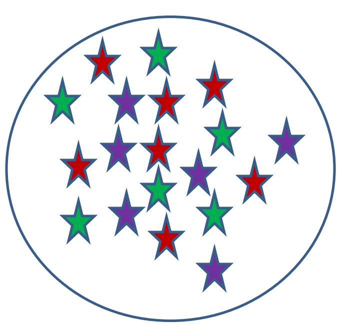
Cluster Analytics

Definition : Given a collection of data objects group them so that

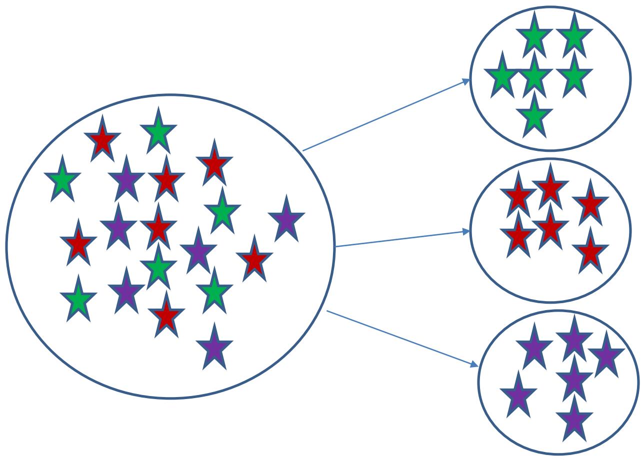
- Similar to the objects within the same cluster(group)
- Dissimilar to the objects in other clusters(groups)

Data objects can be set of web pages, set of emails or set of states in India



Definition : Given a collection of data objects group them so that

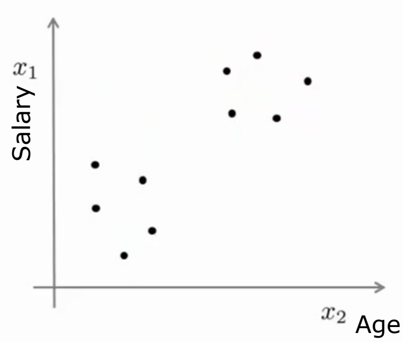
- Similar to the objects within the same cluster(group)
- Dissimilar to the objects in other clusters(groups)

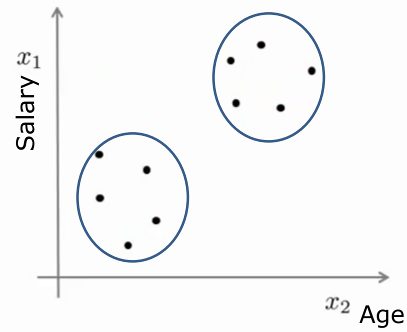


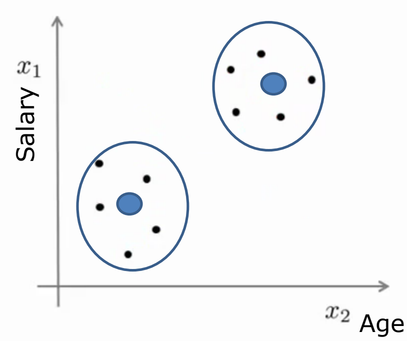
An application in bank

Customer Data

Age	Salary
68	53
93	56
79	66
89	66
73	80
75	75
27	80
59	67
90	48
72	73
45	73
50	56
58	57
62	86
66	91




x_1
Salary
x_2 Age


x_1
Salary
x_2 Age

Applications of clustering for understanding

- Web
  Cluster webpages based on their content
- Market segmentation
  Cluster groups of customers based on their spending pattern
- Bioinformatics
  Cluster similar proteins together (similarity based on chemical structure and/or functionality etc..)
- Document classification

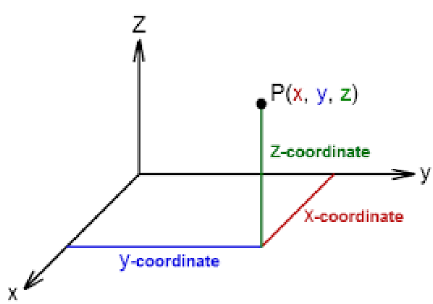
Applications of clustering for utility

- Data compression for image, sound and video data.
- Finding the nearest neighbor efficiently

K-Mean algorithm with data in Euclidean Space

Age	Salary
68	53
93	56
79	66
89	66
73	80
75	75
27	80
59	67
90	48
72	73
45	73
50	56
58	57
62	86
66	91

Euclidean Distance Between Two Records



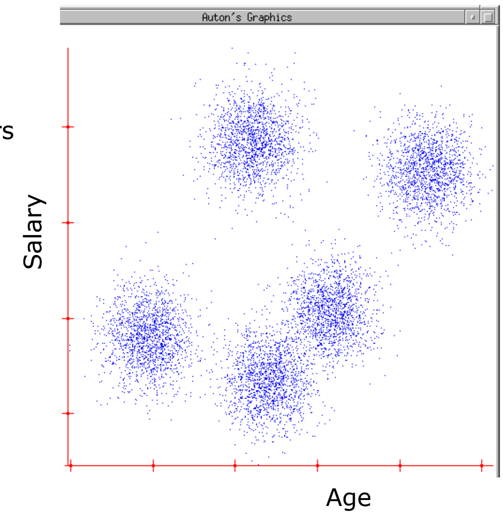
$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

Example: Distance between (2,4,2,2) and (4,2,1,3) is $\sqrt{4+4+1+1}$

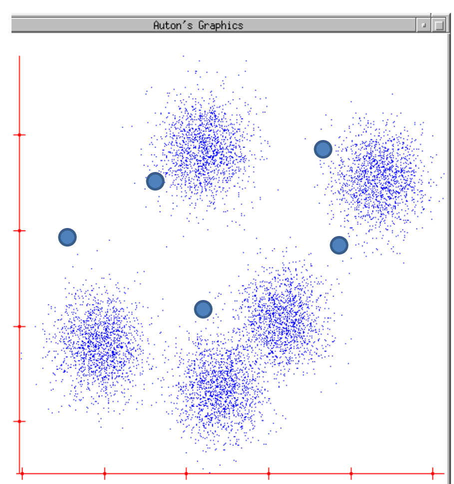
K-means for Clustering

We decided to have five clusters



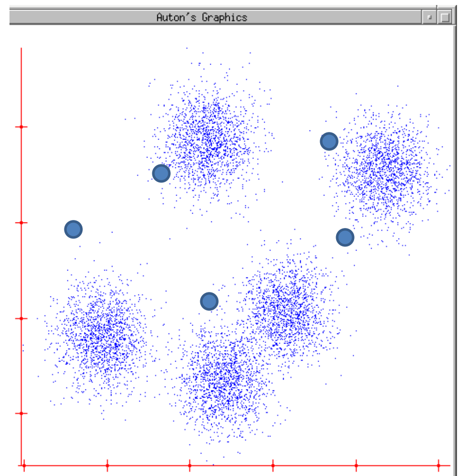
K-means for Clustering

- Start with a random guess of 5 cluster centers
- Assign each point to the nearest center
- Adjust the cluster centers



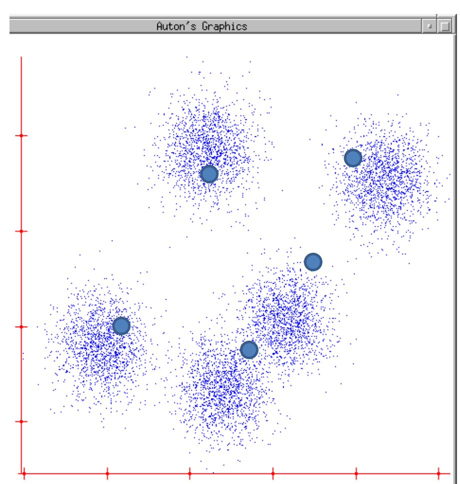
K-means for Clustering

– Start with a random guess of 5 cluster centers
– Assign each point to the nearest center
– Adjust the cluster centers



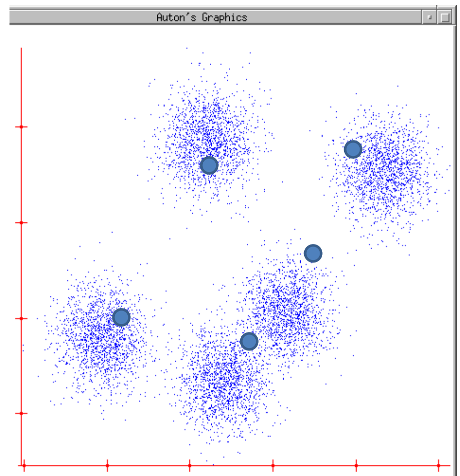
K-means for Clustering

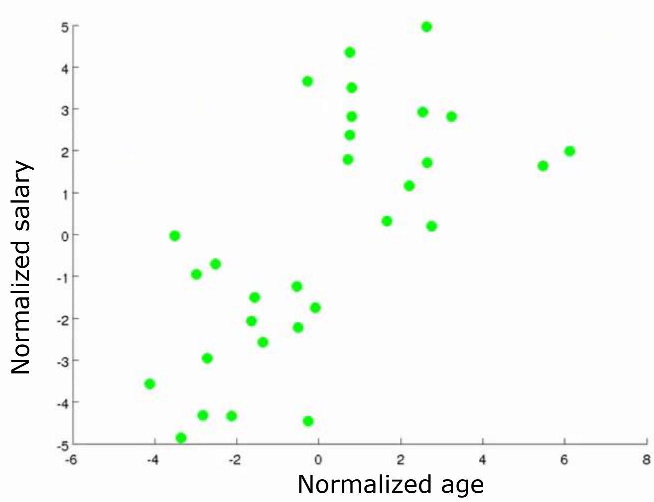
– Start with a random guess of 5 cluster centers
– Assign each point to the nearest center
– Adjust the cluster centers



