Dynamic Structured (Big) Data Recovery

Namrata Vaswani

Dept. of Electrical and Computer Engineering
Iowa State University
Web: http://www.ece.iastate.edu/~namrata
Acknowledgements

- Based on joint work with various graduate students
  - Dynamic Compressive Sensing (CS): Wei Lu
  - Dynamic Robust PCA: C. Qiu, B. Lois, J. Zhan, P. Narayanamurthy
  - Low Rank Phase Retrieval: Seyedehsara Nayer
  - Computer Vision: Han Guo, C. Qiu

- Other collaborators:
  - Dr. Ian Atkinson (previously at UIC, Radiology), Prof. Leslie Hogben (ISU), Prof. Yonina Eldar (Technion),

- The work was supported by
  - National Science Foundation (NSF)
    - dynamic CS and fast dynamic MRI: CIF-0917015
    - dynamic robust PCA and other dynamic structured data recovery problems: CIF-1117125, CIF-1526870
    - computer vision and video analytics: IIS-1117509, ECCS-0725849, ECCS-1509372, IDBR-1353819

- Rockwell Collins: low-light video enhancement and denoising
Introduction

- In today’s big data age, a lot of data is generated everywhere
  - e.g., tweets, video surveillance camera feeds, Netflix movie ratings’ data, social network connectivity patterns across time, etc

- A lot of it is streaming big data that is not stored or not for too long
  - and often needs to be analyzed on-the-fly.
Introduction

- In today’s big data age, a lot of data is generated everywhere
  - e.g., tweets, video surveillance camera feeds, Netflix movie ratings’ data, social network connectivity patterns across time, etc

A lot of it is streaming big data that is not stored or not for too long
  - and often needs to be analyzed on-the-fly.

Acquired data is often an undersampled, outlier-corrupted, or nonlinear function of the “clean data”.
  - nonlinear: e.g., phaseless
Introduction

- In today’s big data age, a lot of data is generated everywhere
  - e.g., tweets, video surveillance camera feeds, Netflix movie ratings’ data, social network connectivity patterns across time, etc

  ![Movie-User Matrix Sparsity Pattern](image)

- A lot of it is streaming big data that is not stored or not for too long
  - and often needs to be analyzed on-the-fly.

- Acquired data is often an undersampled, outlier-corrupted, or nonlinear function of the “clean data”.
  - nonlinear: e.g., phaseless

- “Clean data” usually has structure, e.g., sparsity or low-rank.
  - in a long sequence, structure properties are dynamic (change with time)
This Talk

Three Dynamic Structured Big (high-dimensional) Data Recovery Problems

- Dynamic Compressive Sensing (CS) - older work
  - clean data is (approx) sparse in a known transform domain
  - measurements: undersampled linear projections
  - useful when data acquisition is slow, e.g., in dynamic MRI or CT
This Talk

Three Dynamic Structured Big (high-dimensional) Data Recovery Problems

- **Dynamic Compressive Sensing (CS) -** older work
  - clean data is (approx) sparse in a known transform domain
  - measurements: undersampled linear projections
  - useful when data acquisition is slow, e.g., in dynamic MRI or CT

- **Dynamic Robust Principal Components Analysis (RPCA) -** focus of this talk
  - clean data lies in a fixed or slowly changing low-dimensional subspace
  - measurements: outlier-corrupted
  - useful for outlier removal and dimension reduction
Three Dynamic Structured Big (high-dimensional) Data Recovery Problems

- **Dynamic Compressive Sensing (CS) - older work**
  - clean data is (approx) sparse in a known transform domain
  - measurements: undersampled linear projections
  - useful when data acquisition is slow, e.g., in dynamic MRI or CT

- **Dynamic Robust Principal Components Analysis (RPCA) - focus of this talk**
  - clean data lies in a fixed or slowly changing low-dimensional subspace
  - measurements: outlier-corrupted
  - useful for outlier removal and dimension reduction

- **Low Rank Phase Retrieval - recently started work**
  - clean data lies in a low-dimensional subspace (is low rank)
  - measurements: phaseless (magnitude-only) linear-projections
  - useful when phase is hard or impossible to obtain, e.g., astronomy, sub-diffraction imaging, Fourier ptychography, ...
Outline

1. Dynamic CS: brief overview

2. Dynamic Robust PCA
   - Background and Problem Formulation
   - Recursive Projected CS (ReProCS) solution
   - ReProCS guarantees
   - Experiments - simulation and real-data (video analytics)

3. Low Rank Phase Retrieval: brief overview

4. Open Questions and Future Plans

5. Other Work (Extra Slides)
   - Correlated-PCA: PCA in data-dependent noise
   - Dynamic Compressive Sensing (CS)
Recursively recover a time sequence of (approximately) sparse signals from highly undersampled linear measurements; using two assumptions

- slow support change over time – introduced in [Vaswani, KF-CS, ICIP’08], and
- slow signal value change over time – commonly used assumption
Dynamic Compressed Sensing (CS): key contributions

- **Kalman Filtered Compressed Sensing and LS-CS** [Vaswani, ICIP’08], [Vaswani, T-SP, 2010]
  1. First recursive solutions to the dynamic CS problem

  1. First approach that achieved *provably exact* recovery using fewer measurements than solutions for static CS need
  2. Reformulated dynamic CS = CS with partial support knowledge
    - Much more general problem

---

$^1$ Wei Lu was my first graduate student
Dynamic Compressed Sensing (CS): key contributions

- **Kalman Filtered Compressed Sensing and LS-CS** [Vaswani, ICIP’08], [Vaswani, T-SP, 2010]
  - First recursive solutions to the dynamic CS problem

  - First approach that achieved *provably exact* recovery using fewer measurements than solutions for static CS need

- Reformulated dynamic CS = *CS with partial support knowledge*
  - Much more general problem

- Noisy measurements: errors provably stable over time [Zhan, Vaswani, T-IT’15]
  - using fewer measurements than solutions for static CS need

- Promising experimental results for fast dynamic & functional MRI

---

¹ Wei Lu was my first graduate student

Namrata Vaswani (Iowa State Univ) Dynamic Structured (Big) Data Recovery
Dynamic Robust PCA: our main message

- In dynamic CS, we used dynamics to reduce sample complexity (number of measurements needed), w/o increasing time complexity.

- In robust PCA, we will show how we can use dynamics to significantly improve robustness to outliers while getting a faster & online algorithm.
  - both theoretically (order-wise) and in practice.
Many datasets = low-rank + sparse

- Clean Netflix users’ data lies close to a low-dimensional subspace (users’ preferences governed by only a few factors) [Candes et al., 2009],
  - but is corrupted by data from lazy or malicious users (sparse outliers)

- Slow changing videos (e.g., video of moving waters) lie close to a low-dimensional subspace [Candes et al., 2009], but are often corrupted by foreground moving objects (occlusions)

- Foreground image sequence is usually sparse, but the background image sequence may not be sparse or easily sparsifiable;

- Social media users' connectivity patterns are often well-approximated by a low-dimensional tensor, but those of anomalous / outlier / suspicious users may not
Many datasets = \textit{low-rank} + sparse

- Clean Netflix users’ data lies close to a low-dimensional subspace (users’ preferences governed by only a few factors) [Candes et al, 2009],
  - but is corrupted by data from lazy or malicious users (sparse outliers)

- Slow changing videos (e.g., video of moving waters) lie close to a low-dimensional subspace [Candes et al, 2009],
  - but are often corrupted by foreground moving objects (occlusions)

- Foreground image sequence is usually sparse,
  - but the background image sequence may not be sparse or easily sparsifiable;
Many datasets = low-rank + sparse

- Clean Netflix users’ data lies close to a low-dimensional subspace (users’ preferences governed by only a few factors) [Candes et al., 2009],
  - but is corrupted by data from lazy or malicious users (sparse outliers)

- Slow changing videos (e.g., video of moving waters) lie close to a low-dimensional subspace [Candes et al., 2009],
  - but are often corrupted by foreground moving objects (occlusions)

- Foreground image sequence is usually sparse,
  - but the background image sequence may not be sparse or easily sparsifiable;

- Social media users’ connectivity patterns often well-approximated by a low-dimensional tensor,
  - but those of anomalous / outlier / suspicious users may not
PCA: find low-dimensional subspace that best approximates a given dataset
- first step before most data analytics’ tasks
- PCA is easy to solve: SVD on data matrix
- but, the SVD solution is very sensitive to outliers
Principal Components Analysis (PCA) and Robust PCA (RPCA)

- PCA: find low-dimensional subspace that best approximates a given dataset
  - first step before most data analytics’ tasks
  - PCA is easy to solve: SVD on data matrix
  - but, the SVD solution is very sensitive to outliers

- Robust PCA: problem of PCA in presence of outliers; classical problem, many heuristics exist for solving it
  - best old solution: Robust Subspace Learning (RSL) [de la Torre, Black, '03]
Principal Components Analysis (PCA) and Robust PCA (RPCA)

- **PCA**: find low-dimensional subspace that best approximates a given dataset
  - first step before most data analytics’ tasks
  - PCA is easy to solve: SVD on data matrix
  - but, the SVD solution is very sensitive to outliers

- **Robust PCA**: problem of PCA in presence of outliers; classical problem, many heuristics exist for solving it
  - best old solution: Robust Subspace Learning (RSL) [de la Torre, Black,’03]

