Optimization for Sparse Solutions, A Tutorial

Wotao Yin (CAAM, Rice U.)

MIIS 2012, Xi’an, China
Structure means simplicity

Examples

- signals that are sparse (left figure) or sparse under bases, frames, or transforms
- a set of jointly sparse signals
- low rank matrix (right figure); low rank + sparse; low rank + jointly sparse
- low dimensional manifold

More structures will be discovered and used
Why *structured solutions*?

- cheap to store / transmit
- more meaningful, easy to understand,
- easy to use
- more robust to errors
- need fewer data to begin with
- easy to find by optimization (both speed and storage)
Minimization $\ell_1$ has a long history in data processing (geophysical: Claerbout and Muir’73, Santosa and Symes’86), image processing (Rudin, Osher, and Fatemi’92), and statistics (Chen, Donoho, and Saunders’98; Tibshirani’96).

I learned from S. Osher that Galileo Galilei (1564–1642) used the median filter, which for given vector $y$ solves

$$
\min_x \|x - y\|_1.
$$

Heuristically, $\ell_1$ is known as a good convex approximate (convex envelop) of $\ell_0$, and $\ell_1$ tends to give sparse solutions. This is so widely known that many people would just try but not bother with checking the theory.
\( \ell_1 \) gives sparse solutions

Goal: to recover sparse vector \( x^0 \) from \( b = Ax^0 \)

- Support \( S := \text{supp}(x^0) \)
- Zero set \( Z := S^c \)
- Sparsity \( k = \|x^0\|_0 = |\text{supp}(x^0)| \).

For \( x^0 \) to be the solution of

\[
\min_x \|x\|_1, \text{ subject to } Ax = b,
\]

we need

\[
\|x^0 + v\|_1 \geq \|x^0\|_1, \forall v \in \mathcal{N}(A).
\]
When the diamond-shaped $\ell_1$-ball touches the sparse $x^0$, we need the affine space $\{x : Ax = b\} = \{x^0 + v : v \in \mathcal{N}(A)\}$ to not cut through the $\ell_1$-ball.

$x^0$ is recovered, good

$x^0$ is not recovered, bad
\[
\|x^0 + v\|_1 = \|x_S^0 + v_S\|_1 + \|v_Z\|_1 \\
\geq \|x_S^0\|_1 + \|v_Z\|_1 - \|v_S\|_1 \\
= \|x^0\|_1 + \|v\|_1 - 2\|v_S\|_1 \\
\geq \|x^0\|_1 + \|v\|_1 - 2\sqrt{k}\|v\|_2.
\]

Therefore, \(x^0\) is \(\ell_1\)-recoverable provided

\[
\frac{\|v\|_1}{\|v\|_2} \geq 2\sqrt{k}, \quad \forall v \in \mathcal{N}(A), \ v \neq 0.
\]

In general, we only have

\[
1 \leq \frac{\|v\|_1}{\|v\|_2} \leq \sqrt{n}.
\]
However, $\frac{\|v\|_1}{\|v\|_2}$ is significantly greater than 1 if $V = \mathcal{N}(A)$ is random.

Kashin’77, Garvaev and Gluskin’84 show that a randomly drawn $(n - m)$-dimensional subspace $V$ of $\mathbb{R}^n$ satisfies

$$\frac{\|v\|_1}{\|v\|_2} \geq \frac{c_1 \sqrt{m}}{\sqrt{1 + \log(n/m)}}, \quad \forall v \in V, v \neq 0$$

with probability at least $1 - \exp(-c_0(n - m))$, where $c_0, c_1 > 0$ are indep. of $m$ and $n$.

Hence, $x^0$ is $\ell_1$-recoverable with high probability if $A$ is random and

$$k \leq O(m \log^{-1}(n/m)) \quad \text{or} \quad m \geq O(k \log(n/k)),$$

The number of measurements $m$ is a multiple of $k \log(n/k)$ — typically much less than $n$. 
Other Analyses

- Gorodnitsky, Rao’97: Spark – Smallest number of columns of $A$ that are linearly dependent.
- Donoho, Elad’03: Coherence – Smallest angles between any two columns of $A$.
- Donoho, Tanner’06: analysis similar to the above
- Candes, Romberg, and Tao’04 ’06: Restricted Isometry Property (RIP) – any $2k$-column submatrices of $A$ must be almost orthogonal.
- RIP $\Rightarrow \ell_1$ recovery: above, Cohen-Dahmen-DeVore, Forcart-Lai, Cai-Wang-Xu, Li-Mo, ...
- More ......
See Rice Compressed Sensing Repository: http://dsp.rice.edu/cs

