clustering

k–means like algorithms and beyond

CIKM 2003

Jacob Kogan
Charles Nicholas
Marc Teboulle

Outline of the talk

- how to build a partition
- how to improve a partition
- how to evaluate a partition

Partitions

$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ is a set of vectors in $\mathbf{R}^n$.

A partition Π of $\mathbf{X}$ is

$$\Pi = \{\pi_1, \ldots, \pi_k\}$$

$$\pi_1 \cup \ldots \cup \pi_k = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}, \text{ and } \pi_i \bigcap \pi_j = \emptyset \text{ if } i \neq j.$$

q is a real valued function whose domain is the set of subsets of $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$.

The quality of the partition is given by

$$Q(\Pi) = q(\pi_1) + \ldots + q(\pi_k).$$

What do we want?

To identify an optimal partition

$$\Pi^o = \{\pi_1^o, \ldots, \pi_k^o\},$$

i.e., one that optimizes

$$Q(\Pi) = q(\pi_1) + \ldots + q(\pi_k).$$

In general the solution is available when the dimension of the vector space is **ONE**.

Data sets

Vector sets generated for large document collections contain vectors which are:

- sparse
- high dimensional
- have non-negative entries
- normalized (usually with l_2 norm 1)

For example the Reuters business news collection (available from David D. Lewis' home page: `http://www.research.att.com/~lewis`) contains 19043 non-empty documents with 44749 unique words.

The simplest way to go

Given a partition $\Pi^{(t)} = \left\{\pi_1^{(t)}, \ldots, \pi_k^{(t)}\right\}$,

build a partition $\Pi^{(t+1)} = \left\{\pi_1^{(t+1)}, \ldots, \pi_k^{(t+1)}\right\}$,

such that:

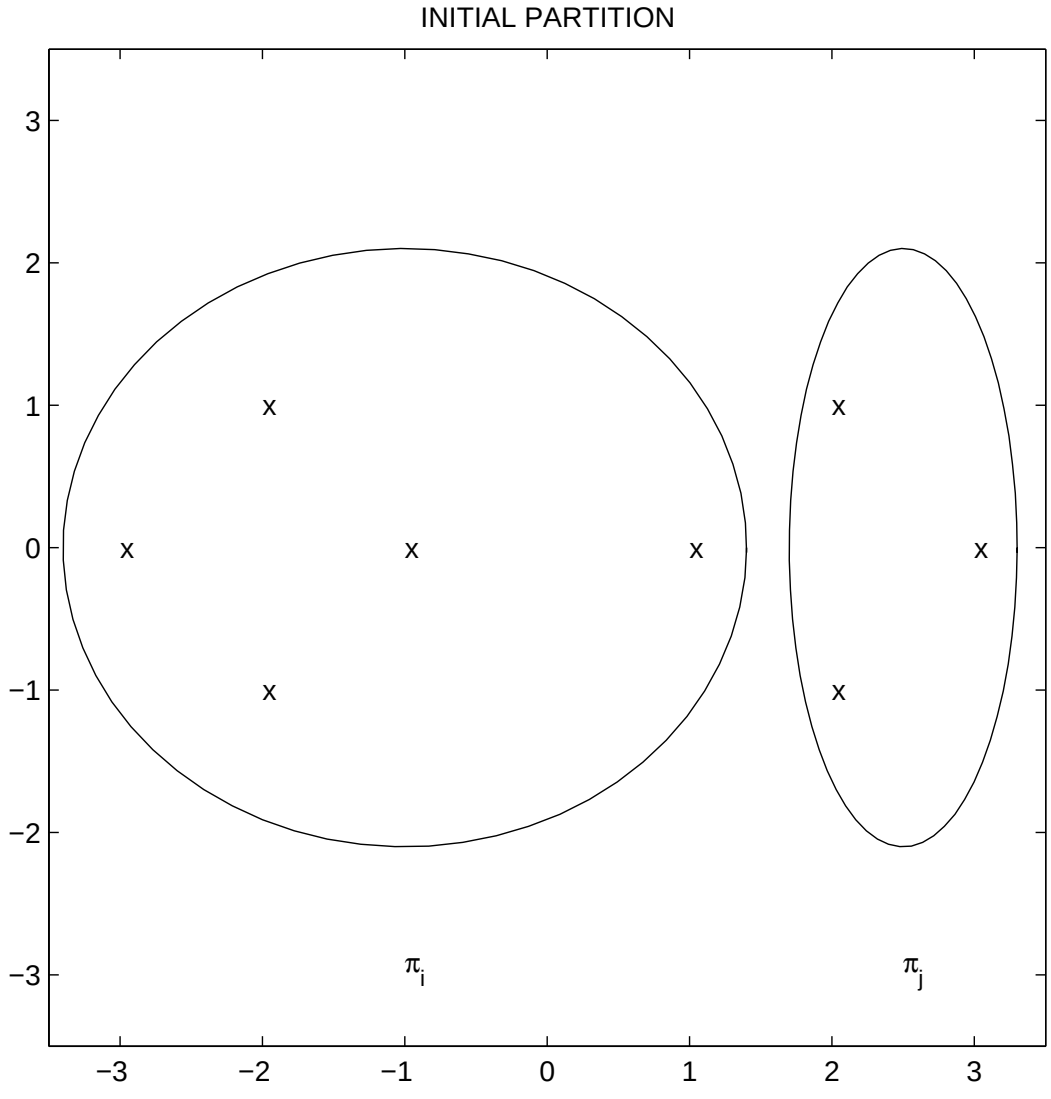
- there are clusters $\pi_i^{(t)}$, $\pi_j^{(t)}$, and $\mathbf{x} \in \pi_i^{(t)}$,
- $\pi_i^{(t+1)} = \pi_i^{(t)} - \{\mathbf{x}\}$, $\pi_j^{(t+1)} = \pi_j^{(t)} \bigcup \{\mathbf{x}\}$

and

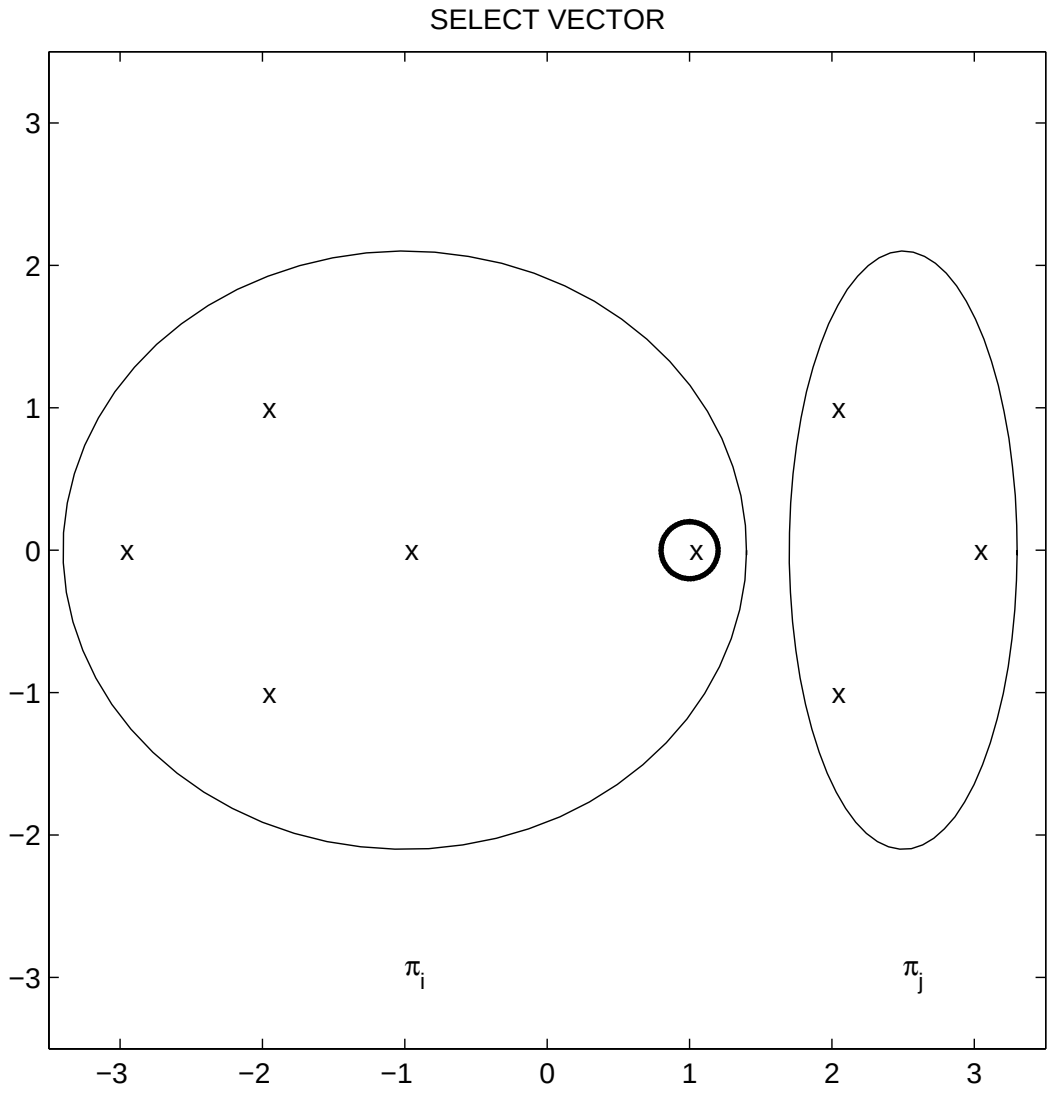
$$q\left(\pi_i^{(t+1)}\right) + q\left(\pi_j^{(t+1)}\right) < q\left(\pi_i^{(t)}\right) + q\left(\pi_j^{(t)}\right).$$

