Digital Image & Video Processing

Lecture 9
Image Segmentation

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

9. Image Segmentation

9.1. Problem Statement

9.2. Method

9.3. Applications

9.1. Problem Statement

Let us suppose that an image domain I must be segmented in N different regions $R_1, R_2, \ldots, R_N$.

It is necessary to determine a segmentation rule is a logical predicate of the form $P(R)$ having the following properties:

$$I = \bigcup_{i=1}^{N} R_i$$

$$R_i \bigcap R_j = \varnothing, \quad i \neq j$$

$$P(R_i) = True, \quad i = 1,2,\ldots,N$$

$$P(R_i \bigcup R_j) = False, \quad i \neq j$$

9.2. Method

9.2.1. Region growing method

Principle

Geometrical proximity + homogeneity -> connected image regions.

9.2. Method

9.2.1. Region growing method

Method

- Starting from some pixels (seeds) representing distinct iamge regions and to grow them, until they cover the entire image.

- In order to implement region growing, we need determine:

. Seeds

. A rule describing a growth mechanism.

. A rule checking the homogeneity of the regions after each growth step..

9.2. Method

9.2.1. Region growing method

Method

. **Seeds**

Based on the histogram, choose the seed points corresponding to the histogram peaks.

9.2. Method

9.2.1. Region growing method

Method

. A rule describing a growth mechanism

Growing based on 8-neighborhood.

9.2. Method

9.2.1. Region growing method

Method

. A rule checking the homogeneity of the regions after each growth step.

At each step k, for each region $R_i^{(k)}, i = 1,2,\ldots,N$

We check if there are unclassified pixels in the 8-neighbourhood of each pixel of the region border.

If $P(R_i^{(k)} \bigcup \{b_i^{(k)}(x,y)\}) = True$ assigning $b_i^{(k)}(x,y)$ to region $R_i^{(k)}$.

9.2. Method

9.2.1. Region growing method

Method

. A rule checking the homogeneity of the regions after each growth step.

$$| f(x,y) - m(R_i^{(k)}) | < T$$

$$m(R_i^{(k)}) = (1/N(R_i^{(k)})) \sum_{(k,l)\in R_i^{(k)}} f(k,l)$$

$$\sigma(R_i^{(k)}) = \left[(1/N(R_i^{(k)})) \sum_{(k,l)\in R_i^{(k)}} (f(k,l) - m(R_i^{(k)}))^2 \right]^{1/2}$$

9.2. Method

9.2.1. Region growing method

Method

. A rule checking the homogeneity of the regions after each growth step.

$$m(R_i^{(k+1)}) = (1/(N(R_i^{(k)})+1))[f(x,y) + N(R_i^{(k)})m(R_i^{(k)})]$$

$$\sigma(R_i^{(k+1)}) = \begin{bmatrix} (1/(N(R_i^{(k)})+1))[N(R_i^{(k)})\sigma^2(R_i^{(k)}) + \\ (N(R_i^{(k)})/(N(R_i^{(k)})+1))(f(x,y) - m(R_i^{(k)}))] \end{bmatrix}^{1/2}$$

9.2. Method

9.2.1. Region growing method

Method

. A region merging rule.

$$| m(R_i^{(k+1)}) - m(R_{i'}^{(k+1)}) | < k\sigma(R_i^{(k+1)})$$

$$| m(R_i^{(k+1)}) - m(R_{i'}^{(k+1)}) | < k\sigma(R_{i'}^{(k+1)})$$

9.2. Method

9.2.2. K-means method

Principle

Unsupervised Clustering homogeneity regions -> Image Regions.

