Sparse and Low-Rank Representations for Computer Vision

Presenter:

David Wipf

Microsoft Research

Slides courtesy of:

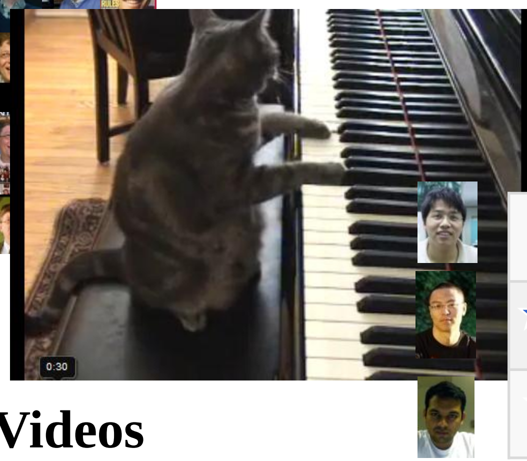
Yi Ma

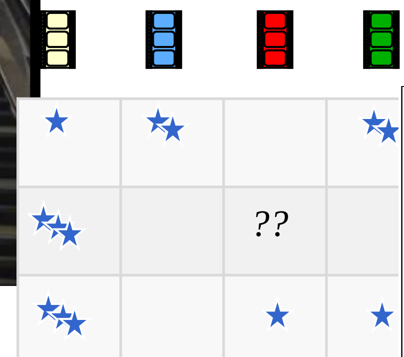
Shanghai Tech

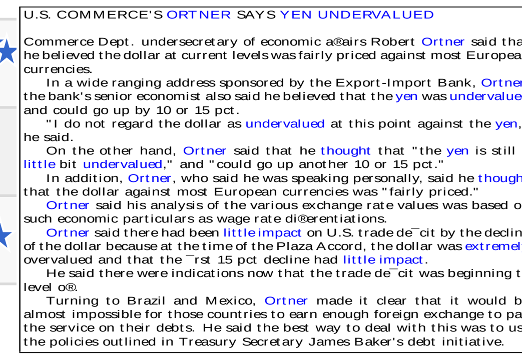
John Wright

Columbia University

CONTEXT: Data increasingly massive, high-dimensional…







U.S. COMMERCE'S ORTNER SAYS YEN UNDERVALUED

Commerce Dept. undersecretary of economic a®airs Robert Ortner said that he believed the dollar at current levels was fairly priced against most European currencies.

In a wide ranging address sponsored by the Export-Import Bank, Ortner, the bank's senior economist also said he believed that the yen was undervalued and could go up by 10 or 15 pct.

"I do not regard the dollar as undervalued at this point against the yen," he said.

On the other hand, Ortner said that he thought that "the yen is still a little bit undervalued," and "could go up another 10 or 15 pct."

In addition, Ortner, who said he was speaking personally, said he thought that the dollar against most European currencies was "fairly priced."

Ortner said his analysis of the various exchange rate values was based on such economic particulars as wage rate di®erentiations.

Ortner said there had been little impact on U.S. trade de¯cit by the decline of the dollar because at the time of the Plaza Accord, the dollar was extremely overvalued and that the ¯rst 15 pct decline had little impact.

He said there were indications now that the trade de¯cit was beginning to level o®.

Turning to Brazil and Mexico, Ortner made it clear that it would be almost impossible for those countries to earn enough foreign exchange to pay the service on their debts. He said the best way to deal with this was to use the policies outlined in Treasury Secretary James Baker's debt initiative.

Images

⇓ ***1M pixels***

Compression

De-noising

Super-resolution

Recognition…

Videos

⇓ ***1B voxels***

Streaming

Tracking

Stabilization…

User data

⇓ ***1B users***

Clustering

Classification

Collaborative filtering…

Web data

⇓

Indexing

Ranking

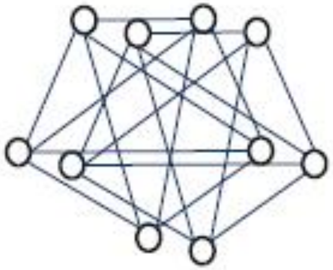
Search…

How to extract low-dim structures from such high-dim data?

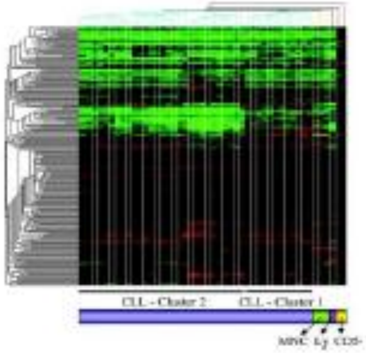
CONTEXT: Data increasingly massive, high-dimensional…

Recognition

Surveillance



Search and Ranking

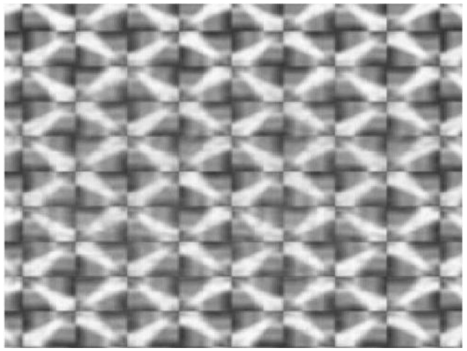


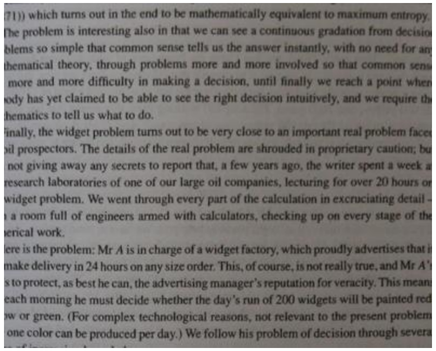
Bioinformatics

The curse of dimensionality:
…increasingly demand inference with limited samples for very high-dimensional data.

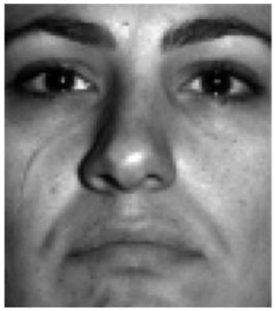
The blessing of dimensionality:
… real data highly concentrate on low-dimensional, sparse, or degenerate structures in the high-dimensional space.

CONTEXT: Low dimensional structures in visual data



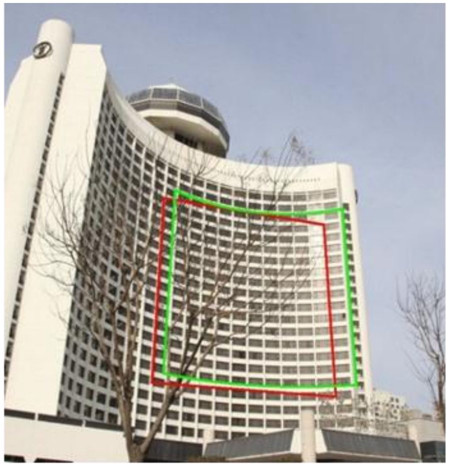


(71)) which turns out in the end to be mathematically equivalent to maximum entropy.
The problem is interesting also in that we can see a continuous gradation from decisio
blems so simple that common sense tells us the answer instantly, with no need for an
thematical theory, through problems more and more involved so that common sens
more and more difficulty in making a decision, until finally we reach a point when
ody has yet claimed to be able to see the right decision intuitively, and we require th
hematics to tell us what to do.
Finally, the widget problem turns out to be very close to an important real problem face
oil prospectors. The details of the real problem are shrouded in proprietary caution; bu
not giving away any secrets to report that, a few years ago, the writer spent a week a
research laboratories of one of our large oil companies, lecturing for over 20 hours o
widget problem. We went through every part of the calculation in excruciating detail –
a room full of engineers armed with calculators, checking up on every stage of the
erical work.
Here is the problem: Mr *A* is in charge of a widget factory, which proudly advertises that i
make delivery in 24 hours on any size order. This, of course, is not really true, and Mr *A*'
s to protect, as best he can, the advertising manager's reputation for veracity. This mean
each morning he must decide whether the day's run of 200 widgets will be painted red
ow or green. (For complex technological reasons, not relevant to the present problem
one color can be produced per day.) We follow his problem of decision through severa



Visual data exhibit ***low-dimensional structures*** due to rich ***local*** regularities, ***global*** symmetries, ***repetitive*** patterns, or ***redundant*** sampling.

CONTEXT: But life is not so easy…



Real application data often contain ***missing observations****,* ***corruptions****, or subject to unknown* ***deformation or misalignment.***

Classical methods (e.g., PCA, least square regression) break down…

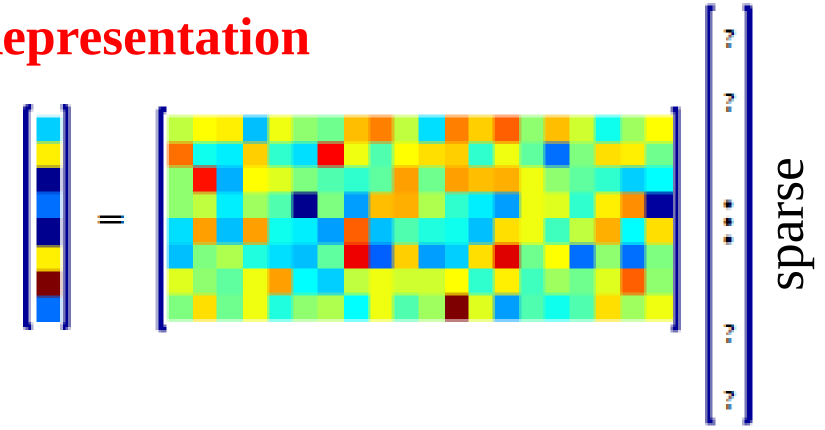
In their place: Sparse representations, robust PCA, and many others

Two Low-Dimensional Representations

Sparse Representation

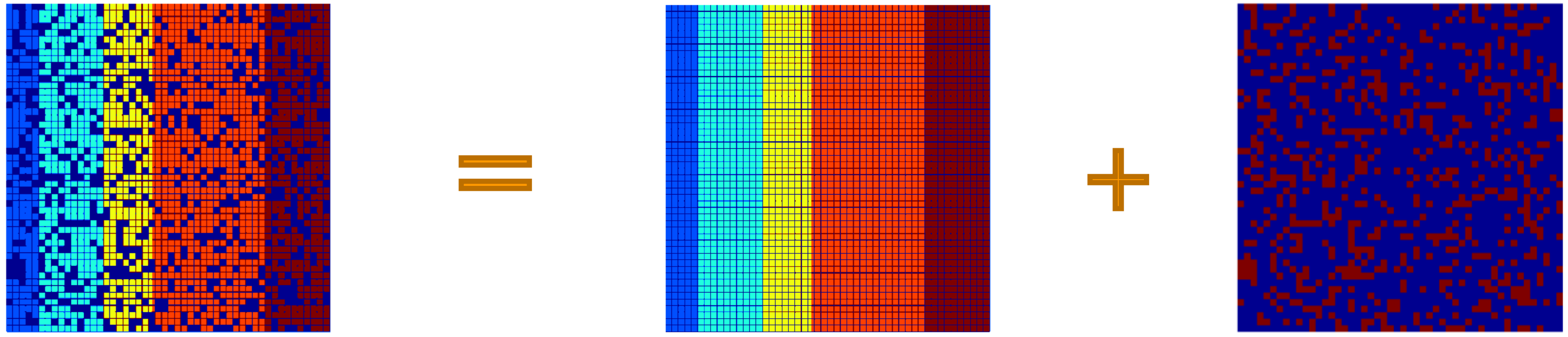
Underdetermined system

$$y = Ax$$



Robust PCA

Corrupted Observations | Low-rank Structures | Sparse Structures



Vast number of candidate applications

Overview

- Part I: Motivation, Theory, Applications
- Part II: Efficient Convex Algorithms
- Part III: Non-Convex Alternatives

Part I: Motivation, Theory, Applications

Sparse Representations

- Linear generative model:

$$\mathbf{y} = A\mathbf{x} + \boldsymbol{\varepsilon}$$

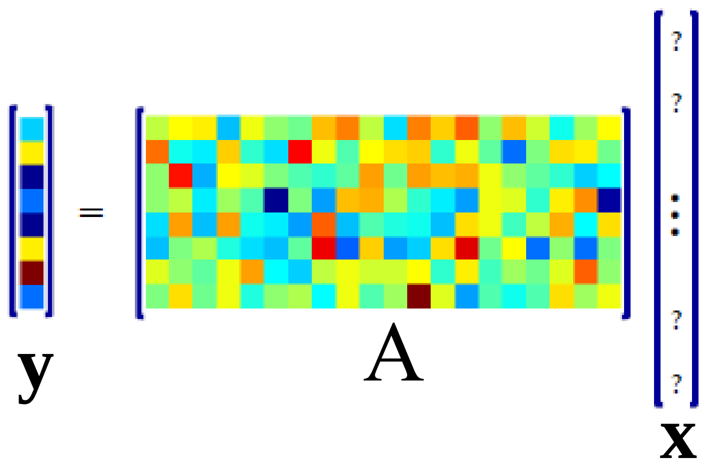
m-dimensional observations → $\mathbf{y}$

matrix of *n* basis vectors or features → A

unknown sparse coefficients → $\mathbf{x}$

noise → $\boldsymbol{\varepsilon}$

- **Objective**: Estimate the *sparse* $\mathbf{x}$ assuming $n >> m$



underdetermined system

Example

$$\mathbf{y} = \begin{bmatrix} -4 \\ -5 \\ 3 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 4 & 1 & 1 & 6 \\ -2 & 1 & -4 & 2 & -3 \\ 3 & 3 & 2 & -2 & 1 \end{bmatrix}$$

Want to find an $\mathbf{x}$ that solves

$$\mathbf{y} = A\mathbf{x}$$

non-sparse

$$\mathbf{x} = \begin{bmatrix} 4 \\ -1 \\ 3 \\ 5 \\ -2 \end{bmatrix}$$

sparse

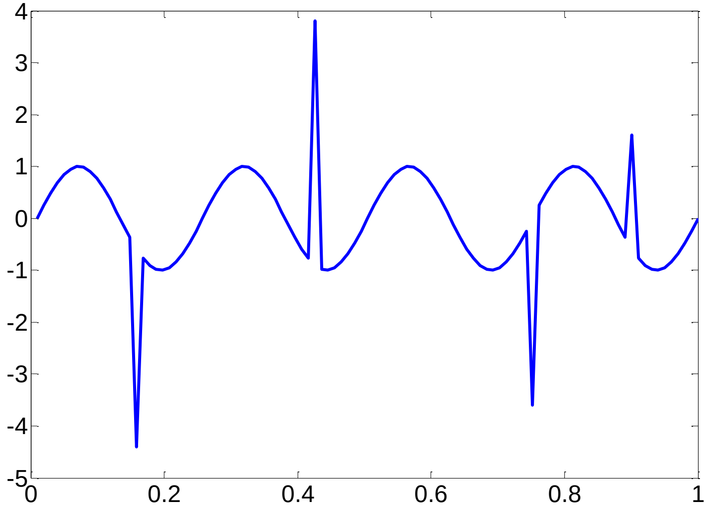
$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ -1 \end{bmatrix}$$

Sparse representations reflect low-dimensional structure

Sinusoid and Spikes Example

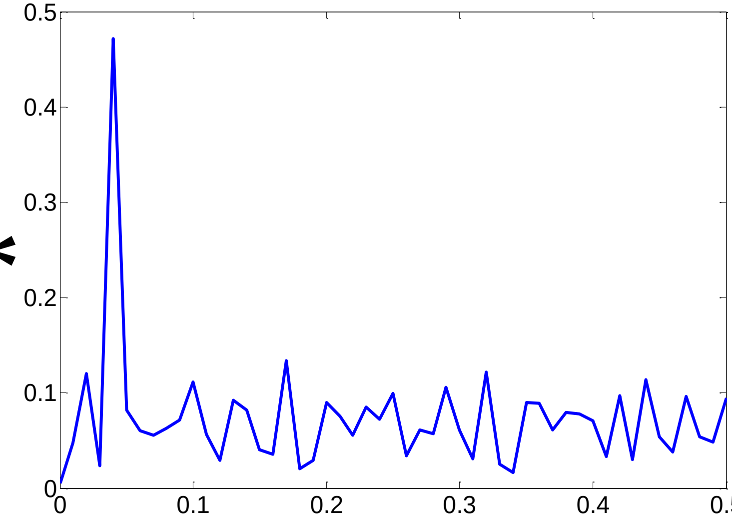
$$A = [\text{ DFT basis }]$$

Observed Signal (**y**)



$$= A *$$

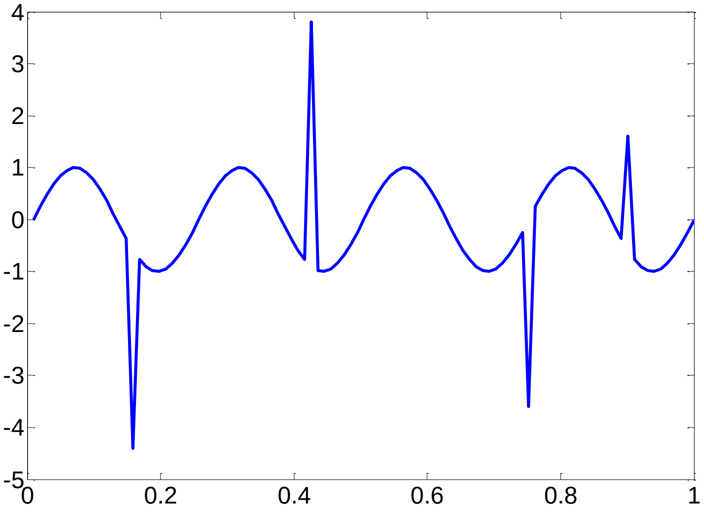
Spectrum (**x**)



Sinusoid and Spikes Example

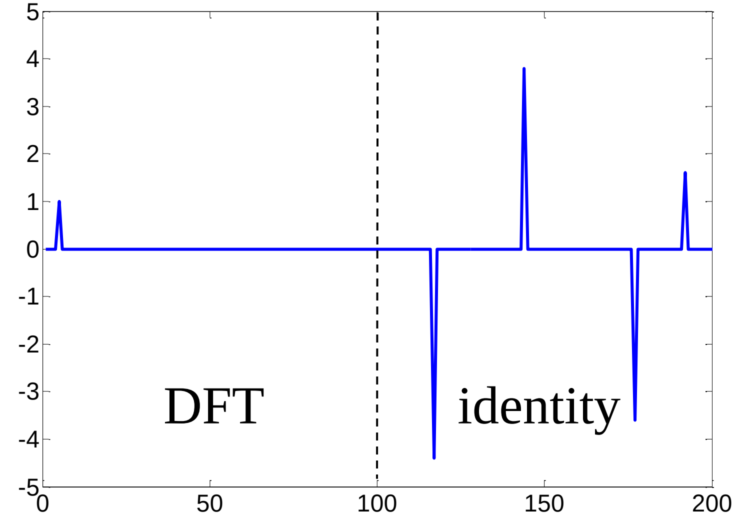
$$A = [\ \text{DFT basis} + \text{identity}]$$

Observed Signal (**y**)

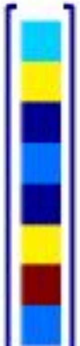
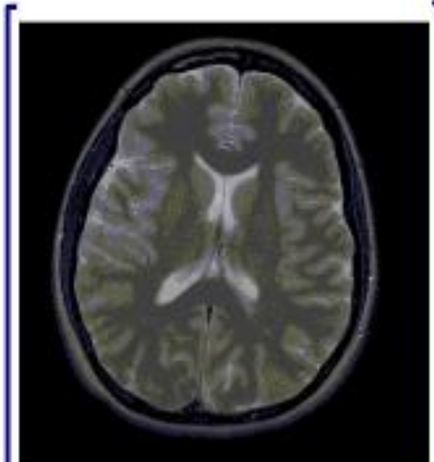


$$= A *$$

Sparse Decomposition (**x**)



Signal Acquisition

$\boldsymbol{y}$  $=$  

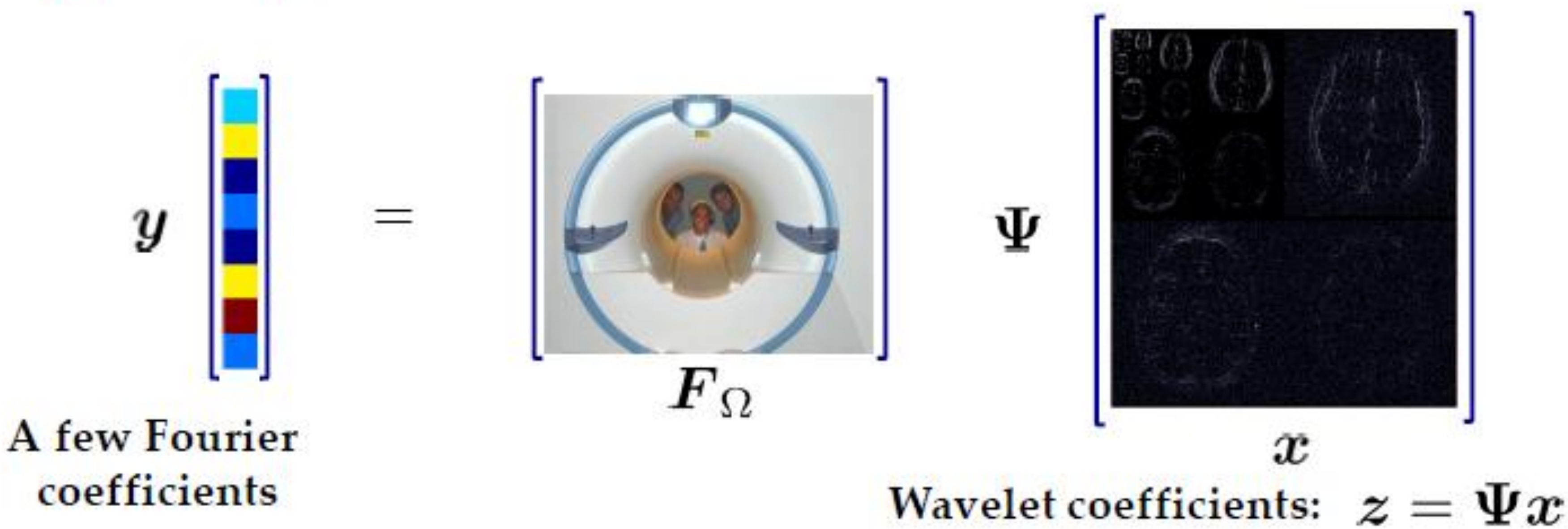
$\boldsymbol{z}$

Image to be sensed

$$y_i = \int_{\boldsymbol{u}} \boldsymbol{z}(u) \exp(-2\pi j \boldsymbol{k}(t_i)^* \boldsymbol{u}) d\boldsymbol{u}$$

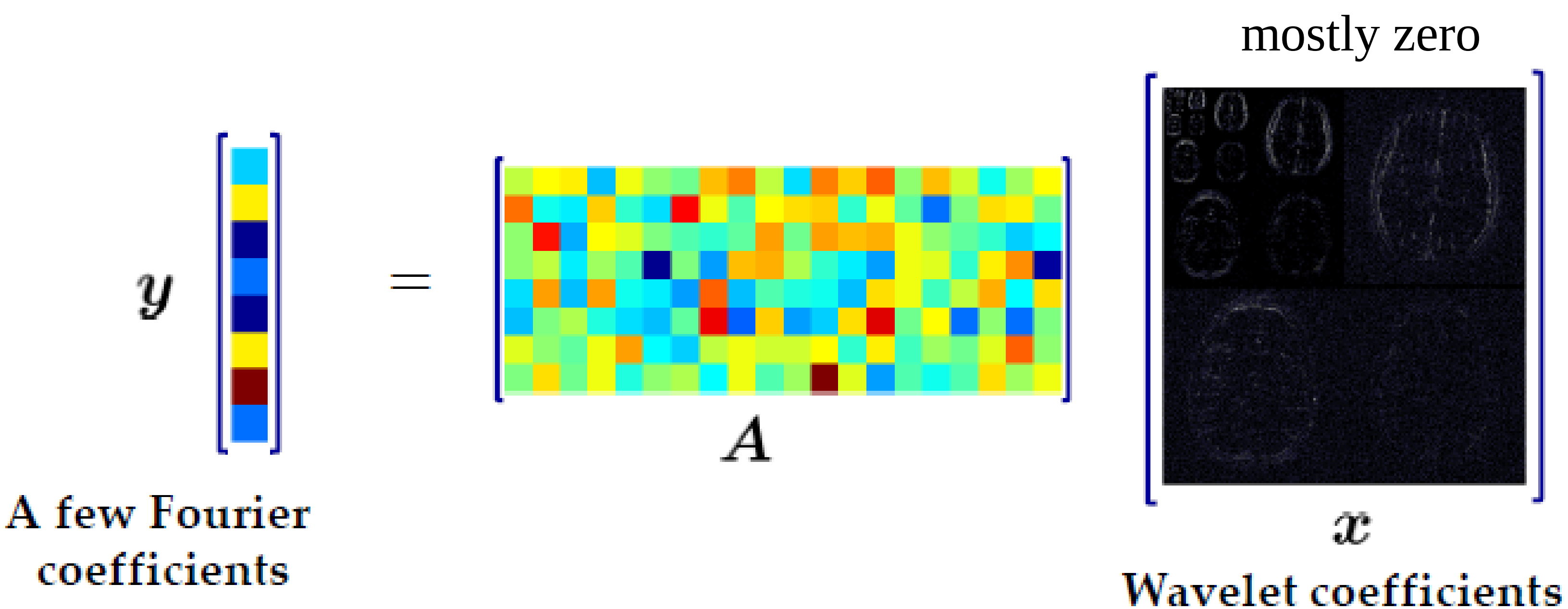
Observations are Fourier coefficients!

