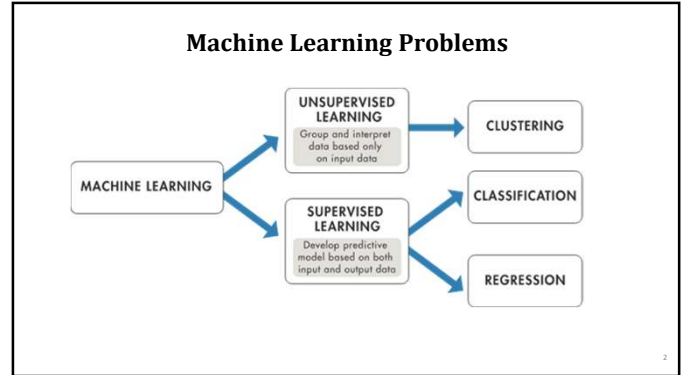


[illegible]


1



2

High Dimensional Data

- Given a cloud of data points we want to understand its structure



The image displays a large number of small, dark, irregularly shaped data points scattered across a white background. These points are organized into two primary clusters. The cluster on the left is more elongated and spread out, while the cluster on the right is more compact and rounded. There is a clear gap between the two clusters, suggesting two distinct classes or categories of data. The points themselves appear to be composed of many small, overlapping shapes, giving them a noisy or textured appearance.

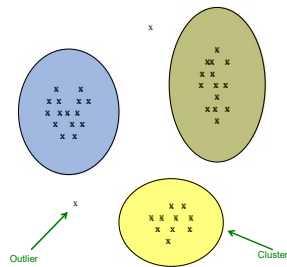
3

The Problem of Clustering

- Given a **set of points**, with a notion of **distance** between points, **group the points** into some number of *clusters*, so that
 - Members of a cluster are **close/similar** to each other
 - Members of different clusters are **dissimilar**
- Usually:**
 - Points are in a high-dimensional space
 - Similarity is defined using a distance measure
 - Euclidean, Cosine, Jaccard, Edit distance, ...

4

Example: Clusters & Outliers



5

Clustering is a Hard Problem!



6

Why is it hard?

- Clustering in two dimensions looks easy
- Clustering small amounts of data looks easy
- And in most cases, looks are **not** deceiving
- Many applications involve not 2, but 10 or 10,000 dimensions
- **High-dimensional spaces look different:** Almost all pairs of points are at about the same distance

7

Clustering Problem: Galaxies

- A catalog of 2 billion "sky objects" represents objects by their radiation in 7 dimensions (frequency bands)
- **Problem:** Cluster into similar objects, e.g., galaxies, nearby stars etc.
- Sloan Digital Sky Survey



8

Clustering Problem: Music CDs

- **Intuitively: Music divides into categories, and customers prefer a few categories**
 - But what are categories really?
- Represent a CD by a set of customers who bought it:
- Similar CDs have similar sets of customers, and vice-versa

9

Clustering Problem: Music CDs

Space of all CDs:

- Think of a space with one dim. for each customer
 - Values in a dimension may be 0 or 1 only
 - A CD is a point in this space (x_1, x_2, \dots, x_k) , where $x_i = 1$ iff the i^{th} customer bought the CD
- For Amazon, the dimension is tens of millions
- **Task:** Find clusters of similar CDs

10

Clustering Problem: Documents

Finding topics:

- Represent a document by a vector (x_1, x_2, \dots, x_k) , where $x_i = 1$ iff the i^{th} word (in some order) appears in the document
 - It actually doesn't matter if k is infinite; i.e., we don't limit the set of words
- **Documents with similar sets of words may be about the same topic**

11

Cosine, Jaccard, and Euclidean

- **As with CDs we have a choice when we think of documents as sets of words:**
 - **Sets as vectors:** Measure similarity by the **cosine distance**
 - **Sets as sets:** Measure similarity by the **Jaccard distance**
 - **Sets as points:** Measure similarity by **Euclidean distance**

