Minimizing the Difference of L_1 and L_2 Norms with Applications

Yifei Lou

Department of Mathematical Sciences
University of Texas Dallas

May 31, 2017

Partially supported by NSF DMS 1522786

Outline

1. A nonconvex approach: L_1-L_2
2. Minimization algorithms
3. Some applications
4. Conclusions

Background

We aim to find a sparse vector from an under-determined linear system,

$$\hat{x}_0 = \text{argmin}_x \|x\|_0 \quad \text{s.t.} \quad Ax = b.$$

This is NP-hard.

Background

We aim to find a sparse vector from an under-determined linear system,

$$\hat{x}_0 = \text{argmin}_x \|x\|_0 \quad \text{s.t.} \quad Ax = b.$$

This is NP-hard.

A popular approach is to replace L_0 by L_1, i.e.,

$$\hat{x}_1 = \text{argmin}_x \|x\|_1 \quad \text{s.t.} \quad Ax = b.$$

Background

We aim to find a sparse vector from an under-determined linear system,

$$\hat{x}_0 = \text{argmin}_x \|x\|_0 \quad \text{s.t.} \quad Ax = b.$$

This is NP-hard.

A popular approach is to replace L_0 by L_1, i.e.,

$$\hat{x}_1 = \text{argmin}_x \|x\|_1 \quad \text{s.t.} \quad Ax = b.$$

The equivalence between L_0 and L_1 norms holds when the matrix A satisfies the restricted isometry property (RIP).

Candes-Romberg-Tao (2006)

Coherence

Another sparse recovery guarantee is based on coherence.

$$\|x\|_0 \leqslant \frac{1}{2}(1 + \mu(A)^{-1}),$$

where coherence of a matrix $A = [a_1, \cdots, a_N]$ is defined as

$$\mu(A) = \max_{i \neq j} \frac{|a_i^T a_j|}{\|a_i\| \|a_j\|}.$$

Coherence

Another sparse recovery guarantee is based on coherence.

$$\|x\|_0 \leqslant \frac{1}{2}(1+\mu(A)^{-1}),$$

where coherence of a matrix $A = [a_1, \cdots, a_N]$ is defined as

$$\mu(A) = \max_{i \neq j} \frac{|a_i^T a_j|}{\|a_i\| \|a_j\|}.$$

Two extreme cases are

- $\mu \sim 0 \Rightarrow$ incoherent matrix
- $\mu \sim 1 \Rightarrow$ coherent matrix

Coherence

Another sparse recovery guarantee is based on coherence.

$$\|x\|_0 \leqslant \frac{1}{2}(1 + \mu(A)^{-1}),$$

where coherence of a matrix $A = [a_1, \cdots, a_N]$ is defined as

$$\mu(A) = \max_{i \neq j} \frac{|a_i^T a_j|}{\|a_i\| \|a_j\|} \ .$$

Two extreme cases are

- $\mu \sim 0 \Rightarrow$ incoherent matrix
- $\mu \sim 1 \Rightarrow$ coherent matrix

<u>What if the matrix is coherent?</u>

L_1-L_2 works well for coherent matrix

We consider an over-sampled DCT matrix with each column as

$$\mathbf{a}_j = \frac{1}{\sqrt{N}} \cos\left(\frac{2\pi j \mathbf{w}}{F}\right), j = 1, \cdots, N$$

where $\mathbf{w}$ is a random vector of length M.

The larger F is, the more coherent the matrix. Take a 100×1000 matrix for an example:

