Introduction to Machine Learning for Social Scientists

Class 9: Regularization 2

Edgar Franco Vivanco

Stanford University
Department of Political Science

edgarf1@stanford.edu

Summer 2018

Logistics

- Homework 4 due today
- Homework 5 (Available today, due Wed 15th)
- Keep working on your team projects

Last class

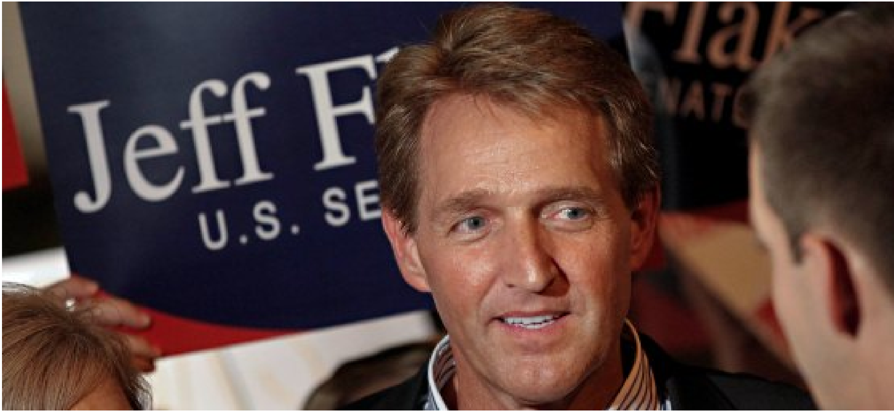
Key terms

- Multidimensional space
- Distance Metrics
- Euclidean
- Cosine
- Multidimensional scaling

Key functions

- dist
- cosine
- apply
- cmdscale

Questions?


Jeff F
U.S. SE

Today : 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

Key Functions:

- kmeans

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

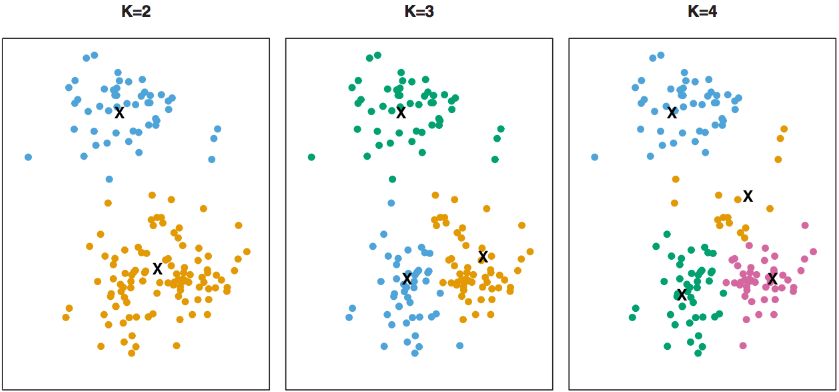
1. A document term matrix (or any multidimensional dataset)
2. K: the desired number of clusters.

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. μ_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 (μ_k): The mean for each K – a vector representing the average values of all observations in that cluster.

K-Means Clustering: Outputs

Centroid (μ_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:

$$\begin{aligned} \boldsymbol{X_1} &= [1, 0, 3] \\ \boldsymbol{X_2} &= [0, 4, 1] \end{aligned}$$

K-Means Clustering: Outputs

Centroid (μ_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:

$$\begin{aligned} \boldsymbol{X_1} &= [1, 0, 3] \\ \boldsymbol{X_2} &= [0, 4, 1] \end{aligned}$$

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 (μ_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:

$$\begin{aligned} \boldsymbol{X_1} &= [1, 0, 3] \\ \boldsymbol{X_2} &= [0, 4, 1] \end{aligned}$$

Then its mean is:

$$\begin{aligned} \mu &= [\text{mean}(X_{1,1}, X_{2,1}), \text{mean}(X_{1,2}, X_{2,2}), \text{mean}(X_{1,3}, X_{2,3})] \\ \mu &= [0.5, 2, 2] \end{aligned}$$

K-Means Clustering: Outputs

Centroid (μ_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:

$$\begin{aligned} \boldsymbol{X_1} &= [1, 0, 3] \\ \boldsymbol{X_2} &= [0, 4, 1] \end{aligned}$$

Then its mean is:

$$\begin{aligned} \mu &= [\text{mean}(X_{1,1}, X_{2,1}), \text{mean}(X_{1,2}, X_{2,2}), \text{mean}(X_{1,3}, X_{2,3})] \\ \mu &= [0.5, 2, 2] \end{aligned}$$

