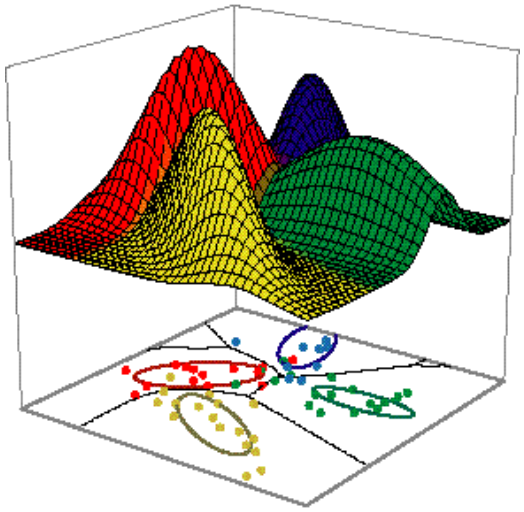
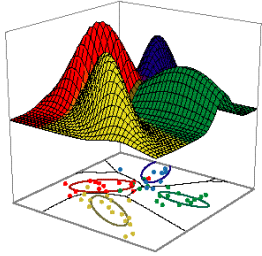
Part 4: Measuring Classification Accuracy



Measures of accuracy
True error vs. apparent error
Confidence in accuracy
Performance limits (Bayes error rate)
Comparing classifiers
Hypothesis Testing
Effects of experiment design

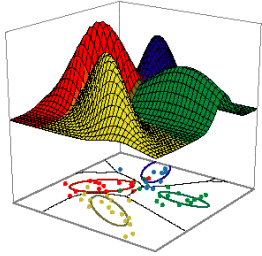
Some materials in these slides were taken from Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000 and Empirical Methods for Artificial Intelligence by Paul R. Cohen, MIT Press, 1995

Testing & reporting results



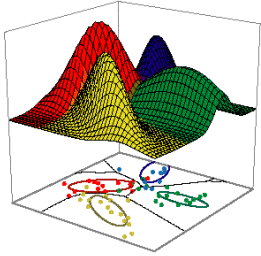
- How do we accurately measure and report the accuracy of a pattern classifier?
- How do we objectively compare two classifiers over a given problem?
- How can we predict how well a classifier will generalize, given its performance over our training data / testing data?
- How confident can we be in any of these estimates?
  - Effects of experiment design

Measures of classification accuracy

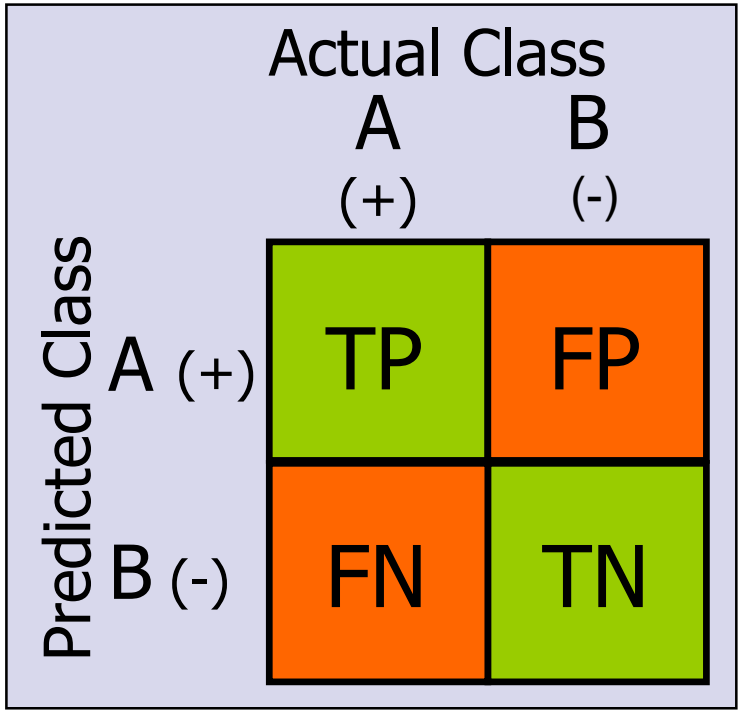


- Confusion table/matrix
  - Accuracy
  - Sensitivity / recall / true positive rate
  - Specificity
  - False Positive Rate
  - False Negative Rate
  - Positive Predictive Value / precision
  - Negative Predictive Value
  - False Discovery Rate
  - Matthews' correlation coefficient
  - Application-specific measures
  - F-measure
  - G-mean
- Receiver Operator Characteristic Curves
  - Area under curve

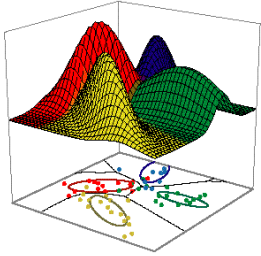
Confusion Table



- Correct predictions shown in green, errors in red.
  - Type I errors (or α error, or false positive)
  - Type II errors (β error, or a false negative)



Confusion Table

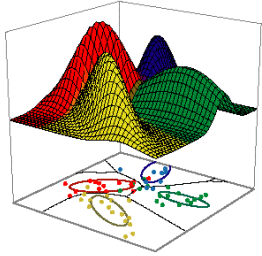


- Accuracy = (TP+TN) / (TP+TN+FN+FP)
- Sensitivity = Sn = TP / (TP+FN)
  - aka ‘recall’, ‘true positive rate’
- Specificity = Sp = TN / (TN+FP)
- False Positive Rate = 1-Sp
  - = FP/(TN+FP)
- False Negative Rate = 1-Sn
  - = FN/(TP+FN)
- Positive Predictive Value = TP / (TP+FP)
  - aka ‘precision’
- Negative Predictive Value = TN / (TN+FN)
- False Discovery Rate = FP / (TP+FP)
- F-measure = harmonic mean of Sn & PPV
- G-mean = geometric mean of Sn&Sp

Predicted Class \ Actual Class	A (+)	B (-)
A (+)	TP	FP
B (-)	FN	TN

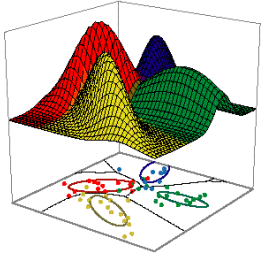
None of these in isolation can tell us how ‘accurate’ the classifier is.

Case study: PIPE II

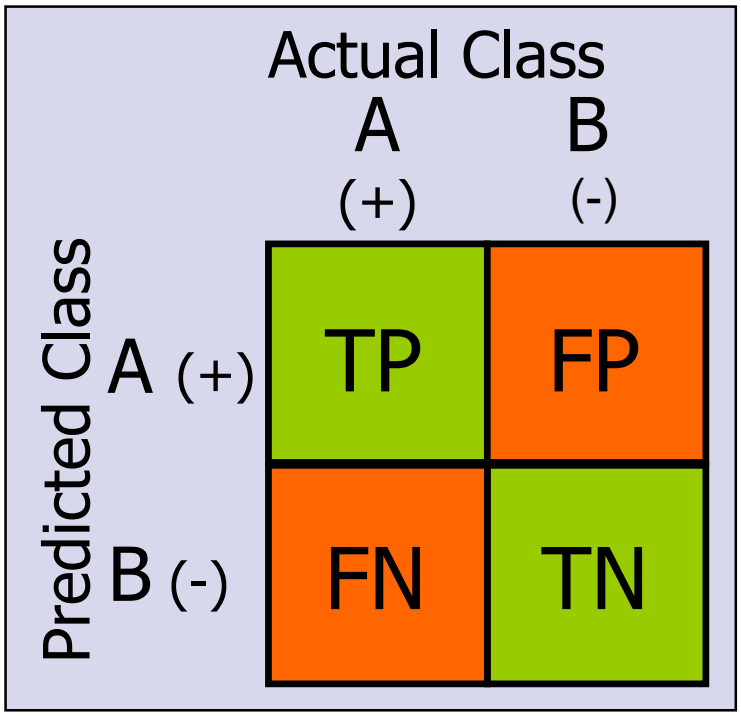


- The challenge:
  - Yeast has 6200 proteins in its proteome.
  - Every possible pair of yeast proteins could potentially interact.
  - Based on biological evidence, it is believed that approx 50K interactions exist in yeast.
  - Would like to computationally predict from sequence alone whether a given pair will interact.
  - It is very expensive to verify a prediction experimentally.
- The solution:
  - We have developed a classifier which tests a given pair of protein sequences and predicts whether they will interact *in vitro*.
  - We have reduced the computational complexity to the point where we can run it on all 18 million pairs in a day.
  - Through parameter tuning, we can achieve either:
    - 1) High specificity of 99% with medium sensitivity (%50)
    - 2) Very high specificity of 99.9% at the cost of a low sensitivity (25%)
- The $1M questions:
  - **Which parameter set is preferred?** *(aside: which one has highest g-mean?)*
  - **How many of the predicted interactions are likely to be true interactions?**

