Introduction

ℓ₀-induced Sparse Subspace Clustering (ℓ₀-SSC)

Approximate ℓ₀-SSC

Results

ℓ₀-Sparse Subspace Clustering

Yingzhen Yang¹, Jiashi Feng², Nebojsa Jojic³, Jianchao Yang⁴, Thomas S. Huang¹

¹ Beckman Institute, University of Illinois at Urbana-Champaign, USA
² Department of ECE, National University of Singapore, Singapore
³ Microsoft Research, USA
⁴ Snapchat, USA
Introduction

- Sparse Subspace Clustering (SSC) aims to partition the data according to their underlying subspaces.

Figure 1: Black dots and red dots indicate the data that lie in subspace $S_1$ and $S_2$ respectively.
Sparse Subspace Clustering (SSC) aims to partition the data according to their underlying subspaces.

SSC and its robust version solve the following sparse representation problems:

\[
\begin{align*}
\min_{\alpha} & \quad \|\alpha\|_1 \quad \text{s.t.} \quad X = X\alpha, \quad \text{diag}(\alpha) = 0 \\
\min_{\alpha} & \quad \|X - X\alpha\|_F^2 + \lambda_1 \|\alpha\|_1 \quad \text{s.t.} \quad \text{diag}(\alpha) = 0
\end{align*}
\]

Under certain assumptions on the underlying subspaces and the data, \(\alpha\) satisfies Subspace Detection Property (SDP): its nonzero elements correspond to the data that lie in the same subspace as point \(x_i\).
\(\ell^0\)-induced Sparse Subspace Clustering

- Subspace Detection Property (SDP) is crucial for its success: data belonging to different subspaces are disconnected in the sparse graph.

![Figure 2: Block-diagonal similarity matrix due to SDP](image)

- We propose \(\ell^0\)-induced Sparse Subspace Clustering (\(\ell^0\)-SSC), which solves the \(\ell^0\) problem:

\[
\min_{\alpha} \|\alpha\|_0 \quad s.t. \ X = X\alpha, \ \text{diag}(\alpha) = 0
\]
Models for Analyzing the Subspace Detection Property

- **Deterministic Model:** the subspaces and the data in each subspace are fixed.
- **Randomized Model:**
  - **Semi-Random Model:** the subspaces are fixed but the data are distributed at random in each of the subspaces.
  - **Full-Random Model:** the subspaces and the data of each subspace are random.
\( \ell^0 \)-induced Sparse Subspace Clustering

- The sparse subspace clustering literature does not have the answer to the fundamental problem: what is the relationship between sparse representation and SDP?
- Almost surely equivalence between \( \ell^0 \)-sparsity and SDP, under the mildest assumption to the best of our knowledge.

**Theorem 1 (\( \ell^0 \)-sparsity \( \Rightarrow \) SDP)**

*Under semi-random or full-random model, suppose data in each subspace are generated i.i.d. according to any continuous distribution. Then with probability 1 over the data for semi-random model, or over both the data and the subspaces for the full-random model, the optimal solution to the \( \ell^0 \) sparse representation problem satisfies the subspace detection property.*
$\ell^0$-induced Sparse Subspace Clustering

- Inter-subspace hyperplane: the hyperplane spanned by data from different subspaces. The source where the confusion comes from.
- Key element in the proof: the probability of the intersection of the inter-subspace hyperplane and any associated subspace is 0.

Figure 3: Illustration of a inter-subspace hyperplane spanned by $x_i$ and $x_j$. 
\( \ell^0 \)-induced Sparse Subspace Clustering (\( \ell^0 \)-SSC)

Compared to previous subspace clustering methods, \( \ell^0 \)-SSC achieves SDP under far less restrictive assumptions on both the underlying subspaces and the random data generation.

