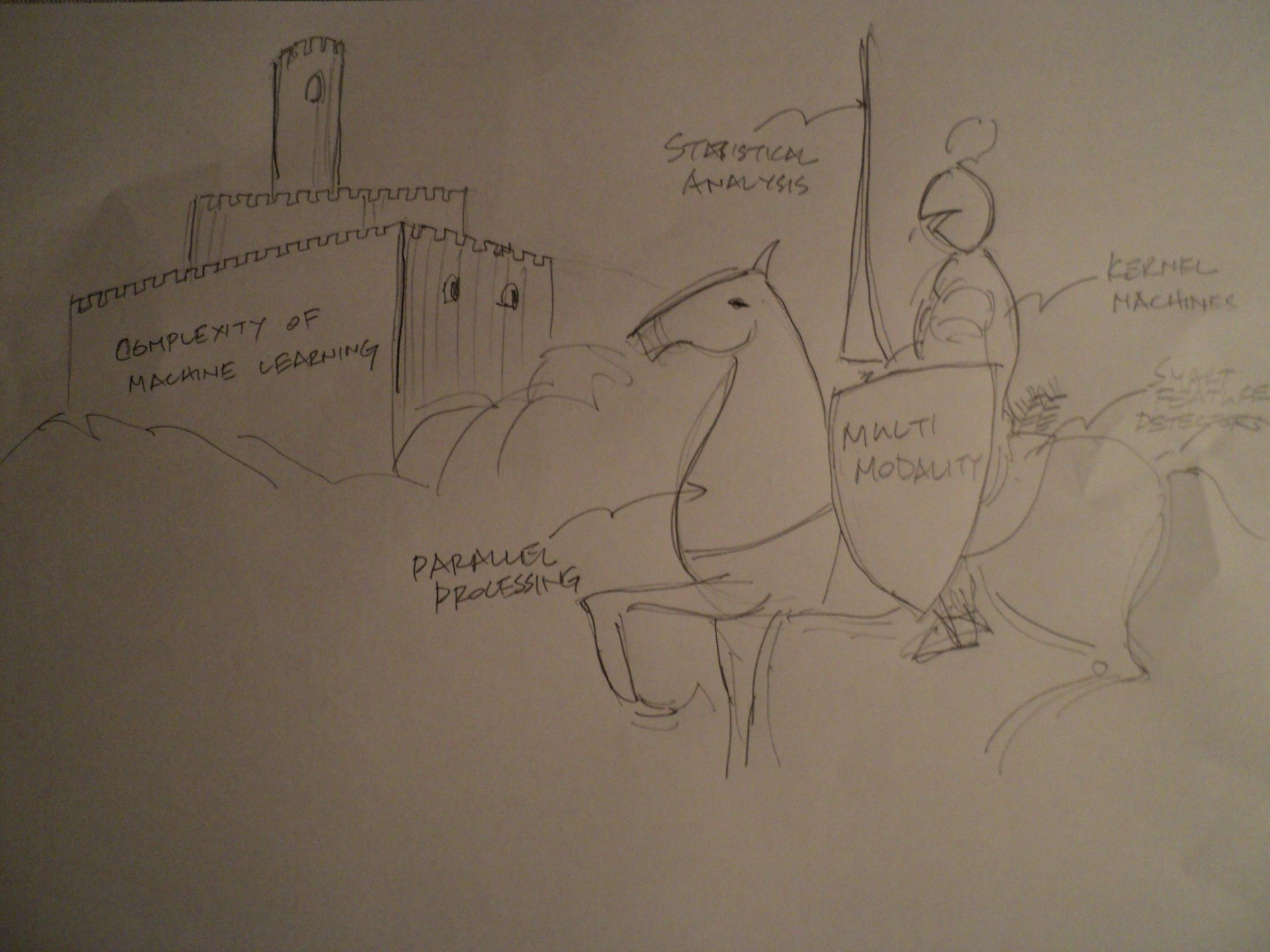
Design and Analysis of Machine Learning Experiments

Ethem Alpaydın
Boğaziçi University, Istanbul

AERFAISS 2010


COMPLEXITY OF MACHINE LEARNING
STATISTICAL ANALYSIS
KERNEL MACHINES
MULTI MODALITY
PARALLEL PROCESSING

Introduction

- Questions:
  - Is the error rate of my classifier less than 2%?
  - Is *k*-NN more accurate than MLP?
  - Does having PCA before improve accuracy?
  - Which kernel leads to highest accuracy with SVM?

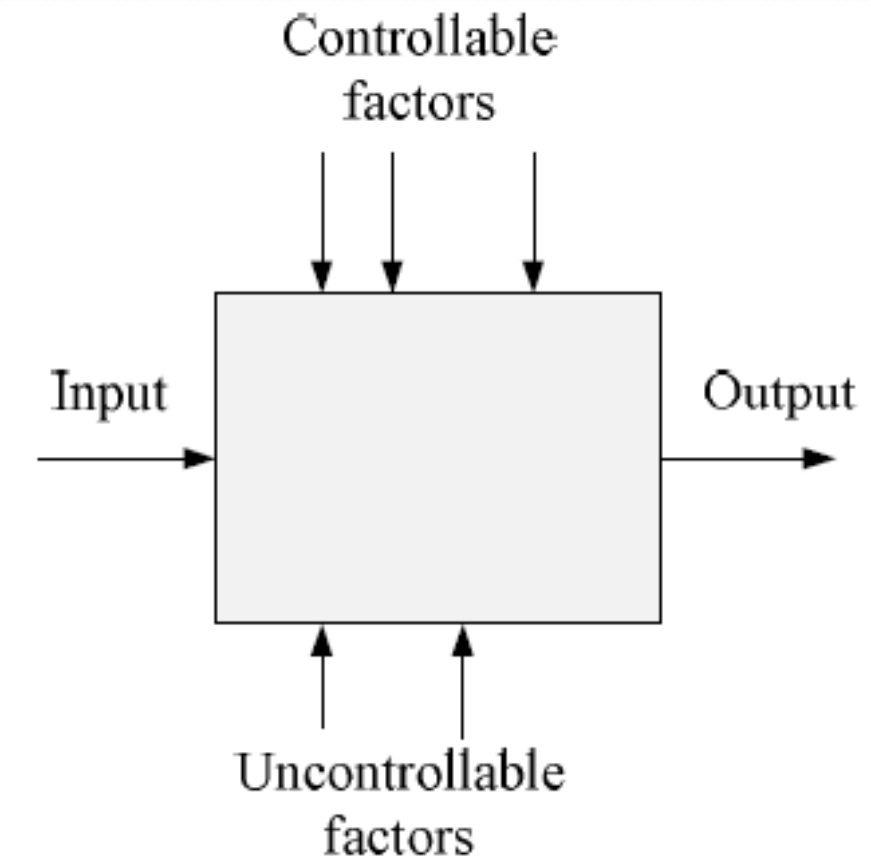
Material

- Training/validation/test sets
- Resampling methods
- Comparing multiple algorithms on a single data set
- Comparison on multiple data sets

Algorithm Preference

- Criteria (Application-dependent):
  - Misclassification error, or risk (loss functions)
  - Training time/space complexity
  - Testing time/space complexity
  - Interpretability
  - Easy programmability
- Cost-sensitive learning

Experiment Design: Factors and Response



Controllable factors:

-Learning algorithm

-Hyperparameters

-Input representation

Uncontrollable factors:

