Evaluation & Validation

Credibility: Evaluating what has been learned

How predictive is a learned model?

- **How can we evaluate a model**
  - Test the model
  - Statistical tests
- **Considerations in evaluating a Model**
  - How was the model created, contributing factors
    - Amount of Data
    - “Quality” of data
      - – Labeled data is usually limited
  - How will model perform on unseen data?
    - Training/Validation/Test Data sets
    - Splitting Data

Evaluation

- **Significance tests - Statistical reliability of estimated differences in performance**
- **Performance measures**
  - Number of correct classifications
  - Accuracy of probability estimates
  - Error in numeric predictions
- **Different costs assigned to different types of errors**

Training and Testing

- **Error rate - performance measure for classification**
  - Success vs. Error
  - Instance's class is predicted correctly vs. instance's class is predicted incorrectly
  - Error rate
    - proportion of errors made over the whole set of instances
- **Re-substitution error**
  - error rate obtained from the training data

Training and Testing

- **Test set**
  - set of independent instances that have not been used in formation of classifier in any way
  - Assumption
    - both training data and test data are representative samples of the underlying problem
- **Example: classifiers built using customer data from two different towns A and B**
  - To estimate performance of classifier from town in completely new town, test it on data from B

Parameter tuning

- **Test data should NOT be used in any way to create the classifier!**
- **Some learning schemes operate in two stages**
  - Build the basic model
  - Optimize parameters
- **Test data can NOT be used for parameter tuning!**
- **Three sets**
  - training data, validation data, and test data

Evaluation

- **After evaluation is complete - all the data can be used to build the final classifier**
- **If we have lots of data - take a large sample and use for training - and another independent large sample for training**
  - provided both samples are representative
- **The larger the training data - the better the classifier**
- **The larger the test data - the more accurate the error estimate**

Evaluation

- **In many situations the training data must be classified manually - and so must the test data**
- **How to make most of limited data set?**
- **Holdout procedure**
  - splitting original data into training and test set
- **We want both-large training and a large test set**

Holdout Estimation Method

- **What if the amount of data is limited?**
- **Reserve a certain amount for testing and uses the remainder for training**
  - 1/3 for testing, the rest for training
- **The samples might not be representative**
  - Class might be missing in the test data
- **Stratification**
  - Ensures that each class is represented with approximately equal proportions in both subsets

Repeated Holdout Method

- **More reliable estimates made by repeating the process with different sub-samples**
- **In each iteration**
  - a certain proportion is randomly selected for training (stratified)
- **The error rates on the different iterations are averaged to form an overall error rate**
- **Still not optimum: the different test set overlap**
- **Can we prevent overlapping?**