12

Other Distance Measures

- **City-block distance (Manhattan distance)**
 - Add absolute value of differences
- **Cosine similarity** - Measure angle formed by the two samples (with the origin)
- **Jaccard distance** - Determine percentage of exact matches between the samples
- **Others**

17

13

Distance Measures

Minkowsky:	Euclidean:	Manhattan / city-block:
$D(x,y) = \left(\sum_{i=1}^m x_i - y_i ^p \right)^{1/p}$	$D(x,y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$	$D(x,y) = \sum_{i=1}^m x_i - y_i $
Canberra:	Chebyshev:	
$D(x,y) = \sum_{i=1}^m \frac{ x_i - y_i }{x_i + y_i}$	$D(x,y) = \max_{i=1}^m x_i - y_i $	
Quadratic: $D(x,y) = (x-y)^T Q (x-y)$ Q is a problem-specific positive definite $m \times m$ weight matrix.	$D(x,y) = \sum_{j=1}^m \sum_{k=1}^m (x_j - y_j)(x_k - y_k) V_{jk}$ V is the covariance matrix of A_1, A_{m_0} and A_j is the vector of values for attribute j occurring in the training set instances $1..n$.	
Mahalanobis: $D(x,y) = \sqrt{(x-y)^T V^{-1} (x-y)}$		
Correlation: $D(x,y) = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}}$		
Chi-square: $D(x,y) = \sum_{i=1}^m \frac{1}{\sum_{j=1}^m \frac{x_j}{x_j}} \left(\frac{x_i}{x_i} - \frac{y_i}{y_i} \right)^2$		
Kendall's Rank Correlation: $\text{sign}(x) = 1, 0 \text{ or } -1 \text{ if } x < 0, x = 0, \text{ or } x > 0, \text{ respectively.}$	$D(x,y) = 1 - \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{sign}(x_i - x_j) \text{sign}(y_i - y_j)$	

Figure 1. Equations of selected distance functions.
(x and y are vectors of m attribute values).

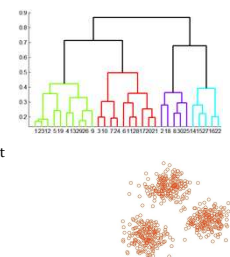
14

Hierarchical Clustering

15

Overview: Methods of Clustering

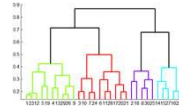
- **Hierarchical:**
 - **Agglomerative** (bottom up):
 - Initially, each point is a cluster
 - Repeatedly combine the two "nearest" clusters into one
 - **Divisive** (top down):
 - Start with one cluster and recursively split it
- **Point assignment:**
 - Maintain a set of clusters
 - Points belong to "nearest" cluster



16

Hierarchical Clustering

- **Key operation:** Repeatedly combine two nearest clusters



- **Three important questions:**

- 1) How do you represent a cluster of more than one point?
- 2) How do you determine the "nearness" of clusters?
- 3) When to stop combining clusters?

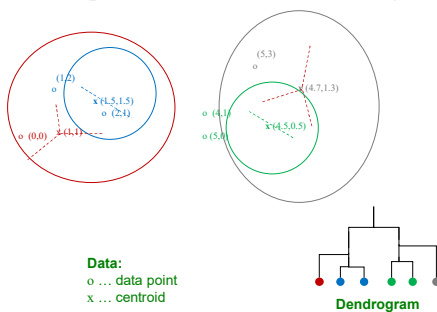
Hierarchical Clustering

- **Key operation:** Repeatedly combine two nearest clusters
- **(1) How to represent a cluster of many points?**
 - **Key problem:** As you merge clusters, how do you represent the "location" of each cluster; to tell which pair of clusters is closest?
 - **Euclidean case:** each cluster has a **centroid** = average of its (data)points
- **(2) How to determine "nearness" of clusters?**
 - Measure cluster distances by distances of centroids

17

18

Example: Hierarchical clustering



Dendograms

- ♦ What are dendograms?
 - Dendograms are used to represent the distances at which the different clusters meet.
 - They provide us an idea as to how the clustering looks like diagrammatically.
 - Tree/Hierarchical Structure

19

20

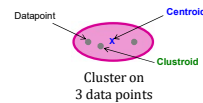
In the Non-Euclidean Case

What about the Non-Euclidean case?

