

An Introduction to Boosting
Part II

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 Performance
  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



4. Generalization Performance



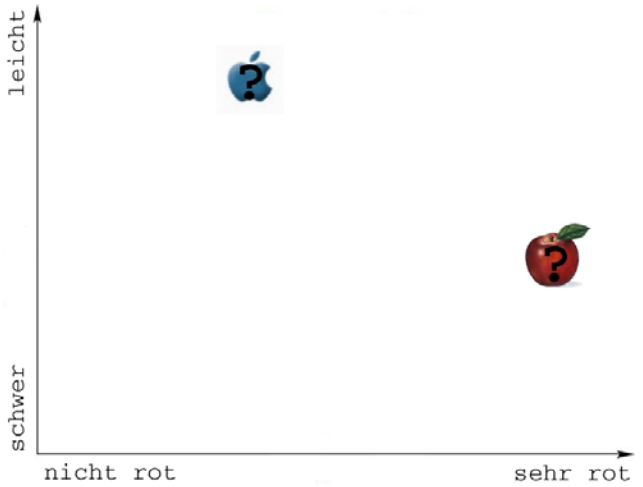
Hill Inlet, Whitsunday Island

Main Sources:

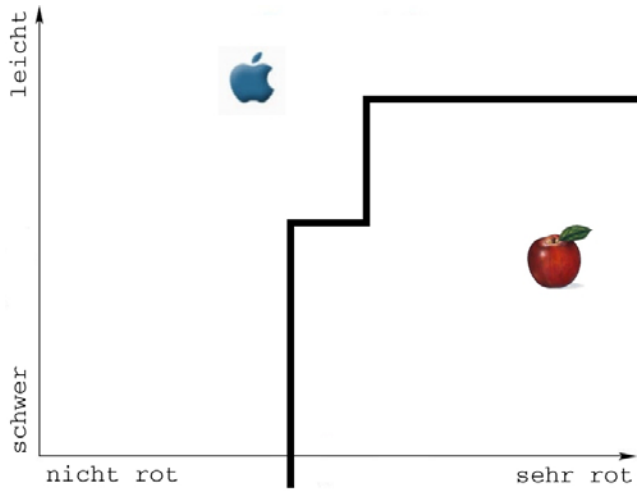
- Freund & Schapire 1996
- Schapire, Freund, Bartlett & Lee 1998



Unseen Examples



Unseen Examples

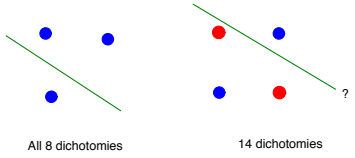


Setup

Expected error:	$L(f) = \mathbf{E}\{I[yf(x) \leq 0]\}$
Data:	$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
Empirical error:	$\hat{L}_N(f) = \frac{1}{N}\sum_{n=1}^{N} I[y_n f(x_n) \leq 0]$
Empirical margin error:	$\hat{L}_N^{\theta}(f) = \frac{1}{N}\sum_{i=1}^{N} I[y_n f(x_n) \leq \theta]$
Empirical classifier:	$\hat{f}_N$
Main issue:	Bound $L(\hat{f}_N)$ in terms of $\hat{L}_N(\hat{f}_N)$
Standard VC bounds:	Use $\hat{L}_N(\hat{f}_N)$
Margin bounds:	Use $\hat{L}_N^{\theta}(\hat{f}_N)$

VC Dimension

Given:	$\mathcal{F} = \{f : \mathbf{R}^d \mapsto \{-1, +1\}\}$
Question:	How complex is the class?
Shattering:	$\mathcal{F}$ shatters a set $X \subset \mathbf{R}^d$, if $\mathcal{F}$ achieves all dichotomies on X
VC-dimension:	The size of the largest shattered subset of X





$d_{\mathcal{F}} = 3$

VC Bounds

For any $f \in \mathcal{F}$, with probability larger than $1 - \delta$

$$|L(f) - \hat{L}_N(f)| \leq c_1\sqrt{\frac{d_{\mathcal{F}}}{N}} + c_2\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

Confidence interval: Standard statistical interpretation

Let

$$\hat{f}_N = \operatorname*{argmin}_{f \in \mathcal{F}} \hat{L}_N(f)$$

Then

$$\mathbf{E}L(\hat{f}_N) \leq \inf_{f \in \mathcal{F}} L(f) + c\sqrt{\frac{d_{\mathcal{F}}}{N}}$$

(Original proof: Vapnik and Chervonenkis 1971)

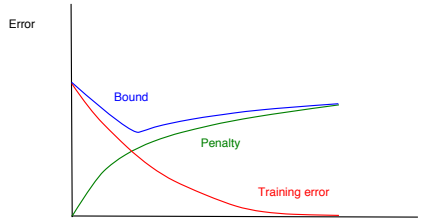
Interpretation of VC Bounds



Recall

$$L(f) \leq \hat{L}_N(f) + c_1\sqrt{\frac{d_{\mathcal{F}}}{N}} + c_2\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

Tightness:	Bound becomes tight for increasing sample size
Empirical Error:	Decreases with complexity of $\mathcal{F}$
Confidence term:	Increases with complexity classes $\mathcal{F}$
Overfitting:	Occurs when $d_{\mathcal{F}}$ becomes very large



VC Bounds for Convex Combination



Recall

$$f_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Let

$$\mathrm{co}_T(\mathcal{H}) = \left\{ f : \; f(x) = \sum_{t=1}^{T} \alpha_t h_t(x), \; \alpha_n \geq 0, \; \sum_{t=1}^{T} \alpha_t = 1 \right\}$$

For any $f \in \mathcal{F} := \mathrm{co}_T(\mathcal{H})$, with probability $1-\delta$

$$L(f) \leq \hat{L}_N(f) + c_1\sqrt{\frac{d(\mathrm{co}_T(\mathcal{H}))}{N}} + c_2\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

Problem with VC Bound: VC-dimension of $\mathrm{co}_T(\mathcal{H})$ may be large, often

$$d(\mathrm{co}_T(\mathcal{H})) \approx T\log(T)\; d(\mathcal{H})$$

But sufficient to get original Boosting results . . .

PAC Boosting – VC dim. of comb. hypothesis

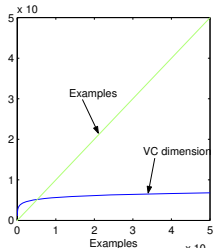


- AdaBoost needs at most $\lceil \frac{2\log N}{\gamma^2} \rceil$ iterations to obtain consistent hypothesis
- Let d be the VC dimension of the hypothesis class $\mathcal{H}$.
- Then the VC dimension of the class of AdaBoost's combined functions is bounded by

$$d_{Ens}(N,\gamma) = \mathcal{O}\Big(\underbrace{d\frac{\log(N)}{\gamma^2}}_{\sim T}\log\left(\frac{\log(N)}{\gamma^2}\right)\Big) = \mathcal{O}\left(d\log(N)\log^2(N)\right).$$

An Example

- VC dimension $d = 2$ (e.g. decision stumps)
- $\epsilon_t \leq 0.4 = \frac{1}{2} - \frac{1}{2}\gamma$ $\Rightarrow \gamma \geq 0.2$



PAC Boosting – putting things together



Assume weak learner is strong enough (weak PAC learner):

- AdaBoost generates consistent hypothesis: $\hat{L}_N(f) = 0$
- VC dimension of a consistent combined hypotheses class

$$d_N \leq \mathcal{O}(d\log(N)\log^2(N))$$

- With probability $1-\delta$ holds

$$L(f) \leq 0 + c_1\sqrt{\frac{d_N}{N}} + c_2\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

- The **generalization** error "$0 + \varepsilon(N)$" can be made arbitrarily small by increasing the sample size N.

