Session 3

- Clustering
- Evaluation of classifiers
  - Confidence intervals
  - McNemar's test
  - ROC analysis
- Computational learning theory
- Artificial neural networks
- Support vector machines

Clustering

- Find clusters (sets of instances) such that.
  - Instances in same cluster similar.
  - Instances in different cluster different.
- Predictability: $P(A_i = V_{i,j} | C_k)$
- Predictiveness: $P(C_k | A_i = V_{i,j})$
- Category utility:

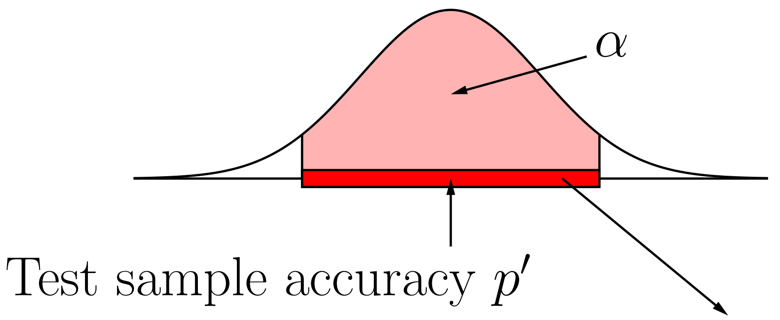
$$\begin{aligned} \mathrm{CU}(C_1 \ldots C_n) = \textstyle\sum_k \sum_i \sum_j \; & P(A_i = V_{i,j}) \cdot \\ & P(C_k \mid A_i = V_{i,j}) \cdot \\ & P(A_i = V_{i,j} \mid C_k) \end{aligned}$$

Evaluation of Classifiers

- Compute accuracy on test sample

  Accuracy $p' =$ proportion of correctly classified examples

- Confidence interval for accuracy



α	z_α
90%	1.64
95%	1.96
99%	2.58

$p' \pm z_\alpha \cdot \sqrt{\frac{p'(1-p')}{n}}$ contains population accuracy p with confidence α

- Confidence interval for difference of accuracy

  $$p'_1 - p'_2 \pm z_\alpha \cdot \sqrt{\frac{p'_1(1-p'_1)}{n_1} + \frac{p'_2(1-p'_2)}{n_2}}$$

McNemar's Test

- Evaluate given classifier on test sample and compute the following table

	# h_1 correct	# h_1 wrong
# h_2 correct	a'	b'
# h_2 wrong	c'	d'

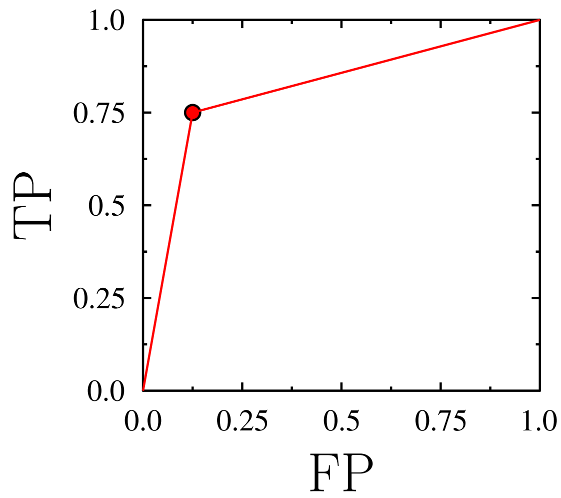
- Assume h_1 and h_2 equally good (assumption H_0):
  $\mathrm{P}[h_1 \text{ correct} \wedge h_2 \text{ wrong}] = \mathrm{P}[h_1 \text{ wrong} \wedge h_2 \text{ correct}] = 0.5$
- Significance = probability of obtaining a result at least as extreme under the assumption H_0
  $= P[b \geq b' | H_0] + P[b \leq c' | H_0]$
  (assuming $b' > c'$)
  $= \ldots$
- Use the binomial distribution to compute this probability.
  $P(b = x) = \binom{n}{x} \cdot p_0^x \cdot (1 - p_0)^{n-x} \qquad \binom{n}{x} = \frac{n!}{x! \cdot (n-x)!}$
  $P(x)$ is the probability of having x successes in n experiments if the probability of success is p_0

ROC Analysis

- Evaluate given classifier on test sample and compute the following table

Predicted	Actual $\oplus$	$\ominus$	
$\oplus$	a	b	$T^p_\oplus$
$\ominus$	c	d	$T^p_\ominus$
	$T^a_\oplus$	$T^a_\ominus$	T

