DM534

Introduction to Computer Science

Fall 2019

Clustering and Feature Spaces

About Me

- Richard Roettger:
  - Computer Science (Technical University of Munich and thesis at the ICSI at the University of California at Berkeley)
  - PhD at the Max Planck Institute for Computer Science in Saarbrücken
  - Since 2014: Assistant Professor at SDU

- **Research Interests:**
  - **Bioinformatics**
  - **Machine Learning**
  - **Clustering**
  - **Biological Networks**

- **Part of the Slides are taken from Arthur Zimek**

Clustering & Feature Spaces
Learning Objectives

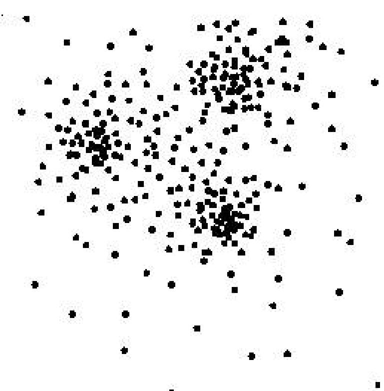
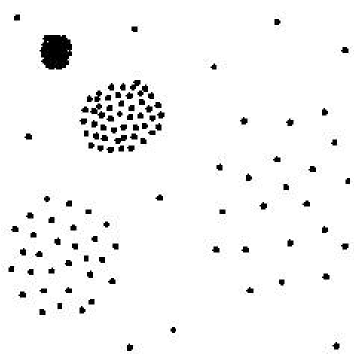
- Understand the problem of clustering in general
- Learn about k-means
- Understand the importance of feature spaces and object representation
- Understand the influence of distance functions

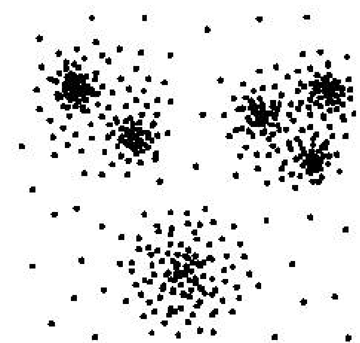
Clustering & Feature Spaces
Lecture Content

- Clustering
  - Clustering in General
  - Partitional Clustering
  - Visualization: Algorithmic Differences
  - Summary
- Feature Spaces
  - Distances
  - Features for Images
  - Summary

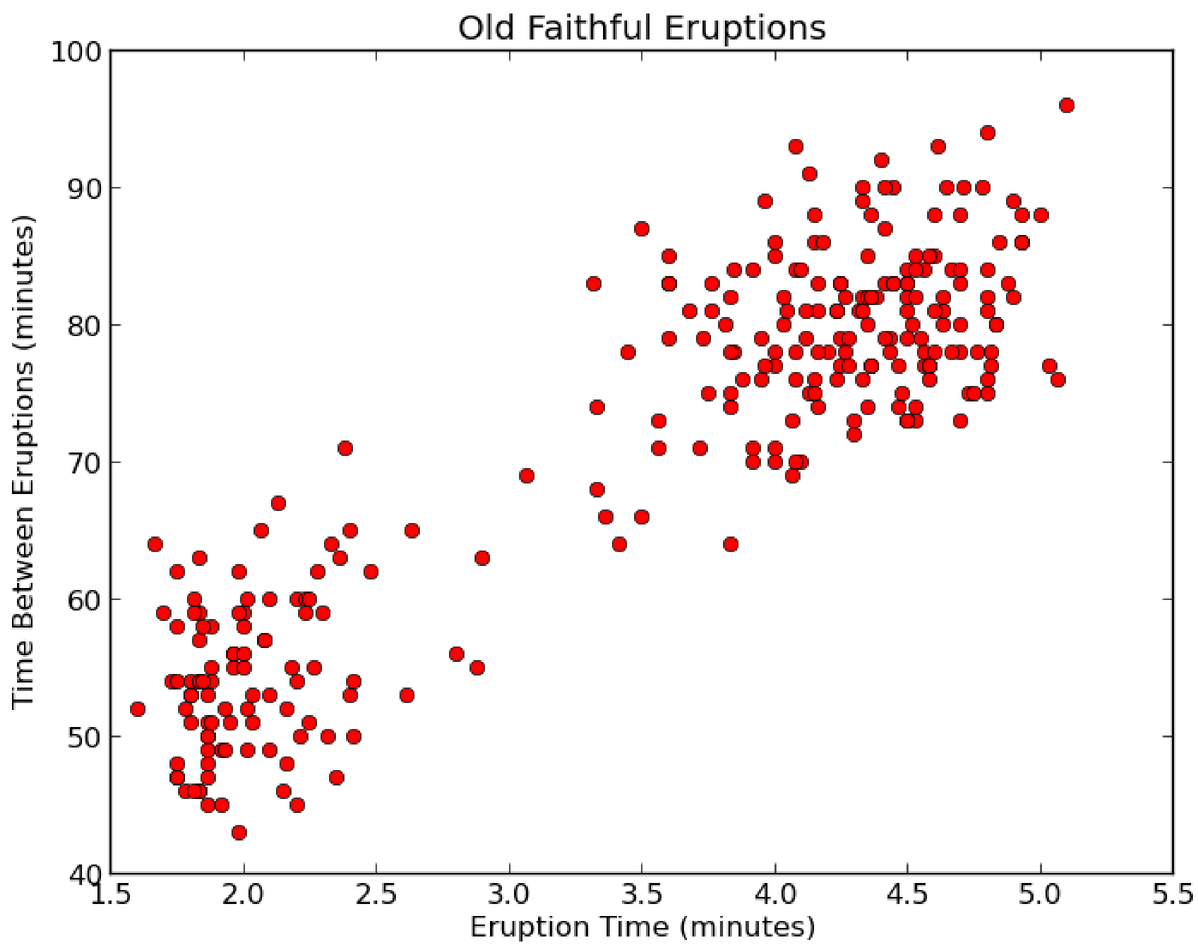
Purpose of Clustering

- 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

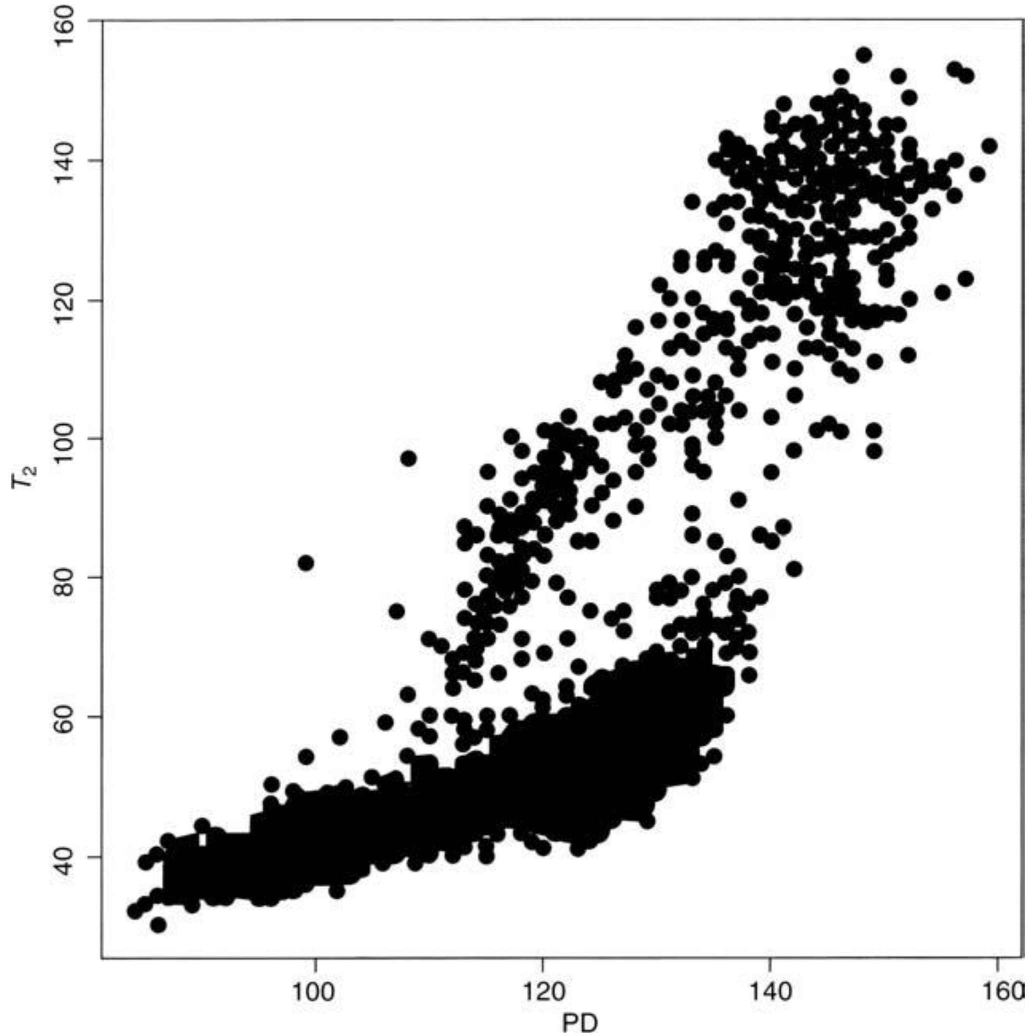


How Many Clusters?



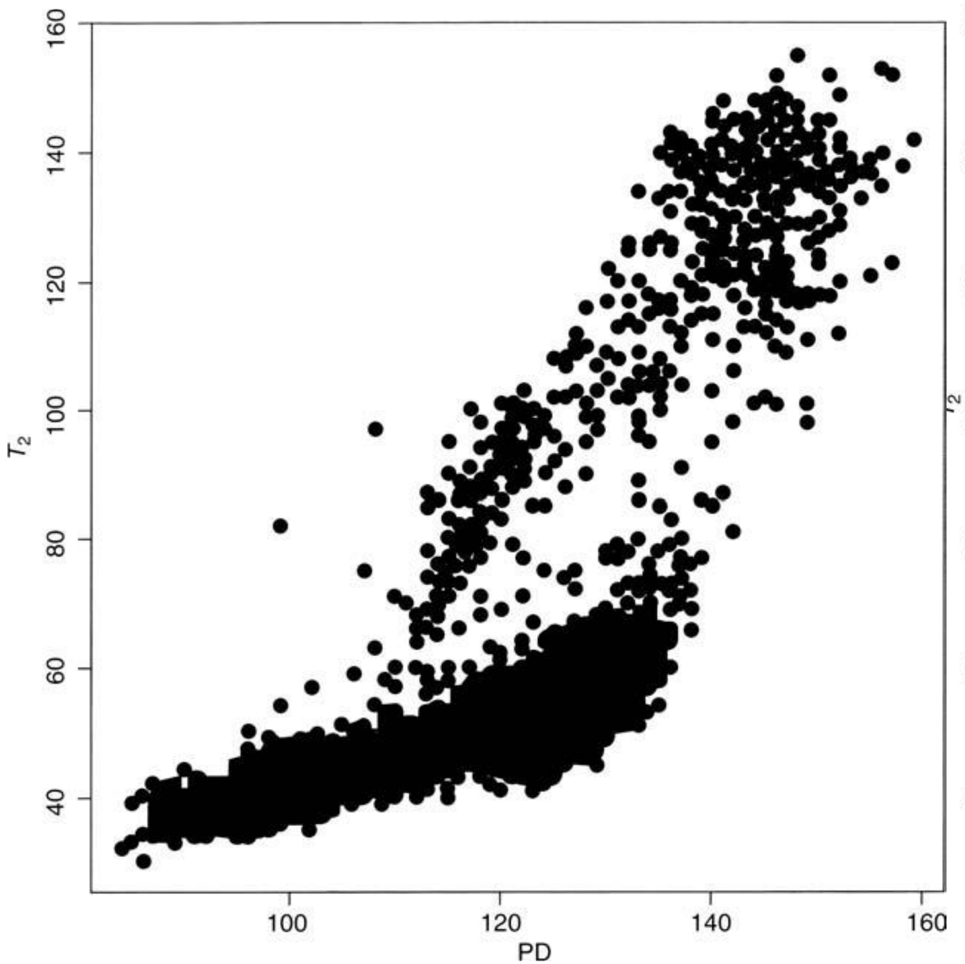
- Image taken from: http://fromdatawithlove.thegovans.us/

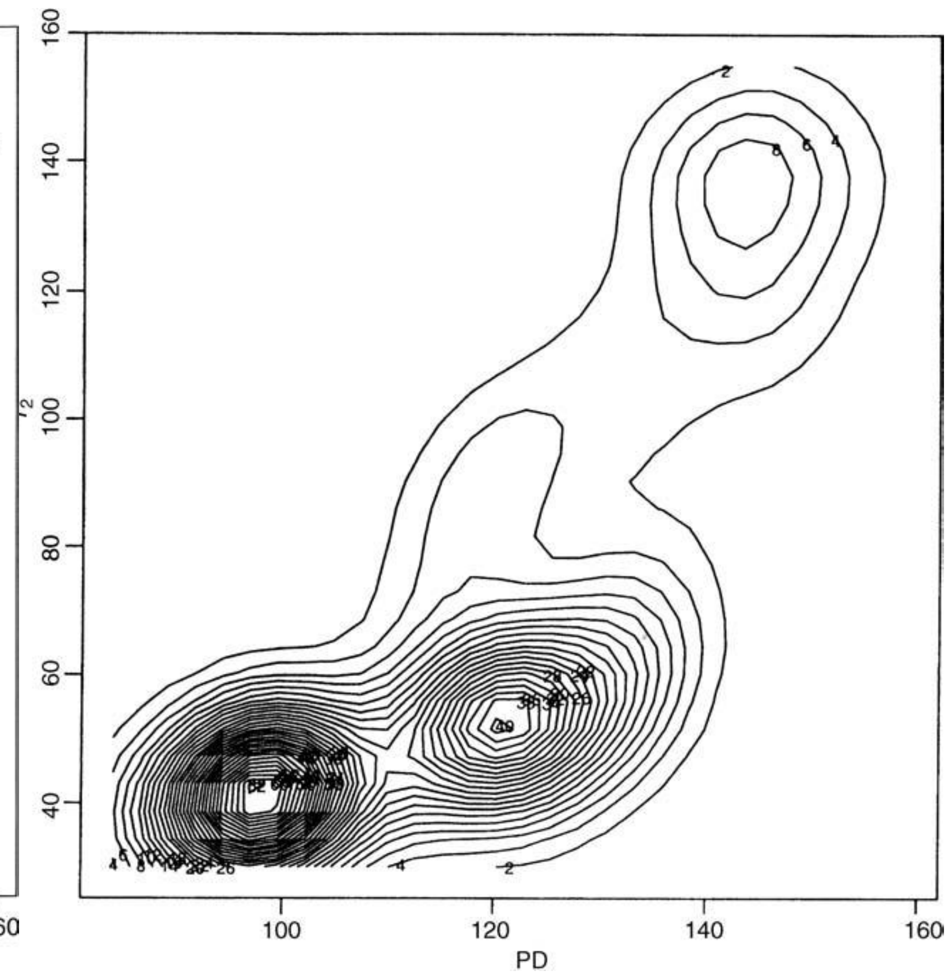
How Many Clusters?



➢ Everitt, Brian S., et al. “Cluster Analysis”, 5th Edition (2011).

How Many Clusters?





- Everitt, Brian S., et al. “Cluster Analysis”, 5th Edition (2011).

A Seemingly Simple Problem

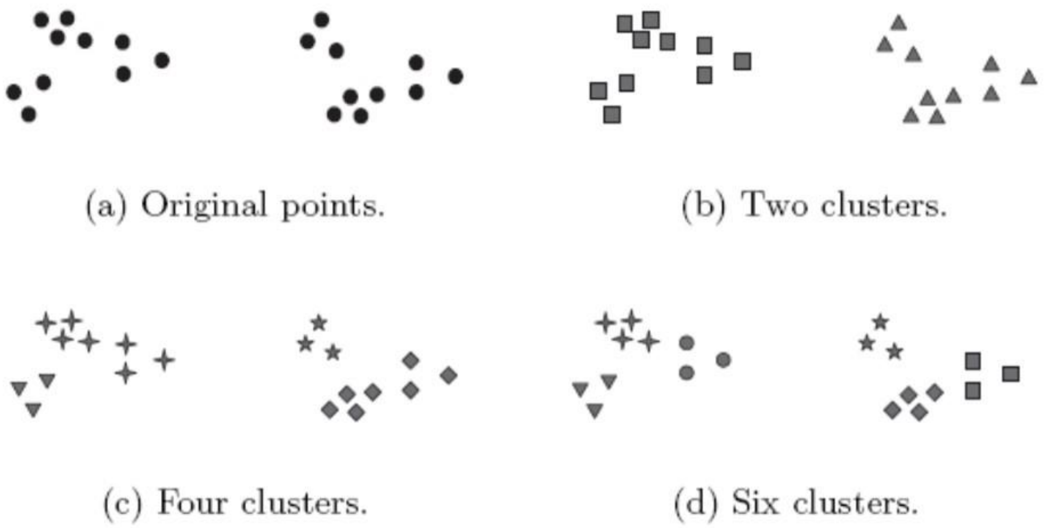


Figure 8.1. Different ways of clustering the same set of points.

- Each dataset can be clustered in many meaningful ways
  - Highly problem depended
  - Not known by the algorithm a priori

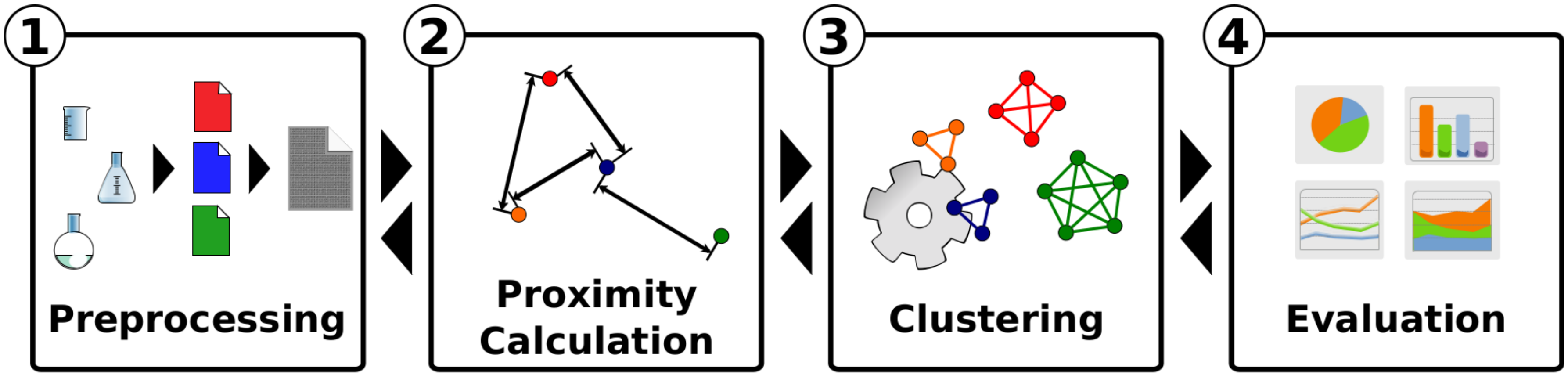
➢ Figure from Tan et al. [2006].

About Clustering

“Clustering is the unsupervised machine-learning task of “grouping or segmenting a collection of objects into subsets or ’clusters’ such that those within each cluster are more closely related to one another than objects assigned to different clusters.“

- What is related? For example Customers?
  - Age?
  - Behavior?
  - Kinship?
- Treatment of Outliers?
- Ill-posed Problem ...
  - That means there exist multiple solutions
  - What is the best?

Overview of a Cluster Analysis



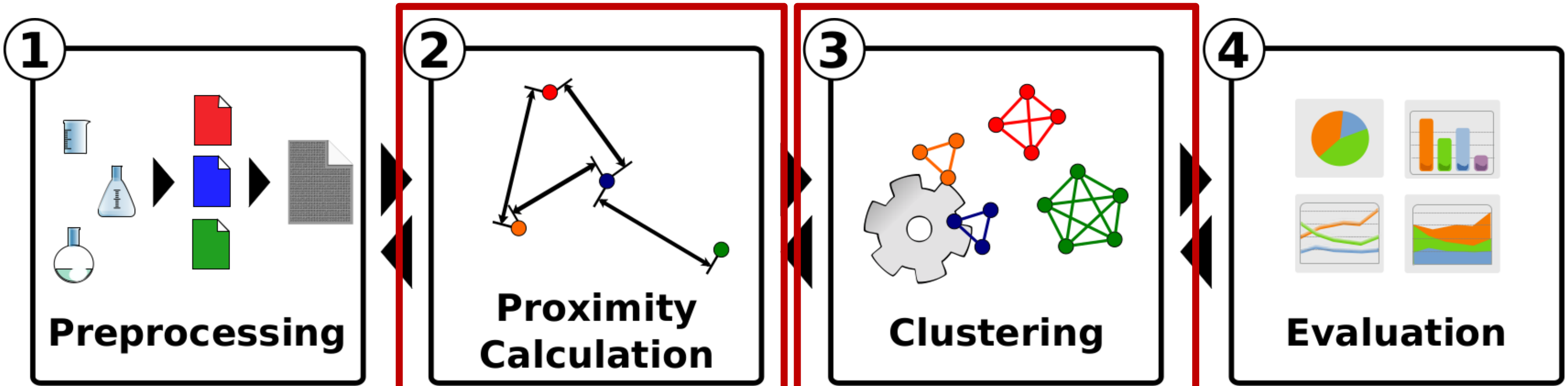
Preprocessing
- Feature Selection
- Feature Extraction
- Normalization Standardization
- Missing Values

Proximity Calculation
- Continous Data
- Categorical Data
- Mixed Data
- Specialized Measures

Clustering
- k-Means-Based
- Hierarchical
- Graph-Based
- Density-Based
- Model-Based
- Large-Scale

Evaluation
- Internal Indices
- External Indices
- Relative Indices
- Parameter Optimization

Overview of a Cluster Analysis



Preprocessing
- Feature Selection
- Feature Extraction
- Normalization Standardization
- Missing Values

Proximity Calculation
- Continous Data
- Categorical Data
- Mixed Data
- Specialized Measures

Clustering
- k-Means-Based
- Hierarchical
- Graph-Based
- Density-Based
- Model-Based
- Large-Scale

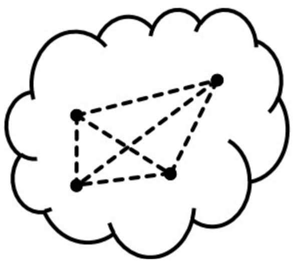
Evaluation
- Internal Indices
- External Indices
- Relative Indices
- Parameter Optimization

Clustering & Feature Spaces
Lecture Content

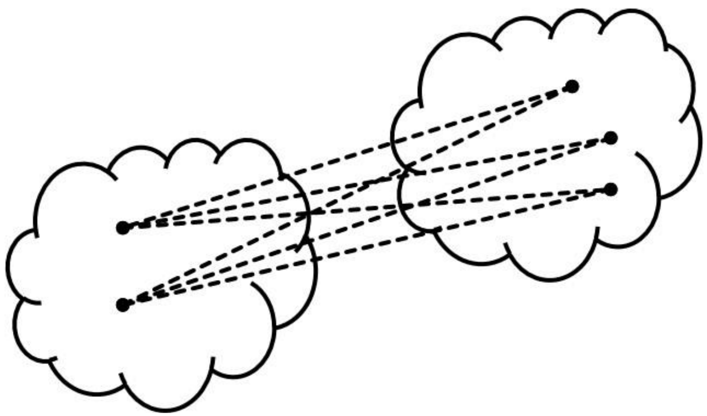
- Clustering
  - Clustering in General
  - Partitional Clustering
  - Visualization: Algorithmic Differences
  - Summary
- Feature Spaces
  - Distances
  - Features for Images
  - Summary

Steps to Automatization: Cluster Criteria

- **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 distance



large between cluster distance

Steps to Automatization: Cluster Criteria

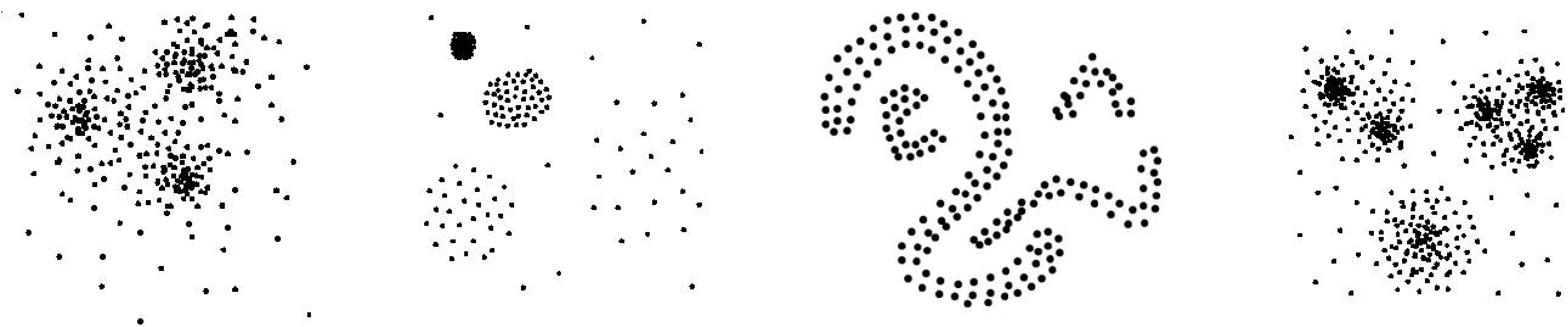
- **Cohesion**: how strong are the cluster objects connected (how similar, pairwise, to each other)?
- **Separation**: how well is a cluster separated from other clusters?