- Signal sensing and processing, image processing
- Medical imaging (faster imaging, less radiation, multi-modality)
- (Wireless) communications (joint sparsity, channel estimation)
- Multi/hyperspectral imaging (spectral+spatial structure, data to information)
- Statistics, machine learning (feature selection; LASSO, logistic regression, SVM)
- Geophysical (seismic) data analysis
- Face recognition, video processing (robust PCA: common structure + sparse difference)
- Distributed computation / cloud computing (terabytes of data are spread out, centralized computing is infeasible, sparse recovery at minimal communication cost?)

Many applications require *nonnegativity, structured (model) sparsity, complex data.*
First-order: (sub)gradient descent, gradient projection, continuation, coordinate descents, prox-point, conjugate gradient, forward backward splitting, conjugate gradient, accelerated first-order

High-order: interior-point, semismooth Newton, hybrid 1st and 2nd order, BFGS, L-BFGS

Final debiasing

Duality: augmented Lagrangian, alternating direction, Bregman (original, linearized, split), primal-dual methods. They are applied to both primal and dual formulations.

Homotopy: parametric quadratic programming
Optimization/Algorithmic Techniques Used, Part 2

- Greedy algorithms (OMP family), support detection
- Non-convex approximations to $\ell_0$, $\ell_q$ minimization, reweighted itr
- Stochastic approximation (gradient, Hessian), sample-average approx
- Distributed reconstruction
- Parallel and GPU-friendly algorithms
- Numerical linear algebra
- Heuristics. Solution structure may be domain specific.

Sources: Optimization Online and Rice CS Repository. We pick a few to discuss...
To solve

$$\min \mu J(x) + F(x),$$

iterate

$$x^{k+1} \leftarrow \mu J(x) + \langle \nabla F(x^k), x \rangle + \frac{1}{2\delta_k} \| x - x^k \|_2^2.$$

$$\delta_k : \text{step size. Can be obtained from forward-backward operator splitting}$$

**Simple for many choices of** $$J(x): \ell_1, \text{TV}, \| \cdot \|_*, \| \cdot \|_{2,1}, \text{etc.}$$

**Properties:**

- If $$\delta_k < 2/L$$ (Lipschitz const of $$\nabla F$$), $$\| x^k - x^* \|$$ is non-expansive; a fixed-point is a solution

- Adaptive $$\delta_k$$: accelerate convergence, Barzilai-Borwein and nonmonotone line search give sufficient descent
Sparse Vector Recovery

For \( J(x) = \|x\|_1 \) and given \( \nabla F(x^k) \), the subproblem is \( O(n) \) and parallel. It is called **shrinkage** or **soft-thresholding**.

In compressed sensing, \( \nabla F(x^k) = A^\top (Ax^k - b) \) is often cheap (e.g., DFT, discrete cosine transform, chirp or convolution/circulant based sensing). **Codes**: SpaRSA, FPC, SPGL1

\( \nabla F(x^k) \) can be demanding in logistic regression and when dealing with non-Gaussian noise. **Code**: LPS
Low-Rank Matrix Recovery

For low-rank matrix recovery, $J(X) = \|X\|_* := \sum_i \sigma_i(X)$. The subproblem is

$$\min_X \mu \|X\|_* + \frac{1}{2} \|X - Y^k\|_F^2$$

Since both terms are unitary-invariant, it is solved by SVD and then shrinking the singular values.

**Codes**: FPCA, SVT (linearized Bregman). Apply approximate SVD or compute partial SVD.

Lee et al’10 uses the same framework with the max-norm.
Application to Transform-$\ell_1$ and Total Variation

If $J(x) = \|Lx\|_1$ and $L$ is non-singular or even unitary (e.g., orthogonal wavelets), the subproblem has closed-form solutions.

If $J(x) = \|\nabla x\|_1$ (total variation or TV), there are graph-cut (computing max-flow/min-cut) algorithms to solve the subproblems.

If $\|Lx\|_1$ and $L$ is non-invertible, the subproblem is less trivial. Solutions: operator splitting, smoothing (e.g., Huber norm to replace $\ell_1$).
Prox-Linear Theory and Practice

Forward-backward splitting theory applies. (Combettes and Wajs’05)

Convergence for convex $J(x)$ is quite standard; also holds with line search and other variants. (e.g., Wright, Figueiredo, Nowak’08).

