Clustering

kMeans,

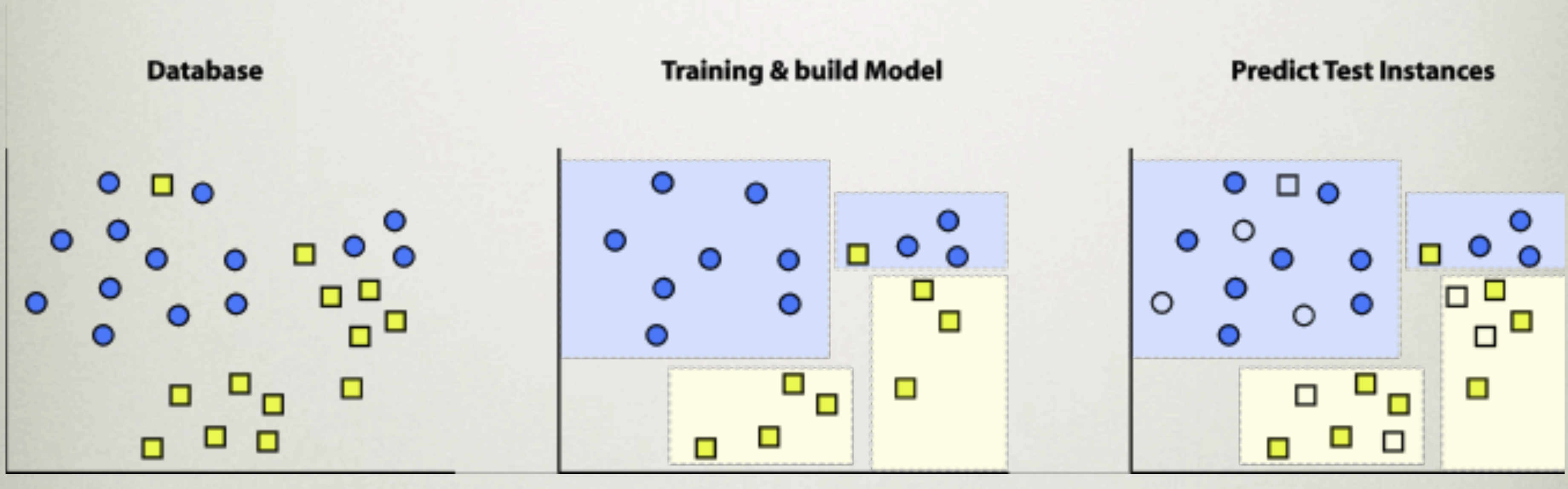
Expectation Maximization,

Self-Organizing Maps

Outline

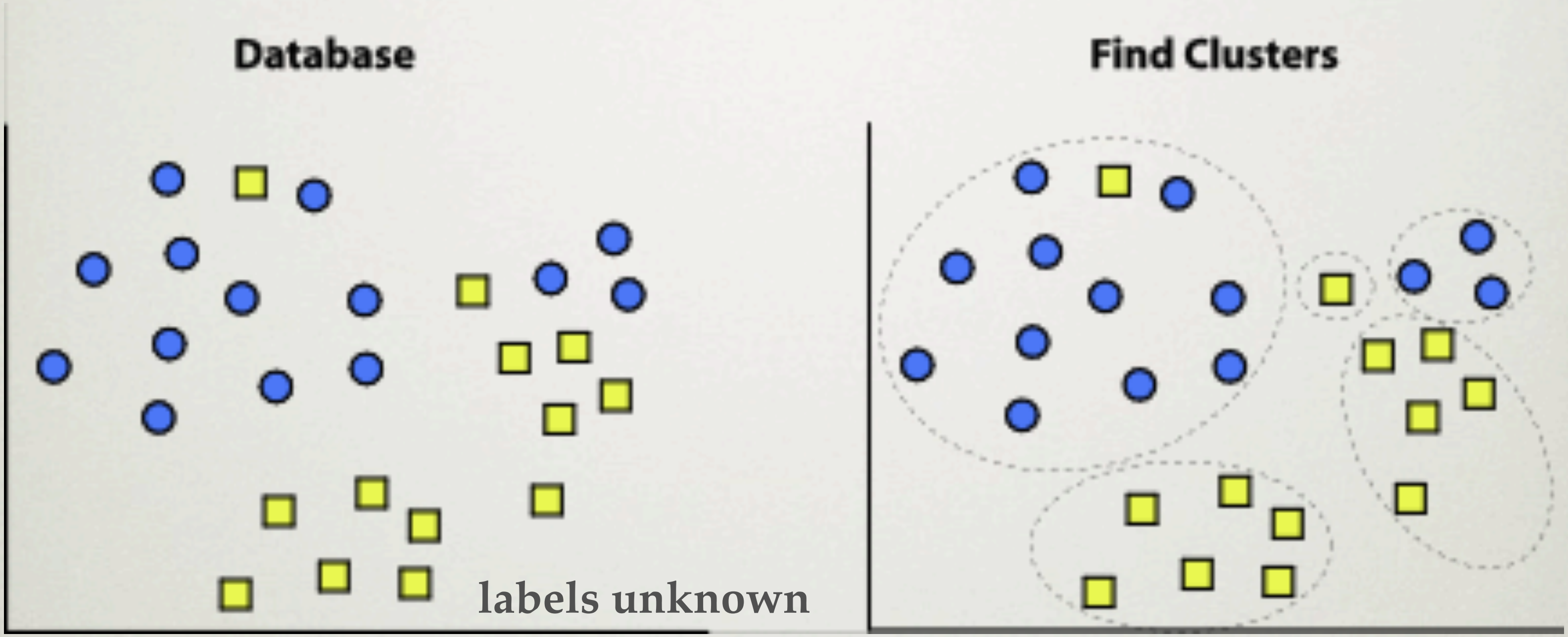
- K-means clustering
- Hierarchical clustering
- Incremental clustering
- Probability-based clustering
- Self-Organising Maps

Classification vs. Clustering



Classification: ***Supervised*** **learning (labels given)**

Classification vs. Clustering



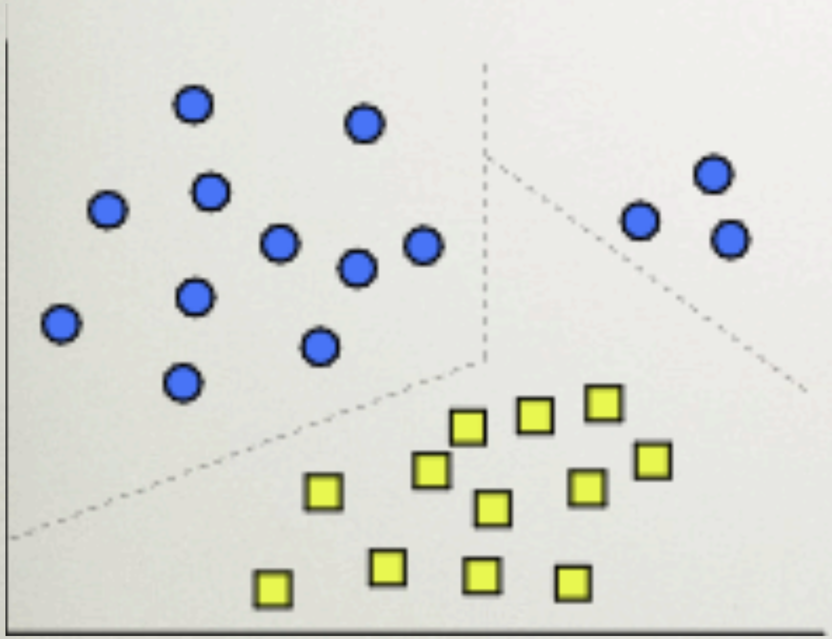
Clustering: ***Unsupervised*** **learning**

No labels, find “natural” grouping of instances

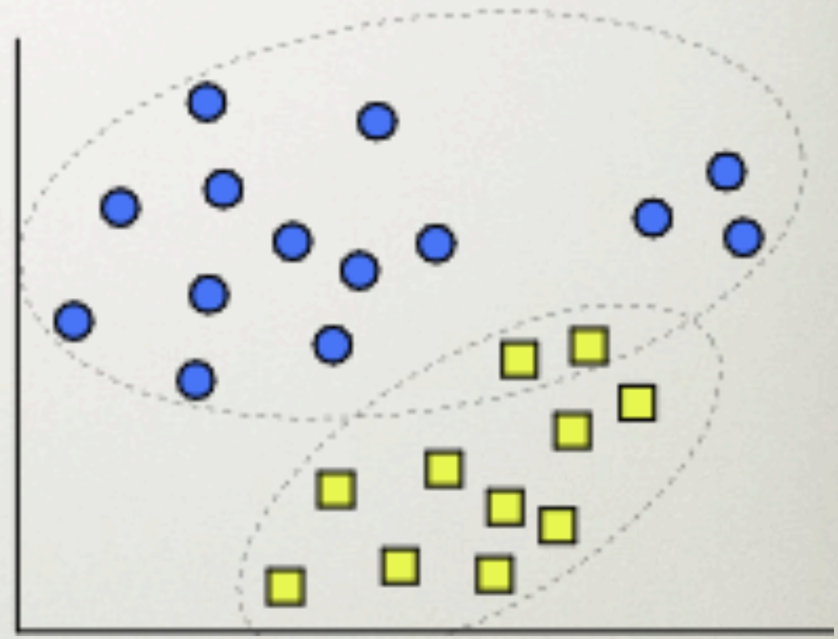
Many Applications!

- Basically, everywhere labels are unknown/uncertain/too expensive
  - **Marketing:** find groups of similar customers
  - **Astronomy:** find groups of similar stars, galaxies
  - **Earth-quake studies:** cluster earth quake epicenters along continent faults
  - **Genomics:** find groups of genes with similar expressions

Clustering Methods: Terminology

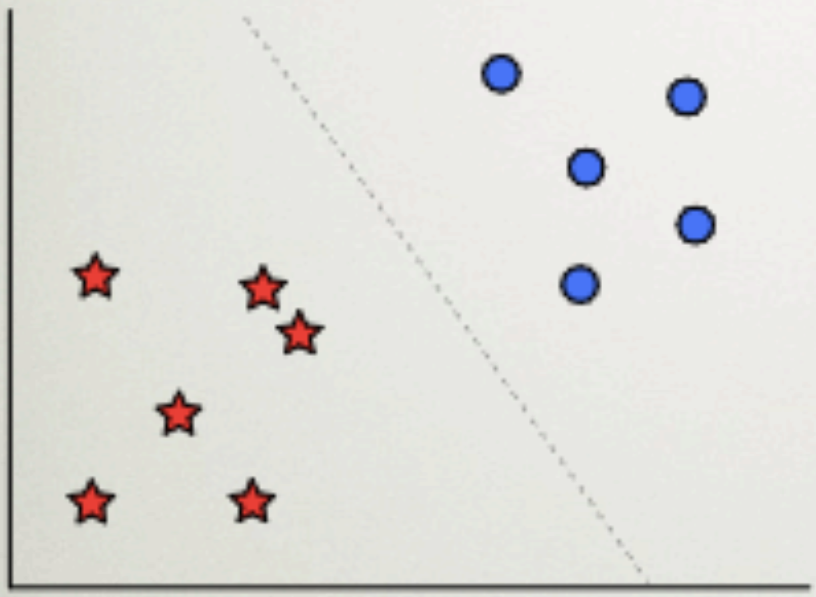


Non-overlapping

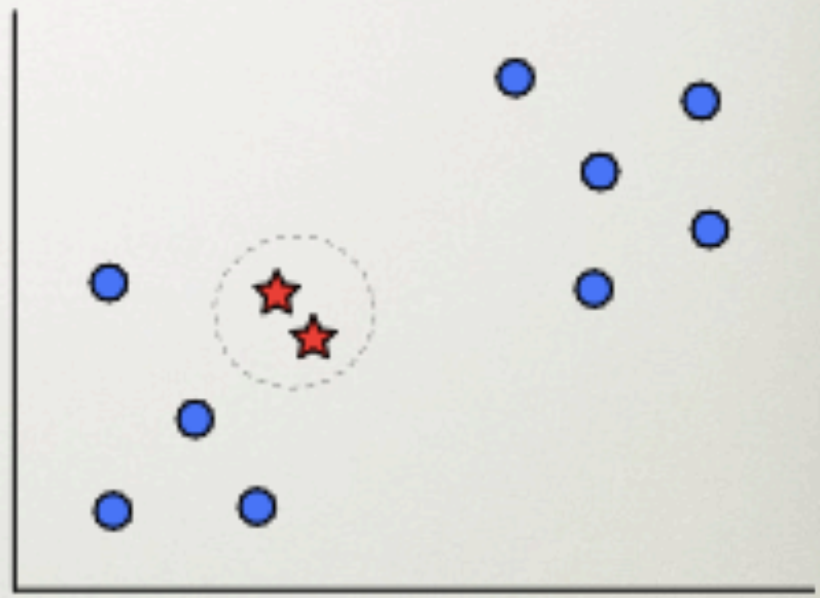


Overlapping

Clustering Methods: Terminology

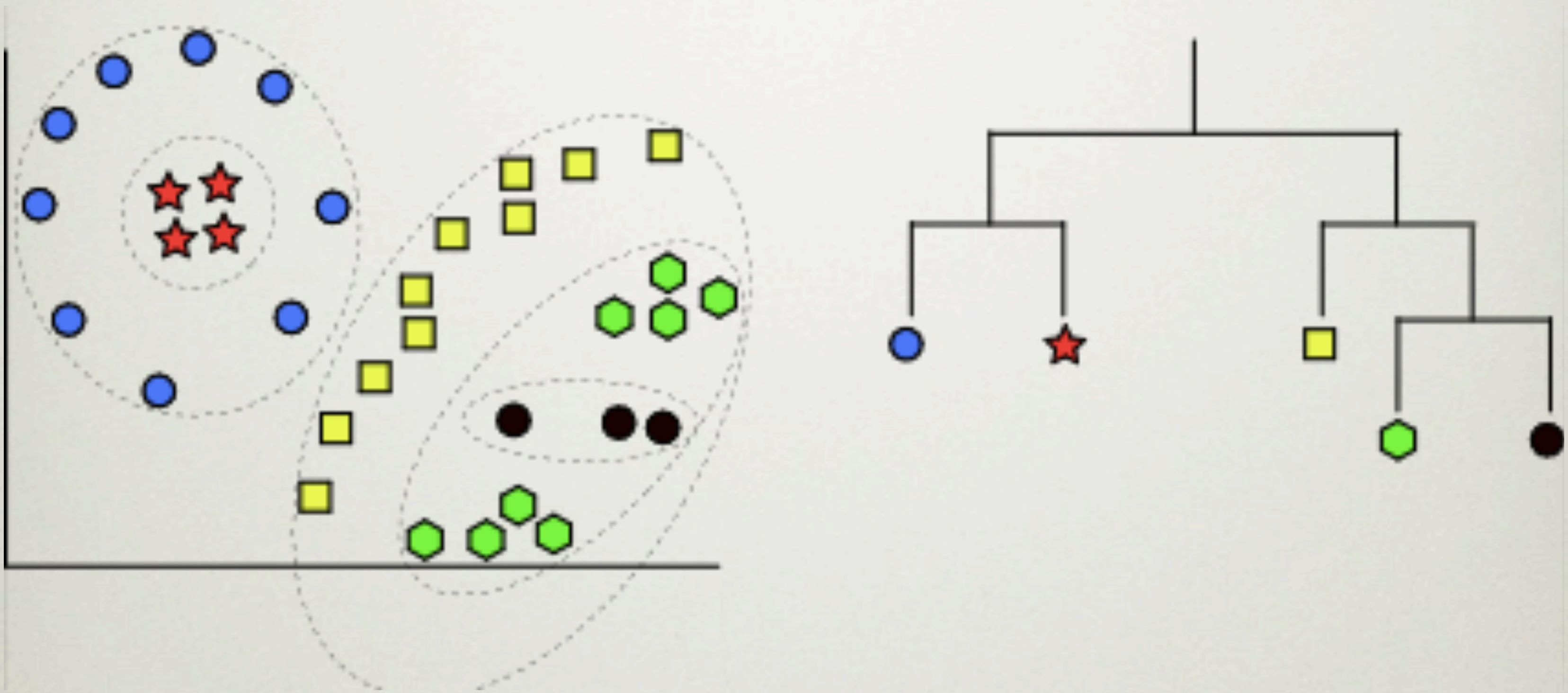


Top-down



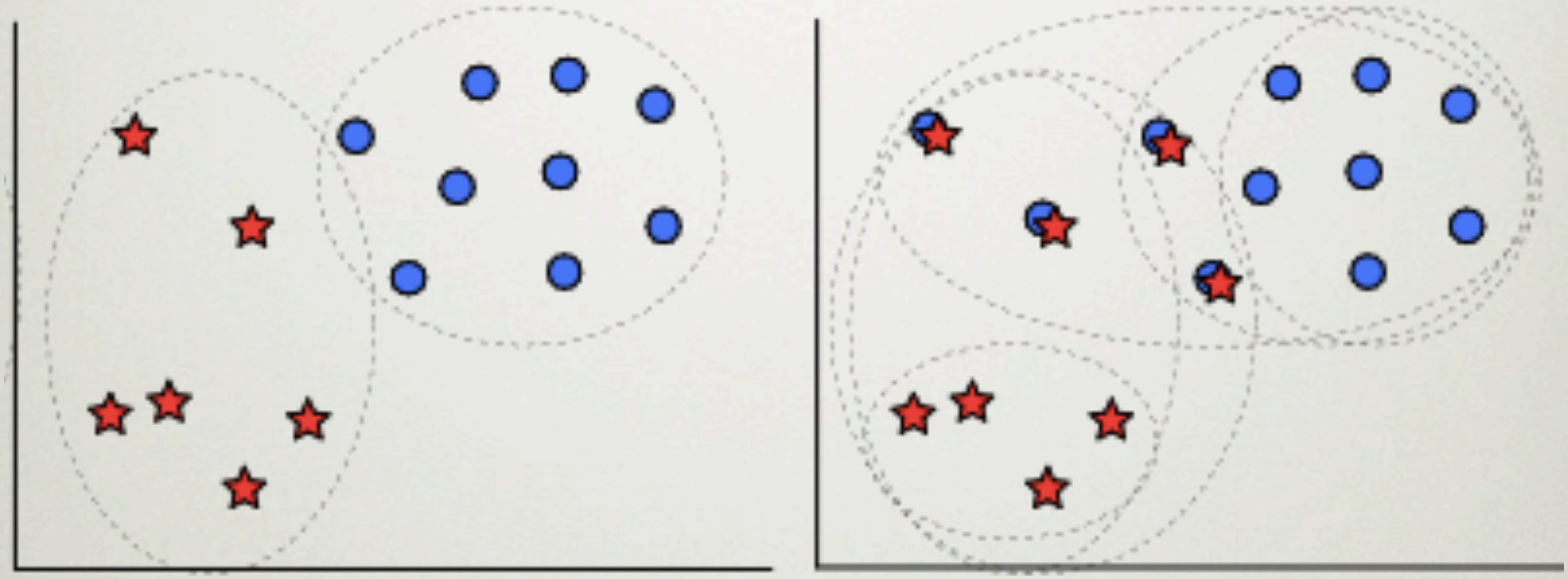
Bottom-up
(agglomerative)

