Model selection

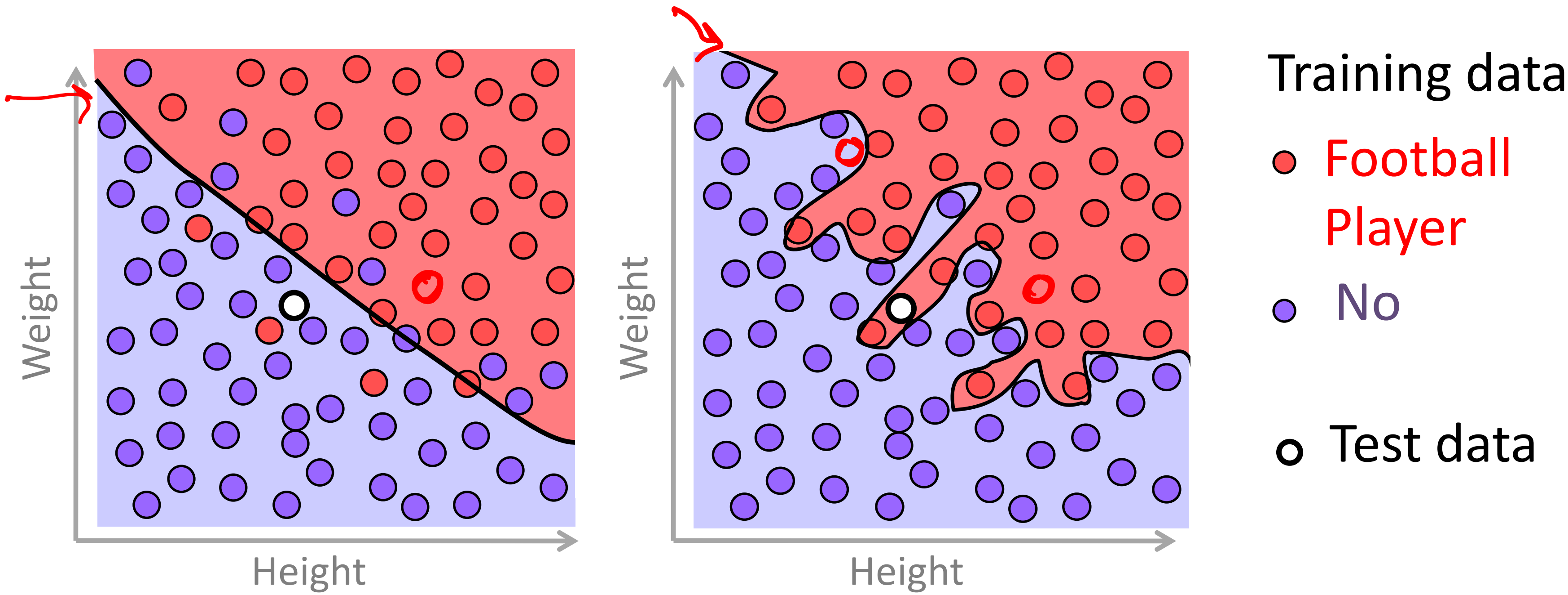
Aarti Singh

Machine Learning 10-315
Nov 11 , 2020

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science

Judging Test error

Training Data vs. Test Data



- A good machine learning algorithm
  - Does not **overfit** training data
  - **Generalizes** well to test data

Training error

- Training error of a classifier f

$$\frac{1}{n}\sum_{i=1}^{n} 1_{f(X_i)\neq Y_i}$$

loss(f(x_i), y_i)

Training Data

$$\{X_i, Y_i\}_{i=1}^n$$

- What about test error?

  Can't compute it.

$E[\text{loss}(f(x), y)] = P(f(x) \neq y)$

$\{0, 1\}$

- How can we know classifier is not overfitting?

  Hold-out or Cross-validation

Hold-out method

Can judge test error by using an independent sample of data.

Hold - out procedure:

n data points available $D \equiv \{X_i, Y_i\}_{i=1}^n$

1) Randomly split into two sets (preserving label proportion):

Training dataset $D_T = \{X_i, Y_i\}_{i=1}^m$

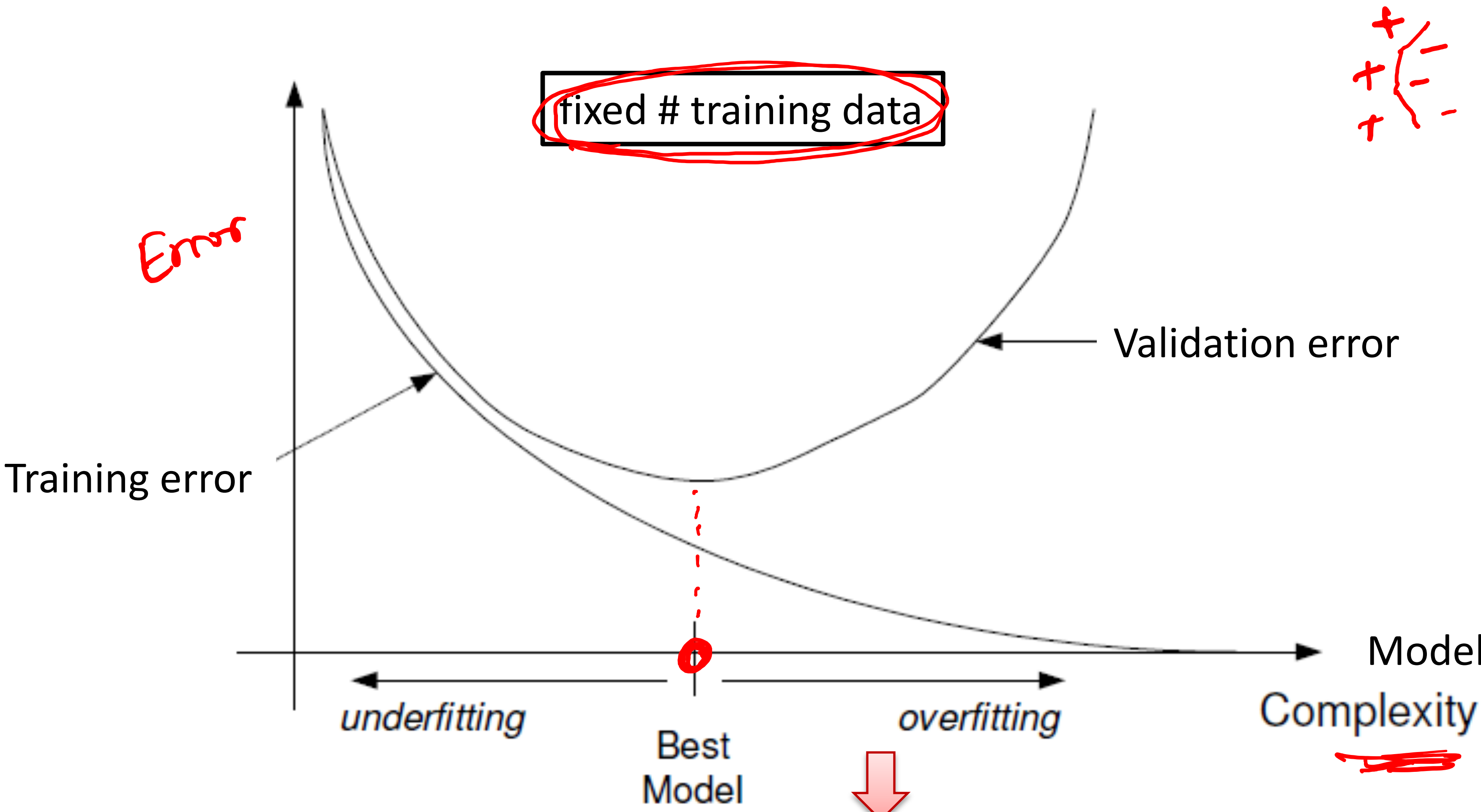
Validation/Hold-out dataset $D_V = \{X_i, Y_i\}_{i=m+1}^n$

often m = n/2

2) Train classifier on D_T. Report error on validation dataset D_V.

