Sparse Optimization: Algorithms and Applications

Stephen Wright

University of Wisconsin-Madison

Caltech, 21 April 2007

+Mario Figueiredo, Rob Nowak, Mingqiang Zhu, Tony Chan.

Motivation

Many applications give rise to optimization problems for which *simple, approximate* solutions are required, rather than complex exact solutions.

- Occam's Razor
- Data quality doesn't justify exactness
- Possibly more robust to data perturbations (not "overoptimized")
- Easier to actuate / implement / store simple solutions
- Conforms better to prior knowledge.

When formulated with variable $x \in \mathbb{R}^n$, simplicity is often manifested as sparsity in x (few nonzero components) or in a simple transformation of x.

Formulating Sparse Optimization

Two basic ingredients:

- The underlying optimization problem — often of data-fitting type
- Regularization term or constraints to "encourage" sparsity — often nonsmooth.

Usually very large problems. Need techniques from

- Large-scale optimization
- Nonsmooth optimization
- Lots of domain-specific knowledge.

Example: Regularized Logistic Regression

Have attribute vectors $x(1), x(2), \ldots, x(n)$ (real vectors) and labels $y(1), y(2), \ldots, y(n)$ (binary 0/1).

Probability of outcome $Y = 1$ given attribute vector X is $p(X) = E(Y = 1|X)$. Model *log odds* or *logit* function as linear combination of basis functions of x:

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \sum_{l=0}^{N} a_l B_l(x),$$

for a (possibly huge) number of basis functions B_l.

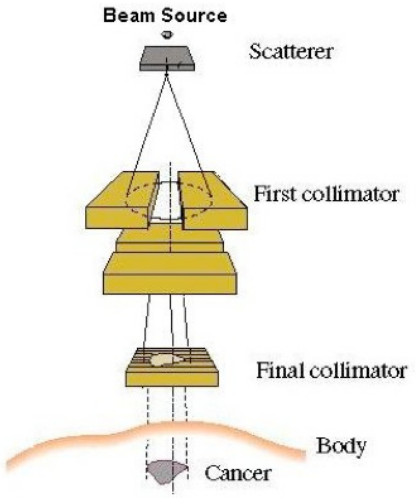
Define a log-likelihood function based on observations:

$$\frac{1}{n}\sum_{i=1}^{n}\left[y(i)\log p(x(i)) + (1-y(i))\log(1-p(x(i)))\right].$$

Choose coefficients $(a_1, a_2, \ldots, a_N)$ sparsely to approximately maximize this function. [Wahba-Shi-SW et al, 2008]

Example: Radiotherapy for Cancer

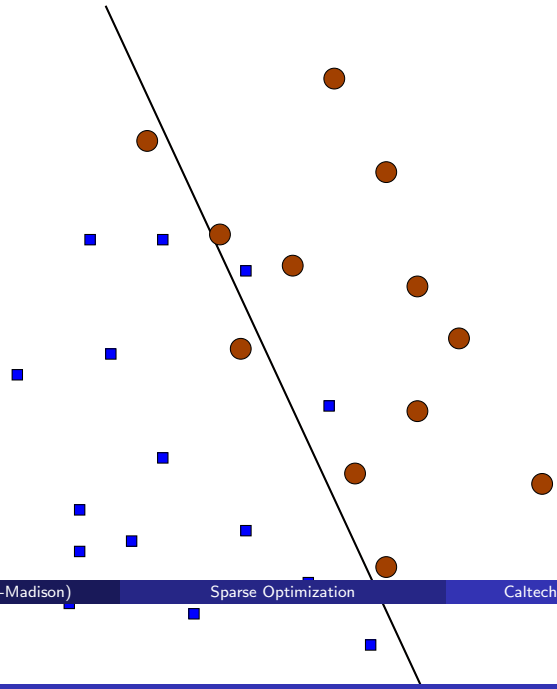
Multileaf collimator. Leaves move up and down to shape the beam.



Linear accelerator, showing cone and collimators

- Irradiate tumor while sparing surrounding tissue and critical organs
- Choose angles of delivery and aperture shapes to match the oncologist's specifications (many possible configurations)
- Many optimization formulations used: linear, quadratic, second-order cone, nonlinear, integer programs.
- Uncertainty in the model:
  - dose distribution
  - cancer / organ location and movement.
- Want a sparse, approximate, robust solution that can be delivered efficiently using few aperture shapes/angles.

Example: Support Vector Machines



- Have attribute vectors $x(1), x(2), \ldots, x(n)$ (real vectors) and labels $y(1), y(2), \ldots, y(n)$ (binary 0/1).
- See a hyperplane that separates the points according to their classification — usually after transforming attributes to higher dimensional space (using a kernel).
- The usual optimization formulation has dimension n, with number of nonzeros equal to number of misclassified points.

Example: Image Processing

TV Denoising

Rudin-Osher-Fatemi (ROF) model. Given a domain $\Omega \subset \mathbb{R}^2$ and an observed image $f : \Omega \to \mathbb{R}$, seek a restored image $u : \Omega \to \mathbb{R}$ that preserves edges while removing noise.

The regularized image u can typically be stored more economically.

Seek to "minimize" both

- $\|u - f\|_2$ and
- the total-variation (TV) norm $\int_\Omega |\nabla u| \, dx$.

