AdaBoost and Information Geometry

Yuncong Chen

CSE 254: Online Learning

Department of Computer Science
University of California, San Diego

March 11, 2014

AdaBoost [1]

- Input: a pool of weak rules $\mathcal{H}$, labeled training data (x_i, y_i), initial sample weight distribution D_0.
- Weak Learner : Find a weak rule $h_t \in \mathcal{H}$ that gives the smallest **weighted** error ϵ_t under D_t
- Booster : Adjust sample weights

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

  where $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$ and $Z_t = \sum_i D_t(i) \exp\{-\alpha_t y_i h_t(x_i)\}$
- Repeat until convergence or stop early
- Output: Strong rule $F(x) = \sum_t \alpha_t h_t(x)$

Some Insight

- $D_t(i) = \frac{1}{m} \prod_{t'=1}^{t-1} \frac{e^{-y_i \alpha_{t'} h_{t'}(x_i)}}{Z_{t'}} \propto e^{-y_i F_{t-1}(x_i)}$
  - at the end of each round, the weight of an example is proportional to its loss
- $\sum_i e^{-y_i F_t(x_i)} = \sum_i \exp(-y_i(F_{t-1}(x_i) + \alpha_t h_t(x_i)) \propto \sum_i D_t(i) e^{-y_i \alpha_t h_t(x_i)} \doteq Z_t$
  - total loss is proportional to Z_t
- α_t and h_t are chosen to minimize $Z_t(\alpha_t, \epsilon_t) = e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t$
  - $\min_{\epsilon_t, \alpha_t} Z_t = \min_{\epsilon_t}(\min_{\alpha_t} Z_t) = \min_{\epsilon_t} 2\sqrt{\epsilon_t(1-\epsilon_t)}$. This justifies the choice of α_t.
  - Optimal ϵ_t is as close to 0 as possible. This justifies that h_t should minimize weighted error, or maximize correlation with labels under current distribution $\sum_i D_t(i) y_i h_t(x_i)$.
- After α_t and h_t are chosen, booster constructs new distribution D_{t+1} such that correlation with h_t is zero: $\sum_i D_{t+1}(i) y_i h_t(x_i) = \frac{1}{Z_t} \sum_i D_t(i) e^{-\alpha_t y_i h_t(x_i)} y_i h_t(x_i) = -\frac{1}{Z_t} \frac{dZ_t}{d\alpha_t} = 0$

Alternative View of one AdaBoost Iteration

- Weak Learner : Given D_t, find $h_t \in \mathcal{H}$ to

$$\max_{h_t} \sum_i D_t(i) y_i h_t(x_i)$$

- Booster : Given h_t, compute D_{t+1} such that

$$\sum_i D_{t+1}(i) y_i h_t(x_i) = 0$$

Goal of Booster

Pursue a distribution D such that

$$\sum_i D(i) y_i h_j(x_i) = 0$$

for every $h_j \in \mathcal{H}$.

Alternative View of one AdaBoost Iteration

- Weak Learner : Given D_t, find $h_t \in \mathcal{H}$ to

$$\max_{h_t} \sum_i D_t(i) y_i h_t(x_i)$$

- Booster : Given h_t, compute D_{t+1} such that

$$\sum_i D_{t+1}(i) y_i h_t(x_i) = 0$$

Goal of Booster

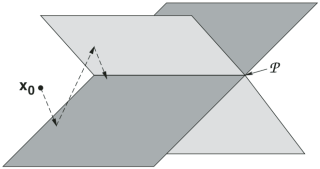
Pursue a distribution D such that

$$\sum_i D(i) y_i h_j(x_i) = 0$$

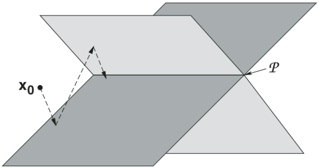
for every $h_j \in \mathcal{H}$.