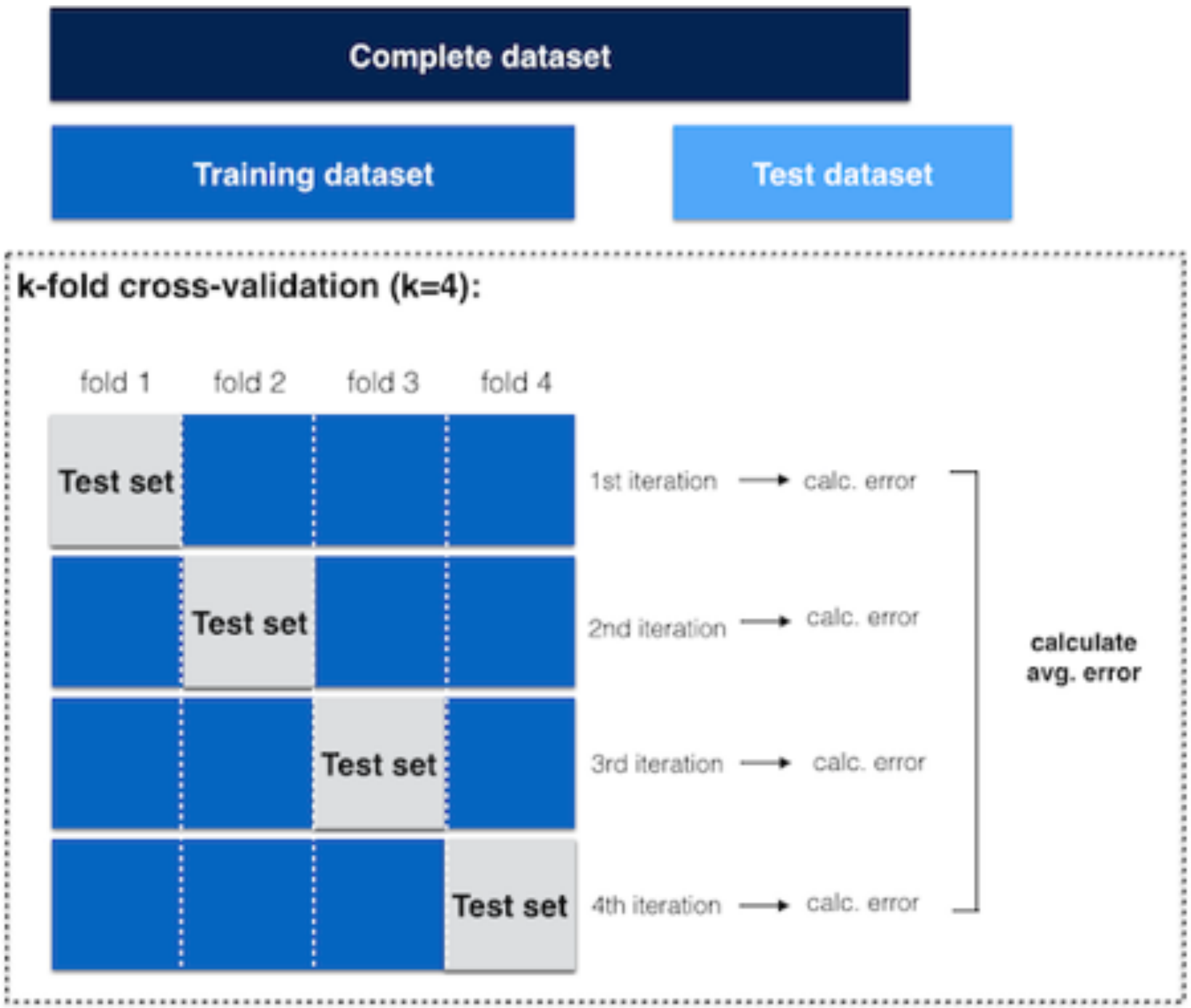
Cross-validation

- **Avoids overlapping test sets**
- **K-fold cross-validation:**
  - First step
    - data is split into k subsets of equal size
  - Second step
    - each subset in turn is used for testing and the remainder for training
- **Subsets are stratified before the cross-validation is performed**
- **The error estimates are averaged to form an overall error estimate**

Cross-validation

- **Standard, very popular method for evaluation**
  - stratified ten-fold cross-validation
- **Why 10?**
  - Extensive experiments
  - Some theoretical evidence
- **Stratification reduces the estimate's variance**
- **Repeated stratified cross-validation**
  - Ten-fold cross-validation is repeated ten times and results are averaged

Example of a 4-fold Cross-validation



Leave-one-out Method

- **Makes maximum use of the data**
  - The number of folds = number of training instances
  - I.e., a classifier has to be built n times
    - n is the number of training instances
- **Judged by the correctness of the remaining instance 0 or 1**
- **No random sub-sampling, no repeating**
- **Very computationally expensive!**

The Bootstrap Method

- **CV uses sampling WITHOUT replacement**
  - The same instance, once selected, can NOT be selected again for a particular training/test set
- **Estimation method that uses sampling WITH replacement**
  - A data set of n instances is sampled n times with replacement to form a new data set of n instances
  - This data is used as the training set
  - The instances from the original data set that do not occur in the new training set are used for testing

The 0.632 Bootstrap

- **Where does the name come from?**
  - A particular instance has a probability of 1-1/n of not being picked
  - Thus its probability of ending up in the test data is:

$$\left(1-\frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

  - Training data will contain approximately 63.2% of the instances

Estimating Error With The Bootstrap

- **A very pessimistic error estimate on the test data**
- **It contains only ~63% of the instances**
- **Combined with the re-substitution error**
  - The re-substitution error gets less weight than the error on the test data
- **Process is repeated several times**
  - with different replacement samples
  - the results averaged

Comparing the Learning Methods

- **Which one of learning schemes performs better?**
- **Domain dependent!**
- **Compare 10-fold CV estimates?**
- **Problem - variance in estimate**
  - Can be reduced using repeated CV
- **Are the results reliable?**

Significance Tests

- **How confident we can be that there really is a difference**
- **Example: 10 times 10-fold CV**
  - Sample from a distribution of a CV experiment
  - Use different random partitions of the data set
  - Treat calculated error rates as different independent samples from a probability distribution
  - Are the two means of the 10 CV estimates significantly different?

The T-test

- **Student's t-test**
  - Are the means of two samples significantly different?
- **The individual samples are taken from the set of all possible cross-validation estimates**
- **We can use a paired t-test because the individual samples are paired**
- **Student's distribution with k-1 degrees of freedom**
  - Use table of confidence intervals for Student distribution (not normal distribution)

Interpreting Results

- All cross-validation estimates are based on the same data set
- The test only tells if a complete k-fold CV for this data set would show a difference
- Complete k-fold CV generates all possible partitions of the data into k folds and averages the results
- A different data set sample for each of the k-fold CV estimates used in the test to judge performance across different training sets would be ideal!

Calculating the cost

- **There many other types of costs!**
  - Cost of collecting training data
  - Cost of cleaning data
  - Cost of classification errors
- **Different types of classification errors incur different costs**
  - Examples:
    - Loan decisions
    - Oil-slick detection
    - Fault diagnosis
    - Promotional mailing
    - Cancerous Cells

Counting the Cost

- **The confusion matrix:**

Predicted Class (expectation)			
		Yes	***No***
Actual Class (observation)	***Yes***	***True Positive*** *(Correct Result)*	***False Negative*** *(Missing Result)*
	No	***False Positive*** *(Unexpected Result)*	***True Negative*** *(correct absence of*

Counting the Cost

- **Evaluation Methods**
  - Overall Success Rate == (TP + TN) / (TP+TN+FP+FN)
    - Number of correct classifications divided by the total number of classifications
  - Error rate = 1- overall success rate
  - True Positive Rate (or recall or sensitivity) = TP /(TP + FN)
    - True positives, divided by the total number of positives
  - False Positive Rate = FP /(FP + TN) .
    - False Positives divided by the total number of negatives
  - Precision = TP / (TP+FP) == Positive predictive value
  - F-measure = 2 * ((precision * recall)/(precision + recall))
    - combines precision and recall using harmonic mean.

