Boosting Methods: Implicit Combinatorial Optimization via First-Order Convex Optimization

Robert M. Freund Paul Grigas Rahul Mazumder

rfreund@mit.edu

M.I.T.

ADGO October 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

Review of Subgradient Descent and Frank-Wolfe Methods

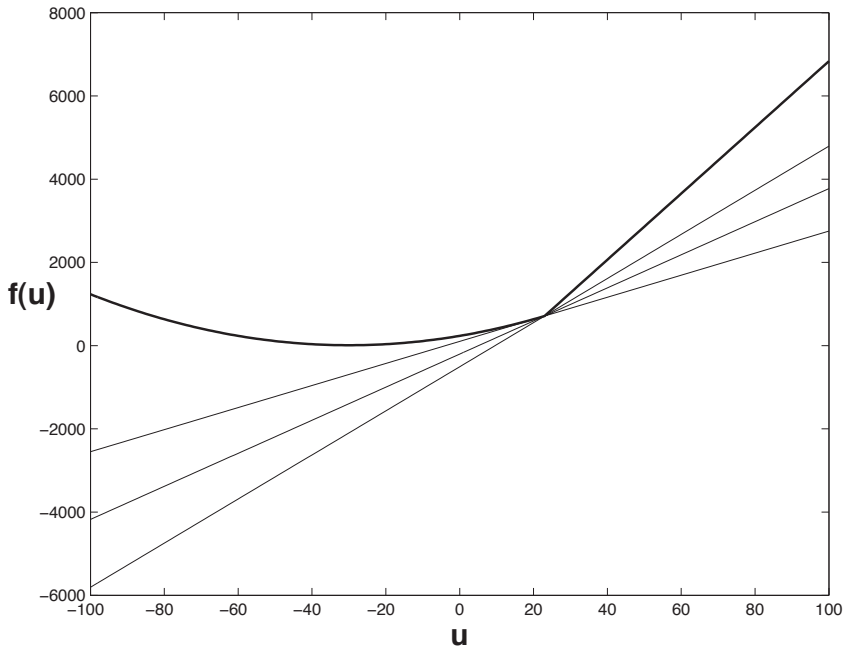
1. Subgradient Descent method
2. Frank-Wolfe method (also known as Conditional Gradient method)

Subgradient Descent

Our problem of interest is:

$$\begin{array}{lll} f^* & := & \min\limits_{x} \quad f(x) \\ & & \text{s.t.} \quad x \in \mathbb{R}^n \end{array}$$

where $f(x)$ is convex but not differentiable. Then $f(x)$ has subgradients.



Subgradient Descent, continued

$$\begin{array}{rcll} f^* & := & \min\limits_{x} & f(x) \\ & & \text{s.t.} & x \in \mathbb{R}^n \end{array}$$

$f(\cdot)$ is a (non-smooth) Lipschitz continuous convex function with Lipschitz value L_f:

$$|f(x) - f(y)| \leq L_f \|x - y\| \qquad \text{for any } x, y$$

$\|\cdot\|$ is prescribed norm on $\mathbb{R}^n$

Subgradient Descent, continued

$$\begin{array}{rcll} f^* & := & \min\limits_{x} & f(x) \\ & & \text{s.t.} & x \in \mathbb{R}^n \end{array}$$

Subgradient Descent method for minimizing $f(x)$ on $\mathbb{R}^n$

Initialize at $x_1 \in \mathbb{R}^n$, $k \leftarrow 1$.

At iteration k :

1. Compute a subgradient g_k of $f(x_k)$.
2. Choose step-size α_k .
3. Set $x_{k+1} \leftarrow x_k - \alpha_k g_k$.

Computational Guarantees for SD

Computational Guarantees for Subgradient Descent

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

$$\min_{i \in \{0,\ldots,k\}} f(x^i) - f(x) \leq \frac{\|x - x^0\|_2^2 + L_f^2 \sum_{i=0}^k \alpha_i^2}{2 \sum_{i=0}^k \alpha_i}$$

Frank-Wolfe Method (Conditional Gradient method)

Here the problem of interest is:

$$\begin{array}{rcll} f^* & := & \min\limits_{x} & f(x) \\ & & \text{s.t.} & x \in P \end{array}$$

