SDSC Summer Institute 2020

ML5 - Unsupervised Learning with Clustering
Paul Rodriguez
08/05/20
Location: Breakout Room
Gitter (session support): Breakout Room
Gitter (general system support): Help Desk

K-means cluster idea

Group points into clusters by some measure of distance

Assign points to 1 cluster only

Use clusters as a summary of data

Distance Measures

For numeric data

Euclidean distance (or sum squared differences)

Distance Measures

For numeric data

Euclidean distance (or sum squared differences)

For categorical data

‘Gower’ metric uses ‘same’ or ‘different’ counts

Lots of others…

Kmeans Clustering

Objective: minimize within-cluster distances

Kmeans Clustering

Objective: minimize within-cluster distances

Start with K initial cluster centers

spread out initial guesses

Kmeans Clustering

Objective: minimize within-cluster distances

Start with K initial cluster centers

spread out initial guesses

Loop:

Assign each data point to nearest cluster center

Kmeans Clustering

Objective: minimize within-cluster distances

Start with K initial cluster centers

Loop:

- Assign each data point to nearest cluster center
- Calculate mean of cluster for new center

Kmeans Clustering

Objective: minimize within-cluster distances

Start with K initial cluster centers

Loop:

- Assign each data point to nearest cluster center
- Calculate mean of cluster for new center
- Stop when assignments don't change

converges (but not to global min)

Kmeans Clustering

Objective: minimize within-cluster distances

Start with K initial cluster centers

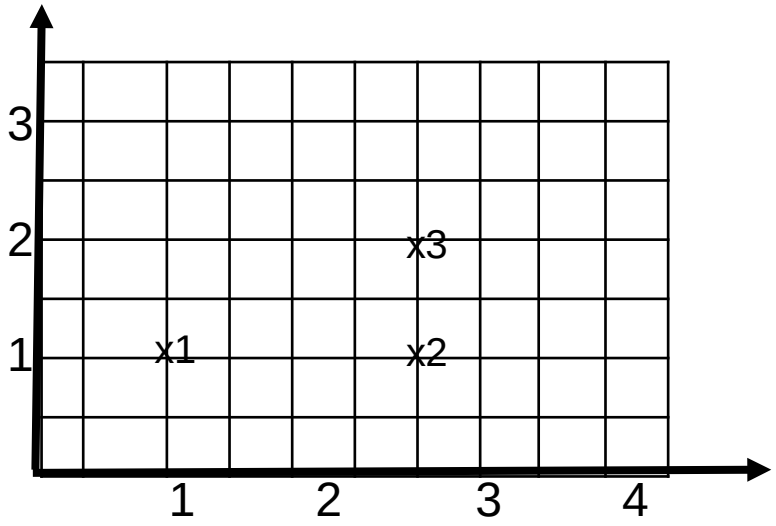
Loop:

- Assign each data point to nearest cluster center
- Calculate mean of cluster for new center
- Stop when assignments don't change

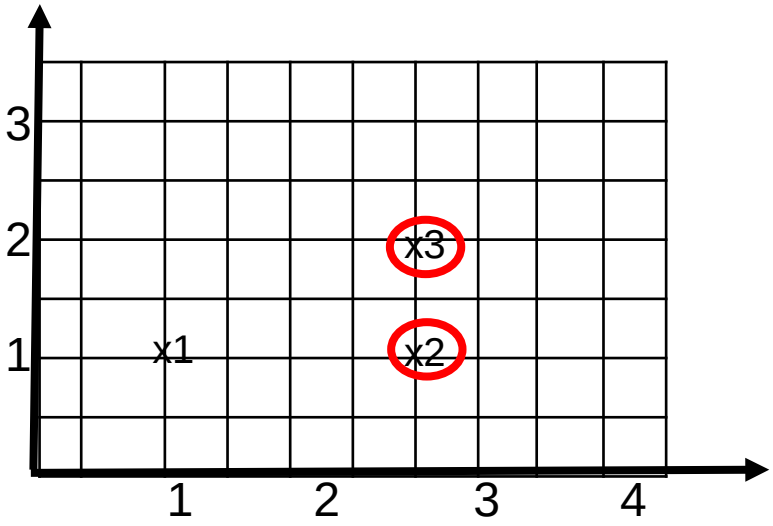
Kmeans works in a variety of situations, but choosing K is sometimes difficult

A quick simple example:

For these points
x1=(1,1)
x2=(3,1)
x3=(3,1.99)



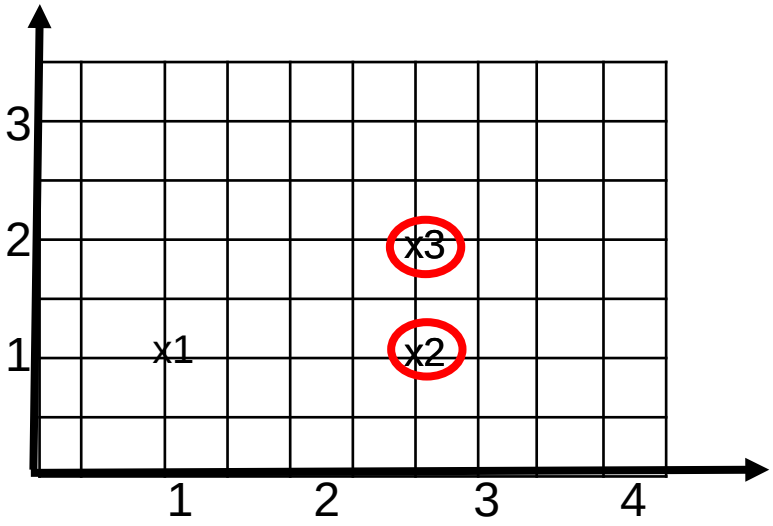
For these points
x1=(1,1)
x2=(3,1)
x3=(3,1.99)



Use K=2, set initial centers as:

$\mu_1 = x_2$ and $\mu_2 = x_3$

For these points
x1=(1,1)
x2=(3,1)
x3=(3,1.99)



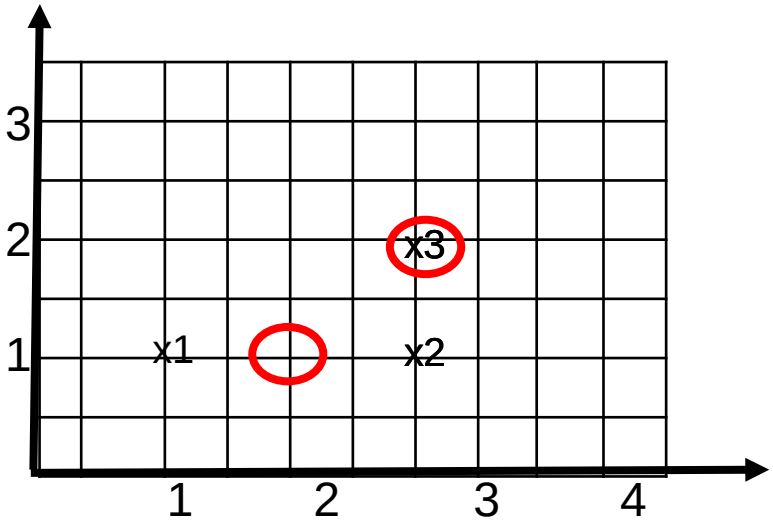
Use K=2, set initial centers as:

