

# Introduction to Machine Learning for Social Science

## Class 13: Clustering

Rochelle Terman

Postdoctoral Fellow  
Center for International Security Cooperation  
Stanford University

February 27, 2018



Today (and Tuesday): Cluster press releases

**Goal:** partition documents such that:

- **similar** documents are together
- **dissimilar** documents are apart

**Method:** Clustering methods

**Game Plan:**

- 1) What makes two data points (i.e. documents) similar?
- 2) How do we find a good partition?
- 3) How do we interpret the clusters?

## Key Terms:

- (Multidimensional) Space
- Distance
- Euclidean Distance
- Cosine Distance
- Cluster Analysis / Clustering
- K-means
- Centroid

# K-Means Clustering

**K-means** clustering is popular method to partition a data set into  $K$  distinct, non-overlapping clusters.

# K-Means Clustering

**K-means** clustering is popular method to partition a data set into  $K$  distinct, non-overlapping clusters.

## Inputs

- 1 A document term matrix (or any multidimensional dataset)
- 2  $K$ : the desired number of clusters.

# K-Means Clustering

**K-means** clustering is popular method to partition a data set into  $K$  distinct, non-overlapping clusters.

## Inputs

- 1 A document term matrix (or any multidimensional dataset)
- 2  $K$ : the desired number of clusters.

Then the  $K$ -means algorithm will assign each observation into exactly one of the  $K$  clusters.

# K-Means Clustering

**K-means** clustering is popular method to partition a data set into  $K$  distinct, non-overlapping clusters.

## Inputs

- 1 A document term matrix (or any multidimensional dataset)
- 2  $K$ : the desired number of clusters.

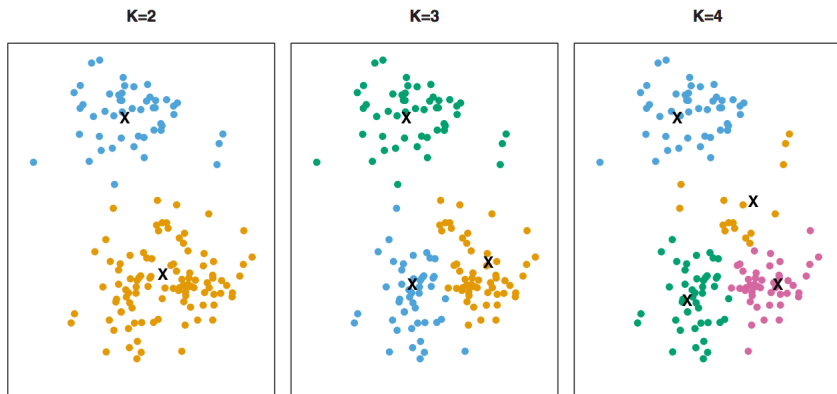
Then the  $K$ -means algorithm will assign each observation into exactly one of the  $K$  clusters.

## Outputs

- 1  $C_k$ : The set of observations assigned to each cluster.
- 2  $\mu_k$ : The mean for each  $K$  – a vector representing the average values of all observations in that cluster. Also called **centroid**.



# K-Means Clustering



# K-Means Clustering: Outputs

**Centroid ( $\mu_k$ ):** The mean for each  $K$  – a vector representing the average values of all observations in that cluster.

# K-Means Clustering: Outputs

**Centroid ( $\mu_k$ ):** The mean for each  $K$  – a vector representing the average values of all observations in that cluster.

Consider the following cluster with two vectors:

$$\mathbf{x}_1 = [1, 0, 3]$$

$$\mathbf{x}_2 = [0, 4, 1]$$

# K-Means Clustering: Outputs

**Centroid ( $\mu_k$ ):** The mean for each  $K$  – a vector representing the average values of all observations in that cluster.

Consider the following cluster with two vectors:

$$\mathbf{x}_1 = [1, 0, 3]$$

$$\mathbf{x}_2 = [0, 4, 1]$$

Then its mean is:

$$\mu = [\text{mean}(x_{1,1}, x_{2,1}), \text{mean}(x_{1,2}, x_{2,2}), \text{mean}(x_{1,3}, x_{2,3})]$$

# K-Means Clustering: Outputs

**Centroid ( $\mu_k$ ):** The mean for each  $K$  – a vector representing the average values of all observations in that cluster.

