Introduction to C Programming

Lecture 13: basic pattern recognition

Wenjin Wang
wangwj3@sustech.edu.cn

12-16-2022

Course syllabus

Nr.	Lecture	Date
1	Introduction	2022.9.9
2	Basics	2022.9.16
3	Decision and looping	2022.9.23
4	Array & string	2022.9.30
5	Functions	2022.10.9 (补)
6	Pointer	2022.10.14
7	Self-defined types	2022.10.21
8	I/O	2022.10.28

Nr.	Lecture	Date
9	Head files	2022.11.4
10	Review of lectures I	2022.11.25
11	Review of lectures II	2022.12.2
12	Review of lectures III	2022.12.9
13	AI in C programming	2022.12.16
14	AI in C programming	2022.12.23
15	Summary	2022.12.30

Objective of this lecture

You can use C to write basic AI algorithms!

Content

1. Introduction to pattern recognition (AI)
2. Clustering (Kmeans & DBscan)
3. Classification (KNN & perceptron)

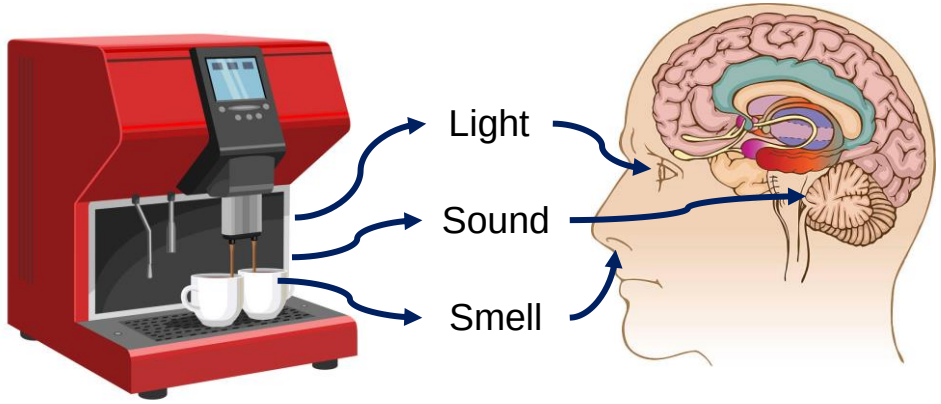
Content

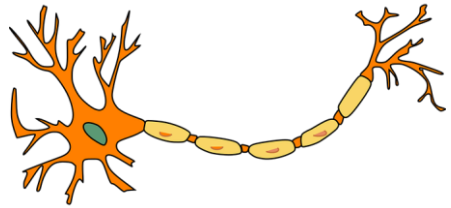
1. **Introduction to pattern recognition (AI)**
2. Clustering (Kmeans & DBscan)
3. Classification (KNN & perceptron)

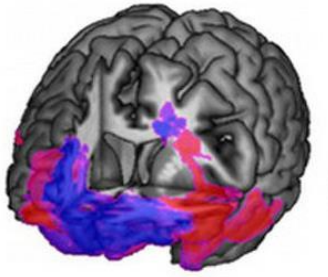
Human perceptions

Humans have developed highly sophisticated skills: **sensing the environment** and **make decisions**

Stimulus → **Sensing** → **Neural impulses** → **Brain**

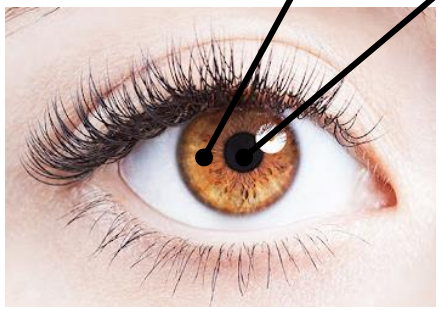






Human perceptions

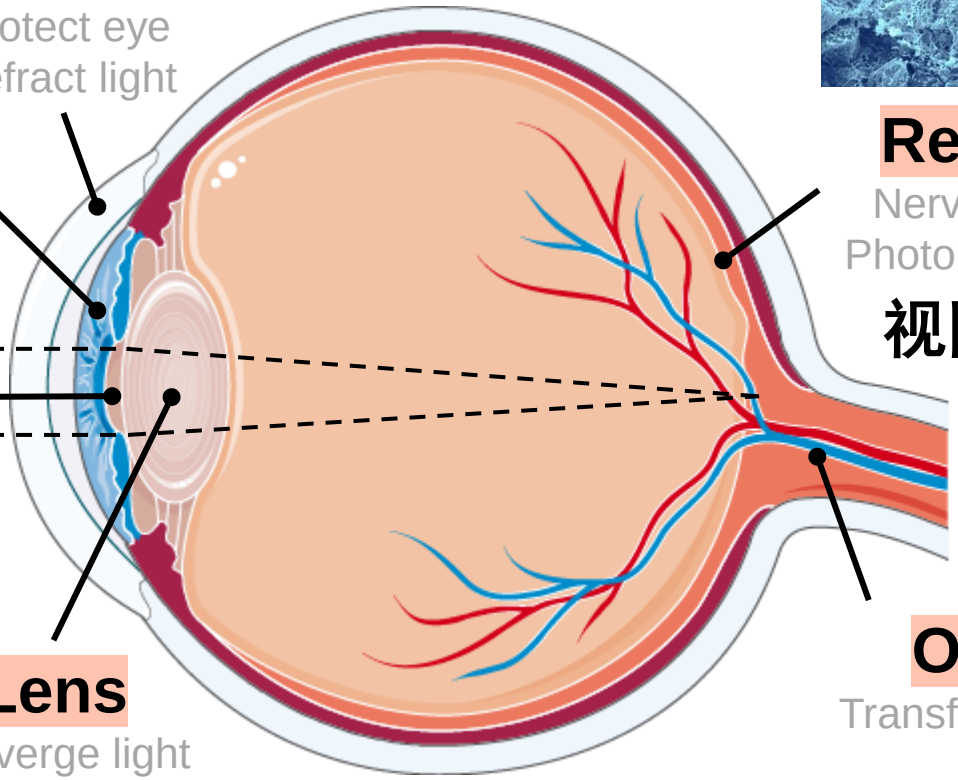
Take vision system as an example



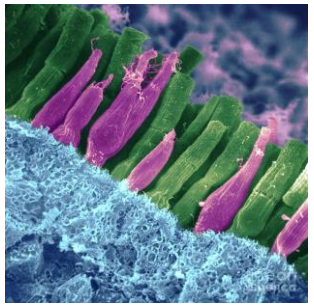
角膜 **Cornea**
Protect eye
Refract light

虹膜 **Iris**
Eye colour
Control amount of light

Pupil
瞳孔

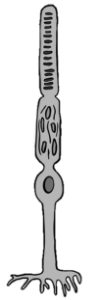


Lens
Converge light

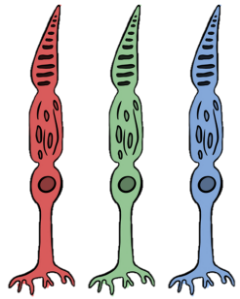


Retina
Nerve cells
Photoreceptor
视网膜

Rods
Dim light
视杆细胞

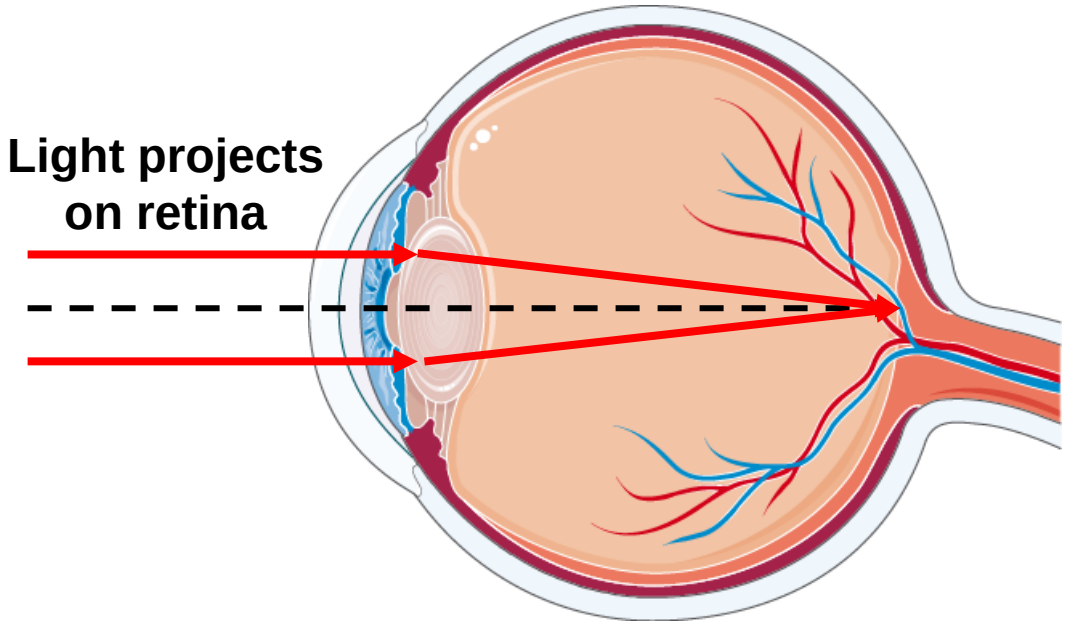


Cones
Bright light
Colours
视锥细胞

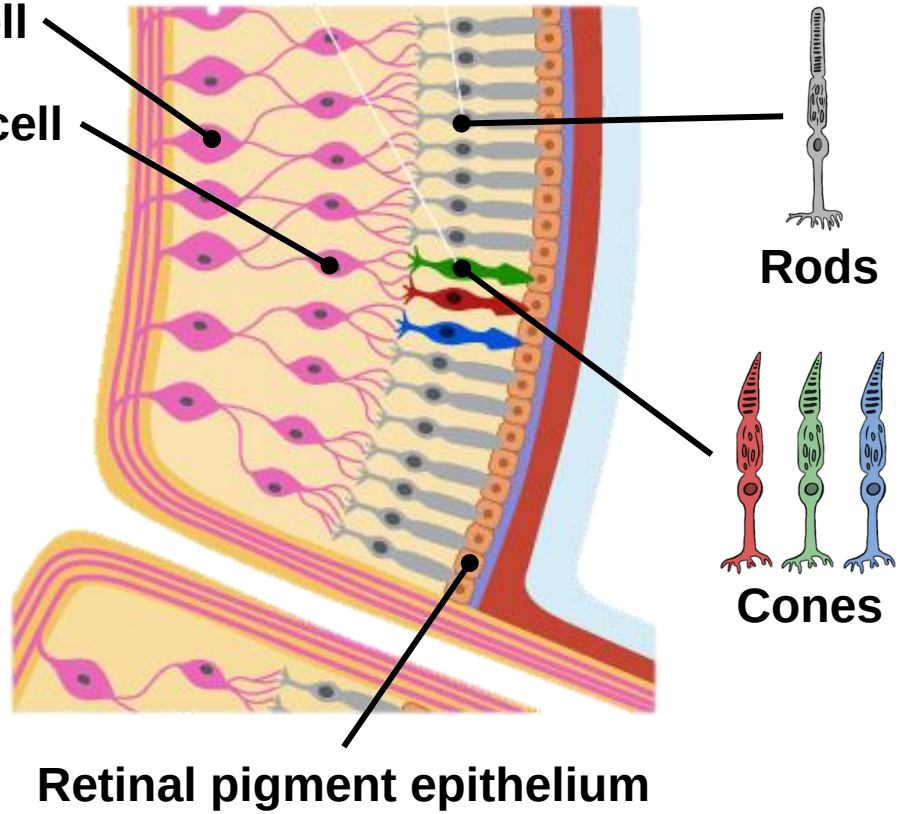


Optic nerve
Transfer message to brain

Human perceptions

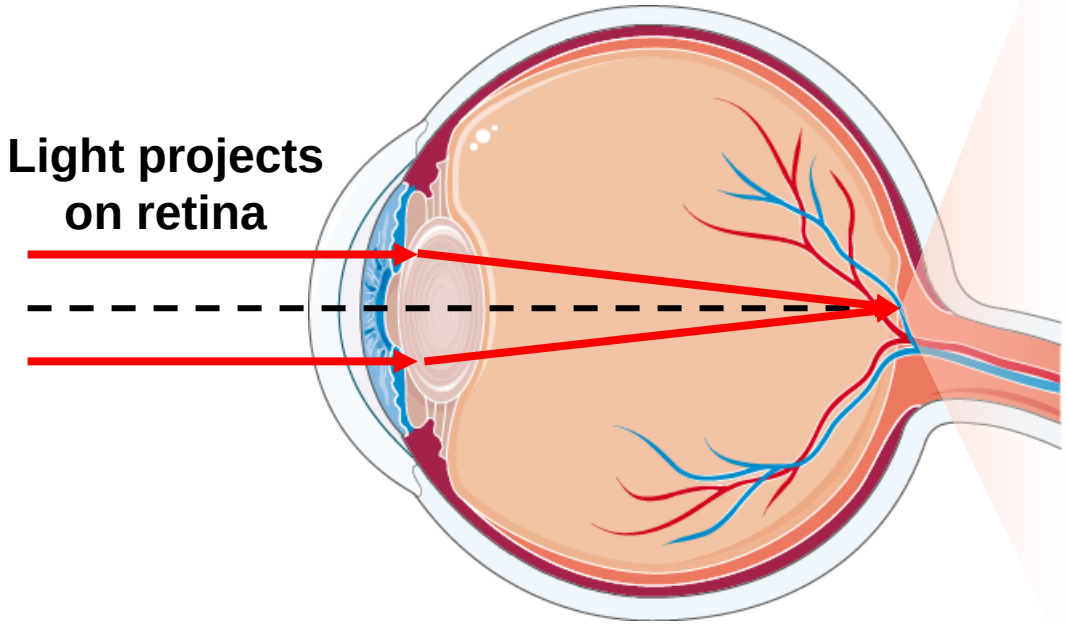


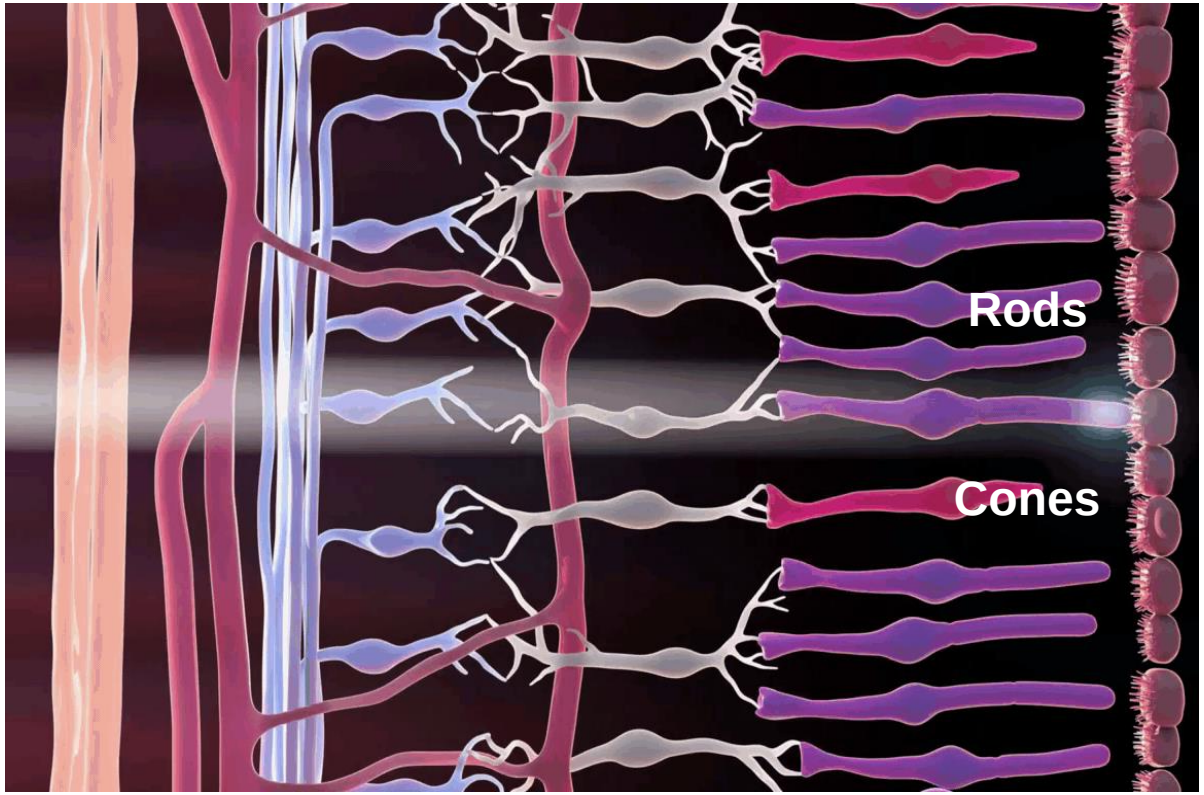
Convert optical signal to electrical impulse!



Human perceptions

Convert optical signal to electrical impulse!



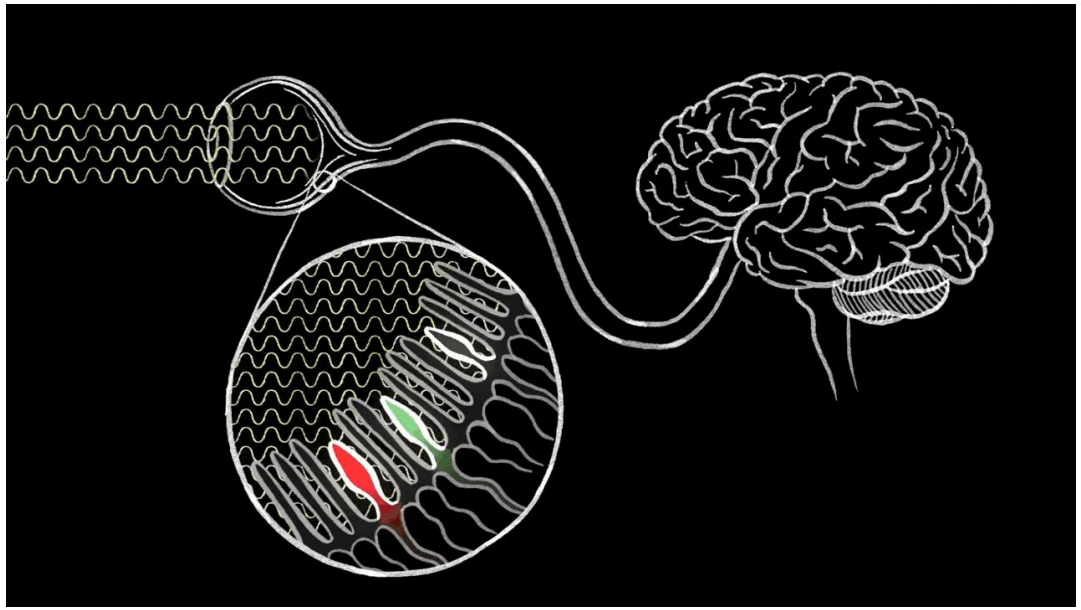


Human perceptions

Sense R/G/B light

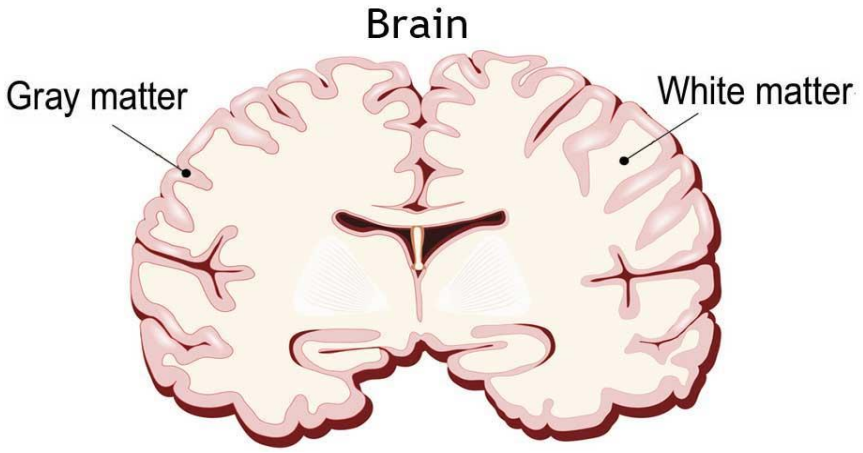
Red, green, blue cones are activated <u>separately</u>

Sense yellow light

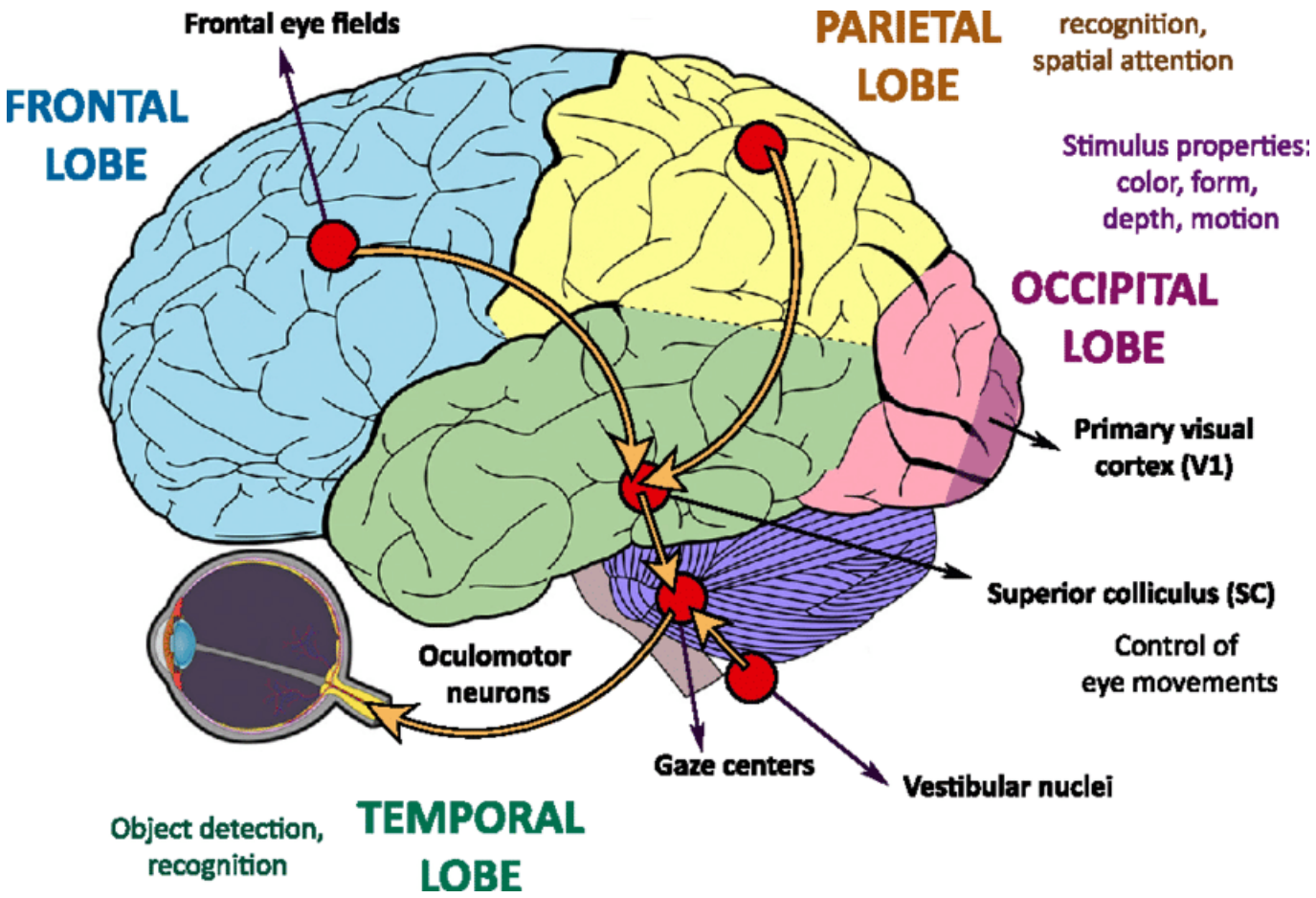


Red and green cones are activated <u>jointly</u>

Human perceptions

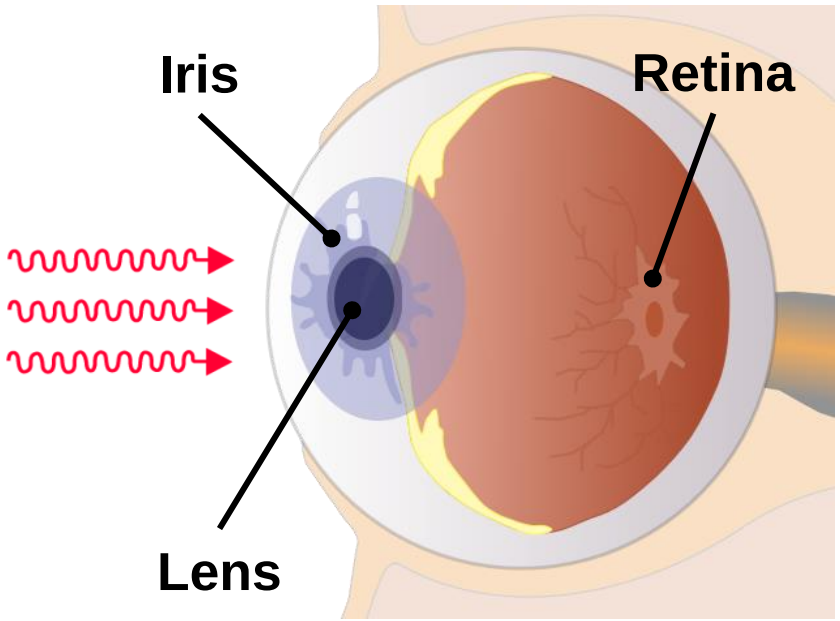


- Gray matter and white matter are two different regions of the central nervous system.
- Gray matter is for processing and interpreting information
- white matter transmits information to other parts of the nervous system.



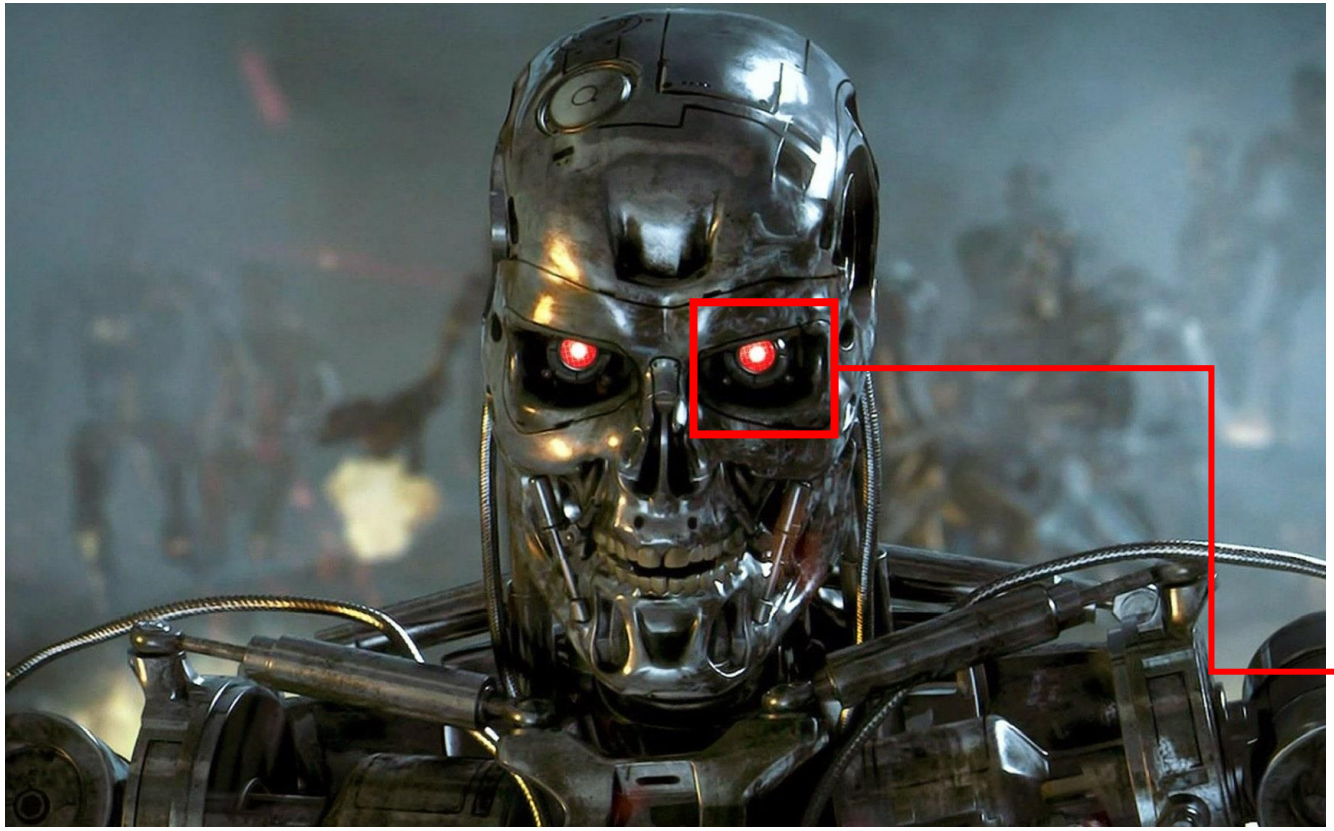
Human perceptions

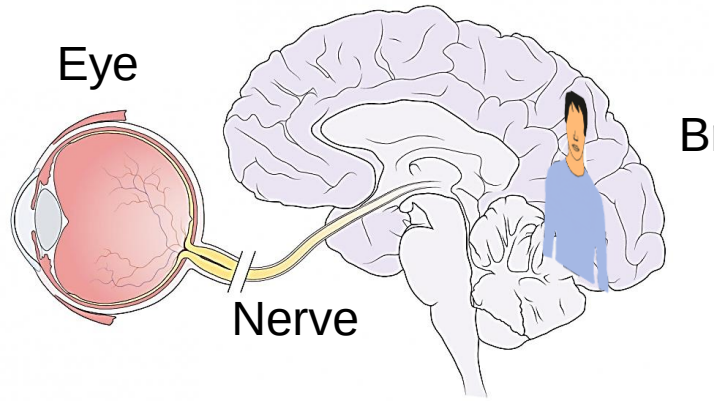
We would like to give similar capabilities to machines.

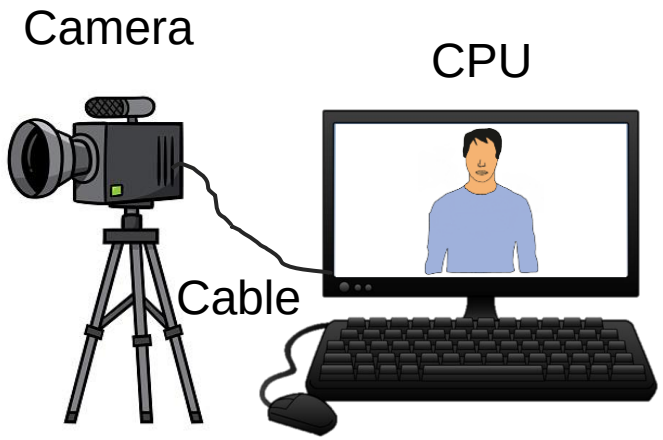


Machine perception

Sensing and recognition







Pattern

What is a pattern?

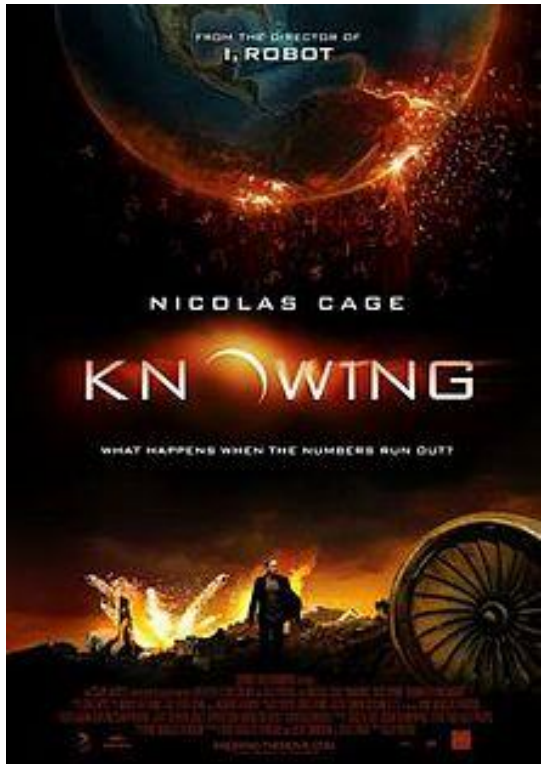
- ✓ Pattern is opposite to chaos. It means **orders** and **relations**.
- ✓ Pattern is an entity that can be **repeated**.
- ✓ Pattern is a **meaningful** representation.

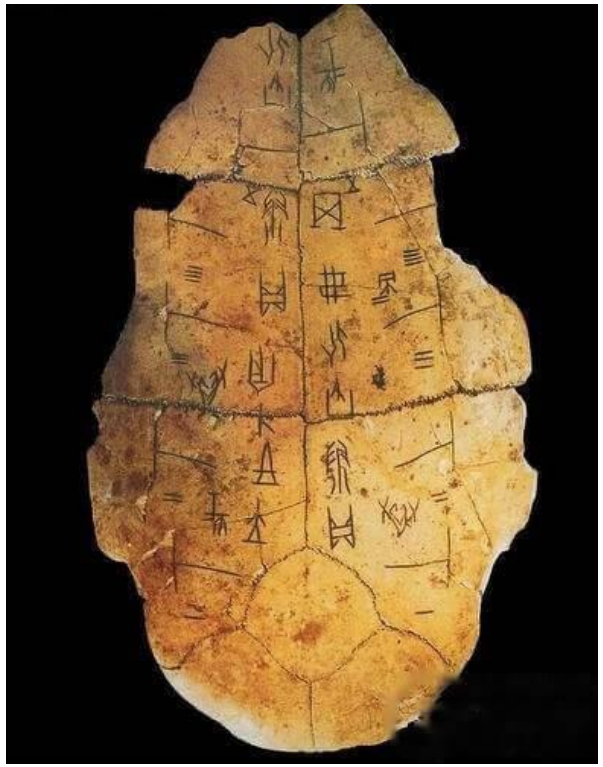
Pattern

Pattern is an entity that can be reproduced, e.g. fingerprint image, handwritten word, human face, speech signal, DNA sequence, etc.

02313427219321
49273840321438
42284262112373
02**4927**01932820
92382086121234
31193838298028
82764**4927**19203

Is it a pattern?



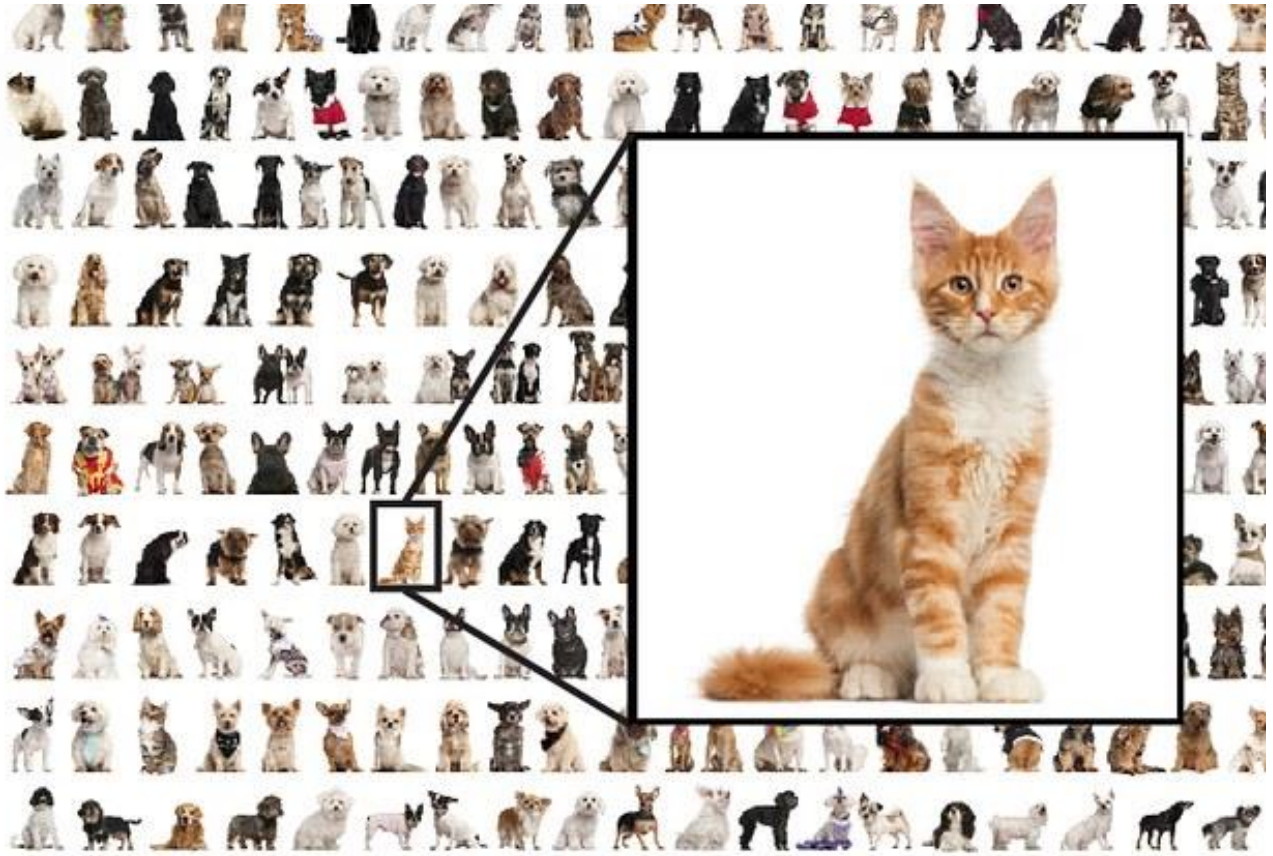


Pattern recognition

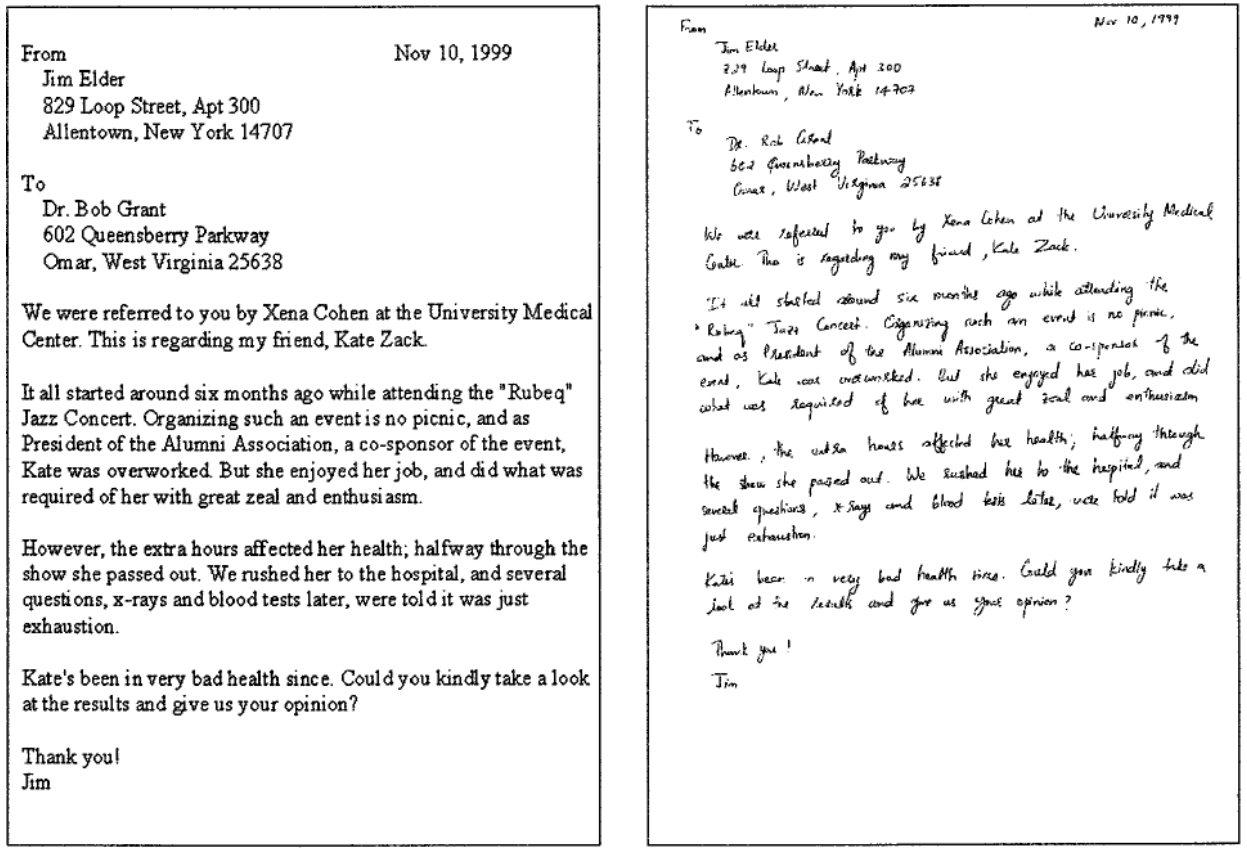
Pattern recognition is the way how machine can identify the pattern.

1. Collect data
2. Find the patterns of interest in data
3. Make reasonable decisions

Label the pattern: This is a cat!



Examples of pattern recognition



From
Jim Elder
829 Loop Street, Apt 300
Allentown, New York 14707

Nov 10, 1999

To
Dr. Bob Grant
602 Queensberry Parkway
Omar, West Virginia 25638

We were referred to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.

It all started around six months ago while attending the "Rubeq" Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm.

However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, x-rays and blood tests later, were told it was just exhaustion.

Kate's been in very bad health since. Could you kindly take a look at the results and give us your opinion?