Overfitting if validation error is much larger than training error

Training vs. Validation Error



Training error is no longer a
good indicator of validation or test error

Hold-out method

Drawbacks:

- May not have enough data to afford setting one subset aside for getting a sense of generalization abilities
- Validation error may be misleading (bad estimate of test error) if we get an "unfortunate" split

Limitations of hold-out can be overcome by a family of sub-sampling methods at the expense of more computation.

Cross-validation

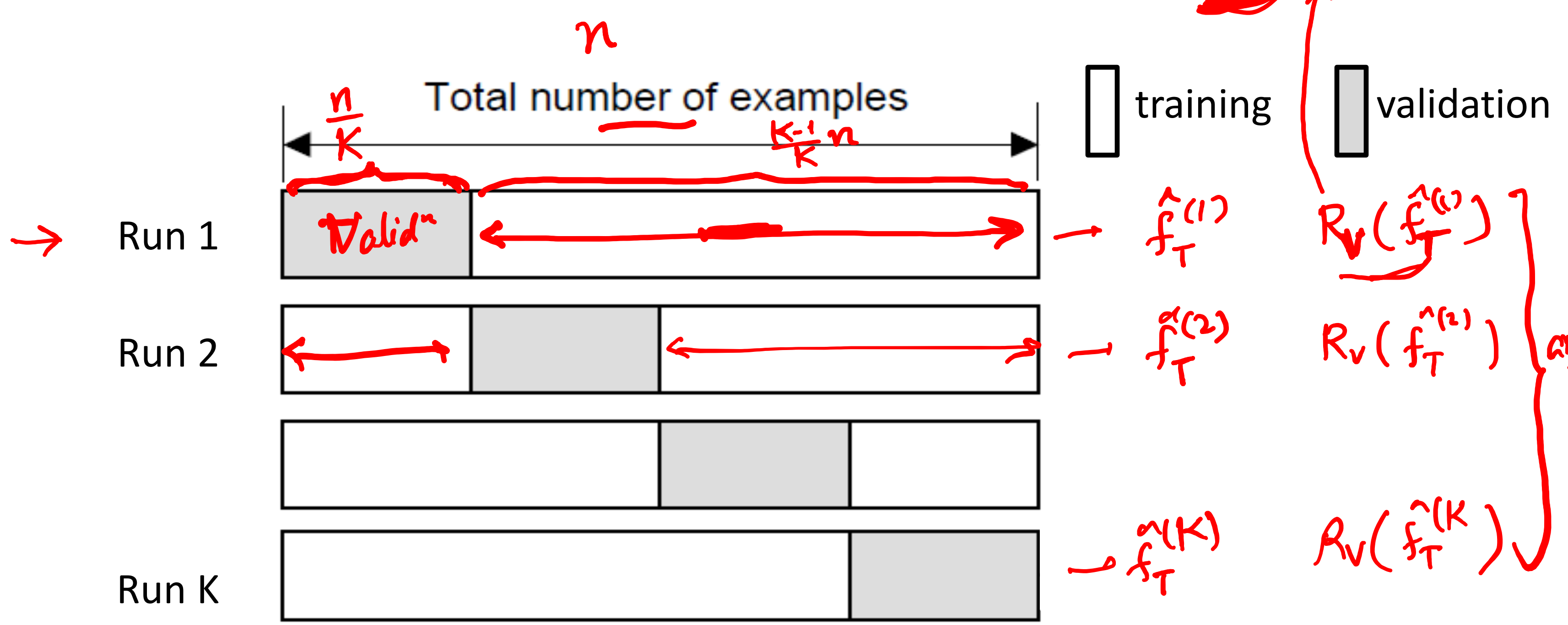
K-fold cross-validation

Create K-fold partition of the dataset.

Do K runs: train using K-1 partitions and calculate validation error on remaining partition (rotating validation partition on each run).

Report average validation error

$$\frac{1}{n_v}\sum_{i=1}^{n_v} 1_{(\hat{f}_T^{(1)}(X_i)\neq Y_i)}$$



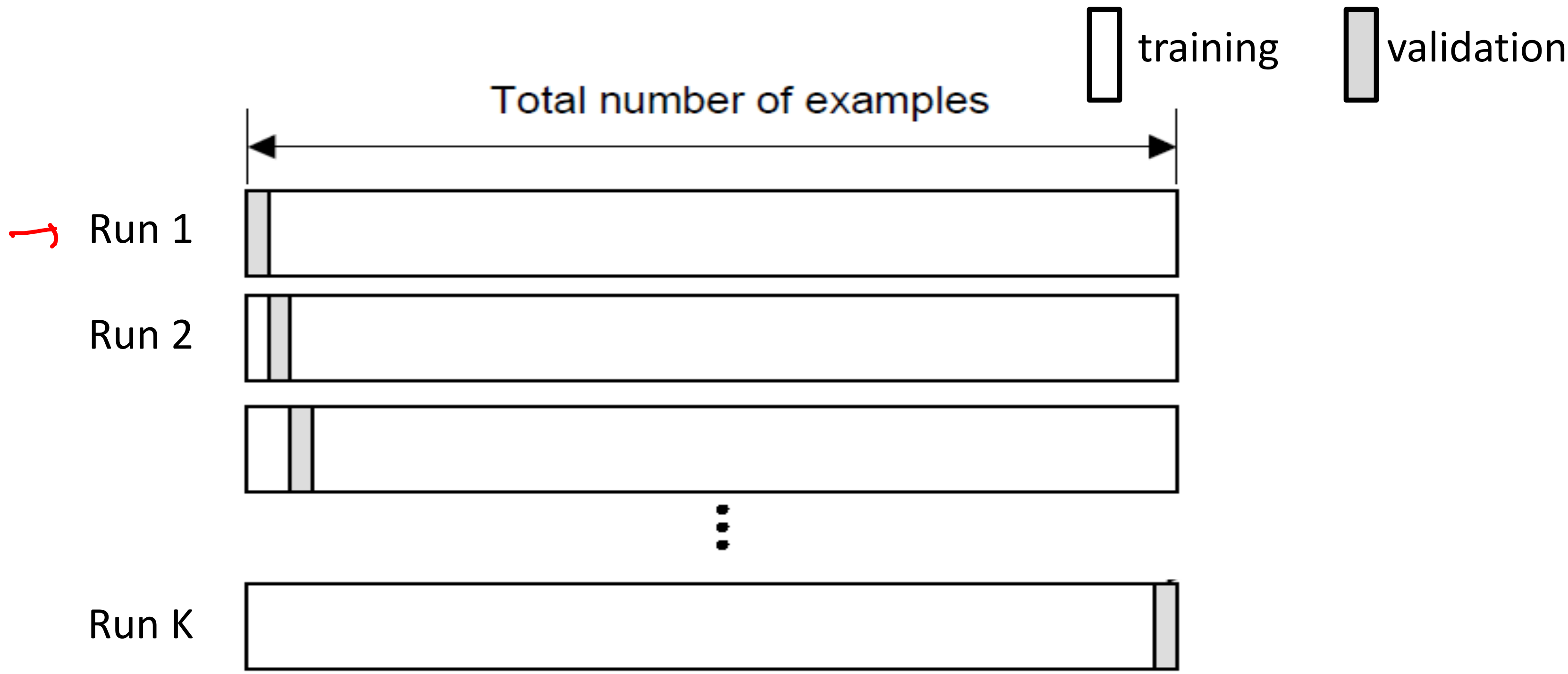
Cross-validation

Leave-one-out (LOO) cross-validation

size of dataset

Special case of K-fold with K=n partitions

Equivalently, train on n-1 samples and validate on only one sample per run for n runs