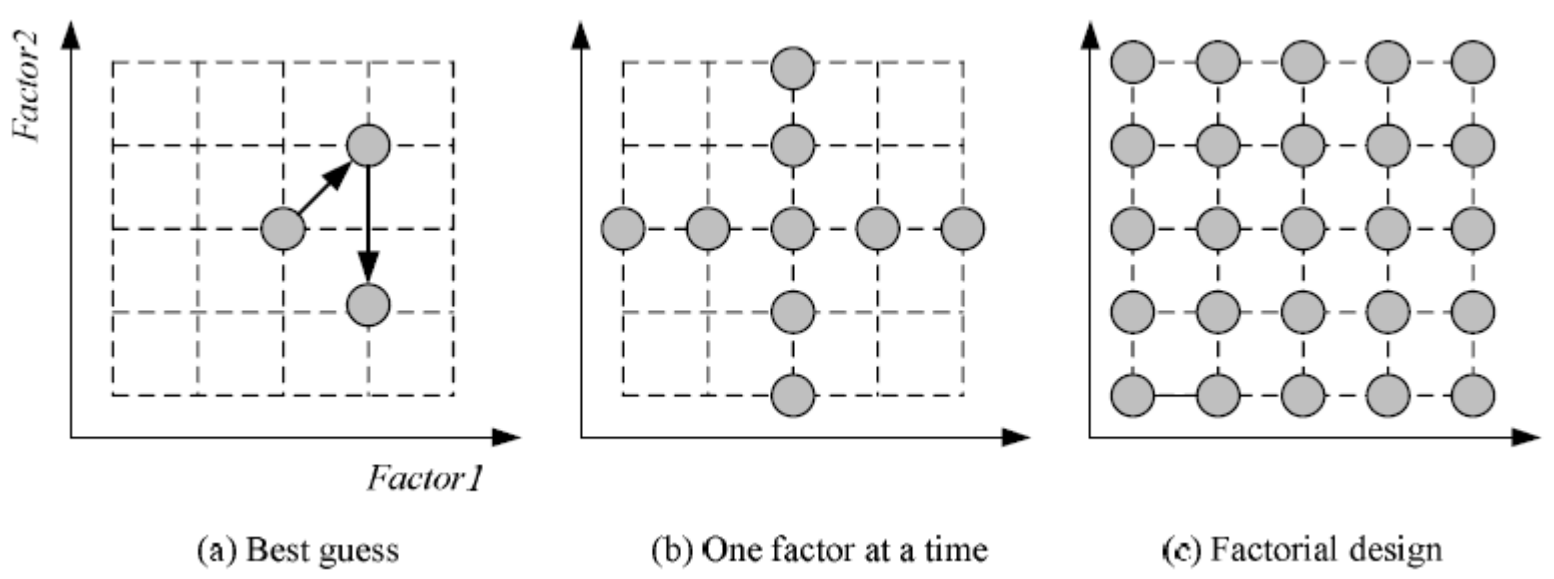
-Noise in data

-Randomness in splitting

-Randomness in optimization

Arrive to conclusions not affected by chance, i.e., statistically significant.

Strategies of Experimentation



Response surface design

Basic Principles of Experimental Design

1. Randomization: Independence of results, unaffected by order
2. Replication: Average over chance and uncontrollable factors (k-fold cv)
3. Blocking: Reduce or eliminate the variability due to nuisance factors: Paired tests

Guidelines for ML experiments

A. Aim of the study:

Compare hyperparameters or two or more algorithms

Single/multiple data sets

B. Selection of the response variable

Accuracy/precision-recall/loss function

Cost-conscious framework

C. Choice of factors and levels

What are the factors to be played with?

What are the factor levels?

Guidelines (cont'd)

D. Choice of experimental design
   - Factorial design (grid search)
   - How many replicates?

E. Performing the experiment
   - Unbiased in experimentation, a separate tester
   - Good code and documentation

F. Statistical Analysis of the Data
   - Hypothesis testing
   - Visualization of results: Histograms, plots

G. Conclusions and Recommendations
   - Draw objective conclusions

Splitting Data

- The need for training, validation, and test sets
  - Training set: Optimize parameters
  - Validation set: Optimize hyperparameters
  - Test set: Measure generalization performance
- Use data once.

Resampling and K-Fold Cross-Validation

- The need for multiple training/validation sets $\{X_i,V_i\}_i$: Training/validation sets of fold i
- Stratification
- K-fold cross-validation: Divide X into k, $X_i, i=1,\ldots,K$

$$\mathcal{V}_1 = \mathcal{X}_1 \quad \mathcal{T}_1 = \mathcal{X}_2 \cup \mathcal{X}_3 \cup \cdots \cup \mathcal{X}_K$$
$$\mathcal{V}_2 = \mathcal{X}_2 \quad \mathcal{T}_2 = \mathcal{X}_1 \cup \mathcal{X}_3 \cup \cdots \cup \mathcal{X}_K$$
$$\vdots$$
$$\mathcal{V}_K = \mathcal{X}_K \quad \mathcal{T}_K = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_{K-1}$$

- T_i share K-2 parts

5×2 Cross-Validation

(Dietterich, 1998, Neural Computation)

$$\mathcal{T}_1 = \mathcal{X}_1^{(1)} \quad \mathcal{V}_1 = \mathcal{X}_1^{(2)}$$
$$\mathcal{T}_2 = \mathcal{X}_1^{(2)} \quad \mathcal{V}_2 = \mathcal{X}_1^{(1)}$$

$$\mathcal{T}_3 = \mathcal{X}_2^{(1)} \quad \mathcal{V}_3 = \mathcal{X}_2^{(2)}$$
$$\mathcal{T}_4 = \mathcal{X}_2^{(2)} \quad \mathcal{V}_4 = \mathcal{X}_2^{(1)}$$

$$\vdots$$

$$\mathcal{T}_9 = \mathcal{X}_5^{(1)} \quad \mathcal{V}_9 = \mathcal{X}_5^{(2)}$$
$$\mathcal{T}_{10} = \mathcal{X}_5^{(2)} \quad \mathcal{V}_{10} = \mathcal{X}_5^{(1)}$$

Bootstrapping

- Draw instances from a dataset *with replacement*
- Prob that we do not pick an instance after N draws

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

that is, only 36.8% is new!