Thank you!
Jim

English handwriting recognition

故天将降大任于是人也，必先苦其心志，劳其筋骨，饿其体肤，空乏其身，行拂乱其所为，所以动心忍性，曾益其所不能。

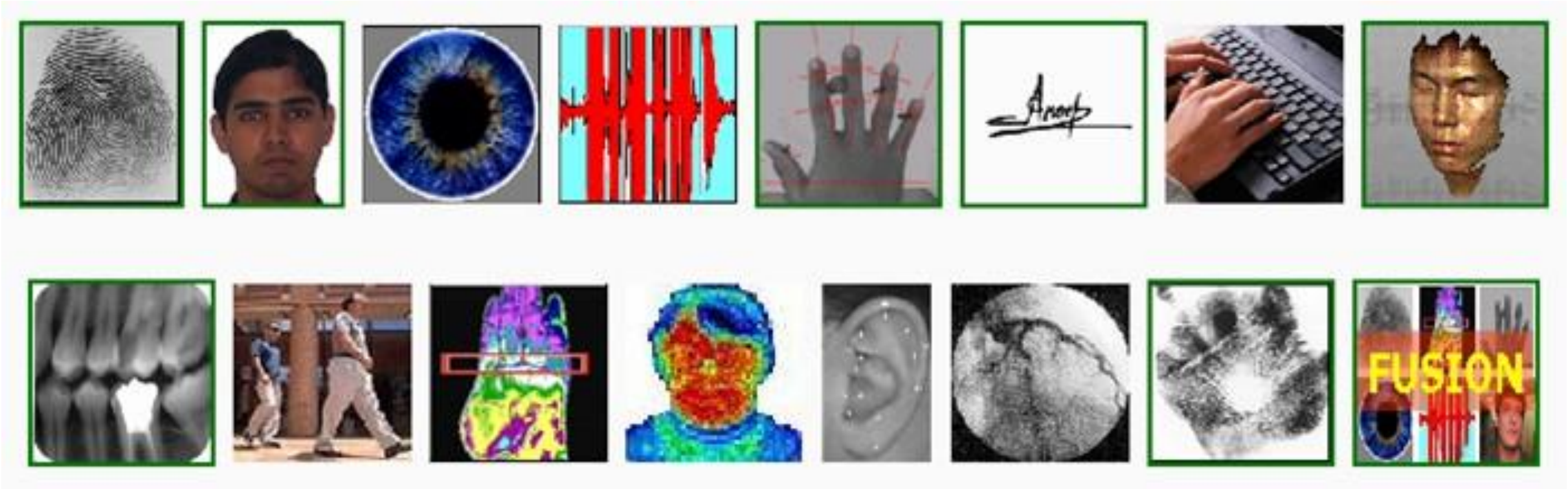
(a) Handwriting

故天将降大任于是人也，必先苦其心志，劳其筋骨，饿其体肤，空乏其身，行拂乱其所为，所以动心忍性，曾益其所不能。

(b) Corresponding Machine Print

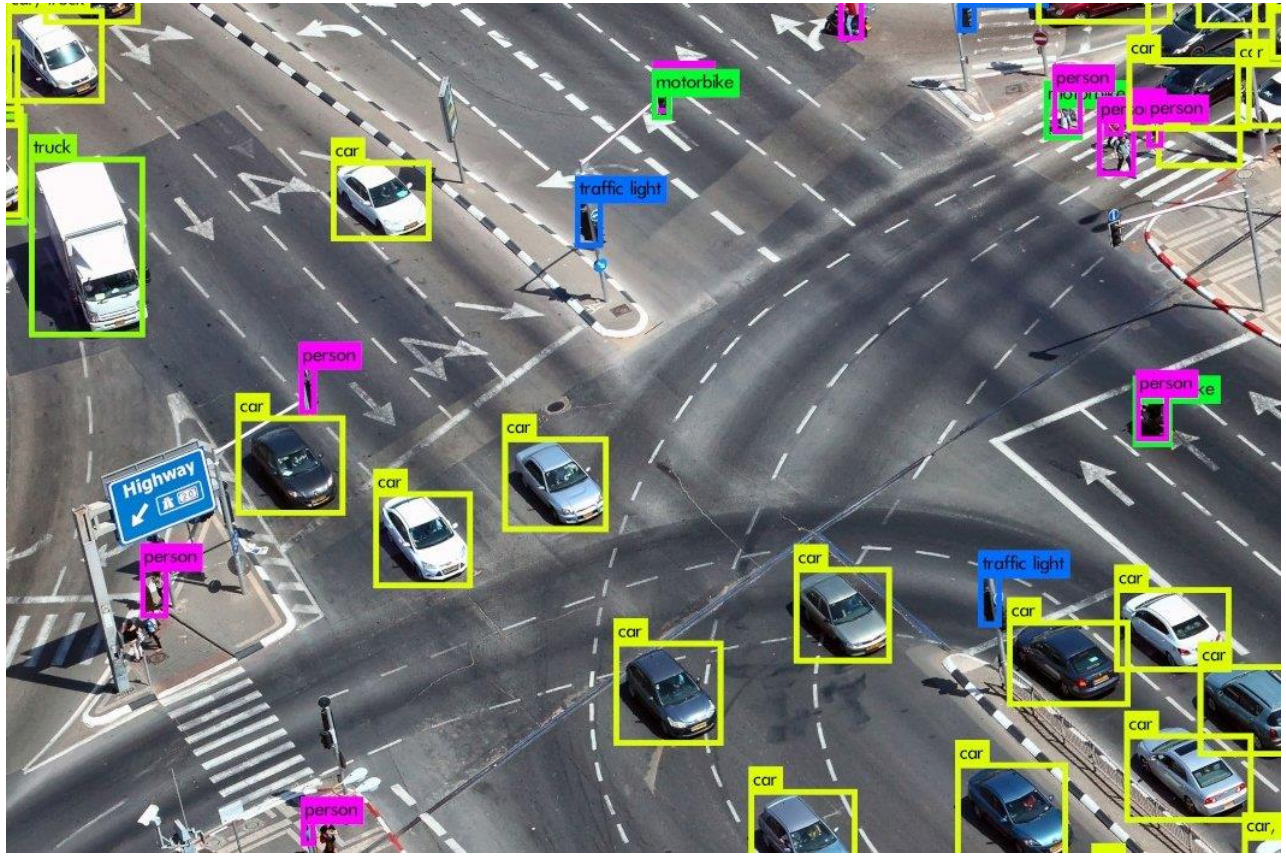
Chinese handwriting recognition

Examples of pattern recognition



Biometric recognition

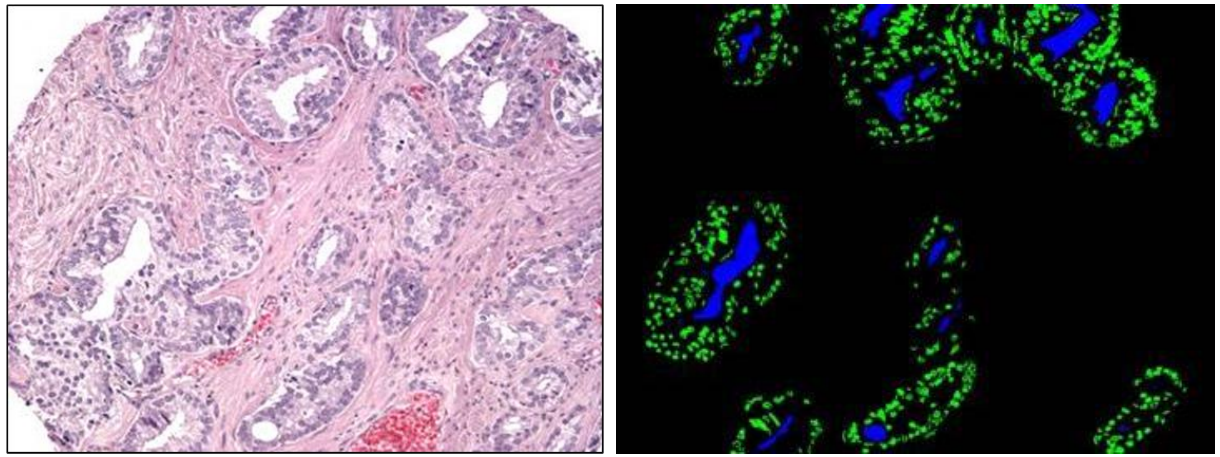
Examples of pattern recognition



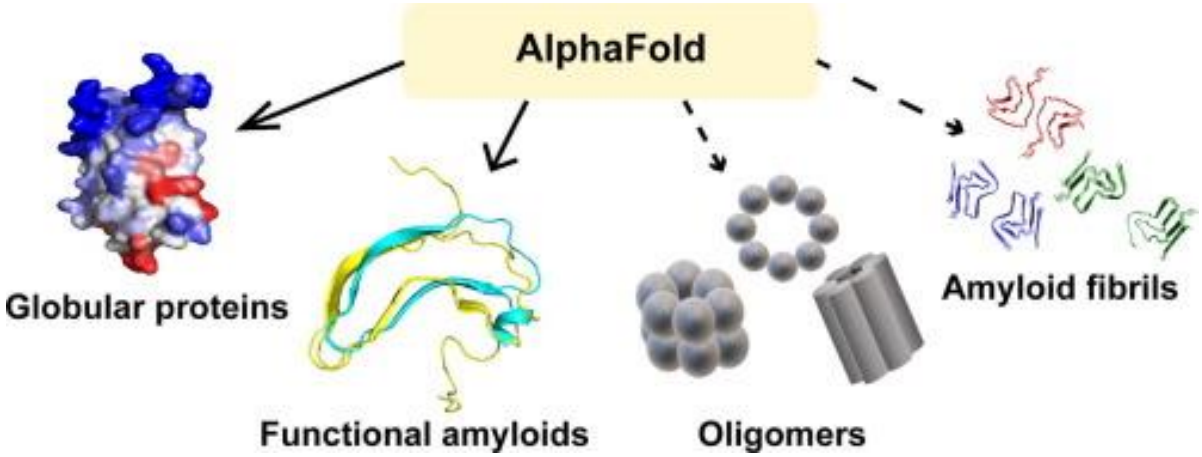
Transportation surveillance

License plate recognition

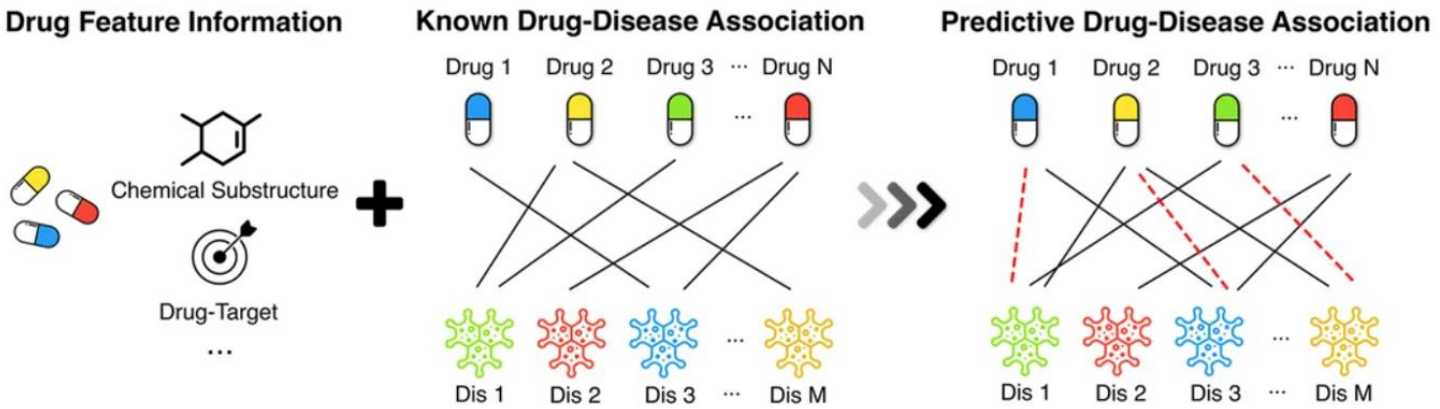
Examples of pattern recognition



Cancer detection using microscopic tissue data

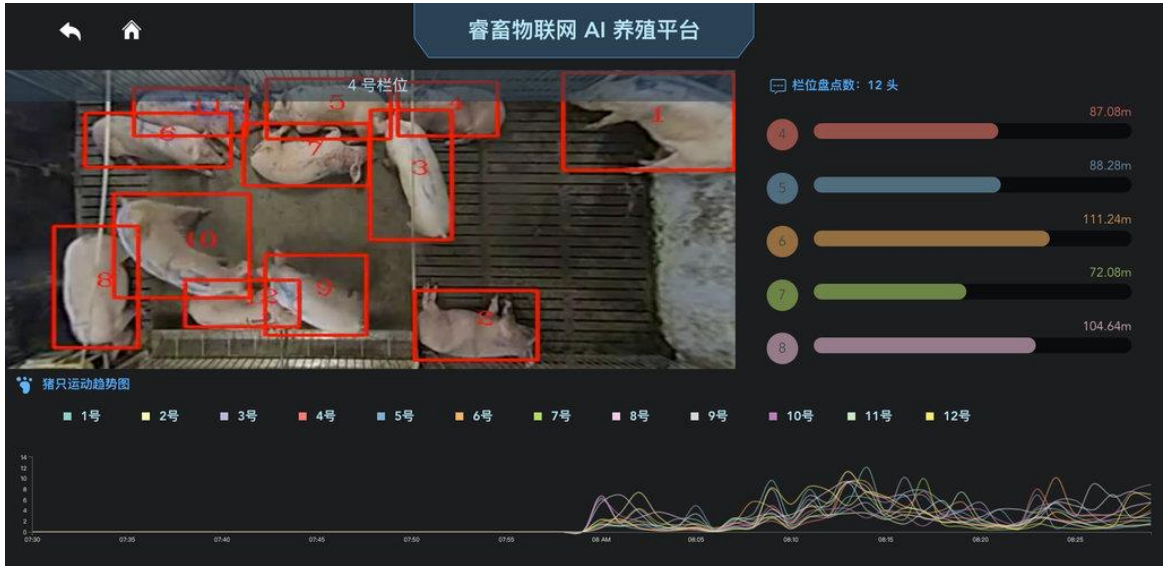


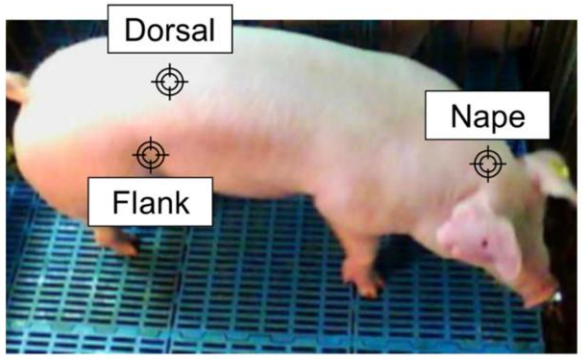
AlphaFold

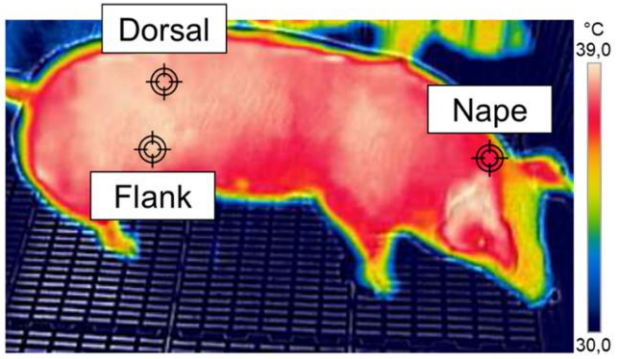


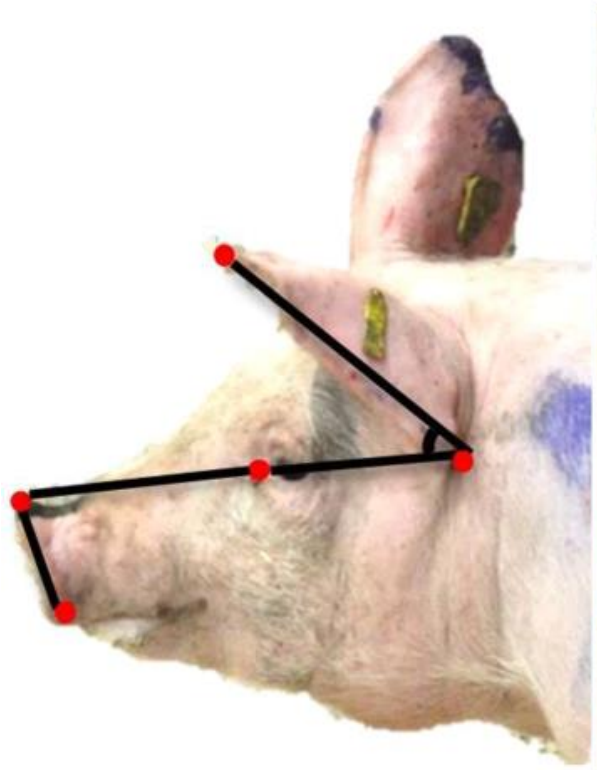
Drug-disease association prediction

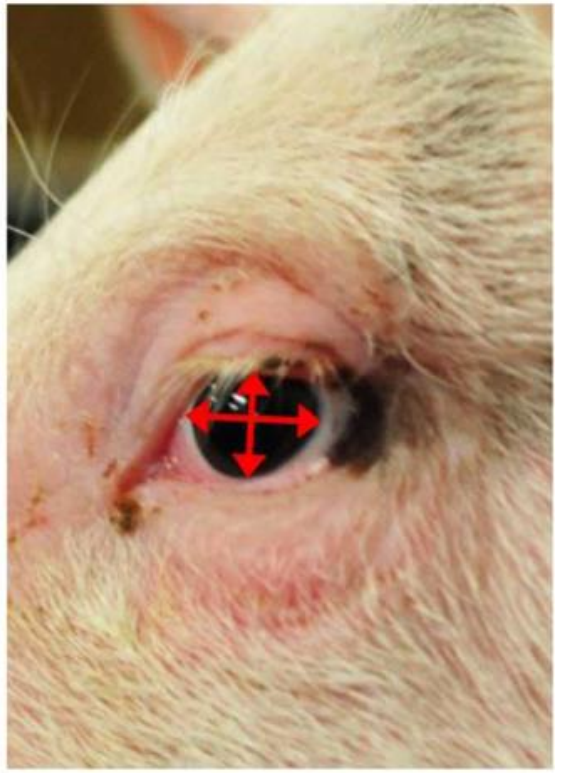
Examples of pattern recognition











AI farming (animal behavior analysis)

How to create machine perception?

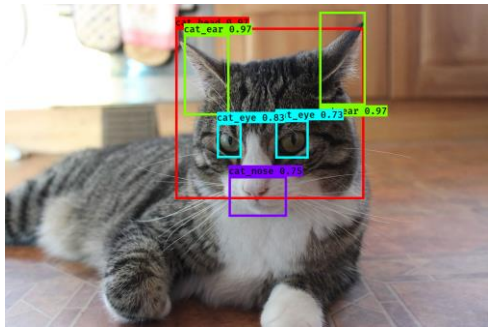
Mimic the way that human perceive the world

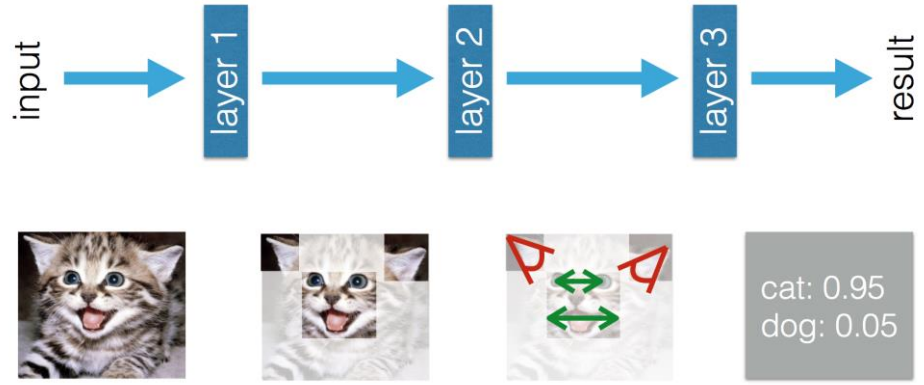
Purely numerical and no correspondence in nature

How to create machine perception

Mimic the way that human perceive the world

Purely numerical and no correspondence in nature





AI algorithms & programming

AI的核心是算法 (algorithm)

AI algorithm is more focused on modeling (not deterministic)

Algorithm creates a workflow that organizes different parts for an objective

Numerical operators, arrays, decision-making (if-else), looping (for), etc.

编程实现算法 (programming)

C/C++, Java, Python, Matlab, etc.

AI algorithms & programming

AI的核心是算法 (algorithm)

编程实现算法 (programming)

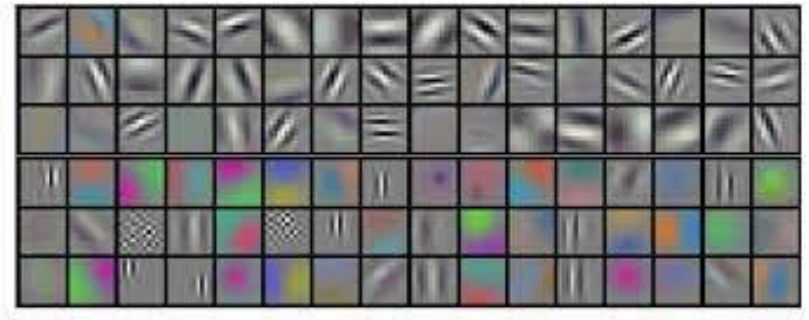
别人做的

你做的

Pattern recognition algorithms

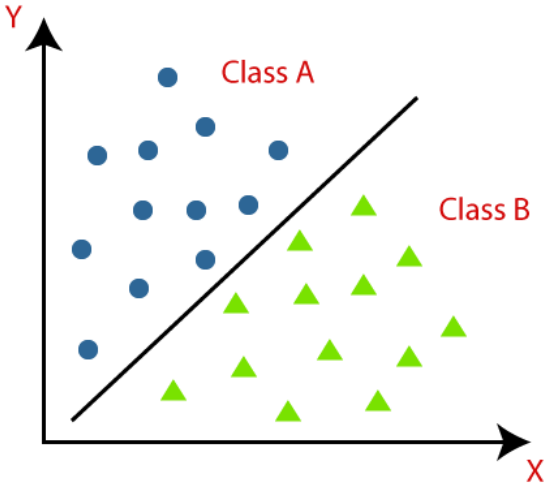
Data（数据）

Features（特征）



Colour, edge, texture, gradients, etc.

Model（模型）

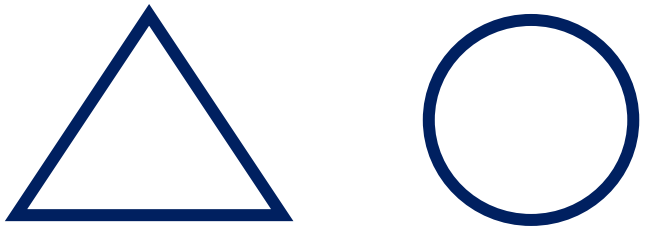


Pattern recognition algorithms

Unsupervised algorithms

Clustering problem (聚类问题）

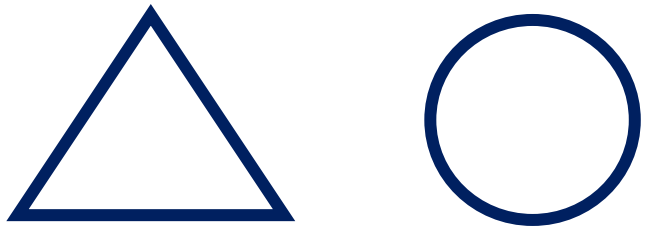
“These are two different shapes!”



Supervised algorithms

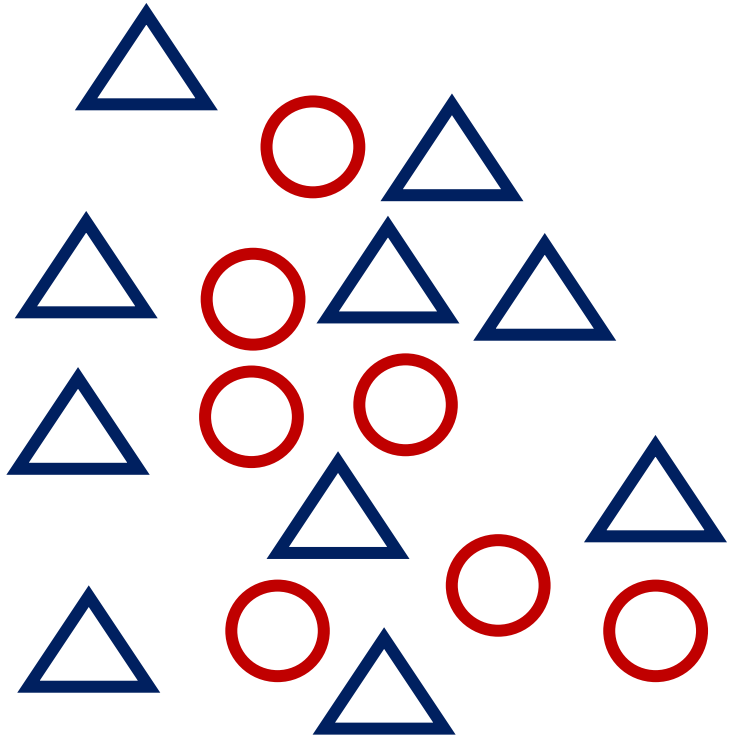
Classification problem (分类问题）

“Left is triangle, right is circle!”



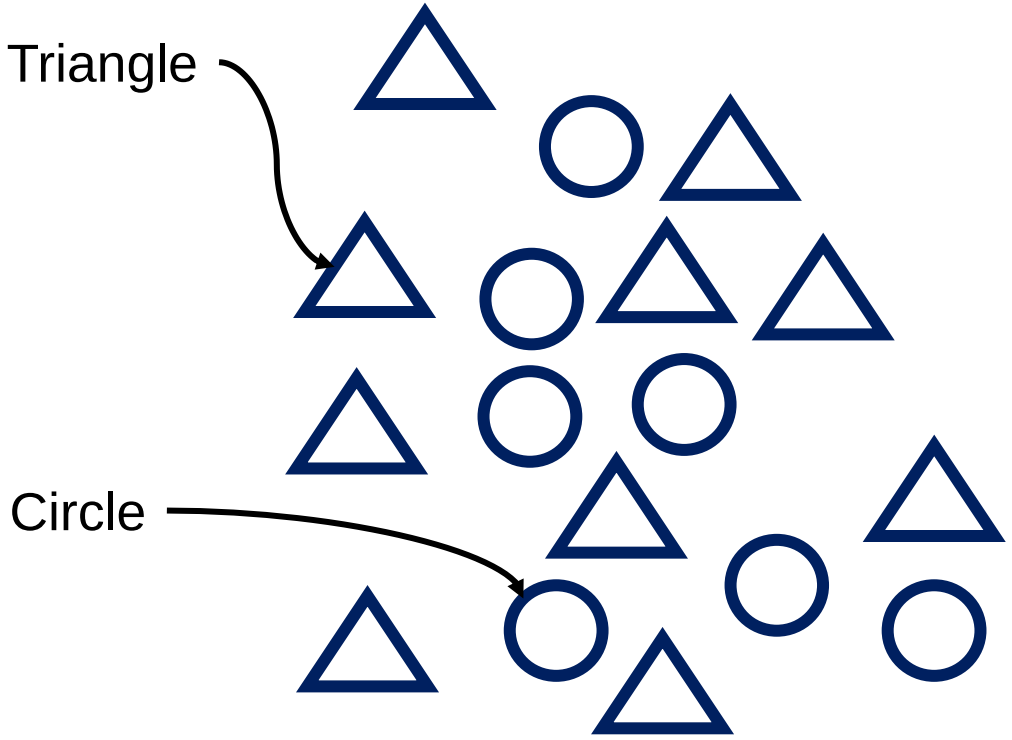
Pattern recognition algorithms

Clustering



There are two clusters, labelled as red and blue.

Classification



There are two clusters, labelled as triangle and circle.

Case study: sorting fish

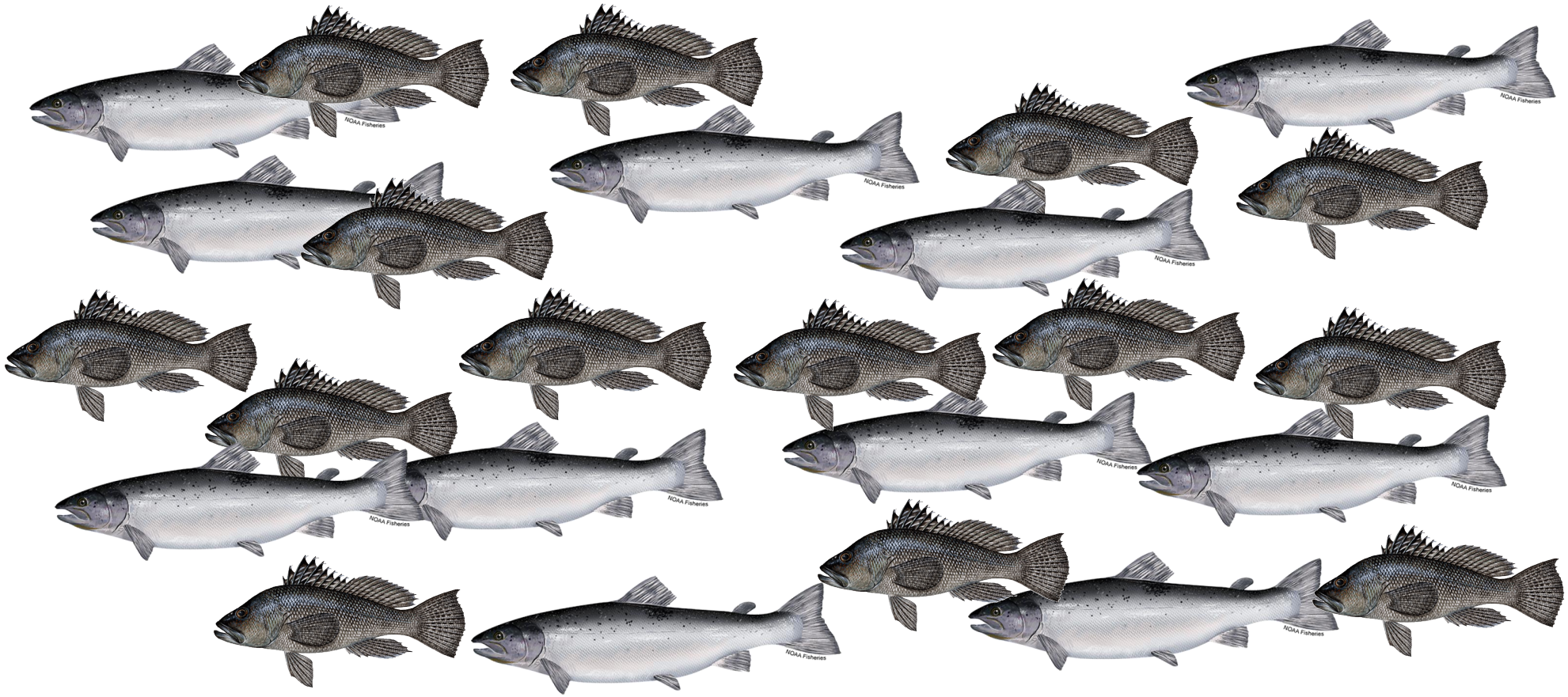
Case: Sorting the incoming fishes according to species: seabass, salmon.

Seabass (海鲈鱼)

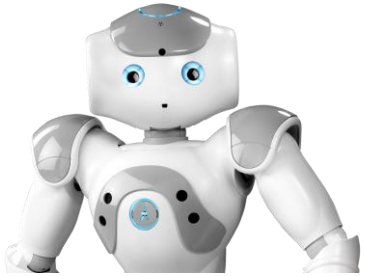
Salmon (三文鱼)

Case study: sorting fish

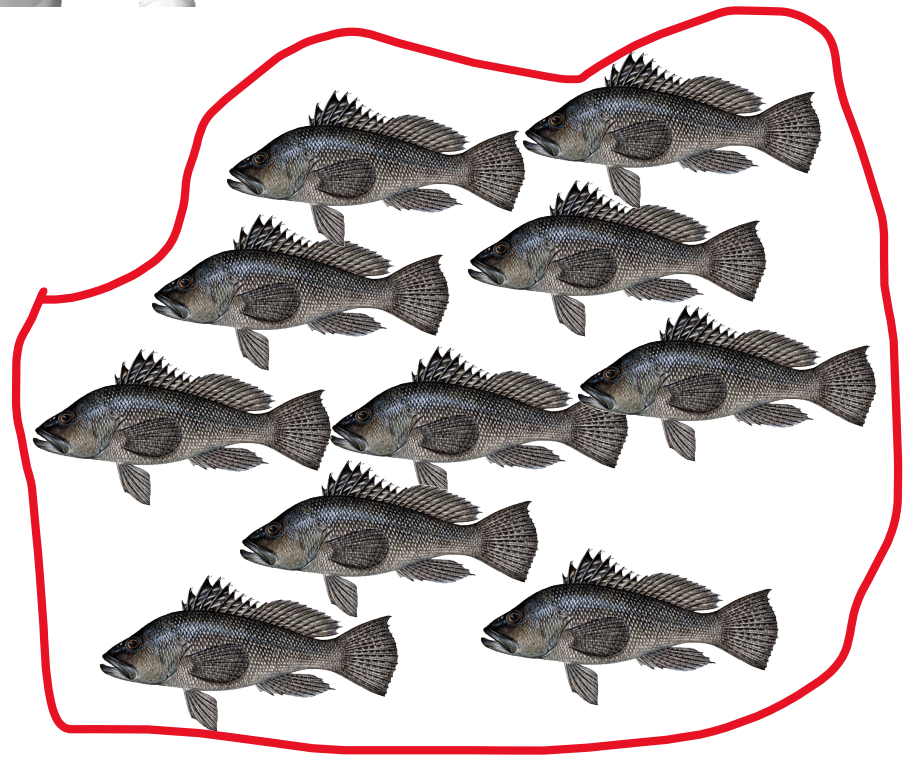
If we caught 100 seabass and 100 salmon…

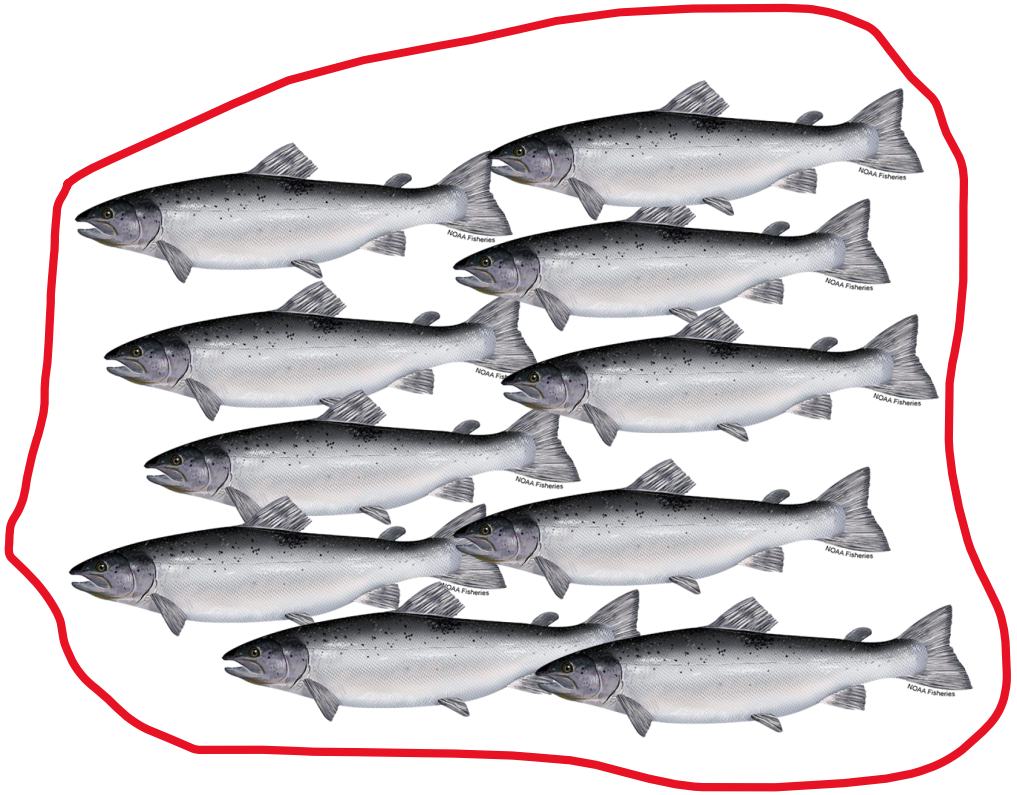


Case study: sorting fish



We need a robot to separate two types of fish!





Case study: sorting fish

How can we visually distinguish seabass and salmon?

Sea bass (海鲈鱼)

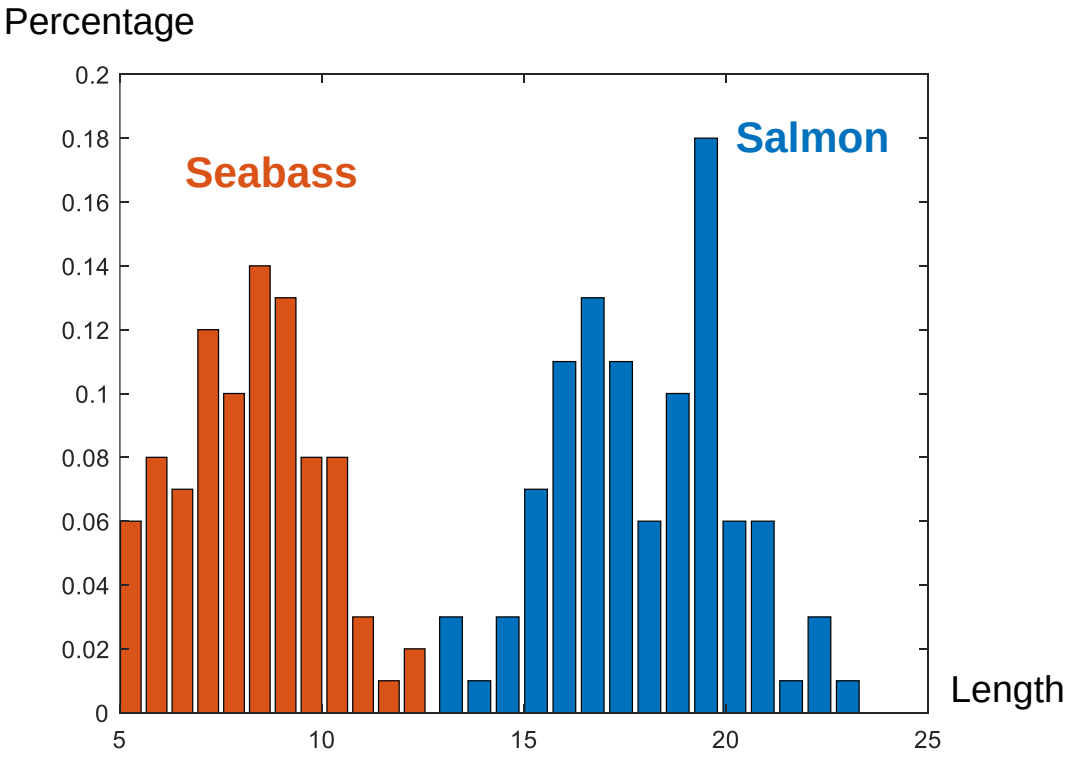
- **Length**: salmon is longer than seabass

Salmon (三文鱼)

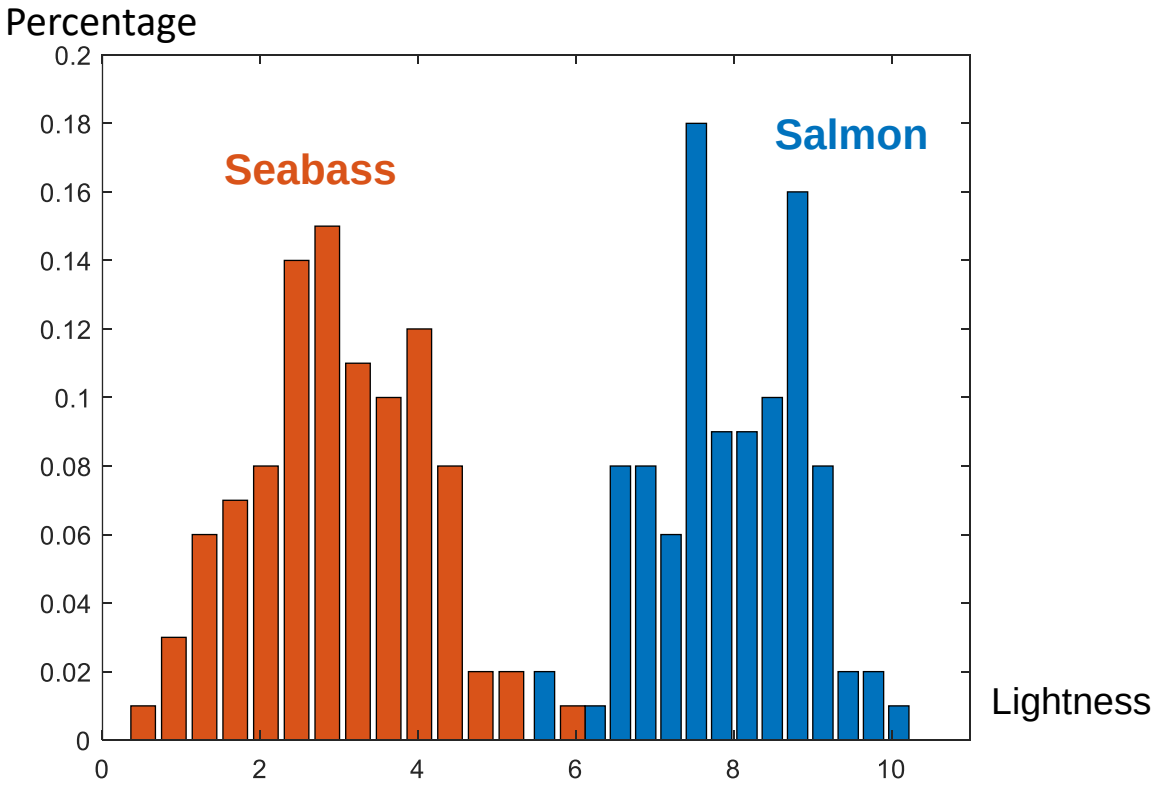
- **Color**: salmon is brighter than seabass

Distribution of fish features

In our nature, many patterns follow Gaussian/normal distribution!



"Salmon is longer than seabass"



"Salmon is brighter than seabass"

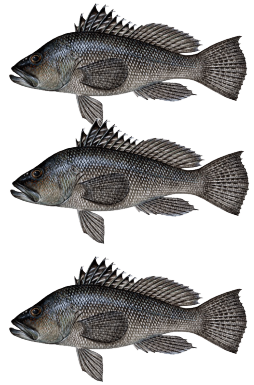
Distribution of fish features

If we use 2 features (length and color) to represent a fish, we can create a feature vector (1D array) for a fish:

$$\text{fish} = [\text{length}, \text{color}];$$

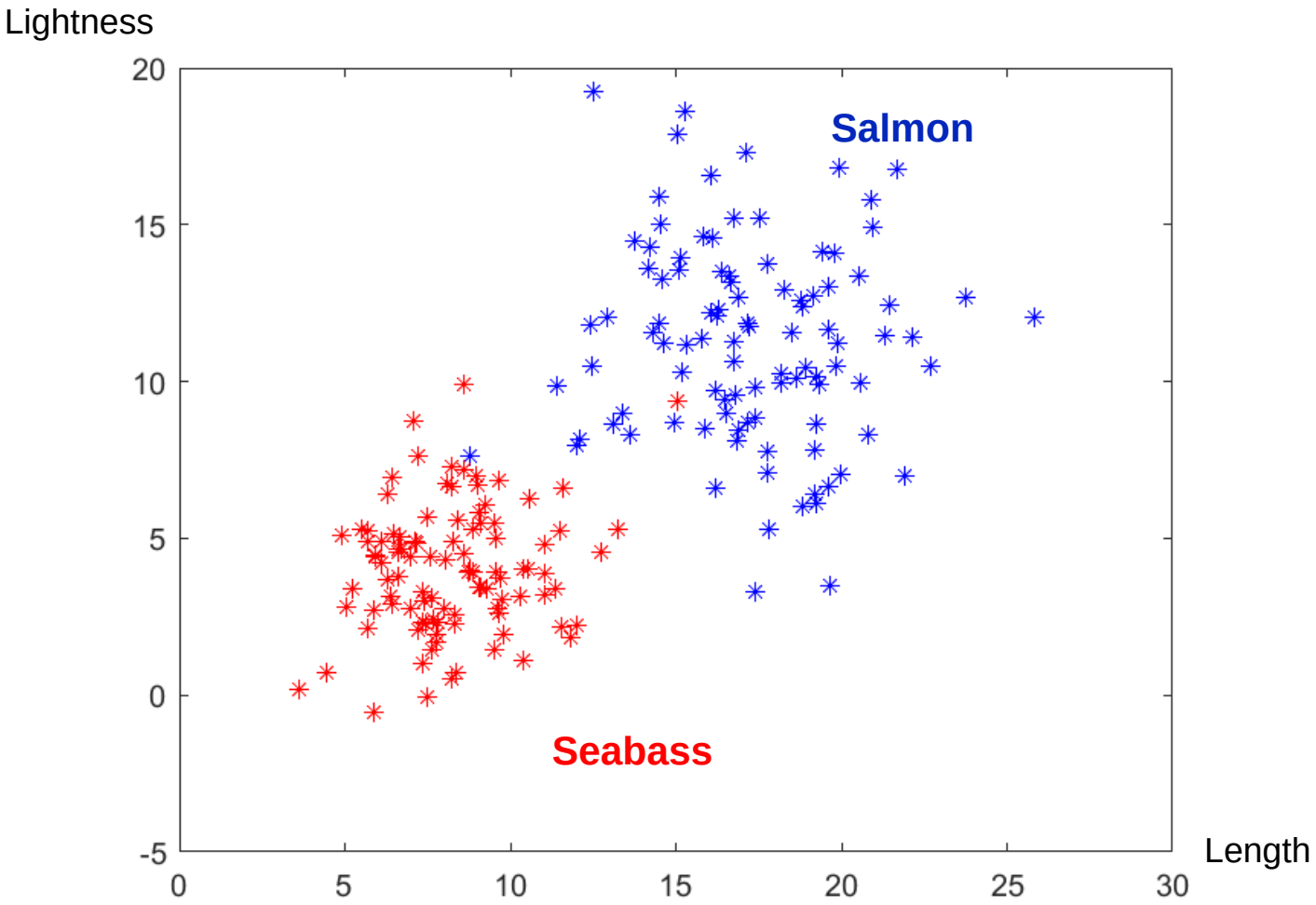
If we have multiple fishes, we can formulate a 2D feature matrix:

$$\text{fishes} = \begin{bmatrix} \text{length}_1 & \text{color}_1 \\ \text{length}_2 & \text{color}_2 \\ \cdots & \cdots \\ \text{length}_n & \text{color}_n \end{bmatrix};$$



Distribution of fish features

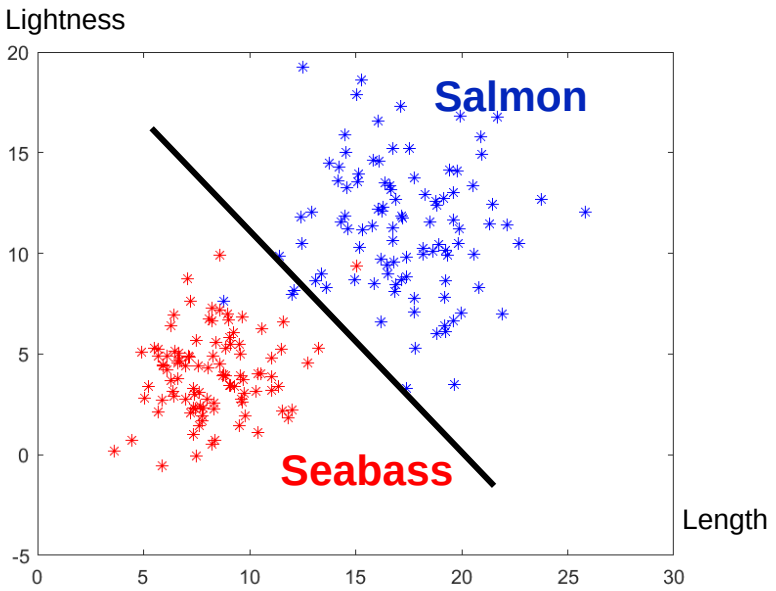
Distribution of 2D features on a plane



How to separate two types of fish?

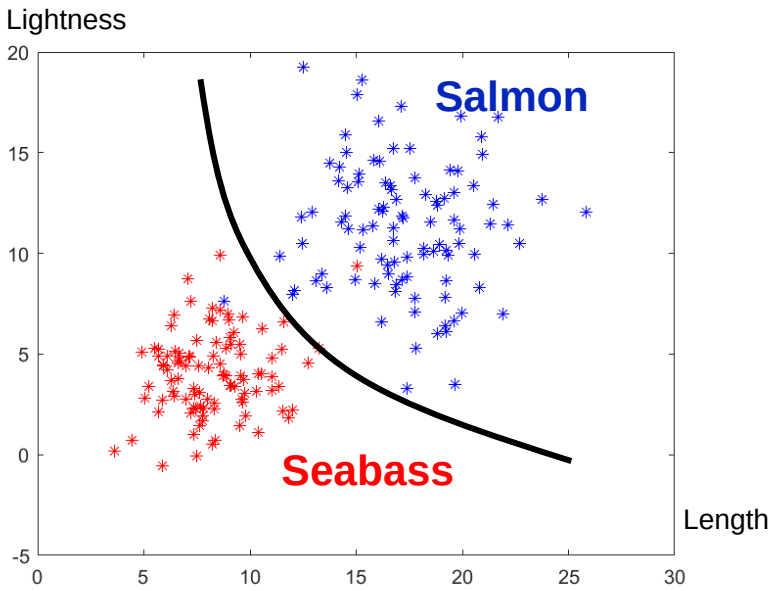
Draw a decision boundary!

How to draw the decision boundary?



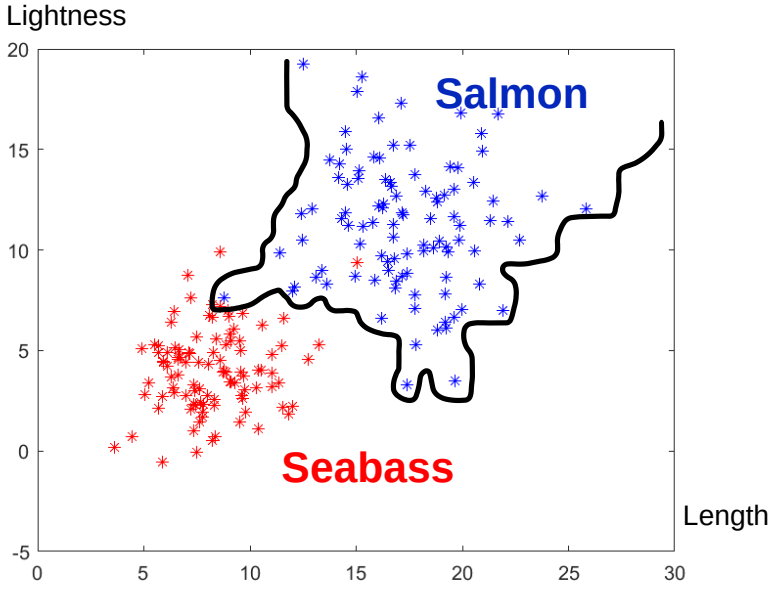
$$Y = a \cdot X + b$$

If Y > 0, seabass
If Y < 0, salmon



$$Y = a^2 \cdot x + b \cdot x + c$$

If Y > 0, seabass
If Y < 0, salmon

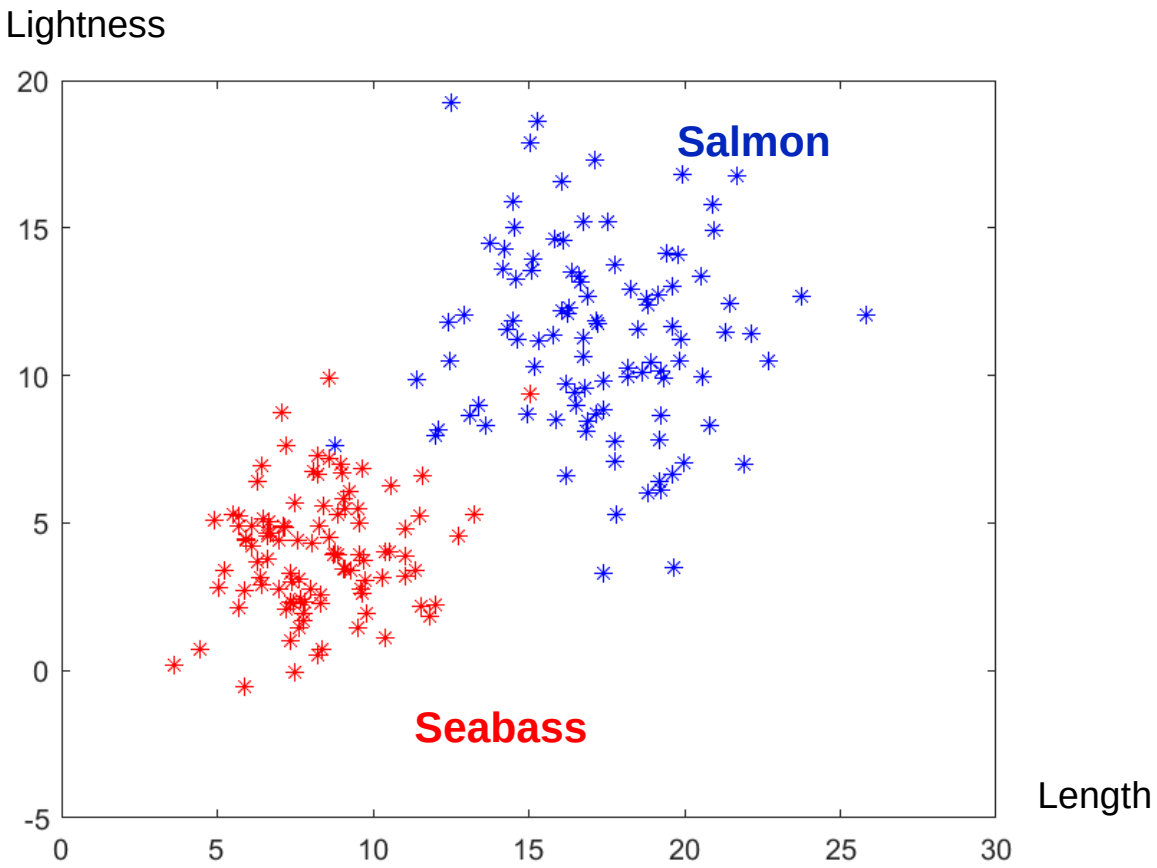


Highly non-linear!!!

If Y > 0, seabass
If Y < 0, salmon

Underfitting and overfitting!!!