Case study: PIPE II

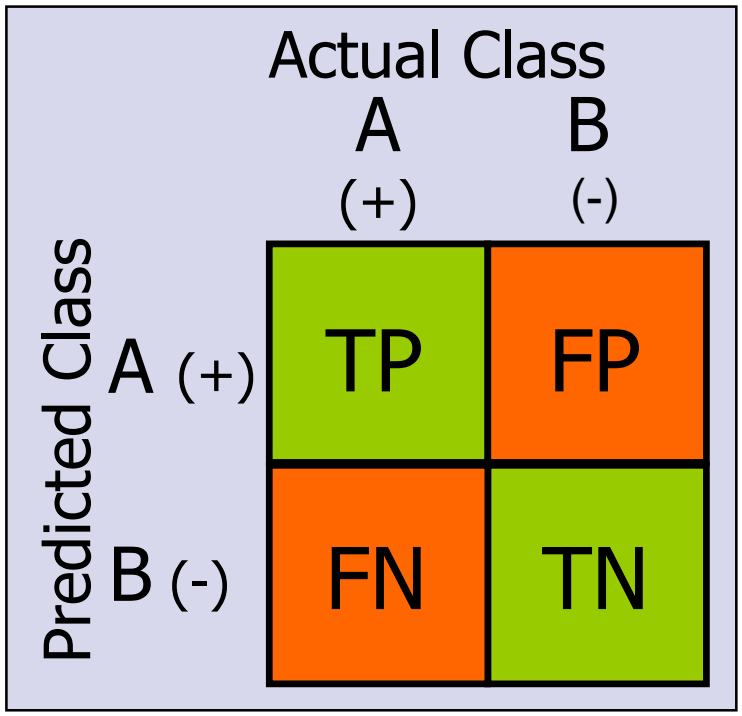


1) Sp=99%, Sn=50%



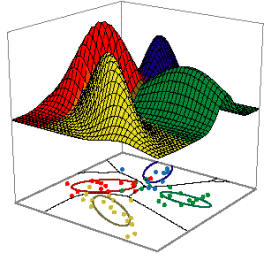
Pr_1 =

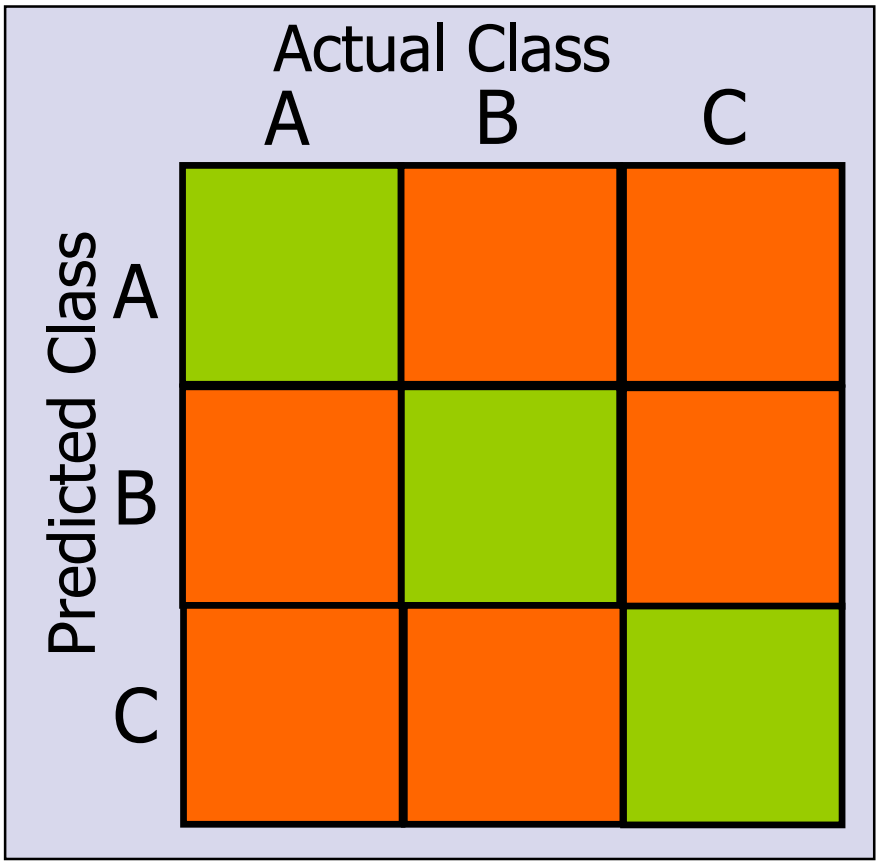
2) Sp=99.9%, Sn=25%

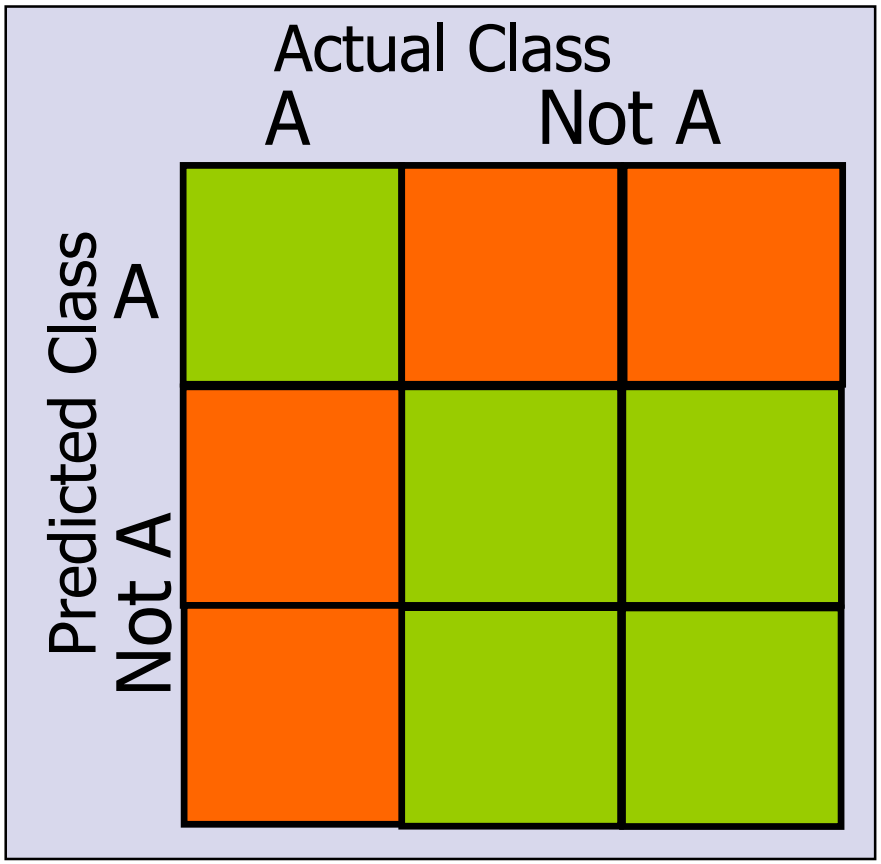


Pr_1 =

Confusion Table – multi-class

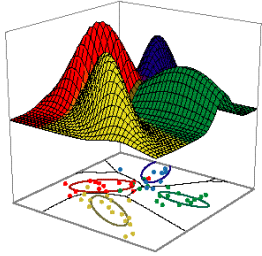




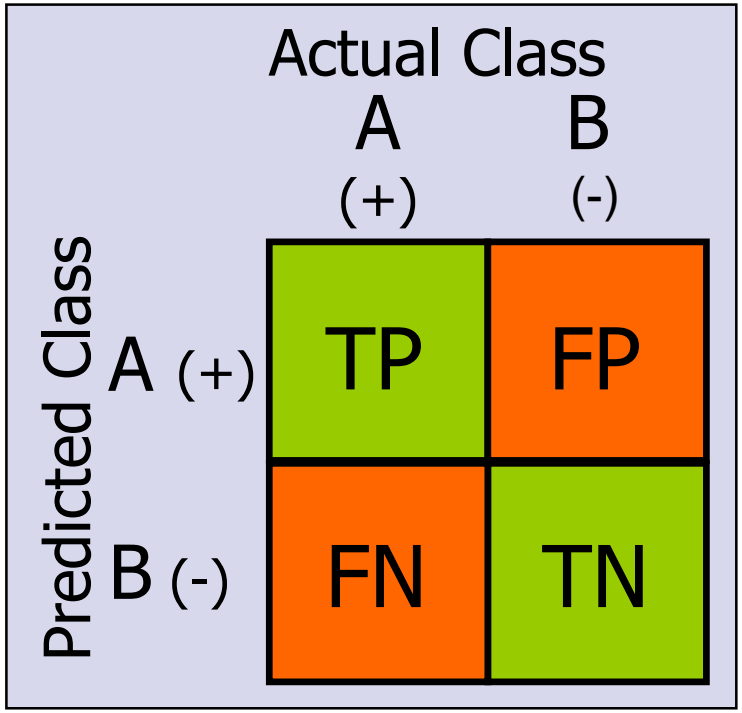


- Mass along diagonal indicates strength of classifier
- Highlights common errors.
- Simplify to a binary decision – can then compute accuracy measures for each possible binary decision

Mathew's Correlation Coefficient

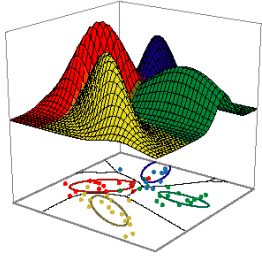


- Combines sensitivity & specificity into a single measure
- $CC_A = CC_B$ for 2-class case
  - For multiple classes, use average CC over all classes.
- Somewhat sensitive to class imbalance…
- -1<CC<1

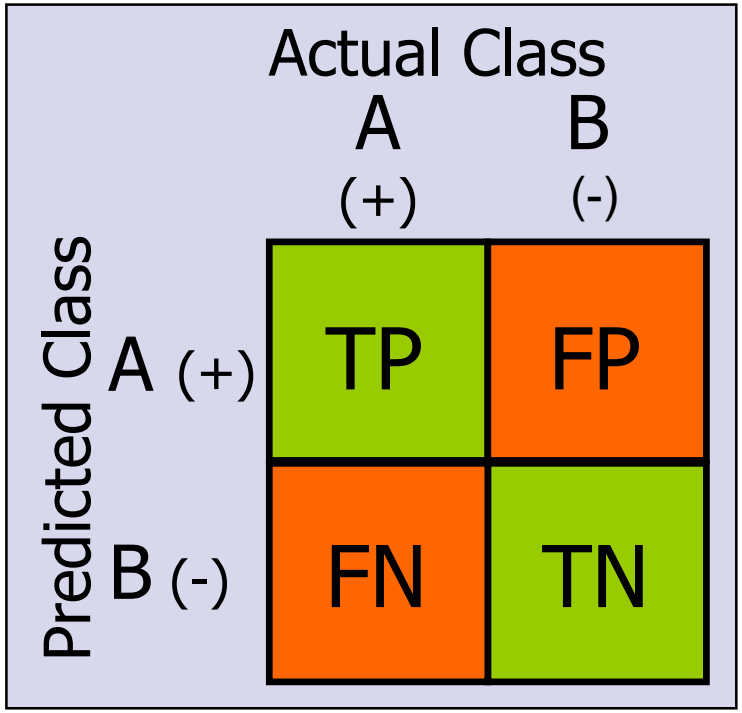


$$CC_A = \frac{(TP*TN-FP*FN)}{[(TP+FP)(TN+FN)(TP+FN)(TN+FP)]^{0.5}}$$

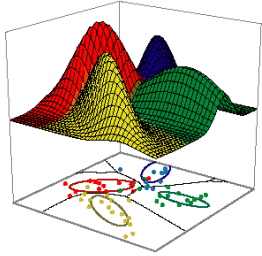
Custom Accuracy Measures



- Custom accuracy measures may depend on cost function
  - e.g. FN / (FN + FP)
    - Measures the ratio all observed errors which are of type FN
    - Important if FN errors are more costly than FP



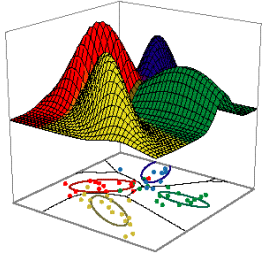
Lift



- Useful performance measure when faced with large class imbalance *(rare positive class)*
  - Example application: Telemarketing firm deciding which 10,000 homes (out of 10M) in Canada to contact, where score = likelihood that home owner will decide to purchase the product.
- Consider classifier as a ranker
  - Assigns score to each test point
  - Higher score = more likely to be positive
- Lift@x%
  - Consider classifying top-scoring x% of test data as positive
  - Compute proportion which are actually positive

$$Lift\,@\,x\% = \frac{\text{Proportion of actual positive cases in topscoring } x\% \text{ of cases}}{\text{"Background" proportion of actual positive cases among all cases}}$$

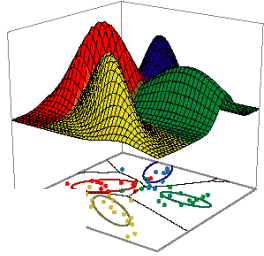
ROC Curves



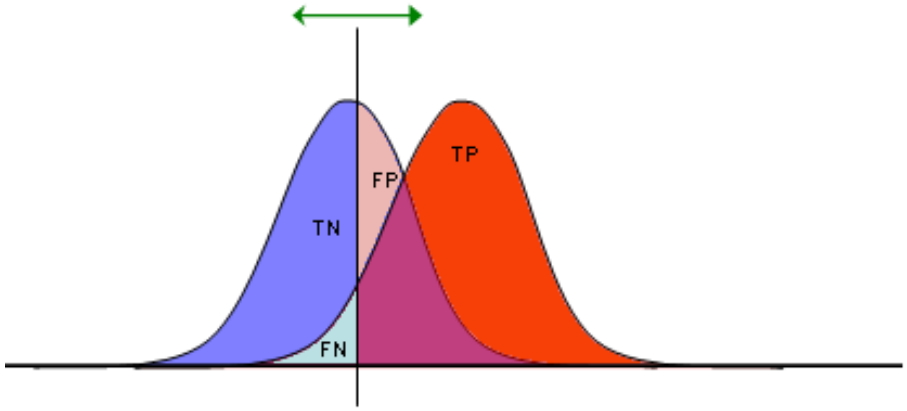
- Many classifiers have a tuneable parameter to trade-off between Sp and Sn.
  - Consider the classifier as a 'ranker'* – then apply decision threshold
  - Can tune parameter for different situations/uses
  - e.g. a neural network has a continuous output indicating likelihood of 'Class A'. Decision rule requires selection of threshold to apply to network output.
- Would like to represent performance at all possible parameter settings → **Receiver Operator Characteristic Curve**
  - First developed for signal detection theory.
  - Widely used in medical decision making.
- Plot TPR vs. FPR (i.e. Sn vs. 1-Sp) for all values of parameter
  - Starts with TPR=0, FPR=0 (i.e. threshold set to ∞)
  - Decrease threshold one step at a time and plot (FPR,TPR)
  - End in upper right corner where all samples are classified as positive → Sn=FPR=1

*Peter Flach, "The many faces of ROC analysis" tutorial
http://www.cs.bris.ac.uk/~flach/ICML04tutorial/

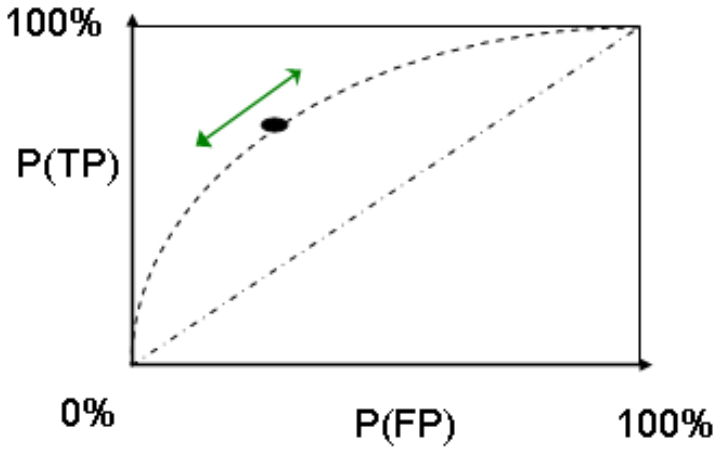
Interpreting ROC Curves



Tunable decision threshold



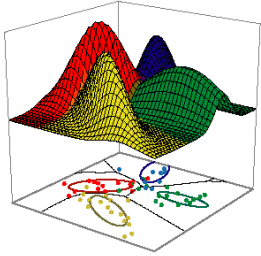
TP	FP
FN	TN
1	1



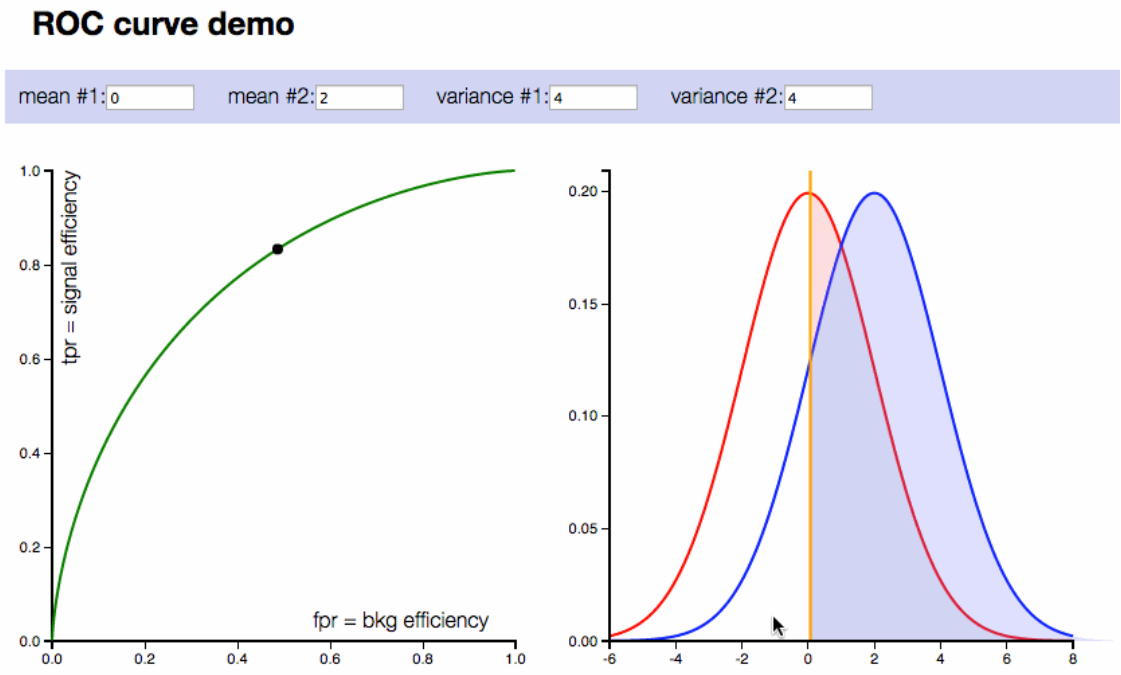
Wikipedia Contributors, https://commons.wikimedia.org/wiki/File:ROC_general.svg

