パターン認識
Pattern Recognition

佐藤真一
Shin'ichi Satoh

国立情報学研究所
National Institute of Informatics

June 27, 2023

Final Report

- Find any PR paper in top journals or top conferences
- e.g., IEEE TPAMI, IJCV, CVPR, ICCV, NeurIPS, ICML, ACMMM...
- Describe the following:
  - bibliographic info of the paper
  - brief of the paper
  - what is the problem, why it's important, how it's solved, validation?
  - why you selected the paper, what is exciting?
  - feedback to the lecture, any comments
- 2-4 pages A4
- due: 07/31/2023
- send via ITC-LMS

Final Report

- Geman and Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, TPAMI, 1984.
- Moghaddam and Pentland, Probabilistic Visual Learning for Object Detection, TPAMI, 1997.
- Belhumeur, Hespanha, and Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, TPAMI, 1997.
- Shi and Malik, Normalized Cuts and Image Segmentation, TPAMI, 2000.
- Belkin and Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, NIPS, 2001.
- Zhang and Sim, When Fisher meets Fukunaga-Koontz: A New Look at Linear Discriminants, CVPR, 2006.
- Felzenszwalb, Girshick, McAllester, and Ramanan, Object Detection with Discriminatively Trained Part Based Models, TPAMI, 2009.
- Antonio Torralba and Alexei A. Efros, Unbiased look at dataset bias, CVPR 2011.

Today's topics

Clustering Techniques

- k-Means
- Agglomerative Hierarchical Clustering
- Dendrogram
- Evaluation

Clustering

- So far we assumed that the class labels are given for training samples.
- Sometimes it's very costly to provide class labels.
- What can we do if we don't know class labels?
- Unsupervised methods, or smart preprocessing methods
- Clustering discovers distinct subclasses observed in the data distribution.

Clustering

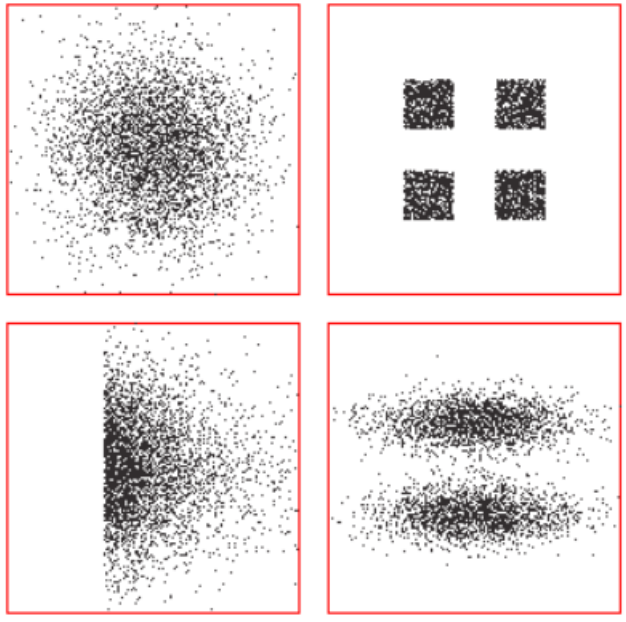


FIGURE 10.6. These four data sets have identical statistics up to second-order—that is, the same mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. In such cases it is important to include in the model more parameters to represent the structure more completely. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

