The First-Order View of Boosting Methods: Computational Complexity and Connections to Regularization

Robert M. Freund Paul Grigas Rahul Mazumder
pgrigas@mit.edu

Massachusetts Institute of Technology

July 2013

Motivation

Boosting methods are learning methods for combining weak models into accurate and predictive models

- Add one new weak model per iteration
- The weight on each weak model is typically small

We consider boosting methods in two modeling contexts:

- Binary (confidence-rated) classification
- (Regularized/sparse) Linear regression

Boosting methods are typically tuned to perform implicit regularization

Motivation, continued

To properly balance the bias-variance tradeoff, a direct approach is to use models that solve explicitly defined regularized optimization problems

We therefore ask:

1. Are boosting methods solving any optimization problem(s)?
2. If so, what computational guarantees can we derive?
3. Can we adapt boosting methods to solve regularized problems?

Our Results:

- AdaBoost for binary classification is Mirror Descent to minimize the edge and, through dual iterates, maximize the margin
- Incremental Forward Stagewise Regression (FS_ε) is subgradient descent to minimize the correlation between the residuals and the predictors
- Computational complexity guarantees through both of these interpretations
- Conditional Gradient/Frank-Wolfe to minimize log-exponential loss/LASSO directly solves a regularized loss function minimization problem and is a very slight modification of AdaBoost/FS_ε

Mirror Descent for Minmax Optimization

Our problem of interest is:

$$(\text{P}): \quad \min_{x \in P} \; f(x)$$

- $P \subseteq \mathbb{R}^n$ is convex and closed
- $f(\cdot) : P \to \mathbb{R}$ is a (non-smooth) Lipschitz continuous convex function with Lipschitz value L_f

Mirror Descent for Minmax Optimization, continued

We assume that $f(\cdot)$ arises from minmax structure:

$$f(x) := \max_{\lambda \in Q} \ \phi(x, \lambda)$$

- $Q \subseteq \mathbb{R}^m$ is convex and compact
- $\phi(\cdot, \cdot) : P \times Q \to \mathbb{R}$ is convex-concave

Danskin's Theorem says that computing subgradients of $f(\cdot)$ depends on solving the maximization problem that defines $f(\cdot)$:

$$\partial f(x) = \text{conv}\left(\left\{ \nabla_x \phi(x, \tilde{\lambda}) : \tilde{\lambda} \in \arg\max_{\lambda \in Q} \phi(x, \lambda) \right\}\right)$$

Mirror Descent for Minmax Optimization, continued

When P is bounded, define $p(\lambda) := \min_{x \in P} \ \phi(x, \lambda)$. Then a dual problem is:

$$\text{(D):} \quad \max_{\lambda \in Q} \ p(\lambda)$$

MD uses a 1-strongly convex prox function $d(\cdot) : P \to \mathbb{R}$

- $d(\cdot)$ needs to be chosen such that solving $\min_{x \in P}\{c^T x + d(x)\}$ is easy for any $c \in \mathbb{R}^n$
- The Bregman distance associated with $d(\cdot)$ is:

$$D(x, y) := d(x) - d(y) - \nabla d(y)^T (x - y) \geq \frac{1}{2}\|x - y\|^2$$

Mirror Descent for Minmax Optimization, continued

Mirror Descent Method

Initialize at $x^0 \in P$, $\lambda^0 = 0, k = 0$

At iteration $k \geq 0$:

- Compute:

$$\tilde{\lambda}^k \leftarrow \arg\max_{\lambda \in Q} \; \phi(x^k, \lambda)$$

$$g^k \leftarrow \nabla_x \phi(x^k, \tilde{\lambda}^k)$$

- Choose $\alpha_k \geq 0$ and set:

$$x^{k+1} \leftarrow \arg\min_{x \in P} \left\{ \alpha_k (g^k)^T x + D(x, x^k) \right\}$$

$$\lambda^{k+1} \leftarrow \frac{\sum_{i=0}^k \alpha_i \tilde{\lambda}^i}{\sum_{i=0}^k \alpha_i}$$