- **Recent work** [Candes, Wright, Li, Ma, 2009] defined Robust PCA as the problem of separating a low-rank \( L \) and a sparse matrix \( X \) from

\[
Y := L + X
\]

- idea: outliers occur occasionally and usually on only a few data indices; their magnitude can be large - model as sparse corruptions

- Henceforth, RPCA = Sparse + Low-Rank Matrix Recovery
Sparse+Low-rank Recovery: separate low-rank $L$ and sparse $X$ from

$$Y := X + L$$

or from a subset of entries of $(X + L)$

- if $L$ or span($L$) is the quantity of interest: robust PCA
- if $X$ is quantity of interest: robust sparse recovery

Video analytics, e.g. for surveillance, tracking, mobile video chat, occlusion removal [Candes et al, 2009]

$$X = [x_1, x_2, \ldots, x_t, \ldots x_{t_{\text{max}}}], \quad L = [\ell_1, \ell_2, \ldots, \ell_t, \ldots \ell_{t_{\text{max}}}]$$

- $\ell_t$: background - usually slow changing,
- $x_t$: foreground - sparse, consists of one or more moving objects (technically $x_t$: (fg-bg) on fg support)
Dynamic Robust PCA

Background and Problem Formulation

original  background  foreground

original  background  foreground

original  background  foreground
Recommendation systems design (Netflix problem) [Candes et al’2009] (robust PCA with missing entries / robust matrix completion)

- $\ell_t$: ratings of movies by user $t$
- the matrix $L$ is low-rank: user preferences governed by only a few factors
- $x_t$: some users may enter completely incorrect ratings due to laziness or malicious intent or just typos: outliers
- goal: recover the matrix $L$ in order to recommend movies
Applications – 2

- **Recommendation systems design (Netflix problem)** [Candes et al’2009]
  (robust PCA with missing entries / robust matrix completion)
  - $\ell_t$: ratings of movies by user $t$
  - the matrix $L$ is low-rank: user preferences governed by only a few factors
  - $x_t$: some users may enter completely incorrect ratings due to laziness or malicious intent or just typos: outliers
  - goal: recover the matrix $L$ in order to recommend movies

- **Detecting anomalous connectivity patterns in social networks or in computer networks** [Mateos et al.,2011]
  - $\ell_t$: vector of n/w link “strengths” at time $t$ when no anomalous behavior
  - $x_t$: outliers or anomalies on a few links
Applications – 2

- **Recommendation systems design (Netflix problem) [Candes et al’2009]** (robust PCA with missing entries / robust matrix completion)
  - $\ell_t$: ratings of movies by user $t$
  - the matrix $L$ is low-rank: user preferences governed by only a few factors
  - $x_t$: some users may enter completely incorrect ratings due to laziness or malicious intent or just typos: outliers
  - goal: recover the matrix $L$ in order to recommend movies

- **Detecting anomalous connectivity patterns in social networks or in computer networks** [Mateos et al.,2011]
  - $\ell_t$: vector of n/w link “strengths” at time $t$ when no anomalous behavior
  - $x_t$: outliers or anomalies on a few links

- **Functional MRI based brain activity detection or other dynamic MRI based region-of-interest detection problems** [Otazo, Candes, et al. 2014]
  - only a sparse brain region activated in response to stimuli, everything else: very slow changes
Practical and Provably Correct Solutions to RPCA

- [Candes et al. 2009] introduced and studied a convex optimization program called Principal Components Pursuit (PCP):

  \[
  \min_{\tilde{X}, \tilde{L}} \|\tilde{L}\|_* + \lambda \|\tilde{X}\|_1 \quad \text{s.t.} \quad Y = \tilde{X} + \tilde{L}
  \]

  showed: PCP indeed “works” for real videos; and has a provable guarantee - first guarantee for any practical robust PCA approach.
Practical and Provably Correct Solutions to RPCA

- [Candes et al, 2009] introduced and studied a convex opt program called Principal Components Pursuit (PCP):

\[
\min_{\tilde{X}, \tilde{L}} \|\tilde{L}\|_* + \lambda \|\tilde{X}\|_1 \text{ s.t. } Y = \tilde{X} + \tilde{L}
\]

- showed: PCP indeed “works” for real videos; and has a provable guarantee - first guarantee for any practical robust PCA approach

- Parallel and later work on provably correct static RPCA:
  - PCP guarantee by [Chandrasekharan et al, 2009] – deterministic guarantee
  - Improved guarantee for PCP by [Hsu et al, 2011]
  - AltProj: provably correct Alternating Min [Netrapalli et al, NIPS’14]
  - RPCA-GD: provably correct Gradient Descent [Yi et al, NIPS’16]
Practical and Provably Correct Solutions to RPCA

- [Candes et al, 2009] introduced and studied a convex opt program called Principal Components Pursuit (PCP):

\[
\min_{\tilde{X}, \tilde{L}} \|\tilde{L}\|_* + \lambda \|\tilde{X}\|_1 \text{ s.t. } Y = \tilde{X} + \tilde{L}
\]

- showed: PCP indeed “works” for real videos; and has a provable guarantee - first guarantee for any practical robust PCA approach

Parallel and later work on provably correct static RPCA:

- PCP guarantee by [Chandrasekharan et al, 2009] – deterministic guarantee
- Improved guarantee for PCP by [Hsu et al, 2011]
- AltProj: provably correct Alternating Min [Netrapalli et al, NIPS’14]
- RPCA-GD: provably correct Gradient Descent [Yi et al, NIPS’16]

Our work on Dynamic Robust PCA [Qiu, Vaswani, Allerton, 2010] (algorithm), [Lois, Vaswani, ISIT’15] (guarantee)
Motivation for solving Dynamic RPCA

- **Limitations of static RPCA solutions:**
  1. slower and memory intensive
  2. need tight bounds on fraction of outliers per row or column of $X$
     - so that $X$ does not become rank deficient: needed to separate it from low rank $L$

- **The outlier fraction bound is often violated, e.g.,**
  - in video analytics: often have occasionally static or slow moving foreground (fg) objects: large outlier fractions per row
  - can also have large-sized fg objects: large outlier fractions per column
  - in network anomaly detection: anomalous behavior continues on most of the same edges for a period of time after begins

By **exploiting dynamics (slow subspace change)**, above limitations can be removed.
Figure: Slow moving person \(\Rightarrow\) sparse matrix \(X\) is also low rank \(\Rightarrow\) PCP confuses person for background. Proposed method (ReProCS) works because exploits dynamics.
Some definitions for rest of the talk

- $P'$ denotes transpose of matrix $P$
- $P$ is a basis matrix: $P$ is a tall matrix with mutually orthonormal columns
- Estimate $P$: estimate $\text{span}(P)$: subspace spanned by col's of $P$
- $\hat{P}$ is an accurate estimate of $P$: $\text{span}(\hat{P})$ is an accurate estimate of the $\text{span}(P)$
- Subspace Error (SE):

$$SE(\hat{P}, P) := \| (I - \hat{P}\hat{P}^t)P \|_2$$

measures the principal angle b/w subspaces spanned by $\hat{P}$ and $P$
Given sequentially arriving length $n$ data vectors $y_t$ satisfying

$$y_t := \ell_t + x_t, \quad t = 1, 2, \ldots, d$$

- $\ell_t$ lies in a fixed or slowly-changing low-dimensional subspace of $\mathbb{R}^n$;
- $\ell_t = P(t) a_t$, $P(t)$: $n \times r$ matrix with $r \ll n$, changes at most a “little” every so often,
- columns of $P(t)$ are dense vectors
- $x_t$: sparse outlier vector with support set $\mathcal{T}_t$;
Given sequentially arriving length $n$ data vectors $y_t$ satisfying

$$y_t := \ell_t + x_t, \quad t = 1, 2, \ldots, d$$

- $\ell_t$ lies in a fixed or slowly-changing low-dimensional subspace of $\mathbb{R}^n$;
  - $\ell_t = P(t)a_t$, $P(t)$: $n \times r$ matrix with $r \ll n$, changes at most a “little” every so often,
  - columns of $P(t)$ are dense vectors
- $x_t$: sparse outlier vector with support set $T_t$;

Goal: recursively estimate $x_t$, $\ell_t$, and $\text{span}(P(t))$, starting with initial estimate of $\text{span}(P(0))$

- initial subspace estimate: either assume outlier-free data available, or apply PCP or AltProj on $Y_{\text{init}} := [y_1, y_2, \ldots, y_{t_{\text{train}}}]$
Initial outlier-free sequence: easy to obtain in certain applications, e.g.,

- in video surveillance, easy to get a short background-only training sequence before foreground objects start appearing
- for fMRI, this corresponds to acquiring a short sequence without any activation
Initial outlier-free sequence: easy to obtain in certain applications, e.g.,

- in video surveillance, easy to get a short background-only training sequence before foreground objects start appearing
- for fMRI, this corresponds to acquiring a short sequence without any activation

When above not possible: use a batch method (e.g., PCP or AltProj) for first $t_{\text{train}}$ frames

works if assumptions needed by PCP or AltProj hold for $Y_{\text{init}} := [y_1, y_2, \ldots, y_{t_{\text{train}}}]$
Initial outlier-free sequence: easy to obtain in certain applications, e.g.,

