The Frank-Wolfe Algorithm:

New Results, and Connections to Statistical Boosting

Paul Grigas, Robert Freund, and Rahul Mazumder

http://web.mit.edu/rfreund/www/talks.html

Massachusetts Institute of Technology

May 2013

Modified Title

Original title:

The Frank-Wolfe Algorithm:

New Results, Connections to Statistical Boosting, ~~and Computation~~

Problem of Interest

Our problem of interest is:

$$h^* := \max_{\lambda} \quad h(\lambda) \quad \text{s.t.} \quad \lambda \in Q$$

$Q \subset E$ is convex and compact

$h(\cdot) : Q \to \mathbb{R}$ is concave and differentiable with Lipschitz gradient on Q

Assume it is "easy" to solve linear optimization problems on Q

Frank-Wolfe (FW) Method

Frank-Wolfe Method for maximizing $h(\lambda)$ on Q

Initialize at $\lambda_1 \in Q$, (optional) initial upper bound B_0, $k \leftarrow 1$.

1. Compute $\nabla h(\lambda_k)$.
2. Compute $\tilde{\lambda}_k \leftarrow \arg\max_{\lambda \in Q}\{h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k)\}$.

   $$B_k^w \leftarrow h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) \ .$$
   $$G_k \leftarrow \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) \ .$$

3. (Optional: compute other upper bound B_k^o), update best bound $B_k \leftarrow \min\{B_{k-1}, B_k^w, B_k^o\}$.
4. Set $\lambda_{k+1} \leftarrow \lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$, where $\bar{\alpha}_k \in [0, 1)$.

Note the condition $\bar{\alpha}_k \in [0, 1)$

Wolfe Bound

"Wolfe Bound" is $B_k^w := h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k)$

$$B_k^w \geq h^*$$

Suppose $h(\cdot)$ arises from minmax structure:

- $h(\lambda) = \min_{x \in P} \ \phi(x, \lambda)$
- P is convex and compact
- $\phi(x, \lambda) : P \times Q \to \mathbb{R}$ is convex in x and concave λ

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

Then a dual problem is $\min_{x \in P} f(x)$

In FW method, let $B_k^m := f(x_k)$ where $x_k \in \arg\min_{x \in P}\{\phi(x, \lambda_k)\}$

$$\text{Then } B_k^w \geq B_k^m$$

Wolfe Gap

"Wolfe Gap" is $G_k := B_k^w - h(\lambda_k) = \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k)$

Recall $B_k^m \leq B_k^w$

Unlike generic duality gaps, the Wolfe gap is determined completely by the current iterate λ_k . It arises in:

- Khachiyan's analysis of FW for rounding of polytopes,
- FW for boosting methods in statistics and learning,
- perhaps elsewhere . . . .

Pre-start Step for FW

Pre-start Step of Frank-Wolfe method given $\lambda_0 \in Q$ and (optional) upper bound B_{-1}

1. Compute $\nabla h(\lambda_0)$.
2. Compute $\tilde{\lambda}_0 \leftarrow \arg\max_{\lambda \in Q}\{h(\lambda_0) + \nabla h(\lambda_0)^T(\lambda - \lambda_0)\}$.

$$B_0^w \leftarrow h(\lambda_0) + \nabla h(\lambda_0)^T(\tilde{\lambda}_0 - \lambda_0) \ .$$
$$G_0 \leftarrow \nabla h(\lambda_0)^T(\tilde{\lambda}_0 - \lambda_0) \ .$$

3. (Optional: compute other upper bound B_0^o), update best bound $B_0 \leftarrow \min\{B_{-1}, B_0^w, B_0^o\}$.
4. Set $\lambda_1 \leftarrow \tilde{\lambda}_0$.

This is the same as a regular FW step at λ_0 with $\bar{\alpha}_0 = 1$

Curvature Constant $C_{h,Q}$ and Classical Metrics

Following Clarkson, define $C_{h,Q}$ to be the minimal value of C for which $\lambda, \bar{\lambda} \in Q$ and $\alpha \in [0,1]$ implies:

$$h(\lambda + \alpha(\bar{\lambda} - \lambda)) \geq h(\lambda) + \nabla h(\lambda)^T(\alpha(\bar{\lambda} - \lambda)) - C\alpha^2$$

Let $\mathrm{Diam}_Q := \max_{\lambda, \bar{\lambda} \in Q}\{\|\lambda - \bar{\lambda}\|\}$

