Optimization for Learning and Big Data

Donald Goldfarb

Department of IEOR
Columbia University

Department of Mathematics
Distinguished Lecture Series
May 17 - 19, 2016.

Lecture 1. First-Order Methods for Convex Optimization

Lecture 2. Stochastic Quasi-Newton Methods for Machine Learning

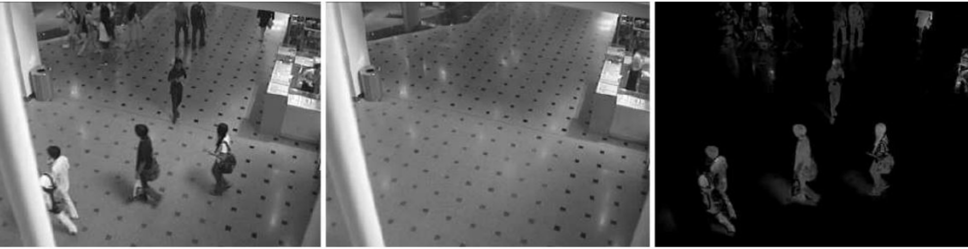
Lecture 3. Optimization for Tensor Models

”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)

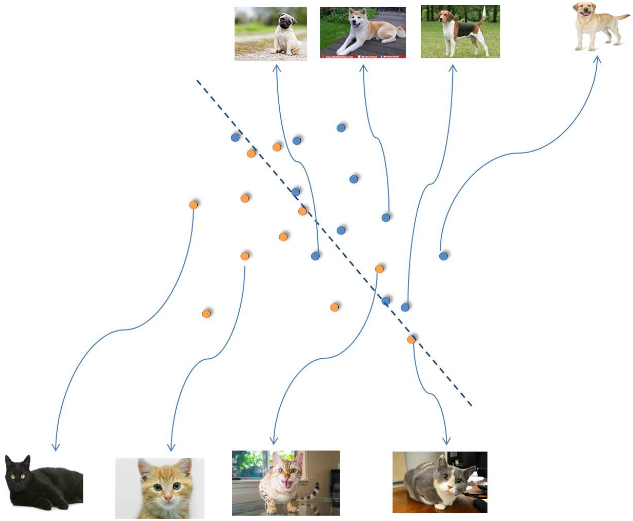
Lec 1. Unsupervised Learning

- Background - Foreground Separation in Videos



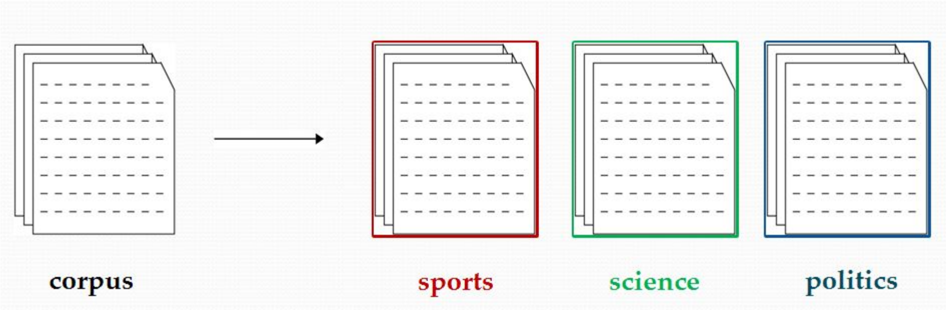
Lec 2. Supervised Learning

- Classification



Lec 3 Unsupervised Learning

- Topic Model



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

- $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $\quad \text{dom(f)} = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$

- f is proper: $\quad \text{dom}(f) \neq \emptyset$

- f is (strictly) convex if

$$f(\lambda x + (1-\lambda)y) \underset{(<)}{\leq} \lambda f(x) + (1-\lambda) f(y), \quad \underset{(\lambda \in (0,1))}{\lambda \in [0,1]}$$

