SDU
SYDDANSK UNIVERSITET

Introduction to Computer Science

Melih Kandemir

University of Southern Denmark

DM534, Fall 2021

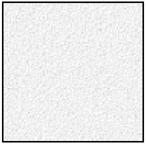
Outline

Outline

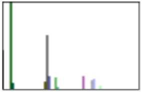
Outline

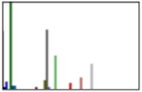
Categories of Feature Descriptors for Images

- distribution of colors
- texture
- shapes (contoures)



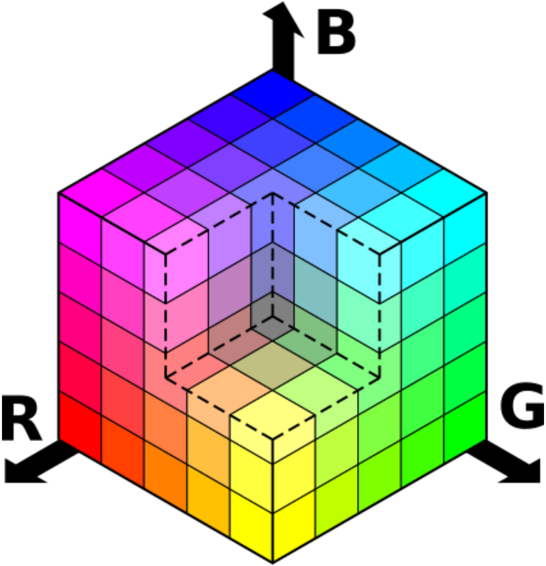
Color Histogram



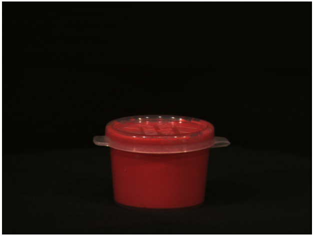
- a histogram represents the distribution of colors over the pixels of an image
- definition of an color histogram:
  - choose a color space (RGB, HSV, HLS, . . .)
  - choose number of representants (sample points) in the color space
  - possibly normalization (to account for different image sizes)

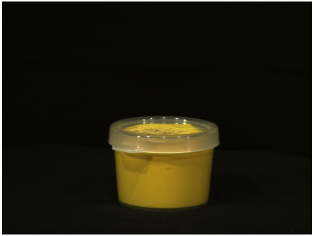
Color Space Example: RGB cube



Impact of Number of Representants

original images in full RGB space ($256^3 = 16,777,216$)





Impact of Number of Representants

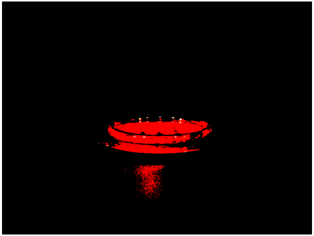
2^3

3^3

4^3

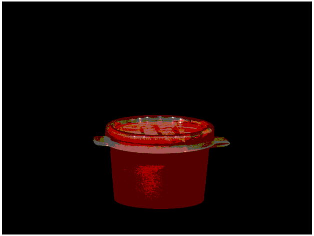
16^3

Impact of Number of Representants

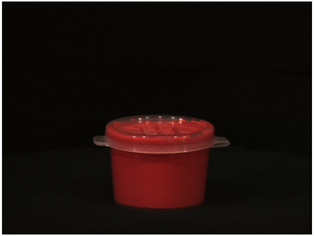


2^3

3^3

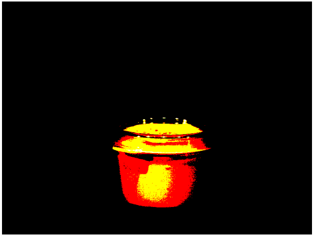


4^3

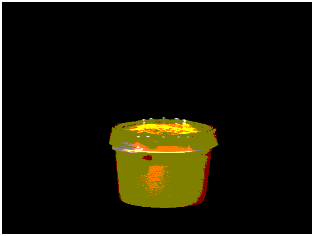


16^3

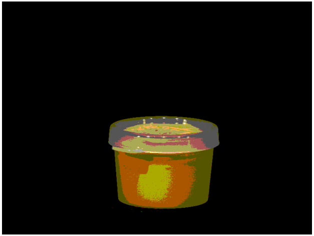
Impact of Number of Representants



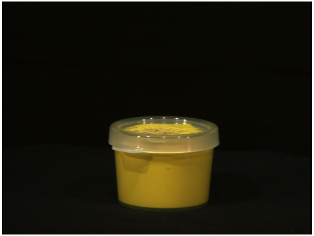
2^3



3^3



4^3



16^3

Impact of Number of Representants

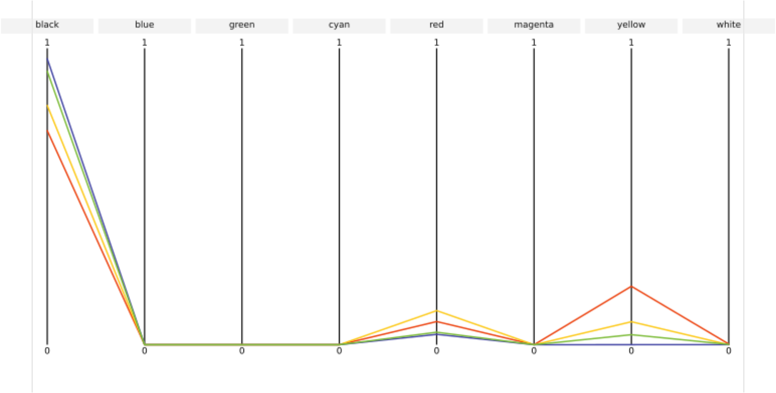
2^3

3^3

4^3

16^3

Impact of Number of Representants

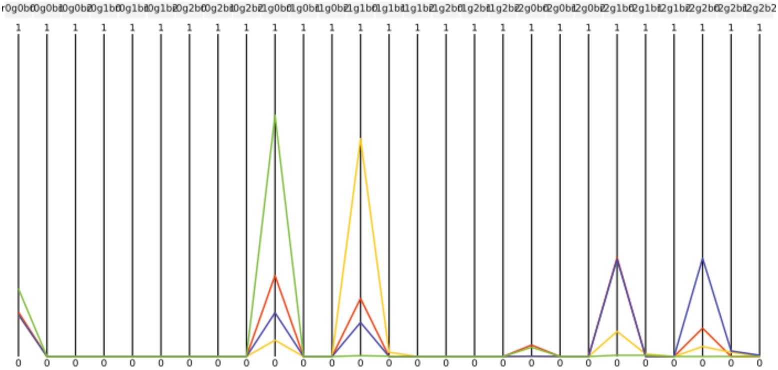


The histogram for each image is essentially a visualization of a vector:

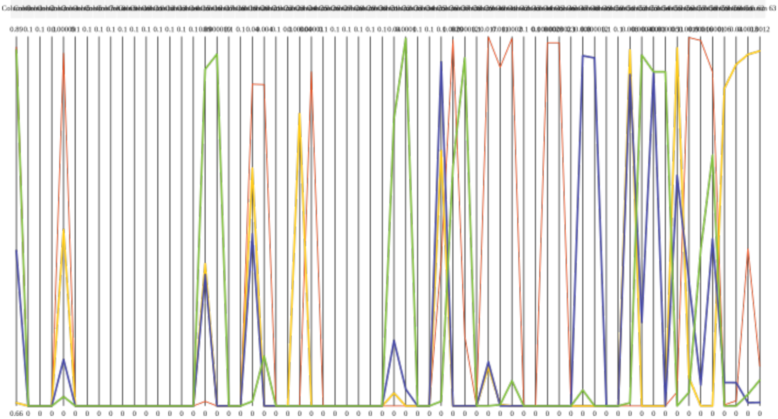
$(0.77, 0, 0, 0, 0.08, 0, 0.15, 0)$ $(0.8, 0, 0, 0, 0.11, 0, 0.09, 0)$

$(0.9, 0, 0, 0, 0.05, 0, 0.05, 0)$ $(0.955, 0, 0, 0, 0.045, 0, 0, 0)$

Impact of Number of Representants



Impact of Number of Representants

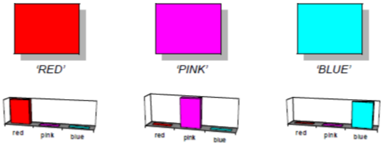


Outline

Distances for Color Histograms

Euclidean distance for images P and Q using the color histograms h_P and h_Q:

$$\mathrm{dist}(P, Q) = \sqrt{(h_P - h_Q) \cdot (h_P - h_Q)}$$



$(1, 0, 0)$ $(0, 1, 0)$ $(0, 0, 1)$

$$\mathrm{dist}(\text{RED}, \text{PINK}) = \sqrt{2}$$
$$\mathrm{dist}(\text{RED}, \text{BLUE}) = \sqrt{2}$$
$$\mathrm{dist}(\text{BLUE}, \text{PINK}) = \sqrt{2}$$

A 'psychologic' distance would consider that red is (in our perception) more similar to pink than to blue.

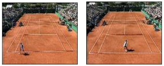
Example for the Distance Computation of Histograms

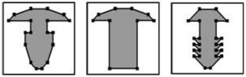
$$\mathrm{dist}(P, Q) = \sqrt{(h_P - h_Q) \cdot (h_P - h_Q)}$$

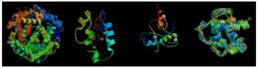
$$\begin{aligned}
\mathrm{dist}(\mathsf{RED}, \mathsf{PINK}) &= \sqrt{((1,0,0) - (0,1,0)) \cdot ((1,0,0) - (0,1,0))} \\
&= \sqrt{(1,-1,0) \cdot (1,-1,0)} \\
&= \sqrt{(1 \cdot 1 + (-1) \cdot (-1) + 0 \cdot 0)} \\
&= \sqrt{2}
\end{aligned}$$

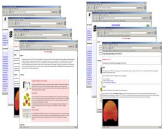
Similarity

- Similarity (as given by some distance measure) is a central concept in data mining, e.g.:
  - clustering: group similar objects in the same cluster, separate dissimilar objects to different clusters
  - outlier detection: identify objects that are dissimilar (by some characteristic) from most other objects
- definition of a suitable distance measure is often crucial for deriving a meaningful solution in the data mining task
  - images
  - CAD objects
  - proteins
  - texts
  - . . .







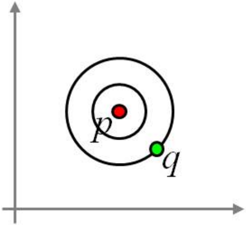


Spaces and Distance Functions

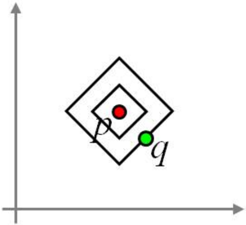
Common distance measure for (Euclidean) feature vectors: L_P-norm

$$\text{dist}_P(p,q) = \left(|p_1 - q_1|^P + |p_2 - q_2|^P + \ldots + |p_n - q_n|^P\right)^{\frac{1}{P}}$$

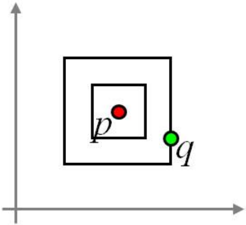
Euclidean norm (L_2):



Manhattan norm (L_1):



Maximum norm (L_∞, also: $L_{\max}$, supremum dist., Chebyshev dist.)



Weighted Euclidean norm:

$$\text{dist}(p,q) = \left(w_1|p_1 - q_1|^2 + w_2|p_2 - q_2|^2 + \ldots + w_n|p_n - q_n|^2\right)^{\frac{1}{2}}$$

*note that we assume vectors to be row vectors here

Outline

Color Histograms as Feature Spaces for Representation of Images
Features for Images
Distances
Summary

A First Glimpse on Clustering

Your Choice of a Distance Measure

There are hundreds of distance functions [Deza and Deza, 2009].

- For time series: DTW, EDR, ERP, LCSS, . . .
- For texts: Cosine and normalizations
- For sets – based on intersection, union, . . . (Jaccard)
- For clusters (single-link, average-link, etc.)
- For histograms: histogram intersection, “Earth movers distance”, quadratic forms with color similarity
- With normalization: Canberra, . . .

