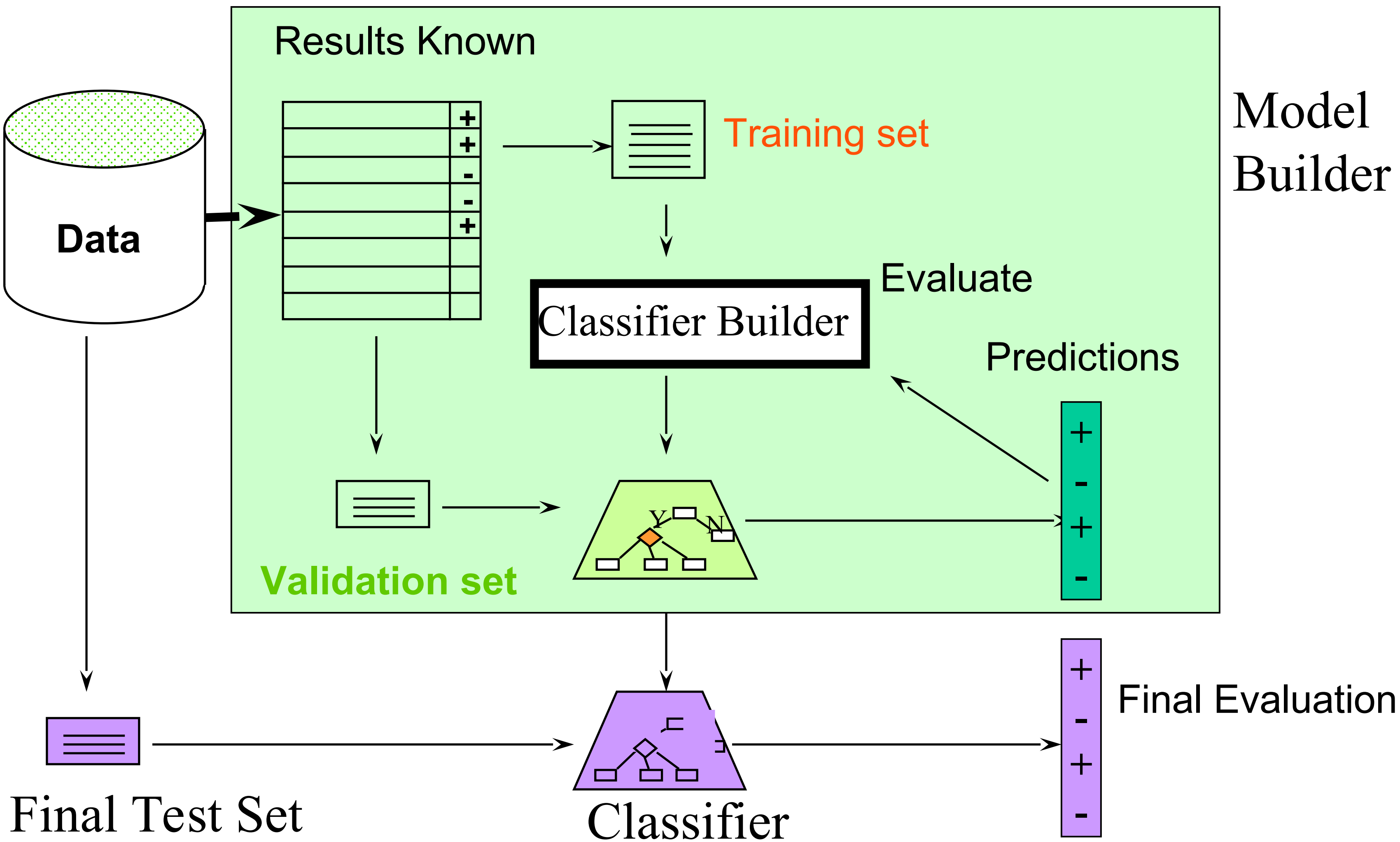
Hold-out Method (留出法)

- The hold-out method splits the data into training data and test data (usually 2/3 for train, 1/3 for test). Then we build a classifier using the train data and test it using the test data.