Can extend to prox-regular functions (Lewis and Wright’08) and hybrid 1st/2nd-order iterations (Wen, Yin, Zhang, Goldfarb’11)

Convergence speed depends on solution structure, controlled by $\mu$

- **Larger** $\mu$ gives faster and more structured solution. **Reason** for $\ell_1$: smaller optimal support is easier identify and afterward, iterate essentially on a smaller support with a better condition number

- **Smaller** $\mu$ gives slower and less structured solution.

**Continuation**: start from a large $\mu$ and decrease $\mu$ sequentially, using previous solution to warm-start the current iteration (e.g., code FPC).
Techniques Applied with Prox-Linear

- **Two-step descents** (TwIST)
- **Barzilai-Borwein steps, non-monotone line search** (Wen et al.’09)
- **(Block) coordinate descent**: apply descent only to a subset of components. (Tseng and Yun’09, Li and Osher’09, Wright’11).
- **Active set**: estimate the optimal support and solve a reduced subproblem (Shi et al.’08, Wen et al.’09)
- **Use (approximate) second-order information**, e.g. in logistic regression (Byrd et al.’10; Shi et al.’08)
- **Accelerated first-order methods**: given a convex function with $L$-Lipschitz gradients and no other structures, iterations is reduced from $O(L/\epsilon)$ to $O(L/\sqrt{\epsilon})$ for $\epsilon$-optimal in objective value
  - $\min F(x)$: Nesterov’83;
  - $\min J(x) + F(x)$: Nesterov’04, Bech and Teboulle’08, Tseng’08;
  - **ALM**: Ma, Scheinberg, Goldfarb’10
Variable Splitting

Consider a linear operator $L$ and

$$
\min_x J(Lx) + F(x)
$$

Rewrite

$$
\min_{x,y} J(y) + F(x), \quad \text{s.t.} \ y = Lx,
$$

which “separates” $J(\cdot)$ from $F(\cdot)$.

**Augmented Lagrangian**: generates $x^k, y^k$ along with multipliers $\lambda^k$

$$(x^{k+1}, y^{k+1}) \leftarrow \min_{x,y} \mathcal{L}(x,y; \lambda^k) = J(y) + F(x) + \langle \lambda, Lx - y \rangle + \frac{c}{2} \|Lx - y\|_2^2,
\lambda^{k+1} \leftarrow \lambda^k + c(Lx^{k+1} - y^{k+1}).$$

Convergence needs bounded $\lambda^k > 0$ if $J$ and $F$ are convex.

The joint minimization subproblem is still hard to solve.
Variable Splitting

Consider a linear operator $L$ and

$$\min_x J(Lx) + F(x)$$

Introduce

$$\min_{x,y} J(y) + F(x), \quad \text{s.t. } y = Lx,$$

which “separates” $J(\cdot)$ from $F(\cdot)$.

**Alternating Direction Method (ADM):** alternate $x^k$ and $y^k$ updates

$$x^{k+1} \leftarrow \min_x L(x, y^k; \lambda^k),$$

$$y^{k+1} \leftarrow \min_y L(x^{k+1}, y; \lambda^k),$$

$$\lambda^{k+1} \leftarrow \lambda^k + \delta c(Lx^{k+1} - y^{k+1}).$$

Converges if $\delta \in (0, (\sqrt{5} + 1)/2)$. Subproblems $x/y$ are decoupled.
Variable Splitting

If a subproblem, say the $y$-subproblem, is still expensive to solve, one can consider

**Gradient descent:**

$$y^{k+1} \leftarrow y^k - \tau^k \nabla_y L(x^{k+1}, y^k; \lambda^k).$$

**Proximal descent:**

$$y^{k+1} \leftarrow \min_y J(y) + \langle \nabla_y (\langle \lambda, Lx^{k+1} - y \rangle + \frac{c}{2} \|Lx^{k+1} - y\|_2^2), y - y^k \rangle + \frac{1}{2\tau} \|y - y^k\|_2^2.$$
Yin Zhang: “the idea of ADM goes back to Sun-Tze (400 BC) and Caesar (100 BC):

“Divide and Conquer.” — Julius Caesar (100-44 BC)

“远交近攻”，“各个击破”. — Sun-Tzu (400 BC)

Back to 1950s–70s, it appears as operator splitting for PDEs and studied by Douglas, Peaceman, and Rachford, and then Glowinsky et al.’81-89, Gabay ’83.