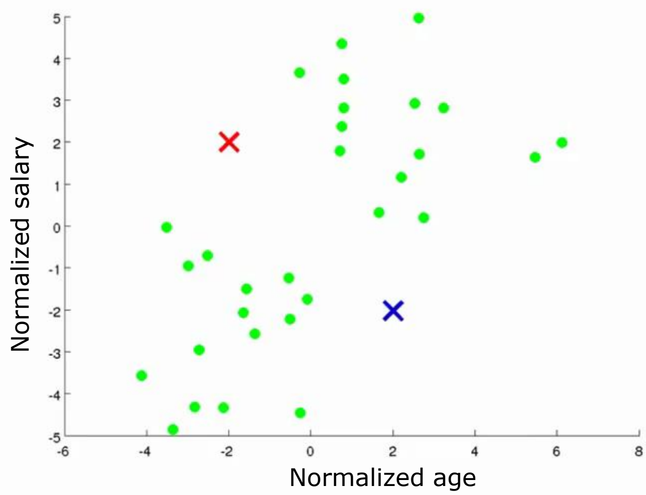
K-means for Clustering

- Start with a random guess of 5 cluster centers
- Assign each point to the nearest center
- Adjust the cluster centers

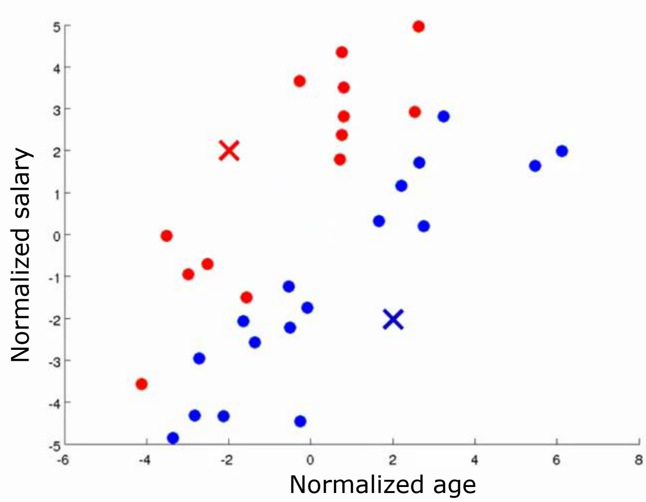




Pre Processing: Select centers by tossing a coin

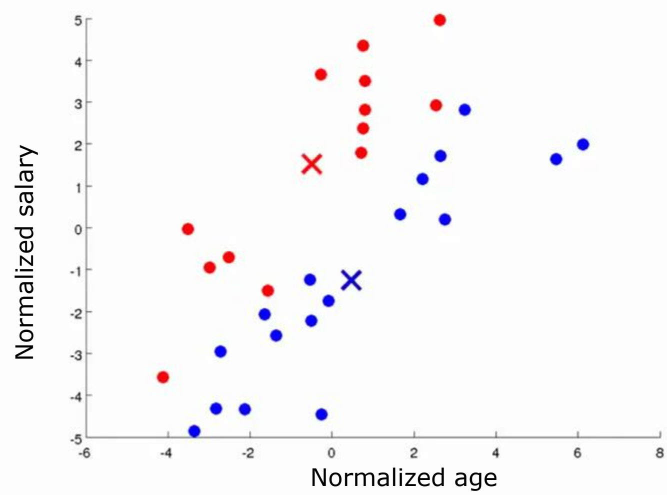


Step 1: Assigning points to nearest center



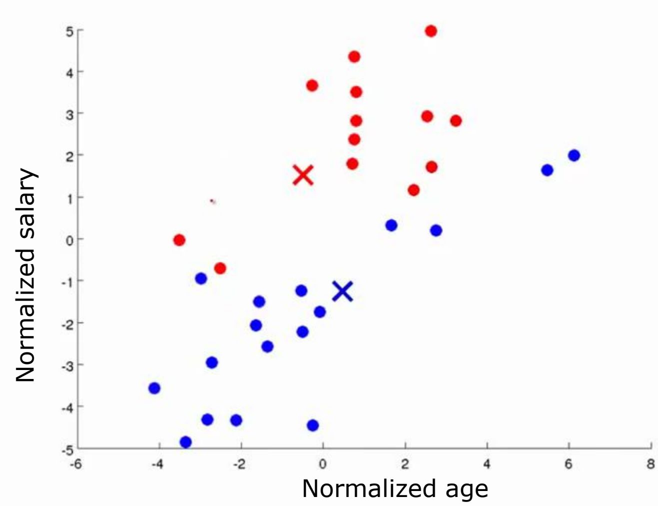
Here we use Euclidean distance measure

Step 2: Changing the clusters centers

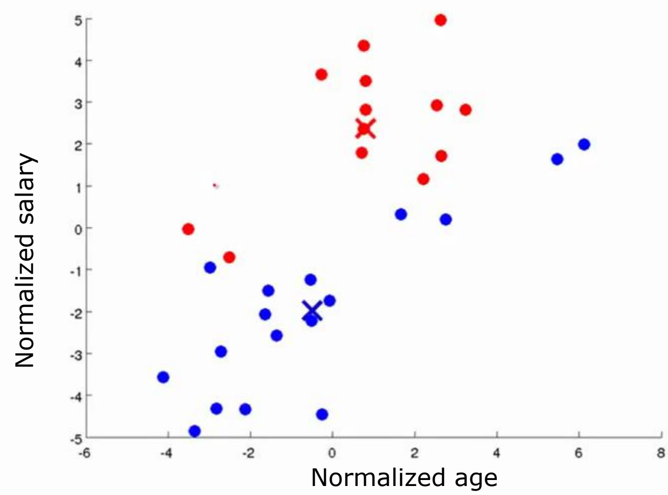


Cost function : The sum-of-squared distances from each data point to its cluster center is minimum

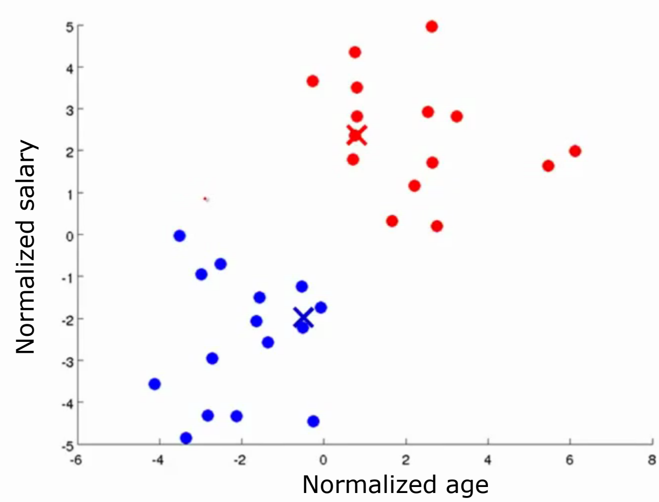
Step 1: Assigning points to nearest center



