Subspace Clustering

Shihua Zhang

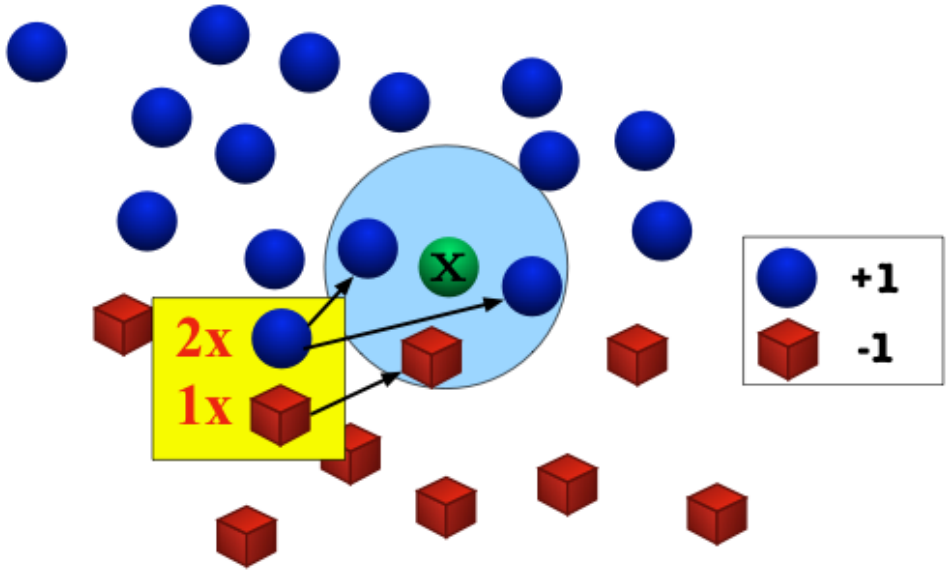
Fall 2019

Overview

Outline

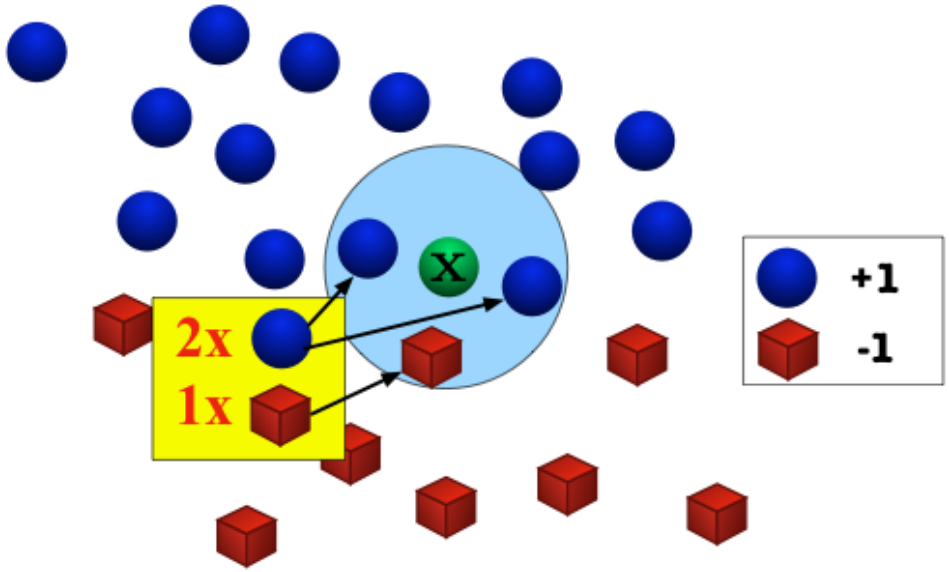
Let's Begin with *k*-NN

***k*-nearest neighbors algorithm (*k*-NN)** [Cover et al., 1967]:



Let's Begin with *k*-NN

***k*-nearest neighbors algorithm (*k*-NN)** [Cover et al., 1967]:



- **Assumption:** Similar inputs have similar outputs
- **Classification rule:** For a test input x, assign the most common label amongst its k most similar training inputs

Bayes Optimal Classifier

Bayes optimal classifier:

- Assume we know $P(y|\boldsymbol{x})$. What would you predict?
- **Examples:** $y \in \{-1, 1\}$, $P(+1|\boldsymbol{x}) = 0.8$, $P(-1|\boldsymbol{x}) = 0.2$.
- **Best prediction:** $y^* = h_{\text{opt}} = \arg\max_y P(y|\boldsymbol{x})$
- You predict the most likely class.

Bayes Optimal Classifier

Bayes optimal classifier:

- Assume we know $P(y|\boldsymbol{x})$. What would you predict?
- **Examples:** $y \in \{-1, 1\}$, $P(+1|\boldsymbol{x}) = 0.8$, $P(-1|\boldsymbol{x}) = 0.2$.
- **Best prediction:** $y^* = h_{\text{opt}} = \arg\max_y P(y|\boldsymbol{x})$
- You predict the most likely class.

Error of the BayesOpt classifier:

$$\epsilon_{BayesOpt} = 1 - P(h_{\text{opt}}(\boldsymbol{x})|\boldsymbol{x}) = 1 - P(y^*|\boldsymbol{x})$$

- In our example, that is $\epsilon_{BayesOpt} = 0.2$.
- You can never do better than the Bayes Optimal Classifier.

1-NN Convergence Proof

Theorem 1

As $n \to \infty$, the 1-NN error is no more than twice the error of the Bayes Optimal classifier. (Similar guarantees hold for $k > 1$.)

1-NN Convergence Proof

Theorem 1

As $n \to \infty$, the 1-NN error is no more than twice the error of the Bayes Optimal classifier. (Similar guarantees hold for $k > 1$.)

Proof.

Let $\boldsymbol{x}_{NN}$ be the nearest neighbor of our test point $\boldsymbol{x}_t$. As $n \to \infty$, $\text{dist}(\boldsymbol{x}_{NN}, \boldsymbol{x}) \to 0$, i.e. $\boldsymbol{x}_{NN} \to \boldsymbol{x}_t$. Then we return the label of $\boldsymbol{x}_{NN}$. What is the probability that this is not the label of $\boldsymbol{x}$?

$$\begin{aligned}
&\epsilon_{NN} = P(y^*|\boldsymbol{x}_t)(1 - P(y^*|\boldsymbol{x}_{NN})) + P(y^*|\boldsymbol{x}_{NN})(1 - P(y^*|\boldsymbol{x}_t)) \\
&\leqslant (1 - P(y^*|\boldsymbol{x}_{NN}) + (1 - P(y^*|\boldsymbol{x}_t)) = 2(1 - P(y^*|\boldsymbol{x}_t)) = 2\epsilon_{\text{BayesOpt}}.
\end{aligned}$$

Test 1-NN on CIFAR-10

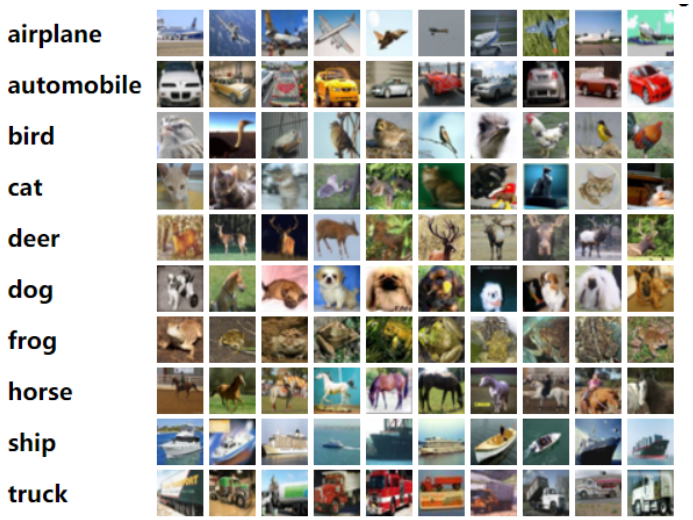
Good news: As $n \to \infty$, the 1-NN classifier is only a factor 2 worse than the best possible classifier. It seems that all of classification problems can be solved perfectly via k-NN!

Test 1-NN on CIFAR-10

Good news: As $n \to \infty$, the 1-NN classifier is only a factor 2 worse than the best possible classifier. It seems that all of classification problems can be solved perfectly via k-NN!

Let's have a try on the CIFAR-10 Dataset:

- 60000 32×32 colour images in 10 classes (6000 per class).
- There are 50000 training images and 10000 test images.



Test 1-NN on CIFAR-10

Actual performance:

- The accuracy on training set: 100%!
- The accuracy on testing set: 38.6%!

Test 1-NN on CIFAR-10

Actual performance:

- The accuracy on training set: 100%!
- The accuracy on testing set: 38.6%!

What's wrong with k-NN?

Test 1-NN on CIFAR-10

Actual performance:

- The accuracy on training set: 100%!
- The accuracy on testing set: 38.6%!

What's wrong with k-NN?

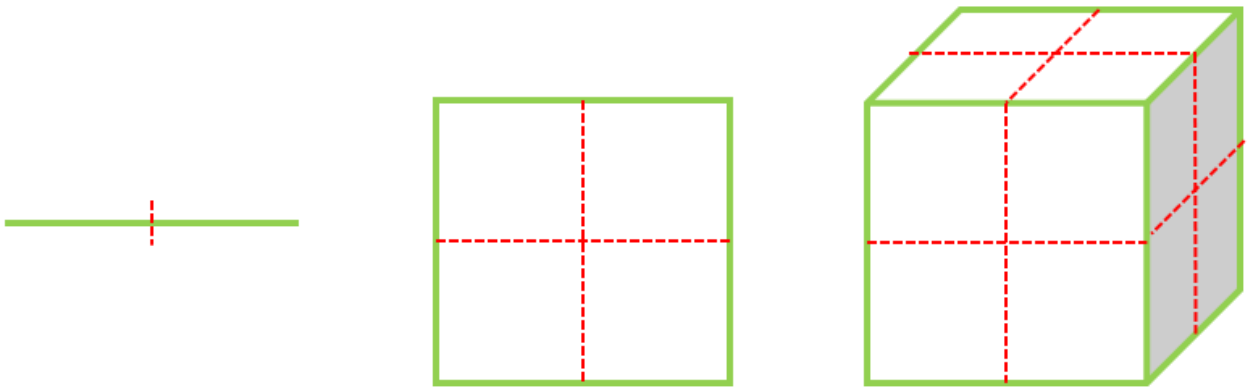
We are cursed by high dimensionality!

- Insufficient Data
- Invalid Distance
- Redundant Information

Insufficient Data

60000 images in CIFAR-10, still insufficient?

Problem: Given a cubic in $\mathbb{R}^n$, divide each dimension into 2 parts and put a data point in each sub-cubic. How many data points do we need?

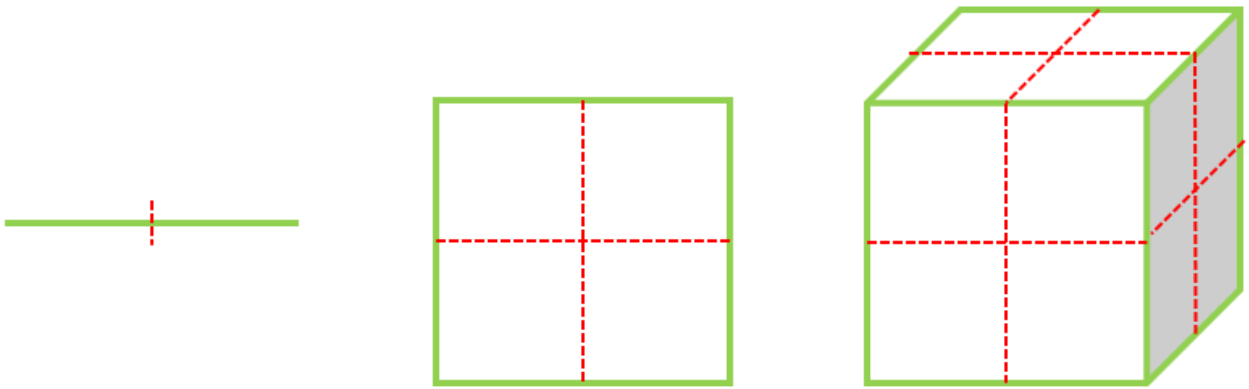


2 when $n = 1$ 4 when $n = 2$ 8 when $n = 3$

Insufficient Data

60000 images in CIFAR-10, still insufficient?

Problem: Given a cubic in $\mathbb{R}^n$, divide each dimension into 2 parts and put a data point in each sub-cubic. How many data points do we need?



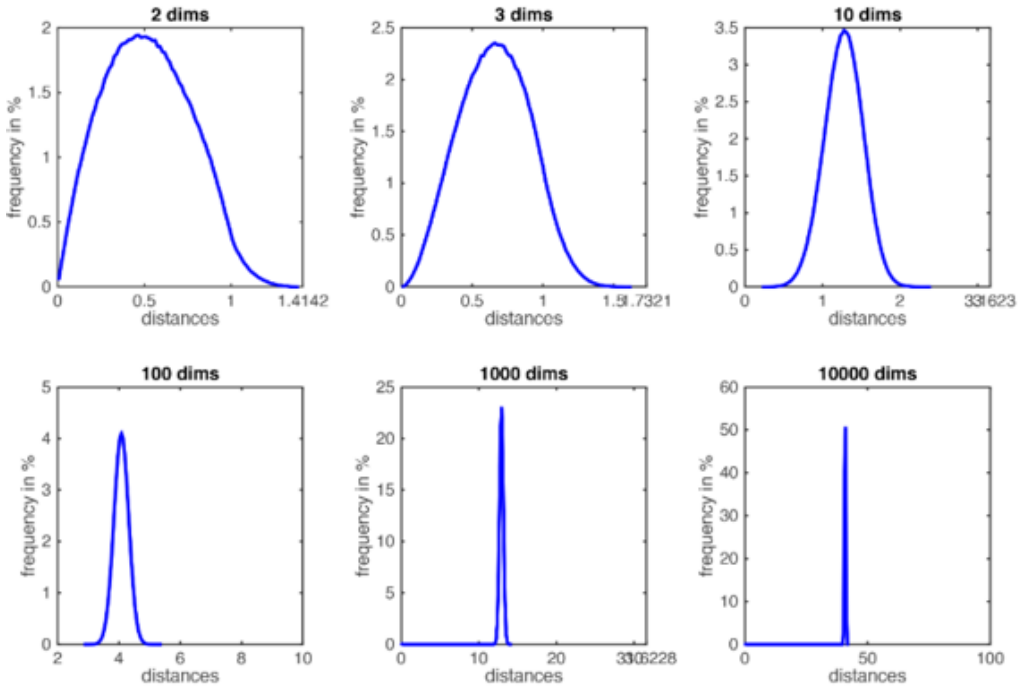
2 when $n = 1$ 4 when $n = 2$ 8 when $n = 3$

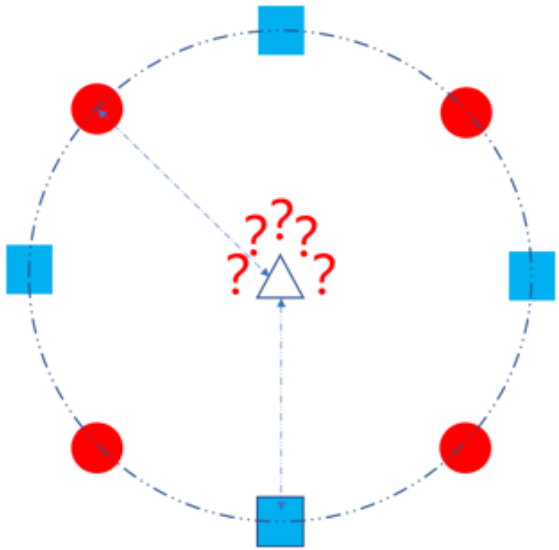
What if $n = 32 \times 32 \times 3$?