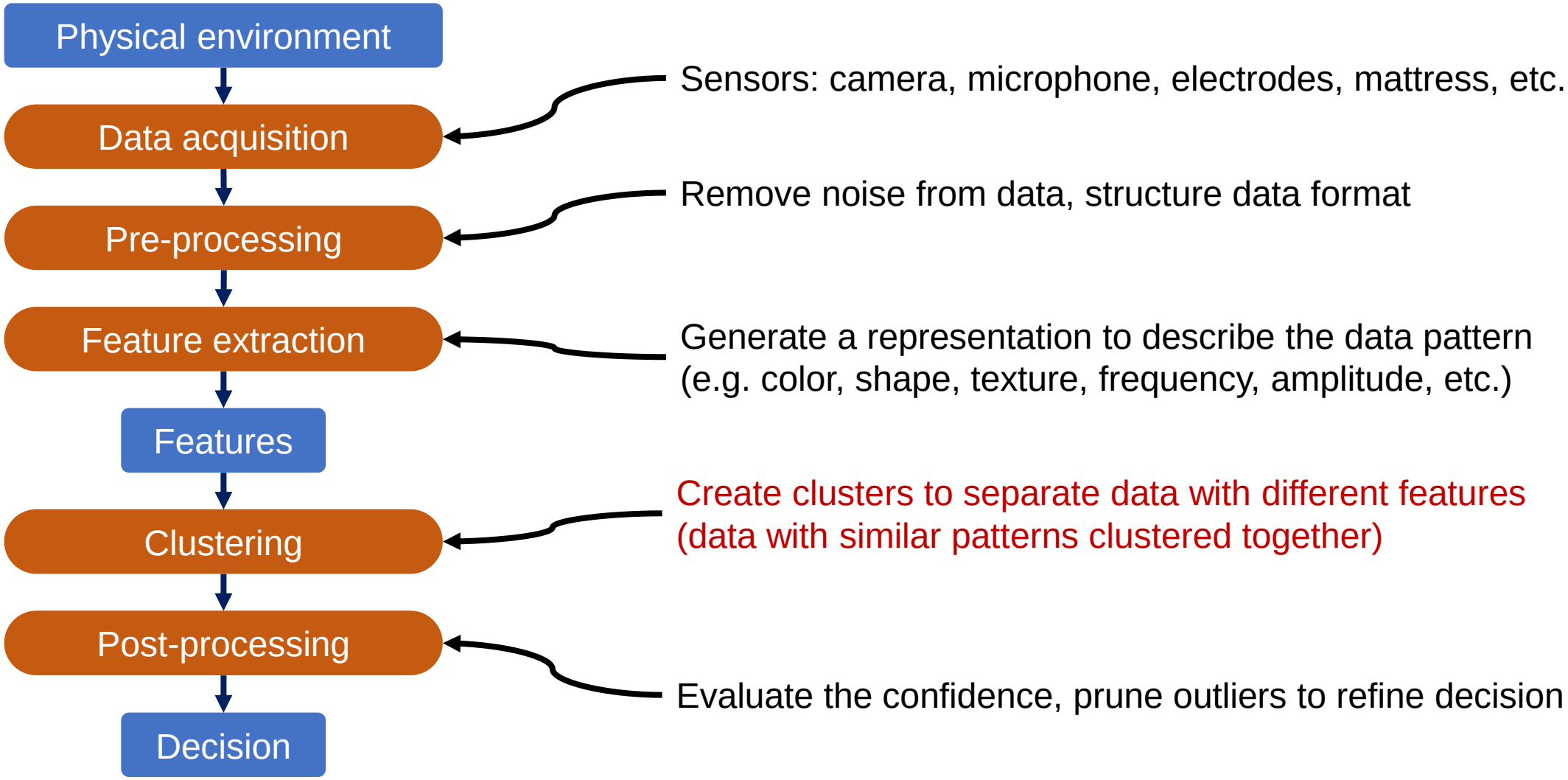
How to separate two types of fish?



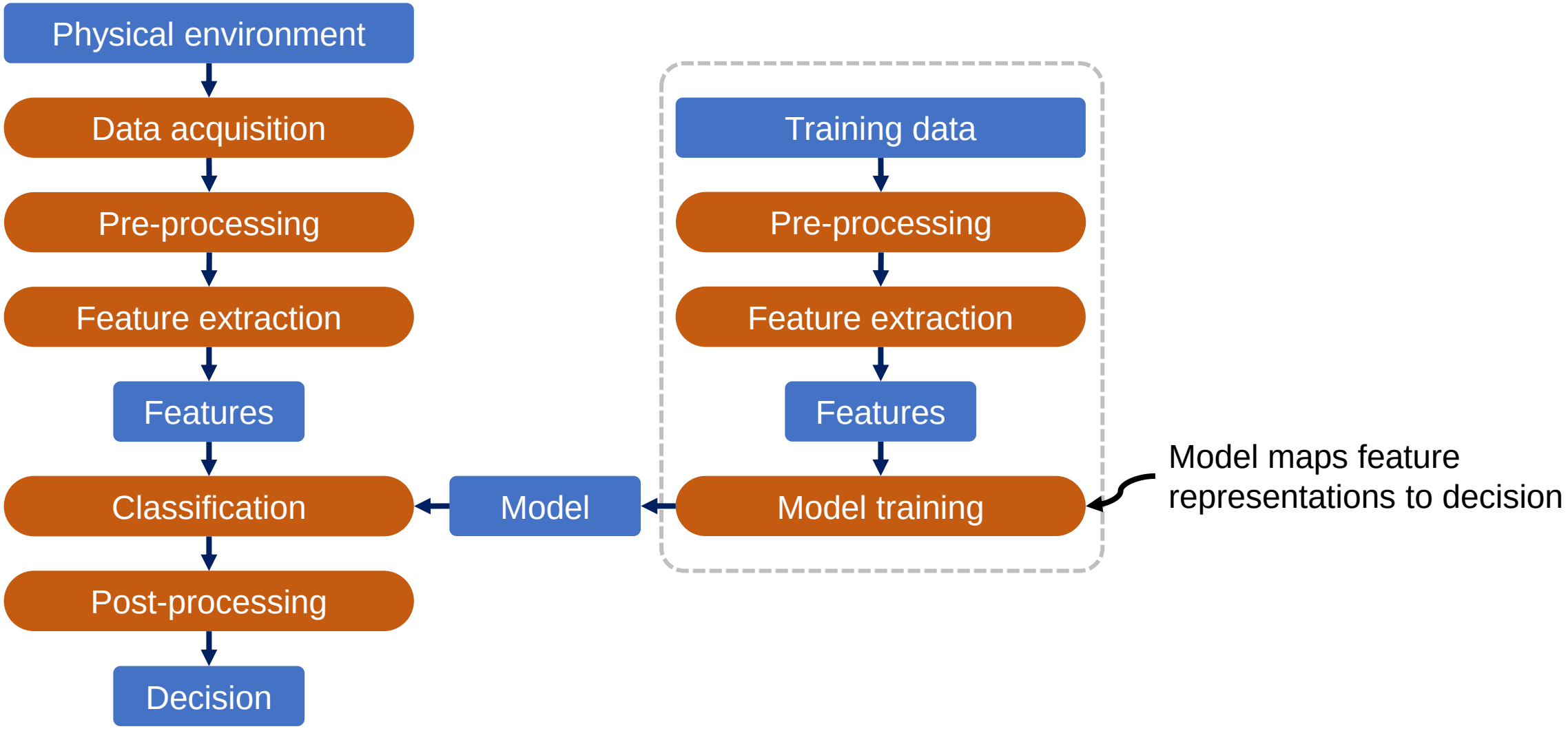
Clustering problem
(聚类问题）
“These are two different fishes!”

Classification problem
(分类问题）
“This is seabass, this is salmon!”

Unsupervised learning system



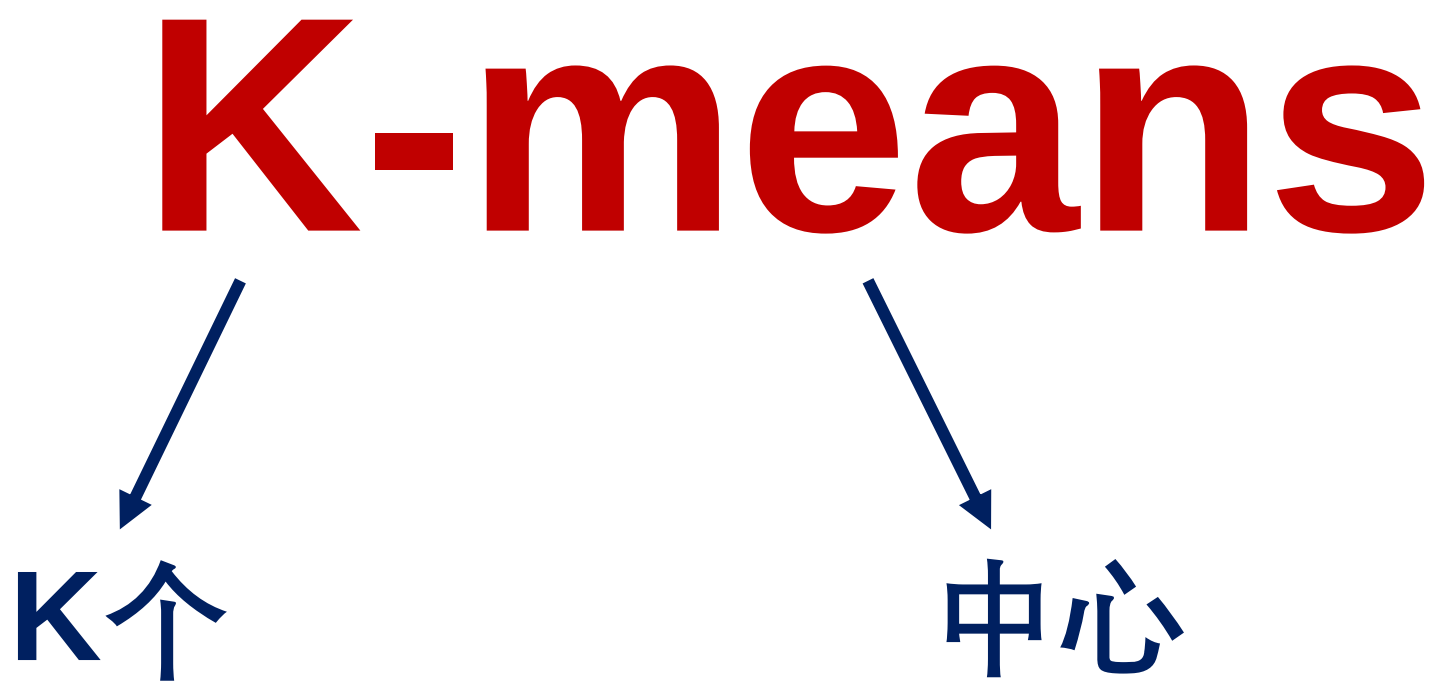
Unsupervised learning system



Content

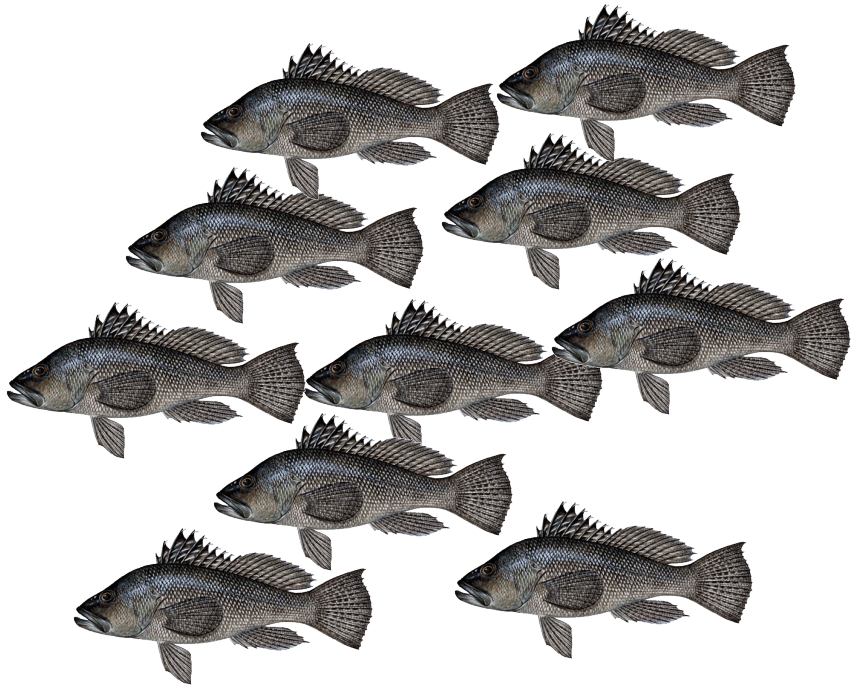
1. Introduction to pattern recognition (AI)
2. **Clustering (Kmeans & DBscan)**
3. Classification (KNN & perception)

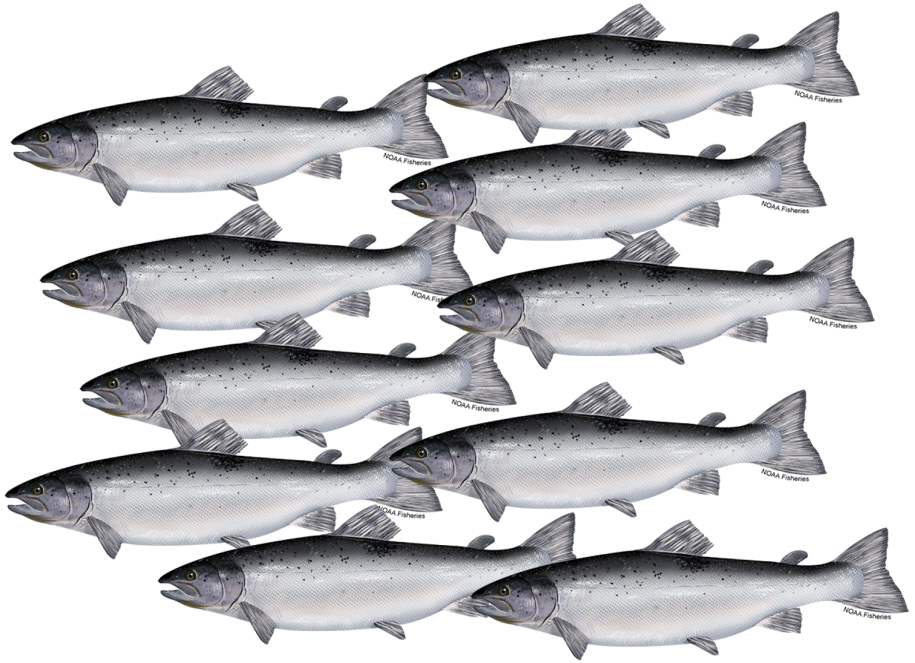
What is Kmeans?



What is Kmeans?

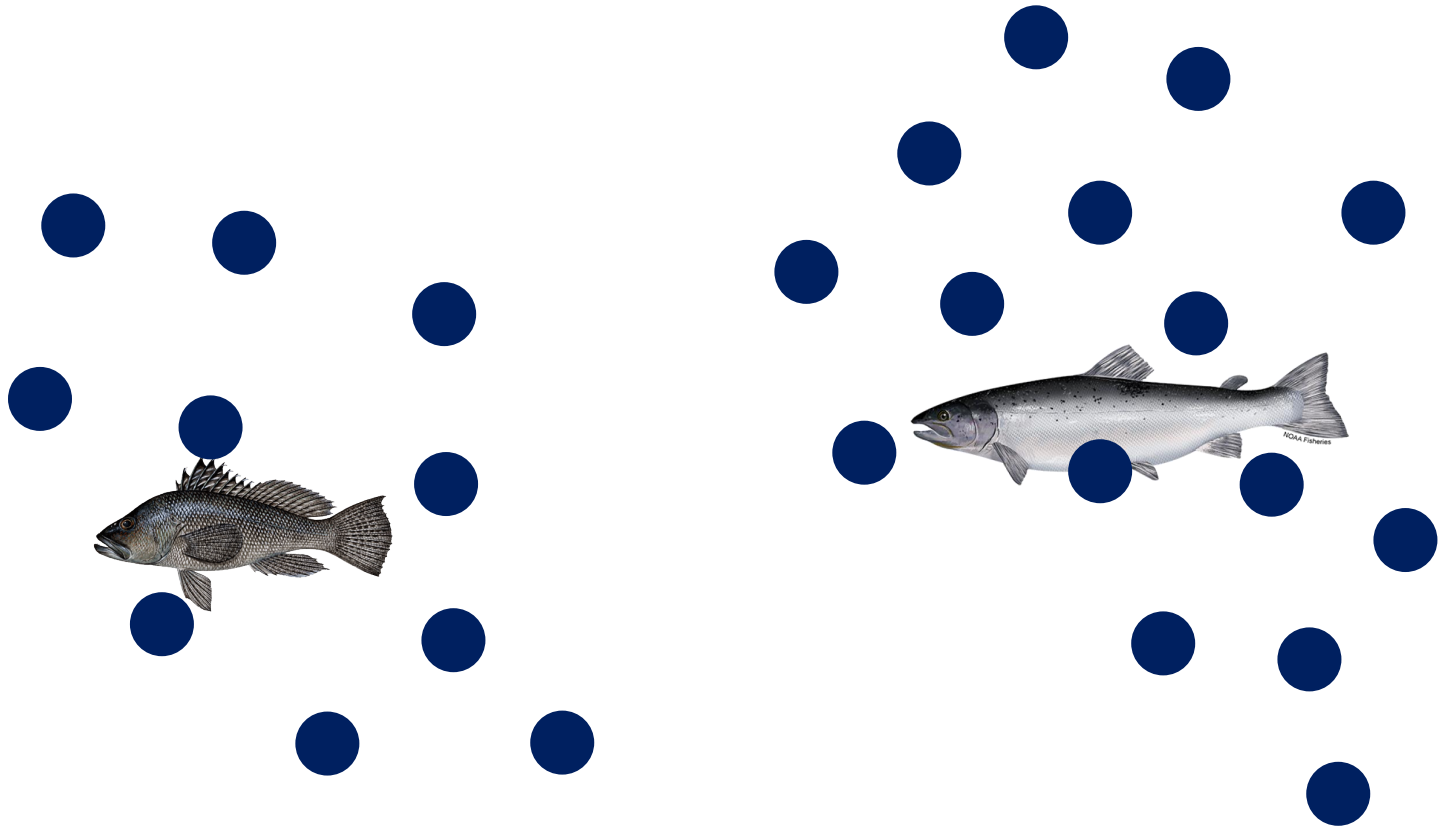
Take the fish sorting example, we want to cluster salmon and seabass.





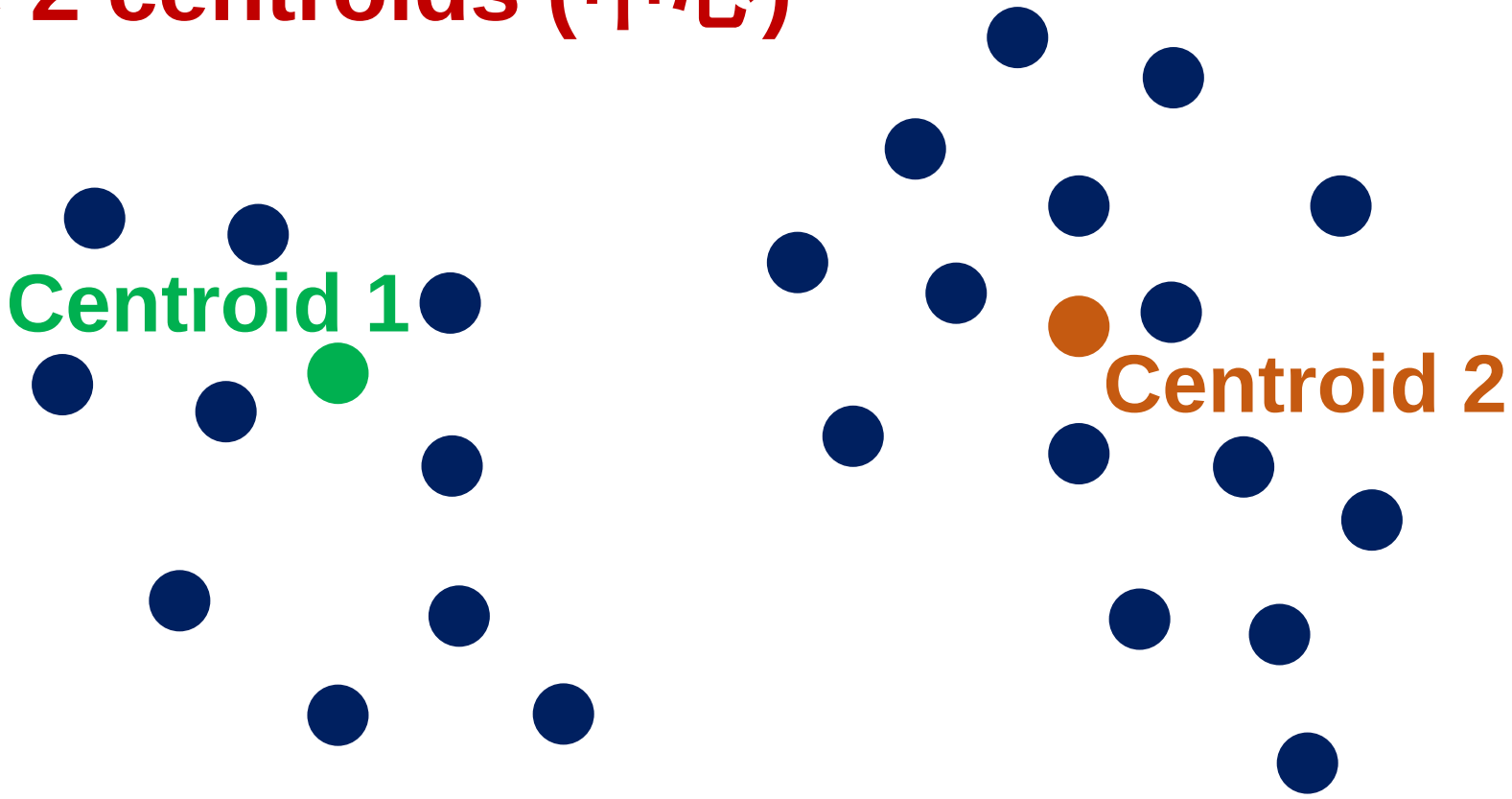
What is Kmeans?

Use 2D points to represent fishes in the feature space.



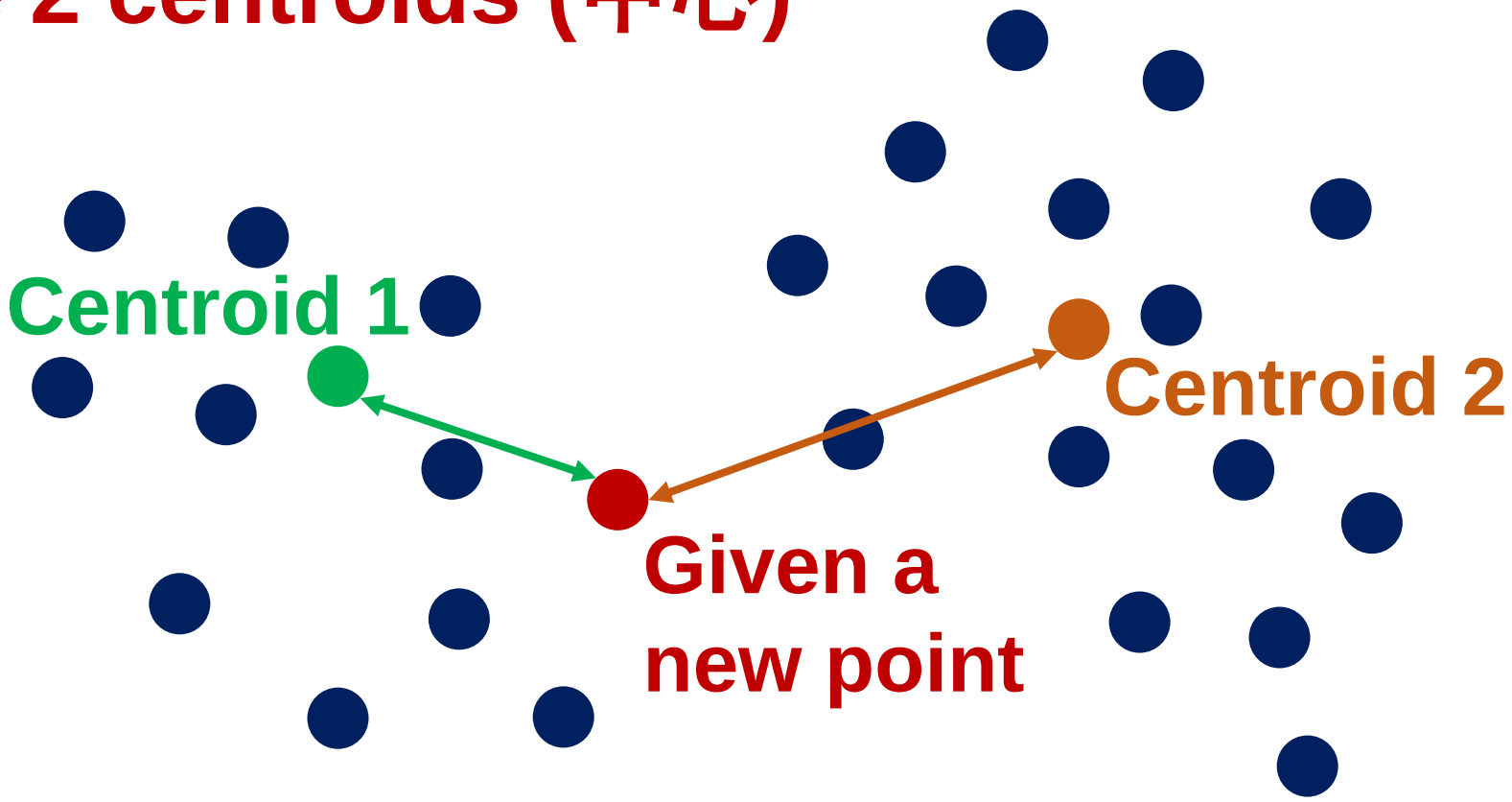
What is Kmeans?

We define 2 centroids (中心)



What is Kmeans?

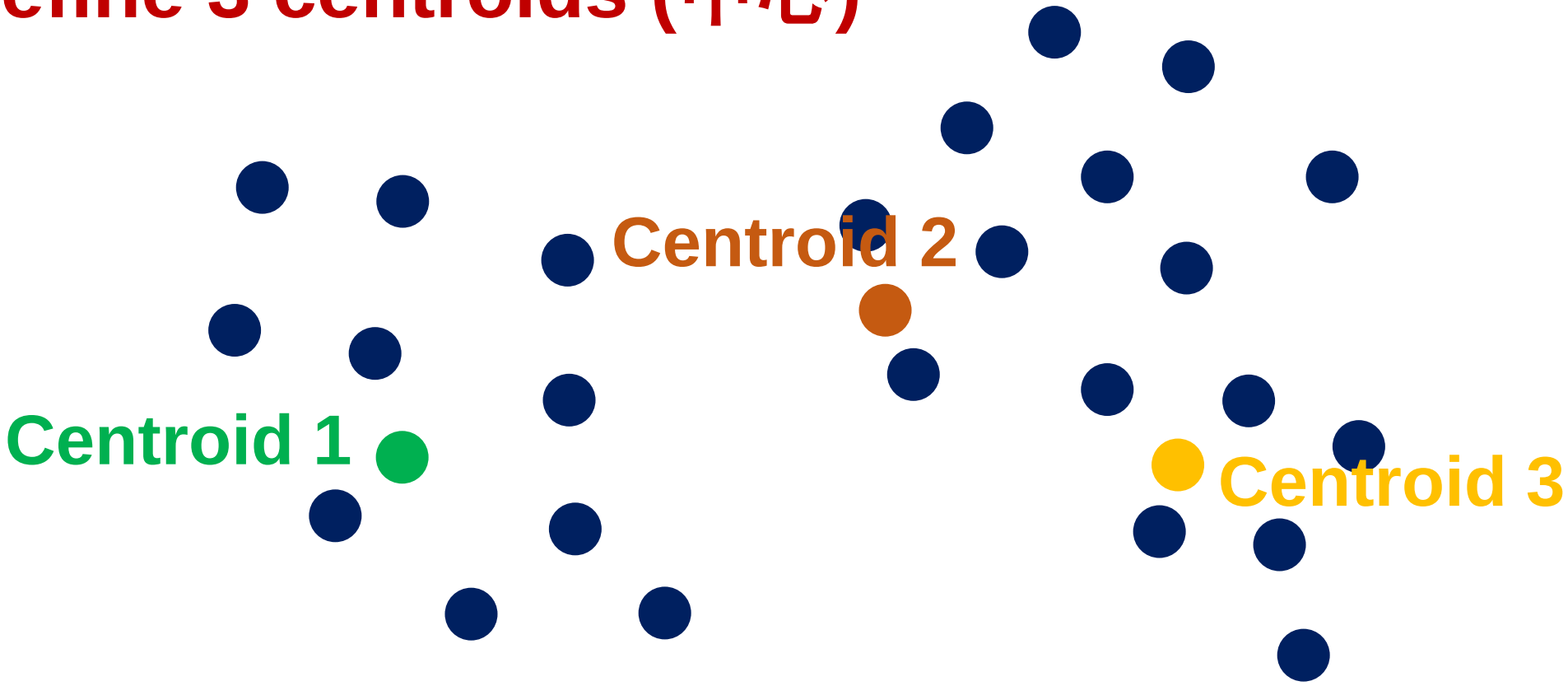
We define 2 centroids (中心)



The point is closer to center 1, so it belongs to cluster 1.

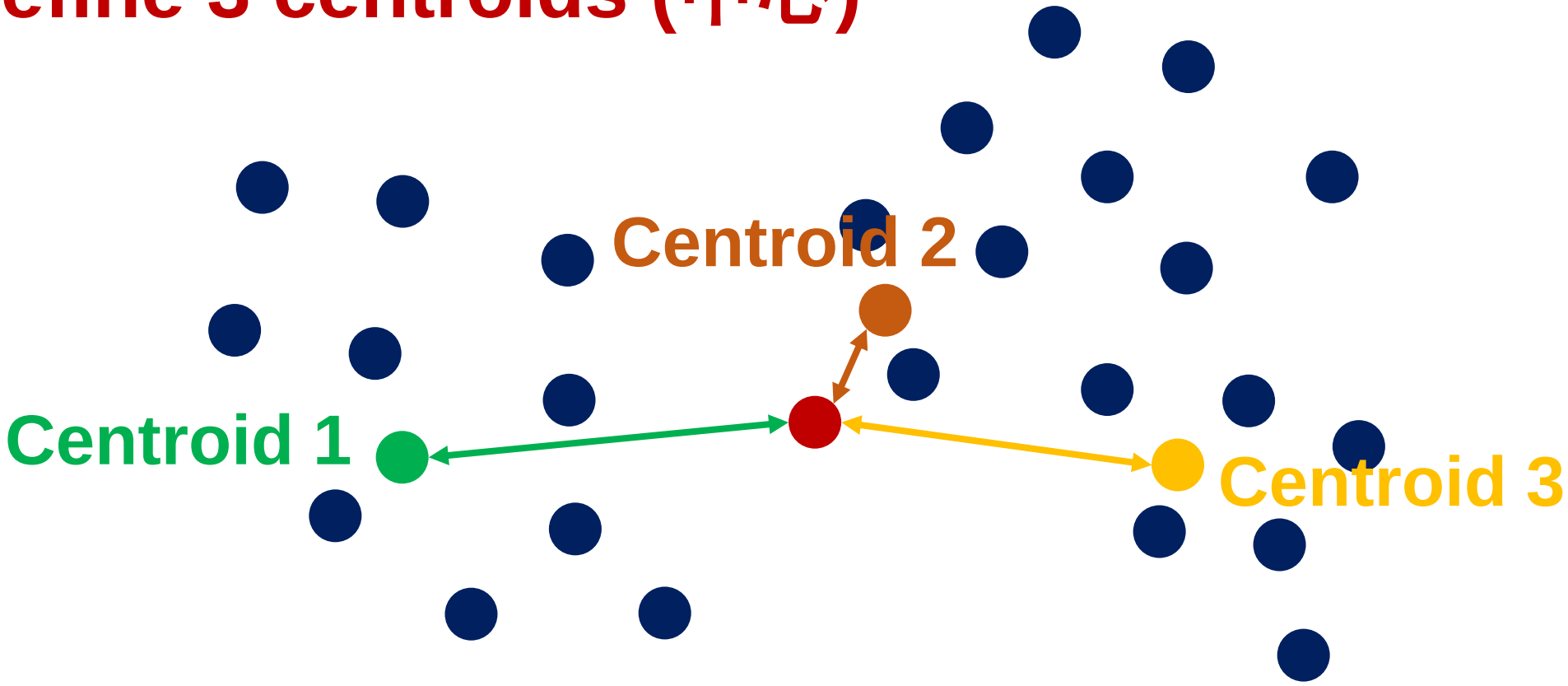
What is Kmeans?

We define 3 centroids (中心)



What is Kmeans?

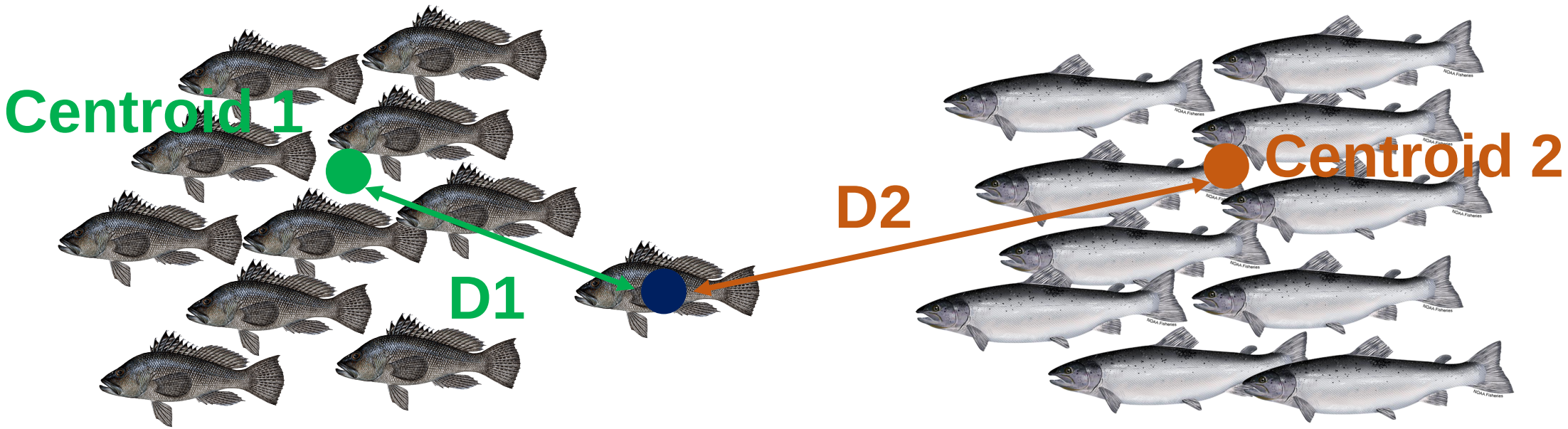
We define 3 centroids (中心)



The point is closer to centroid 2, so it belongs to cluster 2.

What is Kmeans?

In our fish sorting case, define 2 centroids (seabass and salmon)



D1 < D2, this is seabass!

Now, how to define the centroids?

Kmeans definition

Kmeans is an iterative algorithm that automatically finds K clusters in the feature space.

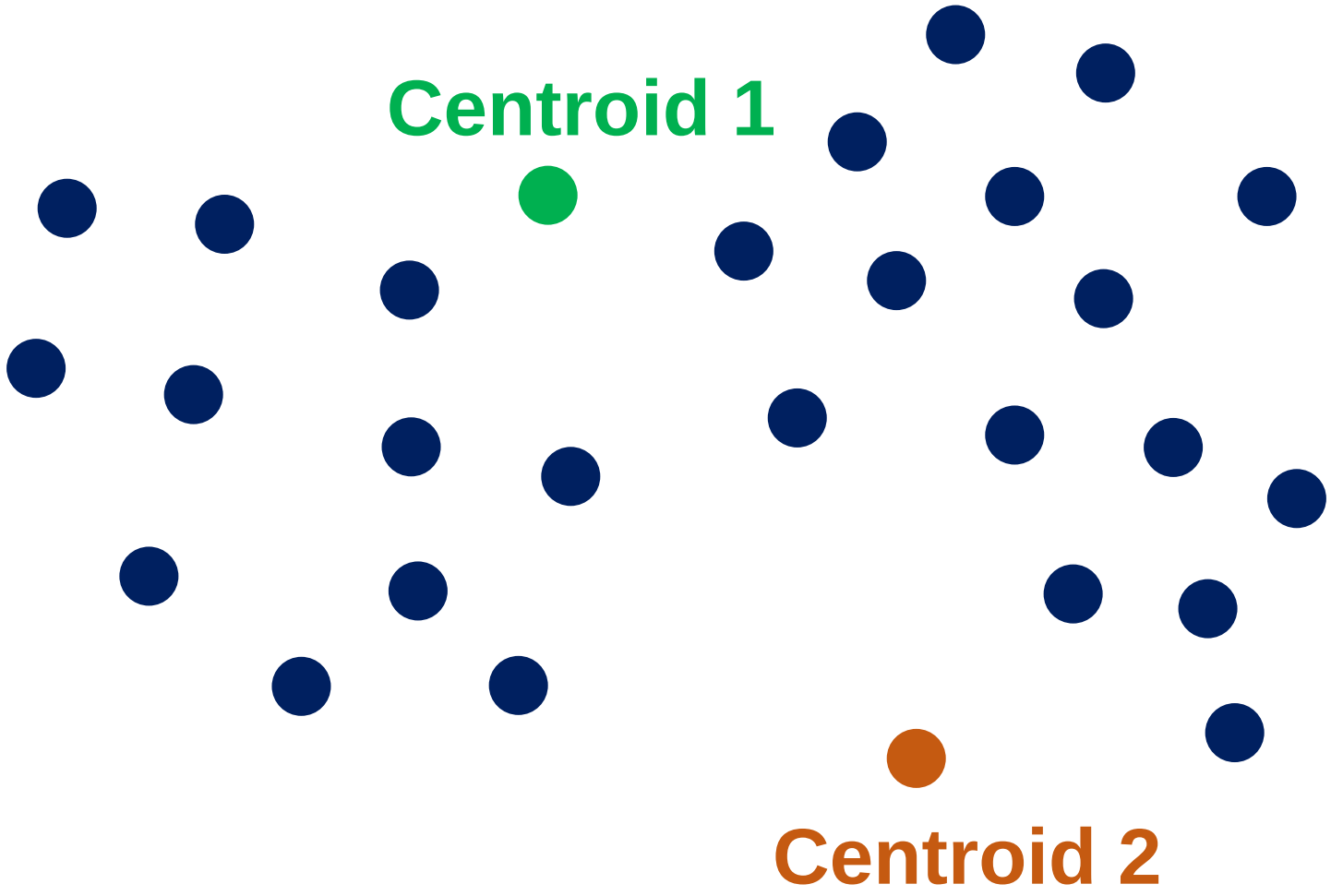
Initialize: randomly define k centers in space
while (true)
{

- assign data points to the closest center
- update the cluster center by averaging the points assigned to the cluster
- if (center locations have no change) {break;}

}
Output: k centers

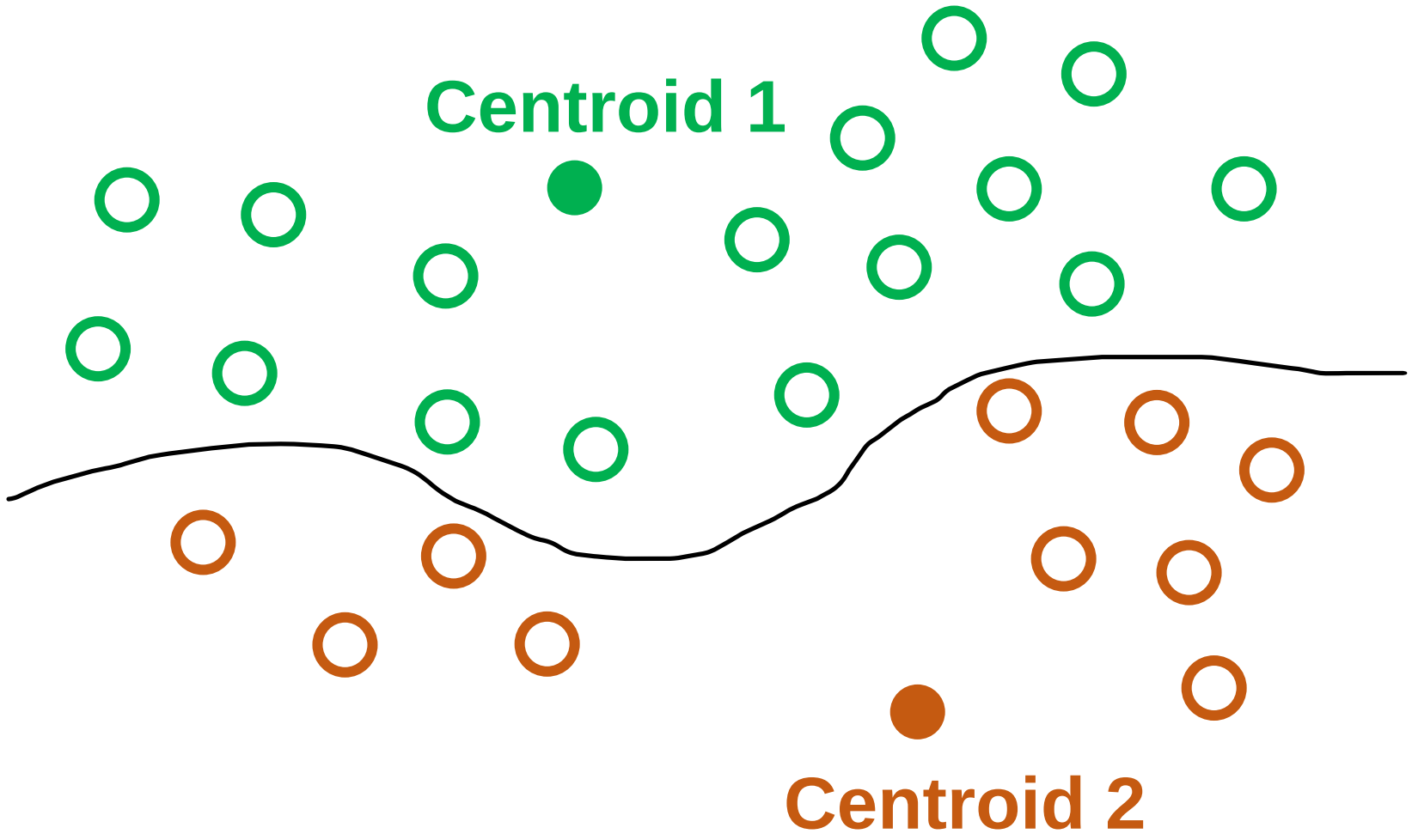
Kmeans step-by-step

Step 1: randomly initialize k centroids (随机初始化)



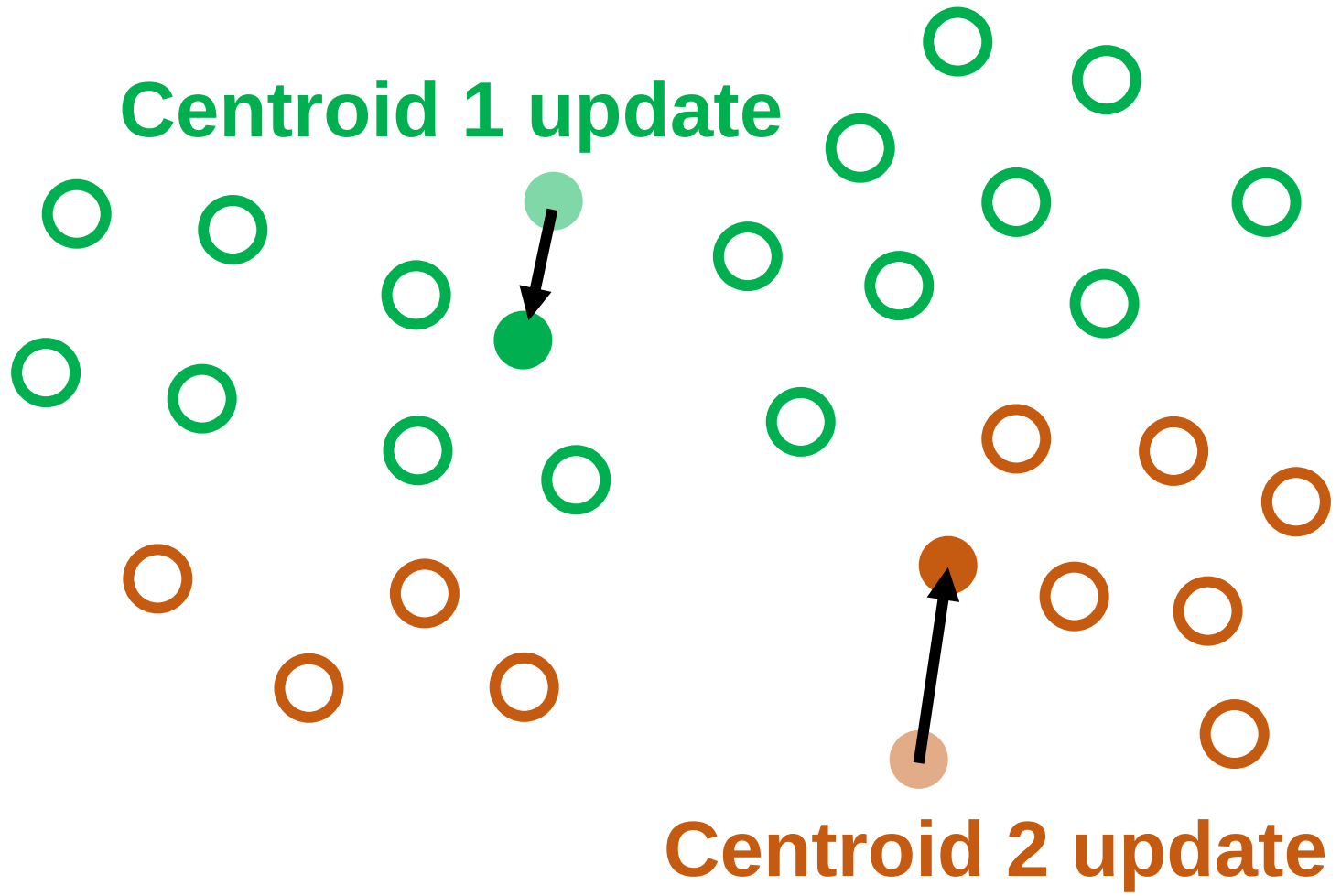
Kmeans step-by-step

Step 2: assign points to the closest centroid



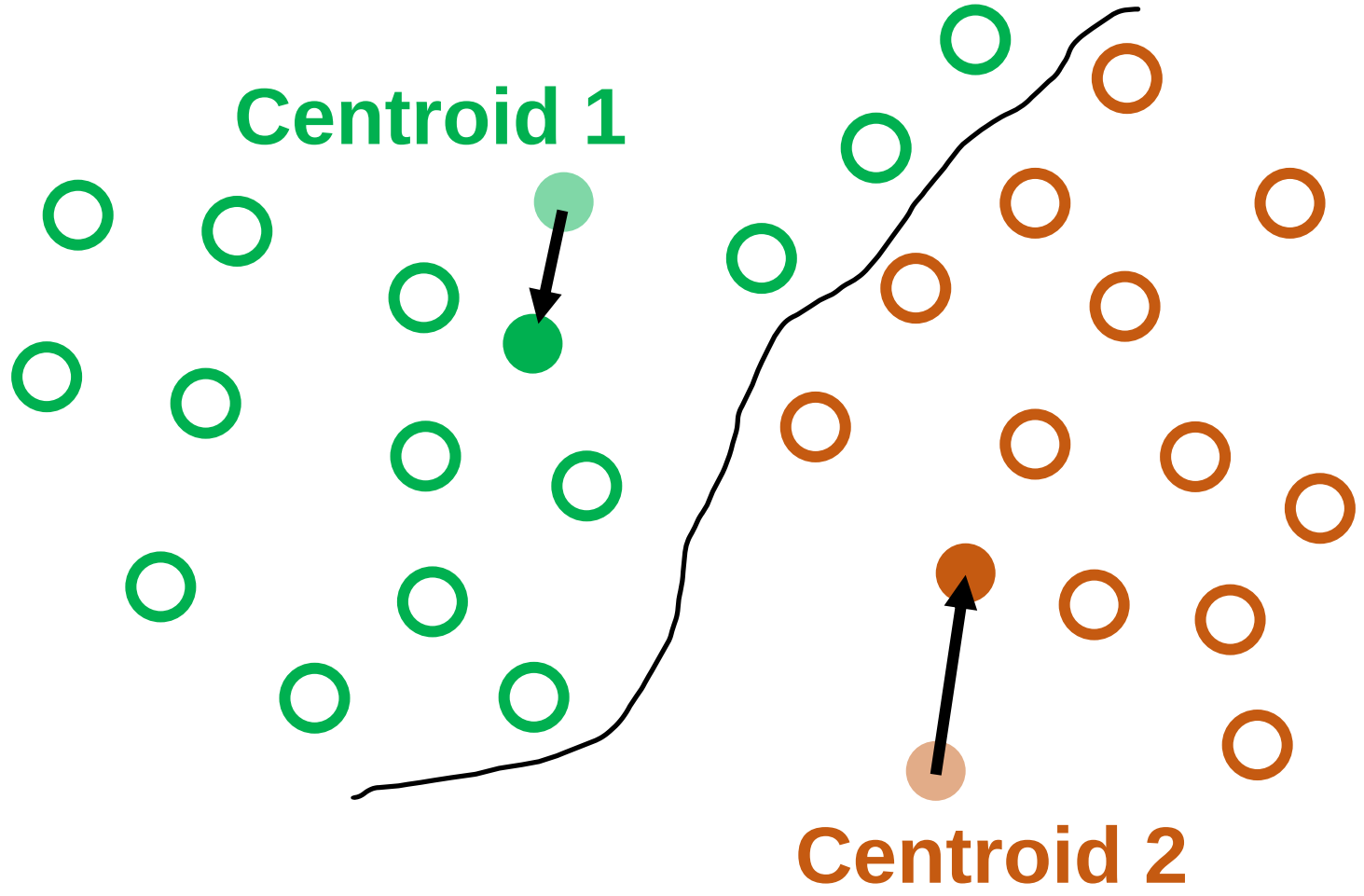
Kmeans step-by-step

Step 3: update the centroids (mean of clusters)