Clustering Methods: Terminology



Hierarchical

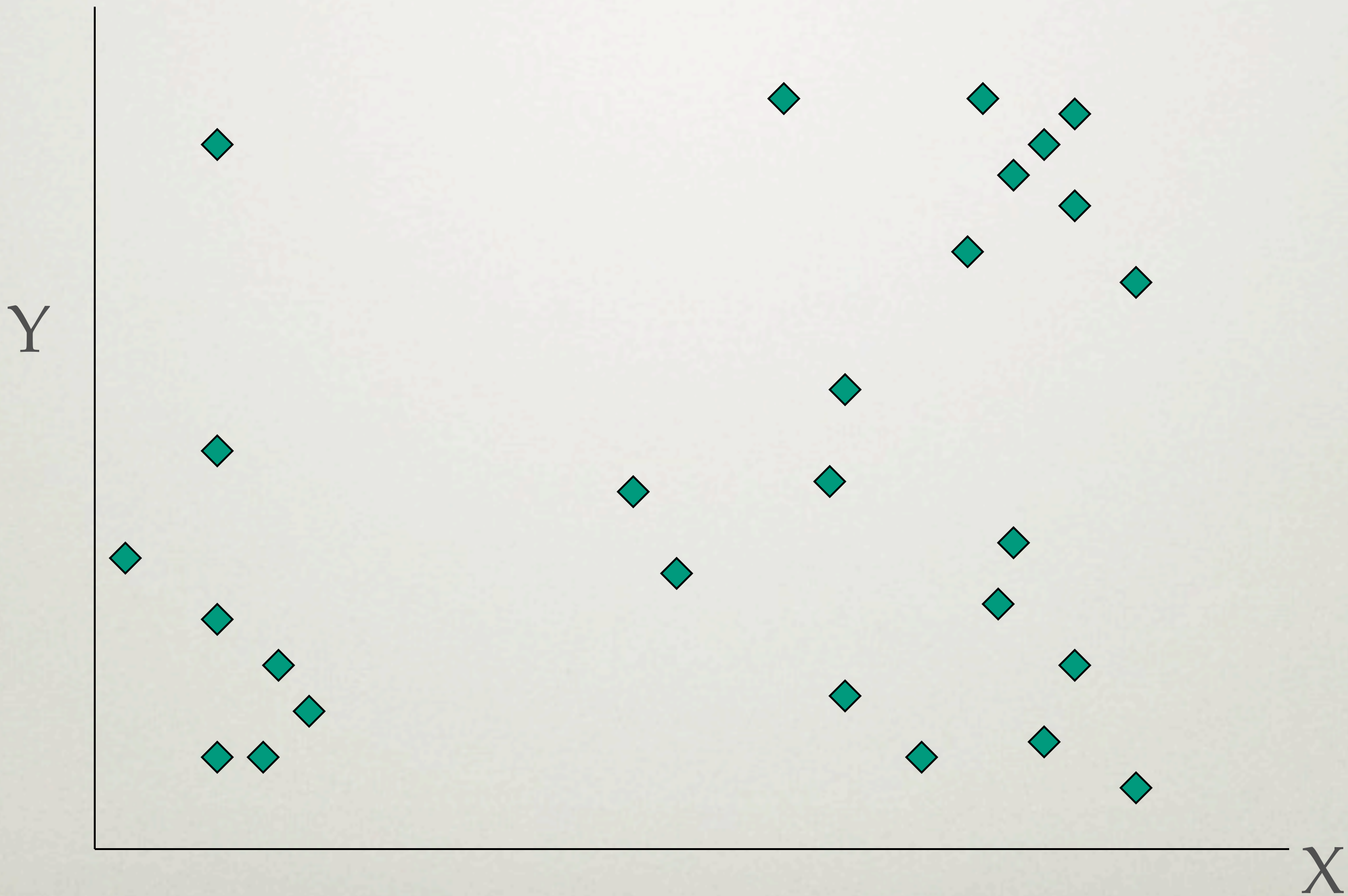
Clustering Methods: Terminology



Deterministic Probabilistic

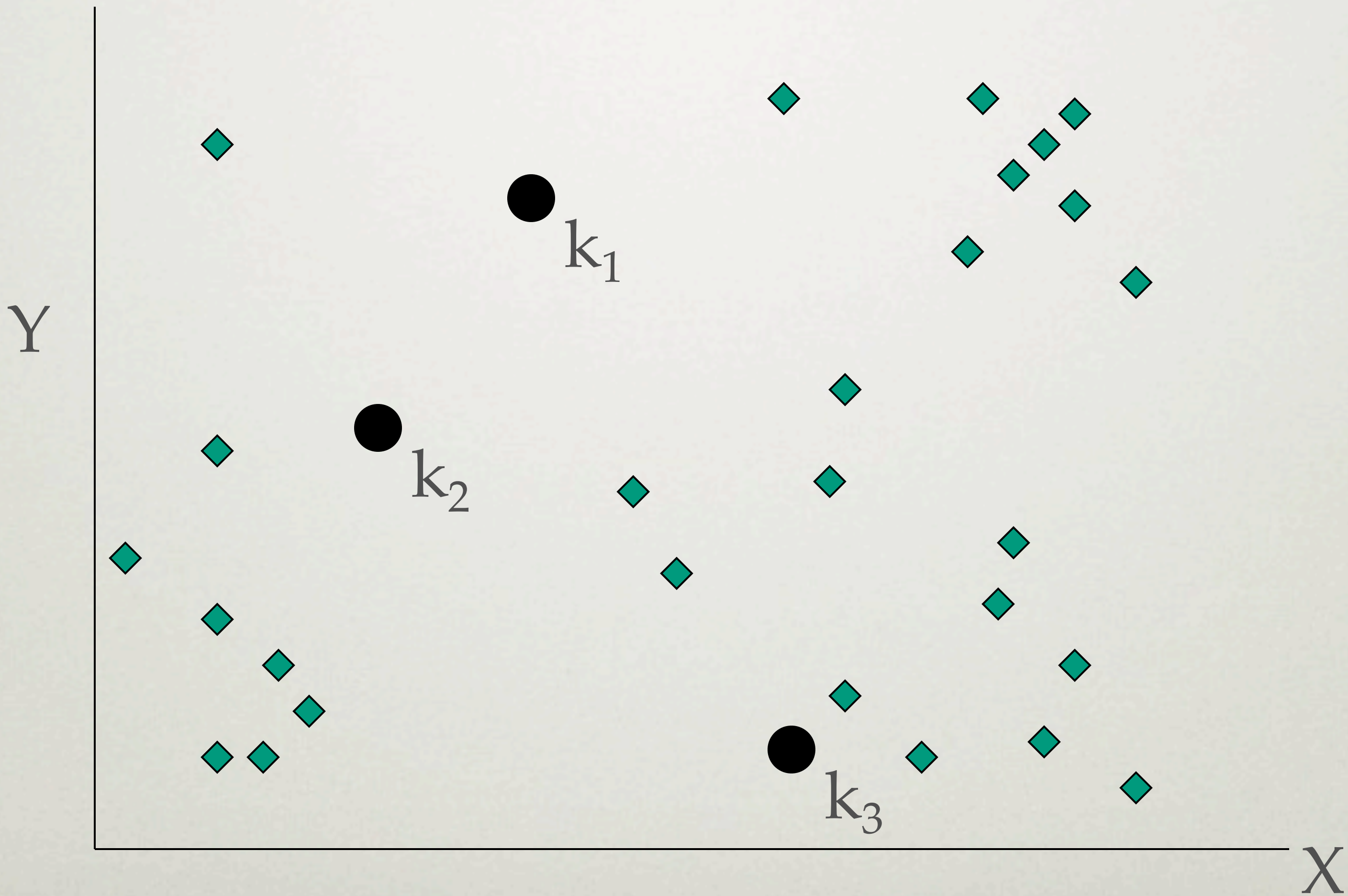
K-Means Clustering

K-means clustering (k=3)



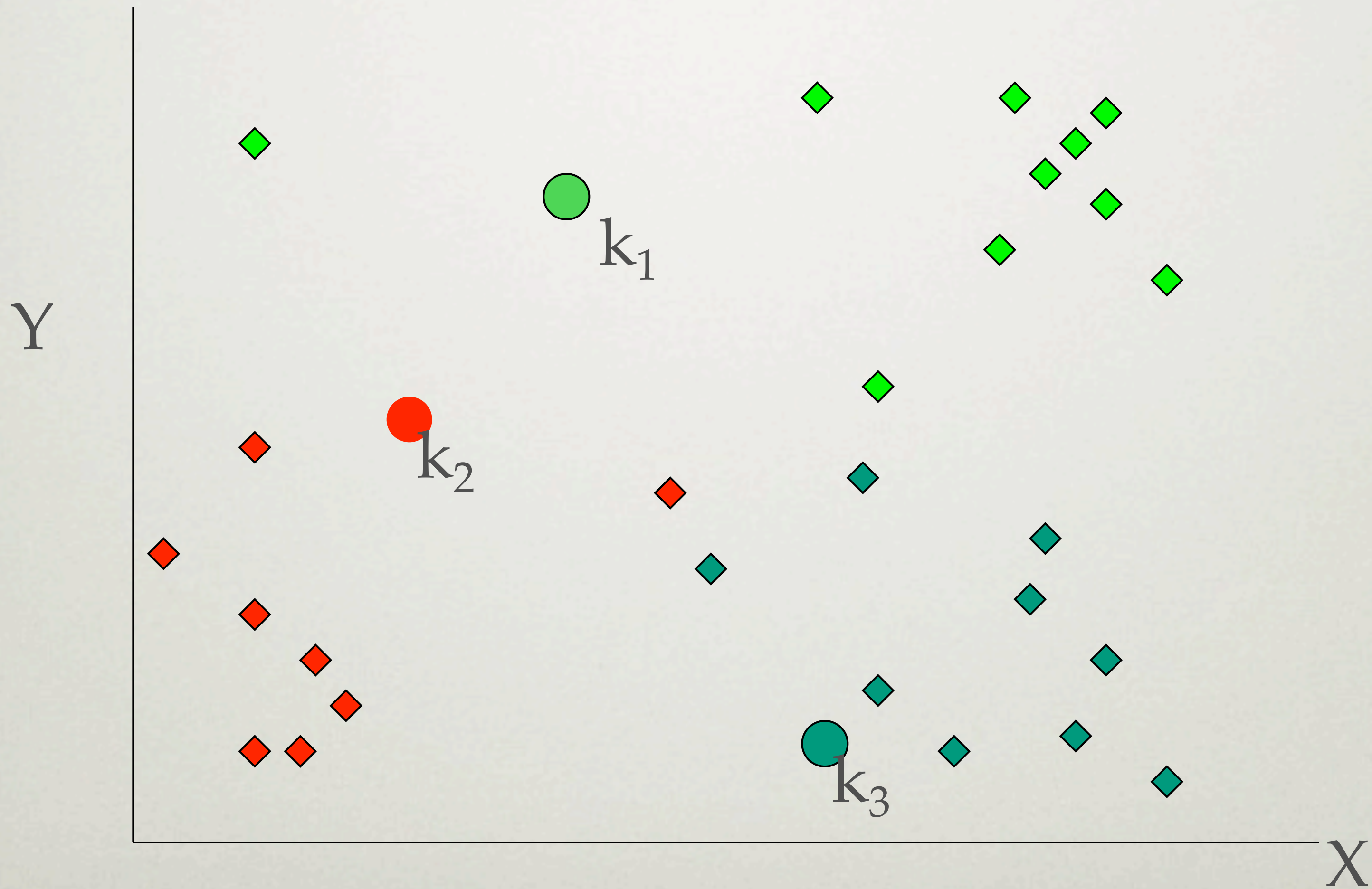
Pick k random points: initial cluster centers

K-means clustering (k=3)



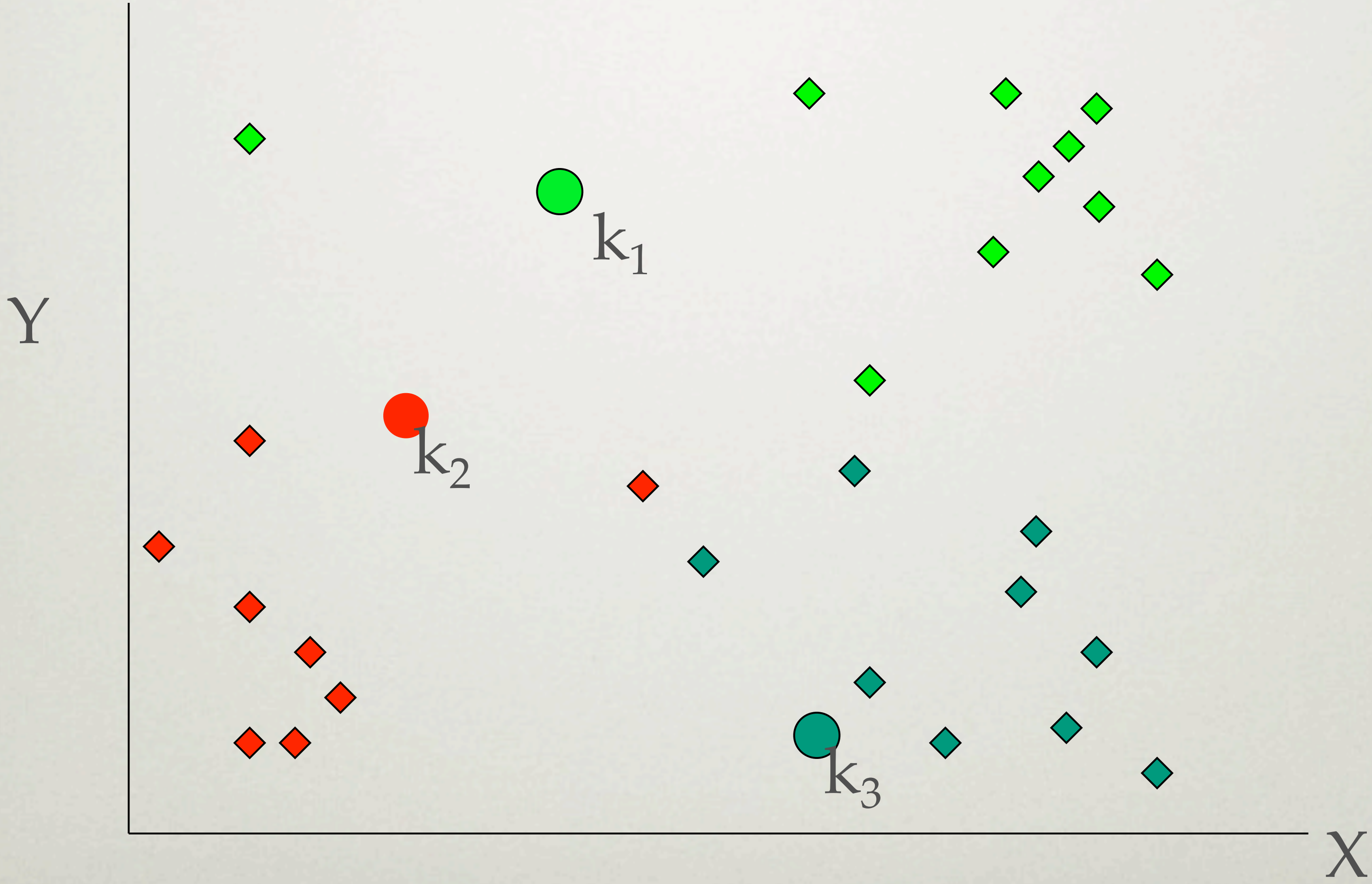
Pick k random points: initial cluster centers

K-means clustering (k=3)



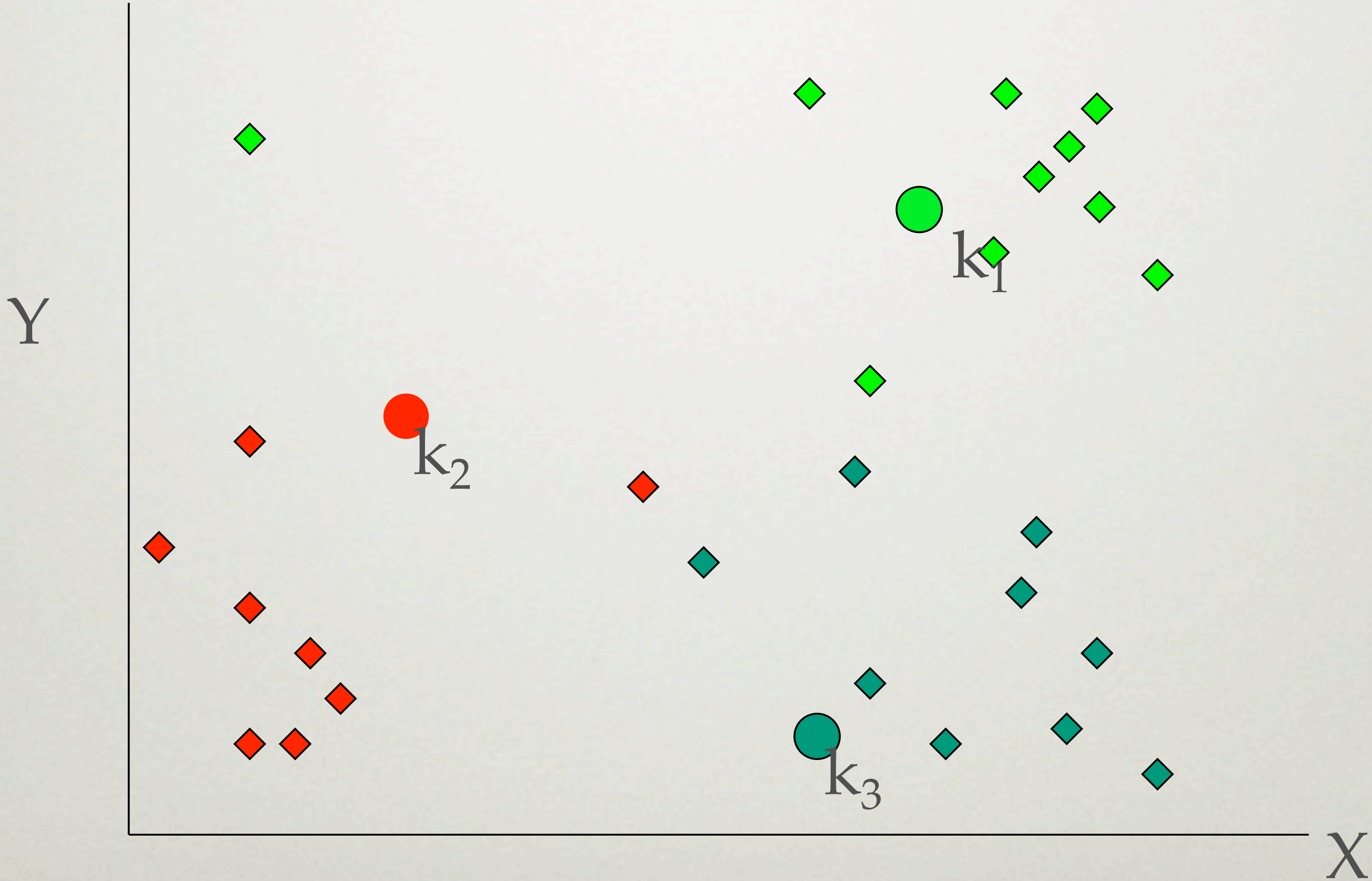
Assign each point to nearest cluster center