We need $2^{3072} \simeq 10^{900}$ **data points!!!**

Invalid Distance

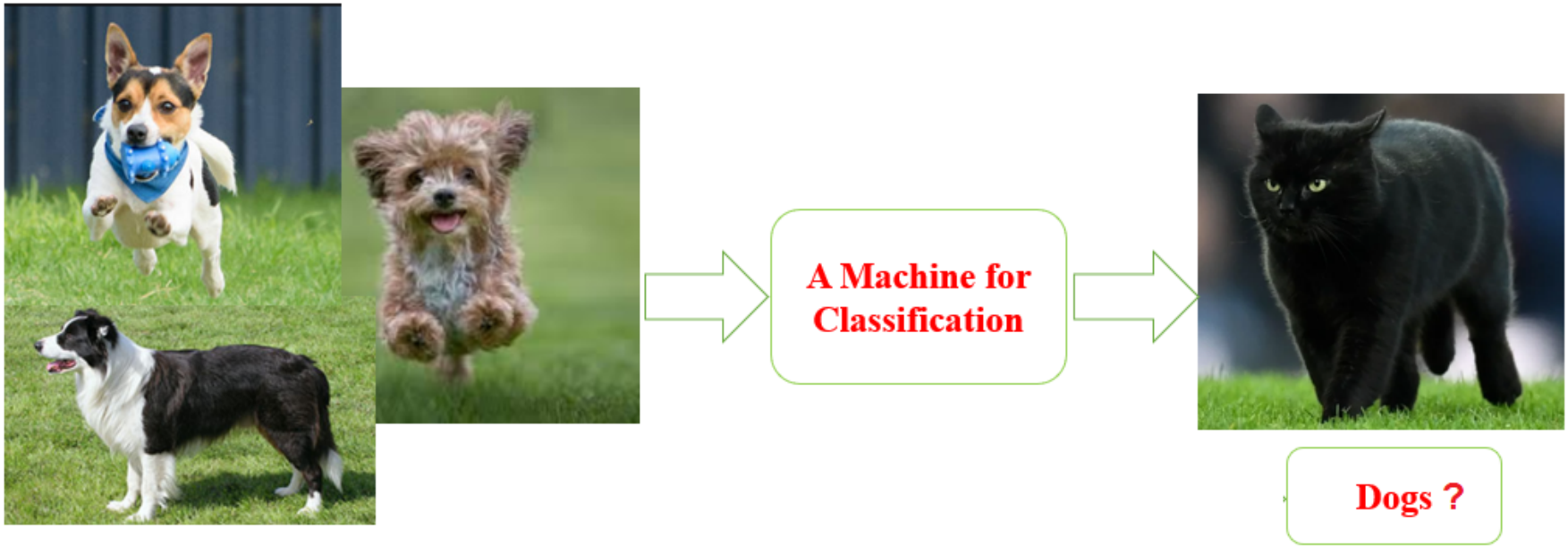
As the number of dimensions n grows, all distances concentrate within a very small range.





Traditional Euclidean distance will loss its meaning!

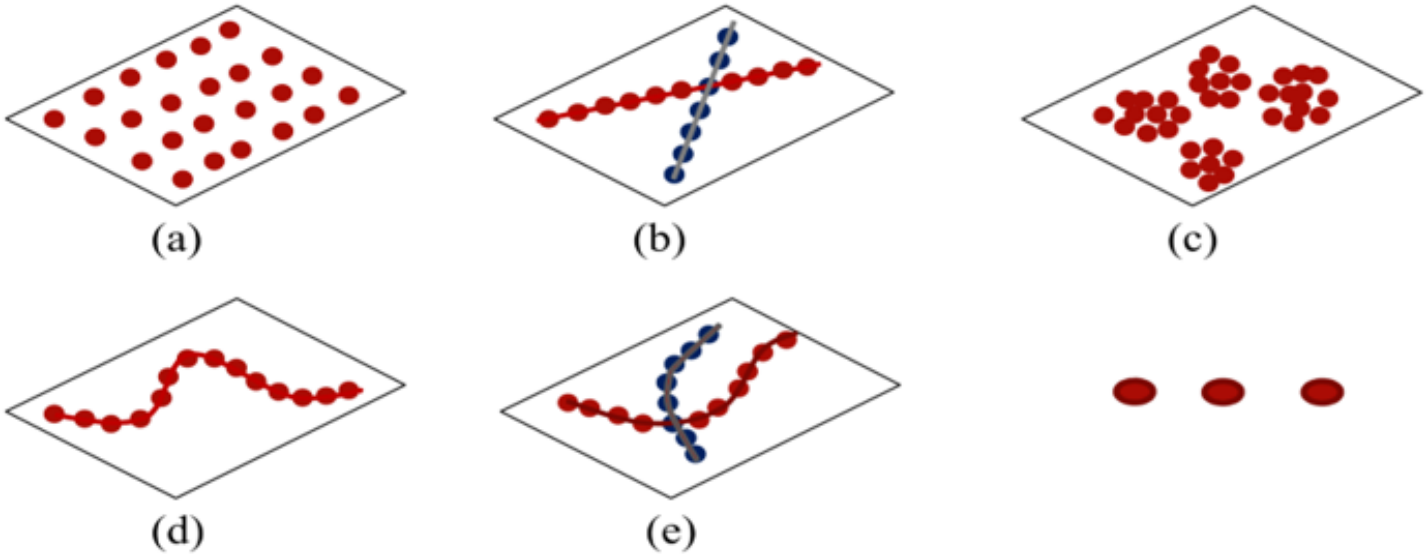
Redundant Information



Redundant information may ‘dilute’ the main information!

Blessings of Dimensionality

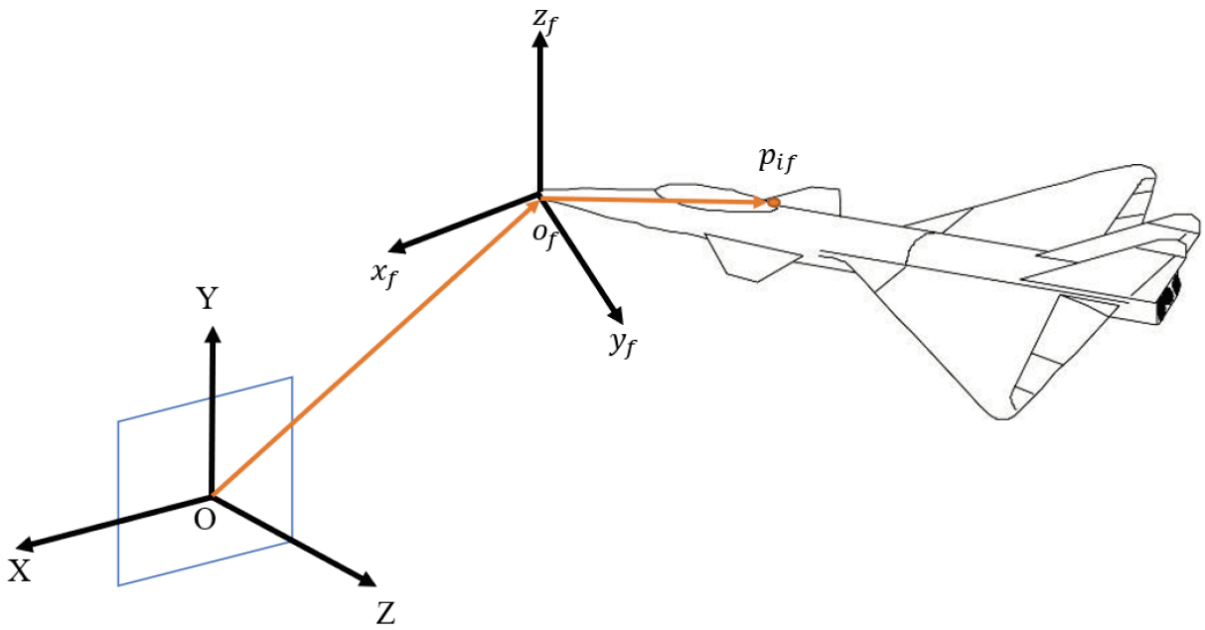
- **Bad news:** We are cursed!!
- **Good news:** Low dimensionality structure is hidden from high dimensionality
  - Principal Component Analysis
  - **Subspace Clustering**
  - Manifold Learning
  - ...



Motion Segmentation

Let us consider how a static camera observes a single moving object:

- Let $\boldsymbol{p}_{if}$ be the 2D coordinate of point $\boldsymbol{p}_i$ at f-th frame.
- $\boldsymbol{x}_i = [\boldsymbol{p}_{i1}, \boldsymbol{p}_{i2}, ..., \boldsymbol{p}_{iF}]^T$ represents the trajectory of point $\boldsymbol{p}_i$ in F frames.



Motion Segmentation

Let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]_{2F \times N}$ be the data matrix representing trajectories of N points. Then we have:

$$\text{rank}(X) = 4$$

In other words, despite each point in X with dimension of $2F$, the ambient dimension is only 4 [Costeira and Kanade, 1995]!

Motion Segmentation

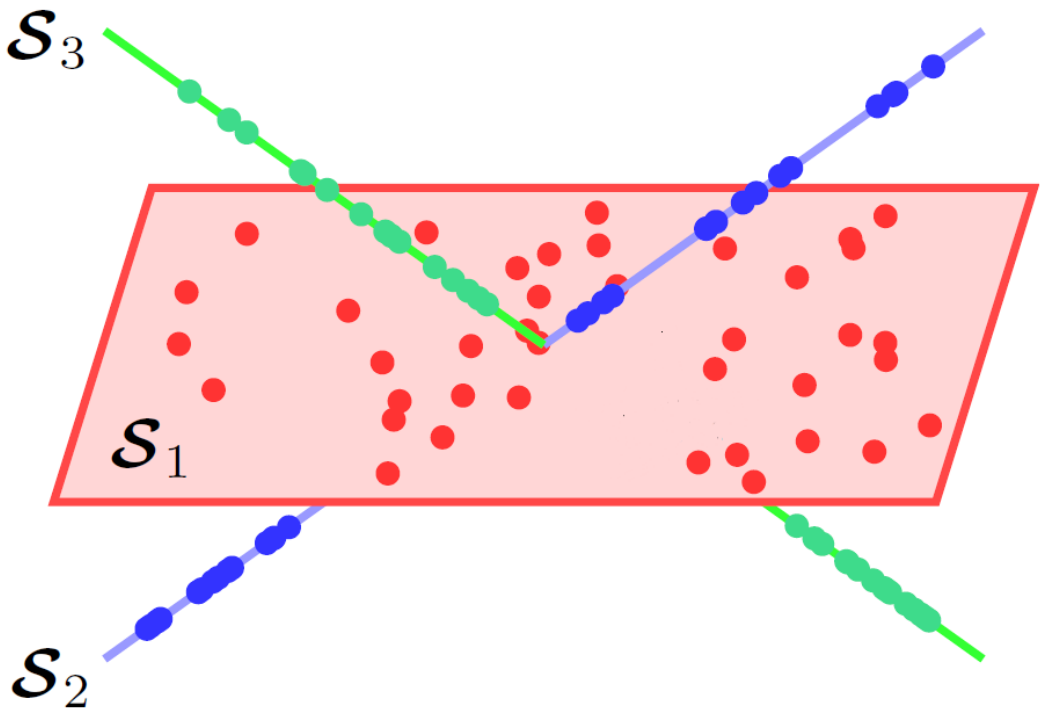
- **Data**: Given feature points on multiple rigidly moving objects tracked in multiple frames of a video.
- **Task**: Separate the feature trajectories according to the moving objects.
- **Assumption**: Point trajectories associated with each moving object across multiple frames lie in a linear subspace of dimension at most 4.

Face Clustering

- **Data**: Face images of multiple subjects.
- **Task**: Find images that belong to the same subject.
- **Assumption**: The set of all reflectance functions produced by Lambertian objects under distant, isotropic lighting lies close to a 9D linear subspace [Basri and Jacobs, 2003].

What is Subspace Clustering?

- **Problem:** Given a set of sufficient amount of data points $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$, drawn from a union of L linear or affine subspaces $\{\mathcal{S}\}_{i=1}^{L}$ of unknown dimensions, segment all data vectors into their respective subspaces.



Prior Work on Subspace Clustering

Existing algorithms can be divided into four main categories:

- Iterative Methods [Tseng, 2000]
- Algebraic Approaches [Kanatani, 2001]
- Statistical Methods [Tipping and Bishop, 1999]
- **Self-Expressiveness Method** [Elhamifar and Vidal, 2009]

Prior Work on Subspace Clustering

Existing algorithms can be divided into four main categories:

- Iterative Methods [Tseng, 2000]
- Algebraic Approaches [Kanatani, 2001]
- Statistical Methods [Tipping and Bishop, 1999]
- **Self-Expressiveness Method** [Elhamifar and Vidal, 2009]

Self-Expressiveness method can handle noise and outliers in the data well and doesn't need to know the dimension of each subspace in advance.

Summary

- The Curses of Dimensionality
  - Insufficient Data
  - Invalid Distance
  - Redundant Information

Summary

- The Curses of Dimensionality
  - Insufficient Data
  - Invalid Distance
  - Redundant Information
- The Blessings of Dimensionality
  - Low dimensionality structure is hidden from high dimensionality

Summary

- The Curses of Dimensionality
  - Insufficient Data
  - Invalid Distance
  - Redundant Information
- The Blessings of Dimensionality
  - Low dimensionality structure is hidden from high dimensionality
- Subspace Clustering Model
  - Motion Segmentation
  - Face Clustering

Outline

Self-Expressiveness Method

- **Idea:** each data point in a union of subspaces can be efficiently represented as a linear or affine combination of other points.
- **Difficulty:** This is an ill-posed problem.

Self-Expressiveness Method

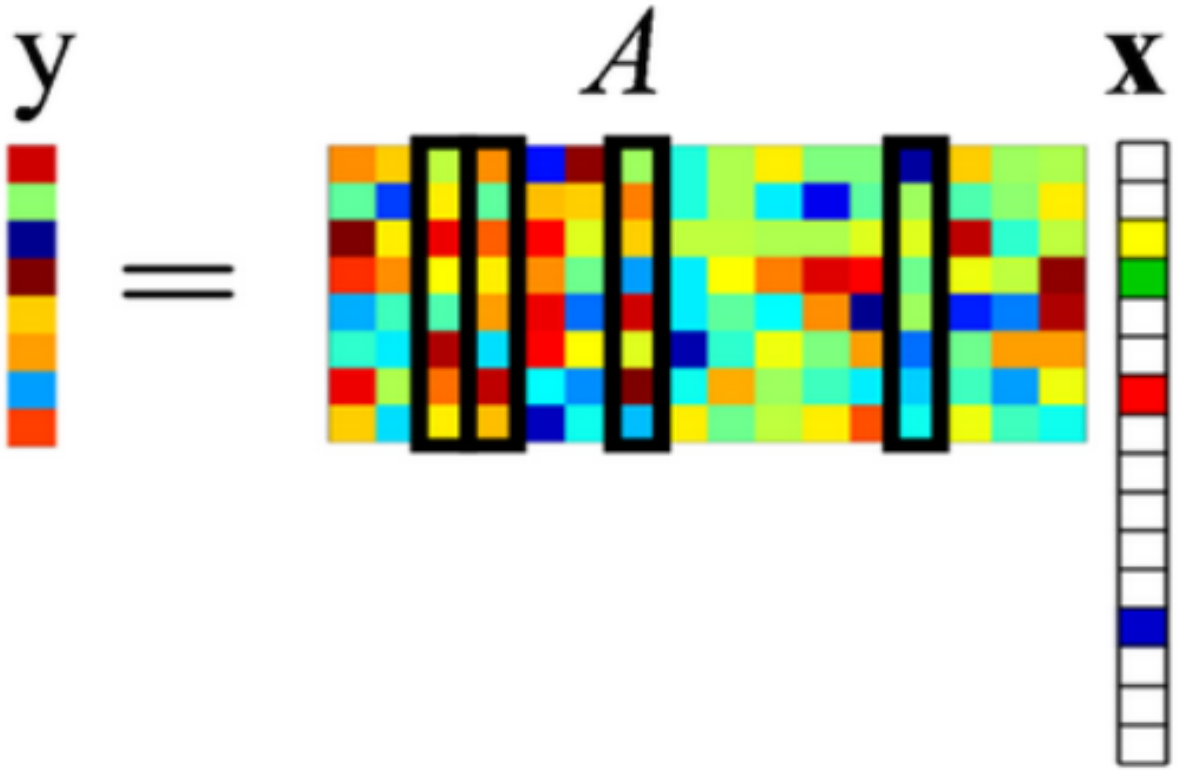
- **Idea:** each data point in a union of subspaces can be efficiently represented as a linear or affine combination of other points.
- **Difficulty:** This is an ill-posed problem.

Example 2

Given a data matrix $X = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 0 & 0 \end{bmatrix}$, then $\begin{bmatrix} 0 \\ 2 \end{bmatrix}$ may be represented as $2\begin{bmatrix} 0 \\ 1 \end{bmatrix} - 2\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix}$.

- **Inspiration:** Compressed Sensing

Inspired by Compressed Sensing



$$\min_{\boldsymbol{x}} \ \|\boldsymbol{x}\|_1$$
$$\text{s.t. } \boldsymbol{y} = A\boldsymbol{x}$$

- l_1 norm not only forces $\boldsymbol{x}$ to be sparse, but also guarantee the uniqueness of the solution.