- **There exist many other criteria, e.g., areas with the same density.**
- **It is important to choose a criterion which fits the data!**

distance **distance**

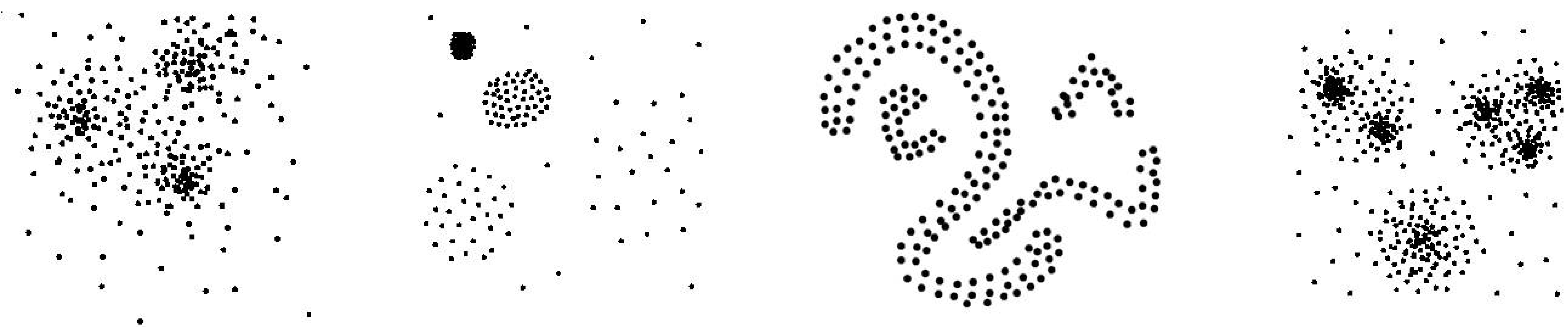
Optimization

- Partitional clustering algorithms partition a dataset into k clusters, typically minimizing some cost function
  - no overlaps
  - all points must be part of a cluster
- (compactness criterion), i.e., optimizing cohesion.



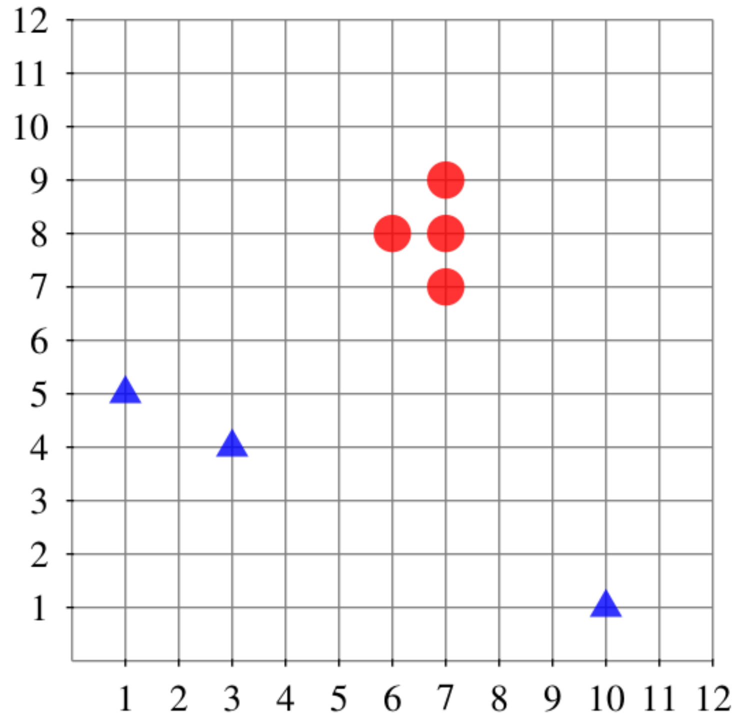
Assumptions for Partitioning Clustering

- 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



Construction of Central Points: Basics

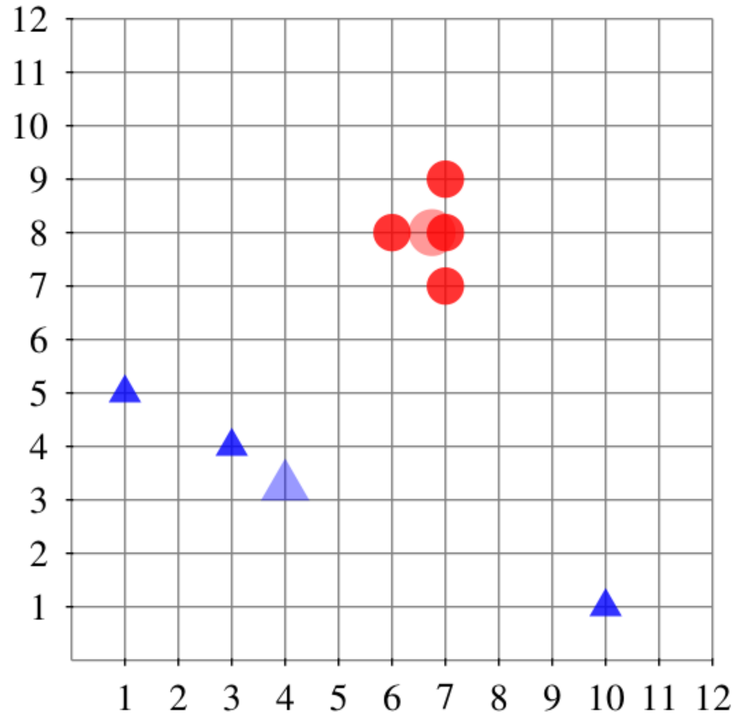
- objects are points $x = (x_1, \dots, x_d)$ in Euclidean vector space $\mathbb{R}^d$
- dist = Euclidean distance (L_2)
- I centroid μ_C: mean vector of all points in cluster C



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

Construction of Central Points: Basics

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



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

Cluster Criteria

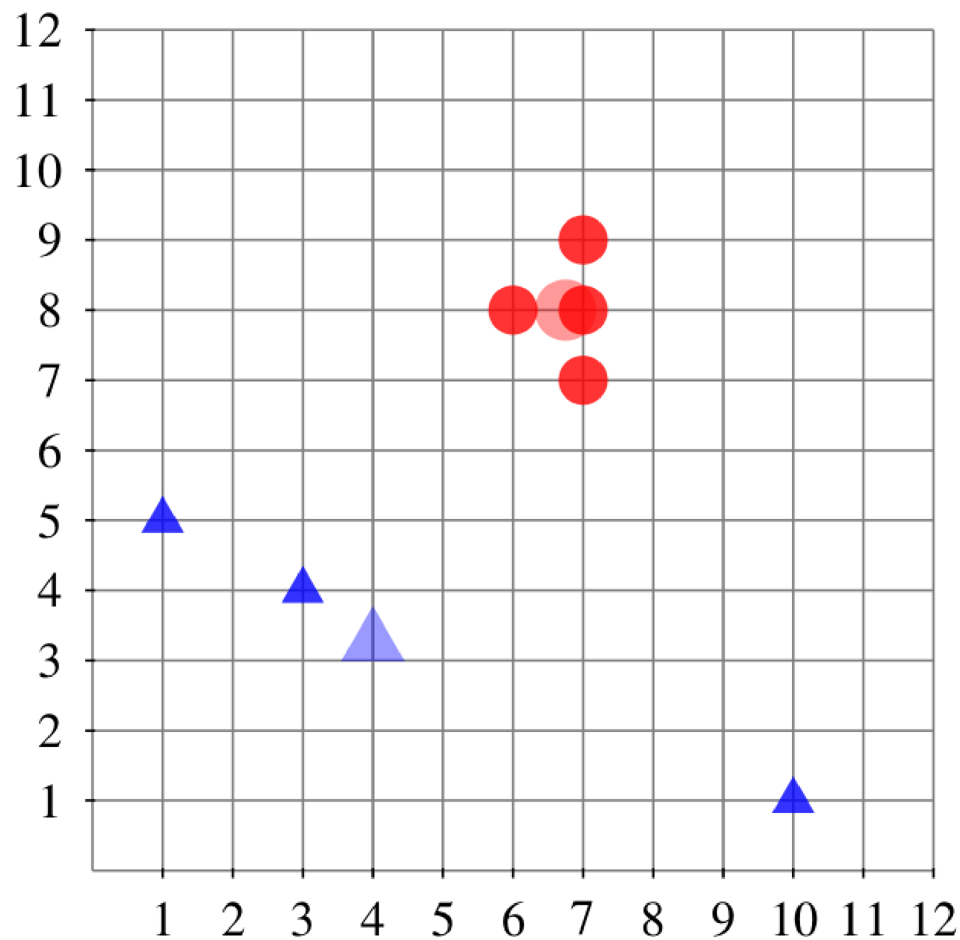
- Measure for compactness:

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

(sum of squares)

- Measure of compactness for a clustering:

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



Cluster Criteria

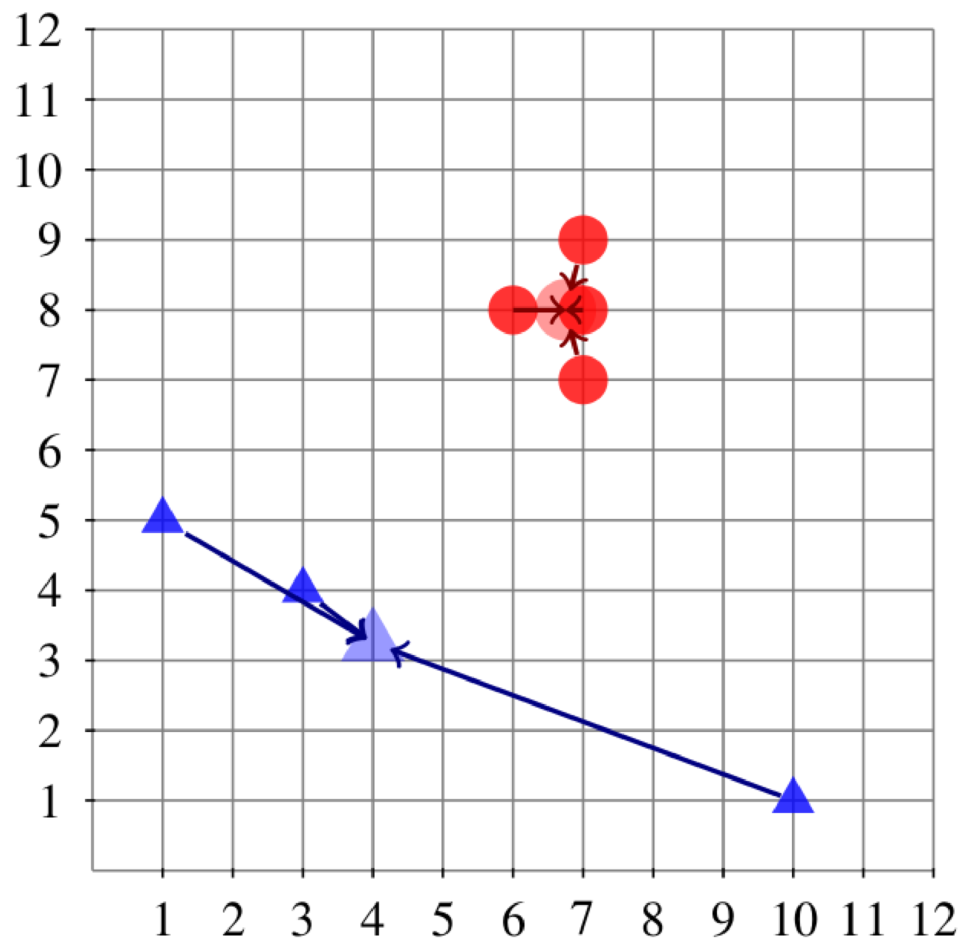
- Measure for compactness:

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

(sum of squares)

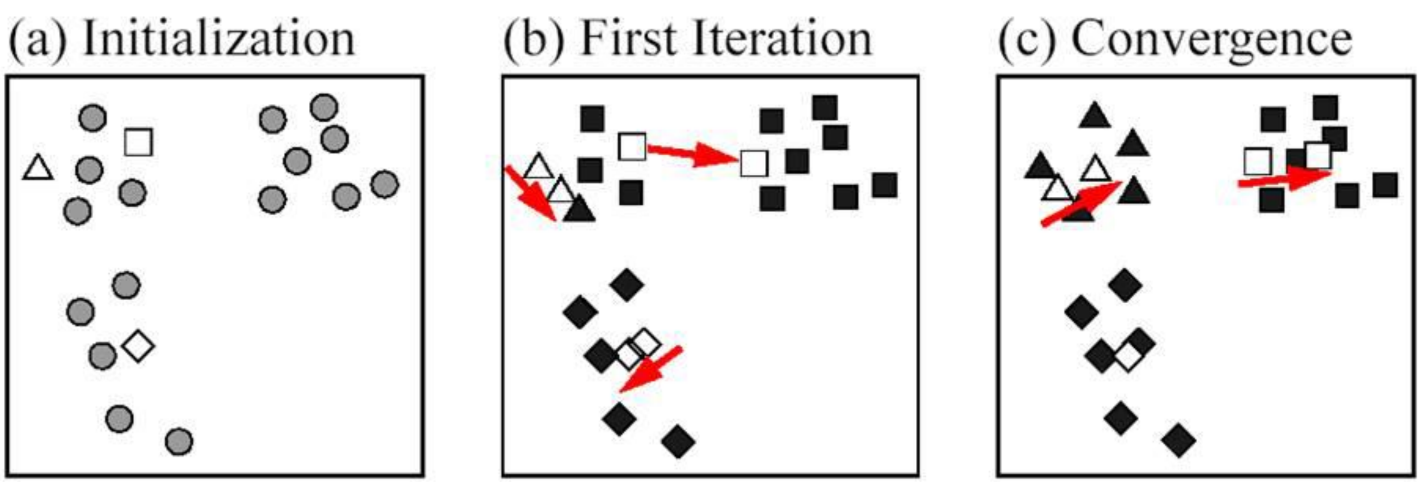
- Measure of compactness for a clustering:

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



Basic Algorithm: Clustering by Minimization of Variance [Forgy, 1965, Lloyd, 1982]

- start with k (e.g., randomly selected) points as cluster representatives (or with a random partition into k "clusters")
- repeat:
  1. assign each point to the closest representative
  2. 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:

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

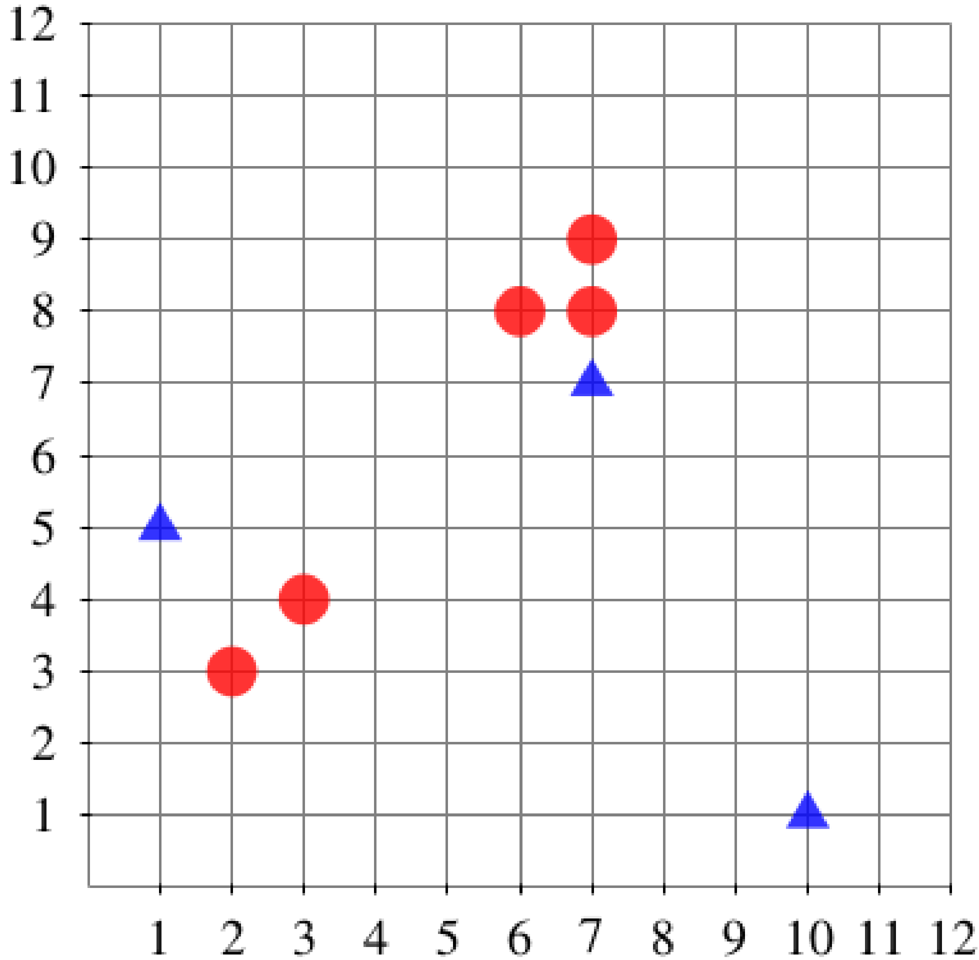
Clustering & Feature Spaces
Lecture Content

- Clustering
  - Clustering in General
  - Partitional Clustering
  - Visualization: Algorithmic Differences
  - Summary
- Feature Spaces
  - Distances
  - Features for Images
  - Summary

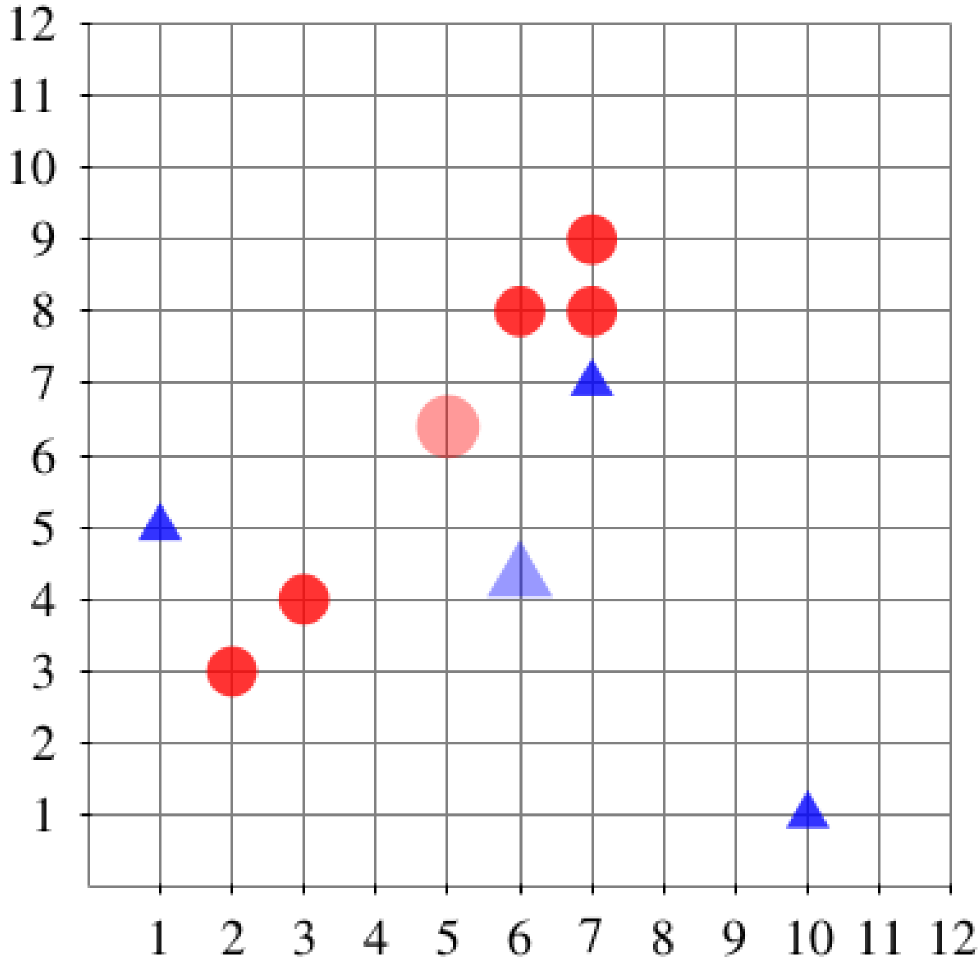
k-means Clustering

Lloyd/Forgy Algorithm

k-means Clustering – Lloyd/Forgy Algorithm



k-means Clustering – Lloyd/Forgy Algorithm

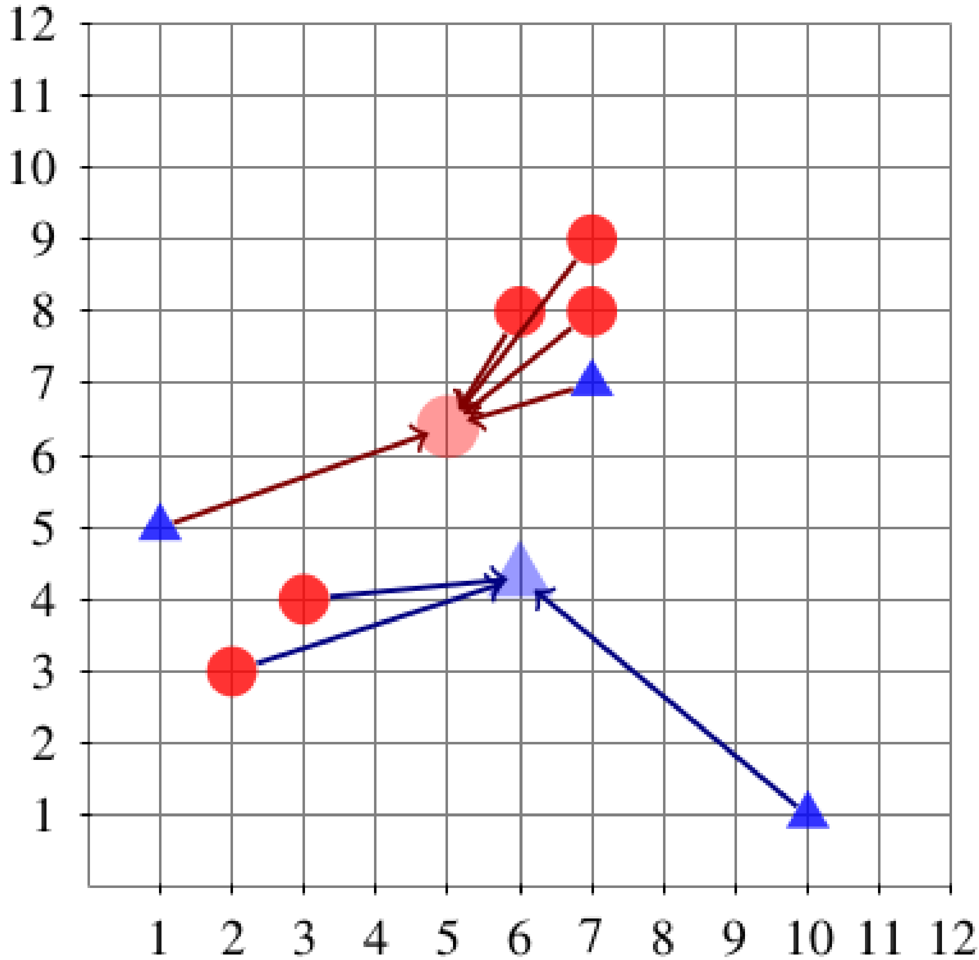


recompute centroids:

$\mu \approx (6.0, 4.3)$

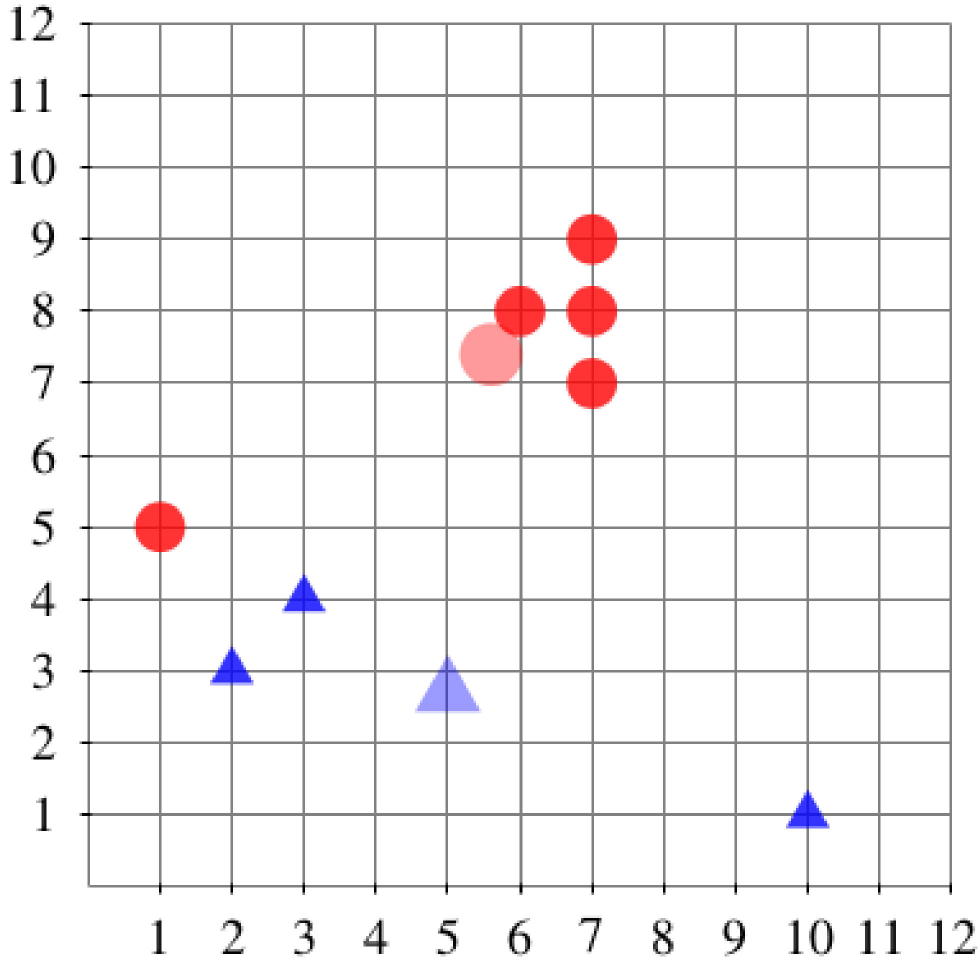
$\mu \approx (5.0, 6.4)$

k-means Clustering – Lloyd/Forgy Algorithm



reassign points

k-means Clustering – Lloyd/Forgy Algorithm

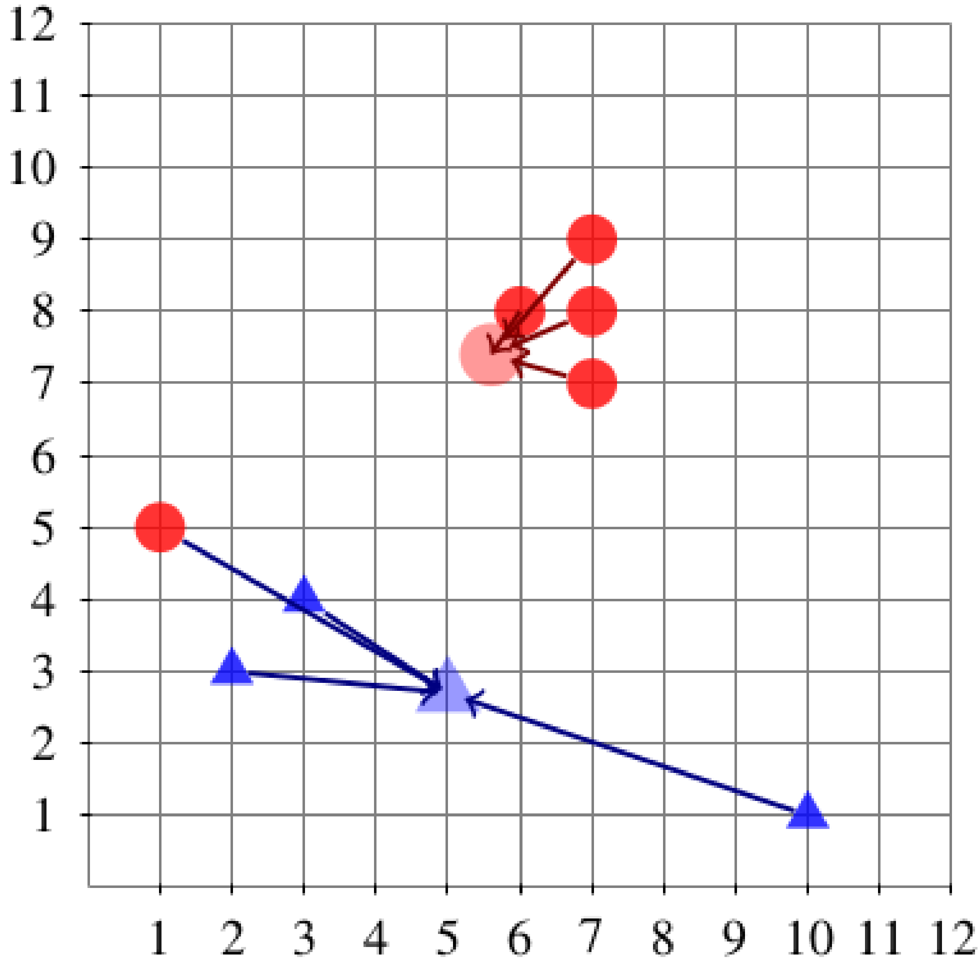


recompute centroids:

$$\mu \approx (5.0, 2.7)$$

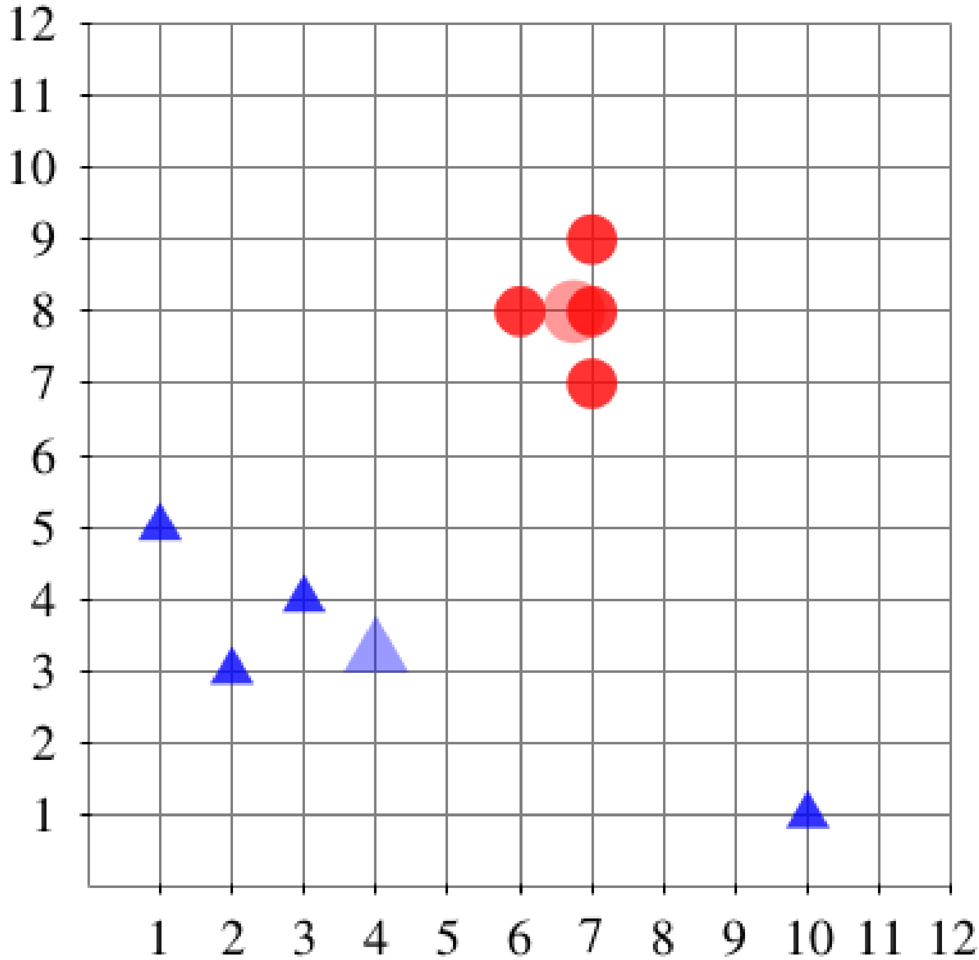
$$\mu \approx (5.6, 7.4)$$

k-means Clustering – Lloyd/Forgy Algorithm



reassign points

k-means Clustering – Lloyd/Forgy Algorithm

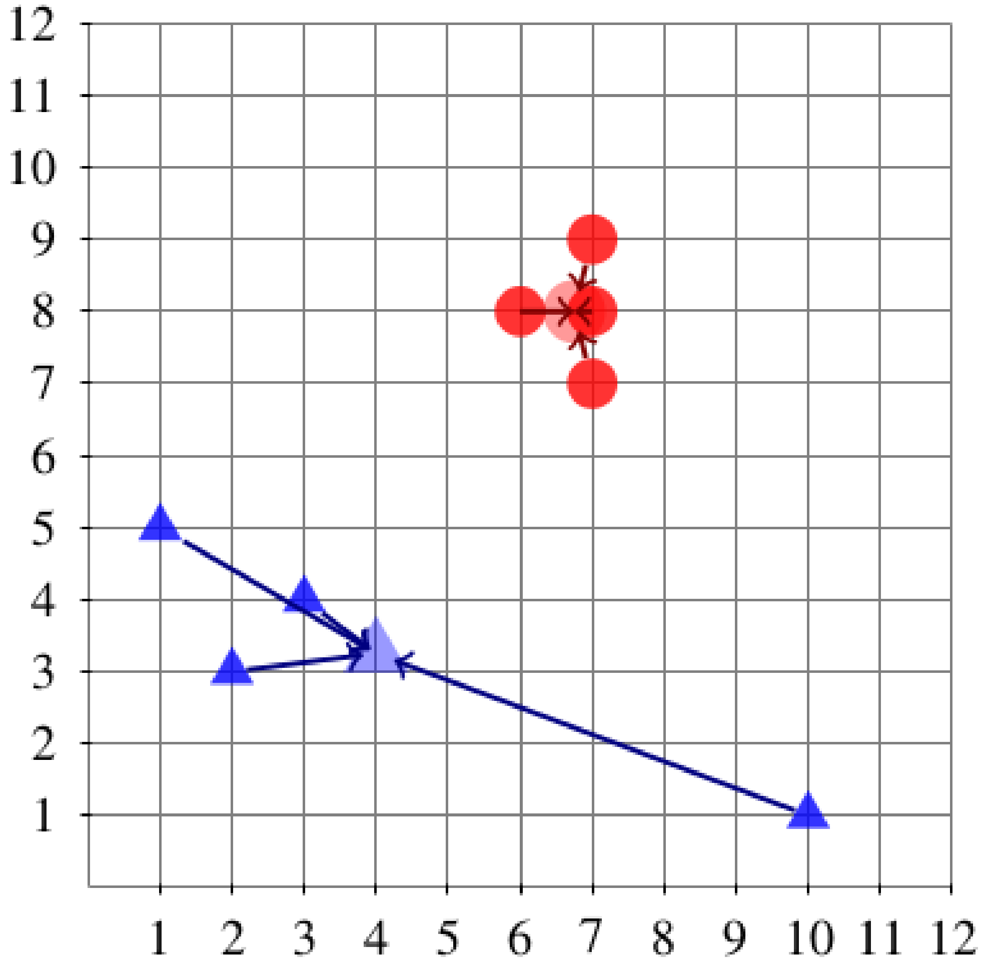


recompute centroids:

$$\mu \approx (4.0, 3.25)$$

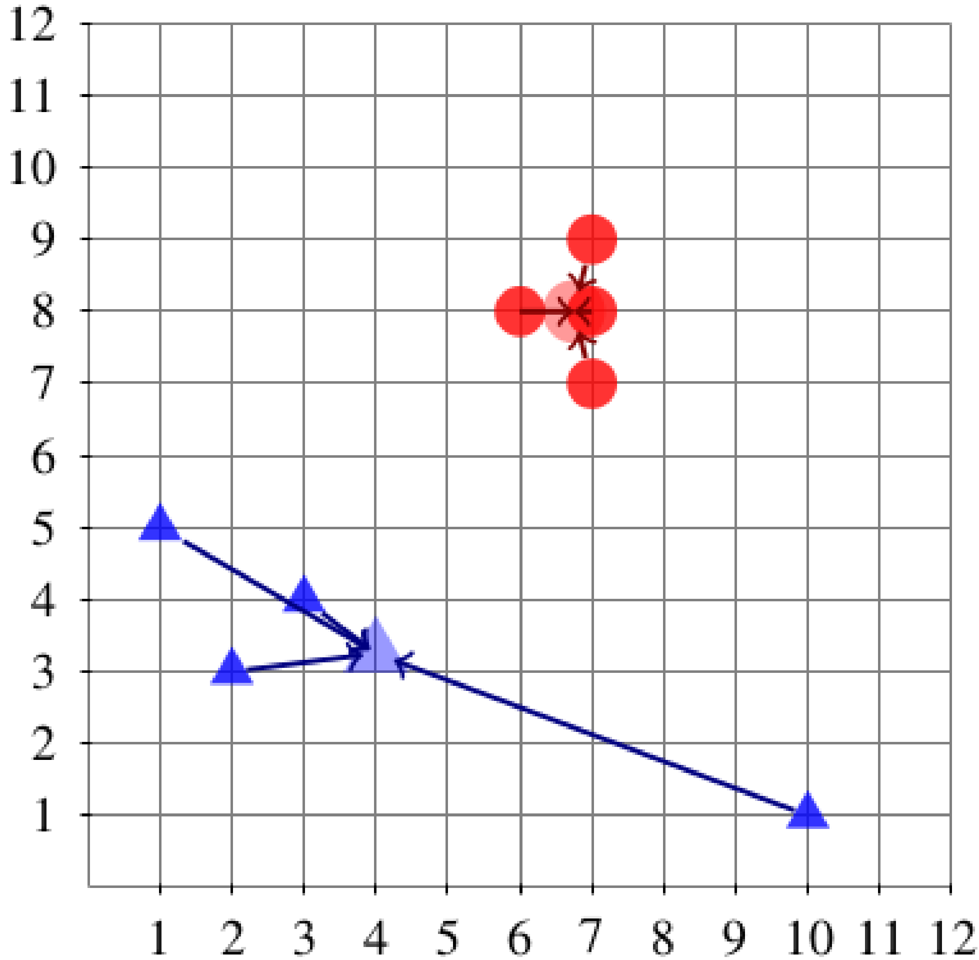
$$\mu \approx (6.75, 8.0)$$

k-means Clustering – Lloyd/Forgy Algorithm



reassign points

k-means Clustering – Lloyd/Forgy Algorithm

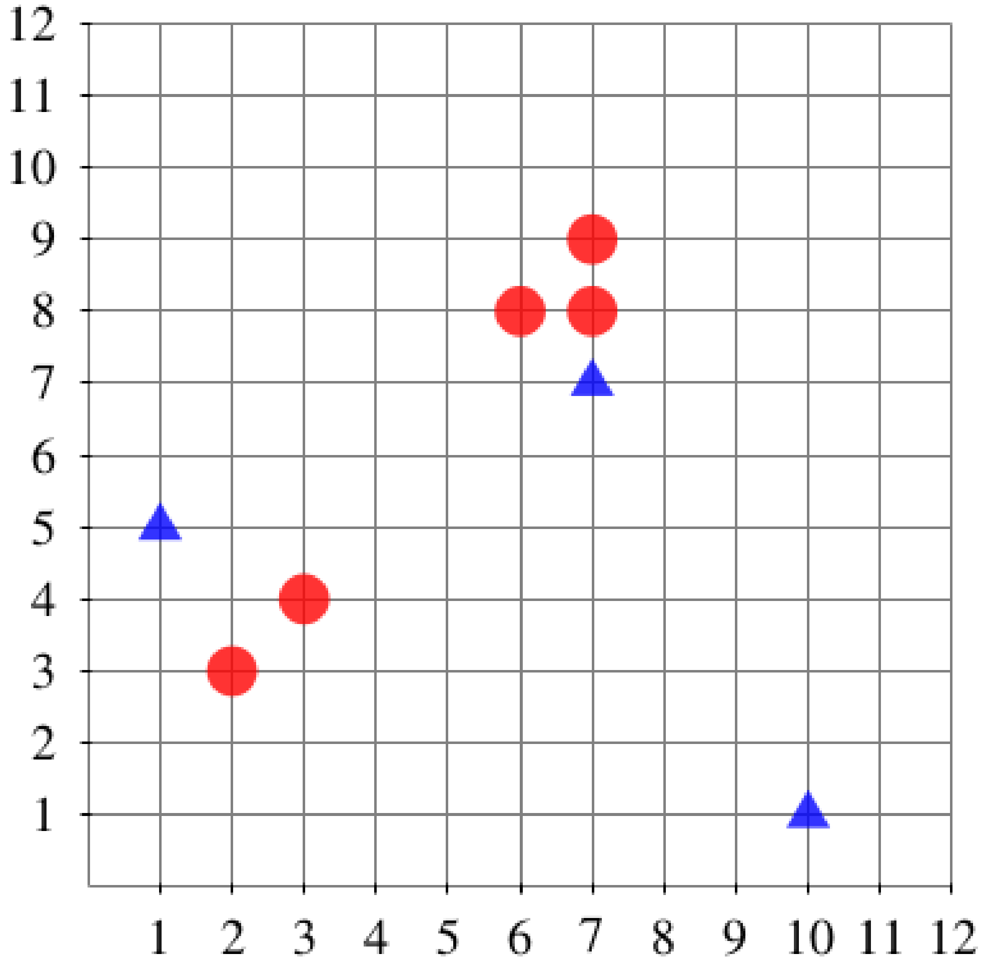


reassign points
no change
convergence!

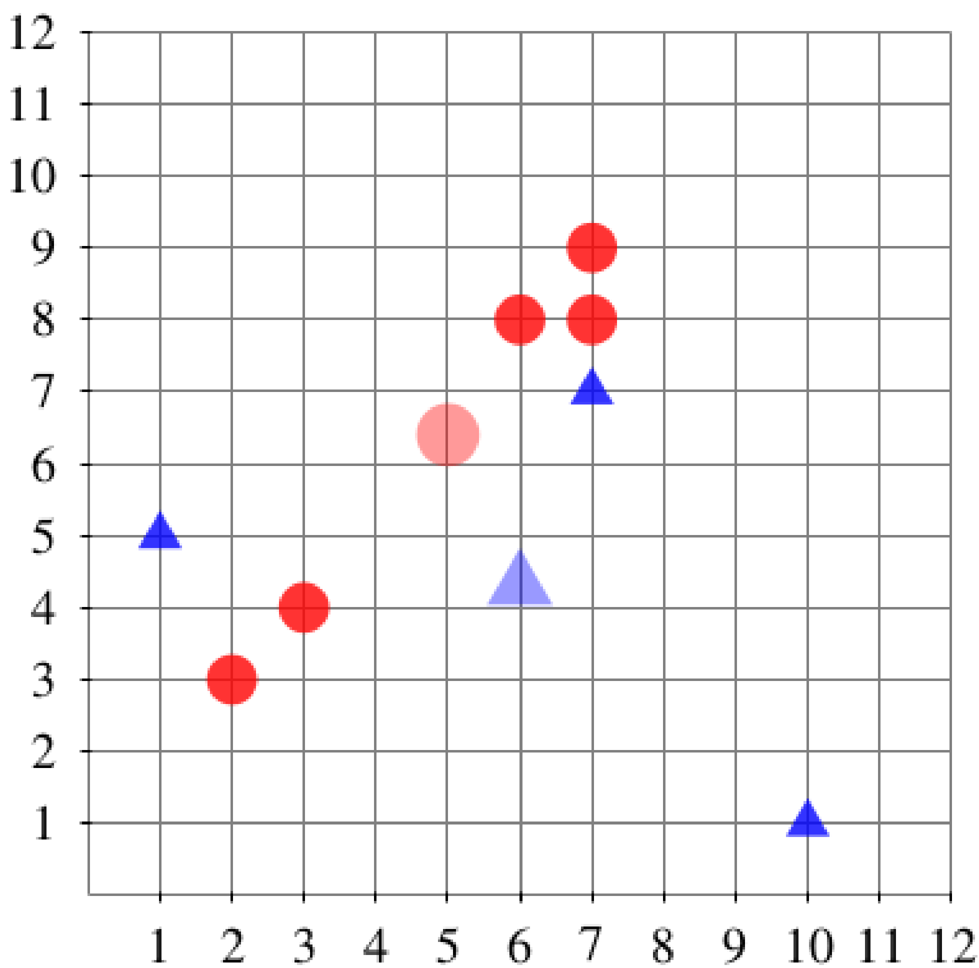
k-means Clustering

MacQueen Algorithm

k-means Clustering – MacQueen Algorithm



k-means Clustering – MacQueen Algorithm

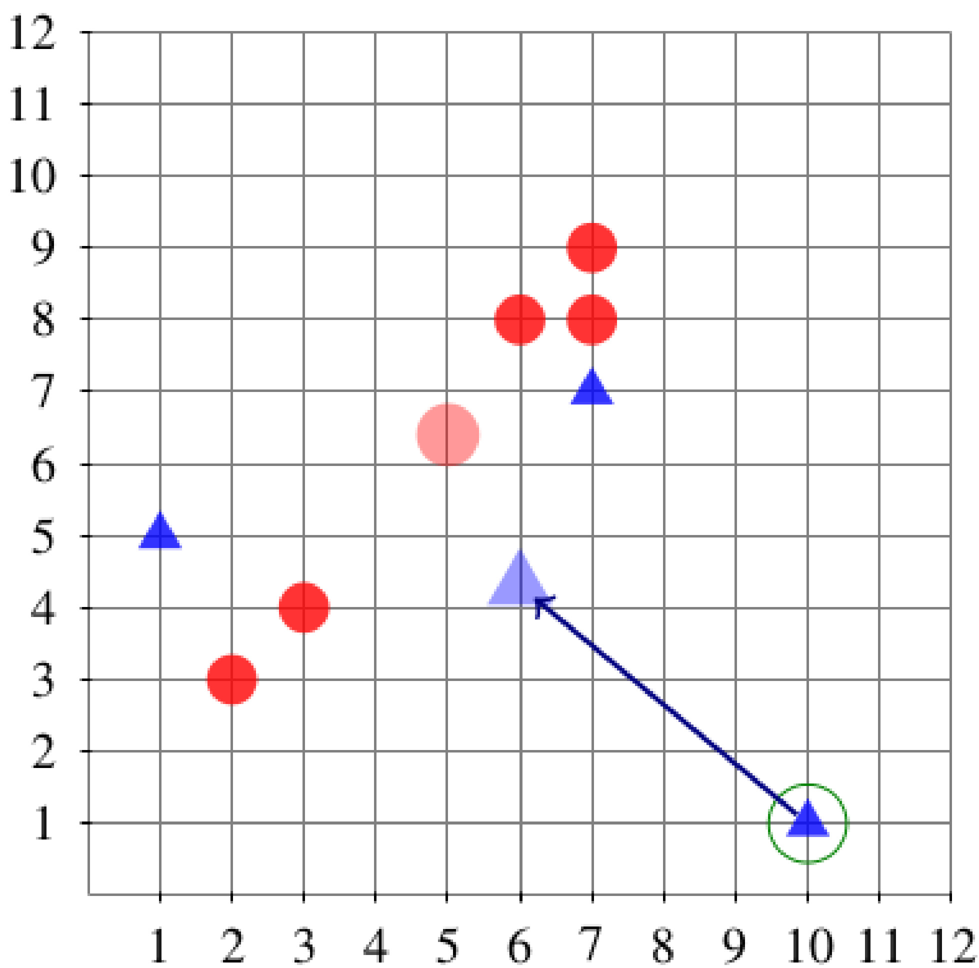


Centroids (e.g.: from previous iteration):

