TFOCS: A General First-Order Framework for Constrained Optimization

Stephen Becker

Laboratoire Jacques-Louis Lions, Paris VI;
Fondation Sciences Mathématiques de Paris

November 18, 2011

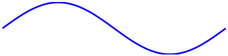
TFOCS = Templates for First-Order Conic Solvers

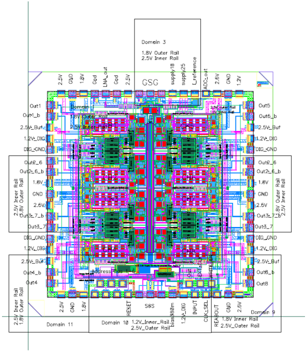
Collaborators:
Michael Grant (Caltech, CVX Research)
Emmanuel Candès (Stanford)

Motivation: new analog-to-digital converter

Analog-to-*information* (A2I) converter using compressed sensing

Input signal $x(t)$
2.5 GHz





Sample at 400 MHz
($12.5\times$ below Nyquist)
Output:

$$b_k, \quad k \in \mathbb{Z}$$

Measurements are *linear* and x is *finite dimensional*:

$$b = Ax$$

Convex optimization to recover x, exploiting prior knowledge of the signal class

Applicable to many other fields: machine learning, image processing, economics, operations research, ...

Typical problems

$$b = Ax + z, \quad A \in \mathbb{R}^{m \times n}, \quad z \text{ is noise}$$

x is sparse, $m \ll n$. Define $\|x\|_1 = \sum_i |x_i|$.

Basis Pursuit BP

$$\min_x \|x\|_1 \quad \text{such that} \quad Ax = b$$

or if x is "W-sparse": i.e. $\exists\, \alpha \in \mathbb{R}^d$ sparse, such that $W^T \alpha \simeq x$.

Basis Pursuit Denoising BP_ε, analysis (includes TV denoising)

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

Alternatives:

Dantzig Selector

$$\min_x \|x\|_1 \quad \text{such that} \quad \|A^T(Ax - b)\|_\infty \leq \delta$$

Typical problems: matrix completion

Matrix completion

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

$\|X\|_{\mathrm{tr}}$ is the nuclear norm (sum of singular values).
$\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ 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:

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

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

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 Low-rank and Sparse

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

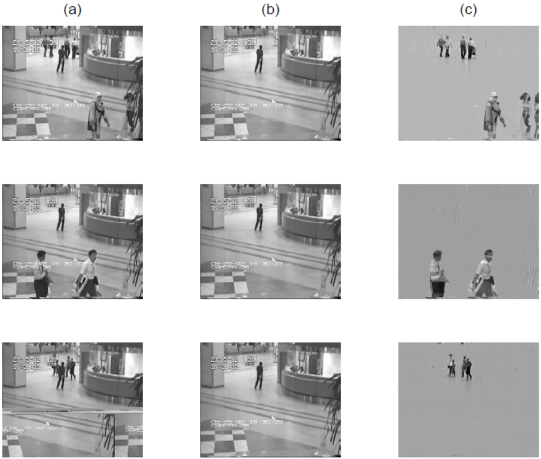


image from Goldfarb, Ma, Sheinberg '10

Typical problems: variational image processing

Goal: denoise, deblur, inpaint, or improve resolution of an image... or do combinations!

Variational image denoising: add regularizers $r(x)$.

Example:

- $r(x) = \|x\|_{TV} = \sum_{i,j} \sqrt{(\Delta_X x)^2 + (\Delta_Y x)^2}$ "total-variation" aka "TV"
- $r(x) = \|W^* x\|_1$ where W^* is a wavelet or curvelet analysis operator
- $\|x\|_{1,2}^{\mathcal{B}} = \sum_{p \in \mathcal{B}} \|x_{(p)}\|_2$, block-variant of ℓ_1, where $\mathcal{B}$ is some partition of $\{1, \ldots, n\}$

Image deblurring and denoising and inpainting

$$\min_u \|u\|_{1,2}^{\mathcal{B}} + \|Wu\|_{TV} + \frac{1}{2}\|b - BS(Wu)\|_2^2$$

