Parallel Sparse K-Means

Application to document clustering

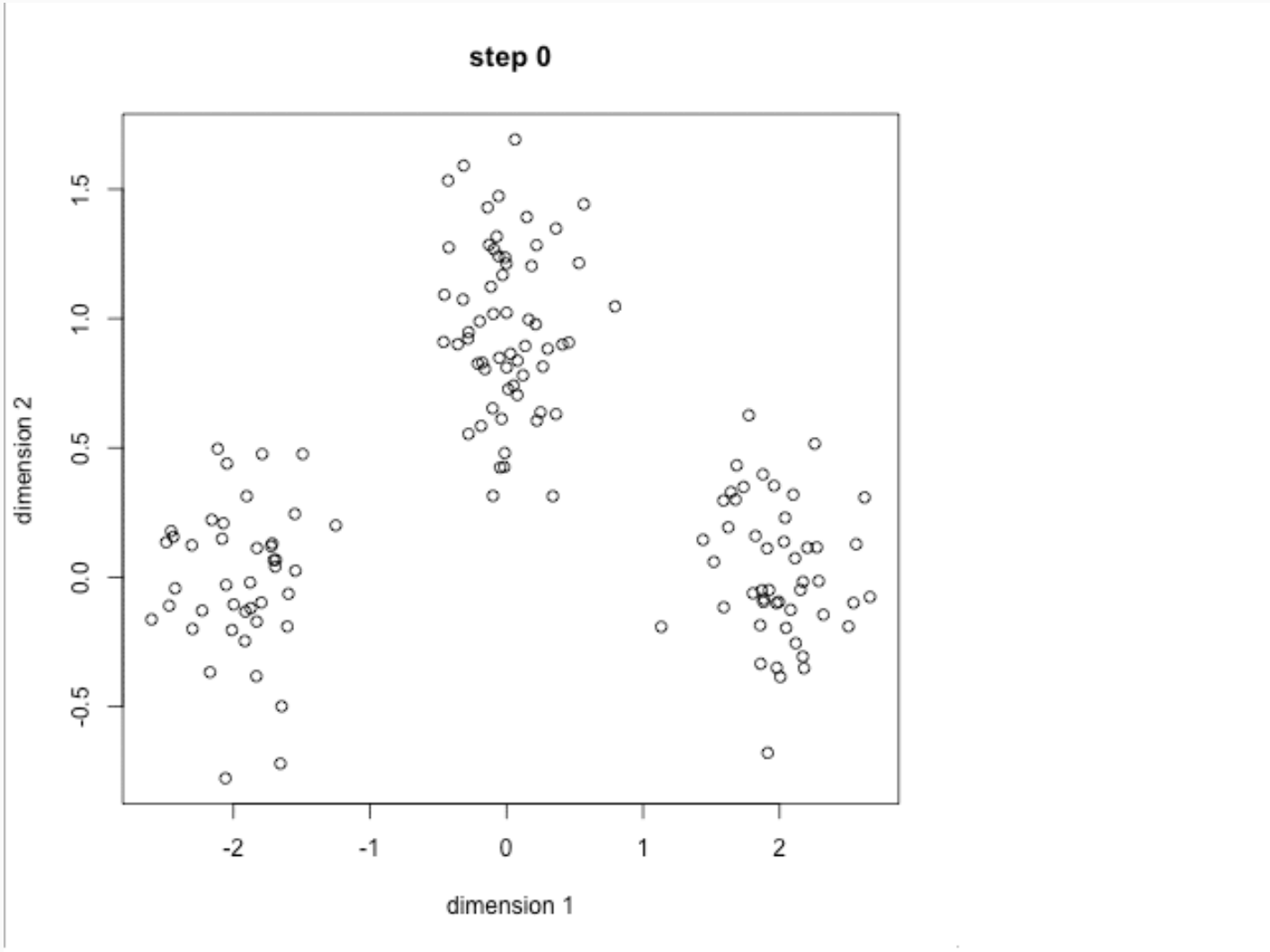
Victor Storchan

May 31, 2016

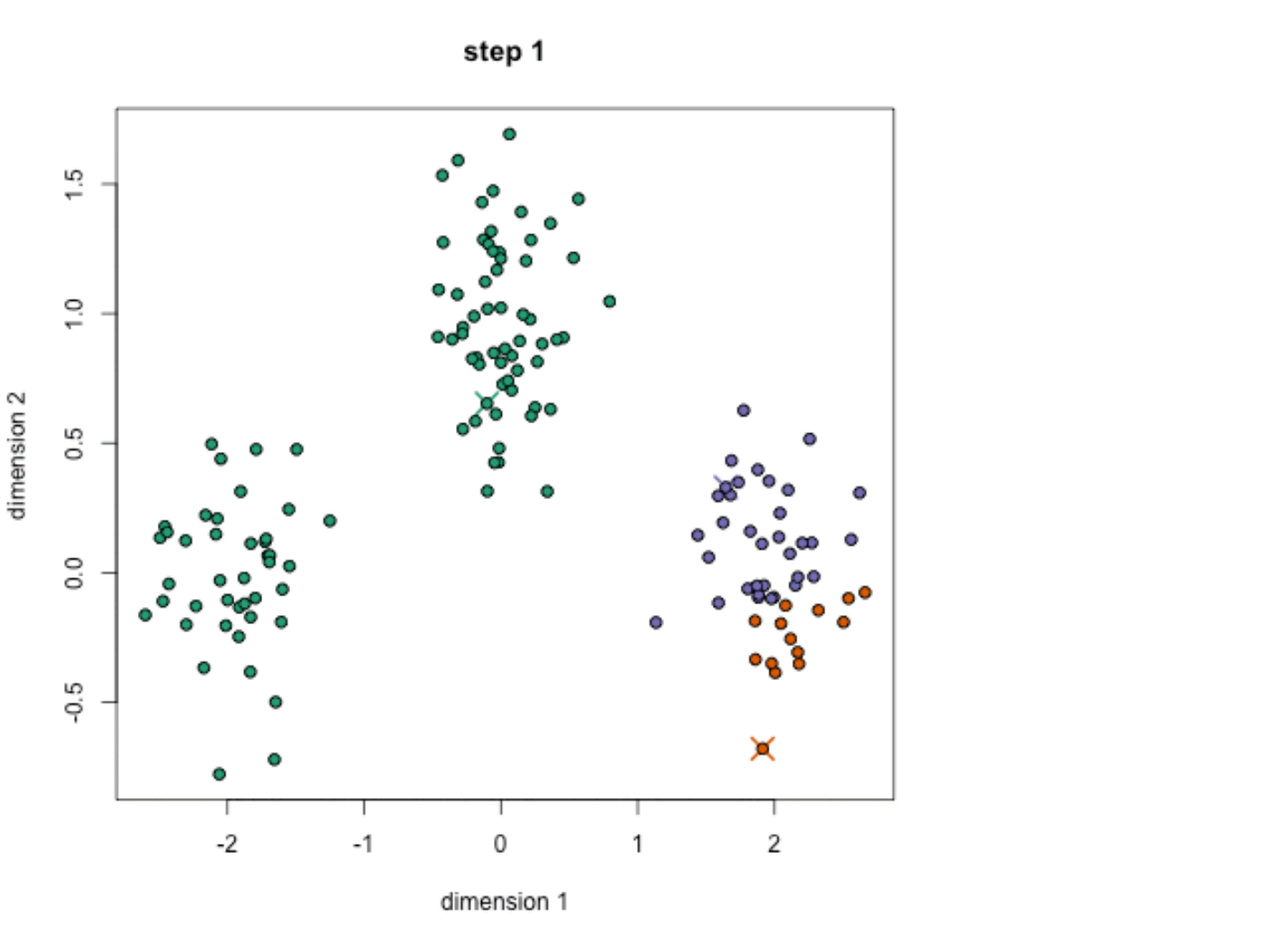
Stanford University, CME 323 (R.Zadeh)

