K-MEANS CLUSTERING

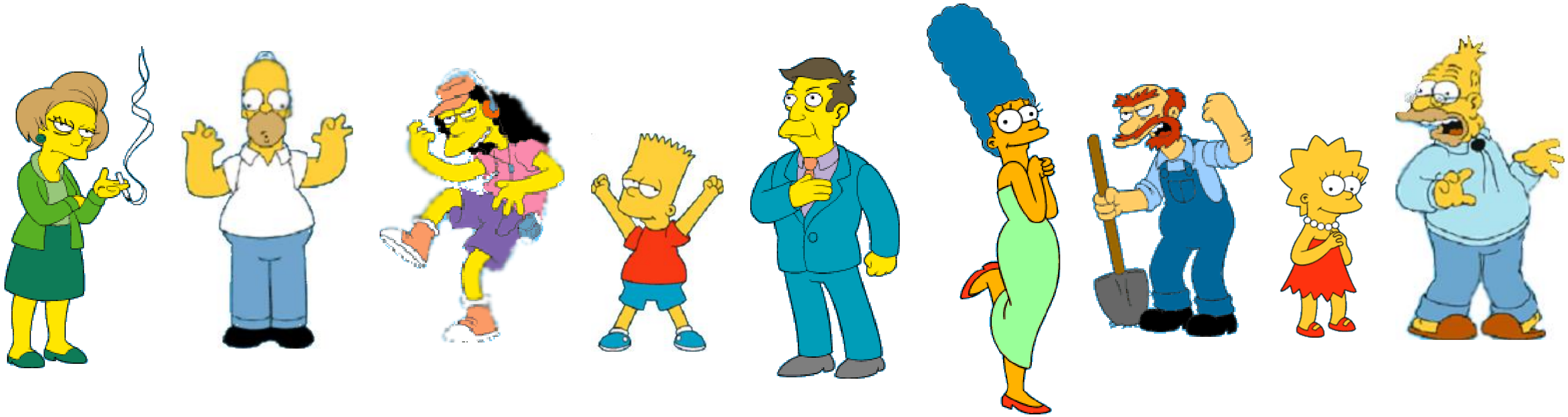
- Abhishek Kumar
- Jasdeep Prataap Singh
- Jayashree Biswal
- Om Patra
- Raunaq Kumar

What is Clustering?

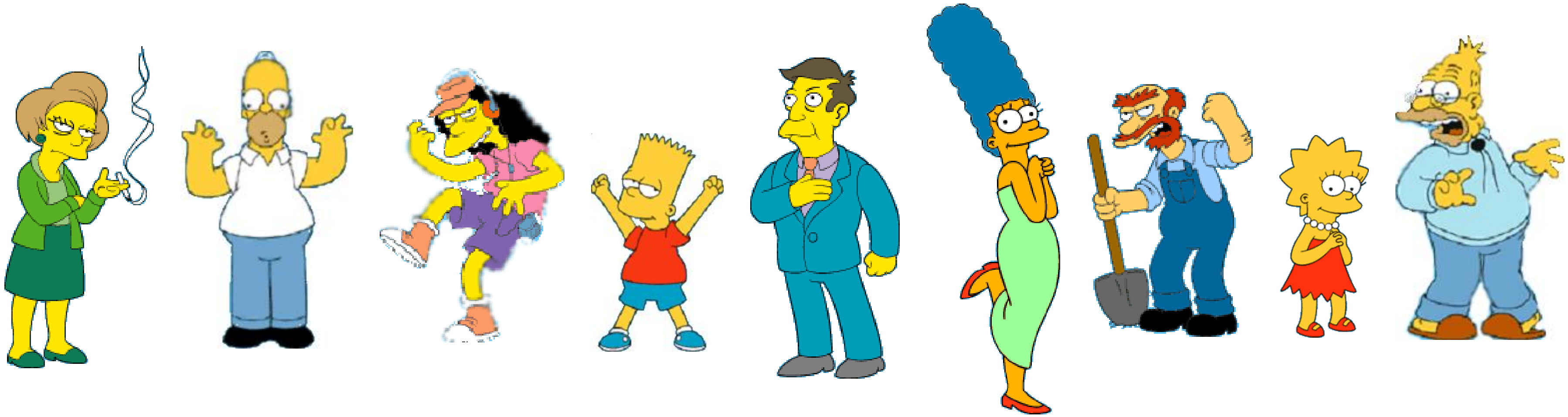
Also called *unsupervised learning*, sometimes called *classification* by statisticians and *sorting* by psychologists and *segmentation* by people in marketing

- Organizing data into classes such that there is
    - high intra-class similarity
    - low inter-class similarity
- Finding the class labels and the number of classes directly from the data (in contrast to classification).
- More informally, finding natural groupings among objects.

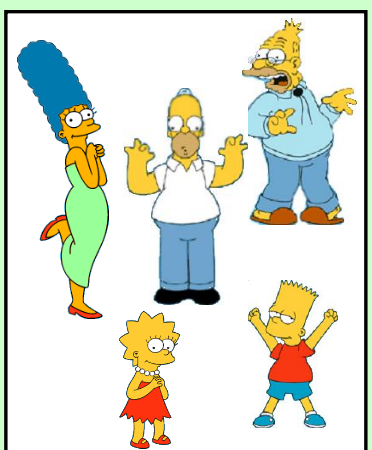
What is a natural grouping among these objects?



What is a natural grouping among these objects?