Partition

$$\Pi^{(t)} = \left\{ \pi_1^{(t)}, \pi_2^{(t)} \right\}$$

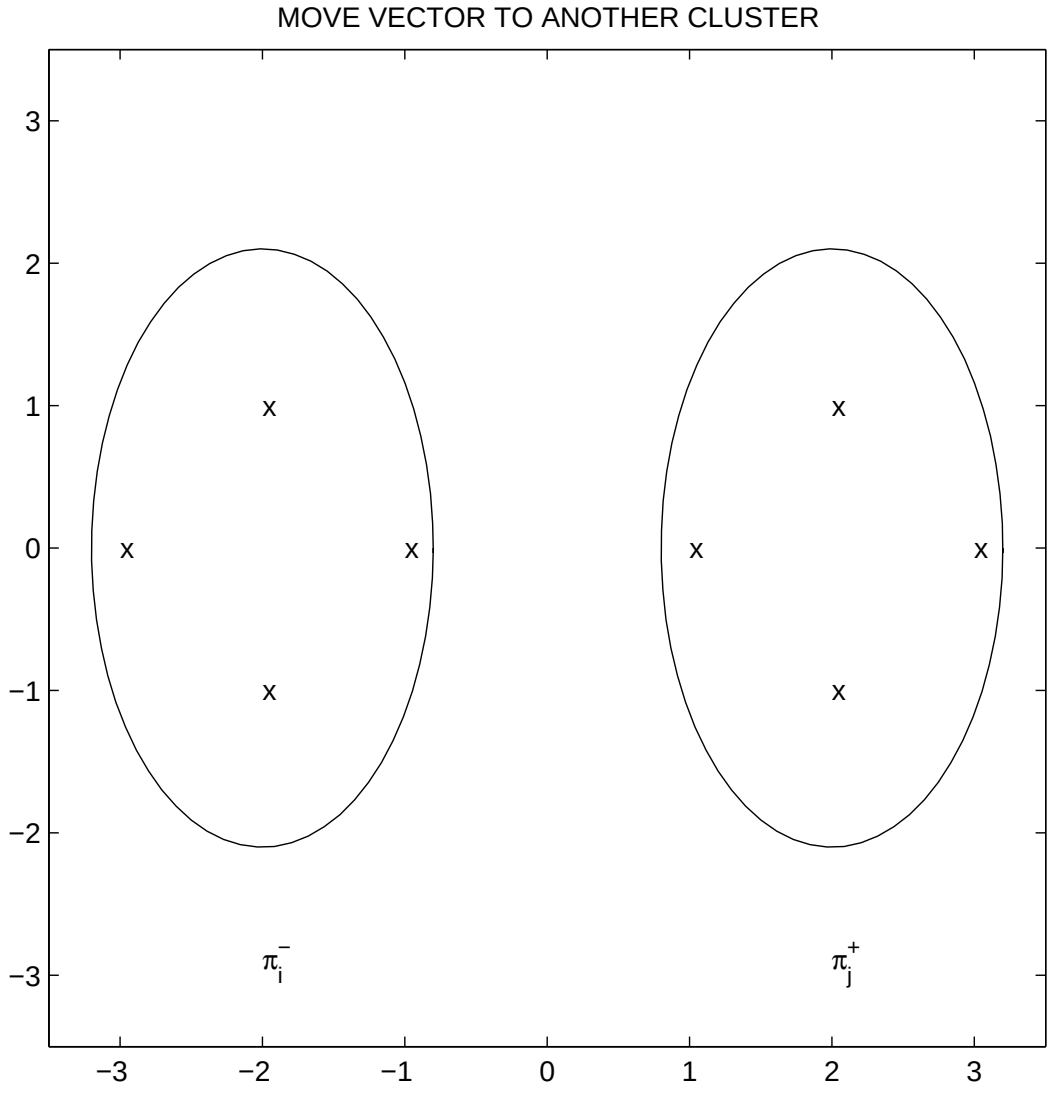


Pick a vector

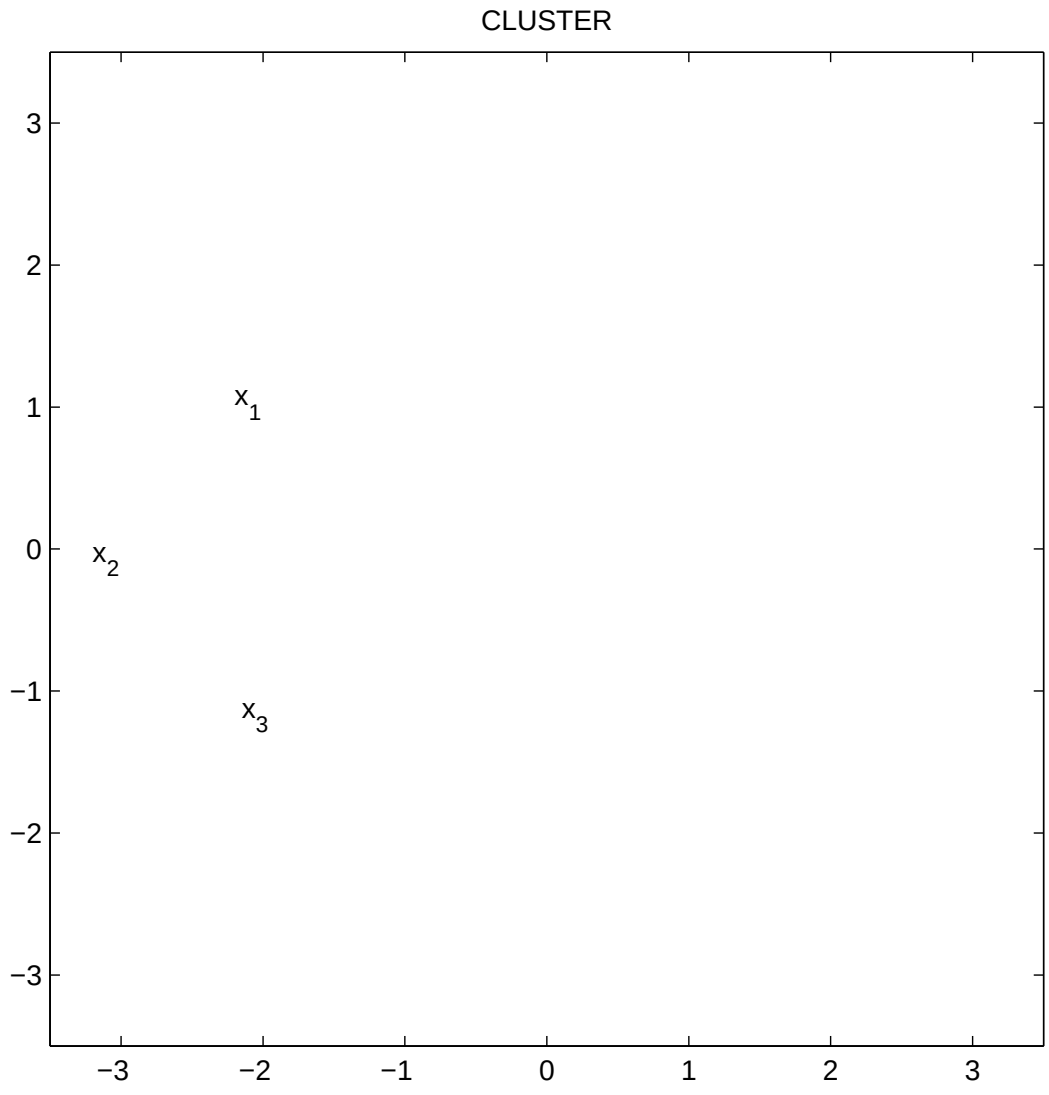


New partition

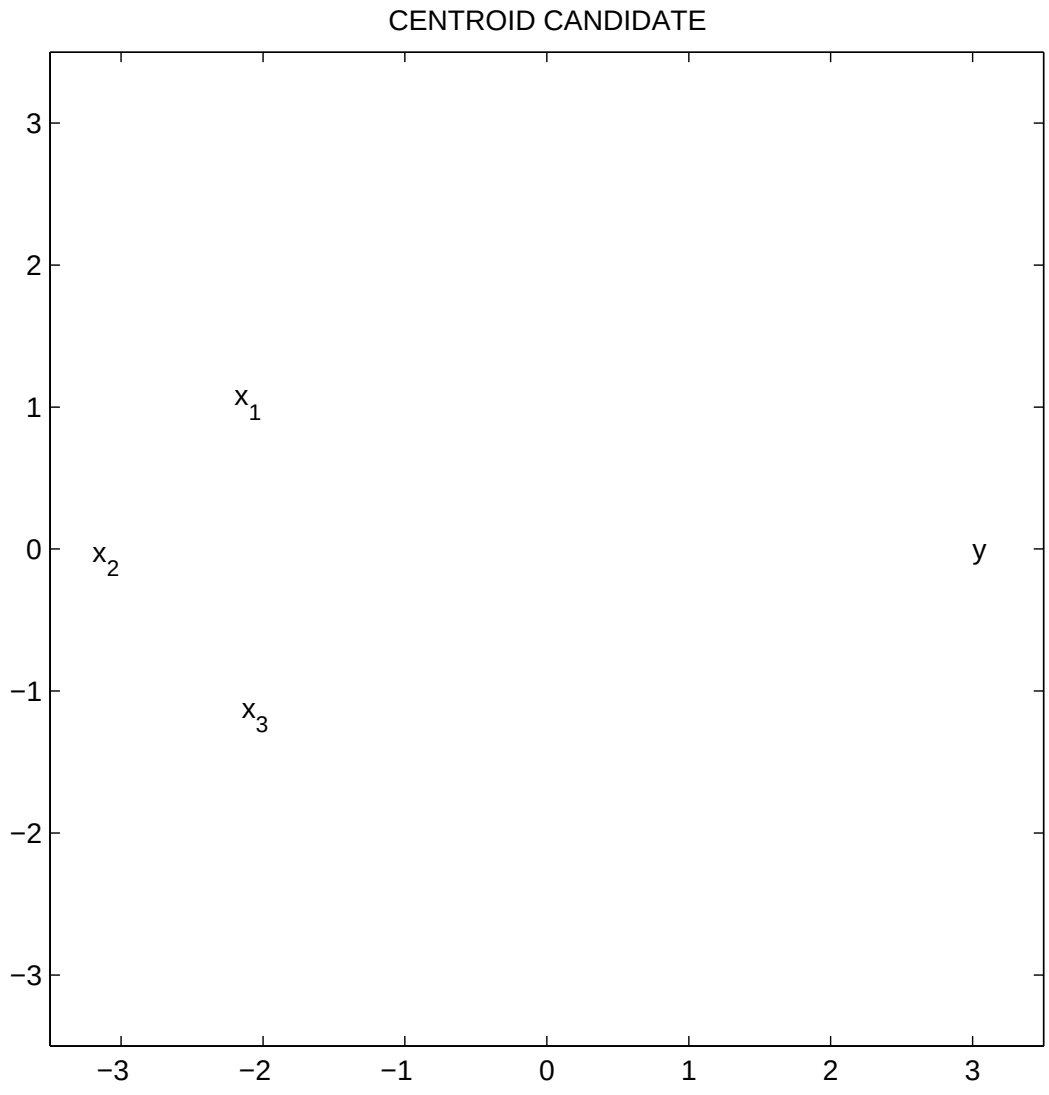
$\Pi^{(t+1)}$



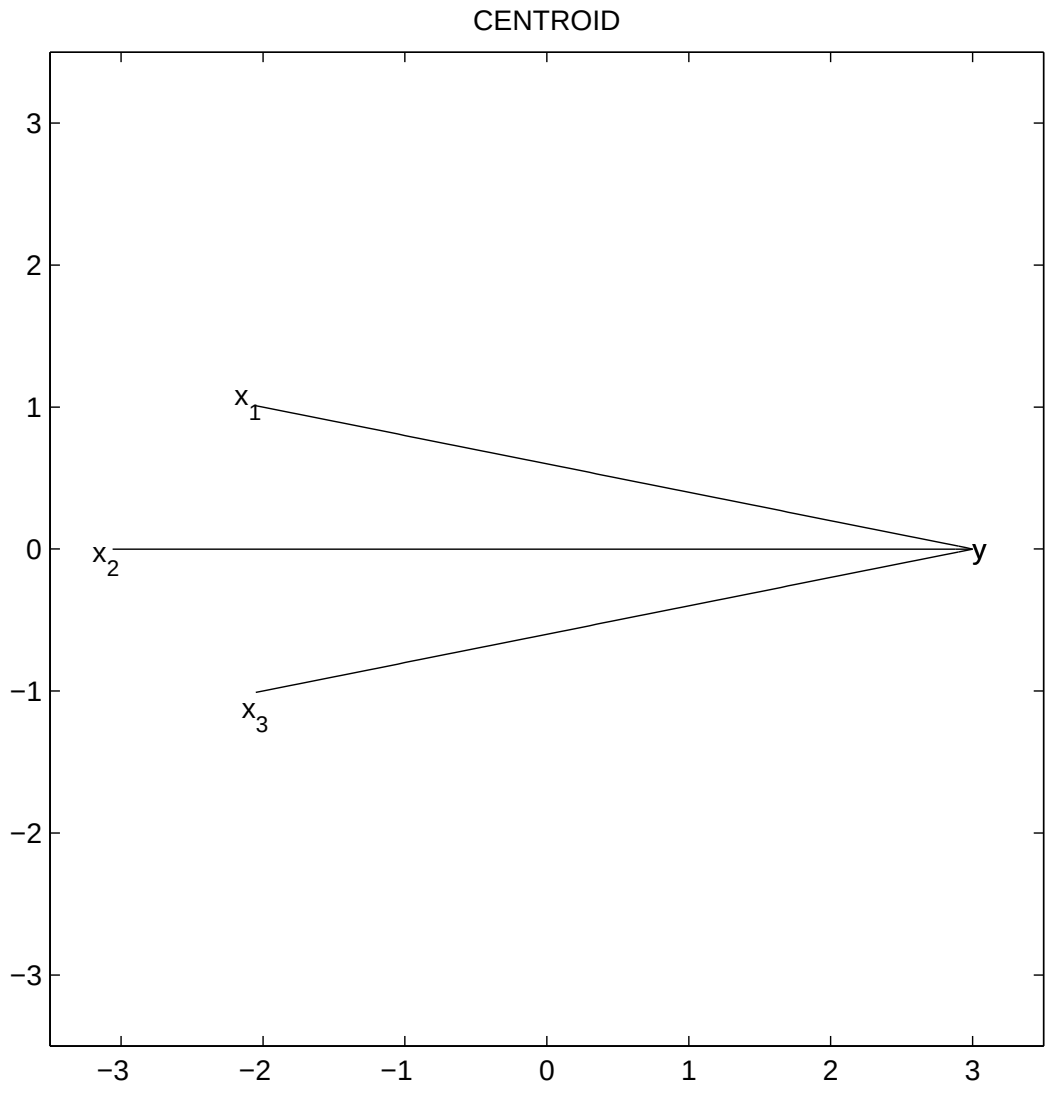
Centroid



Centroid



Centroid



$$f(\mathbf{y}) = d(\mathbf{y}, \mathbf{x}_1) + d(\mathbf{y}, \mathbf{x}_2) + d(\mathbf{y}, \mathbf{x}_3)$$

Distance-like function

$d(\mathbf{y}, \mathbf{x})$ and q can be associated.

The relation between q and d can be defined through a centroid $\mathbf{c}$ of a cluster π

$$\mathbf{c} = \mathbf{c}(\pi) = \arg\min \left\{ \sum_{\mathbf{x} \in \pi} d(\mathbf{y}, \mathbf{x}),\ \mathbf{y} \in \mathbf{C} \right\}.$$

If $q(\pi)$ is defined as $\sum_{\mathbf{x} \in \pi} d(\mathbf{c}(\pi), \mathbf{x})$, then centroids and partitions can be associated.

Centroid-partition association

1. For a set of centroids $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ define a partition $\{\pi_1, \ldots, \pi_k\}$ of the set $\mathbf{X}$ by:

$$\pi_i = \{\mathbf{x} \mid d(\mathbf{c}_i, \mathbf{x}) \leq d(\mathbf{c}_j, \mathbf{x}) \text{ for each } j \neq i\}$$

   (we break ties arbitrarily).

