Online Sparse + Low-Rank Matrix Recovery

Namrata Vaswani

Iowa State University
Web: http://www.ece.iastate.edu/~namrata

(joint work with Wei Lu, Chenlu Qiu and Brian Lois)
Most of this talk is joint work with my students

- Wei Lu (online sparse matrix recovery)
- Chenlu Qiu and Brian Lois (online sparse + low-rank matrix recovery)

Funded by NSF grants CCF-1117125 and CCF-0917015

Other collaborators: Han Guo (new student) and Prof. Leslie Hogben (Math, ISU)
Recovery from incomplete data: the question

- In many applications, data acquisition is slow, e.g. in MRI, acquire one Fourier coefficient of the cross-section of interest at a time
  - Question: can we recover the cross-section’s image from undersampled data?
Recovery from incomplete data: the question

- In many applications, data acquisition is slow, e.g. in MRI, acquire one Fourier coefficient of the cross-section of interest at a time
  - Question: can we recover the cross-section’s image from undersampled data?
  - Yes: if spatially-limited or if exploit sparsity of the image in an appropriate domain
Recovery from incomplete data: the question

- In many applications, data acquisition is slow, e.g. in MRI, acquire one Fourier coefficient of the cross-section of interest at a time
  - Question: can we recover the cross-section’s image from undersampled data?
  - Yes: if spatially-limited or if exploit sparsity of the image in an appropriate domain

- In many other applications, data acquisition is fast but cannot see everything, e.g. in video,
  \[ \text{image} = \text{background} + \text{foreground} \]
  
  - Question: can we recover two image sequences from one?
Introduction

Online Sparse Matrix Recovery

Online Sparse + Low-Rank Matrix Recovery

Our Work

Recovery from incomplete data: the question

▶ In many applications, data acquisition is slow, e.g. in MRI, acquire one Fourier coefficient of the cross-section of interest at a time

▶ Question: can we recover the cross-section’s image from undersampled data?

▶ Yes: if spatially-limited or if exploit sparsity of the image in an appropriate domain

▶ In many other applications, data acquisition is fast but cannot see everything, e.g. in video,

\[
\text{image} = \text{background} + \text{foreground}
\]

▶ Question: can we recover two image sequences from one?

▶ Yes: if exploit the low-rank structure of the background sequence and sparseness of the foreground
Sparse recovery: Magnetic Resonance Imaging (MRI)

- (a) Shepp-Logan phantom: 256 × 256 image
- (b) MR imaging pattern: 256-point DFT along 22 radial lines
- (c) Inverse-DFT
- (d) Basis Pursuit solution (uses sparsity: gives exact recovery!)

Example taken from [Candes, Romberg, Tao, T-IT, Feb 2006]
Sparse recovery / Compressive sensing [Mallat et al'93], [Feng,Bresler'96], [Gordinsky,Rao'97], [Chen,Donoho'98], [Candes,Romberg,Tao'05],[Donoho'05]

- Recover a sparse vector $x$, with support size at most $s$, from

  $$y := Ax + w$$

  when $A$ is a known fat matrix and $\|w\|_2 \leq \epsilon$ (small noise).
Sparse recovery / Compressive sensing [Mallat et al'93], [Feng,Bresler'96], [Gordinsky,Rao'97], [Chen,Donoho'98], [Candes,Romberg,Tao'05],[Donoho'05]

- Recover a sparse vector $x$, with support size at most $s$, from

$$y := Ax + w$$

when $A$ is a known fat matrix and $\|w\|_2 \leq \epsilon$ (small noise).
Sparse recovery / Compressive sensing [Mallat et al'93], [Feng,Bresler’96], [Gordinsky,Rao’97], [Chen,Donoho’98], [Candes,Romberg,Tao'05],[Donoho’05]

- Recover a sparse vector $x$, with support size at most $s$, from
  
  $$y := Ax + w$$

  when $A$ is a known fat matrix and $\|w\|_2 \leq \epsilon$ (small noise).

- Solution by convex relaxation: $\ell_1$ minimization [Chen,Donoho’98]:
  
  $$\min \|\tilde{x}\|_1 \text{ subject to } \|y - A\tilde{x}\|_2 \leq \epsilon$$

  if $\delta_{2s}(A) < 0.4$, error bounded by $C\epsilon$ [Candes et al'05,'06,'08]
Sparse recovery / Compressive sensing [Mallat et al'93], [Feng,Bresler’96], [Gordinsky,Rao’97], [Chen,Donoho’98], [Candes,Romberg,Tao'05],[Donoho’05]

- Recover a sparse vector $x$, with support size at most $s$, from
  \[ y := Ax + w \]
  when $A$ is a known fat matrix and $\|w\|_2 \leq \epsilon$ (small noise).

- Solution by convex relaxation: $\ell_1$ minimization [Chen,Donoho’98]:
  \[ \min \|\tilde{x}\|_1 \text{ subject to } \|y - A\tilde{x}\|_2 \leq \epsilon \]
  if $\delta_{2s}(A) < 0.4$, error bounded by $C\epsilon$ [Candes et al'05,'06,'08]

  - $\delta_s(A)$: smallest real number s.t.
    \[ (1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2 \]
    for all $s$-sparse $x$ [Candes,Tao,T-IT’05]
Sparse recovery / Compressive sensing [Mallat et al'93], [Feng,Bresler’96], [Gordinsky,Rao’97], [Chen,Donoho’98], [Candes,Romberg,Tao'05], [Donoho’05]

- Recover a sparse vector $x$, with support size at most $s$, from
  
  $$ y := Ax + w $$

  when $A$ is a known fat matrix and $\|w\|_2 \leq \epsilon$ (small noise).

- Solution by convex relaxation: $\ell_1$ minimization [Chen,Donoho’98]:
  
  $$ \min \|\tilde{x}\|_1 \text{ subject to } \|y - A\tilde{x}\|_2 \leq \epsilon $$

  if $\delta_{2s}(A) < 0.4$, error bounded by $C\epsilon$ [Candes et al'05,'06,'08]

  - $\delta_s(A)$: smallest real number s.t.
    
    $$ (1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2 $$

    for all $s$-sparse $x$ [Candes,Tao,T-IT’05]

- Applications: projection imaging - MRI, CT, astronomy, single-pixel camera
Low-rank matrix recovery (completion)

- Recover a low-rank matrix from a subset of its entries

\[ Y := \mathcal{P}_\Omega(L) \]

\( \Omega \) is the set of missing entries [Fazel et al, Recht et al, 2009]

- Applications: recommendation system design, e.g. Netflix problem; survey data analysis, ...
  - \( \ell_k \): ratings of movies by user \( k \)
  - A given user will rate only a subset of all the movies: missing entries; goal: complete the matrix in order to recommend movies
  - Matrix is low-rank: user preferences governed by only a few factors
Sparse + Low-rank matrix recovery - 1

- Separate a low-rank matrix $L$ and a sparse matrix $X$ 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

- Applications: video analytics (e.g. for surveillance, tracking, mobile video chat, occlusion removal, ...) [Candes et al,2009]

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

- $\ell_t$: bg - usually slow changing, global (dense) changes
- $x_t$: fg - sparse, consists of one or more moving objects (technically $x_t$: (fg-bg) on fg support)
Sparse + Low-rank matrix recovery - II

- Applications: detecting anomalous connectivity patterns in social networks or in computer networks
  - $\ell_t$: vector of n/w link “strengths” at time $t$ when no anomalous behavior
  - $x_t$: outliers or anomalies on a few links

- Applications: functional MRI based brain activity detection
  - only a sparse brain region activated in response to stimuli, everything else: very slow changes

- Applications: recommendation systems design in the presence of outliers and missing data
  - some users may enter completely incorrect ratings: due to laziness or malicious intent or just typos
Our work: the question

- How to solve the above problems for dynamically arriving data?
  - e.g., dynamic or functional MRI, online video analytics, ...

- Option 1: batch methods
  - recover the entire sequence in a batch fashion (e.g. for sparse recovery - use Fourier sparsity along the time axis)
  - slow and memory-intensive
Our work: the question

- How to solve the above problems for dynamically arriving data?
  - e.g., dynamic or functional MRI, online video analytics, ...

- Option 1: batch methods
  - recover the entire sequence in a batch fashion (e.g. for sparse recovery - use Fourier sparsity along the time axis)
  - slow and memory-intensive

- Option 2: do not use past knowledge
  - fast and memory-efficient, but will need more measurements
Our work: the question

- How to solve the above problems for dynamically arriving data?
  - e.g., dynamic or functional MRI, online video analytics, ...

