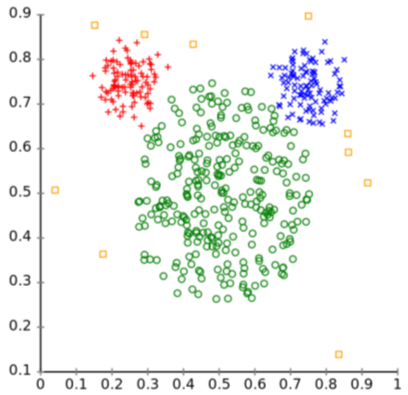
Lecture 4: Clustering

Clustering

- Unsupervised learning
- Generating “classes”
- Distance/similarity measures
- Agglomerative methods
- Divisive methods

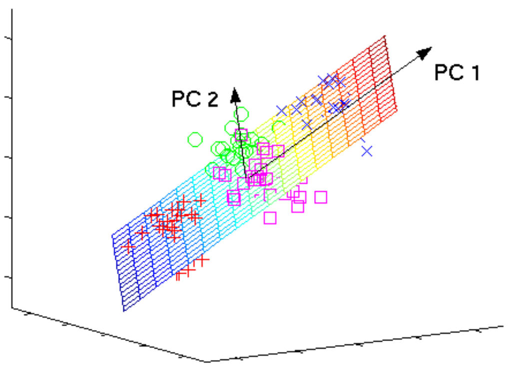
Unsupervised Learning

- No labels/responses. Finding structure in data.
- Dimensionality Reduction.



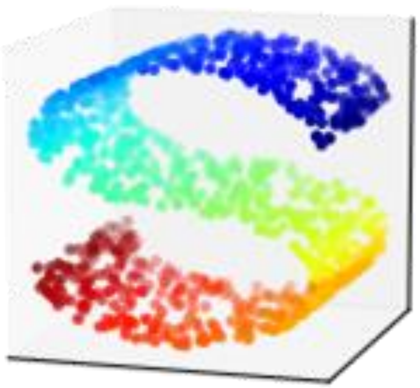
Clustering

$T: \mathbb{R}^d \rightarrow \{1, 2, \dots, k\}$



Subspace Learning

$T: \mathbb{R}^d \rightarrow \mathbb{R}^m$

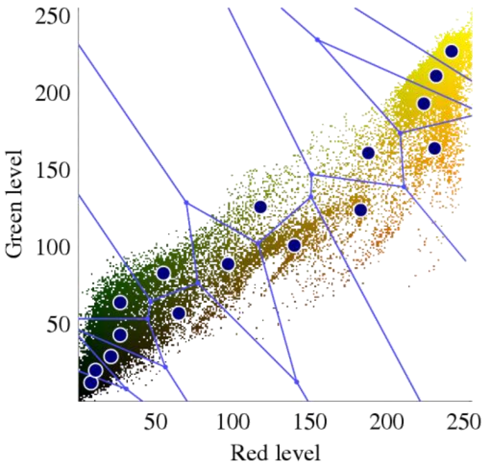


Manifold Learning

Uses of Unsupervised Learning

- Data compression

RGB
(12,165,73)
(11,167,79)
:



Labels
3
43
:

Dictionary
1 ~ (10, 160, 70)
2 ~ (40, 240, 20)
:

Uses of Unsupervised Learning

- Improve classification/regression (semi-supervised learning)

1. From *unlabeled data,* learn a good features $T: \mathbb{R}^d \to \mathbb{R}^m$.
2. To *labeled data*, apply transformation $T: \mathbb{R}^d \to \mathbb{R}^m$.

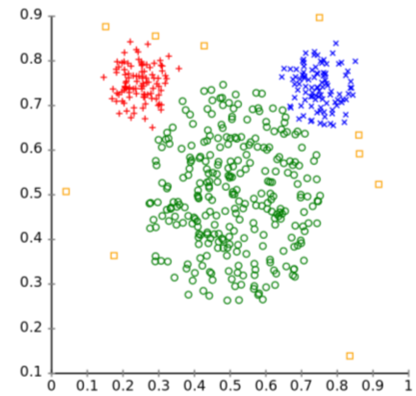
$$\left(T\left(x^{(1)}\right), y^{(1)}\right), \ldots, \left(T\left(x^{(n)}\right), y^{(n)}\right)$$

3. Perform classification/regression on transformed data.

What is Clustering?

- Form of *unsupervised* learning - no information from teacher
- The process of partitioning a set of data into a set of meaningful (hopefully) sub-classes, called *clusters*
- Cluster:
  - collection of data points that are "similar" to one another and collectively should be treated as group
  - as a collection, are sufficiently different from other groups

What is Clustering



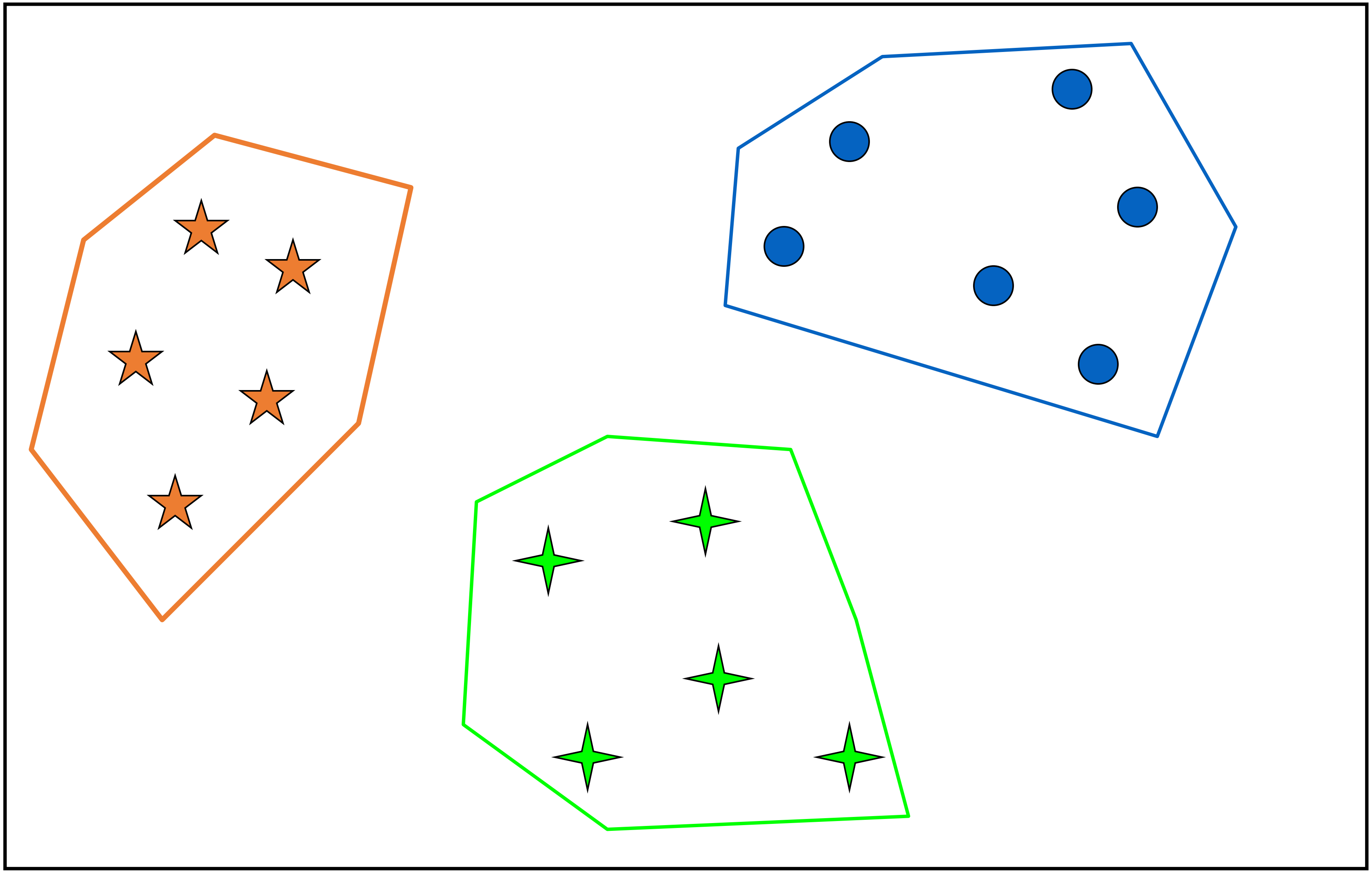
Clustering Problem.