Signal Acquisition



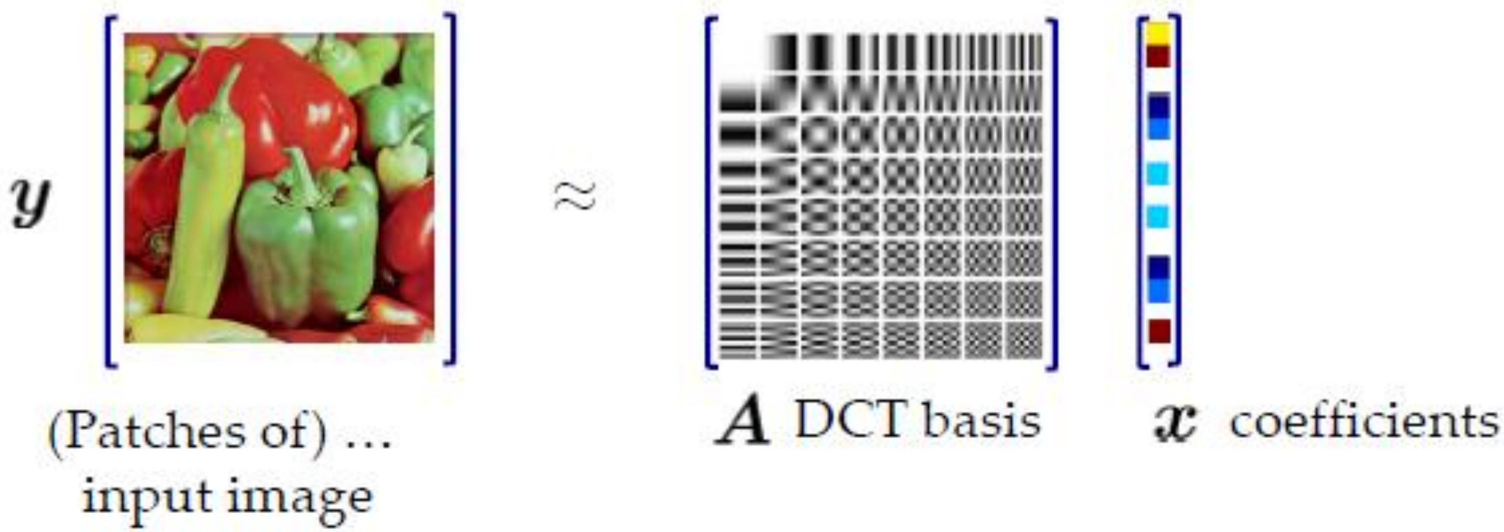
[Lustig, Donoho + Pauly ‘10] … brain image – Lustig ‘12

Signal Acquisition



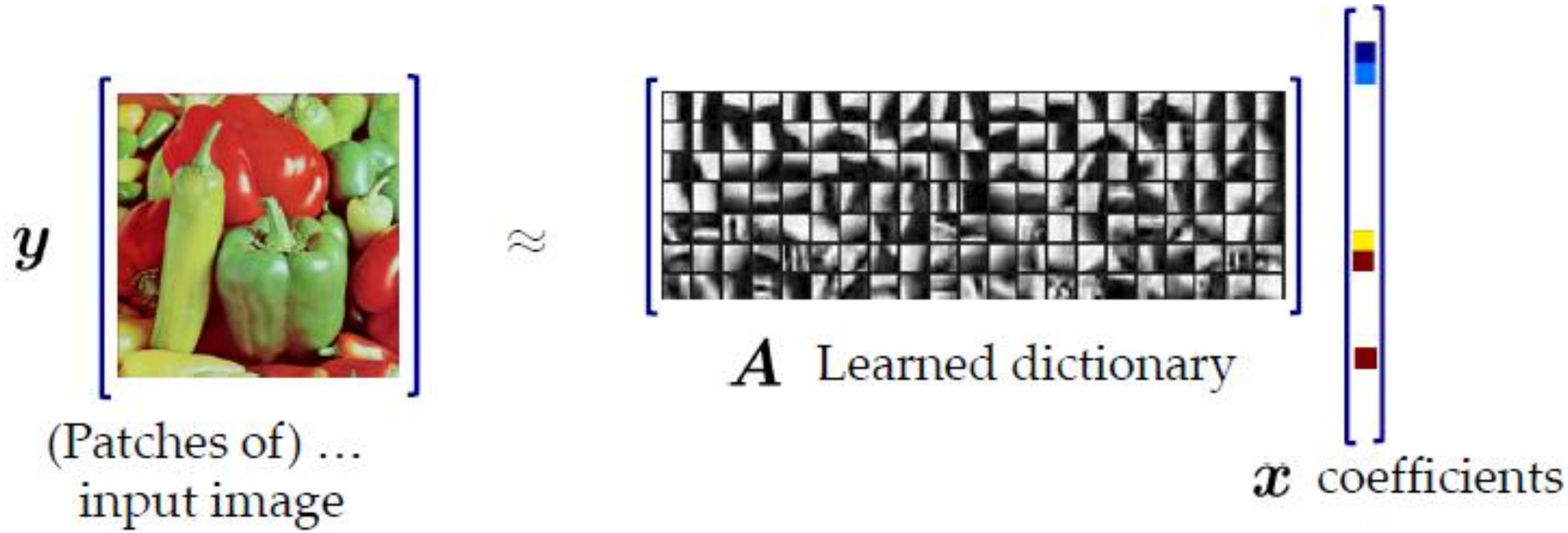
[Lustig, Donoho + Pauly '10] … brain image – Lustig '12

Compression - JPEG



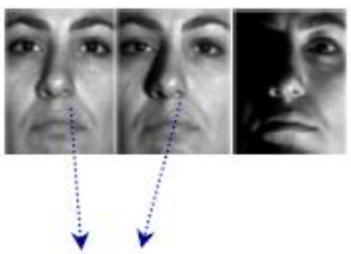
[Wallace '91]

Compression – Learned Dictionary

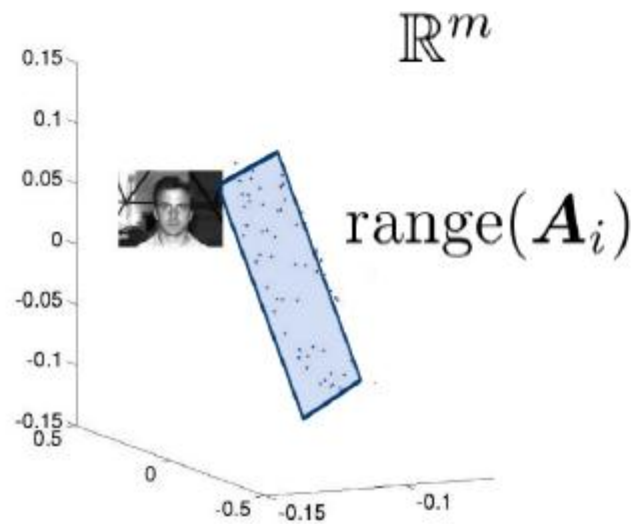


See [Elad+Bryt '08], [Horev et. Al., '12] … Image: [Aharon+Elad '05]

Representing Faces under Different Lighting



$\boldsymbol{A}_i = [\ |\ |\ \dots\ ] \in \mathbb{R}^{m \times n_i}$



$\boldsymbol{y}$  $\approx x_{i,1}$  $+\ x_{i,2}$  $+ \dots +\ x_{i,n}$  $= \boldsymbol{A}_i \boldsymbol{x}_i$

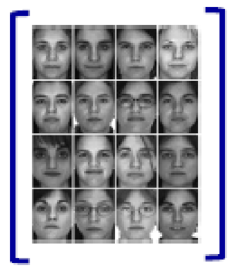
Face Recognition

Generative model for faces, given a database of images from k subjects

$\boldsymbol{y} \in \mathbb{R}^m$

Test image

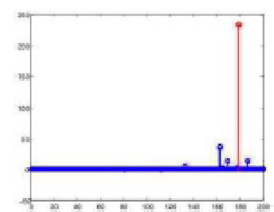
$=$



$\boldsymbol{A} = [\boldsymbol{A}_1 \mid \boldsymbol{A}_2 \mid \cdots \mid \boldsymbol{A}_k]$

Combined training dictionary

$\times$



$\boldsymbol{x} \in \mathbb{R}^n$

coefficients

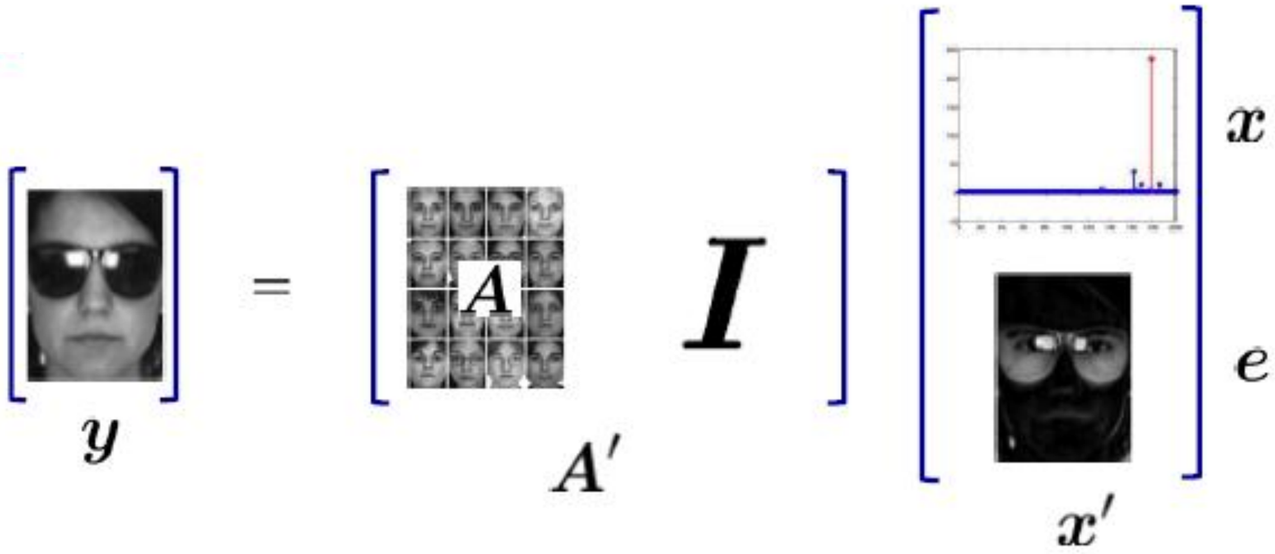
$+$

$\boldsymbol{e} \in \mathbb{R}^m$

corruption, occlusion

[W., Yang, Ganesh, Sastry, Ma '09]

Face Recognition



One large underdetermined system: $y = A'x'$

Sparse Representation:

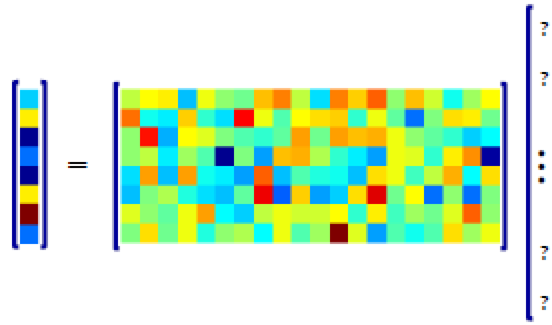
- Given a sparse feasible solution $\mathbf{y} \approx \Phi'\mathbf{x}'$
- Location of large nonzeros in $\mathbf{x}$ should reveal identity

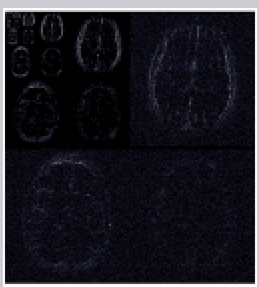
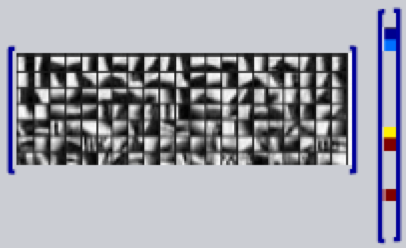
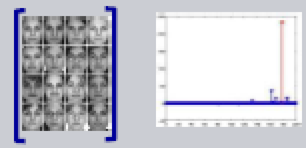
[Wright et al., PAMI 2009]

Prevalence of Sparse Representations

Underdetermined system

$$y = Ax$$



Signal acquisition	Image compression	Face Recognition
 $x^\star$ contains **just a few** significant wavelet coefficients.	 $x^\star$ uses **just a few** dictionary elements.	 $x^\star$ uses **just a few** training faces.  $e^\star$ corrects **a few** gross errors.

Optimization

- Ideal (noiseless) case:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t. } \mathbf{y} = \mathrm{A}\,\mathbf{x}$$

$$\|\mathbf{x}\|_0 = \lim_{p \to 0} \sum_{\mathrm{i}} |x_i|^p = \#\ \text{of nonzero elements in } \mathbf{x}$$

- Approximate case:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathrm{A}\,\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0$$

Uniqueness

Theorem (Gorodnitsky+Rao '97) .
Suppose $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0$*, and let* $k = \|\boldsymbol{x}_0\|_0$*. If* $\operatorname{null}(\boldsymbol{A})$ *contains no* $2k$*-sparse vectors,* $\boldsymbol{x}_0$ *is the unique optimal solution to*

$$\text{minimize } \|\boldsymbol{x}\|_0 \quad \textit{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}.$$

Difficulties

Forward model is linear, the inverse problem is difficult:

1. Combinatorial number of local minima (NP-hard)

2. Objective is discontinuous

$$\text{minimize} \quad \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

INTRACTABLE

Computationally tractable approximate methods are needed …

Replace ℓ_0 Norm with Convex ℓ_1 Norm

- Ideal (noiseless) case:

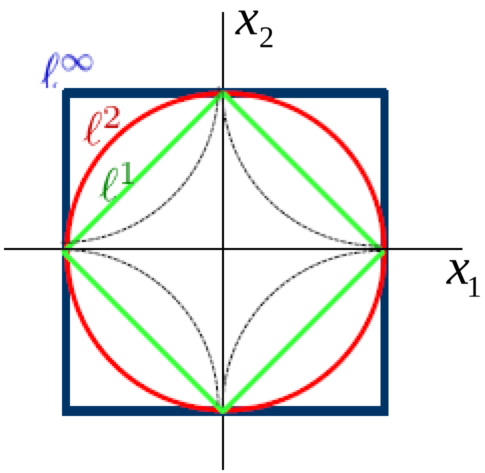
$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t. } \mathbf{y} = \Phi\mathbf{x}$$

$$\|\mathbf{x}\|_1 = \sum_{\mathrm{i}} |x_i|$$

- Approximate case:

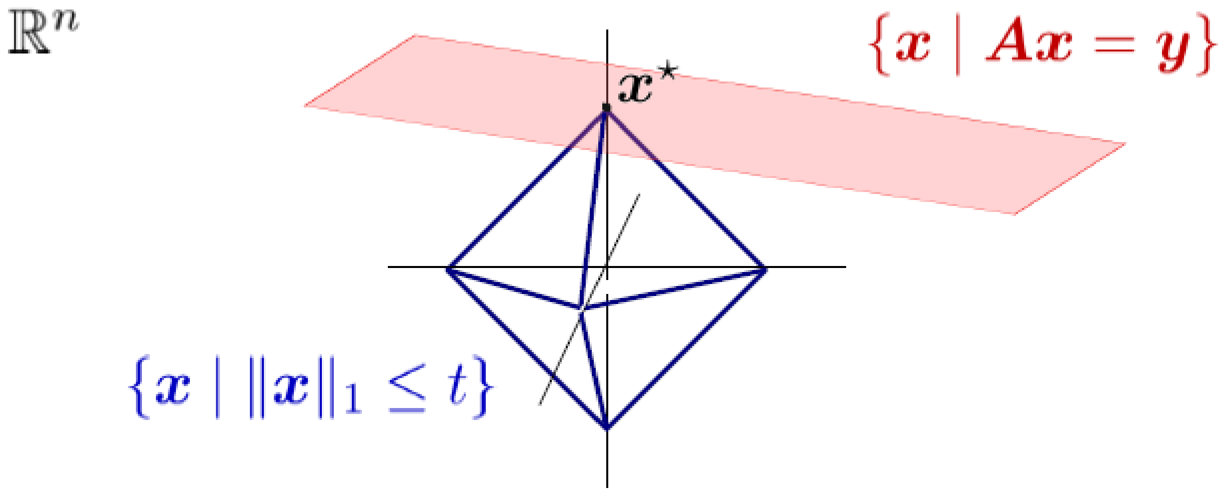
$$\min_{\mathbf{x}} \|\mathbf{y} - \Phi\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

Tightest convex relaxation over unit ball



Why might this work?

$$\text{minimize} \quad \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$



Advantages of ℓ_1 Substitution

- Many fast efficient algorithms (more on this later …)

[Bertsekas, 2003; Yang et al., 2012]

- Many performance guarantees:

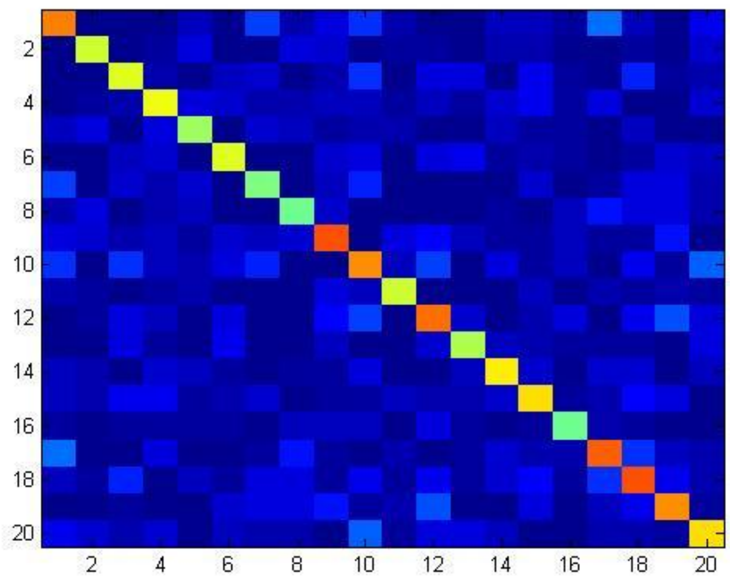
$$\begin{aligned} \mathbf{x}_0 &= \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathrm{A}\,\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0 \\ &\approx \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathrm{A}\,\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \end{aligned}$$

[Candès et al., 2006; Donoho, 2006]

Dictionary Correlation Structure

Low Correlation: Easy

$$A^T A$$



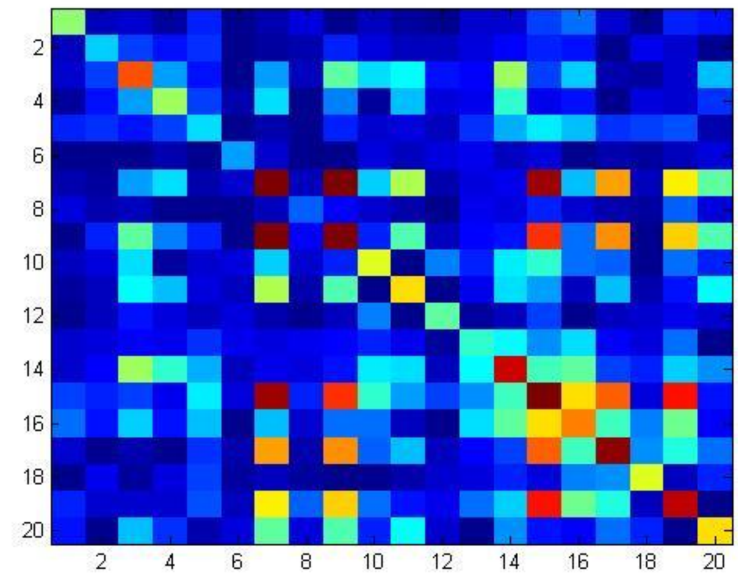
Examples:

$A_{(uncor)} \sim$ iid $N(0,1)$ entries

$A_{(uncor)} \sim$ random rows of DFT

High Correlation: Hard

$$A^T A$$



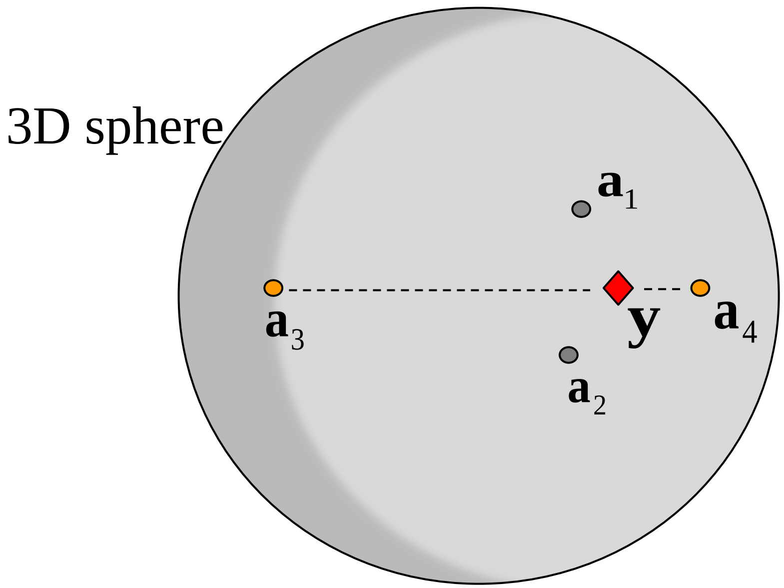
Example:

$$A_{(cor)} = \underbrace{\Psi}_{\text{arbitrary}} A_{(uncor)} \underbrace{\Phi}_{\text{block diagonal}}$$

Example

$$A = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4, \mathbf{a}_4] \qquad \mathbf{x}_0 = [0, 0, 1, 1]^T$$

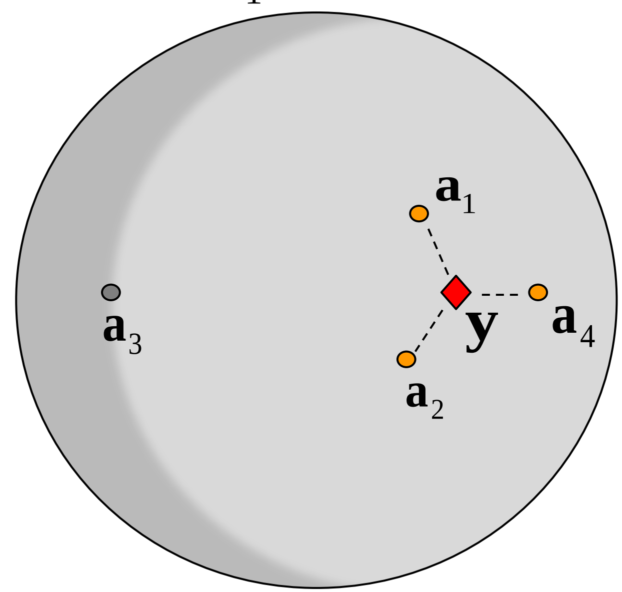
Sparse Generative Solution



$$\mathbf{y} = \mathbf{a}_3 + \mathbf{a}_4$$

$$\|\mathbf{x}\|_1 = 2$$

Minimum ℓ_1 Norm Solution



$$\mathbf{y} = \tfrac{1}{4}\mathbf{a}_1 + \tfrac{1}{4}\mathbf{a}_2 + \tfrac{1}{4}\mathbf{a}_4$$

$$\|\mathbf{x}\|_1 = \tfrac{3}{4}$$

Require conditions to disallow correlated basis vectors in a restricted space

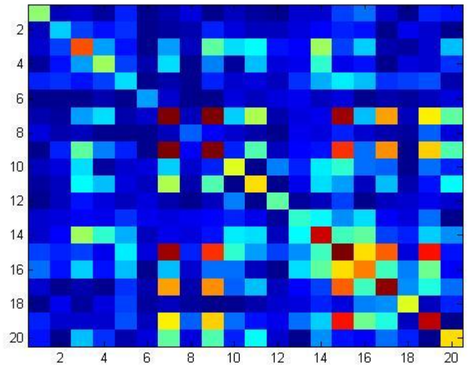
Mutual Coherence

- Let $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$

- Mutual coherence: $\mu(A) = \max_{i \neq j} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}$

- Measures maximum (off-diagonal) correlation among dictionary columns.

$A^T A$



Noiseless Analysis of ℓ_1

Theorem

Assume

$$\|\mathbf{x}_0\|_0 < \frac{1}{2}\left[1 + \frac{1}{\mu(\mathrm{A})}\right]$$

Then $\mathbf{x}_0$ is the unique solution to

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathrm{A}\,\mathbf{x}_0 = \mathrm{A}\,\mathbf{x}$$

[Donoho and Elad, 2003]

Noisy Analysis of ℓ_1

Theorem

Assume $\mathbf{y} = \mathrm{A}\,\mathbf{x}_0 + \boldsymbol{\varepsilon}$ with

$$\|\boldsymbol{\varepsilon}\|_2 \leq \beta \qquad \|\mathbf{x}_0\|_0 < \frac{1}{4}\left[1 + \frac{1}{\mu(\mathrm{A})}\right]$$

Then $$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathrm{A}\,\mathbf{x}\|_2 \leq \beta$$

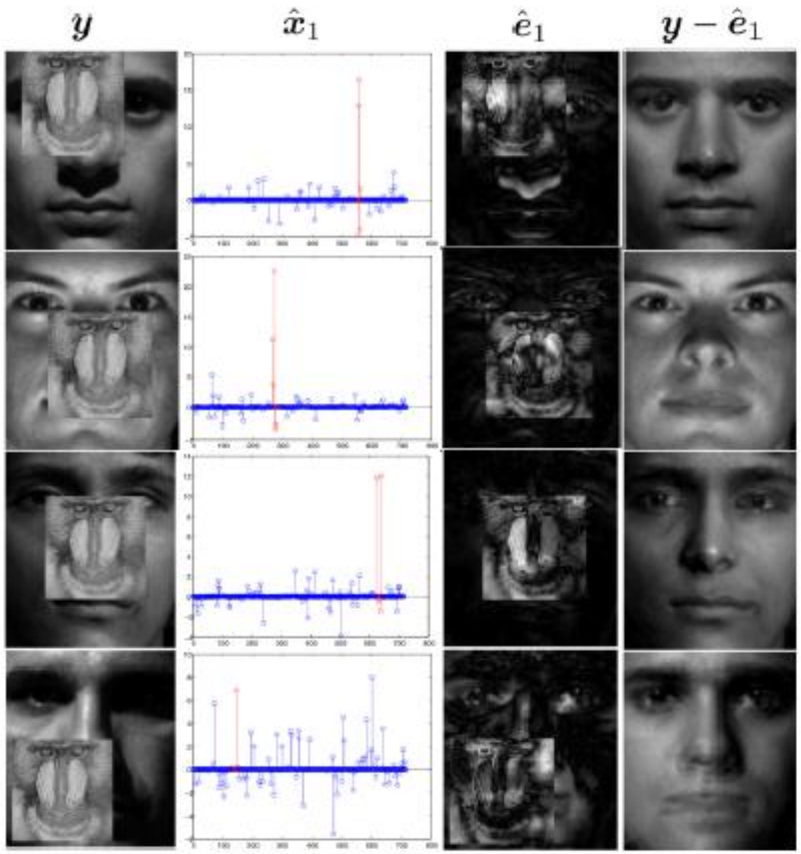
satisfies $$\|\hat{\mathbf{x}} - \mathbf{x}_0\|_2^2 \leq \frac{4\beta^2}{1 - \mu(\mathrm{A})\left[4\|\mathbf{x}_0\|_0 - 1\right]}$$

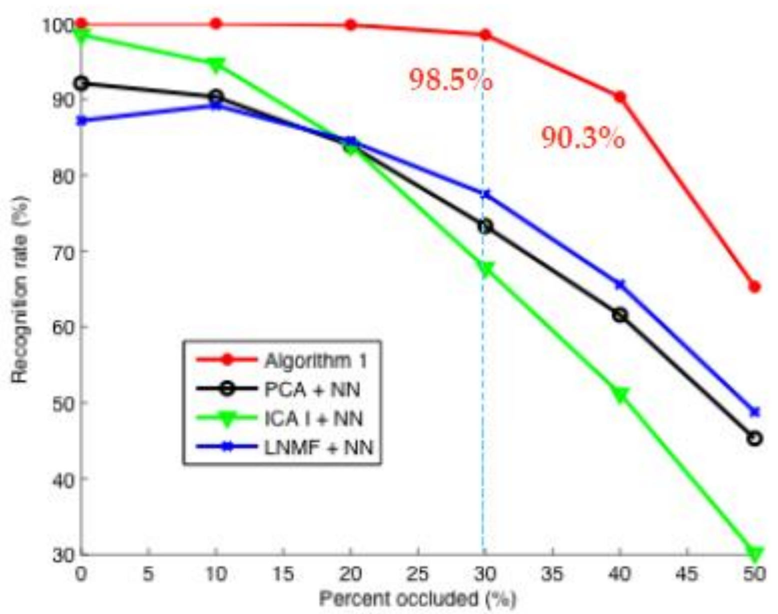
[Donoho et al., 2006]