- The hold-out method is usually used when we have thousands of instances, including several hundred instances from each class.

Classification: Train, Validation, Test Split



The test data can't be used for parameter tuning!

Making the Most of the Data

- Once evaluation is complete, *all the data* can be used to build the final classifier.
- Generally, the larger the training data the better the classifier (but returns diminish).
- The larger the test data the more accurate the error estimate.

Stratification

- The *holdout* method (留出法) reserves a certain amount for testing and uses the remainder for training.
  - *Usually: one third for testing, the rest for training.*
- For "unbalanced" datasets, samples might not be representative.
  - *Few or none instances of some classes.*
- **Stratified (分层) sample: advanced version of balancing the data.**
  - *Make sure that each class is represented with approximately equal proportions in both subsets.*

Repeated Holdout Method

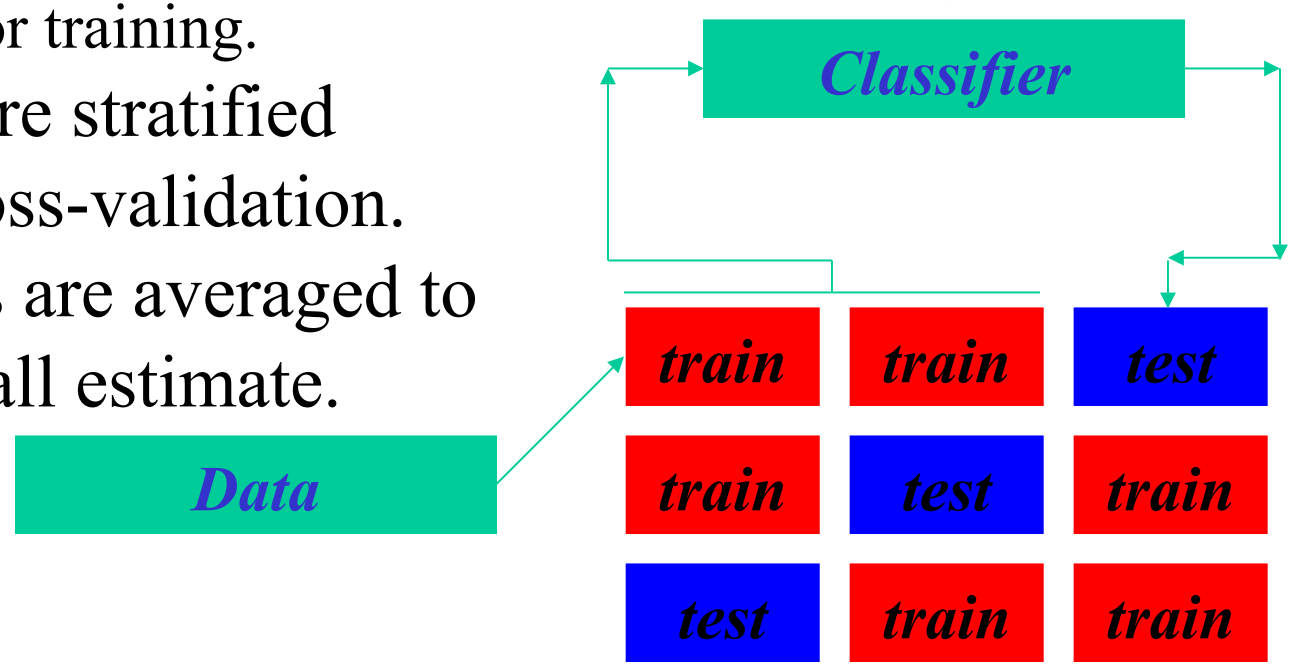
- Holdout estimate can be made more reliable by repeating the process with different subsamples.
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratification).
  - The error rates on the different iterations are averaged to yield an overall error rate.
- This is called the *repeated holdout* method.

