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
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.

$\Omega$ 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

\[ P_r : \quad z^* := \min_{Z \in \mathbb{R}^{m \times n}} \quad \frac{1}{2} \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 \]
\[ \text{s.t.} \quad \text{rank}(Z) \leq r \]

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

Nuclear norm is: \( \|Z\|_N := \sum_{j=1}^{r} \sigma_j \)

where

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

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

Instead we will solve:

\[ \text{NN}_\delta : \quad f^* := \min_{Z \in \mathbb{R}^{m \times n}} \quad f(Z) := \frac{1}{2} \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 \]
\[ \text{s.t.} \quad \|Z\|_N \leq \delta \]
Relaxation of the “hard” problem is the nuclear norm regularized least squares problem:

\[
\text{NN}_\delta : \quad f^* := \min_{Z \in \mathbb{R}^{m \times n}} \ f(Z) := \frac{1}{2} \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2
\]

s.t. \( \|Z\|_N \leq \delta \)

The above is a convex optimization problem

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

- Think \( \ell_1 \) 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:

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

Solve the equivalent problem on spectrahedron:

\[
S_\delta : \quad f^* := \min_{Z,W,Y} f(Z) := \frac{1}{2} \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 \\
\text{s.t. } \begin{bmatrix} W & Z \\ Z^T & Y \end{bmatrix} \succeq 0 \\
\text{trace}(W) + \text{trace}(Y) & \leq 2\delta
\]
Nuclear Norm Regularized Low-Rank Matrix Completion, cont.

Nuclear Norm Regularized Matrix Completion [Fazel 2002]

\[ \text{NN}_\delta : \quad f^* := \min_{Z \in \mathbb{R}^{m \times n}} f(Z) := \frac{1}{2} \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 \]

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

Extensive work studying data generating mechanisms that ensure that optimal solutions of \( \text{NN}_\delta \) have low rank (e.g. [Candes and Recht 2009], [Candes and Tao 2010], [Recht et al. 2010], . . . )

It is imperative that the algorithm for \( \text{NN}_\delta \) reliably delivers solutions with good predictive performance and low rank after a reasonable amount of time.
**NN$_{\delta}$ Alignment with Frank-Wolfe Algorithm**

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

s.t. \[\|Z\|_N \leq \delta\]

$NN_{\delta}$ 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 $\sigma_1$ 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
Problem of interest is:

\[
\text{CP} : \quad f^* := \min_{x} f(x) \\
\text{s.t.} \quad x \in S
\]

- $S \subset \mathbb{R}^n$ 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.

\[ \text{CP} : \quad f^* := \min_{x} f(x) \quad \text{s.t.} \quad x \in S \]

At iteration \( k \) of the Frank-Wolfe method:
At iteration $k$ of the Frank-Wolfe method:

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

\[ \text{CP: } f^* := \min_{x} f(x) \quad \text{s.t. } x \in S \]

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 \( \tilde{x}_k \leftarrow \arg \min_{x \in S} \{\nabla f(x_k)^T x\} \).
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”**: \( \tilde{\alpha}_k = \frac{2}{k+2} \)

- **Exact line-search**: \( \tilde{\alpha}_k = \arg \min_{\alpha \in [0,1]} \{ f(x_k + \alpha(\tilde{x}_k - x_k)) \} \)

- **QA (Quadratic approximation) step-size**: 
  \[
  \tilde{\alpha}_k = \min \left\{ 1, \frac{-\nabla f(x_k)^T (\tilde{x}_k - x_k)}{L\|\tilde{x}_k - x_k\|^2} \right\}
  \]

- **Dynamic strategy**: determined by some history of optimality bounds, see [FG]

- **Simple averaging**: \( \tilde{\alpha}_k = \frac{1}{k+1} \)

- **Constant step-size**: \( \tilde{\alpha}_k = \bar{\alpha} \) for some given \( \bar{\alpha} \in [0, 1] \)
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:

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

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

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 \( \| s \|_* := \max_{\| x \| \leq 1} \{ s^T x \} \)

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

\[ \text{Diam}(S) := \max_{x, y \in S} \{ \| x - y \| \} \]

