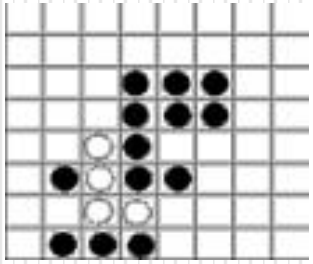
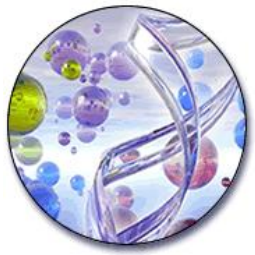




Welcome to

Introduction to Machine Learning!



Coffee Time

Discussion: Microblog User Profiling

Topic 8. Unsupervised learning

Min Zhang

z-m@tsinghua.edu.cn

Introduction to unsupervised learning

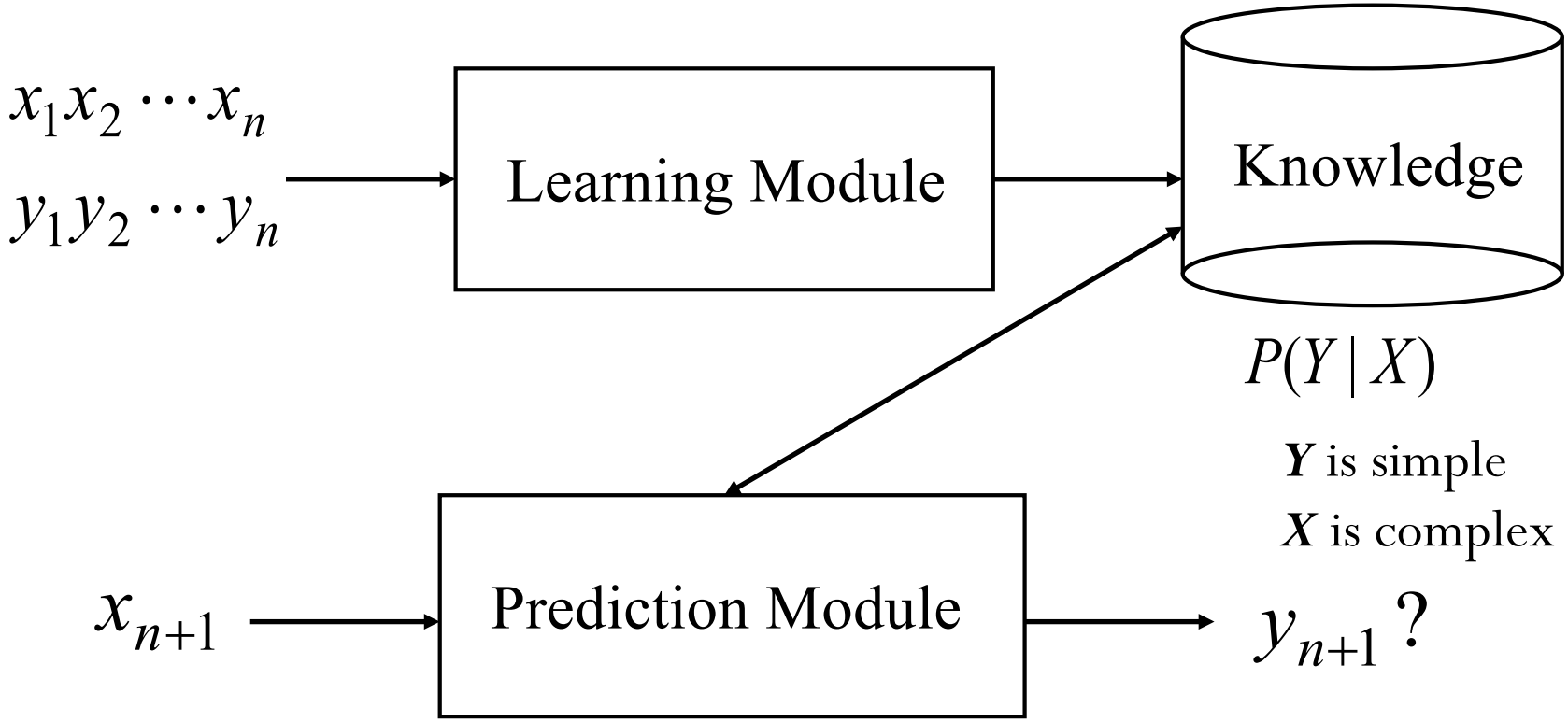
Introduction to unsupervised learning

- Categorization of machine learning algorithms
  - Eager vs. Lazy
  - Parametric vs. Non-parametric
  - Supervised vs. Unsupervised vs. Semi-supervised
  - ……

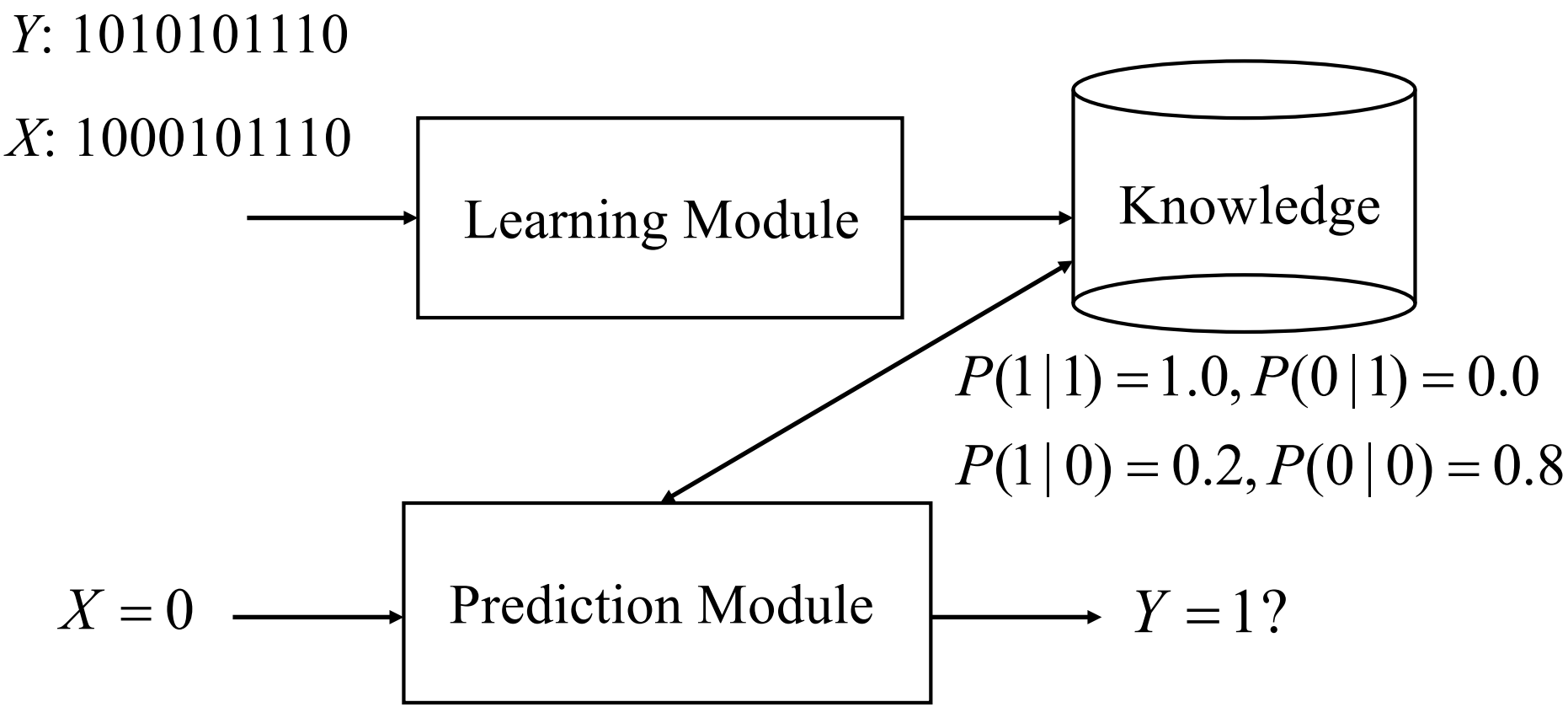
What is Unsupervised Learning?

- Three types of interpretations
- Interpretation 1
  - Supervised: human efforts involved
  - Unsupervised: no human efforts
- Interpretation 2
  - A machine experiences a series of inputs: $x_1, x_2, \ldots, x_N$
  - Supervised: The machine is also given desired outputs $y_1, y_2, \ldots, y_N$
  - Unsupervised: No desired outputs
- Interpretation 3
  - Supervised: learning knowledge concerning to conditional distribution $P(Y|X)$, X=features, Y=*class*
  - Unsupervised: learning knowledge concerning to joint distribution $P(X)$, X=features
  - Semi-supervised learning: learning conditional distribution $P(Y|X)$ with few instances

