

聚类算法

—— K-means聚类

将对象的集合分成由类似的对象组成的多个类别的过程。



◆ NBA球员能力分析

◆ 球员能力指标？

得分

篮板

助攻

防守

.....

K-means算法 NBA球星能力分析

詹姆斯



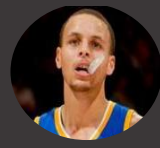
韦德



威少



库里



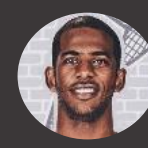
杜兰特



哈登



保罗



欧文



助攻

9.0

3.6

7.5

4.0

4.3

6.5

5.8

7.3

篮板

8.7

4.2

12.0

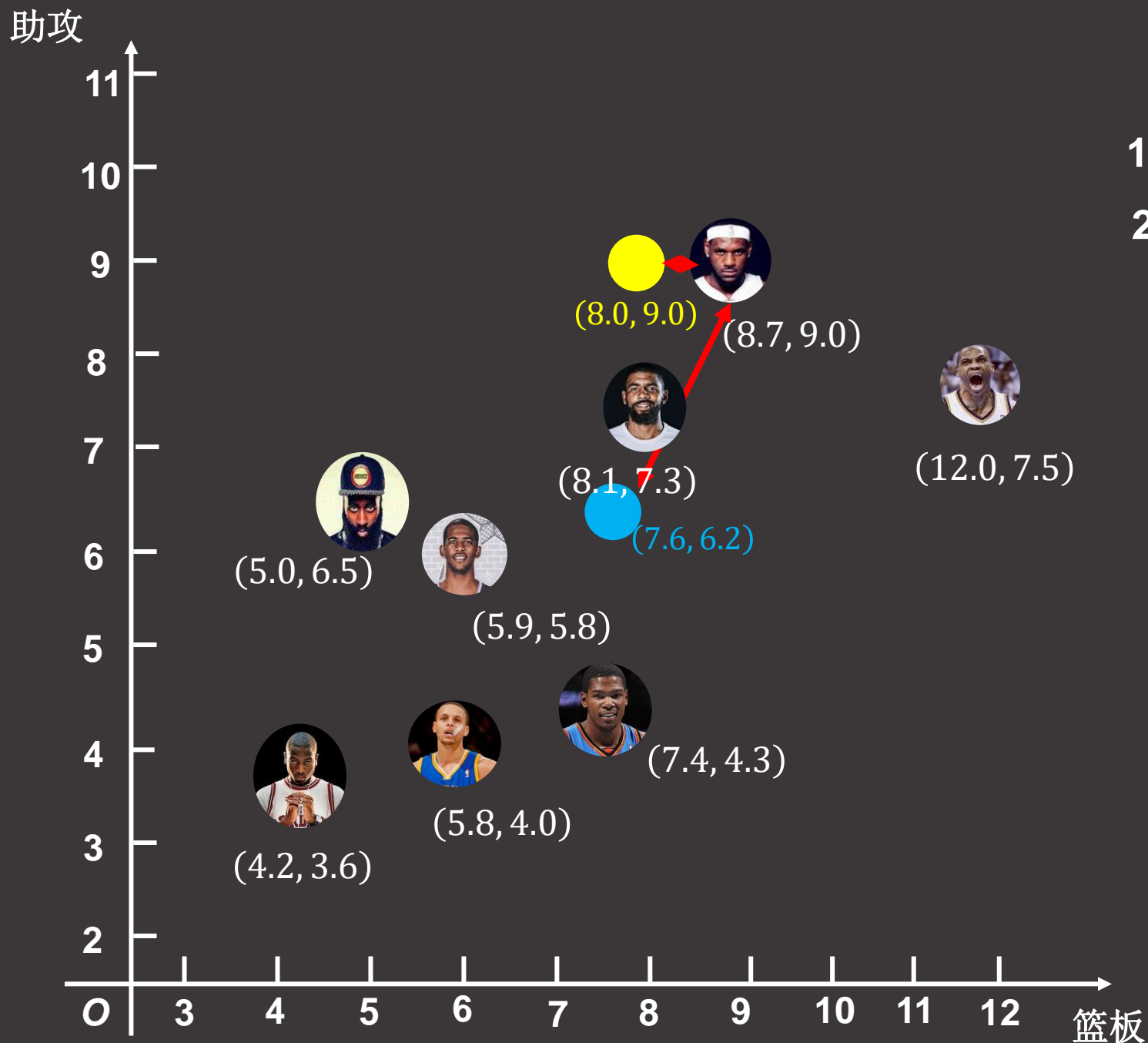
5.8

7.4

5.0

5.9

8.1

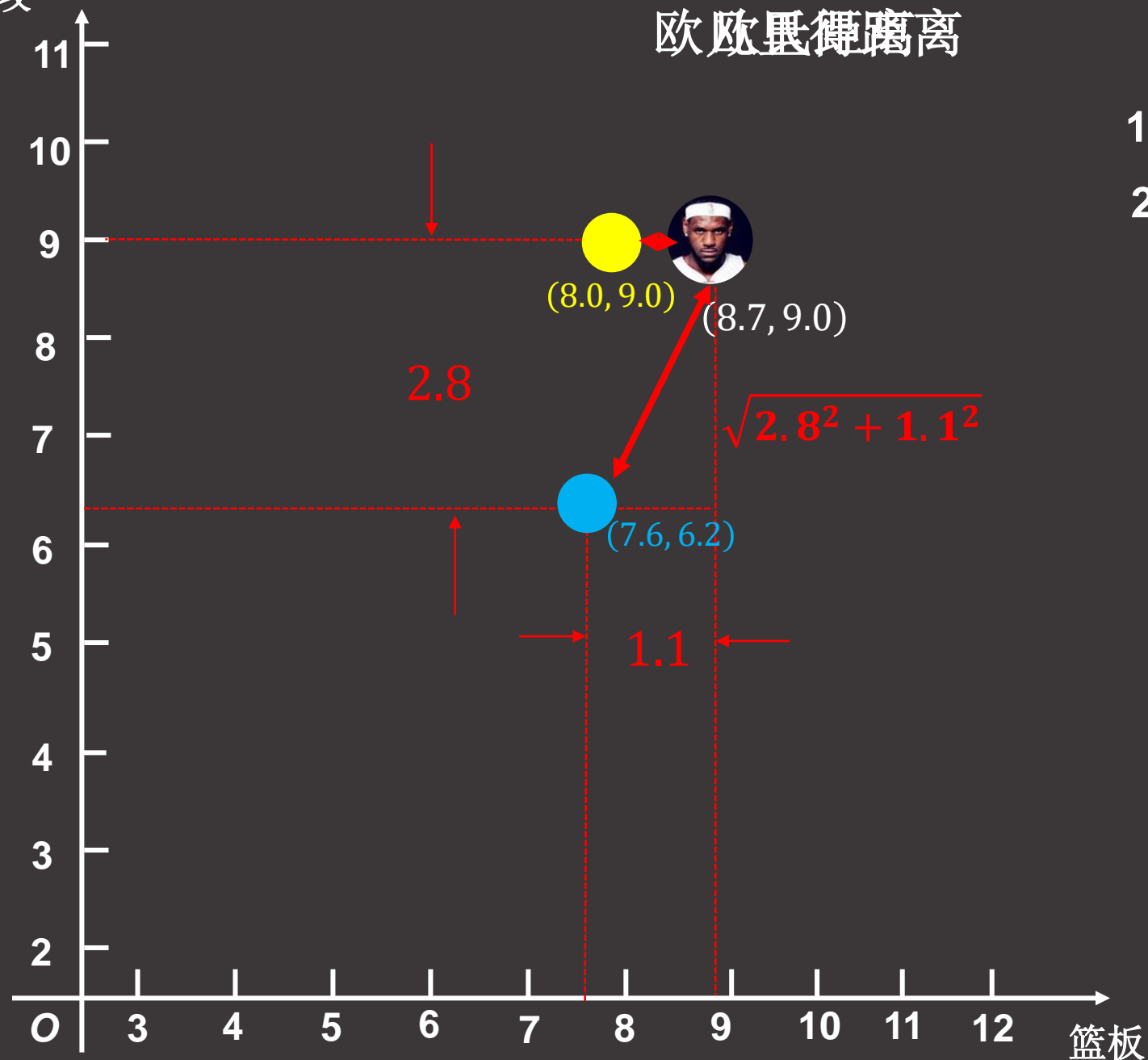


- 1、随机生成两个点
- 2、类别划分

```
if |詹姆斯,黄点| < |詹姆斯,蓝点| :
    詹姆斯 -> 黄队
else
    詹姆斯->蓝队
```