Sparse Subspace Clustering

Problem: Given a set of sufficient amount of data points $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$, drawn from a union of n linear or affine subspaces $\{\mathcal{S}\}_{i=1}^{n}$ of unknown dimensions, segment all data vectors into their respective subspaces.

Sparse Subspace Clustering

Problem: Given a set of sufficient amount of data points $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$, drawn from a union of n linear or affine subspaces $\{\mathcal{S}\}_{i=1}^{n}$ of unknown dimensions, segment all data vectors into their respective subspaces.

Goal: aim to find a matrix $Z \in \mathbb{R}^{N \times N}$, such that:

$$\begin{aligned} &\min_Z \quad \|Z\|_1 \\ &\text{s.t. } X = XZ, \ \operatorname{diag}(Z) = 0 \end{aligned}$$

Sparse Subspace Clustering

Problem: Given a set of sufficient amount of data points $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$, drawn from a union of n linear or affine subspaces $\{\mathcal{S}\}_{i=1}^{n}$ of unknown dimensions, segment all data vectors into their respective subspaces.

Goal: aim to find a matrix $Z \in \mathbb{R}^{N \times N}$, such that:

$$\begin{aligned} &\min_Z \quad \|Z\|_1 \\ &\text{s.t. } X = XZ, \ \text{diag}(Z) = 0 \end{aligned}$$

- $X = XZ$: X is a self-expressive dictionary in which each point can be written as a linear combination of other points.
- $\min_Z \|Z\|_1$: guarantee the sparsity and uniqueness.
- $\text{diag}(Z) = 0$: avoid trivial solution I.

Sparse Subspace Clustering

Noiseless data in linear subspace:

$$\min_Z \|Z\|_1$$
$$\text{s.t. } X = XZ,\ \text{diag}(Z) = 0$$

Sparse Subspace Clustering

Noiseless data in linear subspace:

$$\begin{aligned} &\min_Z \|Z\|_1 \\ &\text{s.t. } X = XZ,\ \operatorname{diag}(Z) = 0 \end{aligned}$$

Noiseless data in affine subspace:

$$\begin{aligned} &\min_Z \|Z\|_1 \\ &\text{s.t. } X = XZ,\ Z^T\mathbf{1} = \mathbf{1},\ \operatorname{diag}(Z) = 0 \end{aligned}$$

Sparse Subspace Clustering

Noiseless data in linear subspace:

$$\begin{aligned} &\min_Z \|Z\|_1 \\ &\text{s.t. } X = XZ,\ \text{diag}(Z) = 0 \end{aligned}$$

Noiseless data in affine subspace:

$$\begin{aligned} &\min_Z \|Z\|_1 \\ &\text{s.t. } X = XZ,\ Z^T\mathbf{1} = \mathbf{1},\ \text{diag}(Z) = 0 \end{aligned}$$

Noise and sparse outlying entries:

$$\begin{aligned} \min \quad & \|Z\|_1 + \lambda_e\|E\|_1 + \tfrac{\lambda_S}{2}\|S\|_F^2 \\ \text{s.t.} \quad & X = XZ + E + S,\ \text{diag}(Z) = 0 \end{aligned}$$

Sparse Subspace Clustering

Let's first consider the most general situation:

$$\begin{array}{ll} \min & \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_S}{2}\|S\|_F^2 \\ \text{s.t.} & X = XZ + E + S,\ \text{diag}(Z) = 0 \end{array}$$

Sparse Subspace Clustering

Let's first consider the most general situation:

$$\begin{array}{ll} \min & \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_S}{2}\|S\|_F^2 \\ \text{s.t.} & X = XZ + E + S,\ \text{diag}(Z) = 0 \end{array}$$

- How to optimize?
- Get an optimal Z, what's next?
- Is there any theoretical guarantee?

Algorithm for this Sparse Optimization

Adopt an Alternating Direction Method of Multipliers (ADMM) and first consider:

$$\begin{aligned} &\min_{(Z,E,S)} \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_s}{2}\|S\|_F^2 \\ &\text{s.t. } X = XZ + E + S,\ Z^T 1 = 1,\ \text{diag}(Z) = 0 \end{aligned}$$

Eliminate S from the optimization problem:

$$\begin{aligned} &\min_{(Z,E)} \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_s}{2}\|X - XZ - E\|_F^2 \\ &\text{s.t. } Z^T 1 = 1,\ \text{diag}(Z) = 0 \end{aligned}$$

Introduce an auxiliary matrix $A \in \mathbb{R}^{N\times N}$ and consider:

$$\begin{aligned} &\min_{(Z,E,A)} \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_s}{2}\|X - XA - E\|_F^2 \\ &\text{s.t. } A^T 1 = 1,\ A = Z - \text{diag}(Z) \end{aligned}$$

Algorithm for this Sparse Optimization

Add two penalty terms:

$$\min_{(Z,E,A)} \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_S}{2}\|X - XA - E\|_F^2$$
$$+ \frac{\rho}{2}\left\|A^T\mathbf{1} - \mathbf{1}\right\|_2^2 + \frac{\rho}{2}\|A - (Z - \mathrm{diag}(Z))\|_F^2$$
$$\text{s.t. } A^T\mathbf{1} = \mathbf{1},\ A = Z - \mathrm{diag}(Z)$$

Introduce a vector $\delta \in \mathbb{R}^N$ and a matrix $\Delta \in \mathbb{R}^{N\times N}$ of Lagrange multipliers for the two equality constraints:

$$\mathcal{L}(Z, A, E, \delta, \Delta) = \|Z\|_1 + \lambda_e\|E\|_1 + \tfrac{\lambda_S}{2}\|X - XA - E\|_F^2$$
$$+ \tfrac{\rho}{2}\left\|A^T\mathbf{1} - \mathbf{1}\right\|_2^2 + \tfrac{\rho}{2}\|A - (C - \mathrm{diag}(Z))\|_F^2$$
$$+ \delta^T\left(A^T\mathbf{1} - \mathbf{1}\right) + \mathrm{tr}\left(\Delta^T(A - Z + \mathrm{diag}(Z))\right)$$

All of sub-problems can be solved efficiently!

ADMM Algorithm

Algorithm : Solving SSC via an ADMM Algorithm

Initialization: Set maxIter $= 10^4$, $k = 0$, and Terminate $\leftarrow$ False. Initialize $Z^{(0)}$, $\boldsymbol{A}^{(0)}$, $\boldsymbol{E}^{(0)}$, $\delta^{(0)}$, and $\boldsymbol{\Delta}^{(0)}$ to zero.

1: **while** (Terminate == False) **do**

2: update $\boldsymbol{A}^{(k+1)}$ by solving the following system of linear equations

$$(\lambda_s \boldsymbol{Y}^\top \boldsymbol{Y} + \rho \boldsymbol{I} + \rho \mathbf{1}\mathbf{1}^\top)\boldsymbol{A}^{(k+1)} = \lambda_s \boldsymbol{Y}^\top (\boldsymbol{Y} - \boldsymbol{E}^{(k)}) + \rho(\mathbf{1}\mathbf{1}^\top + Z^{(k)}) - \mathbf{1}\delta^{(k)\top} - \boldsymbol{\Delta}^{(k)},$$

3: update $Z^{(k+1)}$ as $Z^{(k+1)} = \boldsymbol{J} - \mathrm{diag}(\boldsymbol{J})$, where $\boldsymbol{J} \triangleq \mathcal{T}_{\frac{1}{\rho}}(\boldsymbol{A}^{(k+1)} + \boldsymbol{\Delta}^{(k)}/\rho)$,

4: update $\boldsymbol{E}^{(k+1)}$ as $\boldsymbol{E}^{(k+1)} = \mathcal{T}_{\frac{\lambda_e}{\lambda_z}}(\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{A}^{(k+1)})$,

5: update $\delta^{(k+1)}$ as $\delta^{(k+1)} = \delta^{(k)} + \rho\,(\boldsymbol{A}^{(k+1)\top}\mathbf{1} - 1)$,

6: update $\boldsymbol{\Delta}^{(k+1)}$ as $\boldsymbol{\Delta}^{(k+1)} = \boldsymbol{\Delta}^{(k)} + \rho\,(\boldsymbol{A}^{(k+1)} - Z^{(k+1)})$,

7: $k \leftarrow k + 1$,

8: **if** $(\|\boldsymbol{A}^{(k)\top}\mathbf{1} - 1\|_\infty \le \epsilon$ and $\|\boldsymbol{A}^{(k)} - Z^{(k)}\|_\infty \le \epsilon$ and $\|\boldsymbol{A}^{(k)} - \boldsymbol{A}^{(k-1)}\|_\infty \le \epsilon$ and $\|\boldsymbol{E}^{(k)} - \boldsymbol{E}^{(k-1)}\|_\infty \le \epsilon$ or $(k \ge \text{maxIter})$ **then**

9: Terminate $\leftarrow$ True

10: **end if**

11: **end while**

Output: Optimal sparse coefficient matrix $Z^* = Z^{(k)}$.

Clustering using Sparse Coefficients

$$\begin{aligned} \min \quad & \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_s}{2} \|S\|_F^2 \\ \text{s.t.} \quad & X = XZ + E + S, \ \text{diag}(Z) = 0 \end{aligned}$$

Obtain a sparse representation for each data point whose nonzero elements ideally correspond to points from the same subspace.

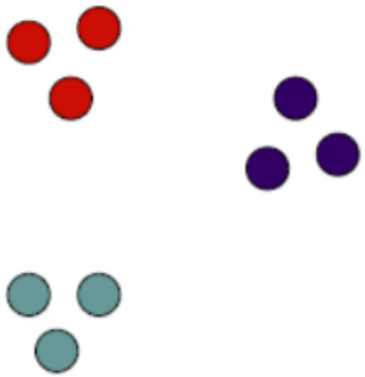
What's next?

Spectral clustering!

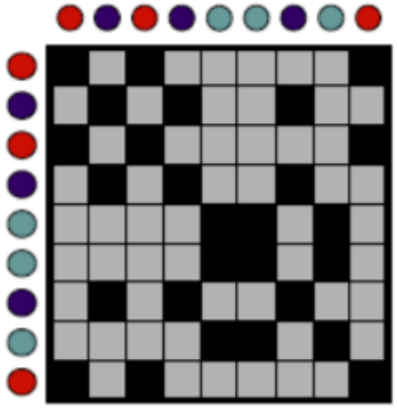
What is spectral clustering?

Key steps:

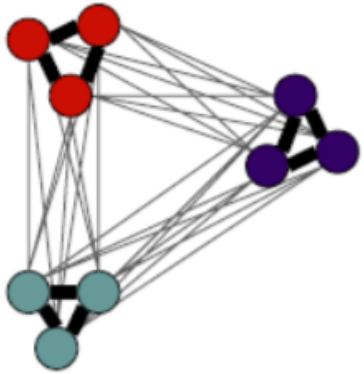
- Take a similarity graph
- Construct its graph Laplacian matrix
- Do something with its bottom eigenvectors to get clusters



Data



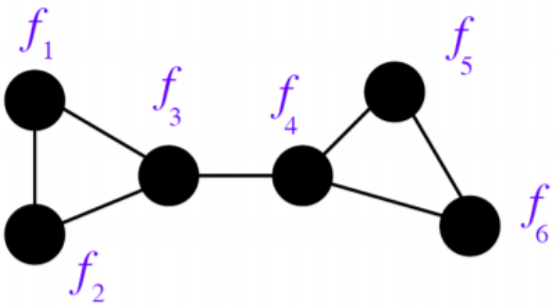
Similarities



Similarity graph

What is Spectral Clustering?

Idea: within-similarity high, between similarity low



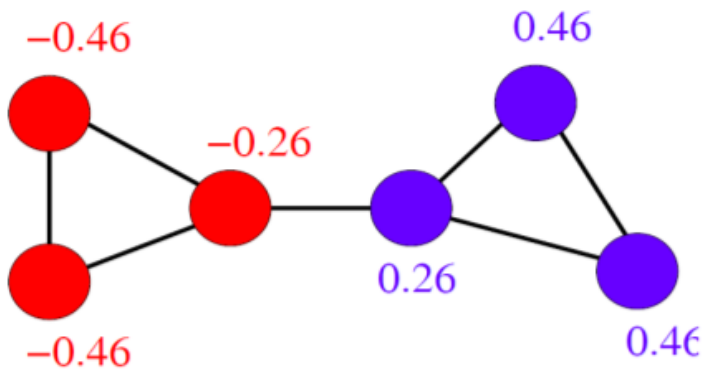
$$\mathbf{L} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix} = D - A$$

$$\operatorname*{argmin}_{S} \sum_{i \in S, j \in V - S} w_{ij} = \operatorname*{argmin}_{f_i \in \{-1,1\}} \sum_{i \sim j} w_{ij} \left(f_i - f_j\right)^2 = \frac{1}{8} \operatorname*{argmin}_{f_i \in \{-1,1\}} \boldsymbol{f}^T L \boldsymbol{f}$$

Relaxation of integrality constraint + optimality give eigenvectors:

$$L\boldsymbol{f} = \lambda \boldsymbol{f}$$

What is Spectral Clustering?



$$\mathbf{L} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

$$L\boldsymbol{f}_1 = \lambda_1 \boldsymbol{f}_1 \quad \boldsymbol{f}_1 = [-0.46, -0.46, -0.26, 0.26, 0.46, 0.46]$$

- This is a traditional spectral clustering [Von Luxburg, 2007].
- It can be easily extended more general situation.

The Framework of Sparse Subspace Clustering

1) Solve the sparse optimization problem:

$$\begin{array}{ll} \min & \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_s}{2}\|S\|_F^2 \\ \text{s.t.} & X = XZ + E + S,\ \text{diag}(Z) = 0 \end{array}$$

2) Normalize the columns of Z as $\boldsymbol{z}_i \leftarrow \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_\infty}$

3) Form a similarity graph with N nodes representing the data points. Set the weights on the edges between the nodes by $W = |Z| + |Z|^T$.

4) Apply spectral clustering to the similarity graph.

Subspace Sparse Recovery Theory

Recovery conditions for two classes of subspace arrangements:

Definition 3

A collection of subspaces $\{\mathcal{S}_i\}_{i=1}^n$ is said to be independent if $\dim\left(\oplus_{i=1}^n \mathcal{S}_i\right) = \sum_{i=1}^n \dim\left(\mathcal{S}_i\right)$, where $\oplus$ denotes the direct sum operator.

Definition 4

A collection of subspaces $\{\mathcal{S}_i\}_{i=1}^n$ is said to be disjoint if every pair of subspaces intersect only at the origin. In other words, for every pair of subspaces we have $\dim(\mathcal{S}_i \oplus \mathcal{S}_j) = \dim(\mathcal{S}_i) + \dim(\mathcal{S}_j)$.

Subspace Sparse Recovery Theory

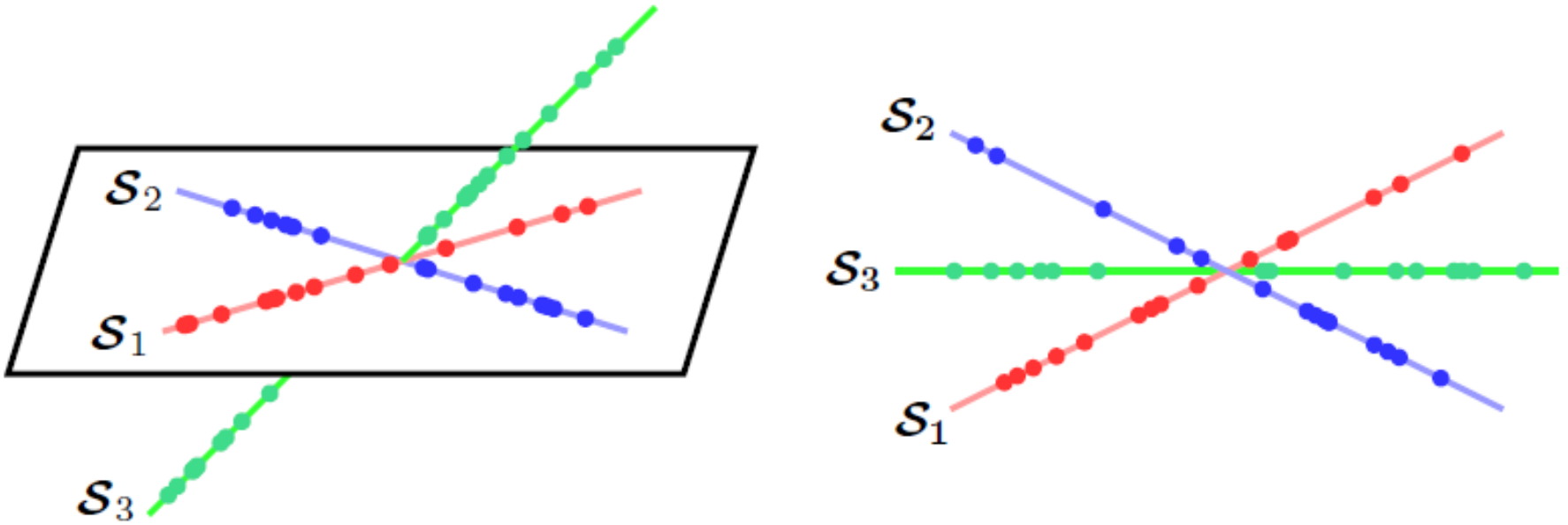


Figure: Left: the three 1-dimensional subspaces are independent as they span the 3-dimensional space and the sum of their dimensions is also 3. Right: the three 1-dimensional are disjoint as any two subspaces intersect at the origin.

