Boosting: theoretical foundations and algorithms

Marc Sebban

LABORATOIRE HUBERT CURIEN, UMR CNRS 5516
University of Jean Monnet Saint-Étienne (France)

Doctoral school CIL

december 2009

Outline

1. Introduction to Learning Theory
2. Ensemble Methods
   - Heterogeneous ensemble methods
   - Homogeneous ensemble methods
3. Boosting
   - ADABOOST
   - Theoretical Results
   - SVM versus boosting
   - Boosting and Game Theory
   - Advantages -Caveats of ADABOOST

Supervised learning problem

Notations

- *Let S be a set of m training examples $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_m}, y_m)\}$.*
- *The $\mathbf{x_i}$ values are typically vectors of the form $< x_{i1}, \ldots, x_{ip} >$, whose components are usually called features. The $\mathbf{x_i}$ are drawn from a feature space $\mathcal{X}$ according to a given distribution $\mathcal{D}_{\mathcal{X}}$.*
- *The y values ($y \in \mathcal{Y}$) are drawn from a discrete set of classes (classification) or are continuous values (regression).*
- *We assume that there exists a* **target function** *f such that $y = f(\mathbf{x}), \forall(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$.*

Definition

A supervised learning algorithm L automatically outputs from S a classifier (or a hypothesis) $h \in \mathcal{H}$ about the target function f.

Empirical and generalization errors

Definition

The empirical error (or risk) $\hat{\epsilon}_h$ of a hypothesis $h \in \mathcal{H}$ is the proportion of errors that h makes over the learning sample S.

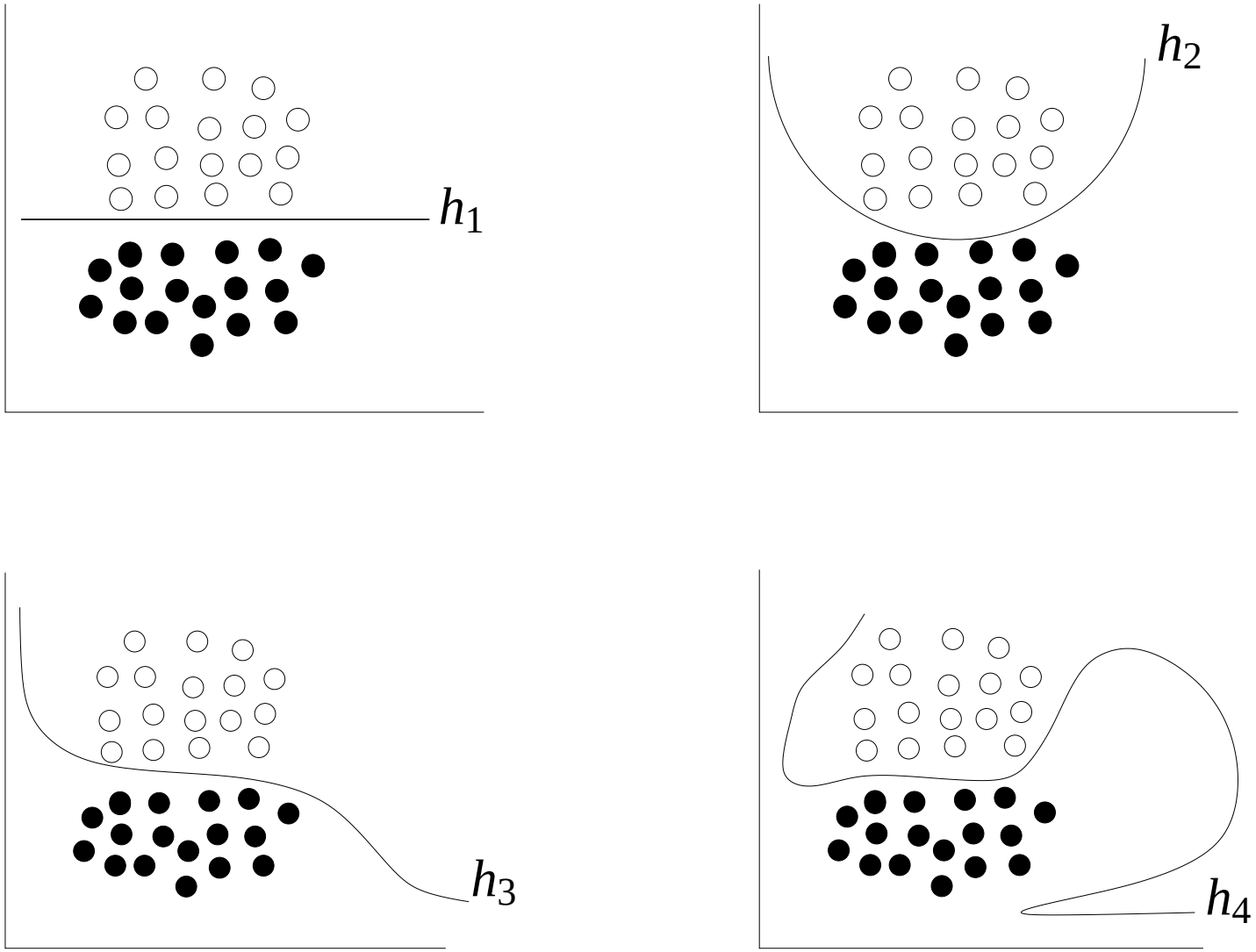
$$\hat{\epsilon}_h = \frac{1}{|S|} \sum_i \mathbb{1}_{h(\mathbf{x_i}) \neq y_i}$$

Definition

The (unknown) generalization error (or real risk) ϵ_h of a hypothesis h is the error probability of h over $\mathcal{X}$ according to $\mathcal{D}_{\mathcal{X}}$.

$$\epsilon_h = P_{\mathbf{x_i} \in \mathcal{X}, \mathcal{D}_{\mathcal{X}}}[h(\mathbf{x_i}) \neq y_i]$$

What is a good theory? (Example 1)



What is a good theory? (Example 2)

Question

Let us observe the following sequence of integers:

$$1\ 2\ 3\ 5\ \ldots$$

What comes next?

What is a good theory? (Example 2)

1 2 3 5 ...?

Several solutions s are possible?

- $s = 7 \rightarrow$ Why? ``This is the next prime number''
- $s = 8 \rightarrow$ Why? ``This is the next Fibonacci number'' $s.t.\ u_{n+1} = u_n + u_{n-1}$
- $s = 7 \rightarrow$ Why? ``This is the next element of the binary series 1(1), 10(2), 11(3), 101(5), 111(7), 1011(11), 1111(15), etc.''
- $s = 6 \rightarrow$ Why? ``It is the sequence of all integers, except 4...''

What is a good theory? (Example 3)

Question

What are the missing values?

A 1	B 2	C 3
C 2	A 3	B 1
B 3	C 1	? ?

What is a good theory? (Example 3)

A natural solution is the following:

A 1	B 2	C 3
C 2	A 3	B 1
B 3	C 1	**A** **2**

but...

... if other learning data are available...

A 1	B 2	C 3	D 4	E 5	F 6
C 2	A 3	B 1	F 5	D 6	E 4
B 3	C 1	**D** **4**	E 6	F 2	A 5
D 4	F 5	E 6	A 2	B 3	C 1
E 5	D 6	F 2	B 1	A 4	C 3
F 6	E 4	A 5	C 3	B 1	D 2

Moral of these examples

It is always possible to find a (possibly complex) consistent theory that explains a phenomenon.

Question

(once again...) what is a good theory?

No free lunch:

- if there is no restriction on the target concept, *generalization is impossible*.
- if there is no assumption on how the past is related to the future, *generalization is impossible*.

Conclusion 1

We need to have **knowledge** about the target concept $\Rightarrow$ Choose a class of hypotheses $\mathcal{H}$ (*inductive bias*).

Conclusion 2

We need to make **assumptions** about the phenomenon:

- *Stationarity*: future observations are related to past ones.
- *Inductive principle*: choose the simplest explanation consistent with past data (*Occam's razor*).

Definition

Updated definition

Learning can be viewed as a problem of function estimation, whose goal is to determine the simplest consistent (w.r.t. S) hypothesis $h \in \mathcal{H}$ of the target concept $f \in \mathcal{F}$.

Bias/Variance trade-off

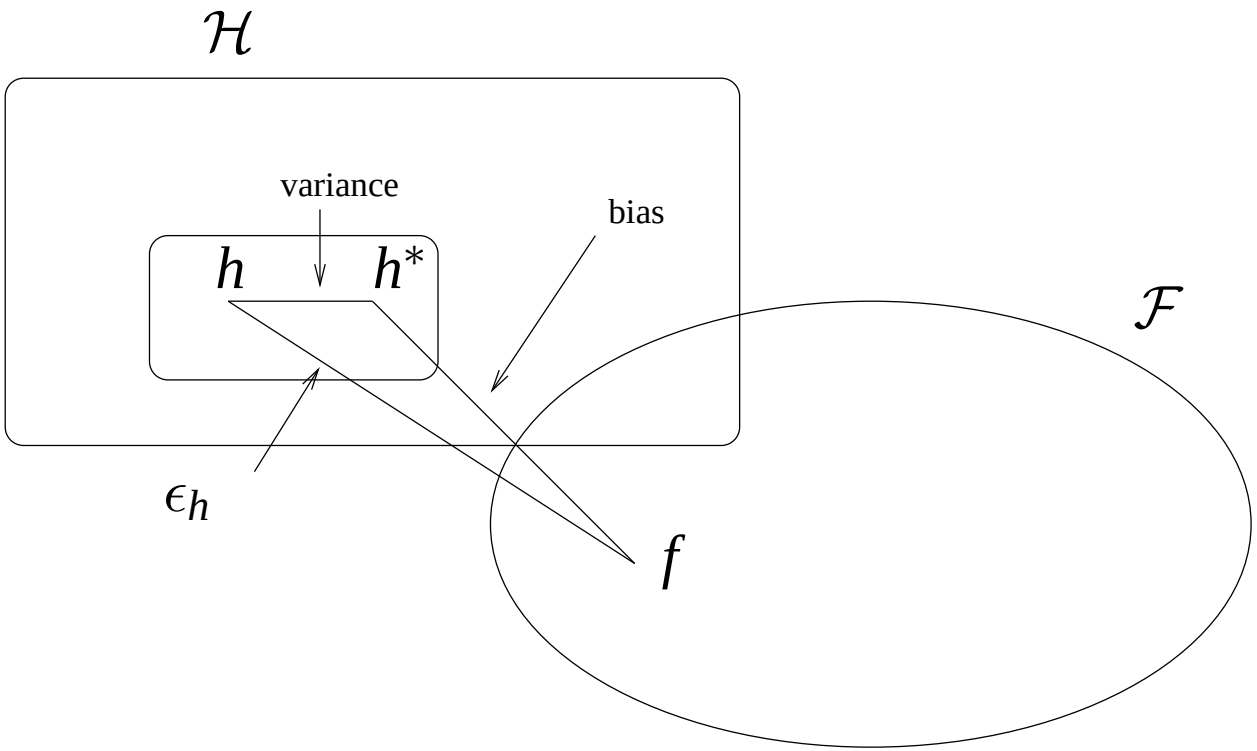
The real risk of the induced hypothesis h w.r.t f comes from two main sources:

Definition

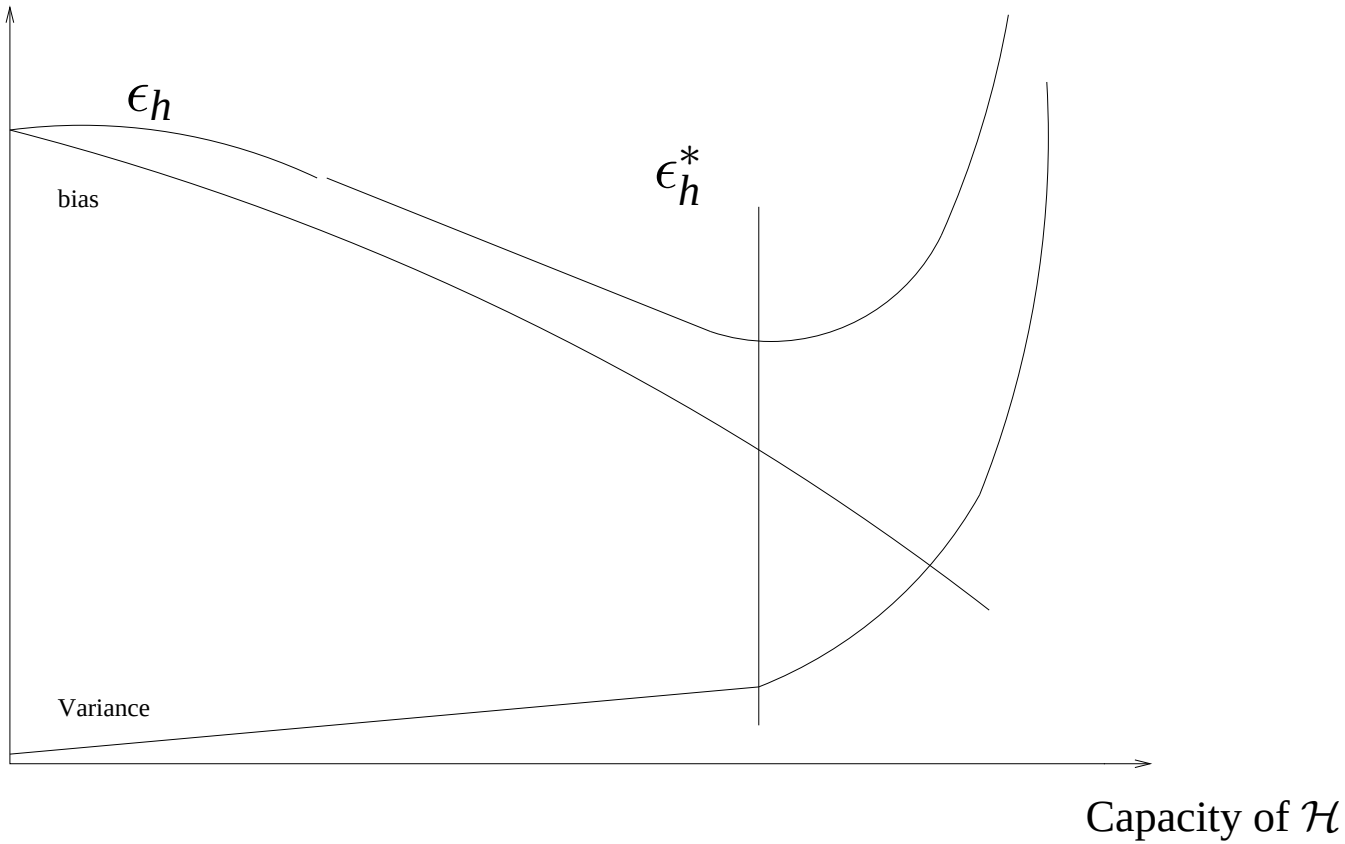
1. **The inductive bias**: nothing guarantees the equality between the target concept space $\mathcal{F}$ and the selected class of hypotheses $\mathcal{H}$. In other words, even if the learner is able to provide an optimal hypothesis h^* from $\mathcal{H}$, h^* and f can be different. $\Rightarrow$ **approximation error**.
2. **The variance**: since the learning process depends on the quality and the quantity of learning data, the learner usually does not provide the optimal hypothesis h^*. The distance between h^* and the induced hypothesis h is the **estimation error**.

