Recovery of simultaneous low rank and two-way sparse coefficient matrices

Mladen Kolar (mkolar@chicagobooth.edu)
Collaborators

Ming Yu (Chicago Booth)

Varun Gupta (Chicago Booth)

Zhaoran Wang (Northwestern)
Motivation — Multi-task learning

Learning multiple related tasks leads to better statistical performance compared to learning the tasks separately.

We consider the following linear regression multi-task learning setting

\[ Y = X\Theta^* + E, \]

where

- \( Y \in \mathbb{R}^{n \times k} \) is a matrix of responses
- \( X \in \mathbb{R}^{n \times p} \) is a matrix of predictors
- \( \Theta^* \in \mathbb{R}^{p \times k} \) is an unknown parameter matrix
- \( E \in \mathbb{R}^{n \times k} \) is an error matrix with i.i.d. mean zero and variance \( \sigma^2 \) entries

Relatedness of tasks is modeled through structural assumptions on the matrix \( \Theta^* \).
Motivation — Multi-task learning

In a high-dimensional setting, with large number of variables, it is common to assume that there are a few variables predictive of all tasks, while others are not predictive

- Turlach et al. (2005), Obozinski et al. (2011), Lounici et al. (2011), Kolar et al. (2011), Wang et al. (2016b)

\[
\hat{\Theta} = \arg \min_{\Theta} \frac{1}{2n} \| Y - X\Theta \|^2_F + \sum_{j \in [p]} \text{pen}(\Theta_j)
\]

where \(\text{pen}(\cdot)\) is usually \(\ell_2\) or \(\ell_\infty\) norm.
Motivation — Multi-task learning

Another way to relate tasks is to assume that predictors lie in a shared lower dimensional subspace

- Ando and Zhang (2005), Amit et al. (2007), Yuan et al. (2007), Argyriou et al. (2008), Wang et al. (2016a)

That is, $\Theta^*$ is assumed to be a low rank matrix.

Bunea et al. (2011) show optimality for the following reduced rank estimator

$$
\hat{\Theta} = \arg \min \frac{1}{2n} \| Y - X\Theta \|_F^2 + \lambda \cdot \text{rank}(\Theta),
$$

which can be efficiently computed using SVD (Reinsel and Velu, 1998).
Motivation — Multi-task learning

More commonly, one uses a relaxation of the rank constraint.

\[ \hat{\Theta} = \arg \min_{\Theta} \frac{1}{2n} \| Y - X\Theta \|_F^2 + \lambda \cdot \| \Theta \|_* , \]

where \( \| \Theta \|_* = \sum_{j=1}^{\text{rank}(\Theta)} \sigma_j(\Theta) \) is the nuclear norm.

See, for example, (Candès and Recht, 2009, Chandrasekaran et al. (2011), Koltchinskii et al. (2011), Harchaoui et al. (2012), Negahban and Wainwright (2011), . . . )
Sparse reduced rank regression

In contemporary applications it is increasingly common that both the number of predictors and the number of tasks is large compared to the sample size.

- In a study of regulatory relationships between genome-wide measurements, where micro-RNA measurements are used to explain the gene expression levels, a small number of micro-RNAs regulate genes participating in few regulatory pathways (Ma et al., 2014a).

$\Theta^*$ is assumed to be both sparse and low rank.

- predictors can be combined into fewer latent features that drive the variation in the multiple response variables and are composed only of relevant predictor variables
- Bunea et al. (2012), Chen et al. (2012), Chen and Huang (2012), She (2017)
More applications

Sparse SVD

- Chen et al. (2012), Ma et al. (2014a), Yang et al. (2014), ...

Biclustering:

- Lee et al. (2010), Balakrishnan et al. (2011), Balakrishnan et al. (2017)
Optimization over sparse and low-rank matrices

We consider a statistical model with true parameter $\Theta^* \in \Omega$, where $\Omega \subset \mathbb{R}^{m_1 \times m_2}$ is a nonconvex set comprising of low rank matrices that are also row and/or column sparse,

$$\Omega = \Omega(r, s_1, s_2) = \{\Theta \mid \text{rank}(\Theta) \leq r, \|\Theta\|_{2,0} \leq s_1, \|\Theta^T\|_{2,0} \leq s_2\},$$

with $\|\Theta\|_{2,0}$ is the number of non-zero rows of $\Theta$.

