

TFOCS: Flexible First-order Methods for Rank Minimization

Stephen Becker

California Institute of Technology

May 18, 2011

Collaborators:

Michael Grant (Caltech, CVX Research)

Emmanuel Candès (Stanford)

Typical problems: matrix completion

Matrix completion

$$\min_X \|X\|_{\text{tr}} \quad \text{such that} \quad \mathcal{A}(X) = b, X \in \mathbb{R}^{n_1 \times n_2}.$$

$\|X\|_{\text{tr}}$ is the **nuclear norm** (sum of singular values).

$\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \rightarrow \mathbb{R}^m$ is linear

$$\mathcal{A}(X) = \begin{bmatrix} \times & ? & ? & \times & ? \\ ? & \times & ? & \times & \times \\ \times & \times & ? & ? & ? \\ ? & \times & \times & \times & ? \\ ? & ? & \times & ? & \times \end{bmatrix}$$

If $m \ll n_1 \times n_2$, want prior on X . Convenient prior: X is **low-rank**.

Variants:

$$\min_X \|X\|_{\text{tr}} \quad \text{such that} \quad \|\mathcal{A}(X) - b\|_2 \leq \varepsilon$$

$$\min_X \|X\|_{\text{tr}} + \tau \|\mathcal{A}(X) - b\|_2^2$$

Typical problems: RPCA

Robust PCA (one type):

RPCA

$$\min_{L,S} \|L\|_{\text{tr}} + \lambda \|S\|_1 \quad \text{such that} \quad L + S = X, \mathcal{A}(X) = b$$

Idea: X is composed of **L**ow-rank and **S**parsity

May use $\mathcal{A} = I$

variants, e.g. AWGN noise:

Stable Principal Component Pursuit

$$\min_{L,S} \|L\|_{\text{tr}} + \lambda \|S\|_1 \quad \text{such that} \quad \|\mathcal{A}(X) - b\|_2 \leq \varepsilon$$

or constraints appropriate for quantization error (e.g. $[0, 255]$ indexed image):

$$\|\mathcal{A}(X) - b\|_{\infty} \leq \varepsilon$$

Example of RPCA

Background subtraction

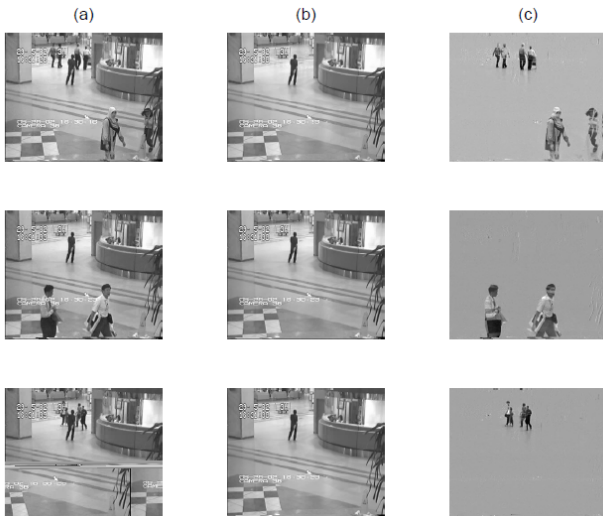


image from Goldfarb, Ma, Sheinberg '10

Typical problems: sparse covariance selection

$\hat{\Sigma}$ is sample covariance matrix, $x \sim \mathcal{N}(0, \Sigma)$.

$(\Sigma^{-1})_{i,j} = 0 \implies x_i, x_j$ conditionally independent

Sparse Covariance Selection

$$X^* = \underset{X}{\operatorname{argmin}} -\log \det X + \langle \hat{\Sigma}, X \rangle + \rho \|X\|_{\ell_1} \quad \text{such that} \quad X \succeq 0$$

If $\rho = 0$, then $X^* = \Sigma^{-1}$.

Fancier: adding latent variables (Chandrasekaran, Parrilo, Willsky), which adds new constraint and trace norm.

Existing approaches for RPCA

IPM: too slow (cf. Defeng Sun's talk)

First-order methods for RPCA (all 2008 – 2011)

Method	Name of code (Authors)
ADMM	LRSD (Yuan, Yang)
Non-convex ADMM	LMaFit (Shen, Wen, Zhang)
Fast ADMM	(Goldfarb, Ma, Sheinberg)
ADMM	IALM, Perceptions Lab at UIUC (Lin, Chen, Ma)
GP on dual	(Lin, Ganesh, Wright, Wu, Chen, Ma)
AGP on primal	Lin, Ganesh, Wright, Wu, Chen, Ma)

Augmented Lagrangian and variants

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad \text{such that} \quad L + S = X, \mathcal{A}(X) = b \quad (\text{RPCA})$$

Augmented Lagrangian Method (ALM), aka Method of Multipliers (MM). Very simple when no inequality constraints.

$$\mathcal{L}_\mu(L, S, y) = \|L\|_* + \lambda \|S\|_1 + \langle y, \mathcal{A}(L + S) - b \rangle + \frac{\mu}{2} \|\mathcal{A}(L + S) - b\|_2^2$$

Alternating Direction Method

aka Alternating Direction Method of Multipliers (ADMM)

aka inexact ALM.

