Boosting

Foundations and Algorithms

Tuan Quoc Do

Colloquium of Department of Statistics
University of South Carolina

September 21, 2017

Outline

1. Core Analysis
   - Introduction
   - Minimizing training error
   - Margin maximization
2. Fundamental perspectives
   - Loss minimization and generalizations of Boosting
   - Convex optimization and Information geometry
3. Algorithm extensions
   - Confidence-rated weak predictions
   - Multiclass classification
   - Ranking with Boosting

Outline

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \chi$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, ..., m$
For t=1, ... ,T:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \chi \to \{-1, +1\}$.
- Aim: select h_t to minimalize the weighted error:

$$\epsilon_t = \boldsymbol{Pr}_{i \sim D_t}[h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$.
- Update, for i = 1, ..., m:

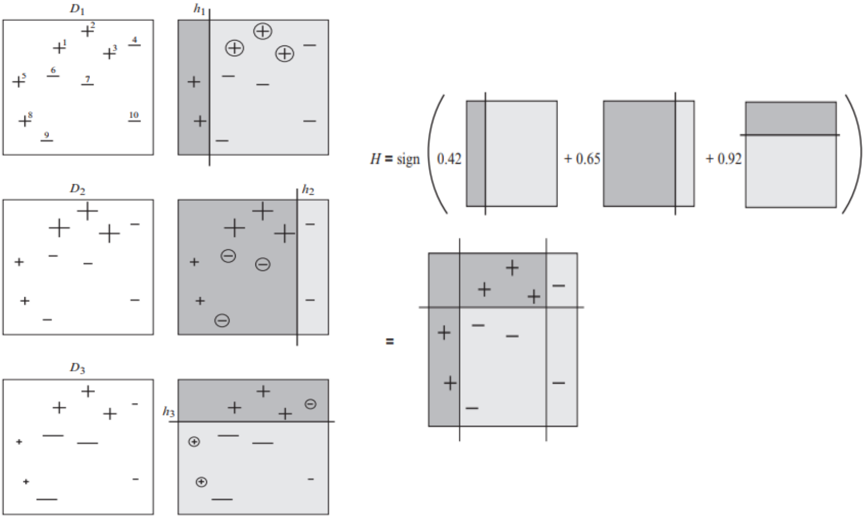
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} x \begin{cases} \mathrm{e}^{-\alpha_t} \textit{ if } h_t(x_i) = y_i \\ \mathrm{e}^{\alpha_t} \textit{ if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) exp\Big(- \alpha_t y_i h_t(x_i)\Big)}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).
Output the final hypothesis:

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

Illustration



Margin and Weak Learnability

Margin

$H(x) = sign(F(x))$ where $F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
Normalize the weights: let $a_t = \frac{\alpha_t}{\sum_{t'=1}^{T} \alpha'_t}$ and $f(x) = \sum_{t=1}^{T} a_t h_t(x) = \frac{F(x)}{\sum_{t=1}^{T} \alpha_t}$
$\Rightarrow$ Then the margin is $yf(x)$ which has range $[-1, +1]$

γ -weak learning assumption

For any distribution D on the indices $\{1, ..., m\}$ of the training examples, the weak learning algorithm A is able to find a hypothesis h with weighted training error at most $\frac{1}{2} - \gamma$ (for $\gamma > 0$):

$$\boldsymbol{Pr}_{i \sim D}[h(x_i) \neq y_i] \leq \frac{1}{2} - \gamma$$

Learnability equivalent

Strong and weak learnability are equivalent. There is nothing in between.

Outline

1. Core Analysis
   - Introduction
   - Minimizing training error
   - Margin maximization
2. Fundamental perspectives
   - Loss minimization and generalizations of Boosting
   - Convex optimization and Information geometry
3. Algorithm extensions
   - Confidence-rated weak predictions
   - Multiclass classification
   - Ranking with Boosting

A Bound on AdaBoosts Training Error

Theorem

Given the notation of AdaBoost, let $\gamma_t = \frac{1}{2} - \epsilon_t$, and let D_1 be an arbitrary initial distribution over the training set. Then the weighted training error of the combined classifier H with respect to D_1 is bounded as:

$$\boldsymbol{Pr}_{i \sim D_1}[H(x_i) \neq y_i] \leq \prod_{t=1}^{T} \sqrt{1 - 4\gamma_t^2} \leq exp(-2 \sum_{t=1}^{T} \gamma_t^2) = exp(-2\gamma^2 T)$$

- The training error drops exponentially fast as a function of the number of base classifiers combined
- If $T > \frac{ln(m)}{2\gamma^2}$ so that $\mathrm{e}^{2\gamma^2 T} < 1/m$, then the training error of the combined classifier, which is always an integer multiple of $1/m$, must in fact be zero