(Note: the assignment of λ^{k+1} plays no role in the dynamics of the method)

Mirror Descent for Minmax Optimization, continued

Example: Subgradient Descent

- Take $P = \mathbb{R}^n$ and $d(x) = \frac{1}{2}\|x\|_2^2$
- Step (2.) of MD becomes

$$x^{k+1} \leftarrow x^k - \alpha_k g^k$$

Example: Multiplicative Weight Updates

- Take $P = \Delta_n := \{x : e^T x = 1, x \geq 0\}$ and let $d(x) = e(x) := \sum_{i=1}^n x_i \ln(x_i) + \ln(n)$
- Step (2.) of MD becomes

$$x_i^{k+1} \propto x_i^k \cdot \exp(-\alpha_k g_i^k) \quad \text{for all } i = 1, \ldots, n$$

Mirror Descent for Minmax Optimization, continued

Computational Guarantees for MD [Beck and Teboulle, Nesterov, Polyak, etc.]

For each $k \geq 0$ and for any $x \in P$, the following inequality holds:

$$\min_{i \in \{0,\ldots,k\}} f(x^i) - f(x) \leq \frac{D(x, x^0) + \frac{1}{2} L_f^2 \sum_{i=0}^{k} \alpha_i^2}{\sum_{i=0}^{k} \alpha_i}$$

Furthermore, if P is compact and $\bar{D} \geq \max_{x \in P} D(x, x^0)$, then for each $k \geq 0$ the following inequality holds:

$$\min_{i \in \{0,\ldots,k\}} f(x^i) - p(\lambda^{k+1}) \leq \frac{\bar{D} + \frac{1}{2} L_f^2 \sum_{i=0}^{k} \alpha_i^2}{\sum_{i=0}^{k} \alpha_i}$$

Boosting and AdaBoost

The set-up of the general boosting problem consists of:

- Data/training examples $(x_1, y_1), \ldots, (x_m, y_m)$ where each $x_i \in \mathcal{X}$ and each $y_i \in [-1, +1]$
- A set of base classifiers $\mathcal{H} = \{h_1, \ldots, h_n\}$ where each $h_j : \mathcal{X} \to [-1, +1]$
- Assume that $\mathcal{H}$ is closed under negation $(h_j \in \mathcal{H} \Rightarrow -h_j \in \mathcal{H})$

We would like to construct a nonnegative combination of weak classifiers

$$H_\lambda = \lambda_1 h_1 + \cdots + \lambda_n h_n$$

that performs significantly better than any individual classifier in $\mathcal{H}$.

Boosting and AdaBoost, continued

Recall

$$H_\lambda = \lambda_1 h_1 + \cdots + \lambda_n h_n$$

In the high-dimensional regime with $n \gg m \gg 0$, we desire:

- Good performance on the training data ($y_i H_\lambda(x_i) > 0$ for "most" $i = 1, \ldots, m$)
- Good predictive performance
- Shrinkage in the coefficients ($\|\lambda\|_1$ is small)
- Sparsity in the coefficients ($\|\lambda\|_0$ is small)

Some Loss Functions for Boosting

Define the feature matrix $A \in \mathbb{R}^{m \times n}$ by $A_{ij} = y_i h_j(x_i)$

Two loss functions are often considered in this context:

- The margin
  $p(\lambda) := \min_{i \in \{1,\dots,m\}} y_i H_\lambda(x_i) = \min_{i \in \{1,\dots,m\}} (A\lambda)_i = \min_{w \in \Delta_m} w^T A\lambda$
- The exponential loss $L_{\exp}(\lambda) := \frac{1}{m} \sum_{i=1}^{m} \exp\left(-(A\lambda)_i\right)$
- ($\equiv$ the log-exponential loss $L(\lambda) := \log(L_{\exp}(\lambda))$)

It is known that a high margin implies good generalization properties [Schapire 97]. On the other hand, the exponential loss upper bounds the empirical probability of misclassification.