2. Given a partition $\{\pi_1, \ldots, \pi_k\}$ of the set $\mathbf{X}$ define the corresponding centroids $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ by

$$\mathbf{c}_i = \arg\min \left\{ \sum_{\mathbf{x} \in \pi_i} d(\mathbf{y}, \mathbf{x}),\ \mathbf{y} \in \mathbf{C} \right\}.$$

Example

"distance-like" function

$$d(\mathbf{y}, \mathbf{x}) = \|\mathbf{x} - \mathbf{y}\|^2 \text{ and } \mathbf{C} = \mathbf{R}^n$$

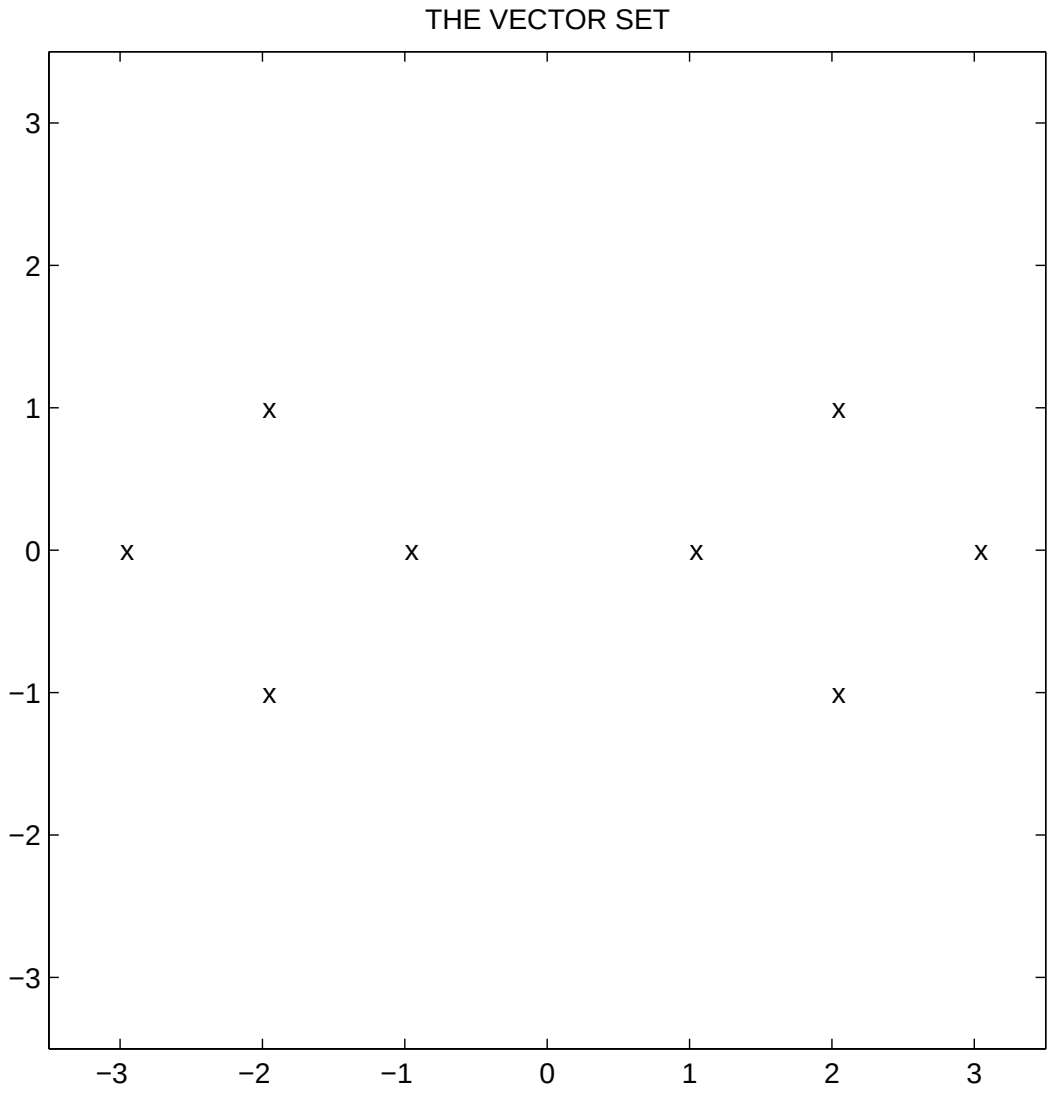
if $\pi = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$, then

$$\mathbf{c} = \arg\min \left\{ \sum_{\mathbf{x} \in \pi} d(\mathbf{y}, \mathbf{x}),\ \mathbf{y} \in \mathbf{C} \right\} = \frac{1}{l} \sum_{i=1}^{l} \mathbf{x}_i$$