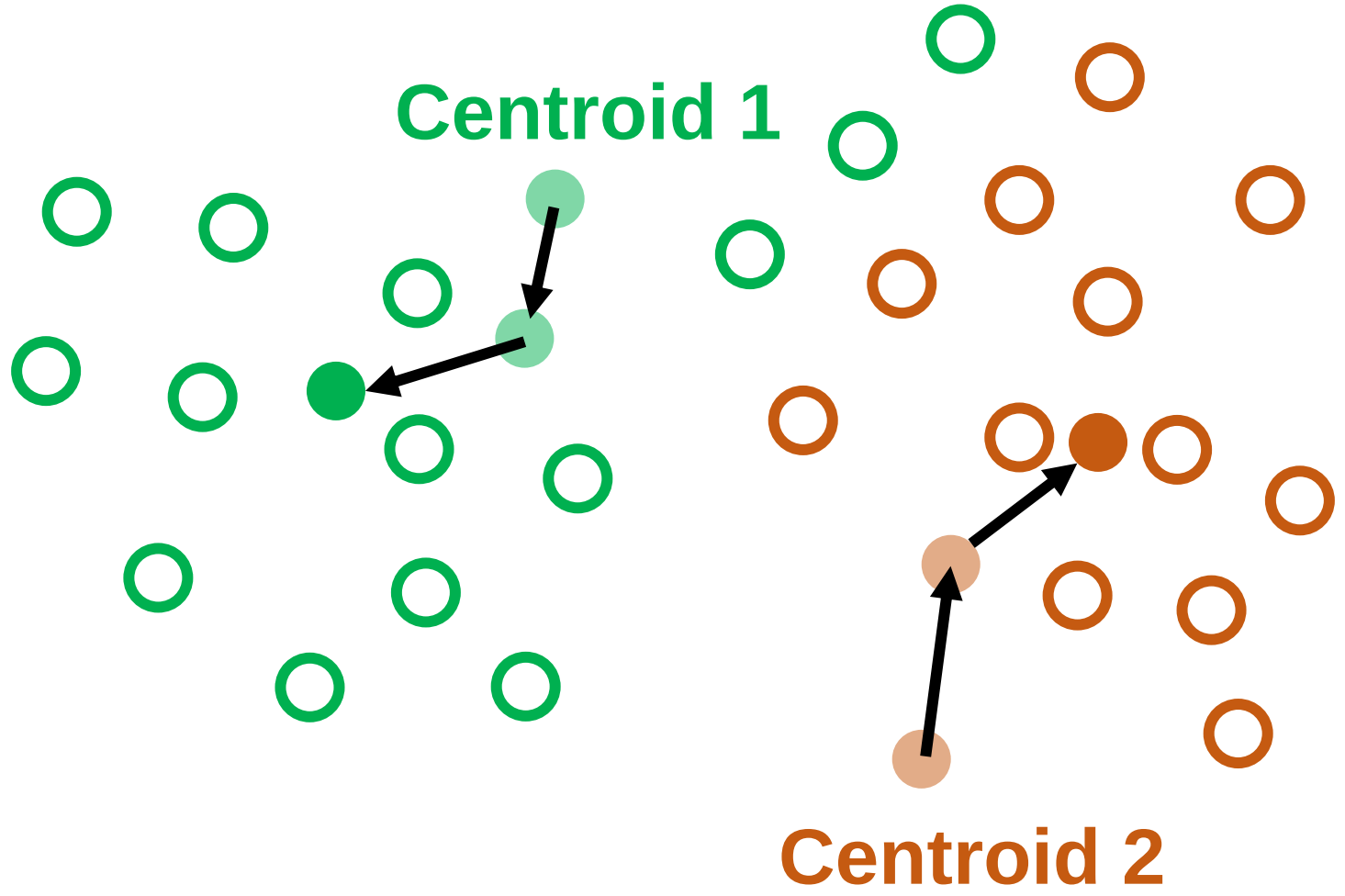
Kmeans step-by-step

Repeat step 2: assign points to the closest centroid



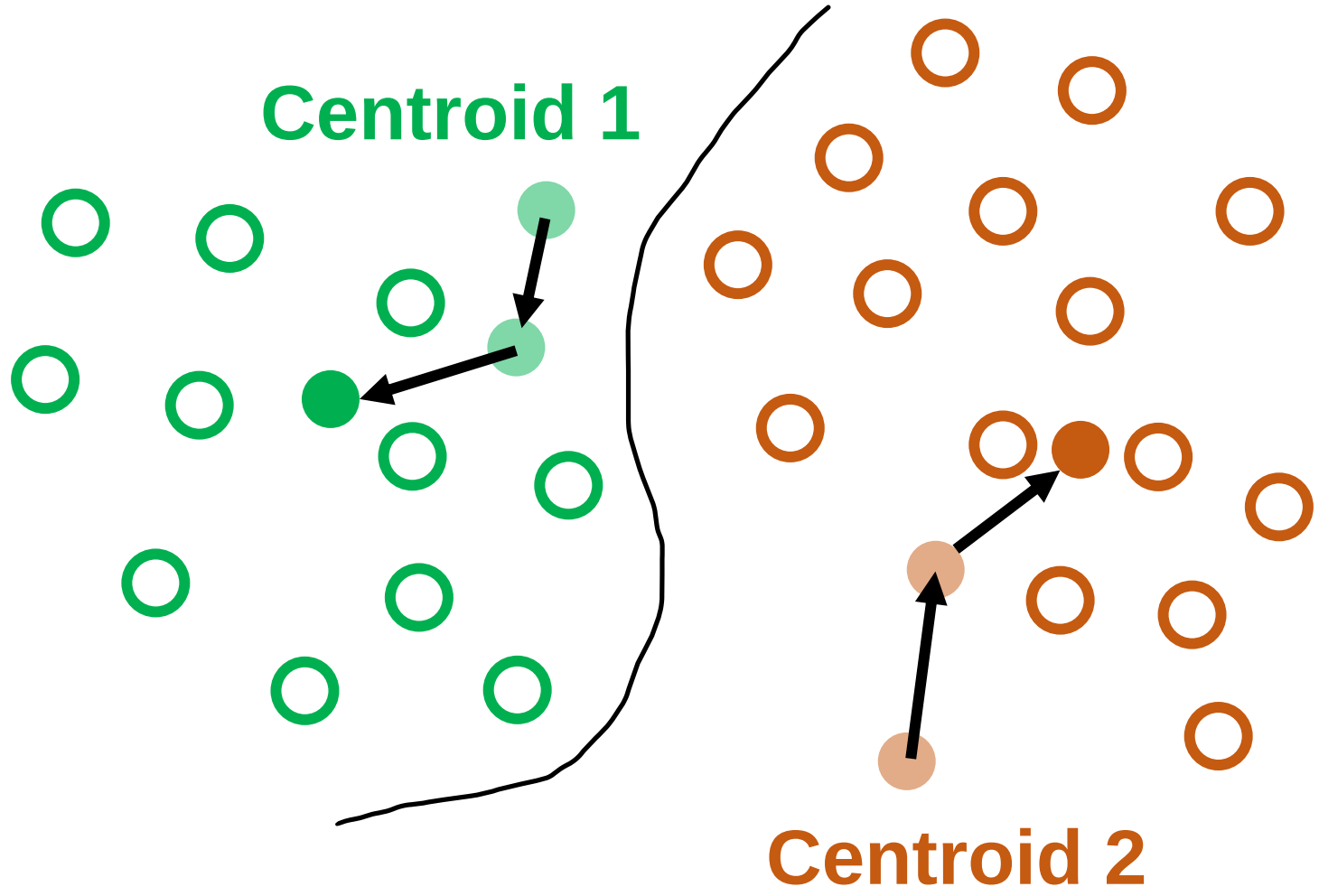
Kmeans step-by-step

Repeat step 3: update the centroids again



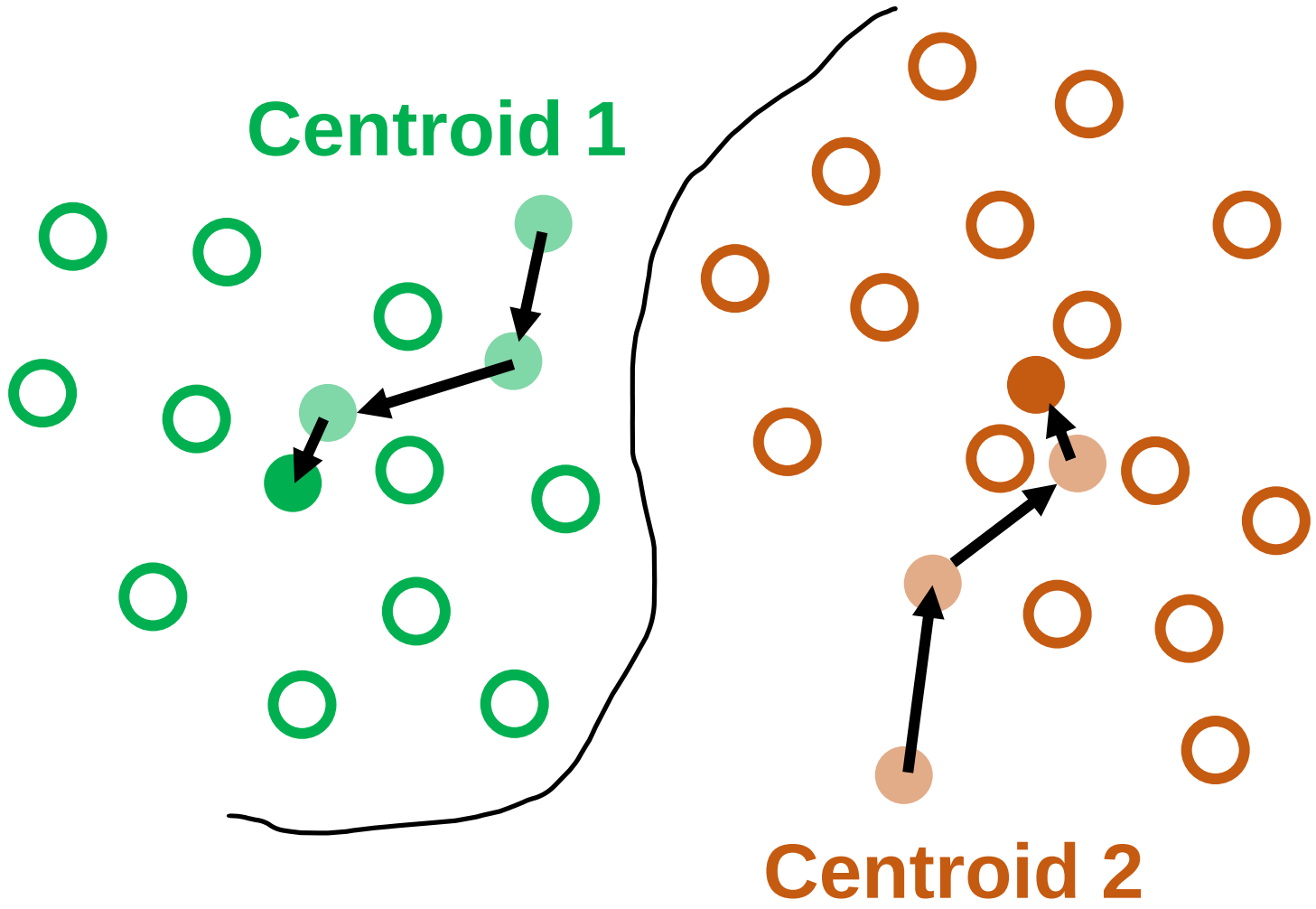
Kmeans step-by-step

Repeat step 2: assign points to the closest centroid



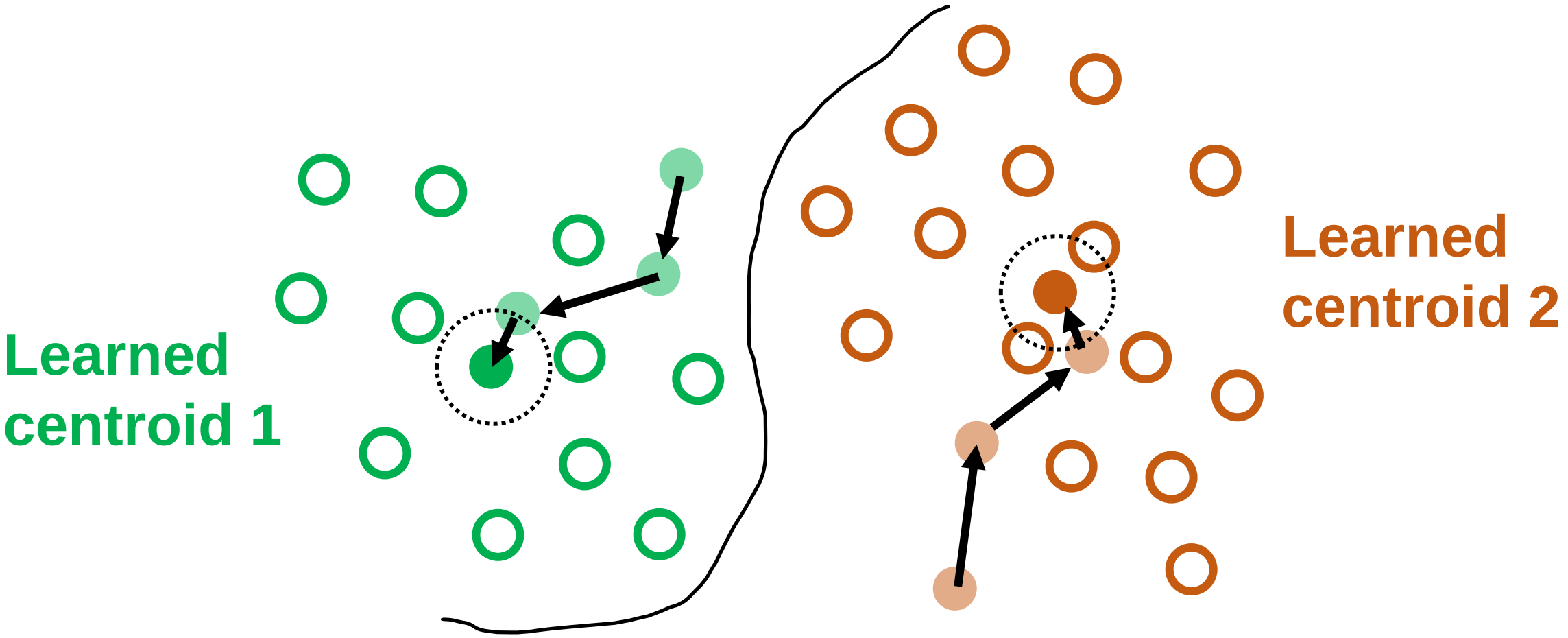
Kmeans step-by-step

Repeat step 3: re-calculate the mean of centroids



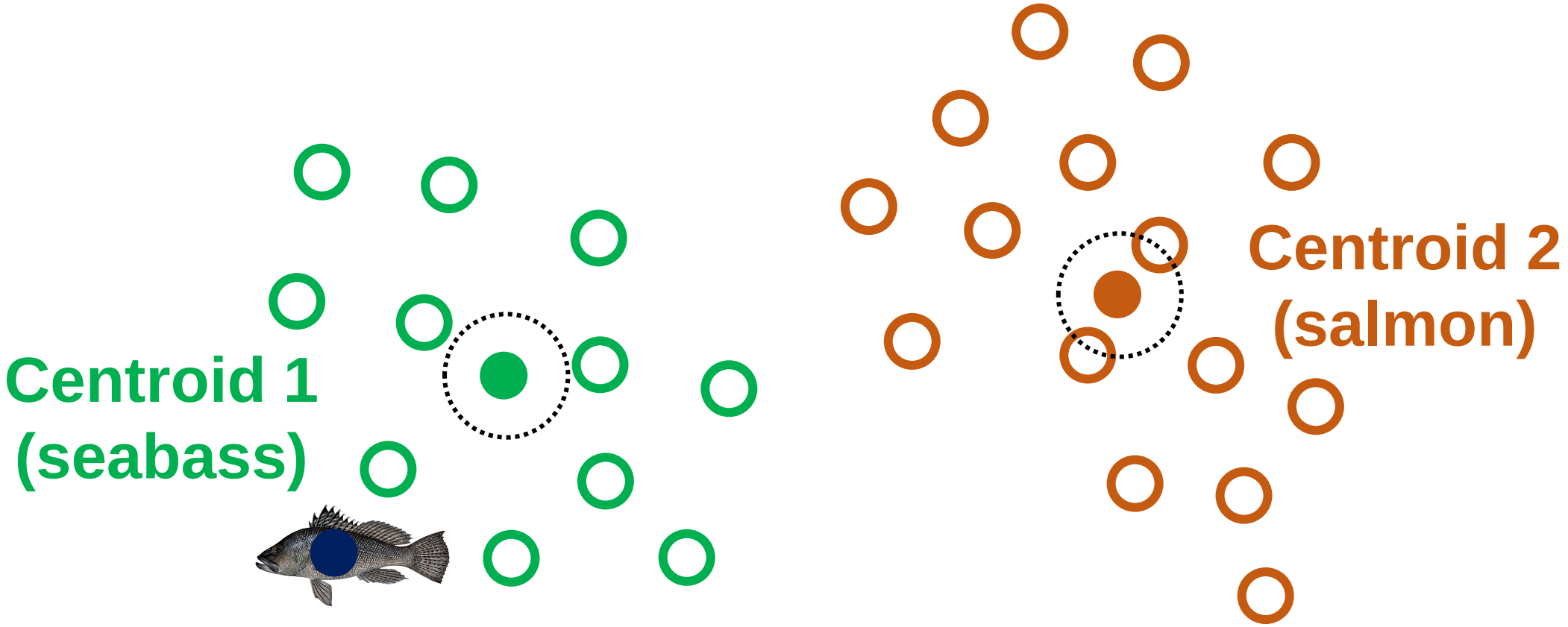
Kmeans step-by-step

Step 4: stop when centroids has no/tiny update



Kmeans step-by-step

Output the K centroids and separate fishes



This is seabass!

Summarize Kmeans algorithm

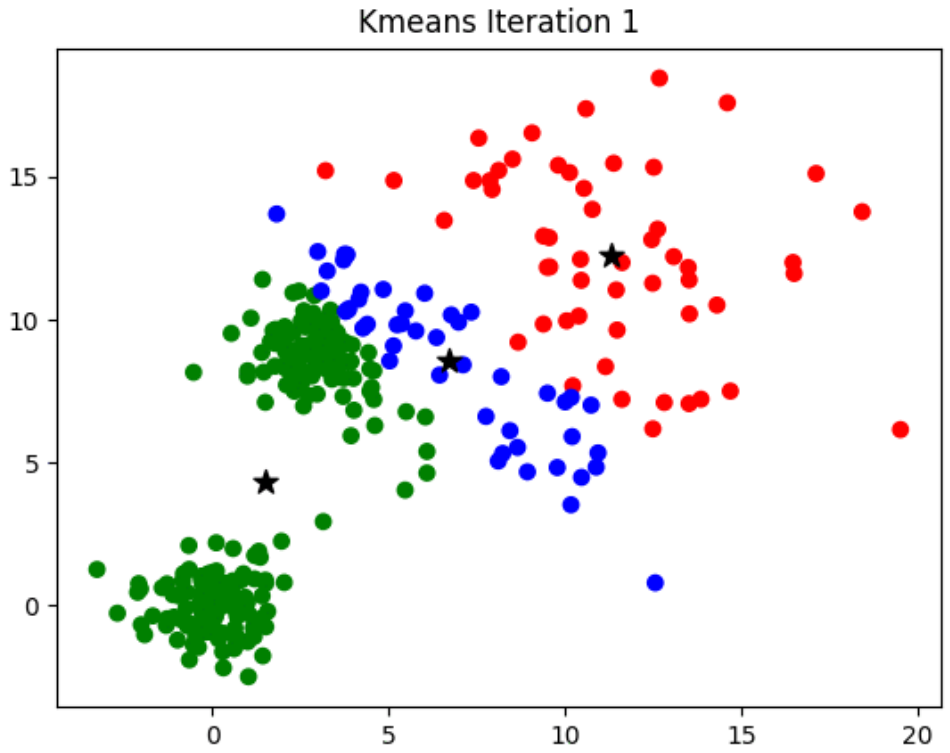
输入：K值
初始化：随机选择K个中心

```
while (true)
{
```

- 计算所有样本到中心的距离，将样本配备给最近的中心
- 利用新分配的样本更新K中心位置
- If(K个中心的位置不再改变)

```
    break;
```

输出：K个中心

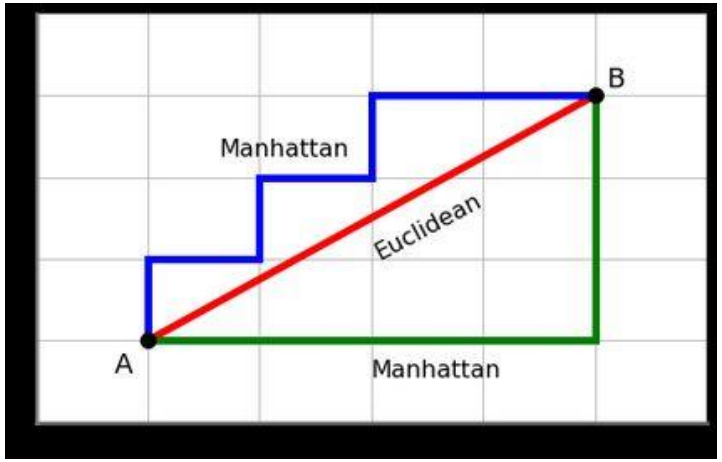


What distance metric to use?

Measure distance between samples and centre: L2-norm (Euclidean) and L1-norm.

- The *Euclidean distance* is the L_2 norm

$$L_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|^2 \right)^{1/2}.$$

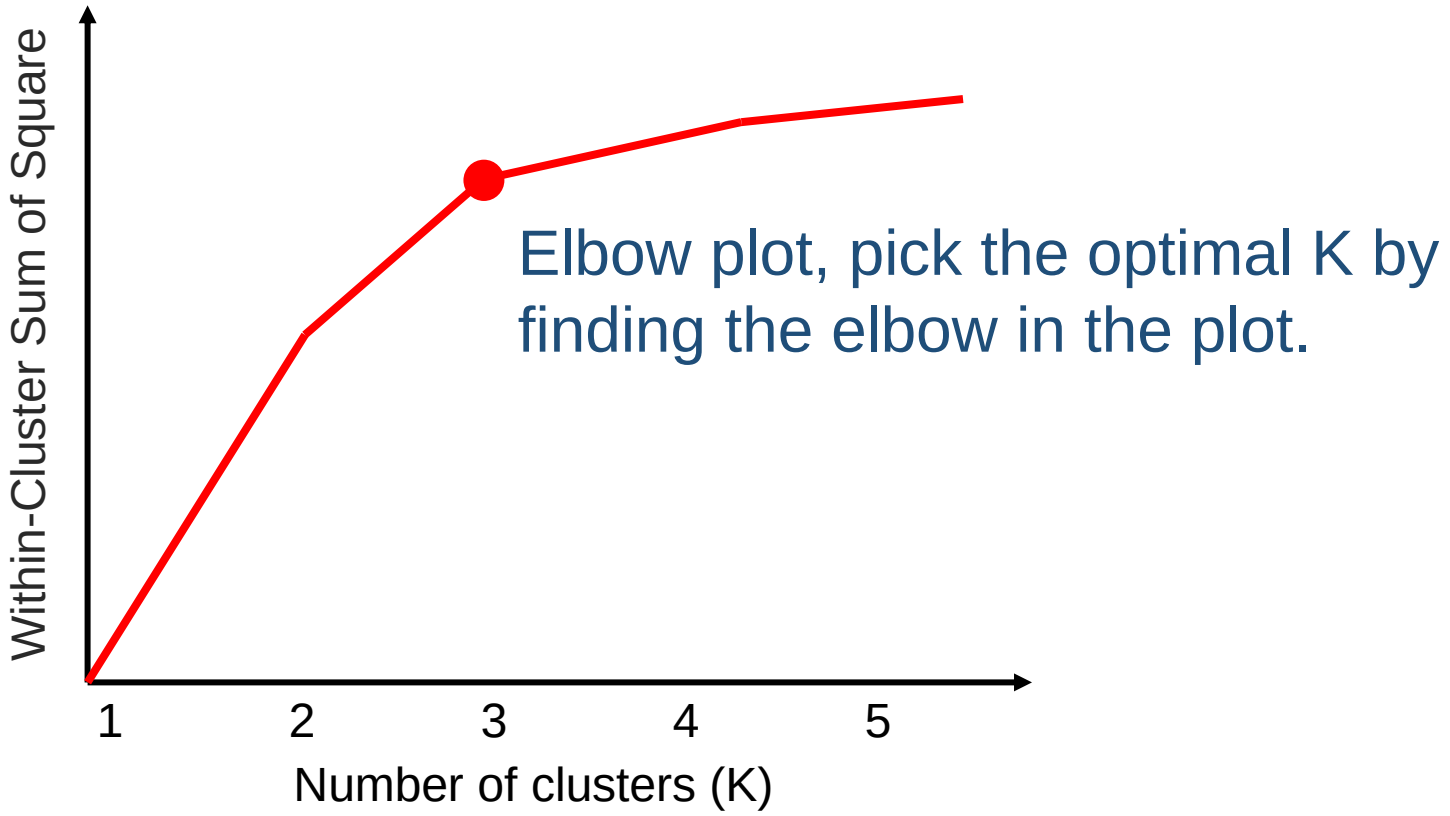


- The *Manhattan* or *city block distance* is the L_1 norm

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|.$$

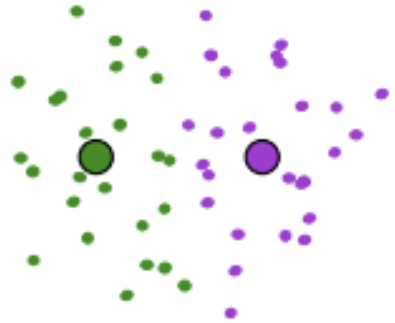
How to define K?

- Empirical: you have the prior knowledge about the data
- Trial and error: set K = 1, 2, 3, …, N, and compare the total variation and selecting the K with the least total variation.

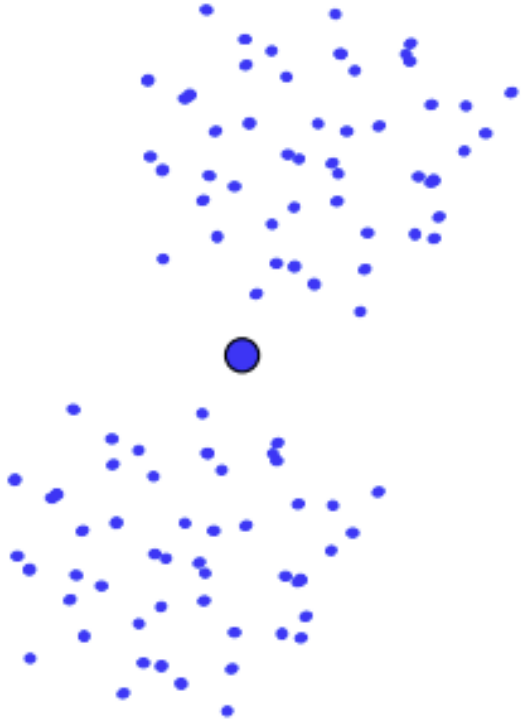


Challenges with Kmeans

You need to define K, K = 2, 3, 4… Many times you have no idea about cluster patterns in data



Would be better to have one cluster here

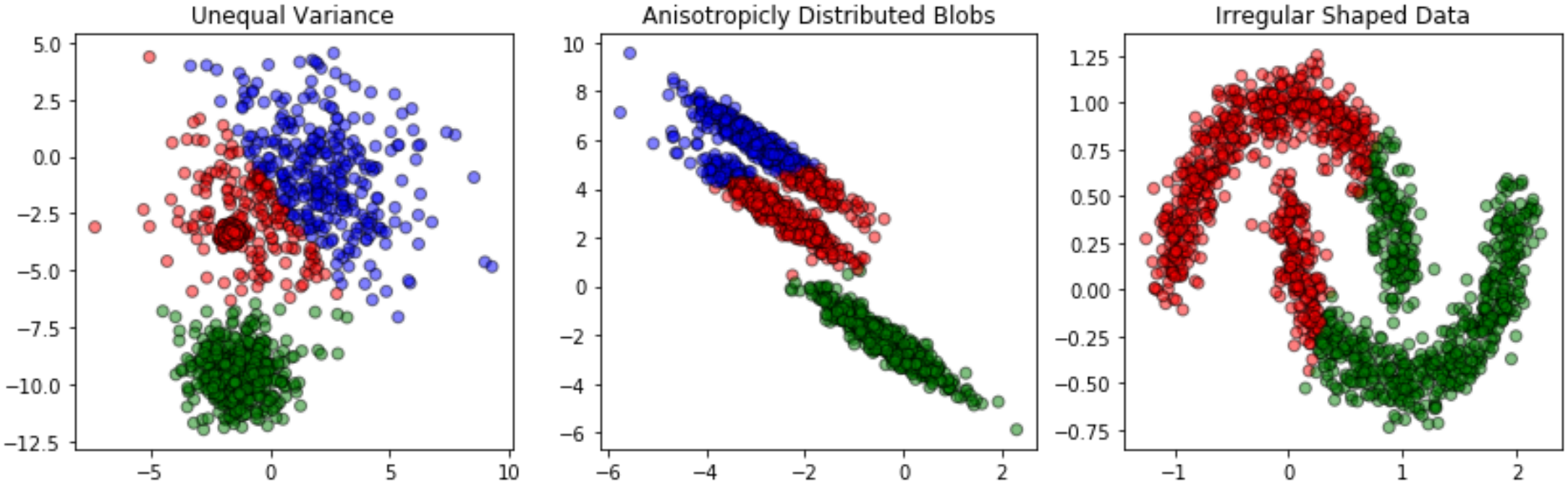


… and two clusters here

Challenges with Kmeans

Kmeans has geometric constraints due to the definition of the distance metric in the linear space.

Some failed or non-optimal examples



What is DBscan?

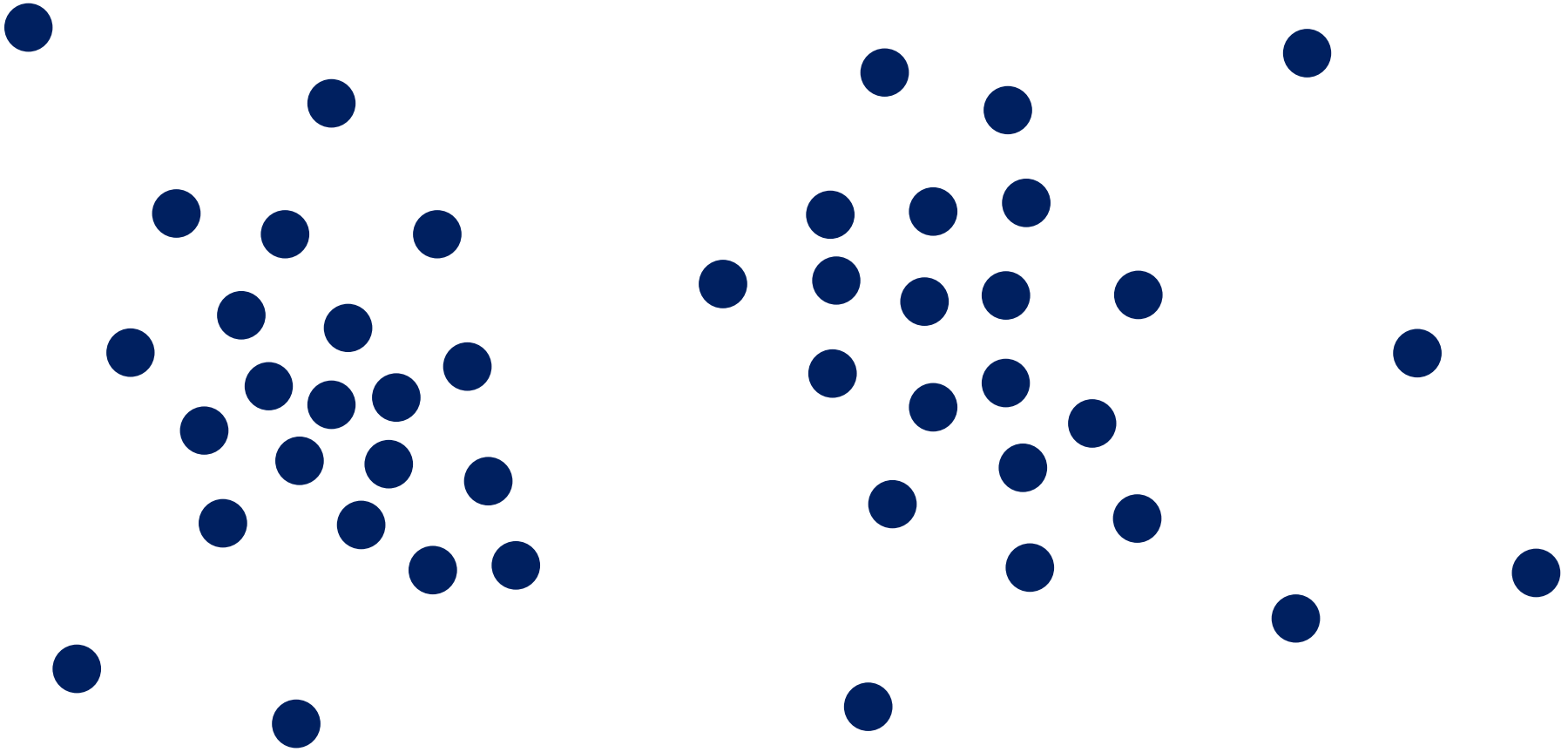
Density-Based scan

基于密度的

扫描/遍历

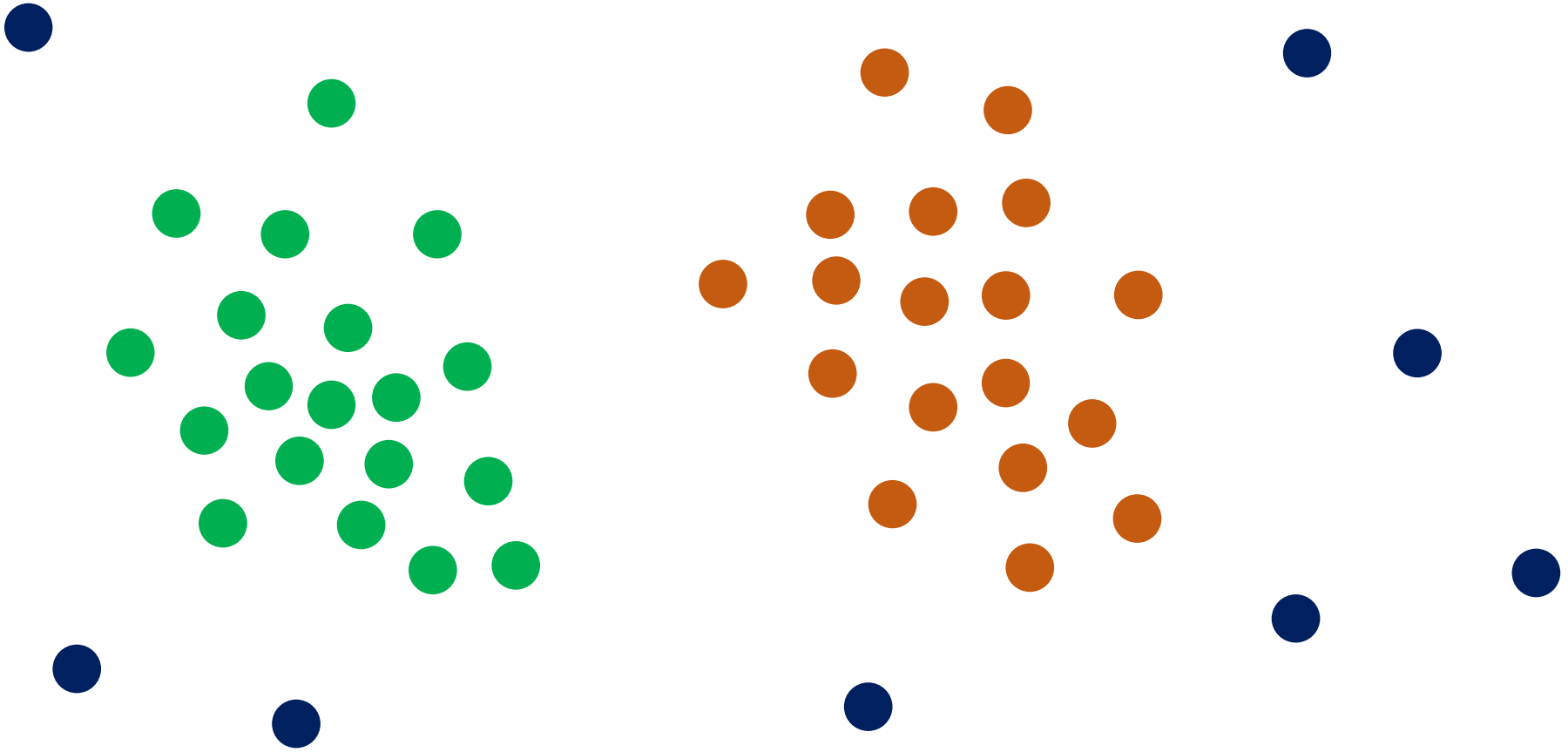
What is DBscan?

Clusters are **high-density** regions



What is DBscan?

Clusters are **high-density** regions

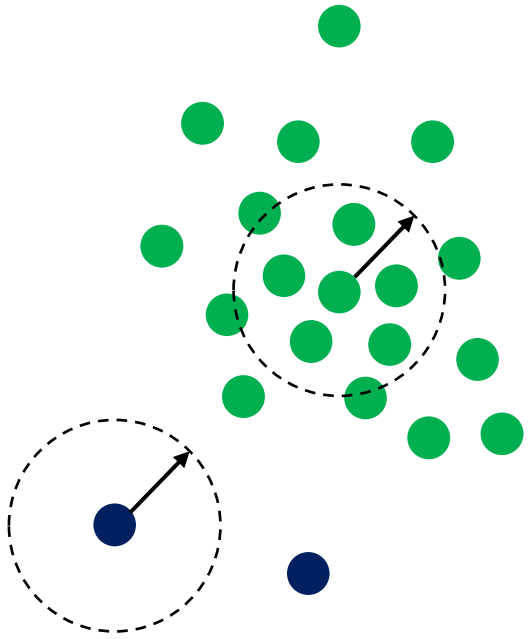


What is DBscan?

Definition of **high-density**:

It has number of points in an area

N **radius r**



It has **5** neighbouring points within **radius r**

It has **0** neighbouring points within **radius r**

DBscan definition

Clusters are high-density regions separated by low-density regions, only two parameters are required: N and r

Define N (number of points) and r (in a radius)

- Separate points as **core points** and **non-core points**

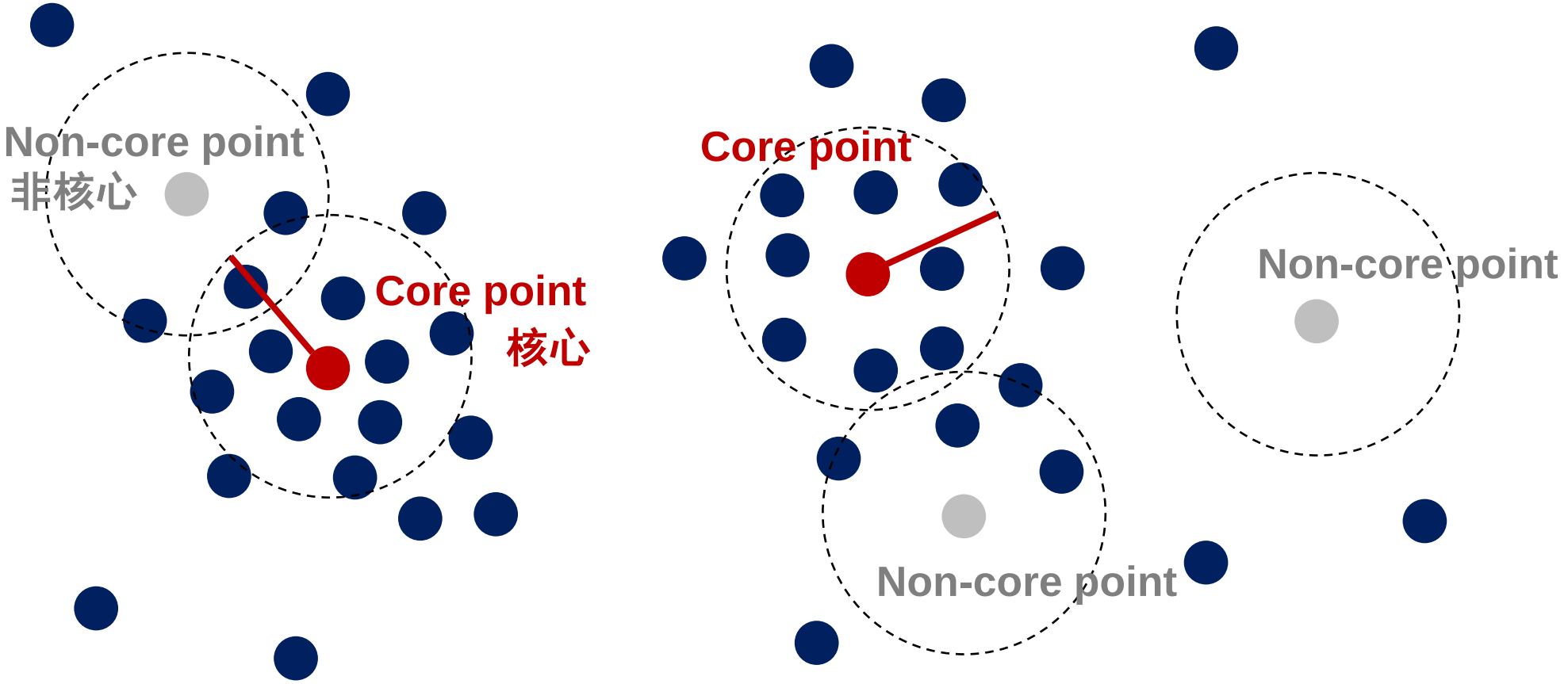
For each core point

- If it is not assigned to a cluster, create a new cluster
- Find connected core points (within r) and non-core points
- Use connected core points to grow

