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…

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?
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.
Visual data exhibit **low-dimensional structures** due to rich **local** regularities, **global** symmetries, **repetitive** patterns, or **redundant** sampling.
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:**
  \[ y = A x + \varepsilon \]
  - \( y \): \( m \)-dimensional observations
  - \( A \): matrix of \( n \) basis vectors or features
  - \( x \): unknown sparse coefficients
  - \( \varepsilon \): noise

- **Objective:** Estimate the *sparse* \( x \) assuming \( n \gg m \)

underdetermined system
Example

\[
y = \begin{bmatrix} -4 \\ -5 \\ 3 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 4 & 1 & 1 & 6 \\ -2 & 1 & -4 & 2 & -3 \\ 3 & 3 & 2 & -2 & 1 \end{bmatrix}
\]

Want to find an \( x \) that solves

\[
y = A x
\]

non-sparse

\[
x = \begin{bmatrix} 4 \\ -1 \\ 3 \\ 5 \\ -2 \end{bmatrix}
\]

sparse

\[
x_0 = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ -1 \end{bmatrix}
\]

Sparse representations reflect low-dimensional structure
Sinusoid and Spikes Example

\[ A = [ \text{DFT basis} ] \]
Sinusoid and Spikes Example

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

\[ y_i = \int_u z(u) \exp(-2\pi j k(t_i) \cdot u) du \]

Observations are Fourier coefficients!
Signal Acquisition

\[ y = F_\Omega \Psi x \]

A few Fourier coefficients

Wavelet coefficients: \[ z = \Psi x \]

[Lustig, Donoho + Pauly ‘10] ... brain image – Lustig ‘12
Signal Acquisition

\[ y \approx Ax \]

A few Fourier coefficients

 mostly zero

Wavelet coefficients

Compression - JPEG

(Patches of) ... input image

\( y \approx \begin{bmatrix} A \end{bmatrix} x \) coefficients

[Wallace '91]
Compression – Learned Dictionary

\[ y \approx A x \]

(Patches of) ... input image

\( A \) Learned dictionary

\( x \) coefficients

See [Elad+Bryt ’08], [Horev et. Al., ‘12] ... Image: [Aharon+Elad ‘05]
Representing Faces under Different Lighting

\[ A_i = \begin{bmatrix} \cdots \end{bmatrix} \in \mathbb{R}^{m \times n_i} \]

\[ y \approx x_{i,1} + x_{i,2} + \ldots + x_{i,n} = A_i x_i \]
Face Recognition

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

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

Test image

$$A = [A_1 | A_2 | \cdots | A_k]$$

Combined training dictionary

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

coefficients

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

corruption, occlusion

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

\[
\begin{bmatrix}
y \\
\end{bmatrix} =
\begin{bmatrix}
A' \\
\end{bmatrix}
\begin{bmatrix}
I \\
\end{bmatrix}
\begin{bmatrix}
x' \\
e \\
\end{bmatrix}
\]

One large underdetermined system: \( y = A'x' \)

**Sparse Representation:**
- Given a sparse feasible solution \( y \approx \Phi'x' \)
- Location of large nonzeros in \( x \) should reveal identity

[Wright et al., PAMI 2009]
Prevalence of Sparse Representations

\[ y = Ax \]

<table>
<thead>
<tr>
<th>Signal acquisition</th>
<th>Image compression</th>
<th>Face Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Signal acquisition" /></td>
<td><img src="image2" alt="Image compression" /></td>
<td><img src="image3" alt="Face Recognition" /></td>
</tr>
<tr>
<td>( x^* ) contains just a few significant wavelet coefficients.</td>
<td>( x^* ) uses just a few dictionary elements.</td>
<td>( e^* ) corrects a few gross errors.</td>
</tr>
</tbody>
</table>
Optimization

- Ideal (noiseless) case:

\[
\min_x \|x\|_0 \quad \text{s.t. } y = Ax
\]

- Approximate case:

\[
\min_x \|y - Ax\|_2^2 + \lambda \|x\|_0
\]
Uniqueness

Theorem (Gorodnitsky + Rao ’97).
Suppose $y = Ax_0$, and let $k = \|x_0\|_0$. If $\text{null}(A)$ contains no $2k$-sparse vectors, $x_0$ is the unique optimal solution to

$$\text{minimize } \|x\|_0 \text{ subject to } y = Ax.$$
Difficulties

Forward model is linear, the inverse problem is difficult:

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

2. Objective is discontinuous

\[
\text{minimize } \|x\|_0 \quad \text{subject to} \quad Ax = y. 
\]

INTRACTABLE

Computationally tractable approximate methods are needed …
Replace $\ell_0$ Norm with Convex $\ell_1$ Norm

- Ideal (noiseless) case:

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

- Approximate case:

$$\min_x \|y - \Phi x\|_2^2 + \lambda \|x\|_1$$

Tightest convex relaxation over unit ball
Why might this work?

$$\text{minimize } \|x\|_1 \text{ subject to } Ax = y.$$
Advantages of $\ell_1$ Substitution

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

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

- Many performance guarantees:

  $$x_0 = \arg\min_x \|y - Ax\|_2^2 + \lambda \|x\|_0$$

  $$\approx \arg\min_x \|y - Ax\|_2^2 + \lambda \|x\|_1$$

  [Candès et al., 2006; Donoho, 2006]
Dictionary Correlation Structure

Low Correlation: Easy

$$A^T A$$

Examples:

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

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

High Correlation: Hard

$$A^T A$$

Example:

$$A_{(cor)} = \Psi A_{(uncor)} \Phi$$

arbitrary

block diagonal
Example

\[ A = [a_1, a_2, a_4, a_4] \quad x_0 = [0, 0, 1, 1]^T \]

Sparse Generative Solution

Minimum \( \ell_1 \) Norm Solution

\[ y = a_3 + a_4 \]
\[ \|x\|_1 = 2 \]

\[ y = \frac{1}{4} a_1 + \frac{1}{4} a_2 + \frac{1}{4} a_4 \]
\[ \|x\|_1 = \frac{3}{4} \]

Require conditions to disallow correlated basis vectors in a restricted space
Mutual Coherence

♦ Let $A = [a_1, \ldots, a_n]$

♦ Mutual coherence: $\mu(A) = \max_{i \neq j} \frac{|a_i^T a_j|}{\|a_i\|_2 \|a_j\|_2}$

♦ Measures maximum (off-diagonal) correlation among dictionary columns.
Noiseless Analysis of $\ell_1$