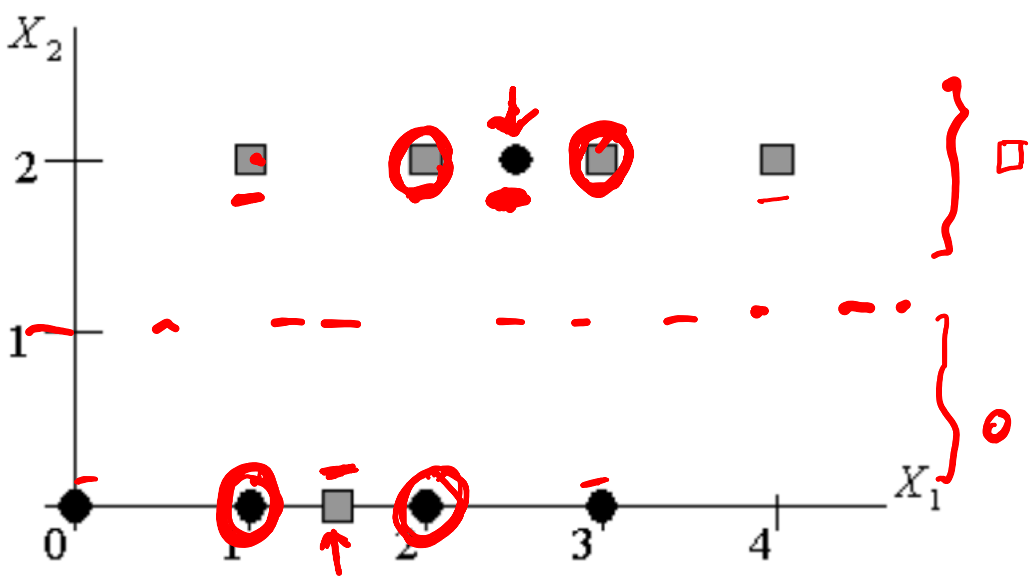
Cross-validation

What is the leave-one-out cross-validation error of the given classifiers on the following dataset?

No mistake
4 runs

Mistakes
4
+1
+1

x_i



- Poll 1: Decision stump using best feature X_2 2/10
- Poll 2: 1-NN classifier 6/10

Cross-validation

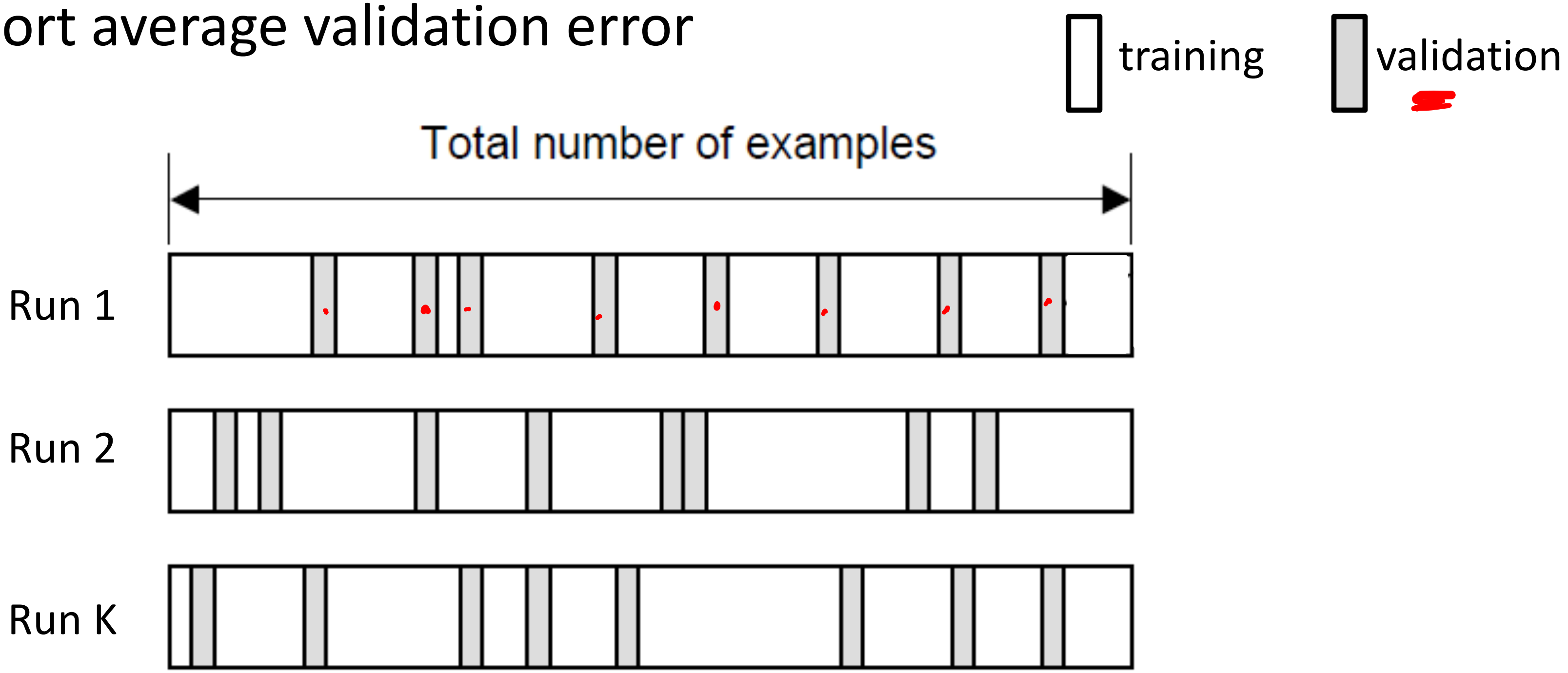
Random subsampling

Randomly subsample a fixed fraction αn (0< α <1) of the dataset for validation.

Compute validation error with remaining data as training data.

Repeat K times

Report average validation error



Practical Issues in Cross-validation

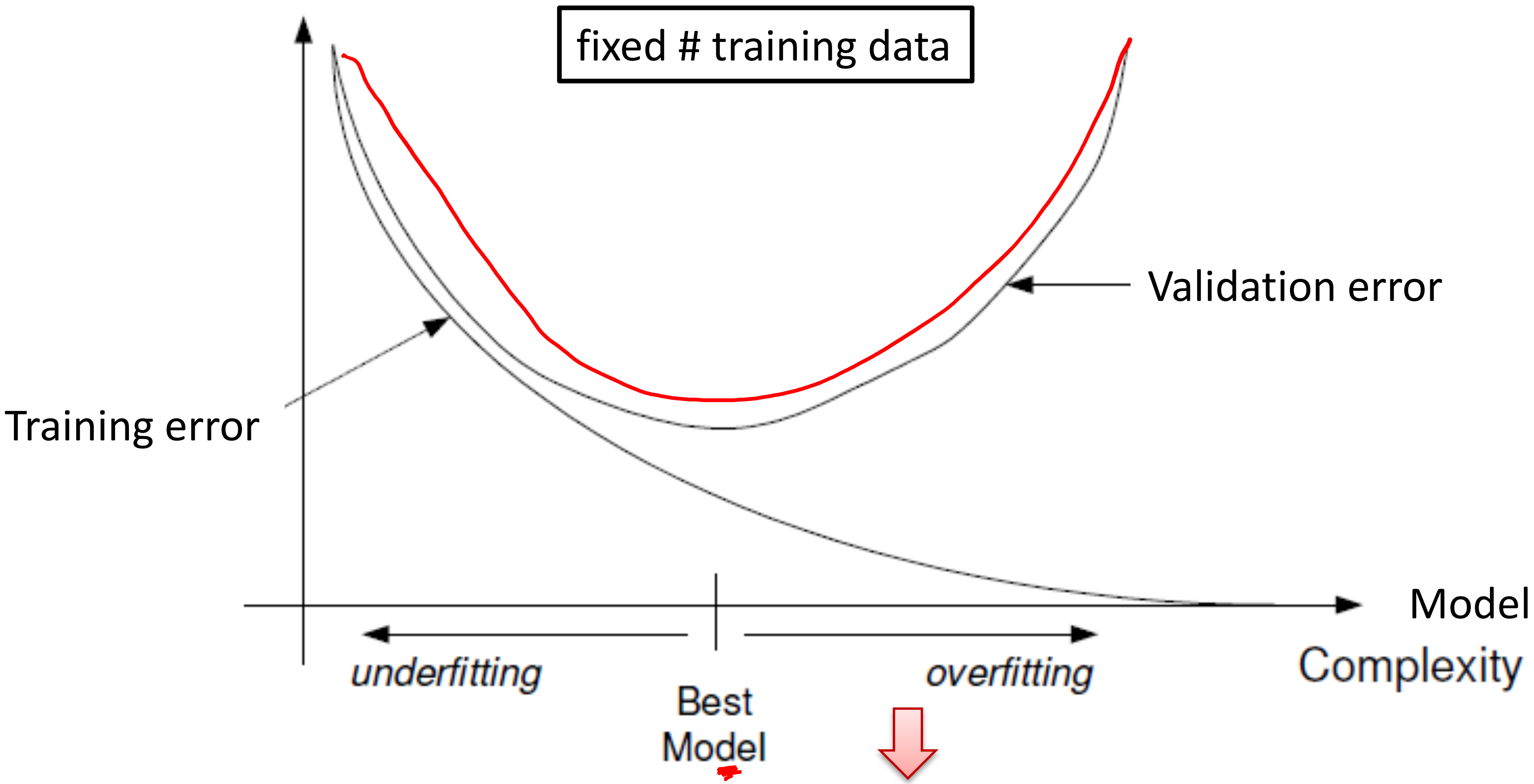
How to decide the values for K and α?

