Optimization for Learning and Big Data

Donald Goldfarb

Department of IEOR
Columbia University

Department of Mathematics
Distinguished Lecture Series
Lecture 1. First-Order Methods for Convex Optimization

Lecture 2. Stochastic Quasi-Newton Methods for Machine Learning

Lecture 3. Optimization for Tensor Models
"In whatever happens in the world, one can find the concept of maximum or minimum; hence there is no doubt that all phenomena in nature can be explained via the maximum and minimum method..."

_Euler, Leonhard (1744)_
Lec 1. Unsupervised Learning

- Background - Foreground Separation in Videos
Lec 2. Supervised Learning

- Classification
Lec 3 Unsupervised Learning

- Topic Model

Diagram:
- Corpus
- Sports
- Science
- Politics
First-Order Methods for Convex Optimization

Convex Functions - Basic Definitions

Proximal Algorithms

Augmented Lagrangian Method (of Multipliers)

Alternating Direction Method of Multipliers (ADMM)

Conditional Gradient (Frank-Wolfe) Method
Convex Functions

- \( f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}, \quad \text{dom}(f) = \{x \in \mathbb{R}^n \mid f(x) < +\infty\} \)
- \( f \) is proper: \( \text{dom}(f) \neq \emptyset \)
- \( f \) is (strictly) convex if
  \[
  f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \quad \lambda \in [0, 1] \tag{\lambda \in (0,1))
  \]
- \( f \) is \( \mu \)-strongly convex if for every \( \lambda \in [0, 1] \)
  \[
  f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{\mu}{2} \lambda(1 - \lambda)\|x - y\|^2
  \]
- If \( f \in C^2 \), \( \mu I \leq \nabla^2 f(x) \leq LI \), then \( f \) is \( \mu \)-strongly convex and
  \[
  f(y) \geq f(x) + \nabla f^\top(x)(y - x) + \frac{\mu}{2}\|y - x\|
  \]
Non-smooth Convex Functions

For convex functions, **subgradient** take the place of gradients.

- \( v \) is a subgradient of \( f \) at \( x \) if
  \[
f(y) \geq f(x) + v^\top(y - x)
  \]
- Recall for \( f \in C^1 \), \( f(y) \geq f(x) + \nabla f(x)^\top(y - x) \)
- Subdifferential: \( \partial f(x) = \{ \text{all subgradients of } f \text{ at } x \} \)

\[
\text{Slope } v \\
f(y) \geq f(x) + \langle v, y - x \rangle
\]
Optimality for Non-smooth Convex Functions

∂f is a set-valued functions

Example:

\[ f(x) = \begin{cases} 
    x^2 & \text{if } x < 0 \\
    x & \text{if } x \geq 0 
\end{cases} \]

\[ \partial f(x) = \begin{cases} 
    2x & \text{if } x < 0 \\
    [0, 1] & \text{if } x = 0 \\
    1 & \text{if } x > 0 
\end{cases} \]

▶ x minimize \( f(x) \) \iff 0 \in \partial f(x). \]
Moreau Proximal Envelopes

- History: Moreau and Yosida (1960’s)

- Moreau Envelope: $f^\gamma(x) = \min_y \{ f(y) + \frac{1}{2\gamma} ||y - x||^2 \}$

- $f^\gamma(x) \leq f(x)$; $f^\gamma(x)$ is a regularized version of $f$

- $f^\gamma(x)$ has the same set of minimizer as $f(x)$
Moreau Proximity Operator

- Proximity Operator: \( \text{prox}_{\gamma f} : \mathbb{R}^n \rightarrow \mathbb{R}^n \) of \( \gamma f \), where \( \gamma > 0 \) is a scale factor in

\[
\text{prox}_{\gamma f}(x) = \arg\min_y \{ f(y) + \frac{1}{2\gamma} \| y - x \|^2 \} \tag{1}
\]

- The function in \( \{} \) in (1) is strongly convex and hence has a unique minimizer for every \( x \).

- \( \text{prox}_{\gamma f}(\cdot) \) is closer to minimizers of \( f(\cdot) \) (and \( f_{\gamma}(\cdot) \)) than \( x \).