- in video surveillance, easy to get a short background-only training sequence before foreground objects start appearing
- for fMRI, this corresponds to acquiring a short sequence without any activation

When above not possible: use a batch method (e.g., PCP or AltProj) for first $t_{\text{train}}$ frames

- works if assumptions needed by PCP or AltProj hold for $Y_{\text{init}} := [y_1, y_2, \ldots, y_{t_{\text{train}}}]$
Recall: $y_t := x_t + \ell_t$, $\ell_t = P(t)a_t$, $P(t)$: tall $n \times r$ basis matrix
Recursive Projected CS (ReProCS) [Qiu, Vaswani, Allerton'10, Allerton'11], [Guo, Qiu, Vaswani, T-SP'14]

Recall: $y_t := x_t + \ell_t$, $\ell_t = P(t)a_t$, $P(t)$: tall $n \times r$ basis matrix

Given $\hat{P}_0$. For $t > t_{\text{train}}$, do

1. Projection: compute $\tilde{y}_t := \Phi y_t$, where $\Phi := I - \hat{P}(t-1)\hat{P}(t-1)'$

   then $\tilde{y}_t = \Phi x_t + \beta_t$, $\beta_t := \Phi \ell_t$ is small "noise" because of slow subspace change
Recall: \( y_t := x_t + \ell_t, \ell_t = P(t)a_t, P(t) \): tall \( n \times r \) basis matrix

Given \( \hat{P}_0 \). For \( t > t_{\text{train}} \), do

1. **Projection**: compute \( \tilde{y}_t := \Phi y_t \), where \( \Phi := I - \hat{P}(t-1)\hat{P}(t-1)' \)
   - then \( \tilde{y}_t = \Phi x_t + \beta_t, \beta_t := \Phi \ell_t \) is small “noise” because of slow subspace change

2. **Noisy Compressive Sensing (CS)**: CoSaMP + support estimate + LS: get \( \hat{x}_t \)
   - denseness of columns of \( P(t) \) \( \Rightarrow \) sparse \( x_t \) recoverable from \( \tilde{y}_t \)
Recursive Projected CS (ReProCS) [Qiu, Vaswani, Allerton’10, Allerton’11], [Guo, Qiu, Vaswani, T-SP’14]

Recall: \( y_t := x_t + \ell_t \), \( \ell_t = P(t) a_t \), \( P(t) \): tall \( n \times r \) basis matrix

Given \( \hat{P}_0 \). For \( t > t_{\text{train}} \), do

1. **Projection:** compute \( \tilde{y}_t := \Phi y_t \), where \( \Phi := I - \hat{P}_{(t-1)} \hat{P}_{(t-1)}' \)
   
   then \( \tilde{y}_t = \Phi x_t + \beta_t \), \( \beta_t := \Phi \ell_t \) is small “noise” because of slow subspace change

2. **Noisy Compressive Sensing (CS): CoSaMP + support estimate + LS:** get \( \hat{x}_t \)
   
   denseness of columns of \( P(t) \) ⇒ sparse \( x_t \) recoverable from \( \tilde{y}_t \)

3. **Recover \( \ell_t \):** compute \( \hat{\ell}_t = y_t - \hat{x}_t \)
Recursive Projected CS (ReProCS) [Qiu, Vaswani, Allerton’10, Allerton’11], [Guo, Qiu, Vaswani, T-SP’14]

Recall: \( y_t := x_t + \ell_t \), \( \ell_t = P(t)a_t \), \( P(t) \): tall \( n \times r \) basis matrix

Given \( \hat{P}_0 \). For \( t > t_{\text{train}} \), do

1. **Projection:** compute \( \tilde{y}_t := \Phi y_t \), where \( \Phi := I - \hat{P}_{(t-1)} \hat{P}_{(t-1)}^{'} \)
   - then \( \tilde{y}_t = \Phi x_t + \beta_t \), \( \beta_t := \Phi \ell_t \) is small “noise” because of slow subspace change

2. **Noisy Compressive Sensing (CS):** CoSaMP + support estimate + LS: get \( \hat{x}_t \)
   - denseness of columns of \( P(t) \) \( \Rightarrow \) sparse \( x_t \) recoverable from \( \tilde{y}_t \)

3. **Recover \( \ell_t \):** compute \( \hat{\ell}_t = y_t - \hat{x}_t \)

4. **Subspace update:** use \( \hat{\ell}_t \)'s to update \( \hat{P}_{(t)} \) every \( \alpha \) frames
Why ReProCS works - intuition [Qiu, Vaswani, Lois, Hogben, T-IT, 2014]

- Slow subspace change $\Rightarrow$ noise $\beta_t := \Phi \ell_t$ seen by CS step small

- Denseness of columns of $P(t)$ and slow subspace change $\Rightarrow$ RIP constant of $\Phi := I - \hat{P}(t-1)\hat{P}(t-1)'$ small. Reason:

$$\delta_{2s}(I - PP') = \max_{|T| \leq 2s} \| I_T P \|_2^2$$ [Qiu, Lois, Vaswani, Hogben, T-IT'14]

- Above facts + CoSaMP guarantee $\Rightarrow x_t$ is accurately recovered; and hence $\ell_t = y_t - x_t$ is accurately recovered
Why ReProCS works - intuition [Qiu,Vaswani,Lois,Hogben,T-IT,2014] 4

- Slow subspace change ⇒ noise $\beta_t := \Phi \ell_t$ seen by CS step small
- Denseness of columns of $P(t)$ and slow subspace change ⇒ RIP constant of $\Phi := I - \hat{P}_{(t-1)} \hat{P}_{(t-1)}'$ small. Reason:
  \[
  \delta_{2s}(I - PP') = \max_{|T| \leq 2s} \|I_T'P\|_2^2 \quad [\text{Qiu, Lois, Vaswani, Hogben, T-IT'14}]
  \]
- Above facts + CoSaMP guarantee ⇒ $x_t$ is accurately recovered; and hence $\ell_t = y_t - x_t$ is accurately recovered
- Most of the work: show accurate subspace recovery $\hat{P}(t) \approx P(t)$
  - standard PCA results not applicable: $e_t := \hat{\ell}_t - \ell_t$ correlated with $\ell_t$
    - reason: $e_t = x_t - \hat{x}_t$ and this depends on $\beta_t := \Phi \ell_t$
  - all existing guarantees for PCA assume data, noise uncorrelated;

---

4 C. Qiu, N. Vaswani, B. Lois and L. Hogben, Recursive Robust PCA or Recursive Sparse Recovery in Large but Structured Noise, IEEE Trans. IT, 2014
Why ReProCS works - intuition [Qiu, Vaswani, Lois, Hogben, T-IT, 2014]

- Slow subspace change ⇒ noise $\beta_t := \Phi \ell_t$ seen by CS step small
- Denseness of columns of $P(t)$ and slow subspace change ⇒ RIP constant of $\Phi := I - \hat{P}_{(t-1)} \hat{P}_{(t-1)}'$ small. Reason:
  
  $\delta_{2s}(I - PP') = \max_{|T| \leq 2s} \| I_T' P \|^2_2$  [Qiu, Lois, Vaswani, Hogben, T-IT’14]

- Above facts + CoSaMP guarantee ⇒ $x_t$ is accurately recovered; and hence $\ell_t = y_t - x_t$ is accurately recovered
- Most of the work: show accurate subspace recovery $\hat{P}(t) \approx P(t)$
  - standard PCA results not applicable: $e_t := \hat{\ell}_t - \ell_t$ correlated with $\ell_t$
    - reason: $e_t = x_t - \hat{x}_t$ and this depends on $\beta_t := \Phi \ell_t$
  - all existing guarantees for PCA assume data, noise uncorrelated; except [Vaswani, Guo, Correlated-PCA, NIPS’16]

---

4 C. Qiu, N. Vaswani, B. Lois and L. Hogben, Recursive Robust PCA or Recursive Sparse Recovery in Large but Structured Noise, IEEE Trans. IT, 2014
**Figure:** A visualization of the ReProCS algorithm

```
\[ \hat{y}_t = (I - \hat{P}_{t-1} \hat{P}_{t-1}^\prime) y_t \]
```
Subspace update

- Toggles between “detect” phase and “estimate” phase
- In “detect” phase: detect change every $\alpha$ frames; suppose detected at $\hat{t}_j$

\[ \hat{P}_\ast \text{denote the subspace estimate from the last "estimate" phase} \]

"Estimate" phase: estimate changed subspace - "projection-SVD" repeated $K$ times

1st projection-SVD at $\hat{t}_j + \alpha$: let $\hat{P}_\ast \leftarrow \hat{t}_j + \alpha$; $\hat{P}_{ch,1} \leftarrow \text{top singular vector(s) of } \left( I - \hat{P}_\ast \hat{P}_\ast ' \right) [\hat{\ell}_t^\ast - \alpha, \hat{\ell}_t^\ast - \alpha + 1, ..., \hat{\ell}_t^\ast]$; use for projected-CS in next interval

