Learning a Self-Expressive Network for Subspace Clustering




Shangzhi Zhang[1], **Chong You**[2], **Rene Vidal**[3], **Chun-Guang Li**[1]

[1] School of Artificial Intelligence, Beijing University of Posts and Telecommunications

[2] Department of EECS, University of California, Berkeley, CA

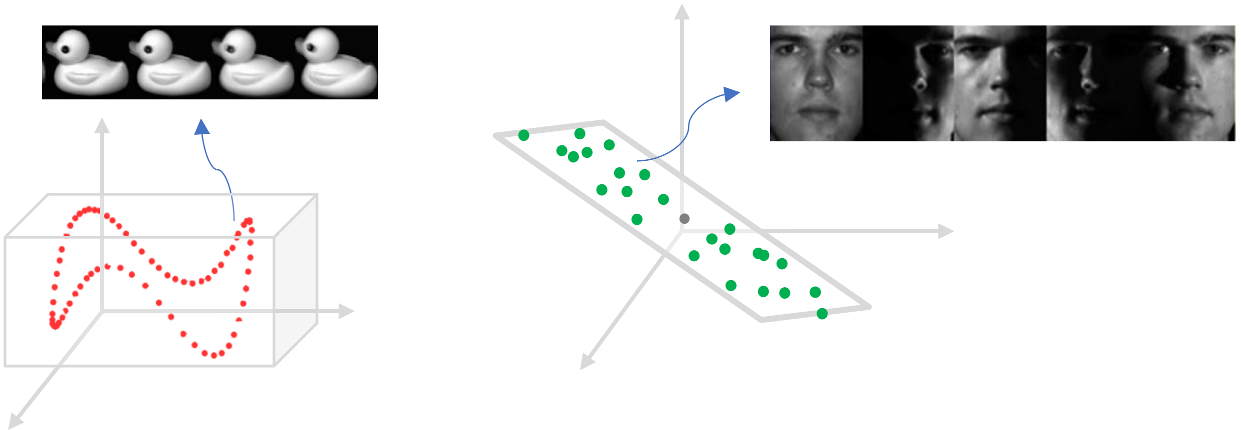
[3] Mathematical Institute for Data Science, Johns Hopkins University, Baltimore, MD

Outline

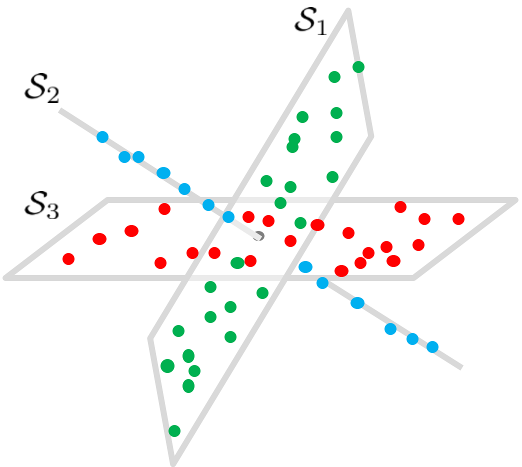
- Introduction to Subspace Clustering
- Related Work and Motivations
- Our Proposal: SENet
- Experiments
- Summary

Latent Structures in High-Dimensional Data

- High dimensional data with a specific semantic pattern (class or category) usually lies in submanifold or **a linear subspace**