In ALM, primal update is too difficult due to coupling:

$$(L_k, S_k) = \operatorname{argmin} \mathcal{L}_\mu(L, S, y_{k-1})$$

So relax... (one-step of Jacobi method)

$$L_k = \operatorname{argmin} \mathcal{L}_\mu(L, S_{k-1}, y_{k-1})$$

$$S_k = \operatorname{argmin} \mathcal{L}_\mu(L_{k-1}, S, y_{k-1})$$

ADMM with constraints, fancier terms? Perhaps; ask Don Goldfarb.

Challenges

- 1 Keep iterates low-rank when possible
- 2 Exploit sparsity
- 3 Allow constraints
- 4 Non-smooth, so *slower* convergence
- 5 How to project onto $\|Ax - b\|_2 \leq \varepsilon$?
- 6 Flexible

TFOCS main idea

$$\min_x f(x) + \psi(\bar{\mathcal{A}}x + \bar{b})$$

- 1 Find conic formulation*
- 2 Add strongly convex term
 - $f_\mu(x) = f(x) + \frac{\mu}{2} \|x - x_0\|^2$
 - can now calculate dual
 - dual is smooth
- 3 Solve dual problem
 - composite approach
 - $g = g_{\text{smooth}} + h$
 - h nonsmooth but “nice”

Extends (e.g., atomic norms)

TFOCS main idea

$$\min_x f(x) + \psi(\bar{A}x + \bar{b})$$

- 1 Find conic formulation*
- 2 Add strongly convex term
 - $f_\mu(x) = f(x) + \frac{\mu}{2} \|x - x_0\|^2$
 - can now calculate dual
 - dual is smooth
- 3 Solve dual problem
 - composite approach
 - $g = g_{\text{smooth}} + h$
 - h nonsmooth but “nice”

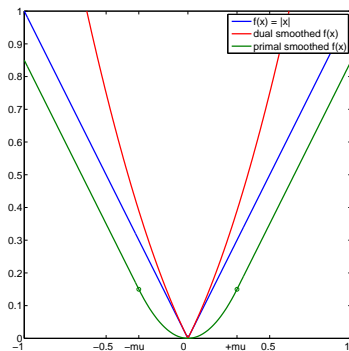
Extends (e.g., atomic norms)

Potential drawbacks:

- Primal iterate is not feasible
 - $\|Ax - b\| \leq \varepsilon$, but ε is estimate!
- Effect of smoothing
 - use continuation
 - made rigorous in proximal point framework
 - accelerated continuation
 - sometimes no effect even for $\mu > 0$

Benefits of duality

- 1 Projection onto dual cone has no linear \mathcal{A} term
- 2 Better smoothing: primal retains its **kink**



$$f^*(\lambda) \equiv \sup_x \langle \lambda, x \rangle - f(x)$$

f strongly convex $\implies f^*$
differentiable and ∇f^* Lipschitz

Smooth problems: *much* faster
convergence, i.e. $\mathcal{O}(\frac{1}{k^2})$ vs $\mathcal{O}(\frac{1}{\sqrt{k}})$

Example: matrix completion

$$\begin{array}{ll} \text{minimize} & \|X\|_{\text{tr}} \\ \text{subject to} & \|\mathcal{A}(X) - b\| \leq \varepsilon \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \text{minimize} & \|X\|_1 + \frac{\mu}{2} \|X - X_0\|_F^2 \\ \text{subject to} & (\mathcal{A}(X) - b, \varepsilon) \in \mathcal{K} \end{array}$$

Dual problem

$$\text{maximize}_{\lambda} \quad \underbrace{\inf_X \left\{ \|X\|_{\text{tr}} + \frac{\mu}{2} \|X - X_0\|^2 - \langle \lambda, \mathcal{A}(X) - b \rangle \right\}}_{-g_{\text{smooth}}(\lambda)} - \underbrace{\varepsilon \|\lambda\|_*}_{h(\lambda)}$$

“Simple gradient” (X_λ unique minimizer above)