$\mu \approx (6.0, 4.3)$

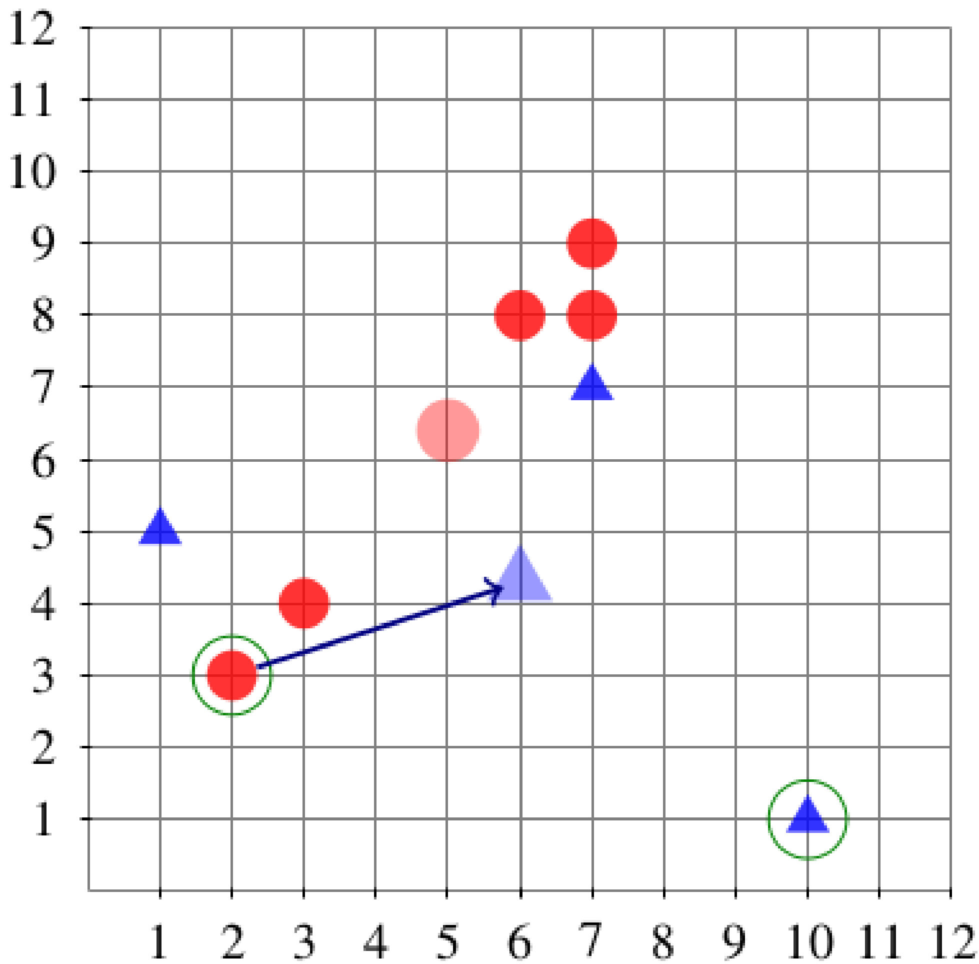
$\mu \approx (5.0, 6.4)$

k-means Clustering – MacQueen Algorithm



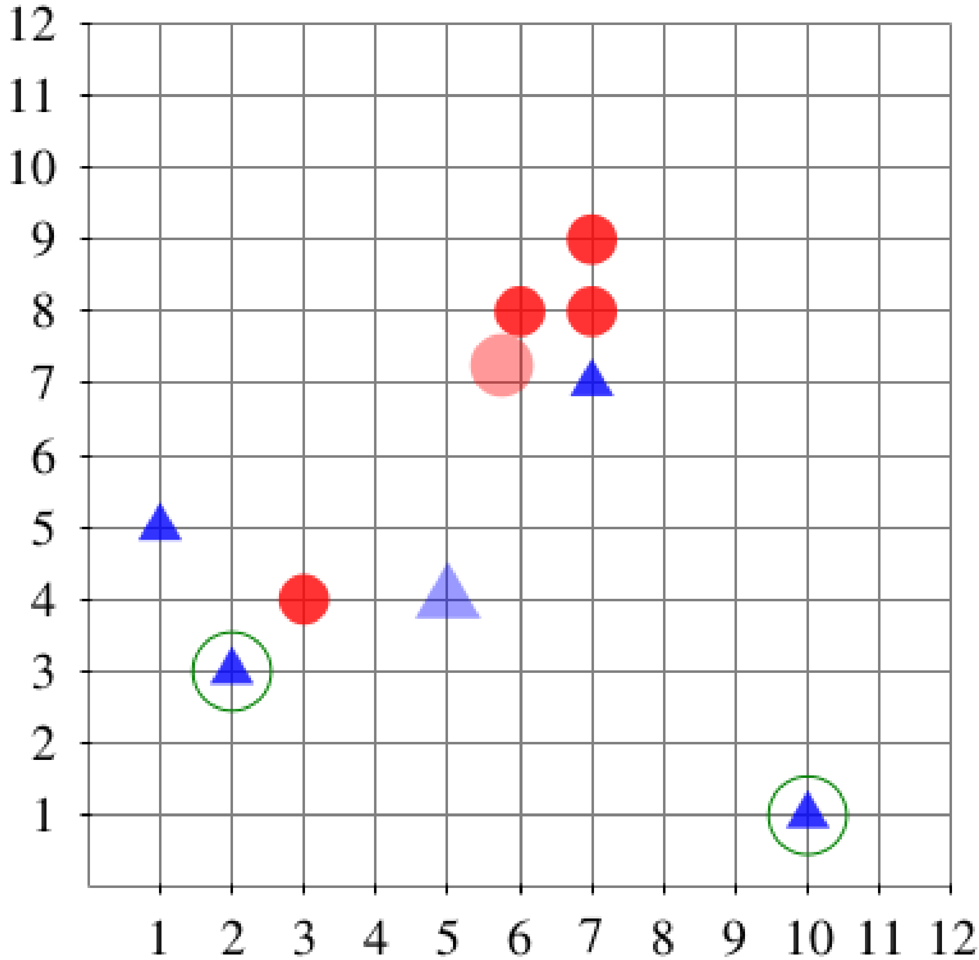
assign first point

k-means Clustering – MacQueen Algorithm



assign second point

k-means Clustering – MacQueen Algorithm

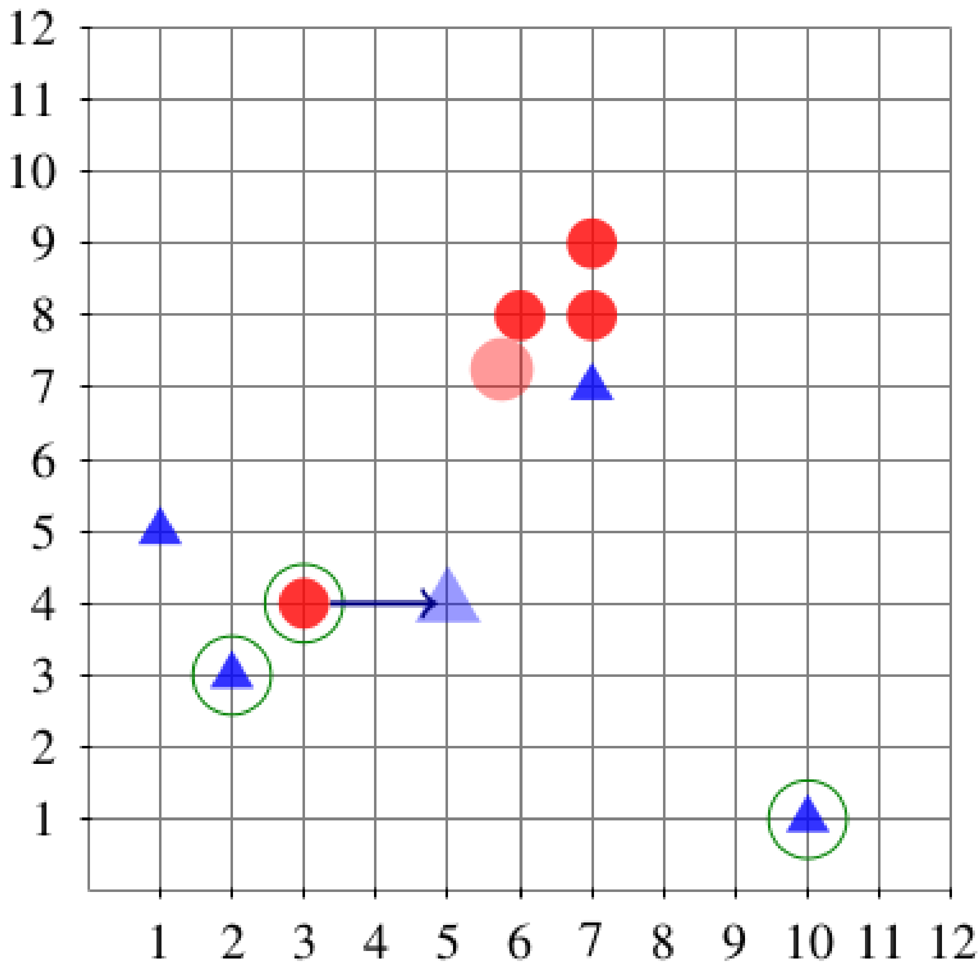


recompute centroids:

$\mu \approx (5.0, 4.0)$

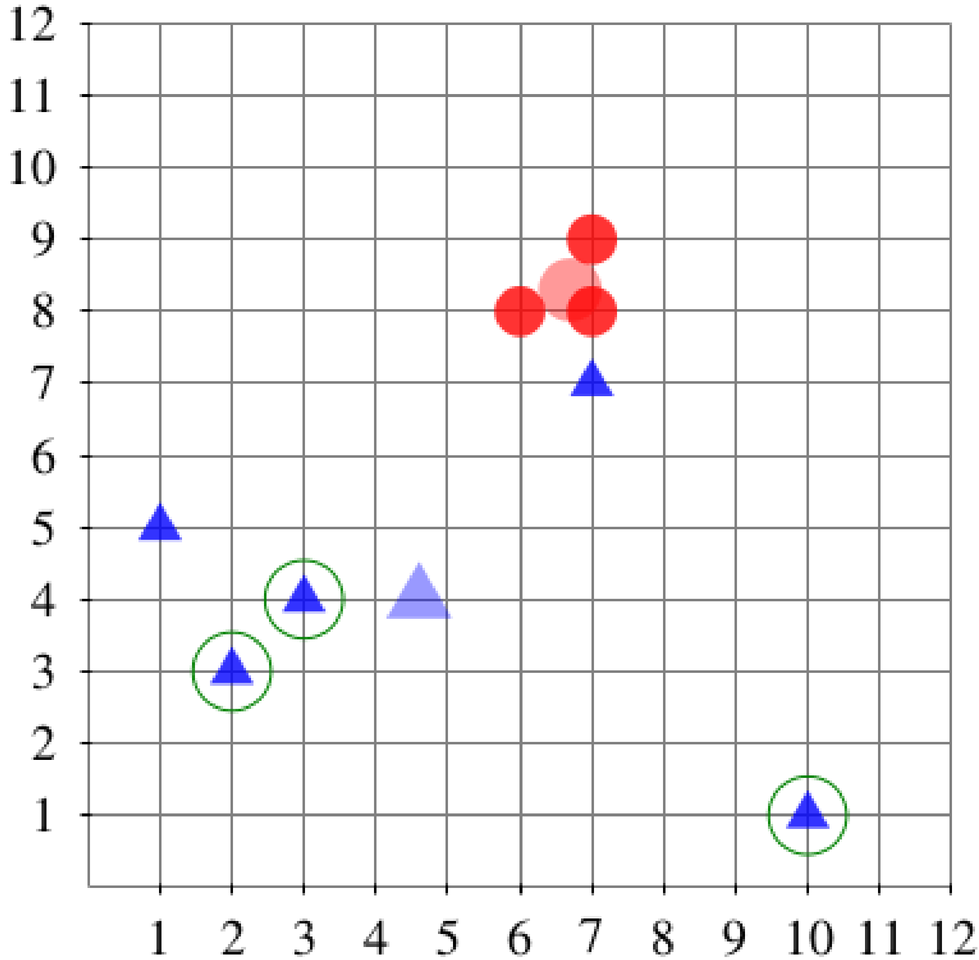
$\mu \approx (5.75, 7.25)$

k-means Clustering – MacQueen Algorithm



assign third point

k-means Clustering – MacQueen Algorithm

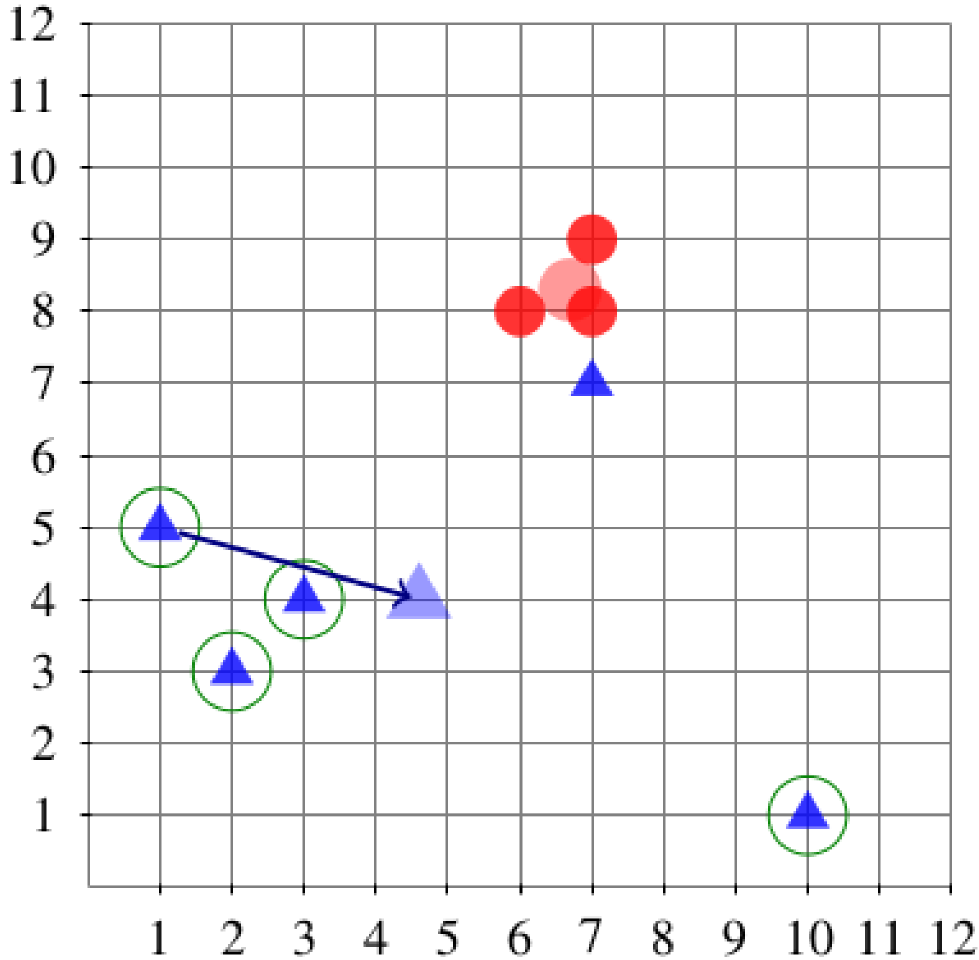


recompute centroids:

$\mu \approx (4.6, 4.0)$

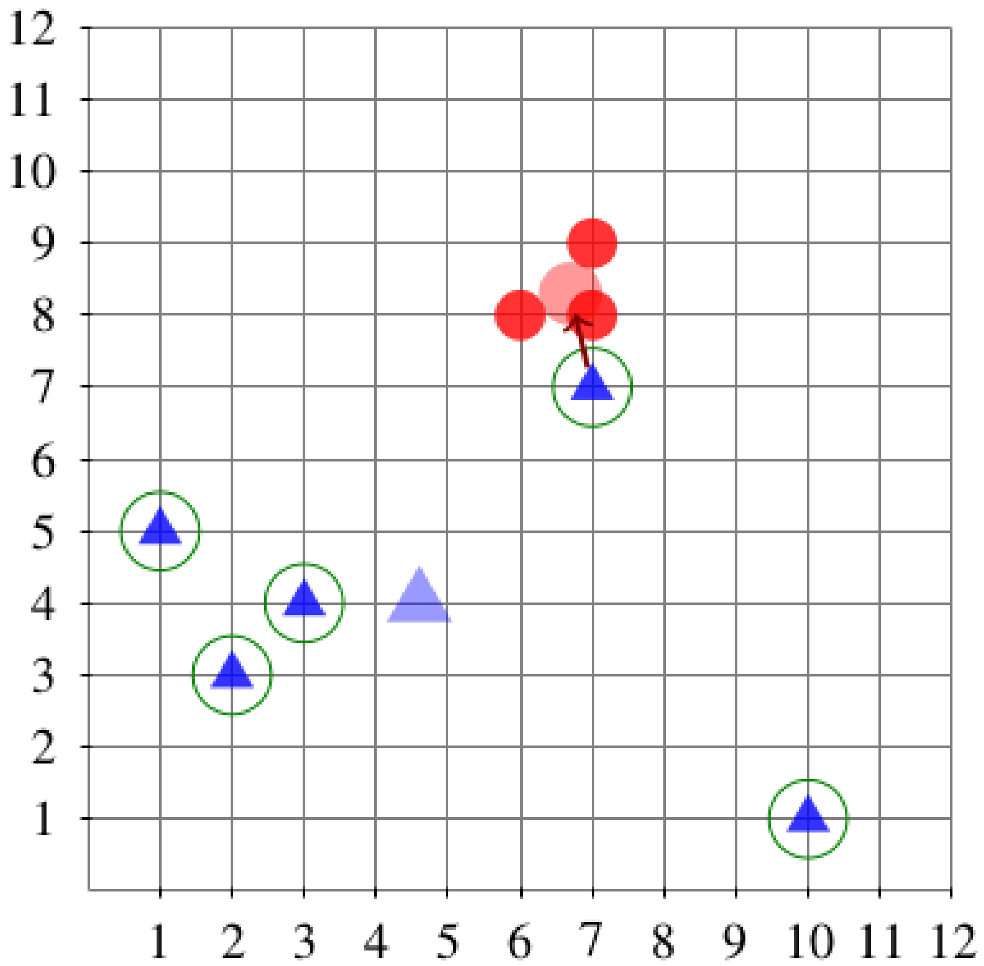
$\mu \approx (6.7, 8.3)$

k-means Clustering – MacQueen Algorithm



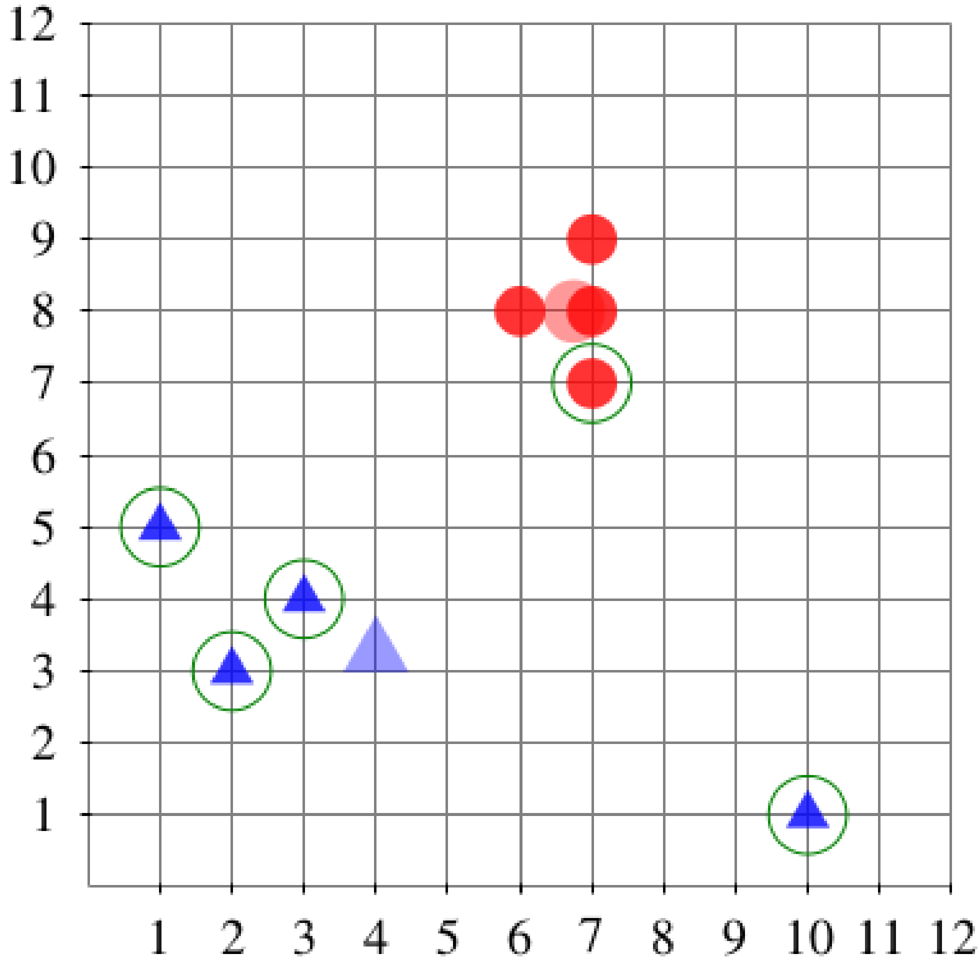
assign fourth point

k-means Clustering – MacQueen Algorithm



assing fifth point

k-means Clustering – MacQueen Algorithm

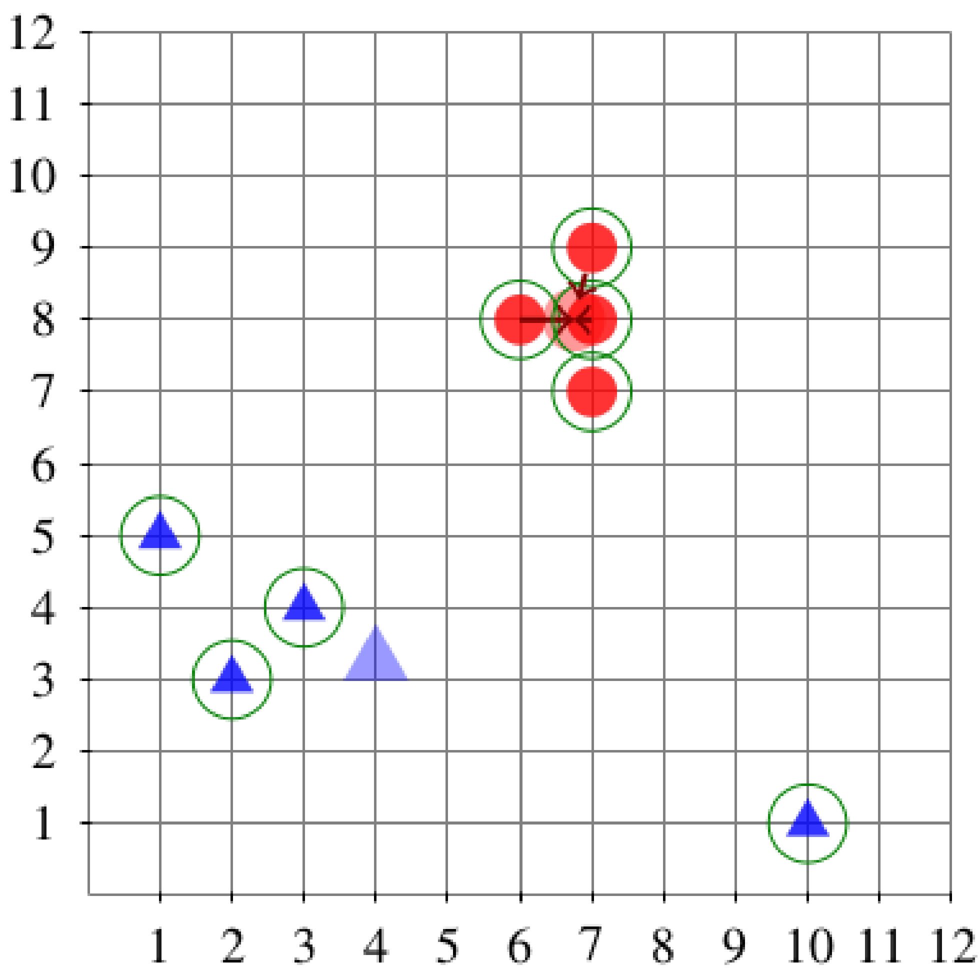


recompute centroids:

$$\mu \approx (4.0, 3.25)$$

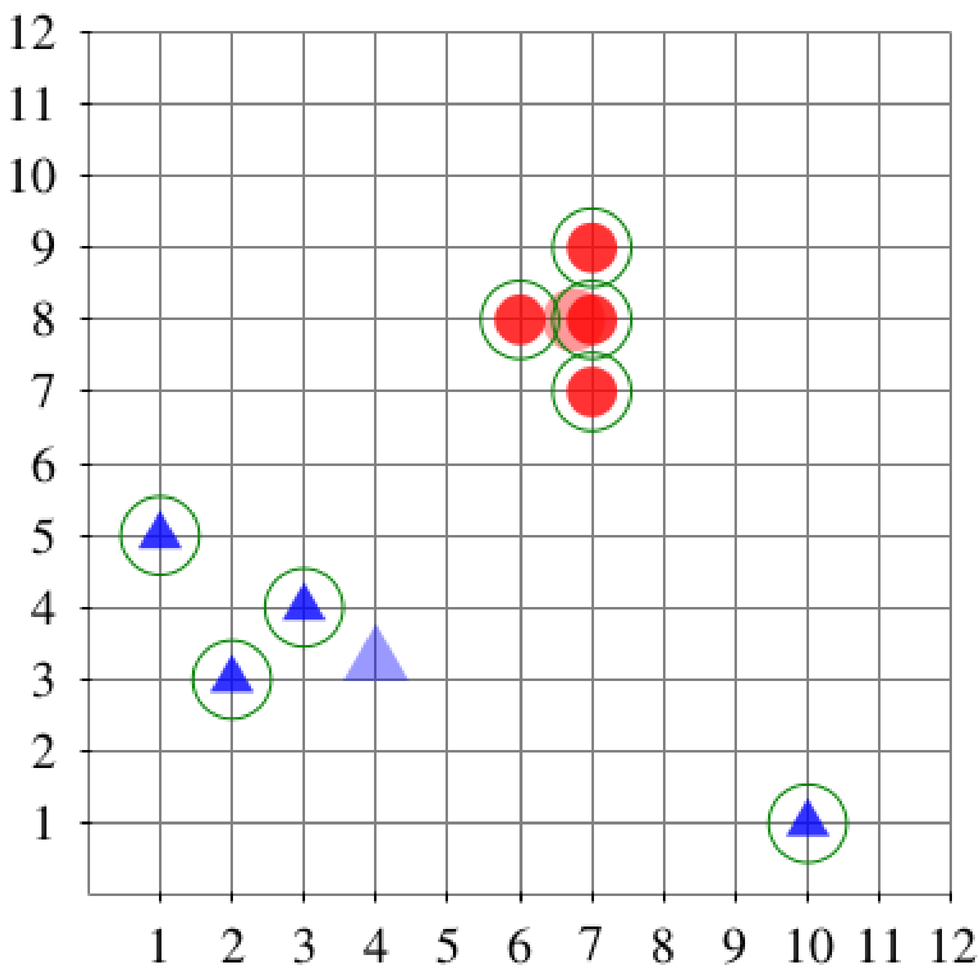
$$\mu \approx (6.75, 8.0)$$

k-means Clustering – MacQueen Algorithm



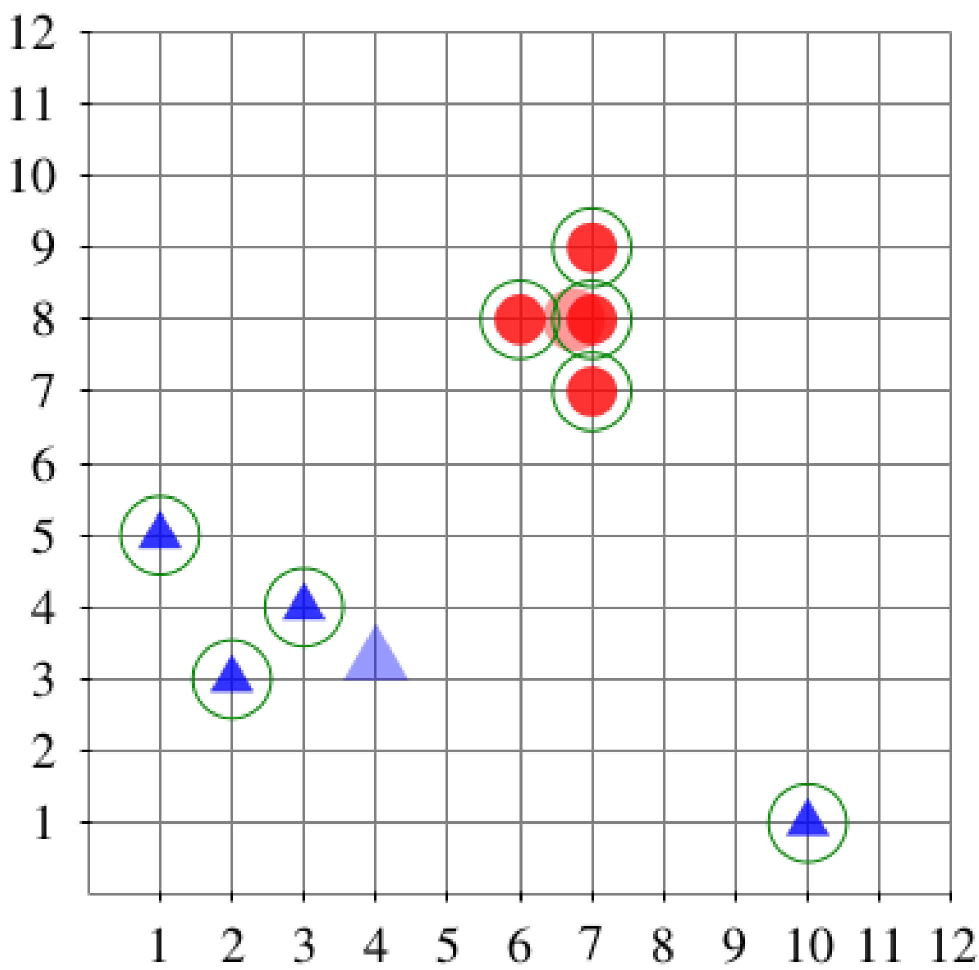
reassign more points

k-means Clustering – MacQueen Algorithm



reassign more points
possibly more iterations

k-means Clustering – MacQueen Algorithm



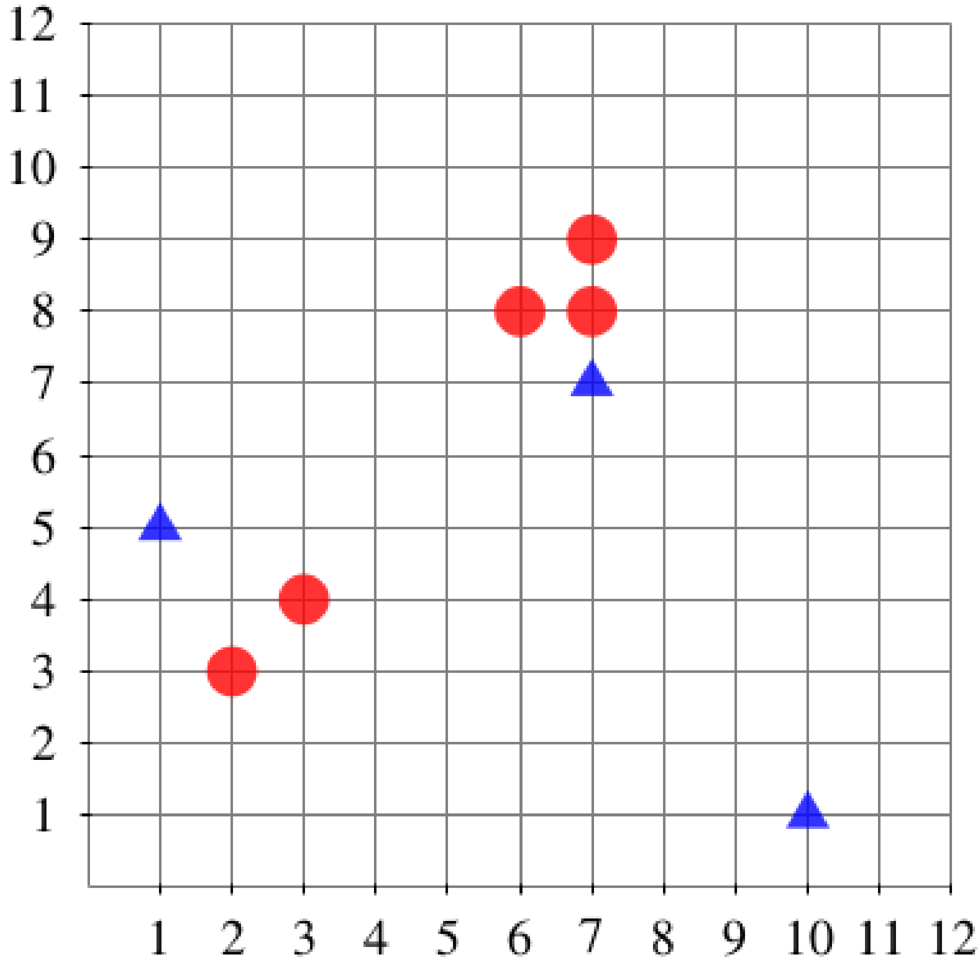
reassign more points
possibly more iterations
convergence

k-means Clustering

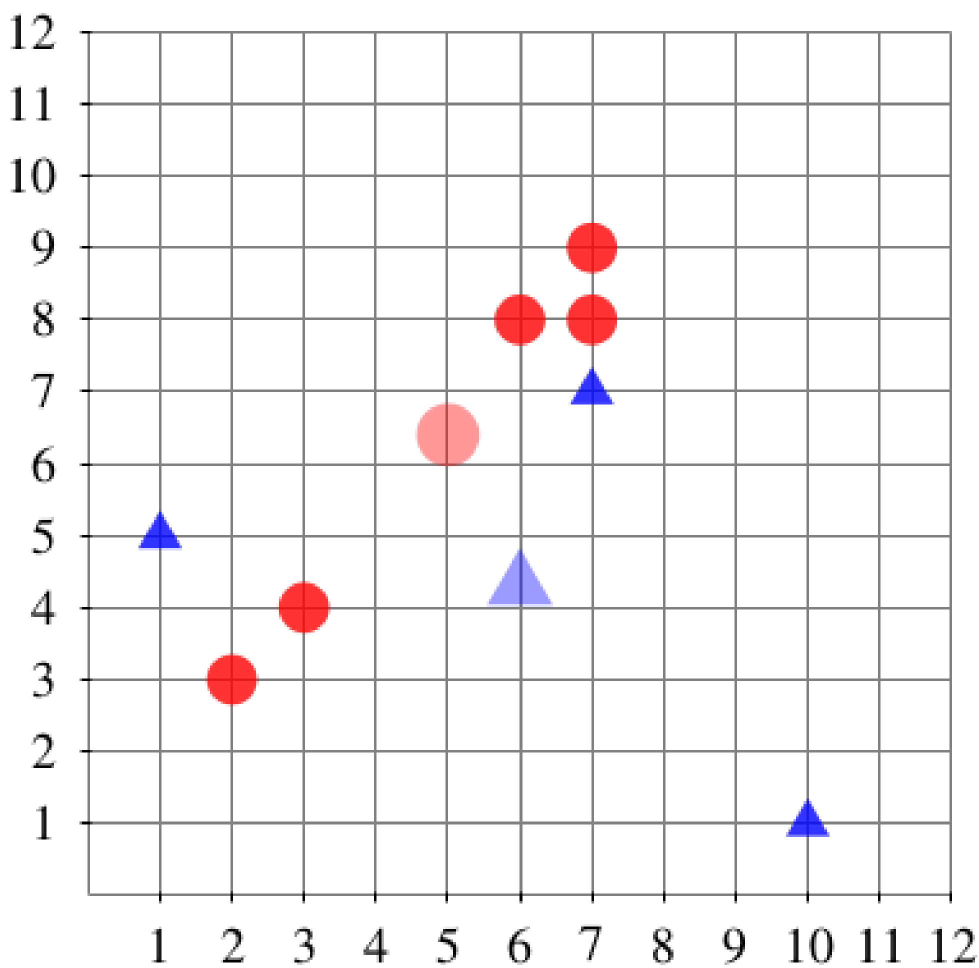
MacQueen Algorithm

Alternative Ordering

k-means Clustering – MacQueen Algorithm



k-means Clustering – MacQueen Algorithm

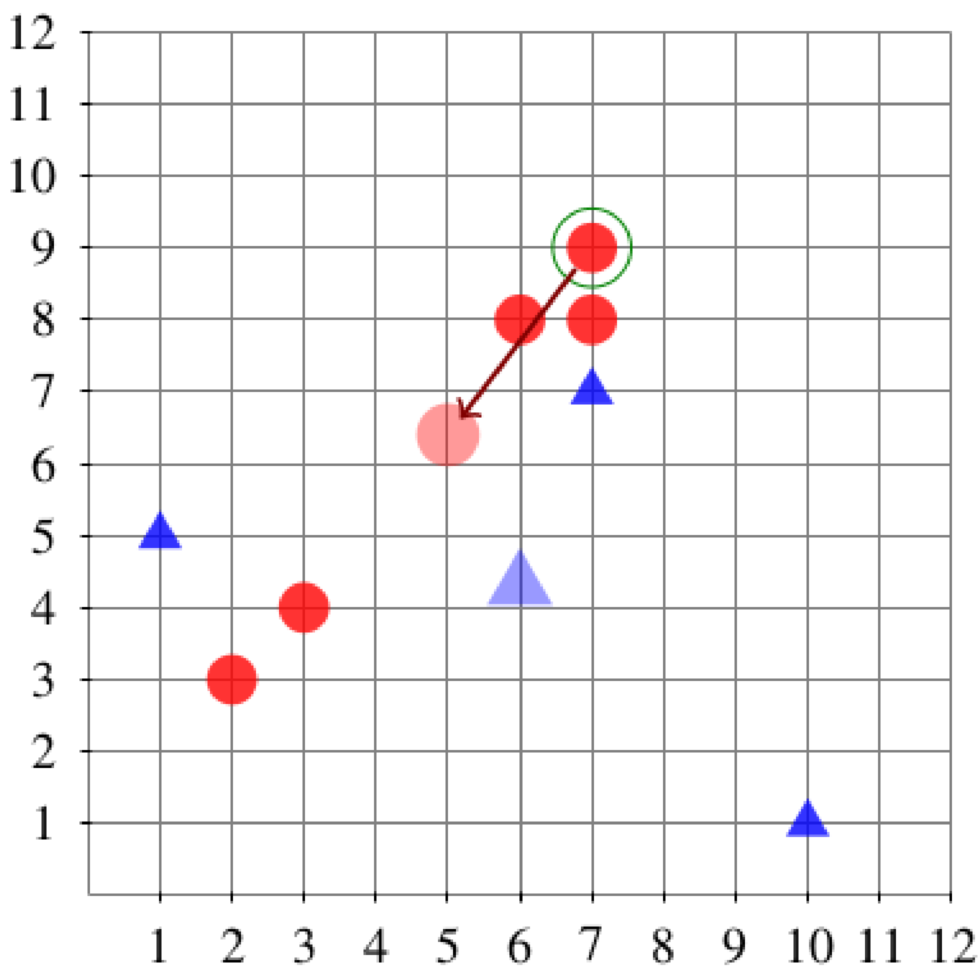


Centroids (e.g.: from previous iteration):

