An Extended Frank-Wolfe Method, with Application to Low-Rank Matrix Completion

Robert M. Freund, MIT

joint with Paul Grigas (UC Berkeley) and Rahul Mazumder (MIT)

CDC, December 2016

Outline of Topics

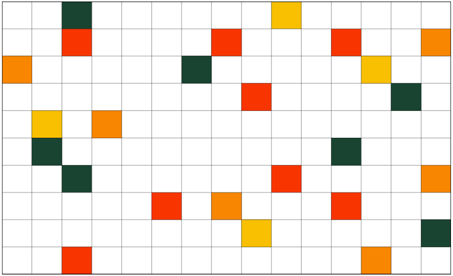
- Motivating Application: Low-rank Matrix Completion
- Review of Frank-Wolfe Method
- Frank-Wolfe for Low-rank Matrix Completion
- "In-Face Step" Extension of Frank-Wolfe Method
- Computational Experiments and Results

Outline of Topics

- Motivating Application: Low-rank Matrix Completion
- Review of Frank-Wolfe Method
- Frank-Wolfe for Low-rank Matrix Completion
- "In-Face Step" Extension of Frank-Wolfe Method
- Computational Experiments and Results

Motivating Problem: Matrix Completion

Task: given a partially observable data matrix X, predict the unobserved entries



Application to recommender systems, sensor networks, microarray data, etc.

A popular example is the Netflix challenge: users are rows, movies are columns, ratings (1 to 5 stars) are entries

Low-Rank Matrix Completion

Let $X \in \mathbb{R}^{m \times n}$ be the partially observed data matrix

Ω denotes the entries of X that are observable, $|\Omega| \ll m \times n$

It is natural to presume that the "true" data matrix has low rank structure (analogous to sparsity in linear regression)

The estimated data matrix Z should have:

- Good predictive performance on the unobserved entries
- Interpretability via low-rank structure

Nuclear Norm Regularization for Matrix Completion

Low-Rank Least Squares Problem

$$
\begin{aligned}
P_r: \quad z^* \quad := \quad & \min_{Z \in \mathbb{R}^{m \times n}} \quad \tfrac{1}{2} \textstyle\sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 \\
& \text{s.t.} \quad \operatorname{rank}(Z) \leq r
\end{aligned}
$$

Replace rank constraint with constraint/penalty on the nuclear norm of Z

$$\text{Nuclear norm is:} \quad \|Z\|_N := \sum_{j=1}^{r} \sigma_j$$

where

- $Z = UDV^T$
- $U \in \mathbb{R}^{m \times r}$ is orthonormal, $V \in \mathbb{R}^{n \times r}$ is orthonormal
- $D = \operatorname{Diag}(\sigma_1, \ldots, \sigma_r)$ comprises the non-zero singular values of Z

Nuclear Norm Regularized Problem

We aspire to solve:

$$P_r: \quad z^* \quad := \quad \min_{Z \in \mathbb{R}^{m \times n}} \quad f(Z) := \tfrac{1}{2} \textstyle\sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2$$
$$\text{s.t.} \quad \operatorname{rank}(Z) \le r$$

Instead we will solve:

$$NN_\delta: \quad f^* \quad := \quad \min_{Z \in \mathbb{R}^{m \times n}} \quad f(Z) := \tfrac{1}{2} \textstyle\sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2$$
$$\text{s.t.} \quad \|Z\|_N \le \delta$$

Nuclear Norm Regularized Low-Rank Matrix Completion

Relaxation of the "hard" problem is the nuclear norm regularized least squares problem:

Nuclear Norm Regularized Matrix Completion [Fazel 2002]

$$NN_\delta : \quad f^* \quad := \quad \min_{Z \in \mathbb{R}^{m \times n}} \quad f(Z) := \tfrac{1}{2} \textstyle\sum_{(i,j)\in\Omega} (Z_{ij} - X_{ij})^2$$

$$\text{s.t.} \quad \|Z\|_N \le \delta$$

The above is a convex optimization problem

The nuclear norm constraint is intended to induce low-rank solutions

- Think ℓ_1 norm on the singular values of Z

Equivalent SDP Problem on Spectrahedron

Instead of working directly with $\|Z\|_N \leq \delta$, perhaps work with spectrahedral representation:

$$\|Z\|_N \leq \delta \quad \text{iff} \quad \text{there exists } W, Y \text{ for which } \begin{cases} A := \begin{bmatrix} W & Z \\ Z^T & Y \end{bmatrix} \succeq 0 \\ \\ \mathrm{trace}(A) \leq 2\delta \end{cases}$$

Solve the equivalent problem on spectrahedron:

$$\begin{array}{llll} S_\delta : & f^* & := & \min\limits_{Z,W,Y} \quad f(Z) := \frac{1}{2}\sum_{(i,j)\in\Omega}(Z_{ij} - X_{ij})^2 \\ & & & \quad \text{s.t.} \quad \begin{bmatrix} W & Z \\ Z^T & Y \end{bmatrix} \succeq 0 \\ & & & \qquad\quad \mathrm{trace}(W) + \mathrm{trace}(Y) \leq 2\delta \end{array}$$

Nuclear Norm Regularized Low-Rank Matrix Completion, cont.