9.2. Method

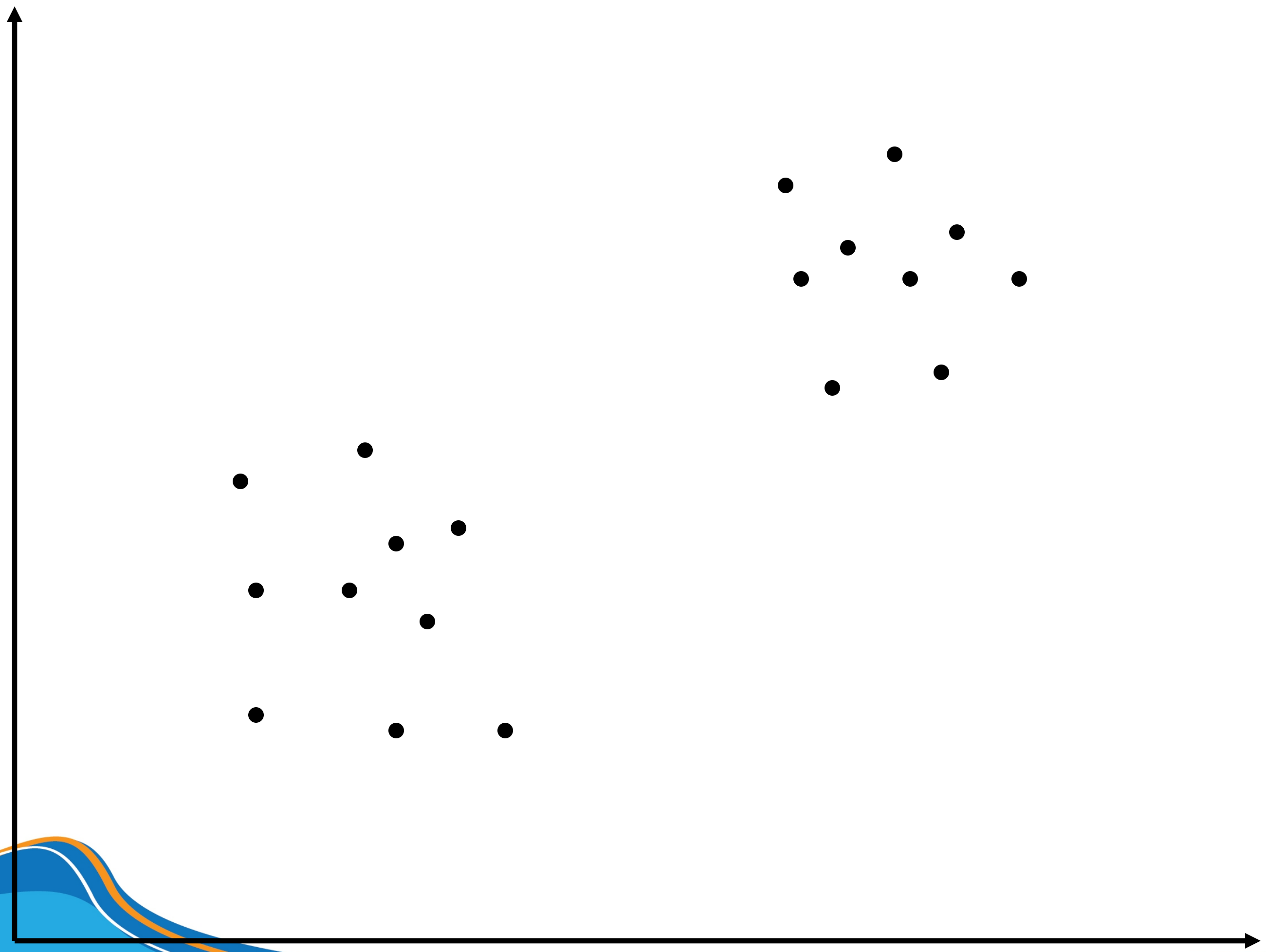
9.2.2. K-means method

Method

1. Partition the data points into K clusters randomly. Find the centroids of each cluster.
2. For each data point:
    - Calculate the distance from the data point to each cluster.
    - Assign the data point to the closest cluster.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until there is no further change in the assignment of data points (or in the centroids).