F	coherence
1	0.3981
10	0.9981
20	0.9999

P. Yin, Y. Lou, Q. He and J. Xin, SIAM Sci. Comput., 2015

Y. Lou, P. Yin, Q. He and J. Xin, J. Sci. Comput., 2015

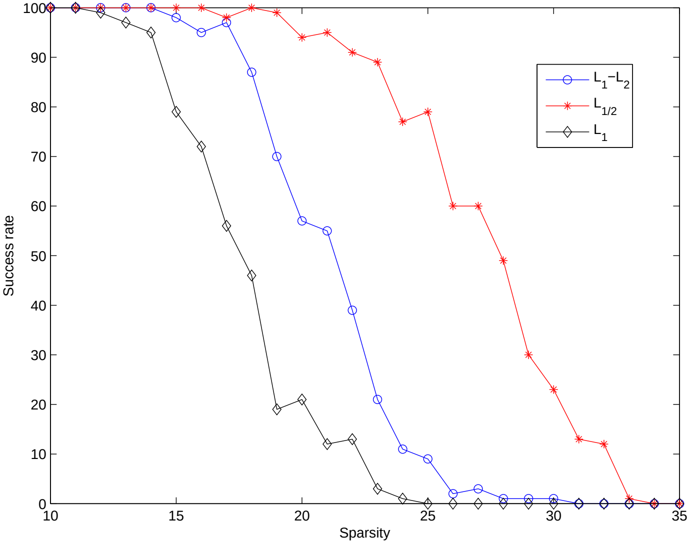


Figure: Success rates of incoherent matrices, $F = 1$.

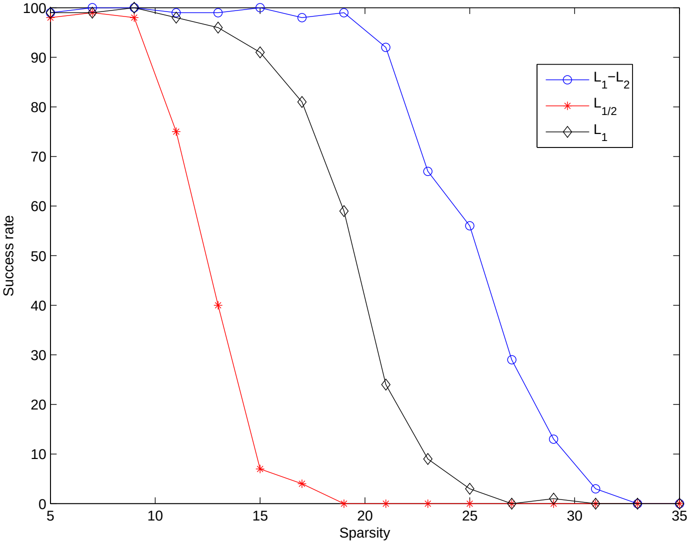


Figure: Success rates of coherent matrices, $F = 20$.

Comparing metrics

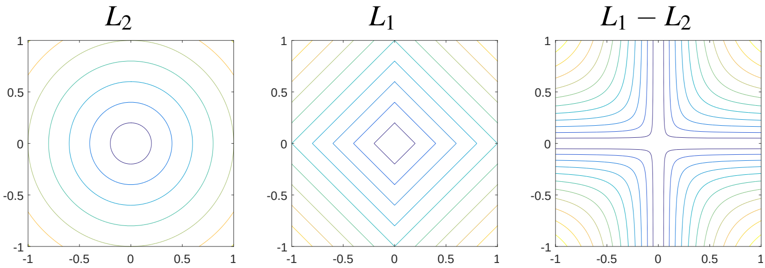


Figure: Level lines of three metrics: L_2 (strictly convex), L_1 (convex), and $L_1 - L_2$ (nonconvex).

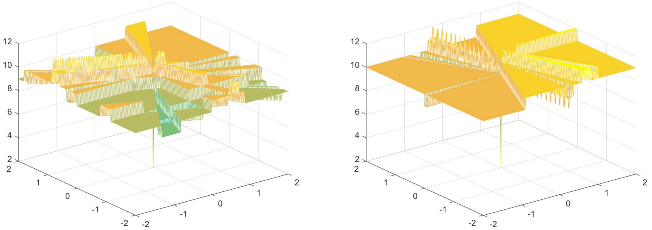
Comparing nonconvex metrics

Consider a matrix A of size 17×19 and $V \in \mathrm{R}^{19\times 2}$ be the basis of the null space of A, i.e. $AV = 0$.