$\mu_1 = x_2$ and $\mu_2 = x_3$

Loop 1:

get distance of all points to all clusters means, a NxK distance matrix

For these points
x1=(1,1)
x2=(3,1)
x3=(3,1.99)



Use K=2, set initial centers as:

$\mu_1 = x_2$ and $\mu_2 = x_3$

Loop 1:

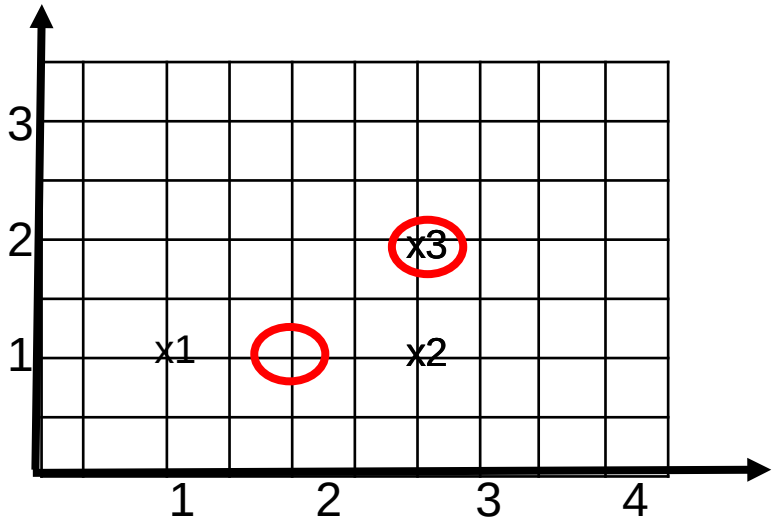
get distance of all points to all clusters means, a NxK distance matrix

.

assign x_1, x_2 to cluster 1

assign x_3 to cluster 2

For these points
x1=(1,1)
x2=(3,1)
x3=(3,1.99)



Use K=2, set initial centers as:

$\mu_1 = x_2$ and $\mu_2 = x_3$

Loop 1:

get distance of all points to all clusters means, a NxK distance matrix

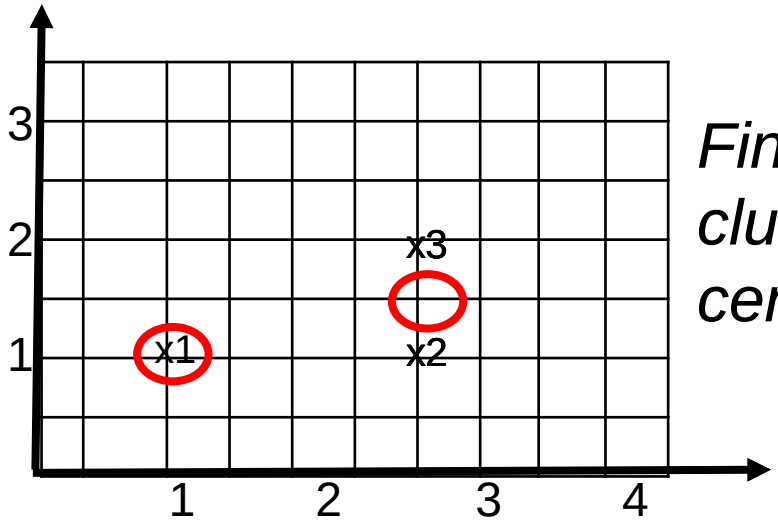
.

assign x_1, x_2 to cluster 1

assign x_3 to cluster 2

After calculating new center, what happens to x2 assignment in next loop?

For these points
x1=(1,1)
x2=(3,1)
x3=(3,1.99)



Final cluster centers

Use K=2, set initial centers as:

$\mu_1 = x_2$ and $\mu_2 = x_3$

Loop 1:

get distance of all points to all clusters means, a NxK distance matrix

.

assign x_1, x_2 to cluster 1

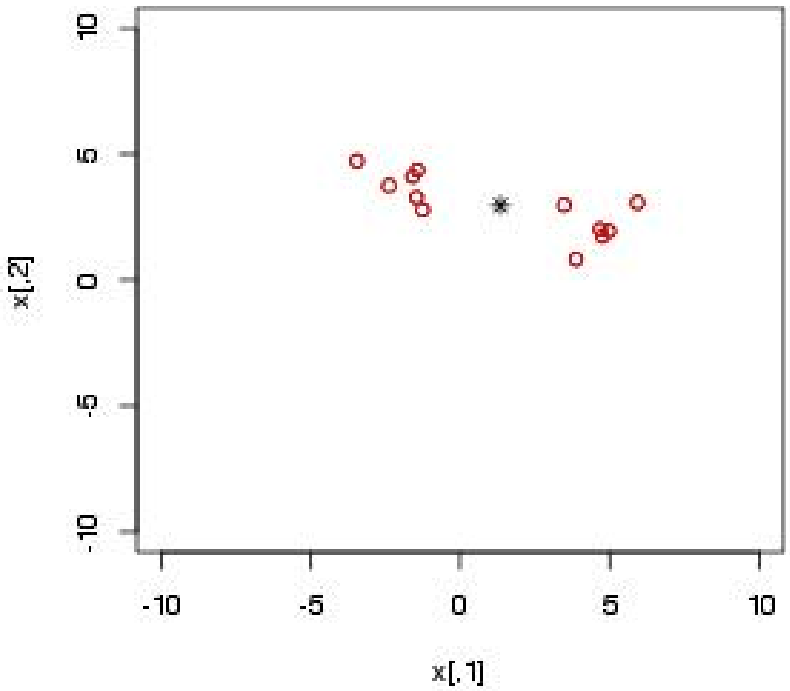
assign x_3 to cluster 2

After calculating new center, what happens to x2 assignment in next loop?

Kmeans Examples

Kmeans Example

- **For K=1 where is the cluster center?**

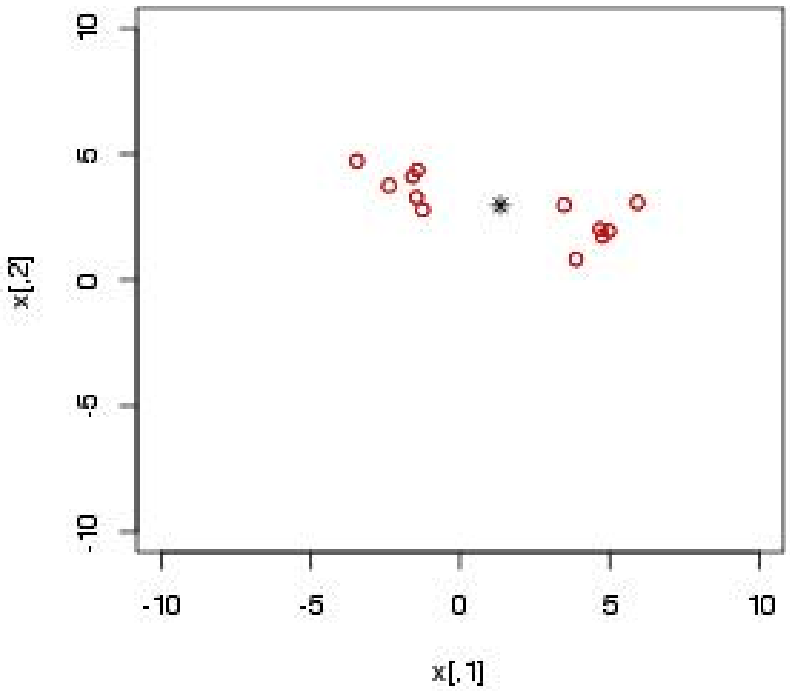


Essential R commands:
Kresult = kmeans(X,1,10,1)