Input. Training data $\mathcal{S}_n = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$, each $x^{(i)} \in \mathbb{R}^d$.
Integer k

Output. Clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k \subset \{1, 2, \ldots, n\}$ such that every data point is in one and only one cluster.

Some clusters could be empty!

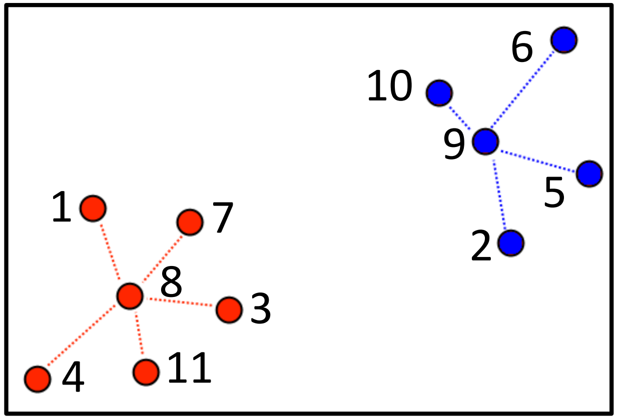
Clusters



How to Specify a Cluster

- By listing all its elements

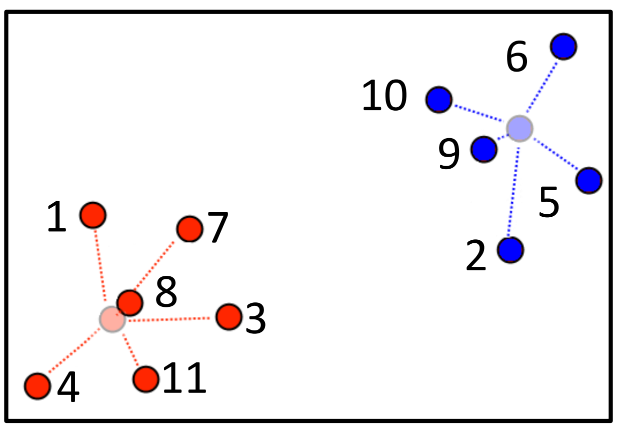
$$\mathcal{C}_1 = \{1,3,4,7,8,11\}$$
$$\mathcal{C}_2 = \{2,5,6,9,10\}$$



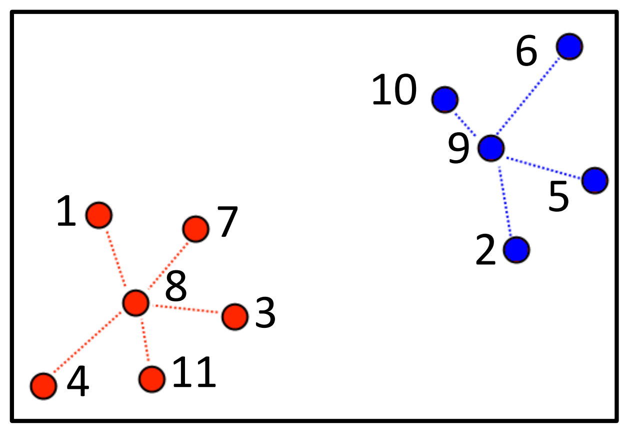
How to Specify a Cluster

- Using a representative
  a. A point in center of cluster (centroid)
  b. A point in the training data (exemplar)

Each point $x^{(i)}$ will be assigned the closest representative.



centroid



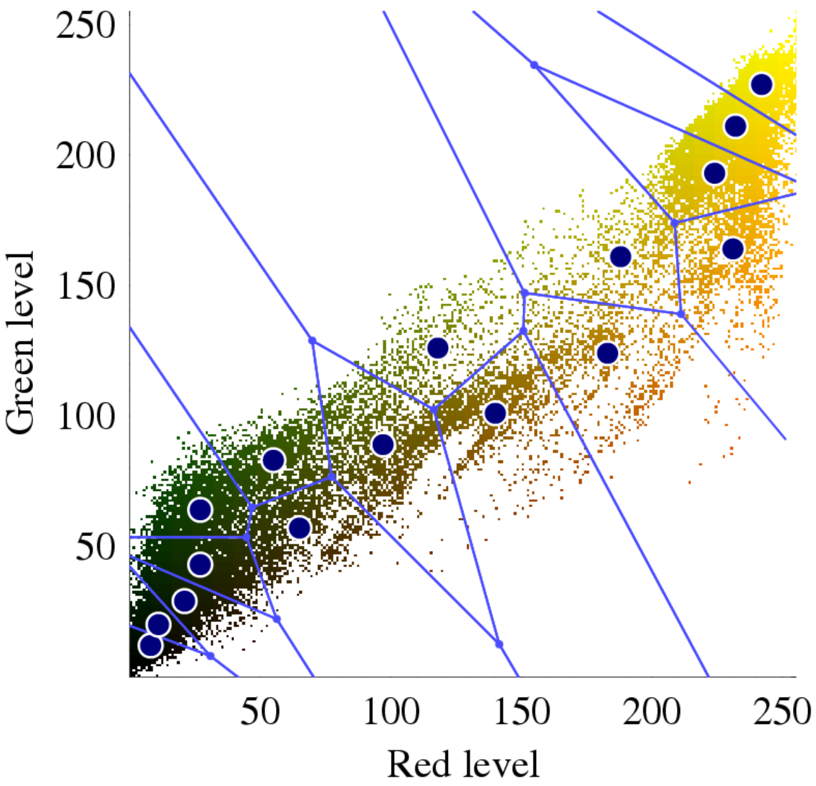
exemplar

Characterizing Cluster Methods

- Class - label applied by clustering algorithm
  - hard versus fuzzy:
    - hard - either is or is not a member of cluster
    - fuzzy - member of cluster with probability
- Distance (similarity) measure - value indicating how similar data points are
- Deterministic versus stochastic
  - deterministic - same clusters produced every time
  - stochastic - different clusters may result
- Hierarchical - points connected into clusters using a hierarchical structure