Let $L := L_{h,Q}$ be the smallest constant L for which $\lambda, \bar{\lambda} \in Q$ implies:

$$\|\nabla h(\lambda) - \nabla h(\bar{\lambda})\|_* \leq L\|\lambda - \bar{\lambda}\|$$

It is straightforward to bound

$$C_{h,Q} \leq \frac{1}{2} L_{h,Q}(\mathrm{Diam}_Q)^2$$

Auxiliary Sequences $\{\alpha_k\}$ and $\{\beta_k\}$

Define the following two auxiliary sequences as functions of the step-sizes $\{\alpha_k\}$ from the Frank-Wolfe method:

$$\beta_k = \frac{1}{\prod_{j=1}^{k-1}(1-\bar{\alpha}_j)} \qquad \text{and} \qquad \alpha_k = \frac{\beta_k \bar{\alpha}_k}{1-\bar{\alpha}_k}\ , \qquad k \geq 1$$

(By convention $\prod_{j=1}^{0} \cdot = 1$ and $\sum_{i=1}^{0} \cdot = 0$)

Two Technical Theorems

Theorem

Consider the iterate sequences of the Frank-Wolfe Method $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$ and the sequence of upper bounds $\{B_k\}$ on h^*, using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$, and for any $k \geq 1$, the following inequality holds:

$$B_k - h(\lambda_{k+1}) \leq \frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{C_{h,Q} \sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}}$$

The summation expression appears also in the dual averages method of Nesterov. This is not a coincidence, indeed it is by design, see Grigas.

We will henceforth refer to the sequences $\{\alpha_k\}$ and $\{\beta_k\}$ as the "dual averages sequences" associated with the FW step-sizes.

Second Technical Theorem

Theorem

Consider the iterate sequences of the Frank-Wolfe Method $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$ and the sequence of Wolfe gaps $\{G_k\}$ from Step (2.), using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$, and for any $k \geq 1$ and $\ell \geq k+1$, the following inequality holds:

$$\min_{i \in \{k+1,\ldots,\ell\}} G_i \leq \frac{1}{\sum_{i=k+1}^{\ell} \bar{\alpha}_i} \left[\frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{C_{h,Q} \sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} \right] + \frac{C_{h,Q} \sum_{i=k+1}^{\ell} \bar{\alpha}_i^2}{\sum_{i=1}^{\ell} \bar{\alpha}_i}$$

A Well-Studied Step-size Sequence

Suppose we initiate the Frank-Wolfe method with the Pre-start step from a given value $\lambda_0 \in Q$ (which by definition assigns the step-size $\bar{\alpha}_0 = 1$ as discussed earlier), and then use the step-sizes:

$$\bar{\alpha}_i = \frac{2}{i+2} \qquad \text{for } i \geq 0 \tag{1}$$

Guarantee

Under the step-size sequence (1), the following inequalities hold for all $k \geq 1$:

$$B_k - h(\lambda_{k+1}) \ \leq \ \frac{4C_{h,Q}}{k+4}$$

and

$$\min_{i \in \{1,\ldots,k\}} G_i \ \leq \ \frac{8.7C_{h,Q}}{k}$$

Simple Averaging

Suppose we initiate the Frank-Wolfe method with the Pre-start step, and then use the following step-size sequence:

$$\bar{\alpha}_i = \frac{1}{i+1} \qquad \text{for } i \geq 1 \tag{2}$$

This has the property that λ_{k+1} is the simple average of $\tilde{\lambda}_0, \tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_k$

Guarantee

Under the step-size sequence (2), the following inequalities holds for all $k \geq 1$:

$$B_k - h(\lambda_{k+1}) \;\leq\; \frac{C_{h,Q}(1 + \ln(k+1))}{k+1}$$

and

$$\min_{i \in \{1,\ldots,k\}} G_i \;\leq\; \frac{2.9\; C_{h,Q}(2 + \ln(k+1))}{k - \frac{1}{2}}$$

Constant Step-size

Given $\bar{\alpha} \in (0, 1)$, suppose we initiate the Frank-Wolfe method with the Pre-start step, and then use the following step-size sequence:

$$\bar{\alpha}_i = \bar{\alpha} \qquad \text{for } i \geq 1 \tag{3}$$

(This step-size rule arises in the analysis of the Incremental Forward Stagewise Regression algorithm FS_ε)