See http://arogozhnikov.github.io/2015/10/05/roc-curve.html for a great ROC demo

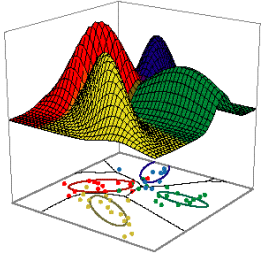
Great ROC Demo



- See http://arogozhnikov.github.io/2015/10/05/roc-curve.html



ROC Curves



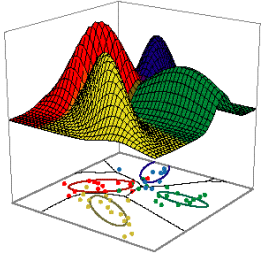
Algorithm 1 Conceptual method for calculating an ROC curve. See algorithm 2 for a practical method.

Inputs: L, the set of test instances; $f(i)$, the probabilistic classifier's estimate that instance i is positive; min and max, the smallest and largest values returned by f; $increment$, the smallest difference between any two f values.

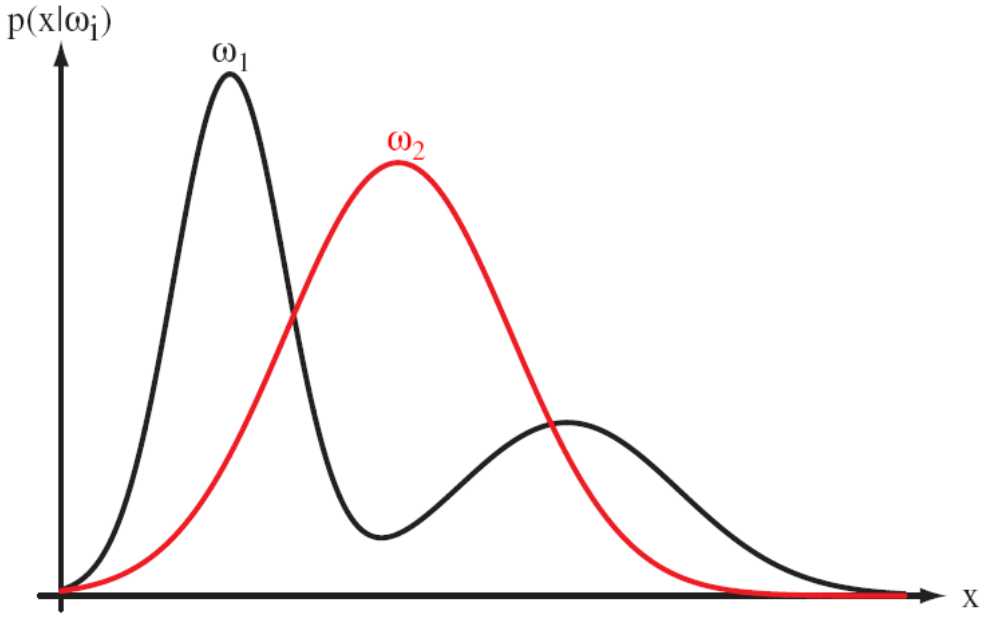
```
1:  for t = min to max by increment do
2:      FP ← 0
3:      TP ← 0
4:      for i ∈ L do
5:          if f(i) ≥ t then                    /* This example is over threshold */
6:              if i is a positive example then
7:                  TP ← TP + 1
8:              else        /* i is a negative example, so this is a false positive */
9:                  FP ← FP + 1
10:             end if
11:         end if
12:     end for
13:     Add point (FP/N, TP/P) to ROC curve
14: end for
15: end
```

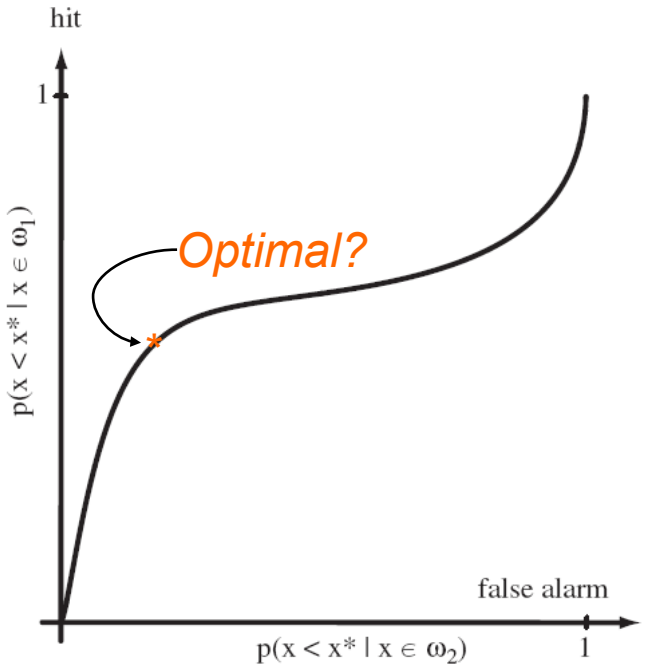

From: T. Fawcett (2004). "ROC Graphs: Notes and Practical Considerations for Researchers". *Technical report*, Palo Alto, USA: HP Laboratories.

ROC Curve

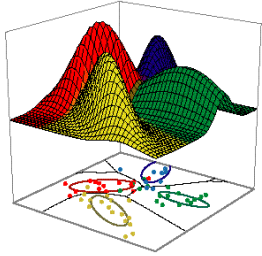


- Curve is not necessarily symmetric
- Can be informative in setting threshold to balance benefit of TP against cost of FP



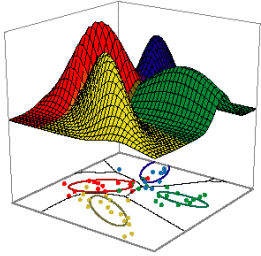


Area under the ROC Curve

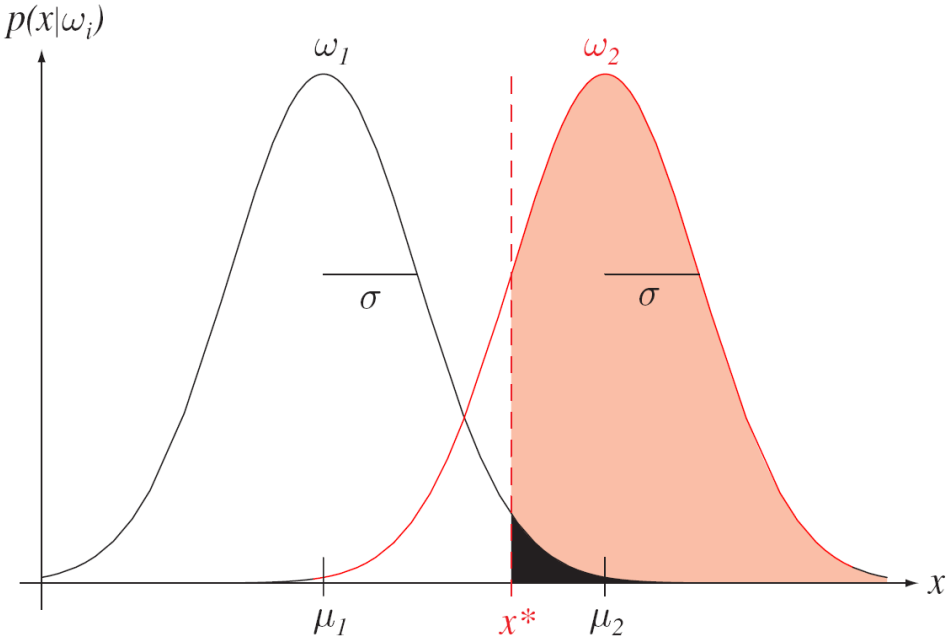


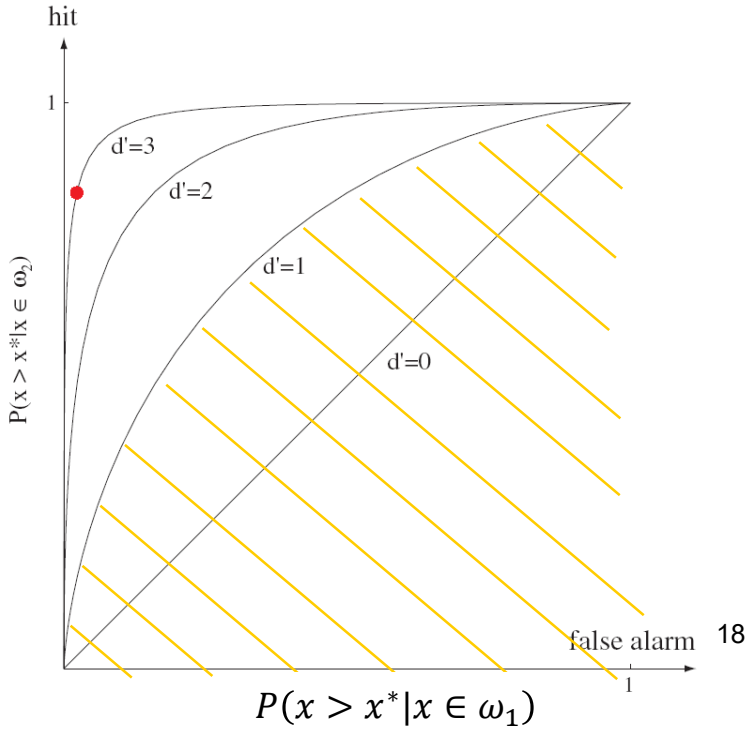
- Area under an ROC curve (AUC) summarizes performance of a classifier over its full range of sensitivity/specificity operating points.
  - *But sometimes we don't care about the whole range…*
  - Independent of particular cost function which might influence threshold placement
  - Ranges from 1 (perfect) to 0 (worst)
    - Random = 0.5

Area under the ROC Curve

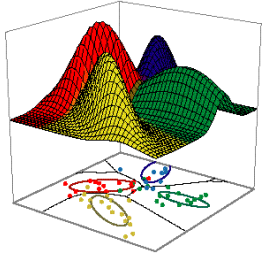


- Assume distribution of classifier output to be thresholded is normally distributed for each class
  - Define *discriminability*: $d' = \frac{|\mu_2 - \mu_1|}{\sigma}$
- Here, area under ROC curve is a function only of d'