Note that:

Choosing the appropriate distance function can be seen as a part of “preprocessing”.

Summary

You learned in this section:

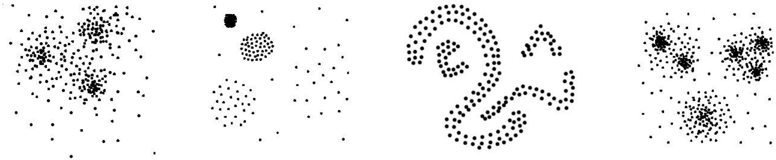
- *distances (L_p-norms, weighted norms)*
- *color histograms as feature (vector) descriptors for images*
- *impact of the granularity of color histograms on similarity measures*

Outline

Outline

Purpose of Clustering

- identify a finite number of categories (classes, groups: clusters) in a given dataset
- *similar* objects shall be grouped in the same cluster, *dissimilar* objects in different clusters
- "similarity" is highly subjective, depending on the application scenario



A Dataset can be Clustered in Different Meaningful Ways

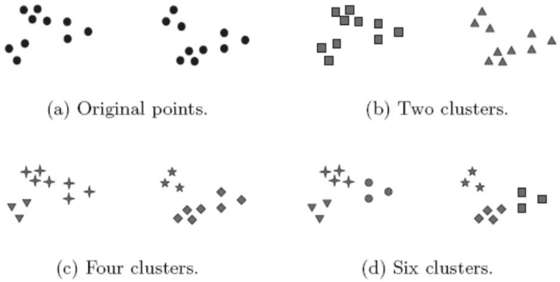


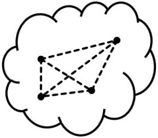
Figure 8.1. Different ways of clustering the same set of points.

(Figure from Tan et al. [2006].)

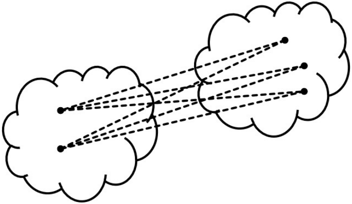
Outline

Criteria of Quality: Cohesion and Separation

- cohesion: how strong are the cluster objects connected (how similar, pairwise, to each other)?
- separation: how well is a cluster separated from other clusters?



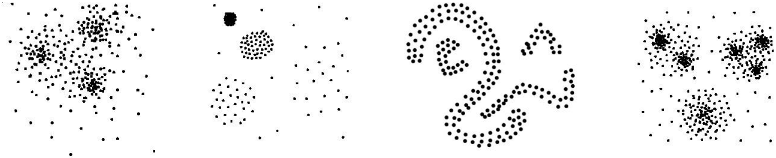
small within cluster distances



large between cluster distances

Optimization of Cohesion

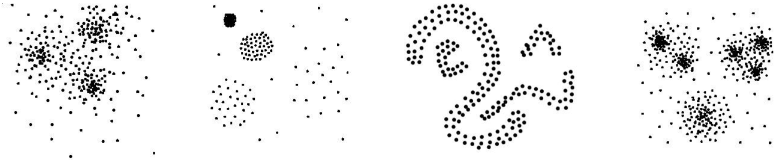
Partitional clustering algorithms partition a dataset into k clusters, typically minimizing some cost function (compactness criterion), i.e., optimizing cohesion.



Assumptions for Partitioning Clustering

Central assumptions for approaches in this family are typically:

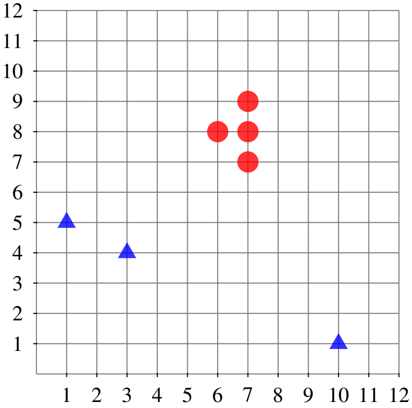
- number k of clusters known (i.e., given as input)
- clusters are characterized by their compactness
- compactness measured by some distance function (e.g., distance of all objects in a cluster from some cluster representative is minimal)
- criterion of compactness typically leads to convex or even spherically shaped clusters



Outline

Construction of Central Points: Basics

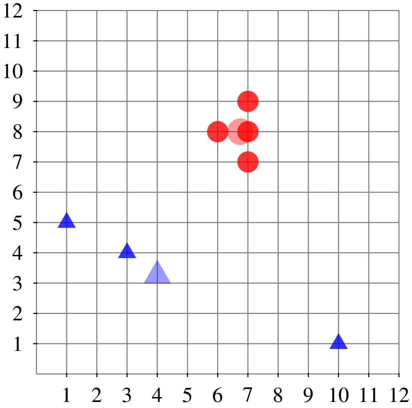
- objects are points $x = (x_1, \ldots, x_d)$ in Euclidean vector space $\mathbb{R}^d$, dist = Euclidean distance (L_2)
- centroid μ_C: mean vector of all points in cluster C



$$\mu_{C_i} = \frac{1}{|C_i|} \cdot \sum_{o \in C_i} o$$

Construction of Central Points: Basics

- objects are points $x = (x_1, \ldots, x_d)$ in Euclidean vector space $\mathbb{R}^d$, dist = Euclidean distance (L_2)
- centroid μ_C: mean vector of all points in cluster C



$$\mu_{C_i} = \frac{1}{|C_i|} \cdot \sum_{o \in C_i} o$$

Construction of Central Points: Basics

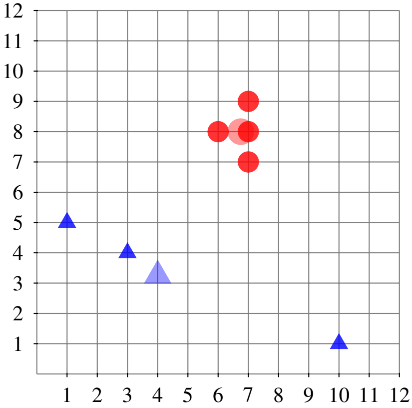
- measure of compactness for a cluster C:

$$TD^2(C) = \sum_{p \in C} \mathrm{dist}(p, \mu_C)^2$$

(a.k.a. SSQ: sum of squares)