Consider the following cluster with two vectors:

$$\mathbf{x}_1 = [1, 0, 3]$$

$$\mathbf{x}_2 = [0, 4, 1]$$

Then its mean is:

$$\mu = [\text{mean}(x_{1,1}, x_{2,1}), \text{mean}(x_{1,2}, x_{2,2}), \text{mean}(x_{1,3}, x_{2,3})]$$

$$\mu = [0.5, 2, 2]$$

# K-Means Clustering: Outputs

**Centroid ( $\mu_k$ ):** The mean for each  $K$  – a vector representing the average values of all observations in that cluster.

Consider the following cluster with two vectors:

$$\mathbf{x}_1 = [1, 0, 3]$$

$$\mathbf{x}_2 = [0, 4, 1]$$

Then its mean is:

$$\mu = [\text{mean}(X_{1,1}, X_{2,1}), \text{mean}(X_{1,2}, X_{2,2}), \text{mean}(X_{1,3}, X_{2,3})]$$

$$\mu = [0.5, 2, 2]$$

The K-means algorithm will assign each observation to the cluster with the closest mean.

# K-Means Clustering: Example

**Goal:** Cluster the following documents:

- I like to eat broccoli and bananas.
- I eat a banana smoothie for breakfast.
- Hamsters and kittens are cute.
- She adopted a cute kitten.

# K-Means Clustering: Example

## Inputs

### 1 A document term matrix

	adopt	banana	breakfast	broccoli	cute	eat	hamster	kitten	like	smoothi
1	0	1	0	1	0	1	0	0	1	0
2	0	1	1	0	0	1	0	0	0	1
3	0	0	0	0	1	0	1	1	0	0
4	1	0	0	0	1	0	0	1	0	0



# K-Means Clustering: Example

## Inputs

1 A document term matrix

	adopt	banana	breakfast	broccoli	cute	eat	hamster	kitten	like	smoothie
1	0	1	0	1	0	1	0	0	1	0
2	0	1	1	0	0	1	0	0	0	1
3	0	0	0	0	1	0	1	1	0	0
4	1	0	0	0	1	0	0	1	0	0

2  $K$ : 2

# K-Means Clustering: Example

## Inputs

- 1 A document term matrix

	adopt	banana	breakfast	broccoli	cute	eat	hamster	kitten	like	smoothie
1	0	1	0	1	0	1	0	0	1	0
2	0	1	1	0	0	1	0	0	0	1
3	0	0	0	0	1	0	1	1	0	0
4	1	0	0	0	1	0	0	1	0	0

- 2  $K$ : 2

## Outputs

- 1  $C_k$ : Cluster assignment:
  - $C_1$ : [1, 2]
  - $C_2$ : [3, 4]

# K-Means Clustering: Example

## Inputs

- 1 A document term matrix

	adopt	banana	breakfast	broccoli	cute	eat	hamster	kitten	like	smoothi
1	0	1	0	1	0	1	0	0	1	0
2	0	1	1	0	0	1	0	0	0	1
3	0	0	0	0	1	0	1	1	0	0
4	1	0	0	0	1	0	0	1	0	0

- 2  $K$ : 2

## Outputs

- 1  $C_k$ : Cluster assignment:

- $C_1$ : [1, 2]
- $C_2$ : [3, 4]

- 2  $\mu_k$ : Cluster means / centroids:

	adopt	banana	breakfast	broccoli	cute	eat	hamster	kitten	like	smoothi
$\mu_1$	0.0	1.0	0.5	0.5	0.0	1.0	0.0	0.0	0.5	0.5
$\mu_2$	0.5	0.0	0.0	0.0	1.0	0.0	0.5	1.0	0.0	0.0

# K-Means Clustering

A chicken and egg problem:

# K-Means Clustering

A chicken and egg problem:

- Means  $\rightsquigarrow$  Assignments

# K-Means Clustering

A chicken and egg problem:

- Means  $\rightsquigarrow$  Assignments
- Assignments  $\rightsquigarrow$  Means

# K-Means Clustering

A chicken and egg problem:

- Means  $\rightsquigarrow$  Assignments
- Assignments  $\rightsquigarrow$  Means

How do we find a good partition?