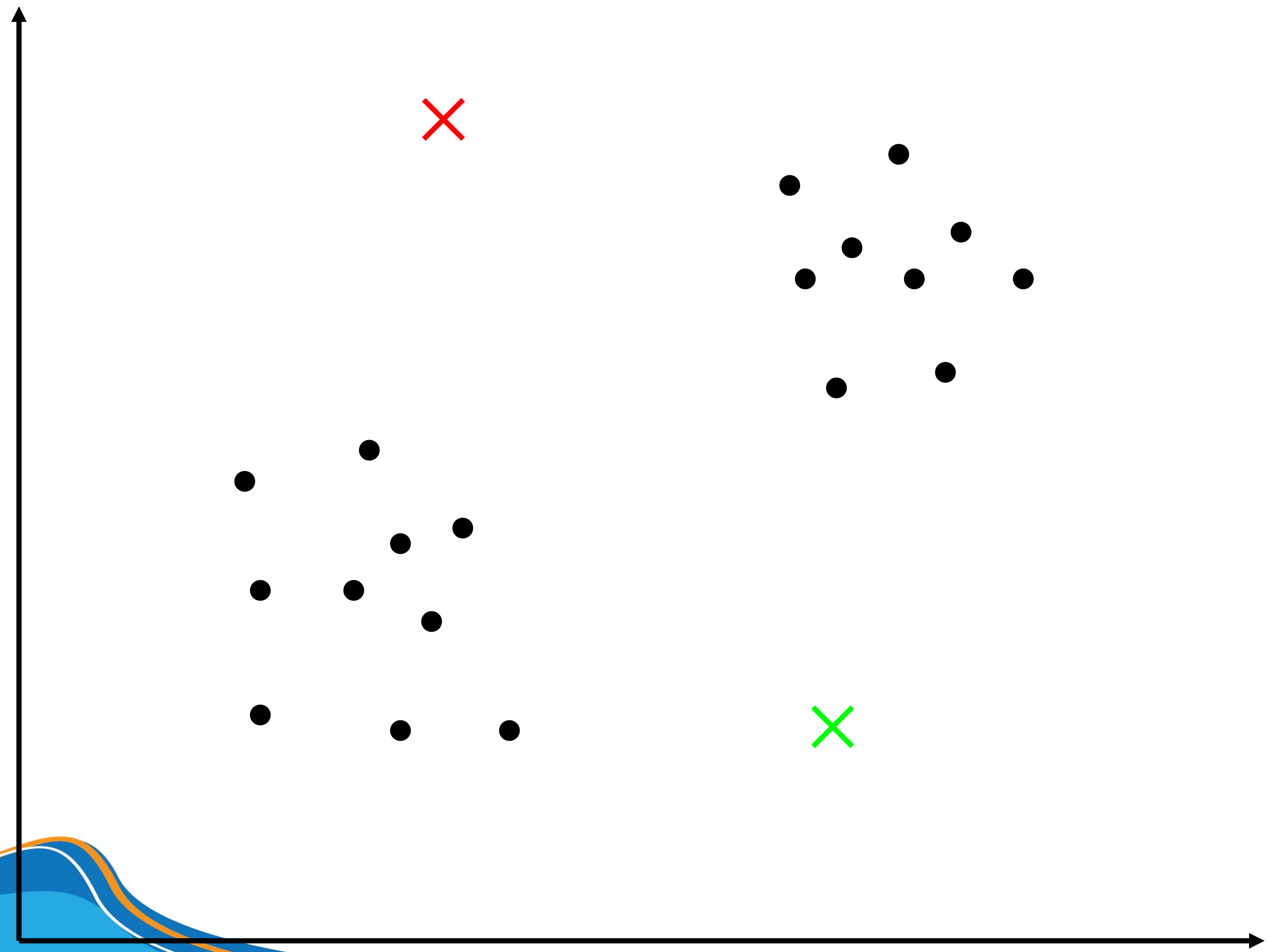
9.2. Method

9.2.2. K-means method



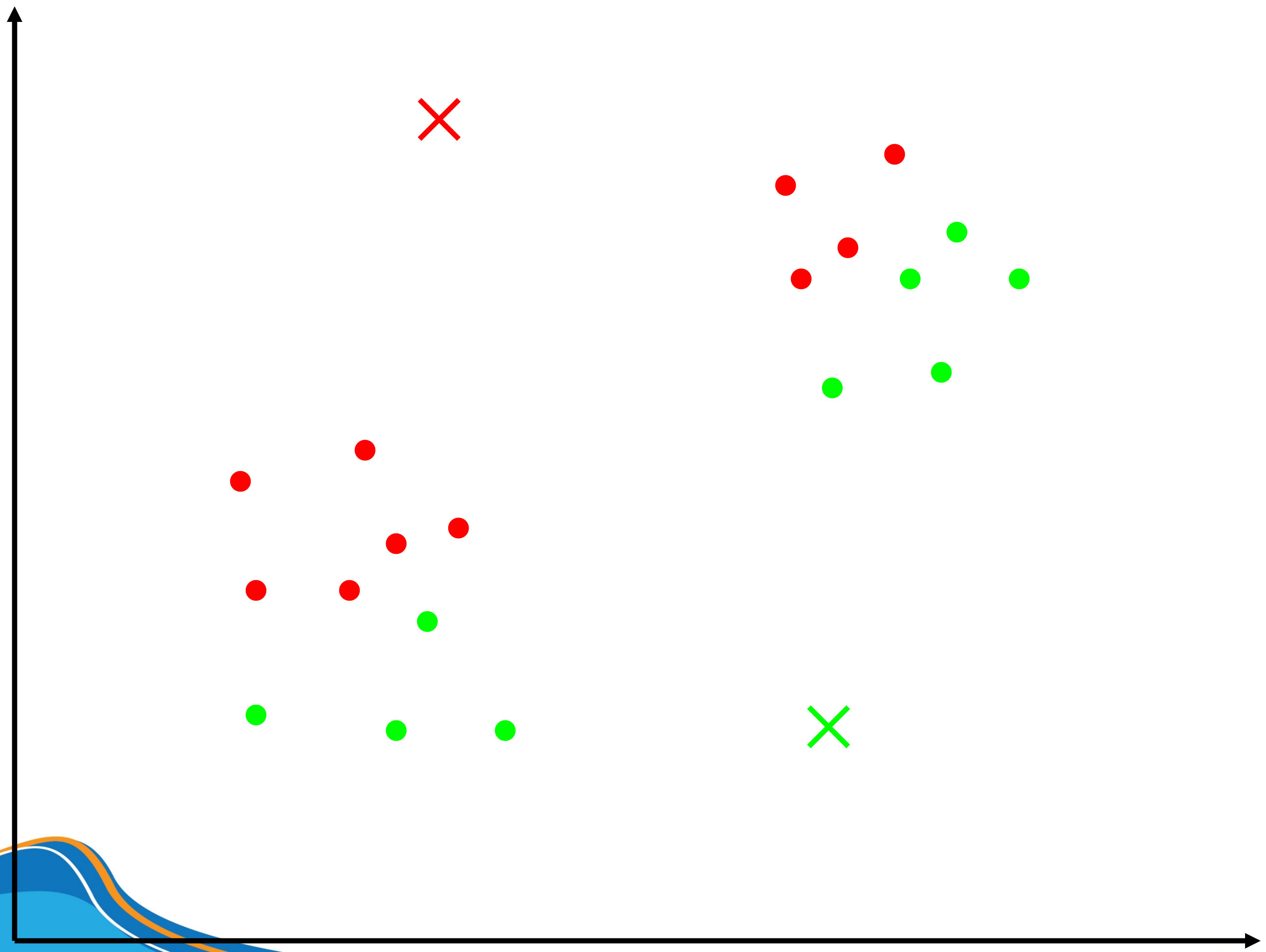
9.2. Method

9.2.2. K-means method



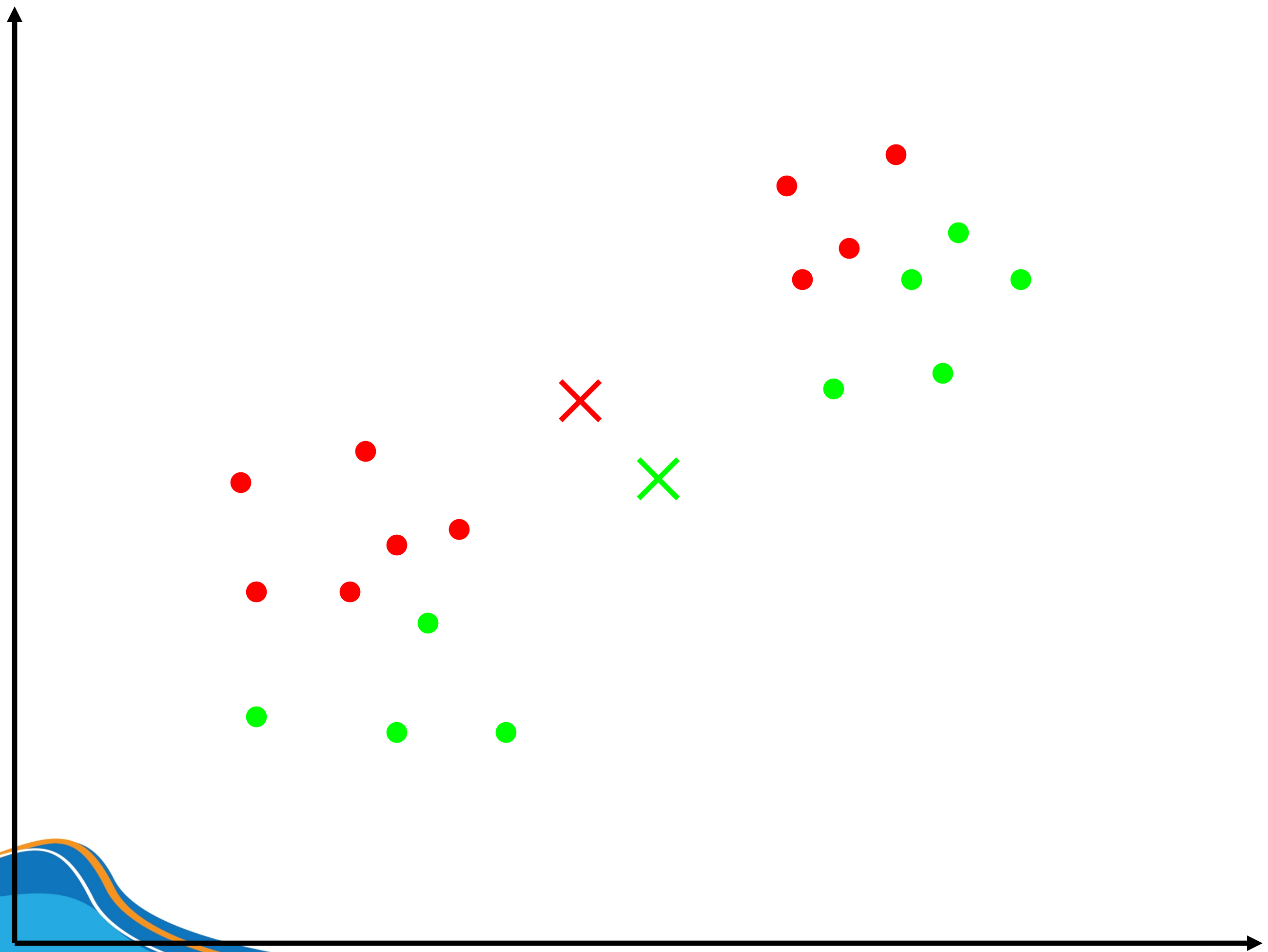
9.2. Method

9.2.2. K-means method



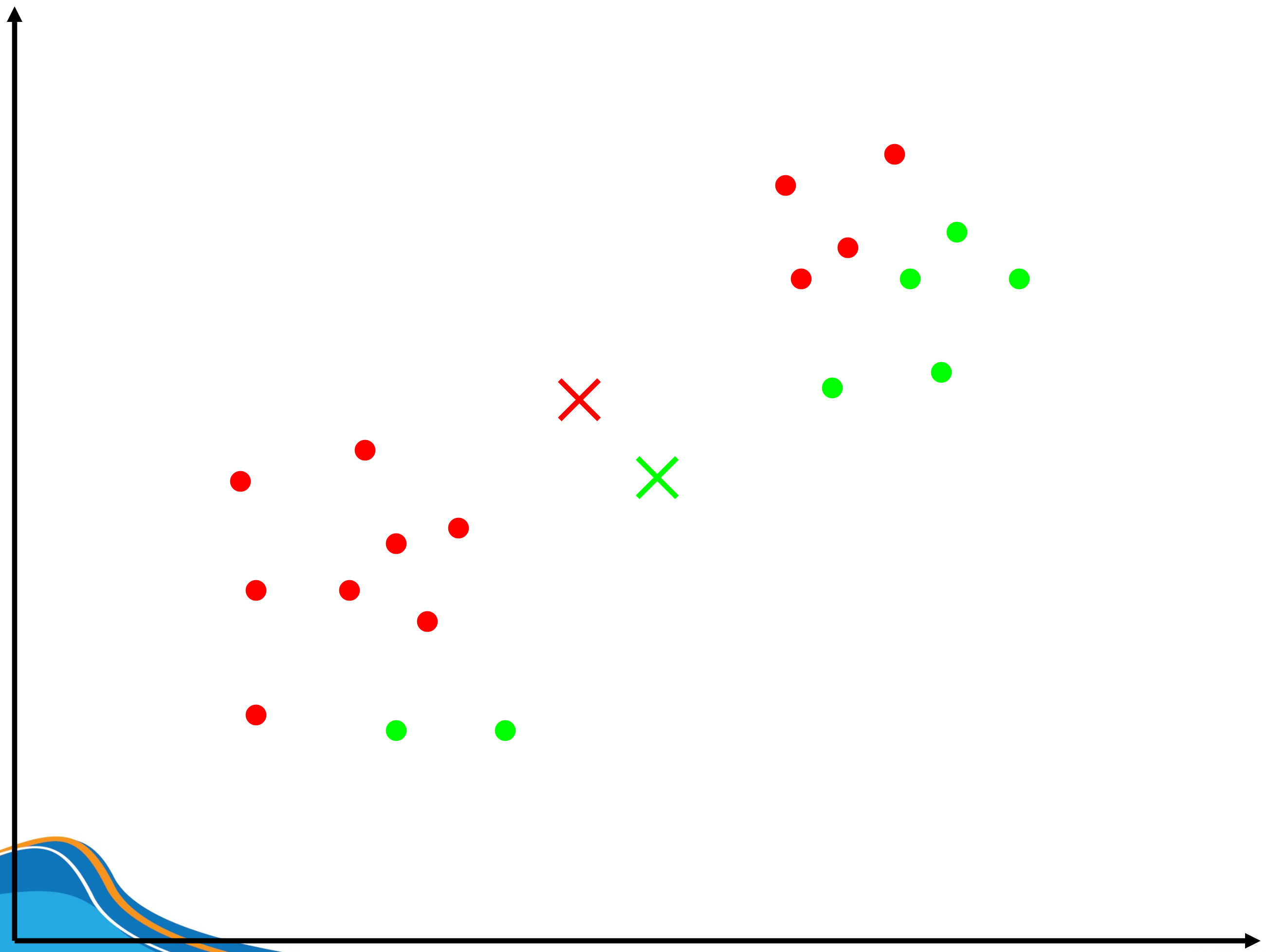
