Machine Learning: More Review

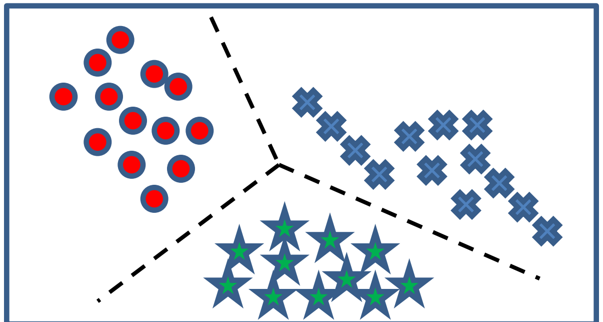
Alexander G. Ororbia II

Introduction to Machine Learning

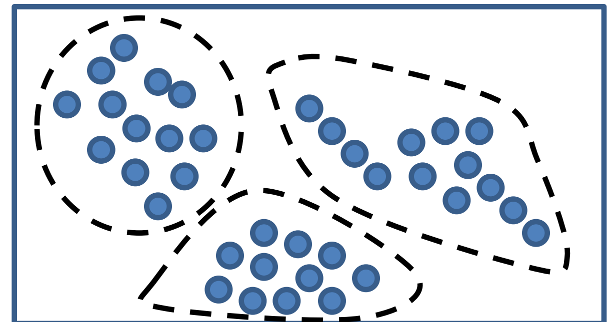
CSCI-736

1/16/2025

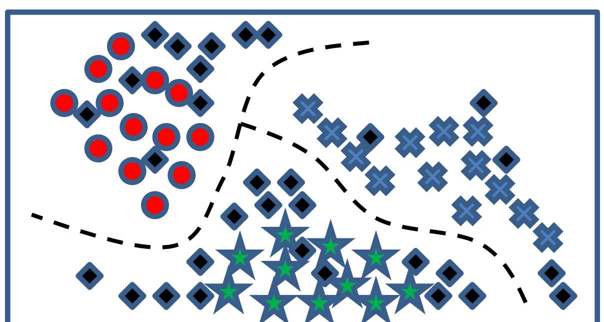
Visualizing Types of Learning



Supervised learning



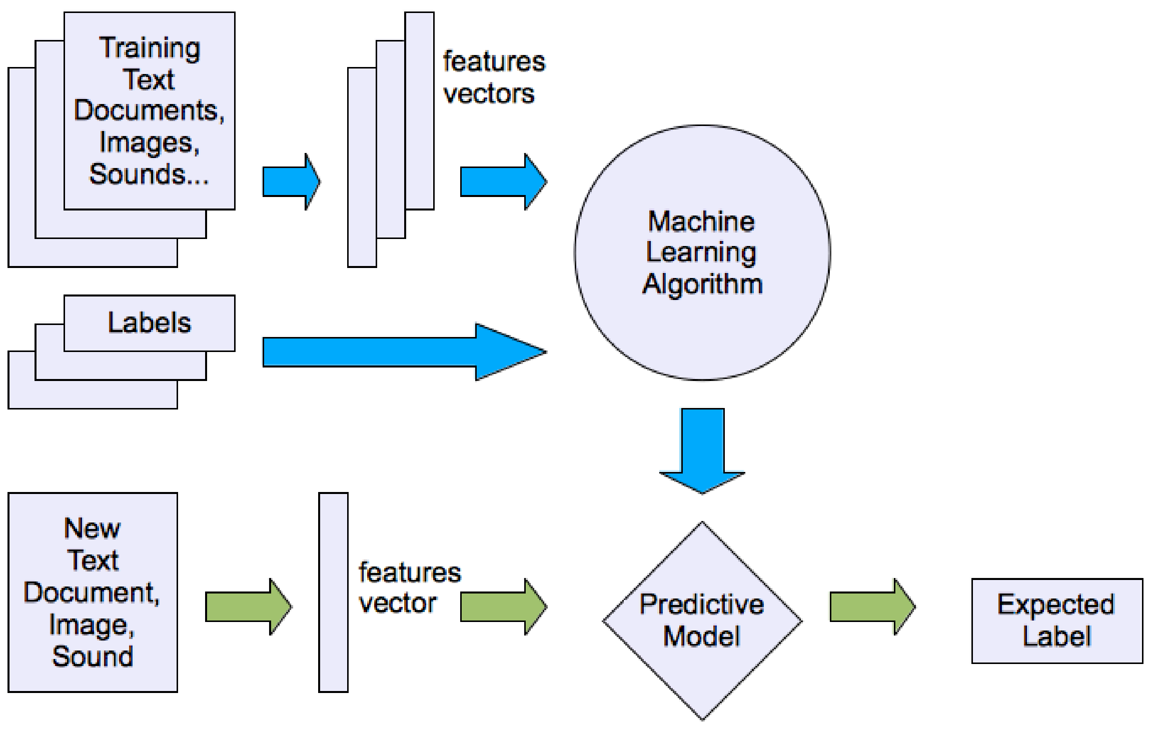
Unsupervised learning



Semi-supervised learning

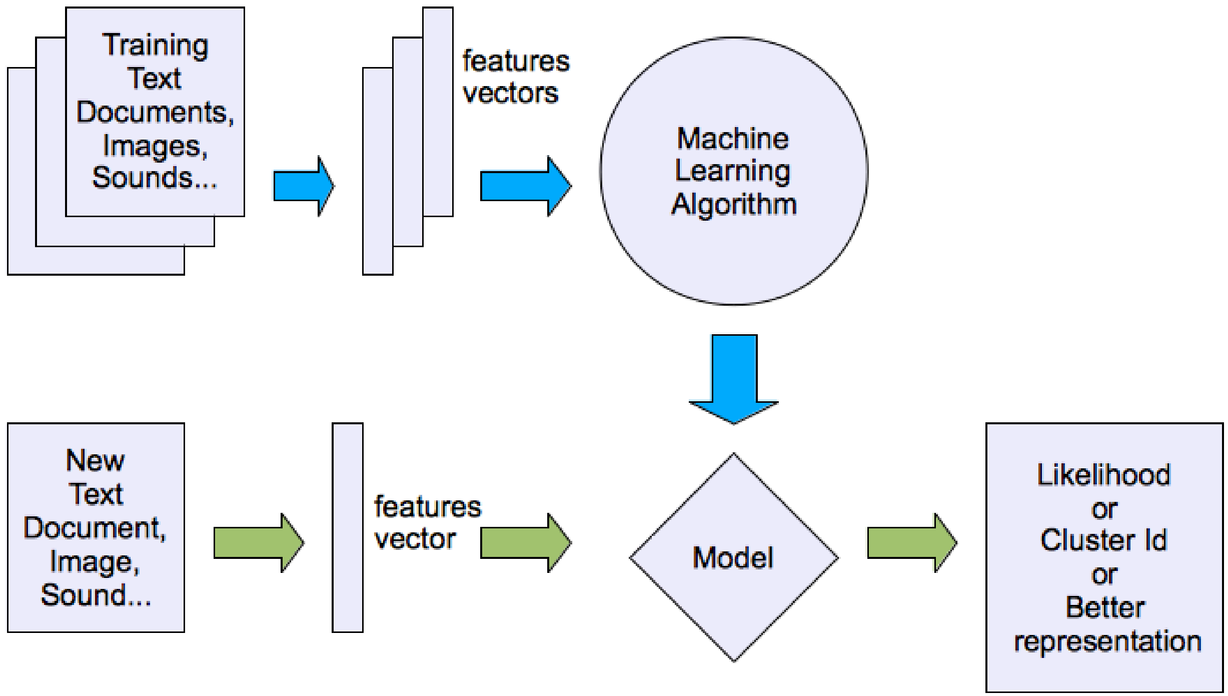