助攻

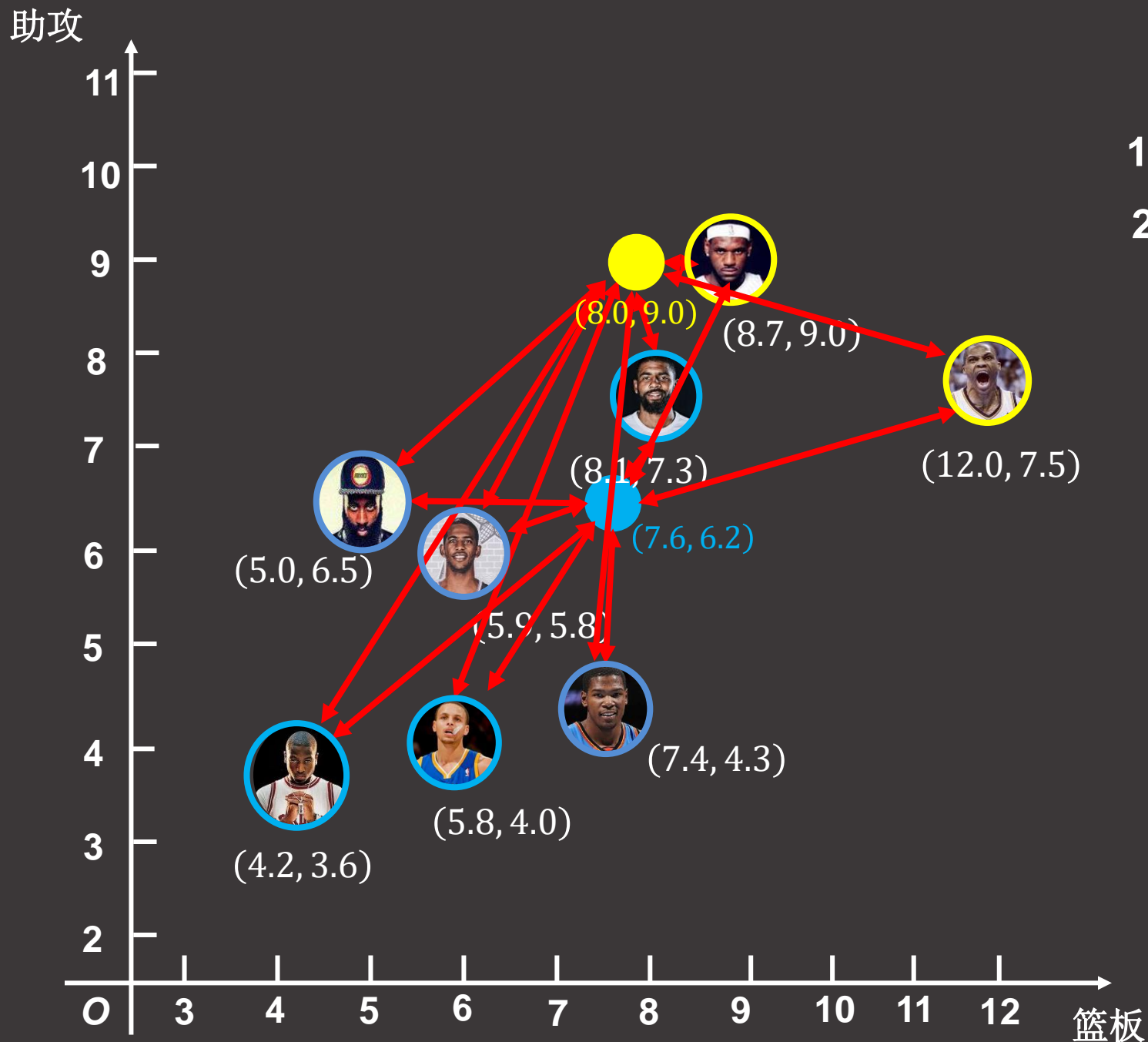
欧欧畏御离离



1、随机生成两个点

2、类别划分

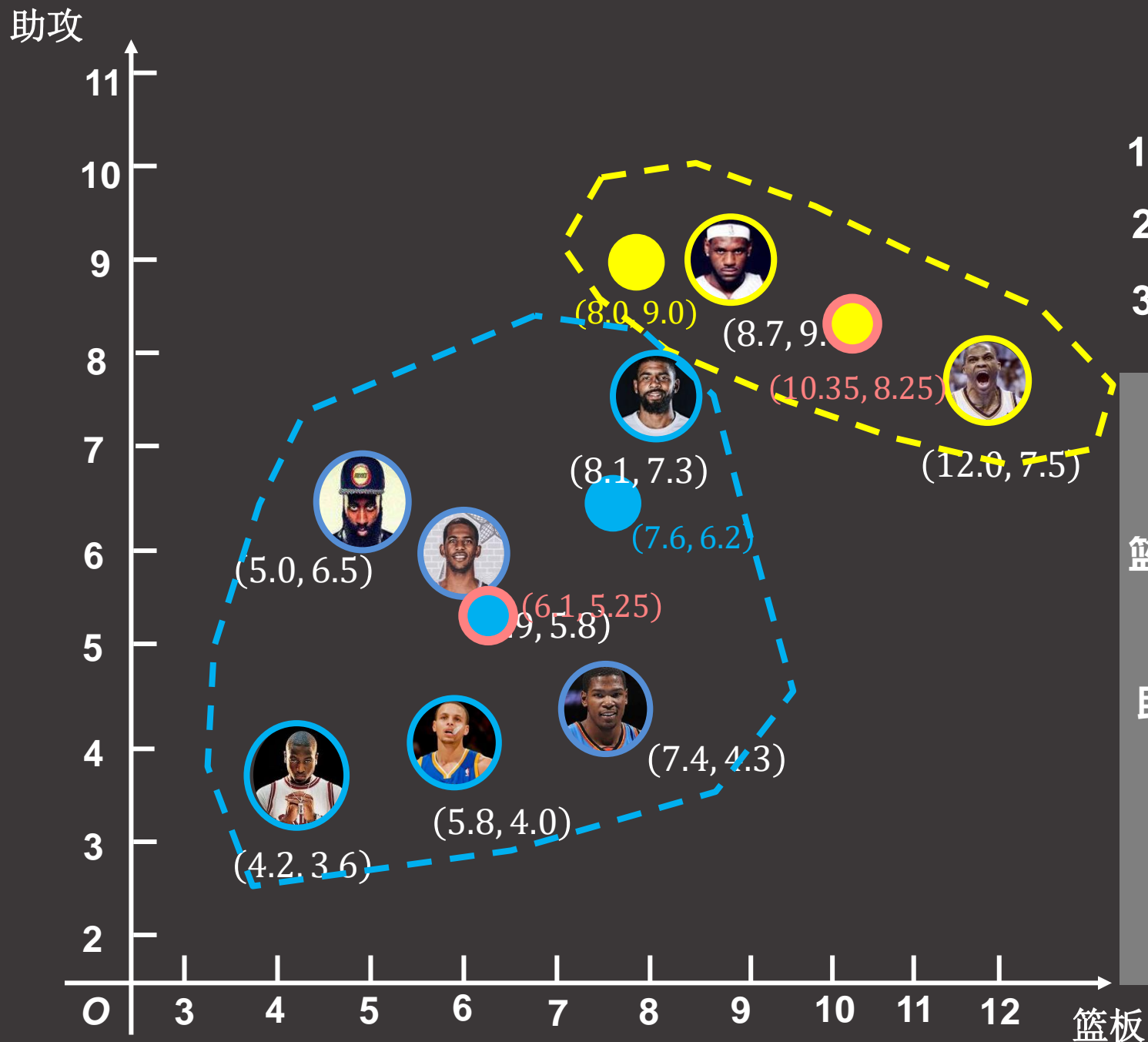
```
if |詹姆斯,黄点| < |詹姆斯,蓝点| :
    詹姆斯 -> 黄队
else
    詹姆斯-> 蓝队
```