Repeated Holdout Method

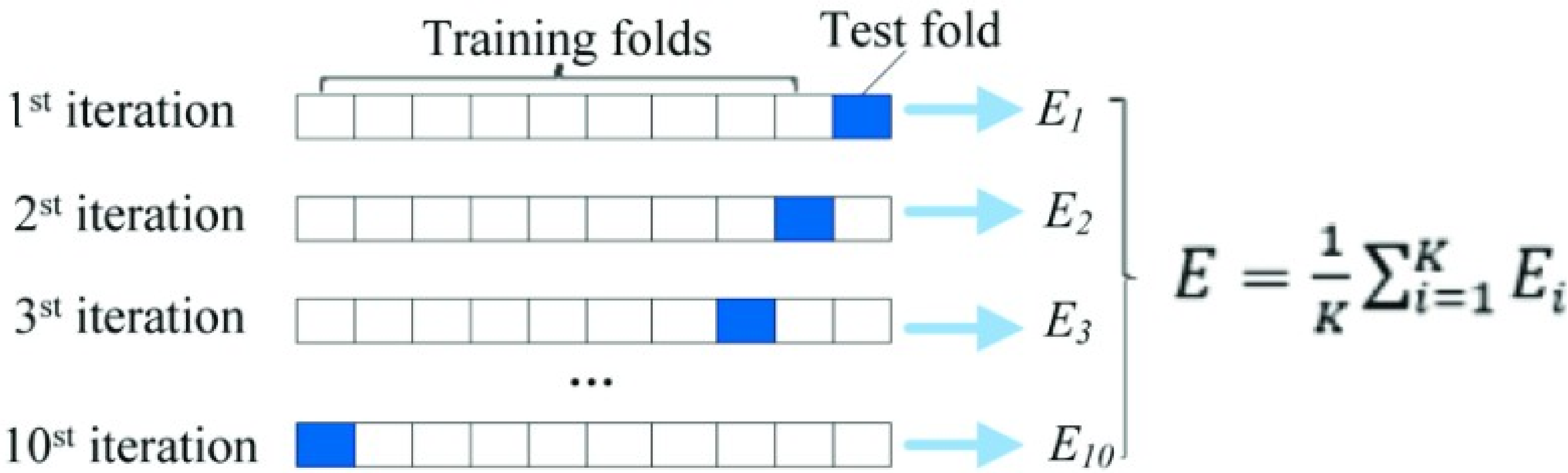
- Still not optimum: the different test sets overlap, but we would like all our instances from the data to be tested at least ones.
- Can we prevent overlapping?

k-Fold Cross-Validation

- *k-fold cross-validation* avoids overlapping test sets:
  - *First step*: data is split into *k* subsets of equal size;
  - *Second step*: each subset in turn is used for testing and the remainder for training.
- The subsets are stratified before the cross-validation.
- The estimates are averaged to yield an overall estimate.