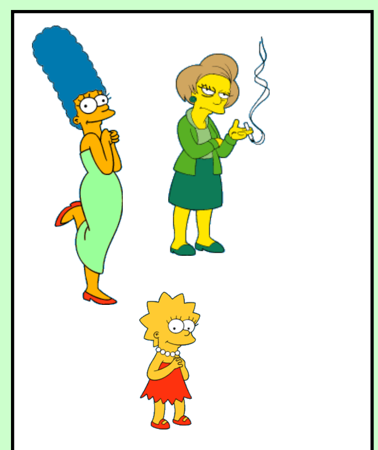


Clustering is subjective

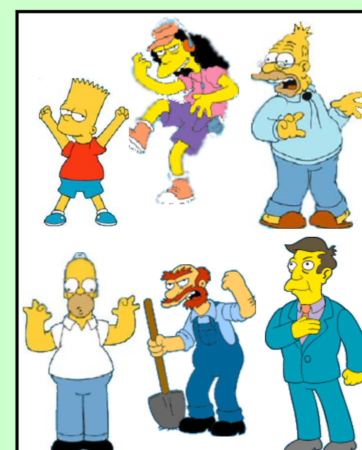


Simpson's Family

School Employees



Females



Males

What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features.

Webster's Dictionary

Similarity is hard to define, but… "*We know it when we see it*"

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- Urban planning: Identifying groups of houses according to their house type, value, and geographical location
- Seismology: Observed earth quake epicenters should be clustered along continent faults

What Is a Good Clustering?

- A good clustering method will produce clusters with
  - High <u>intra-class</u> similarity
  - Low <u>inter-class</u> similarity
- Precise definition of clustering quality is difficult
  - Application-dependent
  - Ultimately subjective

Requirements for Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal domain knowledge required to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- Robustness wrt high dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Major Clustering Approaches

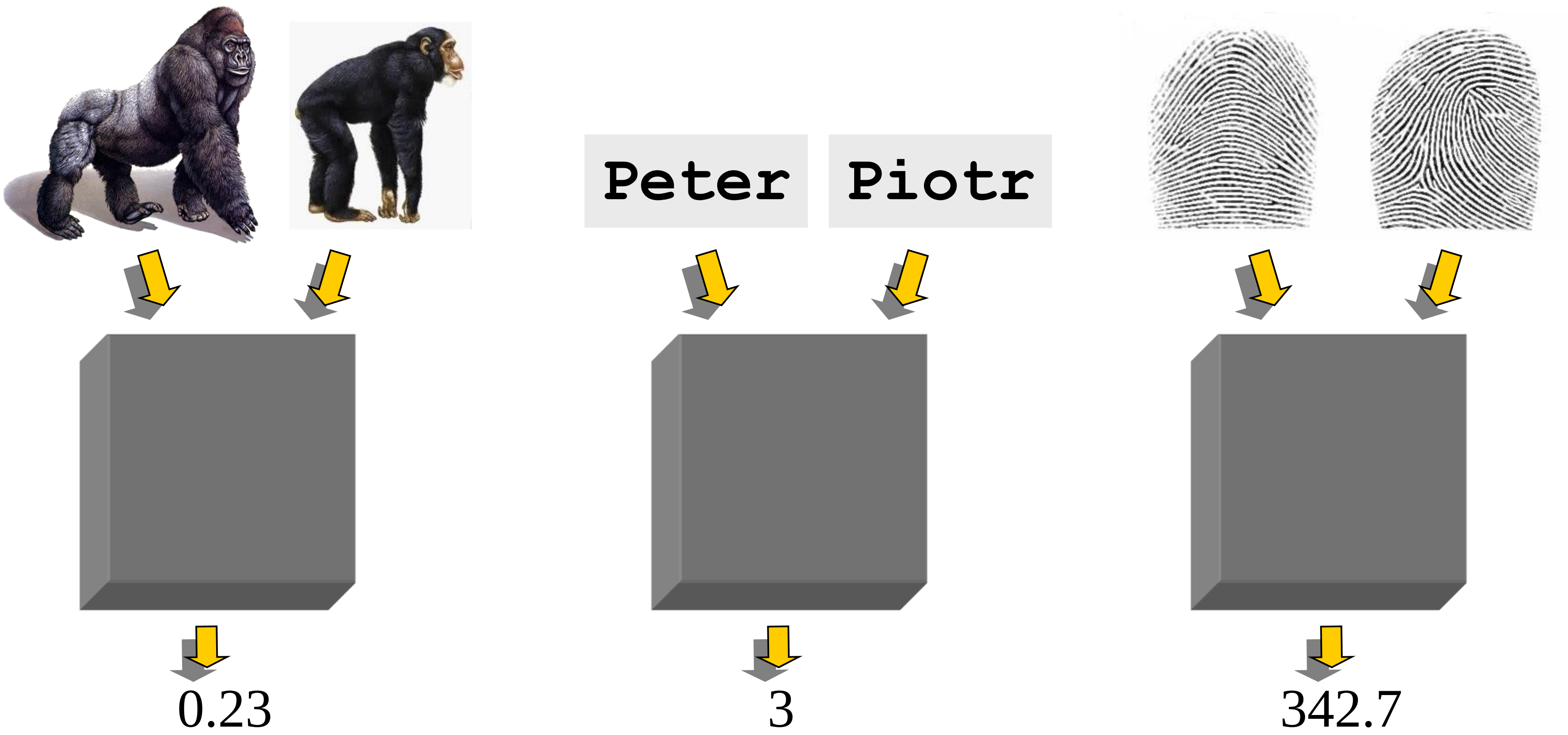
- Partitioning: Construct various partitions and then evaluate them by some criterion
- Hierarchical: Create a hierarchical decomposition of the set of objects using some criterion
- Model-based: Hypothesize a model for each cluster and find best fit of models to data
- Density-based: Guided by connectivity and density functions

Partitioning Algorithms

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen, 1967): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw, 1987): Each cluster is represented by one of the objects in the cluster

Defining Distance Measures

Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1,O_2)$



Similarity and Dissimilarity Between Objects

- Same we used for IBL (e.g, Lp norm)
- Euclidean distance (p = 2):

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

What properties should a distance measure have?