Output: clustered points

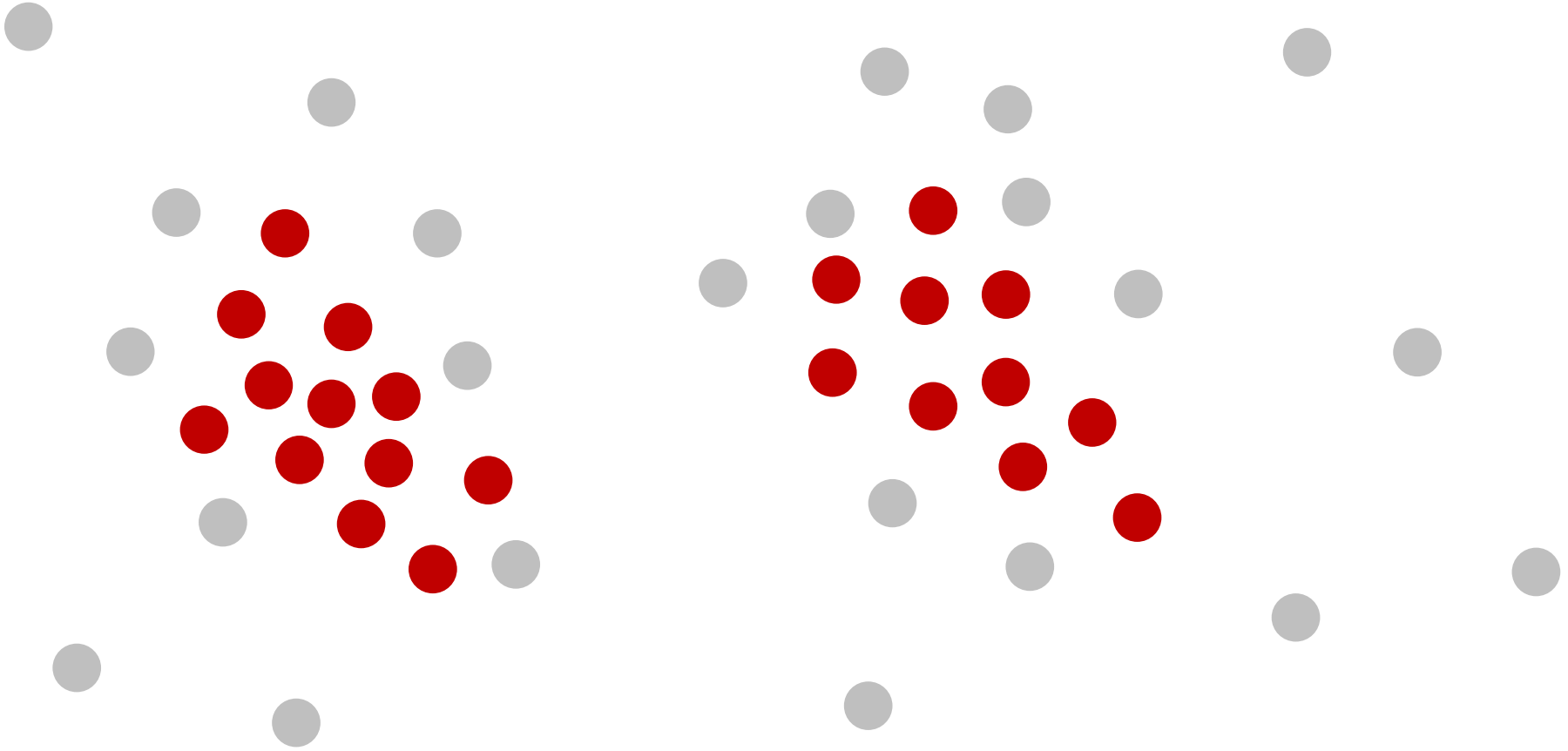
DBscan step-by-step

High-density regions: **N=4** points in radius **r=2**



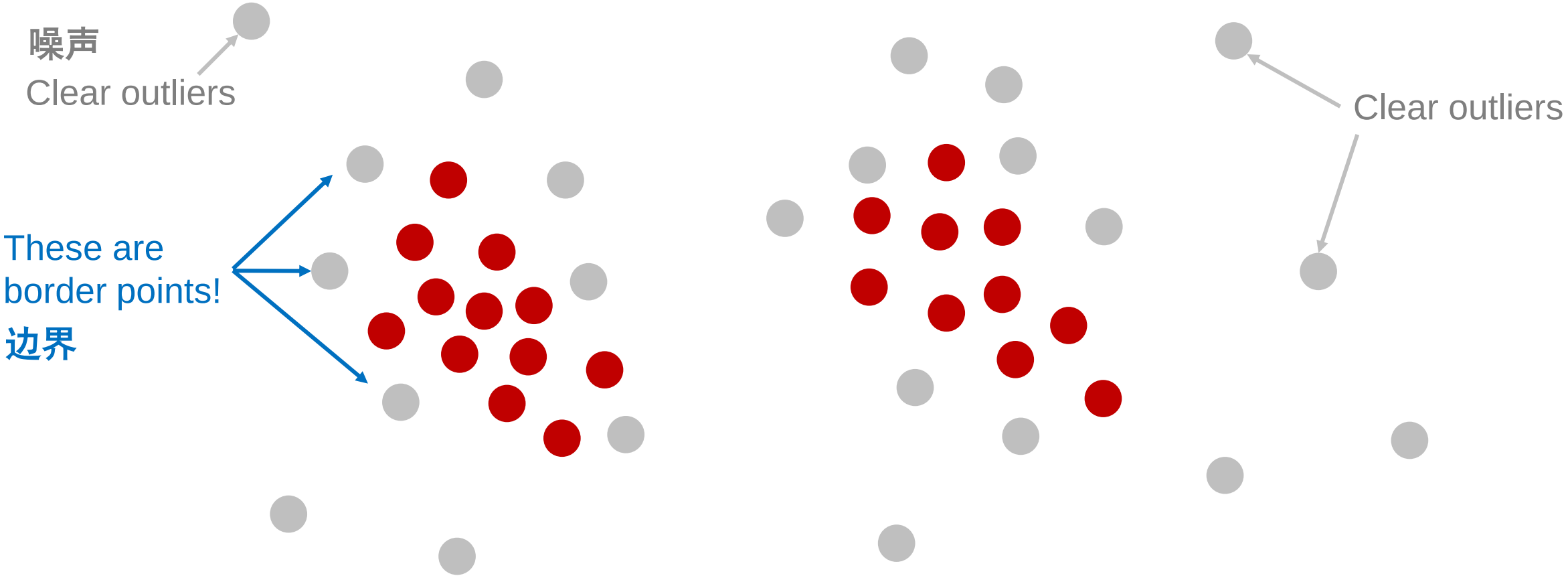
DBscan step-by-step

Step 1: separate core points and non-core points



DBscan step-by-step

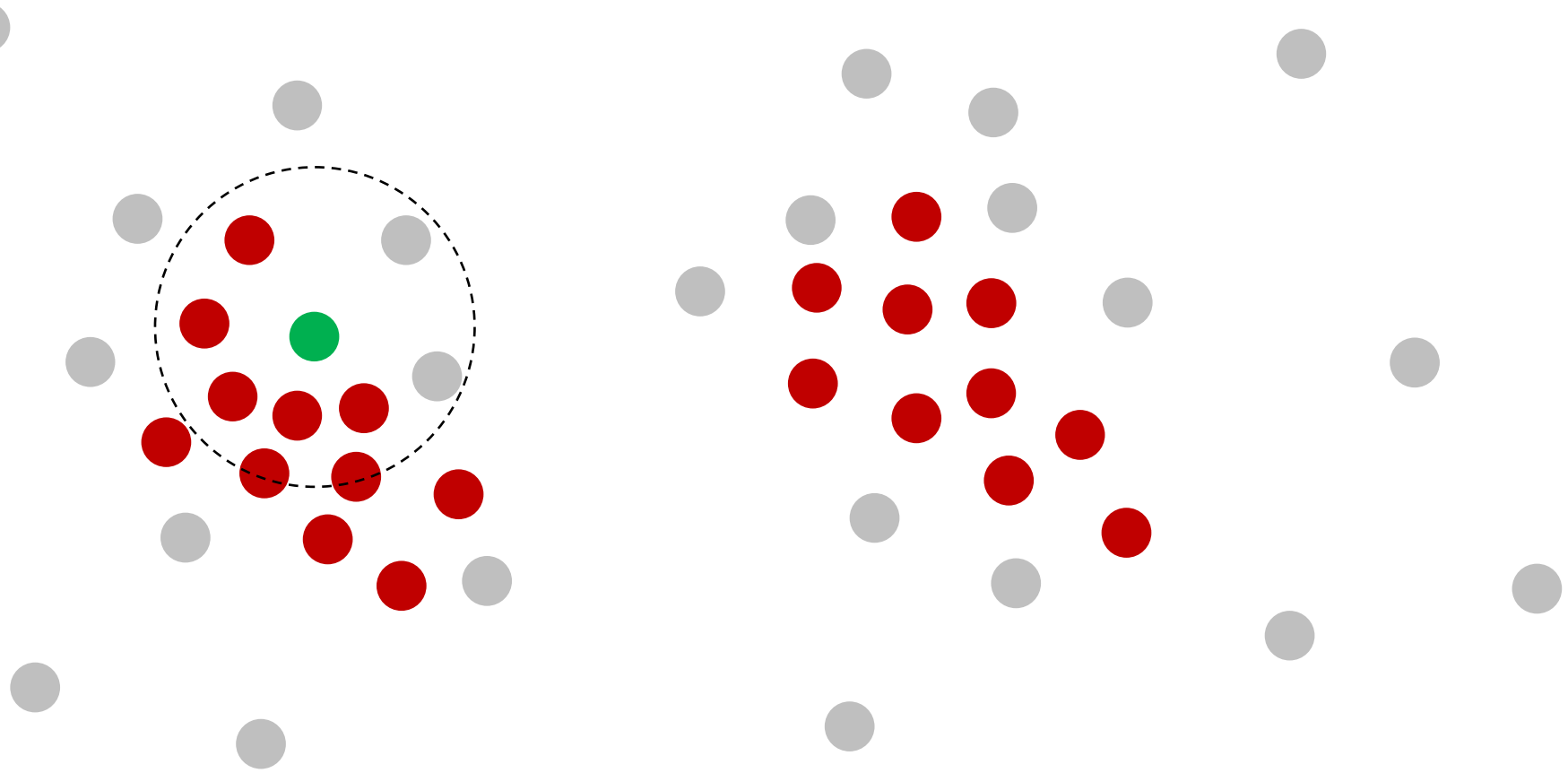
Step 1: separate core points and non-core points



DBscan step-by-step

Step 2: randomly select a core point to grow

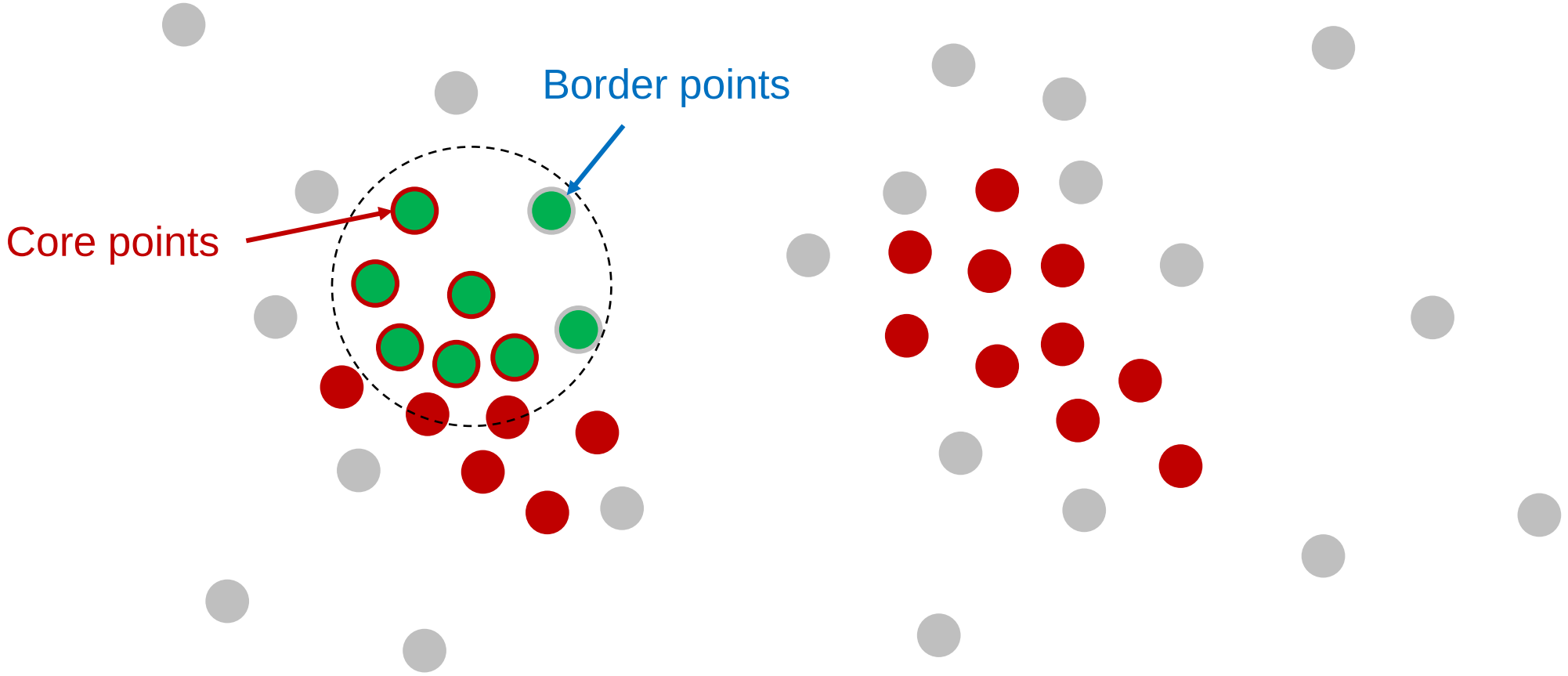
(find reachable points)



DBscan step-by-step

Step 2: randomly select a core point to grow

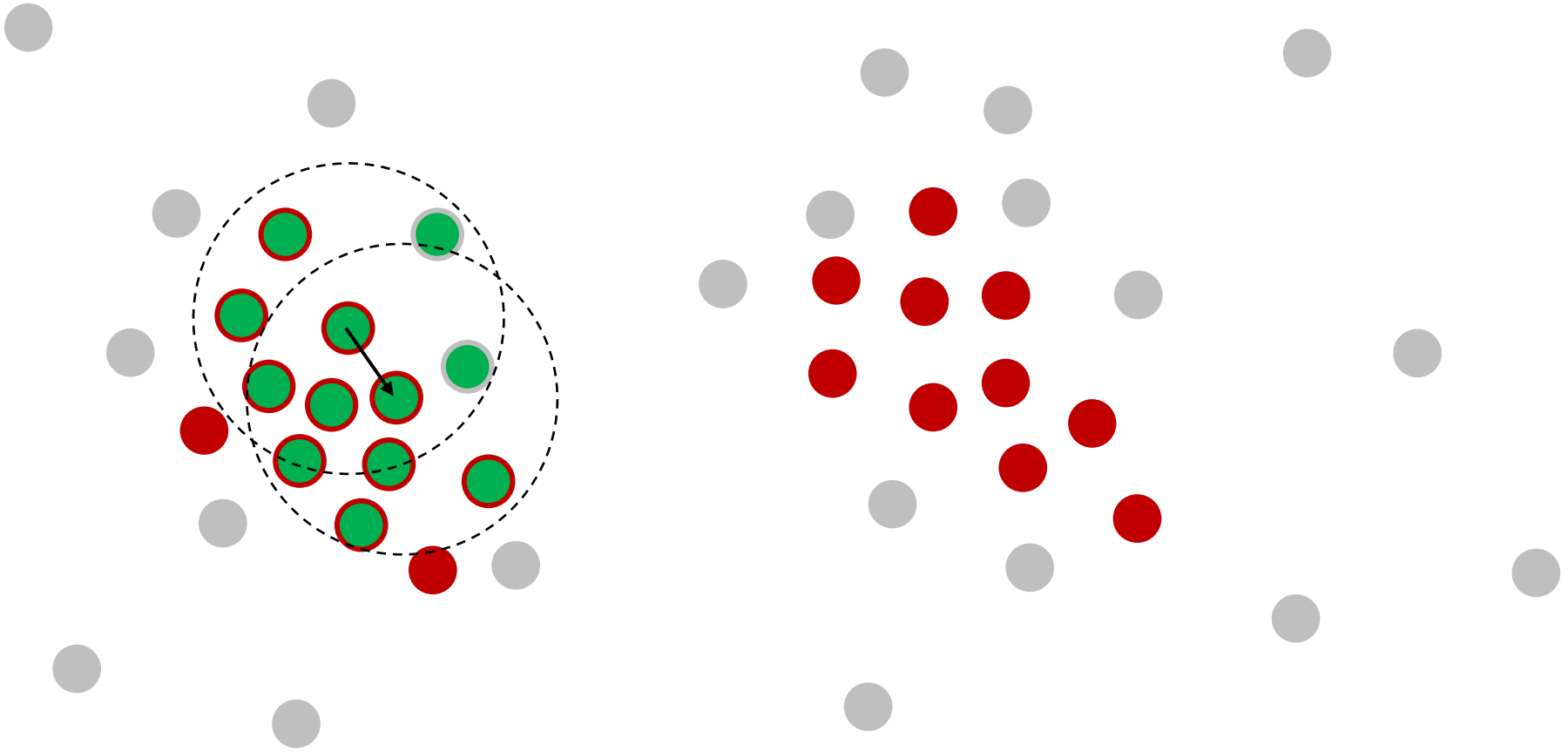
(core points can grow, border points can only add)



DBscan step-by-step

Step 3: move to next core point to grow

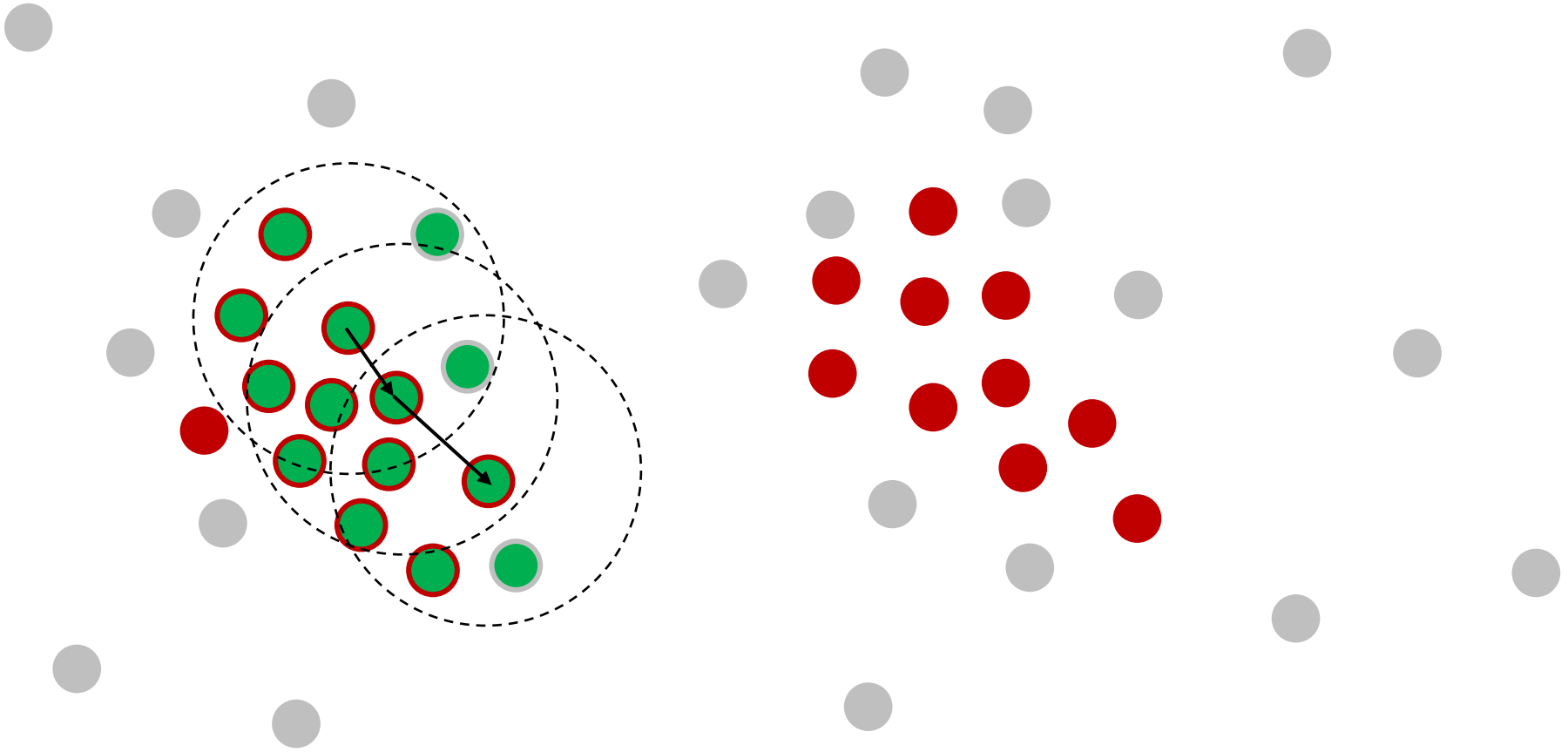
(core points can grow, border points can only add)



DBscan step-by-step

Step 3: move to next core point to grow

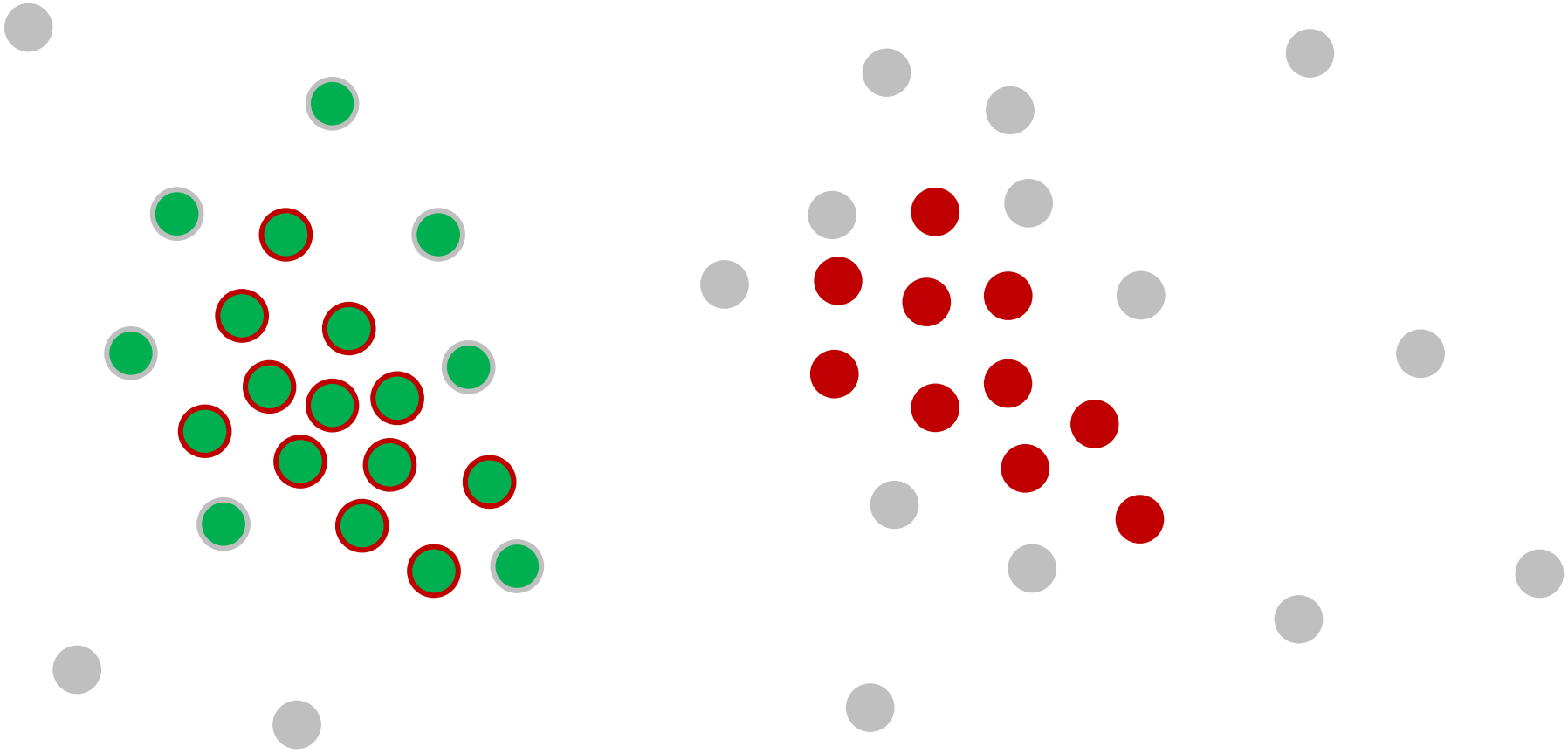
(core points can grow, border points can only add)



DBscan step-by-step

Step 3: scan over all connected core points

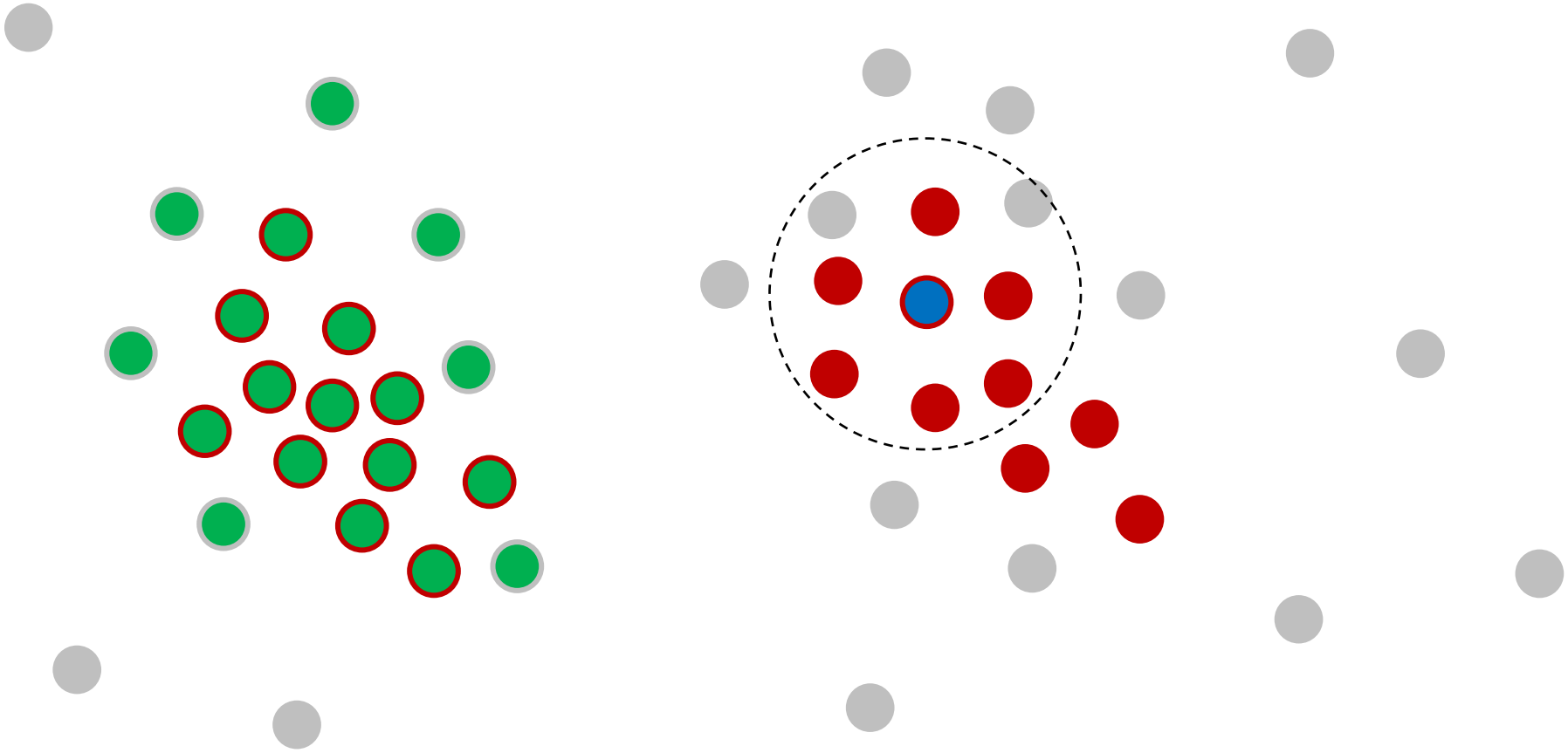
(core points can grow, border points can only add)



DBscan step-by-step

Repeat step 2: randomly select a core point to grow

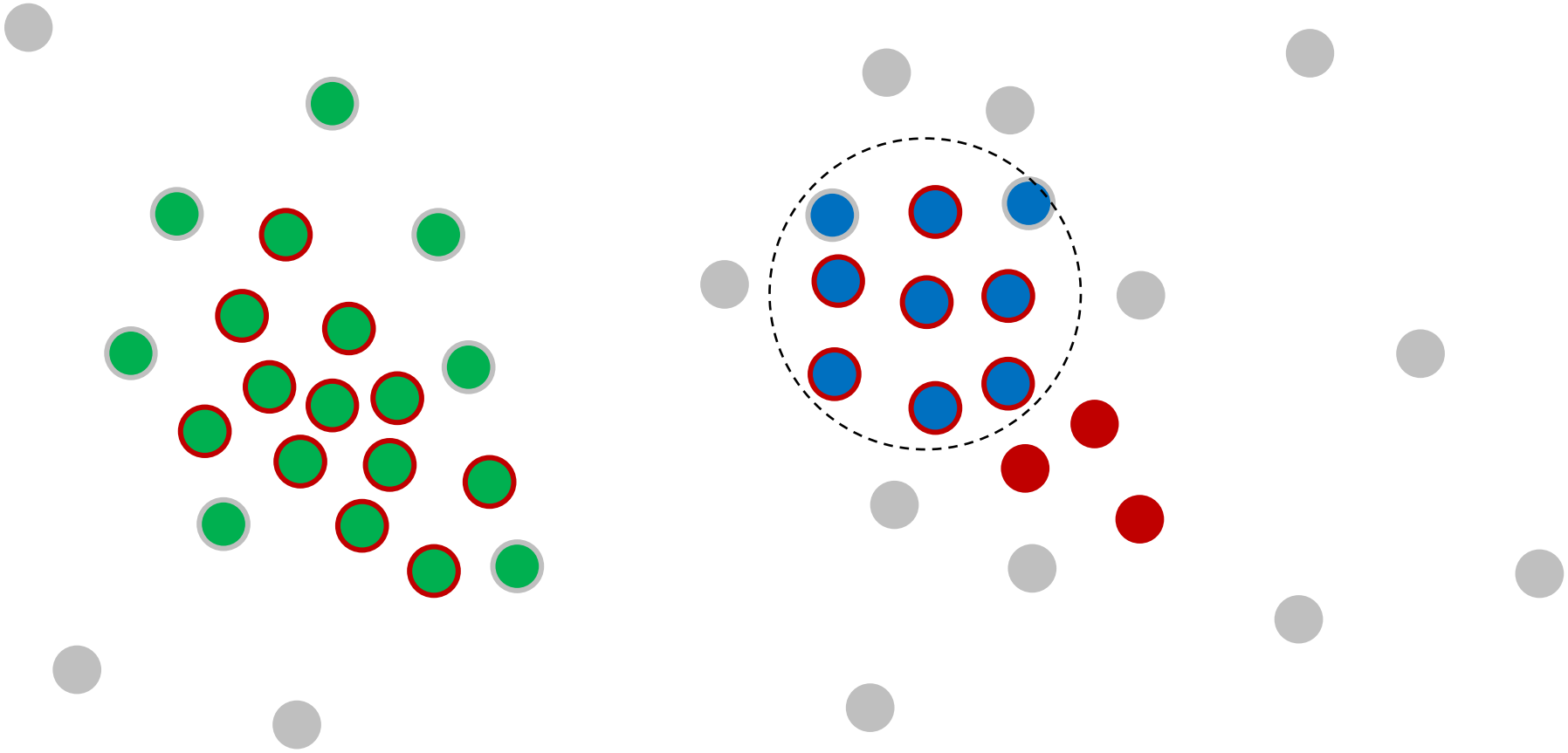
(core points can grow, border points can only add)



DBscan step-by-step

Repeat step 2: randomly select a core point to grow

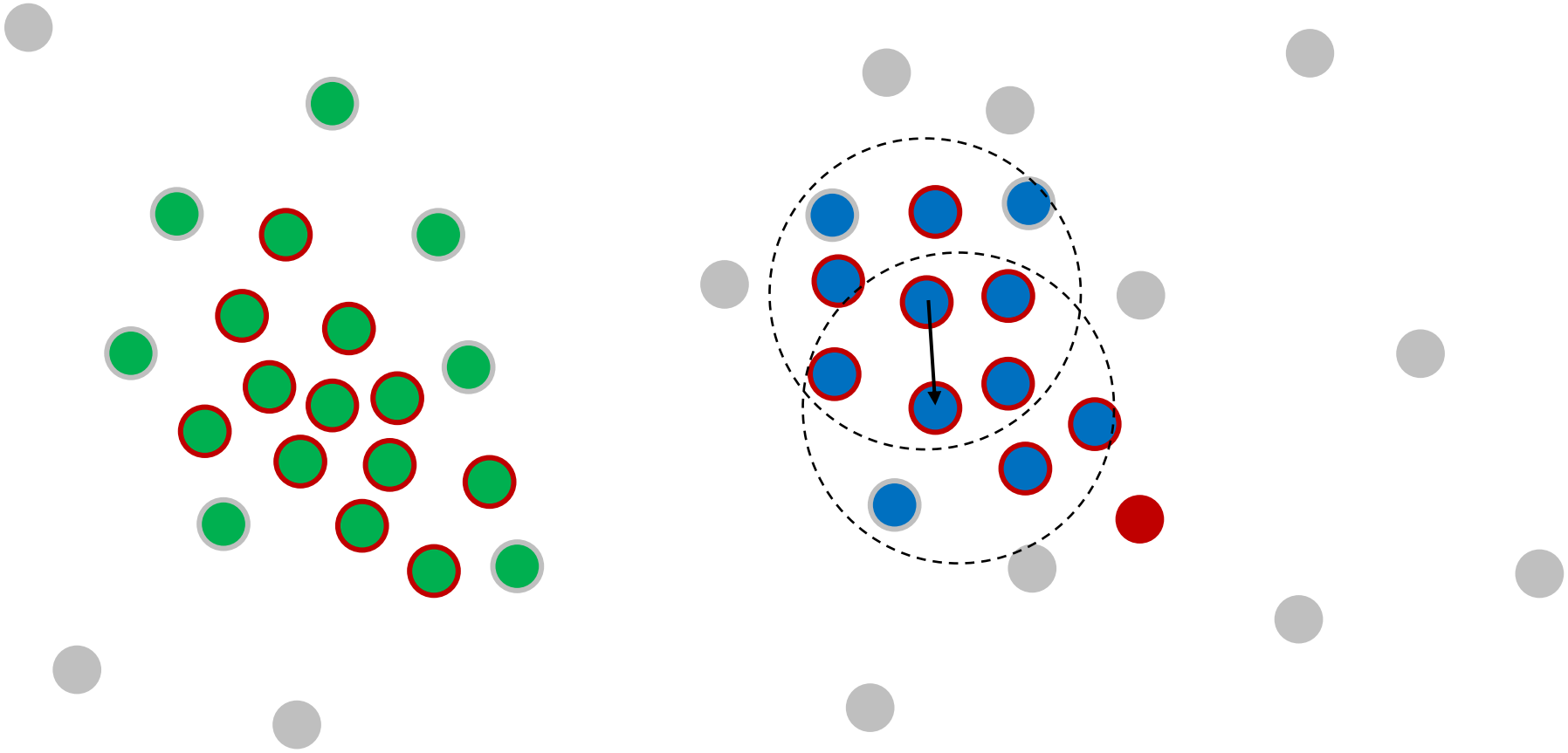
(core points can grow, border points can only add)



DBscan step-by-step

Repeat step 3: move to next core point to grow

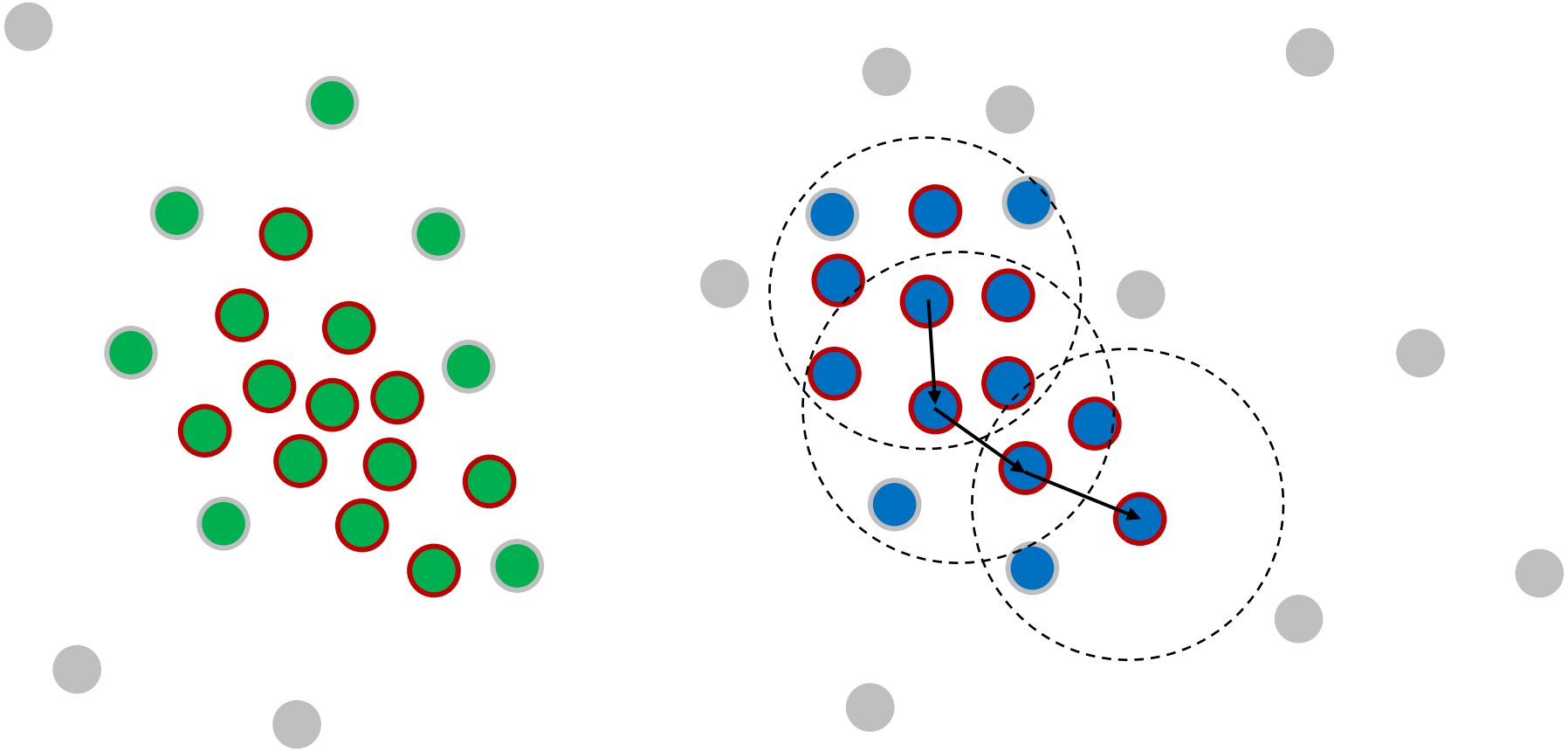
(core points can grow, border points can only add)



DBscan step-by-step

Repeat step 3: move to next core point to grow

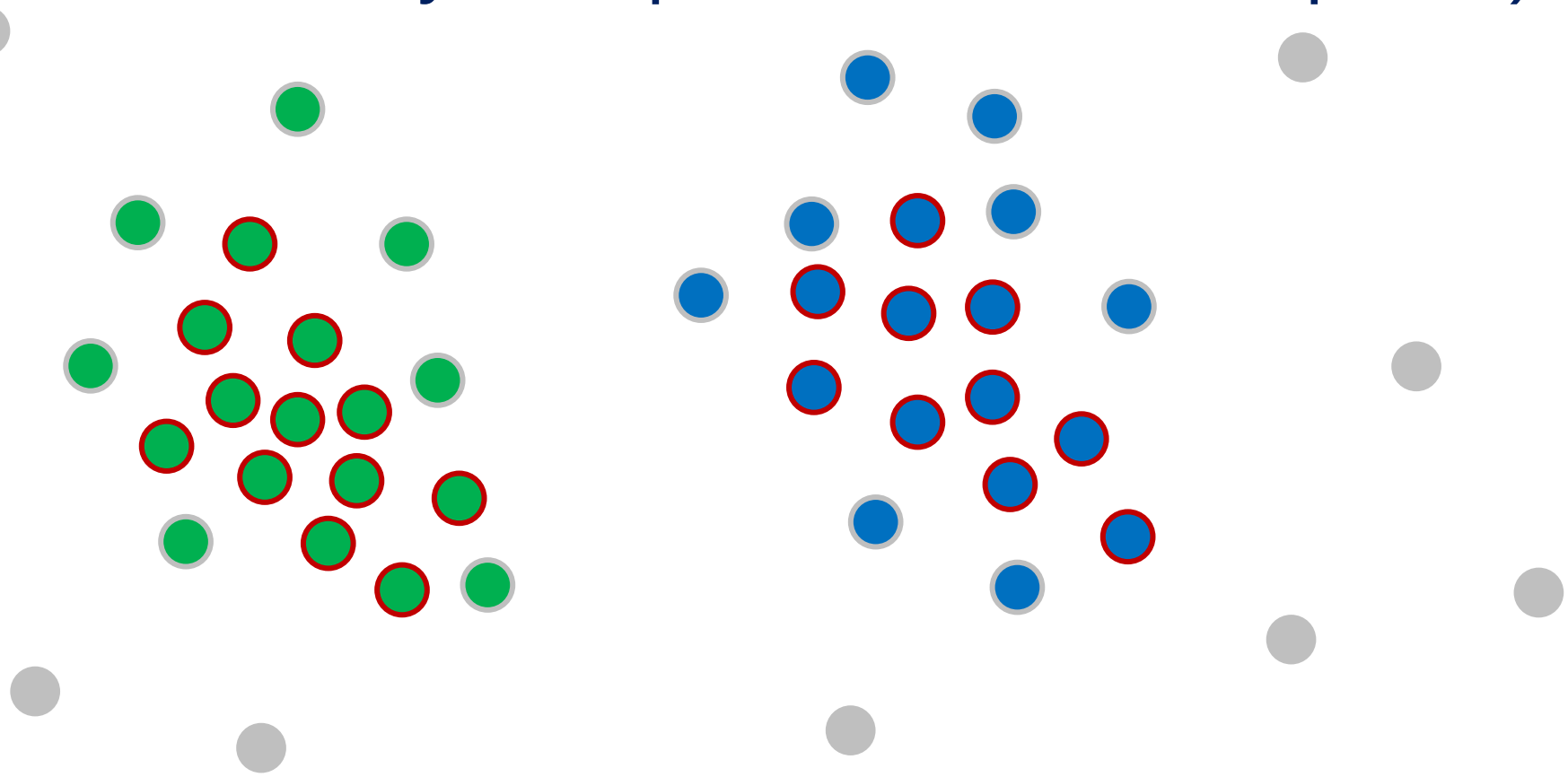
(core points can grow, border points can only add)



DBscan step-by-step

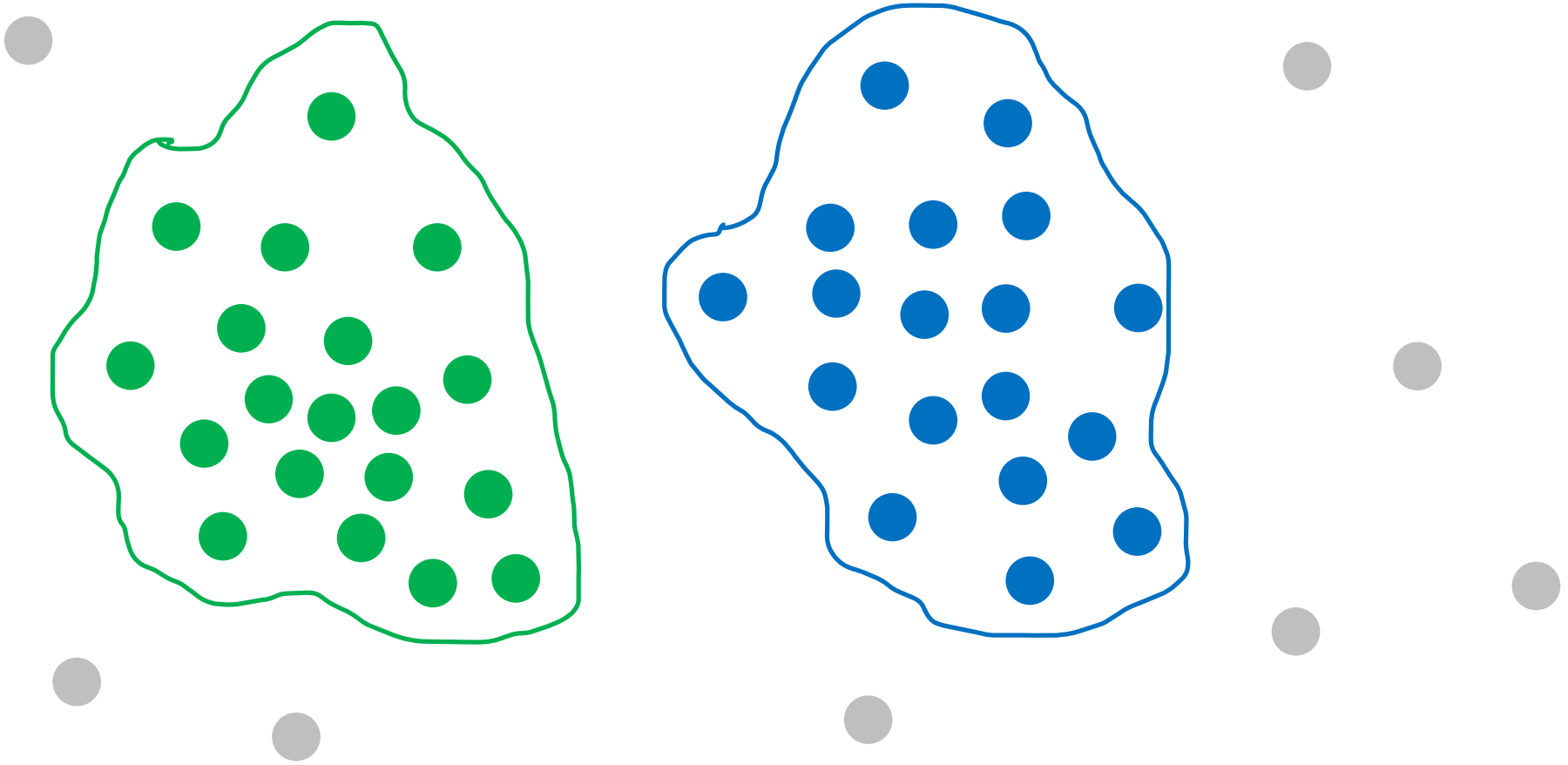
Scan over all core points

(connect nearby core points and boarder points)



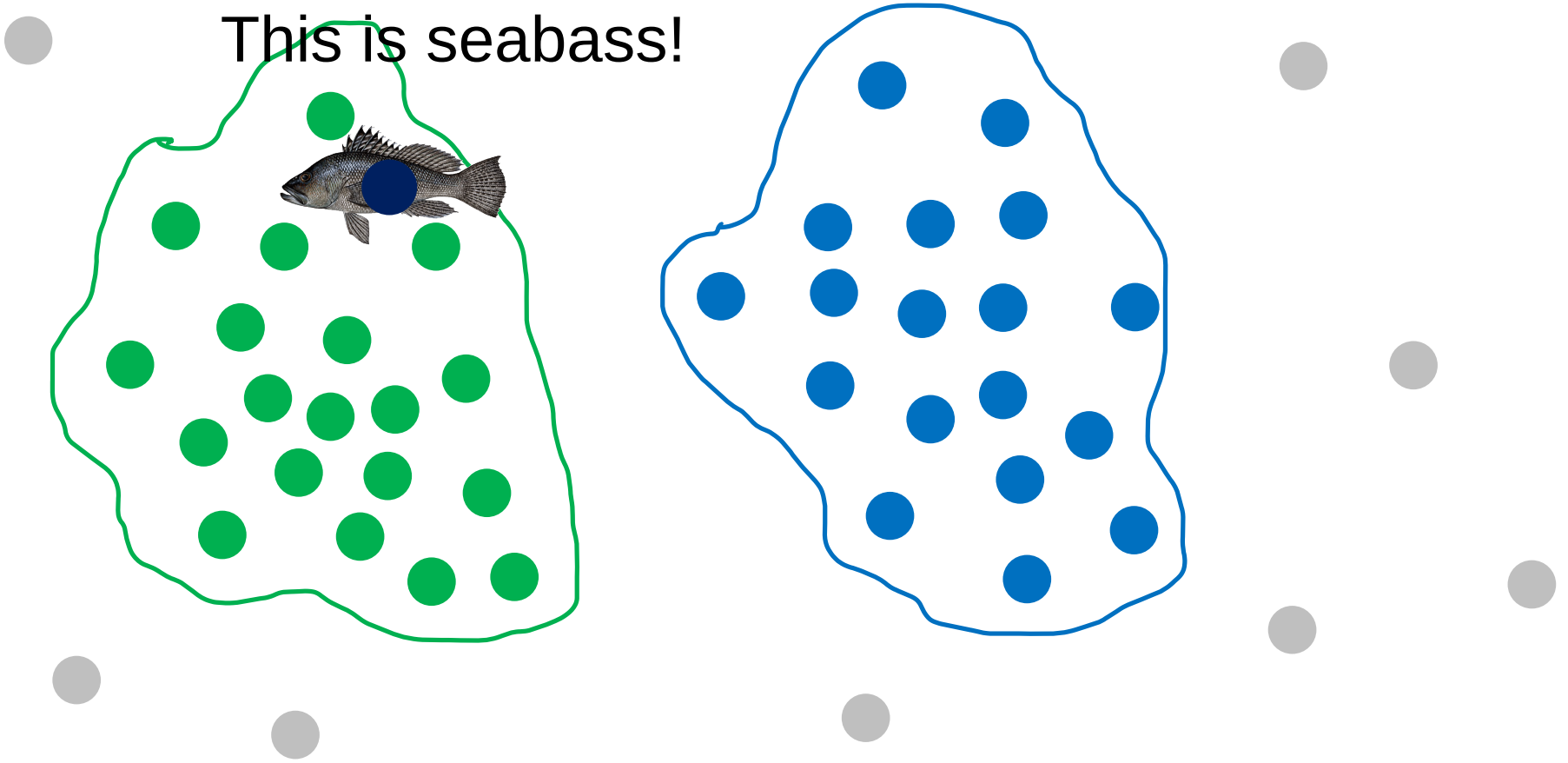
DBscan step-by-step

Output clustering results

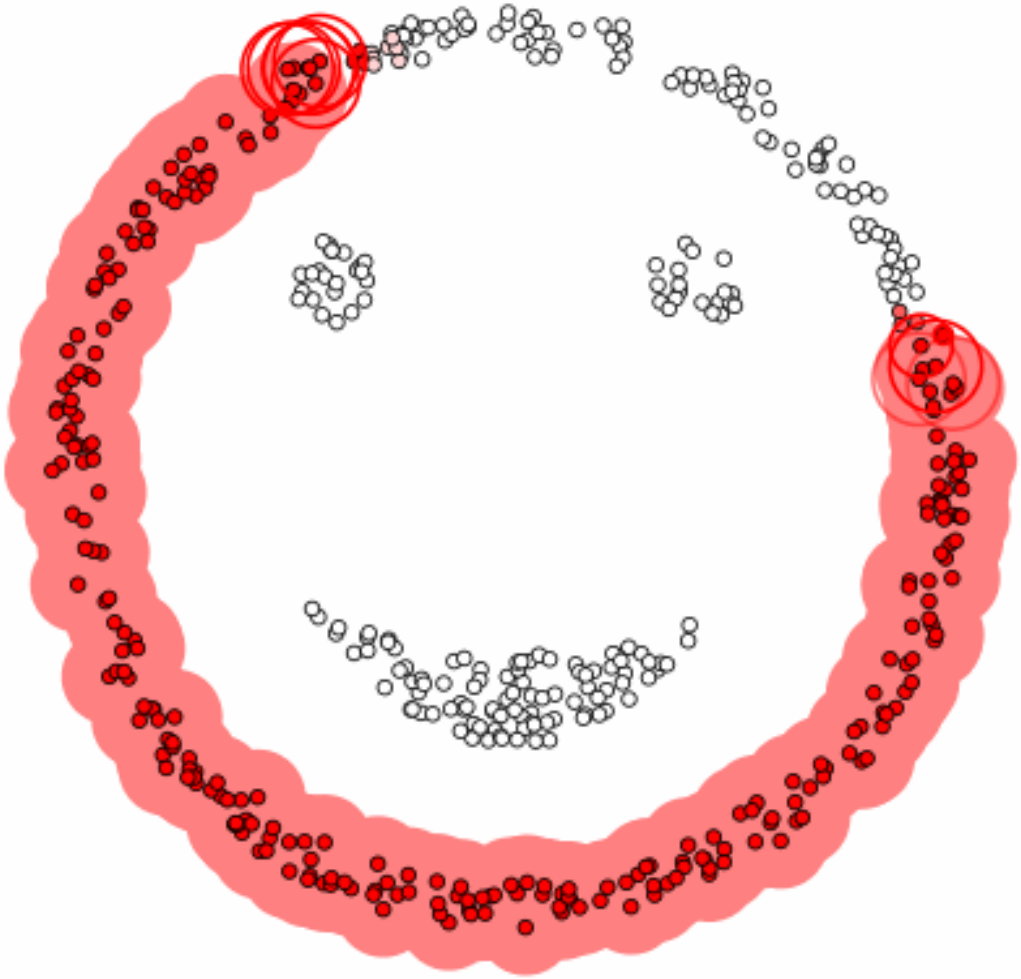


DBscan step-by-step

In our fish sorting case, find 2 clusters (seabass and salmon)



