Terminology and Evaluating Hypotheses

- Statistics
  - Basic terms
  - Sample error, true error
  - Distributions
  - Cost/utility
  - Tests for significance
- Comparing Learning Methods

Basic Statistics Terms

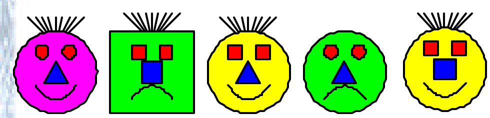
- Sample mean – average of a sample of numbers

  $\text{mean} = \bar{y} = \frac{1}{n}(y_1 + y_2 + y_3 + \ldots + y_n)$
- Sample median – middle (in sorted order) of a sample of numbers
- Sample mode – sample value appearing most frequently
- $\text{variance} = \text{var}_y = s_y = \frac{(y_1-\bar{y})^2+(y_2-\bar{y})^2+(y_3-\bar{y})^2+\ldots+(y_n-\bar{y})^2}{n-1}$
- $\text{standard deviation} = s_y = \sqrt{s_y^2}$

Data Sets

- Data set – set of examples of a problem
- Feature (attribute,field,variable) – one value that defines an instance
  - Categorical (nominal) with a set of possible values versus continuous (qualitative) – numeric range of possible values
  - Input feature (independent variable) versus output feature (dependent variable)
  - Can be missing (value not known)
- Example (instance, case, record, feature vector, tuple) – the values of the input (and in some cases output) features of variables
- Skewed data set – one class occurs far more than others
- Multi-class problem – more than 2 output values
- Regression problem – output value is continuous

Data Set Concepts



Eyes	Nose	Head	Fcolor	Hair?	Smile?
Round	Triangle	Round	Purple	Yes	Yes
Square	Square	Square	Green	Yes	No
Square	Triangle	Round	Yellow	Yes	Yes
Round	Triangle	Round	Green	No	No
Square	Square	Round	Yellow	Yes	Yes

Data Sets (continued)

- Training data set – the set of data used to learn (create) a model of a problem
- Test data set – the set of data used to estimate some value (often accuracy) related to a model
- Validation set – a set of data used to select parameters for a model, often as follows
  - Divide training data into a “sub” training set and validation set
  - For each possible set of parameters
    - Create a model using the “sub” training set
    - Evaluate the model on the validation set and pick the one that performs the best

Evaluating Models

- Need a measure of value – the cost (loss, utility) of a model
- Often use accuracy (or error)
  - Accuracy – how many examples we get “right”
  - Error – how many examples we get wrong
- Can be weighted
  - If examples are not equal, could count the cost (or utility) of mispredicted (correct) examples

Confusion Matrix

		Predicted		
		Positive	Negative	Total
Actual	Positive	True Positive (TP)	False Negative (FN)	#Positives
	Negative	False Positive (FP)	True Negative (TN)	#Negatives
	Total	TP+FP	FN+TN	#Examples

- Accuracy = (TP+TN) / #Examples
- Error = (FP+FN) / #Examples
- Recall (sensitivity, true positive rate) = TP / #Positives
- Precision = TP / (FP+TP)
- True Negative Rate (specificity) = TN / #Negatives
- False Positive Rate = FP / (FP+TP)
- False Negative Rate = FN / #Negatives

Confusion Matrix – Multi Class

- For many problems (especially multiclass problems), often useful to examine the sources of error
- Confusion matrix:

		Predicted			
		ClassA	ClassB	ClassC	Total
Expected	ClassA	25	5	20	50
	ClassB	0	45	5	50
	ClassC	25	0	25	50
	Total	50	50	50	150

Results Analysis: Confusion Matrix

- Building a confusion matrix
  - Zero all entries
  - For each data point add one in row corresponding to actual class of problem under column corresponding to predicted class
- Perfect prediction has all values down the diagonal
- Off diagonal entries can often tell us about what is being mis-predicted

Problems Estimating Error

1. *Bias*: If S is training set, $error_S(h)$ is optimistically biased

$$bias \equiv E[error_S(h)] - error_D(h)$$

For unbiased estimate, h and S must be chosen independently

2. *Variance*: Even with unbiased S, $error_S(h)$ may still vary from $error_D(h)$

Two Definitions of Error

The true error of hypothesis h with respect to target function f and distribution D is the probability that h will misclassify an instance drawn at random according to D.

$$error_D(h) \equiv \Pr_{x \in D}[f(x) \neq h(x)]$$

The sample error of h with respect to target function f and data sample S is the proportion of examples h misclassifies

$$error_S(h) \equiv \frac{1}{n}\sum_{x \in S} \delta(f(x) \neq h(x))$$

where $\delta(f(x) \neq h(x))$ is 1 if $f(x) \neq h(x)$, and 0 otherwise

How well does $error_S(h)$ estimate $error_D(h)$?

