Big-data Clustering: K-means vs K-indicators

Yin Zhang

Dept. of Computational & Applied Math.
Rice University, Houston, Texas, U.S.A.

Joint work with
Feiyu Chen & Taiping Zhang (CQU), Liwei Xu (UESTC)

2016 International Workshop on
Mathematical Issues in Information Sciences
Dec. 20, 2016. Shenzhen, China

Outline

Outline

Partition data points into clusters

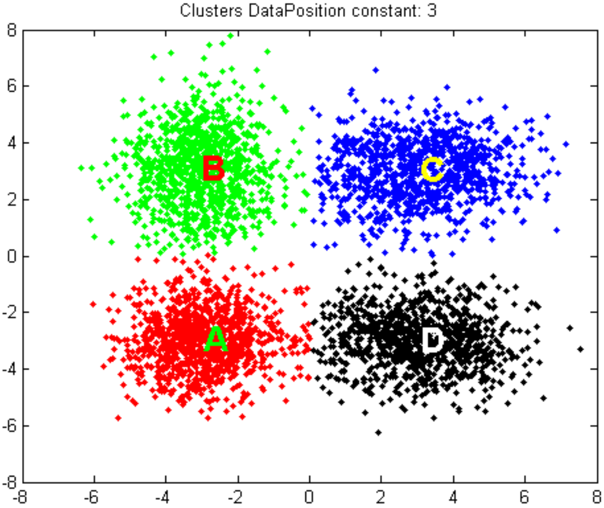


Figure: Group n data points into $k = 4$ clusters ($n \gg k$)

Non-globular and globular clusters

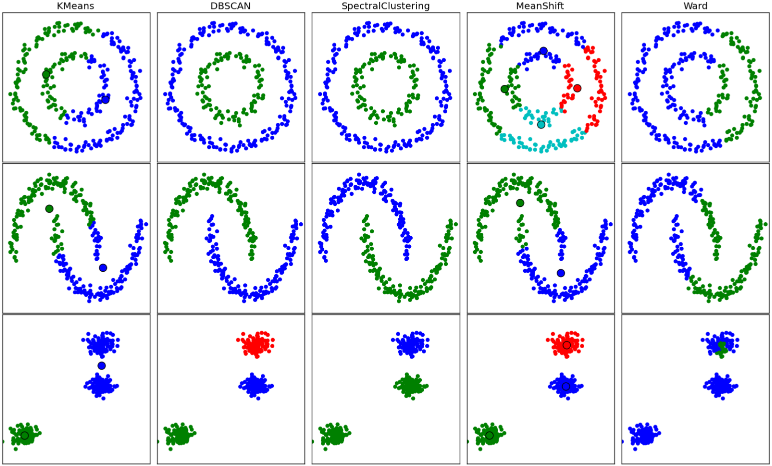


Figure: Rows 1–2: non-globular. Row 3: Globular

Clustering Images

Figure: Group images into 2 clusters (cats and dogs)

Clustering Images II

Figure: Group faces by individuals

Formal Description:

Input:

- data objects $m_i \in \mathbb{R}^d$, $i = 1, 2, \cdots, n$
- (estimated) number of clusters $k < n$
- a similarity measure ("distance")

Output:

- Assignments to k clusters:
  - **(a)** within clusters objects are more similar to each other
  - **(b)** between clusters objects are less similar to each other

A fundamental technique in unsupervised learning

Stage 1: Pre-processing data

Transform data to "latent features"

- dependent on definition of "similarity"
- various data-specific techniques existing
- e.g.: PCA clustering, spectral clustering[1]

Stage 2: Clustering "latent features"

Method of choice: **K-means** (model[2] + algorithm[3])

Our focus is on Stage 2

[1] See von Luxburg, A tutorial on spectral clustering, 2007
[2] MacQueen, Some Methods for classification & Analysis of Multivariate Observations, 1967
[3] Lloyd, Least square quantization in PCM, 1957.

Pre-processing

Spectral Clustering[4]: Given a similarity measure, do:

1. Construct a similarity graph (many options, e.g. KNN)
2. Compute k leading eigenvectors of graph Laplacian
3. Cluster the rows via K-means (i.e., Lloyd algorithm)

Dimension Reduction:

- Principle Component Analysis (PCA)
- Non-negative Matrix Factorization (NMF)
- Random Projection (PR)

Kernel Tricks: nonlinear change of spaces

Afterwards, K-means is applied to do clustering

[4] See von Luxburg, A Tutorial on Spectral Clustering, 2007 for references.