Example: matrix completion, version 2

$$\begin{array}{ll} \text{minimize} & \|X\|_{\text{tr}} \\ \text{subject to} & \|\mathcal{A}(X) - b\| \leq \varepsilon \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \text{minimize} & t + \frac{\mu}{2} \|X - X_0\|_F^2 \\ \text{subject to} & (\mathcal{A}(X) - \bar{b}, \varepsilon) \in \mathcal{K} \\ & (X, t) \in \mathcal{K}_{\text{tr}} \end{array}$$

Dual problem

$$\begin{array}{ll} \text{maximize} & -\varepsilon \underbrace{\|\lambda\|_*}_{h(\lambda)} + \dots \\ \lambda, (\nu, s) \in \mathcal{K}_{\text{spectral}} & \end{array}$$
$$\underbrace{\inf_{X, t} \left\{ t + \frac{\mu}{2} \|X - X_0\|_F^2 - \langle \lambda, \mathcal{A}(X) - b \rangle - \langle \nu, X \rangle - st \right\}}_{-g_{\text{smooth}}(\lambda)}$$

Similar algorithm, but now $X_{\lambda, \nu}$ is linear in λ and ν , so dual is constrained quadratic (and with $2 \times$ variables).

General form

Exploit structure, not just “black-box”

Two viewpoints: **conic dual** or **Fenchel dual**

Fenchel duality view

$$\min f(x) + \sum_i \psi_i(A_i x - b_i)$$

where f, ψ_i^* are “prox-capable”, $\psi_i \rightarrow \bar{\mathbb{R}}$

$$\text{prox}_f(y) = \underset{x}{\text{argmin}} f(x) + \frac{1}{2} \|x - y\|^2$$

Matrix completion: $\psi_1(X) = \iota_{\{X: \|X\| \leq \varepsilon\}}$, $A_1 = \mathcal{A}$, $b_1 = b$.

- 1 matrix completion style 1 corresponds to:

$$f(x) = \|X\|_1, \quad \psi_2 = 0$$

- 2 matrix completion style 2 corresponds to:

$$f = 0, \quad \psi_2(x) = \|X\|_{\text{tr}}, A_2 = I, b_2 = 0$$

If $f = 0$, dual is always (constrained) quadratic.

Solving the dual

“Proximal gradient descent”, aka “forward-backward” algorithm. Handles smooth + nonsmooth (Fukushima and Mine, 1981).

- Gradient projection step for minimizing smooth g :

$$\lambda_{k+1} \leftarrow \operatorname{argmin}_{\lambda \in \mathcal{K}^*} g(\lambda_k) + \langle \nabla g(\lambda_k), \lambda - \lambda_k \rangle + \frac{L}{2} \|\lambda - \lambda_k\|^2$$

- Generalized gradient projection for minimizing $g + h$ (h nonsmooth)

$$\lambda_{k+1} \leftarrow \operatorname{argmin}_{\lambda} g(\lambda_k) + \langle \nabla g(\lambda_k), \lambda - \lambda_k \rangle + \frac{L}{2} \|\lambda - \lambda_k\|^2 + h(\lambda)$$

- Solution is **proximity operator** of h . Often known.
 - Ex. $h = \chi_C$, then proximity operator is just projection onto C
 - Ex. $h = \|x\|_1$, then proximity operator is shrinkage
- Works with backtracking and Nesterov acceleration
- Popularized in 2005: Nesterov, Beck, Teboulle

Generic algorithm (Nesterov's style)

Auslender-Teboulle version, no backtracking

$$\min_x f(x) + \psi(\bar{A}x + \bar{b}), \quad h \stackrel{\text{def}}{=} \psi^*$$

Algorithm 1 Generic algorithm for the conic standard form

Require: $\lambda_0, x_0 \in \mathbb{R}^n$, $\mu > 0$, step sizes $\{t_k\}$

1: $\theta_0 \leftarrow 1$, $v_0 \leftarrow \lambda_0$

2: **for** $k = 0, 1, 2, \dots$ **do**

3: $\nu_k \leftarrow (1 - \theta_k)v_k + \theta_k \lambda_k$

4: $x_k \leftarrow \operatorname{argmin}_x f(x) + \mu/2 \|x - x_0\|^2 - \langle \bar{A}^T(\nu_k), x \rangle$

5: $\lambda_{k+1} \leftarrow \operatorname{argmin}_\lambda h(\lambda) + \frac{\theta_k}{2t_k} \|\lambda - \lambda_k\|^2 + \langle \bar{A}(x_k) + \bar{b}, \lambda \rangle$