Use constrained formulations, or a weighting of the two objectives:

$$\min_u \; P(u) := \int_\Omega |\nabla u| \, dx + \frac{\lambda}{2} \|u - f\|_2^2.$$

Minimizing u tends to have regions in which u is constant ($\nabla u = 0$). More pronounced for small λ.

More later...

Example: Compressed Sensing

- Many signals are close to being *sparse* when the correct representation is used; that is,
- When expressed as a linear combination of basis elements, few of the coefficients are nonzero.

Rather than sample the signal as though it were possibly dense, we can take a much smaller set of well chosen samples and use these to reconstruct the nonzero coefficients of the (approximate) representation.

Allows big savings in cost of sensing and possibly storage, but novel formulations and algorithms are needed to extract the sparse representation.

The optimization perspective is proving to be useful.

Given a signal $x \in \mathbb{R}^n$ with at most S nonzero components, observe the inner products of x with some chosen vectors $a_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, k$, plus some noise. Observations y_i are therefore

$$y_i = a_i^T x + e_i,$$

(where e_i is the noise term). Defining

$$y := [y_i]_k^{i=1} \in \mathbb{R}^k, \qquad A := \left[a_i^T\right]_k^{i=1} \in \mathbb{R}^{k \times n}, \qquad e := [e_i]_k^{i=1} \in \mathbb{R}^k,$$

we have

$$y = Ax + e.$$

- How to recover x, given y and A (and possibly some information about e)?
- What properties does A need to have?
- How big does k need to be?

Choosing A

The choice of vectors a_i, $i = 1, 2, \ldots, k$ (and hence A) is critical.

A Bad Idea: If a_i have the form $(0, \ldots, 0, 1, 0, \ldots, 0)^T$, i.e. we randomly sample components of x — we won't recover all the nonzero components of x with reasonable probability until $k \gg n$.

For other kinds of random A, much smaller values of k will give enough information to recover x, as closely as e allows, to high probability. The critical property is restricted isometry [Candès, Tao].

Given sparsity level $S \leq k$, A satisfies the restricted isometry property with isometry constant $\delta_S < 1$ if for any column submatrix $A_{\cdot T}$ of A with at most S columns, we have

$$(1 - \delta_S)\|c\|_2^2 \leq \|A_{\cdot T} c\|_2^2 \leq (1 + \delta_S)\|c\|_2^2, \qquad \text{for all } c \in \mathbb{R}^S.$$

That is, $A_{\cdot T}$ has close-to-orthonormal columns.

Formulating the Recovery Problem

Similarly to TV denoising, seek to control both

- Fit between observation and model: $\|Ax - y\|_2^2$
- ℓ_1-norm of signal: $\|x\|_1$.

The term $\|x\|_1$ serves as surrogate for $\|x\|_0$ (which is a count of the number of nonzero coefficients), but

- $\|x\|_1$ is convex and can lead to smooth convex formulations;
- Problems with $\|x\|_1$ often give the same (sparse) solutions as those with $\|x\|_0$!

A term $\|x\|_2$ does not have the latter property.

Three Formulations

LASSO with parameter $\beta > 0$:

$$\min \frac{1}{2}\|Ax - y\|_2^2 \qquad \text{subject to } \|x\|_1 \leq \beta.$$

Reconstruction with noise bound ϵ:

$$\min \|x\|_1 \qquad \text{subject to } \|Ax - y\|_2 \leq \epsilon.$$

Unconstrained nonsmooth formulation with regularization $\tau > 0$.

$$\min \frac{1}{2}\|Ax - y\|_2^2 + \tau\|x\|_1.$$

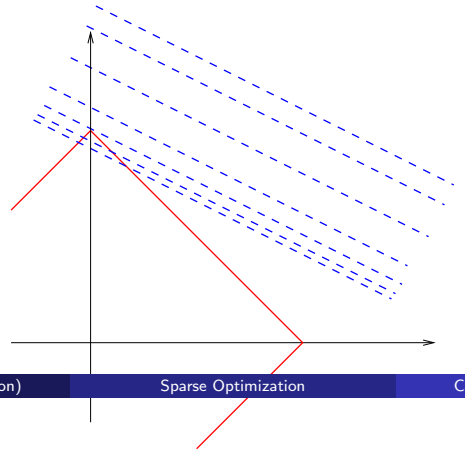
- By varying their parameters, all three formulations generally lead to the same path of solutions.
- The "correct" choice of parameter usually is not known a priori; need to solve for a selection or range of values and choose it in some "outer loop."

Why Does $\|x\|_1$ Work?

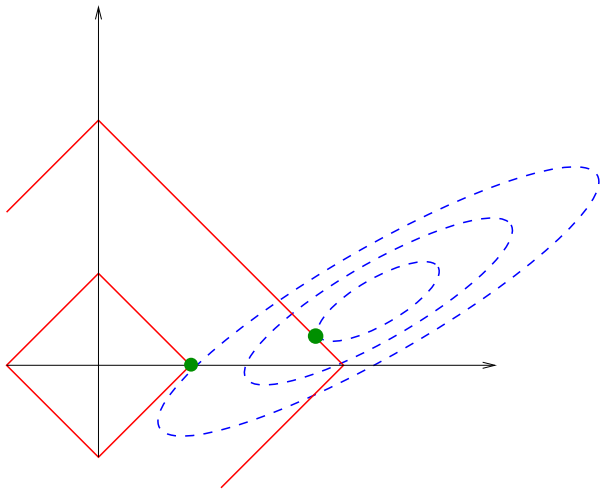
Can give some more or less intuitive justifications as to why the $\|x\|_1$ term promotes sparsity in the solution. Consider LASSO formulation

$$\min \frac{1}{2}\|Ax - y\|_2^2 \qquad \text{subject to } \|x\|_1 \leq \beta,$$

where $A \in \mathbb{R}^{k\times n}$ with $k < n$: in particular $k = 1$, $n = 2$:



$k = 2$, $n = 2$. Solution has a single nonzero for small β, two nonzeros for larger β.