- f is μ-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 μ-strongly convex and

$$f(y) \geq f(x) + \nabla f^\top(x)(y-x) + \frac{\mu}{2}||y-x||$$

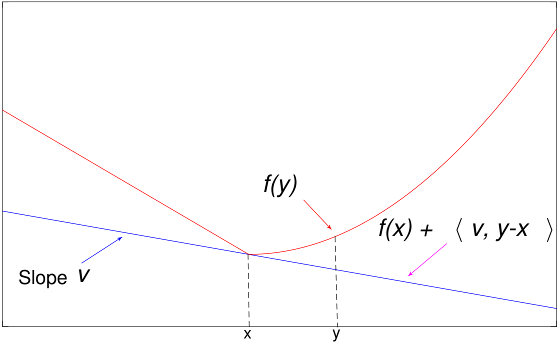
Non-smooth Convex Functions

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: $\partial f(x) = \{\text{all subgradients of f at x}\}$



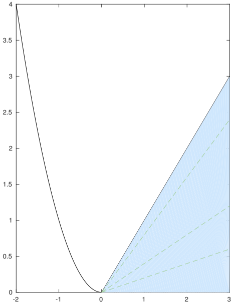
Optimality for Non-smooth Convex Functions

∂f is a set-valued functions

Example:

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

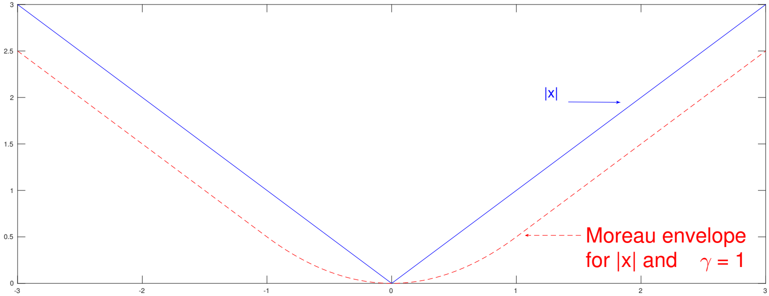
$$\partial f(x) = \begin{cases} 2x & \text{if } x < 0 \\ [0, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$



- x minimize $f(x) \quad \iff \quad 0 \in \partial f(x)$.

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: $\mathrm{prox}_{\gamma f} : \mathbb{R}^n \to \mathbb{R}^n$ of γf, where $\gamma > 0$ is a scale factor in

$$\mathrm{prox}_{\gamma f}(x) = \operatorname*{argmin}_y \{ f(y) + \frac{1}{2\gamma} ||y - x||^2 \} \qquad (1)$$

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

- $\mathrm{prox}_{\gamma f}(\cdot)$ 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

  $\mathrm{prox}_{\gamma \tilde{f}}(x) = x - \gamma \nabla f(x)$ gradient descent with step size γ

Proximity Operators: Examples

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

$$I_C(x) = \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{if } x \notin C \end{cases}$$

$$\text{prox}_f(x) = \operatorname*{argmin}_{y \in C} ||y - x||^2 \qquad \text{(projection of } x \text{ onto } C\text{)}$$

- $f = \gamma|x|$

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

- Nuclear (trace) norm: $||X||_* = \sum$ of singular values of X. Let SVD of X be $U \Lambda V^\top$, then

$$\text{prox}_{\gamma ||\cdot||_*}(X) = U \tilde{\Lambda} V^\top, \qquad \tilde{\Lambda}_{ii} = \text{soft}(\Lambda_{ii}, \gamma)$$

Proximal Minimization

$$x^{k+1} \leftarrow \mathrm{prox}_{\gamma f}(x^k) \tag{2}$$

- Minimizer x^* of γf is a fixed point of $\mathrm{prox}_{\gamma f}$, i.e. $x^* = \mathrm{prox}_{\gamma f}(x^*)$
- $\mathrm{prox}_{\gamma f} = x - \gamma \nabla f^{\gamma}(x)$, is a steepest descent step, with step length γ for minimizing the Moreau envelope.
- w.r.t f, if $f \in C^1$, $\mathrm{prox}_{\gamma f}$ is equivalent to an implicit gradient (backward Euler) step.
- Iteration (2) converges to the set of minimizers of f.

