Online Learning: Wrap Up
&
Boosting: Interpretations, Extensions, Theory

Arindam Banerjee
Online: “Things we liked”

- Adaptive (Natural)
- Computationally efficient
- Good guarantees [2]
- Gradient descent based
- No assumptions [2]
Online: “Things we didn’t like”

- Assumes some good expert or oracle [6]
- Not “robust” (noisy data)
- Tracking of outliers
- Intuition about convergence rates
- Unsupervised versions
- Stock market assumptions
- Worst-case focus (conservative)
- Learning rate (how to choose?)
Online: “General Questions”

- Relations with Bayesian inference (e.g., Dirichlet)
- Convergence proofs are good, but what about intuitive algorithms
- Is there a Batch to Online continuum
Some Key Ideas

- Hypothesis Spaces and Oracles
Some Key Ideas

- Hypothesis Spaces and Oracles
- Approximation error and Estimation error
Some Key Ideas

- Hypothesis Spaces and Oracles
- Approximation error and Estimation error
- Complexity of Hypothesis Spaces (VC dimensions)
Some Key Ideas

- Hypothesis Spaces and Oracles
- Approximation error and Estimation error
- Complexity of Hypothesis Spaces (VC dimensions)
- Regularization and “Being Conservative”
Some Key Ideas

- Hypothesis Spaces and Oracles
- Approximation error and Estimation error
- Complexity of Hypothesis Spaces (VC dimensions)
- Regularization and “Being Conservative”
- Sufficient conditions for “Learning”
Some Key Ideas

- Hypothesis Spaces and Oracles
- Approximation error and Estimation error
- Complexity of Hypothesis Spaces (VC dimensions)
- Regularization and “Being Conservative”
- Sufficient conditions for “Learning”
- Empirical Estimation Theory and Falsifiability
Some Key Ideas

- Hypothesis Spaces and Oracles
- Approximation error and Estimation error
- Complexity of Hypothesis Spaces (VC dimensions)
- Regularization and “Being Conservative”
- Sufficient conditions for “Learning”
- Empirical Estimation Theory and Falsifiability

- Lets get back to real life ... to Boosting ...
Weak Learning

- A weak learner predicts “slightly better” than random.

The PAC setting (basics)
- Let \( \mathcal{X} \) be a instance space, \( c : \mathcal{X} \mapsto \{0, 1\} \) be a target concept, \( \mathcal{H} \) be a hypothesis space (\( h : \mathcal{X} \mapsto \{0, 1\} \)).
- Let \( Q \) be a fixed (but unknown) distribution on \( \mathcal{X} \).

An algorithm, after training on \((x_i, c(x_i)), [i]_m^m\), selects \( h \in \mathcal{H} \) such that

\[
P_{x \sim Q}[h(x) \neq c(x)] \leq \frac{1}{2} - \gamma
\]

- The algorithm is called a \( \gamma \)-weak learner.
- We assume the existence of such a learner.
The Boosting Model

- Boosting converts a weak learner to a strong learner
- Boosting proceeds in rounds
  - Booster constructs $D_t$ on $X$, the train set
  - Weak learner produces a hypothesis $h_t \in \mathcal{H}$ so that $P_{x \sim D_t}[h_t(x) \neq c(x)] \leq \frac{1}{2} - \gamma_t$
  - After $T$ rounds, the weak hypotheses $h_t, [t]_1^T$ are combined into a final hypothesis $h_{\text{final}}$
- We need procedures
  - for obtaining $D_t$ at each step
  - for combining the weak hypotheses
Adaboost

Input: Training set \((x_1, y_1), \ldots, (x_m, y_m)\)

Algorithm: Initialize \(D_1(i) = 1/m\)
For \(t = 1, \ldots, T\)
  - Train a weak learner using distribution \(D_t\)
  - Get weak hypothesis \(h_t\) with error \(\epsilon_t = \Pr_{x \sim D_t}[h_t(x) \neq y]\)
  - Choose \(\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)\)
  - Update
    \[
    D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}
    \]
where \(Z_t\) is the normalization factor

Output: \(h(x) = \text{sign} \left[ \sum_{t=1}^{T} \alpha_t h_t(x) \right] \)
The 0-1 training set loss with convex upper bounds: exponential loss and logistic loss
More on the Training Error

- The training error of the final classifier is bounded

\[ \frac{1}{m} |\{h(x_i) \neq y_i\}| \leq \frac{1}{m} \sum_{i=1}^{m} \exp(-y_i h(x_i)) = \prod_{t=1}^{T} Z_t \]