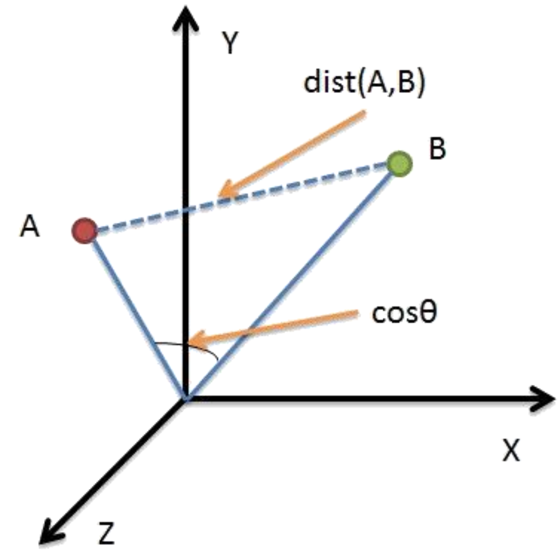
Voronoi Diagram

We can partition all the points in the space into regions, according to their closest representative.



Training Loss

Distance/Similarity Measures

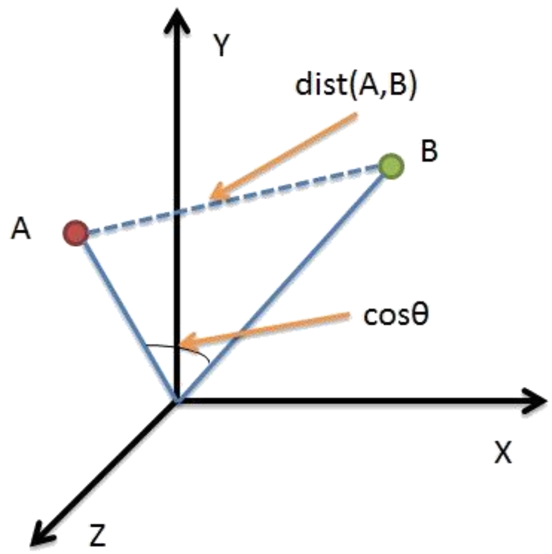


(sometimes called *loss functions)*

A measure of how close two data points are. Nearby points (i.e. distance is *small)* are more likely they belong to the same cluster.

- Euclidean Distance $\quad \mathrm{dist}(x, y) = \|x - y\|^2$

Distance/Similarity Measures



(sometimes called *kernels, correlation)*

A measure of how alike two data points are. Similar points (i.e. similarity is *large)* are more likely they belong to the same cluster.

- Cosine Similarity $\cos(x, y) = \dfrac{x^{\top} y}{\|x\|\,\|y\|}$

Distance/Similarity Measures

- Key to grouping points

  distance = inverse of similarity

- Often based on representation of objects as feature vectors

An Employee DB

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

Term Frequencies for Documents

	T1	T2	T3	T4	T5	T6
Doc1	0	4	0	0	0	2
Doc2	3	1	4	3	1	2
Doc3	3	0	0	0	3	0
Doc4	0	1	0	3	0	0
Doc5	2	2	2	3	1	4

Which objects are more similar?

Distance/Similarity Measures

Properties of measures:

based on feature values $x_{instance\#,\ feature\#}$

for all objects x_i, x_j, dist(x_i, x_j) $\geq$ 0, dist(x_i, x_j)=dist(x_j, x_i)

for any object x_i, dist(x_i, x_i) = 0

dist(x_i, x_j) $\leq$ dist(x_i, x_k) + dist(x_k, x_j)

Manhattan distance: $\sum_{f=1}^{|features|} | x_{i,f} - x_{j,f} |$

Euclidean distance: $\sqrt{\sum_{f=1}^{|features|} (x_{i,f} - x_{j,f})^2}$

Distance/Similarity Measures

Minkowski distance (p): $\sqrt[p]{\sum_{f=1}^{|features|}(x_{i,f}-x_{j,f})^p}$

Mahalanobis distance: $(x_i - x_j)\nabla^{-1}(x_i - x_j)^T$

where ∇^{-1} is covariance matrix of the data

More complex measures:

Mutual Neighbor Distance (MND) - based on a count of number of neighbors

Distance (Similarity) Matrix

- Similarity (Distance) Matrix
  - based on the distance or similarity measure we can construct a symmetric matrix of distance (or similarity values)
  - (*i*, *j*) entry in the matrix is the distance (similarity) between items *i* and *j*

$$
\begin{array}{c|cccc|}
 & I_1 & I_2 & \cdots & I_n \\
\hline
I_1 & \bullet & d_{12} & \cdots & d_{1n} \\
I_2 & d_{21} & \bullet & \cdots & d_{2n} \\
\vdots & \vdots & \vdots & \bullet & \vdots \\
I_n & d_{n1} & d_{n2} & \cdots & \bullet \\
\hline
\end{array}
$$

$d_{ij} = \text{similarity (or distance) of } D_i \text{ to } D_j$

Note that $d_{ij} = d_{ji}$ (i.e., the matrix is symmetric). So, we only need the lower triangle part of the matrix.

The diagonal is all 1's (similarity) or all 0's (distance)

Example: Term Similarities in Documents

	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	0	4	0	0	0	2	1	3
Doc2	3	1	4	3	1	2	0	1
Doc3	3	0	0	0	3	0	3	0
Doc4	0	1	0	3	0	0	2	0
Doc5	2	2	2	3	1	4	0	2

$$\boxed{sim(T_i, T_j) = \sum_{k=1}^{N} (w_{ik} \cdot w_{jk})}$$

Term-Term Similarity Matrix

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

Similarity (Distance) Thresholds

- A similarity (distance) threshold may be used to mark pairs that are "sufficiently" similar