2nd projection-SVD at $\hat{t}_j + 2\alpha$: let $\hat{P}_\ast \leftarrow \hat{t}_j + 2\alpha$; $\hat{P}_{ch,2} \leftarrow \text{top singular vector(s) of } \left( I - \hat{P}_\ast \hat{P}_\ast ' \right) [\hat{\ell}_t^\ast - \alpha, \hat{\ell}_t^\ast - \alpha + 1, ..., \hat{\ell}_t^\ast]$; use this for projected-CS in next interval

continue for $K$ steps; update $\hat{P}_\ast \leftarrow [\hat{P}_\ast, \hat{P}_{ch,K}]$

Simple SVD at $t = \hat{t}_j + K\alpha + \alpha$; Enter "detect" phase
**Subspace update**

- Toggles between “detect” phase and “estimate” phase
- In “detect” phase: detect change every $\alpha$ frames; suppose detected at $\hat{t}_j$
- Let $\hat{P}_*$ denote the subspace estimate from the last “estimate” phase
- “Estimate” phase: estimate changed subspace - “projection-SVD” repeated $K$ times
  - 1st projection-SVD at $\hat{t}_j + \alpha$: let $t_* \leftarrow \hat{t}_j + \alpha$
    - $\hat{P}_{ch,1} \leftarrow \text{top singular vector(s) of } (I - \hat{P}_*\hat{P}_*')[\hat{\ell}_{t_*-\alpha} , \hat{\ell}_{t_*-\alpha+1} , \ldots , \hat{\ell}_{t_*} ]$
    - $\hat{P}_{(t)} \leftarrow [\hat{P}_*, \hat{P}_{ch,1} ]; \text{ use for projected-CS in next interval}$
Subspace update

- Toggles between “detect” phase and “estimate” phase
- In “detect” phase: detect change every $\alpha$ frames; suppose detected at $\hat{t}_j$
- Let $\hat{P}_*$ denote the subspace estimate from the last “estimate” phase
- “Estimate” phase: estimate changed subspace - “projection-SVD” repeated $K$ times

  1st projection-SVD at $\hat{t}_j + \alpha$: let $t_* \leftarrow \hat{t}_j + \alpha$
  - $\hat{P}_{\text{ch},1} \leftarrow$ top singular vector(s) of $(I - \hat{P}_* \hat{P}_*^{'})[\hat{\ell}_{t_* - \alpha}, \hat{\ell}_{t_* - \alpha + 1}, \ldots, \hat{\ell}_{t_*}]$
  - $\hat{P}(t) \leftarrow [\hat{P}_*, \hat{P}_{\text{ch},1}]$; use for projected-CS in next interval

  2nd projection-SVD at $\hat{t}_j + 2\alpha$: let $t_* \leftarrow \hat{t}_j + 2\alpha$
  - $\hat{P}_{\text{ch},2} \leftarrow$ top singular vector(s) of $(I - \hat{P}_* \hat{P}_*^{'})[\hat{\ell}_{t_* - \alpha}, \hat{\ell}_{t_* - \alpha + 1}, \ldots, \hat{\ell}_{t_*}]$
  - $\hat{P}(t) \leftarrow [\hat{P}_*, \hat{P}_{\text{ch},2}]$; use this for projected-CS in next interval

  continue for $K$ steps; update $\hat{P}_* \leftarrow [\hat{P}_*, \hat{P}_{\text{ch},K}]$

- Simple SVD at $t = \hat{t}_j + K\alpha + \alpha$; Enter “detect” phase
Figure: Subspace Error $\log_{10} SE(\hat{P}_t, P_t)$ versus time - plotted at $t = t_j$, and at projection-SVD times, $t = t_j + k\alpha$.

- With each proj-SVD step, the subspace error decreases approx exponentially
  - better estimate of $P(t) \Rightarrow$ smaller noise $\beta_t$ seen by CS step in next $\alpha$-frame interval $\Rightarrow$ smaller CS step error $e_t := x_t - \hat{x}_t = \hat{\ell}_t - \ell_t \Rightarrow$ smaller perturbation seen at next proj-SVD step $\Rightarrow$ improved next estimate of $P(t)$
Subspace change model

- \( \ell_t = P_j a_t \) for all \( t \in [t_j, t_{j+1}) \) and \( P_j \) changes by
  - adding a new direction, \( P_{\text{new}} \), from \( \text{span}(P_j, \perp) \),
  - and rotating it in with angle \( \theta_j \), i.e.

\[
P_{j+1} = \begin{bmatrix} P_{j,\text{fix}} & P_{j,\text{ch}} \cos \theta_j - P_{\text{new}} \sin \theta_j \\ \end{bmatrix}_{n \times (r-1)} P_{j,\text{rot}} : n \times 1
\]
Subspace change model

- $\ell_t = P_j a_t$ for all $t \in [t_j, t_{j+1})$ and $P_j$ changes by
  - adding a new direction, $P_{\text{new}}$, from $\text{span}(P_j, \perp)$,
  - and rotating it in with angle $\theta_j$, i.e.

$$P_{j+1} = \begin{bmatrix} P_{j,\text{fix}} & P_{j,\text{ch}} \cos \theta_j - P_{\text{new}} \sin \theta_j \end{bmatrix}$$

$$P_{j,\text{rot}}: n \times 1$$

- $a_t$’s mutually independent, zero mean, bounded, with diagonal cov,

$$\Lambda = \begin{bmatrix} \Lambda_{\text{fix}} & 0 \\ 0 & \lambda_{\text{ch}} \end{bmatrix}.$$
Theorem

Let $\theta^- := \min_j \theta_j$, $\theta^+ := \max_j \theta_j$; $\kappa$: cond. #; $r$: subspace dim; $n$: data dim; If initial subspace estimate is accurate enough: $\zeta_0 := \text{SE}(\hat{P}_0, P_0)$ satisfies

$$\zeta_0 \kappa \leq 0.01 \sin \theta^+, \quad \zeta_0 \sqrt{r \kappa} \leq 0.1 \sin \theta^+$$

3 fraction of outliers in any column and in any row is bounded: outlier-fraction-col $\leq 0$, outlier-fraction-row $\leq 0$. If algorithm parameters appropriately set then, w.h.p., $\text{SE}(\hat{P}(t), P(t)) \leq \epsilon$ within at most $C(r \log n)(-\log \epsilon)$ frames;
ReProCS Performance Guarantee [Lois,Vaswani,ISIT,2015],[Zhan,Lois,Guo,Vaswani,ICML,2016]

Theorem

Let $\theta^- := \min_j \theta_j$, $\theta^+ := \max_j \theta_j$; $\kappa$: cond. #; $r$: subspace dim; $n$: data dim; If

1. initial subspace estimate is accurate enough: $\zeta_0 := \text{SE}(\hat{P}_0, P_0)$ satisfies
   $$\zeta_0 \kappa \leq 0.01 \sin \theta^+ , \quad \zeta_0 \sqrt{r \kappa} \leq 0.1 \sin \theta^+$$

2. subspace changes slowly enough:
   - $t_{j+1} - t_j \geq C(r \log n)(- \log \epsilon)$
   - and $\theta^+$ small enough: $33|\sin \theta^+| \sqrt{\lambda_{ch}} \leq x_{\min} - 0.02 x_{\max}$

($x_{\min}, x_{\max}$: min, max nonzero outlier magnitude; $\lambda_{ch}$: eigenvalue along changed direc)
**Theorem**

Let \( \theta^- := \min_j \theta_j, \theta^+ := \max_j \theta_j \); \( \kappa \): cond. \#; \( r \): subspace dim; \( n \): data dim; If

1. initial subspace estimate is accurate enough: \( \zeta_0 := SE(\hat{P}_0, P_0) \) satisfies
   \[
   \zeta_0 \kappa \leq 0.01 \sin \theta^+, \quad \zeta_0 \sqrt{r \kappa} \leq 0.1 \sin \theta^+
   \]

2. subspace changes slowly enough:
   - \( t_{j+1} - t_j \geq C(r \log n)(-\log \epsilon) \)
   - \( \theta^+ \) small enough: 
     \[
     33 |\sin \theta^+| \sqrt{\lambda_{ch}} \leq x_{\min} - 0.02 x_{\max}
     \]
     \( (x_{\min}, x_{\max}: \min, \max \text{ nonzero outlier magnitude; } \lambda_{ch}: \text{eigenvalue along changed direc}) \)

3. fraction of outliers in any column and in any row is bounded:
   \[
   \text{outlier-fraction-col} \leq \frac{0.09}{\mu r} \quad \text{and} \quad \text{outlier-fraction-row} \leq 0.01
   \]
Theorem

Let $\theta^- := \min_j \theta_j$, $\theta^+ := \max_j \theta_j$; $\kappa$: cond. #; $r$: subspace dim; $n$: data dim; If

1. initial subspace estimate is accurate enough: $\zeta_0 := SE(\hat{P}_0, P_0)$ satisfies

$$\zeta_0 \kappa \leq 0.01 \sin \theta^+, \quad \zeta_0 \sqrt{r \kappa} \leq 0.1 \sin \theta^+$$

2. subspace changes slowly enough:
   - $t_{j+1} - t_j \geq C(r \log n)(- \log \epsilon)$
   - and $\theta^+$ small enough: $33 |\sin \theta^+| \sqrt{\lambda_{ch}} \leq x_{\min} - 0.02 x_{\max}$
     ($x_{\min}, x_{\max}$: min, max nonzero outlier magnitude; $\lambda_{ch}$: eigenvalue along changed direc)