Proximal Gradient Method

- Consider:

$$\text{minimize} \quad f(x) + g(x)$$

  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

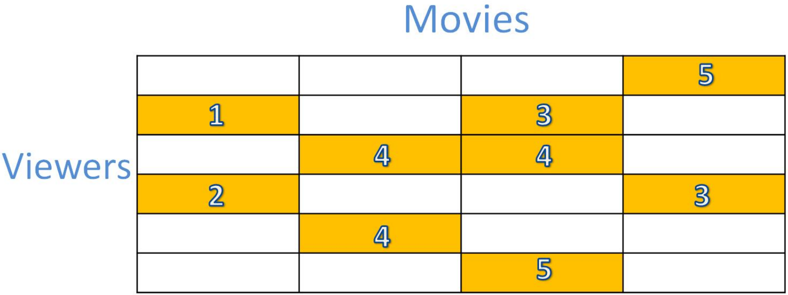
$$x^{k+1} \leftarrow \text{prox}_{\alpha_k g}(x_k - \alpha_k \nabla f(x_k))$$

- 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



Movies

Viewers

			5
1		3	
	4	4	
2			3
	4		
		5	

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

objective function **value**: $1,000,000

Unsupervised Learning: Proximal Gradient Method

- Netflix Problem ⇒ Matrix Completion

$$\min_X \{\mathrm{rank}(X) \,|\, \mathcal{P}_\Omega(X - M) = 0\}$$

- 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

$$g(X) := \text{gradient of } \frac{1}{2}\|\mathcal{P}_\Omega(X - M)\|_F^2$$

$$S_\nu(Y) := \text{matrix shrinkage operatior}$$

(Ma, G, Chen, 2009)

Augmented Lagrangian Methods

- Consider the linearly constrained problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

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

- Augmented Lagrangian with penalty parameter $\rho > 0$

$$\mathcal{L}(x, \lambda; \rho) := \underbrace{f(x) + \lambda^\top (Ax - b)}_{\text{Lagrangian}} + \underbrace{\frac{\rho}{2} ||Ax - b||_2^2}_{\text{"augmentation"}}$$

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

$$\begin{aligned} x_k &= \operatorname*{argmin}_x \mathcal{L}(x, \lambda_{k-1}; \rho), \\ \lambda_k &= \lambda_{k-1} + \rho(Ax_k - b). \end{aligned}$$

A Non-standard Derivation

- $\min_x f(x)$ s.t. $Ax = b \Leftrightarrow \min_x \max_\lambda \{f(x) + \lambda^\top (Ax - b)\}$
- To smooth $\max_\lambda \{f(x) + \lambda^\top (Ax - b)\}$, add a proximal term given an estimate $\bar{\lambda}$:

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

- Maximizing w.r.t. λ yields

$$\hat{\lambda} = \bar{\lambda} + \rho(Ax - b) \qquad \text{and}$$

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

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

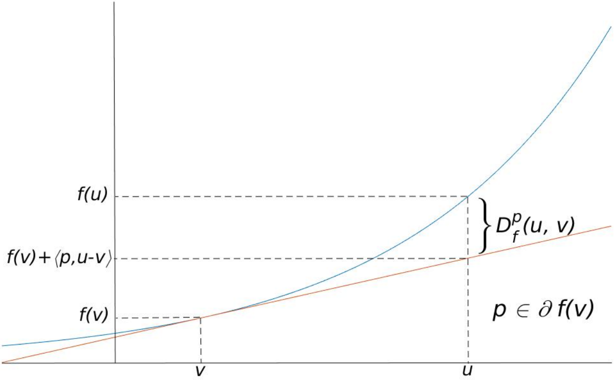
Another Non-standard Derivation

- 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:

$$\begin{aligned}
&\text{set } x^0 \leftarrow 0,\ p^0 \leftarrow 0 \\
&x^{k+1} \leftarrow \operatorname*{argmin}_x D_f^{p^k}(x, x^k) + \frac{1}{2}||Ax - b||_2^2 \\
&p^{k+1} \leftarrow p^k + A^\top(Ax^{k+1} - b)
\end{aligned}$$