Bias/Variance trade-off

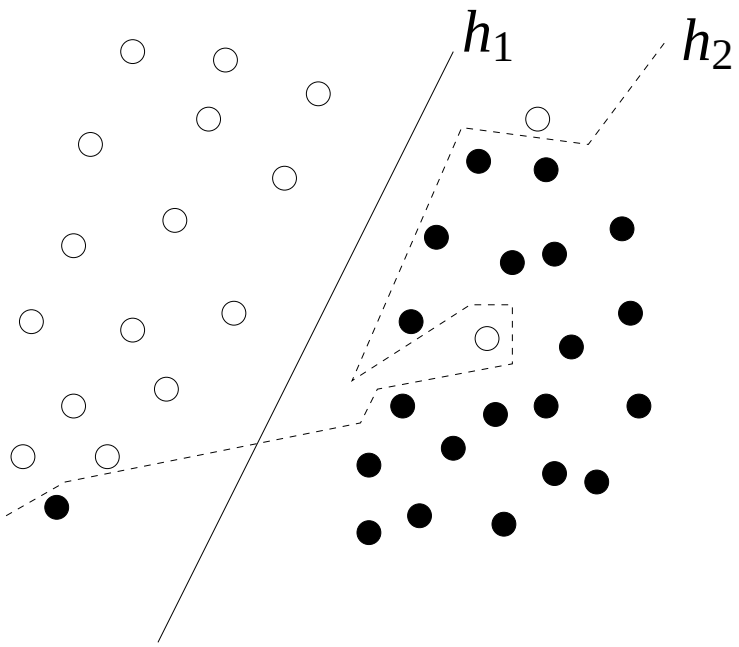
Graphically, we get:



Bias/Variance trade-off



Overfitting



Question

Is h_2 reasonable?

Statistical learning theory investigates under what conditions empirical risk minimization (ERM) is admissible.

Empirical Risk Minimization

Definition

The ERM principle is valid if the real risk of the hypothesis $h \in \mathcal{H}$ induced from S is close to the real risk of the optimal hypothesis $h^* \in \mathcal{H}$.

$$h = \arg\min_{h_i \in \mathcal{H}} \hat{\epsilon}_{h_i}$$

$$h^* = \arg\min_{h_i \in \mathcal{H}} \epsilon_{h_i}$$

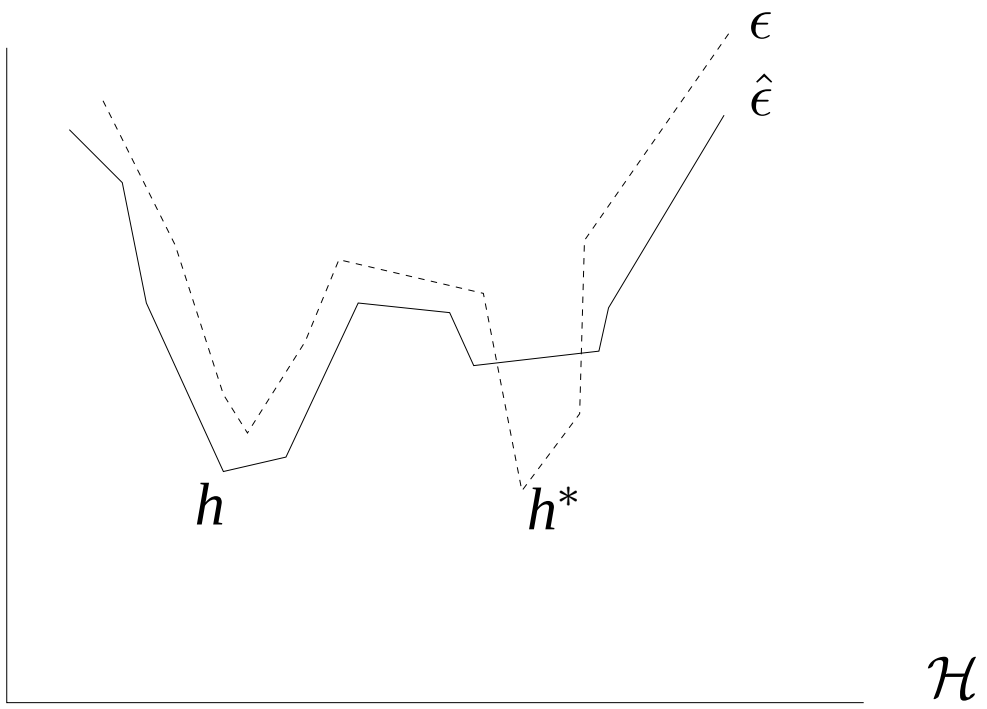
Condition of validity of the ERM principle:

$$\forall \mathcal{D}_{\mathcal{X}}, \forall \gamma \geq 0, \forall \delta \leq 1, P(|\epsilon_h - \epsilon_{h^*}| \geq \gamma) \leq \delta$$

Empirical Risk Minimization

Remark

Selecting $h = \arg\min_{h_i \in \mathcal{H}} \hat{\epsilon}_{h_i}$ *does not guarantee the discovery of the optimal hypothesis* $h^* = \arg\min_{h_i \in \mathcal{H}} \epsilon_{h_i}$.



PAC model

The PAC model (Probably Approximately Correct) and the works of Valiant (1984) provide a theoretical answer to this problem.

Definition

The ERM principle is valid if

$$\epsilon_h \rightsquigarrow_{m \to \infty} \epsilon_{h^*}$$

$$\hat{\epsilon}_h \rightsquigarrow_{m \to \infty} \epsilon_{h^*}$$

Question

Under what conditions do the empirical and real risks of the induced hypothesis h converge towards the real risk of h^* ?

When $|\mathcal{H}|$ is finite

Hoeffding's inequality (that gives an upper bound on the probability for a random variable to deviate from its expected value) can be applied on the convergence of the empirical risk of a **given** hypothesis h to its real risk.

Theorem

For a given $h \in \mathcal{H}, \forall \gamma \geq 0, \forall m > 0, \forall \mathcal{D}_{\mathcal{X}}$

$$P(|\epsilon_h - \hat{\epsilon}_h| \geq \gamma) < 2e^{-2\gamma^2 m}$$

From this theorem, we can bound the difference between the empirical risk and the real risk of **any** hypothesis of $\mathcal{H}$. Since we have a union of independent events, we get:

Theorem

$\forall h \in \mathcal{H}, \forall \mathcal{D}_{\mathcal{X}}, \forall \gamma \geq 0, \forall m > 0,$

$$P(|\epsilon_h - \hat{\epsilon}_h| \geq \gamma) < 2|\mathcal{H}|e^{-2\gamma^2 m}$$

When $|\mathcal{H}|$ is finite

Therefore, we know that

$$P(|\epsilon_h - \hat{\epsilon}_h| \geq \gamma) < 2|\mathcal{H}|e^{-2\gamma^2 m}$$

and we want to satisfy that

$$P(|\epsilon_h - \epsilon_{h^*}| \geq \gamma) \leq \delta$$

By equating the two upper bounds, we get:

$$\gamma \geq \left(\frac{1}{2m} ln \frac{2|\mathcal{H}|}{\delta}\right)^{1/2}$$

When $|\mathcal{H}|$ is finite

So, with a probability $\geq 1-\delta, \forall h \in \mathcal{H}$:

$$\epsilon_h - \gamma < \hat{\epsilon}_h < \epsilon_h + \gamma$$

Therefore,

$$\begin{aligned}
\epsilon_h &\leq \hat{\epsilon}_h + \gamma \\
&\leq \hat{\epsilon}_{h^*} + \gamma \text{ (because } h^* = \arg\min_{h \in \mathcal{H}} \hat{\epsilon}_h) \\
&< (\epsilon_{h^*} + \gamma) + \gamma \\
&= \epsilon_{h^*} + 2\gamma
\end{aligned}$$

Plugging the expression $\gamma \geq \left(\frac{1}{2m} ln \frac{2|\mathcal{H}|}{\delta}\right)^{1/2}$ we get with probability $\geq 1 - \delta$:

$$\epsilon_h \leq \epsilon_{h^*} + \left(\frac{2}{m} ln \frac{2|\mathcal{H}|}{\delta}\right)^{1/2}$$

So, by fixing $\gamma = \left(\frac{2}{m} ln \frac{2|\mathcal{H}|}{\delta}\right)^{1/2}$ we deduce that $m = \frac{2}{\epsilon^2} ln \frac{2|\mathcal{H}|}{\delta}$.

Theorem

The ERM principle requiring that

$$\forall \gamma \geq 0, \delta \leq 1, P(|\epsilon_h - \epsilon_{h^*}| \geq \gamma) \leq \delta$$

is valid if $m \geq \frac{2}{\epsilon^2} ln \frac{2|\mathcal{H}|}{\delta}$.

Conclusion

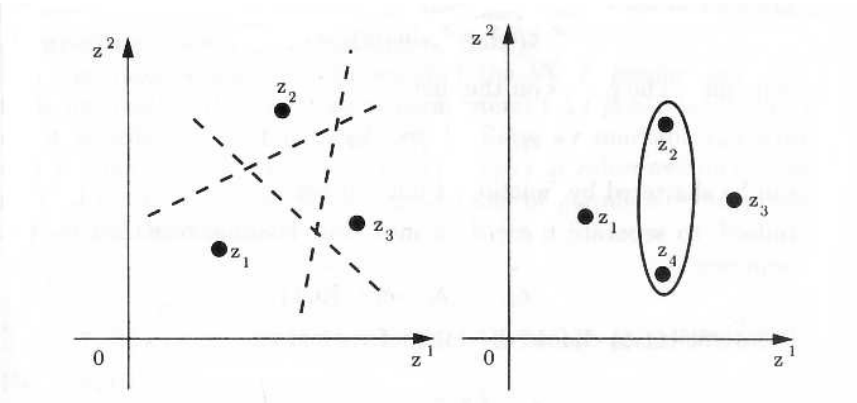
There is nothing so practical as a good theory !! [Kurt Lewin]

When $|\mathcal{H}|$ is infinite

We can exploit the VC dimension $d_{\mathcal{H}}$ (for Vapnik-Chervonenkis dimension) that is a measure of the **capacity** of the class of hypotheses $\mathcal{H}$.

Definition

- The VC dimension $d_{\mathcal{H}}$ of a class of hypotheses $\mathcal{H}$ is defined as the cardinality of the largest set of points that a hypothesis $h \in \mathcal{H}$ can **shatter**.
- A set of points is **shattered** if for all assignments of labels to those points, there exists a hypothesis $h \in \mathcal{H}$ that makes no error of classification when evaluating that set of data points.



When $|\mathcal{H}|$ is infinite

From the VC dimension $d_{\mathcal{H}}$, we can define an upper bound on the real error:

Theorem

Let $\mathcal{H}$ be a class of hypotheses, $\forall h \in \mathcal{H}, \forall \delta \geq 0, \forall m > 0$, the following bound holds:

$$\epsilon_h \leq \hat{\epsilon}_h + \sqrt{\frac{d_{\mathcal{H}}(ln\frac{2m}{d_{\mathcal{H}}} + 1) + 1 - ln\frac{\delta}{4}}{m}}$$

Remark

Theoretical upper bounds are often pessimistic... For instance, with the class $\mathcal{H}$ of 2D-linear separators, and fixing $\gamma = 0.01$ and $\delta = 0.05$, we get $m \geq 1600$...

Conclusion

- Due to the bias/variance trade-off, the choice of a **single** "good" hypothesis is constrained by a compromise to found between a variance control and a bias control.
- How can we try to reduce one of the two quantities while controlling (or reducing) the other?
- **Ensemble methods** are very effective techniques to achieve this task.

Ensemble Methods

Definition

Ensemble methods are learning algorithms that construct a set of classifiers $h_1, \ldots, h_T$ whose individual decisions are combined in some way to classify new examples.

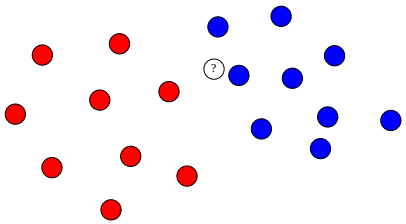
Necessary and sufficient conditions for an ensemble of classifiers to be efficient:

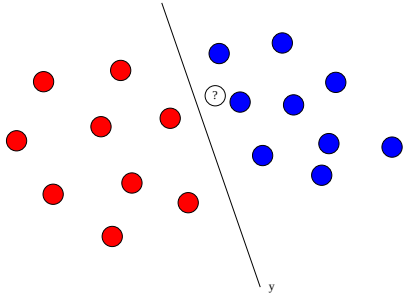
- the individual classifiers (or hypotheses) are accurate, *i.e.* they have an error rate of better than random guessing.
- the classifiers are diverse, *i.e.* they make different errors on new data points.

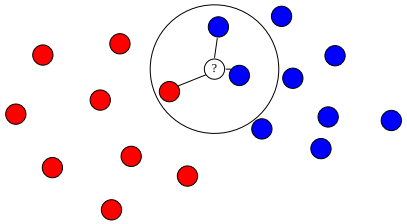
Question

Is it possible to construct good ensembles?

Why and how do we get various classifiers?



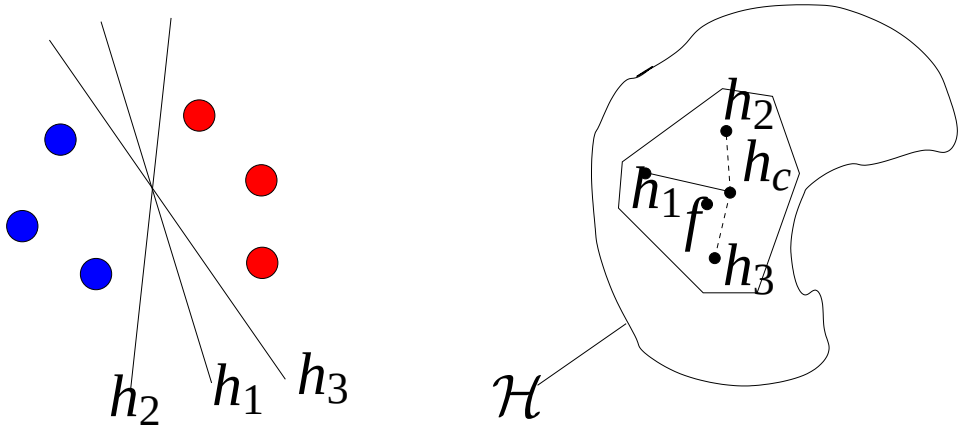




- Different learning techniques (*e.g.* k-NNs, linear separator, decision trees, SVMs, etc.) on the same learning sample S.
- Different learning parameters (*e.g.* number of neighbors k) on the same learning sample S.
- Different learning sample S with the same learning technique.
- Different representations of the same learning set.

Limitations of a single classifier [Dietterich 2002] (1/3)

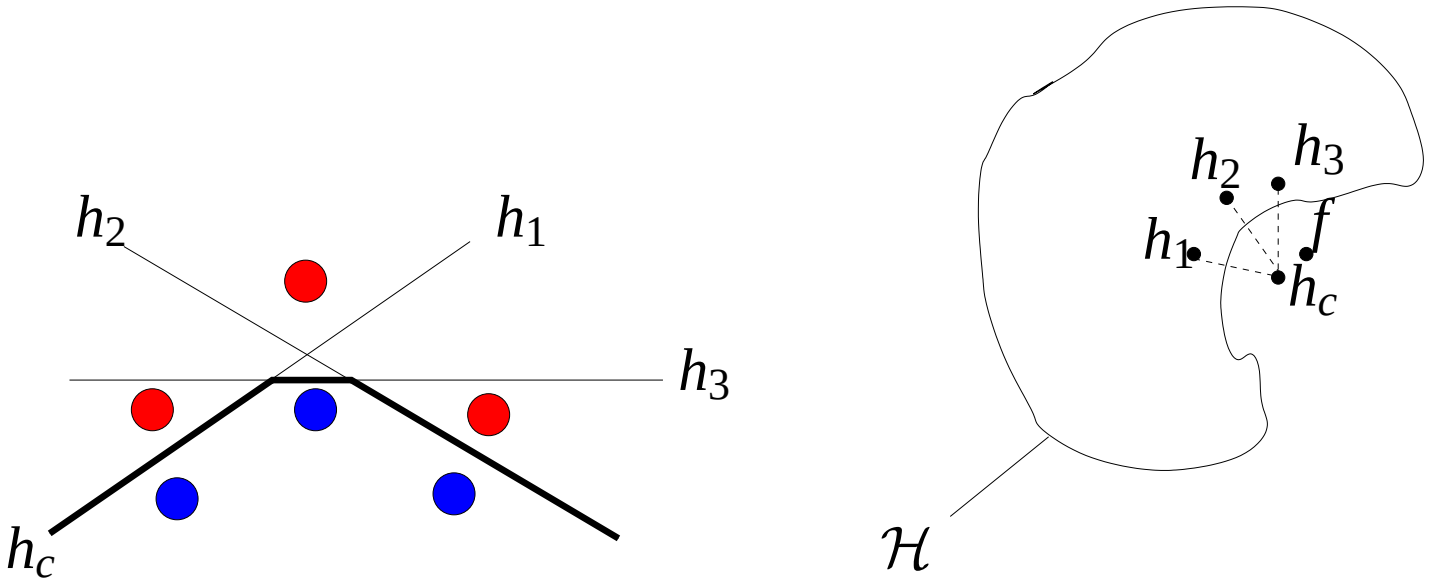
- **Statistical problem (variance)**: Without sufficient data, the learning algorithm can find many different hypotheses in $\mathcal{H}$ that all give the same empirical accuracy on S.