To estimate $\Theta^*$, we minimize an empirical loss function

$$\hat{\Theta} \in \arg\min_{\Theta \in \Omega} f(\Theta)$$

over the set $\Omega$.

Not clear how to do a convex relaxation for the set $\Omega$. 
Optimization over sparse and low-rank matrices

We write $\Theta = UV^\top$ with $U \in \mathbb{R}^{m_1 \times r}$, $V \in \mathbb{R}^{m_2 \times r}$ and consider the following optimization problem

$$(\hat{U}, \hat{V}) \in \arg \min_{U \in \mathcal{U}, V \in \mathcal{V}} f(U, V),$$

where

$$\mathcal{U} = \mathcal{U}(s_1) = \{U : \|U\|_{2,0} \leq s_1\},$$
$$\mathcal{V} = \mathcal{V}(s_2) = \{V : \|V\|_{2,0} \leq s_2\}.$$

- $\hat{U}$ and $\hat{V}$ are only unique up to rotation: $(\hat{U}R, \hat{V}R)$ is also a solution for any orthogonal matrix $R$. 

Burer-Monteiro factorization for low rank matrices

Low-rank Matrix Recovery

\[
\min_{\Theta \in \mathbb{R}^{m_1 \times m_2}} f(\Theta) \quad \text{subject to} \quad \text{rank}(\Theta) \leq r,
\]

Convex relaxation

\[
\min_{\Theta \in \mathbb{R}^{m_1 \times m_2}} f(\Theta) + \lambda \|\Theta\|_*.
\]

Nonconvex approach

- Write \( \Theta = UV^\top \) with \( U \in \mathbb{R}^{m_1 \times r} \) and \( V \in \mathbb{R}^{m_2 \times r} \) and minimize

\[
\min_{U,V} f(U, V)
\]
Burer-Monteiro factorization in practice

More efficient than solving convex relaxation

Good performance with good objective function and initialization

Nonconvex optimization in theory

We use the penalty function $g(U, V)$ defined as

$$g(U, V) = \frac{1}{4} \| U^\top U - V^\top V \|^2_F,$$

which forces $U$ and $V$ to be balanced (Zheng and Lafferty, 2015).

We now consider the following problem

$$(\hat{U}, \hat{V}) \in \arg \min_{U \in \mathcal{U}, V \in \mathcal{V}} f(U, V) + g(U, V).$$

The solution will be the same as the previous problem.
Subspace distance.

Denote \( Z = [U; V] \), \( Z^\ast = [U^\ast; V^\ast] \) with \( U^\ast(V^\ast)^\top = \Theta^\ast \) and 
\( U^\ast(U^\ast)^\top = V^\ast(V^\ast)^\top \), we define the subspace distance as:

\[
d(Z, Z^\ast) = \min_{R \in \mathbb{Q}_r} \left\{ \|U - U^\ast R\|_F + \|V - V^\ast R\|_F \right\},
\]

where \( \mathbb{Q}_r \) denotes the set of \( r \)-by-\( r \) orthogonal matrixes.

We will show that \( d(Z^t, Z^\ast) \) converges linearly up to statistical error.
Algorithm 1 Gradient Descent with Hard Thresholding (GDT)

1. **Input:** Initial estimate $\tilde{\Theta}$
2. **Parameters:** Step size $\eta$, Rank $r$, Sparsity $s_1, s_2$, Number of iterations $T$
3. $(\tilde{U}, \tilde{\Sigma}, \tilde{V}) = \text{rank} \ r \ \text{SVD of} \ \tilde{\Theta}$
4. $U^0 = \text{Hard}(\tilde{U}(\tilde{\Sigma})^{\frac{1}{2}}, s_1), V^0 = \text{Hard}(\tilde{V}(\tilde{\Sigma})^{\frac{1}{2}}, s_2)$
5. **for** $t = 1$ **to** $T$ **do**
6. $V^{t+0.5} = V^t - \eta \nabla_V f(U^t, V^t) - \eta \nabla_V g(U^t, V^t)$
7. $V^{t+1} = \text{Hard}(V^{t+0.5}, s_2)$
8. $U^{t+0.5} = U^t - \eta \nabla_U f(U^t, V^t) - \eta \nabla_U g(U^t, V^t)$
9. $U^{t+1} = \text{Hard}(U^{t+0.5}, s_1)$
10. **end for**
11. **Output:** $\Theta^T = U^T(V^T)^\top$
Hyperparameters

Rank $r$

- Using ideas from Bunea et al. (2011).