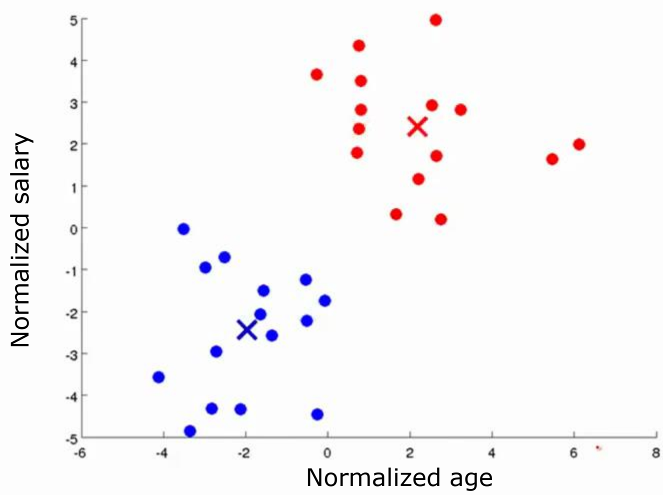
Step 2: Changing the clusters centers



Step 1: Assigning points to nearest center

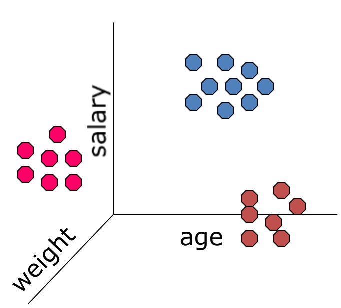


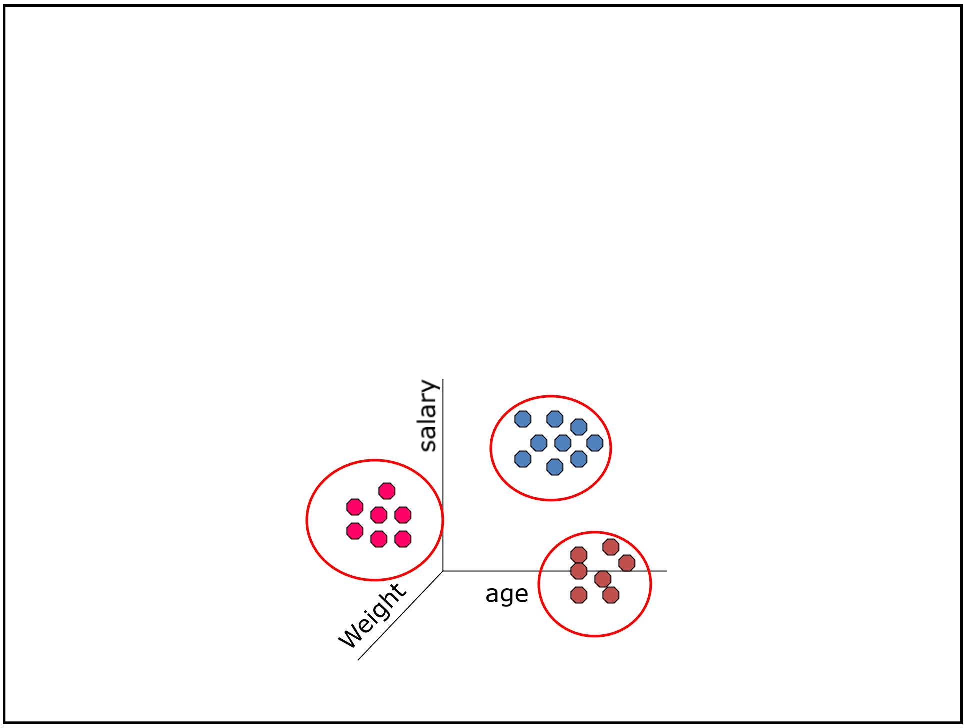
Step 2: Changing the clusters centers

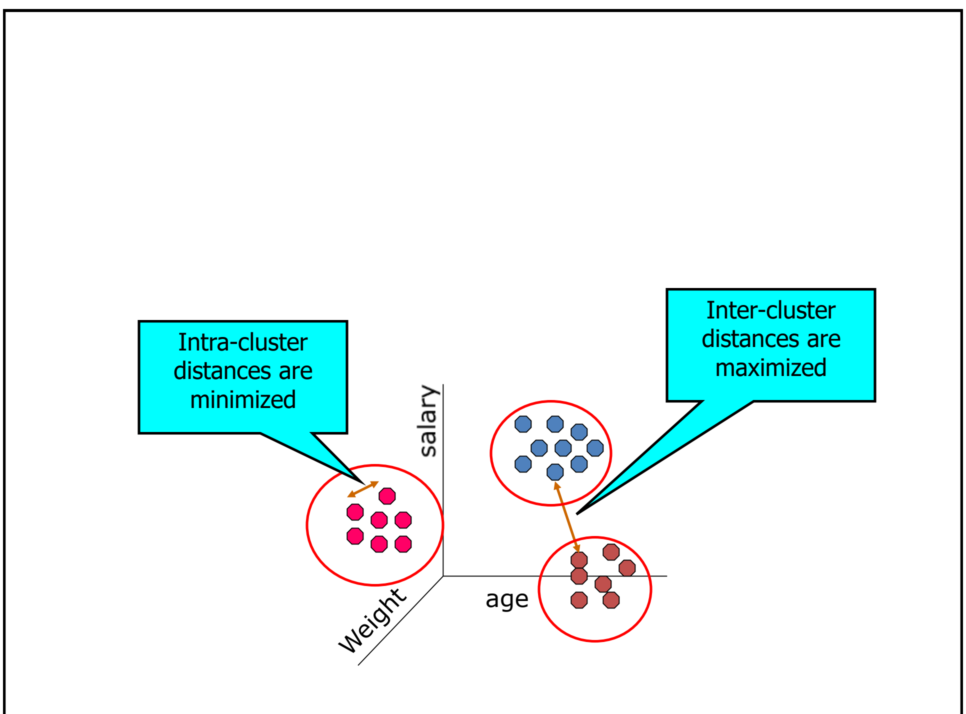


Lab

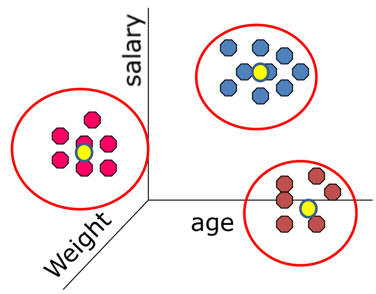
Quality of a solution
and
Model validation




salary
Weight
age

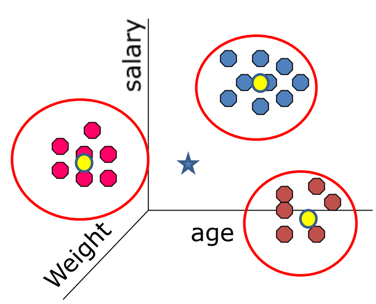

Intra-cluster distances are minimized
Inter-cluster distances are maximized
salary
Weight
age

- Cost function: the sum-of-squared distances from each data point to its cluster center should be minimum

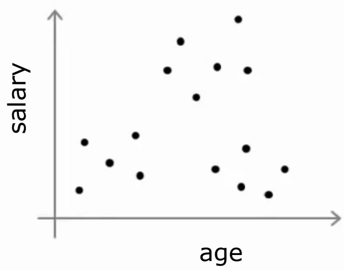


- Total sum of squares = Between sum of squares +
- Within sum of squares(cost function)

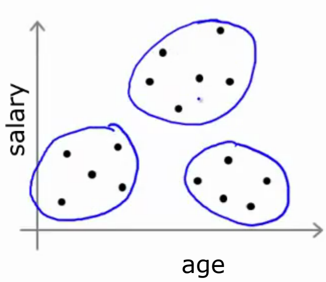
- Between sum of squares/Total sum of squares should be large

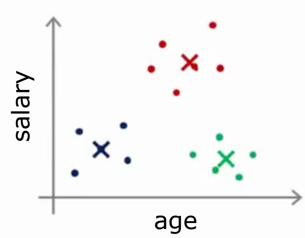


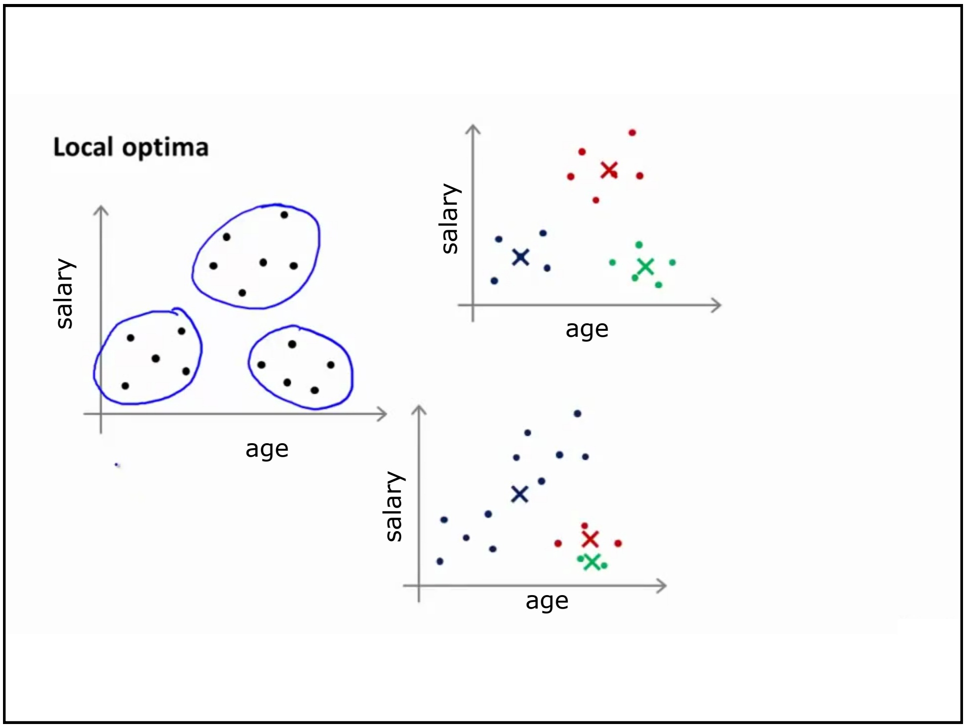
Local optima

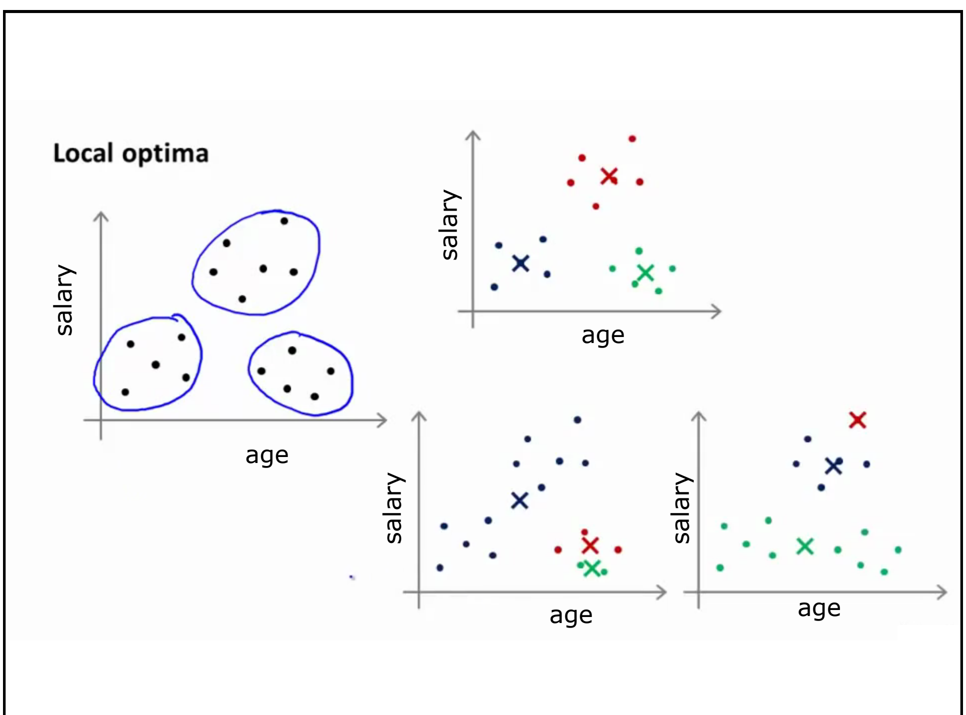


Local optima






Local optima
salary
age
salary
age
salary
age


Local optima
salary
age
salary
age
salary
age
salary
age

How to choose initial centers

- Method 1:Perform multiple run, each with a randomly chosen initial centers
- Best solution is not assured, but commonly followed approach

(a) Iteration 1. (b) Iteration 2.