k-Fold Cross-Validation

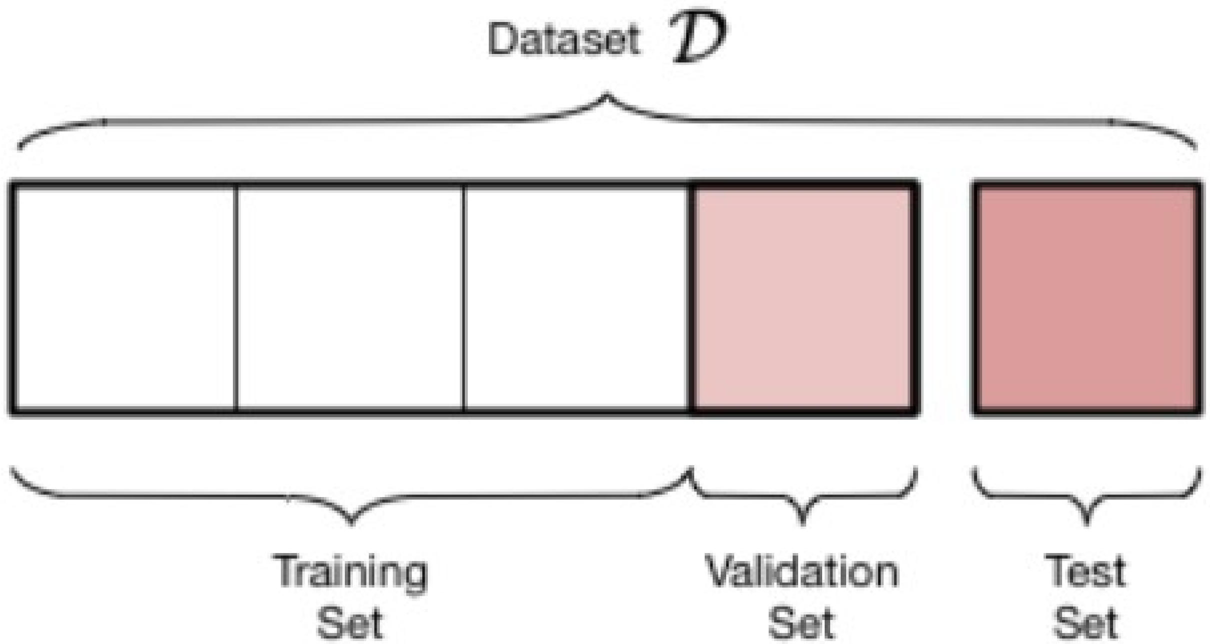


k-Fold Cross-Validation with Validation and Test Set

- A slightly less granular approach is to use a single k-fold cross validation with both a validation and test set.
  - The total data set is split in *k* sets.
  - One by one, a set is selected as test set.
  - Then, one by one, one of the remaining sets is used as a validation sets.
  - The other *k-2* sets are used as training sets until all possible combinations have been evaluated.

k-Fold Cross-Validation with Validation and Test Set

- A slightly less granular approach is to use a single k-fold cross validation with both a validation and test set.



More on Cross-Validation

- Standard method for evaluation: stratified 10-fold cross-validation.
- Why 10? Extensive experiments have shown that this is the best choice to get an accurate estimate.
- Stratification reduces the estimate's variance.
- Even better: repeated stratified cross-validation:
  - E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance).

Leave-One-Out Cross-Validation

- Leave-One-Out is a particular form of cross-validation:
  - Set number of folds to number of training instances;
  - I.e., for *n* training instances, build classifier *n* times.
- Makes best use of the data.
- Involves no random sub-sampling.
- Very computationally expensive.

Leave-One-Out Cross-Validation and Stratification

- A disadvantage of Leave-One-Out-CV is that stratification is not possible:
  - It *guarantees* a non-stratified sample because there is only one instance in the test set!
- Extreme example - random dataset split equally into two classes:
  - Best inducer predicts majority class;
  - 50% accuracy on fresh data;
  - Leave-One-Out-CV estimate is 100% error!

Bootstrap Method (自助法)

- Cross validation uses sampling *without replacement:*
  - The same instance, once selected, can not be selected again for a particular training/test set
- The *bootstrap* uses sampling *with replacement* to form the training set:
  - Sample a dataset of n instances n times *with replacement* to form a new dataset of n instances;
  - Use this data as the training set;
  - Use the instances from the original dataset that don’t occur in the new training set for testing.

Bootstrap Method

- The bootstrap method is also called the *0.632 bootstrap:*
  - A particular instance has a probability of 1–1/*n* of *not* being picked;
  - Thus its probability of ending up in the test data is:

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} = 0.368$$

  - This means the training data will contain approximately 63.2% of the instances and the test data will contain approximately 36.8% of the instances.

Estimating Error with the Bootstrap Method

- The error estimate on the test data will be very pessimistic because the classifier is trained on just ~63% of the instances.
- Therefore, combine it with the training error:

$$err = 0.632 \cdot e_{\text{test instances}} + 0.368 \cdot e_{\text{training instances}}$$

- The training error gets less weight than the error on the test data.
- Repeat process several times with different replacement samples; average the results.

Metric Evaluation Summary:

- Use test sets and the hold-out method for “large” data;
- Use the cross-validation method for “middle-sized” data;
- Use the leave-one-out and bootstrap methods for small data;
- Don’t use test data for parameter tuning - use separate validation data.