(a) manifold (embedded in R^3) (b) linear subspace

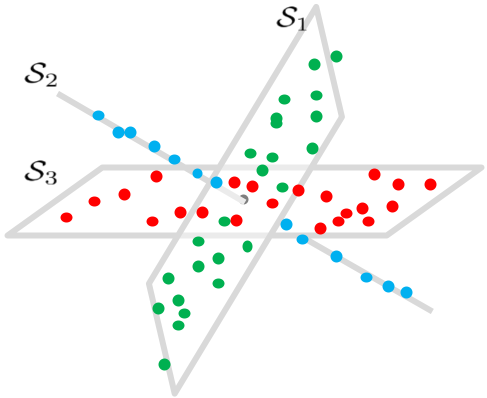


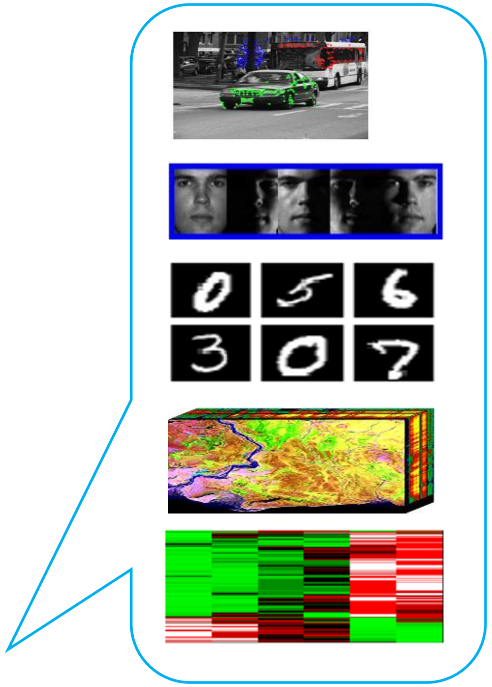
Union of Subspaces

- Complex high dimension data with multiple classes can be modeled by **a union of low dimension subspaces**

Problem Statement: Subspace Clustering

- **Given a set of data points lying (approximately) in a union of subspaces, our goal is to segment data points into each subspace**





- ➢ A subspace corresponds to a pattern (or cluster) in data set
  - ✓ e.g. an object, an individual, a digit, a type of area, a cancer subtype

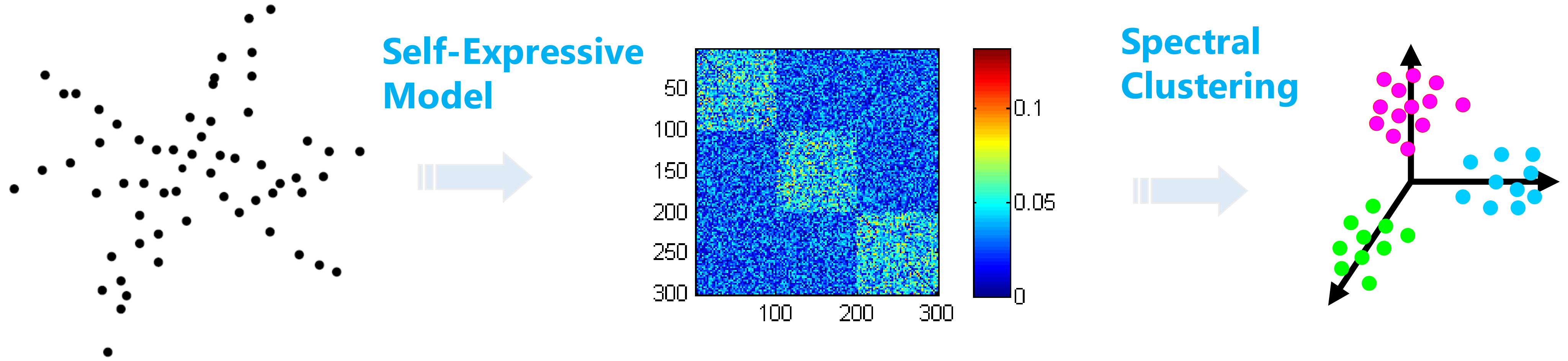
- **Applications:**
  - ➢ e.g. face clustering, cancer subtype discovery, etc.

Outline

- Introduction to Subspace Clustering
- Related Work and Motivations
- Our Proposal: SENet
- Experiments
- Summary

Self-Expressive Model based Methods

- **Step 1: Perform Self-Expressive Model to get self-expressive coefficients**
- **Step 2: Apply Spectral Clustering to the induced affinity matrix**



[1] R. Vidal. Subspace clustering. IEEE Signal Processing Magazine, 28(3):52–68, March 2011.
[2] C.-G. Li, C. You, & R. Vidal, "Structured Sparse Subspace Clustering: A Joint Affinity Learning and Subspace Clustering Framework", IEEE Transactions on Image Processing, 2017.

Subspace Preserving Property

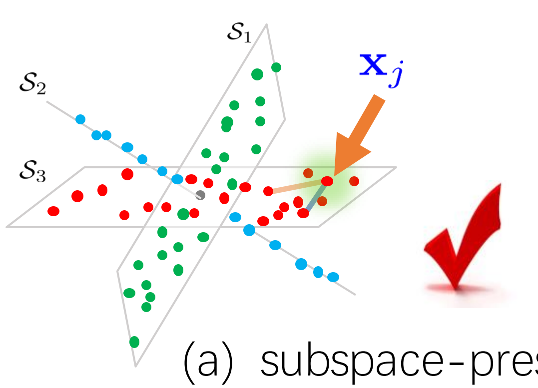
- **Self-Expressive Model**
  - Represent a data point x_j as a linear combination of other points

$$\boldsymbol{x}_j = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2 + \cdots + c_{j-1}\boldsymbol{x}_{j-1} + 0\cdot\boldsymbol{x}_j + c_{j+1}\boldsymbol{x}_{j+1} + \cdots + c_N\boldsymbol{x}_N$$