K-means clustering (k=3)



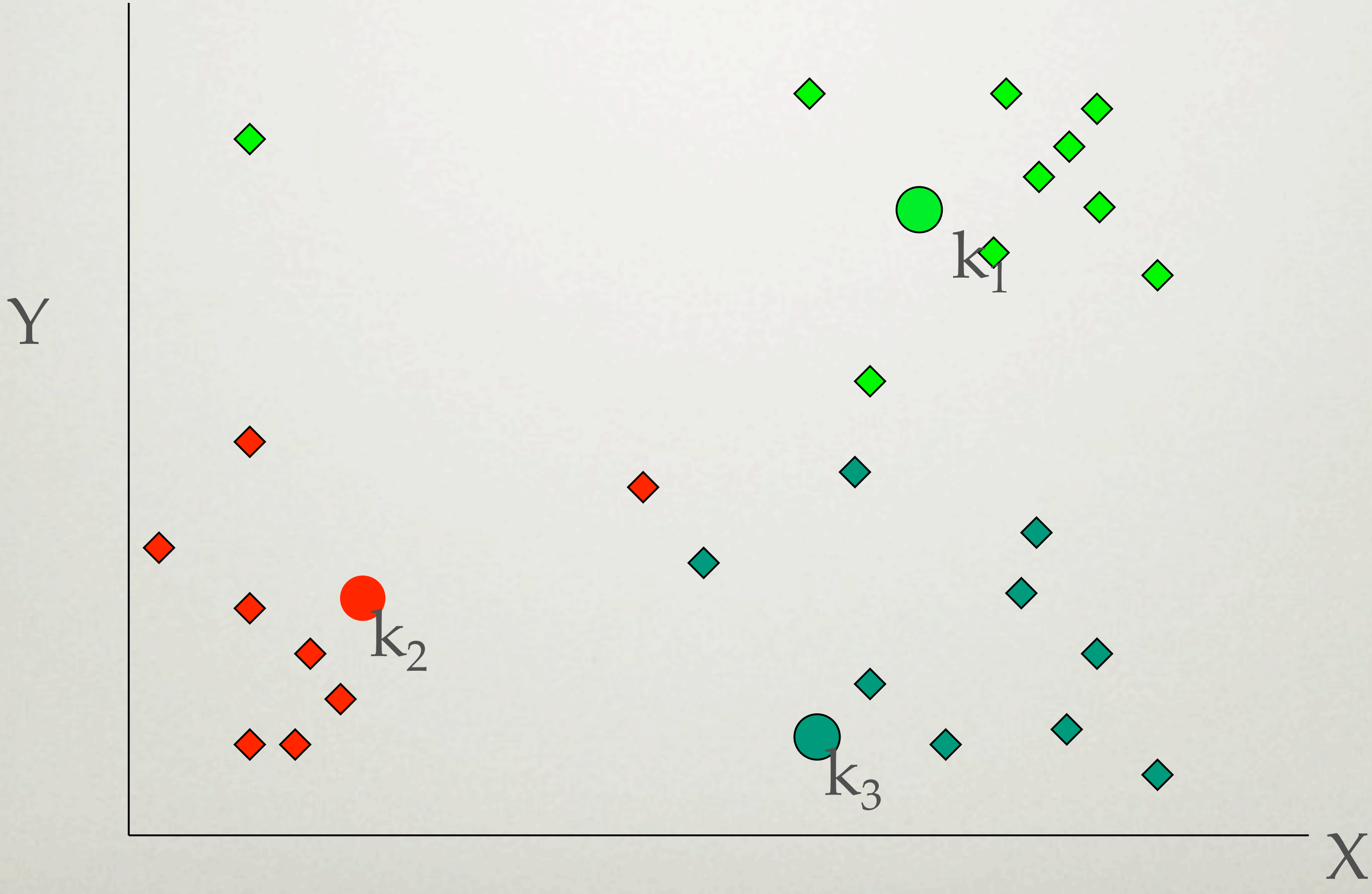
Move cluster centers to *mean* of each cluster

K-means clustering (k=3)



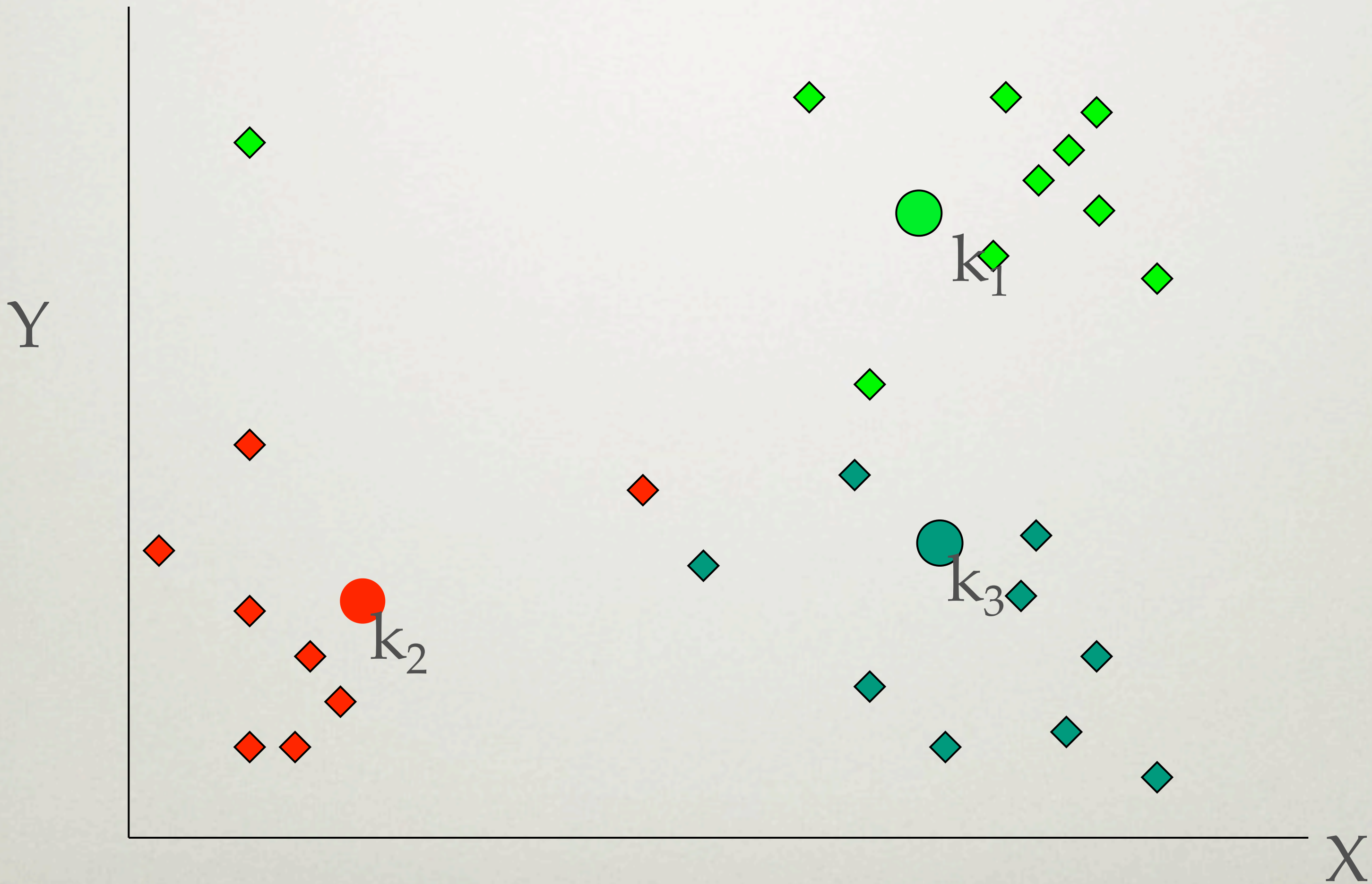
Move cluster centers to *mean* of each cluster

K-means clustering (k=3)



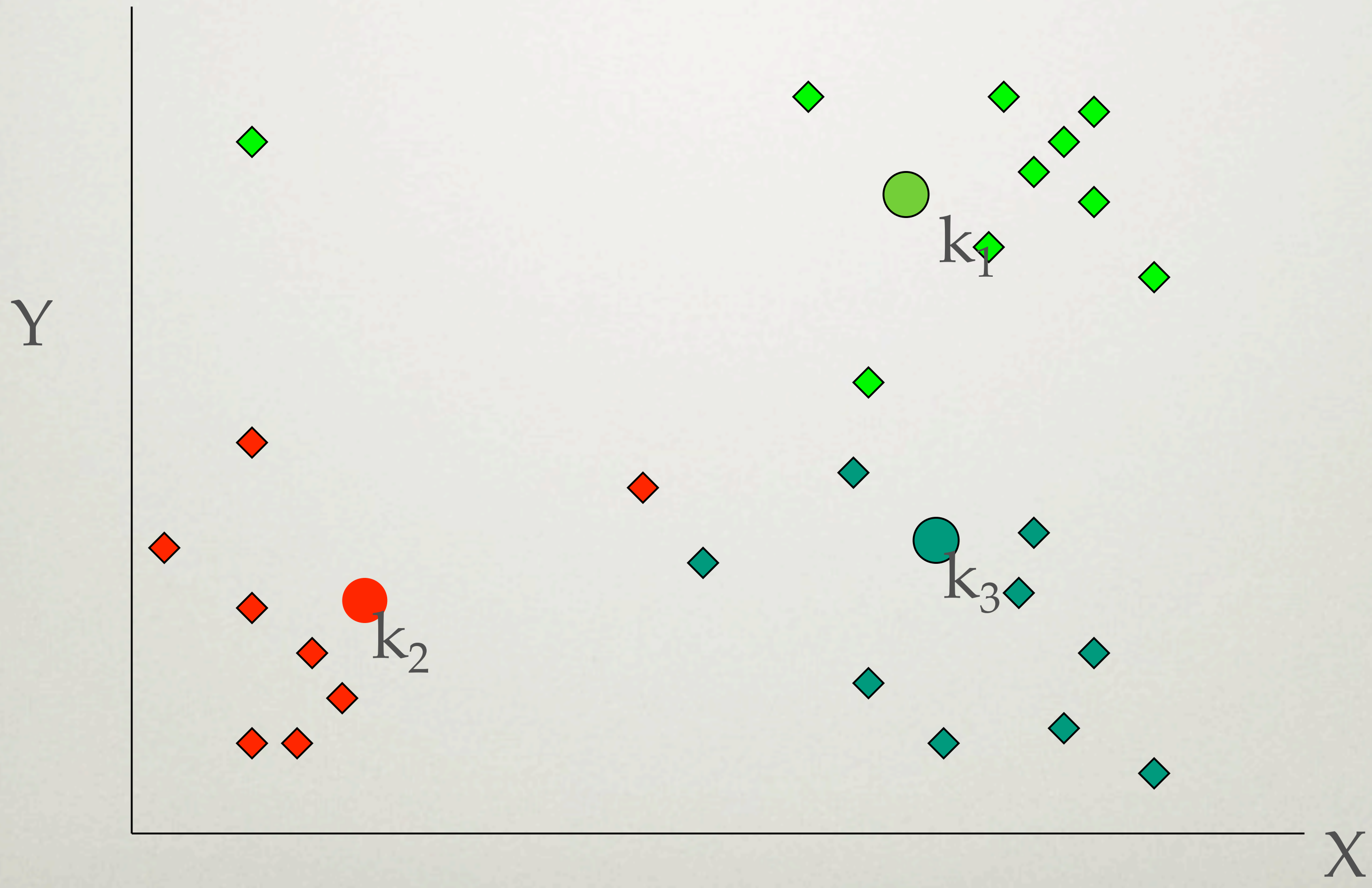
Move cluster centers to *mean* of each cluster

K-means clustering (k=3)



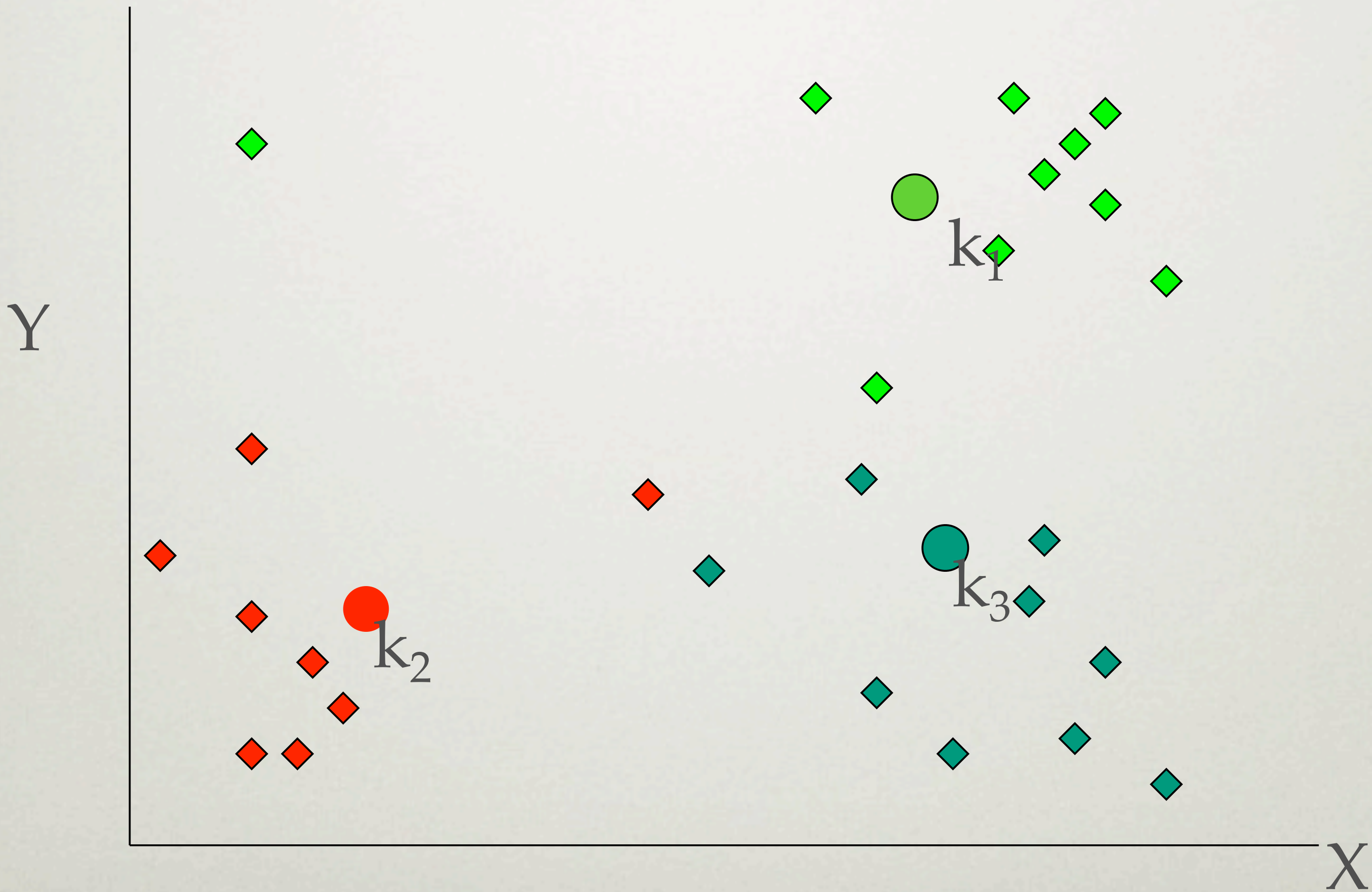
Move cluster centers to *mean* of each cluster

K-means clustering (k=3)