<table>
<thead>
<tr>
<th>Assumption on Subspaces</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 ): Independent Subspaces</td>
<td>( \dim[S_1 \oplus S_2 \ldots S_K] = \sum_k \dim[S_k] )</td>
</tr>
<tr>
<td>( S_2 ): Disjoint Subspaces</td>
<td>( S_k \cap S_{k'} = 0 ) for ( k \neq k' )</td>
</tr>
<tr>
<td>( S_3 ): Overlapping Subspaces</td>
<td>( 1 \leq \dim[S_k \cap S_{k'}] &lt; \min{\dim[S_k], \dim[S_{k'}]} ) for ( k \neq k' )</td>
</tr>
<tr>
<td>( S_4 ): Distinct Subspaces (( \ell^0 )-SSC)</td>
<td>( S_k \neq S_{k'} ) for ( k \neq k' )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assumption on Random Data Generation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_1 ): Semi-Random Model or Full-Random Model</td>
<td>i.i.d. uniformly on the unit sphere.</td>
</tr>
<tr>
<td>( D_2 ): IID (( \ell^0 )-SSC)</td>
<td>i.i.d. from arbitrary continuous distribution.</td>
</tr>
</tbody>
</table>

No requirement for other complex geometric conditions, such as ingradius and subspace incoherence.

![Figure 4: Independent (left) and disjoint (right) subspaces](image_url)
\( \ell^0 \)-induced Sparse Subspace Clustering (\( \ell^0 \)-SSC)

- No free lunch! The price we pay for SDP under such much milder assumptions is solving the NP-hard \( \ell^0 \) problem.
- No better deal! The converse of Theorem 1:

**Theorem 2 (No free lunch: SDP \( \Rightarrow \ell^0 \)-sparsity)**

*Under the semi-random or full-random model and the assumptions of Theorem 1, if there is an algorithm which, for any data point \( x_i \in S_k, 1 \leq i \leq n, 1 \leq k \leq K \), can find the data from the same subspace as \( x_i \) that linearly represent \( x_i \), i.e.*

\[
x_i = X\beta \quad (\beta_i = 0)
\]

*where nonzero elements of \( \beta \) correspond to the data that lie in the subspace \( S_k \). Then, with probability 1, solution to the \( \ell^0 \) problem (for \( x_i \)) can be obtained from \( \beta \) in \( O(\hat{n}^3) \) time, where \( \hat{n} \) is the number of nonzero elements in \( \beta \).*
Approximate $\ell^0$-SSC (A$\ell^0$-SSC)

- Allowing for some tolerance to noise, the optimization problem of $\ell^0$-SSC is

$$
\min_{\alpha \in \mathbb{R}^{n \times n}, \text{diag}(\alpha) = 0} L(\alpha) = \|X - X\alpha\|_F^2 + \lambda \|\alpha\|_0
$$

- Optimization by proximal gradient descent, using SSC as initialization

$$
\alpha^{i(t)} = h\sqrt{\frac{2\lambda}{T_S}} (\alpha^{i(t-1)} - \frac{2}{T_S} (X^\top X\alpha^{i(t-1)} - X^\top x_i))
$$

where $h$ is an element-wise hard thresholding operator.
Approximate $\ell^0$-SSC

- The objective value $\{L(\alpha^{i(t)})\}_t$ is non-increasing and consequently it converges.

- But does $\{\alpha^{i(t)}\}_t$ converge?

- If $\{\alpha^{i(t)}\}_t$ converges, how far is the resultant sub-optimal solution from the globally optimal solution?
Approximate $\ell^0$-SSC

- Definition of sparse eigenvalues

$$\kappa-(m) := \min_{\|u\|_0 \leq m; \|u\|_2 = 1} \|Xu\|_2^2 \quad \kappa+(m) := \max_{\|u\|_0 \leq m; \|u\|_2 = 1} \|Xu\|_2^2$$

**Proposition 1**

If $\kappa-(|\text{supp}(\alpha^{i(0)})|) > 0$, $\{\alpha^{i(t)}\}_t$ is a bounded sequence that converges to a critical point of $L$, denoted by $\hat{\alpha}^i$. 
Approximate $\ell^0$-SSC

- Now how far is $\hat{\alpha}^i$ from $\alpha^{i*}$ (the globally optimal solution)?