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

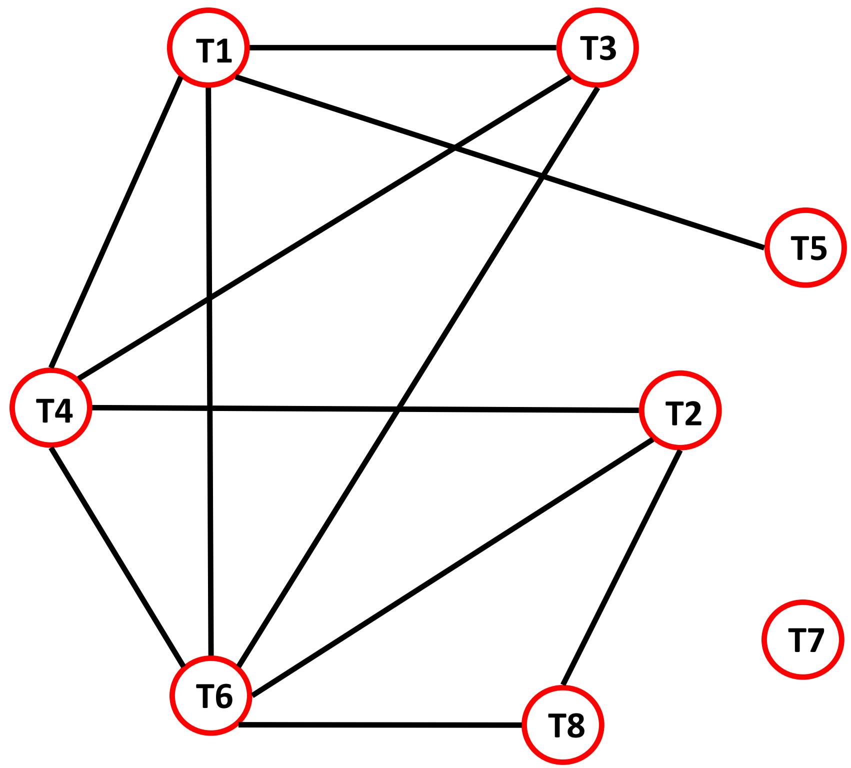
Using a threshold value of 10 in the previous example

	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0

Graph Representation

- The similarity matrix can be visualized as an undirected graph
  - each item is represented by a node, and edges represent the fact that two items are similar (a 1 in the similarity threshold matrix)

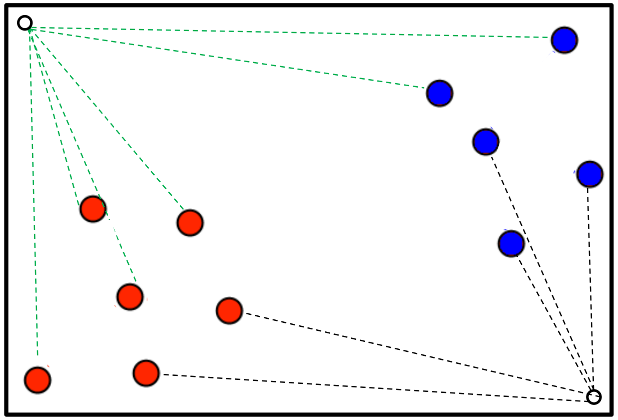
	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0



If no threshold is used, then matrix can be represented as a weighted graph

Training Loss

Sum of squared distances to closest representative.

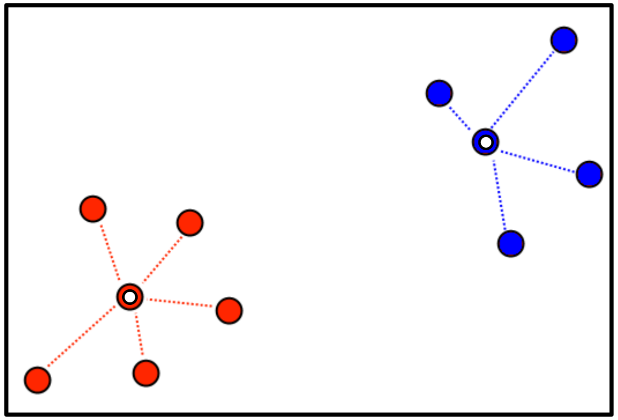


$$\text{loss} \approx 11 \times (1)^2 = 11$$

assume length of each
edge is about 1

Training Loss

Sum of squared distances to closest representative (cluster center).



$$\text{loss} \approx 9 \times (0.1)^2 = 0.09$$

assume length of each
edge is about 0.1

Training Loss

Optimizing over representatives (cluster centers).

How do I use a similarity function instead?

$$\mathcal{L}_{n,k}\left(z^{(1)}, \ldots, z^{(k)}; \mathcal{S}_n\right) = \sum_{i=1}^{n} \min_{1 \leq j \leq k} \left\| x^{(i)} - z^{(j)} \right\|^2 .$$

Training Loss

Optimizing over clusters.

$$\mathcal{L}_{n,k}(\mathcal{C}_1, \ldots, \mathcal{C}_n; \mathcal{S}_n) = \sum_{j=1}^{n} \sum_{i \in \mathcal{C}_j} \left\| x^{(i)} - \frac{1}{|\mathcal{C}_j|} \sum_{i' \in \mathcal{C}_j} x^{(i')} \right\|^2 .$$

Training loss

Instead of the distance metric, you can use the *negative* similarity function.

Optimizing both clusters and representatives.

$$\mathcal{L}_{n,k}\left(\mathcal{C}_1, \ldots, \mathcal{C}_k, z^{(1)}, \ldots, z^{(k)}; \mathcal{S}_n\right) = \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \left\| x^{(i)} - z^{(j)} \right\|^2$$

Basic Clustering Methodology

Two approaches:

- Agglomerative: pairs of items/clusters are successively linked to produce larger clusters
- Divisive (partitioning): items are initially placed in one cluster and successively divided into separate groups

Cluster Validity

- One difficult question: how *good* are the clusters produced by a particular algorithm?
- Difficult to develop an objective measure
- Some approaches:
  - external assessment: compare clustering to *a priori* clustering
  - internal assessment: determine if clustering intrinsically appropriate for data
  - relative assessment: compare one clustering methods results to another methods

Basic Questions

- Data preparation - getting/setting up data for clustering
  - extraction
  - normalization
- Similarity/Distance measure - how is the distance between points defined
- Use of domain knowledge (prior knowledge)
  - can influence preparation, Similarity/Distance measure
- Efficiency - how to construct clusters in a reasonable amount of time

Agglomerative Single-Link

- Single-link: connect all points together that are within a threshold distance
- Algorithm:
  1. place all points in a cluster
  2. pick a point to start a cluster
  3. for each point in current cluster

     add all points within threshold not already in cluster

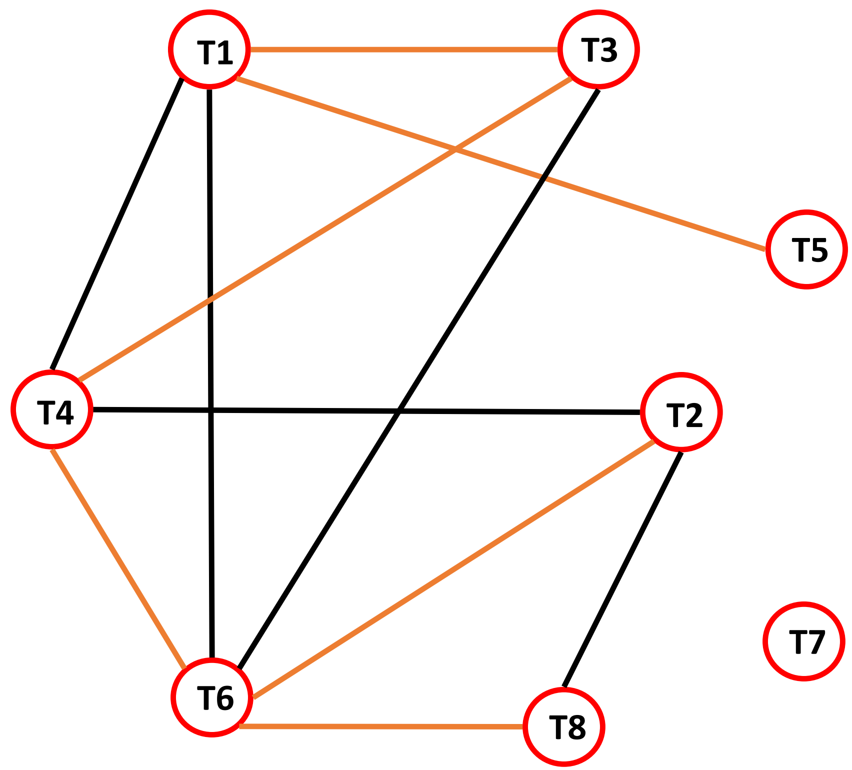
     repeat until no more items added to cluster
  4. remove points in current cluster from graph
  5. Repeat step 2 until no more points in graph