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

\[ \| \nabla f(x) - \nabla f(y) \|_* \leq L \| x - y \| \quad \text{for all } x, y \in S \]
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 \( \sim [W,GM] \)

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:

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

where \( \rho = \text{dist}(x^*, \partial S) \).
Frank-Wolfe For Low-Rank Matrix Completion

\[ \mathcal{NN}_\delta : \quad f^* := \min_{Z \in \mathbb{R}^{m \times n}} f(Z) := \frac{1}{2} \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 \]

s.t. \[ \|Z\|_N \leq \delta \]

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 \( \mathcal{NN}_\delta \) by [Jaggi and Sulovsk 2010], [Harchaoui, Juditsky, and Nemirovski 2012], [Mu et al. 2014], and [Rao, Shah, and Wright 2014]
\[ f^* := \min_{Z \in \mathbb{R}^{m \times n}} f(Z) := \frac{1}{2} \sum_{(i,j) \in \Omega} (Z_{ij} - X_{ij})^2 \]

s.t. \[ \|Z\|_N \leq \delta \]

As applied to \( NN_\delta \), at iteration \( k \) Frank-Wolfe computes

\[ \tilde{Z}^k \leftarrow \arg \min_{\|Z\|_N \leq \delta} \{ \nabla f(Z^k) \cdot Z \} \]

and updates:

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

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 $\mathcal{NN}_\delta$

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

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

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

What happens in practice?
Practical Behavior of Frank-Wolfe Applied to $NN_\delta$, cont.

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

Frank-Wolfe Applied to a Typical Instance of $NN_\delta$: 37 Iterations

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

$\text{Log}_{10} \left( \frac{f(Z^k) - f^*}{f^*} \right)$ vs. $k$
Instance with \( m = 2000, \; n = 2500 \) and \( \text{rank}(Z^*) = 37 \)

Frank-Wolfe Applied to a Typical Instance of \( NN_\delta \): \( \sim 450 \) Iterations

\[
\text{rank}(Z^k) \quad \text{vs.} \quad k
\]

\[
\log_{10} \left( \frac{f(Z^k) - f^*}{f^*} \right) \quad \text{vs.} \quad k
\]
Practical Behavior of Frank-Wolfe Applied to $NN_\delta$, cont.

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

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

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

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

![Graph 1](attachment:rank_zk_vs_k.png)

![Graph 2](attachment:log10_error_vs_k.png)
Theoretical bounds for Frank-Wolfe:

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

We propose an extension of Frank-Wolfe that:

- In theory has computational guarantees for \( f(Z^k) - f^* \) and \( \text{rank}(Z^k) \) 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 and better training error than Frank-Wolfe
Preview of In-Face Extended Frank-Wolfe Behavior

Preview of IF Extended FW (versus FW)

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

For this problem, \( \text{rank}(Z^*) = 37 \) \( (m = 2000, n = 2500) \)
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 $\mathcal{NN}_\delta$, 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 $x_k$. 
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
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 \( \delta \):
\[
\mathcal{B}(0, \delta) := \{ Z \in \mathbb{R}^{m \times n} : \|Z\|_N \leq \delta \}
\]

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

- $x^B_k$: full step to the relative boundary of $\mathcal{F}_S(x_k)$
- $x^A_k$: partial step that remains in the relative interior of $\mathcal{F}_S(x_k)$
- $x^R_k$: 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.

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

At iteration $k$:

1. Decide which of $x^A_k$, $x^B_k$, $x^R_k$ to accept as next iterate:

   1. (Go to a lower-dimensional face.) Set $x_{k+1} \leftarrow x^B_k$ if

      \[
      \frac{1}{f(x^B_k) - B_k} \geq \frac{1}{f(x_k) - B_k} + \frac{\gamma_1}{2C}.
      \]

   2. (Stay in current face.) Else, set $x_{k+1} \leftarrow x^A_k$ if

      \[
      \frac{1}{f(x^A_k) - B_k} \geq \frac{1}{f(x_k) - B_k} + \frac{\gamma_2}{2C}.
      \]

   3. (Do regular FW step and update lower bound.) Else, set $x_{k+1} \leftarrow x^R_k$. 