Guarantee

Under the step-size sequence (3), the following inequality holds for all $k \geq 1$:

$$B_k - h(\lambda_{k+1}) \leq C_{h,Q} \left[(1 - \bar{\alpha})^{k+1} + \bar{\alpha} \right]$$

Constant Step-size, continued

If we decide *a priori* to run the Frank-Wolfe method for k iterations, then the optimized value of $\bar{\alpha}$ in the previous guarantee is:

$$\bar{\alpha}^* = 1 - \frac{1}{\sqrt[k]{k+1}} \tag{4}$$

which yields the following:

Guarantee

For any $k \geq 1$, using the constant step-size (4) for *all* iterations, the following inequality holds:

$$B_k - h(\lambda_{k+1}) \leq \frac{C_{h,Q}\left[1 + \ln(k+1)\right]}{k}$$

Furthermore, after $\ell = 2k + 2$ iterations it also holds that:

$$\min_{i \in \{1,\ldots,\ell\}} G_i \leq \frac{2C_{h,Q}\left[-.35 + 2\ln(\ell)\right]}{\ell - 2}$$

On the Need for Warm Start Analysis

Recall the well-studied step-size sequence initiated with a Pre-start step from $\lambda_0 \in Q$:

$$\bar{\alpha}_i = \frac{2}{i+2} \qquad \text{for } i \geq 0$$

and computational guarantee:

$$B_k - h(\lambda_{k+1}) \ \leq \ \frac{4C_{h,Q}}{k+4}$$

This guarantee is *independent* of λ_0

If $h(\lambda_0) \ll h^*$ this is good

But if $h(\lambda_0) \geq h^* - \varepsilon$ then this is not good

Let us see how we can take advantageous of a "warm start" λ_0 ...

A Warm Start Step-size Rule

Let us start the Frank-Wolfe method at an initial point λ_1 and an upper bound B_0

Let C_1 be a given *estimate* of the curvature constant $C_{h,Q}$

Use the step-size sequence: $$\bar{\alpha}_i = \frac{2}{\frac{4C_1}{B_1 - h(\lambda_1)} + i + 1} \quad \text{for } i \geq 1 \tag{5}$$

One can think of this "as if" the Frank-Wolfe method had run for $\frac{4C_1}{B_1 - h(\lambda_1)}$ iterations before arriving at λ_1

Guarantee

Using (5), the following inequality holds for all $k \geq 1$:

$$B_k - h(\lambda_{k+1}) \leq \frac{4\max\{C_1, C_{h,Q}\}}{\frac{4C_1}{B_1 - h(\lambda_1)} + k}$$

Warm Start Step-size Rule, continued

$\lambda_1 \in Q$ is initial value

C_1 is a given *estimate* of $C_{h,Q}$

Guarantee

Using (5), the following inequality holds for all $k \geq 1$:

$$B_k - h(\lambda_{k+1}) \;\leq\; \frac{4\max\{C_1, C_{h,Q}\}}{\dfrac{4C_1}{B_1 - h(\lambda_1)} + k}$$

Easy to see $C_1 \leftarrow C_{h,Q}$ optimizes the above guarantee

If $B_1 - h(\lambda_1)$ is small, the incremental decrease in the guarantee from an additional iteration is lessened. This is different from first-order methods that use prox functions and/or projections

Dynamic Version of Warm-Start Analysis

The warm-start step-sizes:

$$\bar{\alpha}_i = \frac{2}{\frac{4C_1}{B_1 - h(\lambda_1)} + i + 1} \qquad \text{for } i \geq 1$$

are based-on two pieces of information at λ_1:

- $B_1 - h(\lambda_1)$, and
- C_1

This is a *static* warm-start step-size strategy

Let us see how we can improve the computational guarantee by treating every iterate as if it were the initial iterate . . .

Dynamic Version of Warm-Starts, continued

At a given iteration k of FW, we will presume that we have:

- $\lambda_k \in Q$,
- an upper bound B_{k-1} on h^* (from previous iteration), and
- an estimate C_{k-1} of $C_{h,Q}$ (also from the previous iteration)

Consider $\bar{\alpha}_k$ of the form:

$$\bar{\alpha}_k := \frac{2}{\frac{4C_k}{B_k - h(\lambda_k)} + 2} \tag{6}$$

where $\bar{\alpha}_k$ will depend explicitly on the value of C_k.

Let us now discuss how C_k is computed . . .