PAC Booosting – Digestion

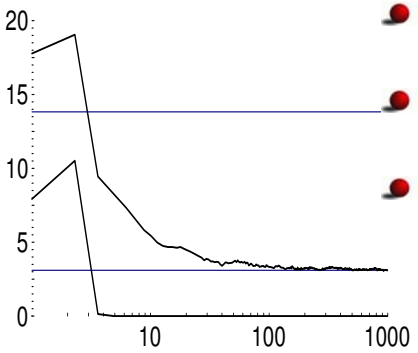


- properties of weak learner imply exponential convergence to a consistent hypothesis
- Fast convergence ensures small VC dimension of the combined hypothesis
- small VC implies small deviation from the empirical risk
- for any $\varepsilon > 0$ and $\delta > 0$ exists a sample size N, such that with probability $1 - \delta$ the expected risk is smaller than ε
- Any weak learner can be boosted to achieve an arbitrary high accuracy! ($\rightsquigarrow$ strong learner)

A "strange" Phenomenon



boosting C4.5 on "letter" data



- test error does not increase
  $\rightsquigarrow$ even after 1000 iterations!
- it continues to drop
  $\rightsquigarrow$ even after training error is 0!
- Occam's razor predicts simpler rule is better
  $\rightsquigarrow$ wrong in this case!?

Needs a better explanation!

Margin Bounds (Schapire, Freund, Bartlett & Lee 1998)



For any $f \in \mathrm{co}_T(\mathcal{H})$ and $\theta \in (0, 1]$, with probability $1 - \delta$

$$L(f) \leq \hat{L}_N^\theta(f) + \frac{c_1}{\theta}\sqrt{\frac{d(\mathcal{H})}{N}} + c_2\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

Observe:

- ⋆ No dependence on T - is overfitting absent?
- ⋆ Luckiness incorporated through $\hat{L}_N^\theta(f)$
- ⋆ Limit $\theta \to 0$ diverges
- ⋆ Much better that VC bounds (if lucky)
- ⋆ Overfitting absent (if lucky) - no T-dependence

Luckiness Tradeoff



$$L(f) \leq \hat{L}_N^\theta(f) + \frac{c_1}{\theta}\sqrt{\frac{d(\mathcal{H})}{N}} + c_2\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

Large θ: $\hat{L}_N^\theta(f)$ small (if lucky)

$\frac{1}{\theta}\sqrt{\frac{d(\mathcal{H})}{N}}$ small

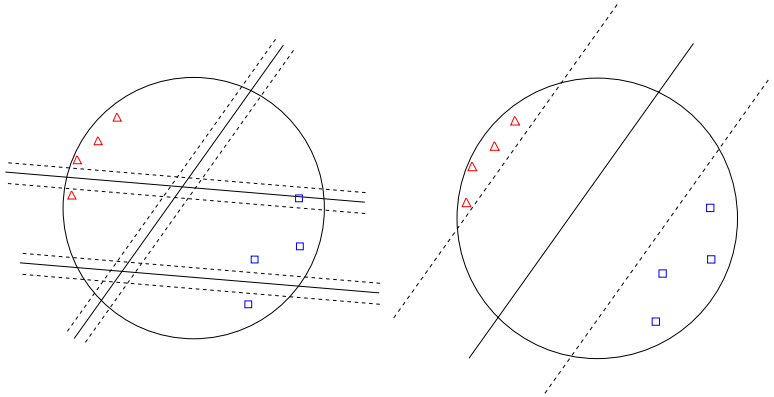
Small θ: $\hat{L}_N^\theta(f) \approx \hat{L}_N(f)$

$\frac{1}{\theta}\sqrt{\frac{d(\mathcal{H})}{N}}$ large

Optimum: If large θ can be achieved with small error
Extreme: AdaBoost – maximize θ while $\hat{L}_N^\theta(f) = 0$

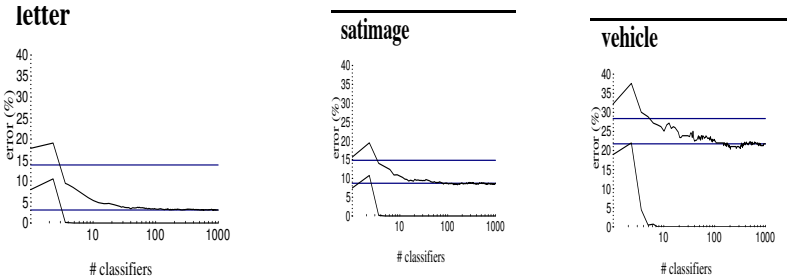
(cf. next section)

Large Margin ⇒ low Complexity? – Why?



Experiments - Is the Theory Corroborated?

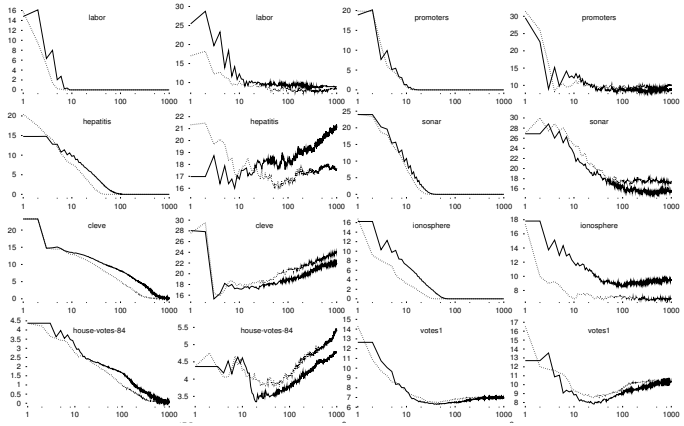
Weak learner: C4.5 decision tree



Training Error: Converges to zero
Test Error: Asymptotes - no overfitting observed
Continues to decrease after training error vanishes

Source: Schapire, Freund, Bartlett and Lee 1998

Overfitting Observed



Source: Schapire and Singer 1999

Regularizing Boosting

Observe: If weak learner attains large advantage, can 'boost forever'

Question: What happens in noisy situations?