Theorem

Assume

$$\|x_0\|_0 < \frac{1}{2} \left[ 1 + \frac{1}{\mu(A)} \right]$$

Then $x_0$ is the unique solution to

$$\min_x \|x\|_1 \quad \text{s.t.} \quad y = A x_0 = A x$$

[Donoho and Elad, 2003]
Noisy Analysis of $\ell_1$

Theorem

Assume $y = Ax_0 + \varepsilon$ with

$$\|\varepsilon\|_2 \leq \beta \quad \|x_0\|_0 < \frac{1}{4} \left[ 1 + \frac{1}{\mu(A)} \right]$$

Then $\hat{x} = \arg\min_x \|x\|_1$ s.t. $\|y - Ax\|_2 \leq \beta$

satisfies

$$\|\hat{x} - x_0\|_2^2 \leq \frac{4\beta^2}{1 - \mu(A)[4\|x_0\|_0 - 1]}$$

[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

Observation Matrix = Low-rank Structures + Sparse Component
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
- \( \eta \) : Gaussian errors
Classical PCA

\[
\min_X \frac{1}{\lambda} \|Y - X\|_F^2 + \text{rank}[X]
\]

- 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
[Candès et al. 2011]

\[
\text{rank}(X) = \#\{\sigma_i(X) \neq 0\}. \quad \|E\|_0 = \#\{E_{ij} \neq 0\}. \\
\quad \downarrow \downarrow \\
\|X\|_* = \sum_i \sigma_i(X). \quad \|E\|_1 = \sum_{ij} |E_{ij}|. 
\]

♦ 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:

\[
\begin{align*}
Y &= 1_{i,j} \\
X &= 1_{i,j} \\
E &= 0 \\
X &= 0 \\
E &= 1_{i,j}
\end{align*}
\]

Can we recover \( X \) that are incoherent with the standard basis?

Certain sparse error patterns \( E \) make recovering \( X \) impossible:

\[
\begin{align*}
X \\
E &= e_i v^* \\
Y &= X + E
\end{align*}
\]

Can we correct \( E \) whose support is not adversarial?
Non-Uniqueness Issues

Some very sparse matrices are also low-rank:

\[
\begin{align*}
Y &= 1_{ij} \\
X &= 1_{ij} \\
E &= 0 \\
X &= 0 \\
E &= 1_{ij}
\end{align*}
\]

Can we recover \( X \) that are incoherent with the standard basis?

Certain sparse error patterns \( E \) make recovering \( X \) impossible:

\[
\begin{align*}
X &+ E = e_iv^* \\
Y &= X + E
\end{align*}
\]

Can we correct \( E \) whose support is not adversarial?
Non-Uniqueness Issues

Some very sparse matrices are also low-rank:

\[
\begin{bmatrix}
1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1
\end{bmatrix}
+ \begin{bmatrix}
0
\end{bmatrix}
\quad \text{or} \quad
\begin{bmatrix}
0
\end{bmatrix}
+ \begin{bmatrix}
1
\end{bmatrix}
\]

Can we recover \( X \) that are incoherent with the standard basis?

Certain sparse error patterns \( E \) make recovering \( X \) impossible:

\[
\begin{bmatrix}
X
\end{bmatrix}
+ \begin{bmatrix}
e_i v^*
\end{bmatrix}
= \begin{bmatrix}
Y
\end{bmatrix}
\]

Can we correct \( E \) whose support is not adversarial?
Resolving Ambiguity with Incoherence Conditions

Can we recover $X$ that are incoherent with the standard basis from almost all errors $E$?

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

Singular vectors of $X$ not too spiky:

$$\max_i \|U_i\|^2 \leq \mu r/m.$$  
$$\max_i \|V_i\|^2 \leq \mu r/n.$$  

not too cross-correlated:

$$\|UV^*\|_\infty \leq \sqrt{\mu r/mn}$$

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

$$\text{support}(E) \sim \text{uni}(\frac{[m] \times [n]}{\rho mn})$$

Incoherence condition: [Candès + Recht ’08]
## Main Result – Correct Recovery

### Theorem

If $X_0 \in \mathbb{R}^{m \times n}$, $n \geq m$ has rank $r$, and $E_0$ has Bernoulli support with error probability $\varepsilon \leq \rho_s nm$, then with very high probability

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

and the minimizer is unique

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

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"
A Suite of Models and Theoretical Guarantees

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

- [Zhou et. al. ‘09] Spatiaally 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+Sanhavi ‘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
  \[ (X^0_1, X^0_2) = \arg \min \|X_1\|_{(1)} + \lambda \|X_2\|_{(2)} \quad \text{s.t.} \quad X_1 + X_2 = X^0_1 + X^0_2 \]
- [W.+Ganesh+Min+Ma, I&I’13] – superposition of decomposable structures
  \[ (X^0_1, \ldots, X^0_k) = \arg \min \sum \lambda_i \|X_i\|_{(i)} \quad \text{s.t.} \quad \mathcal{P}_Q(\sum_i X_i) = \mathcal{P}_Q(\sum_i X^0_i) \]

Take home message: Let the data and application tell you the structure…
Applications – *Low rank structures in visual data*

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

**corrupted data**

Basic Decomposition:

Low-rank Structures + Sparse Structures

Generalization to visual data: *add nonlinear deformation* $\tau$ ?
Real Face Images from the Internet: Low-Rank Structures?
Robust Alignment of Multiple (Face) Images

\[ D \text{ – corrupted & misaligned observation} \quad A \text{ – aligned low-rank images} \quad E \text{ – sparse errors} \]

\[ D \circ \tau = A_0 + E_0 \]

**Problem:** Given \( D \circ \tau = A_0 + E_0 \), recover \( \tau, A_0 \) and \( E_0 \).

**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 ($D$)
Output: aligned faces (\( D \circ \tau \))
Output: clean low-rank faces ($A$)
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

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

- **Low-rank component**: (regular patterns…)

- **Sparse component**: (occlusion, corruption, foreground…)

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

$D$ – deformed observation

$A$ – low-rank structures

$E$ – sparse errors
$D$ – deformed observation  $A$ – 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

Congratulations
Structured Texture Completion and Repairing

Input

Output

TILT

Photoshop
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)

Po-Sen Huang, Scott Chen, Paris Smaragdis, Mark Hasegawa-Johnson, ICASSP 2012.
Fig. 1. The diagram of the workflow of the method presented in this paper.

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...**
Core References:

- **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](http://perception.csl.illinois.edu/matrix-rank/home.html)

Colleagues:

- Prof. Emmanuel Candès (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

$\ell^1$ minimization seeks a sparse solution to an underdetermined linear system of equations:

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

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

$$\min \| L \|_* + \lambda \| S \|_1 \text{ s.t. } L + S = D$$
Two noise-aware variants

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

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

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

$$\min \|L\|_* + \lambda\|S\|_1 + \frac{\gamma}{2}\|L + S - D\|_F^2$$
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 raised to $0.35 on a pre-split basis, equal to a 25 pct basis.

