# Theoretical Advances in Clustering with Applications to Nonnegative Matrix Factorization

Vincent Y. F. Tan
(Joint work with Zhaoqiang Liu)

Department of Mathematics, NUS

November 10, 2017

NUS
National University
of Singapore

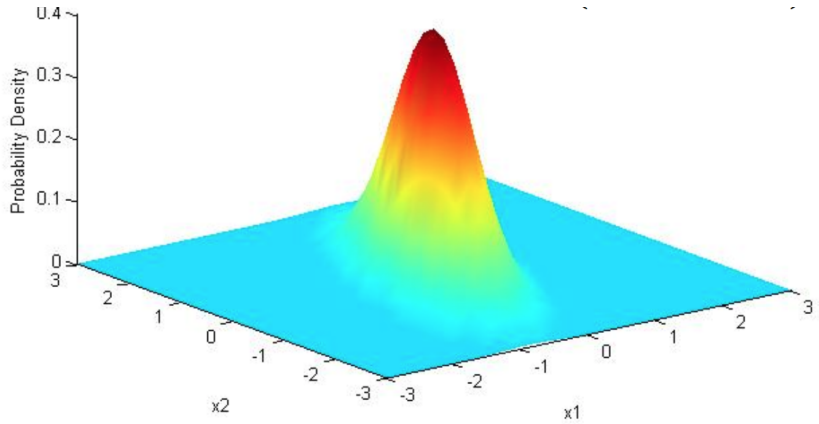
# Outline

# Gaussian distribution

For $F$ dimensions, the Gaussian distribution of a vector $\mathbf{x} \in \mathbb{R}^F$ is defined by:

$$\mathcal{N}(\mathbf{x}|\mathbf{u}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{F/2}\sqrt{|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{u})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{u})\right),$$

where $\mathbf{u}$ is the mean vector, $\mathbf{\Sigma}$ is the covariance matrix of the Gaussian.
Example: $\mathbf{u} = [0; 0]$, $\mathbf{\Sigma} = [0.25, 0.3; 0.3, 0.1]$.



# Gaussian mixture model (GMM)

$$\mathbb{P}(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}|\mathbf{u}_k, \mathbf{\Sigma}_k).$$

- $w_k$: mixing weight
- $\mathbf{u}_k$: component mean vector
- $\mathbf{\Sigma}_k$: component covariance matrix; if $\mathbf{\Sigma}_k = \sigma_k^2 \mathbf{I}$, the GMM is said to be spherical

Data samples independently generated from a GMM $\Rightarrow$
**Correct target clustering** of the samples according to which Gaussian distribution they come from

**Definition 1 (correct target clustering)**

Suppose

$$\mathbf{V} := [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$$

are samples independently generated from a $K$-component GMM. The **correct target clustering**

$$\mathscr{I} := \{\mathscr{I}_1, \mathscr{I}_2, \ldots, \mathscr{I}_K\}$$

of them satisfies $n \in \mathscr{I}_k$ iff $\mathbf{v}_n$ comes from the $k$-th component.

# Learning GMM

Data samples independently generated from a GMM $\Rightarrow$
**Correct target clustering** of the samples according to which Gaussian distribution they come from

**Definition 1 (correct target clustering)**

Suppose

$$\mathbf{V} := [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]$$

are samples independently generated from a $K$-component GMM. The **correct target clustering**

$$\mathscr{I} := \{\mathscr{I}_1, \mathscr{I}_2, \ldots, \mathscr{I}_K\}$$

of them satisfies $n \in \mathscr{I}_k$ iff $\mathbf{v}_n$ comes from the $k$-th component.

Thereby inferring the important parameters of the GMM.

# Algorithms for learning GMM

i) Expectation Maximization (EM)

- A local-search heuristic approach for maximum likelihood estimation in the presence of incomplete data;
- Cannot guarantee the convergence to global optima.

ii) Algorithms based on spectral decomposition and method of moments;

**Definition 2 (non-degeneracy condition)**

The component mean vectors

$$\mathbf{u}_1, \ldots, \mathbf{u}_K$$

span a $K$-dimensional subspace, and the mixing weight $w_k > 0$, for $k \in \{1, 2, \ldots, K\}$.

iii) Algorithms proposed by pure computer scientists;
Need to assume **separability assumptions**.
Vempala and Wang [2002]: for any $i, j \in [K]$, $i \neq j$,

$$\|\mathbf{u}_i - \mathbf{u}_j\|_2 > C \max\{\sigma_i, \sigma_j\} K^{\frac{1}{4}} \log^{\frac{1}{4}}(\frac{F}{w_{\min}}).$$

iii) Algorithms proposed by pure computer scientists;
Need to assume **separability assumptions**.
Vempala and Wang [2002]: for any $i, j \in [K]$, $i \neq j$,

$$\|\mathbf{u}_i - \mathbf{u}_j\|_2 > C \max\{\sigma_i, \sigma_j\} K^{\frac{1}{4}} \log^{\frac{1}{4}}(\frac{F}{w_{\min}}).$$

A simple spectral algorithm with running time polynomial in both $F$ and $K$ works well for correctly clustering samples.

# The $k$-means algorithm

Large number of algorithms for finding the (approximately) correct clustering of GMM;

# The $k$-means algorithm

Large number of algorithms for finding the (approximately) correct clustering of GMM;

Many practitioners stick with $k$-**means** algorithm because of its simplicity and successful applications in various fields.

# The objective function of $k$-means

Objective function: the so-called **distortion**.

$$\mathcal{D}(\mathbf{V}, \mathscr{I}) := \sum_{k=1}^{K} \sum_{n \in \mathscr{I}_k} \|\mathbf{v}_n - \mathbf{c}_k\|_2^2,$$

where

- $\mathscr{I}_k$: the index set of $k$-th cluster;
- $\mathbf{c}_k := \frac{1}{|\mathscr{I}_k|} \sum_{n \in \mathscr{I}_k} \mathbf{v}_n$ is the centroid of the $k$-th cluster.

# The objective function of $k$-means

Objective function: the so-called **distortion**.

$$\mathcal{D}(\mathbf{V}, \mathscr{I}) := \sum_{k=1}^{K} \sum_{n \in \mathscr{I}_k} \|\mathbf{v}_n - \mathbf{c}_k\|_2^2,$$

where

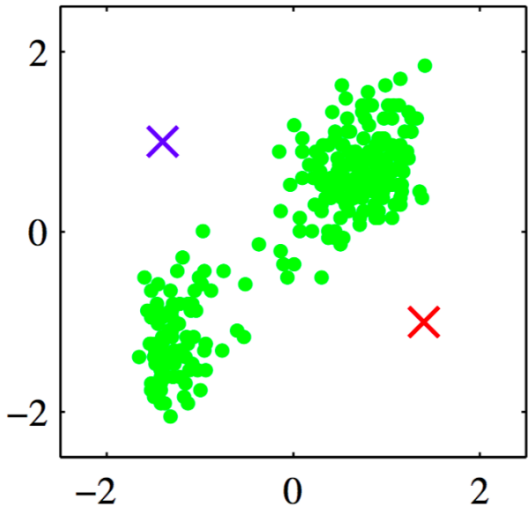
- $\mathscr{I}_k$: the index set of $k$-th cluster;
- $\mathbf{c}_k := \frac{1}{|\mathscr{I}_k|} \sum_{n \in \mathscr{I}_k} \mathbf{v}_n$ is the centroid of the $k$-th cluster.

Finding an **optimal clustering** $\mathscr{I}^{\mathrm{opt}}$ that satisfies

$$\mathcal{D}(\mathbf{V}, \mathscr{I}^{\mathrm{opt}}) = \min_{\mathscr{I}} \mathcal{D}(\mathbf{V}, \mathscr{I}).$$

## $k$-Means: By Example

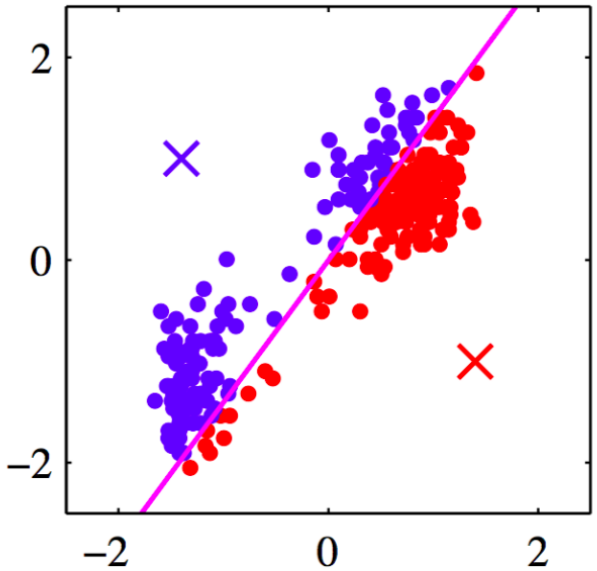
- Standardize the data.
- Choose two cluster centers.




From Bishop's *Pattern recognition and machine learning*, Figure 9.1(a).