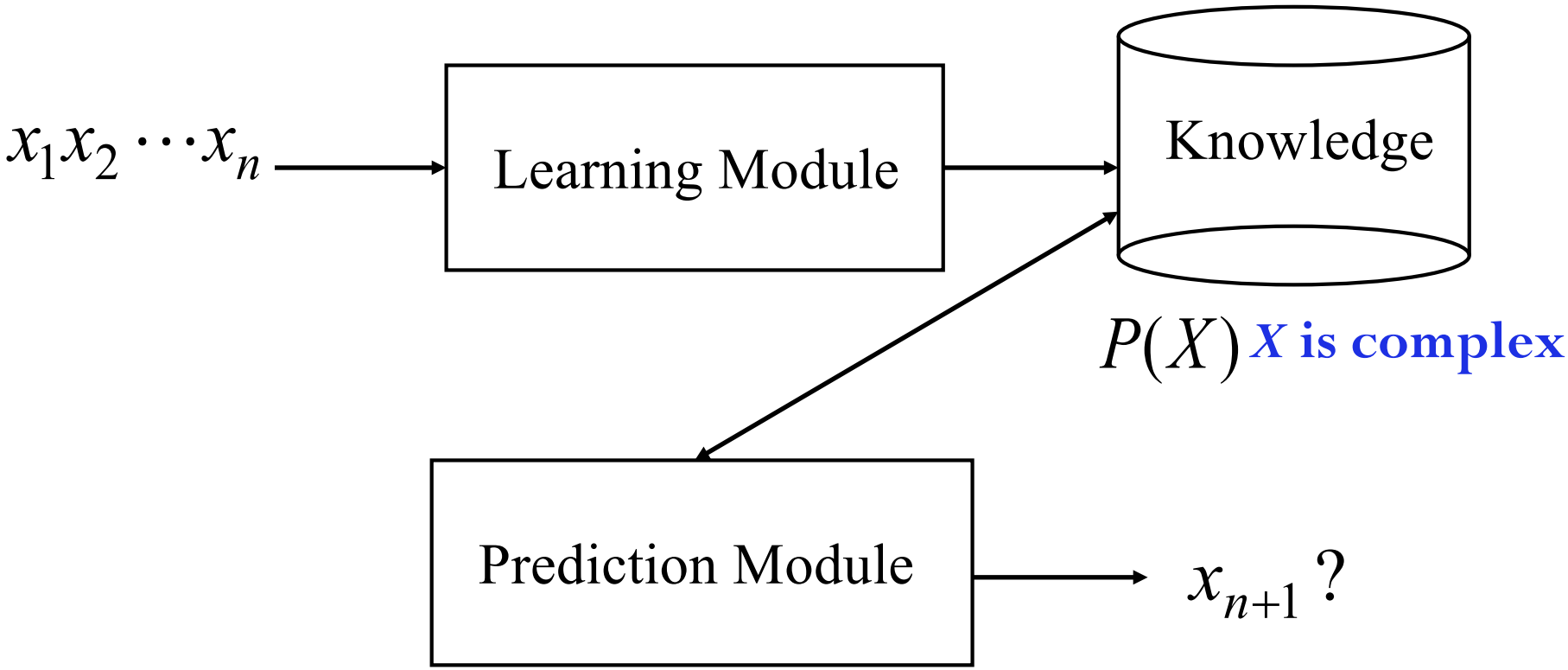
Supervised learning



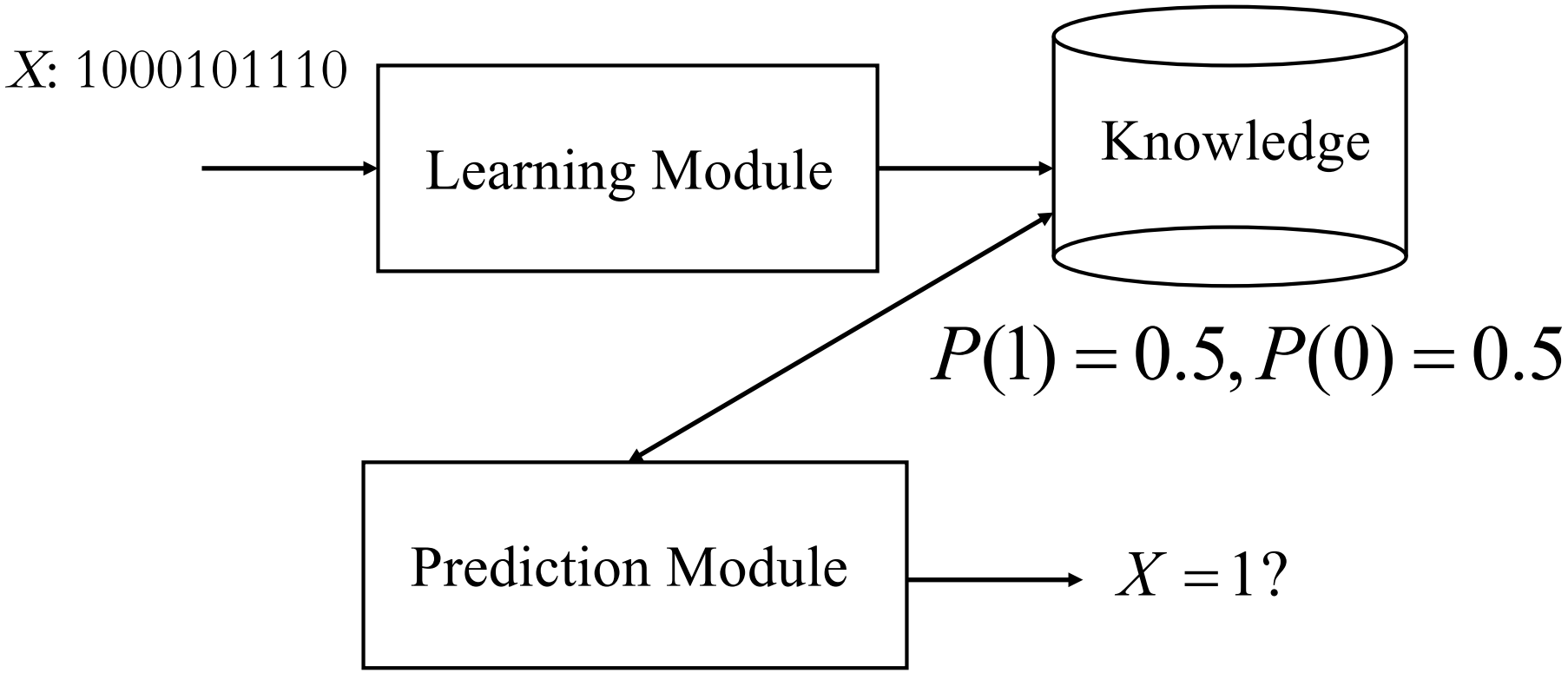
Supervised learning examples



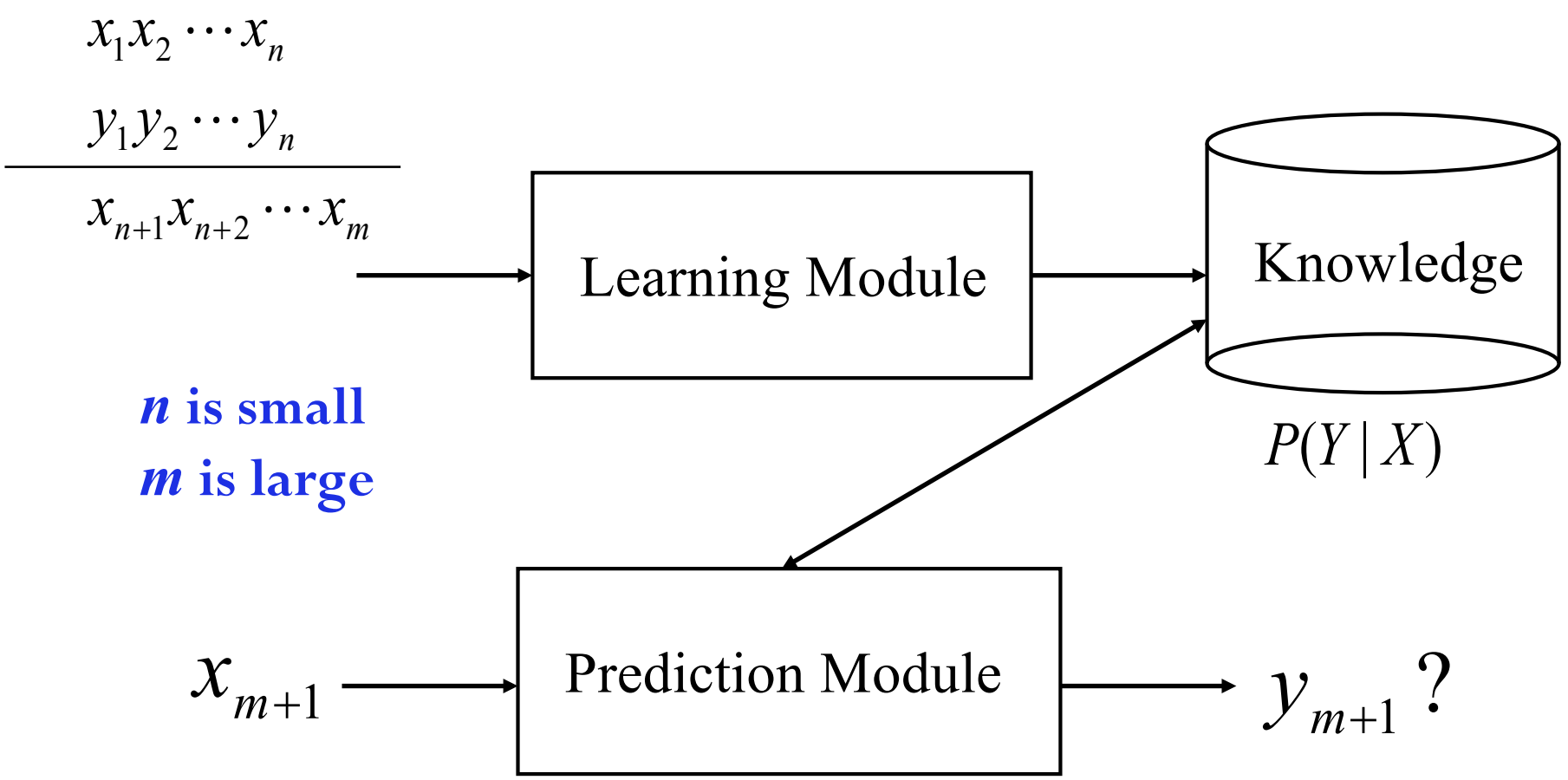
Unsupervised learning



Unsupervised learning example



Semi-supervised learning



Supervised vs Unsupervised

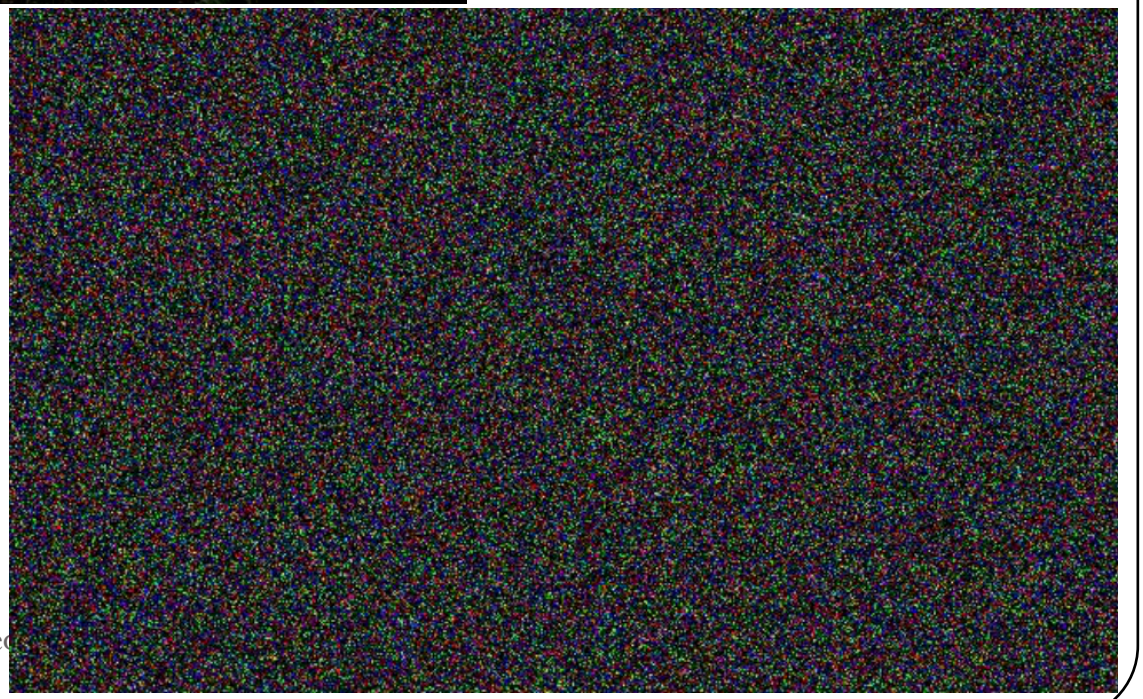
	Supervised	Unsupervised
Instances for learning	(*X*, *Y*) pair, usually with human involvement	*X* only, usually without human involvement

Structure of unlabeled data

- Build a model or find useful representations of the input
  - that can be used for decision making, predicting future inputs, efficiently communicating the inputs to another machine, etc.
- Discover structures in the data
  - An research paper has title, abstract, …
  - Webpages are organized
  - Pixels in images are not randomly generated
  - Different user interests groups
  - …

Original image

After randomly interchanging the pixels and RGB values



What can we do with unlabeled data?

