Evaluation

Anietie Andy (Andy)

These slides were assembled by Eric Eaton, with grateful acknowledgement of the many others who made their course materials freely available online. Feel free to reuse or adapt these slides for your own academic purposes, provided that you include proper attribution. Please send comments and corrections to Eric.

Announcements

- TA office hours
- http://www.seas.upenn.edu/~cis519/spring2018/staff.html

- **Recitation at 3401 Walnut St, Rm 401B**
  - Tuesdays: 6:30pm – 7:30pm
  - Wednesdays: 5:30pm – 6:30pm
- Homework submission Instructions

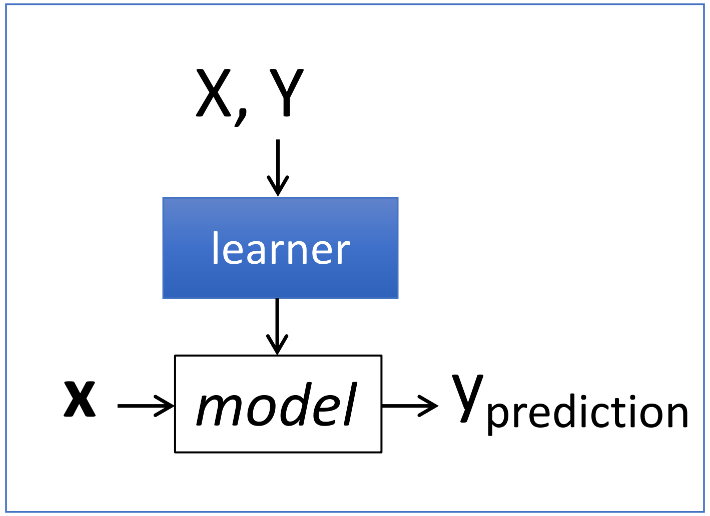
Stages of (Batch) Machine Learning

Given: labeled training data $X, Y = \{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^n$

- Assumes each $\boldsymbol{x}_i \sim \mathcal{D}(\mathcal{X})$ with $y_i = f_{target}(\boldsymbol{x}_i)$

Train the model:

model ← *classifier*.train(X, Y)



Apply the model to new data:

- Given: new unlabeled instance $\boldsymbol{x} \sim \mathcal{D}(\mathcal{X})$

$y_{\text{prediction}}$ ← *model*.predict(**x**)

Classification Metrics

$$\text{accuracy} = \frac{\#\ \text{correct predictions}}{\#\ \text{test instances}}$$

$$\text{error} = 1 - \text{accuracy} = \frac{\#\ \text{incorrect predictions}}{\#\ \text{test instances}}$$

Recall

$$\text{Recall} = \frac{\text{No. of relevant records retrieved}}{\text{Total no. of relevant records in the database}}$$

Precision

$$\text{Precision} = \frac{\text{No. of relevant records retrieved}}{\text{Total no. of records retrieved from the database}}$$

Precision vs. Recall

An inverse relationship

As the level of recall rises the level of precision generally declines and vice versa.

The Cranfield experiments (1957 & 1962)
Cyril Cleverdon, p.i.

Confusion Matrix

- Given a dataset of P positive instances and N negative instances:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

- Imagine using classifier to identify positive cases (i.e., for information retrieval)

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

Probability that a randomly selected result is relevant

Probability that a randomly selected relevant document is retrieved

Example

N = 165	Predicted: Yes	Predicted: No	Total
Actual: Yes	TP = 100	FN=5	**105**
Actual: No	FP = 10	TN = 50	**60**
Total	**110**	**55**	

Accuracy = (TP+TN)/N = (100+50)/165 = 0.91

Precision = TP/TP + FP = 100/110 = 0.91

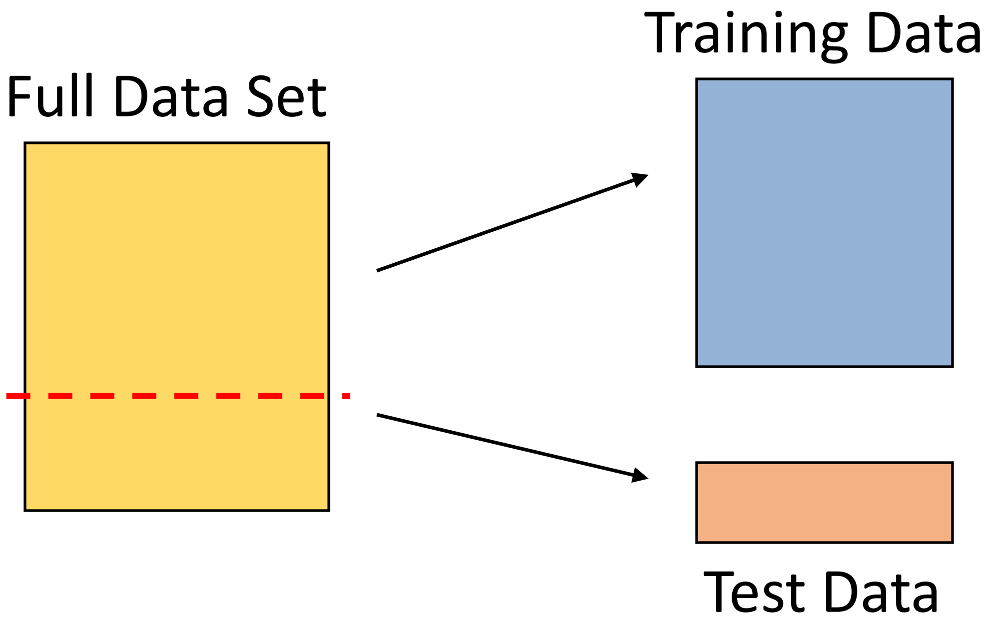
Recall = TP / TP + FN = 100/105 = 0.95

Training Data and Test Data

- Training data: data used to build the model
- Test data: new data, not used in the training process