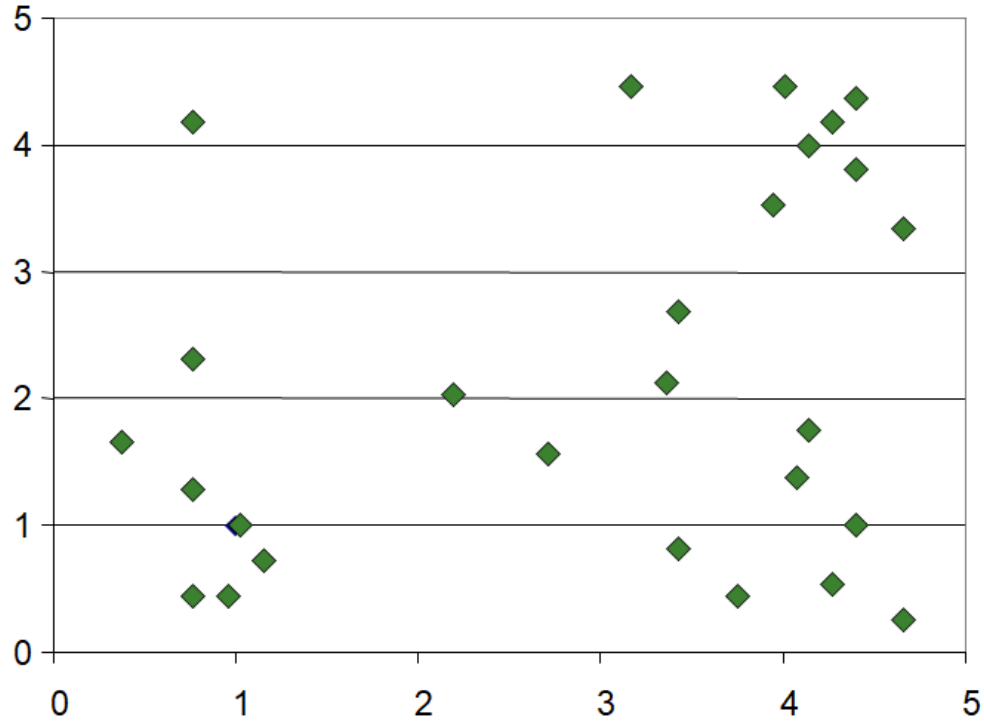
Algorithm k-means

Input: N data points
Output: k cluter centers

1. Determine the number of clusters: k
2. (Randomly) guess k cluster center locations
3. Each data point finds out which center it's closest to
4. Each center finds the centroid of the points it owns
5. Terminate if assignment of N data points does not change
6. Repeat from 3 otherwise

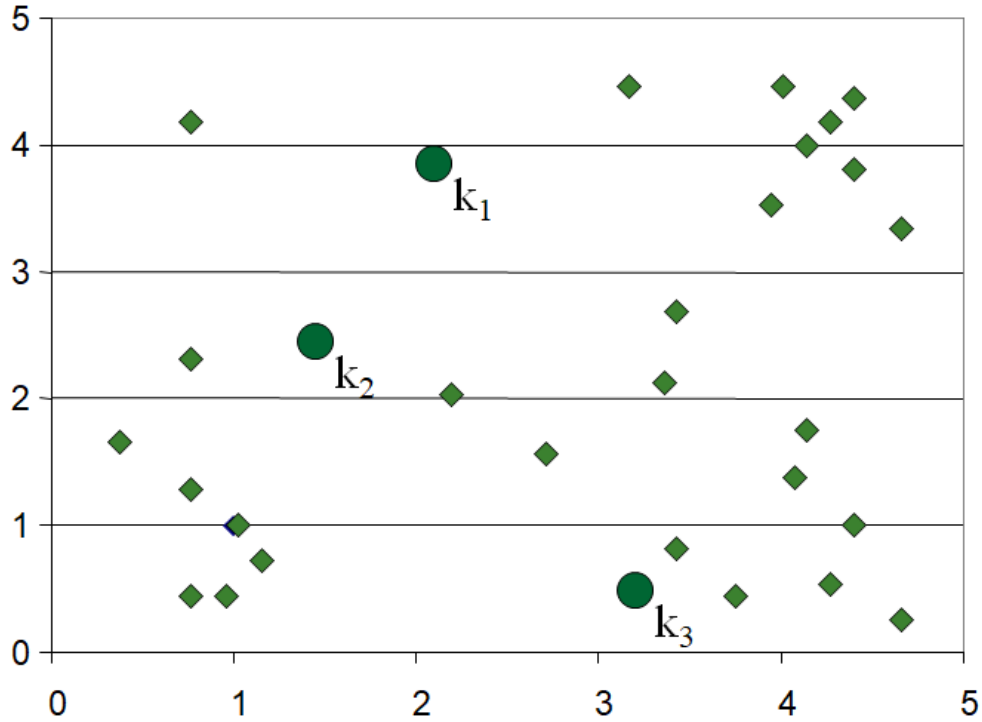
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