Many stronger results are possible with added assumptions

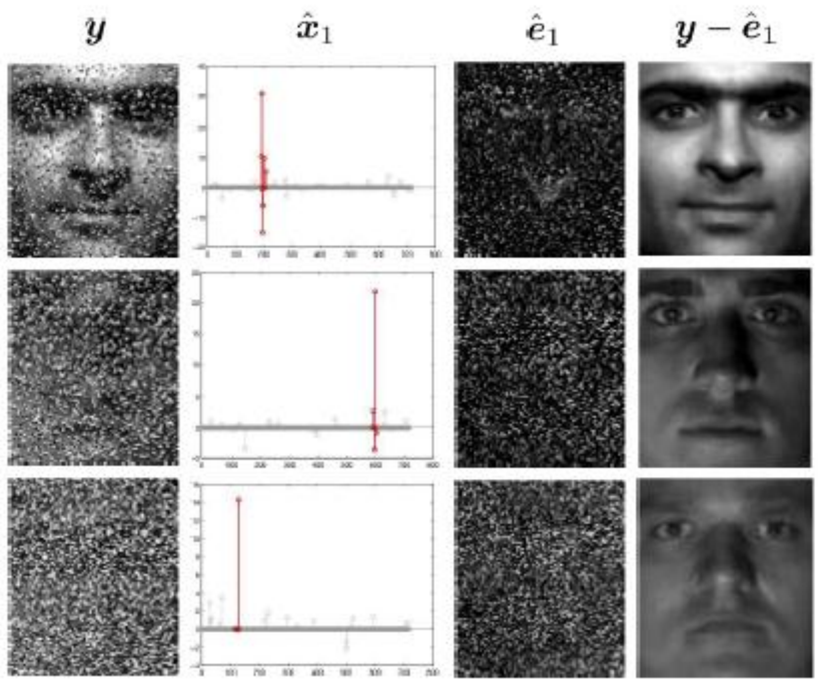
[Candes and Tao, 2005; Candes, 2008]

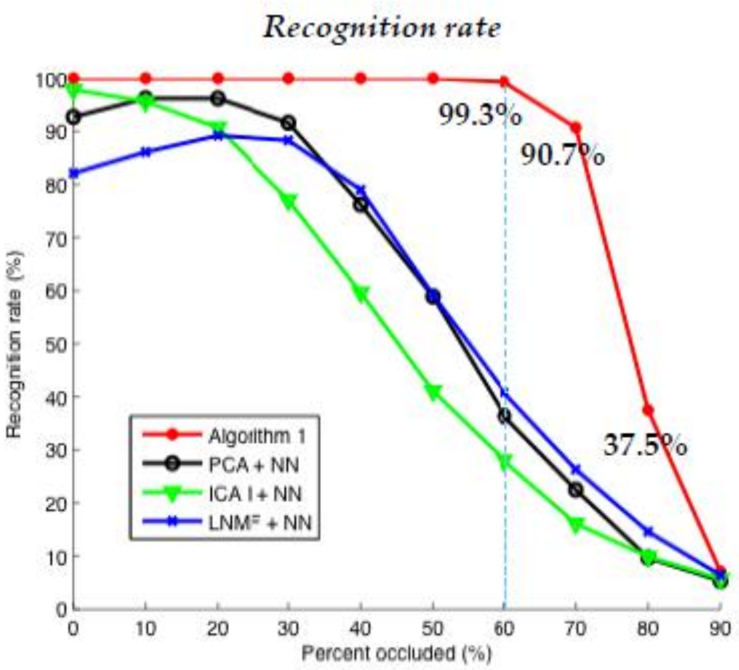
Motivating Example: Face Recognition with Occlusions



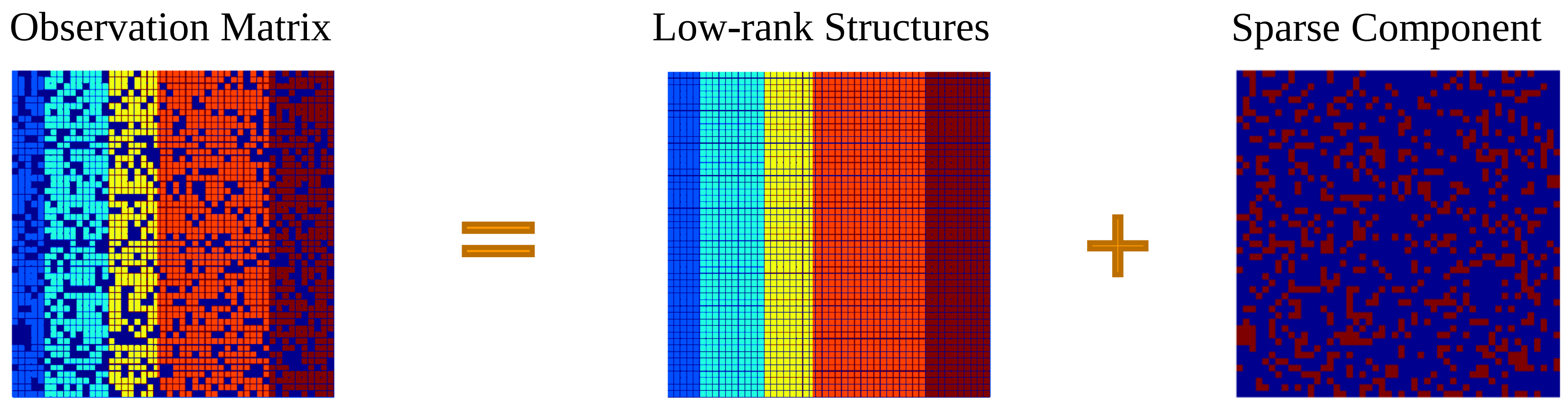


Motivating Example: Face Recognition with Occlusions





Robust PCA



Basic Observation Model

$$Y = X + E + \eta$$

Y	:	$m \times n$ observation matrix, $m \leq n$
X	:	low rank approximation AB^T
E	:	large sparse errors
η	:	Gaussian errors

Classical PCA

$$\min_X \frac{1}{\lambda} \left\| Y - X \right\|_F^2 + \operatorname{rank}\left[X \right]$$

- Simple closed-form solution via SVD.

- **Limitation**: Assumes $E = 0$, i.e., no significant outliers, otherwise the estimate will be poor.

Robust PCA

$$\min_{X,E} \frac{1}{\lambda} \|Y - X - E\|_F^2 + \text{rank}[X] + \frac{1}{n} \|E\|_0$$

- Note: 1/n factor ensures both penalty terms scale between 0 and m (i.e., balanced).

- **Problems**:

  1. Non-convex, NP-hard optimization

  2. Solution may be non-unique

Convex Relaxation
[Candes et al. 2011]

$$\mathrm{rank}(\boldsymbol{X}) = \#\{\sigma_i(\boldsymbol{X}) \neq 0\}. \qquad \|\boldsymbol{E}\|_0 = \#\{\boldsymbol{E}_{ij} \neq 0\}.$$

$$\downarrow\downarrow \qquad\qquad \downarrow\downarrow$$

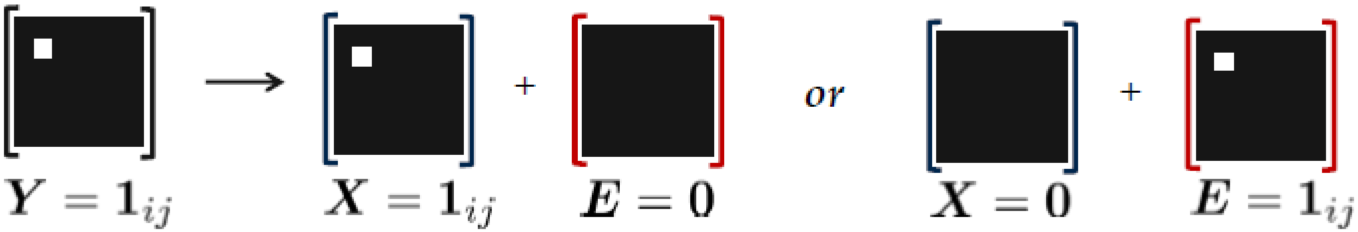
$$\|\boldsymbol{X}\|_* = \sum_i \sigma_i(\boldsymbol{X}). \qquad \|\boldsymbol{E}\|_1 = \sum_{ij} |\boldsymbol{E}_{ij}|.$$

- **Solve:** $$\min_{X,E} \frac{1}{\lambda}\|Y - X - E\|_F^2 + \|X\|_* + \frac{1}{\sqrt{n}}\|E\|_1$$

- **Problem**: Provable recovery guarantees exist, but must still resolve non-uniqueness issues.

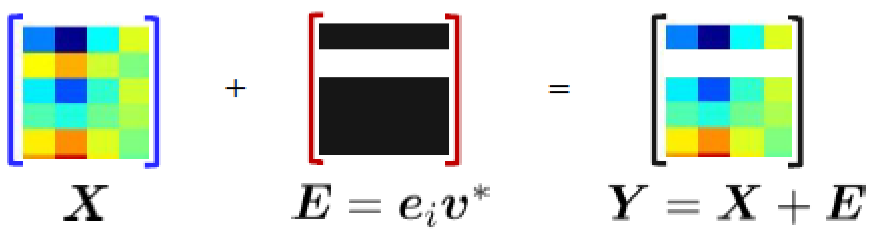
Non-Uniqueness Issues

Some very sparse matrices are also low-rank:



Can we recover $\boldsymbol{X}$ that are incoherent with the standard basis?

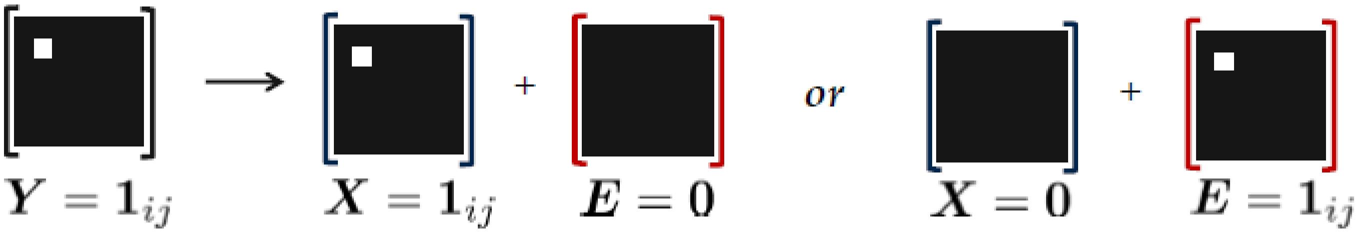
Certain sparse error patterns $\boldsymbol{E}$ make recovering $\boldsymbol{X}$ impossible:



Can we correct $\boldsymbol{E}$ whose support is not adversarial?

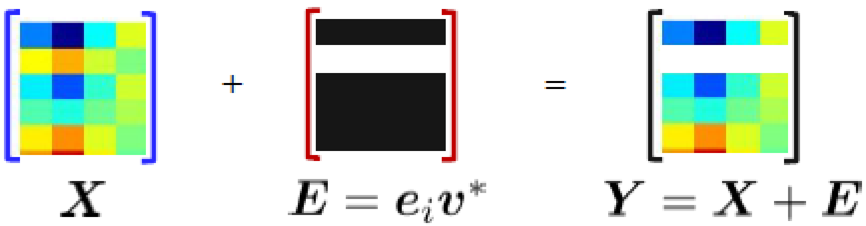
Non-Uniqueness Issues

Some very sparse matrices are also low-rank:



Can we recover $\boldsymbol{X}$ *that are incoherent with the standard basis?*

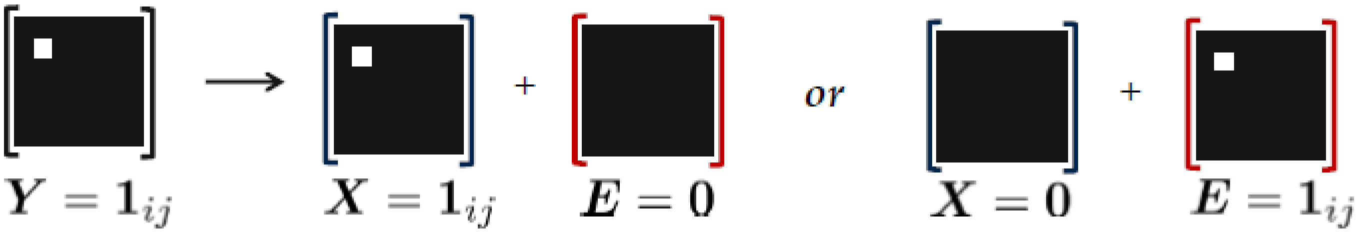
Certain sparse error patterns $\boldsymbol{E}$ *make recovering* $\boldsymbol{X}$ *impossible:*



Can we correct $\boldsymbol{E}$ *whose support is not adversarial?*

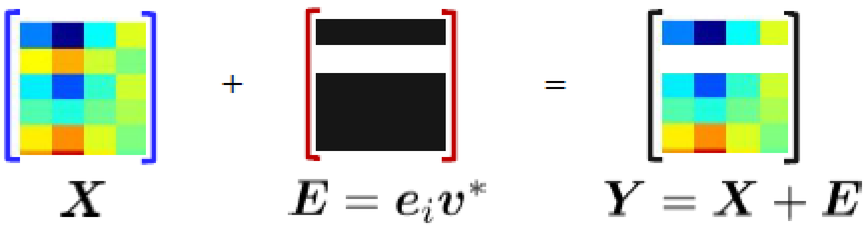
Non-Uniqueness Issues

Some very sparse matrices are also low-rank:



Can we recover $\boldsymbol{X}$ *that are incoherent with the standard basis?*

Certain sparse error patterns $\boldsymbol{E}$ *make recovering* $\boldsymbol{X}$ *impossible:*



Can we correct $\boldsymbol{E}$ *whose support is not adversarial?*

Resolving Ambiguity with Incoherence Conditions

Can we recover $\boldsymbol{X}$ that are ***incoherent*** *with the standard basis from* ***almost all*** *errors $\boldsymbol{E}$?*

Incoherence condition on singular vectors, **singular values arbitrary:**

Singular vectors of $\boldsymbol{X}$ not too spiky: $\begin{cases} \max_i \|\boldsymbol{U}_i\|^2 \leq \mu r/m. \\ \max_i \|\boldsymbol{V}_i\|^2 \leq \mu r/n. \end{cases}$

not too cross-correlated: $\|\boldsymbol{U}\boldsymbol{V}^*\|_\infty \leq \sqrt{\mu r/mn}$

Uniform model on error support, **signs and magnitudes arbitrary:**

$$\mathrm{support}(\boldsymbol{E}) \sim \mathrm{uni}\binom{[m]\times[n]}{\rho mn}$$

Incoherence condition: [Candès + Recht '08]

Main Result – Correct Recovery

Theorem

If $X_0 \in \Re^{m \times n}$, $n \geq m$ has rank

$$r \leq \rho_r \frac{m}{\mu[\log(n)]^2}$$

and E_0 has Bernoulli support with error probability $\varepsilon \leq \rho_s nm$,
then with very high probability

$$\{X_0, E_0\} = \arg\min_{X,E} \|X\|_* + \frac{1}{\sqrt{n}}\|E\|_1 \quad \text{s.t. } Y = X + E$$

and the minimizer is unique

"Convex optimization recovers matrices of rank $O\left(\frac{m}{\log^2(n)}\right)$ from errors corrupting $O(mn)$ entries"

[Candes, Li, Ma, Wright; 2009]

A Suite of Models and Theoretical Guarantees

For robust recovery of a family of low-dimensional structures:

- [Zhou et. al. '09] **Spatially contiguous** sparse errors via MRF
- [Bach '10] – structured relaxations from **submodular functions**
- [Negahban+Yu+Wainwright '10] – **geometric analysis** of recovery
- [Becker+Candès+Grant '10] – **algorithmic templates**
- [Xu+Caramanis+Sanghavi '11] **column sparse errors $L_{2,1}$ norm**
- [Recht+Parillo+Chandrasekaran+Wilsky '11] – **compressive sensing of various structures**
- **[Candes+Recht '11] – compressive sensing of decomposable structures**

$$X^0 = \arg\min \|X\|_\diamond \quad \text{s.t.} \quad \mathcal{P}_Q(X) = \mathcal{P}_Q(X^0)$$

- **[McCoy+Tropp'11] – decomposition of sparse and low-rank structures**

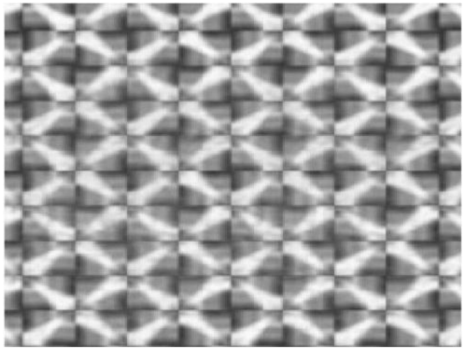
$$(X_1^0, X_2^0) = \arg\min \|X_1\|_{(1)} + \lambda \|X_2\|_{(2)} \;\; \text{s.t.} \;\; X_1 + X_2 = X_1^0 + X_2^0$$

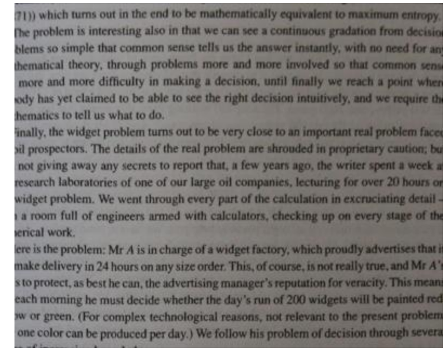
- **[W.+Ganesh+Min+Ma, I&I'13] – superposition of decomposable structures**

$$(X_1^0, \ldots, X_k^0) = \arg\min \sum \lambda_i \|X_i\|_{(i)} \;\; \text{s.t.} \;\; \mathcal{P}_Q(\textstyle\sum_i X_i) = \mathcal{P}_Q(\textstyle\sum_i X_i^0)$$

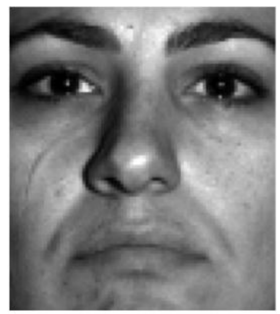
Take home message: Let the data and application tell you the structure…

Applications – *Low rank structures in visual data*





(71)) which turns out in the end to be mathematically equivalent to maximum entropy.
The problem is interesting also in that we can see a continuous gradation from decisio
blems so simple that common sense tells us the answer instantly, with no need for an
thematical theory, through problems more and more involved so that common sens
more and more difficulty in making a decision, until finally we reach a point when
ody has yet claimed to be able to see the right decision intuitively, and we require th
hematics to tell us what to do.
inally, the widget problem turns out to be very close to an important real problem face
il prospectors. The details of the real problem are shrouded in proprietary caution; bu
not giving away any secrets to report that, a few years ago, the writer spent a week a
research laboratories of one of our large oil companies, lecturing for over 20 hours or
widget problem. We went through every part of the calculation in excruciating detail –
a room full of engineers armed with calculators, checking up on every stage of the
erical work.
ere is the problem: Mr *A* is in charge of a widget factory, which proudly advertises that i
make delivery in 24 hours on any size order. This, of course, is not really true, and Mr *A*'
s to protect, as best he can, the advertising manager's reputation for veracity. This mean
each morning he must decide whether the day's run of 200 widgets will be painted red
w or green. (For complex technological reasons, not relevant to the present problem
one color can be produced per day.) We follow his problem of decision through severa

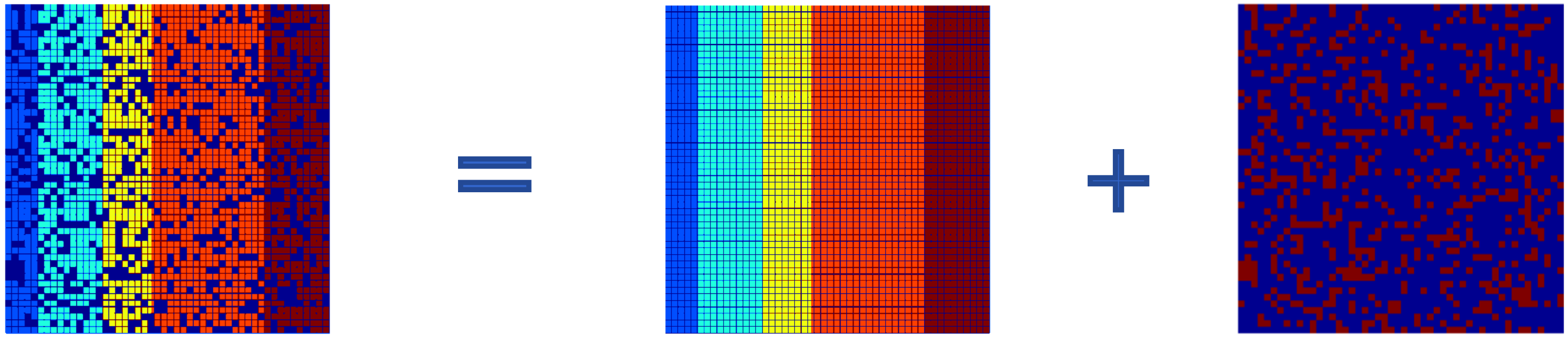


Visual data exhibit ***low-dimensional structures*** due to rich ***local*** regularities, ***global*** symmetries, ***repetitive*** patterns, or ***redundant*** sampling.

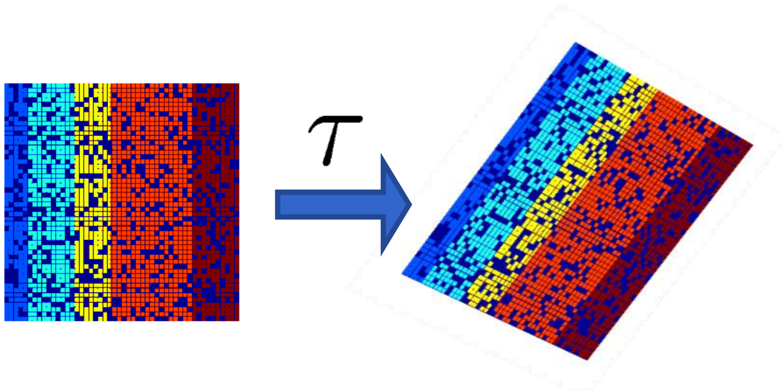
Sensing or Imaging of Low-Rank and Sparse Structures

Basic Decomposition:

corrupted data Low-rank Structures Sparse Structures



Generalization to visual data: add nonlinear deformation τ?



Real Face Images from the Internet: Low-Rank Structures?

*48 images collected from internet

Robust Alignment of Multiple (Face) Images

D – corrupted & misaligned observation | A – aligned low-rank images | E – sparse errors

Problem: Given $D \circ \tau = A_0 + E_0$, recover τ, A_0 and E_0.

Parametric deformations (rigid, affine, projective…)

Low-rank component

Sparse component

Objective: Robust Alignment via Low-rank and Sparse (**RASL**) Decomposition

$$\min \ \|A\|_* + \lambda \|E\|_1 \quad \text{subj} \quad A + E = D \circ \tau$$

Solution: *Iteratively solving the linearized convex program:*

$$\min \ \|A\|_* + \lambda \|E\|_1 \quad \text{subj} \quad A + E = D \circ \tau_k + J \cdot \Delta\tau$$

RASL: *Detected Faces*

Input: faces from a face detector (D)

Average

RASL: *Faces Aligned*

Output: aligned faces ($D \circ \tau$)

Average

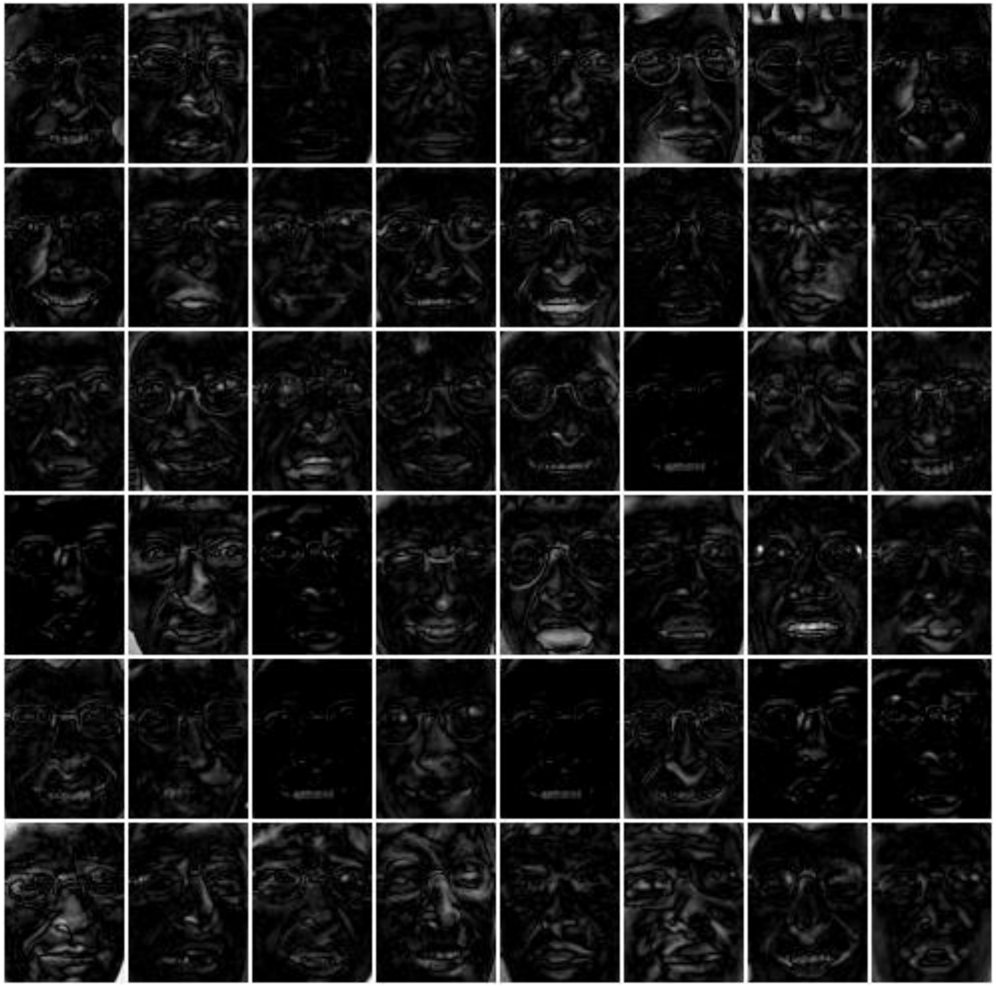
RASL: *Faces Cleaned as the Low-Rank Component*

Output: clean low-rank faces (A)

Average

RASL: *Sparse Errors of the Face Images*

Output: sparse error images (E)



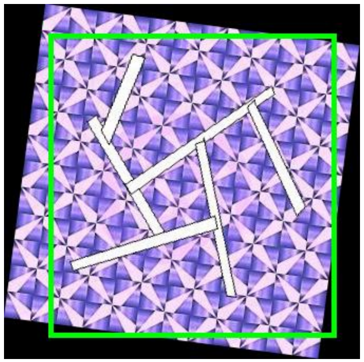
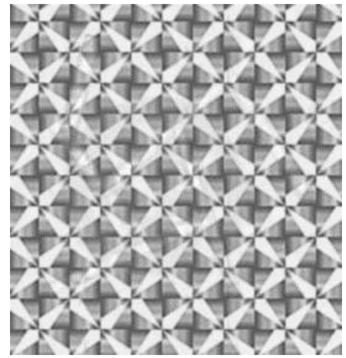
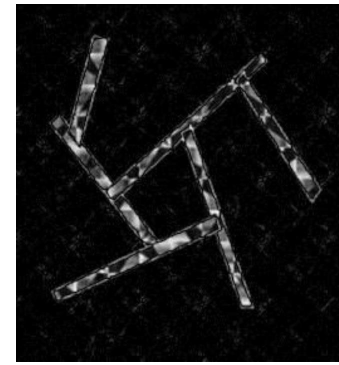
RASL: *Video Stabilization and Enhancement*

Original video (D)   Aligned video ($D \circ \tau$)   Low-rank part (A)   Sparse part (E)

Peng, Ganesh, Wright, Ma, CVPR'10, TPAMI'11

Reconstructing 3D Geometry and Structures

D – deformed observation | A – low-rank structures | E – sparse errors

 $\circ\,\tau \;=$  $+$ 

Problem: Given $D \circ \tau = A_0 + E_0$, recover τ, A_0 and E_0 simultaneously.