- Training performance is often a poor indicator of generalization performance
  - Generalization is what we <u>really</u> care about in ML
  - Easy to overfit the training data
  - Performance on test data is a good indicator of generalization performance
  - i.e., test accuracy is more important than training accuracy

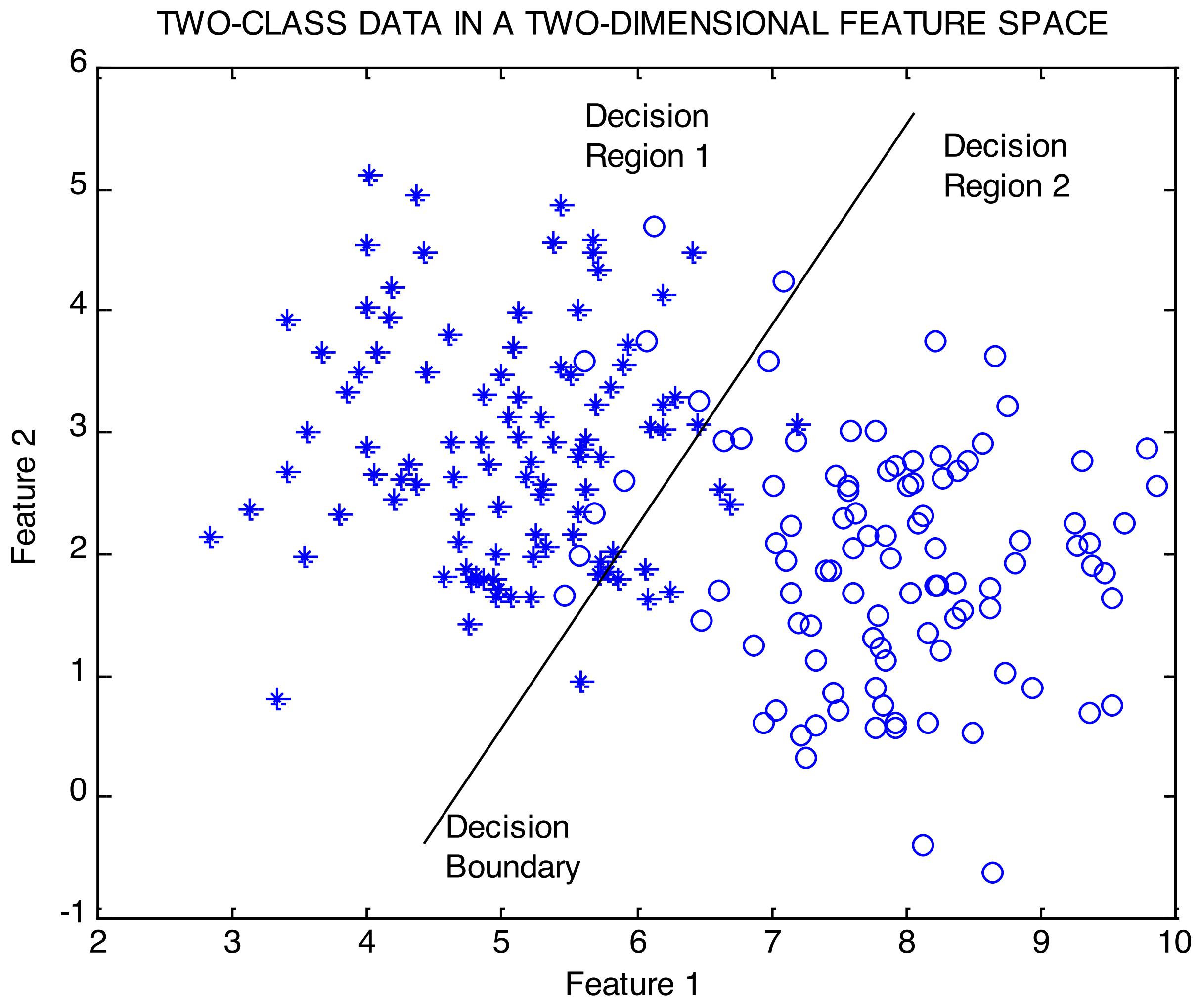
Training and Test Data



Idea:
Train each model on the "training data"...