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	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

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:
   - $\boldsymbol{C_1}$: [1, 2]
   - $\boldsymbol{C_2}$: [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:
   - $\boldsymbol{C_1}$: [1, 2]
   - $\boldsymbol{C_2}$: [3, 4]
2. μ_k: Cluster means / centroids:

	adopt	banana	breakfast	broccoli	cute	eat	hamster	kitten	like	smoothi
μ_1	0.0	1.0	0.5	0.5	0.0	1.0	0.0	0.0	0.5	0.5
μ_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

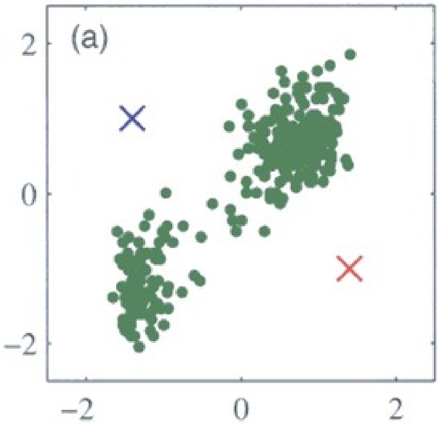
How do we find a good partition?

K-Means Clustering: Algorithm

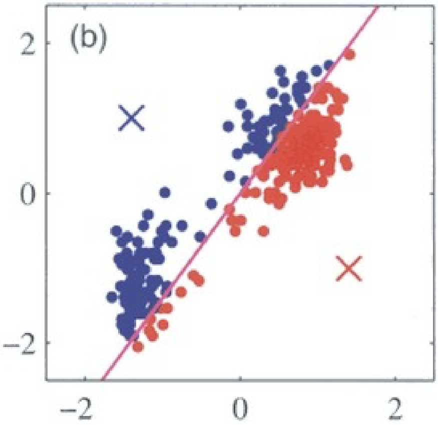
1) Randomly initialize K cluster centroids $(\mu_1, \mu_2, \cdots, \mu_k)$ in random locations.
2) Repeat:
   - Assignment: Assign each observation $\boldsymbol{X}$ to cluster with closest mean μ_k.
   - Update: Calculate new centroids μ_k by averaging all points assigned to each cluster.

   Stop when cluster assignments stop changing.