- Roadmap: prove that both are local solutions to a capped-$\ell^1$ problem, and then we can obtain the following bound:

**Theorem 3**

*Bounded distance between sub-optimal solution and the globally optimal solution*

Under certain assumptions on the sparse eigenvalues of the data matrix, the sequence $\{\alpha^{i(t)}\}_t$ converges to a critical point of $L(\alpha^i), \hat{\alpha}^i$. Then

$$
\| (\hat{\alpha}^i - \alpha^{i*}) \|_2^2 \leq \frac{2}{(\kappa_- (|\hat{S}_i \cup \hat{S}^*_i|) - \kappa)^2}

( \sum_{j \in \hat{S}_i} (\max\{0, \frac{\lambda}{b} - \kappa|\hat{\alpha}_{i,j}^i - b|\}^2 + |\hat{S}^*_i \setminus \hat{S}_i| (\max\{0, \frac{\lambda}{b} - \kappa b\})^2 )
$$
Approximate $\ell^0$-SSC

- Remember that

$$\alpha^i(t) = h \sqrt{\frac{2\lambda}{\tau s}} \big(\alpha^i(t-1) - \frac{2}{\tau s} (X^\top X \alpha^i(t-1) - X^\top x_i)\big)$$

**Proposition 2**

If $s > \max\{2|\text{supp}(\alpha^i(0))|, \frac{2(1+\lambda|\text{supp}(\alpha^i(0))|)}{\lambda \tau}\}$, then

$$\text{supp}(\alpha^i(t)) \subseteq \text{supp}(\alpha^i(t-1)), t \geq 1$$

- Significantly reduces computational cost with efficient optimization:

$$\min_{\alpha \in \mathbb{R}^n, \alpha^i = 0} \|x_i - X\alpha^i\|_2^2 + \lambda \|\alpha^i\|_0 \Leftrightarrow \min_{\alpha \in \mathbb{R}^n, \alpha^i = 0} \|x_i - X_{S_i} \alpha^i\|_2^2 + \lambda \|\alpha^i\|_0$$
Approximate $\ell^0$-SSC

Algorithm 1 (Data Clustering by A$\ell^0$-SSC)

**Input:**
- The data set $X = \{x_i\}_{i=1}^n$, the number of clusters $c$, the parameter $\lambda$ for A$\ell^0$-SSC, maximum iteration number $M$, stopping threshold $\varepsilon$.

1. Obtain the sub-optimal solution $\tilde{\alpha}$ by proximal gradient descent.

2. Build the sparse similarity matrix by symmetrizing $\tilde{\alpha}$: $\tilde{W} = \frac{|\tilde{\alpha}| + (\tilde{\alpha}^\top)}{2}$