Example

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0

All points except T7 end up in one cluster



Agglomerative Complete-Link (Clique)

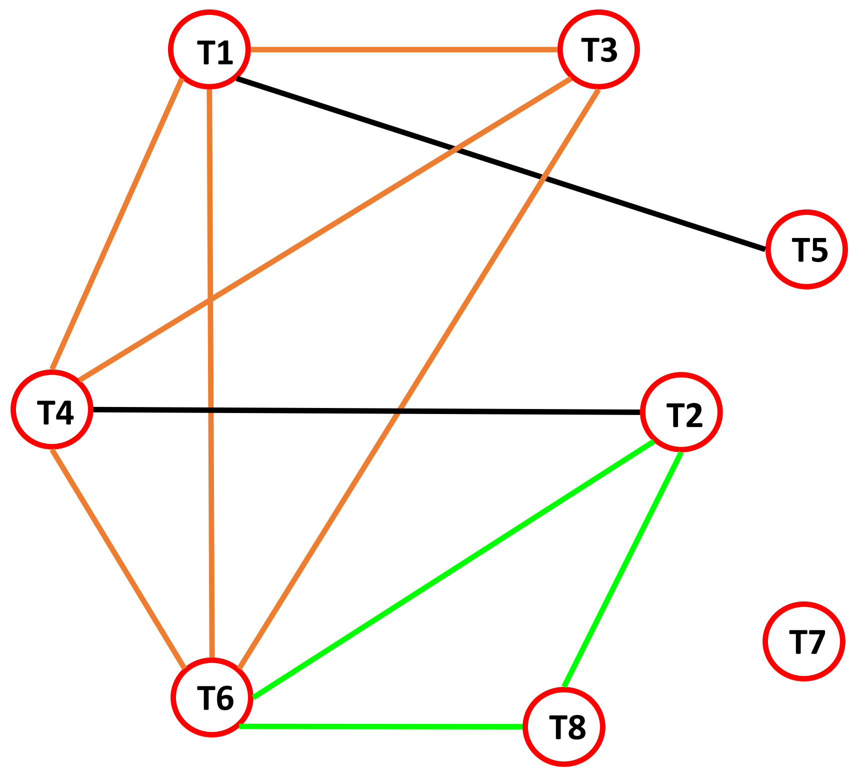
- Complete-link (clique): all of the points in a cluster must be within the threshold distance
- In the threshold distance matrix, a clique is a complete graph
- Algorithms based on finding maximal cliques (once a point is chosen, pick the largest clique it is part of)
  - not an easy problem

Example

Different clusters possible based on where cliques start

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3

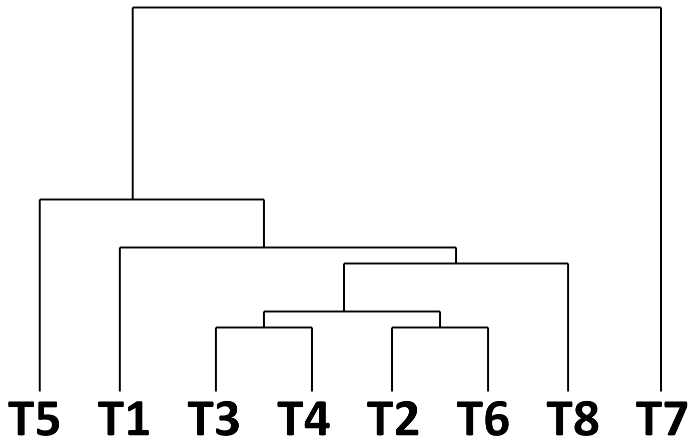
	T1	T2	T3	T4	T5	T6	T7
T2	0						
T3	1	0					
T4	1	1	1				
T5	1	0	0	0			
T6	1	1	1	1	0		
T7	0	0	0	0	0	0	
T8	0	1	0	0	0	1	0



Hierarchical Methods

- Based on some methods of representing hierarchy of data points
- One idea: hierarchical dendogram (connects points based on similarity)

	T1	T2	T3	T4	T5	T6	T7
T2	7						
T3	16	8					
T4	15	12	18				
T5	14	3	6	6			
T6	14	18	16	18	6		
T7	9	6	0	6	9	2	
T8	7	17	8	9	3	16	3



Hierarchical Agglomerative

- Compute distance matrix
- Put each data point in its own cluster
- Find most similar pair of clusters
  - merge pairs of clusters (show merger in dendogram)
  - update proximity matrix
  - repeat until all patterns in one cluster

K-Means

Optimization Algorithm

Goal. Minimize $\mathcal{L}(x, y)$.

Coordinate Descent (Optimization).

Repeat until convergence:
1. Find optimal x while holding y constant.
2. Find optimal y while holding x constant.

Optimization Algorithm

Coordinate Descent (Optimization)

Repeat until convergence:

- Find best clusters given centroids
- Find best centroid given clusters

$$\mathcal{L}_{n,k}\left(\mathcal{C}_1, \dots, \mathcal{C}_k, z^{(1)}, \dots, z^{(k)}; \mathcal{S}_n\right) = \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \left\| x^{(i)} - z^{(j)} \right\|^2$$

Optimization Algorithm

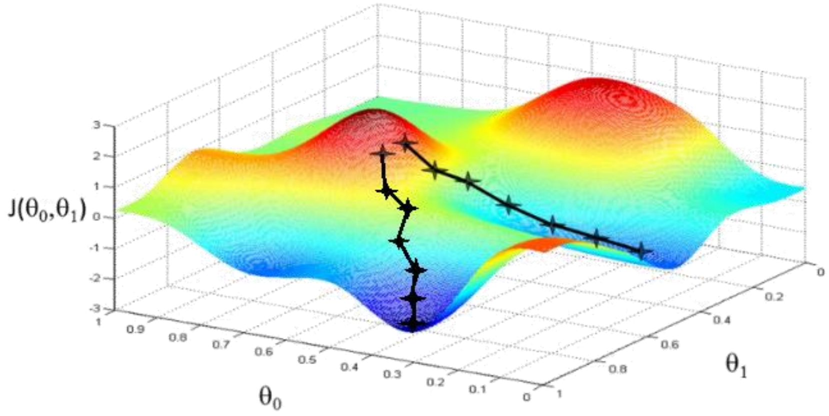
1. Initialize centroids $z^{(1)}, \dots, z^{(k)}$ from the data.
2. Repeat until no further change in training loss:

   a. For each $j \in \{1, \dots, k\}$,
   $$\mathcal{C}_j = \{\, i \text{ such that } x^{(i)} \text{ is closest to } z^{(j)} \,\}.$$

   b. For each $j \in \{1, \dots, k\}$,
   $$z^{(j)} = \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} x^{(i)} \quad \text{(cluster mean)}$$