Regularization: Need to control complexity
"Don't insist on driving training error to 0"
"Early stopping"
"Do not allow weights on examples to become very different"
"Restrict the optimization process"

New algorithms:

- Maximize the margin subject to constraint on the number of errors
- Trade-off determined by a regularization parameter

5. Boosting and Large Margins

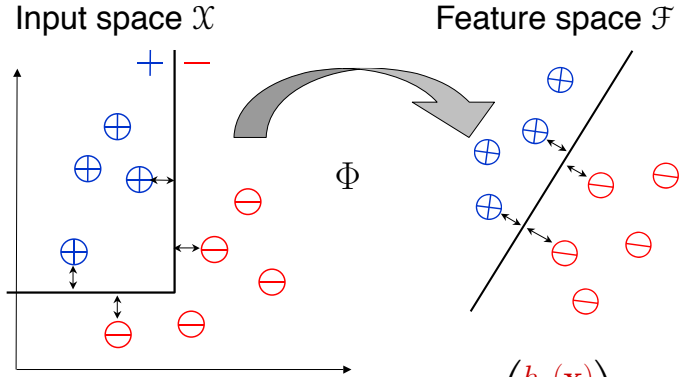


Hill Inlet, Whitsunday Island

Main Sources:

- Schapire, Freund, Bartlett & Lee 1998
- Breiman 1999
- Bennett, Demiriz & Shawe Taylor 2000
- Rätsch 2001, Rätsch & Warmuth 2002

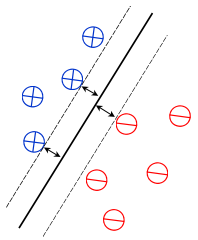
Large Margin and Linear Separation



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

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

Boosting and Large Margins



- Hyperplane = combined hypothesis

$$f_{\boldsymbol{\alpha}}(\mathbf{x}) = (\Phi(\mathbf{x}) \cdot \boldsymbol{\alpha}) = \sum_{j=1}^{J} \alpha_j h_j(\mathbf{x})$$

- margin $\rho(\boldsymbol{\alpha}) = \min_n \underbrace{\frac{y_n(\Phi(\mathbf{x}_n) \cdot \boldsymbol{\alpha})}{\|\boldsymbol{\alpha}\|_1}}_{\text{distance}}$
- distance = ℓ_∞-distance (Mangasarian 1999)

Weak Learner $\mathcal{L}$

- One uses an example weighting $d_1, \ldots, d_N$
- Weak learner $\mathcal{L}$ returns hypotheses $\mathcal{H} \ni h : \mathbf{x} \mapsto \{\pm 1\}$
  ... which are consistently *better than random guessing*

$$\underbrace{\sum_{n=1}^{N} d_n \ \mathrm{I}(h(\mathbf{x}_n) \neq y_n)}_{\text{weighted tr. error of } h} \leq \frac{1}{2} - \frac{1}{2}\gamma$$

  for any weighting $\mathbf{d}$ and some fixed $\gamma > 0$
- For real valued hypotheses $h : \mathbf{x} \mapsto [-1, 1]$

$$\underbrace{\sum_{n=1}^{N} d_n \ y_n h(\mathbf{x}_n)}_{\text{Edge of } h} \geq \gamma$$

Edge and Margin



- Edge of a weak learner $\mathcal{L}$ for a probability vector $\mathbf{d}$ on the examples

$$\gamma(\mathbf{d}) := \max_{h \in \mathcal{H}} \sum_{n=1}^{N} d_n \, y_n h(x_n)$$

- Margin of the combined hypothesis $f_{\boldsymbol{\alpha}}$

$$\rho(\boldsymbol{\alpha}) := \min_{n=1\ldots N} y_n \frac{f_{\boldsymbol{\alpha}}(\mathbf{x}_n)}{\sum_t \alpha_t} = \min_{n=1\ldots N} y_n \sum_{t=1}^{T} \frac{\alpha_t}{\sum_r \alpha_r} h_t(\mathbf{x}_n)$$

Min-Max-Theorem:

$$\min_{\mathbf{d}} \gamma(\mathbf{d}) = \max_{\boldsymbol{\alpha}} \rho(\boldsymbol{\alpha})$$

If $\gamma(\mathbf{d}) \geq \gamma$, the margin is at least γ!

Discussion



What is the margin achieved by AdaBoost?

$\Rightarrow$ finds a solution with large, but not maximal margin

$\Rightarrow$ only about a half of the maximal margin

Modify AdaBoost for maximizing margin

- Arc-GV (Breiman 1999) asymptotically maximizes the margin
  - no converge rates
- AdaBoost* and TotalBoost requires $\frac{2\log(N)}{\epsilon^2}$ iterations to get margin in $[\rho^* - \epsilon, \rho^*]$
- Column Generation uses a Linear Programming solver
  - usually very fast
  - no converge rates

AdaBoost$_\varrho$ (Breiman 1997)



1. **Input:** $S = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, T, margin target ϱ
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots, T$,
   (a) Train classifier on $\{S, \mathbf{d}^t\}$ and obtain $h_t : \mathbf{x} \mapsto [-1, 1]$
   (b) Calculate the edge γ_t of h_t: $\gamma_t = \sum_{n=1}^{N} d_n^t y_n h_t(\mathbf{x}_n)$.
   (c) **if** $\gamma_t \leq \varrho$ or $\gamma_t = 1$, **then** break
   (d) Set $\alpha_t = \frac{1}{2} \log \frac{1+\gamma_t}{1-\gamma_t} \boxed{- \frac{1}{2} \log \frac{1+\varrho}{1-\varrho}}$.
   (e) Update $d_n^{t+1} = d_n^t \exp\left\{-\alpha_t y_n h_t(\mathbf{x}_n)\right\} / Z_t$, s.t. $\sum_{n=1}^{N} d_n^{t+1} = 1$
4. **Output:** $f(\mathbf{x}) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_r \alpha_r} h_t(\mathbf{x})$

Asymptotic Margin for AdaBoost$_\varrho$



Corollary 1 *Assume AdaBoost$_\varrho$ generates hypotheses $h_1, h_2, \ldots$ with edges $\gamma_1, \gamma_2, \ldots$ and coefficients $\alpha_1, \alpha_2, \ldots$. Let $\gamma = \inf_{t=1,2,\ldots} \gamma_t$ and assume $\gamma > \varrho$. Then, asymptotically ($t \to \infty$), the margin ρ of the combined hypothesis is bounded from below by*

$$\rho \geq \frac{\log(1-\varrho^2) - \log(1-\gamma^2)}{\log\left(\frac{1+\gamma}{1-\gamma}\right) - \log\left(\frac{1+\varrho}{1-\varrho}\right)} \geq \frac{\varrho + \gamma}{2}.$$

Arc-GV: Set $\varrho := \max_{r=1,\ldots,t-1} \rho_t$

$\Rightarrow$ easy convergence proof

$\Rightarrow$ no convergence rate!

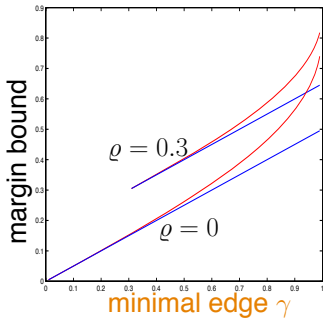
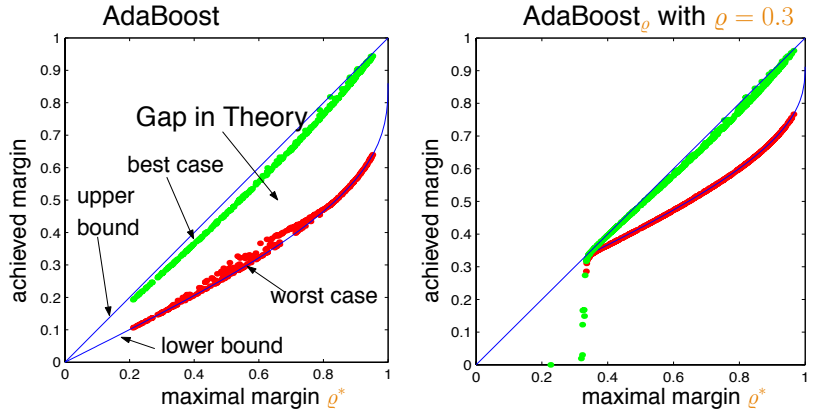


Illustration on Random Problems



$\Rightarrow$ There is a big GAP!