- \( \tilde{f}(y) \equiv f(x) + \nabla f(x)^\top(y - x) \) linearizion of \( f(\cdot) \) at \( x \)

\[
\text{prox}_{\gamma \tilde{f}}(x) = x - \gamma \nabla f(x) \quad \text{gradient descent with step size} \ \gamma
Proximity Operators: Examples

- \( f = I_C(x) \), the indicator function for the convex set \( C \subseteq \mathbb{R}^n \)

\[
l_C(x) = \begin{cases} 
0 & \text{if } x \in C \\
+\infty & \text{if } x \notin C
\end{cases}
\]

\[
\text{prox}_f(x) = \text{argmin}_{y \in C} \| y - x \|^2 \quad \text{(projection of } x \text{ onto } C)\]

- \( f = \gamma |x| \)

\[
\text{prox}_{\gamma f}(x) = \text{soft}(x, \gamma) = \text{sgn}(x) \max(|x| - \gamma, 0)
\]

- Nuclear (trace) norm: \( \|X\|_* = \sum \) of singular values of \( X \).

Let SVD of \( X \) be \( U\Lambda V^\top \), then

\[
\text{prox}_{\gamma \|\cdot\|_*}(X) = U\tilde{\Lambda} V^\top, \quad \tilde{\Lambda}_{ii} = \text{soft}(\Lambda_{ii}, \gamma)
\]
Proximal Minimization

\[ x^{k+1} \leftarrow \text{prox}_{\gamma f}(x^k) \]  \hspace{1cm} (2)

- Minimizer \( x^* \) of \( \gamma f \) is a fixed point of \( \text{prox}_{\gamma f} \), i.e.
  \[ x^* = \text{prox}_{\gamma f}(x^*) \]

- \( \text{prox}_{\gamma f} = x - \gamma \nabla f\gamma(x) \) is a steepest descent step, with step length \( \gamma \) for minimizing the Moreau envelope.

- \( \text{w.r.t } f \), if \( f \in C^1 \), \( \text{prox}_{\gamma f} \) is equivalent to an implicit gradient (backward Euler) step.

- Iteration (2) converges to the set of minimizers of \( f \).
Proximal Gradient Method

Consider:

\[
\text{minimize } f(x) + g(x)
\]

where \( f : \mathbb{R}^n \to \mathbb{R}, \; f \in C^1, \; g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\} \) are both closed and proper convex functions.

Proximal gradient method

\[
x^{k+1} \leftarrow \text{prox}_{\alpha_k g} (x_k - \alpha_k \nabla f(x_k))
\]

Re-discovered in optimization, convex analysis, machine learning, signal processing, PDE, etc

- "Fixed-Point Continuation" (FPC)
- "Iterative Shrinkage Thresholding" (IST)
- "Forward-Backward Splitting" (FBS)

Let \( \tilde{f}(x) = f(x_k) + \nabla f(x_k) \top (x - x_k) \)

\[
x^{k+1} \leftarrow \text{prox}_{\alpha_k g} (\text{prox}_{\alpha_k \tilde{f}} (x_k))
\]
Unsupervised Learning: Proximal Gradient Method

- Recommendation Systems: Netflix problem
  
<table>
<thead>
<tr>
<th>Movies</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

17,000 movies, 500,000 customers, 100,000,000 ratings

Objective function value: $1,000,000
Unsupervised Learning: Proximal Gradient Method

- Netflix Problem $\Rightarrow$ Matrix Completion
  \[
  \min_X \{ \text{rank}(X) \mid \mathcal{P}_\Omega(X - M) = 0 \}
  \]

- Convex Relaxation
  (Candes and Recht, 2009) (Candes and Tao, 2009)

- Prox gradient method:
  \[
  \min \mu \|X\|_* + \frac{1}{2} \|\mathcal{P}_\Omega(X - M)\|_F^2
  \]
  \[
  Y^k \leftarrow X^k - \tau g(X^k)
  \]
  \[
  X^{k+1} \leftarrow S_{\tau\mu}(Y^k)
  \]

  where
  \[
  g(X) := \text{gradient of } \frac{1}{2} \|\mathcal{P}_\Omega(X - M)\|_F^2
  \]
  \[
  S_\nu(Y) := \text{matrix shrinkage operator}
  \]

  (Ma, G, Chen, 2009)
Augmented Lagrangian Methods

- Consider the linearly constrained problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

where \( f \) is a proper, lower semi-continuous, convex function.