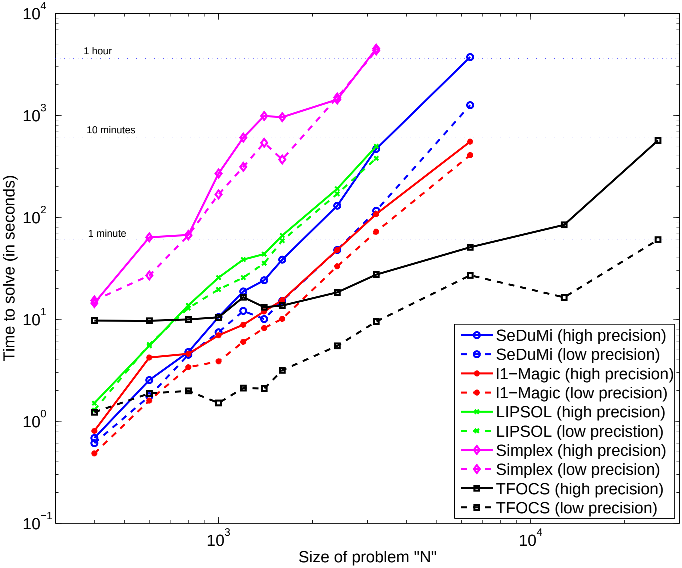
where signal is recovered via $x = Wu$, and B is a blur and S is a sub-sample.

(c) LaBoute y_0 (d) $y = MKy_0 + w$, 3.93 dB (e) $\hat{y}_0 = W\hat{x}$, 23.83 dB

image from Raguet, Fadili, Peyré '11 (using W a wavelet tight frame)

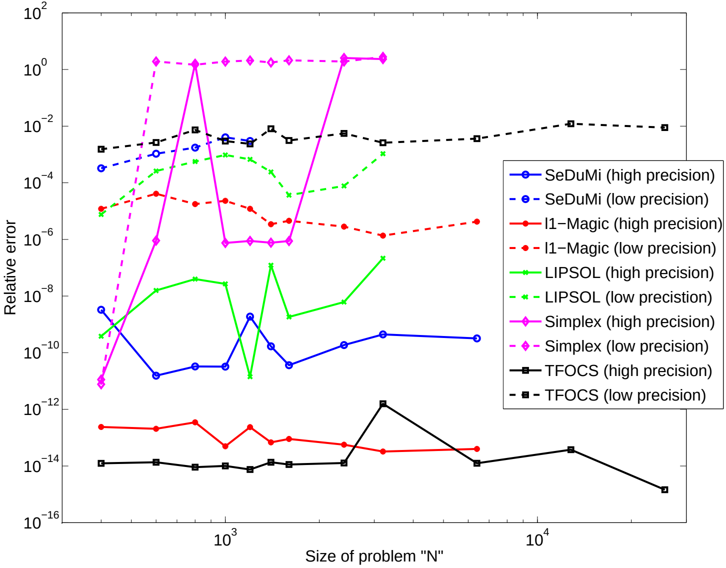
Interior Point methods

Experiments that ran on a cluster (2008) are now run on a laptop.



Interior Point methods (2)

But *accuracy* of first-order methods. . . ? Not a problem.



First-order methods

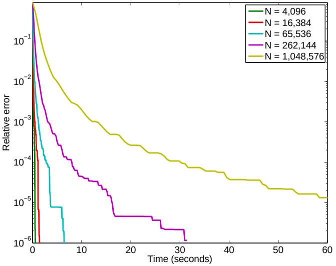
Conclusion: due to size of problem, first-order methods beat IPM for this application since they scale better.

Also, first-order methods easily exploit fast operators (FFT,. . .)

Similar fact: homotopy-type methods only competitive in special cases

There are *fast* alternatives that solve *similar* problems: greedy (OMP, CoSaMP), hybrid (ALPS), message passing (AMP), iterative hard-thresholding.