- Data clustering
  - Partition examples into groups when no pre-defined categories/classes are available
- Dimensionality reduction
  - Reduce the number of variables under consideration
- Outlier detection
  - Identification of new or unknown data or signal that a machine learning system is not aware of during training
- Modeling the data density

What can we do with unlabeled data?

- Data clustering
  - Partition examples into groups when no pre-defined categories/classes are available
- Dimensionality reduction
  - Reduce the number of variables under consideration
- Outlier detection
  - Identification of new or unknown data or signal that a machine learning system is not aware of during training
- Modeling the data density

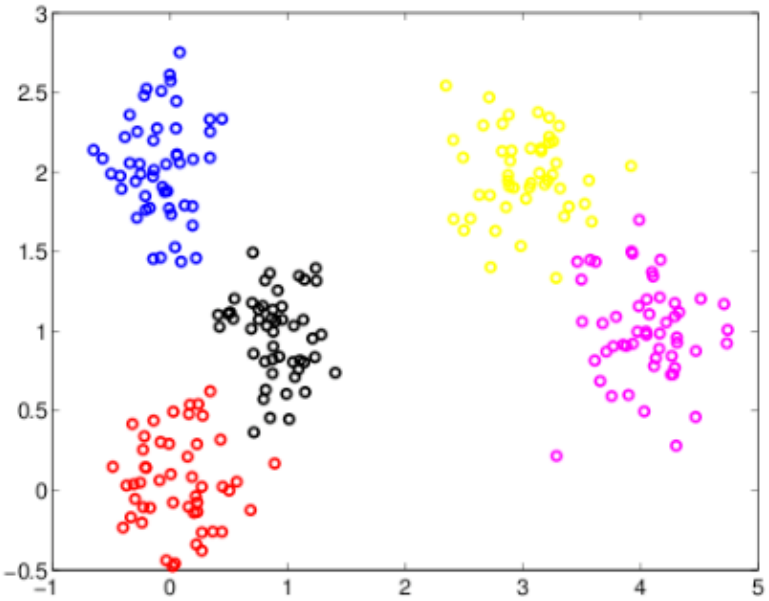
Clustering (聚类)

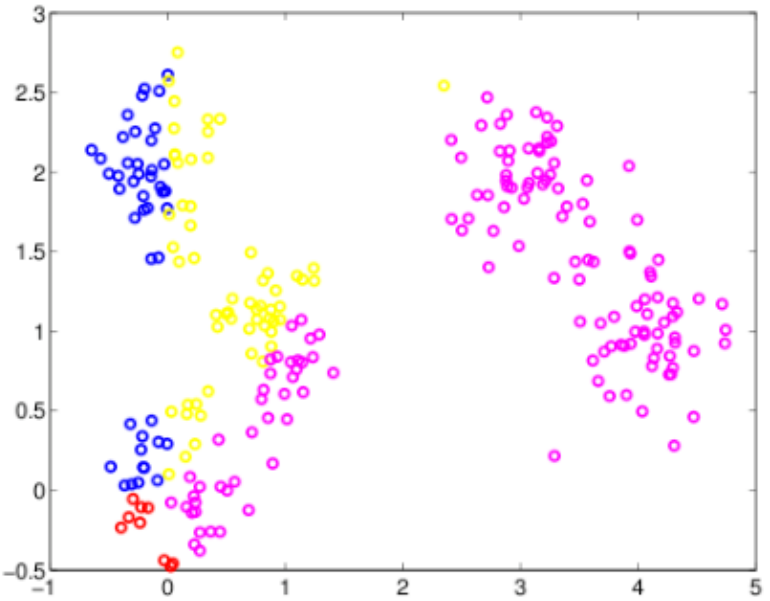
What is Clustering

- Grouping *similar* objects into same "cluster"
  - "Birds of a feather flock together. "

    "物以类聚，人以群分"
  - Discovering structure of data
- Such that those within each cluster are more closely related
  - Objects within the same cluster are similar
  - Objects in different clusters are different
- Core problem: similarity definition
  - Intra cluster similarity (簇/类内相似度)
  - Inter cluster similarity (簇/类间相似度)

What are good clusters?





- The intra-cluster distance is small
- The inter-cluster distance is large

Types of Clustering (1)

- Soft clustering vs. hard clustering
  - Soft: same object can belong to different clusters
  - Hard: same object can only belong to single cluster

Example (1)

e.g. Noun verb co-occurrence data

	eat	drink	make
wine	0	3	1
beer	0	5	1
bread	4	0	2
rice	4	0	0

Hard clustering

Example (2)

e.g. Document data

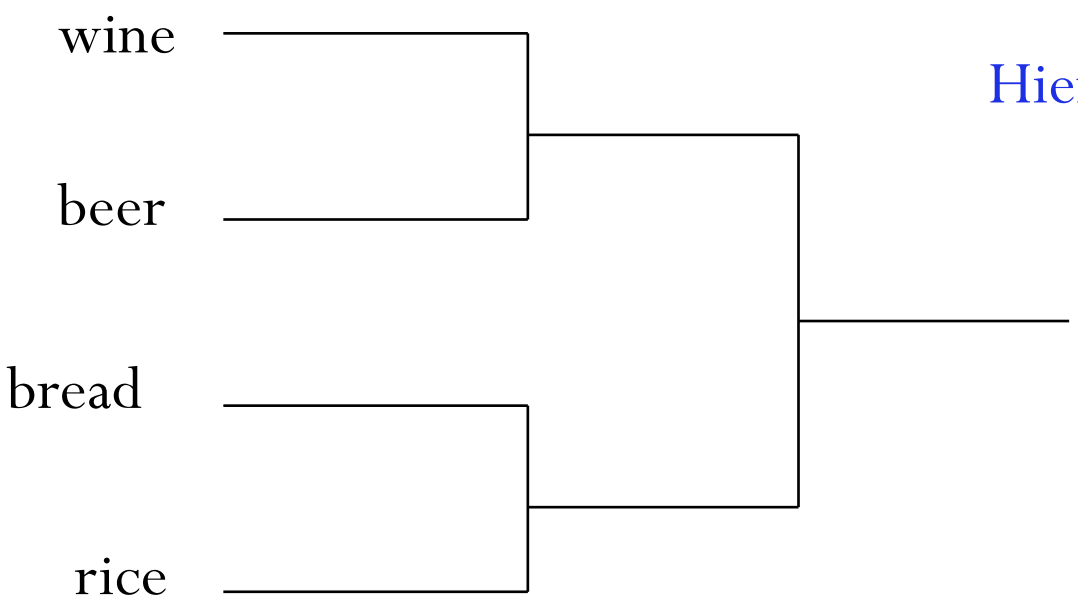
Soft clustering

	parsing	estimation	prediction	translation
document 1	0	3	2	0
document 2	0	5	1	0
document 3	1	1	1	0
document 4	4	0	0	3

Types of Clustering (2)

- Soft clustering vs. hard clustering
  - Soft: same object can belong to different clusters
  - Hard: same object can only belong to single cluster
- Hierarchical clustering vs non-hierarchical clustering
  - Hierarchical:
    - A hierarchy (tree) of clusters
  - Non-hierarchical:
    - Flat , one layer

Example (3)

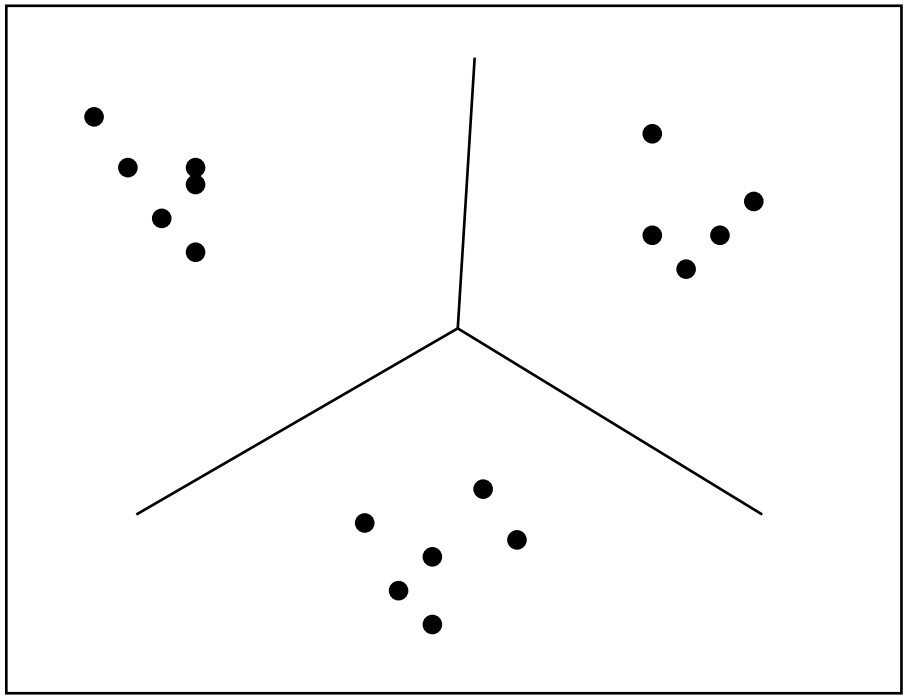


Hierarchy of clusters

Example （4）

Data points in vector space

Non-hierarchical clustering



Application background

- Spatial and GIS data analyses
  - Land usage observation
  - Earthquake analysis
- Image processing
- Economics – especially marketing business intelligence
  - Find different customer groups, e.g. Insurance
- WWW
  - Document/event clustering
    - E.g. Summarize the news for the past month
  - Web log analyses – to find similar user behavior

What do we need to cluster data?