The Margin Maximization Problem

The problem of maximizing the margin over all normalized classifiers is:

$$\text{(D):} \quad \rho^* = \max_{\lambda \in \Delta_n} \; p(\lambda)$$

And its dual is the problem of minimizing the edge:

$$\text{(P):} \quad \min_{w \in \Delta_m} \; f(w) := \max_{\lambda \in \Delta_n} w^T A \lambda$$

Suppose that we have access to a weak learner $\mathcal{W}(\cdot)$ that, for any distribution w on the examples ($w \in \Delta_m$), returns the base classifier h_{j^*} in $\mathcal{H}$ that does best on the weighted example determined by w:

$$j^* \in \arg\max_{j=1,\ldots,n} w^T A_j = \arg\max_{\lambda \in \Delta_n} w^T A \lambda$$

AdaBoost Algorithm Description

AdaBoost Algorithm

Initialize at $w^0 = (1/m, \ldots, 1/m), H_0 = 0, k = 0$

At iteration $k \geq 0$:

- Compute $j_k \in \mathcal{W}(w^k)$
- Choose $\alpha_k \geq 0$ and set:
  $H_{k+1} \leftarrow H_k + \alpha_k h_{j_k}$
  $w_i^{k+1} \leftarrow w_i^k \exp(-\alpha_k y_i h_{j_k}(x_i)) \quad i = 1, \ldots, m$, and
  re-normalize w^{k+1} so that $e^T w^{k+1} = 1$

AdaBoost has the following sparsity/regularization properties:

$$\|\lambda^k\|_1 \leq \sum_{i=0}^{k-1} \alpha_i \quad \text{and} \quad \|\lambda^k\|_0 \leq k \ .$$

Optimization Perspectives on AdaBoost

What has been known about AdaBoost in the context of optimization:

- AdaBoost has been interpreted as a coordinate descent method to minimize the exponential loss [Mason et al., Mukherjee et al., etc.]
- A related method, the Hedge Algorithm, has been interpreted as dual averaging [Baes and Bürgisser]
- Rudin et al. in fact show that AdaBoost can fail to maximize the margin, but this is under the particular step-size $\alpha_k := \frac{1}{2} \ln\left(\frac{1+r_k}{1-r_k}\right)$
- Lots of other work as well...

AdaBoost is Mirror Descent

Define a sequence of normalized classifiers from AdaBoost by:

$$\bar{H}_0 := 0 \ , \ \ \bar{H}_k := \frac{H_k}{\sum_{i=0}^{k-1} \alpha_i} \ , \ k \geq 1 \ .$$

Equivalence Theorem

The sequence of weight vectors $\{w^k\}$ in AdaBoost arise as primal variables in MD applied to the minimum edge problem

$$\text{(P):} \quad \min_{w \in \Delta_m} \ f(w)$$

using the entropy prox function. Moreover, the sequence of dual variables $\{\lambda^k\}$ in MD define the normalized classifiers in AdaBoost, i.e.,

$$\bar{H}_k = \sum_{j=1}^{n} \lambda_i^k h_j \ .$$

Interpretations of the (P)/(D) Objective Functions in MD

Note that $p(\lambda^k)$ is the margin of the normalized classifier $\bar{H}_k$

Let $\hat{\lambda}^k$ be the coefficient vector of the un-normalized classifier H_k

Lemma

For every iteration $k \geq 0$ of AdaBoost, the edge $f(w^k)$ and the un-normalized classifier H_k with coefficient vector $\hat{\lambda}^k$ satisfy:

$$f(w^k) = \|\nabla L(\hat{\lambda}^k)\|_\infty \ .$$

Complexity of AdaBoost: General Case

Complexity of AdaBoost

For all $k \geq 1$, the sequence of normalized and un-normalized classifiers produced by AdaBoost satisfy:

$$\min_{i \in \{0,\ldots,k-1\}} \|\nabla L(\hat{\lambda}^i)\|_\infty - p(\lambda^k) \leq \frac{\ln(m) + \frac{1}{2}\sum_{i=0}^{k-1} \alpha_i^2}{\sum_{i=0}^{k-1} \alpha_i} .$$