Example: basis pursuit with DCT using TFOCS. Solve 10^6 variables to 10^{-5} relative error in 1 minute



Existing first-order solvers (2010)

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

$$\min_x \|x\|_1 + \frac{\lambda}{2}\|Ax - b\|_2^2$$

$$\min_x \|x\|_1 \text{ such that } Ax = b$$

1. Require $AA^T = I$ or solve inner subproblem
   - NESTA (B., Bobin, Candès)
   - C-SALSA (Afonso, Figueiredo, Bioucas-Dias)
   - Z. Lu (for the Dantzig)
   - (Lu, Pong, Zhang) ADM for Dantzig
2. Restrictions on W
   - SPGL1 (Friedlander, van den Berg)
3. Solve un-constrained version, or equality constraints only
   - FPC, FPC-AS, GPSR, SpaRSA, FISTA, Bregman, ...
   - (cannot handle W)

Brand-new first-order solvers (2011)

First-order methods that can solve these complicated variants:

- TFOCS: Becker-Candès-Grant (2010)
- Chambolle-Pock (2010)
  - . . . extended by He and Yuan (2010)
  - . . . extended by Condat (2011) and Vũ (2011)
- Briceño-Arias–Combettes (2011) "monotone + skew"
  - use modified Forward-Backward ("Forward-Backward-Forward") of Tseng 1998
  - or Monteiro-Svaiter 2010
- Chen-Teboulle (1994)
- Combettes-Pesquet (2011)

(Almost) all of these since September 2010!

Review

What is a first-order method?
Uses first-derivate information ∇f, as opposed to Hessian $\nabla^2 f$

$$\min_x f(x) \text{ such that } x \in C$$

Projected gradient descent, aka forward-backward algorithm

$$x_{k+1} = \mathcal{P}_C(x_k - t\nabla f(x_k))$$

Works if:

- f is smooth so that ∇f exists (also need ∇f Lipschitz)
- $\mathcal{P}_C$, the projection onto C, is easy to compute

Challenges and desiderata

1. Allow difficult constraints and arbitrary linear operators
   - How to project onto $\|Ax - b\|_2 \leq \varepsilon$?
   - Allow several constraints, like $\|Ax - b\|_2 \leq \varepsilon,\ x \geq 0, x \leq 1$
2. Allow non-smooth objectives, so *slower* convergence: how to fix?
3. Allow complicated objectives, like $\|Wx\|_1 + \|x\|_{TV}$
4. Flexible: allow prototyping, like CVX
5. Few parameters
6. Exploit sparsity or FFT-based operators
7. (Matrix problems) Keep iterates low-rank when possible

TFOCS main idea

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

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

TFOCS main idea

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

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

Potential drawbacks:

Q: Primal iterate is not feasible

A: $\|Ax - b\| \leq \varepsilon$, but ε is estimate!

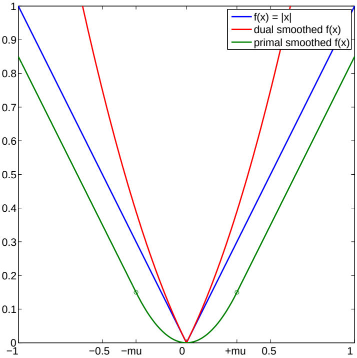
Q: Effect of smoothing

A: Use continuation

- rigorous via 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



Fenchel dual

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

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

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

Dual problem

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

$g_{\mathsf{smooth}}(\lambda)$ has gradient $\mathcal{A}(X_\lambda) - b$, where X_λ is unique minimizer above

Example: matrix completion, version 2

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

Dual problem

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

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", $\psi_i : \mathbb{R}^{m_i} \to \bar{\mathbb{R}}$

$$\mathrm{prox}_f(y) = \operatorname*{argmin}_x f(x) + \frac{1}{2}\|x - y\|^2$$

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

1. matrix completion style 1 corresponds to:
   $f(x) = \|X\|_{\mathsf{tr}}, \quad \psi_2 = 0$
2. matrix completion style 2 corresponds to:
   $f = 0, \quad \psi_2(x) = \|X\|_{\mathsf{tr}}, A_2 = I, b_2 = 0$

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:

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

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

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

- 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 (Nesterov, Beck/Teboulle 2005)