By constructing an ensemble out of all of these accurate classifiers, the algorithm can "average" their votes and reduce the risk of choosing the wrong classifier.

Limitations of a single classifier (2/3)

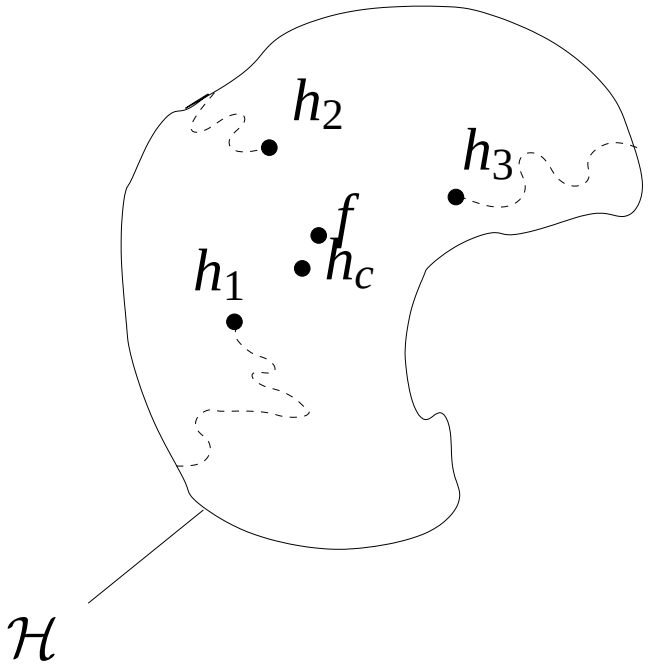
- **Representational problem (bias)**: In most applications of machine learning, the true function f cannot be represented by any of the hypotheses in $\mathcal{H}$.



By forming weighted sums of hypotheses drawn from $\mathcal{H}$, it may be possible to expand the space of representable functions.

Limitations of a single classifier (3/3)

- **Computational problem**: Many learning algorithms work by performing some form of local search that may get stuck in local optima. An ensemble constructed by running the local search from many different starting points may provide a better approximation to the unknown function.



Ensemble Methods

A clustering of the ensemble methods can be performed according to the origin of the diversity of the hypotheses [Džeroski & Ženko 2004]

- **Heterogeneous ensemble methods**: several classifiers $h_1, \ldots h_T$ are generated by applying different learning algorithms $L_1, \ldots, L_T$ to a single training dataset, *i.e.* to a constant distribution D of the training data.

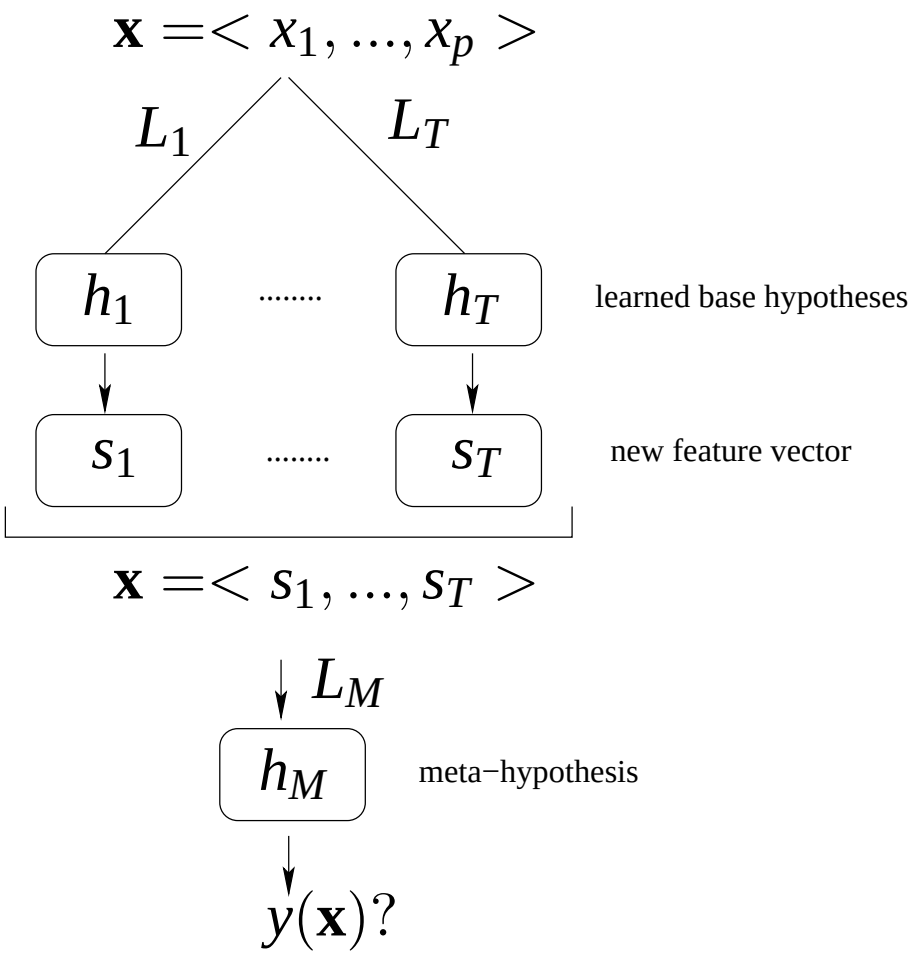
- **Homogeneous ensemble methods**: several hypotheses $h_1, \ldots h_T$ are generated from a single learning algorithm L. The diversity of the hypotheses is obtained by modifying the statistical distribution D_t of the training examples used to build h_t.

Heterogeneous ensemble methods

The diversity comes from the learning algorithms

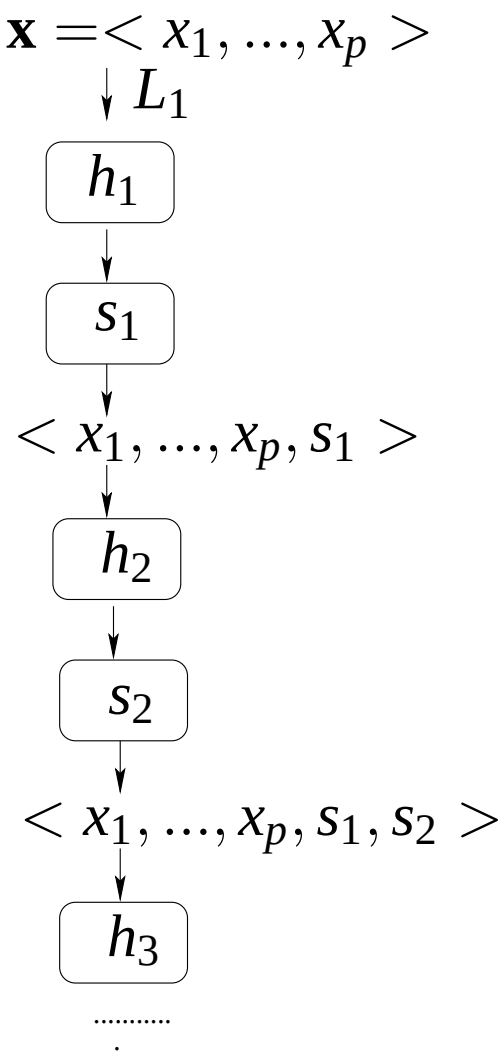
- **Stacking** [Wolpert, D.H. 1992]
- **Cascade Generalization** [Gamma, J. & Brazdil, P. 2000]

Stacking



1. Learn T hypotheses $h_1, \ldots, h_T$ with T different learning algorithms $L_1, \ldots, L_T$.
2. The decisions (scores) of $h_1, \ldots, h_T$ on $\mathbf{x}$ are seen as new features
3. Learn a meta hypothesis in this new T dimensional space.

Cascade Generalization



1. Learn a hypothesis h_1 with a learning algorithm L_1. Classify the learning examples with h_1.
2. Learn a hypothesis h_2 with a learning algorithm L_2 from the original features and the label (or the score) predicted at the previous step. Classify the learning examples with h_2.
3. Repeat the process.

Homogeneous ensemble methods

- The diversity comes from the distribution of the learning examples.
- We consider now the problem of combining several classifiers built from different subsets of the training data.
- All classifiers are generally of the same type.
- Why resampling the training data? It allows us to build more robust estimates by reducing the variance of the estimator.
- Homogeneous ensemble methods:
  - **Bagging** [Breiman 1996]
  - **Random Forests** [Breiman 2001]
  - **Boosting** [Freund & Schapire 1999]

Bagging

BAGGING

Input: *A learning sample* $S = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_m}, y_m)\}$
Input: *A total number* T *of bagging rounds*
Input: *A learning algorithm* L *returning a binary classifier*
Output: *A combined classifier*
for all t *from* 1 *to* T **do**
    $S_t = Resample(S)$ *// Randomly sample* S *with replacement;*
    $h_t(\mathbf{x}) = L(S_t)$ *// Build a classifier on* S_t *using learning algorithm* L*;*

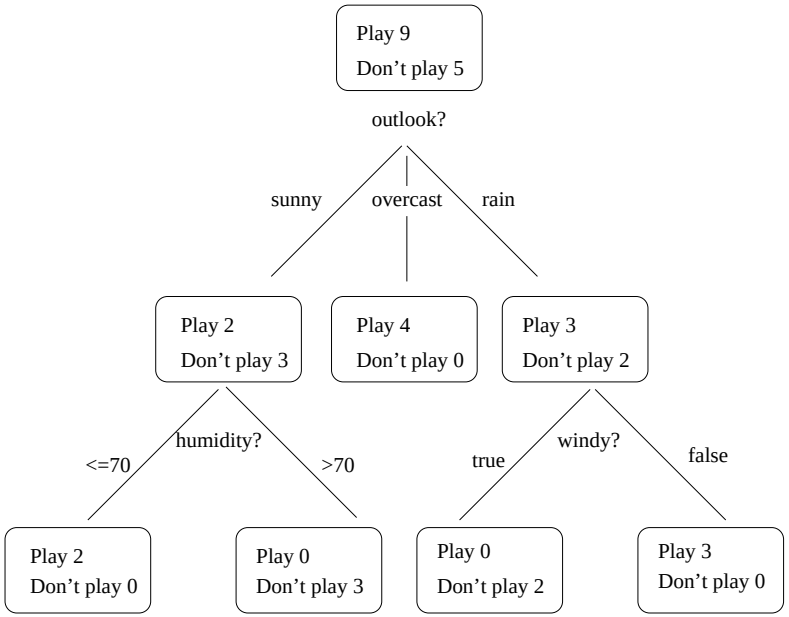
Return H_T **such that**
    $H_T(\mathbf{x}) = sign\left(\sum_t h_t(\mathbf{x})\right)$

Random Forests

Definition

A decision tree is a tree that can be learned by splitting S into subsets based on a feature value test. This process is repeated on each derived subset in a recursive manner.

Example: the decision to play or not play based on climate conditions.



Random Forests

Definition

Decision Trees + Random Feature Selection + Bagging = Random Forests

- **Aim**: generate diversity in decision trees.
- The general approach is like bagging:
  - build model on successive resampling (with replacement) of S;
  - make a majority vote to form the combined classifier.
- Decision trees are built with no pruning.
- While growing the tree, the feature selected from splitting maximizes the impurity (information gain, gini index) among a random selection of F features out of p possible attributes (typical values are $F = \lceil log_2 p \rceil$ or $F = \lceil \sqrt{p} \rceil$.

Introduction to boosting

Boosting [Freund & Schapire 1999]

Let us start from an example [Freund & Schapire 1999]....

- **Aim**: A horse-racing gambler, hoping to maximize his winnings, decides to create a computer program that will accurately predict the winner of a horse race.
- **Strategy 1**: ask a highly successful expert gambler to explain his betting strategy. Not surprisingly, the expert is unable to articulate a large set of rules for selecting a horse.
- **Strategy 2**: But, when presented with the data for a specific set of races, he is able to express some rules such as:
  - h_1 : `''Bet on the horse that has recently won the most races''`.
  - h_2 : `''Bet on the horse with the most favored odds''`.

Boosting

In order to use these rules to maximum advantage, there are two problems faced by the gambler:

1. How to choose the collections of races presented to the expert so as to extract rules that will be the most useful?
2. Once we have collected many rules, how to combine them into a single, highly accurate prediction rule?

Solutions:

1. If the combination is not weighted and the learning examples are randomly selected $\rightarrow$ **Bagging**.
2. If the combination is weighted and the selection of the learning examples is driven by a "hard" examples $\rightarrow$ **Boosting**.

Strong vs Weak Learnability

Definition

Boosting is a general method for improving (under some constraints) the accuracy of any given learning algorithm.

Boosting combines *weak* hypotheses (*i.e.* just better than a random guessing) into a *strong* hypothesis.

Strong vs Weak Learnability

Boosting has its roots in the theoretical PAC learning model.

Definition

A learning algorithm L is *strong PAC* iff:

- with a polynomial number of learning examples S,
- $\forall$ distribution D over S,
- with a high probability $1 - \delta$ ($\delta > 0$),
- L is able to induce a hypothesis h with a generalization error $\leq \epsilon$ ($\epsilon > 0$).

