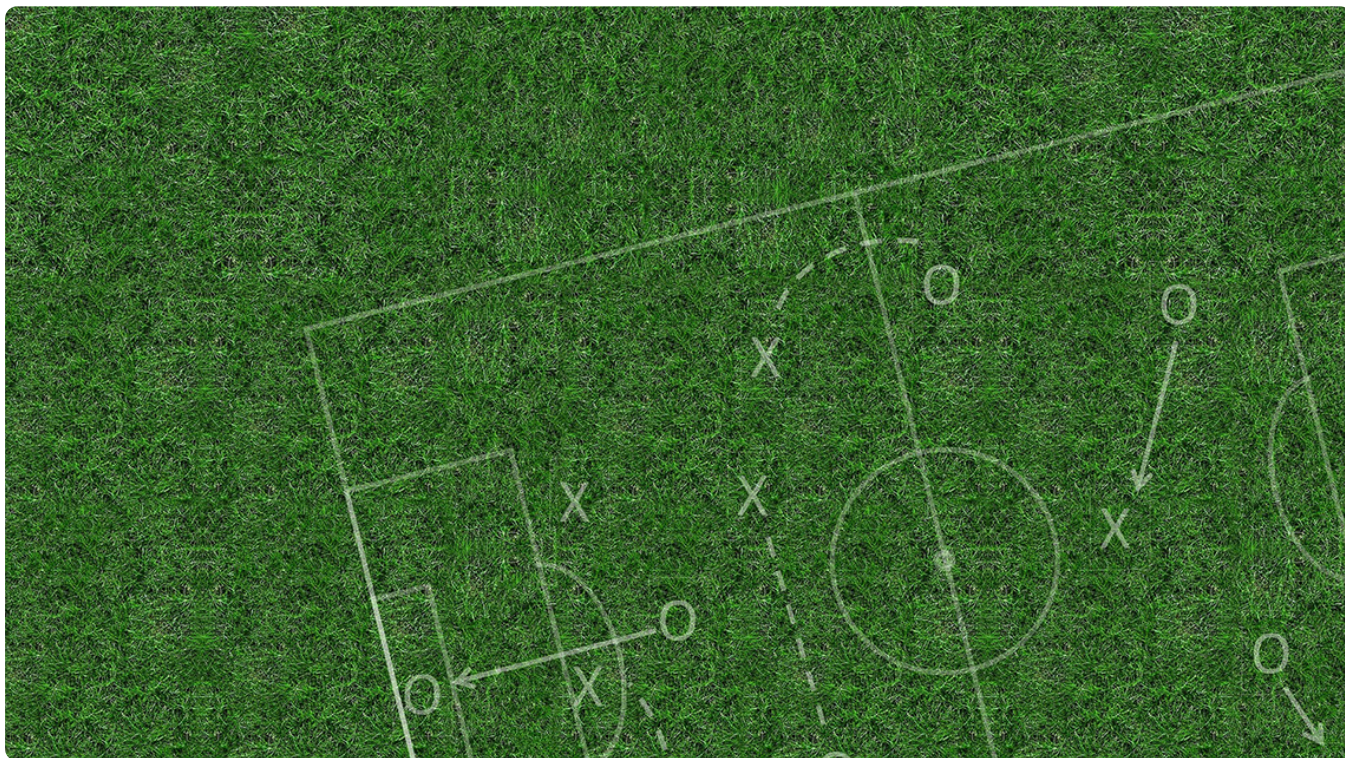


## 26 | K-Means (上) : 如何给20支亚洲球队做聚类?

2019-02-11 陈旸



讲述：陈旸

时长 10:08 大小 9.29M



今天我来带你进行 K-Means 的学习。K-Means 是一种非监督学习，解决的是聚类问题。K 代表的是 K 类，Means 代表的是中心，你可以理解这个算法的本质是确定 K 类的中心点，当你找到了这些中心点，也就完成了聚类。