Nuclear Norm Regularized Matrix Completion [Fazel 2002]

$$NN_\delta : \quad f^* \quad := \quad \min_{Z \in \mathbb{R}^{m \times n}} \quad f(Z) := \tfrac{1}{2} \textstyle\sum_{(i,j)\in\Omega} (Z_{ij} - X_{ij})^2$$

$$\text{s.t.} \quad \|Z\|_N \leq \delta$$

Extensive work studying data generating mechanisms that ensure that optimal solutions of NN_δ have low rank (e.g. [Candes and Recht 2009], [Candes and Tao 2010], [Recht et al. 2010], ...)

It is imperative that the algorithm for NN_δ reliably delivers solutions with *good predictive performance* and *low rank* after a reasonable amount of time

NN_δ Alignment with Frank-Wolfe Algorithm

$$NN_\delta: \quad f^* \quad := \quad \min_{Z \in \mathbb{R}^{m\times n}} \quad f(Z) := \tfrac{1}{2}\textstyle\sum_{(i,j)\in\Omega}(Z_{ij} - X_{ij})^2$$

$$\text{s.t.} \quad \|Z\|_N \leq \delta$$

NN_δ aligns well for computing solutions using Frank-Wolfe algorithm:

- $\nabla f(Z) = P_\Omega(Z - X) := (Z - X)_\Omega$ is viable to compute
- the Frank-Wolfe method needs to solve a linear optimization subproblem at each iteration of the form:

  $$\tilde{Z} \leftarrow \arg \min_{\|Z\|_N \leq \delta} \{C \bullet Z\}$$

  where $C \in \mathbb{R}^{m\times n}$ and $C \bullet Z := \text{trace}(C^T Z)$
- The subproblem solution is straightforward:
  - compute largest singular value σ_1 of C with associated left and right normalized eigenvectors u_1, v_1
  - $\tilde{Z} \leftarrow -\delta u_1 (v_1)^T$

Outline of Topics

- Motivating Application: Low-rank Matrix Completion
- Review of Frank-Wolfe Method
- Frank-Wolfe for Low-rank Matrix Completion
- "In-Face Step" Extension of Frank-Wolfe Method
- Computational Experiments and Results

Frank-Wolfe Method

Problem of interest is:

$$\begin{array}{rlll} \mathrm{CP}: & f^* & := & \min\limits_{x} & f(x) \\ & & & \text{s.t.} & x \in S \end{array}$$

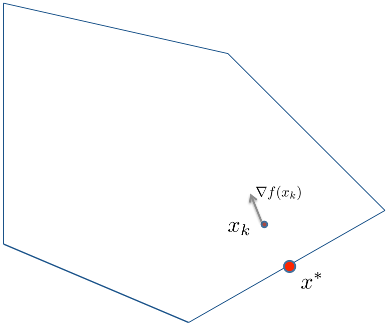
- $S \subset \mathbb{R}^n$ is compact and convex
- $f(\cdot)$ is convex on S
- let x^* denote any optimal solution of CP
- $\nabla f(\cdot)$ is Lipschitz on S: $\|\nabla f(x) - \nabla f(y)\|_* \leq L\|s - y\|$ for all $x, y \in S$
- it is "easy" to do linear optimization on S for any c :

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

Frank-Wolfe Method, cont.

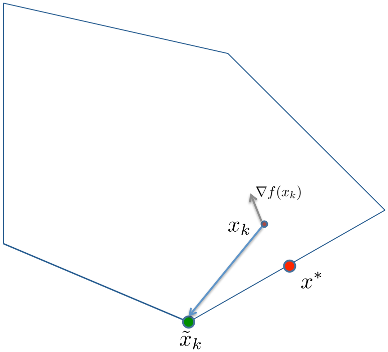
$$\begin{array}{rrcll} \mathrm{CP}: & f^* & := & \min_x & f(x) \\ & & & \text{s.t.} & x \in S \end{array}$$

At iteration k of the Frank-Wolfe method:



Frank-Wolfe Method, cont.

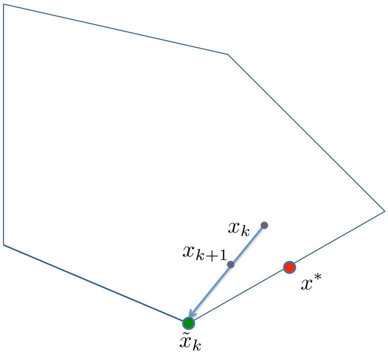
At iteration k of the Frank-Wolfe method:



Set $\tilde{x}_k \leftarrow \arg\min_{x \in S} \{ f(x_k) + \nabla f(x_k)^T (x - x_k) \}$

Frank-Wolfe Method, cont.

At iteration k of the Frank-Wolfe method:



Set $x_{k+1} \leftarrow x_k + \bar{\alpha}_k(\tilde{x}_k - x_k)$, where $\bar{\alpha}_k \in [0, 1]$

Frank-Wolfe Method, cont.

$$\text{CP}: \quad f^* \quad := \quad \begin{array}{ll} \min\limits_{x} & f(x) \\ \text{s.t.} & x \in S \end{array}$$

Basic Frank-Wolfe method for minimizing $f(x)$ on S

Initialize at $x_0 \in S$, (optional) initial lower bound $B_{-1} \leq f^*$, $k \leftarrow 0$.

At iteration k :

1. Compute $\nabla f(x_k)$.
2. Compute $\tilde{x}_k \leftarrow \arg\min\limits_{x \in S}\{\nabla f(x_k)^T x\}$.
3. Update lower bound: $B_k \leftarrow \max\{B_{k-1}, f(x_k) + \nabla f(x_k)^T(\tilde{x}_k - x_k)\}$
4. Set $x_{k+1} \leftarrow x_k + \bar{\alpha}_k(\tilde{x}_k - x_k)$, where $\bar{\alpha}_k \in [0, 1]$.