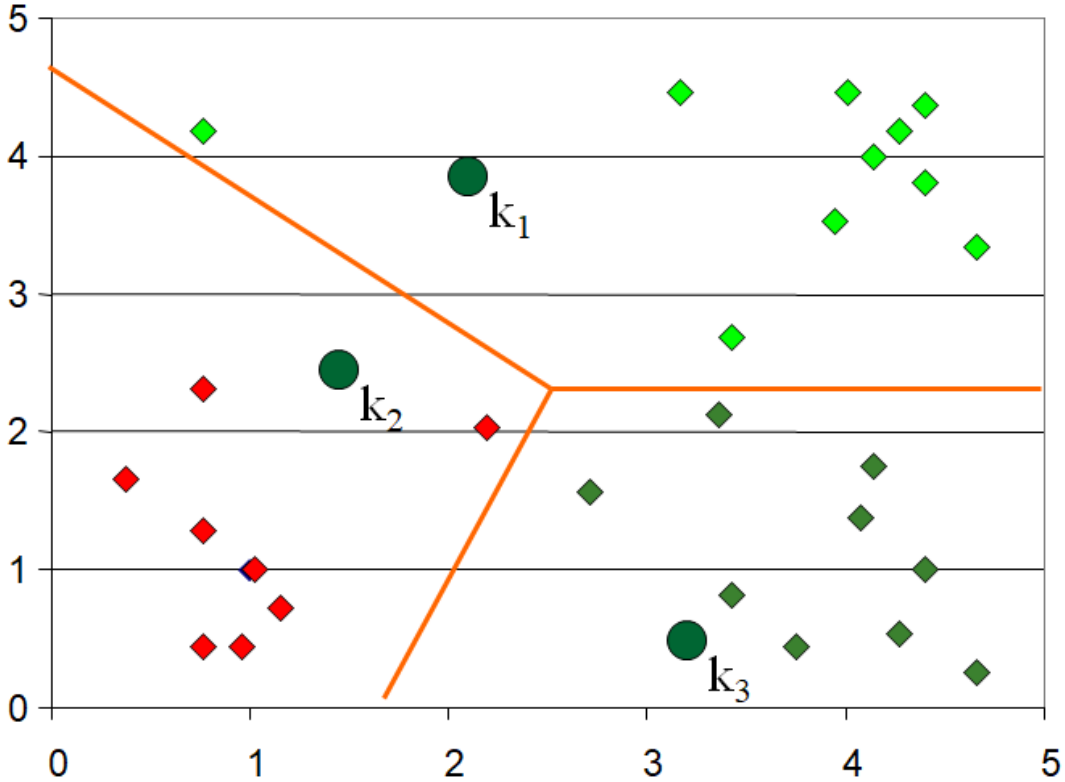
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



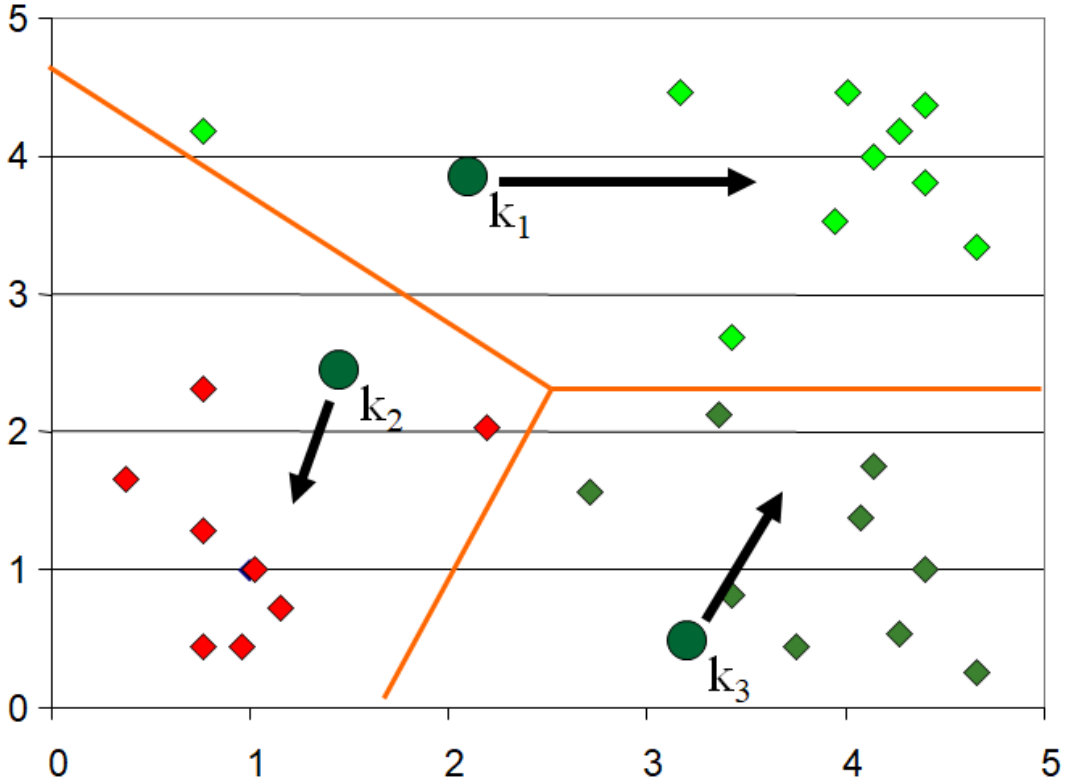
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