Generic algorithm (Nesterov's style)

Auslender-Teboulle version, no backtracking

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

Algorithm 1 Generic algorithm for the conic standard form

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

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

for $k = 0, 1, 2, \ldots$ **do**

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

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

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

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

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

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: $X_k \leftarrow \mathrm{SoftThresholdSingVal}(X_0 - \mu^{-1}\mathcal{A}^T(\nu_k), \mu^{-1})$
5: $\lambda_{k+1} \leftarrow \mathrm{Shrink}(\lambda_k - \theta_k^{-1} t_k (b - \mathcal{A}(X_k)), \theta_k^{-1} t_k \epsilon)$

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

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

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

Significantly extends SVT

Other new algorithms

Algorithm 3 Algorithm excerpt for Dantzig

$x_k \leftarrow \mathrm{SoftThreshold}(x_0 - \mu^{-1} A^T A \nu_k, \mu^{-1})$.

$\lambda_{k+1} \leftarrow \mathrm{SoftThreshold}(\lambda_k - \theta_k^{-1} t_k A^T (b - A x_k), \theta_k^{-1} t_k \delta)$

Algorithm 4 Algorithm excerpt for LASSO

$x_k \leftarrow \mathrm{SoftThreshold}(x_0 - \mu^{-1} A^T \nu_k, \mu^{-1})$

$\lambda_{k+1} \leftarrow \mathrm{Shrink}(\lambda_k - \theta_k^{-1} t_k (b - A x_k), \theta_k^{-1} t_k \epsilon)$

Algorithm 5 Algorithm excerpt for TV minimization

$x_k \leftarrow x_0 + \mu^{-1}(\Re(D^* \nu_k^{(1)}) - A^* \nu_k^{(2)})$

$$\begin{aligned} \lambda_{k+1}^{(1)} &\leftarrow \mathrm{CTrunc}(\lambda_k^{(1)} - \theta_k^{-1} t_k^{(1)} D x_k, \theta_k^{-1} t_k^{(1)}) \\ \lambda_{k+1}^{(2)} &\leftarrow \mathrm{Shrink}(\lambda_k^{(2)} - \theta_k^{-1} t_k^{(2)} (b - A x_k), \theta_k^{-1} t_k^{(2)} \epsilon) \end{aligned}$$

Conic Programs

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

$$\begin{array}{lcl} \mathcal{K} = \mathbb{R}^n_+ & \Longrightarrow & \text{LP} \\ \mathcal{K} = \{(x,t) \in \mathbb{R}^{n+1} : ||x||_2 \leq t\} & \Longrightarrow & \text{SOCP} \\ \mathcal{K} = S^n_+ & \Longrightarrow & \text{SDP} \end{array}$$

Dual, before smoothing

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

Dual, after smoothing

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

TFOCS ideas: extras

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

(Important) details

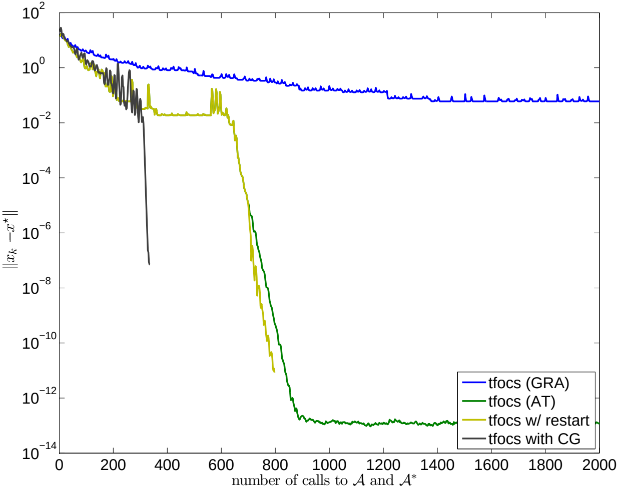
- 6 first-order methods (GRA + 5 accelerated methods)
- Efficient step size procedures (based on Tseng's convergence analysis): no Lipschitz constant needed. Key idea: if L updated, θ must be updated as well
- Easy testing and benchmarking
- Efficient use of linear operator structure: crucial when backtracking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

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

- Accelerated continuation: remove effect of μ
- Exact perturbation
- Restart strategies to ensure optimal performance

Conjugate Gradient

Advantage of modularity: easy to try new solvers, line search.
CG, (L-)BFGS, SESOP . . .