Some Step-size Rules/Strategies

- "Recent standard": $\bar{\alpha}_k = \frac{2}{k+2}$
- Exact line-search: $\bar{\alpha}_k = \arg\min_{\alpha \in [0,1]} \{ f(x_k + \alpha(\tilde{x}_k - x_k)) \}$
- QA (Quadratic approximation) step-size:

$$\bar{\alpha}_k = \min \left\{ 1, \frac{-\nabla f(x_k)^T (\tilde{x}_k - x_k)}{L \|\tilde{x}_k - x_k\|^2} \right\}$$

- Dynamic strategy: determined by some history of optimality bounds, see [FG]
- Simple averaging: $\bar{\alpha}_k = \frac{1}{k+1}$
- Constant step-size: $\bar{\alpha}_k = \bar{\alpha}$ for some given $\bar{\alpha} \in [0, 1]$

Computational Guarantee for Frank-Wolfe

A Computational Guarantee for the Frank-Wolfe algorithm

If the step-size sequence $\{\bar{\alpha}_k\}$ is chosen by exact line-search or a certain quadratic approximation (QA) line-search rule, then for all $k \geq 1$ it holds that:

$$f(x_k) - f^* \;\leq\; f(x_k) - B_k \;\leq\; \frac{1}{\frac{1}{f(x_0)-B_0} + \frac{k}{2C}} \;<\; \frac{2C}{k}$$

where $C = L \cdot \mathrm{diam}(S)^2$.

It will be useful to understand this guarantee as arising from:

$$\frac{1}{f(x_{i+1}) - B_{i+1}} \;\geq\; \frac{1}{f(x_i) - B_i} + \frac{1}{2C} \quad \text{for} \quad i = 0, 1, \ldots$$

Diameter and Lipschitz Gradient

Let $\|\cdot\|$ be a prescribed norm on $\mathbb{R}^n$

Dual norm is $\|s\|_* := \max_{\|x\| \leq 1}\{s^T x\}$

$B(x, \rho) := \{y : \|y - x\| \leq \rho\}$

$\mathrm{Diam}(S) := \max_{x,y \in S}\{\|x - y\|\}$

Let L be the Lipschitz constant of $\nabla f(\cdot)$ on S:

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

Renewed Interest in Frank-Wolfe Algorithm

Renewed interest in Frank-Wolfe algorithm due to:

- Relevance of applications
  - Regression
  - Boosting/classification
  - Matrix completion
  - Image construction
  - ...
- Requirements for only moderately high accuracy solutions
- Necessity of simple methods for huge-scale problems
- Structural implications (sparsity, low-rank) induced by the algorithm itself

A Linear Convergence Result

$f(\cdot)$ is u-strongly convex on S if there exists $u > 0$ for which:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{u}{2}\|y - x\|^2 \quad \text{for all } x, y \in S$$

Sublinear and Linear Convergence under Interior Solutions and Strong Convexity ∼[W,GM]

Suppose the step-size sequence $\{\bar{\alpha}_k\}$ is chosen using the QA rule or by line-search. Then for all $k \geq 1$ it holds that:

$$f(x^k) - f^* \leq \min\left\{ \frac{2L(\mathrm{Diam}(S))^2}{k} \ , \ (f(x^0) - f^*)\left[1 - \left(\frac{u}{L}\frac{\rho^2}{(\mathrm{Diam}(S))^2}\right)\right]^k \right\}$$

where $\rho = \mathrm{dist}(x^*, \partial S)$.

Frank-Wolfe For Low-Rank Matrix Completion

$$NN_\delta : \quad f^* \quad := \quad \min_{Z \in \mathbb{R}^{m \times n}} \quad f(Z) := \tfrac{1}{2} \textstyle\sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2$$

$$\text{s.t.} \quad \|Z\|_N \le \delta$$

We focus on the Frank-Wolfe method and its extensions

- A key driver of our work is the favorable low-rank structural properties of Frank-Wolfe

Frank-Wolfe has been directly (and indirectly) applied to NN_δ by [Jaggi and Sulovsk 2010], [Harchaoui, Juditsky, and Nemirovski 2012], [Mu et al. 2014], and [Rao, Shah, and Wright 2014]

Frank-Wolfe For Low-Rank Matrix Completion, cont.

$$NN_\delta : \quad f^* \quad := \quad \min_{Z \in \mathbb{R}^{m \times n}} \quad f(Z) := \tfrac{1}{2} \textstyle\sum_{(i,j)\in\Omega} (Z_{ij} - X_{ij})^2$$

$$\text{s.t.} \quad \|Z\|_N \leq \delta$$

As applied to NN_δ, at iteration k Frank-Wolfe computes

$$\tilde{Z}^k \leftarrow \arg \min_{\|Z\|_N \leq \delta} \{\nabla f(Z^k) \bullet Z\}$$

and updates:

$$Z^{k+1} \leftarrow (1 - \bar{\alpha}_k) Z^k + \bar{\alpha}_k \tilde{Z}^k \quad \text{for some } \bar{\alpha}_k \in [0,1]$$

Note that $\tilde{Z}^k \leftarrow -\delta u_1 (v_1)^T$ is a rank-one matrix where u_1, v_1 are the singular vectors associated with the largest singular value of $\nabla f(Z^k)$