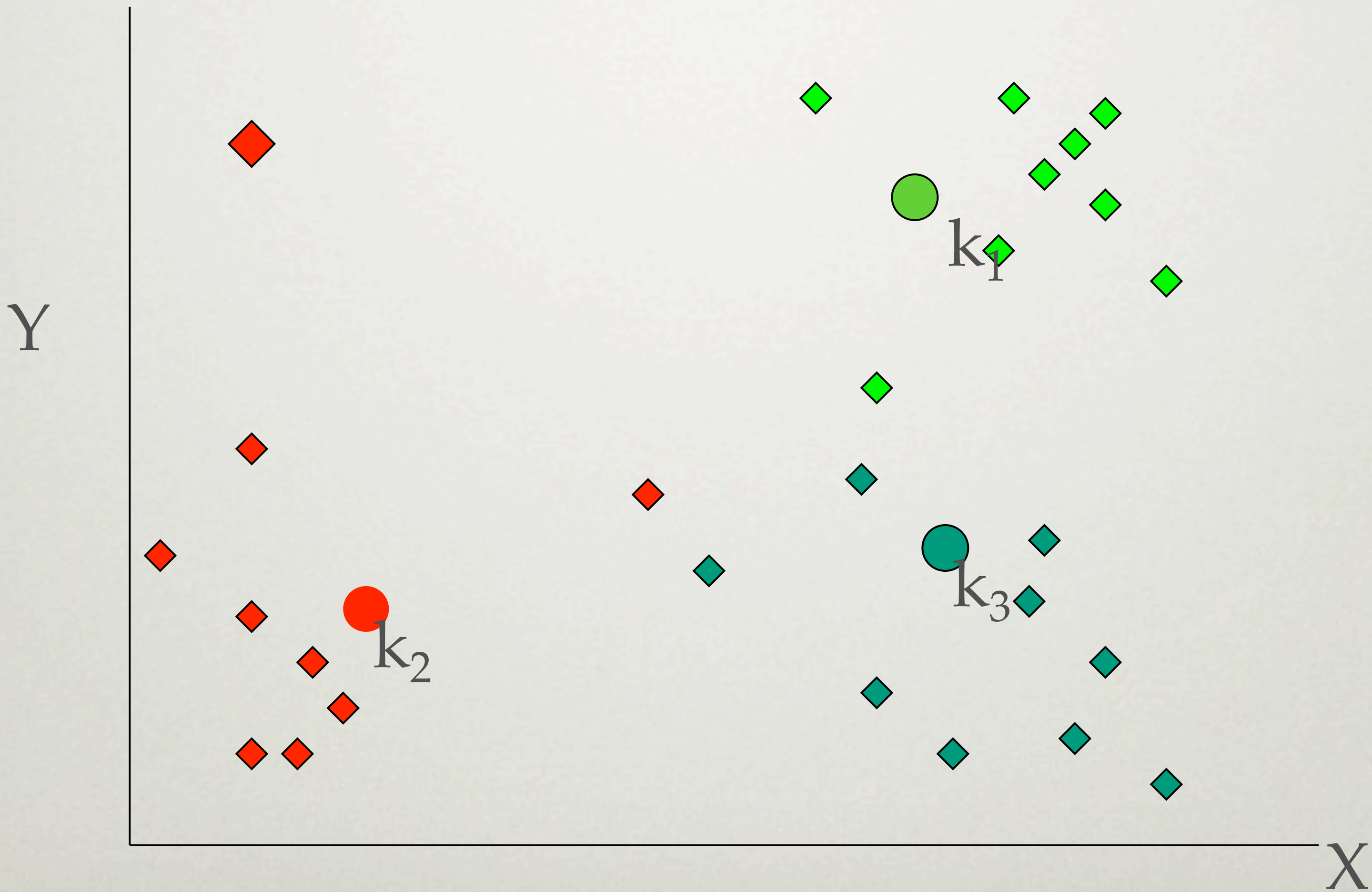
Reassign points to nearest cluster center

K-means clustering (k=3)



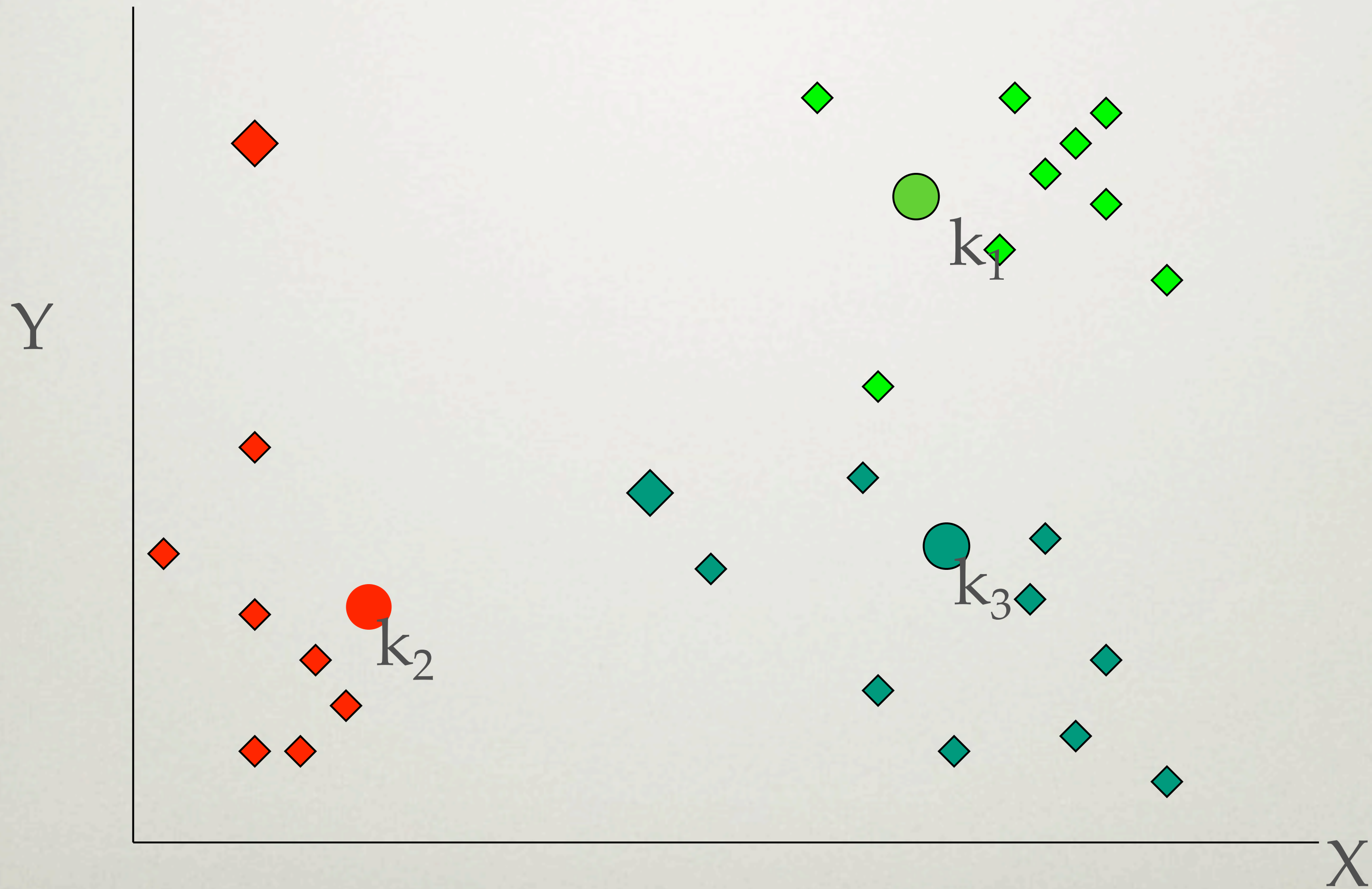
Reassign points to nearest cluster center

K-means clustering (k=3)



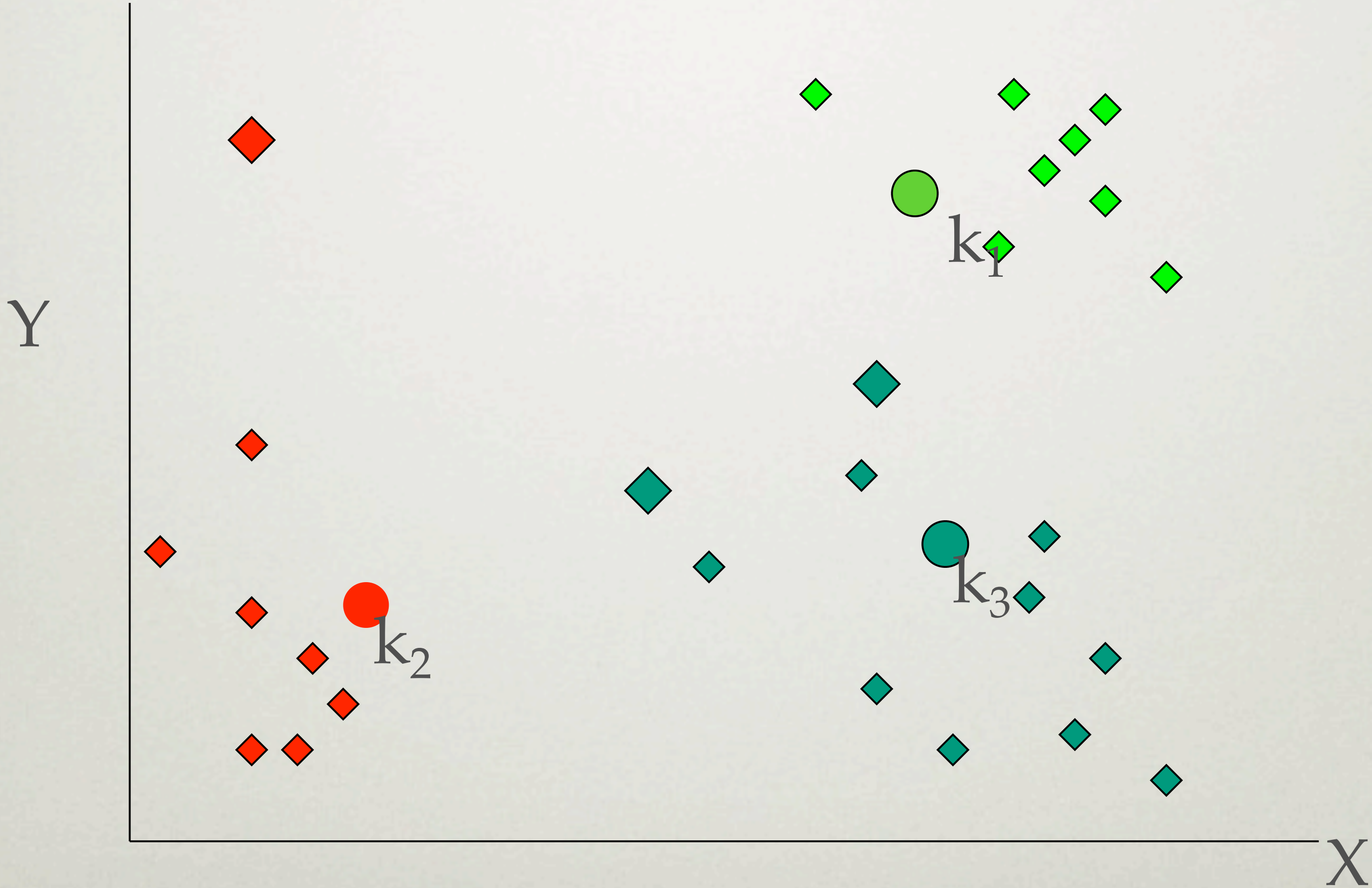
Reassign points to nearest cluster center

K-means clustering (k=3)



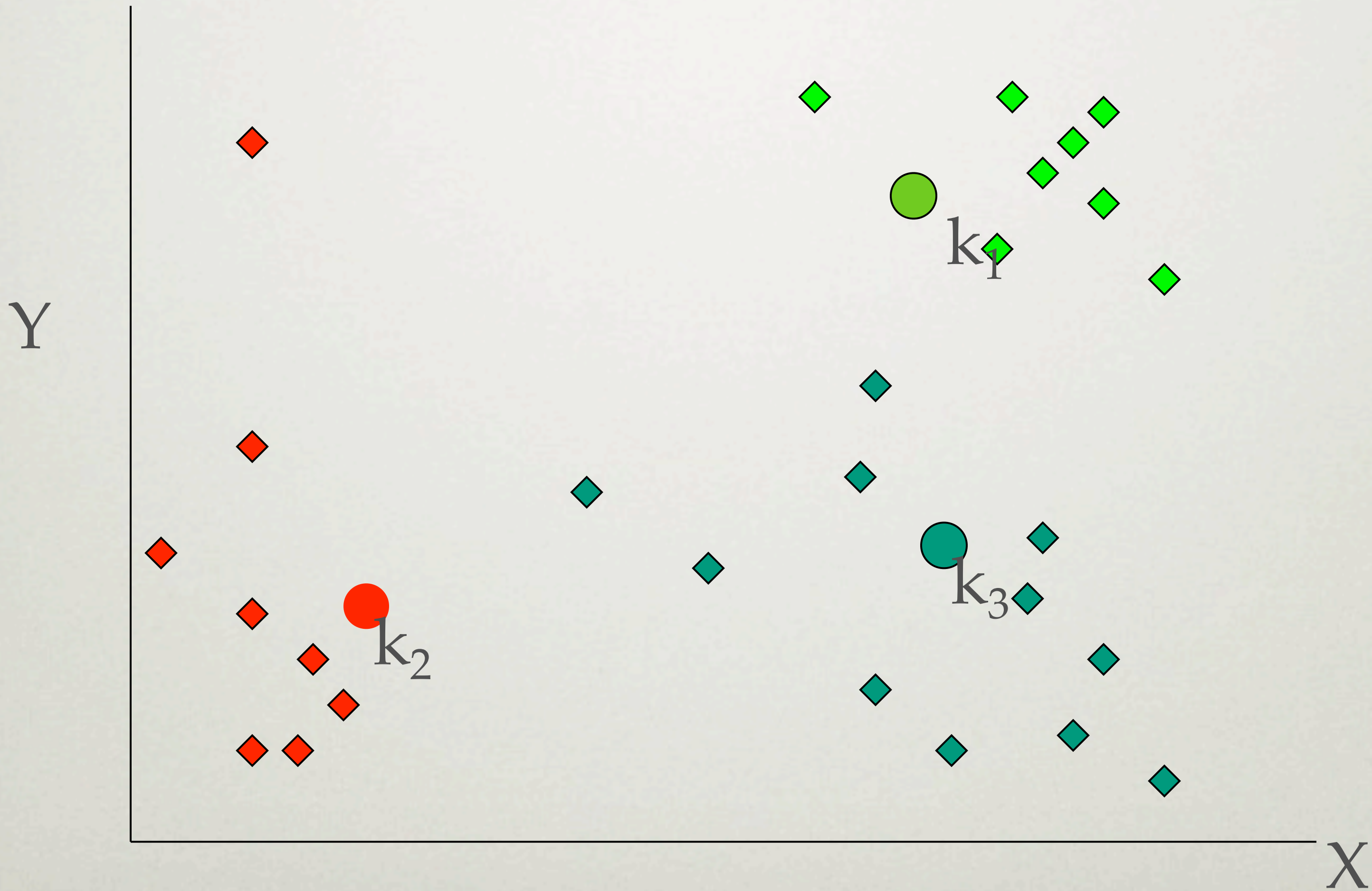
Reassign points to nearest cluster center

K-means clustering (k=3)



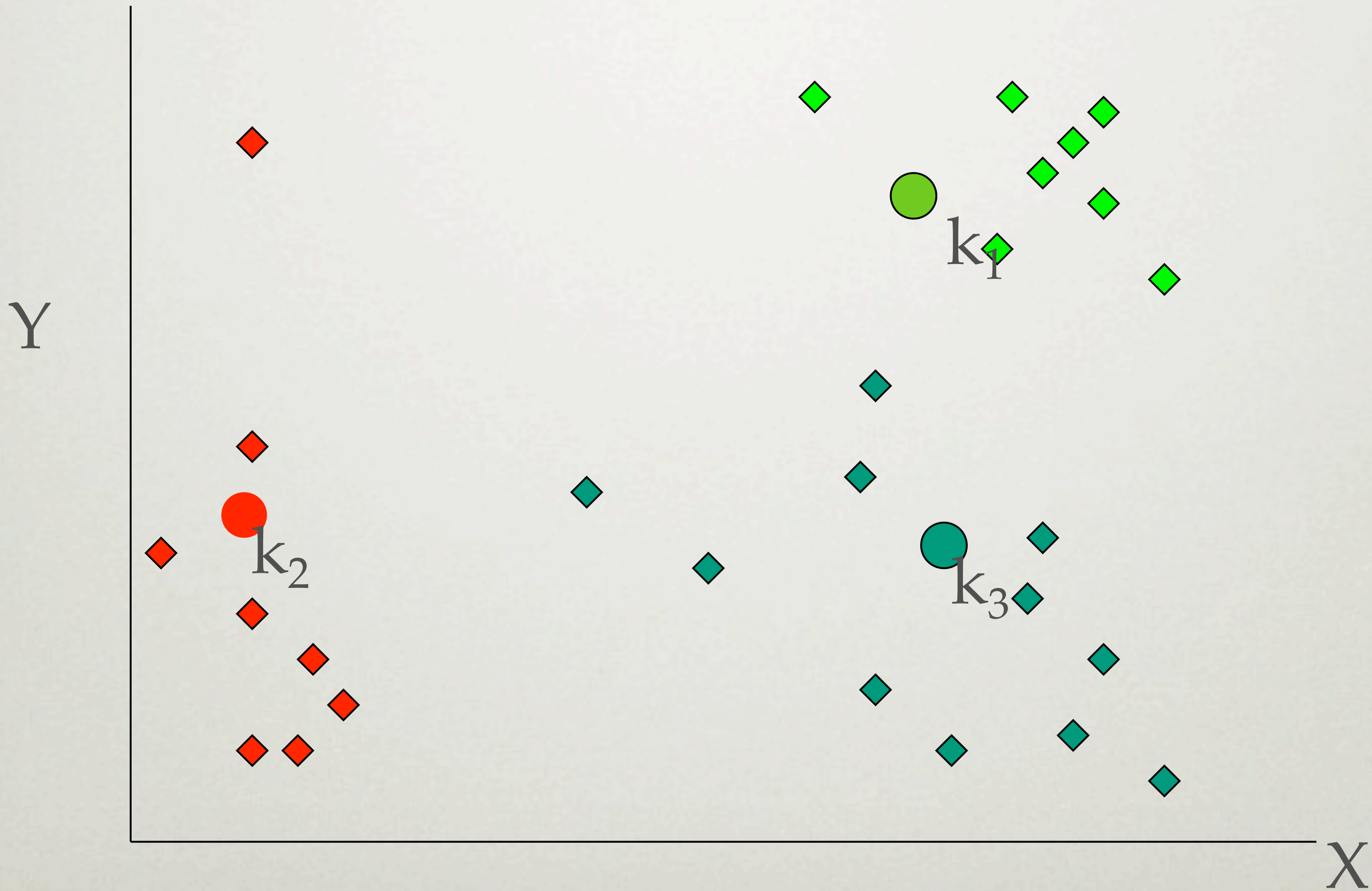
Reassign points to nearest cluster center

K-means clustering (k=3)



