CSDS 440: Machine Learning

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

Olin 516

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

Announcements

- Quiz 1 next Thursday, in class, 30-45 minutes, closed book/notes
  - Topics: everything up to and including decision trees
  - Remember to review probability and statistics
- HW2, Programming 1 assigned

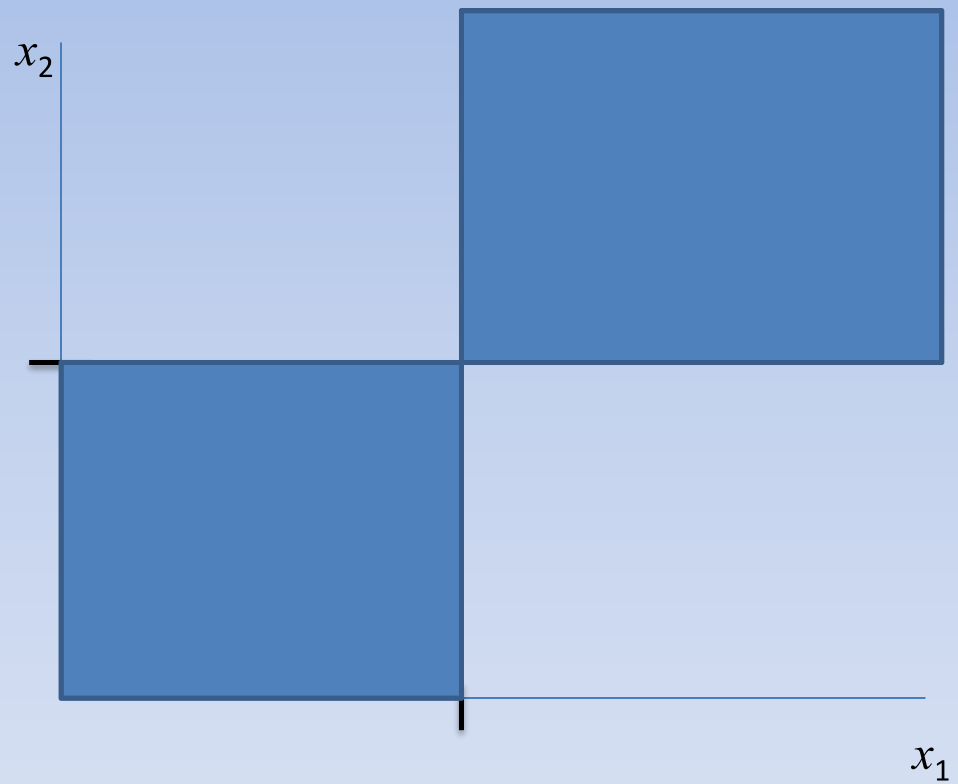
Recap

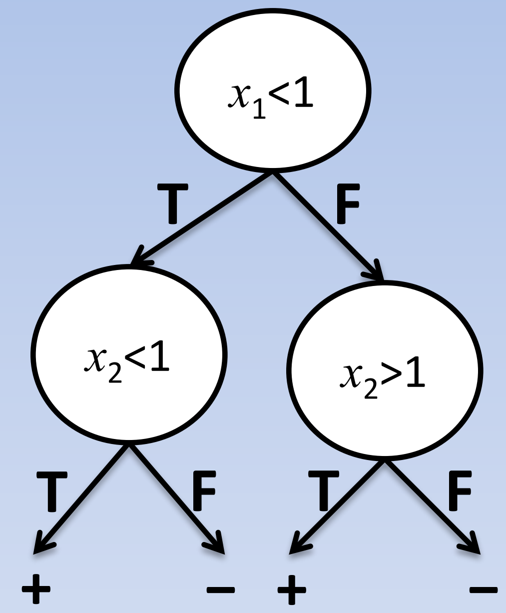
- We can/cannot remove continuous attributes after partitioning with them. Why?
- What is overfitting?
- We control overfitting in trees through (ES) and (PP).
- How does ES work? Why might it not work well in practice?
- PP uses a v_____ set. It iteratively d______ a node and evaluates the result on the v_____ set. It keeps the tree that has best performance. It stops when ______.

