An Introduction to Boosting
Part I

Gunnar Rätsch

Friedrich Miescher Laboratory
of the Max Planck Society

http://www.fml.tuebingen.mpg.de/~raetsch

Help with earlier presentations: Ron Meir, Klaus-R. Müller

Contents of this Course

- Part I: Introduction to Boosting
  1. Basic Issues, PAC Learning
  2. The Idea of Boosting
  3. AdaBoost's Behavior on the Sample
- Part II: Some Theory
  4. Generalization Error Bounds
  5. Boosting and Large Margins
  6. Leveraging as Greedy Optimization
- Part III: Extensions and Applications
  7. Boosting vs SVMs
  8. Improving Robustness & Extensions
  9. Applications

Slides available from the wiki and from
http://www.fml.mpg.de/~raetsch/boostintro

Sources of Information

Internet http://www.boosting.org*
http://www.cs.princeton.edu/~schapire/boost.html

Conferences Computational Learning Theory (COLT), Neural Information Processing Systems (NIPS), Int. Conference on Machine Learning (ICML), . . .

Journals Machine Learning, Journal of Machine Learning Research, Information and Computation, Annals of Statistics

People List available at http://www.boosting.org

Software Only few implementations: algorithms are "too simple" (cf. http://www.boosting.org)

Review Papers Schapire, 1999, and Meir & Rätsch, 2002.

* New version: http://www.fml.mpg.de/boosting

Learning – Problem Formulation I



The 'World':

Data:	$\{(\mathbf{x}_n, y_n)\}_{n=1}^N, \mathbf{x}_n \in \mathbf{R}^d, y_n \in \{\pm 1\}$
Unknown target function:	$y = f(\mathbf{x})$ (or $y \sim P(y\|\mathbf{x})$)
Unknown distribution:	$\mathbf{x} \sim p(\mathbf{x})$
Objective:	Given new $\mathbf{x}$, predict y

Problem: $P(\mathbf{x}, y)$ is unknown!

Learning – Problem Formulation II



The 'Model'

Hypothesis class: $\mathcal{H} = \{h \mid h : \mathbf{R}^d \to \{\pm 1\}\}$

Loss: $l(y, h(\mathbf{x}))$ (e.g. $\mathbf{I}[y \neq h(\mathbf{x})]$)

Objective: Minimize the true (expected) loss – "generalization error"

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} L(h) \text{ with } \boxed{L(h) := \mathbf{E}_{\mathbb{X} \times \mathbb{Y}}\, l(\mathbb{Y}, h(\mathbb{X}))}$$

Problem: Only have a data sample available, $P(\mathbf{x}, y)$ is unknown!

Solution: Find empirical minimizer

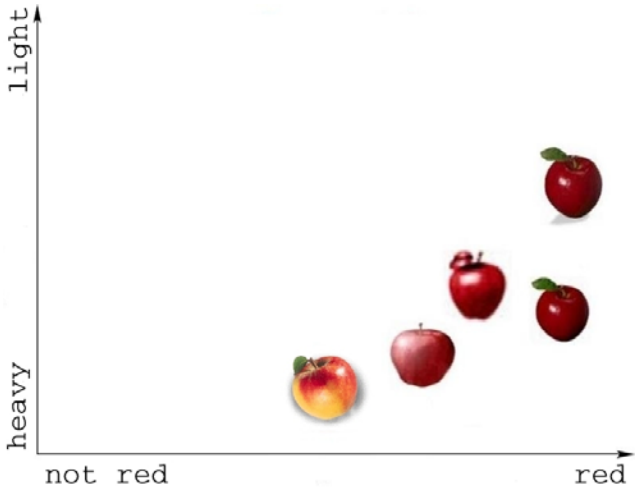
$$\hat{h}_N = \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} l(y_n, h(\mathbf{x}_n))$$

How can we efficiently construct complex hypotheses with small generalization errors?