- P is compact and convex
- $f(x)$ is differentiable and $\nabla f(\cdot)$ is Lipschitz on P:

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L_\nabla \|x - y\| \quad \text{for all } x, y \in P$$

- it is "very easy" to do linear optimization on P for any c :

$$\tilde{x} \leftarrow \arg\min_{x \in P} \{c^T x\}$$

Frank-Wolfe Method, continued

$$\begin{array}{rcll} f^* & := & \min\limits_{x} & f(x) \\ & & \text{s.t.} & x \in P \end{array}$$

Frank-Wolfe Method for minimizing $f(x)$ on P

Initialize at $x_0 \in P$, $k \leftarrow 0$.

At iteration k :

1. Compute $\nabla f(x_k)$.
2. Compute $\tilde{x}_k \leftarrow \arg\min\limits_{x \in P} \{\nabla f(x_k)^T x\}$.
3. Set $x_{k+1} \leftarrow x_k + \bar{\alpha}_k(\tilde{x}_k - x_k)$, where $\bar{\alpha}_k \in [0, 1]$.

Computational Guarantees for Frank-Wolfe Method

Here is one (simplified) computational guarantee:

A Computational Guarantee for Frank-Wolfe Method

If the step-size sequence $\{\bar{\alpha}_k\}$ is chosen as $\bar{\alpha}_k = \frac{2}{k+2}$, $k \geq 0$, then for all $k \geq 1$ it holds that:

$$f(x_k) - f^* \leq \frac{C}{k+3}$$

where $C = 2 \cdot L_\nabla \cdot \text{diam}(P)^2$.

Binary Classification

Binary Classification

Binary Classification

The set-up of the general binary classification 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 $\mathcal{H}$.

Binary Classification Feature Matrix

Define the feature matrix $A \in \mathbb{R}^{m \times n}$ by $A_{ij} = y_i h_j(x_i)$
We seek $\lambda \geq 0$ for which:

$$A\lambda > 0 \quad \text{or perhaps} \quad A\lambda >\approx 0$$

In application/academic context:

- m is large-scale
- n is huge-scale, too large for many computational tasks
- we wish only to work with very sparse λ, namely $\|\lambda\|_0$ is small
- we have access to a weak learner $\mathcal{W}(\cdot)$ that, for any distribution w on the examples ($w \geq 0,\ e^T w = 1$), returns the base classifier $j^* \in \{1, \ldots, n\}$ that does best on the weighted example determined by w:

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

Binary Classification Aspirations

We seek $\lambda \geq 0$ for which:

$$A\lambda > 0 \quad \text{or perhaps} \quad A\lambda >\approx 0$$

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

- Good predictive performance (on out-of-sample examples)
- Good performance on the training data ($A_i\lambda > 0$ for "most" $i = 1, \ldots, m$)
- Sparsity of the coefficients ($\|\lambda\|_0$ is small)
- Regularization of the coefficients ($\|\lambda\|_1$ is small)

Two Objective Functions for Boosting

We seek $\lambda \geq 0$ for which:

$$A\lambda > 0 \quad \text{or perhaps} \quad A\lambda >\approx 0$$

Two objective functions are often considered in this context:

- when the data are separable, maximize the margin: $p(\lambda) := \min_{i \in \{1,\ldots,m\}} (A\lambda)_i$
- when the data are non-separable, minimize 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.

Margin Maximization Problem

The margin is $p(\lambda) := \min_{i \in \{1,\ldots,m\}} (A\lambda)_i$

$p(\lambda)$ is positively homogeneous, so we normalize the variables λ

Let $\Delta_n := \{\lambda \in \mathbb{R}^n : e^T\lambda = 1,\ \lambda \geq 0\}$

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

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

Recall that we have access to a weak learner $\mathcal{W}(\cdot)$ that, for any distribution w on the examples $(w \geq 0,\ e^T w = 1)$, returns the base classifier $j^* \in \{1, \ldots, n\}$ that does best on the weighted example determined by w:

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

AdaBoost Algorithm

AdaBoost Algorithm

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

At iteration $k \geq 0$:

- Compute $j_k \in \mathcal{W}(w^k)$
- Choose step-size $\alpha_k \geq 0$ and set:
  $\lambda^{k+1} \leftarrow \lambda^k + \alpha_k e^{j_k}$ $\quad \bar{\lambda}^{k+1} \leftarrow \frac{\lambda^{k+1}}{e^T \lambda^{k+1}}$
  $w_i^{k+1} \leftarrow w_i^k \exp(-\alpha_k A_{ij_k})$ $\;\; i = 1, \ldots, m$, and re-normalize w^{k+1} so that $e^T w^{k+1} = 1$

AdaBoost has the following sparsity/regularization properties:

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

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 popular "optimized" step-size $\alpha_k := \frac{1}{2} \ln\left(\frac{1+r_k}{1-r_k}\right)$
- Lots of other work as well...

Complexity of AdaBoost: General Case

Recall $L(\lambda) := \log\left(\frac{1}{m}\sum_{i=1}^{m}\exp\left(-(A\lambda)_i\right)\right)$ and ρ^* is the maximum (normalized) margin

Complexity of AdaBoost

For all $k \geq 1$, the sequence of variables λ^k and $\bar{\lambda}^k$ produced by AdaBoost satisfy:

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

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:

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

Complexity of AdaBoost: Separable Case

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

Complexity of AdaBoost: Separable Case

For all $k \geq 1$, the sequence $\bar{\lambda}^k$ produced by AdaBoost satisfies:

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

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:

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

Complexity of AdaBoost: Non-separable Case

If the data is not separable, then $\rho^* = 0$ and the log-exponential loss function is informative

Complexity of AdaBoost: Non-separable Case

If the data is not separable, then for all $k \geq 1$, the sequence λ^k produced by AdaBoost satisfies:

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

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:

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

What drives these results?

- observation that AdaBoost corresponds to the Mirror Descent method [N-Y, B-M-T, B-T] of non-differentiable convex optimization, using the "entropy prox function" applied to the dual of the maximum margin problem
- application of Mirror Descent convergence theory for various step-size sequences
- development of some new algorithmic properties of the Mirror Descent method in general

What about Regularized Log-Exponential Loss Minimization?

Log-exponential loss function is:

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

In the non-separable case, AdaBoost guarantees $\|\nabla L(\lambda^i)\|_\infty \searrow 0$

Let us consider directly tackling $L(\lambda)$ in the regularized setting:

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

FW Method for Log-Exponential Loss Minimization

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

Consider using the FW method. At iteration k the 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 cannot necessarily do first two steps .... But we do have access to a weak learner $\mathcal{W}(\cdot)$: for $w \in \Delta_m$, $\mathcal{W}(w)$ computes:

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

Log-Exponential Loss Minimization, continued

Instead, work with log-exponential loss function in conjugate (adjoint) form. Let $e(w) := \sum_{i=1}^{m} w_i \ln(w_i) + \ln(m)$ be the entropy function.

Proposition

- $L(\lambda^k) = \max_{w \in \Delta_m} \left\{ -w^T A \lambda^k - e(w) \right\}$
- $\nabla L(\lambda^k) = -A^T w^k$ where

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

- Weak learner can be used to solve the linear optimization subproblem using w^k:

$$j_k \in \mathcal{W}(w^k) \iff \delta e_{j_k} \in \arg \min_{\lambda : \|\lambda\|_1 \leq \delta, \lambda \geq 0} \nabla L(\lambda^k)^T \lambda$$

FW-Boost Algorithm Description

FW-Boost Algorithm

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

At iteration $k \geq 0$:

- Compute:
$$j_k \in \mathcal{W}(w^k)$$
- Choose $\bar{\alpha}_k \in [0, 1]$ and set:
$$\lambda^{k+1} \leftarrow (1 - \bar{\alpha}_k)\lambda^k_{j_k} + \bar{\alpha}_k \delta e^{j_k}$$
$$w_i^{k+1} \leftarrow (w_i^k)^{1-\bar{\alpha}_k} \exp(-\bar{\alpha}_k \delta A_{ij_k}) \quad i = 1, \ldots, m$$
Re-normalize w^{k+1} so that $e^T w^{k+1} = 1$

