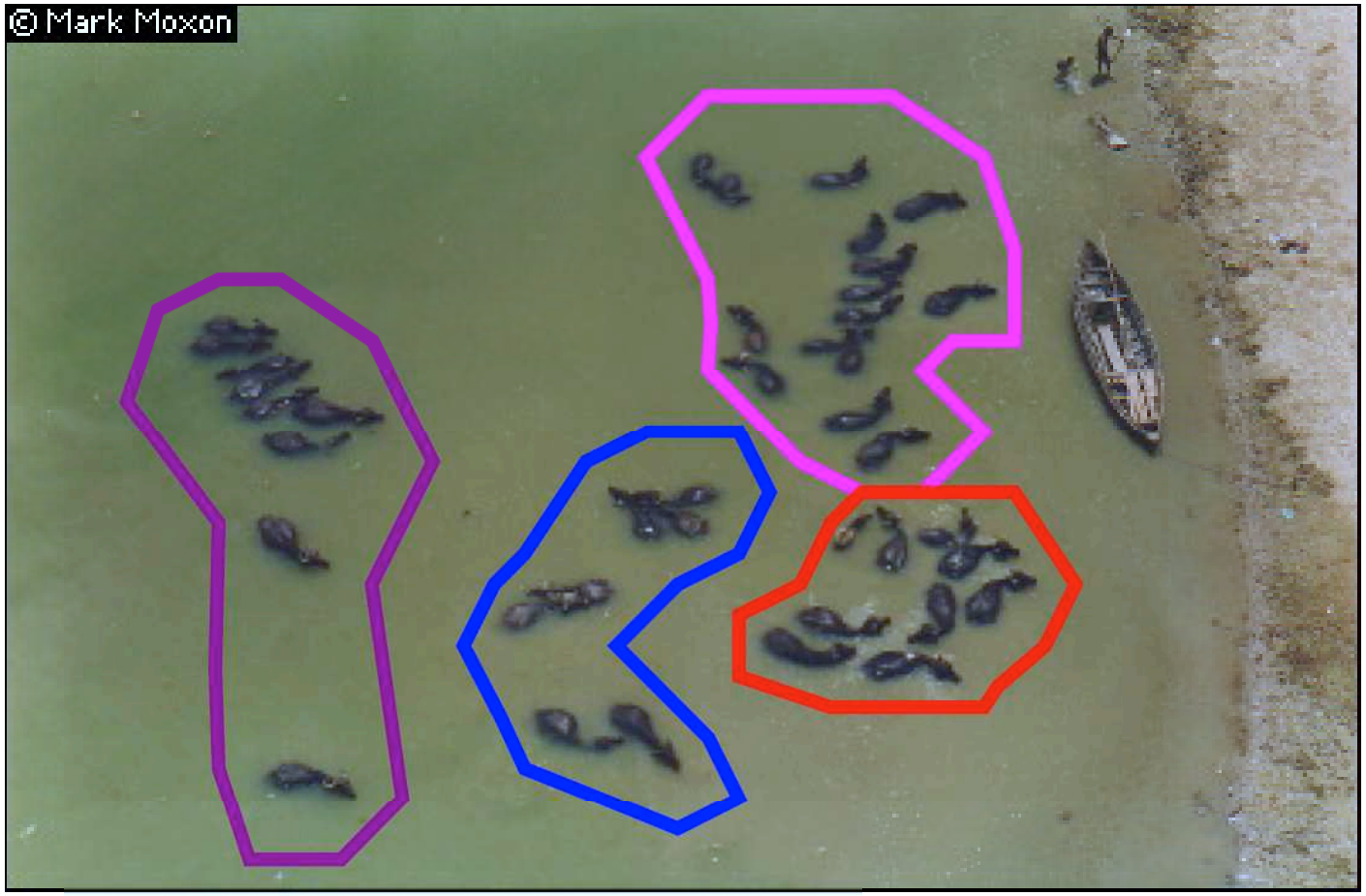
Announcements

- Lectures slides will be on the course web page: http://www.mcb.mcgill.ca/∼perkins/COMP76602/COMP76602.html
- By later today, I will also put up links to background readings on biology/bioinfo and produce a detailed schedule for the next few weeks at least.
- Don’t forget homework 0! Email me from your preferred and tell me briefly what your cs/math/bio/bioinfo background is.
- Today we begin Section 1 – Summarization – with a discussion of clustering.

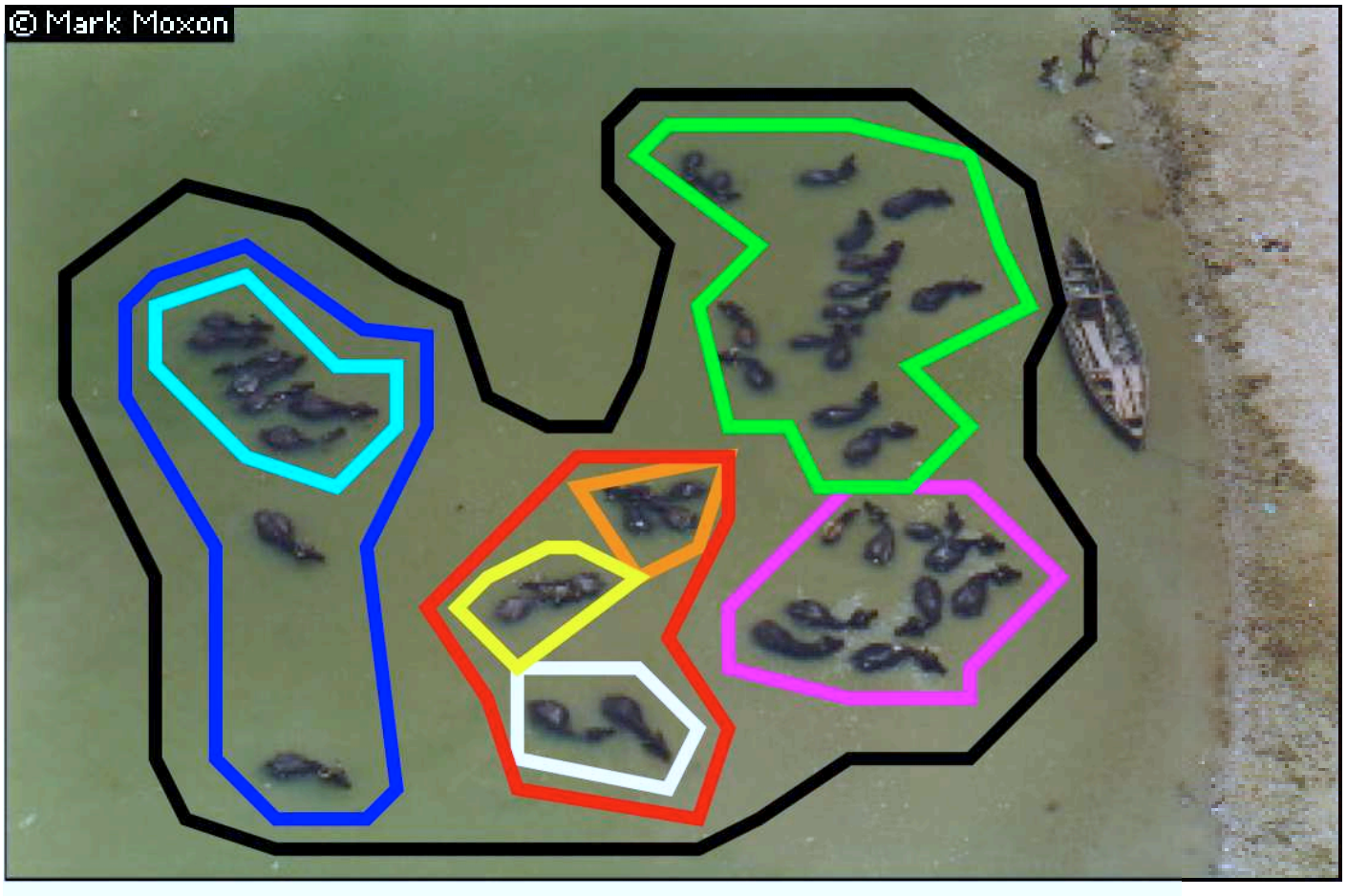
What is clustering?

- Clustering is grouping similar objects together.
  - To help visualize data.
  - To establish prototypes, or detect outliers.
  - To simplify data for further analysis/learning.
- Clusterings are not right or wrong – different clusterings can reveal different things about the data.
- There are two major types of clustering,“flat” and hierarchical.

Flat clustering divides, or partitions, the set of objects into disjoint sets.



Hierarchical clustering organizes the objects into a tree.



Today: Flat Clustering

- K-means clustering
- A more general formulation and approaches

K-means clustering

- ... is one of the most commonly-used clustering algorithms.
- It is easy to implement and quick to run.
- Assumes the objects to be clustered are n-dimensional real vectors. (E.g., a list of expression values for different genes under some conditions; or, for the same gene under different conditions)
- Similarity between the vectors is measured by Euclidean distance.