6: $v_{k+1} \leftarrow (1 - \theta_k)v_k + \theta_k \lambda_{k+1}$

7: $\theta_{k+1} \leftarrow 2/(1 + (1 + 4/\theta_k^2)^{1/2})$

8: **end for**

x is primal

λ, ν, v are dual, θ is scalar

Algorithm for Matrix Completion

Matrix completion, style 1

Algorithm 2 Algorithm for nuclear-norm minimization (ℓ_2 constraint)

4: $X_k \leftarrow \text{SoftThresholdSingVal}(X_0 - \mu^{-1} \mathcal{A}^T(\nu_k), \mu^{-1})$

5: $\lambda_{k+1} \leftarrow \text{Shrink}(\lambda_k - \theta_k^{-1} t_k (b - \mathcal{A}(X_k)), \theta_k^{-1} t_k \epsilon)$

$$\text{SoftThreshold}(x, \tau) = \text{sgn}(x) \cdot \max\{|x| - \tau, 0\}$$

$$\text{SoftThresholdSingVal}(X, t) = U \cdot \text{SoftThreshold}(\Sigma, t) \cdot V^T,$$

$$\text{Shrink}(z, t) \triangleq \max\{1 - t/\|z\|_2, 0\} \cdot z = \begin{cases} 0, & \|z\|_2 \leq t, \\ (1 - t/\|z\|_2) \cdot z, & \|z\|_2 > t. \end{cases}$$

Significantly extends SVT

Other new algorithms

Algorithm 3 Algorithm excerpt for Dantzig

- 4: $x_k \leftarrow \text{SoftThreshold}(x_0 - \mu^{-1} A^T A \nu_k, \mu^{-1})$.
 - 5: $\lambda_{k+1} \leftarrow \text{SoftThreshold}(\lambda_k - \theta_k^{-1} t_k A^T (b - Ax_k), \theta_k^{-1} t_k \delta)$
-

Algorithm 4 Algorithm excerpt for LASSO

- 4: $x_k \leftarrow \text{SoftThreshold}(x_0 - \mu^{-1} A^T \nu_k, \mu^{-1})$
 - 5: $\lambda_{k+1} \leftarrow \text{Shrink}(\lambda_k - \theta_k^{-1} t_k (b - Ax_k), \theta_k^{-1} t_k \epsilon)$
-

Algorithm 5 Algorithm excerpt for TV minimization

- 4: $x_k \leftarrow x_0 + \mu^{-1} (\Re(D^* \nu_k^{(1)}) - A^* \nu_k^{(2)})$
 - 5: $\lambda_{k+1}^{(1)} \leftarrow \text{CTrunc}(\lambda_k^{(1)} - \theta_k^{-1} t_k^{(1)} D x_k, \theta_k^{-1} t_k^{(1)})$
 $\lambda_{k+1}^{(2)} \leftarrow \text{Shrink}(\lambda_k^{(2)} - \theta_k^{-1} t_k^{(2)} (b - Ax_k), \theta_k^{-1} t_k^{(2)} \epsilon)$
-

Conic Programs

$$\min_x \langle c, x \rangle \quad \text{such that} \quad x \succeq_{\mathcal{K}} 0, \quad Ax = b$$

$$\begin{aligned} \mathcal{K} = \mathbb{R}_+^n & \implies \text{LP} \\ \mathcal{K} = \{(x, t) \in \mathbb{R}^{n+1} : \|x\|_2 \leq t\} & \implies \text{SOCP} \\ \mathcal{K} = S_+^n & \implies \text{SDP} \end{aligned}$$

Dual, before smoothing

$$\max_{\nu, \lambda} -\langle b, \nu \rangle \quad \text{such that} \quad \lambda \succeq_{\mathcal{K}^*} 0, \quad \lambda = c + A^* \nu$$

Dual, after smoothing

$$\max_{\nu, \lambda} -\langle b, \nu \rangle - \frac{1}{2\mu} \|c - \lambda + A^* \nu\|^2 + \langle c - \lambda + A^* \nu, x_0 \rangle \quad \text{such that} \quad \lambda \succeq_{\mathcal{K}^*} 0.$$

Related work

Inspiration: apply SVT 2008 (aka linearized Bregman 2008, aka Uzawa) to Dantzig.

A good idea should be discovered many times! Much recent work in similar flavors:

- O. Mangasarian 1981. Add quadratic term to objective of linear program and solve dual.
- PPPA, F. Malgouyres and T. Zeng 2008. Use continuation, prove convergence of Ax , not x . Conclude line search is not worth it.
- Y.-J. Liu, D. Sun, K.C. Toh 2009. Similar approach: apply proximal point algorithm (not accelerated), solve inner problems with APG (like TFOCS). For matrix completion problem.
- Chambolle and Pock 2010, unconstrained versions, for general functions
- Combettes and Pesquet 2009, discuss forward-backward applied to dual of smoothed problem, for general functions.
- Combettes, Dũng and Vũ 2010, extend previous work and prove convergence (no rate).