3. fraction of outliers in any column and in any row is bounded:

$$\text{outlier-fraction-col} \leq \frac{0.09}{\mu r} \quad \text{and outlier-fraction-row} \leq 0.01$$

4. algorithm parameters appropriately set

then, w.h.p., $SE(\hat{P}(t), P(t)) \leq \epsilon$ within at most $C(r \log n)(- \log \epsilon)$ frames;
Detailed conclusions

Under theorem's assumptions, with probability at least $1 - 22dn^{-10}$,

1. outlier support is exactly recovered ($\hat{T}_t = T_t$) at all times $t$;
2. change gets detected within at most $2\alpha = C(r \log n)$ frames;
3. $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq 0.25|\sin \theta^+| \sqrt{\lambda_{ch}}$ at all times;
4. offline: $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq 2.4\epsilon \|\ell_t\|_2$;
Detailed conclusions

Under theorem’s assumptions, with probability at least $1 - 22dn^{-10}$,

1. outlier support is exactly recovered ($\hat{T}_t = T_t$) at all times $t$;
2. change gets detected within at most $2\alpha = C(r \log n)$ frames;
3. $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq 0.25|\sin \theta^+|\sqrt{\lambda_{ch}}$ at all times;
4. offline: $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq 2.4\epsilon\|\ell\|_2$;
5. $SE(\hat{P}, P)$ and $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2$ decay roughly exponentially with each proj-SVD step
   - for $t \in [t_j, \hat{t}_j + \alpha)$, $SE(\hat{P}_t, P_t) \leq 2\zeta_0 + \sin \theta^+$,
   - for $t \in [\hat{t}_j + (k - 1)\alpha, \hat{t}_j + k\alpha)$,
     $SE(\hat{P}_t, P_t) \leq 1.9\zeta_0 + (0.3)^{k-1} \cdot 0.006 \sin \theta^+$, for $k = 1, 2, \ldots, K$
   - for $t \in [\hat{t}_j + K\alpha, \hat{t}_j + K\alpha + \alpha_{del})$, $SE(\hat{P}_t, P_t) \leq 2\zeta_0$
   - for $t \in [\hat{t}_j + K\alpha + \alpha_{del}, t_{j+1})$, $SE(\hat{P}_t, P_t) \leq \zeta_0$
Above result is new; is a significant simplification of the results from AISTATS’16 or ISIT’15

- taps into the simplifications introduced in [Vaswani, Guo, Correlated-PCA, NIPS’16] while studying the general correlated-PCA problem

Proof uses

- Davis-Kahan sin $\theta$ theorem (1970)
  - bounds subspace error b/w space of top $r$ eigenvectors of a given symmetric matrix and that of its perturbed version

- Matrix Bernstein inequality
**Comparison with guarantees for static RPCA: assumptions**

<table>
<thead>
<tr>
<th></th>
<th>PCP</th>
<th>AltProj</th>
<th>GD</th>
<th>ReProCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>outlier-fraction-row $\leq$</td>
<td>$\frac{c}{r_{\text{mat}}}$</td>
<td>$\frac{c}{r_{\text{mat}}}$</td>
<td>$\frac{1}{\mu\sqrt{r_{\text{mat}}^3}}$</td>
<td>$c$</td>
</tr>
<tr>
<td>outlier-fraction-col $\leq$</td>
<td>$\frac{c}{r_{\text{mat}}}$</td>
<td>$\frac{c}{r_{\text{mat}}}$</td>
<td>$\frac{1}{\mu\sqrt{r_{\text{mat}}^3}}$</td>
<td>$c/r$</td>
</tr>
<tr>
<td>slow subspace change</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>initial data $Y_{\text{init}}$</td>
<td></td>
<td></td>
<td>assumptions of AltProj: incoherence, outlier-fraction $\leq c/r$</td>
<td></td>
</tr>
<tr>
<td>algo parameters</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

**Table:** An $n \times d$ data matrix $Y := L + X$; rank of $L$ is $r_{\text{mat}} = r + J$. $r$ is the subspace dimension at any time, $r \leq r_{\text{mat}}$. Above: $\kappa$ is assumed constant, ignored.

**Streaming RPCA** [Niranjan, Shi, ArXiv, Dec’16]: only works for $r_{\text{mat}} = r = 1$. 
Comparison with guarantees for static RPCA: time, storage complexity

<table>
<thead>
<tr>
<th></th>
<th>PCP</th>
<th>AltProj</th>
<th>GD</th>
<th>ReProCS</th>
<th>S-RPCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>$O(nd^2\frac{1}{\epsilon})$</td>
<td>$O(ndr_{mat}^2 \log \frac{1}{\epsilon})$</td>
<td>$O(ndr_{mat} \log \frac{1}{\epsilon})$</td>
<td>$O(n dr \log \frac{1}{\epsilon})$</td>
<td>$O(n dr \log \frac{1}{\epsilon})$</td>
</tr>
<tr>
<td>Storage</td>
<td>$O(nd)$</td>
<td>$O(nd)$</td>
<td>$O(nd)$</td>
<td>$O(nr (\log n))$</td>
<td>$O(nr_{mat})$</td>
</tr>
</tbody>
</table>

Table: *Time and storage complexity comparison for an $n \times d$ data matrix $Y := L + X$; rank of $L$ is $r_{mat}$. Above: $\kappa$ is assumed constant, ignored*

Observe

- ReProCS has the best time complexity: it is $\frac{r}{r_{mat}}$ times that of GD
- Its storage complexity is only $(\log n)$ times worse than the optimal - $O(nr)$ - achieved by streaming RPCA, but streaming RPCA only works for $r = 1$
Discussion - Pros and Cons of ReProCS

Pros

- Allows video objects that move every so often or move very slowly
  - tolerates outlier-fraction-row \( \leq c \); others need \( \leq \frac{c}{r_{\text{mat}}} \) \( (r_{\text{mat}} = \text{rank}(L)) \)
- Typically, also allows larger-sized foreground objects than other methods
  - tolerates outlier-fraction-col \( \leq \frac{c}{r_{\text{mat}}} \); others need \( \frac{c}{r_{\text{mat}}} \)
  - e.g., if \( r = O(\log n) \), but \( J = O(n) \), then \( r_{\text{mat}} = r + J = O(n) \): ReProCS works, others fail
- ReProCS is the fastest; has nearly optimal storage complexity; and is online

Cons:

- Needs the slow subspace change assumption
- Needs to know a few (5) model parameters to set algorithm parameters (so does RPCA-GD)

\[ r_{\text{mat}} := \text{rank}(L) = r + J, \quad r := \text{rank}(L_{[t_j, t_{j+1}]}) \], typical: \( \kappa \ll r \ll r_{\text{mat}} \), e.g., \( \kappa = O(1), \quad r = O(\log n), \quad r_{\text{mat}} = O(n) \)
First set of complete guarantees for any online / dynamic / streaming RPCA solution.
First set of complete guarantees for any online / dynamic / streaming RPCA solution.

- partial results (req. assumptions on intermediate algo. estimates): ReProCS [Qiu, Vaswani, Lois, Hogben, ISIT’13, T-IT’14] and ORPCA [Feng et al., NIPS’13]

- complete guarantee: ReProCS [Lois, Vaswani, ICASSP’15, ISIT’15], [Zhan, Lois, Guo, Vaswani, AISTATS’16]

- complete guarantee for a streaming algorithm for static RPCA: [Niranjan, Shi, ArXiv, Dec’16] holds only for $r = 1$ case

New proof techniques needed to be developed

- useful for various other problems, e.g., correlated-PCA [Vaswani, Guo, NIPS’16]
Simulation Experiments

Compare

- ReProCS - [Qiu, Vaswani, Allerton 2010], [Qiu et al., T-IT’14],
- GRASTA - [He, Balzano, et al, CVPR 2012] – online algorithm
- ORPCA - [Feng, Xu, et al, NIPS 2013] – online algorithm to solve PCP
Simulation Experiments

Compare

- **ReProCS** - [Qiu, Vaswani, Allerton 2010], [Qiu et al., T-IT’14],
- **GRASTA** - [He, Balzano, et al, CVPR 2012] – online algorithm
- **ORPCA** - [Feng, Xu, et al, NIPS 2013] – online algorithm to solve PCP
- **offline-ReProCS** (allowed to go back and improve previous estimates) - [Zhan, Lois, Guo, Vaswani, AISTATS’16]
- **PCP (IALM)** – batch algo. for static RPCA - convex opt.; provably correct
- **AltProj** – batch algo. for static RPCA - Alt-Min; provably correct
- **RPCA-GD** – batch algo. for static RPCA - Grad. Desc.; provably correct
Figure: Comparisons for a simulated slow moving foreground object case (large outlier fraction per row): outlier frac per col \((s/n)\): 0.09, outlier frac per row \((b_0)\): 0.1 for first \(t_{\text{train}}\) frames, 0.7 after that; \(n = 500, r = 25, \kappa = 25, J = 2\): \(t_1 = 1000, t_2 = 2000, \theta_1 = \theta_2 = 30^\circ\). Offline ReProCS: improved ReProCS estimates by offline processing. ReProCS initialized using AltProj for \(t_{\text{train}} = 500\); used \(\alpha = 200\)
Dynamic Robust PCA
Experiments - simulation and real-data (video analytics)