Outline

K-means Model

Given $\{m_i\}_{i=1}^n \in \mathbb{R}^d$ and $k > 0$, the default K-means model is

$$\min_{x_j} \sum_{i=1}^{n} \min \left\{ \|m_i - x_j\|^2 \mid j = 1, \cdots, k \right\}$$

where j-th centroid x_j = the mean of the points in j-th cluster.

K-means model

- searches for k centroids in data space
- is essentially discrete and generally NP-hard
- is suitable for globular clusters (2-norm)

Lloyd Algorithm[5] (or Lloyd-Forgy):

Select k points as the initial centroids;
while "not converged" **do**
  (1) Assign each point to the closest centroid;
  (2) Recompute the centroid for each cluster.
end while

Algorithm

- converges to a local minimum at $O(dkn)$ per iter.
- is sensitive to initialization due to greedy nature
- requires (costly) multiple restarts for consistency

[5] Lloyd, Least square quantization in PCM, 1957.

K-means++[6]:

1. Randomly select a data point in M as the first centroid.
2. Randomly select a new centroid in M so that it is probabilistically far away from the existing centroids.
3. Repeat step 2 until k centroids are selected.

Advantages:

- Achieving $O(\log k) * (\text{optimal value})$ is expected
- much better than uniformly random in practice

K-means Algorithm = K-means++/Lloyd: **method of choice**

[6] Arthur and Vassilvitskii, K-means++: The advantages of careful seeding, 2007.

Convex Relaxation Models

- LP Relaxation [7] $\Longrightarrow$ $O\left(n^2\right)$ variables
- SDP Relaxation [8] $\Longrightarrow$ $O\left(n^2\right)$ variables
- Sum of Norm Relaxation: [9] $\Longrightarrow$ $O\left(n^2\right)$ evaluation cost

$$\min_{x_i} \sum_{i=1}^{n} \|m_i - x_i\|^2 + \gamma \sum_{i<j} \|x_i - x_j\| \quad \leftarrow \text{sparsity promoting}$$

Due to high computational cost, fully convexified models/methods have not shaken the dominance of K-means

Need to keep the same $O(n)$ per-iteration cost as K-means

[7] Awasthi et al., Relax, no need to round: Integrality of clustering formulations, 2015

[8] Peng and Xia, Approximating k-means-type clustering via semidefinite programming, 2007

[9] Lindsten et al., Just Relax and Come Clustering! A Convexication of k-Means..., 2011

A Revealing Example

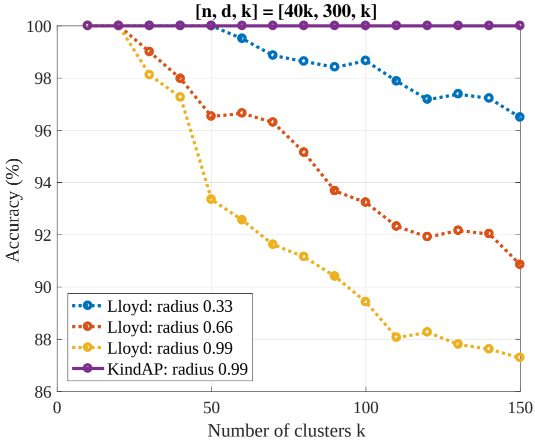
Synthetic Data:

- Generate k centers with mutual distance = 2
- Create a cloud around each center within radius < 1

Globular, perfectly separated, ground truth known

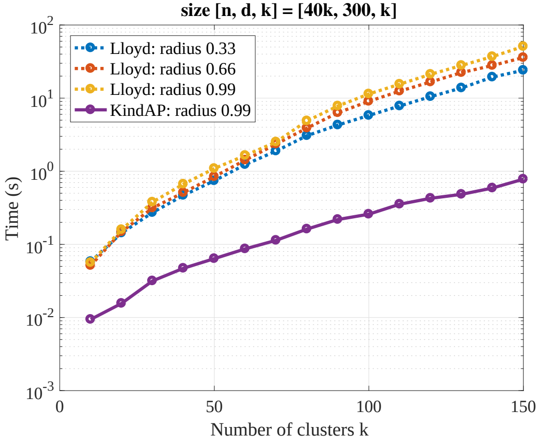
Experiment:

- Pre-processing: k principal components
- Clustering: apply K-means and KindAP (new)
- k increases from 10 to 150



Such behavior hinders K-means in big-data applications

Speed



KindAP runs faster than Lloyd (1 run) with GPU acceleration

Outline

Ideal Data

Ideal K-rays data contains n points lying on k rays in $\mathbb{R}^d$.