#K=1
#10 is max loop iterations
#1 is number of initial sets to try

Kmeans Example

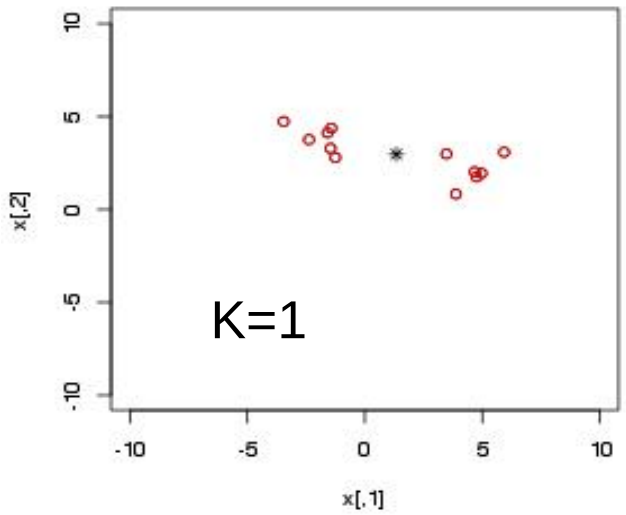
- **For K=1 where is the cluster center?** ***At the overall mean***

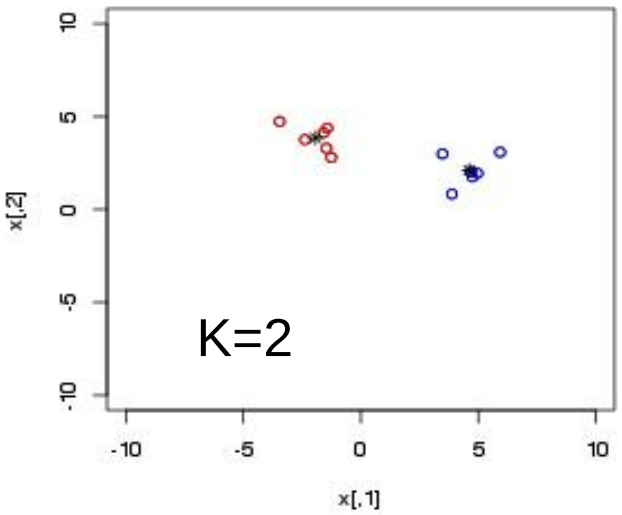


Essential R commands:
Kresult = kmeans(X,1,10,1)

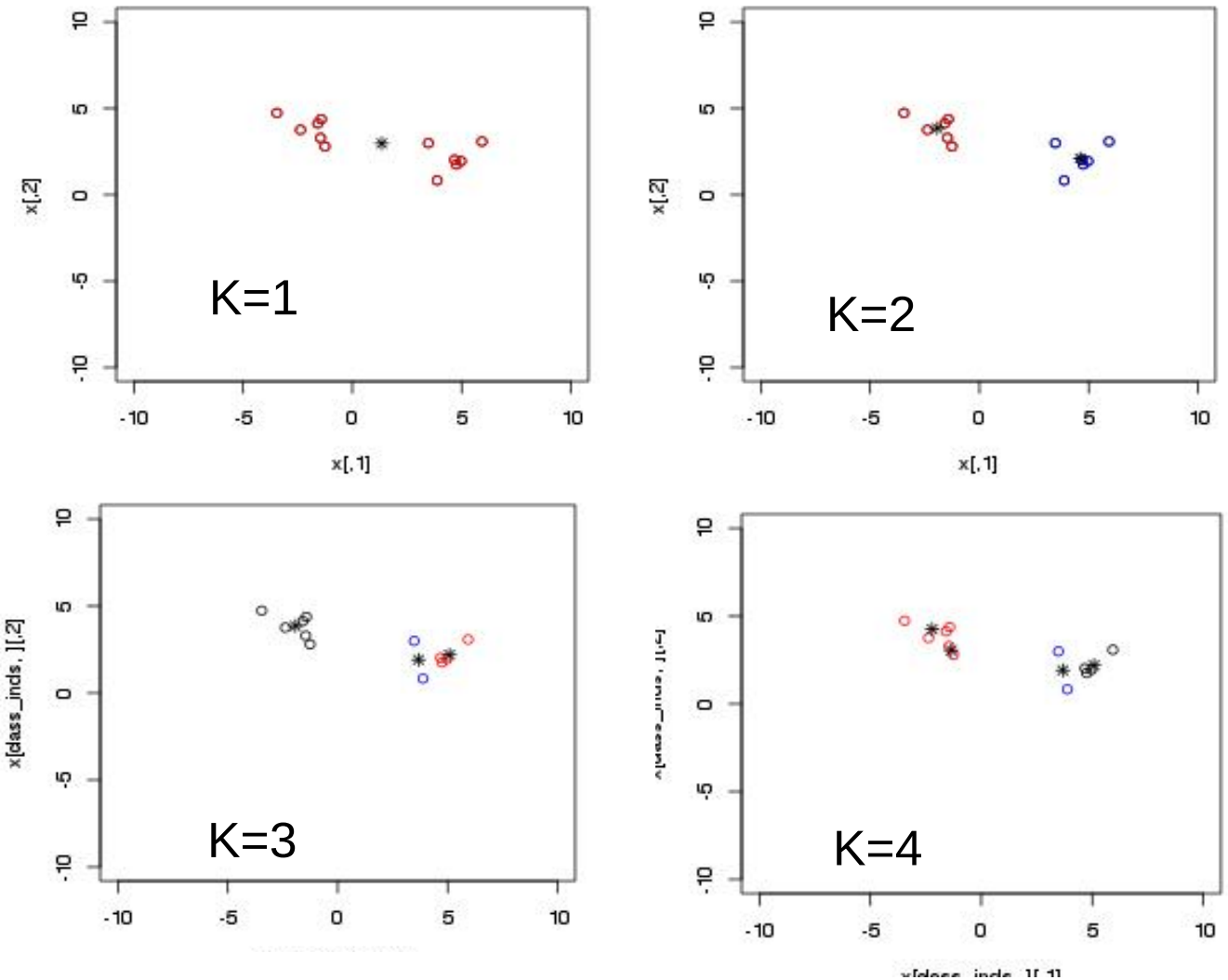
#K=1
#10 is max loop iterations
#1 is number of initial sets to try