Sparsity levels $s_1$, $s_2$

- Use $s_1 = c \cdot s_1^*$ and $s_2 = c \cdot s_2^*$ with some $c > 1$.
- Information criteria, such as She (2017).
- Not very sensitive to the choice of $c$.

Our algorithm does not require tuning parameters that need to be selected carefully other than the rank, which is required for most of the methods.
Assumptions

Restricted Strong Convexity and Smoothness (RSC/RSS)

There exist universal constants $\mu$ and $L$ such that

$$\frac{\mu}{2} \| \Theta_2 - \Theta_1 \|_F^2 \leq f(\Theta_2) - f(\Theta_1) - \langle \nabla f(\Theta_1), \Theta_2 - \Theta_1 \rangle \leq \frac{L}{2} \| \Theta_2 - \Theta_1 \|_F^2$$

for all $\Theta_1, \Theta_2 \in \Omega(2r, \tilde{s}_1, \tilde{s}_2)$ where $\tilde{s}_1 = (2c + 1)s_1^*$ and $\tilde{s}_2 = (2c + 1)s_2^*$. 
Assumptions

Initialization (I)

Define $\mu_{\min} = \frac{1}{8} \min\{1, \frac{\mu L}{\mu + L}\}$ and

$$I_0 = \frac{4}{5} \mu_{\min} \sigma_r(\Theta^*) \cdot \min\left\{ \frac{1}{\mu + L}, 2 \right\}.$$  

We require

$$\|\Theta^0 - \Theta^*\|_F \leq \frac{1}{5} \min\left\{ \sigma_r(\Theta^*), \frac{I_0}{\xi} \sqrt{\sigma_r(\Theta^*)} \right\},$$

where $\xi^2 = 1 + \frac{2}{\sqrt{c - 1}}$.  

Assumptions

We define the notion of the statistical error,

\[ e_{\text{stat}} = \sup_{\Delta \in \Omega(2r, \tilde{s}_1, \tilde{s}_2), \|\Delta\|_F \leq 1} \langle \nabla f(\Theta^*), \Delta \rangle. \]

**Step Size Selection:** We choose the step size \( \eta \) to satisfy

\[ \eta \leq \frac{1}{16\|Z_0\|_2^2} \cdot \min \left\{ \frac{1}{2(\mu + L)}, 1 \right\}. \]

Furthermore, we require \( \eta \) and \( c \) to satisfy

\[ \beta = \xi^2 \left( 1 - \eta \cdot \frac{2}{5} \mu_{\min} \sigma_r(\Theta^*) \right) < 1, \]

and

\[ e_{\text{stat}}^2 \leq \frac{1 - \beta}{\xi^2 \eta} \cdot \frac{L\mu}{L + \mu} \cdot f_0^2. \]
Key Lemma

Suppose the conditions \((\text{RSC/RSS}), \, (I)\) are satisfied. Assume that the point \(Z = \begin{bmatrix} U \\ V \end{bmatrix}\) satisfies \(d(Z, Z^*) \leq l_0\). Let \((U^+, V^+)\) denote the next iterate obtained with GDT with the step size \(\eta\) satisfying

\[
\eta \leq \frac{1}{8\|Z\|_2^2} \cdot \min \left\{ \frac{1}{2(\mu + L)}, 1 \right\}.
\]

Then we have

\[
d^2(Z^+, Z^*) \leq \xi^2 \left[ \left(1 - \eta \cdot \frac{2}{5}\mu_{\min}\sigma_r(\Theta^*)\right) \cdot d^2(Z, Z^*) + \eta \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2 \right],
\]

where \(\xi^2 = 1 + \frac{2}{\sqrt{c-1}}\).
Main Result

Suppose the conditions (RSC/RSS), (I) are satisfied and the step size \( \eta \) satisfies the conditions stated before. Then after \( T \) iterations of GDT, we have

\[
d^2(Z^T, Z^*) \leq \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1 - \beta} \cdot \frac{L + \mu}{L \cdot \mu} \cdot e^{2}_{\text{stat}}.\]