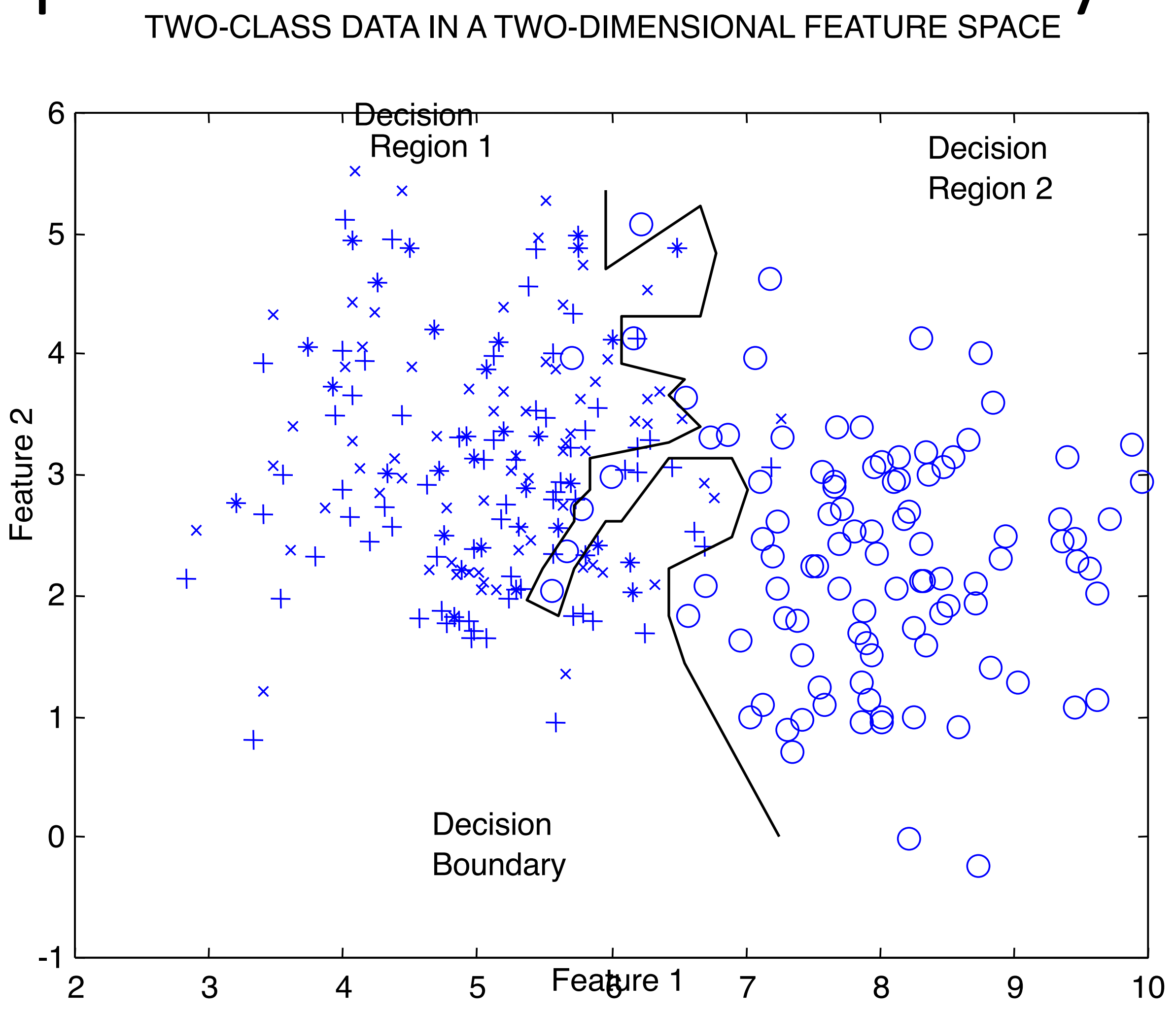
...and then test each model's accuracy on the test data