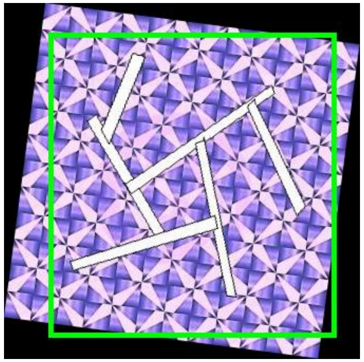
Low-rank component (regular patterns…)

Sparse component (occlusion, corruption, foreground…)

Parametric deformations (affine, projective, radial distortion, 3D shape…)

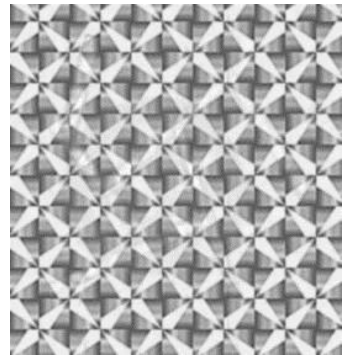
Transform Invariant Low-rank Textures (TILT)

D – deformed observation



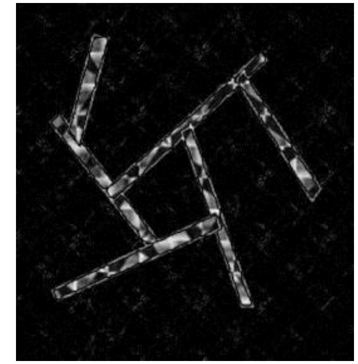
$\circ \tau =$

A – low-rank structures



$+$

E – sparse errors



Objective: *Transformed Robust PCA:*

$$\min \ \|A\|_* + \lambda \|E\|_1 \ \ \text{subj} \ \ A + E = D \circ \tau$$

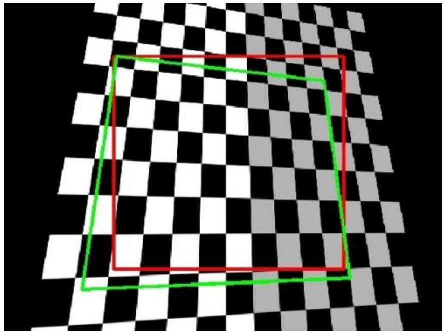
Solution: *Iteratively solving the linearized convex program:*

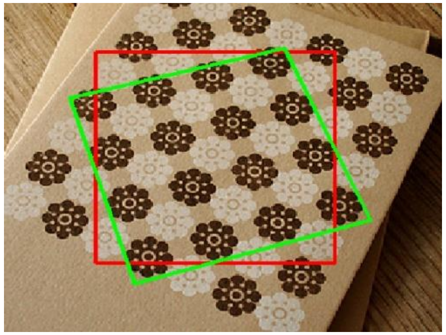
$$\min \ \|A\|_* + \lambda \|E\|_1 \ \ \text{subj} \ \ A + E = D \circ \tau_k + J \cdot \Delta\tau$$

Zhang, Liang, Ganesh, Ma, ACCV'10, IJCV'12

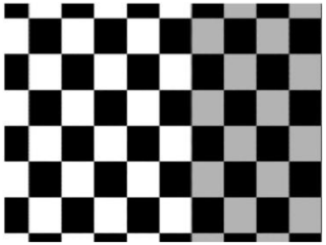
TILT: *Shape from texture*

Input (red window D)



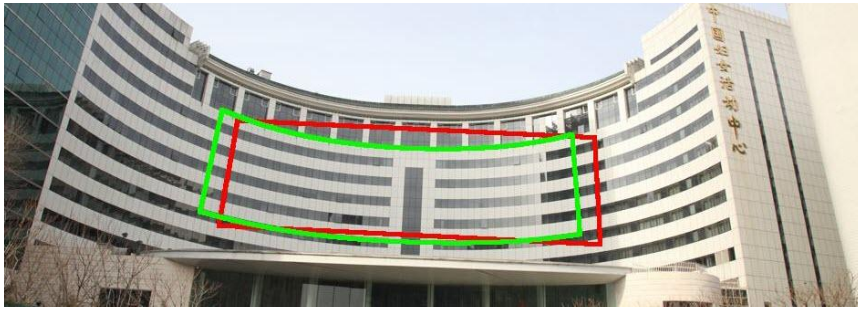


Output (rectified green window A)



Zhang, Liang, Ganesh, Ma, ACCV'10, IJCV'12

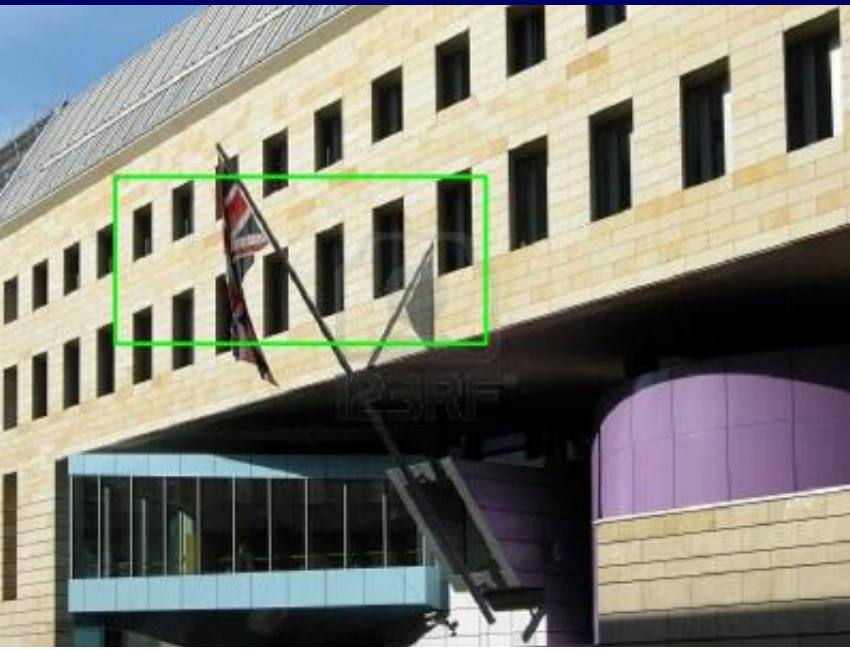
TILT: *Virtual reality*

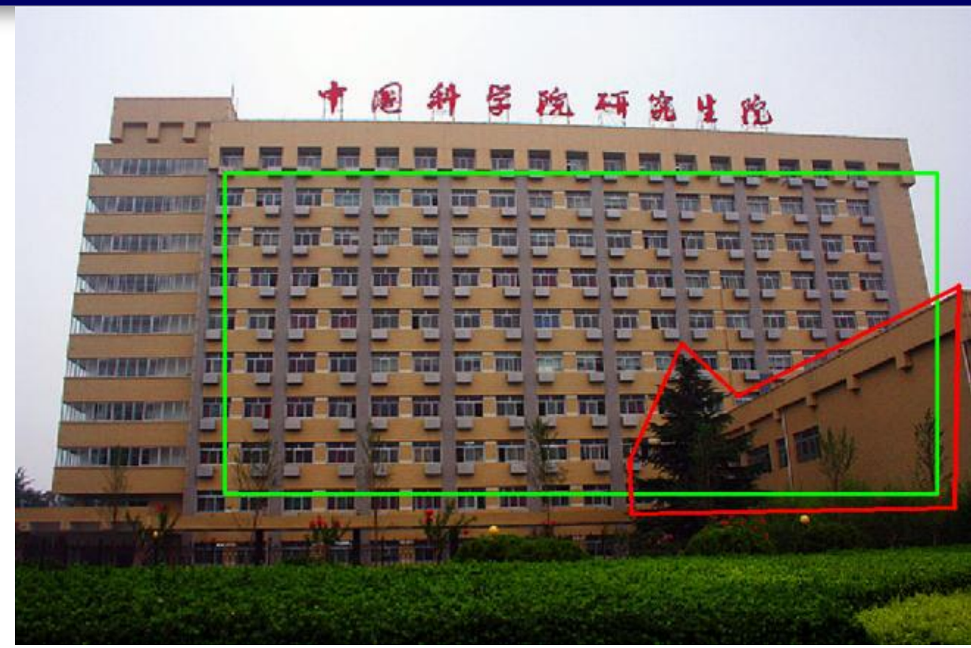


Zhang, Liang, and Ma, in ICCV 2011

Virtual Reality in Urban Scenes

Structured Texture Completion and Repairing





Structured Texture Completion and Repairing

TILT

Photoshop

Input

Output

Regularity of Texts at All Scales

Input (red window D)

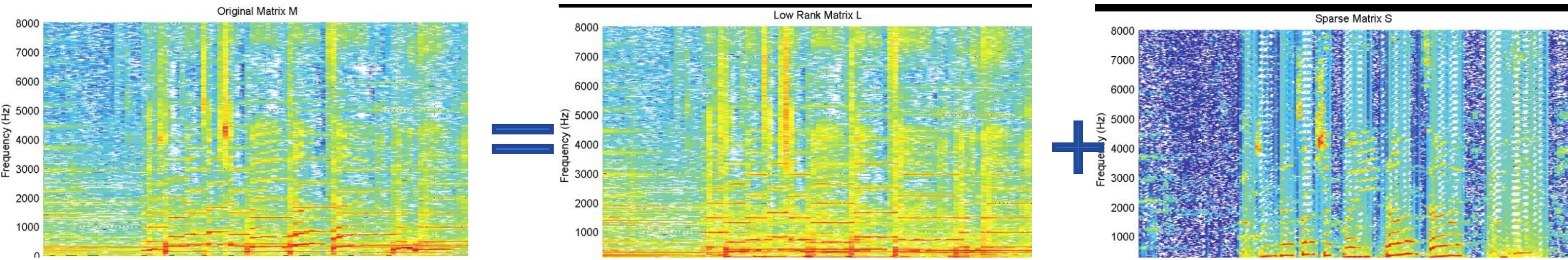
Output (rectified green window A)

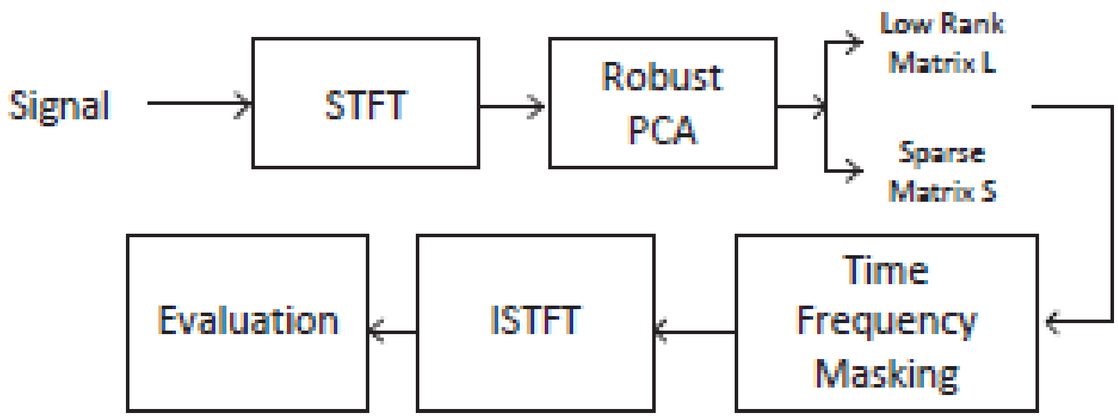
Rectification can lead to more robust recognition

Zhang, Liang, Ganesh, Ma, ACCV'10 and IJCV'12

Other Data/Applications: Lyrics and Music Separation

Songs (STFT) Low-rank (music) Sparse (voices)





Other Data/Applications: Protein-Gene Correlation

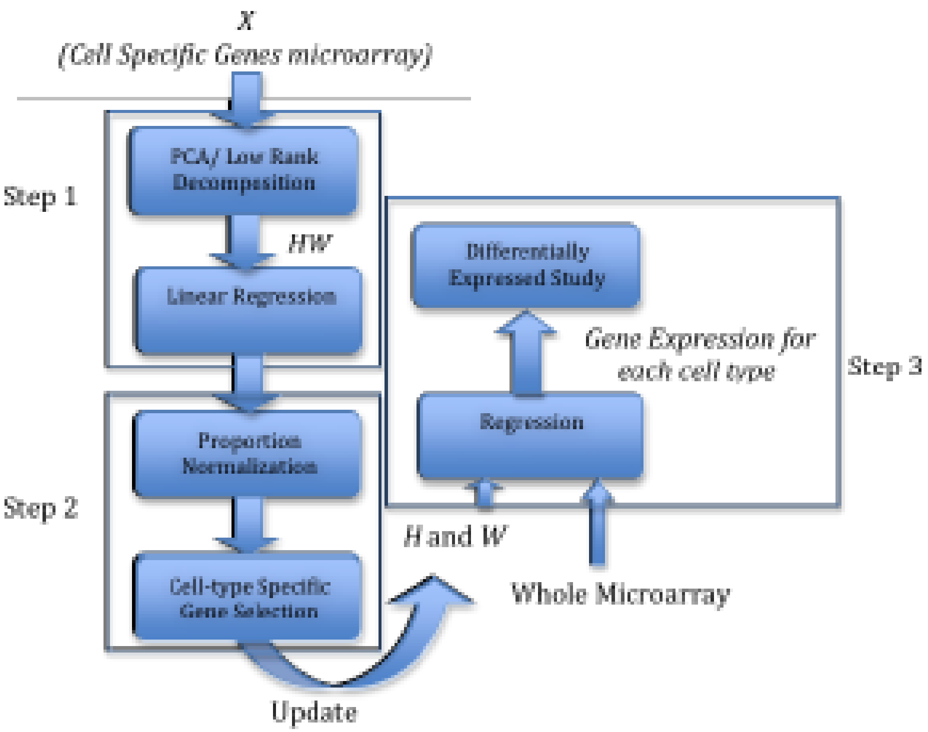


Fig. 1. The diagram of the workflow of the method presented in this paper.

Microarray data

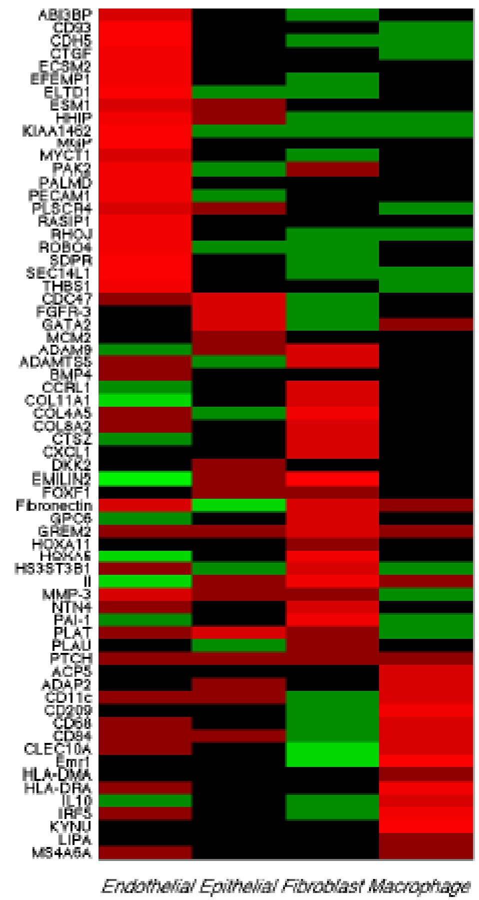


Fig. 6. HeatMap of estimated gene signatures for the sorted cell specific genes after adjustments based on fold changes. RPCA is used in the first step. It is clear that this matrix is close to a block diagonal structure.

Wang, Machiraju, and Huang, submitted to Bioinformatics 2012.

Take-home Messages for Visual Data Processing:

1. (Transformed) **low-rank and sparse** structures are central to visual data modeling, processing, and analyzing;

2. Such structures can now be extracted **correctly, robustly, and efficiently**, from raw image pixels (or high-dim features);

3. These new algorithms **unleash tremendous local or global information** from multiple or single images, emulating or surpassing human perception;

4. These algorithms start to exert significant impact on **image/video processing, 3D reconstruction, and object recognition**.

… …

But try not to abuse or misuse them…

OTHER REFERENCES + ACKNOWLEDGEMENT

Core References:

- *RASL: Robust Alignment by Sparse and Low-rank Decomposition*? Peng, Ganesh, Wright, Xu, and Ma, Trans. PAMI, 2012.
- *TILT: Transform Invariant Low-rank Textures,* Zhang, Liang, Ganesh, and Ma, IJCV 2012.
- *Compressive Principal Component Pursuit*, Wright, Ganesh, Min, and Ma, ISIT 2012.

More references, codes, and applications on the website:

http://perception.csl.illinois.edu/matrix-rank/home.html

Colleagues:

- Prof. Emmanuel Candes (Stanford)
- Prof. John Wright (Columbia)
- Prof. Zhouchen Lin (Peking University)
- Dr. Yasuyuki Matsushita (MSRA)
- Dr. Allen Yang (Berkeley)
- Dr. Arvind Ganesh (IBM Research, India)
- Prof. Shuicheng Yan (Na. Univ. Singapore)
- Prof. Jian Zhang (Sydney Tech. Univ.)
- Prof. Lei Zhang (HK Polytech Univ.)
- Prof. Liangshen Zhuang (USTC)

Students:

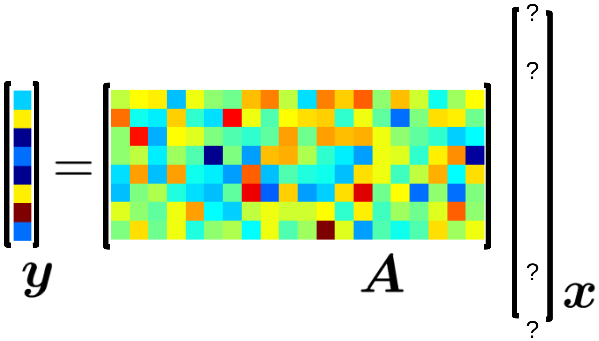
- Zhengdong Zhang (MSRA, Tsinghua University)
- Xiao Liang (MSRA, Tsinghua University)
- Xin Zhang (MSRA, Tsinghua University)
- Kerui Min (UIUC)
- Dr. Zhihan Zhou (UIUC)
- Dr. Hossein Mobahi (UIUC)
- Dr. Guangcan Liu (UIUC)
- Dr. Xiaodong Li (Stanford)

Part II: Optimization for Low-Dimensional Structures

Two convex optimization problems

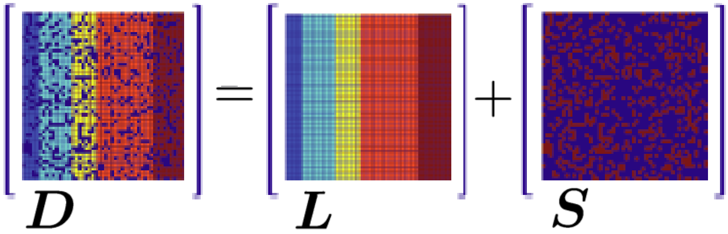
ℓ^1 **minimization** seeks a **sparse solution** to an **underdetermined** linear system of equations:

$$\min \ \|\boldsymbol{x}\|_1 \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$$



Robust PCA expresses an input data matrix as a sum of a **low-rank** matrix $\boldsymbol{L}$ and a **sparse** matrix $\boldsymbol{S}$.

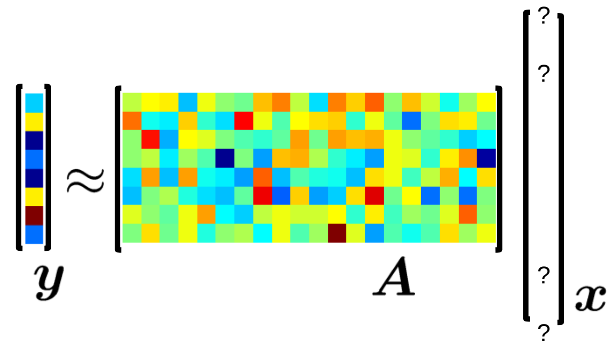
$$\min \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \ \text{ s.t. } \ \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$



Two noise-aware variants

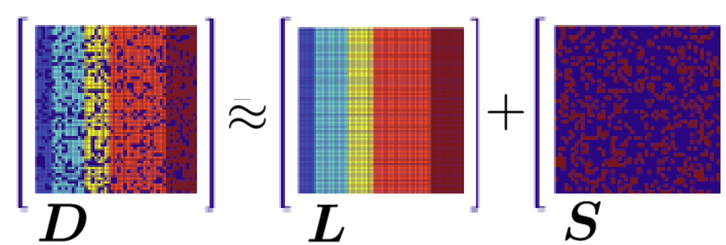
Basis pursuit denoising seeks a **sparse** ***near*-solution** to an **underdetermined** linear system:

$$\min \ \|\boldsymbol{x}\|_1 \ + \ \tfrac{\lambda}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

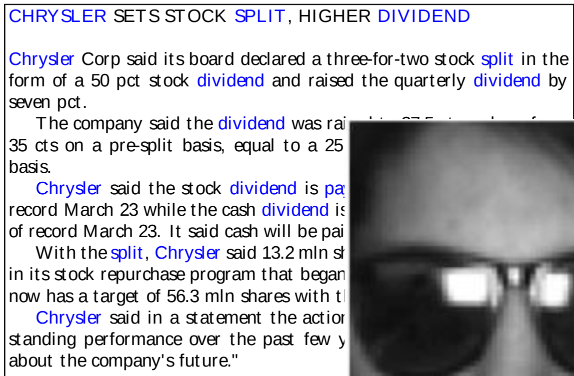


Noise-aware Robust PCA *approximates* an input data matrix as a sum of a **low-rank** matrix $\boldsymbol{L}$ and a **sparse** matrix $\boldsymbol{S}$.

$$\min \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \tfrac{\gamma}{2}\|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$$



Many possible applications …



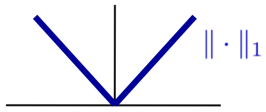
CHRYSLER SETS STOCK SPLIT, HIGHER DIVIDEND

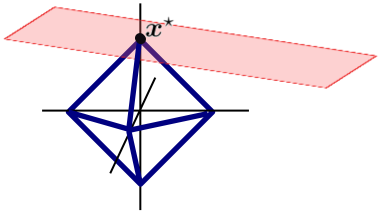
Chrysler Corp said its board declared a three-for-two stock split in the form of a 50 pct stock dividend and raised the quarterly dividend by seven pct.
The company said the dividend was rai
35 cts on a pre-split basis, equal to a 25
basis.
Chrysler said the stock dividend is pa
record March 23 while the cash dividend is
of record March 23. It said cash will be pai
With the split, Chrysler said 13.2 mln sh
in its stock repurchase program that began
now has a target of 56.3 mln shares with t
Chrysler said in a statement the actio
standing performance over the past few y
about the company's future."

… *if* we can solve these core optimization problems
accurately, efficiently, *and* ***scalably.***

Key challenges: nonsmoothness and scale

Nonsmoothness: structure-inducing regularizers such as $\| \cdot \|_1, \ \| \cdot \|_*$ are **not differentiable**:



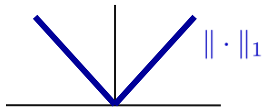


Great for structure recovery …

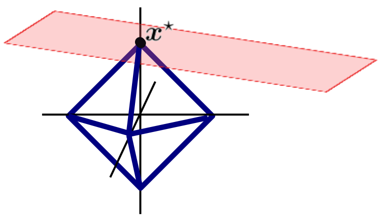
… challenging for optimization.

Key challenges: nonsmoothness and scale

Nonsmoothness: structure-inducing regularizers such as $\|\cdot\|_1$, $\|\cdot\|_*$ are **not differentiable**:



Great for structure recovery …
… challenging for optimization.



Scale … typical problems involve $\mathbf{10^4 - 10^6}$ **unknowns**, or more.

$$\text{Time} = (\#\text{iterations for an } \varepsilon\text{-accurate soln.}) \times (\text{time per iteration})$$

Classical **interior point methods** (e.g., SeDuMi, SDPT3): great convergence rate (linear or better), but $\Omega(\#\text{unknowns}^3)$ cost per iteration. *High accuracy for small problems.*

First-order (gradient-like) algorithms: slower (sublinear) convergence rate, but very cheap iterations. *Moderate accuracy even for large problems.*

Why care? Practical impact of algorithm choice

Time required to solve a 1,000 x 1,000 matrix recovery problem:

Algorithm	Accuracy	Rank	$\|E\|_0$	# iterations	time (sec)
IT	5.99e-006	50	101,268	8,550	**119,370.3**
DUAL	8.65e-006	50	100,024	822	1,855.4
APG	5.85e-006	50	100,347	134	1,468.9
APG_P	5.91e-006	50	100,347	134	82.7
$EALM_P$	2.07e-007	50	100,014	34	37.5
$IALM_P$	3.83e-007	50	99,996	23	**11.8**

Four orders of magnitude improvement, just by choosing the right algorithm to solve the convex program.

This is the difference between theory that will have impact "someday" and practical computational techniques that can be applied right now…

This lecture: Three key techniques

In this hour lecture, we will focus on **three recurring ideas** that allow us to address the challenges of nonsmoothness and scale:

Proximal gradient methods: coping with *nonsmoothness*

Optimal first-order methods: *accelerating convergence*

Augmented Lagrangian methods: handling *constraints*

Why worry about nonsmoothness?

The best uniform **rate of convergence** for **first-order methods*** for minimizing $f \in \mathcal{F}$ depends very strongly on smoothness:

Function class $\mathcal{F}$	$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth  f convex, differentiable $\quad \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \le L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
nonsmooth  f convex $\quad \lvert f(\boldsymbol{x}) - f(\boldsymbol{x}')\rvert \le M\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CM\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

* *Such as gradient descent. See e.g., Nesterov, "Introductory Lectures on Convex Optimization"*

Why worry about nonsmoothness?

The best uniform **rate of convergence** for **first-order methods*** for minimizing $f \in \mathcal{F}$ depends very strongly on smoothness:

Function class $\mathcal{F}$		$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth 	f convex, differentiable $\\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\\| \leq L\\|\boldsymbol{x} - \boldsymbol{x}'\\|$	$\frac{CL\\|\boldsymbol{x}_0 - \boldsymbol{x}^*\\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
nonsmooth 	f convex $\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq M\\|\boldsymbol{x} - \boldsymbol{x}'\\|$	$\frac{CM\\|\boldsymbol{x}_0 - \boldsymbol{x}^*\\|}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

For $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \leq \varepsilon$, need $k = O(\varepsilon^{-2})$ iter. for worst **nonsmooth** f

Can we exploit special structure of $\|\cdot\|_1, \|\cdot\|_*$ to get accuracy comparable to gradient descent (for smooth functions) ?

What does gradient descent do anyway?

Consider $\min\ f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

What does gradient descent do anyway?

Consider $\min\ f(\boldsymbol{x}),$ with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \doteq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2$$

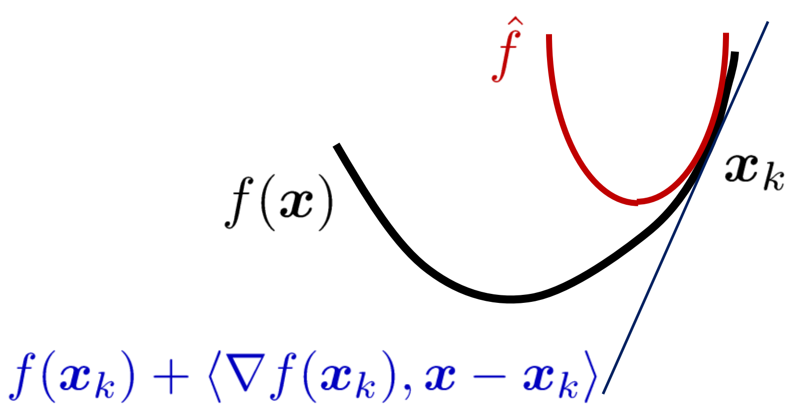
What does gradient descent do anyway?