9.2. Method

9.2.2. K-means method



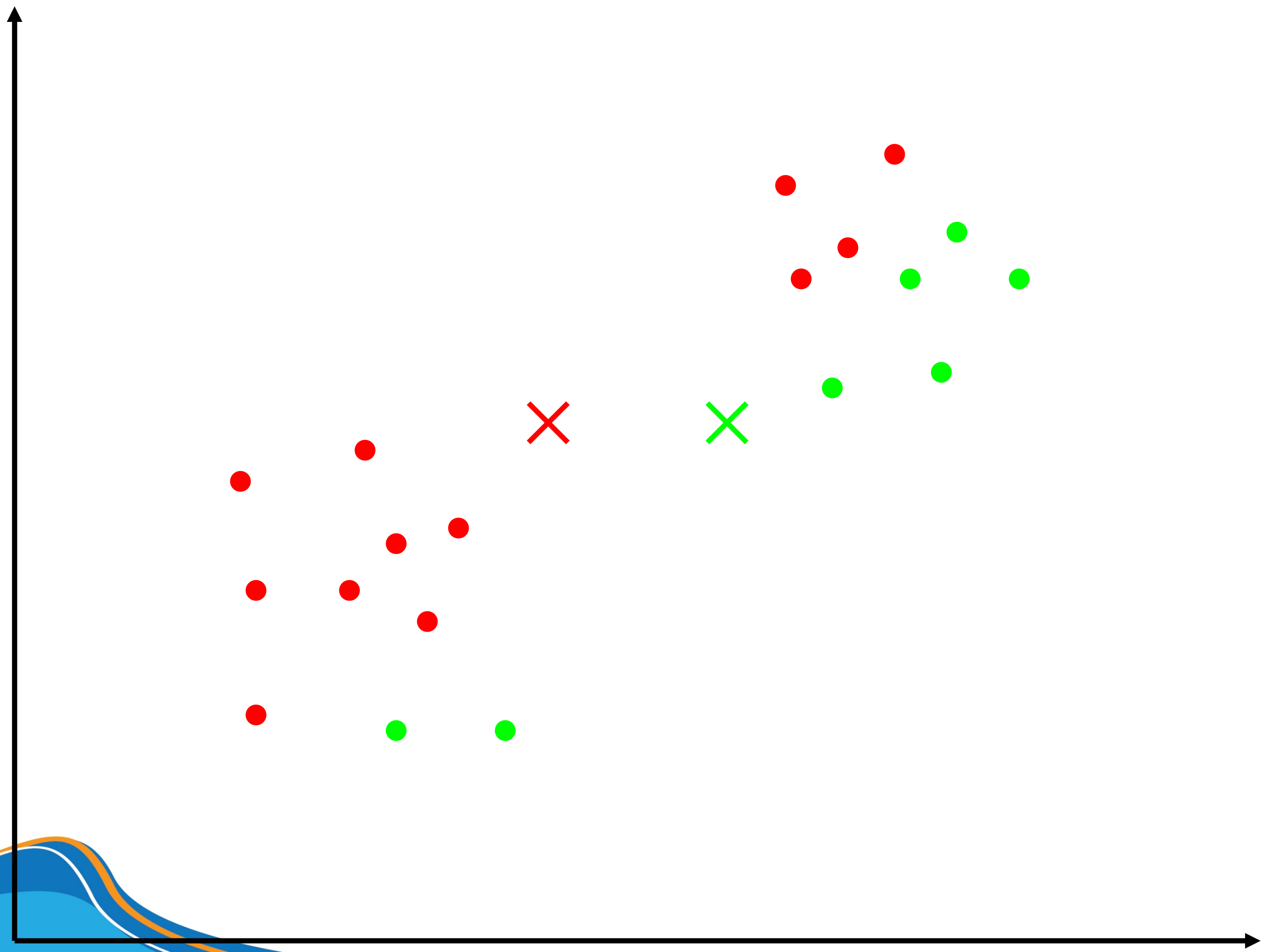
9.2. Method

9.2.2. K-means method



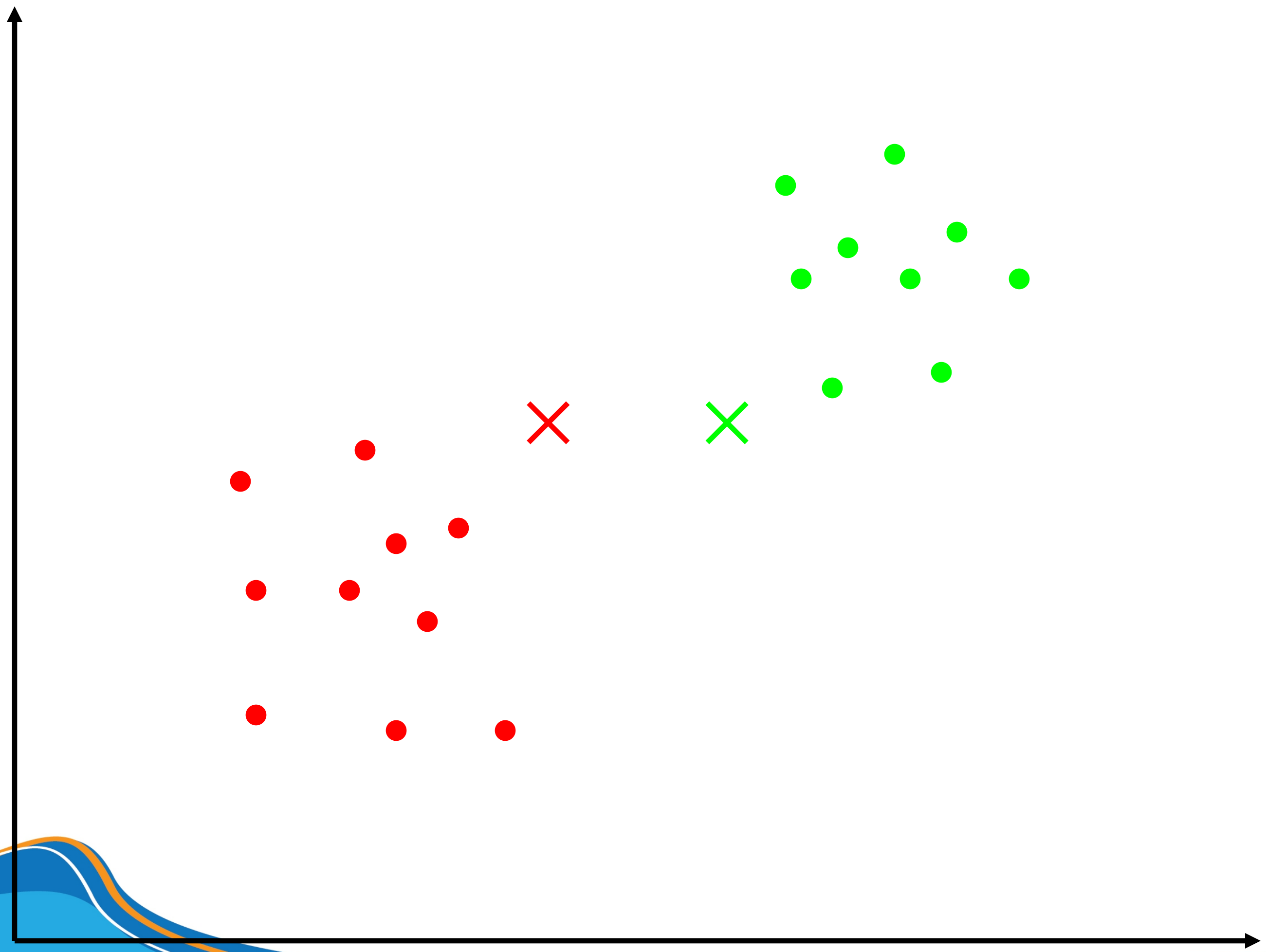
9.2. Method

9.2.2. K-means method



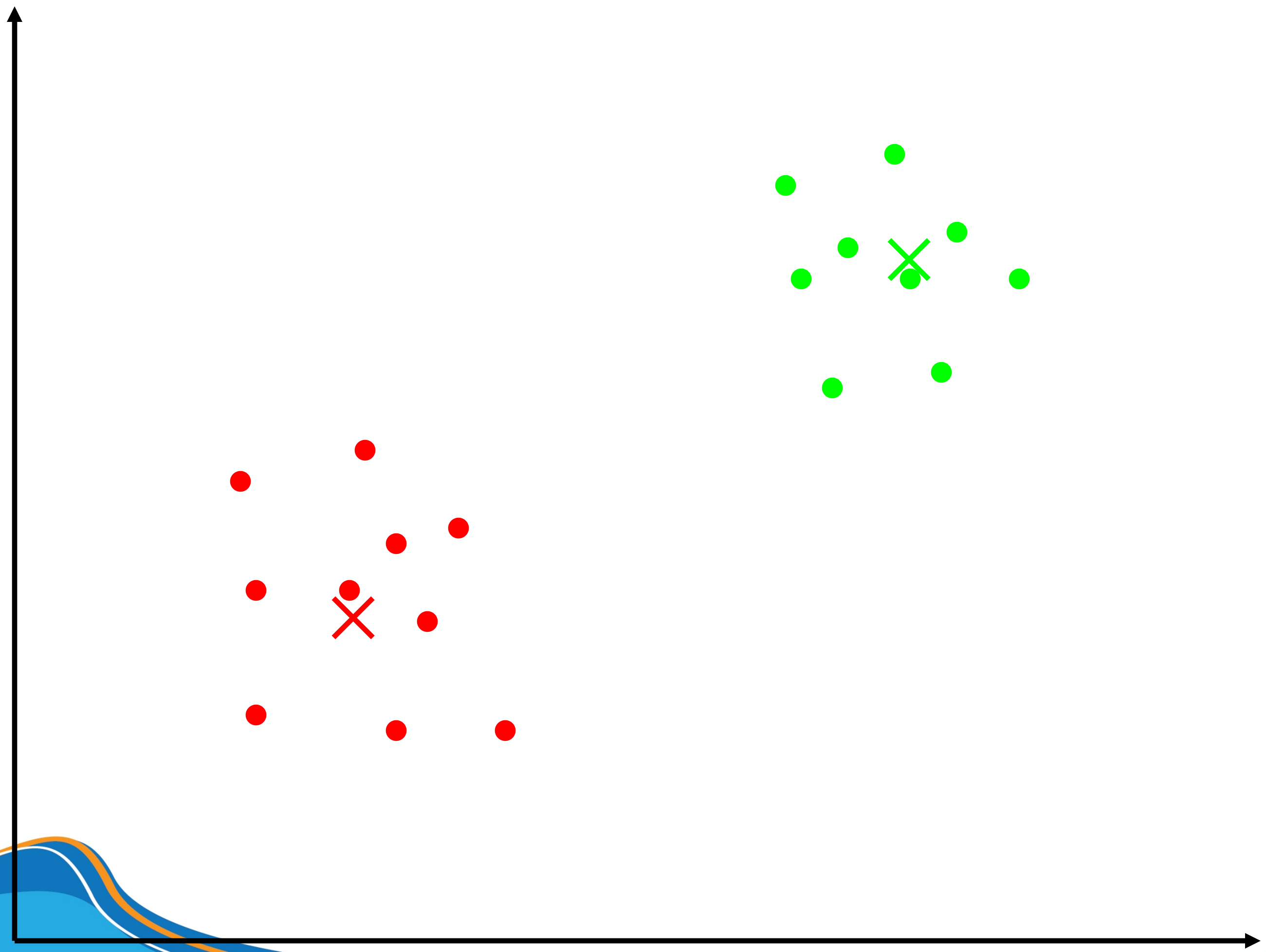
9.2. Method

9.2.2. K-means method



9.2. Method