Lift charts

- **Costs are rarely known**
- **Decisions made by comparing possible scenarios**
- **Example - Promotional mail-out**
  - Situation 1: classifier predicts that 0.1% of all households will respond
  - Situation 2: classifier predicts that 0.4% of the 10000 most promising households will respond
- **A lift chart allows for a visual comparison**

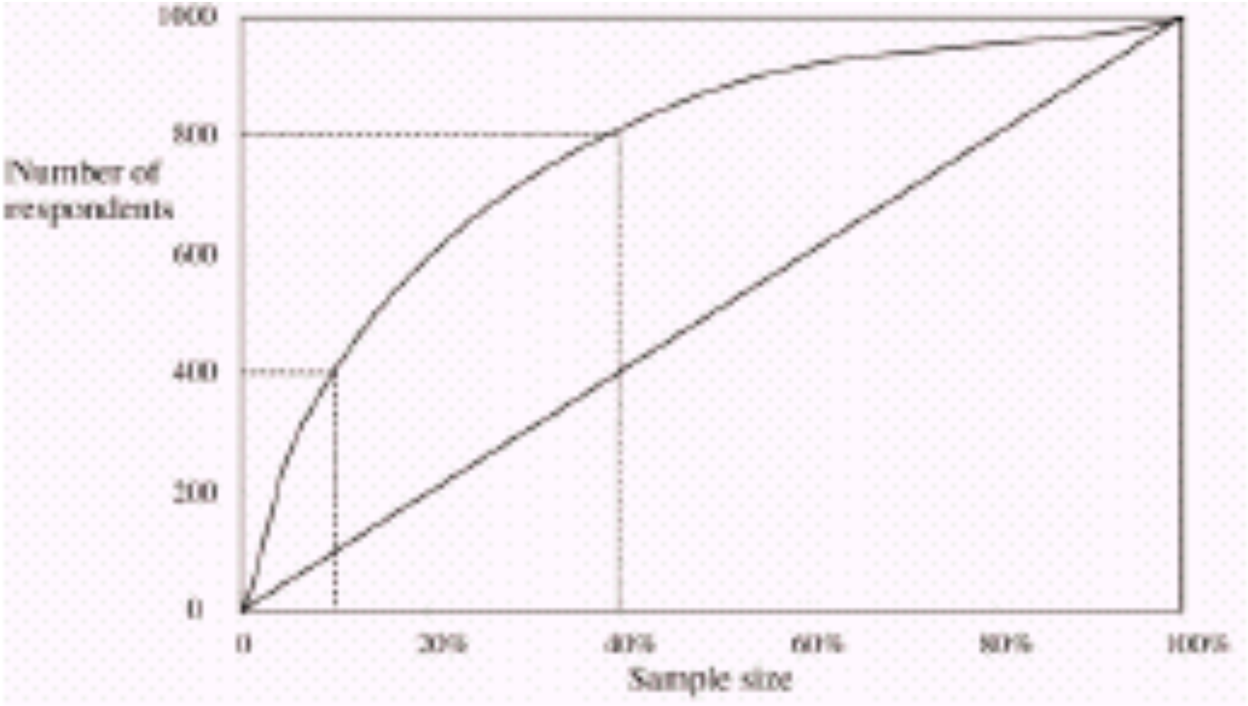
Generating a lift chart

- **Instances are sorted by their predicted probability of being a true positive:**

Rank	Predicted Probability	Actual Class
1	*0.95*	*Yes*
2	*0.93*	*Yes*
3	*0.93*	*No*
4	*0.88*	*Yes*
…	*…*	*…*

- **In lift chart, x axis is sample size and y axis is number of true positives**