Consider $\min f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) \doteq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2$$



What does gradient descent do anyway?

Consider $\min \; f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\begin{aligned}\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\ &= \frac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + \underline{\varphi(\boldsymbol{x}_k)}.\end{aligned}$$

Doesn't depend on $\boldsymbol{x}$

What does gradient descent do anyway?

Consider $\min f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\begin{aligned}\hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\ &= \frac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + \varphi(\boldsymbol{x}_k).\end{aligned}$$

Key observation: $\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{x}_k)$.

At each iteration, the gradient descent minimizes a (separable) quadratic approximation to the objective function, formed at $\boldsymbol{x}_k$.

What does gradient descent do anyway?

Consider $\min \; f(\boldsymbol{x})$, with f convex, differentiable, and ∇f L-Lipschitz.

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)$

Quadratic approximation to f around $\boldsymbol{x}_k$:

$$\begin{aligned} \hat{f}(\boldsymbol{x}, \boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\ &= \frac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + \varphi(\boldsymbol{x}_k). \end{aligned}$$

Key observation: $\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{x}_k)$.

At each iteration, the gradient descent minimizes a (separable) quadratic approximation to the objective function, formed at $\boldsymbol{x}_k$.

Rate for gradient descent: $f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \;\leq\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k} \;=\; O\left(\frac{1}{k}\right)$

Borrowing the approximation idea…

$$\min \ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1$$

Borrowing the approximation idea…

$$\min \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1$$

smooth  *nonsmooth* 

Borrowing the approximation idea…

$$\min \ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad f(\boldsymbol{x}) \ + \ g(\boldsymbol{x})$$

smooth *nonsmooth*

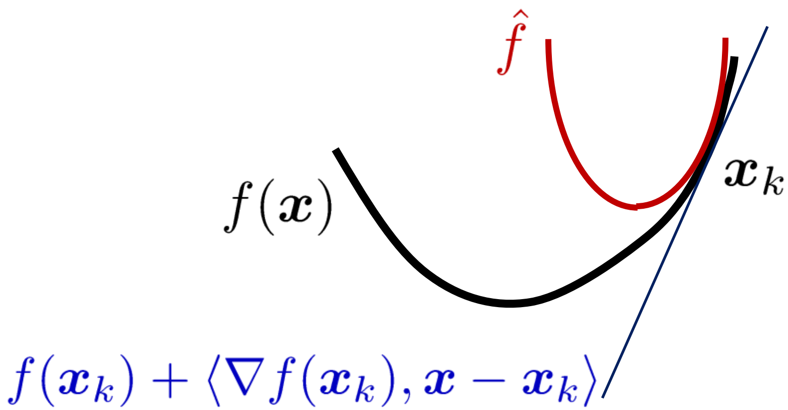
Borrowing the approximation idea…

$$\min \; \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad f(\boldsymbol{x}) \; + \; g(\boldsymbol{x})$$

smooth *nonsmooth*

Just **approximate the smooth part:**

$$\hat{F}(\boldsymbol{x}, \boldsymbol{x}_k) \quad \doteq \quad f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \tfrac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 + g(\boldsymbol{x})$$



Borrowing the approximation idea…

$$\min \ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad f(\boldsymbol{x}) \ + \ g(\boldsymbol{x})$$

smooth *nonsmooth*

Just **approximate the smooth part:**

$$\begin{aligned}\hat{F}(\boldsymbol{x},\boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k)+\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}-\boldsymbol{x}_k\rangle+\tfrac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2+g(\boldsymbol{x})\\ &= \tfrac{L}{2}\|\boldsymbol{x}-(\boldsymbol{x}_k-\tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2+g(\boldsymbol{x})+\varphi(\boldsymbol{x}_k).\end{aligned}$$

Borrowing the approximation idea…

$$\min\ \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \quad + \quad \lambda\|\boldsymbol{x}\|_1 \qquad \equiv \qquad \min \quad f(\boldsymbol{x}) \ + \ g(\boldsymbol{x})$$

smooth *nonsmooth*

Just **approximate the smooth part:**

$$\begin{aligned}\hat{F}(\boldsymbol{x},\boldsymbol{x}_k) &\doteq f(\boldsymbol{x}_k)+\langle\nabla f(\boldsymbol{x}_k),\boldsymbol{x}-\boldsymbol{x}_k\rangle+\tfrac{L}{2}\|\boldsymbol{x}-\boldsymbol{x}_k\|^2+g(\boldsymbol{x})\\ &= \tfrac{L}{2}\|\boldsymbol{x}-(\boldsymbol{x}_k-\tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2+g(\boldsymbol{x})+\varphi(\boldsymbol{x}_k).\end{aligned}$$

… and then **minimize to get the next iterate:**

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x},\boldsymbol{x}_k)\\ &= \arg\min_{\boldsymbol{x}}\ \tfrac{L}{2}\|\boldsymbol{x}-(\boldsymbol{x}_k-\tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2+g(\boldsymbol{x}).\end{aligned}$$

This is called a **proximal gradient algorithm**.

Proximal gradient algorithm

$$\min \; f(\boldsymbol{x}) + g(\boldsymbol{x}), \quad \text{with } f \text{ convex differentiable, } \nabla f \; L\text{-Lipschitz.}$$

Proximal Gradient:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \frac{L}{2} \|\boldsymbol{x} - (\boldsymbol{x}_k - \tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + g(\boldsymbol{x})$$

Converges at the **same rate as gradient descent**:

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*) \;\le\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k} \;=\; O\left(\frac{1}{k}\right)$$

Efficient whenever we can easily solve the **proximal problem**

$$\text{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

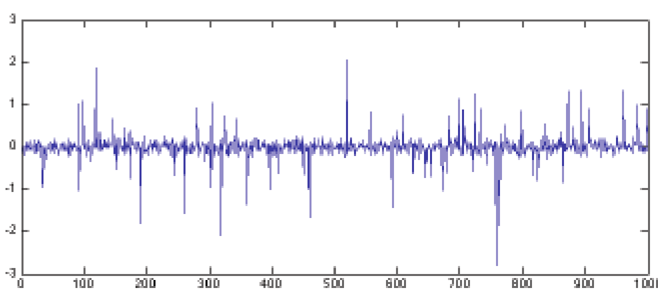
i.e., minimize g plus a separable quadratic.

Prox. operators for structure-inducing norms

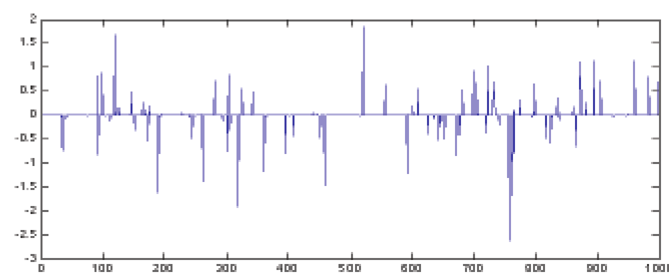
$$\text{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

For $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, $\text{prox}_{\mu g}(\boldsymbol{z})$ is given by **soft thresholding** the elements of $\boldsymbol{z}$: $\mathcal{S}_\mu(z) = \text{sign}(z)\max\{|z|-\mu, 0\}.$

This operator shrinks all of the elements of $\boldsymbol{z}$ towards zero:



$\boldsymbol{z}$



$\mathcal{S}_\mu(\boldsymbol{z})$

It can be computed in linear time (very efficient).

Prox. operators for structure-inducing norms

$$\text{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

For $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, $\text{prox}_{\mu g}(\boldsymbol{z})$ is given by **soft thresholding** the elements of $\boldsymbol{z}$: $\mathcal{S}_\mu(z) = \text{sign}(z)\max\{|z|-\mu, 0\}.$

For $g(\boldsymbol{X}) = \|\boldsymbol{X}\|_*$, $\text{prox}_{\mu g}(\boldsymbol{Z})$ is given by **soft thresholding** the **singular values** of $\boldsymbol{Z}$: for $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$,

$$\text{prox}_{\mu g}(\boldsymbol{Z}) \;=\; \boldsymbol{U}\mathcal{S}_\mu[\boldsymbol{\Sigma}]\boldsymbol{V}^*.$$

Again efficient (same cost as a singular value decomposition).

Similar expressions exist for other structure inducing norms.

Summing up: proximal gradient

$$\min \; f(\boldsymbol{x}) + g(\boldsymbol{x}), \quad \text{with } f \text{ convex differentiable, } \nabla f \; L\text{-Lipschitz.}$$

Proximal Gradient:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \tfrac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \tfrac{1}{L}\nabla f(\boldsymbol{x}_k))\|_2^2 + g(\boldsymbol{x})$$

Converges at the **same rate as gradient descent**:

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*) \;\leq\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k} \;=\; O\left(\tfrac{1}{k}\right)$$

Efficient whenever we can easily solve the **proximal problem**

$$\text{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

This is the case for many structure-inducing norms.

What have we accomplished so far?

Function class $\mathcal{F}$	$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth  f convex, differentiable $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
smooth + structured nonsmooth: $F = f + g$  +  f, g convex, $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k} = O\left(\frac{1}{k}\right)$
nonsmooth  f convex $\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq M\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CM\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

Still a gap between convergence rate of proximal gradient,
$O(1/k)$ ***and the optimal*** $O(1/k^2)$ ***rate for smooth*** f **.**

Can we close this gap?

Why is the gradient method suboptimal?

For smooth f, gradient descent is also suboptimal…
intuitively, for badly conditioned functions it may "chatter":

Gradient descent

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)$$

Why is the gradient method suboptimal?

For smooth f, gradient descent is also suboptimal…
intuitively, for badly conditioned functions it may "chatter":

Gradient descent

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k)$$

The *heavy ball method* treats the iterate as a point mass with momentum,
and hence, a tendency to continue moving in direction $\boldsymbol{x}_k - \boldsymbol{x}_{k-1}$:

Heavy ball

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k) + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

Nesterov's optimal method

Shares some intuition with heavy ball, but not identical.

Heavy ball : $$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k) + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

Nesterov : $$\boldsymbol{y}_k = \boldsymbol{x}_k + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{y}_k - \alpha \nabla f(\boldsymbol{y}_k)$$

with a very special choice of β_k to ensure the optimal rate:

$$\beta_k = \frac{t_k - 1}{t_{k+1}} \qquad t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \qquad \alpha = 1/L$$

Theorem 6 (Nesterov '83) *Let f be a convex function with L-Lipschitz gradient. The accelerated gradient algorithm achieves*

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \;\leq\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2}{(k+1)^2}. \tag{1}$$

This is optimal up to constants.

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \; + \; \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \tfrac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} \; + \; \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \tfrac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\boldsymbol{x}_{k+1} \quad = \quad \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k)$$

What about smooth + nonsmooth?

$$\min \quad \underbrace{f(\boldsymbol{x})}_{\textit{smooth}} + \underbrace{g(\boldsymbol{x})}_{\textit{nonsmooth}}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \doteq f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \\ &= \arg\min_{\boldsymbol{x}} \frac{1}{L} \|\boldsymbol{x} - (\boldsymbol{y}_k - \frac{1}{L} \nabla f(\boldsymbol{y}_k))\|^2 + g(\boldsymbol{x}) \end{aligned}$$

What about smooth + nonsmooth?

$$\min \quad \underset{\textit{smooth}}{f(\boldsymbol{x})} \; + \; \underset{\textit{nonsmooth}}{g(\boldsymbol{x})}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \doteq f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \\ &= \arg\min_{\boldsymbol{x}} \frac{1}{L} \|\boldsymbol{x} - (\boldsymbol{y}_k - \frac{1}{L} \nabla f(\boldsymbol{y}_k))\|^2 + g(\boldsymbol{x}) \\ &= \operatorname{prox}_{L^{-1} g}(\boldsymbol{y}_k - \frac{1}{L} \nabla f(\boldsymbol{y}_k)). \end{aligned}$$

What about smooth + nonsmooth?

$$\min \quad \underset{\textit{smooth}}{f(\boldsymbol{x})} \; + \; \underset{\textit{nonsmooth}}{g(\boldsymbol{x})}$$

Again form a separable quadratic upper bound, but **now at** $\boldsymbol{y}_k$:

$$\hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \quad \doteq \quad f(\boldsymbol{y}_k) + \langle \nabla f(\boldsymbol{y}_k), \boldsymbol{x} - \boldsymbol{y}_k \rangle + \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{y}_k\|^2 + g(\boldsymbol{x})$$

Again, **replace the gradient step** with minimization of the upper bound:

$$\begin{aligned} \boldsymbol{x}_{k+1} \quad &= \quad \arg\min_{\boldsymbol{x}} \hat{F}(\boldsymbol{x}, \boldsymbol{y}_k) \\ &= \quad \arg\min_{\boldsymbol{x}} \frac{1}{L} \|\boldsymbol{x} - (\boldsymbol{y}_k - \frac{1}{L} \nabla f(\boldsymbol{y}_k))\|^2 + g(\boldsymbol{x}) \\ &= \quad \mathrm{prox}_{L^{-1} g}(\boldsymbol{y}_k - \frac{1}{L} \nabla f(\boldsymbol{y}_k)). \end{aligned}$$

Making the **same special choice** $\boldsymbol{y}_k = \boldsymbol{x}_k + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$, we obtain an ***accelerated* proximal gradient** algorithm.

Accelerated proximal gradient algorithm

$$\min \; f(\boldsymbol{x}) + g(\boldsymbol{x}),$$ with f convex, differentiable, ∇f L-Lipschitz.

Accelerated Proximal Gradient:

$$\text{Repeat} \begin{cases} \boldsymbol{y}_k = \boldsymbol{x}_k + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\ \boldsymbol{x}_{k+1} = \text{prox}_{L^{-1}g}(\boldsymbol{y}_k - \frac{1}{L}\nabla f(\boldsymbol{y}_k)) \end{cases}$$

with $\beta_k = \frac{t_k - 1}{t_{k+1}}$ and $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$.

Converges at the **same rate as Nesterov's optimal gradient method**:

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}^*) \;\leq\; \frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{(k+1)^2} \;=\; O\left(\frac{1}{k^2}\right)$$

Again, efficient whenever we can easily solve the **proximal problem**

$$\text{prox}_{\mu g}(\boldsymbol{z}) \;=\; \arg\min_{\boldsymbol{x}} \; \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu g(\boldsymbol{x})$$

What have we accomplished so far?

Function class $\mathcal{F}$	$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)$
smooth  f convex, differentiable $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
smooth + structured nonsmooth: $F = f + g$  f, g convex, $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CL\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{k^2} = \Theta\left(\frac{1}{k^2}\right)$
nonsmooth  f convex $\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq M\|\boldsymbol{x} - \boldsymbol{x}'\|$	$\frac{CM\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|}{\sqrt{k}} = \Theta\left(\frac{1}{\sqrt{k}}\right)$

For composite functions $F = f + g$, with f smooth,
***if g has an efficient proximal operator**, we achieve*
the same (optimal) rate as if F was smooth.

What about constraints?

Consider the **equality constrained** problem

$$\min \ \|\boldsymbol{x}\|_1 \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

Continuation: solve a sequence of unconstrained problems of form

$$\min \ \|\boldsymbol{x}\|_1 + \tfrac{\mu}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2,$$

with $\mu \nearrow \infty$. Solutions converge to the solution to $(*)$.

Big downside: **conditioning**. For $f(\boldsymbol{x}) = \frac{\mu}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$, the gradient is L-Lipschitz, with $L = \mu\|\boldsymbol{A}^*\boldsymbol{A}\|$. As $\mu \nearrow \infty$, the unconstrained problems get harder and harder to solve.

Is there a better-structured way to enforce equality constraints?

The method of multipliers

$$\min \; F(\boldsymbol{x}) \;\; \text{s.t.} \;\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

The **Lagrangian** is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \; = \; F(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle$$

The method of multipliers

$$\min \; F(\boldsymbol{x}) \;\; \text{s.t.} \;\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

The **augmented Lagrangian** is

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) \; = \; F(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle + \underline{\tfrac{\rho}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \|_2^2}.$$

Extra penalty term

The method of multipliers

$$\min\ F(\boldsymbol{x})\ \text{ s.t. }\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

The **augmented Lagrangian** is

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) \ = \ F(\boldsymbol{x}) \ + \ \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle \ + \ \tfrac{\rho}{2} \| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \|_2^2.$$

The **method of multipliers** solves $(*)$ by seeking a saddle point of $\mathcal{L}_\rho$:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}). \end{aligned}$$

The method of multipliers

$$\min\ F(\boldsymbol{x}) \ \text{ s.t. } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \qquad (*)$$

The **augmented Lagrangian** is

$$\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}) \ = \ F(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \rangle + \tfrac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

The **method of multipliers** solves $(*)$ by seeking a saddle point of $\mathcal{L}_\rho$:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}). \end{aligned}$$

Solves a **sequence of unconstrained problems**: $\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k)$

Penalty parameter $\rho > 0$ can be constant (**avoids ill-conditioning**) , or increasing for (faster convergence).

Summing up: Method of multipliers

Solves, e.g., $\min\ F(\boldsymbol{x})$ s.t. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$, with F convex, lsc.

Method of multipliers (augmented Lagrangian)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{\lambda}_k) \\ \boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{y}). \end{aligned}$$

Classical method [Hestenes '69, Powell '69], see also [Bertsekas '82].

Avoids conditioning problems with the continuation / penalty method.

Under very general conditions $\boldsymbol{\lambda}_k$ converges to a dual optimal point,

$$\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{y}\| \to 0, \text{ and } F(\boldsymbol{x}_k) \to \inf\{\, F(\boldsymbol{x}) \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \,\}\,.$$

[Rockafellar '73, Eckstein '12] .

What have we accomplished so far?

Consider the robust PCA problem

$$\min \; \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Augmented Lagrangian

$$\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$$

The **method of multipliers** is

$$\begin{aligned} (\boldsymbol{L}_{k+1}, \boldsymbol{S}_{k+1}) &= \arg\min_{\boldsymbol{L}, \boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}_k, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2 \\ \boldsymbol{\Lambda}_{k+1} &= \boldsymbol{\Lambda}_k + \rho(\boldsymbol{L}_k + \boldsymbol{S}_k - \boldsymbol{D}) \end{aligned}$$

Each iteration is a large nonsmooth optimization problem…

Is there special structure we can exploit to simplify the iterations?

Special structure: Separable objectives

$$\min \; \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to S is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \arg\min_{\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$$

Special structure: Separable objectives

$$\min \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to S is easy:

$$\begin{aligned} \arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) &= \arg\min_{\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2 \\ &= \arg\min_{\boldsymbol{S}} \lambda \|\boldsymbol{S}\|_1 + \frac{\rho}{2} \|\boldsymbol{S} - (\boldsymbol{D} - \boldsymbol{L} - \tfrac{1}{\rho}\boldsymbol{\Lambda})\|_F^2 + \varphi(\boldsymbol{L}, \boldsymbol{D}, \boldsymbol{\Lambda}) \end{aligned}$$

Special structure: Separable objectives

$$\min \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to S is easy:

$$\begin{aligned} \arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) &= \arg\min_{\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2} \|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2 \\ &= \arg\min_{\boldsymbol{S}} \lambda \|\boldsymbol{S}\|_1 + \frac{\rho}{2} \|\boldsymbol{S} - (\boldsymbol{D} - \boldsymbol{L} - \frac{1}{\rho} \boldsymbol{\Lambda})\|_F^2 + \varphi(\boldsymbol{L}, \boldsymbol{D}, \boldsymbol{\Lambda}) \\ &= \operatorname{prox}_{\lambda \rho^{-1} \|\cdot\|_1} (\boldsymbol{D} - \boldsymbol{L} - \rho^{-1} \boldsymbol{\Lambda}). \end{aligned}$$

Special structure: Separable objectives

$$\min \; \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2}\|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to S is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D} - \boldsymbol{L} - \rho^{-1}\boldsymbol{\Lambda}).$$

Special structure: Separable objectives

$$\min \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\rangle + \frac{\rho}{2}\|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{S}$ is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D} - \boldsymbol{L} - \rho^{-1}\boldsymbol{\Lambda}).$$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{L}$ is also easy:

$$\arg\min_{\boldsymbol{L}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\rho^{-1}\|\cdot\|_*}(\boldsymbol{D} - \boldsymbol{S} - \rho^{-1}\boldsymbol{\Lambda}).$$

Special structure: Separable objectives

$$\min \ \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \quad \text{s.t.} \quad \boldsymbol{L} + \boldsymbol{S} = \boldsymbol{D}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D} \rangle + \frac{\rho}{2}\|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{S}$ is easy:

$$\arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D} - \boldsymbol{L} - \rho^{-1}\boldsymbol{\Lambda}).$$

Minimizing $\mathcal{L}_\rho$ with respect to $\boldsymbol{L}$ is also easy:

$$\arg\min_{\boldsymbol{L}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \quad = \quad \operatorname{prox}_{\rho^{-1}\|\cdot\|_*}(\boldsymbol{D} - \boldsymbol{S} - \rho^{-1}\boldsymbol{\Lambda}).$$

Why not just alternate?

$$\begin{aligned}
\boldsymbol{L}_{k+1} &= \arg\min_{\boldsymbol{L}} \mathcal{L}_\rho(\boldsymbol{L}, \boldsymbol{S}_k, \boldsymbol{\Lambda}_k) &&= \operatorname{prox}_{\rho^{-1}\|\cdot\|_*}(\boldsymbol{D} - \boldsymbol{S}_k - \rho^{-1}\boldsymbol{\Lambda}_k). \\
\boldsymbol{S}_{k+1} &= \arg\min_{\boldsymbol{S}} \mathcal{L}_\rho(\boldsymbol{L}_{k+1}, \boldsymbol{S}, \boldsymbol{\Lambda}_k) &&= \operatorname{prox}_{\lambda\rho^{-1}\|\cdot\|_1}(\boldsymbol{D} - \boldsymbol{L}_{k+1} - \rho^{-1}\boldsymbol{\Lambda}_k). \\
\boldsymbol{\Lambda}_{k+1} &= \boldsymbol{\Lambda}_k + \rho(\boldsymbol{L}_{k+1} + \boldsymbol{S}_{k+1} - \boldsymbol{D})
\end{aligned}$$

More generally: Alternating Directions MoM

$$\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y}\|_F^2$

Alternating Directions Method of Multipliers (ADMM)

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{\lambda}_k) \\
\boldsymbol{z}_{k+1} &= \arg\min_{\boldsymbol{z}} \mathcal{L}_\rho(\boldsymbol{x}_{k+1}, \boldsymbol{z}, \boldsymbol{\lambda}_k) \\
\boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y})
\end{aligned}$$

Alternating Directions MoM

$$\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2} \| \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \|_F^2$

Alternating Directions Method of Multipliers (ADMM)

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{\lambda}_k) \\
\boldsymbol{z}_{k+1} &= \arg\min_{\boldsymbol{z}} \mathcal{L}_\rho(\boldsymbol{x}_{k+1}, \boldsymbol{z}, \boldsymbol{\lambda}_k) \\
\boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y})
\end{aligned}$$

Convergence: if f, h closed, proper, convex functions, and $\mathcal{L}$ has a saddle point, then … $\boldsymbol{\lambda}_k$ converges to a dual optimal point, $\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k \to \boldsymbol{y}$ and $f(\boldsymbol{x}_k) + h(\boldsymbol{z}_k) \to \inf\{ f(\boldsymbol{x}) + h(\boldsymbol{z}) \mid \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y} \}$.

Convergence rate $O(1/k)$, in a certain sense [He+Yuan '11].

Linearized Alternating Directions MoM

$$\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y}\|_F^2$

ADMM:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{\lambda}_k) \\ &= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \frac{1}{\rho}\boldsymbol{\lambda}_k\|_2^2 \end{aligned}$$

Complicated if $\boldsymbol{A}, \boldsymbol{B} \neq \boldsymbol{I}$

Linearized ADMM: just take a proximal gradient step…

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\rho}{2\tau}\|\boldsymbol{x} - (\boldsymbol{x}_k - \tau \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \frac{1}{\rho}\boldsymbol{\lambda}_k))\|_2^2 \\ &= \operatorname{prox}_{\frac{\tau}{\rho} f}(\boldsymbol{x}_k - \tau \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} - \frac{1}{\rho}\boldsymbol{\lambda}_k)) \end{aligned}$$

Much more efficient if f has a simple proximal operator.

Linearized Alternating Directions MoM

$$\min \; f(\boldsymbol{x}) + h(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{y}$$

Aug. Lagrangian: $\mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + h(\boldsymbol{z}) + \langle \boldsymbol{\lambda}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y} \rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{y}\|_F^2$

Linearized ADMM

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \operatorname{prox}_{\frac{\tau}{\rho} f}(\boldsymbol{x}_k - \tau \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \tfrac{1}{\rho}\boldsymbol{\lambda}_k)) \\
\boldsymbol{z}_{k+1} &= \operatorname{prox}_{\frac{\tau}{\rho} h}(\boldsymbol{z}_k - \tau \boldsymbol{B}^*(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_k - \boldsymbol{y} + \tfrac{1}{\rho}\boldsymbol{\lambda}_k)) \\
\boldsymbol{\lambda}_{k+1} &= \boldsymbol{\lambda}_k + \rho(\boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{z}_{k+1} - \boldsymbol{y})
\end{aligned}$$

See, e.g., [S. Ma 2012]. Convergent if $\tau < \min\{\|\boldsymbol{A}\|^2, \|\boldsymbol{B}\|^2\}$.

Handles problems with more than two terms, e.g., $\sum_i f_i(\boldsymbol{x}_i)$.

Now can take advantage of two types of special structure …
separability of the objective and *prox capability* of f, h.

Finally, what have we accomplished?

Time required to solve a 1,000 x 1,000 robust PCA problem:

Algorithm	Accuracy	Rank	$\|E\|_0$	# iterations	time (sec)
IT	5.99e-006	50	101,268	8,550	**119,370.3**
DUAL	8.65e-006	50	100,024	822	1,855.4
APG	5.85e-006	50	100,347	134	1,468.9
APG_P	5.91e-006	50	100,347	134	82.7
$EALM_P$	2.07e-007	50	100,014	34	37.5
$IALM_P$	3.83e-007	50	99,996	23	**11.8**

THIS LECTURE

Four orders of magnitude improvement, just by choosing the right algorithm to solve the convex program:

Proximal gradient $\Rightarrow$ Accelerated proximal gradient $\Rightarrow$ ALM $\Rightarrow$ ADMoM

Recap and Conclusions

Key challenges of **nonsmoothness** and **scale** can be mitigated by using **special structure** in sparse and low-rank optimization problems:

Efficient proximity operators $\Rightarrow$ *proximal gradient methods*

Separable objectives $\Rightarrow$ *alternating directions methods*

Efficient **moderate-accuracy solutions** for **very large problems**.

Special tricks can further improve specific cases (factorization for low-rank)

Techniques in this literature apply quite broadly.

Extremely useful tools for creative problem formulation / solution.

Fundamental **theory** guiding engineering **practice**:

What are the basic principles and limitations?
What specific structure in my problem can allow me to do better?