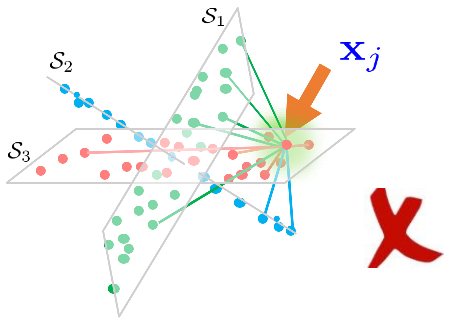
$$\Rightarrow \quad \min_{\boldsymbol{c}_{ij}} \sum_{i\neq j} r(c_{ij}) \quad \text{s.t.} \quad \boldsymbol{x}_j = \sum_{i\neq j} c_{ij}\boldsymbol{x}_i$$

  - **Subspace Preserving Property:** nonzero coefficients correspond *only* to data points in the same subspace as x_j

✓ $r(\boldsymbol{c})$: e.g. $\|\boldsymbol{c}\|_1$ in SSC, $\|\boldsymbol{c}\|_2^2$ in LSR, $\|\boldsymbol{C}\|_*$ in LRR / LRSC , $\|\boldsymbol{c}\|_1 + \|\boldsymbol{c}\|_2^2$ in EnSC



(a) subspace-preserving



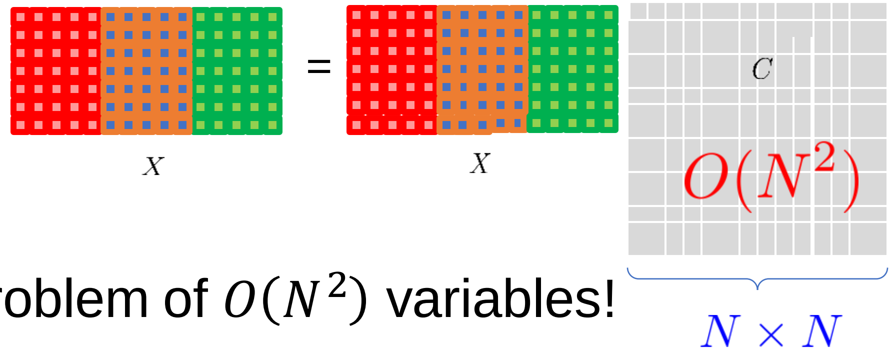
(b) not subspace preserving

[1] E. Elhamifar & R. Vidal: "Sparse subspace clustering", CVPR 2009. [2] E. Elhamifar & R. Vidal: "Sparse subspace clustering: Algorithm, theory, and applications", IEEE TPAMI 2013.

Scalability Issue in Self-Expressive Model

- **Self-Expressive Model**
  - For data point $\boldsymbol{x}_j$, we solve for the self-expressive coefficients from the optimization problem as follows:

$$\min_{\boldsymbol{c}_{ij}} \frac{\gamma}{2}\left\|\boldsymbol{x}_j - \sum_{i\neq j} c_{ij}\boldsymbol{x}_i\right\|_2^2 + \sum_{i\neq j} r(c_{ij})$$



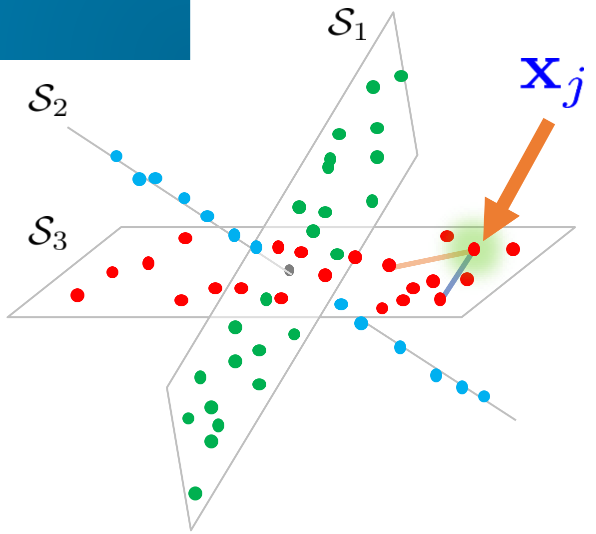
    - Cannot handle large-scale dataset due to solving problem of $O(N^2)$ variables!