Making Decisions and Error

- Classifier predicts + if $P(+|x)>\theta$ and predicts – otherwise

	Prediction	
Truth	+	−
+	0	λ
−	1	0

$$R(+|x) = \lambda_{11}P(+|x) + \lambda_{12}P(-|x) = P(-|x)$$

$$R(-|x) = \lambda P(+|x)$$

predict x as positive if

$$R(+|x) < R(-|x), \text{ or if } P(-|x) < \lambda P(+|x)$$

$$P(+|x) > \frac{1}{1+\lambda}$$

Measures of Performance

2×2 CONFUSION MATRIX

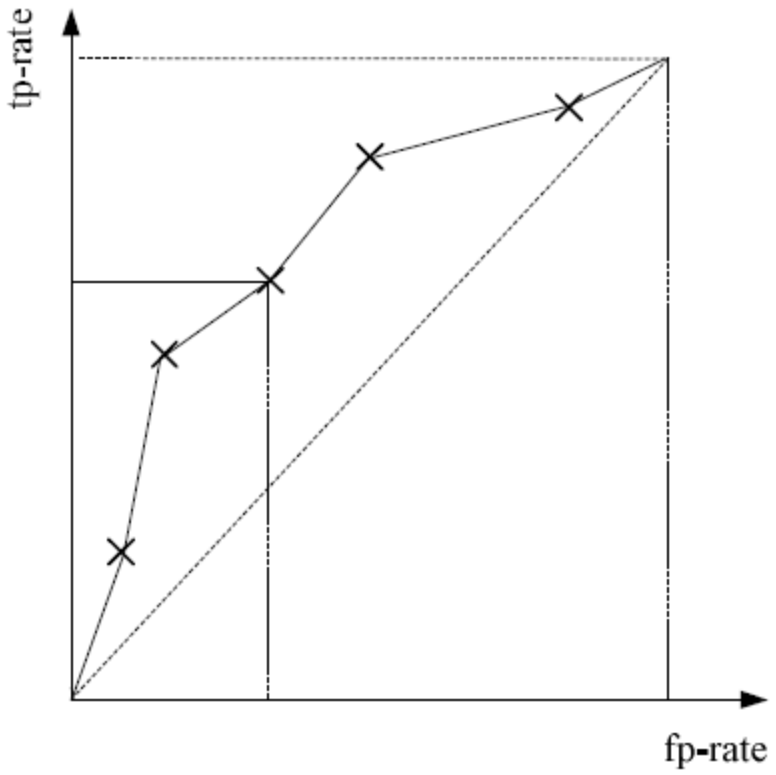
	Prediction		
Truth	+	-	Total
+	tp	fn	p
-	fp	tn	n
Total	p'	n'	N

$$error\ rate = \frac{fp+fn}{N} \qquad accuracy = \frac{tp+tn}{N}$$

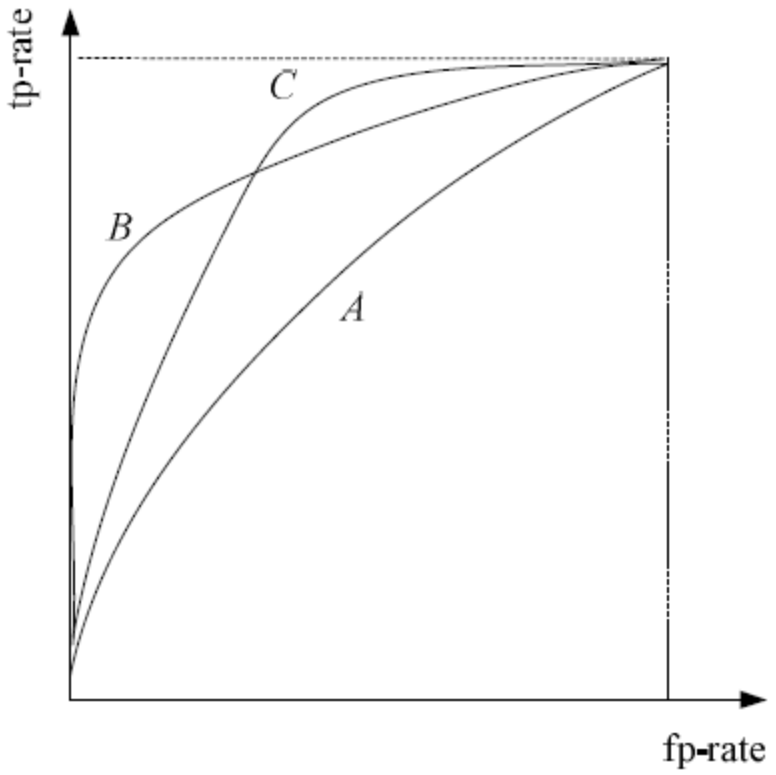
$$tp\text{-}rate = \frac{tp}{p} \qquad fp\text{-}rate = \frac{fp}{n}$$