# $k$-means algorithm

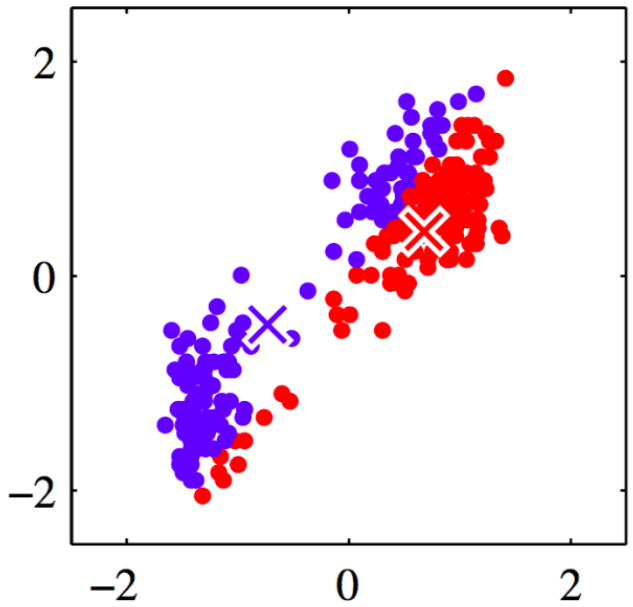
- Assign each point to closest center.




From Bishop's *Pattern recognition and machine learning*, Figure 9.1(b).

# $k$-means algorithm

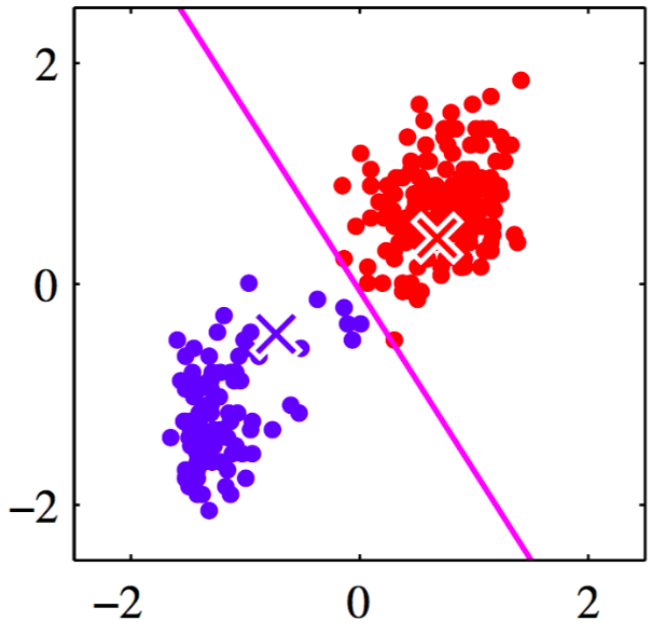
- Compute new class centers.




From Bishop's *Pattern recognition and machine learning*, Figure 9.1(c).

# $k$-means algorithm

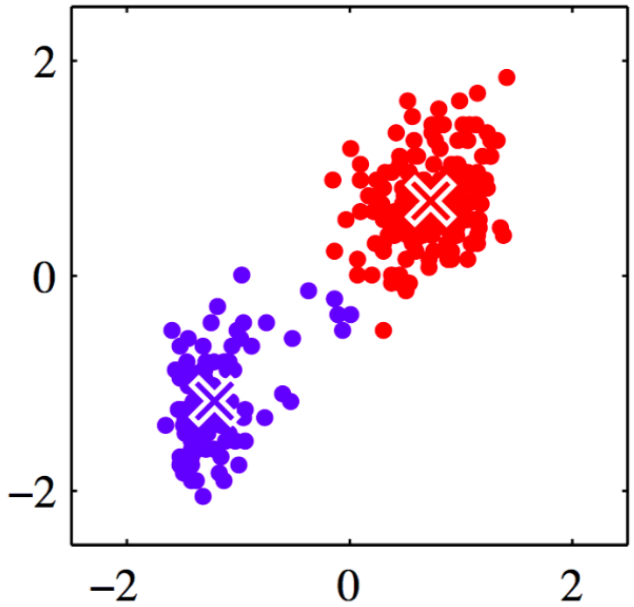
- Assign points to closest center.




From Bishop's *Pattern recognition and machine learning*, Figure 9.1(d).

# $k$-means algorithm

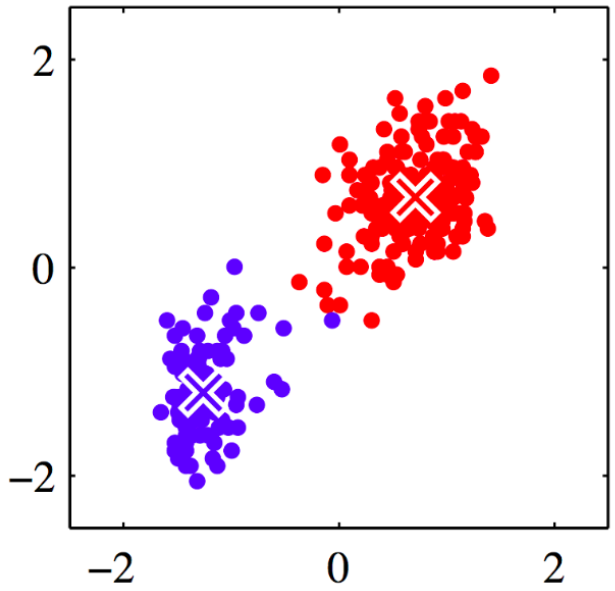
- Compute cluster centers.




From Bishop's *Pattern recognition and machine learning*, Figure 9.1(e).

# $k$-means algorithm

- Iterate until convergence.




From Bishop's *Pattern recognition and machine learning*, Figure 9.1(i).

# Using $k$-means to learn GMM?

**Can we simply use $k$-means to learn the correct clustering of GMM?**

# Using $k$-means to learn GMM?

**Yes!**

Kumar and Kannan [2010]:
Data points satisfy a so-called proximity condition (which is satisfied by the data points independently generated from a GMM with a certain separability assumption)
$\Rightarrow$
$k$-means algorithm with a proper initialization can correctly cluster nearly all data points

# Using $k$-means to learn GMM?

The key condition to be satisfied for performing $k$-means to learn GMM?

# Using $k$-means to learn GMM?

The key condition to be satisfied for performing $k$-means to learn GMM?

**The correct clustering $\approx$ Any optimal clustering**

# Main contributions

We prove if

- data points generated from a $K$-component spherical GMM;
- non-degeneracy condition and an separability assumption;

**The correct clustering $\approx$ Any optimal clustering**

# Main contributions

We also prove if

- data points generated from a $K$-component spherical GMM;
- projected onto the low-dimensional space;
- non-degeneracy condition and an even **weaker** separability assumption;

**The correct clustering $\approx$ Any optimal clustering for the dimensionality-reduced dataset**

# Advantages of dimensionality reduction

- Significantly faster running time
- Reduced memory usage
- Weaker separability assumption
- Other advantages

# Lower bound of distortion

Let $\mathbf{Z}$ be the centralized data matrix of $\mathbf{V}$ and denote $\mathbf{S} = \mathbf{Z}^T\mathbf{Z}$. According to Ding and He [2004], for any $K$-clustering $\mathscr{I}$,

$$\mathcal{D}(\mathbf{V}, \mathscr{I}) \geq \mathcal{D}^*(\mathbf{V}) := \mathrm{tr}(\mathbf{S}) - \sum_{k=1}^{K-1} \lambda_k(\mathbf{S}),$$

where

$$\lambda_1(\mathbf{S}) \geq \lambda_2(\mathbf{S}) \geq \ldots \geq 0$$

are the sorted eigenvalues of $\mathbf{S}$.

# Misclassification error (ME) distance

## Definition 3 (ME distance)

The misclassification error distance of any two $K$-clusterings

$$\mathscr{I}^1 := \{\mathscr{I}_1^1, \mathscr{I}_2^1, \ldots, \mathscr{I}_K^1\}, \quad \text{and}$$
$$\mathscr{I}^2 := \{\mathscr{I}_1^2, \mathscr{I}_2^2, \ldots, \mathscr{I}_K^2\}$$

is defined as

$$\mathrm{d}(\mathscr{I}^1, \mathscr{I}^2) := 1 - \frac{1}{N} \max_{\pi \in \mathcal{P}_K} \sum_{k=1}^{K} |\mathscr{I}_k^1 \bigcap \mathscr{I}_{\pi(k)}^2|,$$

where $\pi \in \mathcal{P}_K$ represents that the distance is minimized over all permutations of the labels $\{1, 2, \ldots, K\}$.

Meilă [2005]: ME distance defined above is indeed a metric.

# Important lemma

## Lemma 1 (Meilă, 2006)

- *Given* $\mathscr{I} := \{\mathscr{I}_1, \mathscr{I}_2, \ldots, \mathscr{I}_K\}$; *dataset* $\mathbf{V}$;
- $p_{\max} := \max_k \frac{1}{N}|\mathscr{I}_k|$, $p_{\min} := \min_k \frac{1}{N}|\mathscr{I}_k|$. *Denote*

$$\delta := \frac{\mathcal{D}(\mathbf{V}, \mathscr{I}) - \mathcal{D}^*(\mathbf{V})}{\lambda_{K-1}(\mathbf{S}) - \lambda_K(\mathbf{S})}.$$

- $\delta \leq \frac{1}{2}(K-1)$ *and* $\tau(\delta) := 2\delta(1 - \delta/(K-1)) \leq p_{\min}$.
  $\Rightarrow$

$$\mathrm{d}(\mathscr{I}, \textbf{optimal}) \leq p_{\max}\tau(\delta),$$

# Definitions

Define the increasing function

$$\zeta(p) := \frac{p}{1 + \sqrt{1 - 2p/(K-1)}},$$

the average variances

$$\bar{\sigma}^2 := \sum_{k=1}^{K} w_k \sigma_k^2$$

and the minimum eigenvalue

$$\lambda_{\min} := \lambda_{K-1} \left( \sum_{k=1}^{K} w_k (\mathbf{u}_k - \bar{\mathbf{u}})(\mathbf{u}_k - \bar{\mathbf{u}})^T \right).$$

# Theorem for original datasets

## Theorem 1

- $\mathbf{V} \in \mathbb{R}^{F \times N}$: *samples generated from a $K$-component spherical GMM ($N > F > K$);*
- *The non-degeneracy condition;*
- $w_{\min} := \min_k w_k$, $w_{\max} := \max_k w_k$ *and assume*

$$\delta_0 := \frac{(K-1)\bar{\sigma}^2}{\lambda_{\min}} < \zeta(w_{\min}).$$

*For sufficiently large $N$, w.h.p.,*

$$\mathrm{d}(\textbf{correct}, \textbf{optimal}) \leq \tau\left(\delta_0\right) w_{\max}.$$

# Remark for separability assumption

## Remark 1

*The condition $\delta_0 < \zeta(w_{\min})$ can be considered as a separability assumption. For example,*

- *$K = 2$: $\lambda_{\min} = w_1 w_2 \|\mathbf{u}_1 - \mathbf{u}_2\|_2^2$ and we have*

$$\|\mathbf{u}_1 - \mathbf{u}_2\|_2 > \frac{\bar{\sigma}}{\sqrt{w_1 w_2 \zeta(w_{\min})}}.$$

# Remark for non-degeneracy condition

**Remark 2**

*The non-degeneracy condition is used to ensure that* $\lambda_{\min} > 0$.