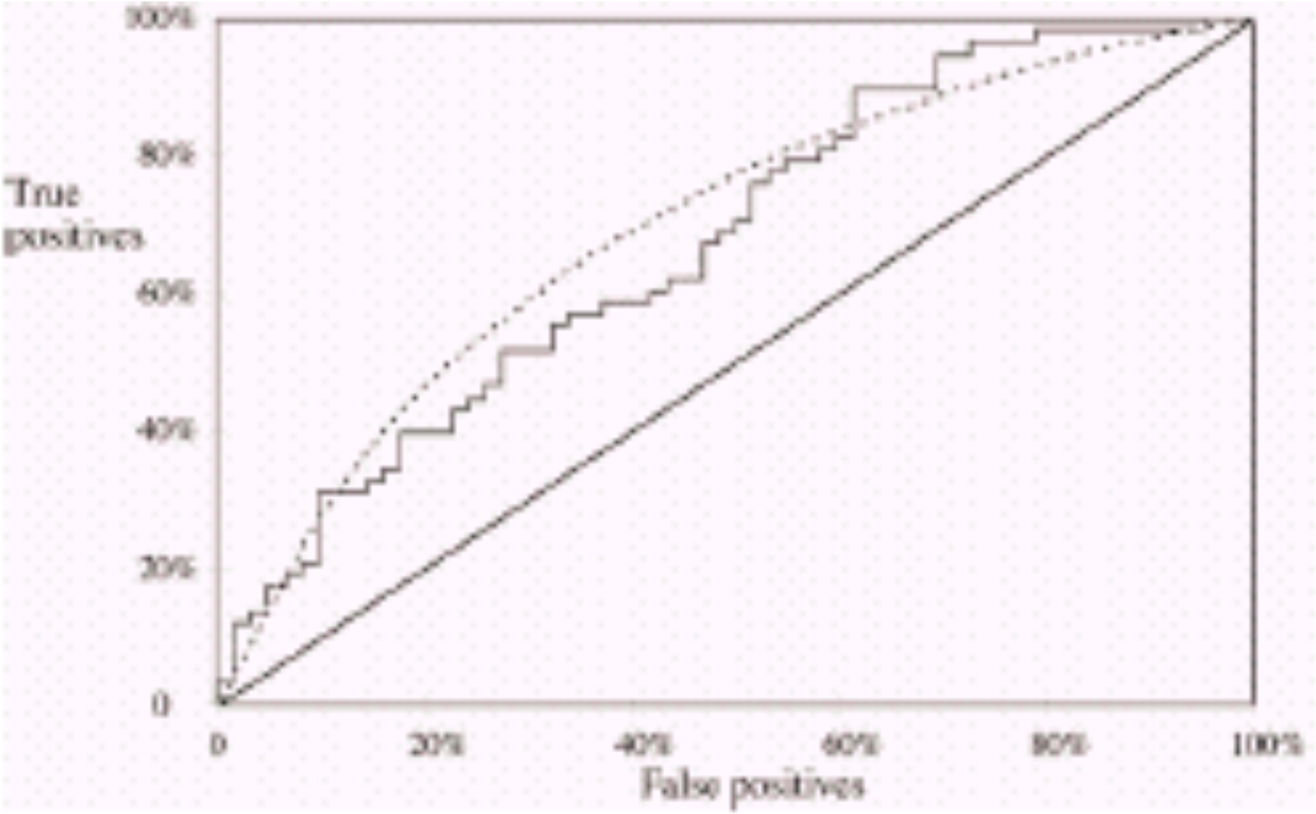
Lift Chart Example



ROC curves

- **Similar to lift charts**
  - “ROC” stands for “receiver operation characteristic”
  - Used in signal detection to show tradeoff between hit rate and false alarm rate over noisy channel
- **ROC curves vs. lift chart:**
  - y axis shows percentage of true positives in sample (rather than absolute number)
  - x axis shows percentage of false positives in sample (rather than sample size)

