Data Science Week 2023 (Purdue University Fort Wayne)

# Fast, memory-efficient spectral clustering with cosine similarity

Guangliang Chen (cheng@hope.edu)
Hope College, Holland, Michigan, USA

## Introduction

Spectral clustering is a modern, powerful clustering approach. It uses the eigenvectors of a normalized graph Laplacian for embedding the data into a low-dimensional space for easy clustering. However, it is well known to face two major challenges:

- scalability (speed and memory),
- out of sample extension.

We present a memory and speed efficient spectral clustering algorithm in the setting of *cosine similarity* that only uses the following efficient linear algebra operations:

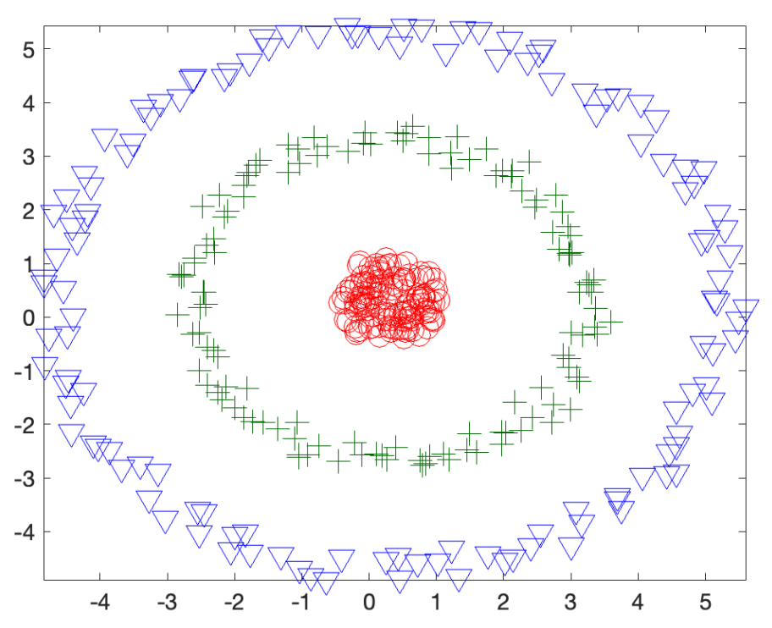
- elementwise manipulation,
- matrix-vector multiplication, and
- low-rank SVD.

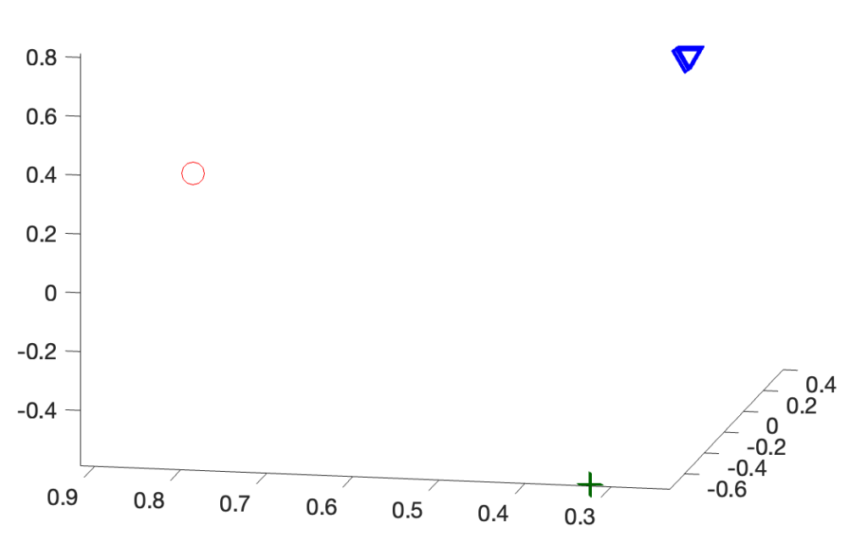
## What is spectral clustering?

There are different versions of spectral clustering; here we present the formulation by Ng, Jordan and Weiss (2001).

```
Input: Data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, number of clusters $k$, scale parameter $\sigma$
Output: Clusters $C_1, \dots, C_k$
1: Construct a pairwise similarities matrix
       $\mathbf{W} = (w_{ij}), \quad w_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}, \; i \neq j$
2: Find the row sums of $\mathbf{W}$ and use them to define a diagonal matrix $\mathbf{D} = \mathrm{diag}(\mathbf{W}\mathbf{1})$. Let $\widetilde{\mathbf{W}} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.
3: Find the $k$ largest eigenvectors of $\widetilde{\mathbf{W}}$ and form an embedding matrix
       $\mathbf{X} \;\mapsto\; \mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_k] \in \mathbb{R}^{n\times k}$.
4: Apply $k$-means to group the rows of $\mathbf{V}$ into $k$ clusters.
```