$\boxed{\textbf{Linear constraints in } D}$

Information Geometry Perspective



Information Geometry Perspective



Optimization Problem corresp. to AdaBoost

$$\min_D RE(D||U)$$

s.t.

$$\sum_i D(i) y_i h_j(x_i) = 0, \forall j$$

$$D(i) \geq 0, \forall i$$

$$\sum_i D(i) = 1$$

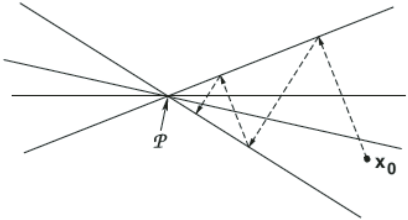
$$RE(p, q) = \sum_i p_i \log \frac{p_i}{q_i}$$

Assume feasible set $\mathcal{P}$ is not empty for now.

Solve the Program using Iterative Projection

- Initialize $D_1 = U$
- choose $h_t \in \mathcal{H}$ defining one of the constraints
- let $D_{t+1} = \arg\min_{D: \sum_i D(i) y_i h_t(x_i) = 0} RE(D || D_t)$
- repeat

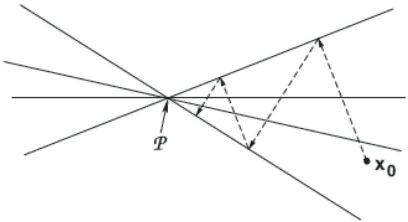
Greedy constraint selection: Choose h_t so that $RE(D_{t+1} || D_t)$ is maximized.



Solve the Program using Iterative Projection

- Initialize $D_1 = U$
- choose $h_t \in \mathcal{H}$ defining one of the constraints (weak learner)
- let $D_{t+1} = \arg\min_{D:\sum_i D(i) y_i h_t(x_i)=0} RE(D||D_t)$ (booster)
- repeat

Greedy constraint selection: Choose h_t so that $RE(D_{t+1}||D_t)$ is maximized.



Claim

Each round of Iterative Projection is equivalent to that of AdaBoost.

Proof

Weak Learner

Find h_t to maximize

$$RE(D_{t+1}||D_t) = \sum_i D_{t+1}(i)(-\alpha_t y_i h_t(x_i) - \ln Z_t) = -\ln Z_t$$

Equiv. to choosing h_t to minimize Z_t, exactly what AdaBoost does.

Booster

$$\max_{\alpha,\mu} \min_D \mathcal{L}(\alpha, \mu, D) = RE(D||D_t) + \alpha \sum_i D(i) y_i h_t(x_i) + \mu \left(\sum_i D(i) - 1 \right)$$

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

$$D^*(i) = D_t(i) \exp\{-\alpha y_i h_t(x_i) - 1 - \mu\} = \frac{1}{Z(\alpha)} D_t(i) \exp\{-\alpha y_i h_t(x_i)\}$$

$$\mathcal{L}(\alpha) = -\ln Z(\alpha)$$

AdaBoost also chooses α to minimize Z, same α gives same D.

What if Feasible Set $\mathcal{P}$ is Empty

$$\mathcal{P} = \left\{ D : \sum_i D(i) y_i h_j(x_i) = 0, \forall h_j \in \mathcal{H} \right\}$$

- $\mathcal{P}$ is empty = data is weakly learnable = data linearly seperable
- iterative projection never converge

Alternative Characterization of AdaBoost

Original characterization using normalized distribution

$$\min_{D \in \Delta^{m-1}} RE(D||U)$$

s.t.

$$\sum_i D(i) y_i h_j(x_i) = 0, \forall j$$

Optimization Problem using unnormalized weight vector

$$\min_{d \in R_+^m} RE_u(d||\mathbf{1})$$

s.t.

$$\sum_i d_i y_i h_j(x_i) = 0, \forall j$$

- Distance measure is **unnormalized relative entropy**

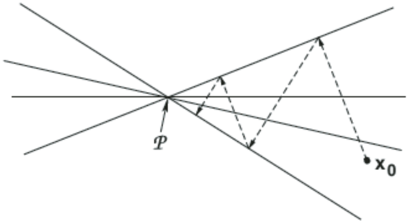
$$RE_u(p||q) = \sum_i p_i \log \frac{p_i}{q_i} + q_i - p_i$$

- $\mathcal{P}$ contains at least $\mathbf{0}$

Iterative Projection using Unnormalized *RE*

- Initialize $d_1 = \mathbf{1}$
- choose $h_t \in \mathcal{H}$ defining one of the constraints (weak learner)
- let $d_{t+1} = \arg\min_{d: \sum_i d(i) y_i h_t(x_i) = 0} RE_u(d || d_t)$ (booster)
- repeat

Greedy constraint selection: Choose h_t so that $RE_u(d_{t+1} || d_t)$ is maximized.