A View from the Dual

[Nesterov, 2007] The formulation

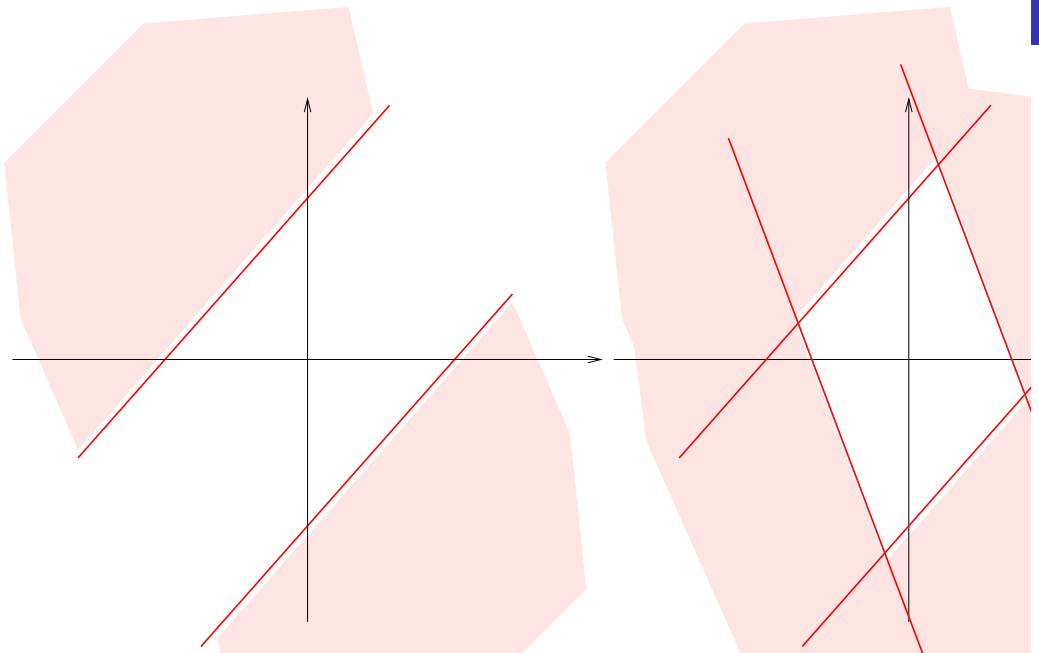
$$\min \frac{1}{2}\|Ax - y\|_2^2 + \tau\|x\|_1$$

has dual

$$\min_u \|u - y/\tau\|_2^2 \qquad \text{subject to } \|A^T u\|_\infty \leq 1.$$

- Components of x are Lagrange multipliers for the constraints $\|A^T u\|_\infty \leq 1$.
- The number of nonzero components of x corresponds to the number of active facets in this constraint set.

We illustrate a case of $k = 2$, $n = 3$.



Compressed Sensing Algorithms

Many algorithms and heuristics have been proposed for all three of the $\ell_2 - \ell_1$ formulations of compressed sensing. The problem has certain properties that drive certain algorithmic choices:

- n very large, possibly also k.
- A often dense, can't store substantial submatrices explicitly (but a small column submatrix may be OK).
- A often a product of a representation matrix and an observation matrix;
  - Representation matrix (e.g. wavelet basis, FFT) often dense, but allows fast matrix-vector multiplies;
- The solution x is sparse, for interesting choices of regularization parameter.
- Often want to solve for a selection of regularization values.

Interior-Point Algorithms

ℓ_1-magic: Log-barrier (primal interior-point) approach for the second-order cone program formulation: min $\|x\|_1$ s.t. $\|Ax - y\|_2 \leq \epsilon$ [Candès, Romberg]:

- Newton method used for inner iteration
- CG used for inner-inner iteration (if A not explicitly available).

`l1_ls`: Apply a log-barrier method to a reformulation of the unconstrained problem:

$$\min \frac{1}{2}\|Ax - y\|_2^2 + \tau \mathbf{1}^T u \text{ subject to } -u \leq x \leq u.$$

Preconditioned CG used for the inner loop. [Kim et al, 2007]

SparseLab/PDCO: Primal-dual formulation, with linear equations solved iteratively with LSQR for large A. [Saunders, 2002]

Interior-Point Properties

- Generally few outer iterations, but expensive.
- Linear systems at innermost level become increasingly ill conditioned.
  - Requires many more CG / LSQR iterations.
  - Clever preconditioning can help.
- Difficult to warm-start.
  - No big savings from using the solution for one value of τ to warm-start for the next value in the sequence.

Matching Pursuit and Descendants

MP, OMP heuristics build up x one component at a time, in a greedy fashion.

CoSaMP [Needell, Tropp, 2008] extends this idea, adding ideas from other approaches, and adds includes a convergence theory.

(CoSaMP is related to a constraint generation strategy for the ℓ_2-ℓ_1 problem above.)

