Incremental Gradient on the Grassmannian for Online Foreground and Background Separation in Subsampled Video

Laura Balzano sunbeam@ece.wisc.edu
Jun He (Nanjing UIST) and Arthur Szlam (CUNY)
Online Subspace Learning from Subsampled Data

Laura Balzano sunbeam@ece.wisc.edu
Jun He (Nanjing UIST) and Arthur Szlam (CUNY)
Subspace Representations: Imaging

• For each frame we have n pixels.
• The background of a collection of frames lies in a low-dimensional (d<n) subspace, possibly time-varying.
Low-Rank Matrix Completion
i.e., Subspace Learning with Subsampled Data

We receive $X$, a subset $\Omega$ of entries of a matrix.

$$\text{minimize } \text{rank}(M)$$
subject to $M_\Omega = X$
Low-Rank Matrix Completion
i.e., Subspace Learning with Subsampled Data

We receive $X$, a subset $\Omega$ of entries of a matrix.

\[
\begin{align*}
\text{minimize} & \quad \text{rank}(M) \\
\text{subject to} & \quad M_{\Omega} = X
\end{align*}
\]

• Relax this to a convex problem using the Nuclear Norm.
  – “Nuclear Norm heuristic” proposed by Fazel (2002).
Low-Rank Matrix Completion
i.e., Subspace Learning with Subsampled Data

We receive $X$, a subset $\Omega$ of entries of a matrix.

$$\begin{align*}
\text{minimize} & \quad \text{rank}(M) \\
\text{subject to} & \quad M_{\Omega} = X
\end{align*}$$

- Relax this to a convex problem using the Nuclear Norm.
  - “Nuclear Norm heuristic” proposed by Fazel (2002).
  - Recht (2007) proved this relaxation is tight and the solution is unique in certain circumstances.
  - Several algorithms have been developed to solve this SDP exactly or approximately (SVT, OPT-Space, ADMiRA, SDPLR, APGL, FPCA, GROUSE, SET, Jellyfish, DFC…).
Subsampling for speed
Suppose we receive a sequence of incomplete length-$n$ vectors that lie in a $d$-dimensional subspace $S$. Let $\Omega_t \in \{1, \ldots, n\}$ refer to the observed indices at time $t$:

$$v_{\Omega_1}, v_{\Omega_2}, \ldots, v_{\Omega_t}, \ldots \in S \subset \mathbb{R}^n$$
Subspace Identification: Problem Definition

Suppose we receive a sequence of incomplete length-$n$ vectors that lie in a $d$-dimensional subspace $S$. Let $\Omega_t \in \{1, \ldots, n\}$ refer to the observed indices at time $t$:

$v_{\Omega_1}, v_{\Omega_2}, \ldots, v_{\Omega_t}, \cdots \in S \subset \mathbb{R}^n$

For each vector we wish to reduce the subspace approximation error on the observed entries:

$$F_{\text{grouse}}(S; t) = \min_w \| U_{\Omega_t} w - v_{\Omega_t} \|_2^2$$

where the orthonormal columns of $U$ span $S$ and $U_{\Omega_t}$ refers to the rows of $U$ indexed by $\Omega_t$. 
GROUSE (Balzano, Recht, Nowak)

- Given step size $\eta_t$, subspace basis $U_t \in \mathbb{R}^{n \times d}$, observations $v_{\Omega_t}$

- Calculate Weights:
  $$w = \arg \min_a \| U_{\Omega_t} a - v_{\Omega_t} \|_2^2$$

- Predict full vector: $v_\parallel = U_t w$

- Compute Residual on observed entries: $v_\perp = v_{\Omega_t} - (v_\parallel)_{\Omega_t}$ and zero-pad.

- Update subspace:
  $$U_{t+1} = U_t + \left( \sin(\sigma \eta_t) \frac{v_\perp}{\| v_\perp \|} + (\cos(\sigma \eta_t) - 1) \frac{v_\parallel}{\| v_\parallel \|} \right) \frac{w^T}{\| w \|}$$
  where $\sigma = \| v_\perp \| \| v_\parallel \|$

- One iteration involves a projection and an outer product.
- The algorithm is *simple* and *fast*. 

We receive $Xv$ a subset $\Omega$ of entries of a matrix.
Background Subtraction

Figure 8: Real-time video background and foreground separation from partial information. We show the separation quality at $t = 1, 230, 1400$. The resolution of the video is $144 \times 176$. The first row is the original video frame at each time; the middle row is the recovered background at each time only from 5r information; and bottom row is the foreground calculated by Equation $u4.7v$.