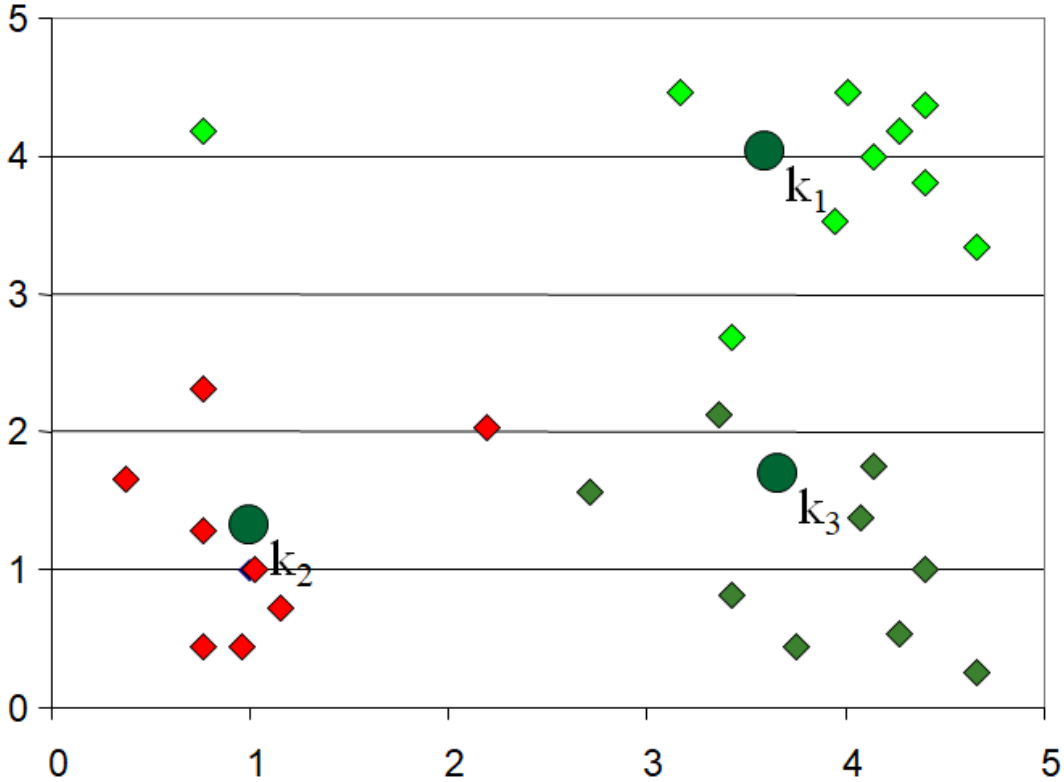
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



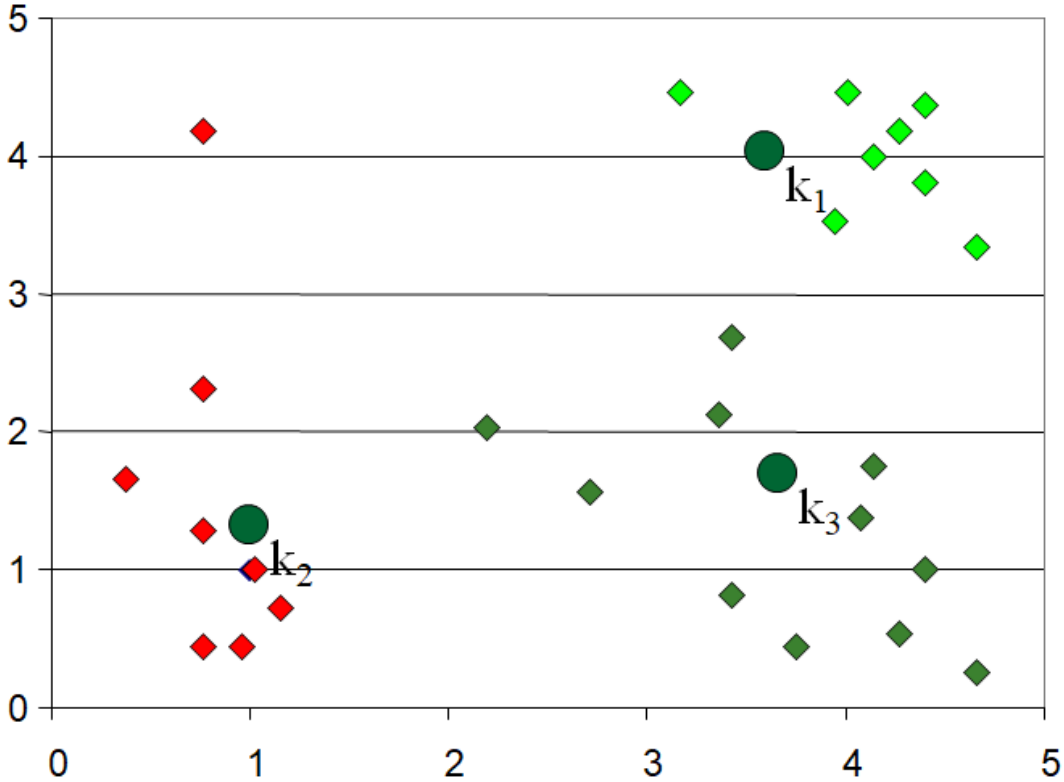
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