After permutation:

$$\hat{M} = \underbrace{\begin{pmatrix} h_1 p_1^T \\ h_2 p_2^T \\ \vdots \\ h_k p_k^T \end{pmatrix}}_{n \times d} = \underbrace{\begin{pmatrix} h_1 & & & \\ & h_2 & & \\ & & \ddots & \\ & & & h_k \end{pmatrix}}_{n \times k} \underbrace{\begin{pmatrix} p_1^T \\ p_2^T \\ \vdots \\ p_k^T \end{pmatrix}}_{k \times d} \triangleq \hat{H} P^T$$

- $p_1, \cdots, p_k \in \mathbb{R}^d$ are ray vectors.
- $h_i \in \mathbb{R}^{n_i}$ are positive vectors.

Each data vector ($\hat{M}$ row) is a positive multiple of a ray vector.

Ideal K-points data are special cases of ideal K-rays data.

k Indicators

Indicator matrix $\hat{H} \geq 0$ contains k indicators (orth. columns)

$$\left[\hat{H}\right]_{ij} > 0 \iff \text{Point } i \text{ is on Ray } j.$$

The j-th column of H is an indicator for Cluster j.

E.g., $\hat{H} \in \mathbb{R}^{6\times 2}$ consists of two indicators

$$\hat{H}e_1 = \begin{pmatrix} 3 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \hat{H}e_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

Points 1-3 are in Cluster 1, points 4-6 (identical) in Cluster 2.

For ideal K-rays data, range spaces of $\hat{M}$ and $\hat{H}$ are the same:

$$\mathcal{R}(\hat{M}) = \mathcal{R}(\hat{H})$$

Consider data $M = \hat{M} + E$ with small perturbation E,

$$\mathcal{R}(U_{[k]}) \approx \mathcal{R}(M) \approx \mathcal{R}(\hat{M}) = \mathcal{R}(\hat{H}),$$

where $U_{[k]} \in \mathbb{R}^{n\times k}$ contains k leading left singular vectors of M.

The approximations become exact as E vanishes.

Two Choices of subspace distance

Minimizing the distance between 2 subspaces:

K-means' choice:

$$\mathrm{dist}(\mathcal{R}(U_{[k]}), \mathcal{R}(H)) = \|(I - P_{\mathcal{R}(H)})\, U_{[k]}\|_F$$

where $P_{\mathcal{R}(H)}$ is the orthogonal projection onto $\mathcal{R}(H)$.

Our Choice:

$$\mathrm{dist}(\mathcal{R}(U_{[k]}), \mathcal{R}(H)) = \min_{U \in \mathcal{U}_o} \|U - H\|_F$$

where $\mathcal{U}_o$ is the set of all the orthonormal bases for $\mathcal{R}(U_{[k]})$.

K-means model (a matrix version[10]):

$$\min_{H} \|(I - HH^T)\, U_{[k]}\|_F^2, \;\; \text{s.t.} \;\; H \in \mathcal{H} \cap \ldots$$

where the objective is highly non-convex in H.

K-indicators model:

$$\min_{U,H} \;\|U - H\|_F^2, \;\; \text{s.t.} \;\; U \in \mathcal{U}_o, \; H \in \mathcal{H}$$

where the objective is convex, and

$$\mathcal{U}_o = \left\{ U_{[k]} Z \in \mathbb{R}^{n \times k} : Z^T Z = I \right\}, \; \mathcal{H} = \left\{ H \in \mathbb{R}_+^{n \times k} : H^T H = I \right\}$$

K-indicators model looks more tractable

[10] Boutsidis et al., Unsupervised Feature Selection for k-means Clustering Prob., *NIPS*, 2009

K-indicators Model is Geometric

$$\min_{U,H} \|U - H\|_F^2, \text{ s.t. } U \in \mathcal{U}_o, \ H \in \mathcal{H} \tag{1}$$

is to find the distance between 2 nonconvex sets $\mathcal{U}_o$ and $\mathcal{H}$.