- measure of compactness for a clustering

$$TD^2(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} TD^2(C_i)$$



Construction of Central Points: Basics

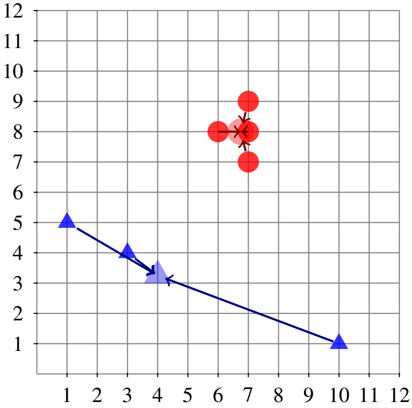
- measure of compactness for a cluster C:

$$TD^2(C) = \sum_{p \in C} \text{dist}(p, \mu_C)^2$$

(a.k.a. SSQ: sum of squares)

- measure of compactness for a clustering

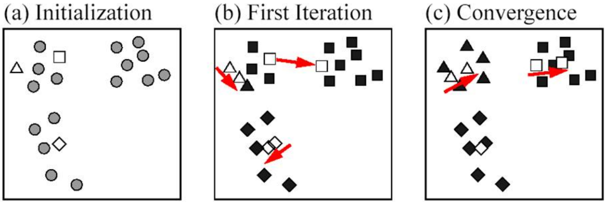
$$TD^2(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} TD^2(C_i)$$



Basic Algorithm [Forgy, 1965, Lloyd, 1982]

Algorithm 2.1 (Clustering by Minimization of Variance)

- *start with k (e.g., randomly selected) points as cluster representatives (or with a random partition into k "clusters")*
- *repeat:*
  - *assign each point to the closest representative*
  - *compute new representatives based on the given partitions (centroid of the assigned points)*
- *until there is no change in assignment*



k-means

k-means [MacQueen, 1967] is a variant of the basic algorithm:

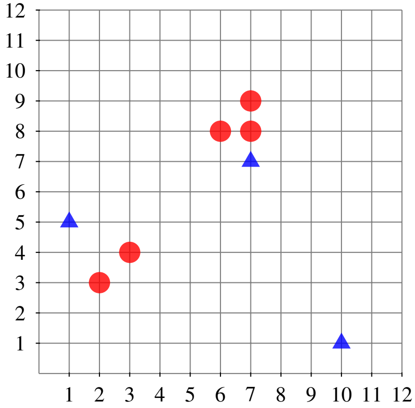
- a centroid is immediately updated when some point changes its assignment
- k-means has very similar properties, but the result now depends on the order of data points in the input file

Note that:

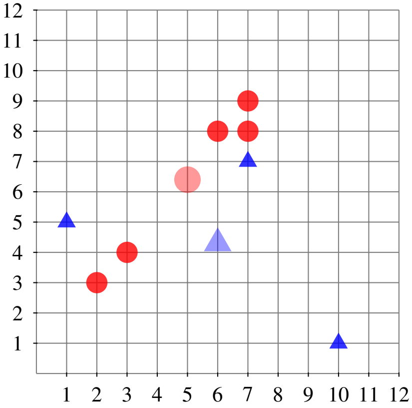
The name “k-means” is often used indifferently for any variant of the basic algorithm, in particular also for Algorithm 2.1 [Forgy, 1965, Lloyd, 1982].

Outline

k-means Clustering – Lloyd/Forgy Algorithm



k-means Clustering – Lloyd/Forgy Algorithm

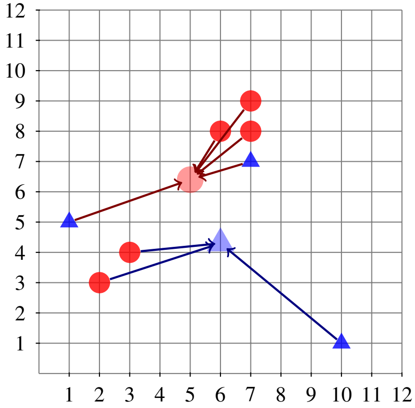


recompute centroids:

$\mu \approx (6.0, 4.3)$

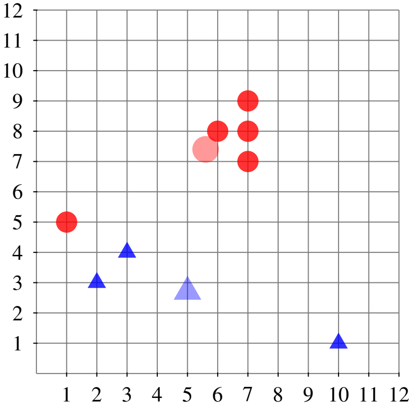
$\mu \approx (5.0, 6.4)$

k-means Clustering – Lloyd/Forgy Algorithm



reassign points

k-means Clustering – Lloyd/Forgy Algorithm

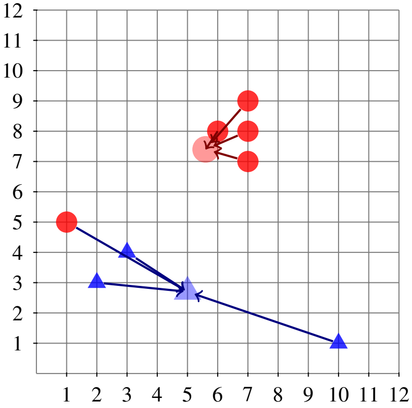


recompute centroids:

$\mu \approx (5.0, 2.7)$

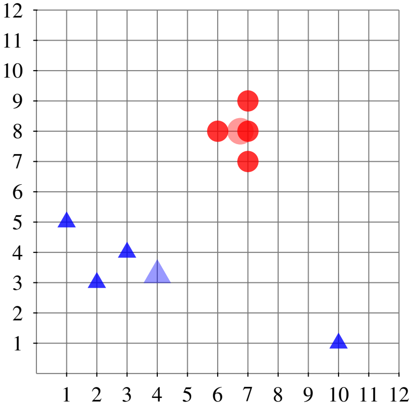
$\mu \approx (5.6, 7.4)$

k-means Clustering – Lloyd/Forgy Algorithm



reassign points

k-means Clustering – Lloyd/Forgy Algorithm

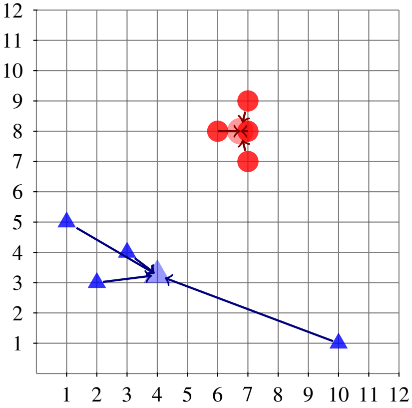


recompute centroids:

$\mu \approx (4.0, 3.25)$

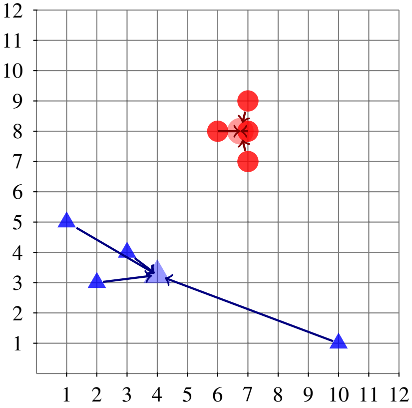
$\mu \approx (6.75, 8.0)$

k-means Clustering – Lloyd/Forgy Algorithm



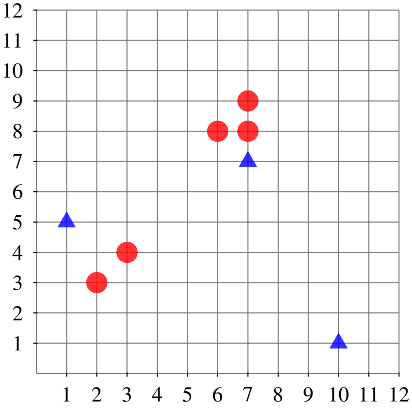
reassign points

k-means Clustering – Lloyd/Forgy Algorithm

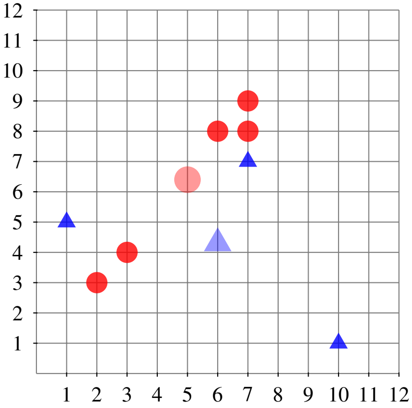


reassign points
no change
convergence!

k-means Clustering – MacQueen Algorithm



k-means Clustering – MacQueen Algorithm

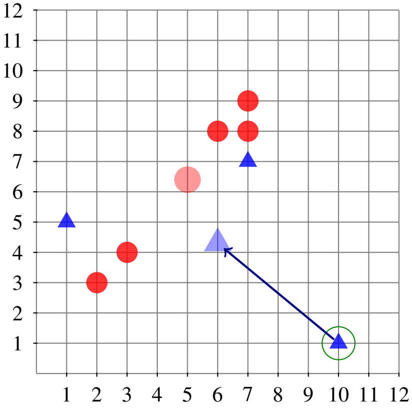


Centroids (e.g.: from previous iteration):

$$\mu \approx (6.0, 4.3)$$

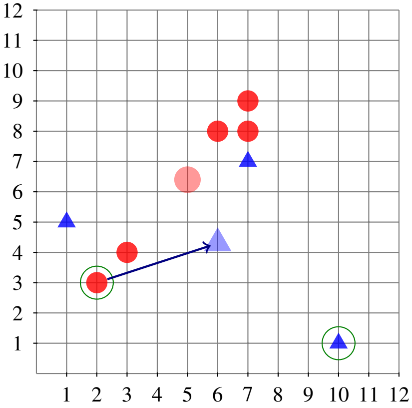
$$\mu \approx (5.0, 6.4)$$

k-means Clustering – MacQueen Algorithm



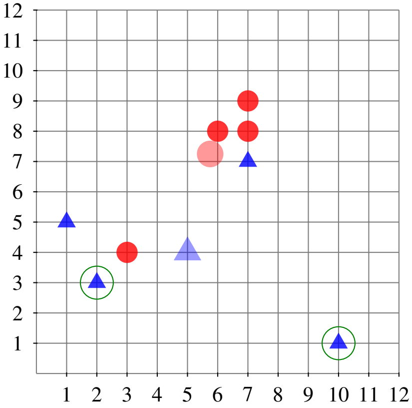
assign first point

k-means Clustering – MacQueen Algorithm



assign second point

k-means Clustering – MacQueen Algorithm

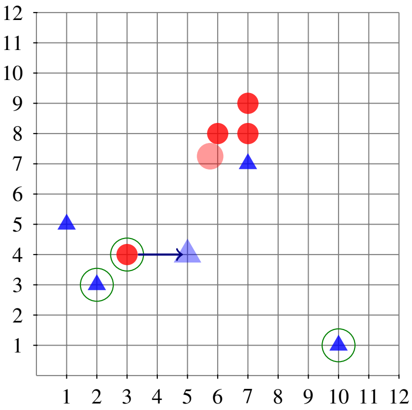


recompute centroids:

$\mu \approx (5.0, 4.0)$

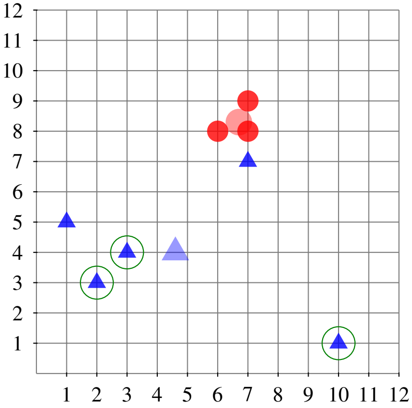
$\mu \approx (5.75, 7.25)$

k-means Clustering – MacQueen Algorithm



assign third point

k-means Clustering – MacQueen Algorithm

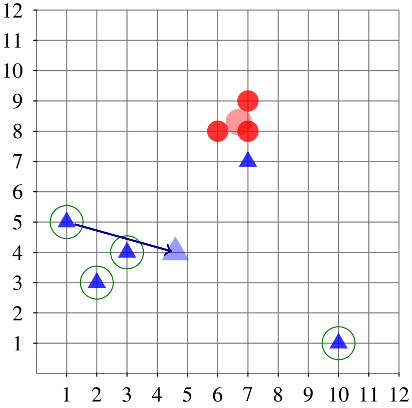


recompute centroids:

$\mu \approx (4.6, 4.0)$

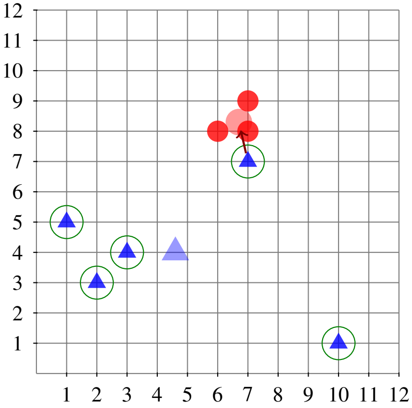
$\mu \approx (6.7, 8.3)$

k-means Clustering – MacQueen Algorithm



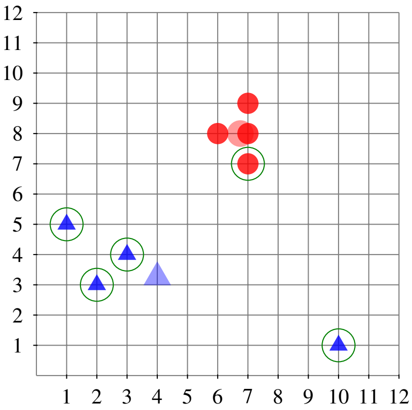
assign fourth point

k-means Clustering – MacQueen Algorithm



assing fifth point

k-means Clustering – MacQueen Algorithm

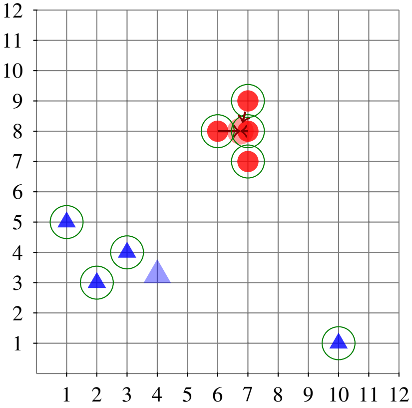


recompute centroids:

$\mu \approx (4.0, 3.25)$

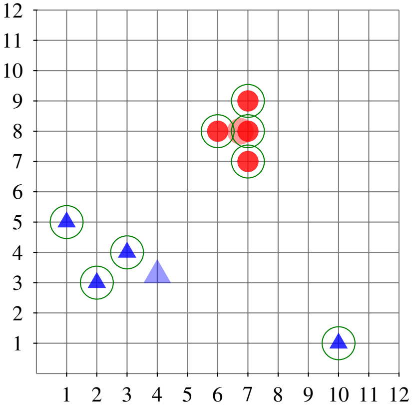
$\mu \approx (6.75, 8.0)$

k-means Clustering – MacQueen Algorithm



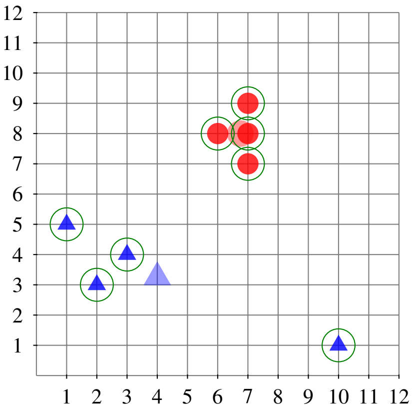
reassign more points

k-means Clustering – MacQueen Algorithm



reassign more points
possibly more iterations

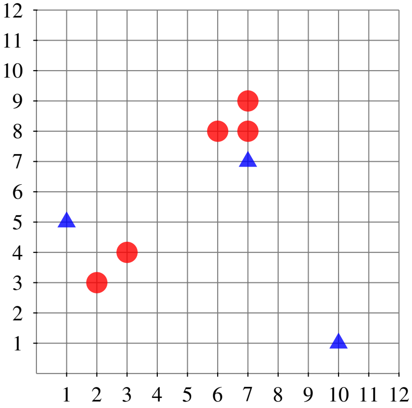
k-means Clustering – MacQueen Algorithm



reassign more points
possibly more iterations
convergence

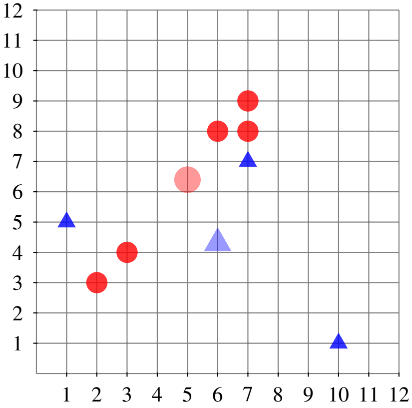
k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



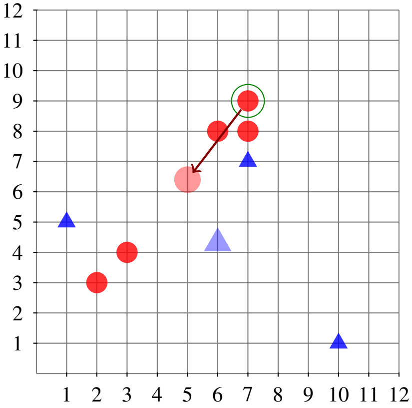
Centroids (e.g.: from previous iteration):

