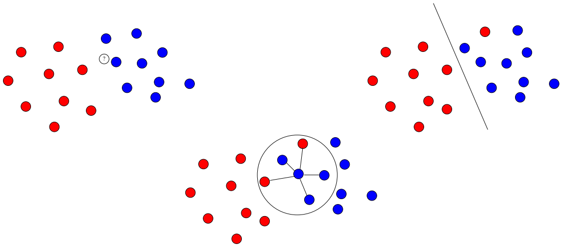
Ensemble Methods

Marc Sebban

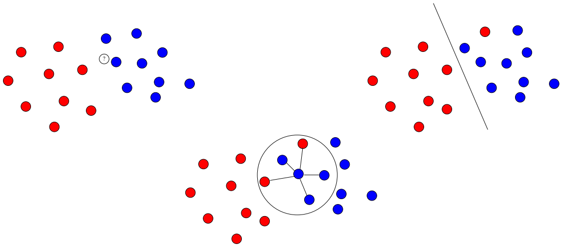
Hubert Curien lab, UMR CNRS 5516
University of Jean Monnet Saint-Étienne (France)

Outline

Many classifiers can be induced from the same task

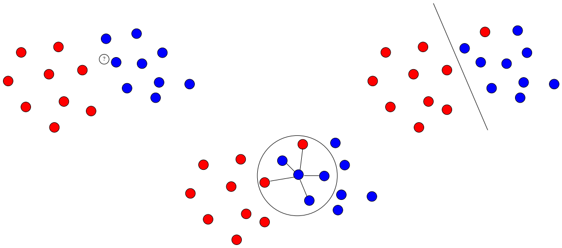


Many classifiers can be induced from the same task



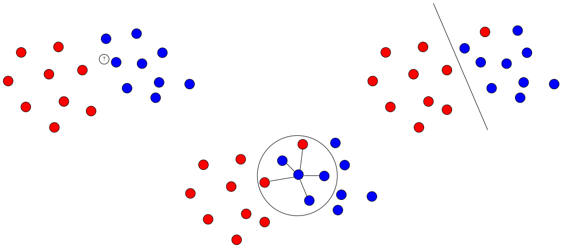
- Different learning algorithms (*e.g.* k-NNs, linear separator, decision trees, SVMs, etc.).

Many classifiers can be induced from the same task



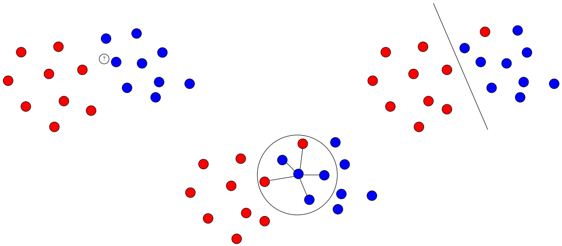
- Different learning algorithms (*e.g.* k-NNs, linear separator, decision trees, SVMs, etc.).
- Different hyperparameters (*e.g.* number of neighbors k, regularization parameter λ, learning rate α, degree d of the polynomial, etc.).

Many classifiers can be induced from the same task



- Different learning algorithms (*e.g.* k-NNs, linear separator, decision trees, SVMs, etc.).
- Different hyperparameters (*e.g.* number of neighbors k, regularization parameter λ, learning rate α, degree d of the polynomial, etc.).
- Different (randomly drawn) training sets S.

Many classifiers can be induced from the same task



- Different learning algorithms (*e.g.* k-NNs, linear separator, decision trees, SVMs, etc.).
- Different hyperparameters (*e.g.* number of neighbors k, regularization parameter λ, learning rate α, degree d of the polynomial, etc.).
- Different (randomly drawn) training sets S.
- Different representations of the same learning set.

Select the best one or combine them?

Model selection versus ensemble methods

Rather than selecting the best model (w.r.t. some cross-validation procedure), why not try combining the whole set of classifiers and taking advantage of their diversity?

$\rightarrow$ **Ensemble methods**

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.

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.

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.

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:

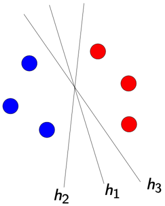
- the individual classifiers (or hypotheses) are accurate, *i.e.* they have an error rate of better than random guessing.
- the classifiers are diverse, *i.e.* they make different errors on new data points.