Simple Decision Boundary



Slide by Padhraic Smyth, UCIrvine

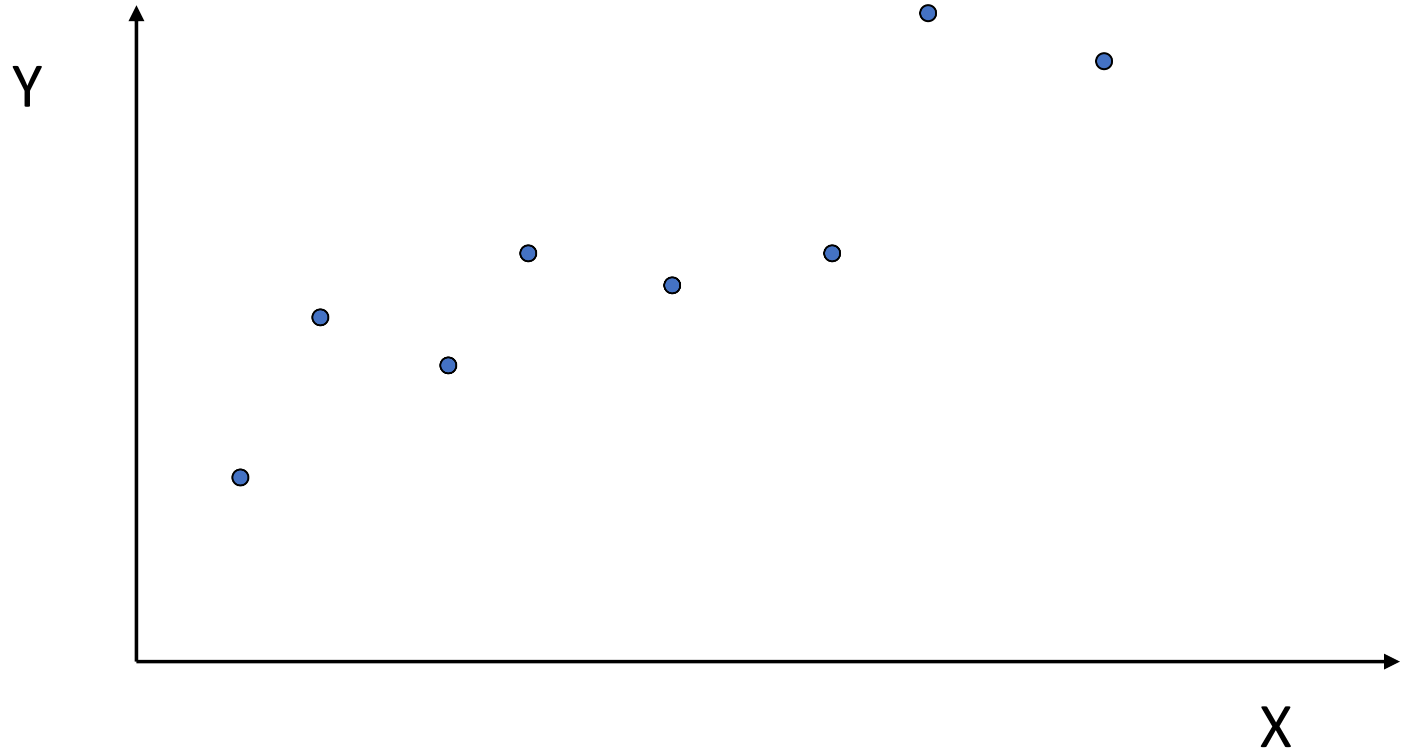
More Complex Decision Boundary



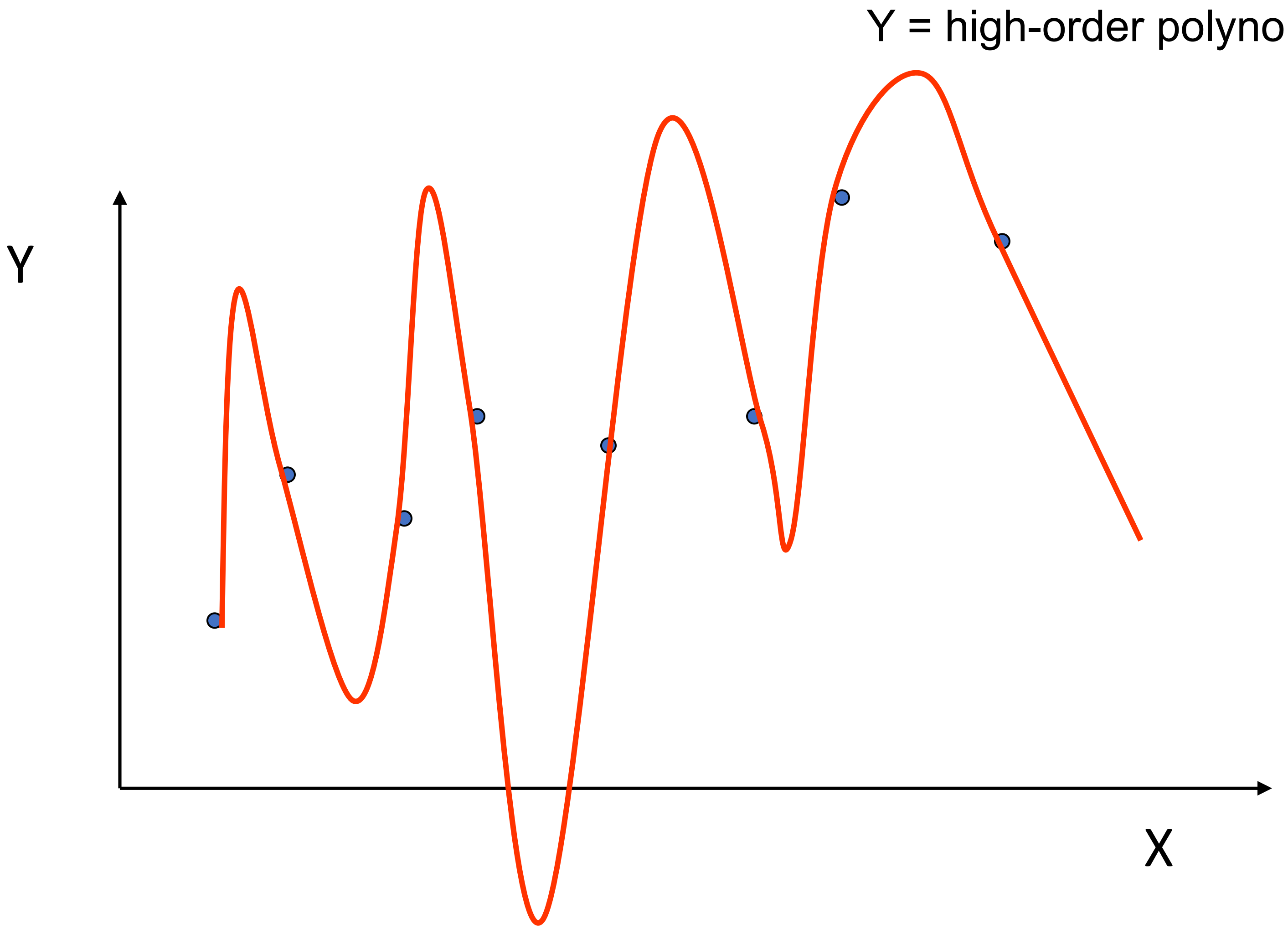
Overfitting

- “Fitting the data more than is warranted”