Example

Hypothesis h misclassifies 12 of 40 examples in S.

$$error_S(h) = \frac{12}{40} = .30$$

What is $error_D(h)$?

Estimators

Experiment:

1. Choose sample S of size n according to distribution D
2. Measure $error_S(h)$

$error_S(h)$ is a random variable (i.e., result of an experiment)

$error_S(h)$ is an unbiased estimator for $error_D(h)$

Given observed $error_S(h)$ what can we conclude about $error_D(h)$?

Confidence Intervals

If

- S contains n examples, drawn independently of h and each other
- $n \geq 30$

Then

- With approximately N% probability, $error_D(h)$ lies in interval

$$error_S(h) \pm z_N \sqrt{\frac{error_S(h)(1-error_S(h))}{n}}$$

where

N%:	50%	68%	80%	90%	95%	98%	99%
z_N :	0.67	1.00	1.28	1.64	1.96	2.33	2.53

Confidence Intervals

If

- S contains n examples, drawn independently of h and each other
- $n \geq 30$

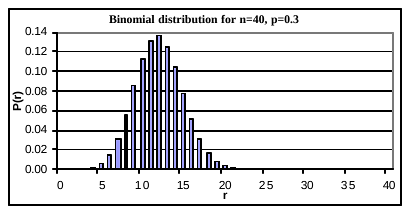
Then

- With approximately 95% probability, $error_D(h)$ lies in interval

$$error_S(h) \pm \mathbf{1.96} \sqrt{\frac{error_S(h)(1-error_S(h))}{n}}$$

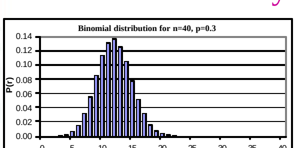
$error_S(h)$ is a Random Variable

- Rerun experiment with different randomly drawn S (size n)
- Probability of observing r misclassified examples:



$$P(r) = \frac{n!}{r!(n-r)!} error_D(h)^r (1-error_D(h))^{n-r}$$

Binomial Probability Distribution

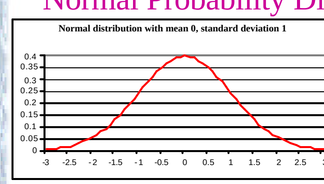


$$P(r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r}$$

Probability $P(r)$ of r heads in n coin flips, if $p = \Pr(heads)$

- Expected, or mean value of X : $E[X] \equiv \sum_{i=0}^{n} iP(i) = np$
- Variance of X : $Var(X) \equiv E[(X - E[X])^2] = np(1-p)$
- Standard deviation of X : $\sigma_X \equiv \sqrt{E[(X - E[X])^2]} = \sqrt{np(1-p)}$

Normal Probability Distribution



$$P(r) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

The probability that X will fall into the interval (a,b) is given by

$$\int_a^b p(x)\,dx$$

- Expected, or mean value of X : $E[X] = \mu$
- Variance of X : $Var(X) = \sigma^2$
- Standard deviation of X : $\sigma_X = \sigma$

Normal Distribution Approximates Binomial

$error_S(h)$ follows a Binomial distribution, with

- mean $\mu_{error_S(h)} = error_D(h)$
- standard deviation

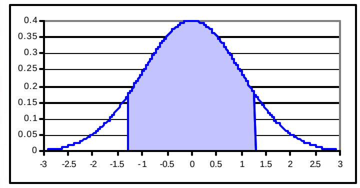
$$\sigma_{error_S(h)} = \sqrt{\frac{error_D(h)(1-error_D(h))}{n}}$$

Approximate this by a Normal distribution with

- mean $\mu_{error_S(h)} = error_D(h)$
- standard deviation

$$\sigma_{error_S(h)} \approx \sqrt{\frac{error_S(h)(1-error_S(h))}{n}}$$

Normal Probability Distribution



80% of area (probability) lies in $\mu \pm 1.28\sigma$

N% of area (probability) lies in $\mu \pm z_N \sigma$

N%:	50%	68%	80%	90%	95%	98%	99%
z_N:	0.67	1.00	1.28	1.64	1.96	2.33	2.53

Confidence Intervals, More Correctly

If

- S contains n examples, drawn independently of h and each other
- $n \geq 30$

Then

- With approximately 95% probability, $error_S(h)$ lies in interval

$$error_D(h) \pm 1.96\sqrt{\frac{error_D(h)(1-error_D(h))}{n}}$$

- equivalently, $error_D(h)$ lies in interval

$$error_S(h) \pm 1.96\sqrt{\frac{error_D(h)(1-error_D(h))}{n}}$$

- which is approximately

$$error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1-error_S(h))}{n}}$$

Calculating Confidence Intervals

1. Pick parameter p to estimate
   - $error_D(h)$
2. Choose an estimator
   - $error_S(h)$
3. Determine probability distribution that governs estimator
   - $error_S(h)$ governed by Binomial distribution, approximated by Normal when $n \geq 30$
4. Find interval (L,U) such that N% of probability mass falls in the interval
   - Use table of z_N values