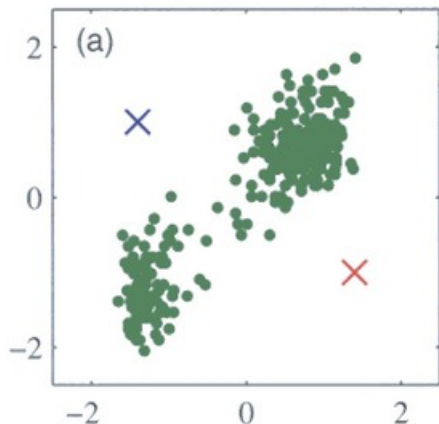
# K-Means Clustering: Algorithm

- 1) Randomly initialize  $K$  cluster centroids  $(\mu_1, \mu_2, \dots, \mu_k)$  in random locations.
- 2) Repeat:
  - **Assignment:** Assign each observation  $\mathbf{X}$  to cluster with closest mean  $\mu_k$ .
  - **Update:** Calculate new centroids  $\mu_k$  by averaging all points assigned to each cluster.

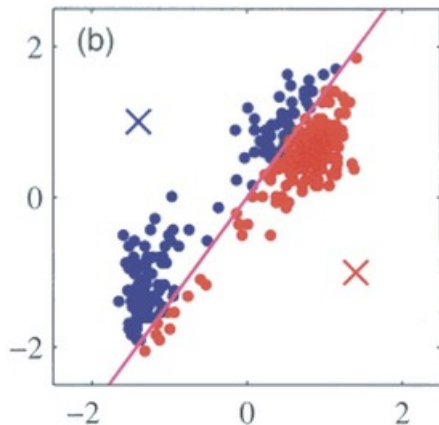
Stop when cluster assignments stop changing.