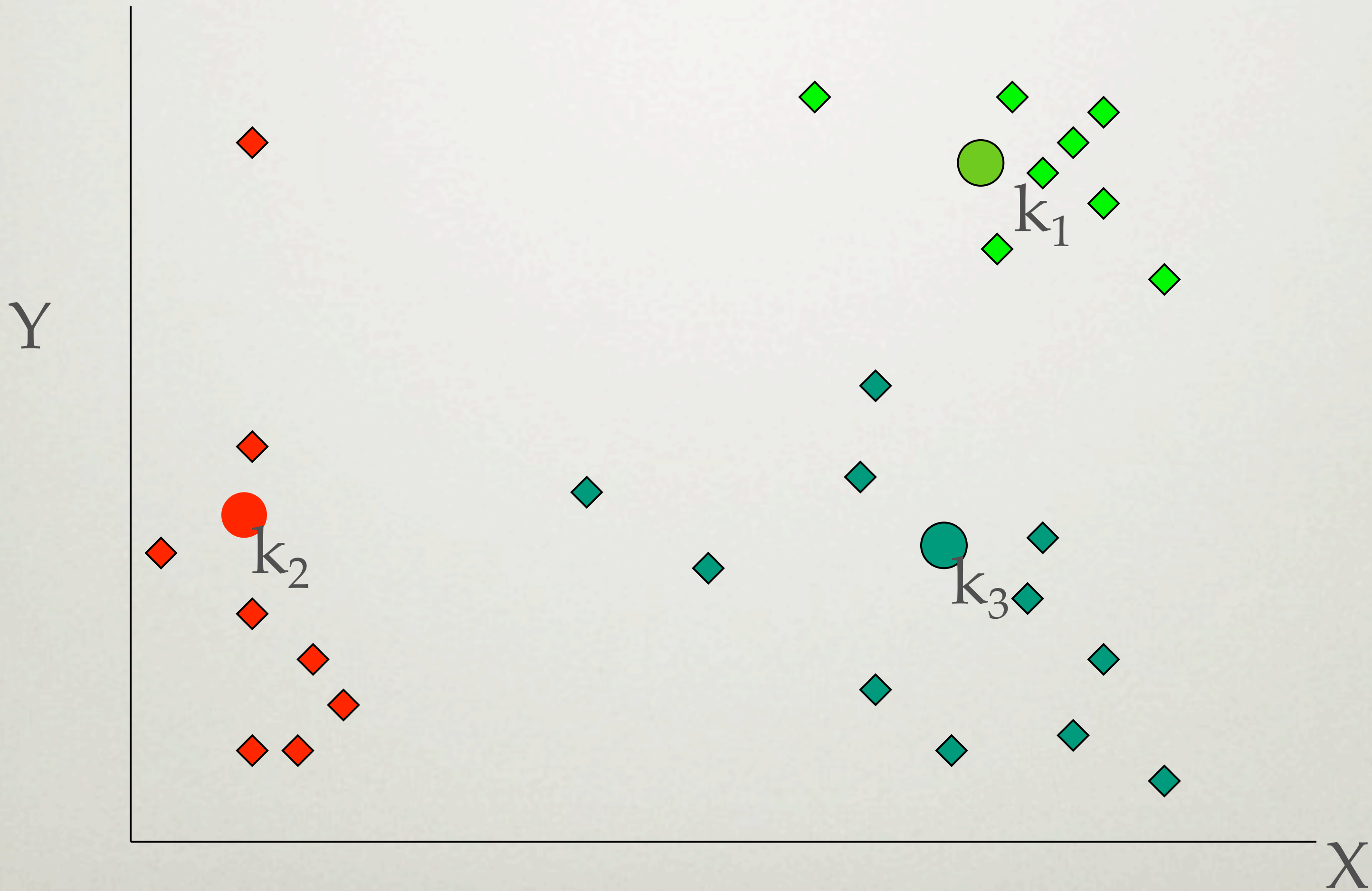
Repeat step 3-4 until cluster centers converge (don't/hardly move)

K-means clustering (k=3)



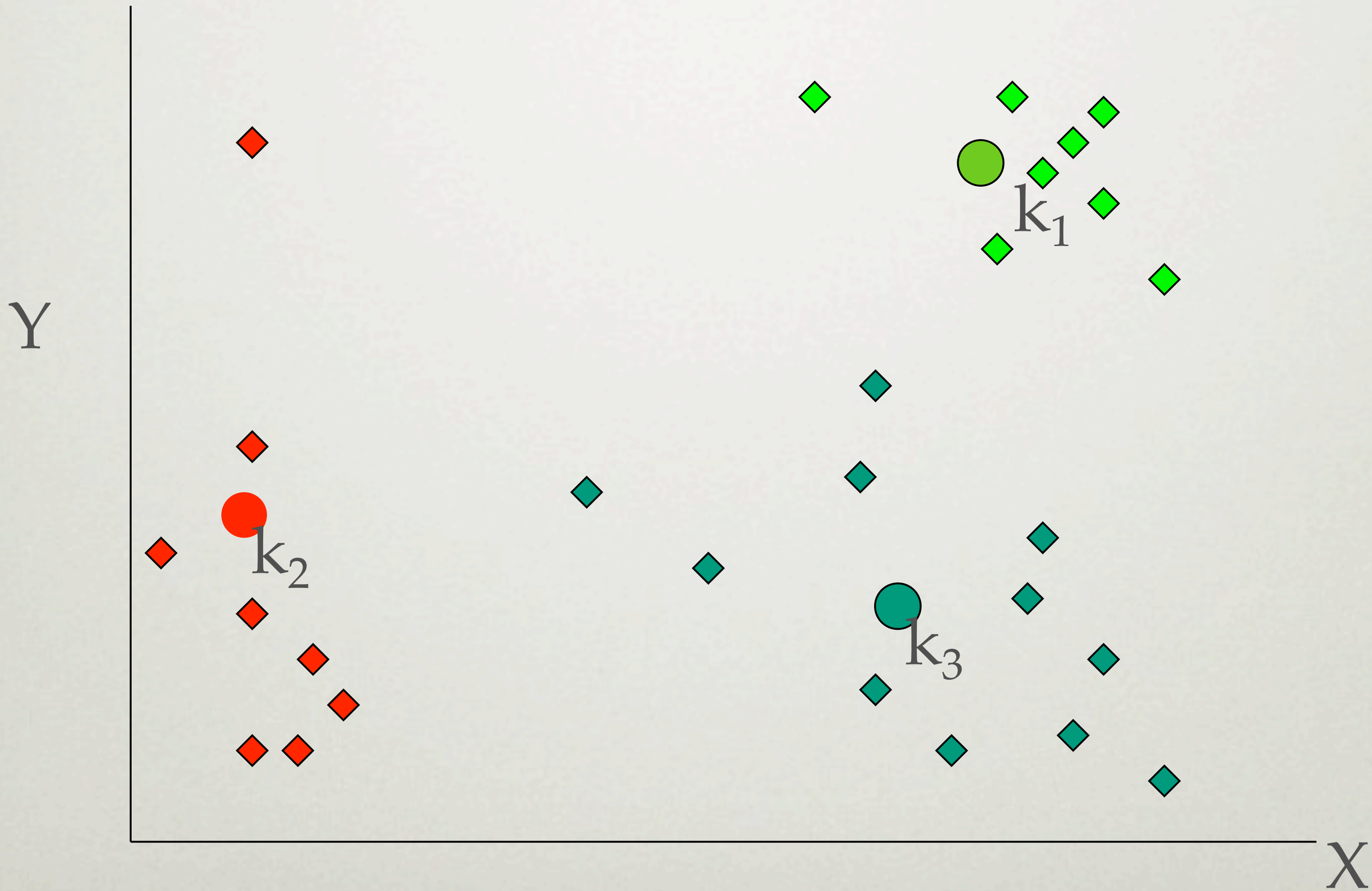
Repeat step 3-4 until cluster centers converge (don't/hardly move)

K-means clustering (k=3)



Repeat step 3-4 until cluster centers converge (don't/hardly move)

K-means clustering (k=3)



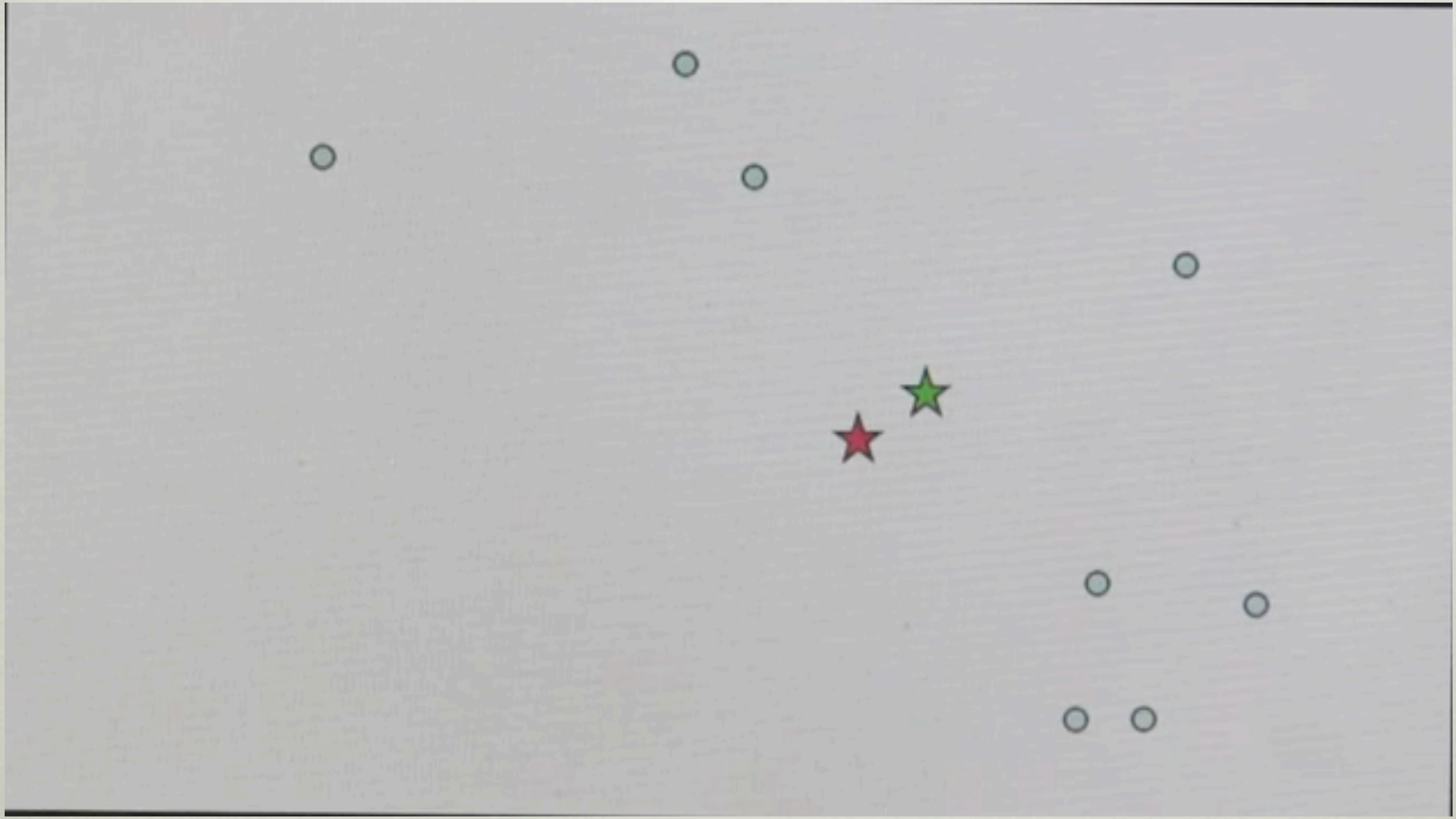
Repeat step 3-4 until cluster centers converge (don't/hardly move)

K-means

Works with numeric data only

1) Pick K random points: initial cluster centers
2) Assign every item to its nearest cluster center (e.g. using Euclidean distance)
3) Move each cluster center to the mean of its assigned items
4) Repeat steps 2,3 until convergence (change in cluster assignments less than a threshold)

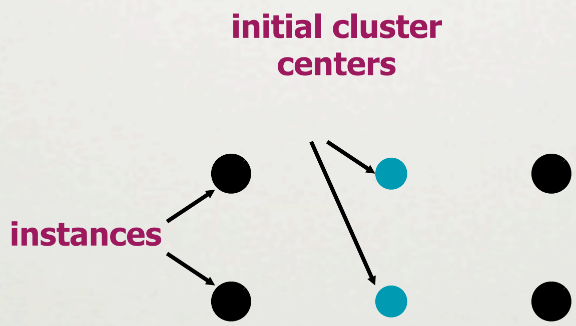
K-means clustering: another example



http://www.youtube.com/watch?v=zaKjh2N8jN4#!

Discussion

- Result can vary significantly depending on initial choice of centers
- Can get trapped in local minimum
    - Example:



- To increase chance of finding global optimum: restart with different random seeds

K-means clustering summary

Advantages

- Simple, understandable
- Items automatically assigned to clusters

Disadvantages

- Must pick number of clusters before hand
- All items forced into a single cluster
- Sensitive to outliers

K-means: variations

- K-medoids – instead of mean, use medians of each cluster
  - Mean of 1, 3, 5, 7, 1009 is
  - Median of 1, 3, 5, 7, 1009 is
- For large databases, use sampling

K-means: variations

- K-medoids – instead of mean, use medians of each cluster
  - Mean of 1, 3, 5, 7, 1009 is 205
  - Median of 1, 3, 5, 7, 1009 is
- For large databases, use sampling

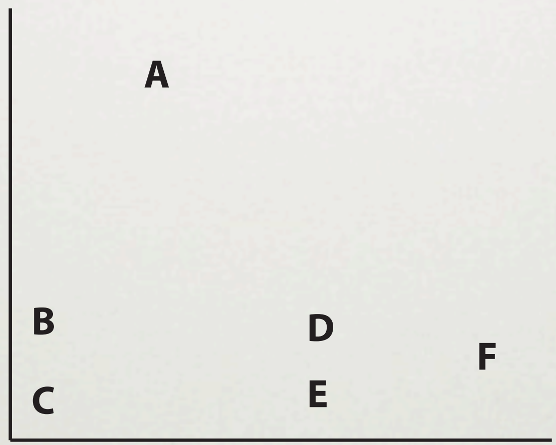
K-means: variations

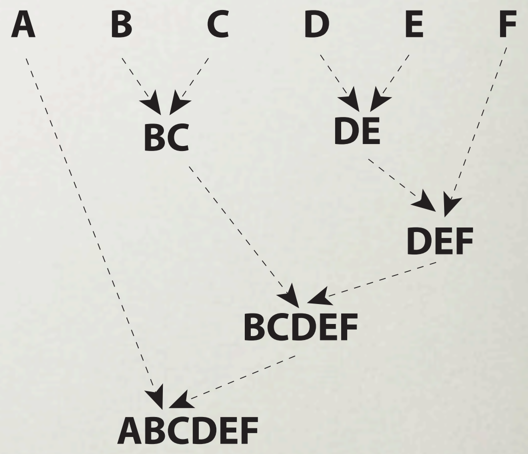
- K-medoids – instead of mean, use medians of each cluster
    - Mean of 1, 3, 5, 7, 1009 is 205
    - Median of 1, 3, 5, 7, 1009 is 5
- For large databases, use sampling

Hierarchical Clustering

Bottom-up vs top-down clustering

- Bottom up / Agglomerative
  - Start with single-instance clusters
  - At each step, join two "closest" clusters

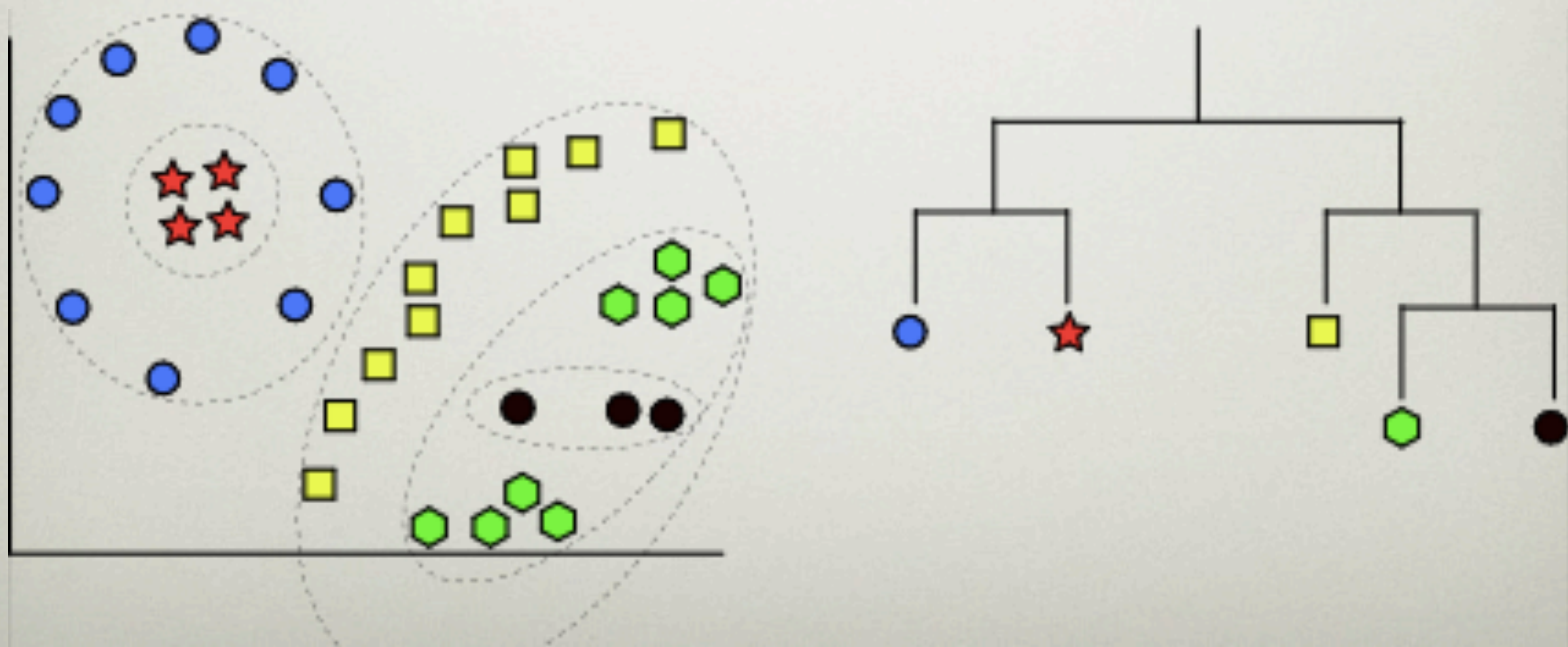




- Top down
  - Start with one universal cluster
  - Split in two clusters
  - Proceed recursively on each subset