Today

- Decision Tree Geometry
- Evaluation Methodology and Metrics

Decision Tree Geometry (Continuous features)





Extensions

- Idea can be extended to handle:
  - Multiclass classification
  - Regression
  - Functional tests in internal nodes (Function Trees)
  - More complex functions in leaves (Model Trees)
  - Density functions in leaves (PETs)

Pros and Cons of Decision Trees

- + Does not require metric space representation
- + Produces human-comprehensible concepts
- + Can produce concepts with range of complexity
- + Easily extendable to various other scenarios
- + Easy to combine with other algorithms (general purpose partitioning)

- − Attributes with lots of values (including continuous attributes)
- − Attributes with complex interactions
- − Partitioning strategy means easier to overfit as depth increases

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?

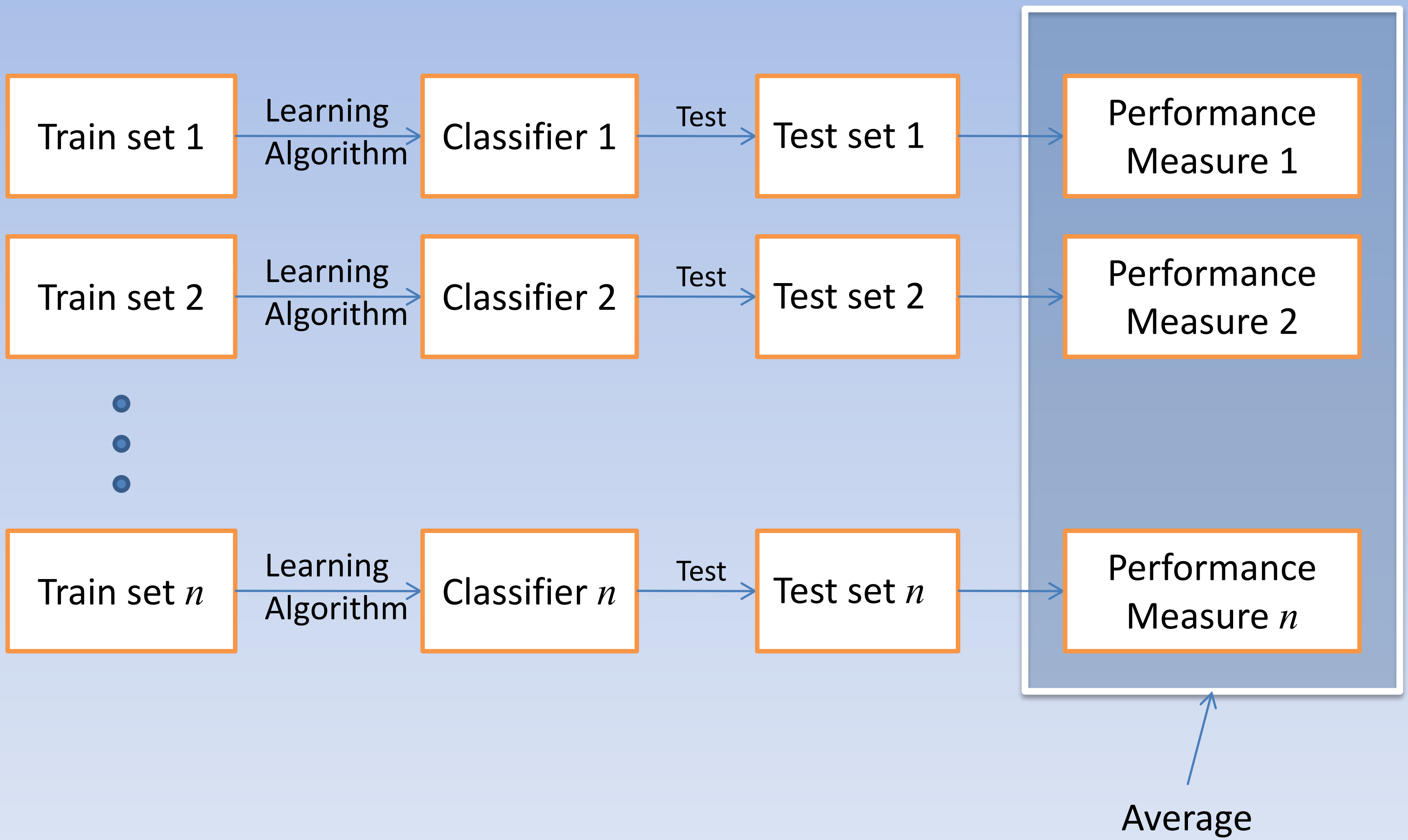
Idea

- 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

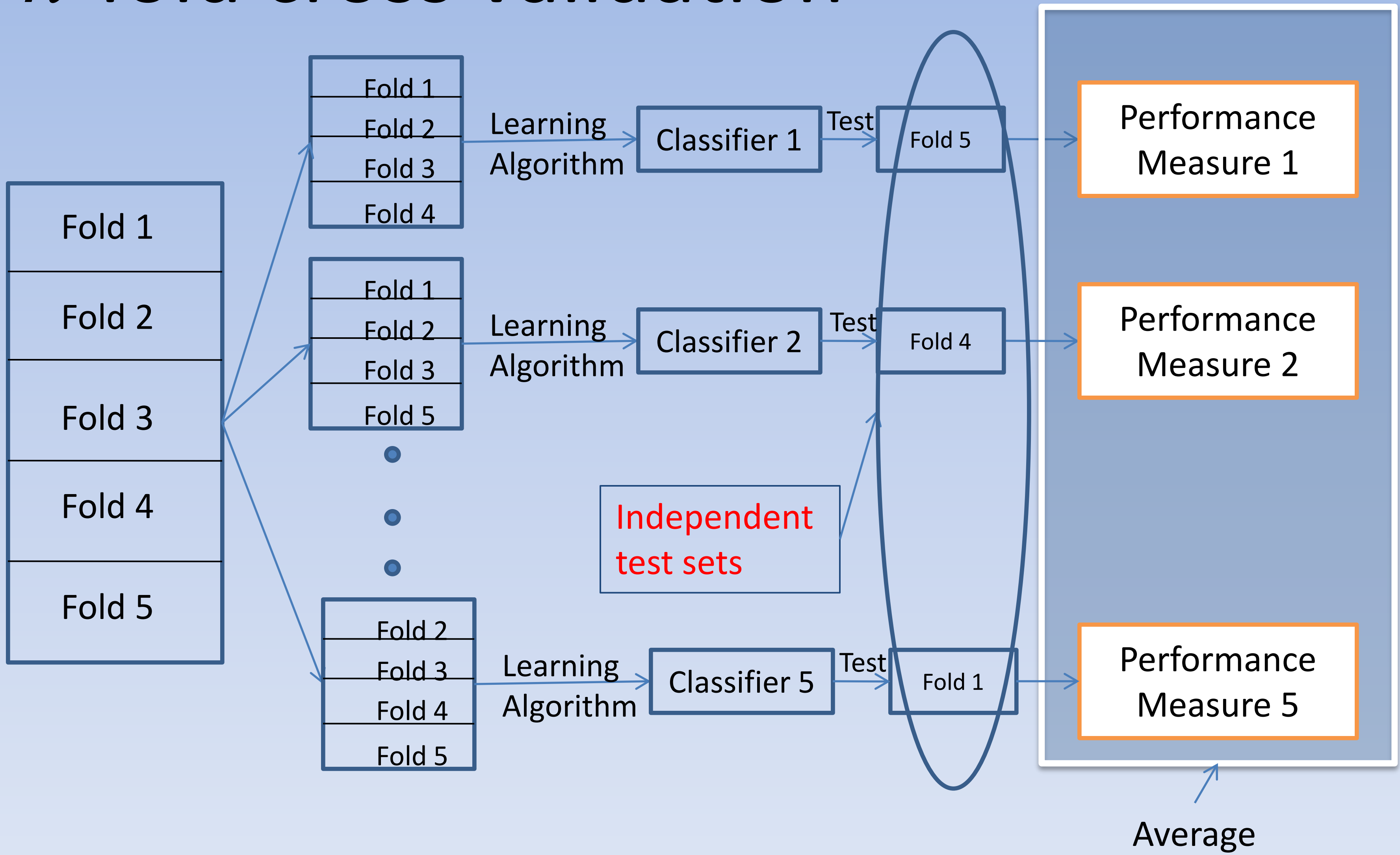


Train set 1
Learning Algorithm
Classifier 1
Test
Test set 1
Performance Measure 1
Train set 2
Learning Algorithm
Classifier 2
Test
Test set 2
Performance Measure 2
Train set n
Learning Algorithm
Classifier n
Test
Test set n
Performance Measure n
Average

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