If we decide a priori to run AdaBoost for $k \geq 1$ iterations and use a constant step-size $\alpha_i := \sqrt{\frac{2\ln(m)}{k}}$ for all $i = 0, \ldots, k-1$, then we have:

$$\min_{i \in \{0,\ldots,k-1\}} \|\nabla L(\hat{\lambda}^i)\|_\infty - p(\lambda^k) \leq \sqrt{\frac{2\ln(m)}{k}} .$$

Recall that $\rho^* = \max_{\lambda \in \Delta_n} p(\lambda)$ is the maximum margin and $\rho^* \geq 0$

Complexity of AdaBoost: Separable Case

If $\rho^* > 0$, then the data is separable and the margin is informative

Complexity of AdaBoost: Separable Case

For all $k \geq 1$, the sequence of normalized classifiers produced by AdaBoost satisfy:

$$\rho^* - p(\lambda^k) \leq \frac{\ln(m) + \frac{1}{2}\sum_{i=0}^{k-1} \alpha_i^2}{\sum_{i=0}^{k-1} \alpha_i} \ .$$

If we decide a priori to run AdaBoost for $k \geq 1$ iterations and use a constant step-size $\alpha_i := \sqrt{\frac{2\ln(m)}{k}}$ for all $i = 0, \ldots, k-1$, then we have:

$$\rho^* - p(\lambda^k) \leq \sqrt{\frac{2\ln(m)}{k}} \ .$$

Complexity of AdaBoost: Non-separable Case

If $\rho^* = 0$, then the data is not separable and the margin is no longer informative

Complexity of AdaBoost: Non-separable Case

If $\rho^* = 0$, then for all $k \geq 1$, the sequence of normalized classifiers produced by AdaBoost satisfy:

$$\min_{i \in \{0,\ldots,k-1\}} \|\nabla L(\hat{\lambda}^i)\|_\infty \leq \frac{\ln(m) + \frac{1}{2}\sum_{i=0}^{k-1} \alpha_i^2}{\sum_{i=0}^{k-1} \alpha_i} \ .$$

If we decide a priori to run AdaBoost for $k \geq 1$ iterations and use a constant step-size $\alpha_i := \sqrt{\frac{2\ln(m)}{k}}$ for all $i = 0, \ldots, k-1$, then we have:

$$\min_{i \in \{0,\ldots,k-1\}} \|\nabla L(\hat{\lambda}^i)\|_\infty \leq \sqrt{\frac{2\ln(m)}{k}} \ .$$

Conditional Gradient Method for Regularized Log-Exponential Loss Minimization

In the non-separable case, AdaBoost has guarantees for $\|\nabla L(\lambda)\|_\infty$

What about guarantees for $L(\lambda) := \log\left(\frac{1}{m}\sum_{i=1}^m \exp\left(-(A\lambda)_i\right)\right)$?

Let us consider applying the conditional gradient method to solve:

$$
\begin{aligned}
L^*_\delta = \quad & \min_{\lambda} \quad L(\lambda) \\
& \text{s.t.} \quad \|\lambda\|_1 \leq \delta \\
& \qquad\quad \lambda \geq 0
\end{aligned}
$$

Structure of Conditional Gradient Method Updates

At iteration k, the conditional gradient method needs to:

- Compute $\nabla L(\lambda^k)$
- Solve $\min_{\lambda : \|\lambda\|_1 \leq \delta, \lambda \geq 0} \nabla L(\lambda^k)^T \lambda$
- Update λ^{k+1}

We could compute $\nabla L(\lambda^k)$ directly...