A Typical Supervised Learning Pipeline

- Supervised learning



A Typical Unsupervised Learning Pipeline

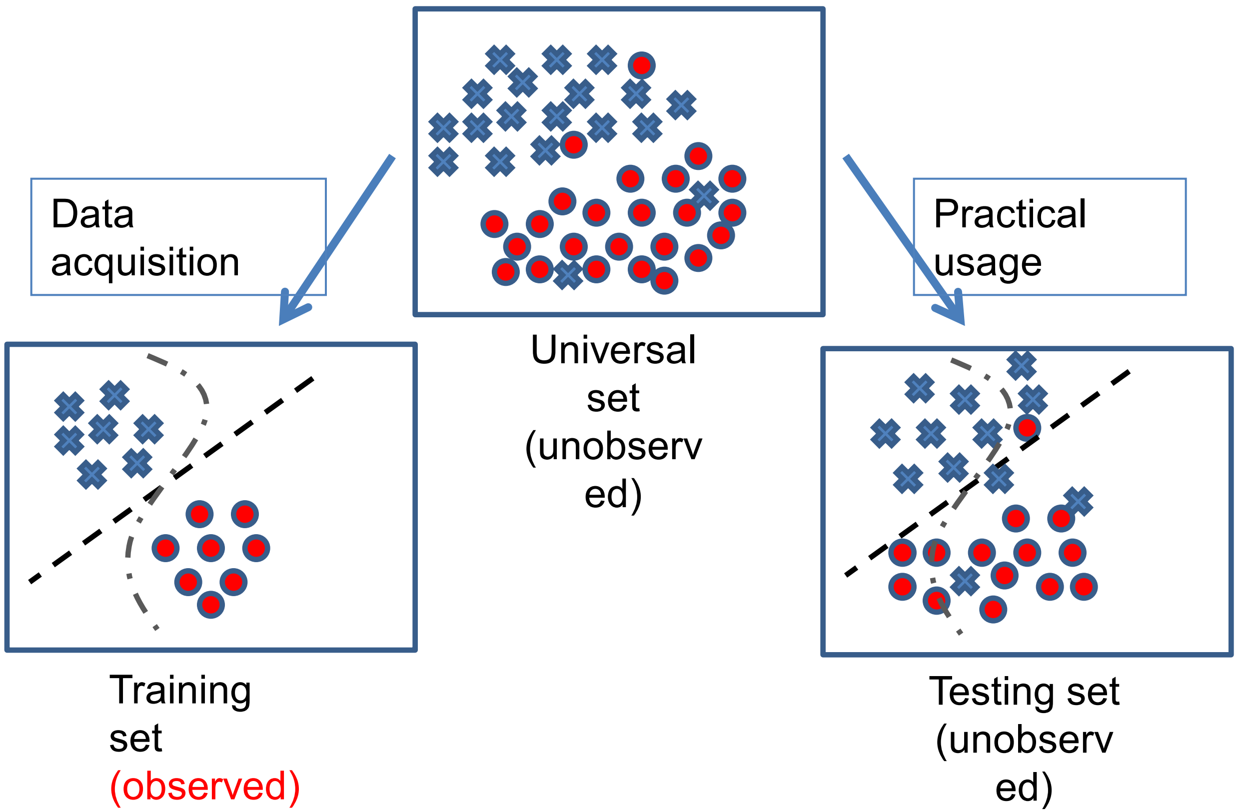
- Unsupervised learning



ML in Practice

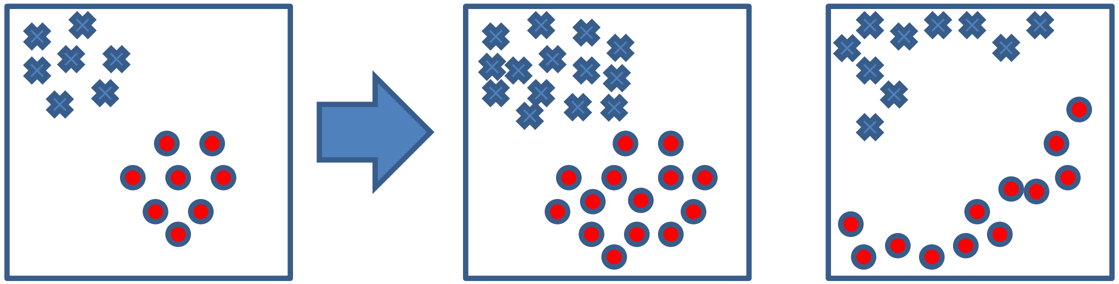
- Understanding domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing, etc.
- Learning models
- Interpreting results
- Consolidating and deploying discovered knowledge
- Loop

First, let there be data!



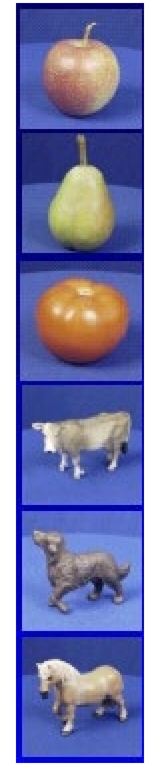
Training and Testing

- Training = the process of making the system able to learn
- Testing = the process of seeing how well the system learned
  - Simulates “real world” usage
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias
- Deployment = actually using the learned system in practice



Generalization

Training set (labels known)



Test set (labels unknown)