- *D*(A,B) = *D*(B,A) *Symmetry*
- *D*(A,A) = 0 *Constancy of Self-Similarity*
- *D*(A,B) = 0 IIf A= B *Positivity (Separation)*
- *D*(A,B) ≤ *D*(A,C) + *D*(B,C) *Triangular Inequality*

Intuitions behind desirable distance measure properties

D(A,B) = D(B,A) *Symmetry*

Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex."

D(A,A) = 0 *Constancy of Self-Similarity*

Otherwise you could claim "Alex looks more like Bob, than Bob does."

D(A,B) = 0 IIf A=B *Positivity (Separation)*

Otherwise there are objects in your world that are different, but you cannot tell apart.

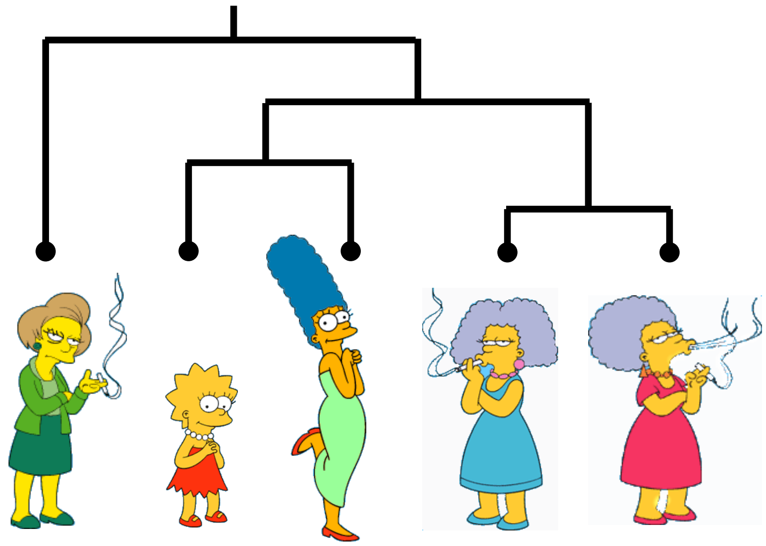
D(A,B) ≤ D(A,C) + D(B,C) *Triangular Inequality*

Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl."

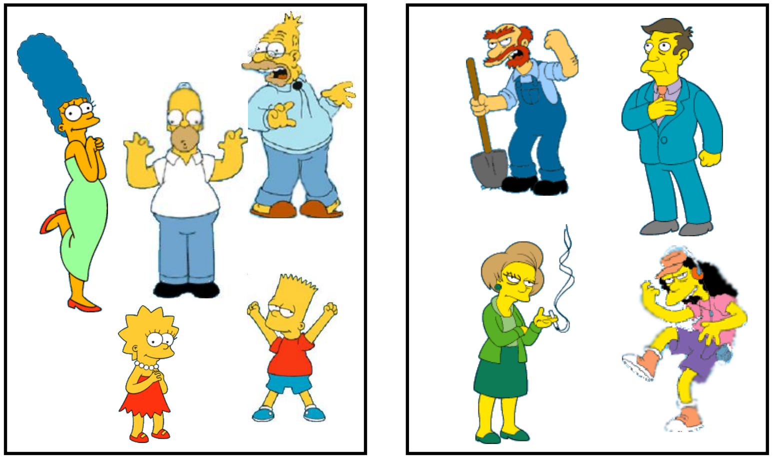
Two Types of Clustering

- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion (we will see an example called BIRCH)
- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