TFOCS ideas: extras

Software is **modular**. Easy to add constraints, change solver. . .

(Important) details

- 6 first-order methods (GRA + 5 accelerated methods)
- Efficient step size procedures (based on Tseng's convergence analysis): no Lipschitz constant needed; like Gonzaga/Karas/Rossetto. Key idea: if L updated, θ must be updated as well
- Easy testing and benchmarking
- Efficient use of linear operator structure: crucial when backtracking occurs

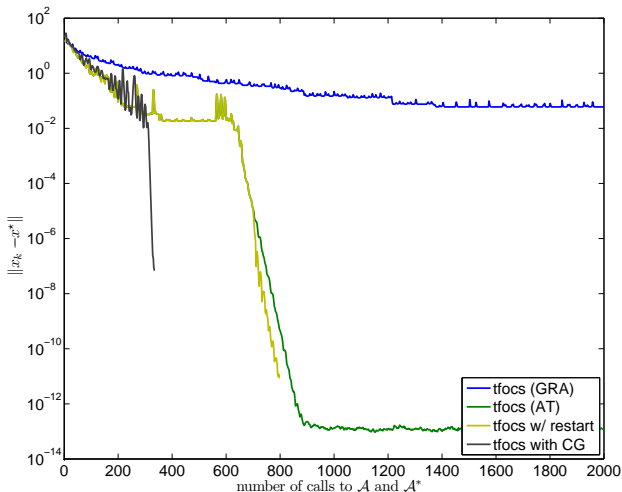
$$\text{minimize } g_{\text{smooth}}(\mathcal{A}^T \lambda) + h(\lambda)$$

- Accelerated continuation: remove effect of μ
- Exact perturbation
- Restart strategies
- Convergence proofs

Conjugate Gradient

Advantage of modularity: easy to try new solvers, line search.

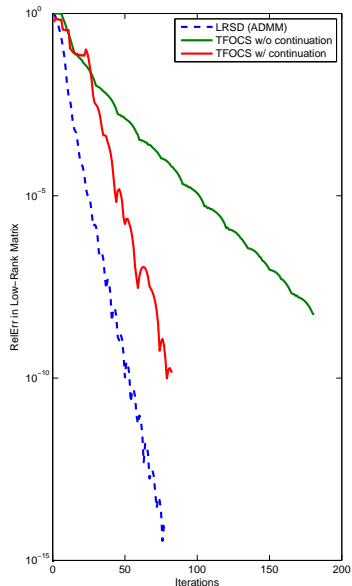
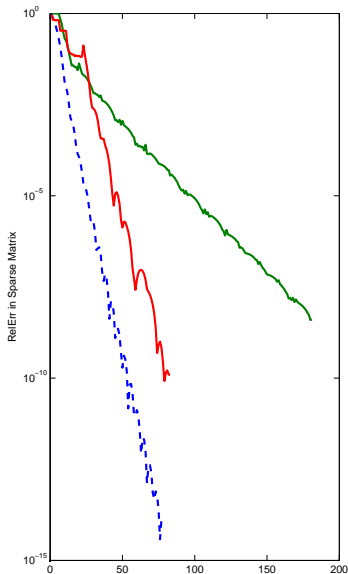
Plans for non-linear CG, (L-)BFGS, SESOP, Karimi/Vavasis ...



Ex: Non-linear CG (Polak-Ribiere), noiseless basis pursuit, $N = 2048$.

Comparison on RPCA

Compare with LRSD (ADMM code by Yuan, Yang)



Standard continuation

Want perturbation small

$$\begin{array}{ll} \text{minimize} & f(x) + \frac{1}{2}\mu\|x - x_0\|^2 \\ \text{subject to} & \mathcal{A}(x) + b \in \mathcal{K} \end{array}$$

Problem: $L \propto 1/\mu$

Algorithm 6 Standard continuation

Require: $Y_0, \mu_0 > 0, \beta < 1$

1: **for** $j = 0, 1, 2, \dots$ **do**

2: $X_{j+1} \leftarrow \operatorname{argmin}_{\mathcal{A}(x)+b \in \mathcal{K}} f(x) + \frac{\mu_j}{2}\|x - Y_j\|_2^2$

3: $Y_{j+1} \leftarrow X_{j+1}$ (or $Y_{j+1} \leftarrow Y_0$)

4: $\mu_{j+1} \leftarrow \beta\mu_j$

5: **end for**

FPC: Hale, Yin, and Zhang ('08)