- Large K ⇒ smaller validation set, larger training set
  - + Validation error can approximate test error well
  - - Observed validation error will be unstable (few validation pts)
  - - The computational time will be very large as well (several runs)

- Small K ⇒ larger validation set, fewer training pts.
  - + The #runs and, therefore, computation time are reduced
  - + Observed validation error will be stable (many validation pts)
  - - Validation error cannot approximate test error well

Common choice: K = 10, α = 0.1 ☺

Training vs. Validation Error



Training error is no longer a
good indicator of validation or test error

Bias-Variance Tradeoff

- Why does test/validation error go up with increasing model complexity?

Two sources of error:

e.g. Regression   Mean square error

$E[(f(x)-y)^2]$

Bias/Approximation

$|E[Err(f_n)] - Err(f^*)|$

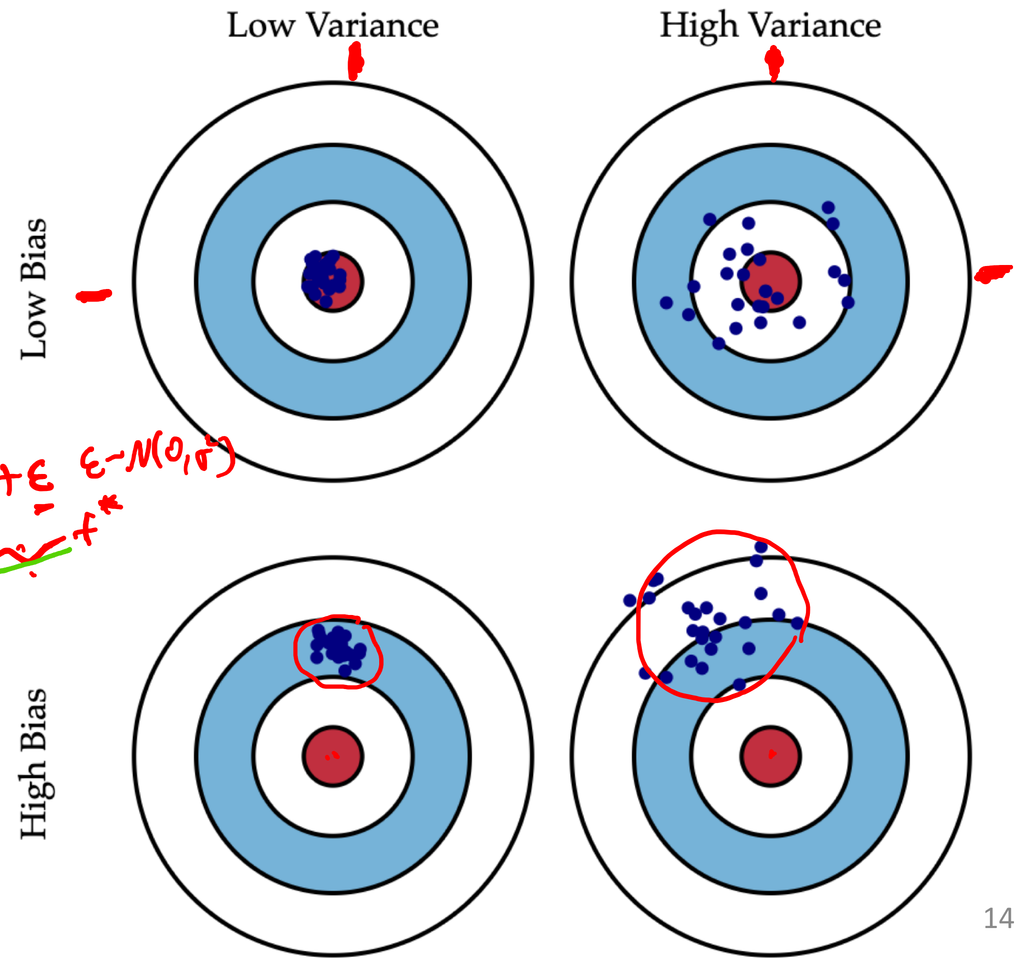
$y = f^*(x) + \epsilon \quad \epsilon \sim N(0,\sigma^2)$

f^*

f_n

Variance/Stability

$E[\,|Err(f_n) - E[Err(f_n)]|^2]$

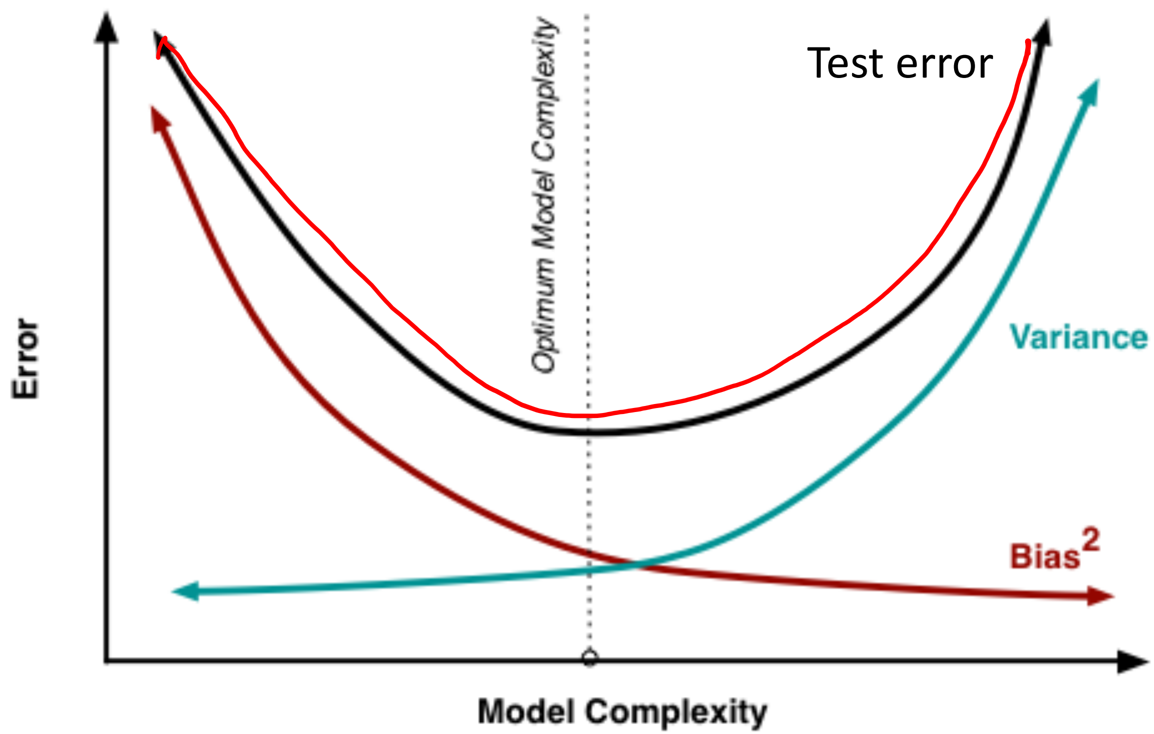


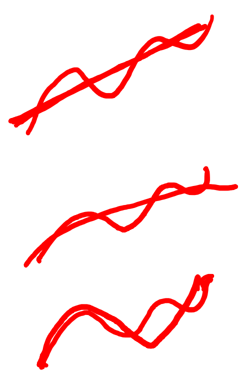
Bias-Variance Tradeoff

- Why does test/validation error go up with increasing model complexity?

