COMP229: Introduction to Data Science
Lecture 18: Lloyds k-means clustering

Olga Anosova, O.Anosova@liverpool.ac.uk
Autumn 2023, Computer Science department
University of Liverpool, United Kingdom

Lecture plan

- k-means clustering method
- Lloyds algorithm
- Voronoi cell and diagram
- How to choose k?

Reminder: clustering

- Clustering is grouping a set of objects, it requires choice of measurement for similarity and dissimilarity.
- Each clustering is problem-specific.
- The are multiple types of clustering to choose from:
  - connectivity-based (hierarchical)
  - centroid-based (k-means)
  - density-based (DBSCAN)
  - distribution-based.
- Each of clustering methods depends on the cheoice of the distance (or similarity measure).

The centre of a point cloud in $\mathbb{R}^m$

Definition 18.1. For any cloud (cluster) $C \subset \mathbb{R}^m$ of $|C|$ points, its **centre** is $\vec{c} = \frac{1}{|C|} \sum_{p \in C} \vec{p}$, where each $p \in C$ is considered as a vector $\vec{p} \in \mathbb{R}^m$.

This centre is a high-dimensional analogue of the sample mean.

Theorem 18.2. The centre $\vec{c}$ of $C \subset \mathbb{R}^m$ minimises the sum of squared distances $f(c) = \sum_{p \in C} |\vec{c} - \vec{p}|^2$.

Proof. At the global minimum $\vec{c} = (c_1, \ldots, c_m)$, the partial derivative for the j-th coordinate vanishes:

$$\frac{\partial f}{\partial c_j} = \sum_{p \in C} 2(c_j - p_j) = 0, \; |C| c_j = \sum_{p \in C} p_j, \; \vec{c} = \frac{1}{|C|} \sum_{p \in C} \vec{p}.$$

Centres of clouds in $\mathbb{R}^m$

Example 18.3. Any cloud $C = \{p, q\} \subset \mathbb{R}^m$ has the centre at the mid-point $0.5(\vec{p} + \vec{q})$ of $[p, q]$.

Any $C = \{p, q, r\} \subset \mathbb{R}^2$ has the centre at the point $\dfrac{\vec{p} + \vec{q} + \vec{r}}{3}$ called **the barycentre** of the triangle.

Problem 18.4. Find the centre of this cloud in $\mathbb{R}^2$
$(3, 2)$, $(-4, -1)$, $(1, -5)$, $(-1, -4)$, $(2, -3)$, $(4, 1)$, $(-5, 4)$, $(-3, 5)$, $(5, -2)$, $(-2, 3)$.

Solution 18.4. The centre of the cloud is $(0, 0)$.

The k-means clustering problem

Definition 18.5. For any given k, the k-**means clustering** aims to split a cloud $C \subset \mathbb{R}^m$ into disjoint clusters $C_1, \ldots, C_k$ to minimise $\sum_{i=1}^{k} \sum_{p \in C_i} |\vec{c_i} - \vec{p}|^2$, where $\vec{c_i}$ is the centre of the i-th cluster $C_i \subset C$.

The k-means problem is to find an optimal splitting of n points into k disjoint subsets. An optimal solution is computationally expensive.

Standard Lloyd's heuristic algorithm

is the simplest algorithm that naively approximates an optimal splitting.

This fast *heuristic* algorithm has no guarantees of optimality and was invented 50+ years ago when a 'computer' was a job title, not a hardware device.

Improvements to this algorithm are still researched.

Pipeline of Lloyd's algorithm

Input: a cloud $C \subset \mathbb{R}^m$, a number of k of clusters.

Initialisation stage : choose initial centres.

Iteration stage : the assignment step and update step are repeated one after another many times.

Termination : many criteria are possible to decide when the iterations stop, e.g. centres are no longer updated or a max number of iterations are reached.

Output : k clusters, e.g. as lists of points from C.

Initialisation in Lloyd's algorithm

Here are the common methods, others are possible.

The *Forgy* method chooses k initial centres as random points of C (usually spread out), e.g. *k-farthest* points: choose a random seed point at random, the 1st first centre is the farthest from the seed, the 2nd centre is farthest from the 1st etc.

The *Random Partition* method randomly splits a cloud into k subsets and takes their centres, which often turn out to be close to the centre of C.

The neighbourhood of a centre

After initial centres are chosen, the clusters are formed as neighbouhoods of these centres.

Definition 18.6. For a cloud $C \subset \mathbb{R}^m$ and given centres $c_1, \ldots, c_k \in \mathbb{R}^m$, the *neighbourhood* of c_i is

$$N(c_i) = \{p \in C \ : \ |\vec{c_i} - \vec{p}| \leqslant |\vec{c_j} - \vec{p}| \text{ for } j \neq i\}.$$

If a point $p \in C$ is the mid-point between c_i, c_j, one can make a random (or other) choice for p.

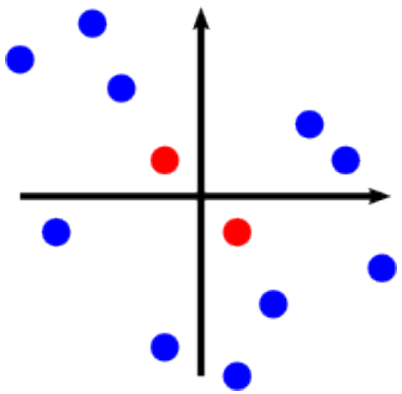
The distances (lengths of vectors) are Euclidean. Other choices, e.g. L_s-metrics are possible.