1、随机生成两个点

2、类别划分

```
if |詹姆斯,黄点| < |詹姆斯,蓝点| :
    詹姆斯 -> 黄队
else
    詹姆斯->蓝队
```



1、随机生成两个点

2、类别划分

3、分析计算
 $|詹姆斯, 黄点| < |詹姆斯, 蓝点|$:
 计算每类球员的真实平均能力
 詹姆斯 -> 黄队

else

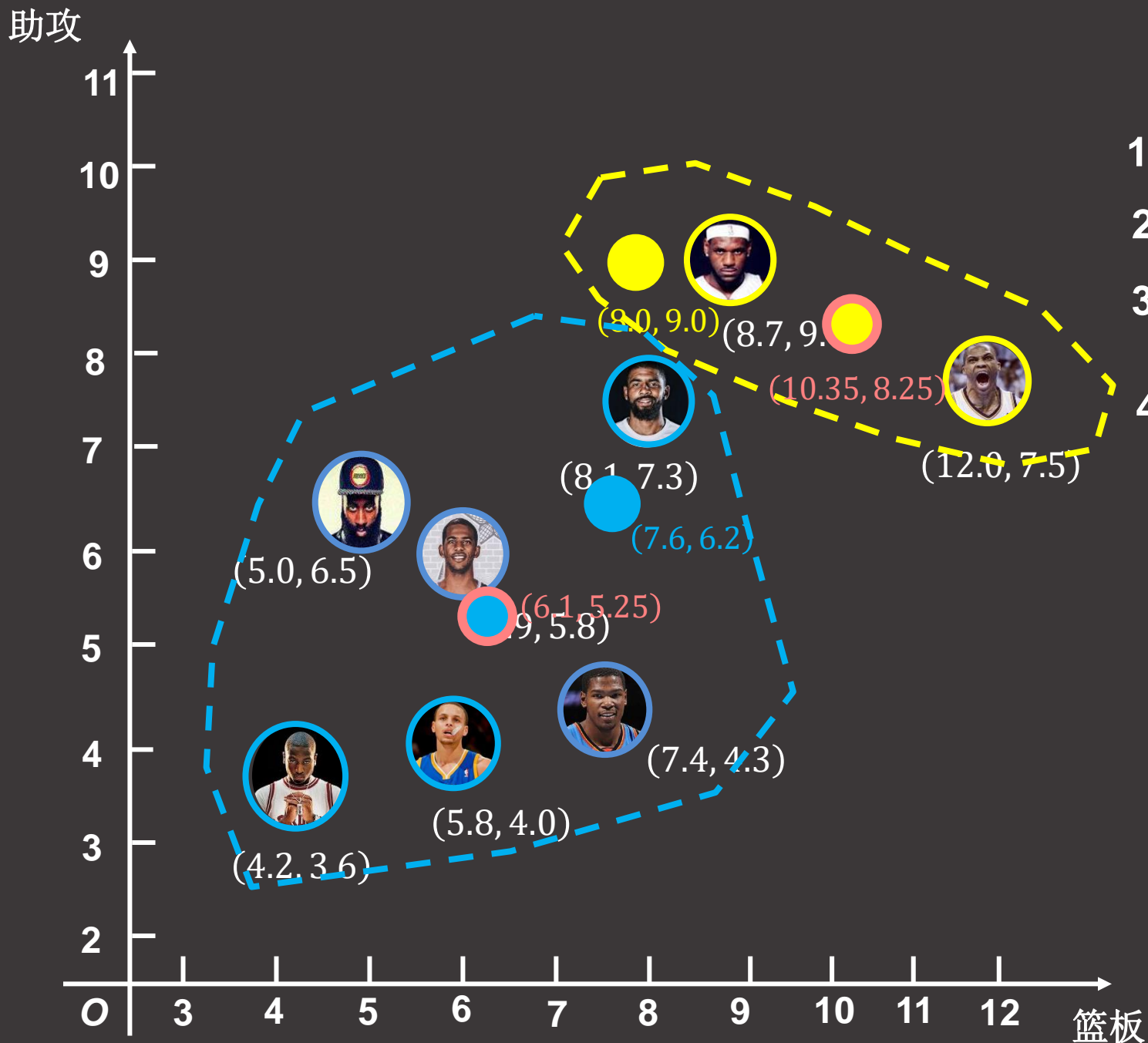
詹姆斯 -> 蓝队

$$\text{篮板} : \frac{\quad}{6} = 6.1$$

$$\text{助攻} : \frac{3.6+4.0+4.3+5.8+6.5+7.3}{6} = 5.25$$

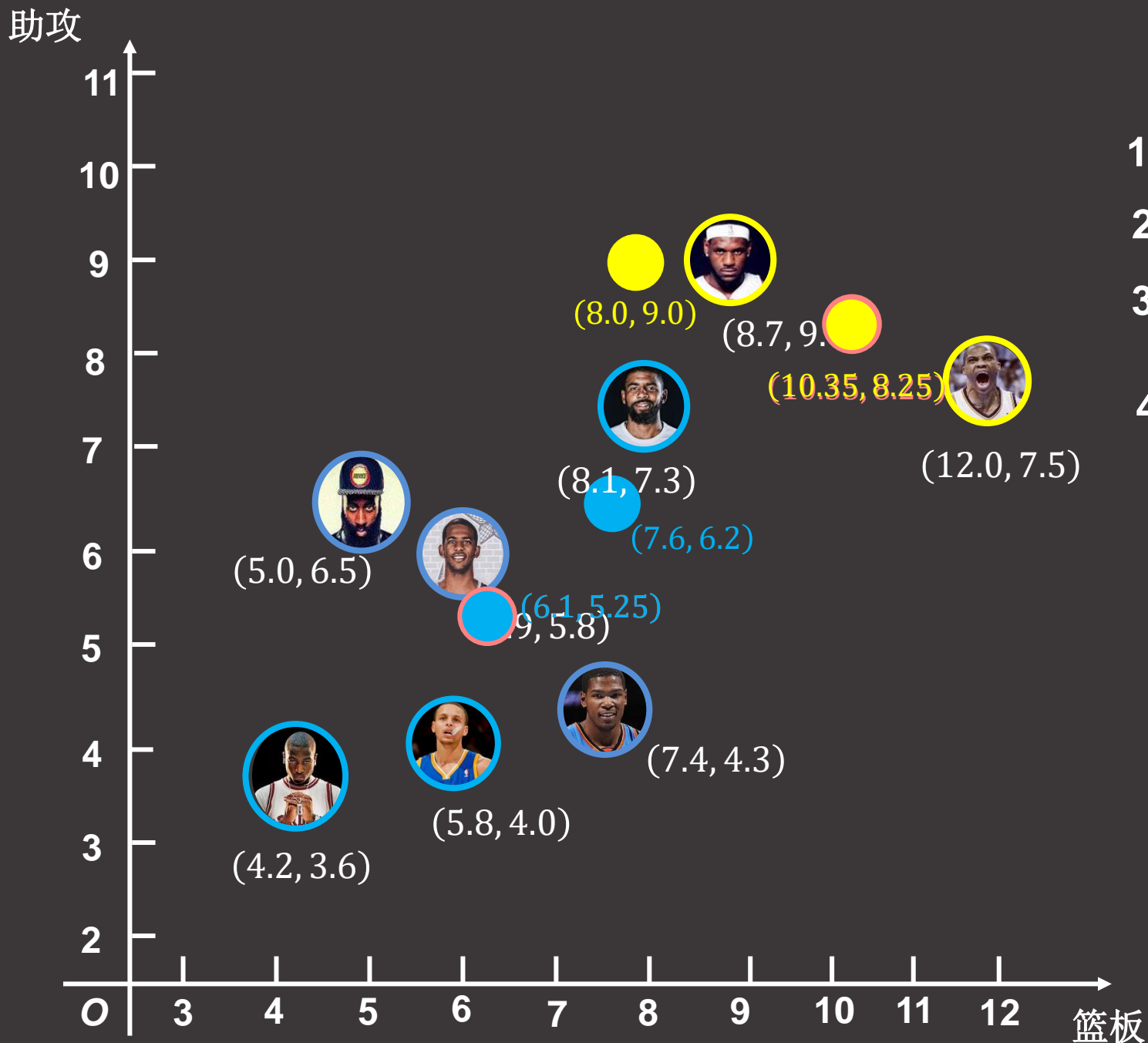
黄队真实能力

(10.35, 8.25)



