

ALPCA HUS: Heteroscedastic Subspace Clustering

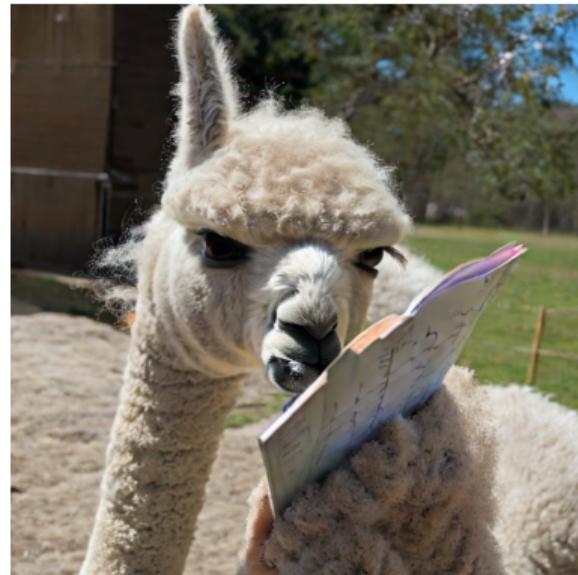
Javier Salazar Cavazos

Joint collaboration with
Laura Balzano & Jeff Fessler

University of Michigan

May 3, 2023

Image Credit: ???



Preface

- ① Originally, presentation about convergence guarantees (nonconvex)

- ② Gave a presentation a year ago about Sparse Subspace Clustering
- ③ Goal: Heteroscedastic version of SSC
- ④ Result thus far = pain :(
- ⑤ Instead, worked on **heteroscedastic PCA** (quals presentation)
- ⑥ This presentation covers **ALPCA** → Union of Subspace model

Subspaces

For a vector space \mathcal{V} defined on a field \mathbb{F} , a nonempty set $\mathcal{S} \subseteq \mathcal{V}$ is called a *linear* subspace iff

- \mathcal{S} is closed under vector addition (i.e. for $\mathbf{u}, \mathbf{v} \in \mathcal{S} \implies \mathbf{u} + \mathbf{v} \in \mathcal{S}$)
- \mathcal{S} is closed under scalar multiplication (i.e. $\mathbf{v} \in \mathcal{S} \implies \alpha\mathbf{v} \in \mathcal{S}$)

Key Points:

- Every *linear* subspace includes $\mathbf{0}$
- A shifted *linear* subspace is called an affine subspace
- Left singular matrix contains subspace basis

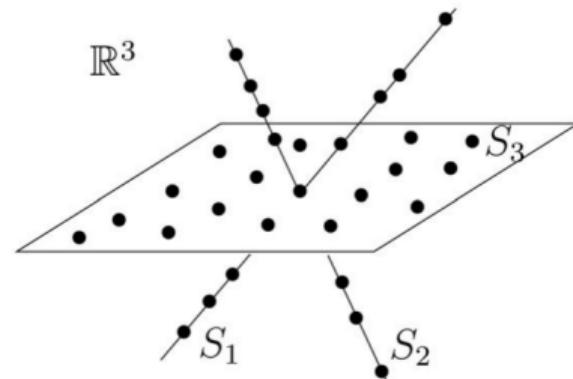
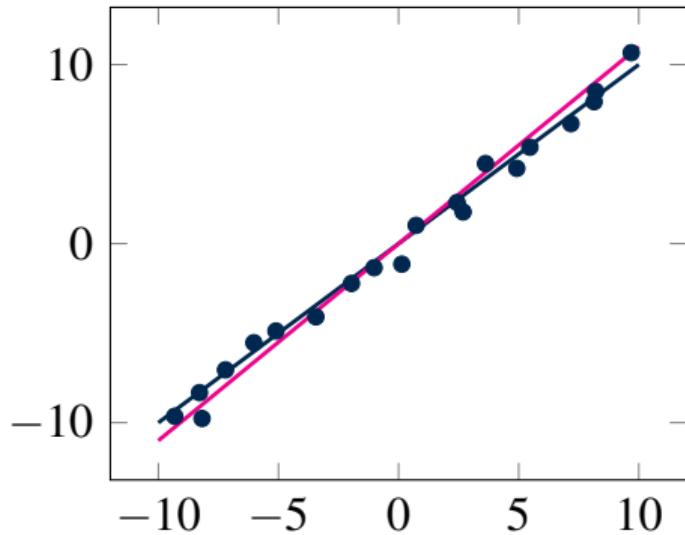


Image Credit: René Vidal @ Johns Hopkins University/University of Pennsylvania

Heteroscedasticity

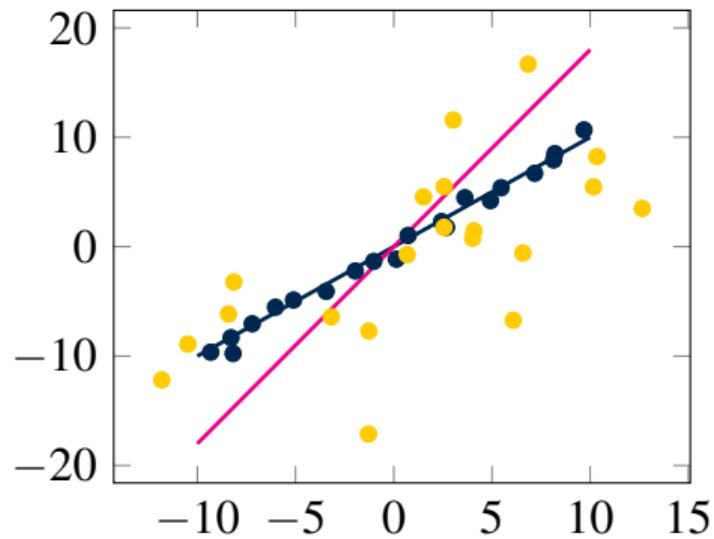
Homoscedastic Data

$$\mathbf{y}_i = \mathbf{x}_i + \boldsymbol{\epsilon} \text{ s.t. } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \nu I)$$



Heteroscedastic Data

$$\mathbf{y}_i = \mathbf{x}_i + \boldsymbol{\epsilon}_i \text{ s.t. } \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \nu_i I)$$



$\mathbf{x}_i = \mathbf{U}\mathbf{z}_i$ where $\hat{\mathbf{U}}$ = estimated subspace basis, \mathbf{z}_i = basis coordinates

ALPCA [1] (Algorithm Low-rank PCA Hetero. data)

Key Idea

ALPCA = Robust PCA (RPCA) + Heteroscedastic Noise

Let $Y = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{D \times N}$ and $\Pi = \text{diagm}(\nu_1, \dots, \nu_n) \in \mathbb{R}^{N \times N}$

Decompose Y such that $\underbrace{Y}_{\text{data matrix}} = \underbrace{X}_{\text{low rank data}} + \underbrace{Z}_{\text{noise matrix}}$

Then the optimization problem we posed is the following:

$$\arg \min_{X, Z, \Pi} \lambda \underbrace{f_k(X)}_{\text{low rank}} + \underbrace{\frac{1}{2} \|Z\Pi^{-1/2}\|_F^2}_{\text{weighted noise}} + \underbrace{\frac{D}{2} \log \det |\Pi|}_{\text{unk. variance}} \quad \text{s. t. } Y = X + Z$$

ALPCA [1] (Algorithm Low-rank PCA Hetero. data)

Let $Y = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{D \times N}$ and $\Pi = \text{diagm}(\nu_1, \dots, \nu_n) \in \mathbb{R}^{N \times N}$

$$\arg \min_{X, Z, \Pi} \underbrace{\lambda f_k(X)}_{\text{low rank}} + \underbrace{\frac{1}{2} \|Z\Pi^{-1/2}\|_F^2}_{\text{weighted noise}} + \underbrace{\frac{D}{2} \log \det |\Pi|}_{\text{unk. variance}} \quad \text{s. t. } Y = X + Z$$

$$f_k(X) \triangleq \sum_{i=k+1}^{\min(M,N)} \sigma_i(X) = \sum_{i=1}^{\min(M,N)} \sigma_i(X) - \sum_{i=1}^k \sigma_i(X) = \|X\|_* - \|X\|_{\text{Ky-Fan}(k)}$$

Note!