- Augmented Lagrangian method:

$$\begin{aligned}
&\text{set } x^0 \leftarrow 0,\ \lambda^0 \leftarrow 0 \\
&x^{k+1} \leftarrow \operatorname*{argmin}_x f(x) + \langle \lambda^k, Ax \rangle + \frac{1}{2}||Ax - b||_2^2 \\
&\lambda^{k+1} \leftarrow \lambda^k + Ax^{k+1} - b
\end{aligned}$$

- Augmented Lagrangian $\Longleftrightarrow$ 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 ℓ_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:

$$\begin{aligned} x_k &= \operatorname*{argmin}_x \mathcal{L}(x, z_{k-1}, \lambda_{k-1}; \rho_k); \\ z_k &= \operatorname*{argmin}_z \mathcal{L}(x_k, z, \lambda_{k-1}; \rho_k); \\ \lambda_k &= \lambda_{k-1} + \rho_k (Ax_k + Bz_k - c). \end{aligned}$$

ADMM: A Simpler Form

- Consider the simpler problem $\min_x f(x) + h(Ax) \Longleftrightarrow \min_{(x,z)} f(x) + h(z)\ s.t.\ Ax = z.$
- In this case, the ADMM can be written as

$$\begin{aligned} x_k &= \operatorname*{argmin}_x f(x) + \frac{\rho}{2}||Ax - z_{k-1} - d_{k-1}||_2^2 \\ z_k &= \operatorname*{argmin}_z h(z) + \frac{\rho}{2}||Ax_{k-1} - z - d_{k-1}||_2^2 \\ d_k &= d_{k-1} - (Ax_k - z_k) \end{aligned}$$

  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:

$$\min \quad \rho\|X\|_* + \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2$$

- Robust PCA:

$$\min_{X,Y} \quad \|X\|_* + \rho\|Y\|_1 : X + Y = M$$

- Sparse Inverse Covariance Selection:

$$\min -\log\det(X) + \langle \Sigma, X \rangle + \rho\|X\|_1$$

- Group Lasso:

$$\min \quad \rho\|x\|_{1,2} + \frac{1}{2}\|Ax - b\|_2^2$$

Variable Splitting

$$\min \quad f(x) + g(x) \Longleftrightarrow \min f(x) + g(y) \text{ s.t. } x = y$$

- 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

$$\left\{ \begin{array}{lcl} x^{k+1} & := & \arg\min_x \mathcal{L}(x, y^k; \lambda^k) \\ y^{k+1} & := & \arg\min_y \mathcal{L}(x^{k+1}, y; \lambda^k) \\ \lambda^{k+1} & := & \lambda^k - (x^{k+1} - y^{k+1})/\mu \end{array} \right.$$

Symmetric ADMM $\Rightarrow$ Alternating Linearization Method

- Symmetric version

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

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

$$\lambda^{k+\frac{1}{2}} = \nabla f(x^{k+1}), \qquad \lambda^{k+1} = -\nabla g(y^{k+1})$$

- Alternating Linerization Method (ALM)

$$\begin{cases} x^{k+1} = \arg\min_x f(x) + g(y^k) + \langle \nabla g(y^k), x - y^k \rangle + \frac{1}{2\mu}\|x - y^k\|^2 \\ y^{k+1} = \arg\min_x f(x^{k+1}) + \langle \nabla f(x^{k+1}), y - x^{k+1} \rangle + \frac{1}{2\mu}\|x^{k+1} - y\|^2 + g(y) \end{cases}$$

- Gauss-Seidel like algorithm

Complexity Bound for ALM

Theorem (G, Ma and Scheinberg, 2013)

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

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

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

- Can we improve the complexity ?
- Can we extend this result to ADMM ?