- The only "locations" we can talk about are the points themselves i.e., there is no "average" of two points
- Approach 1:**
 - (1) How to represent a cluster of many points?**
clustroid = (data)point "closest" to other points
 - (2) How do you determine the "nearness" of clusters?** Treat clustroid as if it were centroid, when computing inter-cluster distances

"Closest" Point?

- (1) How to represent a cluster of many points?**
clustroid = point "closest" to other points
- Possible meanings of "closest":**
 - Smallest maximum distance to other points
 - Smallest average distance to other points
 - Smallest sum of squares of distances to other points
 - For distance metric d clustroid c of cluster C is: $\min_c \sum_{x \in C} d(x, c)^2$



Centroid is the avg. of all (data)points in the cluster. This means centroid is an "artificial" point.
Clustroid is an **existing** (data)point that is "closest" to all other points in the cluster.

21

22

Defining "Nearness" of Clusters

- (2) How do you determine the "nearness" of clusters?**
 - Approach 2:**
Intercluster distance = minimum of the distances between any two points, one from each cluster
 - Approach 3:**
Pick a notion of "**cohesion**" of clusters, e.g., maximum distance from the clustroid
 - Merge clusters whose **union** is most cohesive

Cohesion

- Approach 1:** Use the **diameter** of the merged cluster = maximum distance between points in the cluster
- Approach 2:** Use the **average distance** between points in the cluster
- Approach 3:** Use a **density-based approach**
 - Take the diameter or avg. distance, e.g., and divide by the number of points in the cluster

23

24

Implementation

- **Naïve implementation of hierarchical clustering:**
 - At each step, compute pairwise distances between all pairs of clusters, then merge $O(N^3)$
- Careful implementation using priority queue can reduce time to $O(N^2 \log N)$
 - Still too expensive for really big datasets that do not fit in memory

25

K-Means Clustering

26

K-Means Algorithm

- Assumes Euclidean space/distance
- Start by picking k , the number of clusters
- Initialize clusters by picking one point per cluster
 - **Example:** Pick one point at random, then $k-1$ other points, each as far away as possible from the previous points

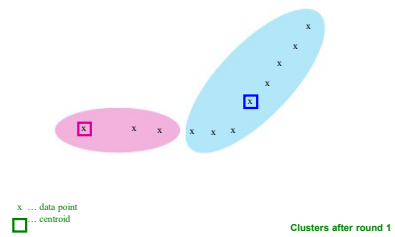
27

Populating Clusters

- **1)** For each point, place it in the cluster whose current centroid it is nearest
- **2)** After all points are assigned, update the locations of centroids of the k clusters
- **3)** Reassign all points to their closest centroid
- Sometimes moves points between clusters
- **Repeat 2 and 3 until convergence**
 - **Convergence:** Points don't move between clusters and centroids stabilize