- $\mathcal{H}$ is nasty, permitting no easy projection
- $\mathcal{U}_o$ is milder, easily projectable

Projection onto $\mathcal{U}_o$:

$$\begin{aligned} U_{[k]}^T X &= PDQ^T \quad (k \text{ by } k \text{ SVD}) \\ P_{\mathcal{U}_o}(X) &= U_{[k]}(PQ^T). \end{aligned}$$

An intermediate problem:

$$\min \|U - N\|_F^2, \quad \text{s.t.} \quad U \in \mathcal{U}_o, \ N \in \mathcal{N} \tag{2}$$

where $\mathcal{N} = \{N \in \mathbb{R}^{n\times k} | \ N \geq 0\}$ is a closed convex set.

Reasons:

- $\mathcal{N}$ is convex, $P_{\mathcal{N}}(X) = \max(0, X)$.
- The boundary of $\mathcal{N}$ contains $\mathcal{H}$.
- Local minimum of (2) can be solved by alternating projection.

2-level alternating "projections" algorithmic scheme:

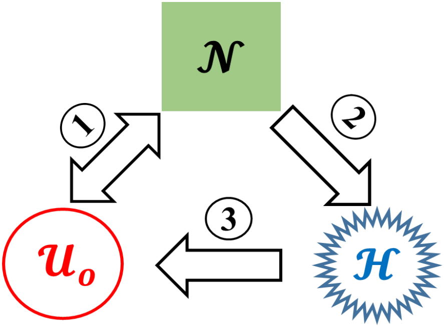


Figure: (1) alternating projection (2) "proj-like" operator (3) projection

Magnitude of N_{ij} reflects likelihood of point i in cluster j.

Let $\hat{N}$ hold the elements of N sorted row-wise in descending order.

Define *soft indicator vector:*

$$s_i = 1 - \hat{N}_{i2}/\hat{N}_{i1} \in [0, 1], \quad i = 1, ..., n,$$

Intuition:

- The closer is s_i to 0, the more uncertainty there is in assignment.
- The closer is s_i to 1, the less uncertainty there is in assignment.

Input: $M \in R^{n \times d}$ and integer $k \in (0, \min(d, n)]$
  Compute k leading left singular vectors of M to form U_k
  Set $U = U_k \in \mathcal{U}_o$
  while not converged **do**
    Starting from U, find a minimizing pair
      $(U, N) \leftarrow \text{argmin}\, dist(\mathcal{U}_o, \mathcal{N})$.
    Find an $H \in \mathcal{H}$ close to N.
    $U \leftarrow P_{\mathcal{U}_o}(H)$.
  end while
Output: $U \in \mathcal{U}_o$, $N \in \mathcal{N}$, $H \in \mathcal{H}$.

Outline

A typical run on synthetic data

```
[d, n, k] = [500, 10000, 100]

***** radius 1.00 *****

Outer 1:  43 dUH: 5.08712480e+00 idxchg:  9968
Outer 2:  13 dUH: 3.02146591e+00 idxchg:  866
Outer 3:   2 dUH: 3.02144864e+00 idxchg:  0

KindAP 100.00% Elapsed time is 1.923610 seconds
Kmeans  88.66% Elapsed time is 4.920525 seconds

***** radius 2.00 *****

Outer 1:  42 dUH: 5.98082240e+00 idxchg:  9900
Outer 2:   3 dUH: 5.55739995e+00 idxchg:  250
Outer 3:   2 dUH: 5.55442009e+00 idxchg:  110
Outer 4:   2 dUH: 5.55442000e+00 idxchg:  0

KindAP 100.00% Elapsed time is 1.797684 seconds
Kmeans  82.73% Elapsed time is 7.909552 seconds
```

Example 1: Three tangential circular clusters in 2-D

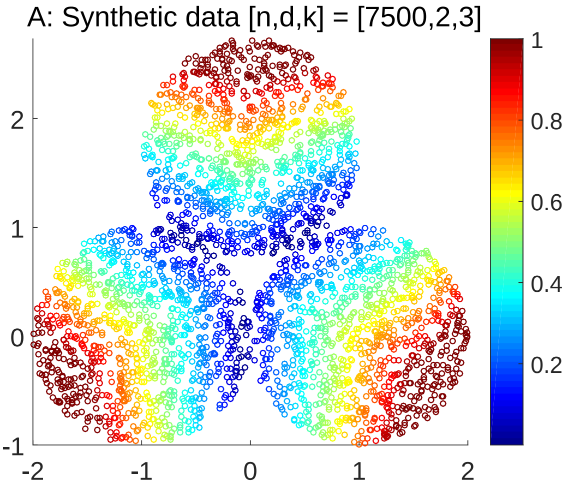


Figure: Points colored according to soft indicator values

Example 2: Groups of 40 faces from 4 persons (ORL data)