- 1、随机生成两个点
- 2、类别划分
- 3、分析调整
— 计算每类球员的真实平均能力
- 4、判断聚类是否结束

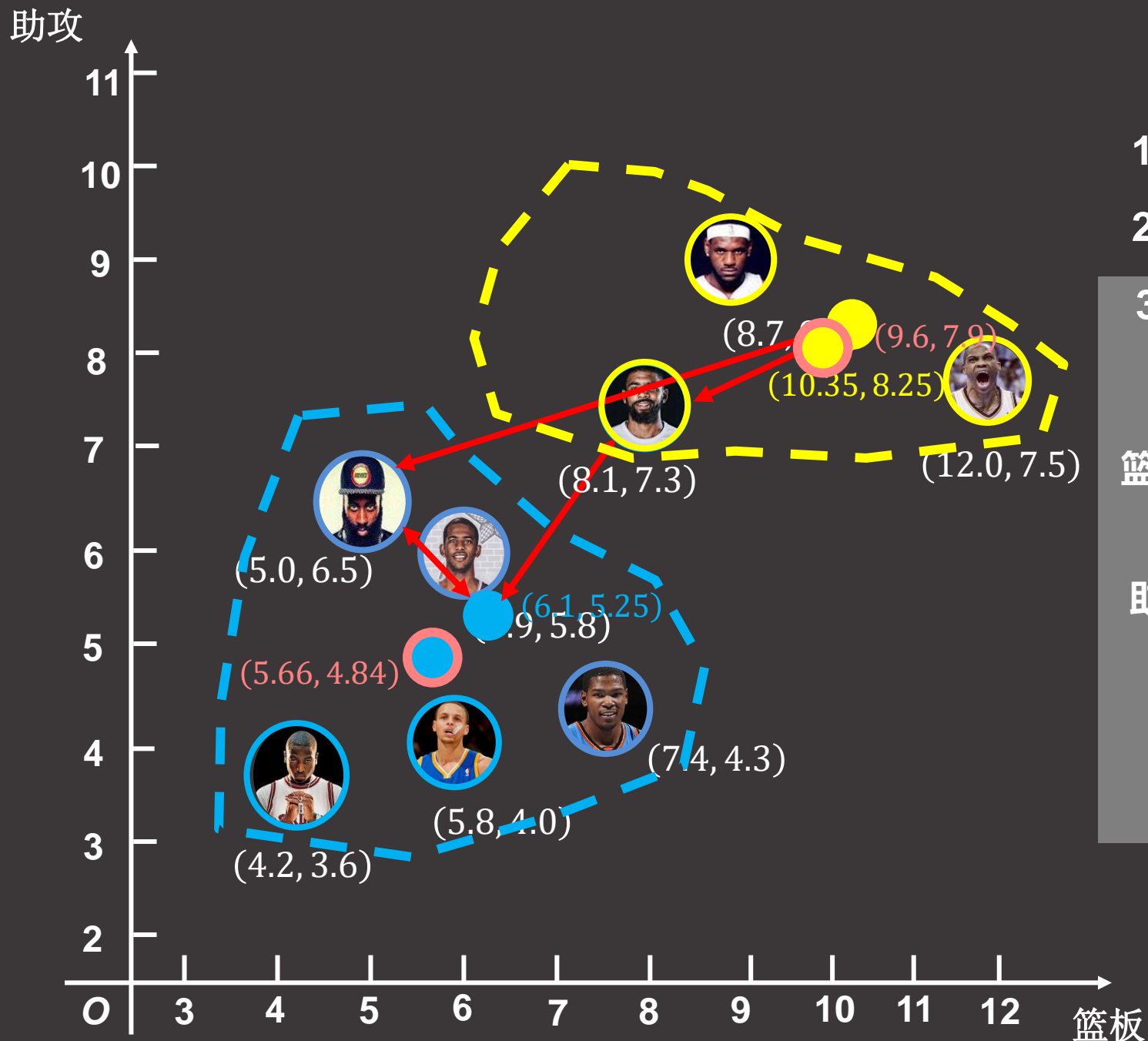
if 真实平均能力 != 假定平均能力 :
 假定平均能力 = 真实平均能力
 else:
 return 聚类中心 , 聚类结果



- 1、随机生成两个点
- 2、类别划分
- 3、分析调整
— 计算每类球员的真实平均能力
- 4、判断聚类是否结束

```

if 真实平均能力 != 假定平均能力 :
    假定平均能力 = 真实平均能力
else:
    return 聚类中心 , 聚类结果
  
```



1、类别划分

2、计算每类球员的真实平均能力

3、判断聚类是否结束

蓝队真实能力

if 真实平均能力 != 假定平均能力 :