Finding a neighbourhood

Problem 18.7. Find neighbourhoods of $(1,-1)$ and $(-1,1)$ in the cloud: $(3,2),(-4,-1),(1,-5)$, $(-1,-4),(2,-3),(4,1),(-5,4),(-3,5),(5,-2),(-2,3)$.

Solution 18.7. We just need to do all comparisons, for example

$$L_2((-1,1),(3,2)) = \sqrt{17} > L_2((1,-1),(3,2)) = \sqrt{13}$$



The neighbourhood of the centre $(-1,1)$ consists of $(-4,-1),(-2,3),(-5,4),(-3,5)$, while $(1,-1)$ attracts all others.

The iteration stage

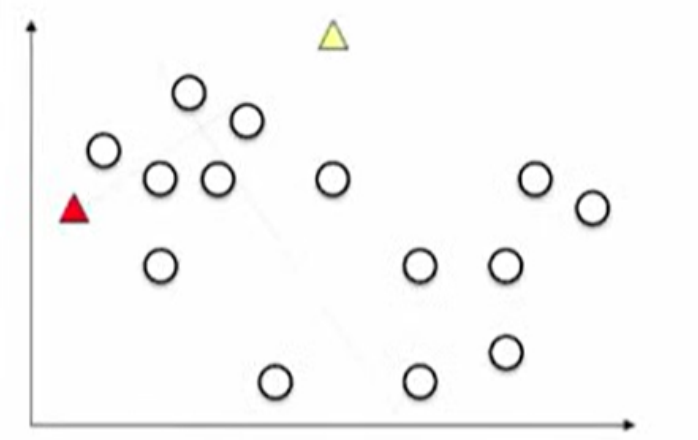
Alternate between the two steps below.

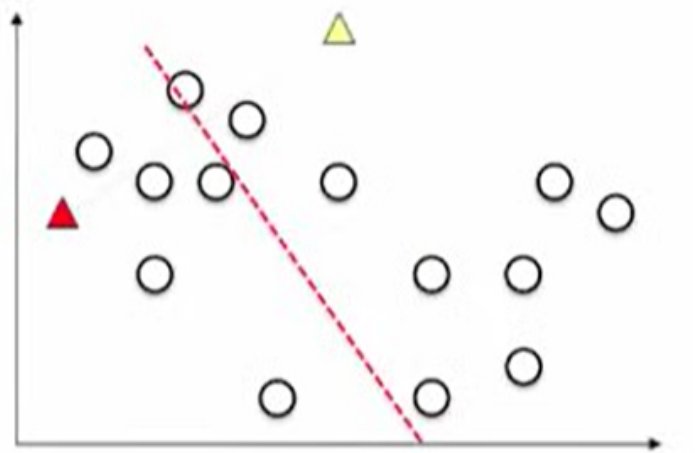
1) **Assignment**: for current centres $c_1, \ldots, c_k$, split the cloud C into their neighbourhoods $N(c_1), \ldots, N(c_k)$

2) **Update**: for any cluster C_i, its centre is updated as the mean point: $\vec{c_i} = \frac{1}{|C_i|} \sum_{p \in C_i} \vec{p}$.

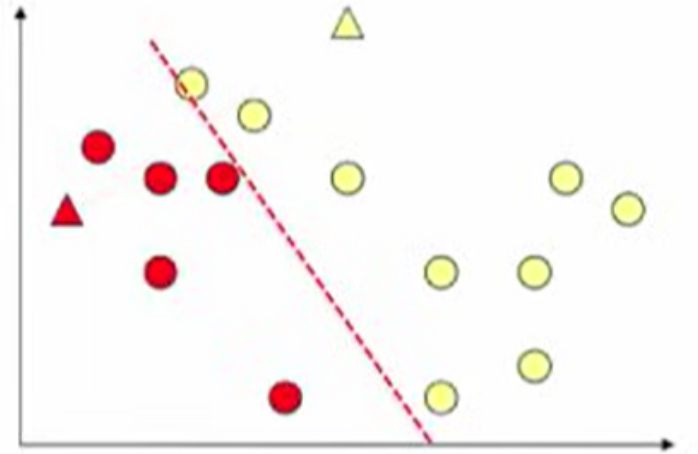
We might stop when the centres (or clusters) no longer change. There is no guarantee that a final clustering minimises the function $\sum_{i=1}^{k} \sum_{p \in C_i} |\vec{c_i} - \vec{p}|^2$.

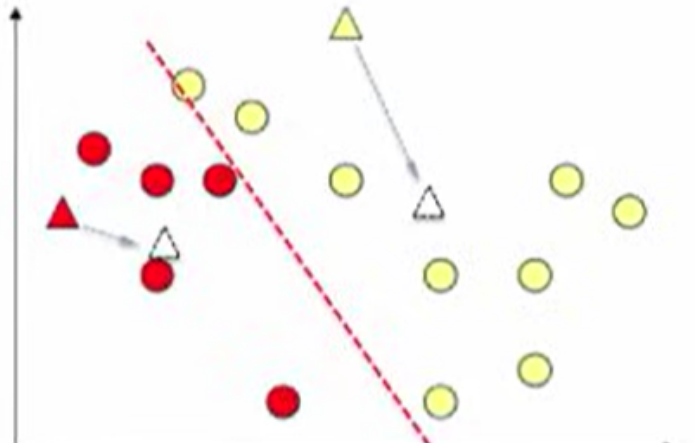
Example





Update the clusters and the centers:





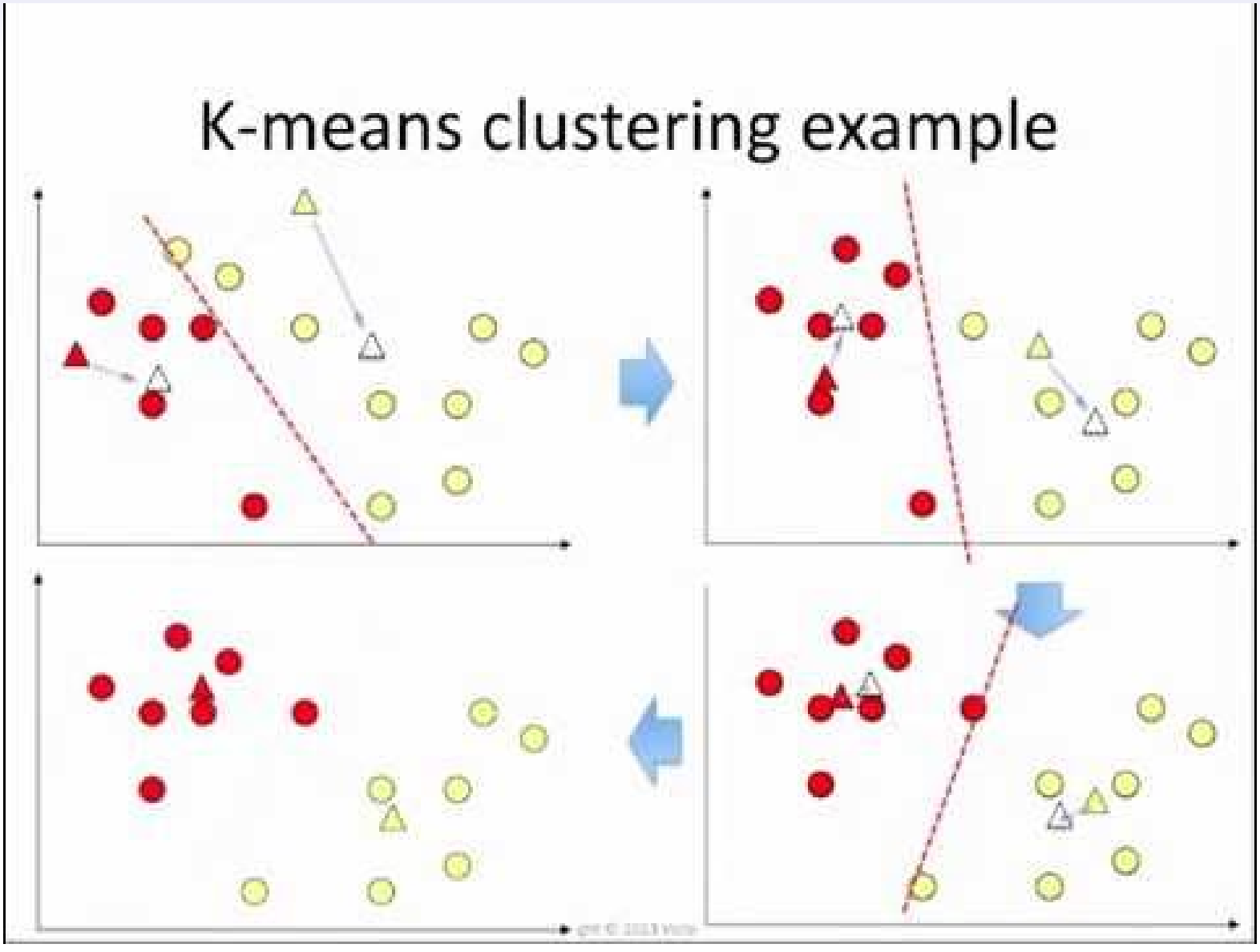
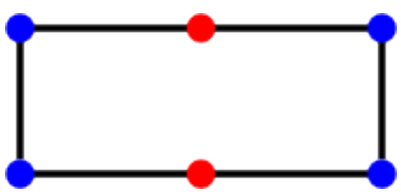