Introduction

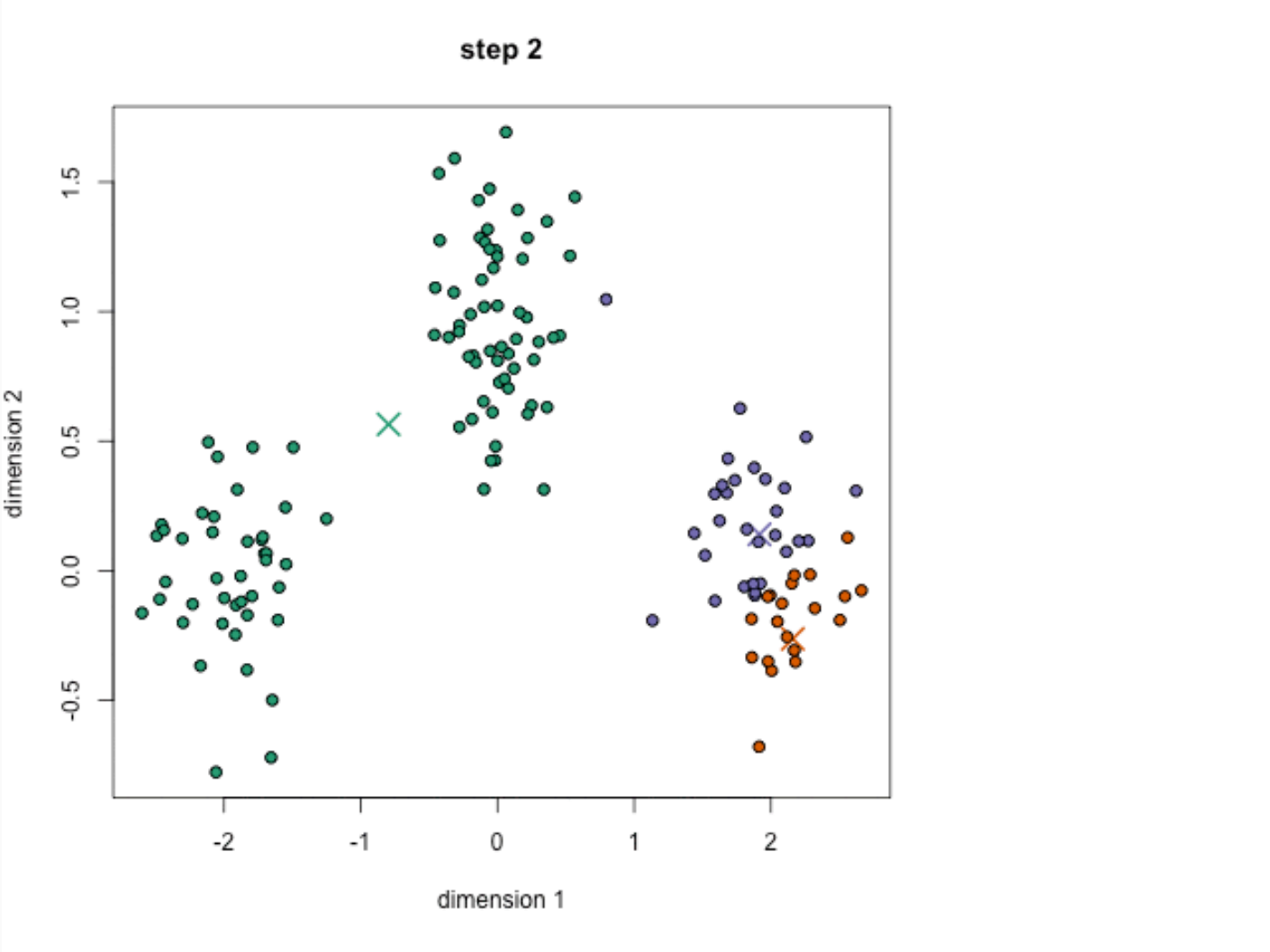
Introduction: K-Means in action



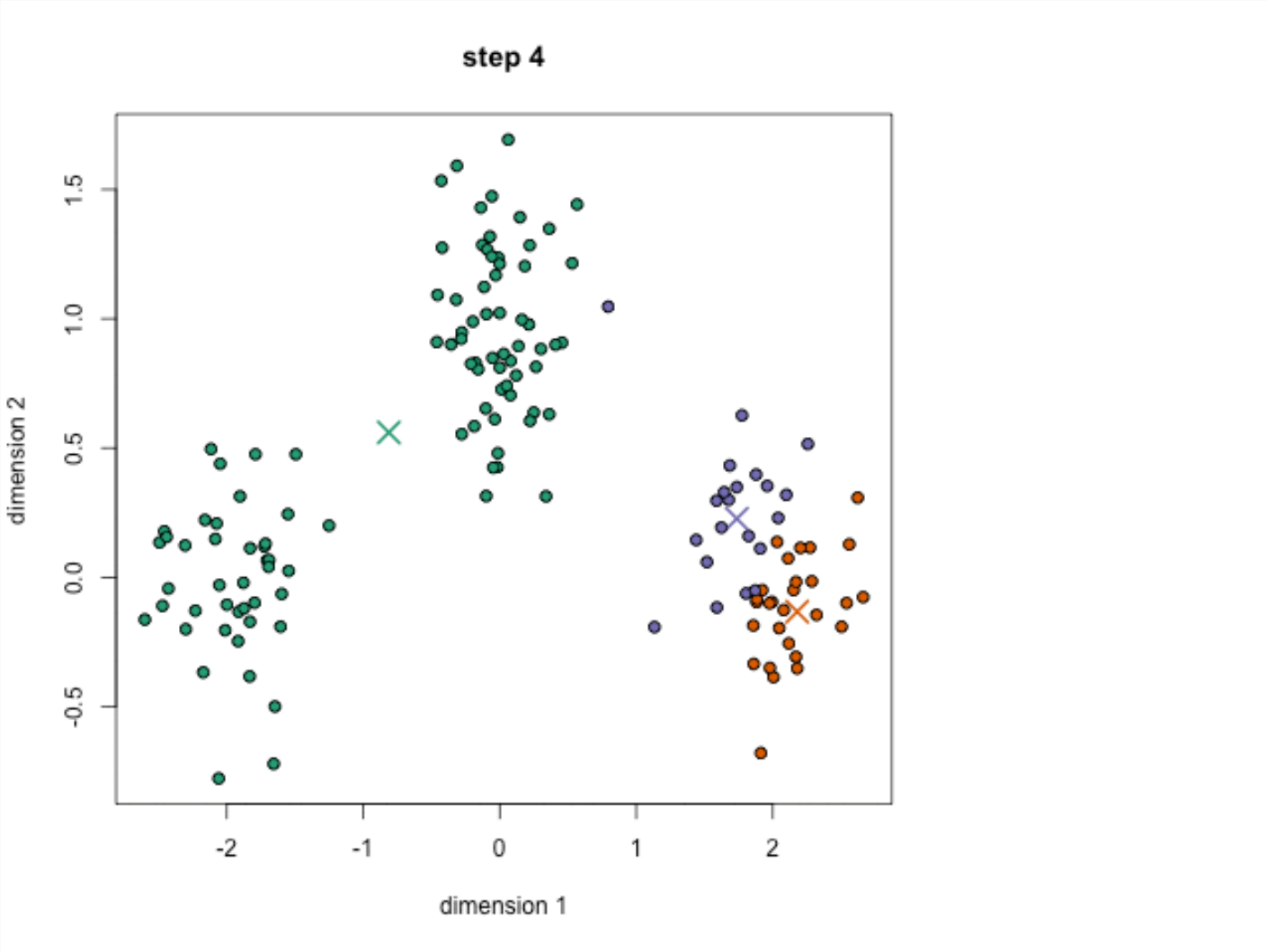
Introduction: K-Means in action



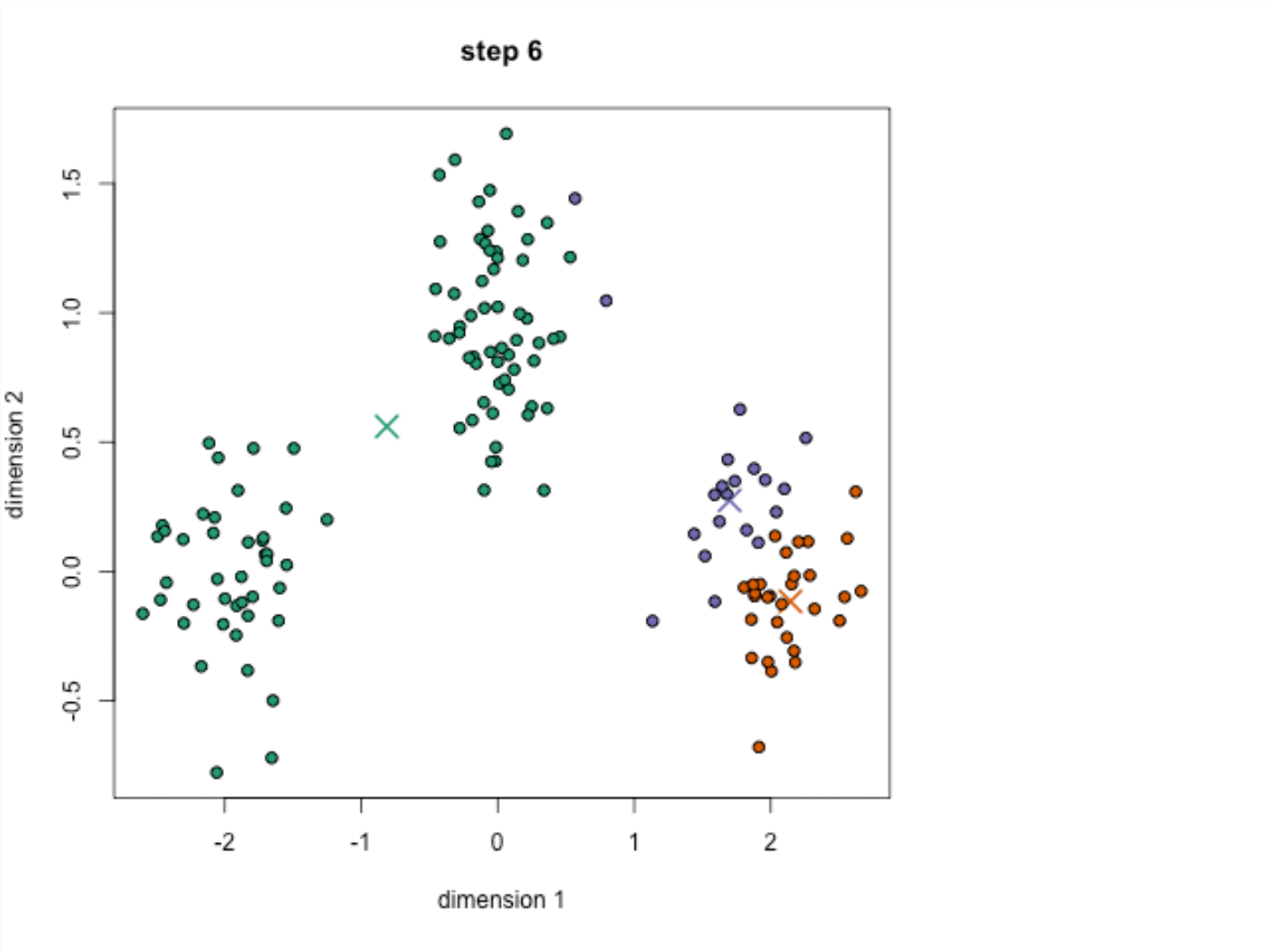
Introduction: K-Means in action



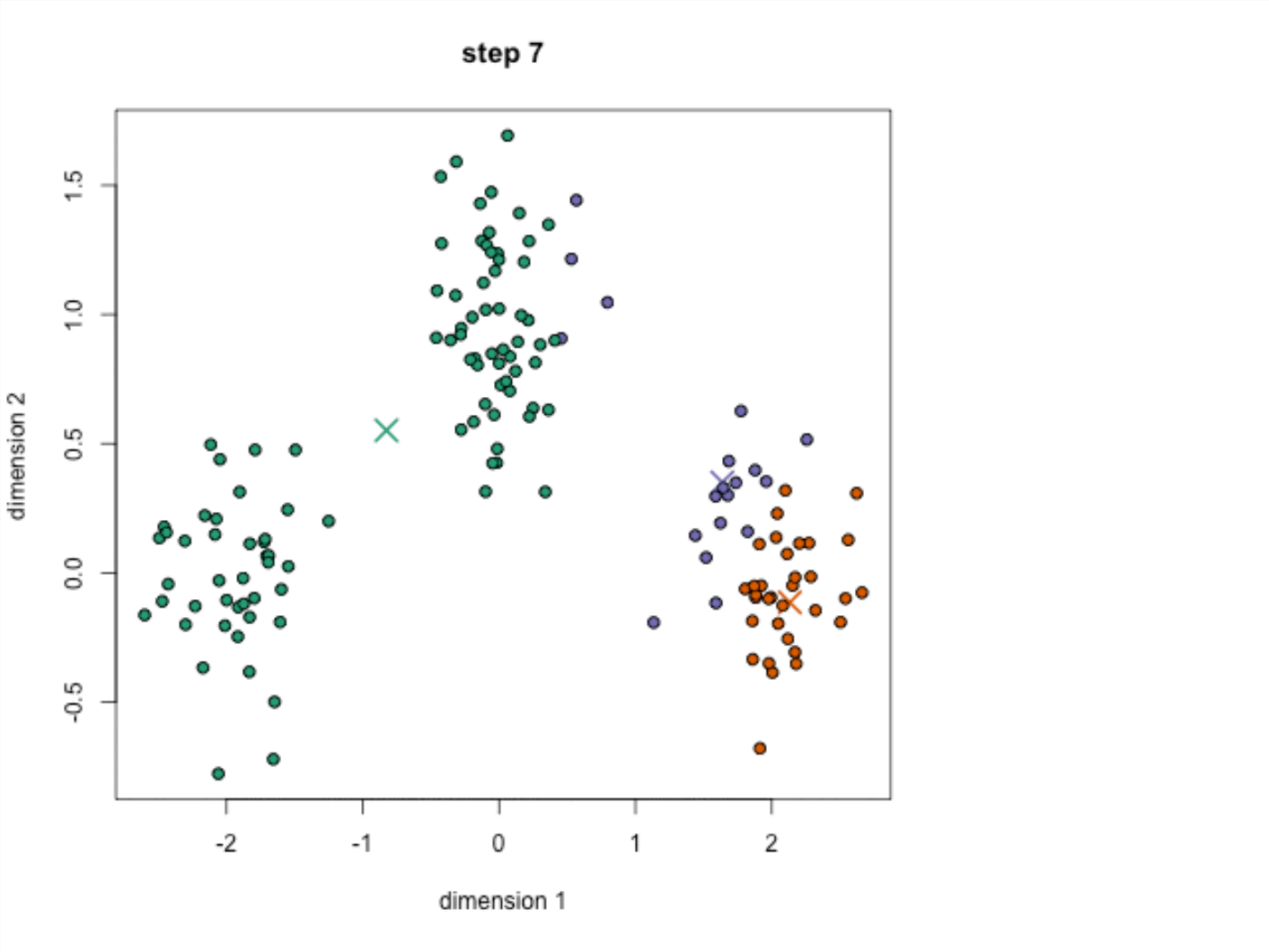
Introduction: K-Means in action



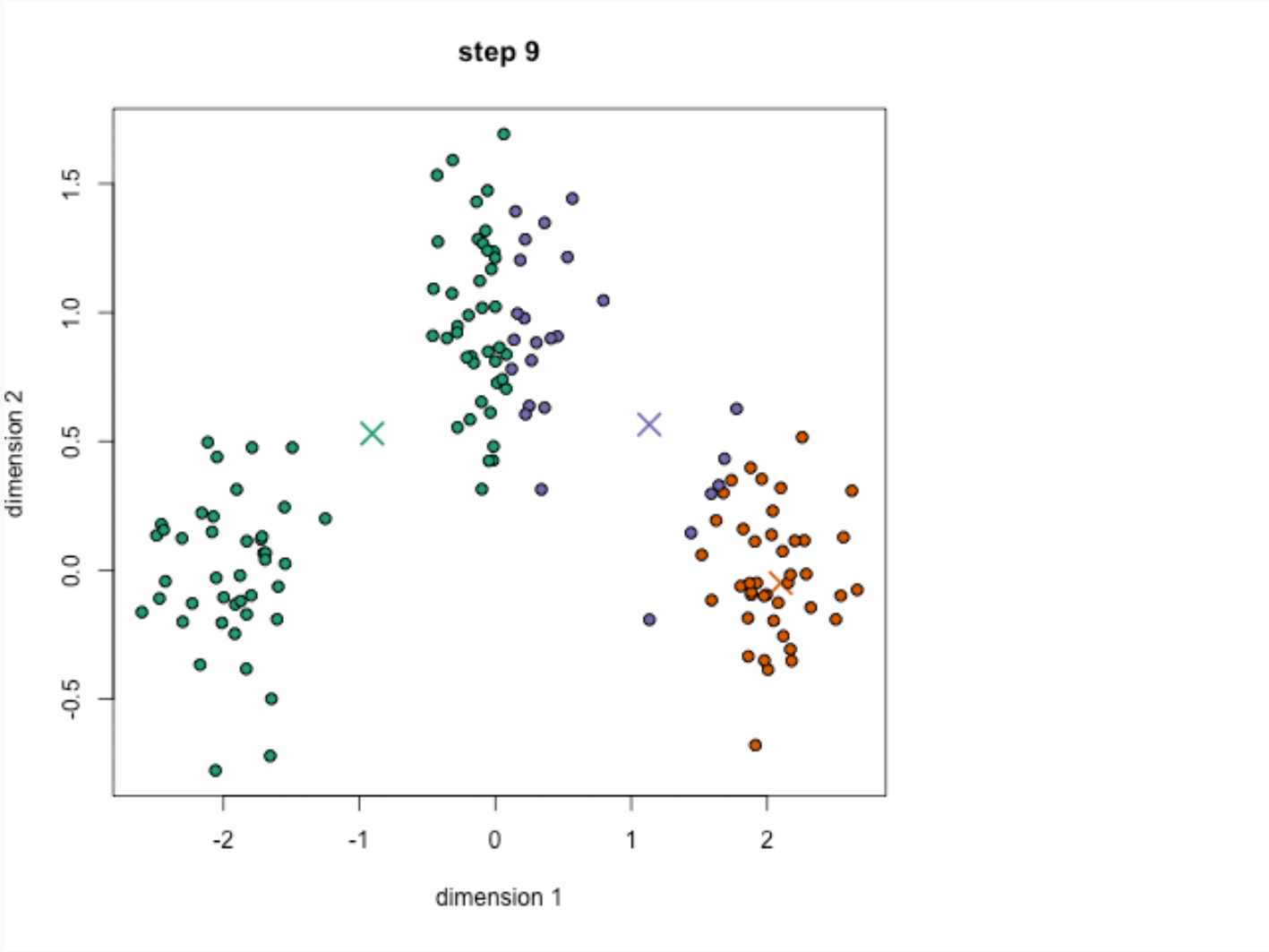
Introduction: K-Means in action



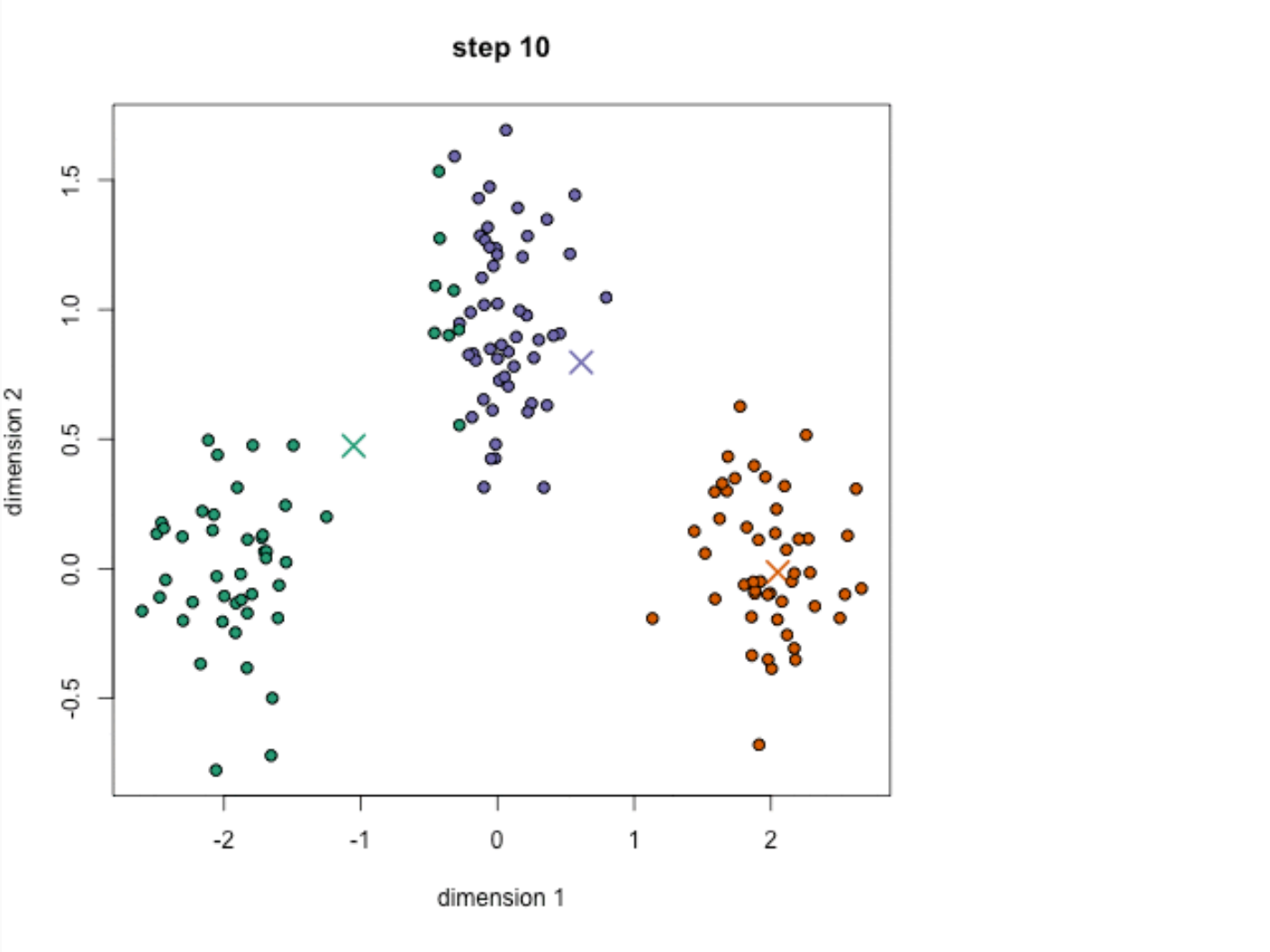
Introduction: K-Means in action



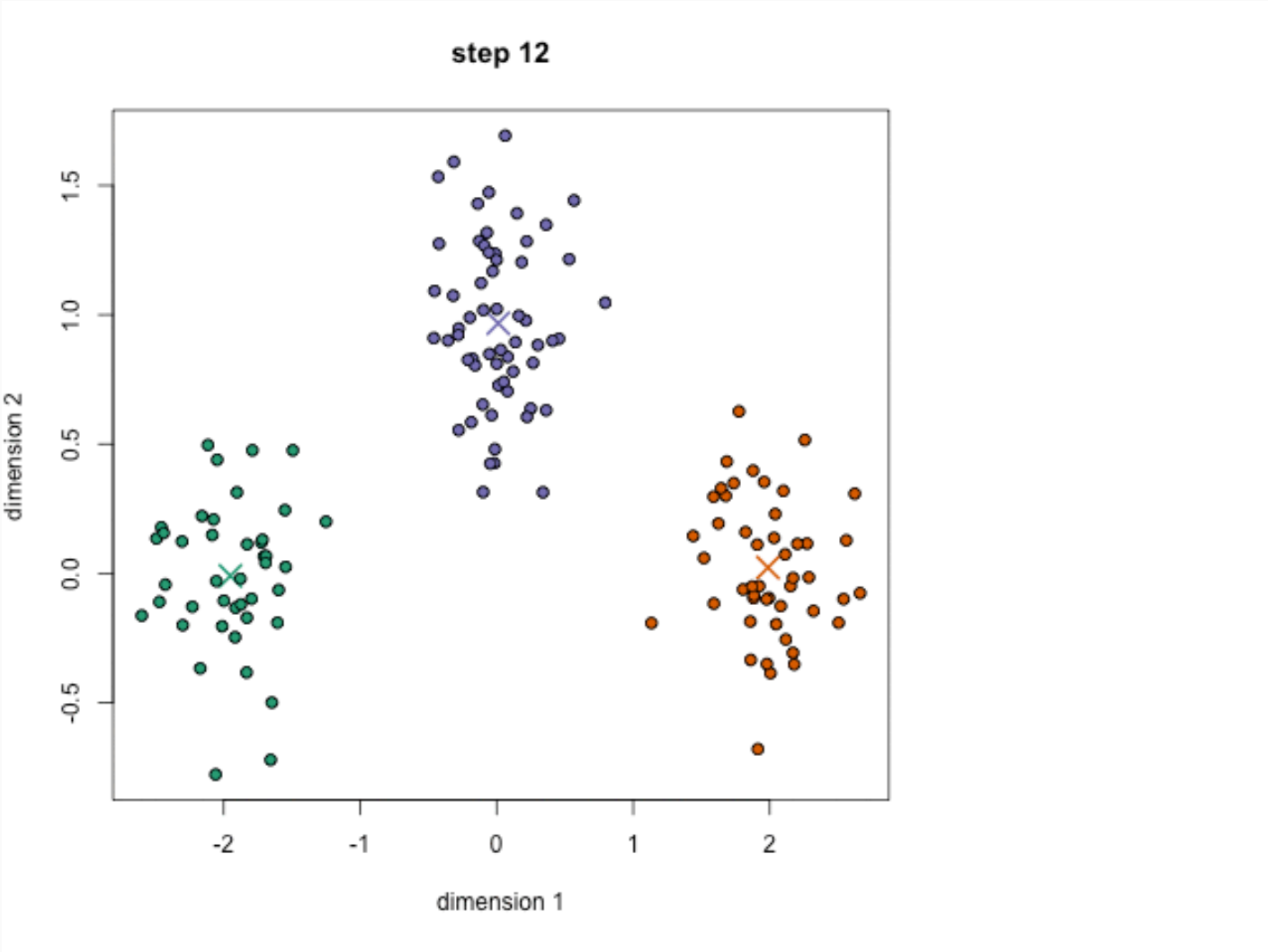
Introduction: K-Means in action



Introduction: K-Means in action



Introduction: K-Means in action



Introduction

Goal of the **Parallel Sparse KMeans-Algorithm**'s project:

Introduction

Goal of the **Parallel Sparse KMeans-Algorithm**'s project:

- **Implement** a Parallel Sparse version of K-Means to typically cluster very sparse data (e.g text data).

Introduction

Goal of the **Parallel Sparse KMeans-Algorithm**'s project:

- **Implement** a Parallel Sparse version of K-Means to typically cluster very sparse data (e.g text data).