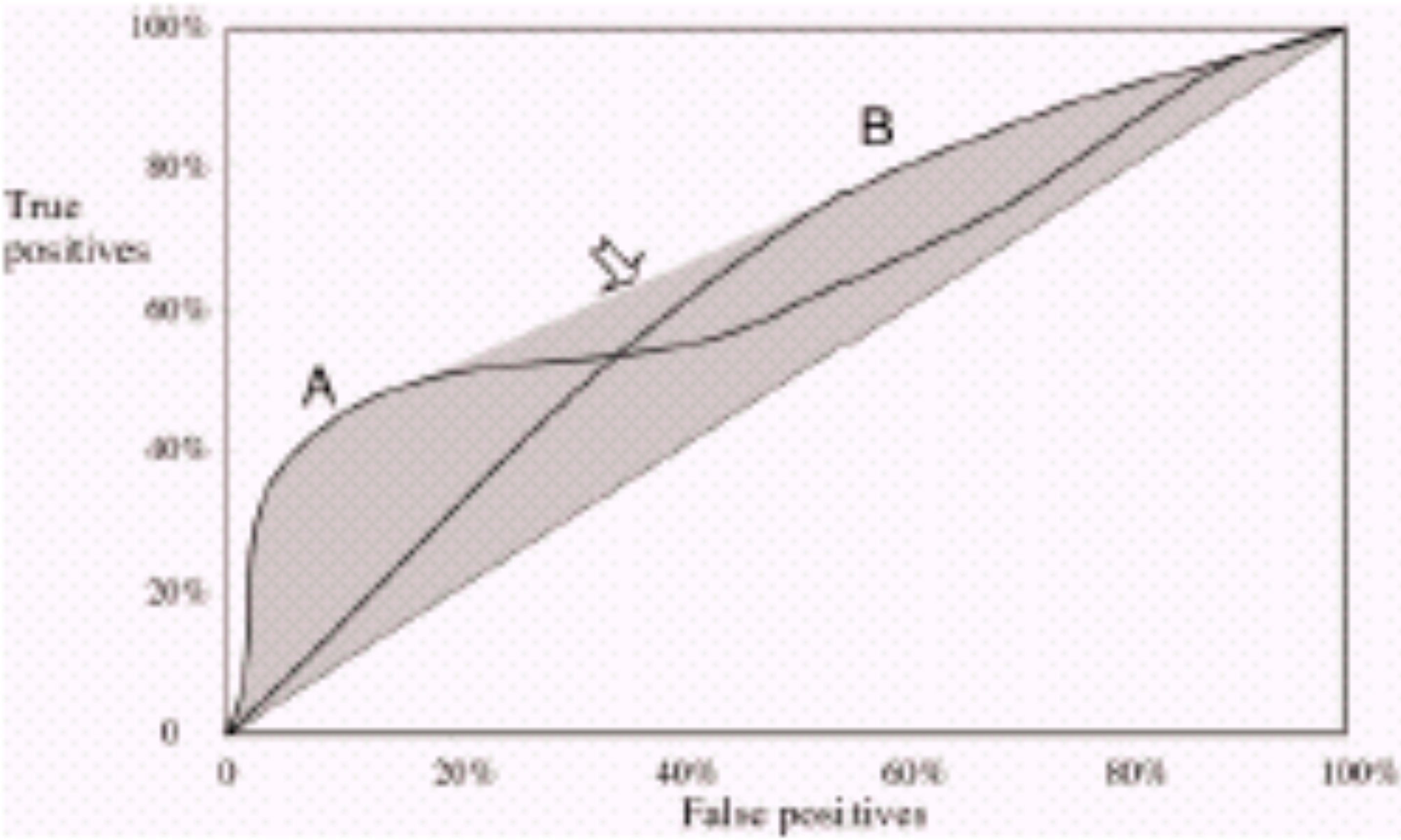
A sample ROC curve



ROC curves and Cross-validation

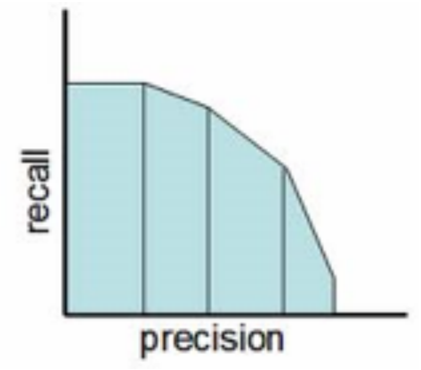
- **Simple method of getting a ROC curve using cross-validation:**
  - Collect probabilities for instances in test folds
  - Sort instances according to probabilities
- **Method implemented in WEKA**
  - Generates an ROC curve for each fold and averages them

ROC curves for two schemes



Recall-Precision Curves

- Web search engine example:
- Percentage of retrieved documents that are relevant: Precision=TP/TP+FP
- Percentage of relevant documents that are returned: Recall =TP/TP+FN
- Precision/recall curves have hyperbolic shape



Summary of Measures

	Domain	Plot	Explanation
Lift Chart	*Marketing*	*TP vs.* *Subset Size*	*# true positives* *TP(TP+FP)/(TP+FP*
ROC Curve	*Communications*	*TP Rate vs.* *FP Rate*	*tp = (TP/(TP+RN)) *100* *fp = (FP/(FP*
Recall-Precision Curve	*Information Retrieval*	*Recall vs.* *Precision*	*Recall = TP Rate* *(TP/(TP+FP))*100*

Single Measure to Characterize Performance

- **Two that are used in information retrieval are:**
  - Average recall
    - 3-point average recall
      - takes the average precision at recall values of 20%, 50% and 80%
    - 11-point average recall:
      - takes the average precision at recall values of 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%
  - F-Measure = (2 x recall x precision)/(recall+precision)
- **Success Rate**
  - Success Rate: TP+TN/(TP+FP+TN+FN)

Evaluating Numeric Prediction

- **Strategies used to evaluate are the same**
  - independent test set, cross-validation, significance tests, etc.
- **How they are evaluated will be different**
  - errors are not present/absent they come in different sizes

- **Actual target values: a1, a2,…, an**
- **Predicted target values: p1, p2,…, pn**

Numeric Prediction Performance Measures

- Mean-squared error $\frac{(p_1-a_1)^2+\cdots+(p_n-a_n)^2}{n}$
- The root mean-squared error

$$\sqrt{\frac{(p_1-a_1)^2+\ \ldots+\ (p_n-a_n)^2}{n}}$$

- The mean absolute error

$$\frac{|p_1-a_1|+\ \ldots+|p_n-a_n|}{n}$$

  - is less sensitive to outliers then mean-squared error

Numeric Prediction Performance Measures

- **Sometimes relative error values are more appropriate**
  - Relative Squared Error
  - Root Relative Squared Error
  - Relative absolute error
- **Correlation Coefficients**
  - Measures statistical coefficients between actual and predicted values.

Evaluation in Scikit

http://scikit-learn.org/stable/modules/cross_validation.html#cross-validation-and-model-selection

Bias-Variance Trade-off

Bias and Variance

- **No "best classifier" in general**
  - Necessity for exploring a variety of methods
- **How to evaluate if the learning algorithm "matches" the classification problem**
- **Bias: measures the quality of the match**
  - High-bias implies *poor* match
- **Variance: measures the specificity of the match**
  - High-variance implies a *weak* match
- **Bias and variance are not independent of each other**

Bias Variance Dilemma

- **Procedures with increased flexibility to adapt to training data have lower bias, but higher variance**
  - Large number of parameters
  - Fits well and have low bias, but high variance
- **Inflexible procedures have higher bias, but lower variance**
  - Fewer number of parameters
  - May not fit well to data: have high bias, but low variance
- **A large amount of training data generally helps improve performance of estimation if the model is sufficiently general to represent the target function**