Kmeans Example

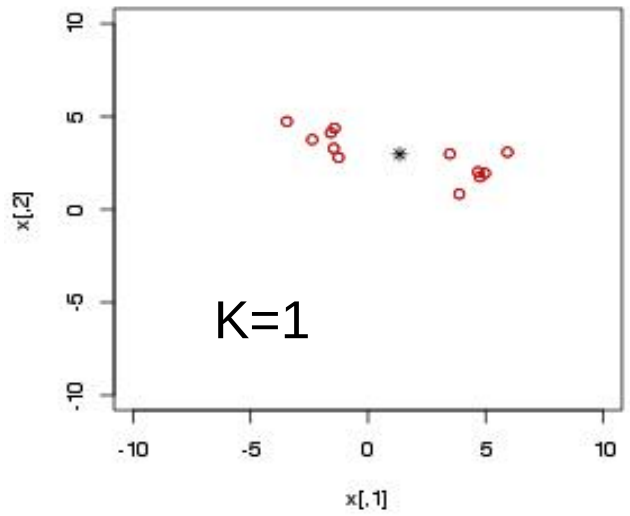


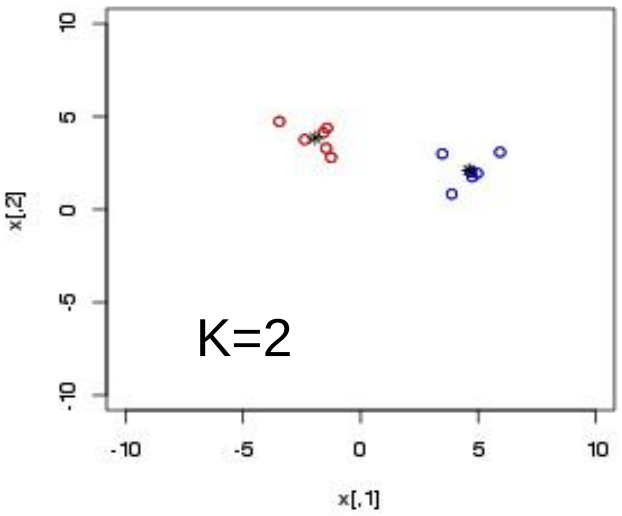


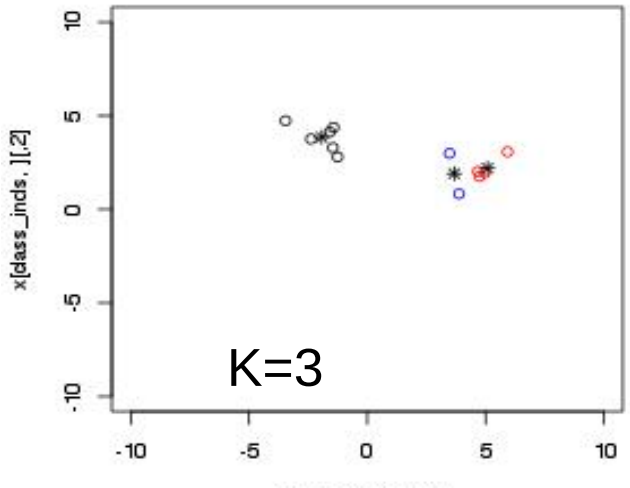
Kmeans Example

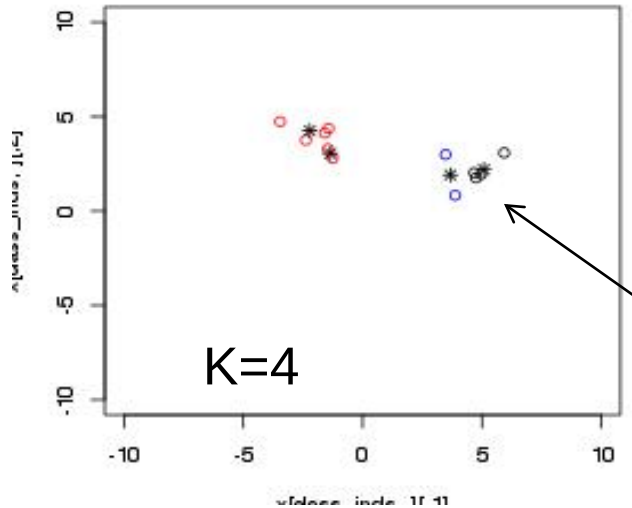


Kmeans Example





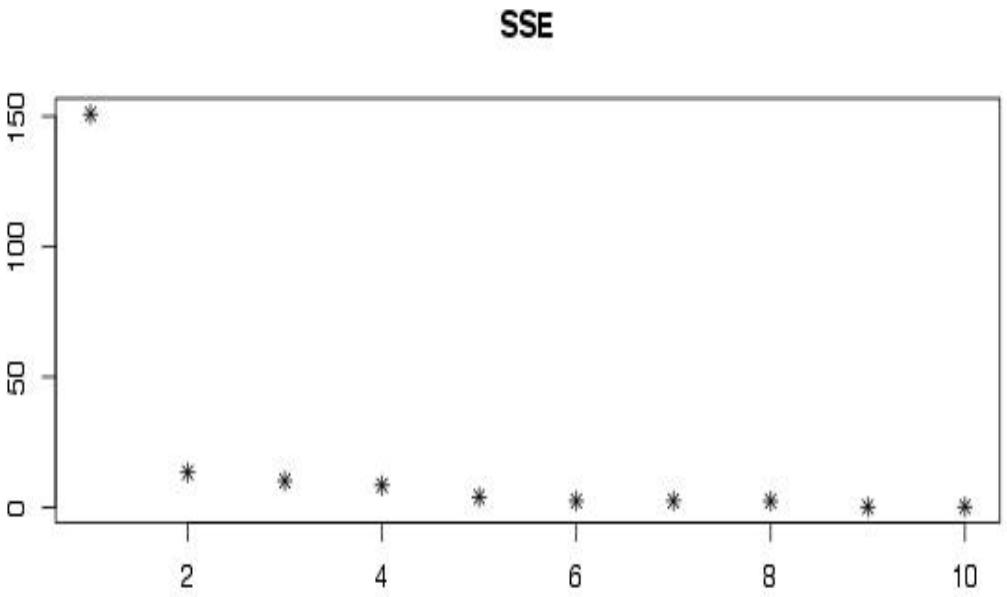




As K increases individual points get a cluster

Choosing K for Kmeans

Total Within Cluster SSE



K=1 to 10

Essential R commands:

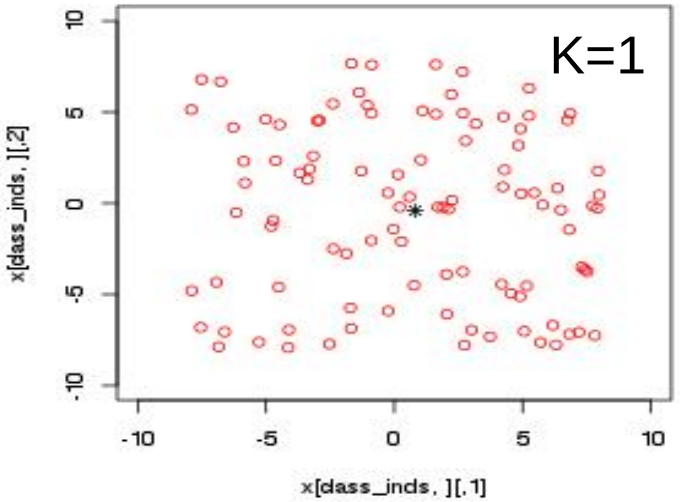
```
for (num_k in 1:10) {
Kres=kmeans(X,num_k,10,1);
```