Convergence

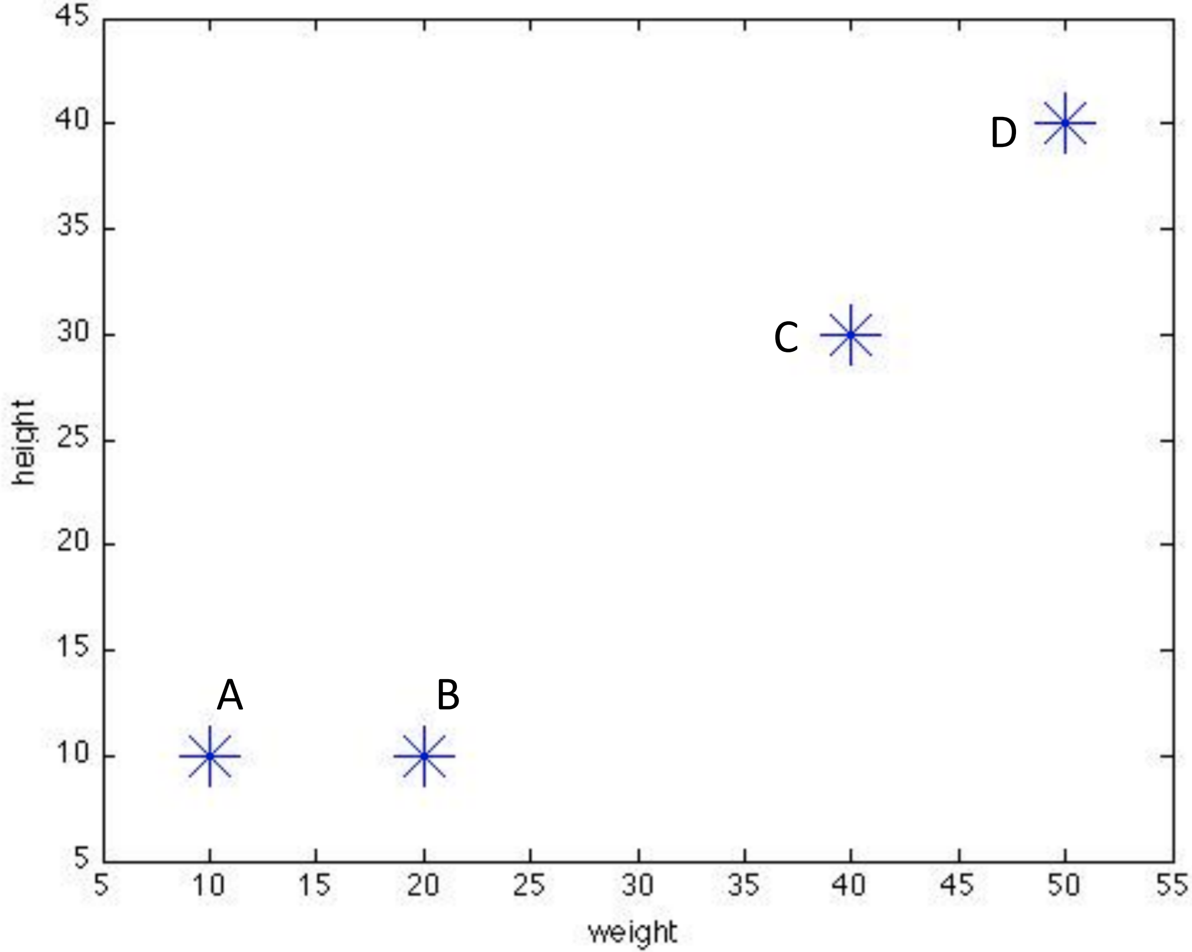
- Training loss always decreases in each step (coordinate descent).
- Converges to local minimum, not necessarily global minimum.



Repeat algorithm over many initial points, and pick the configuration with the smallest training loss.

An example – kmeans clustering

- Suppose we have 4 boxes of different sizes and we want to divide them into 2 classes
- Each box represents one point with two attributes (X,Y):



- Initial centers: suppose we choose points A and B as the initial centers, so c1 = (10, 10) and c2 = (20, 10)
- Object - centre distance: calculate the Euclidean distance between cluster centres and the objects. For example, the distance of object C from the first center is:

$$\sqrt{(40-10)^2+(30-10)^2}=36.06$$

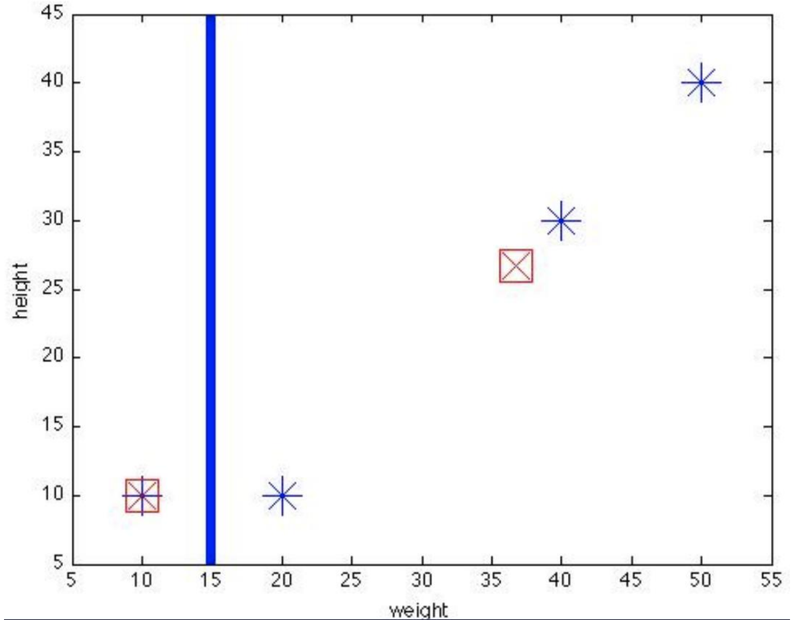
- We obtain the following distance matrix:

Centre 1	0	10	36.06	50
Centre 2	10	0	28.28	43.43

- Object clustering: We assign each object to one of the clusters based on the minimum distance from the centre:

Centre 1	1	0	0	0
Centre 2	0	1	1	1

- Determine centres: Based on the group membership, we compute the new centers
- $c_1 = (10, 10), c_2 = \left(\frac{20+40+50}{3}, \frac{10+30+40}{3}\right) = (36.7, 26.7)$



- Recompute the object-centre distances: We compute the distances of each data point from the new centres:

Centre 1	0	10	36.06	50
Centre 2	31.4	23.6	4.7	18.9

- Object clustering: We reassign the objects to the clusters based on the minimum distance from the centre:

Centre 1	1	1	0	0
Centre 2	0	0	1	1

- Determine the new centres:

$$c_1 = \left(\frac{10+20}{2}, \frac{10+10}{2}\right) = (15, 10)$$
$$c_2 = \left(\frac{40+50}{2}, \frac{30+40}{2}\right) = (45, 35)$$

- Recompute the object-centres distances:

Centre 1	5	5	32	46.1
Centre 2	43	35.4	7.1	7.1

- Object clustering:

Centre 1	1	1	0	0
Centre 2	0	0	1	1

- The cluster membership did not change from one iteration to another and so the k-means computation terminates.