- Option 1: batch methods
  - recover the entire sequence in a batch fashion (e.g. for sparse recovery - use Fourier sparsity along the time axis)
  - slow and memory-intensive

- Option 2: do not use past knowledge
  - fast and memory-efficient, but will need more measurements

- Option 3: design recursive algorithms (our work)
  - use previously recovered images and current observed data to recover the current image
  - fast and memory-efficient and need fewer measurements
Our Work: Online (Recursive) Solutions

- Provably accurate recursive solutions for
  - online sparse matrix recovery
    (recursive recovery of sparse signal sequences) [KF-CS, ICIP’08]
    - brief overview
  - online sparse + low-rank matrix recovery
    (online robust PCA) [Qiu, Vaswani, Allerton 2010]
    - most of this talk

- In this talk “recursive” ⇔ “online” (used interchangeably)
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], [LS-CS, ICASSP’09, TSP, Aug’10] [Modified-CS, ISIT’09, TSP, Sept’10]

- Kalman filtered CS (KF-CS), LS-CS: recursive algorithms that needed fewer measurements for accurate recovery than simple $\ell_1$
  [Vaswani, ICIP, 2008], [Vaswani, Trans. SP, 2010]
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], [LS-CS, ICASSP’09, TSP, Aug’10] [Modified-CS, ISIT’09, TSP, Sept’10]

- Kalman filtered CS (KF-CS), LS-CS: recursive algorithms that needed fewer measurements for accurate recovery than simple $\ell_1$ norm [Vaswani, ICIP, 2008], [Vaswani, Trans. SP, 2010]

- Modified-CS: provably exact recovery under weaker RIP assumptions (needs fewer measurements) [Vaswani, Lu, ISIT’09, Trans. SP, 2010]
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], 
[LS-CS, ICASSP’09, TSP, Aug10] [Modified-CS, ISIT’09, TSP, Sept’10]

- Kalman filtered CS (KF-CS), LS-CS: recursive algorithms that needed fewer measurements for accurate recovery than simple $\ell_1$ 
  [Vaswani, ICIP, 2008], [Vaswani, Trans.SP, 2010]

- Modified-CS: provably exact recovery under weaker RIP assumptions (needs fewer measurements) [Vaswani, Lu, ISIT’09, Trans.SP., 2010]

- Time-invariant error bounds in the noisy case under weaker RIP assumptions [Vaswani, Trans.SP, 2010], [Zhan, Vaswani, ISIT’13, T-IT(revised)]

- Stable & accurate recovery of dynamic MRI sequences from much fewer simulated MR (Fourier) measurements than simple $\ell_1$
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], [LS-CS, ICASSP’09, TSP, Aug’10] [Modified-CS, ISIT’09, TSP, Sept’10]

- Kalman filtered CS (KF-CS), LS-CS: recursive algorithms that needed fewer measurements for accurate recovery than simple $\ell_1$ [Vaswani, ICIP, 2008], [Vaswani, Trans.SP, 2010]

- Modified-CS: provably exact recovery under weaker RIP assumptions (needs fewer measurements) [Vaswani, Lu, ISIT’09, Trans.SP., 2010]

- Time-invariant error bounds in the noisy case under weaker RIP assumptions [Vaswani, Trans.SP, 2010], [Zhan, Vaswani, ISIT’13, T-IT(revised)]

- Stable & accurate recovery of dynamic MRI sequences from much fewer simulated MR (Fourier) measurements than simple $\ell_1$

- Modified-CS solved the sparse recovery w/ partially known support problem [Vaswani, Lu, ISIT’09, Trans.SP., 2010]
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], [Modified-CS, ISIT’09]

- Idea: support at \((t - 1)\) is a good predictor of support at \(t\)
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], [Modified-CS, ISIT’09]

- **Idea**: support at \((t - 1)\) is a good predictor of support at \(t\)
- **Reformulate**: sparse recovery with partial support knowledge \(T\)
  - \(\text{support}(x) = T \cup \Delta \setminus \Delta_e\): \(\Delta, \Delta_e\) unknown
  - **dynamic problem**: use \(\hat{T}_{t-1}\) as the set \(T\)
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], [Modified-CS, ISIT’09]

- Idea: support at \((t - 1)\) is a good predictor of support at \(t\)

- Reformulate: sparse recovery with partial support knowledge \(T\)
  - \(\text{support}(x) = T \cup \Delta \setminus \Delta_e: \Delta, \Delta_e \text{ unknown}\)
  - dynamic problem: use \(\hat{T}_{t-1}\) as the set \(T\)

- Modified-CS: find a vector \(\hat{x}\) that is sparsest outside the set \(T\) while satisfying the data constraint

\[
\min_{\hat{x}} \|\hat{x}_{T^c}\|_1 \text{ subject to } \|y - A\hat{x}\|_2 \leq \epsilon
\]

- \(\epsilon = 0: \) exact recovery if \(\delta_{s+|\Delta|+|\Delta_e|} < 0.4\) [Vaswani,Lu, ISIT’09, TSP’10]
Recursive recovery of sparse signal sequences [KF-CS, ICIP’08], [Modified-CS, ISIT’09]

- Idea: support at \((t - 1)\) is a good predictor of support at \(t\)
- Reformulate: sparse recovery with partial support knowledge \(T\)
  - \(\text{support}(x) = T \cup \Delta \setminus \Delta_e: \Delta, \Delta_e\) unknown
  - dynamic problem: use \(\hat{T}_{t-1}\) as the set \(T\)
- Modified-CS: find a vector \(\tilde{x}\) that is sparsest outside the set \(T\) while satisfying the data constraint
  \[
  \min_{\tilde{x}} \|\tilde{x}_{T^c}\|_1 \text{ subject to } \|y - A\tilde{x}\|_2 \leq \epsilon
  \]
  - \(\epsilon = 0\): exact recovery if \(\delta_{s+|\Delta|+|\Delta_e|} < 0.4\) [Vaswani, Lu, ISIT’09, TSP’10]
- Sparse signal sequences: time-invariant error bounds under a realistic signal change model and \(\delta_{s+k_s} < 0.4\) [Zhan, Vaswani, ISIT’13, T-IT(revised)]
Idea: support at \((t-1)\) is a good predictor of support at \(t\)

Reformulate: sparse recovery with partial support knowledge \(T\)
- support\((x) = T \cup \Delta \setminus \Delta_e\): \(\Delta, \Delta_e\) unknown
- dynamic problem: use \(\hat{T}_{t-1}\) as the set \(T\)

Modified-CS: find a vector \(\tilde{x}\) that is sparsest outside the set \(T\) while satisfying the data constraint

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

- \(\epsilon = 0\): exact recovery if \(\delta_{s+|\Delta|+|\Delta_e|} < 0.4\) [Vaswani,Lu, ISIT’09,TSP’10]

Sparse signal sequences: time-invariant error bounds under a realistic signal change model and \(\delta_{s+k_s} < 0.4\) [Zhan,Vaswani, ISIT’13,T-IT(revised)]

KF-CS or KF-modified-CS: also use slow signal value change (when valid)
Online Robust PCA: background

- Principal Components’ Analysis (PCA): estimate the low-dimensional subspace that best approximates a given dataset
  - SVD on data matrix, compute top left singular vectors

- Robust PCA: PCA in presence of outliers; many useful heuristics in older work, e.g., RSL [De la Torre et al., 2003]

- Online robust PCA: start with a good initial estimate of the low-dimensional subspace, keep updating it as more data comes in, while being robust to outliers
Principal Components’ Analysis (PCA): estimate the low-dimensional subspace that best approximates a given dataset

- SVD on data matrix, compute top left singular vectors

Robust PCA: PCA in presence of outliers; many useful heuristics in older work, e.g., RSL [De la Torre et al, 2003]

Online robust PCA: start with a good initial estimate of the low-dimensional subspace, keep updating it as more data comes in, while being robust to outliers

[Candes et al, 2009] posed robust PCA as: separate low-rank matrix $L$, sparse $X$ from

$$Y := X + L$$
Convex optimization solution for batch robust PCA

- [Candes et al, 2009; Chandrasekharan et al, 2009; Hsu et al, 2011] proposed a convex optimization program called PCP:

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

- If (a) left and right singular vectors of \(L\) are dense enough; (b) support of \(X\) is generated uniformly at random; (c) rank and sparsity are bounded, then PCP exactly recovers \(X\) and \(L\) from \(Y := X + L\) w.h.p. [Candes et al, 2009]

- [Chandrasekharan et al, 2009; Hsu et al, 2011]: similar flavor; replace ‘unif rand supp’ by upper bound on \# of nonzeros in any row of \(X\).