$k = 0 \implies f_k(X) = \|X\|_*$ (convex assuming known variance)

$k > 0, \lambda \rightarrow \infty \implies \hat{X} \implies SVP(X, \alpha) = U_k \Sigma_k V_k^T$ (nonconvex)

Fast ALPCA

Key Idea

FAST ALPCA = ALPCA + Matrix Factorization

Let $Y = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{D \times N}$ and $\Pi = \text{Diagonal}(\nu_1, \dots, \nu_n) \in \mathbb{R}^{N \times N}$

Decompose X such that $X = LR'$ where $L, R \in \mathbb{R}^{D \times k}$

Then the optimization problem we posed is the following:

$$\arg \min_{L, R, \Pi} \underbrace{\frac{1}{2} \| (Y - \textcolor{red}{LR'}) \Pi^{-1/2} \|_F^2}_{\text{new term}} + \frac{D}{2} \log |\Pi|$$

note: not the focus for today, but it appears fast at the cost of accuracy

Subspace Clustering

Motivation



Face Clustering [2]

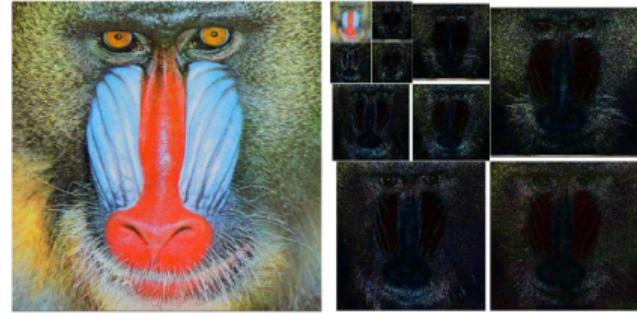


Image Compression [3]

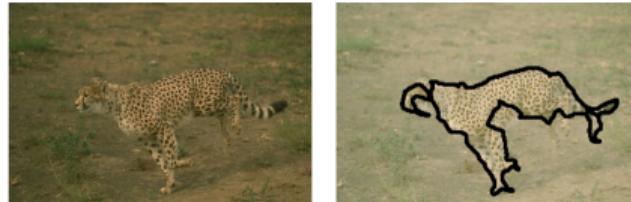
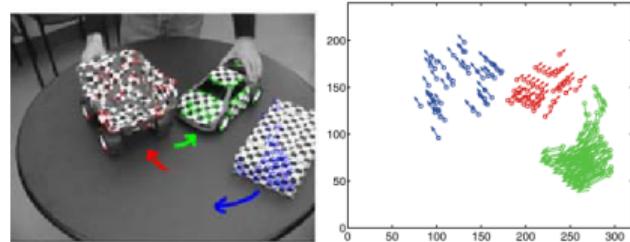


Image Segmentation [4]



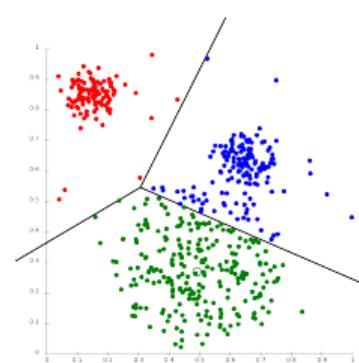
Motion Estimation [5]

Image Credit: Respective Papers

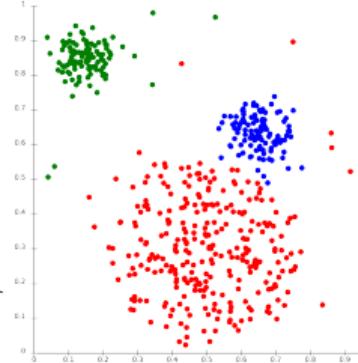
Clustering

An unsupervised machine learning method to identify and group “similar” unlabeled data points in order to find structure or patterns

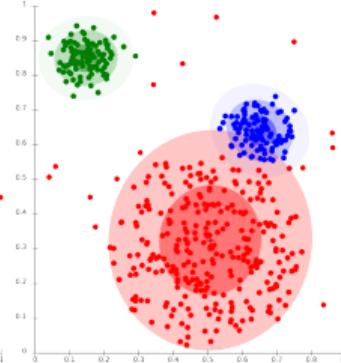
By “similar”, this can mean different things:



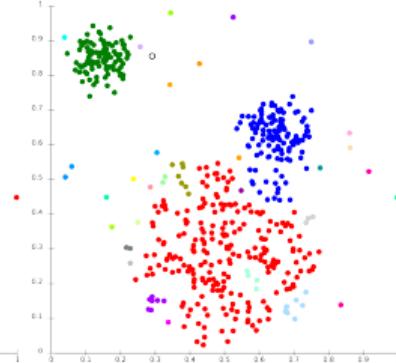
Centroid-based



Density-based



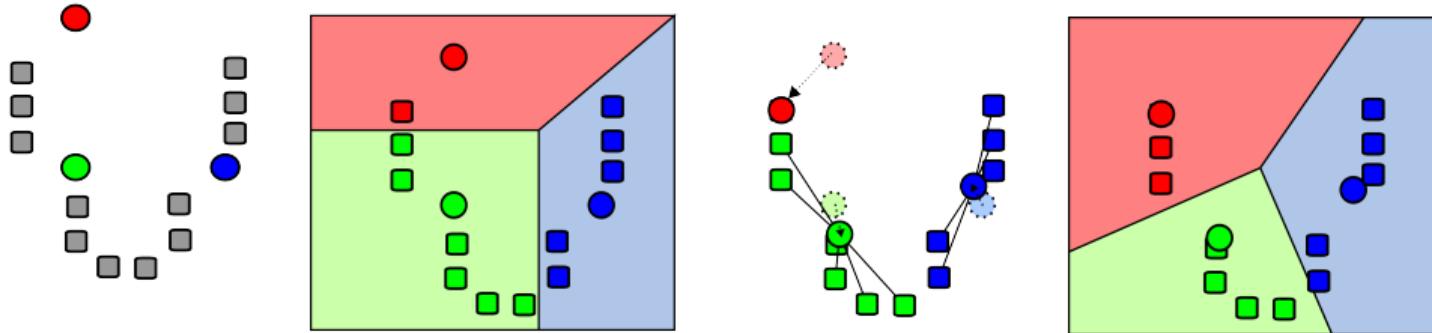
Model-based



Hierarchical-based

Image Credit: Chire @ Wikipedia, CC BY-SA 3.0

K-Means Clustering



1. Initialize
centroids

2. Generate
clusters

3. Update
centroids

4. Repeat steps
2 & 3

Note!

NP-hard problem \implies initialization is extremely important!