Properties of Frank-Wolfe Applied to NN_δ

At each iteration, Frank-Wolfe forms Z^{k+1} by adding a rank-one matrix $\tilde{Z}^k$ to a scaling of the current iterate Z^k:

$$Z^{k+1} \leftarrow (1-\bar{\alpha}_k)Z^k + \bar{\alpha}_k \tilde{Z}^k = (1-\bar{\alpha}_k)Z^k - \bar{\alpha}_k \delta u_1 (v_1)^T$$

Assuming that $\mathrm{rank}(Z^0) = 1$, this implies that $\mathrm{rank}(Z^k) \leq k+1$

Combined with the optimality guarantee for Frank-Wolfe, we have a nice tradeoff between data-fidelity and low-rank structure:

$$f(Z^k) - f^* \leq \frac{8\delta^2}{k+3} \quad \text{and} \quad \mathrm{rank}(Z^k) \leq k+1$$

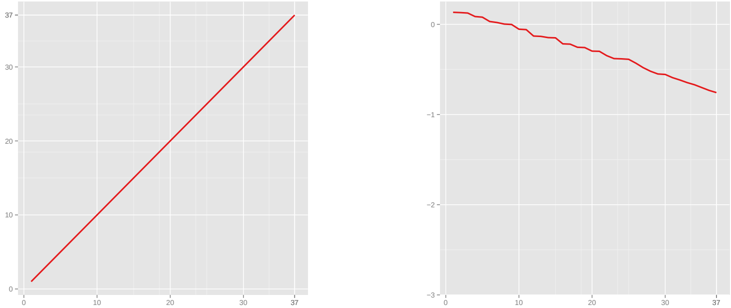
What happens in practice?

Practical Behavior of Frank-Wolfe Applied to NN_δ, cont.

Instance with $m = 2000$, $n = 2500$ and $\text{rank}(Z^*) = 37$

Frank-Wolfe Applied to a Typical Instance of NN_δ: 37 Iterations

$\text{rank}(Z^k)$ vs. k $\qquad$ $\text{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. k

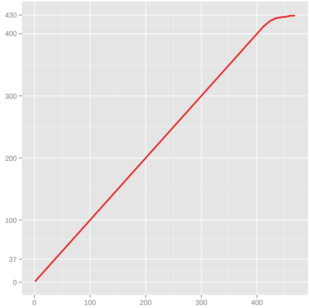


Practical Behavior of Frank-Wolfe Applied to NN_δ, cont.

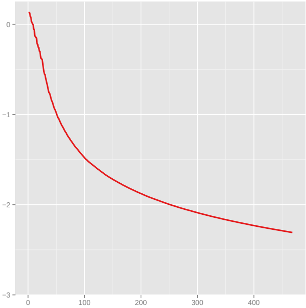
Instance with $m = 2000$, $n = 2500$ and $\text{rank}(Z^*) = 37$

Frank-Wolfe Applied to a Typical Instance of NN_δ: ~ 450 Iterations

$\text{rank}(Z^k)$ vs. k



$\text{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. k

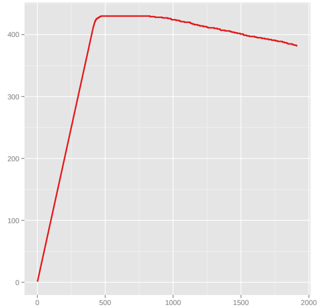


Practical Behavior of Frank-Wolfe Applied to NN_δ, cont.

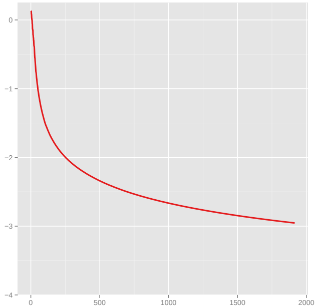
Instance with $m = 2000$, $n = 2500$ and $\text{rank}(Z^*) = 37$

Frank-Wolfe Applied to a Typical Instance of NN_δ: $\sim$ 2000 Iterations

$\text{rank}(Z^k)$ vs. k



$\text{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. k



Practical Behavior of Frank-Wolfe Applied to NN_δ

Theoretical bounds for Frank-Wolfe:

$$f(Z^k) - f^* \leq \frac{8\delta^2}{k+3} \quad \text{and} \quad \mathrm{rank}(Z^k) \leq k+1$$

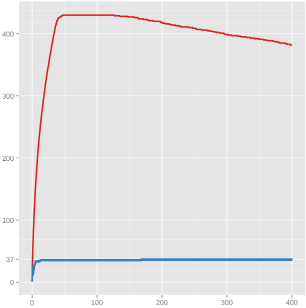
We propose an extension of Frank-Wolfe that:

- In theory has computational guarantees for $f(Z^k) - f^*$ and $\mathrm{rank}(Z^k)$ that are at least as good as (and sometimes better than) Frank-Wolfe
- In practice is able to efficiently deliver a solution with the correct optimal rank <u>and</u> better training error than Frank-Wolfe

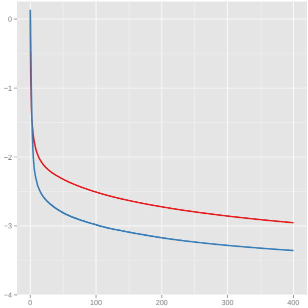
Preview of In-Face Extended Frank-Wolfe Behavior