- **Scalable Subspace Clustering**
  - Dictionary based methods (e.g., SSSC, OLRSC, ESC, RPCM):
    - Scarify performance when dictionary is small but suffer from **a quadratic complexity** with dictionary size
  - Decomposition based methods: greedy method (e.g., SSCOMP), active support (e.g., EnSC), dropout (e.g., S³COMP)
    - Still have $O(N^2)$ time and memory requirement

Outline

- Introduction to Subspace Clustering
- Related Work and Motivations
- Our Proposal: SENet
- Experiments
- Summary

Reparametrize Self-Expressive Coefficients



- **Reparametrize Self-Expressive Model**
  - For data point $\boldsymbol{x}_j$ in a **linear** subspace, we have:

$$\boldsymbol{x}_j = \sum\nolimits_{i \neq j} c_{ij} \boldsymbol{x}_i \quad \Rightarrow \quad \boldsymbol{x}_j = \sum\nolimits_{i \neq j} f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta) \boldsymbol{x}_i$$

  - **(Subspace-Preserving Property):** desire that $f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta)$ yields nonzero outputs only correspond to data points in the same subspace as $\boldsymbol{x}_j$

$$\min_{\boldsymbol{c}_{ij}} \frac{\gamma}{2} \|\boldsymbol{x}_j - \sum_{i \neq j} c_{ij} \boldsymbol{x}_i\|_2^2 + \sum_{i \neq j} r(c_{ij}), \qquad \text{where} \quad r(c_{ij}) = \lambda |c_{ij}| + \frac{1-\lambda}{2} c_{ij}^2,$$

EnSC (You et al. CVPR16)

Reparameterization

$$\min_{\Theta} \frac{\gamma}{2} \|\boldsymbol{x}_j - \sum_{i \neq j} f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta) \boldsymbol{x}_i\|_2^2 + \sum_{i \neq j} r(f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta)),$$

Self-Expressive Network (SENet)

- **We use a query-key network with learnable soft thresholding activation function to implement $f(x_i, x_j; \Theta)$**

$$f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta) := \mathcal{T}_b(\boldsymbol{u}_j^\top \boldsymbol{v}_i) \in \mathbb{R}$$

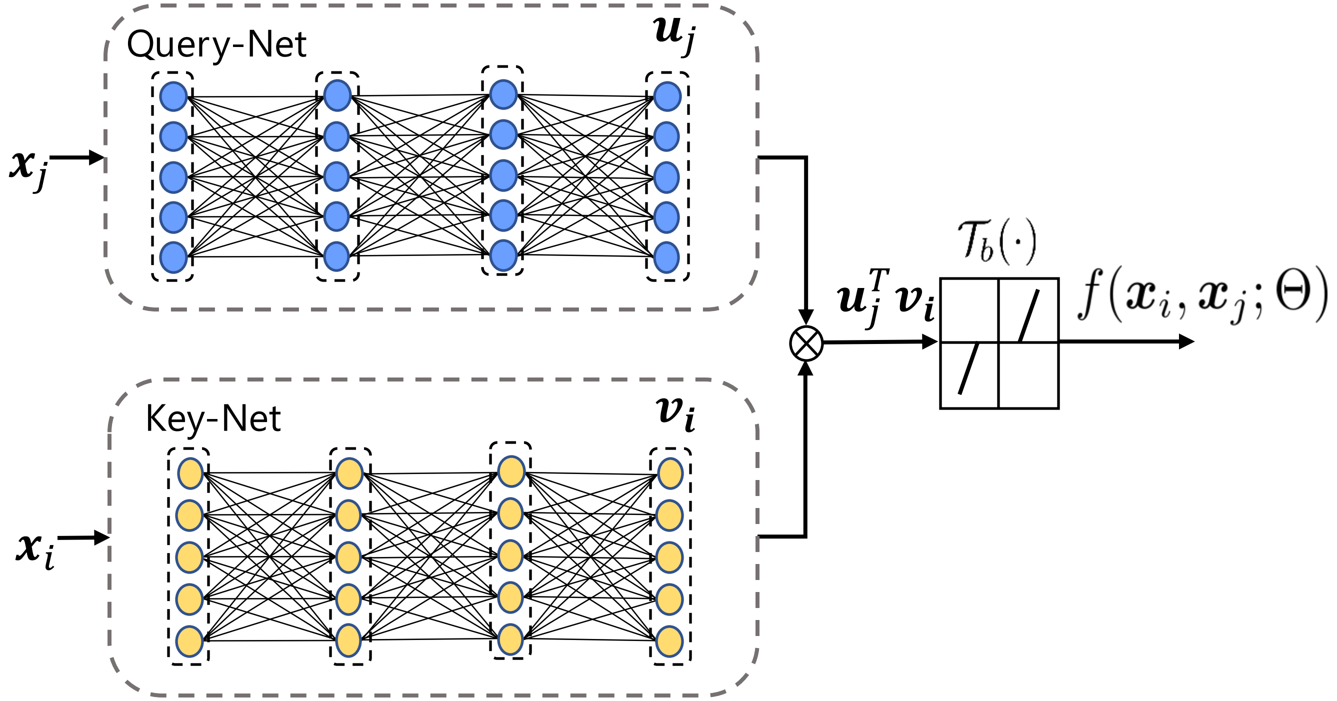
$$\boldsymbol{u}_j := \boldsymbol{u}(\boldsymbol{x}_j; \Theta_u) \in \mathbb{R}^p$$