那么请你和我思考以下三个问题：

如何确定 K 类的中心点？

如何将其他点划分到 K 类中？

如何区分 K-Means 与 KNN？

如果理解了上面这 3 个问题，那么对 K-Means 的原理掌握得也就差不多了。

先请你和我思考一个场景，假设我有 20 支亚洲足球队，想要将它们按照成绩划分成 3 个等级，可以怎样划分？

## K-Means 的工作原理

对亚洲足球队的水平，你可能也有自己的判断。比如一流的亚洲球队有谁？你可能会说伊朗或韩国。二流的亚洲球队呢？你可能说是中国。三流的亚洲球队呢？你可能会说越南。

其实这些都是靠我们的经验来划分的，那么伊朗、中国、越南可以说是三个等级的典型代表，也就是我们每个类的中心点。

所以回过头来，如何确定 K 类的中心点？一开始我们是随机指派的，当你确认了中心点后，就可以按照距离将其他足球队划分到不同的类别中。

这也就是 K-Means 的中心思想，就是这么简单直接。你可能会问：如果一开始，选择一流球队是中国，二流球队是伊朗，三流球队是韩国，中心点选择错了怎么办？其实不用担心，K-Means 有自我纠正机制，在不断的迭代过程中，会纠正中心点。中心点在整个迭代过程中，并不是唯一的，只是你需要一个初始值，一般算法会随机设置初始的中心点。

好了，那我来把 K-Means 的工作原理给你总结下：

1. 选取 K 个点作为初始的类中心点，这些点一般都是从数据集中随机抽取的；
2. 将每个点分配到最近的类中心点，这样就形成了 K 个类，然后重新计算每个类的中心点；
3. 重复第二步，直到类不发生变化，或者你也可以设置最大迭代次数，这样即使类中心点发生变化，但是只要达到最大迭代次数就会结束。

## 如何给亚洲球队做聚类

对于机器来说需要数据才能判断类中心点，所以我整理了 2015-2019 年亚洲球队的排名，如下表所示。

我来说明一下数据概况。

其中 2019 年国际足联的世界排名，2015 年亚洲杯排名均为实际排名。2018 年世界杯中，很多球队没有进入到决赛圈，所以只有进入到决赛圈的球队才有实际的排名。如果是

亚洲区预选赛 12 强的球队，排名会设置为 40。如果没有进入亚洲区预选赛 12 强，球队排名会设置为 50。

国家	2019年国际排名	2018世界杯	2015亚洲杯
中国	73	40	7
日本	60	15	5
韩国	61	19	2
伊朗	34	18	6
沙特	67	26	10
伊拉克	91	40	4
卡塔尔	101	40	13
阿联酋	81	40	6
乌兹别克斯坦	88	40	8
泰国	122	40	17
越南	102	50	17
阿曼	87	50	12
巴林	116	50	11
朝鲜	110	50	14
印尼	164	50	17
澳洲	40	30	1
叙利亚	76	40	17
约旦	118	50	9
科威特	160	50	15
巴勒斯坦	96	50	16

针对上面的排名，我们首先需要做的是数据规范化。你可以把这些值划分到 [0,1] 或者按照均值为 0，方差为 1 的正态分布进行规范化。具体数据规范化的步骤可以看下 13 篇，也就是[数据变换](#)那一篇。

我先把数值都规范化到 [0,1] 的空间中，得到了以下的数值表：

国家	2019年国际排名	2018世界杯	2015亚洲杯
中国	0.3	0.71428571	0.375
日本	0.2	0	0.25
韩国	0.20769231	0.11428571	0.0625
伊朗	0	0.08571429	0.3125
沙特	0.25384615	0.31428571	0.5625
伊拉克	0.43846154	0.71428571	0.1875
卡塔尔	0.51538462	0.71428571	0.75
阿联酋	0.36153846	0.71428571	0.3125
乌兹别克斯坦	0.41538462	0.71428571	0.4375
泰国	0.67692308	0.71428571	1
越南	0.52307692	1	1
阿曼	0.40769231	1	0.6875
巴林	0.63076923	1	0.625
朝鲜	0.58461538	1	0.8125
印尼	1	1	1
澳洲	0.04615385	0.42857143	0
叙利亚	0.32307692	0.71428571	1
约旦	0.64615385	1	0.5
科威特	0.96923077	1	0.875
巴勒斯坦	0.47692308	1	0.9375