$\Rightarrow$ AdaBoost might not converge

Some Results to get to AdaBoost*

Lemma Assume that in a run of AdaBoost$_\varrho$ holds $\alpha_t \geq 0$ for all t. Then we have

$$\frac{1}{N}\sum_{n=1}^{N} \mathrm{I}\left(\mathbf{y}_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n) \leq \theta\right) \leq \prod_{t=1}^{T} \exp\left\{\theta\alpha_t + \log Z_t\right\}$$

where Z_t is as in AdaBoost$_\varrho$.

Lemma Let γ_t be the edge of h_t in the t-th step of AdaBoost$_\varrho$. Assume $-1 \leq \varrho_t \leq \gamma_t$. Then holds for all $\theta \in [-1, 1]$

$$\exp\left\{\theta\alpha_t + \log Z_t\right\} \leq \exp\left(-\frac{1+\theta}{2}\log\left(\frac{1+\varrho_t}{1+\gamma_t}\right) - \frac{1-\theta}{2}\log\left(\frac{1-\varrho_t}{1-\gamma_t}\right)\right)$$

A Margin-Maximizing Algorithm: AdaBoost*

Lemma If $\theta \leq \varrho_t \leq \gamma_t$:

$$\exp\left(-\frac{1+\theta}{2}\log\left(\frac{1+\varrho_t}{1+\gamma_t}\right) - \frac{1-\theta}{2}\log\left(\frac{1-\varrho_t}{1-\gamma_t}\right)\right)$$

$$\leq \exp\left(-\frac{1+\varrho_t}{2}\log\left(\frac{1+\varrho_t}{1+\gamma_t}\right) - \frac{1-\varrho_t}{2}\log\left(\frac{1-\varrho_t}{1-\gamma_t}\right)\right) \leq 1 - \frac{1}{2}\frac{(\varrho_t-\gamma_t)^2}{1-{\varrho_t}^2}$$

- Assume: $\gamma_t \geq \rho^*$ for all $t = 1, \ldots, T$
- Would like to achieve $\theta = \varrho^* - \epsilon \quad \Rightarrow \quad \theta \leq \min_t \gamma_t - \epsilon$
- Set $\varrho_t = \min_t \gamma_t - \epsilon \quad \Rightarrow \quad \theta \leq \varrho_t \leq \gamma_t \quad \Rightarrow$ apply Lemma

$$\frac{1}{N}\sum_{n=1}^{N} \mathrm{I}\left(\mathbf{y}_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n) \leq \theta\right) \leq \prod_{t=1}^{T}\left(1 - \frac{1}{2}\frac{(\varrho_t-\gamma_t)^2}{1-{\varrho_t}^2}\right) \leq \left(1 - \frac{\epsilon^2}{2}\right)^T$$

$\Rightarrow$ needs $\dfrac{2\log(N)}{\epsilon^2}$ iterations to achieve margin $\rho^* - \epsilon$

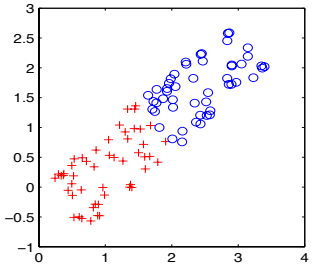
AdaBoost*

1. **Input:** $S = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\rangle$, desired accuracy ν
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   (a) Train classifier on $\{S, \mathbf{d}^t\}$ and obtain $h_t : \mathbf{x} \mapsto [-1, 1]$
   (b) Calculate the edge γ_t of h_t: $\gamma_t = \sum_i d_i^t y_i h_t(\mathbf{x}_i)$
   (c) Set $\varrho_t = \left(\min_{q=1,\ldots,t} \gamma_q\right) - \nu$
   (d) Set $\alpha_t = \frac{1}{2}\log\frac{1+\gamma_t}{1-\gamma_t}\ \boxed{-\frac{1}{2}\log\frac{1+\varrho_t}{1-\varrho_t}}$.
   (e) Update $d_n^{t+1} = d_n^t \exp\left\{-\alpha_t y_n h_t(\mathbf{x}_n)\right\}/Z_t$, s.t. $\sum_{n=1}^{N} d_n^{t+1} = 1$
4. **Output:** $f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

Does a Larger Margin Help?



- 100 dimensional data with two discriminative directions (other noise)
- C4.5, AdaBoost and Marginal AdaBoost on 200 realizations of data



	C4.5	AB	Marginal AB
E_{gen}	$7.4 \pm 0.11\%$	$4.0 \pm 0.11\%$	$\mathbf{3.6 \pm 0.10\%}$
ϱ	–	0.31 ± 0.01	$\mathbf{0.58 \pm 0.01}$
wins	$1/200$	$59/200$	$\mathbf{140/200}$

- generalization error improves with larger margins
- this is only good for separable data $\Rightarrow$ **Soft-Margins**

(cf. Section 8)

Efficient Solution of Semi-infinite Problems



- Marginal AdaBoost and AdaBoost* can solve:

$$\max_{\boldsymbol{\alpha} \in \mathbf{R}^J} \rho \quad \text{with} \quad y_n \sum_{j=1}^{J} \alpha_j h_j(\mathbf{x}_n) \geq \rho \quad \text{for all} \quad n = 1, \ldots, N$$

with $\|\boldsymbol{\alpha}\|_1 = 1$

- Dual optimization Problem:

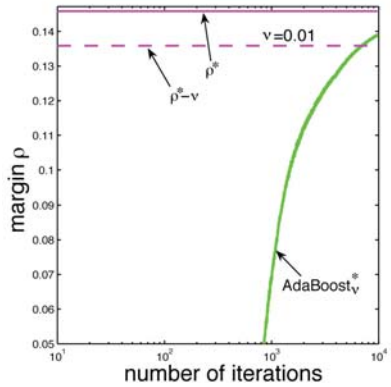
$$\min_{\varepsilon, \mathbf{d} \in \mathbf{R}^N} \varepsilon \quad \text{with} \quad \sum_{n=1}^{N} d_n \, y_n h_j(\mathbf{x}_n) \leq \varepsilon \quad \text{for all} \quad h_j \in \mathcal{H}$$

and $\mathbf{d} \geq \mathbf{0}$, $\sum_{n=1}^{N} d_n = 1$

- if $\mathcal{H}$ has infinite number of elements, there are infinitely many variables (primal) and constraints (dual)
- AdaBoost* works anyway and can solve the problem quite efficiently!