Strong vs Weak Learnability

Definition

A learning algorithm L is *weak PAC* [Kearns, M. & Valiant, L.G. 1988] iff:

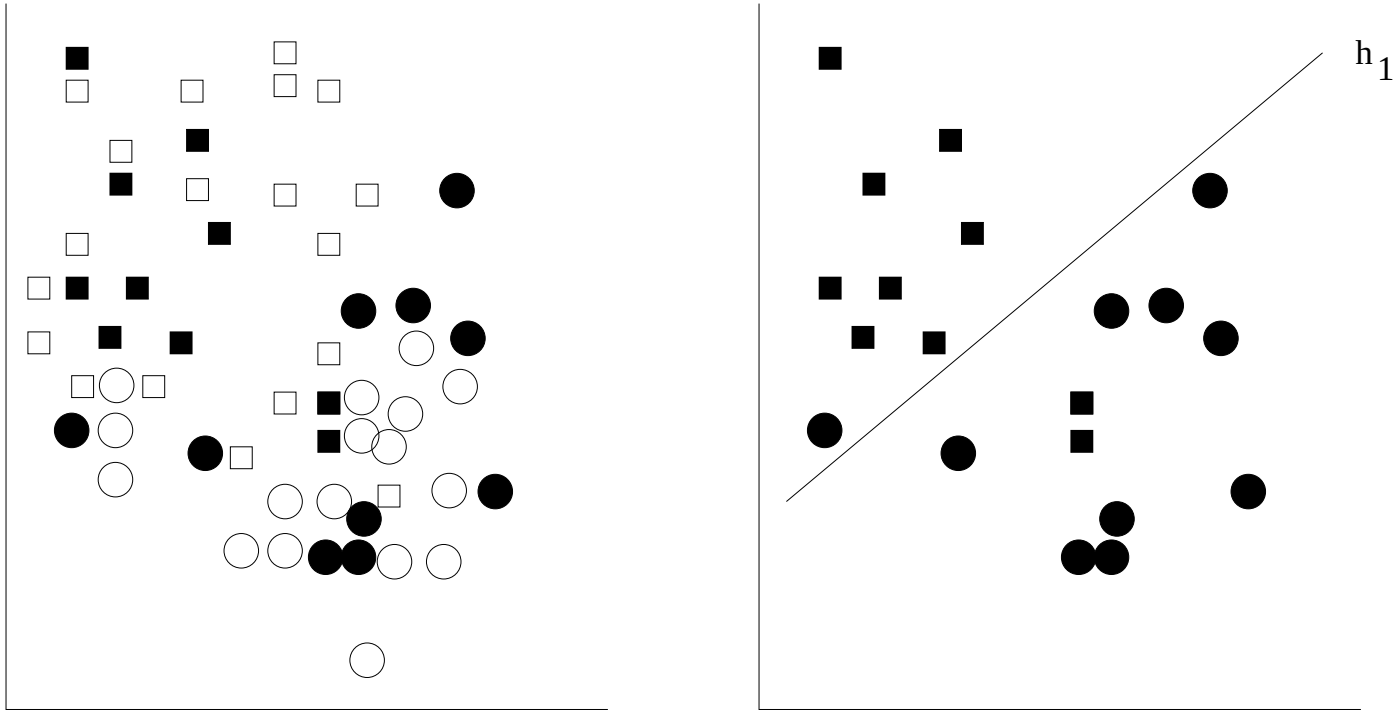
- Same definition as *strong PAC* except the fact that the generalization error just has to be slightly better than a random guessing.
- *i.e.* $\epsilon \geq \frac{1}{2} - \gamma$.

Question

Can a weak learning algorithm which performs just slightly better than random guessing in the PAC model be "boosted" into an arbitrarily accurate "strong" learning algorithm?

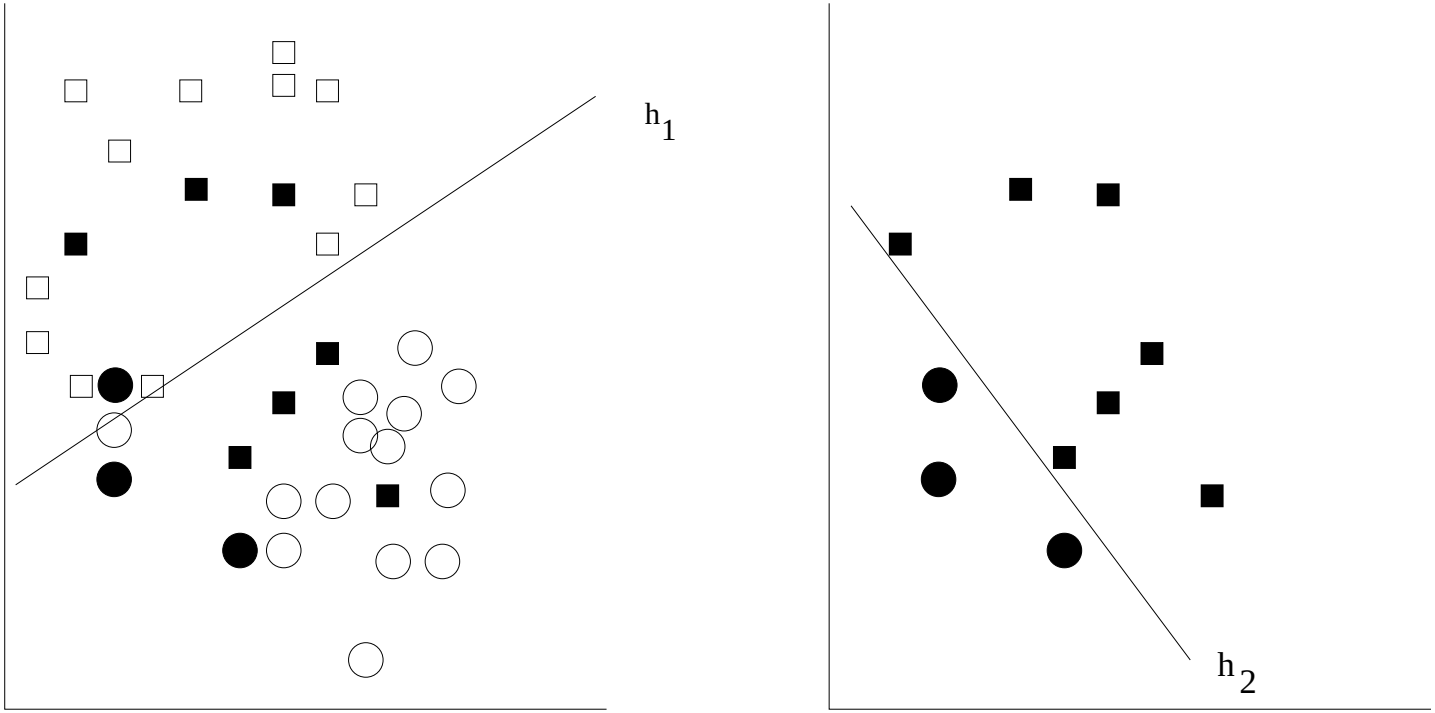
First boosting algorithm (1/4)

Step 1: Extract from S a learning sample S_1. Use a learning algorithm L to produce a first hypothesis h_1.



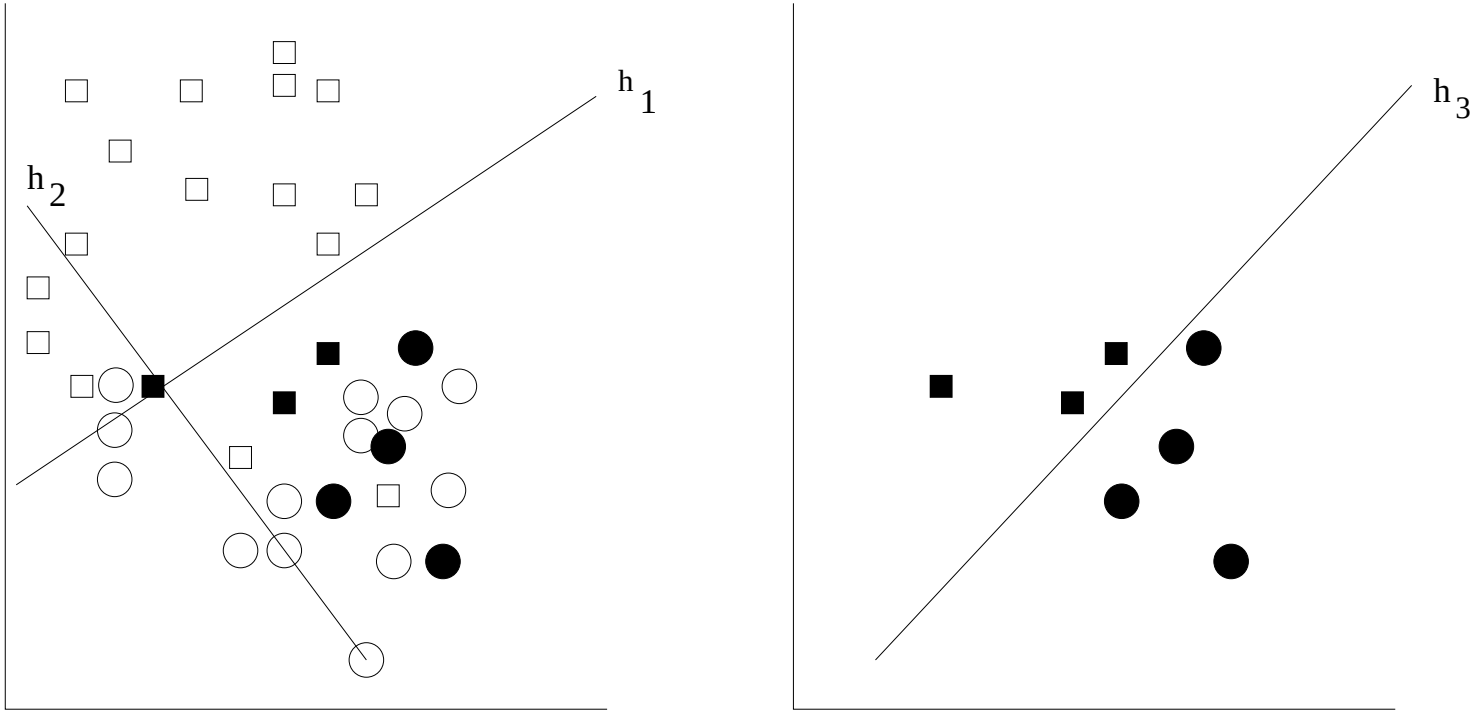
First boosting algorithm (2/4)

Step 2: Generate a second learning sample S_2, in which an instance has a roughly equal chance of being correctly or incorrectly classified by h_1. L is used again to infer a new hypothesis h_2.



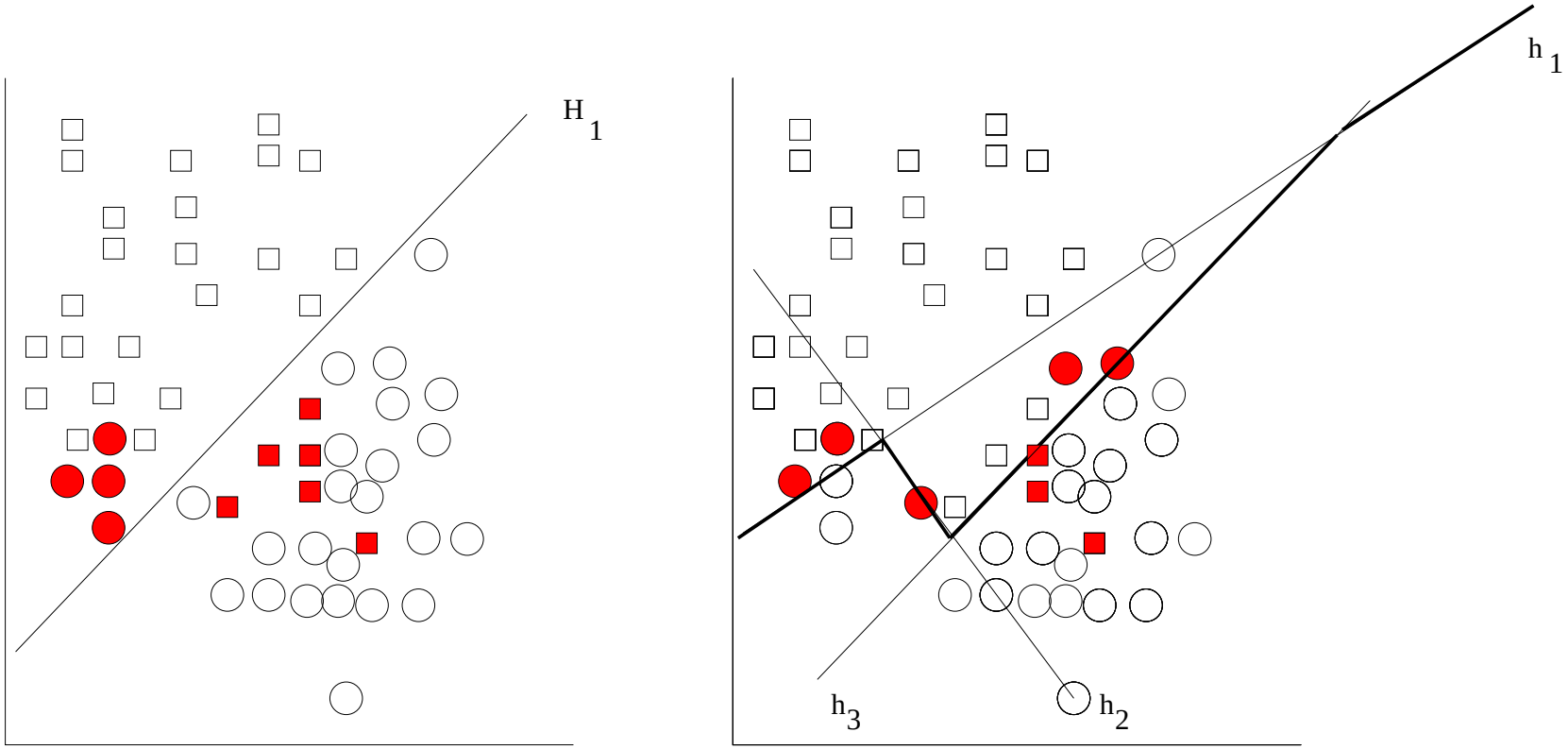
First boosting algorithm (3/4)

Step 3: Generate a third learning sample S_3 by removing from S the instances on which h_1 and h_2 agree. Once again, L is used to induce a third hypothesis h_3.



First boosting algorithm (4/4)

The final hypothesis takes the "majority vote" of h_1, h_2 and h_3.



ADABOOST

ADABOOST ([Freund & Schapire 1996a])

Input: *A learning sample S, a number of iterations T, a weak learner* L
Output: *A global hypothesis H_T*
for all *i from* 1 *to m* **do**
$D_1(\mathbf{x_i}) = 1/m;$

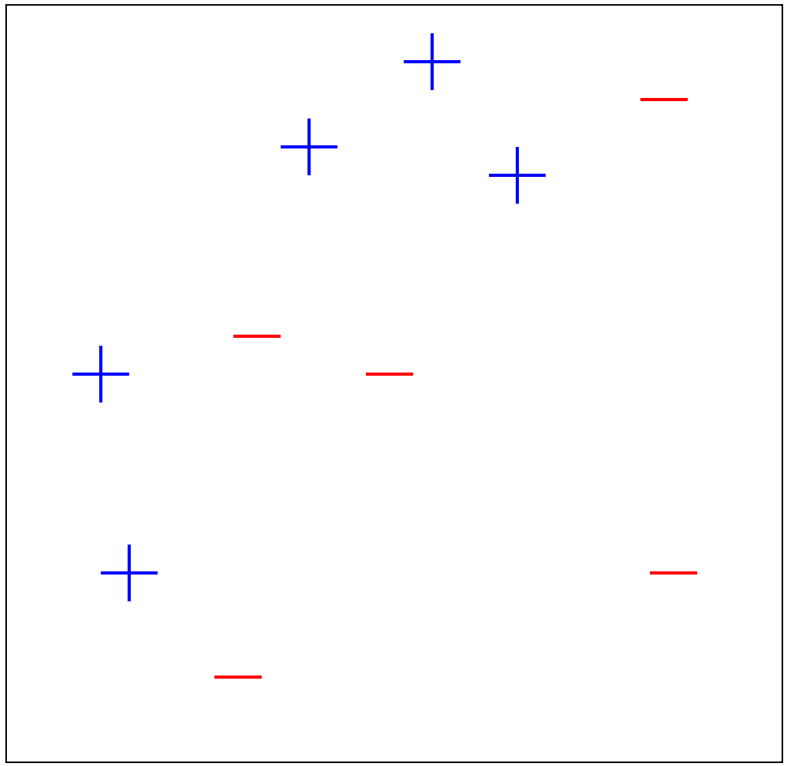
for all *t from* 1 *to T* **do**
$h_t = \mathrm{L}(S, \boldsymbol{D}_t);$
$\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i});$
$\alpha_t = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t};$
for all *i from* 1 *to m* **do**
$D_{t+1}(\mathbf{x_i}) = D_t(\mathbf{x_i}) \exp\left(-\alpha_t y_i h_t(\mathbf{x_i})\right) / Z_t;$
/* *Z_t is a normalization coefficient*/

$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x});$
Return H_T **such that**
$H_T(\mathbf{x}) = sign\left(f(\mathbf{x})\right)$

Toy example (1/5)