$$recall = \frac{tp}{p} \qquad precision = \frac{tp}{p'}$$

$$sensitivity = \frac{tp}{p} \qquad specificity = \frac{tn}{n}$$

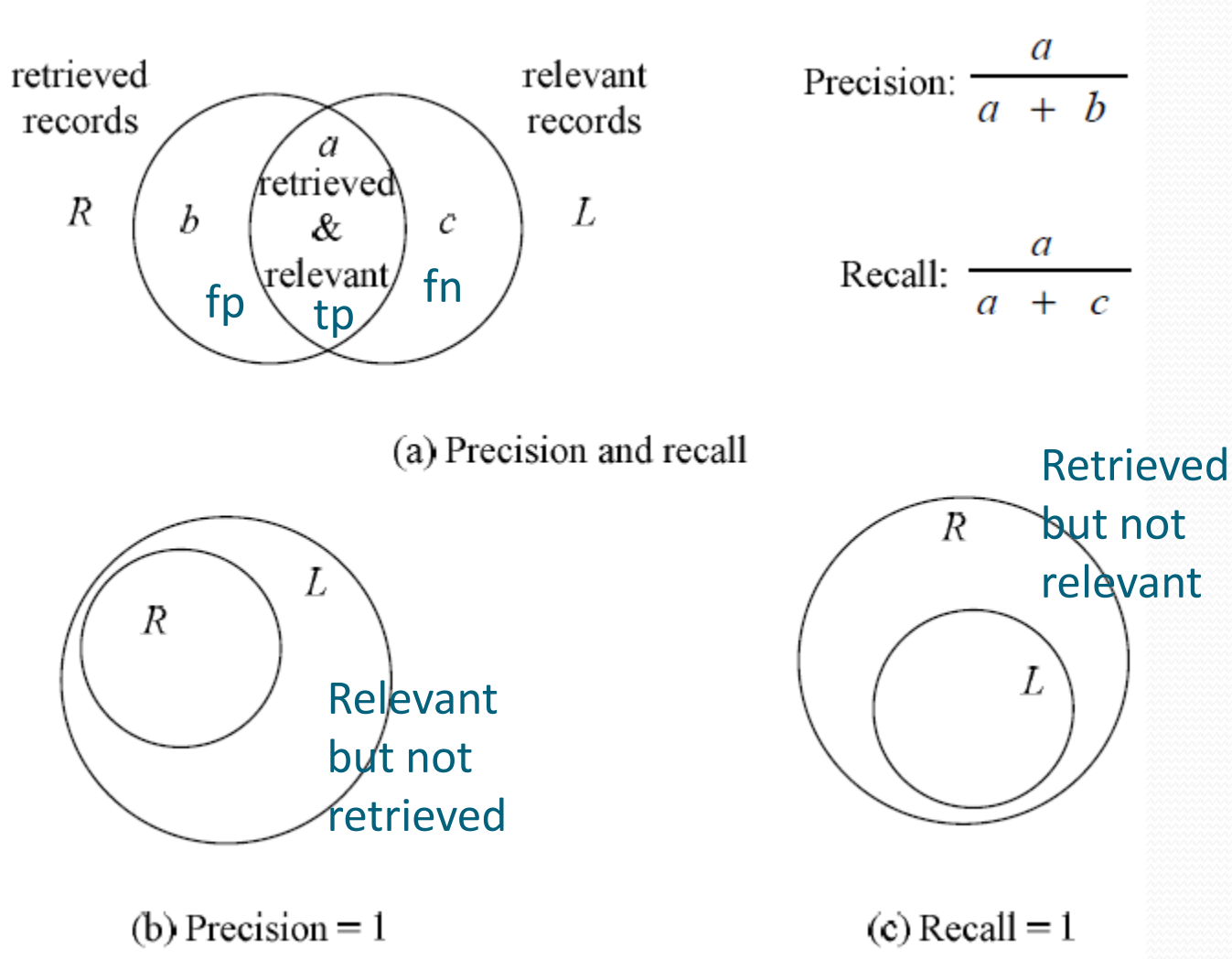


(a) Example ROC curve

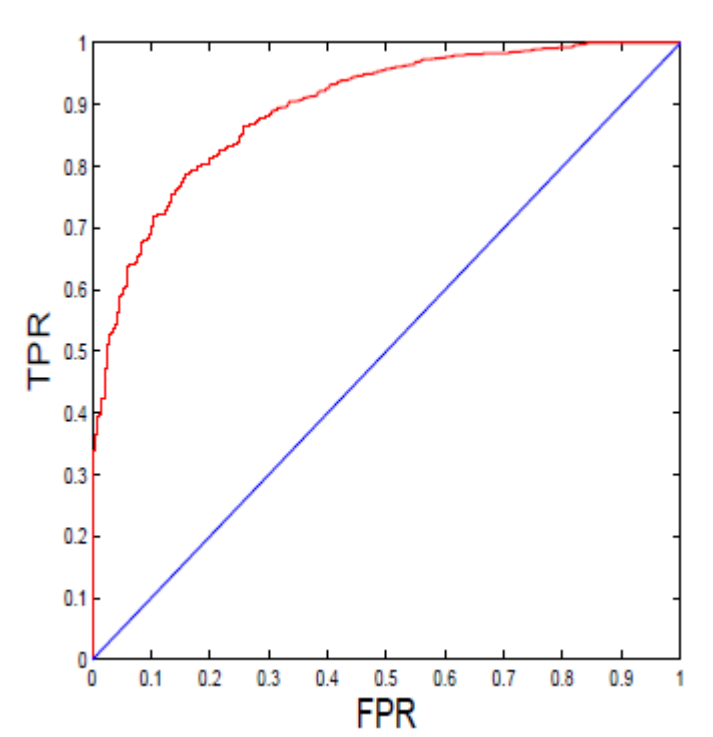


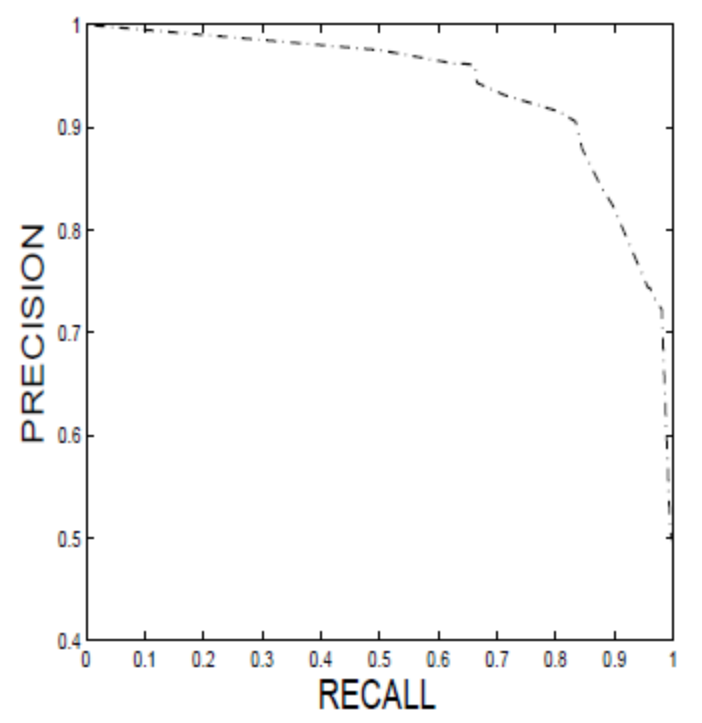
(b) Different ROC curves for different classifiers

Precision and Recall



ROC – Precision/Recall Curves





(a)

	Prediction		
Truth	+	−	Sum
+	80	20	100
−	20	180	200
Sum	100	200	300

tp-rate$= 80/100 = 0.8$

fp-rate$= 20/200 = 0.1$

precision $= 80/100 = 0.8$

recall $= 80/100 = 0.8$

(b)

	Prediction		
Truth	+	−	Sum
+	64	16	80
−	22	198	220
Sum	86	214	300

tp-rate$= 64/80 = 0.8$

fp-rate$= 22/220 = 0.1$

precision $= 64/86 = 0.74$

recall $= 64/80 = 0.8$

(c)

	Prediction		
Truth	+	−	Sum
+	80	20	100
−	20	580	600
Sum	100	600	700

tp-rate$= 80/100 = 0.8$

fp-rate$= 20/600 = 0.03$

precision $= 80/100 = 0.8$

recall $= 80/100 = 0.8$

Statistics Review: Sampling

- $X = \{ x^t \}_t$ where $x^t \sim N (\mu, \sigma^2)$
- $m \sim N (\mu, \sigma^2/N)$

$$m = \frac{\sum_t x^t}{N}$$

- Implication for model combination

$$E[y] = E\left[\sum_j \frac{1}{L} d_j\right] = \frac{1}{L} L E[d_j] = E[d_j]$$

$$\mathrm{Var}(y) = \mathrm{Var}\left(\sum_j \frac{1}{L} d_j\right) = \frac{1}{L^2}\mathrm{Var}\left(\sum_j d_j\right)$$

$$\mathrm{Var}(y) = \frac{1}{L^2}\mathrm{Var}\left(\sum_j d_j\right) = \frac{1}{L^2}\left[\sum_j \mathrm{Var}(d_j) + 2\sum_j \sum_{i<j} \mathrm{Cov}(d_j, d_i)\right]$$

Ulaş et al (2009), Info Sci

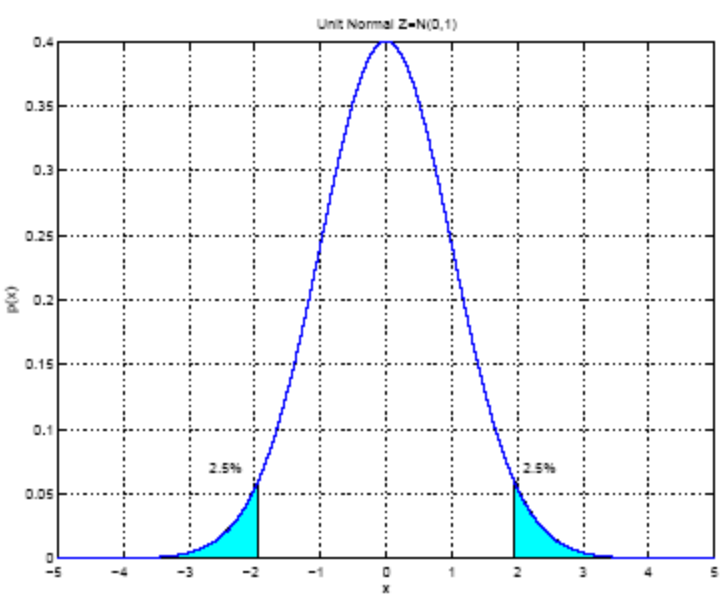
Interval Estimation

$$\sqrt{N}\frac{(m-\mu)}{\sigma} \sim Z$$

$$P\left\{-1.96 < \sqrt{N}\frac{(m-\mu)}{\sigma} < 1.96\right\} = 0.95$$

$$P\left\{m - 1.96\frac{\sigma}{\sqrt{N}} < \mu < m + 1.96\frac{\sigma}{\sqrt{N}}\right\} = 0.95$$

$$P\left\{m - z_{\alpha/2}\frac{\sigma}{\sqrt{N}} < \mu < m + z_{\alpha/2}\frac{\sigma}{\sqrt{N}}\right\} = 1-\alpha$$