Subspace Sparse Recovery Theory

Key: an important notion that can be used to characterize two disjoint subspaces is the smallest principal angle:

Definition 5

The smallest principal angle between two subspaces $\mathcal{S}_i$ and $\mathcal{S}_j$, denoted by θ_{ij} , is defined as

$$\cos\left(\theta_{ij}\right) \triangleq \max_{\boldsymbol{v}_i \in \mathcal{S}_i, \boldsymbol{v}_j \in \mathcal{S}_j} \frac{\boldsymbol{v}_i^T \boldsymbol{v}_j}{\left\|\boldsymbol{v}_i\right\|_2 \left\|\boldsymbol{v}_j\right\|_2}$$

Independent Subspace Model

Theorem 6

Consider a collection of data points drawn from n independent subspaces $\{\mathcal{S}_i\}_{i=1}^n$ *of dimensions* $\{d_i\}_{i=1}^n$. *Let* X_i *denote* N_i *data points in* $\mathcal{S}_i$, *where* $\text{rank}(X_i) = d_i$, *and let* X_{-i} *denote data points in all subspaces except* $\mathcal{S}_i$. *Then, for every* $\mathcal{S}_i$ *and every nonzero* $\boldsymbol{x}$ *in* $\mathcal{S}_i$, *the* l_q*-minimization program:*

$$\begin{bmatrix} \boldsymbol{z}^* \\ \boldsymbol{z}_-^* \end{bmatrix} = \text{argmin} \left\| \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{z}_- \end{bmatrix} \right\|_q \quad \text{s.t.} \quad \boldsymbol{x} = \begin{bmatrix} X_i & X_{-i} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{z}_- \end{bmatrix}$$

for $q \geqslant 1$, *recovers a subspace-sparse representation, i.e.,* $\boldsymbol{z}^* \neq \boldsymbol{0}$ *and* $\boldsymbol{z}_-^* = \boldsymbol{0}$.

Disjoint Subspace Model

Theorem 7

Consider a collection of data points drawn from n disjoint subspaces $\{\mathcal{S}_i\}_{i=1}^n$ *of dimensions* $\{d_i\}_{i=1}^n$. *Let* X_i *denote* N_i *data points in* $\mathcal{S}_i$, *where* $\text{rank}(X_i) = d_i$, *and let* X_{-i} *denote data points in all subspaces except* $\mathcal{S}_i$. *Let* $\mathcal{W}_i$ *be the set of all full-rank submatrices* $X_i \in \mathbb{R}^{D \times d_i}$ *of* X_i, *where* $\text{rank}(X_i) = d_i$. *If the condition:*

$$\max_{\tilde{X}_i \in \mathcal{W}_i} \sigma_{d_i}\left(\tilde{X}_i\right) > \sqrt{d_i}\, \|X_{-i}\|_{1,2} \max_{j \neq i} \cos\left(\theta_{ij}\right)$$

holds, then for every nonzero $\boldsymbol{x}$ *in* $\mathcal{S}_i$, *the* l_1*-minimization recovers a subspace-sparse solution, i.e.,* $\boldsymbol{z}^* \neq \boldsymbol{0}$ *and* $\boldsymbol{z}^*_- = \boldsymbol{0}$.

Experiments with Synthetic Data

- 3 disjoint subspaces $\{\mathcal{S}_i\}_{i=1}^{3}$
- Principal angels θ
- Number of data points N_g
- Subspace sparse recovery error (ssr error):

$$\frac{1}{3N_g}\sum_{i=1}^{3N_g}\left(1-\frac{\|\boldsymbol{z}_{ik_i}\|_1}{\|\boldsymbol{z}_i\|_1}\right)$$

- Subspace clustering error:

$$\frac{\#\text{ of misclassified points}}{\text{total }\#\text{ of points}}$$

Experiments with Synthetic Data

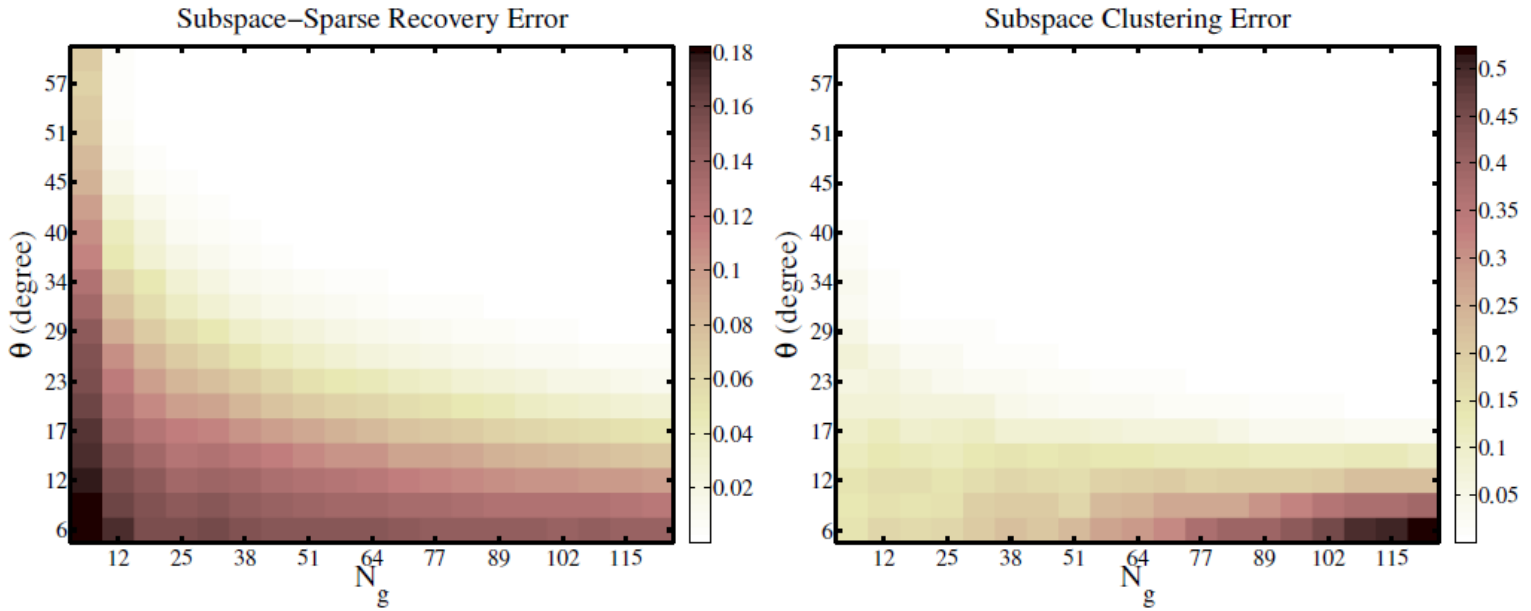


Figure: Subspace-sparse recovery error (left) and subspace clustering error (right) for three disjoint subspaces

- In general, when θ or N_g increases, the errors decrease.
- Note that for small θ as we increase N_g, the subspace sparse recovery error is large and slightly decreases, while the clustering error increases.

Summary

The framework of sparse subspace clustering:

1) Formulate into the following problem:

$$\begin{aligned} \min \quad & \|Z\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_s}{2}\|S\|_F^2 \\ \text{s.t.} \quad & X = XZ + E + S,\ \text{diag}(Z) = 0 \end{aligned}$$

2) Normalize the columns of Z

3) Form a similarity graph

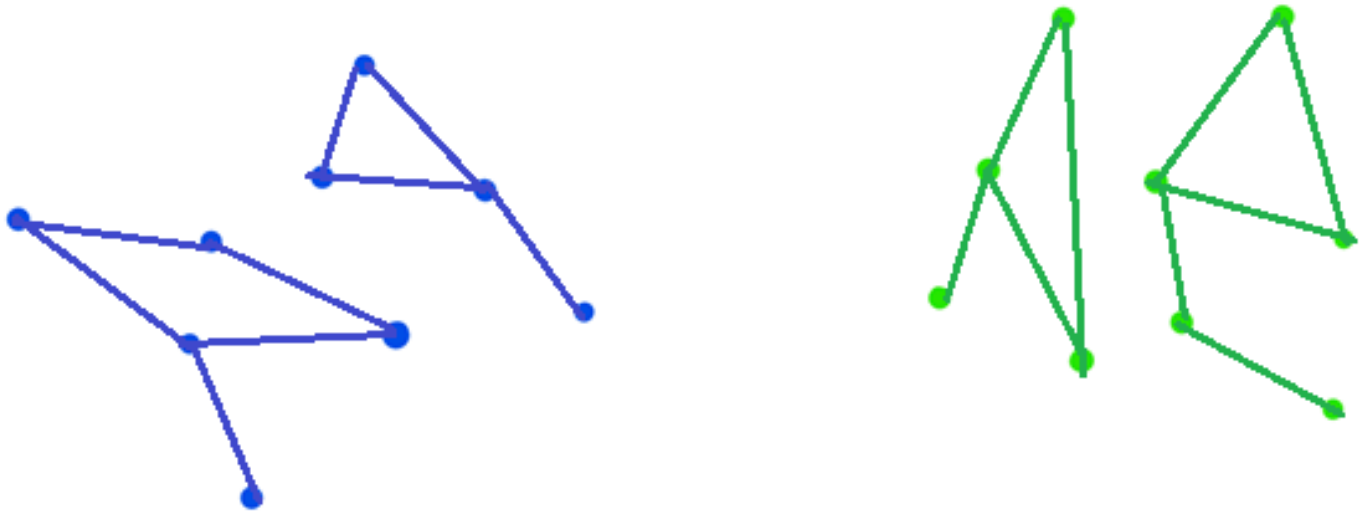
4) Apply spectral clustering

- Self-expressive
- Handel noise and outliers
- Good theoretical guarantee

Outline

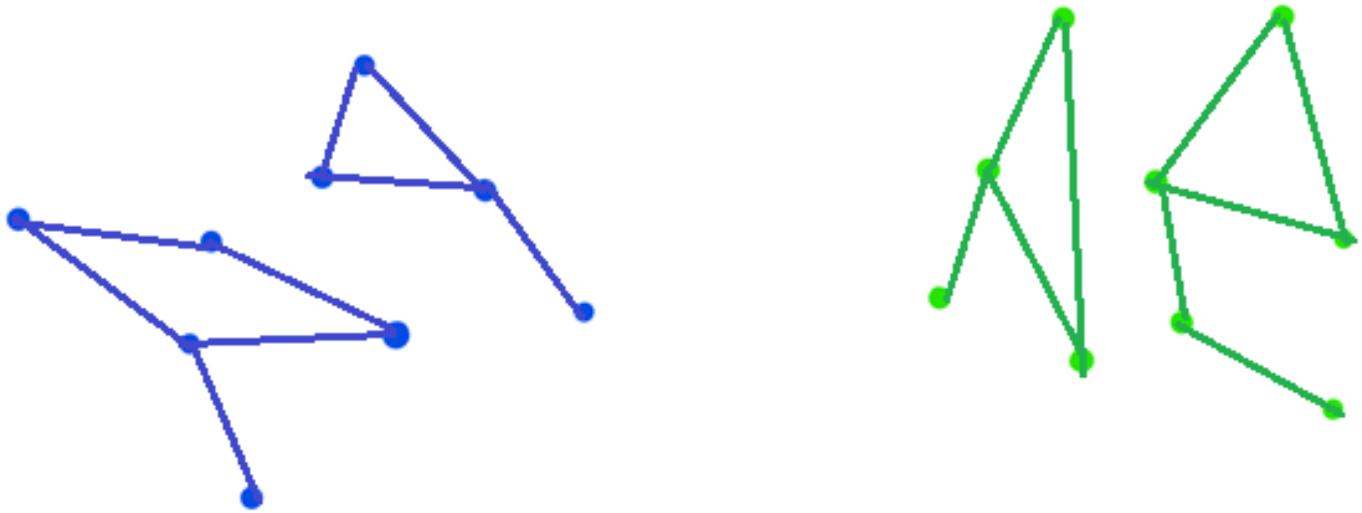
Weakness of Sparse Subspace Clustering

Weakness: maybe too sparse in the same class



Weakness of Sparse Subspace Clustering

Weakness: maybe too sparse in the same class



Solution:

- Robust Subspace Clustering [Soltanolkotabi et al., 2014]
- Low Rank Representation [Liu et al., 2010]
- Least Squares Subspace Clustering [Lu et al., 2012]

Robust Subspace Clustering

Definition 8

Fix i and $j \in \{1, \cdots, n\}$ and let Z be the similarity matrix. Then we say that (i, j) obeying $Z_{ij} \neq 0$ is **a false discovery** if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ do not originate from the same subspace.

Definition 9

In the same situation, (i, j) obeying $Z_{ij} \neq 0$ is **a true discovery** if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ originate from the same subspace.

Main Idea: We wish to make few false discoveries (and not link too many pairs belonging to different subspaces). At the same time, we wish to make many true discoveries, whence a natural trade off.

Robust Subspace Clustering

LASSO with data-driven regularization:
For each $\boldsymbol{x}_i$, consider a natural sparse regression strategy LASSO:

$$\min_{\boldsymbol{z}} \frac{1}{2} \|\boldsymbol{x}_i - X\boldsymbol{z}_i\|_2^2 + \lambda \|\boldsymbol{z}_i\|_1$$

λ controls sparsity in $\boldsymbol{z}_i$.

We need to choose λ wisely:

- Take λ as large as possible (as to prevent false discoveries)
- Make sure that the number of true discoveries is also on the order of the dimension d, typically in the range $[0.5d, 0.8d]$, where d is the ambient dimension of $\boldsymbol{x}_i$.

Robust Subspace Clustering

How to Choose λ:

Let $\hat{\boldsymbol{z}}$ be the minimizer of the LASSO function:

$$K(\boldsymbol{z}, \lambda) = \frac{1}{2}\|\boldsymbol{x} - X\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

Let $\hat{\boldsymbol{z}}_{eq}$ be the solution of the problem when $\lambda = 0$.

Then we obtain the following inequation:

$$\frac{1}{2}\|\boldsymbol{x} - X\hat{\boldsymbol{z}}\|_2^2 \leqslant K(\hat{\boldsymbol{z}}, \lambda) \leqslant K\left(\hat{\boldsymbol{z}}_{eq}, \lambda\right) = \lambda\left\|\hat{\boldsymbol{z}}_{eq}\right\|_1$$

How to Choose λ

- λ has to scale at least like $1/\sqrt{d}$
  - To make sure $\hat{\boldsymbol{z}}$ has a number of nonzero components in the range $[0.5d, 0.8d]$, then $\|\boldsymbol{x} - X\hat{\boldsymbol{z}}\|_2^2$ has to greater than a fixed constant.
  - $\|\hat{\boldsymbol{z}}_{eq}\|_1$ is on the order of $\sqrt{d}$

How to Choose λ

- λ has to scale at least like $1/\sqrt{d}$
  - To make sure $\hat{\boldsymbol{z}}$ has a number of nonzero components in the range $[0.5d, 0.8d]$, then $\|\boldsymbol{x} - X\hat{\boldsymbol{z}}\|_2^2$ has to greater than a fixed constant.
  - $\|\hat{\boldsymbol{z}}_{eq}\|_1$ is on the order of $\sqrt{d}$

- λ has to scale at most like $\sqrt{(\log N)/d}$
  - $\hat{\boldsymbol{z}} = \boldsymbol{0}$ if $\lambda \geqslant \|X^T\boldsymbol{x}\|_\infty$
  - $\|X^T\boldsymbol{x}\|_\infty$ scales at most like $\sqrt{(\log N)/d}$

Experiments with Synthetic Data

- A single subspace in $\mathbb{R}^n$ with $n = 2000$.
- 5, 4, 3, 4, 4, and 2 subspaces of respective dimensions 200, 150, 100, 50, 20 and 10.
- Noise level $\sigma = 0.3$.