- $K = 2$: $\lambda_{\min} = w_1 w_2 \|\mathbf{u}_1 - \mathbf{u}_2\|_2^2$ *and we only need the two component mean vectors are distinct and we do not need that they are linearly independent.*

# Theorem for dimensionality-reduced datasets

## Theorem 2

- $\mathbf{V} \in \mathbb{R}^{F \times N}$: *generated under the same conditions given in Theorem 1;*
- *The separability assumption being modified to*

$$\delta_1 := \frac{(K-1)\bar{\sigma}^2}{\lambda_{\min} + \bar{\sigma}^2} < \zeta(w_{\min}).$$

- $\tilde{\mathbf{V}} \in \mathbb{R}^{(K-1) \times N}$: *the post-*$(K-1)$*-PCA dataset of* $\mathbf{V}$.

*For sufficiently large* $N$*, w.h.p.,*

$$\mathrm{d}(\textbf{correct}, \tilde{\textbf{optimal}}) \leq \tau\left(\delta_1\right) w_{\max}.$$

# PCA vs. PCA with no centering

## Corollary 1

- $\mathbf{V} \in \mathbb{R}^{F \times N}$: *generated under the same conditions given in Theorem 1;*
- $\hat{\mathbf{V}}$: *the post-K-SVD dataset of* $\mathbf{V}$;

*For sufficiently large N, w.h.p.,*

$$\mathrm{d}(\mathbf{correct}, \hat{\mathbf{optimal}}) \leq \tau\left(\delta_0\right) w_{\max}.$$

# Advantages of PCA over PCA with no centering

- Requires weaker separability assumption;
- Smaller upper bound for ME distance;
- $K = 2$: projecting to 1-D subspace by PCA instead of projecting to 2-D subspace by PCA with no centering.

# Upper bound for ME distance between optimal clusterings

Combining the results of Theorem 1 and Theorem 2, by the triangle inequality:

**Corollary 2**

- $\mathbf{V} \in \mathbb{R}^{F \times N}$: *generated under the same conditions given in Theorem 1;*
- $\tilde{\mathbf{V}}$: *the post-*$(K-1)$*-PCA dataset of* $\mathbf{V}$.

*For sufficiently large* $N$*, w.h.p.*

$$\mathrm{d}(\mathbf{optimal}, \tilde{\mathbf{optimal}}) \leq (\tau(\delta_0) + \tau(\delta_1))\, w_{\max}.$$

# Parameter settings

$K = 2$, for all $k = 1, 2$, we set

$$\sigma_k^2 = \frac{\lambda_{\min}\zeta(w_{\min} - \varepsilon)}{4(K-1)}, \text{ corr. to } \frac{\delta_0}{\zeta(w_{\min})} \approx \frac{1}{4},$$

or

$$\sigma_k^2 = \frac{\lambda_{\min}\zeta(w_{\min} - \varepsilon)}{K-1}, \text{ corr. to } \frac{\delta_0}{\zeta(w_{\min})} \approx 1,$$

where $\varepsilon = 10^{-6}$.

# Parameter settings

$K = 2$, for all $k = 1, 2$, we set

$$\sigma_k^2 = \frac{\lambda_{\min}\zeta(w_{\min} - \varepsilon)}{4(K-1)}, \text{ corr. to } \frac{\delta_0}{\zeta(w_{\min})} \approx \frac{1}{4},$$

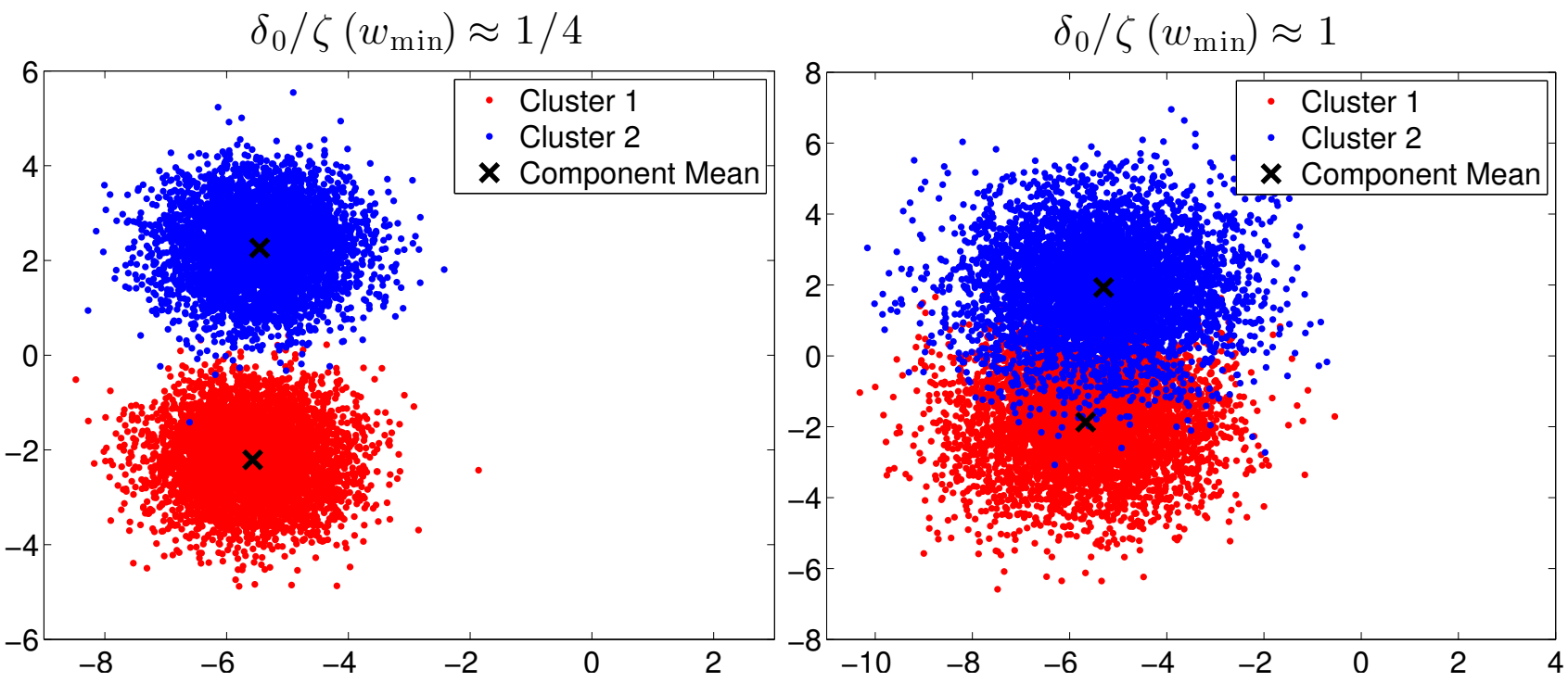
or

$$\sigma_k^2 = \frac{\lambda_{\min}\zeta(w_{\min} - \varepsilon)}{K-1}, \text{ corr. to } \frac{\delta_0}{\zeta(w_{\min})} \approx 1,$$

where $\varepsilon = 10^{-6}$.

The former corresponds to well-separated clusters; the latter corresponds to moderately well-separated clusters

# Visualization of post-2-SVD datasets



# Original datasets

$\mathrm{d}_{\mathrm{org}} := \mathrm{d}(\mathscr{I}, \mathscr{I}^{\mathrm{opt}})$, $\bar{\mathrm{d}}_{\mathrm{org}} := \tau(\delta_0) w_{\max}$.

$\delta_0^{\mathrm{emp}} := \frac{\mathcal{D}(\mathbf{V}, \mathscr{I}) - \mathcal{D}^*(\mathbf{V})}{\lambda_{K-1}(\mathbf{S}) - \lambda_K(\mathbf{S})}$ is an approximation of $\delta_0$,

$\bar{\mathrm{d}}_{\mathrm{org}}^{\mathrm{emp}} := \tau(\delta_0^{\mathrm{emp}}) p_{\max}$ is an approximation of $\bar{\mathrm{d}}_{\mathrm{org}}$.