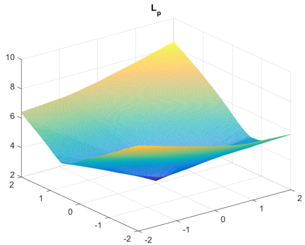
So the feasible set is a two-dimensional affine space, i.e.
$\{x : Ax = Ax_g\} = \{x = x_g + V \begin{bmatrix} s \\ t \end{bmatrix} : \ s, t \in \mathrm{R}\}$.

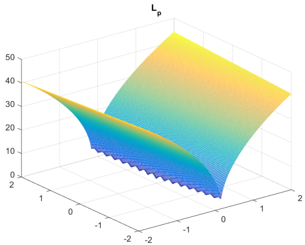
Visualize objective functions L_0, $L_{1/2}$, and L_1-L_2 over 2D st-plane.

L_0: incoherent (left) and coherent (right)

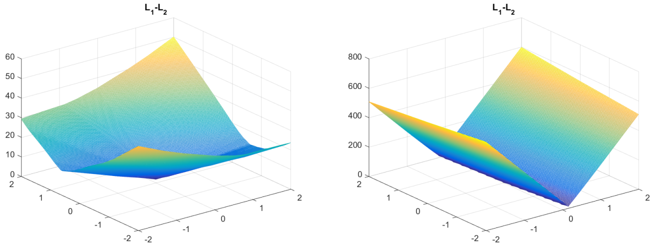


$L_{1/2}$: incoherent (left) and coherent (right)

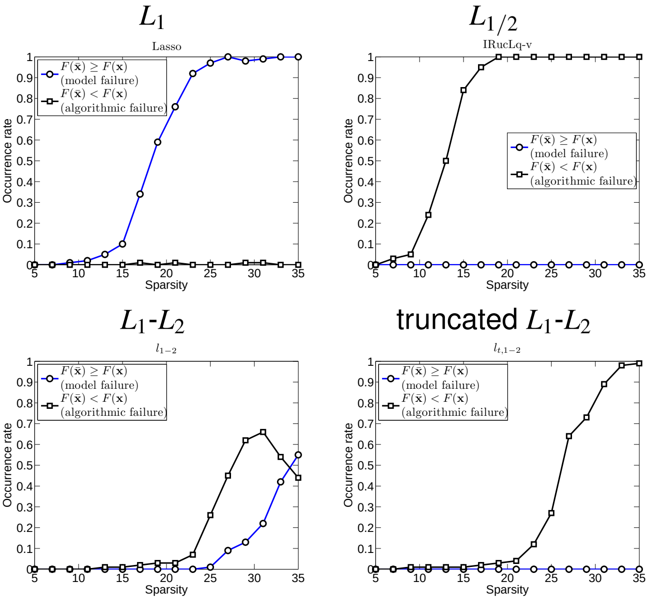




L_1-L_2: incoherent (left) and coherent (right)



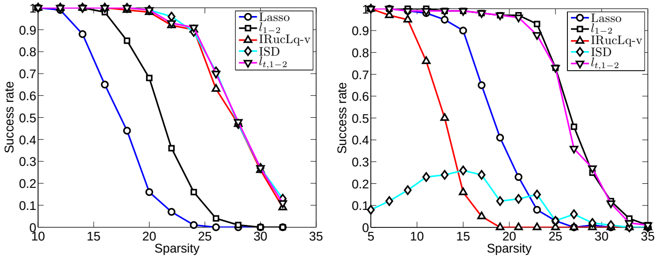
Model failure v.s. algorithm failure



Truncated L_1-L_2

$$\|\mathbf{x}\|_{t,1-2} := \sum_{i \notin \Gamma_{\mathbf{x},t}} |x_i| - \sqrt{\sum_{i \notin \Gamma_{\mathbf{x},t}} x_i^2},$$

where $\Gamma_{\mathbf{x},t} \subseteq \{1, \ldots, N\}$ with cardinality t is a set containing the indices of the entries of $\mathbf{x}$ with the t largest magnitudes.