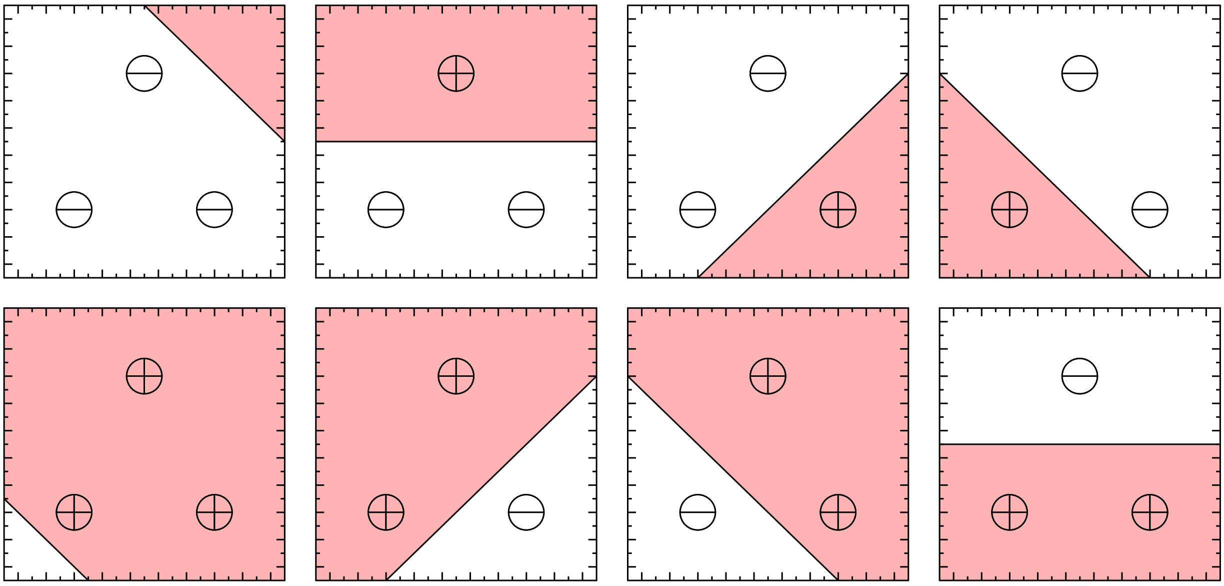
- True positive rate TP is proportion of positive examples that is correctly classified: $\mathrm{TP} = a/T^a_\oplus$
- False positive rate FP is proportion of negative examples that is incorrectly classified as positive: $\mathrm{FP} = b/T^a_\ominus$
- Plot a point for each classifier on the ROC diagram



- Convex hull = rope around points
- Iso-cost: points on this line have equal misclassification cost $\widehat{C}$
  $\widehat{C} = C_{\mathrm{FP}} \cdot \mathrm{FP} \cdot P(-) + C_{\mathrm{FN}} \cdot \mathrm{FN} \cdot P(+)$, where
  $FN = c/T^a_\oplus = 1 - TP$

Computational Learning Theory

- A set of instances S is shattered by $H \Leftrightarrow \forall$ possible concept c defined over S, $\exists h \in H$ consistent with c

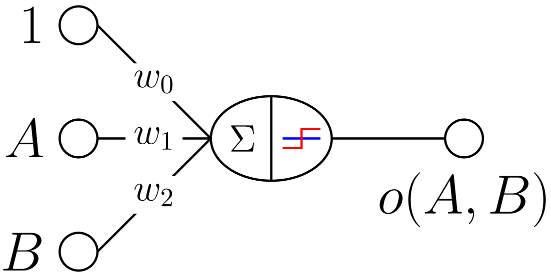


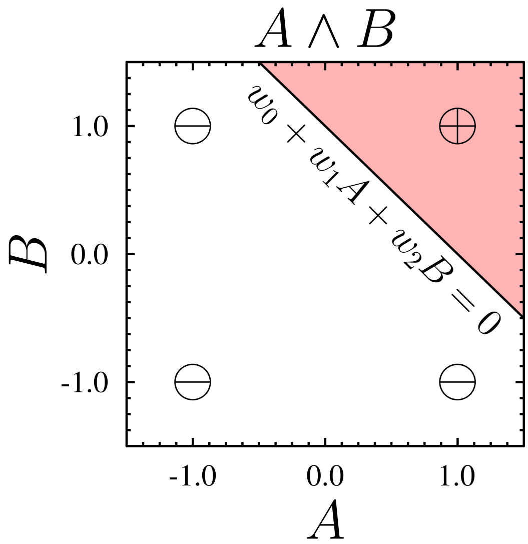
- The Vapnik-Chervonenkis dimension $\mathrm{VC}(H)$ given an instance space X is the size of the largest finite $S \subseteq X$ shattered by H
- $\mathrm{VC}(H) < d \Leftrightarrow$ there is no $S \subseteq X$, with $|S| = d$ that can be shattered by H
- How many randomly drawn training examples suffice to probably (with probability $1-\delta$) approximately (error $\leq \epsilon$) learn any target concept in C?

  $m \geq \frac{1}{\epsilon} \cdot \left(4 \log_2 \frac{2}{\delta} + 8\mathrm{VC}(H) \log_2 \frac{13}{\epsilon}\right)$

Artificial Neural Networks

- Perceptron





$$o(A, B) = \mathrm{sign}(w_0 + w_1 A + w_2 B)$$

- Multi-layer networks
- Threshold unit $\mathrm{sign}(x)$ or $\sigma(x) = \frac{1}{1+e^{-x}}$

Support Vector Machines

- Similar to perceptron
- Weights defined by "maximal margin"
- More expressive by using different kernel functions $K(x, y)$