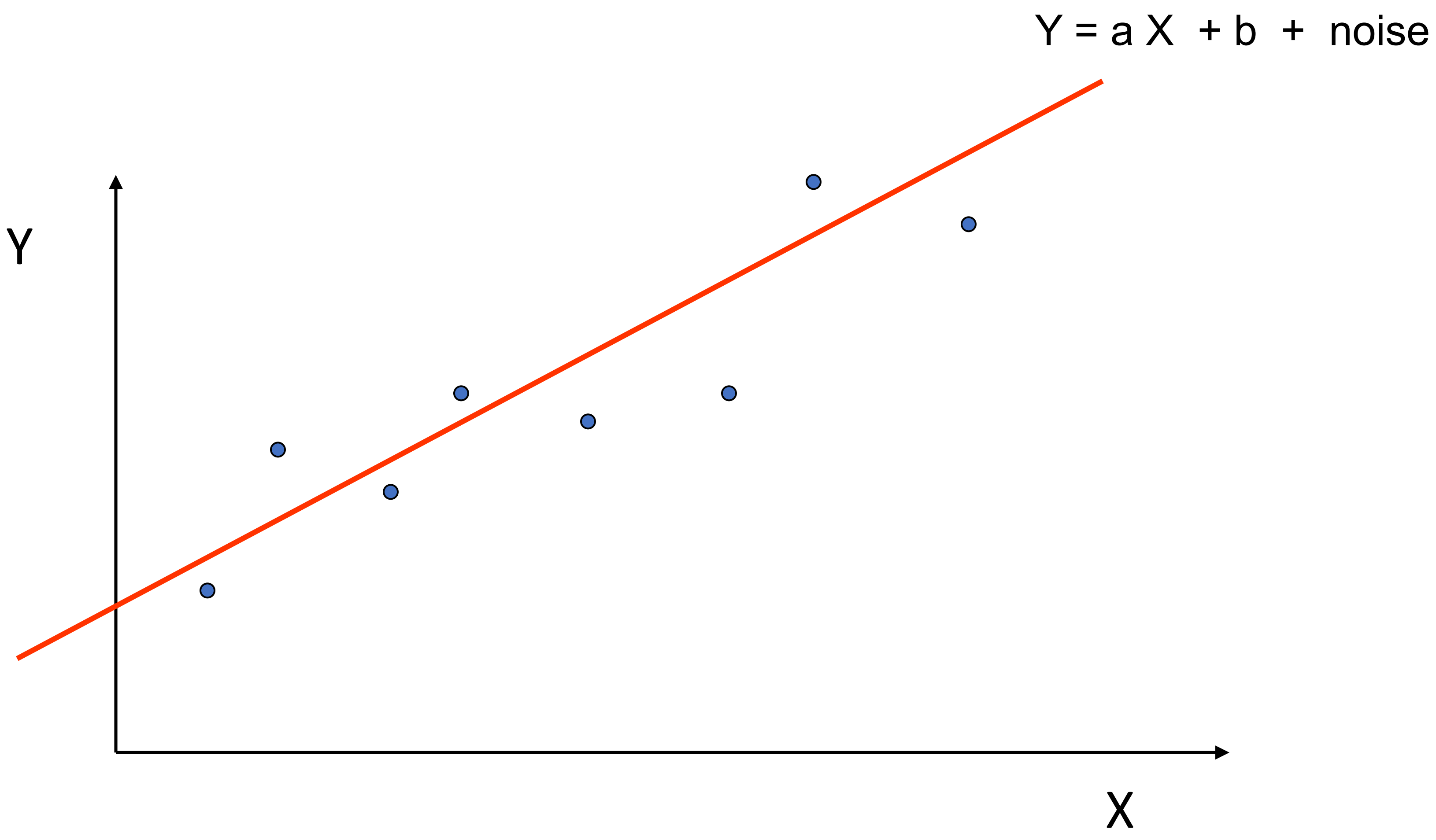
Example: The Overfitting Phenomenon



A Complex Model

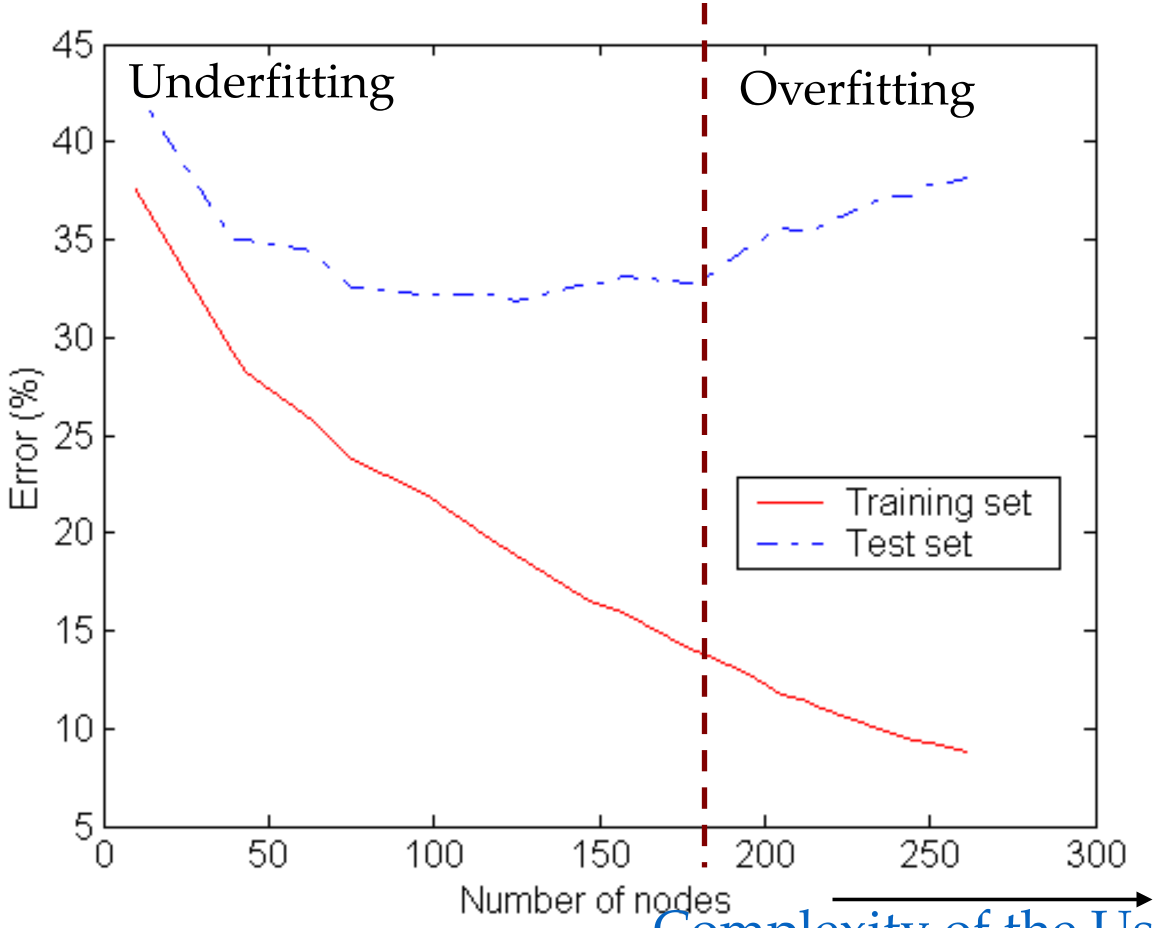


The True (simpler) Model



Slide by Padhraic Smyth, UCIrvine

Underfitting and Overfitting



Complexity of the Used Model

Complexity of a Decision Tree := number of nodes It uses

Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex and test errors are large although training errors are small.

Notes on Overfitting

- Overfitting results in models that are more complex than necessary: after learning knowledge they “tend to learn noise”

- More complex models tend to have more complicated decision boundaries and tend to be more sensitive to noise, missing examples,…

- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records

Comparing Classifiers

Say we have two classifiers, *C1* and *C2*, and want to choose the best one to use for future predictions

Can we use training accuracy to choose between them?

