Compressive acquisition of linear dynamical systems

Amirafshar Moshtaghpour

May 2015
Outline

- Background
- CS-LDS Architecture
- Estimating the state sequence
- Estimating the observation matrix
- Conclusion
Outline

- **Background**
- CS-LDS Architecture
- Estimating the state sequence
- Estimating the observation matrix
- Conclusion
Background

Compressed Sensing (CS)

- Original signal: \( y \in \mathbb{R}^N \)
- \( K \)-sparse signal: \( s \in \mathbb{R}^N \)
  - \( y = \Psi s \)
  - \( s \) has at most \( K \) non-zero elements
- Measurement matrix: \( \Phi \in \mathbb{R}^{M \times N} \)
  - \( K < M \ll N \)
- Measurement vector: \( z \in \mathbb{R}^M \)
- Measurement noise: \( e \in \mathbb{R}^M \)

One possibility to recover \( y \)
\( \Phi \sim i.i.d \) Gaussian
\( M = 4K \log \frac{N}{K} \)
Background

Compressed Sensing (CS)
- Sparse Signals
- Structured-Sparse Signals
Compressed Sensing (CS)

K-sparse signals comprise a particular set of K-dim subspaces

\[ \|signal\|_0 \leq K \]

union of K-dimensional subspaces

A K-sparse **signal model** comprises a particular (reduced) set of K-dim subspaces

Compressive acquisition of linear dynamical systems
Background

**Compressed Sensing (CS) [4, 5]**

<table>
<thead>
<tr>
<th>a = CoSaMP(Φ, u, s)</th>
</tr>
</thead>
</table>

**Input:** Sampling Matrix Φ, measurement vector u, sparsity level s

**Output:** An s-sparse approximation a of the target signal

| a^0 ← 0 |
| v ← u |
| k ← 0 |

**repeat**

| k ← k + 1 |
| y ← Φ*v |

| Ω ← supp(M_2(y, s)) |

| Ω ← supp(y_{2s}) |

| T ← Ω ∪ supp(a^{k-1}) |

| b|_T ← Φ^†u |

| b|_{T^c} ← 0 |

| a^k ← b_s |

| v ← u - Φa^k |

**until** halting criterion true

**Trivial initial approximation**

**Current samples = input samples**

**Iteration index**

**Form signal proxy**

**Identify large components**

**Merge supports**

**Signal estimation by least-square**

**Prune to obtain next approximation**

**Update current samples**

Compressive acquisition of linear dynamical systems
Video compressive sensing

- $y_t$: the image of a scene at time $t$
- $Y = y_{1:T} = [y_1, \ldots, y_T]$: video of the scene from time 1 to $T$

**Goal:** to recover $y_{1:T}$ given $z_{1:T}$

1. Single Pixel Camera (SPC)
   - Duarte et al, 2008
2. Programmable Pixel Camera (P2C)
   - Hitomi et al, 2011
   - Reddy et al, 2011
   - Veeraraghavan et al, 2011

Compressive acquisition of linear dynamical systems
Background

Linear Dynamical System (LDS)

- **Dynamical system**: Change of some variables (state variables)
  - Continuous vs Discrete
  - Linear vs Non-linear

Discrete-time LDS:

\[
\begin{align*}
  x_{t+1} &= A_t x_t + B_t u_t \\
  y_t &= C_t x_t + D_t u_t
\end{align*}
\]

TI autonomous discrete-time LDS:

\[
\begin{align*}
  x_{t+1} &= A x_t \\
  y_t &= C x_t
\end{align*}
\]

- \( t \in \mathbb{R} \): time
- \( x \in \mathbb{R}^d \): state vector (variables)
- \( u \in \mathbb{R}^m \): input vector
- \( y \in \mathbb{R}^N \): observation (output) vector ≠ measurement vector
- \( A \in \mathbb{R}^{d \times d} \): state transition (dynamic) matrix
- \( B \in \mathbb{R}^{d \times m} \): input matrix
- \( C \in \mathbb{R}^{N \times d} \): observation (output or sensor) matrix
- \( D \in \mathbb{R}^{N \times m} \): feed-through matrix

Compressive acquisition of linear dynamical systems
Background

Linear Dynamical System (LDS)

- A matrix $\mathbf{H}$ is called *Hankel matrix* if the entries on the anti-diagonals be the same, i.e. $H_{i,j} = H_{i-1,j+1}$