σ^2

Regression test error = Variance + Bias^2 + Irreducible error



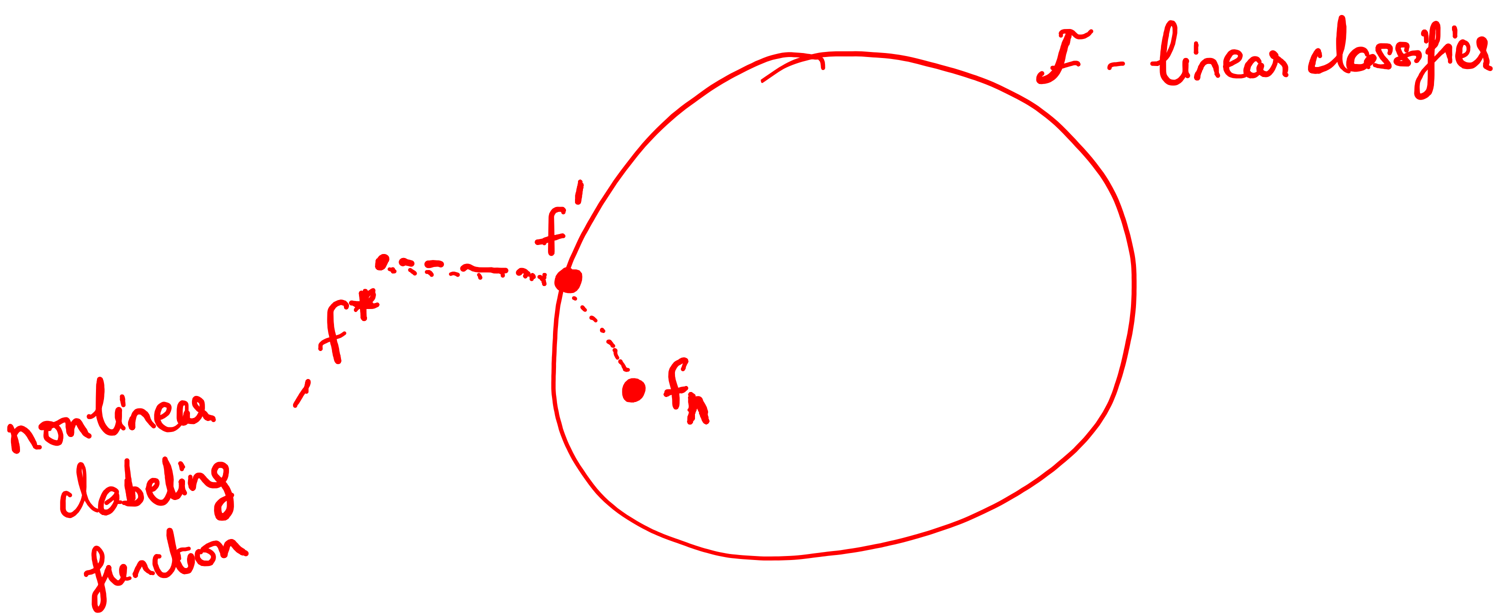


Bias-Variance Tradeoff

More generally: Let f' – best predictor in class $\mathcal{F}$

Test error = Estimation error + Approximation error + Irreducible error

$$\mathrm{Err}(f_n) = \mathrm{Err}(f_n) - \mathrm{Err}(f') + \mathrm{Err}(f') - \mathrm{Err}(f^*) + \mathrm{Err}(f^*)$$