K-means clustering

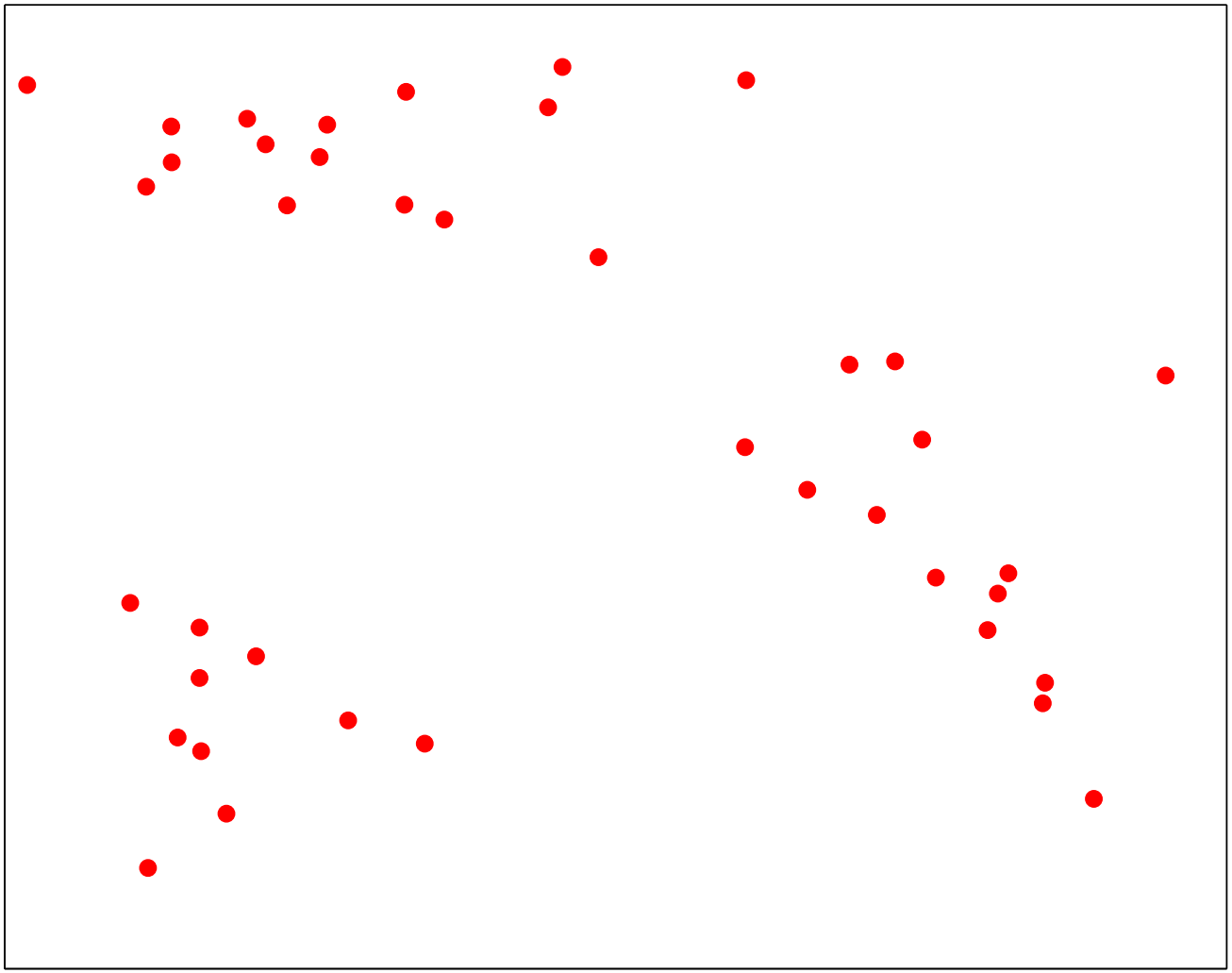
- Inputs:
  - A set of n-dimensional real vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$.
  - K, the desired number of clusters.
- Outputs:
  - A partitioning of the vectors into K clusters (disjoint subsets), $C : \{1, \ldots, m\} \mapsto \{1, \ldots, K\}$.

K-means clustering: the algorithm

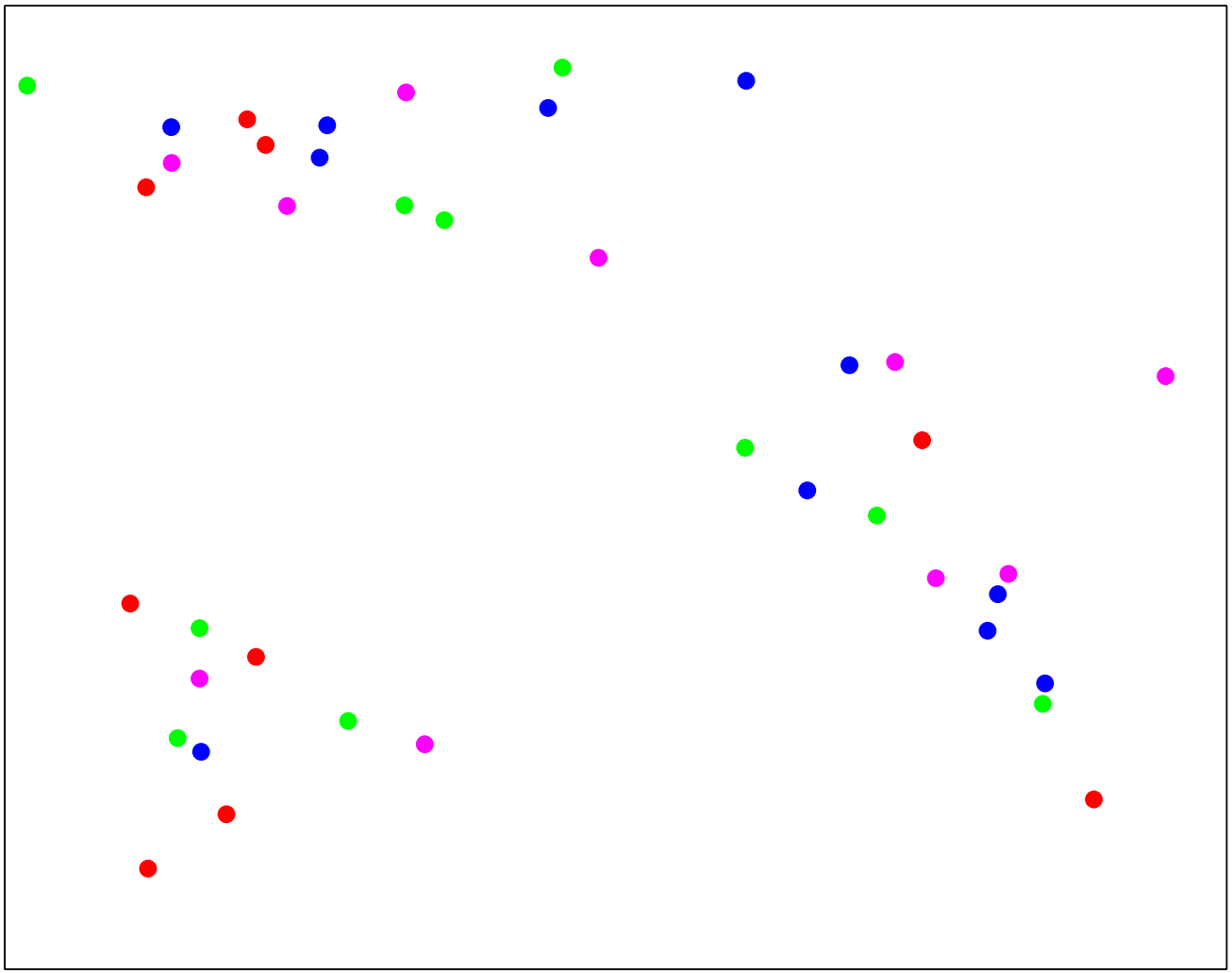
- Initialize C randomly.
- Repeat
  - Compute the *centroid* of each cluster.
    (The centroid is just the arithmetic average of the vectors in the cluster.)
  - Assign each vector to the cluster whose centroid is closest, in terms of Euclidean distance.

  Until C stops changing.