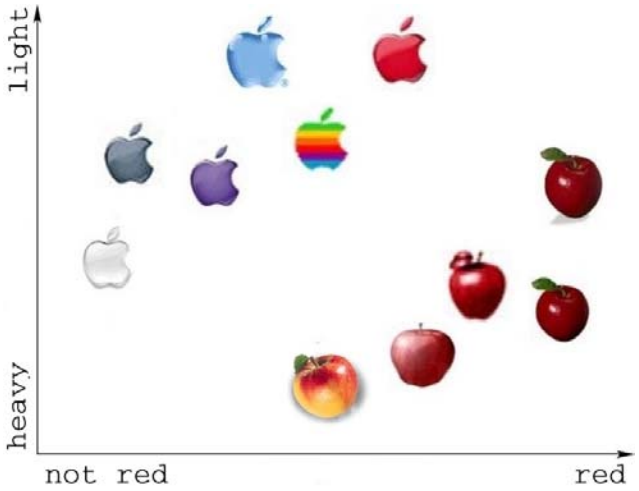
Example: natural vs. plastic Apples



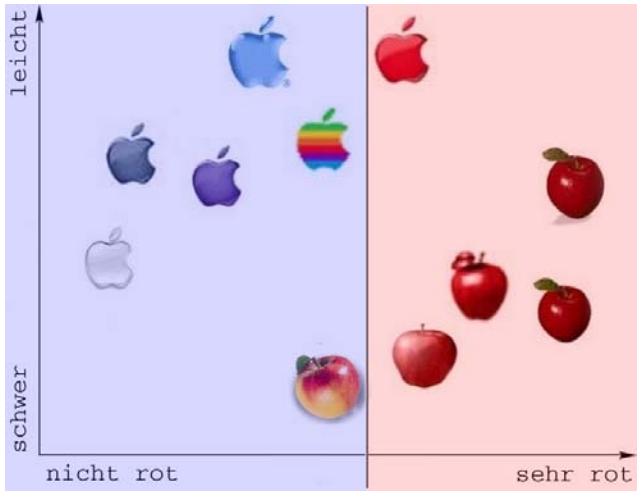
Example: natural vs. plastic Apples



Example: natural vs. plastic Apples



Simple Hypotheses: Cuts on Coordinate Axes



PAC Learning

Input: Sample $S_N = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_n) \sim P(\mathbf{x}, y)$
Hypothesis class $\mathcal{H} \ni h : \mathbf{x} \mapsto y$
Accuracy parameter ϵ
Confidence parameter δ

Algorithm a mapping from S_N to $\mathcal{H}$, $\mathcal{L} : S_N \mapsto \hat{h}_N$

Output function $\hat{h}_N$ from $\mathcal{H}$

Requirements 1. **P**robably **A**pproximately **C**orrect:

$$\mathbf{Pr}\left(S_N : L(\hat{h}_N) - L(h^*) \geq \epsilon\right) \leq \delta$$

where $h^* = \operatorname{argmin}_{h \in \mathcal{H}} \mathbf{E}\, l(\mathbb{Y}, h(\mathbb{X}))$

2. "efficient":
$\Rightarrow$ polynomial runtime in $N, 1/\epsilon, 1/\delta$

Levels of Generality

Recall

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} L(h)$$
$$h^B = \operatorname*{argmin}_{h} L(h) \qquad \text{(h unrestricted)}$$

Restricted Setting: Assume $h^B = h^* \in \mathcal{H}$, require

$$L(\hat{h}_N) \xrightarrow{P} L(h^*)$$

Agnostic Setting: Assume nothing about h^B, require

$$L(\hat{h}_N) \xrightarrow{P} L(h^*)$$

Universal Setting: Assume nothing about h^B, require

$$L(\hat{h}_N) \xrightarrow{P} L(h^B)$$

Weak and Strong Learning

Assumption: Restricted setting, i.e. $h^B \in \mathcal{H}$

Strong PAC Learning: Demand small error with high probability:

$$\mathbf{Pr}\left(S_N : L(\hat{h}_N) - L(h^*) \geq \epsilon\right) \leq \delta \qquad (1)$$

for arbitrary (small) ϵ and δ

Weak PAC Learning: Demand that (1) holds for ϵ and δ

- Example: Binary classification, require that error ϵ is smaller than $1/2$:

$$\epsilon \leq \frac{1}{2}(1 - \gamma) \qquad \text{for some fixed } \gamma > 0$$

With Boosting one can obtain a strong learner from a weak learner!