Updating the Estimate of the Curvature Constant

We require that C_k (and $\bar{\alpha}_k$ which depends explicitly on C_k) satisfy $C_k \geq C_{k-1}$ and:

$$h(\lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)) \geq h(\lambda_k) + \bar{\alpha}_k(B_k - h(\lambda_k)) - C_k\bar{\alpha}_k^2 \qquad (7)$$

We first test if $C_k := C_{k-1}$ satisfies (7), and if so we set $C_k \leftarrow C_{k-1}$

If not, perform a standard doubling strategy, testing values $C_k \leftarrow 2C_{k-1}, 4C_{k-1}, 8C_{k-1}, \ldots$, until (7) is satisfied

Alternatively, say if $h(\cdot)$ is quadratic, C_k can be determined analytically

It will always hold that $C_k \leq \max\{C_0, 2C_{h,Q}\}$

Frank-Wolfe Method with Dynamic Step-sizes

FW Method with Dynamic Step-sizes for maximizing $h(\lambda)$ on Q

Initialize at $\lambda_1 \in Q$, initial estimate C_0 of $C_{h,Q}$, (optional) initial upper bound B_0, $k \leftarrow 1$.

1. Compute $\nabla h(\lambda_k)$.
2. Compute $\tilde{\lambda}_k \leftarrow \arg\max_{\lambda \in Q}\{h(\lambda_k) + \nabla h(\lambda_k)^T(\lambda - \lambda_k)\}$.

   $$B_k^w \leftarrow h(\lambda_k) + \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) \; .$$
   $$G_k \leftarrow \nabla h(\lambda_k)^T(\tilde{\lambda}_k - \lambda_k) \; .$$
3. (Optional: compute other upper bound B_k^o), update best bound $B_k \leftarrow \min\{B_{k-1}, B_k^w, B_k^o\}$.
4. Compute C_k for which the following conditions hold:
   - $C_k \geq C_{k-1}$, and
   - $h(\lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)) \geq h(\lambda_k) + \bar{\alpha}_k(B_k - h(\lambda_k)) - C_k\bar{\alpha}_k^2$, where $\bar{\alpha}_k := \frac{2}{\frac{4C_k}{B_k - h(\lambda_k)} + 2}$
5. Set $\lambda_{k+1} \leftarrow \lambda_k + \bar{\alpha}_k(\tilde{\lambda}_k - \lambda_k)$, where $\bar{\alpha}_k \in [0,1)$.

Dynamic Warm Start Step-size Rule, continued

Guarantee

Using Frank-Wolfe method with Dynamic Step-sizes, the following inequality holds for all $k \geq 1$ and $\ell \geq 1$:

$$B_{k+\ell} - h(\lambda_{k+\ell}) \;\; \leq \;\; \frac{4C_{k+\ell}}{\dfrac{4C_{k+\ell}}{B_k - h(\lambda_k)} \; + \; \ell}$$

Furthermore, if the doubling strategy is used to update the estimates C_k of $C_{h,Q}$, it holds that $C_{k+\ell} \leq \max\{C_0, 2C_{h,Q}\}$

Frank-Wolfe with Inexact Gradient

Consider the case when the gradient $\nabla h(\cdot)$ is given inexactly

d'Aspremont's δ-oracle for the gradient: given $\bar{\lambda} \in Q$, the δ-oracle returns $g_\delta(\bar{\lambda})$ that satisfies:

$$|(\nabla h(\bar{\lambda}) - g_\delta(\bar{\lambda}))^T(\hat{\lambda} - \tilde{\lambda})| \leq \delta \text{ for all } \hat{\lambda}, \tilde{\lambda} \in Q$$

Devolder/Glineur/Nesterov's (δ, L)-oracle for the the function value and the gradient: given $\bar{\lambda} \in Q$, the (δ, L)-oracle returns $h_{(\delta,L)}(\bar{\lambda})$ and $g_{(\delta,L)}(\bar{\lambda})$ that satisfies for all $\lambda \in Q$:

$$h(\lambda) \leq h_{(\delta,L)}(\bar{\lambda}) + g_{(\delta,L)}(\bar{\lambda})^T(\lambda - \tilde{\lambda})$$

and

$$h(\lambda) \geq h_{(\delta,L)}(\bar{\lambda}) + g_{(\delta,L)}(\bar{\lambda})^T(\lambda - \tilde{\lambda}) - \tfrac{L}{2}\|\lambda - \bar{\lambda}\| - \delta$$