Discussion

Initialization

- Empty clusters
    - Pick data points to initialize clusters
- Bad local minima
    - Initialize many times and pick solution with smallest training loss
    - Pick good starting positions

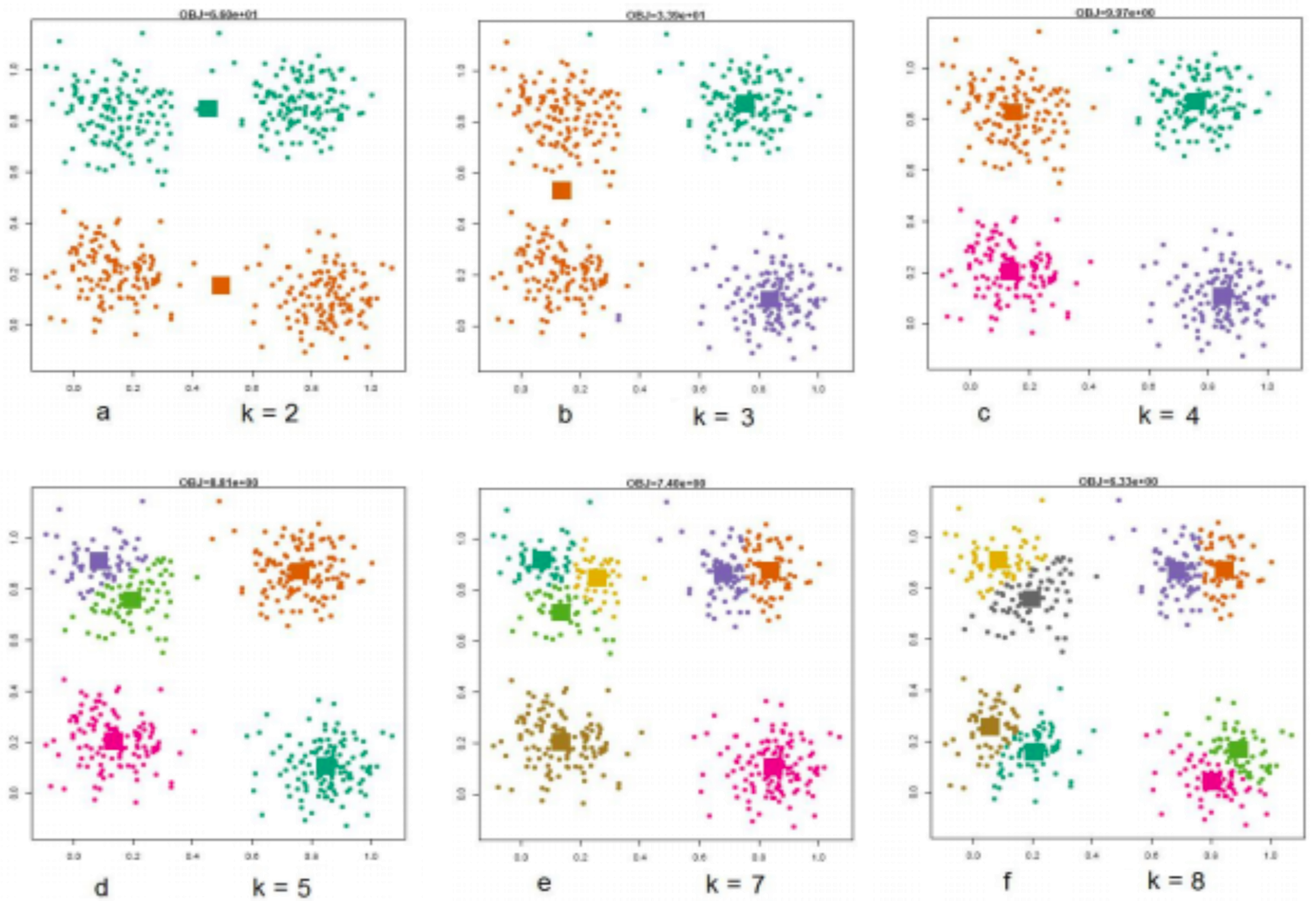
Initialization

Starting position of centroids

Final position of centroids

Problem. How to choose good starting positions?
Solution. Place them far apart with high probability.

Number of Clusters

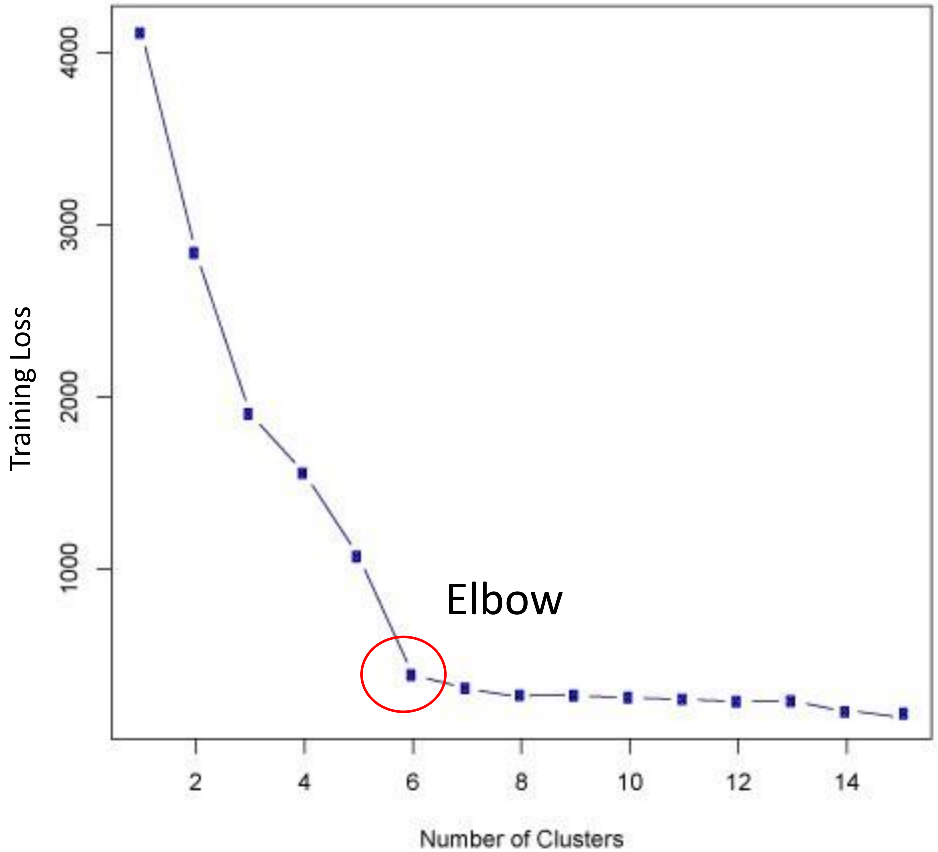


Number of Clusters

How do we choose k, the optimal number of clusters?