Given $\mathbf{h} \in \mathbb{R}^N \rightarrow \text{build } \mathbf{H} \in \mathbb{R}^{L \times K}$

Hankel matrix

$$
\mathbf{H} = 
\begin{bmatrix}
    h_1 & h_2 & \cdots & h_K \\
    h_2 & h_3 & \cdots & h_{K+1} \\
    \vdots & \vdots & \ddots & \vdots \\
    h_L & h_{L+1} & \cdots & h_N \\
\end{bmatrix}
$$

$K = N - L + 1$

Given $\mathbf{Y} = y_{1:T} \in \mathbb{R}^{N \times T} \rightarrow \text{build } \mathbf{H} \in \mathbb{R}^{LN \times K}$

Block-Hankel matrix

$$
\mathbf{H} = 
\begin{bmatrix}
    \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_K \\
    \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{K+1} \\
    \vdots & \vdots & \ddots & \vdots \\
    \mathbf{y}_L & \mathbf{y}_{L+1} & \cdots & \mathbf{y}_T \\
\end{bmatrix}
$$

$K = T - L + 1$
Background

Linear Dynamical System (LDS)

\[
H = \begin{bmatrix}
y_1 & y_2 & \cdots & y_K \\
y_2 & y_3 & \cdots & y_{K+1} \\
\vdots & \vdots & \ddots & \vdots \\
y_L & y_{L+1} & \cdots & y_T \\
\end{bmatrix} = \begin{bmatrix}
Cx_1 & Cx_2 & \cdots & Cx_K \\
CAx_1 & CAx_2 & \cdots & CAx_K \\
\vdots & \vdots & \ddots & \vdots \\
CA^{L-1}x_1 & CA^{L-1}x_2 & \cdots & CA^{L-1}x_K \\
\end{bmatrix}
\]

= \begin{bmatrix}
C \\
CA \\
\vdots \\
CA^{L-1} \\
\end{bmatrix} \times \begin{bmatrix} x_1 & x_2 & \cdots & x_K \end{bmatrix}

= O(C, A)C(x),

\[H = U_dS_dV_d^T.\]

Compressive acquisition of linear dynamical systems
Background

**LDS model for video sequences**

- Challenges for video sequences:
  - Ephemeral nature of videos
  - High-dimensional signals

Few frames

Six basis frames

All frames can be estimated using linear combinations of SIX images
Outline

- Background
- CS-LDS Architecture
- Estimating the state sequence
- Estimating the observation matrix
- Conclusion
**CS-LDS Architecture**

**Authors:** A. C. Sankaranarayanan, P. K. Turaga, R. Chellappa, and R. G. Baraniuk, 2013

**Goal:** to build a CS framework, implementable on the SPC, for videos that are modeled as LDS.

- We seek to recover $\mathbf{C}$ and $\mathbf{x}_{1:T}$, given compressive measurements of the form

$$\mathbf{z}_t = \Phi_t \mathbf{y}_t = \Phi_t \mathbf{C} \mathbf{x}_t$$

- $\mathbf{z}_t \in \mathbb{R}^M$, $\Phi_t \in \mathbb{R}^{M \times N}$
- Bilinear unknowns $\rightarrow$ non-convex optimization problem

**Fig. 2.** Block diagram of the CS-LDS framework.

Compressive acquisition of linear dynamical systems
CS-LDS Architecture

\[ \frac{N}{M} = 20, \text{ SNR: 25.81 dB} \]

\[ \frac{N}{M} = 50, \text{ SNR: 24.09 dB} \]
CS-LDS Architecture

\[
\begin{align*}
\mathbf{z}_t &= \begin{bmatrix} \tilde{z}_t \\ \tilde{z}_t \end{bmatrix} = \begin{bmatrix} \tilde{\Phi} \\ \tilde{\Phi}_t \end{bmatrix} y_t, \quad \tilde{z}_t = \tilde{\Phi} C x_t, \\
\tilde{z}_t &\in \mathbb{R}^{\tilde{M}} \\
\tilde{z}_t &\in \mathbb{R}^{\tilde{M}} \\
M &= \tilde{M} + \tilde{M}
\end{align*}
\]

1. **State sequence estimation:**
   1. Build Hankel Matrix
   2. Compute SVD
   3. Compute estimated state sequences

Compressive acquisition of linear dynamical systems
2. Observation matrix estimation:

- \( \mathbf{C} \) is time-invariant
- Given \( \mathbf{Z} \) and \( \hat{\mathbf{X}} \), recover \( \mathbf{C} \)

\[
\min_{\mathbf{C}} \sum_{i=1}^{d} \left\| \Psi^T \mathbf{c}_i \right\|_1 \quad \text{s.t.} \quad \forall t, \left\| \mathbf{z}_t - \Phi_t \mathbf{C} \hat{\mathbf{x}}_t \right\|_2 \leq \epsilon,
\]

- \( \Psi \) is sparsifying basis for the columns of \( \mathbf{C} \)
Outline

- Background
- CS-LDS Architecture
- **Estimating the state sequence**
- Estimating the observation matrix
- Conclusion
Estimating the state sequence

**QS#1:** What are the sufficient conditions for reliable estimation?

*Definition:* (Observability of an LDS) An LDS is observable if the current state can be estimated from a finite number of observations (for any possible state sequence).

*Lemma:* Observable LDS($\mathbf{A}, \mathbf{C}$) $\iff$ the observability matrix $\mathbf{O}(\mathbf{A}, \mathbf{C})$ is full rank.

*Remark:* $N \gg d \rightarrow$ LDS($\mathbf{A}, \mathbf{C}$) is observable with high probability

*Lemma:* for $N > d$, the LDS($\mathbf{A}, \tilde{\Phi}\mathbf{C}$) is observable with high probability, if
  - $\tilde{M} \geq d$
  - Entries of $\tilde{\Phi}$ are i.i.d samples of a sub-Gaussian distribution.

*Sum up:* Then we can estimate state sequences by factorizing the block-Hankel matrix.
Estimating the state sequence

**QS#2:** How about $\tilde{M} = 1$? (one common measurement for each video sequence)

**Theorem:** $\tilde{M} = 1$ and the elements of $\tilde{\Phi} \in \mathbb{R}^{1 \times N}$ be i.i.d from a sub-Gaussian distribution. With high probability $\mathbf{O}(\mathbf{A}, \Phi \mathbf{C})$ is full rank if
- The state transition matrix is diagonalizable,
- Its eigenvalues and eigenvectors are unique.

**QS#3:** How about $\tilde{M} < 1$? (missing measurements in some time instants)
- We obtain common measurements at some time instants $I \subset \{1, \ldots, T\}$
- We have knowledge of $\tilde{z}_i, i \in I$
- Incomplete knowledge of the block-Hankel matrix

**Matrix completion:** $\min \text{rank } (\mathbf{H}(\tilde{z}_i))$ s.t. $i \in I$
- Non-convex

**Solution:** (Nuclear norm) $\min \|\mathbf{H}(\tilde{z}_i)\|_*$ s.t. $i \in I$
Estimating the state sequence

Accuracy of state sequence estimation from common measurements

- $T = 500, d = 10$
- Reconstruction SNR $= 10 \log_{10} \left( \frac{\sum_{t=1}^{T} \|y_t\|_2^2}{\sum_{t=1}^{T} \|y_t - \hat{y}_t\|_2^2} \right)$

Compressive acquisition of linear dynamical systems
Outline

- Background
- CS-LDS Architecture
- Estimating the state sequence
- Estimating the observation matrix
- Conclusion
Estimating the observation matrix

- Images are sparse in some domains like Wavelet and DCT.
- Smooth changes in sequential frames
  - The motion is spatially correlated.
  - The supports of frames are highly overlapping.
  - The columns of $\mathbf{C}$ captures dominant motion patterns.
  - $\mathbf{C}$ can be interpreted as a basis for the frames of the video.
  - The columns of $\mathbf{C}$ are sparse in the same domain.

$$\min_{\mathbf{c}_i} \sum_{i=1}^{d} \left\| \Psi^T \mathbf{c}_i \right\|_1 \quad \text{s.t.} \quad \forall t, \left\| \mathbf{z}_t - \Phi_t \mathbf{C} \hat{\mathbf{x}}_t \right\|_2 \leq \epsilon,$$

- Insufficient for recovering $\mathbf{C}$
  - $\hat{\mathbf{x}}_t \approx \mathbf{L}^{-1} \mathbf{x}_t$

LDS ($\mathbf{A}, \mathbf{C}, \mathbf{x}$) $\equiv$ LDS ($\mathbf{L}^{-1} \mathbf{A} \mathbf{L}, \mathbf{C} \mathbf{L}, \mathbf{L}^{-1} \mathbf{x}$)