T. Ma, Y. Lou, and T. Huang, SIAM Imaging Sci., to appear 2017

Advantages of L_1-L_2

- Lipschitz continuous

Advantages of L_1-L_2

- Lipschitz continuous

- Correct L_1's biasedness by subtracting something with smooth gradient a.e.

Advantages of L_1-L_2

- Lipschitz continuous

- Correct L_1's biasedness by subtracting something with smooth gradient a.e.

- Exact recovery of 1-sparse vectors (truncated version yields exact recovery of t-sparse vectors)

Advantages of L_1-L_2

- Lipschitz continuous
- Correct L_1's biasedness by subtracting something with smooth gradient a.e.
- Exact recovery of 1-sparse vectors (truncated version yields exact recovery of t-sparse vectors)
- Good for coherent compressive sensing

Outline

1. A nonconvex approach: L_1-L_2
2. Minimization algorithms
3. Some applications
4. Conclusions

Algorithms

We consider an unconstrained $L_1 - L_2$ formulation, i.e.,

$$\min_{x \in \mathbb{R}^N} F(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda(\|x\|_1 - \|x\|_2).$$

Our first attempt is using the difference of convex algorithm (DCA) by composing decompose $F(x) = G(x) - H(x)$ into

$$\begin{cases} G(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \\ H(x) = \lambda\|x\|_2. \end{cases}$$

An iterative scheme is,

$$x^{n+1} = \arg\min_{x \in \mathbb{R}^N} \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 - \langle x, \frac{\lambda x^n}{\|x^n\|_2}\rangle.$$

We then derive a proximal operator for L_1-αL_2 ($\alpha \geq 0$)

$$x^* = \arg\min_x \lambda \left(\|x\|_1 - \alpha\|x\|_2\right) + \frac{1}{2}\|x - y\|_2^2,$$

which has a closed-form solution:

1. If $\|y\|_\infty > \lambda$, then $x^* = z(\|z\|_2 + \alpha\lambda)/\|z\|_2$, where $z = \text{shrink}(y, \lambda)$;
2. if $\|y\|_\infty = \lambda$, then $\|x^*\|_2 = \alpha\lambda$, $x_i^* = 0$ for $|y_i| < \lambda$;
3. If $(1-\alpha)\lambda < \|y\|_\infty < \lambda$, then x^* is 1-sparse vector satisfying $x_i^* = 0$ for $|y_i| < \|y\|_\infty$;
4. If $\|y\|_\infty \leq (1-\alpha)\lambda$, then $x^* = 0$.

Y. Lou and M. Yan, J. Sci Comput., to appear 2017

Remarks

- Most L_1 solves are applicable for L_1-αL_2 by replacing soft shrinkage with this proximal operator.

Remarks

- Most L_1 solves are applicable for L_1-αL_2 by replacing soft shrinkage with this proximal operator.
- The algorithm of combining ADMM and this operator (nonconvex-ADMM), is much faster than the DCA.

Remarks

- Most L_1 solves are applicable for L_1-αL_2 by replacing soft shrinkage with this proximal operator.
- The algorithm of combining ADMM and this operator (nonconvex-ADMM), is much faster than the DCA.
- Both nonconvex-ADMM and DCA converge to stationary points.

Remarks

- Most L_1 solves are applicable for L_1-αL_2 by replacing soft shrinkage with this proximal operator.
- The algorithm of combining ADMM and this operator (nonconvex-ADMM), is much faster than the DCA.
- Both nonconvex-ADMM and DCA converge to stationary points.
- However, nonconvex-ADMM does not give better performance than DCA for coherent CS.