- Training error can be reduced most rapidly by choosing \( \alpha_t \) that minimizes

\[ Z_t = \sum_{i=1}^{n} D_t(i) \exp(-\alpha_t y_i h_t(x_i)) \]

- Adaboost chooses the optimal \( \alpha_t \)
  - Margin is (sort of) maximized

- Other boosting algorithms minimize other upper bounds
Boosting as Entropy Projection

For a general $\alpha$ we have

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

where $Z_t(\alpha) = \sum_{i=1}^m \exp(-\alpha y_i h_t(x_i))$. Then

$$\min_{E_D[yh(x)]=0} KL(D, D_t) = \max_{\alpha} (-\ln Z_t(\alpha)) .$$

Further

$$\arg\min_{E_D[yh(x)]=0} KL(D, D_t) = D_{t+1}^\alpha ,$$

which is the update Adaboost uses.

But what if labels are noisy?
Additive Models

An additive model

\[
F_T(x) = \sum_{t=1}^{T} w_t f_t(x).
\]

Fit the \( t \)-th component to minimize the “residual” error

Error is measured by \( C(yf(x)) \), an upper bound on \([y \neq f(x)]\)

For Adaboost,

\[
C(yf(x)) = \exp(-yf(x))
\]

For Logitboost,

\[
C(yf(x)) = \log(1 + \exp(-yf(x)))
\]
The 0-1 training set loss with convex upper bounds: exponential loss and logistic loss
Adaboost as Gradient Descent

- Margin Cost function $C(yh(x)) = \exp(-yh(x))$
- Current classifier $H_{t-1}(x)$, next weak learner $h_t(x)$
- Next classifier $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$
- The cost function to be minimized

$$\sum_{i=1}^{m} C(y_i[H_{t-1}(x_i) + \alpha_t h_t(x_i)])$$

- The optimum solution

$$\alpha_t = \frac{1}{2} \ln \left( \frac{\sum_{i:h_t(x_i)=y_i} D_t(i)}{\sum_{i:h_t(x_i)\neq y_i} D_t(i)} \right) = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)$$
Choose any appropriate margin cost function $C$

Goal is to minimize margin cost on the training set

Current classifier $H_{t-1}(x)$, next weak learner $h_t(x)$

Next classifier $H_t(x) = H_{t-1}(x) + \alpha_t h_t(x)$

Choose $\alpha_t$ to minimize

$$C(H_t(x) = C(H_{t-1}(x) + \alpha_t h_t(x))$$

“Anyboost” class of algorithms
Margin Bounds: Why Boosting works

- Test set error decreases even after train set error is 0
- No simple “bias-variance” explanation

Bounds on the error rate of the boosted classifier depends on
- the number of training examples
- the margin achieved on the train set
- the complexity of the weak learners

It does not depend on the number of classifiers combined
Margin Bound for Boosting

$\mathcal{H}$: Class of binary classifiers of VC-dimension $d_{\mathcal{H}}$

$$\text{co}(\mathcal{H}) = \left\{ h : h(x) = \sum_i \alpha_i h_i(x), \alpha_i \geq 0, \sum_i \alpha_i = 1 \right\}$$

With probability at least $(1 - \delta)$ over the draw of train set $S$ of size $m$, $\forall h \in \text{co}(\mathcal{H}), \theta > 0$ we have

$$\Pr[yh(x) \leq 0] \leq \Pr_S[yh(x) \leq \theta] + O \left( \frac{1}{\theta} \sqrt{\frac{d_{\mathcal{H}}}{m}} \right) + O \left( \sqrt{\frac{\log \frac{1}{\delta}}{m}} \right)$$
Maximizing the Margin

Do boosting algorithms maximize the margin?
- Minimize upper bound on training error
- Often get large margin in the process
- Do not explicitly maximize the margin

Explicit margin maximization
- Classical Boost: Minimize bound on $\Pr_S[yh(x) \leq 0]$
- Margin Boost: Minimize (bound on) $\Pr_S[yh(x) \leq \theta]$
Boosting: Wrap Up (for now)

- Strong learner from convex combinations of weak learners
- Choice of loss function is critical
  - Aggressive loss functions for clean data
  - Robust loss functions for noisy data
- Gradient descent in function space of additive models
- Generalization by implicit margin maximization
- Explicit maximum margin boosting algorithms possible
  - Is bound minimization a good idea?