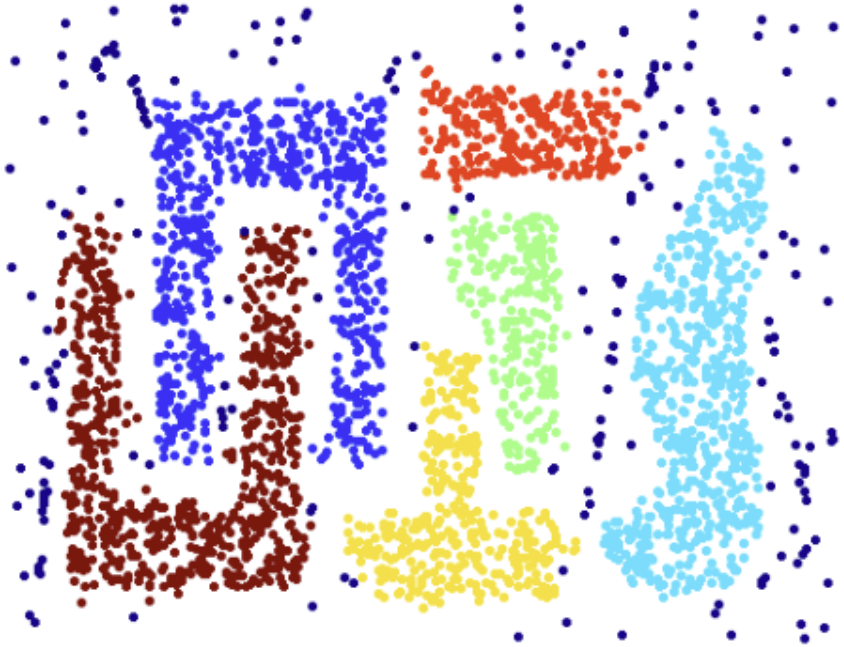
Overview of DBscan



Strengths of DBscan

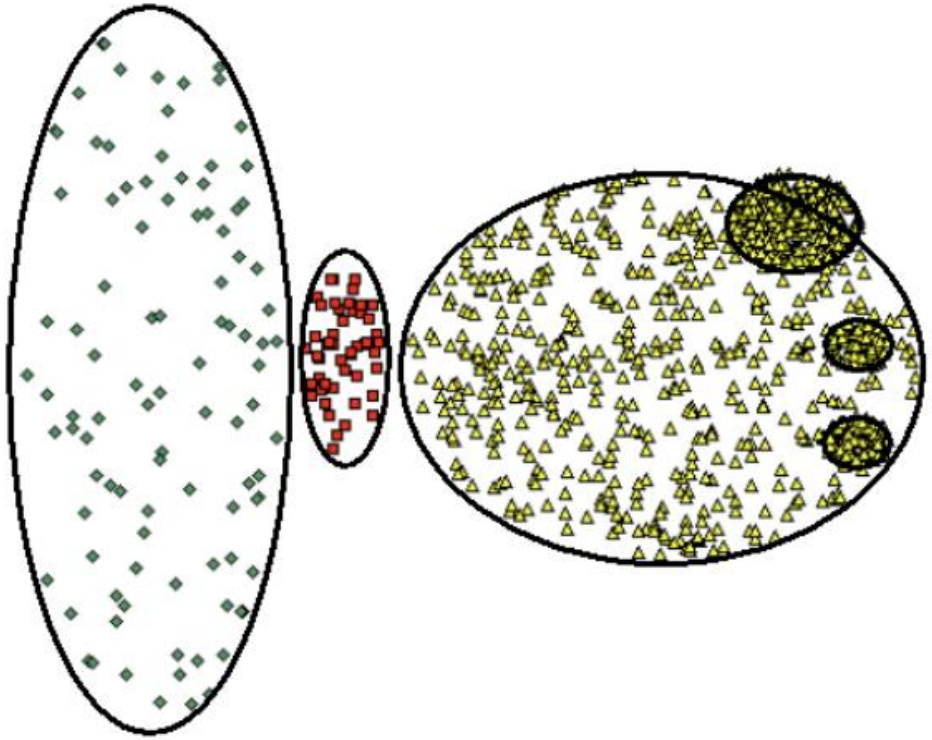
Original points

Clustered points



- Do not need to set the number of clusters
- Resistant to Noise
- Can handle clusters of different shapes and sizes

Weakness of DBscan



- If data is completely unknown, difficult to set N (number of points) and r (radius)
- Does not work well with multiple clusters with varying densities

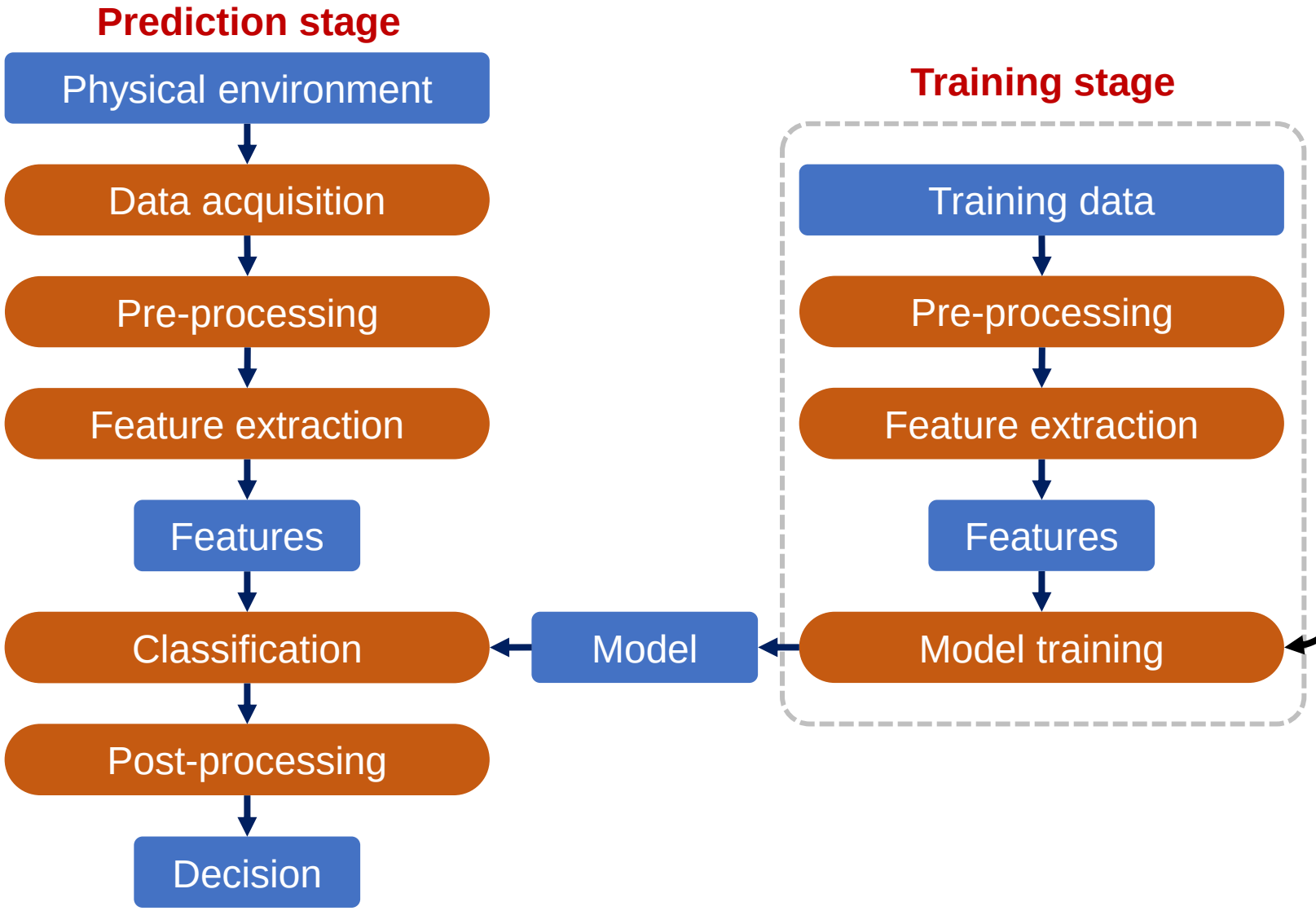
What are important for clustering?

- Selection of clustering strategy (Kmeans, DBscan) is task dependent
- Features
  - Avoid using unreliable features.
  - Be careful about correlations with existing features.
  - Be careful about measurement of distance.
  - Be careful about noise in the measurements.
- Definition of clustering parameters

Content

1. Introduction to pattern recognition (AI)
2. Clustering (Kmeans & DBscan)
3. **Classification (KNN & perceptron)**

Classification



Model maps feature representations to decision

What is KNN?

K Nearest Neighbor

K个

最邻近点

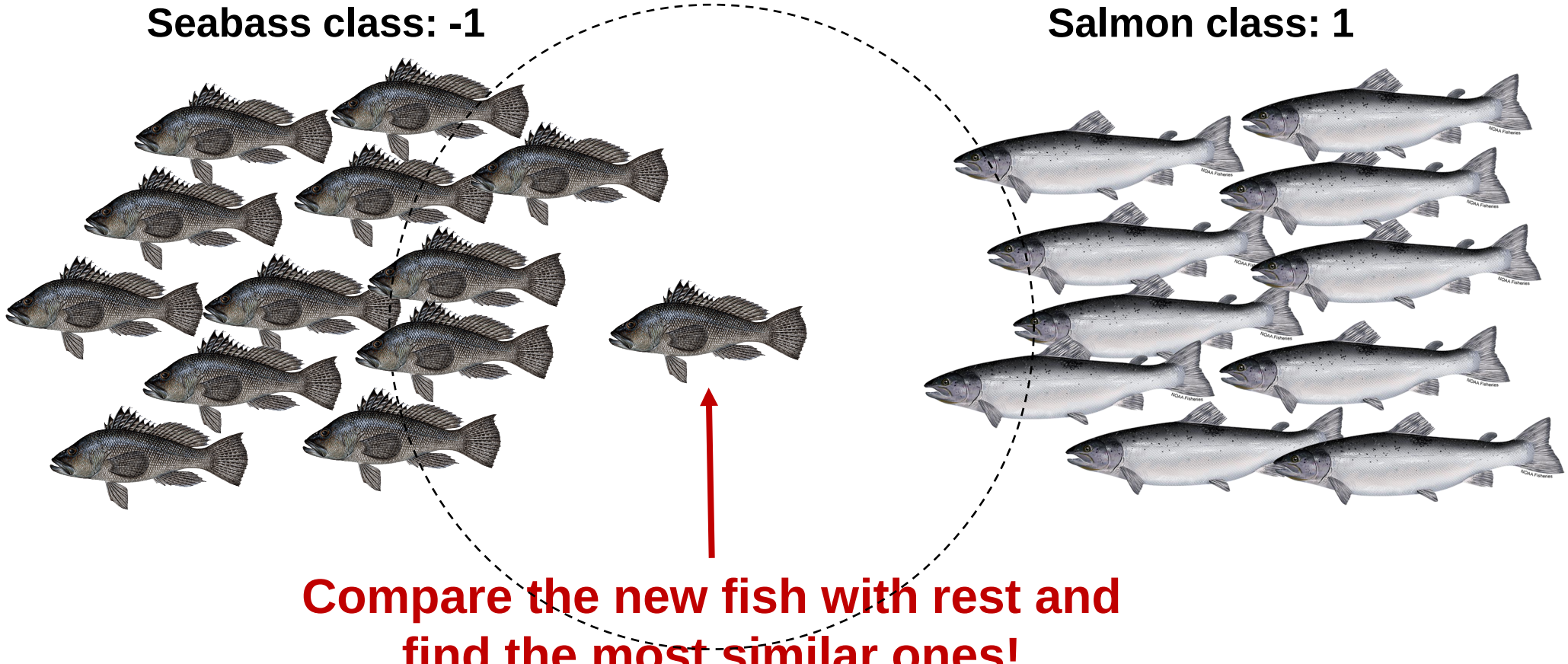
What is KNN?

K Nearest Neighbor

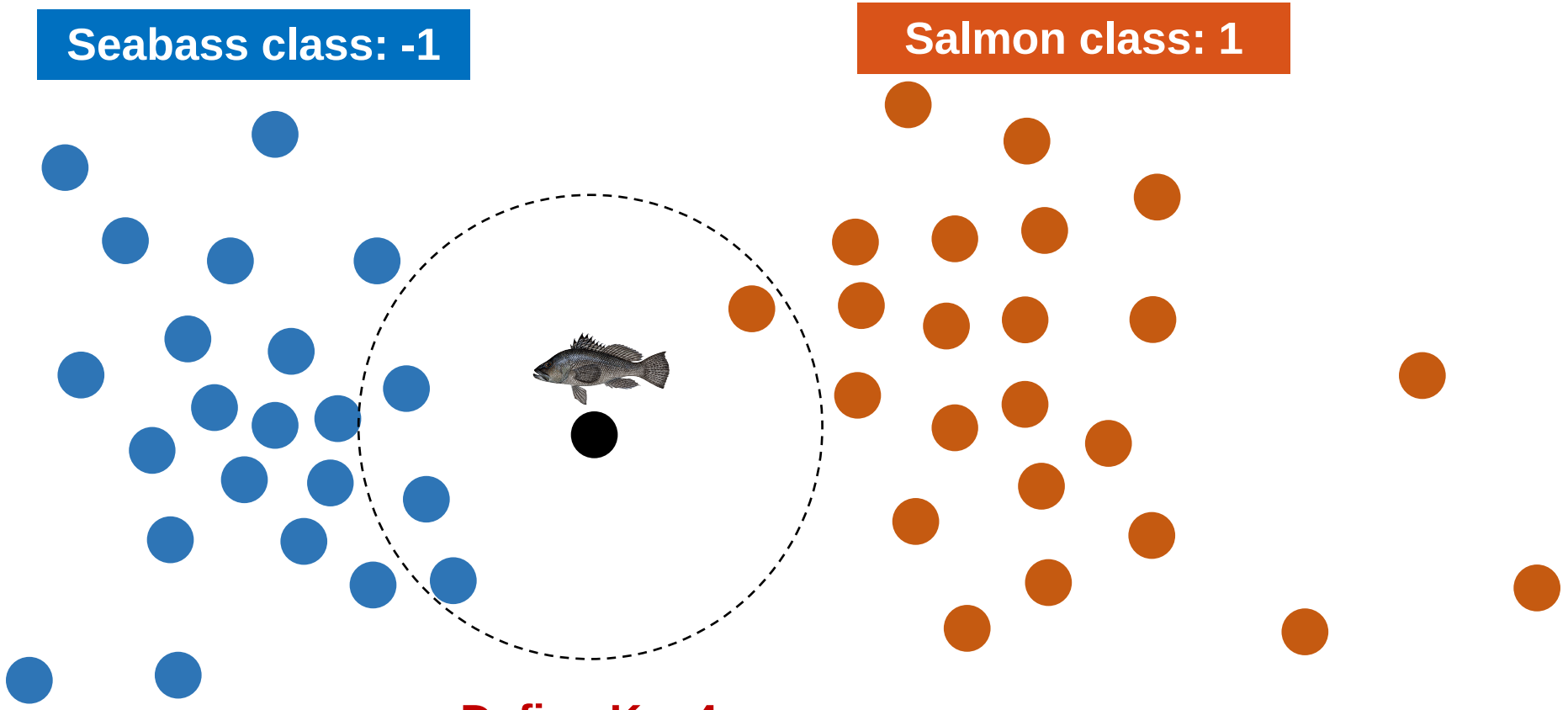
Majority of votes from neighborhood

What is KNN?

In our fish sorting case, seabass and salmon are collected as training samples

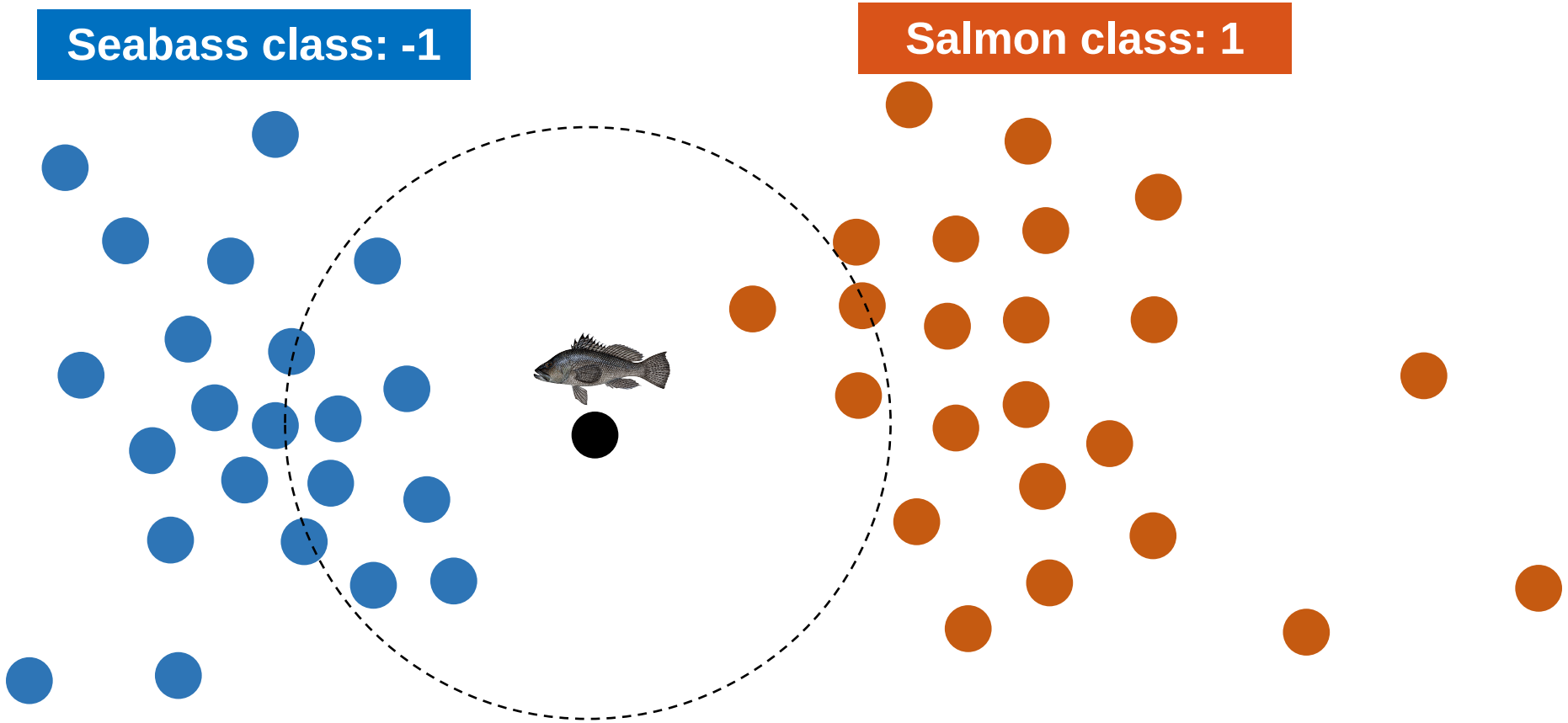


What is KNN?



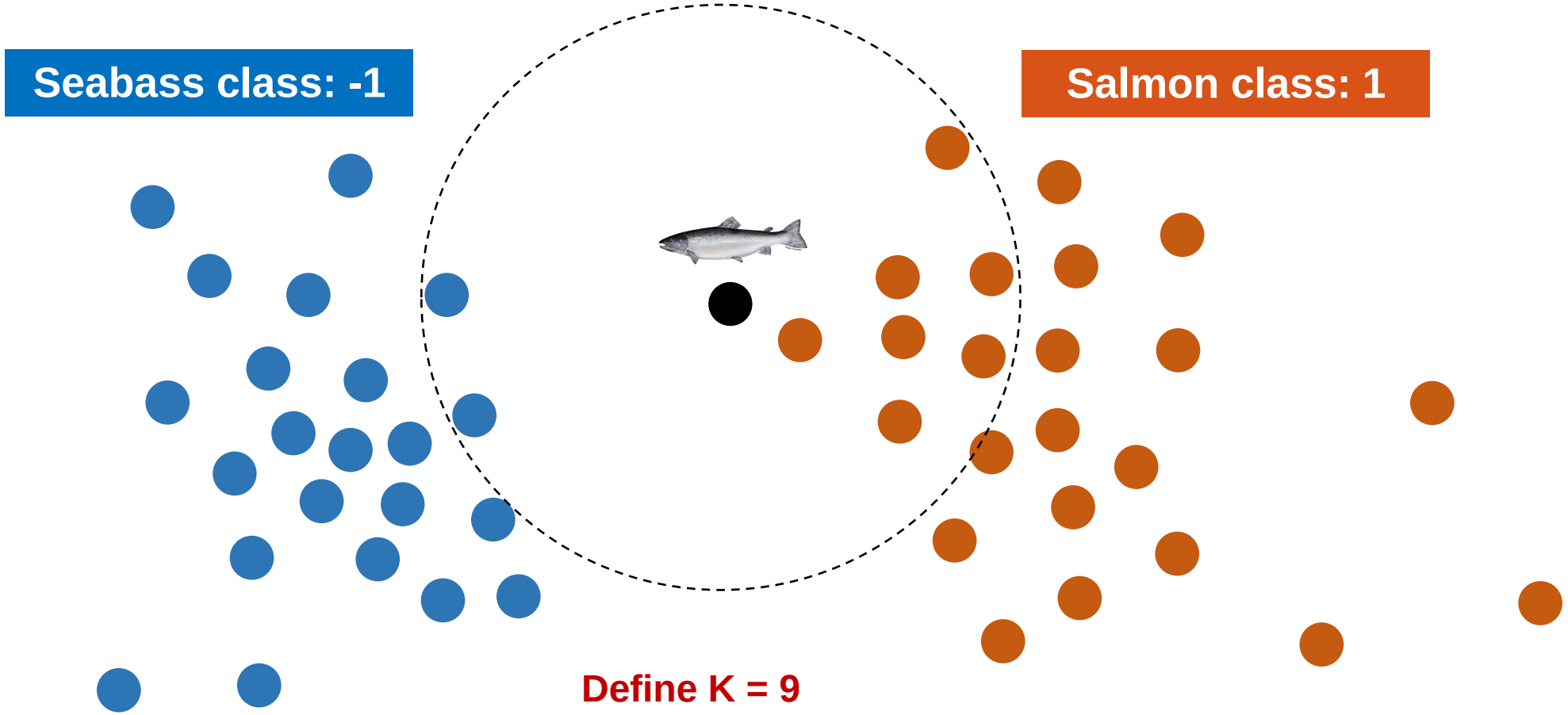
Define K = 4,
in 4 nearest neighbors, 3 from class -1, 1 from class 1

What is KNN?



Define K = 9,
in 9 nearest neighbors, 7 from class-1, 2 from class 1

What is KNN?



Define K = 9

in 9 nearest neighbors, 2 from class -1, 7 from class 1

KNN definition

KNN finds a number of K nearest neighbors in the space and use the majority of class to label the sample.

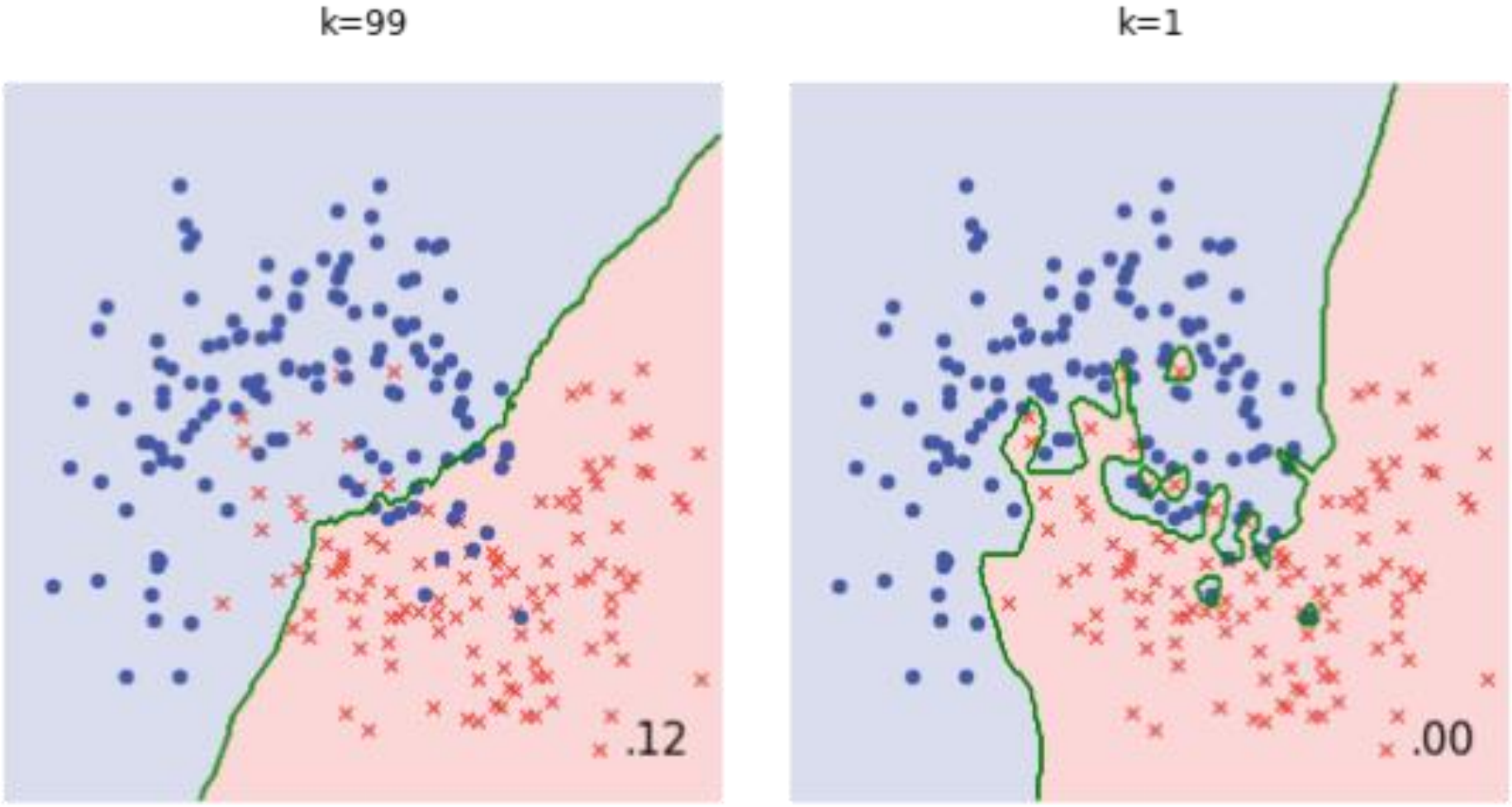
Input: samples

- Calculate the distance between sample and training samples
- Sort the distances and select the K nearest samples
- Use the majority class of K nearest samples to label the sample.

Output: class of sample

Effect of K

K is the ONLY parameter you need to define



Pros and cons of KNN

Pros

- Learning and implementation is extremely simple and intuitive
- Flexible decision boundaries (can be highly non-linear)

Cons

- Irrelevant or redundant features have negative impact
- Rely on the distance metric, cannot handle clusters with different densities

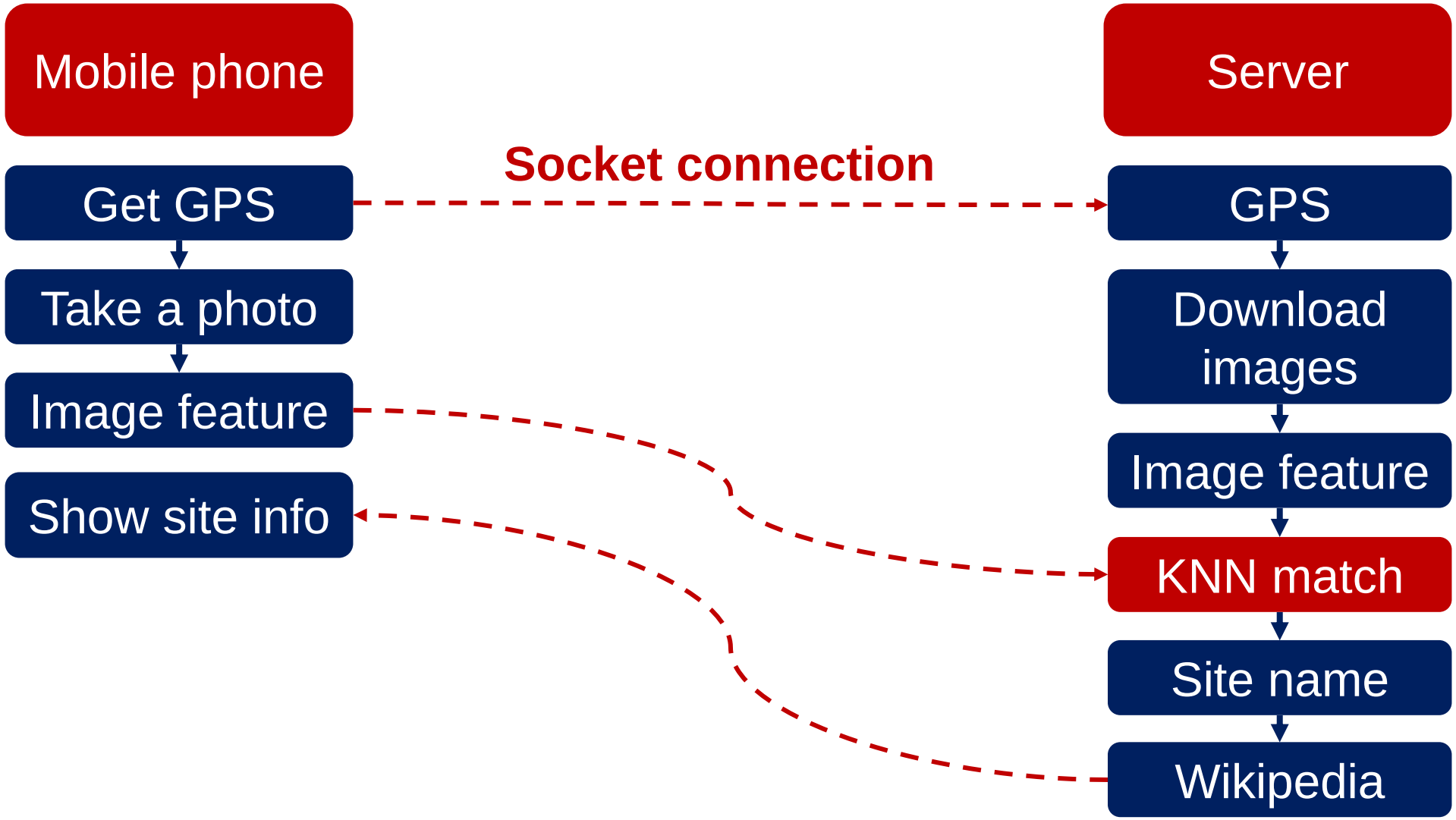
Applications of KNN

My MSc assignment 10 years ago in University of Amsterdam!

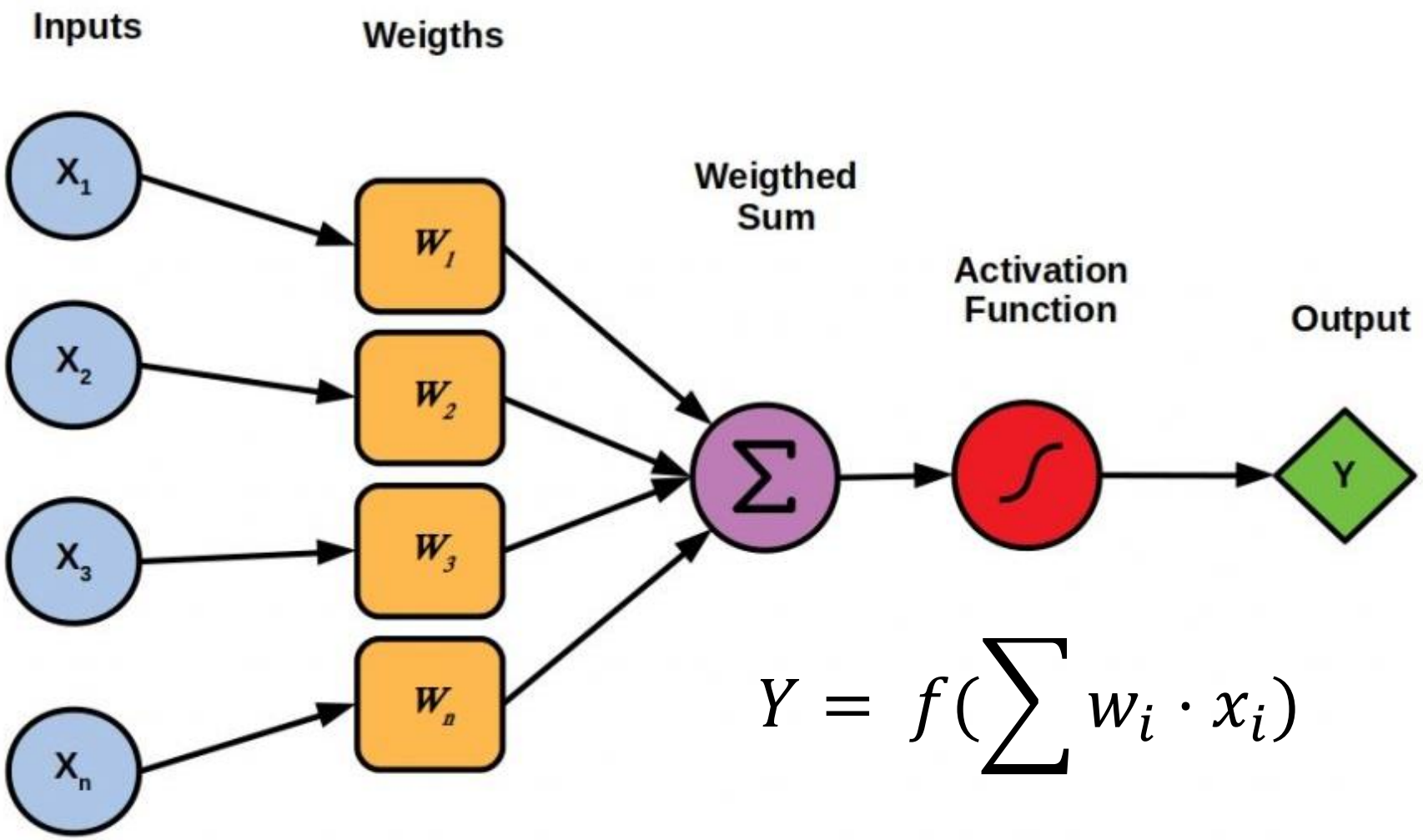
Develop an APP that search the information of an image taken by a phone!

KNN algorithm was implemented to search the image.

Applications of KNN

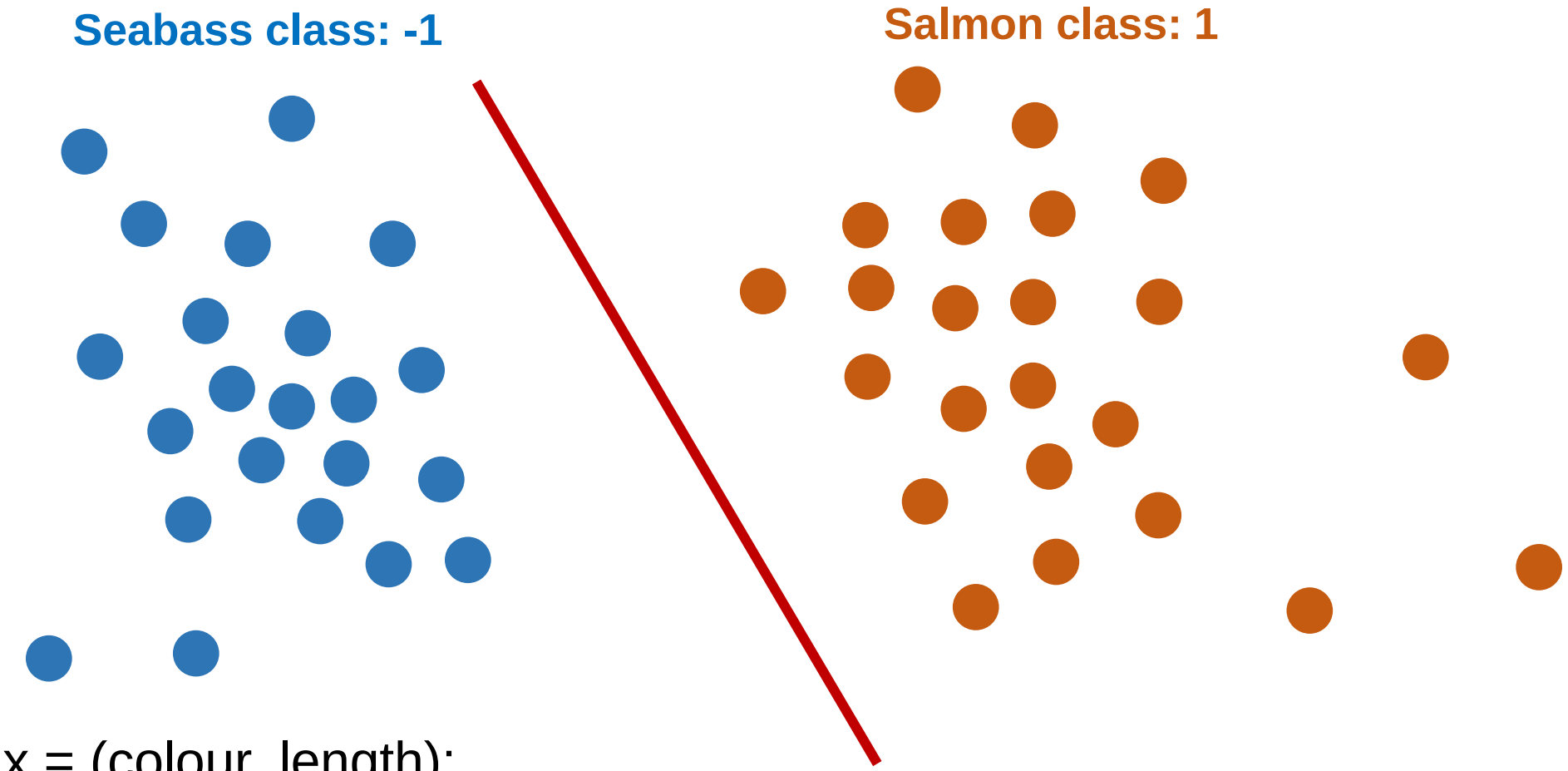


What is perceptron?



$$Y = f(\sum w_i \cdot x_i)$$

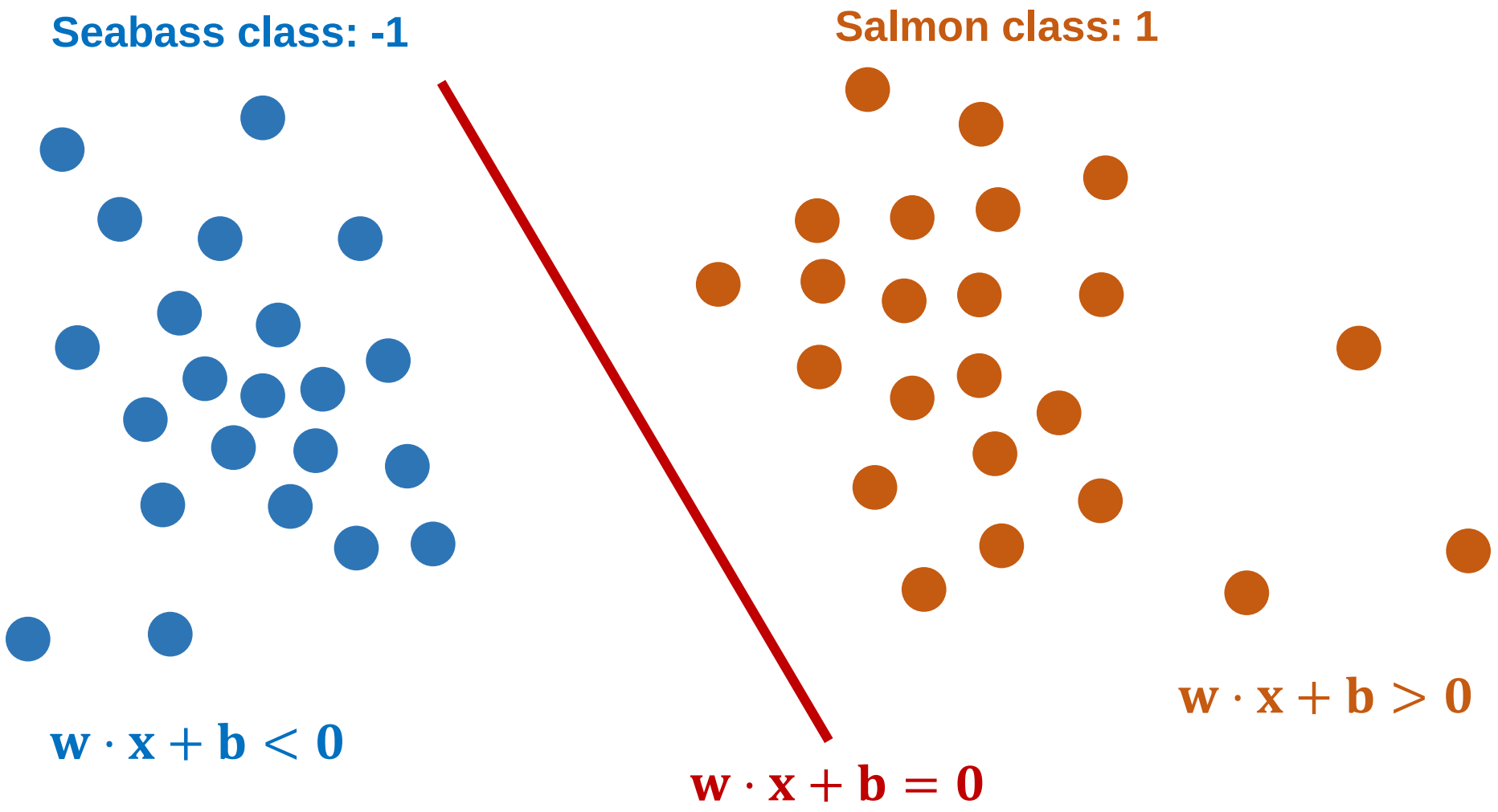
Problem definition in perceptron



x = (colour, length);
y = label; // -1 or 1

$$w \cdot x + b = 0$$

Problem definition in perceptron



Perceptron

Label:
y = {-1, 1}

Data:
x = (color, length)

$$f(x) = \text{sign}(w \cdot x + b)$$

Model coefficients

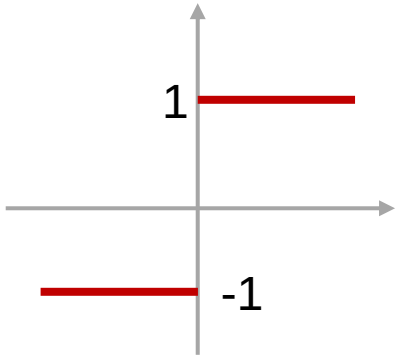
Perceptron

Label:
y = {-1, 1}

Data:
x = (color, length)

$$f(x) = \text{sign}(w \cdot x + b)$$

$$\textbf{sign}(x) = \begin{cases} +1, x \geq 0 \\ -1, x < 0 \end{cases}$$



Perceptron

$$f(x) = \text{sign}(w \cdot x + b)$$

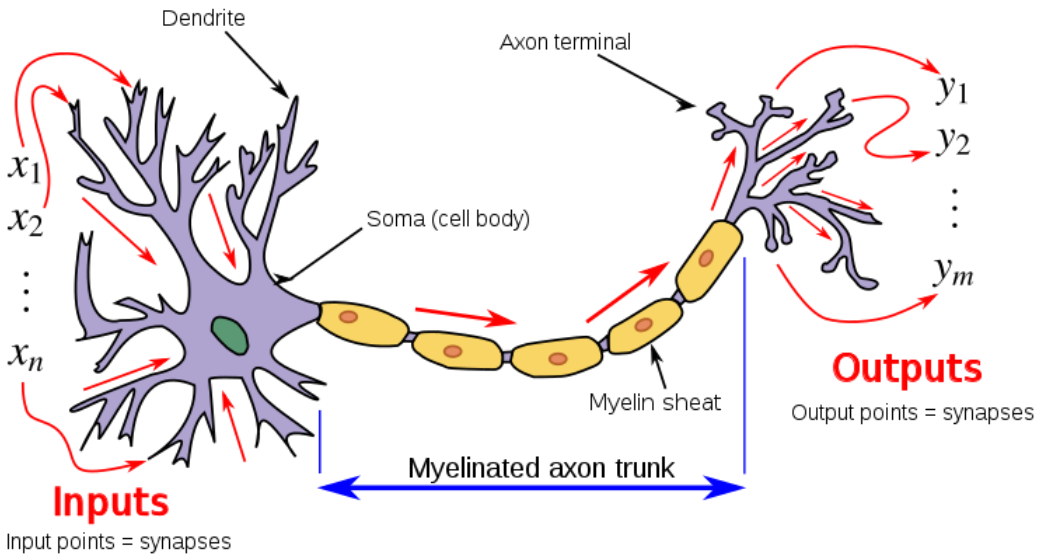
$$w_1 \cdot x_1 + w_2 \cdot x_2 + \cdots + w_n \cdot x_n + b \begin{cases} +1 \\ -1 \end{cases}$$

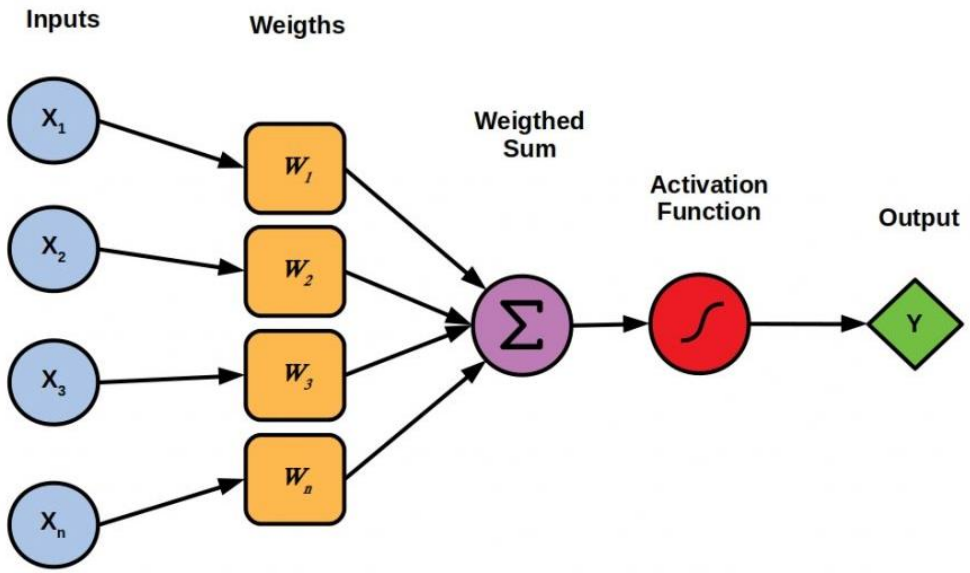
weights

bias

Perceptron

$$f(x) = \text{sign}(w \cdot x + b)$$





Perceptron

$$f(x) = \text{sign}(w \cdot x + b)$$

Decision

From sensor

How to get model parameters???

Train it!!!