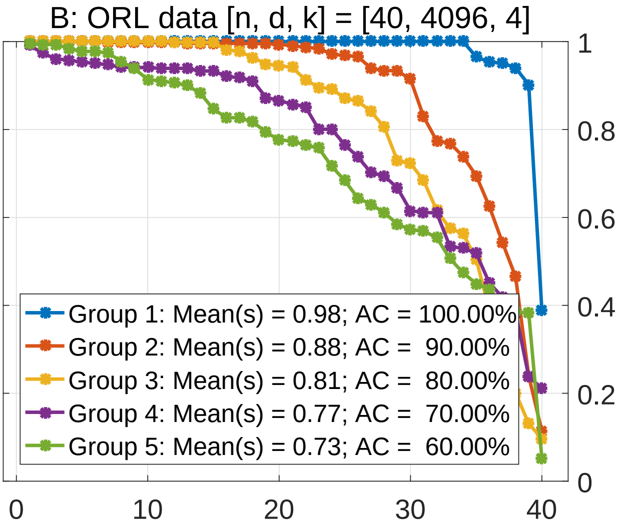


Figure: Five soft indicators sorted in a descending order

K-rays data Example

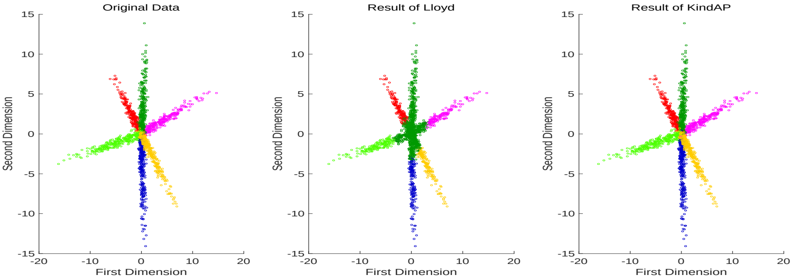


Figure: Points are colored according to their clusters

Kernel data Example

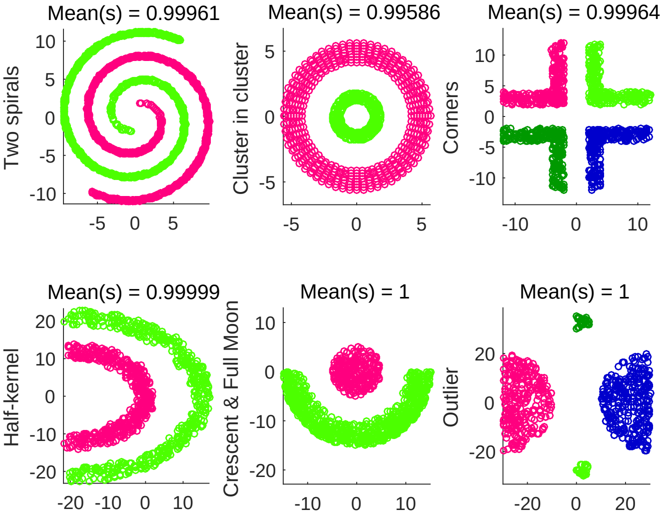


Figure: Non-globular clusters in 2-D

Real Examples: two popular datasets with “large” k

COIL image dataset

- $k = 100$ objects, each having 72 different images
- $n = 7200$ images
- $d = 1024$ pixels (image size: 32×32)

TDT2 document dataset

- $k = 96$ different usenet newsgroups
- $n = 10212$ documents
- $d = 36771$ words

Pre-processing:

KNN + Normalized Laplacian + Spectral Clustering

3 Algorithms Compared

- KindAP
- KindAP+ L: Run 1 Lloyd starting from KindAP centers
- Lloyd10000: K-means with 10000 replications

K-Means Code

- **kmeans** in *Matlab Statistics and Machine Learning Toolbox* (R2016a with GPU acceleration)