2. The Idea of Boosting

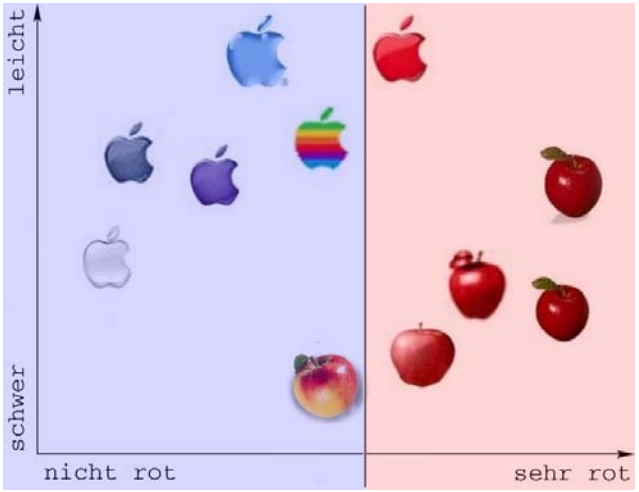


Hill Inlet, Whitsunday Island

Main Sources:

- Freund 1995, Freund & Schapire 1996
- Schapire, Freund, Bartlett & Lee 1998
- Breiman 1996

Simple Hypotheses: Cuts on Coordinate Axes

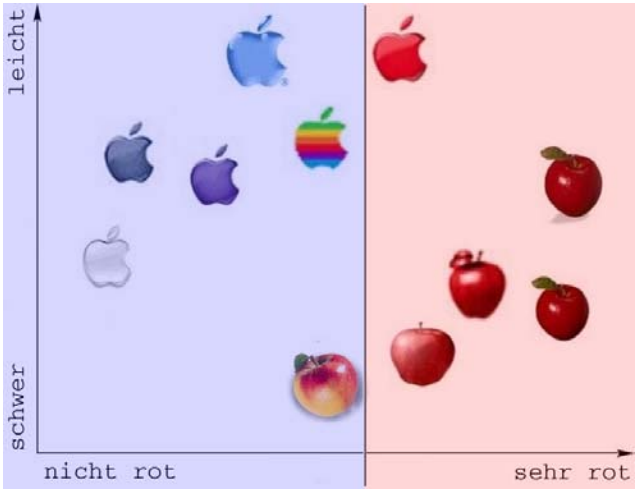


AdaBoost (Freund & Schapire 1996)

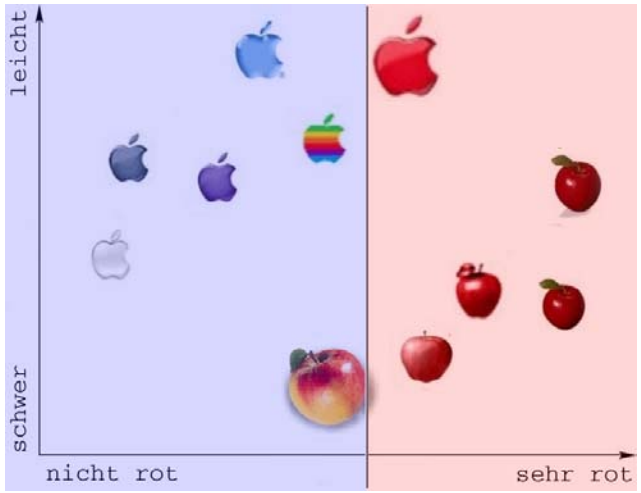
- **Idea:**
  - Simple hypotheses are not perfect!
  - Hypotheses combination $\Rightarrow$ increased accuracy
- **Problems:**
  - How to generate different hypotheses?
  - How to combine them?
- **Method:**
  - Compute distribution $d_1, \ldots, d_N$ on examples
  - Find hypothesis on the weighted training sample $(\mathbf{x}_1, y_1, d_1), \ldots, (\mathbf{x}_N, y_N, d_N)$
  - Combine hypotheses $h_1, h_2, \ldots$ linearly:

$$f = \sum_{t=1}^{T} \alpha_t h_t$$

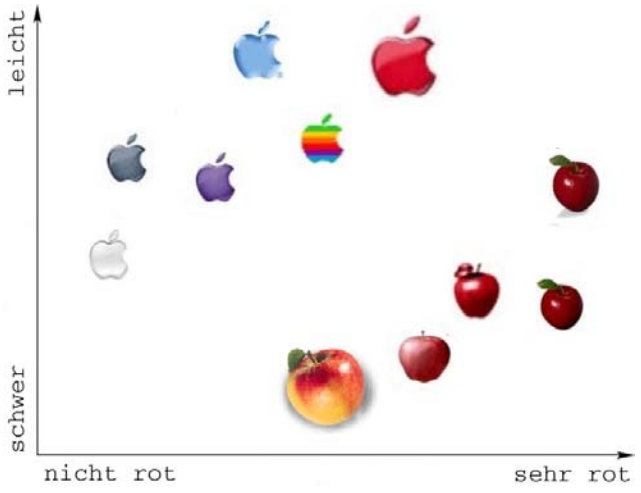
Boosting: 1st Iteration



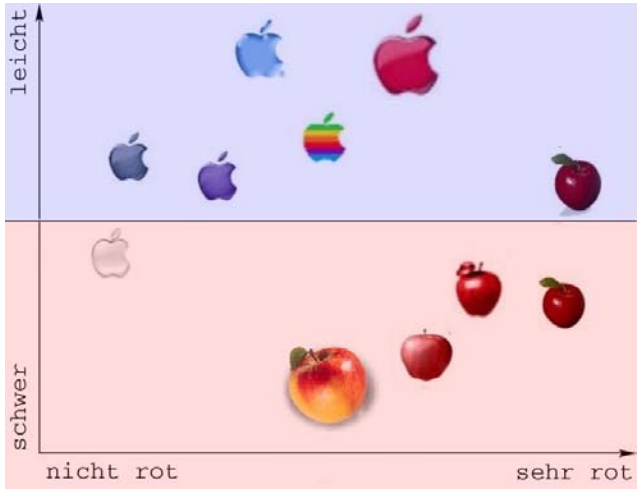
Boosting: Recompute weighting



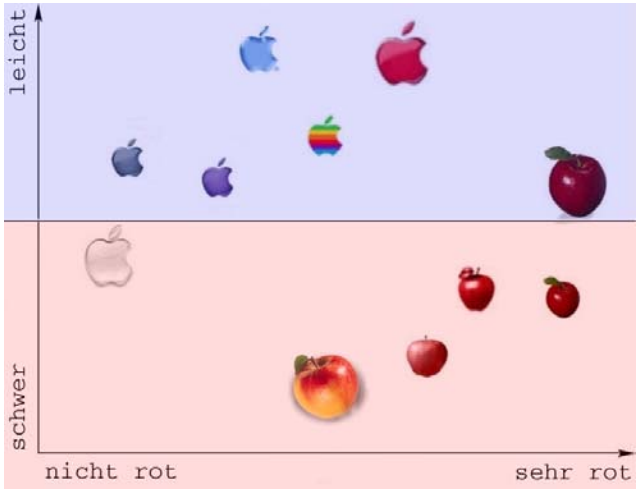
Boosting: 2nd Iteration



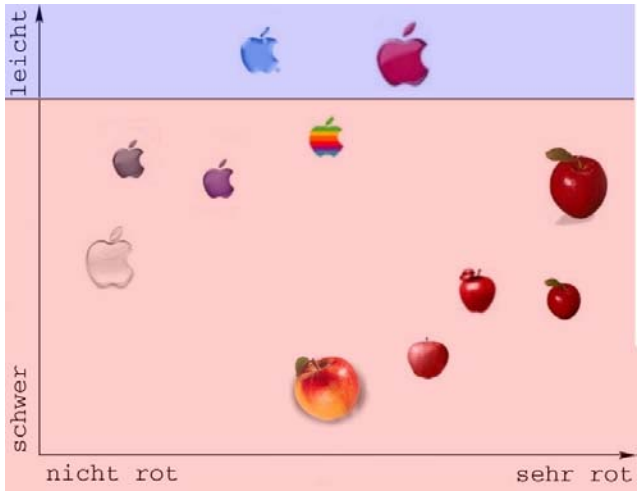
Boosting: 2nd Hypothesis



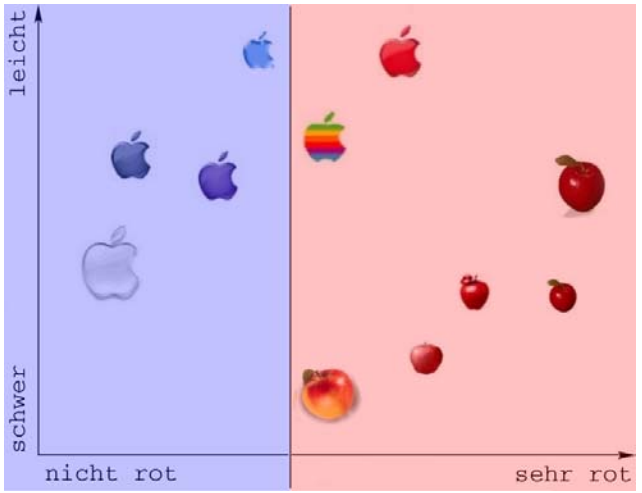
Boosting: Recompute weighting



Boosting: 3nd Hypothesis



Boosting: 4th Hypothesis



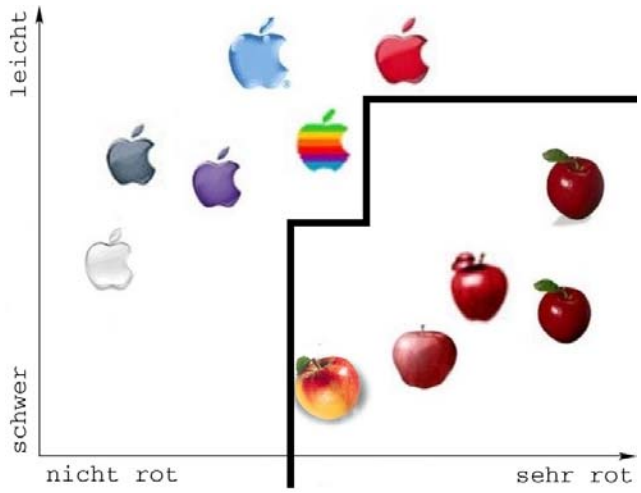
Combination of Hypotheses



Decision



Decision



Ensemble learning for classification

- Ensemble for binary classification consists of
  - Hypotheses (basis functions) $\{h_t(\mathbf{x}) : t = 1, \ldots, T\}$
    - of some hypothesis class $\mathcal{H} = \{h \mid h : \mathbf{x} \mapsto \pm 1\}$
  - Weights $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_T]$
    - satisfying $\alpha_t \geq 0$
- Classification Output: weighted majority of the votes
  - $f_{\text{Ens}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$
- How to find the hypotheses and their weights?
  - $+$ Boosting by Filtering (Schapire 1990)
  - $+$ Boosting by Majority (Freund 1995)
  - $-$ Bagging (Breiman 1996): $\alpha_t = 1/T$
  - $+$ AdaBoost (Freund & Schapire 1996)

Early Algorithms

Boosting by Filtering (Schapire 1990)

- Run weak learner on differently filtered example sets