Model Loss (Error)

- **Squared loss of model on test case i:**

$$\left(Learn(x_i, D) - Truth(x_i)\right)^2$$

- **Expected prediction error:**

$$\left\langle \left(Learn(x, D) - Truth(x)\right)^2 \right\rangle_D$$

Bias/Variance Decomposition

$$\left\langle \left(L(x,D) - T(x)\right)^2 \right\rangle_D = Noise^2 + Bias^2 + Variance$$

$Noise^2$ = lower bound on performance

$Bias^2 = \left(\text{expected error due to model mismatch}\right)^2$

$Variance$ = variation due to train sample and randomization

Bias^2

- Low bias
  - linear regression applied to linear data
  - 2nd degree polynomial applied to quadratic data
  - ANN with many hidden units trained to completion
- High bias
  - constant function
  - linear regression applied to non-linear data
  - ANN with few hidden units applied to non-linear data

Variance

- Low variance
  - constant function
  - model independent of training data
  - model depends on stable measures of data
    - mean
    - median
- High variance
  - high degree polynomial
  - ANN with many hidden units trained to completion

Sources of Variance in Supervised Learning

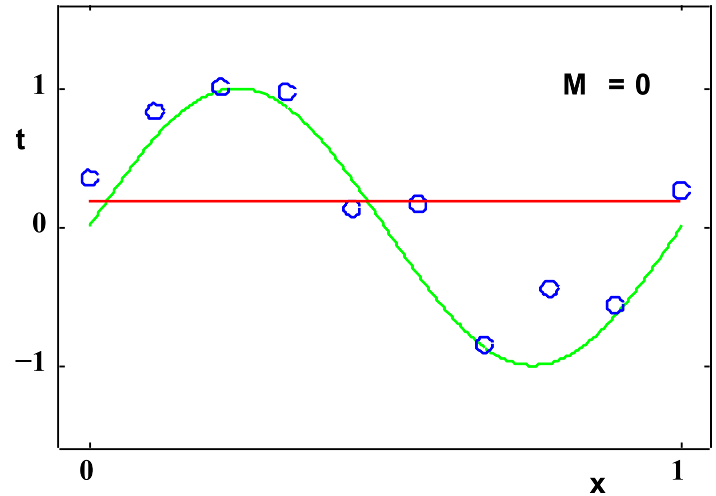
- noise in targets or input attributes
- bias (model mismatch)
- training sample
- randomness in learning algorithm
  - neural net weight initialization
- randomized subsetting of train set:
  - cross validation, train and early stopping set

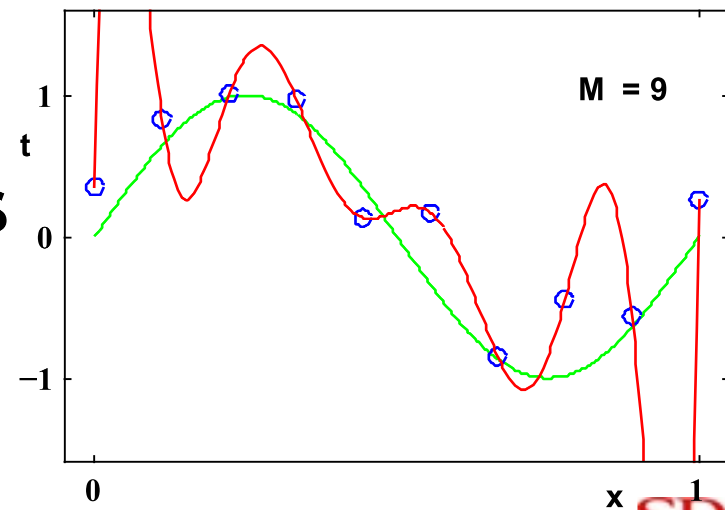
Bias/Variance Tradeoff

- (bias^2+variance) is what counts for prediction
- Often:
  - low bias => high variance
  - low variance => high bias
- Tradeoff:
  - bias^2 vs. variance