$$\boldsymbol{v}_i := \boldsymbol{v}(\boldsymbol{x}_i; \Theta_v) \in \mathbb{R}^p$$

$$\mathcal{T}_b := \operatorname{sgn}(t) \max\{0, |t| - b\}$$

$$\boldsymbol{x}_j = \sum_{i \neq j} f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta) \boldsymbol{x}_i$$

$$\boldsymbol{x}_j = \sum_{i \neq j} f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta) \boldsymbol{x}_i = \sum_{i \neq j} \mathcal{T}_b(\boldsymbol{u}_j^\top \boldsymbol{v}_i; \Theta) \boldsymbol{x}_i$$



SGD based Training Algorithms

- **Naïve SGD and Two-pass SGD:**
  - **Naïve SGD:** For data point $\boldsymbol{x}_j$ and data matrix X, we have:

$$\ell(\boldsymbol{x}_j, X; \Theta) := \frac{\gamma}{2}\|\boldsymbol{x}_j - \sum_{i \neq j} f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta)\boldsymbol{x}_i\|_2^2 + \sum_{i \neq j} r(f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta)),$$

where $r(\cdot) = \lambda|\cdot| + \frac{1-\lambda}{2}(\cdot)^2$ and compute gradients as follows:

$$\frac{\partial \ell(\boldsymbol{x}_j, X; \Theta)}{\partial \Theta} := \sum_{i \neq j} \left(r'(f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta)) - \boldsymbol{x}_i^\top \boldsymbol{q}_j\right) \frac{\partial f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta)}{\partial \Theta},$$

where $\boldsymbol{q}_j := \gamma\left(\boldsymbol{x}_j - \sum_{i \neq j} f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta)\boldsymbol{x}_i\right).$

  - **Two-pass SGD:** gradient is a weighted sum and **accumulated in an online fashion** (constant space requirement)

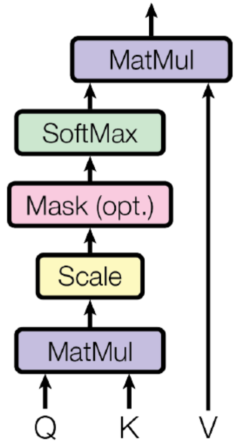
Connections to Attention Models

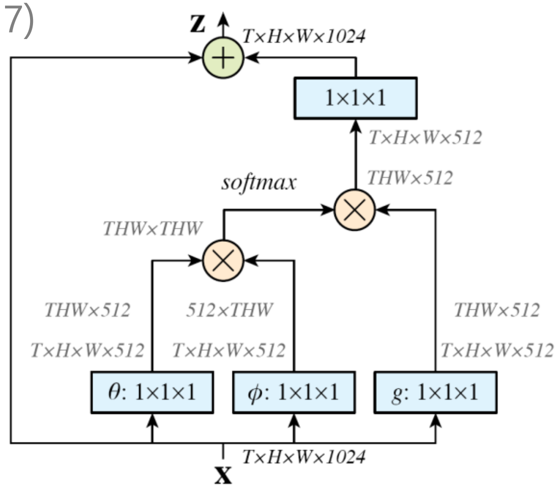
- **Three Self-Attention Models**

  - Scaled Dot-Product Attention

$$\text{Attention}(Q,K,V) := \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)$$

Dot-Product Att. (Vasvani et al. NerIPS'17)





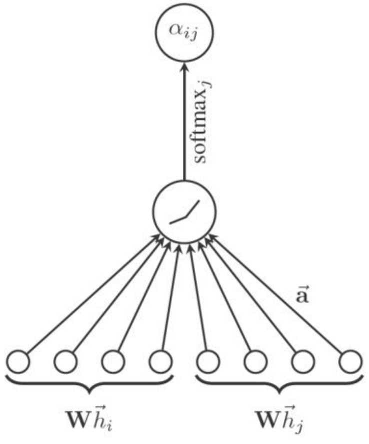
Non-Local NNs (Wang et al. CVPR'18)