Furthermore, for \( \Theta^T = U^T (V^T)^\top \) we have

\[
\|\Theta^T - \Theta^*\|_F^2 \leq 4\sigma_1(\Theta^*) \cdot \left[ \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1 - \beta} \cdot \frac{L + \mu}{L \cdot \mu} \cdot e^{2}_{\text{stat}} \right].
\]

▶ Our analysis also works for optimization problem without statistical model, where we replace true values \( U^*, V^* \) with global minimum \( \hat{U}, \hat{V} \). If we further assume no sparsity, the statistical error is 0.
Application to Multi-task Learning

Recall that we are interested in a multi-task learning problem

$$Y = X\Theta^* + E,$$

where

- $Y \in \mathbb{R}^{n \times k}$ is a matrix of responses
- $X \in \mathbb{R}^{n \times p}$ is a matrix of predictors
- $\Theta^* \in \mathbb{R}^{p \times k}$ is an unknown parameter matrix
- $E \in \mathbb{R}^{n \times k}$ is an error matrix with i.i.d. mean zero and variance $\sigma^2$ entries

The objective function is

$$f(U, V) = \frac{1}{2n} \|Y - XUV^\top\|_F^2$$

with $U \in \mathbb{R}^{p \times r}$ and $V \in \mathbb{R}^{k \times r}$ with $U \in \mathcal{U}(s_1)$ and $V \in \mathcal{U}(s_2)$. 
We assume $X$ satisfies the Restricted Eigenvalue (RE) condition (Negahban et al., 2012) for some constant $\kappa(s_1)$ and $\bar{\kappa}(s_1)$

$$\kappa(s_1) \cdot \|\theta\|_2^2 \leq \frac{1}{n} \|X\theta\|_2^2 \leq \bar{\kappa}(s_1) \cdot \|\theta\|_2^2$$

for all $\|\theta\|_0 \leq s_1$, which implies that the (RSC/RSS) condition is satisfied.

Initialization is done using a lasso estimator. The condition (I) is effectively a requirement on the sample size.
Application to Multi-task Learning

Suppose all the conditions are satisfied, for all

\[ T \geq C_1 \log \left[ \frac{n}{(s_1^* + s_2^*)(r + \log(p \lor k))} \right], \]

with probability at least \( 1 - (p \land k)^{-1} \), we have

\[ \| \Theta^T - \Theta^* \|_F \leq C_\sigma \sqrt{\frac{(s_1^* + s_2^*)(r + \log(p \lor k))}{n}} \]

for some constant \( C_1 \) and \( C \).
Application to Multi-task Learning

We compare the error rate

$$\sigma \sqrt{\frac{1}{n} (s_1^* + s_2^*)(r + \log(p \lor k))}$$

with the minimax rate established in (Ma et al., 2014b):

$$\sigma \sqrt{\frac{1}{n} \left[ (s_1^* + s_2^*)r + s_1^* \log \frac{ep}{s_1^*} + s_2^* \log \frac{ek}{s_2^*} \right]}$$

They match up to a $\log(p \lor k)$ factor. When $r$ is comparable to $\log(p \lor k)$ they match up to a constant multiplier.

For large enough $T$, GDT algorithm attains near optimal rate.
Application to Multi-task Learning

If we consider row sparsity only, then we have $s^*_2 = k$ and

$$\|\Theta^T - \Theta^*\|_F \leq C\sigma \sqrt{\frac{kr + s^*_1(r + \log p)}{n}}.$$  

This gives prediction error

$$\|X\Theta^T - X\Theta^*\|_F^2 \leq C\sigma^2 \left( kr + s^*_1(r + \log p) \right).$$

GDT error matches the prediction error $(k + s^*_1 - r)r + s^*_1 \log p$ provided in (She, 2017), as long as $k \geq Cr$ which is typically satisfied.
Experiment

Linear convergence
Simulation

Comparison with other methods

- Double Projected Penalization (DPP) — Ma et al. (2014b)
- thresholding SVD method (TSVD) — Ma et al. (2014a)
- exclusive extraction algorithm (EEA) — Chen et al. (2012)
- RCGL and JRRS — Bunea et al. (2012)
- standard Multitask learning method (MTL, with $L_{2,1}$ penalty)

Setup: $n = 50$, $p = 100$, $k = 50$, $r = 8$, $s_1^* = s_2^* = 10$.

For the methods that rely on a tuning parameter $\lambda$, we generate an independent validation set to select the “best” $\lambda$.