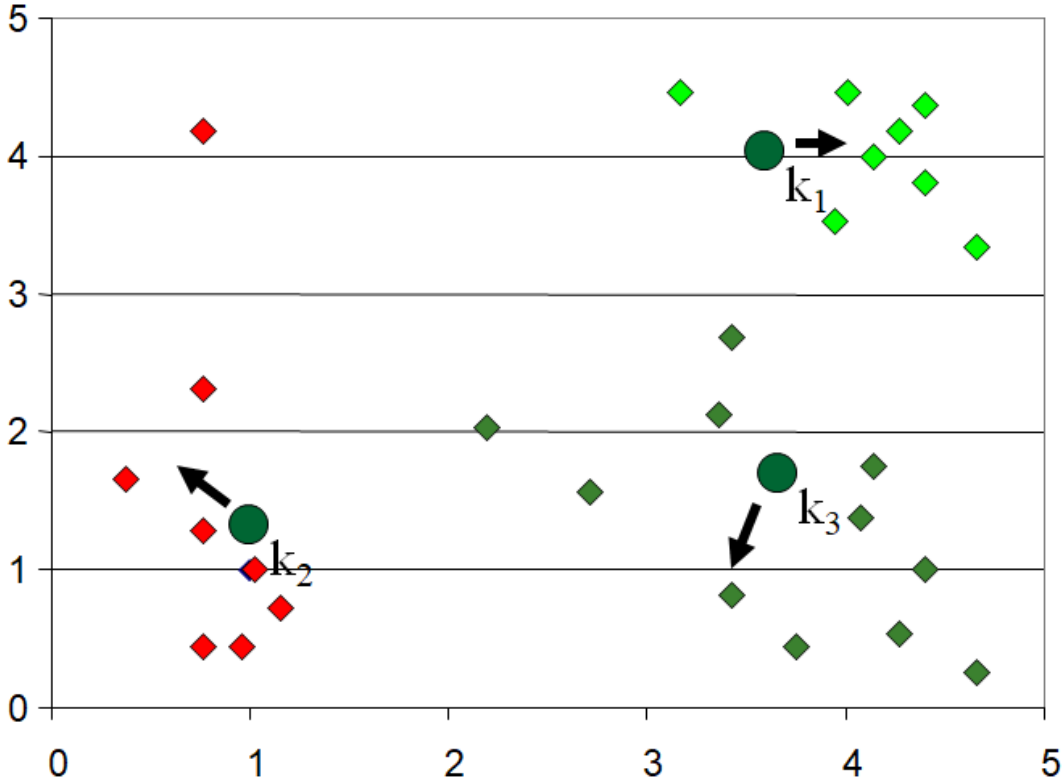
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



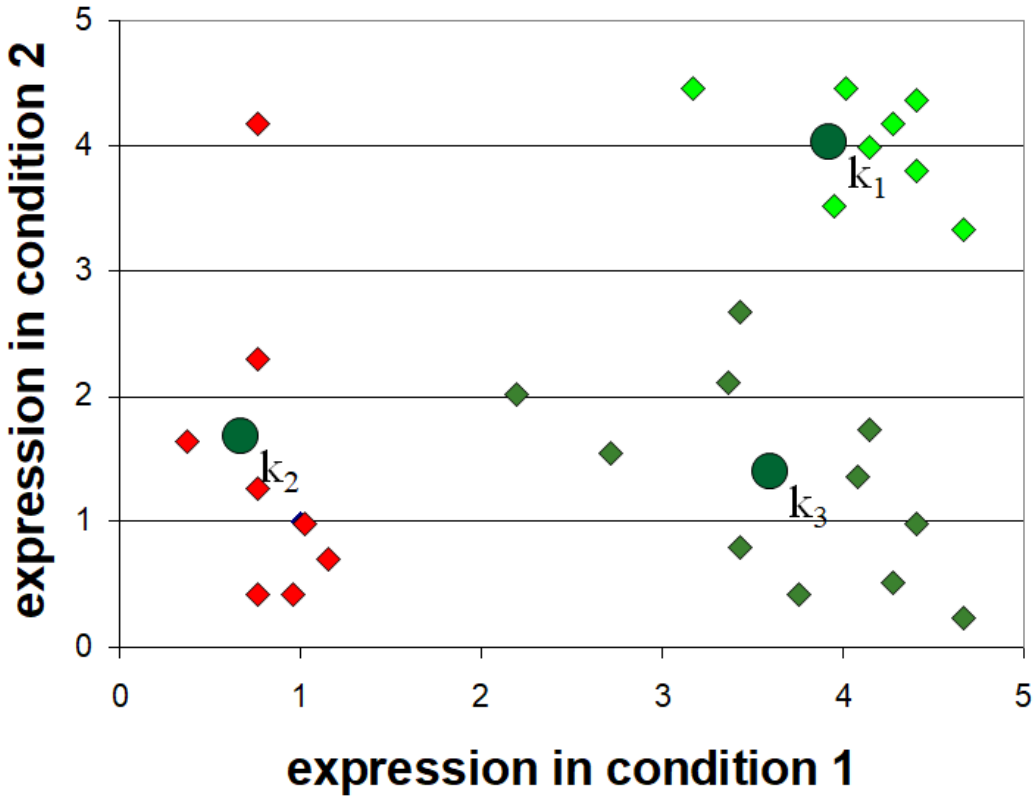
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