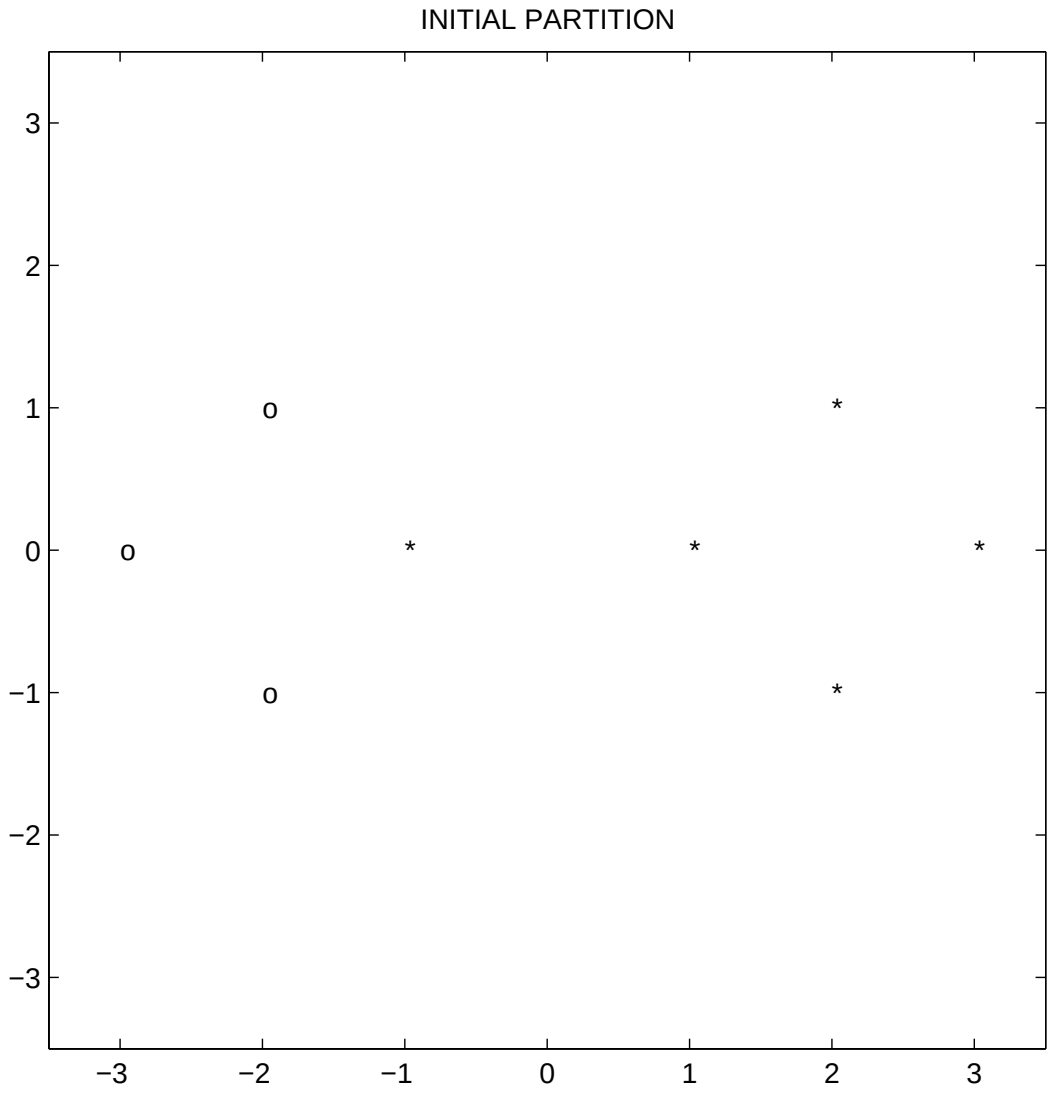
and

$$q(\pi) = \sum_{\mathbf{x} \in \pi} \|\mathbf{x} - \mathbf{c}\|^2$$

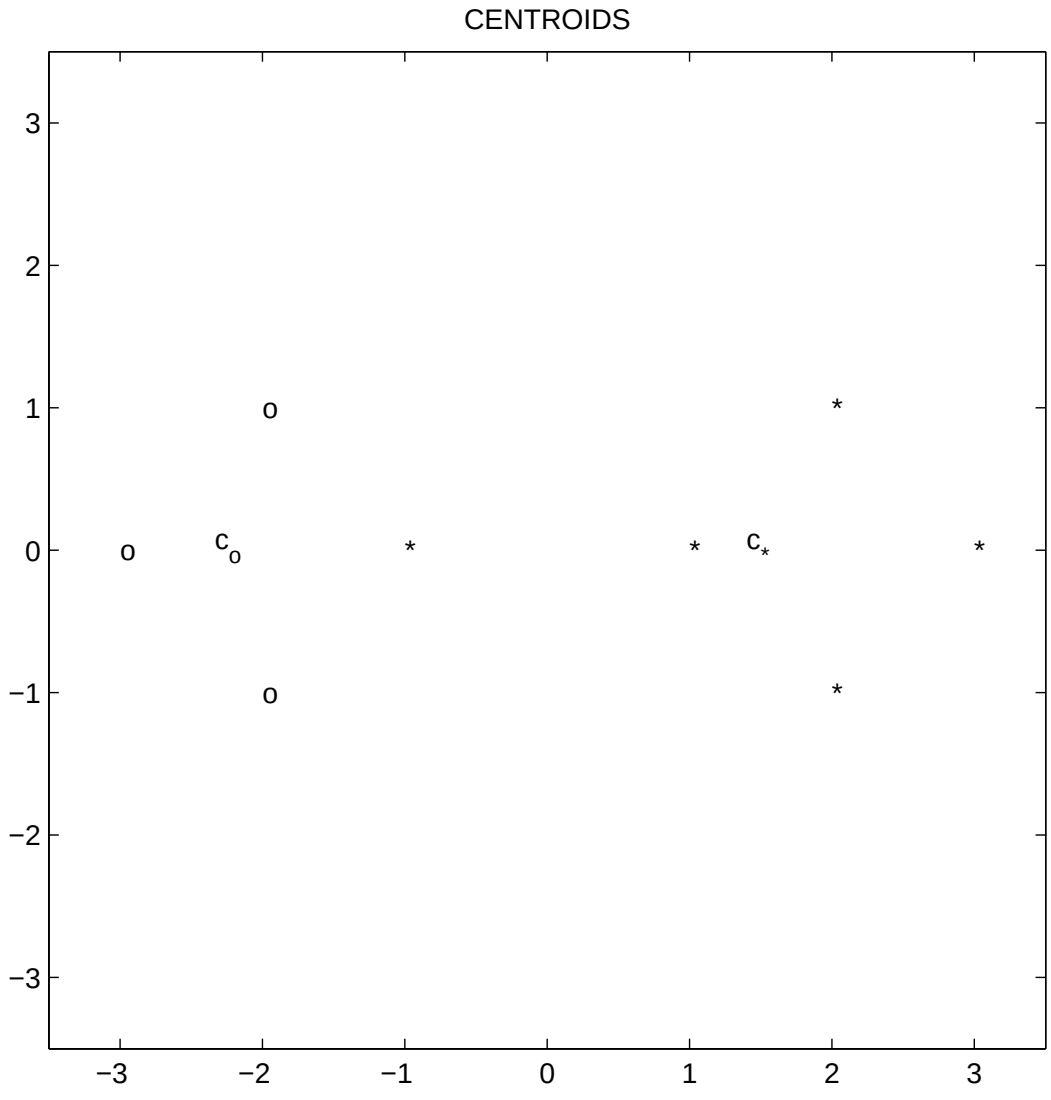
Example



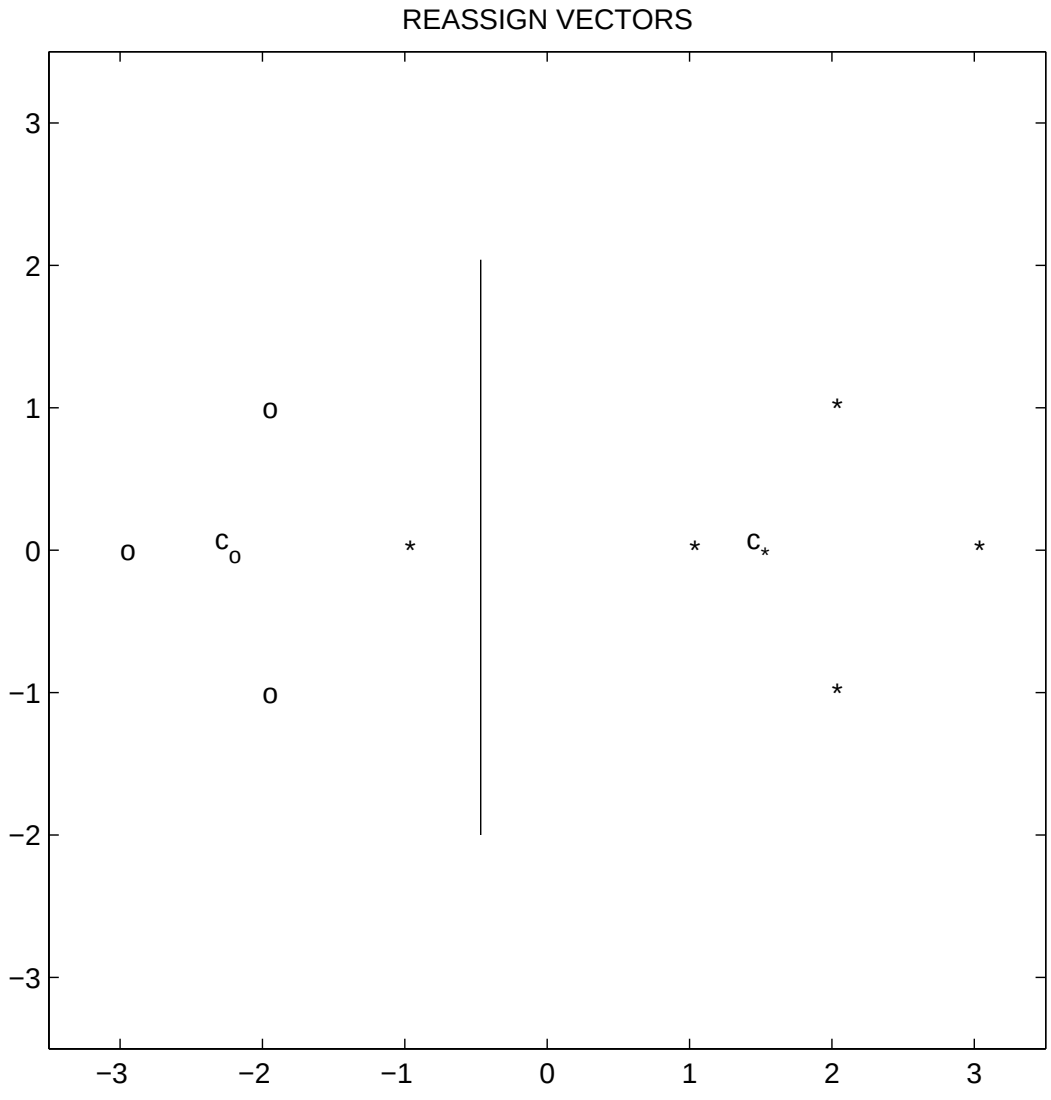
Example



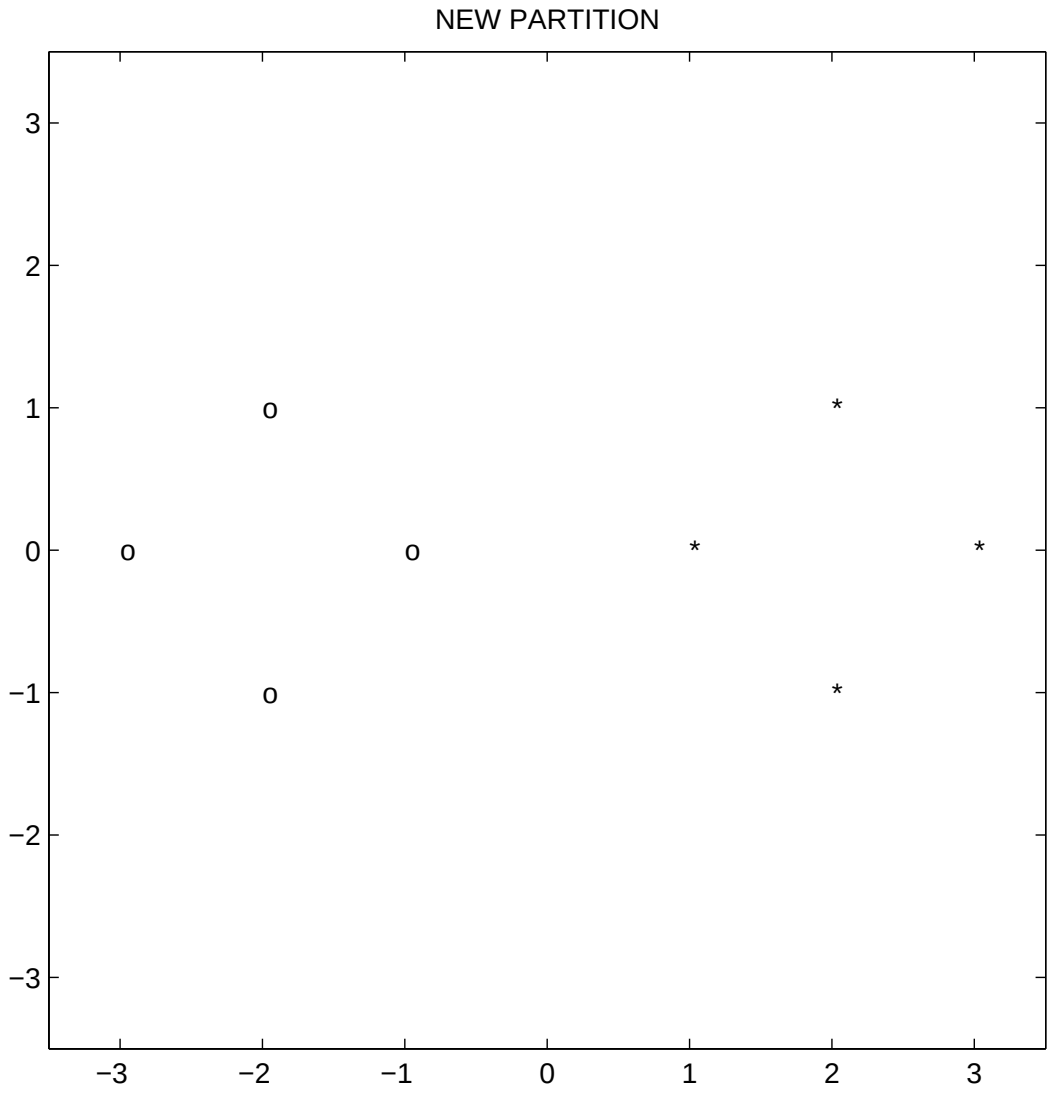
Example



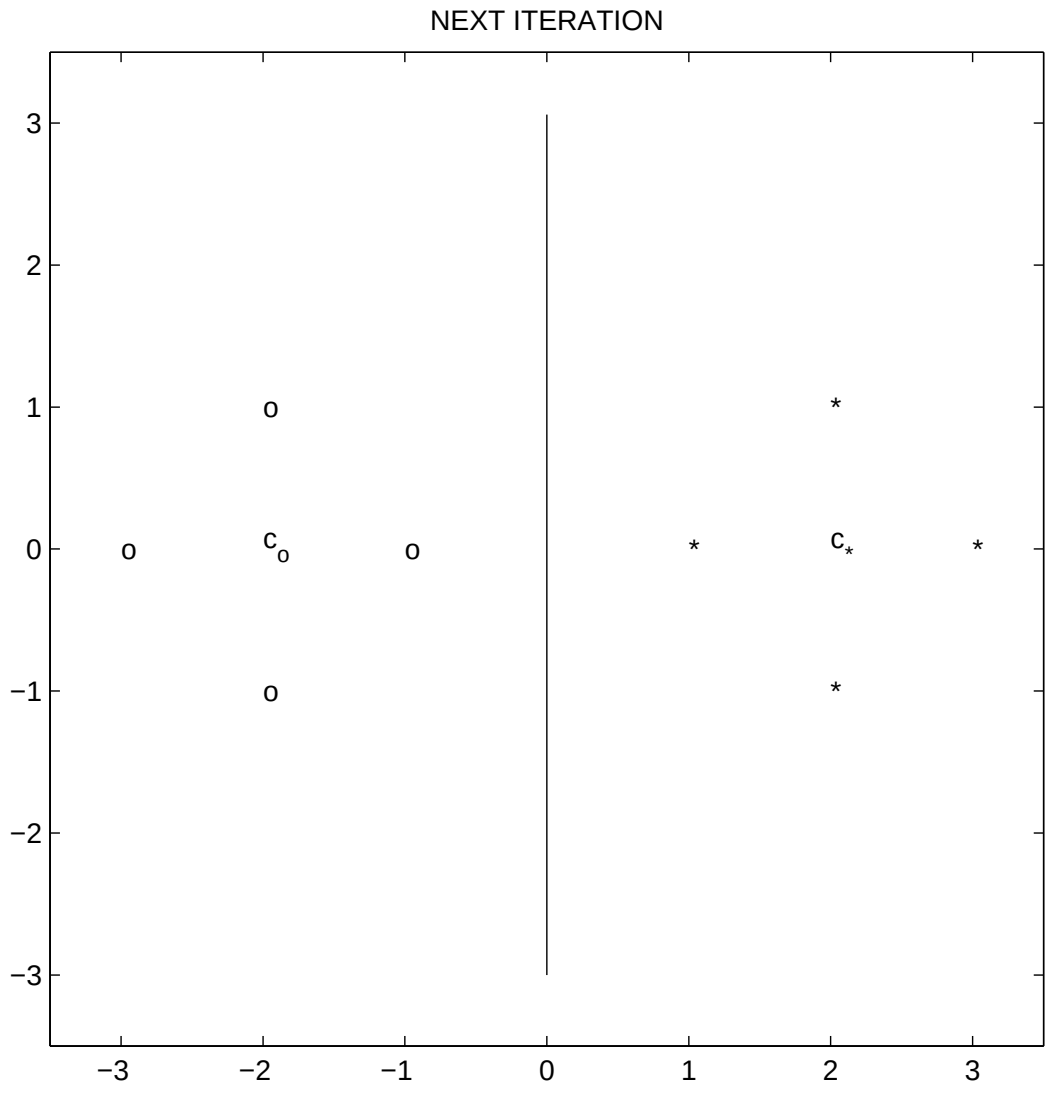
Example



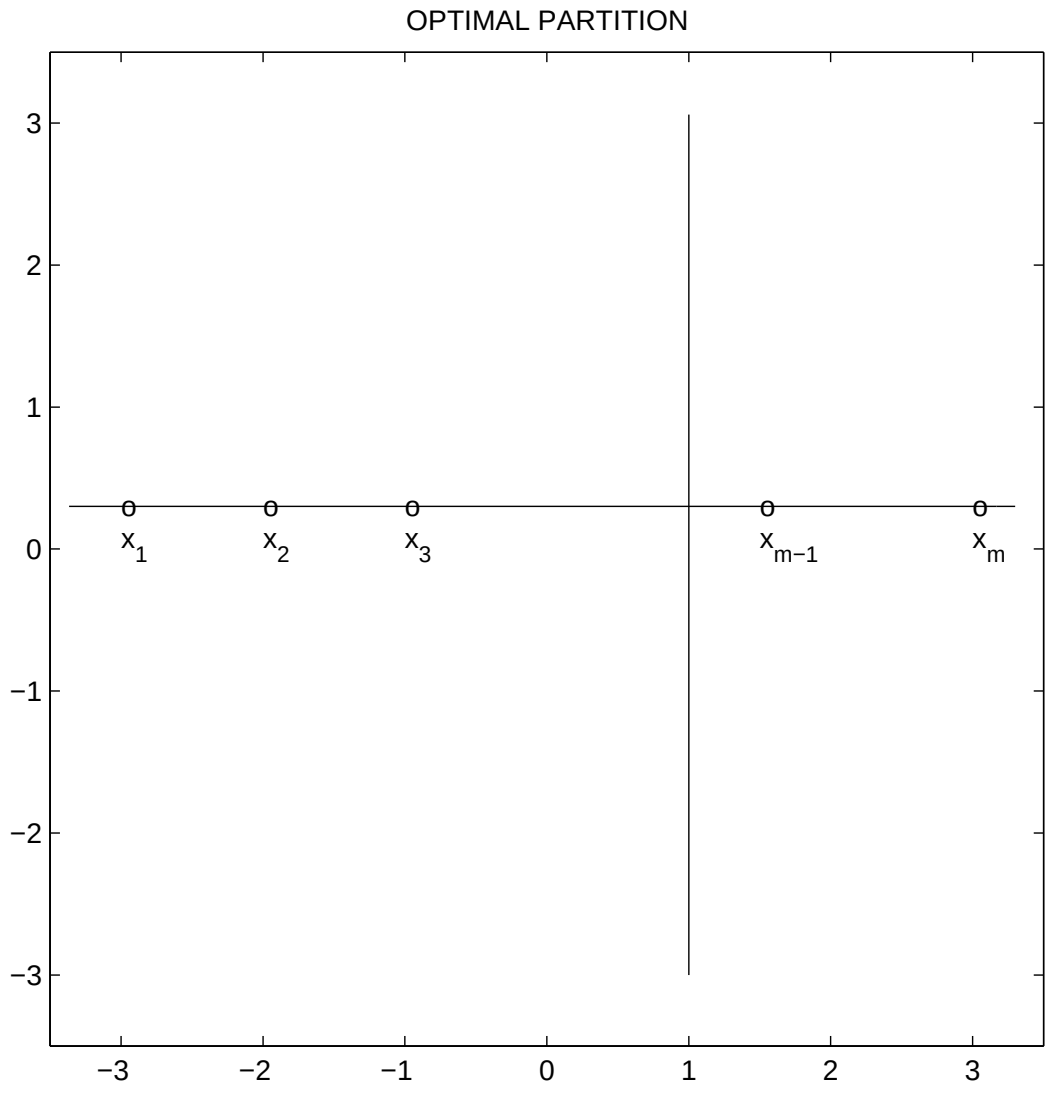
Example



convexity of final partition



optimal 1D two cluster partition



Batch k-means like algorithms

1. For a set of centroids $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ define a partition $\{\pi_1, \ldots, \pi_k\}$ of the set $\mathbf{X}$ by:

$$\pi_i = \left\{\mathbf{x} \mid \|\mathbf{c}_i - \mathbf{x}\|^2 \leq \|\mathbf{c}_j - \mathbf{x}\|^2 \text{ for each } j \neq i\right\}$$

   (we break ties arbitrarily).