See Joel for details!

Active-Set / Pivoting Methods

LARS / LASSO: trace the solution path for a range of values of the regularization parameter.

For the formulation

$$\min \frac{1}{2}\|Ax - y\|_2^2 + \tau\|x\|_1$$

the solution is $x = 0$ for $\tau \geq \|A^T y\|_\infty$. Can decrease τ progressively from this value, seeking *breakpoints* at which another component of x moves away from zero.

The approach can be implemented carefully in a way that requires only matrix-vector multiplications with A and A^T, and storage of the "active" columns of A. Possibly suitable for very sparse signals.

`SolveLasso` function in the SparseLab toolbox.

QP Formulation and Gradient Projection

Can formulate as bound-constrained least squares by splitting

$$x = u - v, \quad (u, v) \geq 0,$$

and writing

least sq: $$\min_{u\geq 0, v\geq 0} \phi(u, v) := \frac{1}{2}\|A(u - v) - y\|_2^2 + \tau \mathbf{1}^T u + \tau \mathbf{1}^T v.$$

Gradient of objective is

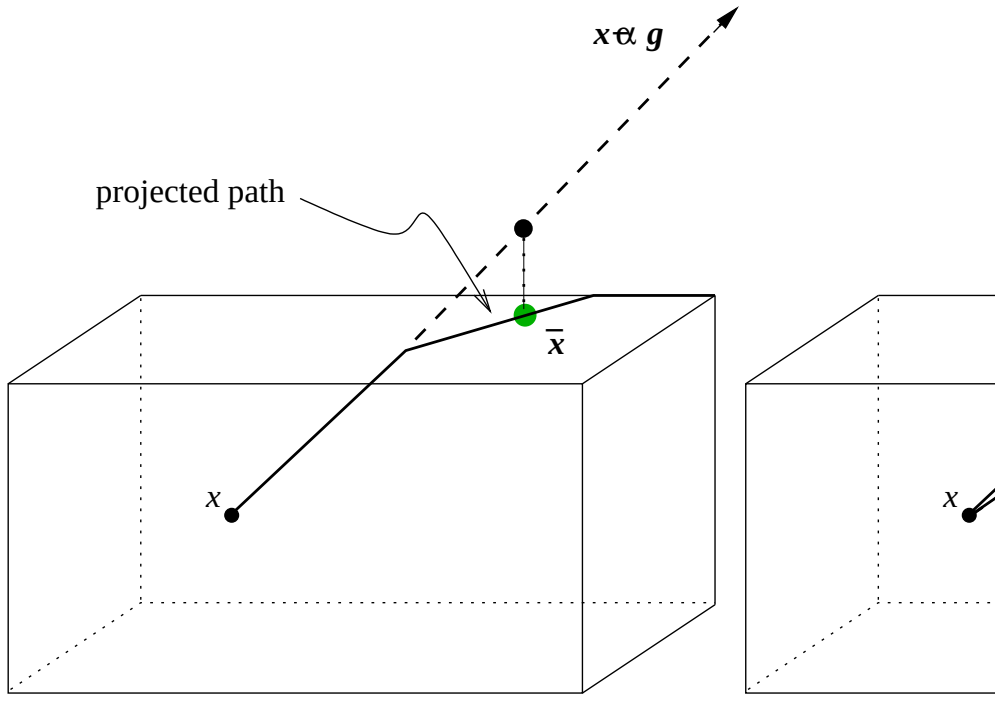
$$\begin{bmatrix} \nabla_u \phi(u, v) \\ \nabla_v \phi(u, v) \end{bmatrix} = \begin{bmatrix} A^T A(u - v) - A^T y + \tau \mathbf{1} \\ -A^T A(u - v) + A^T y + \tau \mathbf{1} \end{bmatrix}.$$

Set

$$(\bar{u}^{k+1}, \bar{v}^{k+1}) = \left[(u^k, v^k) - \alpha(\nabla_u \phi^k, \nabla_v \phi^k)\right]_+$$

for $\alpha > 0$. Then possibly do a second "internal" line search, choosing $\gamma \in [0, 1]$ to reduce ϕ, and setting

$$(u^{k+1}, v^{k+1}) = \left[(u^k, v^k) + \gamma\left\{(\bar{u}^{k+1}, \bar{v}^{k+1}) - (u^k, v^k)\right\}\right]_+.$$



Variants: Choice of α_k and γ_k

- α minimizes ϕ along projection of $-\nabla\phi$ onto the current face ("GPSR-Basic" variant).
- Barzilai-Borwein: Choose α_k to mimic behavior of inverse Hessian over the step just taken. Then step mimics a Newton step:

$$-\alpha_k \nabla\phi \sim -[\nabla^2\phi]^{-1}\nabla\phi.$$

  e.g., do a least-squares fit to get α_k^{-1}:

$$\left[(u^k, v^k) - (u^{k-1}, v^{k-1})\right] \approx \alpha_k^{-1}\left[\nabla\phi(u^k, v^k) - \nabla\phi(u^{k-1}, v^{k-1})\right].$$

  Many variations (e.g. cyclic).
- Can accept α even if it increases ϕ (nonmonotone), or backtrack until ϕ is reduced, by setting $\alpha_k \leftarrow \alpha_k/2$ repeatedly.

For choice of γ_k, use either

- $\gamma_k = 1$, or
- γ_k is the minimizer of ϕ along the line from (u^k, v^k) to $(\bar{u}^{k+1}, \bar{v}^{k+1})$ (gives monotonic decrease in ϕ).

