



Department of Computer Science
UNIVERSITY OF COLORADO **BOULDER**



Clustering

Introduction to Data Science
University of Colorado Boulder

SLIDES ADAPTED FROM LAUREN HANNAH

Clustering

Questions:

- how do we fit clusters?
- how many clusters should we use?
- how should we evaluate model fit?

K-Means

How do we fit the clusters?

- simplest method: K-means
- requires: real-valued data
- idea:
 - pick K initial cluster means
 - associate all points closest to mean k with cluster k
 - use points in cluster k to update mean for that cluster
 - re-associate points closest to new mean for k with cluster k
 - use new points in cluster k to update mean for that cluster
 - ...
 - stop when no change between updates

K-Means

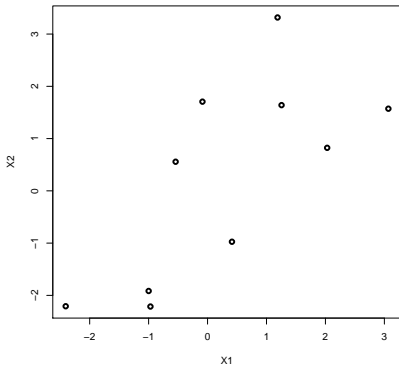
Animation at:

<http://shabal.in/visuals/kmeans/1.html>

K-Means: Example

Data:

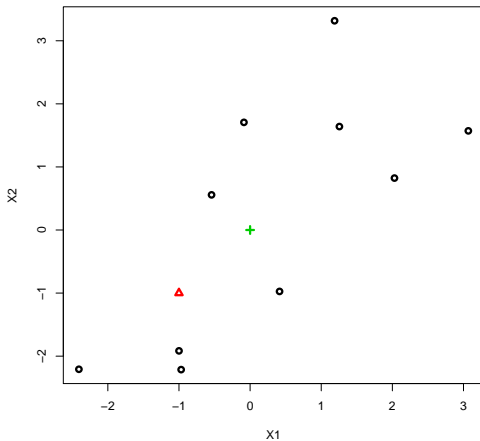
x_1	x_2
0.4	-1.0
-1.0	-2.2
-2.4	-2.2
-1.0	-1.9
-0.5	0.6
-0.1	1.7
1.2	3.3
3.1	1.6
1.3	1.6
2.0	0.8



K-Means: Example

Pick K centers (randomly):

$(-1, -1)$ and $(0, 0)$



K-Means: Example

Calculate distance between points and those centers:

x_1	x_2	$(-1, -1)$	$(0, 0)$
0.4	-1.0	1.4	1.1
-1.0	-2.2	1.2	2.4
-2.4	-2.2	1.9	3.3
-1.0	-1.9	0.9	2.2
-0.5	0.6	1.6	0.8
-0.1	1.7	2.9	1.7
1.2	3.3	4.8	3.5
3.1	1.6	4.8	3.4
1.3	1.6	3.5	2.1
2.0	0.8	3.5	2.2

```
> centers <- rbind(c(-1,-1),c(0,0))  
> dist1 <- apply(x,1,function(x) sqrt(sum((x-centers[1,])^2))  
> dist2 <- apply(x,1,function(x) sqrt(sum((x-centers[2,])^2))
```

K-Means: Example

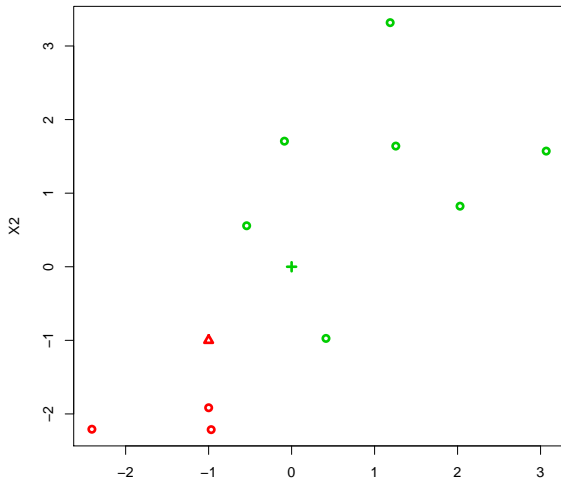
Choose mean with smaller distance:

x_1	x_2	$(-1, -1)$	$(0, 0)$
0.4	-1.0	1.4	1.1
-1.0	-2.2	1.2	2.4
-2.4	-2.2	1.9	3.3
-1.0	-1.9	0.9	2.2
-0.5	0.6	1.6	0.8
-0.1	1.7	2.9	1.7
1.2	3.3	4.8	3.5
3.1	1.6	4.8	3.4
1.3	1.6	3.5	2.1
2.0	0.8	3.5	2.2

```
> dists <- cbind(dist1, dist2)
> cluster.ind <- apply(dists, 1, which.min)
```