Hierarchical Clustering

Algorithm (Agglomerative Hierarchical Clustering)
Input: N data points x_i, $i = 1, \cdots, N$
Output? c clusters D_j, $j = 1, \cdots, c$

1. initialize c: desired number of clusters, $c_1 = n$, $D_i = x_i$ for $i = 1, \cdots, n$
2. $c_1 = c_1 - 1$
3. find nearest clusters, say, D_i and D_j
4. merge D_i and D_j
5. repeat from 2 until $c = c_1$
6. return c clusters

Hierarchical Clustering

Dendrogram

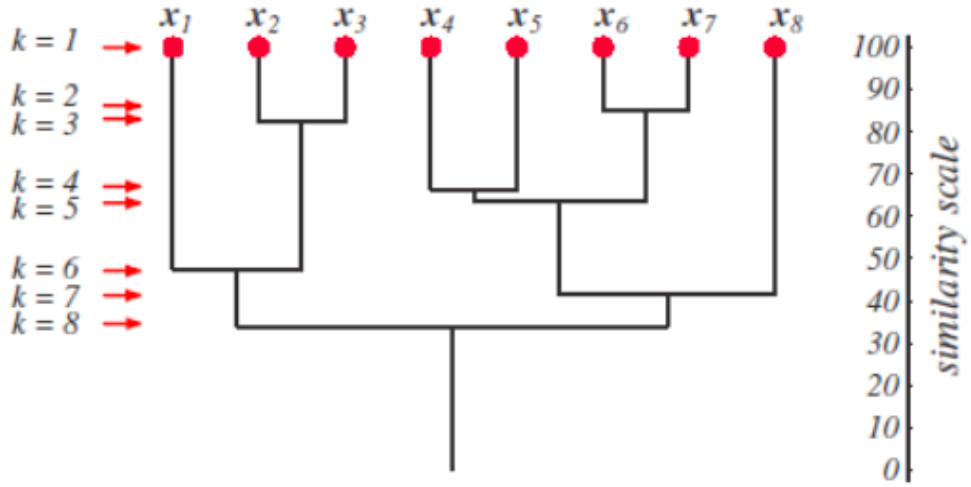


FIGURE 10.11. A dendrogram can represent the results of hierarchical clustering algorithms. The vertical axis shows a generalized measure of similarity among clusters. Here, at level 1 all eight points lie in singleton clusters; each point in a cluster is highly similar to itself, of course. Points $\mathbf{x}_6$ and $\mathbf{x}_7$ happen to be the most similar, and are merged at level 2, and so forth. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

The Nearest-Neighbor Algorithm

- If minimum distance between elements of two clusters is used, the method is called the nearest-neighbor cluster algorithm.
- If it is terminated when the distance between nearest clusters exceeds an arbitrary threshold, it is called the single-linkage algorithm.