Moreau-Yosida regularization

Moreau envelope $h(\mathbf{Y}) = \min_{x \in C} f(x) + \frac{\mu}{2} \|x - \mathbf{Y}\|_2^2$

Moreau proximity operator $X_{\mathbf{Y}} = \operatorname{argmin}_{x \in C} f(x) + \frac{\mu}{2} \|x - \mathbf{Y}\|_2^2$

Theorem

h is continuously differentiable with gradient

$$\nabla h(\mathbf{Y}) = \mu(\mathbf{Y} - X_{\mathbf{Y}})$$

The gradient is Lipschitz continuous with constant $L = \mu$

Minimizing h by gradient descent \rightarrow proximal point algorithm (PPA) (Rockafellar, 70s)

Accelerated continuation (Nesterov style)

If proximal-point algorithm is gradient descent, then why not accelerate?

Algorithm 7 Accelerated continuation

Require: $Y_0, \mu_0 > 0$

1: $X_0 \leftarrow Y_0$

2: **for** $j = 0, 1, 2, \dots$ **do**

3: $X_{j+1} \leftarrow \operatorname{argmin}_{A(x)+b \in \mathcal{K}} f(x) + \frac{\mu_j}{2} \|x - Y_j\|_2^2$

4: $Y_{j+1} \leftarrow X_{j+1} + \frac{j}{j+3} (X_{j+1} - X_j)$

5: (optional) increase or decrease μ_j

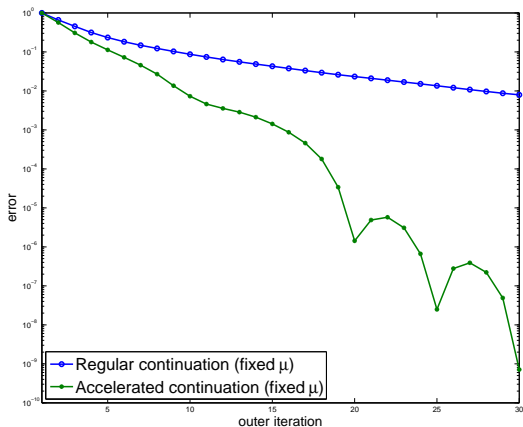
6: **end for**

Keep $\mu_j \equiv \mu$ so subproblems quick to solve

Warm-start subproblems

For small μ , typically 5 iterations

Simple vs. accelerated continuation: LASSO example



$\|x_k - x^*\| / \|x_0 - x^*\|$ vs. outer iteration count

Effect of perturbation

Nice surprise:

Linear programs (ex. Dantzig, Basis Pursuit) have exact penalty

Theorem (Exact penalty)

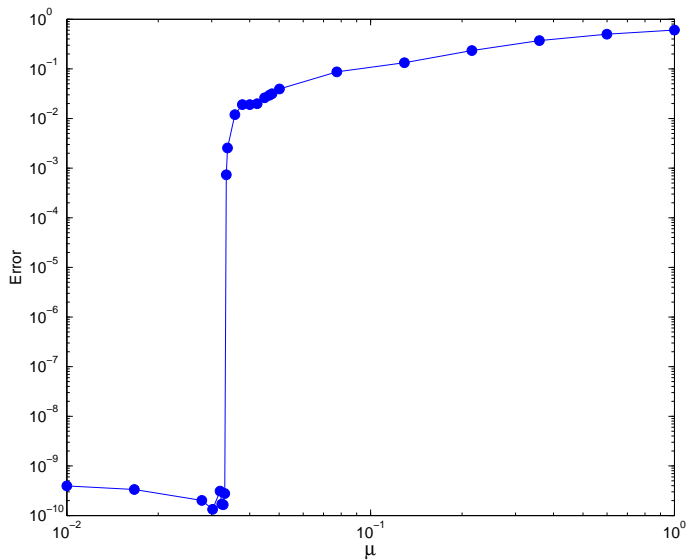
- Arbitrary LP with objective $\langle c, x \rangle$ and with opt. solution
- Perturbed LP with objective $\langle c, x \rangle + \frac{1}{2}\mu\|x - x_0\|_Q^2$, $Q \succeq 0$

There is $\mu_0 > 0$ s.t. for $0 < \mu \leq \mu_0$, any solution to perturbed problem is a solution to LP

- Special case (BP): Yin ('10)
- More general result: Friedlander and Tseng ('07)
- Combine with continuation \implies finite termination
Known since Bertsekas '75, Polyak and Tretjakov '74, Mangasarian '79

Illustration

Exact penalty for Dantzig Selector (since linear program)