For any invertible matrix $\mathbf{L} \in \mathbb{R}^{d \times d}$
Estimating the observation matrix

- suppose \( \mathbf{C} \) is canonical sparse: \( \Psi = \mathbf{I} \) (wlog)
- **Worst case:** disjoint sparsity pattern
- **Best case:** same sparsity pattern
- Recovering \( \mathbf{C} \) using column group sparsity

\[
\begin{align*}
(P_{\ell_2-\ell_1}) \min \sum_{i=1}^{N} \|s_i\|_2 & \quad \text{s.t } \mathbf{C} = \Psi S, \forall t, \|z_t - \Phi_t C \hat{\mathbf{x}}_t\|_2 \leq \epsilon, \\
\end{align*}
\]

- **Solver:** Model-based CoSaMP
- Value of \( \tilde{M} \):

\[
\tilde{M}T = 4dK \log(N/K) \quad \Longrightarrow \quad \tilde{M} = 4 \frac{dK}{T} \log(N/K)
\]
Estimating the observation matrix

**Model-based CoSaMP**

**Algorithm 1**: \( \hat{C} = \text{Model-based CoSAMP} (\Psi, K, z_t, \hat{x}_t, \Phi_t, t = 1, \ldots, T) \)

**Notation:**
- \( \text{supp}(\text{vec}; K) \) returns the support of \( K \) largest elements of \( \text{vec} \)
- \( A_{\Omega,} \) represents the submatrix of \( A \) with rows indexed by \( \Omega \) and all columns.
- \( A_{\cdot,\Omega} \) represents the submatrix of \( A \) with columns indexed by \( \Omega \) and all rows.

**Initialization**
- \( \forall t, \Theta_t \leftarrow \Phi_t \Psi \)
- \( \forall t, v_t \leftarrow 0 \in \mathbb{R}^M \)
- \( \Omega_{\text{old}} \leftarrow \phi \)

**while** (stopping conditions are not met) **do**
- Compute signal proxy:
  \( R = \sum_t \Theta_t^T v_t \hat{x}_t^T \)
- Compute energy in each row:
  \( r(k) = \sum_i R^2(k, i) \) for \( k \in [1, \ldots, N] \)
- Support identification and merger:
  \( \Omega \leftarrow \Omega_{\text{old}} \cup \text{supp}(r; 2K) \)
- Least squares estimation:
  - Find \( A \in \mathbb{R}^{(\Omega) \times d} \) that minimizes \( \sum_t \| z_t - (\Theta_t)_{\cdot,\Omega} A \hat{x}_t \|_2 \)
  - \( B_{\Omega,} \leftarrow A, B_{\Omega^c,} \leftarrow 0 \)
- Pruning support:
  - \( b(k) = \sum_i B^2(k, i) \) for \( k \in [1, \ldots, N] \)
  - \( \Omega \leftarrow \text{supp}(b; K), S_{\Omega,} \leftarrow B_{\Omega,}, S_{\Omega^c,} \leftarrow 0 \)
- Form new estimate of \( C \):
  \( \hat{C} \leftarrow \Psi S \)
- Update residue:
  - \( \forall t, v_t \leftarrow z_t - \Theta_t S \hat{x}_t \)
  - \( \Omega_{\text{old}} \leftarrow \Omega \)

**Compressive acquisition of linear dynamical systems**

2
Estimating the observation matrix

Ground truth

Oracle LDS: 24.97 dB

CS-LDS: 22.08 dB

Frame-to-Frame CS: 11.75 dB

\[
\frac{N}{M} = 234 \text{ for all methods}
\]

Oracle LDS:
No CS (Nyquist sampling) + knowledge of \(d\)

Sparsity: DCT, Wavelet
Meas.: Noiselet, Gaussian

Compressive acquisition of linear dynamical systems
Outline

- Background
- CS-LDS Architecture
- Estimating the state sequence
- Estimating the observation matrix
- Conclusion
Conclusion

- Not efficient to use conventional CS for video sequences
  - Ephemeral nature
  - High-dimensional

- Model video sequences as
  - Low-dimensional dynamic parameters (the state sequences)
  - High-dimensional static parameters (the observation matrix)

- Solution included
  - SVD
  - Convex optimization


[3] [Online]: CS-LDS, [www.ece.rice.edu/~as48/research/cslds](http://www.ece.rice.edu/~as48/research/cslds).


Thanks for Your Attention.

Any Question?