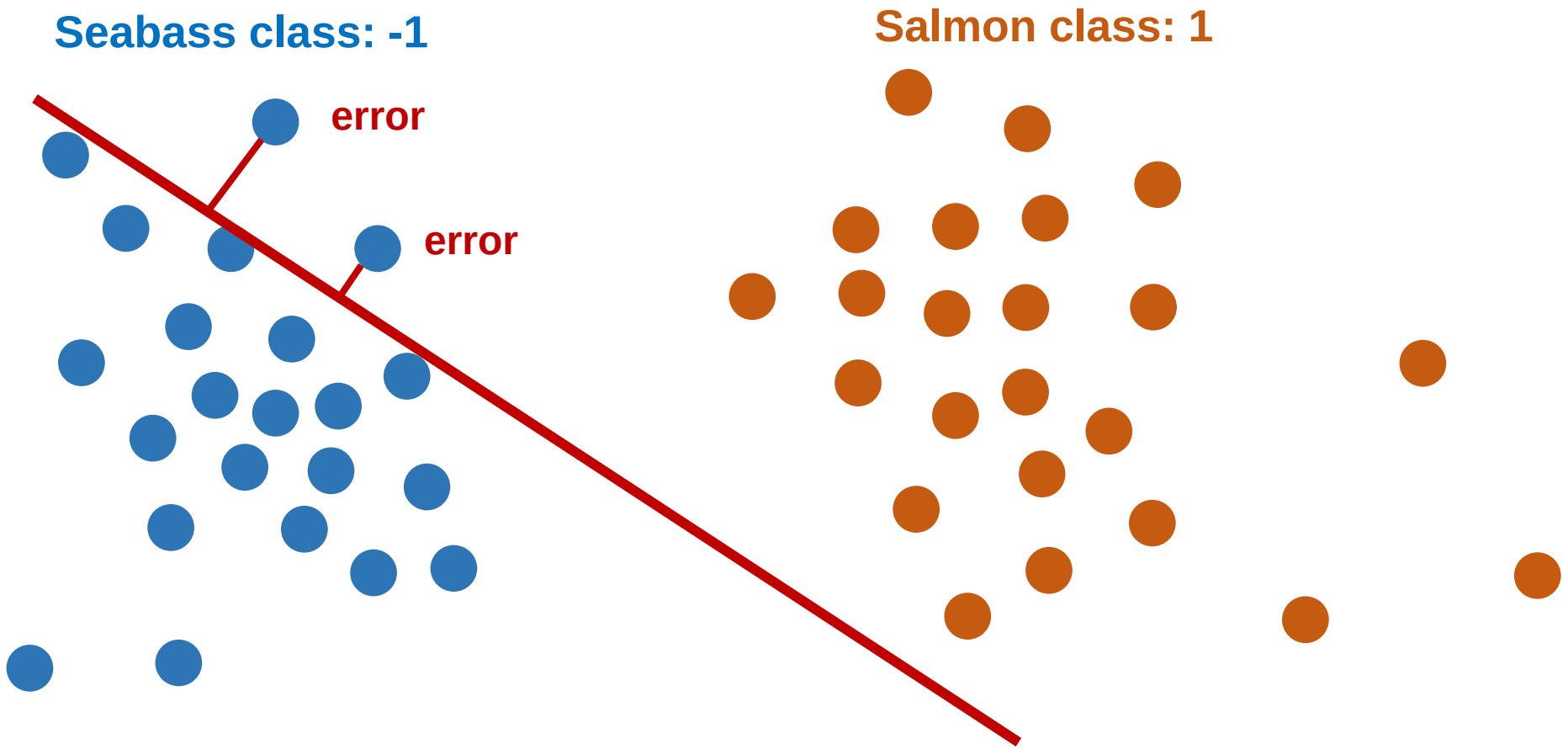
Perceptron

How to train a perceptron?

Minimize the error in classification!

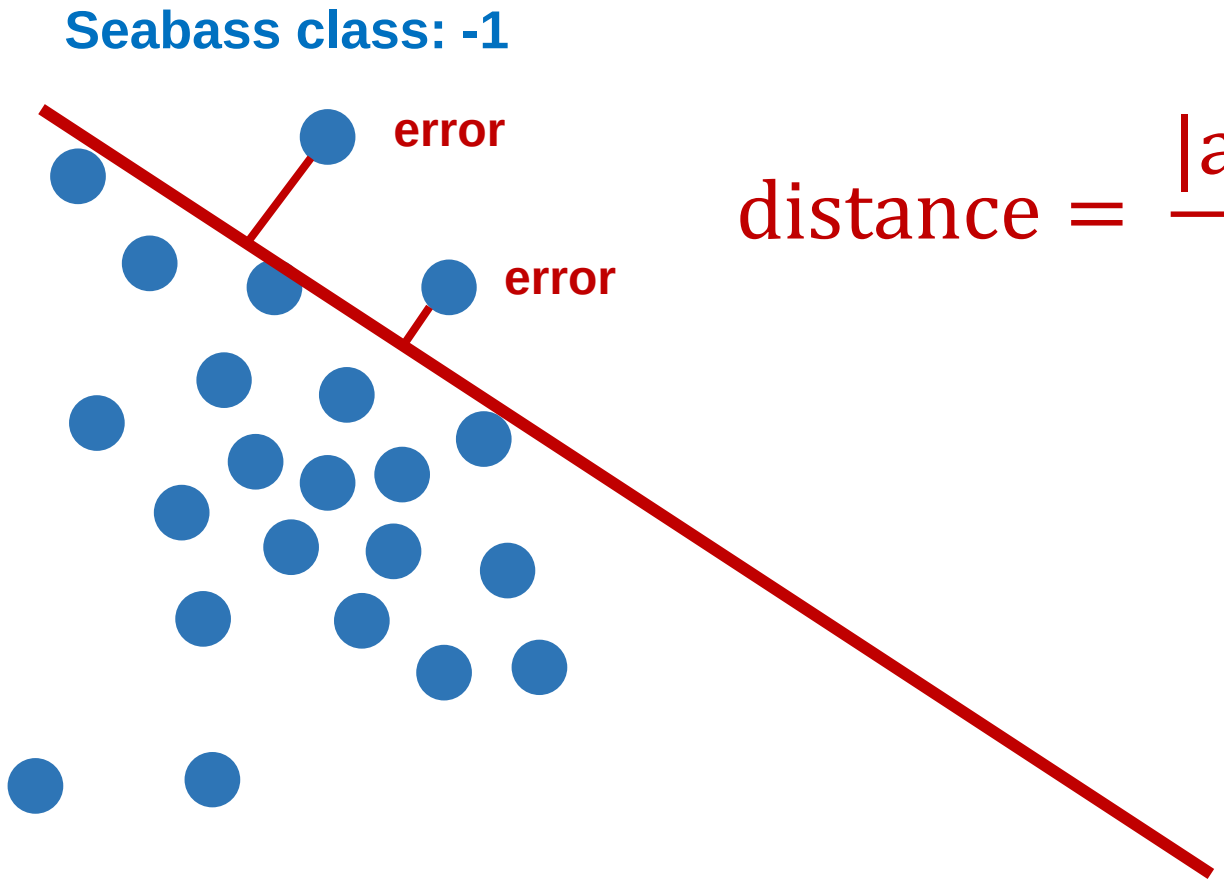
Perceptron

Minimize the error in classification



Perceptron

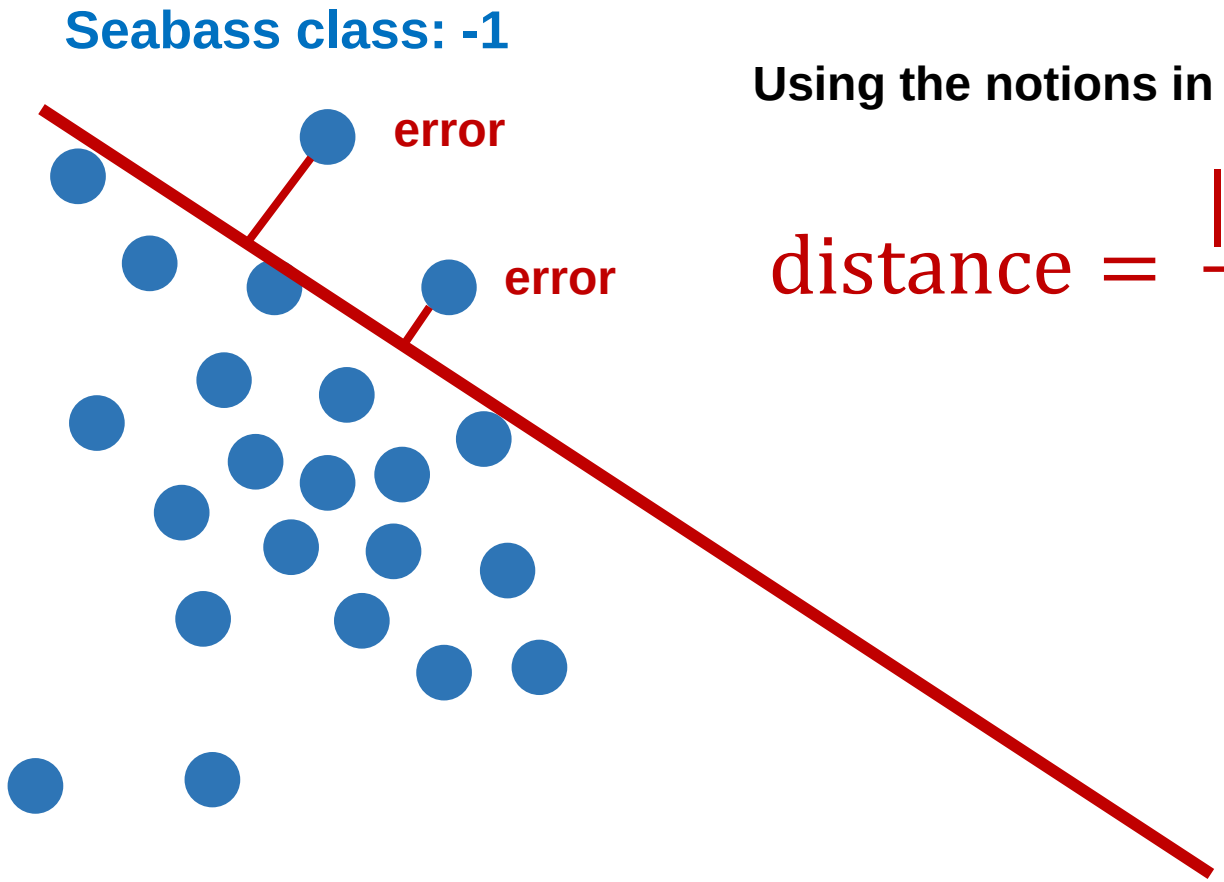
Minimize the error in classification



$$\text{distance} = \frac{|a \cdot x + b \cdot y + c|}{\sqrt{a^2 + b^2}}$$

Perceptron

Minimize the error in classification

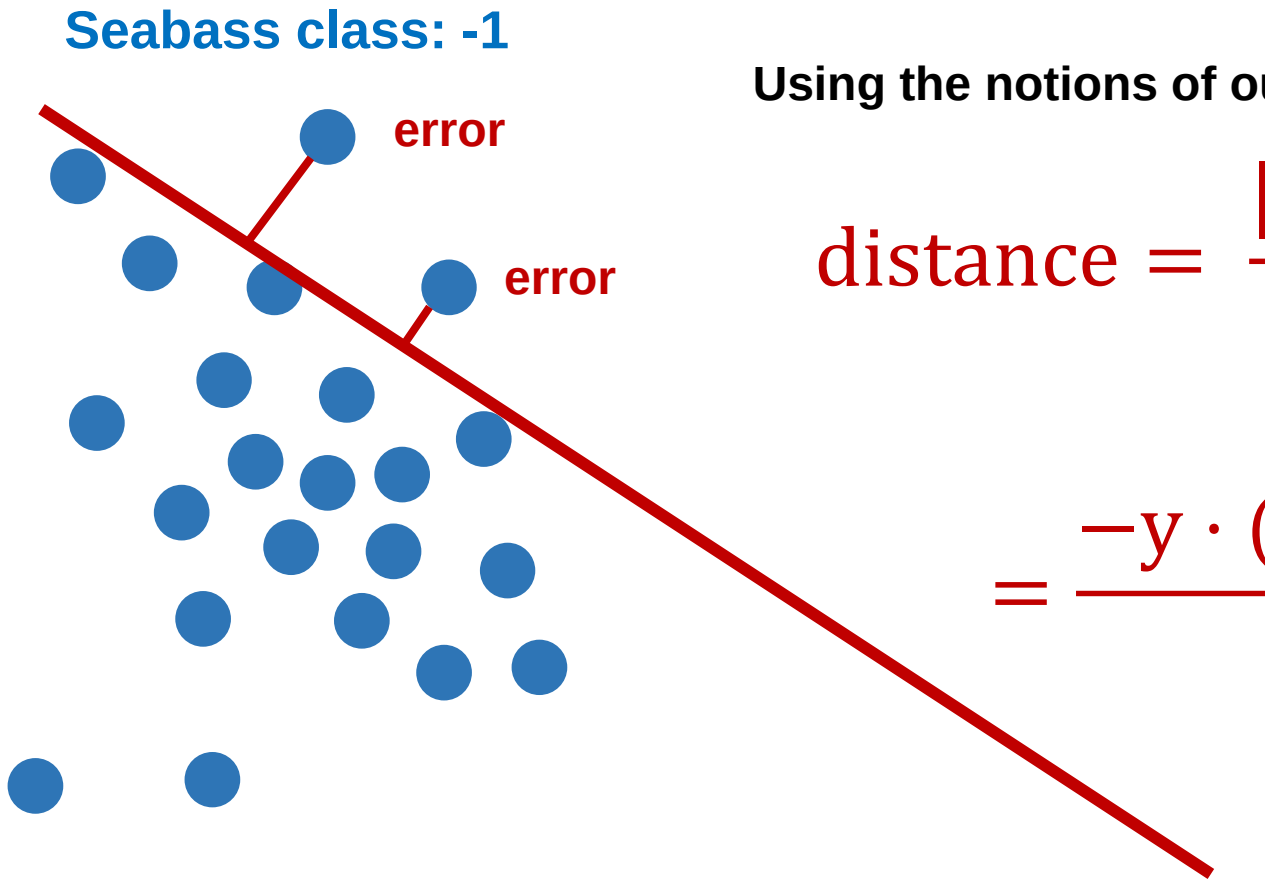


Using the notions in our task:

$$\text{distance} = \frac{|w_1 \cdot x_1 + w_2 \cdot x_2 + b|}{\sqrt{w_1^2 + w_2^2}}$$

Perceptron

Minimize the error in classification



Using the notions of our task:

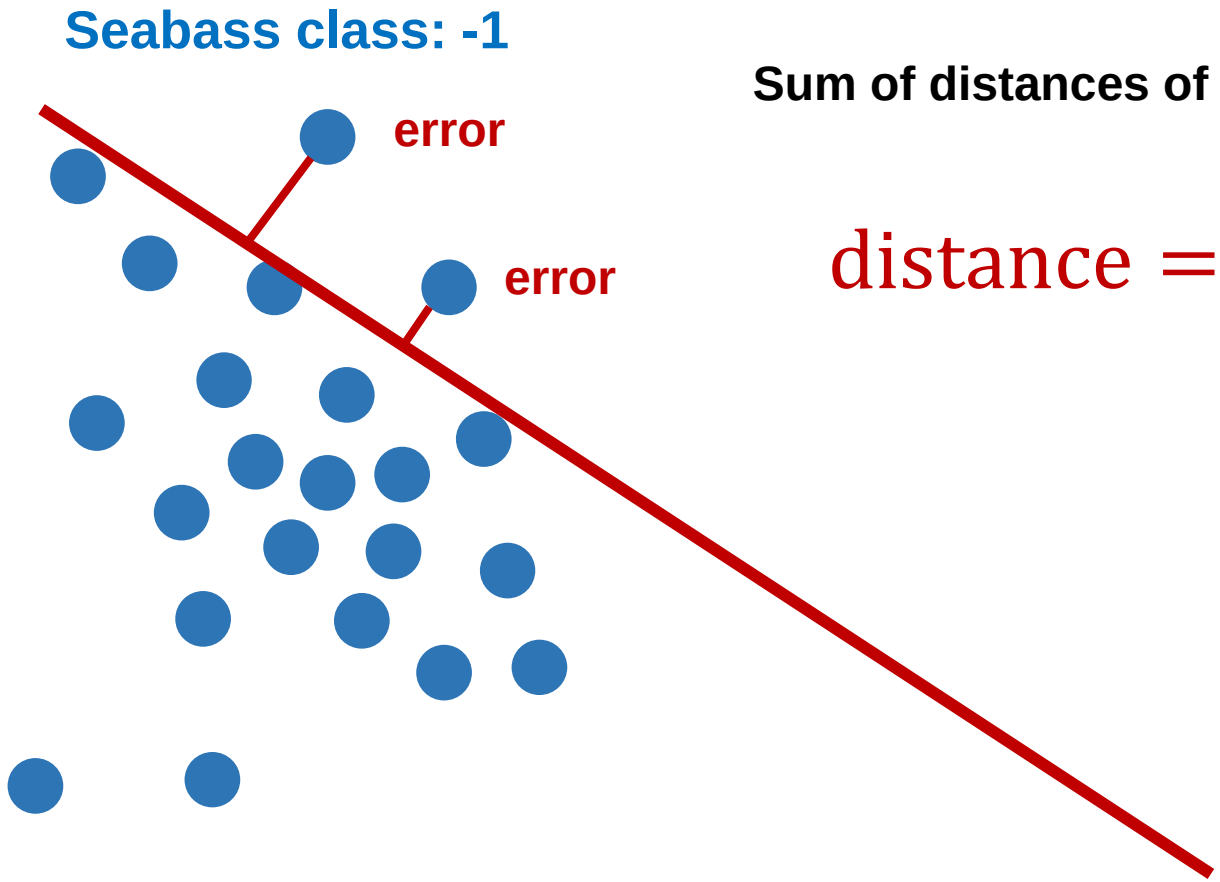
$$\text{distance} = \frac{|w \cdot x + b|}{||w||}$$

$$= \frac{-y \cdot (w \cdot x + b)}{||w||}$$

L2 norm

Perceptron

Minimize the error in classification



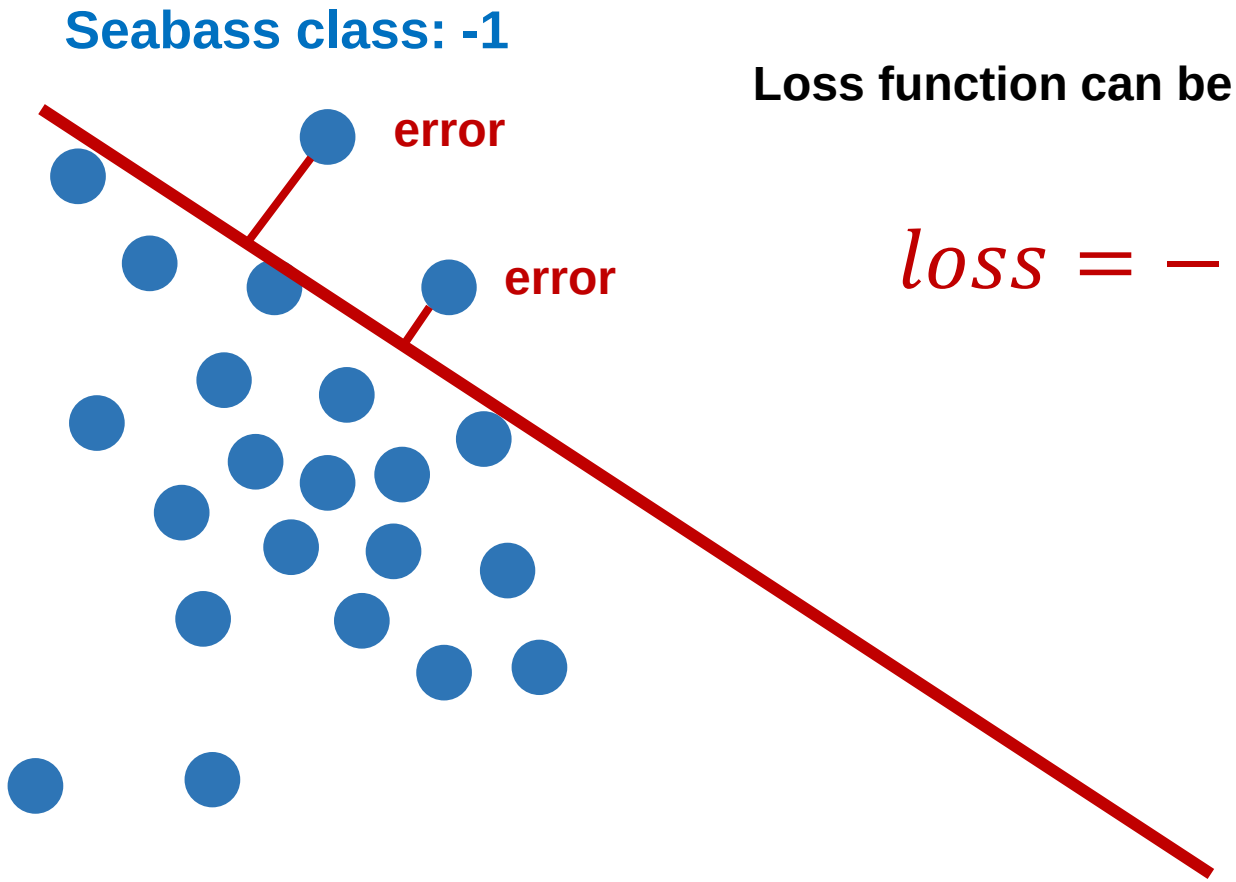
Sum of distances of all errors

$$\text{distance} = \sum \frac{-\text{y} \cdot (w \cdot x + b)}{||w||}$$

Norm is positive, no impact on results

Perceptron

Minimize the error in classification



Loss function can be defined as:

$$loss = -\sum y \cdot (w \cdot x + b)$$

Perceptron

The goal is to find **w and b** that minimize loss

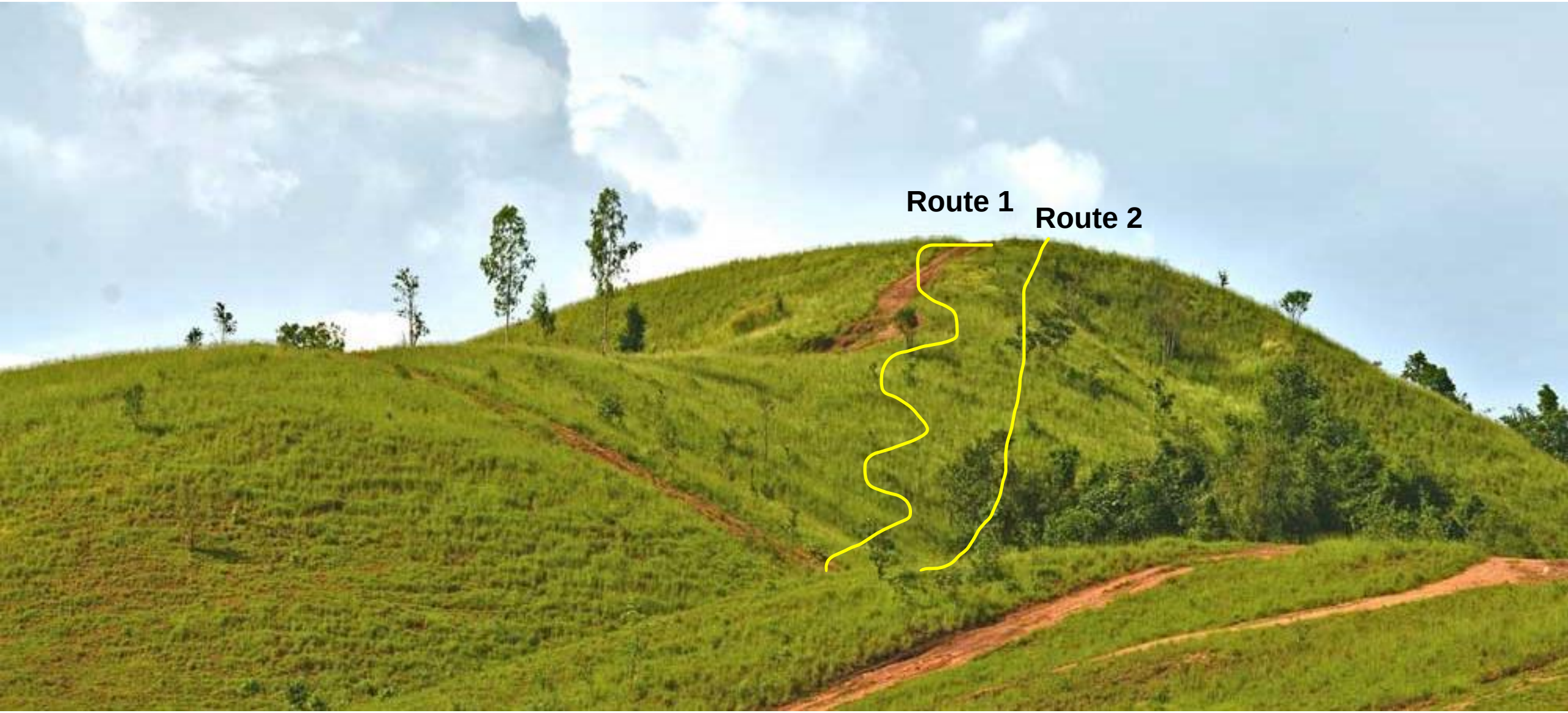
$$loss = -\sum y \cdot (w \cdot x + b)$$

损失函数

Training label

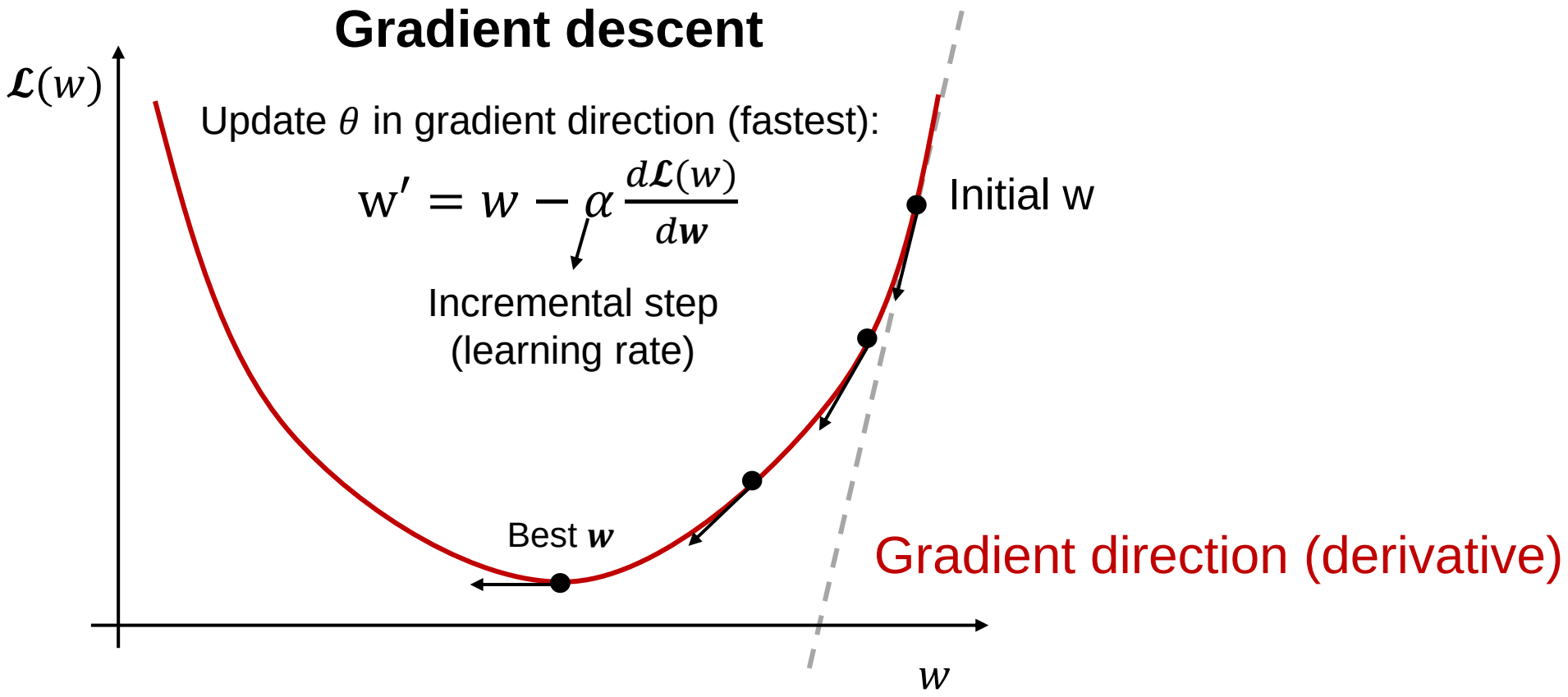
Training data

Gradient descent optimization



Gradient descent optimization

Update the coefficients along the gradient direction of loss function is the fastest path to achieve the minimum!!!



Perceptron

The gradient of loss is calculated by (partial) derivative

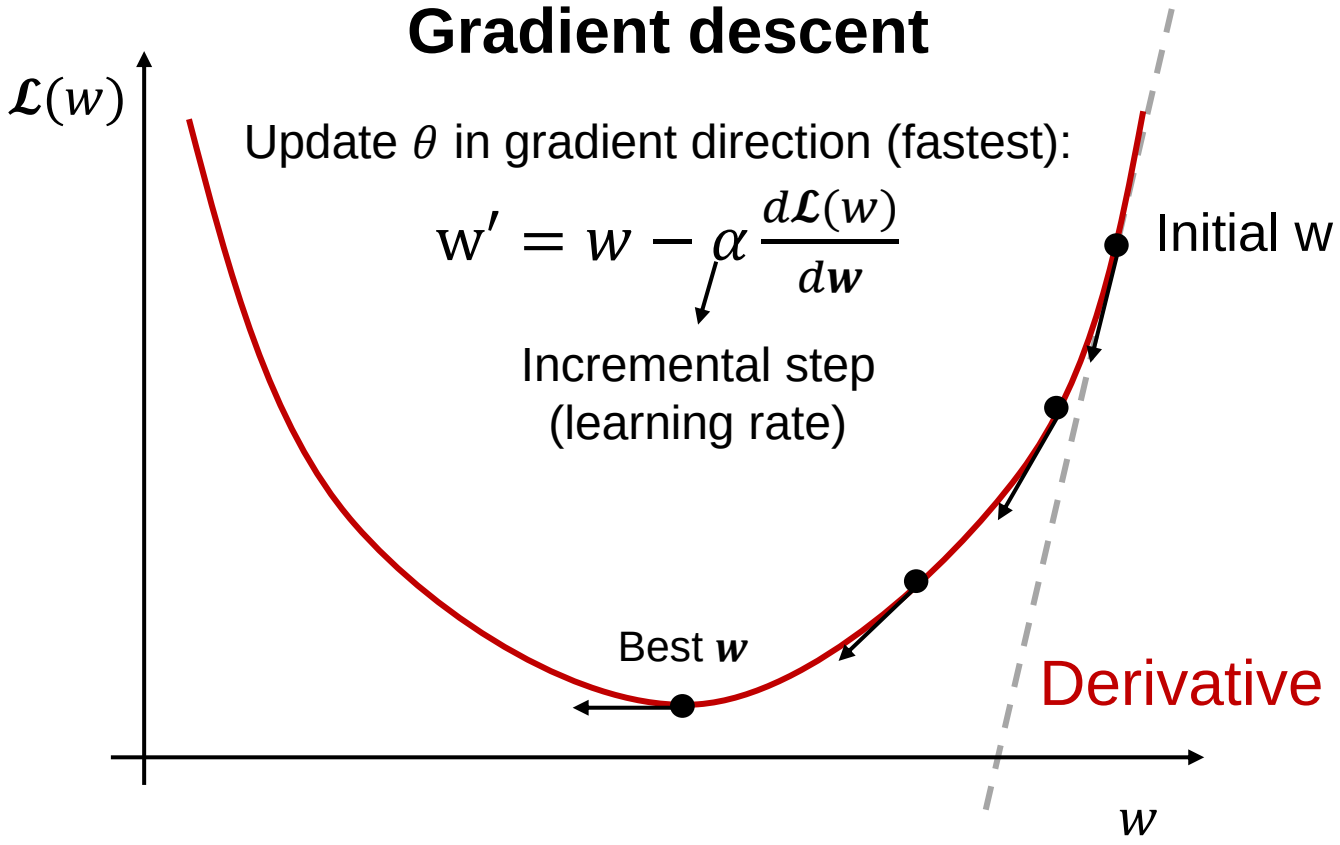
$$\boldsymbol{loss} = -\sum \boldsymbol{y} \cdot (\boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{b})$$

$$\nabla_w L(w, b) = -\sum y_i \cdot x_i$$

$$\nabla_b L(w, b) = -\sum y_i$$

Perceptron

Update the coefficients step-wise along the gradient direction



$$w \leftarrow w + \alpha \cdot \nabla_{\mathrm{w}} \mathrm{L}(\mathrm{w}, \mathrm{b})$$

$$b \leftarrow b + \alpha \cdot \nabla_{\mathrm{b}} \mathrm{L}(\mathrm{w}, \mathrm{b})$$

α $(0 < \alpha < 1)$ learning rate

Overview

感知机算法

输入：训练集T={(x1,l1),(x2,l2),(x3,l3)}，x是特征，l是标签；学习率a(0<a<1)

- 选择模型y=sign(wx+b),初始化w和b

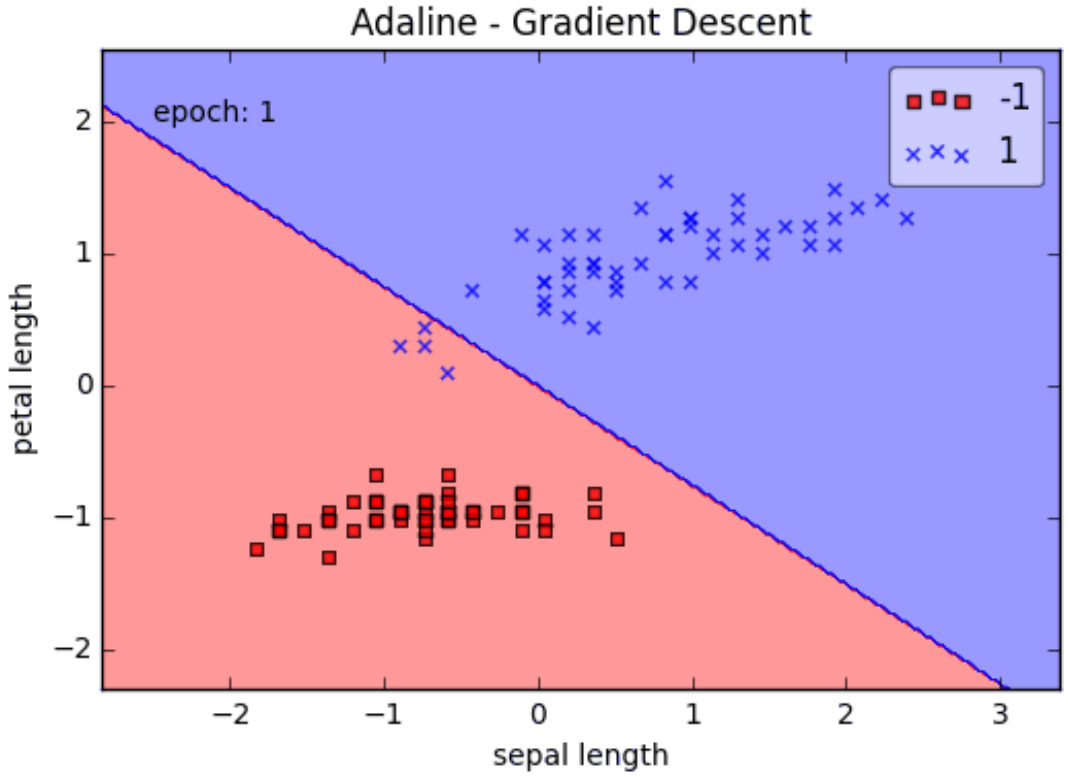
while (true)

{

- 计算每个数据的损失函数
- If y(wx+b) <= 0
  - 更新w: w = w + ayx
  - 更新b：b = b + ay
- If w 和 b停止更新（或很小）

  break;

输出：模型参数w和b



Pros and cons of perceptron

Pros:

- Simplicity
- Generally applicable for many tasks

Cons:

- Data must be linearly separable
- No guarantee on non-convex problem

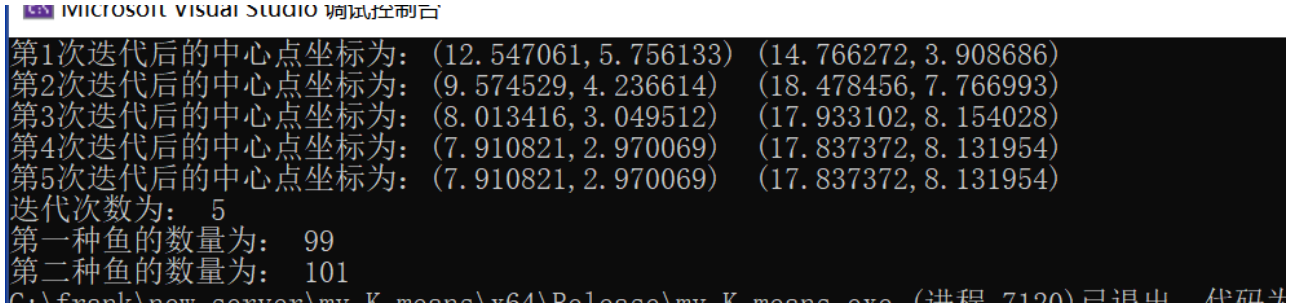
Summary

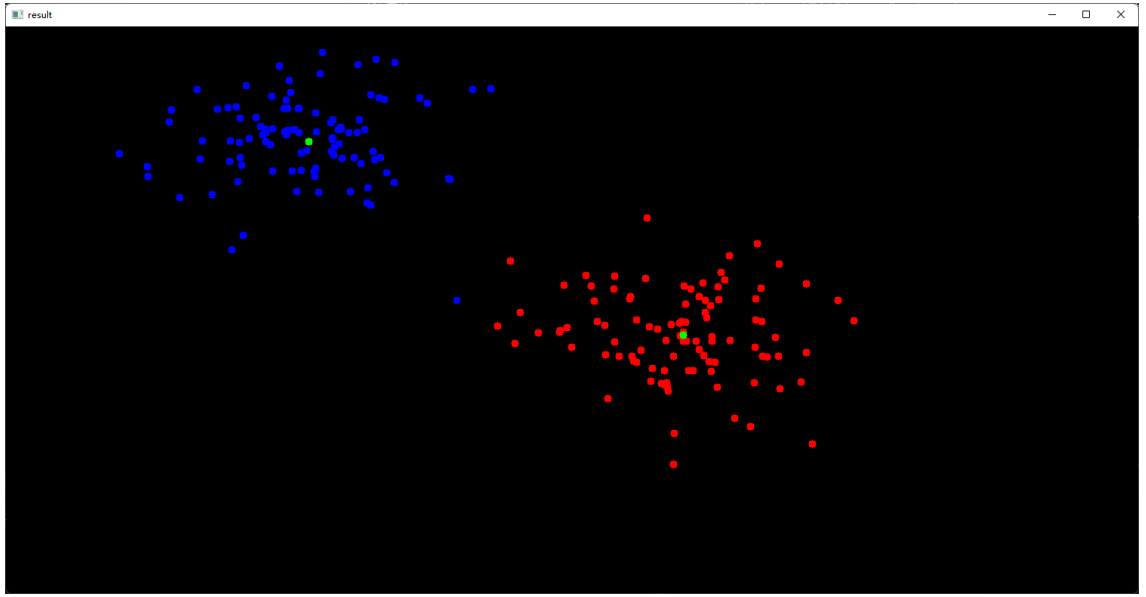
- **Pattern recognition** is the key to make machine intelligent
- Two major types of pattern recognition algorithms: **clustering** and **classification**
- Two representative clustering algorithms: **Kmeans** and **DBscan**
- Two representative classification algorithms: **KNN (non-linear)** and **perceptron (linear)**
- Each method has pros and cons, there is no optimal approach.
- Time to write your first Kmeans in C!!!

Assignment

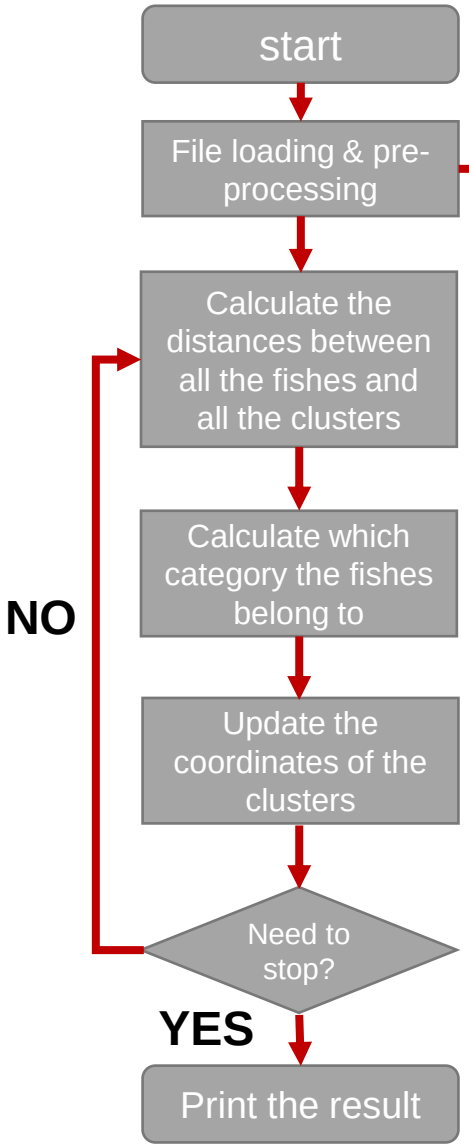
Use K-means to classify salmon and seabass based on their length and color: you will be provided with a csv file which contains the length and color of 200 fishes. Use C to implement K-means and print how many seabass and salmons we have.

a) The csv file has been uploaded on bb
b) If you don't know how to read a csv file, you can refer to the answers to the previous lesson assignment
c) If you want to plot the result in an image you can install OpenCV, I'll put the installation method behind.





Assignment



Read the csv file and convert ASCII data into float

Initialize the coordinate of clusters

Set some parameters(the number of clusters is 2; the number of fishes is 200)

400 distances in total

If a fish is closer to the salmon cluster,
it belongs to the cluster of salmon.

The coordinate of a cluster is the average of the coordinates of all the fishes belonging to this cluster.

If the coordinates of clusters do not have further changes in the loop, iterations can be stop.

We can only separate the fishes into two clusters by K-means.
But we don't know which cluster is salmon, unless you have prior knowledge.