Learning sample S

Distribution D_1

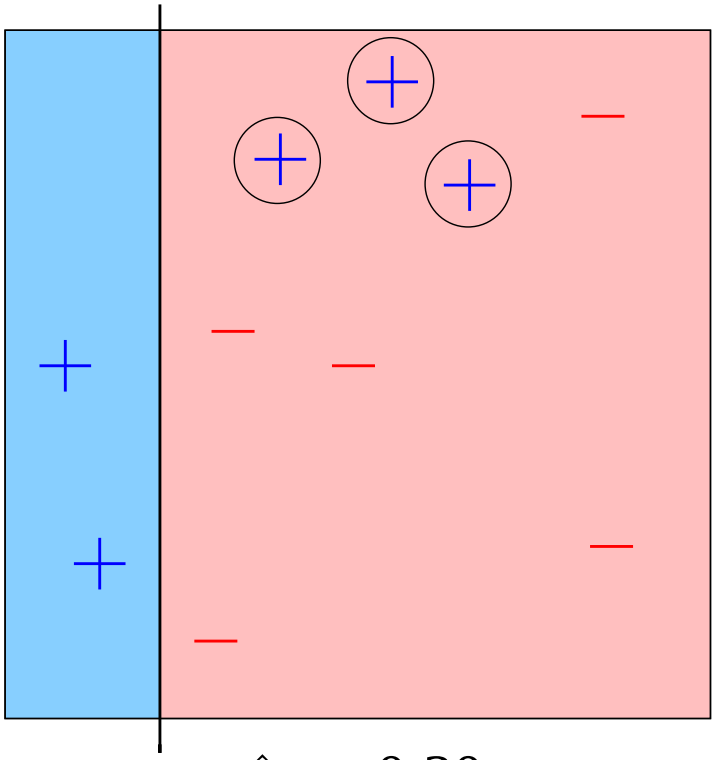


Weak Hypotheses: linear separators parallel to the axis

Toy example (2/5)

Step 1

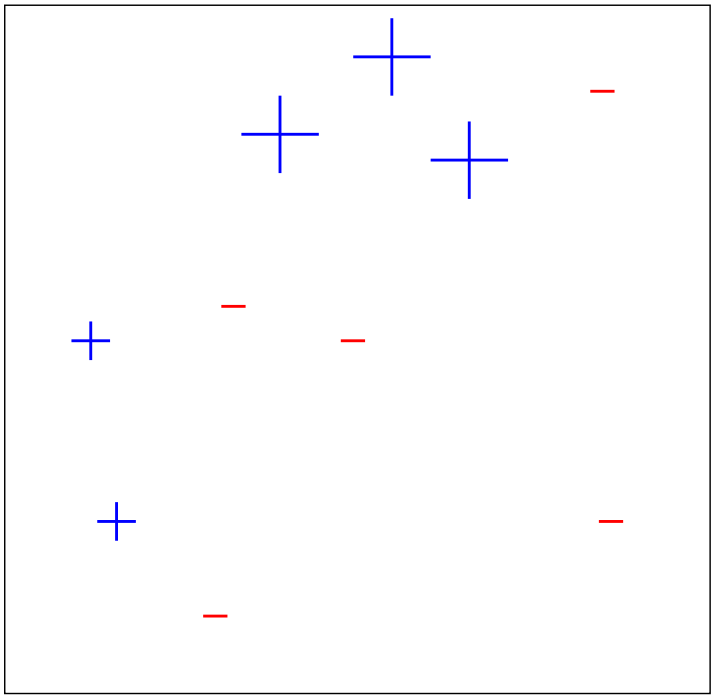
h_1



$\hat{\epsilon}_1 = 0.30$

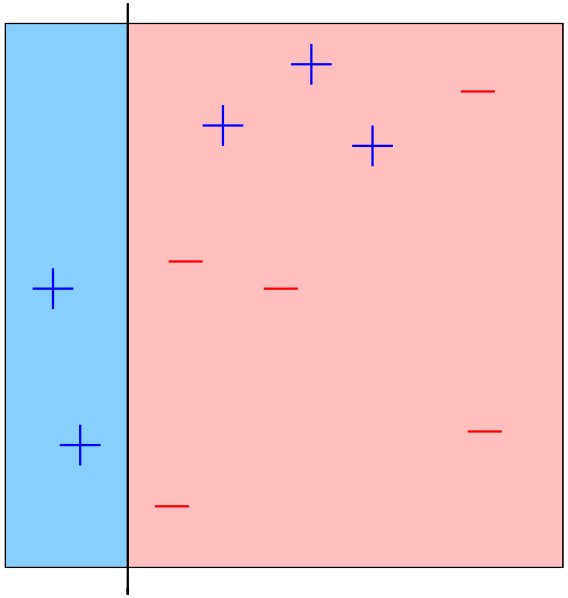
$\alpha_1 = 0.42$

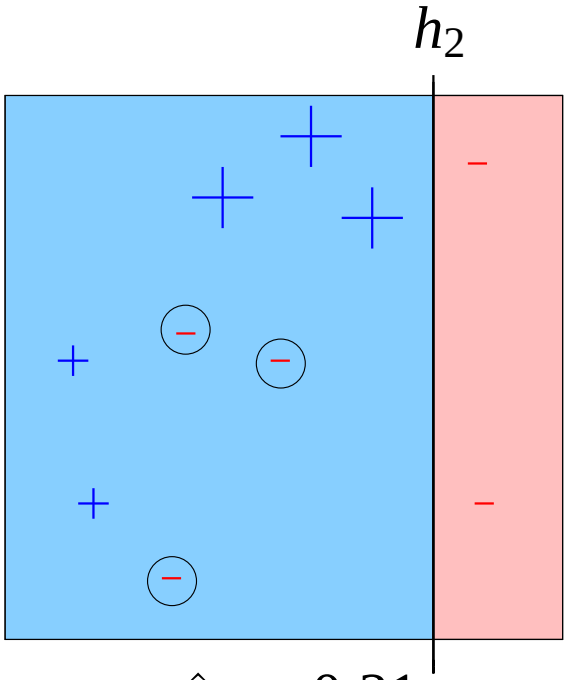
Distribution D_2



Toy example (3/5)

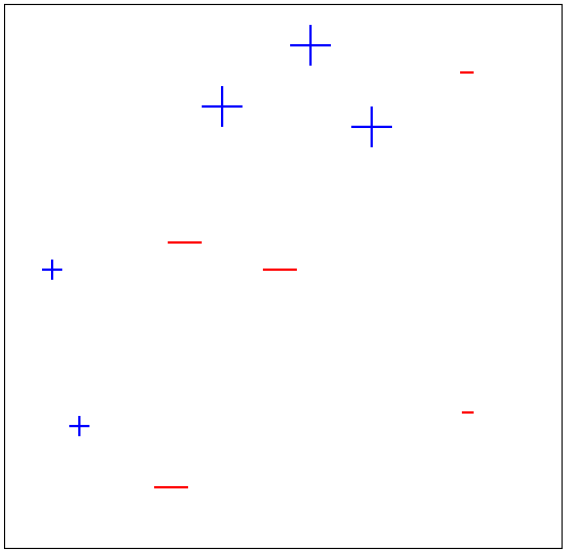
Step 2





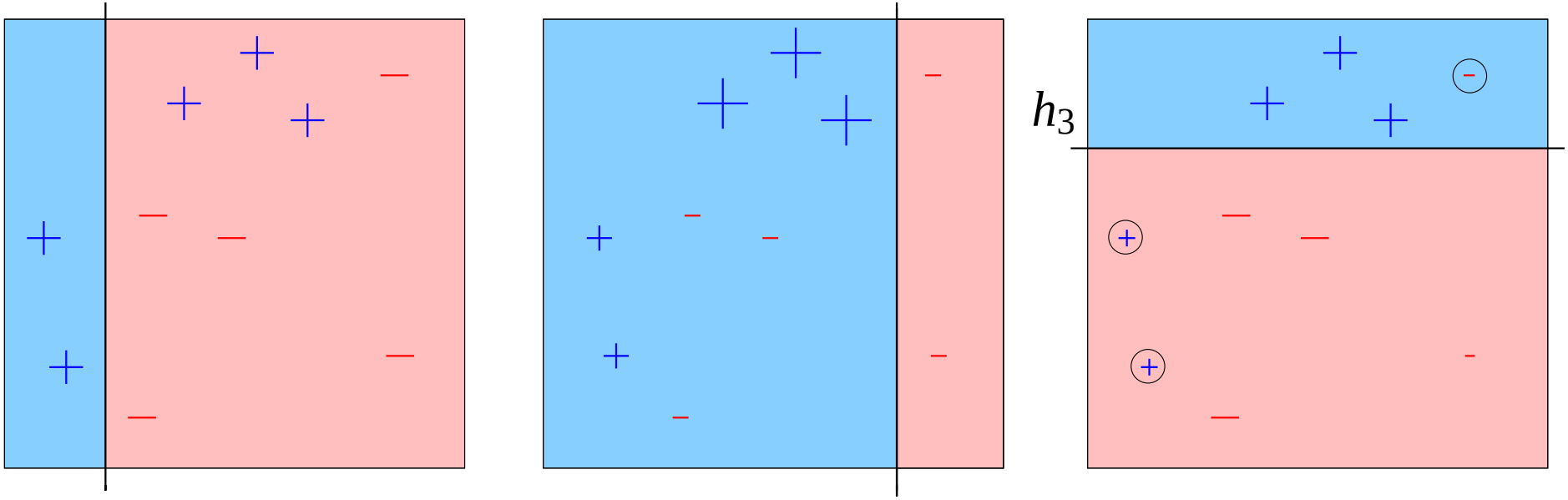
$\hat{\epsilon}_2 = 0.21$
$\alpha_2 = 0.65$

Distribution D_3



Toy example (4/5)

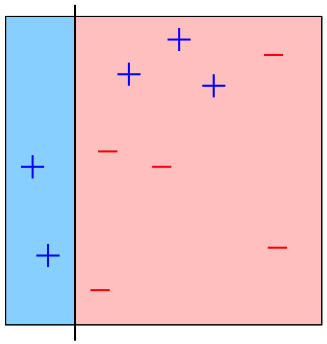
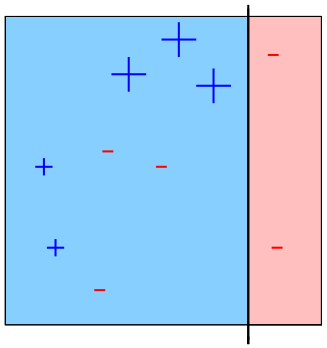
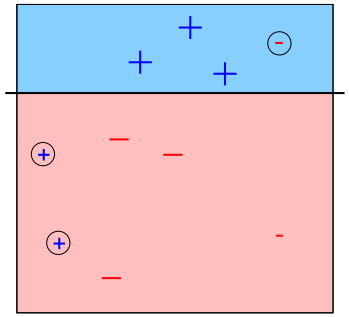
Step 3

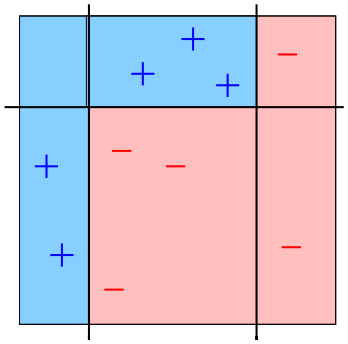


$\hat{\epsilon}_3 = 0.14$
$\alpha_3 = 0.92$

Toy example (5/5)

Final Classifier

$H_{final} = sign(0.42\times$  $+0.65\times$  $+0.92\times$  $)$

$=$ 

Theoretical results on the empirical risk

Theoretical results on the empirical risk

Theorem

Upper bound on the empirical error of H_T

$$\hat{\epsilon}_{H_T} = \frac{1}{m}|\{i/H(\mathbf{x_i}| \neq y_i\}) \leq \frac{1}{m}\sum_i exp(-y_i f(\mathbf{x_i})) = \prod_t Z_t$$

This theorem means that to minimize the empirical error, we have to minimize the product of the Z_t.

Theoretical results on the empirical risk

The previous theorem is proven in two steps.

Step 1: $\hat{\epsilon}_{H_T} \leq \frac{1}{m}\sum_i exp(-y_i f(\mathbf{x_i}))$

Proof.

$$\begin{aligned}
\hat{\epsilon}_{H_T} &= \frac{1}{m}\sum_i \mathbb{1}_{H_T(\mathbf{x_i})\neq y_i} \\
&= \frac{1}{m}\sum_i \mathbb{1}_{y_i f(\mathbf{x_i})<0} \\
&\leq \frac{1}{m}\sum_i exp(-y_i f(\mathbf{x_i}))
\end{aligned}$$

□

Theoretical results on the empirical risk

Step 2: $\frac{1}{m}\sum_i exp(-y_i f(\mathbf{x_i})) = \prod_t Z_t$. To simplify, let us replace $\mathbf{x_i}$ by i.

Proof.

$$
\begin{aligned}
D_{T+1}(i) &= \frac{D_T(i)exp(-\alpha_T y_i h_T(i))}{Z_T} \\
&= \frac{D_1(i)exp(\sum_t -\alpha_t y_i h_t(i))}{\prod_{t=1}^T Z_t} \\
&= \frac{1}{m}\frac{exp(\sum_{t=1}^T -\alpha_t y_i h_t(i))}{\prod_{t=1}^T Z_t} \\
&= \frac{1}{m}\frac{exp(-y_i f(i))}{\prod_{t=1}^T Z_t}
\end{aligned}
$$

□

Theoretical results on the empirical risk

Proof.

since $\sum_i D_{T+1}(i) = 1$ because it is a statistical distribution, we get

$$\prod_{t=1}^{T} Z_t = \frac{1}{m} \sum_i exp(-y_i f(i))$$

□

ADABOOST ([Freund & Schapire 1996a])

Input: *A learning sample S, a number of iterations T, a weak learner* L
Output: *A global hypothesis H_T*
for all *i from* 1 *to m* **do**
$D_1(\mathbf{x_i}) = 1/m;$

for all *t from* 1 *to T* **do**
$h_t = \mathrm{L}(S, \boldsymbol{D}_t);$
$\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i});$
$\alpha_t = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t};$
for all $i = 1$ *from* 1 *to m* **do**
$D_{t+1}(\mathbf{x_i}) = D_t(\mathbf{x_i}) \exp\left(-\alpha_t y_i h_t(\mathbf{x_i})\right) / Z_t;$
/ Z_t is a normalization coefficient*/*

$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x});$
Return H_T **such that**
$H_T(\mathbf{x}) = sign\left(f(\mathbf{x})\right)$

Theoretical results on the empirical risk

Theorem

To minimize Z_t, α_t must be equal to:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \hat{\epsilon}_t}{\hat{\epsilon}_t} \right)$$

Theoretical results on the empirical risk

Proof.

Let us assume that the outputs of h_t are -1 or $+1$. Let W^{-1} and W^{+1} be two sums such that

$$W^b = \sum_{\mathbf{x} \in LS : y(\mathbf{x}) h_t(\mathbf{x}) = b} D_t(\mathbf{x})$$

$$Z_t = \sum_{\mathbf{x} \in LS} D_t(\mathbf{x}) e^{-\alpha_t y(\mathbf{x}) h_t(\mathbf{x})}$$

$$= \sum_b \sum_{\mathbf{x} \in LS : y(\mathbf{x}) h_t(\mathbf{x}) = b} D_t(\mathbf{x}) e^{-\alpha_t b} = W^{-1} e^{\alpha_t} + W^{+1} e^{-\alpha_t}$$

Z_t is minimal when

$$\alpha_t = \frac{1}{2} \ln \left(\frac{W^{+1}}{W^{-1}} \right) = \frac{1}{2} \ln \left(\frac{1 - \hat{\epsilon}_t}{\hat{\epsilon}_t} \right)$$

□

Theoretical results on the empirical risk

Information

Freund and Schapire have proposed another value for α_t, such that:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{W^{+1} + \frac{1}{2} W^0}{W^{-1} + \frac{1}{2} W^0} \right)$$

where W^0 corresponds to the density of the learning examples that can not be classified by h_t (e.g.: points on the separator).
Assuming that half of W^0 is correctly classified, and half is misclassified, we get this new value for α_t calling on the same proof.