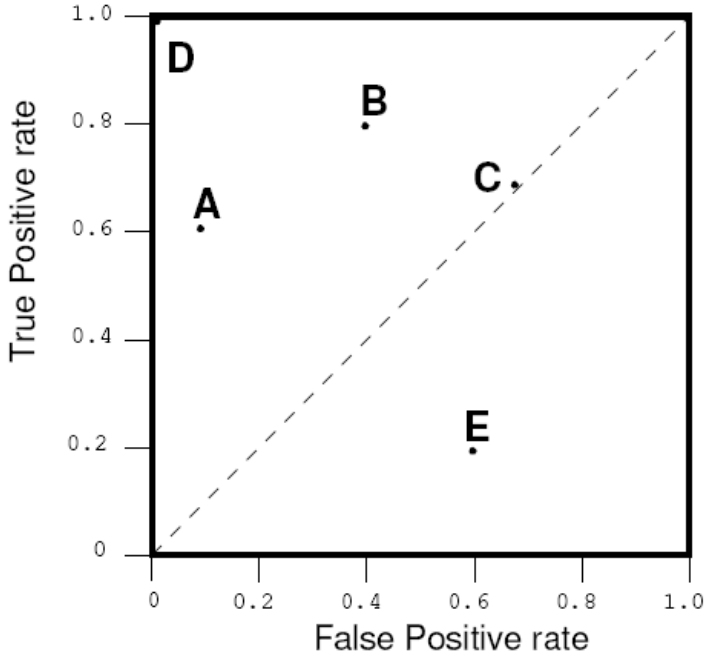




ROC Curves

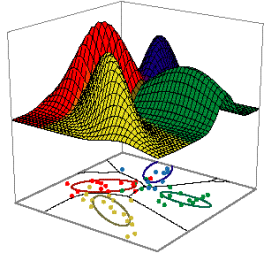


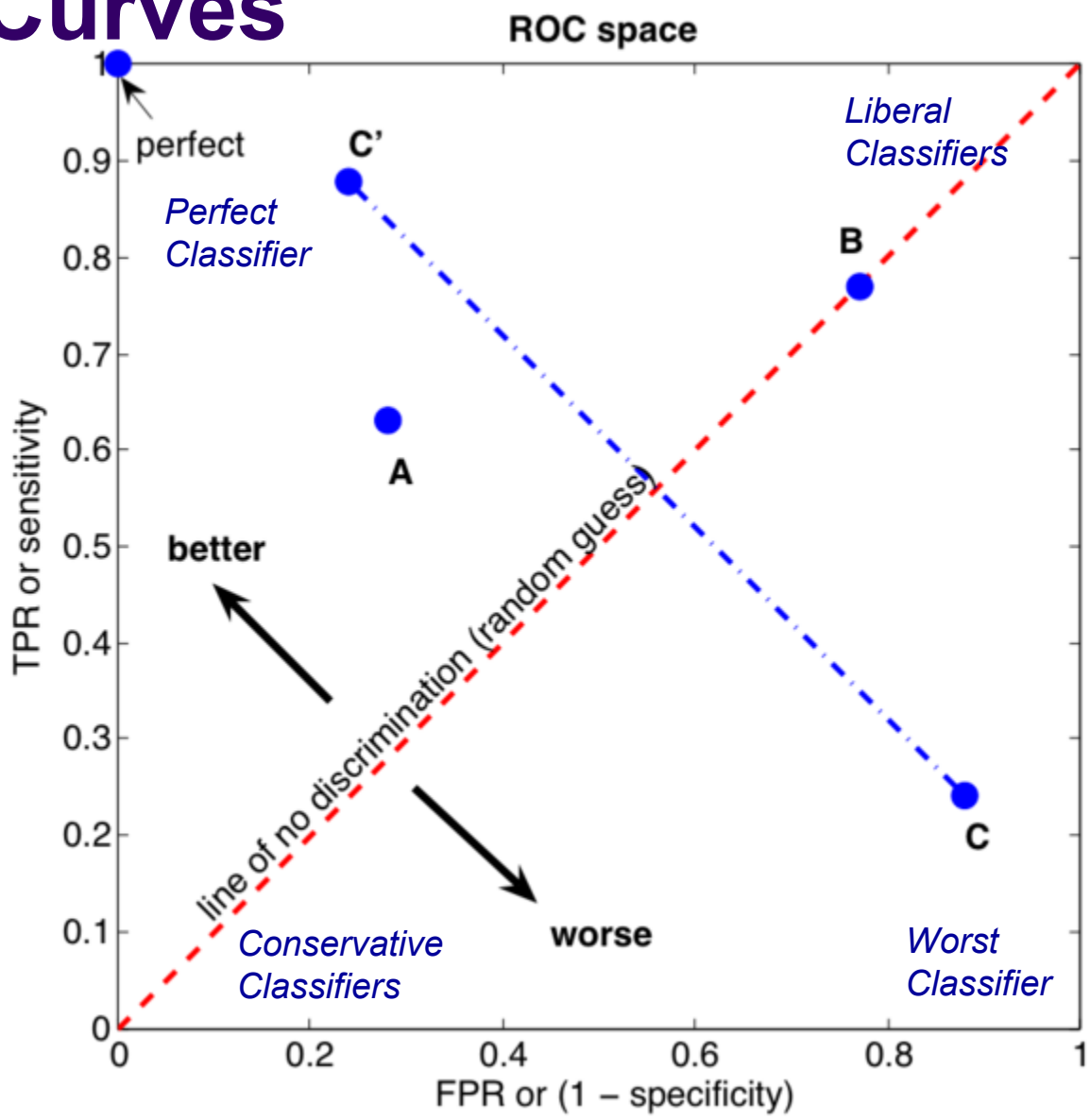
- Can also use ROC curves to compare classifiers which are not ‘tunable’
  - i.e. those which produce a single (TP,FP) result.



From: T. Fawcett (2004). "ROC Graphs: Notes and Practical Considerations for Researchers". *Technical report*, Palo Alto, USA: HP Laboratories.

ROC Curves





A

TP=63	FP=28	91
FN=37	TN=72	109
100	100	200

TPR = 0.63

FPR = 0.28

ACC = 0.68

B

TP=77	FP=77	154
FN=23	TN=23	46
100	100	200

TPR = 0.77

FPR = 0.77

ACC = 0.50

C

TP=24	FP=88	112
FN=76	TN=12	88
100	100	200

TPR = 0.24

C'

TP=88	FP=24	112
FN=12	TN=76	88
100	100	200

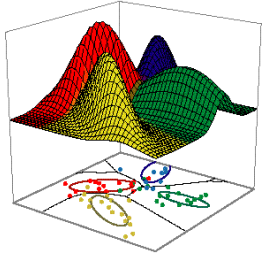
TPR = 0.88

FPR = 0.24

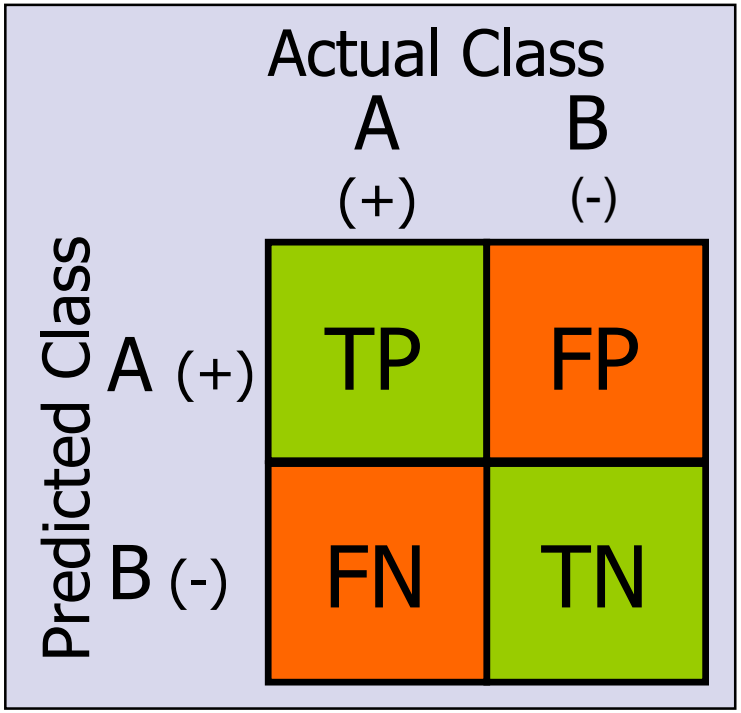
ACC = 0.82

Wikipedia Contributors, http://en.wikipedia.org/wiki/Image:ROC_space.png

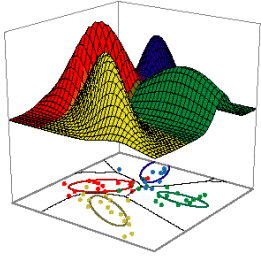
ROC Curves



- ROC curves are not affected by class skew compared to accuracy.
  - Why?
  - Recall, we are plotting FP/(FP+TN) vs. TP/(TP+FN)



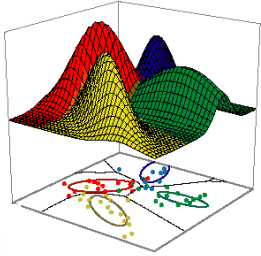
Multi-class ROC Surfaces

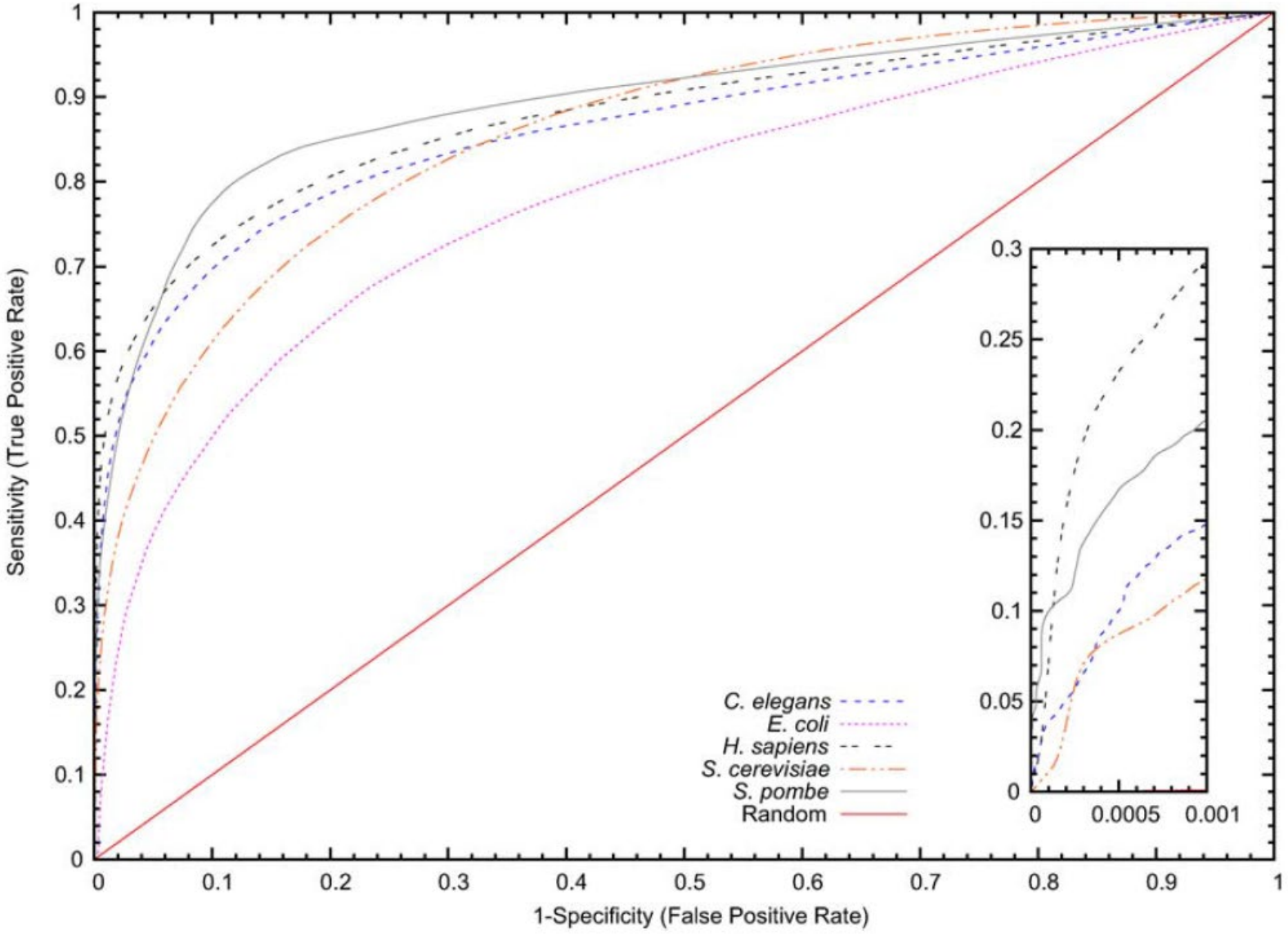


- For multi-class case, construct an ROC surface
- Each dimension is either:
  - Class *i* vs. all
  - Class *i* vs. class *j*
- Calculate volume under ROC surface (VUS)
  - Can project down to set of 2D curves and average
    - MAUC (Hand & Till, 2001): 1-vs-1, unweighted average
    - (Provost & Domingos, 2001): 1-vs-rest, AUC for class c weighted by P(c)

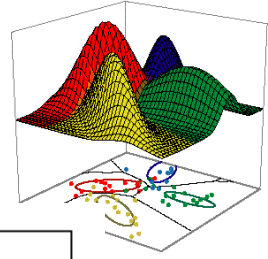
See Peter Flach's tutorial (esp. part 3) at: http://www.cs.bris.ac.uk/~flach/ICML04tutorial/