The Nearest-Neighbor Algorithm

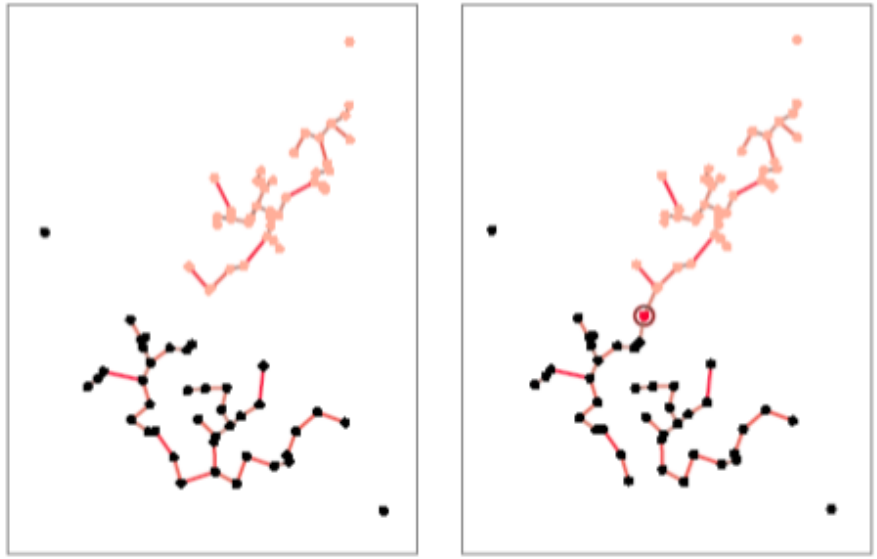


FIGURE 10.13. Two Gaussians were used to generate two-dimensional samples, shown in pink and black. The nearest-neighbor clustering algorithm gives two clusters that well approximate the generating Gaussians (left). If, however, another particular sample is generated (circled red point at the right) and the procedure is restarted, the clusters do not well approximate the Gaussians. This illustrates how the algorithm is sensitive to the details of the samples. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

The Farthest-Neighbor Algorithm

- If maximum distance between elements of two clusters is used, the method is called the farthest-neighbor cluster algorithm.
- If it is terminated when the distance between nearest clusters exceeds an arbitrary threshold, it is called the complete-linkage algorithm.

The Farthest-Neighbor Algorithm

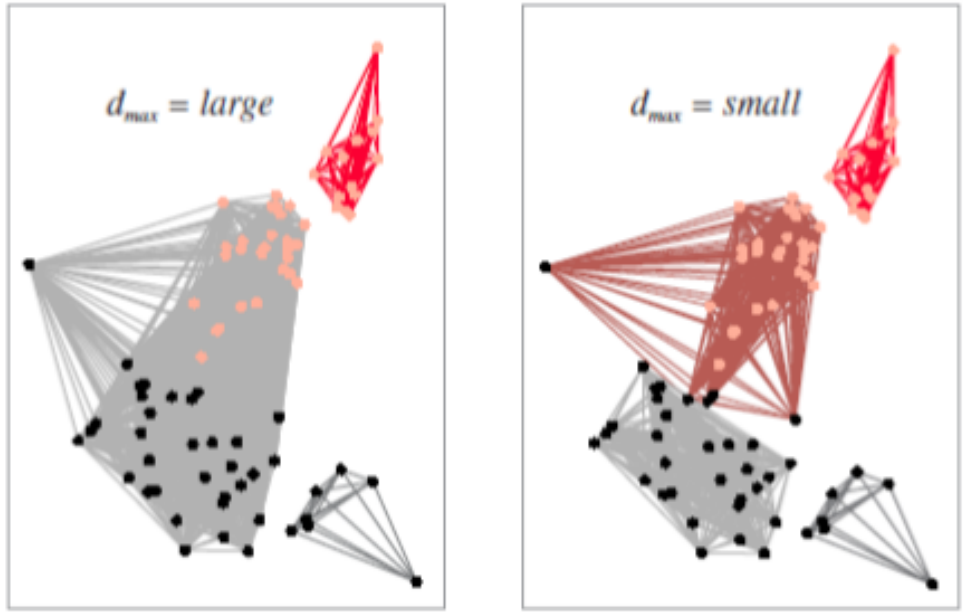


FIGURE 10.14. The farthest-neighbor clustering algorithm uses the separation between the most distant points as a criterion for cluster membership. If this distance is set very large, then all points lie in the same cluster. In the case shown at the left, a fairly large d_{max} leads to three clusters; a smaller d_{max} gives four clusters, as shown at the right. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Clustering Validation

- How to evaluate the performance of clutering?
- Assume that $X_1, \cdots, X_r$ are ground truth clusters (classes), and $Y_1, \cdots, Y_s$ are generated clusters.

Purity

$$Purity(X, Y) = \frac{1}{N} \sum_k \max_j |X_j \cap Y_k|$$

- Purity measure sums purities of all clusters.
- Problem: high purity is easy to achieve when the number of clusters is large.
- Extreme case: one item per cluster.