Proof

$$D_{t+1}(i) = D_1(i) \times \frac{\mathrm{e}^{-y_i \alpha_1 h_1(x_i)}}{Z_1} \times ... \times \frac{\mathrm{e}^{-y_i \alpha_T h_T(x_i)}}{Z_T}$$

$$= \frac{D_1(i) exp(-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i))}{\prod_{t=1}^{T} Z_t} = \frac{D_1(i) exp(-y_i F(x_i))}{\prod_{t=1}^{T} Z_t}$$

Therefore

$$\boldsymbol{Pr}_{i \sim D}[H(x_i) \neq y_i] = \sum_{i=1}^{m} D_1(i) \mathbf{1}[H(x_i) \neq y_i] \leq \sum_{i=1}^{m} D_1(i) exp(-y_i F(x_i))$$

$$= \sum_{i=1}^{m} D_{T+1}(i) \prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} Z_t$$

Finally, by the choise of α, we have:

$$Z_t = \sum_{i=1}^{m} D_t(i) \mathrm{e}^{-\alpha_t y_i h_t(x_i)} = \sum_{i:y_i = h_t(x_i)} D_t(i) \mathrm{e}^{-\alpha_t} + \sum_{i:y_i \neq h_t(x_i)} D_t(i) \mathrm{e}^{\alpha_t}$$

$$= \mathrm{e}^{-\alpha_t}(1 - \epsilon_t) + \mathrm{e}^{\alpha_t} \epsilon_t = \mathrm{e}^{-\alpha_t}(\frac{1}{2} + \delta_t) + \mathrm{e}^{\alpha_t}(\frac{1}{2} - \delta_t) = \sqrt{1 - 4\delta_t^2}$$

The Condition for Weak Learnability

Linear separability

Suppose our training sample S is such that for some weak hypotheses $g_1, \ldots, g_k$ from a given space $\mathcal{H}$, and for some nonnegative coefficients $a_1, \ldots, a_k$ with $\sum_{j=1}^{k} a_j = 1$, for some $\gamma > 0$, if the below condition holds:

$$y_i \sum_{j=1}^{k} a_j g_j(x_i) \geq 2\gamma$$

(Or S is linearly separable with margin 2γ), then the assumption of γ-empirical weak learnability holds as well

Proof

Suppose D is any distribution over S, then

$$\sum_{j=1}^{k} a_j \boldsymbol{E}_{i\sim D}[y_i g_j(x_i)] \geq 2\gamma$$

Since a_j's form a distribution, there exist j such that $\boldsymbol{E}_{i\sim D}[y_i g_j(x_i)] \geq 2\gamma$. We have:

$$\boldsymbol{E}_{i\sim D}[y_i g_j(x_i)] = 1 \cdot \boldsymbol{Pr}_{i\sim D}[y_i = g_j(x_i)] + (-1) \cdot \boldsymbol{Pr}_{i\sim D}[y_i \neq g_j(x_i)]$$

$$= 1 - 2\boldsymbol{Pr}_{i\sim D}[y_i \neq g_j(x_i)]$$

Therefore

$$\boldsymbol{Pr}_{i\sim D}[y_i \neq g_j(x_i)] = \frac{1 - \boldsymbol{E}_{i\sim D}[y_i g_j(x_i)]}{2} \leq \frac{1}{2} - \gamma$$

Outline

Generalization Error

Definition: convex hull $co(\mathcal{H})$ of $\mathcal{H}$

$$co(\mathcal{H}) = \{f : x \to \sum_{t=1}^{T} a_t h_t(x) \Big| a_1, \ldots, a_T \geq 0; \sum_{t=1}^{T} a_t = 1; h_1, \ldots, h_T \in \mathcal{H}; T \geq 1\}$$

Finite Base Hypothesis Spaces

Let D be a distribution over $\chi \times \{-1, +1\}$, and let S be a sample of m examples chosen independently at random according to D. Assume that the base classifier space $\mathcal{H}$ is finite, and let $\delta > 0$. Then with probability at least $1 - \delta$ over the random choice of the training set S, every weighted average function $f \in co(\mathcal{H})$ satisfies the following bound:

$$\boldsymbol{Pr}_D[yf(x) \leq 0] \leq \boldsymbol{Pr}_S[yf(x) \leq \theta] + O(\sqrt{\frac{log|\mathcal{H}|}{m\theta^2} \cdot log(\frac{m\theta^2}{log|\mathcal{H}|}) + \frac{log(1/\delta)}{m}})$$