Preview of IF Extended FW (versus FW)

rank(Z^k) vs. Time



$\text{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. Time



For this problem, rank$(Z^*) = 37$ $(m = 2000, n = 2500)$

Outline of Topics

- Motivating Application: Low-rank Matrix Completion
- Review of Frank-Wolfe Method
- Frank-Wolfe for Low-rank Matrix Completion
- "In-Face Step" Extension of Frank-Wolfe Method
- Computational Experiments and Results

In-Face Extended Frank-Wolfe Overview

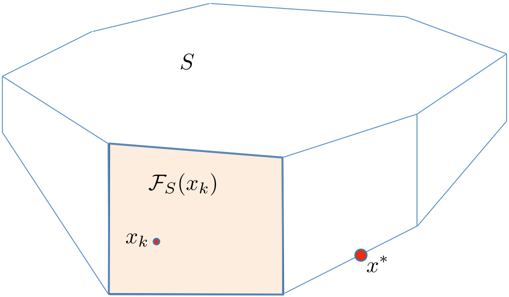
We develop a general methodological approach for preserving structure (low rank) while making objective function improvements based on "in-face directions"

For the matrix completion problem NN_δ, in-face directions preserve low-rank solutions

- This is good since Z^* should (hopefully) be low-rank
- Working with low-rank matrices also yields computational savings at all intermediate iterations

In-Face Directions

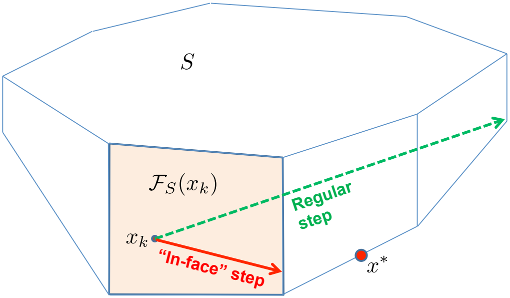
Let $\mathcal{F}_S(x_k)$ denote the minimal face of S that contains x_k



In-Face Directions, cont.

An in-face step moves in any "reasonably good" direction that remains in $\mathcal{F}_S(x_k)$

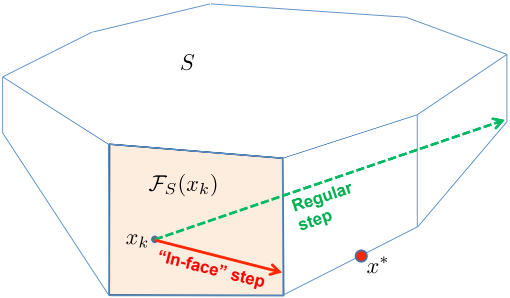
The in-face direction should be relatively easy to compute



In-Face Directions, cont.

Main examples of "in-face" directions:

- Wolfe's "away step" direction
- Fully optimizing $f(\cdot)$ over $\mathcal{F}_S(x_k)$



What are the Faces of the Nuclear Norm Ball?

The nuclear norm ball of radius δ:

$$\mathcal{B}(0,\delta) := \{Z \in \mathbb{R}^{m\times n} : \|Z\|_N \leq \delta\}$$

Theorem [So 1990]: Minimal Faces of the Nuclear Norm Ball

Let $Z \in \partial\mathcal{B}(0,\delta)$ be given, and consider the thin SVD of $Z = UDV^T$. Then the minimal face of $\partial\mathcal{B}(0,\delta)$ containing Z is:

$$\mathcal{F}(Z) = \{UMV^T : M \in \mathbb{S}^{r\times r},\ M \succeq 0\ ,\ \text{trace}(M) = \delta\}\ ,$$

and $\dim(\mathcal{F}(Z)) = r(r+1)/2 - 1$.

- Low-dimensional faces of the nuclear norm ball correspond to low-rank matrices on its boundary.
- All matrices lying on $\mathcal{F}(Z)$ have rank at most r
- $\mathcal{F}(Z)$ is a linear transformation of a standard spectrahedron

In-Face Extended Frank-Wolfe Method

In-face directions have two important properties:

1. They keep the next iterate within $\mathcal{F}_S(x_k)$
2. They should be relatively easy to compute

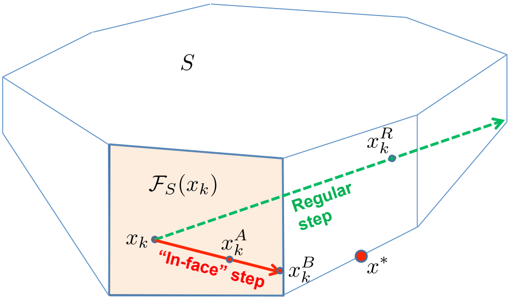
Outline of each iteration of the In-Face Extended Frank-Wolfe Method:

1. Compute the in-face direction
2. Decide whether or not to accept an in-face step (partial or full), by checking its objective function value progress
3. If we reject an in-face step, compute a regular FW step

In-Face Steps

Three points to choose from:

- x_k^B: full step to the relative boundary of $\mathcal{F}_S(x_k)$
- x_k^A: partial step that remains in the relative interior of $\mathcal{F}_S(x_k)$
- x_k^R: the "regular" Frank-Wolfe step



Decision Rule for In-Face Extended Frank-Wolfe Method

Recall the following useful property for regular FW with line-search or quadratic approximation line-search step-sizes:

$$\frac{1}{f(x_{i+1}) - B_{i+1}} \geq \frac{1}{f(x_i) - B_i} + \frac{1}{2C}$$

Note that these "reciprocal gaps" are available at every iteration

We will use these reciprocal gaps to measure the progress made by in-face directions

Decision Rule for In-Face Extended Frank-Wolfe Method, cont.