- first set of guarantees for a practical robust PCA approach
Convex optimization solution for batch robust PCA

- [Candes et al., 2009; Chandrasekharan et al., 2009; Hsu et al., 2011] proposed a convex optimization program called PCP:

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

- If (a) left and right singular vectors of \(L\) are dense enough; (b) support of \(X\) is generated uniformly at random; (c) rank and sparsity are bounded, then PCP exactly recovers \(X\) and \(L\) from \(Y := X + L\) w.h.p. [Candes et al., 2009]

  - [Chandrasekharan et al., 2009; Hsu et al., 2011]: similar flavor; replace ‘unif rand supp’ by upper bound on \# of nonzeros in any row of \(X\).

  - first set of guarantees for a practical robust PCA approach

- Much later work on batch methods for robust PCA w/ guarantees
Need for an online method

- Disadvantages of batch methods:
  - slower especially for online applications;
  - memory intensive;
  - do not allow infrequent/slow support change of columns of $X$
    - reason: this results in $X$ being rank deficient

- Video analytics: need online solution; and have occasionally static or slow moving fg objects

- Functional MRI: the activated brain region does not change a lot from frame to frame

- Network anomaly detection: need online solution; anomalous behavior continues for a period of time
Online Sparse + Low-Rank Recovery: Problem formulation

[Qiu,Vaswani,Allerton'10,'11] [Guo,Qiu,Vaswani,TSP'14] ¹

▶ Given sequentially arriving $n$-length data vectors $y_t$ satisfying

$$y_t := \ell_t, \quad t = 1, 2, \ldots, t_0$$

and

$$y_t := x_t + \ell_t, \quad t = t_0 + 1, t_0 + 2, \ldots, t_{\text{max}}$$

▶ $x_t$’s are sparse vectors with support sets, $T_t$, that have at least some changes over time;

▶ $\ell_t$’s lie in a fixed or slowly changing low-dimensional subspace of $\mathbb{R}^n$;

¹ C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
Online Sparse + Low-Rank Recovery: Problem formulation

[Qiu, Vaswani, Allerton'10,'11] [Guo, Qiu, Vaswani, TSP’14] ¹

- Given sequentially arriving $n$-length data vectors $y_t$ satisfying

$$y_t := \ell_t, \quad t = 1, 2, \ldots, t_0$$

and

$$y_t := x_t + \ell_t, \quad t = t_0 + 1, t_0 + 2, \ldots, t_{\text{max}}$$

- $x_t$’s are sparse vectors with support sets, $T_t$, that have at least some changes over time;

- $\ell_t$’s lie in a fixed or slowly changing low-dimensional subspace of $\mathbb{R}^n$; and left singular vectors of $[\ell_1, \ell_2, \ldots \ell_t]$ are dense;

---

¹ C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
Online Sparse + Low-Rank Recovery: Problem formulation

[Qiu, Vaswani, Allerton'10,'11] [Guo, Qiu, Vaswani, TSP’14] ¹

▷ Given sequentially arriving $n$-length data vectors $y_t$ satisfying

$$y_t := \ell_t, \quad t = 1, 2, \ldots, t_0$$

and

$$y_t := x_t + \ell_t, \quad t = t_0 + 1, t_0 + 2, \ldots, t_{\text{max}}$$

▷ $x_t$’s are sparse vectors with support sets, $T_t$, that have at least some changes over time;

▷ $\ell_t$’s lie in a fixed or slowly changing low-dimensional subspace of $\mathbb{R}^n$; and left singular vectors of $[\ell_1, \ell_2, \ldots \ell_t]$ are dense;

▷ recursively recover $x_t, \ell_t$ and $\text{span}([\ell_1, \ell_2, \ldots \ell_t])$ at all $t > t_0$.

¹ C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
Interpretations:

- **Online / Recursive Robust PCA:**
  - $x_t$ is outlier, recover $l_t$ and span([$l_1, l_2, \ldots, l_t$])

- **Online Robust Sparse Recovery** (robust to large but structured noise)
  - recover $x_t$, $l_t$ is noise, $\|l_t\|$ can be much larger than $\|x_t\|$
  - extension to the undersampled case $y_t := Ax_t + l_t$ is easy

- **Online matrix completion:** simpler special case w/ $T_t$ known
  - $T_t$ is the set of the unknown entries of $l_t$ at time $t$. 
Related Work

Batch robust PCA

- Older work, e.g. RSL [de la Torre et al, IJCV'03]: PCP and then much later work

Recursive / incremental / online robust PCA: algorithms

- Older work (before PCP): [Li et al, ICIP 2003]: iRSL: doesn’t work
- [Qiu, Vaswani, Allerton’10, Allerton’11, TSP’14]: ReProCS (RecursiveProjected CS)
- [Balzano et al, CVPR 2012]: GRASTA
- [Mateos et al, JSTSP 2013]: batch, online; online: not enough info, no code

Theoretical guarantees

- [Qiu, Vaswani, Lois, Hogben, ICASSP, ICASP, ISIT’13, Trans. Info. Th’14]: partial result;
- [Lois, Vaswani, arXiv:1409.3959]: complete correctness result
- [Feng et al, NIPS’13 OR-PCA Stoch Opt]: partial result and only asymptotic
- [Feng et al, NIPS’13, OR-PCA Contam Data]: outlier is anything far from data subspace, also asymptotic result
Experiments [Guo, Qiu, Vaswani, TSP’14]\(^2\)

1. Real background simulated foreground: background of moving lake water video with a simulated moving rectangular object overlaid on it; object intensity similar to background intensity and object moving slowly (making it a difficult seq)

2. Real video: background consisted of white colored curtains moving due to the wind and foreground consisted of a person with a white shirt coming on writing on the board and leaving, then a second person doing that.

\(^2\) H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
Figure: Foreground layer recovery error plot. Black: batch methods, Red: online methods, Red Circles: ReProCS
**Introduction**

Online Sparse Matrix Recovery

Online Sparse + Low-Rank Matrix Recovery

**Background and Problem Formulation**

ReProCS algorithm and experiments

Correctness Result and Proof Outline

---

**Figure:** Foreground layer recovery at $t = t_{train} + 60, 120, 199, 475, 1148$. 

Namrata Vaswani
Figure: Background layer recovery at $t = t_{\text{train}} + 60, 120, 199, 475, 1148$. 

<table>
<thead>
<tr>
<th>original</th>
<th>ReProCS (bg)</th>
<th>PCP (bg)</th>
<th>RSL (bg)</th>
<th>GRASTA (bg)</th>
<th>MG (bg)</th>
</tr>
</thead>
</table>

Namrata Vaswani, Online Sparse + Low-Rank Recovery
Some definitions

- $P$ is a basis matrix $\iff P'P = I$
- Estimate $P \iff$ estimate span($P$): subspace spanned by col's of $P$
- $\hat{P}$ is an accurate estimate of $P \iff \| (I - \hat{P}\hat{P}')P \|_2 \ll 1$
Recall: \( y_t := x_t + \ell_t \), suppose \( \ell_t = P_t a_t \), \( P_t \): tall \( n \times r \) basis matrix
Recall: $y_t := x_t + \ell_t$, suppose $\ell_t = P_t a_t$, $P_t$: tall $n \times r$ basis matrix

**Initialize:** compute $\hat{P}_0 = \text{top left singular vectors of } [\ell_1, \ell_2, \ldots, \ell_{t_0}]$. 

---

3. C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
ReProCS algorithm \cite{Qiu,Vaswani,Allerton'10,Allerton'11,Guo,Qiu,Vaswani,TSP'14}³

Recall: $y_t := x_t + \ell_t$, suppose $\ell_t = P_t a_t$, $P_t$: tall $n \times r$ basis matrix

**Initialize:** compute $\hat{P}_0 = \text{top left singular vectors of } [\ell_1, \ell_2, \ldots \ell_{t_0}]$.

For $t > t_0$, do

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

---

³ C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
ReProCS algorithm [Qiu, Vaswani, Allerton’10, Allerton’11], [Guo, Qiu, Vaswani, TSP’14]³

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

**Initialize:** compute \( \hat{P}_0 = \) top left singular vectors of \( [\ell_1, \ell_2, \ldots, \ell_{t_0}] \).

For \( t > t_0 \), do

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

- **Noisy Sparse Recovery:** \( \ell_1 \) min + support estimate + LS: get \( \hat{x}_t \)
  - denseness of \( P_t \)’s ⇒ sparse \( x_t \) recoverable from \( \tilde{y}_t \)

---

³ C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
ReProCS algorithm \cite{Qiu,Vaswani,Allerton'10,Allerton'11},\cite{Guo,Qiu,Vaswani,TSP'14}\textsuperscript{3}

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

\textbf{Initialize:} compute \( \hat{P}_0 = \text{top left singular vectors of } [\ell_1, \ell_2, \ldots \ell_{t_0}] \).