2. Given a partition $\{\pi_1, \ldots, \pi_k\}$ of the set $\mathbf{X}$ define the corresponding centroids $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ by

$$\mathbf{c}_i = \arg\min \left\{ \sum_{\mathbf{x} \in \pi_i} \|\mathbf{y} - \mathbf{x}\|^2,\ \mathbf{y} \in \mathbf{R}^n \right\}.$$

Deficiencies

- k-the “right" number of clusters should be supplied,
- the initial partition

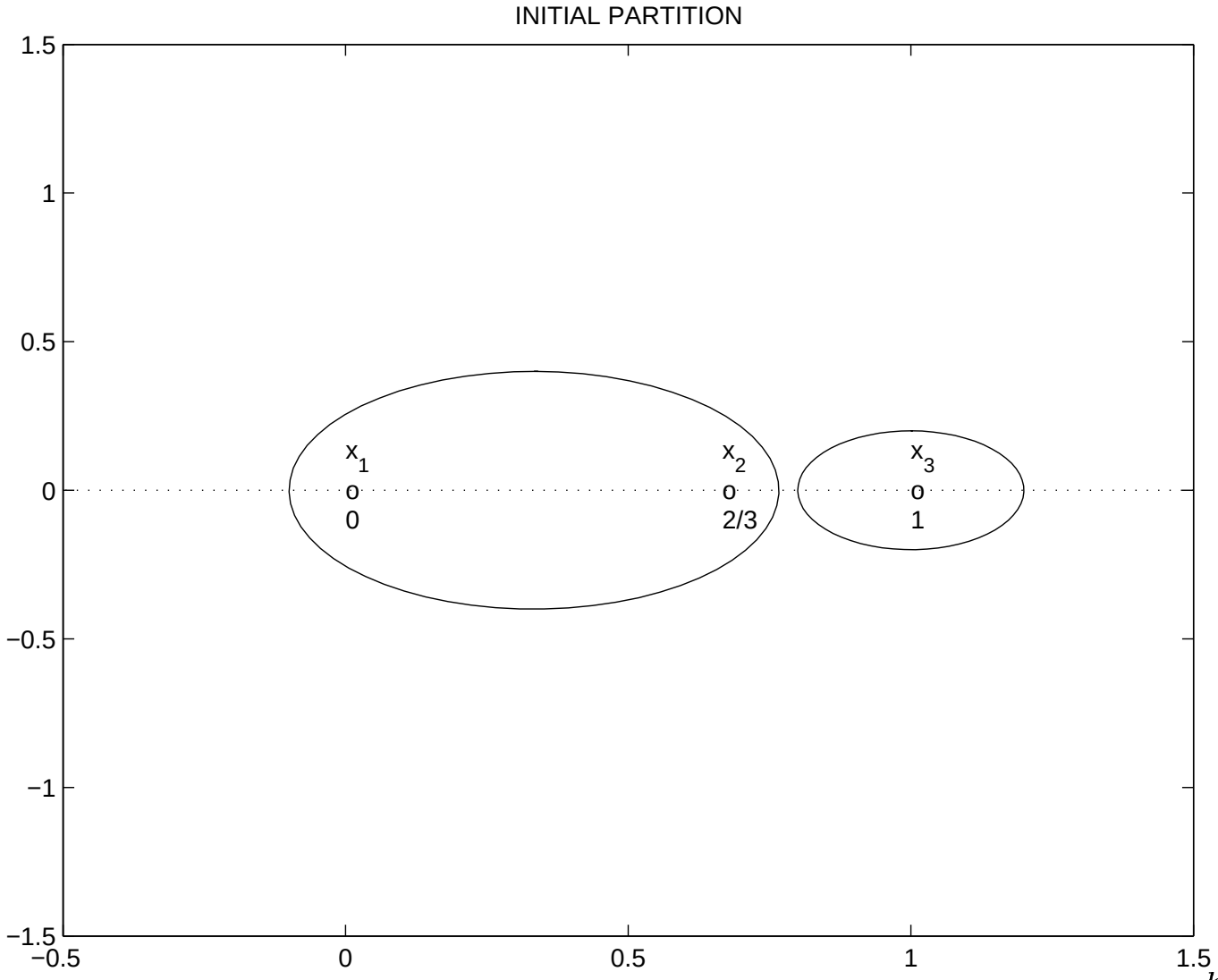
$$\Pi^{(0)} = \left\{ \pi_1^{(0)}, \ldots, \pi_k^{(0)} \right\}$$

should be supplied,

- the batch k–means often gets trapped at a local minimum.

Incremental k–means

$$\mathbf{X} = \left\{0, \tfrac{2}{3}, 1\right\},\ \pi_1^{(0)} = \left\{0, \tfrac{2}{3}\right\},\ \pi_2^{(0)} = \{1\}.$$



Enhanced k-means algorithm

1. Set $t = 0$.
2. Start with an arbitrary partitioning $\Pi^{(t)} = \left\{\pi_1^{(t)}, \ldots, \pi_k^{(t)}\right\}$.
3. Run batch k-means until no vector movement is detected.
4. Run one iteration of incremental k-means.
   if (vector movement is detected) go to Step 3.
5. Stop.

Cost of incremental step

The decision whether a vector $\mathbf{x} \in \pi_i$ should be moved from cluster π_i with m_i vectors to cluster π_j with m_j vectors made by the batch k–means algorithm based on the sign of

$$\Delta = -\left\|\mathbf{x} - \mathbf{c}(\pi_i)\right\|^2 + \left\|\mathbf{x} - \mathbf{c}(\pi_j)\right\|^2 .$$

The vector $\mathbf{x}$ is moved by the batch k–means algorithm if $\Delta < 0$.

The exact change in the value of the objective function caused by the move is

$$\Delta_{\text{exact}} = -\frac{m_i}{m_i - 1}\left\|\mathbf{x} - \mathbf{c}(\pi_i)\right\|^2 + \frac{m_j}{m_j + 1}\left\|\mathbf{x} - \mathbf{c}(\pi_j)\right\|^2 .$$

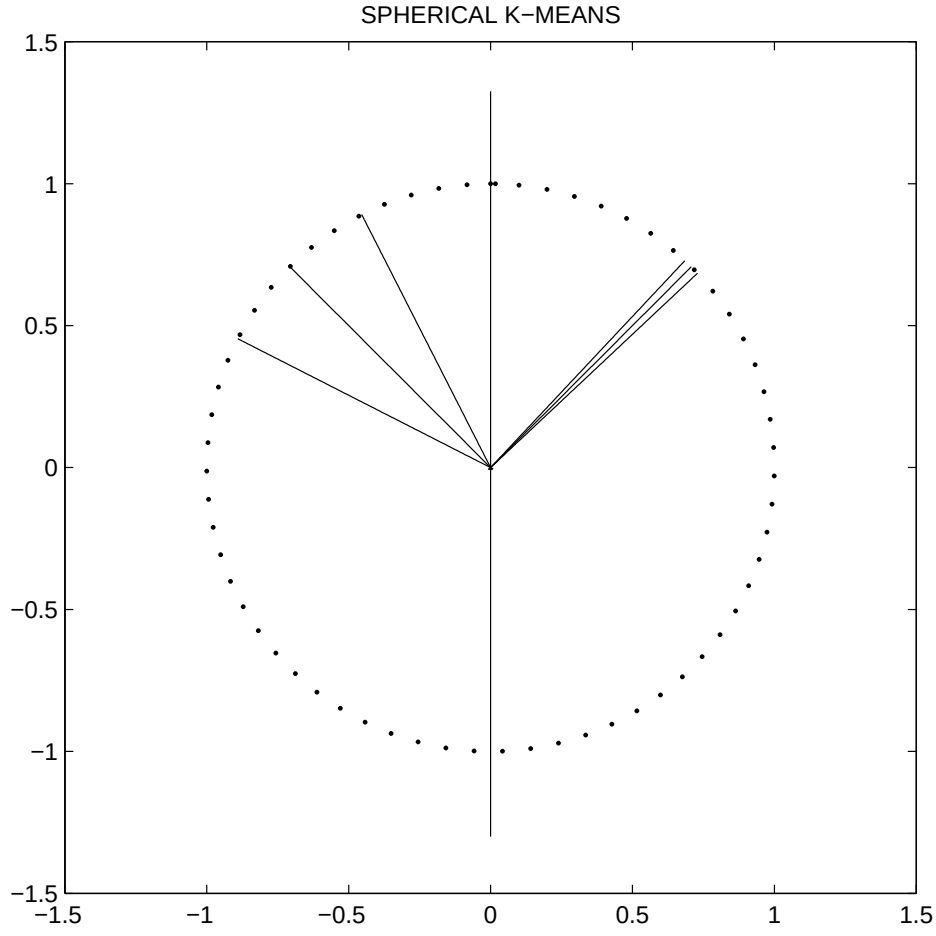
"distance–like" functions

A vector $\mathbf{x} = (\mathbf{x}[1], \ldots, \mathbf{x}[n])^T \in \mathbf{R}^n$.

- $d(\mathbf{c}, \mathbf{x}) = \|\mathbf{c} - \mathbf{x}\|^2$
- $d(\mathbf{c}, \mathbf{x}) = \mathbf{c}^T \mathbf{x}$
- $d(\mathbf{c}, \mathbf{x}) = \sum_{j=1}^{n} \left[\mathbf{x}[j] \log \frac{\mathbf{x}[j]}{\mathbf{c}[j]} + \mathbf{c}[j] - \mathbf{x}[j] \right]$
- $d(\mathbf{c}, \mathbf{x}) = \sum_{j=1}^{n} \left[c[j] \log \frac{\mathbf{c}[j]}{\mathbf{x}[j]} + \mathbf{x}[j] - \mathbf{c}[j] \right]$

Spherical k–means

$$\Pi = \{\pi_1, \ldots, \pi_k\},\ q(\pi) = \left\| \sum_{\mathbf{x} \in \pi} \mathbf{x} \right\|$$



optimal 1D two cluster partition

when the data belongs to $\mathbf{S}^1$ the optimal two cluster partition can be obtained by splitting the circle $\mathbf{S}^1$ into two semi-circles by a line passing through the origin

data

- DC0 (Medlars Collection, 1033 medical abstracts).
- DC1 (CISI Collection, 1460 information science abstracts).
- DC2 (Cranfield Collection, 1400 aerodynamics abstracts).

	DC0	DC1	DC2
cluster 0	1004	5	4
cluster 1	18	1440	16
cluster 2	11	15	1380

69 “misclassified" documents using **4099** terms

Average document

The Pythagorean Theorem employed **24** words,
the Lord's Prayer has **66** words,
Archimedes Principle has **67** words,
the 10 Commandments have **179** words,
the Gettysburg Address had **286** words,
the Declaration of Independence has **1,300** words

and finally

the European Commission's regulation on the sale of cabbage: **26,911** words.

what do we want to do:

- to use the same data
- to select SMALLER set of terms (and to reduce the dimensionality of the problem)
- to apply a hybrid clustering scheme
  (a sequence of clustering algorithms so that the output of algorithm i becomes the input of algorithm $i + 1$).

and to get better clustering results

PDDP

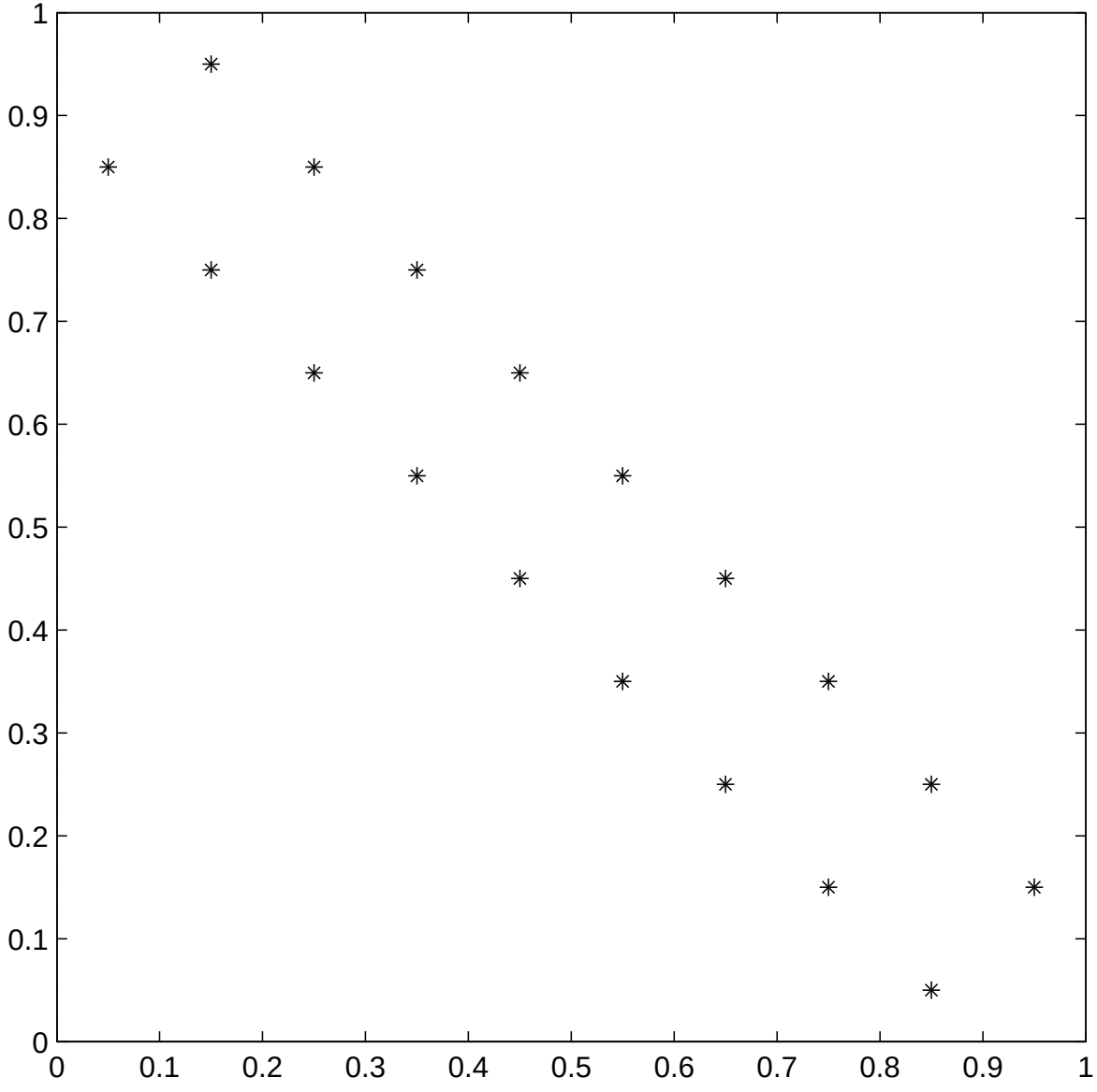
Principal Direction Divisive Partitioning