Chrysler said the stock dividend is payable March 23 while the cash dividend is payable of record March 23. It said cash will be paid on April 9.

With the split, Chrysler said 13.2 mn shares in its stock repurchase program that began last quarter now has a target of 55.3 mn shares with 26.5 mn outstanding.

Chrysler said in a statement the action "is a reflection of our outstanding performance over the past few years and our optimism 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 $10^4 - 10^6$ **unknowns**, or more.

Time = (#iterations for an $\varepsilon$-accurate soln.) $\times$ (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:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Rank</th>
<th>$|E|_0$</th>
<th># iterations</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT</td>
<td>5.99e-006</td>
<td>50</td>
<td>101,268</td>
<td>8,550</td>
<td>119,370.3</td>
</tr>
<tr>
<td>DUAL</td>
<td>8.65e-006</td>
<td>50</td>
<td>100,024</td>
<td>822</td>
<td>1,855.4</td>
</tr>
<tr>
<td>APG</td>
<td>5.85e-006</td>
<td>50</td>
<td>100,347</td>
<td>134</td>
<td>1,468.9</td>
</tr>
<tr>
<td>APG$_p$</td>
<td>5.91e-006</td>
<td>50</td>
<td>100,347</td>
<td>134</td>
<td>82.7</td>
</tr>
<tr>
<td>EALM$_p$</td>
<td>2.07e-007</td>
<td>50</td>
<td>100,014</td>
<td>34</td>
<td>37.5</td>
</tr>
<tr>
<td>IALM$_p$</td>
<td>3.83e-007</td>
<td>50</td>
<td>99,996</td>
<td>23</td>
<td><strong>11.8</strong></td>
</tr>
</tbody>
</table>

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…
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:

<table>
<thead>
<tr>
<th>Function class ( \mathcal{F} )</th>
<th>( f(x_k) - f(x^*) )</th>
</tr>
</thead>
</table>
| **smooth**

\( f \) convex, differentiable

\[ \| \nabla f(x) - \nabla f(x') \| \leq L \| x - x' \| \]

\[ \frac{CL\|x_0 - x^*\|^2}{k^2} = \Theta \left( \frac{1}{k^2} \right) \]

| **nonsmooth**

\( f \) convex

\[ |f(x) - f(x')| \leq M \| x - x' \| \]

\[ \frac{CM\|x_0 - x^*\|}{\sqrt{k}} = \Theta \left( \frac{1}{\sqrt{k}} \right) \]

* 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:

<table>
<thead>
<tr>
<th>Function class ( \mathcal{F} )</th>
<th>( f(x_k) - f(x^*) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>smooth ( f ) convex, differentiable ( |\nabla f(x) - \nabla f(x')| \leq L|x - x'| )</td>
<td>( \frac{CL|x_0 - x^*|^2}{k^2} = \Theta \left( \frac{1}{k^2} \right) )</td>
</tr>
<tr>
<td>nonsmooth ( f ) convex (</td>
<td>f(x) - f(x')</td>
</tr>
</tbody>
</table>

For \( f(x_k) - f(x^*) \leq \varepsilon \), 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 \( \min f(x) \), with \( f \) convex, differentiable, and \( \nabla f \) \( L \)-Lipschitz.

**Gradient descent:** \( x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k) \)
What does gradient descent do anyway?

Consider $\min f(x)$, with $f$ convex, differentiable, and $\nabla f$ $L$-Lipschitz.

**Gradient descent:** $x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$

Quadratic approximation to $f$ around $x_k$:

$$\hat{f}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \| x - x_k \|^2$$
What does gradient descent do anyway?

Consider \( \min f(x) \), with \( f \) convex, differentiable, and \( \nabla f \) \( L \)-Lipschitz.

**Gradient descent:** \( x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k) \)

Quadratic approximation to \( f \) around \( x_k \):

\[
\hat{f}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \| x - x_k \|^2
\]
What does gradient descent do anyway?

Consider \( \min f(x) \), with \( f \) convex, differentiable, and \( \nabla f \) \( L \)-Lipschitz.

Gradient descent: \( x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k) \)

Quadratic approximation to \( f \) around \( x_k \):

\[
\hat{f}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \| x - x_k \|^2
\]

\[
= \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|^2 + \varphi(x_k).
\]

Doesn't depend on \( x \)
What does gradient descent do anyway?

Consider \( \min f(x) \), with \( f \) convex, differentiable, and \( \nabla f \) \( L \)-Lipschitz.

Gradient descent: 

\[
x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)
\]

Quadratic approximation to \( f \) around \( x_k \):

\[
\hat{f}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \| x - x_k \|^2
\]

\[
= \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|^2 + \varphi(x_k).
\]

Key observation: 

\[
x_{k+1} = \arg\min_x f(x, x_k).
\]

At each iteration, the gradient descent minimizes a (separable) quadratic approximation to the objective function, formed at \( x_k \).
What does gradient descent do anyway?

Consider \( \min f(x) \), with \( f \) convex, differentiable, and \( \nabla f \) \( L \)-Lipschitz.

**Gradient descent:** \( x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k) \)

Quadratic approximation to \( f \) around \( x_k \):

\[
\hat{f}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \| x - x_k \|^2
\]

\[
= \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|^2 + \varphi(x_k).
\]

**Key observation:** \( x_{k+1} = \arg \min_x f(x, x_k) \).

*At each iteration, the gradient descent minimizes a (separable) quadratic approximation to the objective function, formed at \( x_k \).*

**Rate for gradient descent:** \( f(x_k) - f(x^*) \leq \frac{CL\|x_0 - x^*\|^2}{k} = O\left(\frac{1}{k}\right) \)
Borrowing the approximation idea...

$$\min \frac{1}{2} ||Ax - y||_2^2 + \lambda ||x||_1$$
Borrowing the approximation idea...