ADABOOST ([Freund & Schapire 1996a])

Input: *A learning sample S, a number of iterations T, a weak learner* L
Output: *A global hypothesis H_T*
for all *i from* 1 *to m* **do**
$D_1(\mathbf{x_i}) = 1/m;$

for all *t from* 1 *to T* **do**
$h_t = \mathrm{L}(S, \boldsymbol{D}_t);$
$\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i});$
$\alpha_t = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t};$
for all $i = 1$ *from* 1 *to m* **do**
$D_{t+1}(\mathbf{x_i}) = D_t(\mathbf{x_i}) \exp\left(-\alpha_t y_i h_t(\mathbf{x_i})\right) / Z_t;$
/* *Z_t is a normalization coefficient*/

$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x});$
Return H_T **such that**
$H_T(\mathbf{x}) = sign\left(f(\mathbf{x})\right)$

Theoretical results on the empirical risk

Theorem

Exponential decrease of the empirical risk

$$\prod_t (Z_t) = \prod_t (2\sqrt{\hat{\epsilon}_t(1-\hat{\epsilon}_t)} = \prod_t \sqrt{1-4\gamma_t^2} < \exp\left(-2\sum_t \gamma_t^2\right)$$

where $\hat{\epsilon}_t = \frac{1}{2} - \gamma_t$ *(weak hypothesis)*

- This theorem means that the empirical risk exponentially decreases towards 0 with the number T of iterations.

Theoretical results on the empirical risk

Proof.

$$Z_t = \sum_{\mathbf{x}} D_t(\mathbf{x}) exp^{-\alpha_t y_{\mathbf{x}} h_t(\mathbf{x})} = W^{+1} e^{-\alpha_t} + W^{-1} e^{\alpha_t}$$

$$= (1-\hat{\epsilon}_t) e^{-\frac{1}{2}\ln\left(\frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t}\right)} + \hat{\epsilon}_t e^{\frac{1}{2}\ln\left(\frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t}\right)} = 2\sqrt{\hat{\epsilon}_t(1-\hat{\epsilon}_t)}$$

Then,

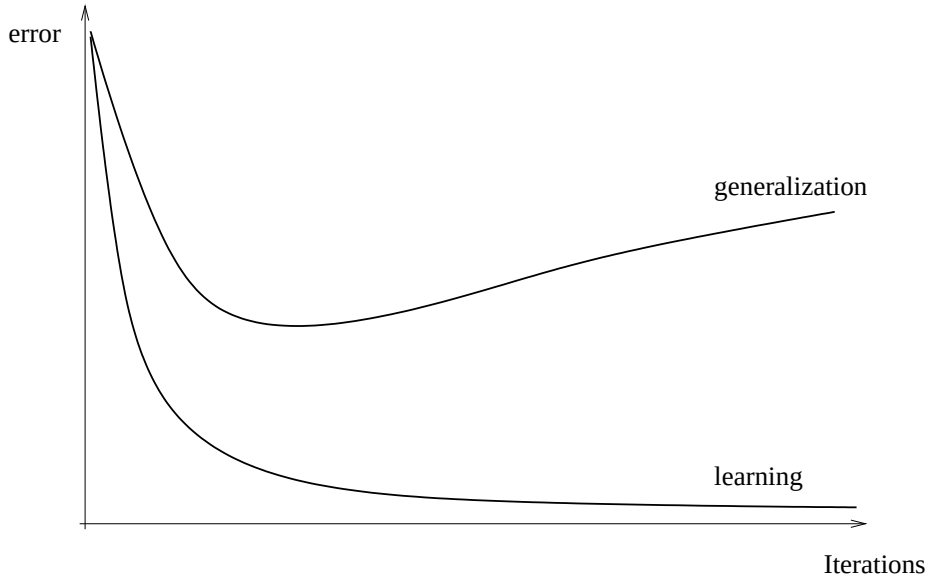
$$\prod_t (Z_t) = \prod_t (2\sqrt{\hat{\epsilon}_t(1-\hat{\epsilon}_t)} = \prod_t \sqrt{4(\frac{1}{2}-\gamma_t)(1-\frac{1}{2}+\gamma_t)} = \prod_t \sqrt{1-4\gamma_t^2}$$

$$= e^{ln(\prod_t \sqrt{1-4\gamma_t^2})} = e^{\frac{1}{2}\sum_t ln(1-4\gamma_t^2)} \leq e^{-2\sum_t \gamma_t^2}$$

because $ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \ldots - \frac{x^n}{n} + 0(x^{n+1})$

□

Expected behavior of boosting



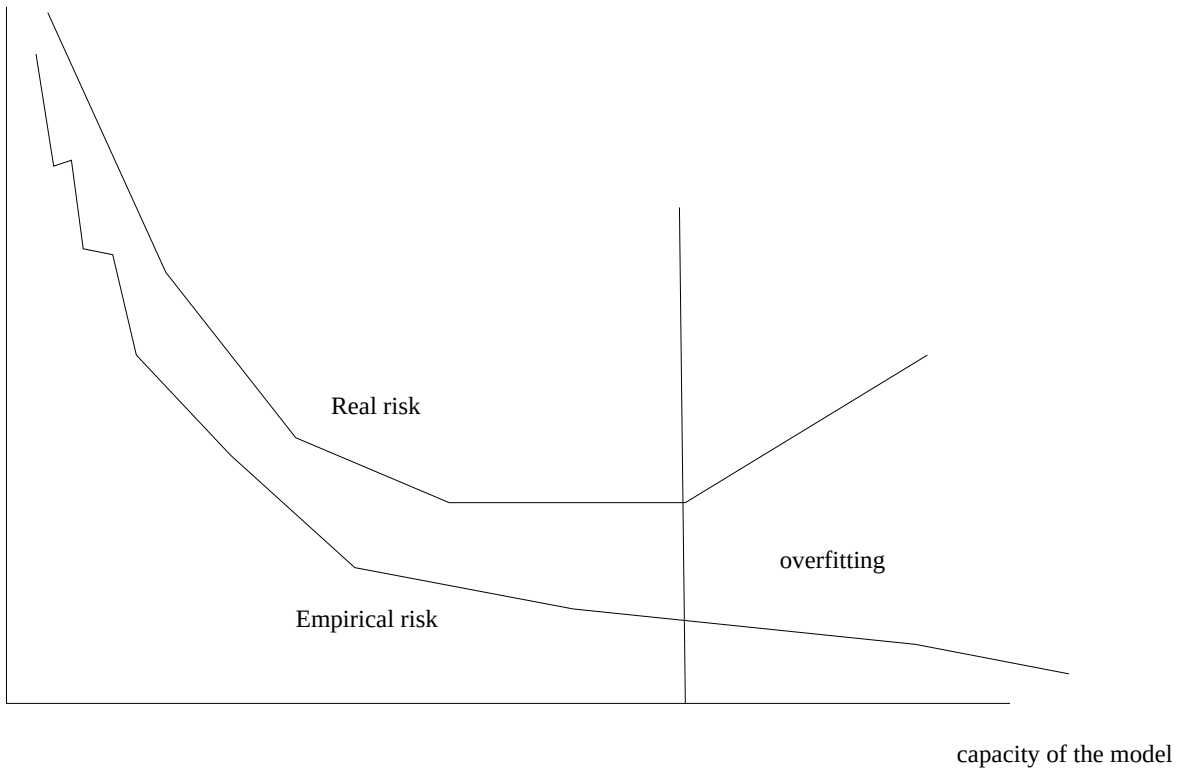
Expected behavior

- $\hat{\epsilon}_{H_T}$ *decreases towards (eventually) 0.*
- ϵ_{H_T} *first decreases; then* H_T *becomes too complex* $\rightarrow$ *overfitting*

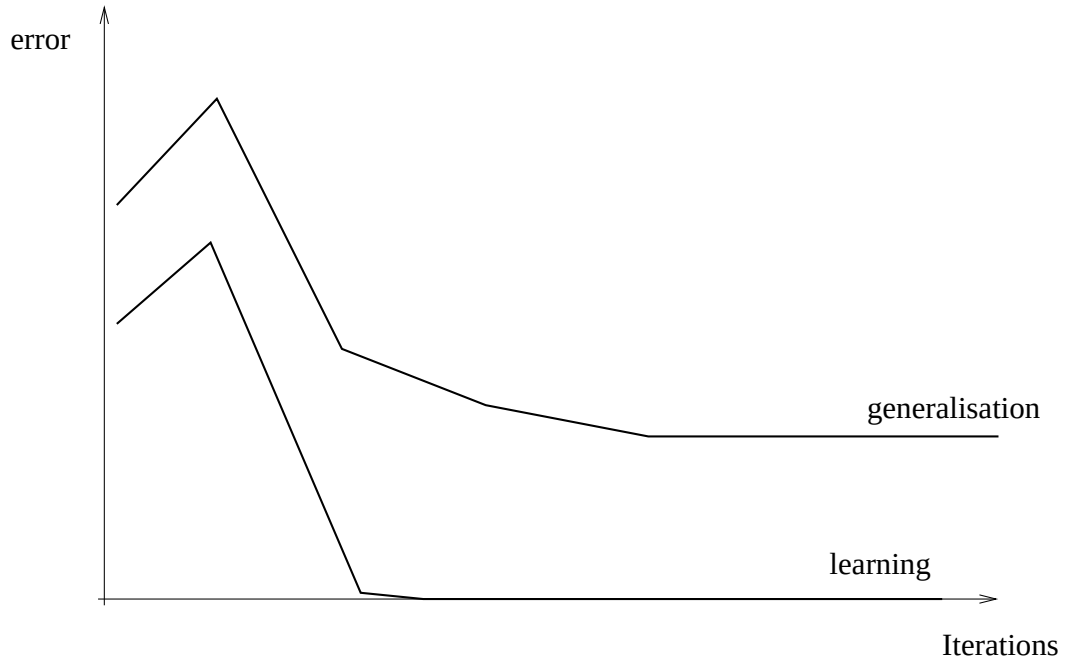
Overfitting

Definition

Overfitting: artificially good agreement with the learning data.



Observed behavior of boosting



Observed behavior

- $\hat{\epsilon}_{H_T}$ *decreases towards (eventually) 0.*
- ϵ_{H_T} *drops and continues to decrease even when* $\hat{\epsilon}_T$ *has reached 0!!*

Experimental proof

Experimental proof with ADABOOST

What are the theoretical reasons that justify this behavior?

Explanation in terms of margins of the training examples

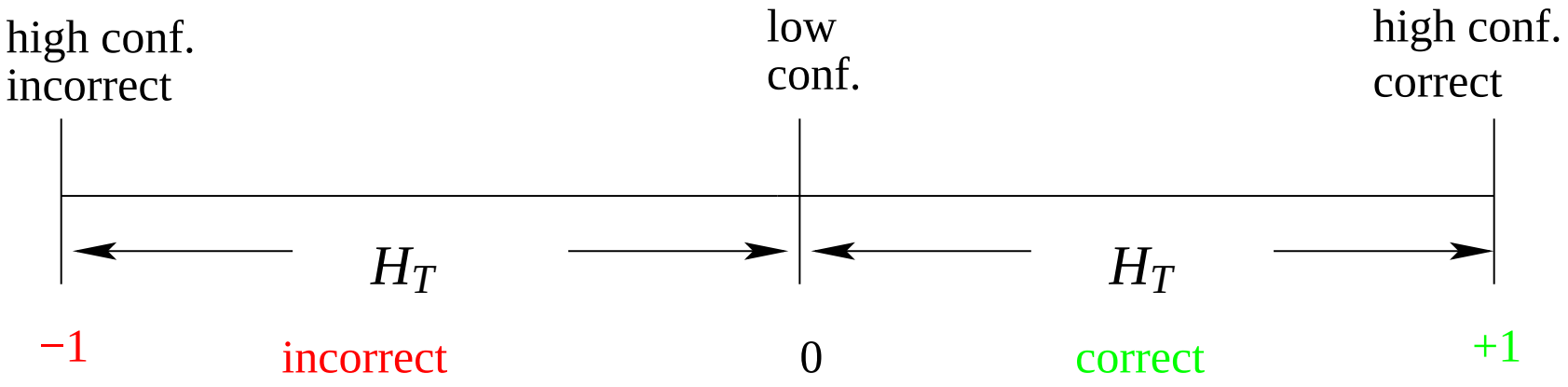
- $\hat{\epsilon}_{H_T}$ only takes into account the fact that the classification of an example is correct or not correct.
- $\hat{\epsilon}_{H_T}$ should also take into account the confidence into the classification of each example.

Definition

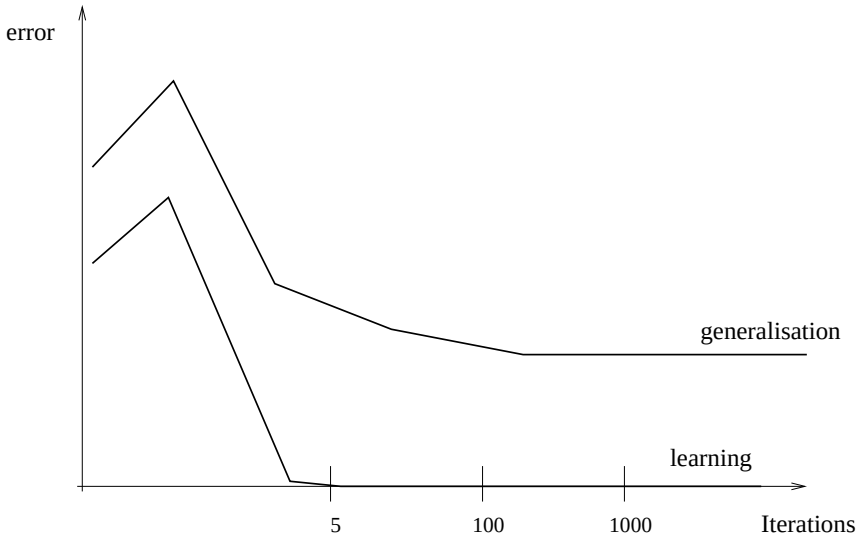
The margin of an example is defined to be

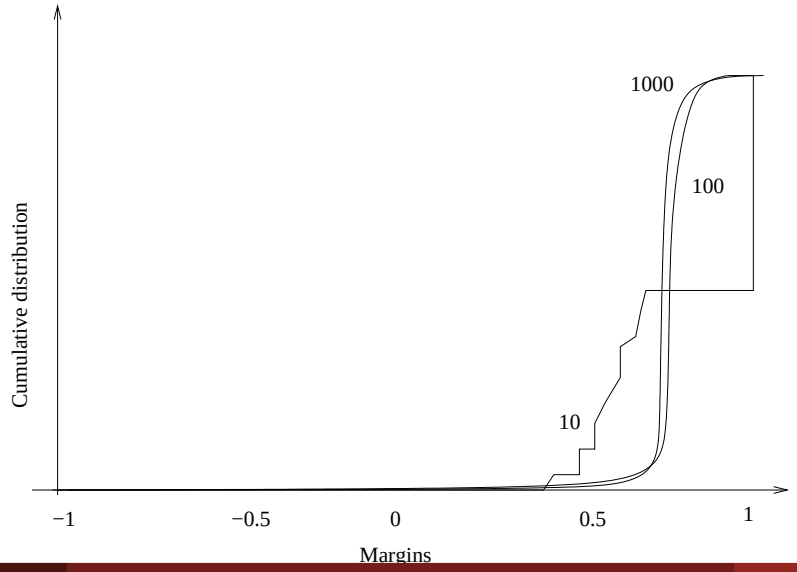
$$margin(\mathbf{x}) = \frac{y \sum_t \alpha_t h_t(\mathbf{x})}{\sum_t \alpha_t}$$

Explanation in terms of margins of the training examples



Empirical observation





Intuition

- Larger margins on the training set translate into a superior upper bound on the generalization error.
- Boosting tends to increase the margins of the learning examples
- Despite the increase of its complexity, the final performing classifier is becoming easier and easier to build because of the increase of the margins.