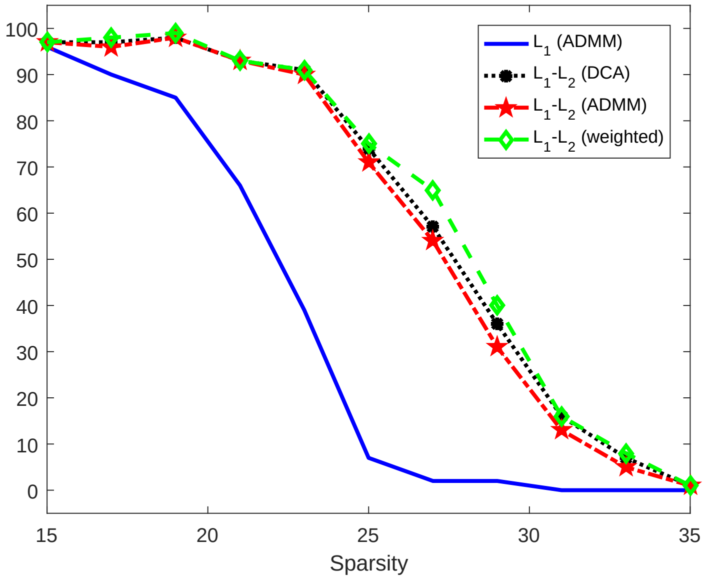


Figure: Success rates of coherent matrices, F=20.

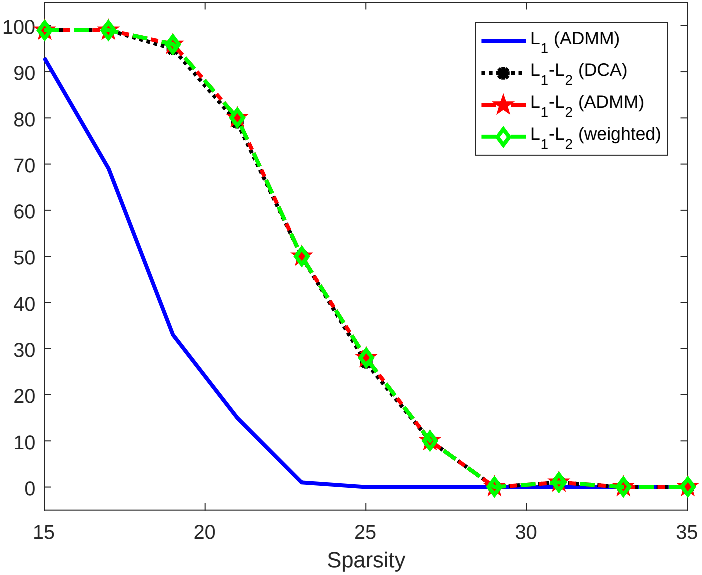


Figure: Success rates of incoherent matrices, F=5.

- DCA is more stable than nonconvex-ADMM, as each DCA subproblem is convex.

- DCA is more stable than nonconvex-ADMM, as each DCA subproblem is convex.

- Since it is convex at $\alpha = 0$, we consider a continuation scheme of gradually increasing α from 0 to 1, referred to as “weighted”.

- DCA is more stable than nonconvex-ADMM, as each DCA subproblem is convex.

- Since it is convex at $\alpha = 0$, we consider a continuation scheme of gradually increasing α from 0 to 1, referred to as “weighted”.

- How to update the weight α?

- DCA is more stable than nonconvex-ADMM, as each DCA subproblem is convex.
- Since it is convex at $\alpha = 0$, we consider a continuation scheme of gradually increasing α from 0 to 1, referred to as “weighted”.
- How to update the weight α?
    - For incoherence matrices, a linear increase for α with a large slope until reaching one;

- DCA is more stable than nonconvex-ADMM, as each DCA subproblem is convex.

- Since it is convex at $\alpha = 0$, we consider a continuation scheme of gradually increasing α from 0 to 1, referred to as "weighted".

- How to update the weight α?
  - For incoherence matrices, a linear increase for α with a large slope until reaching one;
  - For coherent cases, a sigmoid function to update α, which may or may not reach one.