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

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, $\mu_0 > 0$, $\beta \leq 1$

for $j = 0, 1, 2, \ldots$ **do**

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

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

$\quad \mu_{j+1} \leftarrow \beta\mu_j$

end for

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

Moreau-Yosida regularization

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

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

Theorem

h is continuously differentiable with gradient

$$\nabla h(Y) = \mu(Y - X_Y)$$

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

Minimizing h by gradient descent $\rightarrow$ proximal point algorithm (PPA)
(Martinet, 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** $j = 0, 1, 2, \ldots$ **do**

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

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

5: $\quad$ (optional) increase or decrease μ_j

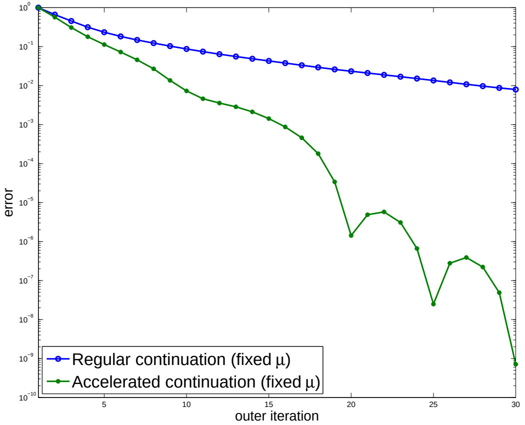
6: **end for**

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



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

Effect of perturbation

Nice surprise:

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

Theorem (Exact penalty)

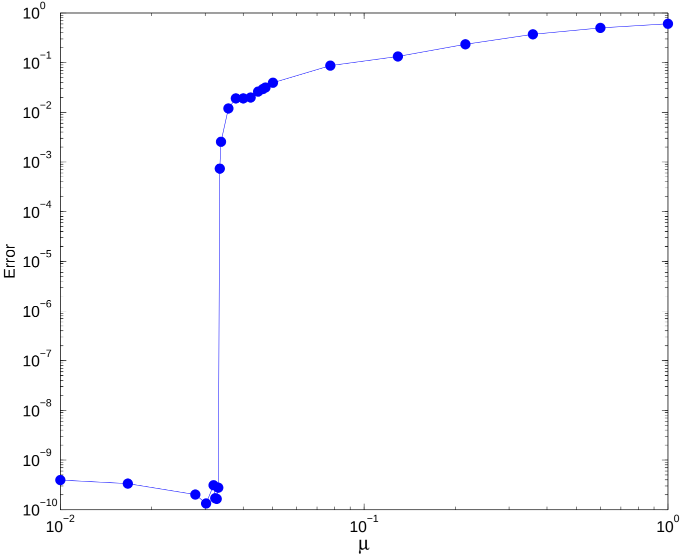
- *Arbitrary LP with objective* $\langle c, x \rangle$ *and with optimal solution*
- *Perturbed LP with objective* $\langle c, x \rangle + \frac{1}{2}\mu\|x - x_0\|_2^2$

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

Linesearch

Tentative new point y_{k+1} using stepsize $1/L_k$ must satisfy:

$$f(y_{k+1}) \le f(y_k) + \langle y_{k+1} - y_k, \nabla f(y_k) \rangle + \frac{L_k}{2} \|y_{k+1} - y_k\|_2^2$$

Problem: suffers from cancellation issues in finite precision. To see this: Let $y_{k+1} - y_k = \varepsilon h$ where $\|h\| = 1$. As $k \to \infty$, $\varepsilon \to 0$. Then

$$f(y_{k+1}) \le f(y_k) + \varepsilon \langle h, \nabla f(y_k) \rangle + \frac{\varepsilon^2 L_k}{2}$$

If $\langle h, \nabla f(y_k) \rangle \gg \varepsilon$, this term dominates the $\varepsilon^2 L_k$ term.