Theoretical results in generalization

Theoretical results in generalization

Theorem ([Schapire, Freund, Bartlett & Lee 1997])

Let $\mathcal{H}$ be a class of classifiers with VC dim d_h (i.e. the capacity of $\mathcal{H}$). For any $\delta > 0$ and $\theta > 0$, with probability $1 - \delta$, any classifier ensemble H_T built from m learning examples satisfies:

$$\epsilon_{H_T} \leq \hat{Pr}(margin(\mathbf{x}) \leq \theta) + \mathcal{O}\left(\sqrt{\frac{d_h}{m}\frac{log^2(m/d_h)}{\theta^2} + log(1/\delta)}\right)$$

- This bound depends on:
  - constant parameters m, d_h, θ and δ.
  - the distribution of the margins of the learning examples $\hat{Pr}$.

Theorem

$\hat{Pr}(margin(\mathbf{x}) \leq \theta)$ exponentially decreases towards 0 with T.

Margin maximization

Formally, [Schapire, Freund, Bartlett & Lee 1997] proved the following theorem:

Theorem

Let $margin(\mathbf{x}) = yf(\mathbf{x})$ *the margin of an example:*

$$P(yf(\mathbf{x}) \leq \theta) \leq 2^T \prod_t \sqrt{\hat{\epsilon}_t^{1-\theta}(1-\hat{\epsilon}_t)^{1+\theta}}$$

If $\hat{\epsilon}_t \leq \frac{1}{2} - \gamma$*, then* $\forall \theta < \gamma$*, this bound exponentially with* T*.*

Margin maximization

Proof.

If $yf(\mathbf{x}) \leq \theta$ then $y \sum_t \alpha_t h_t(\mathbf{x}) \leq \theta \sum_t \alpha_t$

$$\Rightarrow exp^{-y \sum_t \alpha_t h_t(\mathbf{x}) + \theta \sum_t \alpha_t} \geq 1$$

$$\begin{aligned}
\hat{Pr}(yf(\mathbf{x}) \leq \theta) &\leq \hat{Pr}(exp^{-y \sum_t \alpha_t h_t(\mathbf{x}) + \theta \sum_t \alpha_t}) \\
&= \frac{exp^{\theta \sum_t \alpha_t}}{m} \sum_i exp^{-y_i \sum_t \alpha_t h_t(\mathbf{x_i})} \\
&= exp^{\theta \sum_t \alpha_t} \prod_t Z_t
\end{aligned}$$

□

Margin maximization

Proof.

By replacing α_t and Z_t by their expressions with $\hat{\epsilon}_t$, we get:

$$
\begin{aligned}
&= exp^{\sum_t \theta \frac{1}{2} ln(\frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t})} 2^T \prod_t \sqrt{\hat{\epsilon}_t(1-\hat{\epsilon}_t)} \\
&= 2^T \prod \sqrt{\frac{(1-\hat{\epsilon}_t)^\theta \hat{\epsilon}_t (1-\hat{\epsilon}_t)}{\hat{\epsilon}_t^\theta}} = 2^T \prod_t \sqrt{\hat{\epsilon}_t^{1-\theta}(1-\hat{\epsilon}_t)^{1+\theta}}
\end{aligned}
$$

Considering that $\hat{\epsilon}_t \leq \frac{1}{2} - \gamma$, this upper bound can be rewritten such that:

$$
\left(\sqrt{(1-2\gamma)^{1-\theta}(1+2\gamma)^{1+\theta}} \right)^T
$$

If $\theta < \gamma$ then we can prove that the expression between brackets is < 1 and so the probability to have a small margin decreases with T.

□

First conclusions

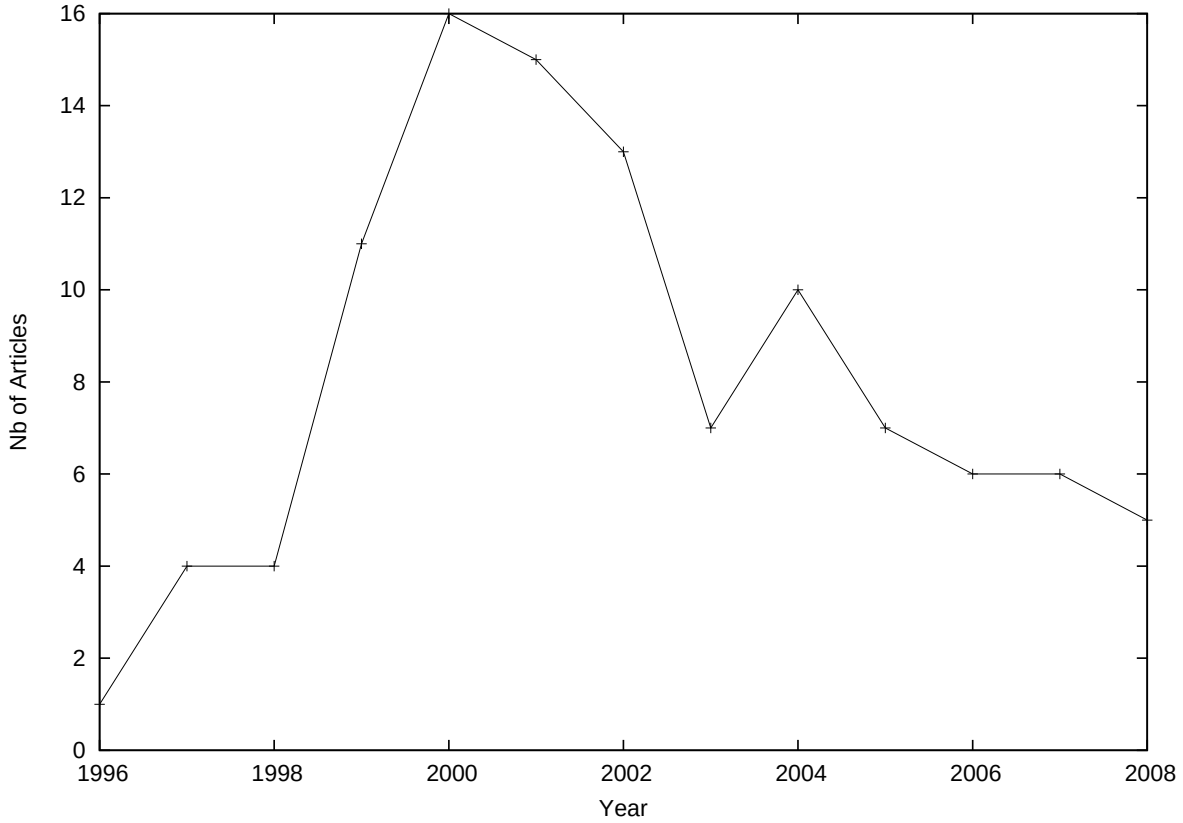
- ADABOOST works in practice...
- and is theoretically well-founded!

Question

What has been the impact of boosting in the machine learning community?

Impact of boosting in Machine Learning

Number of articles presented at important ML conferences such as ICML, ECML, COLT, ALT from 1996 to 2008 whose title contained the word “boosting”.



Impact of boosting in Machine Learning

About R.Schapire... (Google Scholar)

- **h-index** (**h-number**) of R. Schapire: 39
- **Nb of citations**: 13953
- **Maximal nb of citations for an article**: 2126
- **Most cited article**: *Experiments with a new boosting algorithm*

`www.boosting.org` (Gunnar Ratsch)

Some platforms for testing boosting

Some platforms:

- Yoav Freund's home page
  `http://www.cs.ucsd.edu/~yfreund/adaboost`
- Ran El-Yaniv's home page
  `http://www.cs.technion.ac.il/~rani/LocBoost/`
- WEKA (University of Waikato)
  `http://www.cs.waikato.ac.nz/ml/weka/`

SVM versus Boosting

SVM versus Boosting [Freund & Schapire 1999]

SVM vs Boosting

SVMs and ADABOOST *search for a linear separator in a high dimensional space.*

SVM versus Boosting [Freund & Schapire 1999]

SVM vs Boosting

Overfitting is avoided thanks to the margin maximization.

- Let $\mathbf{h}(\mathbf{x})$ be the vector of weak hypotheses $(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x}))$. We aim at selecting the vector of coefficients $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_T)$.
- Boosting chooses the coefficients α_t such that the bound $\hat{Pr}(margin(\mathbf{x}) \leq \theta) + O(\sqrt{\frac{d_h}{m\theta}})$ is minimized.
- Freund and Schapire have shown that to deal with this problem, we have to increase the margin of each learning example.
- Therefore, boosting also aims at maximizing the minimal margin of the learning examples.

SVM versus Boosting [Freund & Schapire 1999]

- $margin(\mathbf{x}) = \frac{y \sum_t \alpha_t h_t(\mathbf{x})}{\sum_t \alpha_t}$.
- The shared goal is to maximize the minimal margin:

$$\max_{\alpha} \min_{i} \frac{(\alpha.\mathbf{h}(\mathbf{x_i}))y_i}{||\alpha||.||\mathbf{h}(\mathbf{x_i})||}$$

where

- for boosting, the norms of the denominator are defined as follows: $||\alpha||_1 = \sum_t |\alpha_t|$ and $||\mathbf{h}(\mathbf{x})||_\infty = max_t |h_t(\mathbf{x})|$
(NB: if $h_t(\mathbf{x}) \in \{-1, +1\}$ then $||\mathbf{h}(\mathbf{x})||_\infty = 1$)
- for SVMs, $||\alpha||_2 = \sqrt{\sum_t \alpha_t^2}$ and $||\mathbf{h}(\mathbf{x})||_2 = \sqrt{\sum_t h_t(\mathbf{x})^2}$.

SVM versus Boosting [Freund & Schapire 1999]

Despite these resemblances...

1. **The norms are different**. The difference between the L1-norm, L2-norm and l_∞ is not significant when one considers low dimensional space. However, this is not the case with boosting and overall with SVMs.
2. **The computational constraints are different**
   - SVMs correspond to a quadratic programming problem.
   - Boosting is a matter for linear programming.
3. **The management of large spaces is different**.
   - SVM exploit *kernels* that allow to perform calculations in small spaces that are equivalent to inner products in potential very large spaces.
   - Boosting uses a greedy strategy. Each new weak hypothesis of $\mathbf{h}(\mathbf{x})$ is correlated with the class to predict y.

Boosting and Game Theory

Edge versus Margin

In a boosting algorithm, we optimize two sets of weights:

- a distribution D_t over the learning examples.
- a distribution α_t over the weak hypotheses.

Edge versus Margin

Definition

The *Edge* of a hypothesis for a given distribution D_t on the examples is

$$\sum_{i=1}^{m} y_i h_t(x_i) \times D_t(x_i)$$

Definition

The *Margin* of a learning example x_n for a current distribution α_t on the learned hypotheses is

$$\sum_{t=1}^{T} y_n h_t(x_n) \times \alpha_t$$

Edge versus Margin

The objectives of boosting:

- Edge
  - Edges of past hypotheses should be small after update. In other words, the new hypothesis must learn something new.
  - Therefore,we aim at **minimizing the maximum edge** of past hypotheses.
- Margin
  - Choose a convex combination of weak hypotheses that **maximizes the minimum margin**.

Question

Connection between objectives?

Edge versus Margin

Duality

Linear Programming Duality

$$\min \max Edge = \max \min Margin$$

$$min_D max_{t=1,\ldots,T-1} \sum_{i=1}^{m} y_i h_t(x_i) D_t(x_i) = max_\alpha min_{i=1,\ldots,m} \sum_{t=1}^{T-1} y_i h_t(x_i) \alpha_t$$

$\rightarrow$ Adaboost and Game Theory

Game Theory

[Freund & Schapire 1996b]

Definition

- A game between two players is defined by a matrix M.
- The first player is called "row player" **RP**.
- The second player is called "column player" **CP**.
- To play, **RP** chooses a row i of M and **simultaneously CP** plays a column j.
- The cell $M(i,j)$ that is played is the loss of **RP**.

Game Theory

Example

The game Rock-Paper-Scissors

M	*rock*	*paper*	*scissors*
rock	*0*	*1*	*-1*
paper	*-1*	*0*	*1*
scissors	*1*	*-1*	*0*

- *RP's goal: minimize loss* $M(i,j)$.
- *CP's goal: maximize this loss.*
- *Zero sum game*

Game Theory

Definition

Randomized play: players choose distributions P and Q over rows and columns.
The learner's (expected) loss is:

$$\sum_{i,j} P(i)M(i,j)Q(j) = P^T MQ$$

For the sake of simplicity, this quantity if often written $M(P,Q)$.

Sequential game

MinMax Strategy

If **CP** *chooses his strategy Q after* **RP**

- **CP** *will choose Q to maximize* $M(P,Q)$.
- *Knowing it,* **RP** *chooses his strategy P to minimize his maximal loss, i.e.* $min_P max_Q M(P,Q)$.
- *Such strategy* P^* *is called minmax strategy.*

MaxMin Strategy

If **CP** *plays at first,* **RP** *can use this knowledge in his strategy.*

- **CP** *aims at maximizing the minimal loss of* **RP**.
- *Therefore, the loss is* $max_Q min_P M(P,Q)$.
- *This strategy* Q^* *is called maxmin strategy.*

The MinMax Theorem (Von Neumann)

- We expect the player who chooses its strategy last to have the advantage since it plays knowing its opponent's strategy exactly. Thus, we expect $max_Q min_P M(P,Q) \leq min_P max_Q M(P,Q)$
- Surprisingly, it turns out not to matter which player plays first. Von Neumann's well-known minmax theorem states that the outcome is the same in either case so that

Theorem (Von Neuman)

$$max_Q min_P M(P,Q) = min_P max_Q M(P,Q), \forall M.$$

The MinMax Theorem (Von Neumann)

Theorem (Von Neuman)

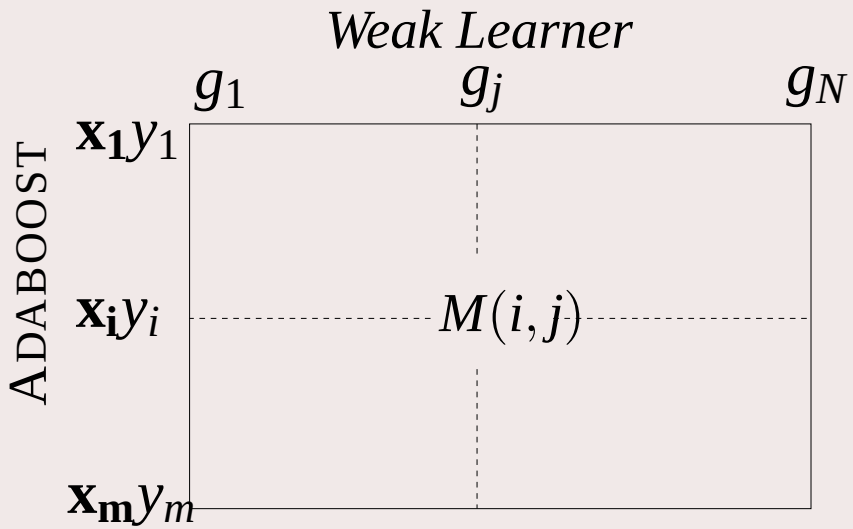
$$max_Q min_P M(P,Q) = min_P max_Q M(P,Q), \forall M.$$

The common value v of the two sides of the equality is called the *value of the game* M and can be found by linear programming.

- The **RP** has a (min-max) strategy P^* such that regardless of the strategy Q played by **CP**, the loss suffered $M(P^*,Q) \leq v$.
- Symmetrically, it means that **CP** has a (max-min) strategy Q^* such that regardless of the strategy P played by **RP** the loss $M(P,Q^*) \geq v$.