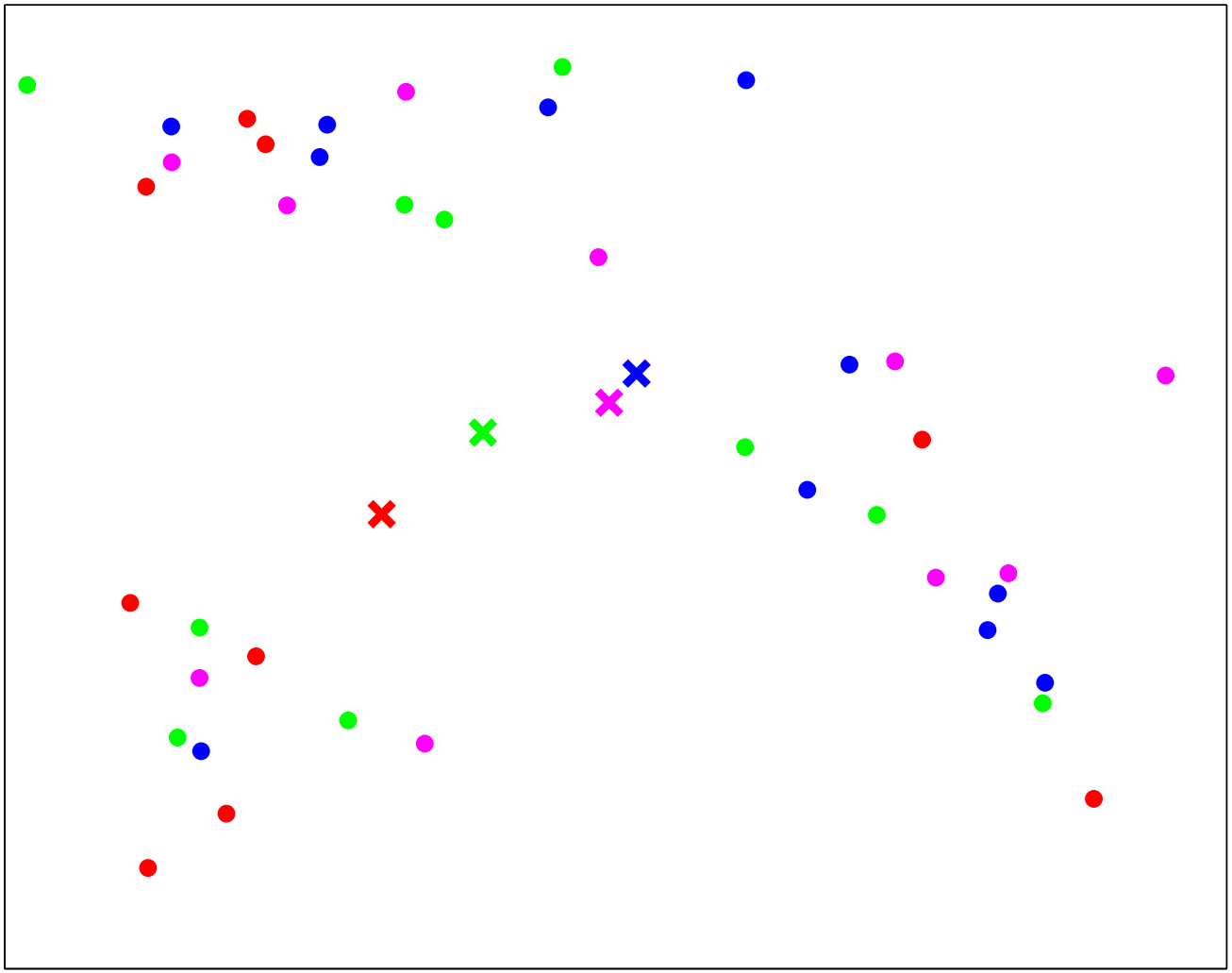
Example: initial data



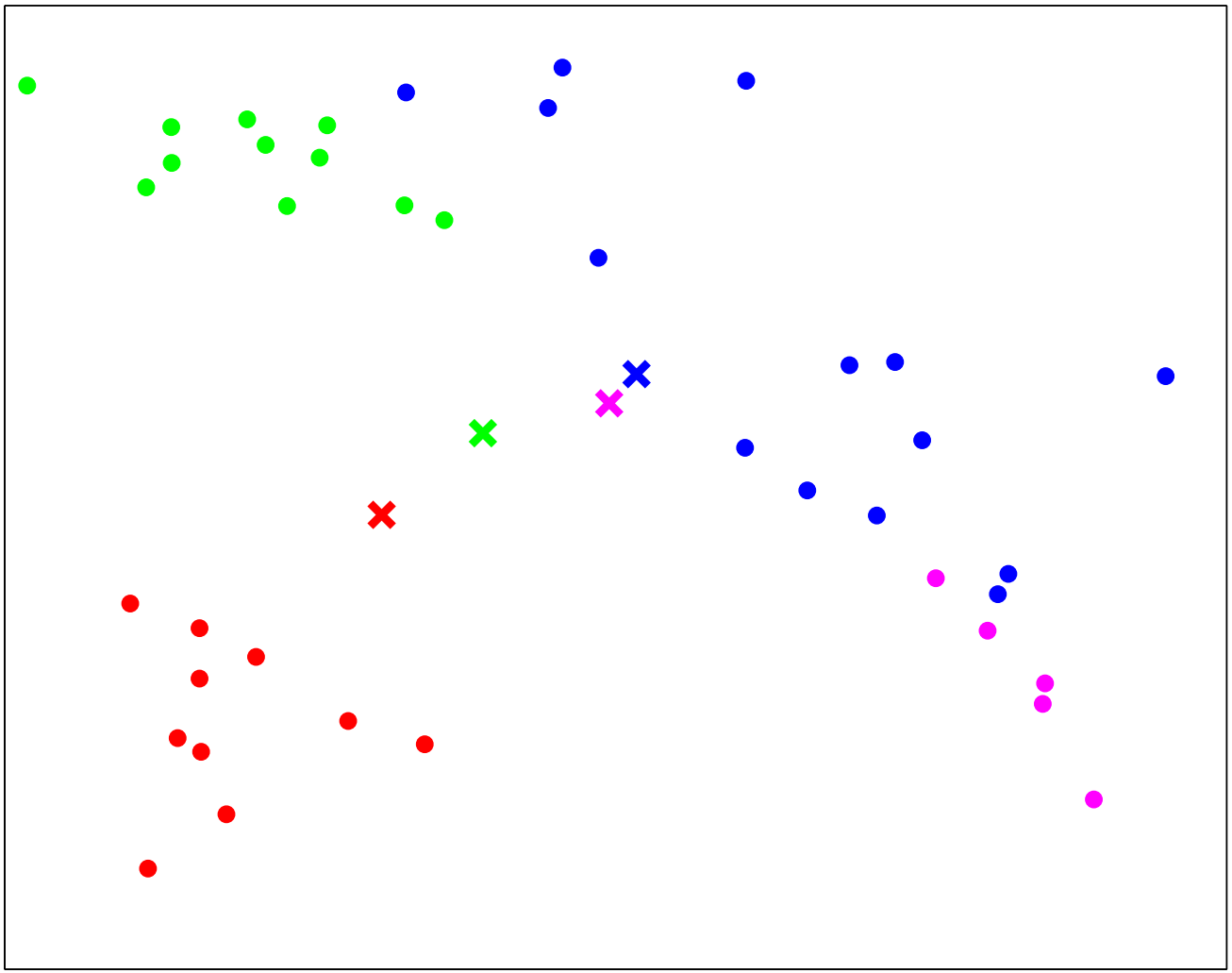
Example: assign into 4 clusters randomly