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

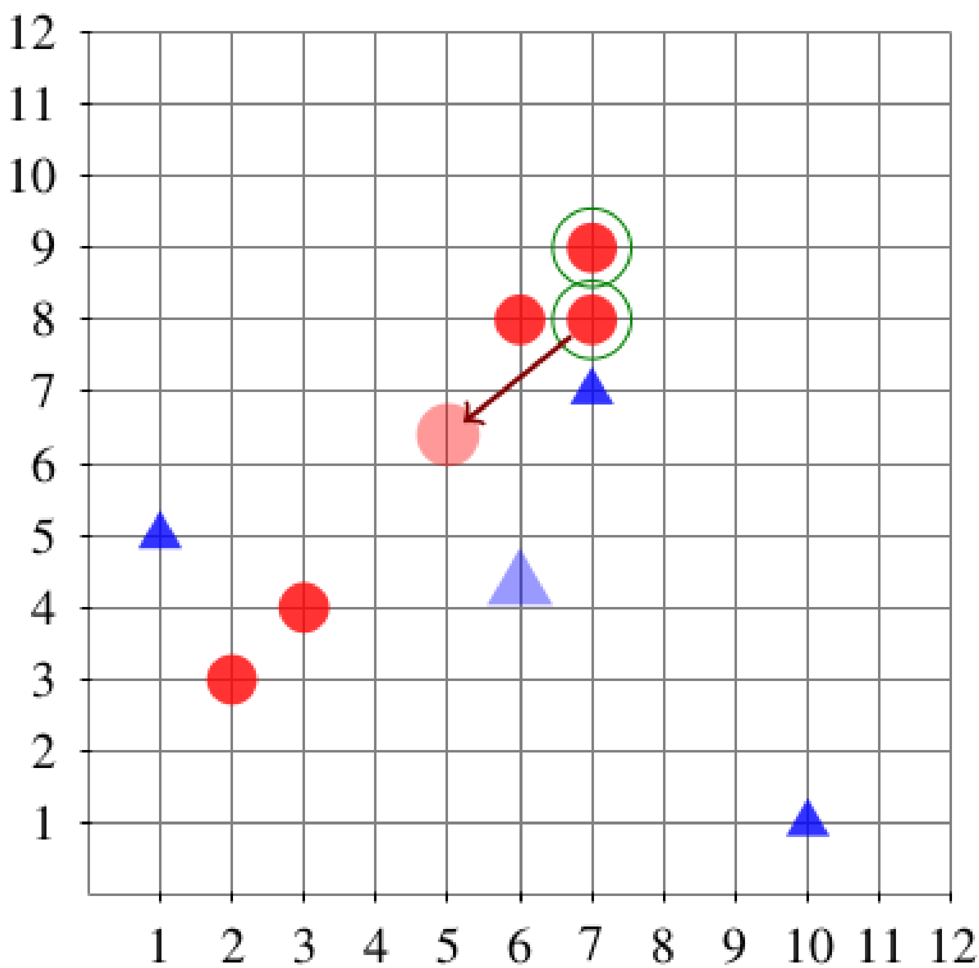
$$\mu \approx (5.0, 6.4)$$

k-means Clustering – MacQueen Algorithm



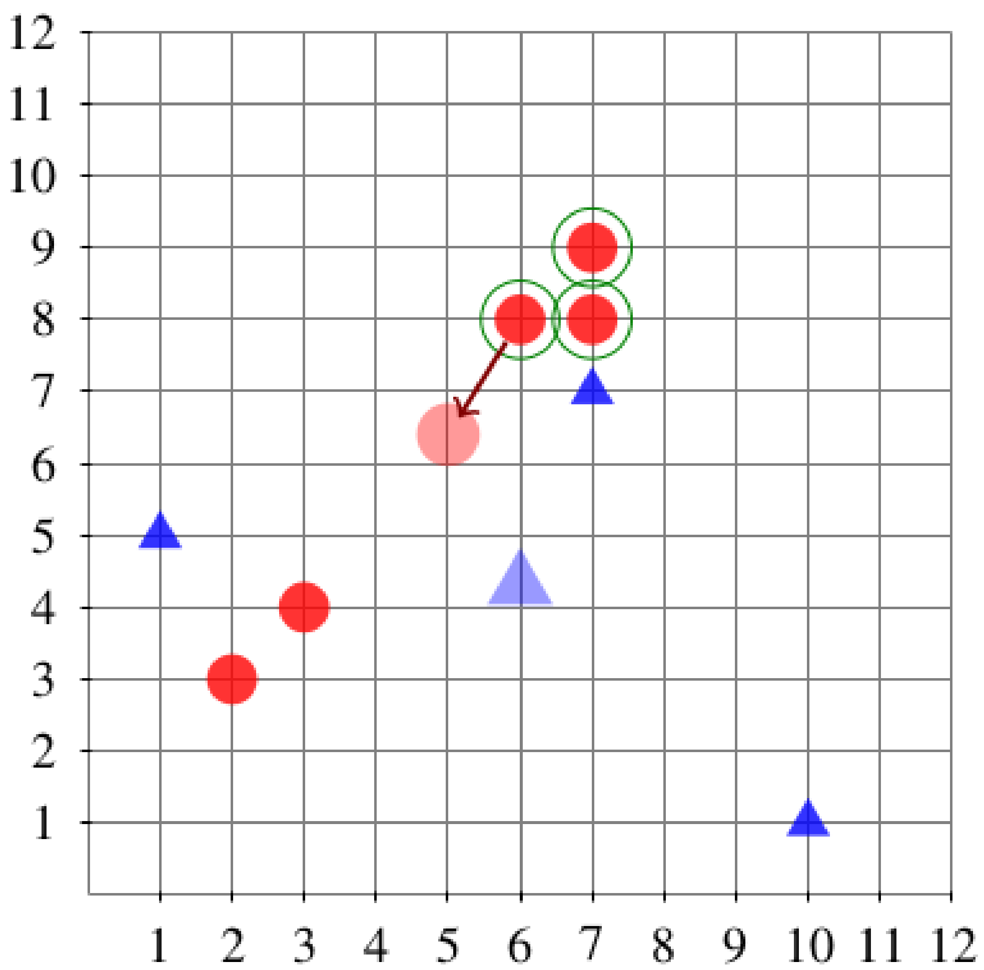
assign first point

k-means Clustering – MacQueen Algorithm



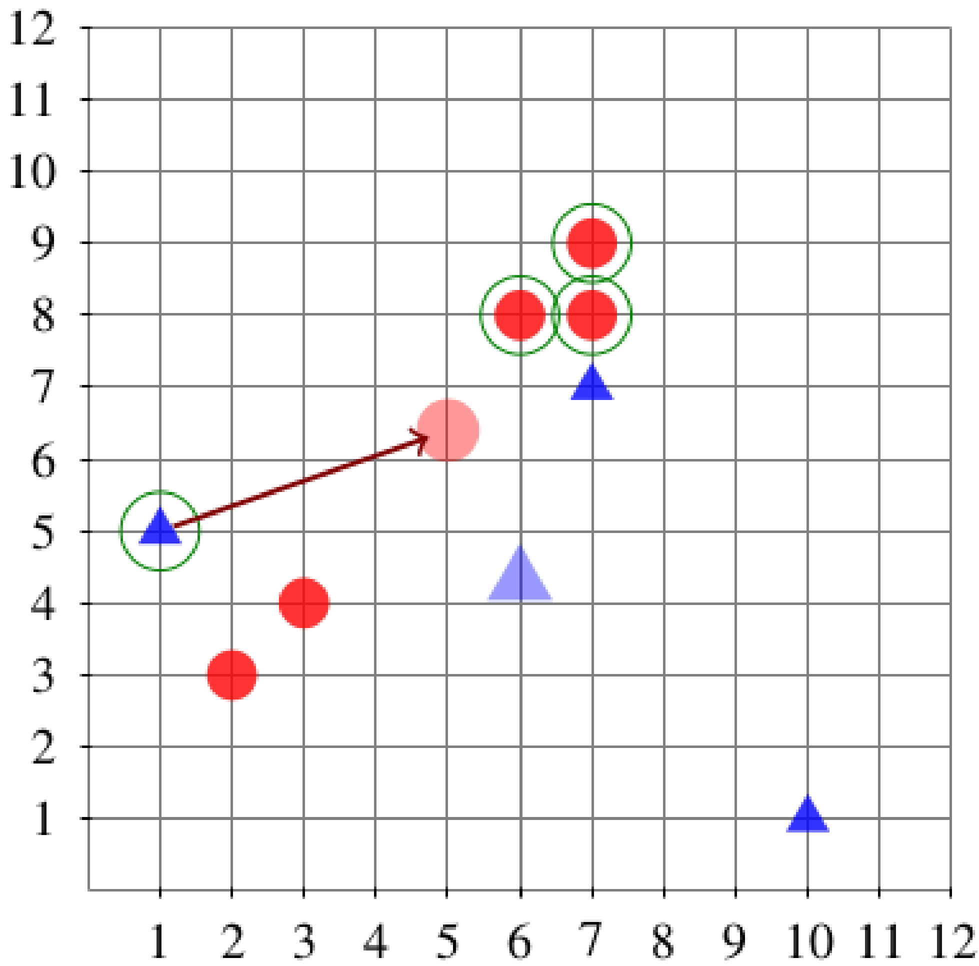
assign second point

k-means Clustering – MacQueen Algorithm



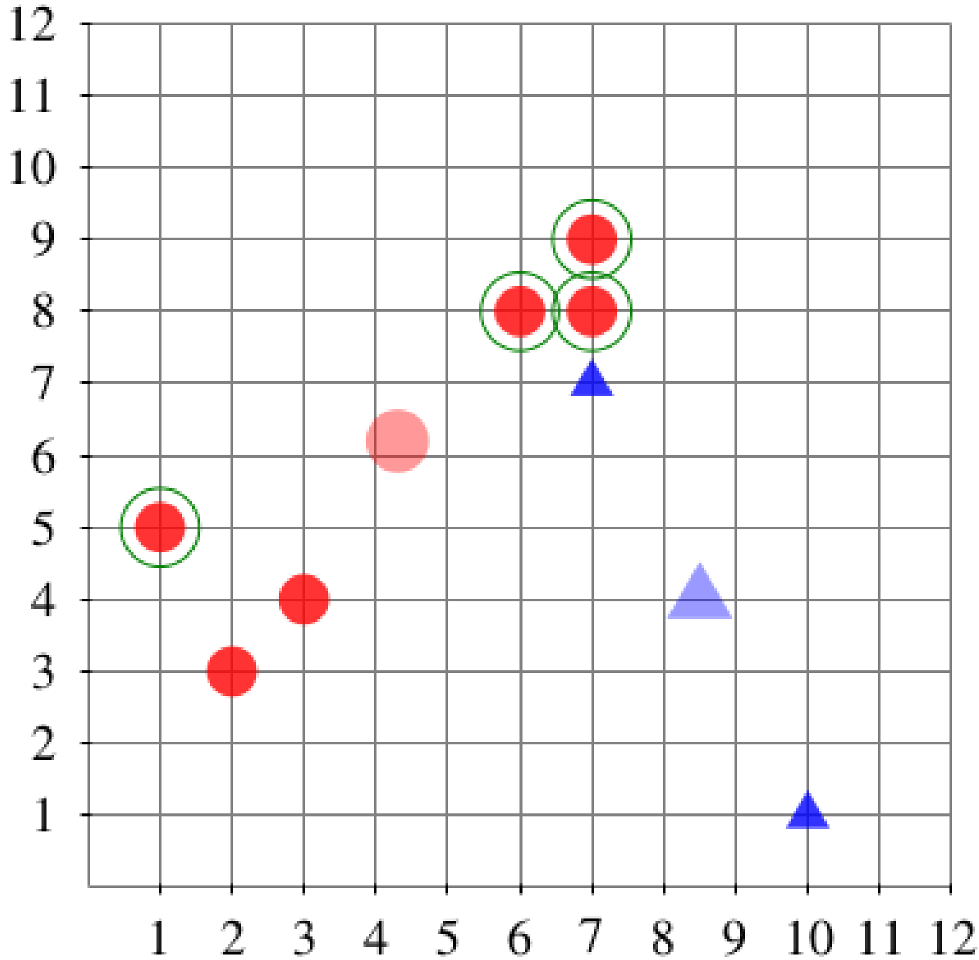
assign third point

k-means Clustering – MacQueen Algorithm



assign fourth point

k-means Clustering – MacQueen Algorithm

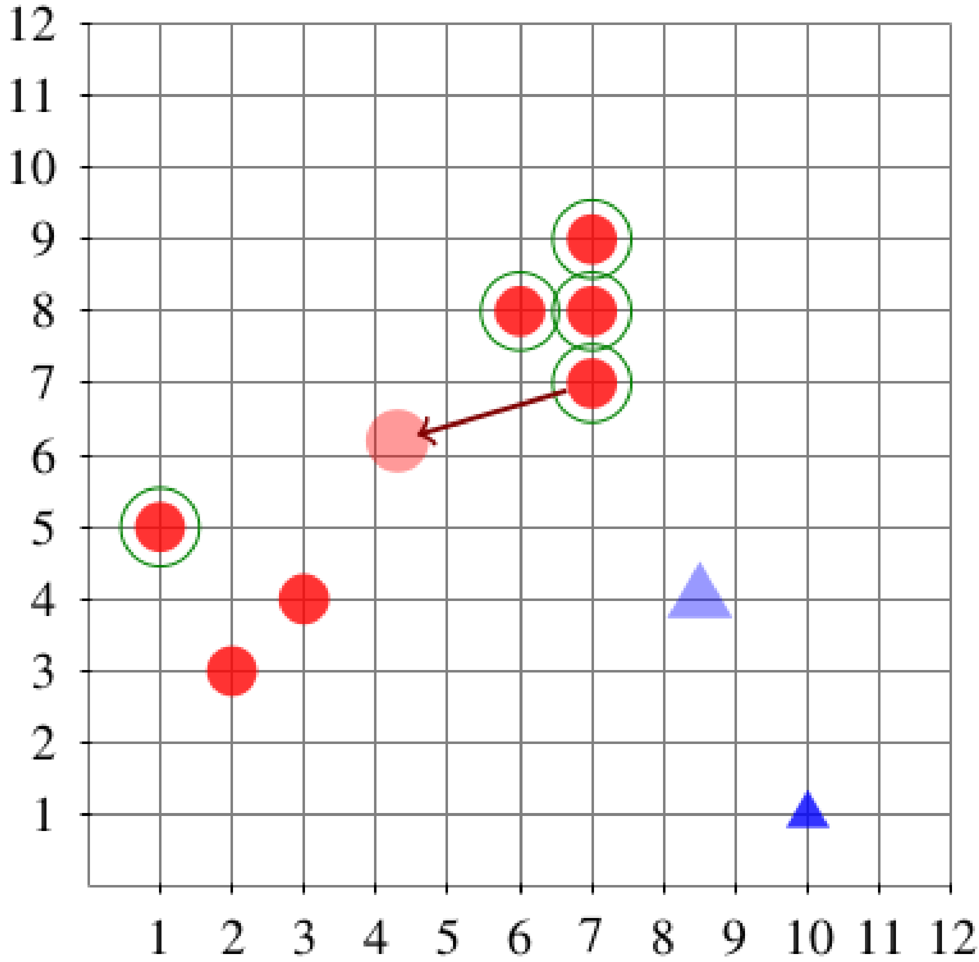


recompute centroids:

$\mu \approx (4.0, 8.5)$

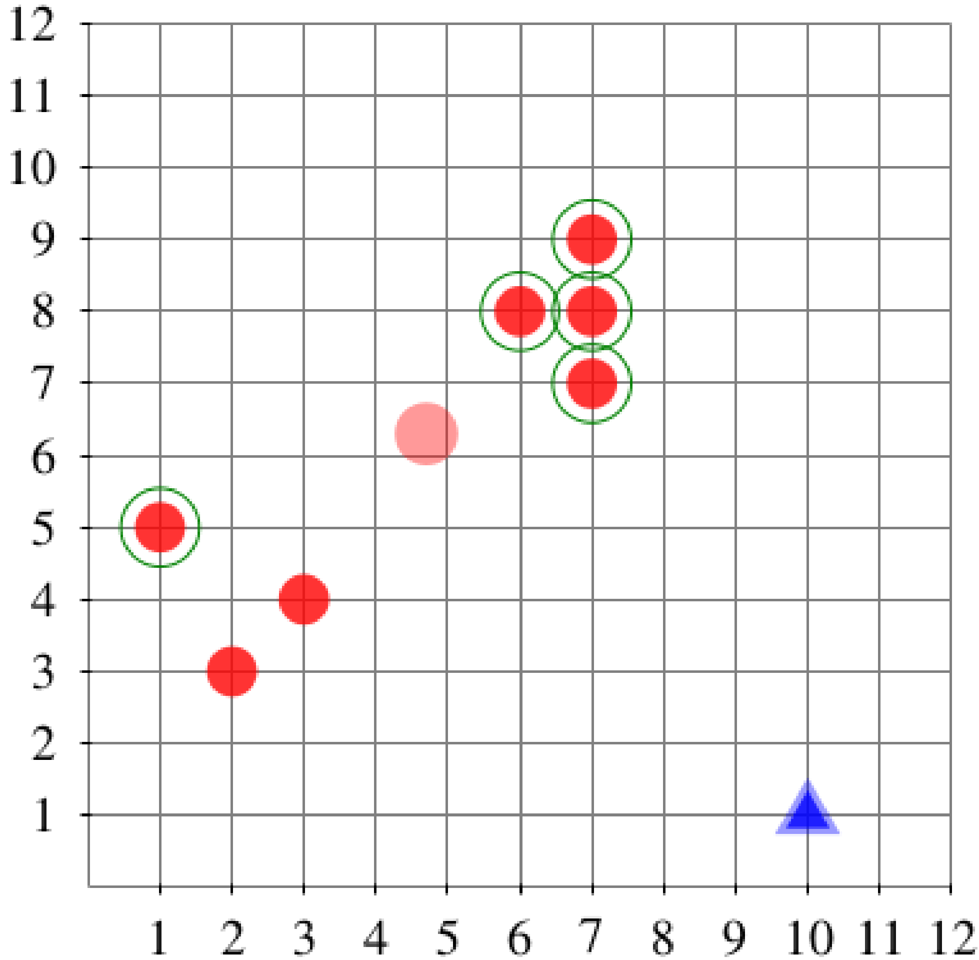
$\mu \approx (4.3, 6.2)$

k-means Clustering – MacQueen Algorithm



assign fifth point

k-means Clustering – MacQueen Algorithm

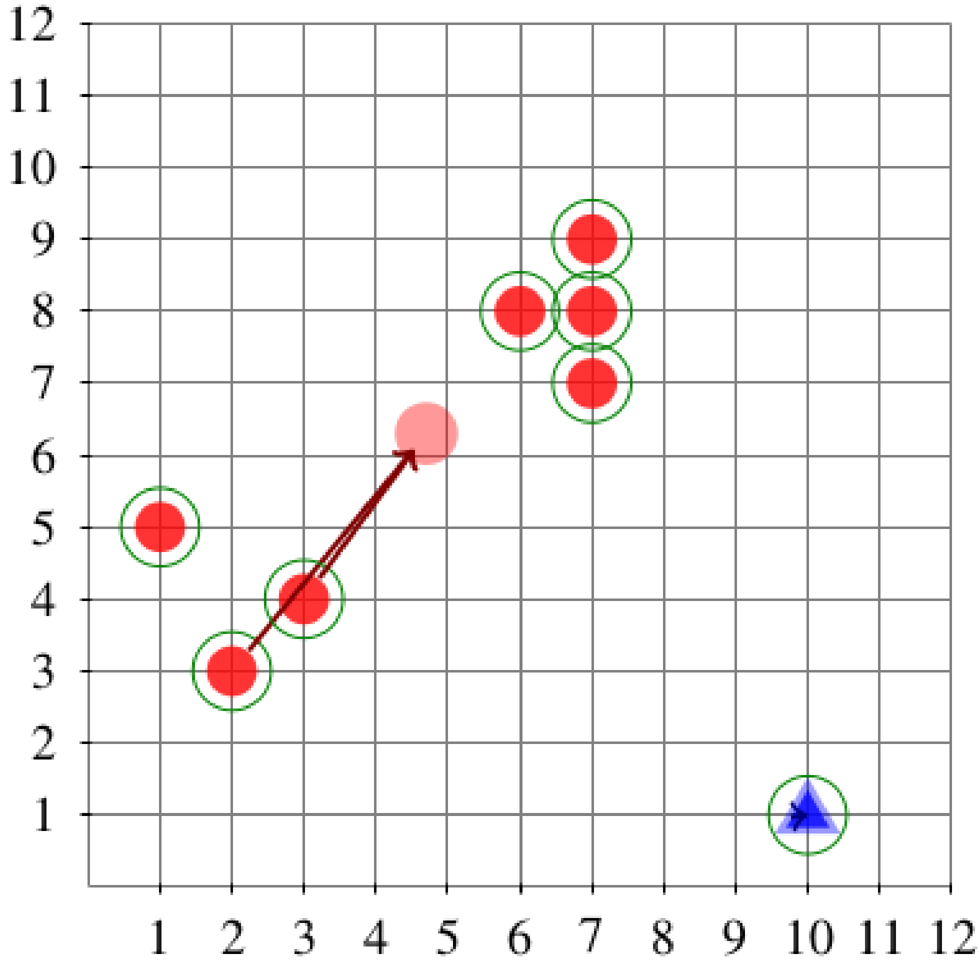


recompute centroids:

$\mu \approx (10.0, 1.0)$

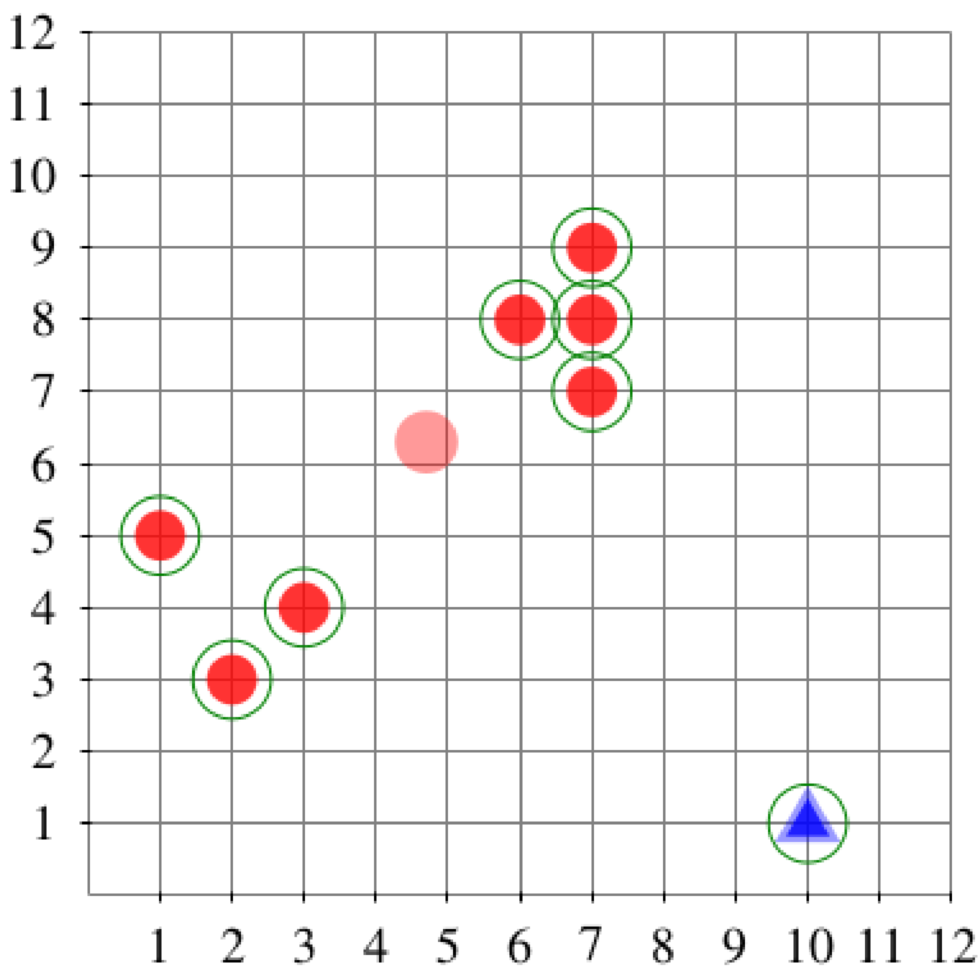
$\mu \approx (4.7, 6.3)$

k-means Clustering – MacQueen Algorithm



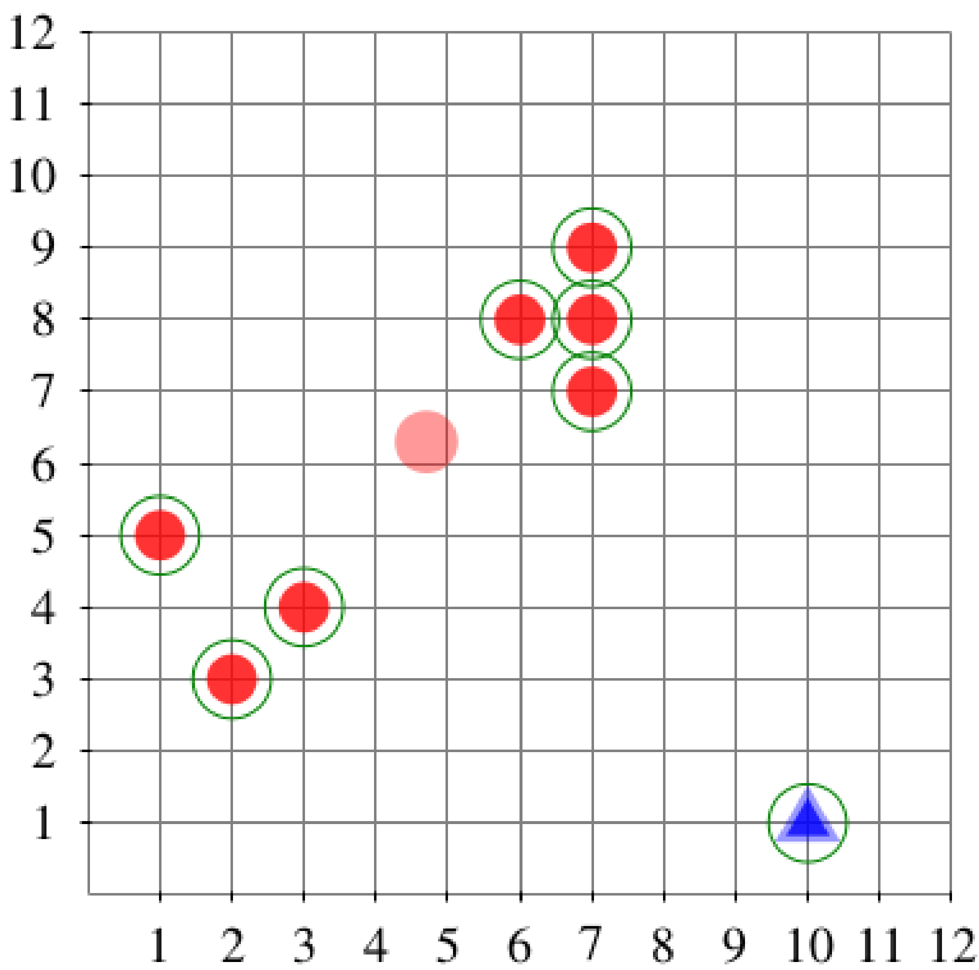
reasign more points

k-means Clustering – MacQueen Algorithm



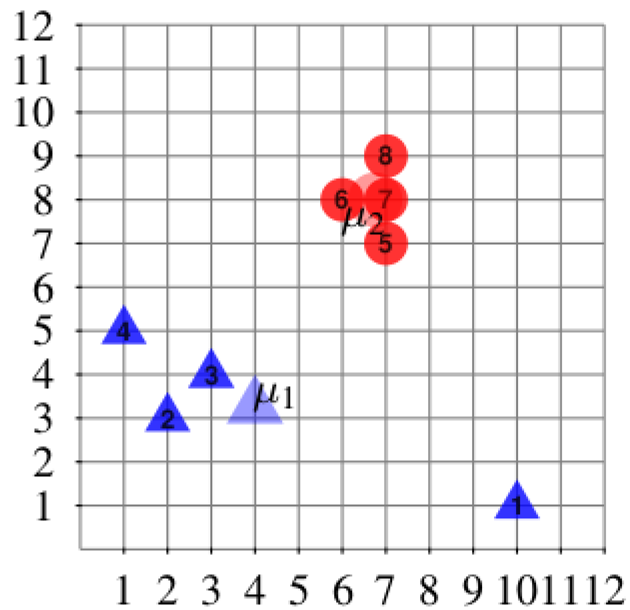
reasign more points
possibly more iterations

k-means Clustering – MacQueen Algorithm



reasign more points
possibly more iterations
convergence

k-means Clustering – Quality



$SSQ(\mu_1, p_1) = |4 - 10|^2 + |3.25 - 1|^2 = 36 + 5\frac{1}{16} = 41\frac{1}{16}$

$SSQ(\mu_1, p_2) = |4 - 2|^2 + |3.25 - 3|^2 = 4 + \frac{1}{16} = 4\frac{1}{16}$

$SSQ(\mu_1, p_3) = |4 - 3|^2 + |3.25 - 4|^2 = 1 + \frac{9}{16} = 1\frac{9}{16}$

$SSQ(\mu_1, p_4) = |4 - 1|^2 + |3.25 - 5|^2 = 9 + 3\frac{1}{16} = 12\frac{1}{16}$

$TD^2(C_1) = 58\frac{3}{4}$

$SSQ(\mu_2, p_5) = |6.75 - 7|^2 + |8 - 7|^2 = \frac{1}{16} + 1 = 1\frac{1}{16}$

$SSQ(\mu_2, p_6) = |6.75 - 6|^2 + |8 - 8|^2 = \frac{9}{16} + 0 = \frac{9}{16}$

$SSQ(\mu_2, p_7) = |6.75 - 7|^2 + |8 - 8|^2 = \frac{1}{16} + 0 = \frac{1}{16}$

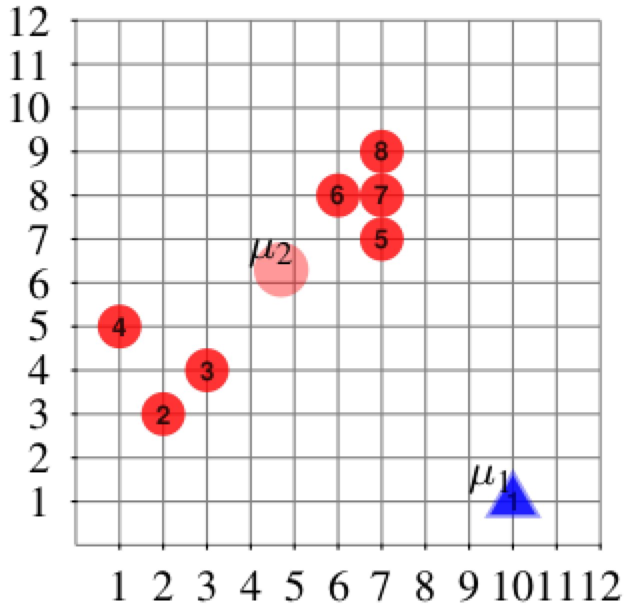
$SSQ(\mu_2, p_8) = |6.75 - 7|^2 + |8 - 9|^2 = \frac{1}{16} + 1 = 1\frac{1}{16}$

$TD^2(C_2) = 2\frac{3}{4}$

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



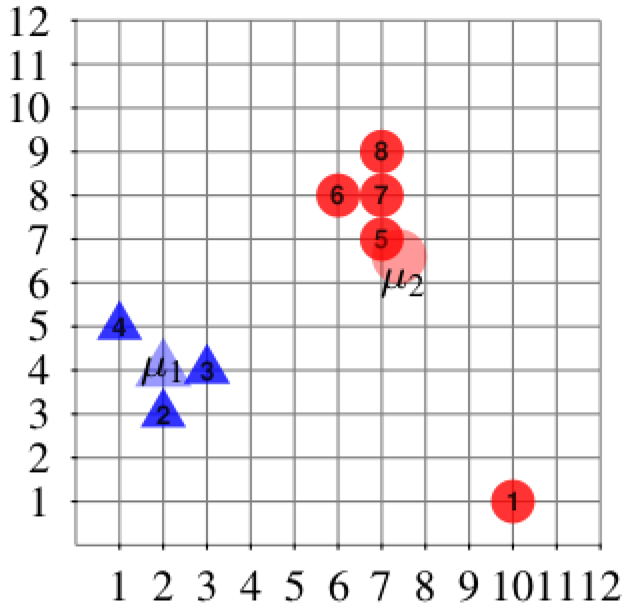
$SSQ(\mu_1, p_1) = |10 - 10|^2 + |1 - 1|^2 = 0$
$TD^2(C_1) = 0$

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

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

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

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

Clustering & Feature Spaces

Lecture Content

k-means: Pros

- Efficient: $O(k \cdot n)$ per iteration, number of iterations is usually in the order of 10.
- Easy to implement, thus very popular
- Only one parameter, easy to understand
- Well understood and researched

- Different variants exists
  - Fuzzy clustering
  - Variants without n-dimensional embedding

k-means: Disadvantages

- 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)

What to do?