Hierarchical



Partitional

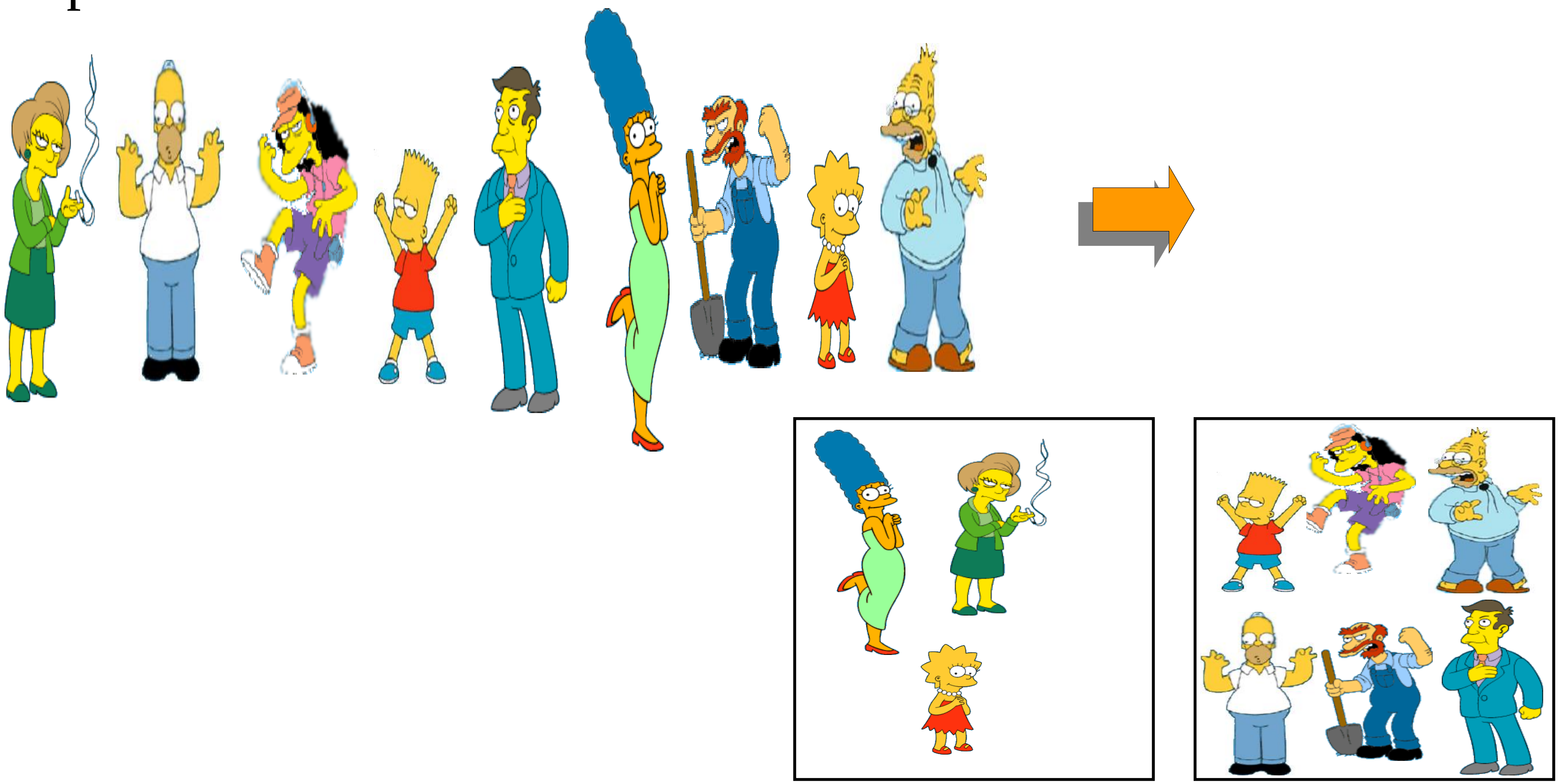


Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Incorporation of user-specified constraints
- Interpretability and usability

Partitional Clustering

- Nonhierarchical, each instance is placed in exactly one of K nonoverlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K.