K-Means: Example

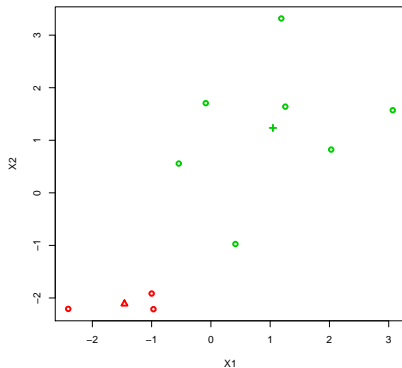
New clusters:



K-Means: Example

Refit means for each cluster:

- cluster 1: $(-1.0, -2.2)$, $(-2.4, -2.2)$, $(-1.0, -1.9)$
- new mean: $(-1.5, -2.1)$
- cluster 2: $(0.4, -1.0)$, $(-0.5, 0.6)$, $(-0.1, 1.7)$, $(1.2, 3.3)$, $(3.1, 1.6)$, $(1.3, 1.6)$, $(2.0, 0.8)$
- new mean: $(1.0, 1.2)$



K-Means: Example

Recalculate distances for each cluster:

x_1	x_2	$(-1.5, -2.1)$	$(1.0, 1.2)$
0.4	-1.0	2.2	2.3
-1.0	-2.2	0.5	4.0
-2.4	-2.2	1.0	4.9
-1.0	-1.9	0.5	3.8
-0.5	0.6	2.8	1.7
-0.1	1.7	4.1	1.2
1.2	3.3	6.0	2.1
3.1	1.6	5.8	2.0
1.3	1.6	4.6	0.5
2.0	0.8	4.6	1.1

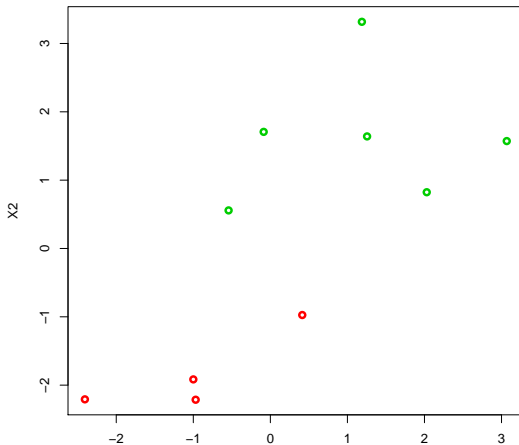
K-Means: Example

Choose mean with smaller distance:

x_1	x_2	$(-1.5, -2.1)$	$(1.0, 1.2)$
0.4	-1.0	2.2	2.3
-1.0	-2.2	0.5	4.0
-2.4	-2.2	1.0	4.9
-1.0	-1.9	0.5	3.8
-0.5	0.6	2.8	1.7
-0.1	1.7	4.1	1.2
1.2	3.3	6.0	2.1
3.1	1.6	5.8	2.0
1.3	1.6	4.6	0.5
2.0	0.8	4.6	1.1

K-Means: Example

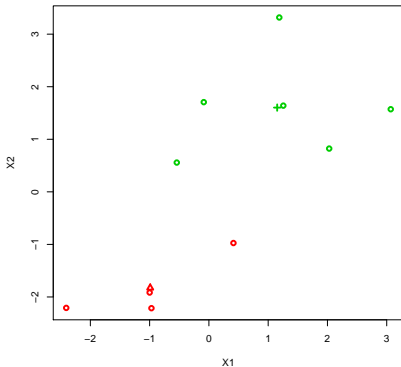
New clusters:



K-Means: Example

Refit means for each cluster:

- cluster 1: $(0.4, -1.0)$,
 $(-1.0, -2.2)$, $(-2.4, -2.2)$,
 $(-1.0, -1.9)$
- new mean: $(-1.0, -1.8)$
- cluster 2: $(-0.5, 0.6)$, $(-0.1, 1.7)$,
 $(1.2, 3.3)$, $(3.1, 1.6)$, $(1.3, 1.6)$,
 $(2.0, 0.8)$
- new mean: $(1.2, 1.6)$



K-Means: Example

Recalculate distances for each cluster:

x_1	x_2	$(-1.0, -1.8)$	$(1.2, 1.6)$
0.4	-1.0	1.6	2.7
-1.0	-2.2	0.4	4.4
-2.4	-2.2	1.5	5.2
-1.0	-1.9	0.1	4.1
-0.5	0.6	2.4	2.0
-0.1	1.7	3.6	1.2
1.2	3.3	5.6	1.7
3.1	1.6	5.3	1.9
1.3	1.6	4.1	0.1
2.0	0.8	4.0	1.2