Gradient Projection Properties

- Can make large changes to the active manifold on a single step (like interior-point, unlike pivoting).
- Each iteration is cheap: 2-3 multiplications with A or A^T.
- Would reduce to steepest descent if there were no bounds.
- For very sparse problems (large τ) can sometimes identify the correct active set in few iterations.
- Once the correct nonzero components of x are identified, the approach reduces to steepest descent on subspace of nonzero components.
- Benefits from warm starting.

Final Stages, Debiasing

When the support of x has been identified correctly for a given τ, GPSR reduces to steepest descent on a convex quadratic, on the reduced space of nonzero x_j.

This quadratic has Hessian $\bar{A}^T\bar{A}$, where $\bar{A}$ is the column submatrix of A corresponding to support of x. When the support is small and the restricted isometry property holds, $\bar{A}^T\bar{A} \approx I$, so steepest descent is not too slow.

GPSR optionally postprocesss x with a debiasing step: Discard the regularization term, and apply conjugate gradient (CG) to the reduced problem. Since $\bar{A}^T\bar{A} \approx I$, CG converges rapidly.

(Similar observations are key to analysis of CoSaMP.)

Larger Support: Continuation Strategy

When the support is not so sparse, GPSR is much slower to both identify the correct support and to converge in its final stages.

- How interesting are such problems? Are there interesting practical problems for which we care about instances like this?
- Other approaches are also slower on these cases.

Can alleviate with a continuation strategy: Solve for a decreasing sequence of τ values:

$$\tau_1 > \tau_2 > \cdots > \tau_m,$$

using the solution for τ_i to warm-start for τ_{i+1}.

- If you really want the solution for τ_m only, need only *approximate* solutions for $\tau_1, \tau_2, \ldots, \tau_{m-1}$.
- Typically faster than solving for τ_m alone from a cold start.

SpaRSA: Separable Approximation

Formulation:

$$\min \frac{1}{2}\|Ax - y\|_2^2 + \tau\|x\|_1.$$

From iterate x^k, get step d by solving

$$\min_d \nabla q(x^k)^T d + \frac{1}{2}\alpha_k d^T d + \tau\|x^k + d\|_1.$$

Can view the α_k term as

- an approximation to the Hessian: $\alpha_k I \approx \nabla^2 q = A^T A$;
- Lagrange multiplier for a trust-region constraint $\|d\|_2 \leq \Delta$.

Subproblem is trivial to solve in $O(n)$ operations, since it is separable in the components of d.

Related to GPSR, also to previously proposed approaches, e.g. iterative shrinking-thresholding, proximal forward-backward splitting [Combettes]. Main difference is adaptive choice of α_k.

SpaRSA Variants and Properties

- IST makes a large, constant choice $\alpha_k \equiv \alpha$ (or at least requires all α_k to be greater than α).
- Can choose α_k using Barzilai-Borwein strategies.
- Not obvious to extend the "basic" GPSR strategy for choosing α_k to SpaRSA.
- Can get monotone variants by doing backtracking $\alpha_k \leftarrow \alpha_k/2$ until the objective is decreased (sufficiently).
- Generalizes nicely to regularizers other than ℓ_1:
  - sum-of-ℓ_2;
  - sum-of-ℓ_∞;
  - hierarchical regularization terms?

  In each case the subproblem decomposes; closed-form solutions can be found in $O(n)$ time.
- Continuation strategy can be used.

Nesterov's Primal-Dual Approach

- Solves subproblems of same type as SpaRSA.
- For a technique like SpaRSA that directly manipulates α_k, proves convergence of the objective function to its optimal value at rate k^{-1}.
- Proposes a more complex "accelerated" scheme in which each iterate z^k is a linear combination of two vectors:
  - An vector x^k obtained from the SpaRSA subproblem
  - An vector v^k obtained from a subproblem with a modified linear term (a weighted average of gradients $A^T(Ax - y)$ encountered at earlier iterations.
- Similar methods known to engineers as *two-step* and *heavy-ball* methods.
- Proves convergence of objective value at rate k^{-2}.

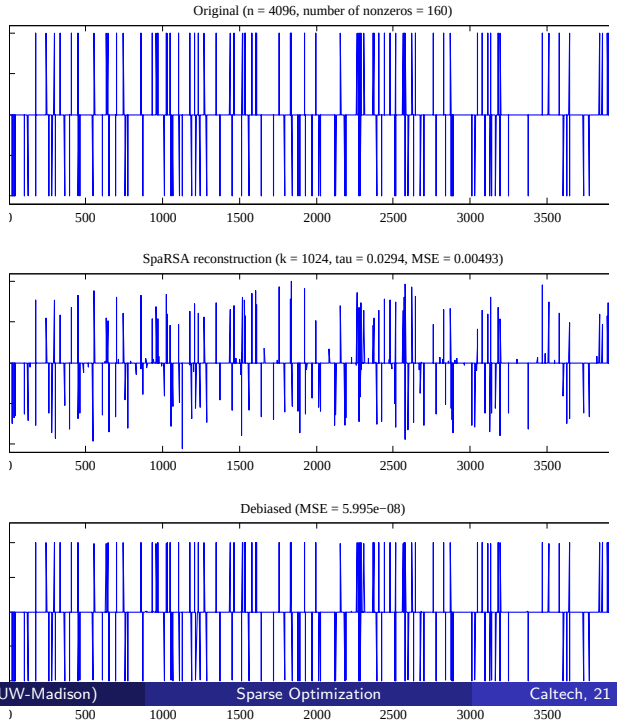
Computational Results

A small explicit problem with an easy signal (not very sparse).

