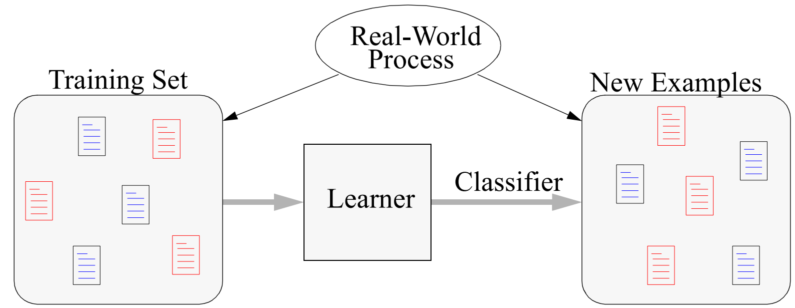
Statistical Learning Theory and PAC-Learning

CS678 Advanced Topics in Machine Learning
Thorsten Joachims
Spring 2003

Outline:

- What is the true (prediction) error of classification rule h?
- How to bound the true error given the training error?
- Finite hypothesis space and zero training error
- Finite hypothesis space and non-zero training error
- Infinite hypothesis spaces: VC-Dimension and Growth Function

Learning Classifiers



Goal:

- Learner uses training set to find classifier with low prediction error.

Learning Classifiers from Examples (Scenario)

Scenario:

- Generator: Generates descriptions $\vec{x}$ according to distribution $P(\vec{x})$.
- Teacher: Assigns a value y to each description $\vec{x}$ based on distribution $P(y|\vec{x})$.

Given:

- Training examples $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(\vec{x}, y)$ $\quad \vec{x}_i \in \Re^N \quad y_i \in \{1, -1\}$
- Set H of classification rules h (hypotheses) that map descriptions $\vec{x}$ to values y $(h: \vec{x} \to y)$.

Goal of Learner:

- Classification rule h from H that classifies new examples (again from $P(\vec{x}, y)$) with low error rate!

$$P(h(\vec{x}) \neq y) = \int \Delta(h(\vec{x}) \neq y) dP(\vec{x}, y) = Err_P(h)$$

Principle: Empirical Risk Minimization (ERM)

Learning Principle:

Find the decision rule $h^\circ \in H$ for which the training error is minimal:

$$h^\circ = \operatorname{arg}min_{h \in H}\{Err_S(h)\}$$

Training Error:

$$Err_S(h) = \frac{1}{n}\sum_{i=1}^{n}\Delta(y_i \neq h(\vec{x}_i))$$

==> Number of misclassifications on training examples.

Central Problem of Statistical Learning Theory:

When does a low training error lead to a low generalization error?

Sources of Variation

Learning Task:

- Generator: Generates descriptions $\vec{x}$ according to distribution $P(\vec{x})$.
- Teacher: Assigns a value y to each description $\vec{x}$ based on $P(y|\vec{x})$.

=> Learning Task: $P(\vec{x}, y) = P(y|\vec{x})P(\vec{x})$

Process:

- Select task $P(\vec{x}, y)$
- Training sample S (depends on $P(\vec{x}, y)$)
- Train learning algorithm A (e.g. randomized search)
- Test sample V (depends on $P(\vec{x}, y)$)
- Apply classification rule h (e.g. randomized prediction)

What is the true error of classification rule h?

Includes variation from different test sets.

Problem Setting:

- given rule h
- given (independent) test sample $S = (\vec{x}_1, y_1), \ldots, (\vec{x}_k, y_k)$ of size k

estimate

$$P(h(\vec{x}) \neq y) = \int \Delta(h(\vec{x}) \neq y) dP(\vec{x}, y) = Err_P(h)$$

Approach: measure error of h on test set

$$Err_V(h) = \frac{1}{k} \sum_{i=1}^{n} \Delta(y_i \neq h(\vec{x}_i))$$

Binomial Distribution

The probability of observing x heads in a sample of n independent coin tosses, when the probability of heads is p in each toss, is

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

Confidence interval:

Given x observed heads, with at leat 95% confidence the true value of p fulfills

$$P(X \geq x|p, n) \geq 0.025 \qquad and \qquad P(X \leq x|p, n) \geq 0.025$$

Cross-Validation Estimation

Given:

- training set S of size n

Method:

- partition S into m subsets of equal size
- for i from 1 to m
  - train learner on all subsets except the i' th
  - test learner on i' th subset
  - record error rates on test set

=> Result: average over recorded error rates

Bias of estimate: see leave-one-out

Warning: Test sets are independent, but not the training sets!
=> no strictly valid hypothesis test is known for general learning algorithms (see [Dietterich/97])

Psychic Game

- I guess a 4 bit code
- You all guess a 4 bit code

=> The student who guesses my code clearly has telepathic abilities - right!?

How can You Convince Me of Your Psychic Abilities?

Setting:

- n bits
- $|H|$ players

Question: For which n and $|H|$ is prediction of zero-error player significantly different from random ($p = 0.5$) with probability $1 - \delta$?

=> Hypothesis test for

$$P(h_1 correct \vee \ldots \vee h_{|H|} correct, allnonpsychic) < \delta$$

or

$$P(\exists h \in H; Err_s(h) = 0, \forall h \in H; Err_P(h) = 0.5) < \delta$$

PAC Learning

Definition:

- C = class of concepts $c;X \rightarrow \{1, -1\}$ (functions to be learned)
- H = class of hypotheses $h;X \rightarrow \{1, -1\}$ (functions used by learner A)
- S = training set (of size n)
- ε = desired error rate of learned hypothesis
- δ = probability, with which the learner A is allowed to fail

C is PAC-learnable by Algorithm A using H and n examples, if

$$P(Err(h_{A(S)}) \leq \varepsilon) \geq (1 - \delta)$$

for all $c \in C$, ε, δ, and P(X) so that A runs in polynomial time dependent on ε, δ, the size of the training examples and the size of the concepts.