<table>
<thead>
<tr>
<th>$n$</th>
<th>ReProCS (Offline)</th>
<th>GRASTA</th>
<th>ORPCA</th>
<th>AltProj</th>
<th>GD</th>
<th>PCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.0005 (0.0008)</td>
<td>0.0003</td>
<td>0.0009</td>
<td>0.0116</td>
<td>0.0226</td>
<td>0.0051</td>
</tr>
<tr>
<td>8000</td>
<td>0.24 (0.31)</td>
<td>–</td>
<td>0.16</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**Table:** Time comparisons (in seconds). Time per frame. When $n = 8000$, PCP, AltProj, GD: out of memory

**Conclusion:**

- By exploiting dynamics (slow subspace change)
  - ReProCS can tolerate *much larger outlier fractions per row*, and
  - it is also *much faster & memory-efficient* than all batch methods

- Online methods (ORPCA, GRASTA) do not work for large outlier fractions; do not have provably guarantees
Applications being explored

- **Video Analytics**
  - Video foreground tracking – video surveillance application
  - Background recovery and subspace tracking – needed to simulate realistic video textures [Dynamic Textures, Soatto et al, ICCV 2001]
  - Video denoising
    - with Rockwell Collins
  - Video enhancement “seeing in the dark”
    - with Rockwell Collins

- Detecting anomalous connectivity patterns in social networks data on-the-fly using Tensor-ReProCS
  - work of Selin Aviyente et al. at Michigan State (inspired by ReProCS); ongoing discussion about joint work
Practical ReProCS for the video experiments

- Used heuristics to estimate model parameters on-the-fly (to set algorithm parameters)
- Also exploited slow support change of the foreground object(s) when possible
- Most of the results shown here (except the video denoising ones) used an initial background-only sequence to initialize
  - same sequence also provided to GRASTA
Video surveillance application - foreground recovery

Figure: Foreground recovery \((t = t_{\text{train}} + 35, 500, 1300)\)
Background recovery and subspace tracking - useful for simulating video textures

**Figure:** Background recovery for modeling \((t = t_{train} + 30, 80, 140)\).
**Video denoising of very noisy videos**

- **Idea:** large variance noise can always be split as frequently occurring small noise and occasionally occurring large outliers.

- **Approach:**
  - use ReProCS to get $\hat{x}_t$ and $\hat{\ell}_t$ for each frame $t$
  - apply a state-of-art denoiser, VBM-3D, to each layer separately
  - use denoised $\hat{\ell}_t$ in most cases; sometimes use denoised image (add up denoised layers)

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>ReProCS-LD</th>
<th>PCP-LD</th>
<th>AltProj-LD</th>
<th>GRASTA-LD</th>
<th>VBM3D</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>32.78 (73)</td>
<td>32.84 (198)</td>
<td>31.98 (101)</td>
<td>28.11 (59)</td>
<td>32.02 (24)</td>
<td>28.26 (477)</td>
</tr>
<tr>
<td>50</td>
<td>32.27 (73)</td>
<td>31.65 (195)</td>
<td>30.09 (128)</td>
<td>23.97 (58)</td>
<td>27.99 (24)</td>
<td>18.87 (477)</td>
</tr>
<tr>
<td>70</td>
<td>31.79 (69)</td>
<td>30.67 (197)</td>
<td>29.63 (133)</td>
<td>21.81 (55)</td>
<td>24.42 (21)</td>
<td>15.03 (478)</td>
</tr>
</tbody>
</table>

**Table:** Comparison of denoising performance on waterfall dataset ($n = 108 \times 192$, $d = 650$) corrupted by Gaussian $\mathcal{N}(0, \sigma^2)$ noise. Displaying PSNR (run time in seconds). VBM-3D: best denoising algorithm; MLP: multi-layer perceptron (neural network based method). **ReProCS-LD is fast-enough & achieves a 1dB improvement over other approaches in case of large variance noise.**
<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>ReLD</th>
<th>VBM3D</th>
<th>MLP</th>
<th>SLMA</th>
<th>ReLD</th>
<th>VBM3D</th>
<th>MLP</th>
<th>SLMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>32.67(16.70)</td>
<td>31.18(5.44)</td>
<td>26.86(105.64)</td>
<td>22.93(3.05 × 10^4)</td>
<td>31.01(16.64)</td>
<td>30.32(5.34)</td>
<td>25.53(107.51)</td>
<td>21.17(3.09 × 10^4)</td>
</tr>
<tr>
<td>30</td>
<td>32.25(15.84)</td>
<td>30.26(5.17)</td>
<td>25.67(107.41)</td>
<td>21.85(3.06 × 10^4)</td>
<td>30.27(16.45)</td>
<td>29.29(5.38)</td>
<td>24.54(108.65)</td>
<td>20.49(3.15 × 10^4)</td>
</tr>
<tr>
<td>50</td>
<td>30.53(15.82)</td>
<td>26.55(5.24)</td>
<td>18.53(109.79)</td>
<td>18.55(3.13 × 10^4)</td>
<td>27.84(16.03)</td>
<td>25.10(5.27)</td>
<td>18.83(109.40)</td>
<td>17.98(3.21 × 10^4)</td>
</tr>
<tr>
<td>70</td>
<td>27.53(15.03)</td>
<td>22.08(4.69)</td>
<td>14.85(107.52)</td>
<td>16.25(3.19 × 10^4)</td>
<td>25.15(15.28)</td>
<td>20.20(4.72)</td>
<td>15.20(108.78)</td>
<td>15.90(3.18 × 10^4)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>ReProCS-LD</th>
<th>VBM3D</th>
<th>MLP</th>
<th>SLMA</th>
<th>ReLD</th>
<th>VBM3D</th>
<th>MLP</th>
<th>SLMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>35.47(16.78)</td>
<td>34.60(4.15)</td>
<td>31.14(189.14)</td>
<td>23.28(7.75 × 10^4)</td>
<td>39.78(57.96)</td>
<td>35.00(19.57)</td>
<td>29.22(384.11)</td>
<td>23.43(3.75 × 10^5)</td>
</tr>
<tr>
<td>30</td>
<td>34.58(17.35)</td>
<td>33.59(4.37)</td>
<td>28.90(191.14)</td>
<td>22.74(9.05 × 10^4)</td>
<td>38.76(57.99)</td>
<td>33.64(19.09)</td>
<td>27.72(395.67)</td>
<td>21.15(3.82 × 10^5)</td>
</tr>
<tr>
<td>50</td>
<td>31.91(17.17)</td>
<td>30.29(4.42)</td>
<td>18.58(188.30)</td>
<td>19.12(7.86 × 10^4)</td>
<td>35.15(58.41)</td>
<td>29.23(19.35)</td>
<td>18.66(403.59)</td>
<td>18.21(3.99 × 10^5)</td>
</tr>
<tr>
<td>70</td>
<td>28.10(16.50)</td>
<td>26.15(3.85)</td>
<td>14.73(192.00)</td>
<td>16.68(8.30 × 10^4)</td>
<td>29.68(56.51)</td>
<td>24.90(17.00)</td>
<td>14.85(401.29)</td>
<td>16.82(4.09 × 10^5)</td>
</tr>
</tbody>
</table>

**Table:** PSNR (run time in seconds) for 4 different datasets. VBM-3D: best denoising algorithm; MLP: multi-layer perceptron (neural network based method). SLMA: another sparse + low-rank method for denoising.
Low-light video enhancement: “seeing in the dark”

Figure: Original, V-BM-3D, K-SVD, ReProCS. In the video, a person is walking through a hallway. ReProCS successfully “sees” the person.
Dynamic Robust PCA

Experiments - simulation and real-data (video analytics)

Related Work

- **Batch RPCA:**
  - **RSL** [de la Torre et al., IJCV’03], **PCP** (Candès et al., 2011; Hsu et al., 2011), **AltProj** (Netrapalli et al., 2014), **GD** (Yi et al., 2016), ...

- **Dynamic, Online or Streaming RPCA or Robust Subspace Tracking**
  - **iRSL** (Skocaj & Leonardis, 2003): *does not work*
  - Recursive Projected Compressive Sensing (ReProCS): (Qiu & Vaswani, 2010)
  - **GRASTA** (He et al., 2012)
  - robust subspace tracking: (Chouvardas et al., 2015, 2014), (Mansour & Jiang, 2015)
  - online RPCA via stoch. opt. (Feng et al., 2013)
  - (Mardani et al., 2013): *batch and online; online: not enough info, no code*
  - streaming RPCA (Niranjan & Shi, 2016)

- **Guarantees**
  - **ReProCS partial guarantee** [Qiu, Vaswani, Lois, Hogben, ICASSP’13, ISIT’13, T-IT’14] (Qiu et al., 2014)
  - Online RPCA via stoch. opt. partial guarantee [Feng et al., NIPS’13]
  - **ReProCS complete guarantee** [Zhan, Lois, Guo, Vaswani, AISTATS’16, Lois, Vaswani, ISIT’15, ICASSP’15]
  - streaming RPCA complete guarantee for $r = 1$ case (Niranjan & Shi, 2016)
Low Rank Phase Retrieval (LRPR) [Vaswani, Nayer, Eldar, T-SP, 2017, to appear]

Problem:

- Recover a low rank matrix $X$ from magnitude-only measurements of linear projections of each of its columns
- $X$ is an $n \times q$ matrix with rank $r \ll \min(n, q)$; have $m$ measurements of each column of $X$
- Useful for PR of a sequence of images that change gradually over time, e.g., for dynamic solar imaging in astronomy, dynamic sub-diffraction imaging,...