- A is 1024×4096, elements from $N(0, 1)$.
- True signal x has 160 nonzeros with value ± 1.
- Observations y include noise of variance $\sigma^2 = 10^{-4}$.
- Choose $\tau = 0.1\|A^T y\|_\infty$ — sufficient to recover the signal accuracy (after debiasing). Continuation not needed for this value.

Compare several codes:

- FPC [Hale, Yin, Zhang, 2007]
- `l1_ls` [Kim et al, 2007]
- SpaRSA: monotone and nonmonotone, BB selection of initial α_k.
- GPSR: monotone and "basic"
- Nesterov's accelerated scheme.
- IST

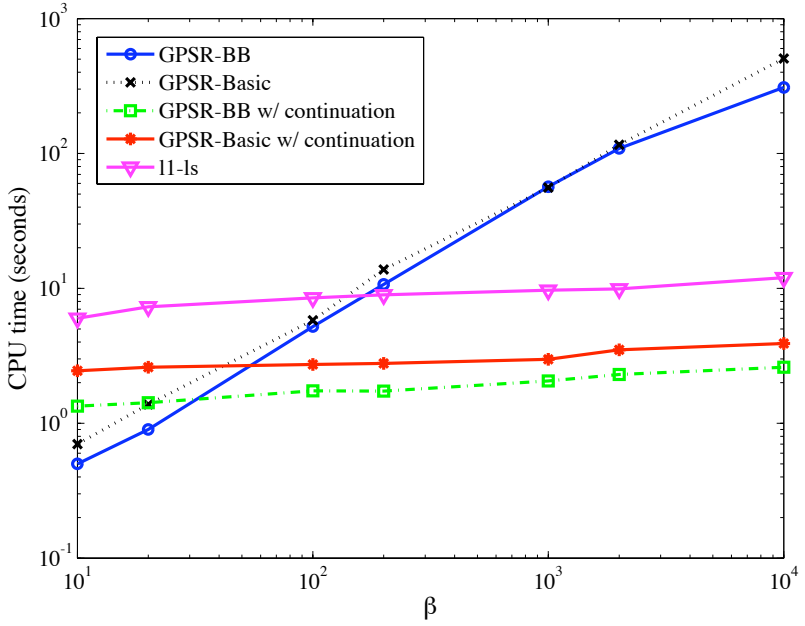


	iterations	time
`l1_ls`	13	10.5
FPC	111	2.47
IST	55	1.35
GPSR (monotone)	52	1.15
GPSR (basic)	39	1.64
SpaRSA (monotone)	35	.90
SpaRSA (nonmonotone)	33	.72
Nesterov-AC	157	17.57

Table: Results for Spikes test problem (times in secs on a MacBook)

Effectiveness of Continuation

- Tested a similar example for different values of τ with continuation turned on/off.
- Plot total runtime against $\beta = \|A^T y\|_\infty / \tau$.
- Benchmarked against `l1_ls`, whose runtimes are less sensitive to this value.
- Showed large advantage for continuation over a one-off approach, for GPSR codes. (SpaRSA results are similar.)



MRI Data: Sum-of-ℓ_2 Regularizer: Comparison with EM

Regularization term of the form $\sum_{l=1}^{M} \|x_{[l]}\|_2$, for a disjoint collection of subvectors $x_{[l]}$.

Table: Computational Results From $X = 0$ Starting Point. Times in seconds.. $*$maximum iteration count reached.

	EM		SpaRSA		SpaRSA-monotone				
τ	its	time	its	time	its	evals	time	final cost	blocks
0.7	8961	2464.	60	18.	30	53	14.	1.6(−6)	2
0.5	10000*	2749.*	90	26.	80	129	34.	1.5(−6)	4
0.4	10000*	2754.*	90	26.	70	117	31.	1.3(−6)	4
0.3	life's too short		210	60.	140	248	64.	1.2(−6)	4
0.2	to run these		360	102.	210	369	95.	9.6(−7)	8

TV-Regularized Image Denoising

Work with the formulation:

$$\min_u P(u) := \int_\Omega |\nabla u| \, dx + \frac{\lambda}{2}\|u - f\|_2^2.$$

The first term is the (nonsmooth) TV-reg term, while the second is the data-fitting term.

Can't apply GPSR or SpaRSA approaches directly, after discretization, as

- Doesn't seem possible to come up with a constrained formulation with a feasible set that allows easy projection (needed for GPSR);
- The SpaRSA subproblem has the same form as the original problem (since $A^T A = \lambda I$) and hence just as hard to solve.

However, if we discretize and take the dual we obtain a problem amenable to gradient-projection approaches.