Optimal Gradient Methods Lipschitz continuous ∇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)

$$\left\{ \begin{array}{lcl} x^k & := & y^{k-1} - \tau_k \nabla f(y^{k-1}) \\ y^k & := & x^k + \frac{k-1}{k+2}(x^k - x^{k-1}) \end{array} \right.$$

  Complexity $O(1/\sqrt{\epsilon})$

- Optimal first-order method; best one can get

ISTA and FISTA (Beck and Teboulle, 2009)

- Assume g is smooth

$$\min \quad F(x) \equiv f(x) + g(x)$$

- 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 $O(1/\sqrt{\epsilon})$

$$\left\{\begin{array}{lcl} x^k & := & \arg\min_x \tau f(x) + \frac{1}{2}\|x - (y^k - \tau\nabla g(y^k))\|^2 \\ t_{k+1} & := & \left(1 + \sqrt{1 + 4t_k^2}\right)/2 \\ y^{k+1} & := & x^k + \frac{t_k - 1}{t_{k+1}}(x^k - x^{k-1}) \end{array}\right.$$

Fast Alternating Linearization Method (FALM)

- ALM (symmetric ADMM)

$$\begin{cases} x^{k+1} & := & \arg\min_x Q_g(x, y^k) \\ y^{k+1} & := & \arg\min_y Q_f(x^{k+1}, y) \end{cases}$$

- Accelerate ALM in the same way as FISTA
- Fast Alternating Linearization Method (FALM)

$$\begin{cases} x^k & := & \arg\min_x Q_g(x, z^k) \\ y^k & := & \arg\min_y Q_f(x^k, y) \\ w^k & := & (x^k + y^k)/2 \\ t_{k+1} & := & \left(1 + \sqrt{1 + 4t_k^2}\right)/2 \\ z^{k+1} & := & w^k + \frac{1}{t_{k+1}}(t_k(y^k - w^{k-1}) - (w^k - w^{k-1})) \end{cases}$$

- computational effort at each iteration is almost unchanged
- both f and g must be smooth; however, both are minimized

FALM (cont.)

Theorem (G, Ma and Scheinberg, 2013)

Assume ∇f and ∇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 ϵ-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)

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

$$\left\{ \begin{array}{lcl} x^{k+1} & := & \arg\min Q_{gh}(x, y^k, z^k) \\ y^{k+1} & := & \arg\min Q_{fh}(x^{k+1}, y, z^k) \\ z^{k+1} & := & \arg\min Q_{fg}(x^{k+1}, y^{k+1}, z) \end{array} \right.$$

- Gauss-Seidel like algorithm! Convergence ?

Multiple Splitting Algorithm (MSA) (cont.)

- Jacobi type algorithm

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

- Convergent
- Complexity $O(1/\epsilon)$ (G and Ma, 2012)

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

- Fast Multiple Splitting Algorithm (FaMSA)

$$\begin{cases} x^k & := & \arg\min Q_{gh}(x, w_x^k, w_x^k) \\ y^k & := & \arg\min Q_{fh}(w_y^k, y, w_y^k) \\ z^k & := & \arg\min Q_{fg}(w_z^k, w_z^k, z) \\ w^k & := & (x^k + y^k + z^k)/3 \\ t_{k+1} & := & \left(1 + \sqrt{1 + 4t_k^2}\right)/2 \\ w_x^{k+1} & := & w^k + \frac{1}{t_{k+1}}[t_k(x^k - w^k) - (w^k - w^{k-1})] \\ w_y^{k+1} & := & w^k + \frac{1}{t_{k+1}}[t_k(y^k - w^k) - (w^k - w^{k-1})] \\ w_z^{k+1} & := & w^k + \frac{1}{t_{k+1}}[t_k(z^k - w^k) - (w^k - w^{k-1})] \end{cases}$$