Subsequent studies are in the context of variational inequality (Eckstein and Bertsekas’92, He et al.’02)

Extensions to multiple blocks (He, Yuan, and collaborators)
For sparse optimization, one subproblem is **shrinkage or its variant**. The other subproblem is **smooth**, very often solving a linear system with $(A^T A + c^k L^T L)$.

For $TV(\cdot) = \|\nabla \cdot\|_1$, breaking $L = \nabla$ from $\|\cdot\|_1$. (Wang, Yin, and Zhang’08). The linear system can be diagonalized by DFT for various $A$ (e.g., $I$ and convolution) and suitable boundary cond.

**Codes**: FTVd, Split Bregman, YALL1, RecPF, SALSA.
Variable Splitting is Versatile

Variable splitting lets one apply ADM to many extensions of $\ell_1$: transform-$\ell_1$, frame-$\ell_1$, constrained/penalized, weighted, nonnegative, isotropic/anisotropic TV, complex, group $\ell_{2,1}$, nuclear norm, and even some non-convex models.

For example, codes YALL1 and YALL1-Group solve

$$
\text{BP:} \quad \min_{x \in \mathbb{C}^n} \|Wx\|_{w,1}, \quad \text{s.t.} \ Ax = b \\
\text{L1/L1:} \quad \min_{x \in \mathbb{C}^n} \|Wx\|_{w,1} + \frac{1}{\nu}\|Ax - b\|_1 \\
\text{L1/L2:} \quad \min_{x \in \mathbb{C}^n} \|Wx\|_{w,1} + \frac{1}{2\rho}\|Ax - b\|_2^2 \\
\text{BP+:} \quad \min_{x \in \mathbb{R}^n} \|x\|_{w,1}, \quad \text{s.t.} \ Ax = b, \ x \geq 0 \\
\text{L1/L1+:} \quad \min_{x \in \mathbb{R}^n} \|x\|_{w,1} + \frac{1}{\nu}\|Ax - b\|_1, \quad \text{s.t.} \ x \geq 0 \\
\text{L1/L2+:} \quad \min_{x \in \mathbb{R}^n} \|x\|_{w,1} + \frac{1}{2\rho}\|Ax - b\|_2^2, \quad \text{s.t.} \ x \geq 0
$$

and their group/joint sparse version.
Related: the Bregman Methods (Osher et al'06, Yin et al'08)

Bregman dist: \( D(u, u^k) := J(u) - (J(u^k) + \langle p^k, u - u^k \rangle) \), \( p^k \in \partial J(u^k) \).

For minimization \( \min_x F(x) \) subject to \( Ax = b \), Bregman Alg updates \( x^k \) and \( p^k \):

\[
\begin{align*}
x^{k+1} &\leftarrow \min \mu \left[ J(x) - (J(x^k) + \langle p^k, x - x^k \rangle) \right] + \frac{1}{2} \|Ax - b\|_2^2 \\
p^{k+1} &\leftarrow p^k + A^\top (b - Ax^{k+1}).
\end{align*}
\]

Relation to augmented Lagrangian: \( p^k = A^\top \lambda^k \). Split Bregman (applying Bregman to the splitting formulation) is ADM.

Bregman is not novel, but leads to new observations:

- Successively linearizing \( J(\cdot) \) gives rise to \( Ax = b \) in the limit
- Better denoising; error forgetting
Example: to recover $x^0$ from noisy measurements $b = Ax^0 + \omega$

1. $\ell_1$ denoising: \[ \min \mu \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2 \]

\[ \mu = 48.5 \]

2. $\ell_1$ Bregman iteration

\[ \mu = 49 \]

$\mu = 150$, 5 iterations

Better denoising results are also observed on image reconstruction.
The Add-Back Update, Error Forgetting

After a change of variable, the Bregman update has an add-back form:

\[ x^{k+1} \leftarrow \min \mu J(x) + \frac{1}{2} \|Ax - b^k\|_2^2, \]

\[ b^{k+1} \leftarrow b + (b^k - Ax^{k+1}). \]

- Subproblem is in the same form of \( \ell_1 + \ell_2^2 \)
- Subproblem solution errors are iterative cancelled

Simulation: subproblems solved at low accuracy \(10^{-6}\) by ISTA, FPC-BB, GPSR, and SpaRSA, but relative errors drop to machine precision
Non-Convex $\ell_p$ Minimization, $p < 1$

Pioneered by Rao et al’97; popularized by Chartrand et al for CS

However, there are **multiple local minima** for $\min_x \|x\|_p$, s.t $Ax = b$.