  - Non-Local Neural Networks:

$$\boldsymbol{y}_j := \frac{1}{C(\boldsymbol{x}_j)} \sum_j f(\boldsymbol{x}_i, \boldsymbol{x}_j; \Theta) g(\boldsymbol{x}_i)$$

  - GAT (Graph Attention Networks):

GAT (Velickovic et al. ICLR'18)



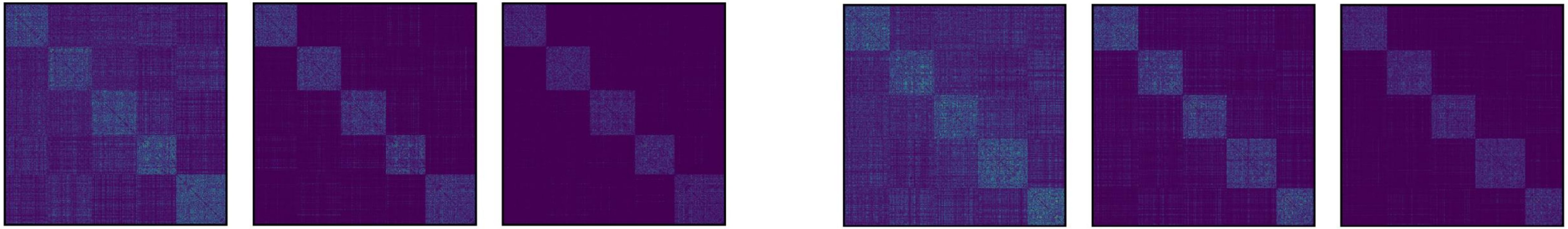
$$\alpha_{ij} := \frac{\exp(\text{LeakyReLU}(\boldsymbol{a}^\top [W\boldsymbol{h}_i \| W\boldsymbol{h}_j]))}{\sum_{k\in\mathcal{N}_i} \exp(\text{LeakyReLU}(\boldsymbol{a}^\top [W\boldsymbol{h}_i \| W\boldsymbol{h}_k]))}$$

Outline

- Introduction to Subspace Clustering
- Related Work and Motivations
- Our Proposal: SENet
- Experiments
- Summary

Experiments on Synthetic Data

- **Synthetic Data**: Generate 5 subspaces of dimension 6 in R^{15}, and sample 500 points as training data and 500 points as test data



(a) $C_{tr}^{(100)}$ (64%) (b) $C_{tr}^{(300)}$ (23%) (c) $C_{tr}^{(500)}$ (10%) (d) $C_{ts}^{(100)}$ (66%) (e) $C_{ts}^{(300)}$ (37%) (f) $C_{ts}^{(500)}$ (24%)

- Left: show the evaluated self-expressive coefficient matrix $|C_{\text{train}}^{(t)}|$ of SENet at the t-th iteration
- Right: show the inferred self-expressive coefficients matrix $|C_{\text{test}}^{(t)}|$ on test data

  - SRE (%): Subspace Recovery Error

Experiments on Synthetic Data

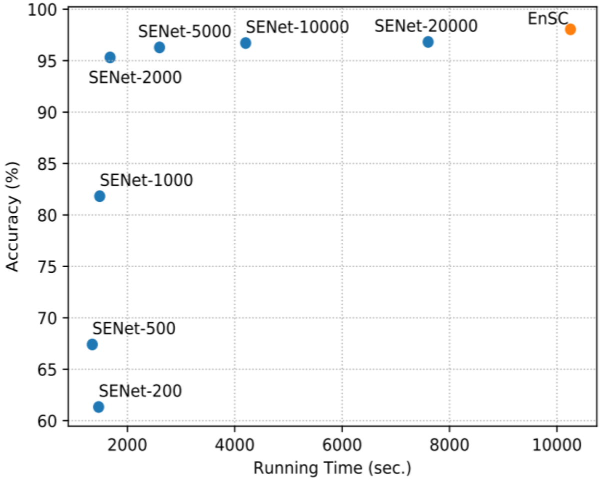
- **Synthetic Data**: Generate 5 subspaces of dimension 6 in R^9, and sample N_i points per subspace as training data