To read more…

Problem complexity and lower bounds:

Nesterov – Introductory Lectures on Convex Optimization: A Basic Course 2004
Nemirovsky – Problem Complexity and Method Efficiency in Convex Optimization

Proximal gradient methods:

Accelerated gradient methods:

Nesterov – A method of solving a convex programming problem with convergence rate O(1/k^2), 1983
Tseng – On Accelerated Proximal Gradient Methods for Convex-Concave Optimization, 2008
Beck+Teboulle – A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, 2009

Augmented Lagrangian:

Hestenes – Multiplier and gradient methods, 1969
Powell – A method for nonlinear constraints in minimization problems, 1969
Rockafellar – Augmented Lagrangians and the Proximal Point Algorithm in Convex Programming, 1973
Bertsekas – Constrained Optimization and Lagrange Multiplier Methods, 1982

Alternating directions:

Glowinski+Marocco – Sur l'approximation, par elements finis d'ordre un, et la resolution, par … 1975
Gabay+Mercier – A dual algorithm for the solution of nonlinear variational problems … 1976
Eckstein+Bertsekas – On the Douglas-Rachford splitting method and the proximal point … 1992
Boyd et. al. – Distributed optimization and statistical learning via the alternating directions … 2010
Eckstein – Augmented Lagrangian and Alternating Directions Methods for Convex Optimization 2012

Part III: Non-Convex Alternatives

Previous Strategy for Sparse Estimation

Replace ℓ_0 Norm with Convex ℓ_1 Norm

Ideal (noiseless) case:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t. } \mathbf{y} = \Phi\mathbf{x}$$

$$\|\mathbf{x}\|_1 = \sum_{i} |x_i|$$

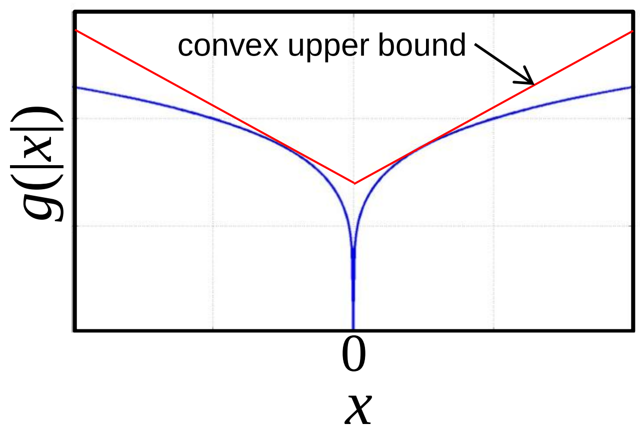
Relaxed case:

$$\min_{\mathbf{x}} \|\mathbf{y} - \Phi\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1$$

Non-Convexity via Iterative Reweighted ℓ_1

Non-convex penalty $g(|\mathbf{x}|)$

concave, non-decreasing



Updates:

$$\mathbf{x}^{(k+1)} \leftarrow \arg\min_{\mathbf{x}} \sum_i w_i^{(k)} |x_i| \quad \text{s.t. } \mathbf{y} = A\,\mathbf{x}$$

$$\mathbf{w}^{(k+1)} \leftarrow \left.\frac{\partial g(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=|\mathbf{x}^{(k+1)}|}$$

slope of convex upper bound

[Fazel et al., 2003]

Example

Penalty function:

$$g(|\mathbf{x}|) = \sum_i \log\left(|x_i| + \varepsilon\right), \quad \varepsilon > 0$$

Updates:

$$\mathbf{x}^{(k+1)} \leftarrow \arg\min_{\mathbf{x}} \sum_i w_i^{(k)} |x_i| \quad \text{s.t. } \mathbf{y} = \mathrm{A}\mathbf{x}$$

$$w_i^{(k+1)} \leftarrow \frac{1}{\left(|x_i^{(k+1)}| + \varepsilon\right)}$$

[Fazel et al., 2003; Candès et al., 2008]

Variational Bayes (VB) can provide even more robust alternative penalties with provable guarantees

[Bishop 2006; Wipf et al., 2011]

Why bother with non-convexity?

Three important (interrelated) cases:

1. **Scaling/Shrinkage Problem**: The ℓ_1 norm may over-shrink large magnitude coefficients.

2. **Correlation Problem**: The dictionary A has some correlated columns which disrupt ℓ_0-ℓ_1 equivalence.

3. **Extra Parameters**: There are additional parameters to estimate, potentially embedded in A.

Similar principles hold regarding robust PCA

Case 1: Scaling and Shrinkage Issues

- The ℓ_1 penalty favors both ***sparse*** and ***low-variance*** solutions:

$$\|\mathbf{x}\|_0 \Longleftrightarrow \|\mathbf{x}\|_1 \Longleftrightarrow \|\mathbf{x}\|_2$$

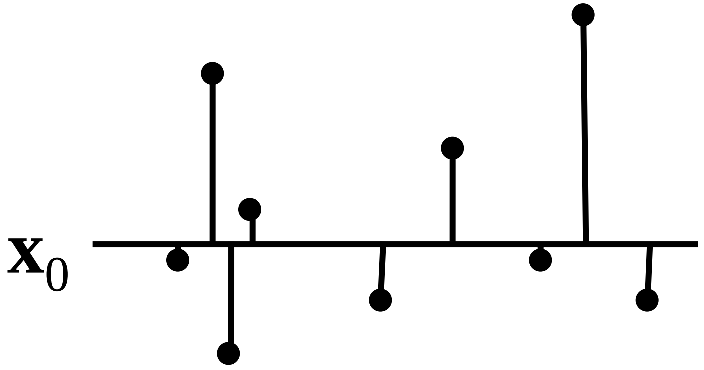
sparse — low variance

- Scale-sensitive ℓ_1 solutions may over-shrink large coefficients, possibly at the expense of sparsity.

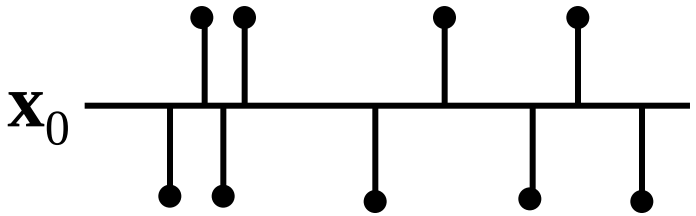
[Fan and Li, 2001; Levin et al., 2011]

Scaling Issues

- If the magnitudes of the non-zero elements in $\mathbf{x}_0$ are highly *scaled*, then the sparse recovery problem should be easier.



scaled coefficients (easy)

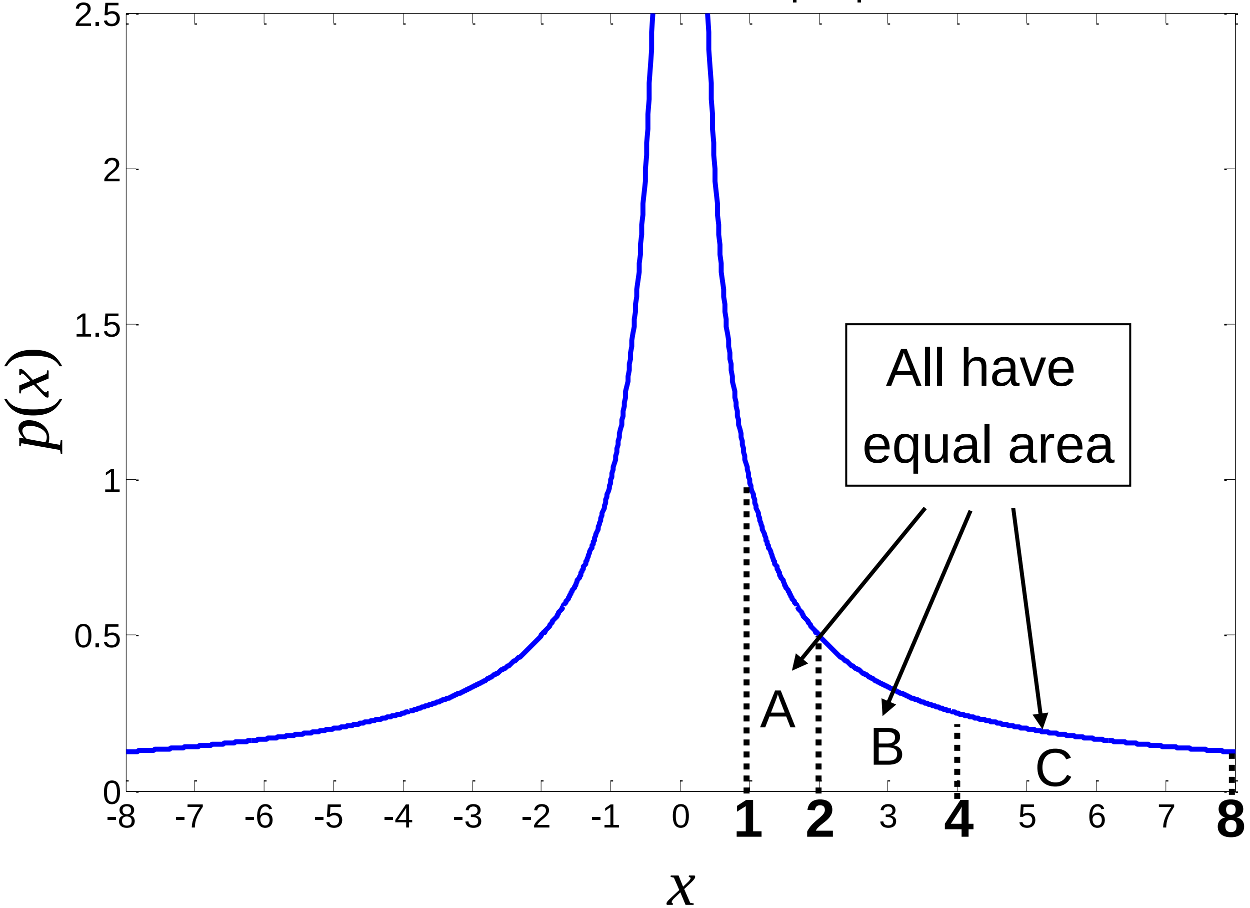


uniform coefficients (hard)

- The ℓ_1 solution may overly shrink large coefficients to achieve lower variance, and hence may not exploit the simpler scenario.

Extreme Case: Jeffreys Distribution

$$\text{Density:} \quad p(x) \propto \frac{1}{|x|}$$



Even a simple greedy estimation strategy should work well here

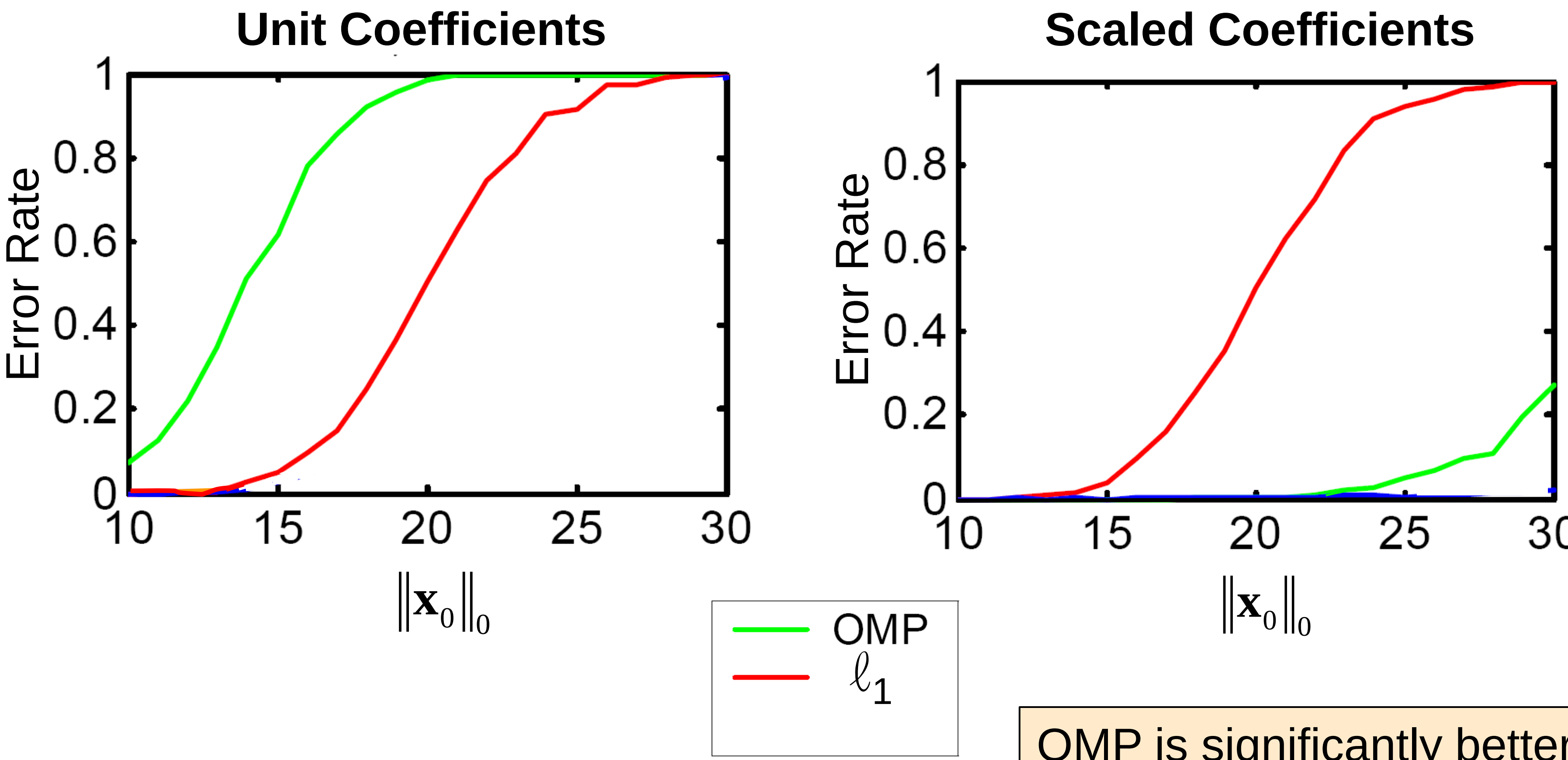
Simulation Example

- For each test case:

  1. Generate a random dictionary A with 50 rows and 100 columns.
  2. Generate a sparse coefficient vector $\mathbf{x}_0$.
  3. Compute signal via $\mathbf{y} = \mathrm{A}\,\mathbf{x}_0$.
  4. Run ℓ_1 and ***OMP*** (a very simple greedy strategy) to try and correctly estimate $\mathbf{x}_0$.
  5. Average over 1000 trials to compute empirical probability of failure.

- Repeat with different sparsity values, i.e., $\|\mathbf{x}_0\|_0$.

Results



OMP is significantly better!

Underlying Problem

Example:

$\Psi(u,v)$ = set of sparse vectors $\mathbf{x}_0$ with support pattern u and sign pattern v

$$\mathbf{x}_0 = \begin{bmatrix} 2.3 \\ 0 \\ -1.6 \\ 0 \end{bmatrix} \in \Psi\left(\{1,3\},\{+,-\}\right)$$

Theorem

If $$\arg\min_{\mathbf{x}:\mathbf{y}=A\,\mathbf{x}} \|\mathbf{x}\|_0 \neq \arg\min_{\mathbf{x}:\mathbf{y}=A\,\mathbf{x}} \|\mathbf{x}\|_1$$

for some $\mathbf{x}_0 \in \Psi(u,v)$, $\mathbf{y} = A\,\mathbf{x}_0$, then ℓ_1 fails for all elements in this set.

[Malioutov et al., 2004]

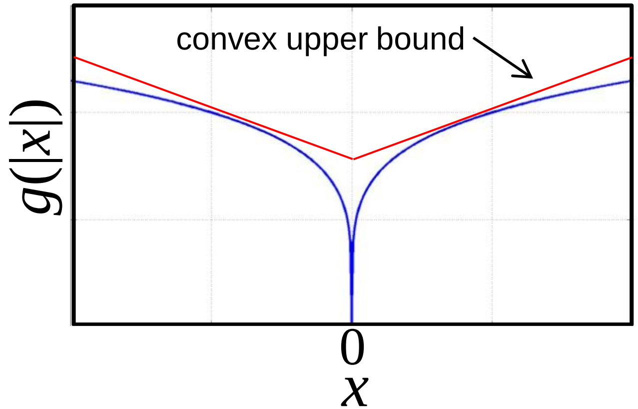
Always Room for Improvement

Theorem

In noiseless case, under mild conditions VB will:

1. Never do worse than the regular convex ℓ_1-norm solution.
2. For any A and $\Psi(u,v)$, there will ***always*** be cases where it performs better (… *helps with scaling/shrinkage issues*).

[Wipf, 2011]



With large coefficients, convex bound becomes flat ⇒ small penalty in next iteration

Simulation Example Revisited

- For each test case:

1. Generate a random dictionary Φ with 50 rows and 100 columns.

2. Generate a sparse coefficient vector $\mathbf{x}_0$.

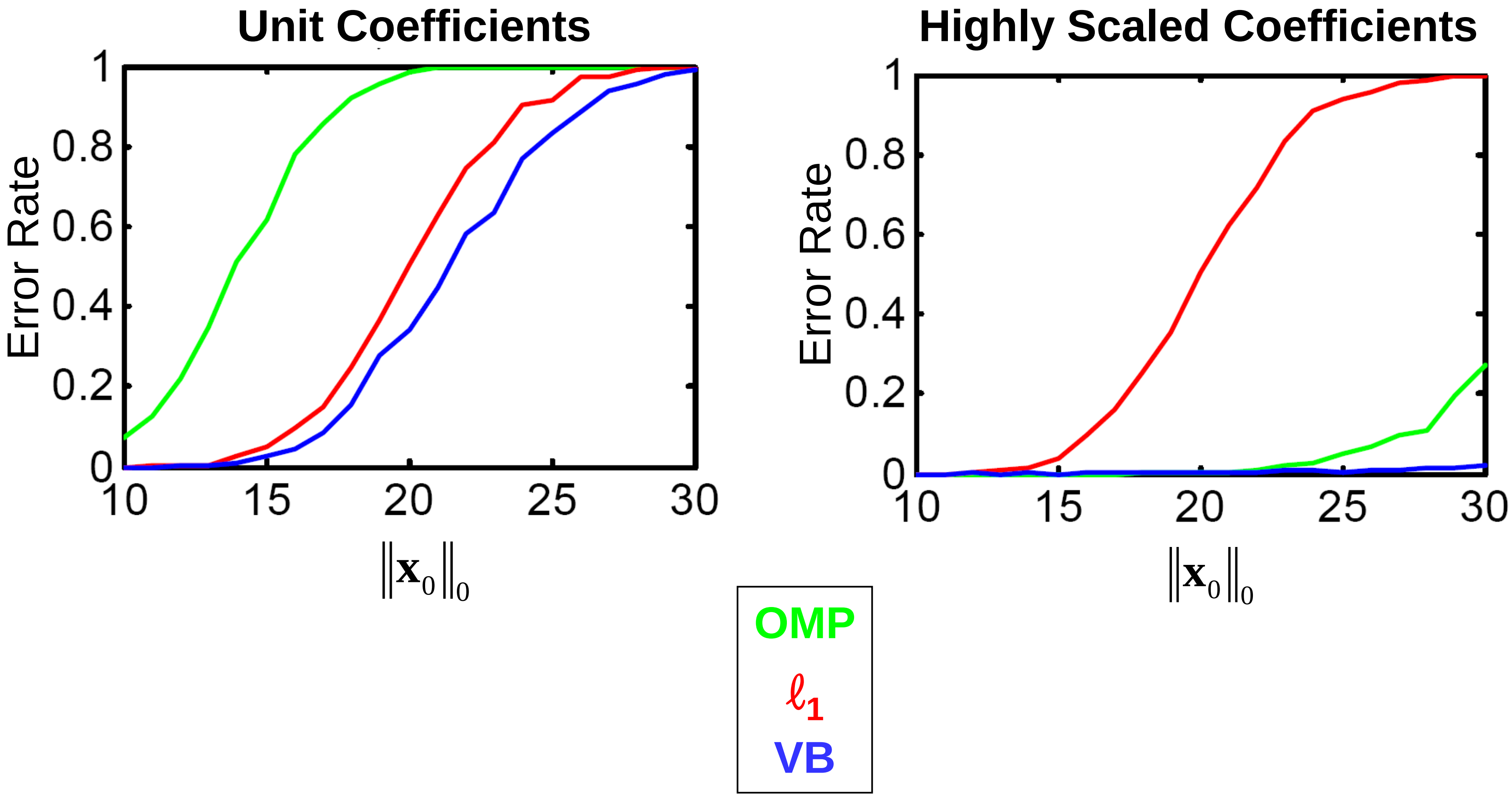
3. Compute signal via $\mathbf{y} = \mathrm{A}\,\mathbf{x}_0$.

4. Run **VB**, ℓ_1 and ***OMP*** (simple greedy strategy) to try and correctly estimate $\mathbf{x}_0$.

5. Average over 1000 trials to compute empirical probability of failure.

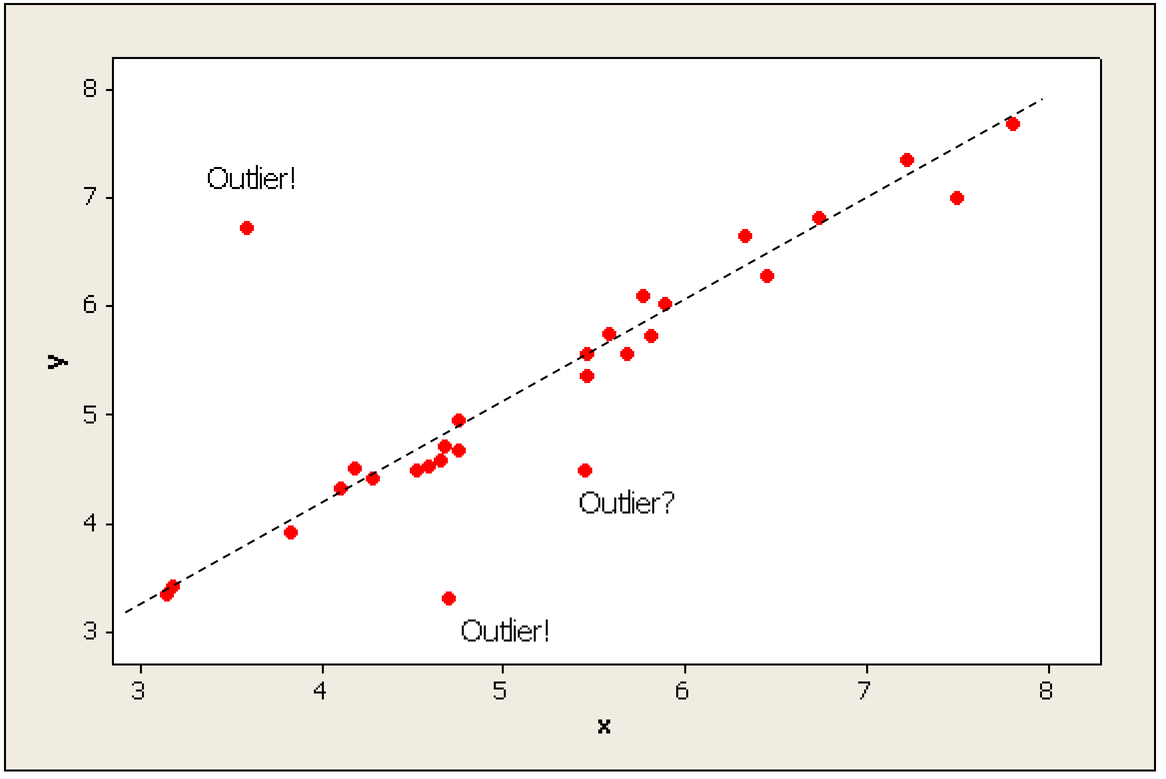
- Repeat with different sparsity values, i.e., $\|\mathbf{x}_0\|_0$.

Results



Unit Coefficients
Highly Scaled Coefficients
Error Rate
0
0.2
0.4
0.6
0.8
1
10
15
20
25
30
$\|\mathbf{x}_0\|_0$
OMP
ℓ_1
VB

Practical Example: Outlier Detection



Outlier Problem Cont.

- Linear generative model:

$$\mathbf{y} = A\,\mathbf{x} + \boldsymbol{\varepsilon}$$

m-dimensional observations

predictor variables

unknown coefficients, non-sparse

sparse noise

- **Objective**: Estimate $\mathbf{x}$ while rejecting outliers

Convert to Sparse Estimation Problem

$$\underbrace{\mathrm{Proj}_{Null\left[\mathrm{A}^T\right]}\left(\mathbf{y}\right)}_{\tilde{\mathbf{y}}} = \mathrm{Proj}_{Null\left[\mathrm{A}^T\right]}\left(\mathrm{A}\,\mathbf{x} + \boldsymbol{\varepsilon}\right) = \underbrace{\mathrm{Proj}_{Null\left[\mathrm{A}^T\right]}}_{\Phi}\left(\boldsymbol{\varepsilon}\right)$$

$$\min_{\boldsymbol{\varepsilon}} \left\|\boldsymbol{\varepsilon}\right\|_0 \quad \text{s.t. } \tilde{\mathbf{y}} = \Phi\boldsymbol{\varepsilon}$$

Once outliers are known, can estimate **x** via:

$$\hat{\mathbf{x}} = \left(\mathrm{A}^T\mathrm{A}\right)^{-1}\mathrm{A}^T\left(\mathbf{y} - \boldsymbol{\varepsilon}\right)$$

[Candès and Tao, 2004]

Practical Solutions

- But unknown outliers are likely unconstrained (different scales), and convex substitution may be suboptimal:

$$\min_{\boldsymbol{\varepsilon}} \|\boldsymbol{\varepsilon}\|_1 \quad \text{s.t.} \ \tilde{\mathbf{y}} = \Phi\boldsymbol{\varepsilon}$$

- Can instead use non-convex VB …

Practical Example:
Surface Normal Estimation via Photometric Stereo

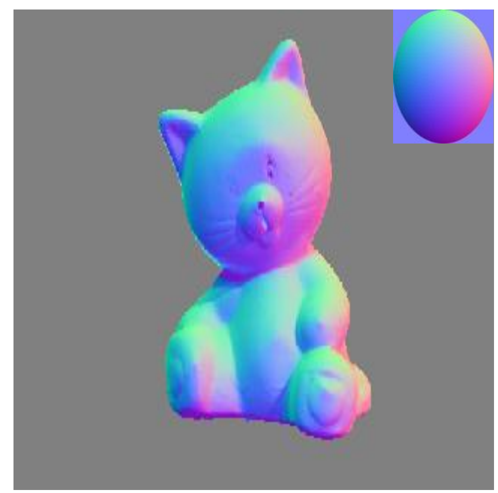
For basic Lambertian surface

[Woodham, 1980]

$$Y = \rho N \, L$$

Observations — Normal — Known Lighting

$$\rho N = Y L^{\dagger}$$



Surface Normal Map

Robust Surface Normal Estimation

- Basic Lambertian model ignores specular reflections, shadows, and other artifacts.