Image Credit: Weston.pace @ Wikipedia, CC BY-SA 3.0

Subspaces + Clustering

Points:

$Y = [y_1, \dots, y_N]$ for $\{y_i \in \mathbb{R}^D\}_{i=1}^N$ drawn from $\{U_i\}_{i=1}^K$

Subspaces:

$\mathcal{U} = \{U_i \in \mathbb{R}^{D \times d_i}\}$ s.t. $\exists \alpha \in \{1, \dots, K\} \rightarrow y_i = U_\alpha z_i + \epsilon_i$

Clusters:

$\mathcal{C} = \{c_i : c_i \subset \{1, \dots, N\}\}$ s.t. $Y^{(c_i)} \in \mathbb{R}^{D \times |c_i|} \subset Y \in \mathbb{R}^{D \times N}$

Goal:

Find subspace bases \mathcal{U} and clusters \mathcal{C}

Literature Review [6]

Algebraic Methods (e.g. Generalized PCA)

- Geometric and algebraic interpretations

Iterative Methods (e.g. K-Subspaces)

- Alternate between segmentation and subspaces

Statistical Methods (e.g. Mixture of Probabilistic PCA)

- Model assumptions combined with prior beliefs

Affinity-based Methods (e.g. Sparse Subspace Clustering)

- Measures “similarity” between points

K-Subspaces (KSS)

KSS = K-means + Subspaces

KSS cost function [7]:

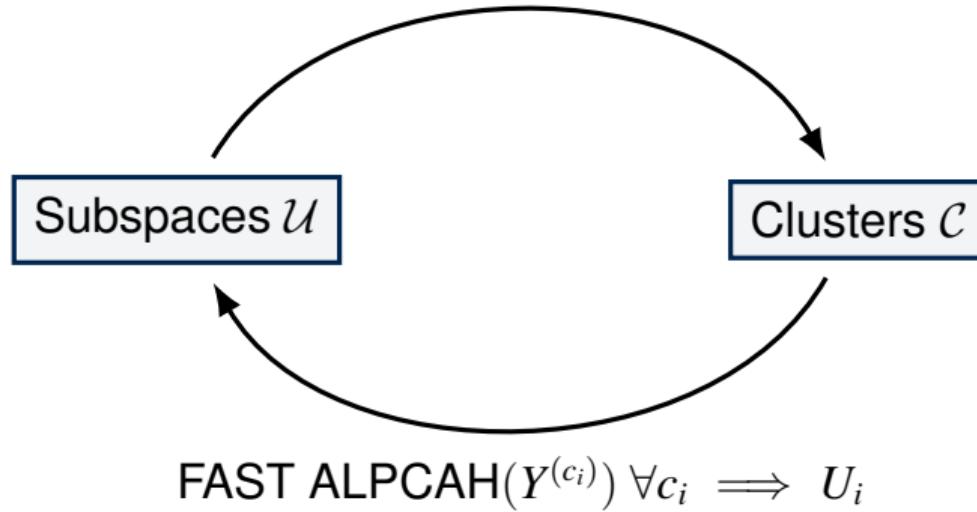
$$\min_{\mathcal{C}, \mathcal{U}} \sum_{k=1}^K \sum_{i: i \in c_k} \|y_i - U_k U_k^T y_i\|_2^2$$

ALPCA HUS cost function:

$$\min_{\mathcal{C}, \mathcal{U}} \sum_{k=1}^K \|Y^{(c_k)} - \underbrace{\mathcal{P}(L_k)}_{\text{projection matrix}} Y^{(c_k)}\|_F^2 \quad \text{s.t.} \quad L_k, R_k = \min_{L, R, \Pi} \frac{1}{2} \|(Y^{(c_k)} - LR')\Pi^{-1/2}\|_F^2 + \frac{D}{2} \log |\Pi|$$

Intuition

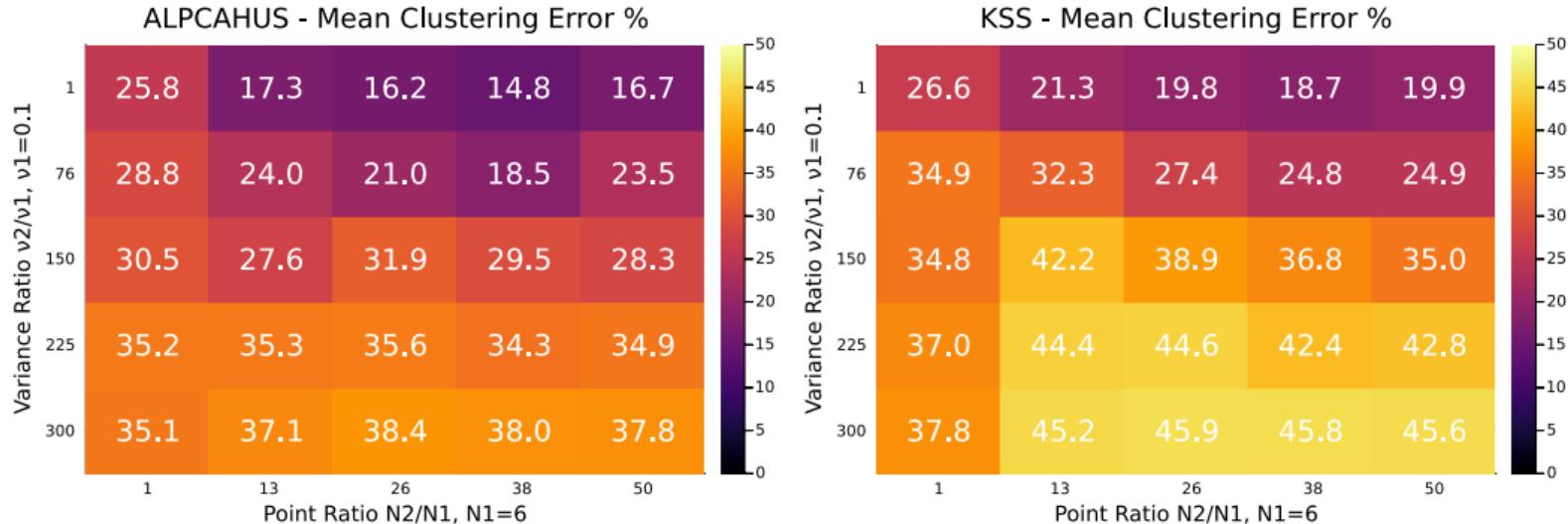
$$\min_i \underbrace{\|y_p - \underbrace{U_i U_i' y_p}_{\text{residual}}\|}_{\text{subspace projector}} \quad \forall y_p \implies p \in c_i$$