How can we tackle the initialization problem?

What to do?

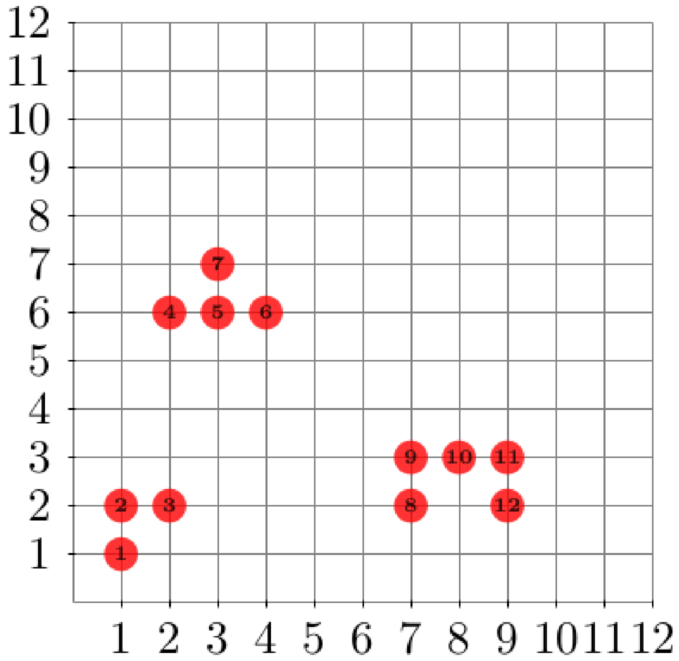
How can we tackle the initialization problem?

- Repeated runs
- **Furthest-first initialization**
- Subset Furthest-first initialization

Insertion: Furthest First Initialization

- Select a random point as start
- For each point, the minimum of its distances to the selected centers is maintained.
- While less than k points selected, repeat:
  - Selected point p with the maximum distance to the existing centers
  - Remove p from the not-yet-selected points and add it to the center points
  - For each remaining not-yet-selected point q, replace the distance stored for q by the minimum of its old value and the distance from p to q.

Furthest First Initialization: Visualization



Selected Centers: -

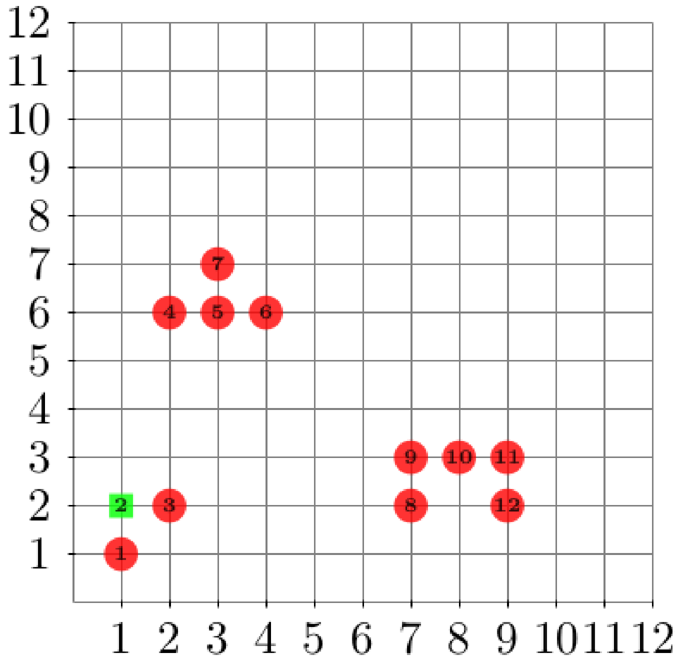
Distances:

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
-	-	-	-	-	-	-	-	-	-	-	-

Next Step:

Start by selecting the center randomly.

Furthest First Initialization: Visualization



Selected Centers: p2

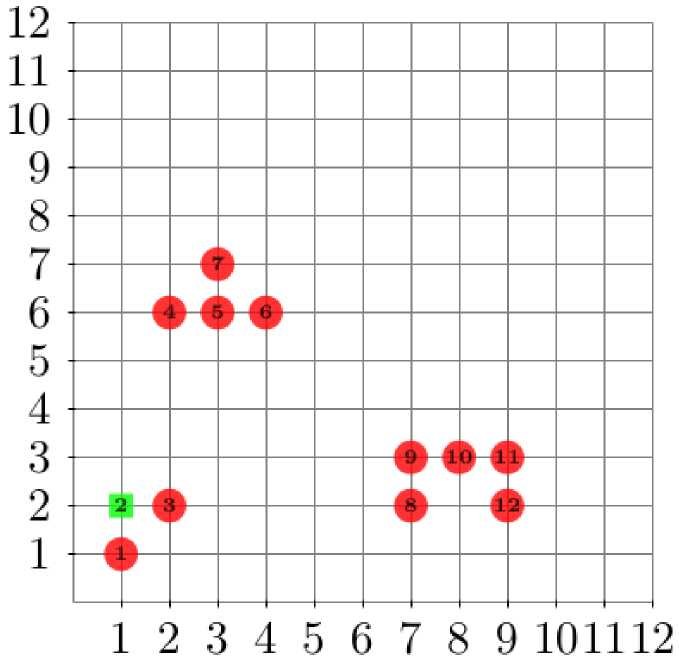
Distances:

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
-	X	-	-	-	-	-	-	-	-	-	-

Next Step:

Calculate the distances.

Furthest First Initialization: Visualization



Selected Centers: p2

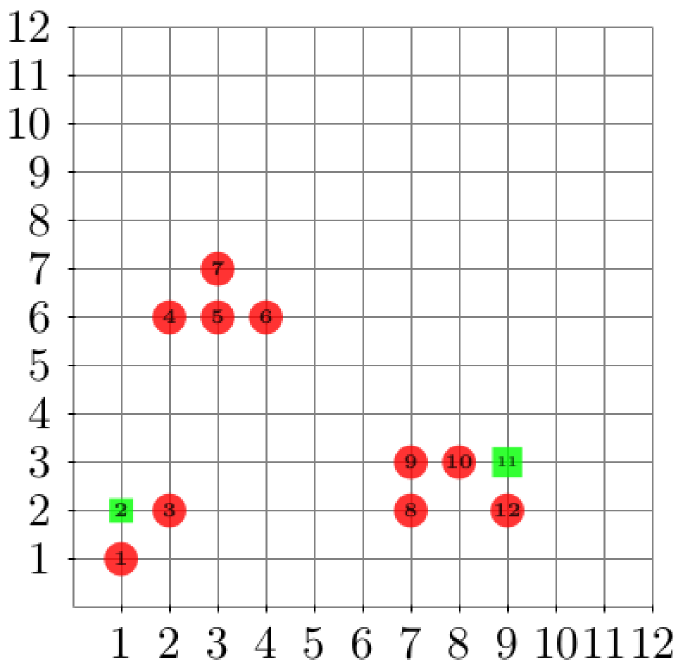
Distances:

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
1	X	1	4.1	4.5	5	5.3	6	6.1	7.1	8.1	8

Next Step:

Selected Furthest Point.

Furthest First Initialization: Visualization



Selected Centers: p2, p11

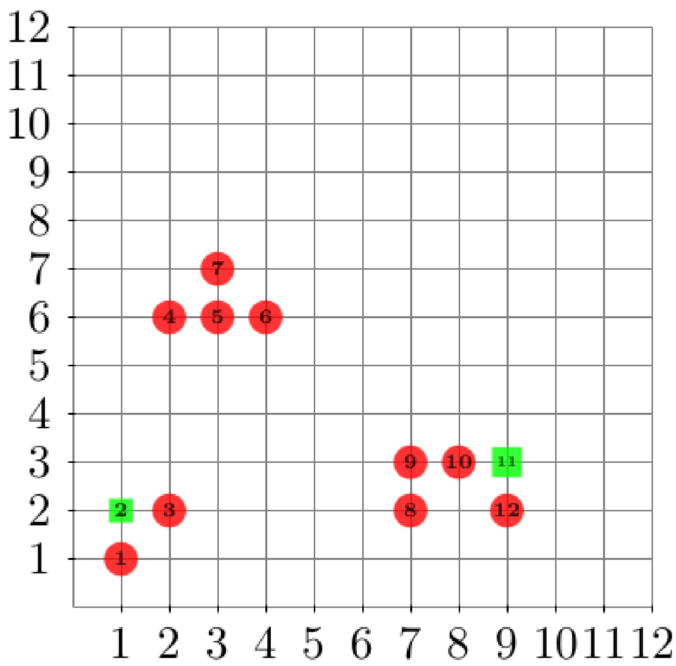
Distances:

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
1	X	1	4.1	4.5	5	5.3	6	6.1	7.1	X	8

Next Step:

Calculate the distance to new center.

Furthest First Initialization: Visualization



Selected Centers: p2, p11

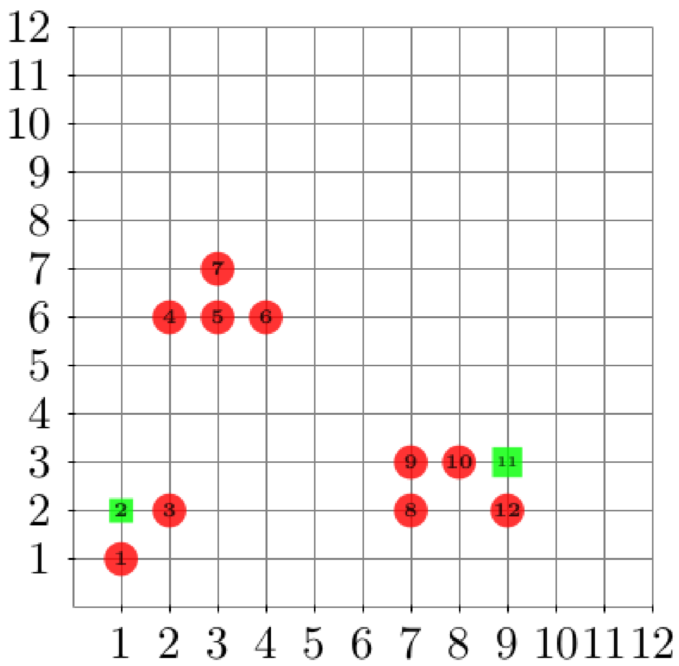
Distances:

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
1	X	1	4.1	4.5	5	5.3	6	6.1	7.1	X	8
8.2	X	7.1	7.6	6.7	5.8	7.2	2.1	2	1	X	1

Next Step:

Update the minimum distances

Furthest First Initialization: Visualization



Selected Centers: p2, p11

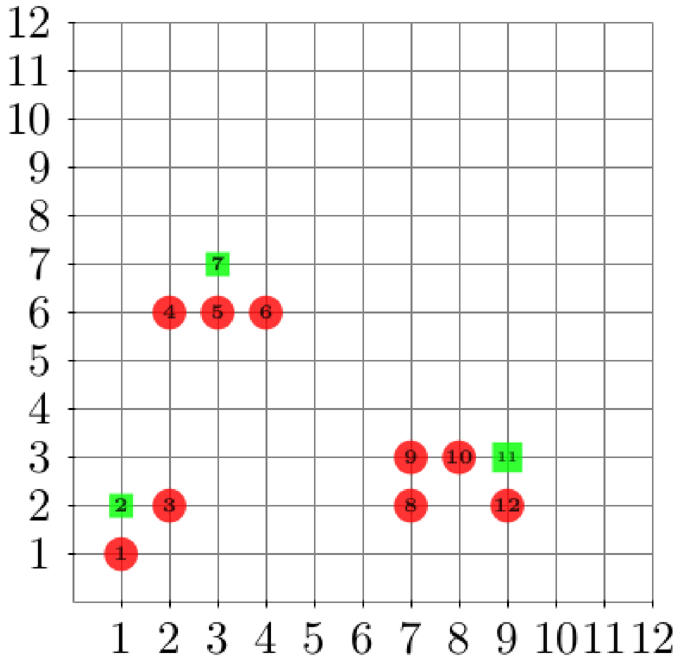
Distances:

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
1	X	1	4.1	4.5	5	5.3	2.1	2	1	X	1

Next Step:

Select next center.

Furthest First Initialization: Visualization



Selected Centers: p2, p11, p7

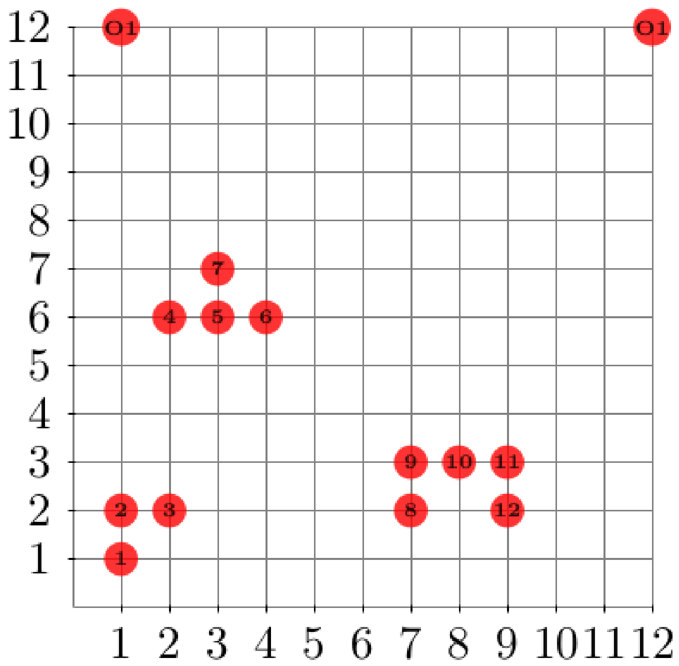
Distances:

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
1	X	1	4.1	4.5	5	X	2.1	2	1	X	1

Next Step:

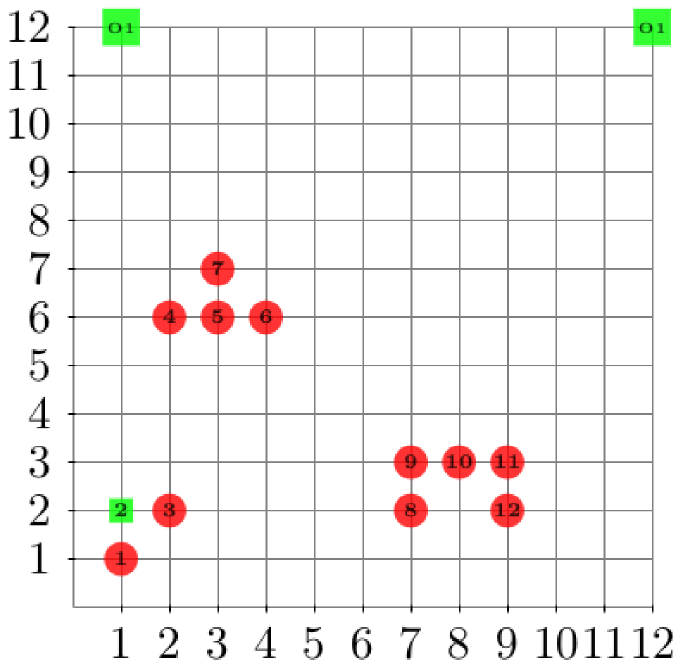
Repeat.

Furthest First Initialization: Visualization



Same situation as before with two outliers

Furthest First Initialization: Visualization



Outliers tend to be chosen

Learnings of this Section

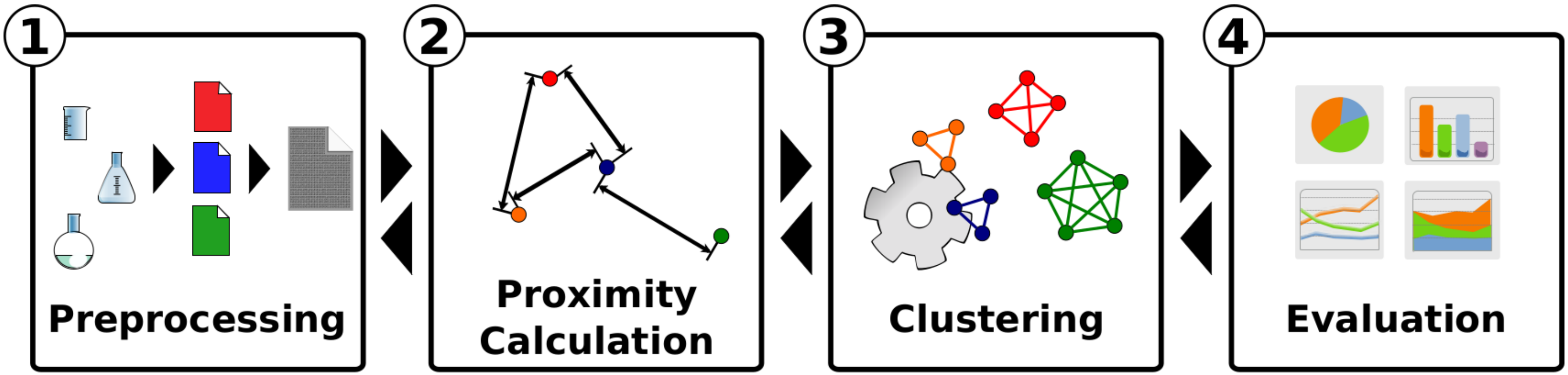
- What is Clustering?
- Basic idea for identifying “good” partitions into k clusters
- Selection of representative points
- Iterative refinement
- Local optimum
- k-means variants [Forgy, 1965, Lloyd, 1982, MacQueen, 1967]
- Different initialization methods

Clustering & Feature Spaces

Lecture Content

- Clustering
  - Clustering in General
  - Partitional Clustering
  - Visualization: Algorithmic Differences
  - Summary
- Feature Spaces
  - Distances
  - Features for Images
  - Summary

Recall: Clustering as a Workflow



1 Preprocessing
- Feature Selection
- Feature Extraction
- Normalization Standardization
- Missing Values

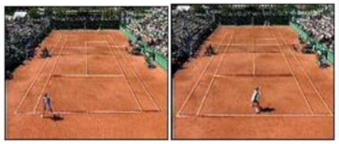
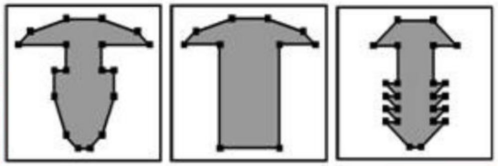
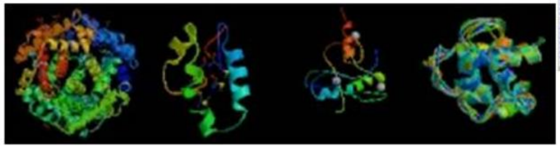
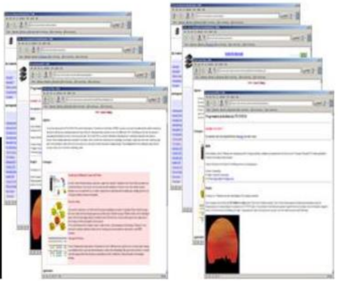
2 Proximity Calculation
- Continous Data
- Categorical Data
- Mixed Data
- Specialized Measures

3 Clustering
- k-Means-Based
- Hierarchical
- Graph-Based
- Density-Based
- Model-Based
- Large-Scale

4 Evaluation
- Internal Indices
- External Indices
- Relative Indices
- Parameter Optimization

Similarities

- 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
  - . . .