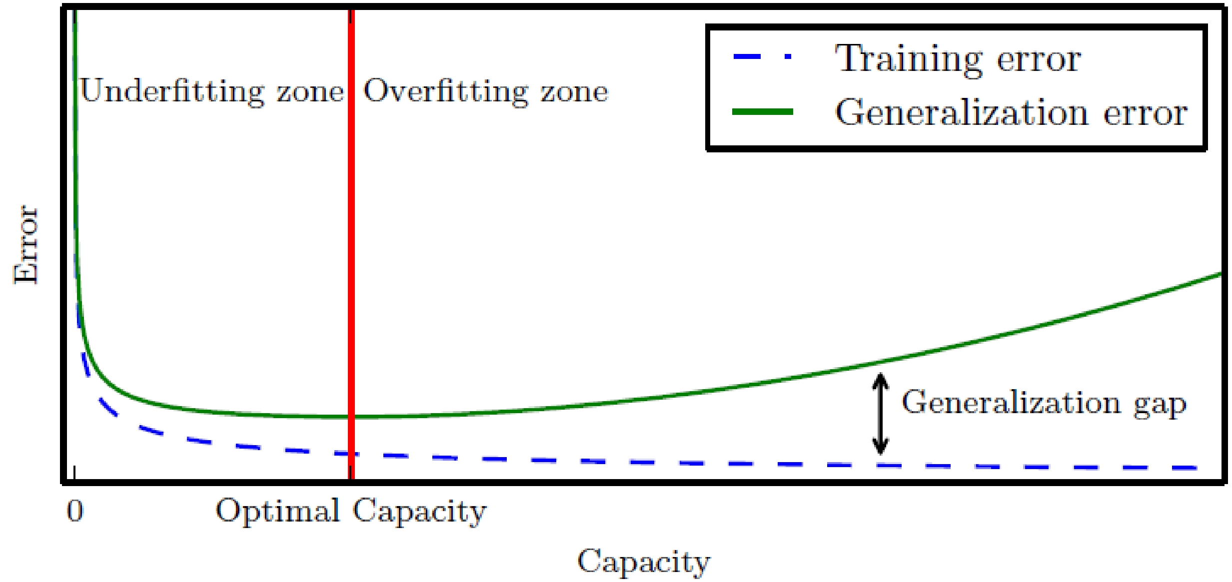
- How well does a learned model generalize from the data it was trained on to a new test set?

Slide credit: L. Lazebnik

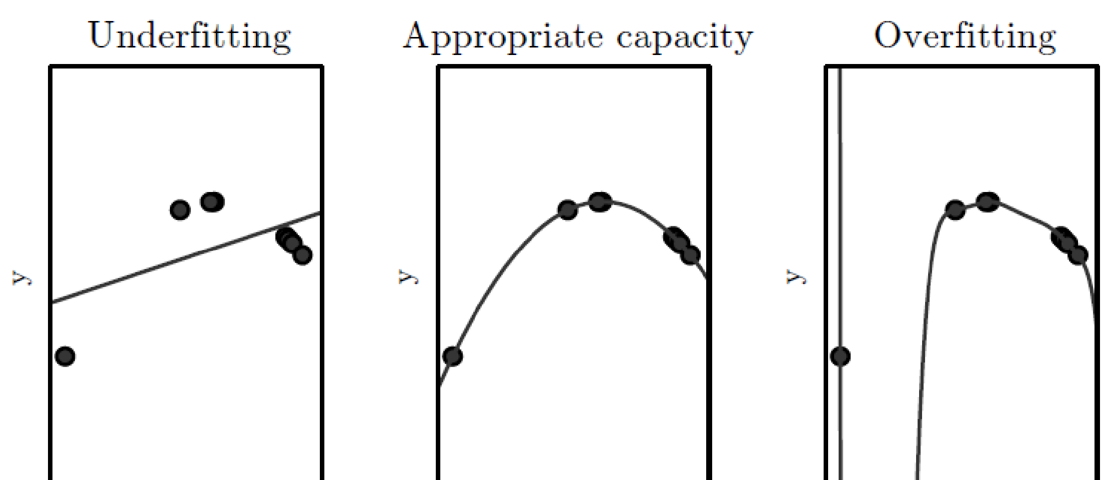
Generalization

- Components of generalization error
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - **Variance:** how much models estimated from different training sets differ from each other
- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Generalization and Capacity



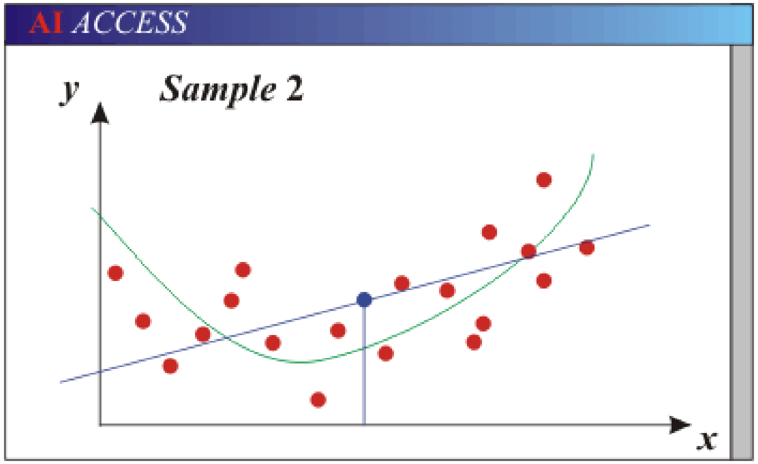
Underfitting and Overfitting in Polynomial Estimation