Experimental Setup

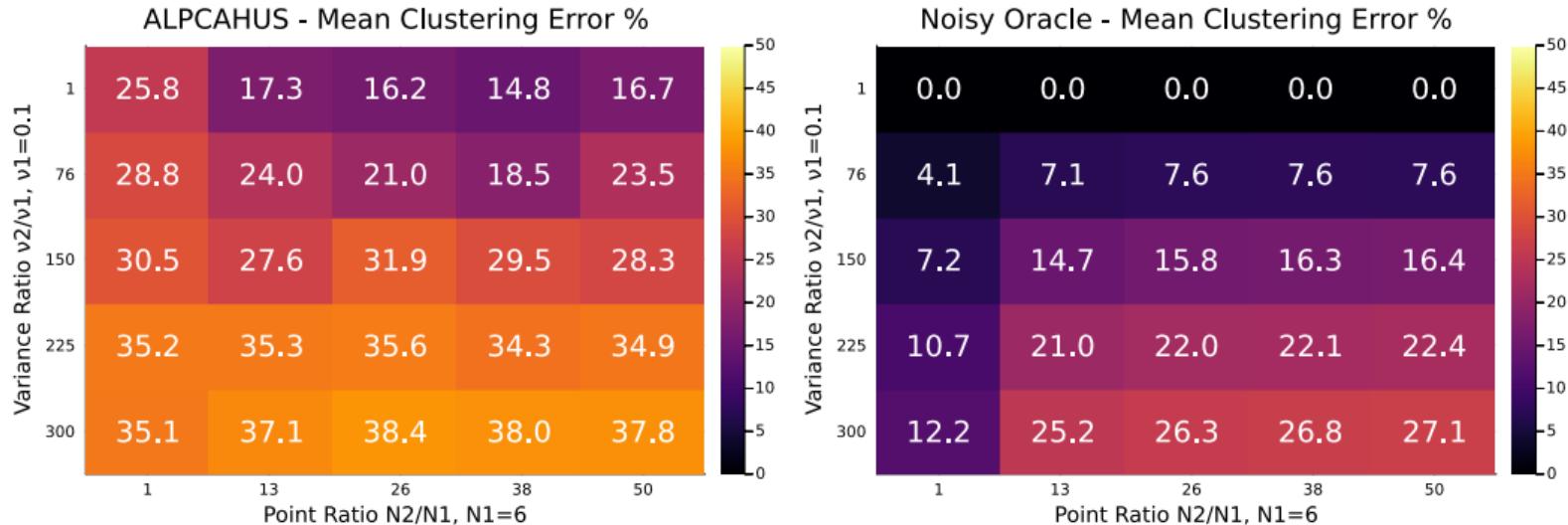
- ① Gaussian random matrix \rightarrow SVD $\rightarrow U_k \quad \forall k \in \{1, 2\} \quad d_k = 3$
- ② Uniform random vector $\rightarrow z_i \sim U[-10, 10] \rightarrow x_i = U_k z_i$
- ③ Gaussian noise $\rightarrow \epsilon_i \sim \mathcal{N}(0, \nu_i I) \rightarrow y_i = x_i + \epsilon_i$
- ④ Each subspace composed of data with noise variances $\nu_1 = 0.1, \nu_2$
- ⑤ Each subspace contains good data and bad data $N1 = 6, N2$
- ⑥ Mean clustering error will be measured (misclassification rate)

ALPCA HUS vs. KSS - Result I



always performs better, even in homoscedastic setting?

ALPCA HUS vs. KSS - Result II



Is it possible to get closer?
question: what is a disadvantage of KSS?

Ensemble KSS (EKSS) [7]

Key Idea

KSS very sensitive to initialization \therefore leverage info from many trials!

For each trial $b \in \{1, \dots, B\}$, collect all clustering results $C = [c_1, \dots, c_B]$

Form co-association matrix (affinity matrix)

$$A_{ij} \leftarrow \frac{1}{B} |\{b : x_i, x_j \text{ are co-clustered in } C^{(b)}\}|$$

essentially, points classified similarly have a high affinity

Affinity Matrix

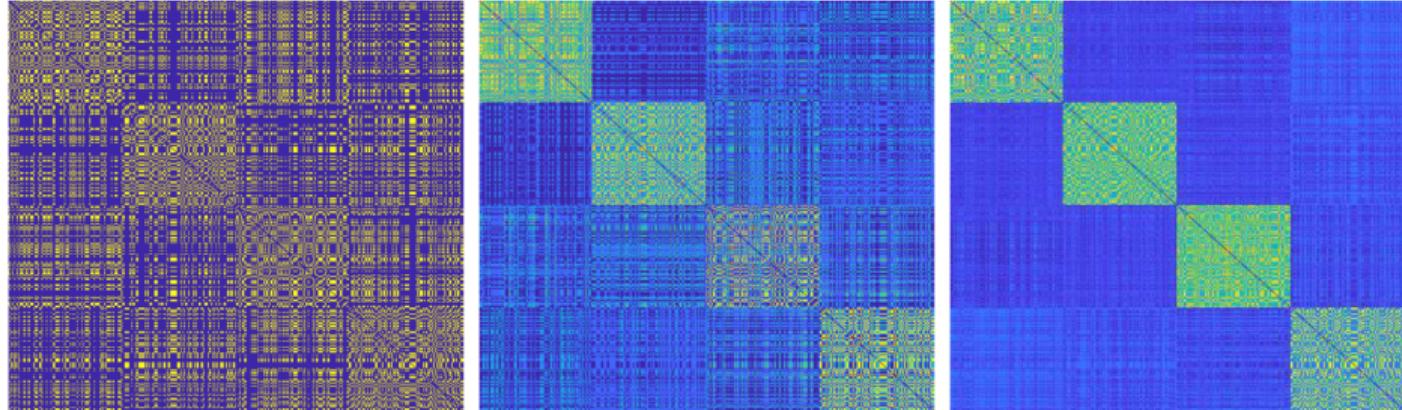


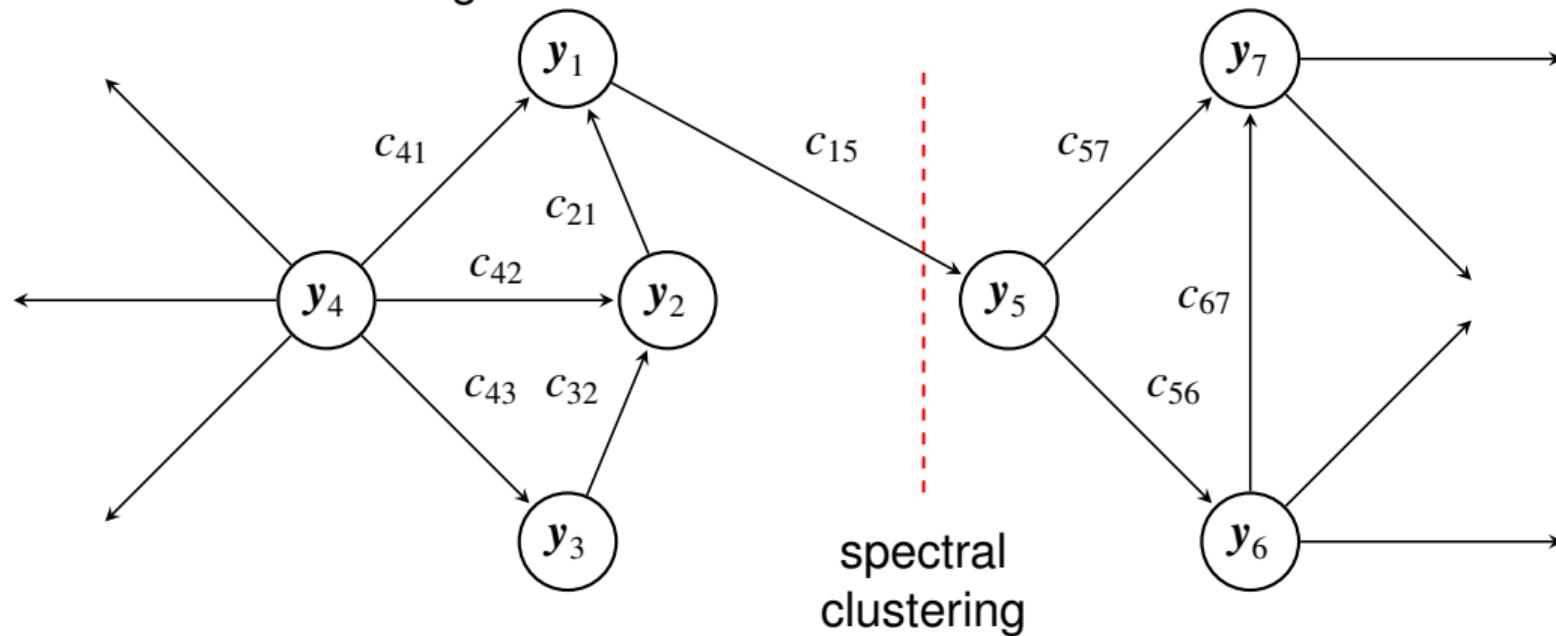
Fig. 1. Co-association matrix of EKSS for $B = 1, 5, 50$ base clusterings. Data generation parameters are $D = 100$, $d = 3$, $K = 4$, $N = 400$, and the data is noise-free; the algorithm uses $\bar{K} = 4$ candidate subspaces of dimension $\bar{d} = 3$ and no thresholding. Resulting clustering errors are 61%, 25%, and 0%.

note: structure is nice only because data is ordered!
question: how do we get the clusters from this matrix?