Claim (proof similar to before)

Each round of Iterative Projection, after normalizing d, is equivalent to that of AdaBoost. (can also think as if we directly give unnormalized weights to weak learner)

Prove convergence of AdaBoost

Goal 1: Prove d converges to the optimum via iterative projection

$\boxed{d_t \to \arg\min_p RE_u(p||\mathbf{1})}$

Goal 2: Prove AdaBoost minimizes exponential loss

$$\text{Define } \mathcal{Q} = \left\{ q : q_i = \exp\{-y_i \sum_{j=1}^{N} \lambda_j h_j(x_i)\}, \forall \lambda_j \in \mathbb{R} \right\}$$

$$\text{minimum loss} = \inf_{q \in \mathcal{Q}} \sum_i q_i = \min_{q \in \bar{\mathcal{Q}}} \sum_i q_i = \min_{q \in \bar{\mathcal{Q}}} RE_u(\mathbf{0}||q)$$

$$\text{algorithm loss} = \sum_i \exp\{-y_i \sum_{\tau=1}^{t} \alpha_\tau h_\tau(x_i)\} = \sum_i d_{t+1,i} = \text{total weight}$$

algorithm loss $\to$ minimal loss can be shown by proving

$\boxed{d_t \to \arg\min_{q \in \bar{\mathcal{Q}}} RE_u(\mathbf{0}||q)}$.

Prove convergence of AdaBoost

Proof Outline

- If $d \in \mathcal{P} \cap \bar{\mathcal{Q}}$,
  then $RE_u(p||q) = RE_u(p||d) + RE_u(d||q)$ (Pythagorean Thm.);
  thus d uniquely solves $\min_{p \in \mathcal{P}} RE_u(p||\mathbf{1})$ and $\min_{q \in \bar{\mathcal{Q}}} RE_u(\mathbf{0}||q)$
- d_t computed by iterative projection converges to the unique point $d^* \in \mathcal{P} \cap \bar{\mathcal{Q}}$.
  - Since loss ≥ 0 and non-increasing, the drop in loss must converge to zero.
  - If the drop in loss $= 0$, then $d \in \mathcal{P}$. Thus, $d^* \in \mathcal{P}$.
  - The way d is constructed implies $d^* \in \bar{\mathcal{Q}}$

- From the weights' perspective, this shows
  - if data is not weakly learnable, d converges to $d^* \neq \mathbf{0}$; normalizing $d*$ gives D^*.
  - if data weakly learnable, d converges to $\mathbf{0}$, the only element in $\mathcal{P} \cap \bar{\mathcal{Q}}$. No conclusion about the limit behavior of the normalized distribution.
- From the loss's perspective, this proves AdaBoost minimizes exponential loss asymptotically in the limit of a large number of iterations.

Two Optimization Problems are Duals

Primal

$$\min_{p \in \mathcal{P}} RE_u(p||\mathbf{1})$$

where

$$\mathcal{P} \doteq \{p \in \mathbb{R}_+^m | \sum_j p_j y_i h_j(x_i) = \mathbf{0}\}$$

Dual

$$\min_{q \in \bar{\mathcal{Q}}} RE_u(\mathbf{0}||q)$$

$$= \min_{\lambda \in \mathbb{R}^n} \sum_i e^{-\sum_j y_i h_j(x_i)\lambda_j}$$

where

$$\mathcal{Q} = \{q \in \mathbb{R}_+^m | q_i = e^{-\sum_j y_i h_j(x_i)\lambda_j}, \lambda \in \mathbb{R}^n\}$$

