CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Today

- Evaluation Methodology and Metrics

Goal

- Want a reliable measure of expected future performance of the learning algorithm on a specific learning problem

- How to measure future performance?
- How to get expectation?

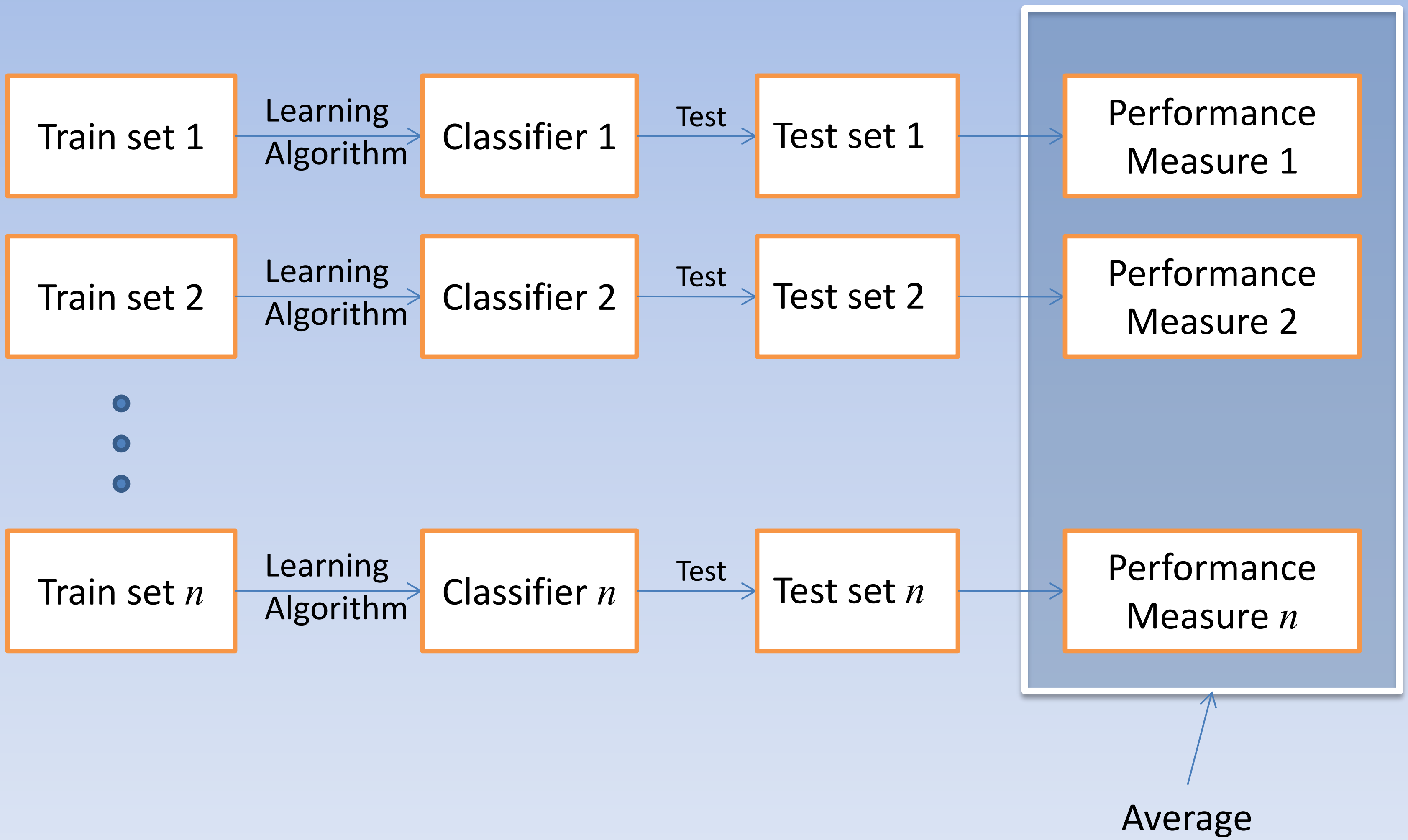
Idea

- Separate available data into sets for training and evaluation

- The examples for evaluation will be new to the learned classifier
  - Proxy for "future examples"