Image Credit: Authors from [7]

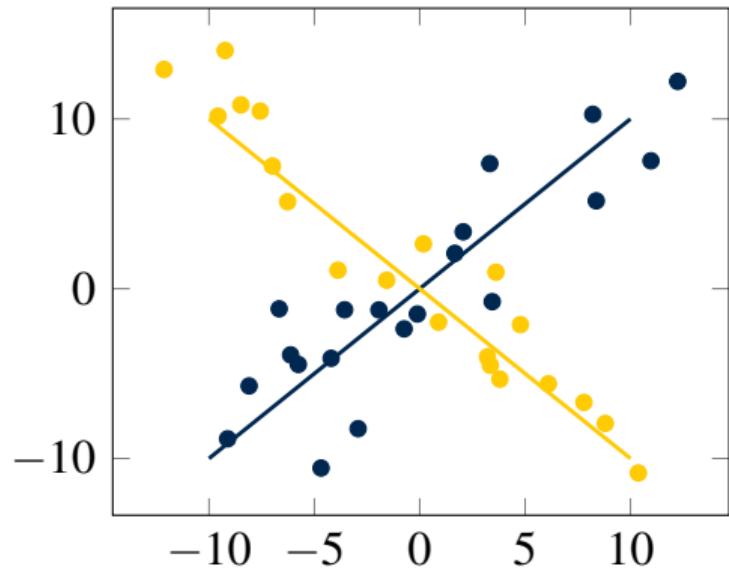
Spectral Clustering [8]

Data points \Rightarrow nodes/vertices
coefficients \Rightarrow edges

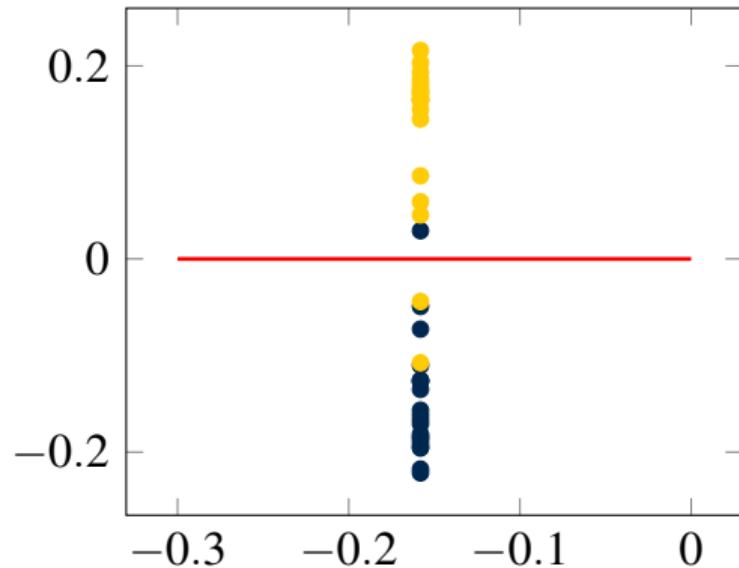


Toy Example

sample data



spectral embedding

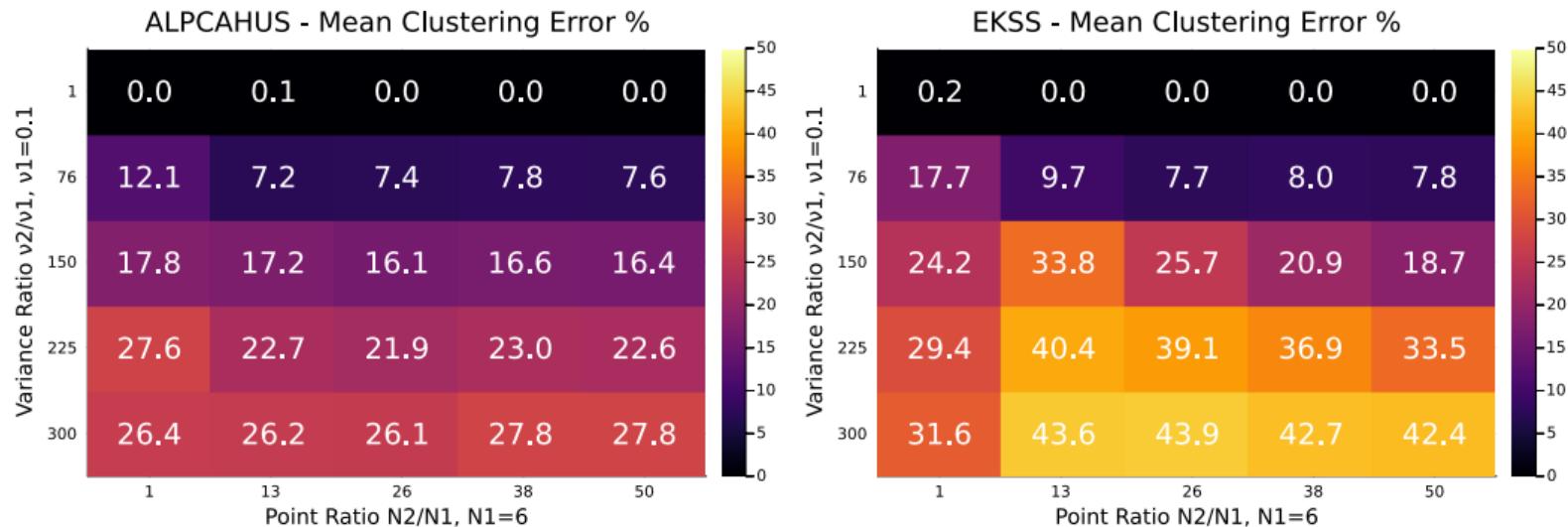


$\text{EKSS}(Y) \rightarrow A \rightarrow D_{ii} = \sum_j A_{ij} \rightarrow L = I - D^{-1}A \rightarrow \text{EVD}(L) \rightarrow V'_K \rightarrow \text{kmeans}(V'_K)$

Short Summary

- ① Run EKSS and form affinity matrix A
- ② Threshold matrix to reduce false connections (not discussed)
- ③ Perform spectral clustering on A to get final clusters
- ④ Perform PCA on each cluster to get subspaces