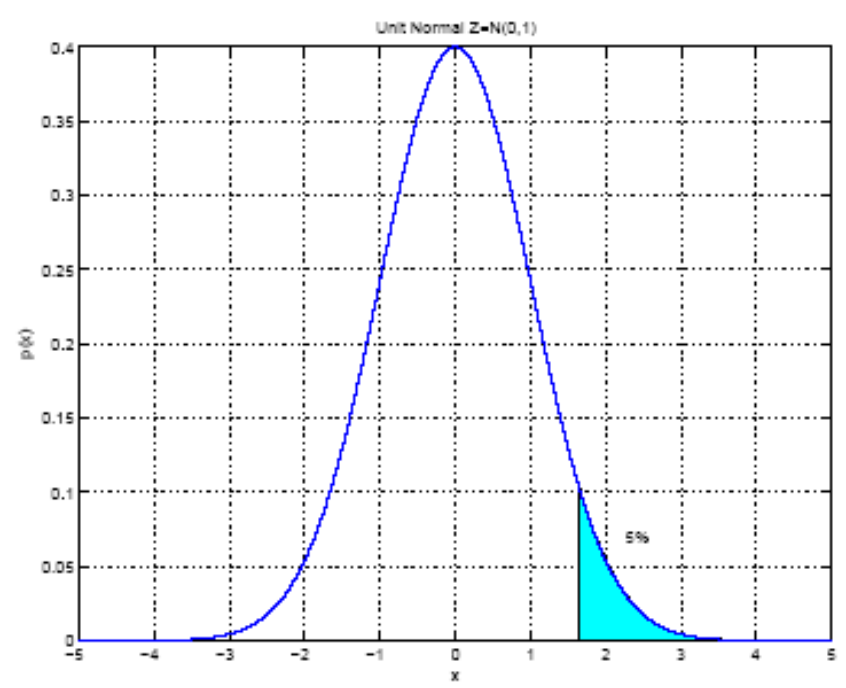


100(1- α) percent confidence interval

$$P\left\{\sqrt{N}\frac{(m-\mu)}{\sigma}<1.64\right\}=0.95$$

$$P\left\{m-1.64\frac{\sigma}{\sqrt{N}}<\mu\right\}=0.95$$

$$P\left\{m-z_{\alpha}\frac{\sigma}{\sqrt{N}}<\mu\right\}=1-\alpha$$



When σ^2 is not known:

$$S^2 = \sum_t \left(x^t - m\right)^2 / (N-1)$$

$$\frac{\sqrt{N}(m-\mu)}{S} \sim t_{N-1}$$

$$P\left\{m - t_{\alpha/2,N-1}\frac{S}{\sqrt{N}} < \mu < m + t_{\alpha/2,N-1}\frac{S}{\sqrt{N}}\right\} = 1-\alpha$$

Hypothesis Testing

- Reject a null hypothesis if not supported by the sample with enough confidence

	Decision	
Truth	Fail to reject	Reject
True	Correct	Type I error
False	Type II error	Correct (power)

- $H_0: \mu = \mu_0$ vs. $H_1: \mu \neq \mu_0$

  Accept H_0 with level of significance α if μ_0 is in the 100(1- α) confidence interval

$$\frac{\sqrt{N}(m-\mu_0)}{\sigma} \in \left(-z_{\alpha/2}, z_{\alpha/2}\right)$$

  Two-sided test

- Type II error

$$\beta(\mu) = P_\mu\left\{-z_{\alpha/2} \leq \frac{m-\mu_0}{\sigma/\sqrt{N}} \leq z_{\alpha/2}\right\}$$

- How large a sample?

- One-sided test: $H_0: \mu \leq \mu_0$ vs. $H_1: \mu > \mu_0$

  Accept if

$$\frac{\sqrt{N}(m-\mu_0)}{\sigma} \in (-\infty, z_\alpha)$$

- Variance unknown: Use t, instead of z

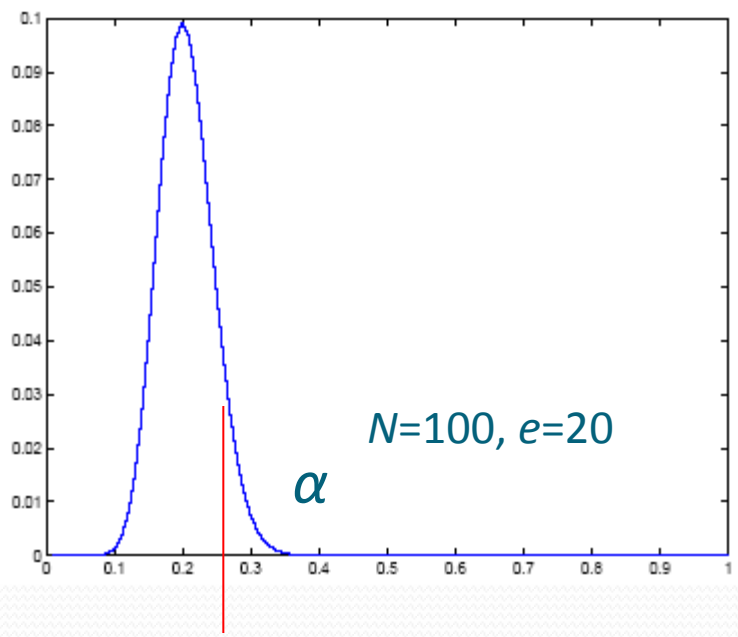
  Accept $H_0: \mu = \mu_0$ if

$$\frac{\sqrt{N}(m-\mu_0)}{S} \in (-t_{\alpha/2,N-1}, t_{\alpha/2,N-1})$$

Assessing Error: $H_0: p \leq p_0$ vs. $H_1: p > p_0$

- Single training/validation set: Binomial Test

  If error prob is p_0, prob that there are e errors or more

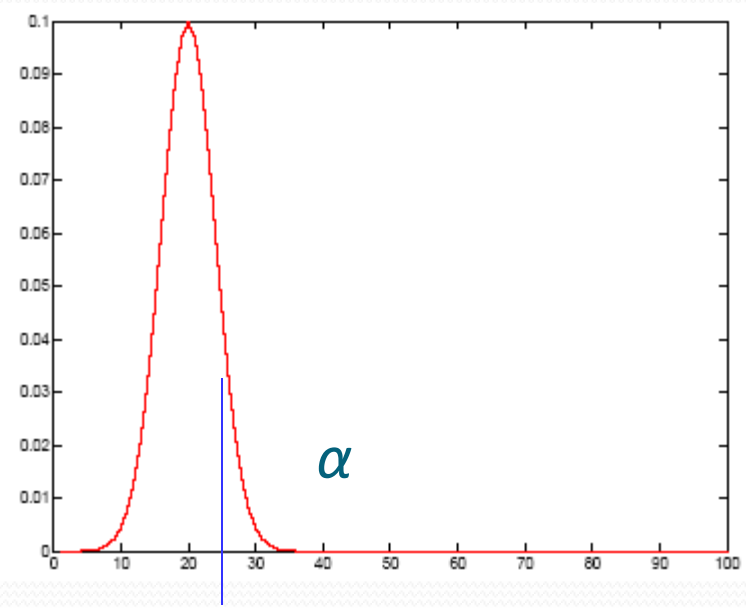


$$P\{X \geq e\} = \sum_{j=e}^{N} \binom{N}{j} p_0^{\,j} \left(1 - p_0^{\,j}\right)^{N-j}$$