Decision Rule for In-Face Extended Frank-Wolfe Method

Set $\gamma_2 \geq \gamma_1 \geq 0$ (think $\gamma_2 = 1, \gamma_1 = 0.3$)

At iteration k :

1. Decide which of x_k^A, x_k^B, x_k^R to accept as next iterate:
    1. (Go to a lower-dimensional face.) Set $x_{k+1} \leftarrow x_k^B$ if
    $$\frac{1}{f(x_k^B) - B_k} \geq \frac{1}{f(x_k) - B_k} + \frac{\gamma_1}{2C} \ .$$
    2. (Stay in current face.) Else, set $x_{k+1} \leftarrow x_k^A$ if
    $$\frac{1}{f(x_k^A) - B_k} \geq \frac{1}{f(x_k) - B_k} + \frac{\gamma_2}{2C} \ .$$
    3. (Do regular FW step and update lower bound.) Else, set $x_{k+1} \leftarrow x_k^R$.

Computational Guarantee for Extended FW Method

In the first k iterations, let:

$$\begin{aligned} N_k^B &= \text{number of steps to the boundary of the minimal face} \\ N_k^A &= \text{number of steps to the interior of the minimal face} \\ N_k^R &= \text{number of regular Frank-Wolfe steps} \end{aligned}$$

$$k = N_k^B + N_k^A + N_k^R$$

Computational Guarantee for Extended Frank-Wolfe Method

Theorem: Suppose that the step-sizes are determined by exact line-search or QA line-search rule. After k iterations of the Extended Frank-Wolfe method it holds that:

$$f(x_k) - f^* < \frac{2LD^2}{\gamma_1 N_k^B + \gamma_2 N_k^A + N_k^R} \ ,$$

where $D := \operatorname{diam}(S)$.

In-Face Extended Frank-Wolfe Summary

In-Face Extended Frank-Wolfe method intelligently combines "in-face directions" with "regular Frank-Wolfe directions"

Computational guarantees improve upon regular Frank-Wolfe

- Objective function value guarantee is still $O(1/k)$
- Guarantee bound on the rank of the iterates is stronger:

$$\text{rank}(Z^k) \;\leq\; k + 1 - 2N_k^B - N_k^A \ .$$

Outline of Topics

- Motivating Application: Low-rank Matrix Completion
- Review of Frank-Wolfe Method
- Frank-Wolfe for Low-rank Matrix Completion
- "In-Face Step" Extension of Frank-Wolfe Method
- Computational Experiments and Results

Computational Experiments and Results

Here, we consider three versions of the In-Face Extended Frank-Wolfe method (IF-...):

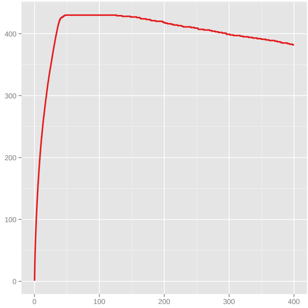
- IF-(0, ∞) – uses the away-step strategy and sets $\gamma_1 = 0$ and $\gamma_2 = +\infty$
- IF-Optimization – based on using the in-face optimization strategy (does not require setting γ_1, γ_2)
- IF-Rank-Strategy – uses the away-step strategy and adjusts the values of γ_1, γ_2 based on rank(Z^k)

We compare against regular Frank-Wolfe and two other away-step modified Frank-Wolfe algorithms

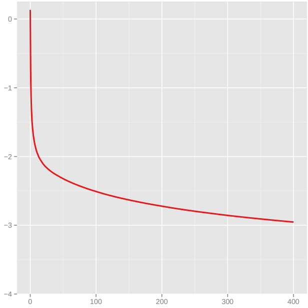
Example with $m = 2000$, $n = 2500$, 1% observed entries, and $\delta = 8.01$

Frank-Wolfe

rank(Z^k) vs. Time



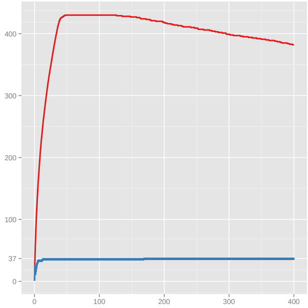
$\mathrm{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. Time



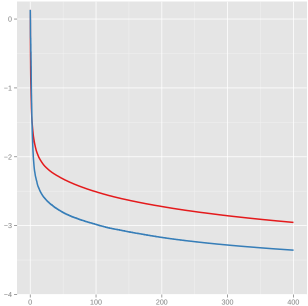
Example with $m = 2000$, $n = 2500$, 1% observed entries, and $\delta = 8.01$

IF-$(0, \infty)$

rank(Z^k) vs. Time



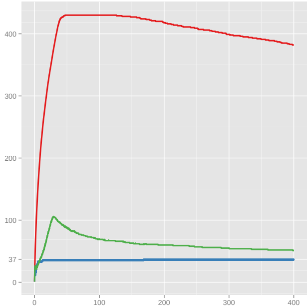
$\text{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. Time



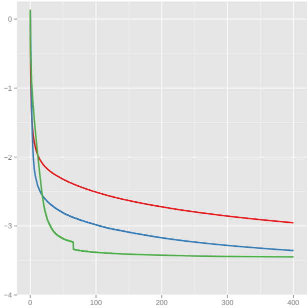
Example with $m = 2000$, $n = 2500$, 1% observed entries, and $\delta = 8.01$