Solution: instead, check this (equivalent) condition

$$\langle y_{k+1} - y_k, \nabla f(y_{k+1}) - \nabla f(y_k) \rangle \le \frac{L_k}{2} \|y_{k+1} - y_k\|_2^2$$

Since ∇f is Lipschitz, $\|\nabla f(y_{k+1}) - \nabla f(y_k)\| \le L\varepsilon$, so both sides of the inequality are $\mathcal{O}(\varepsilon^2)$. Cost of ∇f is often similar to cost of f.

Linesearch subtleties

Linesearch test:

$$\langle y_{k+1} - y_k, \nabla f(y_{k+1}) - \nabla f(y_k) \rangle \le \frac{L_k}{2} \|y_{k+1} - y_k\|_2^2$$

Often, f has structure $f(x) = g(Ax)$, so $\nabla f(x) = A^* \nabla g(Ax)$.
Algorithm is aware of this and computes

$$\langle Ay_{k+1} - Ay_k, \nabla g(Ay_{k+1}) - \nabla g(Ay_k) \rangle \le \frac{L_k}{2} \|y_{k+1} - y_k\|_2^2$$

so Ay_{k+1} and Ay_k can be re-used.

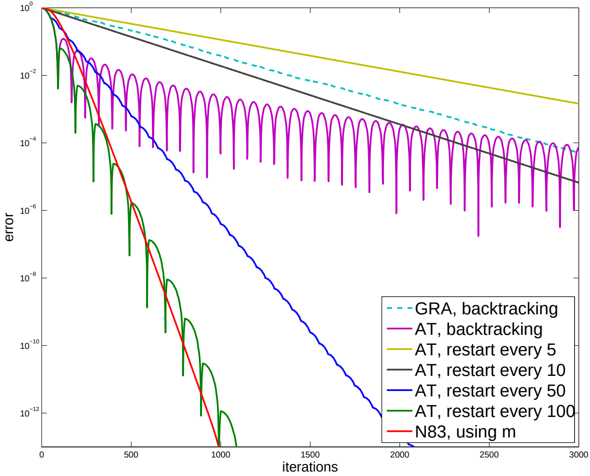
If L_k changes, convergence rate bound is improved if weight parameter θ is also updated:

$$\theta_{k+1} = \frac{2}{1 + \sqrt{1 + 4L_{k+1}/(L_k \theta_k^2)}}.$$

Restart

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

$$\text{i.e. } m_f \text{ unknown} \implies \text{no linear convergence}$$



Restart

Convergence of accelerated method:

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

If f is strongly convex with parameter m_f,

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

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

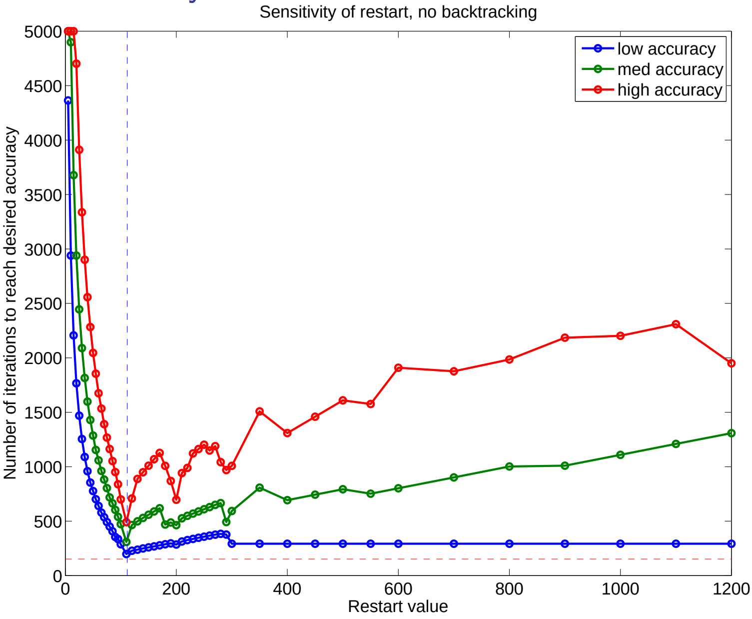
$$\|x_{jk} - x^\star\| \le \left(\sqrt{\frac{2L}{m_f}}\frac{1}{k}\right)^j \|x^\star - \hat{x}_0\|$$