- Elbow method
  - Training Loss
  - Validation Loss
- Semi-supervised learning
  - Accuracy in supervised task

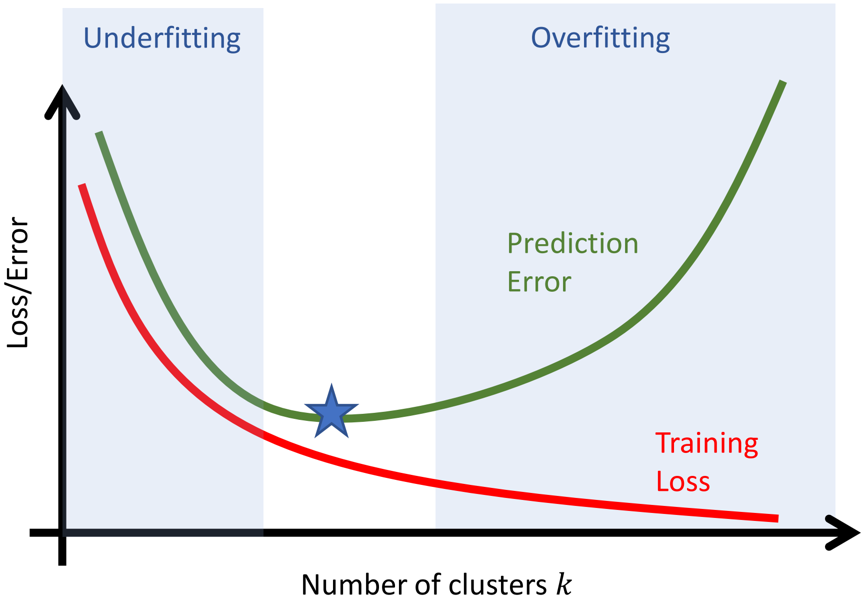
Elbow Method



Semi-Supervised Learning

Supervised task with small *labeled* data $\mathcal{S}'$

- For each number of clusters k,
  - Perform k–means on *unlabeled* data.
  - Transform $\mathcal{S}'$ using learned clusters e.g. compute distance to each centroid.
  - compute prediction error.
- Pick k with smallest prediction error.



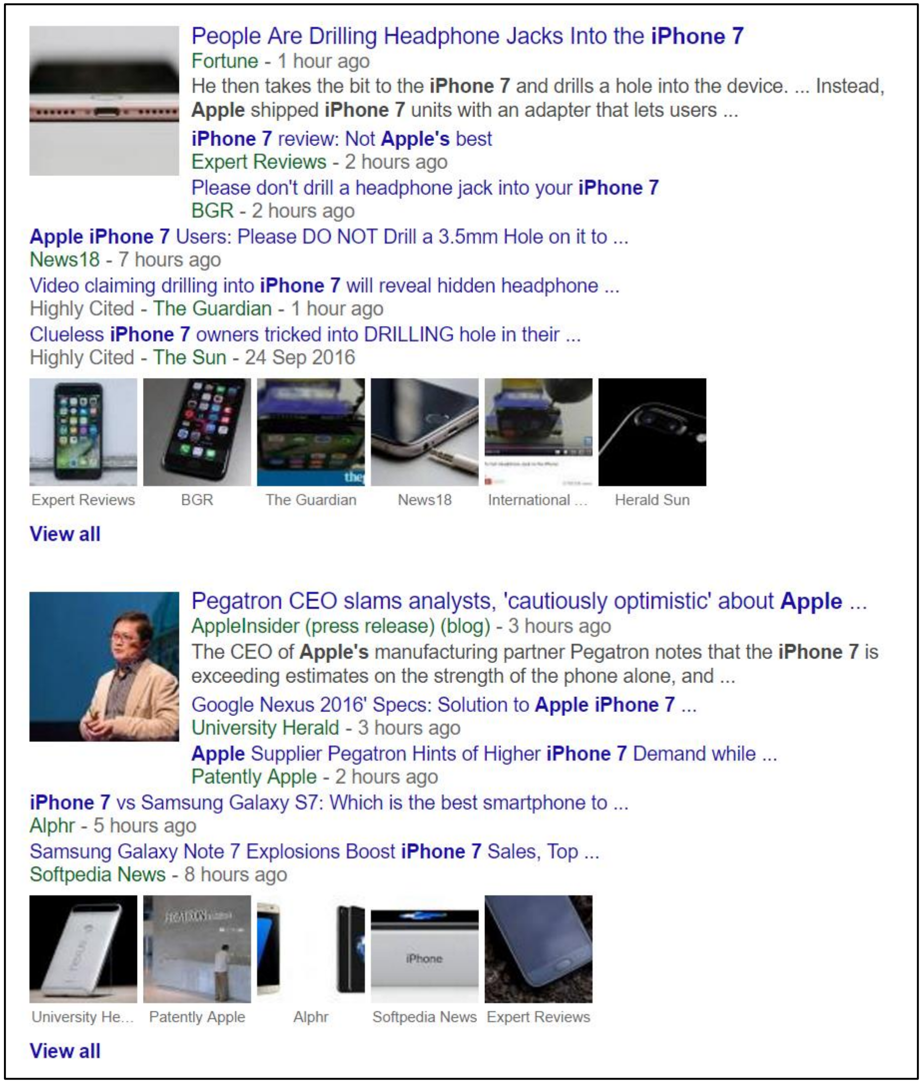
K-MeDroids

Use exemplars
instead of centroids.

e.g. Google News.

Repeat until convergence

- Find best clusters given exemplars
- Find best exemplars given clusters



Summary

- Clustering
  - Distance Metric
  - Similarity Function
  - Training Loss
- Representatives
  - Centroids
  - Exemplars
  - Voronoi Diagrams
- k-Means Algorithm

- Optimization
  - Coordinate Descent
  - Initialization
  - Software
- Generalization
  - Number of Clusters
- Applications
  - Dimensionality Reduction
  - Data Compression
  - Semi-Supervised Learning

Intended Learning Outcomes

Clustering

- Describe the differences between distance metrics and similarity functions. List examples of each of them.
- Write down the training loss using the Euclidean distance.
- Describe two ways of picking representatives for clusters. Explain how Voronoi diagrams are derived from the representatives.
- List two important applications of clustering, and how they are related to dimensionality reduction.

Intended Learning Outcomes

K-Means Algorithm

- Describe the k-means algorithm, and point out how it is based on coordinate descent.
- Explain why it is important to run the k-means algorithm several times at various starting points.
- Describe a procedure for estimating k, the number of clusters.