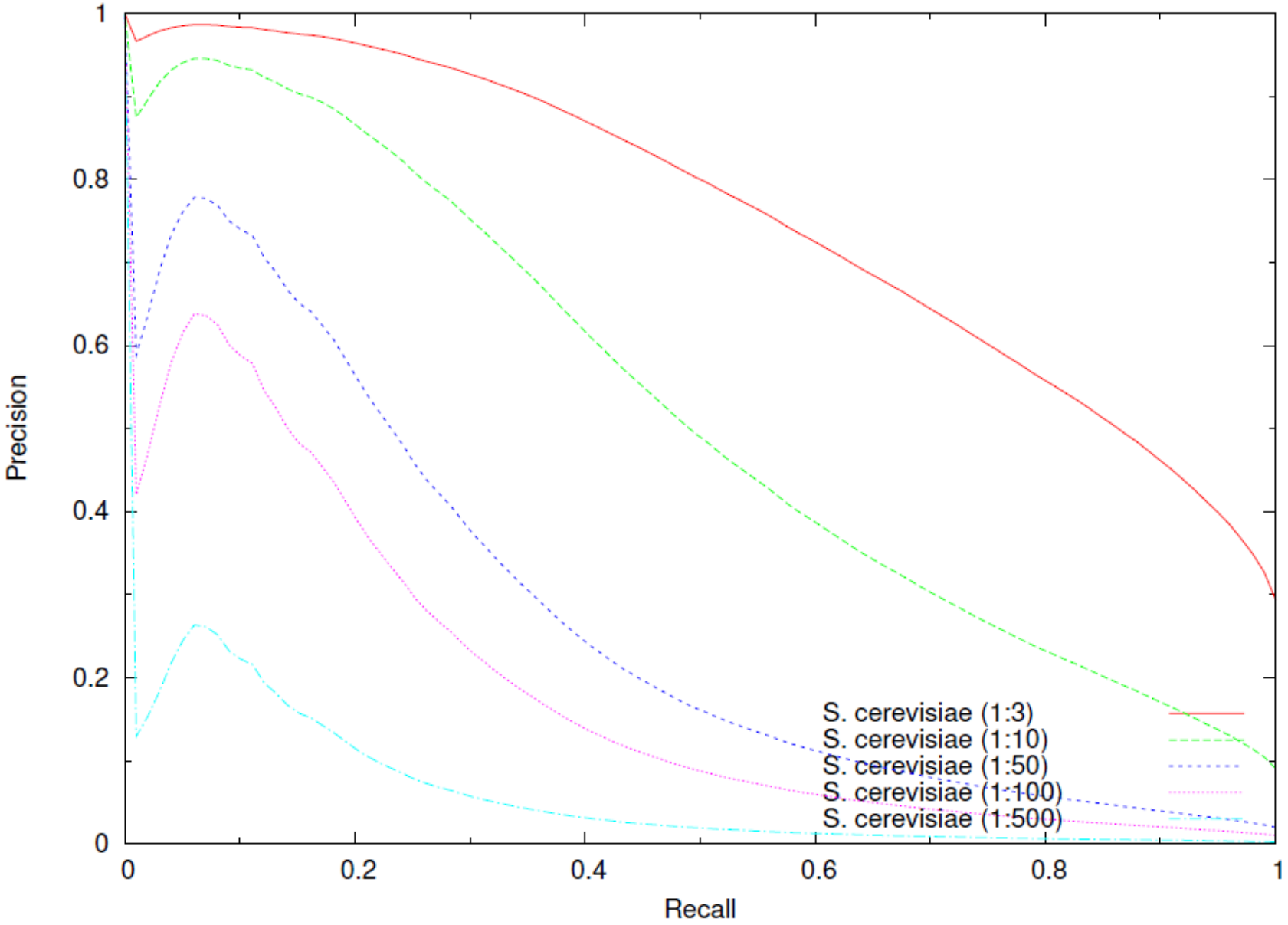
PIPE ROC Curve



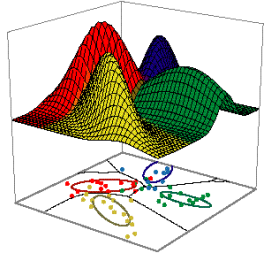


PIPE Precision-Recall Curve



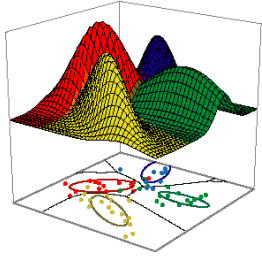


Estimating ‘true error’



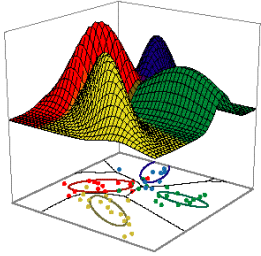
- Goal of pattern classification is to learn from training data in order to perform accurately over **new future data** (generalization)
  - How do we estimate this performance from our limited training data?
- Definitions:
  - Error rate = (number of errors / number of samples)
    - (1 − error rate) = accuracy
  - *Apparent error rate* (aka *resubstitution error rate*)
    - Error rate observed when trained classifier applied to training data.
  - *True error rate* (aka *generalization error rate)*
    - Error rate observed when trained classifier applied to new independent test data
- True error > apparent error (almost always)

Estimating 'true error'



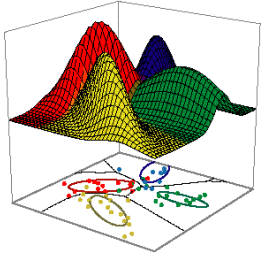
- Want to know generalization error for 2 reasons:
  - 1) Is our classifier accurate enough to be useful?
  - 2) Comparing 2 methods (later)
- To estimate true error requires making assumptions about method or problem or both
  - Failure of assumptions leads to incorrect performance estimates
  - e.g. Assume that all training points were drawn i.i.d from some distribution. Assume that future test points will be drawn from the same distribution (i.e. that training data is *representative* of future test data).
  - Some assumptions explicit (e.g. parametric models), other are more subtle and difficult to test for validity (e.g. Daily factory performance & hold-out testing)
- Only heuristics are available to guess which classifier will generalize better for a given problem
  - If there were a fool-proof way, would incorporate it into learning algorithm → violate No Free Lunch Theorem.

Estimating 'true error' for parametric models



- For parametric models, could estimate generalization rate directly from assumed parametric model
  - e.g. for 2-class multivariate normal case, can estimate error using Bhattacharyya or Chernoff bounds by substituting estimated parameters (μ, Σ) into equations
- 3 problems:
  - 1) Often overly optimistic. Characteristics that make training data peculiar or unrepresentative will not be revealed.
  - 2) Must always suspect validity of assumed parametric model. A performance evaluation based on same model cannot reveal that assumed model is incorrect.
  - 3) When distributions are not simple (Normal), very difficult to compute error estimates even when class distributions known.

Estimating 'true error' from 'apparent error'



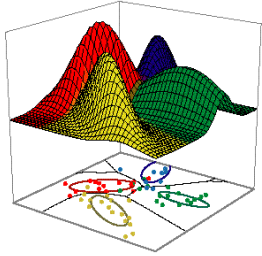
- True error > apparent error (almost always)
  - Apparent error is a biased estimate (lower than reality), however, it has low variance.
    - Sometimes preferred to a cross-validation estimate which has zero bias, but high variance*
- If test data drawn i.i.d. from same distribution, we have an upper bound on test error from apparent error: (N=# training samples, h=VC Dimension, h<N)**

$$P\left(test\_error \le train\_error + \sqrt{\frac{h\left(\log\left(\frac{2N}{h}\right)+1\right)-\log\left(\frac{\eta}{4}\right)}{N}} \right) = (1-\eta)$$

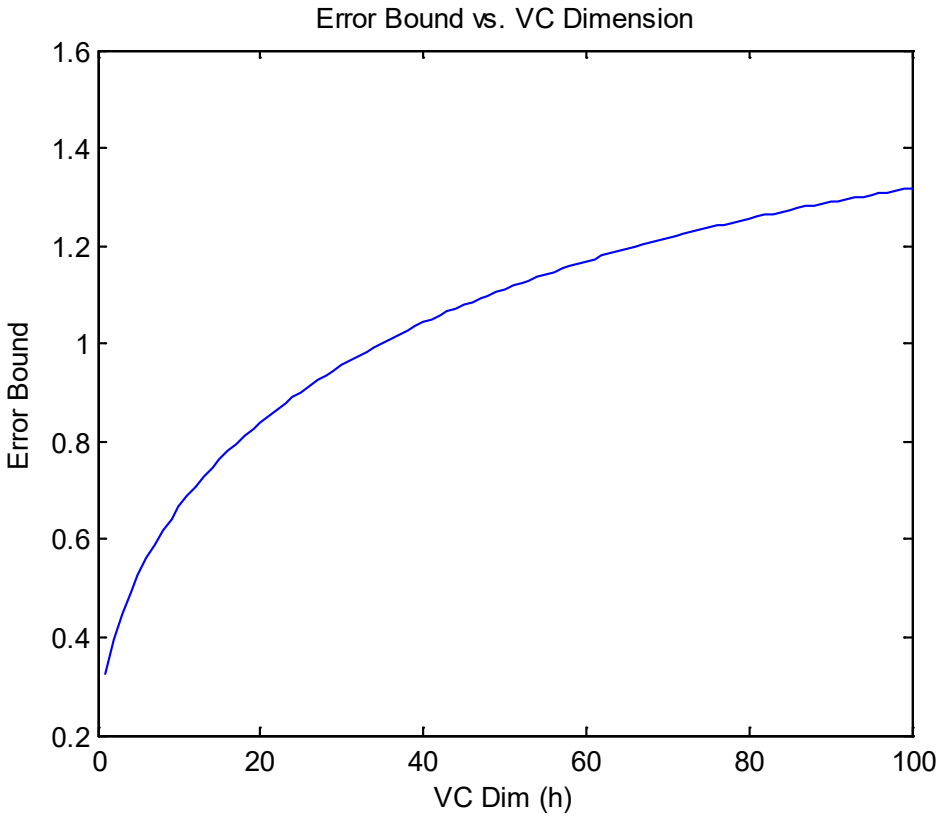
** See for example Braga-Neto, "Fads and Fallacies in the name of Small-Sample Microarray Classification" IEEE Signal Processing Magazine, 2007, 24(1):91-99.*

*** Andrew Moore, "VC Dimension Tutorial",* http://www.autonlab.org/tutorials/vcdim.html

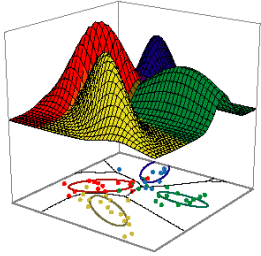
VC-Dimension Example



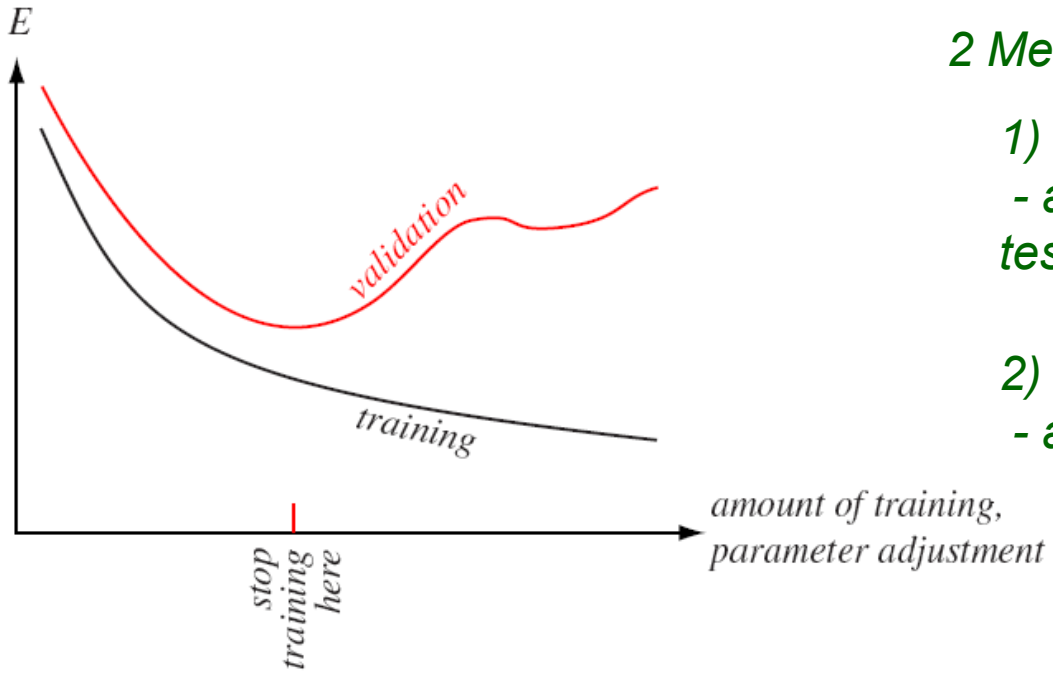
- VC-Dimension equation where N=100, apparent error = 0.3, eta=0.05:
  - (note that it is only valid for h<N)
  - Bound is useless beyond h=40 (becomes greater than 1)



Estimating 'true error' through hold out testing



- Simplest form: split training data into 'train' and 'validation' set.
  - "Hold out testing"
- Repeatedly train on 'training data' while adjusting model parameters until minimum error observed on 'validation data'.
- Estimate true error using validation data. Stop training when min observed. Goal is to avoid overfitting training data.