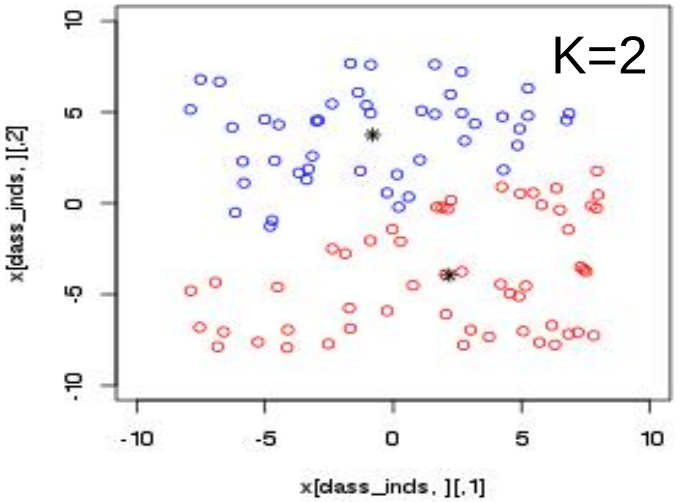
Save and then plot
Kres$tot.withinss

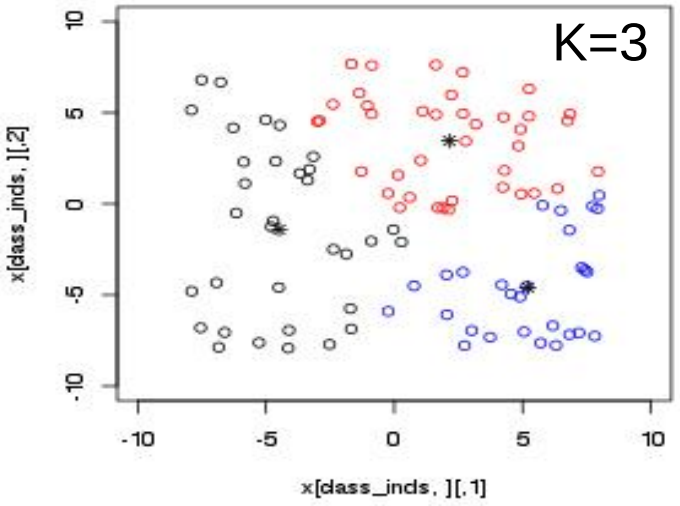
…

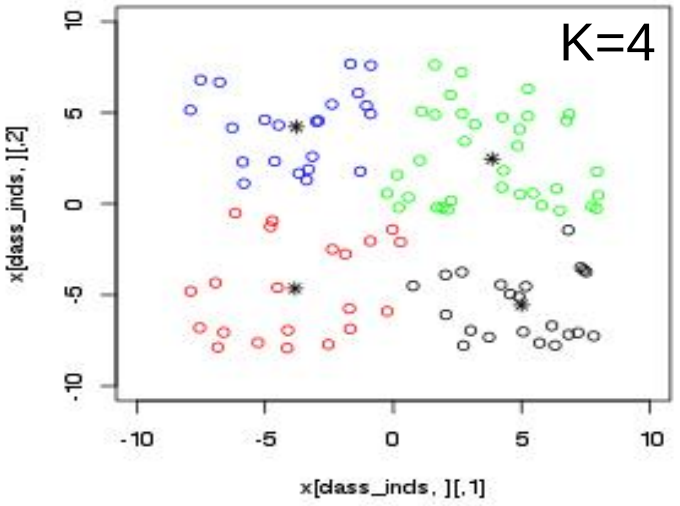
- Not much improvement after K=2 (“elbow”)

Kmeans Example: uniform dist.