Frank-Wolfe with d'Aspremont δ-oracle for Gradient

Suppose that we use a δ-oracle for the gradient. Then the main technical theorem is amended as follows:

Theorem

Consider the iterate sequences of the Frank-Wolfe Method $\{\lambda_k\}$ and $\{\tilde{\lambda}_k\}$ and the sequence of upper bounds $\{B_k\}$ on h^*, using the step-size sequence $\{\bar{\alpha}_k\}$. For the auxiliary sequences $\{\alpha_k\}$ and $\{\beta_k\}$, and for any $k \geq 1$, the following inequality holds:

$$B_k - h(\lambda_{k+1}) \leq \frac{B_k - h(\lambda_1)}{\beta_{k+1}} + \frac{C_{h,Q} \sum_{i=1}^{k} \frac{\alpha_i^2}{\beta_{i+1}}}{\beta_{k+1}} + 2\delta$$

The error δ does not accumulate

All bounds presented earlier are amended by 2δ

Frank-Wolfe with DGN (δ, L)-oracle for Functions and Gradient

Suppose that we use a (δ, L)-oracle for the function and gradient.

Then the errors accumulate.

(δ, L)-oracle for the function and gradient			
Method	Guarantee without Errors	Guarantee with Errors	"Sparsity" of Iterates
Frank-Wolfe	$O\left(\frac{1}{k}\right)$	$O\left(\frac{1}{k}\right) + O(\delta k)$	YES
Prox Gradient	$O\left(\frac{1}{k}\right)$	$O\left(\frac{1}{k}\right) + O(\delta)$	NO
Acclerated Prox Grad.	$O\left(\frac{1}{k^2}\right)$	$O\left(\frac{1}{k^2}\right) + O(\delta k)$	NO

No method dominates all three criteria

Applications to Statistical Boosting

We consider two applications in statistical boosting:

1. Linear Regression
2. Binary Classification / Supervised Learning

Linear Regression

Consider linear regression:

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

$\mathbf{y} \in \mathbb{R}^n$ is the response , $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the model matrix

$\beta \in \mathbb{R}^p$ are the coefficients , and $\mathbf{e} \in \mathbb{R}^n$ are the errors

In high-dimension statistical regime, especially with $p \gg n$, we desire:

- good predictive performance,
- good performance on samples (residuals $r := \mathbf{y} - \mathbf{X}\beta$ are small),
- an "interpretable model" β
- coefficients are not excessively large ($\|\beta\| \leq \delta$), and
- a sparse solution (β has few non-zero coefficients)

LASSO

The form of the LASSO we consider is:

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

$\delta > 0$ is the regularization parameter

As is well-known, the $\|\cdot\|_1$-norm regularizer induces sparse solutions

Let $\|\beta\|_0$ denote the number of non-zero coefficients of the vector β

LASSO with Frank-Wolfe

Consider using Frank-Wolfe to solve the LASSO:

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

- the linear optimization subproblem $\min\limits_{\|\beta\|_1 \leq \delta} c^T\beta$ is trivial to solve for any c:

  let $j^* \leftarrow \arg \max\limits_{j \in \{1,\ldots,p\}} |c_j|$

  then $\arg \min\limits_{\|\beta\|_1 \leq \delta} c^T\beta \to -\delta \text{sgn}(c_{j^*}) e^{j^*}$

- extreme points of feasible region are $\pm\delta e^1, \ldots, \pm\delta e^p$
- therefore $\|\beta^{k+1}\|_0 \leq \|\beta^k\|_0 + 1$

Frank-Wolfe for LASSO

Adaptation of Frank-Wolfe Method for LASSO

Initialize at β_0 with $\|\beta_0\|_1 \leq \delta$.
At iteration k:

1. Compute:

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

2. Set:

$$\beta_{j_k}^{k+1} \leftarrow (1 - \bar{\alpha}_k)\beta_{j_k}^k + \bar{\alpha}_k \delta \mathrm{sgn}((r^k)^T \mathbf{X}_{j_k})$$
$$\beta_j^{k+1} \leftarrow (1 - \bar{\alpha}_k)\beta_j^k \text{ for } j \neq j_k \text{, and where } \bar{\alpha}_k \in [0,1]$$