K-Means: Example

Select smallest distance and compare these clusters with previous:

Table: New Clusters

x_1	x_2	$(-1.0, -1.8)$	$(1.2, 1.6)$
0.4	-1.0	1.6	2.7
-1.0	-2.2	0.4	4.4
-2.4	-2.2	1.5	5.2
-1.0	-1.9	0.1	4.1
-0.5	0.6	2.4	2.0
-0.1	1.7	3.6	1.2
1.2	3.3	5.6	1.7
3.1	1.6	5.3	1.9
1.3	1.6	4.1	0.1
2.0	0.8	4.0	1.2

Table: Old Clusters

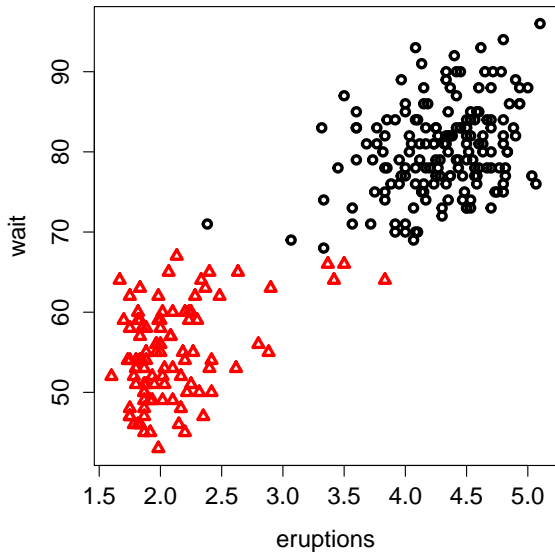
$(-1.5, -2.1)$	$(1.0, 1.2)$
2.2	2.3
0.5	4.0
1.0	4.9
0.5	3.8
2.8	1.7
4.1	1.2
6.0	2.1
5.8	2.0
4.6	0.5
4.6	1.1

K-Means in Practice

R has a function for K-means in the `stats` package; this is probably already loaded

- let's use this for the Old Faithful data

```
> library(datasets)
> faith.2 <- kmeans(faithful,2)
> names(faith.2)
> plot(faithful[,1],faithful[,2],col=faith.2$clu
+      pch=faith.2$cluster,lwd=3)
```



K-Means in R

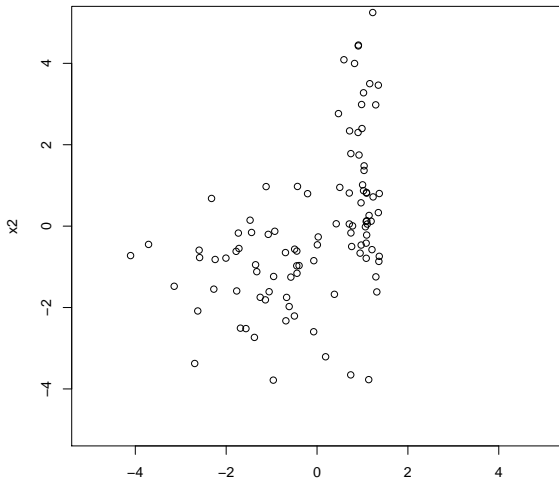
K-means can be used for *image segmentation*

- partition image into multiple segments
- find boundaries of objects
- make art



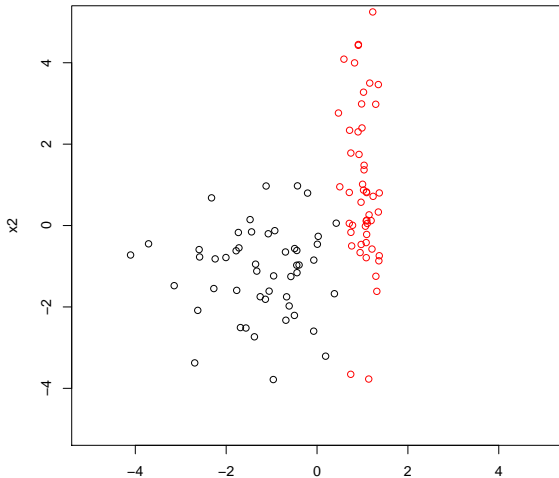
K-Means Clustering

What is our data look like this?



K-Means Clustering

True clustering:



K-Means Clustering

K-means clustering ($K = 2$):