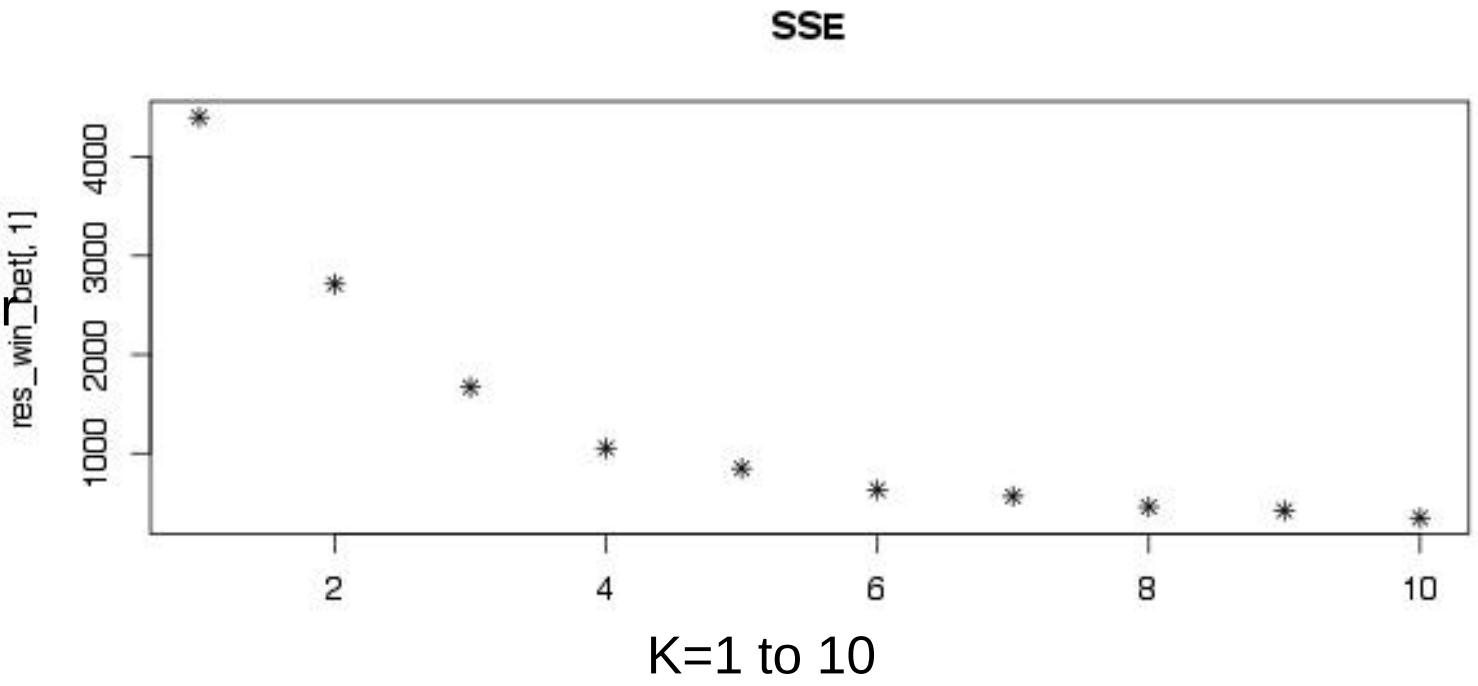






Choosing K - uniform

Total Within Cluster SSE



K=1 to 10

- Smooth decrease across K => less structure

Choosing K methods

- **“Elbow” of Sum Squared Error Within Clusters**

Choosing K methods

- **“Elbow” of Sum Squared Error Within Clusters**

- **“Silhouette”: mean SSE within a cluster vs to next best cluster**

  take maximum value over K=1…Kmax

Choosing K methods

- **“Elbow” of Sum Squared Error Within Clusters**

- **“Silhouette”: mean SSE within a cluster vs to next best cluster**

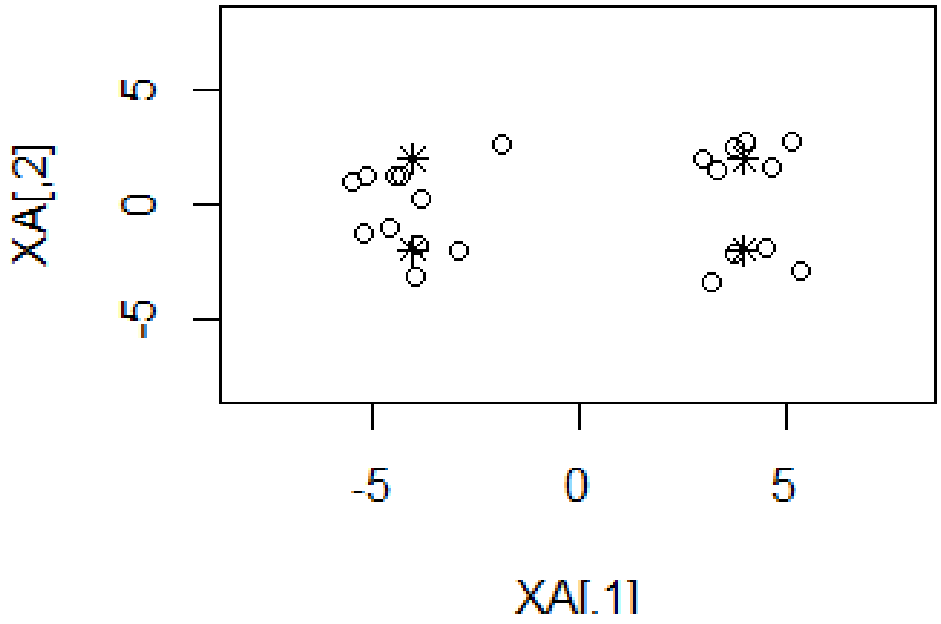
  take maximum value over K=1…Kmax

- **“Gap” value of SSE-within-cluster of data vs uniform distribution**

  take maximum value over K=1 …Kmax

Choosing K methods

- **Example, 4 clusters normal distribution, small sample**

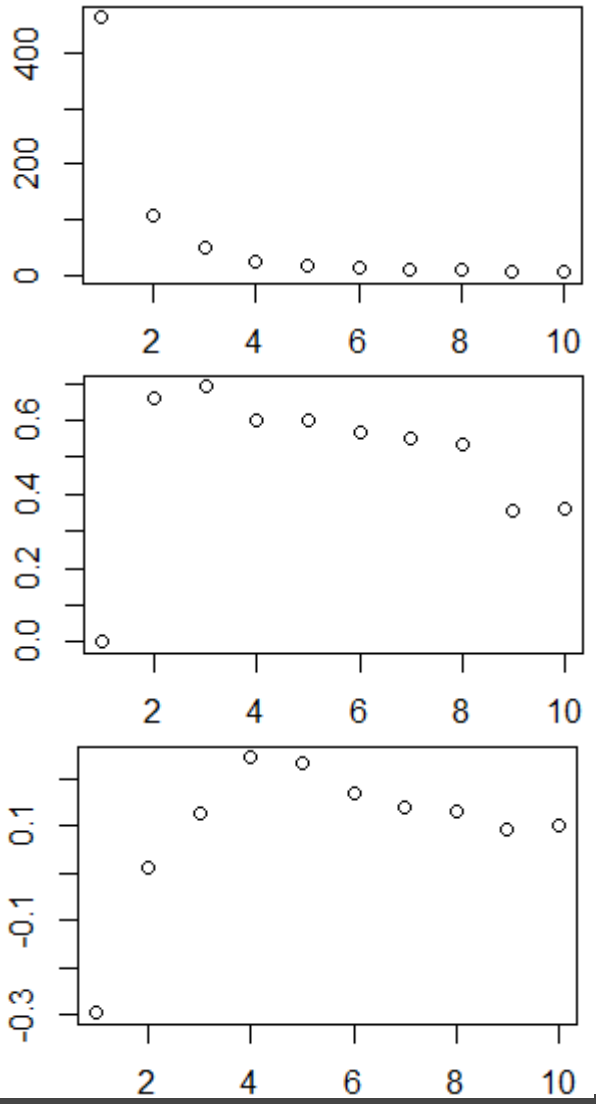


Choosing K methods

SSE within cluster (elbow)

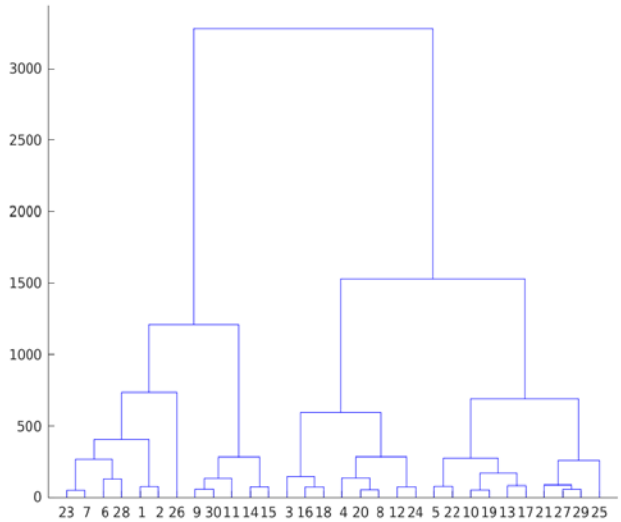
Silhouette within vs next best

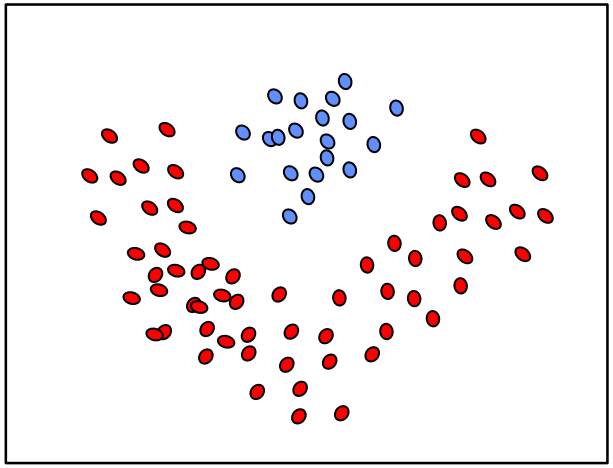
Gap SSE within cluster vs uniform data baseline



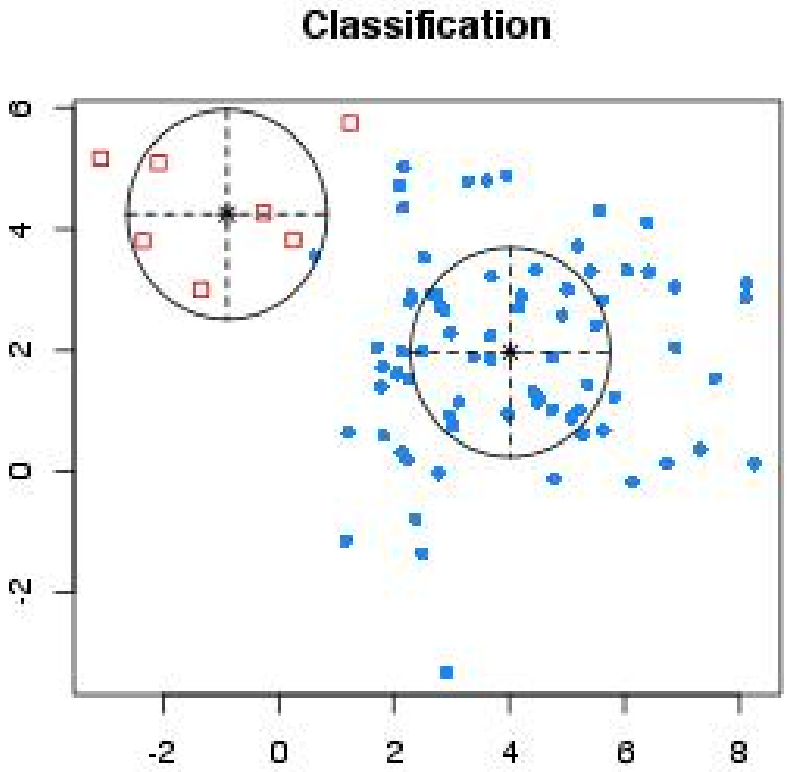
Many other clustering variations -

- **Hierarchical clustering – start with N clusters and merge points into large clusters (good to get whole tree)**
- **Density based clustering - build and link neighborhoods (good for spatial data)**