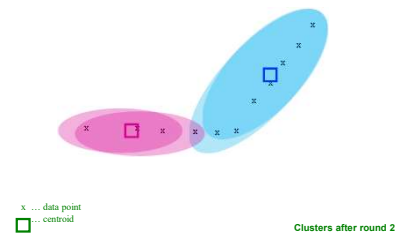
28

Example: Assigning Clusters



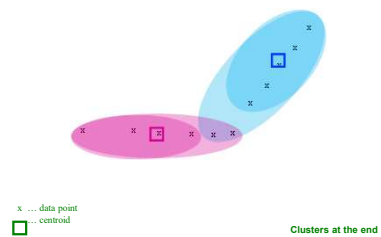
29

Example: Assigning Clusters



30

Example: Assigning Clusters

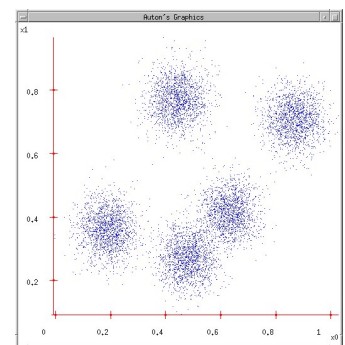


31

Some Data

Consider two dimensional data point in a 2D Cartesian system – just for the sake of visualization

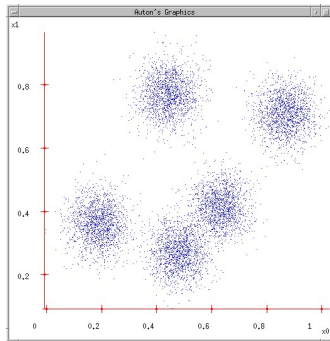
Each data point, in real life is multidimensional vector in n-dimensional space.



32

K-Means

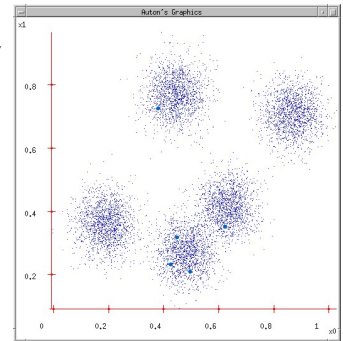
1. Ask user how many clusters they'd like.
(e.g. $k=5$)



33

K-Means

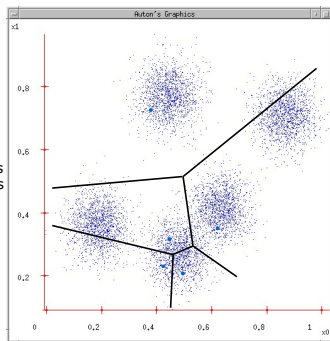
1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations



34

K-Means

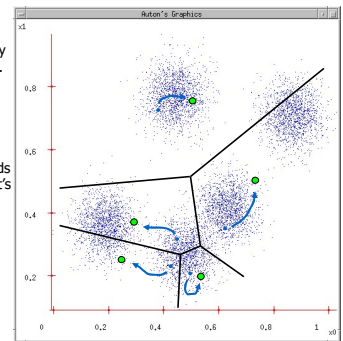
1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



35

K-Means

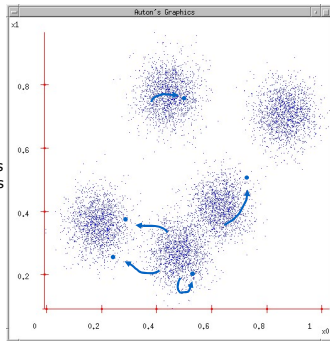
1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



36

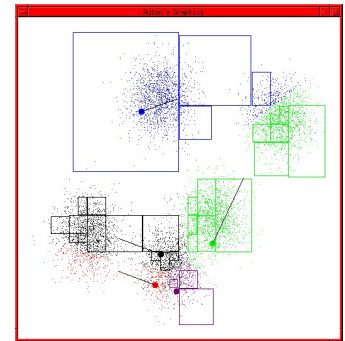
K-Means

1. Ask user how many clusters they'd like. (e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!



37

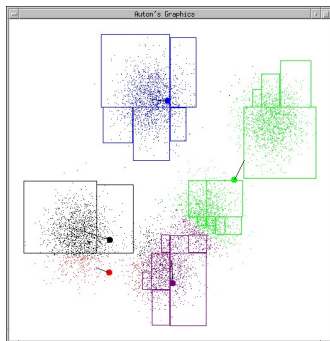
K-Means Starts



38

K-Means continues

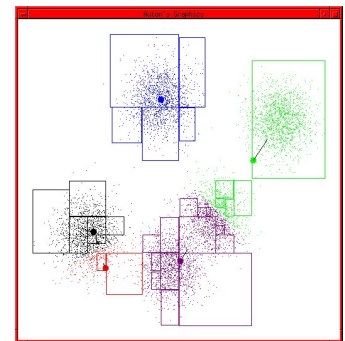
...