3. Apply spectral clustering method to $\tilde{W}$.

**Output:** The cluster labels.
### Clustering Results

**Table 1: Clustering Results on Various Image Data Sets**

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Measure</th>
<th>KM</th>
<th>SC</th>
<th>SSC</th>
<th>SMCE</th>
<th>SSC-OMP</th>
<th>$\ell^0$-SSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST (random sampling)</td>
<td>AC</td>
<td>0.5621</td>
<td>0.4922</td>
<td>0.4948</td>
<td>0.5784</td>
<td>0.5754</td>
<td>0.6590</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.5113</td>
<td>0.4755</td>
<td>0.5210</td>
<td>0.6332</td>
<td>0.5463</td>
<td>0.6709</td>
</tr>
<tr>
<td>COIL-20</td>
<td>AC</td>
<td>0.6554</td>
<td>0.4278</td>
<td>0.7854</td>
<td>0.7549</td>
<td>0.3389</td>
<td>0.8472</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.7630</td>
<td>0.6217</td>
<td>0.9148</td>
<td>0.8754</td>
<td>0.4853</td>
<td>0.9428</td>
</tr>
<tr>
<td>COIL-100</td>
<td>AC</td>
<td>0.4996</td>
<td>0.2835</td>
<td>0.5275</td>
<td>0.5639</td>
<td>0.1667</td>
<td>0.7683</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.7539</td>
<td>0.5923</td>
<td>0.8041</td>
<td>0.8064</td>
<td>0.3757</td>
<td>0.9182</td>
</tr>
<tr>
<td>Extended Yale-B</td>
<td>AC</td>
<td>0.0954</td>
<td>0.1077</td>
<td>0.7850</td>
<td>0.3293</td>
<td>0.6529</td>
<td>0.8480</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.1258</td>
<td>0.1485</td>
<td>0.7760</td>
<td>0.3812</td>
<td>0.7024</td>
<td>0.8612</td>
</tr>
<tr>
<td>UMIST Face</td>
<td>AC</td>
<td>0.4275</td>
<td>0.4052</td>
<td>0.4904</td>
<td>0.4487</td>
<td>0.4835</td>
<td>0.6730</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.6426</td>
<td>0.6159</td>
<td>0.6885</td>
<td>0.6696</td>
<td>0.6310</td>
<td>0.7924</td>
</tr>
<tr>
<td>CMU PIE</td>
<td>AC</td>
<td>0.0845</td>
<td>0.0729</td>
<td>0.2287</td>
<td>0.1733</td>
<td>0.0821</td>
<td>0.2591</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.1884</td>
<td>0.1789</td>
<td>0.3659</td>
<td>0.3343</td>
<td>0.1494</td>
<td>0.4435</td>
</tr>
<tr>
<td>AR Face</td>
<td>AC</td>
<td>0.2752</td>
<td>0.2957</td>
<td>0.5914</td>
<td>0.3543</td>
<td>0.4229</td>
<td>0.6086</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.5941</td>
<td>0.6248</td>
<td>0.8060</td>
<td>0.6973</td>
<td>0.6835</td>
<td>0.8117</td>
</tr>
<tr>
<td>MPIE S1</td>
<td>AC</td>
<td>0.1164</td>
<td>0.1285</td>
<td>0.5892</td>
<td>0.1721</td>
<td>0.1695</td>
<td>0.6741</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.5049</td>
<td>0.5292</td>
<td>0.7653</td>
<td>0.5514</td>
<td>0.3395</td>
<td>0.8622</td>
</tr>
<tr>
<td>MPIE S2</td>
<td>AC</td>
<td>0.1315</td>
<td>0.1410</td>
<td>0.6994</td>
<td>0.1898</td>
<td>0.2093</td>
<td>0.7527</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4834</td>
<td>0.5128</td>
<td>0.8149</td>
<td>0.5293</td>
<td>0.4292</td>
<td>0.8939</td>
</tr>
<tr>
<td>MPIE S3</td>
<td>AC</td>
<td>0.1291</td>
<td>0.1459</td>
<td>0.6316</td>
<td>0.1856</td>
<td>0.1787</td>
<td>0.7050</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4811</td>
<td>0.5185</td>
<td>0.7858</td>
<td>0.5155</td>
<td>0.3415</td>
<td>0.8750</td>
</tr>
<tr>
<td>MPIE S4</td>
<td>AC</td>
<td>0.1308</td>
<td>0.1463</td>
<td>0.6803</td>
<td>0.1823</td>
<td>0.1680</td>
<td>0.7246</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.4866</td>
<td>0.5280</td>
<td>0.8063</td>
<td>0.5294</td>
<td>0.3345</td>
<td>0.8837</td>
</tr>
<tr>
<td>Georgia Face</td>
<td>AC</td>
<td>0.4987</td>
<td>0.5187</td>
<td>0.5413</td>
<td>0.6053</td>
<td>0.4733</td>
<td>0.6187</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>0.6856</td>
<td>0.7014</td>
<td>0.6968</td>
<td>0.7394</td>
<td>0.6622</td>
<td>0.7400</td>
</tr>
</tbody>
</table>
Parameter Sensitivity

Figure 5: The performance change with varying $\lambda$ on Extended Yale B
Parameter Sensitivity

Figure 6: The performance change with varying $\lambda$ on COIL-20
Summary

- **Theory:** Almost surely equivalence between $\ell^0$-sparsity and the subspace detection property, under the mildest assumption to the best of our knowledge.

- **Practice:** Implemented by both MATLAB and CUDA C++ for extreme efficiency, with effectiveness evidenced by extensive experiments.
Thank you!