- Unlabeled data
- Distance or similarity measure for objects
  - Minimizing Euclidean distances between objects within clusters
  - Statistical estimation (maximum likelihood estimation, Bayesian estimation) for probability model
- (optional) Distance or similarity measure for clusters
- Clustering algorithm
  - Heuristics, no guarantee to find optimal solution
  - K-means, K-mediods
  - Hierarchical clustering
  - ……

Data and Similarity Measure

Data

- Vector $x \in D_1 \times D_2 \cdots \times D_N$
- Type
  - Real values: D=R
  - Binary values
    - D = {v1,v2}
    - e.g., { Female, Male}
  - Nominal values
    - D = {v1,v2,…,vM}
    - e.g., { Mon, Tue, Wed, Thu, Fri, Sat, Sun}
  - Ordinal values
    - D = R or D = {v1,v2,…,vM}
    - Order is important, e.g., rank

Similarity Measures (1)

- Similarity = $(\text{Distance})^{-1}$
- Real valued data
  - Inner product
  - Cosine Kernels
  - Kernels
  - … and many more
- Minkowski distance
  - Manhattan distance
  - Euclidean distance
  - Chebyshev distance

Similarity Measures (2)

- Nominal values
  - E.g. "Boston", "LA", "Pittsburgh",
  - or "male", "female",
  - or "diffuse", "globular", "spiral", "pinwheel"
  - Binary rule
    - If $x_i = x_j$, then $\text{sim}(x_i, x_j) = 1$, else $\text{sim}(x_i, x_j) = 0$
  - Use underlying semantic property
    - E.g. $\text{Sim}(\text{Boston}, \text{LA}) = \alpha \text{dist}(\text{Boston}, \text{LA})^{-1}$,
    - or $\text{Sim}(\text{Boston}, \text{LA}) = \alpha(|\text{size}(\text{Boston}) - \text{size}(\text{LA})|) / \text{Max}(\text{size}(\text{cities}))$
  - Use similarity Matrix

Similarity Measures (2): Similarity Matrix

	tiny	little	small	medium	large	huge
tiny	1.0	0.8	0.7	0.5	0.2	0.0
little		1.0	0.9	0.7	0.3	0.1
small			1.0	0.7	0.3	0.2
medium				1.0	0.5	0.3
large					1.0	0.8
huge						1.0

- Diagonal must be 1.0
- No linearity (value interpolation) assumed
- Qualitative Transitive property must hold

Similarity Measures (3)

- For Ordinal Values
  - E.g. "small," "medium," "large," "X-large"
  - Convert to real variable on a normalized [0,1] scale, where:
    - max(v)=1, min(v)=0, others interpolate
    - E.g. "small"=0, "medium"=0.33, etc.
  - Then, use similarity measures for real variable
  - Or, use similarity matrix

K-means clustering
(K-均值聚类法)

K-means

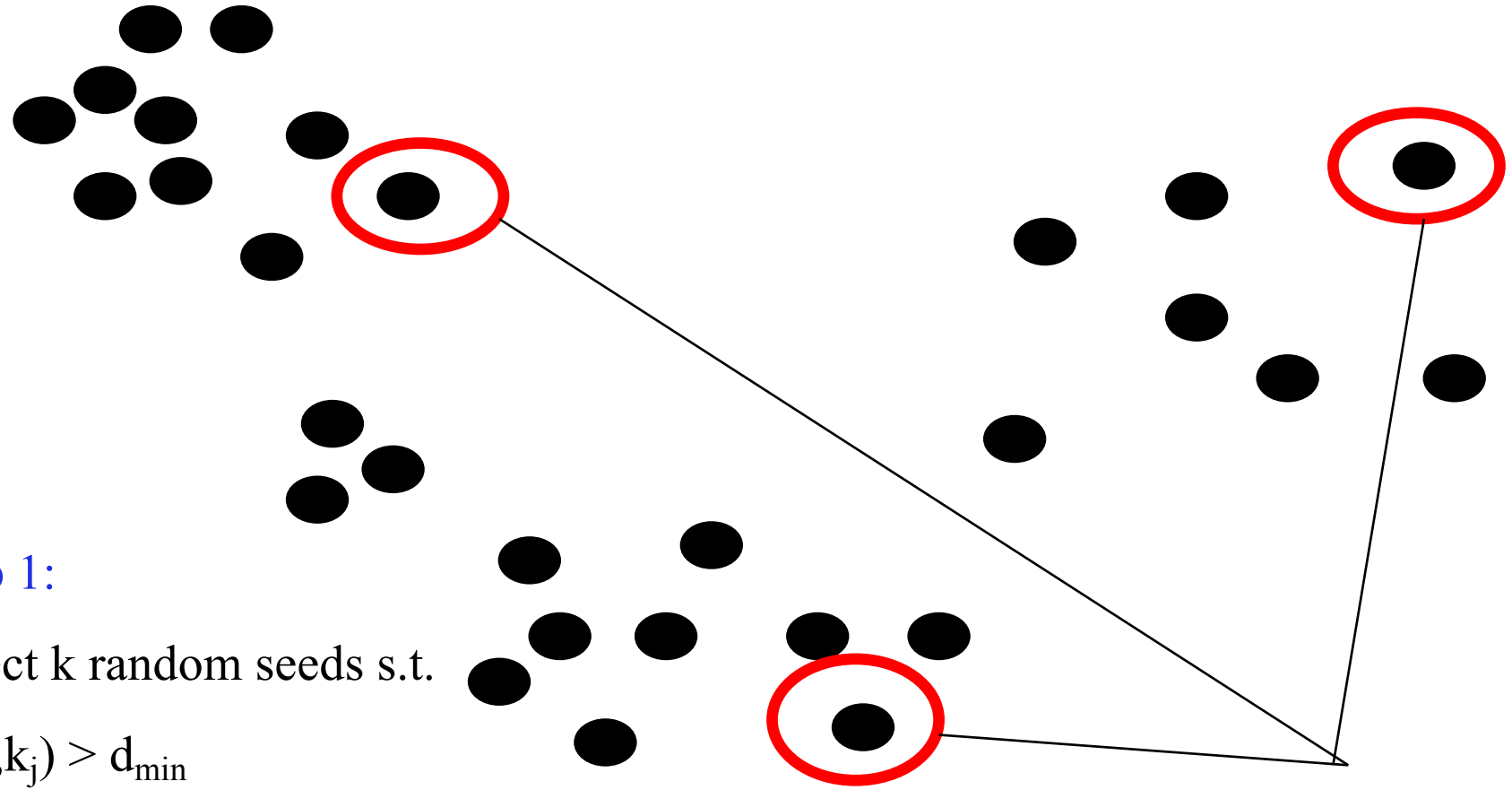
- Model: vector space model
- Strategy: minimizing Euclidean distances between objects within clusters
- Algorithm: *iterative algorithm*
- Hard clustering
- Non-hierarchical

Algorithm

- For a given cluster assignment C, finding the mean vector for each cluster: $\{m_1, \cdots m_k\}$
- Given a set of k mean vectors $\{m_1, \cdots m_k\}$, assign each object to the closest cluster mean
- Repeat the above steps until no change in evaluation function

- No guarantee to find the optimal solution

K-means example: initial data



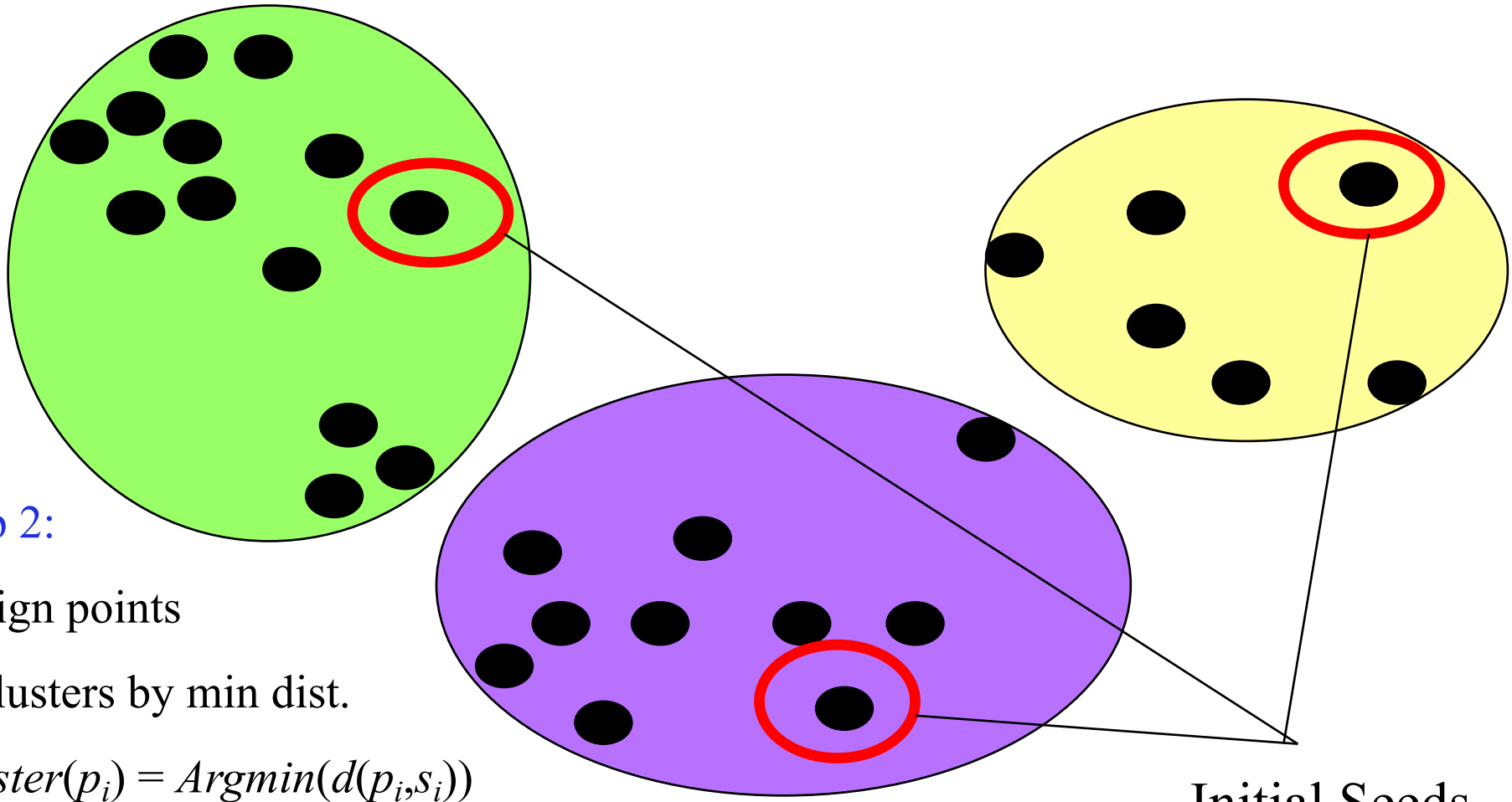
Step 1:

Select k random seeds s.t.