Computer

A desktop with Intel Core i7-3770 CPU at 3.4GHz/8GB RAM

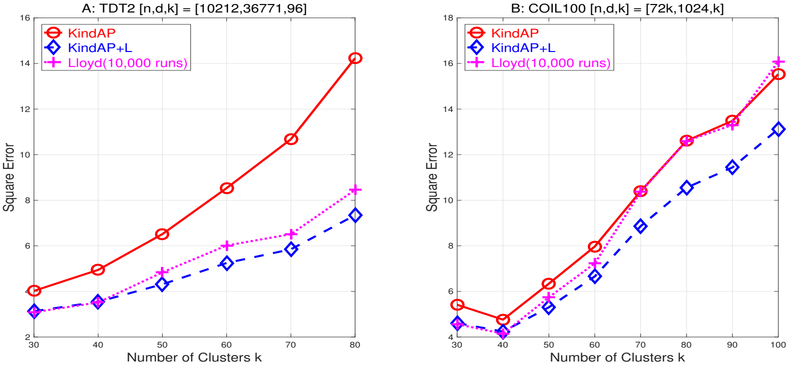
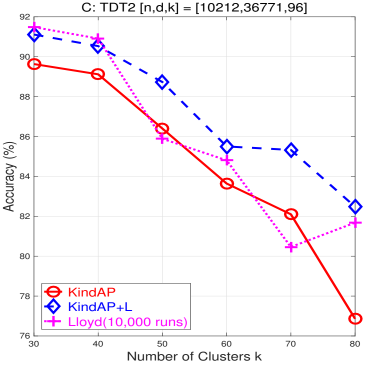


Figure: A: KindAP+L best B: KindAP+L best

Clustering Accuracy



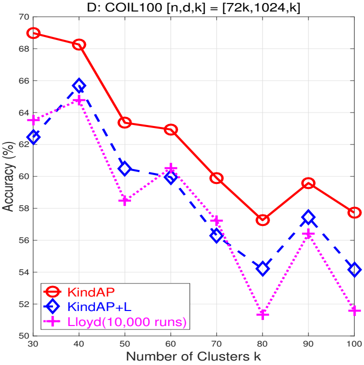
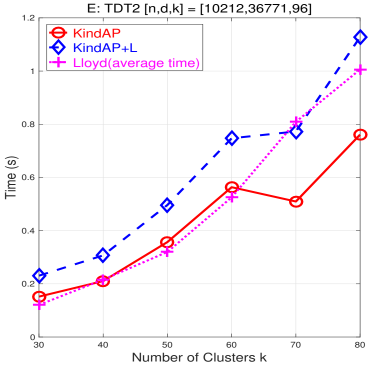


Figure: A: KindAP+L best B: KindAP best

K-means (K-indicators) model fits TDT2 (COIL100) better

Running Time (average time for Lloyd)



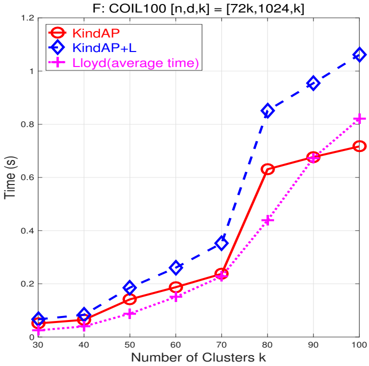


Figure: 1 KindAP run $\approx$ 1 Lloyd run

Visualization on Yale Face Dataset: $n = 165$, $k = 16$ ($k^* = 15$)

$M \approx M_k = U_k \Sigma_k V_k^T \qquad U_k \rightarrow$ **KindAP** $\rightarrow (U, N, H) \qquad W = M_k^T U$

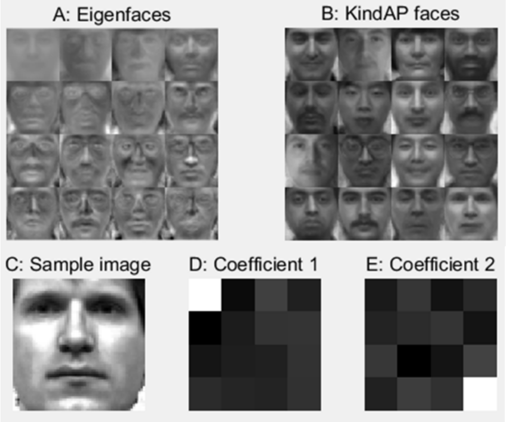


Figure: A: V_k. B: W. C-E: face $\approx V_k c_1 = W c_2$

c_1: a row of $\Sigma_k U_k \qquad c_2$: a row of U ($\rightarrow$ face in cluster 16)

Summary

Property	K-indicators	K-means
parameter-free	yes	yes
$O(n)$ cost per iter.	yes	yes
non-greedy	yes	no
not need replications	yes	no
suitable for big-k data	yes	no
posterior info available	yes	no

Contribution:

- enhanced infrastructure for unsupervised learning

Further Work:

- global optimality under favorable conditions (already proven for ideal data)

Thank you!