- Combine the weak hypotheses; **Complex**
- Requires knowledge on the performance of weak learner

Boosting by Majority (Freund 1995)

- Run weak learner on same, but differently weighted set
- Combine the weak hypotheses linearly – Majority vote
- Requires knowledge on the performance of weak learner

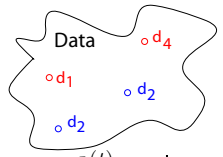
Bagging (Breiman 1996)

- Run learner on bootstrap replicates of the training set
- Average weak hypotheses – Majority vote
- Shown to reduce the variance (in regression)
- **not emphasizing "difficult" examples**

AdaBoost algorithm

Input: N examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

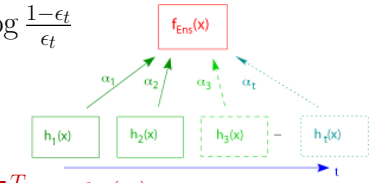
Initialize: $d_n^{(1)} = 1/N$ for all $n = 1 \ldots N$



Do for $t = 1, \ldots, T$,

1. Train base learner according to example distribution $\mathbf{d}^{(t)}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto \{\pm 1\}$.
2. compute weighted error $\epsilon_t = \sum_{n=1}^{N} d_n^{(t)} \mathbf{I}(y_n \neq h_t(\mathbf{x}_n))$
3. compute hypothesis weight $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
4. update example distribution

$$d_n^{(t+1)} = d_n^{(t)} \exp\left(-\alpha_t y_n h_t(\mathbf{x}_n)\right) / Z_t$$



Output: final hypothesis $f_{\text{Ens}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$