## Demonstrations





## Computational challenges

- **Memory requirement**: $\mathcal{O}(n^2)$
- **Computational cost**:
  - Construction of $\mathbf{W}$: $\mathcal{O}(n^2 d)$
  - Decomposition of $\mathbf{W}$: $\mathcal{O}(n^3)$

| Data sets | $n$ | $p$ | $k$ |
|---|---|---|---|
| usps | 9,298 | 256 | 10 |
| pendigit | 10,992 | 16 | 10 |
| mnist | 70,000 | 184 | 10 |
| 20news | 18,768 | 55,570 | 20 |
| protein | 24,387 | 357 | 3 |
| covtype | 581,012 | 54 | 7 |

## Speed scalability (ICPR18′, PRL 19′)

Given data $\mathbf{X} \in \mathbb{R}^{n\times d}$ with $L_2$-normalized rows, the cosine similarity matrix is

$$\mathbf{W} = \mathbf{X}\mathbf{X}^T - \mathbf{I}.$$

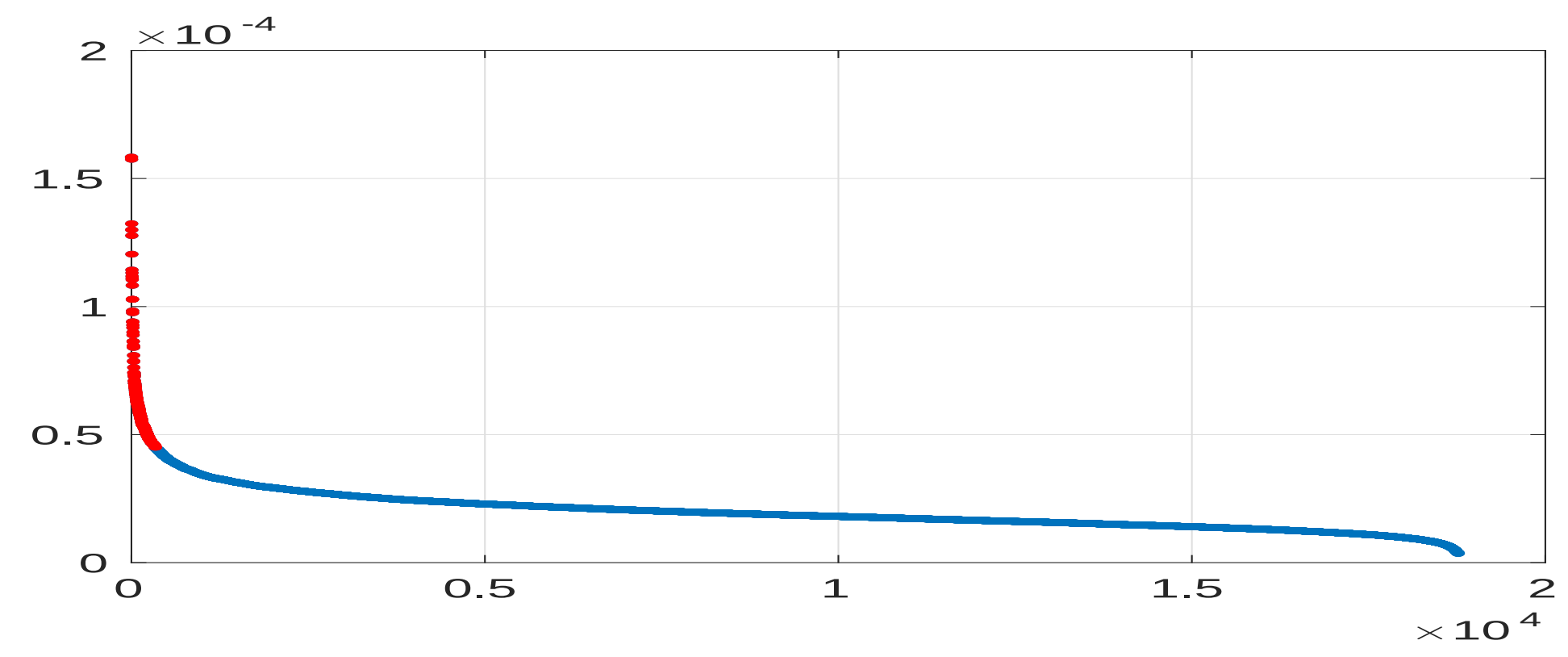
First, we can compute $\mathbf{D}$ directly from $\mathbf{X}$:

$$\mathbf{D} = \mathrm{diag}((\mathbf{X}\mathbf{X}^T - \mathbf{I})\mathbf{1}) = \mathrm{diag}(\mathbf{X}(\mathbf{X}^T\mathbf{1}) - \mathbf{1}).$$