IF-Optimization

rank(Z^k) vs. Time



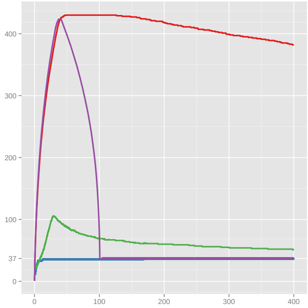
$\text{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. Time



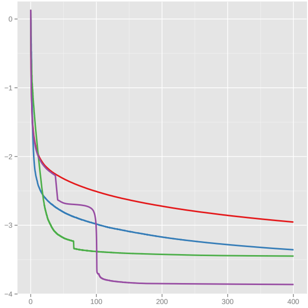
Example with $m = 2000$, $n = 2500$, 1% observed entries, and $\delta = 8.01$

IF-Rank-Strategy

rank(Z^k) vs. Time



$\mathrm{Log}_{10}\left(\frac{f(Z^k)-f^*}{f^*}\right)$ vs. Time



Small-Scale Examples (results averaged over 25 samples)

We generated artificial examples via the model $X =$ low-rank $+$ noise, controlling for:

- SNR – signal-to-noise ratio
- ρ – fraction of observed entries
- r – true underlying rank

Small-Scale Examples (25 samples per example)

Data	Metric	Regular FW	In-Face Extended FW (IF-...) γ_1, γ_2 1,1	0,1	0,∞	In-Face Opt.	Rank Strategy	Away Steps Natural	Atomic	Fully Corrective FW	CoGEnT
$m = 200, n = 400, \rho = 0.10$	Time (secs)	29.51	22.86	23.07	7.89	**2.34**	**2.30**	14.71	6.21	8.76	20.85
$r = 10$, SNR $= 5, \delta_{\text{avg}} = 3.75$	Final Rank	118.68	16.36	16.36	**16.44**	29.32	28.20	16.72	119.00	92.84	79.96
	Maximum Rank	146.48	19.04	17.28	**17.56**	32.08	145.20	18.04	121.96	991.60*	**
$m = 200, n = 400, \rho = 0.20$	Time (secs)	115.75	153.42	150.89	27.60	20.62	**3.48**	50.52	24.52	196.29	65.88
$r = 15$, SNR $= 4, \delta_{\text{avg}} = 3.82$	Final Rank	96.44	16.16	16.12	**16.52**	19.88	21.24	16.68	106.60	107.04	93.40
	Maximum Rank	156.52	26.72	17.96	**17.80**	31.48	160.36	18.84	106.80	1812.92*	**
$m = 200, n = 400, \rho = 0.30$	Time (secs)	171.23	198.96	202.01	35.93	31.67	**5.04**	66.22	67.72	>381.91	93.93
$r = 20$, SNR $= 3, \delta_{\text{avg}} = 3.63$	Final Rank	91.80	20.08	20.08	**20.60**	**21.72**	25.56	20.44	94.64	113.84	104.60
	Maximum Rank	162.24	25.80	22.04	**21.96**	33.36	168.72	22.16	95.08	1609.40*	**

Small-Scale Examples (results averaged over 25 samples)

		In-Face Extended FW (IF-...)		
		γ_1, γ_2	**In-Face**	**Rank**
Data	**Methods**	**0,∞**	**Opt.**	**Strat.**
$m = 200, n = 400, \rho = 0.10$	Time (secs)	7.89	**2.34**	**2.30**
$r = 10, \text{SNR} = 5, \delta_{\text{avg}} = 3.75$	Final Rank	**16.44**	29.32	28.20
	Maximum Rank	**17.56**	32.08	145.20
$m = 200, n = 400, \rho = 0.20$	Time (secs)	27.60	20.62	**3.48**
$r = 15, \text{SNR} = 4, \delta_{\text{avg}} = 3.82$	Final Rank	**16.52**	19.88	21.24
	Maximum Rank	**17.80**	31.48	160.36
$m = 200, n = 400, \rho = 0.30$	Time (secs)	35.93	31.67	**5.04**
$r = 20, \text{SNR} = 3, \delta_{\text{avg}} = 3.63$	Final Rank	**20.60**	**21.72**	25.56
	Maximum Rank	**21.96**	33.36	168.72

Small-Scale Examples (results averaged over 25 samples)

		In-Face Extended FW (IF-...)		
		γ_1, γ_2	In-Face	Rank
Data	Methods	**0,∞**	**Opt.**	**Strat.**
$m = 200, n = 400, \rho = 0.10$ $r = 10$, SNR $= 5, \delta_{\text{avg}} = 3.75$	Time (secs)	7.89	**2.34**	**2.30**
	Final Rank	**16.44**	29.32	28.20
	Maximum Rank	**17.56**	32.08	145.20
$m = 200, n = 400, \rho = 0.20$ $r = 15$, SNR $= 4, \delta_{\text{avg}} = 3.82$	Time (secs)	27.60	20.62	**3.48**
	Final Rank	**16.52**	19.88	21.24
	Maximum Rank	**17.80**	31.48	160.36
$m = 200, n = 400, \rho = 0.30$ $r = 20$, SNR $= 3, \delta_{\text{avg}} = 3.63$	Time (secs)	35.93	31.67	**5.04**
	Final Rank	**20.60**	**21.72**	25.56
	Maximum Rank	**21.96**	33.36	168.72