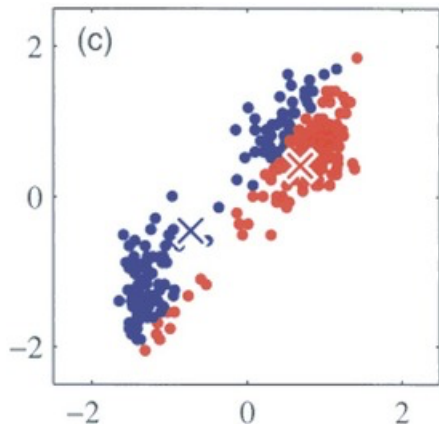
# K-Means Clustering: Algorithm



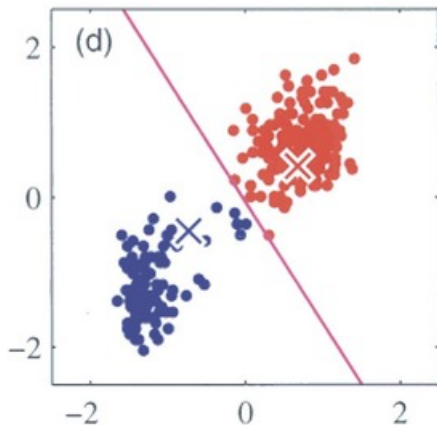
# K-Means Clustering: Algorithm



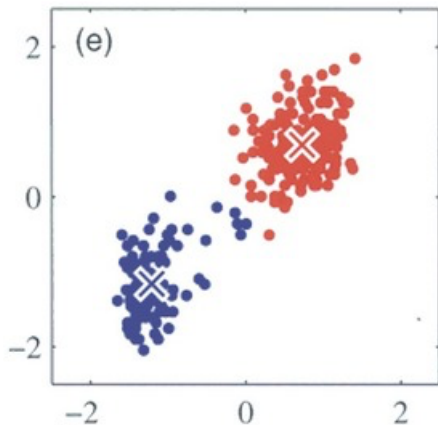
# K-Means Clustering: Algorithm



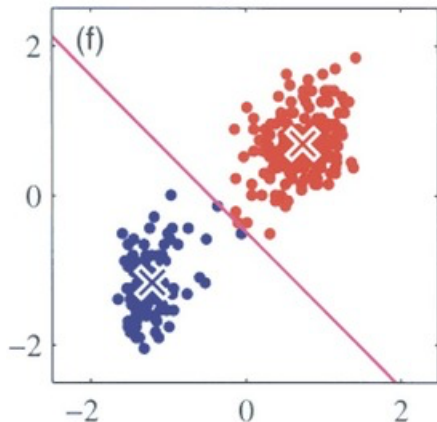
# K-Means Clustering: Algorithm



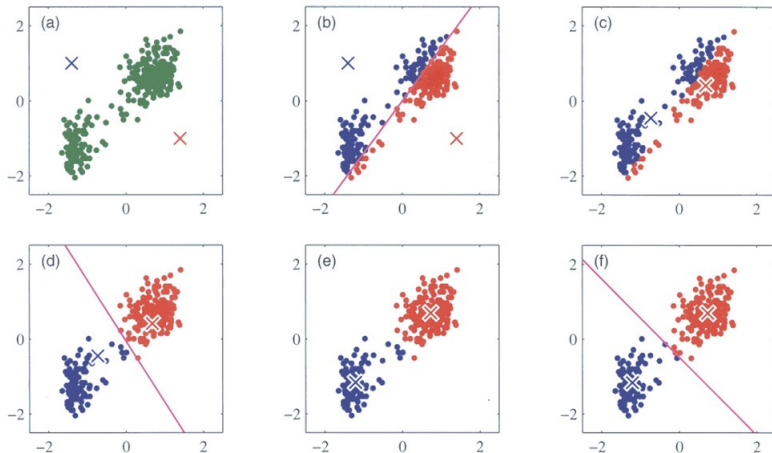
# K-Means Clustering: Algorithm



# K-Means Clustering: Algorithm



# K-Means Clustering: Algorithm



# K-Means Clustering: Decisions

Small Decisions with Big Consequences:



# K-Means Clustering: Decisions

## Small Decisions with Big Consequences:

### 1) How should we preprocess the data?

- k-means are very sensitive to feature scaling / weighting.
- Common to normalize the DTM in some way, e.g. by dividing each vector by the vector length.

# K-Means Clustering: Decisions

## Small Decisions with Big Consequences:

### 1) How should we preprocess the data?

- k-means are very sensitive to feature scaling / weighting.
- Common to normalize the DTM in some way, e.g. by dividing each vector by the vector length.

### 2) How to chose $K$ ?

- User must assign the number of clusters ( $K$ )
- Different values of  $K$  will lead to different partitions.

# K-Means Clustering: Decisions

## Small Decisions with Big Consequences:

### 1) How should we preprocess the data?

- k-means are very sensitive to feature scaling / weighting.
- Common to normalize the DTM in some way, e.g. by dividing each vector by the vector length.

### 2) How to chose $K$ ?

- User must assign the number of clusters ( $K$ )
- Different values of  $K$  will lead to different partitions.

### 3) Random starting values!

- Results will depend on the initial (random) cluster centroid assignment (in step 1).
- Important to run the algorithm multiple times from different random starting values.

# K-Means Clustering: Decisions

## Small Decisions with Big Consequences:

### 1) How should we preprocess the data?

- k-means are very sensitive to feature scaling / weighting.
- Common to normalize the DTM in some way, e.g. by dividing each vector by the vector length.

### 2) How to chose $K$ ?

- User must assign the number of clusters ( $K$ )
- Different values of  $K$  will lead to different partitions.

### 3) Random starting values!

- Results will depend on the initial (random) cluster centroid assignment (in step 1).
- Important to run the algorithm multiple times from different random starting values.

## How do we decide?

# K-Means Clustering: How do we decide?

What makes a good partition?