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

smooth \hspace{1cm} nonsmooth
Borrowing the approximation idea…

\[
\min \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1 \equiv \min \ f(x) + g(x)
\]

smooth \hspace{1cm} nonsmooth
Borrowing the approximation idea...

\[
\min \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1 \equiv \min f(x) + g(x)
\]

\text{smooth} \quad \text{nonsmooth}

Just approximate the smooth part:

\[
\hat{F}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)
\]
Borrowing the approximation idea…

\[
\min \frac{1}{2} \| Ax - y \|_2^2 + \lambda \| x \|_1 \quad \equiv \quad \min \ f(x) + g(x) \\
\text{smooth} \quad \text{nonsmooth}
\]

Just approximate the smooth part:

\[
\hat{F}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \| x - x_k \|_2^2 + g(x) \\
= \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|_2^2 + g(x) + \varphi(x_k).
\]
Borrowing the approximation idea...

\[
\min \frac{1}{2} \| Ax - y \|_2^2 + \lambda \| x \|_1 \equiv \min f(x) + g(x)
\]

smooth nonsmooth

Just approximate the smooth part:

\[
\hat{F}(x, x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \| x - x_k \|_2^2 + g(x)
\]

\[
= \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|_2^2 + g(x) + \varphi(x_k).
\]

... and then minimize to get the next iterate:

\[
x_{k+1} = \arg \min_x \hat{F}(x, x_k)
\]

\[
= \arg \min_x \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|_2^2 + g(x).
\]

This is called a proximal gradient algorithm.
Proximal gradient algorithm

\[ \min f(x) + g(x), \text{ with } f \text{ convex differentiable, } \nabla f \text{ } L\text{-Lipschitz.} \]

**Proximal Gradient:**

\[ x_{k+1} = \arg \min_x \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|^2 + g(x) \]

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

\[ F(x_k) - F(x^*) \leq \frac{CL\|x_0 - x^*\|^2}{k} = O\left(\frac{1}{k}\right) \]

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

\[ \text{prox}_{\mu g}(z) = \arg \min_x \frac{1}{2} \| x - z \|^2 + \mu g(x) \]

i.e., minimize \( g \) plus a separable quadratic.
Prox. operators for structure-inducing norms

\[
\text{prox}_{\mu g}(z) = \arg \min_x \frac{1}{2} \|x - z\|_2^2 + \mu g(x)
\]

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

This operator shrinks all of the elements of \( z \) towards zero:

\[
S_\mu(z)
\]

It can be computed in linear time (very efficient).
Prox. operators for structure-inducing norms

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

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

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

\[
\text{prox}_{\mu g}(Z) = US_\mu[\Sigma]V^*.
\]

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

Similar expressions exist for other structure inducing norms.
Summing up: proximal gradient

\[ \min f(x) + g(x), \text{ with } f \text{ convex differentiable, } \nabla f \text{ } L\text{-Lipschitz.} \]

**Proximal Gradient:**

\[ x_{k+1} = \arg \min_x \frac{L}{2} \| x - (x_k - \frac{1}{L} \nabla f(x_k)) \|_2^2 + g(x) \]

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

\[ F(x_k) - F(x^*) \leq \frac{CL\|x_0 - x^*\|_2^2}{k} = O\left(\frac{1}{k}\right) \]

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

\[ \text{prox}_{\mu g}(z) = \arg \min_x \frac{1}{2} \| x - z \|_2^2 + \mu g(x) \]

This is the case for many structure-inducing norms.
What have we accomplished so far?

<table>
<thead>
<tr>
<th>Function class $\mathcal{F}$</th>
<th>$f(x_k) - f(x^*)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>smooth</strong></td>
<td>$f$ convex, differentiable $|\nabla f(x) - \nabla f(x')| \leq L|x - x'|$ $\frac{CL|x_0 - x^*|^2}{k^2} = \Theta \left( \frac{1}{k^2} \right)$</td>
</tr>
<tr>
<td><strong>smooth + structured nonsmooth:</strong></td>
<td>$F = f + g$ $f, g$ convex, $|\nabla f(x) - \nabla f(x')| \leq L|x - x'|$ $\frac{CL|x_0 - x^*|^2}{k} = O \left( \frac{1}{k} \right)$</td>
</tr>
<tr>
<td><strong>nonsmooth</strong></td>
<td>$f$ convex $</td>
</tr>
</tbody>
</table>

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**

\[
x_{k+1} = x_k - \alpha \nabla f(x_k)
\]
Why is the gradient method suboptimal?

For smooth $f$, gradient descent is also suboptimal... intuitively, for badly conditioned functions it may “chatter”:

Gradient descent

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

The heavy ball method treats the iterate as a point mass with momentum, and hence, a tendency to continue moving in direction $x_k - x_{k-1}$:

Heavy ball

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1})$$
Nesterov’s optimal method

Shares some intuition with heavy ball, but not identical.

**Heavy ball:** \[ x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1}) \]

**Nesterov:** \[ y_k = x_k + \beta_k (x_k - x_{k-1}) \]
\[ x_{k+1} = y_k - \alpha \nabla f(y_k) \]

with a very special choice of \( \beta_k \) to ensure the optimal rate:

\[ \beta_k = \frac{t_k - 1}{t_{k+1}} \quad t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \quad \alpha = 1/L \]

---

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

\[ f(x_k) - f(x^*) \leq \frac{C L \| x_0 - x^* \|^2_2}{(k + 1)^2}. \quad (1) \]

This is optimal up to constants.
What about smooth + nonsmooth?

\[ \min \ f(x) + g(x) \]

smooth \ nonsmooth

Again form a separable quadratic upper bound, but now at \( y_k \):

\[ \hat{F}(x, y_k) = f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \frac{L}{2} \| x - y_k \|^2 + g(x) \]
What about smooth + nonsmooth?

\[
\min_{\text{smooth }} f(x) + g(x)
\]

Again form a separable quadratic upper bound, but now at \( y_k \):

\[
\hat{F}(x, y_k) = f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \frac{L}{2} \| x - y_k \|^2 + g(x)
\]

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

\[
x_{k+1} = \arg \min_x \hat{F}(x, y_k)
\]
What about smooth + nonsmooth?

\[
\min \ f(x) + g(x)
\]

\text{smooth} \quad \text{nonsmooth}

Again form a separable quadratic upper bound, but \textbf{now at} \( y_k \):

\[
\hat{F}(x, y_k) = f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \frac{L}{2} \| x - y_k \|^2 + g(x)
\]

Again, \textbf{replace the gradient step} with minimization of the upper bound:

\[
x_{k+1} = \arg \min_x \hat{F}(x, y_k)
\]

\[
= \arg \min_x \frac{1}{L} \| x - (y_k - \frac{1}{L} \nabla f(y_k)) \|^2 + g(x)
\]
What about smooth + nonsmooth?

\[
\min \ f(x) + g(x)
\]

\text{smooth} \quad \text{nonsmooth}

Again form a separable quadratic upper bound, but now at \( y_k \):

\[
\hat{F}(x, y_k) = f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \frac{L}{2} \| x - y_k \|^2 + g(x)
\]

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

\[
x_{k+1} = \arg \min_x \hat{F}(x, y_k)
\]

\[
= \arg \min_x \frac{1}{L} \| x - (y_k - \frac{1}{L} \nabla f(y_k)) \|^2 + g(x)
\]

\[
= \text{prox}_{L^{-1} g}(y_k - \frac{1}{L} \nabla f(y_k)).
\]
What about smooth + nonsmooth?

