Hyper-Parameter Tuning & Practical ML

Rishabh Iyer

Based on slides from Jude Shavlik and Tom Dietterich

Proper Experimental Methodology Can Have a Huge Impact:

A 2002 paper in *Nature* (a major journal) needed to be corrected due to "training on the testing set"

Original report : 95% accuracy (5% error rate)

Corrected report (which still is buggy):

73% accuracy (27% error rate)

Error rate increased over 400%!!!

Experimental Methodology

1) Start with a dataset of labeled examples
2) Randomly partition into *N* groups
3a) *N* times, combine *N* -1 groups into a train set
3b) Provide training set to learning system
3c) Measure accuracy on "left out" group (the test set)

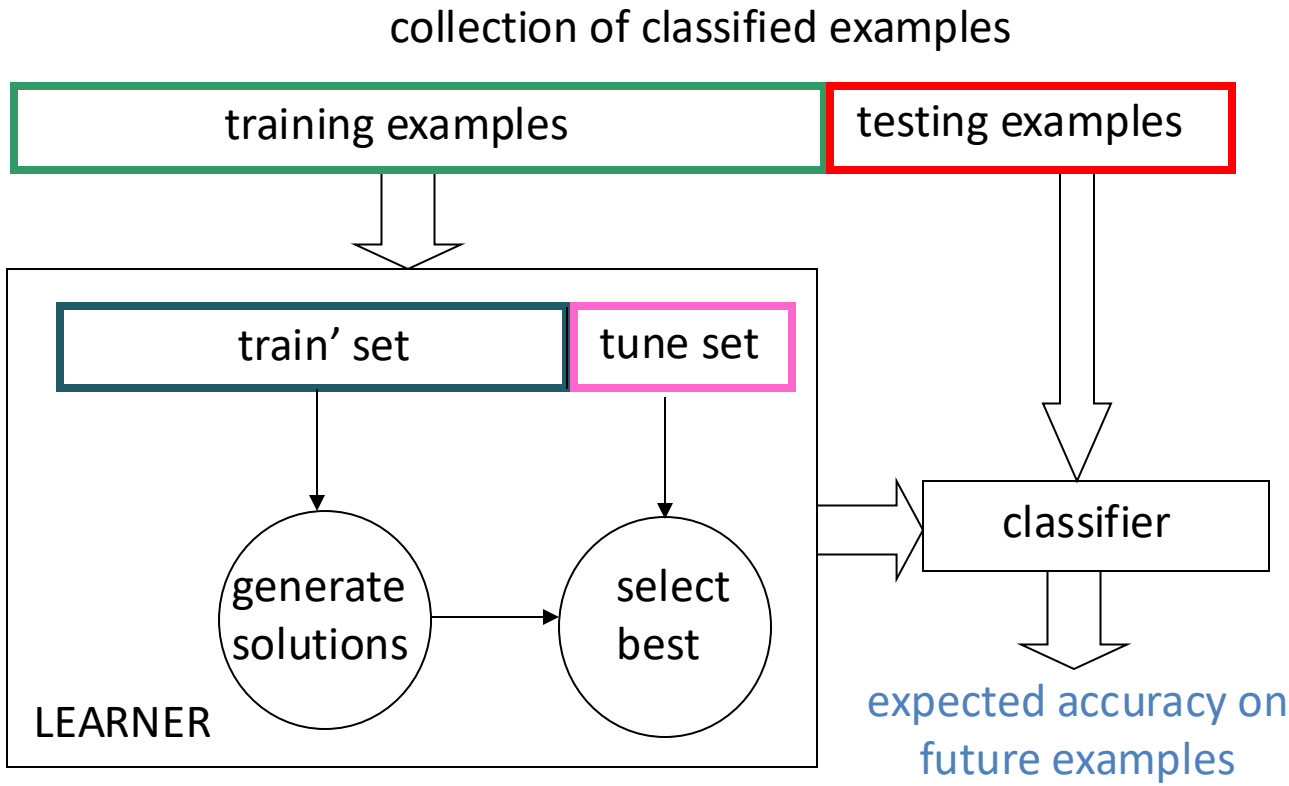
Called *N*-fold cross validation

Validation Sets

- Often, an ML system has to choose when to stop learning, select among alternative answers, etc.
- One wants the model that produces the highest accuracy on **future** examples ("overfitting avoidance")
- It is a **"cheat"** to look at the **test** set while still learning
- Better method
  - Set aside part of the training set
  - Measure performance on this validation data to estimate future performance for a given set of hyperparameters
  - Use best hyperparameter settings, train with **all** training data (except **test** set) to estimate future performance on **new** examples

A typical Learning system

Statistical techniques such as 10-fold cross validation and *t*-tests are used to get meaningful results