The Frank-Wolfe Algorithm

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

$$\text{minimize} \quad f(x) \qquad \text{subject to} \quad x \in \mathcal{D}$$

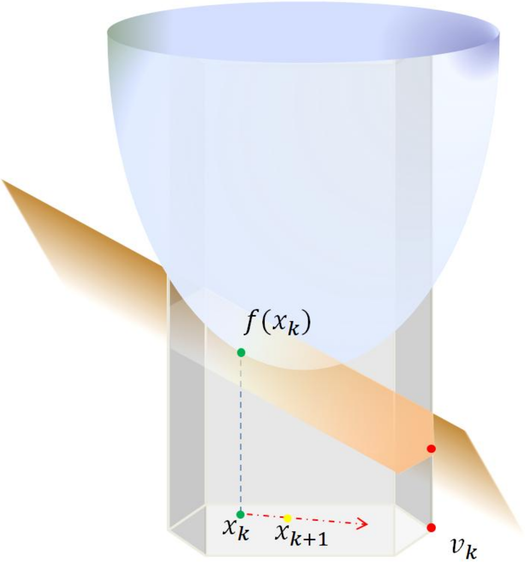
where

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

Frank-Wolfe Algorithm

```
**Initialization:** $x_0 \in \mathcal{D}$
**for** k = 0, 1, ... **do**
    $v_k = \arg\min_{x \in \mathcal{D}} \langle v, \nabla f(x_k) \rangle$
    Set $\gamma_k = \frac{2}{k+2}$ or by line search
    $x_{k+1} = x_k + \gamma_k (v_k - x_k)$,
**end for**
**Output:** $N$.
```

The Frank-Wolfe Algorithm



Application:Signal Processing

- Recover a sparse signal x from noisy measurements b
- Convex Relaxation ⇒ Exact Recovery with high probability (Candes, Romberg and Tao, 2006; Donoho, 2006)
- Consider

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

$$\text{Frank - Wolfe} \xLeftrightarrow[\text{vertex each step}]{\text{select same}} \text{Matching Pursuit}$$

$$\text{Fully corective Frank-Wolfe} \Longleftrightarrow \text{Orthogonal Matching Pursuit}$$

(Tropp & Gilbert, 2007)

Application: Robust and Stable Principal Component Pursuit (RPCP and SPCP)

$$M = \underset{\text{low-rank}}{L_0} + \underset{\text{sparse}}{S_0} + \underset{\text{small, dense noise}}{N_0}$$

- 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

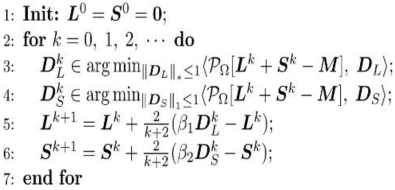
- They also us the closed-form expression for the proximal operator of the l_1-norm; i.e. vector shrinkage to compute S.

Frank-Wolfe for Norm-Constrained SPCP

- Solve

$$\min_{L,S} \frac{1}{2}\|\mathcal{P}_\Omega(L+S-M)\|_F^2$$
$$\text{s.t.}\|L\|_* \le \beta_1, \|S\|_1 \le \beta_2$$

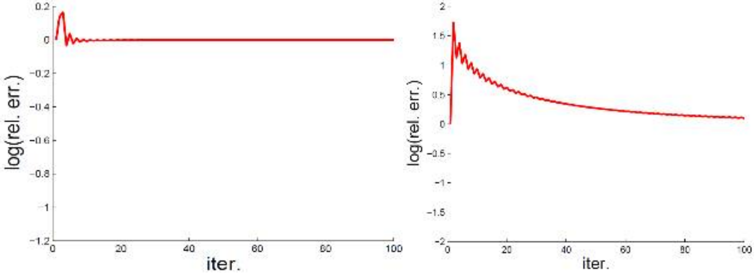
- Frank-Wolfe algorithm for SPCP:



```
Init: $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$;
for $k = 0,\ 1,\ 2,\ \cdots$ do
    $\boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \langle \mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L \rangle$;
    $\boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \le 1} \langle \mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S \rangle$;
    $\boldsymbol{L}^{k+1} = \boldsymbol{L}^k + \frac{2}{k+2}(\beta_1 \boldsymbol{D}_L^k - \boldsymbol{L}^k)$;
    $\boldsymbol{S}^{k+1} = \boldsymbol{S}^k + \frac{2}{k+2}(\beta_2 \boldsymbol{D}_S^k - \boldsymbol{S}^k)$;
end for
```