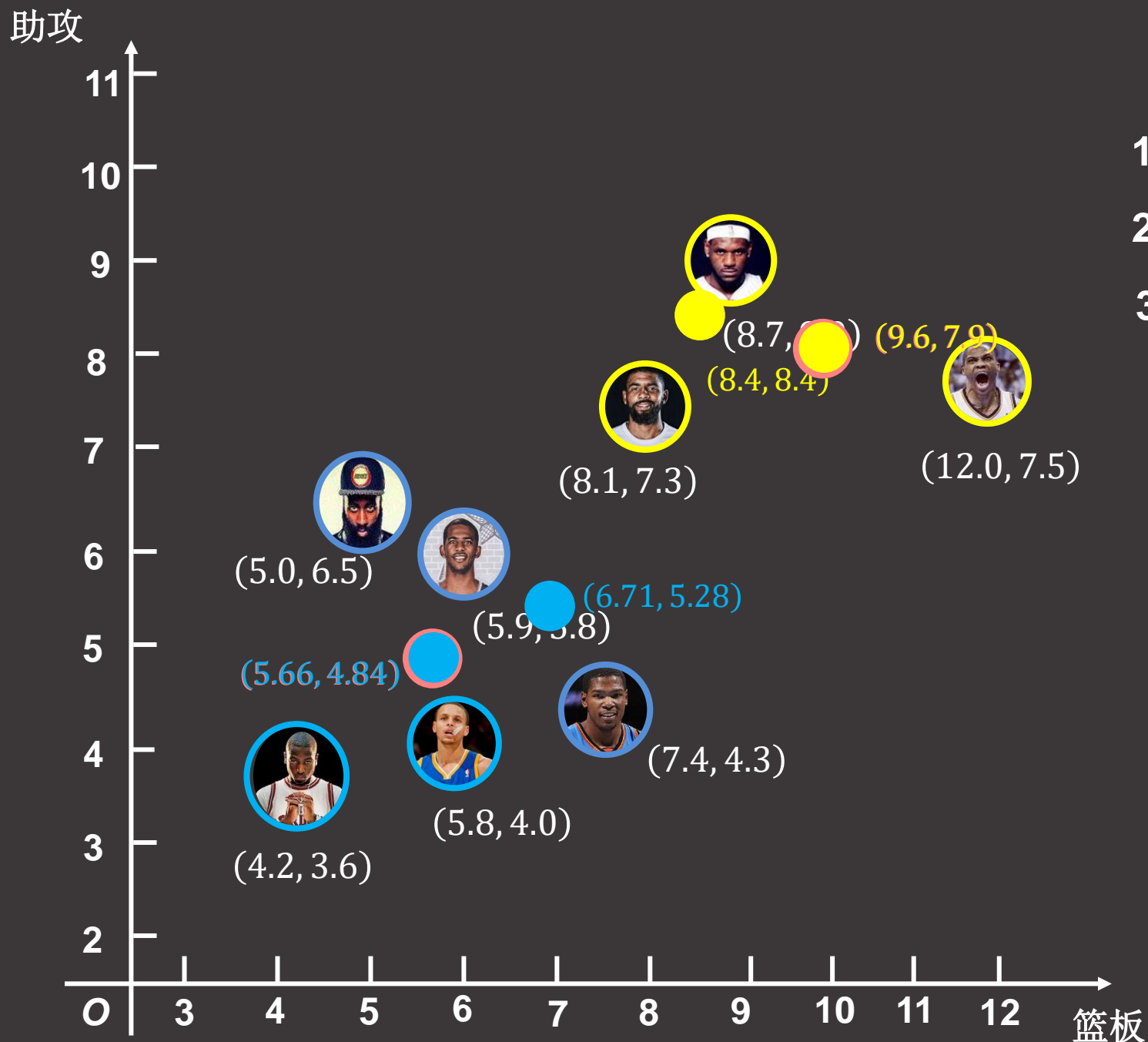
篮板 假定平均能力 = 真实平均能力 5.66

else:

助攻: return 聚类中心, 聚类结果 4.84

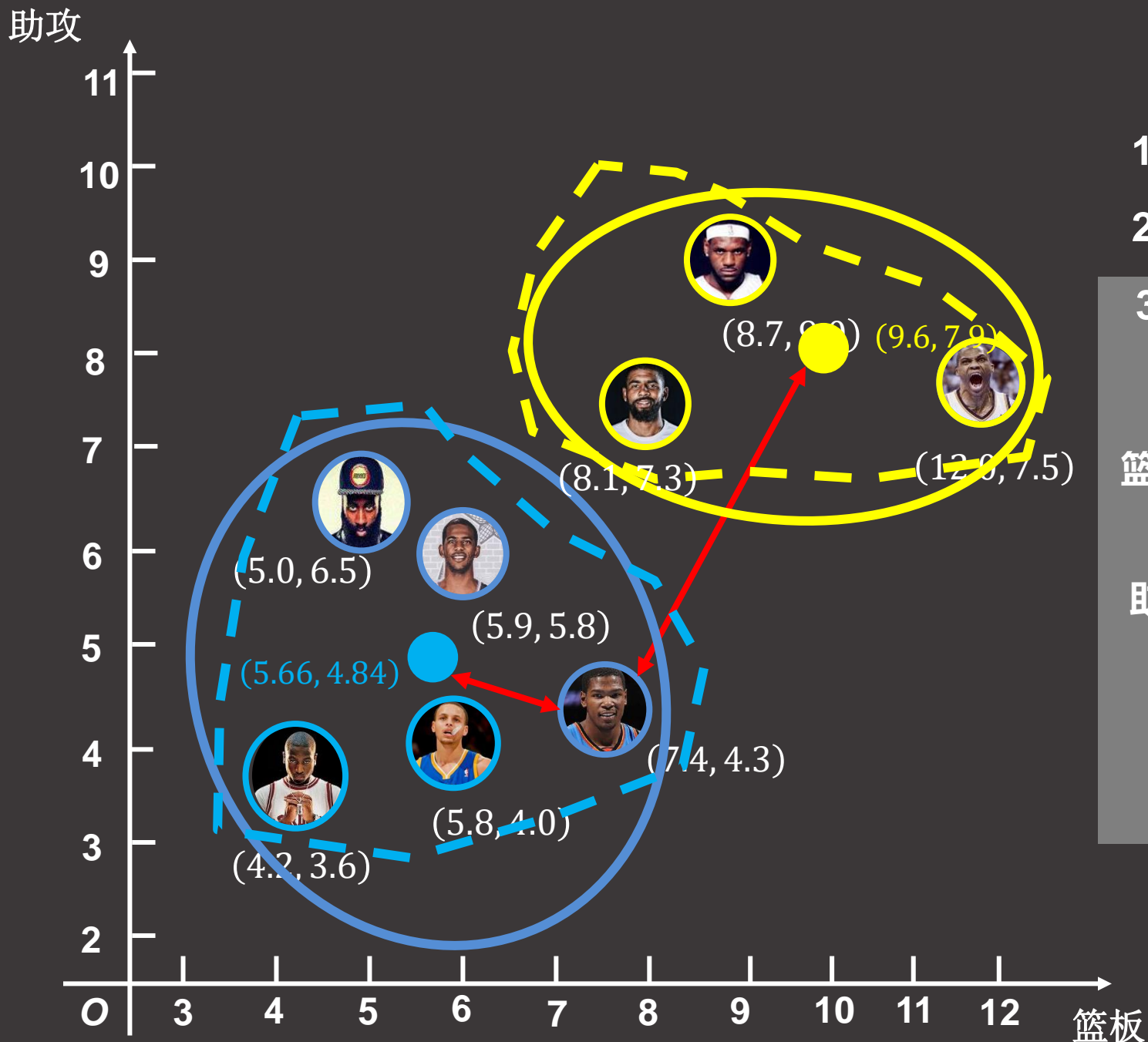
黄队真实能力

(9.6, 7.9)



- 1、类别划分
- 2、计算每类球员的真实平均能力
- 3、判断聚类是否结束

if 真实平均能力 \neq 假定平均能力 :
 假定平均能力 = 真实平均能力
 else:
 return 聚类中心 , 聚类结果



1、类别划分

2、计算每类球员的真实平均能力

3、判断聚类是否结束

蓝队真实能力

if 真实平均能力 != 假定平均能力 :