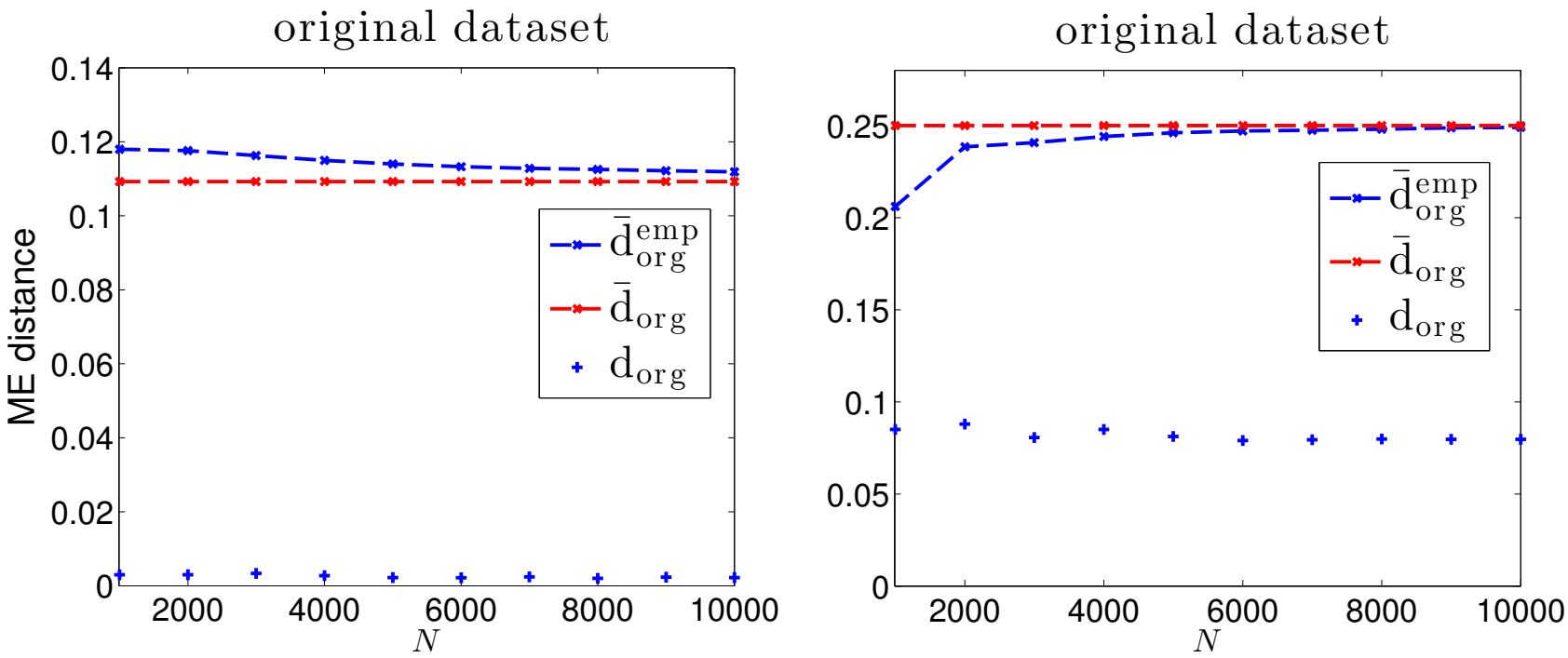



Figure: True distances and their corresponding upper bounds for original datasets.

# Dimensionality-reduced datasets

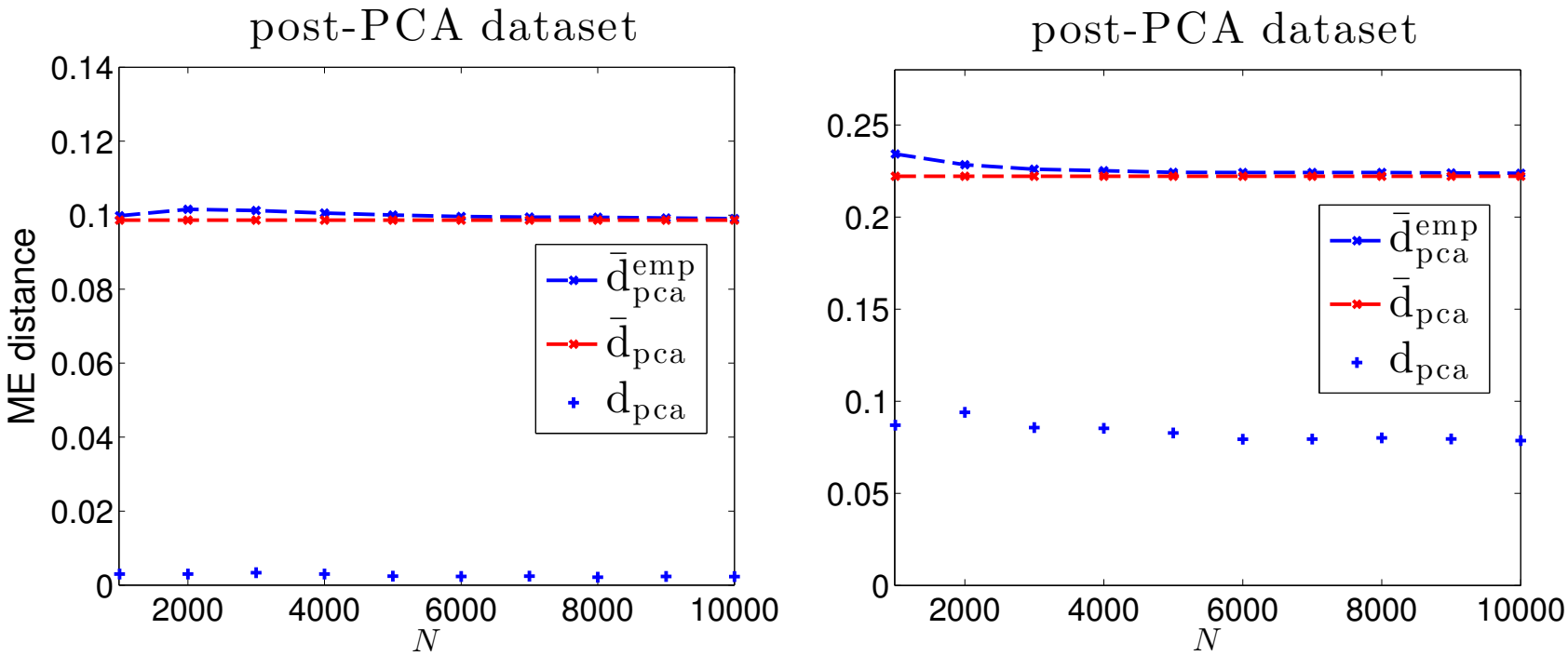
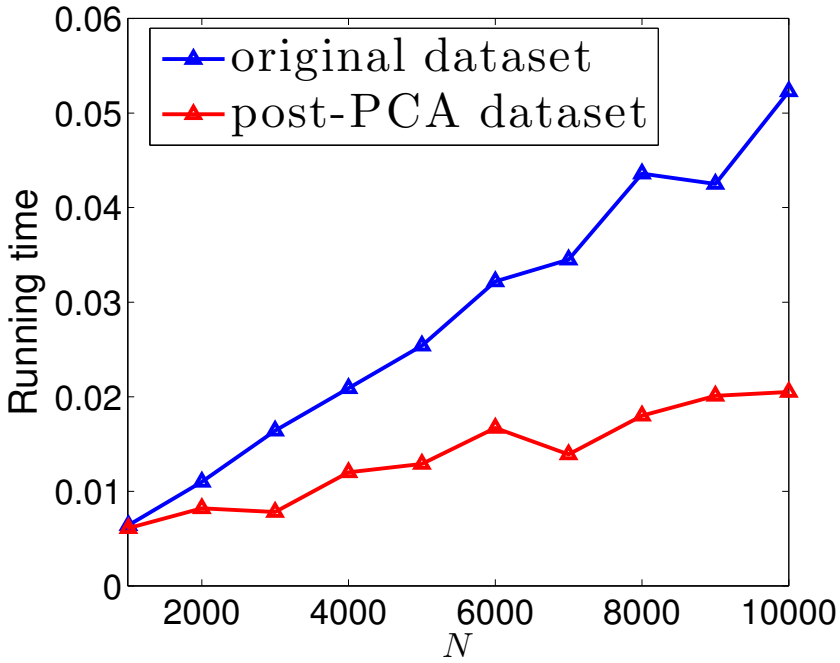
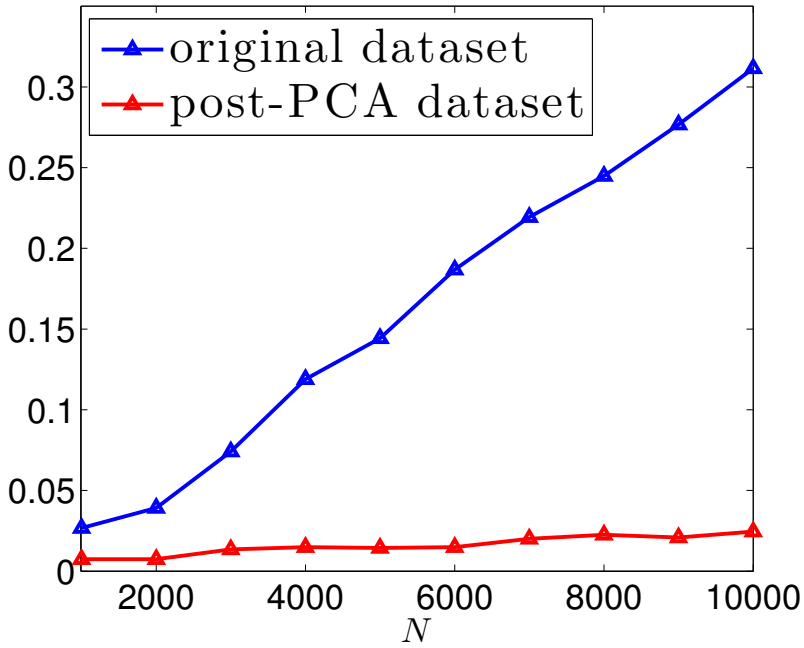



Figure: True distances and their corresponding upper bounds for post-PCA datasets.