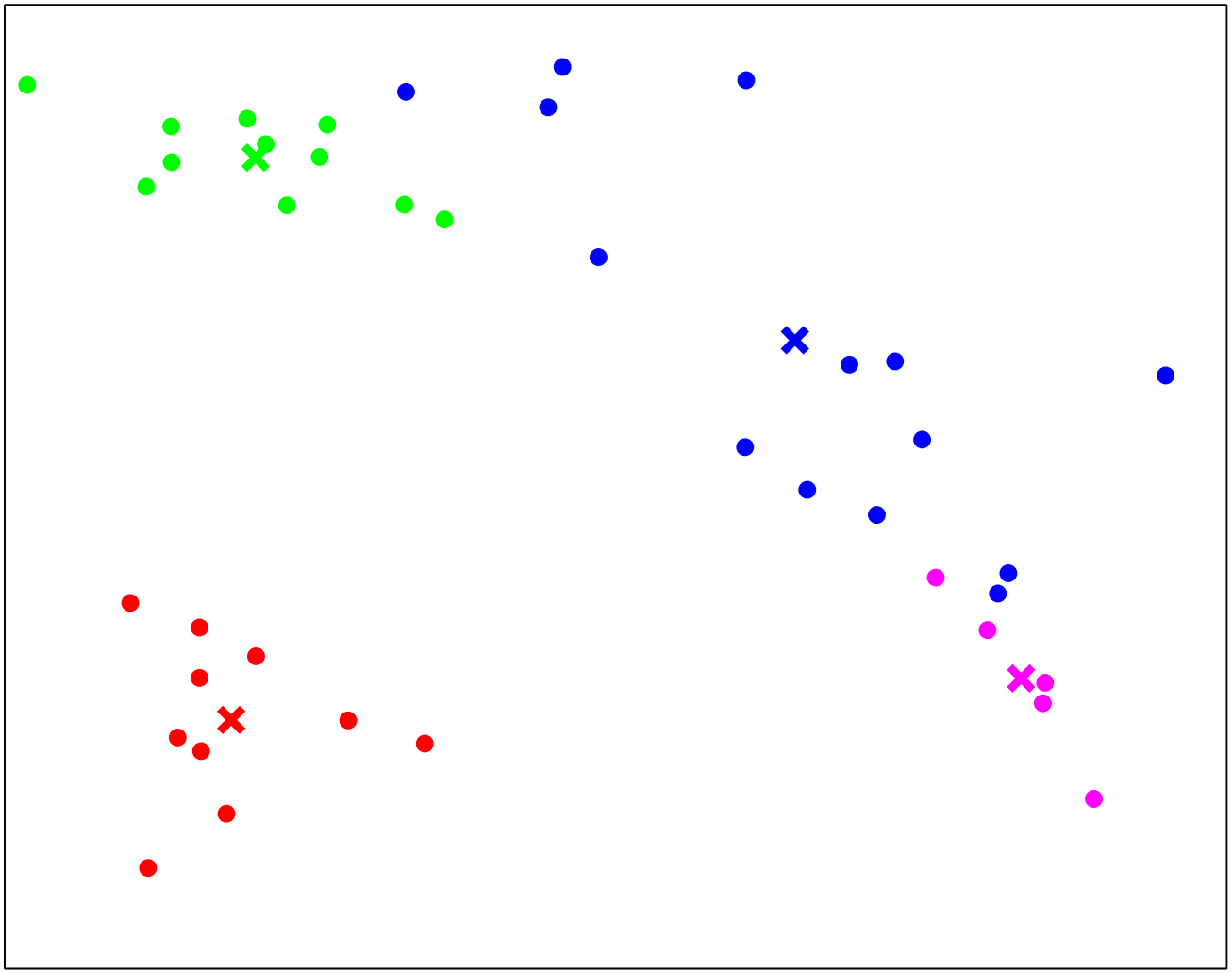
Example: compute centroids



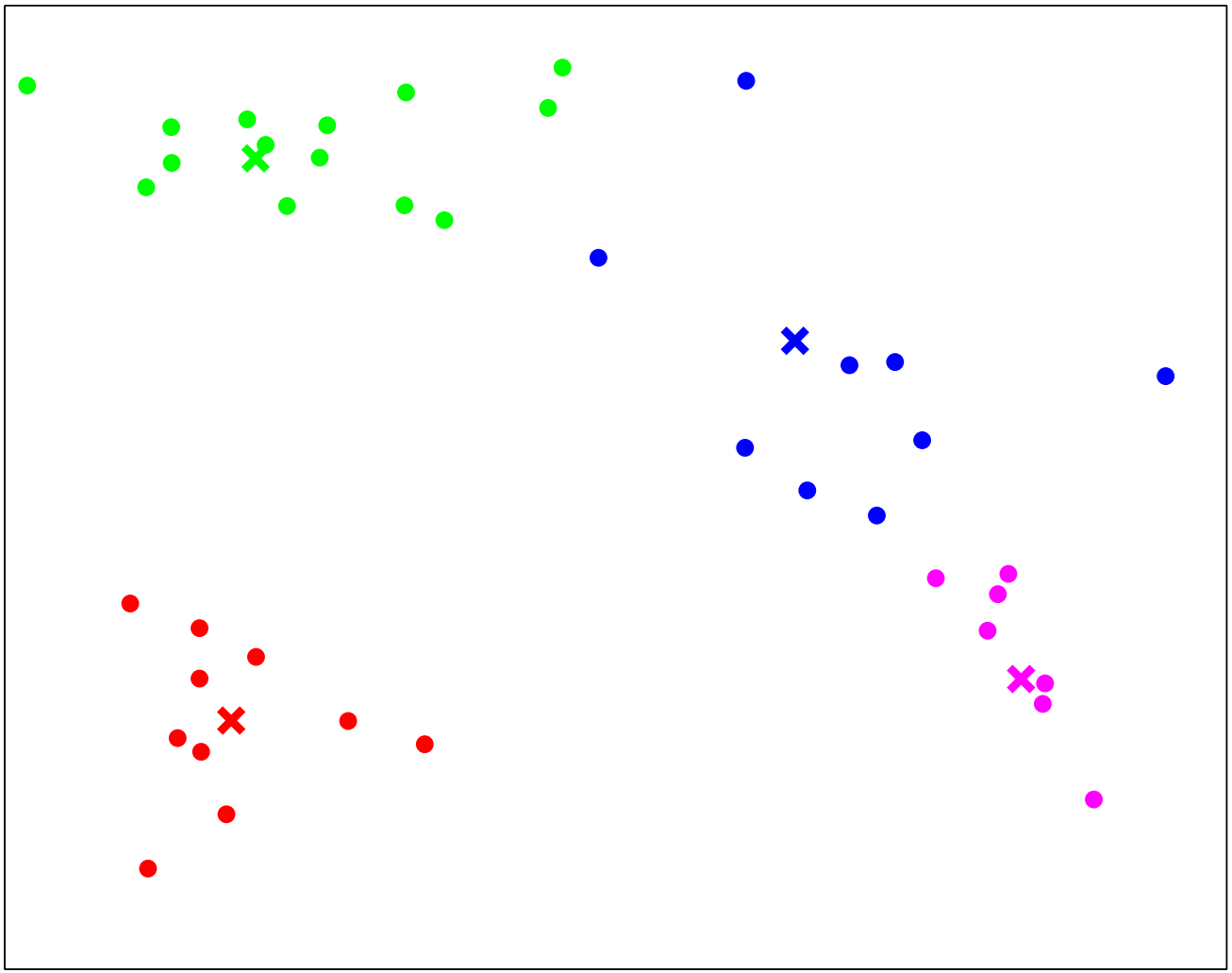
Example: reassign clusters



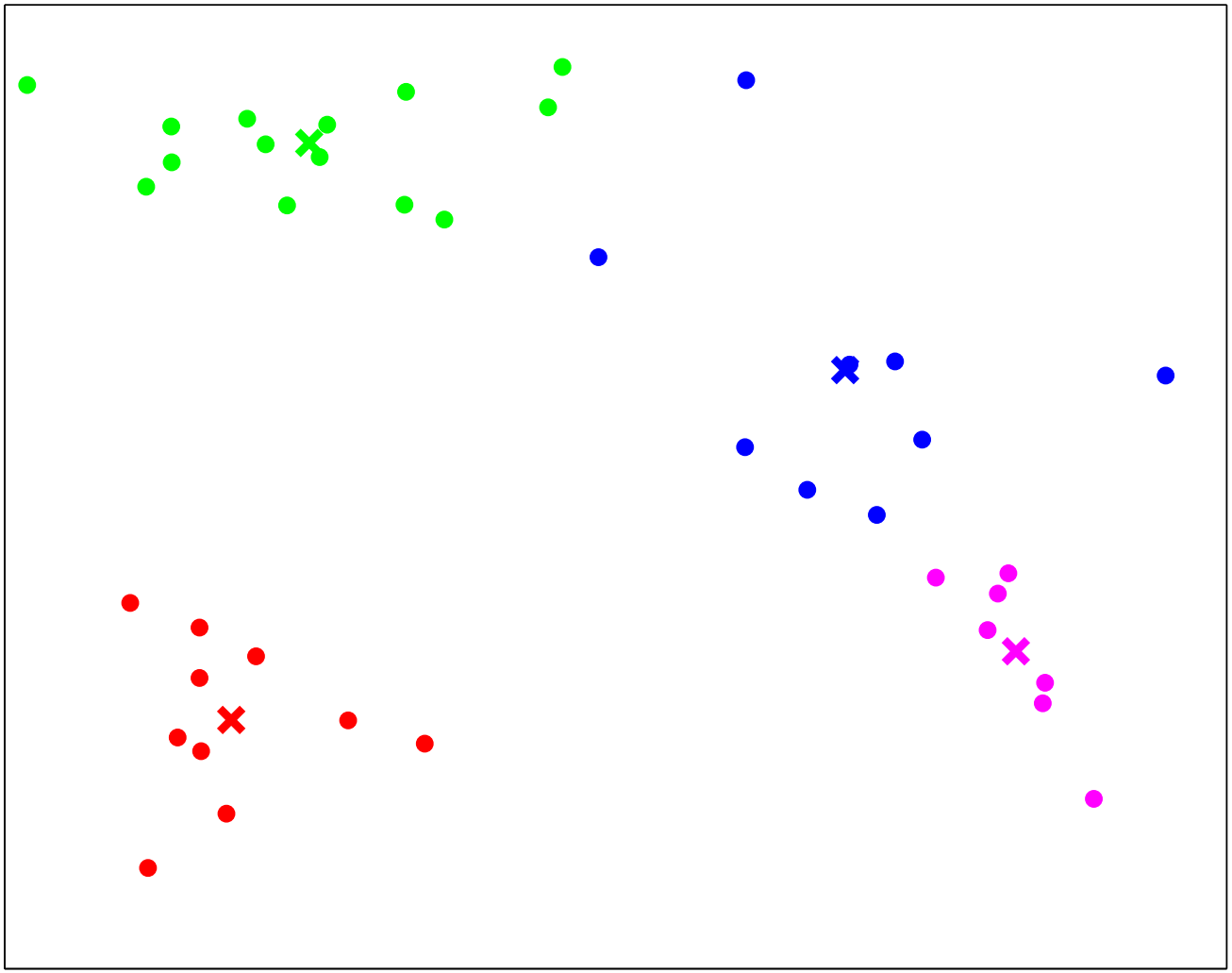