Image by Victor Lavrenko http://bit.ly/K-means

Arbitrarily bad Lloyd's output

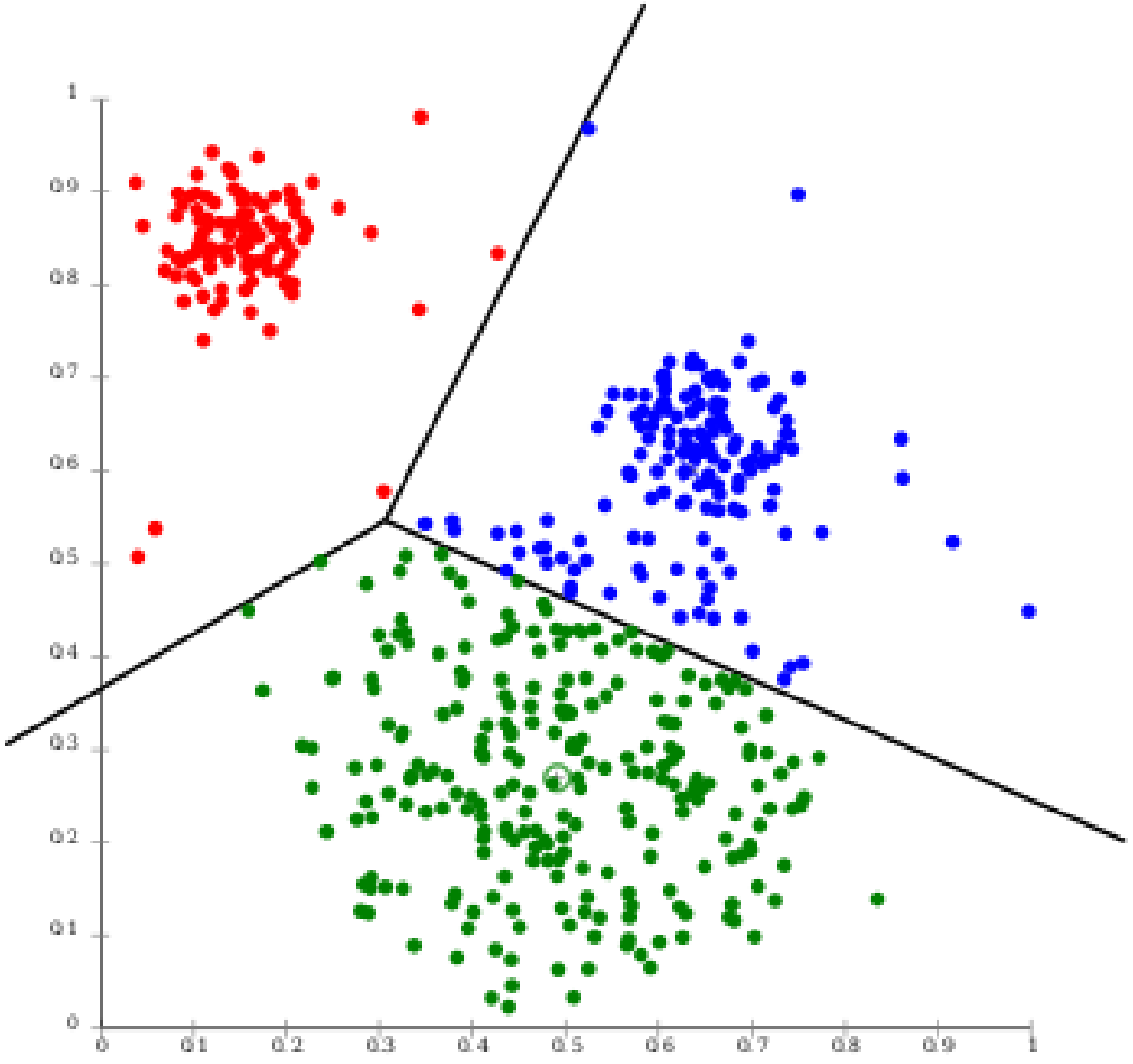
Different initial centres may lead to different outputs, so k-means is often run with different initialisations.

Let the cloud C be 4 vertices of an axes-aligned rectangle. If 2 initial centres are the mid-points of the horizontal edges, then 2-means outputs these two centres without iterations.



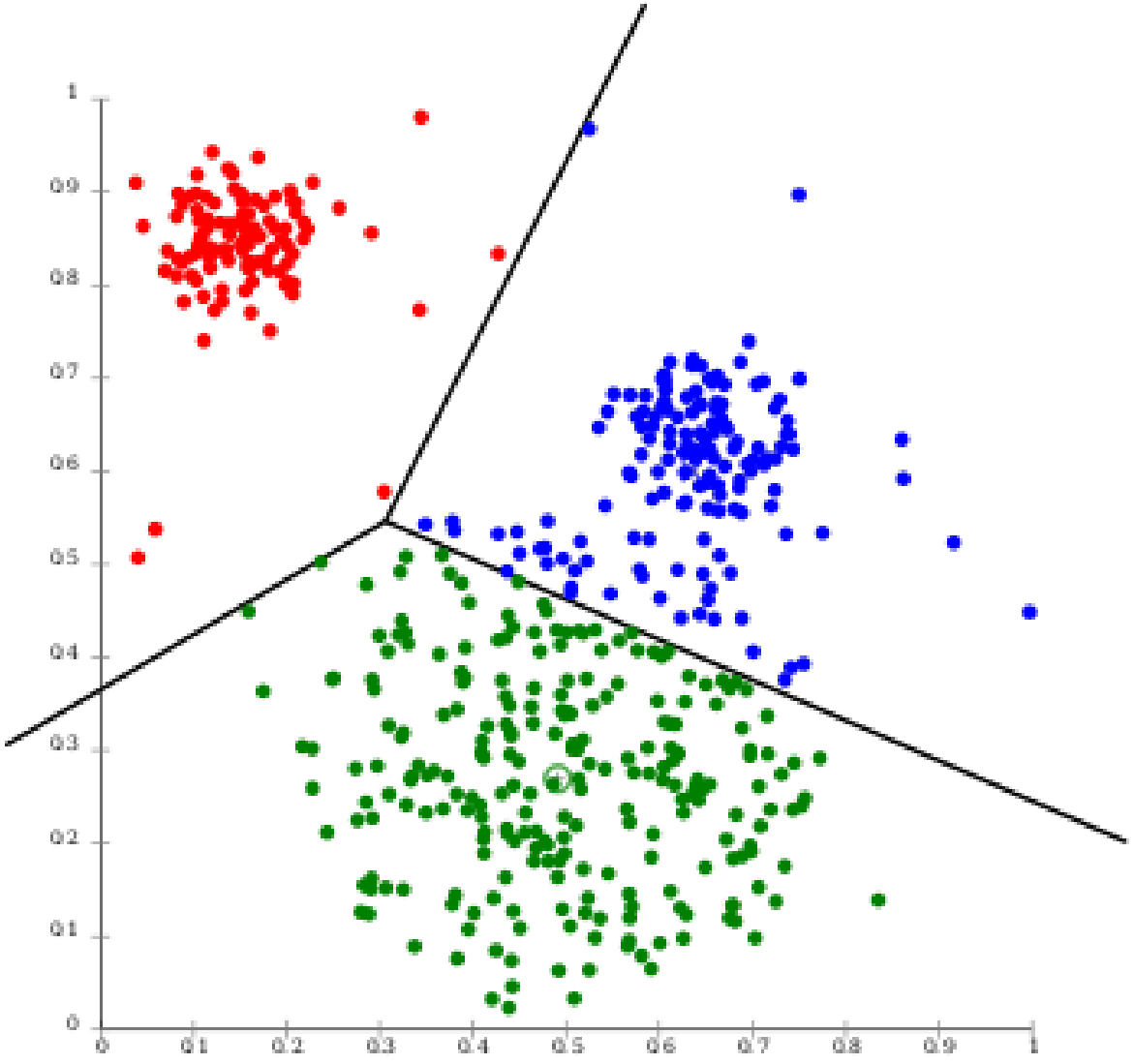
If the horizontal sides are much longer than the vertical ones, this output is far away from the optimal clustering with centres at the mid-points of the short vertical edges.