Contributions so far

- Two novel iterative algorithms: LRP1 and LRPR2
  - each column of $X$ belongs to same $r$ dimensional subspace;
  - have (nearly) $mq$ measurements to recover this subspace
  - if subspace known, recovering each coefficient vector: easy PR problem
- Exciting preliminary experimental results
  - on real videos with simulated coded diffraction pattern measurements
- High probability sample complexity bounds for their initialization step.
Low Rank Phase Retrieval

- Phase Retrieval (PR): recover a signal/vector $x$ from magnitude-only (phaseless) measurements of its random linear projections, i.e., from $y_i := |a_i'x|^2$, $i = 1, 2, \ldots, m$.

- Low rank PR: recover a low-rank matrix, $X$, from phaseless measurements of random linear projections of its columns
  - we have a set of $q$ vectors, $x_1, x_2, \ldots, x_q$ which are such that the $n \times q$ matrix $X := [x_1, x_2, \ldots, x_q]$ has rank $r \ll \min(n, q)$;
  - for each $x_k$, there are a set of $m$ measurements of the form $y_{i,k} := |a_{i,k}'x_k|^2$, $i = 1, 2, \ldots m$, $k = 1, 2, \ldots, q$.
Key idea of proposed algorithms: LRPR1 and LRPR2

- Use the fact that a low rank matrix $X$ can be factored as $X = UB$: $U$: $n \times r$, $r \ll n$
  - each vector $x_k = Ub_k$, $k = 1, 2, \ldots, q$: all vectors share the same subspace
  - thus, for recovering $\text{span}(U)$, we have “nearly” $mq$ measurements
    $$y_{i,k} := |a_i,k'x_k|^2, \quad i = 1, 2, \ldots m, \quad k = 1, 2, \ldots, q$$
- can show that
  $$\mathbb{E} \left[ \frac{1}{mq} \sum_k \sum_i y_{i,k} a_i,k a_i,k' \right] = 2UU' + cl, \quad \Lambda \text{ is diagonal}$$
- We show that $U$ recovered using above idea satisfies $\text{SE}(\hat{U}, U) \leq \epsilon$ if
  $$mq \geq nr^2 \frac{1}{\epsilon^2}$$
- Once $U$ recovered, recovering each $b_k$ is a $r$-dimensional regular PR problem: easy since $r \ll n$
Figure: Recovering a real video from coded diffraction pattern (CDP) measurements. First column: frames 1, 50 and 104, of the original plane video. Next three columns: frames recovered using the various methods from $m = 3n$ CDP measurements. TWF: Truncated Wirtinger Flow [Chen, Candes, NIPS’15], TWFproj: projected TWF output at initialization and each iteration to space of rank $r$ matrices. LRPR1 and LRPR2: proposed algo’s.
Figure: This figure shows the power of LRPR2 for recovering a real video from coded diffraction pattern (CDP) measurements. First column: frames 2, 53 and 102, of the original bacteria video. Next three columns: frames recovered using the various methods from $m = 3n$ CDP measurements. TWF: Truncated Wirtinger Flow [Chen, Candes, NIPS’15], TWFproj: projected TWF output at initialization and each iteration to space of rank $r$ matrices. LRPR1 and LRPR2: proposed algo’s.
Open Questions

- **Dynamic Robust PCA:**
  1. more general subspace change models
  2. extensions to dynamic robust matrix completion, undersampled RPCA
  3. applications in
     - functional MRI based brain activity pattern tracking;
     - tracking user preferences over time
  4. dynamic subspace clustering?

- **Low Rank Phase Retrieval (LRPR) and Dynamic LRPR**
  1. performance guarantee for the complete LRPR algorithm
  2. speed-up & applications
  3. dynamic LRPR: use dynamics (slow subspace change) to
     - further reduce sample complexity, or deal with outliers or both

- **Correlated-PCA: PCA when data and noise are correlated** [Vaswani, Guo, NIPS, 2016]
  1. ongoing
PCA w/ Correlated Data and Noise [Vaswani,Guo,NIPS'16, Correlated-PCA],
[Vaswani,Narayanamurthy,arXiv'17]

- For $t = 1, 2, \ldots, \alpha$, we are given $n$-length data vectors,

$$y_t := \ell_t + w_t + v_t,$$

where $\ell_t = Pa_t$, $w_t = Mt\ell_t$,

where
- $P$ is an $n \times r$ matrix with orthonormal columns and $r \ll n$;
- $\ell_t$ is the true data vector that lies in $\text{span}(P)$;
- $w_t$ is the data-dependent (correlated) noise component; and
- $v_t$ is the uncorrelated noise, i.e., $\mathbb{E}[\ell_t v_t'] = 0$. 
PCA w/ Correlated Data and Noise \cite{VaswaniGuoNIPS16,Correlated-PCA}, \cite{VaswaniNarayanamurthyarXiv17}

For $t = 1, 2, \ldots, \alpha$, we are given $n$-length data vectors,

$$y_t := \ell_t + w_t + v_t, \text{ where } \ell_t = Pa_t, \ w_t = M_t \ell_t,$$

where

- $P$ is an $n \times r$ matrix with orthonormal columns and $r \ll n$;
- $\ell_t$ is the true data vector that lies in span($P$);
- $w_t$ is the data-dependent (correlated) noise component; and
- $v_t$ is the uncorrelated noise, i.e., $\mathbb{E}[\ell_t v_t'] = 0$.

The matrices $M_t$ are unknown and such that $\mathbb{E}[\ell_t w_t'] \neq 0$

Observe: in general, $w_t$’s do not lie in a lower dim subspace of $\mathbb{R}^n$. 

Examples: subspace update step of ReProCS; static robust PCA when outlier values are data-dependent; interference due to signal leakage

Almost all existing work that studies the SVD solution: assumes data and noise are either independent or, at least, uncorrelated
For \( t = 1, 2, \ldots, \alpha \), we are given \( n \)-length data vectors,

\[
y_t := \ell_t + w_t + v_t, \quad \text{where } \ell_t = Pa_t, \quad w_t = M_t \ell_t,
\]

where

- \( P \) is an \( n \times r \) matrix with orthonormal columns and \( r \ll n \);
- \( \ell_t \) is the true data vector that lies in \( \text{span}(P) \);
- \( w_t \) is the data-dependent (correlated) noise component; and
- \( v_t \) is the uncorrelated noise, i.e., \( \mathbb{E}[\ell_t v'_t] = 0 \).

The matrices \( M_t \) are unknown and such that \( \mathbb{E}[\ell_t w'_t] \neq 0 \).

Observe: in general, \( w_t \)'s do not lie in a lower dim subspace of \( \mathbb{R}^n \).

Examples: subspace update step of ReProCS; static robust PCA when outlier values are data-dependent; interference due to signal leakage.
**PCA w/ Correlated Data and Noise** [Vaswani,Guo,NIPS'16, Correlated-PCA], [Vaswani,Narayanamurthy,arXiv'17]

- For $t = 1, 2, \ldots, \alpha$, we are given $n$-length data vectors,

$$y_t := \ell_t + w_t + v_t, \text{ where } \ell_t = Pa_t, \ w_t = Mt\ell_t,$$

where

- $P$ is an $n \times r$ matrix with orthonormal columns and $r \ll n$;
- $\ell_t$ is the true data vector that lies in span($P$);
- $w_t$ is the data-dependent (correlated) noise component; and
- $v_t$ is the uncorrelated noise, i.e., $E[\ell_tv_t'] = 0$.

- The matrices $M_t$ are unknown and such that $E[\ell_tw_t'] \neq 0$

- Observe: in general, $w_t$‘s do not lie in a lower dim subspace of $\mathbb{R}^n$.

- Examples: subspace update step of ReProCS; static robust PCA when outlier values are data-dependent; interference due to signal leakage

- Almost all existing work that studies the SVD solution: assumes data and noise are either independent or, at least, uncorrelated
Simplified version of our main result

Theorem ($v_t = 0$)

Assume that $y_t = \ell_t + w_t$ where

- $\ell_t = Pa_t$ with $a_t$'s zero mean, mutually independent, and bounded r.v.'s, with diagonal covariance matrix, $\Lambda$; and
- $w_t := M_t\ell_t$ and $M_t$ can be split as $M_t = M_{2,t}M_{1,t}$ s.t. for a $q < 1$, $\|M_{1,t}P\| \leq q$, $\|M_{2,t}\| \leq 1$; and for a $b_0 < 1$, $\|\frac{1}{\alpha} \sum_t M_{2,t}M_{2,t}'\| \leq b_0$.