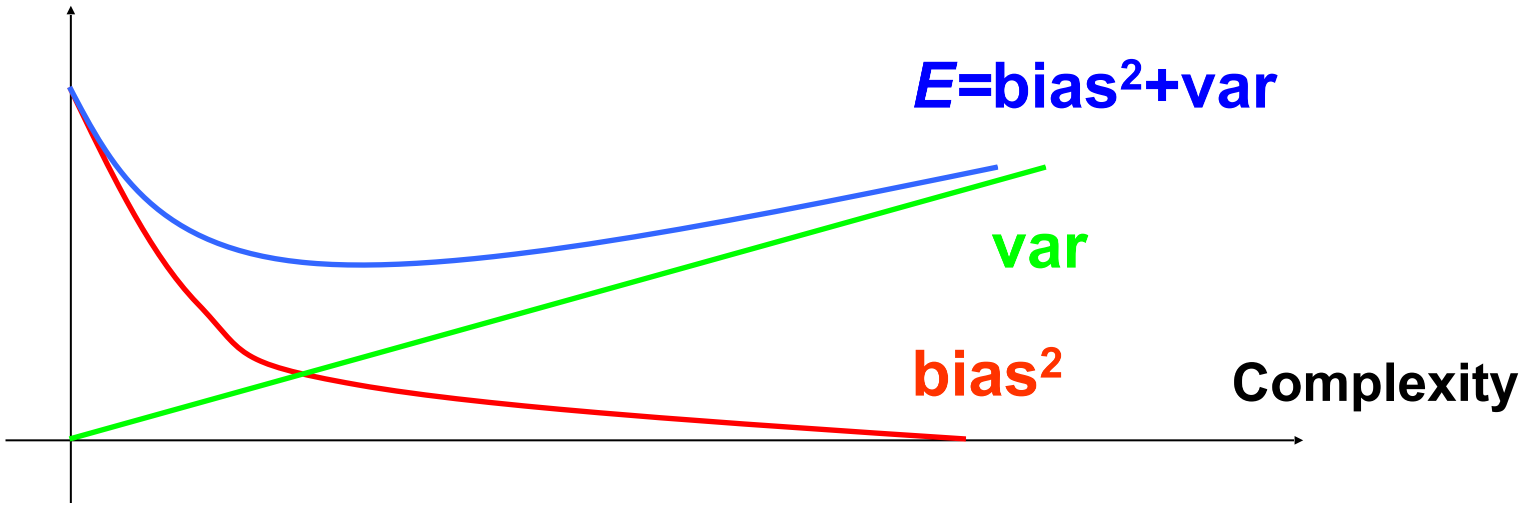
Bias-Variance Trade-off

- **Model too simple:** does not fit the data well
  - A *biased* soluJon
- **Model too complex:** small changes to the data, soluJ on changes a lot
  - A *high-variance* soluJon





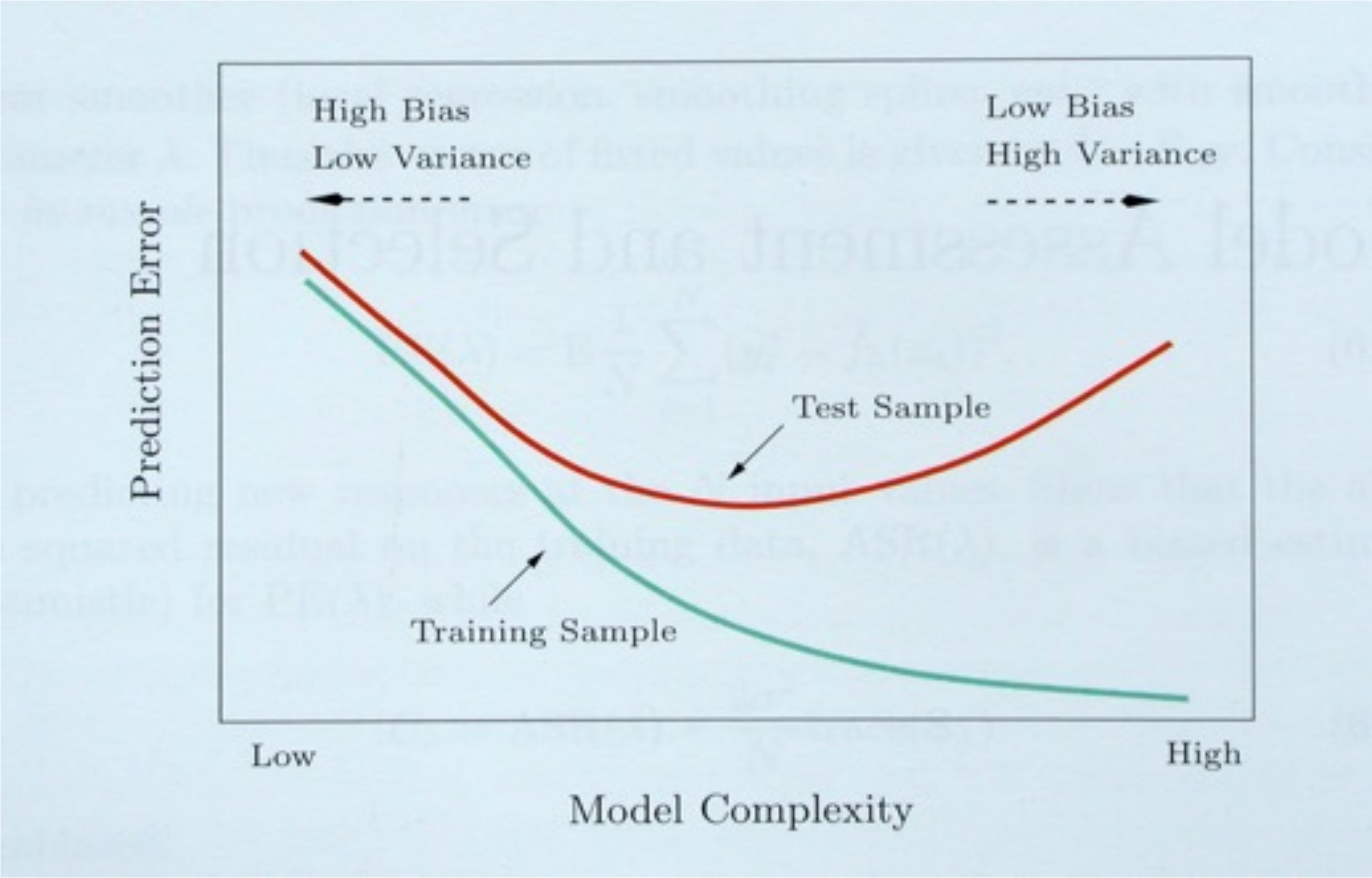
Complexity of the model



Typically the bias is a decreasing function of the complexity

while variance is an increasing function of the complexity

Bias/Variance Tradeoff



Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001

Reduce Variance Without Increasing Bias

- **Averaging reduces variance:**

$$Var(\bar{X}) = \frac{Var(X)}{N}$$

- **Average models to reduce model variance**
- **One problem:**
  - **only one train set**
  - **where do multiple models come from?**