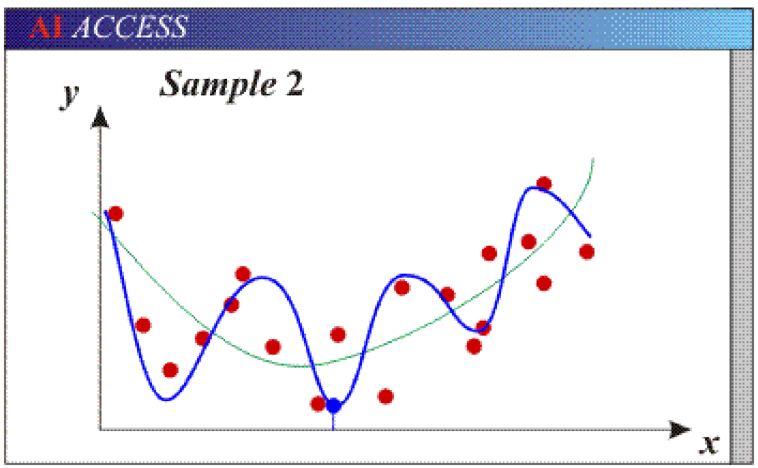


No Free Lunch Theorem

You can only get generalization through *assumptions*. *No one algorithm will solve all problems* (some will work better than others in some instances).

Bias-Variance Trade-off

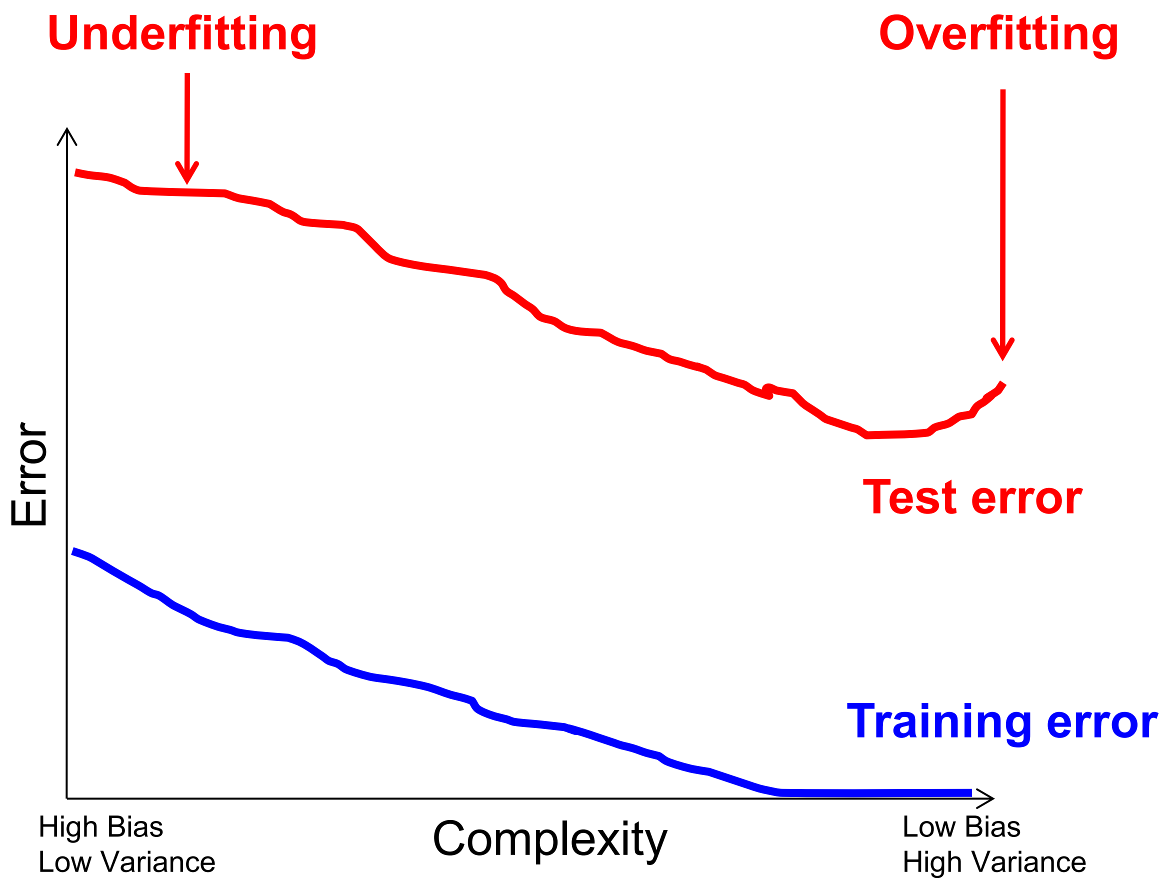




- Models with too few parameters are inaccurate because of a large bias (not enough flexibility)
- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample)