Generalize to Bregman Divergence Optimization [2]

For convex function F, the induced Bregman divergence:

$$B_F(p||q) = F(p) - F(q) - \nabla F(q)(p - q)$$

Primal

$$\min_{p \in \mathcal{P}} B_F(p||q_0)$$

where

$$\mathcal{P} \doteq \{p \in S : p^T M = \tilde{p}^T M\}$$

Dual

$$\min_{q \in \mathcal{Q}} B_F(\tilde{p}||q)$$

where

$$\mathcal{Q} \doteq \{\mathcal{L}_F(q_0, M\lambda) | \lambda \in \mathbb{R}^n\}$$

$$\mathcal{L}_F : \mathcal{S} \times \mathbb{R}^m \rightarrow \mathcal{S}$$

$$\mathcal{L}_F(q, v) = (\nabla F)^{-1}(\nabla F(q) - v)$$

Theorem: For a large family of Bregman divergences, there exists a unique d^* satisfying:

- $d^* \in \mathcal{P} \cap \bar{\mathcal{Q}}$
- $B_F(p||q) = B_F(p||d^*) + B_F(d^*||q), \forall p \in \mathcal{P}, q \in \bar{\mathcal{Q}}$
- $d^* = \arg\min_{q \in \bar{\mathcal{Q}}} B_F(\tilde{p}||q)$
- $d^* = \arg\min_{p \in \mathcal{P}} B_F(p||q_0)$

AdaBoost

- $F = \sum_i p_i \log p_i$
- $B_F = RE_u$
- $M_{ij} = y_i h_j(x_i)$

Primal

$$\min_{p \in \mathcal{P}} RE_u(p || \mathbf{1})$$

where

$$\mathcal{P} \doteq \{p \in \mathbb{R}^m_+ : p^T M = \mathbf{0}\}$$
$$= \{p \in \mathbb{R}^m_+ \sum_j p_j y_i h_j(x_i) = \mathbf{0}\}$$

Dual

$$\min_{q \in \mathcal{Q}} RE_u(\mathbf{0} || q)$$
$$= \min_{\lambda \in \mathbb{R}^n} \sum_i e^{-\sum_j y_i h_j(x_i) \lambda_j}$$

where

$$\mathcal{L}_F(q, v)_i = q_i e^{-v_i}$$

$$\mathcal{Q} = \{q \in \mathbb{R}^m_+ | q_i = e^{-\sum_j y_i h_j(x_i) \lambda_j}, \lambda \in \mathbb{R}^n\}$$

Logistic Regression

- $F = \sum_i p_i \log p_i + (1 - p_i) \log(1 - p_i)$
- B_F = binary relative entropy $= \sum_i p_i \log \frac{p_i}{q_i} + (1 - p_i) \log \frac{1-p_i}{1-q_i}$
- $M_{ij} = y_i h_j(x_i)$

Primal

$$\min_{p \in \mathcal{P}} BinRelEnt(p || \frac{1}{2}\mathbf{1})$$

where

$$\mathcal{P} \doteq \{p \in [0,1]^m : p^T M = \mathbf{0}\}$$

$$= \{p \in [0,1]^m \sum_j p_j y_i h_j(x_i) = \mathbf{0}\}$$

Dual

$$\min_{q \in \mathcal{Q}} BinRelEnt(\mathbf{0} || q)$$

$$= \min_{\lambda \in \mathbb{R}^n} \sum_i \ln \left(1 + e^{-y_i \sum_j \lambda_j h_j(x_i)}\right)$$

where

$$\mathcal{L}_F(q, v)_i = \frac{q_i e^{-v_i}}{1 - q_i + q_i e^{-v_i}}$$

$$\mathcal{Q} = \{q \in [0,1]^m | q_i = \sigma \left(\sum_j y_i h_j(x_i) \lambda_j\right),$$

$$\lambda \in \mathbb{R}^n\}$$

References

[1] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*.
MIT Press, 2012.

[2] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," *Machine Learning*, vol. 48, no. 1-3, pp. 253–285, 2002.