篮板 假定平均能力 = 真实平均能力 5.66

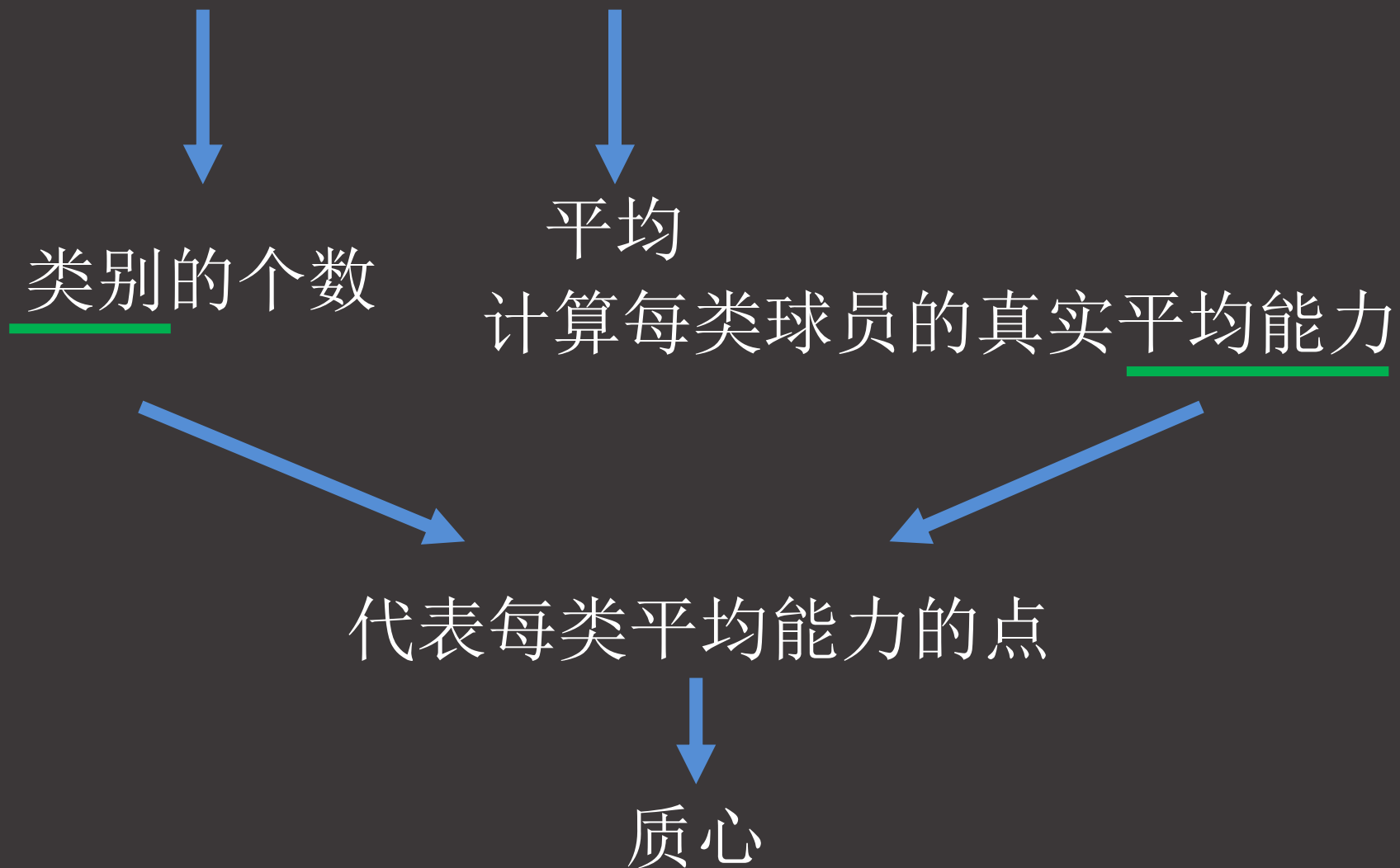
else:

助攻: return 聚类中心 , 聚类结果 4.84

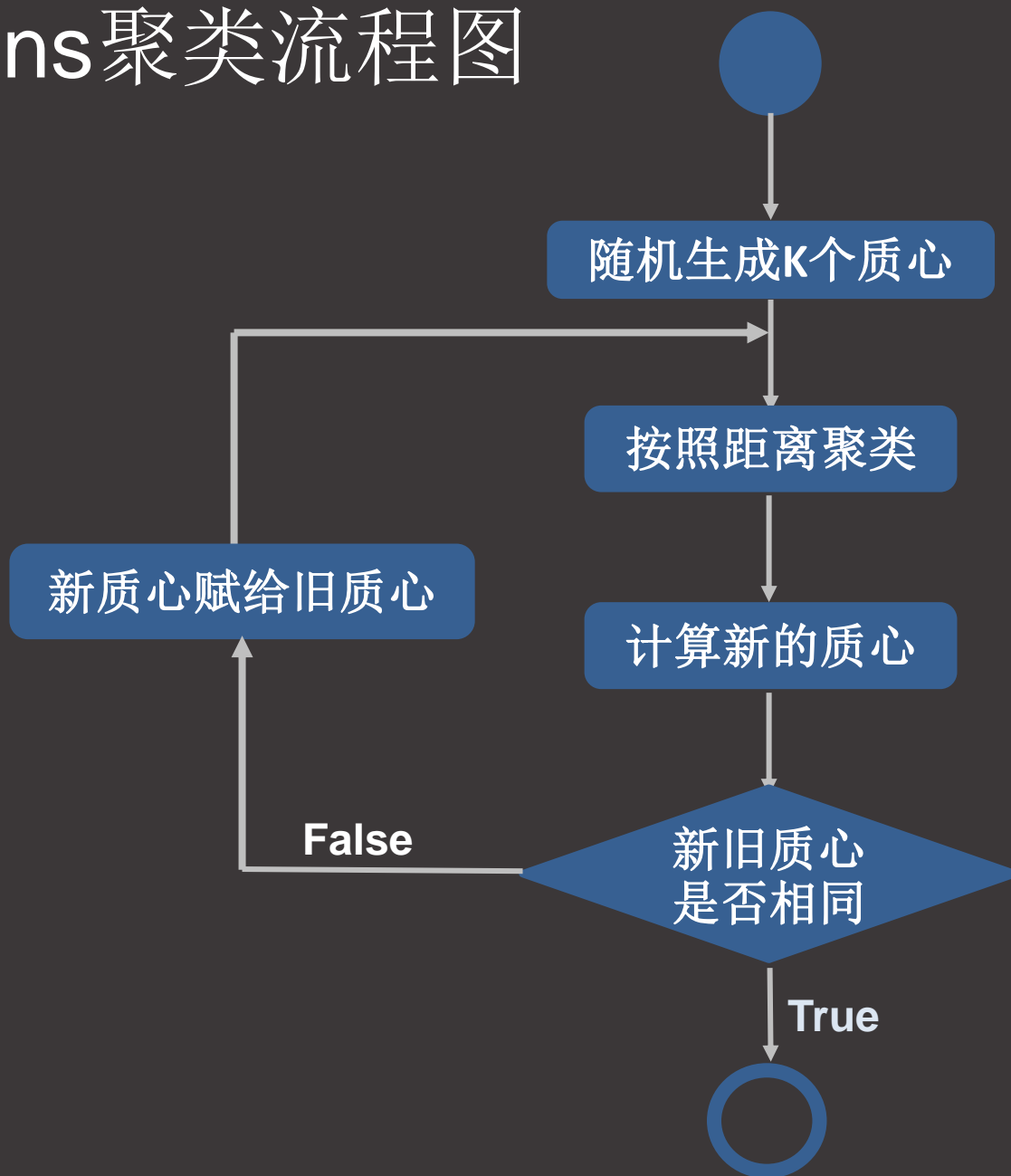
黄队真实能力

(9.6, 7.9)

K - means



K-means聚类流程图



kmeans

+randCent(dataSet, k) 随机生成质心

+distEclud(vecA, vecB) 计算欧式距离

+cluster(dataSet,k,cenBefor) 聚类

+calCent(dataSet,k,clusterAssment) 计算新质心

+KMeans(dataSet, k,reateCent = randCent)



代码实现

课堂
练习



kmean.py x

4

随机生成质心

5

+ def randCent(dataSet, k):...