N_i	Methods	Metrics $\mathcal{L}$	$\mathcal{L}_{rec}$	$\mathcal{L}_{reg}$	ACC (%)	SRE (%)	CONN
20	EnSC	135.127	0.107	132.442	72.0	49.611	0.178
	SENet	135.132	0.109	132.416	71.0	49.720	0.178
	SENet test	1830.107	72.007	29.937	65.0	58.384	0.318
100	EnSC	559.943	0.526	558.009	93.0	27.370	0.163
	SENet	559.972	0.531	558.022	92.8	27.501	0.165
	SENet test	2935.424	89.325	702.309	79.0	56.897	0.387
200	EnSC	1053.086	0.526	1040.097	96.6	20.067	0.155
	SENet	1053.369	0.531	1040.097	96.0	20.195	0.159
	SENet test	17826.273	599.099	2848.779	84.1	56.256	0.398
1000	EnSC	4884.876	2.095	4832.508	99.4	6.493	0.126
	SENet	4932.907	2.205	4877.781	99.5	9.132	0.155
	SENet test	30037.012	887.323	7853.945	92.3	36.054	0.236
2000	EnSC	9576.154	3.958	9477.197	99.7	4.580	0.108
	SENet	10025.874	4.592	9911.074	99.4	13.555	0.201
	SENet test	44458.734	1453.790	8113.975	97.4	21.863	0.220

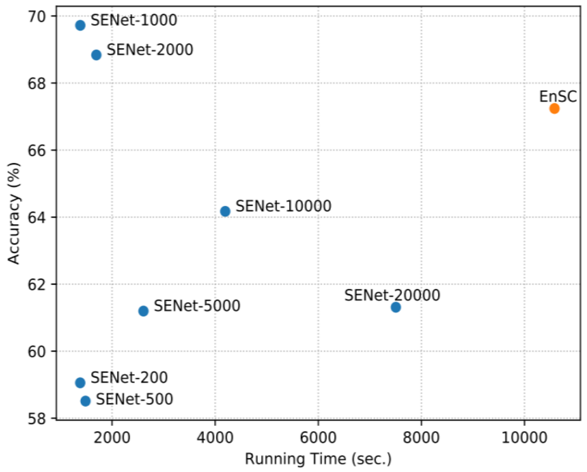
Table 1. Comparing SENet to EnSC on synthetic data

➢ SENet is able to well approach the global solution of EnSC

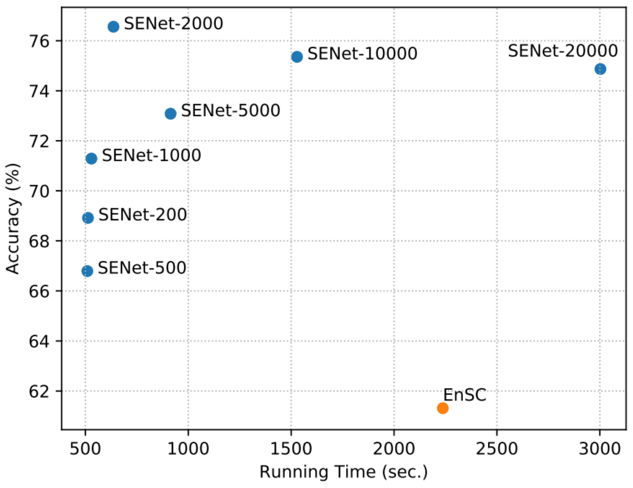
Experiments on Real World Data: Time Costs