$\mu \approx (6.0, 4.3)$

$\mu \approx (5.0, 6.4)$

k-means Clustering – MacQueen Algorithm

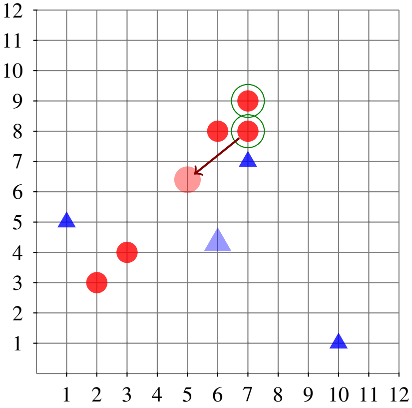
Alternative Run – Different Order



assign first point

k-means Clustering – MacQueen Algorithm

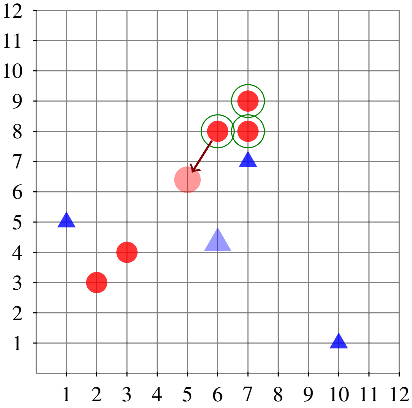
Alternative Run – Different Order



assign second point

k-means Clustering – MacQueen Algorithm

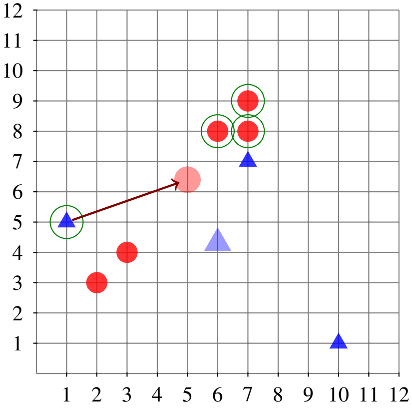
Alternative Run – Different Order



assign third point

k-means Clustering – MacQueen Algorithm

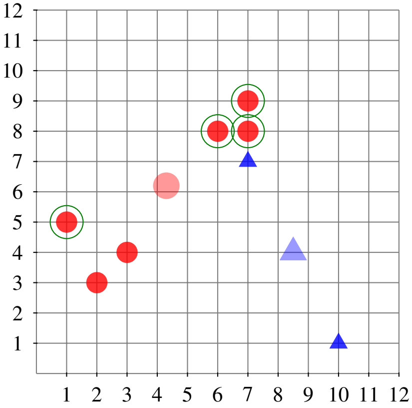
Alternative Run – Different Order



assign fourth point

k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



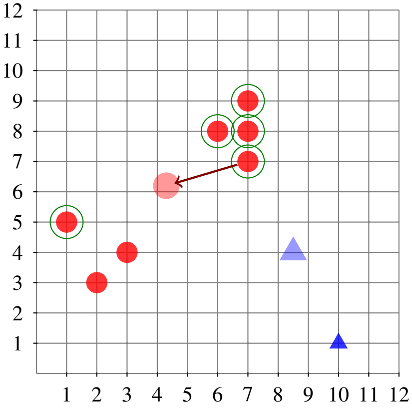
recompute centroids:

$\mu \approx (4.0, 8.5)$

$\mu \approx (4.3, 6.2)$

k-means Clustering – MacQueen Algorithm

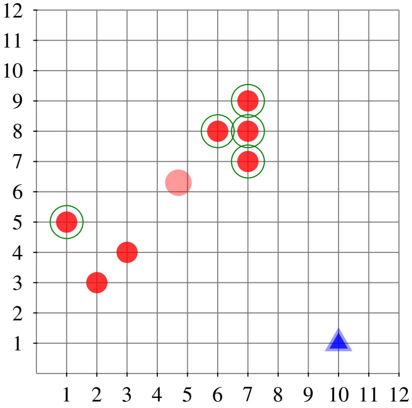
Alternative Run – Different Order



assign fifth point

k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



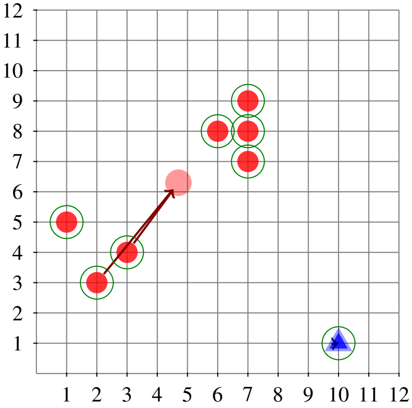
recompute centroids:

$\mu \approx (10.0, 1.0)$

$\mu \approx (4.7, 6.3)$

k-means Clustering – MacQueen Algorithm

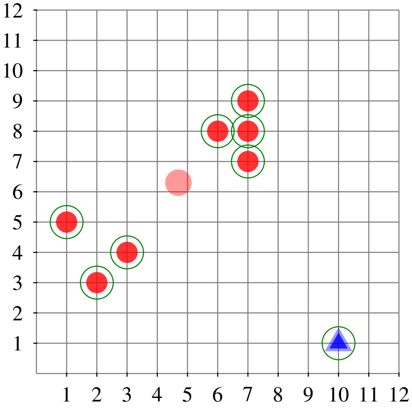
Alternative Run – Different Order



reasign more points

k-means Clustering – MacQueen Algorithm

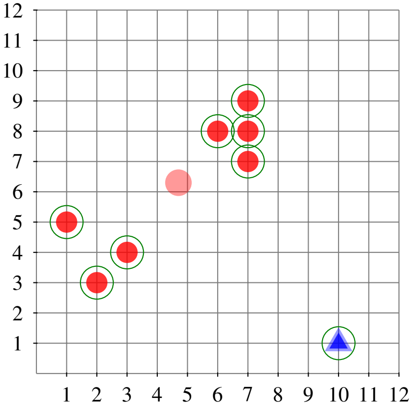
Alternative Run – Different Order



reasign more points
possibly more iterations

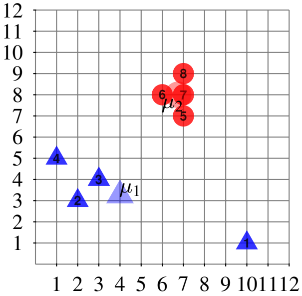
k-means Clustering – MacQueen Algorithm

Alternative Run – Different Order



reasign more points
possibly more iterations
convergence

k-means Clustering – Quality



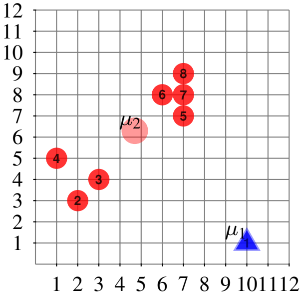
$$SSQ(\mu_1, p_1) = |4-10|^2 + |3.25-1|^2 = 36 + 5\tfrac{1}{16} = 41\tfrac{1}{16}$$
$$SSQ(\mu_1, p_2) = |4-2|^2 + |3.25-3|^2 = 4 + \tfrac{1}{16} = 4\tfrac{1}{16}$$
$$SSQ(\mu_1, p_3) = |4-3|^2 + |3.25-4|^2 = 1 + \tfrac{9}{16} = 1\tfrac{9}{16}$$
$$SSQ(\mu_1, p_4) = |4-1|^2 + |3.25-5|^2 = 9 + 3\tfrac{1}{16} = 12\tfrac{1}{16}$$
$$TD^2(C_1) = 58\tfrac{3}{4}$$