14

计算欧式距离

15

+ def distEclud(vecA, vecB):...

16

按照欧式距离进行聚类

17

+ def cluster(dataSet, k, cenBefor):...

18

重新计算质心

19

+ def calCent(dataSet, k, clusterAssment):...

20

K均值算法实现

21

- def KMeans(dataSet, k, createCent=randCent):

22

- # 随机生成两个点，假定为两类球员的平均能力

23

- # 类别划分

24

- # 计算每类球员的真实平均能力，即质心

25

- # 判断聚类是否结束

26

- pass

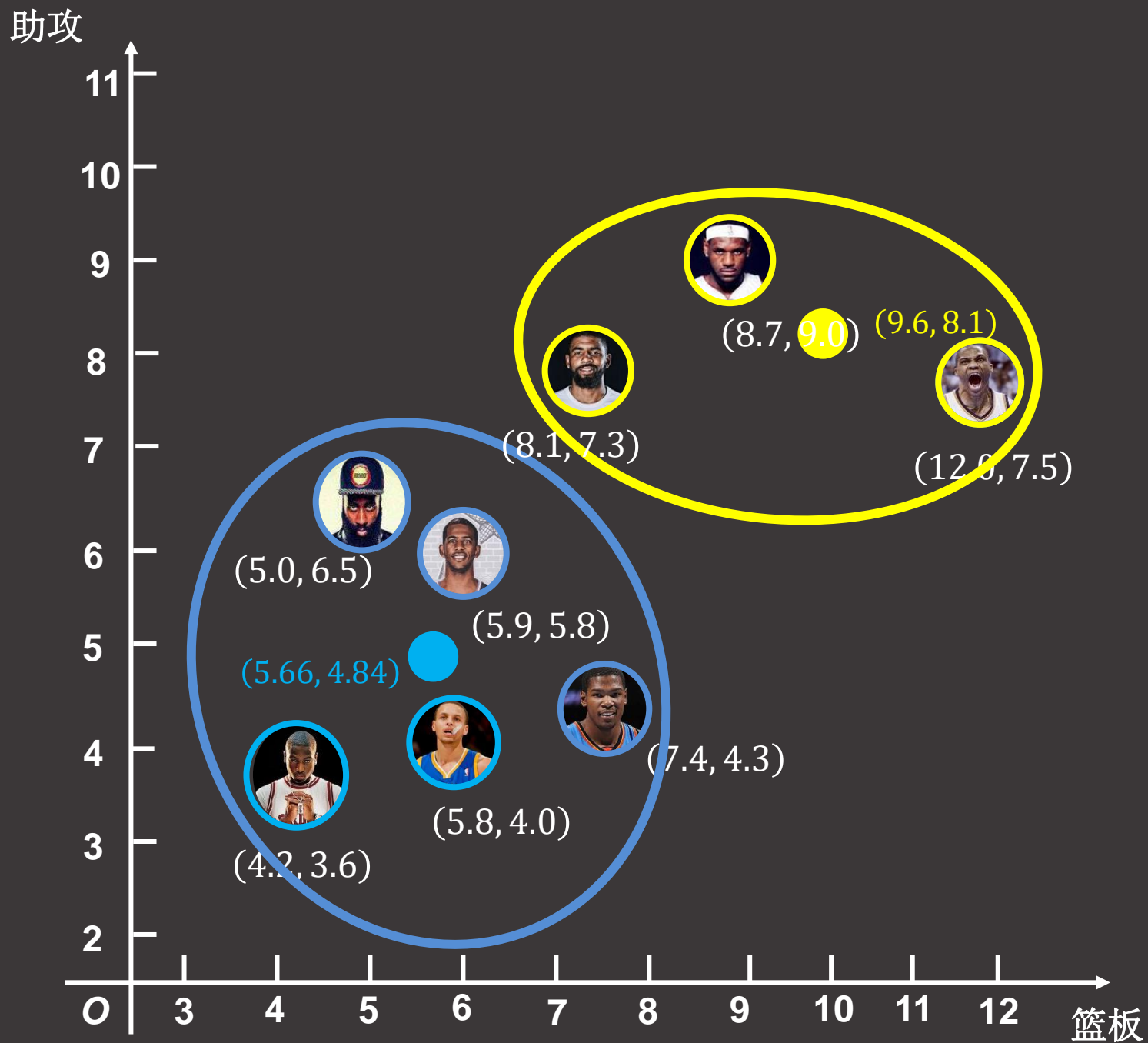
27

- if __name__ == '__main__':

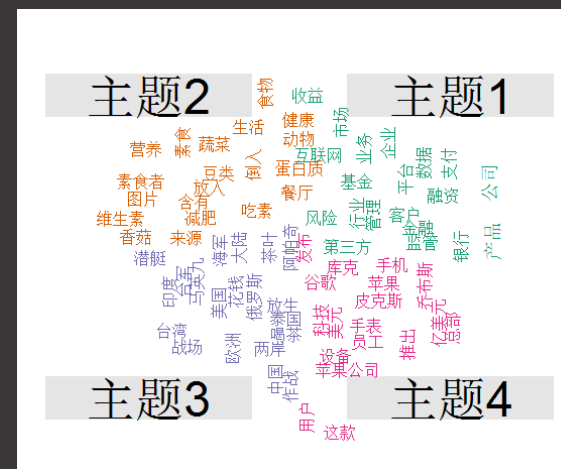
28

-

KMeans()



-
- A diagram illustrating social network structures. It features several groups of stylized human figures (blue and grey) arranged on a white background. The groups are connected by lines of different colors (red, green, yellow, purple) and are surrounded by concentric circles of the same color, suggesting different levels of social interaction or community membership. The groups are arranged in a way that shows how individuals are connected to different clusters within the network.



谢谢！