Example: recompute centroids



Example: reassign clusters

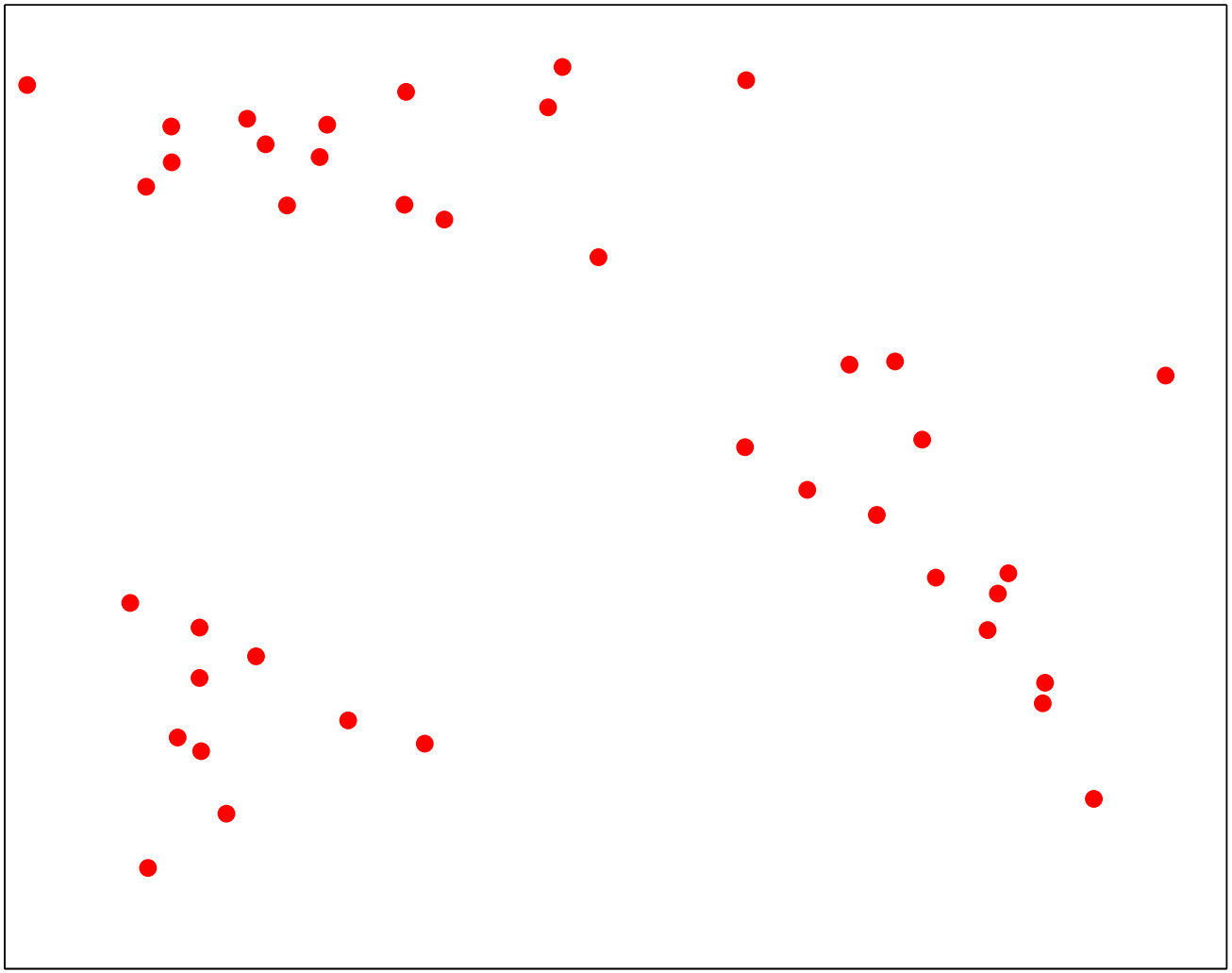


Example: recompute centroids – done!

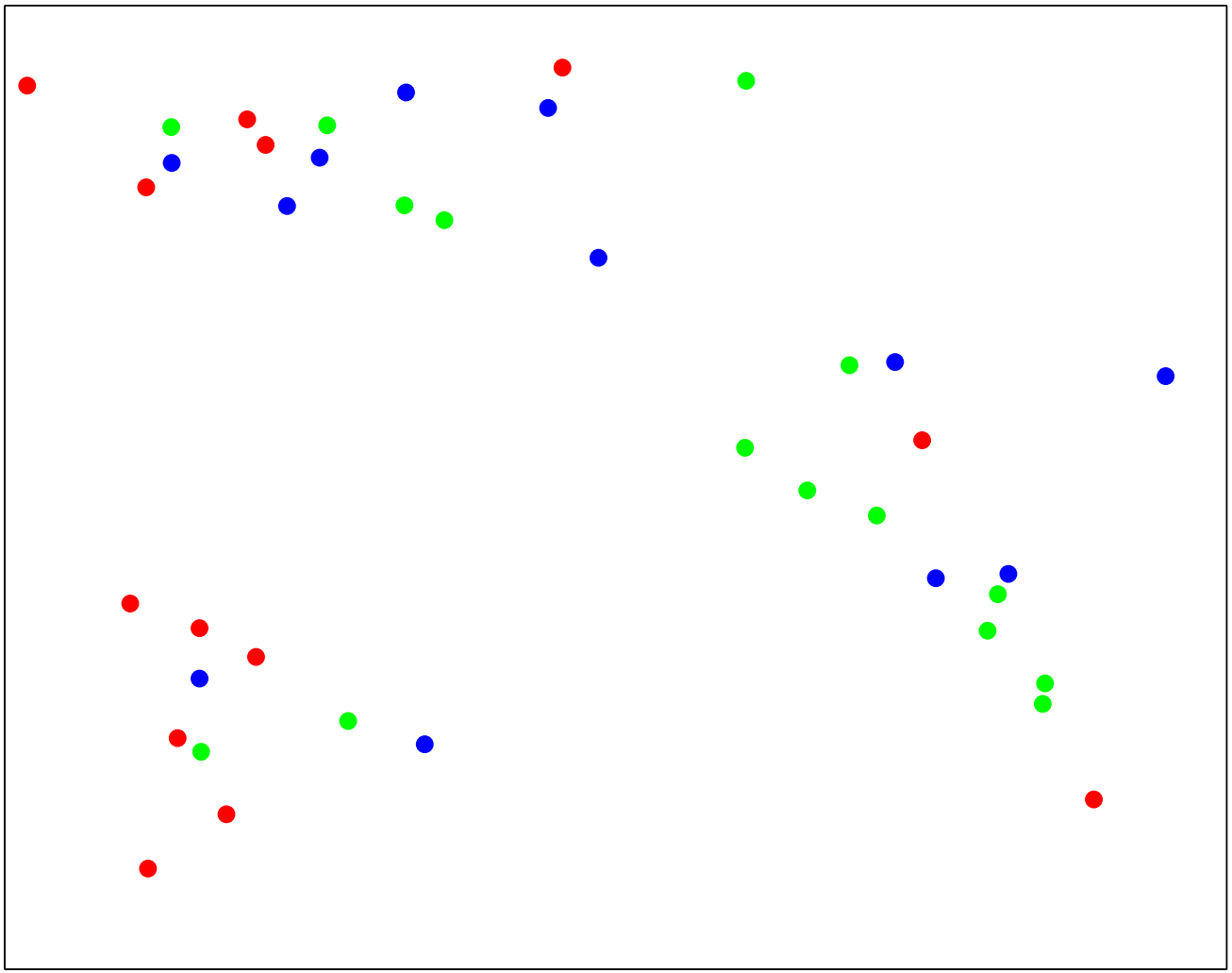


How about 3 clusters?