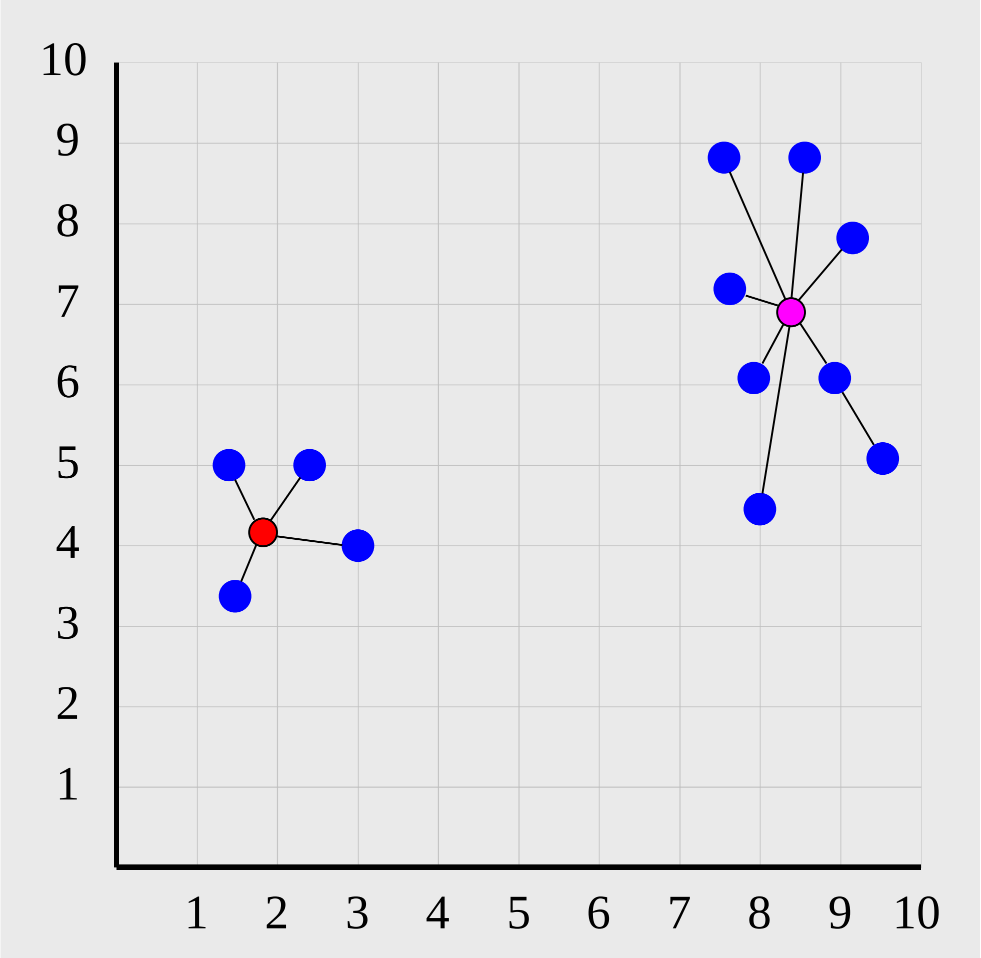


Squared Error

$$se_{K_i} = \sum_{j=1}^{m} \|t_{ij} - C_k\|^2$$

$$se_K = \sum_{j=1}^{k} se_{K_j}$$

Objective Function



Algorithm *k-means*

1. Decide on a value for *k*.

2. Initialize the *k* cluster centers (randomly, if necessary).

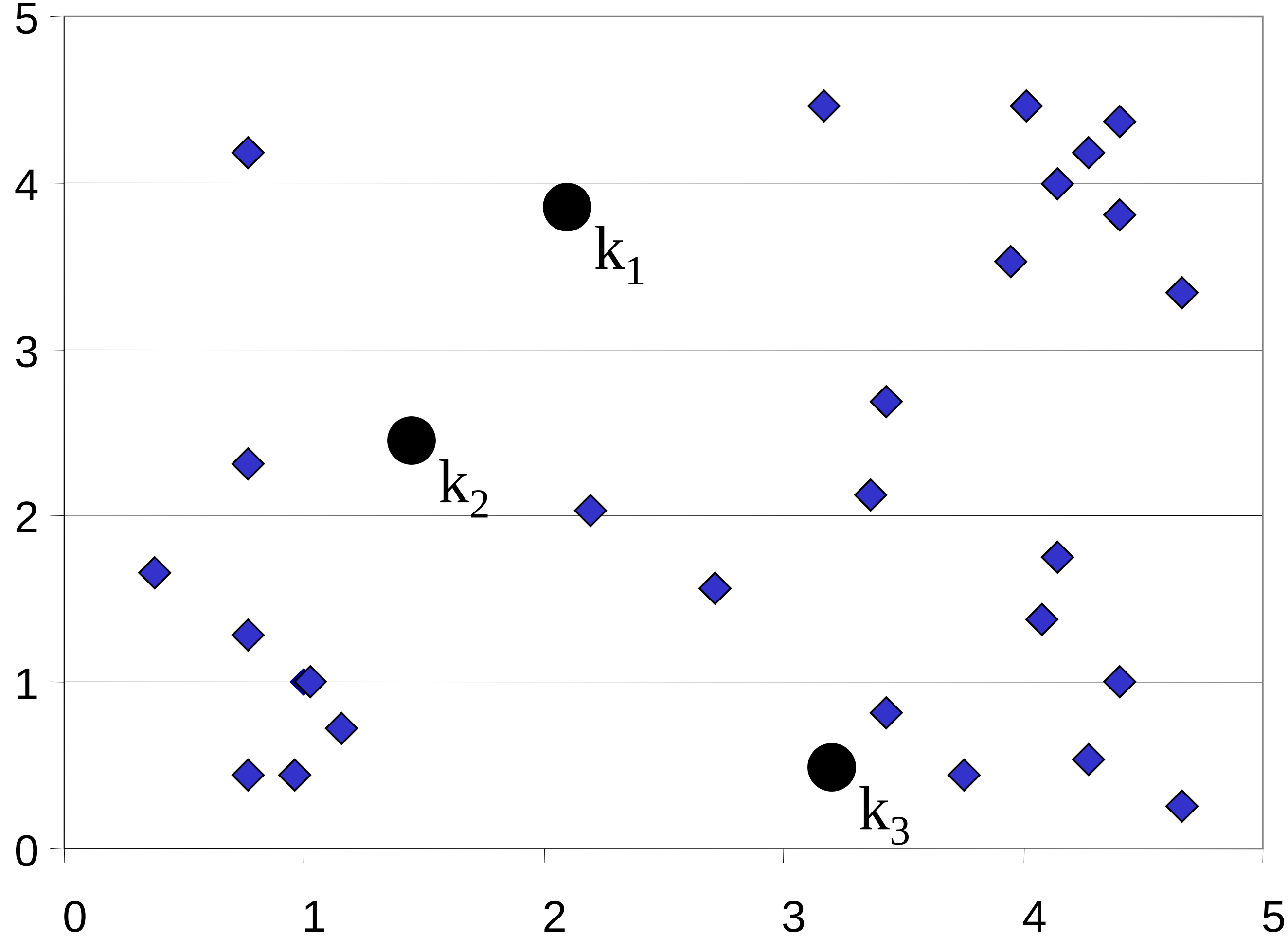
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise goto 3.

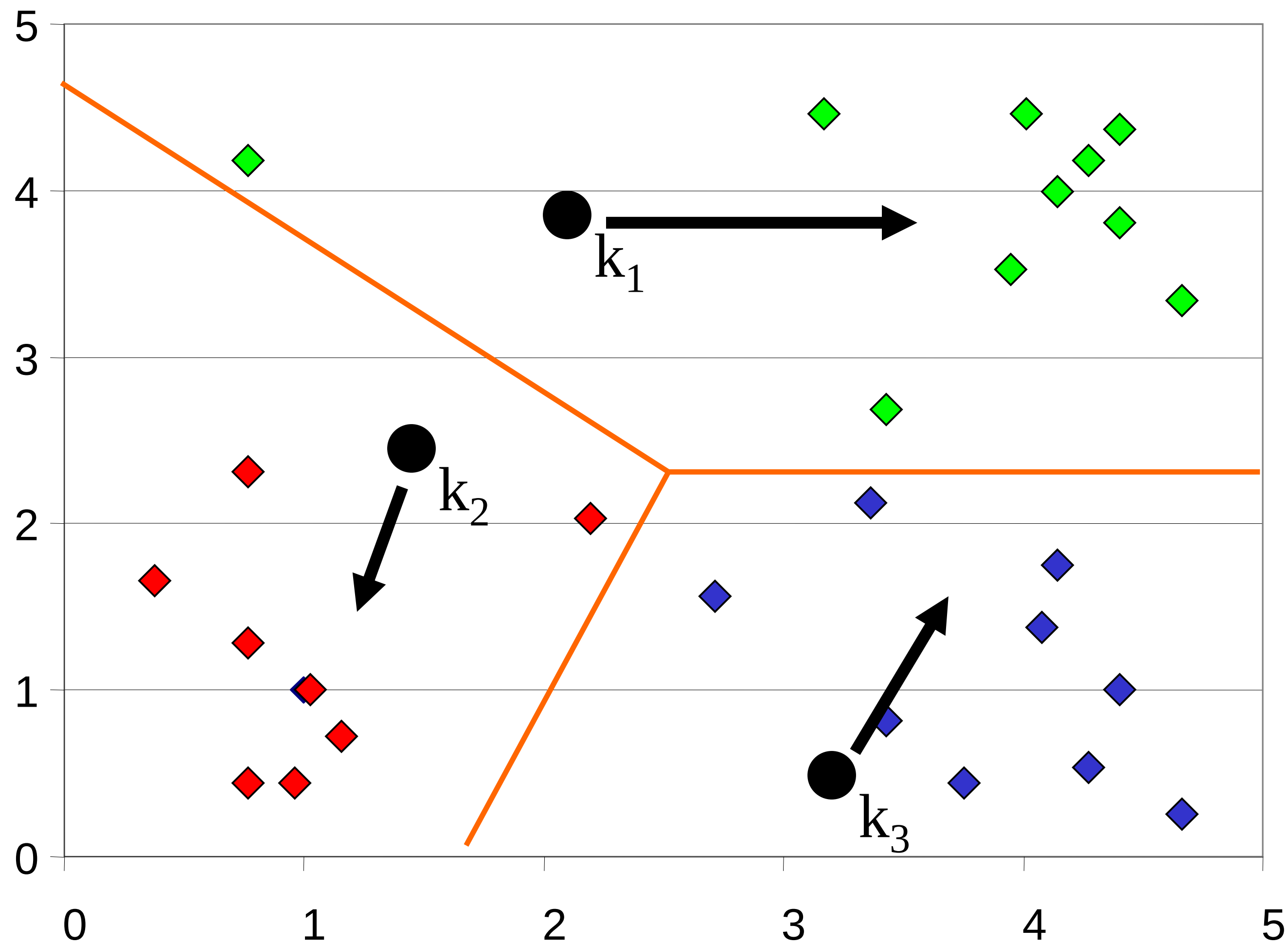
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