Inefficiency of the FW algorithm

- Synthetic data: (Slow convergence)

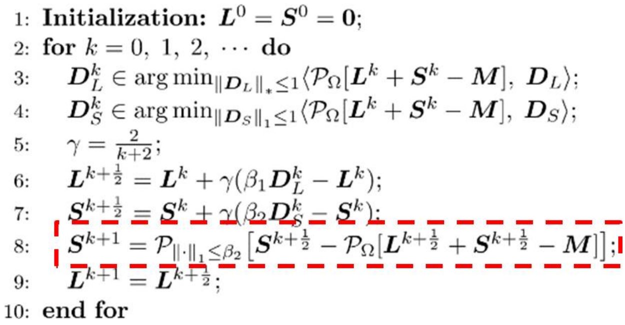


- Inefficient in updateing S:

$$S^{k+1} = \frac{k}{k+2} S^k - \frac{2\beta_2}{k+2} e^k_{i^*} (e^k_{j^*})^\top \Longrightarrow \|S^{k+1}\|_0 \leq \|S^k\|_0 + 1$$

Frank-Wolfe/Prox Gradient (FW-P) Algorithm

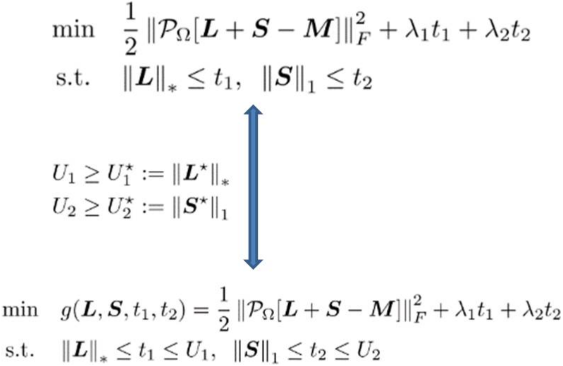
- Key idea: Add a prox gradient step to update S after each F-W step



```
**Initialization:** $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$;
**for** $k = 0,\ 1,\ 2,\ \cdots$ **do**
    $\boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \langle \mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L \rangle$;
    $\boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \le 1} \langle \mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S \rangle$;
    $\gamma = \frac{2}{k+2}$;
    $\boldsymbol{L}^{k+\frac{1}{2}} = \boldsymbol{L}^k + \gamma(\beta_1 \boldsymbol{D}_L^k - \boldsymbol{L}^k)$;
    $\boldsymbol{S}^{k+\frac{1}{2}} = \boldsymbol{S}^k + \gamma(\beta_2 \boldsymbol{D}_S^k - \boldsymbol{S}^k)$;
    $\boldsymbol{S}^{k+1} = \mathcal{P}_{\|\cdot\|_1 \le \beta_2}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_\Omega[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}]\right]$;
    $\boldsymbol{L}^{k+1} = \boldsymbol{L}^{k+\frac{1}{2}}$;
**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 → Epigraph formulation !



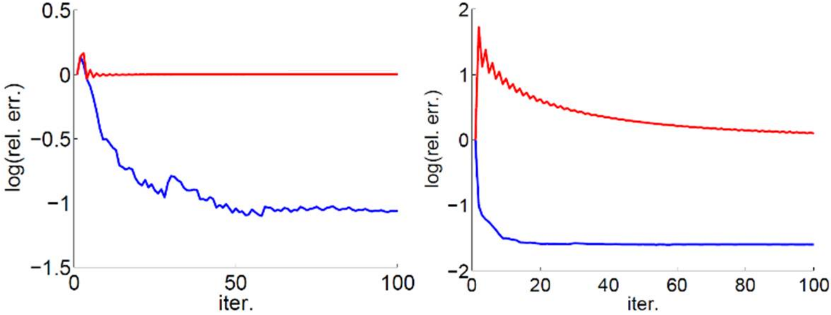
$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathcal{P}_\Omega[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\|_F^2 + \lambda_1 t_1 + \lambda_2 t_2 \\ \text{s.t.} \quad & \|\boldsymbol{L}\|_* \le t_1, \quad \|\boldsymbol{S}\|_1 \le t_2 \end{aligned}$$