# Comparisons of running time



# Further extensions

- Randomized SVD instead of exact SVD;
- Random projection;
- Non-spherical Gaussian or even more general distributions, e.g., logconcave distributions;

# Outline

1. The Informativeness of $k$-Means and Dimensionality Reduction for Learning Mixture Models
   - Gaussian Mixture Models and $k$-Means
   - Main Contributions
   - Lemmas and Our Theorems
   - Experiments
   - Further Extensions
2. Rank-One NMF-Based Initialization for NMF and Relative Error Bounds under a Geometric Assumption (IEEE TSP)
   - NMF and Classical Algorithms
   - Our Geometric Assumption for NMF
   - Non-Probabilistic and Probabilistic Results
   - Automatically Determine $K$
   - Numerical Experiments

**NMF:**

Given $\mathbf{V} \in \mathbb{R}_+^{F \times N}$, $K \in \mathbb{N}_+$, $K \leq \min\{F, N\}$, find $\mathbf{W} \in \mathbb{R}_+^{F \times K}$ and $\mathbf{H} \in \mathbb{R}_+^{K \times N}$, to minimize $\|\mathbf{V} - \mathbf{WH}\|_{\mathrm{F}}$.

- Nonnegativity of $\mathbf{W}$ ensures interpretability of dictionary;
- Nonnegativity of $\mathbf{H}$ tends to produce parts-based representations because subtractive combinations are forbidden;

**Advantages:**

- Enhancing the interpretability;
- Promoting sparsity;

# 49 images among 4429 from MIT's CBCL face dataset

# PCA dictionary with $K = 25$

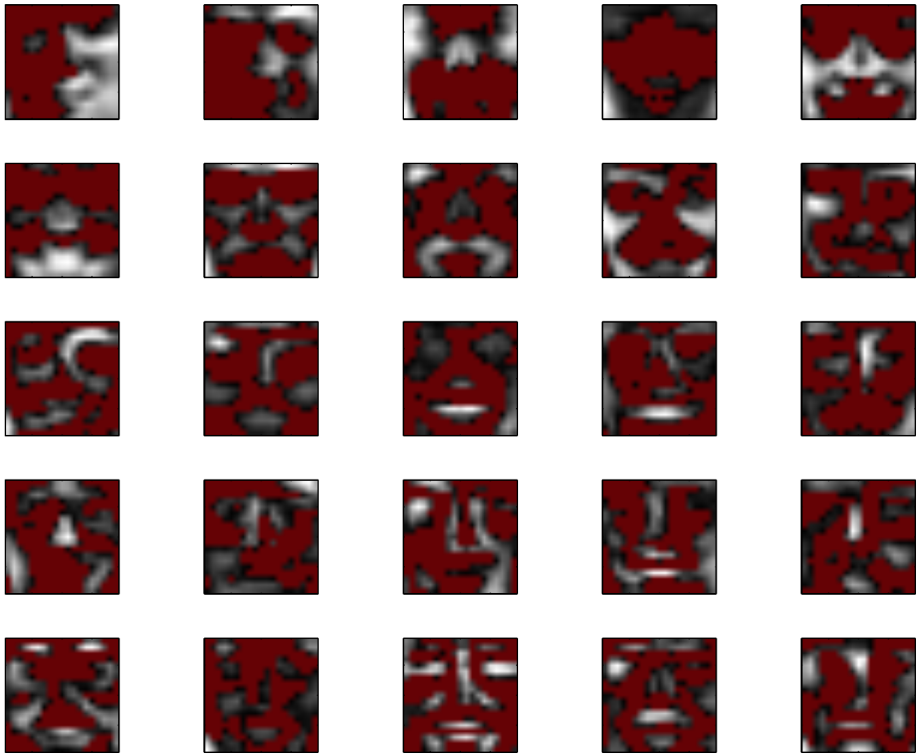


Figure: Red pixels indicate negative values

# NMF dictionary with $K = 25$

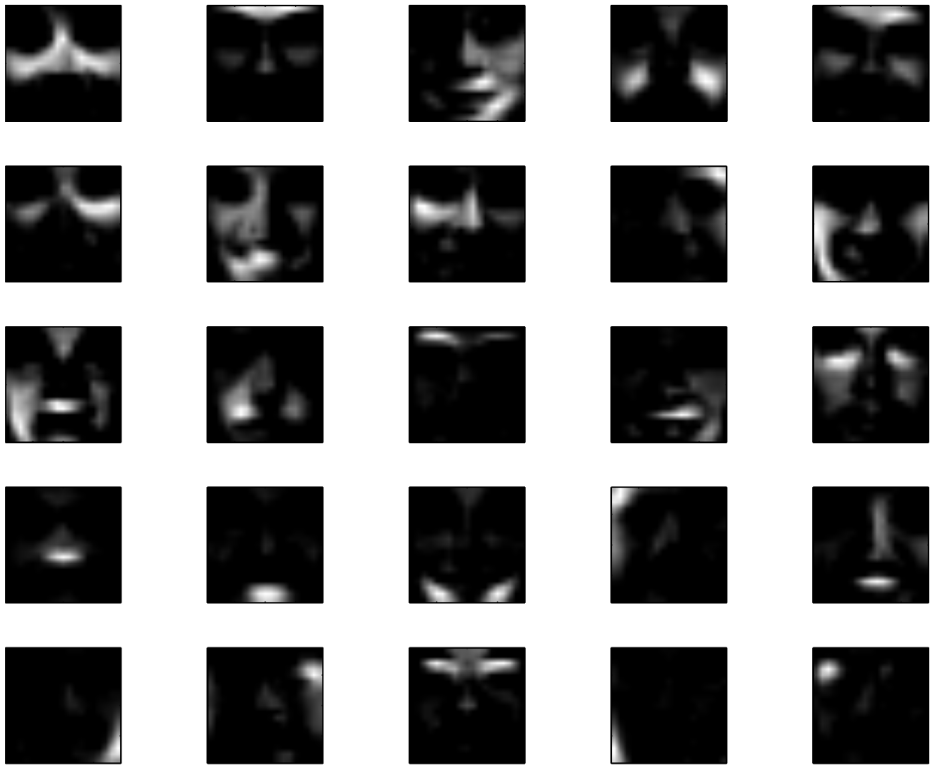


Figure: From Lee and Seung's seminal 1999 paper on NMF

1) Multiplicative update algorithm (MUA):

$$\mathbf{H} \leftarrow \mathbf{H}\frac{\mathbf{W}^T\mathbf{V}}{\mathbf{W}^T\mathbf{W}\mathbf{H}}, \mathbf{W} \leftarrow \mathbf{W}\frac{\mathbf{V}\mathbf{H}^T}{\mathbf{W}\mathbf{H}\mathbf{H}^T}$$

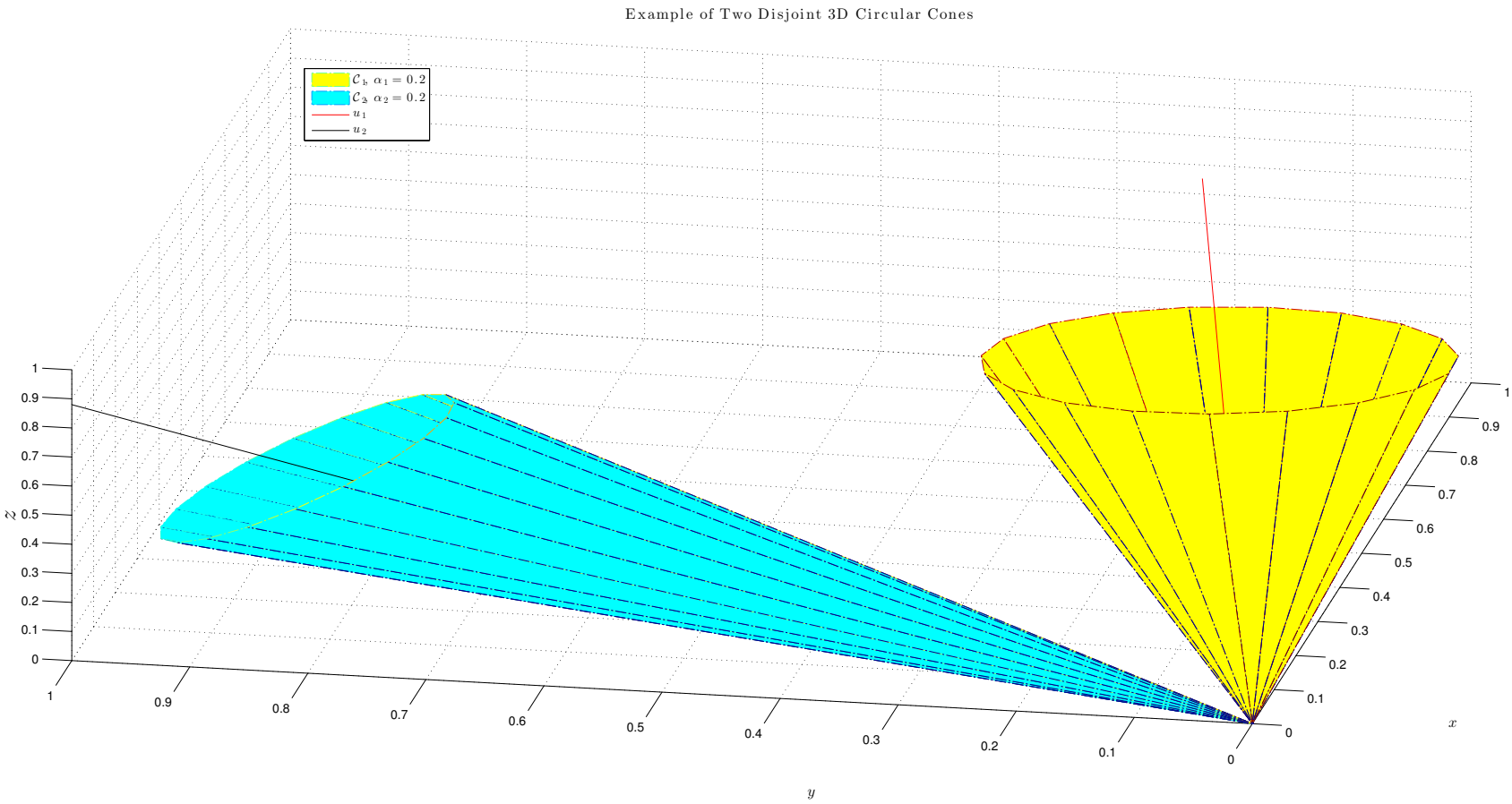
2) Alternating nonnegative least square (ANLS)-type algorithms, e.g.,