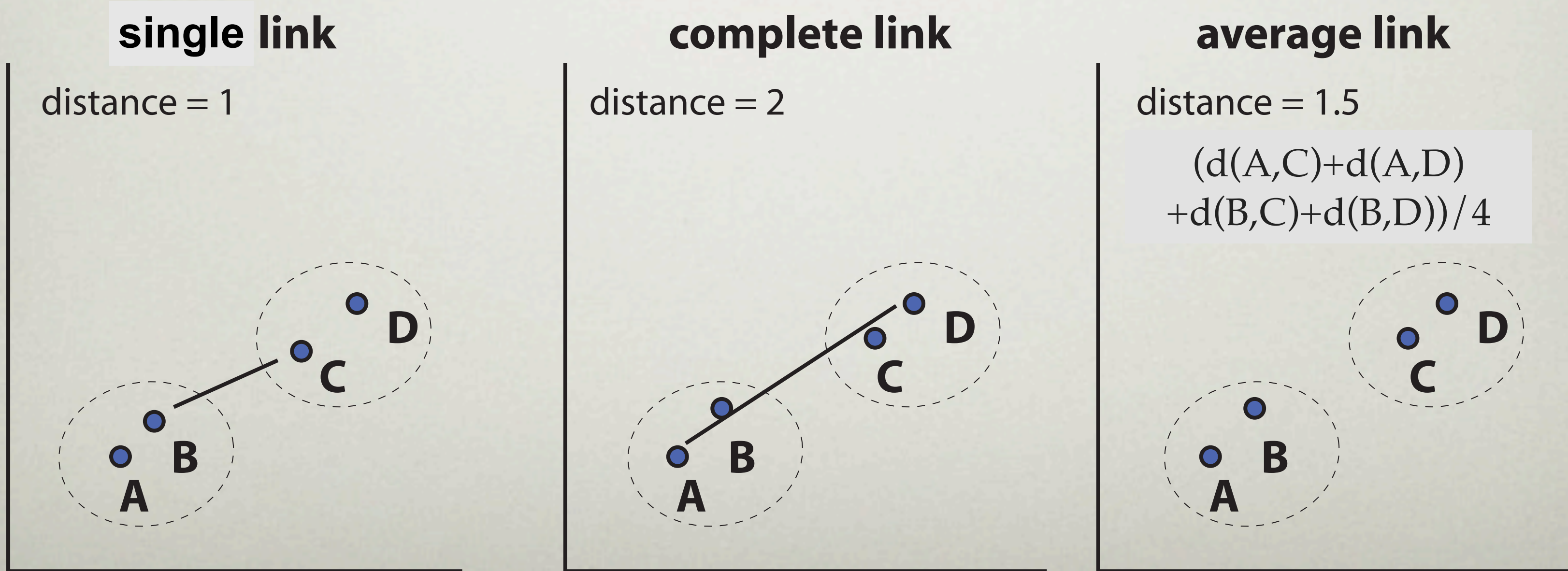
Hierarchical clustering

- Hierarchical clustering represented in *dendrogram*
  - tree structure containing hierarchical clusters
  - clusters in leafs, union of child clusters in nodes



Distance Between Clusters

- *Centroid:* distance between centroids
  - Sometimes hard to compute (e.g. mean of molecules?)
- *Single Link*: smallest distance between points
- *Complete Link:* largest distance between points
- *Average Link:* average distance between points



Distance Between Clusters

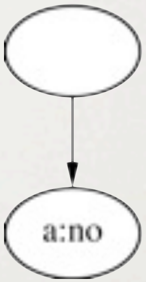
- ***Centroid:*** distance between centroids
  - Sometimes hard to compute (e.g. mean of molecules?)
- ***Single Link***: smallest distance between points
- ***Complete Link:*** largest distance between points
- ***Average Link:*** average distance between points
- ***Group-average***: group two clusters into one, then take average distance between all points *(incl. d(A,B) & d(C,D))*

Incremental Clustering

Clustering weather data

ID	Outlook	Temp.	Humidity	Windy
A	Sunny	Hot	High	False
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	Rainy	Mild	High	False
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	Overcast	Cool	Normal	True
H	Sunny	Mild	High	False
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

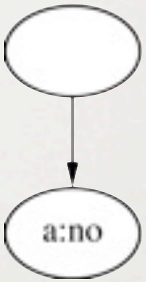
1



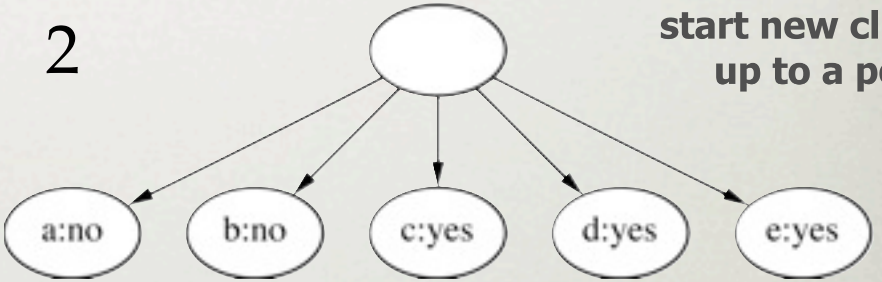
Clustering weather data

ID	Outlook	Temp.	Humidity	Windy
A	Sunny	Hot	High	False
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	Rainy	Mild	High	False
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	Overcast	Cool	Normal	True
H	Sunny	Mild	High	False
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

1



2



start new clusters, up to a point

Category Utility

- Category utility: overall quality of clustering
- Quadratic loss function
  - nominal: clusters C_i, attributes a_i, values v_{ij}:

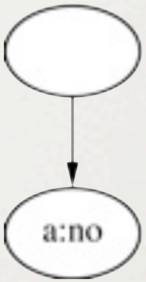
$$CU(C_1, C_2, \ldots, C_k) = \frac{\sum_l \Pr[C_l] \sum_i \sum_j (\Pr[a_i = v_{ij} \mid C_l]^2 - \Pr[a_i = v_{ij}]^2)}{k}$$

  - numeric: similar, assume Gaussian distribution
- Intuitively:
  - good clusters allow to predict value of new data points: $\Pr[a_i{=}v_{ij} \mid C_i] > \Pr[a_i{=}v_{ij}]$
  - 1/k factor: penalty for using many clusters (avoids overfitting)

Clustering weather data

ID	Outlook	Temp.	Humidity	Windy
A	Sunny	Hot	High	False
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	Rainy	Mild	High	False
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	Overcast	Cool	Normal	True
H	Sunny	Mild	High	False
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

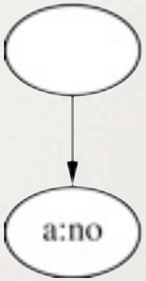
1



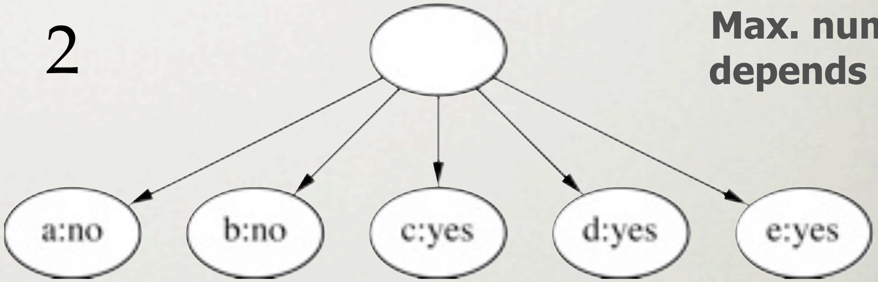
Clustering weather data

ID	Outlook	Temp.	Humidity	Windy
A	Sunny	Hot	High	False
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	Rainy	Mild	High	False
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	Overcast	Cool	Normal	True
H	Sunny	Mild	High	False
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

1



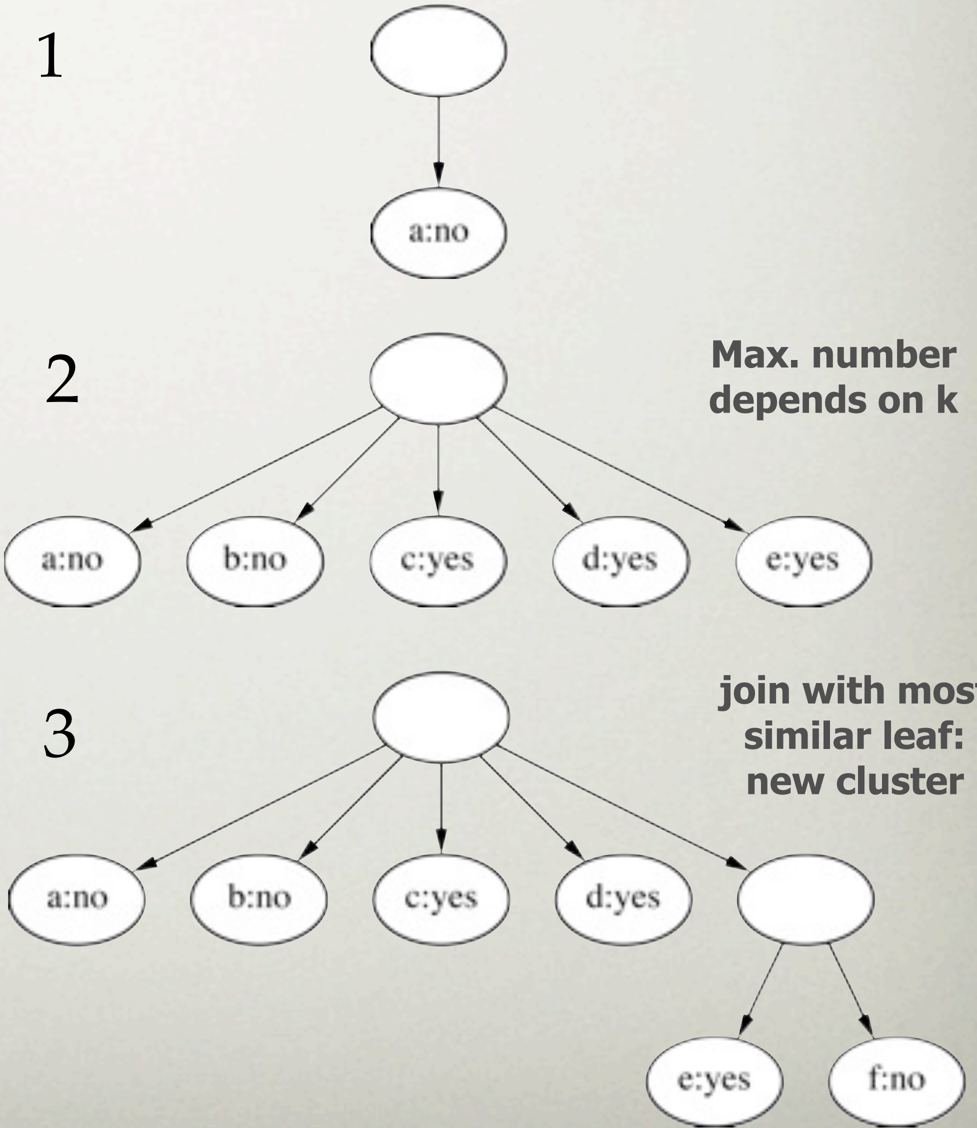
2



Max. number depends on k

Clustering weather data

ID	Outlook	Temp.	Humidity	Windy
A	Sunny	Hot	High	False
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	Rainy	Mild	High	False
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	Overcast	Cool	Normal	True
H	Sunny	Mild	High	False
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True



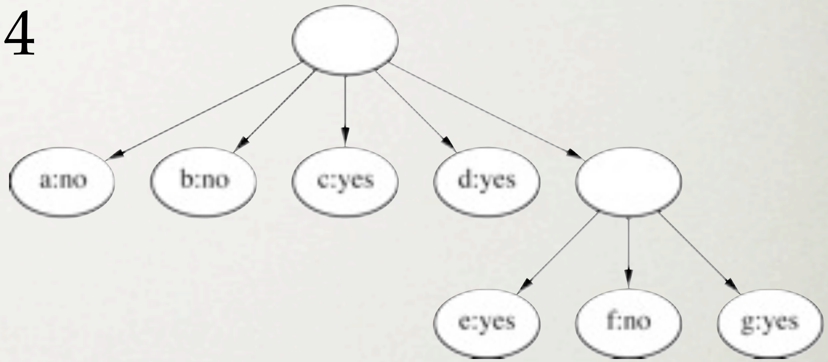
Max. number depends on k

join with most similar leaf: new cluster

Clustering weather data

ID	Outlook	Temp.	Humidity	Windy
A	*Sunny*	*Hot*	*High*	*False*
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	*Rainy*	*Mild*	*High*	*False*
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	**Overcast**	**Cool**	**Normal**	**True**
H	*Sunny*	*Mild*	*High*	*False*
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

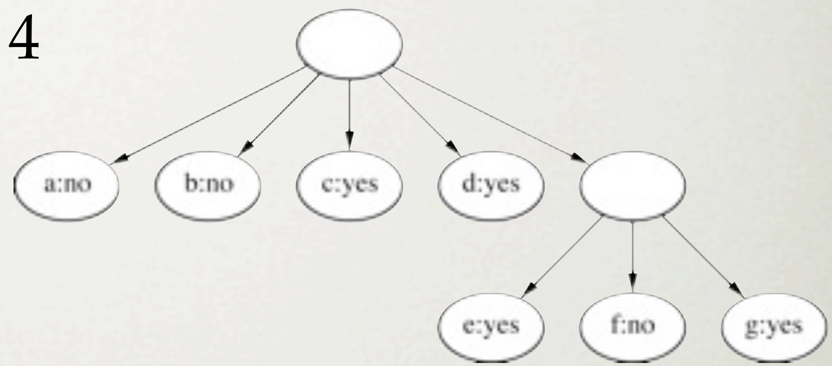
4



Clustering weather data

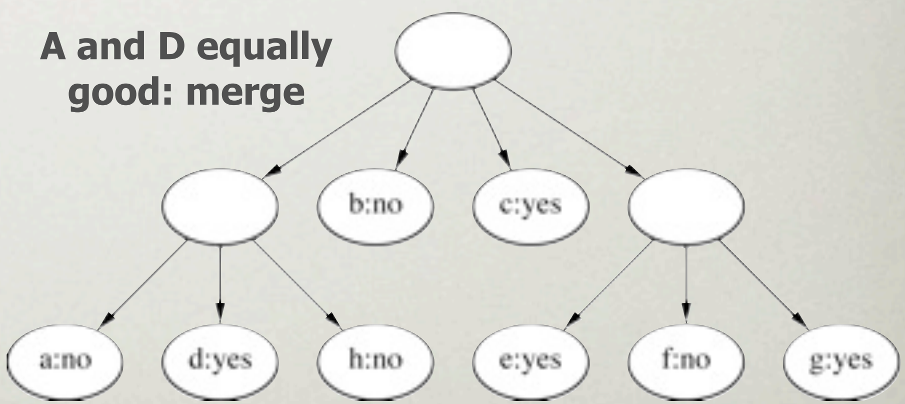
ID	Outlook	Temp.	Humidity	Windy
A	*Sunny*	*Hot*	*High*	*False*
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	*Rainy*	*Mild*	*High*	*False*
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	**Overcast**	**Cool**	**Normal**	**True**
H	*Sunny*	*Mild*	*High*	*False*
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

4



5

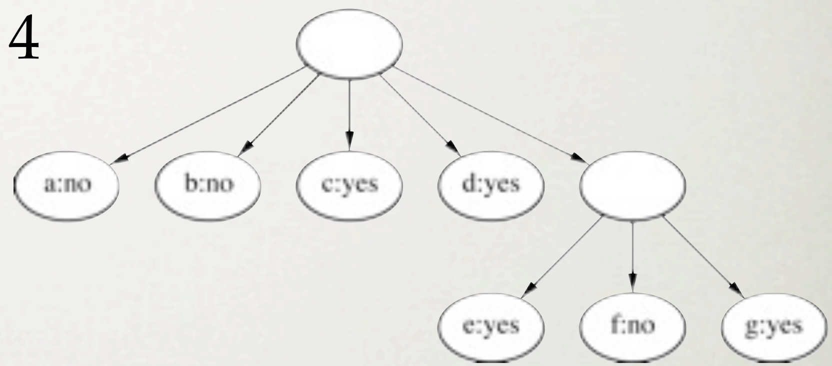
A and D equally good: merge



Clustering weather data

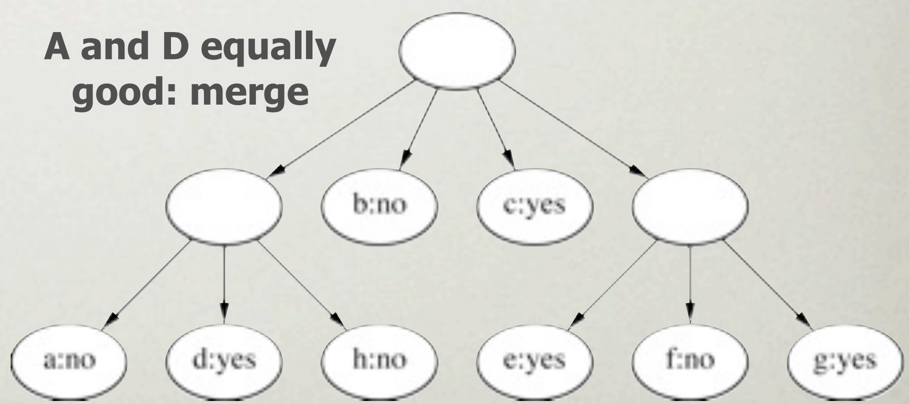
ID	Outlook	Temp.	Humidity	Windy
A	*Sunny*	*Hot*	*High*	*False*
B	Sunny	Hot	High	True
C	Overcast	Hot	High	False
D	*Rainy*	*Mild*	*High*	*False*
E	**Rainy**	**Cool**	**Normal**	**False**
F	**Rainy**	**Cool**	**Normal**	**True**
G	**Overcast**	**Cool**	**Normal**	**True**
H	*Sunny*	*Mild*	*High*	*False*
I	Sunny	Cool	Normal	False
J	Rainy	Mild	Normal	False
K	Sunny	Mild	Normal	True
L	Overcast	Mild	High	True
M	Overcast	Hot	Normal	False
N	Rainy	Mild	High	True