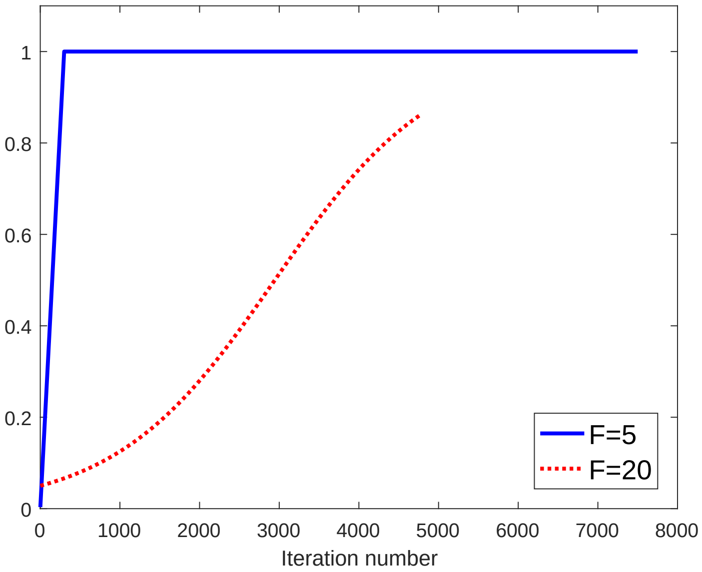


Figure: Different ways of updating α for incoherent (blue) or coherent (red) matrices.

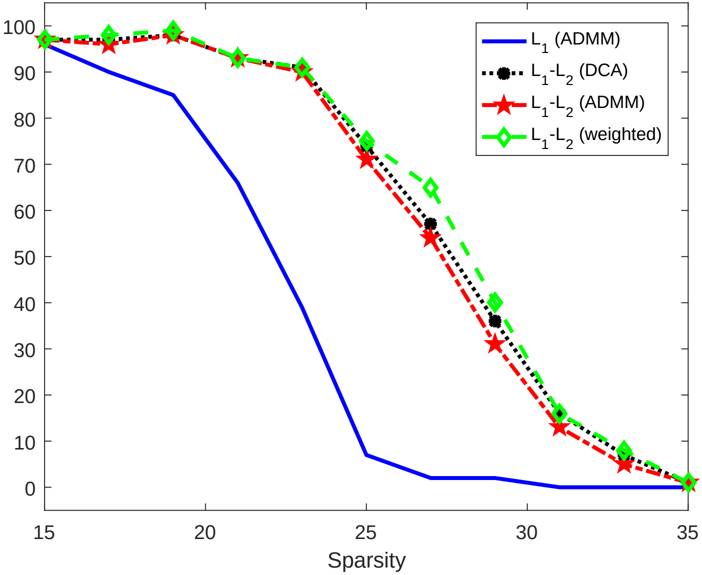


Figure: Success rates of coherent matrices, F=20.

Outline

1. A nonconvex approach: L_1-L_2
2. Minimization algorithms
3. Some applications
4. Conclusions

Super-resolution

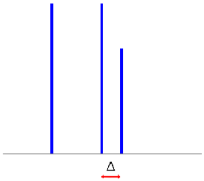
The super-resolution problem discussed here is different to image zooming or magnification, but aiming to recover a real-valued signal from its low-frequency measurements.

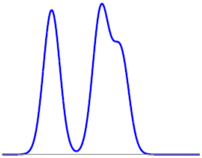
A mathematical model is expressed as

$$b_k = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} x_t e^{-i2\pi kt/N}, \qquad |k| \leq f_c,$$

where $x \in \mathbb{R}^N$ is a vector of interest, and $b \in \mathbb{C}^n$ is the given low frequency information with $n = 2f_c + 1$ $(n < N)$.