$$\mathbf{H} \leftarrow \left[\left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{V}\right]_+, \mathbf{W} \leftarrow \left[\mathbf{V}\mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T\right)^{-1}\right]_+$$

# Problems

- NP-hard;
- No guarantees beyond non-increasing of objective functions and the convergence to stationary points;
- No error bound analysis for classical algorithms;

# An Illustration



# Our Geometric Assumption

## Definition 4

A circular cone $C := \mathcal{C}(\mathbf{u}, \alpha)$ with unit vector ($l_2$ norm) $\mathbf{u}$ and angle $\alpha \in [0, \frac{\pi}{2})$ is defined as

$$C = \{\mathbf{x} \in \mathbb{R}^F_+, \mathbf{x} \neq 0 : \frac{\mathbf{x}^T\mathbf{u}}{\|\mathbf{x}\|_2} \geq \cos\alpha\},$$

$\mathbf{u}$ and $\alpha$ are called the **basis vector** and **size angle** of $C$ respectively.

## Definition 5

**Geometric assumption:**

$$\min_{i,j\in[K]} \alpha_{ij} > \max_{i,j\in[K]} \{\max\{\alpha_i + 3\alpha_j, 3\alpha_i + \alpha_j\}\}, \alpha_{ij} := \arccos\left(\mathbf{u}_i^T\mathbf{u}_j\right)$$

# Our Theorem for Clustering

**Theorem 3**

*Algorithm 1 can correctly cluster all data points generated from K circular cones satisfying the geometric assumption.*

---

**Algorithm 1** Greedy clustering method with geometric assumption in (2)

---

**Input**: Data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$, $K \in \mathbb{N}$

**Output**: A set of non-empty, pairwise disjoint index sets $\mathscr{I}_1, \mathscr{I}_2, \ldots, \mathscr{I}_K \subseteq [N]$ such that their union is $[N]$

1) Normalize $\mathbf{V}$ to obtain $\mathbf{V}'$, such that all the columns of $\mathbf{V}'$ have unit $l_2$ norm.

2) Arbitrarily pick a point $\mathbf{z}_1 \in \mathbf{V}'$ (i.e., $\mathbf{z}_1$ is a column in $\mathbf{V}'$) as the first centroid.

3) **for** $k = 1$ to $K - 1$ **do**

$$\mathbf{z}_{k+1} := \arg\min_{\mathbf{z} \in \mathbf{V}'} \{\max\{\mathbf{z}_i^T \mathbf{z}, i \in [k]\}\} \tag{3}$$

and set $\mathbf{z}_{k+1}$ be the $(k+1)$-st centroid.

4) $\mathscr{I}_k := \{n \in [N] : k = \arg\max_{j \in [K]} \mathbf{z}_j^T \mathbf{V}'(:, n)\}$ for all $k \in [K]$.

---

# A Non-Probabilistic Theorem

**Theorem 4**

*Suppose each column of* $\mathbf{V}$ *is picked from* $C_k := \mathcal{C}(\mathbf{u}_k, \alpha_k), k \in [K]$ *which satisfy the geometric assumption. Algorithm 2 chooses* $\mathbf{W}^* \in \mathbb{R}_+^{F \times K}$, $\mathbf{H}^* \in \mathbb{R}_+^{K \times N}$, *s.t.*

$$\frac{\|\mathbf{V} - \mathbf{W}^*\mathbf{H}^*\|_{\mathrm{F}}}{\|\mathbf{V}\|_{\mathrm{F}}} \leq \max_{k \in [K]} \{\sin \alpha_k\},$$

**Algorithm 2** Approximate NMF under the geometric assumption

**Input**: Data matrix $\mathbf{V} \in \mathbb{R}_+^{F \times N}$, $K \in \mathbb{N}$

**Output**: Factor matrices $\mathbf{W}^* \in \mathbb{R}_+^{F \times K}$, $\mathbf{H}^* \in \mathbb{R}_+^{K \times N}$

1) Use Algorithm 1 to find a set of non-empty, pairwise disjoint index sets $\mathscr{I}_1, \mathscr{I}_2, \ldots, \mathscr{I}_K \subseteq [N]$.

2) **for** $k = 1$ to $K$ **do**

$$\begin{aligned} \mathbf{V}_k &:= \mathbf{V}(:, \mathscr{I}_k), \\ [\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{X}_k] &:= \mathrm{svd}(\mathbf{V}_k), \\ \mathbf{w}_k^* &:= \mathbf{\Sigma}_k(1,1)\,|\mathbf{U}_k(:,1)|, \quad \mathbf{h}_k := |\mathbf{X}_k(:,1)|, \\ \mathbf{h}_k^* &:= \mathrm{zeros}(N,1), \mathbf{h}_k^*(\mathscr{I}_k) = \mathbf{h}_k. \end{aligned}$$

3) $\mathbf{W}^* := [\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_K^*]$, $\mathbf{H}^* := [\mathbf{h}_1^*; \mathbf{h}_2^*; \ldots; \mathbf{h}_K^*]$.

# The Generating Process for Each Column

Let $\lambda := (\lambda_1; \lambda_2; \ldots; \lambda_K) \in \mathbb{R}^K_{++}$, sample each column $\mathbf{v}$ of $\mathbf{V}$:

1. sample $k \in [K]$ with equal probability $1/K$;
2. sample the squared length $l$ from $\mathrm{Exp}(\lambda_k)$;[1]
3. uniformly sample a unit vector $\mathbf{z} \in C_k$;[2]
4. if $\mathbf{z} \notin \mathbb{R}^F_+$, project and rescale it;
5. let $\mathbf{v} = \sqrt{l}\mathbf{z}$;

[1] $\mathrm{Exp}(\lambda)$ is the function $x \mapsto \lambda \exp(-\lambda x) 1\{x \geq 0\}$.

[2] This means we first uniformly sample an angle $\beta \in [0, \alpha_k]$ and subsequently uniformly sample a vector $\mathbf{z}$ from the set $\{\mathbf{x} \in \mathbb{R}^F : \|\mathbf{x}\|_2 = 1, \mathbf{x}^T \mathbf{u}_k = \cos\beta\}$

## Theorem 5

*Let*

$$f(\alpha) := 0.5 - (\sin 2\alpha) / (4\alpha),$$

*then for small* $\epsilon > 0$, *w.p. at least*

$$1 - 8 \exp\left(-\xi N \epsilon^2\right),$$

*we have*

$$\frac{\|\mathbf{V} - \mathbf{W}^* \mathbf{H}^*\|_{\mathrm{F}}}{\|\mathbf{V}\|_{\mathrm{F}}} \leq \sqrt{\frac{\sum_{k=1}^{K} f(\alpha_k) / \lambda_k}{\sum_{k=1}^{K} 1/\lambda_k}} + \epsilon.$$

# The Upper Bound is Tight

**Theorem 6**

*If we do not project the vectors to nonnegative orthant, we have*

$$\frac{\|\mathbf{V}-\mathbf{W}^*\mathbf{H}^*\|_{\mathrm{F}}}{\|\mathbf{V}\|_{\mathrm{F}}} \xrightarrow{p} \sqrt{\frac{\sum_{k=1}^{K} f\left(\alpha_k\right)/\lambda_k}{\sum_{k=1}^{K} 1/\lambda_k}}$$

**Theorem 7**

*Assume*

- *size angle* $= \alpha$;
- *angles between distinct basis vectors of the circular cones* $= \beta$;
- *parameters for the exponential distributions* $= \lambda$;
- *circular cones are in* $\mathbb{R}_+^F$;
- $K \in \{K_{\min}, \ldots, K_{\max}\}$ *with* $K_{\min} > 1$, $K_{\max} < \operatorname{rank}(\mathbf{V})$.