- For each data point, we choose different values of λ around the heuristic $\lambda_0 = 1/\sqrt{d}$, namely, $\lambda \in [0.1\lambda_0, 2.5\lambda_0]$.

Experiments with Synthetic Data

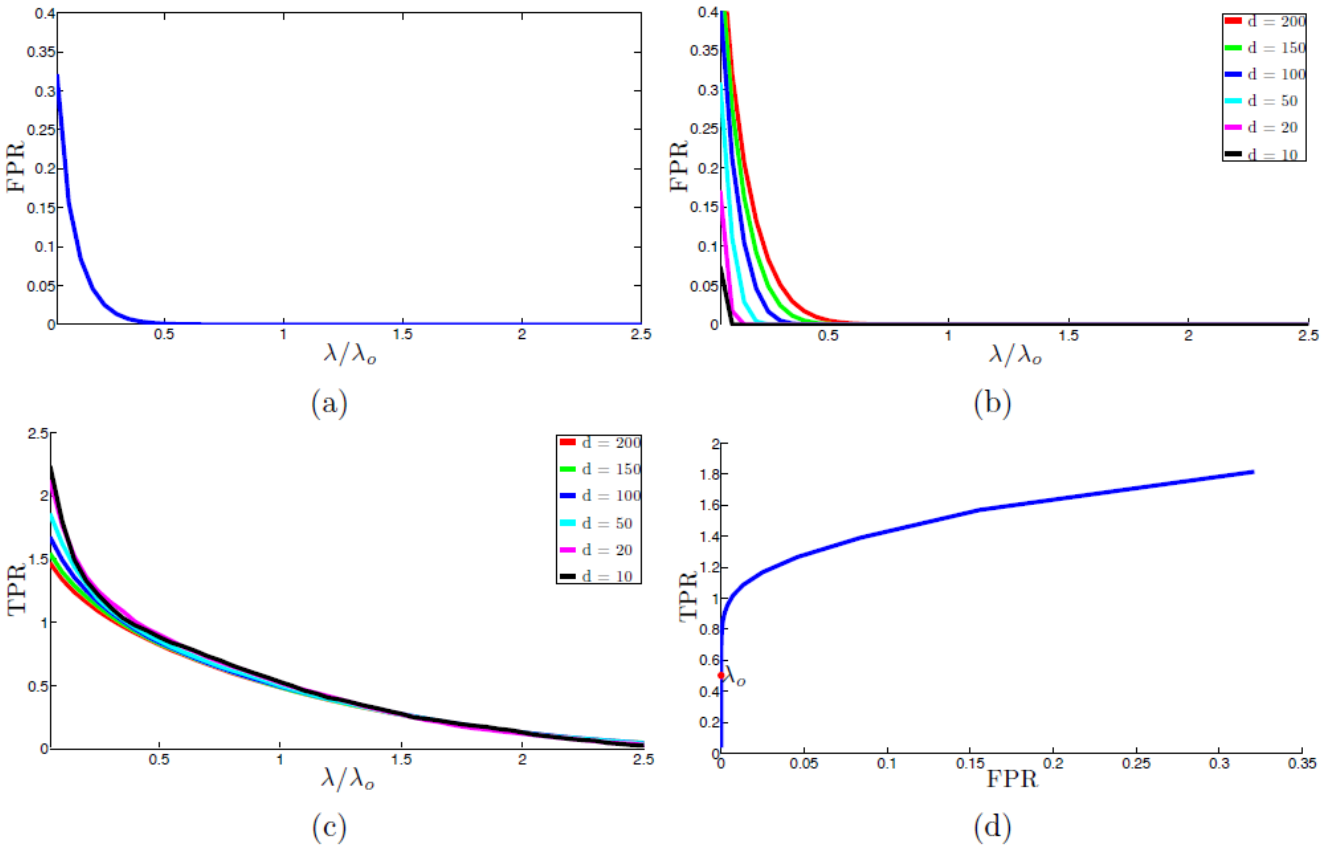


Figure 2: Performance of LASSO for values of λ in an interval including the heuristic $\lambda_o = 1/\sqrt{d}$. (a) Average number of false discoveries normalized by $(n - d)$ (FPR) on all m sampled data points). (b) FPR for different subspace dimensions. Each curve represents the average FPR over those samples originating from subspaces of the same dimension. (c) Average number of true discoveries per dimension for various dimensions (TPR). (d) TPR vs. FPR (ROC curve). The point corresponding to $\lambda = \lambda_o$ is marked as a red dot.

How to Estimate d?

Review:

$$\frac{1}{2}\|\boldsymbol{x} - X\hat{\boldsymbol{z}}\|_2^2 \leqslant K(\hat{\boldsymbol{z}}, \lambda) \leqslant K\left(\hat{\boldsymbol{z}}_{eq}, \lambda\right) = \lambda \left\|\hat{\boldsymbol{z}}_{eq}\right\|_1$$

- $\left\|\hat{\boldsymbol{z}}_{eq}\right\|_1$ is on the order of $\sqrt{d}$.
- Why not use a multiple of $\|\boldsymbol{z}\|_1$ as a proxy for $\sqrt{d}$?

How to Estimate d?

A two-step procedure with data-driven regularization:

- Solve:

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\| \quad \text{s.t. } \|\boldsymbol{x} - X\boldsymbol{z}\|_2^2 < \tau$$

- Set $\lambda = \alpha / \|\boldsymbol{z}^*\|_1$
- Solve

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{x} - X\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

Experiments with Synthetic Data

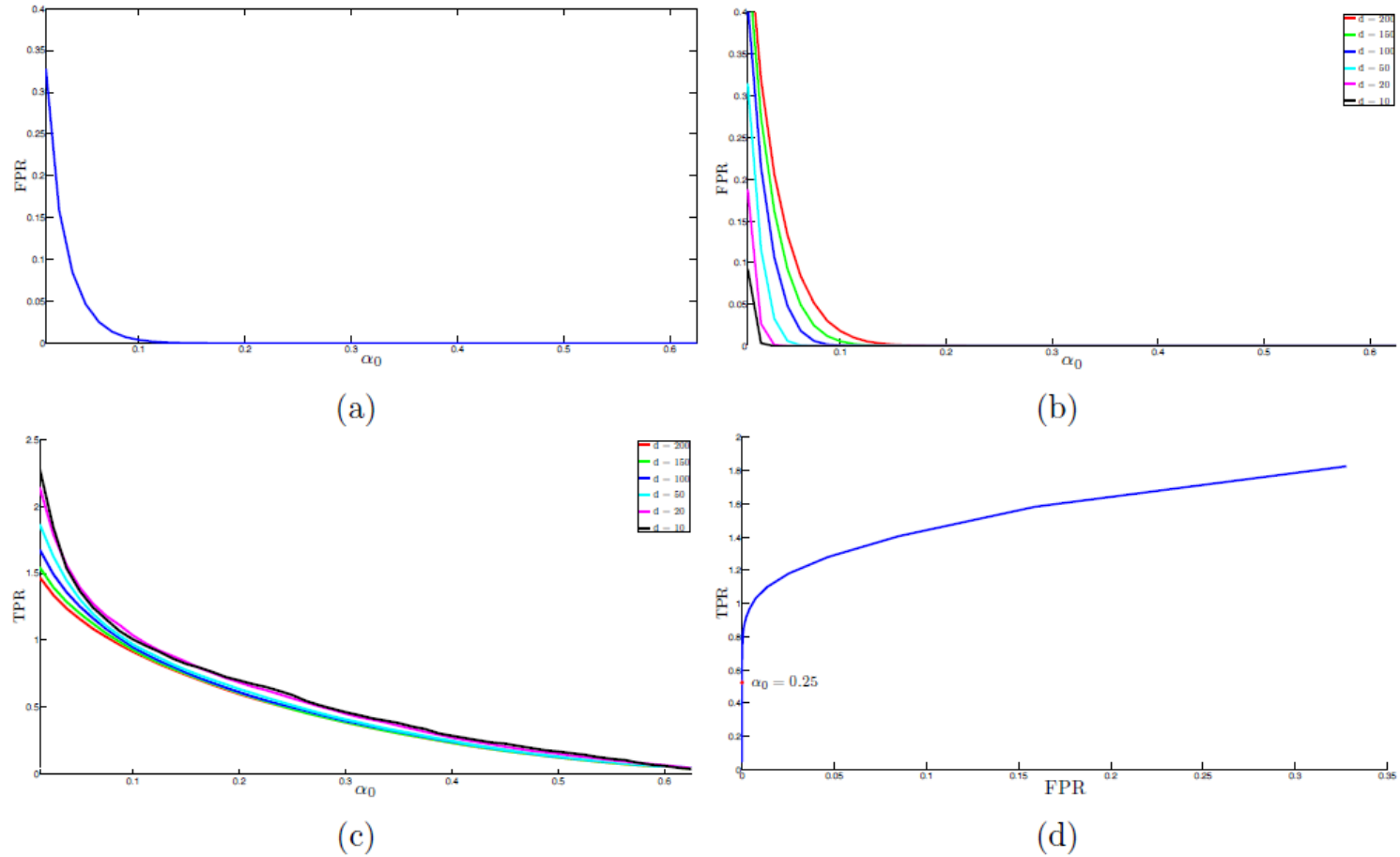


Figure: Performance of the two-step procedure using $\alpha = 0.25$. (a) False positive rate (FPR). (b) FPR for various subspace dimensions. (c) True positive rate (TPR). (d) TPR vs. FPR.

Theoretical Guarantee

Theorem 10 (No false discoveries)

Assume that the subspace attached to the i-th column obeys the affinity and sampling conditions and that the noise level σ is bounded. Then in Robust Subspace Clustering Algorithm, take $\tau = 2\sigma$ and $\alpha > 0.707\sigma$. Then with high probability, there is no false discovery in the i-th column.

Theoretical Guarantee

Theorem 11 (Many true discoveries)

Consider the same setup as in above Theorem with α also obeying $\alpha < \alpha_0$ for some numerical constant α_0. Then with high probability, there are at least

$$c_0 \frac{d(i)}{\log\left(N(i)/d(i)\right)}$$

true discoveries in the i-th column (c_0 is a positive numerical constant).

Low-rank Representation

Main Idea: finding the lowest-rank representation of a collection of vectors jointly to capture the global structure.

Low-rank Representation

Main Idea: finding the lowest-rank representation of a collection of vectors jointly to capture the global structure.

Consider:

$$\begin{aligned} &\min_Z \ \text{rank}(Z) \\ &\text{s.t. } X = XZ \end{aligned}$$

- NP-hard
- Ill-posed

Low-rank Representation

Consider the nuclear norm:

$$\begin{aligned} &\min_Z \ \|Z\|_* \\ &\text{s.t. } X = XZ \end{aligned}$$

- $Z^* = X^+ X$ is the unique minimizer to problem.
- Z^* is also a minimizer to the problem with the constrain of rank.

Low-rank Representation

Consider the nuclear norm:

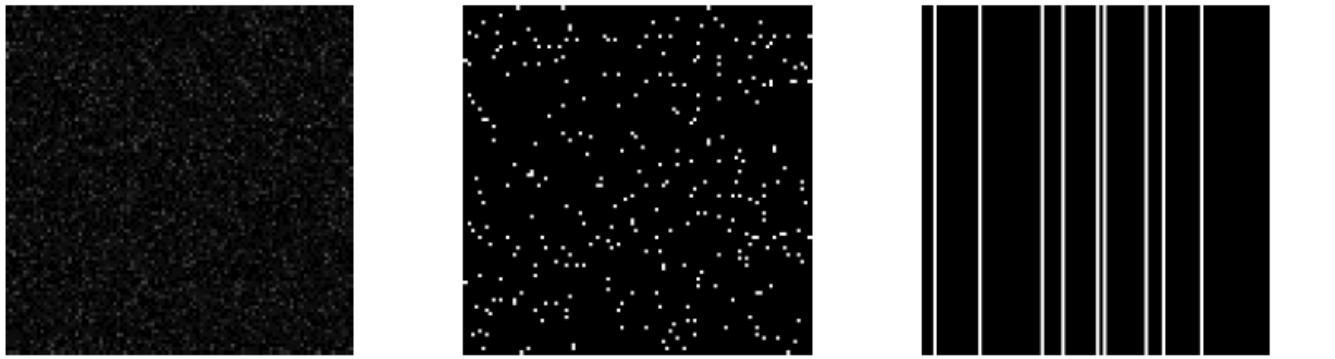
$$\begin{aligned} &\min_Z \ \|Z\|_* \\ &\text{s.t. } X = XZ \end{aligned}$$

- $Z^* = X^+ X$ is the unique minimizer to problem.
- Z^* is also a minimizer to the problem with the constrain of rank.

In other words, the power of nuclear norm is far more than convex relaxation!

Low-rank Representation

In this topic, we want to handle sample-specific corruptions.



(a) noise (b) random corruptions (c) sample-specific corruptions

Consider:

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1}$$

$$\text{s.t. } X = XZ + E$$

where $\|E\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{n} (|E|_{ij})^2}$ is called $l_{2,1}$-norm.

Solving the Optimization Problem

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1}$$
$$\text{s.t. } X = XZ + E$$

First convert it to the following problem:

$$\min_{Z,E,J} \|J\|_* + \lambda \|E\|_{2,1}$$
$$\text{s.t. } X = XZ + E,\ Z = J$$

Solving the Optimization Problem

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1}$$
$$\text{s.t. } X = XZ + E$$

First convert it to the following problem:

$$\min_{Z,E,J} \|J\|_* + \lambda \|E\|_{2,1}$$
$$\text{s.t. } X = XZ + E,\ Z = J$$

Solving the Augmented Lagrange Multiplier (ALM) problem:

$$\min_{Z,E,J,Y_1,Y_2} \|J\|_* + \lambda \|E\|_{2,1} + \text{tr}\left[Y_1^T(X - XZ - E)\right] + \text{tr}\left[Y_2^T(Z - J)\right] + \frac{\mu}{2}\left(\|X - XZ - E\|_F^2 + \|Z - J\|_F^2\right)$$

Solving the Optimization Problem

Algorithm 1 Solving Problem by Inexact ALM

Input: data matrix X, parameter λ
Initialize: $Z = J = 0, E = 0, Y_1 = 0, Y_2 = 0, \mu = 10^{-6}, max_u = 10^{10}, \rho = 1.1, \varepsilon = 10^{-8}$.
while not converged **do**

1. fix the others and update J by

$$J = \arg\min \frac{1}{\mu}||J||_* + \frac{1}{2}||J - (Z + Y_2/\mu)||_F^2$$

2. fix the others and update Z by

$$Z = (I + X^tX)^{-1}(X^tX - X^tE + J + (X^tY_1 - Y_2)/\mu)$$

3. fix the others and update E by

$$E = \arg\min \frac{\lambda}{\mu}||E||_{2,1} + \frac{1}{2}||E - (X - XZ + Y_1/\mu)||_F^2$$

4. update the multipliers

$$\begin{aligned} Y_1 &= Y_1 + \mu(X - XZ - E) \\ Y_2 &= Y_2 + \mu(Z - J) \end{aligned}$$

5. update the parameter μ by $\mu = \min(\rho\mu, max_u)$
6. check the convergence conditions

$$||X - XZ - E||_\infty < \varepsilon \text{ and } ||Z - J||_\infty < \varepsilon.$$

end while

The Framework of Low-rank Representation

1) Solve the low-rank optimization program:

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1}$$
$$\text{s.t. } X = XZ + E$$

2) Normalize the columns of Z as $\boldsymbol{z}_i \leftarrow \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_\infty}$
3) Form a similarity graph with N nodes representing the data points. Set the weights on the edges between the nodes by $W = |Z| + |Z|^T$.
4) Apply spectral clustering to the similarity graph.

Least Squares Subspace Clustering

Review a theorem in sparse subspace clustering:

Theorem 12

Consider a collection of data points drawn from n independent subspaces $\{\mathcal{S}_i\}_{i=1}^n$ *of dimensions* $\{d_i\}_{i=1}^n$. *Let* X_i *denote* N_i *data points in* $\mathcal{S}_i$, *where* $\text{rank}(X_i) = d_i$, *and let* Y_{-i} *denote data points in all subspaces except* $\mathcal{S}_i$. *Then, for every* $\mathcal{S}_i$ *and every nonzero* $\boldsymbol{x}$ *in* $\mathcal{S}_i$, *the* l_q*-minimization program:*

$$\begin{bmatrix} \boldsymbol{z}^* \\ \boldsymbol{z}_-^* \end{bmatrix} = \text{argmin} \left\| \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{z}_- \end{bmatrix} \right\|_q \quad \text{s. t.} \quad \boldsymbol{x} = \begin{bmatrix} X_i & X_{-i} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{z}_- \end{bmatrix}$$

for $q \geqslant 1$, *recovers a subspace-sparse representation, i.e.,* $\boldsymbol{z}^* \neq \boldsymbol{0}$ *and* $\boldsymbol{z}_-^* = \boldsymbol{0}$.