$$SSQ(\mu_2, p_5) = |6.75-7|^2 + |8-7|^2 = \tfrac{1}{16} + 1 = 1\tfrac{1}{16}$$
$$SSQ(\mu_2, p_6) = |6.75-6|^2 + |8-8|^2 = \tfrac{9}{16} + 0 = \tfrac{9}{16}$$
$$SSQ(\mu_2, p_7) = |6.75-7|^2 + |8-8|^2 = \tfrac{1}{16} + 0 = \tfrac{1}{16}$$
$$SSQ(\mu_2, p_8) = |6.75-7|^2 + |8-9|^2 = \tfrac{1}{16} + 1 = 1\tfrac{1}{16}$$
$$TD^2(C_2) = 2\tfrac{3}{4}$$

First solution: $TD^2 = 61\frac{1}{2}$

Note: $SSQ(\mu, p) = \mathit{Euclidean}(\mu, p)^2 = L_2^2(\mu, p)$.

k-means Clustering – Quality



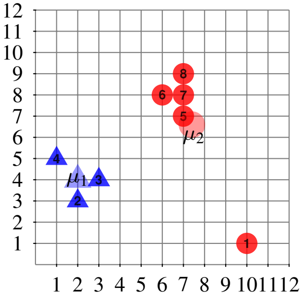
$SSQ(\mu_1, p_1) = |10 - 10|^2 + |1 - 1|^2 = 0$
$TD^2(C_1) = 0$

$SSQ(\mu_2, p_2) \approx |4.7 - 2|^2 + |6.3 - 3|^2 \approx 18.2$
$SSQ(\mu_2, p_3) \approx |4.7 - 3|^2 + |6.3 - 4|^2 \approx 8.2$
$SSQ(\mu_2, p_4) \approx |4.7 - 1|^2 + |6.3 - 5|^2 \approx 15.4$
$SSQ(\mu_2, p_5) \approx |4.7 - 7|^2 + |6.3 - 7|^2 \approx 5.7$
$SSQ(\mu_2, p_6) \approx |4.7 - 6|^2 + |6.3 - 8|^2 \approx 4.6$
$SSQ(\mu_2, p_7) \approx |4.7 - 7|^2 + |6.3 - 8|^2 \approx 8.2$
$SSQ(\mu_2, p_7) \approx |4.7 - 7|^2 + |6.3 - 9|^2 \approx 12.6$
$TD^2(C_2) \approx 72.86$

First solution: $TD^2 = 61\frac{1}{2}$
Second solution: $TD^2 \approx 72.68$

Note: $SSQ(\mu, p) = \textit{Euclidean}(\mu, p)^2 = L_2^2(\mu, p)$.

k-means Clustering – Quality



$SSQ(\mu_1, p_2) = |2-2|^2 + |4-3|^2 = 0 + 1 = 1$
$SSQ(\mu_1, p_3) = |2-3|^2 + |4-4|^2 = 1 + 0 = 1$
$SSQ(\mu_1, p_4) = |2-1|^2 + |4-5|^2 = 1 + 1 = 2$
$TD^2(C_1) = 4$

$SSQ(\mu_2, p_1) = |7.4-10|^2 + |6.6-1|^2 = 6\frac{19}{25} + 31\frac{9}{25} = 38\frac{3}{25}$
$SSQ(\mu_2, p_5) = |7.4-7|^2 + |6.6-7|^2 = \frac{4}{25} + \frac{4}{25} = \frac{8}{25}$
$SSQ(\mu_2, p_6) = |7.4-6|^2 + |6.6-8|^2 = 1\frac{24}{25} + 1\frac{24}{25} = 3\frac{23}{25}$
$SSQ(\mu_2, p_7) = |7.4-7|^2 + |6.6-8|^2 = \frac{4}{25} + 1\frac{24}{25} = 2\frac{3}{25}$
$SSQ(\mu_2, p_8) = |7.4-7|^2 + |6.6-9|^2 = \frac{4}{25} + 5\frac{19}{25} = 5\frac{23}{25}$
$TD^2(C_2) = 50\frac{2}{5}$

First solution: $TD^2 = 61\frac{1}{2}$
Second solution: $TD^2 \approx 72.68$
Optimal solution: $TD^2 = 54\frac{2}{5}$

Note: $SSQ(\mu, p) = \mathit{Euclidean}(\mu, p)^2 = L_2^2(\mu, p)$.

Outline

Discussion

pros

- efficient: $\mathcal{O}(k \cdot n)$ per iteration, number of iterations is usually in the order of 10.
- easy to implement, thus very popular

cons

- k-means converges towards a *local* minimum
- k-means (MacQueen-variant) is order-dependent
- deteriorates with noise and outliers (all points are used to compute centroids)
- clusters need to be convex and of (more or less) equal extension
- number k of clusters is hard to determine
- strong dependency on initial partition (in result quality as well as runtime)

Summary

You learned in this section:

- *What is Clustering?*
- *Basic idea for identifying "good" partitions into k clusters*
- *selection of representative points*
- *iterative refinement*
- *local optimum*
- *k-means variants [Forgy, 1965, Lloyd, 1982, MacQueen, 1967]*

References I

M. M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 3rd edition, 2009. ISBN 9783662443415.

E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769, 1965.

S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982. doi: 10.1109/TIT.1982.1056489.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematics, Statistics, and Probabilistics*, volume 1, pages 281–297, 1967.

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.