*Then, for any* $t \geq 1$, *and small* $\epsilon$, *if* $N$ *is sufficiently large, w.h.p.,*

$$\frac{\sigma_K(\mathbf{V})}{\sigma_{K+1}(\mathbf{V})} = \max_{j \in \{K_{\min}, \ldots, K_{\max}\}} \frac{\sigma_j(\mathbf{V})}{\sigma_{j+1}(\mathbf{V})}.$$

# Automatically Determining $K$

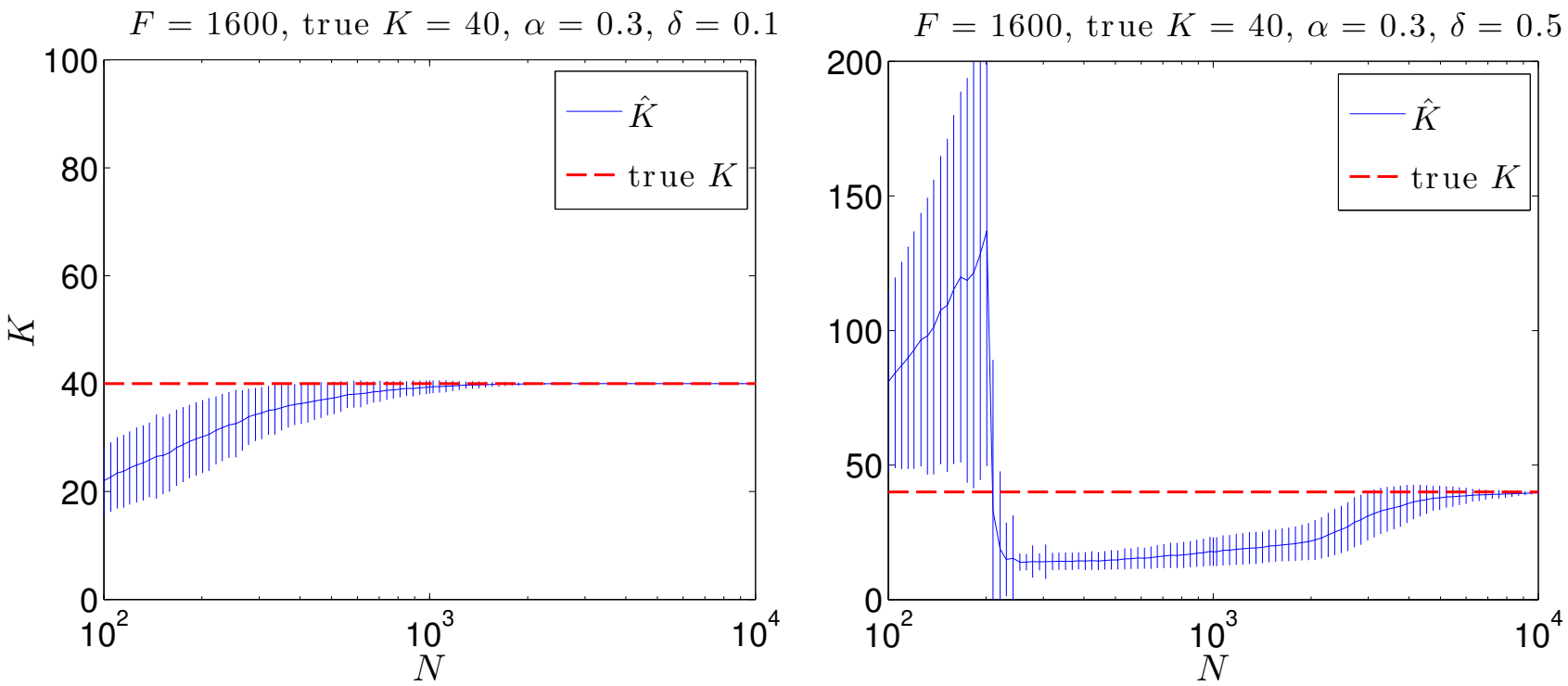



Figure: Estimated number of circular cones $K$ with different noise levels. The error bars denote one standard deviation away from the mean.

# Synthetic Dataset Test

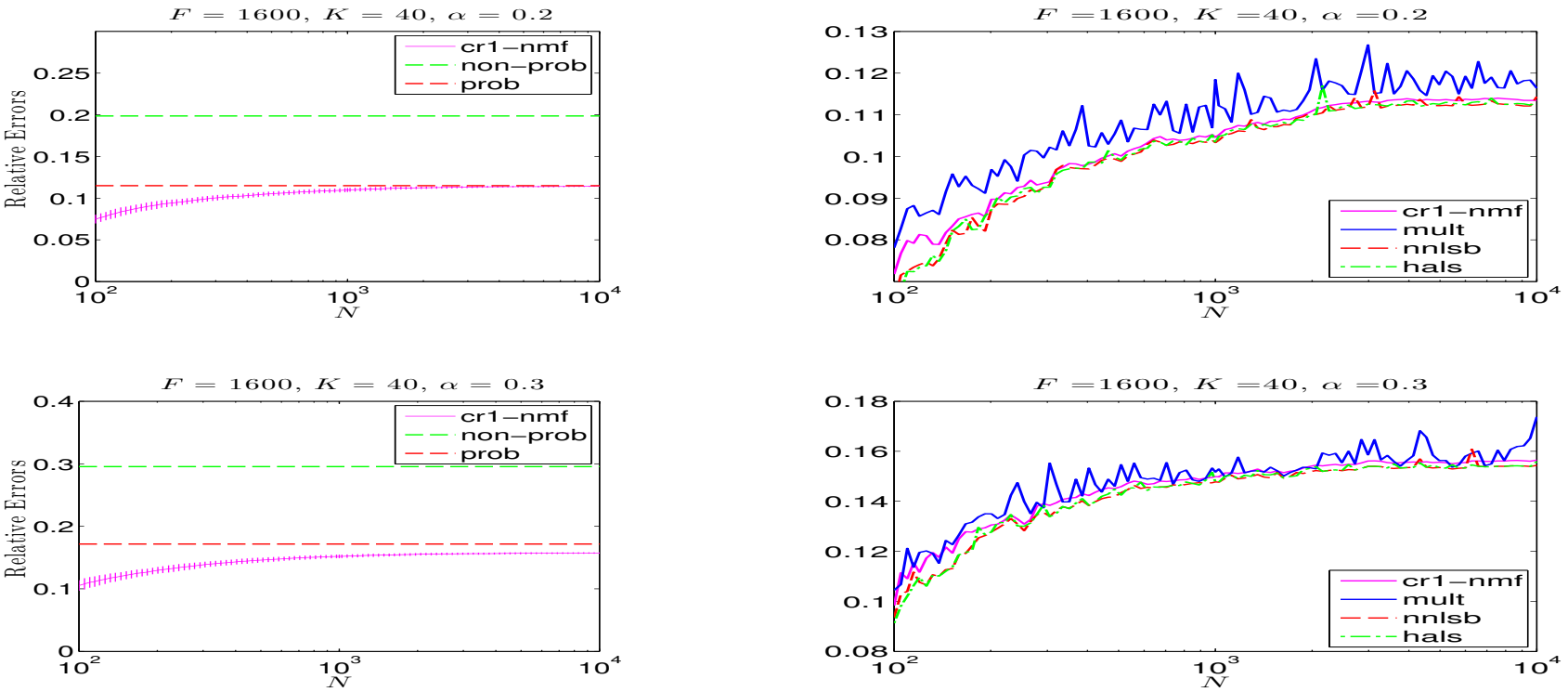



Figure: Errors and performances of various algorithms.

# Real Dataset Test

Table: Information for real datasets used

| Dataset Name | $F$ | $N$ | $K$ | Description |
|---|---|---|---|---|
| CK | $49 \times 64$ | 8795 | 97 | face dataset |
| faces94 | $200 \times 180$ | 3040 | 152 | face dataset |
| Georgia Tech | $480 \times 640$ | 750 | 50 | face dataset |
| PaviaU | 207400 | 103 | 9 | hyperspectral |

# Initialization Performances I

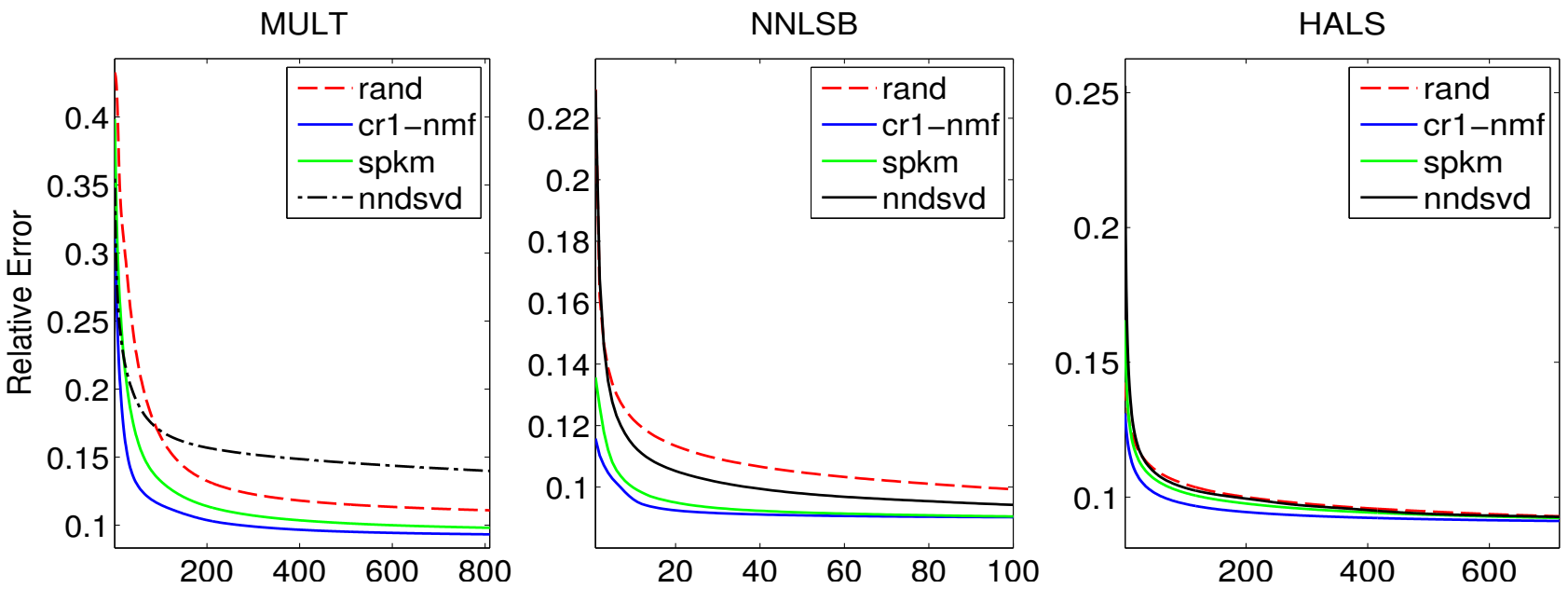



Figure: CK dataset.