$$U_1 \ge U_1^\star := \|\boldsymbol{L}^\star\|_*$$
$$U_2 \ge U_2^\star := \|\boldsymbol{S}^\star\|_1$$

$$\begin{aligned} \min \quad & g(\boldsymbol{L},\boldsymbol{S},t_1,t_2) = \frac{1}{2}\|\mathcal{P}_\Omega[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\|_F^2 + \lambda_1 t_1 + \lambda_2 t_2 \\ \text{s.t.} \quad & \|\boldsymbol{L}\|_* \le t_1 \le U_1, \quad \|\boldsymbol{S}\|_1 \le t_2 \le U_2 \end{aligned}$$

FW-P Algorithm for SPCP

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



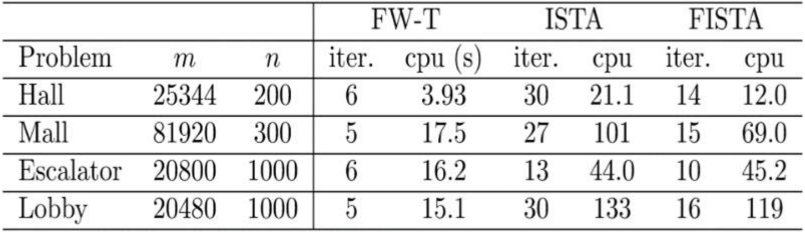
Theorem (Mu, Wright, G. 14)

For $\{(L^k, S^k)\}$ produced by FW-P method, we have

$$f(L^k, S^k) - f(L^*, S^*) \le \frac{16(\beta_1^2 + \beta_2^2)}{k+2}$$

FW-P Algorithm for SPCP

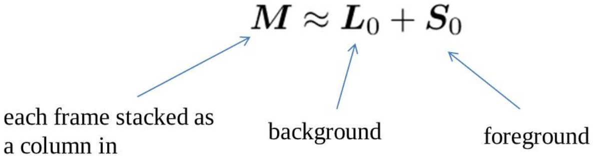
- Comparison with other algorithms



			FW-T		ISTA		FISTA	
Problem	m	n	iter.	cpu (s)	iter.	cpu	iter.	cpu
Hall	25344	200	6	3.93	30	21.1	14	12.0
Mall	81920	300	5	17.5	27	101	15	69.0
Escalator	20800	1000	6	16.2	13	44.0	10	45.2
Lobby	20480	1000	5	15.1	30	133	16	119

FW-P for Matrix SPCP

- Background and foreground extractions from greyscale surveillance videos



$$\boldsymbol{M} \approx \boldsymbol{L}_0 + \boldsymbol{S}_0$$

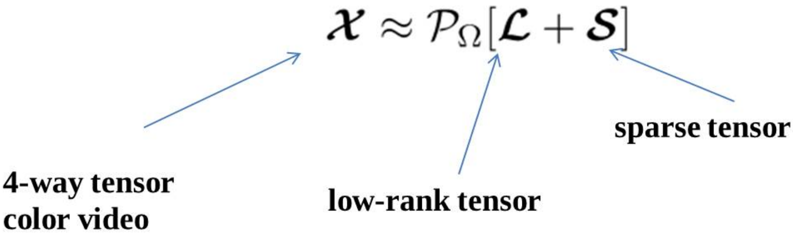
each frame stacked as a column in

background

foreground

- $256 \times 320 \times 800 \approx 65.5M$, 96 seconds using a laptop!

FW-P for Tensor SPCP



$$\boldsymbol{\mathcal{X}} \approx \mathcal{P}_\Omega[\boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{S}}]$$

4-way tensor
color video

low-rank tensor

sparse tensor

- Comvex program:

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

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.