# K-Means Clustering: How do we decide?

What makes a good partition?

Two kinds of validation criteria:

# K-Means Clustering: How do we decide?

## What makes a good partition?

Two kinds of validation criteria:

### 1 Quantitative evaluation:

- A good clustering is one for which the within-cluster variation is as small as possible.

# K-Means Clustering: How do we decide?

## What makes a good partition?

Two kinds of validation criteria:

### 1 Quantitative evaluation:

- A good clustering is one for which the within-cluster variation is as small as possible.

### 2 Qualitative evaluation:

- A good clustering is one for which clusters are substantially / semantically interpretable.



**Quantitative evaluation:** within-cluster variation is as small as possible.

- **Within-cluster variation:** a measure of the amount by which the observations within a cluster differ from each other.
- Common metric: **Sum of Squared Euclidean Distance**

For a given document  $\mathbf{X}$  in cluster  $k$ , the **squared Euclidean distance** is:

$$D(\mathbf{X}, \mu_k)^2 = \sum_{p=1}^P (x_p - \mu_{kp})^2$$

For a given document  $\mathbf{X}$  in cluster  $k$ , the **squared Euclidean distance** is:

$$D(\mathbf{X}, \mu_k)^2 = \sum_{p=1}^P (x_p - \mu_{kp})^2$$

The **within-cluster sum of squared distances** for a given cluster  $C_k$  is:

$$W(C_k) = \sum_{i \in C_k} D(\mathbf{X}_i, \mu_k)^2$$

For a given document  $\mathbf{X}$  in cluster  $k$ , the **squared Euclidean distance** is:

$$D(\mathbf{X}, \mu_k)^2 = \sum_{p=1}^P (x_p - \mu_{kp})^2$$

The **within-cluster sum of squared distances** for a given cluster  $C_k$  is:

$$W(C_k) = \sum_{i \in C_k} D(\mathbf{X}_i, \mu_k)^2$$

Thus our goal is to minimize the **total within-cluster sum of squares**:

$$\sum_{k=1}^K W(C_k)$$

**Qualitative evaluation:** clusters are substantially / semantically interpretable.

How do we interpret the clusters?

**Qualitative evaluation:** clusters are substantially / semantically interpretable.

How do we interpret the clusters?

### 1 Manual identification

- Sample set of documents from same cluster
- Read documents
- Assign cluster “label” by hand
  - I like to eat broccoli and bananas.  $\rightsquigarrow$  “food”
  - Hamsters and kittens are cute.  $\rightsquigarrow$  “pets”

**Qualitative evaluation:** clusters are substantially / semantically interpretable.

How do we interpret the clusters?

### 1 Manual identification

- Sample set of documents from same cluster
- Read documents
- Assign cluster “label” by hand
  - I like to eat broccoli and bananas.  $\rightsquigarrow$  “food”
  - Hamsters and kittens are cute.  $\rightsquigarrow$  “pets”

### 2 Automatic identification

- Use methods to identify separating words between clusters
- Use these to help infer differences across clusters

**Qualitative evaluation:** clusters are substantially / semantically interpretable.

How do we interpret the clusters?

**1** Manual identification

- Sample set of documents from same cluster
- Read documents
- Assign cluster “label” by hand
  - I like to eat broccoli and bananas.  $\rightsquigarrow$  “food”
  - Hamsters and kittens are cute.  $\rightsquigarrow$  “pets”

**2** Automatic identification

- Use methods to identify separating words between clusters
- Use these to help infer differences across clusters

**3** Be **Transparent**

- Provide documents + code
- Detail labeling procedures
- Acknowledge ambiguity



Today (and Tuesday): Cluster press releases

**Goal:** partition documents such that:

- **similar** documents are together
- **dissimilar** documents are apart

**Method:** Clustering methods

**Game Plan:**

- 1) What makes two data points (i.e. documents) similar?
- 2) How do we find a good partition?
- 3) How do we interpret the clusters?

## Key Terms:

- (Multidimensional) Space
- Distance
- Euclidean Distance
- Cosine Distance
- Cluster Analysis / Clustering
- K-means
- Centroid