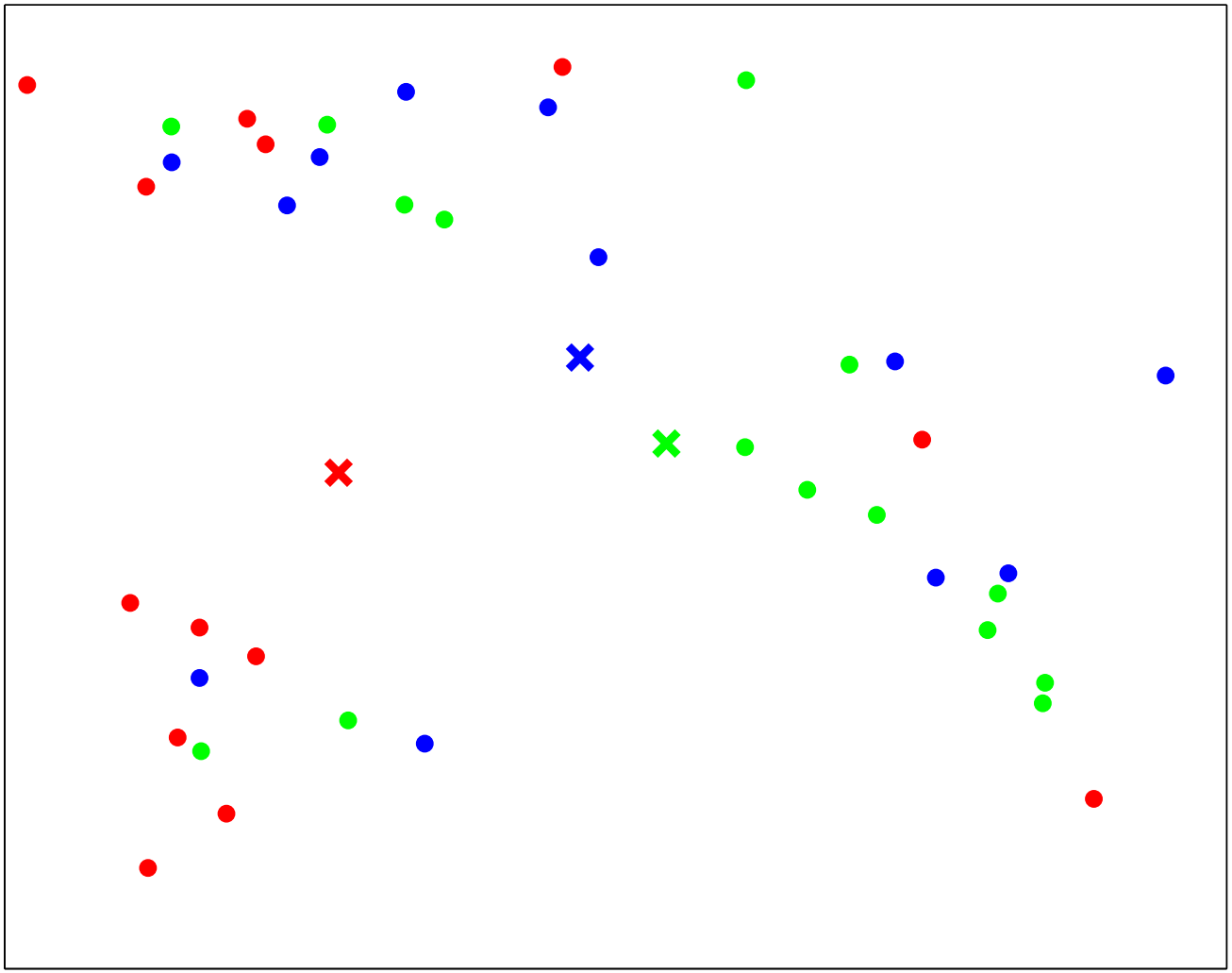
Example: initial data



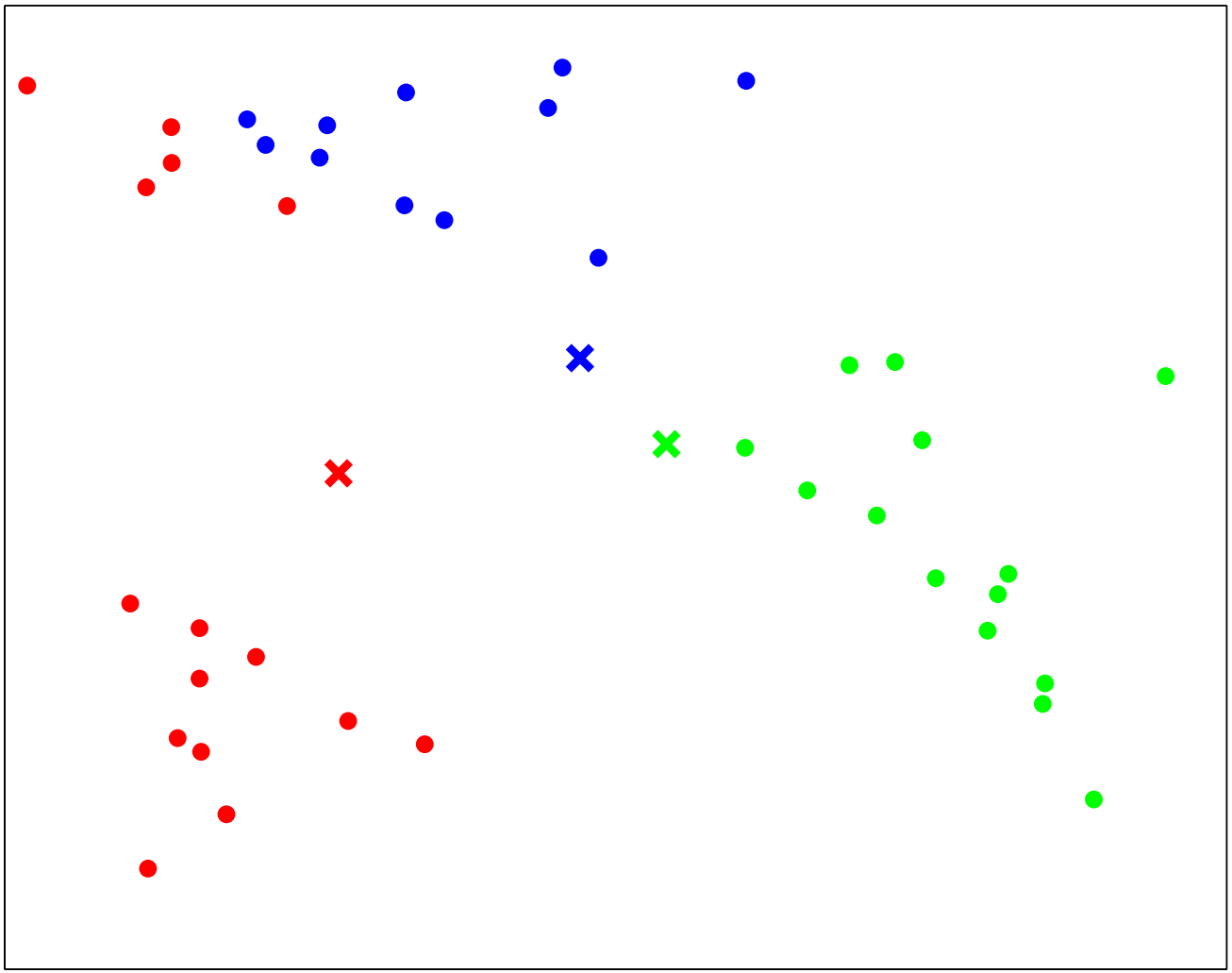
Example: assign into 3 clusters randomly