- Augmented Lagrangian with penalty parameter \( \rho > 0 \)

\[
\mathcal{L}(x, \lambda; \rho) := f(x) + \lambda^\top (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2
\]

"augmentation"

- Augmented Lagrangian method (method of multipliers) (Hestenes, Powell - 1969)

\[
x_k = \arg\min_x \mathcal{L}(x, \lambda_{k-1}; \rho),
\]

\[
\lambda_k = \lambda_{k-1} + \rho(Ax_k - b).
\]
A Non-standard Derivation

- \( \min_x f(x) \) s.t. \( Ax = b \) ⇔ \( \min_x \max_\lambda \{ f(x) + \lambda^\top (Ax - b) \} \)

- To smooth \( \max_\lambda \{ f(x) + \lambda^\top (Ax - b) \} \), add a proximal term given an estimate \( \tilde{\lambda} \):

  \[
  \hat{\varphi}(x) := \max_\lambda \{ f(x) + \lambda^\top (Ax - b) - \frac{1}{2\rho} ||\lambda - \tilde{\lambda}||^2 \}
  \]

- Maximizing w.r.t. \( \lambda \) yields

  \[
  \hat{\lambda} = \tilde{\lambda} + \rho (Ax - b)
  \]

  and

  \[
  \min_x \{ f(x) + \tilde{\lambda}^\top (Ax - b) + \frac{\rho}{2} ||Ax - b||^2 \} = \mathcal{L}(x, \tilde{\lambda}; \rho).
  \]

- Extends immediately to nonlinear constraints \( c(x) = 0 \) or \( c(x) \geq 0 \), and explicit constraints \( \min_{x \in \Omega} \mathcal{L}(x, \tilde{\lambda}, \rho) \).
Another Non-standard Derivation

Consider a penalty method approach

$$\min_x f(x) + \frac{\rho}{2} ||Ax - b||_2^2$$

Bregman distance for convex $f(\cdot)$ between points $u$ and $v$ is

$$D_f^p(u, v) := f(u) - f(v) - \langle p, u - v \rangle$$
Another Non-standard Derivation (Cont.) \((\rho = 1)\)

- **Bregman iteration:**

  set \(x^0 \leftarrow 0, \ p^0 \leftarrow 0\)
  
  \[ x^{k+1} \leftarrow \text{argmin}_x D^p_f(x, x^k) + \frac{1}{2} \|Ax - b\|_2^2 \]
  
  \[ p^{k+1} \leftarrow p^k + A^\top (Ax^{k+1} - b) \]

- **Augmented Lagrangian method:**

  set \(x^0 \leftarrow 0, \ \lambda^0 \leftarrow 0\)
  
  \[ x^{k+1} \leftarrow \text{argmin}_x f(x) + \langle \lambda^k, Ax \rangle + \frac{1}{2} \|Ax - b\|_2^2 \]
  
  \[ \lambda^{k+1} \leftarrow \lambda^k + Ax^{k+1} - b \]

- **Augmented Lagrangian \iff Bregman** \(\{ p^k = -A^\top \lambda^k \}\)
Alternating Direction Method of Multipliers (ADMM)

- Long history: goes back to Gabay and Mercier, Glowinski and Marrocco, Lions and Mercier, and Passty etc.
- Variational problems in partial differential equations
- Maximal monotone operators
- Variational inequalities
- Nonlinear convex optimization
- Linear programming
- Nonsmooth $\ell_1$-minimization, compressive sensing
- Split-Bregman (Goldstein & Osher, 2009) 2139 citations, (Gabay & Mercier, 1976) 970 citations
Alternating Direction Method of Multipliers (ADMM)

Consider problems with a separable objective of the form

$$\min_{(x,z)} f(x) + h(z) \quad s.t. \quad Ax + Bz = c.$$ 

Standard augmented Lagrangian method minimizes

$$\mathcal{L}(x, z, \lambda; \rho) := f(x) + h(z) + \lambda^\top (Ax + Bz - c) + \frac{\rho}{2} ||Ax - Bz - c||_2^2$$

w.r.t. \((x, z)\) jointly.