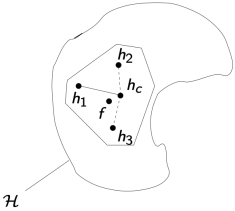
Question

Is it possible to construct (theoretically) good ensembles?

Limitations of a single classifier

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.

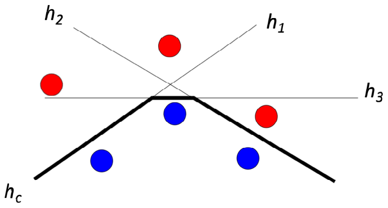


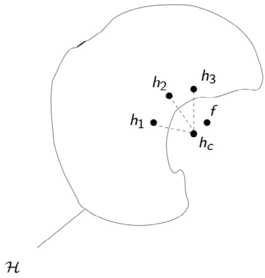


By constructing an ensemble h_c out of all of these accurate classifiers, the algorithm can "average" their votes and reduce the risk of choosing the wrong

Limitations of a single classifier

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

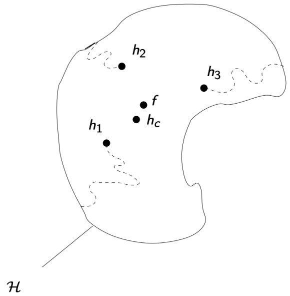




By forming weighted sums of hypotheses drawn from $\mathcal{H}$, it may be possible to expand the space of representable functions.

Limitations of a single classifier

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.

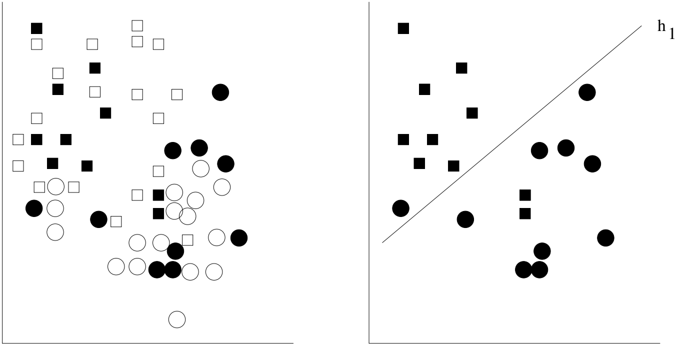
Introduction to boosting

Robert Schapire

Boosting aims at learning a combination of weak classifiers where the update of the distribution is driven by "hard" examples.

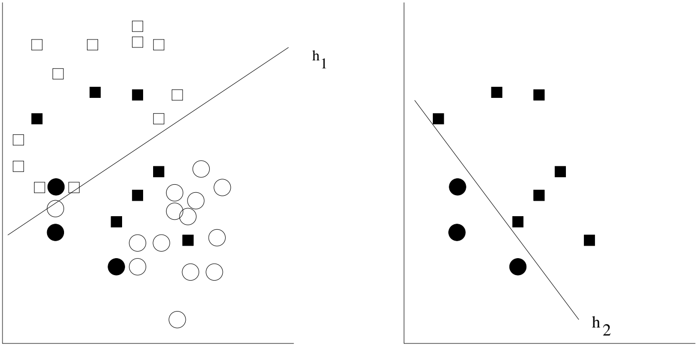
First boosting algorithm (1/4)

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



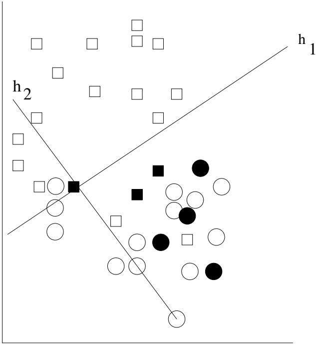
First boosting algorithm (2/4)

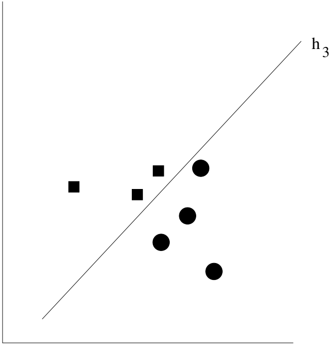
Step 2: Generate a second learning sample S_2, in which an instance has a roughly equal chance of being correctly or incorrectly classified by h_1. L is used again to infer a new hypothesis h_2.