Computational Guarantee for Extended FW Method

In the first $k$ iterations, let:

\[ N_k^B = \text{number of steps to the boundary of the minimal face} \]
\[ N_k^A = \text{number of steps to the interior of the minimal face} \]
\[ N_k^R = \text{number of regular Frank-Wolfe steps} \]

\[ k = N_k^B + N_k^A + N_k^R \]

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 $D := \text{diam}(S)$. 
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
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 $\gamma_1, \gamma_2$)
- **IF-Rank-Strategy** – uses the away-step strategy and adjusts the values of $\gamma_1, \gamma_2$ based on $\text{rank}(Z^k)$

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

\[ \text{IF-}(0, \infty) \]

\[ \text{rank}(Z^k) \text{ vs. Time} \]

\[ \log_{10} \left( \frac{f(Z^k) - f^*}{f^*} \right) \text{ vs. Time} \]
Example with $m = 2000$, $n = 2500$, 1% observed entries, and $\delta = 8.01$

IF-Optimization

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

$\log_{10} \left( \frac{f(Z^k) - f^*}{f^*} \right)$ vs. Time
Example with $m = 2000, n = 2500, 1\%$ observed entries, and $\delta = 8.01$
Small-Scale Examples (results averaged over 25 samples)

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

- **SNR** – signal-to-noise ratio
- **$\rho$** – fraction of observed entries
- **$r$** – true underlying rank

<table>
<thead>
<tr>
<th>Data</th>
<th>Metric</th>
<th>Regular FW</th>
<th>In-Face Extended FW (IF-...)</th>
<th>Away Steps</th>
<th>Fully Corrective FW</th>
<th>CoGEnT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\gamma_1$, $\gamma_2$</td>
<td>In-Face Opt.</td>
<td>Rank Strategy</td>
<td>Natural</td>
<td>Atomic</td>
</tr>
<tr>
<td>$m = 200$, $n = 400$, $\rho = 0.10$</td>
<td>Time (secs)</td>
<td>29.51</td>
<td>22.86</td>
<td>23.07</td>
<td>7.89</td>
<td>$2.34$</td>
</tr>
<tr>
<td>$r = 10$, $\text{SNR} = 5$, $\delta_{2\times5} = 3.75$</td>
<td>Final Rank</td>
<td>118.68</td>
<td>16.36</td>
<td>16.36</td>
<td>$16.44$</td>
<td>29.32</td>
</tr>
<tr>
<td></td>
<td>Maximum Rank</td>
<td>146.48</td>
<td>19.04</td>
<td>17.28</td>
<td>$17.56$</td>
<td>32.08</td>
</tr>
<tr>
<td>$m = 200$, $n = 400$, $\rho = 0.20$</td>
<td>Time (secs)</td>
<td>115.75</td>
<td>153.42</td>
<td>150.89</td>
<td>27.60</td>
<td>20.62</td>
</tr>
<tr>
<td>$r = 15$, $\text{SNR} = 4$, $\delta_{2\times5} = 3.82$</td>
<td>Final Rank</td>
<td>96.44</td>
<td>16.16</td>
<td>16.12</td>
<td>$16.52$</td>
<td>19.88</td>
</tr>
<tr>
<td></td>
<td>Maximum Rank</td>
<td>156.52</td>
<td>26.72</td>
<td>17.96</td>
<td>$17.80$</td>
<td>31.48</td>
</tr>
<tr>
<td>$m = 200$, $n = 400$, $\rho = 0.30$</td>
<td>Time (secs)</td>
<td>171.23</td>
<td>198.96</td>
<td>202.01</td>
<td>35.93</td>
<td>31.67</td>
</tr>
<tr>
<td>$r = 20$, $\text{SNR} = 3$, $\delta_{2\times5} = 3.63$</td>
<td>Final Rank</td>
<td>91.80</td>
<td>20.08</td>
<td>20.08</td>
<td>$20.60$</td>
<td>21.72</td>
</tr>
<tr>
<td></td>
<td>Maximum Rank</td>
<td>162.24</td>
<td>25.80</td>
<td>22.04</td>
<td>$21.96$</td>
<td>33.36</td>
</tr>
</tbody>
</table>
## Small-Scale Examples (results averaged over 25 samples)