In ADMM, minimize over \(x\) and \(z\) separately and sequentially:

$$x_k = \arg\min_x \mathcal{L}(x, z_{k-1}, \lambda_{k-1}; \rho_k);$$

$$z_k = \arg\min_z \mathcal{L}(x_k, z, \lambda_{k-1}; \rho_k);$$

$$\lambda_k = \lambda_{k-1} + \rho_k (Ax_k + Bz_k - c).$$
ADMM: A Simpler Form

▶ Consider the simpler problem
\[ \min_x f(x) + h(Ax) \iff \min_{(x,z)} f(x) + h(z) \text{ s.t. } Ax = z. \]

▶ In this case, the ADMM can be written as
\[
\begin{align*}
x_k &= \arg\min_x f(x) + \frac{\rho}{2} ||Ax - z_{k-1} - d_{k-1}||_2^2 \\
z_k &= \arg\min_z h(z) + \frac{\rho}{2} ||Ax_{k-1} - z - d_{k-1}||_2^2 \\
d_k &= d_{k-1} - (Ax_k - z_k)
\end{align*}
\]
sometimes called the "scaled version" of ADMM.

▶ Note \( z_k = \text{prox}_{h/\rho}(Ax_{k-1} - d_{k-1}) \) and is usually easy.

▶ Updating \( x_k \) may be hard: if \( f \) is not quadratic, may be as hard as the original problem.
Examples  \( \min F(x) \equiv f(x) + g(x) \)

- Compressed sensing (Lasso):
  \[
  \min \quad \rho \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2
  \]

- Matrix Rank Min:
  \[
  \min \quad \rho \|X\|_* + \frac{1}{2} \|A(X) - b\|_2^2
  \]

- Robust PCA:
  \[
  \min_{X,Y} \quad \|X\|_* + \rho \|Y\|_1 : X + Y = M
  \]

- Sparse Inverse Covariance Selection:
  \[
  \min \quad - \log \det(X) + \langle \Sigma, X \rangle + \rho \|X\|_1
  \]

- Group Lasso:
  \[
  \min \quad \rho \|x\|_{1,2} + \frac{1}{2} \|Ax - b\|_2^2
  \]
**Variable Splitting**

\[
\min f(x) + g(x) \iff \min f(x) + g(y) \text{ s.t. } x = y
\]

- **Augmented Lagrangian function:**

\[
\mathcal{L}(x, y; \lambda) := f(x) + g(y) - \langle \lambda, x - y \rangle + \frac{1}{2\mu} \|x - y\|^2
\]

- **ADMM**

\[
\begin{aligned}
 x^{k+1} &:= \arg \min_x \mathcal{L}(x, y^k; \lambda^k) \\
 y^{k+1} &:= \arg \min_y \mathcal{L}(x^{k+1}, y; \lambda^k) \\
 \lambda^{k+1} &:= \lambda^k - (x^{k+1} - y^{k+1})/\mu
\end{aligned}
\]
Symmetric ADMM $\Rightarrow$ Alternating Linearization Method

- Symmetric version

\[
\begin{align*}
    x^{k+1} &:= \arg\min_x \mathcal{L}(x, y^k; \lambda^k) \\
    \lambda^{k+\frac{1}{2}} &:= \lambda^k - (x^{k+1} - y^k)/\mu \\
    y^{k+1} &:= \arg\min_y \mathcal{L}(x^{k+1}, y; \lambda^{k+\frac{1}{2}}) \\
    \lambda^{k+1} &:= \lambda^{k+\frac{1}{2}} - (x^{k+1} - y^{k+1})/\mu
\end{align*}
\]

- Optimality conditions lead to (assuming $f$ and $g$ are smooth)

\[
\begin{align*}
    \lambda^{k+\frac{1}{2}} &= \nabla f(x^{k+1}), \\
    \lambda^{k+1} &= -\nabla g(y^{k+1})
\end{align*}
\]

- Alternating Linearization Method (ALM)

\[
\begin{align*}
    x^{k+1} &= \arg\min_x f(x) + g(y^k) + \langle \nabla g(y^k), x - y^k \rangle + \frac{1}{2\mu} \|x - y^k\|^2 \\
    y^{k+1} &= \arg\min_x f(x^{k+1}) + \langle \nabla f(x^{k+1}), y - x^{k+1} \rangle + \frac{1}{2\mu} \|x^{k+1} - y\|^2 + g(y)
\end{align*}
\]