Desirable Properties of the Learner



- generates “simple” hypotheses

  $\Rightarrow$ Uses small base hypothesis set $\mathcal{H}$
- Should (approximately) minimize weighted error

$$h_t = \operatorname*{argmin}_{h \in \mathcal{H}} \epsilon_t(h)$$

where

$$\epsilon_t(h) = \sum_{n=1}^{N} d_n^{(t)} \mathbf{I}(y_n \neq h(\mathbf{x}_n))$$

- Often resampling techniques are used
  - Randomly draw subsets according to distribution $\mathbf{d}^{(t)}$
  - Run weak learner on the subset

$\Rightarrow$ Allows usage of arbitrary weak learner

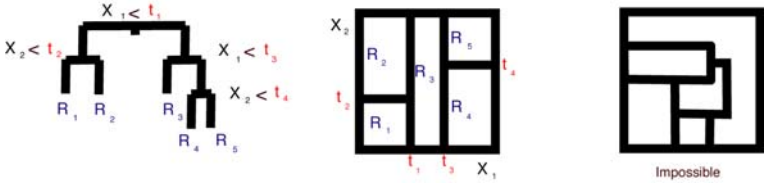
Weak Learners for Boosting



- Decision stumps (axis parallel splits)
- Decision trees (e.g. C4.5 by Quinlan 1996)
- Multi-layer Neural networks (e.g. for OCR)
- Radial basis function networks (e.g. UCI benchmarks, etc)

Decision trees:

- Hierarchical and recursive partitioning on the input space
- Many approaches, usually axis parallel splits



Hypothesis weight $\alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$



. . . is equivalent to minimizing the following loss function in each iteration:

$$E(\alpha) = \sum_{n=1}^{N} d_n^{(t)} \exp(-\alpha y_n h_t(\mathbf{x}_n))$$

Excercise!
Use that $h_t(\mathbf{x}_n) \in \{-1, +1\}$ and $\epsilon_t = \sum_{n=1}^{N} d_n^{(t)} \mathbf{I}(y_n \neq h(\mathbf{x}_n))$

3. AdaBoost’s Behavior on the Sample





Hill Inlet, Whitsunday Island

Main Sources:

- Freund & Schapire 1996
- Schapire, Freund, Bartlett & Lee 1998

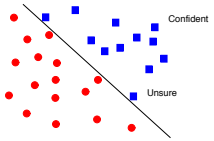
Margins and Classification



Motivation: Lose information in classifying a point as ± 1

Observation: Have higher confidence in correctly classified points, which are far from the decision boundary

Suggestion: Use real number to indicate confidence



Margin: $\rho_f(\mathbf{x}, y) = yf(\mathbf{x})$

Hyperplane: $\rho_{\mathbf{w},b}(\mathbf{x}, y) = y(\mathbf{w}^\top \mathbf{x} + b)$

Which Hyperplane?



- Function set used in boosting: Convex Hull of $\mathcal{H}$

$$\mathcal{F} := \left\{ f : \mathbf{x} \mapsto \sum_{h \in \mathcal{H}} \alpha_h h(\mathbf{x}) \;\middle|\; \alpha_h \geq 0, \sum_{h \in \mathcal{H}} \alpha_h = 1 \right\}$$

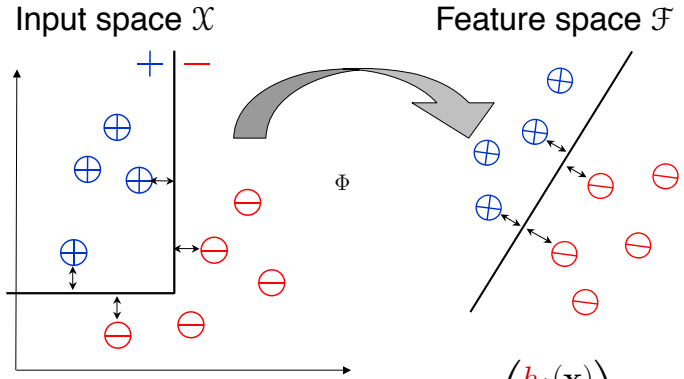
- the α's are the parameters
- Find a hyperplane in the Feature Space spanned by the hypotheses set $\mathcal{H} = \{h_1, h_2, \ldots\}$
- Margin ρ for an example $(\mathbf{x}_i, y_n)$ and $f \in \mathcal{F}$:

$$\rho_n(\boldsymbol{\alpha}) := y_n f(\mathbf{x}_n) = y_i \sum_{t=1}^{T} \frac{\alpha_t}{\sum_t \alpha_t} h_t(\mathbf{x}_i)$$