如果我们随机选取中国、日本、韩国为三个类的中心点，我们就需要看下这些球队到中心点的距离。

距离有多种计算的方式，有关距离的计算我在 KNN 算法中也讲到过：

欧氏距离

曼哈顿距离

切比雪夫距离

余弦距离

欧氏距离是最常用的距离计算方式，这里我选择欧氏距离作为距离的标准，计算每个队伍分别到中国、日本、韩国的距离，然后根据距离远近来划分。我们看到大部分的队，会和中国队聚类到一起。这里我整理了距离的计算过程，比如中国和中国的欧氏距离为 0，中国和日本的欧式距离为 0.732003。如果按照中国、日本、韩国为 3 个分类的中心点，欧氏距离的计算结果如下表所示：



国家	中国	日本	韩国	划分
中国	0	0.732003	0.682772	中国
日本	0.732003	0	0.219719	日本
韩国	0.682772	0.219719	0	韩国
伊朗	0.699291	0.226392	0.32627	日本
沙特	0.444169	0.446465	0.540491	中国
伊拉克	0.233083	0.755628	0.654889	中国
卡塔尔	0.432453	0.927185	0.96298	中国
阿联酋	0.087711	0.734986	0.667959	中国
乌兹别克斯坦	0.131224	0.769253	0.737402	中国
泰国	0.72986	1.140245	1.207925	中国
越南	0.72251	1.291077	1.327729	中国
阿曼	0.436906	1.1111	1.102322	中国
巴林	0.503528	1.151602	1.131322	中国
朝鲜	0.595017	1.210097	1.220271	中国
印尼	0.980947	1.484082	1.513654	中国
澳洲	0.53544	0.519463	0.358854	韩国
叙利亚	0.625426	1.043001	1.119026	中国
约旦	0.465919	1.123189	1.080807	中国
科威特	0.882894	1.407956	1.42288	中国
巴勒斯坦	0.655241	1.244726	1.273813	中国

然后我们再重新计算这三个类的中心点，如何计算呢？最简单的方式就是取平均值，然后根据新的中心点按照距离远近重新分配球队的分类，再根据球队的分类更新中心点的位置。计算过程这里不展开，最后一迭代（重复上述的计算过程：计算中心点和划分分类）到分类不再发生变化，可以得到以下的分类结果：

国家	2019年国际排名	2018世界杯	2015亚洲杯	聚类
中国	73	40	7	0
日本	60	15	5	2
韩国	61	19	2	2
伊朗	34	18	6	2
沙特	67	26	10	2
伊拉克	91	40	4	0
卡塔尔	101	40	13	1
阿联酋	81	40	6	0
乌兹别克斯坦	88	40	8	0
泰国	122	40	17	1
越南	102	50	17	1
阿曼	87	50	12	1
巴林	116	50	11	1
朝鲜	110	50	14	1
印尼	164	50	17	1
澳洲	40	30	1	2
叙利亚	76	40	17	1
约旦	118	50	9	1
科威特	160	50	15	1
巴勒斯坦	96	50	16	1

所以我们能看出来第一梯队有日本、韩国、伊朗、沙特、澳洲；第二梯队有中国、伊拉克、阿联酋、乌兹别克斯坦；第三梯队有卡塔尔、泰国、越南、阿曼、巴林、朝鲜、印尼、叙利亚、约旦、科威特和巴勒斯坦。