Least Squares Subspace Clustering

- When the subspaces are independent, for $q \geqslant 1$, the l_q minimization can recover the representation.
- l_2-norm can prevent over sparisity in the same class.
- l_2-norm is more tractable than l_1 norm.

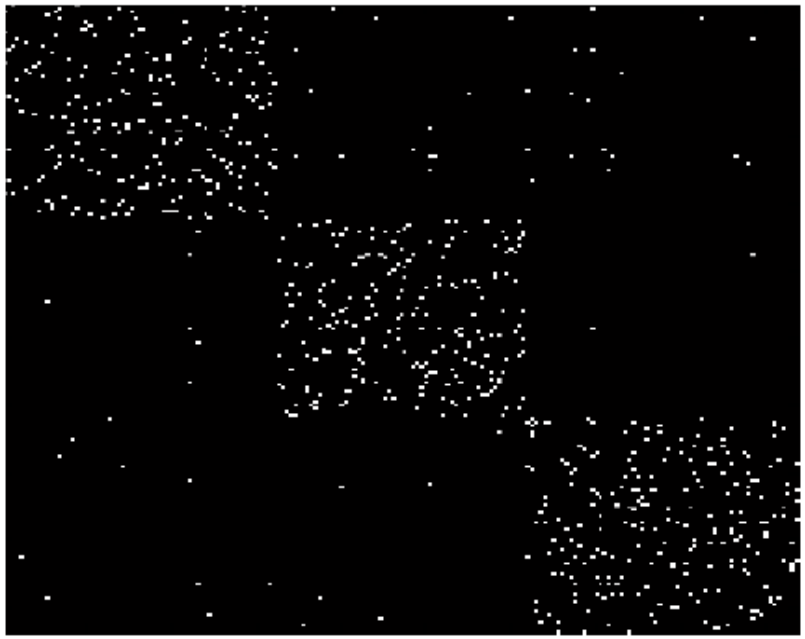
Consider:

$$\min_{Z} \ \|X - XZ\|_F^2 + \lambda \|Z\|_F^2 \quad \text{s.t.} \quad \text{diag}(Z) = 0$$

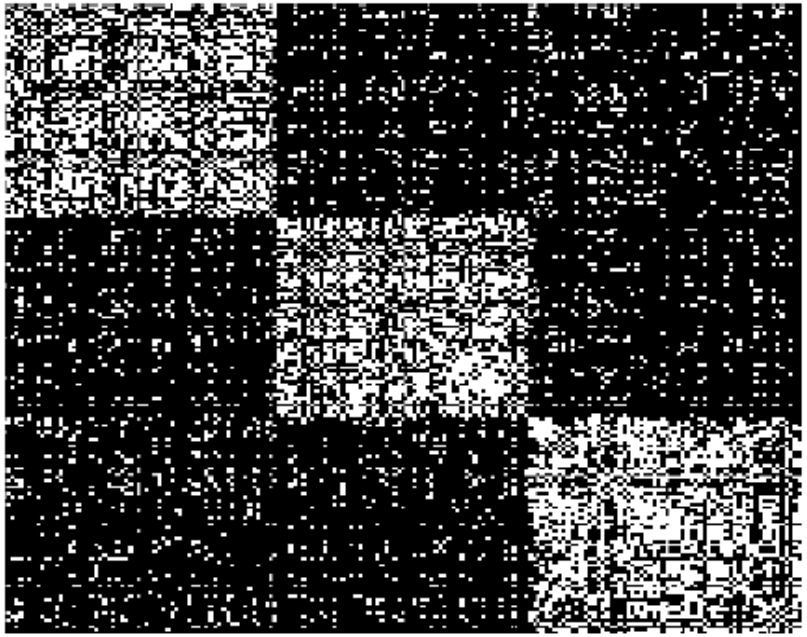
- The above optimization problem can be solved closely!

Least Squares Subspace Clustering

SSC



LSR



✓ Few wrong connections

✗ Not well connected

✗ Many wrong connections

✓ Well-connected

Summary

- **Robust Subspace Clustering**
  Choose λ wisely to guarantee the prevent false discoveries and make sure that the number of true discoveries at the same time.

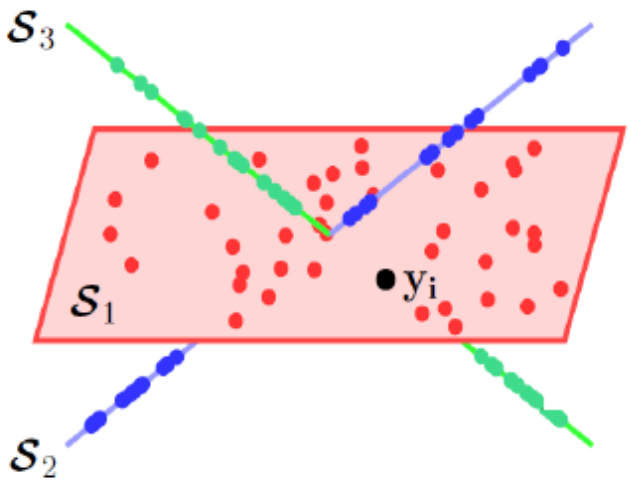
- **Low-rank Representation**
  Lowest-rank representation to capture the global structure.

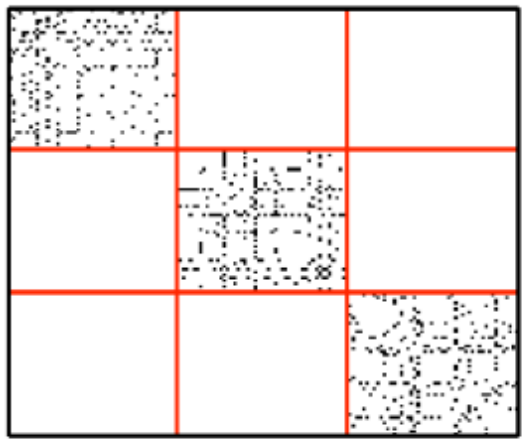
- **Least Squares Subspace Clustering**
  Use l_2-norm to replace l_1-norm and prevent over sparsity in the same class.

Outline

Block Diagonality Properties

If X is arranged, then a good representation matrix Z should be block diagonal.





Unifying Previous Methods

Previous methods can be written as:

$$\min f(Z) \quad \text{s.t. } Z \in \Omega$$

	$f(Z)$	Ω
SSC	$\|Z\|_0$ or $\|Z\|_1$	$\{Z \mid X = XZ,\ \mathrm{diag}(Z) = 0\}$
LRR	$\|Z\|_*$	$\{Z \mid X = XZ\}$
MSR	$\|Z\|_1 + \lambda\|Z\|_*$	$\{Z \mid X = XZ, \mathrm{diag}(Z) = 0\}$
SSQP	$\|Z^\top Z\|_1$	$\{Z \mid X = XZ, Z \geq 0, \mathrm{diag}(Z) = 0\}$
LSR	$\|Z\|_F$	$\{Z \mid X = XZ\}$

- All the above methods give **block diagonal** solution for **independent** subspaces.
- However, they are proved case by case.
- More details about MSR and SSQP can be seen in [Luo et al., 2011] and [Wang et al., 2011] respectively.

Unifying Previous Methods

What f Gives Block Diagonality?

- Consider the following general SC formulation:

$$\min f(Z) \text{ s.t. } Z \in \Omega$$

- When subspaces are independent, What kind of objective functions induce the **block diagonal** solution?
- Enforced Block Diagonal Condition [Lu et al., 2018] will be introduced in the following parts.

Enforced Block Diagonal Condition

Enforced Block Diagonal (EBD) Conditions. Assume the matrix function f is defined on $\Omega(\neq \varnothing)$. For any $Z = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \Omega$, $Z \neq 0$, where A and D are square matrices, B and C are of compatible dimension, A, $D \in \Omega$. Let $Z^D = \begin{bmatrix} A & 0 \\ 0 & D \end{bmatrix} \in \Omega$. The EBD conditions are:

(1) $f(Z) = f(P^\top Z P)$, $\forall$ permutation matrix $P \in \{P | P^\top Z P \in \Omega\}$.

(2) $f(Z) \geq f(Z^D)$, where the equality holds iff $B = C = 0$.

(3) $f(Z^D) = f(A) + f(D)$.

Enforced Block Diagonal Condition

f satisfies EBD cond.
+
subspaces are independent
→ block diagonal Z

Theorem Assume data sampling is sufficient, and subspaces are independent. Consider:

$$\min f(Z) \ \text{ s.t. } Z \in \Omega = \{Z | X = XZ\}.$$

If f satisfies the EBD conditions (1)(2), then Z^* is block diagonal

$$Z^* = \text{blkdiag}(Z_1^*, Z_2^*, \ldots, Z_k^*)$$

with $Z_i^* \in \mathbb{R}^{n_i \times n_i}$ corresponding to X_i, for each i. Furthermore, if f satisfies the EBD conditions (1)(2)(3), for each i, Z_i^* is optimal solution to:

$$\min f(Y) \ \text{ s.t. } X_i = X_i Y$$

Enforced Block Diagonal Condition

- The above theorem gives a general guarantee for almost all subspace clustering algorithms when subspaces are independent.
- Usually, the solution is far from being k-block diagonal since the independent subspaces assumption does not hold due to data noise.
- How to Pursue Block-diagonality directly?
  - How to pursuer exactly block-diagonal structure?
  - Efficient optimization?
  - Performance guarantee?

Block Diagonal Prior

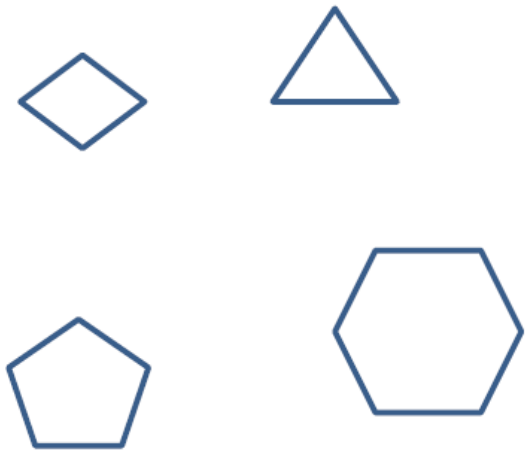
Aim: Give a formal definition of the k-block diagonal matrix.
Consider the following matrix:

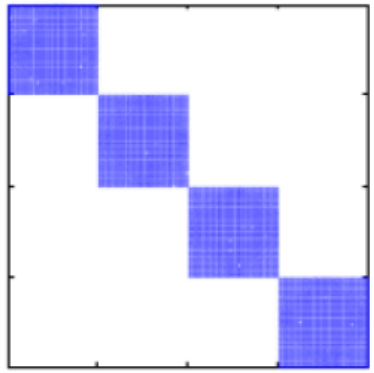
$$B = \begin{bmatrix} B_0 & 0 & 0 \\ 0 & B_0 & 0 \\ 0 & 0 & B_0 \end{bmatrix}, \text{ where } B_0 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

- There are 3 blocks intuitively.
- But we can also say that it has 1 or 2 blocks.
- How to define k-block diagonal property?

Block Diagonal Prior

Recall a spectral graph theory:





Theorem Let W be an affinity matrix. Then the multiplicity k of the eigenvalue 0 of the corresponding Laplacian L_W equals the number of connected components in W.

Here, $L_W(j,j') = -W(j,j')$, if $j \neq j'$; $\sum_{\ell \neq j} W(j,\ell)$ otherwise.

$$W \text{ is } k \text{ block diagonal } \Leftrightarrow W \in \left\{ W \,\middle|\, \text{rank}(L_W) = n-k, W \in \mathbb{R}^{n \times n} \right\}$$

k-Block Diagonal Regularizer

Suppose data lying in k subspaces:

- **Aim:** design a regularizer to enforce B to be k-block diagonal
- L_B has k zero eigenvalues
- Consider the sum of the smallest k eigenvalues of L_Z

Definition 13

For any affinity matrix $B \in \mathbb{R}^{n \times n}$, the k-block diagonal regularizer is defined as the sum of the k smallest eigenvalues of L_B, i.e.,

$$\|B\|_{\mathcal{K}} = \sum_{i=n-k+1}^{n} \lambda_i \left(L_B\right)$$

Block Diagonal Representation

k-block diagonal regularizer:

$$\|B\|_{\mathcal{K}} = \sum_{i=n-k+1}^{n} \lambda_i \left(L_B\right)$$

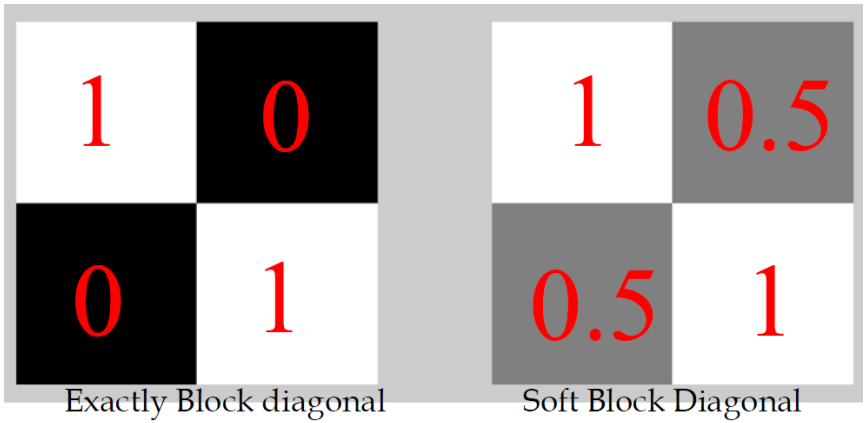
With the proposed k-block diagonal regularizer, the Block Diagonal Representation (BDR) method for subspace clustering is as follows [Lu et al., 2018]:

$$\min_B \frac{1}{2}\|X - XB\|_F^2 + \gamma\|B\|_{\mathcal{K}}$$
$$\text{s.t. } \operatorname{diag}(B) = 0, B \geqslant 0, B = B^T$$

Block Diagonal Representation

$$\min_B \frac{1}{2}\|X - XB\|_F^2 + \gamma\|B\|_{\mathcal{K}}$$
$$\text{s.t. } \text{diag}(B) = 0, B \geqslant 0, B = B^T$$

- The restrictions on B will limit its representation capability.
- An exactly block diagonal matrix may not be necessary.



- Both the above two affinity matrices lead to the same clustering results.

Soft Block Diagonal Representation

$$\min_B \frac{1}{2}\|X - XB\|_F^2 + \gamma\|B\|_{\mathcal{K}}$$
$$\text{s.t. } \text{diag}(B) = 0, B \geqslant 0, B = B^T$$

Consider the soft version of block diagonal representation:

$$\min_{B,Z} \frac{1}{2}\|X - XZ\|^2 + \frac{\lambda}{2}\|Z - B\|^2 + \gamma\|B\|_{\mathcal{K}}$$
$$\text{s.t. } \text{diag}(B) = 0, B \geqslant 0, B = B^T$$

Soft Block Diagonal Representation

$$\min_{B} \frac{1}{2}\|X - XB\|_F^2 + \gamma\|B\|_{\mathcal{K}}$$
$$\text{s.t. } \operatorname{diag}(B) = 0, B \geqslant 0, B = B^T$$

Consider the soft version of block diagonal representation:

$$\min_{B,Z} \frac{1}{2}\|X - XZ\|^2 + \frac{\lambda}{2}\|Z - B\|^2 + \gamma\|B\|_{\mathcal{K}}$$
$$\text{s.t. } \operatorname{diag}(B) = 0, B \geqslant 0, B = B^T$$

Next problem: How to optimize?
The key challenge lies in the nonconvex term $\|B\|_{\mathcal{K}}$.