Figure 9: Real-time video background and foreground separation from partial information. We show the separation quality at $t = 1, 600, 1200$. The resolution of the video is $320 \times 256$. The first row is the original video frame at each time; the middle row is the recovered background at each time only from 1r information; and bottom row is the foreground calculated by Equation $u4.7v$. 
Robust Low-Rank Modeling (Robust PCA)

• Sparse + Low-Rank Model

• Several algorithms have been developed to find such a decomposition from a matrix observation
  – convex optimization and approximations
The augmented Lagrangian of this constrained minimization problem is then
detailed in a fast solver based on the technique of ADMM (Alternating Direction Method of Multipliers).
Boyd [q] has a nice survey of algorithms to solve this problem and describes in more references can be found there.

2.2 Subspace Error Quantification by GROUSE algorithm as we demonstrate in Section nhkh.

It was shown in [o] that this cost function gives an accurate estimate of the same cost function with
arbitrarily large and
are those of

\[ F_{\text{grouse}}(\mathcal{S}; t) = \min_w \| U_{\Omega_t} w - v_{\Omega_t} \|_2 \]

\[ F_{\text{grasta}}(\mathcal{S}; t) = \min_w \| U_{\Omega_t} w - v_{\Omega_t} \|_1 \]

\[ U_{t+1} = U_t + \left( (\cos(\sigma \eta_t) - 1) U_t \frac{w_t}{\| w_t \|} + \sin(\sigma \eta_t) \frac{\Gamma}{\| \Gamma \|} \right) \frac{w_t^T}{\| w_t \|} \]
### GRASTA Performance on Background Subtraction

#### Table

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution</th>
<th>Total Frames</th>
<th>Training Time</th>
<th>Tracking and Separating Time</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport Hall</td>
<td>144 × 176</td>
<td>3584</td>
<td>11.3 sec</td>
<td>20.9 sec</td>
<td>171.5</td>
</tr>
<tr>
<td>Shopping Mall</td>
<td>320 × 256</td>
<td>1286</td>
<td>33.9 sec</td>
<td>27.5 sec</td>
<td>46.8</td>
</tr>
<tr>
<td>Lobby</td>
<td>144 × 176</td>
<td>1546</td>
<td>3.9 sec</td>
<td>71.3 sec</td>
<td>21.7</td>
</tr>
<tr>
<td>Hall with Virtual Pan (1)</td>
<td>144 × 88</td>
<td>3584</td>
<td>3.8 sec</td>
<td>191.3 sec</td>
<td>18.7</td>
</tr>
<tr>
<td>Hall with Virtual Pan (2)</td>
<td>144 × 88</td>
<td>3584</td>
<td>3.7 sec</td>
<td>144.8 sec</td>
<td>24.8</td>
</tr>
</tbody>
</table>

#### Figures

- **Figure 1**: Real-time video background and foreground separation.
- **Figure 2**: Demonstration of panning the "virtual camera" right and left through the video to simulate a dynamic background.
- **Table**: Summary of performance metrics including resolution, total frames, training time, tracking and separating time, and FPS.

- **Performance Metrics**: The table entries show the resolution in pixels, total frames, training time, and tracking and separating time, along with the frames per second (FPS) for each dataset.
GRASTA Performance on Background Subtraction

**Figure 8:** Real-time video background and foreground separation from partial information.

**Figure 9:** Table showing the separation quality at each time only from 55 information; and bottom row is the foreground calculated by Equation 1.

**Figure 0:** Demonstration of panning the "virtual camera" right to left and back through the video to simulate the camera pans.

<table>
<thead>
<tr>
<th>Time (Frames)</th>
<th>Airport Hall</th>
<th>Lobby</th>
<th>Hall with Virtual Pan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>883 sec</td>
<td>683 sec</td>
<td>7,30 sec</td>
</tr>
<tr>
<td>100</td>
<td>144 sec</td>
<td>144 sec</td>
<td>191 sec</td>
</tr>
<tr>
<td>1,000</td>
<td>793 sec</td>
<td>6,638 sec</td>
<td>753 sec</td>
</tr>
<tr>
<td>10,000</td>
<td>793 sec</td>
<td>6,638 sec</td>
<td>753 sec</td>
</tr>
</tbody>
</table>

Our model shows how GRASTA can quickly adapt to the new camera position after adjusting to a new camera scope of the virtual camera to be half the width, so the resolution of the virtual camera is 320 x 160. The total computation time is 6.7 FPS.