Dual Formulation

Redefine the TV seminorm:

$$\int_\Omega |\nabla u| = \max_{w \in C_0^1(\Omega), |w| \le 1} \int_\Omega \nabla u \cdot w = \max_{|w| \le 1} \int_\Omega -u \nabla \cdot w.$$

Rewrite the primal formulation as

$$\min_u \max_{w \in C_0^1(\Omega), |w| \le 1} \int_\Omega -u \nabla \cdot w + \frac{\lambda}{2}\|u - f\|_2^2.$$

Apply min-max theorem to exchange min and max, and do the inner minimization wrt u explicitly:

$$u = f + \frac{1}{\lambda} \nabla \cdot w.$$

Thus obtain the dual:

$$\max_{w \in C_0^1(\Omega), |w| \le 1} D(w) := \frac{\lambda}{2} \left[\|f\|_2^2 - \left\| \frac{1}{\lambda} \nabla \cdot w + f \right\|_2^2 \right].$$

Discretization

Assume $\Omega = [0,1] \times [0,1]$, discretization with an $n \times n$ regular grid, where u_{ij} approximates u at

$$\begin{bmatrix} (i-1)/(n-1) \\ (j-1)/(n-1) \end{bmatrix} \in \Omega.$$

The discrete approximation to the TV norm is thus

$$\mathrm{TV}(u) = \sum_{1 \le i,j, \le n} \|(\nabla u)_{i,j}\|,$$

where

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < n \\ 0 & \text{if } i = n \end{cases}$$

$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < n \\ 0 & \text{if } j = n. \end{cases}$$

By reorganizing the $N = n^2$ components of u into a vector $v \in \mathbb{R}^N$, and f into a vector $g \in R^N$, we write the discrete primal ROF model as

$$\min_v \sum_{l=1}^{N} \|A_l^T v\|_2 + \frac{\lambda}{2}\|v - g\|_2^2,$$

where A_l is an $N \times 2$ matrix with at most 4 nonzero entries ($+1$ or -1).

Introduce a vector representation $x \in \mathbb{R}^{2N}$ of $w : \Omega \to \mathbb{R}^2$. Obtain the discrete dual ROF (scaled and shifted):

$$\begin{aligned} &\min_{x \in X} \frac{1}{2}\|Ax - \lambda g\|_2^2 \\ &\text{where} \quad X := \{(x_1; x_2; \ldots; x_N) \in \mathbb{R}^{2N} : x_l \in \mathbb{R}^2, \\ &\qquad\qquad \|x_l\|_2 \leq 1 \text{ for all } l = 1, 2, \ldots, N\}, \end{aligned}$$

where $A = [A_1, A_2, \ldots, A_N] \in \mathbb{R}^{N \times 2N}$.

Set $X \subset \mathbb{R}^{2N}$ is a Cartesian product of N unit balls in $\mathbb{R}^2$. Projections onto X are trivial. Can apply similar gradient projection ideas as in GPSR. (Curvature of the boundaries of X adds some interesting twists.)

The discrete primal-dual solution (v, x) is a saddle point of

$$\ell(v, x) := x^T A^T v + \frac{\lambda}{2}\|v - g\|_2^2.$$

Since the discrete primal is strictly convex, we have:

Proposition. *Let $\{x^k\}$ be any sequence in X whose accumulation points are all stationary for the dual problem. Then $\{v^k\}$ defined by*

$$v^k = g - \frac{1}{\lambda} A x^k$$

converges to the unique solution of the primal problem.

Fortunately, we can prove that the required property of $\{x^k\}$ holds for many gradient projection algorithms.

Previous Methods

- Embedding in a parabolic PDE [ROF, 1992]
- Apply Newton-like method to the optimality conditions for a smoothed version, in which $|\nabla u|$ is replaced by $\sqrt{|\nabla u|^2 + \beta}$. Parameter $\beta > 0$ is decreased between Newton steps (path-following). [Chan, Golub, Mulet, 1999]
- Semismooth Newton on a perturbed version of the optimality conditions. [Hintermüller, Stadler, 2006]
- SOCP [Goldfarb, Yin, 2005].
- First-order method similar to gradient projection with fixed step size. [Chambolle, 2004]

Gradient Projection Variants

$$\min_{x \in X} F(x), \qquad \text{where} \quad F(x) := \frac{1}{2}\|Ax - \lambda g\|_2^2$$

GP methods choose α_k and set

$$x^k(\alpha_k) := P_X(x^k - \alpha_k \nabla F(x^k)),$$

then choose $\gamma_k \in (0, 1]$ and set

$$x^{k+1} := x^k + \gamma_k (x^k(\alpha_k) - x^k).$$

Choosing α_k and γ_k:

- $\alpha_k \equiv \alpha$ constant, converges for $\alpha < 0.25$.
- Barzilai-Borwein formulae; cyclic variants; alternating variants that switches adaptively between the formulae.
- $\gamma_k \equiv 1$ (non-monotone) or γ_k minimizes F in $[0, 1]$ (monotone).

Sequential Quadratic Programming

Optimality conditions for the dual: There are Lagrange multipliers $z_l \in \mathbb{R}$, $l = 1, 2, \ldots, N$, such that

$$A_l^T(Ax - \lambda g) + 2z_l x_l = 0, \qquad l = 1, 2, \ldots, N,$$
$$0 \le z_l \perp \|x_l\|^2 - 1 \le 0.$$

At iteration k, define the active set $\mathcal{A}_k \subset \{1, 2, \ldots, N\}$ as the l for which $\|x_l^k\| = 1$, and do a Newton-like step on the system

$$A_l^T(Ax - \lambda g) + 2x_l z_l = 0, \qquad l = 1, 2, \ldots, N,$$
$$\|x_l\|_2^2 - 1 = 0, \qquad l \in \mathcal{A}_k,$$
$$z_l = 0, \qquad l \notin \mathcal{A}_k.$$

Still use the Hessian approximation $A^T A \approx \alpha_k^{-1} I$.