- Gauss-Seidel like algorithm
Complexity Bound for ALM

**Theorem (G, Ma and Scheinberg, 2013)**

Assume $\nabla f$ and $\nabla g$ are Lipschitz continuous with constants $L(f)$ and $L(g)$. For $\mu \leq 1/ \max\{L(f), L(g)\}$, ALM satisfies

$$F(y^k) - F(x^*) \leq \frac{\|x^0 - x^*\|^2}{4\mu k}$$

- $O(1/\varepsilon)$ iterations for an $\varepsilon$-optimal solution ($f(x) - f(x^*) \leq \varepsilon$)

- Can we improve the complexity?
- Can we extend this result to ADMM?
Optimal Gradient Methods Lipschitz continuous $\nabla f$

- Classical gradient method
  \[ x^k = x^{k-1} - \tau_k \nabla f(x^{k-1}) \]
  
  Complexity $O(1/\epsilon)$

- Nesterov’s acceleration technique (1983)
  \[
  \begin{cases}
    x^k &:= y^{k-1} - \tau_k \nabla f(y^{k-1}) \\
    y^k &:= x^k + \frac{k-1}{k+2}(x^k - x^{k-1})
  \end{cases}
  \]
  
  Complexity $O(1/\sqrt{\epsilon})$

- Optimal first-order method; best one can get
ISTA and FISTA (Beck and Teboulle, 2009)

- Assume $g$ is smooth

\[
\min \ F(x) \equiv f(x) + g(x)
\]

- ISTA (Proximal gradient method) Complexity $O(1/\epsilon)$

\[
x^{k+1} := \arg \min_x Q_g(x, x^k)
\]

or equivalently

\[
x^{k+1} := \arg \min_x \tau f(x) + \frac{1}{2} \| x - (x^k - \tau \nabla g(x^k)) \|^2
\]

- Never minimize $g$

- Fast ISTA (FISTA) Complexity $O(1/\sqrt{\epsilon})$

\[
\begin{aligned}
x^k &:= \arg \min_x \tau f(x) + \frac{1}{2} \| x - (y^k - \tau \nabla g(y^k)) \|^2 \\
t_{k+1} &:= \left(1 + \sqrt{1 + 4 t_k^2}\right)/2 \\
y^{k+1} &:= x^k + \frac{t_k - 1}{t_{k+1}}(x^k - x^{k-1})
\end{aligned}
\]
Fast Alternating Linearization Method (FALM)

- **ALM** (symmetric ADMM)
  \[
  \begin{aligned}
  x^{k+1} &:= \arg \min_x Q_g(x, y^k) \\
  y^{k+1} &:= \arg \min_y Q_f(x^{k+1}, y)
  \end{aligned}
  \]

- Accelerate ALM in the same way as FISTA

- **Fast Alternating Linearization Method (FALM)**
  \[
  \begin{aligned}
  x^k &:= \arg \min_x Q_g(x, z^k) \\
  y^k &:= \arg \min_y Q_f(x^k, y) \\
  w^k &:= (x^k + y^k)/2 \\
  t_{k+1} &:= \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\
  z^{k+1} &:= w^k + \frac{1}{t_{k+1}}(t_k(y^k - w^{k-1}) - (w^k - w^{k-1}))
  \end{aligned}
  \]

- Computational effort at each iteration is almost unchanged
- Both \(f\) and \(g\) must be smooth; however, both are minimized
Theorem (G, Ma and Scheinberg, 2013)

Assume $\nabla f$ and $\nabla g$ are Lipschitz continuous with constants $L(f)$ and $L(g)$. For $\mu \leq 1/\max\{L(f), L(g)\}$, FALM satisfies

$$F(y^k) - F(x^*) \leq \frac{\|x^0 - x^*\|^2}{\mu(k + 1)^2}$$

Complexity $O(1/\sqrt{\epsilon})$ iterations for an $\epsilon$-optimal solution

Hence, optimal first-order method

- Applied to Total Variation denoising — outperforms split Bregman (Qin, G, Ma, 2013)
ALM with skipping steps

At $k$-th iteration of ALM-S:

- $x^{k+1} := \arg \min_x \mathcal{L}_\mu(x, y^k; \lambda^k)$
- If $F(x^{k+1}) > \mathcal{L}_\mu(x^{k+1}, y^k; \lambda^k)$, then $x^{k+1} := y^k$
- $y^{k+1} := \arg \min_y Q_f(y, x^{k+1})$
- $\lambda^{k+1} := \nabla f(x^{k+1}) - (x^{k+1} - y^{k+1})/\mu$
- Note that only $f$ is required to be smooth.
- If $\mu \leq 1/L(f)$, complexity $O(1/\epsilon)$; if $L(f)$ not known, use backtracking line search (Scheinberg, G, Bai 2014)
- FALM version has complexity $O(1/\sqrt{\epsilon})$.
- Applied to solve Sparse Inverse Covariance Selection (Scheinberg, Ma, G, 2010), Group Lasso (structured sparsity for breast cancer gene expression) (Qin, G, 2012)
Multiple Splitting Algorithm (MSA)

- Generalization from 2 to $K$ convex functions is possible, but non-convergence of ADMM for $K \geq 3$ has been shown.

- Consider

\[
\min \quad F(x) \equiv f(x) + g(x) + h(x)
\]

- ALM (symmetric ADMM)

\[
Q_{gh}(u, v, w) := f(u) + g(v) + \langle \nabla g(v), u - v \rangle + \|u - v\|^2 / 2\mu + h(w) + \langle \nabla h(w), u - w \rangle + \|u - w\|^2 / 2\mu.
\]

\[
\begin{aligned}
x^{k+1} &:= \arg \min \, Q_{gh}(x, y^k, z^k) \\
y^{k+1} &:= \arg \min \, Q_{fh}(x^{k+1}, y, z^k) \\
z^{k+1} &:= \arg \min \, Q_{fg}(x^{k+1}, y^{k+1}, z)
\end{aligned}
\]

- Gauss-Seidel like algorithm! Convergence?
Multiple Splitting Algorithm (MSA) (cont.)

- Jacobi type algorithm

\[
\begin{align*}
  x^{k+1} &:= \arg\min Q_{gh}(x, w^k, w^k) \\
y^{k+1} &:= \arg\min Q_{fh}(w^k, y, w^k) \\
z^{k+1} &:= \arg\min Q_{fg}(w^k, w^k, z) \\
w^{k+1} &:= (x^{k+1} + y^{k+1} + z^{k+1})/3
\end{align*}
\]

- Convergent

- Complexity \(O(1/\epsilon)\) (G and Ma, 2012)
$O(1/\sqrt{\epsilon})$ complexity (G and Ma, 2012)

Fast Multiple Splitting Algorithm (FaMSA)

\[
\begin{align*}
  x^k &:= \arg \min Q_{gh}(x, w^k_x, w^k_x) \\
  y^k &:= \arg \min Q_{fh}(w^k_y, y, w^k_y) \\
  z^k &:= \arg \min Q_{fg}(w^k_z, w^k_z, z) \\
  w^k &:= (x^k + y^k + z^k)/3 \\
  t_{k+1} &:= \left(1 + \sqrt{1 + 4t_k^2}\right)/2 \\
  w_{x}^{k+1} &:= w^k + \frac{1}{t_{k+1}} \left[t_k(x^k - w^k) - (w^k - w^{k-1})\right] \\
  w_{y}^{k+1} &:= w^k + \frac{1}{t_{k+1}} \left[t_k(y^k - w^k) - (w^k - w^{k-1})\right] \\
  w_{z}^{k+1} &:= w^k + \frac{1}{t_{k+1}} \left[t_k(z^k - w^k) - (w^k - w^{k-1})\right]
\end{align*}
\]
The Frank-Wolfe Algorithm

- Discovered in 1956, the Frank-Wolfe (also known as conditional gradient) algorithm is the earliest algorithm to solve:

\[
\text{minimize } f(x) \quad \text{subject to } x \in D
\]

where

- \( f(x) \) is a convex function
- \( D \subset \mathbb{R}^p \) is a compact and convex set.