## 如何使用 sklearn 中的 K-Means 算法

sklearn 是 Python 的机器学习工具库，如果从功能上来划分，sklearn 可以实现分类、聚类、回归、降维、模型选择和预处理等功能。这里我们使用的是 sklearn 的聚类函数库，因此需要引用工具包，具体代码如下：

```
1 from sklearn.cluster import KMeans
```

当然 K-Means 只是 sklearn.cluster 中的一个聚类库，实际上包括 K-Means 在内，sklearn.cluster 一共提供了 9 种聚类方法，比如 Mean-shift，DBSCAN，Spectral clustering（谱聚类）等。这些聚类方法的原理和 K-Means 不同，这里不做介绍。

我们看下 K-Means 如何创建：

复制代码

```
1 KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001, precompute_d
```



我们能看到在 K-Means 类创建的过程中，有一些主要的参数：

**n\_clusters**: 即 K 值，一般需要多试一些 K 值来保证更好的聚类效果。你可以随机设置一些 K 值，然后选择聚类效果最好的作为最终的 K 值；

**max\_iter**：最大迭代次数，如果聚类很难收敛的话，设置最大迭代次数可以让我们及时得到反馈结果，否则程序运行时间会非常长；

**n\_init**：初始化中心点的运算次数，默认是 10。程序是否能快速收敛和中心点的选择关系非常大，所以在中心点选择上多花一些时间，来争取整体时间上的快速收敛还是非常值得的。由于每一次中心点都是随机生成的，这样得到的结果就有好有坏，非常不确定，所以要运行 n\_init 次，取其中最好的作为初始的中心点。如果 K 值比较大的时候，你可以适当增大 n\_init 这个值；


**init**：即初始值选择的方式，默认是采用优化过的 k-means++ 方式，你也可以自己指定中心点，或者采用 random 完全随机的方式。自己设置中心点一般是对于个性化的数据进行设置，很少采用。random 的方式则是完全随机的方式，一般推荐采用优化过的 k-means++ 方式；

**algorithm**：k-means 的实现算法，有 "auto" "full" "elkan" 三种。一般来说建议直接用默认的 "auto"。简单说下这三个取值的区别，如果你选择 "full" 采用的是传统的 K-Means 算法，"auto" 会根据数据的特点自动选择是选择 "full" 还是 "elkan"。我们一般选择默认的取值，即 "auto"。




在创建好 K-Means 类之后，就可以使用它的方法，最常用的是 fit 和 predict 这两个函数。你可以单独使用 fit 函数和 predict 函数，也可以合并使用 fit\_predict 函数。其中 fit(data) 可以对 data 数据进行 k-Means 聚类。 predict(data) 可以针对 data 中的每个样本，计算最近的类。

现在我们要完整地跑一遍 20 支亚洲球队的聚类问题。我把数据上传到了[GitHub](#)上，你可以自行下载。

 复制代码

```
1 # coding: utf-8
2 from sklearn.cluster import KMeans
3 from sklearn import preprocessing
4 import pandas as pd
5 import numpy as np
6 # 输入数据
7 data = pd.read_csv('data.csv', encoding='gbk')
8 train_x = data[["2019 年国际排名 ", "2018 世界杯 ", "2015 亚洲杯 "]]
9 df = pd.DataFrame(train_x)
10 kmeans = KMeans(n_clusters=3)
11 # 规范化到 [0,1] 空间
12 min_max_scaler=preprocessing.MinMaxScaler()
13 train_x=min_max_scaler.fit_transform(train_x)
14 # kmeans 算法
15 kmeans.fit(train_x)
16 predict_y = kmeans.predict(train_x)
17 # 合并聚类结果，插入到原数据中
18 result = pd.concat((data,pd.DataFrame(predict_y)),axis=1)
19 result.rename({0:'聚类'},axis=1,inplace=True)
20 print(result)
```