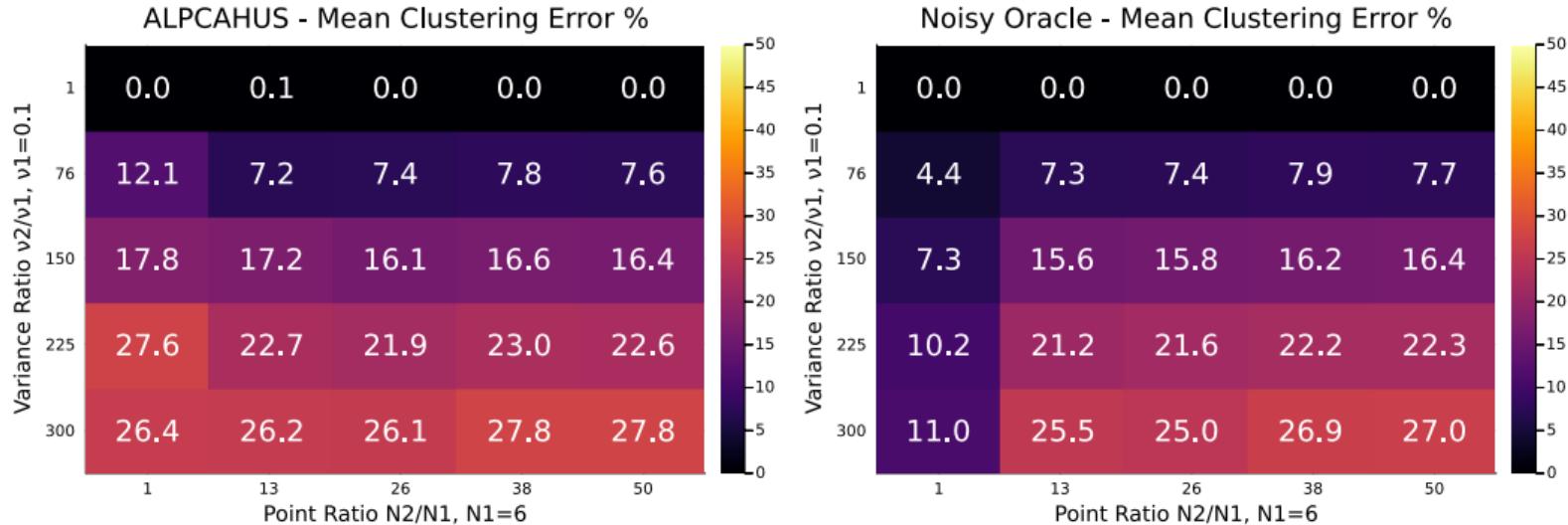
ALPCA HUS vs. EKSS - Result I



we are able to better learn the subspaces!

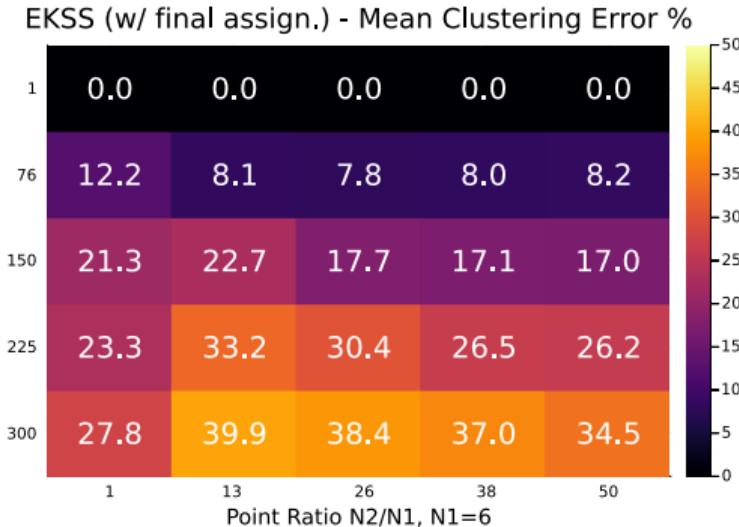
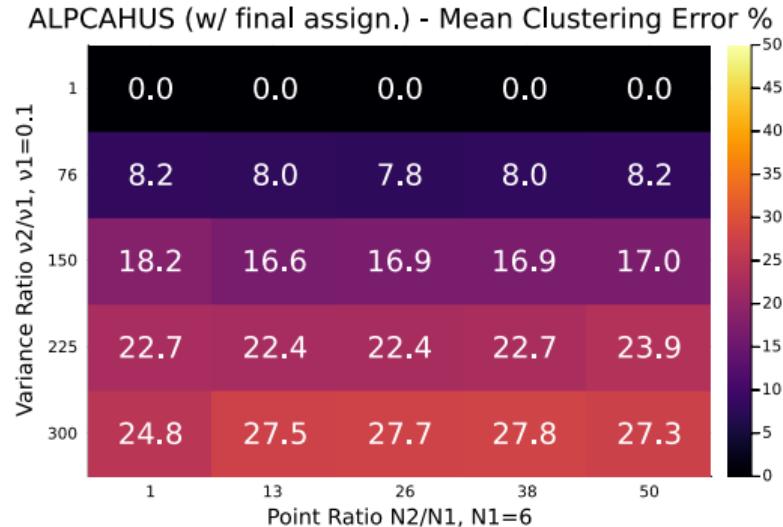
javier: discuss homoscedastic setting

ALPCA HUS vs. EKSS - Result II



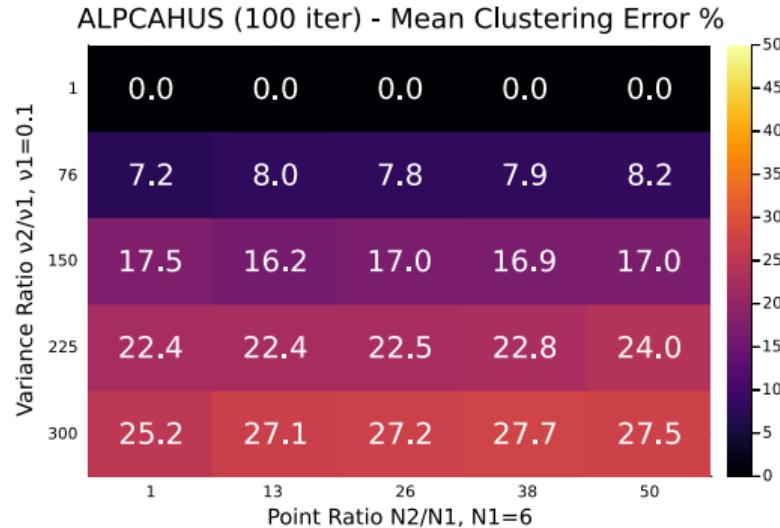
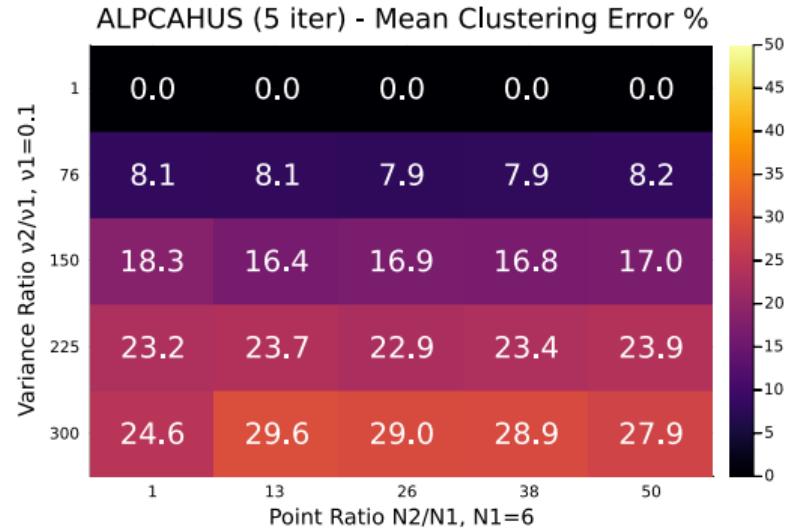
now we are much closer!

ALPCA After EKSS?