- **Apply** it to the 20 Newsgroups data set.

Introduction

Goal of the **Parallel Sparse KMeans-Algorithm**'s project:

- **Implement** a Parallel Sparse version of K-Means to typically cluster very sparse data (e.g text data).
- **Apply** it to the 20 Newsgroups data set.
- **Compare** the scalability of the initialization methods when they are applied to **Parallel Sparse KMeans-Algorithm**

Parallel Sparse KMeans-Algorithm

Parallel Sparse KMeans-Algorithm

K-Means perform badly on noisy features.

Assumption: Dissimilarity is additive with respect to any feature.

Idea: In sparse data, features are either **noisy** or **clustering features**.

- Associate weight 0 to noise features and positive weight to clustering features.
- run K-Means on the modified matrix.

Initialization step

K-Means supports 3 different initialization steps:

Initialization step

K-Means supports 3 different initialization steps:

- **random**, not efficient

Initialization step

K-Means supports 3 different initialization steps:

- **random**, not efficient
- **K-Means++**, $8log(K)$-approximation of the optimum, not scalable

Initialization step

K-Means supports 3 different initialization steps:

- **random**, not efficient
- **K-Means++**, $8log(K)$-approximation of the optimum, not scalable
- **K-Means||**, efficient ($O(log(K)$-Approximation algorithm) and scalable

Initialization step

K-Means supports 3 different initialization steps:

- **random**, not efficient
- **K-Means++**, $8log(K)$-approximation of the optimum, not scalable
- **K-Means||**, efficient ($O(log(K)$-Approximation algorithm) and scalable

Idea: A point x has as many chances to get selected in $\mathcal{C}$ as it is far from the previous centers.

All the algorithms involved here are detailed in the annex of this presentation.

Parallel Sparse KMeans-Algorithm

The Parallel Sparse KMeans-Algorithm is combining:

- **initialization with K-Means||**.
- **K-Means iterations in parallel on the weighted matrix of prediction** using as many CPUs as available.

Work and Depth Analysis

Notations:

- K: Number of clusters
- N: Number of points (ie number of documents)
- p: Dimension of the underlying space: Number of features
- s: Number of nodes in the cluster

Work and Depth Analysis

Total work of K-Means||: $O\big(log(\psi) * K^3 * N^2 * p\big)$.
Total depth for K-Means||: is $O\big(log(p) + log(K) + log(N)\big)$.

Let's suppose K-Means converges in R steps. Then:

Total work of PSK-Algorithm: $O\big(W_{K-Means||} + R * |\tilde{\mathcal{C}}| * p * N\big)$.
Total depth of PSK-Algorithm:
$O\big(D_{K-Means||} + R * (log(p) + log(|\tilde{\mathcal{C}}|) + log(N))\big)$.

PSK-Algorithm on a cluster

Communication cost for K-means:

Master machine reads the clustering center files and shuffles it to the machines.

$Mapper_i$ for $i \in \{1, ..., s\}$ assigns a cluster center for each points $x \in X_i \subset X$ and record the local minimum distance $d(c_i, X_i)$.

s vectors of length k are reduced.

Reducers compute the global sums, update the centers and send this data to all the processors with an *MPI_AllRedude* procedure.

=> Communication cost: $O(sK)$**.**

Communication cost for K-means||: $O(s)$.