k-means partitions the data space



The assignment step of Lloyd's algorithm splits the data cloud into neighbourhoods of cluster centres (three subclouds in the picture).

Bisectors between centres



The black lines are bisectors (the lines of points that have the same distances to the centres), which splits $\mathbb{R}^m$ in *Voronoi* cells.

The Voronoi cell of a point

Definition 18.8. For centres $C = \{c_1, \ldots, c_k\}$ in $\mathbb{R}^m$ and a centre $c_i \in C$, the **Voronoi cell** is

$$V(c_i) = \{q \in \mathbb{R}^m \,:\, |\vec{c}_i - \vec{q}| \leqslant |\vec{c}_j - \vec{q}| \text{ for } j \neq i\}.$$

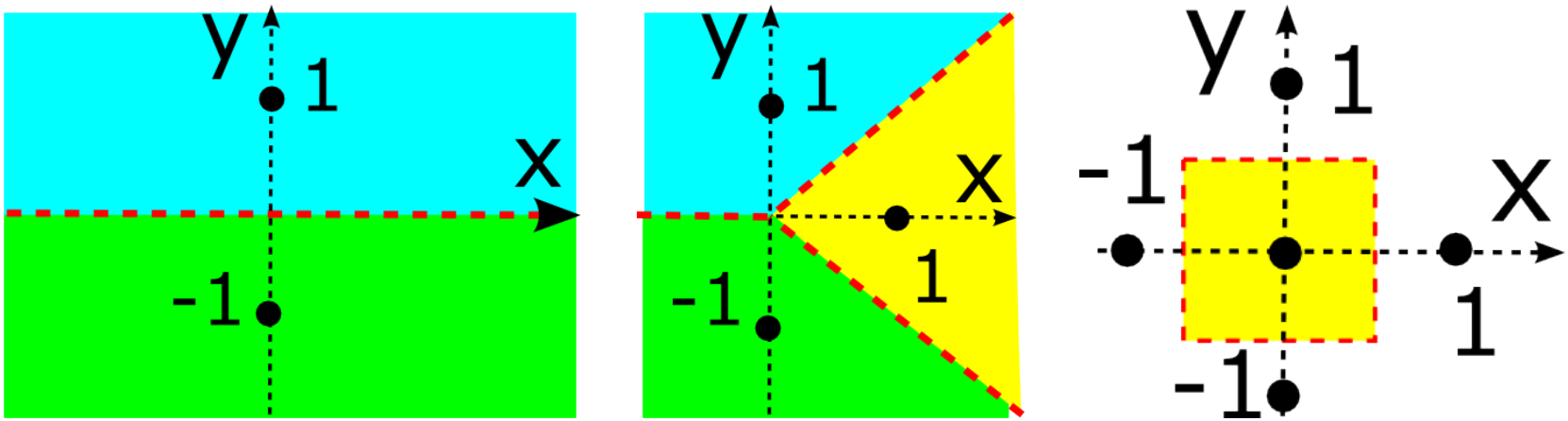
$V(c_i)$ is the neighbourhood of all points $q \in \mathbb{R}^m$ that are closer to c_i than to other centres $c_j \in C$.

The Voronoi cell $V(c_i)$ covers the neighbourhood around a centre c_i from k-means, but $V(c_i)$ is larger as an infinite subset (possibly unbounded) in $\mathbb{R}^m$.