for all $\theta > \sqrt{(ln|\mathcal{H}|)/(4m)}$

Infinite Base Hypothesis Spaces

Let D be a distribution over $\chi \times \{-1, +1\}$, and let S be a sample of m examples chosen independently at random according to D. Assume that the base classifier space $\mathcal{H}$ has VC-dimension d, and let $\delta > 0$. Assume $m \geq d \geq 1$. Then with probability at least $1 - \delta$ over the random choice of the training set S, every weighted average function $f \in co(\mathcal{H})$ satisfies the following bound:

$$\boldsymbol{Pr}_D[yf(x) \leq 0] \leq \boldsymbol{Pr}_S[yf(x) \leq \theta] + O(\sqrt{\frac{d\ log(m/d) log(m\theta^2/d)}{m\theta^2} + \frac{log(1/\delta)}{m}})$$

for all $\theta > \sqrt{8d\ ln(em/d)/m}$

Bounding AdaBoosts Margins

Theorem

Using AdaBoost, given $\gamma_t = \frac{1}{2} - \epsilon_t$, the fraction of training examples with margin at most θ is at most:

$$\prod_{t=1}^{T} \sqrt{(1+2\gamma_t)^{1+\theta}(1-2\gamma_t)^{1-\theta}}$$

- Assume all data points have the same edge γ, if $(1+2\gamma)^{1+\theta}(1-2\gamma)^{1-\theta} < 1$, this bound implies that the fraction of training examples with $yf(x) \leq \theta$ goes to zero exponentially fast with T, and must actually be equal to zero at some point since this fraction must always be a multiple of $1/m$.
- We have $(1+2\gamma)^{1+\theta}(1-2\gamma)^{1-\theta} < 1 \Leftrightarrow \theta < \frac{-ln(1-4\gamma^2)}{ln(\frac{1+2\gamma}{1-2\gamma})} = \Upsilon(\gamma)$
- $\Upsilon(\gamma)$ can be shown to be increasing with γ and in $[\gamma, 2\gamma]$ for $0 \leq \gamma \leq 1/2$

Conclusion: When the γ -weak learning assumption holds, then in the limit of a large number of rounds T , all examples will eventually have margin at least $\Upsilon(\gamma)$
$\Rightarrow$ **Stronger base classifiers lead to larger margins**

Proof

Let $f(x) = \sum_{t=1}^{T} a_t h_t(x) = \frac{F(x)}{\sum_{t=1}^{T} \alpha_t}$ where $a_t = \frac{\alpha_t}{\sum_{t=1}^{T} \alpha_t}$

We have

$$yf(x) \leq \theta \Leftrightarrow y \sum_{t=1}^{T} \alpha_t h_t(x) \leq \theta \sum_{t=1}^{T} \alpha_t \Leftrightarrow exp(-y \sum_{T=1}^{T} \alpha_t h_t(x) + \theta \sum_{t=1}^{T} \alpha_t) \geq 1$$

$\Leftrightarrow \mathbf{1}[yf(x) \leq \theta] \leq exp(-y \sum_{t=1}^{T} \alpha_t h_t(x) + \theta \sum_{t=1}^{T} \alpha_t)$

Therefore

$$\boldsymbol{Pr}_S[yf(x) \leq \theta] = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}[yf(x) \leq \theta] \leq \frac{1}{m} \sum_{i=1}^{m} exp(-y \sum_{T=1}^{T} \alpha_t h_t(x) + \theta \sum_{t=1}^{T} \alpha_t)$$

$$= \frac{exp(\theta \sum_{t=1}^{T} \alpha_t)}{m} \sum_{i=1}^{m} exp(-y \sum_{T=1}^{T} \alpha_t h_t(x)) = exp(\theta \sum_{t=1}^{T} \alpha_t)(\prod_{t=1}^{T} Z_t)$$

More Aggressive Margin Maximization

Goal: Modify AdaBoost to have the minimum margin of 2γ instead of the $\Upsilon(\gamma)$
We have

$$\boldsymbol{Pr}_S[yf(x) \leq \theta] \leq \prod_{t=1}^{T} \left[\mathrm{e}^{(\theta-1)\alpha_t}(\frac{1}{2} + \gamma_t) + \mathrm{e}^{(\theta+1)\alpha_t}(\frac{1}{2} - \gamma_t) \right]$$

Rather than choosing α_t as in AdaBoost, we can instead select α_t to minimize the above equation directly, which gives

$$\alpha_t = \frac{1}{2} ln(\frac{1 + 2\gamma_t}{1 - 2\gamma_t}) - \frac{1}{2} ln(\frac{1+\theta}{1-\theta})$$

which is smaller than α_t chosen by AdaBoost

Assume $\alpha_t \geq 0\,(\gamma_t \geq \theta/2)$, we can plug this choice back and get:

$$\boldsymbol{Pr}_S[yf(x) \leq \theta] \leq exp\Big(- \sum_{t=1}^{T} RE_b(\frac{1}{2} + \frac{\theta}{2} || \frac{1}{2} + \gamma_t) \Big)$$

where $RE_b(p||q) = p\,ln(\frac{p}{q}) + (1-p)ln(\frac{1-p}{1-q})$ for $p, q \in [0, 1]$