Parameters

Lipschitz Gradient

$$f(y) \leq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{L}{2} \|x - y\|_2^2$$

Strong Convexity

$$f(y) \geq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{m_f}{2} \|x - y\|_2^2$$

If $\nabla^2 f$ exists, equivalent to

$$m_f I \preceq \nabla^2 f \preceq LI$$

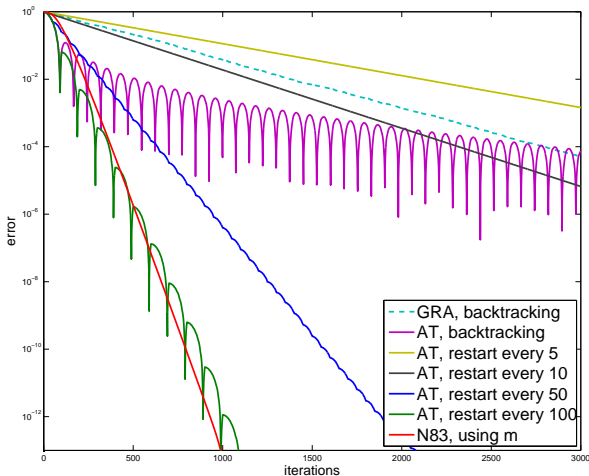
Goal: user needs no knowledge of m_f and L

- For L , trick: backtracking line search
- For m_f , trick: restart

Restart

Problem: accelerated schemes don't **automatically** take advantage of strong convexity.

i.e. m_f unknown \implies no linear convergence



Restart

Convergence of accelerated method:

$$f(x_k) - f^* \leq \frac{L}{k^2} \|x^* - x_0\|^2$$

If f is strongly convex with parameter m_f ,

$$\|x_k - x^*\|^2 \leq \frac{2L}{m_f} \frac{1}{k^2} \|x^* - x_0\|^2$$

With restart, x_0 is x_k of a previous sequence. Do this j times.

$$\|x_{jk} - x^*\| \leq \left(\sqrt{\frac{2L}{m_f} \frac{1}{k}} \right)^j \|x^* - \hat{x}_0\|$$

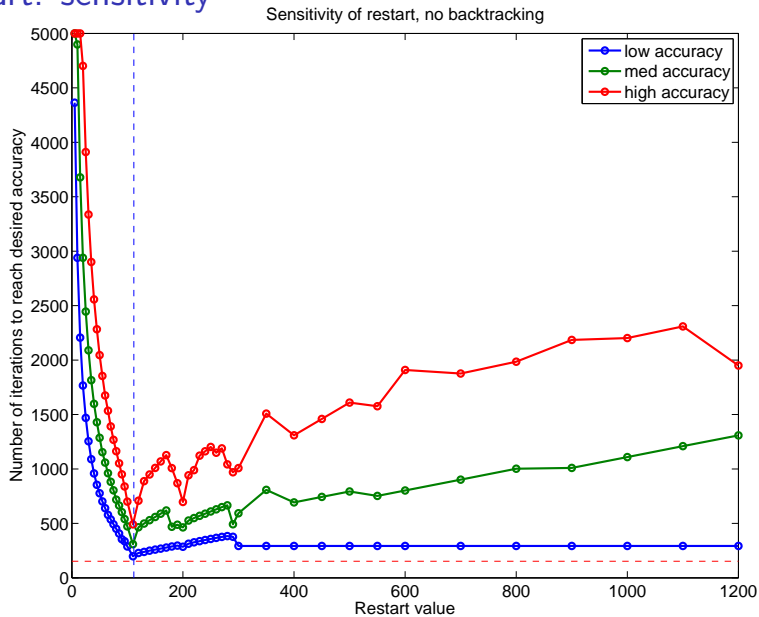
This is linear convergence with rate $\rho = \left(\sqrt{\frac{2L}{m_f} \frac{1}{k}} \right)^{1/k}$.

$$k_{\text{opt}} = e \sqrt{\frac{2L}{m_f}}$$

See PARNES paper (Gu, Lim, Wu 2009), Nesterov 2007, and also Nemirovskii-Yudin (80s).