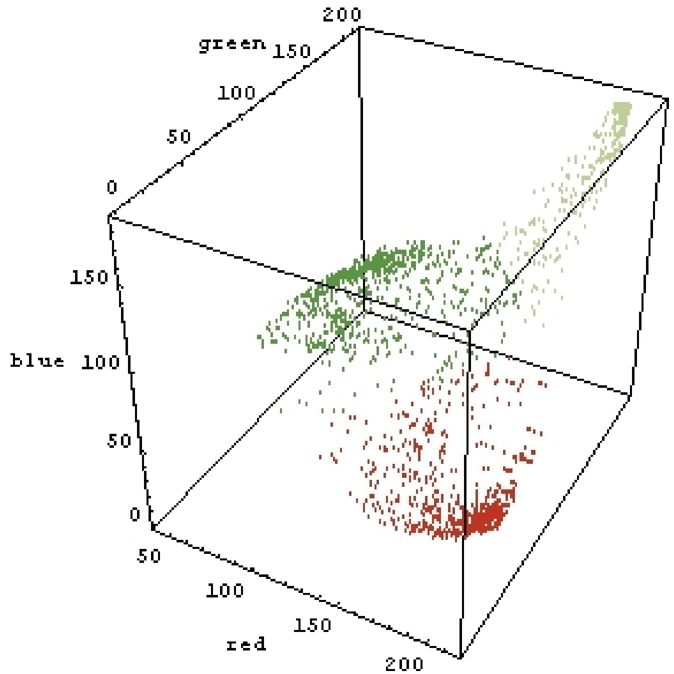
9.2.2. K-means method



9.2. Method

9.2.2. K-means method

- RGB vector



K-means clustering minimizes

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

9.2. Method

9.2.2. K-means method

- Example

Original

K=5

K=11

9.2. Method

9.2.2. K-means method

Algorithm

$Function\ \ K-means()$

$Initialize\ k$ prototypes $(w_1,...,w_k)$ such that $w_j = i_l,\ \mathrm{j} \in \{1,...,\mathrm{k}\}, \mathrm{l} \in \{1,...,\mathrm{n}\}$

Each cluster $\mathrm{C_j}$ is associated with prototype w_j

Repeat

For each input vector i_l , where $\mathrm{l} \in \{1,...,\mathrm{n}\}$,

do

Assign i_l to cluster $\mathrm{C_{j*}}$ with nearest prototype w_{j*}

For each cluster $\mathrm{C_j}$, where $\mathrm{j} \in \{1,...,\mathrm{k}\}$, do

Update the prototype w_j to the centroid of all samples

currently in $\mathrm{C_j}$, so that $w_j = \sum_{i_l \in \mathrm{C_j}} i_l / |\mathrm{C_j}|$

Computer the error function :

$$E = \sum_{j=1}^{k} \sum_{i_l \in \mathrm{C_j}} | i_l - w_j |^2$$

Until E does not change significantly or cluster membership

no longer changes