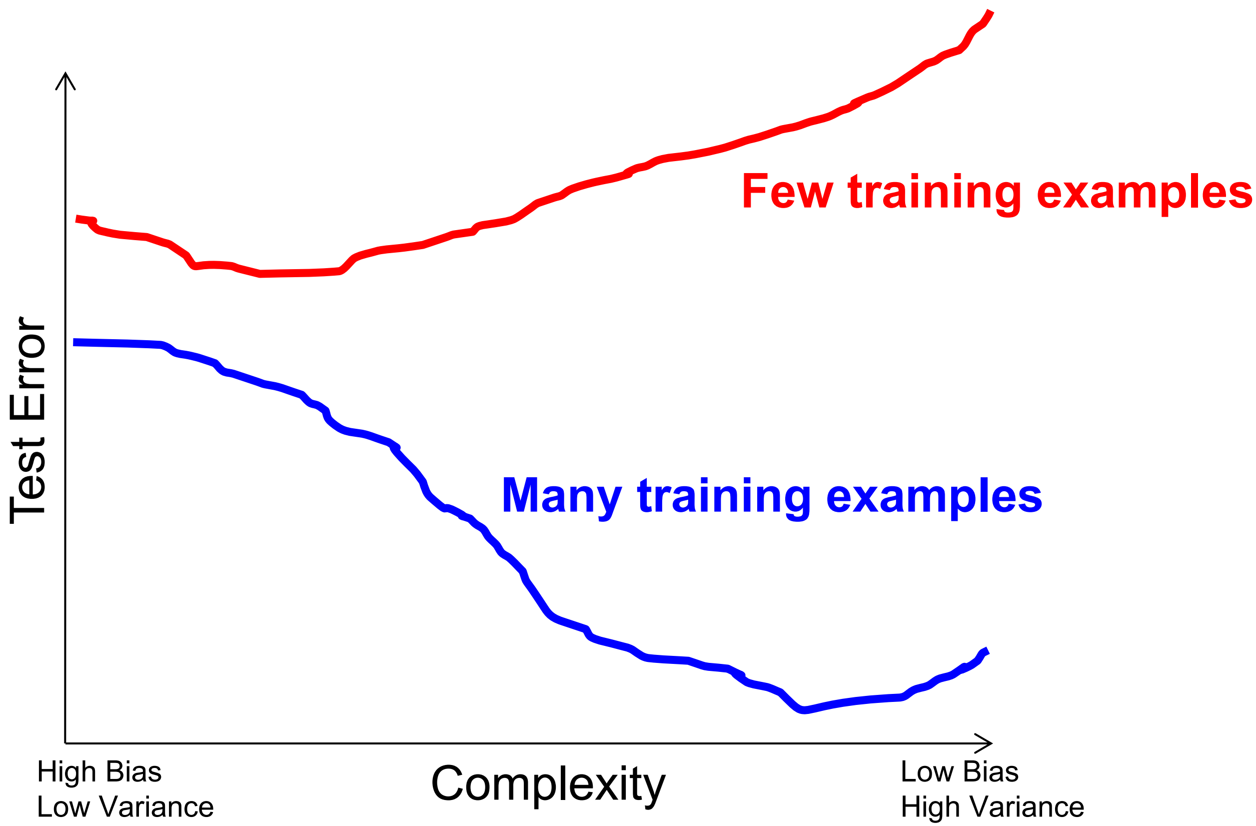
Slide credit: L. Lazebnik

Bias-Variance Trade-off: Underfitting & Overfitting



Slide credit: L. Lazebnik

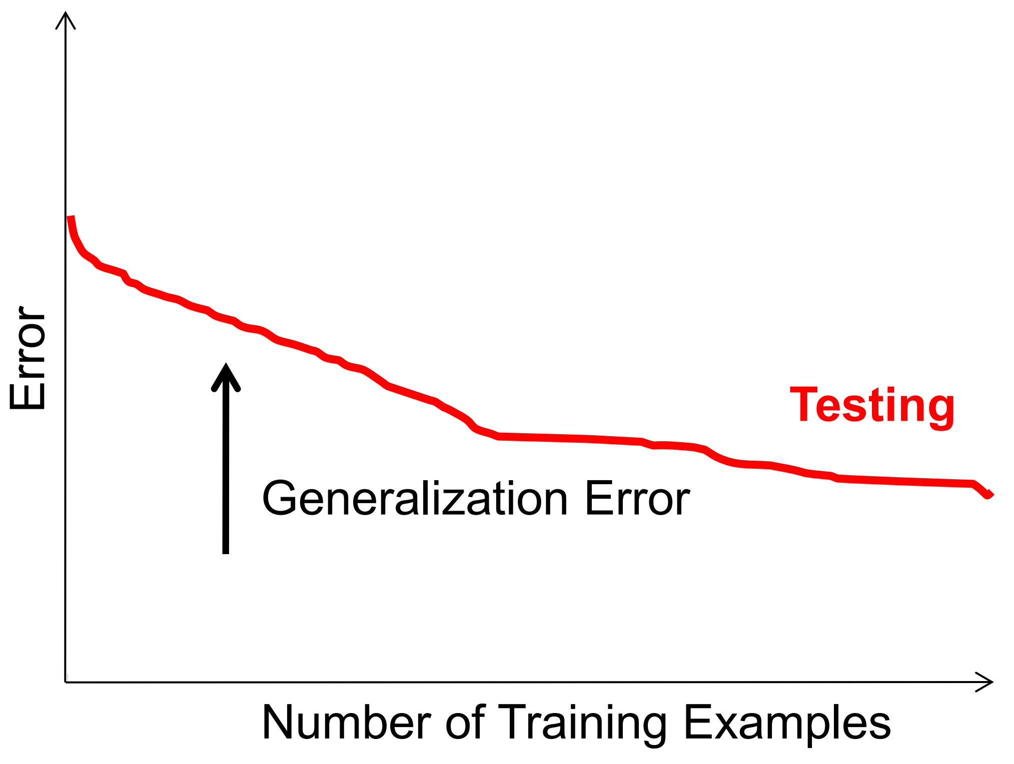
Bias-Variance Trade-off: Effect of Sample Size



Slide credit: L. Lazebnik

Effect of Training Size on Generalization Error

Fixed prediction model



Slide credit: L. Lazebnik

The “Perfect” Classification Algorithm

- Objective function: encodes the right loss for the problem
- Parameterization: makes assumptions that fit the problem
- Regularization: right level of regularization for amount of training data (consider bias-variance tradeoff)
- Training algorithm: can find parameters that maximize objective on training set
- Inference algorithm: can solve for the objective function in evaluation

Remember…

- No classifier is inherently better than any other: you need to make assumptions to generalize

- Three kinds of error
  - Inherent: unavoidable
  - Bias: due to over-simplifications
  - Variance: due to inability to perfectly estimate parameters from limited data

How to Reduce Variance?

- Choose a simpler classifier
  - Occam's Razor: *Among competing hypotheses, the one with the fewest assumptions should be selected.*
- Regularize the parameters
  - Think of L1/L2 penalties in regression
  - Think of Laplacian/Gaussian priors from a Bayesian probabilistic perspective
- Get more training data
  - ***BIG*** data

EXPERT OPINION

Contact Editor: **Brian Brannon**, bbrannon@computer.org

The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, ***Google***

Eugene Wigner's article "The Unreasonable Effectiveness of Mathematics in the Natural Sciences"[1] examines why so much of physics can be neatly explained with simple mathematical formulas such as $f = ma$ or $e = mc^2$. Meanwhile, sciences that involve human beings rather than elementary particles have proven more resistant to elegant mathematics. Economists suffer from physics envy over their inability to neatly model human behavior. An informal, incomplete grammar of the English language runs over 1,700 pages.[2] Perhaps when it comes to natural language processing and related fields, we're doomed to complex theories that will