Point source with minimum separation



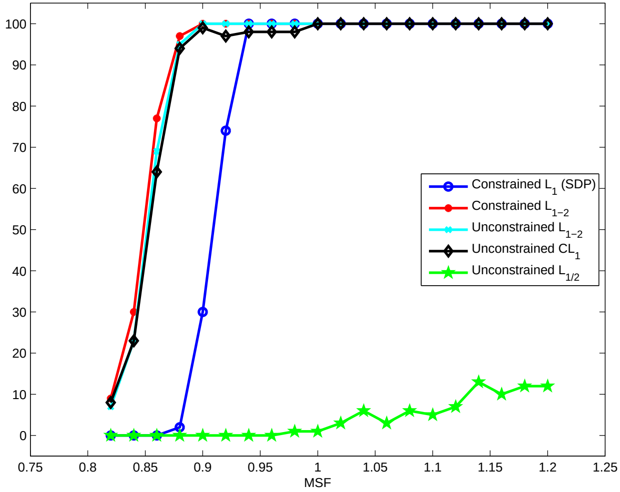


Theorem by Candés and Fernandez-Granda 2012

Let $T = \{t_j\}$ be the support of x. If the minimum distance obeys

$$\triangle(T) \geq 2 \cdot N/f_c,$$

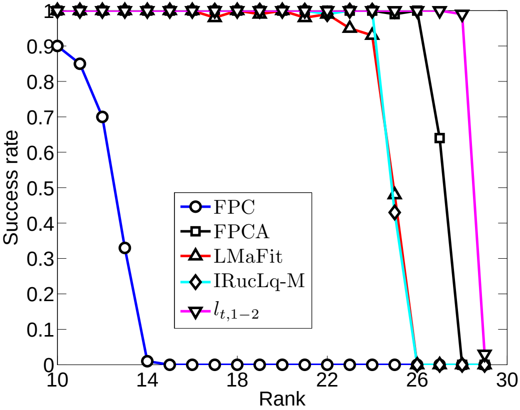
then x is the unique solution to L_1 minimization. If x is real-valued, then the minimum gap can be lowered to $1.26 \cdot N/f_c$.



Y. Lou, P. Yin and J. Xin, J. Sci. Comput., 2016

Low-rank recovery

Replacing nuclear norm with truncated L_1-L_2 of the singular values



T. Ma, Y. Lou, and T. Huang, SIAM Imaging Sci., to appear 2017

Image processing

We consider

$$J(u) = \|D_x u\|_1 + \|D_y u\|_1 - \alpha \|\sqrt{|D_x u|^2 + |D_y u|^2}\|_1 \ ,$$

which turns out to be a weighted difference of anisotropic and isotropic TV:

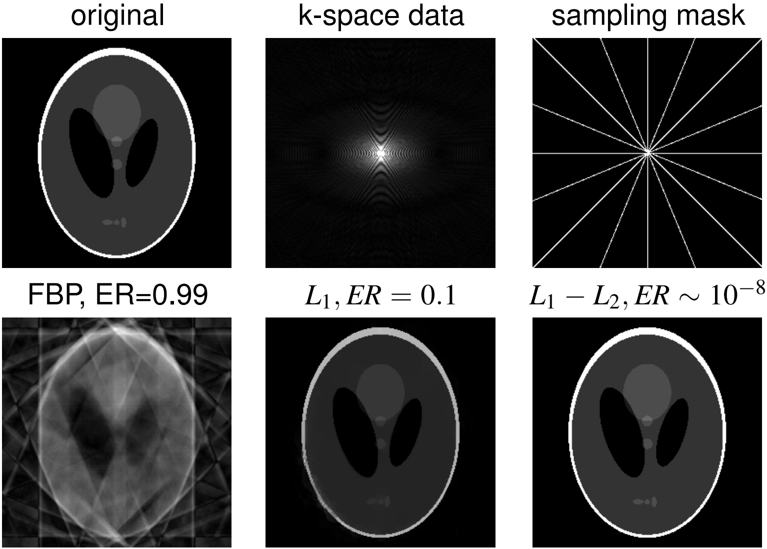
$$J(u) = J_{ani} - \alpha J_{iso},$$

where $\alpha \in [0, 1]$ is a weighting parameter.

Gradient vectors (u_x, u_y) are mostly 1-sparse, and α takes into account the occurrence of non-sparse gradient vectors.