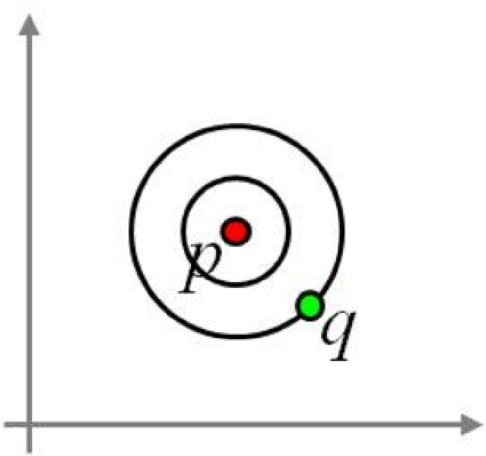





Spaces and Distance Functions

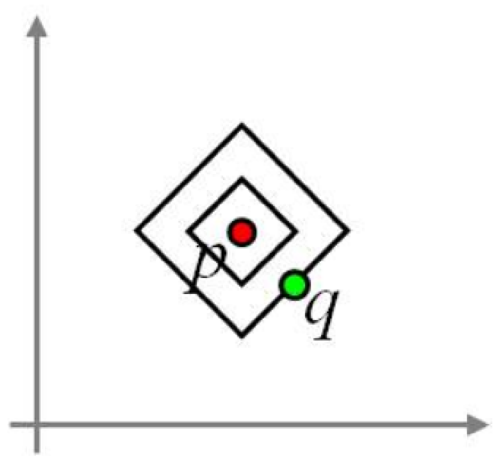
Common distance measure for (Euclidean) feature vectors: L_P-norm

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

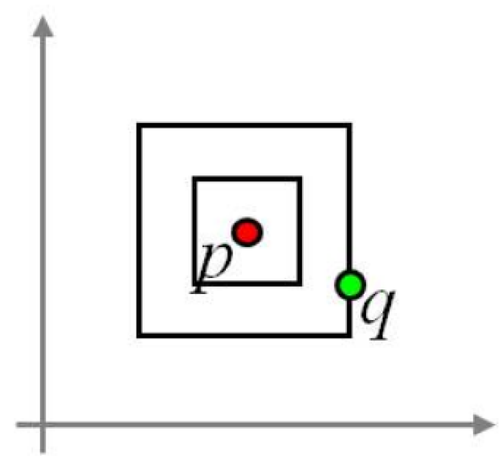
Euclidean norm (L_2):



Manhattan norm (L_1):



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



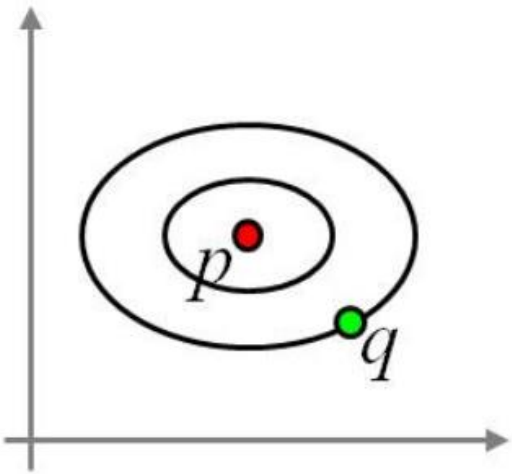
Spaces and Distance Functions

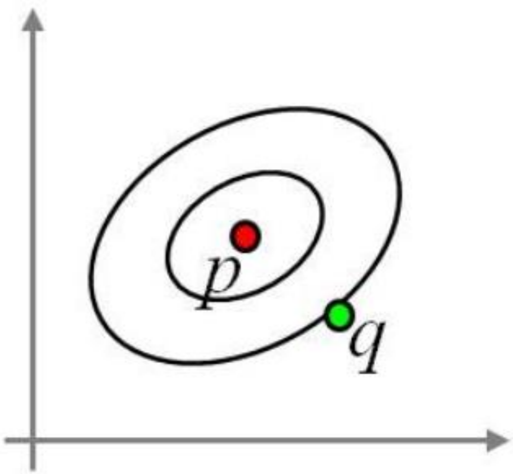
weighted Euclidean norm:

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

quadratic form:

$$\mathrm{dist}(p, q) = \left((p - q)M(p - q)^{\mathsf{T}}\right)^{\frac{1}{2}}$$





Clustering & Feature Spaces

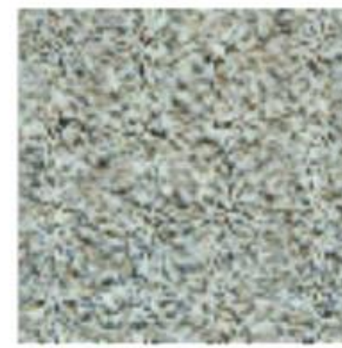
Lecture Content

- Clustering
  - Clustering in General
  - Partitional Clustering
  - Visualization: Algorithmic Differences
  - Summary
- Feature Spaces
  - Distances
  - Features for Images
  - Summary

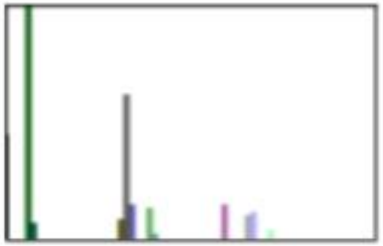
Categories of Feature Descriptors for Images

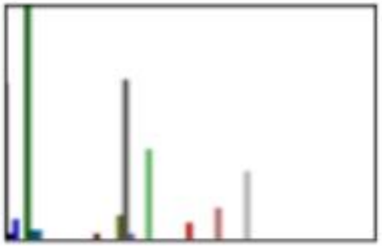
- Distribution of colors
- Texture
- Shapes (contours)
- Many more …



Color Histogram

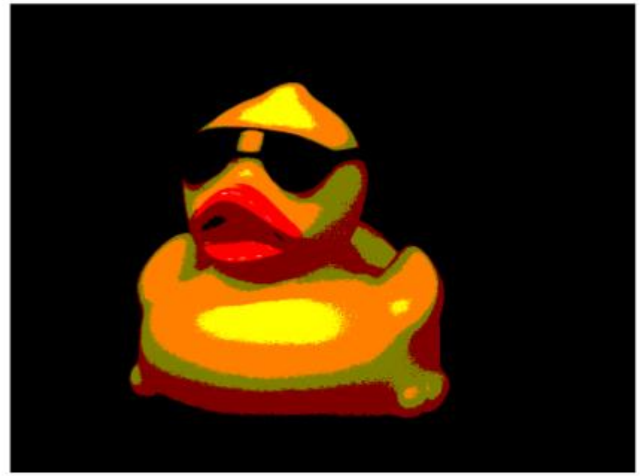


- A histogram represents the distribution of colors over the pixels of an image
- Definition of an color histogram:
  - Choose a color space (RGB, HSV, HLS, . . .)
  - Choose number of representants (sample points) in the color space
- Possibly normalization (to account for different image sizes)

Impact of Number of Representants

Impact of Number of Representants

2^3

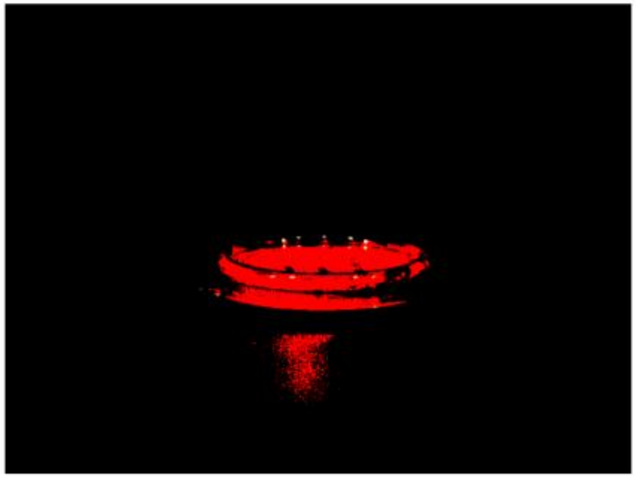


3^3

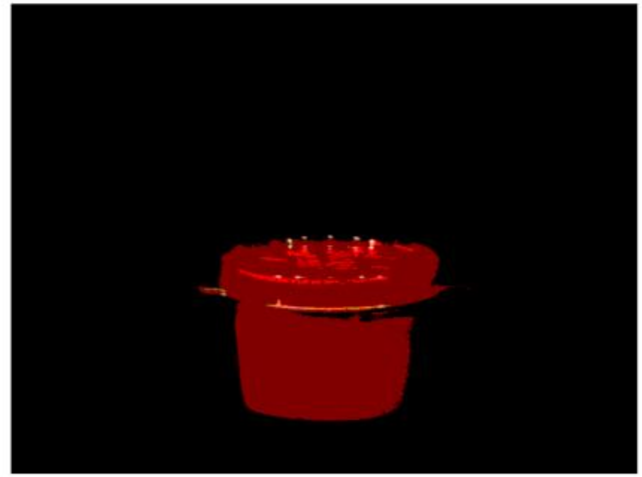
4^3

16^3

Impact of Number of Representants



2^3

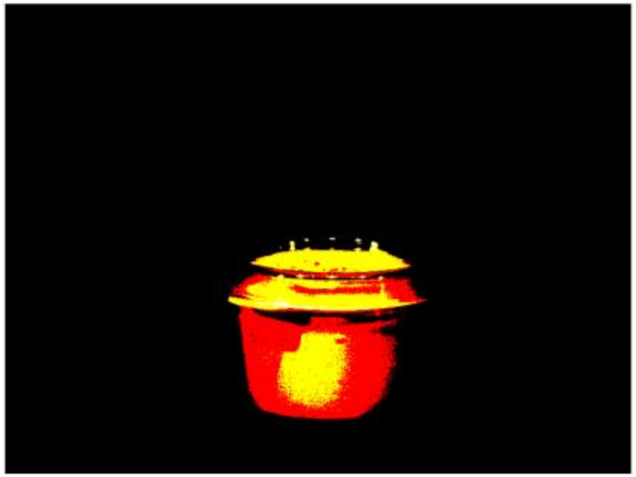


3^3

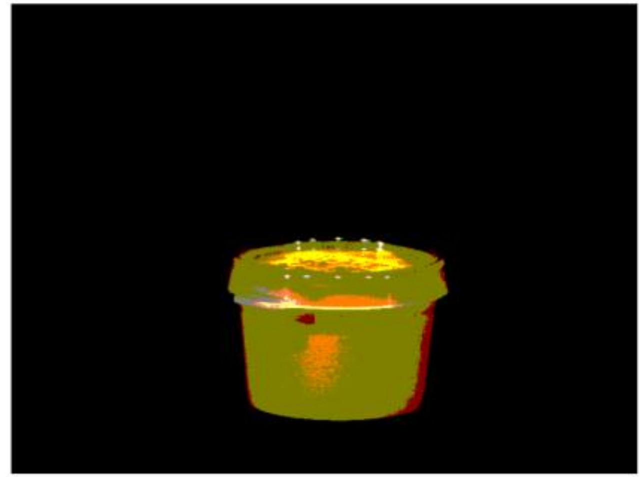
4^3

16^3

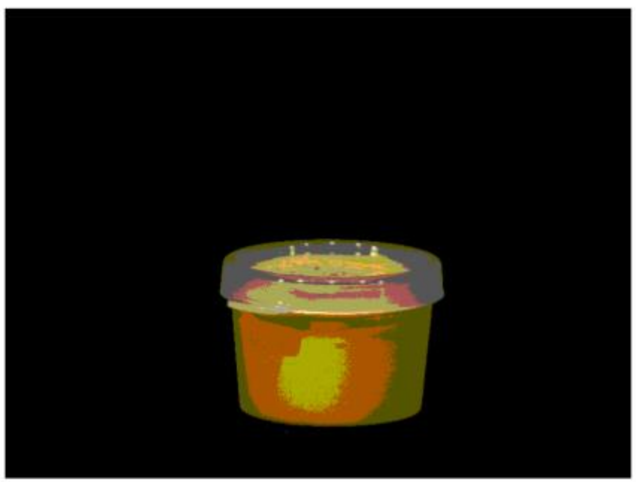
Impact of Number of Representants



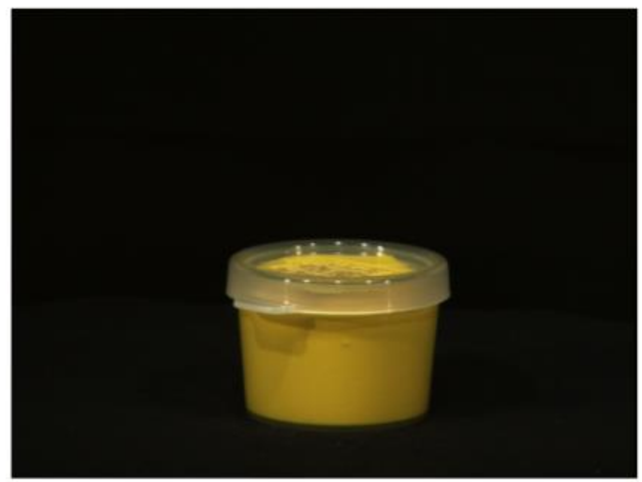
2^3



3^3



4^3



16^3

Impact of Number of Representants

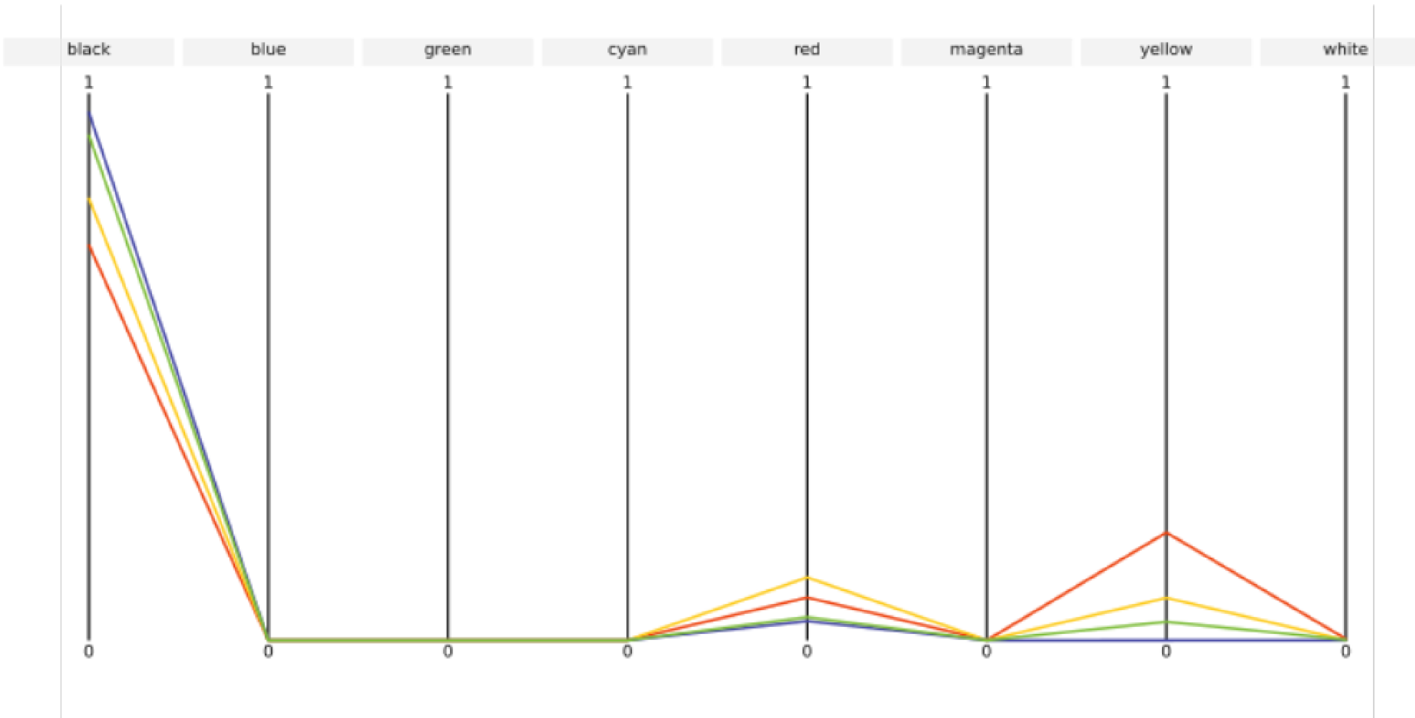
2^3

3^3

4^3

16^3

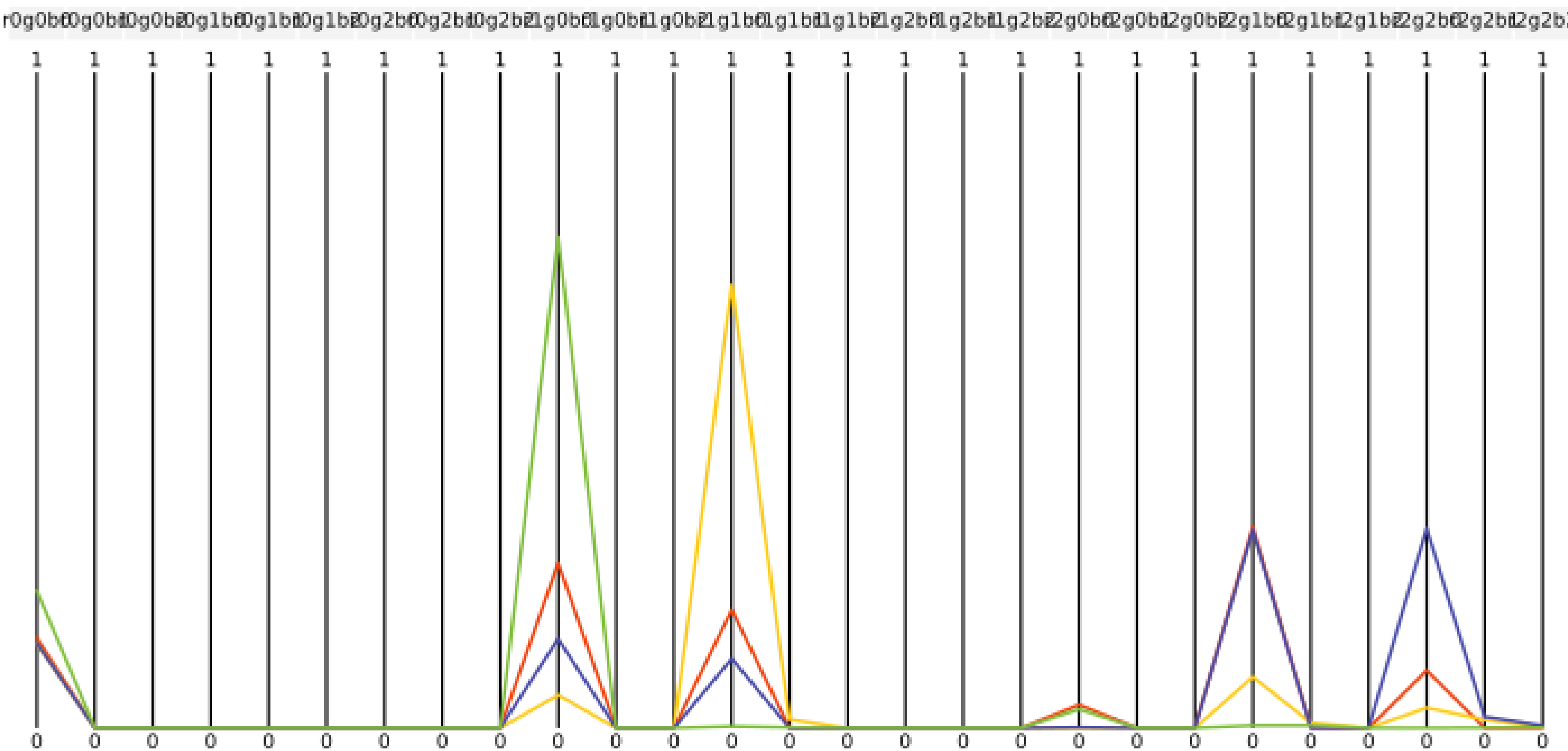
Impact of Number of Representants



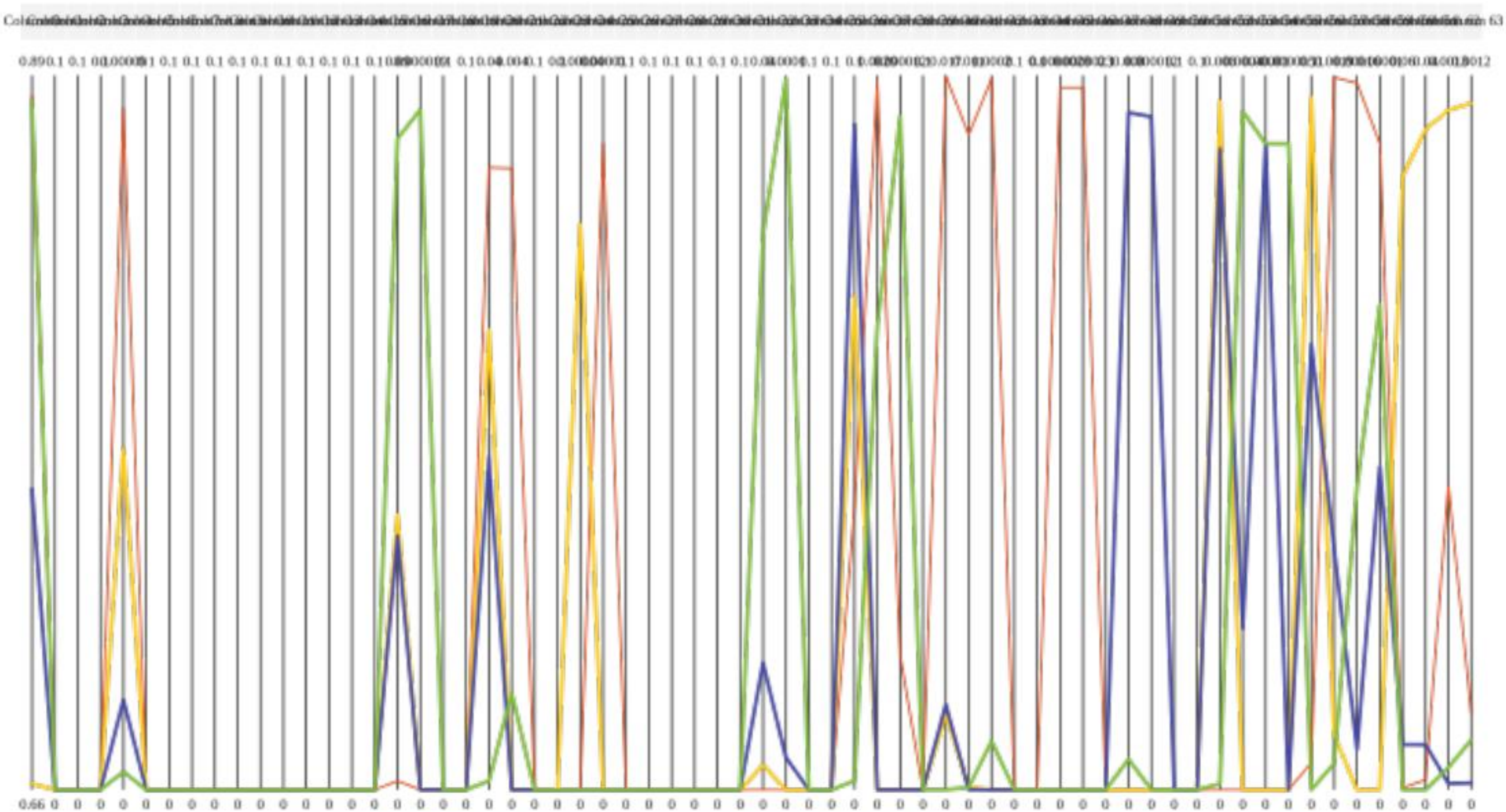
The histogram for each image is essentially a visualization of a vector:

$(0.77, 0, 0, 0, 0.08, 0, 0.15, 0)$ $(0.8, 0, 0, 0, 0.11, 0, 0.09, 0)$
$(0.9, 0, 0, 0, 0.05, 0, 0.05, 0)$ $(0.955, 0, 0, 0, 0.045, 0, 0, 0)$

Impact of Number of Representants



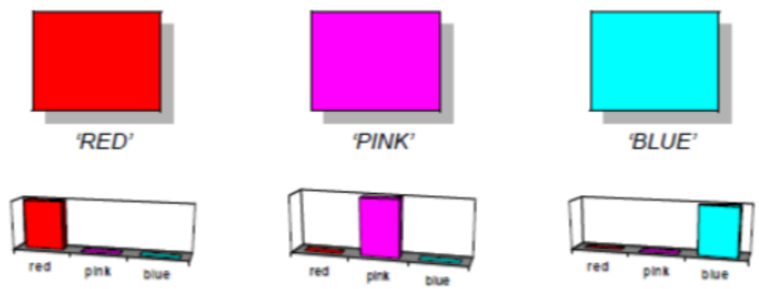
Impact of Number of Representants



Distances for Color Histograms

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

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



$(1, 0, 0)$ $\quad$ $(0, 1, 0)$ $\quad$ $(0, 0, 1)$

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

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

Distances for Color Histograms

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

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

Distances for Color Histograms

Quadratic form with 'psychological' similarity matrix

$$A = \begin{bmatrix} 1 & a_{12} & \dots & \\ a_{21} & 1 & \dots & \\ \vdots & & \ddots & \vdots \\ & & \dots & 1 \end{bmatrix}$$ where a_{ij} ($\stackrel{?}{=} a_{ji}$) describe the

subjective similarity of the features i and j in the color histogram:

$$\mathrm{dist}_A(P, Q) = \sqrt{(h_P - h_Q) \cdot A \cdot (h_P - h_Q)^\mathsf{T}}$$

$$A' = \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{dist}(\mathsf{RED}, \mathsf{PINK}) = \sqrt{0.2}$$
$$\mathrm{dist}(\mathsf{RED}, \mathsf{BLUE}) = \sqrt{2}$$
$$\mathrm{dist}(\mathsf{BLUE}, \mathsf{PINK}) = \sqrt{2}$$

Clustering & Feature Spaces
Lecture Content

- Clustering
  - Clustering in General
  - Partitional Clustering
  - Visualization: Algorithmic Differences
  - Summary
- Feature Spaces
  - Distances
  - Features for Images
  - Summary

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
  - For proteins: Edit distance, structure, …
- With normalization: Canberra, …
- Quadratic forms / bilinear forms: $d(x, y) := x^T M y$ for some positive (usually symmetric) definite matrix M.

Learnings of this Section

- Distances (L_p-norms, weighted, quadratic form)
- Color histograms as feature (vector) descriptors for images
- Impact of the granularity of color histograms on similarity measures