Machine Learning CS 4641



# **Clustering Analysis and K-Means**

Nakul Gopalan Georgia Tech

Some of the slides are based on slides from Chao Zhang, Mahdi Roozbahani, and Le Song.

## Outline

- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

#### **Clustering Images**





Divide object into groups, and objects within a group are more similar than those outside the group







#### **Clustering Hand Digits**

D  $\mathbf{q}$ Э з  $\overline{2}$ っ D ο З а н 2 a ч з з  $\sim$ з З J ŝ R oq 

#### **Clustering is Subjective**



#### What is consider similar/dissimilar?

#### Clustering is subjective



Simpson's Family



School Employees



Females

Males

#### Are they similar or not?



# So What is Clustering in General?

- You pick your similarity/dissimilarity function
- The algorithm figures out the grouping of objects based on the chosen similarity/dissimilarity function
  - Points within a cluster is similar
  - Points across clusters are not so similar
- Issues for clustering
  - How to represent objects? (Vector space? Normalization?)
  - What is a similarity/dissimilarity function for your data?
  - What are the algorithm steps?

## Outline

- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

## **Properties of Similarity Function**

- Desired properties of dissimilarity function
  - Symmetry: d(x, y) = d(y, x)
    - Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"
  - Positive separability: d(x, y) = 0, if and only if x = y
    - Otherwise there are objects that are different, but you cannot tell apart
  - Triangular inequality:  $d(x, y) \le d(x, z) + d(z, y)$ 
    - Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"

#### **Distance Functions for Vectors**

Suppose two data points, both in R<sup>d</sup>

• 
$$x = (x_1, x_2, ..., x_d)^\top$$
  
•  $y = (y_1, y_2, ..., y_d)^\top$ 

• Euclidean distance: 
$$d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

- Minkowski distance:  $d(x, y) = \sqrt[p]{\sum_{i=1}^{d} (x_i y_i)^p}$ 
  - Euclidean distance: p = 2
  - Manhattan distance: p = 1,  $d(x, y) = \sum_{i=1}^{d} |x_i y_i|$

• "inf"-distance: 
$$p = \infty$$
,  $d(x, y) = \max_{i=1}^{d} |x_i - y_i|$ 

#### Example



• Euclidean distance:  $\sqrt{4^2 + 3^2} = 5$ 

• Manhattan distance: 4 + 3 = 7

• "inf"-distance: 
$$max{4,3} = 4$$

#### Some problems with metric distances





d(x,y) and d(x,z)?

Curse of dimensionality

#### Hamming Distance

- Manhattan distance is also called Hamming distance when all features are binary
  - Count the number of difference between two binary vectors

• Example,  $x, y \in \{0,1\}^{17}$ 

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	0	1	1	0	0	1	0	0	1	0	0	1	1	1	0	0	1
<u>y</u>	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1

d(x,y)=5

#### **Edit Distance**

 Transform one of the objects into the other, and measure how much effort it takes

# x INTE\*NTION | | | | | | | | | | | y \*EXECUTION dss is

d: deletion (cost 5)s: substitution (cost 1)i: insertion (cost 2)

$$d(x, y) = 5 \times 1 + 3 \times 1 + 1 \times 2 = 10$$

## Outline

- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

# Results of K-Means Clustering:



Image

Clusters on intensity

Clusters on color

K-means clustering using intensity alone and color alone





Image

Clusters on color

#### K-means using color alone, 11 segments (clusters)



\* Pictures from Mean Shift: A Robust Approach toward Feature Space Analysis, by D. Comaniciu and P. Meer http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

#### **K-Means Algorithm**

Initialize k cluster centers, {c<sup>1</sup>, c<sup>2</sup>, ..., c<sup>k</sup>}, randomly

Do

- Decide the cluster memberships of each data point,  $x^i$ , by assigning it to the nearest cluster center (cluster assignment)  $\pi(i) = argmin_{j=1,...,k} \|x^i - c^j\|^2$
- Adjust the cluster centers (center adjustment)

$$c^{j} = \frac{1}{|\{i:\pi(i)=j\}|} \sum_{i:\pi(i)=j} x^{i}$$

While any cluster center has been changed

#### **K-Means Algorithm**



Visualizing K-Means Clustering











## Outline

- Clustering
- Distance Function
- K-Means Algorithm
- Analysis of K-Means

#### Questions

- Will different initialization lead to different results?
  - Yes
  - No
  - Sometimes

- Will the algorithm always stop after some iteration?
  - Yes
  - No (we have to set a maximum number of iterations)
  - Sometimes

#### Formal Statement of the Clustering Problem

- Given n data points,  $\{x^1, x^2, \dots, x^n\} \in \mathbb{R}^d$
- Find k cluster centers,  $\{c^1, c^2, \dots, c^k\} \in \mathbb{R}^d$
- And assign each data point *i* to one cluster,  $\pi(i) \in \{1, ..., k\}$
- Such that the averaged square distance from each data point to respective cluster center(distortion metric) is minimum:

$$\min_{c,\pi} \frac{1}{n} \sum_{i=1}^{n} \|x^{i} - c^{\pi(i)}\|^{2}$$

## Clustering is <u>NP-Hard</u>

• Find k cluster centers,  $\{c^1, c^2, ..., c^k\} \in R^d$ , and assign each data point i to one cluster,  $\pi(i) \in \{1, ..., k\}$ , to minimize



- A search problem over the space of discrete assignments
  - For all N data point together, there are k N possibility
  - The cluster assignment determines cluster centers, and vice versa



For all n data point together, there are k n possibility

 $X = \{A,B,C\}$ n=3 (data points)

k=2 clusters of two members

Cluster 1 Cluster 2

#### **Convergence of K-Means**

Will kmeans objective oscillate?

$$\frac{1}{n} \sum_{i=1}^{n} \|x^{i} - c^{\pi(i)}\|^{2}$$

- The minimum value of the objective is finite
- Each iteration of kmeans algorithm decrease the objective
  - Cluster assignment step decreases objective
    - $\pi(i) = argmin_{j=1,...,k} \|x^i c^j\|^2$  for each data point *i*
  - Center adjustment step decreases objective

• 
$$c^{j} = \frac{1}{|\{i:\pi(i)=j\}|} \sum_{i:\pi(i)=j} x^{i} = argmin_{c} \sum_{i:\pi(i)=j} ||x^{i} - c||^{2}$$

#### **Time Complexity**

 Assume computing distance between two instances is O(d) where d is the dimensionality of the vectors.

- Reassigning clusters for all datapoints:
  - O(kn) distance computations (when there is one feature)
  - O(knd) (when there is d features)
- Computing centroids: Each instance vector gets added once to some centroid (Finding centroid for each feature): O(nd).
- Assume these two steps are each done once for I iterations: O(Iknd).

#### How to Choose K?



Elbow method

Number of Clusters

**Distortion score**: computing the sum of squared distances from each point to its assigned center

Image credit: Dileka Madushan.

#### k-Nearest Neighbors

#### • Algorithm:

- Find k examples  $\{x_i, t_i\}$  closest to the test instance x
- Classification output is majority class from the set of k instances
- Non-parametric
- Infinite VC dimension
- Dependent on distance matric
- Hard in higher dimensions

#### Takeaways

- Clustering is distance matric dependent
- K-means converges with every step to the minimum distortion metric
- Ideal value of number of clusters( k ) can be identified using the distortion metric for different values of k.