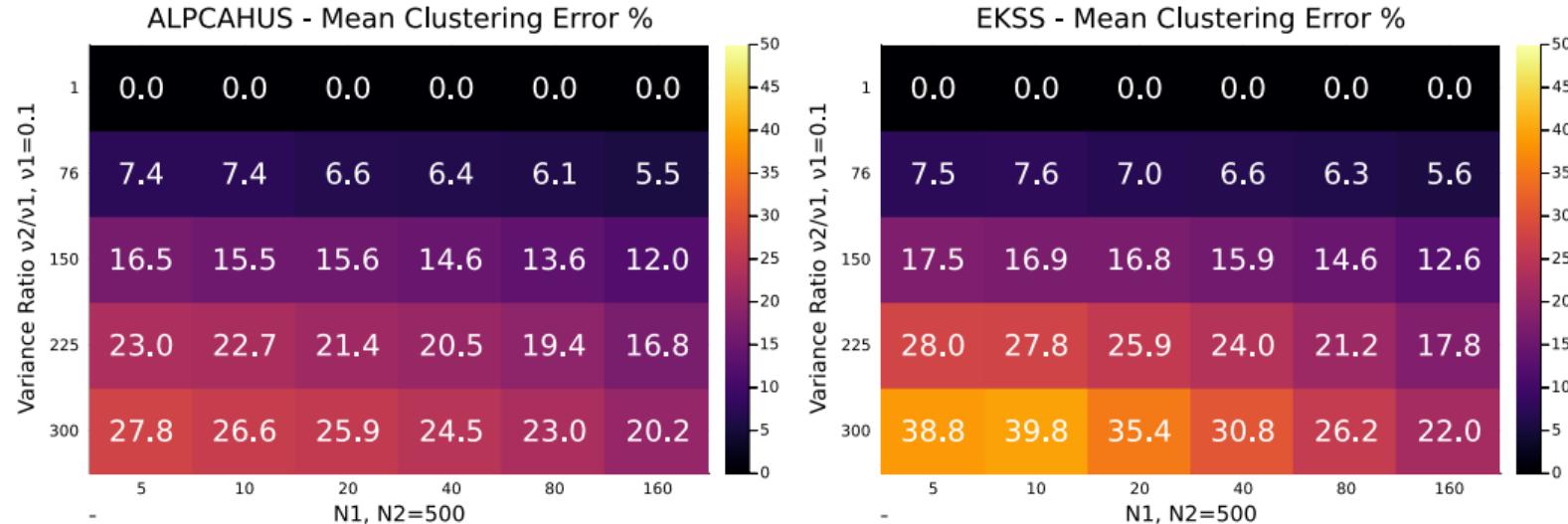
EKSS final cluster is still too noisy!

ALPCA Iterations



not many iterations needed for ALPCA!

Good Data Experiment



works for a “large” range of good data!

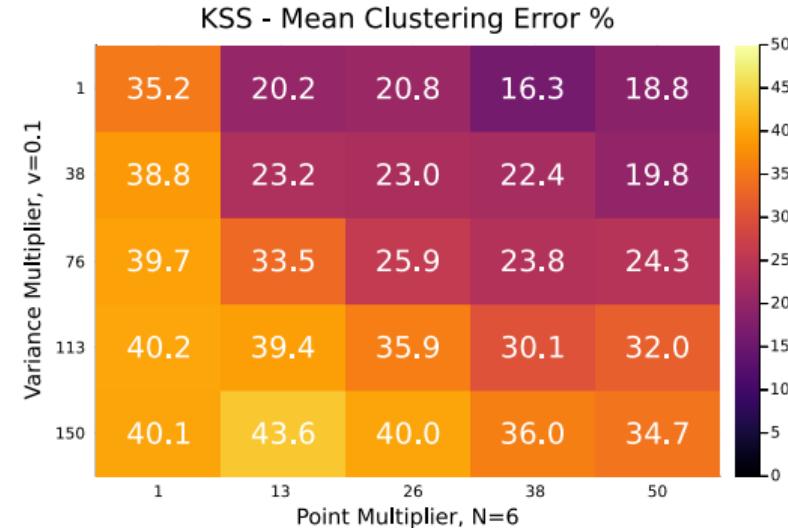
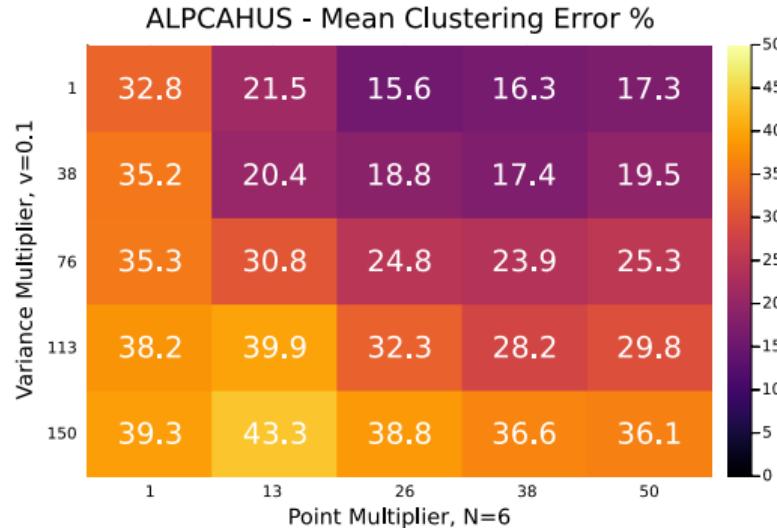
Conclusion

- ① ALPCA HUS appears beneficial in homoscedastic setting (**extra slide**)
- ② Heteroscedasticity is problematic for various research areas like SC
- ③ Both the clusterings and subspace estimations are thrown off
- ④ Without knowing noise variances, we can account for this kind of data
- ⑤ ALPCA H + KSS can improve the clustering/subspaces estimates
- ⑥ No need to solve ALPCA H exactly, few iterations are enough

Future Work

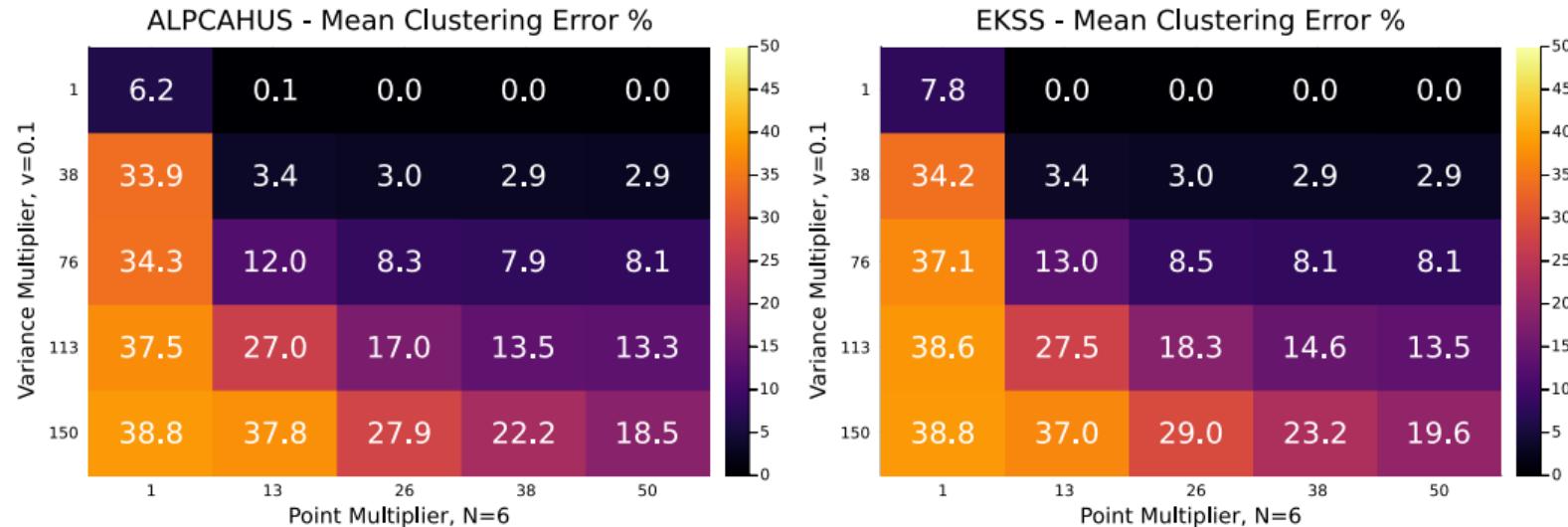
- ① This is very early work! (results from past few weeks)
- ② Real data examples? COIL-100? (homoscedastic but that might be ok)
- ③ Laura gave me an idea for estimating subspace dimensions
- ④ More comparisons with other algorithms like SSC,TSC, ...
- ⑤ Thanks for listening! Suggestions? Comments?