Optimization of BDR

Theorem 14

Let $L \in \mathbb{R}^{n\times n}$ *and* $L \succeq 0$. *Then*

$$\sum_{i=n-k+1}^{n} \lambda_i(L) = \min_{W} \langle L, W \rangle, s.t. 0 \preceq W \preceq I, \mathrm{tr}(W) = k$$

Optimization of BDR

Theorem 14

Let $L \in \mathbb{R}^{n\times n}$ *and* $L \succeq 0$. *Then*

$$\sum_{i=n-k+1}^{n} \lambda_i(L) = \min_W \langle L, W\rangle, s.t. 0 \preceq W \preceq I, \mathrm{tr}(W) = k$$

Then the origin problem is equivalent to:

$$\begin{aligned} &\min_{Z,B,W} \ \tfrac{1}{2}\|X - XZ\|_F^2 + \tfrac{\lambda}{2}\|Z - B\|_F^2 + \gamma\langle \mathrm{diag}(B\mathbf{1}) - B, W\rangle \\ &\text{s.t. } \mathrm{diag}(B) = 0, B \geqslant 0, B = B^T, 0 \preceq W \preceq I, \mathrm{tr}(W) = k \end{aligned}$$

Optimization of BDR

$$\min_{Z,B,W} \ \frac{1}{2}\|X - XZ\|_F^2 + \frac{\lambda}{2}\|Z - B\|_F^2 + \gamma\langle \mathrm{diag}(B\mathbf{1}) - B, W\rangle$$
$$\text{s.t. } \mathrm{diag}(B) = 0, B \geqslant 0, B = B^T, 0 \preceq W \preceq I, \mathrm{tr}(W) = k$$

Optimization of BDR

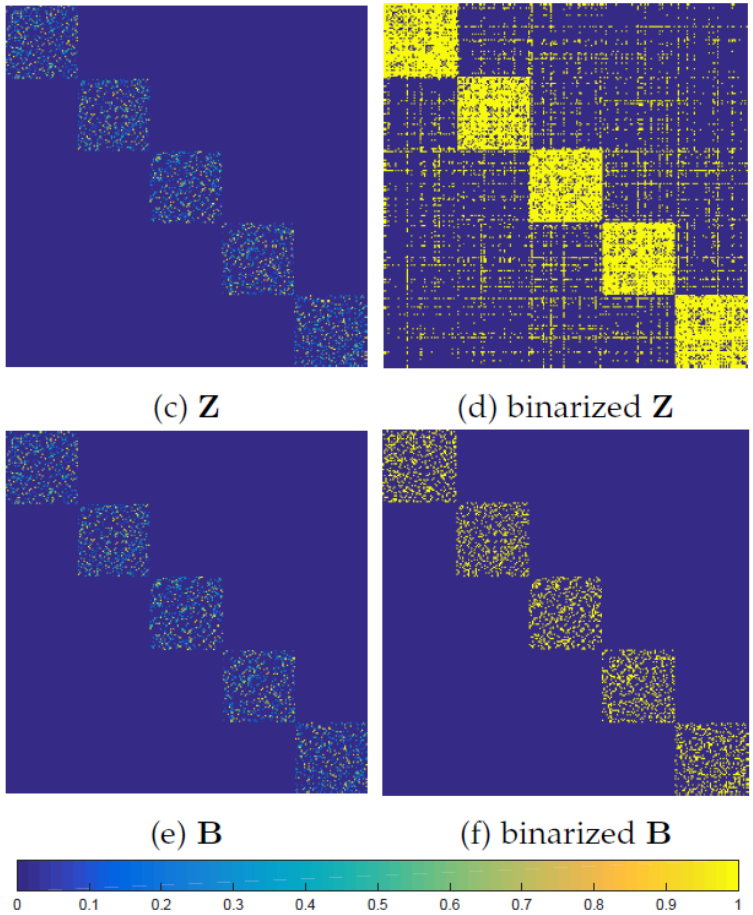
$$\min_{Z,B,W} \frac{1}{2}\|X - XZ\|_F^2 + \frac{\lambda}{2}\|Z - B\|_F^2 + \gamma\langle \mathrm{diag}(B\mathbf{1}) - B, W\rangle$$
$$\text{s.t. } \mathrm{diag}(B) = 0, B \geqslant 0, B = B^T, 0 \preceq W \preceq I, \mathrm{tr}(W) = k$$

Note that W is independent from Z, thus one can group them as a super block $\{W, Z\}$ and treat $\{B\}$ as the other block.

Thus, the problem can be solved by alternating updating $\{W, Z\}$ and $\{B\}$.
All of sub problems can be solved efficiently!

Experiments with Synthetic Data

- 5 disjoint subspaces in $\mathbb{R}^{30}$
- Each subspace of dimension 5
- Pick 50 data points in each subspace



Convergence Guarantee

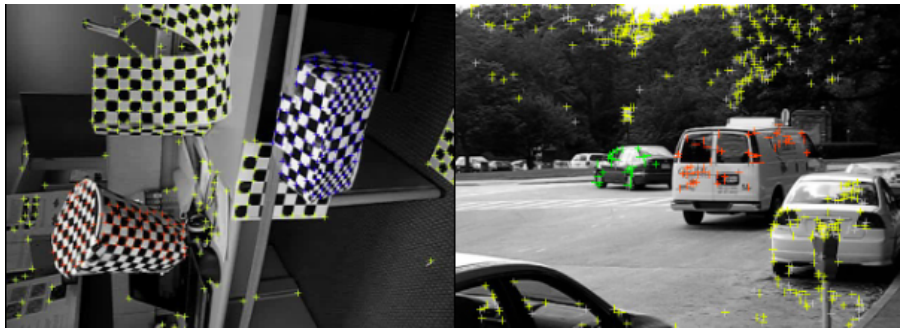
$$\min_{Z,B,W} \ \frac{1}{2}\|X - XZ\|_F^2 + \frac{\lambda}{2}\|Z - B\|_F^2 + \gamma\langle \mathrm{diag}(B\mathbf{1}) - B, W\rangle$$
$$\text{s.t. } \mathrm{diag}(B) = 0, B \geqslant 0, B = B^T, 0 \preceq W \preceq I, \mathrm{tr}(W) = k$$

Theorem 15

The sequence $\left\{W^k, Z^k, B^k\right\}$ *generated by above algorithm has at least one limit point and any limit point* (Z^*, B^*, W^*) *of* $\left\{Z^k, B^k, W^k\right\}$ *is a stationary point.*

Experiment on Motion Segmentation

Hopkins 155 database:



method	SSC	LRR	LSR	BDR-B	BDR-Z
2-motions					
mean	1.52	3.65	3.24	1.00	**0.95**
3-motions					
mean	4.40	9.40	5.94	1.95	**0.85**
all					
mean	2.18	4.95	3.85	1.22	**0.93**

Table: The mean clustering errors (%) of 155 sequences on Hopkins 155 dataset.

Experiment on Face Clustering

Extended Yale B database:

	2 subjects			3 subjects			5 subjects			8 subjects			10 subjects		
method	mean	median	std	mean	median	std	mean	median	std	mean	median	std	mean	median	std
SCC	24.02	19.92	17.82	42.19	41.93	8.93	61.36	62.34	6.10	71.87	72.27	4.72	72.48	73.28	6.14
SSC	1.64	0.78	2.91	3.26	0.52	7.69	6.30	4.22	5.43	8.94	9.67	6.18	10.09	11.33	4.59
LRR	5.39	0.39	14.50	6.04	1.04	12.34	8.13	2.34	9.61	6.79	3.42	6.50	9.49	12.58	5.38
LSR	3.16	0.78	10.18	3.96	1.56	8.72	7.85	6.72	8.72	28.14	31.05	12.32	33.27	33.12	4.57
S^3C	**1.29**	0.00	2.69	2.79	0.52	7.38	4.66	1.88	5.15	6.37	6.35	5.32	6.87	6.17	3.67
BDR-B	3.28	0.78	10.15	3.02	1.30	7.78	4.45	2.19	6.29	**3.08**	**2.93**	1.18	**2.95**	**2.81**	1.09
BDR-Z	2.97	0.00	10.23	**1.15**	1.04	0.95	**3.00**	2.66	2.25	4.46	4.20	2.39	4.04	3.52	1.52

Figure: Clustering error (%) of different algorithms on the Extended Yale B database

Experiment on Face Clustering

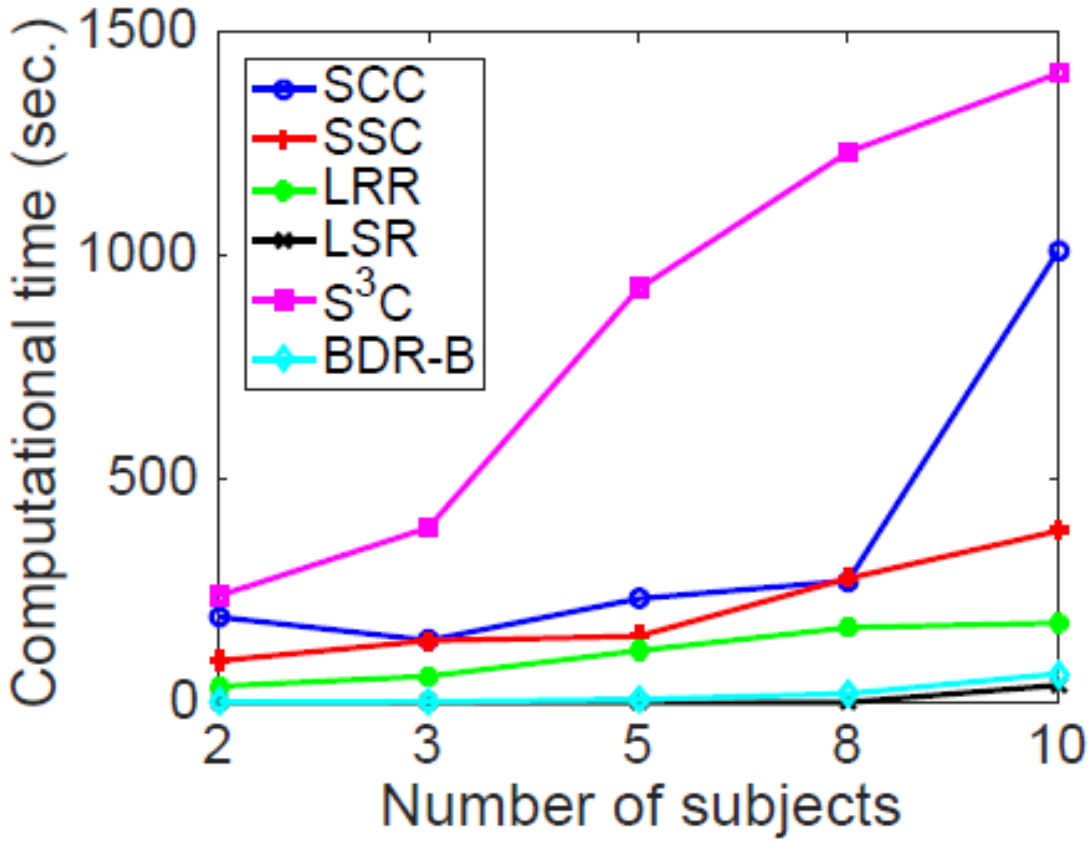


Figure: Average computational time (sec.) of the algorithms on the Extended Yale B database as a function of the number of subjects.

Summary

- Enforced Block Diagonal Condition provided a uniform theoretical guarantee for almost all subspace clustering algorithms when subspaces are independent.
- K-block diagonal regularizer is a powerful tool which encourages a nonnegative symmetric matrix to be k-block diagonal.
- Block Diagonal Representation (BDR) utilizes the block diagonal regularizer method for subspace clustering.

Outline

More than Linear Model

- While the traditional subspace clustering methods achieve state-of-the-art results on several benchmarks, in practice many datasets are better modeled by non-linear manifolds.

- Several methods extend traditional methods to non-linear manifolds.
  - Kernel Sparse Subspace Clustering [Patel and Vidal, 2014]
  - Deep Subspace Clustering Networks [Ji et al., 2017]

Kernel Sparse Subspace Clustering

Main Idea: Using kernel trick

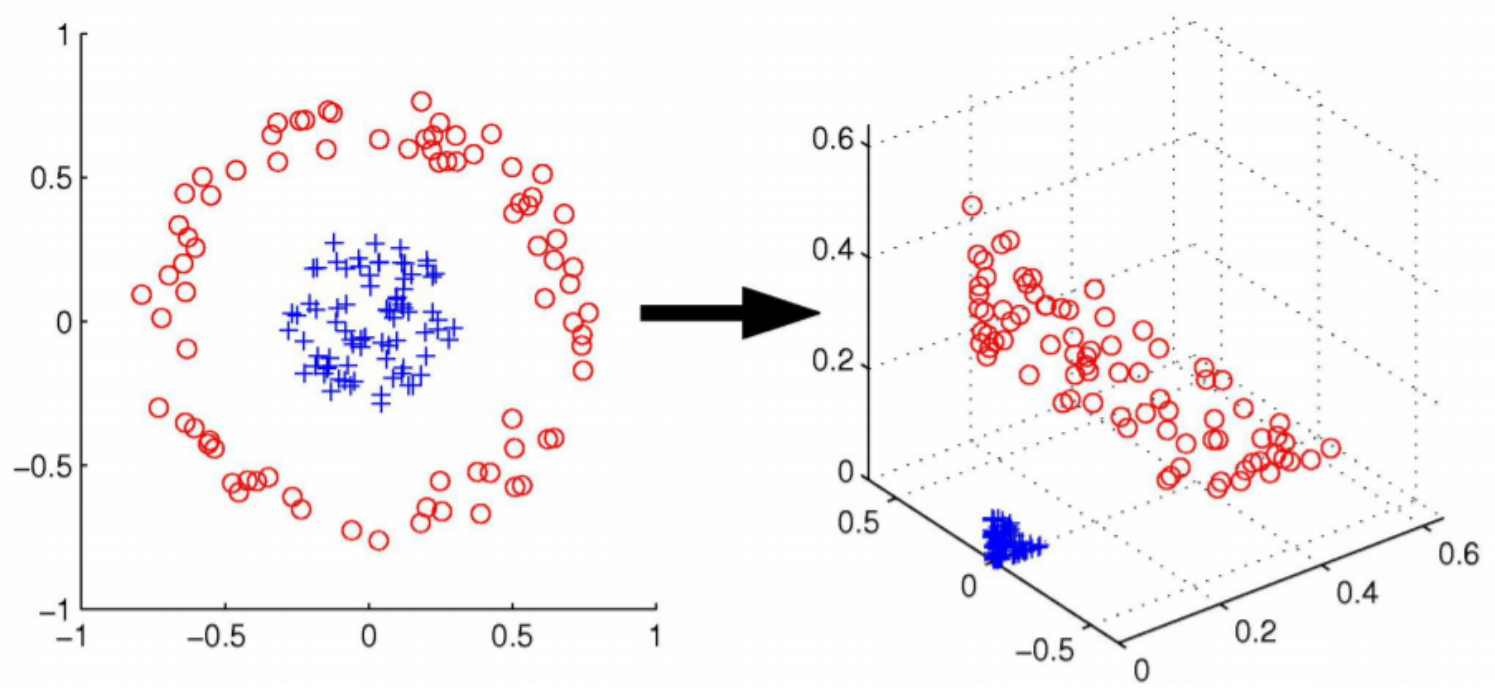


Figure: Non-linear subspaces (or sub-manifolds) are mapped to linear ones in high-dimensional feature space.

Kernel Sparse Subspace Clustering

Let $\phi : \mathbb{R}^D \to \mathcal{H}$ be a mapping from the input space to the reproducing kernel Hilbert space $\mathcal{H}$.

Consider the following optimization problem:

$$\min_{C,Z} \|Z\|_1 + \lambda \|\phi(X) - \phi(X)C\|_F^2$$
$$\text{s.t. } C = Z - \text{diag}(Z),\ C^T\mathbf{1} = \mathbf{1}$$

This problem can be efficiently solved using the ADMM method.

Experiment on Alphadigits Dataset

Algorithms (2 digits)	LRSC	LSA	SSC	LRR	KSSC P(3,2)	KSSC P(2,0.2)	KSSC G(10)	KSSC G(7)
Mean	15.81	10.70	5.70	7.76	5.42	5.40	6.13	5.93
Median	8.97	3.85	2.56	3.84	2.56	2.56	2.56	2.56

Algorithms (3 digits)	LRSC	LSA	SSC	LRR	KSSC P(3,2)	KSSC P(2,0.2)	KSSC G(10)	KSSC G(7)
Mean	25.65	22.69	13.58	14.21	12.85	13.30	14.54	13.64
Median	25.64	22.22	8.54	11.11	7.69	8.54	9.40	8.55

Algorithms (5 digits)	LRSC	LSA	SSC	LRR	KSSC P(3,2)	KSSC P(2,0.2)	KSSC G(10)	KSSC G(7)
Mean	37.77	33.81	23.26	23.34	22.64	23.00	24.50	23.84
Median	37.95	33.85	25.12	23.59	23.08	23.85	27.18	26.15

Deep Subspace Clustering Networks

- Despite kernel trick, the selection of different kernel types is largely empirical, and there is no clear reason to believe that the implicit feature space corresponding to a predefined kernel is truly well-suited to subspace clustering.

- **Main Idea:** Use a novel deep neural network architecture to learn an explicit non-linear mapping of the data that is well-adapted to subspace clustering.