Multiple Tuning sets

- Using a **single** tuning set can be unreliable predictor, plus some data "wasted"

  1) For each possible set of hyperparameters

     a) Divide training data into **train** and **valid.** sets, using ***N*-fold cross validation**

     b) Score this set of hyperparameter values: average **valid.** set accuracy over the *N* folds

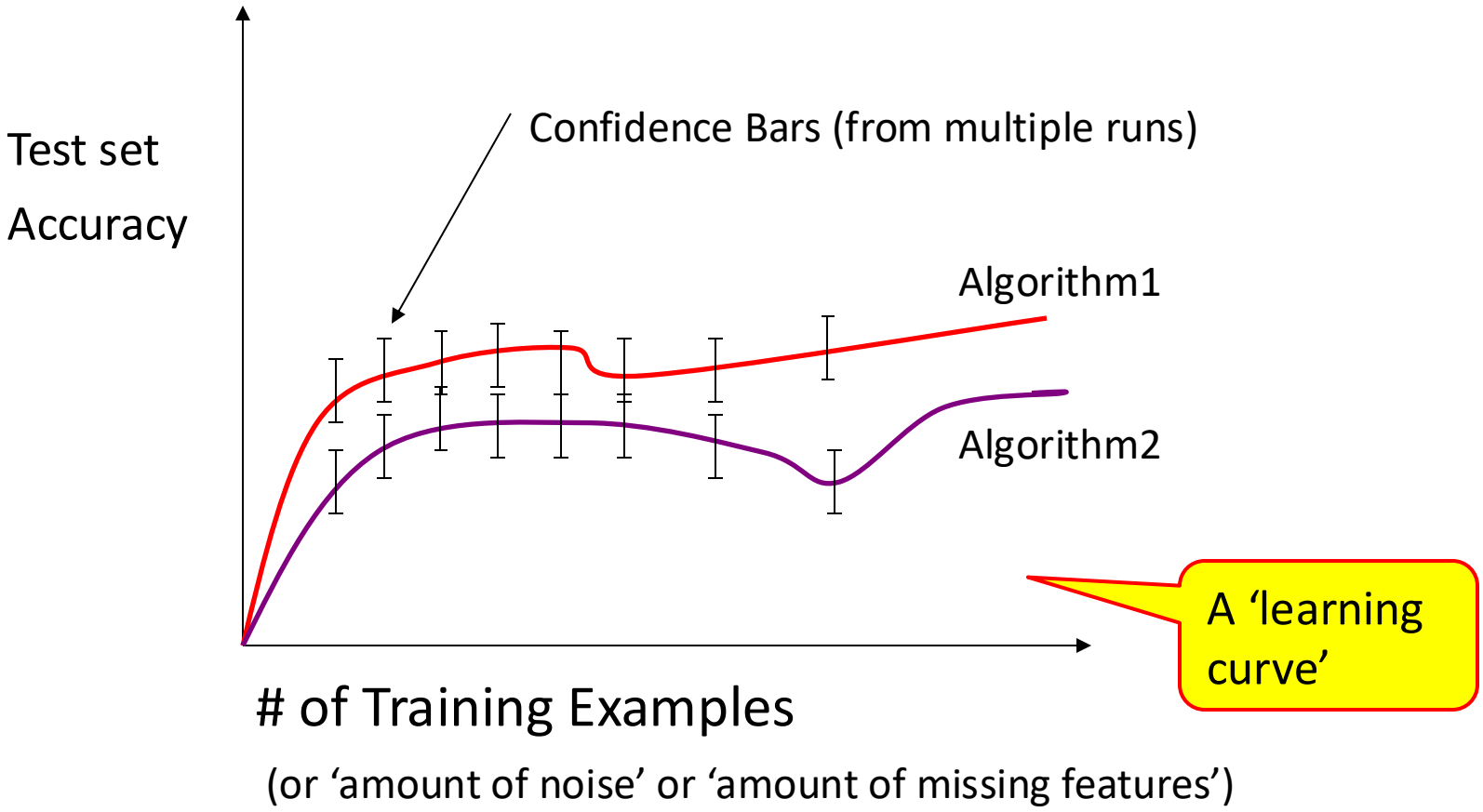
  2) Use **best** set of hyperparameter settings and **all** (train + valid.) examples

  3) Apply resulting model to **test** set

Example Hyper-Parameters

- Linear Regression & Polynomial Regression: Degree of Polynomial, Regression Coeff
- SVM, Perceptron: Regression Coeff
- Decision Trees: Depth of the Tree, Minimum no. of Leaf Nodes
- Nearest Neighbor Methods: K
- ….

Some Typical ML Experiments



Typical Experiments

	Test Set Performance
Full System	80%
Without Module A	75%
Without Module B	62%