Normalized Mutual Information (NMI)

$$
\begin{aligned}
NMI(X, Y) &= \frac{2I(X; Y)}{H(X) + H(Y)} \\
I(X; Y) &= \sum_k \sum_j P(X_j \cap Y_k) \log \frac{P(X_j \cap Y_k)}{P(X_j)P(Y_k)} \\
&= \sum_k \sum_j \frac{|X_j \cap Y_k|}{N} \log \frac{N\,|X_j \cap Y_k|}{|X_j|\,|Y_k|} \\
H(X) &= -\sum_k P(X_k) \log P(X_k) \\
&= -\sum_k \frac{|X_k|}{N} \log \frac{|X_k|}{N}
\end{aligned}
$$

Rand Index

- For each pair of items (p, q),
- TP: number of pairs belong to X_i and Y_j
- TN: not belong to either any X_i or any Y_j
- FP: belong not to any X_i but to Y_j
- FN: belong to X_i but not to any Y_j

$$RI(X, Y) = \frac{TP + TN}{TP + FP + FN + TN}$$
$$= \frac{TP + TN}{N(N-1)/2}$$

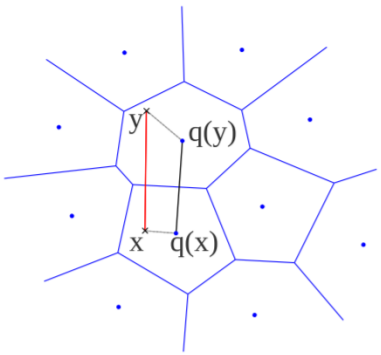
F measure

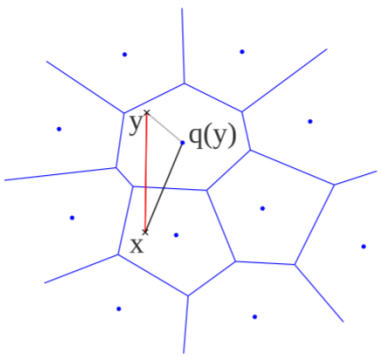
$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN}$$
$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

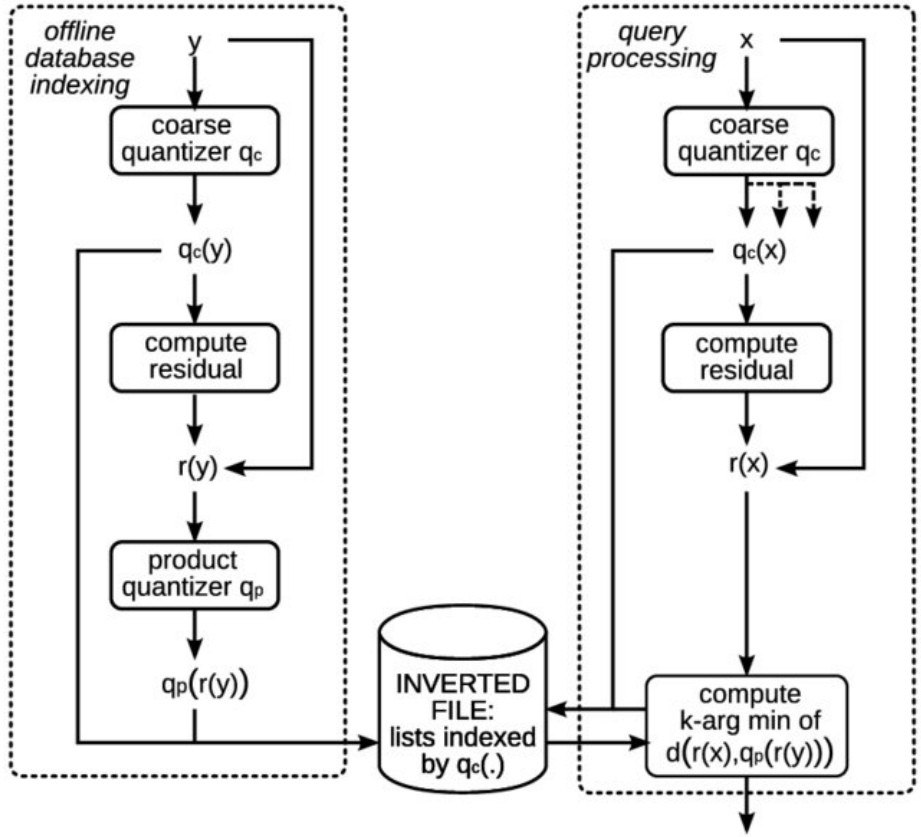
Application of clustering: Fast multi-dimensional data search

- We want to index large number of multi-dimensional vectors y
- Given query vector x we want to return vector(s) in the database nearest from the query
- Basic idea:
  - Coarsely divide vectors via clustering
  - Register each data to an entry corresponding to the cluster center the data belongs to
  - (Optional) Compute residuals of data from cluster centers and register the residuals to a similar structure with clustering
  - In search, inspect data which belong to the entry corresponding to the cluster center closest to x
  - (Optional) Inspect only data which have similar redisuals to the query

Application of clustering: Fast multi-dimensional data search







Jégou, Douze, Schmid, Searching with quantization: approximate nearest neighbor search using short codes and distance estimators, INRIA, 2009