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

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

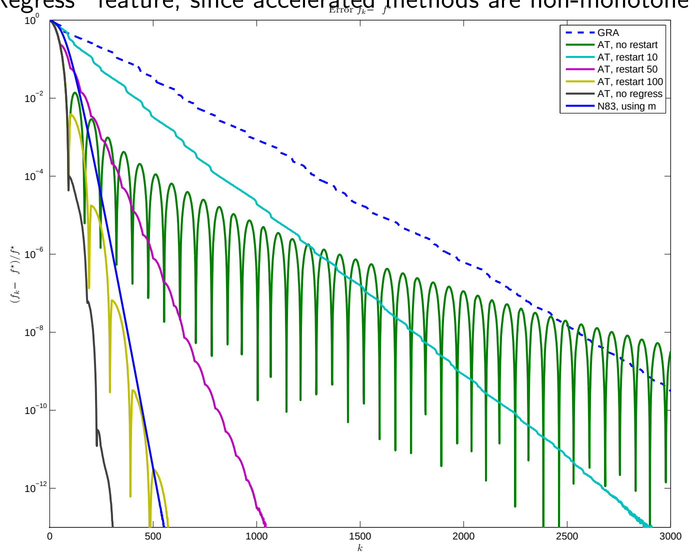
See PARNES paper (Gu, Lim, Wu 2009), Nesterov 2007, and also Nemirovskii-Yudin (80s).
Goes back to Powell (1977) for non-linear CG.

Restart: sensitivity



Restart: improvements

"No Regress" feature, since accelerated methods are non-monotone



Comparison with Chambolle/Pock

$$\min_x \; f(x) + \psi(Ax - b), \qquad \max_\lambda \; \psi^*(\lambda) + f^*(-A^*\lambda) + \langle \lambda, b \rangle, \quad L \equiv \|A\|.$$

TFOCS. Uzawa applied to perturbed problem. Pick $t \le \mu/L^2$.

$$x_{k+1} = \operatorname*{argmin}_x f(x) - \langle \bar{\lambda}, Ax - b \rangle + \frac{\mu}{2}\|x - x_0\|^2$$

$$\lambda_{k+1} = \operatorname*{argmin}_\lambda \psi^*(\lambda) - \langle Ax_{k+1}, \lambda \rangle + \frac{1}{2t}\|\lambda - \lambda_k\|^2$$

$$\bar{\lambda} = \lambda_{k+1} + \theta_k(\lambda_{k+1} - \lambda_k) \quad \text{(e.g. simple Nesterov; other choices possible)}$$

Chambolle/Pock. Arrow-Hurwicz if $\theta = 0$. Gradient descent on primal-dual simultaneously. Pick $\tau\sigma < 1/L^2$.

$$\lambda_{k+1} = \operatorname*{argmin}_\lambda \psi^*(\lambda) - \langle A\bar{x}, \lambda \rangle + \frac{1}{2\sigma}\|\lambda - \lambda_k\|^2$$

$$x_{k+1} = \operatorname*{argmin}_x f(x) - \langle \lambda_{k+1}, Ax - b \rangle + \frac{1}{2\tau}\|x - x_k\|^2$$

$$\bar{x} = x_{k+1} + \theta_k(x_{k+1} - x_k) \quad \text{(acceleration sometimes possible)}$$

Comparison with Chambolle/Pock (2)

Analogous to dual-IPM and primal-dual IPM. Merits to both algorithms.

TFOCS

- ☹ Best choice for μ? Requires outer iteration, stopping criteria
- Accelerated outer iteration (à Nesterov)
- Accelerated inner iteration (*always* accelerated)
- Finite convergence of outer iteration for LP
- Inner iteration is standard: benefits from line search, so t chosen automatically
- For some problems, can use CG, L-BFGS, etc. on inner iteration

Overall, TFOCS will solve the inner problem faster, but it has to solve several inner problems.