4



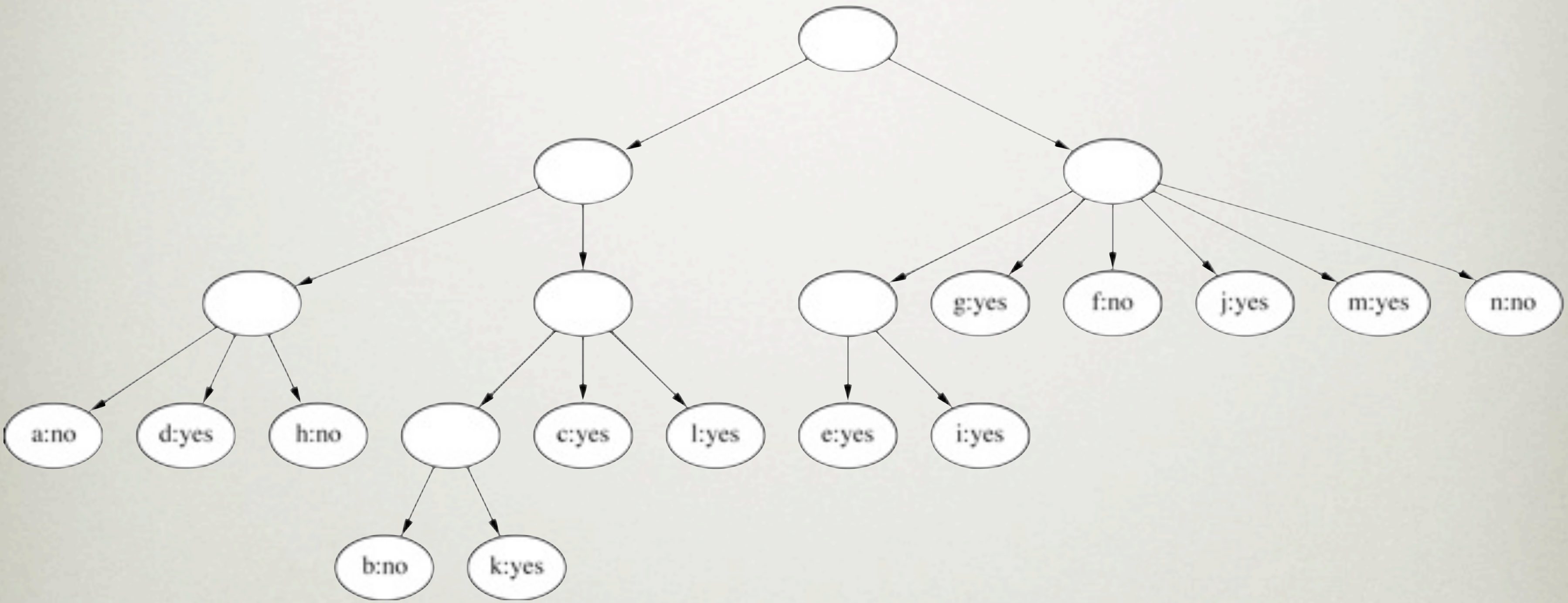
5

A and D equally good: merge



Consider splitting the best host if merging doesn't help

Final hierarchy



ID	Outlook	Temp.	Humidity	Windy
A	**Sunny**	**Hot**	**High**	**False**
B	**Sunny**	**Hot**	**High**	**True**
C	Overcast	Hot	High	False
D	Rainy	Mild	High	False

Note that a and b are actually very similar, but end up in different clusters

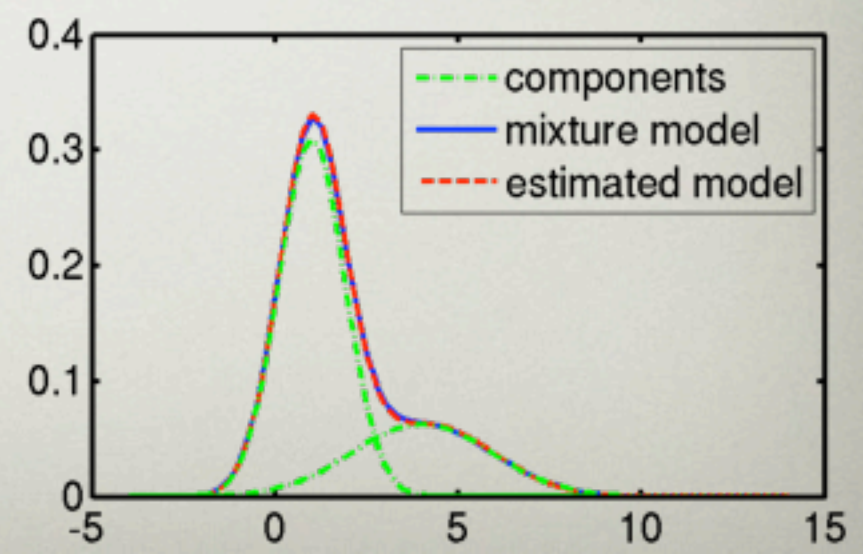
Incremental clustering

- For large, regularly updated databases
  - start with tree and empty root node
  - add instances one by one
  - update tree appropriately at each stage
    - form new leaf
    - join instance with most similar leaf: new node (cluster)
    - *merge* existing leafs (move down one level)
    - *split* node into leafs (move up one level)
- Best decision: *category utility*

Probability-based Clustering

Probability-based Clustering

- Given *k* clusters, each instance belongs to *all* clusters, with a certain probability
  - *mixture model: set of k distributions (one per cluster)*
  - *also: each cluster has prior likelihood*
- If correct clustering known, we know parameters μ, σ and $P(C_i)$ for each cluster: calculate $P(C_i|x)$ using Bayes' rule
- Estimate the unknown parameters
  - How?



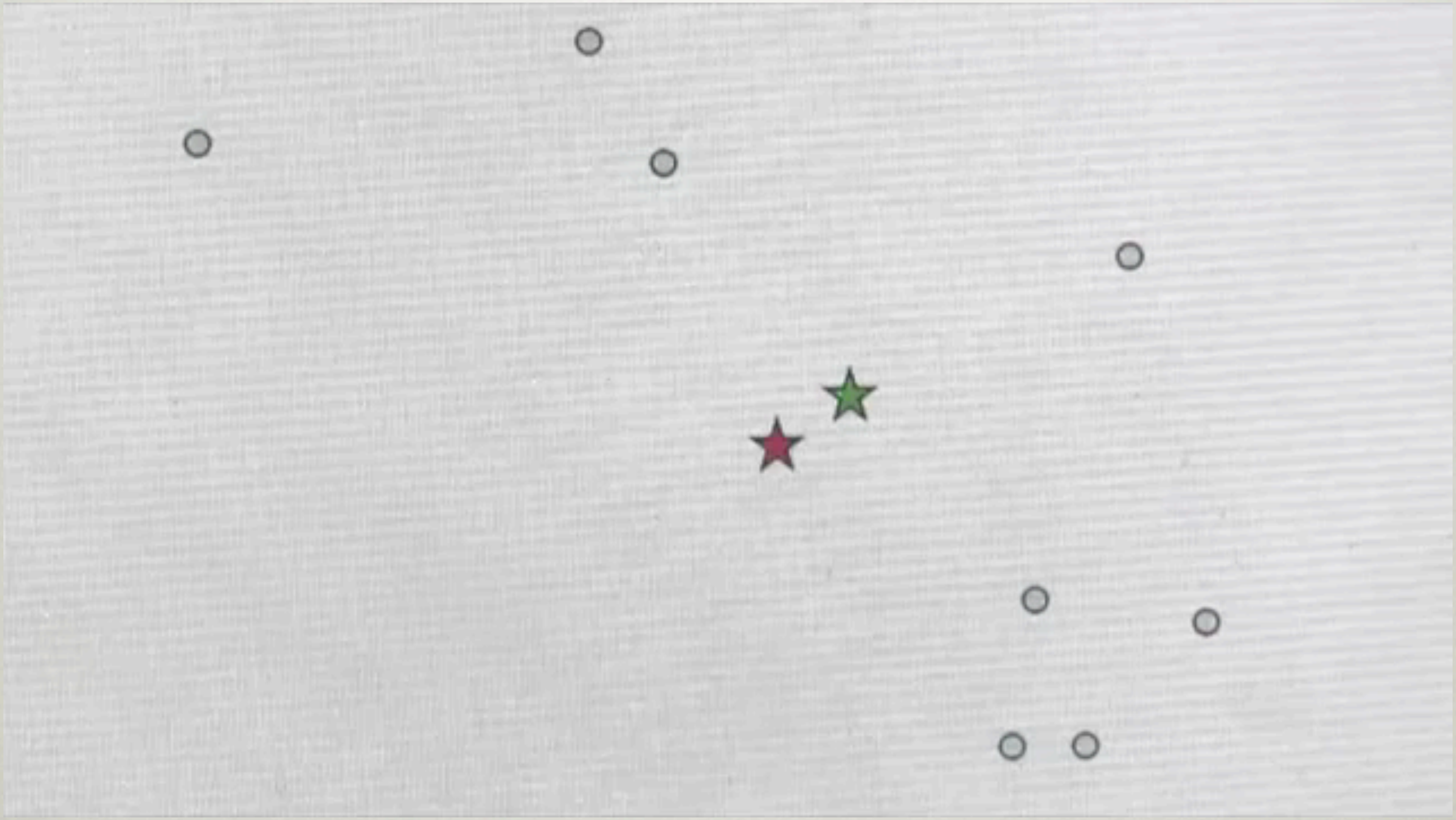
EM
Expectation Maximization

- Finds the parameters μ, σ for the distributions and the cluster membership

$$\mu_A = \frac{w_1 x_1 + \ldots + w_n x_n}{w_1 + \ldots + w_n}$$

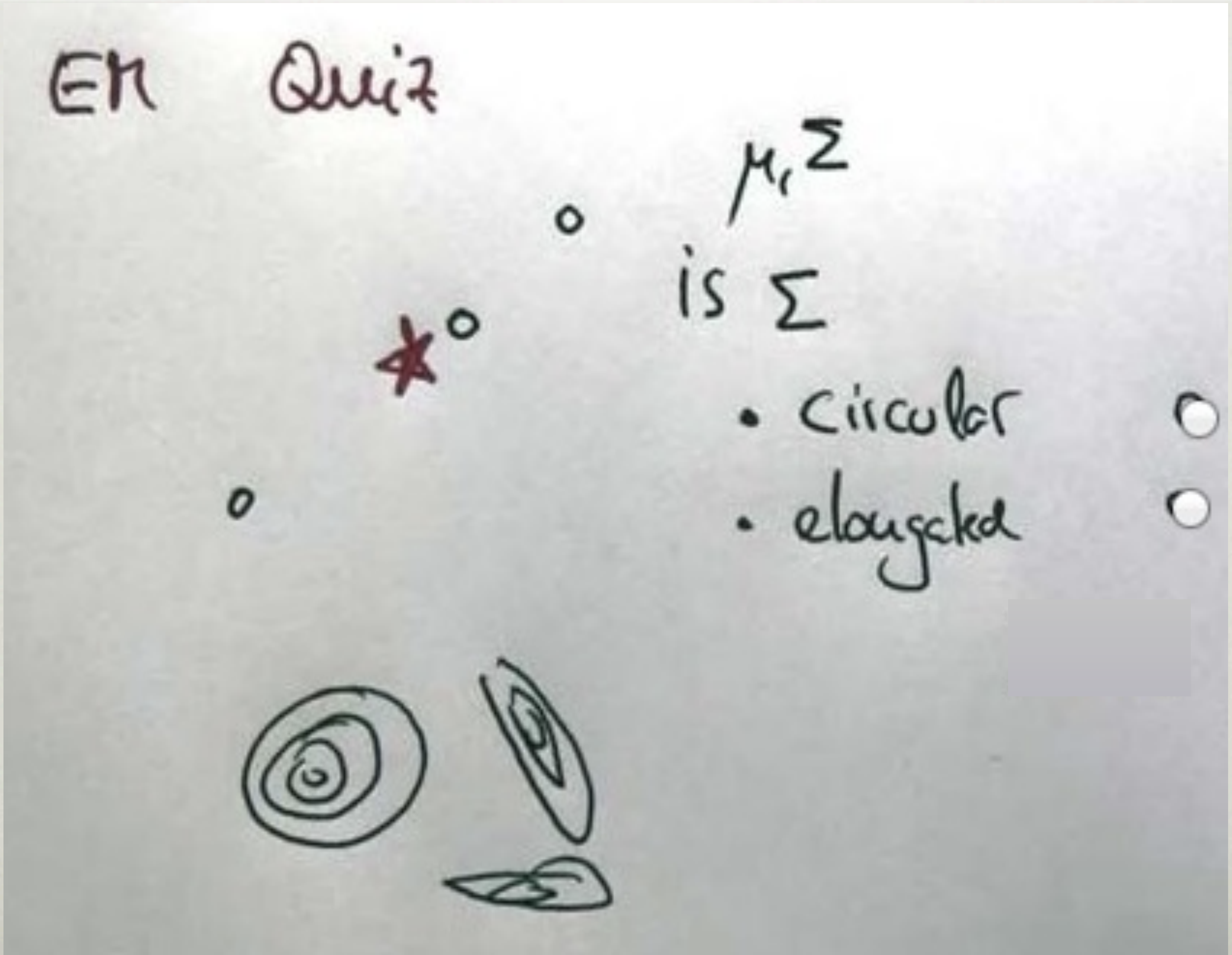
- (Random) initialization
    - Initial parameters μ, σ, $P(C_i)$ for each cluster
- Iterative algorithm:
    - Expectation step: with current parameters, calculate $P(C|x)$
    - Maximization step: update parameters using $P(C|x)$: new μ, σ, $P(C_i)$
- Iterate until converged to local optimum

EM vs K-means

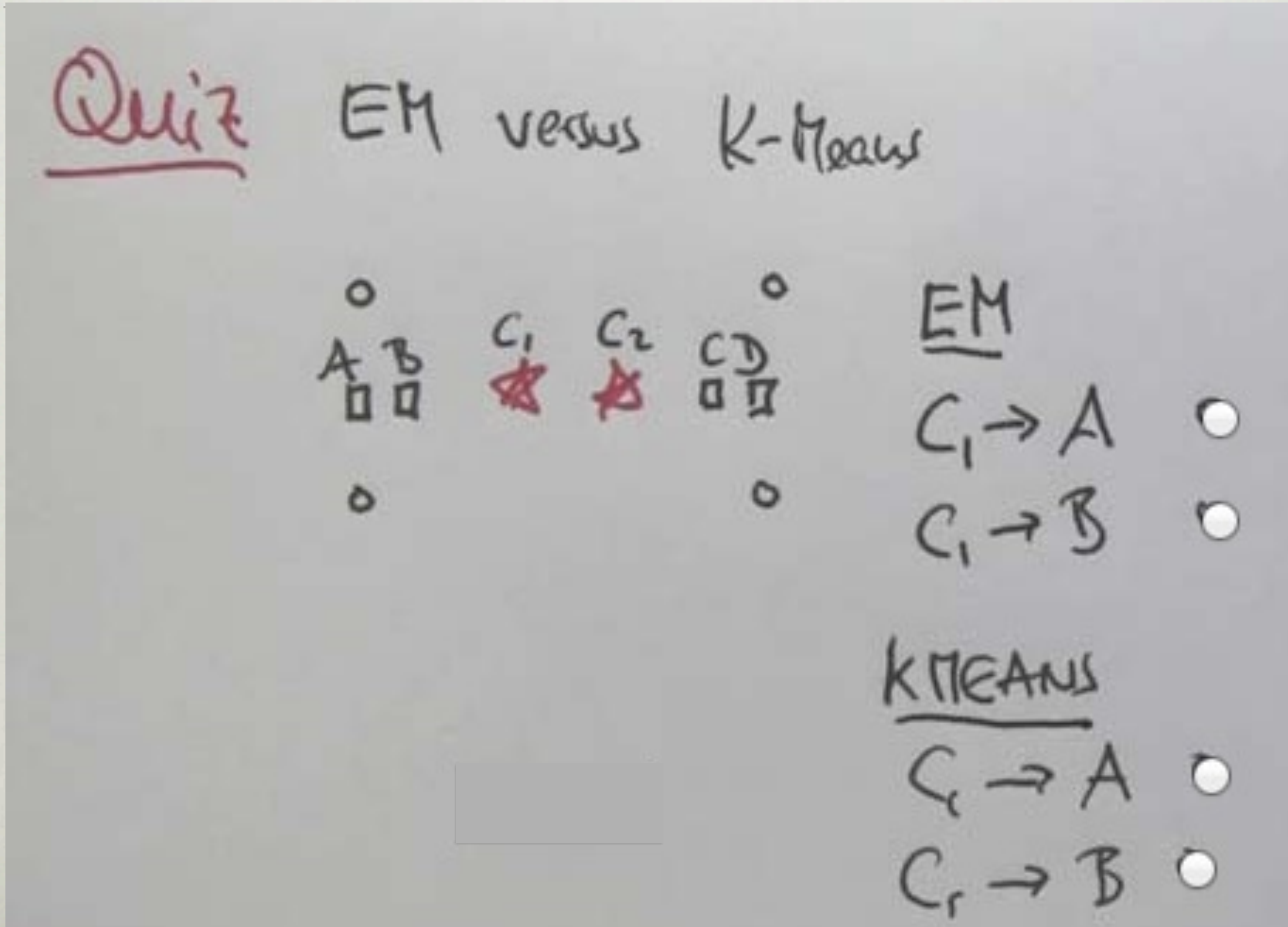


http://www.youtube.com/watch?v=1CWDWmF0i2s

Quiz



Quiz



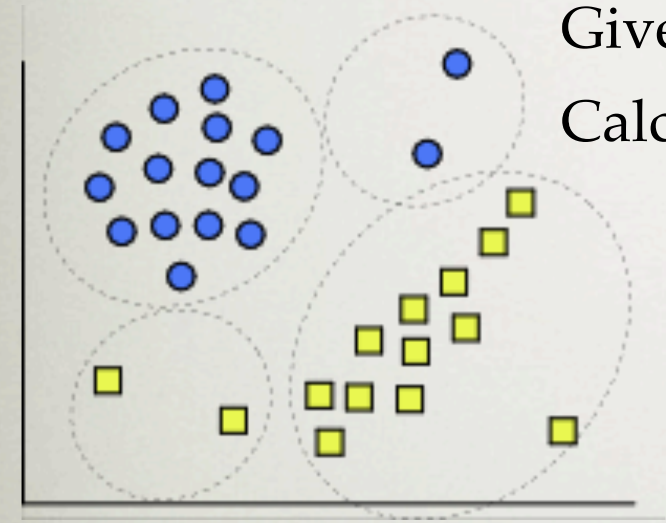
Clustering Evaluation

- Manual inspection
- Benchmarking on existing labels
- Cluster quality measures
    - distance measures
    - high similarity within a cluster, low across clusters