- Margin ρ for a function $f \in \mathcal{F}$ by $\rho(\boldsymbol{\alpha}) := \min_{n=1,\ldots,N} \rho_n(\boldsymbol{\alpha})$

Large Margin and Linear Separation





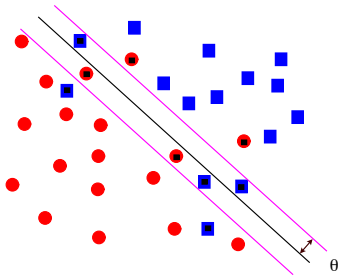
Linear separation in $\mathcal{F}$ is nonlinear separation in input space

$$\Phi(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ \vdots \end{pmatrix},$$

$$\mathcal{H} = \{h_1, h_2, \ldots\}$$

The Empirical Margin Error





Empirical margin error:

$$\hat{L}_N^\theta(f) = \frac{1}{N} \sum_{n=1}^{N} I[y_n f(x_n) \leq \theta]$$

(Scale of θ and f must match!)

The Empirical Margin Error



Lemma: Assume $h_t(x) \in \{-1,+1\}$ and

$$f_T(x) = \frac{\sum_{t=1}^T \alpha_t h_t(x)}{\sum_{t=1}^T \alpha_t} \qquad (f \in [-1,+1]).$$

Then

$$\begin{aligned}\hat{L}_N^\theta(f_T) &= \frac{1}{N}\sum_{n=1}^N I[y_n f_T(x_n) \le \theta] \\ &\le e^{\theta\sum_{t=1}^T \alpha_t}\left(\prod_{t=1}^T Z_t\right),\end{aligned}$$

where Z_t is computed in AdaBoost:

$$Z_t = \sum_{n=1}^N d_n^t e^{-y_n \alpha_t h_t(\mathbf{x}_n)}$$

Proof



$$\begin{aligned}Z_t &= \sum_{n=1}^N d_n^t e^{-y_n \alpha_t h_t(x_n)} \\ &= \sum_{i:y_n = h_t(x_n)} d_n^t e^{-\alpha_t} + \sum_{i:y_n \neq h_t(x_n)} d_n^t e^{\alpha_t} \\ &= (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t}\end{aligned}$$

$$\begin{aligned} y f_T(x) \le \theta \quad &\Rightarrow \quad y\sum_{t=1}^T \alpha_t h_t(x) \le \theta \sum_{t=1}^T \alpha_t \\ &\Rightarrow \quad \exp\left(-y\sum_{t=1}^T \alpha_t h_t(x) + \theta\sum_{t=1}^T \alpha_t\right) \ge 1\end{aligned}$$

Hence, $I[yf(x) \le \theta] \le \exp\left(-y\sum_{t=1}^T \alpha_t h_t(x) + \theta\sum_{t=1}^T \alpha_t\right)$

Proof, Cont'd



$$\begin{aligned} d_n^{t+1} &= d_n^t \frac{\exp\left(-\alpha_t y_n h_t(x_n)\right)}{Z_t} \\ &= \frac{\exp\left(-\sum_{\tau=1}^t \alpha_\tau y_n h_\tau(x_n)\right)}{m\prod_\tau Z_\tau} \qquad \text{(Induction)}\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{E}}\{I[yf(x) \le \theta]\} &\le \hat{\mathbf{E}}\left\{e^{-y\sum_{t=1}^T \alpha_t h_t(x) + \theta\sum_{t=1}^T \alpha_t}\right\} \\ &= \frac{1}{N} e^{\theta\sum_{t=1}^T \alpha_t}\sum_{n=1}^N e^{-y_n\sum_{t=1}^T \alpha_t h_t(x_n)} \\ &= e^{\theta\sum_{t=1}^T \alpha_t}\left(\prod_{t=1}^T Z_t\right)\underbrace{\sum_{n=1}^N d_n^{T+1}}_{=1}\end{aligned}$$

Empirical Margin Error Bound



Claim: Selecting $\alpha_t = (1/2)\ln((1-\epsilon_t)/\epsilon_t)$ leads to the bound

$$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}.$$

Proof: Straightforward by substitution (α_t is minimizer of $Z_t(\alpha_t)$)

Conclusion: Recall $\hat{L}_N^\theta(f) = \frac{1}{N}\sum_{n=1}^N I[y_n f(x_n) \le \theta]$

$$\hat{L}_N^\theta(f_T) \le \prod_{t=1}^T \sqrt{4\epsilon_t^{1-\theta}(1-\epsilon_t)^{1+\theta}}$$

If $\epsilon_t = 1/2 - \gamma_t/2, \quad \gamma_t \ge \theta\ \forall t$
Then $\hat{L}_N^\theta(f_T) \to 0$ exponentially fast!