For \( t > t_0 \), do

\begin{itemize}
  \item Projection: compute \( \tilde{y}_t := \Phi_t y_t \), where \( \Phi_t := I - \hat{P}_{t-1} \hat{P}'_{t-1} \)
    \begin{itemize}
      \item then \( \tilde{y}_t = \Phi_t x_t + \beta_t \), \( \beta_t := \Phi_t \ell_t \) is small “noise” because of slow subspace change
    \end{itemize}
  \item Noisy Sparse Recovery: \( \ell_1 \) min + support estimate + LS: get \( \hat{x}_t \)
    \begin{itemize}
      \item denseness of \( P_t \)'s \( \Rightarrow \) sparse \( x_t \) recoverable from \( \tilde{y}_t \)
    \end{itemize}
  \item Recover \( \ell_t \): compute \( \hat{\ell}_t = y_t - \hat{x}_t \)
\end{itemize}

\textsuperscript{3}C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum”, IEEE Trans. SP, Aug 2014
Recall: \( y_t := x_t + \ell_t \), suppose \( \ell_t = P_t a_t \), \( P_t \): tall \( n \times r \) basis matrix

**Initialize:** compute \( \hat{P}_0 = \text{top left singular vectors of } [\ell_1, \ell_2, \ldots \ell_{t_0}] \).

For \( t > t_0 \), do

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

- **Noisy Sparse Recovery:** \( \ell_1 \min \text{ + support estimate + LS: get } \hat{x}_t \)
  
    - denseness of \( P_t \)’s \( \Rightarrow \) sparse \( x_t \) recoverable from \( \tilde{y}_t \)

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

- **Subspace update:** update \( \hat{P}_t \) every \( \alpha \) frames by projection-PCA

---

3 C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
H. Guo, C. Qiu, N. Vaswani, An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum", IEEE Trans. SP, Aug 2014
ReProCS algorithm: projection-PCA key idea

Model assumption for designing p-PCA: $\ell_t = P_t a_t$ where $P_t = P(j)$ for $t \in [t_j, t_{j+1} - 1]$; at $t = t_j$, changes as $P(j) = [P(j-1) \setminus P_{j,old}, P_{j,new}]$.

At $t > t_0$, we have an accurate $\hat{P}_0$ available; every $\alpha$ frames:

- project $\hat{\ell}_t$’s from the previous $\alpha$ frames perp to $\hat{P}_0$:
  \[
  M_t := (I - \hat{P}_{j-1} \hat{P}_{j-1}')[\hat{\ell}_{t-\alpha+1}, \hat{\ell}_{t-\alpha+2}, \ldots, \hat{\ell}_t]
  \]

- if any sv of $M_t >$ thresh: declare subspace change: $\hat{t}_j = t$
- if change: estimate $P_{j,new}$ $K$ times, use new set of frames each time
  - $\hat{P}_{j,new,1}$: left sing. vec.’s of $M_{\hat{t}_j+\alpha}$ with sv’s above thresh
  - use $\hat{P}_t = [\hat{P}_{j-1}, \hat{P}_{j,new,1}]$ in proj sparse rec for next $\alpha$ frames
  - $\hat{P}_{j,new,2}$: left sing. vec.’s of $M_{\hat{t}_j+2\alpha}$ with sv’s above thresh
  - use $\hat{P}_t = [\hat{P}_{j-1}, \hat{P}_{j,new,2}]$ in projected sparse recovery
  - ... do this $K$ times and finally set $\hat{P}_j = [\hat{P}_{j-1}, \hat{P}_{j,new,K}]$
Why ReProCS works [Qiu, Vaswani, Lois, Hogben, Trans. IT, 2014]  

- Slow subspace change: noise seen by sparse recovery step is small
Why ReProCS works [Qiu, Vaswani, Lois, Hogben, Trans. IT, 2014] 4

- Slow subspace change: noise seen by sparse recovery step is small
- Denseness of columns of $P_t$: RIC of $\Phi_t = I - \hat{P}_{t-1} \hat{P}'_{t-1}$ is small
  - assume that

$$\max_t \max_i \| e_i' P_{t-1} \|_2^2 \leq \frac{\mu r}{n} \quad \text{and} \quad \frac{\mu r(2s)}{n} \leq 0.09$$

- easy to show [Qiu, Vaswani, Lois, Hogben, Trans. IT, 2014] :

$$\delta_{2s}(\Phi_t) = \max_{|T| \leq 2s} \| I_T' \hat{P}_{t-1} \|_2^2 \leq 0.09 + 0.05$$

---

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 \cite{Qiu,Vaswani,Lois,Hogben,Trans.IT,2014} \footnote{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}

- Slow subspace change: noise seen by sparse recovery step is small
- Denseness of columns of $P_t$: RIC of $\Phi_t = I - \hat{P}_{t-1}\hat{P}_{t-1}'$ is small
  - assume that
  $\max_t \max_i \|e'_i P_{t-1}\|^2 \leq \frac{\mu r}{n}$ and $\frac{\mu r(2s)}{n} \leq 0.09$

  easy to show \cite{Qiu,Vaswani,Lois,Hogben,Trans.IT,2014}:

  $\delta_{2s}(\Phi_t) = \max_{|T| \leq 2s} \|l_T' \hat{P}_{t-1}\|^2 \leq 0.09 + 0.05$

- Above two facts + any result for $\ell_1$ min: $x_t$ is accurately recovered
Why ReProCS works [Qiu,Vaswani,Lois,Hogben,Trans.IT,2014]  

- Slow subspace change: noise seen by sparse recovery step is small
- Denseness of columns of $P_t$: RIC of $\Phi_t = I - \hat{P}_{t-1} \hat{P}_{t-1}'$ is small
  - assume that
    \[
    \max_t \max_i \| e_i' P_{t-1} \|_2^2 \leq \frac{\mu r}{n} \text{ and } \frac{\mu r(2s)}{n} \leq 0.09
    \]
    
    easy to show [Qiu,Vaswani,Lois,Hogben,Trans.IT,2014]:
    \[
    \delta_{2s}(\Phi_t) = \max_{|T| \leq 2s} \| I_T' \hat{P}_{t-1} \|_2^2 \leq 0.09 + 0.05
    \]

- Above two facts + any result for $\ell_1$ min: $x_t$ is accurately recovered
- Hence $\ell_t = y_t - x_t$ is accurately recovered
- Most of the work: show accurate subspace recovery $\hat{P}_t \approx P_t$

---

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
ReProCS correctness result \cite{Lois,Vaswani,arXiV:1409.3959}

With probability at least $1 - n^{-10}$,

- support of $x_t$ is exactly recovered at all times,
- recovery error is bounded: $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq b$ at all times,
- subspace error decays to a small value within a short delay of a subspace change,

if

---

\footnote{B. Lois and N. Vaswani, A Correctness Result for Online Robust PCA, arXiV:1409.3959. 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.}
ReProCS correctness result [Lois, Vaswani, arXiV:1409.3959]5

With probability at least $1 - n^{-10}$,

- support of $x_t$ is exactly recovered at all times,
- recovery error is bounded: $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq b$ at all times,
- subspace error decays to a small value within a short delay of a subspace change,

if

- the initial subspace of $\ell_t$ is accurately known,

---

5 B. Lois and N. Vaswani, A Correctness Result for Online Robust PCA, arXiV:1409.3959.  

With probability at least $1 - n^{-10}$,

- support of $x_t$ is exactly recovered at all times,
- recovery error is bounded: $\| x_t - \hat{x}_t \|_2 = \| l_t - \hat{l}_t \|_2 \leq b$ at all times,
- subspace error decays to a small value within a short delay of a subspace change,

if

- the initial subspace of $l_t$ is accurately known,
- $l_t$ lies in a slowly changing low-dimensional subspace

---

5 B. Lois and N. Vaswani, A Correctness Result for Online Robust PCA, arXiv:1409.3959.
ReProCS correctness result \cite{LoisVaswani}

With probability at least $1 - n^{-10}$,

- support of $x_t$ is exactly recovered at all times,
- recovery error is bounded: $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq b$ at all times,
- subspace error decays to a small value within a short delay of a subspace change,

if

- the initial subspace of $\ell_t$ is accurately known,
- $\ell_t$ lies in a slowly changing low-dimensional subspace
- the columns of $P_t$ (subspace basis matrix for $\ell_t$) are dense enough,

---

\cite{LoisVaswani} B. Lois and N. Vaswani, A Correctness Result for Online Robust PCA, arXiv:1409.3959.
\cite{QiuVaswaniLoisHogben} 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.
ReProCS correctness result [Lois, Vaswani, arXiV:1409.3959]$^5$

With probability at least $1 - n^{-10}$,

- support of $x_t$ is exactly recovered at all times,
- recovery error is bounded: $\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq b$ at all times,
- subspace error decays to a small value within a short delay of a subspace change,

if

- the initial subspace of $\ell_t$ is accurately known,
- $\ell_t$ lies in a slowly changing low-dimensional subspace
- the columns of $P_t$ (subspace basis matrix for $\ell_t$) are dense enough,
- the support set of $x_t$ changes by at least a certain amount at least once every $h$ frames

$^5$ B. Lois and N. Vaswani, A Correctness Result for Online Robust PCA, arXiV:1409.3959.