Computational Guarantees for Frank-Wolfe on LASSO

Guarantees

Suppose that we use the Frank-Wolfe Method to solve the LASSO problem, with either the fixed step-size rule $\bar{\alpha}_i = \frac{2}{i+2}$ or a line-search to determine $\bar{\alpha}_i$ for $i \geq 0$. Then after k iterations, there exists an $i \in \{0, \ldots, k\}$ satisfying:

- $\frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta^i\|_2^2 - L_\delta^* \leq \frac{17.4\|\mathbf{X}\|_{1,2}^2\delta^2}{k}$
- $\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta^i)\|_\infty \leq \frac{1}{2\delta}\|\mathbf{X}\beta_{\mathrm{LS}}\|_2^2 + \frac{17.4\|\mathbf{X}\|_{1,2}^2\delta}{k}$
- $\|\beta^i\|_0 \leq k$
- $\|\beta^i\|_1 \leq \delta$

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

Frank-Wolfe on LASSO, and FS$_\varepsilon$

- There are some structural connections between Frank-Wolfe on LASSO and Incremental Forward Stagewise Regression FS$_\varepsilon$
- There are even more connections in the case of Frank-Wolfe with constant step-size

Binary Classification / Supervised Learning

The set-up of the 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 n base classifiers $\mathcal{H} = \{h_1(\cdot), \ldots, h_n(\cdot)\}$ where each $h_j(\cdot) : \mathcal{X} \to [-1, 1]$
- $h_j(x_i)$ is classifier j's score of example x_i
- $y_i h_j(x_i) > 0$ if and only if classifier j correctly classifies example x_i

We would like to construct a classifier $H = \lambda_1 h_1(\cdot) + \cdots + \lambda_n h_n(\cdot)$ that performs significantly better than any individual classifier in $\mathcal{H}$

Binary Classification, continued

We construct a new classifier $H = H_\lambda := \lambda_1 h_1(\cdot) + \cdots + \lambda_n h_n(\cdot)$ from nonnegative multipliers $\lambda \in \mathbb{R}^n_+$

$$H(x) = H_\lambda(x) := \lambda_1 h_1(x) + \cdots + \lambda_n h_n(x)$$

(associate $\operatorname{sgn} H \equiv H$ for ease of notation)

In the high-dimensional case where $n \gg m \gg 0$, we desire:

- good predictive performance,
- good performance on the training data ($y_i H_\lambda(x_i) > 0$ for all examples x_i),
- a good "interpretable model" λ
- coefficients are not excessively large ($\|\lambda\| \leq \delta$), and
- λ has few non-zero coefficients ($\|\lambda\|_0$ is small)

Weak Learner

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

$A_{ij} > 0$ if and only if classifier $h_j(\cdot)$ correctly classifies example x_i

We suppose we have a weak learner $\mathcal{W}(\cdot)$ that, for any distribution w on the examples ($w \in \Delta_m := \{w \in \mathbb{R}^m : e^T w = 1,\ w \geq 0\ \}$), returns the base classifier $h_{j^*}(\cdot)$ in $\mathcal{H}$ that does best on the weighted example determined by w:

$$\sum_{i=1}^{m} w_i y_i h_{j^*}(x_i) = \max_{j=1,\ldots,n} \sum_{i=1}^{m} w_i y_i h_j(x_i) = \max_{j=1,\ldots,n} w^T A_j$$

Performance Objectives: Maximize the Margin

The margin of the classifier H_λ is defined by:

$$p(\lambda) := \min_{i \in \{1,\ldots,m\}} y_i H_\lambda(x_i) = \min_{i \in \{1,\ldots,m\}} (A\lambda)_i$$

One can then solve:

$$\begin{aligned} p^* \quad &= \max_{\lambda} \quad p(\lambda) \\ &\text{s.t.} \quad \lambda \in \Delta_n \end{aligned}$$

Performance Objectives: Minimize the Exponential Loss

The (logarithm of the) exponential loss of the classifier H_λ is defined by:

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

One can then solve:

$$\begin{aligned} L^*_{\exp,\delta} \quad = \min_{\lambda} \quad & L_{\exp}(\lambda) \\ \text{s.t.} \quad & \|\lambda\|_1 \le \delta \\ & \lambda \ge 0 \end{aligned}$$

Minimizing Exponential Loss with Frank-Wolfe

Consider using Frank-Wolfe to minimize the exponential loss problem:

$$\begin{array}{rll} L^*_{\exp,\delta} & = \min_{\lambda} & L_{\exp}(\lambda) \\ & \text{s.t.} & \|\lambda\|_1 \leq \delta \\ & & \lambda \geq 0 \end{array}$$