Training Error



Consider $\theta = 0$: $\hat{L}_N(f) = \frac{1}{N}\sum_{n=1}^{N} I[y_n f(x_n) \leq 0]$

Using $\ln x \leq x - 1$

$$\hat{L}_N(f_T) \leq \prod_{t=1}^{T} \sqrt{(1 - 4\gamma_t^2)}$$

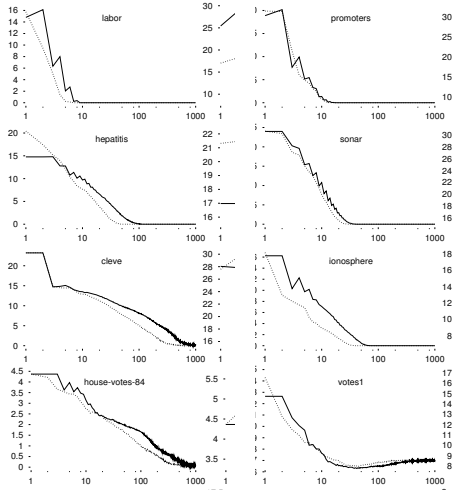
$$\leq e^{-2\sum_{t=1}^{T}\gamma_t^2} \to 0 \quad \text{if} \sum_{t=1}^{T} \gamma_t^2 \to \infty$$

Error can be driven to zero, if weak learner are sufficiently strong

If $\gamma_t \geq \gamma$ $(\forall t)$, then training error is zero after

$$T \leq \left\lceil \frac{2 \log N}{\gamma^2} \right\rceil$$

Training Error for Real Data

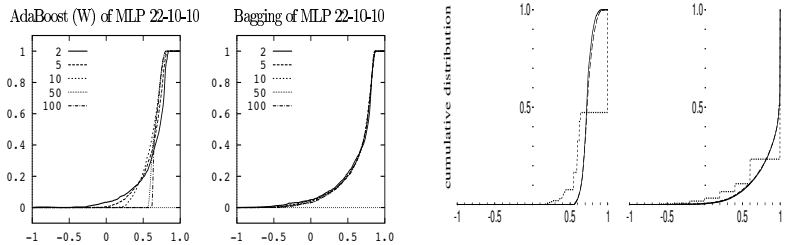


Source: Schapire and Singer 1999

Margin Plots I



Margin plots: Histogram of $y_n f_T(x_n)$, $n = 1, 2, \ldots, N$



Neural Networks

Schwenk and Bengio 2000

Decison Trees

Schapire *et al.* 1998

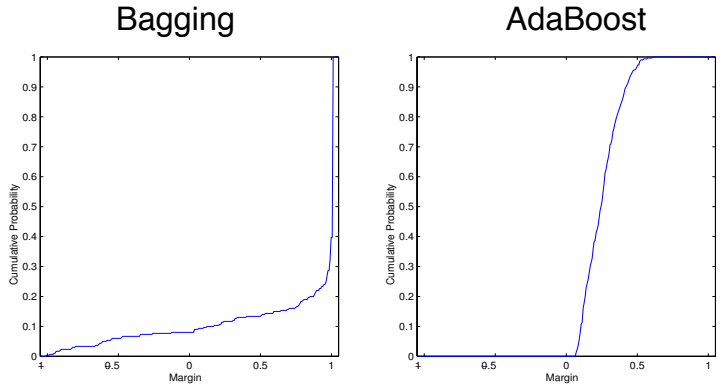
Observation:

- AdaBoost increases the margins of most points
- Bagging has a broader spectrum of distributions

Margin Plots II



AdaBoost tends to increase small margins, while decreasing large margins



Source: Schapire et al. 1998, Rätsch et al., 1998