$d(k_i,k_j) > d_{min}$

Initial Seeds (if k=3)

K-means example: 1st -pass clusters



Step 2:

Assign points

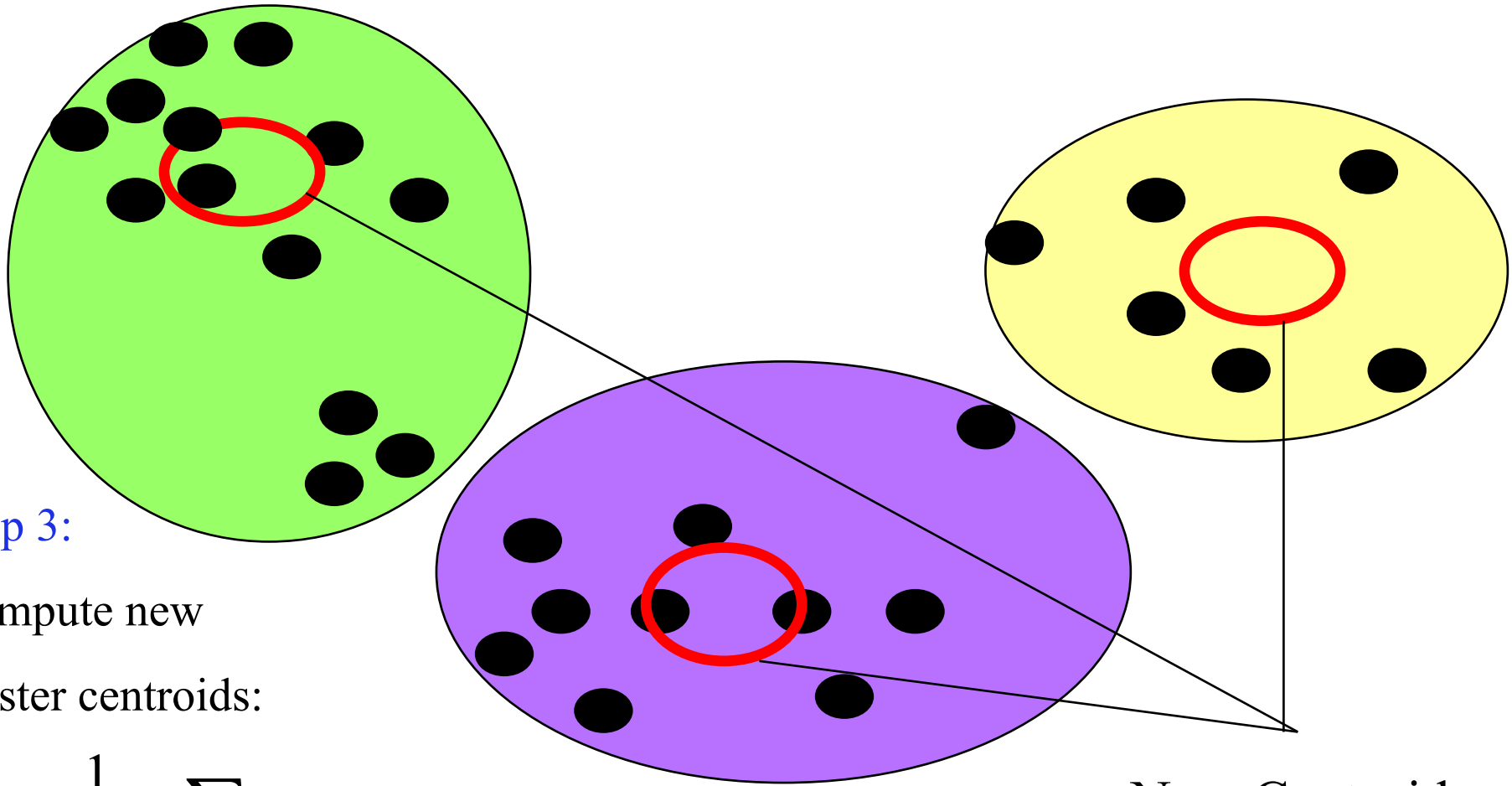
to clusters by min dist.

$Cluster(p_i) = Argmin(d(p_i,s_i))$

$s_i \in \{s_1,\ldots,s_k\}$

Initial Seeds

K-means example: Seeds → Centroids



Step 3:

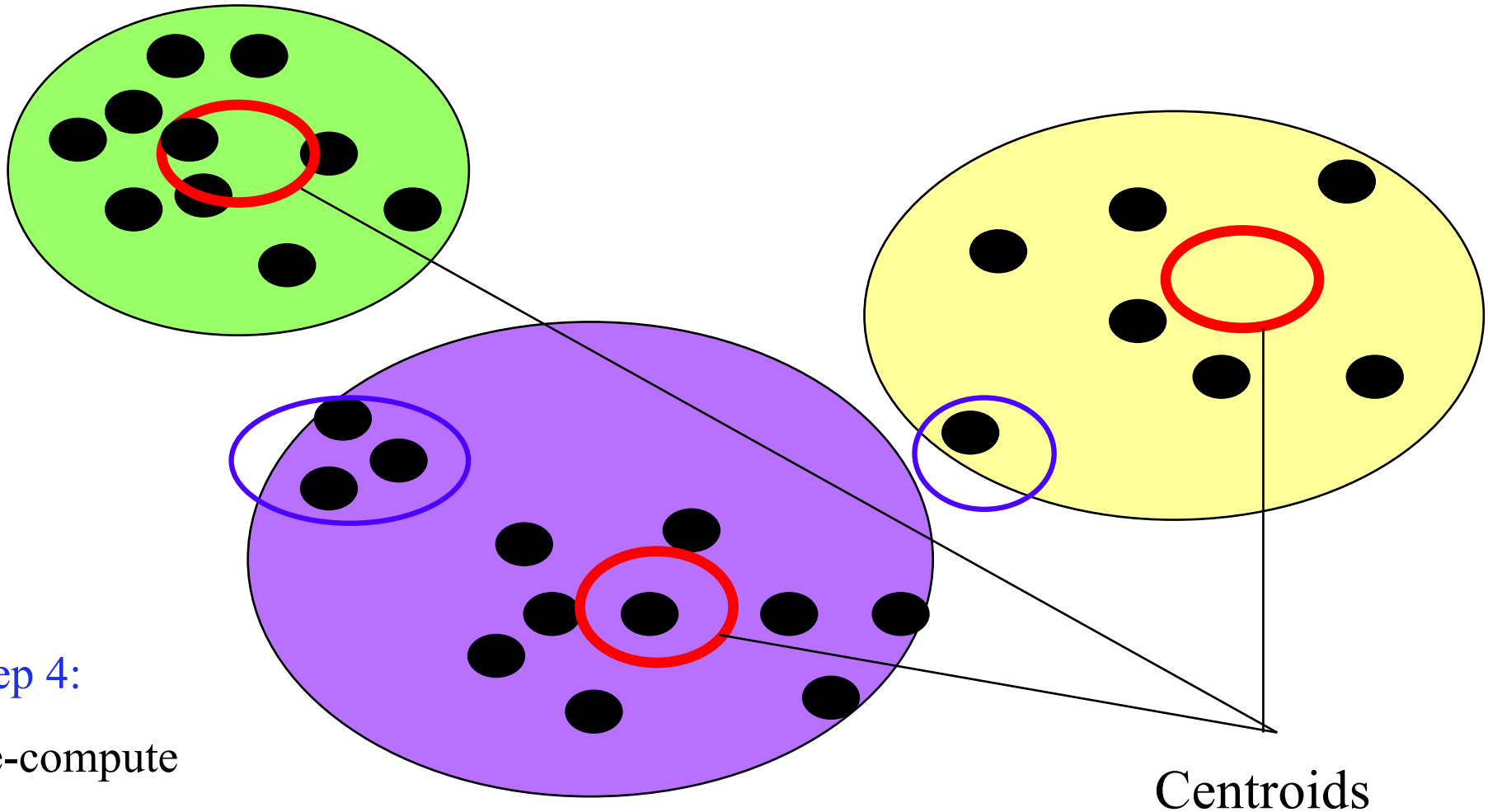
Compute new

cluster centroids:

$$\vec{c}_j = \frac{1}{n} \sum_{p_i \in j^{th} cluster} \vec{p}_i$$

New Centroids

K-means example : 2nd – pass clusters

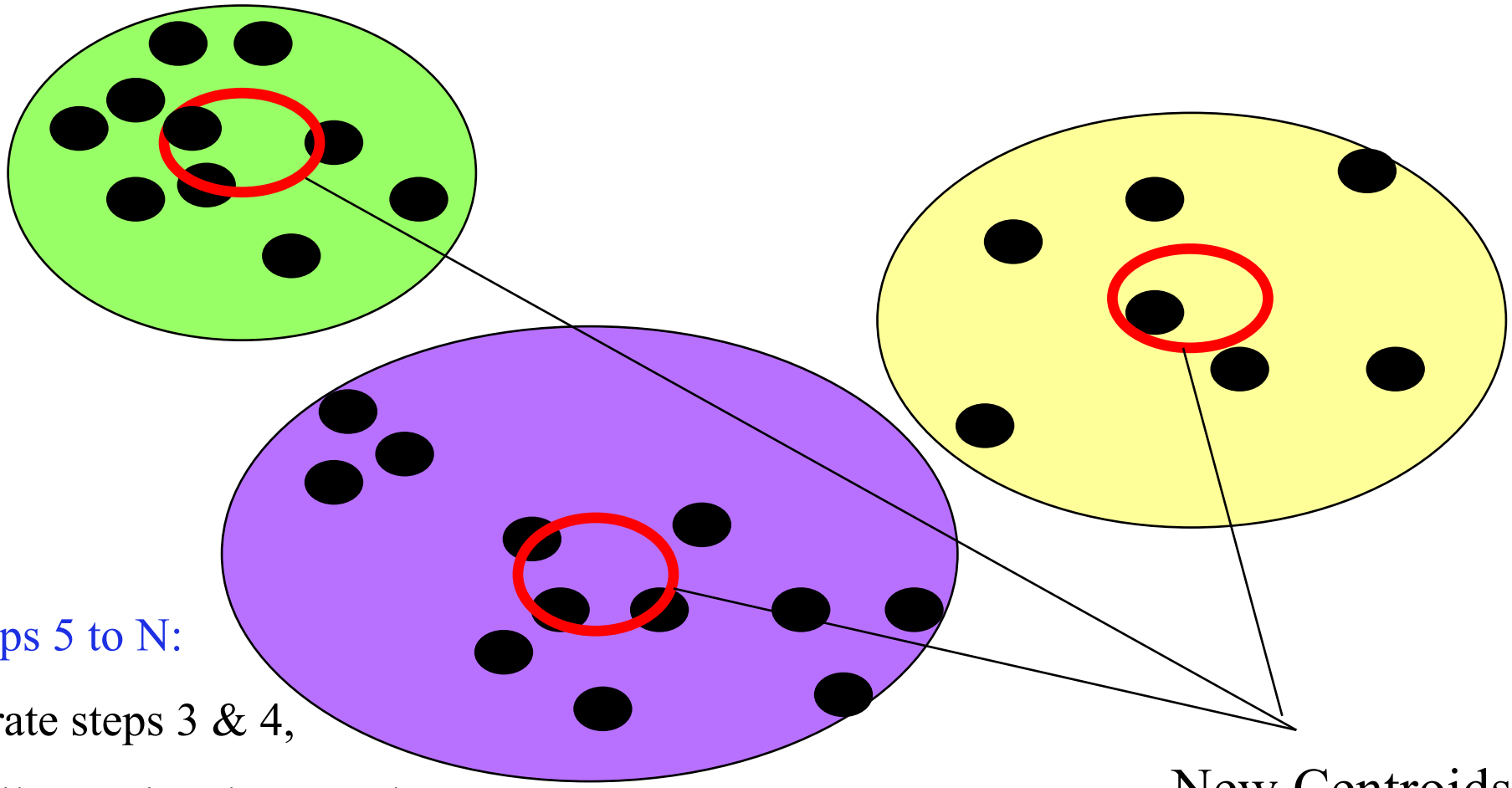


Step 4:

Re-compute

$Cluster(p_i) = Argmin(d(p_i,c_j)) \quad c_j \in \{c_1,\ldots,c_k\}$

K-means example: iterate until stability

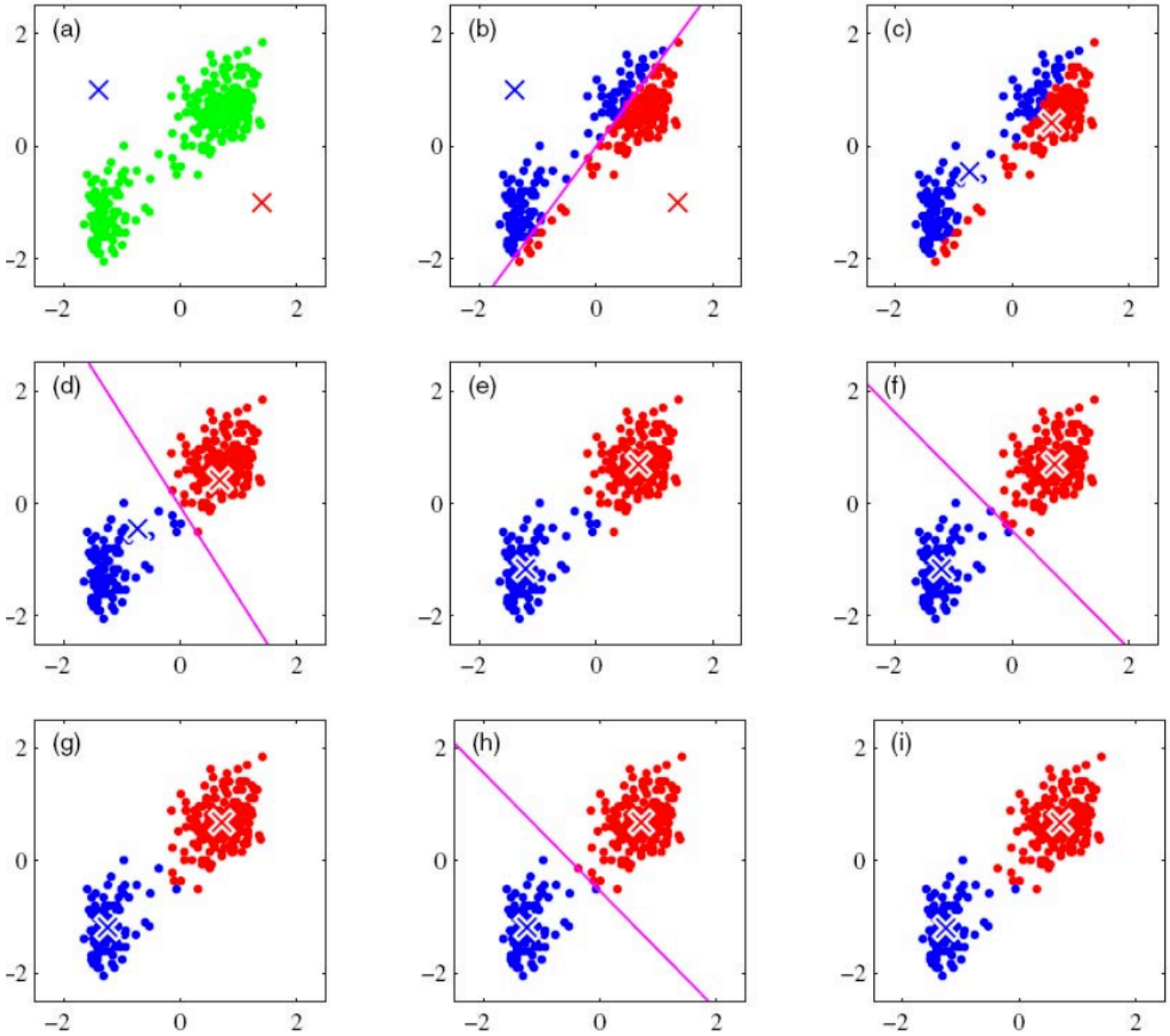


Steps 5 to N:

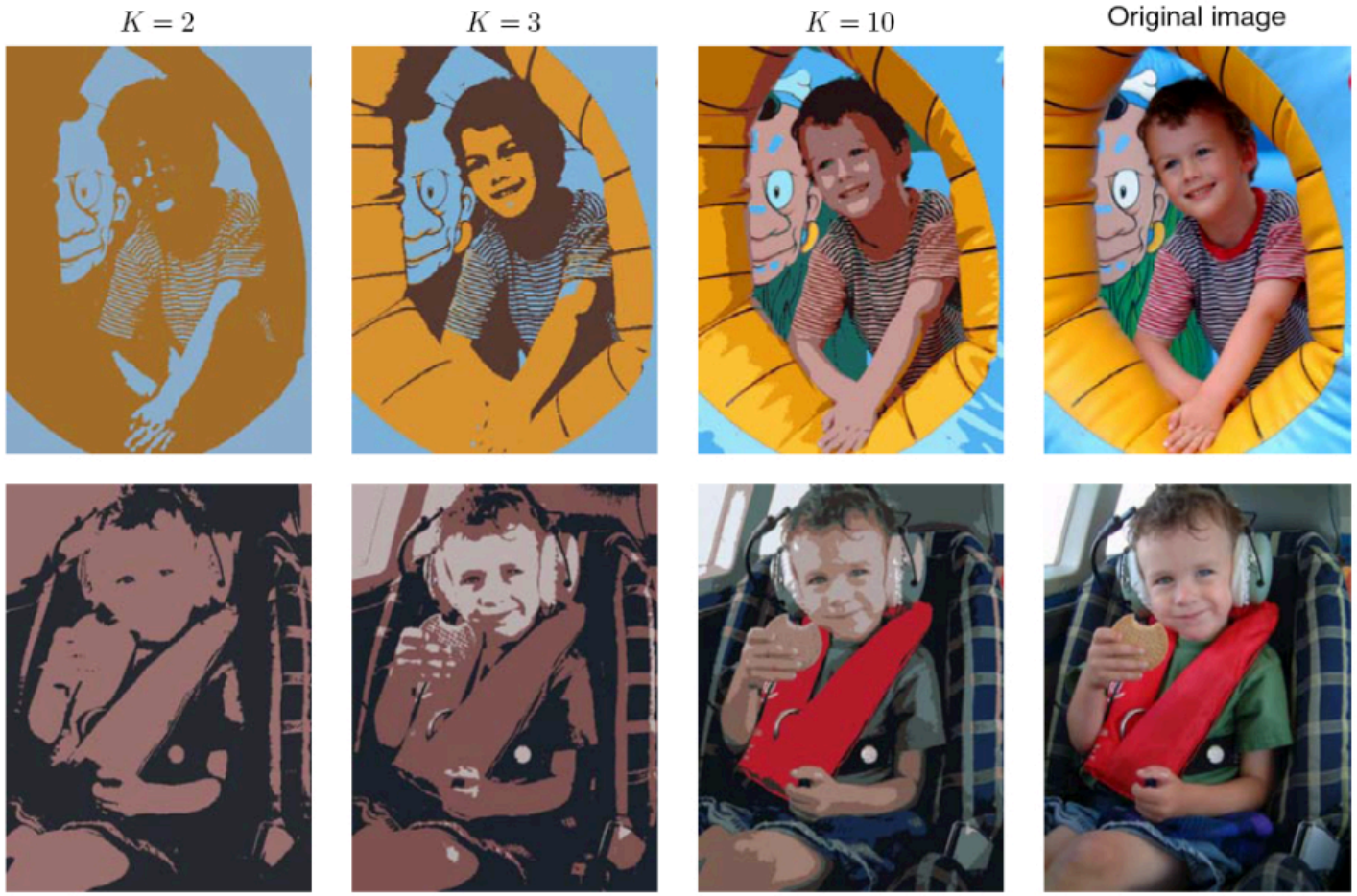
Iterate steps 3 & 4,

until no point changes cluster

New Centroids


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

K-means example: image segmentation

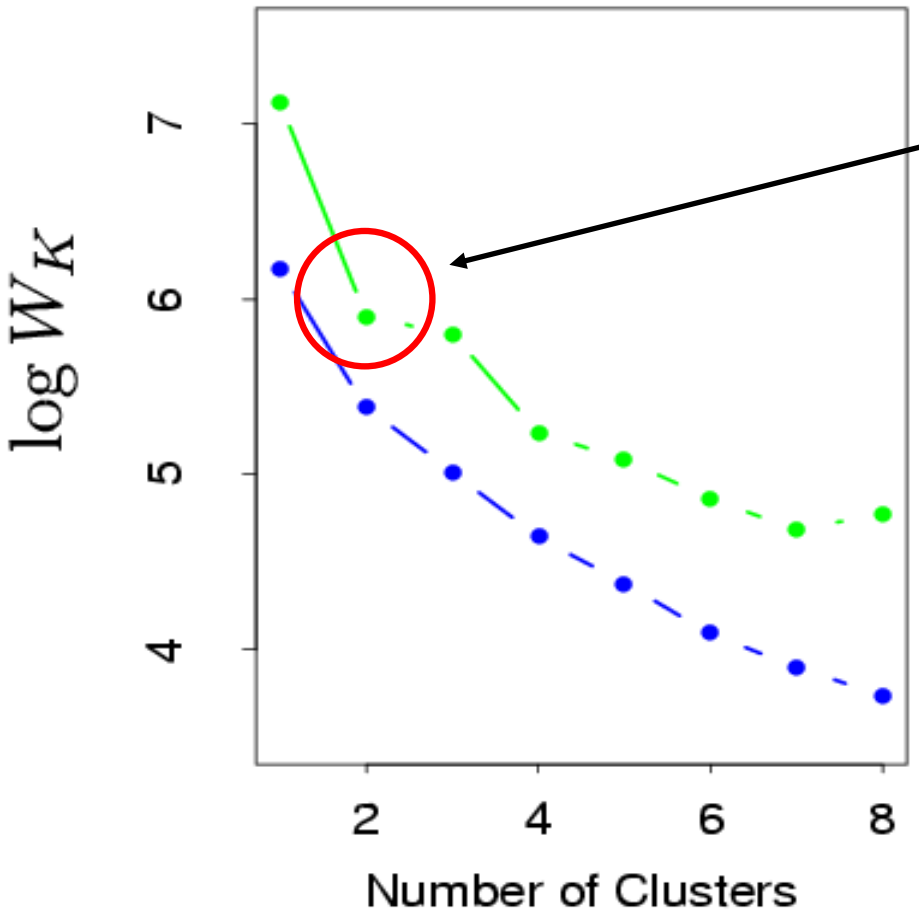


Every pixel is redrawn with the {R, G,B} intensity given by the centre g_v to which that pixel has been assigned.

How to decide “k”?