$\|x\|_p$ colorized over $\{x^0 + v : v \in \mathcal{N}(A)\}$
Non-Convex Minimization

Chartrand and Yin’08, Daubechies et al.’09, Lai-Wang’10 smooth out the “small” local minima by replacing $\|x\|_p$ by

$$\sum_{i=1}^{n} \frac{x_i^2}{(x_i^2 + \epsilon)^{(1-p)/2}}.$$

Work better than $\ell_1$ on fast decaying signals (nonzero entries have a fast decaying distribution)
For fast decaying sparse signals, smoothing + continuation help avoid local minima.

Red: sparse solution; Green: smoothed $\ell_p$ solutions

$\epsilon = 1.0$

Courtesy of Rick Chartrand
For fast decaying sparse signals, **smoothing** + **continuation** help avoid local minima.

Red: sparse solution; Green: smoothed $\ell_p$ solutions

$\epsilon = 0.1$

Courtesy of Rick Chartrand
For fast decaying sparse signals, smoothing + continuation help avoid local minima.

Red: sparse solution; Green: smoothed $\ell_p$ solutions

$\epsilon = 0.01$

Courtesy of Rick Chartrand
For fast decaying sparse signals, **smoothing + continuation** help avoid local minima.

Red: sparse solution; Green: smoothed $\ell_p$ solutions

$\epsilon = 0.001$

Courtesy of Rick Chartrand
Other non-convex techniques

- Candes, Wakin, and Boyd’08 uses $\sum_{i=1}^{n} |x_i|/(|x_i| + \epsilon)^{(1-p)}$.
- Wang and Yin’10 uses $\sum_i w_i |x_i|$ with binary $w_i$.
  **Code**: Threshold–ISD
- Mohimani, Babaie-Zadeh, Jutten use other approximations.
  **Code**: SL0
- Extensions to joint sparse vector and low-rank matrix recovery ......

Nearly all need fast decaying for significantly better recovery than $\ell_1$. 
Matrix Completion

(Candes and Recht’09; Recht, Fazel, and Parrilo’10) A matrix $M$ can be exactly recovered from enough yet few random subsamples with overwhelming probability, provided

- Low rank
- Row and column subspaces are incoherent

Motivating example: low rank due to only a few factors contributing to users’ preferences
Matrix Completion Model

Given samples in $\Omega$, recover $X$ by

$$\min_x \mu J(X) + \frac{1}{2} \|X_\Omega - M_\Omega\|_F^2$$

Choices of $J(X)$:

- $\text{rank}(X)$: computationally intractable
- $\|X\|_*$: convex envelope of $\text{rank}(X)$
- $\|X\|_{\text{max}}$: another convex regularizer of $\text{rank}(X)$
- Nonconvex $\ell_p$ semi-norm of singular values of $X$

**Codes:** SVT, FPCA, APGL, ...

Also, $\text{trace}(X^\top X)^{p/2}$: equals $\|X\|_*$ when $p = 1$; it is non-convex when $p < 1$ and is the $\ell_p$-norm of singular values.
On terabyte data, SVDs become almost forbidden.

**Code** LMaFit avoids SVDs by exploiting low-rank factorization $M \approx XY$

\[
\min_{X, Y} \| \mathcal{P}_\Omega (XY - M) \|_F
\]

Given an approximate rank $r$, $X$ is $m$-by-$r$ and $Y$ is $r$-by-$n$.

**Drawbacks:**
- non-convex, no theoretical guarantees known yet
- need rank estimate or dynamic rank update

**Advantages:**
- cheap (alternating minimization)
- does not store any full matrix

Other models and codes:
- OptSpace (descent on the Grassmanian manifold)
- Low-rank+Sparse, splitting method: Yuan, Yang’09, Lin et al.’10, etc.
To finish up

Some observations:
- The work of finding structured solutions is interdisciplinary
- Recognizing the structure are important
- It is a common ground for existing and novel algorithms
- This field grows quickly, and its tools becoming standard techniques in computational mathematics and engineering

Benefits of structured solutions:
- cheap to store / transmit
- more meaningful, easy to understand,
- easy to use
- robust to errors
- easy to find

Acknowledgements: R. Chartrand for some nice figures, and Yangyang Xu for some slides.