\[
\min \ f(x) + g(x) \\
\text{smooth} \quad \text{nonsmooth}
\]

Again form a separable quadratic upper bound, but now at \( y_k \):

\[
\hat{F}(x, y_k) \ = \ f(y_k) + \langle \nabla f(y_k), x - y_k \rangle + \frac{L}{2} \| x - y_k \|^2 + g(x)
\]

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

\[
x_{k+1} \ = \ \arg \min_x \hat{F}(x, y_k)
\]
\[
= \ \arg \min_x \frac{1}{L} \| x - (y_k - \frac{1}{L} \nabla f(y_k)) \|^2 + g(x)
\]
\[
= \ \text{prox}_{L^{-1}g}(y_k - \frac{1}{L} \nabla f(y_k)).
\]

Making the same special choice \( y_k = x_k + \beta_k(x_k - x_{k-1}) \), we obtain an \textit{accelerated proximal gradient} algorithm.
Accelerated proximal gradient algorithm

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

**Accelerated Proximal Gradient:**

\[
\begin{align*}
\text{Repeat} \quad & \quad y_k = x_k + \beta_k (x_k - x_{k-1}) \\
& \quad x_{k+1} = \text{prox}_{L^{-1}g}(y_k - \frac{1}{L} \nabla f(y_k)) \\
\text{with} \quad & \quad \beta_k = \frac{t_k-1}{t_{k+1}} \quad \text{and} \quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}.
\end{align*}
\]

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

\[
F(x_k) - F(x^*) \leq \frac{CL\|x_0 - x^*\|^2}{(k+1)^2} = O\left(\frac{1}{k^2}\right)
\]

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

\[
\text{prox}_{\mu g}(z) = \arg\min_x \frac{1}{2}\|x - z\|_2^2 + \mu g(x)
\]
What have we accomplished so far?

<table>
<thead>
<tr>
<th>Function class ( \mathcal{F} )</th>
<th>( f(x_k) - f(x^*) )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>smooth</strong> ( f ) convex, differentiable</td>
<td>( \frac{CL|x_0-x^*|^2}{k^2} = \Theta \left( \frac{1}{k^2} \right) )</td>
</tr>
<tr>
<td>( \nabla f(x) - \nabla f(x') \leq L|x-x'| )</td>
<td></td>
</tr>
<tr>
<td><strong>smooth + structured nonsmooth:</strong> ( F = f + g )</td>
<td>( \frac{CL|x_0-x^*|^2}{k^2} = \Theta \left( \frac{1}{k^2} \right) )</td>
</tr>
<tr>
<td>( f, g ) convex, ( \nabla f(x) - \nabla f(x') \leq L|x-x'| )</td>
<td></td>
</tr>
<tr>
<td><strong>nonsmooth</strong> ( f ) convex</td>
<td>( \frac{CM|x_0-x^*|}{\sqrt{k}} = \Theta \left( \frac{1}{\sqrt{k}} \right) )</td>
</tr>
<tr>
<td>(</td>
<td>f(x) - f(x')</td>
</tr>
</tbody>
</table>

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

$$\min \|x\|_1 \text{ s.t. } Ax = y \quad (\ast)$$

**Continuation:** solve a sequence of unconstrained problems of form

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

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

**Big downside: conditioning.** For $f(x) = \frac{\mu}{2} \|Ax - y\|_2^2$, the gradient is $L$-Lipschitz, with $L = \mu\|A^*A\|$. 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

\[ \min F(x) \text{ s.t. } Ax = y \quad (*) \]

The Lagrangian is

\[ \mathcal{L}(x, \lambda) = F(x) + \langle \lambda, Ax - y \rangle \]
The method of multipliers

\[ \min \ F(x) \ \text{s.t.} \ A x = y \quad (\ast) \]

The **augmented Lagrangian** is

\[ L_\rho(x, \lambda) = F(x) + \langle \lambda, A x - y \rangle + \frac{\rho}{2} \| A x - y \|_2^2. \]

*Extra penalty term*
The method of multipliers

\[
\min \ F(x) \ \text{s.t.} \ Ax = y \quad (*)
\]

The augmented Lagrangian is

\[
\mathcal{L}_\rho(x, \lambda) = F(x) + \langle \lambda, Ax - y \rangle + \frac{\rho}{2} \| Ax - y \|_2^2.
\]

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

\[
x_{k+1} = \arg \min_x \mathcal{L}_\rho(x, \lambda_k)
\]

\[
\lambda_{k+1} = \lambda_k + \rho(Ax_{k+1} - y).
\]
The method of multipliers

\[ \min \ F(x) \text{ s.t. } Ax = y \quad (\ast) \]

The augmented Lagrangian is

\[ \mathcal{L}_\rho(x, \lambda) = F(x) + \langle \lambda, Ax - y \rangle + \frac{\rho}{2} \|Ax - y\|^2. \]

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

\[ x_{k+1} = \arg \min_x \mathcal{L}_\rho(x, \lambda_k) \]
\[ \lambda_{k+1} = \lambda_k + \rho(Ax_{k+1} - y). \]

Solves a sequence of unconstrained problems: \( \min_x \mathcal{L}_\rho(x, \lambda_k) \)

Penalty parameter \( \rho > 0 \) can be constant (avoids ill-conditioning), or increasing for (faster convergence).
Summing up: Method of multipliers

Solves, e.g., \( \min F(x) \) s.t. \( Ax = y \), with \( F \) convex, lsc.

Method of multipliers (augmented Lagrangian)

\[
x_{k+1} = \arg\min_x \mathcal{L}_\rho(x, \lambda_k)
\]

\[
\lambda_{k+1} = \lambda_k + \rho(A x_{k+1} - y).
\]

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

Avoids conditioning problems with the continuation / penalty method.

Under very general conditions \( \lambda_k \) converges to a dual optimal point,

\[
\|Ax_k - y\| \to 0, \text{ and } F(x_k) \to \inf \{ F(x) \mid Ax = y \}.
\]

[Rockafellar ‘73, Eckstein ‘12].
What have we accomplished so far?