Correctness result implications [Lois, Vaswani, arXiv:1409.3959], [Qiu, Vaswani, Lois, Hogben, T-IT’14]

For most videos,

- the region occupied by foreground objects (support of $x_t$) is exactly recovered at all times, and
- foreground and background images are accurately recovered at all times ($\|x_t - \hat{x}_t\|_2 = \|l_t - \hat{l}_t\|_2 \leq b$)

if

- an initial background-only training sequence is available (to get an initial subspace estimate)
For most videos,

- the region occupied by foreground objects (support of $x_t$) is exactly recovered at all times, and
- foreground and background images are accurately recovered at all times ($\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq b$)

if

- an initial background-only training sequence is available (to get an initial subspace estimate)
- the background images change slowly
Correctness result implications [Lois,Vaswani, arXiV:1409.3959],[Qiu,Vaswani,Lois,Hogben,T-IT’14]

For most videos,

- the region occupied by foreground objects (support of $x_t$) is exactly recovered at all times, and
- foreground and background images are accurately recovered at all times ($\|x_t - \hat{x}_t\|_2 = \|l_t - \hat{l}_t\|_2 \leq b$)

if

- an initial background-only training sequence is available (to get an initial subspace estimate)
- the background images change slowly
- background changes (w.r.t. a mean background image) are dense,
Correctness result implications [Lois,Vaswani, arXiV:1409.3959],[Qiu,Vaswani,Lois,Hogben,T-IT’14]

For most videos,

- the region occupied by foreground objects (support of $x_t$) is exactly recovered at all times, and
- foreground and background images are accurately recovered at all times ($\|x_t - \hat{x}_t\|_2 = \|\ell_t - \hat{\ell}_t\|_2 \leq b$)

if

- an initial background-only training sequence is available (to get an initial subspace estimate)
- the background images change slowly
- background changes (w.r.t. a mean background image) are dense,
- there is some motion of the foreground objects at least once every so often

Details follow in the next few slides . . .
ReProCS correctness result: Support change model - examples

1. *(Random motion)* all support sets mutually disjoint
   ▶ this satisfies our model as long as $s \in O(\frac{n}{\log n})$
ReProCS correctness result: Support change model - examples

1. *(Random motion)* all support sets mutually disjoint
   - this satisfies our model as long as $s \in O\left(\frac{n}{\log n}\right)$

2. *(Infrequent motion)* a 1D object of length $s$ that moves at least once every $h^+\alpha$ frames; and, when it moves, it moves down by at least $s/\varrho$ pixels
   - each time it moves, it moves by no more than $b_2s$ indices
   - this satisfies our model as long as $s \in O\left(\frac{n}{\log n}\right)$
ReProCS correctness result: Support change model - examples

1. *(Random motion)* all support sets mutually disjoint
   ▶ this satisfies our model as long as $s \in O\left(\frac{n}{\log n}\right)$

2. *(Infrequent motion)* a 1D object of length $s$ that moves at least once every $h^+\alpha$ frames; and, when it moves, it moves down by at least $s/\varrho$ pixels
   ▶ each time it moves, it moves by no more than $b_2 s$ indices
   ▶ this satisfies our model as long as $s \in O\left(\frac{n}{\log n}\right)$

3. *(Slow motion)* An object of length $s$ moves down by at least one pixel in every frame
   ▶ this satisfies our model as long as $s \in O(\log n)$
ReProCS correctness result: Support change model - examples

1. *(Random motion)* all support sets mutually disjoint
   ▶ this satisfies our model as long as $s \in O\left(\frac{n}{\log n}\right)$

2. *(Infrequent motion)* a 1D object of length $s$ that moves at least once every $h^+\alpha$ frames; and, when it moves, it moves down by at least $s/\rho$ pixels
   ▶ each time it moves, it moves by no more than $b_2s$ indices
   ▶ this satisfies our model as long as $s \in O\left(\frac{n}{\log n}\right)$

3. *(Slow motion)* An object of length $s$ moves down by at least one pixel in every frame
   ▶ this satisfies our model as long as $s \in O(\log n)$

Define $x_{\min} := \min_t \min_{i \in \text{support}(x_t)} |(x_t)_i|$
ReProCS correctness result: Support change model - examples

(a) disjoint supports  (b) infrequent motion  (c) slow moving

Figure: In any of these we could have randomly selected pixels (need not be a block) at a given time and also random ordering across time
ReProCS correctness result: Subspace change model

Assume that

- $\ell_t = P_t a_t$ with $P_t = P(j)$ for $t \in [t_j, t_{j+1} - 1]$, $j = 1, 2, \ldots J$
  - $P(j)$: very tall $n \times r_j$ basis matrix that changes as
    $$P(j) = [P(j-1) \setminus P_{j,\text{old}}, P_{j,\text{new}}]$$
  - $a_t$’s zero mean, bounded, and mutually independent r.v.’s with diagonal covariance matrix $\Lambda_t$
ReProCS correctness result: Subspace change model

Assume that

- $\ell_t = P_t a_t$ with $P_t = P(j)$ for $t \in [t_j, t_{j+1} - 1]$, $j = 1, 2, \ldots J$
  - $P(j)$: very tall $n \times r_j$ basis matrix that changes as
    
    $P(j) = [P(j-1) \setminus P_{j,\text{old}}, P_{j,\text{new}}]$
  - $a_t$'s zero mean, bounded, and mutually independent r.v.'s with diagonal covariance matrix $\Lambda_t$

Define

- $r := r_0 + Jc$, $c := \max_j \text{rank}(P(j)_{\text{new}})$
- $f := \frac{\max_t \lambda_{\max}(\Lambda_t)}{\min_t \lambda_{\min}(\Lambda_t)}$, $g := \frac{\max_{t \in [t_j, t_{j+d}]} \lambda_{\max}(\Lambda_{t,\text{new}})}{\min_{t \in [t_j, t_{j+d}]} \lambda_{\min}(\Lambda_{t,\text{new}})}$
- $\gamma_* := \max_t \|a_t\|_\infty$, $\gamma_{\text{new}} := \max_{t \in [t_j, t_{j+d}]} \|a_{t,\text{new}}\|_\infty$

No bound needed on $f$ or on $\gamma_*$: allow large but structured $\ell_t$
Theorem (Correctness result for ReProCS)

Pick a $\zeta \leq \min\left(\frac{10^{-4}}{(r_0 + Jc)^2 f}, \frac{1}{(r_0 + Jc)^3 \gamma_*^2}\right)$. Assume the subspace change model and support change model hold and assume $\| (I - \hat{P}_0 \hat{P}'_0) P_0 \|_2 \leq r_0 \zeta$. If

1. *algorithm parameters* $\alpha, K, \xi, \omega$ are set appropriately,
Theorem (Correctness result for ReProCS)

Pick a $\zeta \leq \min\left(\frac{10^{-4}}{(r_0 + Jc)^2 f}, \frac{1}{(r_0 + Jc)^3 \gamma_*^2}\right)$. Assume the subspace change model and support change model hold and assume $\| (I - \hat{P}_0 \hat{P}_0') P_0 \|_2 \leq r_0 \zeta$. If

1. *algorithm parameters* $\alpha, K, \xi, \omega$ are set appropriately,

2. *subspace basis matrices are dense enough*:

$$\max_{|T| \leq 2s} \| I_T' [P_0, P_{1, new}, P_{2, new}, \ldots, P_{J, new}] \|_2 \leq 0.3, \quad \max_{|T| \leq 2s} \| I_T' P_{j, new} \|_2 \leq 0.02, \quad x_{\text{min}} \quad \text{and} \quad g \leq 1.$$
Theorem (Correctness result for ReProCS)

Pick a $\zeta \leq \min\left(\frac{10^{-4}}{(r_0+Jc)^2f}, \frac{1}{(r_0+Jc)^3\gamma^2_*}\right)$. Assume the subspace change model and support change model hold and assume $\| (I - \hat{P}_0\hat{P}'_0) P_0 \|_2 \leq r_0 \zeta$. If

1. algorithm parameters $\alpha, K, \xi, \omega$ are set appropriately,

2. subspace basis matrices are dense enough:

\[
\max_{|T| \leq 2s} \| I'_T [P_0, P_{1,\text{new}}, P_{2,\text{new}}, \ldots P_{J,\text{new}}] \|_2 \leq 0.3, \quad \max_{|T| \leq 2s} \| I'_T P_{j,\text{new}} \|_2 \leq 0.02,
\]