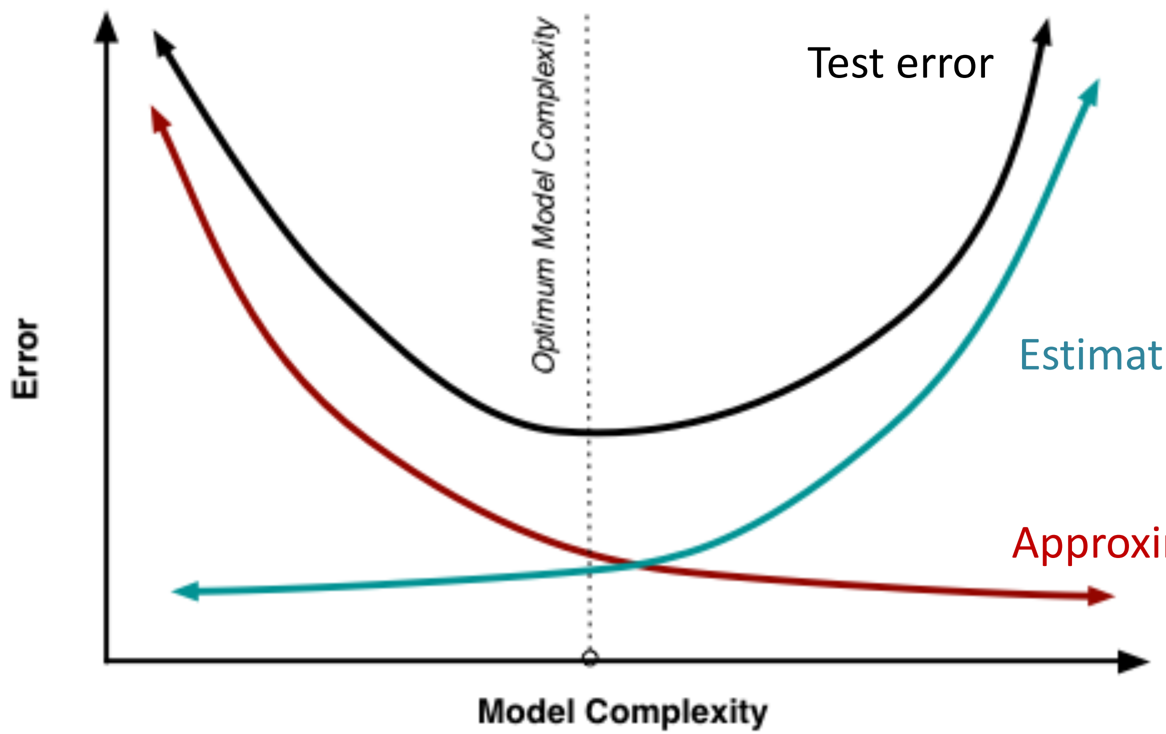


Bias-Variance Tradeoff

More generally: Let f' – best predictor in class $\mathcal{F}$

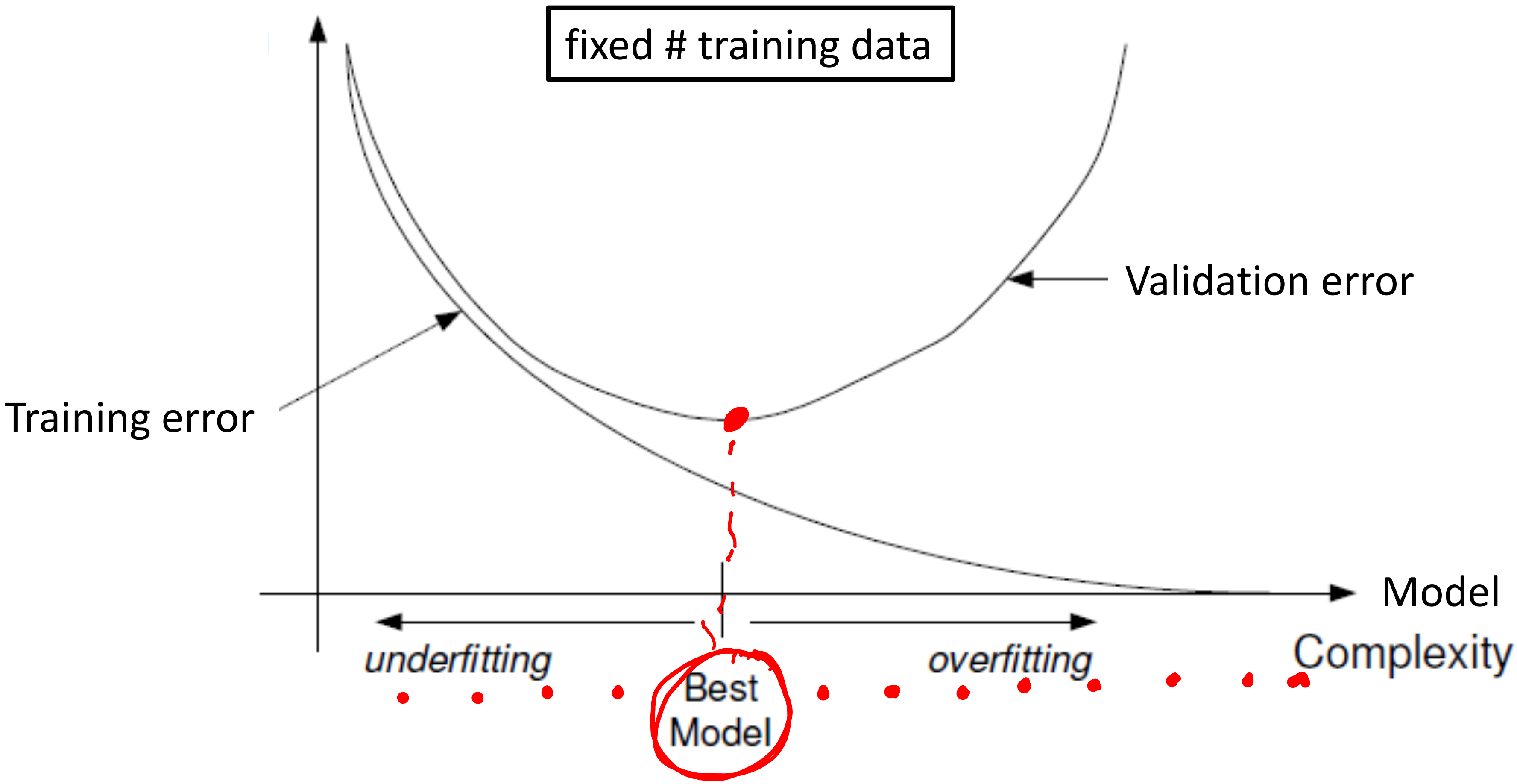
Test error = Estimation error + Approximation error + Irreducible error

$\mathrm{Err}(f_n) = \mathrm{Err}(f_n) - \mathrm{Err}(f') + \mathrm{Err}(f') - \mathrm{Err}(f^*) + \mathrm{Err}(f^*)$



Model selection

Effect of Model Complexity



Can we select good models using hold-out or cross-validation?

Examples of Model Spaces

Model Spaces with increasing complexity:

k

- Nearest-Neighbor classifiers with increasing neighborhood sizes k = 1,2,3,…

  Large neighborhood => lower complexity

- Decision Trees with increasing depth k or with k leaves

  Higher depth/ More # leaves => higher complexity

- Neural Networks with increasing layers or nodes per layer

  More layers/Nodes per layer => higher complexity

- MAP estimates with stronger priors (larger hyper-parameters β_H, β_T for Beta distribution or smaller variance for Gaussian prior)

  loss $+ \lambda$ pen $\rightarrow \|W\|$

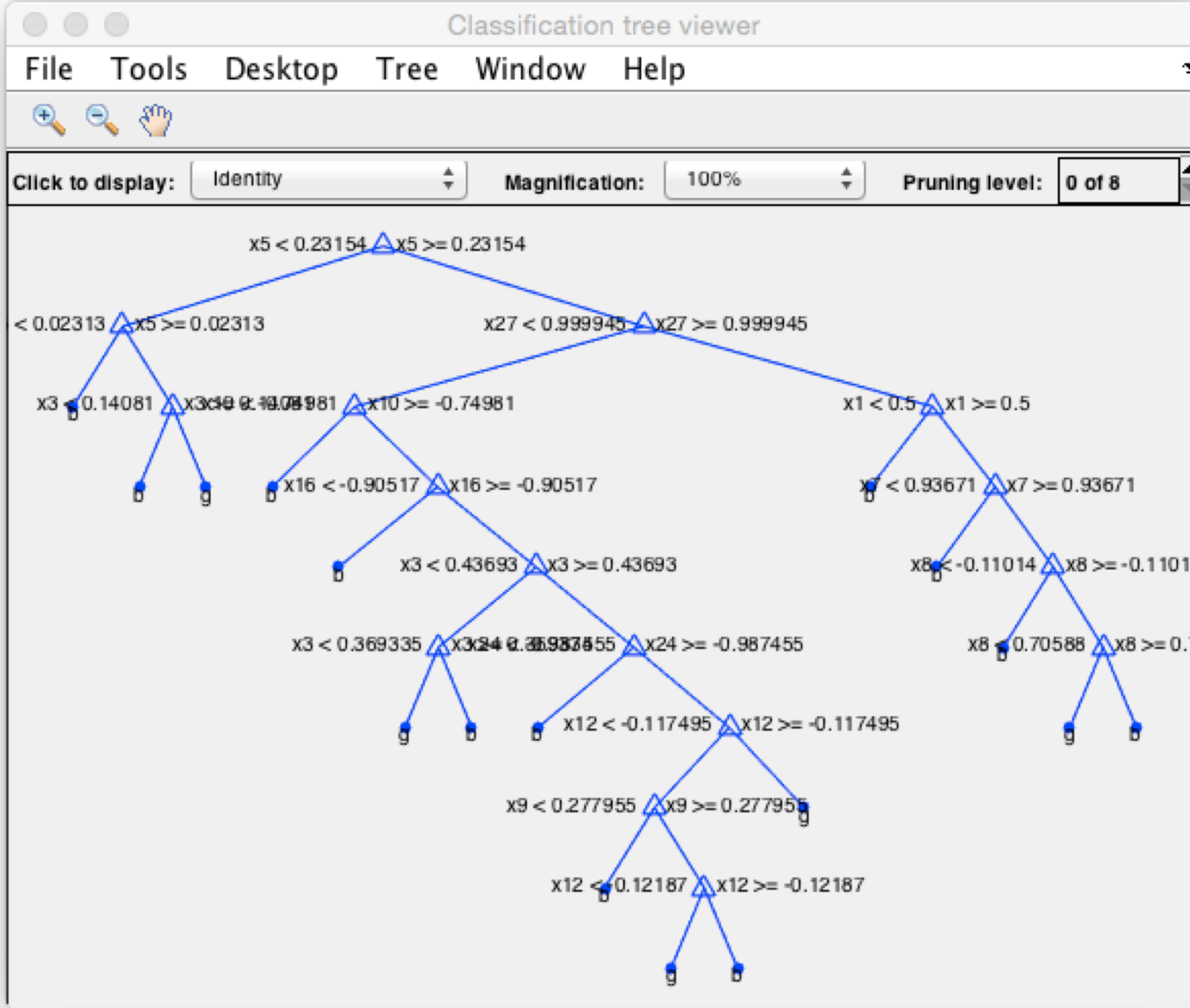
How can we select the right complexity model ?

Model selection using Hold-out/Cross-validation

- Train models of different complexities and evaluate their validation error using hold-out or cross-validation

- Pick model with smallest validation error (averaged over different runs for cross-validation)