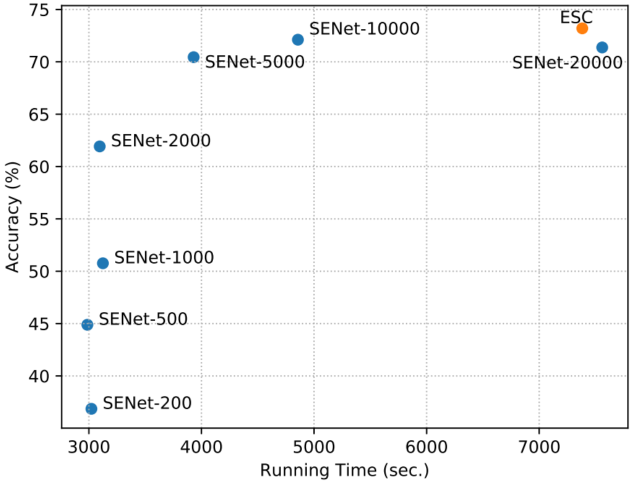
(a) MNIST



(b) Fashion-MNIST



(c) CIFAR-10



(d) EMNIST

Experiments on Real World Data

- **Comparison to state of the art**

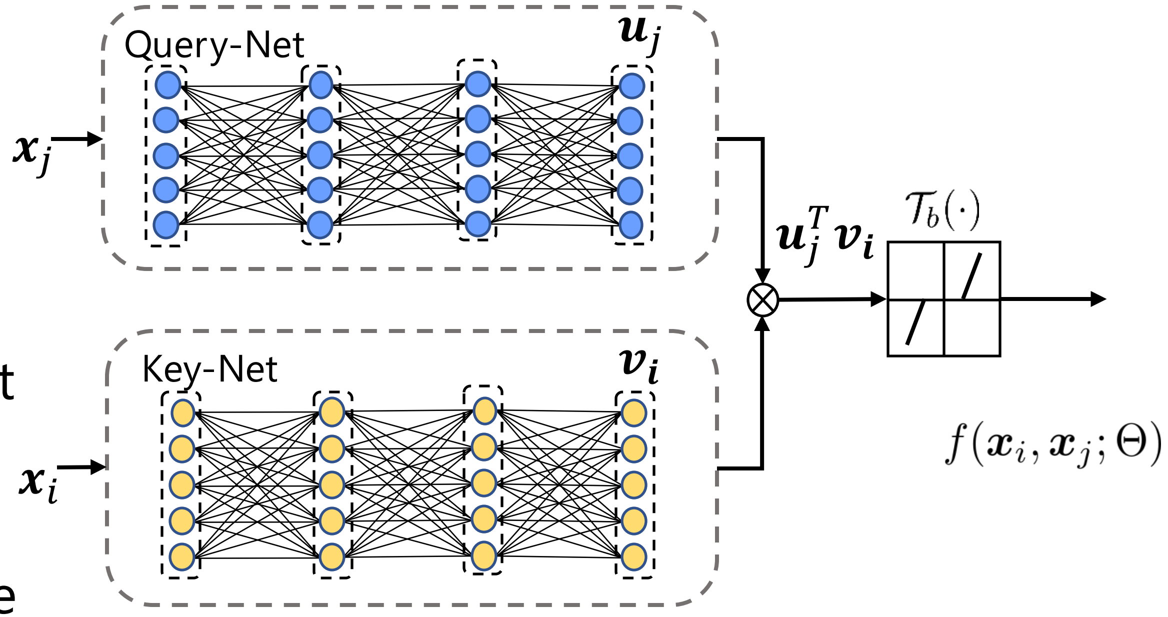
Methods	MNIST-full			Fashion-MNIST-full			CIFAR-10			EMNIST		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
k-means [42]	0.541	0.507	0.367	0.505	0.578	0.403	0.525	0.589	0.276	0.459	0.438	0.316
Spectral [55]	0.728	0.856	0.667	0.625	0.700	0.494	0.455	0.574	0.256	0.662	0.769	0.654
JULE [73]	0.964	0.913	0.927	0.563	0.608	-	0.272	0.192	0.138	-	-	-
DEC [72]	0.863	0.834	-	0.518	0.546	-	0.301	0.257	0.161	-	-	-
DAC [8]	0.978	0.935	0.949	-	-	-	0.522	0.396	0.306	-	-	-
DEPICT [19]	0.965	0.917	-	0.392	0.392	-	-	-	-	-	-	-
ClusterGAN [46]	0.905	0.890	-	0.662	0.645	-	-	-	-	-	-	-
DSCDAN [75]	0.978	0.941	-	0.662	0.645	-	-	-	-	-	-	-
DCCM [70]	-	-	-	-	-	-	0.623	0.496	0.408	-	-	-
SSC-OMP [81]	0.928	0.842	0.849	0.274	0.421	0.196	0.326	0.498	0.196	0.654	0.661	0.634
NCSC [89]	0.941	0.861	0.875	**0.721**	0.686	**0.592**	-	-	-	-	-	-
EnSC [79]	**0.980**	**0.945**	**0.957**	0.672	0.705	0.565	0.613	0.601	0.430	T	T	T
ESC [77]	0.971	0.925	0.936	0.668	**0.708**	0.556	0.653	0.629	0.438	**0.732**	**0.825**	0.759
SENet	0.968	0.918	0.931	0.697	0.663	0.556	**0.765**	**0.655**	**0.573**	0.721	0.798	**0.766**

Outline

- Introduction to Subspace Clustering
- Related Work and Motivations
- Our Proposal: SENet
- Experiments
- Summary

Summary: Take Home Messages

- **We proposed to learn a Self-Expressive Network for subspace clustering**
  - Proposed a **query-key network** with a **learnable soft thresholding activation** to reparametrize self-expressive coefficients
  - Proposed **SGD based algorithms** to train SENet with reconstruction loss and elastic net regularization
  - Conducted extensive experiments to validate the **generalization ability** to out-of-sample data and the **potential ability to handle** subspace clustering on **large-scale** data



Acknowledgement

- S. Zhang and C.-G. Li are supported by NSFC under grant 61876022.

Thank you for your attention!

Shangzhi Zhang

Chong You

Rene Vidal

Chun-Guang Li