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

Complexity of FW-Boost

Complexity of FW-Boost

With the step-size rule $\bar{\alpha}_k := \frac{2}{k+2}$, for all $k \geq 1$ the following inequalities hold:

(i) $L(\lambda^k) - L_\delta^* \leq \frac{8\delta^2}{k+3}$

(ii) $p(\bar{\lambda}^k) \geq \rho^* - \left(\frac{8\delta}{k+3} + \frac{\ln(m)}{\delta}\right)$

(iii) $\|\lambda^k\|_1 \leq \delta$

(iv) $\|\lambda^k\|_0 \leq k$

Binary Classification Boosting Summary

- AdaBoost is interpretable as an instance of the Mirror Descent method applied to the dual of the maximum margin problem
- Computational complexity guarantees for AdaBoost for maximizing the margin, minimizing the log-exponential loss in AdaBoost
- New properties of the Mirror Descent method
- Frank-Wolfe method to minimize the log-exponential loss is seen to be a slight modification of AdaBoost, with associated computational complexity guarantees in the separable and non-separable cases

Linear Regression

Linear Regression

Linear 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 coefficients
- $\mathbf{e} \in \mathbb{R}^n$ is noise

Linear Regression and Boosting

Linear regression model:

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e}$$

In the setting of boosting:

- the column $\mathbf{X}_j$ represents the data of the j^{th} weak model
- β_j is the regression coefficient for the j^{th} weak model

Linear Regression Aspirations

Linear regression model:

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e}$$

In the high-dimensional regime with $p \gg 0$, $n \gg 0$ and often $p > n$, we seek:

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

Traditional Least-Squares Regression

$$\text{LS}: \quad \min_\beta \quad L(\beta) := \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

$L(\beta) := \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$ is the least-squares loss

Let $r := \mathbf{y} - \mathbf{X}\beta$, then $L(\beta) = \frac{1}{2}\|r\|_2^2$

$L^* := \min_\beta L(\beta)$

β_{LS} is any solution of LS

L_1-Regularization and LASSO

L_1-Penalized Least-Squares optimization problem:

$$\text{LASSO}^\tau: \quad \min_\beta \quad \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \tau\|\beta\|_1$$

LASSO stands for Least Absolute Shrinkage and Selection Operator

Let β_τ^* be an optimal solution of LASSO^τ

$\|\beta_\tau^*\|_0 \searrow$ as $\tau \nearrow$

There is a well-developed theory of sparse L_1-regularized solutions

Constraint Version of LASSO

$$\text{LASSO}^{\tau}: \quad \min_{\beta} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \tau\|\beta\|_1$$

$$\text{LASSO}_{\delta}: \quad \min_{\beta} \quad L(\beta) := \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

$$\text{s.t.} \qquad \|\beta\|_1 \leq \delta$$

Both LASSO^{τ} for $\tau \in [0, \infty)$ and LASSO_{δ} for $\delta \in [0, \infty)$ generate the same solution path, which is simply called the LASSO path

Incremental Forward Stagewise Regression Algorithm (FS_ε)

Incremental Forward Stagewise Regression (FS_ε) is a simple boosting algorithm:

Start with $\beta^0 \leftarrow 0$, and hence $r^0 \leftarrow \mathbf{y}$.

Given β^k and $r^k := \mathbf{y} - \mathbf{X}\beta^k$, determine the weak model $\mathbf{X}_j$ most correlated with the current residuals r^k:

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

Adjust $\beta^k_{j_k}$ by $\pm\varepsilon$ depending on $sgn((r^k)^T \mathbf{X}_j)$

FS_ε Algorithm

FS_ε Algorithm

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

At iteration $k \geq 0$:

- Compute:

$$r^k \leftarrow \mathbf{y} - \mathbf{X}\beta^k$$

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

- Set:

$$\beta^{k+1} \leftarrow \beta^k + \varepsilon \, \mathrm{sgn}((r^k)^T \mathbf{X}_{j_k}) e^{j_k}$$

FS_ε has the following regularization/sparsity properties:

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

Complexity of FS_ε

Complexity of FS_ε

With the constant shrinkage factor ε, for any $k \geq 0$ there exists $i \leq k$ for which:

(i) $L(\beta^i) - L^* \leq \frac{p}{2(\lambda_{\min}(\mathbf{X}))^2} \left[\frac{\|\mathbf{X}\beta_{LS}\|_2^2}{\varepsilon(k+1)} + \varepsilon \|\mathbf{X}\|_{1,2}^2 \right]^2$

(ii) there exists a solution β_{LS} for which
$\|\beta^i - \beta_{LS}\|_2 \leq \frac{\sqrt{p}}{(\lambda_{\min}(\mathbf{X}))^2} \left[\frac{\|\mathbf{X}\beta_{LS}\|_2^2}{\varepsilon(k+1)} + \varepsilon \|\mathbf{X}\|_{1,2}^2 \right]$

(iii) $\|\beta^i\|_1 \leq k\varepsilon$

(iv) $\|\beta^i\|_0 \leq k$

(v) $\|\nabla L(\beta^i)\|_\infty \leq \frac{\|\mathbf{X}\beta_{LS}\|_2^2}{2\varepsilon(k+1)} + \frac{\varepsilon \|\mathbf{X}\|_{1,2}^2}{2}$

Notes: Recall L^* is the optimal least-squares loss, β_{LS} is an optimal least-squares solution, therefore $\|\mathbf{X}\beta_{LS}\|_2 \leq \|\mathbf{y}\|_2$

Complexity of FS_ε, continued

Optimized Complexity of FS_ε

For a given number of iterations k, set $\varepsilon := \frac{\|\mathbf{X}\beta_{LS}\|_2}{\|\mathbf{X}\|_{1,2}\sqrt{k+1}}$. Then there exists $i \leq k$ for which:

(i) $L(\beta^i) - L^* \leq \frac{2p}{(\lambda_{\min}(\mathbf{X}))^2} \frac{\|\mathbf{X}\|_{1,2}^2 \|\mathbf{X}\beta_{LS}\|_2^2}{k+1}$

(ii) there exists a solution β_{LS} for which $\|\beta^i - \beta_{LS}\|_2 \leq \frac{\sqrt{4p}}{(\lambda_{\min}(\mathbf{X}))^2} \frac{\|\mathbf{X}\|_{1,2}\|\mathbf{X}\beta_{LS}\|_2}{\sqrt{k+1}}$

(iii) $\|\beta^i\|_1 \leq \frac{\sqrt{k}\|\mathbf{X}\beta_{LS}\|_2}{\|\mathbf{X}\|_{1,2}}$

(iv) $\|\beta^i\|_0 \leq k$

(v) $\|\nabla L(\beta^i)\|_\infty \leq \frac{\|\mathbf{X}\|_{1,2}\|\mathbf{X}\beta_{LS}\|_2}{\sqrt{k+1}}$

A "Smarter" Forward Stagewise Regression Algorithm (FS)

Forward Stagewise regression (FS) chooses $\varepsilon = \varepsilon_k$ "optimally" with respect to $L(\beta)$ at each iterate:

Start with $\beta^0 \leftarrow 0$, and hence $r^0 \leftarrow \mathbf{y}$.

Given β^k and $r^k := \mathbf{y} - \mathbf{X}\beta^k$, determine the weak model $\mathbf{X}_j$ most correlated with the current residuals r^k:

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

Adjust $\beta^k_{j_k}$ by $\varepsilon_k \leftarrow \arg\min_\varepsilon L(\beta^k + \varepsilon e^{j_k})$

FS Algorithm

FS Algorithm

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

At iteration $k \geq 0$:

- Compute:

$$r^k \leftarrow \mathbf{y} - \mathbf{X}\beta^k$$

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

- Set:

$$\varepsilon_k \leftarrow (r^k)^T \mathbf{X}_{j_k} / \|\mathbf{X}_{j_k}\|^2$$

$$\beta^{k+1} \leftarrow \beta^k + \varepsilon_k e^{j_k}$$

Complexity of FS

Complexity of FS

With the shrinkage factors $\varepsilon_k \leftarrow (r^k)^T \mathbf{X}_{j_k} / \|\mathbf{X}_{j_k}\|^2$, for all $k \geq 0$ it holds that:

(i) $L(\beta^k) - L^* \leq (\mathbf{y}^T\mathbf{y} - L^*)\left(1 - \frac{(\lambda_{\min}(\mathbf{X}))^2}{4p\|\mathbf{X}\|_{1,2}^2}\right)^k$

(ii) there exists a solution β_{LS} for which

$$\|\beta^k - \beta_{LS}\|_2 \leq \frac{\sqrt{2(\mathbf{y}^T\mathbf{y} - L^*)}}{\lambda_{\min}(\mathbf{X})}\left(1 - \frac{(\lambda_{\min}(\mathbf{X}))^2}{4p\|\mathbf{X}\|_{1,2}^2}\right)^{k/2}$$

(iii) $\|\beta^k\|_1 \leq \frac{\sqrt{k}\|\mathbf{X}\beta_{LS}\|_2}{\min_j\{\|\mathbf{X}_j\|_2\}}$

(iv) $\|\beta^k\|_0 \leq k$

(v) $\min_{i\in\{0,\ldots,k\}} \|\nabla L(\beta^i)\|_\infty \leq \frac{\|\mathbf{X}\|_{1,2}\|\mathbf{X}\beta_{LS}\|_2}{\sqrt{k+1}}$

What drives these results?

- observation that FS_ε "looks like" subgradient descent for some objective function $f(\cdot)$ and some decision variable $(\cdot)$
  - indeed, the objective function is $f(r) := \|\mathbf{X}^T r\|_\infty$
  - and the variables are the residuals r in the affine space $P_{\text{res}} := \{r \in \mathbb{R}^n : r = \mathbf{y} - \mathbf{X}\beta \text{ for some } \beta \in \mathbb{R}^p\}$
- application of subgradient descent convergence theory for various step-size sequences
- development of some new theory on algorithmic implications of positive semi-definite quadratic functions

What about Explicit Regularized Linear Regression?

Recall LASSO$_\delta$:

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

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

At iteration k, the Frank-Wolfe method 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}

FW for LASSO, Linear Optimization Subproblem

Linear optimization subproblem is:

$$\begin{aligned} \min_{\beta} \quad & \nabla L(\beta^k)^T \beta \\ \text{s.t.} \quad & \|\beta\|_1 \leq \delta \end{aligned}$$

Extreme points of feasible region are $\{\pm \delta e^j : j = 1, \ldots, p\}$

$\nabla L(\beta^k) = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta^k) = -(r^k)^T\mathbf{X}$

Therefore:

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

This is the same subproblem that FS_ε solves, namely find the weak model $\mathbf{X}_{j^*}$ that is most correlated with the current residuals r^k

FW Algorithm for LASSO

FW-LASSO Algorithm

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

At iteration $k \geq 0$:

- Compute:

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

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

$$\beta^{k+1} \leftarrow (1 - \bar{\alpha}_k)\beta^k + \bar{\alpha}_k \delta \ \text{sgn}((r^k)^T \mathbf{X}_{j_k}) e^{j_k}$$

Note that 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:

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

Complexity of FW-LASSO

Complexity of FW-LASSO

With the step-size rule $\bar{\alpha}_k := \frac{2}{k+2}$, after k iterations there exists an $i \in \{1, \ldots, k\}$ such that the following hold:

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

(ii) $\|\beta^k\|_1 \leq \delta$

(iii) $\|\beta^k\|_0 \leq k$

(iv) $\|\nabla L(\beta^i)\|_\infty \leq \frac{1}{2\delta}\|\mathbf{X}\beta_{LS}\|_2^2 + \frac{17.4\|\mathbf{X}\|_{1,2}^2 \delta}{k}$

Linear Regression Summary

- FS_ε and FS are interpretable as subgradient descent to minimize the correlation between the residuals and the predictors in the space of residuals
- Computational complexity guarantees for least-squares loss of iterates, distance of iterate solutions to optimal least-squares loss, sparsity and regularization bounds for FS_ε and FS
- New theory of algorithmic implications of positive semi-definite quadratic functions
- Frank-Wolfe method to minimize the explicitly regularized least-squares loss ($LASSO_\delta$) is seen to be a slight modification of FS_ε, with associated computational complexity guarantees