Homoscedastic Setting (KSS)



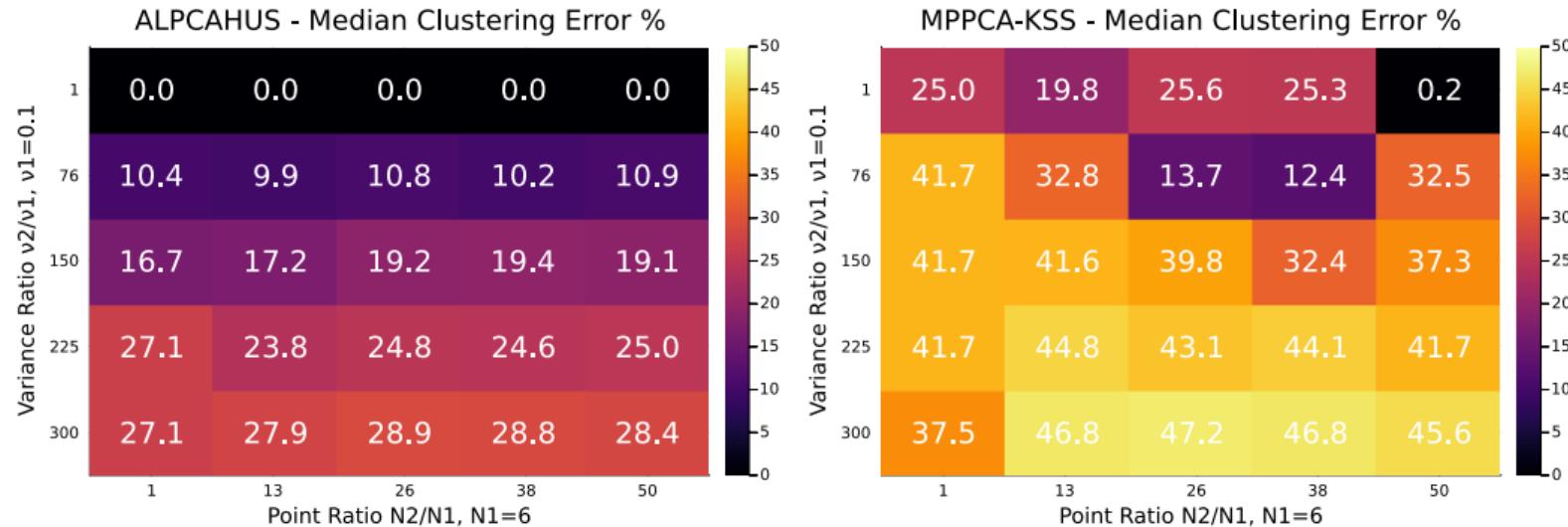
ALPCA HUS can be beneficial in homoscedastic setting

Homoscedastic Setting (Ensemble KSS)



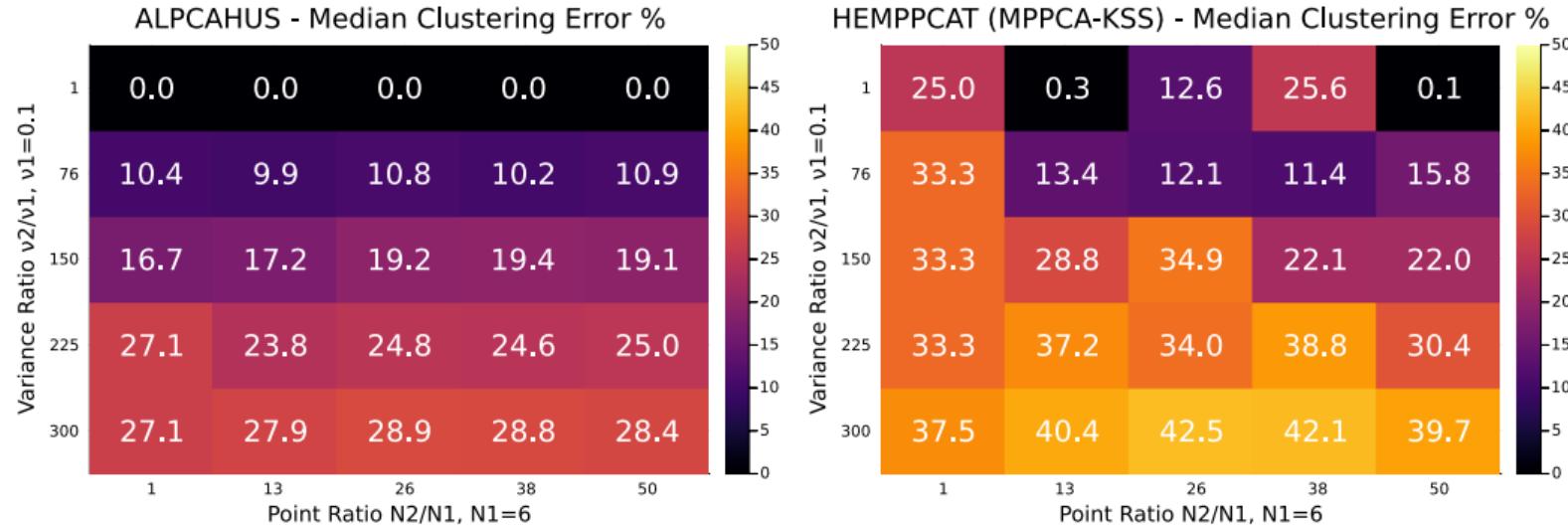
No longer true for the ensemble methods

MPPCA [9]



MPPCA assumes each mixture has the same variance

HEMPPCAT [10]



ALPCA-HUS outperforms heteroscedastic MPPCA

References

- [1] J. S. Cavazos, L. Balzano, and J. A. Fessler, *Sample-wise heteroscedastic pca with tail singular value regularization*, Submitted to SampTA, 2023.
- [2] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, “Clustering appearances of objects under varying illumination conditions,” in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, IEEE, vol. 1, 2003, pp. I–I.
- [3] W. Hong, J. Wright, K. Huang, and Y. Ma, “Multiscale hybrid linear models for lossy image representation,” *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3655–3671, 2006.
- [4] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, “Unsupervised segmentation of natural images via lossy data compression,” *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

References

- [5] R. Vidal, R. Tron, and R. Hartley, “Multiframe motion segmentation with missing data using powerfactorization and gPCA,” *International Journal of Computer Vision*, vol. 79, no. 1, pp. 85–105, 2008.
- [6] R. Vidal, “Subspace clustering,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.
- [7] J. Lipor, D. Hong, Y. S. Tan, and L. Balzano, “Subspace clustering using ensembles of k-subspaces,” *Information and Inference: A Journal of the IMA*, vol. 10, no. 1, pp. 73–107, 2021.
- [8] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [9] M. E. Tipping and C. M. Bishop, “Mixtures of probabilistic principal component analyzers,” *Neural computation*, vol. 11, no. 2, pp. 443–482, 1999.

References

- [10] A. S. Xu, L. Balzano, and J. A. Fessler, *Hemppcat: Mixtures of probabilistic principal component analysers for data with heteroscedastic noise*, 2023.
arXiv: 2301.08852 [stat.ME].