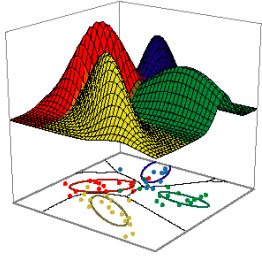


2 Methodological errors to avoid:

1) "Testing on the training set"
- avoid overlap between train, test, and validation sets

2) "Training on the testing set"
- avoid with separate test data

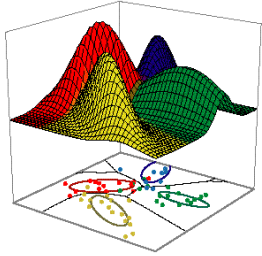
Estimating 'true error' through cross-validation



- General form: m-fold cross-validation
  - Randomly split training data into m disjoint sets of size *n/m*, where *n* is total patterns in $\mathcal{D}$.
  - Classifier trained *m* times, each with a different subset held out as validation set.
  - Estimated performance is average of *m* validation errors.
- Special case: Leave-one-out (aka Jackknife)
  - Taken to extreme where *m=n*
  - Leave out a single sample each time for validation
- Advantages of cross-validation:
  - These error estimation methods can be applied to any method (parametric or not)
  - Always better than a single hold-out test*
    - for nontrivial methods insensitive to training data ordering

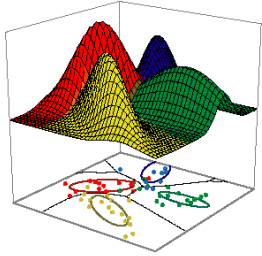
*Blum, Kalai, and Langford, "Beating the Hold-Out" (1999) *ACM*

Estimating 'true error' through cross-validation

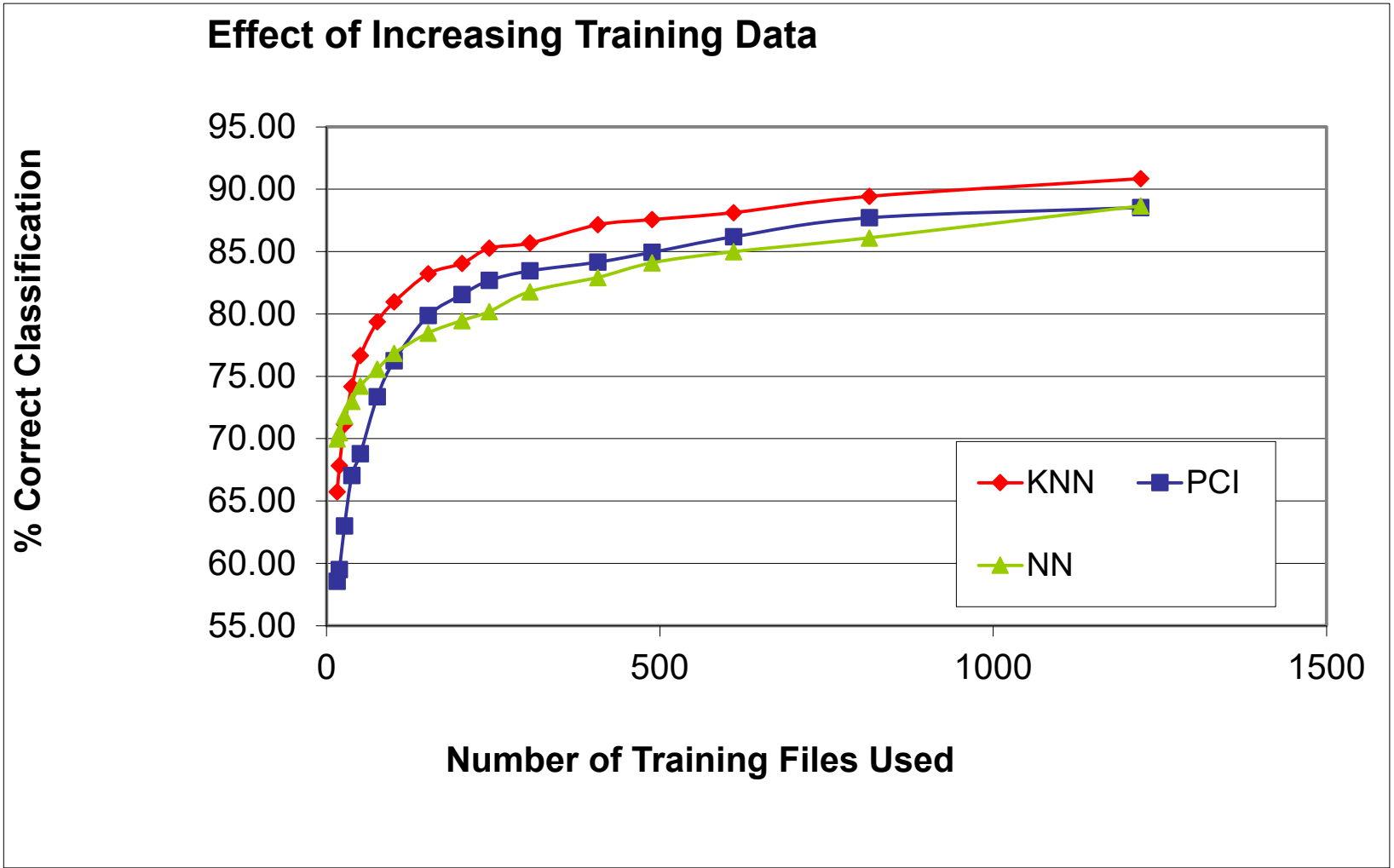


- How much of $\mathcal{D}$ should be used for validation?
  (as opposed to portion used for training)
- Heuristics for choosing proportion λ (0< λ <1)
  - (where λ is proportion of data used for validation)
  - Should always set λ < 0.5
    - Only estimating one parameter ('when to stop training') while the rest of data is used to estimate all remaining model parameters
  - If model has many free parameters (degrees of freedom), then λ should be decreased →more training data
  - If model has few free parameters, error estimate tends not to be sensitive to λ.
- How does λ relate to *m* in *m*-fold cross-validation?
- Caution: cross-validation is heuristic and will not work for every problem.
  - will not always necessarily identify optimal stopping time

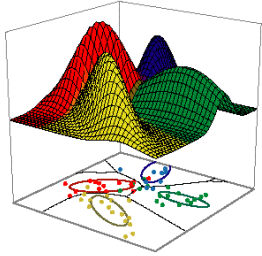
Effect of λ *in* cross-validation



- Here, λ is increased from 1/16 to 1/2
- Three techniques used for ATP protein binding site prediction



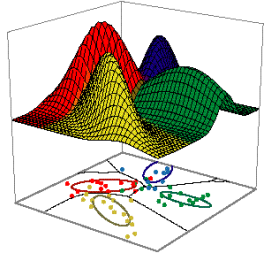
m-Fold Cross-Validation vs. LOO



- Weakness of cross-validation estimates of test error:
  - Get error estimates from classifiers trained on (*m*-1)/*m* of data
  - However, final classifier will be trained on full data set
  - Therefore, the classifiers used to estimate error rate are different from the final classifier.
- Leave-one-out has benefit that each of the *m* classifiers is very similar to final classifier (differ by 1 seq)
  → improved estimate of error rate
- Downside is increased computational complexity.
- Consider *progressive validation**
  - *Use hold-out test set, but after each sample is tested, add to training set and retrain. Requires fewer train/test iterations, but estimate is almost as good as jackknife/LOO.*

*Blum, Kalai, and Langford, "Beating the Hold-Out" (1999) *ACM*

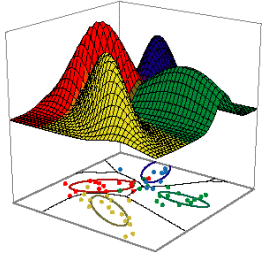
Estimating 'true error' through cross-validation

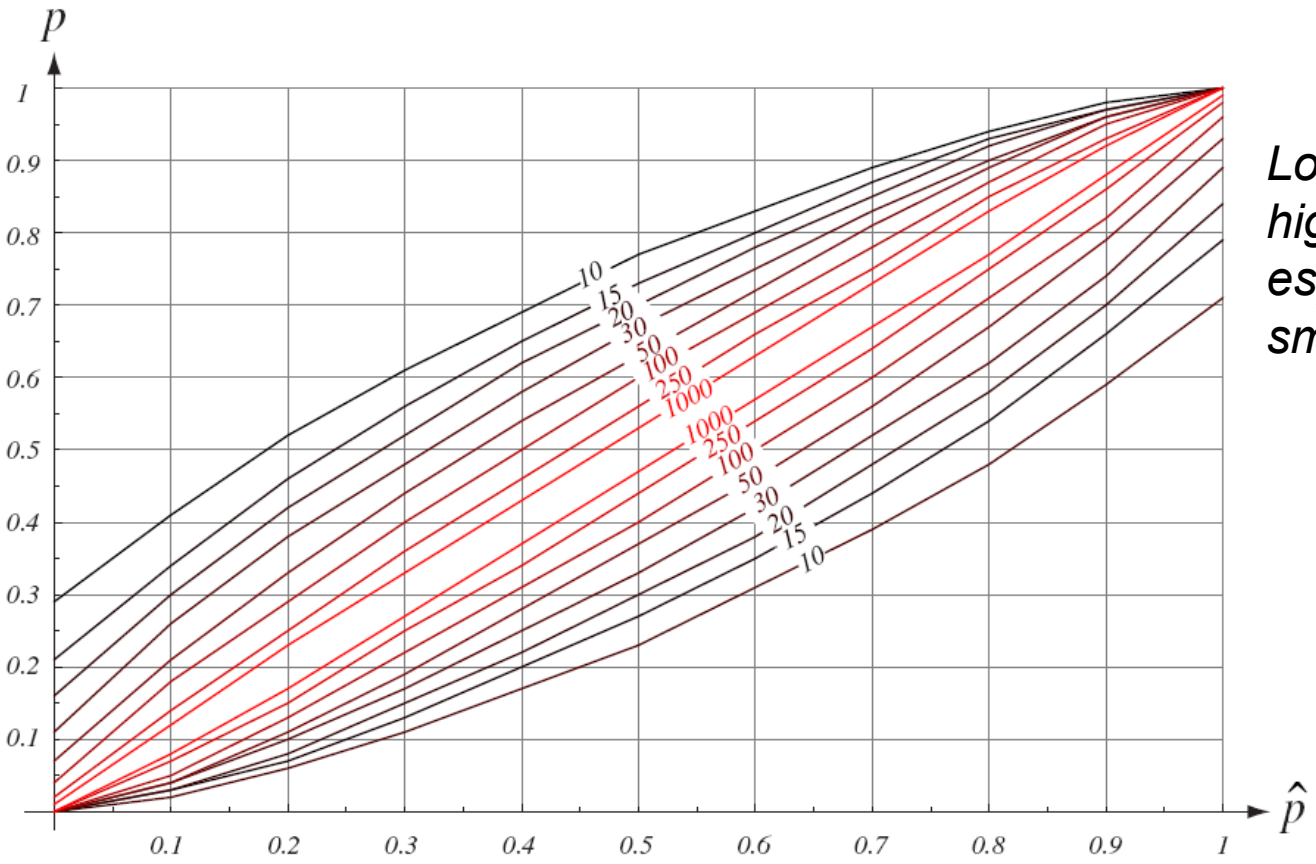


- Once trained using cross-validation, validation error rate gives estimate of independent test error rate
  - How accurate is estimate of error rate?
  - Function of validation set size:
    - If p is true error rate and k of n' samples observed to be in error, P(k) follows binomial distribution: $P(k) = \binom{n'}{k} p^k (1-p)^{n'-k}$
    - Maximum likelihood estimate of p is $\hat{p} = \frac{k}{n'}$ (as expected).
    - Standard error of estimated error rate is: $SE = \sqrt{\frac{\hat{p}(1-\hat{p})}{n'}}$
    - Can plot confidence in estimate (next slide):

** Underlying assumption that validation data set representative of test data*

Estimating 'true error' through cross-validation

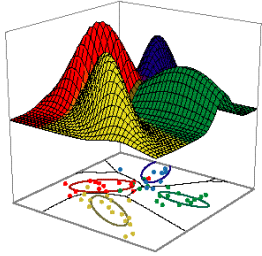




Low bias but high variance, especially for small samples

FIGURE 9.10. The 95% confidence intervals for a given estimated error probability $\hat{p}$ can be derived from a binomial distribution of Eq. 38. For each value of $\hat{p}$, the true probability has a 95% chance of lying between the curves marked by the number of test samples n'. The larger the number of test samples, the more precise the estimate of the true probability and hence the smaller the 95% confidence interval. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

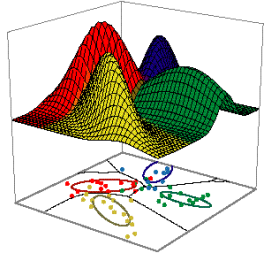
Bootstrapping



- Resampling/cross-validation is unbiased, but has large variance for small samples.
  - Variance effect can dominate in small samples (<30)
- *Bootstrapping* provides lower variance at expense of bias
  - Repeated 2-CV:
    - repeatedly compute 2-CV estimate with 50/50 split (~200 times)
  - *e0* estimate:
    - draw *n* training samples with replacement from original dataset of size *n*. Remaining samples go to test set.
    - *e0* estimate is error rate over test set.
    - Normally repeated (Monte Carlo simulation) ~200 times
    - End up with 63.2% training data and 36.8% unique test data
  - *.632B* bootstrap estimator: *.632B=.368*app+.632*e0*
    - *app* = apparent error over all cases (training & testing data)

Sholom Weiss and Casmir Kulikowski, Computer Systems That Learn, Morgan Kaufmann, 1991.

Bootstrapping – LOO*

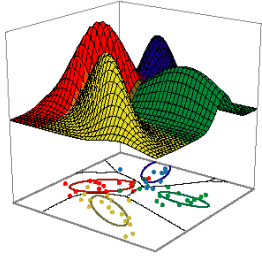


- *e0* & repeated 2-CV have pessimistic bias; *.632B* has optimistic bias
  - Provides upper and lower bounds on error rate.
- Define *corrected leave-one-out* estimate:
  - Use LOO with upper bound of *.632B* and lower limit of repeated 2-CV

$$LOO^* = \begin{cases} .632B \text{ if } LOO < .632B \\ 2-CV \text{ if } LOO > 2-CV \\ LOO \text{ else} \end{cases}$$

Sholom Weiss and Casmir Kulikowski, Computer Systems That Learn, Morgan Kaufmann, 1991.