Document Clustering

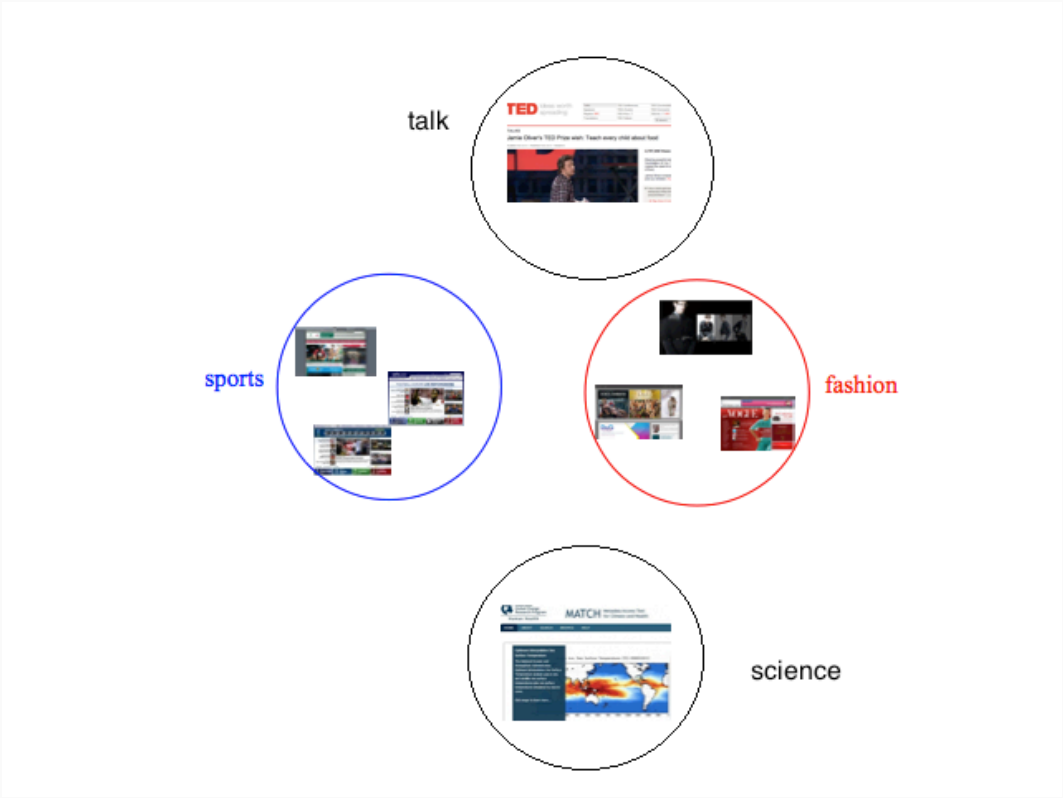
20 newsgroups Data Set

computer	recreational	science
comp.graphics	rec.autos	sci.crypt
comp.os.ms-windows.misc	rec.motorcycles	sci.electronics
comp.sys.ibm.pc.hardware	rec.sport.baseball	sci.med
comp.sys.mac.hardware	rec.sport.hockey	sci.space
comp.windows.x		

misc.forsale	religion	talk.politics
misc.forsale	talk.religion.misc	talk.politics.misc
	alt.atheism	talk.politics.guns
	soc.religion.christian	talk.politics.mideast

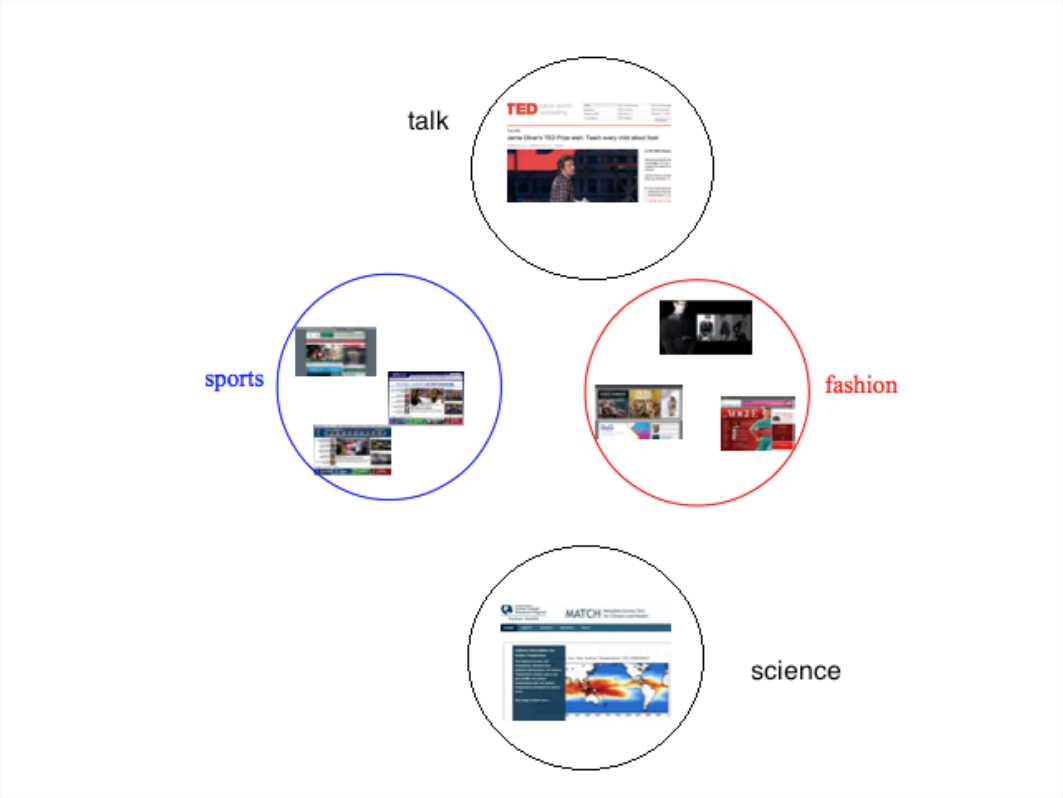
The Term Frequency-Inverse Document Frenquency transformation

TF-IDF transformation: **a document-to-vector map** which is:



The Term Frequency-Inverse Document Frenquency transformation

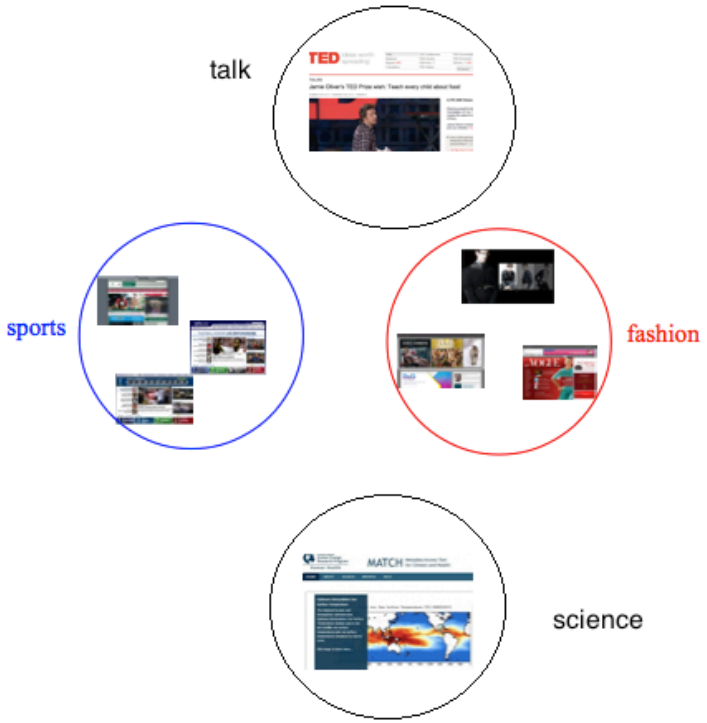
TF-IDF transformation: **a document-to-vector map** which is:



- Relevant to text processing

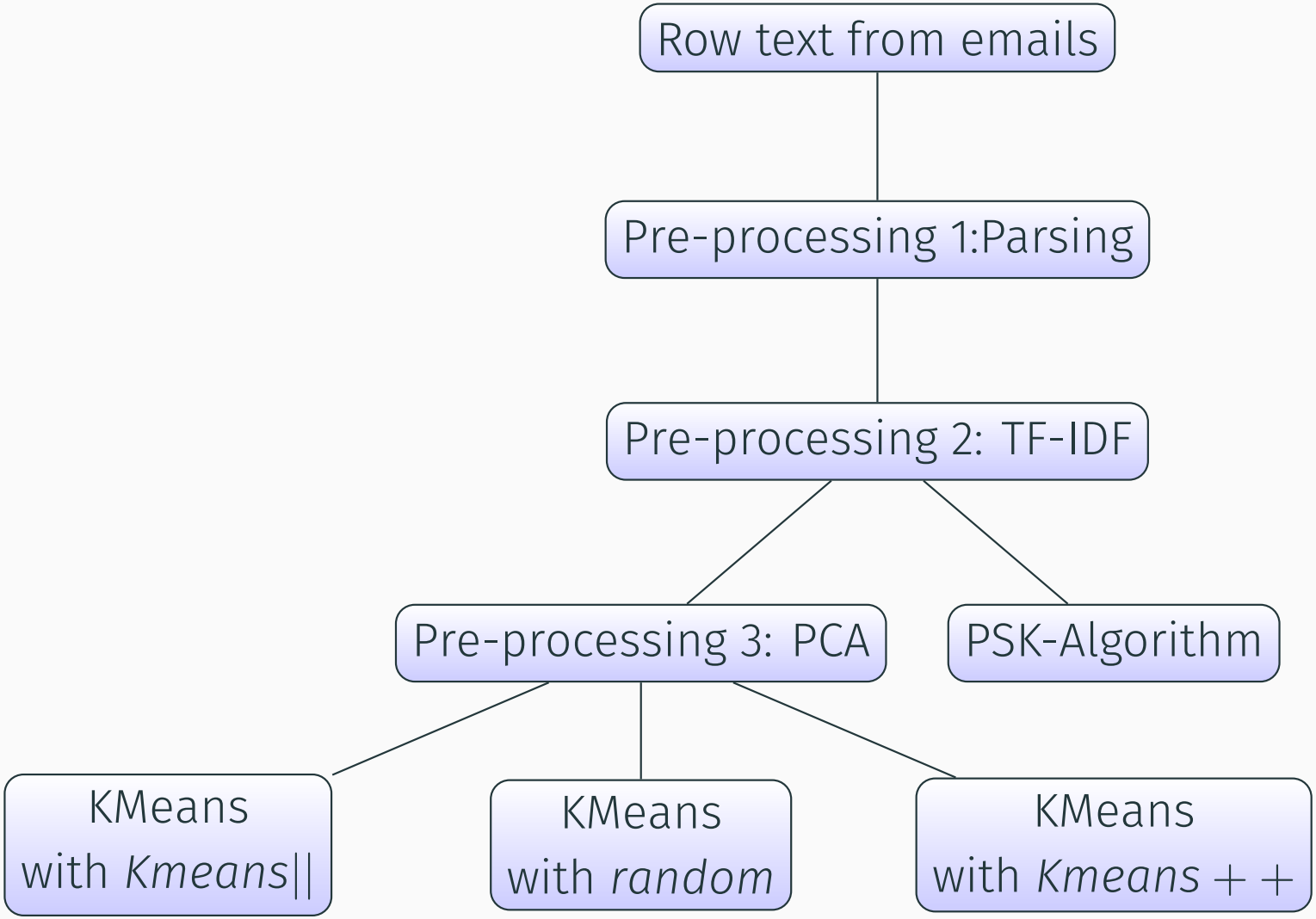
The Term Frequency-Inverse Document Frenquency transformation

TF-IDF transformation: **a document-to-vector map** which is:

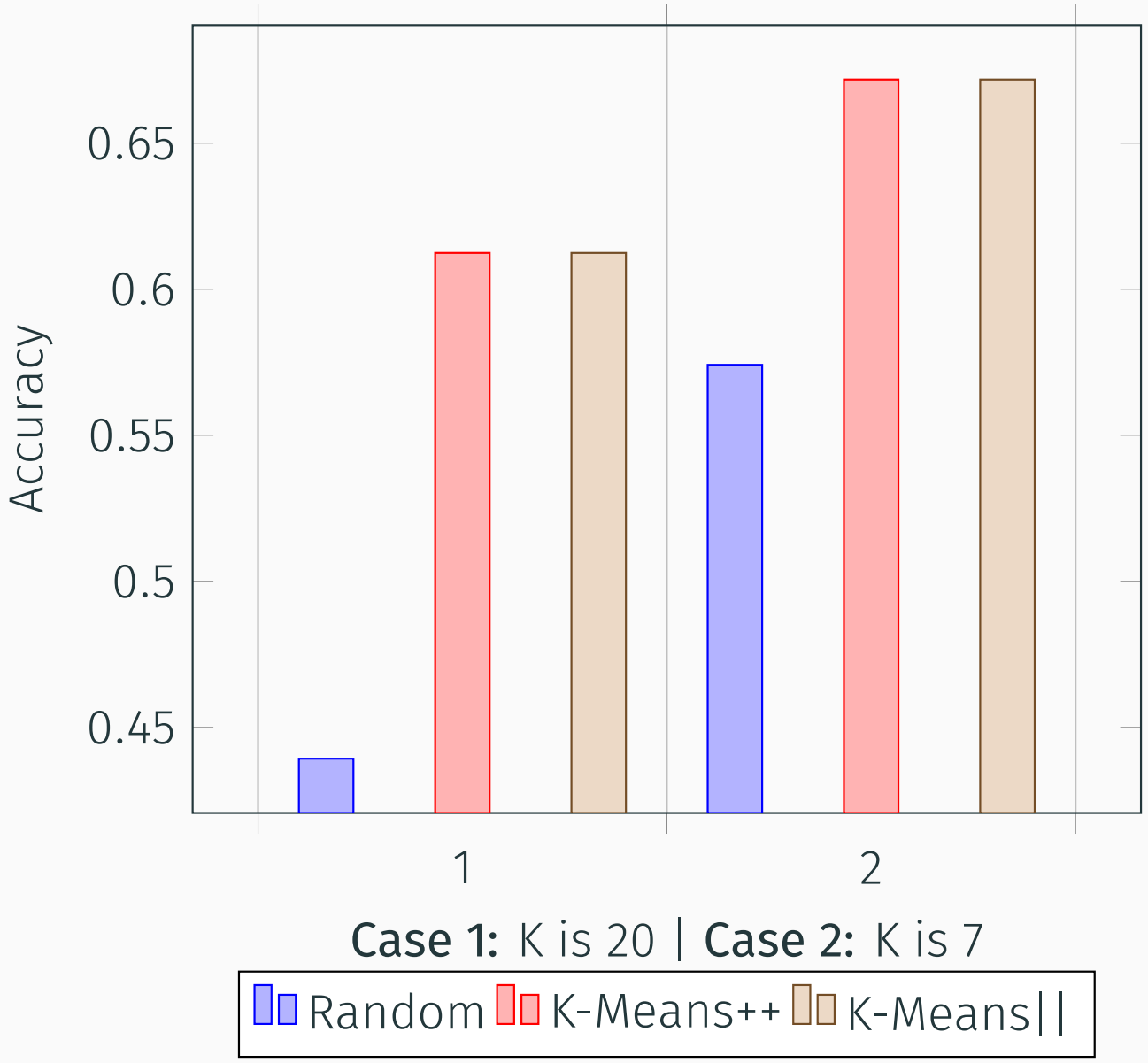


- Relevant to text processing
- Common web analysis algorithm

The Pipeline



Results



To the sake of comparison, Naive Bayses gives an accuracy of 0.4515.