So if θ is chosen ahead of time, and if the γ -weak learning assumption holds for some $\gamma > \theta/2$, then the fraction of training examples with margin at most θ will be no more than

$$exp\Big(- T \cdot RE_b(\frac{1}{2} + \frac{\theta}{2} || \frac{1}{2} + \gamma) \Big)$$

Outline

AdaBoost minimizes the exponential loss

In AdaBoost, α_t and h_t are chosen to minimize the exponential loss, which is the upper bound of the training error

$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}\{y_iF(x_i)\leq 0\}\leq\frac{1}{m}\sum_{i=1}^{m}\mathrm{e}^{-y_iF(x_i)}$$

Proof: We showed that

$$\frac{1}{m}\sum_{i=1}^{m}\mathrm{e}^{-y_iF(x_i)}=D_t(i)\Bigg(\prod_{t'=1}^{t-1}Z_{t'}\Bigg)$$

This implies that

$$\frac{1}{m}\sum_{i=1}^{m}\mathrm{e}^{-y_iF(x_i)}=\frac{1}{m}\sum_{i=1}^{m}exp\Bigg(-y_i(F_{t-1}(x_i)+\alpha_ih_i(x_i))\Bigg)$$

$$=\sum_{i=1}^{m}D_t(i)\Bigg(\prod_{t'=1}^{t-1}Z_{t'}\Bigg)\mathrm{e}^{-y_i\alpha_th_t(x_i)}\propto\sum_{i=1}^{m}D_t(i)\mathrm{e}^{-y_i\alpha_th_t(x_i)}=Z_t$$

Moreover we have $Z_t=\mathrm{e}^{-\alpha_t}(1-\epsilon_t)+\mathrm{e}^{\alpha_t}\epsilon_t$, which is minimized at α_t chosen by AdaBoost. The minimum is $Z_t=2\sqrt{\epsilon_t(1-\epsilon_t)}$ which is monotonically increasing for $0\leq\epsilon_t\leq 1/2$, and decreasing for $1/2\leq\epsilon_t\leq 1$

Algorithm for minimizing exponential loss

Algorithm

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \chi$, $y_i \in \{-1, +1\}$.
Initialize $F_0 \equiv 0$.
For $t = 1, ..., T$:

- Choose $h_t \in \mathcal{H}, \alpha \in \mathcal{R}$ to minimize

$$\frac{1}{m} \sum_{i=1}^{m} exp\Big(- y_i(F_{t-1}(x_i) + \alpha_i h_i(x_i))\Big)$$

- Update $F_t = F_{t-1} + \alpha_t h_t$

Output F_t

Coordinate descent

Algorithm

Goal: minimization of $L(\lambda_1, ..., \lambda_N)$.
Initialize: $\lambda_j \leftarrow 0$ for $j = 1, ..., N$
For $t = 1, ..., T$:

- Let j, α minimize $L(\lambda_1, ..., \lambda_{j-1}, \lambda_j + \alpha, \lambda_{j+1}, ..., \lambda_N)$ over $j \in \{1, ...N\}$, $\alpha \in \mathcal{R}$
- $\lambda_j \leftarrow \lambda_j + \alpha$

Output $\lambda_1, ..., \lambda_N$

* Can be used with other loss functions

Functional Gradient Descent

Algorithm

Goal: minimization of $\mathcal{L}(F)$.
Initialize: $F_0 \leftarrow 0$
For $t = 1, ..., T$:

- Select $h_t \in \mathcal{H}$ that maximize $-\nabla\mathcal{L}(F_{t-1})$
- Choose $\alpha_t > 0$
- Update $F_t = F_{t-1} + \alpha_t h_t$

Output F_t

In AdaBoost, the partial derivative is $\frac{\partial \mathcal{L}(F)}{\partial F(x_i)} = \frac{-y_i \mathrm{e}^{-y_i F(x_i)}}{m}$
Thus on round t, the goal is to find h maximizing

$$\frac{1}{m}\sum_{i=1}^{m} y_i h_t(x_i)\mathrm{e}^{-y_i F(x_i)} \propto \sum_{i=1}^{m} D_t(x_i) y_i h_t(x_i) = 1 - 2\epsilon_t$$

*If we fix α on all round, we have **$\boldsymbol{\alpha}$-Boost**

Gradient Boosted Models

Arguably the most popular boosting model

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \chi$.
Initialize: $f_0(x) = \arg\min_\gamma \sum_{i=1}^m \mathcal{L}(y_i, \gamma)$
For t=1, ... ,T:

- For i = 1,2,...,m compute:

$$r_{it} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{t-1}}$$

- Fit a learner h_t to the targets r_{it}
- Compute step magnitude γ_t using line search :

$$\gamma_t = \arg\min_\gamma {}_t \sum_{i=1}^m \mathcal{L}\Big(y_i, f_{t-1}(x_i) + \gamma_t h_t(x_i)\Big)$$

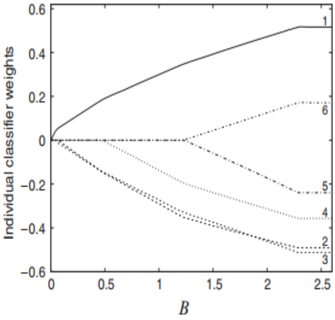
- Update (υ is a fixed shrinkage parameter and in $(0, 1]$):

$$f_t(x) = f_{t-1}(x) + \upsilon\gamma_t h_t(x)$$

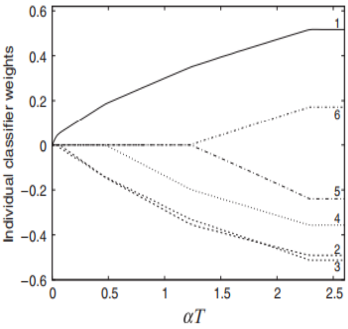
Output: $f_T(x)$

Connection with L-1 regularization

- We can apply L-1 constraint to the weight vector: minimize $L(\boldsymbol{\lambda})$ subject to $||\boldsymbol{\lambda}||_1 \leq B$
- We can also use **BrownBoost** to handle noisy data
- Or we can stop α-Boost early (α-Boost for T rounds $\equiv$ using L-1 penalty with $B = \alpha T$)



(a) L-1



(b) α-Boost

Explanation of the similarity

- Assume we know $\boldsymbol{\lambda}$ is the solution for some $B \geq 0$, and want to find $\boldsymbol{\lambda}'$ which is the solution when we increase B by some tiny $\alpha > 0$
- We have $||\lambda||_1 = B$ and $||\lambda^{'}||_1 = B + \alpha$
- $B + \alpha = ||\lambda^{'}||_1 = ||\lambda + \delta||_1 \leq ||\lambda||_1 + ||\delta||_1 = B + ||\delta||_1$
- Equality hold when $\lambda_j \delta_j \geq 0 \ \forall j$, implying $||\delta||_1 = \alpha$
- By Taylor expansion:

$$L(\boldsymbol{\lambda} + \boldsymbol{\delta}) \approx L(\boldsymbol{\lambda}) + \nabla L(\boldsymbol{\lambda}) \cdot \boldsymbol{\delta} = L(\boldsymbol{\lambda}) + \sum_{j=1}^{N} \frac{\partial L(\boldsymbol{\lambda})}{\partial \lambda_j} \cdot \delta_j$$

- Given $||\boldsymbol{\delta}||_1 = \alpha$, RHS is minimized when $\boldsymbol{\delta}$ is all zeros, except for the component j where $|\partial L(\boldsymbol{\lambda})/\partial \lambda_i|$ is the largest, which is set to $-\alpha \cdot sign(\partial L(\boldsymbol{\lambda})/\partial \lambda_i)$

Outline

Iterative projection algorithm

Algorithm

Given: $\boldsymbol{a}_j \in \mathcal{R}^m, b_j \in \mathcal{R}$ for $j = 1, ..., N$, $\boldsymbol{x}_0 \in \mathcal{R}^m$
Problem: minimize $||\boldsymbol{x} - \boldsymbol{x}_0||_2^2$ subject to $\boldsymbol{a}_j \cdot \boldsymbol{x} = b_j$ for $j = 1, ..., N$
Goal: find sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, ...$ converging to the solution for the problem
Initialize: $\boldsymbol{x}_1 = \boldsymbol{x}_0$
For $t = 1, 2, ...$:

- Choose a constraint j
- Let $\boldsymbol{x}_{t+1} = \arg \min_{\boldsymbol{x}: a_j \cdot \boldsymbol{x} = b_j} ||\boldsymbol{x} - \boldsymbol{x}_t||_2^2$

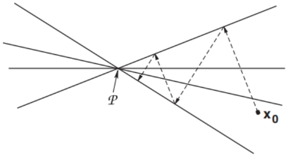


Figure: How the iterative projection algorithm proceeds with greedy selection

Convex optimization perspective

Entropy

- The entropy of a distribution is: $H(P) = -\sum_{i=1}^{m} P(i) lnP(i)$
- The relative entropy, or the KullbackLeibler divergence from Q to P is:

$$RE(P||Q) = (-\sum_{i=1}^{m} P(i) lnQ(i)) - H(P) = \sum_{i=1}^{m} P(i) ln(\frac{Q(i)}{P(i)})$$