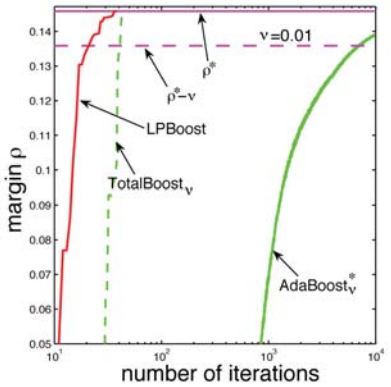
($\mathcal{H}$ needs to be compact)

Discussion: An Example



- AdaBoost* is as fast as expected . . .

Discussion: An Example



- AdaBoost* is as fast as expected . . .
  . . . but much slower than other techniques
  . . . Many calls to the base learning algorithm
- Difference: only optimizes one weight

AdaBoost* as Entropy Projection

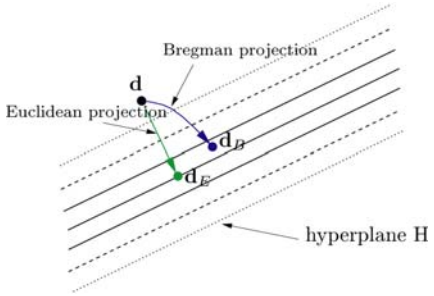
In each iteration of AdaBoost:

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

It turns out that is the solution to

$$\min_{\mathbf{d}\in\mathcal{P}_+^N} \underbrace{\sum_{n=1}^N d_n \log(d_n^t) - d_n + d_n^t}_{\text{KL divergence}}$$

with $\underbrace{\sum_{n=1}^N d_n y_n h_t(\mathbf{x}_n) \le \varrho_t}_{\text{edge of } h_t}$



$\mathbf{d}^{t+1}$ is generalized projection of $\mathbf{d}^t$ onto hyperplane given by h_t

TotalBoost$_\nu$

1. **Input:** $S = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\rangle$, desired accuracy ν
2. **Initialize:** $d_n^1 = 1/N$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots$
   (a) Train classifier on $\{S, \mathbf{d}^t\}$ and obtain $h_t : \mathbf{x} \mapsto [-1, 1]$
   (b) Calculate edge γ_t of h_t: $\gamma_t = \sum_i d_i^t y_i h_t(\mathbf{x}_i)$
   (c) Set $\varrho = (\min_{q=1,\ldots,t} \gamma_q) - \nu$
   (d) Update $\mathbf{d}^{t+1} = \operatorname{argmin}_{\{\mathbf{d}\in\mathcal{P}^N \mid \sum_i d_i y_i h_q(\mathbf{x}_i) \le \varrho, \text{ for } 1\le q\le t\} = \mathcal{C}_t} \Delta(\mathbf{d}, \mathbf{d}^1)$
   (e) **If** above infeasible or at boundary, then break
4. **Output:** $f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$, where the coefficients α_t maximize margin over hypotheses set $\{h_1, \ldots, h_T\}$.

Iteration Bound $\lceil \frac{2\ln N}{\nu^2} \rceil$

Theorem 2 *For $\mathbf{d}^t, \mathbf{d}^{t+1} \in \hat{P}^N$ and $\mathbf{u} \in [-1, 1]^N$, if $\Delta(\mathbf{d}^{t+1}, \mathbf{d}^t)$ finite and $\mathbf{d}^{t+1} \cdot \mathbf{u} \neq \mathbf{d}^t \cdot \mathbf{u}$ then*

$$\Delta(\mathbf{d}^{t+1}, \mathbf{d}^t) > \frac{(\mathbf{d}^{t+1} \cdot \mathbf{u} - \mathbf{d}^t \cdot \mathbf{u})^2}{2}$$

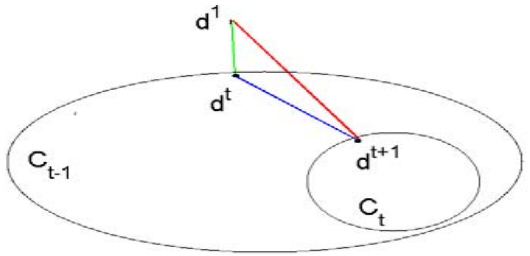
- $\mathbf{d}^t \cdot \mathbf{u}^t = \gamma_t$
- $\mathbf{d}^{t+1} \cdot \mathbf{u}^t \le \varrho \le \gamma_t - \nu$, where $u_i^t := y_i h_t(\mathbf{x}_i)$
- Thus $\mathbf{d}^t \cdot \mathbf{u}^t - \mathbf{d}^{t+1} \cdot \mathbf{u}^t \ge \nu$ and $\Delta(\mathbf{d}^{t+1}, \mathbf{d}^t) > \frac{\nu^2}{2}$

Generalized Pythagorean Theorem

$\mathcal{C}_t = \{\mathbf{d} \in \hat{P}^N | \mathbf{d} \cdot \mathbf{u}^q \le \hat{\gamma}_t, 1 \le q \le t\}, \mathcal{C}_0 = \hat{P}^N, \mathcal{C}_t \subseteq \mathcal{C}_{t-1}$
$\mathbf{d}^t$ is projection of $\mathbf{d}^1$ onto $\mathcal{C}_{t-1}$ at iteration $t-1$

$$\mathbf{d}^t = \operatorname*{argmin}_{\mathbf{d}\in\mathcal{C}_{t-1}} \Delta(\mathbf{d}, \mathbf{d}^1)$$

$$\Delta(\mathbf{d}^{t+1}, \mathbf{d}^1) \ge \Delta(\mathbf{d}^t, \mathbf{d}^1) + \Delta(\mathbf{d}^{t+1}, \mathbf{d}^t)$$



Sketch of Proof

$$
\begin{array}{rccccc}
1: & \Delta(\mathbf{d}^2,\mathbf{d}^1) - \Delta(\mathbf{d}^1,\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^2,\mathbf{d}^1) & > & \frac{\nu^2}{2} \\
2: & \Delta(\mathbf{d}^3,\mathbf{d}^2) - \Delta(\mathbf{d}^2,\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^3,\mathbf{d}^2) & > & \frac{\nu^2}{2} \\
 & \dots & & \dots & & \\
t: & \Delta(\mathbf{d}^{t+1},\mathbf{d}^1) - \Delta(\mathbf{d}^t,\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^{t+1},\mathbf{d}^t) & > & \frac{\nu^2}{2} \\
 & \dots & & \dots & & \\
T-1: & \Delta(\mathbf{d}^T,\mathbf{d}^1) - \Delta(\mathbf{d}^{T-1},\mathbf{d}^1) & \geq & \Delta(\mathbf{d}^T,\mathbf{d}^{T-1}) & > & \frac{\nu^2}{2}
\end{array}
$$

$$
+: \quad \underbrace{\Delta(\mathbf{d}^T,\mathbf{d}^1)}_{\leq \ln N} - \underbrace{\Delta(\mathbf{d}^1,\mathbf{d}^1)}_{=0} \quad > \quad (T-1)\frac{\nu^2}{2}
$$