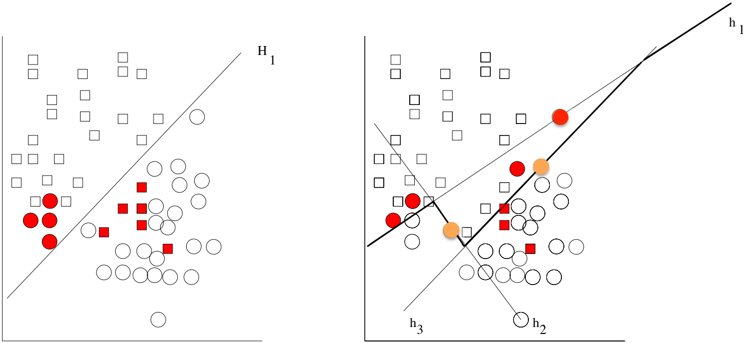
First boosting algorithm (3/4)

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

Input: *A learning sample S, a number of iterations T, a weak learner L*

ADABOOST

Input: *A learning sample* S*, a number of iterations* T*, a weak learner* L
Output: *A global hypothesis* H_T

Adaboost

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**
  $D_1(\mathbf{x_i}) = 1/m$;

ADABOOST

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**
  $h_t = \mathrm{L}(S, \mathbf{D}_t)$;

Adaboost

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**
  $h_t = \mathrm{L}(S, \mathbf{D}_t)$;
  $\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i})$;

ADABOOST

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**

$h_t = \mathrm{L}(S, \mathbf{D}_t)$;
$\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i})$;
$\alpha_t = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t}$;

ADABOOST

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**
    $h_t = \mathrm{L}(S, \mathbf{D}_t)$;
    $\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i})$;
    $\alpha_t = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t}$;
    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**/

ADABOOST

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**
  $h_t = \mathrm{L}(S, \mathbf{D}_t)$;
  $\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i})$;
  $\alpha_t = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t}$;
  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*/

$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$;

ADABOOST

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**
  $h_t = \mathrm{L}(S, \mathbf{D}_t)$;
  $\hat{\epsilon}_t = \sum_{\mathbf{x_i}\ t.q.\ y_i \neq h_t(\mathbf{x_i})} D_t(\mathbf{x_i})$;
  $\alpha_t = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_t}{\hat{\epsilon}_t}$;
  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**/

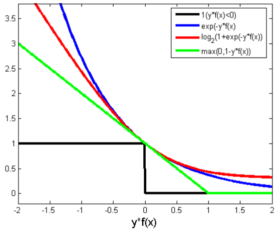
$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$;
Return H_T **such that**
  $H_T(\mathbf{x}) = sign\left(f(\mathbf{x})\right)$

Theoretical results on the empirical risk

Theorem 1

Upper bound on the empirical error of H_T

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



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

Theorem 2

To minimize Z_t, the confidence coefficient α_t must be set to:

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

Exercise

We know that

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

Let us assume that $y(\mathbf{x})$ and $h_t(x) \in \{-1, +1\}$. Let W^b be defined as follows:

$$\forall b \in \{-1, +1\},\ W^b = \sum_{\mathbf{x} \in S : y(\mathbf{x}) h_t(\mathbf{x}) = b} D_t(\mathbf{x})$$

Use W^b to discard the $\sum$ in Z_t and prove Theorem 2.

Theoretical results on the empirical risk

Theorem 3

h_t behaves like random guessing on the new distribution D_{t+1}.

Corollary

A step $t+1$, ADABOOST is forced with h_{t+1} to learn something new about the underlying labelling function which was not captured by h_t.

Theorem 1 (Reminder)

Upper bound on the empirical error of H_T

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

Theorem 4

Exponential decrease of the empirical risk

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

where $\hat{\epsilon}_t = \frac{1}{2} - \gamma_t$ (weak hypothesis). γ_t is the advantage of h_t over random guessing.

- This theorem means that the empirical risk exponentially decreases towards 0 with the number T of iterations.
- if $\forall t : \gamma_t \geq \gamma > 0$ then $\hat{\epsilon}_T \leq e^{-2\gamma^2 T}$

Theoretical results in generalization

Theorem 5

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 6

$\hat{Pr}(margin(\mathbf{x}) \leq \theta)$ exponentially decreases towards 0 with T if error of h_t on $D_t < \frac{1}{2} - \theta$ $(\forall t)$.

Practical advantages of ADABOOST

PROS

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