- No!
  - e.g., C1 = pruned decision tree, C2 = 1-NN

    training_accuracy(1-NN) = 100%, but may not be best

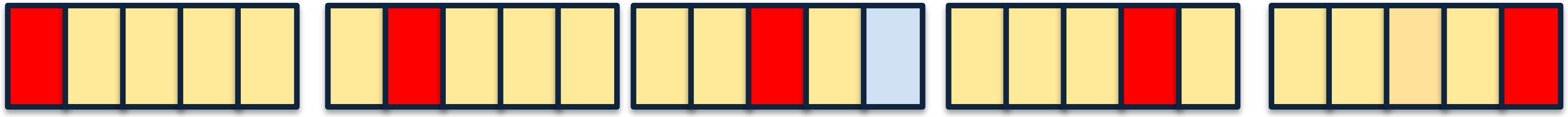
Instead, choose based on test accuracy...

Based on Slide by Padhraic Smyth, UCIrvine

N-fold cross validation

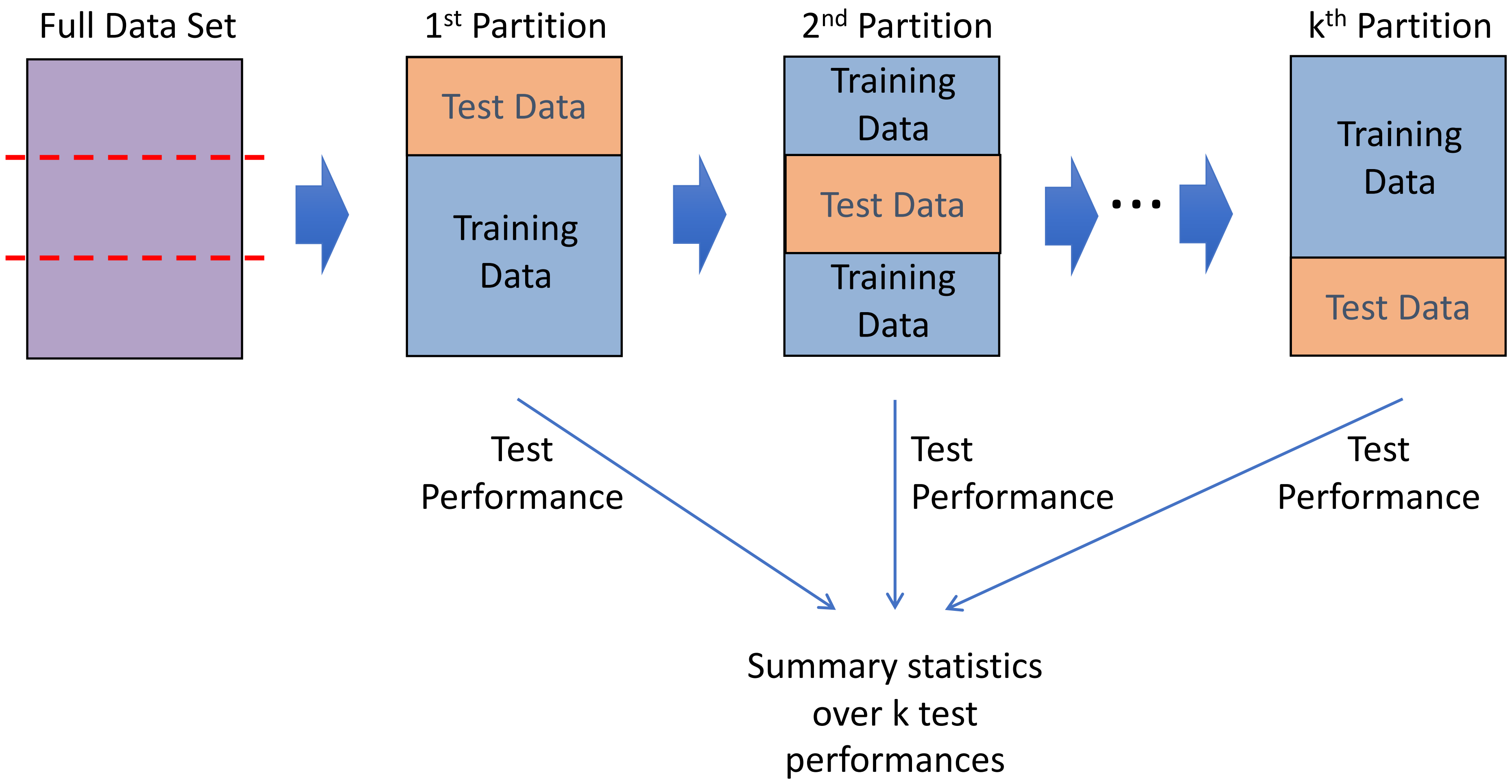
- Instead of a single test-training split:

- Split data into N equal-sized parts



- Train and test N different classifiers
- Report average accuracy and standard deviation of the accuracy

Example 3-Fold CV



More on Cross-Validation

- Cross-validation generates an approximate estimate of how well the classifier will do on “unseen” data
  - As k → n, the model becomes more accurate (more training data)
  - ...but, CV becomes more computationally expensive
  - Choosing k < n is a compromise

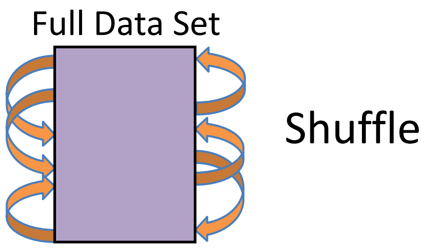
- Averaging over different partitions is more robust than just a single train/validate partition of the data

- It is an even better idea to do CV repeatedly!