Instead, we note that $L(\lambda) = \max_{w \in \Delta_m} \left\{ -w^T A\lambda - e(w) \right\}$ and define a weight vector w^k to be the optimal solution to the above problem:

$$w_i^k = \frac{\exp(-(A\lambda^k)_i)}{\sum_{l=1}^m \exp(-(A\lambda^k)_l)} \quad i = 1, \ldots, m$$

Then, $\nabla L(\lambda^k) = -A^T w^k$

Structure of Conditional Gradient Method Updates, continued

Solving the subproblem $\min_{\lambda:\|\lambda\|_1 \le \delta, \lambda \ge 0} \nabla L(\lambda^k)^T \lambda$ is equivalent to calling the weak learner, that is:

$$j_k \in \mathcal{W}(w^k) \iff \delta e_{j_k} \in \min_{\lambda:\|\lambda\|_1 \le \delta, \lambda \ge 0} -(w^k)^T A \lambda$$

The update for λ^{k+1} is standard

Also easy to show a simple update rule for w^{k+1}

Complete Conditional Gradient Algorithm Description

CG-Boost Algorithm

Initialize at $\lambda^0 = 0$, $w^0 = (1/n, \ldots, 1/n)$, $k = 0$

At iteration $k \geq 0$:

- Compute:

$$j_k \in \mathcal{W}(w^k)$$

- Choose $\bar{\alpha}_k \in [0, 1]$ and set:

$$\lambda_{j_k}^{k+1} \leftarrow (1 - \bar{\alpha}_k)\lambda_{j_k}^k + \bar{\alpha}_k \delta$$
$$\lambda_j^{k+1} \leftarrow (1 - \bar{\alpha}_k)\lambda_{j_k}^k \quad , j \neq j_k$$
$$w_i^{k+1} \leftarrow (w_i^k)^{1-\bar{\alpha}_k} \exp(-\bar{\alpha}_k \delta y_i h_{j_k}(x_i)) \quad i = 1, \ldots, m, \text{ and}$$

re-normalize w^{k+1} so that $e^T w^{k+1} = 1$

Note that CG-Boost has the sparsity property that $\|\lambda^k\|_0 \leq k$

Complexity of CG-Boost

Complexity of CG-Boost

With either the fixed step-size rule $\bar{\alpha}_k := \frac{2}{k+2}$ or a line-search to determine $\bar{\alpha}_k$, then then for all $k \geq 1$ we have the following inequalities:

$$L(\lambda^k) - L^*_\delta \leq \frac{8\delta^2}{k+3}$$
$$\rho^* - p(\bar{\lambda}^k) \leq \frac{8\delta}{k+3} + \frac{\ln(m)}{\delta}$$

where $\bar{\lambda}^k$ is the normalization of λ^k, i.e., $\bar{\lambda}^k := \frac{\lambda^k}{\delta}$

Bounds can also be obtained for the constant step-size $\bar{\alpha}_k := \bar{\alpha}$

Incremental Forward Stagewise Regression

Consider the linear regression model $\mathbf{y} = \mathbf{X}\beta + \mathbf{e}$

- $\mathbf{y} \in \mathbb{R}^n$ is given response data
- $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the given model matrix
- $\beta \in \mathbb{R}^p$ are the coefficents
- $\mathbf{e} \in \mathbb{R}^n$ is noise

In the high-dimensional regime with $p \gg n \gg 0$, we desire:

- Good performance on the training data (residuals $r := \mathbf{y} - \mathbf{X}\beta$ are small)
- Good out of sample predictive performance
- Shrinkage in the coefficients ($\|\beta\|_1$ is small)
- Sparsity in the coefficients ($\|\beta\|_0$ is small)

Incremental Forward Stagewise Regression, continued

In the high-dimensional regime, the LASSO solution often performs very well:

$$\begin{aligned} \min_{\beta} \quad & L(\beta) := \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \\ \text{s.t.} \quad & \|\beta\|_1 \leq \delta \end{aligned}$$

Another loss function measures the correlation between the residuals and the predictors:

$$\min_{r \in P_{\text{res}}} \quad f(r) := \|\mathbf{X}^T r\|_\infty$$

where P_{res} is the set of residuals.