For our method, we use $s_1 = 2s_1^*$ and $s_2 = 2s_2^*$. 
<table>
<thead>
<tr>
<th></th>
<th>Estimation error</th>
<th>Prediction error</th>
<th>Row support</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GDT</strong></td>
<td>0.0488 ± 0.0103</td>
<td>1.1043 ± 0.0144</td>
<td>20 ± 0</td>
</tr>
<tr>
<td><strong>DPP</strong></td>
<td>0.0588 ± 0.0148</td>
<td>1.1079 ± 0.0155</td>
<td>48.96 ± 8.29</td>
</tr>
<tr>
<td><strong>TSVD</strong></td>
<td>0.3169 ± 0.1351</td>
<td>2.4158 ± 0.9899</td>
<td>25.62 ± 8.03</td>
</tr>
<tr>
<td><strong>EEA</strong></td>
<td>0.3053 ± 0.0998</td>
<td>1.2349 ± 0.0362</td>
<td>84.28 ± 6.70</td>
</tr>
<tr>
<td><strong>RCGL</strong></td>
<td>0.0591 ± 0.0148</td>
<td>1.1101 ± 0.0168</td>
<td>49.60 ± 10.62</td>
</tr>
<tr>
<td><strong>JRRS</strong></td>
<td>0.0877 ± 0.0227</td>
<td>1.1857 ± 0.0214</td>
<td>12.26 ± 2.02</td>
</tr>
<tr>
<td><strong>MTL</strong></td>
<td>0.0904 ± 0.0243</td>
<td>1.1753 ± 0.0204</td>
<td>73.40 ± 2.67</td>
</tr>
</tbody>
</table>
## Simulation

### Table 2: Row sparse and column sparse

<table>
<thead>
<tr>
<th></th>
<th>Estimation error</th>
<th>Prediction error</th>
<th>Row support</th>
<th>Column support</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GDT</strong></td>
<td>0.087 ± 0.023</td>
<td>1.062 ± 0.014</td>
<td>20 ± 0</td>
<td>20 ± 0</td>
</tr>
<tr>
<td><strong>DPP</strong></td>
<td>0.098 ± 0.028</td>
<td>1.044 ± 0.014</td>
<td>51.3 ± 13.9</td>
<td>10.2 ± 0.5</td>
</tr>
<tr>
<td><strong>TSVD</strong></td>
<td>0.335 ± 0.105</td>
<td>1.760 ± 0.341</td>
<td>28.6 ± 7.2</td>
<td>30.9 ± 8.5</td>
</tr>
<tr>
<td><strong>EEA</strong></td>
<td>0.260 ± 0.115</td>
<td>1.102 ± 0.022</td>
<td>64.4 ± 9.8</td>
<td>12.1 ± 2.7</td>
</tr>
<tr>
<td><strong>RCGL</strong></td>
<td>0.121 ± 0.032</td>
<td>1.107 ± 0.017</td>
<td>42.0 ± 7.9</td>
<td>50 ± 0</td>
</tr>
<tr>
<td><strong>JRRS</strong></td>
<td>0.168 ± 0.041</td>
<td>1.161 ± 0.017</td>
<td>13.9 ± 4.6</td>
<td>50 ± 0</td>
</tr>
<tr>
<td><strong>MTL</strong></td>
<td>0.183 ± 0.049</td>
<td>1.165 ± 0.016</td>
<td>73.5 ± 3.1</td>
<td>50 ± 0</td>
</tr>
</tbody>
</table>
Simulation

- Increase $n, p, s_1^*, s_2^*$ by a factor of $\zeta$
- Increase $k, r$ by a factor of $\lceil \sqrt{\zeta} \rceil$

Table 3: Running time comparison (in seconds)