运行结果：

 复制代码

1	国家	2019 年国际排名	2018 世界杯	2015 亚洲杯	聚类
2	0	中国	73	40	7 2
3	1	日本	60	15	5 0
4	2	韩国	61	19	2 0
5	3	伊朗	34	18	6 0
6	4	沙特	67	26	10 0
7	5	伊拉克	91	40	4 2
8	6	卡塔尔	101	40	13 1
9	7	阿联酋	81	40	6 2
10	8	乌兹别克斯坦	88	40	8 2
11	9	泰国	122	40	17 1

12	10	越南	102	50	17	1
13	11	阿曼	87	50	12	1
14	12	巴林	116	50	11	1
15	13	朝鲜	110	50	14	1
16	14	印尼	164	50	17	1
17	15	澳洲	40	30	1	0
18	16	叙利亚	76	40	17	1
19	17	约旦	118	50	9	1
20	18	科威特	160	50	15	1
21	19	巴勒斯坦	96	50	16	1

## 总结

今天我给你讲了 K-Means 算法原理，我们再来看下开篇我给你提的三个问题。

如何确定 K 类的中心点？其中包括了初始的设置，以及中间迭代过程中中心点的计算。在初始设置中，会进行  $n_{init}$  次的选择，然后选择初始中心点效果最好的为初始值。在每次分类更新后，你都需要重新确认每一类的中心点，一般采用均值的方式进行确认。

如何将其他点划分到 K 类中？这里实际上是关于距离的定义，我们知道距离有多种定义的方式，在 K-Means 和 KNN 中，我们都可以采用欧氏距离、曼哈顿距离、切比雪夫距离、余弦距离等。对于点的划分，就看它离哪个类的中心点的距离最近，就属于哪一类。

如何区分 K-Means 和 KNN 这两种算法呢？刚学过 K-Means 和 KNN 算法的同学应该能知道两者的区别，但往往过了一段时间，就容易混淆。所以我们可以从三个维度来区分 K-Means 和 KNN 这两个算法：

首先，这两个算法解决数据挖掘的两类问题。K-Means 是聚类算法，KNN 是分类算法。

这两个算法分别是两种不同的学习方式。K-Means 是非监督学习，也就是不需要事先给出分类标签，而 KNN 是有监督学习，需要我们给出训练数据的分类标识。

最后，K 值的含义不同。K-Means 中的 K 值代表 K 类。KNN 中的 K 值代表 K 个最接近的邻居。



那么学完了今天的内容后，你能说一下 K-Means 的算法原理吗？如果我们把上面的 20 支亚洲球队用 K-Means 划分成 5 类，在规范化数据的时候采用标准化的方式（即均值为 0，方差为 1），该如何编写程序呢？运行的结果又是如何？

欢迎你在评论区与我分享你的答案，也欢迎点击“请朋友读”，把这篇文章分享给你的朋友或者同事。

# 数据分析实战 45 讲

即学即用的数据分析入门课

陈旻

清华大学计算机博士



新版升级：点击「👤请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得转载

上一篇 25 | KNN (下) : 如何对手写数字进行识别 ?

下一篇 27 | K-Means (下) : 如何使用K-Means对图像进行分割 ?

## 精选留言 (14)

写留言



Lee 置顶

2019-02-14

👍 3

# coding: utf-8

```
from sklearn.cluster import KMeans
```

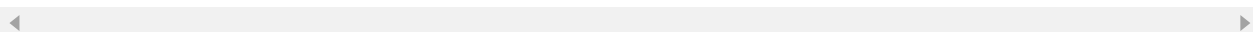
```
from sklearn import preprocessing
```

```
import pandas as pd
```

```
import numpy as np...
```

展开 ▾

编辑回复: 正确





**third** 置顶  
2019-02-19

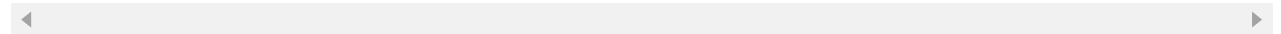
👍 2

两者的区别的比喻是，  
Kmeans开班，选老大，风水轮流转，直到选出最佳中心老大  
Knn小弟加队伍，离那个班相对近，就是那个班的

一群人的有些人想要聚在一起...