or

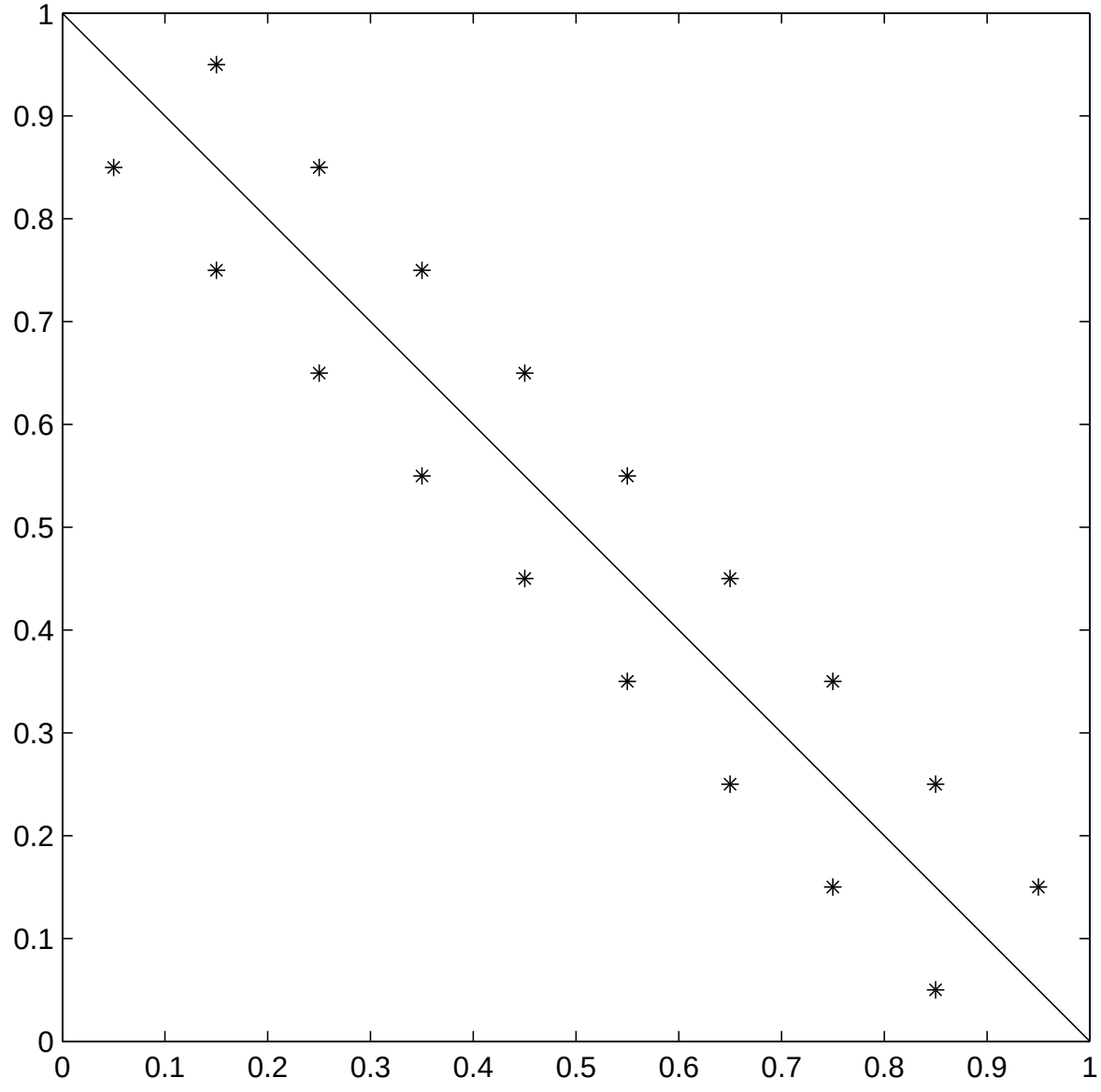
HOW TO GET
A REASONABLE INITIAL PARTITION

PDDP

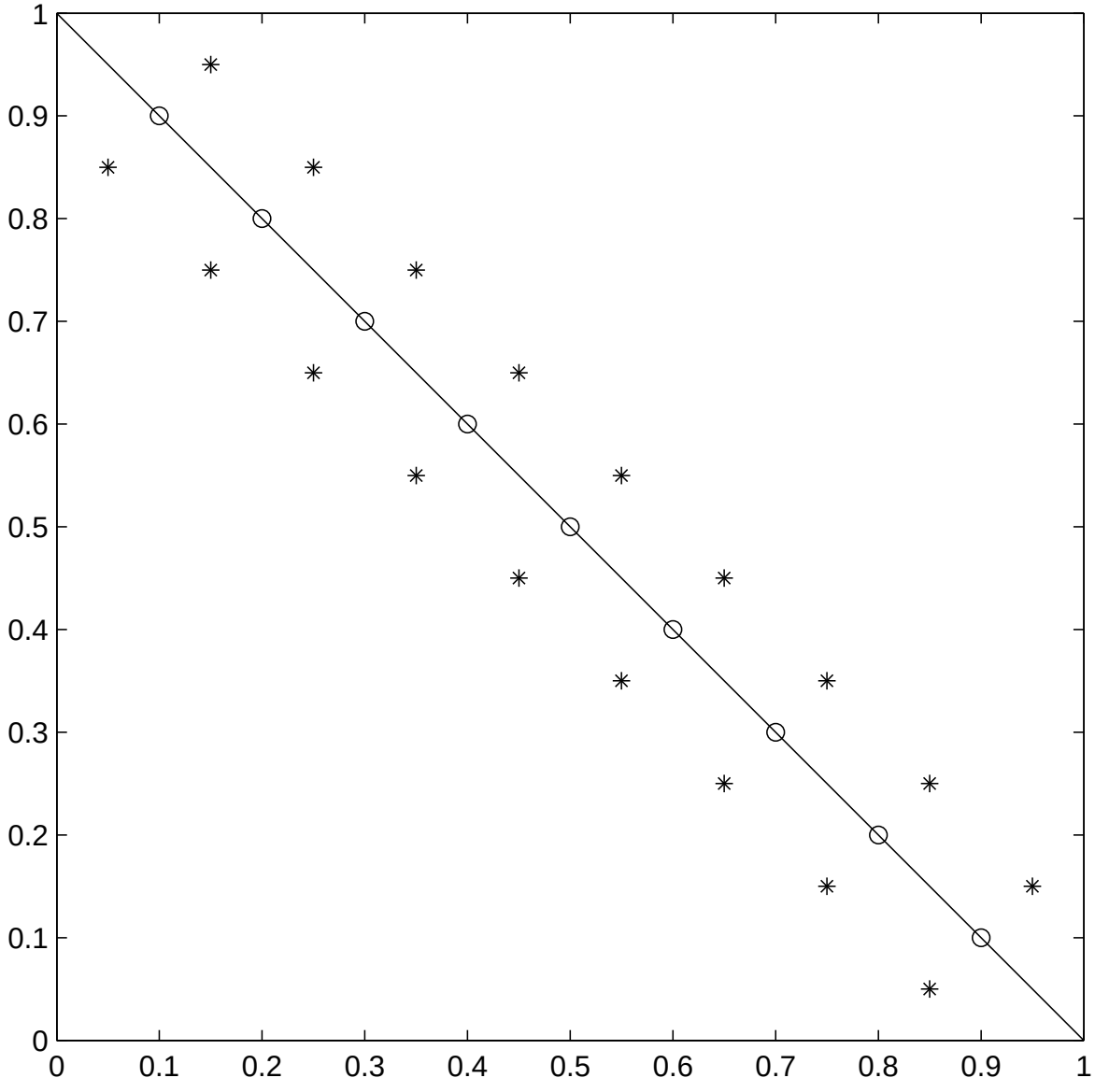
Principal Direction Divisive Partitioning