EM clustering
(expectation maximization)

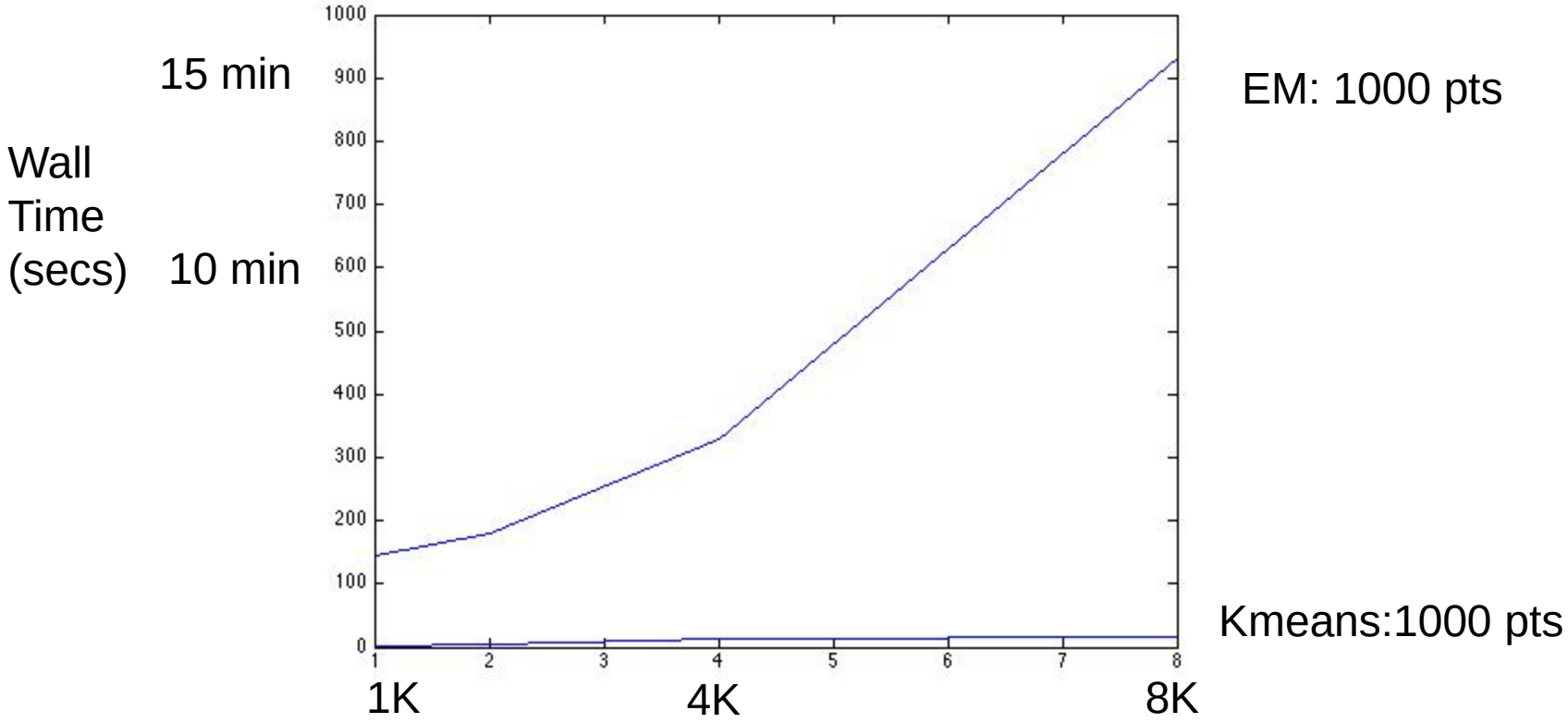


- **A probabilistic model using mixture of Gaussians**
- **Handles unequal variance and/or cluster sizes better than K-means**

R:
library('mclust')
em_fit=Mclust(x);
plot(em_fit);