How to draw Voronoi cells

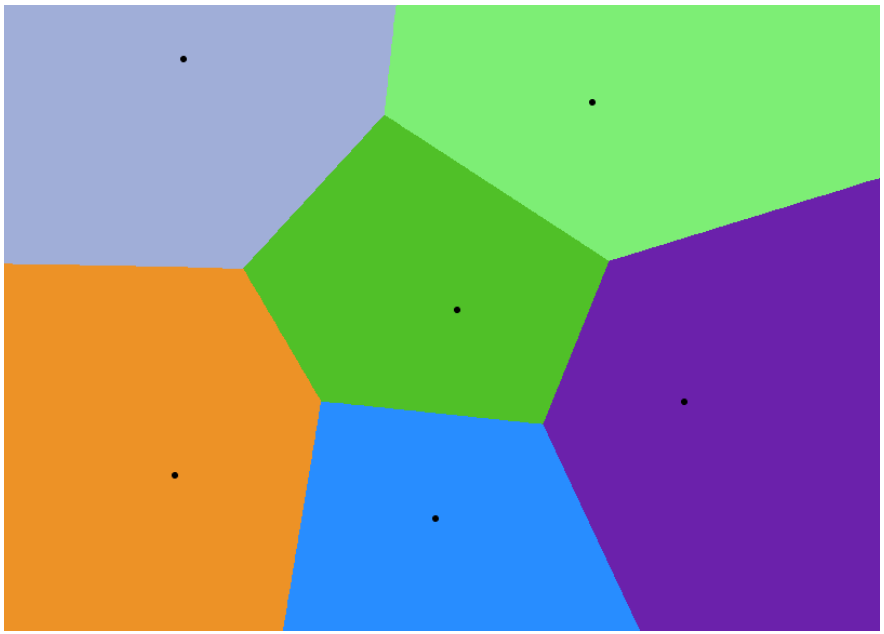
Problem 18.9. Draw Voronoi cells for sets
$\{(0, \pm 1)\}$, $\{(0, \pm 1), (1, 0)\}$, $\{(0, \pm 1), (\pm 1, 0), (0, 0)\}$.

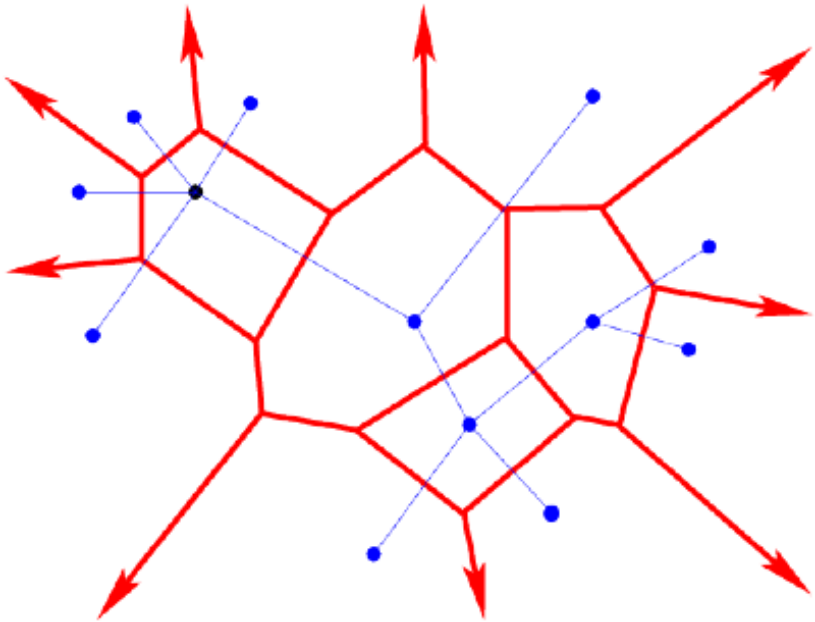
Solution 18.9. For the 2-point cloud $(0, \pm 1)$, the cell $V(0, 1)$ is the upper half-plane including the boundary x-axis, $V(0, -1)$ is the lower half-plane.



The Voronoi diagram of a set C

Definition 18.10. For a set $C = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^m$, the **Voronoi diagram** $V(C)$ is the splitting of $\mathbb{R}^m$ into the Voronoi cells $V(p_i)$, $i = 1, \ldots, n$.