We also have $f(\mathbf{y} - \mathbf{X}\beta) = \|\nabla L(\beta)\|_\infty$

FS_ε as a Boosting Algorithm

In the setting of boosting:

- Each independent variable x_j represents the j^{th} weak model
- β is the vector of weak model coefficients

The boosting method FS_ε adds, at iteration k, the predictor x_{j_k} most correlated with the current residuals r^k

$$j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(r^k)^T \mathbf{X}_j|$$

FS$_\varepsilon$ Algorithm Description

FS$_\varepsilon$ Algorithm

Initialize at $r^0 = \mathbf{y}$, $\beta^0 = 0, k = 0$

At iteration $k \geq 0$:

- Compute:

$$j_k \in \underset{j \in \{1,\ldots,p\}}{\arg\max} |(r^k)^T \mathbf{X}_j|$$

- Set:

$$\begin{aligned} r^{k+1} &\leftarrow r^k - \varepsilon \ \mathrm{sgn}((r^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} \\ \beta^{k+1}_{j_k} &\leftarrow \beta^k_{j_k} + \varepsilon \ \mathrm{sgn}((r^k)^T \mathbf{X}_{j_k}) \\ \beta^{k+1}_{j} &\leftarrow \beta^k_{j} \ , j \neq j_k \end{aligned}$$

FS_ε is Subgradient Descent

FS_ε is known to have the following regularization/sparsity properties:

$$\|\beta^k\|_1 \le k\varepsilon \quad \text{and} \quad \|\beta^k\|_0 \le k \ .$$

What loss function criterion might FS_ε optimize?

FS_ε Equivalence Theorem

The FS_ε algorithm is an instance of the subgradient descent method to solve

$$\min_{r \in P_{\text{res}}} \quad f(r) := \|\mathbf{X}^T r\|_\infty$$

initialized at $r^0 = \mathbf{y}$ and with a constant step-size of ε at each iteration.

Complexity of FS_ε

Complexity of FS_ε

With the constant shrinkage factor ε, for any $k \geq 0$ it holds that:

$$\min_{i \in \{0,\dots,k\}} \|\mathbf{X}^T r^i\|_\infty \leq \frac{\|\mathbf{X}\beta_{LS}\|_2^2}{2\varepsilon(k+1)} + \frac{\varepsilon \|\mathbf{X}\|_{1,2}^2}{2} .$$

If we set $\varepsilon := \frac{\|\mathbf{X}\beta_{LS}\|_2}{\|\mathbf{X}\|_{1,2}\sqrt{k+1}}$ or $\varepsilon_k := \frac{|(r^k)^T \mathbf{X}_{j_k}|}{\|\mathbf{X}_{j_k}\|_2^2}$ then

$$\min_{i \in \{0,\dots,k\}} \|\mathbf{X}^T r^i\|_\infty \leq \frac{\|\mathbf{X}\|_{1,2} \|\mathbf{X}\beta_{LS}\|_2}{\sqrt{k+1}} .$$

where β_{LS} is the least-squares solution so that $\|\mathbf{X}\beta_{LS}\|_2 \leq \|\mathbf{y}\|_2$.

Complexity of FS_ε, continued

The number of iterations k and the shrinkage factor ε should be chosen to balance:

- Guarantees for $\|\mathbf{X}^T r\|_\infty$ through the previous theorem
- Sparsity/regularization guarantees: $\|\beta\|_1 \leq k\varepsilon$ and $\|\beta\|_0 \leq k$

What about guarantees for the least-squares loss $\frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$?