- Do this lots of times to get expectation

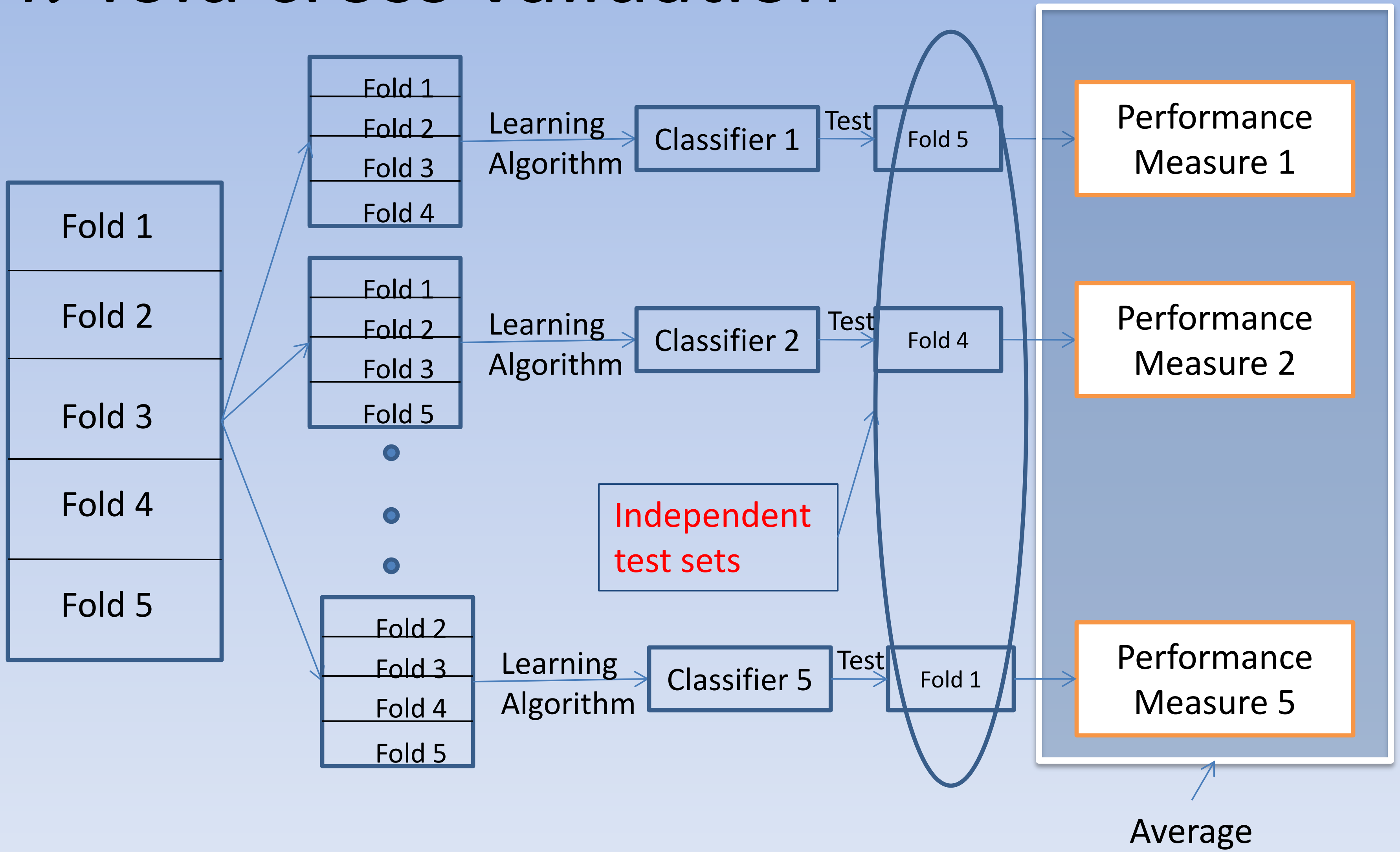
Ideal case


Train set 1
Learning Algorithm
Classifier 1
Test
Test set 1
Performance Measure 1
Train set 2
Learning Algorithm
Classifier 2
Test
Test set 2
Performance Measure 2
Train set n
Learning Algorithm
Classifier n
Test
Test set n
Performance Measure n
Average

n-fold cross validation

- Generally, data is limited
- To learn a good concept, need training sets to be *as large as possible*
- For good estimates of future performance, need a number of *independent test sets*
- Idea: partition the available examples into “folds”

n-fold cross validation



Special case: Leave-one-out

- N examples, N folds
  - Each "test set" has only one example

- Useful if few examples

- Called "jackknife" in statistics literature

Stratified Cross Validation

- Same as cross validation, but folds are sampled so the proportions of class labels are the same in each fold and equal to the overall proportion

- Produces more stable performance estimates overall, recommended

Internal Cross Validation

- Can use same method to tune parameters, select features, prune trees etc

- Do another m-fold c.v. *within each fold*
  - In this case, held out data called “validation set” or “tuning set”
  - Each fold might produce different parameter settings
    - Need a consensus procedure to identify a single setting
- Needs many examples to work well

Contingency Table

		Class according to Target Concept / Oracle (Correct Answer)	
		Positive	**Negative**
Class according to Learned Classifier (Predicted Answer)	**Positive**	True Positives (TP)	False Positives (FP) (Type I error)
	Negative	False Negatives (FN) (Type II error)	True Negatives (TN)

Accuracy

- Most commonly used measure for comparing classification algorithms

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Error Rate

- Inverse of Accuracy

$$ErrorRate = \frac{FP + FN}{TP + TN + FP + FN}$$

Weaknesses of Accuracy

- Does not account for:
  - Skewed class distributions
  - Differential misclassification costs
  - Confidence estimates from learning algorithms

Weighted/Balanced Accuracy

- Corrects for skewed class distributions

$$WAcc = \frac{1}{2}\left(\frac{TP}{Allpos} + \frac{TN}{Allneg}\right)$$

$$= \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

True Positive Rate

True Negative Rate

Measuring one class

- Often, just a single class is "interesting"
  - Call this the "positive" class

	Positive	Negative
Positive	True Positives (TP)	False Positives (FP) (Type I error)
Negative	False Negatives (FN) (Type II error)	~~True Negatives (TN)~~

Precision

- Of the examples the learner predicted positive, how many were actually positive?

$$Precision = \frac{TP}{TP + FP}$$

Recall/TP rate/Sensitivity

- Of the examples that were actually positive, how many did the learner predict correctly?

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Allpos}$$

Specificity/TN rate

- Counterpart of recall for the negative class

$$Specificity = \frac{TN}{TN+FP} = \frac{TN}{Allneg}$$

- So:

$$WAcc = \frac{1}{2}\left(Sensitivity + Specificity\right)$$

F_1 score

- Combines precision and recall into a single measure, giving each equal weight

$$\frac{1}{F_1} = \frac{1}{2}\left(\frac{1}{Precision} + \frac{1}{Recall}\right)$$

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

Beyond point estimates

- Everything above is a "point estimate"
- Because they will be computed on the basis of a sample, we can also compute variance estimates for each quantity
- Important to show "stability" of solutions, and when comparing across algorithms (later)

Learning Curves

- Often useful to plot each metric as a function of training sample size
- Provides insight into how many examples the algorithm needs to become effective