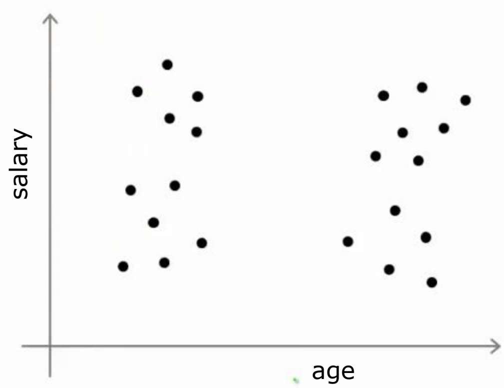
(c) Iteration 3. (d) Iteration 4.

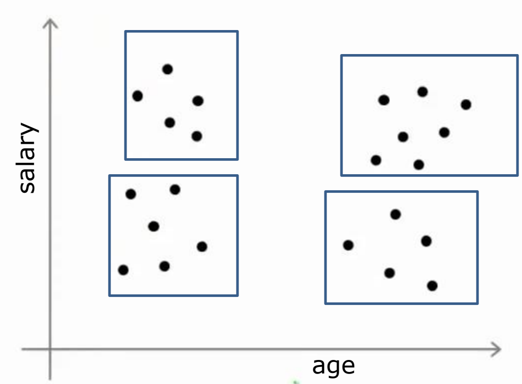
❖ Method 2:

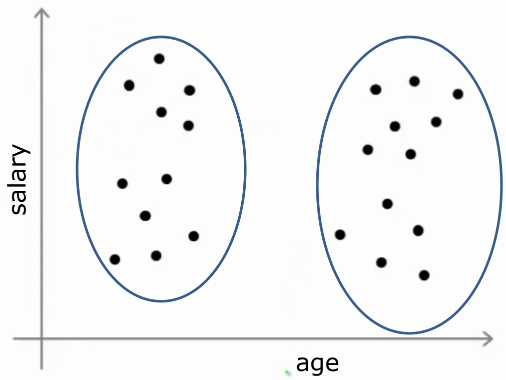
- Select mean of all points as the first center
- Then for each successive center, choose a point that is farthest from the currently chosen centers.
- This approach can pick outliers and computationally quite expensive.
- To overcome this problem apply this approach on a sample of the data points.

Lab

How to choose value of K

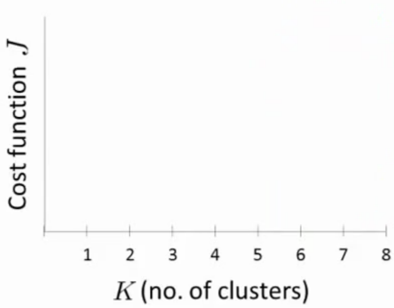

salary
age


salary
age



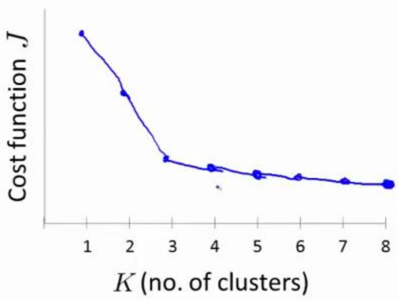
Choosing the value of K

Elbow method:



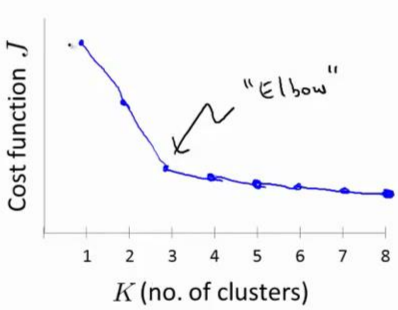
Choosing the value of K

Elbow method:



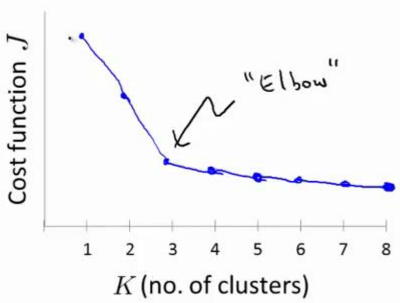
Choosing the value of K

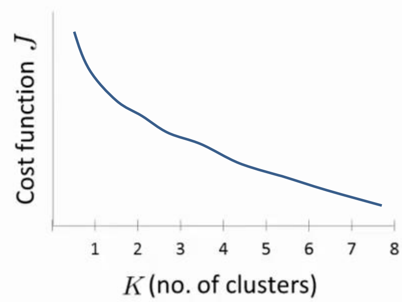
Elbow method:



Choosing the value of K

Elbow method:





Choosing the value of K

Sometimes, you're running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.

E.g.

T-shirt sizing

T-shirt sizing

Few points regarding K-Means

- Normalization : Raw distance measures are highly influenced by scale of measurements. Variables need to be normalized
- Outliers : Outliers need to be removed, unless our requirement is to detect outliers
  - Remove points, which contribute high SSE
  - Remove small clusters
- Empty cluster : This method can give empty cluster. In this case
  - Method 1: Choose a data point farthest away from current centers as the new center.
  - Method 2: Choose a center from the cluster having highest SSE. This will split the cluster with highest SSE into two clusters

K-Means for document Data

Cosine similarity

cos(*d1, d2*) = (*d1* · *d2*) / ||*d1*|| ||*d2*|| : where *dot* indicates vector dot product and ||*d*|| indicates the length of vector *d*

Ex: Find the **similarity** between vectors *d1* and *d2*

d1 **=** (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)
d2 **=** (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

d1 · *d2* = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25

||*d1*|| = sqrt(5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0) = sqrt(42) = 6.481

||*d2*|| = sqrt(3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1) = sqrt(17) = 4.12

cos(*d1, d2*) = 25/(6.481*4.12)=0.94

- Arrange documents in document term matrix format

	eveyrthing	interesting	learning	lerning	like	Machien	machine	not	predicts	problems	solving	sure	What
1	0	1	0	0	1	0	0	0	0	1	1	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	1
3	0	0	0	0	0	0	0	1	0	0	0	1	0
4	1	0	0	1	0	1	0	0	1	0	0	0	0

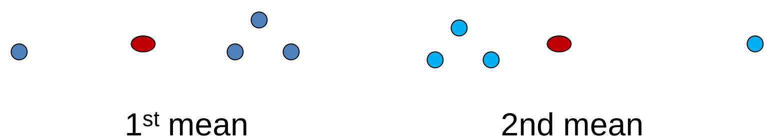
- Use cosine similarity to find nearest center for data point
- Use mean to update cluster centers just like Euclidean distance
- Cost function we are is minimizing is

$$= \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} cosine(\mathbf{x}, \mathbf{c}_i)$$

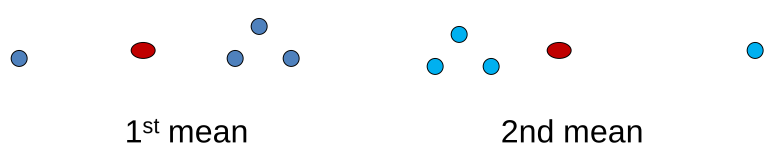
Few drawbacks of K-Means

- K-means not robust.
  - ➢ It gives greater weight to outliers.

- K-means not robust.
  - ➢ It gives greater weight to outliers.



- K-means not robust.
  - ➢ It gives greater weight to outliers.
  - ➢ Handling categorical variable is not straight forward
  - ➢ Work only with few distance measures (Euclidean, Manhattan, Cosine, Bregman divergence)



K-Medoid Algorithm for Euclidean Data

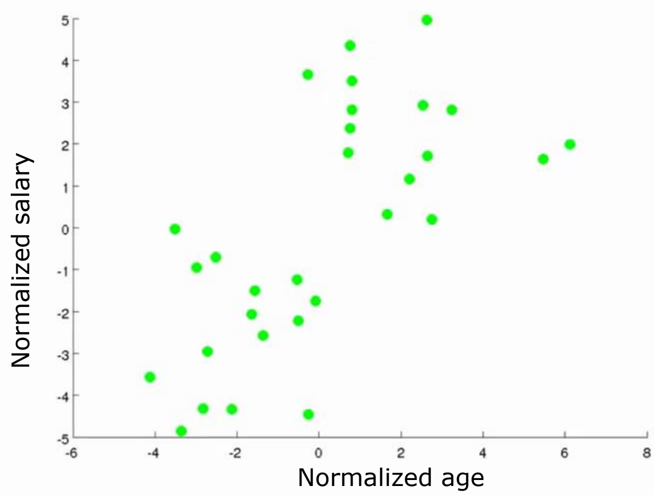
K-medoids

Step 0 : Choose k data points(entities) as initial medoids(centers)