behavior. So, this corpus could serve as the basis of a complete model for certain tasks—if only we knew how to extract the model from the data.

Learning from Text at Web Scale

The biggest successes in natural-language-related machine learning have been statistical speech recognition and statistical machine translation. The reason for these successes is not that these tasks are easier than other tasks; they are in fact much harder than tasks such as document classification that extract just a few bits of information from each document. The reason is that translation is a natural task routinely done every day for a real human need (think of the operations of the European Union or

Many Models / Classifiers!

- Supervised learning categories and techniques
  - **Linear classifier** (numerical functions)
  - **Parametric** (Probabilistic functions)
    - Naïve Bayes, Gaussian discriminant analysis (GDA), Hidden Markov models (HMM), Probabilistic graphical models
  - **Non-parametric** (Instance-based functions)
    - *K*-nearest neighbors, Kernel regression, Kernel density estimation, Local regression
  - **Non-metric** (Symbolic functions)
    - Classification and regression tree (CART), decision tree
  - **Aggregation**
    - Bagging (bootstrap + aggregation), Adaboost, Random forest
- Unsupervised learning categories and techniques
  - **Clustering**
    - K-means clustering / Spectral clustering
  - **Density Estimation**
    - Gaussian mixture model (GMM)
    - Graphical models
  - **Dimensionality reduction**
    - Principal component analysis (PCA)
    - Factor analysis

Generative vs. Discriminative Classifiers

Generative Models

- Represent both the data and the labels
- Often, makes use of conditional independence and priors
- Examples
  - Naïve Bayes classifier
  - Bayesian network