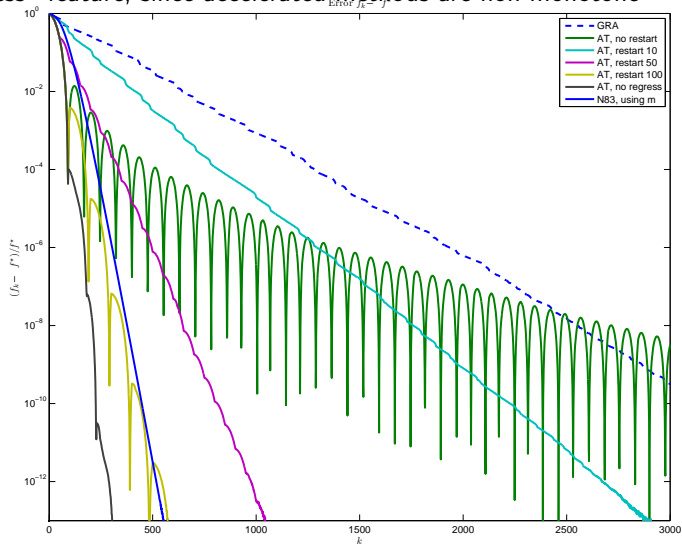
Goes back to Powell (1977) for non-linear CG.

Restart: sensitivity



Restart: improvements

“No Regress” feature, since accelerated methods are non-monotone



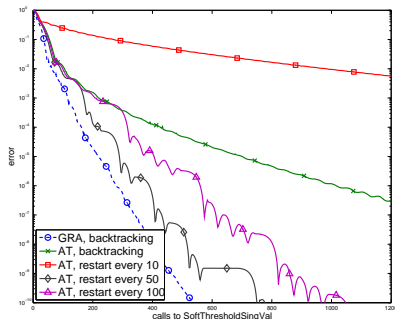
Special considerations for matrix completion

(1) Keep μ small always, to keep X_k low-rank:

$$X_k \leftarrow \text{SoftThresholdSingVal}(X_0 - \mu^{-1} \mathcal{A}^T(\nu_k), \mu^{-1})$$

(2) Gradient descent performs well! Why?

- **automatically** exploits strong convexity
- More robust to errors in SVD calculation (cf. Devolder/Nesterov)



Note: advantage of gradient descent only appears with equality constraints

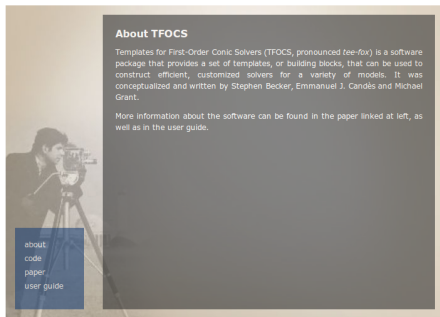
Convergence rates

- Inner iterations: objective converges in $\mathcal{O}(1/k^2)$
- Outer iterations: if via proximal point method, locally linear, or globally $\mathcal{O}(1/j)$. If via accelerated proximal point method, $\mathcal{O}(1/j^2)$.
- How to combine the two? One method: Liu/Sun/Toh 2009
- Or, result of Güler 1990s, on inexact accelerated proximal point method. Need primal variables of inner iterates to converge.
- Key result: Fadili/Peyré March 2011.
- Preliminary work: $\mathcal{O}(\varepsilon^{-5/4})$ iterations to reach ε -solution

Software release

- Paper
- User guide
- Software (MATLAB)
 - solvers
 - many simple examples
 - a few real-world examples
 - continuation wrappers
 - compatible with SPOT
- Parameters: any $\mu > 0$

TFOCS Templates for First-Order Conic Solvers



The screenshot shows a webpage for TFOCS. On the left, there is a navigation menu with the following items: 'about', 'code', 'paper', and 'user guide'. The main content area is titled 'About TFOCS' and contains the following text: 'Templates for First-Order Conic Solvers (TFOCS, pronounced tee-fox) is a software package that provides a set of templates, or building blocks, that can be used to construct efficient, customized solvers for a variety of models. It was conceptualized and written by Stephen Becker, Emmanuel J. Candès and Michael Grant. More information about the software can be found in the paper linked at left, as well as in the user guide.' The background of the webpage features a photograph of a person using a camera on a tripod.

©2010, Caltech.

<http://tfocs.stanford.edu>

Example in TFOCS

Basis Pursuit Denoising BP_ϵ , analysis

$$\min_x \|Wx\|_1 \quad \text{such that} \quad \|Ax - b\|_2 \leq \epsilon$$

```
prox   = { prox_l2( epsilon ), proj_linf };  
linear = { A, -b; W, 0 };  
x      = tfocs_SCD( [], linear, prox, mu, x0 );
```

Easy to add constraints, e.g. $x \geq 0$

```
prox   = { prox_l2( epsilon ), proj_linf, proj_Rplus };  
linear = { A, -b; W, 0; 1, 0 };
```

Of course, this is also builtin...

```
x = solver_sBPDN_W(A,W,b,epsilon,mu)
```

No Lipschitz constant or step size needed!