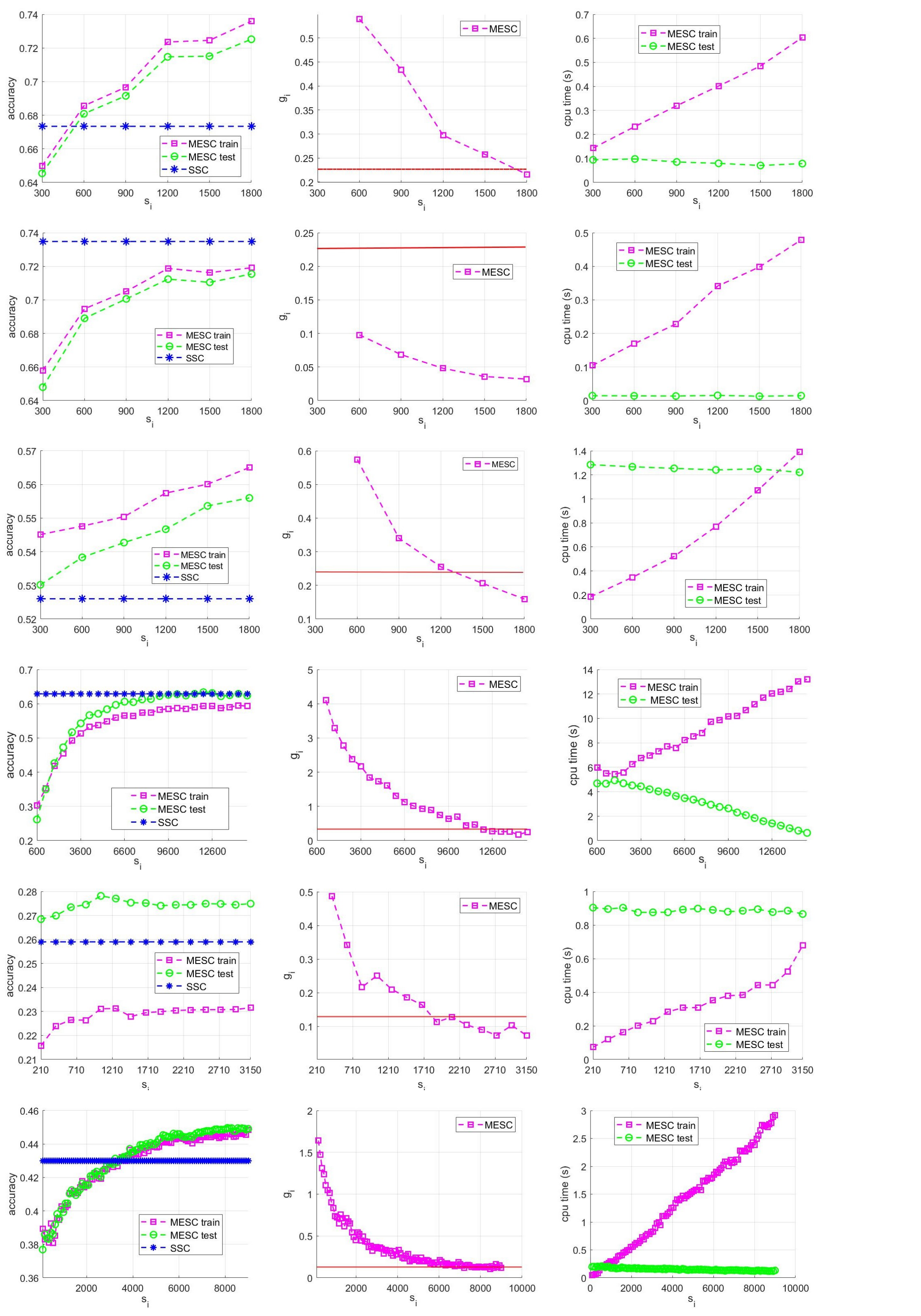
Next, we write

$$\widetilde{\mathbf{W}} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T - \mathbf{D}^{-1}, \quad \widetilde{\mathbf{X}} = \mathbf{D}^{-1/2}\mathbf{X}.$$

Finally, after removing a small fraction ($\alpha$) of low-degree points (in order to make $\mathbf{D}^{-1}$ nearly constant diagonal), we use the left singular vectors of $\widetilde{\mathbf{X}}$ to approximate the eigenvectors $\widetilde{\mathbf{U}}$ of $\widetilde{\mathbf{W}}$.