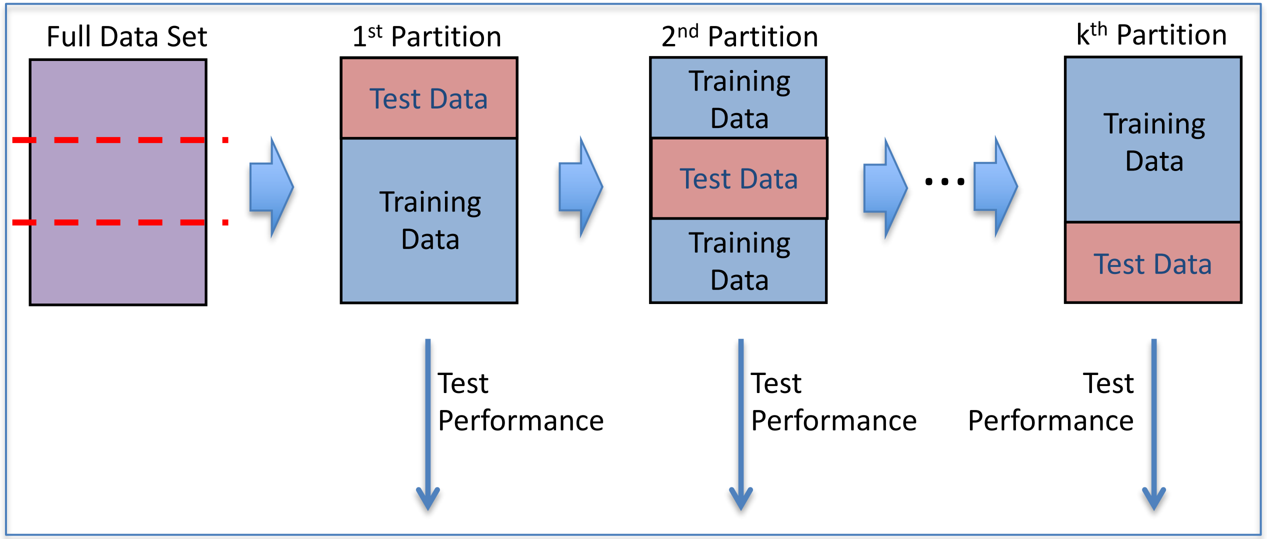
Multiple Trials of k-Fold CV

1.) Loop for t trials:

a.) Randomize Data Set



b.) Perform k-fold CV



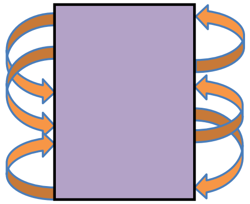
2.) Compute statistics over t x k test performances

Comparing Multiple Classifiers

1.) Loop for t trials:

a.) Randomize Data Set

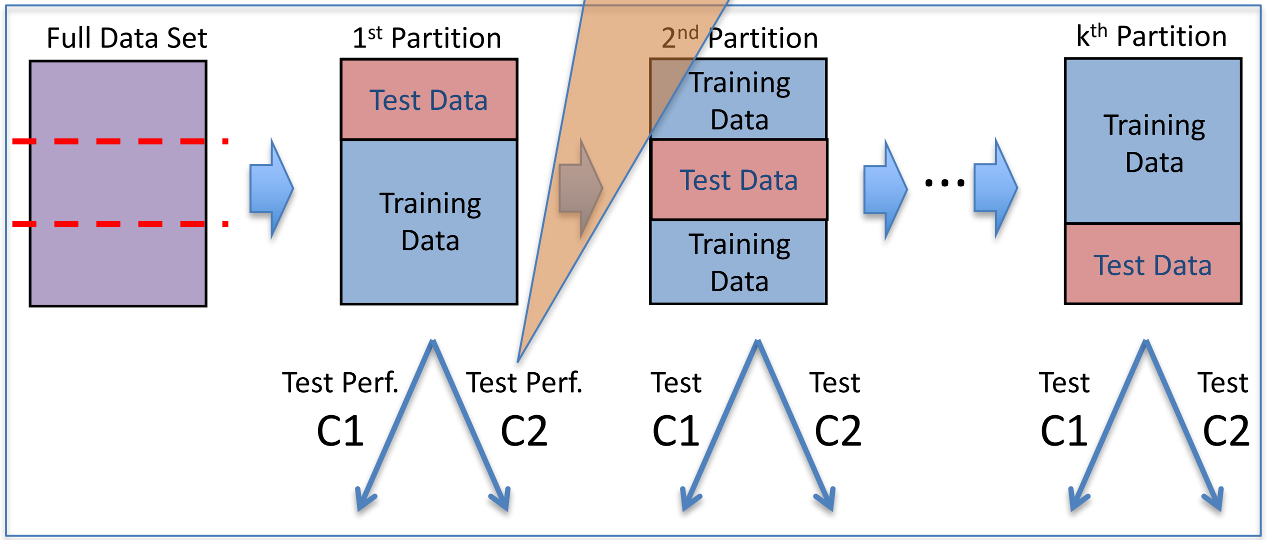
Full Data Set



Shuffle

Test each candidate learner on same training/testing splits

b.) Perform k-fold CV



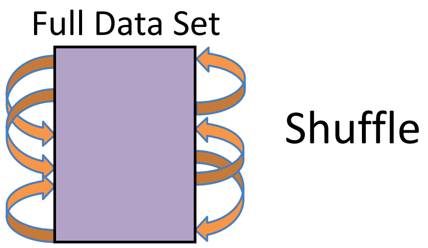
2.) Compute statistics over t x k test performances

Allows us to do paired summary statistics (e.g., paired t-test)