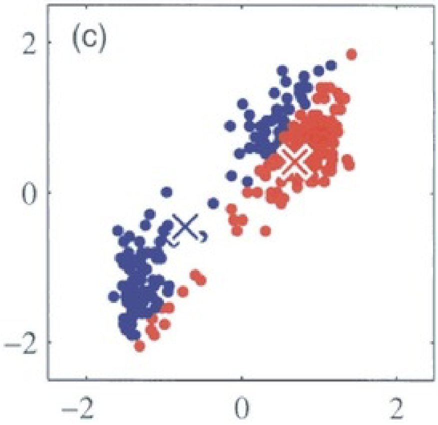
K-Means Clustering: Algorithm



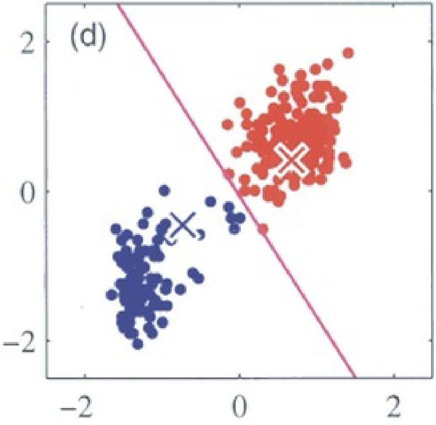
K-Means Clustering: Algorithm



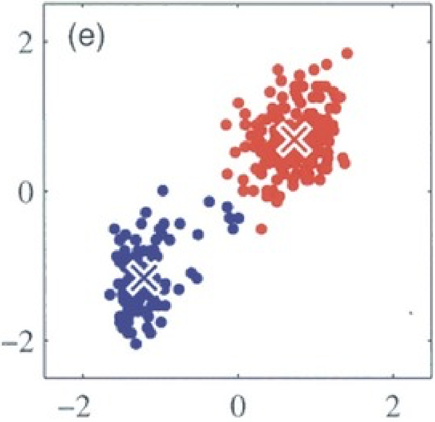
K-Means Clustering: Algorithm



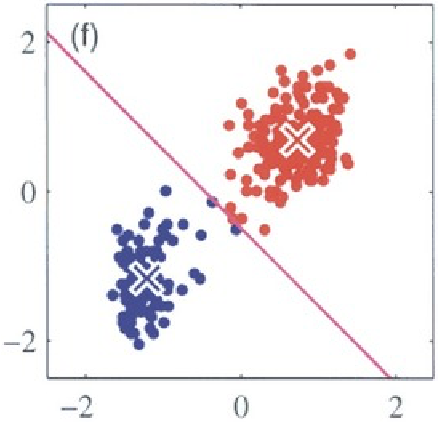
K-Means Clustering: Algorithm



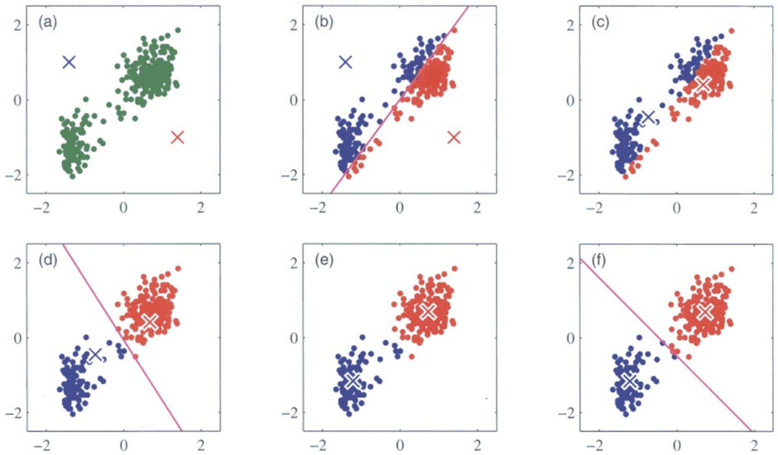
K-Means Clustering: Algorithm



K-Means Clustering: Algorithm



K-Means Clustering: Algorithm



A simple illustration:

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Group the dataset in two clusters. Let A and B be the values of the two individuals further apart (Euclidean distance) :

	Individual	Mean vector(centroid)
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

The remaining individuals are now examined in sequence and allocated to the cluster they are closest.

	Cluster 1		Cluster 2	
Step	Individual	Mean Vector(centroid)	Individual	Mena Vector (Centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1,2	(1.2, 1.5)	4	(5.0, 7.0)
3	1,2,3	(1.8, 2.3)	4	(5.0, 7.0)
4	1,2,3	(1.8, 2.3)	4,5	(4.2, 6.0)
5	1,2,3	(1.8, 2.3)	4,5,6	(4.3, 5.7)
6	1,2,3	(1.8, 2.3)	4,5,6,7	(4.1, 5.4)

Now the clusters look like this:

	Individual	Mean vector (centroid)
Cluster 1	1,2,3	(1.8, 2.3)
Cluster 2	4,5,6,7	(4.1, 5.4)

But we can only be sure that each individual has been assigned to the right cluster by comparing distances to its own cluster mean:

Individual	Distance to mean of c1	Distance to mean of c2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Only individual 3 is closer to the mean of the opposite cluster (2) than its own cluster (1). Thus, individual 3 is relocated:

But we can only be sure that each individual has been assigned to the right cluster by comparing distances to its own cluster mean:

Individual	Distance to mean of c1	Distance to mean of c2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Only individual 3 is closer to the mean of the opposite cluster (2) than its own cluster (1). Thus, individual 3 is relocated:

	Individual	Mean vector (centroid)
Cluster 1	1,2	(1.3, 1.5)
Cluster 2	3, 4,5,6,7	(3.9, 5.1)

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 $\boldsymbol{X}$ in cluster k, the squared Euclidean distance is:

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

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

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

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

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

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

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

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

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

Thus our goal is to minimize the total within-cluster sum of squares (total within-cluster varition, summed over all K clusters is as small as possible.):

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

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

What is the right number of clusters?

Several possibilities

- Direct methods: Optimizing criterion:
  - *Elbow method:* Compares wss and find a tipping point
  - *Silhouette method:* Calculate the average silhouette of observations (avg.sil). That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering.
- : Statistical testing methods:
  - *Gap method:* The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data.

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?

NEXT

- Homework 4 and 5
- General Overview
- Readings on algorithmic bias
- Additional office hours
- Group presentations