3. slow subspace change holds:

   - delay b/w change times: $\min_j (t_{j+1} - t_j) > d \geq (K + 2)\alpha$,
   - projection along new direc's is initially small:
     \[
     \max_{t \in [t_j, t_j + d]} \| a_{t,\text{new}} \|_\infty \leq 0.05x_{\min} \text{ and } g \leq 1.4,
     \]
Theorem (Correctness result for ReProCS)

Pick a $\zeta \leq \min(\frac{10^{-4}}{(r_0+Jc)^2f}, \frac{1}{(r_0+Jc)^3\gamma^2})$. Assume the subspace change model and support change model hold and assume $\|(I - \hat{P}_0\hat{P}'_0)P_0\|_2 \leq r_0\zeta$. If

1. algorithm parameters $\alpha, K, \xi, \omega$ are set appropriately,

2. subspace basis matrices are dense enough:

$$\max_{|T| \leq 2s} \|I_T'[P_0, P_{1,new}, P_{2,new}, \ldots P_{J,new}]\|_2 \leq 0.3, \quad \max_{|T| \leq 2s} \|I_T'P_{j,new}\|_2 \leq 0.02,$$

3. slow subspace change holds:
   ▶ delay b/w change times: $\min_j(t_{j+1} - t_j) > d \geq (K + 2)\alpha$,
   ▶ projection along new direc’s is initially small:
     $$\max_{t \in [t_j, t_{j+d}]} \|a_{t,new}\|_\infty \leq 0.05x_{min} \text{ and } g \leq 1.4,$$

4. support of $x_t$ changes enough: e.g. moves by at least $s/10$ pixels at least once every $\alpha/500$ frames,

then, with probability at least $1 - n^{-10}$,
then, with probability at least $1 - n^{-10}$,

1. $\text{support}(x_t)$ is exactly recovered at all times,

2. $SE_t := \| (I - \hat{P}_t \hat{P}_t') P_t \|_2$ reduces to $(r + c)\zeta$ within $K\alpha$ frames after $t_j$,

3. $\| \ell_t - \hat{\ell}_t \|_2 = \| x_t - \hat{x}_t \|_2 \leq b \ll \| x_t \|_2$

Notice: no bound needed on $f$ or on $\gamma_*$: allow large but structured $\ell_t$
Discussion: Contributions

- To the best of our knowledge, this is the first correctness result for online sparse + low-rank recovery
  - or online robust PCA or recursive sparse recovery in large but low-dimensional noise

- Advantages
  - allows significantly more correlated support change than PCP
    - it allows $O(1)$ fraction of nonzeros per row; PCP only allows $O(1/r)$ [Hsu et al’2011] or needs unif random support [Candes et al]
Discussion: Contributions

- To the best of our knowledge, this is the first correctness result for online sparse + low-rank recovery
  - or online robust PCA or recursive sparse recovery in large but low-dimensional noise

- Advantages
  - allows significantly more correlated support change than PCP
    - it allows $O(1)$ fraction of nonzeros per row; PCP only allows $O(1/r)$ \cite{Hsu2011} or needs unif random support \cite{Candes2011}
  - online algorithm: faster; less storage needed: only $O(n \log n)$ instead of $O(nt_{\text{max}})$
Discussion: Contributions

- To the best of our knowledge, this is the first correctness result for online sparse + low-rank recovery
  - or online robust PCA or recursive sparse recovery in large but low-dimensional noise

- Advantages
  - allows significantly more correlated support change than PCP
    - it allows $O(1)$ fraction of nonzeros per row; PCP only allows $O(1/r)$ \cite{Hsu2011} or needs unif random support \cite{Candes2011}
  - online algorithm: faster; less storage needed: only $O(n \log n)$ instead of $O(nt_{\text{max}})$

- New proof techniques needed to obtain our results
  - almost all existing robust PCA results are for batch approaches
  - previous finite sample PCA results are not useful: assume $e_t := \hat{\ell}_t - \ell_t$ is uncorrelated with $\ell_t$
Discussion: Limitations

- Needs knowledge of $\gamma_{\text{new}}, c$
- Needs initial subspace knowledge and slow subspace change
  - both are usually practically valid; if initial subspace not available, then can use a batch method initially to get it
Discussion: Limitations

- Needs knowledge of $\gamma_{\text{new}}$, $c$
- Needs initial subspace knowledge and slow subspace change
  - both are usually practically valid; if initial subspace not available, then can use a batch method initially to get it
- Needs zero-mean and independence of $a_t$'s and hence $\ell_t$'s over time
  - can replace it by a more practical autoregressive model on $a_t$'s (ongoing)
Discussion: Limitations

- Needs knowledge of $\gamma_{\text{new}}$, $c$
- Needs initial subspace knowledge and slow subspace change
  - both are usually practically valid; if initial subspace not available, then can use a batch method initially to get it
- Needs zero-mean and independence of $a_t$'s and hence $\ell_t$'s over time
  - can replace it by a more practical autoregressive model on $a_t$'s (ongoing)
- Needs a tighter bound on rank and sparsity. Recall
  $\text{support}(X) = st_{\text{max}}$
  - We allow $\text{support}(X) \in O\left(\frac{nt_{\text{max}}}{\log n}\right)$ and $\text{rank}(L) \in O(\log n)$
  - PCP allows $\text{support}(X) \in O(nt_{\text{max}})$, $\text{rank}(L) \in O\left(\frac{n}{\log^2 n}\right)$
Discussion: Limitations

- Needs knowledge of $\gamma_{\text{new}}, c$
- Needs initial subspace knowledge and slow subspace change
  - both are usually practically valid; if initial subspace not available, then can use a batch method initially to get it
- Needs zero-mean and independence of $a_t$’s and hence $\ell_t$’s over time
  - can replace it by a more practical autoregressive model on $a_t$’s (ongoing)
- Needs a tighter bound on rank and sparsity. Recall $\text{support}(X) = st_{\text{max}}$
  - We allow $\text{support}(X) \in O\left(\frac{nt_{\text{max}}}{\log n}\right)$ and $\text{rank}(L) \in O(\log n)$
  - PCP allows $\text{support}(X) \in O(nt_{\text{max}})$, $\text{rank}(L) \in O\left(\frac{n}{\log^2 n}\right)$
- Result for ReProCS-deletion (ongoing) allows $\text{support}(X) \in O\left(\frac{nt_{\text{max}}}{\log n}\right)$ and $\text{rank}(L) \in O(n)$.
  - but it also needs an extra clustering assumption on the eigenvalues after the subspace change has stabilized
Figure: Notice that $\frac{\|a_{t, new}\|_\infty}{\|a_t\|_\infty} \leq 0.1$ for the first 40 frames, i.e. $d = 40$.

Used a low-rankified bg seq from curtain and lake videos. Computed $P_0$ as the 99.99% eigenvectors' set of the first $p = 150$ frames and set $\lambda^-$ as the smallest eigenvalue of this set; computed $P_{1, new}$ as eigenvectors of $(I - P_0 P_0')\sum_{t=p+1}^{2p} \ell_t \ell_t' (I - P_0' P_0')$ with eigenvalues above $\lambda^-$ and so on. Got $c_{max} = 5$, $r_0 = 29$ for curtain; $c_{max} = 3$, $r_0 = 8$ for lake.
Some Generalizations (ongoing work)

- Replace independence of $a_t$’s by an autoregressive model

\[ a_t = ba_{t-1} + \nu_t, \quad b \leq b_0 < 1 \]

The same result holds if we also have

- $b \leq b_0 = 0.1$, $\alpha \geq 10r^{3/2}\eta f$, $\zeta < \frac{1}{r^3\eta f}$
Some Generalizations (ongoing work)

- Replace independence of $a_t$’s by an autoregressive model
  \[ a_t = ba_{t-1} + \nu_t, \quad b \leq b_0 < 1 \]
  The same result holds if we also have
  - $b \leq b_0 = 0.1$, $\alpha \geq 10r^{3/2}f$, $\zeta < \frac{1}{r^3 \eta f}$

- Result for ReProCS-deletion:
  - A similar result holds if we also have
    - enough directions deleted so that $r_j \leq r_0$
    - for the last $(\theta + 2)\alpha$ frames before $t_{j+1}$, the cov mat $\Lambda_t$ is constant and its eigenvalues “clustered” into $\theta$ clusters
  - This result needs a significantly weaker denseness assumption
    \[ \max_j \kappa_{2s}(P_j) \leq 0.3, \quad \text{and} \quad \max_j \kappa_{2s}(P_{j,\text{new}}) \leq 0.02 \]
    and this is why it allows $\text{support}(X) \in O\left(\frac{n}{\log n} t_{\max}\right)$ and $\text{rank}(L) \in O(n)$
Some Generalizations (ongoing work)

- **Extension to**

  \[ y_t = Ax_t + B\ell_t \]

- **Undersampled-ReProCS**: replace \( \Phi_t \) by \( \Phi_tA \) in the \( \ell_1 \) min and LS steps [Qiu, Vaswani, Allerton'11] \(^6\)

---

\(^6\) C. Qiu and N. Vaswani, Recursive Sparse Recovery in Large but Correlated Noise, Allerton 2011
Some Generalizations (ongoing work)

- Extension to
  \[ y_t = Ax_t + B\ell_t \]

- Undersampled-ReProCS: replace \( \Phi_t \) by \( \Phi_tA \) in the \( \ell_1 \) min and LS steps [Qiu, Vaswani, Allerton'11] \(^6\)

- A similar result will hold for u-ReProCS if [Lois, Vaswani, Qiu, GlobalSIP’13]
  - \( \delta_{2s}(A) < 0.1 \), and
  - we replace \( \| I_T'P \|_2 \) by \( \| A_T'P \|_2 \) in the denseness assumption and \( P \)'s now form a basis for the subspace of \( \ell_t := B\ell_t \).