Bias Variance Dilemma

- **Bias/Variance considerations recommend that we gather as much prior information about the problem as possible to find a best match for the classifier, and as large a dataset as possible to reduce the variance**
- **We can virtually never get zero bias and zero variance**

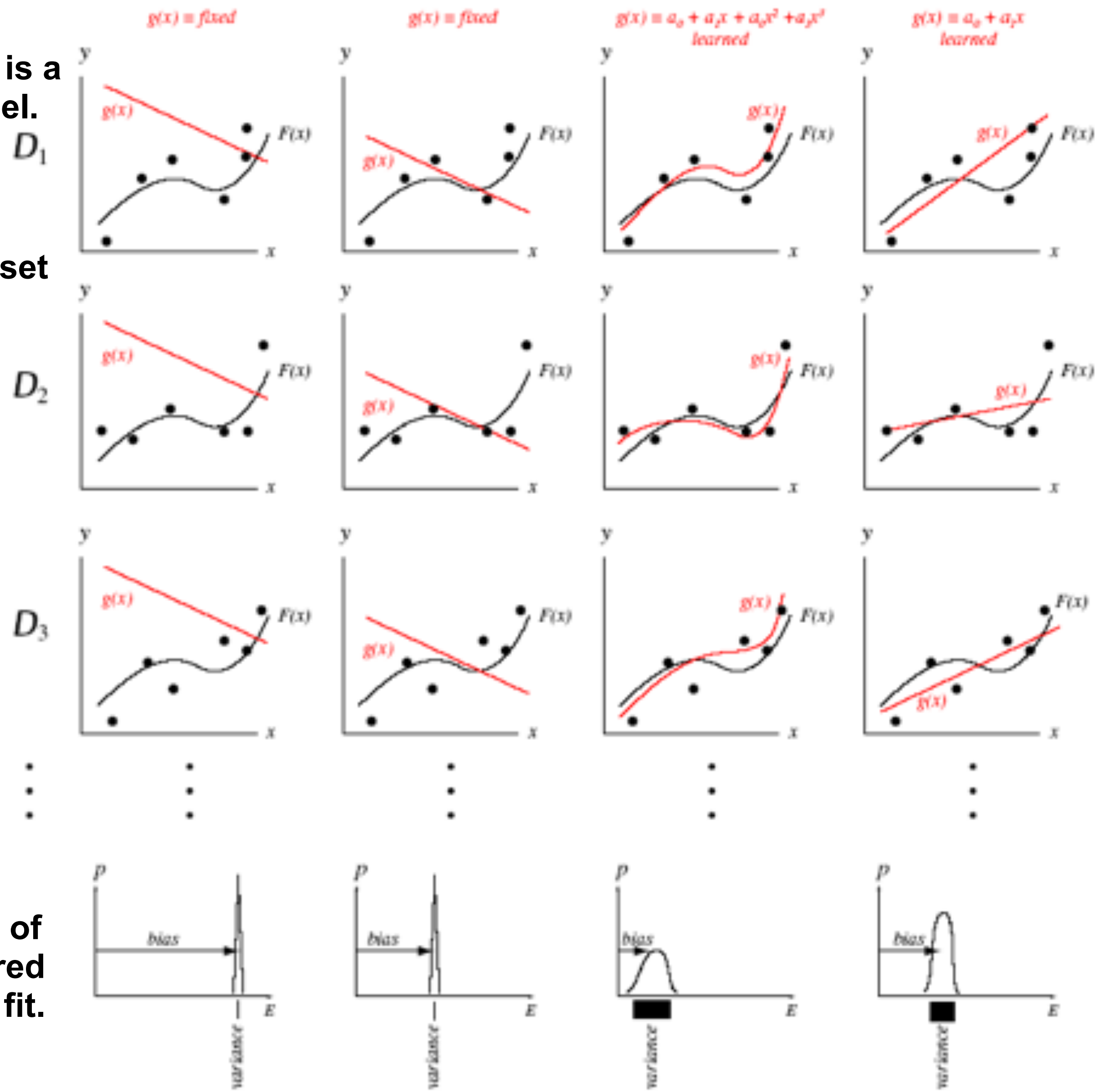
Questions?

Questions?

Bias-Variance Dilemma Example

Each column is a different model.

Each row is a different dataset of 6 points.



Histograms of mean-squared error of the fit.

Col 1: Poor fixed linear model; High bias, zero variance

Col 2: Slightly better fixed linear model; Lower (but high) bias, zero variance.

Col 3: Learned cubic model; Low bias, moderate variance.

Col 4: Learned linear model; Intermediate bias and variance.