- We have $D_{t+1}(i) = \frac{D_t(i)\mathrm{e}^{-\alpha_t y_i h_t(x_i)}}{Z_t}$. Therefore:

$$\sum_{i=1}^{m} D_{t+1}(i) y_i h_t(x_i) = \frac{1}{Z_t} \sum_{i=1}^{m} D_t(i)\mathrm{e}^{-\alpha_t y_i h_t(x_i)} y_i h_t(x_i) = -\frac{1}{Z_t} \cdot \frac{dZ_t}{d\alpha_t} = 0$$

where $Z_t = \sum_{i=1}^{m} D_t(i)\mathrm{e}^{-\alpha_t y_i h_t(x_i)}$

- So the feasible set: $\mathcal{P} = \{D : \sum_{i=1}^{m} D(i) y_i h_j(x_i) = 0 \text{ for } j = 1, ..., N\}$

The problem:

minimize: RE$(D||U)$
subject to: $\sum_{i=1}^{m} D(i) y_i h_j(x_i) = 0$ for $j = 1, ..., N$ and D is a distribution

Iterative projection algorithm corresponding to AdaBoost

Algorithm

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \chi$, $y_i \in \{-1, +1\}$.
Assume: finite, binary hypothesis base $\mathcal{H}$
Goal: find sequence D_1, D_2,... converging to the solution
Initialize $D_1 = U$.
For $t = 1, 2...$

- Choose $h_t \in \mathcal{H}$ defining one of the constraints
- Let $D_{t+1} = \arg \min_{D: \sum_{i=1}^{m} D(i) y_i h_j(x_i) = 0} \mathrm{RE}(D || D_t)$
- Greedy constraint selection: Choose $h_t \in \mathcal{H}$ so that $\mathrm{RE}(D_{t+1} || D_t)$ is maximized

Theorem for nonempty feasible set

The feasible set P is empty if and only if the data is empirically γ -weakly learnable for some $\gamma > 0$

Proof of equivalence

a) We have the Lagrangian:

$$\mathcal{L} = \sum_{i=1}^{m} D(i) ln(\frac{D(i)}{D_t(i)}) + \alpha \sum_{i=1}^{m} D(i) y_i h_j(x_i) + \mu(\sum_{i=1}^{m} D(i) - 1)$$

Computing derivatives and equating with zero, we get:

$$0 = \frac{\partial \mathcal{L}}{\partial D(i)} = ln(\frac{D(i)}{D_t(i)}) + 1 + \alpha y_i h_t(x_i) + \mu$$

Thus: $D(i) = D_t(i) \mathrm{e}^{-\alpha y_i h_t(x_i) - 1 - \mu}$

Since D is a distribution, μ will be chosen so that $D(i) = \frac{D_t(i) \mathrm{e}^{-\alpha y_i h_t(x_i)}}{Z}$

Plugging back to the Lagrangian, we have: $\mathcal{L} = -lnZ$

b)

$$RE(D_{t+1} || D_t) = \sum_{i=1}^{m} D_{t+1}(i)(-\alpha_t y_i h_t(x_i) - lnZ_t)$$

$$= -lnZ_t - \alpha_t \sum_{i=1}^{m} D_{t+1}(i) y_i h_t(x_i) = -lnZ_t$$

Outline

AdaBoost with confidence-rated predictions

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \chi$, $y_i \in \{-1, +1\}$.
Initialize: $D_i(i) = 1/m$ for $i = 1, ..., m$
For t=1, ... ,T:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \chi \to \mathcal{R}$ and $\alpha_t \in \mathcal{R}$
- Aim: select h_t and α_t to minimalize the normalization factor:

$$Z_t = \sum_{i=1}^{m} D_t(i) \mathrm{e}^{-\alpha_t y_i h_t(x_i)}$$

- Update, for i = 1, ..., m:

$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Output the final hypothesis:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Predictions with Bounded Range

Range [-1,+1]