<table>
<thead>
<tr>
<th>Data</th>
<th>In-Face Extended</th>
<th>FW (IF-...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\gamma_1, \gamma_2$</td>
<td>0, $\infty$</td>
</tr>
<tr>
<td><strong>Time (secs)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m = 200, n = 400, \rho = 0.10$</td>
<td>7.89</td>
<td>2.34</td>
</tr>
<tr>
<td>$r = 10, \text{SNR} = 5, \delta_{\text{avg}} = 3.75$</td>
<td>16.44</td>
<td>29.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m = 200, n = 400, \rho = 0.20$</td>
<td>27.60</td>
<td>20.62</td>
</tr>
<tr>
<td>$r = 15, \text{SNR} = 4, \delta_{\text{avg}} = 3.82$</td>
<td>16.52</td>
<td>19.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m = 200, n = 400, \rho = 0.30$</td>
<td>35.93</td>
<td>31.67</td>
</tr>
<tr>
<td>$r = 20, \text{SNR} = 3, \delta_{\text{avg}} = 3.63$</td>
<td>20.60</td>
<td>21.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Small-Scale Examples (results averaged over 25 samples)

<table>
<thead>
<tr>
<th>Data</th>
<th>Time (secs)</th>
<th>Final Rank</th>
<th>Maximum Rank</th>
<th>In-Face Extended FW (IF-...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0,∞</td>
<td>Opt.</td>
<td>Rank Strat.</td>
</tr>
<tr>
<td></td>
<td>Time (secs)</td>
<td>7.89</td>
<td>2.34</td>
<td>2.30</td>
</tr>
<tr>
<td></td>
<td>Final Rank</td>
<td>16.44</td>
<td>29.32</td>
<td>28.20</td>
</tr>
<tr>
<td></td>
<td>Maximum Rank</td>
<td>17.56</td>
<td>32.08</td>
<td>145.20</td>
</tr>
<tr>
<td>$m = 200, n = 400, \rho = 0.10$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r = 10, SNR = 5, \delta_{avg} = 3.75$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time (secs)</td>
<td>27.60</td>
<td>20.62</td>
<td>3.48</td>
</tr>
<tr>
<td></td>
<td>Final Rank</td>
<td>16.52</td>
<td>19.88</td>
<td>21.24</td>
</tr>
<tr>
<td></td>
<td>Maximum Rank</td>
<td>17.80</td>
<td>31.48</td>
<td>160.36</td>
</tr>
<tr>
<td>$m = 200, n = 400, \rho = 0.20$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r = 15, SNR = 4, \delta_{avg} = 3.82$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m = 200, n = 400, \rho = 0.30$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r = 20, SNR = 3, \delta_{avg} = 3.63$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **IF-(0, ∞)** 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 $\delta = 2.59$

<table>
<thead>
<tr>
<th>Relative Optimality Gap</th>
<th>Frank-Wolfe Time (mins)</th>
<th>Rank</th>
<th>IF-(0, $\infty$) Time (mins)</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1.5}$</td>
<td>7.38</td>
<td>103</td>
<td>7.01</td>
<td>44</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>28.69</td>
<td>315</td>
<td>14.73</td>
<td>79</td>
</tr>
<tr>
<td>$10^{-2.25}$</td>
<td>69.53</td>
<td>461</td>
<td>22.80</td>
<td>107</td>
</tr>
<tr>
<td>$10^{-2.5}$</td>
<td>178.54</td>
<td>454</td>
<td>42.24</td>
<td>138</td>
</tr>
</tbody>
</table>

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

- Despite guarantees for $f(Z^k) - f^*$ and 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