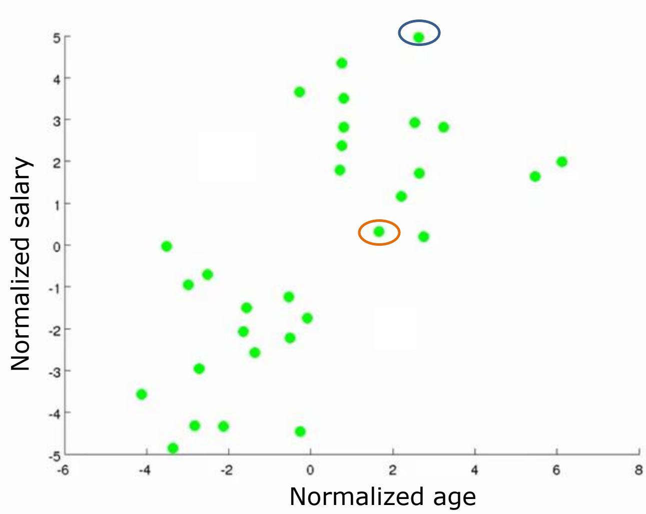
Step 1 : Assign every data point(entity) to its closest medoid(center)

Step 2:
- For each cluster search if any of the data points of the cluster lower the cost function (sum of distances of each data point to its center)
- If it does select the data point that lowers this cost function the most as the medoid(center) for this cluster

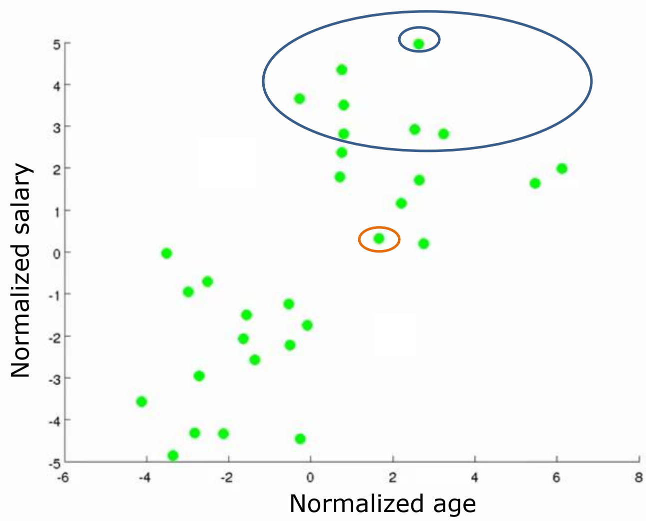
We are restricting the centre to one of the data points assigned to the cluster



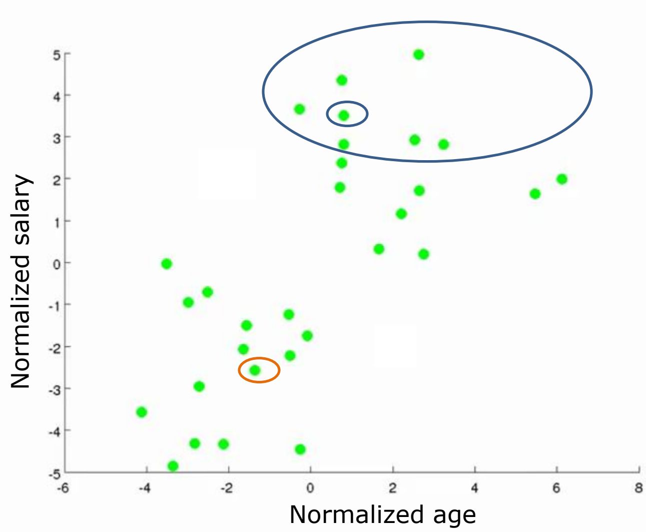
Pre Processing: Select centers by tossing a coin



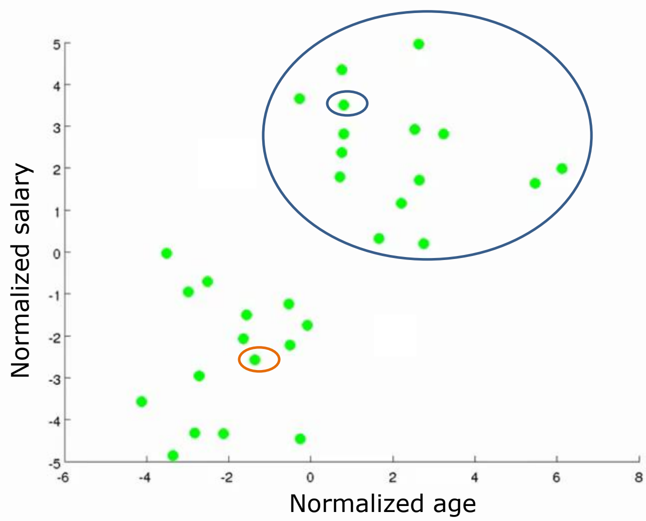
Step 1: Assign every point to nearest center



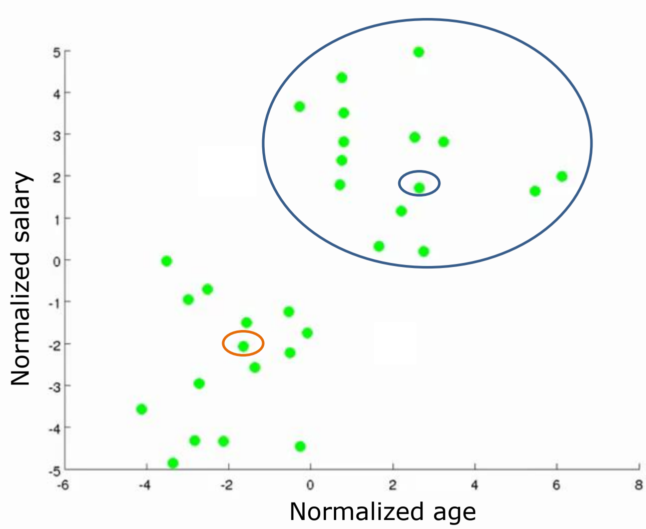
Step 2: Update cluster centers



Step 1: Assign every point to nearest center



Step 2: Update cluster centers



Lab

Manhattan distance

This distance looks at the absolute difference rather than squared Differences

$$d_{ij} = \sum_{m=1}^{p} |x_{im} - x_{jm}|$$

Lab

Other Distance Measures for Numerical Data

Correlation based distance

Removes the influence of scale of measurements and difference in standard deviations

$$r^2{}_{ij} = \frac{\sum_{m=1}^{p}(x_{im}-\bar{x}_i)(x_{jm}-\bar{x}_j)}{\sqrt{\sum_{m=1}^{p}(x_{im}-\bar{x}_i)^2}\sqrt{\sum_{m=1}^{p}(x_{jm}-\bar{x}_j)^2}}$$

Distance measure = $d_{ij} = 1 - r^2{}_{ij}$

Mahalanobis distance

This has an advantage of taking into account the correlation between measurements.

$$d_{ij} = \sqrt{(x_i - x_j)' S^{-1}(x_i - x_j)}$$

Here S^{-1} is the inverse of the covariance matrix

The Mahalanobis distance accounts for the variance of each variable and the covariance between variables.

Geometrically, it does this by transforming the data into standardized uncorrelated data and computing the ordinary Euclidean distance for the transformed data.

Distance Measures for Categorical Data

Distance measure for presence–absence Data

Each variable is a categorical variable with only two possible values

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

	0	1
0	a	b
1	c	d

Data Point 1= 1101000 M=1,F=0
Data Point 2= 0101010 Y=1,N=0

a= 3, b=1 ,c=1, d=2

Similarity metrics based on this table:

- Matching coef. = (a+d)/(a+b+c+d)
- Jaquard's coef. = d/(b+c+d)
  - Use in cases where a matching "1" is much greater evidence of similarity than matching "0"

Matching coef can be extended to nominal data by creating dummy variables

Distance measure for ordinal variables (Kendall's τ)

Company	Q1	Q2	Q3	Q4
XYZ Soft	2	6	4	18
ABC Soft	2	5	4	4

Quarter-wise happiness in the scale of 1 to 20

-	2,2	6,5	4,4	18,4
2,2	-	<<	<<	<<
6,5	>>	-	>>	<>
4,4	>>	<<	-	-(due to equality)
18,4	>>	><	-(due to equality)	-

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{n(n-1)/2}$$

- If the agreement between the two rankings is perfect the coefficient has value 1.
- If the disagreement between the two rankings is perfect the coefficient has value −1.
- If *X* and *Y* are independent, then we would expect the coefficient to be approximately zero.