Reject if this prob is less than α

Normal Approximation to the Binomial

- $H_0: \mu<\mu_o$ vs. $H_1: \mu>\mu_o$
- Number of errors X is approx Normal (CLT) with mean Np_0 and var $Np_0(1-p_0)$



$$z = \frac{e - Np_0}{\sqrt{Np_0(1-p_0)}} \sim Z$$

Reject if $z > z_\alpha$

t Test

- Multiple training/validation sets
- $x^t_i = 1$ if instance t misclassified on fold i
- Error rate of fold i: $p_i = \frac{\sum_{t=1}^{N} x_i^t}{N}$
- With m and s^2 average and var of p_i , we reject p_0 or less error if

$$\frac{\sqrt{K}(m - p_0)}{S} \sim t_{K-1}$$

is greater than $t_{\alpha,K\text{-}1}$

Comparing Classifiers:

$H_0: \mu_1 = \mu_2$ vs. $H_1: \mu_1 \neq \mu_2$

- Single training/validation set: McNemar's Test

e_{00}: Number of examples misclassified by both	e_{01}: Number of examples misclassified by 1 but not 2
e_{10}: Number of examples misclassified by 2 but not 1	e_{11}: Number of examples correctly classified by both

- Under H_0, we expect $e_{01} = e_{10} = (e_{01} + e_{10})/2$

$$\frac{\left(|e_{01} - e_{10}| - 1\right)^2}{e_{01} + e_{10}} \sim \chi_1^2$$

Accept if $< X^2_{\alpha,1}$

K-Fold CV Paired *t* Test

- Use *K*-fold cv to get *K* training/validation folds
- p_i^1, p_i^2: Errors of classifiers 1 and 2 on fold *i*
- $p_i = p_i^1 - p_i^2$: Paired difference on fold *i*
- The null hypothesis is whether p_i has mean 0

$$H_0 : \mu = 0 \quad \text{vs.} \quad H_0 : \mu \neq 0$$

$$m = \frac{\sum_{i=1}^{K} p_i}{K} \qquad s^2 = \frac{\sum_{i=1}^{K} (p_i - m)^2}{K-1}$$

$$\frac{\sqrt{K}(m-0)}{s} = \frac{\sqrt{K} \cdot m}{s} \sim t_{K-1} \quad \text{Accept if in} \left(-t_{\alpha/2,K-1}, t_{\alpha/2,K-1}\right)$$

5×2 cv Paired *t* Test
(Dietterich, 1998, Neural Computation)

- Use 5×2 cv to get 2 folds of 5 tra/val replications
- $p_i^{(j)}$: difference btw errors of 1 and 2 on fold j=1, 2 of replication i=1,...,5

$$\bar{p}_i = \left(p_i^{(1)} + p_i^{(2)}\right)/2 \qquad s_i^2 = \left(p_i^{(1)} - \bar{p}_i\right)^2 + \left(p_i^{(2)} - \bar{p}_i\right)^2$$

$$\frac{p_1^{(1)}}{\sqrt{\sum_{i=1}^{5} s_i^2 / 5}} \sim t_5$$

Two-sided test: Accept H_0: $\mu_0 = \mu_1$ if in $(-t_{\alpha/2,5}, t_{\alpha/2,5})$

One-sided test: Accept H_0: $\mu_0 \leq \mu_1$ if $< t_{\alpha,5}$

5×2 cv Paired *F* Test

(Alpaydın, 1999, Neural Computation)

$$\frac{\sum_{i=1}^{5}\sum_{j=1}^{2}\left(p_i^{(j)}\right)^2}{2\sum_{i=1}^{5}s_i^2} \sim F_{10,5}$$

Two-sided test: Reject H_0: $\mu_0 = \mu_1$ if $> F_{\alpha,10,5}$

Comparing L>2 Algorithms: Analysis of Variance (Anova)

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_L$$

- Errors of L algorithms on K folds

$$X_{ij} \sim \mathcal{N}\left(\mu_j, \sigma^2\right), j = 1,\ldots,L,\ i = 1,\ldots,K$$

- We construct two estimators to σ^2 .

  One is valid if H_0 is true, the other is always valid.

  We reject H_0 if the two estimators disagree.

If H_0 is true:

$$m_j = \sum_{i=1}^{K} \frac{X_{ij}}{K} \sim \mathcal{N}\left(\mu, \sigma^2 / K\right)$$

$$m = \frac{\sum_{j=1}^{L} m_j}{L} \qquad S^2 = \frac{\sum_j \left(m_j - m\right)^2}{L-1}$$

Thus an estimator of σ^2 is $K \cdot S^2$, namely,

$$\hat{\sigma}^2 = K \sum_{j=1}^{L} \frac{\left(m_j - m\right)^2}{L-1}$$

$$\sum_j \frac{\left(m_j - m\right)^2}{\sigma^2 / K} \sim \chi^2_{L-1} \quad SSb \equiv K \sum_j \left(m_j - m\right)^2$$

So when H_0 is true, we have

$$\frac{SSb}{\sigma^2} \sim \chi^2_{L-1}$$

Regardless of H_0 our second estimator to σ^2 is the average of group variances S_j^2:

$$S_j^2 = \frac{\sum_{i=1}^{K}(X_{ij} - m_j)^2}{K-1} \qquad \hat{\sigma}^2 = \sum_{j=1}^{L}\frac{S_j^2}{L} = \sum_j\sum_i \frac{(X_{ij} - m_j)^2}{L(K-1)}$$

$$SSw \equiv \sum_j\sum_i (X_{ij} - m_j)^2$$

$$(K-1)\frac{S_j^2}{\sigma^2} \sim \chi^2_{K-1} \qquad \frac{SSw}{\sigma^2} \sim \chi^2_{L(K-1)}$$

$$\left(\frac{SSb/\sigma^2}{L-1}\right) \Big/ \left(\frac{SSw/\sigma^2}{L(K-1)}\right) = \frac{SSb/(L-1)}{SSw/(L(K-1))} \sim F_{L-1,L(K-1)}$$

Reject $H_0 : \mu_1 = \mu_2 = \cdots = \mu_L$ if $> F_{\alpha,L-1,L(K-1)}$

ANOVA table

Source of variation	Sum of squares	Degrees of freedom	Mean square	F_0
Between groups	$SS_b \equiv K\sum_j (m_j - m)^2$	$L-1$	$MS_b = \frac{SS_b}{L-1}$	$\frac{MS_b}{MS_w}$
Within groups	$SS_w \equiv \sum_j \sum_i (X_{ij} - m_j)^2$	$L(K-1)$	$MS_w = \frac{SS_w}{L(K-1)}$	
Total	$SS_T \equiv \sum_j \sum_i (X_{ij} - m)^2$	$L \cdot K - 1$		

If ANOVA rejects, we do pairwise posthoc tests

$$H_0 : \mu_i = \mu_j \text{ vs } H_1 : \mu_i \neq \mu_j$$

$$t = \frac{m_i - m_j}{\sqrt{2}\sigma_w} \sim t_{L(K-1)}$$

More on Comparing Multiple Populations

- Range tests: Newman-Keuls test

$$\underline{5\ 2}\ \underline{4\ 3}\ 1$$

- Contrasts: Check if significant difference between 1,2 and 3,4,5.