Consider the robust PCA problem

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

Augmented Lagrangian

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

The method of multipliers is

\[
(L_{k+1}, S_{k+1}) = \arg \min_{L, S} \|L\|_* + \lambda \|S\|_1 + \langle \Lambda_k, L + S - D \rangle + \frac{\rho}{2} \|L + S - D\|_F^2
\]

\[
\Lambda_{k+1} = \Lambda_k + \rho(L_k + S_k - D)
\]

Each iteration is a large nonsmooth optimization problem…

Is there special structure we can exploit to simplify the iterations?
Special structure: Separable objectives

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

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

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

$$\arg\min_S \mathcal{L}_\rho(L, S, \Lambda) = \arg\min_S \|L\|_* + \lambda \|S\|_1 + \langle \Lambda, L + S - D \rangle + \frac{\rho}{2} \|L + S - D\|_F^2$$
Special structure: Separable objectives

\[
\begin{align*}
\min & \quad \|L\|_* + \lambda \|S\|_1 \quad \text{s.t.} \quad L + S = D
\end{align*}
\]

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

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

\[
\begin{align*}
\arg \min_S \mathcal{L}_\rho(L, S, \Lambda) &= \arg \min_S \|L\|_* + \lambda \|S\|_1 + \langle \Lambda, L + S - D \rangle + \frac{\rho}{2} \|L + S - D\|_F^2 \\
&= \arg \min_S \lambda \|S\|_1 + \frac{\rho}{2} \|S - (D - L - \frac{1}{\rho} \Lambda)\|_F^2 + \varphi(L, D, \Lambda)
\end{align*}
\]
Special structure: Separable objectives

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

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

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

\[
\begin{align*}
\arg \min_S \mathcal{L}_\rho(L, S, \Lambda) &= \arg \min_S \| L \|_* + \lambda \| S \|_1 + \langle \Lambda, L + S - D \rangle + \frac{\rho}{2} \| L + S - D \|_F^2 \\
&= \arg \min_S \lambda \| S \|_1 + \frac{\rho}{2} \| S - (D - L - \frac{1}{\rho} \Lambda) \|_F^2 + \varphi(L, D, \Lambda) \\
&= \prox_{\lambda \rho^{-1} \cdot 1}(D - L - \rho^{-1} \Lambda).
\end{align*}
\]
Minimizing $\mathcal{L}_\rho$ with respect to $S$ is easy:

$$\arg \min_S \mathcal{L}_\rho(L, S, \Lambda) = \prox_{\lambda \rho^{-1} \cdot :1} (D - L - \rho^{-1} \Lambda).$$
Special structure: Separable objectives

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

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

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

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

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

$$\arg \min_L \mathcal{L}_\rho(L, S, \Lambda) = \text{prox}_{\rho^{-1} \|\cdot\|_*}(D - S - \rho^{-1} \Lambda).$$
Special structure: Separable objectives

\[
\begin{aligned}
\min & \quad \|L\|_* + \lambda \|S\|_1 \quad \text{s.t.} \quad L + S = D \\
\end{aligned}
\]

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

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

\[
\arg \min_S \mathcal{L}_\rho(L, S, \Lambda) = \text{prox}_{\lambda \rho^{-1}\|\cdot\|_1}(D - L - \rho^{-1}\Lambda).
\]

Minimizing \(\mathcal{L}_\rho\) with respect to \(L\) is also easy:

\[
\arg \min_L \mathcal{L}_\rho(L, S, \Lambda) = \text{prox}_{\rho^{-1}\|\cdot\|_*}(D - S - \rho^{-1}\Lambda).
\]

Why not just alternate?

\[
\begin{aligned}
L_{k+1} &= \arg \min_L \mathcal{L}_\rho(L, S_k, \Lambda_k) = \text{prox}_{\rho^{-1}\|\cdot\|_*}(D - S_k - \rho^{-1}\Lambda_k). \\
S_{k+1} &= \arg \min_S \mathcal{L}_\rho(L_{k+1}, S, \Lambda_k) = \text{prox}_{\lambda \rho^{-1}\|\cdot\|_1}(D - L_{k+1} - \rho^{-1}\Lambda_k). \\
\Lambda_{k+1} &= \Lambda_k + \rho(L_{k+1} + S_{k+1} - D)
\end{aligned}
\]
More generally: Alternating Directions MoM

$$\min f(x) + h(z) \quad \text{s.t.} \quad Ax + Bz = y$$

Aug. Lagrangian: \[ \mathcal{L}_\rho(x, z, \lambda) = f(x) + h(z) + \langle \lambda, Ax + Bz - y \rangle + \frac{\rho}{2} \|Ax + Bz - y\|_F^2 \]

**Alternating Directions Method of Multipliers (ADMM)**

\[ x_{k+1} = \arg\min_x \mathcal{L}_\rho(x, z_k, \lambda_k) \]
\[ z_{k+1} = \arg\min_z \mathcal{L}_\rho(x_{k+1}, z, \lambda_k) \]
\[ \lambda_{k+1} = \lambda_k + \rho(Ax_{k+1} + Bz_{k+1} - y) \]
Alternating Directions MoM

\[
\min f(x) + h(z) \quad \text{s.t.} \quad Ax + Bz = y
\]

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

Alternating Directions Method of Multipliers (ADMM)

\[
\begin{align*}
x_{k+1} &= \arg\min_x \mathcal{L}_\rho(x, z_k, \lambda_k) \\
z_{k+1} &= \arg\min_z \mathcal{L}_\rho(x_{k+1}, z, \lambda_k) \\
\lambda_{k+1} &= \lambda_k + \rho(Ax_{k+1} + Bz_{k+1} - y)
\end{align*}
\]

Convergence: if \( f, h \) closed, proper, convex functions, and \( \mathcal{L} \) has a saddle point, then \( \lambda_k \) converges to a dual optimal point, \( Ax_k + Bz_k \to y \) and \( f(x_k) + h(z_k) \to \inf \{ f(x) + h(z) \mid Ax + Bz = y \} \).

Convergence rate \( O(1/k) \), in a certain sense [He+Yuan ‘11].
Linearized Alternating Directions MoM

\[
\min f(x) + h(z) \quad \text{s.t.} \quad Ax + Bz = y
\]

Aug. Lagrangian: \[ \mathcal{L}_\rho(x, z, \lambda) = f(x) + h(z) + \langle \lambda, Ax + Bz - y \rangle + \frac{\rho}{2} \|Ax + Bz - y\|_F^2 \]

ADMM: \[
x_{k+1} = \arg \min_x \mathcal{L}_\rho(x, z_k, \lambda_k)
\]
\[
= \arg \min_x f(x) + \frac{\rho}{2} \|Ax + Bz_k - y + \frac{1}{\rho} \lambda_k\|_2^2
\]

Complicated if \( A, B \neq I \)

Linearized ADMM: just take a proximal gradient step...

\[
x_{k+1} = \arg \min_x f(x) + \frac{\rho}{2\tau} \|x - (x_k - \tau A^*(Ax_k + Bz_k - y + \frac{1}{\rho} \lambda_k))\|_2^2
\]
\[
= \text{prox}_{\frac{\tau}{\rho} f}(x_k - \tau A^*(Ax_k + Bz_k - y - \frac{1}{\rho} \lambda_k))
\]

Much more efficient if \( f \) has a simple proximal operator.
\[ \min f(x) + h(z) \quad \text{s.t.} \quad Ax + Bz = y \]

Aug. Lagrangian: \[ L_\rho(x, z, \lambda) = f(x) + h(z) + \langle \lambda, Ax + Bz - y \rangle + \frac{\rho}{2} \| Ax + Bz - y \|_F^2 \]

**Linearized ADMM**

\[
\begin{align*}
x_{k+1} &= \text{prox}_{\frac{\tau}{\rho} f}(x_k - \tau A^* (Ax_k + Bz_k - y + \frac{1}{\rho} \lambda_k)) \\
z_{k+1} &= \text{prox}_{\frac{\tau}{\rho} h}(z_k - \tau B^* (Ax_{k+1} + Bz_k - y + \frac{1}{\rho} \lambda_k)) \\
\lambda_{k+1} &= \lambda_k + \rho (Ax_{k+1} + Bz_{k+1} - y)
\end{align*}
\]

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

Handles problems with more than two terms, e.g., \( \sum_i f_i(x_i) \).