Distance Measures for Mixed Data

Distance measure for mixed data

Suppose we have both numeric and categorical variables

Compute the distance or dissimilarity metrics D_1, D_2 appropriate to each set of homogeneous variables and then combine these in a weighted average

$$\frac{w_1 * d_1 + w_2 * d_2}{w_1 + w_2}$$

Gowers Distance Measure

Gower's similarity is for mixed variable types: (continuous & categorical)

Lab

Fuzzy Clustering

Hard Clustering

Hard clustering (Ex : Kmeans):

- Data point is deterministically assigned to one and only one cluster
- An object can only belong to single cluster

Hard Clustering

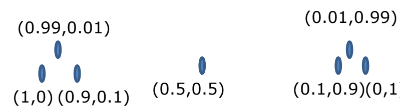
Hard clustering (Ex : Kmeans):

- Data point is deterministically assigned to one and only one cluster
- An object can only belong to single cluster

Hard Vs Fuzzy Clustering

Hard clustering (Ex : Kmeans):

- Data point is deterministically assigned to one and only one cluster
- An object can only belong to single cluster

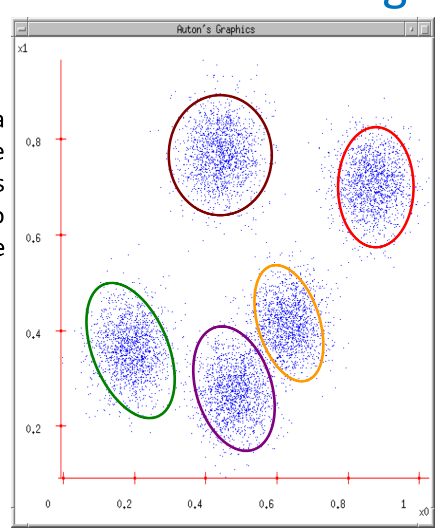


Fuzzy clustering:

- Same object can belong to different clusters
- Given a set of clusters centers, instead of directly assign all data points to their closest clusters, we assign them partially (probabilistically) based on the distances
- Data points are assigned to clusters with certain probabilities

Gaussian Mixture Model for Clustering

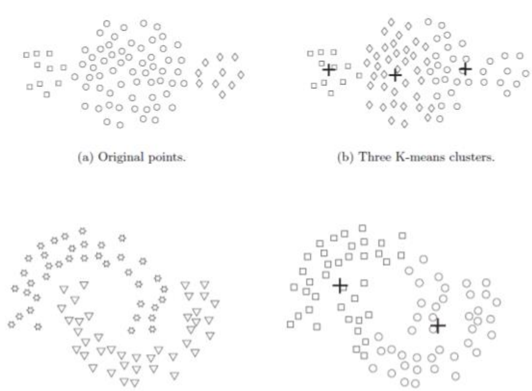
- Assume that data are generated from a mixture of Gaussian distributions. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data
- For each data point
  - Determine membership probabilities
- For each Gaussian distribution
  - Center: μ_i and Variance: Σ_i



Lab

Weakness of K-Means and K-Medoids

- ❖ It cannot handle non-globular clusters
- ❖ It cannot handle cluster of different sizes, densities



(a) Original points. (b) Three K-means clusters.

(a) Original points. (b) Two K-means clusters.

Agglomerative hierarchical clustering

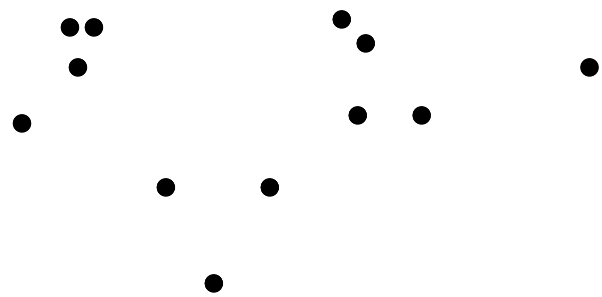
Hierarchical clustering

Agglomerative (Bottom-up)

- Place each of *n* patterns into a class of its own
- Compute inter-cluster similarity scores
- Merge the two most similar clusters into one
  - Replace the two clusters into the new cluster
- Repeat the above two steps until there are *k* clusters left (*k* can be 1)

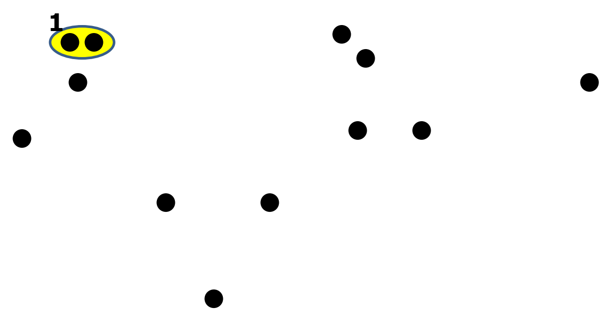
Hierarchical clustering

Agglomerative (Bottom up)



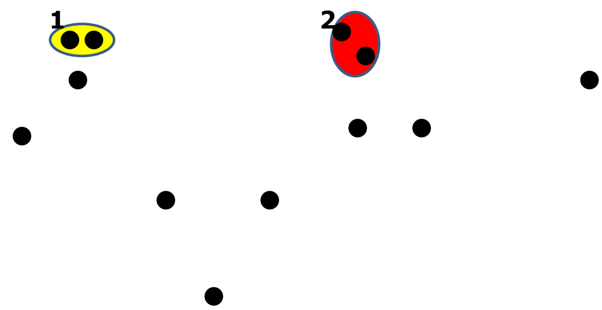
Hierarchical clustering

- Agglomerative (Bottom up)
- 1st iteration



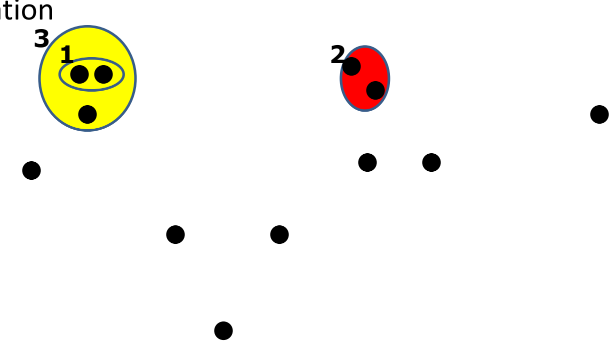
Hierarchical clustering

- Agglomerative (Bottom up)
- 2nd iteration



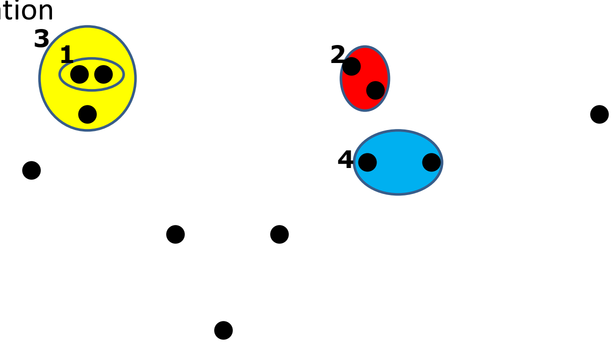
Hierarchical clustering

- Agglomerative (Bottom up)
- 3rd iteration



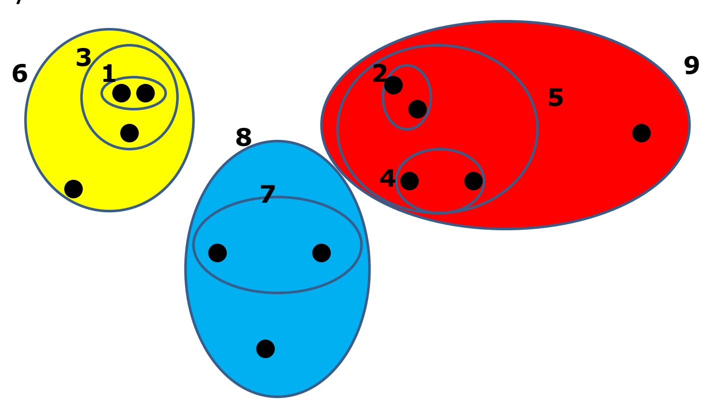
Hierarchical clustering

- Agglomerative (Bottom up)
- 4th iteration



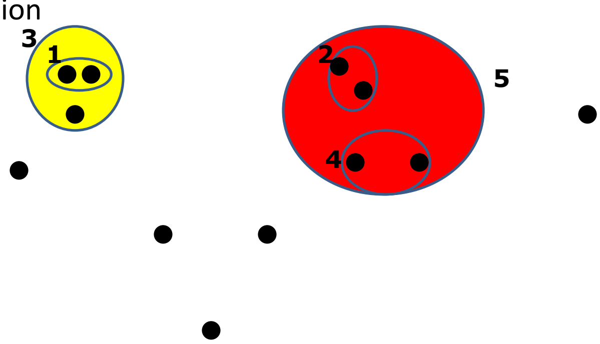
Hierarchical clustering

- Agglomerative (Bottom up)
- Finally k clusters left



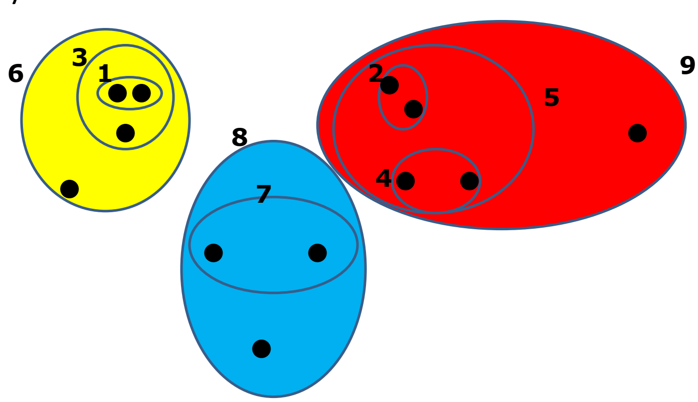
Hierarchical clustering

- Agglomerative (Bottom up)
- 5th iteration



Hierarchical clustering

- Agglomerative (Bottom up)
- Finally k clusters left

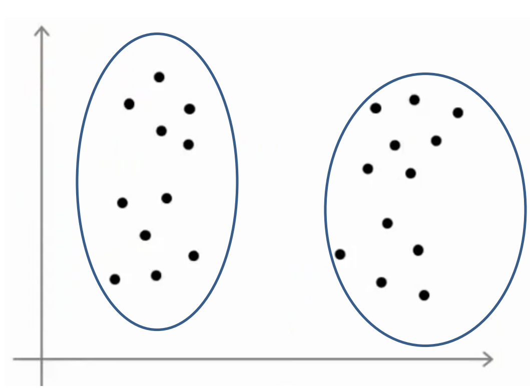


Note that distance between cluster which are getting merged is a non-decreasing function for all three distance measures(single linkage, complete linkage and average linkage)