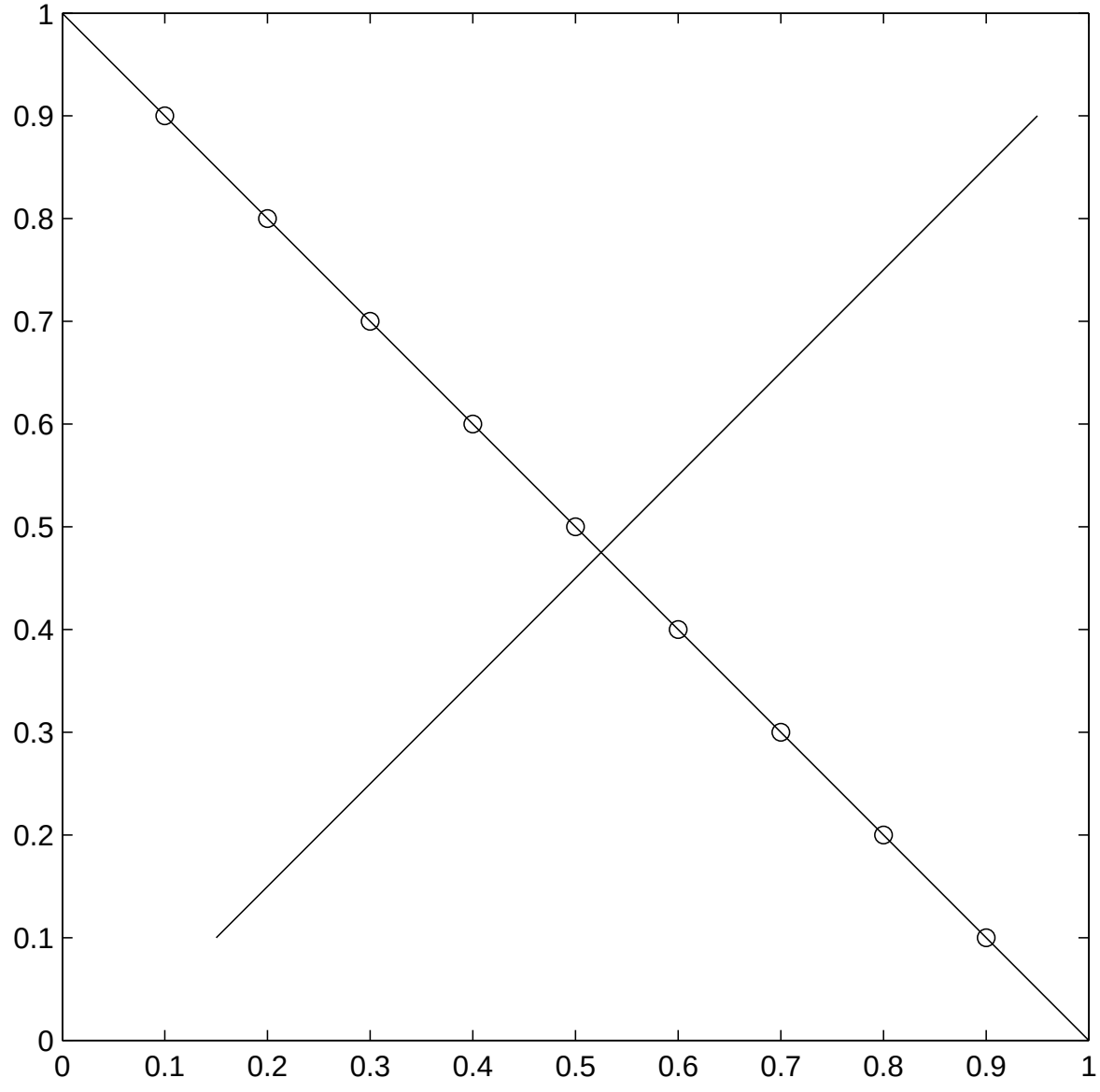
Principal Direction



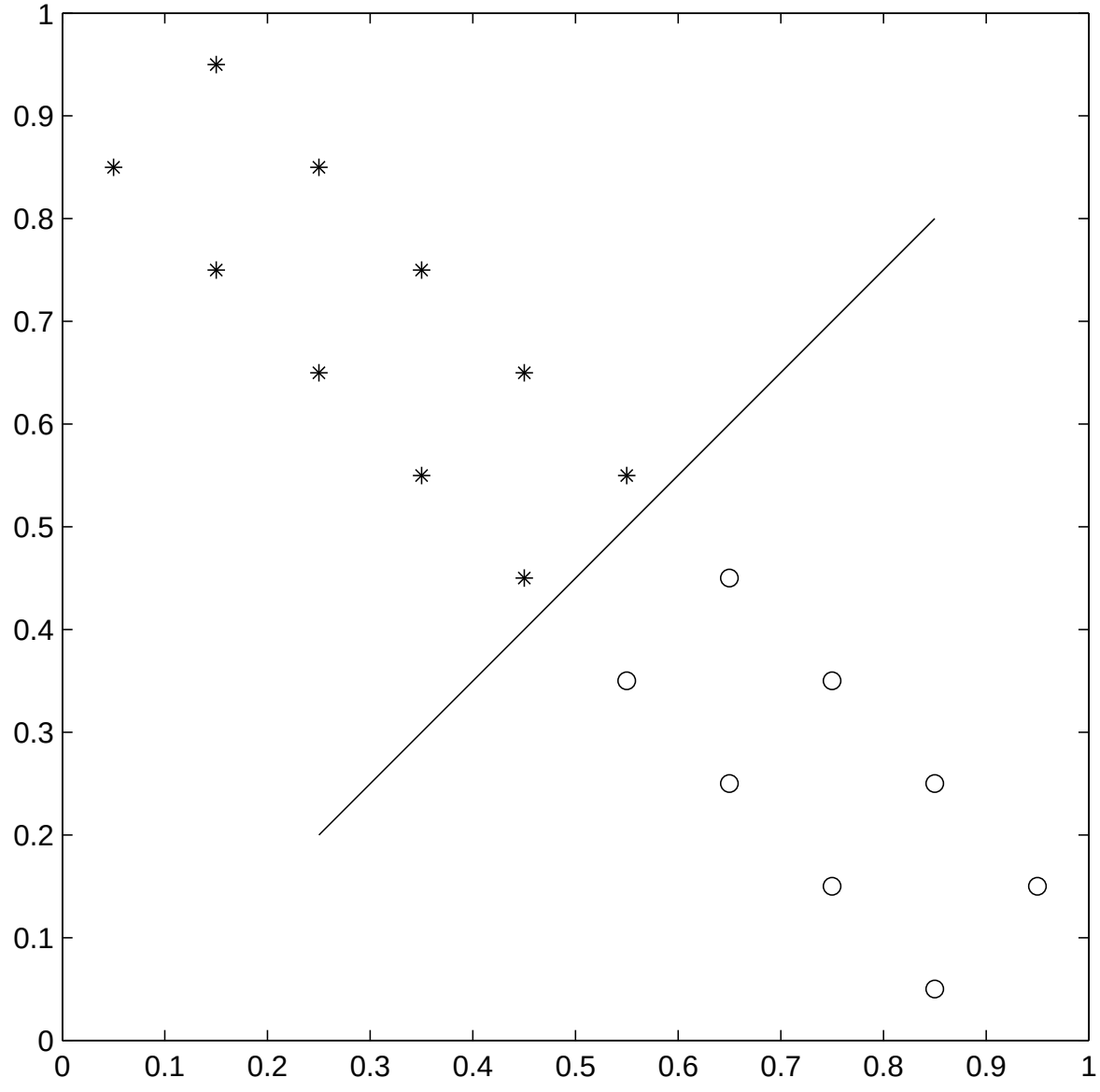
projection



partition one dimensional set



partition original set



How to find the pricipal direction

- Look at the “term by document" matrix

$$X = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \, .$$

- compute the mean $\mathbf{m} = \dfrac{\mathbf{x}_1 + \ldots + \mathbf{x}_m}{m}$
- compute the first singular vector $\mathbf{v}_1$ of the matrix

$$X - \mathbf{m}\mathbf{e}^T = [\mathbf{x}_1 - \mathbf{m}, \ldots, \mathbf{x}_m - \mathbf{m}] \, .$$

$\mathbf{v}_1$ is the pricipal direction vector

clustering results

	DC0	DC1	DC2
cluster 0	272	9	1379
cluster 1	4	1285	11
cluster 2	757	166	8
“empty" documents			
cluster 3	0	0	0

PDDP generated initial “confusion" matrix with **470** “misclassified" documents using 600 terms

constrained data

- PDDP as well as the classical k–means algorithm are general clustering algorithms capable of handling general datasets in $\mathbf{R}^n$.

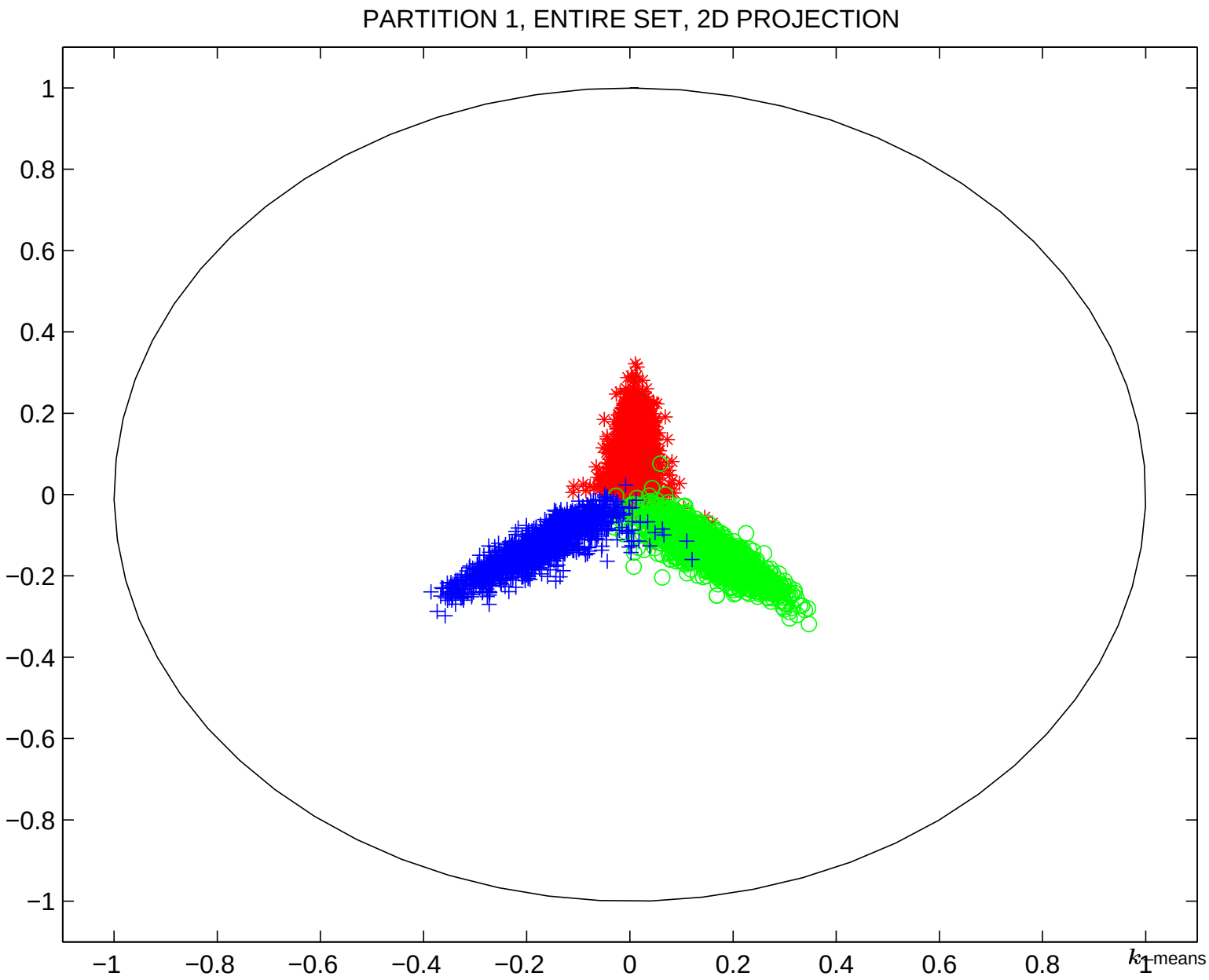
- "document–vectors" reside on an $n-1$ dimensional sphere $\mathbf{S}^{n-1}$.

sPDDP

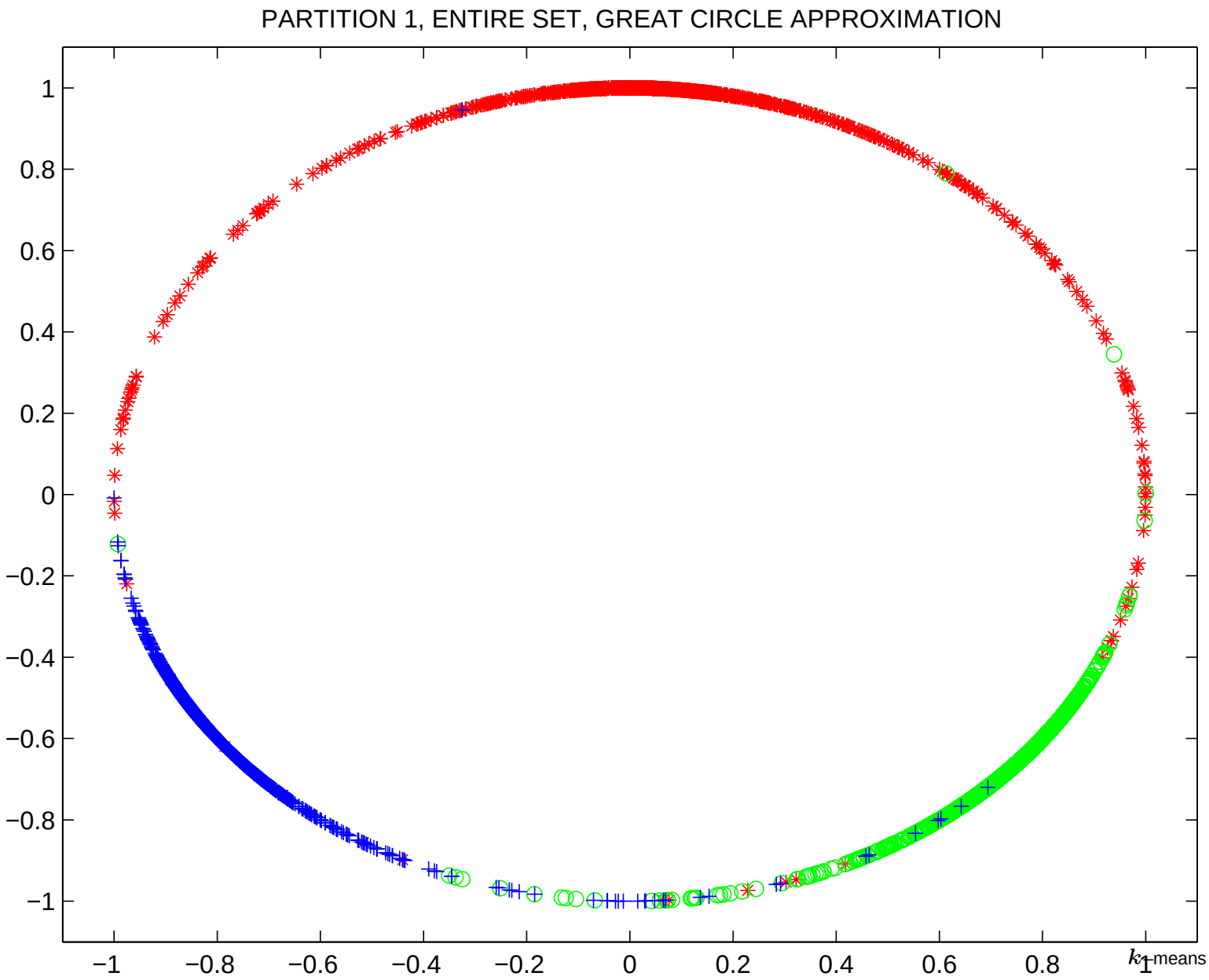
1. Given a set $\mathbf{X} \subset \mathbf{S}_2^{n-1}$ determine the two dimensional plane $\mathbf{P}$ that approximates $\mathbf{X}$ in the “best possible way".
2. Project $\mathbf{X}$ onto $\mathbf{P}$ and denote the projection by $\mathbf{Y}$.
3. “Push" $\mathbf{Y}$ to the great circle, i.e., $\mathbf{y} \rightarrow \mathbf{z} = \dfrac{\mathbf{y}}{\|\mathbf{y}\|}$.
4. Partition $\mathbf{Z} \subset \mathbf{S}_2^1$ into two clusters $\mathbf{Z}_1$ and $\mathbf{Z}_2$.
5. Generate the induced partition $\{\mathbf{X}_1, \mathbf{X}_2\}$ of $\mathbf{X}$ as follows:

$$\mathbf{X}_1 = \{\mathbf{x} \mid \mathbf{z} \in \mathbf{Z}_1\}, \text{ and } \mathbf{X}_2 = \{\mathbf{x} \mid \mathbf{z} \in \mathbf{Z}_2\}.$$