Assignment

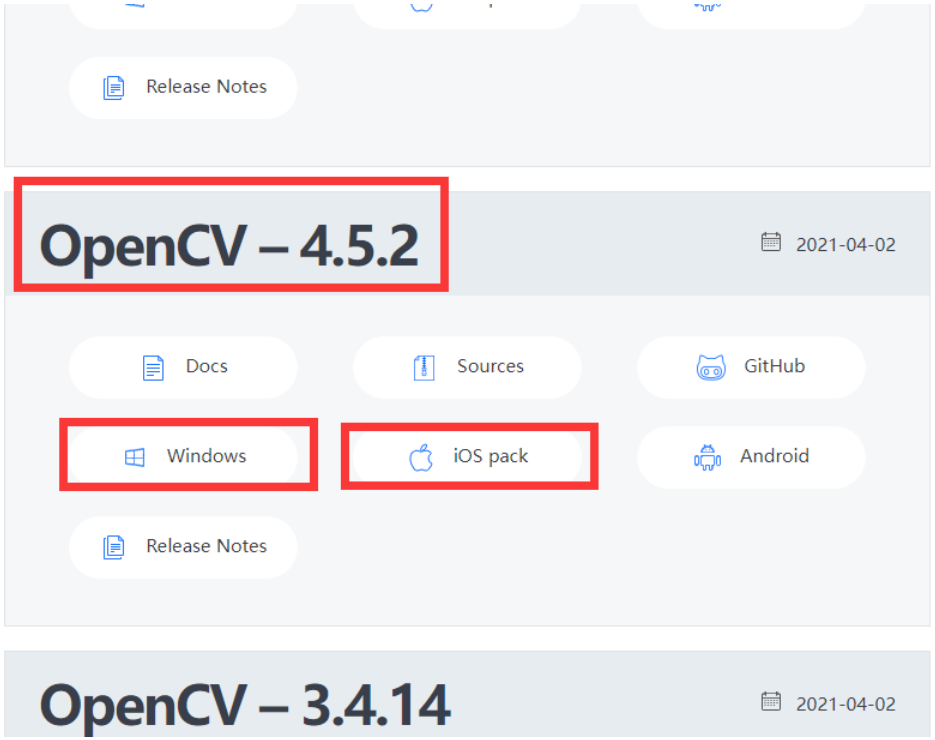
You can use this function (needs OpenCV, see following) to plot the result.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <opencv2/opencv.hpp>

typedef enum {seabass,salmon}category;

typedef struct
{
float x;
float y;
}coordinate;

typedef struct
{
coordinate features;
category cluster;
float distance[K];
}fish;
```

```
void show_result(fish* fishes, coordinate* centers)
{
cv::Mat result = cv::Mat(750, 1500, CV_8UC3, cv::Scalar(0, 0, 0));
for (int i = 0; i < N; i++)
{
cv::Point2f fish_point = cv::Point2f(fishes[i].features.x * 50, fishes[i].features.y * 50);
if (fishes[i].cluster == type1)
    cv::circle(result, fish_point, 5, cv::Scalar(255, 0, 0), -1);
else
cv::circle(result, fish_point, 5, cv::Scalar(0, 0, 255), -1);
}

for (int i = 0; i < K; i++)
{
cv::Point2f center_point = cv::Point2f(centers[i].x * 50, centers[i].y * 50);
cv::circle(result, center_point, 5, cv::Scalar(0, 255, 0), -1);
}

imshow("result", result);

cv::waitKey(0);
}
```

Install OpenCV on VS 2022

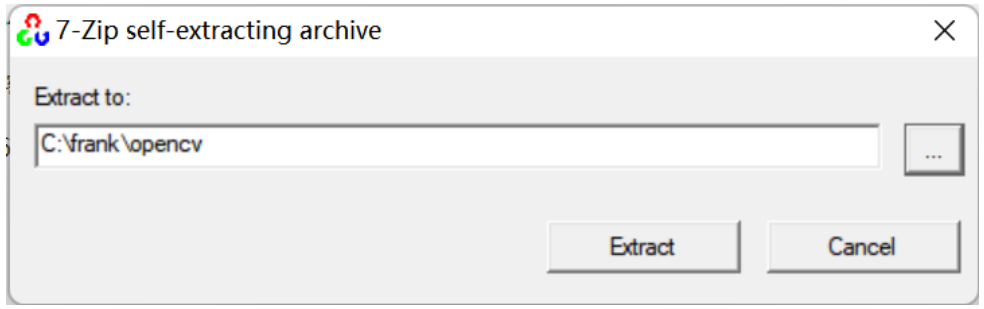
点击这个链接，到opencv官网下载 [Releases - OpenCV]

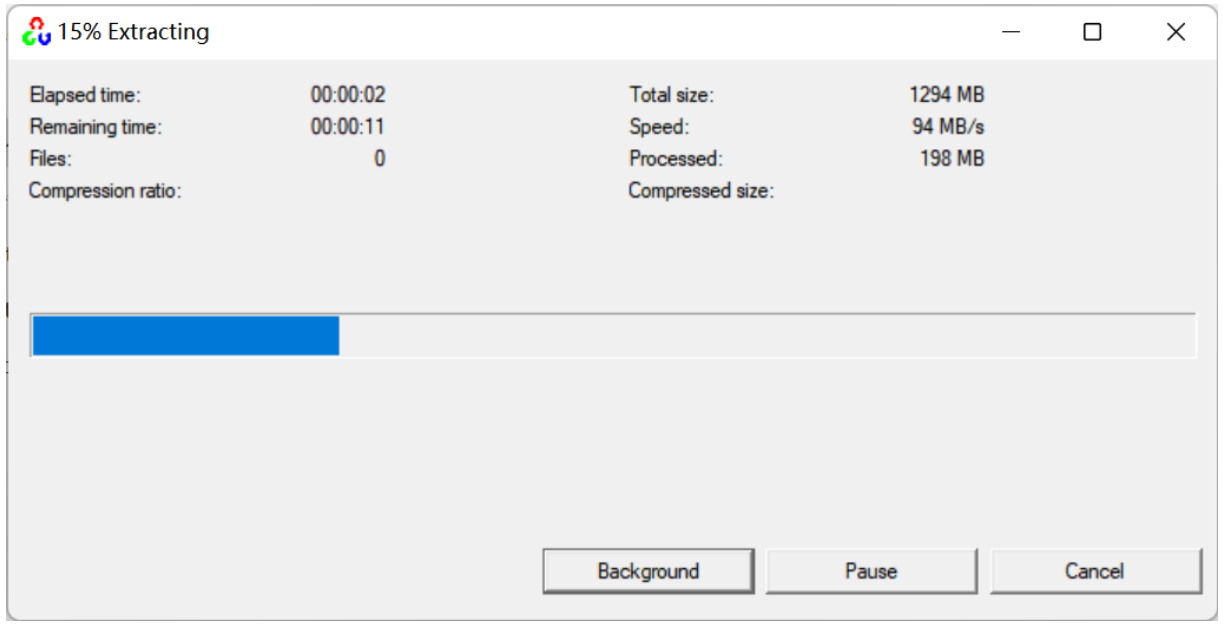


推荐4. 5. 2版本

根据自己的系统选择安装包

Install OpenCV on VS 2022



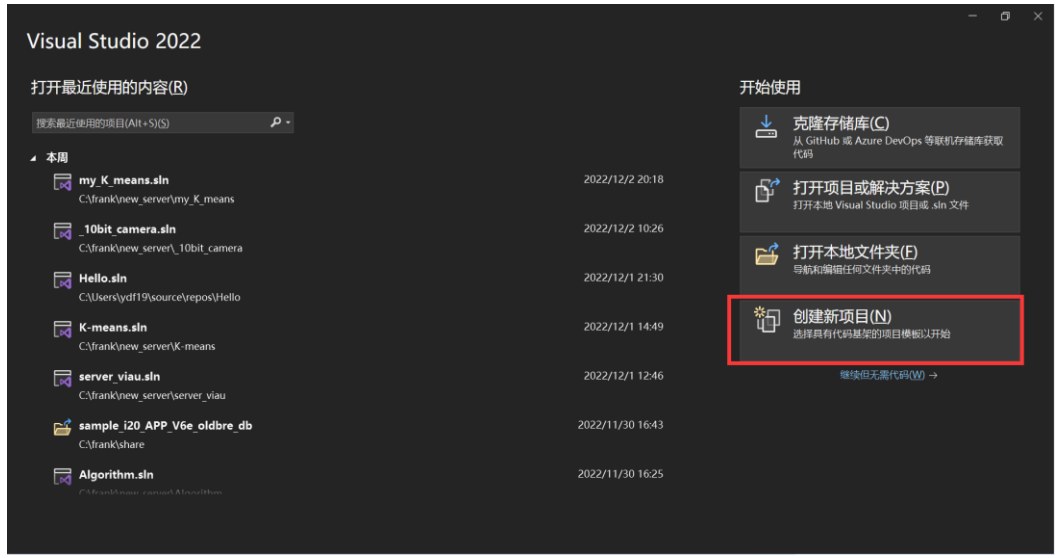


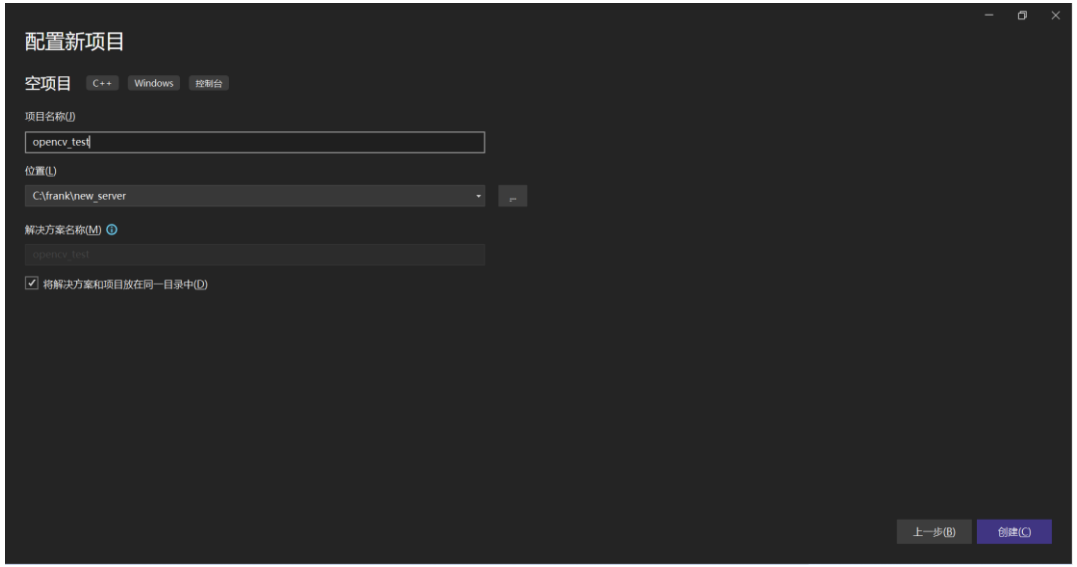
下载完是一个exe文件

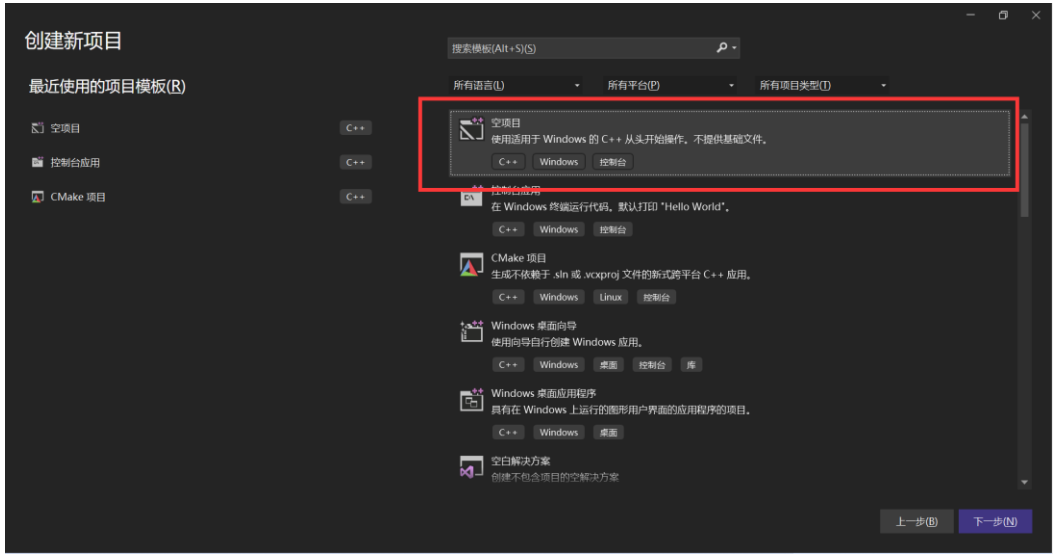
双击后选择一个不包含中文的路径

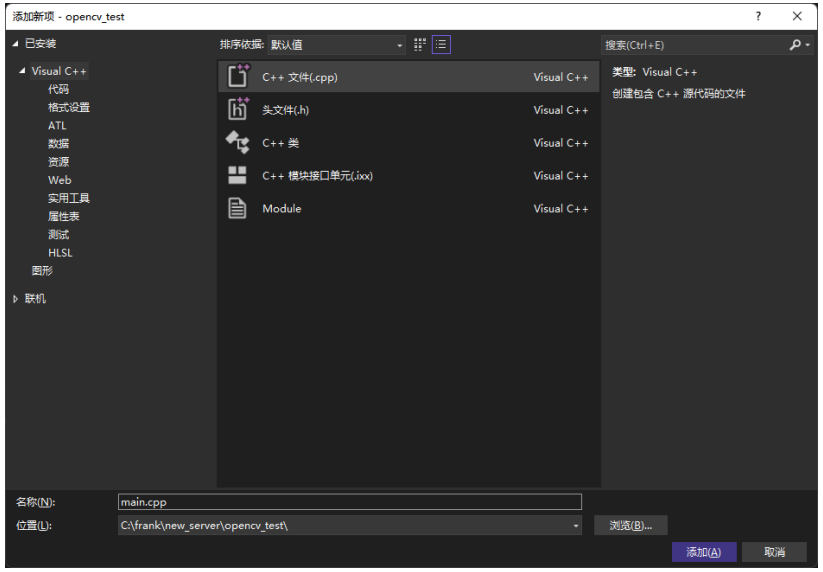
然后像以前一样新建一个工程

Install OpenCV on VS 2022

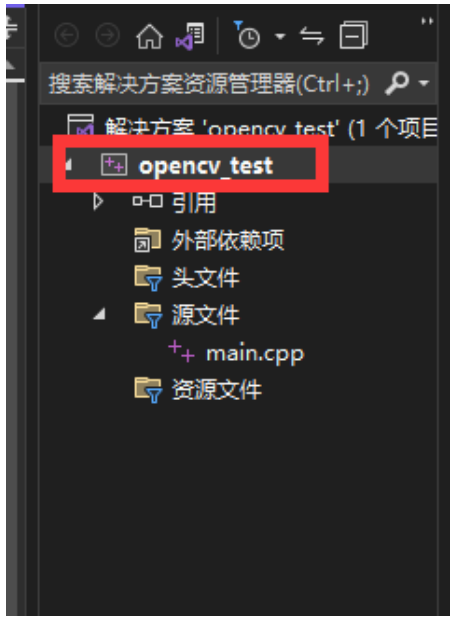






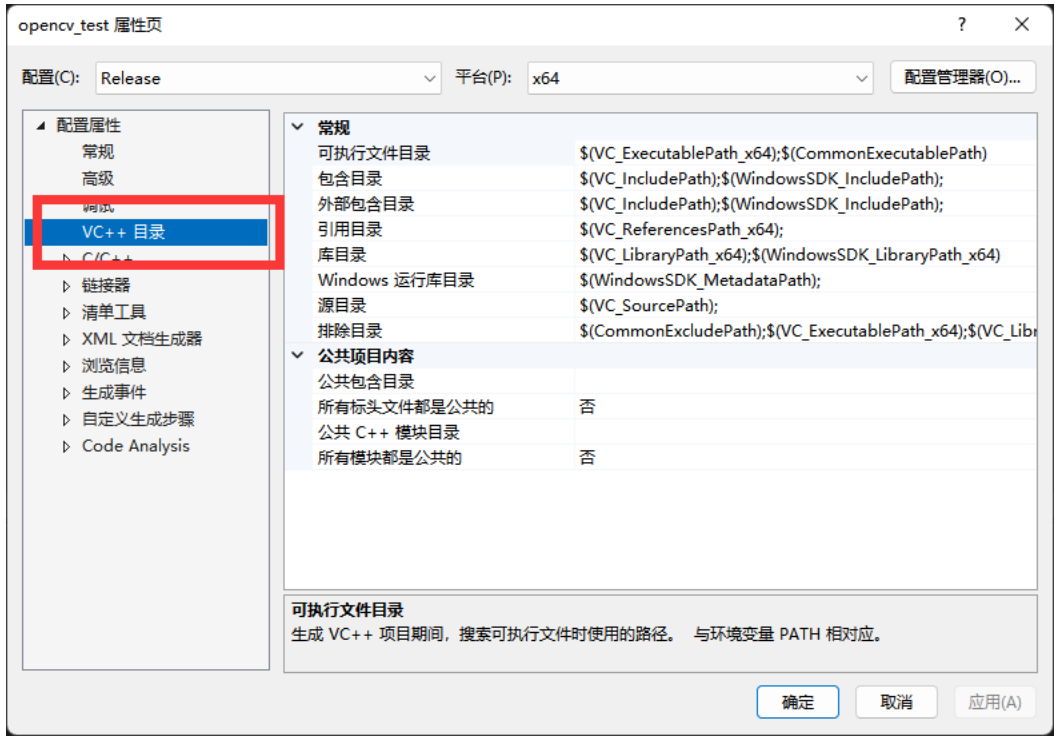


Install OpenCV on VS 2022



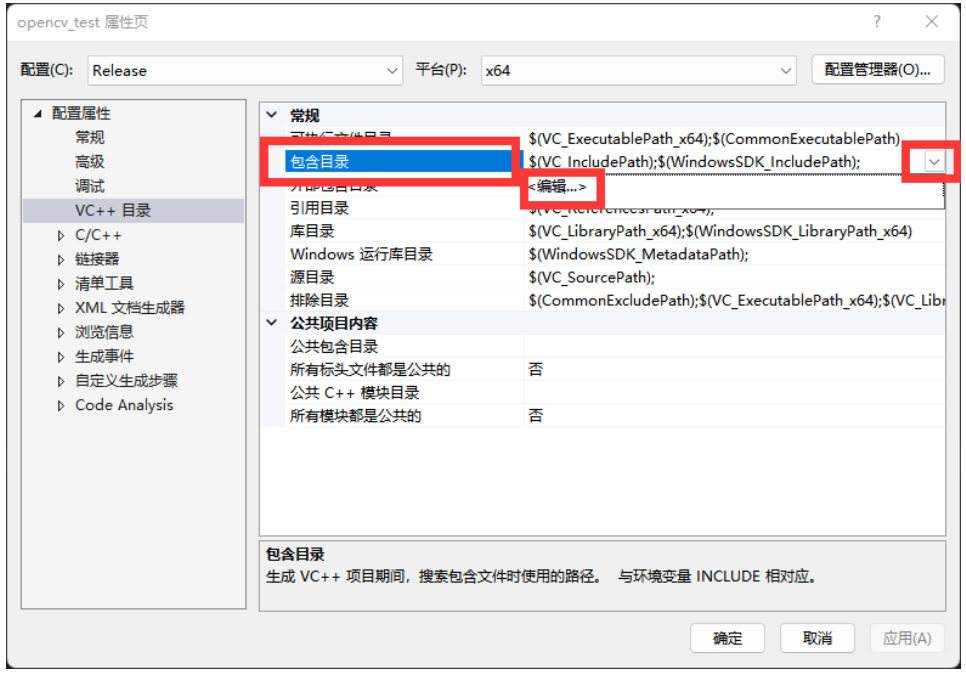
右键新建的项目

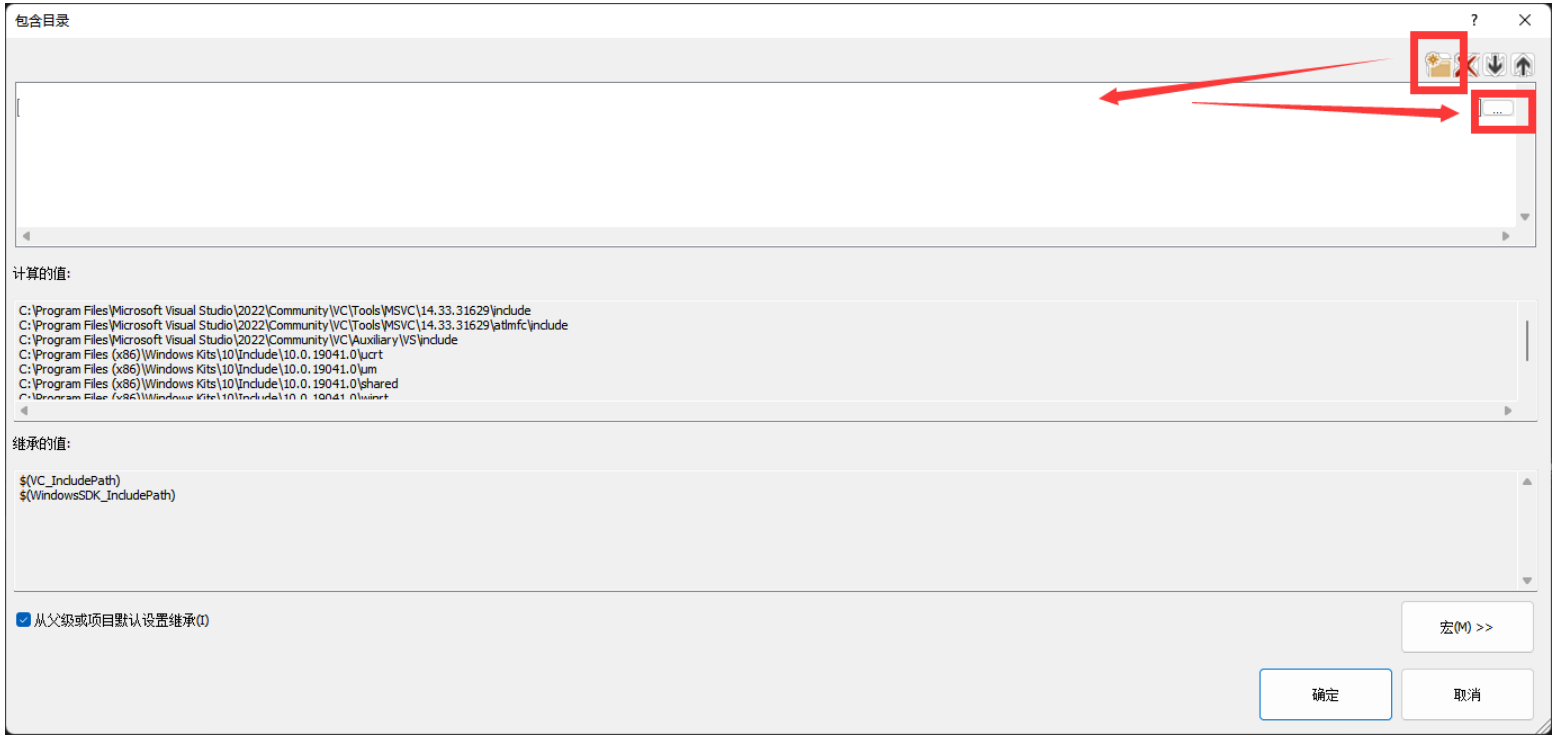
点击属性



点击VC++目录

Install OpenCV on VS 2022

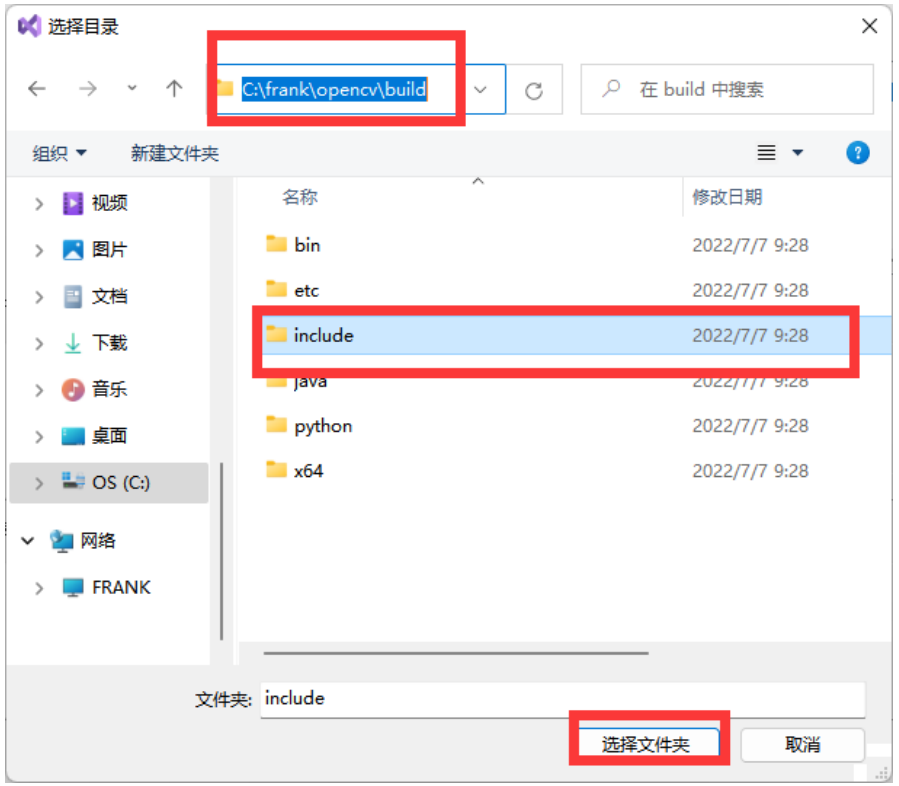


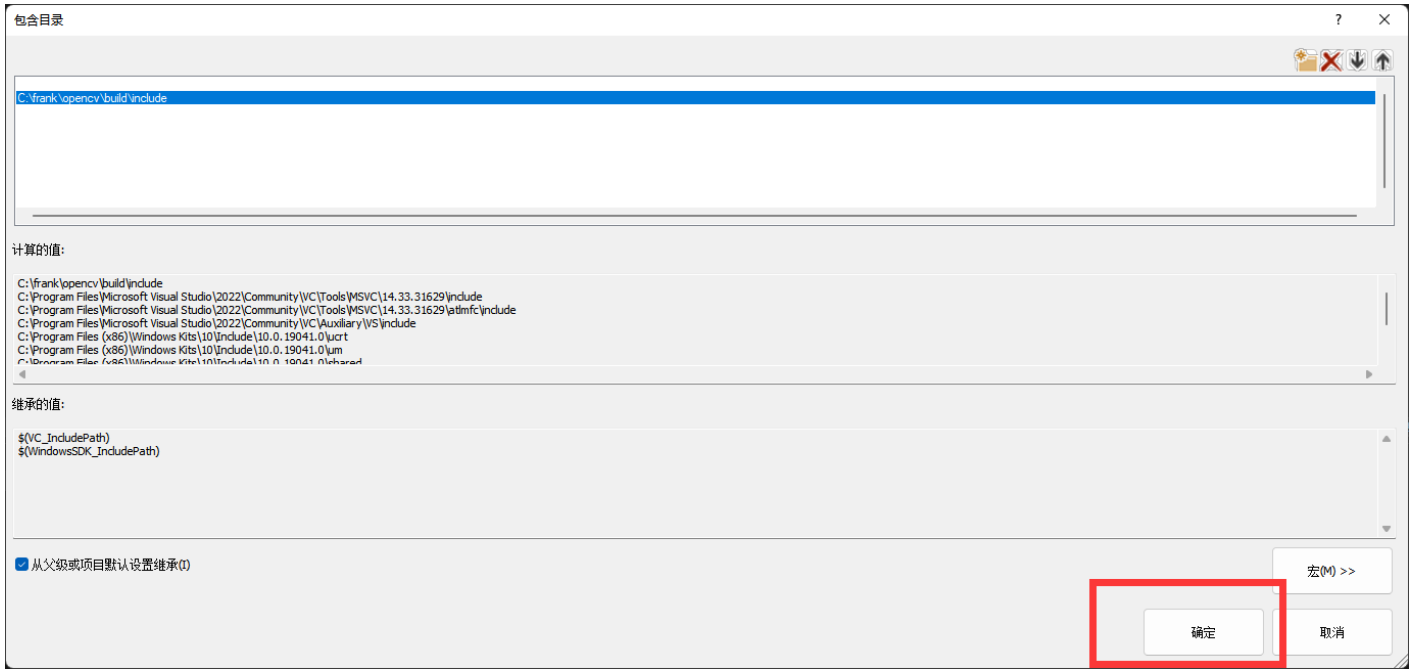


点击包含目录　点击右边的 ⬇ 点击编辑

依次点击

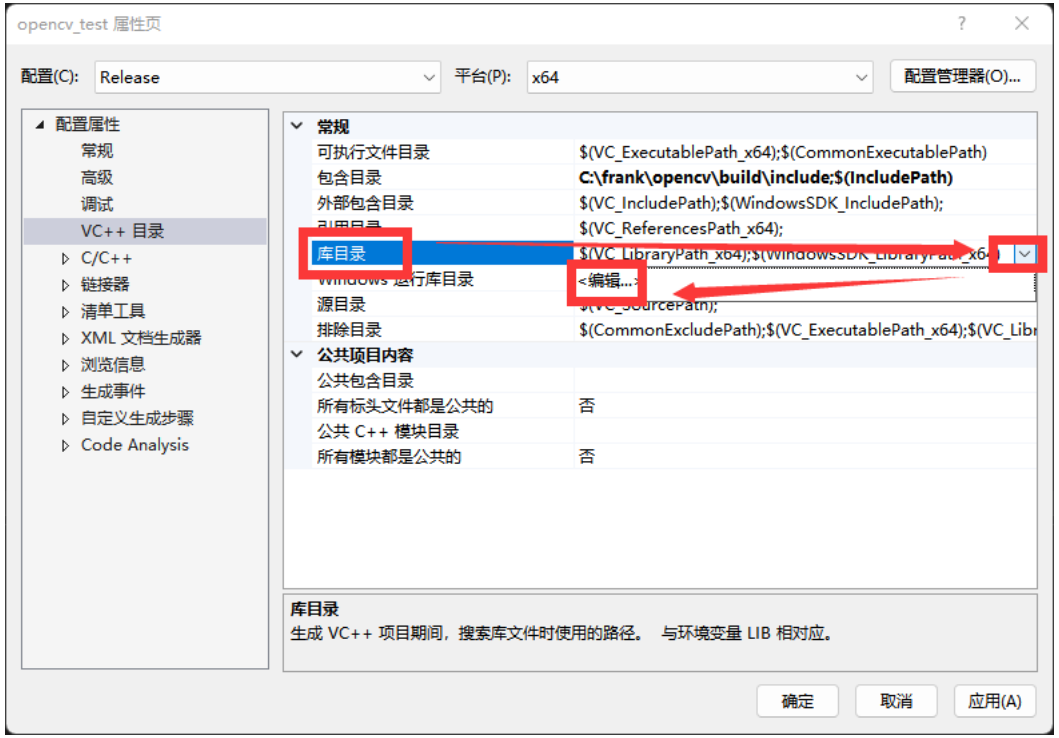
Install OpenCV on VS 2022





选择你安装opencv的路径 opencv/build/include 然后先应用再确定

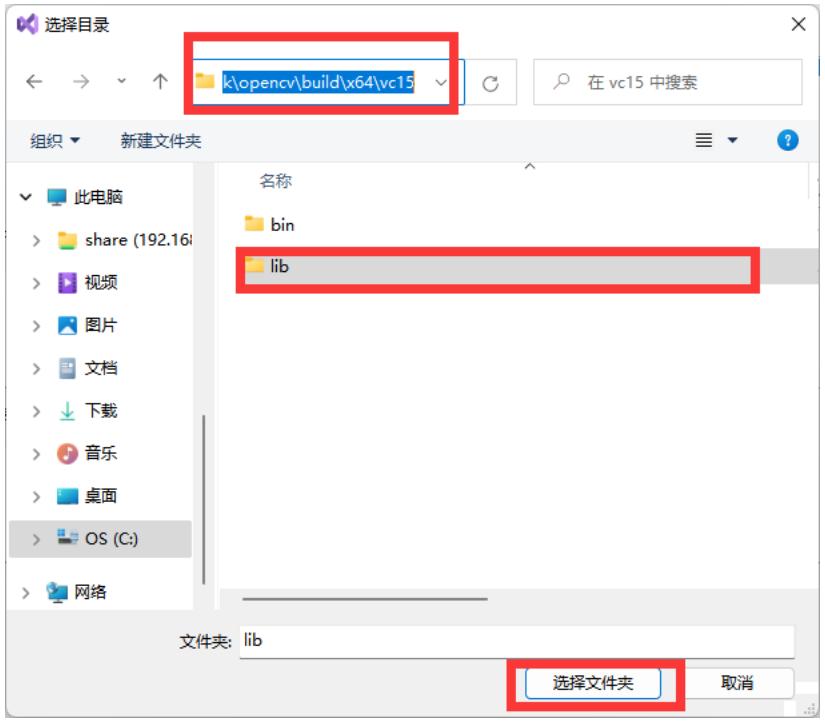
Install OpenCV on VS 2022

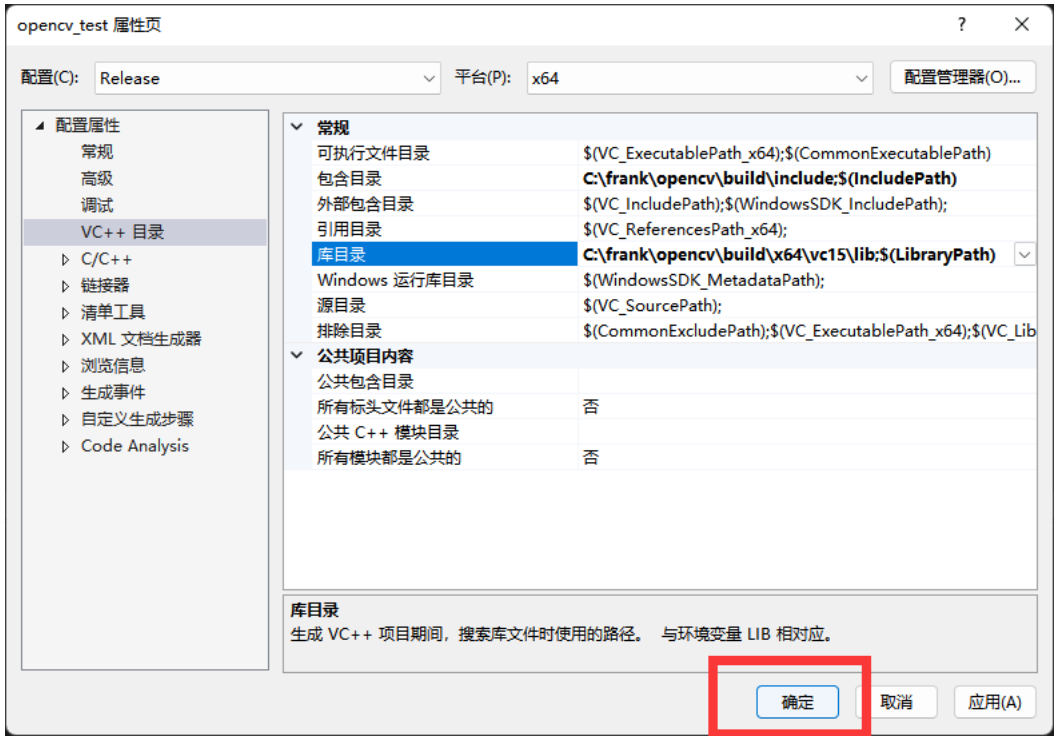


点击库目录　点击右边的 ⬇ 点击编辑

依次点击

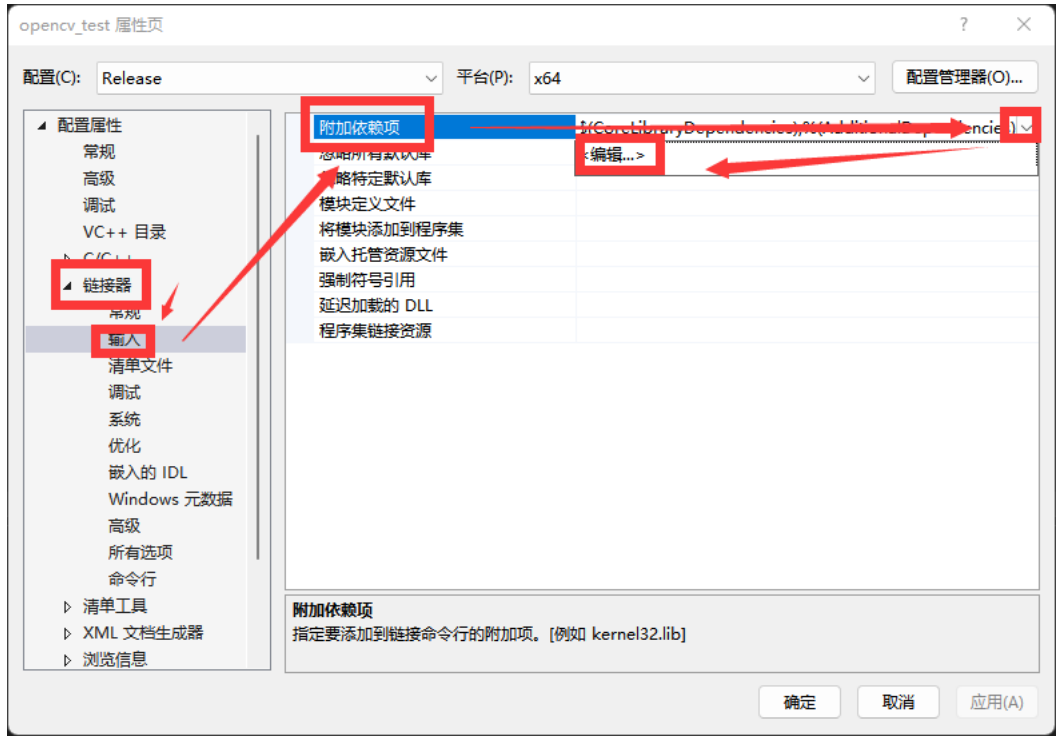
Install OpenCV on VS 2022

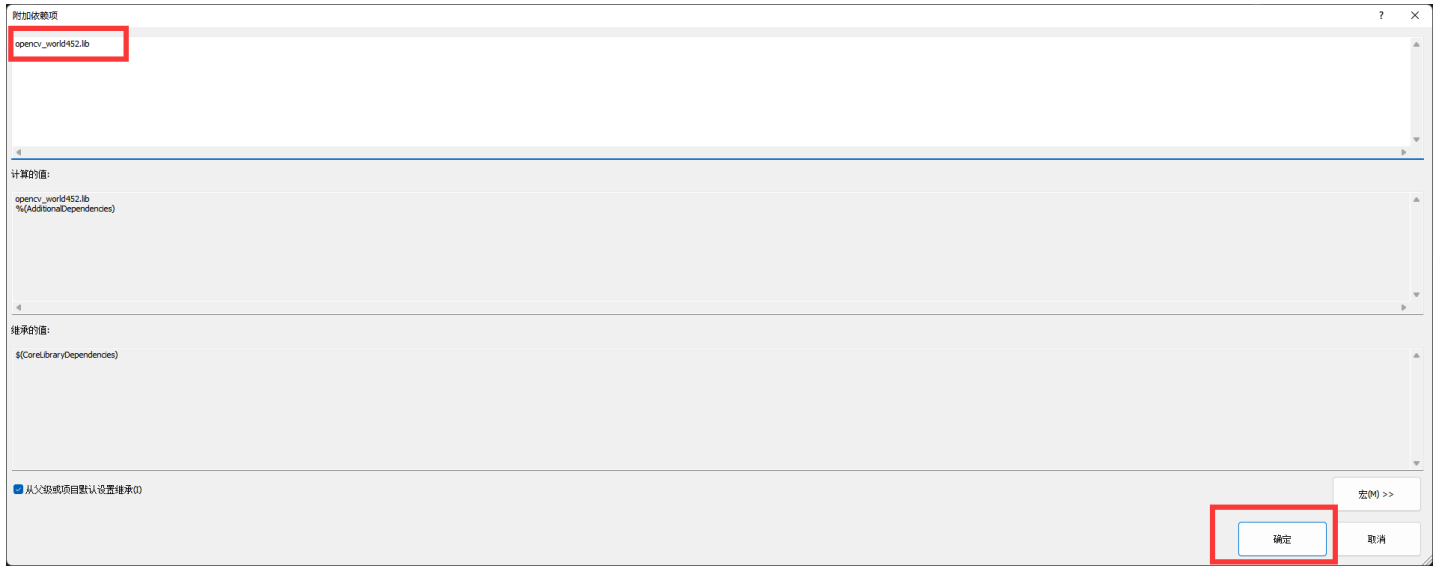




选择你安装opencv的路径 opencv/build/x64/vc15/lib 然后 先应用再确定

Install OpenCV on VS 2022

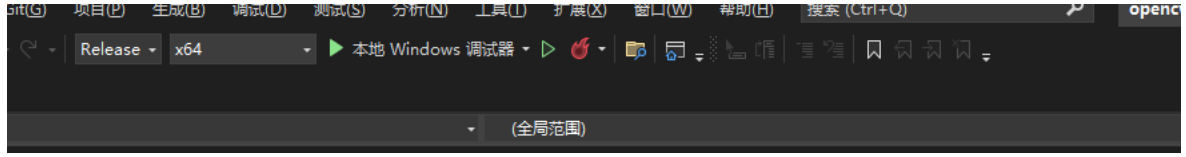




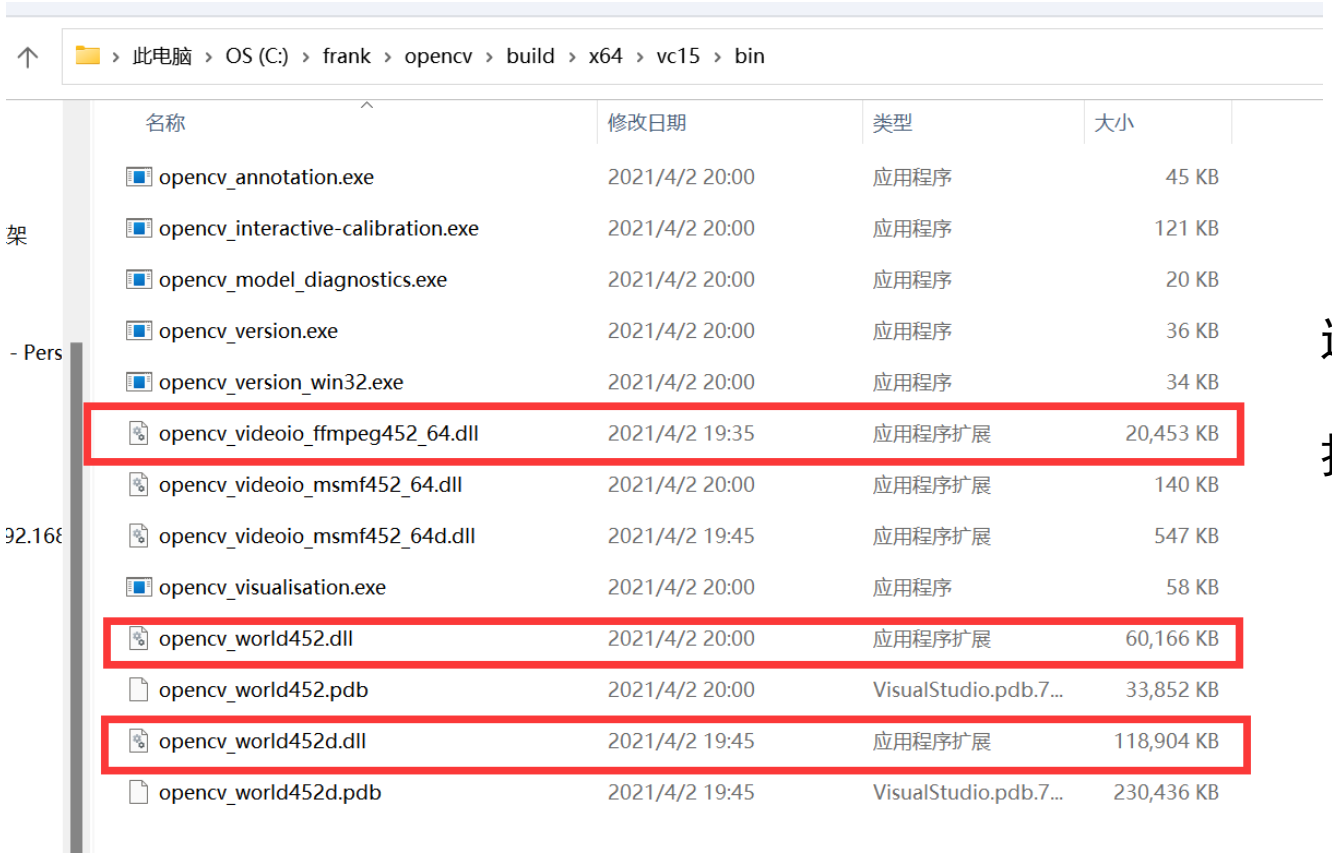
选择 链接器 -> 输入 -> 附加依赖项 -> ⬇ -> 编辑

手动输入 opencv_world452.lib 然后确定

Install OpenCV on VS 2022



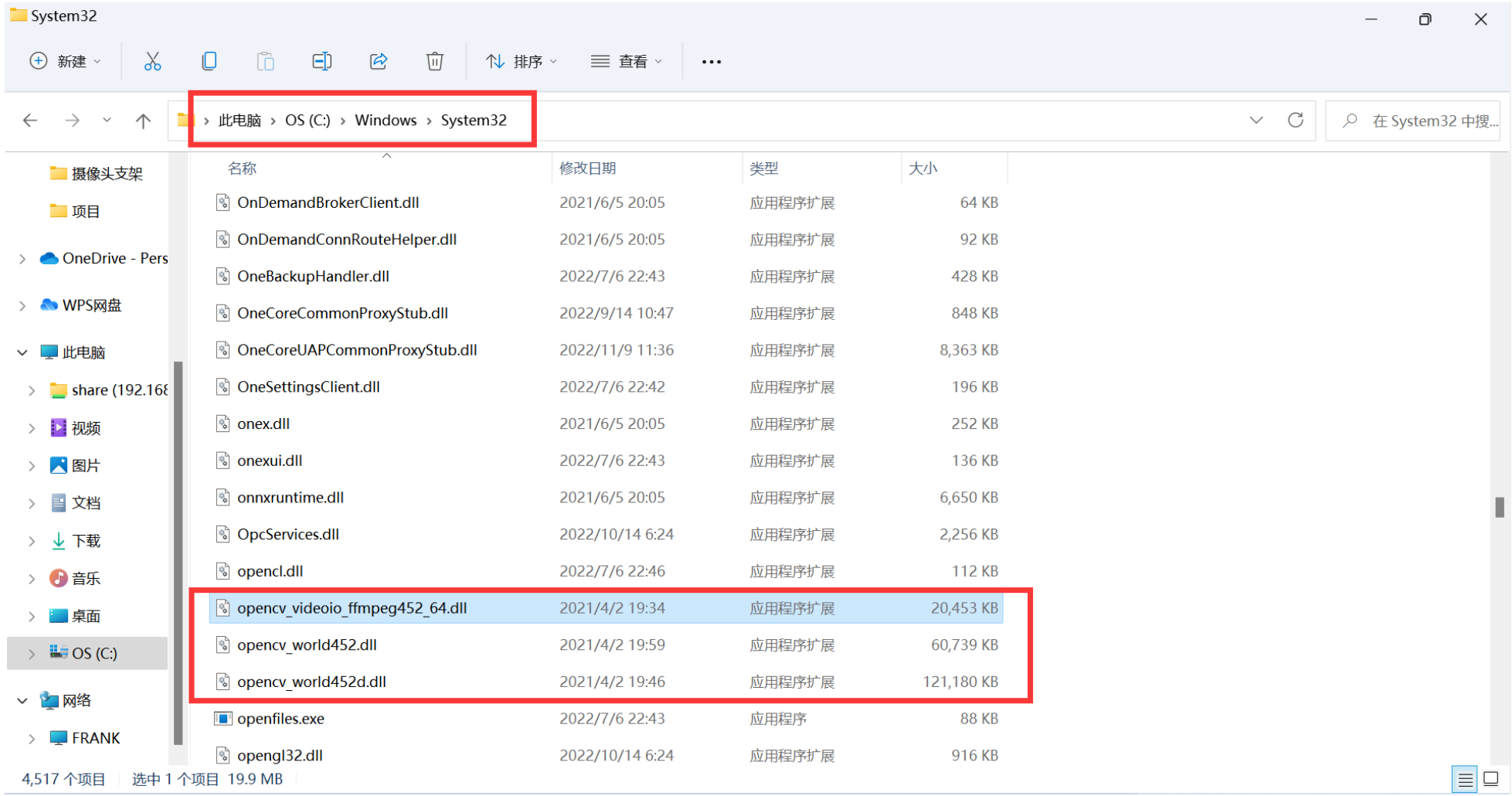
这里改成release　x64



进入到opencv/build/x64/vc15/bin

把这三个 .dll 文件复制到C：//Windows/system32下

Install OpenCV on VS 2022



就像这样

Install OpenCV on VS 2022

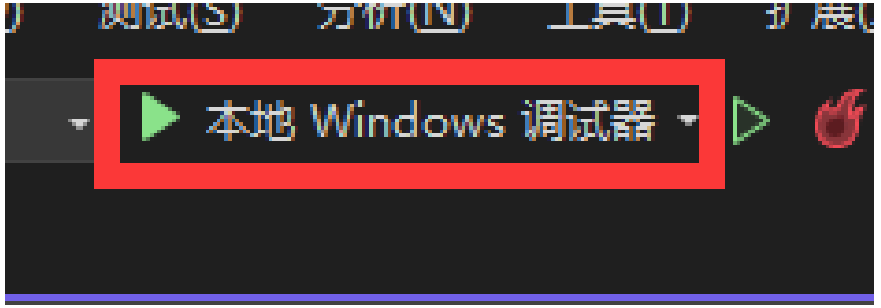
```
#include <iostream>
#include <opencv2/opencv.hpp>
using namespace std;
using namespace cv;
int main()
{
    // 读取图片（使用图片的绝对路径，参考自己的图所在目录）
    Mat srcImg = imread("C://Users//ydf19//Desktop//lenna.jpg");
    if (srcImg.empty()) {
        cout << "could not load image..." << endl;
        return -1;
    }

    imshow("Test opencv setup", srcImg);
    // 显示灰度图
    Mat Gray;
    cvtColor(srcImg, Gray, 6);
    imshow("Gray", Gray);
    // 等待任意按键按下，不添加此语句图片会一闪而过
    waitKey(0);

    cout << "Hello,world." << endl;
    return 0;
}
```

写几行代码测试一下

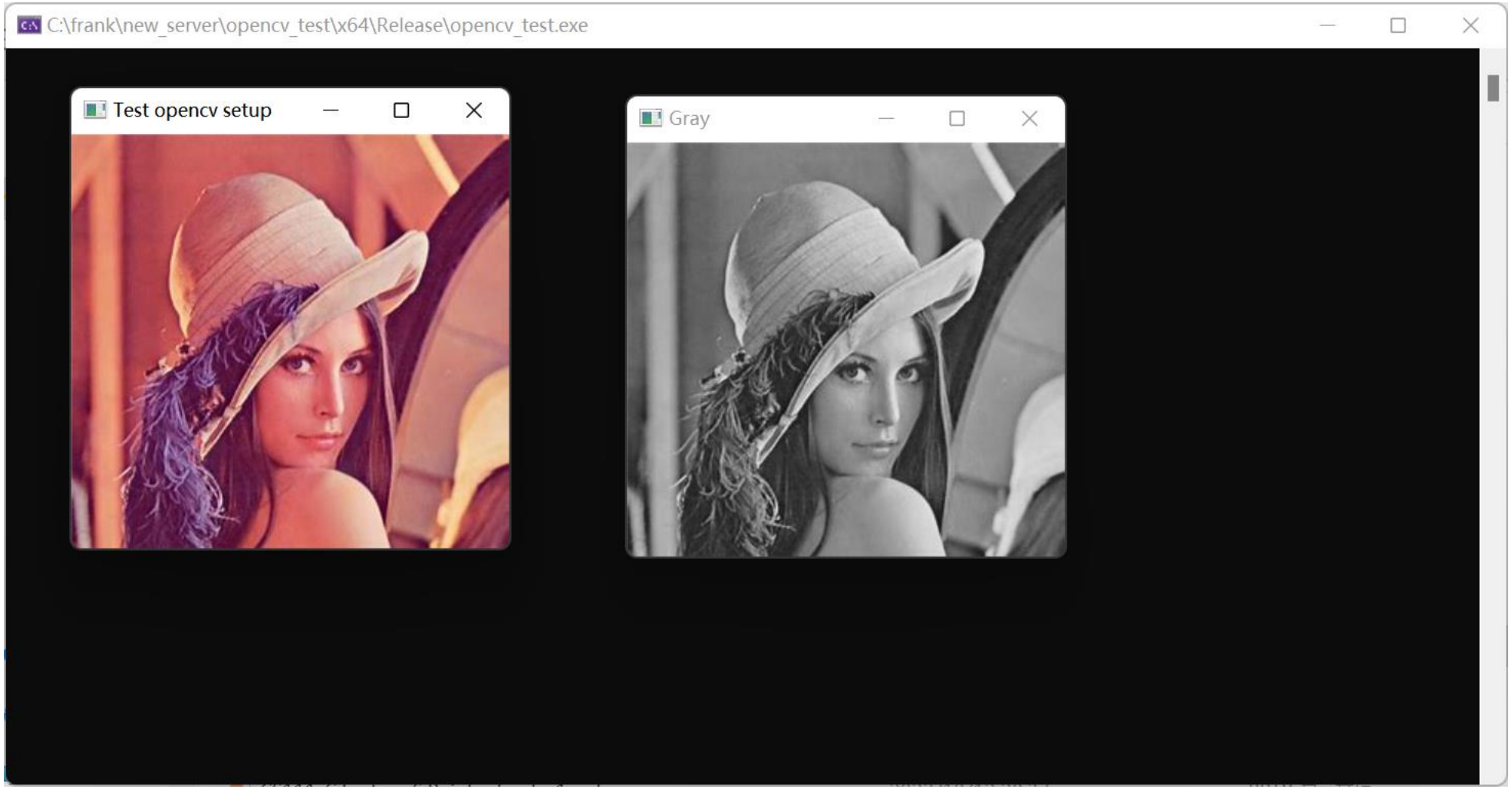
在桌面放一张图片

然后粘贴这些代码

运行



Install OpenCV on VS 2022



成功了!

其他系统和编译器的安装可以参考以下链接

Vscode & macOS https://blog.csdn.net/weixin_43562948/article/details/103956901

Vscode & windows https://blog.csdn.net/Avrilzyx/article/details/107036375

DEV c++ & windows https://blog.csdn.net/wadefelix/article/details/1334515