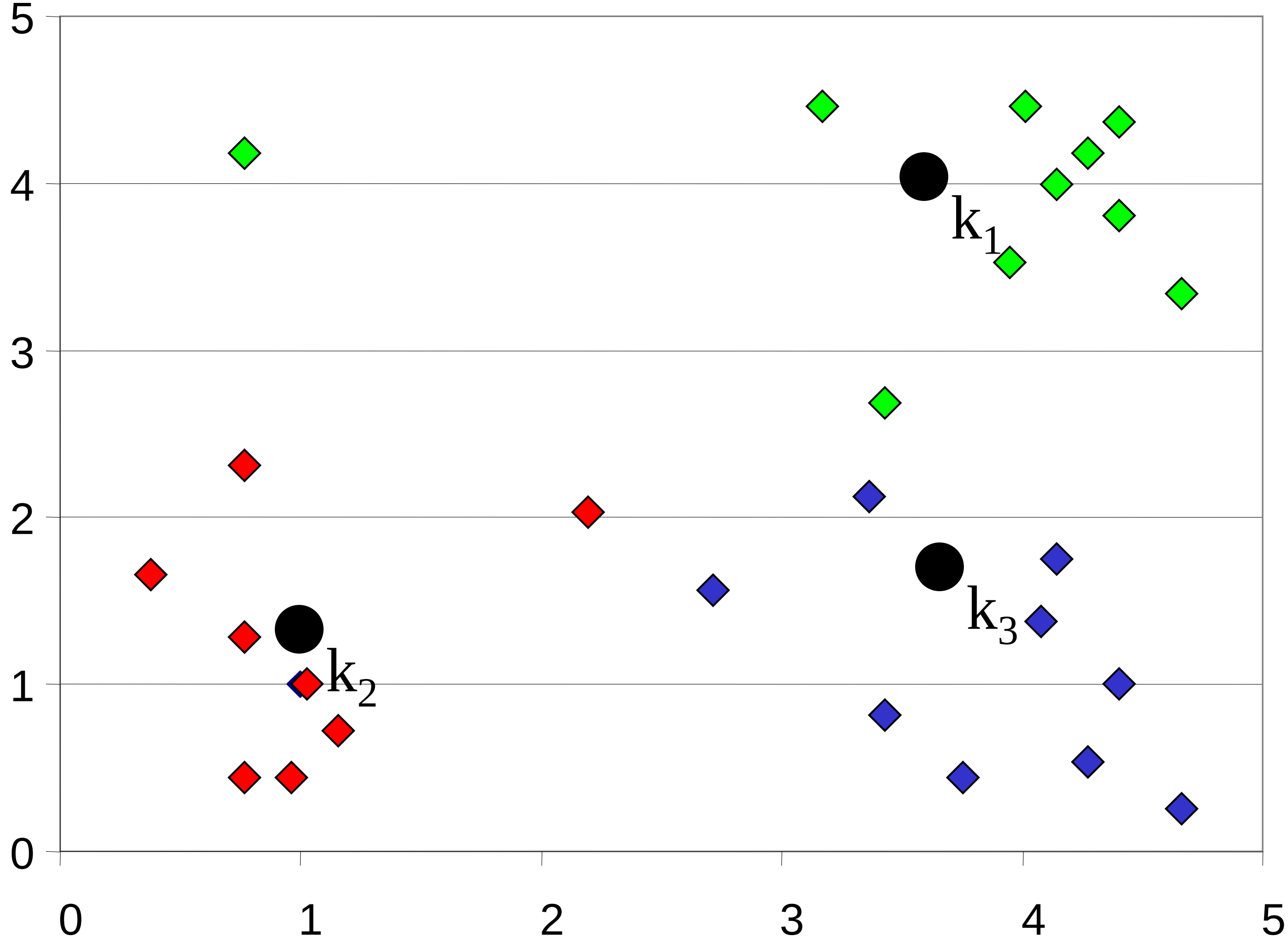
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