39

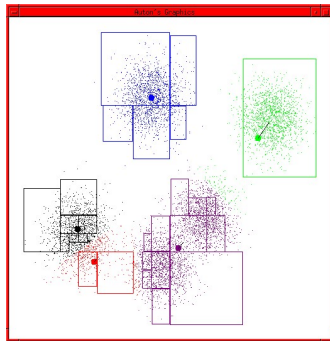
K-Means continues

...



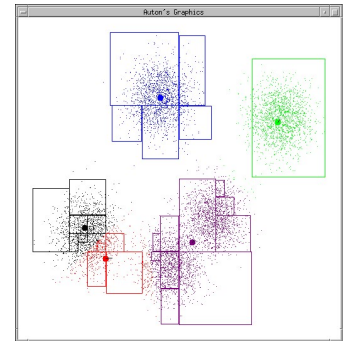
40

**K-Means
continues**
...



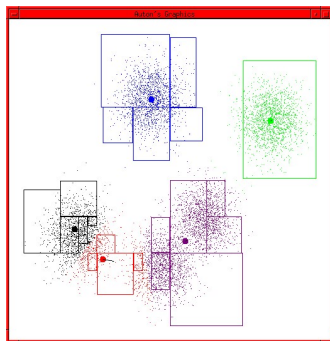
41

**K-Means
continues**
...



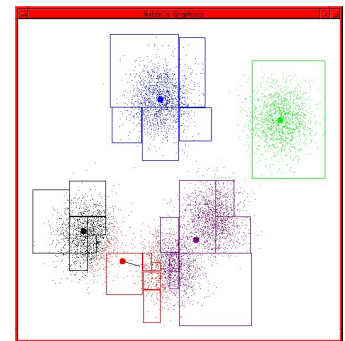
42

**K-Means
continues**
...



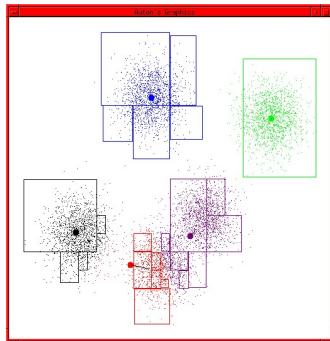
43

**K-Means
continues**
...



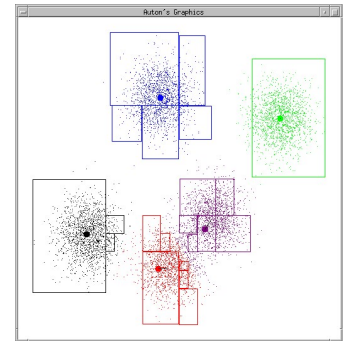
44

**K-Means
continues**
...



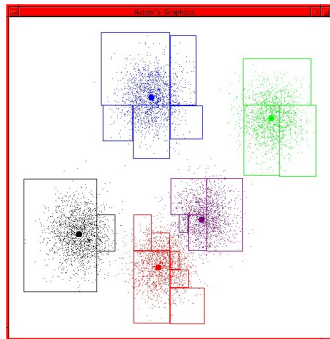
45

**K-Means
continues**
...



46

**K-Means
continues...**

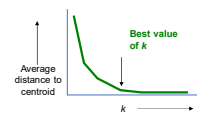


47

Getting the k right

How to select k ?