- the linear optimization subproblem $\min_{\|\lambda\|_1 \leq \delta,\ \lambda \geq 0} c^T\lambda$ is trivial to solve for any c:

  $\arg\min_{\|\lambda\|_1 \leq \delta,\ \lambda \geq 0} c^T\lambda \rightarrow 0$ if $c \geq 0$

  else let $j^* \leftarrow \arg\max_{j \in \{1,\ldots,p\}} c_j$

  then $\arg\min_{\|\lambda\|_1 \leq \delta,\ \lambda \geq 0} c^T\lambda \rightarrow \delta e^{j^*}$

- extreme points of feasible region are $0, \delta e^1, \ldots, \delta e^n$
- therefore $\|\lambda^{k+1}\|_0 \leq \|\lambda^k\|_0 + 1$

Frank-Wolfe for Exponential Loss Minimization

Adaptation of Frank-Wolfe Method for Minimizing Exponential Loss

Initialize at $\lambda^0 \geq 0$ with $\|\lambda^0\|_1 \leq \delta$.

Set $w_i^0 = \frac{\exp(-(A\lambda^0)_i)}{\sum_{l=1}^m \exp(-(A\lambda^0)_l)}$ $i = 1, \ldots, m$. Set $k = 0$.

At iteration k:

1. Compute $j_k \in \mathcal{W}(w^k)$
2. Choose $\bar{\alpha} \in [0, 1]$ and set:

$$\lambda_{j_k}^{k+1} \leftarrow (1 - \bar{\alpha}_k)\lambda_{j_k}^k + \bar{\alpha}_k \delta$$

$\lambda_j^{k+1} \leftarrow (1 - \bar{\alpha}_k)\lambda_j^k$ for $j \neq j_k$, and where $\bar{\alpha}_k \in [0, 1]$

$w_j^{k+1} \leftarrow (w_j^k)^{1-\bar{\alpha}_k} \exp(-\bar{\alpha}_k \delta A_{i,j_k})$, $i = 1, \ldots, m$ and re-normalize w^{k+1} so that $e^T w^{k+1} = 1$

Computational Guarantees for FW for Binary Classification

Guarantees

Suppose that we use the Frank-Wolfe Method to solve the exponential loss minimization problem, with either the fixed step-size rule $\bar{\alpha}_i = \frac{2}{i+2}$ or a line-search to determine $\bar{\alpha}_i$, using $\bar{\alpha}_0 = 1$. Then after k iterations:

- $L_{\exp}(\lambda^k) - L^*_{\exp,\delta} \le \dfrac{8\delta^2}{k+3}$
- $p^* - p(\bar{\lambda}^k) \le \dfrac{8\delta}{k+3} + \dfrac{\ln(m)}{\delta}$
- $\|\lambda^k\|_0 \le k$
- $\|\lambda^k\|_1 \le \delta$

where $\bar{\lambda}^k$ is the normalization of λ^k, namely $\bar{\lambda}^k = \frac{\lambda^k}{e^T\lambda^k}$

Frank-Wolfe for Binary Classification, and AdaBoost

- There are some structural connections between Frank-Wolfe for binary classification, and AdaBoost

- $L_{\exp}(\lambda)$ is a (Nesterov-) smoothing of the margin using smoothing parameter $\mu = 1$:
  - write the margin as $p(\lambda) := \min_{i \in \{1,\ldots,m\}} (A\lambda)_i = \min_{w \in \Delta_m} (w^T A\lambda)$
  - do μ-smoothing of the margin using entropy function $e(w)$:

$$p_\mu(\lambda) := \min_{w \in \Delta_m} (w^T A\lambda + \mu e(w))$$

  - then $p_\mu(\lambda) = -\mu L_{\exp}(\lambda/\mu)$
  - it easily follows that $\frac{1}{\mu} p(\lambda) \le -L_{\exp}(\lambda/\mu) \le \frac{1}{\mu} p(\lambda) + \ln(m)$

Comments

- Computational evaluation of Frank-Wolfe for boosting:
  - LASSO
  - binary classification
- Structural connections in G-F-M:
  - AdaBoost is equivalent to Mirror Descent for maximizing the margin
  - Incremental Forward Stagewise Regression (FS_ε) is equivalent to subgradient descent for minimizing $\|\mathbf{X}^T r\|_\infty$