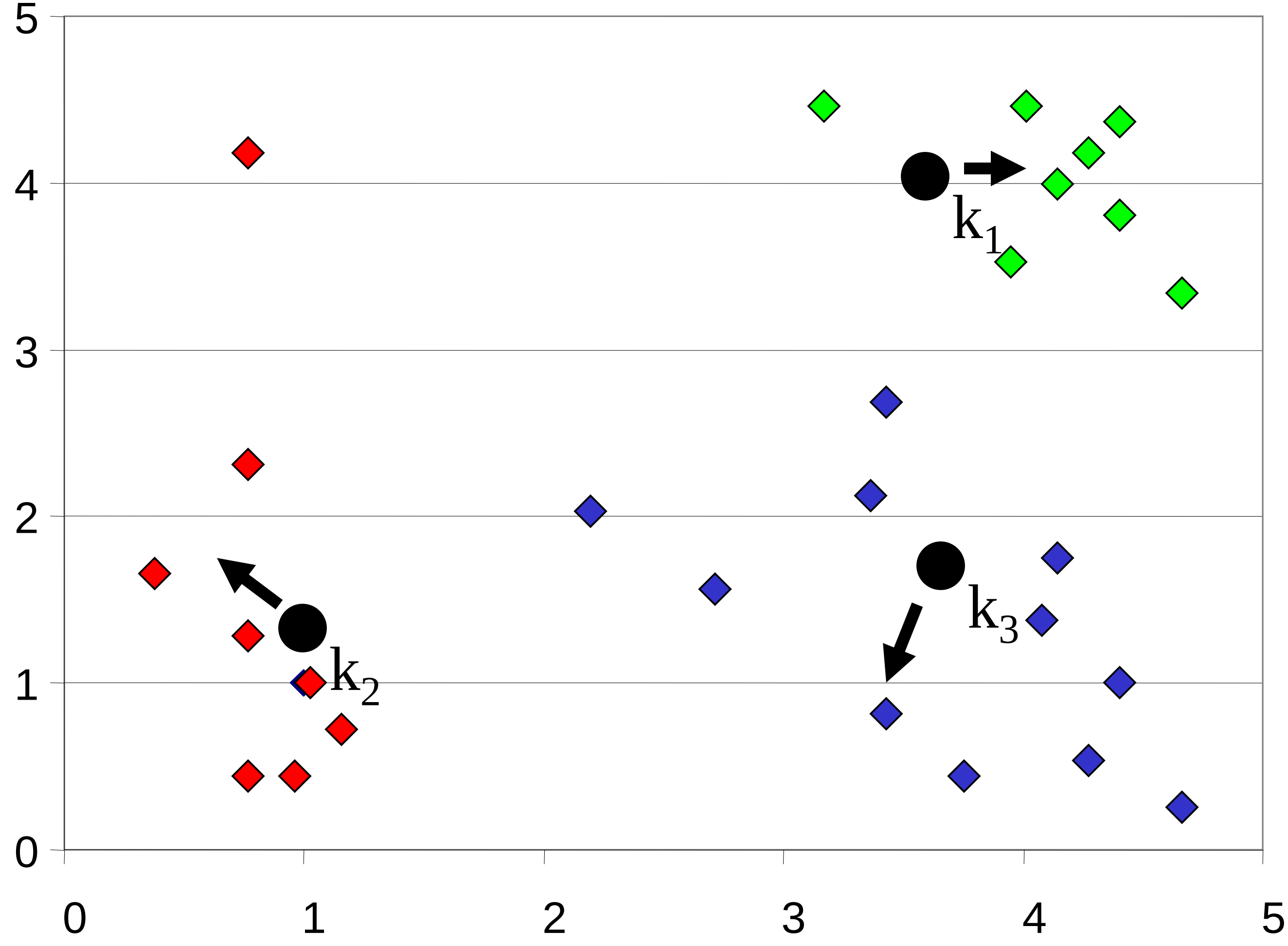
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



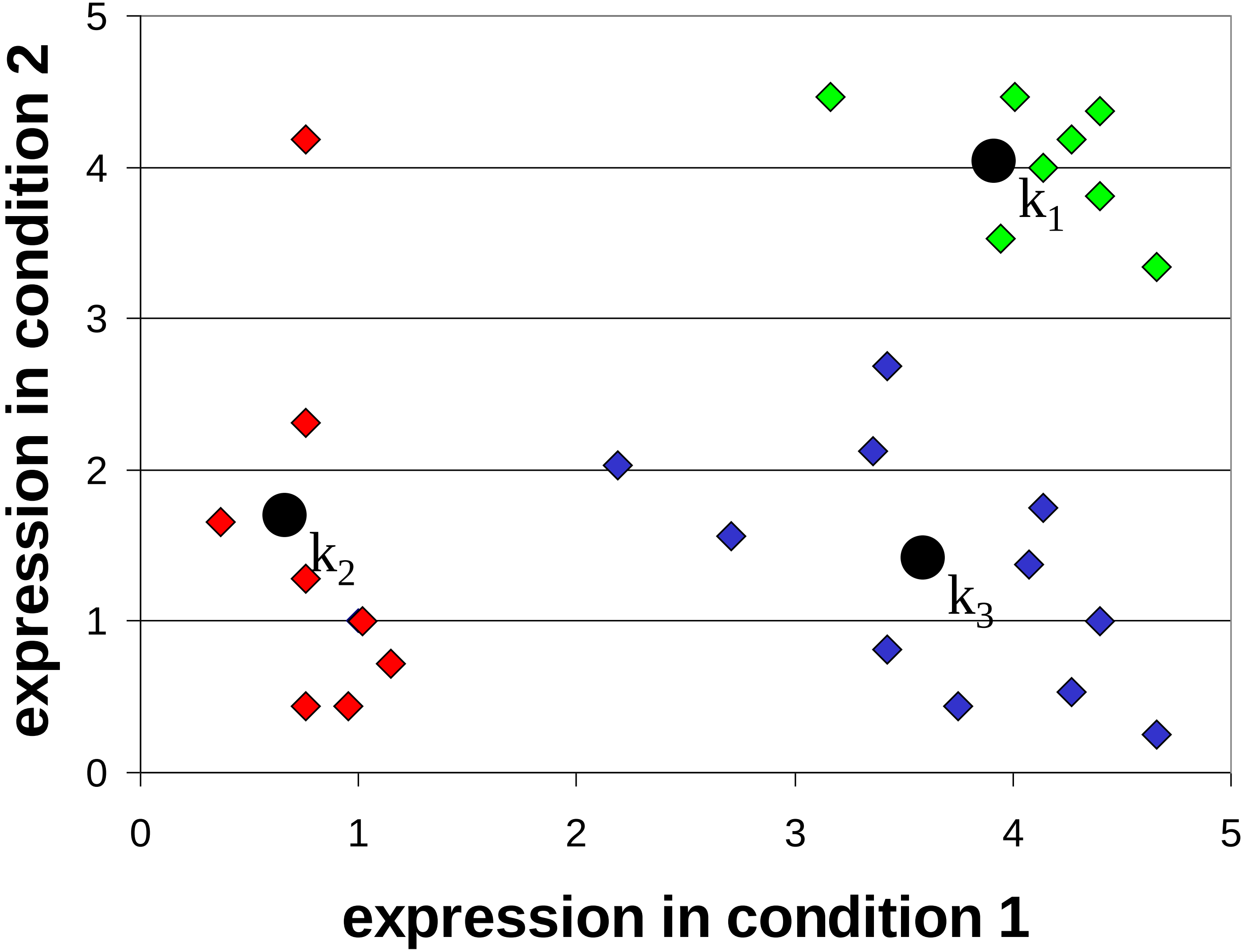
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