Therefore, $T \leq \lceil \frac{2 \ln N}{\nu^2} \rceil$

LPBoost: Column Generation

Recall: Dual optimization Problem:

$$
D(\mathcal{H}): \quad \min_{\varepsilon, \mathbf{d} \in \mathcal{P}^N} \varepsilon \quad \text{with} \quad \sum_{n=1}^{N} d_n \, y_n h_j(\mathbf{x}_n) \leq \varepsilon \quad \text{for all} \quad h_j \in \mathcal{H}
$$

Add one hypothesis per iteration and resolve the restricted master problem:

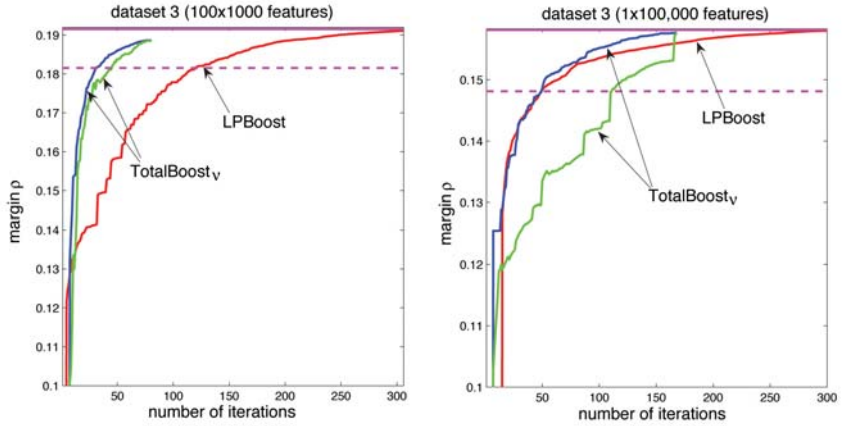
- Solve $D(\mathcal{H}_t)$ starting with $\mathcal{H}_1 = \emptyset$
- Find $h \in \mathcal{H}$ violating constraint in $D(\mathcal{H})$ at solution $D(\mathcal{H}_t)$
  - that is one with large enough edge
- Add hypothesis $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{h_t\}$

$\Rightarrow$ Converges in finite steps, when $\mathcal{H}$ finite
$\Rightarrow$ Converges asymptotically, when $\mathcal{H}$ infinite (how fast?)

We have also solved the primal problem if $\mathcal{H}$ is compact!

LPBoost can perform much worse

. . . when using many redundant features:



LPBoost vs. TotalBoost$_\nu$ on two 100,000 dimensional datasets: [*left*] many redundant features ($N_1 = 1,000, N_2 = 100$) and [*right*] independent features ($N_1 = 100,000, N_2 = 1$).

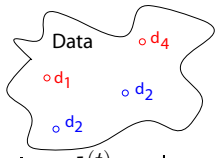
6. Boosting as Greedy Optimization



Main Sources:

- Kivinen & Warmuth 1999, Lafferty 1999
- Collins, Schapire & Singer 2002
- Rätsch, Mika & Warmuth 2001, Rätsch 2001

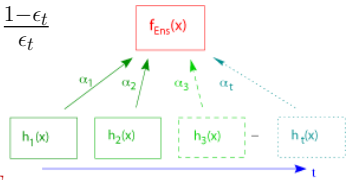
AdaBoost algorithm



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

Loss Function for AdaBoost

- AdaBoost stepwise minimizes a function

$$\mathrm{G}[f_{\boldsymbol{\alpha}}] = \sum_{n=1}^N \exp\left\{-y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)\right\}$$

- The gradient of $\mathrm{G}[f_{\boldsymbol{\alpha}^{(t)}}]$ gives the example weights used for AdaBoost:

$$\frac{\partial\, \mathrm{G}[f_{\boldsymbol{\alpha}^{(t)}}]}{\partial f(\mathbf{x}_n)} = \exp\left\{-y_n f_{\boldsymbol{\alpha}^{(t)}}(\mathbf{x}_n)\right\} =: d_n^{(t+1)}$$

- The hypothesis coefficient α_t is chosen, such that $\mathrm{G}[f_{\boldsymbol{\alpha}^{(t)}}] = \mathrm{G}[f_{\boldsymbol{\alpha}^{(t-1)}} + \alpha_t h_t]$ is minimized:

$$\alpha_t = \operatorname*{argmin}_{\alpha_t \in \mathbf{R}} \mathrm{G}[f_{\boldsymbol{\alpha}^{(t)}}] = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$$

The Generic Leveraging Algorithm

Input: loss $\mathrm{G} : \mathbf{R}^N \to \mathbf{R}$, N examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
Initialize: $f_0 \equiv 0$, $\mathbf{d}^{(1)} = \nabla \mathrm{G}[f_0]$ (often $\sim 1/N$)
Do for $t = 1, \ldots, T$,

1. Run weak learner according to distribution $\mathbf{d}^{(t)}$ and obtain hypothesis h_t.
2. Compute hypothesis weight α_t such that $\mathrm{G}[f_{t-1} + \alpha_t h_t]$ is minimized
3. Update the combined hypothesis: $f_t = f_{t-1} + \alpha_t h_t$
4. Update example distribution: $\mathbf{d}^{t+1} = \nabla \mathrm{G}[f_t]$

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

Leveraging = Boosting without PAC Boosting property

Examples

- AdaBoost (Freund & Schapire, 94) & Confidence Rated Boosting

$$\mathrm{G}[f_{\boldsymbol{\alpha}}] = \sum_{n=1}^N \exp\left\{-y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)\right\} \qquad d_n = \exp(-y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n))$$

- Logistic Regression Boosting (Friedman et al., Collins et al., 2000)

$$\mathrm{G}[f_{\boldsymbol{\alpha}}] = \sum_{n=1}^N \log\left(1 + \exp\left\{-y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)\right\}\right) \qquad d_n = \frac{\exp(-y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n))}{1 + \exp(-y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n))}$$

- Least Square Boosting (Friedman, 1999)

$$\mathrm{G}[f_{\boldsymbol{\alpha}}] = \sum_{n=1}^N (y_n - f_{\boldsymbol{\alpha}}(\mathbf{x}_n))^2 \qquad d_n = y_n - f_{\boldsymbol{\alpha}}(\mathbf{x}_n)$$

- Boosting with Soft Margins (Rätsch et al., 2001a, 2001b)

$$\mathrm{G}[f_{\boldsymbol{\alpha}}] = \sum_{n=1}^N \log\left(1 + \exp\left\{1 - y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)\right\}\right) \qquad d_n = \frac{\exp(1 - y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n))}{1 + \exp(1 - y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n))}$$

Coordinate Descent



- Task: minimize a loss function of the form:

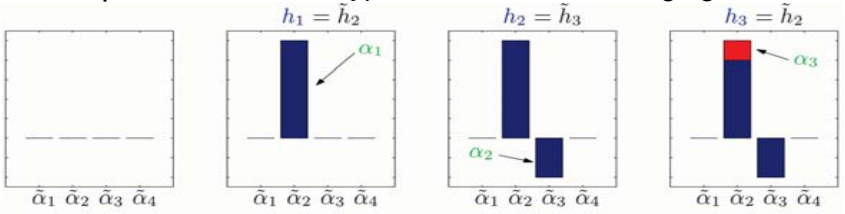
$$\mathrm{F}(\tilde{\boldsymbol{\alpha}}) = \mathrm{F}(\tilde{\alpha}_1, \ldots, \tilde{\alpha}_J)$$

- Particularly simple strategy: Iteratively select a coordinate, say the j-th, and find $\tilde{\alpha}_j$, such that $\mathrm{F}(\tilde{\alpha}_1, \ldots, \tilde{\alpha}_j, \ldots, \tilde{\alpha}_J)$ is minimized.
- How to choose j? Different strategies:
  - At random.
  - Round robin fashion.
  - Gauss-Southwell Scheme (GS): select component with largest, absolute gradient value.
  - Maximum-Improvement Scheme (MI): Choose the coordinate such, that the objective decreases maximally.
- Converges under certain conditions

Leveraging as Coordinate Descent



- Leveraging implements a coordinate gradient descent method in $\tilde{\boldsymbol{\alpha}}$-coefficient space $\mathbf{R}^J$ to minimize $\mathrm{G}[f_{\tilde{\boldsymbol{\alpha}}}]$
- Example: four different hypotheses, three Leveraging iterations



$\tilde{h}_2$ is selected twice and its weight is corrected in the 3rd iteration.