For an $\varepsilon_{SE} < 1$, define

$$\alpha_0 := C\eta \frac{q^2\kappa^2}{\varepsilon_{SE}^2} (r \log n).$$

For an $\alpha \geq \alpha_0$, let $\hat{P}$ be top $r$ left singular vectors of $\sum_{t=1}^{\alpha} y_t y_t' / \alpha$. If

$$3.3 \sqrt{b_0} q \kappa \leq 0.49 \varepsilon_{SE},$$

then, w.p. at least $1 - 6n^{-10}$, $\text{SE}(\hat{P}, P) := \|(I - \hat{P}\hat{P}')P\|_2 \leq \varepsilon_{SE}$.
Discussion

- Nearly optimal sample complexity
- to estimate an $r$-dimensional subspace, one needs at least $r$ samples
- if $\kappa$ is $O(1)$, $\alpha \geq C \kappa^2 (r \log n) \frac{q^2}{\varepsilon_{SE}^2}$ is only $(\log n)$ times the best achievable
Discussion

- Nearly optimal sample complexity
  - to estimate an \( r \)-dimensional subspace, one needs at least \( r \) samples
  - if \( \kappa \) is \( O(1) \), \( \alpha \geq C \kappa^2 (r \log n) \frac{q^2}{\varepsilon^2} \) is only \( (\log n) \) times the best achievable

- Correlated noise case is harder
  - bound on \( SE(\hat{P}, P) \) is governed by \( \frac{||H||}{\lambda} \) where
    \( H := \frac{1}{\alpha} \sum_t y_t y_t' - \frac{1}{\alpha} \sum_t \ell_t \ell_t' \) is the perturbation matrix
  - dominant terms in \( H \) are \( \frac{1}{\alpha} \sum_t \ell_t w_t' \) and its transpose:
    - as a result \( \frac{||H||}{\lambda} \) depends on \( \kappa = \frac{\lambda^+}{\lambda^-} \) and
Discussion

- Nearly optimal sample complexity
  - to estimate an \( r \)-dimensional subspace, one needs at least \( r \) samples
  - if \( \kappa \) is \( O(1) \), \( \alpha \geq C\kappa^2(r \log n)\frac{q^2}{\varepsilon^2 \text{SE}} \) is only \((\log n)\) times the best achievable
- Correlated noise case is harder
  - bound on \( \text{SE}(\hat{P}, P) \) is governed by \( \frac{||H||}{\lambda} \) where \( H := \frac{1}{\alpha} \sum_t y_t y_t' - \frac{1}{\alpha} \sum_t \ell_t \ell_t' \) is the perturbation matrix
  - dominant terms in \( H \) are \( \frac{1}{\alpha} \sum_t \ell_t w_t' \) and its transpose:
    - as a result \( \frac{||H||}{\lambda} \) depends on \( \kappa = \frac{\lambda^+}{\lambda^-} \) and
    - it is larger than it is in the uncorrelated-noise-only case
Dynamic CS: Problem [Vaswani, ICIP’08]\(^5\)

- Given measurements
  \[ y_t := A x_t + w_t, \quad \|w_t\|_2 \leq \epsilon, \quad t = 0, 1, 2, \ldots \]

  - \( A = H \Phi \) (given): \( n \times m, n < m \)
    - \( H \): measurement matrix, \( \Phi \): sparsity basis matrix
    - e.g., in MRI: \( H = \) partial Fourier, \( \Phi = \) inverse wavelet

  - \( y_t \): measurements (given)
  - \( x_t \): sparsity basis vector
  - \( N_t \): support set of \( x_t \)
  - \( w_t \): small noise

- Goal: recursively reconstruct \( x_t \) from \( y_0, y_1, \ldots y_t \),

---

\(^5\) N. Vaswani, Kalman Filtered Compressed Sensing, ICIP, 2008
Dynamic CS: Problem [Vaswani, ICIP’08] \(^5\)

- Given measurements
  \[ y_t := A x_t + w_t, \quad \|w_t\|_2 \leq \epsilon, \quad t = 0, 1, 2, \ldots \]

- \( A = H \Phi \) (given): \( n \times m, \quad n < m \)
  - \( H \): measurement matrix, \( \Phi \): sparsity basis matrix
  - e.g., in MRI: \( H = \) partial Fourier, \( \Phi = \) inverse wavelet

- \( y_t \): measurements (given)
- \( x_t \): sparsity basis vector
- \( \mathcal{N}_t \): support set of \( x_t \)
- \( w_t \): small noise

- Goal: recursively reconstruct \( x_t \) from \( y_0, y_1, \ldots y_t \),

- Use slow support change: \( |\mathcal{N}_t \setminus \mathcal{N}_{t-1}| \approx |\mathcal{N}_{t-1} \setminus \mathcal{N}_t| \ll |\mathcal{N}_t| \)
  - also use slow signal value change when valid

- Applications - dynamic projection imaging, e.g., dynamic MRI, CT; dynamic RPCA when outlier support reliably changes slowly over time, e.g., video

---

\(^5\) N. Vaswani, Kalman Filtered Compressed Sensing, ICIP, 2008
Dynamic CS: Solutions [KF-CS, ICIP’08], [LS-CS, T-SP, Aug10]

- Introduced Kalman filtered CS (KF-CS) and Least Squares CS (LS-CS):
  - first recursive algorithms that needed fewer measurements for accurate recovery than simple $\ell_1$
  - able to obtain time-invariant error bounds on LS-CS error under weaker RIP assumptions (fewer meas’s) than simple $\ell_1$

- But these could not achieve exact recovery with fewer meas’s than what simple $\ell_1$ needed

- Solved by Modified-CS
Modified-CS: sparse rec. with partial support knowledge [Modified-CS, ISIT'09, T-SP'10, T-IT'15]

- Idea: support at \( t - 1, \mathcal{N}_{t-1} \), is a good predictor of \( \mathcal{N}_t \)
- Reformulate: sparse recovery with partial support knowledge \( \mathcal{T} \)
  - \( \text{support}(x) = \mathcal{T} \cup \Delta \setminus \Delta_e: \Delta, \Delta_e \text{ unknown} \)
- Modified-CS: tries to find a vector \( \tilde{x} \) that is sparsest outside \( \mathcal{T} \) among all vectors satisfying the data constraint

\[
\min_{\tilde{x}} \|\tilde{x}_{\mathcal{T}^c}\|_1 \text{ subject to } \|y - A\tilde{x}\|_2 \leq \epsilon
\]
Modified-CS: sparse rec. with partial support knowledge [Modified-CS,ISIT’09,T-SP’10,T-IT’15]

- Idea: support at $t-1$, $\mathcal{N}_{t-1}$, is a good predictor of $\mathcal{N}_t$
- Reformulate: sparse recovery with partial support knowledge $\mathcal{T}$
  - $\text{support}(x) = \mathcal{T} \cup \Delta \setminus \Delta_e$: $\Delta, \Delta_e$ unknown
- Modified-CS: tries to find a vector $\tilde{x}$ that is sparsest outside $\mathcal{T}$ among all vectors satisfying the data constraint

$$\min_{\tilde{x}} \|\tilde{x}_{\mathcal{T}^c}\|_1 \text{ subject to } \|y - A\tilde{x}\|_2 \leq \epsilon$$

- Provably exact recovery in noise-free case if $\delta_{s+|\Delta|+|\Delta_e|} < 0.4$ [Vaswani,Lu, ISIT’09,T-SP’10]
- For noisy case: time-invariant error bounds under a realistic signal change model and $\delta_{s+k_s} < 0.4$ [Zhan,Vaswani, ISIT’13, T-IT’15 (to appear)]
Modified-CS: sparse rec. with partial support knowledge [Modified-CS,ISIT’09,T-SP’10,T-IT’15]

- Idea: support at $t-1$, $\mathcal{N}_{t-1}$, is a good predictor of $\mathcal{N}_t$

- Reformulate: sparse recovery with partial support knowledge $\mathcal{T}$
  - $\text{support}(x) = \mathcal{T} \cup \Delta \setminus \Delta_e$: $\Delta, \Delta_e$ unknown

- Modified-CS: tries to find a vector $\tilde{x}$ that is sparsest outside $\mathcal{T}$ among all vectors satisfying the data constraint

  $$\min_{\tilde{x}} \|\tilde{x}_{\mathcal{T}^c}\|_1 \text{ subject to } \|y - A\tilde{x}\|_2 \leq \epsilon$$

- Provably exact recovery in noise-free case if $\delta_{s+|\Delta|+|\Delta_e|} < 0.4$ [Vaswani,Lu, ISIT’09,T-SP’10]

- For noisy case: time-invariant error bounds under a realistic signal change model and $\delta_{s+k_s} < 0.4$ [Zhan,Vaswani, ISIT’13, T-IT’15 (to appear)]

- Regularized modified-CS & modified-CS-residual: also use slow signal value change (when valid)
Application: Dynamic MRI (larynx imaging example)

- Recovering a larynx sequence from only 19% simulated MRI measurements
- Proposed algorithm: Modified-CS. Here $\text{CS} \Leftrightarrow \ell_1 \text{ min}$
Application: Dynamic MRI (larynx imaging example)

- Recovering a larynx sequence from only 19% simulated MRI measurements
- Proposed algorithm: Modified-CS. Here $\text{CS} \leftrightarrow \ell_1 \text{ min}$
- Modified-CS NRMSE was 3%. Simple $\ell_1$-min NRMSE was 10%. It needed $n = 30\%$ meas’s to get 3% error.
Application: fMRI based brain activation detection

- Activation maps
- Used modified-CS for reconstructing the fMRI sequence; standard tools for active region detection
- Actual MRI scanner data; retrospective undersampling w/ $n_0 = 100\%$, $n = 30\%$
- Joint work with Dr. Ian Atkinson (UIC)