The boosting game

- *Let $g_1, \ldots, g_N$ be the space of all weak classifiers.*
- **CP** *is the weak learner, and* **RP** *is the booster.*
- *Matrix M is defined as follows:*
  - *a row is a learning example, a column is weak classifier*
  - $M(i,j) = \begin{cases} 1 & \text{if} \quad y_i = g_j(\mathbf{x_i}) \\ -1 & \text{otherwise} \end{cases}$



The Boosting Game

If $\forall$ distributions P over the examples, $\exists g_j$ satisfying the weak hypothesis, then:

- $min_P max_Q M(P, j) \geq \frac{1}{2} + \gamma$
- and then (by minmax theorem): $max_Q min_i M(i, Q) \geq \frac{1}{2} + \gamma$

Theorem

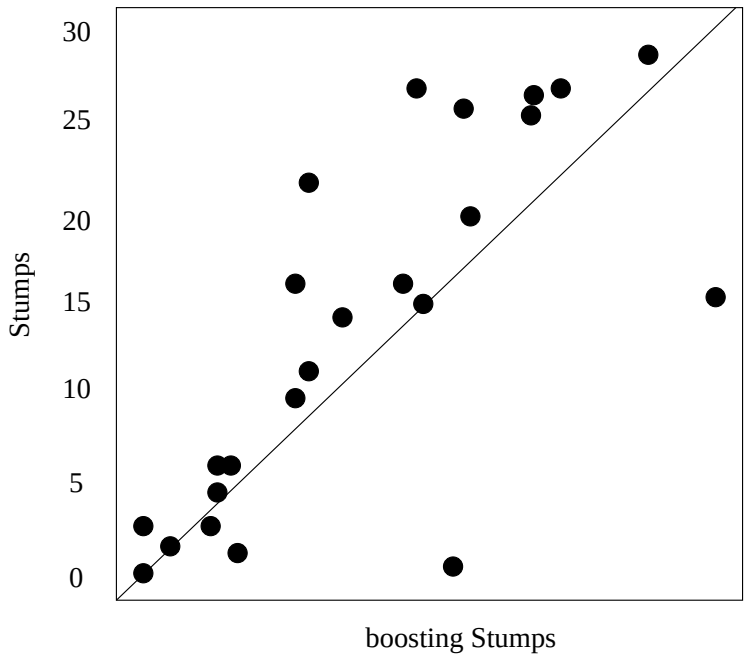
Applying Von Neuman's theorem, it exists a weighted majority of classifiers which correctly classifies all examples with positive margin.

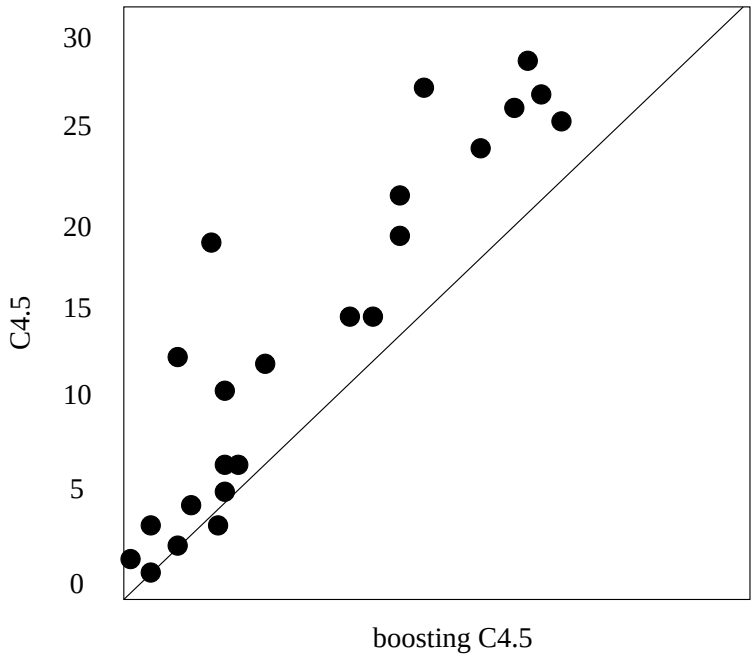
- Optimal margin $\rightarrow$ "value" of the game.
- Distribution over examples converges to (approximate) *minmax* strategy for boosting game.
- Weights on weak classifiers converge to (approximate) *maxmin* strategy.

Advantages -Caveats of ADABOOST

Some experimental results...

Behavior of ADABOOST on a sample of 25 databases from the UCI Machine Learning Repository





http://www.ics.uci.edu/~mlearn/

Practical advantages of ADABOOST

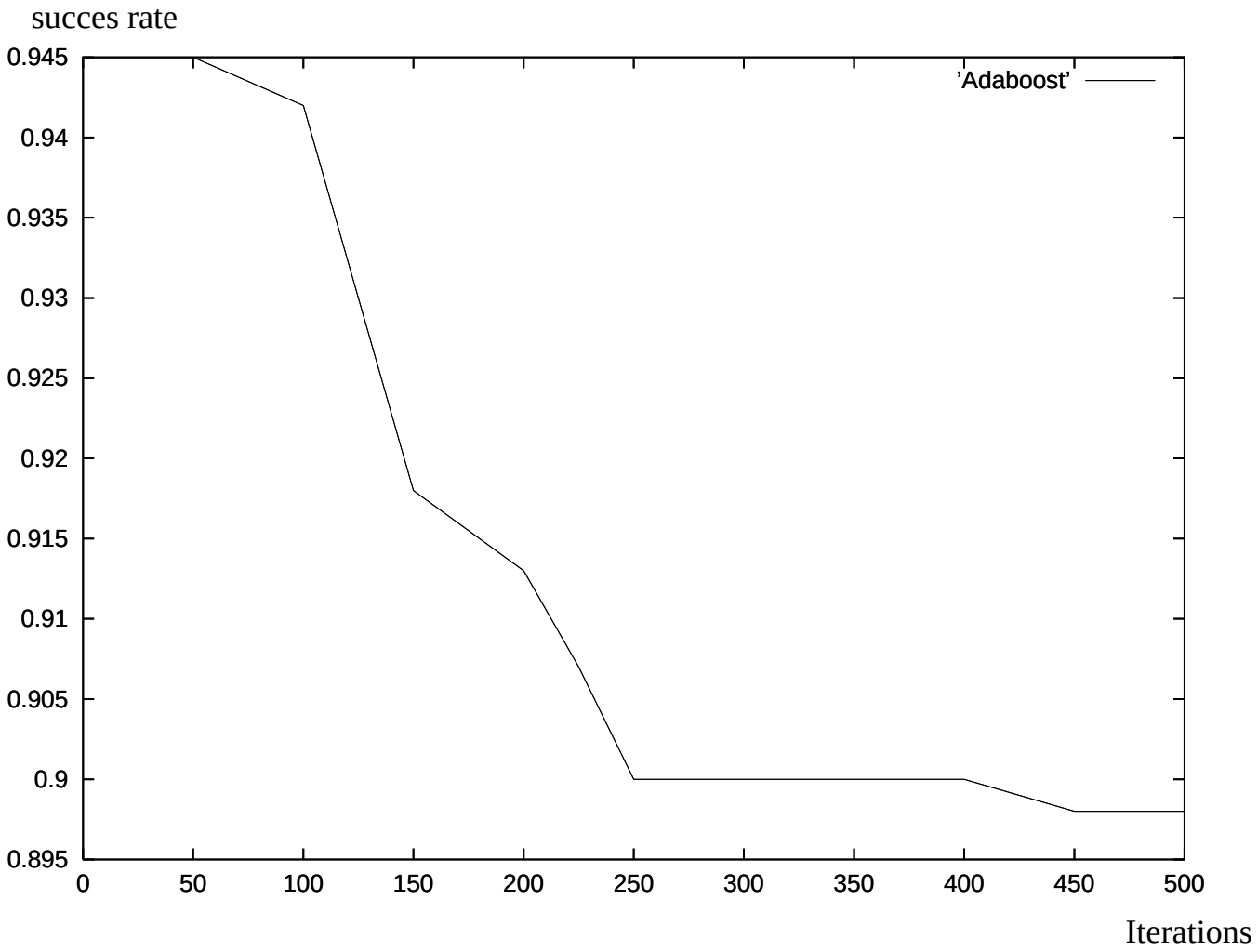
- Fast.
- Simple and easy to program.
- No parameters to tune (except T).
- Flexible - can combine with any learning algorithm.
- Provably effective.

Caveats

- Performances of ADABOOST depends on data and weak learner:
- consistent with theory, ADABOOST can fail if:
  - weak classifiers too complex (*e.g.* kNN, ID3) $\rightarrow$ overfitting.
  - weak classifier too weak ($\gamma_t \rightarrow 0$ too quickly) $\rightarrow$ underfitting or low margins + overfitting.
- empirically, ADABOOST seems especially susceptible to the presence of noise:
  - Presence of outliers $\rightarrow$ exponential increase of their weights $\rightarrow$ overfitting.
  - Presence of large bayesian error $\rightarrow$ slows down the convergence.

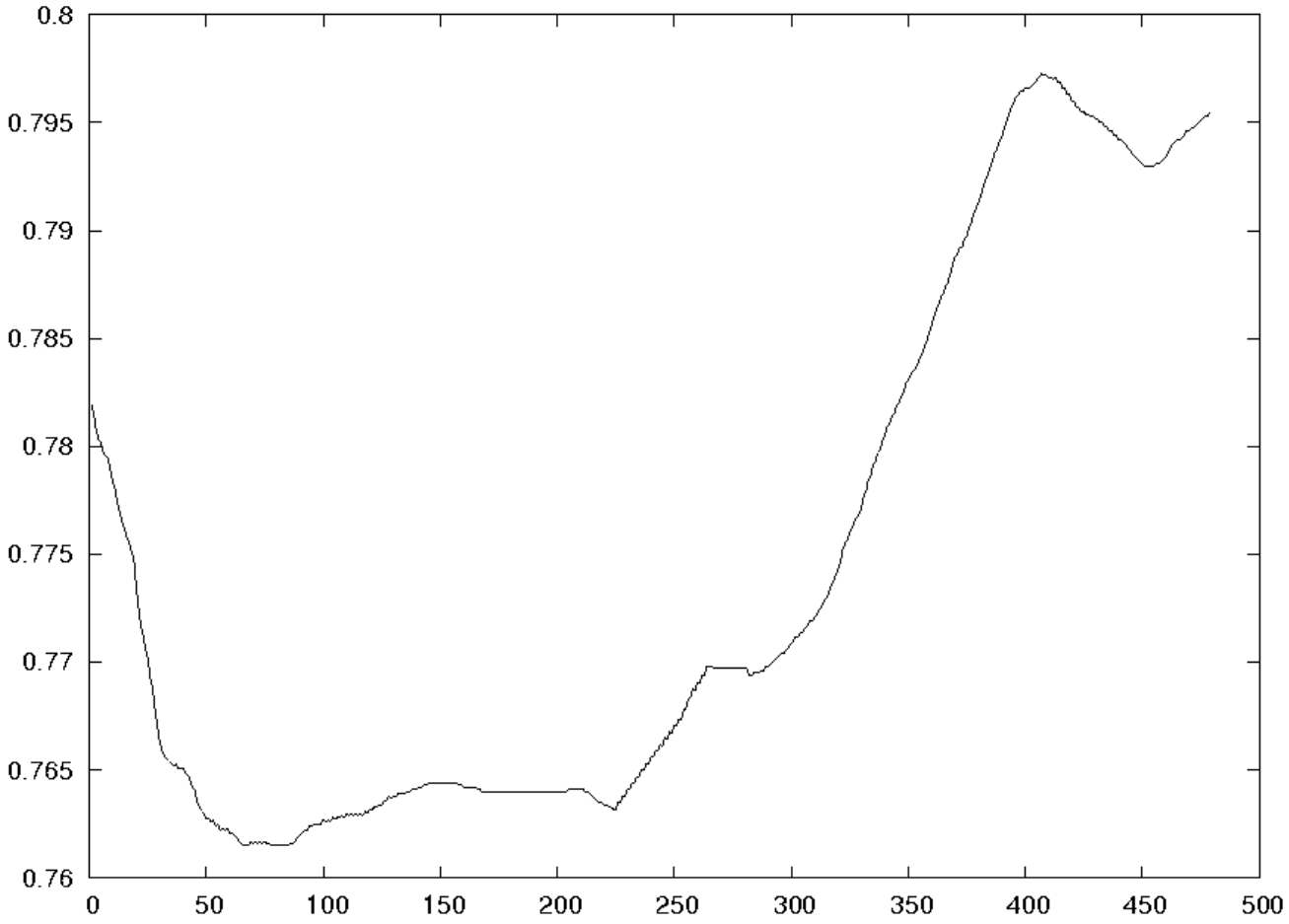
Impact of outliers on ADABOOST behavior

Example: let us run ADABOOST on a linearly separable problem containing 5% of noise (by swapping the original label).



Hard predictions can slow learning

Example: let us run ADABOOST on an artificial sample whose bayesian error is about 20%.



Demo with WEKA

Bibliography

Breiman, L. (1996).
Bagging Predictors, *Machine Learning*, 24(2):pp. 123–140.

Breiman, L. (2001).
Random Forests, *Machine Learning*, 45(1):pp. 4–32.

Dietterich (2002).
Ensemble Methods in Machine Learning, *Lecture Notes in Computer Science*, 1857:pp. 1–15.

Domingo C. & Watanabe O. (2000).
MadaBoost: A Modification of AdaBoost *Proc. 13th Annu. Conference on Comput. Learning Theory*, (2000), 180-189.

Drucker, H., Schapire R., & Simard, P. (1992).
Improving Performance in Neural Networks Using a Boosting Algorithm *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, (1992), 42-49.

Džeroski S. & Ženko B. (2004).
Is Combining Classifiers with Stacking Better than Selecting the Best One? *Mach. Learn.*, 54(3):pp. 255–273.

Freund, Y. (1990).
Boosting a weak learning algorithm by majority, *COLT '90: Proceedings of the third annual workshop on Computational learning theory*, (1990), 202-216.

Freund, Y. (1999).
An adaptive version of the boost by majority algorithm, *Proc. of the twelfth annual conference on Computational learning theory*, (1999), 102-113.

Freund, Y. & Schapire, R. (1996).
Experiments with a New Boosting Algorithm, *International Conference on Machine Learning*, 148-156.

Bibliography

Freund, Y. & Schapire, R. (1996).
Game Theory, On-Line Prediction and Boosting, *COmputational Learning Theory COLT'96*, 325-332.

Freund, Y. & Schapire, R. (1999).
A short introduction to boosting, *J. Japan. Soc. for Artif. Intel.*, 14(5) (1999), 771-780. 11.

Gamma, J. & Brazdil, P. (2000).
Cascade Generalization, *Machine Learning*, 41(3):pp. 315–343.

Kearns, M. & Valiant, L.G. (1988).
Learning Boolean formulae or finite automata is as hard as factoring, *Technical Report TR-14-88, Harvard University Aiken Computation Laboratory*.

Nock, R. & Sebban, M. (2001)
A Bayesian Boosting Theorem, *International Journal of Pattern Recognition Letters*, (2001), (22), 3-4, 413-419.

Ratsch G., Onoda T. & Muller K.R. (1999)
Regularizing AdaBoost, *Proc. of the 1998 conference on Advances in neural information processing systems II*, (1999), 564-570.

Schapire, R. Freund, Y. Bartlett, P. & Lee, W.S. (1997)
Boosting the margin: a new explanation for the effectiveness of voting methods, Proc. 14th International Conference on Machine Learning, 322–330.

Schapire, R. & Singer, Y. (1999)
Improved Boosting Using Confidence-rated Predictions, Machine Learning, 37(3):297–336.

Wolpert, D.H. (1992).
Stacked Generalization, *Neural Networks*, 5:pp. 241–259.