Central Limit Theorem

Consider a set of independent, identically distributed random variables $Y_1 \ldots Y_n$, all governed by an arbitrary probability distribution with mean μ and finite variance σ^2. Define the sample mean

$$\bar{Y} \equiv \frac{1}{n}\sum_{i=1}^{n} Y_i$$

Central Limit Theorem. As $n \to \infty$, the distribution governing $\bar{Y}$ approaches a Normal distribution, with mean μ and variance $\frac{\sigma^2}{n}$.

Difference Between Hypotheses

Test h_1 on sample S_1, test h_2 on S_2

1. Pick parameter to estimate

$$d \equiv error_D(h_1) - error_D(h_2)$$

2. Choose an estimator

$$d \equiv error_{S_1}(h_1) - error_{S_2}(h_2)$$

3. Determine probability distribution that governs estimator

$$\sigma_d \approx \sqrt{\frac{error_{S_1}(h_1)(1-error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1-error_{S_2}(h_2))}{n_2}}$$

4. Find interval (L, U) such that N% of probability mass falls in the interval

$$\hat{d} \pm z_N\sqrt{\frac{error_{S_1}(h_1)(1-error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1-error_{S_2}(h_2))}{n_2}}$$

Paired t test to Compare h_A, h_B

1. Partition data into k disjoint test sets $T_1, T_2, \ldots, T_k$ of equal size, where this size is at least 30.
2. For i from 1 to k do
   $\delta_i \leftarrow error_{T_i}(h_A) - error_{T_i}(h_B)$
3. Return the value $\bar{\delta}$, where
   $\bar{\delta} \equiv \frac{1}{k}\sum_{i=1}^{k}\delta_i$

N% confidence interval estimate for d:

$\bar{\delta} \pm t_{N,k-1} s_{\bar{\delta}}$

$$s_{\bar{\delta}} \equiv \sqrt{\frac{1}{k(k-1)}\sum_{i=1}^{k}(\delta_i - \bar{\delta})^2}$$

Note δ_i approximately Normally distributed

N-Fold Cross Validation

- Popular testing methodology
- Divide data into N even-sized random folds
- For n = 1 to N
  - Train set = all folds except n
  - Test set = fold n
  - Create learner with train set
  - Count number of errors on test set
- Accumulate number of errors across N test sets and divide by N (result is error rate)
- For comparing algorithms, use the same set of folds to create learners (results are paired)

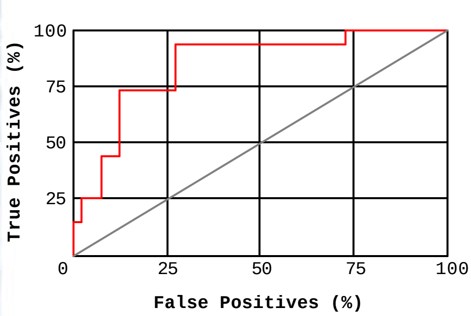
N-Fold Cross Validation

- Advantages/disadvantages
  - Estimate of error within a single data set
  - Every point used once as a test point
  - At the extreme (when N = size of data set), called leave-one-out testing
  - Results affected by random choices of folds (sometimes answered by choosing multiple random folds – Dietterich in a paper expressed significant reservations)

Receiver Operator Characteristic (ROC) Curves

- Originally from signal detection
- Becoming very popular for ML
- Used in:
  - Two class problems
  - Where predictions are ordered in some way (e.g., neural network activation is often taken as an indication of how strong or weak a prediction is)
- Plotting an ROC curve:
  - Sort predictions (right) by their predicted strength
  - Start at the bottom left
  - For each positive example, go up 1/P units where P is the number of positive examples
  - For each negative example, go right 1/N units where N is the number of negative examples

ROC Curve



ROC Properties

- Can visualize the tradeoff between coverage and accuracy (as we lower the threshold for prediction how many more true positives will we get in exchange for more false positives)
- Gives a better feel when comparing algorithms
  - Algorithms may do well in different portions of the curve
- A perfect curve would start in the bottom left, go to the top left, then over to the top right
  - A random prediction curve would be a line from the bottom left to the top right
- When comparing curves:
  - Can look to see if one curve dominates the other (is always better)
  - Can compare the area under the curve (very popular – some people even do t-tests on these numbers)