# Initialization Performances II

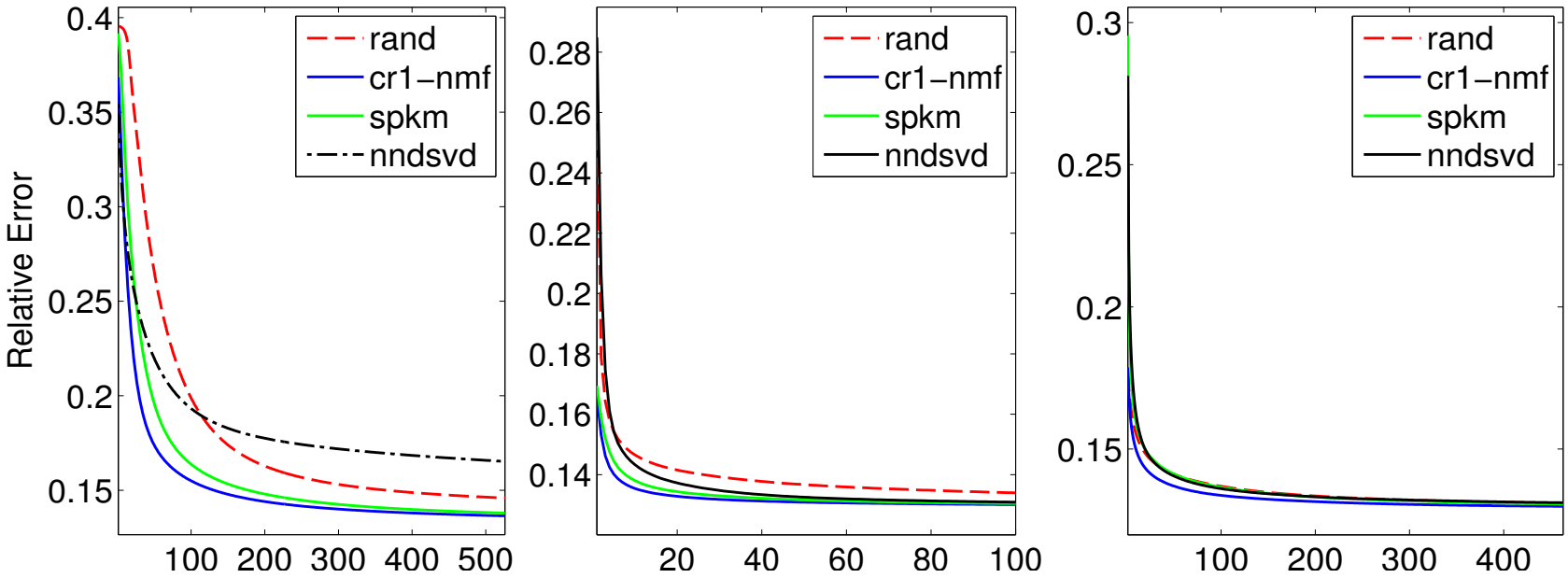



Figure: faces94 dataset.

# Initialization Performances III

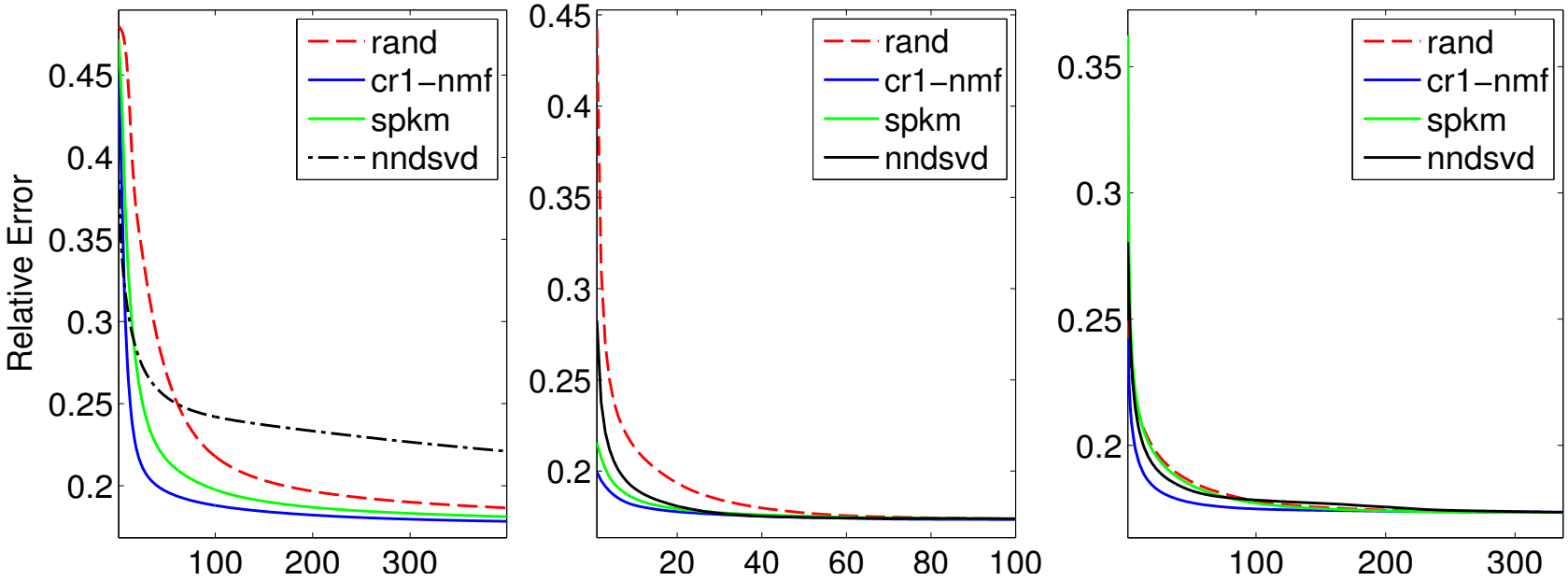



Figure: Georgia Tech dataset.

# Initialization Performances IV

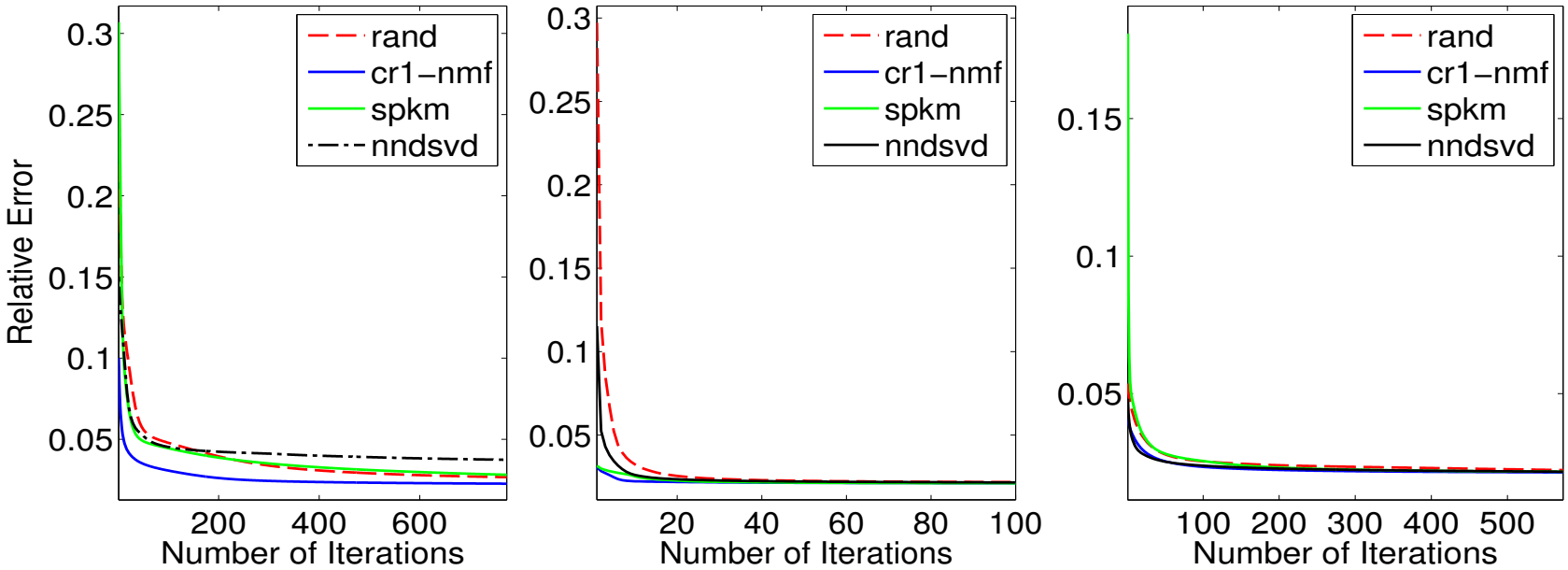



Figure: PaviaU dataset.