- Alternative per-pixel model:

$$\mathbf{y} = L\,\mathbf{n} + \boldsymbol{\varepsilon}$$

observations under different lightings

lighting matrix

raw unknown surface normal

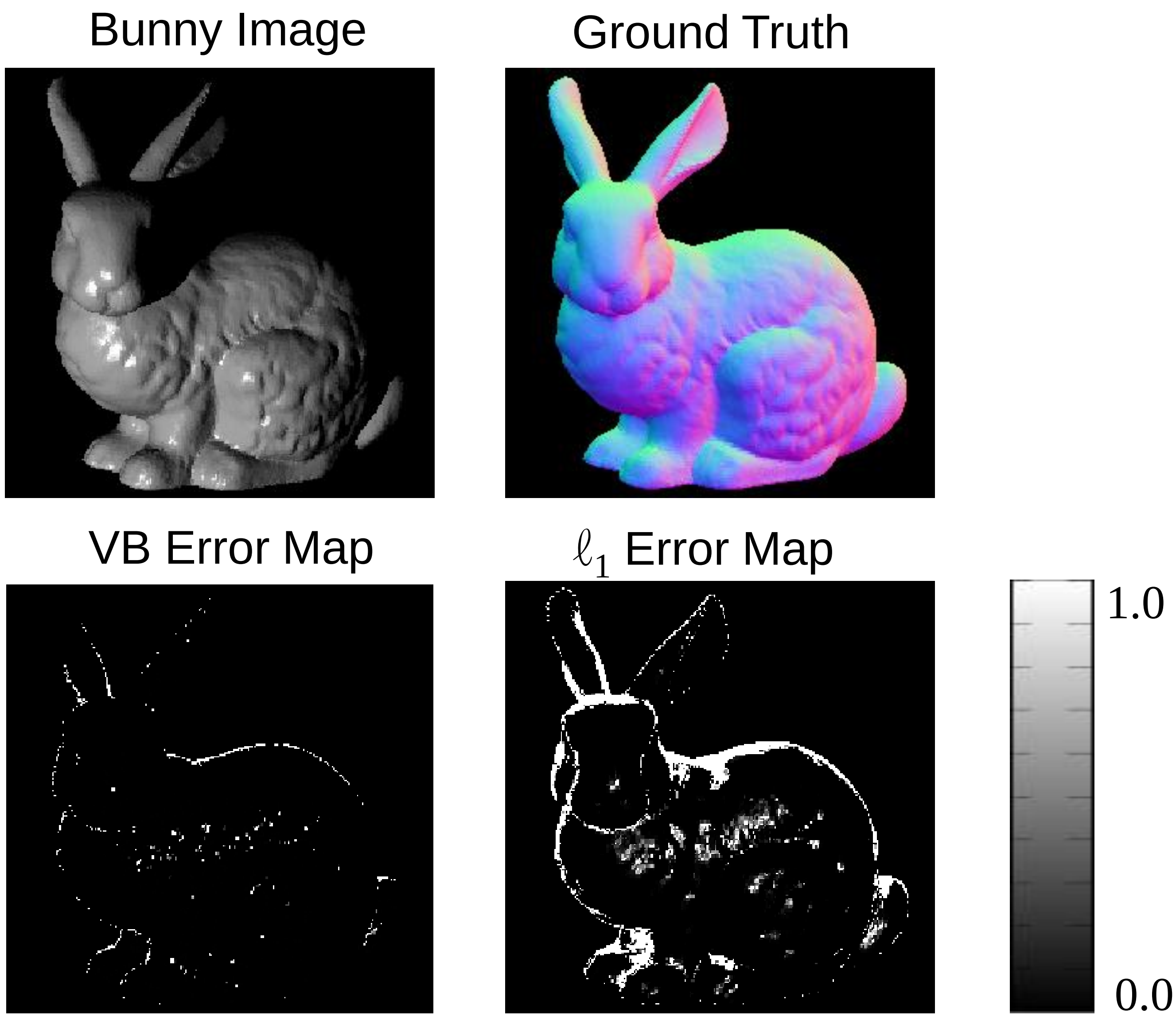
sparse errors

- Can also include a diffuse error term, and apply VB.

[Ikehata et al., 2012]

Results

[8.4% specular corruptions, 24% shadows]



[Ikehata et al., 2012]

Aggregate Results
[# of images varying]

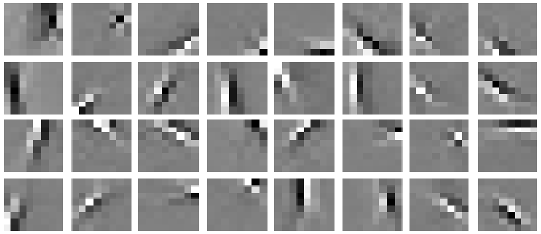
No. of images	Mean Error (deg.)	
	VB	ℓ_1
5	**5.2**	11.9
10	**2.8**	5.6
15	**1.9**	4.0
20	**1.2**	2.7
25	**0.81**	1.9
30	**0.62**	1.6
35	**0.59**	1.5
40	**0.53**	1.2

[Ikehata et al., 2012]

Case 2: Correlated Dictionaries

- Most theory applies to uncorrelated case, but many (most?) practical dictionaries have significant structure.

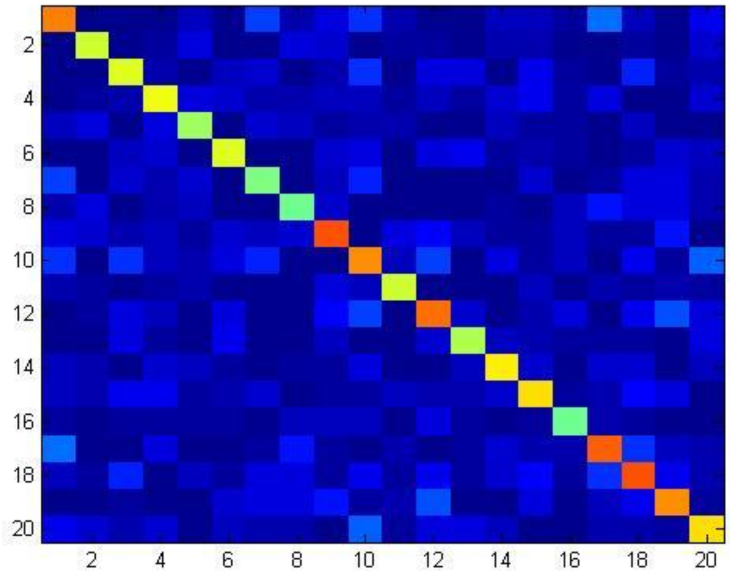
- **Examples**:



Dictionary Correlation Structure

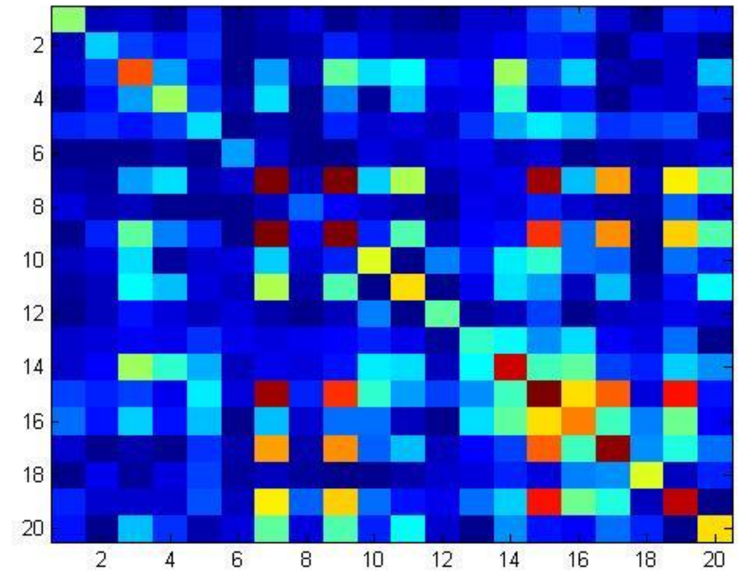
Low Correlation: Easy

$$A^T A$$



High Correlation: Hard

$$A^T A$$



Examples:

$A_{(uncor)} \sim$ iid $N(0,1)$ entries

$A_{(uncor)} \sim$ random rows of DFT

Example:

$$A_{(cor)} = \underbrace{\Psi}_{\text{arbitrary}} A_{(uncor)} \underbrace{\Phi}_{\text{block diagonal}}$$

How do we compensate for dictionary structure?

Simple Example:

Let vector α denote the column norms of A and define

$$g\left(|\mathbf{x}|;\alpha\right) = \sum_{i=1}^{n} \alpha_i^{-1} |x_i|$$

Then the problem

$$\min_{\mathbf{x}} \ \|\mathbf{y} - \mathrm{A}\,\mathbf{x}\|_2^2 + \lambda\, g\left(|\mathbf{x}|;\alpha\right)$$

is invariant to column norms.

So what about some function g that depends on the correlation structure $\mathrm{A}^T \mathrm{A}$

VB and Dictionary Correlations

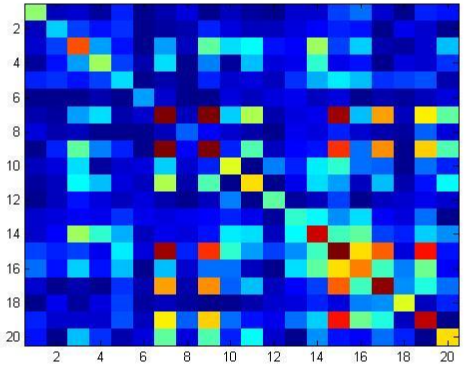
VB is equivalent to solving the penalized regression problem

$$\min_{\mathbf{x}} \ \|\mathbf{y} - \mathrm{A}\,\mathbf{x}\|_2^2 \ + \ \lambda\, g_{VB}\left(|\mathbf{x}|; \mathrm{A}^T \mathrm{A}\right)$$

for some function g_{VB} that favors a sparse $\mathbf{x}$.

[Palmer et al., 2006; Wipf et al., 2011]

$\mathrm{A}^T \mathrm{A}$



Notes on g_{VB} :

- Variables are penalized jointly based on the correlation structure of A.
- This allows VB to compensate for strong dictionary correlations.

Clustered Dictionary Model

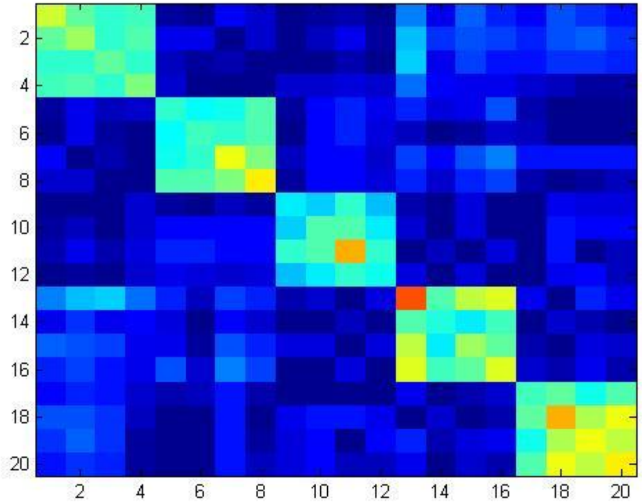
$A_{(uncor,k)}$ ⇒ any $m \times n$ dictionary such that ℓ_1 minimization succeeds for all $\|\mathbf{x}_0\|_0 \le k$

$A_{(cor,k)}$ ⇒ any dictionary obtained by replacing each column of $A_{(uncor,k)}$ with a "cluster" of n_i basis vectors within a radius ε

$\Omega_0 \subset \{1, 2, \ldots, n\}$ ⇒ (*cluster support*) set of cluster indeces whereby some $\mathbf{x}_0$ has at least one nonzero element.

Simple Clustered Example

$$\mathbf{A}^T_{(cor,k)}\mathbf{A}_{(cor,k)}$$



Problem:

- The ℓ_1 solution typically selects either zero or one basis vector from each cluster of correlated columns.
- While the 'cluster support' may be partially correct, the chosen basis vectors likely will not be.

VB and the Correlation Problem

Theorem

- Let $\mathbf{x}_0$ be a sparse signal.
- Under mild conditions, a minor variant of VB will recover $\mathbf{x}_0$ given any $\mathbf{y} = \mathrm{A}_{(cor,k)}\mathbf{x}_0$ provided

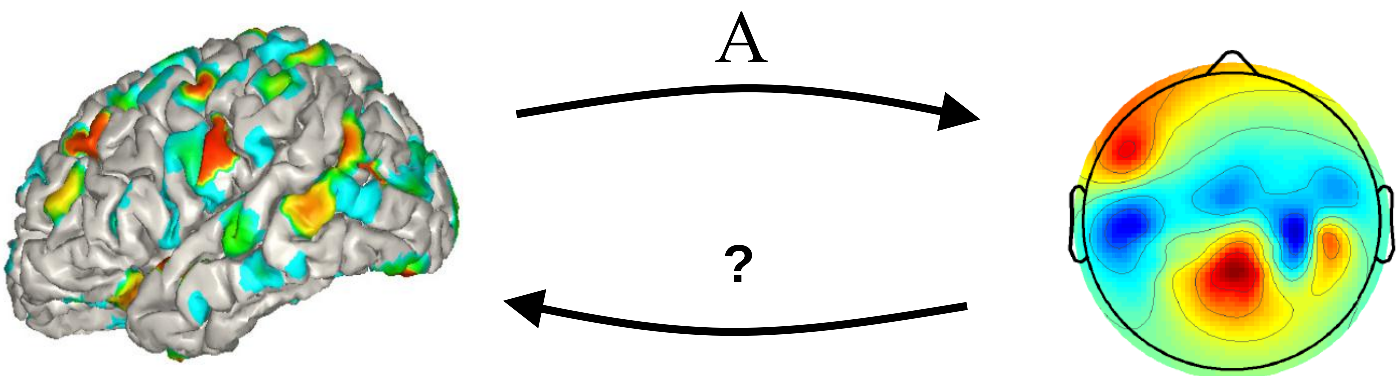
$$\left|\Omega_0\right| \leq k \quad \text{and} \quad \sum_{i\in\Omega_0} n_i \leq m$$

for some ε sufficiently small.

[Wipf and Wu, 2012]

Key Message: Non-convex algorithms can succeed even when strong correlations cause failure with ℓ_1

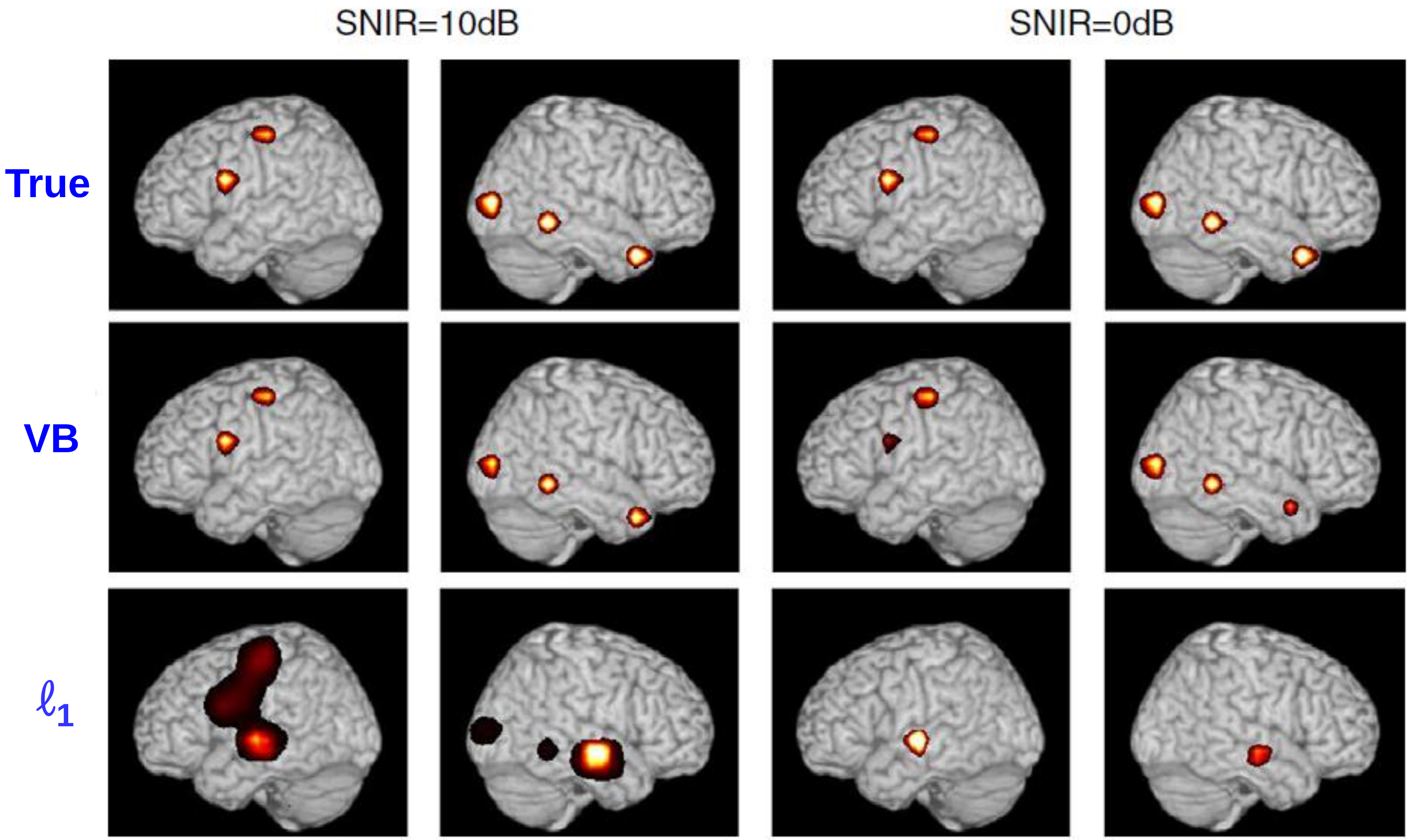
MEG/EEG Example



source space ($\mathbf{x}_0$) sensor space ($\mathbf{y}$)

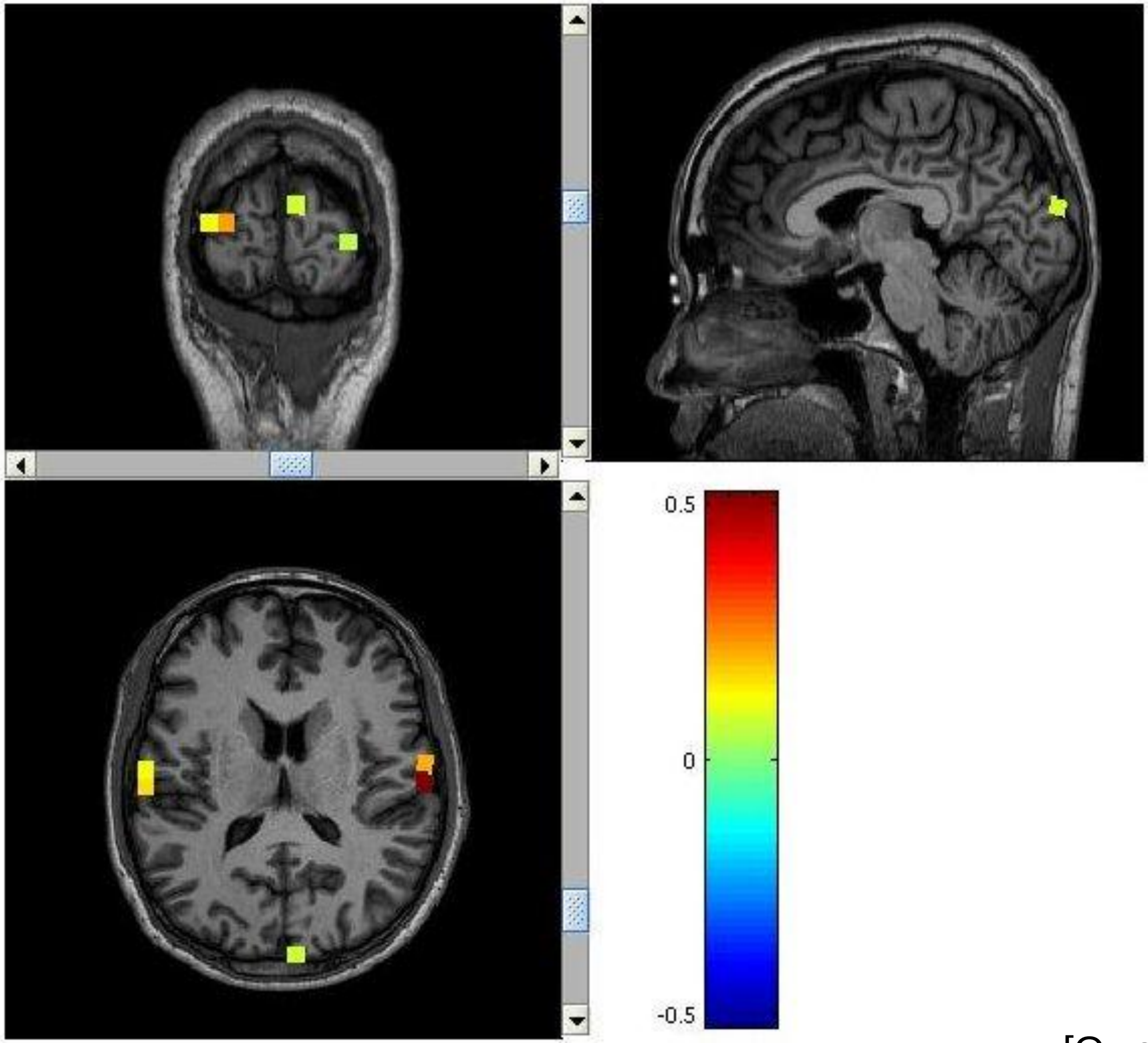
- Forward model dictionary A can be computed using Maxwell's equations [Sarvas,1987].
- Will be dependent on location of sensors, but always highly correlated by physical constraints.

Noisy Localization Results



[Owen et al., 2013]

Real Data



[Owen et al., 2013]

Remarks

- Non-convex VB algorithms implicitly employ a penalty that helps compensate for correlated dictionaries.

- MEG/EEG experiments show advantages of non-convexity when A is:

1. Highly underdetermined, e.g.,

$$m = 275 \quad \text{and} \quad n = 10^5$$

2. Very ill-conditioned and structured, i.e., columns/rows are highly correlated.

Case 3: Dictionary Has Embedded Parameters

- Ideal (noiseless) :

$$\min_{\mathbf{x},\mathbf{k}\in\Omega_k} \|\mathbf{x}\|_0 \quad \text{s.t.} \;\; \mathbf{y} = \mathrm{A}(\mathbf{k})\mathbf{x}$$

- Approximate version:

$$\min_{\mathbf{x},\mathbf{k}\in\Omega_k} \|\mathbf{y} - \mathrm{A}(\mathbf{k})\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0$$

- **Applications**: Bilinear models, blind deconvolution, blind image deblurring, etc.

[Fergus et al., 2006; Levin et al., 2011]

Example: Blind Deconvolution

- Observation model:

$$\mathbf{y} \;=\; \mathbf{k} * \mathbf{x} + \boldsymbol{\varepsilon} \;=\; \mathrm{A}(\mathbf{k})\mathbf{x} + \boldsymbol{\varepsilon}$$

convolution operator — toeplitz matrix

- Would like to estimate the unknown $\mathbf{x}$ blindly since $\mathbf{k}$ is also unknown.

- In many situations (e.g., image deblurring) unknown $\mathbf{x}$ is sparse.

Efficient Convex Substitution?

Solve:

$$\min_{\mathbf{x},\mathbf{k}\in\Omega_k} \|\mathbf{x}\|_1 \quad \text{s.t. } \mathbf{y} = \mathbf{k} * \mathbf{x}$$

$$\Omega_k = \left\{ \mathbf{k} : \sum_i k_i = 1, \;\; k_i \geq 0, \forall i \right\}$$

Problem:

$$\|\mathbf{y}\|_1 = \left\| \sum_t k_t \mathbf{x}_t \right\|_1 \leq \sum_t k_t \|\mathbf{x}_t\|_1 = \|\mathbf{x}\|_1 \qquad \forall \text{ feasible } \mathbf{k}, \mathbf{x}$$

translated signal

- A degenerate solution is favored:

$$\mathbf{k} = \delta, \quad \mathrm{A}(\mathbf{k}) = I$$

We can't use ℓ_1

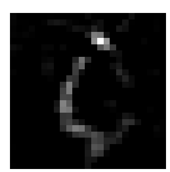
Practical Example: Blind Image Deblurring

- Basic convolution model (can be generalized):

$$\mathbf{y} = \mathbf{k} * \mathbf{x} + \boldsymbol{\varepsilon}$$

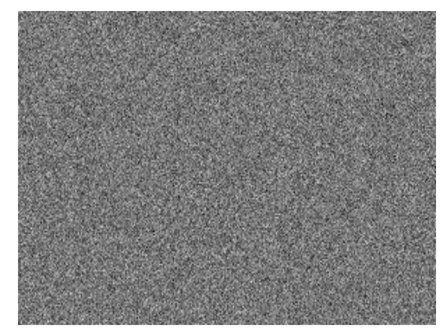
blurry image | blur kernel | sharp image

$=$



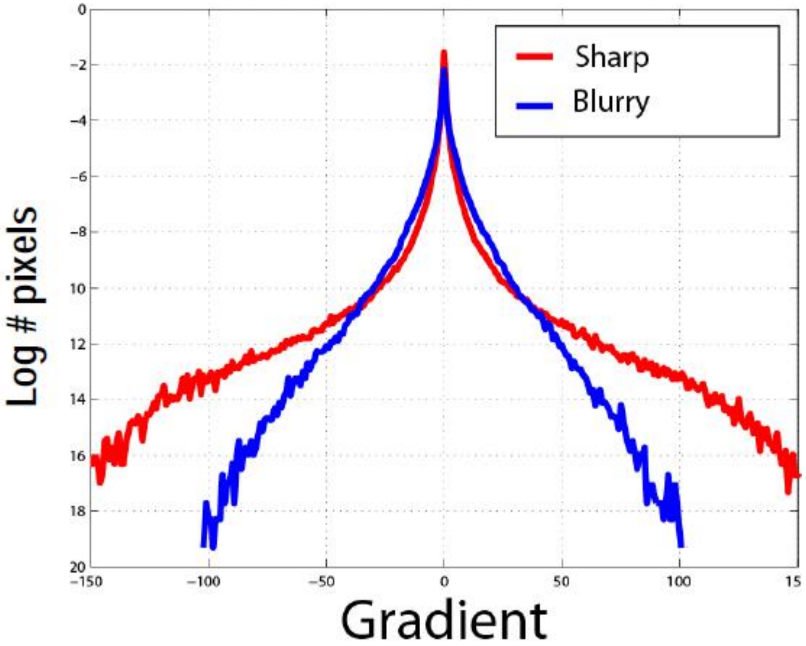
$*$

$+$



Unknown quantities we need to estimate

Gradients of Natural Images are Sparse



Can solve a modified sparse coding problem in gradient domain

$\mathbf{x}$: vectorized derivatives of the sharp image

$\mathbf{y}$: vectorized derivatives of the blurry image

Practical Blind Deblurring Algorithm

- A nearly ideal cost function for blind deblurring is

$$\min_{\mathbf{x},\mathbf{k}\in\Omega_k} \|\mathbf{y}-\mathbf{k}*\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0$$

$$\Omega_k = \left\{\mathbf{k} : \sum_i k_i = 1, \quad k_i \geq 0, \forall i\right\}$$

- But local minima are a huge problem, and convex relaxation provably fails …

- However, can leverage a principled *non-convex* VB substitution:

$$\min_{\mathbf{x},\mathbf{k}\in\Omega_k} \|\mathbf{y}-\mathbf{k}*\mathbf{x}\|_2^2 \; + \; \lambda \, g_{\mathrm{VB}}(\mathbf{x},\mathbf{k})$$

$$g_{VB}(\mathbf{x},\mathbf{k}) \neq g_x(\mathbf{x}) + g_k(\mathbf{k})$$

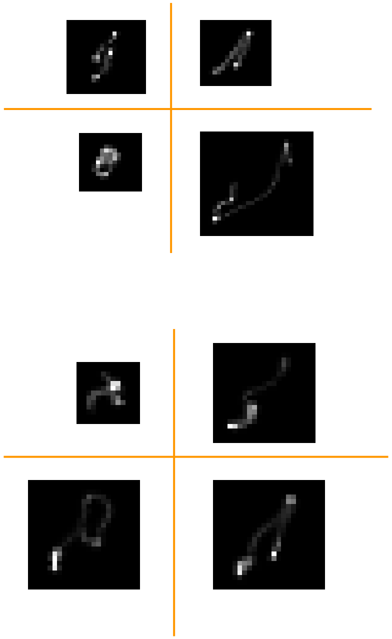
[Zhang and Wipf, 2013]

Blind Deblurring Evaluation Dataset

Levin et al. dataset [CVPR, 2009]