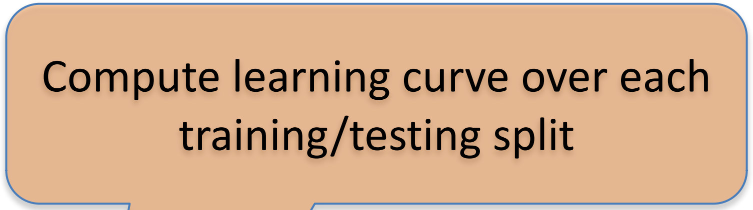
Building Learning Curves

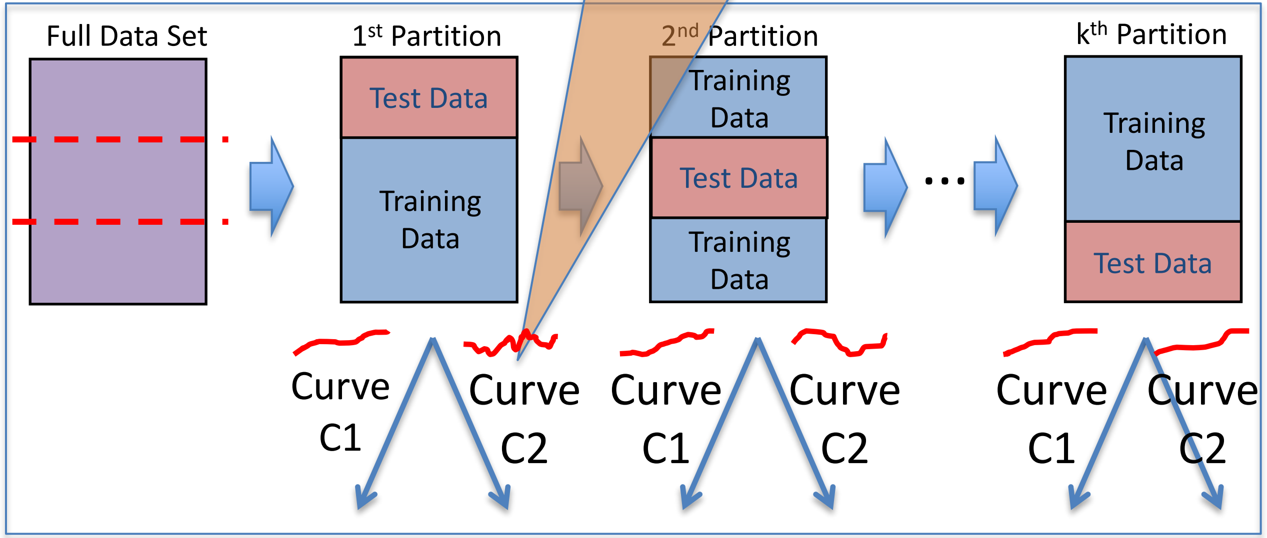
1.) Loop for t trials:

a.) Randomize Data Set



b.) Perform k-fold CV





2.) Compute statistics over t x k learning curves

Hypothesis testing

- You want to show that hypothesis H is true, based on your data
  – (e.g. H = "classifier A and B are different")
- Define a null hypothesis H_0
  – (H_0 is the contrary of what you want to show)
- H_0 defines a distribution $P(m \mid H_0)$ over some statistic
  – e.g. a distribution over the difference in accuracy between A and B
- Can you refute (reject) H_0?

Rejecting H_0

- H_0 defines a distribution $P(M \mid H_0)$ over some statistic M
  - (e.g. M= the difference in accuracy between A and B)
- Select a significance value S
  - (e.g. 0.05, 0.01, etc.)
  - You can only reject H0 if $P(m \mid H_0) \leq S$
- Compute the test statistic m from your data
  - e.g. the average difference in accuracy over your N folds
- Compute $P(m \mid H_0)$
- Refute H_0 with $p \leq S$ if $P(m \mid H_0) \leq S$

Paired t-test

- A paired t-test is used to compare two population means where you have two samples in which observations in one sample can be paired with observations in the other sample.

- E.g.

  Before-and-after observations on the same subjects (e.g. students)

Procedure for carrying out Paired t-test

- Calculate the difference between the two observations on each pair.
- Calculate the mean difference
- Calculate the standard deviation of the differences
- Calculate the error of the mean difference
- Calculate the t-statistic

Paired t-test example

- **<u>Question:</u>** The downtimes (measured in hours) for computer systems in six branches of a major bank were recorded for year 1 and year 2. Compute the test statistics for the paired t-test.
- **<u>Solution:</u>**

Branch	Year 1	Year 2	Difference (Year 1 – Year 2)	Square of Difference
A	40	30	10	100
B	54	41	13	169
C	32	24	8	64
D	36	38	-2	4
E	55	56	-1	1
F	46	37	9	81
			Sum = 37	**Sum = 419**

Paired t-test

- Sample size: n = 6
- Sum of differences $\sum d_i$ = 37
- Sum of squared differences $\sum d_i^2$ = 419
- Mean of case-wise differences: $\bar{d} = \frac{\sum d_i}{n}$ = 37/6 = 6.166
- Standard Deviation : $\mathbf{Sd} = \frac{\sum \boldsymbol{d_i} - \boldsymbol{n}\bar{\boldsymbol{d}}^{\mathbf{2}}}{\boldsymbol{n}-\mathbf{1}}$ = $\sqrt{\frac{419 - 6 \times (6.166)^2}{5}} = 6.177$
- Test statistics for the paired t-test: $t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}$ = 2.445

McNemar's Test

- The test is often used for the situation where one tests for the presence (1) or absence (0) of something and variable A is the state at the first observation (i.e., pretest) and variable B is the state at the second observation (i.e., posttest).

McNemar's Test

- An alternative to Cross Validation, when the test can be run only once
- Divide the sample S into a training set R and a test set T.
- Train algorithms A and B on R, yielding classifiers A,B
- Record how each example in T is classified and compute the number of

Examples misclassified by both A and B N_{00}	Examples misclassified by A but not B N_{01}
Examples misclassified by B but not A N_{10}	Examples misclassified by neither A nor B N_{11}

where N is the total number of examples in the test set T

$$\mathbf{N_{00} + N_{10} + N_{01} + N_{11} = N}$$

McNemar's Test

- The hypothesis: the two learning algorithms have the same error rate on a randomly drawn sample. That is, we expect that

$$N_{10} = N_{01}$$

- The statistics we use to measure deviation from the expected counts:

$$\frac{(|N_{01} - N_{10}| - 1)^2}{N_{01} + N_{10}}$$

END