- The case \( y_t = A_t x_t + B_t\ell_t \) is NOT an easy extension

---

\(^6\) C. Qiu and N. Vaswani, Recursive Sparse Recovery in Large but Correlated Noise, Allerton 2011.
Some Generalizations (ongoing work)

- Extension to

\[ y_t = Ax_t + B\ell_t \]

- Undersampled-ReProCS: replace \( \Phi_t \) by \( \Phi_t A \) in the \( \ell_1 \) min and LS steps [Qiu, Vaswani, Allerton’11] \(^6\)

- A similar result will hold for u-ReProCS if [Lois, Vaswani, Qiu, GlobalSIP’13]

  - \( \delta_{2s}(A) < 0.1 \), and
  - we replace \( \|I_T^TP\|_2 \) by \( \|A_T^TP\|_2 \) in the denseness assumption
  - and \( P \)'s now form a basis for the subspace of \( \bar{\ell}_t := B\ell_t \).

- The case \( y_t = A_t x_t + B_t \ell_t \) is NOT an easy extension

---

\(^6\) C. Qiu and N. Vaswani, Recursive Sparse Recovery in Large but Correlated Noise, Allerton 2011
Some Generalizations (ongoing work)

- **Extension to**
  \[ y_t = A x_t + B \ell_t \]

- **Undersampled-ReProCS**: replace \( \Phi_t \) by \( \Phi_t A \) in the \( \ell_1 \) min and LS steps [Qiu, Vaswani, Allerton’11] \(^6\)

- **A similar result will hold for u-ReProCS if** [Lois, Vaswani, Qiu, GlobalSIP’13]
  - \( \delta_{2s}(A) < 0.1 \), and
  - we replace \( \| I_T' P \|_2 \) by \( \| A'_T P \|_2 \) in the denseness assumption
    and \( P' \)'s now form a basis for the subspace of \( \bar{\ell}_t := B \ell_t \).

- **The case** \( y_t = A_t x_t + B_t \ell_t \) is NOT an easy extension

- **Direct application to online low-rank matrix completion.**

---

\(^6\) C. Qiu and N. Vaswani, Recursive Sparse Recovery in Large but Correlated Noise, Allerton 2011
Online Low-rank Matrix Completion

- Can provide a provably accurate solution for online matrix completion; that also allows highly correlated set of unknown entries
  - but requires slow subspace change and initial subspace knowledge

- Low-rank matrix completion is a special case w/ known support($x_t$)
  - in MC: $T_t$ is the set of unknown entries of $\ell_t$ at time $t$

- ReProCS for online matrix completion:
  - Compute $\Phi_t := (I - \hat{P}_{t-1} \hat{P}'_{t-1})$
  - Given $T_t$, get an estimate of $\ell_t$ as
    \[ \hat{\ell}_t = (I - (\Phi_t)T_t^\dagger)y_t \]
  - Use projection-PCA as before to update the subspace estimate
Extensions and Future Directions - I

- Replace $\ell_1$ min by *modified-CS* or *weighted-$\ell_1$* or batch sparse recovery methods
  - modified-CS [Vaswani, Lu, ISIT’09, TSP’10]: given support prediction, $T$,
    \[
    \min_x \|x_{T^c}\|_1 \text{ s.t. } \|\tilde{y}_t - \Phi_t x\|_2 \leq \xi
    \]
  - weighted-$\ell_1$ [Khajenejad et al, ISIT’09, TSP’11, Friedlander et al, T-IT’12]:
    \[
    \min_x \|x_{T^c}\|_1 + \lambda\|x_T\|_1 \text{ s.t. } \|\tilde{y}_t - \Phi_t x\|_2 \leq \xi
    \]
- ReProCS does not need slow support change but can use it to improve recovery performance
- Show Videos
Future directions

- ReProCS for online matrix completion applications
- Online robust PCA problem from moving sensors’ data, e.g. moving cameras
- Online robust sparse recovery problem from time-varying undersampled measurements $y_t := A_t x_t + B_t \ell_t + w_t$
- Proof techniques applicable to more general problems
We provide a proof outline for the ReProCS algorithm where the projection-PCA step knows the change times $t_j$ and the number of new directions $c_{j,\text{new}}$: easier case.

Let $T_t$ denote the support of $x_t$. 

Proof outline for easier case
ReProCS algorithm [Qiu, Vaswani, Allerton’10, Allerton’11]\(^7\)

Initialize: given \( \hat{P}_0 \) with \( \text{span}(\hat{P}_0) \approx \text{span}([\ell_1, \ell_2, \ldots \ell_{t_0}]) \)

For \( t > t_0 \),

- **Projection**: compute \( \tilde{y}_t := \Phi_t y_t \), where \( \Phi_t := I - \hat{P}_{t-1} \hat{P}'_{t-1} \)
  - then \( \tilde{y}_t = \Phi_t x_t + \beta_t \), \( \beta_t := \Phi_t \ell_t \) is small “noise”
  - above: standard sparse recovery problem in small noise

- **Noisy Sparse Recovery**: \( \ell_1 \) min + support estimate + LS: get \( \hat{x}_t \)
  - \( \hat{x}_{t,cs} = \arg \min_x \|x\|_1 \) s.t. \( \| \tilde{y}_t - \Phi_t x \|_2 \leq \xi \)
  - \( \hat{T}_t = \{i : |(\hat{x}_{t,cs})_i| > \omega\} \)
  - \( \hat{x}_t = I_{\hat{T}_t} (A_{\hat{T}_t} A_{\hat{T}_t}')^{-1} A_{\hat{T}_t}' y_t \)

- **Get** \( \ell_t = y_t - \hat{x}_t \)

- **Subspace update**: update \( \hat{P}_t \) every \( \alpha \) frames by projection-PCA

---

\(^7\) C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010
C. Qiu and N. Vaswani, Recursive Sparse Recovery in Large but Correlated Noise, Allerton 2011
Assume $t_{j+1} - t_j > K\alpha$; recall: $t_j$: subspace change times

$$P(t) = P_j = [P_{j-1}, P_{j,new}] \text{ for } t_j \leq t < t_{j+1}$$

- First Projection PCA
- Second Projection PCA
- Update $P_j$ when the last Projection PCA is done

K times Projection PCA

at $t = t_j + k\alpha$, compute $\hat{P}_{j,new,k}$ as the $c$ “top” left singular vectors of $(I - \hat{P}_{j-1}\hat{P}_{j-1}')[\hat{\ell}_{t_j+(k-1)\alpha}, ..., \hat{\ell}_{t_j+k\alpha-1}]$; update $\hat{P}(t) = [\hat{P}_{j-1}, \hat{P}_{j,new,k}]$
Proof Outline: Key steps

Proof outline for easier case: assume ReProCS knows $t_j, c_j, \text{new}$

- Define subspace error, $SE(P, \hat{P}) := \|(I - \hat{P}\hat{P}')P\|_2$.