Measuring Distance Between Clusters

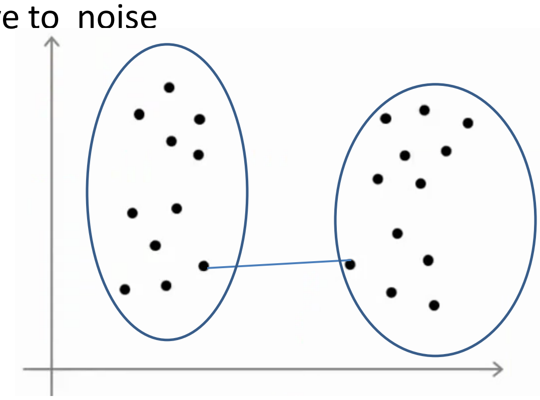
Minimum Distance (Cluster A to Cluster B)

- Also called **single linkage**
- Distance between two clusters is the distance between the pair of records A_i and B_j that are closest



Minimum Distance (Cluster A to Cluster B)

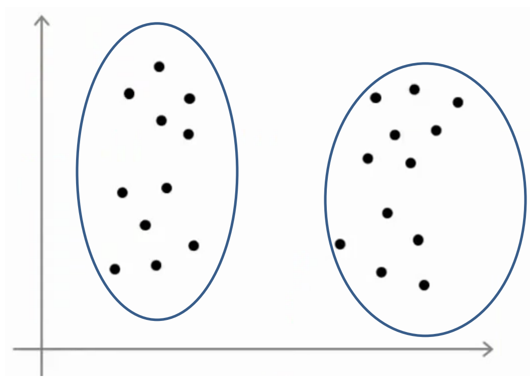
- Also called **single linkage**
- Distance between two clusters is the distance between the pair of records A_i and B_j that are closest
- Sensitive to noise



- A drawback of this method is that it tends to produce long thin clusters in which nearby elements of the same cluster have small distances, but elements at opposite ends of a cluster may be much farther from each other than to elements of other clusters.
- Chaining effect: Noise points that form a bridge between clusters cause single link method to unify these clusters

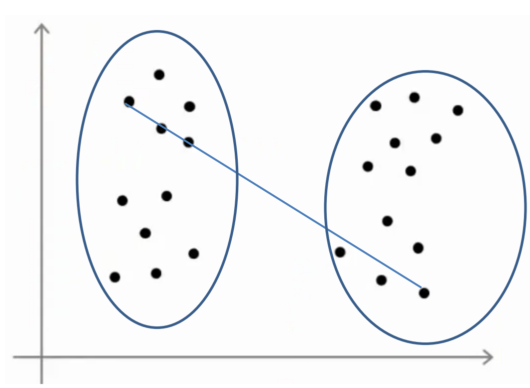
Maximum Distance (Cluster A to Cluster B)

- Also called **complete linkage**
- Distance between two clusters is the distance between the pair of records A_i and B_j that are farthest from each other



Maximum Distance (Cluster A to Cluster B)

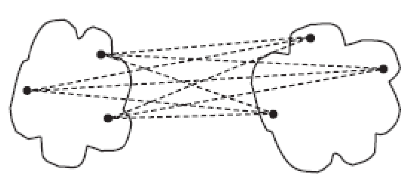
- Also called **complete linkage**
- Distance between two clusters is the distance between the pair of records A_i and B_j that are farthest from each other

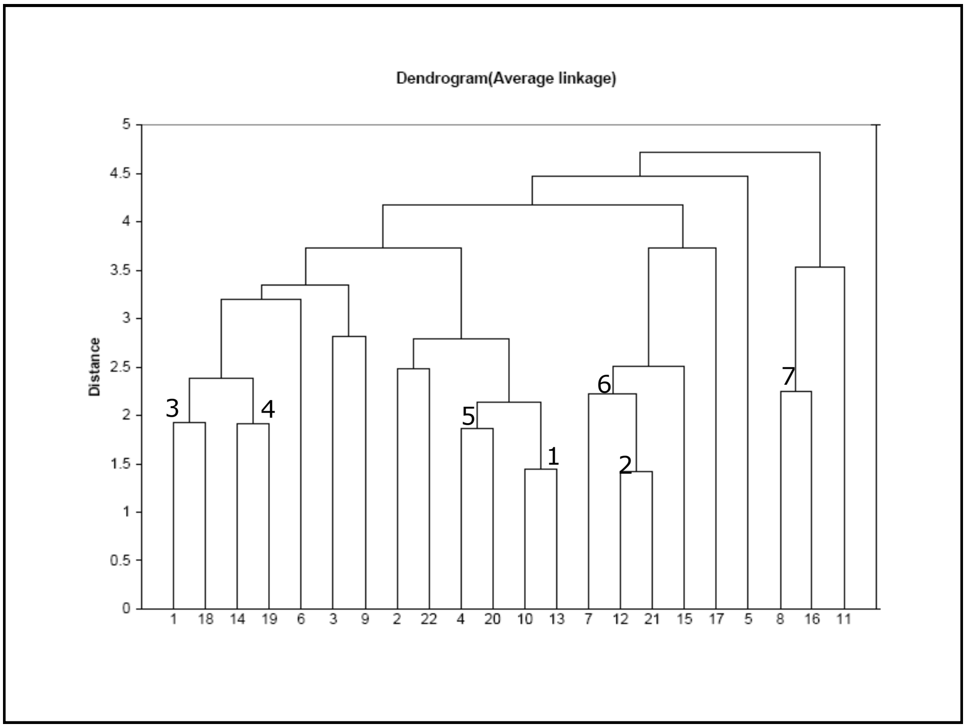


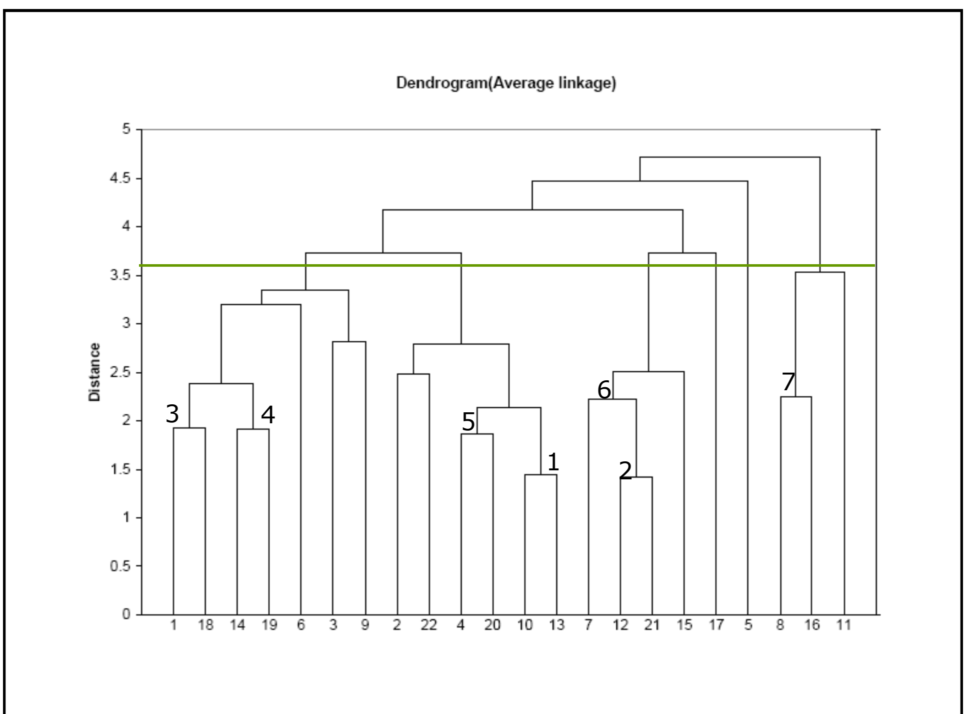
- It gives more weightage to globular shapes
- Sensitive to outliers. A single document far from the center can increase diameters of candidate merge clusters dramatically and completely change the final clustering
- It can break large clusters (if clusters are different sizes)

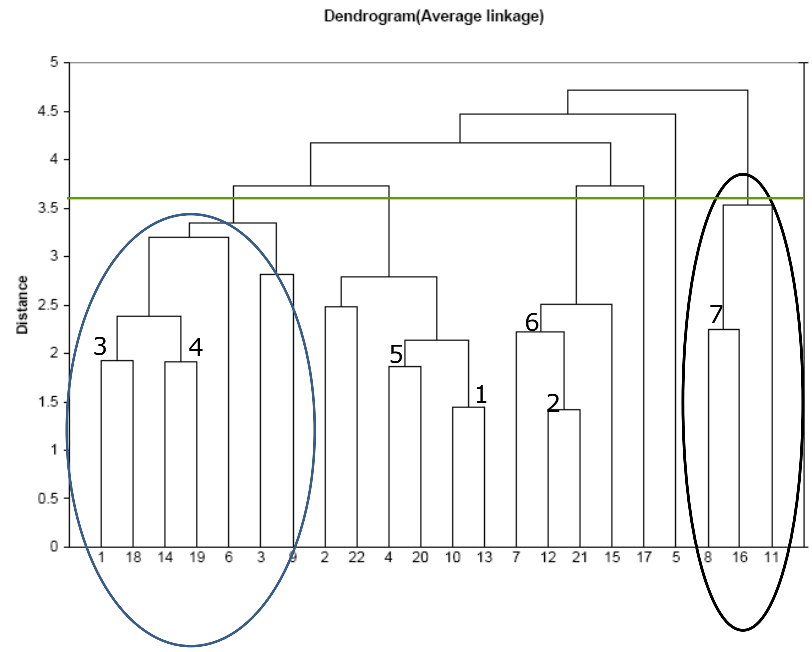
Average Distance

- Also called **average linkage**
- Distance between two clusters is the average of all possible pair-wise distances




Dendrogram(Average linkage)
Distance
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
1 18 14 19 6 3 9 2 22 4 20 10 13 7 12 21 15 17 5 8 16 11
3
4
5
1
6
2
7


Dendrogram(Average linkage)
Distance
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
1 18 14 19 6 3 9 2 22 4 20 10 13 7 12 21 15 17 5 8 16 11
3
4
5
1
6
2
7



Agglomerative Coefficient : which measures the clustering structure of the data set. It is defined as follows:

- Let *d(i)* denote the dissimilarity of object *i* to the first cluster it is merged with, divided by the dissimilarity of the merger in the last step of the algorithm.