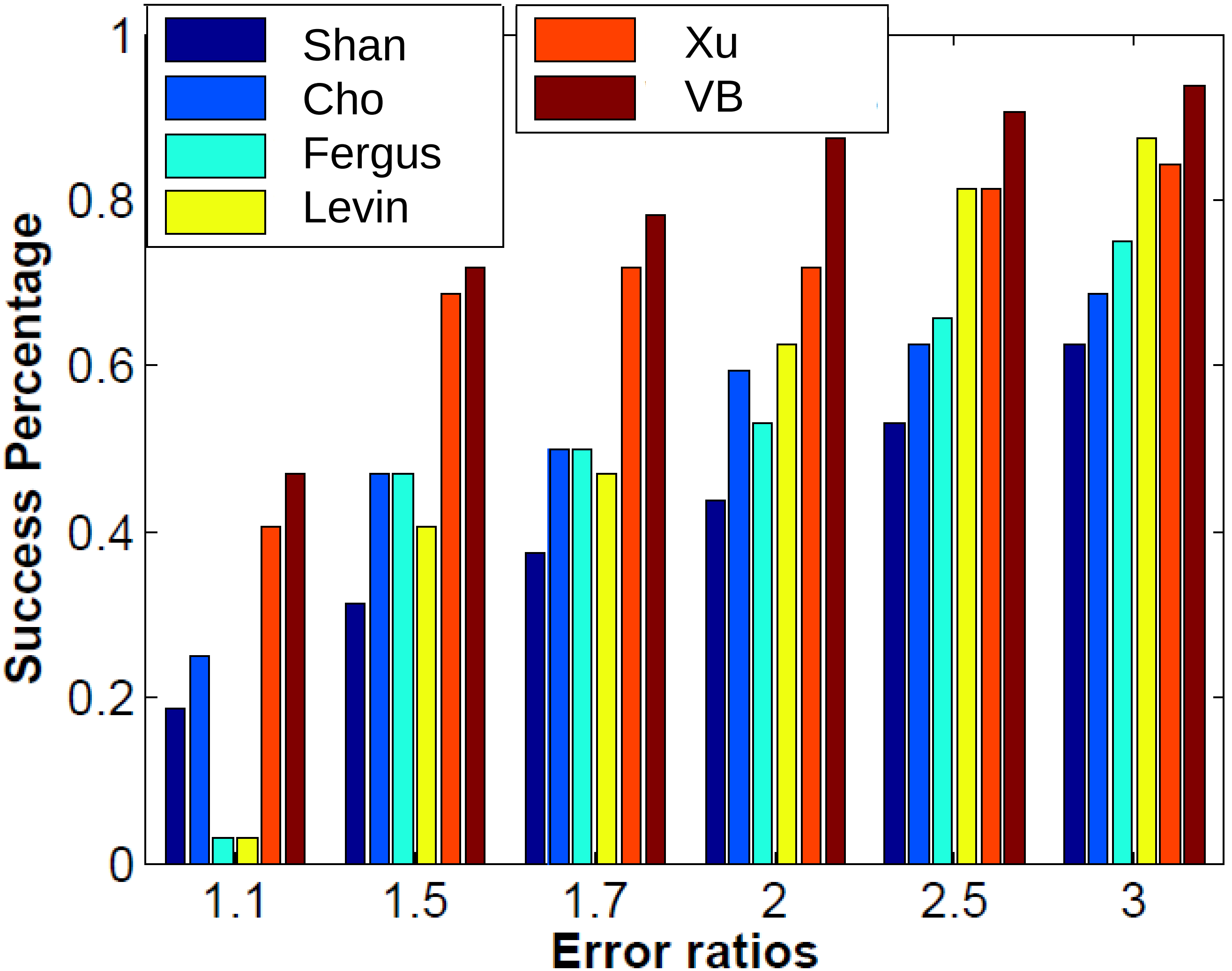
- 4 images of size 255 × 255 and 8 different empirically measured ground-truth blur kernels, giving 32 total blurry images

Images



Blur Kernels

Estimation Results

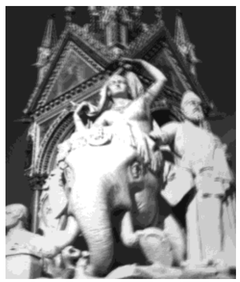


Note: All of these competing methods require considerable heuristics and tuning parameters

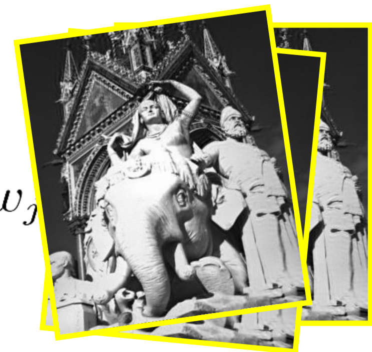
Extensions

Can easily adapt our model to more general scenarios:

1. Non-uniform convolution models



$$= \sum_j w_j$$

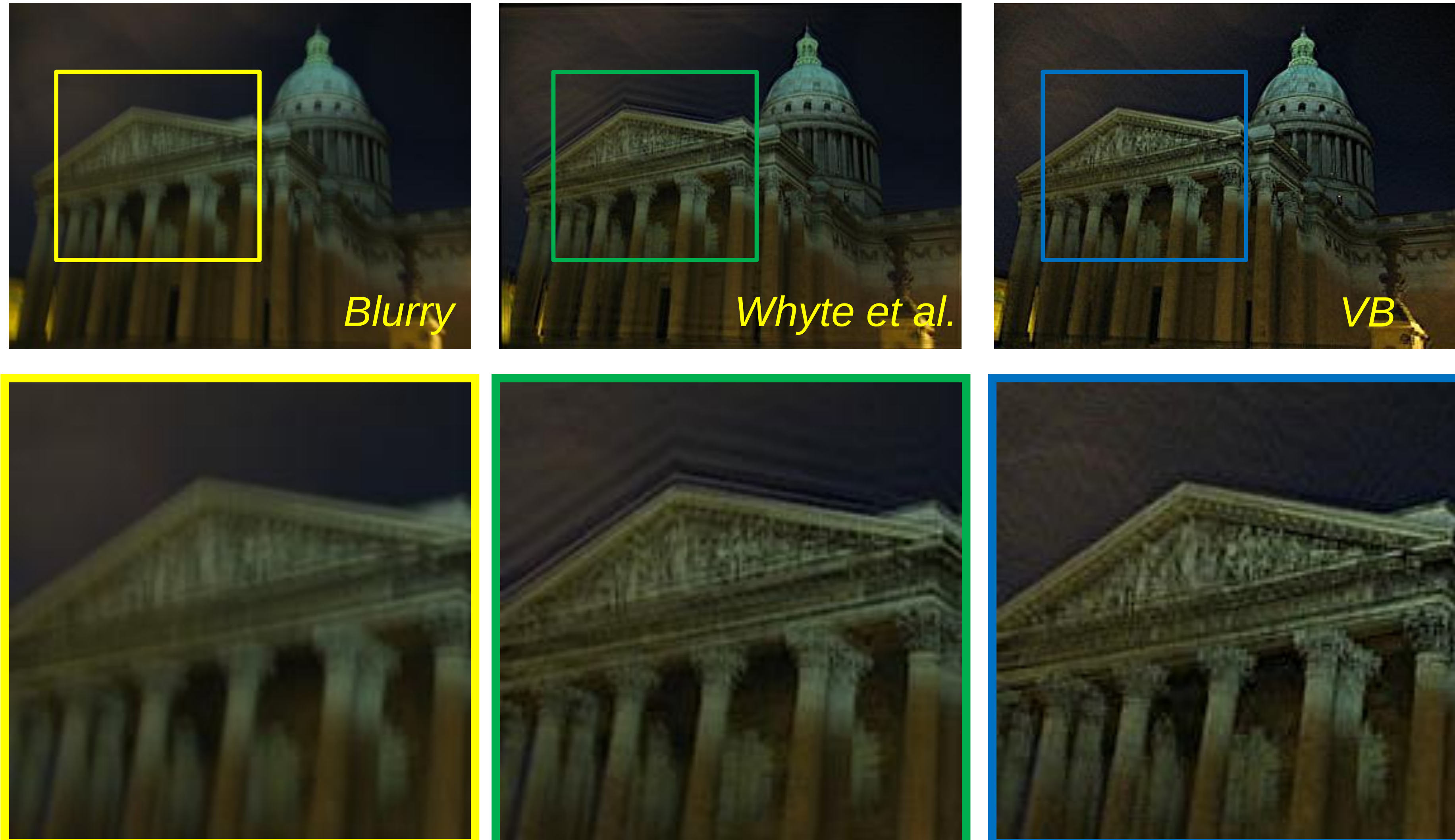


Blurry image is a superposition of translated and rotated sharp images

2. Multiple images for simultaneous denoising and deblurring

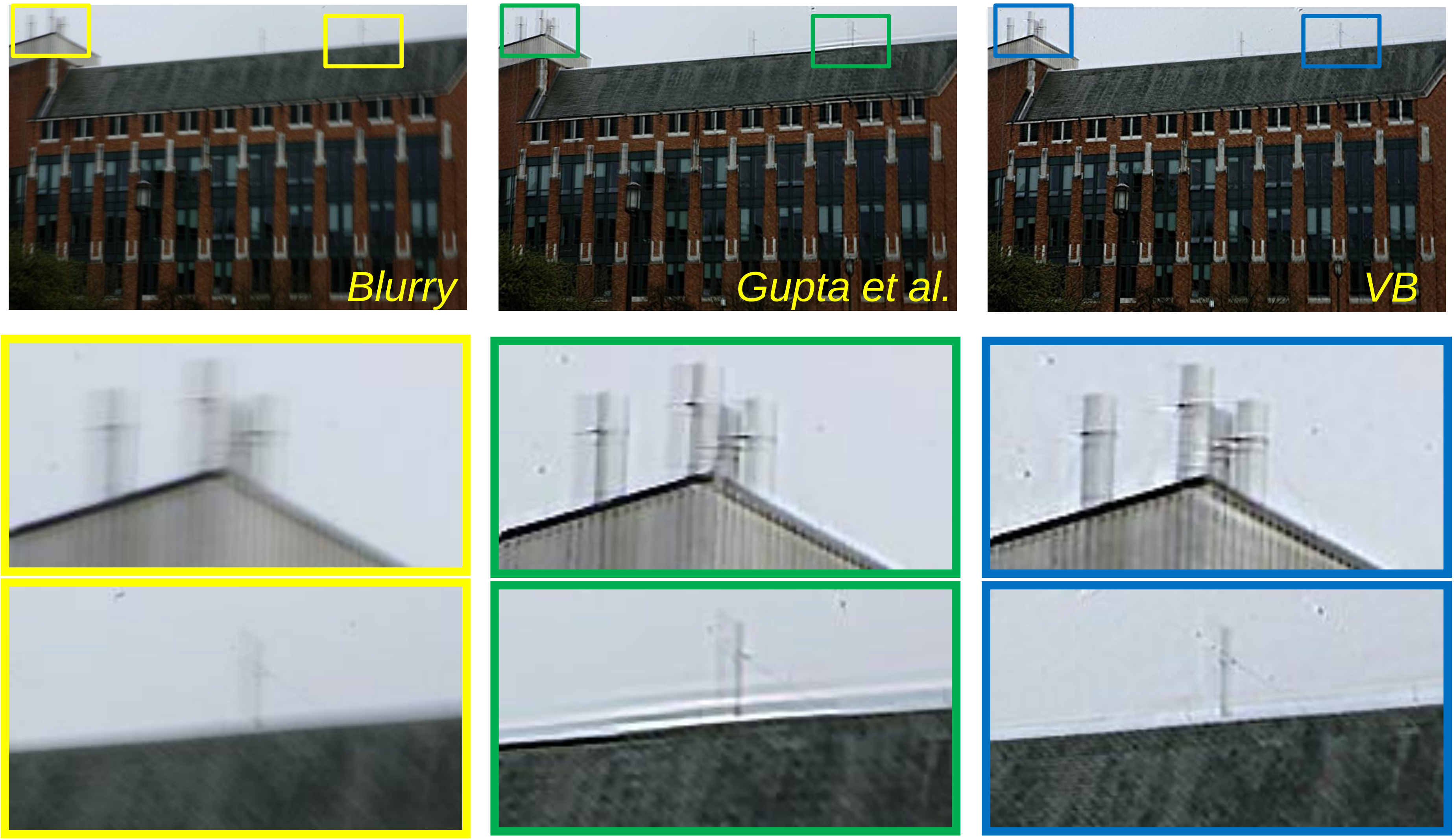
[Yuan, et al., *SIGGRAPH*, 2007]

Non-Uniform Real-World Deblurring



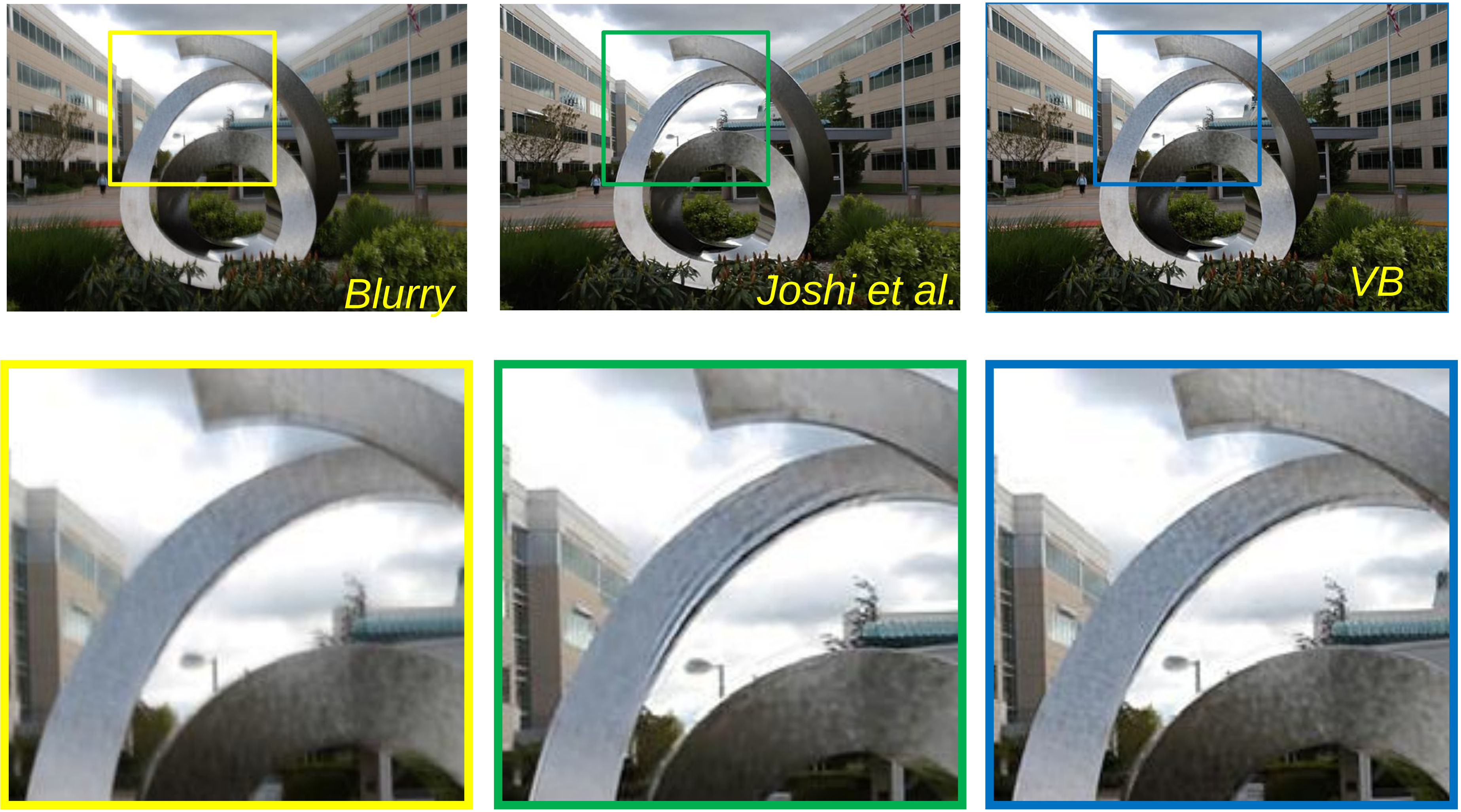
O. Whyte et al. , *Non-uniform deblurring for shaken images*, CVPR, 2010.

Non-Uniform Real-World Deblurring



S. Hirsch et al. , *Single image deblurring using motion density functions*, ECCV, 2010.

Non-Uniform Real-World Deblurring

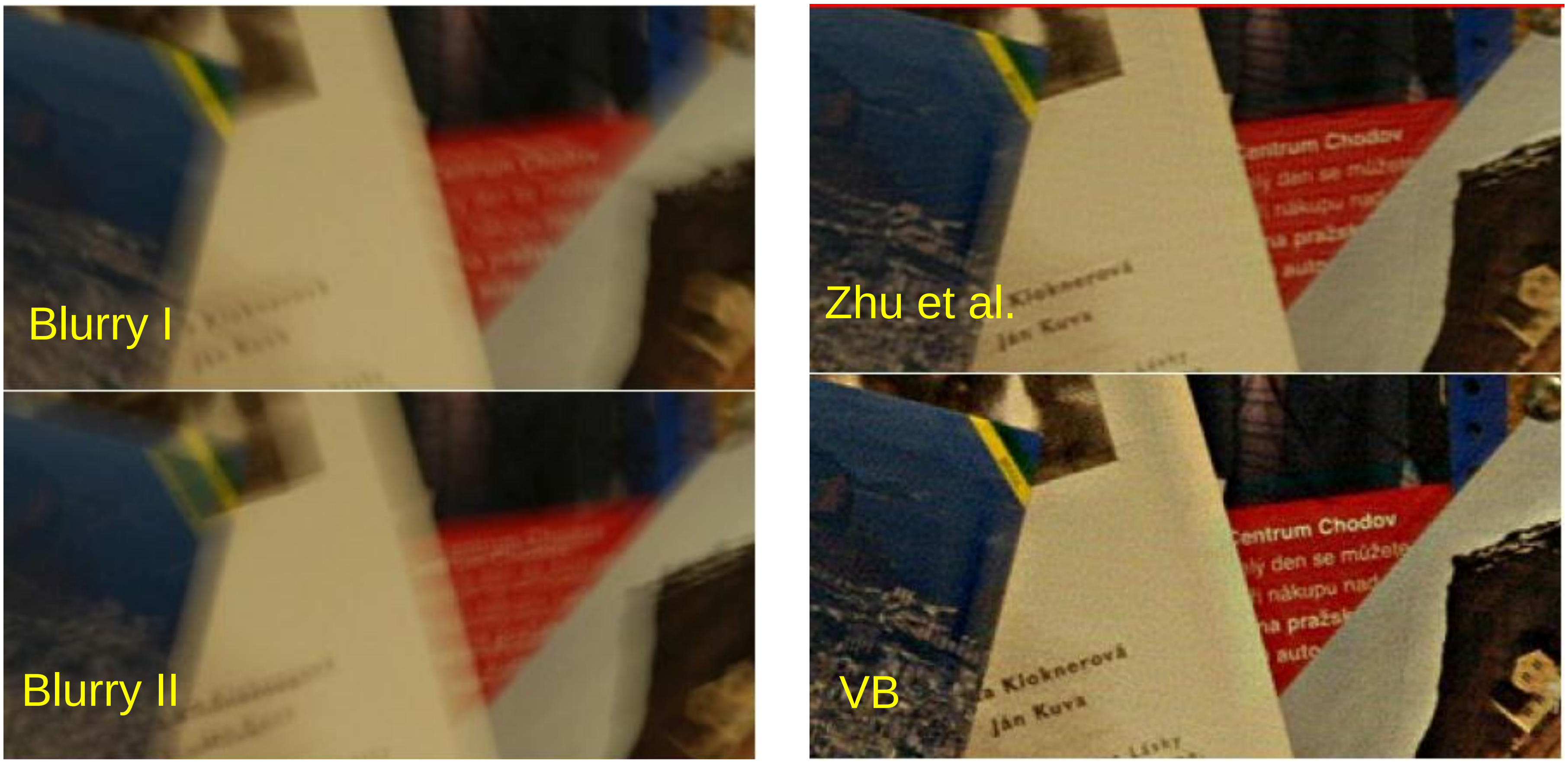


N. Joshi et al. , *Image deblurring using inertial measurement sensors*, SIGGRAPH, 2010.

Non-Uniform Real-World Deblurring

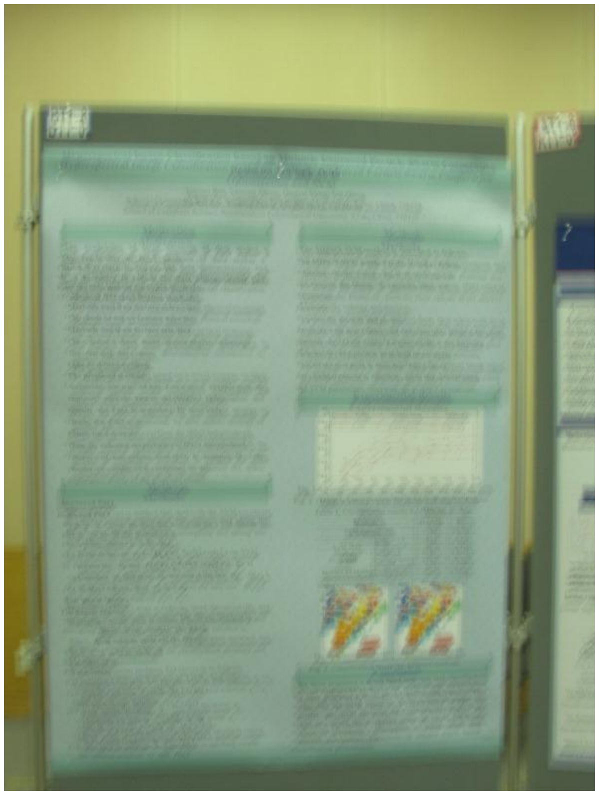
S. Hirsch et al. , *Fast removal of non-uniform camera shake*, ICCV, 2011.

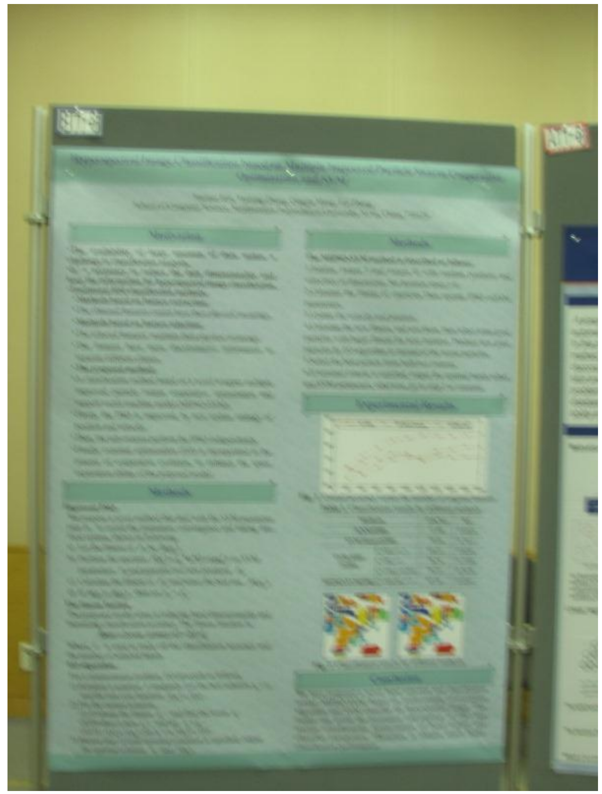
Dual Motion Real-World Deblurring

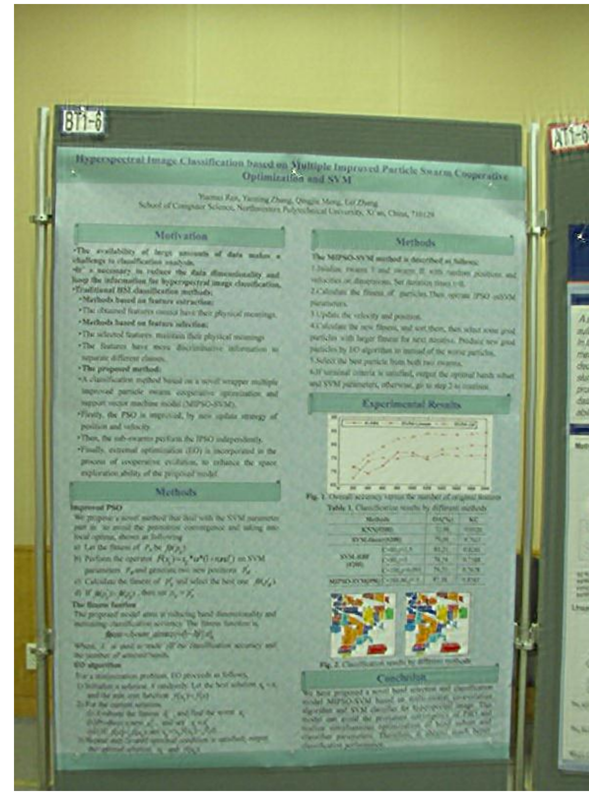


X. Zhu et al. , *Deconvolving PSFs for better motion deblurring using multiple images*, ECCV, 2012.

Personal Photos







two blurry photos taken at a conference

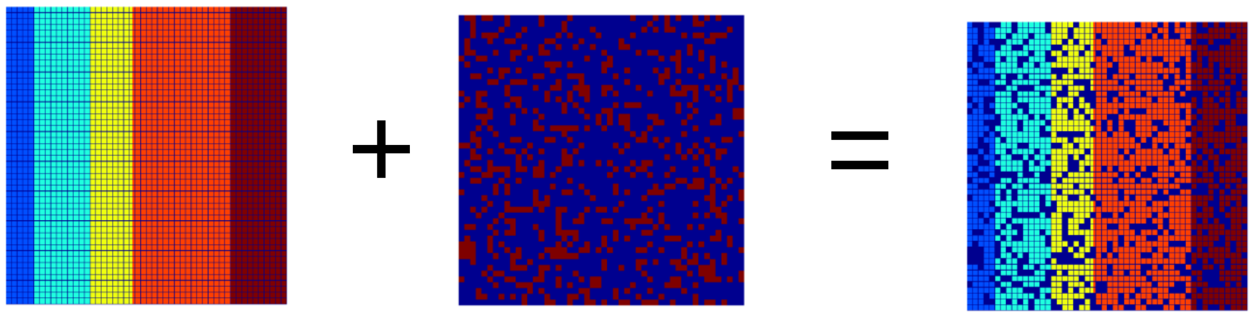
recovered image

Recap

- Three (interrelated) issues with the convex ℓ_1 norm:
  1. Over-shrinkage at the expense of sparsity
  2. Correlated dictionaries disrupt performance
  3. Extra dictionary parameters may be hard to estimate
- In all three, non-convex substitutes can potentially enhance performance dramatically.

Similar Principles Apply to other Low-Dimensional Models

Robust PCA



low rank sparse observation

[Candès et al., 2011; Wipf, 2012]

References

1. C. Bishop and M. Tipping, "Variational Relevance Vector Machines," UAI, 2000.
2. A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, "Understanding and Evaluating Blind Deconvolution Algorithms," *Computer Vision and Pattern Recognition (CVPR)*, 2009.
3. J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, "Variational EM Algorithms for Non-Gaussian Latent Variable Models," NIPS, 2006.
4. D. Wipf, "Sparse Estimation Algorithms that Compensate for Coherent Dictionaries," *MSRA Tech Report*, 2013.
5. D. Wipf, B. Rao, S. Nagarajan, "Latent Variable Bayesian Models for Promoting Sparsity," *IEEE Trans. Info Theory*, 2011.
6. D. Wipf and H. Zhang, "Revisiting Bayesian Blind Deconvolution," *MSRA Tech Report*, 2013.

Thank You