## Results (memory scalability)



## Memory scalability (CIARP 2023, to appear)

**Single batch learning.** Assume a small batch of data of size $s \ll n$, denoted $\mathbf{X}_s \in \mathbb{R}^{s\times d}$, that has become available through sampling. We estimate the right singular vectors of $\widetilde{\mathbf{X}}$ as follows:

$$\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} = \mathbf{X}^T\mathbf{D}^{-1}\mathbf{X} = \sum_{i=1}^{n} \frac{1}{d_i}\mathbf{x}_i\mathbf{x}_i^T \approx \frac{n}{s}\sum_{i=1}^{s}\frac{1}{d_i}\mathbf{x}_i\mathbf{x}_i^T = \frac{n}{s}\widetilde{\mathbf{X}}_s^T\widetilde{\mathbf{X}}_s,$$

where $\widetilde{\mathbf{X}}_s$ and $\mathbf{D}_s$ represent the restrictions of $\widetilde{\mathbf{X}}$ and $\mathbf{D}$ to the sample $\mathbf{X}_s$, respectively:

$$\widetilde{\mathbf{X}}_s = \mathbf{D}_s^{-1/2}\mathbf{X}_s, \quad \mathbf{D}_s = \mathrm{diag}(\mathbf{d}_s), \quad \mathbf{d}_s = \mathbf{X}_s \cdot \sum_{i=1}^{n}\mathbf{x}_i - \mathbf{1}_s \approx \frac{n}{s}\mathbf{X}_s(\mathbf{X}_s^T\mathbf{1}_s) - \mathbf{1}_s.$$

Letting the rank-$k$ SVD of $\widetilde{\mathbf{X}}_s$ be $\widetilde{\mathbf{X}}_s \approx \widetilde{\mathbf{U}}_s\widetilde{\mathbf{\Sigma}}_s\widetilde{\mathbf{V}}_s^T$, we have $\widetilde{\mathbf{V}}\widetilde{\mathbf{V}}^T \approx \widetilde{\mathbf{V}}_s\widetilde{\mathbf{V}}_s^T$ and $\widetilde{\mathbf{\Sigma}} \approx \sqrt{\frac{n}{s}}\widetilde{\mathbf{\Sigma}}_s$. Therefore, the nonlinear embedding of the batch $\mathbf{X}_s \in \mathbb{R}^{s\times d}$ is

$$\mathbf{Y}_s := \widetilde{\mathbf{X}}_s\widetilde{\mathbf{V}}\widetilde{\mathbf{\Sigma}}^{-1} \approx \widetilde{\mathbf{X}}_s\widetilde{\mathbf{V}}_s\left(\sqrt{\frac{n}{s}}\widetilde{\mathbf{\Sigma}}_s\right)^{-1} = \sqrt{\frac{s}{n}}\widetilde{\mathbf{X}}_s\widetilde{\mathbf{V}}_s\widetilde{\mathbf{\Sigma}}_s^{-1} \in \mathbb{R}^{s\times k}.$$

**How to choose** $s$. Apply the above single-batch learning procedure repeatedly and separately on a collection of nested batches of increasing sizes $\{\mathbf{X}_{s_i}\}_{i\geq 0}$ and focus on the convergence of the outputs $\widetilde{\mathbf{V}}_{s_i}$ under the Grassmannian metric:

$$g_i = \left\|\widetilde{\mathbf{V}}_{s_i}\widetilde{\mathbf{V}}_{s_i}^T - \widetilde{\mathbf{V}}_{s_{i-1}}\widetilde{\mathbf{V}}_{s_{i-1}}^T\right\|_F = \sqrt{2k - 2\left\|\widetilde{\mathbf{V}}_{s_i}^T\widetilde{\mathbf{V}}_{s_{i-1}}\right\|_F^2} = \sqrt{2\sum_{j=1}^{k}\sin^2\theta_{ij}}, \quad i = 1, 2, \dots$$

where $0 \leq \theta_{i1} \leq \dots \leq \theta_{ik} \leq \frac{\pi}{2}$ are the principal angles between the column spaces of $\widetilde{\mathbf{V}}_{s_i}$ and $\widetilde{\mathbf{V}}_{s_{i-1}}$. Empirically, we set $s = s_i$ such that all $\theta_{ij} \leq \theta_0$, i.e., $g_i < \sqrt{2\cdot k\cdot \sin^2\theta_0} = \sqrt{2k}\sin\theta_0$.

**Out of sample extension.** Any new point, say $\mathbf{x}_0 \in \mathbb{R}^d$ is embedded as follows:

$$\mathbf{y}_0 = \sqrt{\frac{s}{n}}\left(d_0^{-1/2}\mathbf{x}_0^T\right)\widetilde{\mathbf{V}}_s\widetilde{\mathbf{\Sigma}}_s^{-1} \in \mathbb{R}^k, \quad d_0 = \mathbf{x}_0^T\sum_{i=1}^{n}\mathbf{x}_i - 1 \approx \frac{n}{s}\mathbf{x}_0^T(\mathbf{X}_s^T\mathbf{1}_s) - 1.$$

## Conclusions

We presented some recent and ongoing work on the speed and memory scalability of spectral clustering with cosine similarity. Preliminary results demonstrate their effectiveness.