- The agglomerative coefficient (*AC)* is defined as the average of all [1-*d*(*i*)]

Element	dissimilarity of object to the first cluster it is merged	dissimilarity of object to the first cluster it is merged with, divided by the dissimilarity of the merger in the last step of the algorithm
X1	2	0.2
X2	4	0.4
X3	5	0.5
X4	6	0.6

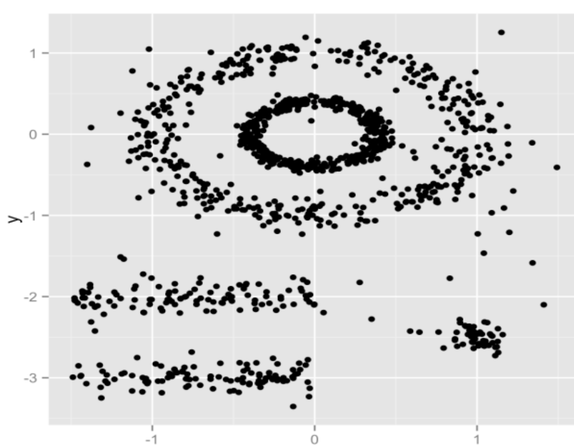
¼*((1-0.2)+(1-0.4)+(1-0.5)+(1-0.6))

Lab

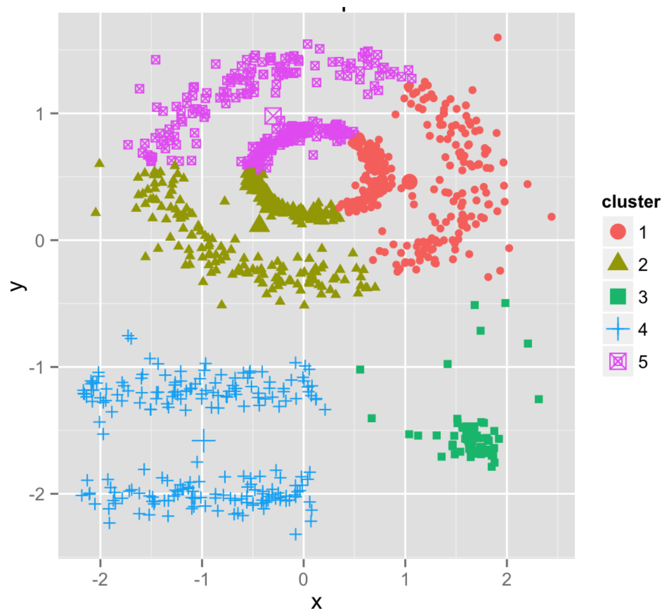
Draw backs of hierarchical

- High time complexity
- It can never undo what was done previously

DBSCAN : Density-based algorithm



The plot above contains 5 clusters and outliers
- 2 ovales clusters
- 2 linear clusters
- 1 compact cluster



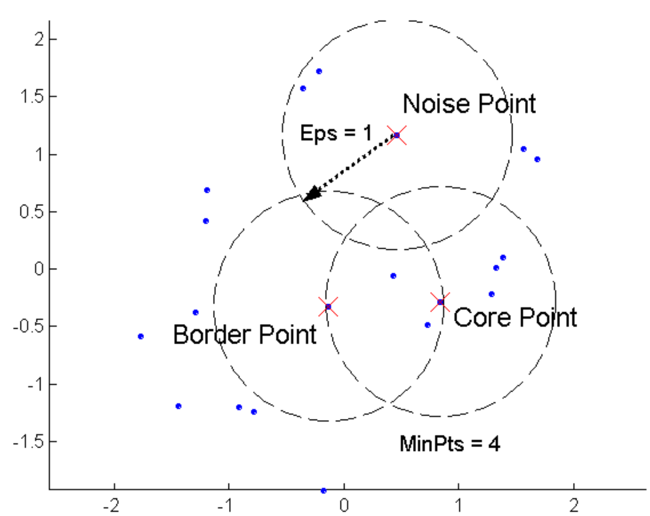
Density Based Clustering

- Clusters are dense regions in the data space, separated by regions of lower object density
- A cluster is defined as a maximal set of density connected points
- Discovers clusters of arbitrary shape and size

- We need to provide two parameters *Eps, MinPts* for this algorithm
- Density of a point is defined as the number points within a specified radius(Eps).
- This algorithm divides the points into three groups based on density

- Core point : A point is a core point if it's density is more than or equal to specified number of points (MinPts)
- Border point : A point is a border point if it's density is less than MinPts, but is in the neighborhood of a core point
- Noise point : A point is a noise point, if it is not a core point or a border point.

Core, Border, and Noise Points



DBSCAN Algorithm

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

DBSCAN Algorithm

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

Determining EPS and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor

Point Number	Distance
1	4
2	3
3	8
4	40
5	6
6	36
7	8
8	30
9	4
10	3

Determining EPS and MinPts

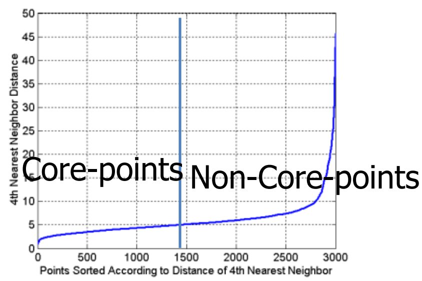
- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor

Point Number	Distance	X Value
2	3	1
10	3	2
1	4	3
9	4	4
5	6	5
3	8	6
7	8	7
8	30	8
6	36	9
4	40	10

Determining EPS and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor

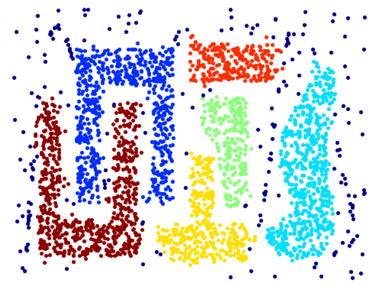
Point Number	Distance	X Value
2	3	1
10	3	2
1	4	3
9	4	4
5	6	5
3	8	6
7	8	7
8	30	8
6	36	9
4	40	10



Lab

When DBSCAN Works Well

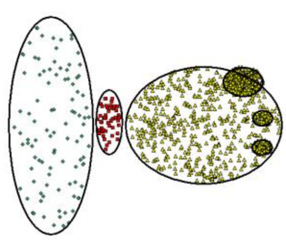
Original Points



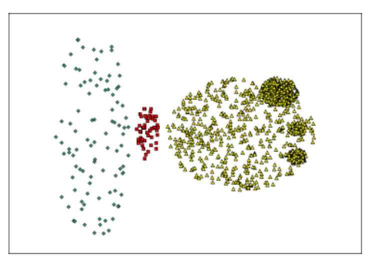
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

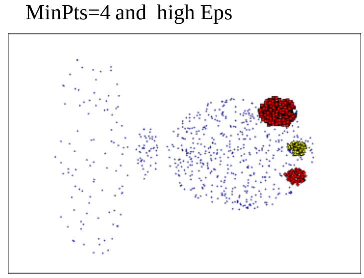
When DBSCAN Does NOT Work Well



Original Points



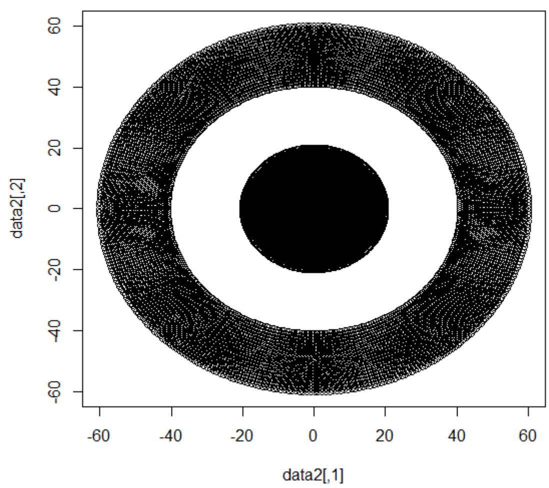
MinPts=4 and high Eps

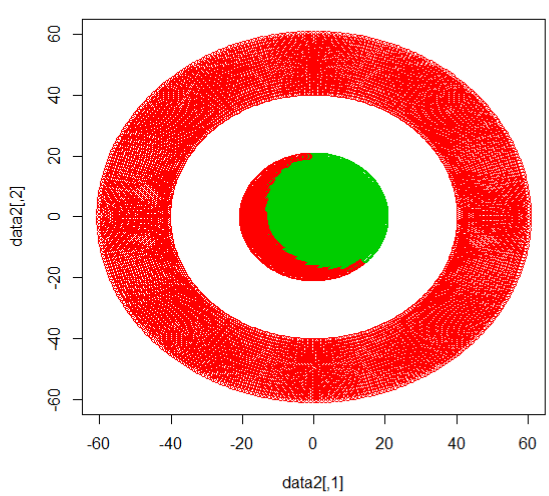


MinPts=4 and low Eps)

- Varying densities
- High-dimensional data

Kernel technique





Lab