Chambolle/Pock

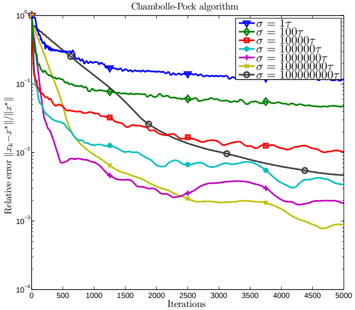
- ☹ How to choose τ and σ? Linesearch possible?
- No outer iteration
- For accelerated version, line search not possible yet: less established framework
- *Sometimes* accelerated, i.e. if f or ψ^* is strongly convex (i.e. f^* or ψ has Lipschitz gradient)

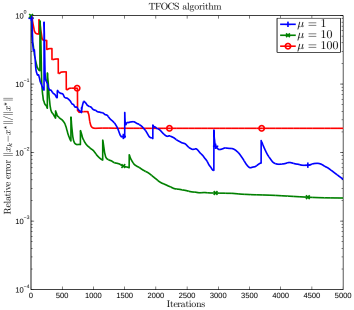
Comparison with Chambolle/Pock (3)

Dantzig Selector with weighted norm

$$\min_x \|W^* x\|_1 \quad \text{such that} \quad \|A^T(Ax - b)\|_\infty \leq \delta$$

Numerical test with W^* an over-sampled DCT transform, and signal x superposition of sine waves.
At solution, $W^* x$ is not sparse, which makes problem harder to solve





Algorithms perform similarly. Knowing correct value of σ/τ is quite helpful.

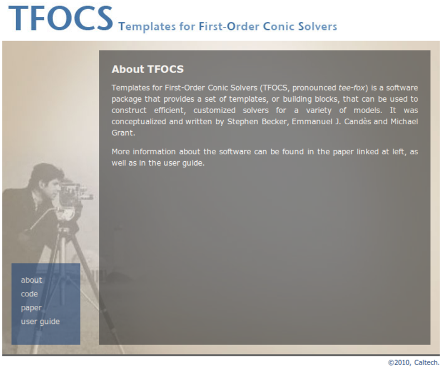
Convergence rates

$$\min_{Y} \underbrace{\min_{x} f(x) + \frac{\mu}{2}\|x - Y\|^2}_{\phi(Y)}$$

- Inner iterations: objective converges in $\mathcal{O}(1/k^2)$ ($g(\lambda_k) \to g^\star \equiv \phi(Y)$)
- 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.
    - via Fadili/Peyré 2011, $\|x_k - x^\star\|^2 \leq g^\star - g(\lambda_k)$
    - Want μ large for inner solve, μ small for outer solve.
    - If f smooth and (Y_k) bounded, then $\mathcal{O}(\varepsilon^{-5/4})$ 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 $\mu > 0$



http://tfocs.stanford.edu

Example in TFOCS

Basis Pursuit Denoising BP_ε, analysis

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

```
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!

Open problems

Optimization

- Compare the new general first-order methods: TFOCS, Chen-Teboulle, Combettes-Pesquet, Briceño-Arias–Combettes (2011), Chambolle-Pock (and He-Yuan/Condat/Vũ extensions)
  - Complexity analysis of TFOCS in general case, stopping criteria
  - Convergence rate of other algorithms
- First-order CVX. Modular, robust (no parameters), fast.
- Improve speed of constrained first-order methods
  - Performance on *non-sparse* problems is still slow
  - Scaling issues are significant
  - For unconstrained problems, non-linear CG and L-BFGS are fantastic. For constrained problems, we have nothing
  - Preconditioned truncated-Newton methods
- Adapt to current computing paradigms
  - Multicore processors
  - Giant datasets with distributed memory
  - Stochastic approaches, error tolerance
- Specific techniques for matrix variables
  - Randomized SVDs: require error tolerance
  - Keep iterates low-rank

Signal processing

- New measurement schemes: easy calibration
- Exploit unconventional prior information (beyond sparsity)
- Non-linear measurements