Matlab example – decision tree

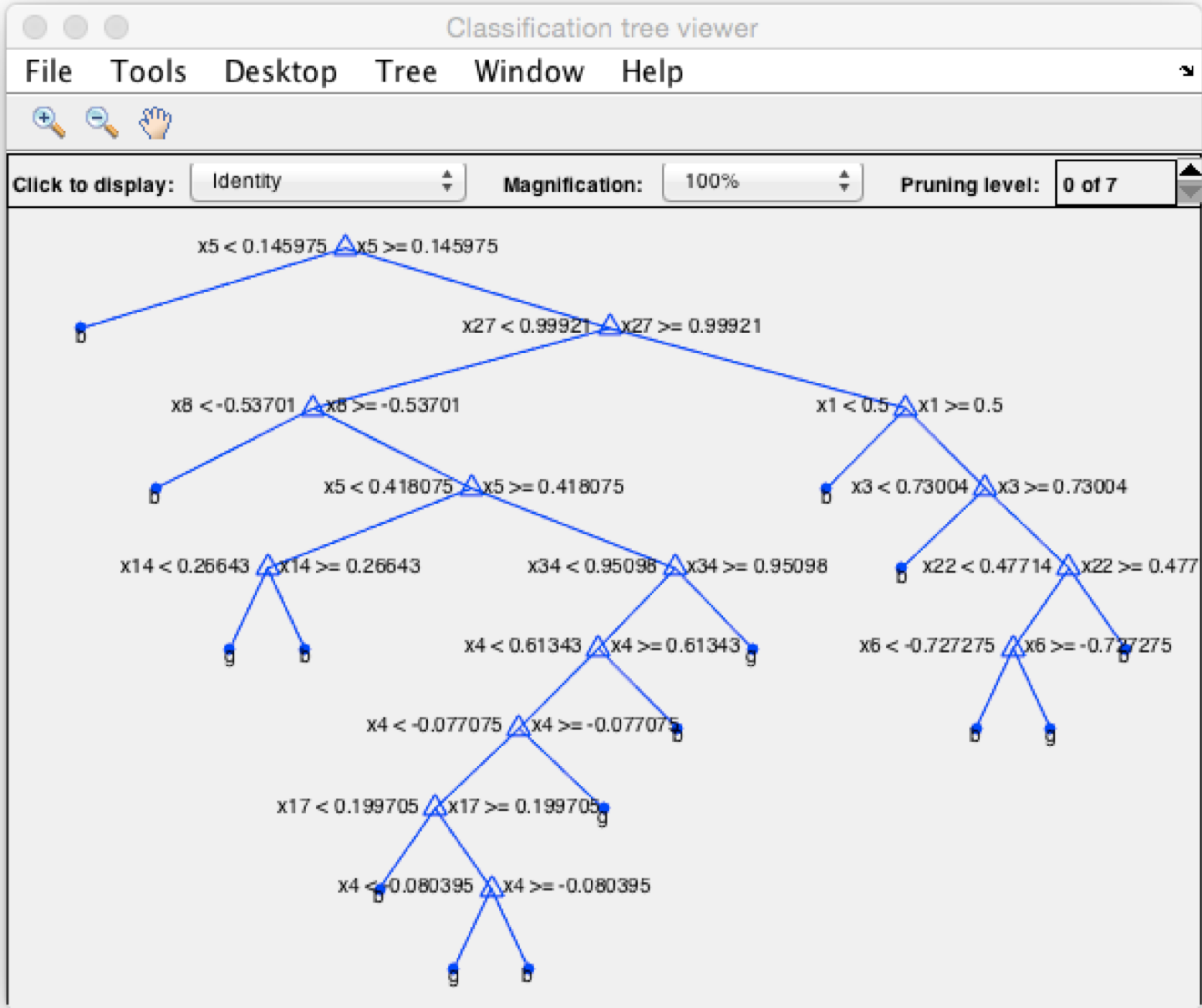



```
load ionosphere
% UCI dataset
% 34 features, 351 samples
% binary classification
rng(100)

%Defaulty MinLeafSize = 1
tc = fitctree(X,Y);
cvmodel = crossval(tc);
view(cvmodel.Trained{1},'Mode','graph')
kfoldLoss(cvmodel)
```

Validation error = 0.1254

Matlab example – decision tree



load ionosphere

% UCI dataset

% 34 features, 351 samples

% binary classification

rng(100)

%Defaulty MinLeafSize = 1

tc = fitctree(X,Y, 'MinLeafSize',2);

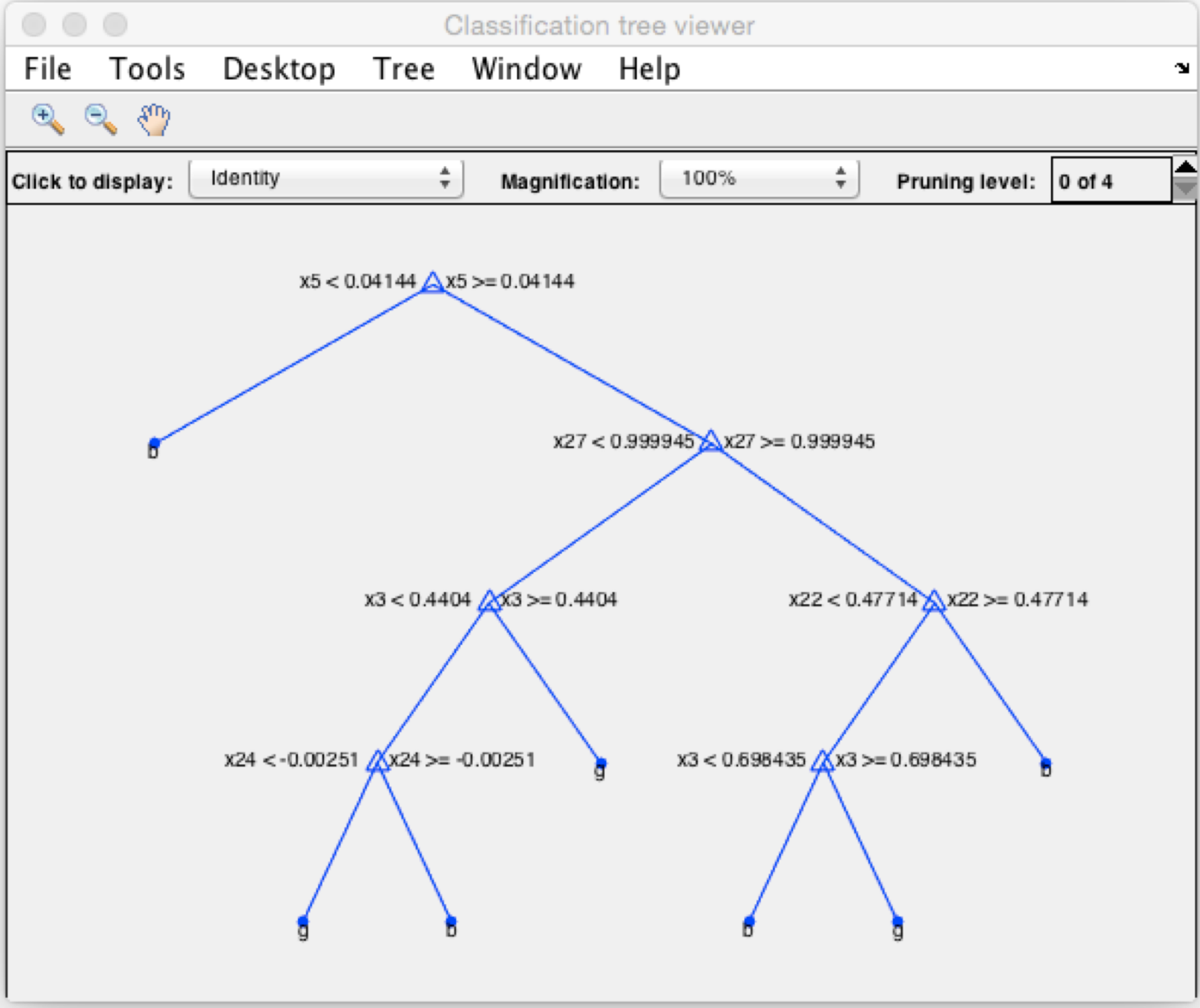
cvmodel = crossval(tc);

view(cvmodel.Trained{1},'Mode','graph')

kfoldLoss(cvmodel)