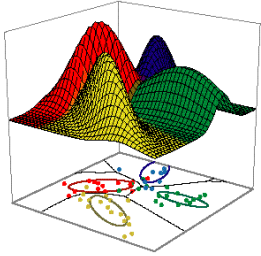
Getting the most out of the data*



- Use resampling to estimate true error rate
  - Repeated train-and-test partitions to estimate error rate
  - Select the classifier (and complexity fit) with the lowest error
    - (e.g., optimize # hidden nodes in ANN)
- Choose resampling method depending on # samples, *n*:
  - *n* >100: LOO or 10-CV
  - *n* <100: LOO
  - *n* <50: 'corrected LOO*'
  - Note that if class sizes are highly imbalanced, treat n as size of smallest class (especially when that small class is of interest!)
- Apply the identical classification method to all samples
  - Train a final classifier on all samples, but don't change the complexity fit.
- Select 'simplest' classifier within 1σ (s.d.) of minimum observed error

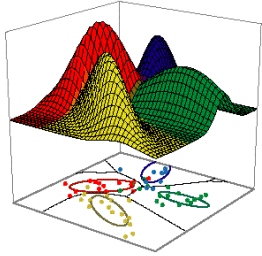
*Sholom Weiss and Casmir Kulikowski, Computer Systems That Learn, Morgan Kaufmann, 1991.

Limits to accuracy



- Bayes or Indistinguishability error rate gives us the minimum theoretical error rate possible for a problem
  - Due to overlapping $p(\mathbf{x}|\omega_i)$ for different *i*
  - Inherent property of the problem and can never be eliminated.
  - Function of '*discriminability*' of the problem…
  - Bayes error often cannot be computed (so not that useful)
- Estimation Error (on top of Bayes error)
  - Due to limited (finite) training data. Estimated parameters will be inexact.
  - Can reduce with more training data.
- Model Error (on top of Bayes error)
  - Error due to having an incorrect model
    - E.g. assumed normal distribution when it was skewed.

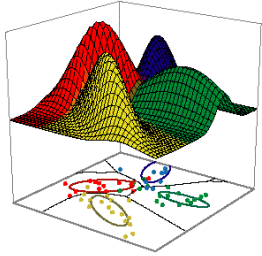
What else can go wrong?



- Poor features, data errors, mislabeled classes
  - Available features may simply not be correlated with outcome.
  - Measurement errors of features
    - e.g. microarrays – high noise
  - Misclassification of sample data
    - e.g. cause of death for 5-year outcome in a medical trial
  - Missing data
    - e.g. not all tests are conducted on all patients

*Sholom Weiss and Casmir Kulikowski, Computer Systems That Learn, Morgan Kaufmann, 1991.

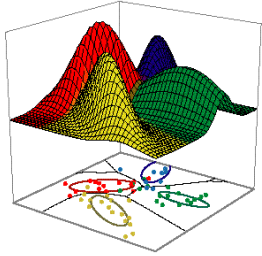
What else can go wrong?



- Unrepresentative samples
  - Assume that training data is sampled randomly
    - Don't throw out samples that don't fit preconceived notion of validity *(goes back to how to treat outliers in training data…)*
  - Assume that training data is *representative*
    - Measurement device may be nonstationary (calibration)
  - Test set must be independent
    - Medical studies are often conducted by one team at one hospital and confirmed by an independent team at another hospital to ensure independence.
    - In a discrete problem (finite number of patterns to be observed) when training and test datasets are very large they necessarily overlap and we are testing on the training data.
      - e.g. protein sequence windows

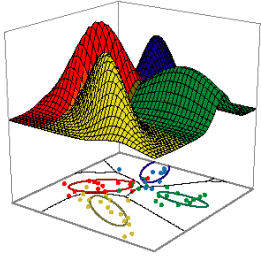
*Sholom Weiss and Casmir Kulikowski, Computer Systems That Learn, Morgan Kaufmann, 1991.

3 Spurious effects of experiment design on reporting accuracy



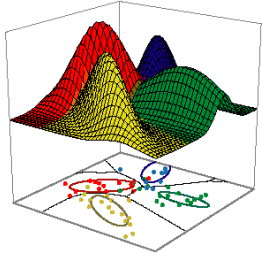
- 1) Order effects
  - Some learners are sensitive to the order that samples are presented. Consider presenting in every possible order to each.
- 2) Issues of nearing the performance floor/ceiling
  - If there is no room for improvement, hard to draw strong conclusions about slight differences between methods.
- 3) Regression towards the mean
  - If chance plays a role in performance score, don't be tempted to choose 10 problems on which the old algorithm did the worst as the basis for comparison.
    - If you reran without changing anything, you are almost certain that better scores will be observed.

Other effects of experiment design on reporting accuracy



- Baseline/control experiments
  - When evaluating your new method, carefully choose ‘control’ method to compare with
  - Select at least one contemporary method, not an out-of-date easy-to-beat has-been method.
  - Also include a simple method (e.g. Random, naïve Bayes, default rule (prior only)) to make sure that the test cases are not trivial.

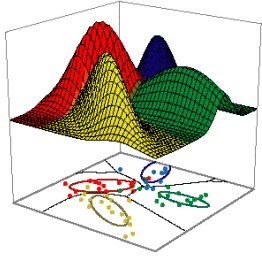
Compare your classifier to a random decision



- Can apply classical hypothesis testing:
  - H_0="the predicted class labels are independent from the actual class assignments"
  - Apply χ^2 test against confusion matrix (as seen in Part 3)
- Statistical distribution rarely known
  → turn to permutation tests:
  - Repeat many times:
    - Use cross-validation
    - Randomize class labels in training dataset, train classifier
    - Test classifier on correctly labelled test data
    - If features and class truly independent, should be no disadvantage to randomize training labels…
  - Count the proportion of randomized classifiers which scored as high or higher than our classifier → p-value.

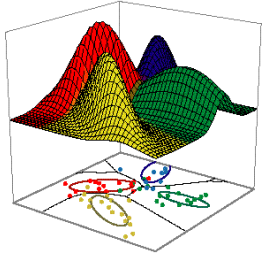
See for example: Ancona et al, "On the statistical assessment of classifiers using DNA microarray data" (2006) *BMC Bioinf*ormatics

Comparing methods over multiple datasets



- Can use paired-t-test to compare 2 methods over many test sets:
  - Run each classifier over a number of test sets
  - H_0: choice of classifier is independent of accuracy
  - Use t-test due to small sample; Why use paired test?
- Or use permutation test to swap scores between classifier randomly
- Or bootstrap by sampling pairs of scores with replacement, then shift distribution to match null hypothesis
  - More later…

Comparing methods over 1 dataset



- 1) Estimate classification accuracy of 2 methods repeatedly using resampling
  - e.g. use repeated 2-CV, or LOO over bootstrapped samples
    - Benefit of LOO is that all of the n classifiers used for estimated performance are highly similar
      → reduced variance of estimated performance
  - Estimated accuracy is mean of accuracy observed over all folds/LOO test sets
- 2) Get variance of estimated error rate

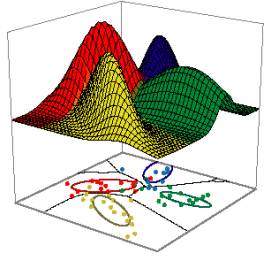
$$Var_{jack}[\hat{\mu}] = \frac{n-1}{n}\sum_{i=1}^{n}\left[\hat{\mu}_{(i)} - \hat{\mu}_{(\cdot)}\right]^2$$

$\hat{\mu}_{(i)} = i^{th}$ estimate of μ

$\hat{\mu}_{(\cdot)} =$ ensemble estimate of μ

- 3) Apply hypothesis testing given mean and variance of accuracy of each classifier
  - Try to reject the null hypothesis that distributions are the same
  - 2-sample t-test or randomization or bootstrap

Comparing methods



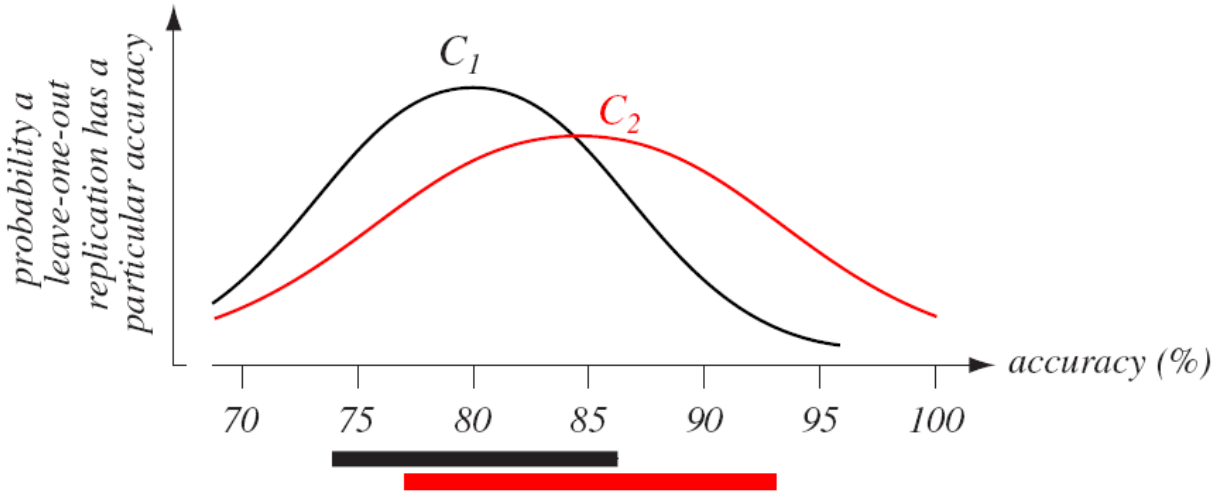
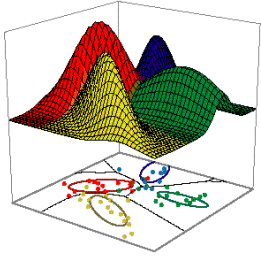


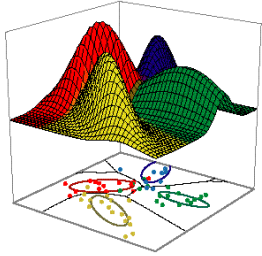
FIGURE 9.11. Jackknife estimation can be used to compare the accuracies of classifiers. The jackknife estimate of classifiers C_1 and C_2 are 80% and 85%, and full widths (twice the square root of the jackknife estimate of the variances) are 12% and 15%, as shown by the bars at the bottom. In this case, traditional hypothesis testing could show that the difference is not statistically significant at some confidence level. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Hypothesis Testing



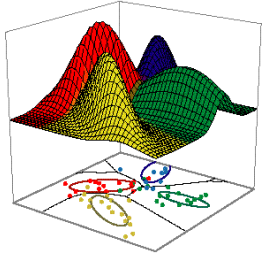
- 1) Assume a null hypothesis, H_0
  - e.g., there is no difference between two classifier accuracies
  - Typically have H_1 which is hypothesis we really want to show
- 2) Run an experiment to find the sample statistic, Δ
  - e.g., run each classifier over 10 problems and compute difference in 2 means. Here $\Delta=\mu_1-\mu_2$
- 3) Find the sampling distribution of Δ under the null hypothesis
  - 2 Fundamental approaches:
    - Statistical tests
      - transform Δ to a test statistic with known distribution
      - Look up critical value for test statistic
    - Randomization/bootstrapping
      - Makes no assumptions about distribution of test statistic
      - Repeatedly randomize data (following null hypothesis) and recompute Δ^*
      - Form distribution of Δ^*
  - e.g., pool 20 accuracies and repeated draw S^*_1 and S^*_2 from pool to create sample distribution of Δ^*.
- 4) If the probability of observing Δ is very low under the null hypothesis, reject H_0
  - The p-value = the probability of incorrectly rejecting H_0

Hypothesis Testing – Statistical Tests



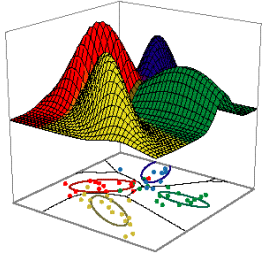
- Populations have *parameters*
  - Constants, may not be known
- Samples have *statistics*
- *Statistical tests* transform statistics like Δ into standard error (s.e.) units.
  - "The standard error is the standard deviation of the sampling distribution of a statistic" Wikipedia
  - It is then trivial to find the region of the distribution bounded by *k* standard error units.

Hypothesis Testing



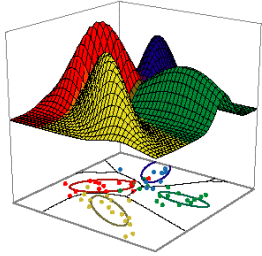
- Will look at two classes of hypotheses:
  - 1-sample tests
    - e.g. Compare sample mean to some fixed value (often 0)
    - e.g. Compare sample mean to population mean
  - 2-sample tests
    - e.g. Compare two samples to test if drawn from same distribution
    - e.g. Is there a significant difference between my two (sets of) samples?
    - e.g. Is there a significant difference between the inter-quartile range of my two samples?