Comments on the *K-Means* Method

- Strength
  - *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify k, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

The *K-Medoids* Clustering Method

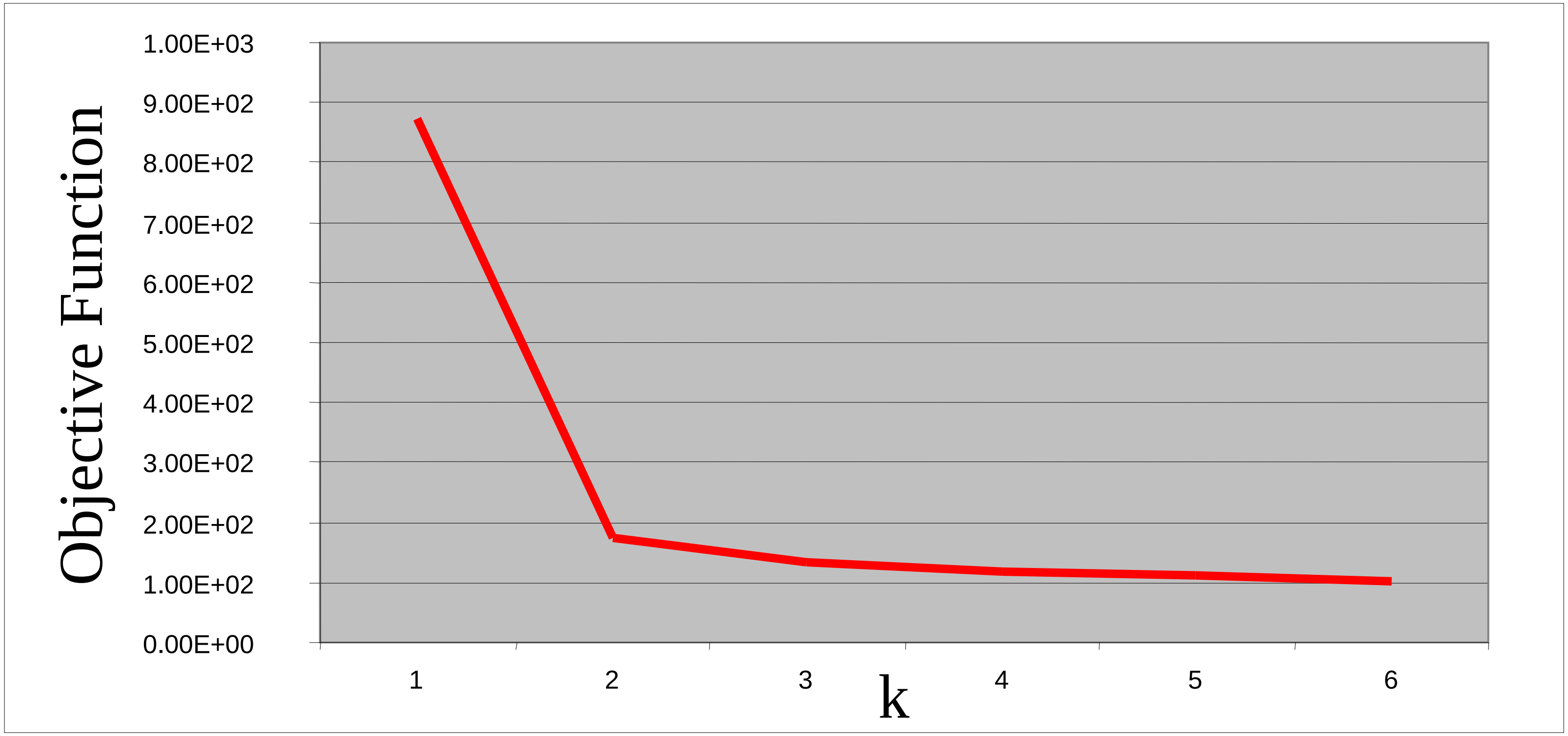
- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets

How Many Clusters?

- Number of clusters K is given
  - Partition n docs into predetermined number of clusters
- Finding the “right” number of clusters is part of the problem
  - Given docs, partition into an “appropriate” number of subsets.
  - E.g., for query results - ideal value of K not known up front - though UI may impose limits.
- Can usually take an algorithm for one flavor and convert to the other.

We can plot the objective function values for k equals 1 to 6…

The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.



Note that the results are not always as clear cut as in this toy example