Scaling

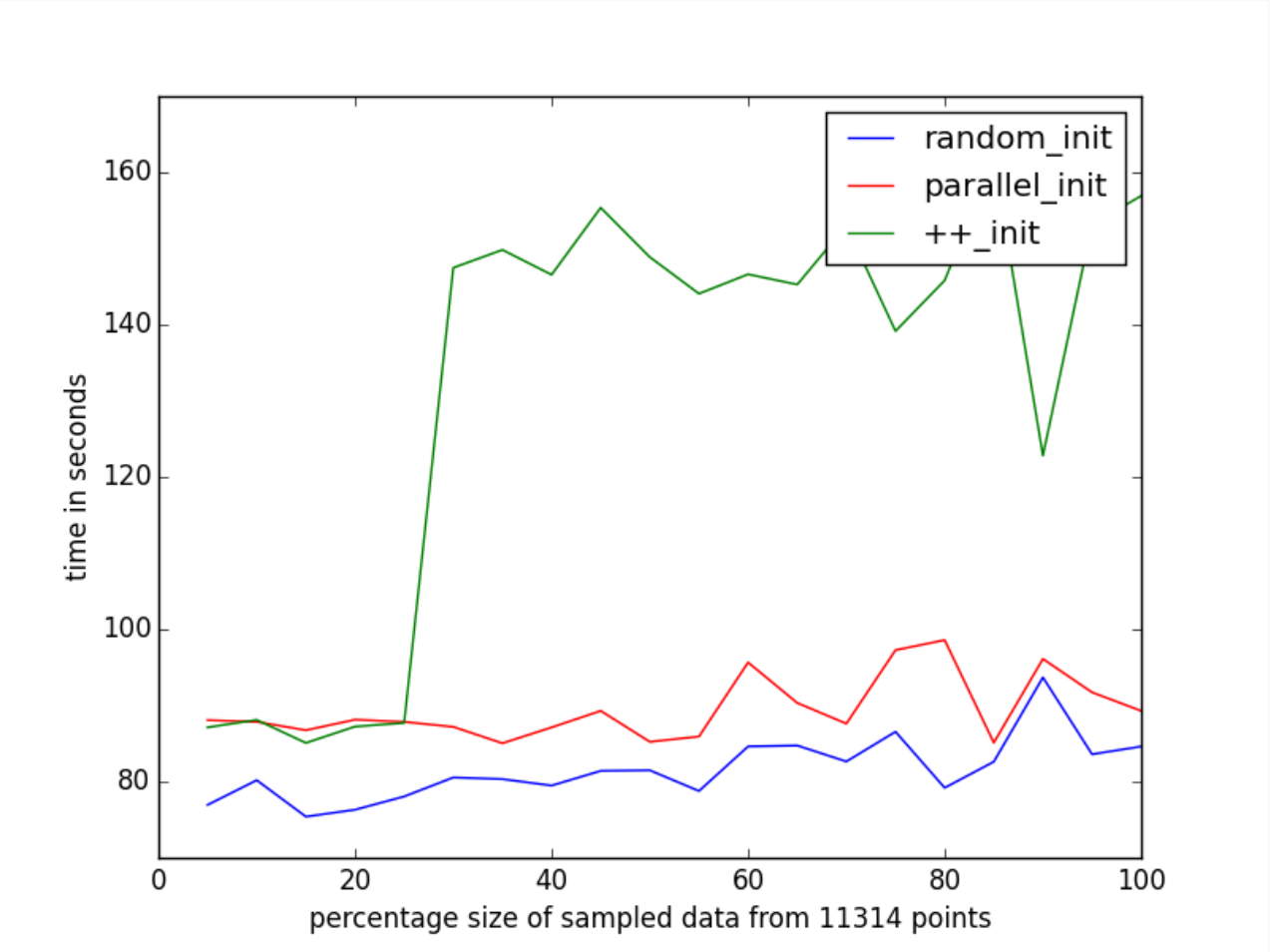


Figure 1: Runtime of initializations with respect to data size

Conclusion

Summary

Parallel Sparse KMeans-Algorithm:

- **Scalable** with respect to the size of the data
- differentiate clustering features from noise.

Code location:

`https://github.com/victorstorchan/doc_clustering`

Thank you! Questions?

Annex

type of initialization	accuracy for K=20	total time
random	0.4394	84.6117
K-Means++	0.6124	156.9197
K-Means||	0.6124	90.6363

type of initialization	accuracy for K=7	total time
random	0.5741	57.9025
K-Means++	0.6718	74.8596
K-Means||	0.6718	60.7070

Term Frequency/Inverse Document Frequency

$$tf_i = \frac{n_i}{\sum n_k} \text{ and } idf_i = log\frac{|D|}{|\{d : t_i \in d\}|}$$

|D|: total number of documents in the corpus

$\{d : t_i \in d\}$: number of documents where the term t_i appears.

Initialization step

Algorithm 2 $K - Means++$

procedure K-MEANS++
  $\mathcal{C} \leftarrow$ Sample one point uniformly at random in X
  while $|\mathcal{C}| < K$ **do**
    Sample $x \in X$ with probability $\frac{d^2(x,\mathcal{C})}{\phi_X(\mathcal{C})}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{x\}$

Algorithm 3 $K - Means||$

procedure K-MEANS||
  $\tilde{\mathcal{C}} \leftarrow$ Sample one point uniformly at random in X
  $\psi \leftarrow \phi_X(\tilde{\mathcal{C}})$
  for $O(log(\psi)$ times **do**
    $C' \leftarrow$ Sample each $x \in X$ independently with probability $\frac{ld^2(x,\tilde{\mathcal{C}})}{\phi_X(\tilde{\mathcal{C}})}$
    $\tilde{\mathcal{C}} \leftarrow \tilde{\mathcal{C}} \cup C'$
  for $x \in \tilde{\mathcal{C}}$ **do**
    set $w_x = \#$ { points in X closer to x than any other points in $\tilde{\mathcal{C}}$ }
  Recluster the weighted points in $\tilde{\mathcal{C}}$ into K clusters: use K-Means++ on
  $\{w_1 c_1 \in R^p, ..., w_{|\tilde{\mathcal{C}}|} c_{|\tilde{\mathcal{C}}|} \in R^p\}$. Return $\mathcal{C}$

Parallel Sparse KMeans-Algorithm

Algorithm 5 *PSK – Algorithm*

```
1: procedure PSK-ALGORITHM
2:     Run SK-Means with modified step (a):
3:         When performing K-means on the weighted data, initialize the procedure with
4:         K-Means|| and run the K-Means iterations in parallel using as many CPUs as
5:         available.
```

Algorithm 1 $SK-Means(K)$

procedure SK-MEANS

$w \leftarrow \frac{1}{p}[1,1,...,1]^T$

Repeat (until convergence):

Step (a):

$Y := [\sqrt{w_1}X_1, ..., \sqrt{w_p}X_p]$

for a given w, perform K-means on the transformed data set Y:

solve $\tilde{C} = argmin_{|C|=K} WSS_y(C)$ where $WSS_y(C)$ is the within cluster sum of squares on Y.

Step (b):

For a given $\tilde{C}$, maximize ith respect to w: set $D = [BSS_1(\tilde{C}), ..., BSS_p(\tilde{C})]$ and solve:

$$\tilde{w} = max_w w^T D$$

$$\text{such that} \begin{cases} \|w\|_1 \leq l_1 \\ \|w\|_2^2 \leq 1 \\ w_j \geq 0 \end{cases}$$

Future work

- Use a parallel spherical K-Means
- **Parallel Sparse KMeans-Algorithm** is affected by a even few outliers. Implement a parallel robustification of it.