<table>
<thead>
<tr>
<th></th>
<th>$\zeta = 1$</th>
<th>$\zeta = 5$</th>
<th>$\zeta = 10$</th>
<th>$\zeta = 20$</th>
<th>$\zeta = 50$</th>
<th>$\zeta = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GDT</strong></td>
<td>0.11</td>
<td>0.20</td>
<td>0.51</td>
<td>2.14</td>
<td>29.3</td>
<td>235.8</td>
</tr>
<tr>
<td><strong>DPP</strong></td>
<td>0.19</td>
<td>0.61</td>
<td>3.18</td>
<td>17.22</td>
<td>315.4</td>
<td>2489</td>
</tr>
<tr>
<td><strong>TSVD</strong></td>
<td>0.07</td>
<td>1.09</td>
<td>6.32</td>
<td>37.8</td>
<td>543</td>
<td>6075</td>
</tr>
<tr>
<td><strong>EEA</strong></td>
<td>0.50</td>
<td>35.6</td>
<td>256</td>
<td>$&gt;2h$</td>
<td>$&gt;2h$</td>
<td>$&gt;2h$</td>
</tr>
<tr>
<td><strong>RCGL</strong></td>
<td>0.18</td>
<td>1.02</td>
<td>7.15</td>
<td>36.4</td>
<td>657.4</td>
<td>$&gt;2h$</td>
</tr>
<tr>
<td><strong>JRRS</strong></td>
<td>0.19</td>
<td>0.82</td>
<td>6.36</td>
<td>30.0</td>
<td>610.2</td>
<td>$&gt;2h$</td>
</tr>
<tr>
<td><strong>MTL</strong></td>
<td>0.18</td>
<td>3.12</td>
<td>30.92</td>
<td>184.3</td>
<td>$&gt;2h$</td>
<td>$&gt;2h$</td>
</tr>
</tbody>
</table>
In vivo Calcium Imaging Data

When a neuron fires an electrical action potential, calcium will enter the cell and then its fluorescent properties.

By recording the movies of this dynamic it allows us to identify the spiking activity from large populations of neurons.

Spatiotemporal model introduced by (Pnevmatikakis et al. (2014))
In vivo Calcium Imaging Data

Observation field: \( k = \ell_1 \times \ell_2 \) pixels

The field contains a total number of (possibly overlapping) \( r \) neurons

Let \( c_i \) denote the calcium activity for each neuron \( i \), it follows AR(1) model:

\[
c_i(t) = \gamma c_i(t - 1) + s_i(t),
\]

where \( s_i(t) \) is the number of spikes that neuron \( i \) fired at time \( t \) and \( \gamma = 1 - 1/(\text{frame rate}) \).

Let \( a_i \) denote the spatial footprint vector for neuron \( i \), our observation at each time step \( t \) is

\[
y(t) = \sum_{i=1}^{K} a_i c_i(t) + \epsilon_t.
\]
In vivo Calcium Imaging Data

In matrix form we can rewrite as

\[ S = GC \]
\[ Y = CA + E \]

with

\[ G = \begin{pmatrix}
1 & 0 & \ldots & 0 \\
-\gamma & 1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \ldots & -\gamma & 1
\end{pmatrix}. \]

Here \( C \in \mathbb{R}^{T \times r} \), \( G \in \mathbb{R}^{T \times T} \), \( S \in \mathbb{R}^{T \times r} \), \( Y \in \mathbb{R}^{T \times k} \) and \( A \in \mathbb{R}^{r \times k} \).
In vivo Calcium Imaging Data

Combine them together we obtain

$$Y = G^{-1}SA + E = X\Theta^* + E$$

where $X = G^{-1}$ is observed and $\Theta^* = SA$ is the coefficient matrix.

$A$ should be row sparse since the area for neurons in the monitored area is small.

$S$ should be column sparse since neurons do not fire very frequently.

$\Theta^*$ is low rank by construction since the number of neurons are usually small.
In vivo Calcium Imaging Data

Multi-task learning problem with simultaneous row-sparse, column-sparse and low rank coefficient matrix where \( n = p = T \) and \( k = \ell_1 \times \ell_2 \).

The dataset is a movie with 559 frames (acquired at approximately 8.64 frames/sec), where each frame is \( 135 \times 131 \) pixels.

We have \( n = p = 559 \) and \( k = 135 \times 131 = 17,685 \).

We use \( r = 50 \), more conservative than the estimator given by (Bunea et al., 2011) and we set \( s_1 = 100 \) row sparsity and \( s_2 = 3000 \) column sparsity.
In vivo Calcium Imaging Data

Figure 1: Manually selected top 5 labeled regions

Figure 2: Corresponding signals estimated by our GDT algorithm
Conclusion

Nonconvex optimization on simultaneous low rank and two-way sparse coefficient matrix

GDT algorithm: alternating gradient descent with hard thresholding converges linearly to statistical error

For multi-task learning, statistical error is near optimal compared to the minimax rate

Better estimation accuracy and much faster running speed
References