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:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>Rank</th>
<th>$|E|_0$</th>
<th># iterations</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT</td>
<td>5.99e-006</td>
<td>50</td>
<td>101,268</td>
<td>8,550</td>
<td>119,370.3</td>
</tr>
<tr>
<td>DUAL</td>
<td>8.65e-006</td>
<td>50</td>
<td>100,024</td>
<td>822</td>
<td>1,855.4</td>
</tr>
<tr>
<td>APG</td>
<td>5.85e-006</td>
<td>50</td>
<td>100,347</td>
<td>134</td>
<td>1,468.9</td>
</tr>
<tr>
<td>APG$_p$</td>
<td>5.91e-006</td>
<td>50</td>
<td>100,347</td>
<td>134</td>
<td>82.7</td>
</tr>
<tr>
<td>EALM$_p$</td>
<td>2.07e-007</td>
<td>50</td>
<td>100,014</td>
<td>34</td>
<td>37.5</td>
</tr>
<tr>
<td>IALM$_p$</td>
<td>3.83e-007</td>
<td>50</td>
<td>99,996</td>
<td>23</td>
<td>11.8</td>
</tr>
</tbody>
</table>

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

Proximal gradient ⇒ Accelerated proximal gradient ⇒ ALM ⇒ 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** ⇒ **proximal gradient methods**
- **Separable objectives** ⇒ **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?
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
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
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 $\ell_0$ Norm with Convex $\ell_1$ Norm

Ideal (noiseless) case:

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

$\|x\|_1 = \sum_i |x_i|$,

Relaxed case:

$$\min_x \|y - \Phi x\|_2^2 + \lambda \|x\|_1$$
Non-Convexity via Iterative Reweighted $\ell_1$

Non-convex penalty $g(|x|)$

- concave, non-decreasing

Updates:

$\begin{align*}
    x^{(k+1)} &\leftarrow \arg\min_{x} \sum_{i} w_{i}^{(k)} |x_{i}| \quad \text{s.t.} \quad y = A x \\
    w^{(k+1)} &\leftarrow \left. \frac{\partial g(u)}{\partial u} \right|_{u=|x^{(k+1)}|}
\end{align*}$

[Ref: Fazel et al., 2003]
Example

Penalty function:

\[ g(|x|) = \sum_i \log(|x_i| + \epsilon), \quad \epsilon > 0 \]

Updates:

\[
x^{(k+1)} \leftarrow \arg\min_x \sum_i w_i^{(k)} |x_i| \quad \text{s.t.} \quad y = Ax
\]

\[
w_i^{(k+1)} \leftarrow \frac{1}{\left( |x_i^{(k+1)}| + \epsilon \right)}
\]

[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 $\ell_1$ norm may over-shrink large magnitude coefficients.

2. **Correlation Problem**: The dictionary $A$ has some correlated columns which disrupt $\ell_0$-$\ell_1$ 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 $\ell_1$ penalty favors both **sparse** and **low-variance** solutions:

\[ \|x\|_0 \quad \leftrightarrow \quad \|x\|_1 \quad \leftrightarrow \quad \|x\|_2 \]

- Scale-sensitive $\ell_1$ 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 $x_0$ are highly scaled, then the sparse recovery problem should be easier.

- The $\ell_1$ solution may overly shrink large coefficients to achieve lower variance, and hence may not exploit the simpler scenario.
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 $x_0$.
  3. Compute signal via $y = A x_0$.
  4. Run $\ell_1$ and OMP (a very simple greedy strategy) to try and correctly estimate $x_0$.
  5. Average over 1000 trials to compute empirical probability of failure.

- Repeat with different sparsity values, i.e., $\|x_0\|_0$. 
Results

Unit Coefficients

Scaled Coefficients

OMP is significantly better!
Underlying Problem

\[ \Psi(u,v) = \text{set of sparse vectors } x_0 \text{ with support pattern } u \text{ and sign pattern } v \]

Example:

\[ x_0 = \begin{bmatrix} 2.3 \\ 0 \\ -1.6 \\ 0 \end{bmatrix} \in \Psi\{(1,3),\{+,+\}\} \]

Theorem

If \[ \arg \min_{x:y=Ax} \|x\|_0 \neq \arg \min_{x:y=Ax} \|x\|_1 \]

for some \( x_0 \in \Psi(u,v), \ y = Ax_0 \), then \( \ell_1 \)

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 $\ell_1$-norm solution.

2. For any $A$ and $\Psi(u,v)$, there will always be cases where it performs better (… *helps with scaling/shrinkage issues*).

$g(|x|)$

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

[Wipf, 2011]
Simulation Example Revisited

- For each test case:
  1. Generate a random dictionary $\Phi$ with 50 rows and 100 columns.
  2. Generate a sparse coefficient vector $x_0$.
  3. Compute signal via $y = A x_0$.
  4. Run $\text{VB}$, $\ell_1$ and $\text{OMP}$ (simple greedy strategy) to try and correctly estimate $x_0$.
  5. Average over 1000 trials to compute empirical probability of failure.

- Repeat with different sparsity values, i.e., $\|x_0\|_0$. 
Results

Unit Coefficients

Highly Scaled Coefficients

Error Rate vs $\|x_0\|_0$ for OMP, $\ell_1$, and VB.
Practical Example: Outlier Detection
Outlier Problem Cont.