Deep Subspace Clustering Networks

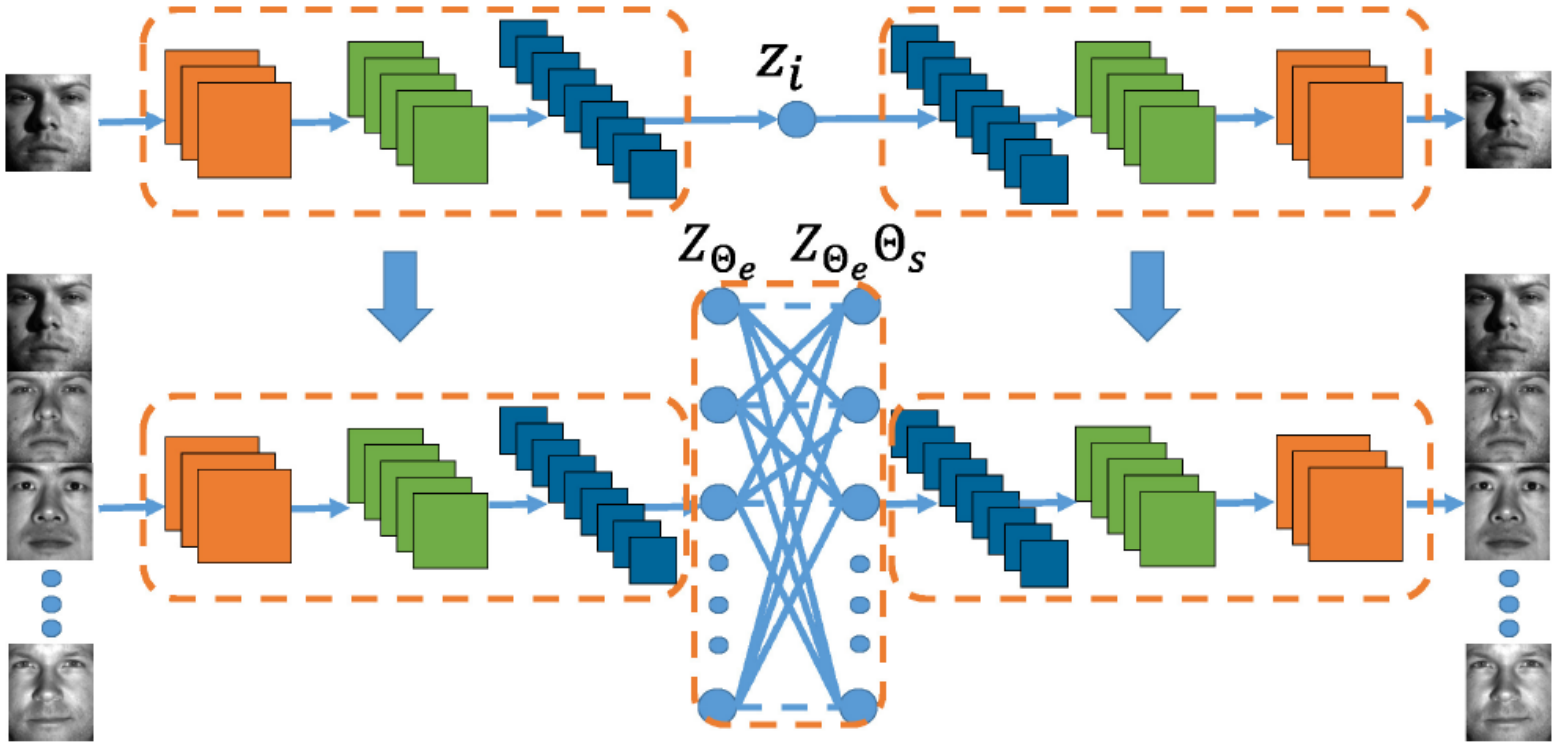


Figure: Deep Subspace Clustering Networks: As an example, we show a deep subspace clustering network with three convolutional encoder layers, one self-expressive layer, and three deconvolutional decoder layer. During training, we first pre-train the deep auto-encoder without the self-expressive layer; we then fine-tune our entire network using this pre-trained model for initialization.

Deep Subspace Clustering Networks

More details:

- Structure of the networks:
  - Convolution encoder
  - Self-expressive layer
  - Deconvolution decoder

Deep Subspace Clustering Networks

More details:

- Structure of the networks:
  - Convolution encoder
  - Self-expressive layer
  - Deconvolution decoder

- Loss Function:

$$L(\Theta, C) = \frac{1}{2}\left\|X - \hat{X}_\Theta\right\|_F^2 + \lambda_1 \|C\|_p + \frac{\lambda_2}{2}\|Z_{\Theta_e} - Z_{\Theta_e}C\|_F^2$$
$$\text{s.t.} \quad \text{diag}(C) = 0$$

Deep Subspace Clustering Networks

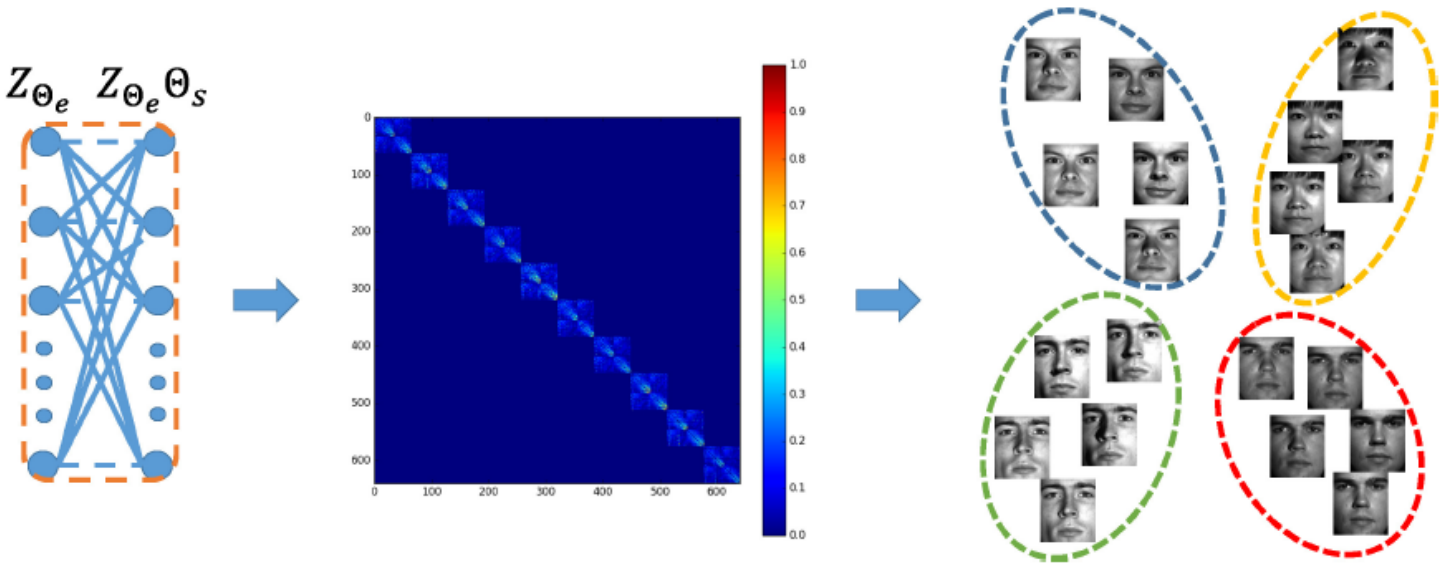


Figure: From the parameters of the self-expressive layer, we construct an affinity matrix, which we use to perform spectral clustering to get the final clusters. Best viewed in color.

Experiments on COIL20 and COIL100 Datasets

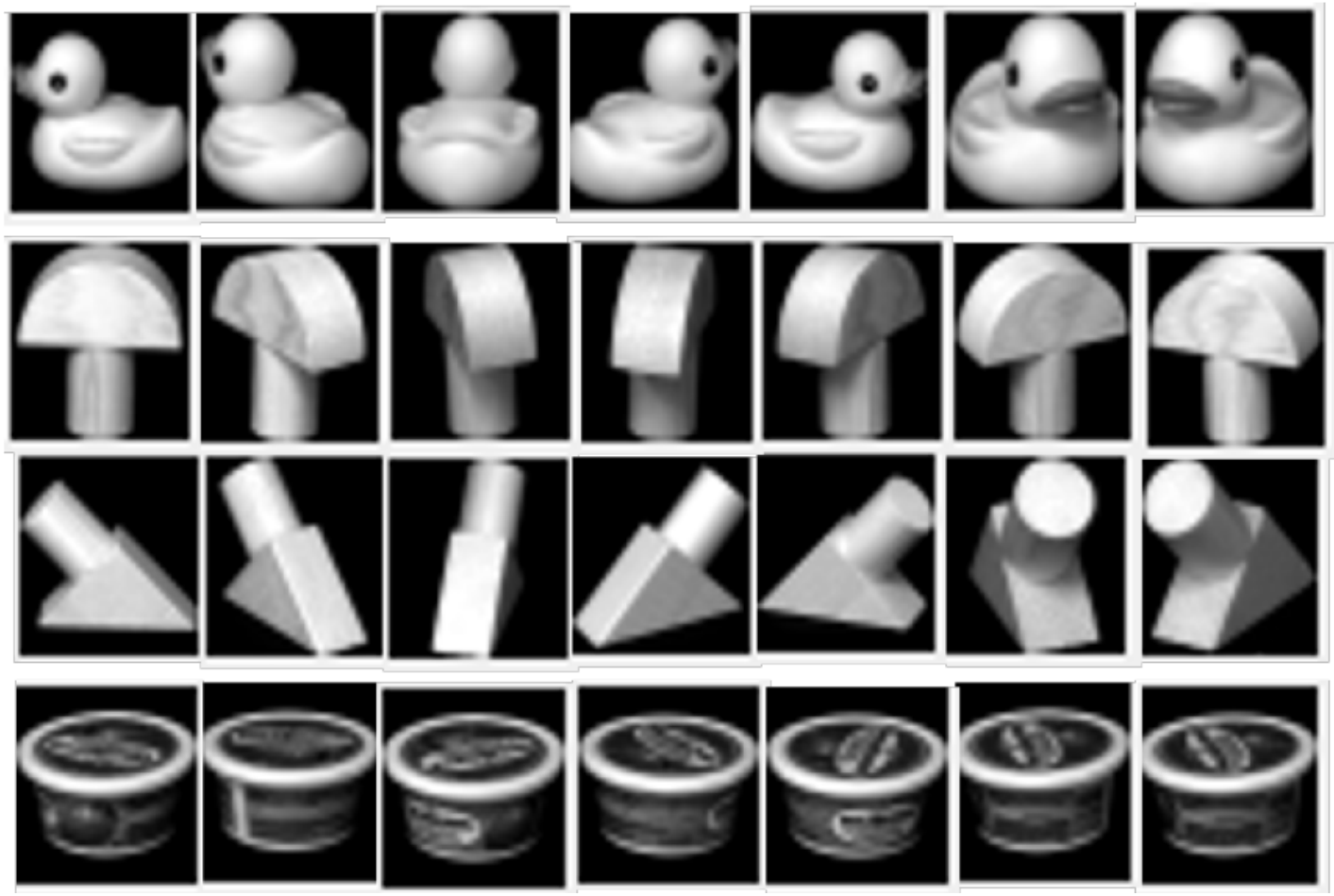


Figure: COIL20 and COIL100

Experiments on COIL20 and COIL100 Datasets

	COIL20			COIL100		
layers	encoder-1	self-expressive	decoder-1	encoder-1	self-expressive	decoder-1
kernel size	3×3	–	3×3	5×5	–	5×5
channels	15	–	15	50	–	50
parameters	150	2073600	136	1300	51840000	1251

Figure: Network settings for COIL20 and COIL100

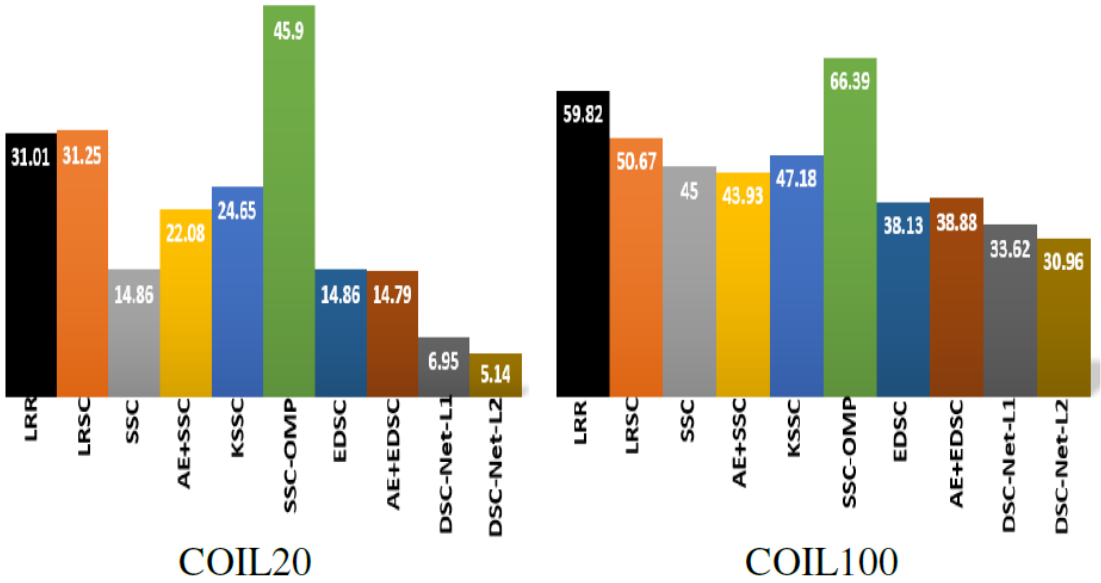


Figure: Subspace clustering error on COIL20 and COIL100 datasets

Summary

- Traditional subspace clustering methods are restricted when data lying on non-linear manifolds.

- We want to handle manifolds data:
  - Kernel Method
  - Neural Network (Self-Expressive Layer)

References I


Basri, R. and Jacobs, D. W. (2003).
Lambertian reflectance and linear subspaces.
IEEE Transactions on Pattern Analysis & Machine Intelligence, (2):218–233.


Costeira, J. and Kanade, T. (1995).
A multi-body factorization method for motion analysis.
In *Proceedings of IEEE International Conference on Computer Vision*, pages 1071–1076. IEEE.


Cover, T. M., Hart, P., et al. (1967).
Nearest neighbor pattern classification.
IEEE transactions on information theory, 13(1):21–27.


Elhamifar, E. and Vidal, R. (2009).
Sparse subspace clustering.
In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797. IEEE.


Ji, P., Zhang, T., Li, H., Salzmann, M., and Reid, I. (2017).
Deep subspace clustering networks.
In *Advances in Neural Information Processing Systems*, pages 24–33.


Kanatani, K.-i. (2001).
Motion segmentation by subspace separation and model selection.
In *Proceedings Eighth IEEE International Conference on computer Vision. ICCV 2001*, volume 2, pages 586–591. IEEE.

References II


Liu, G., Lin, Z., and Yu, Y. (2010).
Robust subspace segmentation by low-rank representation.
In *ICML*, volume 1, page 8.


Lu, C., Feng, J., Lin, Z., Mei, T., and Yan, S. (2018).
Subspace clustering by block diagonal representation.
IEEE transactions on pattern analysis and machine intelligence, 41(2):487–501.


Lu, C.-Y., Min, H., Zhao, Z.-Q., Zhu, L., Huang, D.-S., and Yan, S. (2012).
Robust and efficient subspace segmentation via least squares regression.
In *European conference on computer vision*, pages 347–360. Springer.


Luo, D., Nie, F., Ding, C., and Huang, H. (2011).
Multi-subspace representation and discovery.
In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 405–420. Springer.


Patel, V. M. and Vidal, R. (2014).
Kernel sparse subspace clustering.
In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2849–2853. IEEE.


Soltanolkotabi, M., Elhamifar, E., Candes, E. J., et al. (2014).
Robust subspace clustering.
The Annals of Statistics, 42(2):669–699.

References III


Tipping, M. E. and Bishop, C. M. (1999).
Mixtures of probabilistic principal component analyzers.
Neural computation, 11(2):443–482.


Tseng, P. (2000).
Nearest q-flat to m points.
Journal of Optimization Theory and Applications, 105(1):249–252.


Von Luxburg, U. (2007).
A tutorial on spectral clustering.
Statistics and computing, 17(4):395–416.


Wang, S., Yuan, X., Yao, T., Yan, S., and Shen, J. (2011).
Efficient subspace segmentation via quadratic programming.
In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.