Hypothesis Testing – Statistical Tests



- Specific statistical tests
  - 1-sample tests
    - Z-test
      - Assumes distribution of test statistic is normal or that sample size is large
        - → becomes normal due to Central Limit Theorem.
    - t-test
      - Use for smaller sample sizes (heavier tails than normal)
  - 2-sample tests
    - 2-sample t-test, (matched) paired t-test
    - F-test to compare variances

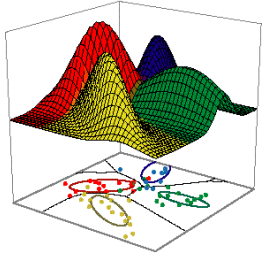
Tests of hypotheses about the mean



- Z test
  - Recall Central Limit Theorem
    - The sampling distribution of the mean of samples of size N approaches a normal distribution as N → ∞.
    - If samples drawn from a population with mean μ and standard deviation σ, then the mean of the sampling distribution is μ and its standard deviation is $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}$

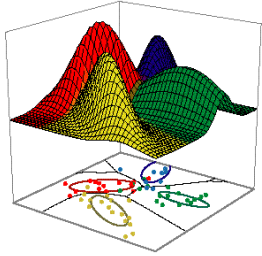
      This is the s.d. of our sample mean not the s.d. of the data itself.
    - **These statements hold irrespective of the shape of the original distribution.**
    - The Z-test uses this fact to test whether a sample mean is significantly different from a population mean.
  - Compute $Z = \frac{\bar{x} - \mu}{\sigma_{\bar{x}}}$
    - Z is distributed as N(0,1)

Tests of hypotheses about the mean



- Z test
  - 1-tailed test: if you are testing if μ is *significantly higher*, or *significantly lower* than $\overline{x}$
    - Alternate hypothesis $H_1 : \overline{x} > \mu$ OR $H_1 : \overline{x} < \mu$
    - Mass to the left or right of Z must be greater than some critical value.
  - 2-tailed test: if you are testing if μ is *significantly different* than $\overline{x}$ (i.e., could be either higher or lower)
    - Alternate hypothesis is simply $H_1 : \overline{x} \neq \mu$
    - Sum of mass both to the left of –Z and to the right of +Z must be greater than critical value. (more stringent test)

Z-test example



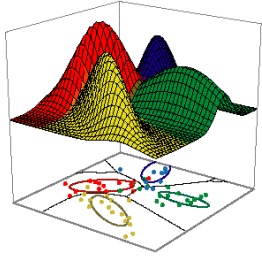
- Sample N=25 of SYSC5405 students has mean IQ m=135. Are they 'smarter than average'?
- Population mean is μ=100 with σ=15 *What is H_1?*
- H_0 is that SYSC5405 students are 'average'
  - i.e. mean IQ of the *population* of SYSC5405 students is 100.
- What is probability, p, of drawing the sample if H_0 were true? If p is small, H_0 is probably false.
- Find the sampling distribution of the mean of a sample of size 25, from a population with mean 100

$$Z = \frac{\bar{x} - \mu}{\sigma_{\bar{x}}} = \frac{(135 - 100)}{\frac{15}{\sqrt{25}}} = 11.67$$

p=Pr(Z>11.67)~0

Why 1-tailed test used?

Tests of hypotheses about the mean

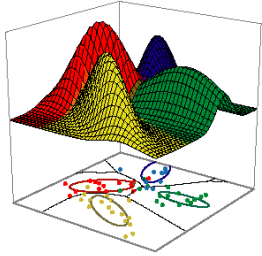


- Student's t-test
  - Same logic as Z test
    - Use when sample sizes are small or when population σ is unknown (estimate $\sigma_{\bar{x}} = \frac{s}{\sqrt{N}}$ instead of $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}$)
  - Sampling distribution is t, not normal. Has heavier tails.
    - Approaches normal as sample size increases
    - Heavier tails reflect under-estimation of population variance due to small sample size
  - Consult tables for t distribution.
  - Must also calculate degrees of freedom
  - t-test is available is several varieties including (matched) paired-t-test
  - Underlying assumption that population ~ Normal

s=sample s.d. (above)

Why do we require this assumption here?

t-test example



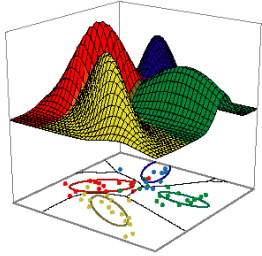
- Sample N=5 of SYSC5405 students has mean IQ m=135 and std=27. Are they 'smarter than average'?
- Population mean is μ=100
- H_0 is that SYSC5405 students are 'average'
  - i.e. mean IQ of the *population* of SYSC5405 students is 100.
- What is probability, p, of drawing the sample if H_0 were true? If p is small, H_0 is probably false.
- Find the sampling distribution of the mean of a sample of size 5, from a population with mean 100

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{N}}} = \frac{(135 - 100)}{\frac{27}{\sqrt{5}}} = \frac{35}{12.1} = 2.89$$

p=Pr(t>2.89)=?

d.f.=(N-1)=4

2-sample statistical tests



- matched pairs t-test
  - When the samples x_1 and x_2 are dependent (e.g., same individuals measured before/after treatment), calculate difference between each pair. Then treat as 1-sample t test.
  - e.g.

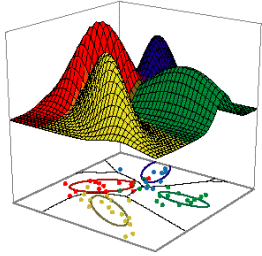
Before	After	Diff
10	11	-1
0	3	-3
60	65	-5
27	31	-4

Mean diff=-13/4=-3.25

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{N}}} = \frac{-3.25 - 0}{\frac{1.71}{\sqrt{4}}} = 3.81$$

p=0.03

2-sample statistical tests

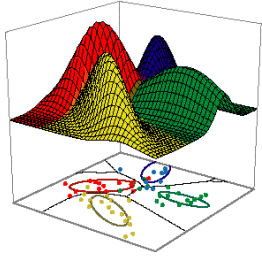


- 2-sample t test (samples independent)
  - Given mean and s.d. of two samples, test whether the are significantly different.
  - H_0: both samples were drawn from a single population.
  - Estimate s.d. of difference in means:

$$\sigma_{(\bar{x}_1-\bar{x}_2)} = \sqrt{\frac{(N_1-1)s_1^2+(N_2-1)s_2^2}{N_1+N_2-2}\left(\frac{1}{N_1}+\frac{1}{N_2}\right)}$$

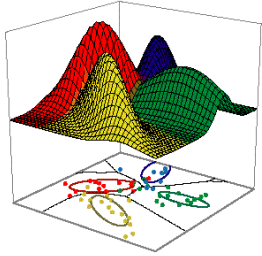
d.f. = N_1+N_2-2. Assumes $\sigma_{1=}\ \sigma_2$ otherwise use a different form. See http://en.wikipedia.org/wiki/T_test or other stats resource.

Hypothesis Testing – Randomization/bootstrapping



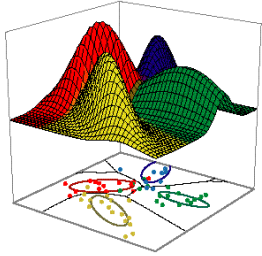
- Construct sampling distributions by simulating on a computer the process of drawing samples (of your test statistic)
  - Use Monte Carlo when one knows population parameters
  - Use bootstrapping when one does not
    - Assumes sample is representative of population
  - Use randomization since it assumes nothing about the population

Hypothesis Testing – Monte Carlo



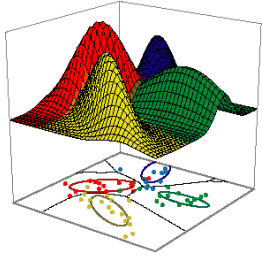
- If you know the population parameters, draw sample from the population
- Estimate test statistic population distribution by repeatedly creating samples of size n by randomly drawing from the population.
  - Works with any test statistic.
- e.g., Determine if your sample of 25 graduate students have a significantly higher inter-quartile range of IQ than do the general population of graduate students. We know the general population has IQ ~ N(100,10).
  - Repeatedly draw samples of size 25 from N(100,10)
  - Compute inter-quartile range on each
  - Determine how extreme observed inter-quartile range for your original sample → p-value.

Hypothesis Testing – Bootstrapping



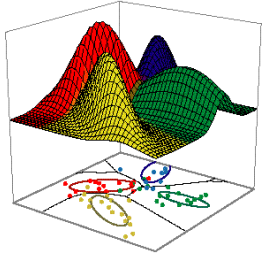
- If you don't know the population parameters, draw samples from your sample!
- Estimate test statistic population distribution by repeatedly creating samples of size N by randomly drawing from the original sample (of size N) with replacement.
  - Makes no assumptions about distribution of overall population
    - Does assume that your sample is representative of the overall population of the test statistic.
  - Works with any test statistic.

Hypothesis Testing – Bootstrapping



- 1-sample test
  - Draw repeated samples S^* of size N with replacement
  - Compute test statistic on each pseudo-sample → dist of test stat
  - Must now correct distribution since we have not yet enforced null hypothesis. 2 options:
    - 1) Calculate μ and σ from bootstrap dist. Assume H_0 dist is normal with ~N(μ_C, σ), where μ_C is mean under null hypothesis. Apply Z-test.
    - 2) Keep shape of bootstrap dist, but shift to have H_0 mean.

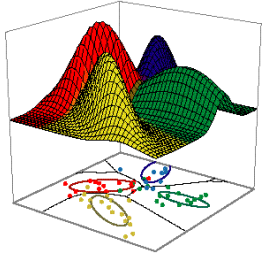
Hypothesis Testing – Bootstrapping



- 2-sample test (Bootstrapping with randomization)
  - Shuffle the elements of both samples S_1 & S_2 together into S.
  - Resample (with replacement) $S^*_{1\ \&}\ S^*_2$ from S.
  - Calculate test statistic for new samples forming a sampling distribution of pseudostatistics
  - Gives us a null hypothesis distribution of test stat.

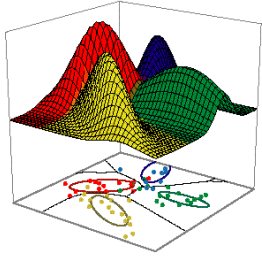
  - Note: for paired tests (where two samples are not independent), resample PAIRS, not individual points.
    - Still need to enforce H0
      - Can randomly swap elements within pairs in each pseudosample to enforce H_0
      - Can compute test statistic distribution for all pseudosamples, and then shift distribution to enforce H0

Hypothesis Testing – Randomization



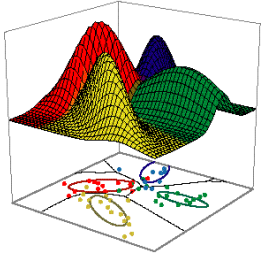
- Does not make any assumptions about your sample being representative of population.
  - Draw samples without replacement
- 2-sample test (independent samples)
  - Shuffle the elements of both samples S_1 & S_2 together into S.
  - Resample (without replacement) $S^*_{1\ \&}\ S^*_2$ from S.
  - Calculate test statistic for new samples forming a sampling distribution of pseudostatistics
    - Can be difference of means, ratio of variances, anything!
  - Gives us a null hypothesis distribution of the test stat.
  - 'Exact randomization' enumerates every possible permutation of the data.
- 2-sample test (paired; dependency between samples)
  - For matched/paired test, shuffle among pairs (switch values within a pair).

Hypothesis Testing – Randomization



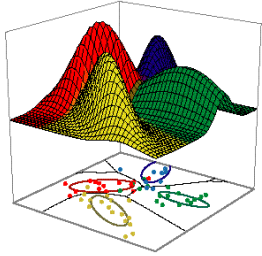
- Example:
  - Four squirrels score 54, 66, 64, and 61.
  - Six chipmunks score 23, 28, 27, 31, 51, 32.
  - Is score independent of species?
  - f = difference of means of chipmunk's and squirrel's scores = 29.25
  - Under H_0 of no association between species and score, the score 54 may equally well have been achieved by a squirrel or a chipmunk.
  - Toss all scores in a hopper, draw four scores at random without replacement, call them squirrel*, call the rest chipmunk*, and calculate f* the difference of means of squirrel* and chipmunk*.
  - Repeat to get a distribution of f*.
  - This is estimate of the sampling distribution of f under H_0: no difference between chipmunk and squirrel scores.

Statistical Tests vs. Bootstrapping vs. Randomization



- When underlying assumptions about population are violated, statistical tests will fail (although they are somewhat robust)
- Randomizations will not perform worse than statistical tests, and we don't need to worry about parametric assumptions.
  - As good as statistical tests when assumptions are met, and better when they are violated.
- Bootstrapping assumes that the sample is representative of the population. For example, the relative frequencies of values in the sample is assumed to mirror the true population.
  - e.g. S={1,2,5,1,3,1}. Prob of drawing a 1 is 0.5. With bootstrapping, we sample with replacement so prob of drawing a 1 each time is 0.5. With randomization, prob of drawing a 1 changes with each draw.
    - Boostrap may draw sample {1,1,1,1,1,1}. Randomization never will.
    - Bootstrapping results in longer tails due to these extreme values.
      - Statistical tests outperform bootstrapping when assumptions are met.
  - Bootsrapping simulates drawing from a population, randomization does not. Therefore, only bootstrapping can lead to confidence intervals or parameter estimation.

Critical assessment of reported results in the literature



- Look for independent test set
  - Test set absent?
  - Repeated or closely related patterns between train and test?
  - Size of test set → confidence interval
  - Did they compare against a baseline method (e.g. random, naïve Bayes) and/or against a reasonable contemporary?
- Watch for methodological errors:
  - Train on test – did they optimize parameters over test set?
  - Test on train – are they reporting apparent error as true error?
- If resampling was used to estimate error, can duplicate experiment over their data
  - Otherwise hard to replicate exact results without knowing the specific train/test split used in their study