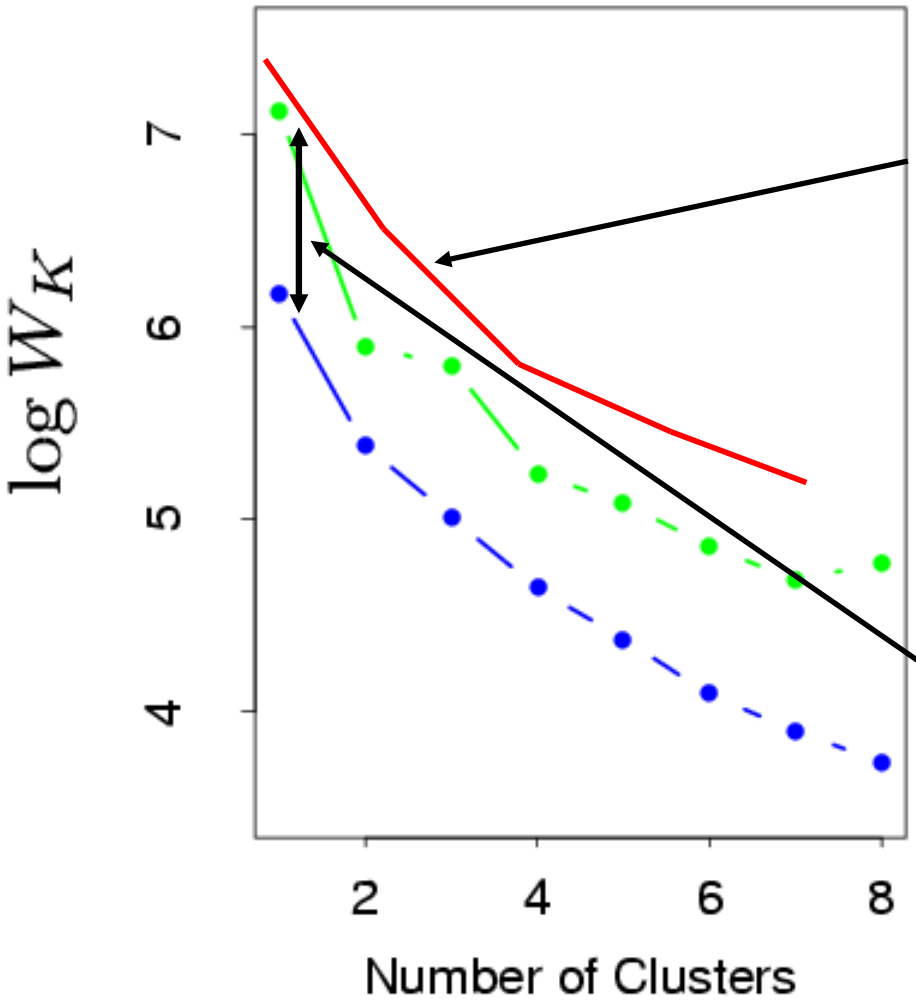
- Problem driven
  - Usually, the problem itself has the setting of K
- Could be ”data driven” only when either
  - Data is not sparse
  - Measurement dimensions are not too noisy
- If K is not given
  - We examine the Intra-cluster dissimilarity W_k (inverse to intra-cluster similarity) (or examine inter-cluster similarity), which is a function of K
- Usually W_k decreases with increasing K

How to decide “k”? （method 1）



A sharp drop of W_k is observed

How to decide “k”? （method 2）



Use the same clustering algorithm on uniformly distributed data

Measure the gap

Choosing“k”: other Approaches

- Bayesian Estimation
  - Estimate posterior distribution on k, given data and prior on k.
  - Difficulty: Computational complexity of integration
  - Auto-class algorithm of (Cheeseman′98) uses approximations
  - (Diebolt′94) suggests sampling techniques
- Cross Validation Likelihood
  - Find ML estimate on part of training data
  - Choose k that maximizes the cross-validated average likelihoods on held-out data D^{test}

K-means analyses

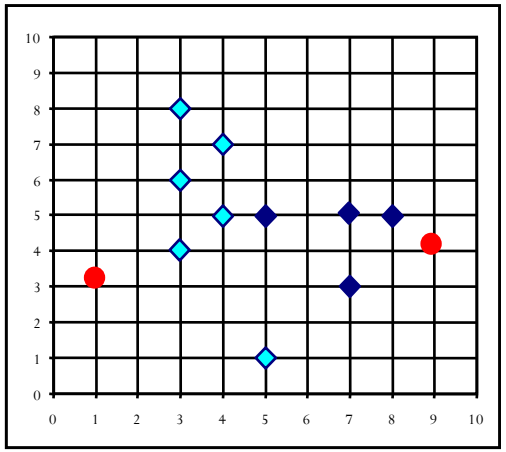
- Attempts to determine k partitions that minimizes the squared-error function.
- It works well when the clusters are compact clouds that are rather well separated from one another.
- It is suitable for discovering clusters with non-convex shapes or clusters with quite different size.
- It is very sensitive to noise and outlier points
  - Since an object with an extremely large value may substantially distort the distribution of data.

K-medoids clustering
(K-中心聚类法)

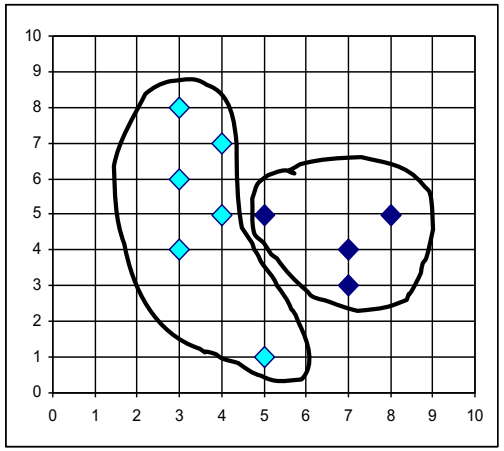
K- medoids

- Use the medoid – most centrally located object in a cluster, as a reference point of a cluster
  - Instead of taking the mean value of the objects
- The basic strategy:
  - Find k clusters in n objects
    - By first arbitrarily finding a representative object for each cluster
  - Iteration:
    - Each remaining object is clustered with the medoid to which it is the most similar
    - Replaces one of the medoids by one of the non-medoids
    - As long as the quality of the resulting clustering is improved
- The quality of the cluster
  - A cost function: the average dissimilarity(object, the medoid)

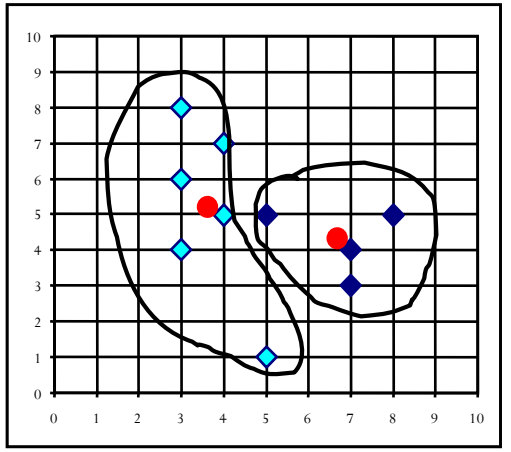
Recall: K-Means (example)



Assign each objects to most similar center

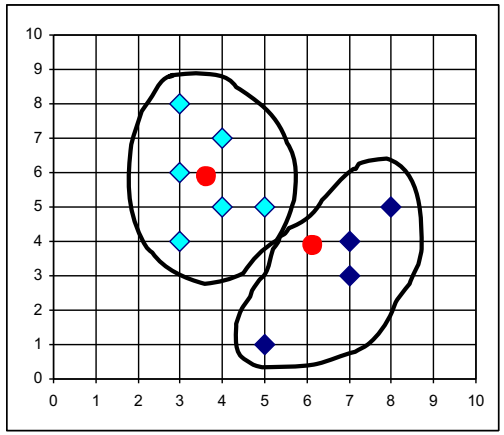


Update the cluster means

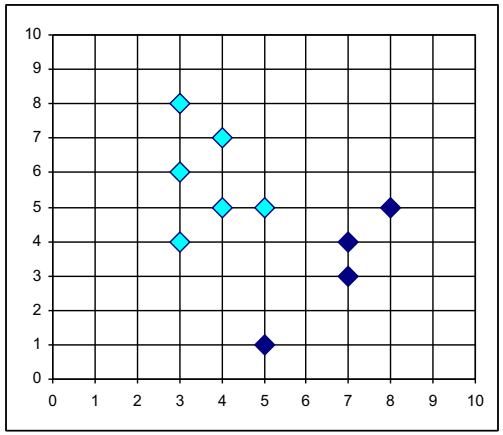


reassign

reassign



Update the cluster means



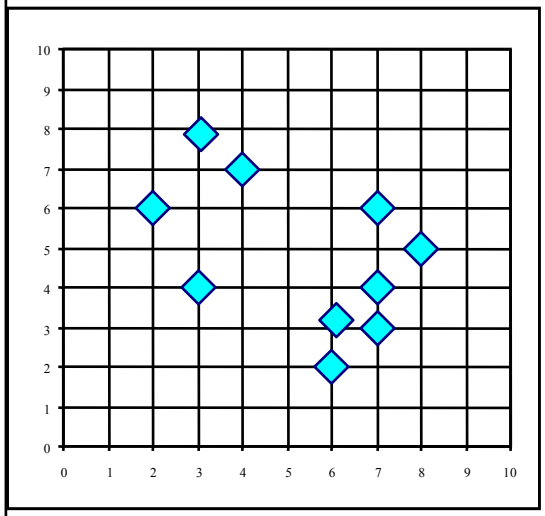
K=2

Arbitrarily choose K object as initial cluster center

Do loop

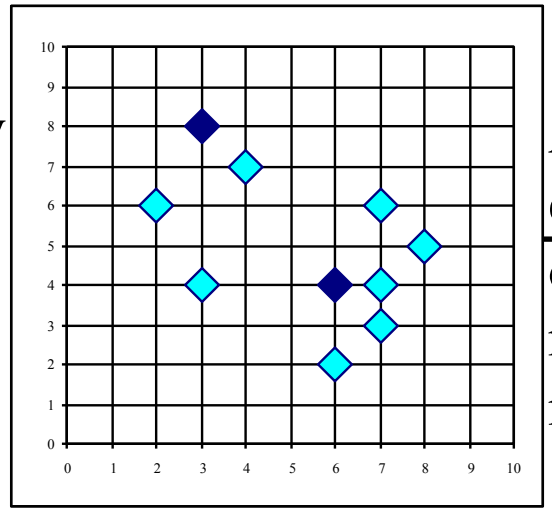
Until no change

Compare: K-Medoids (example)



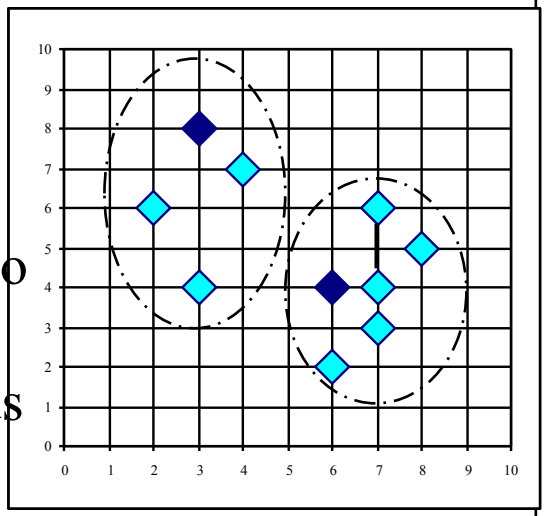
K=2

Arbitrary choose k object as initial medoids

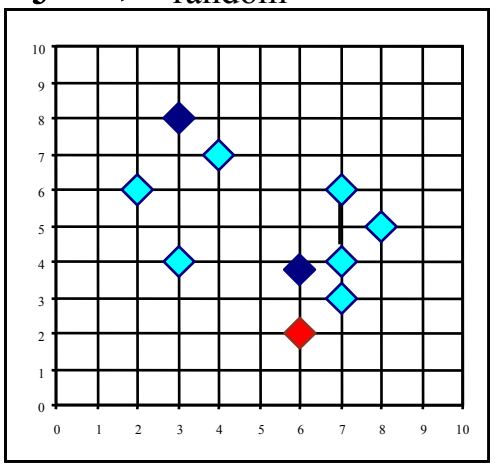


Assign each object to nearest medoids

Total Cost = 20



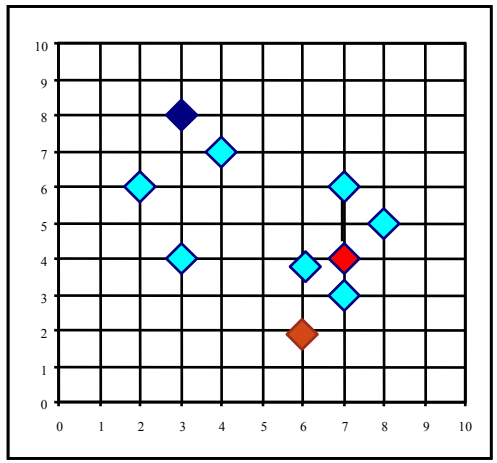
Randomly select a non-medoid object, O_{random}



Compute total cost

Cost= 26

Not swap



Swapping O and O_{random}

If quality is improved.