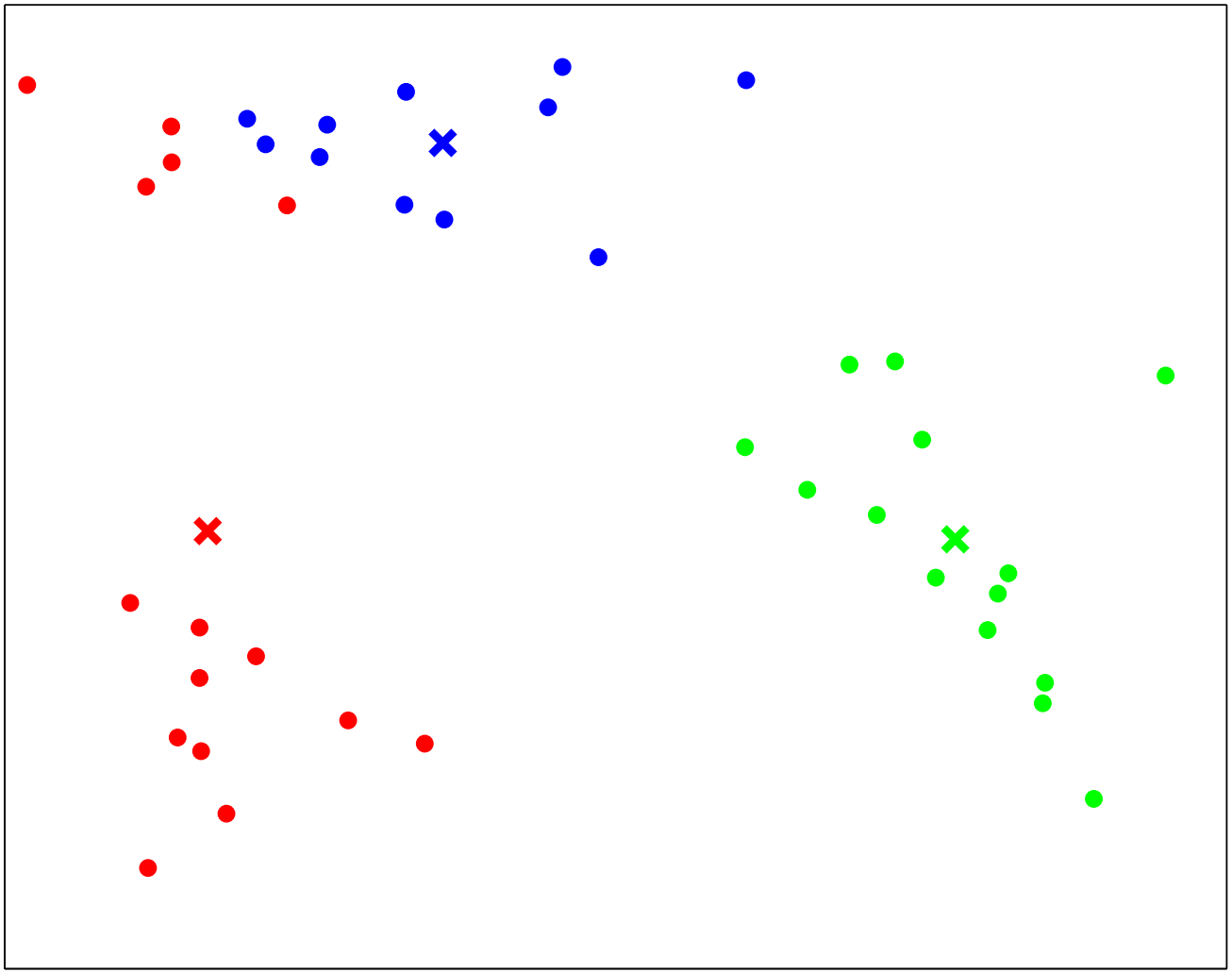
Example: compute centroids



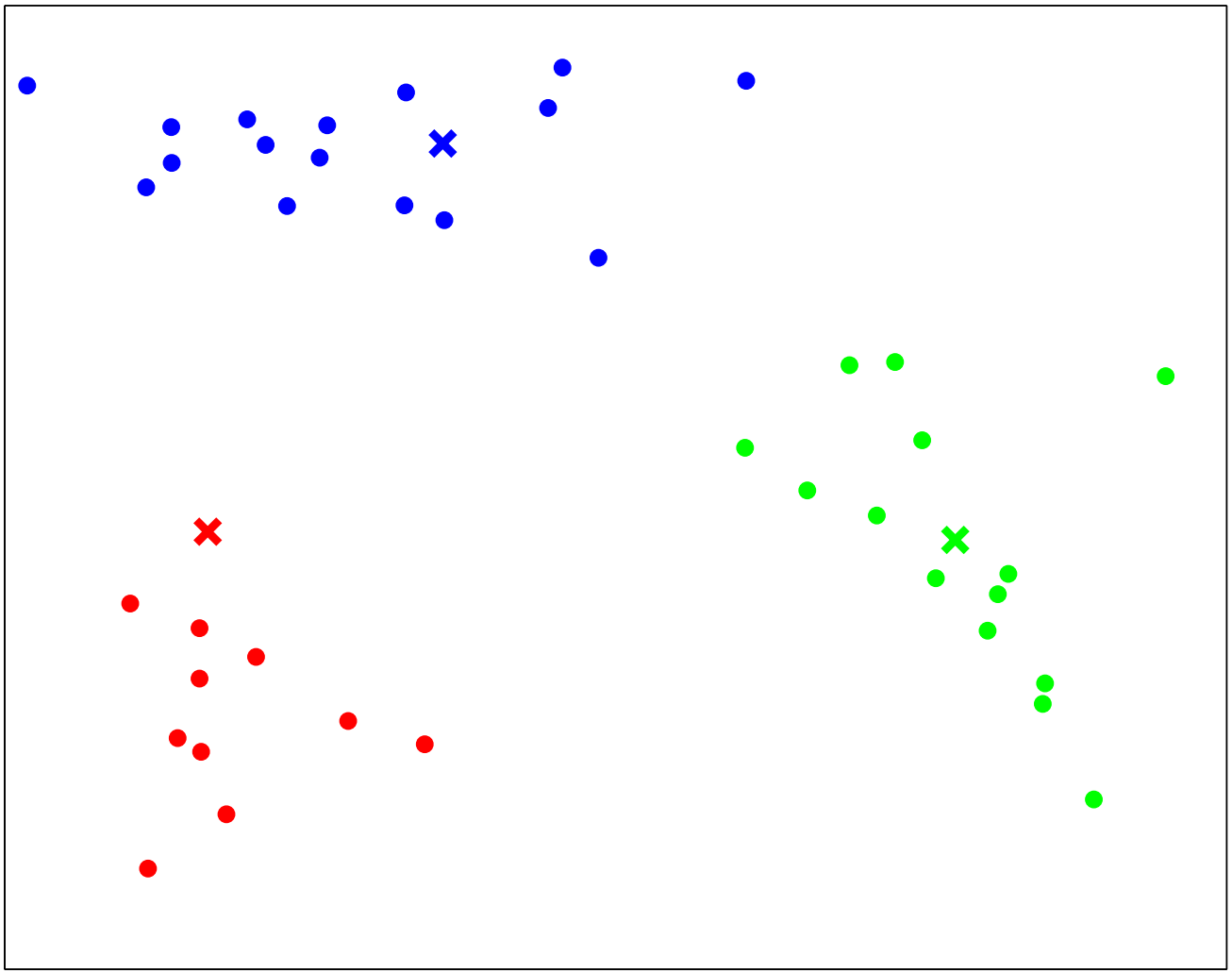
Example: reassign clusters



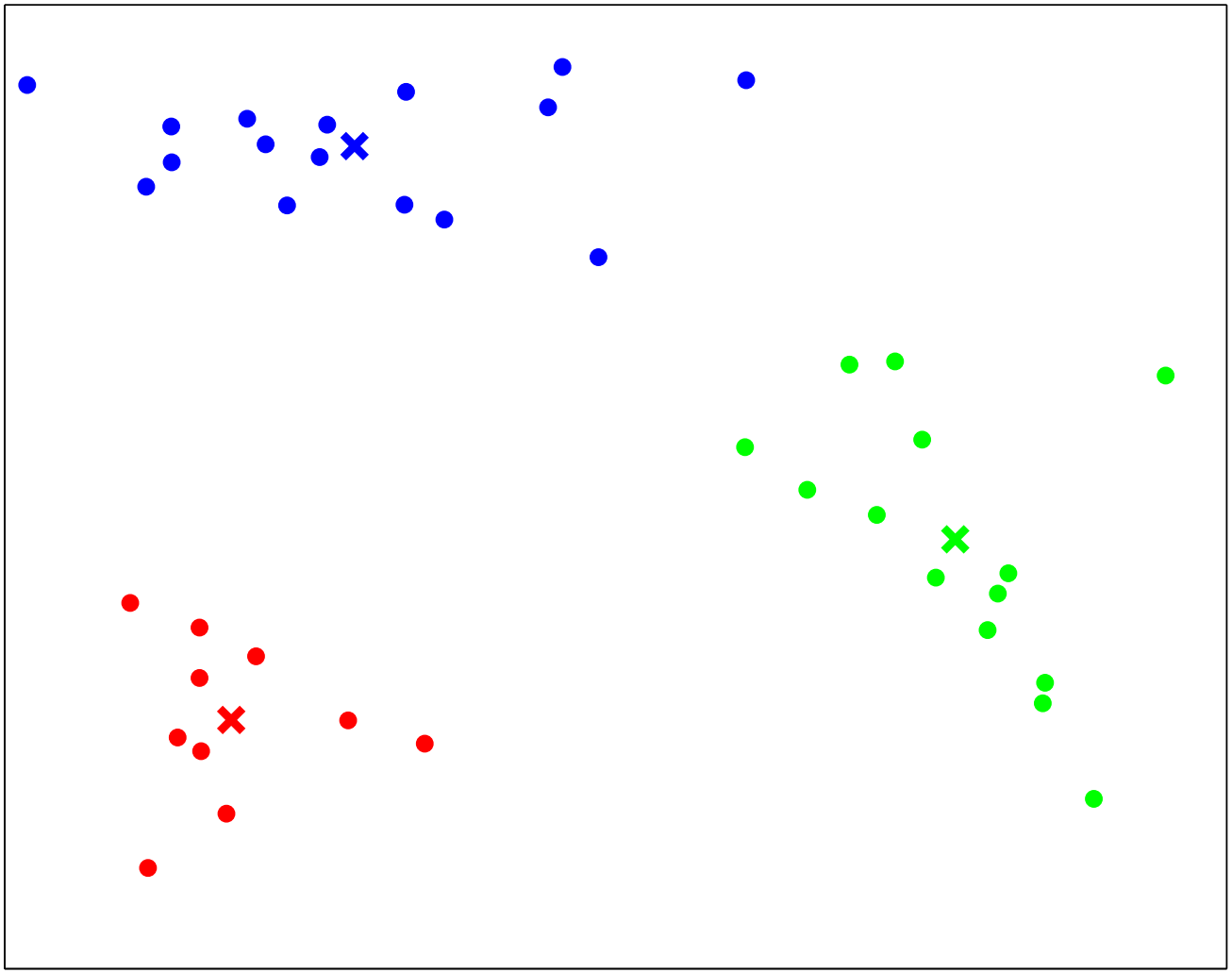
Example: recompute centroids



Example: reassign clusters



Example: recompute centroids – done!



Issues with K-means clustering

- Does the algorithm always terminate?
- Does it always find the same answer?
- How many clusters are there?

Issues with K-means clustering

- Does the algorithm always terminate? yes
- Does it always find the same answer? no
- How many clusters are there? hard to say

Termination of K-means clustering

- For given data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and a clustering C, consider

$$f = \sum_{i=1}^{m} \|\mathbf{x}_i - \mu_{C(i)}\|^2 ,$$

  where μ_j denotes the centroid of the j^{th} cluster.
  (That is, sum the squared Euclidean distance of every vector to the centroid of its cluster.)

- There are finitely many possible clusterings. (At most, K^m.)