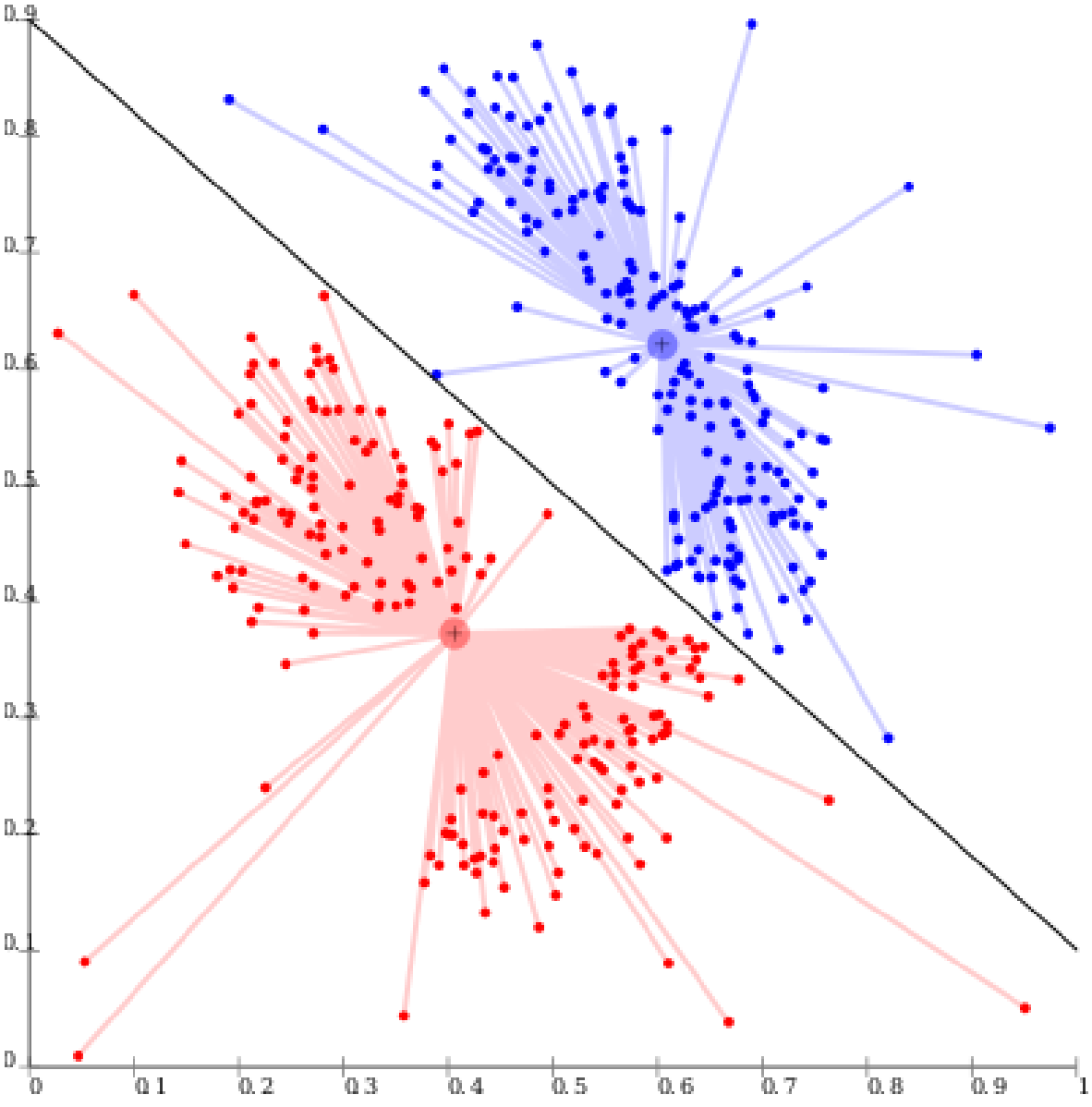




Other uses of Voronoi diagrams

- Biology & medicine: to model cells & tissues.
- Ecology: to study the growth patterns of forests.
- Computational chemistry: to compute atomic charges by the positions of the nuclei in a molecule.
- Networking: to structure a wireless network.
- Autonomous robot navigation: to find routes by setting obstacles as the centres, with the edges being the routes furthest from collisions.
- Computer graphics: Voronoi partition in movies.

k-means requires a good value of k

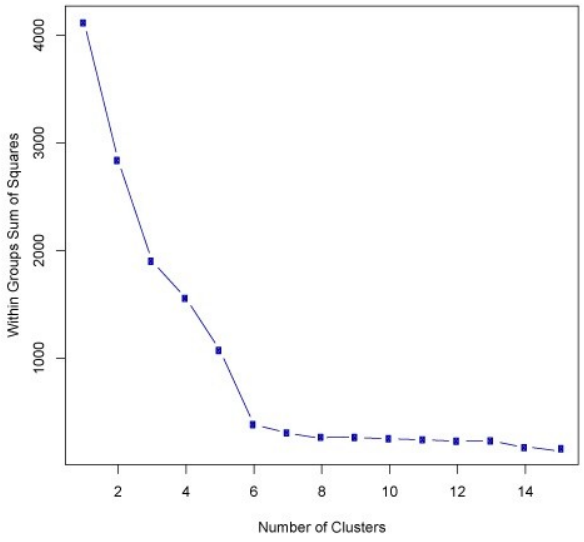


The key drawback of k-means: we need a good value of k. The cloud in the picture has 3 high-density regions, but 2-means clustering outputs only 2 bad clusters if $k = 2$ is chosen.

Finding a good value of k

For which k is the sum $\sum_{i=1}^{k} \sum_{p \in C_i} |\vec{c_i} - \vec{p}|^2$ optimized?