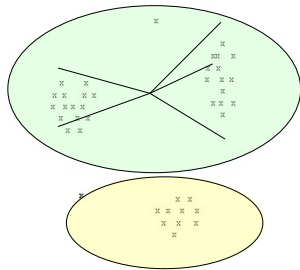
- Try different k , looking at the change in the average distance to centroid as k increases
- Average falls rapidly until right k , then changes little



48

Example: Picking k

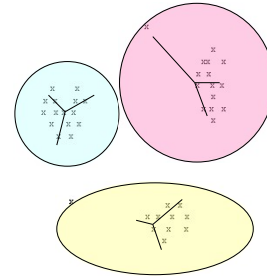
Too few;
many long
distances
to centroid.



49

Example: Picking k

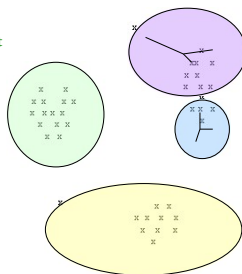
Just right;
distances
rather short.



50

Example: Picking k

Too many;
little improvement
in average
distance.



51

Important Notes on K-Mean Clustering

1. When the numbers of data points is less, initial grouping will determine the cluster significantly.
2. The number of cluster K , must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. It is sensitive to **initial condition** - different initial condition may produce different result of cluster.
4. The algorithm may be trapped in the *local optimum*. The order in which data is coming may produce different cluster if the number of data is few.
5. Runs in $O(tkn)$ time; where n is number of data points, k is number of clusters and t is number of iterations.
6. K-means clustering can be applied to machine learning or data mining, speech understanding (*Vector Quantization*), *Image Segmentation* etc..

52

K-Means

Clustering algorithm

Uses distance from data points to k-centroids to cluster data into k-groups.

Centroids are not necessarily data points.

Updates centroid on each pass by calculations over all data in a class.

Must iterate over data until center point doesn't move.

53

Cophenetic Correlation

The right choice of dendrogram is done by considering a value known as a cophenetic correlation.

- ♦ Dendrogram Distance - distance between two points/clusters as described by that dendrogram.

Cophenetic correlation computes the correlation between the euclidean distance and the dendrogram distance for a particular dendrogram of all possible pair of points.

Performance Measure -

The dendrogram corresponding to highest correlation coefficient is considered to be better representative of the clustered data and is used to produce labels/ clusters for the data set.

54

Case-studies

55

Case-study – K-means Clustering

To determine if there is a relationship between higher levels of "black and white" thinking and higher levels of self-reported depression in psychiatric patients hospitalized for depression. Apply K means clustering and assign groups for model prediction.

It is common for people who tend to think of their reality as a series of black and white events to suffer from depression.

Psybersquare, Inc. describes a few examples of this way of thinking by saying that those who suffer from this way of thinking think that, "If things aren't 'perfect,' then they must be 'horrible.' If your child isn't 'brilliant' then he must be 'stupid.' If you're not 'fascinating' then you must be 'boring.'" This can be a difficult way to live since those suffering from this way of thinking may never feel that their reality is "good enough".

The data used for this study is from the Ginzberg data frame which is based on psychiatric patients hospitalized for depression. Data is from the book Applied Regression Analysis and Generalized Linear Models, Second Edition by Fox, J. (2008).

The dataset includes three variables - simplicity (black and white thinking), fatalism, and depression.

The data also includes these variables each adjusted by regression for other variables thought to influence depression. For the purposes of this study, we will use the non-adjusted values.

Ginzberg Dataset on Depression display_output(Ginzberg, out_type) simplicity fatalism depression adjsimp adjfatal adjdep Here, "black and white thinking" is referred as "Simplicity"

56

Case-study - Hierarchical Clustering

The data set has information about features of silhouette extracted from the images of different cars –
Four "Corgie" model vehicles were used for the experiment
A double decker bus, Chevrolet van, Saab 9000 and an Opel Manta 400 cars.

This particular combination of vehicles was chosen with the expectation that the bus, van and either one of the cars would be readily distinguishable, but it would be more difficult to distinguish between the cars.



57

58

To be continued in the next session.....

59