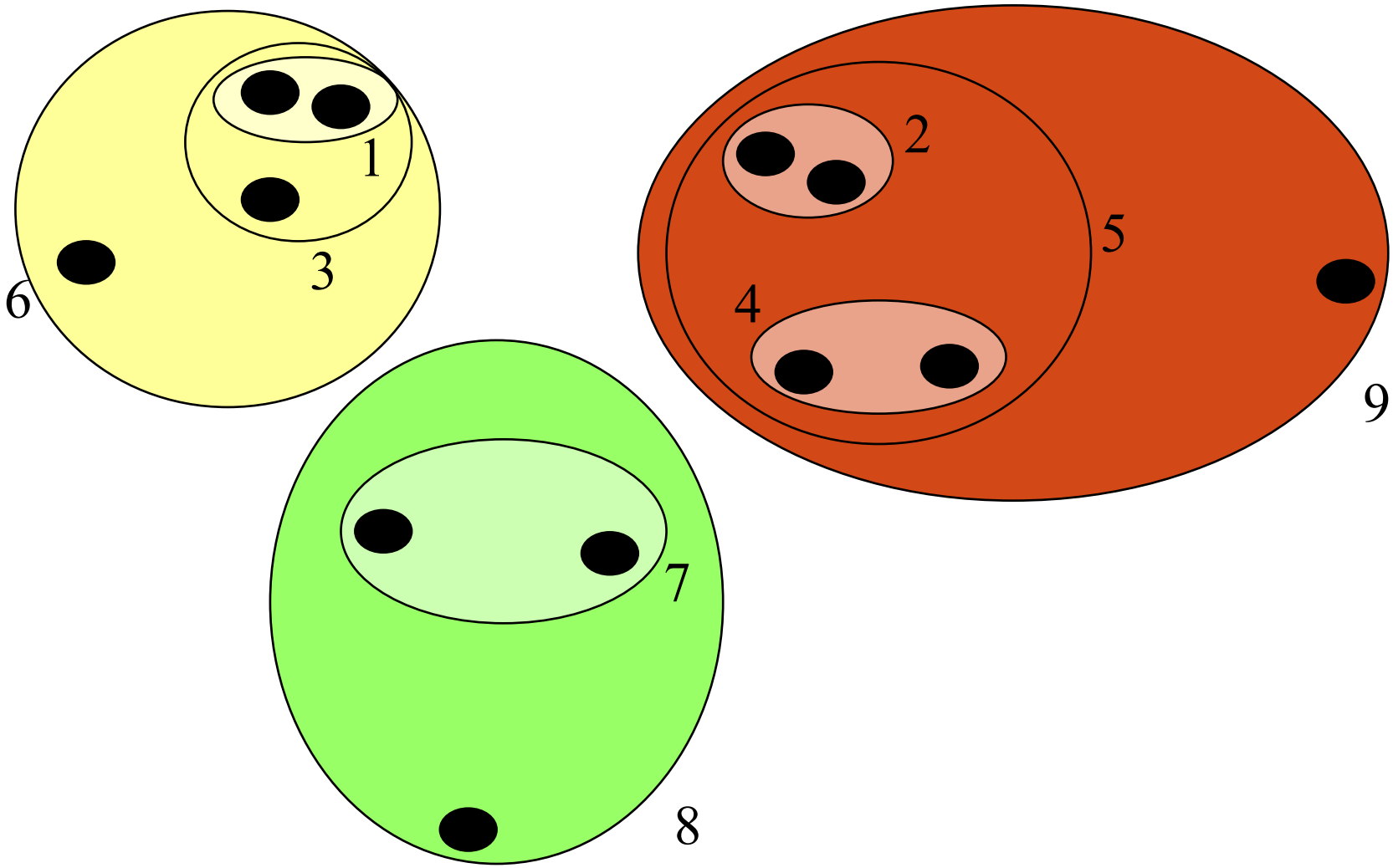
Do loop

Until no change

Hierarchical Agglomerative Clustering
层次凝聚式聚类（自底向上）

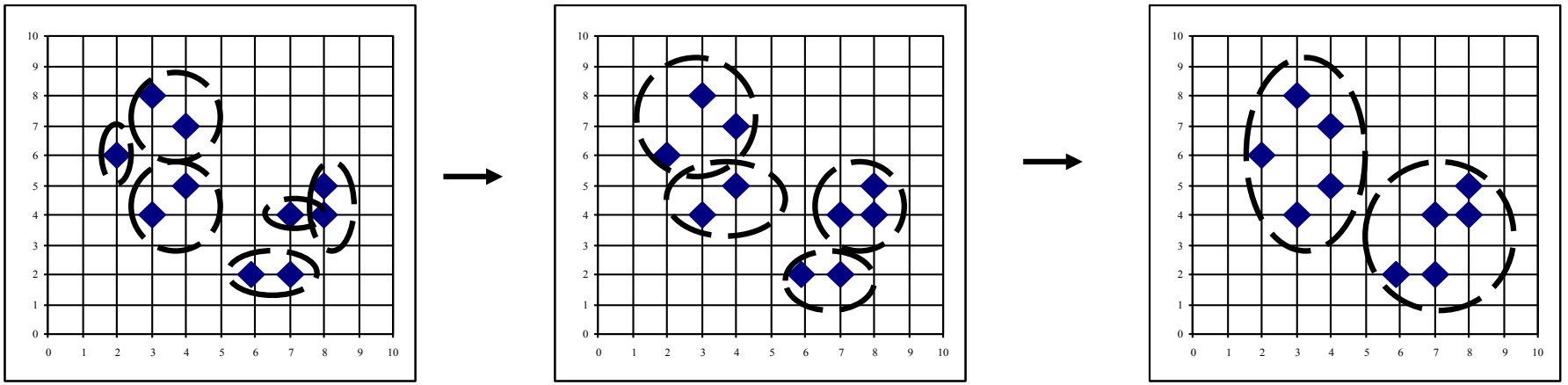
Hierarchical Agglomerative Clustering

- Generic Agglomerative Procedure (Salton '89):
  - Result in nested clusters via iterations
- Algorithm (e.g. document clustering)
  - Compute all pairwise document-document similarity coefficients
  - Place each of n documents into a class of its own
  - Merge the two **most similar clusters** into one;
    - Replace the two (or more) clusters by the new cluster
    - Re-compute inter-cluster similarity scores w.r.t. the new cluster
  - Repeat the above step until there are only k clusters left (note k could = 1).


1
3
6
2
5
4
9
7
8

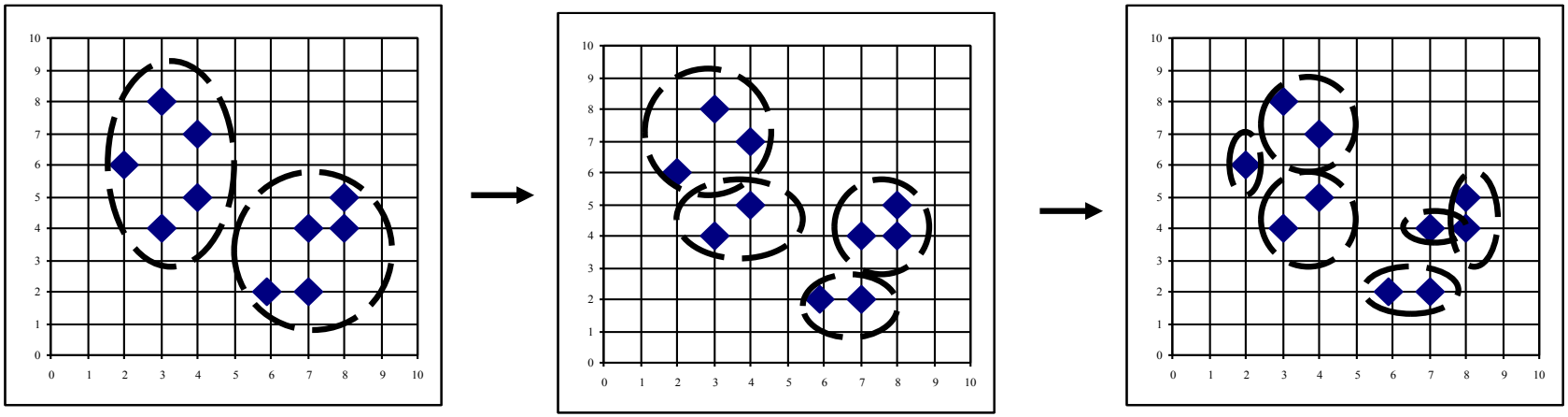
Hierarchical Clustering

- Agglomerative



Hierarchical Clustering

- Divisive



Cluster similarity

- Single linkage: similarity of two most similar members of each cluster

$$\text{sim}(c_i, c_j) = \min_{x \in c_i, y \in c_j} \text{sim}(x, y)$$

$$\text{sim}(c_i \cup c_j, c_k) = \min(\text{sim}(c_i, c_k), \text{sim}(c_j, c_k))$$

- Complete linkage: similarity of two least similar members of each cluster

$$\text{sim}(c_i, c_j) = \max_{x \in c_i, y \in c_j} \text{sim}(x, y)$$

$$\text{sim}(c_i \cup c_j, c_k) = \max(\text{sim}(c_i, c_k), \text{sim}(c_j, c_k))$$

- Average linkage: mean similarity between members of each cluster

$$\text{sim}(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{x \in c_i, y \in c_j} \text{sim}(x, y)$$

$$\text{sim}(c_i \cup c_j, c_k) = \frac{1}{|c_i| + |c_j|}(|c_i|\text{sim}(c_i, c_k), |c_j|\text{sim}(c_j, c_k))$$

Analysis on agglomerative clustering

- Advantages
  - Embedded flexibility regarding the level of granularity
  - Ease of handling of any forms of similarity or distance
  - Consequently, applicability to any attribute types
- Disadvantages
  - Vagueness of termination criteria
  - Computationally expensive, difficult to deal with large datasets
- In general case the complexity of agglomerative clustering is $O(n^3)$.

Conclusion

- Supervised v.s. unsupervised learning
- Clustering
  - Introduction
  - K-means
  - K-medoids
  - Hierarchical Clustering
    - Agglomerative (bottom-up)
    - Division (top-down)