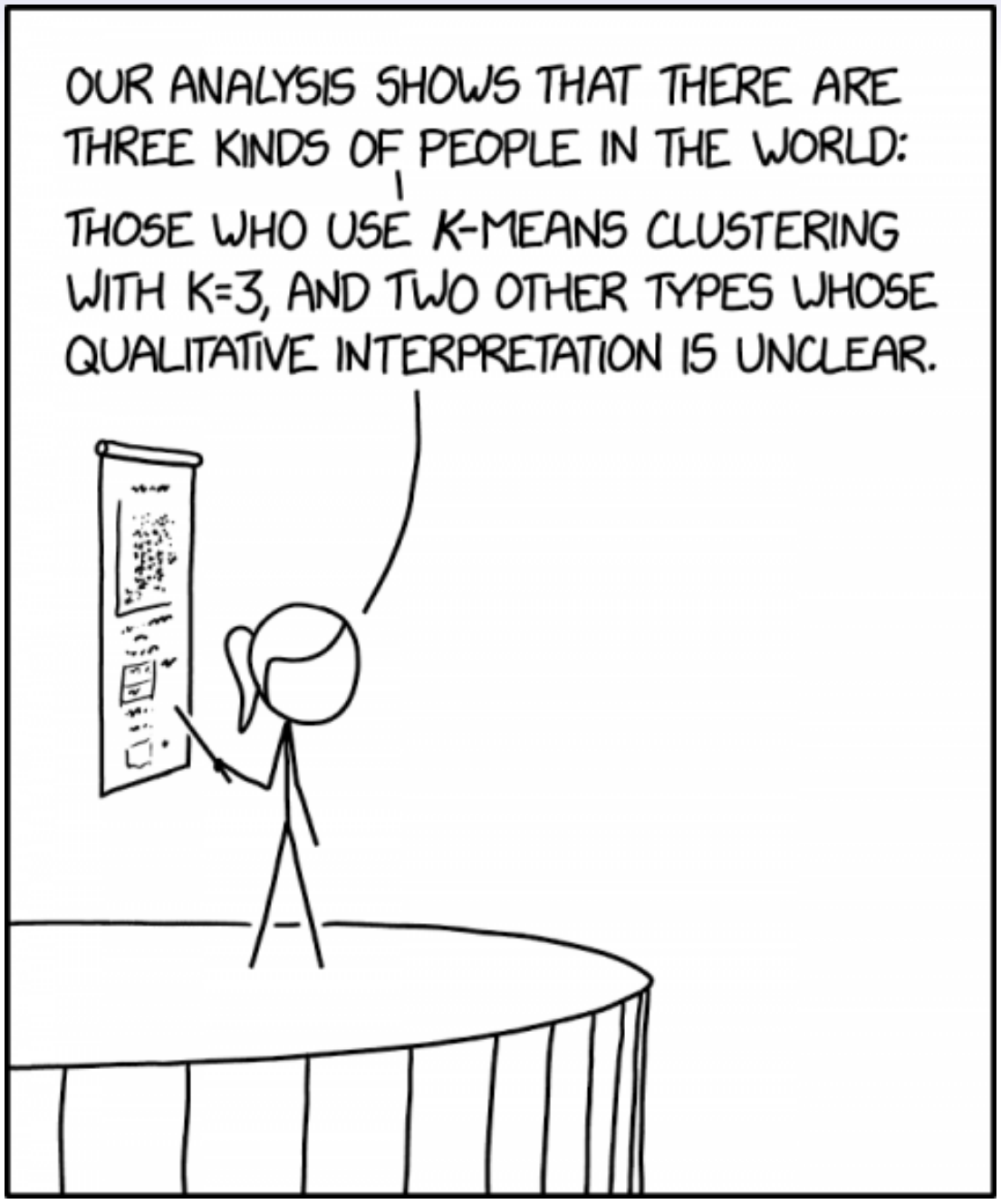
Answer: the sum is minimal if $\vec{c_i} = \vec{p}$ and $k = |C|$, i.e. all clusters are isolated points. Non-informative.



The elbow method is a simple heuristic. The 'elbow' in the diagram is the maximum k when the difference $f(k) - f(k-1)$ is large enough.

Visualisation: find the point when rate of decline in variance changes most.

Main difficulty: it's hard to evaluate if the clustering is any good.


OUR ANALYSIS SHOWS THAT THERE ARE THREE KINDS OF PEOPLE IN THE WORLD:
THOSE WHO USE K-MEANS CLUSTERING WITH K=3, AND TWO OTHER TYPES WHOSE QUALITATIVE INTERPRETATION IS UNCLEAR.

Summary of k-means clustering

k-means clustering minimises $\sum_{i=1}^{k} \sum_{p \in C_i} |\vec{c_i} - \vec{p}|^2$.

Advantages of k-means

- easy to implement
- quickly converges in good cases
- time complexity allows multiple runs

Disadvantages of k-means

- iterative, not uniform
- may not converge at all
- depends on initialisation
- clusters should be convex
- unclear how to choose k

Evaluation of clustering

There is no ground truth in clustering, so evaluation is tricky.

Extrinsic evaluation is used when clustering is a pre-processing for a higher level task (image analysis, noise elimination) e.g. if results are used in a classifier, when ground truch is available.

Intrinsic evaluation is used when clustering is the goal in itself. Clusters can be compared with problem-dependent reference classes or with human evaluation on some examples. For more detail, see pair-based evaluation.

Complexity of k-means algorithms

For a fixed dimension m and a number k of clusters, an optimal partition can be found in time $O(n^{mk+1})$, i.e. the number of operations is proportional to the fast growing function n^{mk+1} when $n \to +\infty$.

Lloyd's algorithm has time $O(nkmI)$, where I is the number of iterations. In the worst-case $I = 2^{\Omega(n)}$, where $\Omega(n)$ denotes a function that grows proportional to n or faster. In practice I is often small on data split into well-separated groups.

Summary and final question

Here are the steps of Lloyd's heuristic algorithm.

- Initialise k centres of clusters in a cloud C.
- For each centre, assign all points of C that are closer to this centre than to all others.
- Re-compute the centre of every cluster that was updated above and re-assign all points.
- Stop when centres of clusters don't change or a maximum number of iterations is reached.

Problem 18.5. Find optimal 2-means clusters for the cloud $C = \{(0,0),(x,0),(0,y),(x,y)\} \subset \mathbb{R}^2$.