- IF-$(0, \infty)$ reliably always delivers a solution with the lowest rank reasonably quickly
- IF-Rank-Strategy delivers the best run times – beating existing methods by a factor of 10 or more
- IF-Rank-Strategy sometimes fails on large problems – IF-Optimization is more robust

MovieLens10M Dataset, $m = 69878$, $n = 10677$, $|\Omega| = 10^7$ (1.3% sparsity), and $\delta = 2.59$

MovieLens10M Dataset

Relative Optimality Gap	**Frank-Wolfe** Time (mins)	**Frank-Wolfe** Rank	**IF-(0, ∞)** Time (mins)	**IF-(0, ∞)** Rank
$10^{-1.5}$	7.38	103	7.01	44
10^{-2}	28.69	315	14.73	79
$10^{-2.25}$	69.53	461	22.80	107
$10^{-2.5}$	178.54	454	42.24	138

For this large-scale instance, we test IF-(0, ∞), which is most promising at delivering a low-rank solution, and benchmark against Frank-Wolfe

Summary

- Despite guarantees for $f(Z^k) - f^*$ and $\text{rank}(Z^k)$, the Frank-Wolfe method can fail at delivering a low-rank solution within a reasonable amount of time.
- In-face directions are a general methodological approach for preserving structure (low rank) while making objective function improvements
- Computational guarantees for In-Face Extended FW Method in terms of optimality gaps
- In the case of matrix completion, In-Face Extended FW
  - has computational guarantees in terms of improved bounds on the rank of the iterates
  - is able to efficiently deliver a low-rank solution reasonably quickly in practice
- Paper includes full computational evaluation on simulated and real data instances

Paper:

"An Extended Frank-Wolfe Method with 'In-Face' Directions, and its Application to Low-Rank Matrix Completion"

Available at `http://arxiv.org/abs/1511.02204`

Back-up: Medium-Large Scale Examples

Medium-Large Scale Examples

Data	Metric	Regular FW	In-Face Extended FW (IF-...) γ_1, γ_2: 1,1	0,1	0,∞	In-Face Opt.	Rank Strategy	Away Steps: Natural	Atomic	Fully Corrective FW
$m = 500, n = 1000, \rho = 0.25$ $r = 15, \mathrm{SNR} = 2, \delta = 3.57$	Time (secs) Final Rank (Max Rank)	137.62 53 (126)	51.95 16 (17)	53.21 15 (17)	18.20 16 (17)	**4.41** **17 (19)**	**6.37** 121 (136)	31.55 15 (17)	157.31 50 (52)	39.81 78 (984*)
$m = 500, n = 1000, \rho = 0.25$ $r = 15, \mathrm{SNR} = 10, \delta = 4.11$	Time (secs) Final Rank (Max Rank)	256.08 41 (128)	110.37 15 (17)	110.77 15 (17)	46.07 16 (17)	**6.76** **15 (18)**	**7.91** **18** (140)	73.95 16 (17)	322.24 48 (48)	227.50 81 (971*)
$m = 1500, n = 2000, \rho = 0.05$ $r = 15, \mathrm{SNR} = 2, \delta = 6.01$	Time (secs) Final Rank (Max Rank)	124.76 169 (210)	108.97 15 (18)	113.58 16 (17)	24.75 **16 (16)**	**11.09** 31 (44)	**12.71** 206 (206)	40.23 16 (16)	60.83 128 (138)	48.76 106 (736*)
$m = 1500, n = 2000, \rho = 0.05$ $r = 15, \mathrm{SNR} = 10, \delta = 8.94$	Time (secs) Final Rank (Max Rank)	>800.01 119 (266)	518.72 15 (17)	496.08 15 (17)	166.01 15 (17)	**21.90** **15 (23)**	**31.41** **15** (256)	309.58 15 (18)	407.22 172 (185)	>801.89 125 (790*)
$m = 2000, n = 2500, \rho = 0.01$ $r = 10, \mathrm{SNR} = 4, \delta = 7.92$	Time (secs) Final Rank (Max Rank)	105.44 436 (436)	45.39 37 (38)	36.47 35 (38)	**23.15** **37 (38)**	**20.07** 67 (107)	47.83 430 (430)	30.07 37 (39)	26.92 245 (276)	39.65 238 (502*)
$m = 2000, n = 2500, \rho = 0.05$ $r = 10, \mathrm{SNR} = 2, \delta = 5.82$	Time (secs) Final Rank (Max Rank)	99.84 68 (98)	51.90 10 (11)	48.26 10 (11)	18.79 **11 (11)**	**6.92** 13 (15)	**6.70** 94 (94)	30.37 10 (11)	89.09 52 (52)	55.11 62 (370*)
$m = 5000, n = 5000, \rho = 0.01$ $r = 10, \mathrm{SNR} = 4, \delta = 12.19$	Time (secs) Final Rank (Max Rank)	251.33 161 (162)	168.66 10 (24)	172.21 11 (18)	64.56 **11 (20)**	**26.25** 22 (34)	**17.70** 20 (112)	96.79 10 (16)	90.41 181 (182)	350.88 92 (616*)
$m = 5000, n = 7500, \rho = 0.01$ $r = 10, \mathrm{SNR} = 4, \delta = 12.19$	Time (secs) Final Rank (Max Rank)	272.19 483 (483)	107.19 33 (43)	116.58 34 (36)	**52.65** **32 (37)**	**54.02** 63 (123)	145.13 476 (476)	107.60 36 (42)	94.96 229 (298)	209.86 204 (331*)