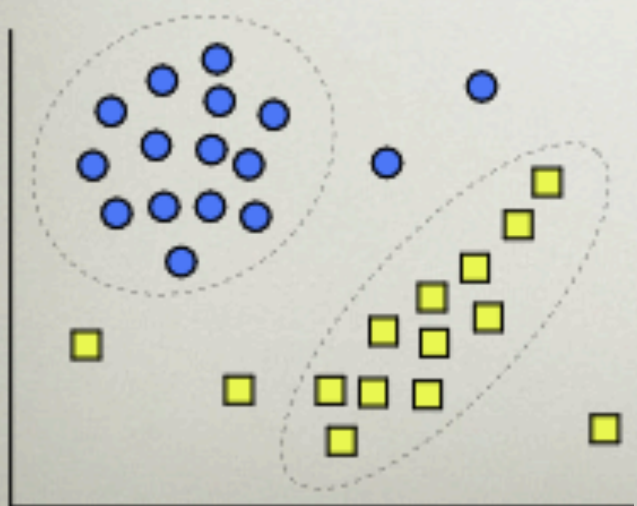
Goodness of Fit

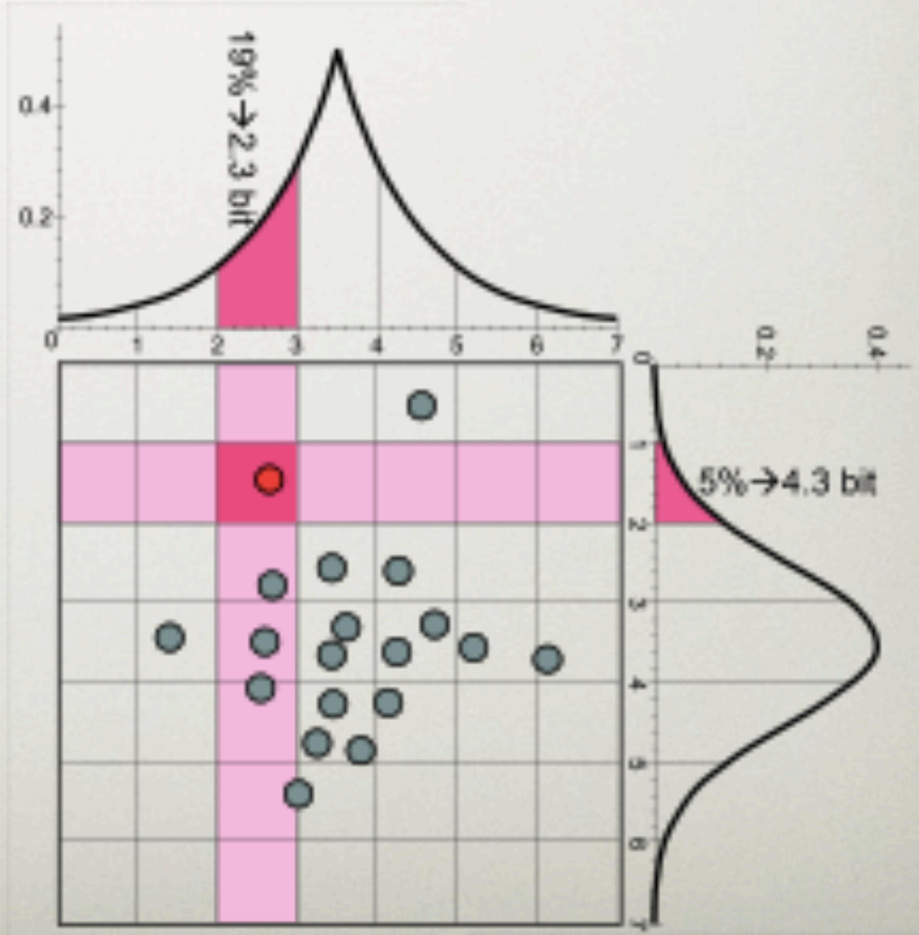
Given a function that defines the cluster

Calculate for each point how well it fits the cluster



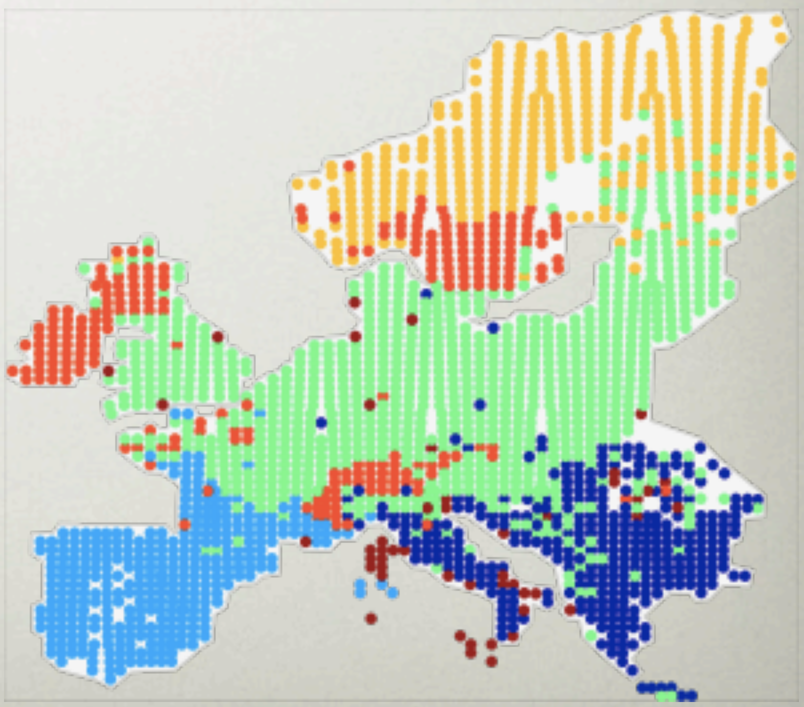
…, or better?





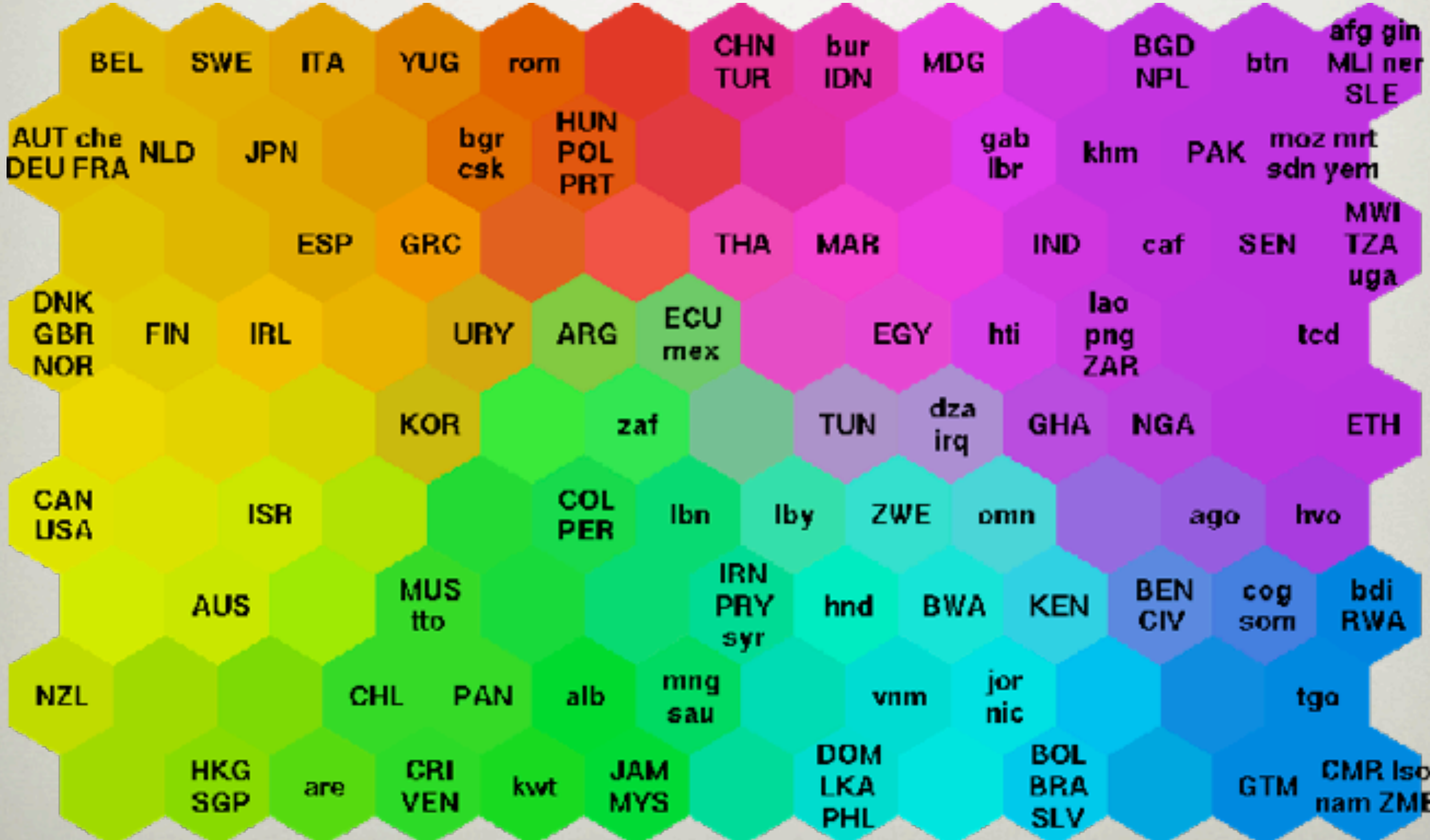
How to choose k?

- One important parameter *k*, but how to choose?
  - Domain dependent, we simply want *k* clusters
- Alternative: repeat for several values of *k* and choose the best
- Example:
  - cluster mammal properties
  - *k* different clusters
  - Use an MDL based encoding
  - Alternative to category theory
  - Each additional cluster introduces a penalty
  - Optimal for k=6

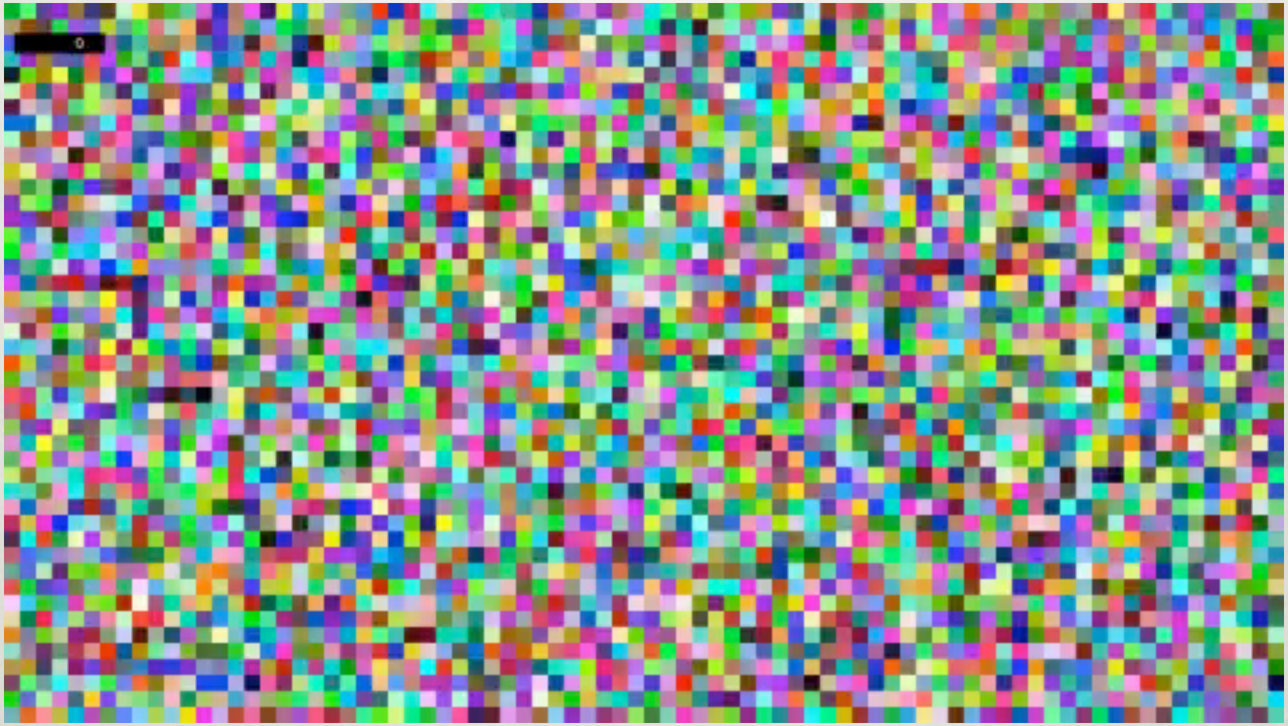


Self-Organizing Maps

Self Organizing Map



Self Organizing Map



http://www.youtube.com/watch?v=71wmOT4lHWc

Self Organizing Map

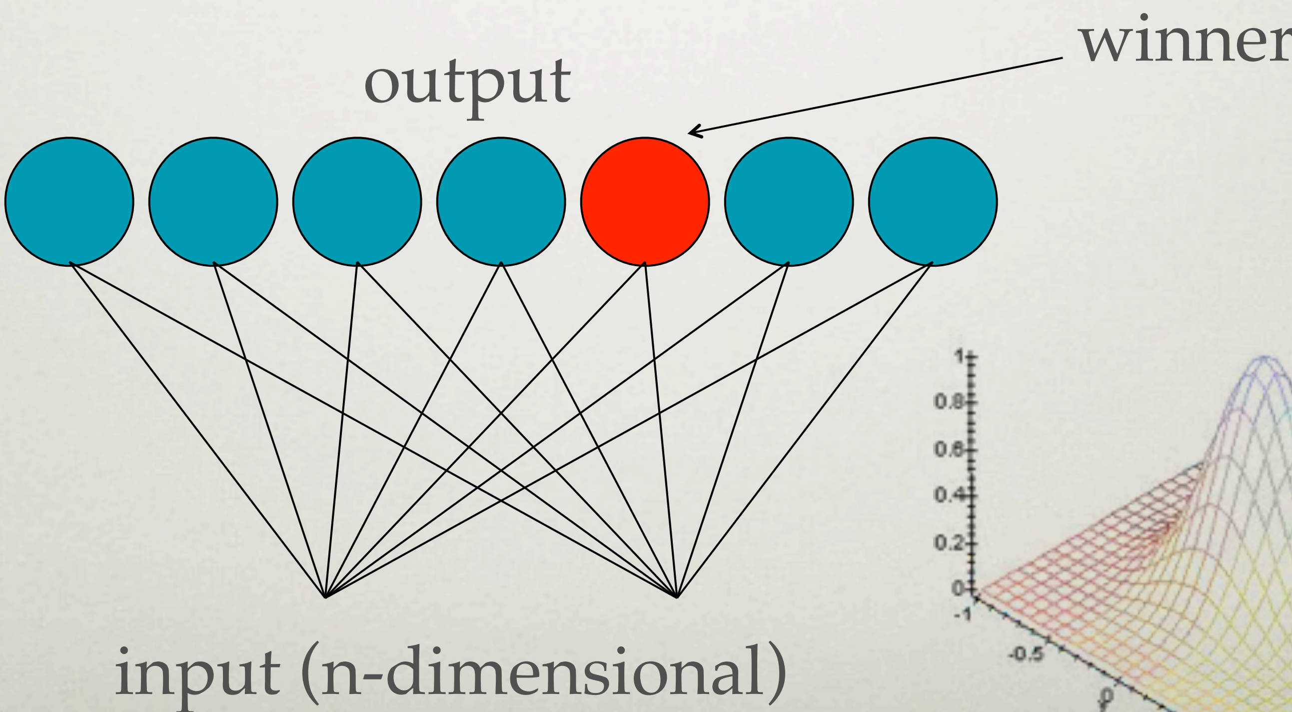
- Applications
  - Group similar data together
  - Dimensionality reduction
  - Data visualization technique
- Similar to neural networks
  - Neurons try to mimic the input vectors
  - The winning neuron (and its neighborhood) wins
  - Topology preserving, using Neighborhood function

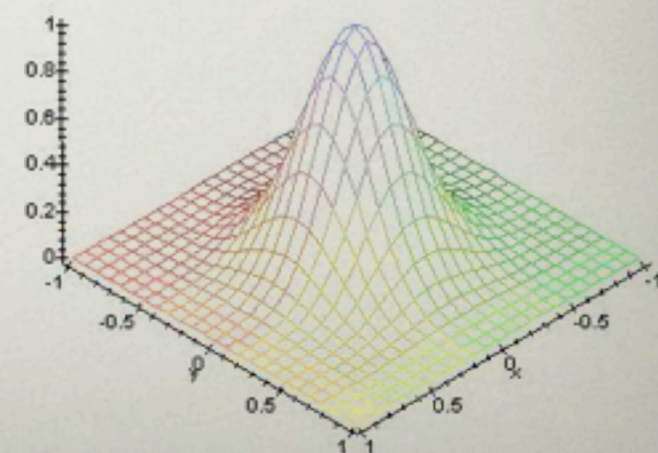
SOM Learning Algorithm

- Initialize SOM (random, or such that dissimilar input is mapped far apart)
- For t from 0 to N
  - Randomly select a training instance
  - Get the best matching neuron
    - calculate distance, e.g. $\sqrt{\sum_{i=0}^{n}(x_i - w_i)^2}$
  - Scale neighbors
    - Who? decrease over time: Hexagons, squares, gaussian, …
    - Update of neighbours towards the training instance

Self Organizing Map

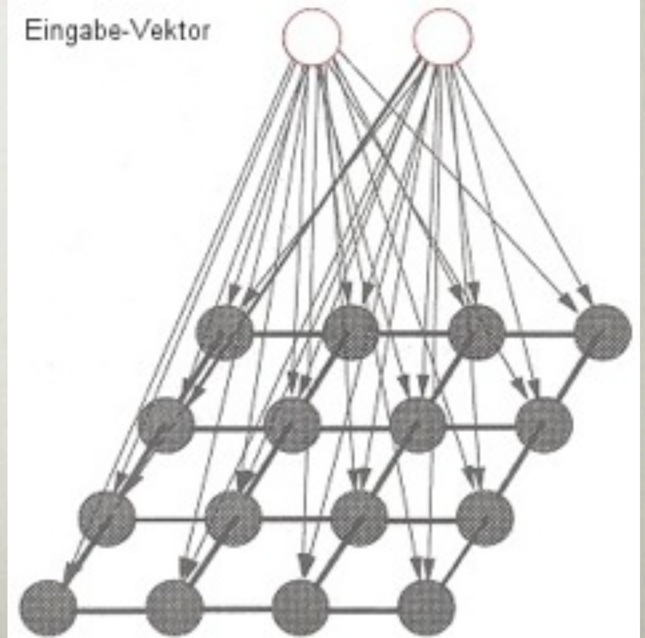
- Neighborhood function to preserve topological properties of the input space
- Neighbors share the prize (but slightly less)





Self Organizing Map

- Input: uniformly randomly distributed points
- Output: Map of 20^2 neurons
- Training: Starting with a large learning rate and neighborhood size, both are gradually decreased to facilitate convergence
- After learning, neurons with similar weights tend to cluster on the map



Discussion

- Can interpret clusters by using supervised learning
  - learn a classifier based on clusters
- Decrease dependence between attributes?
  - pre-processing step
  - E.g. use *principal component analysis*
- Can be used to fill in missing values
- Key advantage of probabilistic clustering:
  - Can estimate likelihood of data
  - Use it to compare different models objectively