Kmeans vs EM performance

1 Gordon compute node, normal random matrices
R: system.time(Mclust())

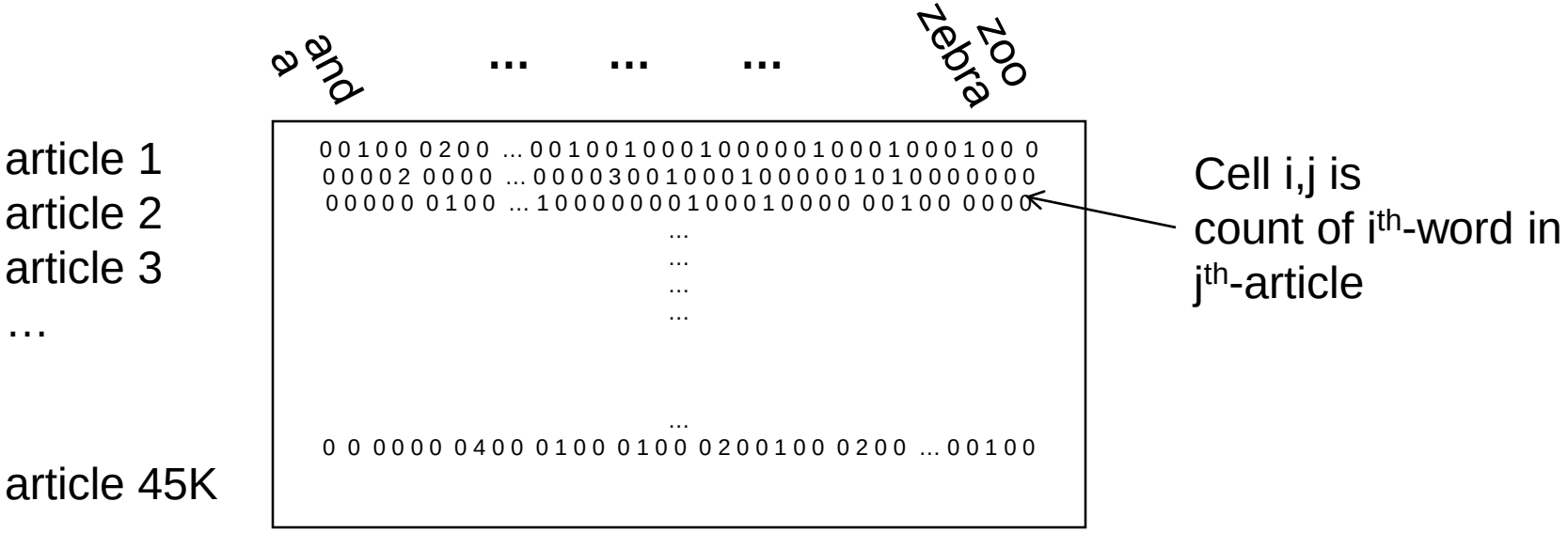


Kmeans big data example

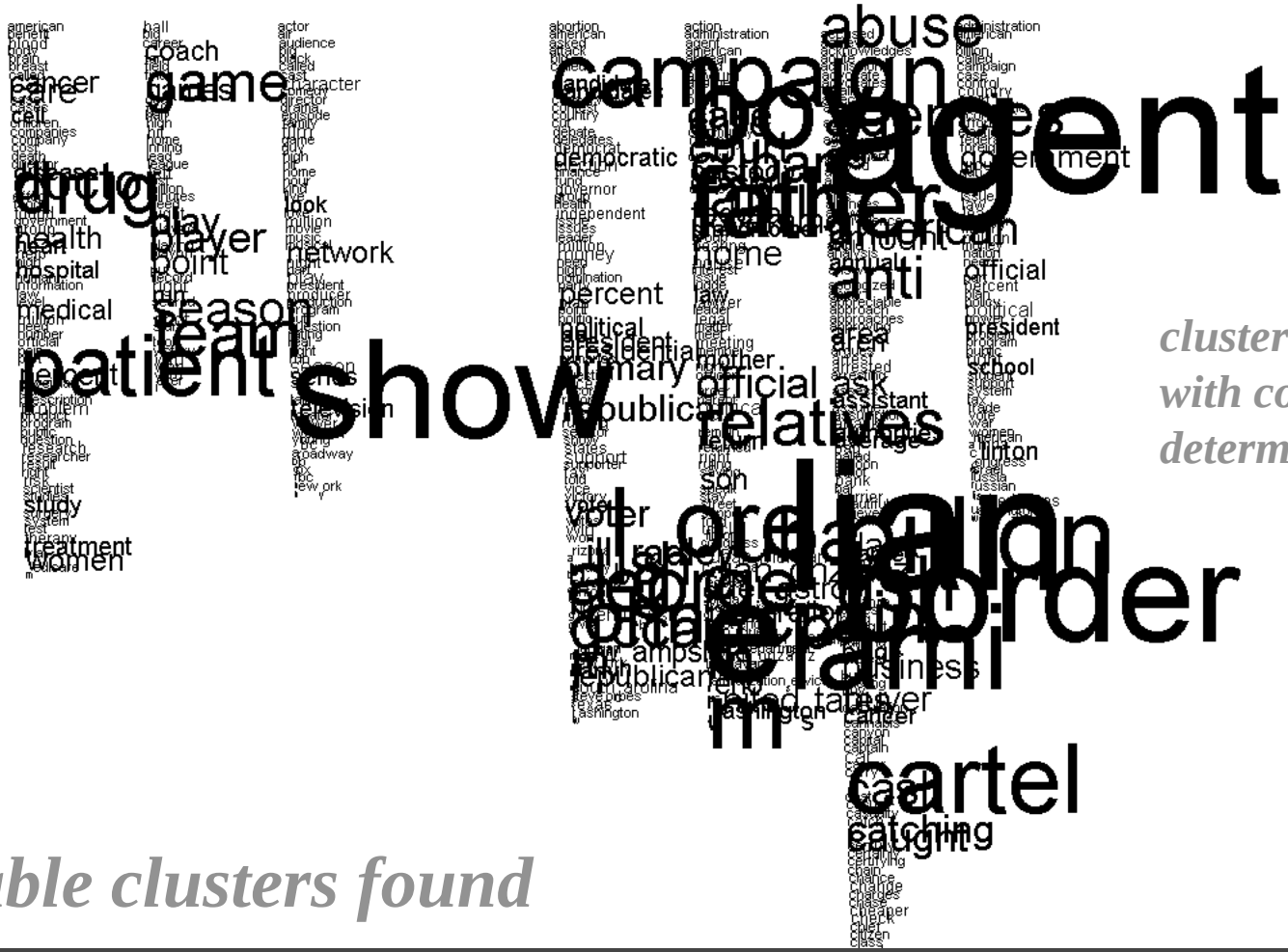
- **45,000 NYTimes articles, 102,000 unique words**

  (UCI Machine Learning repository)

- **Full Data Matrix: 45Kx102K ~ 40Gb**



Kmeans results



cluster means shown with coordinates determining fontsize

7 viable clusters found

Kmeans for image segmentation

R snippet with K=8

```
install.packages('ripa')
library('ripa')

source("http://bioconductor.org/biocLite.R")
biocLite()
biocLite("EBImage")

library('EBImage')
im=readImage('1a34086v.jpg')

library('ripa')
img=rgb2grey(im, coefs=c(0.30, 0.59, 0.11))

imgx1  =as.vector(img)
numk=8
km_imx1=kmeans(imgx1,numk,50,1);
img_km_mat =matrix(km_imx1$cluster,dim(im)[1],dim(im)[2])

display(img_km_mat/numk)
```

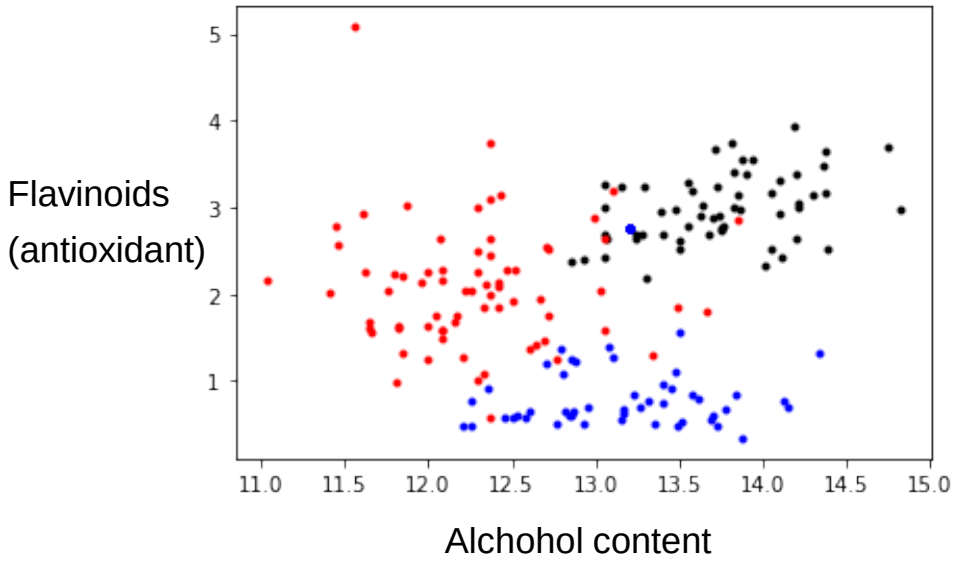
Kmeans with a Winery dataset

(UCI Machine Learning Repository)

178 observations

13 variables of wine characteristics

3 target classes that indicate which winery produced the wine



Black=58 cases
Red=71
Blue=48

Will cluster match classes?

Pause

Principle Components vs Clustering

- PCA, SVD reduces dimensions, Clustering reduces to categorical groups
- In some cases, *k* PCs ⇔ *k* clusters
- It is also useful to visualize clusters in PC space

Summary

- **Having no label doesn’t stop you from finding structure in data**
- **Unsupervised methods are somewhat related**