In the last experience, we demonstrate that GRASTA can effectively track the pixels for separation, whereas IALM and GRASTA will adapt the background to incorporate the unchanging "foreground."
Figure 83 Real-time video background and foreground separation

Table 63 Real-time video background and foreground separation by GRASTA

Dynamic Background: Virtual Pan

Figure 73 ROC curves for benchmark videos. Grasta25 does not perform subspace updates after the background is trained.

Shopping Mall

Airport Hall

Bootstrap

Dataset

Lobby

Resolution

699 × 472

875 × 609

675 × 453

699 × 472

699 × 472

699 × 472

6,709 frames

85,000 frames

6,709 frames

Figure 32 Demonstration of panning the “virtual camera” right with Virtual Pan.

While IALM and GRASTA will adapt the background in just 0.18 s, ReProCS predicts the sparse part at time $t$ estimated by GRASTA.

Hall with Virtual Pan

variable $w_{Ax}$ with $w_{Bx}$ $w_{Cx}$ $w_{Dx}$ $w_{Ex}$ $w_{Fx}$ $w_{Gx}$ $w_{Hx}$ $w_{Ix}$ $w_{Jx}$ $w_{Kx}$ $w_{Lx}$ $w_{Mx}$ $w_{Nx}$ $w_{Ox}$ $w_{Px}$ $w_{Qx}$ $w_{Rx}$ $w_{Sx}$ $w_{Tx}$ $w_{Ux}$ $w_{Vx}$ $w_{Wx}$ $w_{Xx}$ $w_{Yx}$ $w_{Zx}$

You can also let GRASTA track and do the separation task for all frames. The virtual camera pans adapt to the changed background in just 0.24 s. When we use 0.15 of the pixels for the tracking and 0.655 of the pixels for separation, the total computation time is 0.017 seconds.

When we use 0.655 of the pixels for the tracking and 0.15 of the pixels for the separation, the total computation time is 0.0258 seconds. The virtual camera pans position takes around 0.05 frames or 0.0024 s and adjusting to a new camera position after 0.025 s may let GRASTA track.

The virtual camera pans dimension to 0.3 times the scope of the virtual camera to be half the width. The virtual camera pans from left to right and back through the video to simulate a dynamic background. The idea of the virtual camera is illustrated with Figure 03.

3 We set the subspace $w_{Sub2}$ and sampling $w_{Sampling}$.

In the last experiment, we demonstrate that GRASTA can effectively track and separation.
Table 63 Real-time video background and foreground separation by GRASTA

Here we use four different resolution video datasets, the first three experiments they are the first 05 frames. We train from 05 frames, in the first three experiments they are.

We choose “Airport Hall” as the dataset. We set the subspace dimension to 03. Figure 03 Demonstration of panning the “virtual camera” right to left and back through the video to simulate.

The first row is the original video frame; the middle row is the recovered background from only the pixels for separation; the total computation time is.

When we use 05t of the pixels for tracking and 655t of the pixels for separation, the total computation time is.

We also let GRASTA track and do the separation task for all the pixels to the right at position takes around 05 frames.

consider panning a “virtual camera” periodically by 75 pix. The right subspace in video with a dynamic background.

We demonstrate that GRASTA can effectively track.

Hall with Virtual Pan

Figure 83 Real-time video background and foreground separation

The subspace dimension is 0 for all three with static background and the last three with dynamic background.

Figure 4. Real-time video background and foreground separation from partial information. The first row is the original video frame; the middle row is the tracked background; the bottom row is the separated foreground.
GRASTA Performance on Background Subtraction

Virtual camera panning right 20 pixels

$t = 100$  $t = 101$  $t = 105$  $t = 110$  $t = 115$  $t = 120$  $t = 125$
GRASTA demo

• Open CV demo written by Arthur Szlam.
For more information: sites.google.com/site/hejunzz/grasta

Incremental Gradient on the Grassmannian for Background and Foreground Separation in Subsampled Video
Jun He, Laura Balzano, and Arthur Szlam. CVPR, June 2012.

Online Robust Subspace Tracking from Partial Information

Handling Missing Data in High-Dimensional Subspace Modeling

Online Identification and Tracking of Subspaces from Highly Incomplete Information
Laura Balzano, Robert Nowak, and Benjamin Recht. Allerton, September 2010.

High-Dimensional Matched Subspace Detection when Data are Missing
Laura Balzano, Benjamin Recht, and Robert Nowak. ISIT, June 2010.