Leads to the formula

$$\Delta x_l^k = -\left(1/\alpha_k + 2z_l^k\right)^{-1} \left\{[\nabla F(x^k)]_l + 2z_l^k x_l^k\right\}, \qquad l = 1, 2, \ldots, N.$$

Computational Results

- Two images: SHAPE (128×128) and CAMERAMAN (256×256).
- Gaussian noise added with variance .01.
- $\lambda = 0.045$ for both examples.

Tested many variants. Report here on

- Chambolle, with $\alpha \equiv .248$
- Nonmonotone GPBB
- Nonmonotone GBPP with SQP augmentation
- GPABB - alternating adaptively between BB formulae
- CGM with adaptively decreasing β.

Convergence declared when relative duality gaps falls below `tol`.

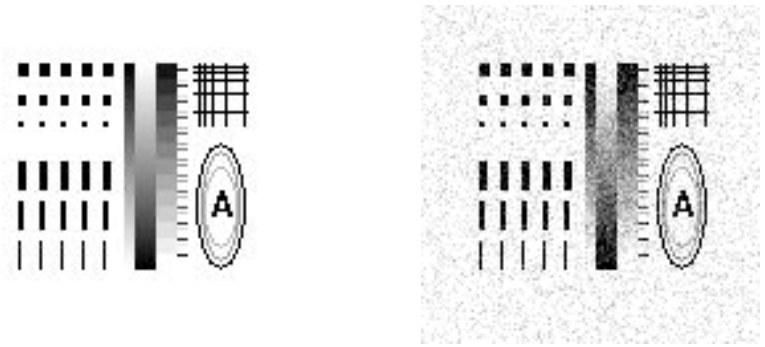


Figure: SHAPE: original (left) and noisy (right)

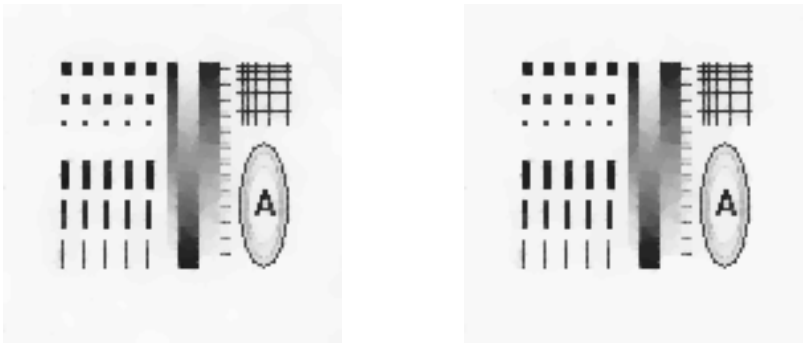


Figure: Denoised SHAPE: Tol=10^{-2} (left) and Tol=10^{-4} (right).

Little visual difference between loose and tight stopping criterion: "convergence in the eyeball norm."

SHAPE Results

	tol=10^{-2}		tol=10^{-3}		tol=10^{-4}		tol=10^{-5}	
Alg	its	time	its	time	its	time	its	time
Chambolle	18	0.22	168	1.97	1054	12.3	7002	83.4
GPBB-NM	10	0.18	48	0.79	216	3.6	1499	25.9
GPCBBZ-NM	10	0.24	50	1.12	210	4.7	1361	31.5
GPABB	13	0.29	57	1.20	238	5.0	1014	22.6
CGM	6	5.95	10	10.00	13	12.9	18	19.4

Table: Runtimes (MATLAB on MacBook) for Denoising Algorithms

- Nonmonotone GPBB generally reliable. Most GPBB variants dominate Chambolle.
- CGM becomes the fastest between 10^{-4} and 10^{-5}.

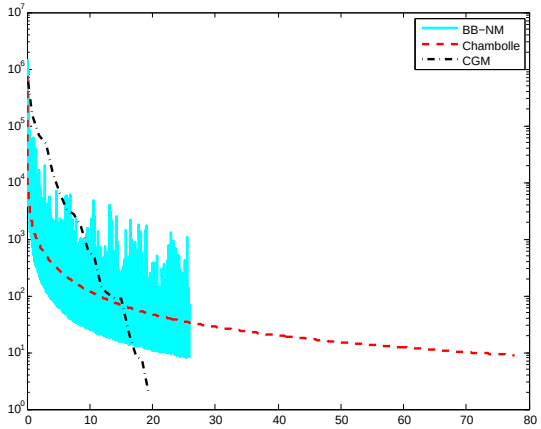


Figure: CAMERAMAN: original (left) and noisy (right)

Figure: Denoised CAMERAMAN: Tol=10^{-2} (left) and Tol=10^{-4} (right).

CAMERAMAN Results

	tol=10^{-2}		tol=10^{-3}		tol=10^{-4}		tol=10^{-5}	
Alg	its	time	its	time	its	time	its	time
Chambolle	27	1.07	163	6.46	827	32.8	3464	137.8
GPBB-NM	16	0.86	48	2.59	183	9.7	721	39.2
GPCBBZ-NM	16	1.17	53	3.91	202	14.6	729	53.8
GPABB	16	1.06	47	3.11	179	11.9	563	37.4
CGM	6	12.53	10	21.15	14	29.7	16	34.1

Table: Runtimes (MATLAB on Linux PC) for Denoising Algorithms