$$H_0: (\mu_1 + \mu_2)/2 = (\mu_3 + \mu_4 + \mu_5)/3 \text{ vs.}$$
$$H_1: (\mu_1 + \mu_2)/2 \neq (\mu_3 + \mu_4 + \mu_5)/3$$

MultiTest: Comparison of $L>2$ algorithms (Yıldız and Alpaydın, 2006, IEEE T Pami)

- Generate a full ordering using pairwise tests and prior ordering
- Order algorithms in decreasing order of prior preference (e.g., based on complexity)
- Form a directed graph using pairwise one-sided tests with *i* preferred over *j*

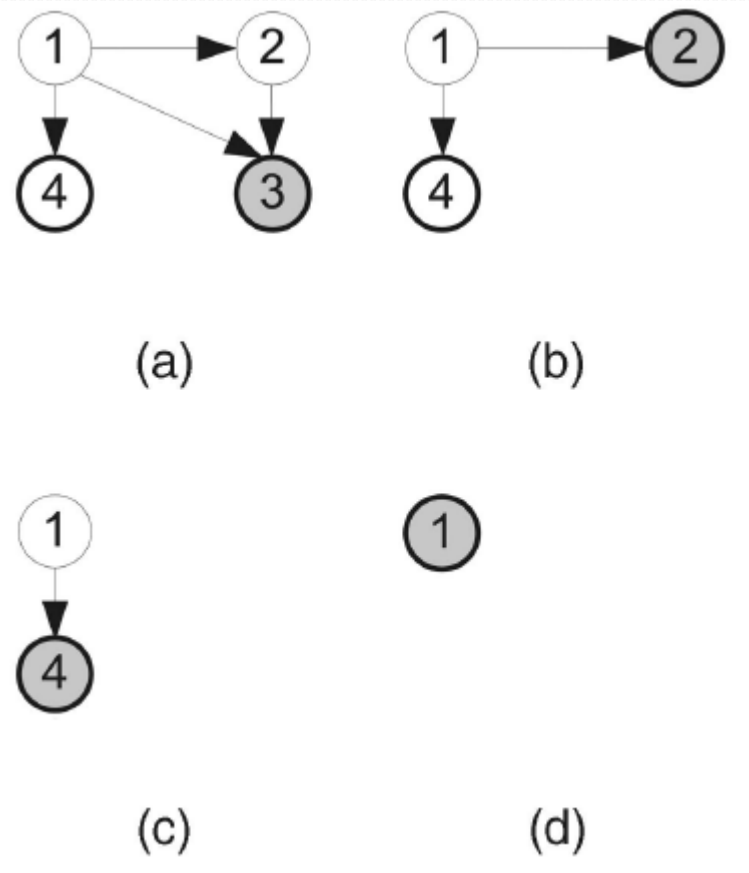
$$H_0 : \mu_i \leq \mu_j$$

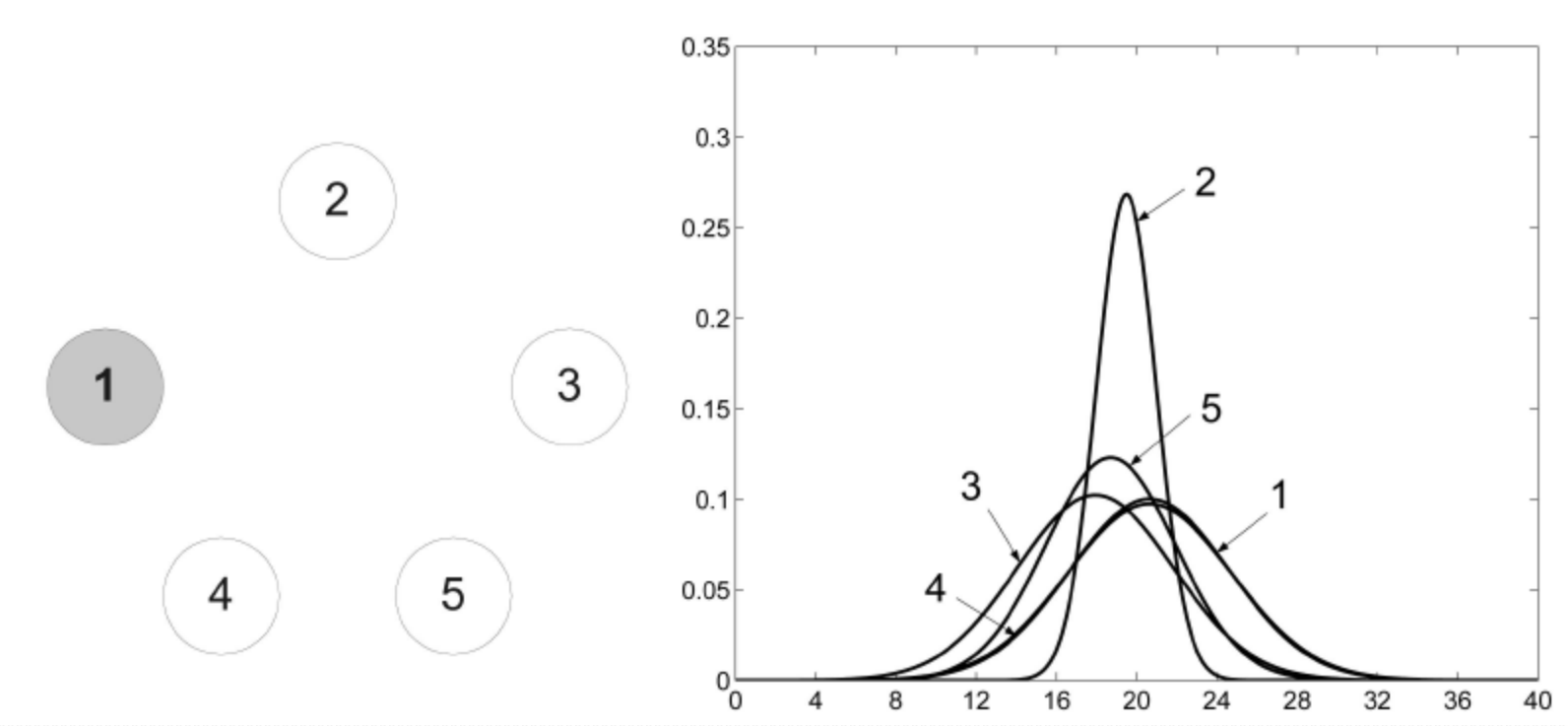
- If the test rejects, we add am edge from *i* to *j*, to show that *j* is to be preferred over *i*.