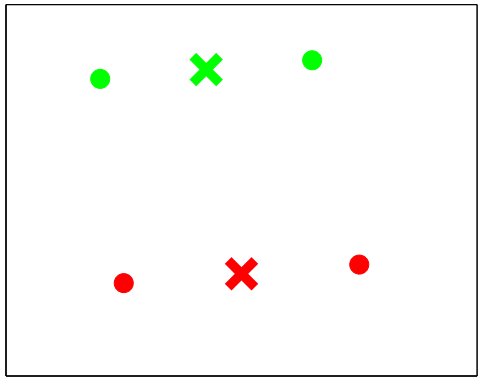
- Each time we reassign a vector to a cluster with a nearer centroid, f decreases.

- Claim: Each time we recompute the centroids of each cluster, f decreases (or stays the same.)

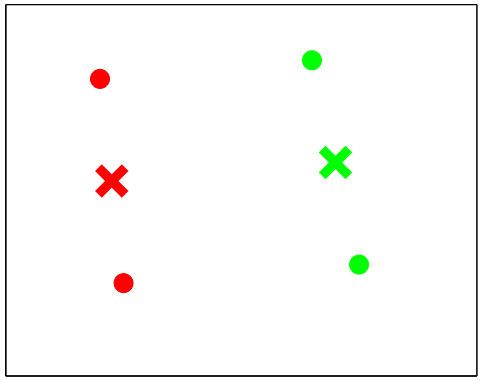
- Thus, the algorithm must terminate.

Interpretation

- So, K-means is an iterative procedure for minimizing the sum of squared Euclidean distances from vectors to their cluster centroid.
- It does not always find the same solution; it may terminate at a suboptimal clustering.



$f = 0.22870$

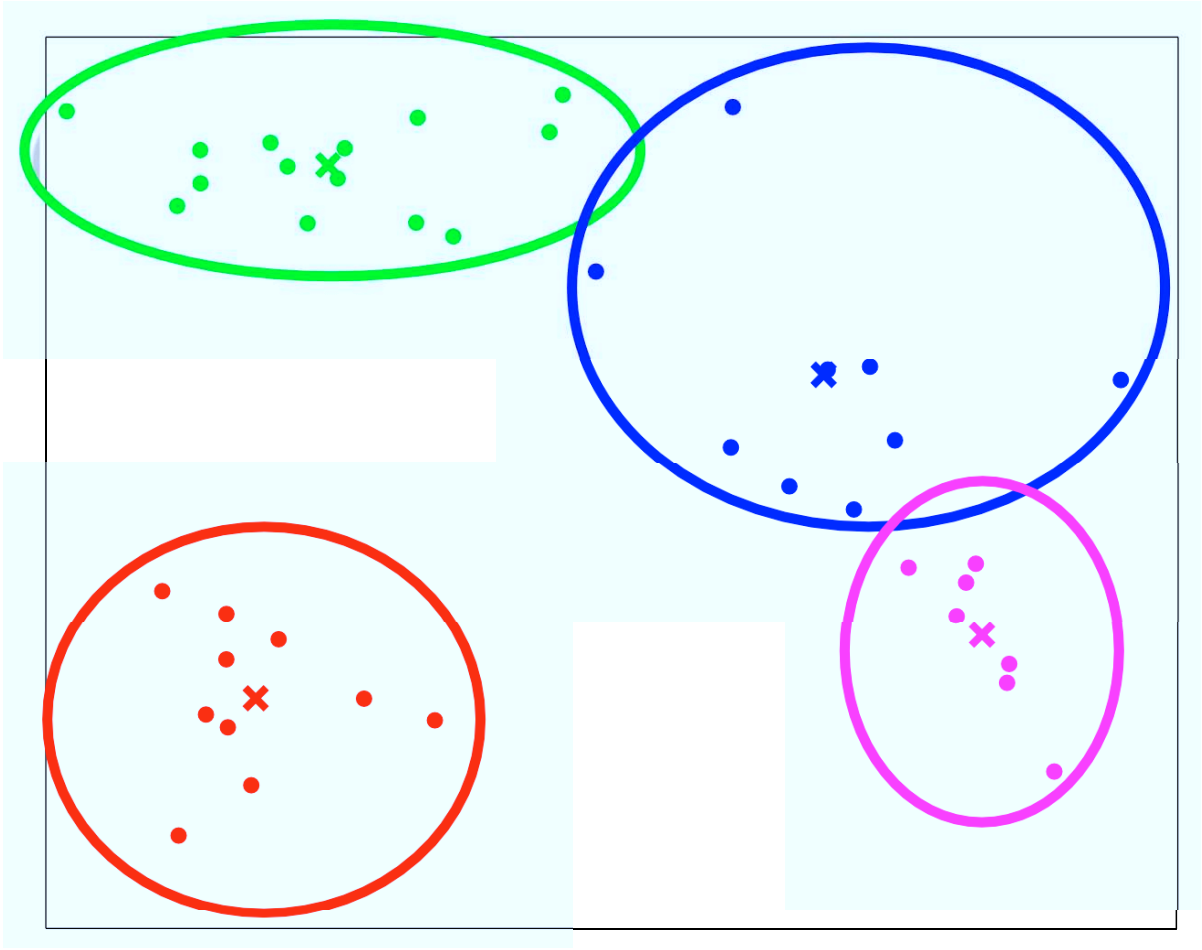


$f = 0.3088$

- Why this choice of f?
- Can we make other choices for f? (Yes, as we'll see shortly.)

Why the sum of squared Euclidean distances?

Reason 1: It produces nice, round clusters.



Why the sum of squared Euclidean distances?

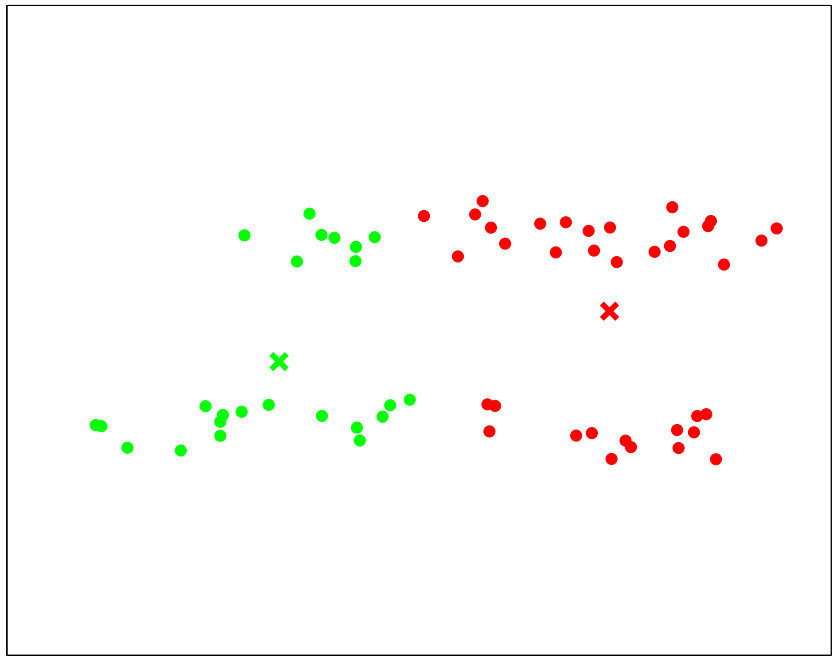
Reason 2: The Principle of Maximum Likelihood.

Details in a future class. Roughly:

- Suppose the data really does divide into K clusters.
- Suppose the data in each cluster is generated by a multivariate Gaussian, where
  - The mean of the Gaussian is the centroid of the cluster.
  - The covariance matrix is of the form $\sigma^2 I$. (It's a "round" Gaussian.)
- Then the probability of the data is highest when the sum of squared Euclidean distances is smallest.

Why *not* the sum of squared Euclidean distances?

Reason 1: It produces nice round clusters!



Reason 2: Differently scaled axes can dramatically affect results.
Reason 3: And what if our objects aren't vectors at all or have symbolic elements (like A,C,G,T)?

A more general approach

Given a set of objects,

- Choose a notion of pairwise distance / similarity between the objects.
- Choose a scoring function, that represents some notion of clustering.
- Optimize the scoring function, to find a good clustering.

(For most choices, the optimization problem will be intractable. Local optimization is the usual recourse.)

Example notions of distance

- Euclidean distance
- Hamming distance
- Estimated evolutionary distance (as between strings of DNA or species)
- Number of shared motifs (as between strings of DNA) or domains (proteins)
- Number of shared transcription factors (between two genes)
- …

Example notions of scoring functions

- Minimize: Summed distances between all pairs of objects in the same cluster. (Also known as ”within-cluster scatter.”)
- Minimize: Maximum distance between any two objects in the same cluster. (Can be hard to optimize.)
- Maximize: Minimum distance between any two objects in different clusters.
- . . .

Summary / Notes

- Flat clustering partitions a data set into disjoint subsets.
- K-means clustering:
  - Is a widely-applied algorithm for partitioning a set of real vectors.
  - It iteratively attempts to optimize the sum of squared Euclidean distances from the vectors to their cluster centers.
- Other clustering algorithms can be generated by different choices of distance measure and scoring function.
- Clusterings are not right or wrong; different clusterings may reveal different aspects of the data.
- Domain knowledge, and experimentation, can help you choose the right notions of distance and clustering.
- How do you know if you have good clusters? Or even the right number of clusters?