<table>
<thead>
<tr>
<th>Data</th>
<th>Metric</th>
<th>Regular FW</th>
<th>In-Face Extended FW (IF-…)</th>
<th>Away Steps</th>
<th>Fully Corrective FW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1,1</td>
<td>0,1</td>
<td>0,∞</td>
</tr>
<tr>
<td>$m = 500, n = 1000, \rho = 0.25$</td>
<td>Time (secs)</td>
<td>137.62</td>
<td>51.95</td>
<td>53.21</td>
<td>18.20</td>
</tr>
<tr>
<td>$r = 15, \text{SNR} = 2, \delta = 3.57$</td>
<td>Final Rank (Max Rank)</td>
<td>53 (126)</td>
<td>16 (17)</td>
<td>15 (17)</td>
<td>16 (17)</td>
</tr>
<tr>
<td>$m = 500, n = 1000, \rho = 0.25$</td>
<td>Time (secs)</td>
<td>256.08</td>
<td>110.37</td>
<td>110.77</td>
<td>46.07</td>
</tr>
<tr>
<td>$r = 15, \text{SNR} = 10, \delta = 4.11$</td>
<td>Final Rank (Max Rank)</td>
<td>41 (128)</td>
<td>15 (17)</td>
<td>15 (17)</td>
<td>16 (17)</td>
</tr>
<tr>
<td>$m = 1500, n = 2000, \rho = 0.05$</td>
<td>Time (secs)</td>
<td>124.76</td>
<td>108.97</td>
<td>113.58</td>
<td>24.75</td>
</tr>
<tr>
<td>$r = 15, \text{SNR} = 2, \delta = 6.01$</td>
<td>Final Rank (Max Rank)</td>
<td>169 (210)</td>
<td>15 (18)</td>
<td>16 (17)</td>
<td>16 (16)</td>
</tr>
<tr>
<td>$m = 1500, n = 2000, \rho = 0.05$</td>
<td>Time (secs)</td>
<td>&gt;800.01</td>
<td>518.72</td>
<td>496.08</td>
<td>166.01</td>
</tr>
<tr>
<td>$r = 15, \text{SNR} = 10, \delta = 8.94$</td>
<td>Final Rank (Max Rank)</td>
<td>119 (266)</td>
<td>15 (17)</td>
<td>15 (17)</td>
<td>15 (17)</td>
</tr>
<tr>
<td>$m = 2000, n = 2500, \rho = 0.01$</td>
<td>Time (secs)</td>
<td>105.44</td>
<td>45.39</td>
<td>36.47</td>
<td>23.15</td>
</tr>
<tr>
<td>$r = 10, \text{SNR} = 4, \delta = 7.92$</td>
<td>Final Rank (Max Rank)</td>
<td>436 (436)</td>
<td>37 (38)</td>
<td>35 (38)</td>
<td>37 (38)</td>
</tr>
<tr>
<td>$m = 2000, n = 2500, \rho = 0.05$</td>
<td>Time (secs)</td>
<td>99.84</td>
<td>51.90</td>
<td>48.26</td>
<td>18.79</td>
</tr>
<tr>
<td>$r = 10, \text{SNR} = 2, \delta = 5.82$</td>
<td>Final Rank (Max Rank)</td>
<td>68 (98)</td>
<td>10 (11)</td>
<td>10 (11)</td>
<td>11 (11)</td>
</tr>
<tr>
<td>$m = 5000, n = 5000, \rho = 0.01$</td>
<td>Time (secs)</td>
<td>251.33</td>
<td>168.66</td>
<td>172.21</td>
<td>64.56</td>
</tr>
<tr>
<td>$r = 10, \text{SNR} = 4, \delta = 12.19$</td>
<td>Final Rank (Max Rank)</td>
<td>161 (162)</td>
<td>10 (24)</td>
<td>11 (18)</td>
<td>11 (20)</td>
</tr>
<tr>
<td>$m = 5000, n = 7500, \rho = 0.01$</td>
<td>Time (secs)</td>
<td>272.19</td>
<td>107.19</td>
<td>116.58</td>
<td>52.65</td>
</tr>
<tr>
<td>$r = 10, \text{SNR} = 4, \delta = 12.19$</td>
<td>Final Rank (Max Rank)</td>
<td>483 (483)</td>
<td>33 (43)</td>
<td>34 (36)</td>
<td>32 (37)</td>
</tr>
</tbody>
</table>