Frank-Wolfe Algorithm

1. **Initialization:** \( x_0 \in D \)
2. **for** \( k = 0, 1, \ldots \) **do**
3. \( v_k = \arg \min_{x \in D} \langle v, \nabla f(x_k) \rangle \)
4. Set \( \gamma_k = \frac{2}{k+2} \) or by line search
5. \( x_{k+1} = x_k + \gamma_k (v_k - x_k) \),
6. **end for**
7. **Output:** \( N \).
The Frank-Wolfe Algorithm
Application: Signal Processing

- Recover a sparse signal $x$ from noisy measurements $b$

- Convex Relaxation $\Rightarrow$ Exact Recovery with high probability (Candes, Romberg and Tao, 2006; Donoho, 2006)

- Consider

  \[
  \min_{\|x\|_1 \leq 1} \|Ax - b\|^2
  \]

  Frank - Wolfe $\leftrightarrow$ Matching Pursuit

  Fully corrective Frank-Wolfe $\leftrightarrow$ Orthogonal Matching Pursuit (Tropp & Gilbert, 2007)
Application: Robust and Stable Principal Component Pursuit (RPCP and SPCP)

\[ M = L_0 + S_0 + N_0 \]

- low-rank
- sparse
- small, dense noise

- Given \( M \), approximately and efficiently recover \( L_0 \) and \( S_0 \).

- Convex approach

\[
\text{SPCP: } \min_{L,S} \|L\|_* + \lambda \|S\|_1 \text{ s.t. } \|L + S - M\|_F \leq \delta
\]

\[
\text{RPCP: } \min_{L,S} \|L\|_* + \lambda \|S\|_1 \text{ s.t. } L + S = M
\]
Algorithms for RPCP and SPCP

Many first-order methods have been developed

▶ Most exploit the closed-form expression for the proximal operator of nuclear norm; i.e. matrix shrinkage

$$\min_L \frac{1}{2} \| L - Z \|_2^2 + \lambda \| L \|_*$$

▶ Using a full or partial SVD, thus limiting their applicability to large-scale problems

▶ They also use the closed-form expression for the proximal operator of the $l_1$-norm; i.e. vector shrinkage to compute $S$. 
Frank-Wolfe for Norm-Constrained SPCP

- Solve

\[
\min_{L,S} \frac{1}{2} \| P_\Omega (L + S - M) \|_F^2 \\
\text{s.t.} \|L\|_* \leq \beta_1, \|S\|_1 \leq \beta_2
\]

- Frank-Wolfe algorithm for SPCP:

1: **Init:** \( L^0 = S^0 = 0 \);
2: for \( k = 0, 1, 2, \ldots \) do
3: \( D_L^k \in \arg \min_{\|D_L\|_* \leq 1} \langle P_\Omega [L^k + S^k - M], D_L \rangle \);
4: \( D_S^k \in \arg \min_{\|D_S\|_1 \leq 1} \langle P_\Omega [L^k + S^k - M], D_S \rangle \);
5: \( L^{k+1} = L^k + \frac{2}{k+2} (\beta_1 D_L^k - L^k) \);
6: \( S^{k+1} = S^k + \frac{2}{k+2} (\beta_2 D_S^k - S^k) \);
7: end for
Inefficiency of the FW algorithm

- Synthetic data: (Slow convergence)

- Inefficient in updateing $S$:

\[
S^{k+1} = \frac{k}{k+2} S^k - \frac{2\beta_2}{k+2} e^k_i (e^k_j)^\top \implies \|S^{k+1}\|_0 \leq \|S^k\|_0 + 1
\]
Frank-Wolfe/Prox Gradient (FW-P) Algorithm

- Key idea: Add a prox gradient step to update $S$ after each F-W step

```
1: Initialization: $L^0 = S^0 = 0$
2: for $k = 0, 1, 2, \cdots$ do
3:     $D_L^k \in \text{arg min}_{\|D_L\|_1 \leq 1} \langle \mathcal{P}_\Omega[L^k + S^k - M], D_L \rangle$
4:     $D_S^k \in \text{arg min}_{\|D_S\|_1 \leq 1} \langle \mathcal{P}_\Omega[L^k + S^k - M], D_S \rangle$
5:     $\gamma = \frac{2}{k+2}$
6:     $L^{k+\frac{1}{2}} = L^k + \gamma(\beta_1 D_L^k - L^k)$
7:     $S^{k+\frac{1}{2}} = S^k + \gamma(\beta_2 D_S^k - S^k)$
8:     $S^{k+1} = \mathcal{P}_{\|\cdot\|_1 \leq \beta_2} \left[ S^{k+\frac{1}{2}} - \mathcal{P}_\Omega[L^{k+\frac{1}{2}} + S^{k+\frac{1}{2}} - M] \right]$,
9:     $L^{k+1} = L^{k+\frac{1}{2}}$
10: end for
```
FW-P Algorithm for SPCP