- Models of data may apply to future prediction problems

Discriminative Models

- Learn to directly predict the labels from the data
- Often, assume a simple boundary (e.g., linear)
- Examples
  - Logistic regression
  - SVM
  - Boosted decision trees

- Often easier to predict a label from the data than to model the data

Applications Abound

- Face detection
- Object detection and recognition
- Image segmentation
- Multimedia event detection
- Web search / information retrieval
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging
- *[Your favorite area]*

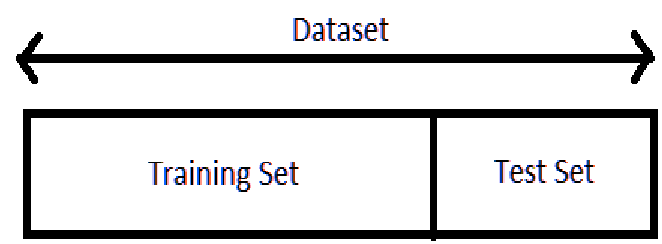
CROSS VALIDATION:

Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

The three steps involved in cross-validation are as follows :

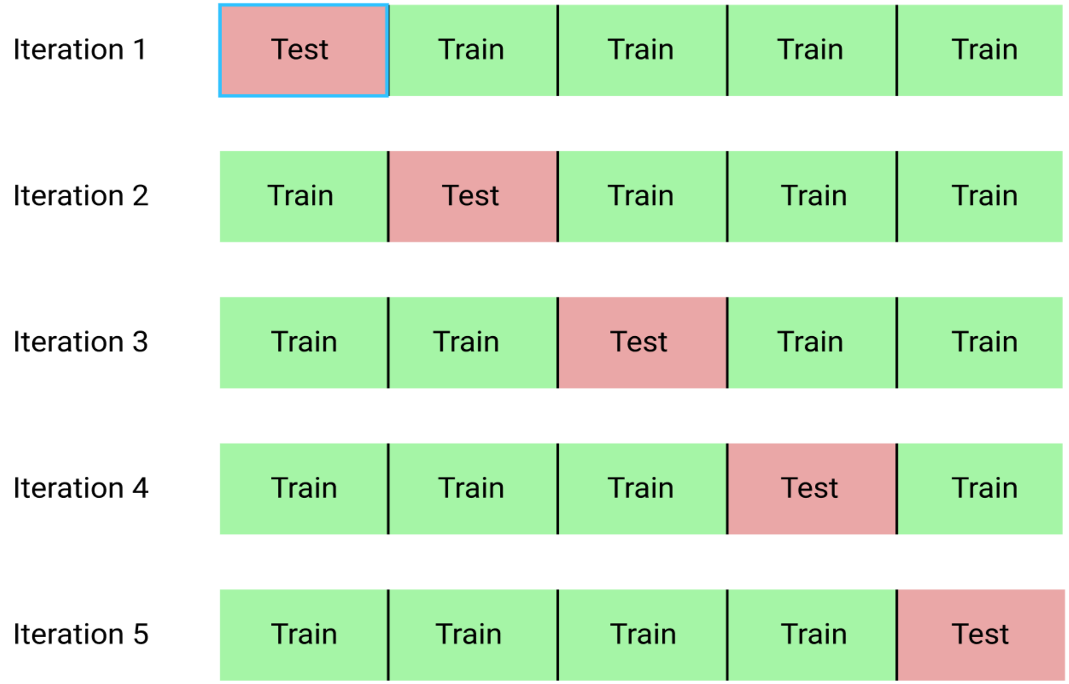
1. Split data set into training and test set
2. Using the training set train the model.
3. Test the model using the test set

USE: To get good out of sample accuracy



Even though we use cross validation technique we get variation in accuracy when we train our model for that we use K-fold cross validation technique

In **K-fold cross validation**, we split the data-set into k number of subsets(known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.



Questions?

References

- Slides/content were adapted from:
  - "Deep Learning" (Machine Learning Basics, Chapter 5, Goodfellow et al., 2016)
  - "An Overview of Machine Learning" (Yi-Fan Chang, 2011)
  - "CSE 446 Machine Learning" (intro) (Pedro Domingos)
  - "Feature learning for image classification" (Kai Yu and Andrew Ng)
  - "Machine Learning" (Computer Vision) James Hays, Brown
- Andrew Ng's Machine Learning course/lectures:
  - http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=MachineLearning
- Data Mining textbook : "Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)" Han et al. 2011

Some Machine Learning References

- General
  - Tom Mitchell, *Machine Learning*, McGraw Hill, 1997
  - Christopher Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995
- Adaboost (to learn about Boosting)
  - Friedman, Hastie, and Tibshirani, "Additive logistic regression: a statistical view of boosting", Annals of Statistics, 2000
- SVMs
  - http://www.support-vector.net/icml-tutorial.pdf

Slide credit: D. Hoiem