Let $z_i = y_i h(x_i)$, which is in the range $[-1, +1]$ as well, so

$$Z = \sum_{i=1}^{m} D(i)\mathrm{e}^{-\alpha z_i} = \sum_{i=1}^{m} D(i) exp\Big(-\alpha(\frac{1+z_i}{2}) + \alpha(\frac{1-z_i}{2})\Big)$$

$$\leq \sum_{i=1}^{m} D(i)\Big[(\frac{1+z_i}{2})\mathrm{e}^{-\alpha} + (\frac{1-z_i}{2})\mathrm{e}^{\alpha}\Big] = \frac{\mathrm{e}^{\alpha}+\mathrm{e}^{-\alpha}}{2} - \frac{\mathrm{e}^{\alpha}-\mathrm{e}^{-\alpha}}{2} r$$

where $r = r_t = \sum_{i=1}^{m} D(i) y_i h_t(x_i)$

The RHS is minimized when $\alpha = \frac{1}{2} ln(\frac{1+r}{1-r})$

Plugging back gives: $Z \leq \sqrt{1-r^2}$

So the upper bound of training error is

$$\prod_{t=1}^{T} \sqrt{1-r_t^2}$$

Weak Hypotheses That Abstain

Range $\{-1, 0, +1\}$

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \chi$, $y_i \in \{-1, +1\}$.
Assume: weak hypotheses $h_1, ..., h_N$ with range $\{-1, 0, +1\}$
Initialize:

- $A_b^j = \{1 \leq i \leq m : y_i h_j(x_i) = b\}$ for $j = 1, ..., N$ and for $b \in \{-1, +1\}$
- $d(i) \leftarrow 1$ for $i = 1, ..., m$

For $t = 1, ..., T$:

- For $j = 1, ..., N$:
  - $U_b^j \leftarrow \sum_{i \in A_b^j} d(i)$ for $b \in \{-1, +1\}$
  - $G_j \leftarrow |\sqrt{U_+^j} - \sqrt{U_-^j}|$
- $j_t = \arg\max_{1 \leq j \leq N} G_j$
- $\alpha_t = \frac{1}{2} ln(\frac{U_+^{j_t}}{U_-^{j_t}})$
- for $b \in \{-1, +1\}$, for $i \in A_b^{j_t}$: $d(i) \leftarrow d(i)\mathrm{e}^{-\alpha_t b}$

Output the final hypothesis:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_{j_t}(x))$$

Domain-Partitioning Weak Hypotheses

Define the weighted fraction of examples which fall in block j and which are labeled b:

$$W_b^j = \sum_{i:x_i \in X_j \wedge y_i = b} D(i) = \boldsymbol{Pr}_{i \sim D}[x_i \in X_j \wedge y_i = b]$$

Then we have

$$Z = \sum_{j=1}^{J} \sum_{i:x_i \in X_j} D(i) exp(-y_i c_j) = \sum_{j=1}^{J} (W_+^j \mathrm{e}^{-c_j} + W_-^j \mathrm{e}^{c_j})$$

assuming without loss of generality that the weak learner can freely scale any weak hypothesis h by any constant factor $\alpha \in R$

Using standard calculus, we see that this is minimized when

$$c_j = \frac{1}{2} ln(\frac{W_+^j}{W_-^j})$$

Plugging back gives

$$Z = 2 \sum_{j=1}^{J} \sqrt{W_+^j W_-^j}$$

Outline

AdaBoost.M1

Direct multiclass extension of AdaBoost

Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in \chi$, $y_i \in \mathcal{Y}$.
Initialize: $D_i(i) = 1/m$ for $i = 1, ..., m$
For t=1, ... ,T:

- Train weak learner using distribution D_t.
- Select weak hypothesis h_t: $\chi \rightarrow \mathcal{Y}$ to minimalize the weighted error:

$$\epsilon_t = \boldsymbol{Pr}_{i \sim D_t}[h_t(x_i) \neq y_i].$$

- If $\epsilon_t \geq \frac{1}{2}$, then set $T = t - 1$ and exit loop
- Choose $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$.
- Update, for i = 1, ..., m:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \epsilon^{-\alpha_t} \; if \; h_t(x_i) = y_i \\ \epsilon^{\alpha_t} \; if \; h_t(x_i) \neq y_i \end{cases}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution)

Output the final hypothesis:

$$H(x) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} \alpha_t \mathbf{1}\{h_t(x) = y\}$$

AdaBoost.MH

Multiclass, multi-label version of AdaBoost based on Hamming loss

Definition: Let $\mathcal{H} : \chi \to 2^{\mathcal{Y}}$, then **Hamming Loss** of H is: $\frac{1}{K} \cdot \boldsymbol{E}_{(x,Y)}\Big[|H(x) \Delta Y|\Big]$

Given: $(x_1, Y_1), ..., (x_m, Y_m)$ where $x_i \in \chi,\ Y_i \subseteq \mathcal{Y}$.
Initialize: $D_1(i, \ell) = 1/(mK)$ for $i = 1, ..., m$ and $\ell \in \mathcal{Y}$ (where $K = |\mathcal{Y}|$)
For t=1, ... ,T:

- Train weak learner using distribution D_t.
- Select weak hypothesis h_t: $\chi \times \mathcal{Y} \to R$ and $\alpha_t \in R$ to minimalize:

$$Z_t = \sum_{i=1}^{m} \sum_{\ell \in \mathcal{Y}} D_t(i, \ell) exp\Big(- \alpha_t Y_i[\ell] h_t(x_i, \ell)\Big)$$

- Update, for i = 1, ..., m and for $\ell \in \mathcal{Y}$:

$$D_{t+1}(i, \ell) = \frac{D_t(i, \ell) exp\Big(- \alpha_t Y_i[\ell] h_t(x_i, \ell)\Big)}{Z_t}$$

Output the final hypothesis:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x, \ell))$$

Analysis

We can use domain-partitioning weak hypotheses in this algorithm

- Suppose h is associated with a partition $X_1, ..., X_J$, we create a partition of the set $\chi \times \mathcal{Y}$ consisting of all set $X_j \times \{\ell\}$ for $j = 1, ..., J$ and $\ell \in \mathcal{Y}$.
- An appropriate hypothesis h can then be formed which predicts $h(x, \ell) = c_{j\ell}$ for $x \in \chi_j$
- Using similar techniques in confidence-rated predictors, we would choose:

$$c_{j\ell} = \frac{1}{2} ln(\frac{W_+^{j\ell}}{W_-^{j\ell}})$$

where

$$W_b^{j\ell} = \sum_{i=1}^{m} D(i, \ell)\mathbf{1}\{x_i \in X_j \wedge Y_i[\ell] = b\}$$

- Plugging back gives:

$$Z_t = \sum_{j=1}^{J} \sum_{\ell \in \mathcal{Y}} \sqrt{W_+^{j\ell} W_-^{j\ell}}$$

Relation to One-Error and Single-Label Classification

If the goal is to minimize one-error, in AdaBoost.MH, we can define the final output as:

$$H^1(x) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} \alpha_t h_t(x, y)$$

Theorem

With respect to any distribution D over observations (x, Y) with $\emptyset \neq Y \subseteq \mathcal{Y}$ we have

$$one - err_D(H^1) \leq K \; hloss_D(H)$$

where $K = |\mathcal{Y}|$

This means that AdaBoost.MH can be applied to single-label multiclass classification problems, and the bound on the training error of the final hypothesis is

$$K \prod_{t=1}^{T} Z_t$$

Outline

RankBoost

Using a pair-based weak learner

Given: a finite set $V \subseteq \chi$ of training instances

the set $E \subseteq V \times V$ of preference pairs $\left((u,v) \text{ such that } u < v\right)$

Initialize: for all u, v, let $D_1(u, v) = 1/|E| \times \mathbf{1}[(u, v) \in E]$

For t=1, ... ,T:

- Train weak learner using distribution D_t.
- Select $h_t : \chi \rightarrow R$ and $\alpha_t \in R$ to minimalize the normalization factor:

$$Z_t = \sum_{u,v} D_t(u, v) exp\left(\frac{1}{2}\alpha_t(h_t(u) - h_t(v))\right)$$

- Update, for all u, v:

$$D_{t+1}(u, v) = \frac{D_t(u, v) exp\left(\frac{1}{2}\alpha_t(h_t(u) - h_t(v))\right)}{Z_t}$$

Output the final ranking:

$$F(x) = \frac{1}{2}\sum_{t=1}^{T} \alpha_t h_t(x)$$

Analysis

a)For any given weak ranking h, Z can be viewed as a convex function of α with a unique minimum that can be found numerically, for instance via a simple binary search

b) If h has the range $\{-1, +1\}$, $\frac{1}{2}(h(v) - h(u))$ has range $\{-1, 0, +1\}$. We can use techniques from "Weak Hypotheses That Abstain" part to minimize Z analytically Specifically, for $b \in \{-1, 0, +1\}$, let

$$U_b = \sum_{u,v} D(u, v)\mathbf{1}\{h(v) - h(u) = 2b\} = \boldsymbol{Pr}_{(u,v)\sim D}[h(v) - h(u) = 2b]$$

Then

$$Z = U_0 + U_- \mathrm{e}^{\alpha} + U_+ \mathrm{e}^{-\alpha}$$

It can be verified that Z is minimized when

$$\alpha = \frac{1}{2} ln(\frac{U_+}{U_-})$$

which yields

$$Z = U_0 + 2\sqrt{U_- U_+}$$

Reference

- Robert E. Schapire and Yoav Freund. Boosting: Foundations and Algorithms. MIT Press, 2012
- Hastie, T., Tibshirani, R., Friedman, J. (2009). The elements of statistical learning. Springer series in statistics.
- James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning. Springer Science Business Media; 2013