Y. Lou, T. Zeng, S. Osher and J. Xin, SIAM J. Imaging Sci., 2015

MRI reconstruction



Block-matching

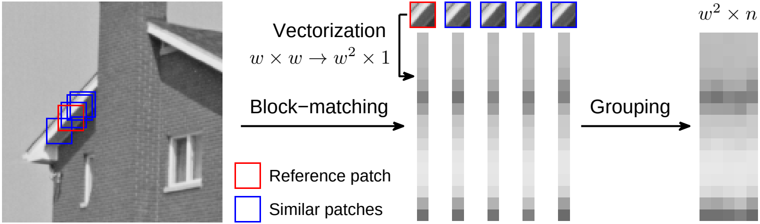
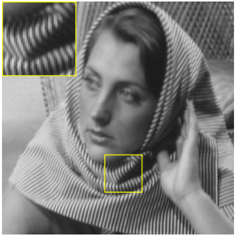


Figure: Illustration of constructing groups by block-matching (BM). For each $w \times w$ reference patch from an $n_1 \times n_2$ image, we use block-matching to search its $n-1$ best matched patches in terms of Euclidean distance, and then vectorize and combine those patches to form a group of size $w^2 \times n$.

Block-matching inpainting

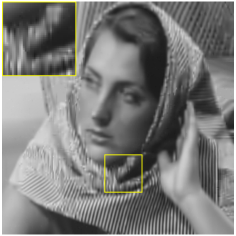
Barbara



85% missing

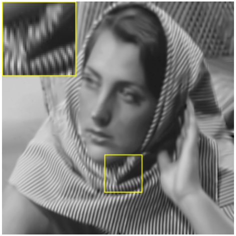
BM3D

PSNR=26.71,SSIM=0.8419



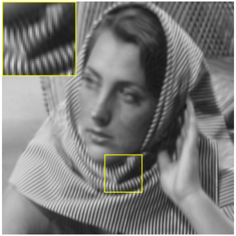
SAIST

PSNR=29.53,SSIM=0.9147



ours

PSNR=30.65,SSIM=0.9264



T. Ma, Y. Lou, T. Huang and X. Zhao, ICIP, to appear 2017

Logistic regression

Given a collection of training data $\mathbf{x}_i \in \mathbb{R}^n$ with a label $y_i \in \{\pm 1\}$ for $i = 1, 2, \cdots, m$, we aim to find a hyperplane defined by $\{\mathbf{x} : \mathbf{w}^T \mathbf{x} + v = 0\}$ by minimizing the following objective function,

$$\min_{\mathbf{w} \in \mathbb{R}^n, v \in \mathbb{R}} F(\mathbf{w}, v) + l_{avg}(\mathbf{w}, v)$$

where the second term is called averaged loss defined as

$$l_{avg}(\mathbf{w}, v) = \frac{1}{m} \sum_{i=1}^{m} \ln\left(1 + \exp(-y_i(\mathbf{w}^T \mathbf{x}_i + v))\right).$$

Preliminary results

Real data of patients with inflammatory bowel disease (IBD) for years 2011 and 2012. For each year, the data set contains 18 types of medical information, such as prescriptions, number of office visits, and whether the patient was hospitalized. We used the 2011 data to train our classifier and the 2012 data to validate its performance.

	L_1	L_2	L_1-L_2
Recall	0.6494	0.6585	**0.6829**
Precision	0.0883	0.0882	**0.0912**
F-Score	0.0573	0.0581	**0.0622**
AUC	0.7342	0.7321	**0.7491**

UCLA REU project in 2015, followed by Q. Jin and Y. Lou for a journal submission

Conclusions

1. L_1-L_2 is always better than L_1, and is better than L_p for highly coherent matrices.

Conclusions

1. L_1-L_2 is always better than L_1, and is better than L_p for highly coherent matrices.

2. Proximal operator can accelerate the minimization, but it tends to obtain a suboptimal solution.

Conclusions

1. L_1-L_2 is always better than L_1, and is better than L_p for highly coherent matrices.
2. Proximal operator can accelerate the minimization, but it tends to obtain a suboptimal solution.
3. In general, nonconvex methods have better empirical performance compared to convex ones, but lack of provable grounds.

Thank you!