- If the learner selects a hypothesis more than once, we merge the weights by:

$$\tilde{\alpha}_j^{(t)} = \sum_{r=1}^{t} \alpha_r \mathrm{I}(h_r = \tilde{h}_j), \quad j = 1, \ldots, J, \qquad (1)$$

Base Learner and Selection Schemes



- If the base learner minimizes the $\mathbf{d}^{(t)}$-weighted training error ϵ_t, then it
  - maximizes the edge γ_t (assume $\tilde{h}_j(\mathbf{x}) \in \{-1, +1\}$) and
  - finds the hypothesis $h \in \mathcal{H}$ that has the largest gradient component (Gauss-Southwell Scheme), since

$$\frac{\partial \mathrm{G}[f_{\tilde{\boldsymbol{\alpha}}^{(t-1)}}]}{\partial \tilde{\alpha}_j} = \sum_{n=1}^{N} d_n^{(t)} \tilde{h}_j(\mathbf{x}_n) = \gamma_t,$$

- The base learner can be used as an "Oracle" to select new hypotheses/directions for coordinate descent.
- In confidence rated boosting (Schapire & Singer, 98) one uses the base learner to find the hypothesis that minimizes G (MI Scheme).

Preliminaries & Conditions



- The general form we consider is:

$$\min_{\tilde{\boldsymbol{\alpha}} \in \mathcal{S}} \mathrm{F}(\tilde{\boldsymbol{\alpha}}), \quad \text{where} \quad \mathrm{F}(\tilde{\boldsymbol{\alpha}}) = \mathrm{G}(H\tilde{\boldsymbol{\alpha}}) + \boldsymbol{\kappa}^\top \tilde{\boldsymbol{\alpha}}, \qquad (2)$$

  where G twice continously differenctiable and strictly convex on $\mathrm{dom}\ \mathrm{G}$, H has no zero columns, and $\mathcal{S}$ is a (not necessarily bounded) box constrained set.
- When/how will coordinate descent for such an loss function (think: AdaBoost) converge? **Only depends on base learner!**
- AdaBoost, Logistic Regression, LS-Boost all have such loss functions where $\boldsymbol{\kappa}$ is zero (hence: forget $\boldsymbol{\kappa}$ for now).
- Here we will only consider finite hypothesis sets (closed under complementation).

Condition: δ-Optimality



Let $\mathcal{H}$ be a finite, complementation closed hypothesis set and G be a functional as e.g. in AdaBoost.

A weak learner is called δ-optimal, if it always returns a hypothesis $\tilde{h}_j \in \mathcal{H}$ that satisfies

1. $\gamma_j \geq \delta \max_{j'=1,\ldots,J} \gamma_{j'}$, where $\gamma_j = \sum_{n=1}^{N} d_n \tilde{h}_j(\mathbf{x}_n)$,

or

2. $\mathrm{G}(\tilde{\boldsymbol{\alpha}}^t) - \mathrm{G}(\tilde{\boldsymbol{\alpha}}^{t+1,j}) \geq \delta \max_{j'=1,\ldots,J} \left(\mathrm{G}(\tilde{\boldsymbol{\alpha}}^t) - \mathrm{G}(\tilde{\boldsymbol{\alpha}}^{t+1,j'}) \right)$

for some fixed $\delta \in (0, 1]$ and all $\mathbf{d} = \nabla \mathrm{G}(\boldsymbol{\alpha})$.

$\Rightarrow$ Approximative Gauss-Southwell vs. Maximum Improvement Scheme

Convergence Theorem for Leveraging



Theorem: Given a training sample S, a finite, complementation closed hypothesis set $\mathcal{H}$. Let $\mathrm{G} : \mathbf{R}^N \to \mathbf{R}$ be twice differentiable, strictly convex and its Hessian be uniformly upper bounded. Suppose Leveraging generates a sequence of combined hypotheses $f_{\tilde{\boldsymbol{\alpha}}^0}, f_{\tilde{\boldsymbol{\alpha}}^1}, \ldots$ using a δ-optimal learning algorithm $\mathcal{L}$ for $\mathcal{H}$.
Then any limit point of $\{\tilde{\boldsymbol{\alpha}}^t\}$ is a solution and the sequence converges *linearly*.

Note: Linear convergence means, that the objective function decreases exponentially fast towards its minimum:

$$\mathrm{G}[f_{\tilde{\boldsymbol{\alpha}}^{(t)}}] - \mathrm{G}[f^*] \leq c^{-t} \quad \text{for some } c < 1$$

And what about AdaBoost?



- **Corollary** Suppose AdaBoost, Logistic Regression or LS-Boost generates a sequence of combined hypotheses $f_{\tilde{\boldsymbol{\alpha}}^{(1)}}, f_{\tilde{\boldsymbol{\alpha}}^{(2)}}, \ldots$ using a δ-optimal base learner.
  Assume solutions of the optimization problem are finite.
  Then $\{f_{\tilde{\boldsymbol{\alpha}}^{(t)}}\}$ converges linearly to a solution!
- Already known for AdaBoost and Logistic Regression if the data is separable (PAC boosting property).
  But here it also holds for nonseparable data and a large class of loss functions!