Validation error = 0.1168

Matlab example – decision tree

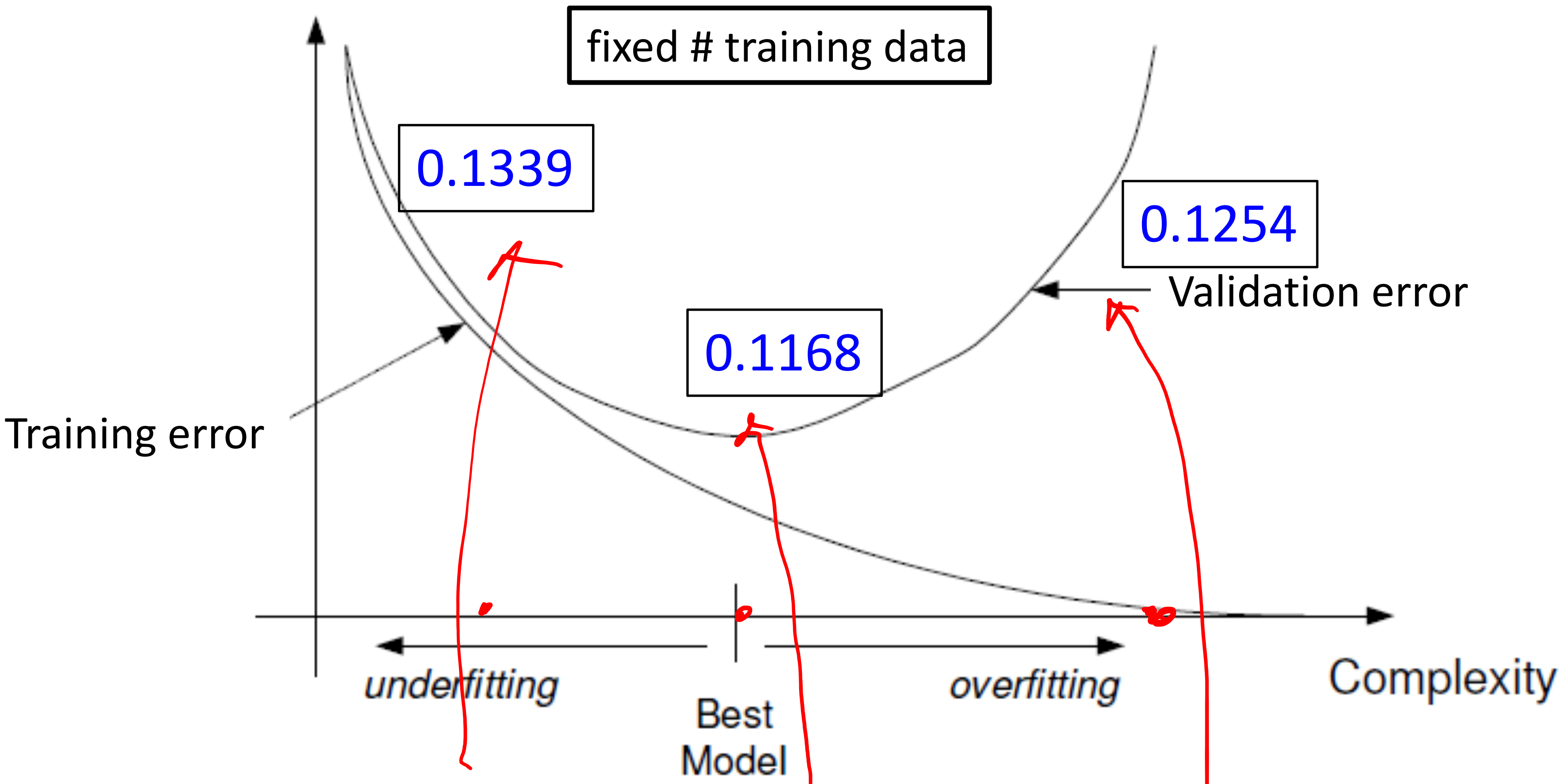


load ionosphere
% UCI dataset
% 34 features, 351 samples
% binary classification
rng(100)

%Defaulty MinLeafSize = 1
tc = fitctree(X,Y, 'MinLeafSize',10);
cvmodel = crossval(tc);
view(cvmodel.Trained{1},'Mode','graph')
kfoldLoss(cvmodel)

Validation error = 0.1339

Matlab example – decision trees

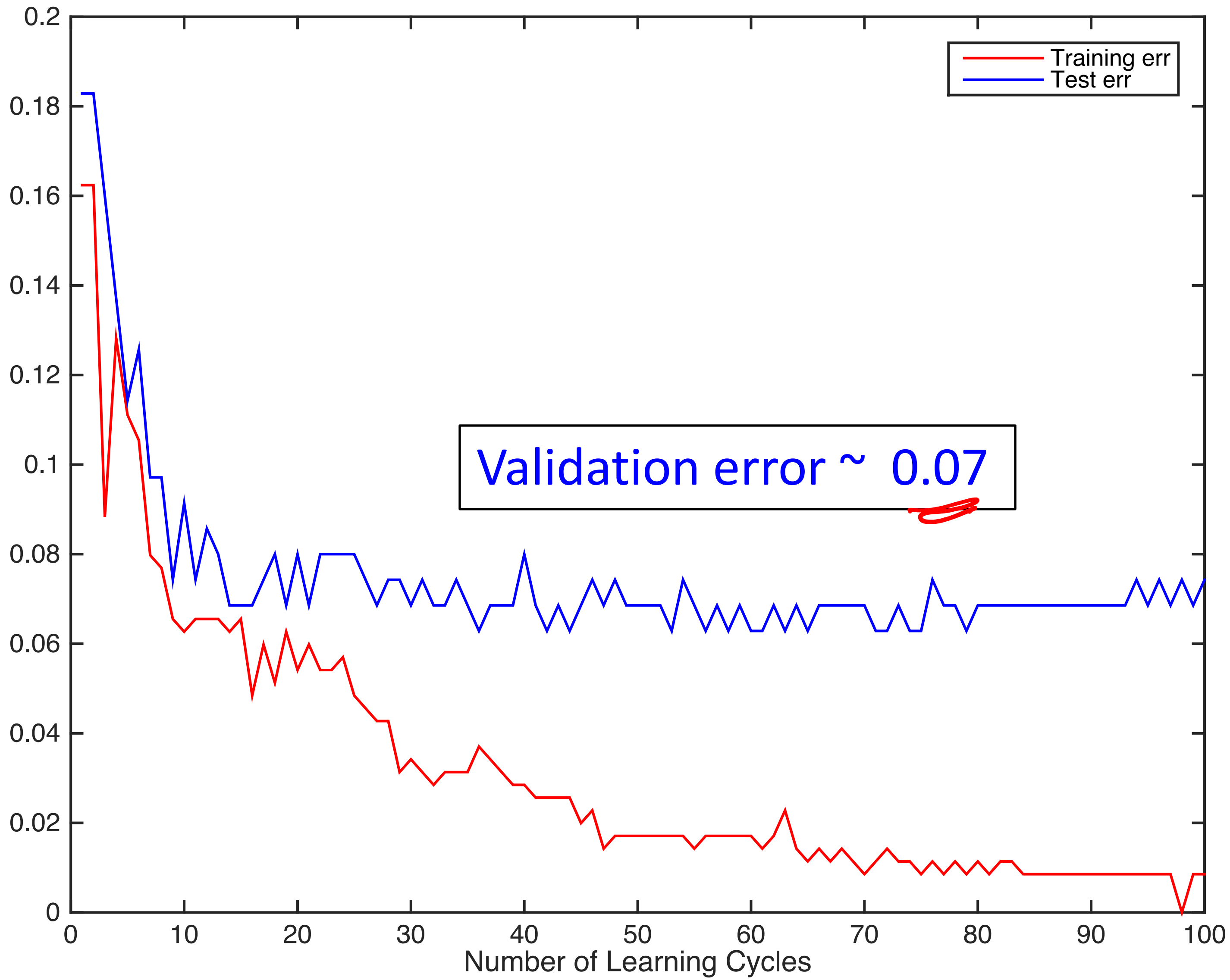


MinLeafSize 10 MinLeafSize 2 MinLeafSize 1

Matlab example - boosting

- % UCI dataset
- % 34 features, 351 samples
- % binary classification
- load ionosphere;
- rng(2); % For reproducibility
- ClassTreeEns = fitensemble(X,Y,'AdaBoostM1',100,'Tree');
- rsLoss = resubLoss(ClassTreeEns,'Mode','Cumulative');
- plot(rsLoss,'r');
- hold on
- ClassTreeEns = fitensemble(X,Y,'AdaBoostM1',100,'Tree',...
-     'Holdout',0.5);
- genError = kfoldLoss(ClassTreeEns,'Mode','Cumulative');
- plot(genError,'b');
- xlabel('Number of Learning Cycles');
- legend('Training err', 'Test err')

Matlab example - boosting



What you should know

- Estimating test error using
  - hold-out
  - cross-validation
- Bias-variance tradeoff
- Model selection using
  - hold-out
  - cross-validation