=> only polynomially many training examples allowed.

Case: Finite H, Zero Error

- The hypothesis space H is finite
- There is always some h with zero training error (A returns one such h)
- Probability that a (single) h with $Err_P(h) \geq \varepsilon$ has training error of zero

$$(1 - \varepsilon)^n$$

- Probability that there exists h in H with $Err_P(h) \geq \varepsilon$ that has training error of zero

$$P(\exists h \in H; Err_s(h) = 0, Err_P(h) > \varepsilon) \leq |H|(1 - \varepsilon)^n \leq |H| e^{-\varepsilon n}$$

Case: Finite H, Non-Zero Error

Goal:

$$P(|Err_S(h_{A(S)}) - Err_D(h_{A(S)})| \le \varepsilon) \ge (1 - \delta)$$

<=

$$P(sup_H |Err_S(h_i) - Err_S(h_i)| \le \varepsilon) \ge (1 - \delta)$$

- Probability that for a fixed h, training error and test error differ by more than ε (Hoeffding / Chernoff Bound)

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} x_i - p\right| > \varepsilon\right) \le 2e^{-2n\varepsilon^2}$$

- Probability over all h in H: union bound => multiply by |H|

Case: Infinite H

- union bound does no longer work.
- maybe not all hypotheses are really different?!

How Many Dichotomies for Fixed Sample?

- Sample S of size *n*
- Hypothesis class H

$$\Pi_H(S) = \{(h(\vec{x}_1), h(\vec{x}_2), \ldots, h(\vec{x}_n)); h \in H\}$$

Definition: H shatters S, if $|\Pi_H(S)| = 2^n$ (i.e. hypotheses from H can split S in all possible ways).

Vapnik/Chervonenkis Dimension

Definition: The VC-dimension of *H* is equal to the maximal number *d* of examples that can be split into two sets in all 2^d ways using functions from *H* (shattering).

	x_1	x_2	x_3	...	x_d
h_1	+	+	+	...	+
h_2	-	+	+	...	+
h_3	+	-	+	...	+
h_4	-	-	+	...	+
...	...	...	...	...	...
h_N	-	-	-	...	-

Growth function $\Phi_d(S)$**:** For all S

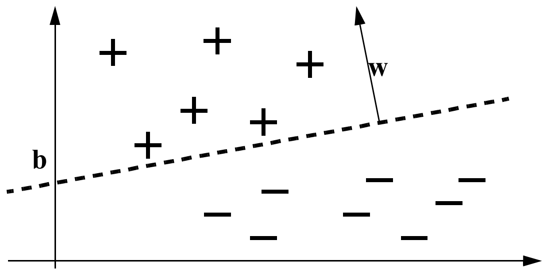
$$|\Pi_H(S)| \le \Phi_{VCdim(H)}(n) \le \left(\frac{en}{VCdim(H)}\right)^{VCdim(H)}$$

Linear Classifiers

Rules of the Form: weight vector $\vec{w}$, threshold b

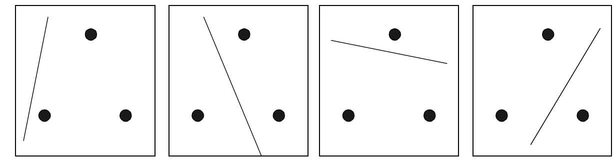
$$h(\vec{x}) = sign\left[\sum_{i=1}^{N} w_i x_i + b\right] = \begin{cases} 1 & if \sum_{i=1}^{N} w_i x_i + b > 0 \\ -1 & else \end{cases}$$

Geometric Interpretation (Hyperplane):

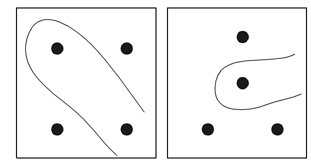


VC-Dimension of Hyperplanes in $\Re^2$

- Three points in $\Re^2$ can be shattered with hyperplanes.



- Four points cannot be shattered.



=> Hyperplanes in $\Re^2$ -> *VCdim=3*

General: Hyperplanes in $\Re^N$ -> *VCdim=N+1*

Error Bound

Question: After n training examples, how close is the training error to the true error?

With probablility η it holds for all $h \in H$:

$$Err_P(h) - Err_S(h) \le \Phi(d, n, \eta)$$

$$\Phi(d, n) = \sqrt{\frac{d\left(\ln\frac{2n}{d} + 1\right) - \ln\frac{\eta}{4}}{n}}$$

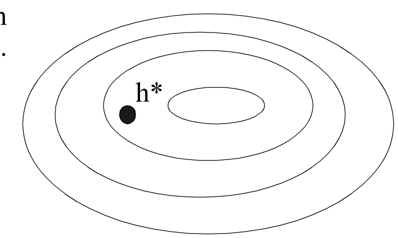
- n number of training examples
- d VC-dimension of hypothesis space H

==> $\boxed{Err_P(h) \le Err_S(h) + \Phi(d,n,\eta)}$

SVM Motivation: Structural Risk Minimization

$$\boxed{Err_P(h_i) \le Err_S(h_i) + \Phi(VCdim(H),n,\eta)}$$

Idea: Structure on hypothesis space.



Goal: Minimize upper bound on true error rate.