- Solve \( \min_{L,S} \frac{1}{2} \| \mathcal{P}_\Omega [L + S - M] \|_F^2 + \lambda_1 \| L \|_* + \lambda_2 \| S \|_1 \)

- Domain unbounded \( \rightarrow \) Epigraph formulation!

\[
\begin{align*}
\min & \quad \frac{1}{2} \| \mathcal{P}_\Omega [L + S - M] \|_F^2 + \lambda_1 t_1 + \lambda_2 t_2 \\
\text{s.t.} & \quad \| L \|_* \leq t_1, \quad \| S \|_1 \leq t_2
\end{align*}
\]

\[
\begin{align*}
U_1 & \geq U_1^* := \| L^* \|_* \\
U_2 & \geq U_2^* := \| S^* \|_1
\end{align*}
\]

\[
\min \quad g(L, S, t_1, t_2) = \frac{1}{2} \| \mathcal{P}_\Omega [L + S - M] \|_F^2 + \lambda_1 t_1 + \lambda_2 t_2 \\
\text{s.t.} & \quad \| L \|_* \leq t_1 \leq U_1, \quad \| S \|_1 \leq t_2 \leq U_2
\]
FW-P Algorithm for SPCP

Synthetic data: (Red: F-W, Blue: UFA)

Theorem (Mu, Wright, G. 14)
For \( \{(L^k, S^k)\} \) produced by FW-P method, we have

\[
f(L^k, S^k) - f(L^*, S^*) \leq \frac{16(\beta_1^2 + \beta_2^2)}{k + 2}
\]
## FW-P Algorithm for SPCP

- Comparison with other algorithms

<table>
<thead>
<tr>
<th>Problem</th>
<th>$m$</th>
<th>$n$</th>
<th>FW-T iter.</th>
<th>FW-T cpu (s)</th>
<th>ISTA iter.</th>
<th>ISTA cpu</th>
<th>FISTA iter.</th>
<th>FISTA cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall</td>
<td>25344</td>
<td>200</td>
<td>6</td>
<td>3.93</td>
<td>30</td>
<td>21.1</td>
<td>14</td>
<td>12.0</td>
</tr>
<tr>
<td>Mall</td>
<td>81920</td>
<td>300</td>
<td>5</td>
<td>17.5</td>
<td>27</td>
<td>101</td>
<td>15</td>
<td>69.0</td>
</tr>
<tr>
<td>Escalator</td>
<td>20800</td>
<td>1000</td>
<td>6</td>
<td>16.2</td>
<td>13</td>
<td>44.0</td>
<td>10</td>
<td>45.2</td>
</tr>
<tr>
<td>Lobby</td>
<td>20480</td>
<td>1000</td>
<td>5</td>
<td>15.1</td>
<td>30</td>
<td>133</td>
<td>16</td>
<td>119</td>
</tr>
</tbody>
</table>
FW-P for Matrix SPCP

- Background and foreground extractions from greyscale surveillance videos

\[ M \approx L_0 + S_0 \]

- Each frame stacked as a column in
- Background
- Foreground

- \(256 \times 320 \times 800 \approx 65.5M\), 96 seconds using a laptop!
FW-P for Tensor SPCP

Convex program:

\[
\min_{\mathcal{L}, S} \frac{1}{2} \| \mathcal{P}_\Omega [\mathcal{X} - \mathcal{L} - S] \|_F + \lambda_1 \| \mathcal{X} \|_* + \lambda_2 \| S \|_1
\]
FW-P for Tensor SPCP

- Background segmentation for color videos:
  background modelling
  (50% missing entries)
- Data size: $128 \times 160 \times 3 \times 300 = 18.4M$, running time: 34 secs.