Frank-Wolfe on the LASSO

Recall the LASSO:

$$
\begin{aligned}
L_\delta^* = \quad \min_{\beta} \quad & L(\beta) := \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \\
\text{s.t.} \quad & \|\beta\|_1 \leq \delta
\end{aligned}
$$

FS_ε guarantees that $\|\beta^k\|_0 \leq k$. A method with similar sparsity properties is Frank-Wolfe on the LASSO

At iteration k, Frank-Wolfe needs to:

- Compute $\nabla L(\beta^k) = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta^k) = -(r^k)^T\mathbf{X}$
- Solve $\min_{\beta:\|\beta\|_1 \leq \delta} \nabla L(\beta^k)^T\beta$
- Update β^{k+1}

Solving the Linear Optimization Subproblem

Extreme points of $\{\beta : \|\beta\|_1 \leq \delta\}$ are $\{\pm\delta e_j : j = 1, \ldots, p\}$ so

$$-\delta \text{sgn}(-(r^k)^T \mathbf{X}_{j^*}) e_{j^*} \in \underset{\beta:\|\beta\|_1 \leq \delta}{\arg\min} \nabla L(\beta^k)^T \beta \iff j^* \in \underset{j \in \{1,\ldots,p\}}{\arg\max} |(r^k)^T \mathbf{X}_j|$$

This is the same subproblem that FS_ε solves (find the predictor that maximizes correlation with the residuals)

Complete Frank-Wolfe Algorithm Description

FW-LASSO Algorithm

Initialize at $\beta^0 = 0$, $k = 0$

At iteration $k \geq 0$:

- Compute:

$$r^k \leftarrow \mathbf{y} - \mathbf{X}\beta^k$$
$$j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(r^k)^T \mathbf{X}_j|$$

- Choose $\bar{\alpha}_k \in [0, 1]$ and set:

$$\beta_{j_k}^{k+1} \leftarrow (1 - \bar{\alpha}_k)\beta_{j_k}^k + \bar{\alpha}_k \delta \ \mathrm{sgn}((r^k)^T \mathbf{X}_{j_k})$$
$$\beta_j^{k+1} \leftarrow (1 - \bar{\alpha}_k)\beta_j^k \ , j \neq j_k$$

FW-LASSO is structurally very similar to FS_ε

Properties of FW-LASSO

Note that FW-LASSO shares similar sparsity/regularization properties as FS_ε:

- $\|\beta^k\|_0 \leq k$
- $\|\beta^k\|_1 \leq \delta$

For a fixed step-size $\bar{\alpha}_k := \frac{\varepsilon}{\delta+\varepsilon}$, observe the update for β^{k+1} can be rearranged to:

$$\beta^{k+1} \leftarrow \frac{\delta}{\varepsilon + \delta}\left[\beta^k + \varepsilon \text{sgn}((r^k)^T \mathbf{X}_{j_k}) e_{j_k}\right]$$

which is equivalent to the FS_ε update modulo a multiplicative factor which keeps the coefficient profile within $\{\beta : \|\beta\|_1 \leq \delta\}$

Complexity of FW-LASSO

Complexity of FW-LASSO

With either the fixed step-size rule $\bar{\alpha}_k := \frac{2}{k+2}$ or a line-search to determine $\bar{\alpha}_k$, then after k iterations there exists an $i \in \{1, \ldots, k\}$ such that the following two inequalities both hold:

$$L(\beta^i) - L^*_\delta \le \frac{17.4 \|\mathbf{X}\|^2_{1,2} \delta^2}{k}$$

$$\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta^i)\|_\infty \le \frac{1}{2\delta}\|\mathbf{X}\beta_{LS}\|_2^2 + \frac{17.4\|\mathbf{X}\|^2_{1,2}\delta}{k} .$$

Bounds can also be obtained for the constant step-size $\bar{\alpha}_k := \frac{\varepsilon}{\delta+\varepsilon}$

Conclusions

We have shown that:

- AdaBoost is equivalent to Mirror Descent with an entropy prox function $\Rightarrow$ complexity guarantees for the margin $p(\lambda)$ in the case of separable data and for $\|\nabla L(\lambda)\|_\infty$ in the case of non-separable data.
- FS_ε is equivalent to subgradient descent $\Rightarrow$ complexity guarantees for $\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)\|_\infty$

The above two results also extend to the functional boosting setting, as long as the loss function is convex and globally smooth

Conditional gradient/Frank-Wolfe on regularized log-exponential loss minimization/LASSO are simple modifications to AdaBoost/FS_ε