- Linear generative model:
  \[ y = Ax + \varepsilon \]
  - \( m \)-dimensional observations
  - predictor variables
  - unknown coefficients, non-sparse
  - sparse noise

- **Objective**: Estimate \( x \) while rejecting outliers
Convert to Sparse Estimation Problem

\[ \text{Proj}_{null[A^T]}(y) = \text{Proj}_{null[A^T]}(Ax + \varepsilon) = \text{Proj}_{null[A^T]}(\varepsilon) \]

\[ \min_\varepsilon \| \varepsilon \|_0 \quad \text{s.t.} \quad \tilde{y} = \Phi \varepsilon \]

Once outliers are known, can estimate \( x \) via:

\[ \hat{x} = (A^T A)^{-1} A^T (y - \varepsilon) \]

[Candès and Tao, 2004]
Practical Solutions

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

\[
\min_{\varepsilon} \|\varepsilon\|_1 \quad \text{s.t.} \quad \tilde{y} = \Phi \varepsilon
\]

- Can instead use non-convex VB …
Practical Example:
Surface Normal Estimation via Photometric Stereo

\[
Y = \rho N \begin{bmatrix} L \end{bmatrix}
\]

\[\rho N = Y L^+\]

[Woodham, 1980]

For basic Lambertian surface
Robust Surface Normal Estimation

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

- Alternative per-pixel model:

  \[ y = \mathbf{L} \mathbf{n} + \mathbf{\epsilon} \]

  - 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]

Bunny Image

Ground Truth

VB Error Map

$\ell_1$ Error Map

[Ikehata et al., 2012]
## Aggregate Results

[# of images varying]

<table>
<thead>
<tr>
<th>No. of images</th>
<th>Mean Error (deg.)</th>
<th>( \ell_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VB</td>
<td>( \ell_1 )</td>
</tr>
<tr>
<td>5</td>
<td>5.2</td>
<td>11.9</td>
</tr>
<tr>
<td>10</td>
<td>2.8</td>
<td>5.6</td>
</tr>
<tr>
<td>15</td>
<td>1.9</td>
<td>4.0</td>
</tr>
<tr>
<td>20</td>
<td>1.2</td>
<td>2.7</td>
</tr>
<tr>
<td>25</td>
<td>0.81</td>
<td>1.9</td>
</tr>
<tr>
<td>30</td>
<td>0.62</td>
<td>1.6</td>
</tr>
<tr>
<td>35</td>
<td>0.59</td>
<td>1.5</td>
</tr>
<tr>
<td>40</td>
<td>0.53</td>
<td>1.2</td>
</tr>
</tbody>
</table>

[ 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:
\[ A_{(uncor)} \sim \text{iid } N(0,1) \text{ entries} \]
\[ A_{(uncor)} \sim \text{random rows of DFT} \]

Example:
\[ A_{(cor)} = \Psi A_{(uncor)} \Phi \]

arbitrary block diagonal
How do we compensate for dictionary structure?

Simple Example:

Let vector $\alpha$ denote the column norms of $A$ and define

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

Then the problem

$$\min_{x} \|y - Ax\|_2^2 + \lambda \ g(|x|; \alpha)$$

is invariant to column norms.

So what about some function $g$ that depends on the correlation structure $A^T A$
VB is equivalent to solving the penalized regression problem

$$\min_x \|y - Ax\|_2^2 + \lambda g_{VB}(\|x\|; A^T A)$$

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

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

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

Any $m \times n$ dictionary such that $\ell_1$ minimization succeeds for all $\|x_0\|_0 \leq k$

Any dictionary obtained by replacing each column of $A_{(uncor,k)}$ with a “cluster” of $n_i$ basis vectors within a radius $\varepsilon$

$(cluster \ support)$ set of cluster indeces whereby some $x_0$ has at least one nonzero element.
Simple Clustered Example

\[
A_{(\text{cor},k)}^T A_{(\text{cor},k)}
\]

**Problem:**

- The \(\ell_1\) 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 $x_0$ be a sparse signal.

♦ Under mild conditions, a minor variant of VB will recover $x_0$ given any $y = A_{(cor,k)} x_0$ provided

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

for some $\varepsilon$ sufficiently small.

[Wipf and Wu, 2012]

Key Message: Non-convex algorithms can succeed even when strong correlations cause failure with $\ell_1$. 
MEG/EEG Example

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

SNIR=10dB

True

VB

$l_1$

SNIR=0dB

[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):

\[
\min_{x, k \in \Omega_k} \|x\|_0 \quad \text{s.t.} \quad y = A(k)x
\]

♦ Approximate version:

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

♦ Applications: Bilinear models, blind deconvolution, blind image deblurring, etc.

[Fergus et al., 2006; Levin et al., 2011]
Example: Blind Deconvolution

- Observation model:

\[ y = k * x + \varepsilon = A(k)x + \varepsilon \]

convolution operator  
toeplitz matrix

- Would like to estimate the unknown \( x \) blindly since \( k \) is also unknown.

- In many situations (e.g., image deblurring) unknown \( x \) is sparse.
Efficient Convex Substitution?

Solve:

\[
\min_{x, k \in \Omega_k} \|x\|_1 \quad \text{s.t.} \quad y = k \ast x
\]

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

Problem:

\[
\|y\|_1 = \left\| \sum_t k_t x_t \right\|_1 \leq \sum_t k_t \|x_t\|_1 = \|x\|_1 \quad \forall \text{ feasible } k, x
\]

A degenerate solution is favored:

\[
k = \delta, \quad A(k) = I
\]

We can’t use \( \ell_1 \).
Practical Example: Blind Image Deblurring

- Basic convolution model (can be generalized):

\[ y = k \ast x + \varepsilon \]

- Unknown quantities we need to estimate
Gradients of Natural Images are Sparse

Can solve a modified sparse coding problem in gradient domain

\[
\begin{align*}
\mathbf{x} & : \text{vectorized derivatives of the sharp image} \\
\mathbf{y} & : \text{vectorized derivatives of the blurry image}
\end{align*}
\]
 Practical Blind Deblurring Algorithm

- A nearly ideal cost function for blind deblurring is

  \[
  \min_{x, k \in \Omega_k} \left\| y - k \ast x \right\|_2^2 + \lambda \| x \|_0
  \]

  \[\Omega_k = \left\{ k : \sum_i k_i = 1, \ k_i \geq 0, \forall i \right\}\]

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

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

  \[
  \min_{x, k \in \Omega_k} \left\| y - k \ast x \right\|_2^2 + \lambda g_{VB}(x, k)
  \]

  \[g_{VB}(x, k) \neq g_x(x) + g_k(k)\]

  [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
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

\[ \sum_j \omega_j \]

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

Non-Uniform Real-World Deblurring

Non-Uniform Real-World Deblurring

Non-Uniform Real-World Deblurring

Dual Motion Real-World Deblurring

Blurry I

Blurry II

Zhu et al.

VB

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 $\ell_1$ 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

\[
\begin{align*}
\text{low rank} & \quad + \quad \text{sparse} & \quad = \quad \text{observation} \\
\text{Candès et al., 2011; Wipf, 2012}
\end{align*}
\]
References


Thank You