- Start with $SE(P_{(j-1)}, \hat{P}_{(j-1)}) \leq r_{j-1}\zeta \ll 1$ at $t = t_j - 1$.
  1. Analyze projected sparse recovery for $t \in [t_j, t_j + \alpha)$
  2. Analyze projection-PCA at $t = t_j + \alpha - 1$
  3. Repeat for each of the $K$ projection-PCA intervals: show that $SE(P_{(j),\text{new}}, \hat{P}_{(j),\text{new}}, k) \leq 0.6^k + 0.4c\zeta$
  4. Pick $K$ s.t. $0.6^K + 0.4c\zeta \leq c\zeta$. Set $\hat{P}_{(j)} = [\hat{P}_{(j-1)}, \hat{P}_{(j),\text{new}}, k]$

- Thus, at $t = t_j + K\alpha - 1$,
  $SE(P_{(j)}, \hat{P}_{(j)}) \leq SE(P_{(j-1)}, \hat{P}_{(j-1)}) + SE(P_{(j),\text{new}}, \hat{P}_{(j),\text{new}}, k) \leq r_{j-1}\zeta + c\zeta = r_j\zeta$

- $t_{j+1} - t_j > K\alpha$ implies $SE(P_{(j)}, \hat{P}_{(j)}) \leq r_j\zeta$ at $t = t_{j+1} - 1$
Proof Outline: Projected sparse recovery for \( t \in [t_j, t_j + \alpha) \)

1. Recall: \( P_t = [P_{j-1}, P_{\text{new}}], \hat{P}_{t-1} = \hat{P}_{j-1}, \tilde{y}_t := \Phi y_t = \Phi x_t + \beta_t, \)
where \( \Phi := I - \hat{P}_{t-1} \hat{P}'_{t-1} \) and \( \beta_t := \Phi \ell_t \)

2. Using slow subspace change,
\[
\| \beta_t \|_2 \leq \sqrt{\zeta} + \sqrt{c} \gamma_{\text{new}}
\]

3. Using denseness,
\[
\delta_s(\Phi) = \kappa_s(\hat{P}_{j-1})^2 \leq \kappa_s(P_{j-1})^2 + r\zeta \leq 0.1
\]

4. Thus, \( \| \hat{x}_{t,cs} - x_t \| \leq 7\sqrt{c} \gamma_{\text{new}} \)

5. Appropriate support threshold & \( \gamma_{\text{new}} \) small \( \Rightarrow \hat{T}_t = T_t \)

6. LS step: \( e_t := x_t - \hat{x}_t = \ell_t - \ell_t \) satisfies
\[
e_t = I_{T_t}[\Phi T_t' \Phi T_t]^{-1} I_{T_t} \Phi \ell_t \quad \text{and} \quad \| e_t \|_2 \leq 1.2 \| \beta_t \|_2
\]
Proof Outline: first Projection-PCA at $t = t_j + \alpha - 1 - 1$

1. Bound $\text{SE}(P_{\text{new}}, \hat{P}_{\text{new},1})$ in terms of $\lambda_{\text{min}}$ of signal subspace part of the true data matrix, $\sum_t \Phi_0 \ell_t \ell_t' \Phi_0'$, $\lambda_{\text{max}}$ of its noise subspace part, and $\| . \|_2$ of the perturbation, $\sum_t \Phi_0 (\ell_t \ell_t' - \hat{\ell}_t \hat{\ell}_t') \Phi_0'$
   - use sin $\theta$ theorem: 1970s linear algebra result of Kahan, Davis

2. Get high probability bounds on each of the terms in this bound
   - upper bound $\| \mathbb{E}[.]. \|_2$ and/or lower bound $\lambda_{\text{min}}(\mathbb{E}[.])$ of the above matrices;
     - $\mathbb{E}[.]$: expectation conditioned on accurate recovery until before the beginning of this interval
   - bound $\| . \|_2$ of the above matrices;
   - use with matrix Hoeffding inequality [Tropp, 2012] to get desired bounds

3. Main difficulty: bounding $\| \mathbb{E}[\text{perturbation-matrix}] \|_2$
Proof Outline: first Projection-PCA at $t = t_j + \alpha - 1$.

$\|\mathbb{E}[\text{perturbation-matrix}]\|_2 \lesssim 2\|\mathbb{E}[\sum_t \ell_t e'_t]\|_2 + \|\mathbb{E}[\sum_t e_t e'_t]\|_2$

- use the expression for $e_t$,
- use the structure of $\mathbb{E}[\sum_t e_t e'_t]$ and of $\mathbb{E}[\sum_t \ell_t e'_t]$,
- simplify using the support change model and the bound on $g$.

4. Finally show $\text{SE}(P_{\text{new}}, \hat{P}_{\text{new},1}) \leq 0.6$ w.h.p.
Proof Outline: $k$-th projection PCA interval

- $P_t = [P_{j-1}, P_{\text{new}}], \ \hat{P}_{(t-1)} = [\hat{P}_{j-1}, \hat{P}_{\text{new}, k-1}]$.
- Start with $\text{SE}(P_{j-1}, \hat{P}_{j-1}) \leq r\zeta$,
  $\text{SE}(P_{\text{new}}, \hat{P}_{\text{new}, k-1}) \leq 0.6^{k-1} + 0.4c\zeta$
- Smaller SE than previous interval $\Rightarrow$ smaller $\beta_t$ $\Rightarrow$ smaller $e_t$ $\Rightarrow$
  smaller $\text{SE}(P_{\text{new}}, \hat{P}_{\text{new}, k})$ $\Rightarrow$ even smaller $\beta_t$ at next iteration
- Analyze projected sparse recovery for $t \in [t_j + (k-1)\alpha, t_j + k\alpha)$
  - show: $\|e_t\|_2 \leq 1.2\|\beta_t\|_2 \leq 1.2(\sqrt{\zeta} + 0.6^{k-1}\sqrt{c\gamma_{\text{new}}})$
- Analyze projection PCA at $t = t_j + k\alpha - 1$
  - show $\text{SE}(P_{\text{new}}, \hat{P}_{\text{new}, k}) \leq 0.6^k + 0.4c\zeta$
- Pick $K$ so that $\text{SE}(P_{\text{new}}, \hat{P}_{\text{new}, K}) \leq c\zeta$. Set $\hat{P}_j = [\hat{P}_{j-1}, \hat{P}_{j, \text{new}, K}]$
- Thus, $\text{SE}(P_j, \hat{P}_j) \leq \text{SE}(P_{j-1}, \hat{P}_{j-1}) + \text{SE}(P_{\text{new}}, \hat{P}_{\text{new}, k}) \leq r_j\zeta$
Summary

➤ To the best of our knowledge, this is the first correctness result for online sparse + low-rank recovery
  ➤ equivalently also for online robust PCA / recursive sparse recovery in large but structured noise

➤ Advantages
  ➤ online algorithm: faster; less storage needed; removes a key limitation of PCP: allows more correlated support change

➤ New proof techniques needed to obtain our results
  ➤ almost all existing robust PCA results are for batch approaches
  ➤ previous finite sample PCA results are not useful: assume $e_t := \hat{\ell}_t - \ell_t$ is uncorrelated with $\ell_t$
Algorithm parameters

Recall that $\zeta \leq \min\left(\frac{10^{-4}}{(r_0+Jc)^2f}, \frac{1}{(r_0+Jc)^3\gamma^2_*}\right)$.

- $\xi = \sqrt{c}\gamma_{\text{new}} + \sqrt{\zeta}(\sqrt{r_0 + Jc} + \sqrt{c})$;
- $\omega$ satisfies $7\xi \leq \omega \leq x_{\min} - 7\xi$;
- $K = \left\lceil \frac{\log(0.16c\zeta)}{\log(0.4)} \right\rceil$;
- $\alpha = C(\log(6KJ) + 11\log(n)), \ C \geq C_{\text{add}} := 20^2 \cdot 8 \cdot 96^2 \frac{(1.2\xi)^4}{(c\zeta\lambda^-)^2}$
- If we assume that min and max eigenvalues are seen in the training data, then can estimate $\lambda^-, \lambda^+, \gamma^*$ from training data
Most general support change model

- Split \([1, 2, \ldots, t_{\text{max}}]\) into \(\left\lceil \frac{t_{\text{max}}}{\alpha} \right\rceil\) intervals of length \(\alpha\). For a given interval, let \(T_{(i)}, \ i = 1, 2 \ldots l\) be mutually disjoint subsets of \(\{1, 2, \ldots n\}\) s.t.

\[
T_t \subseteq T_{(i)} \cup T_{(i+1)} \text{ for some } i.
\]

- Define

\[
h(\alpha) := \max_{k=1, \ldots, t_{\text{max}}} \max_{i=1, \ldots, l} |\{t \in [(k - 1)\alpha + 1 : k\alpha] : T_t \subseteq T_{(i)} \cup T_{(i+1)}\}|
\]

- We assume that \(|T_t| \leq s\) and, for a \(h^+ < 1\),

\[
h(\alpha) \leq h^+ \alpha \text{ for all } \alpha \geq \alpha_0
\]
References


4. C. Qiu and N. Vaswani, Real-time Robust Principal Components’ Pursuit, Allerton, 2010

5. C. Qiu and N. Vaswani, Recursive Sparse Recovery in Large but Correlated Noise, Allerton 2011