Dual Optimization – AdaBoost



In each iteration of AdaBoost:

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

It turns out that is the solution to

$$\min_{\mathbf{d} \in \mathbf{R}_+^N} \underbrace{\sum_{n=1}^{N} d_n \log(d_n^t) - d_n + d_n^t}_{\text{KL divergence}}$$

$$\text{with} \quad \underbrace{\sum_{n=1}^{N} d_n y_n h_t(\mathbf{x}_n) = 0}_{\text{edge of } h_t}$$

$\mathbf{d}^{t+1}$ is generalized projection of $\mathbf{d}^t$ onto hyperplane given by h_t

Dual Optimization – Leveraging



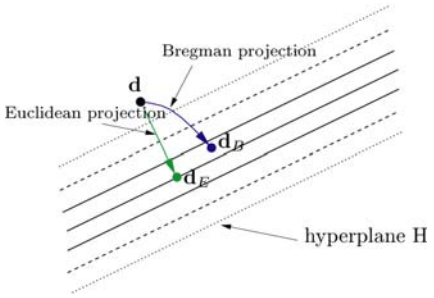
Define generalized "Distance" (Bregman Divergence):

$$\Delta_{\mathcal{G}}(\mathbf{d}, \mathbf{d}^t) := \mathcal{G}(\mathbf{d}) - \mathcal{G}(\mathbf{d}^t) - \langle \nabla \mathcal{G}(\mathbf{d}^t), \mathbf{d} - \mathbf{d}^t \rangle$$

Find the point "closest" to $\mathbf{d}^t$ which is on the hyperplane, i.e. solve

$$\min_{\mathbf{d} \in \mathbf{R}^N} \quad \Delta_{\mathcal{G}}(\mathbf{d}, \mathbf{d}^t)$$

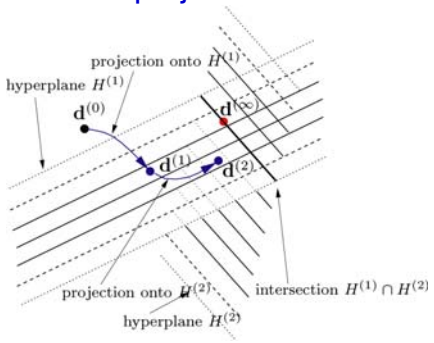
$$\text{with} \quad \sum_{n=1}^{N} d_n y_n h_t(\mathbf{x}_n) = 0$$



Bregman Algorithm and Leveraging



Bregman Algorithm:
– Iterate projections



Leads to Dual solution of:

$$\min_{\mathbf{d} \in \mathbf{R}^N} \quad \Delta_{\mathcal{G}}(\mathbf{d}, \mathbf{d}^0)$$

$$\text{with} \quad \sum_{n=1}^{N} d_n y_n h_j(\mathbf{x}_n) = 0 \qquad \forall h_j \in \mathcal{H}$$

Primal Problem:

$$\min_{\boldsymbol{\alpha}} \quad \mathrm{G}[f_{\boldsymbol{\alpha}}] + \text{const.}$$

G is convex conjugate function to $\mathcal{G}$

Digestion I



Convergence of Leveraging:

- Primal Domain (Coordinate Descent)
- Dual Domain (Generalized Projections)

$\Rightarrow$ obtain similar results as previously for AdaBoost

Many more convergence results!

Most general results:

- Rätsch et al. 2001 (global minimum; finite $\mathcal{H}$; linear convergence)
- Collins et al. 2001 (global minimum; finite $\mathcal{H}$; no rates)
- Zhang 2001 (global minimum; modified alg.; order one convergence)
- Mason et al. 1998 (local convergence; "AnyBoost")

Digestion II



Observe:

- AdaBoost$^{(*)}$ solves Linear Program (Section 5), but ...
- ... also minimizes exponential function (Section 6)!

Many solutions for exponential problem!

Questions:

- **Why is AdaBoost solving the LP?**
- **What is the connection between the exponential approach and the LP?**

$\Rightarrow$ **Barrier Optimization!**

How to solve a Linear Program?



$$\begin{aligned} \max_{\boldsymbol{\alpha},\rho} \quad & \rho \\ \text{subject to} \quad & y_n \sum_{j=1}^{J} \alpha_j h_j(\mathbf{x}_n) \geq \rho \quad n = 1,\ldots,N \\ & \alpha_j \geq 0 \qquad\qquad\qquad j = 1,\ldots,J \\ & \textstyle\sum_j \alpha_j = 1 \end{aligned}$$

- Simplex methods
- Barrier/Penalty methods
- Interior point methods
- ...

The number of hypotheses can be very large or infinite!
How is boosting solving this task?

Barrier Optimization



Constraint Optimization:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & g(\boldsymbol{\theta}) \\ \text{s.t.} \quad & c_n(\boldsymbol{\theta}) \geq 0 \\ & n = 1,\ldots,N \end{aligned}$$

$$E_\beta(\boldsymbol{\theta}) = g(\boldsymbol{\theta}) + \sum_{n=1}^{N} \kappa_\beta(c_n(\boldsymbol{\theta}))$$

Convergence: $\lim_{\beta\to 0} \operatorname{argmin}_{\boldsymbol{\theta}} E_\beta(\boldsymbol{\theta}) = \boldsymbol{\theta}^*$ $\rightsquigarrow$ solution to problem

Barrierfunctions: $\kappa_\beta(t) = -\beta\log(t)$ $\rightsquigarrow$ Log-Barrier $\rightsquigarrow$ Interior Point
$\kappa_\beta(t) = \beta\exp(-t/\beta)$ $\rightsquigarrow$ Exp-Barrier $\rightsquigarrow$ Boosting

Barrier Optimization



Constraint Optimization:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & g(\boldsymbol{\theta}) \\ \text{s.t.} \quad & c_n(\boldsymbol{\theta}) \geq 0 \\ & n = 1,\ldots,N \end{aligned} \qquad \begin{aligned} \min_{\boldsymbol{\alpha},\rho} \quad & -\rho \\ \text{s.t.} \quad & \rho_n(\boldsymbol{\alpha}) - \rho \geq 0 \\ & n = 1,\ldots,N \end{aligned}$$

$$E_\beta(\boldsymbol{\theta}) = g(\boldsymbol{\theta}) + \sum_{n=1}^{N} \kappa_\beta(c_n(\boldsymbol{\theta})) \qquad E_\beta(\boldsymbol{\alpha},\rho) = -\rho + \sum_{n=1}^{N} \kappa_\beta(\rho_N(\boldsymbol{\alpha}) - \rho)$$

Convergence: $\lim_{\beta\to 0} \operatorname{argmin}_{\boldsymbol{\theta}} E_\beta(\boldsymbol{\theta}) = \boldsymbol{\theta}^*$ $\rightsquigarrow$ solution to problem

Barrierfunctions: $\kappa_\beta(t) = -\beta\log(t)$ $\rightsquigarrow$ Log-Barrier $\rightsquigarrow$ Interior Point
$\kappa_\beta(t) = \beta\exp(-t/\beta)$ $\rightsquigarrow$ Exp-Barrier $\rightsquigarrow$ Boosting

AdaBoost$_\varrho$ as Barrier Algorithm



- The exp-barrier objective for the large margin LP:

$$E_\beta(\boldsymbol{\alpha}) = -\rho + \beta \sum_{n=1}^{N} \exp\left(-\frac{\rho_n(\boldsymbol{\alpha}) - \rho}{\beta}\right)$$

- We need $\beta \to 0$, and have $\|\boldsymbol{\alpha}\| \to \infty$: set $\beta := 1/\|\boldsymbol{\alpha}\|_1$:

$$E_\beta(\boldsymbol{\alpha}) = -\rho + \frac{1}{\|\boldsymbol{\alpha}\|_1} \underbrace{\sum_{n=1}^{N} \exp\left(-\rho_n(\boldsymbol{\alpha})\|\boldsymbol{\alpha}\|_1 + \rho\|\boldsymbol{\alpha}\|_1\right)}_{\text{AdaBoost}_\varrho\text{'s Objective}}$$

$\Rightarrow$ **AdaBoost$_\varrho$ minics a Barrier technique for solving a LP!**