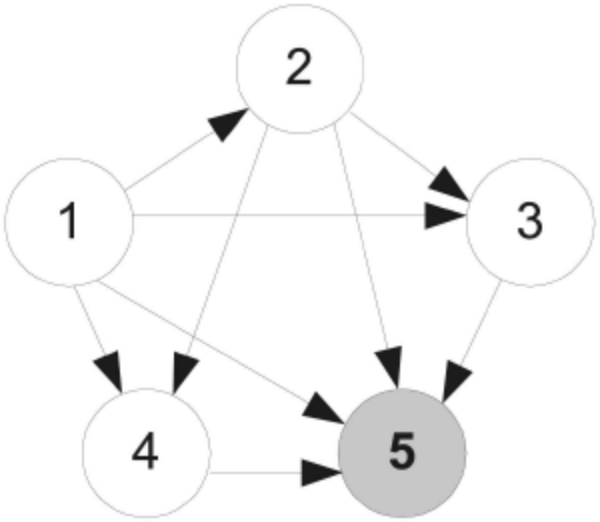
MultiTest: Pseudo-code

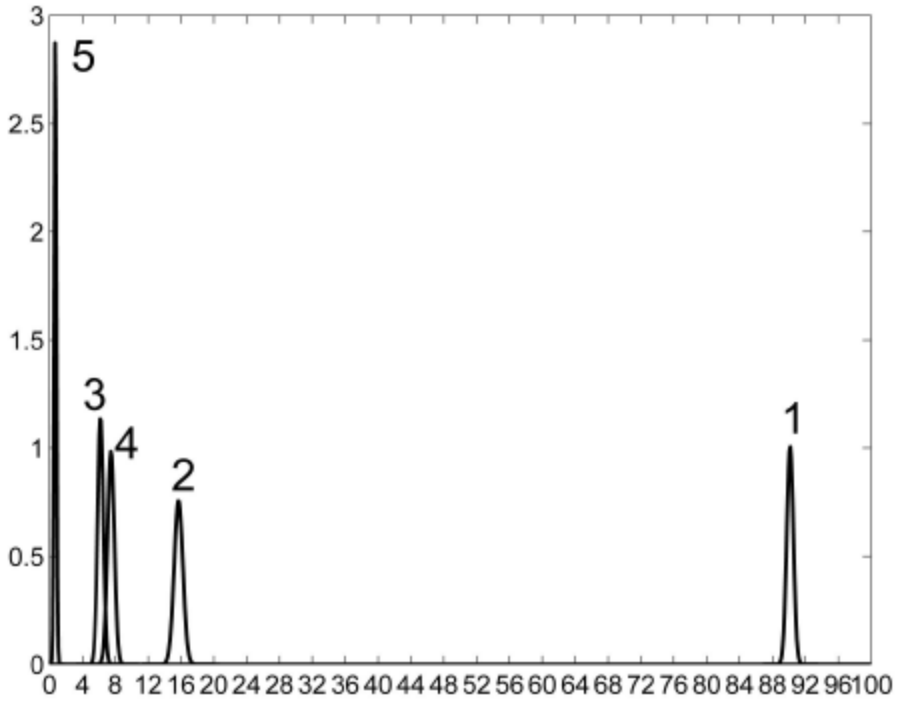
```
1  Best MultiTest(1, ..., K; T; α)
2      E ← ∅ /* Edges of the graph */
3      for i = 1 to K
4          for j = i + 1 to K
5              Test H_0: μ_i ≤ μ_j using T
6              if H_0 rejected E ← E ∪ e[i, j] (α/(K(K − 1)/2)) /* Bonferroni */
7      S_k = {x : ∀e[j, k] ∈ E, j ≠ x}; /* Find the nodes with no outgoing edges */
8      l = ∀j ∈ S_k, l ≤ j; /* Select node l with the lowest index */
9      return l;
```

MultiTest




2
1
3
4
5
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
4
8
12
16
20
24
28
32
36
40
2
5
3
1
4


1
2
3
4
5


5
3
4
2
1
0
0.5
1
1.5
2
2.5
3
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96100

$$5 < 3 < 4 < 2 < 1$$

Nonparametric Tests

- If the normality assumption does not hold, it does not make sense to take or compare averages
- Comparison of training times, memory needs, and so on
- Comparison over multiple data sets
- We can use order and rank information

Sign test

- Comparing two algorithms:

  Sign test: Count how many times *A* beats *B* over *N* datasets, and check if this could have been by chance if A and B did have the same error rate

$$P\{X \leq e\} = \sum_{x=0}^{e} \binom{N}{x} \left(\frac{1}{2}\right)^{x} \left(\frac{1}{2}\right)^{N-x}$$

- Wilcoxon signed rank test

Kruskal-Wallis Test

- Comparing multiple algorithms

  Kruskal-Wallis test: Calculate the average rank of all algorithms on M datasets, and check if these could have been by chance if they all had equal error

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_L$$

$$H = \frac{12}{(M+1)L} \sum_{i=1}^{L} \left(\overline{R}_{i\bullet} - \frac{M+1}{2} \right)$$

If KW rejects, we do pairwise posthoc tests

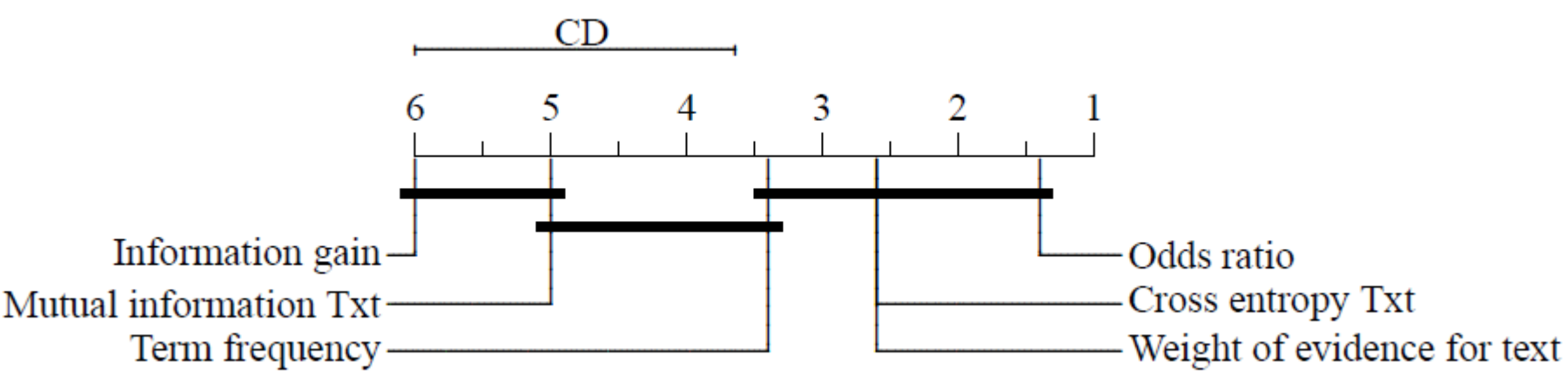
Tukey's test:

$$|\overline{R}_{i\bullet} - \overline{R}_{j\bullet}| > q_\alpha(L, L(K-1))\sigma_w$$

Critical Difference diagrams

(Demsar, 2006, JMLR)

- Friedman's test followed by Nemenyi's posthoc test for pairwise comparisons



Conclusions

- “See first, think later, then test. But always see first. Otherwise you will only see what you were expecting.”
  - Douglas Adams “So long and thanks for all the fish”
- Testing is not a separate step done after all runs are completed, but the whole experimental process should be designed beforehand.

References

- Alpaydın, E. 2010. *Introduction to Machine Learning,* 2nd edition, The MIT Press. This presentation is based on Chapter 19 of this book.
- Demsar, J. 2006. ``Statistical Comparison of Classifiers over Multiple Data Sets.'' *Journal of Machine Learning Research* 7: 1--30.
- Dietterich, T. G. 1998. ``Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms.'' *Neural Computation* 10: 1895--1923.
- Fawcett, T. 2006. ``An Introduction to ROC Analysis.'' *Pattern Recognition Letters* 27: 861--874.
- Montgomery, D. C. 2005. *Design and Analysis of Experiments.* 6th ed., New York: Wiley.
- Yıldız, O. T., and E. Alpaydın. 2006. ``Ordering and Finding the Best of *K*>2 Supervised Learning Algorithms.'' *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28: 392--402.