展开 ▾

编辑回复: 举例很生动，代码也正确。大家可以看下。



**白夜**  
2019-02-15

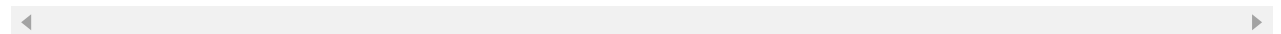
👍 2

然后我们再重新计算这三个类的中心点，如何计算呢？最简单的方式就是取平均值，然后根据新的中心点按照距离远近重新分配球队的分类，再根据球队的分类更新中心点的位置。计算过程这里不展开，最后一直迭代（重复上述的计算过程：计算中心点和划分分类）到分类不再发生变化。

...

展开 ▾

编辑回复: 同一个类别下的平均值。比如都属于同一个类别里面有10个点，那么新的中心点就是这10个点的中心点，一种简单的方式就是取平均值。比如我在文章里举了足球队例子，一共有3个指标，每个球队都有这三个指标的特征值，那么新的中心点，就是取这个类别中的这些点这三个指标特征值的平均值。。



**FORWARD M...**  
2019-02-18

👍 1

如何调整聚类中心没听懂

展开 ▾

编辑回复: 取特征值的平均值为中心点。



**fancy**

👍



2019-03-04

K-Means算法原理：

1. 初始化K类，比如K=3，K1=中国，K2=日本，K3=韩国，所以，此时这三类的中心点分别为K1=(0.3,0.71428571,0.375)  
K2=(0.2,0,0.25)  
K3=(0.20769231,0.11428571,0.0624)...

展开 ▾



**mickey**

2019-02-28



```
# coding: utf-8
from sklearn.cluster import KMeans
from sklearn import preprocessing
import pandas as pd
import numpy as np...
```

展开 ▾



**JingZ**

2019-02-26



```
import numpy as np 这行代码不需要吧~感觉没用到
```

展开 ▾



**liyooo**

2019-02-25



眼界大开！

展开 ▾



**Destroy\_**

2019-02-25



```
# 划分成 5 类
kmeans = KMeans(n_clusters=5)
# Z-Score规范化
from sklearn.preprocessing import StandardScaler
standardscaler = StandardScaler()...
```

展开 ▾



王彬成

2019-02-23



K-Means 的算法原理：

- 1、随机选择K个中心点
- 2、把每个数据点分配到离它最近的中心点；
- 3、重新计算每类中的点到该类中心点距离的平均值
- 4、分配每个数据到它最近的中心点；...

展开 ▾



Chen

2019-02-18



求教k-means可视化

展开 ▾



深白浅黑

2019-02-18



kmean是无监督的聚类算法，先对K类数量进行设置，再设置K类的中心点，算法会自行在迭代过程中对中心点进行调整，按照各点与中心点的距离进行分类划分，直至分类不变。

比较适合数据特征的归类。

国家 2019年国际排名 2018世界杯 2015亚洲杯 聚类...

展开 ▾



切克闹

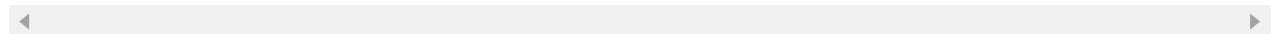
2019-02-17



K-Means中关于如何更新中心点，以及方式不太清楚

展开 ▾

编辑回复: 取特征值的平均值为中心点。



李沛欣

2019-02-15



K-means是聚类算法，属于非监督式学习，K代表了类别数；

KNN 是分类算法，属于有监督式学习，K代表了K个抱团的邻居；

看到Kmeans总会想起被因子分析法支配的恐惧，哈哈😊

展开 ▼

编辑回复: 总结正确