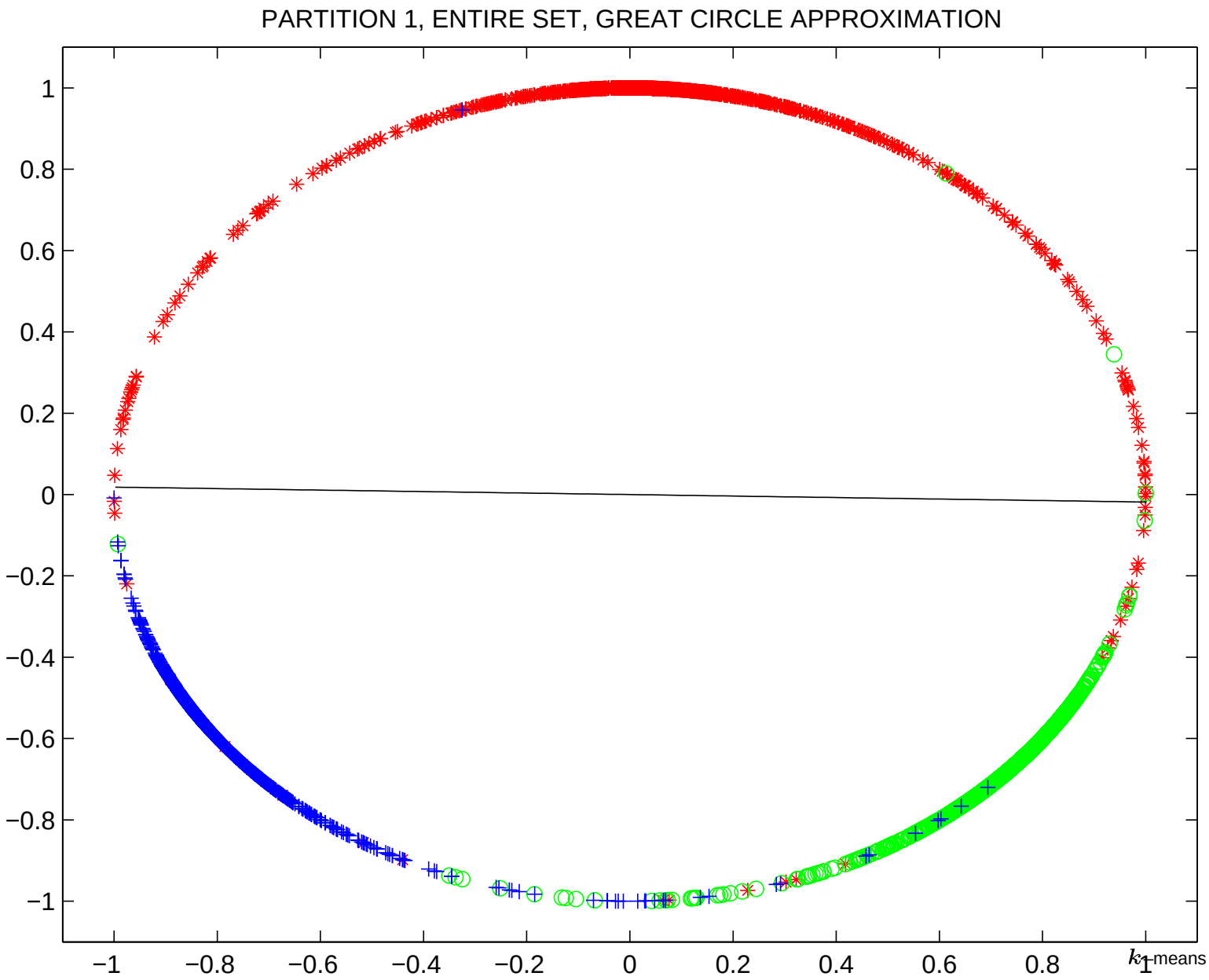
first two dimensional projection



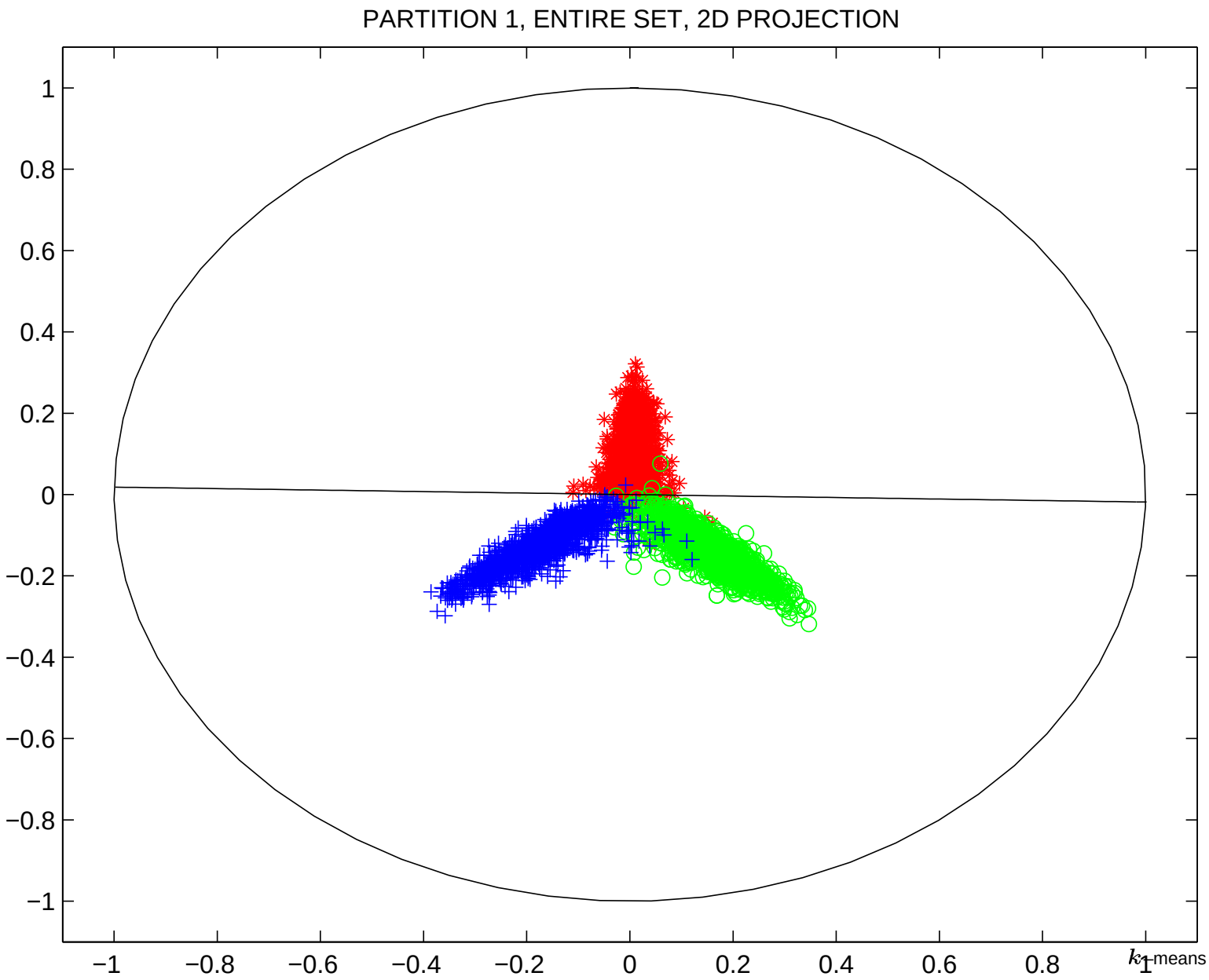
circle approximation



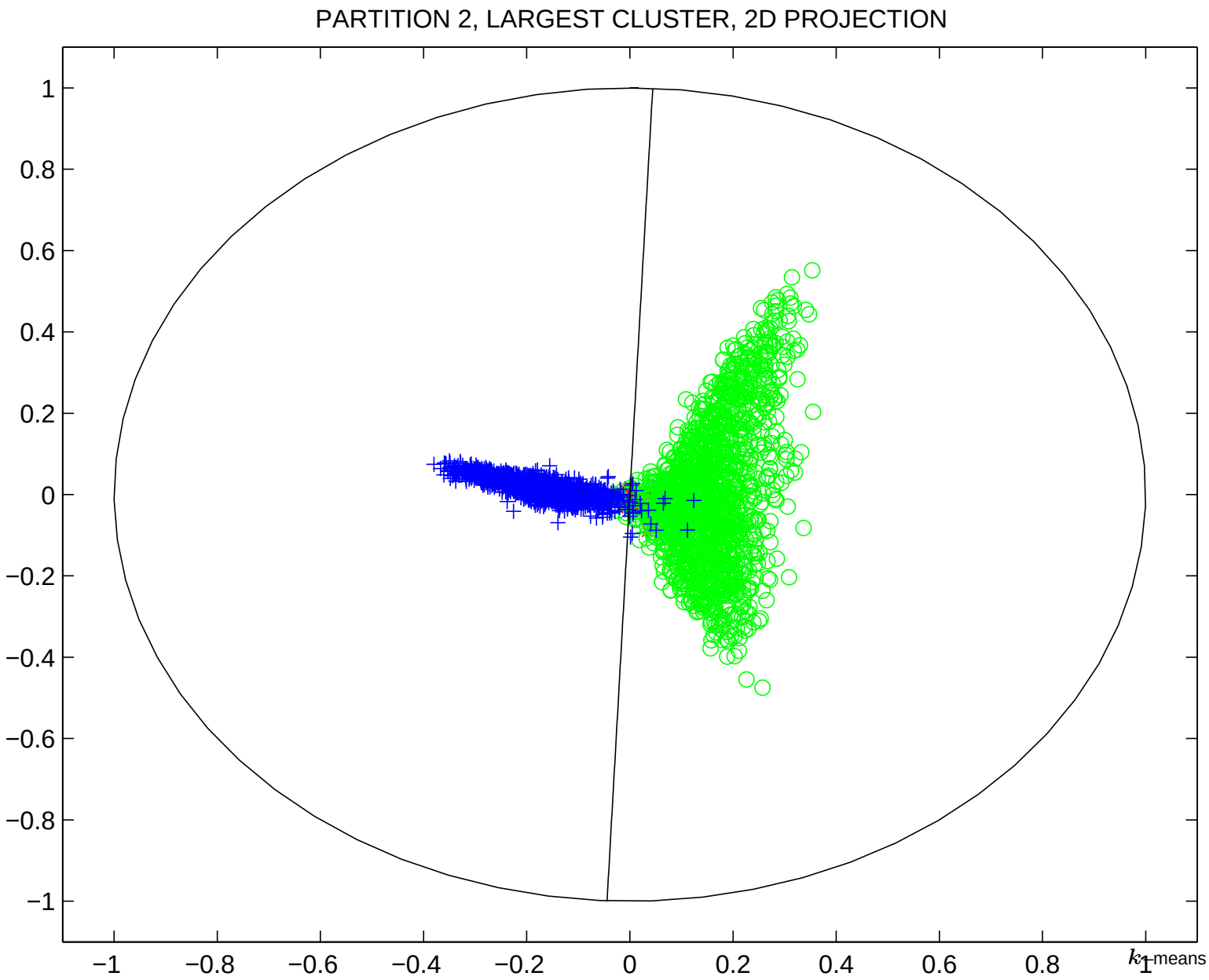
circular partition



plane partition



second plane projection



clustering results

	DC0	DC1	DC2
cluster 0	1000	3	1
cluster 1	8	10	1376
cluster 2	25	1447	21
“empty" documents			
cluster 3	0	0	0

sPDDP generated initial “confusion" matrix with **68** “misclassified" documents using 600 terms

clustering results

		documents misclassified by sPDDP			
# of terms	0 vec	alone	$+$ k–means	$+$ sph k–means	$+$ IT–means
100	12	383	258	229	**168**
200	3	277	133	143	**116**
300	0	228	100	104	**81**
400	0	88	80	78	**56**
500	0	76	62	57	**40**
600	0	68	62	54	**44**

Low Bound for $Q(\Pi)$

$\Pi = \{\pi_1, \ldots, \pi_k\}$, $|\pi_i| = m_i$

$$Y = \begin{bmatrix} \frac{\mathbf{e}_{m_1}}{\sqrt{m_1}} & \cdots & \cdots & \\ \cdots & \frac{\mathbf{e}_{m_2}}{\sqrt{m_2}} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \frac{\mathbf{e}_{m_k}}{\sqrt{m_k}} \end{bmatrix} \qquad (*)$$

$$Y^T Y = I_k$$

$$Q(\Pi) = \operatorname{trace}\left(X^T X\right) - \operatorname{trace}\left(Y^T X^T X Y\right).$$

To minimize $Q(\Pi)$ solve

$$\max\left\{\operatorname{trace}\left(Y^T X^T X Y\right) \ : \ Y \text{ is of the form } (*)\right\}.$$

Relaxed Maximization Problem

$$\max \left\{ \text{trace} \left(Y^T X^T X Y \right) \ : \ Y^T Y = I_k \right\}.$$

Theorem (Ky Fan) If H is a symmetric matrix with eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n,$$

then

$$\max_{Y^T Y = I_k} \text{trace} \left(Y^T H Y \right) = \lambda_1 + \lambda_2 + \ldots + \lambda_k.$$

Low Bound for $Q(\Pi)$

As a by-product we have

$$\begin{aligned} Q(\Pi) &\geq \operatorname{trace}\left(X^T X\right) - \max_{Y^T Y = I_k} \operatorname{trace}\left(Y^T X^T X Y\right) \\ &= \sum_{i=k+1}^{\min\{m,n\}} \sigma_i^2(X). \end{aligned}$$