**基于WEKA的数据挖掘算法学习（六）**

# 聚类

## 1.1聚类任务

在“无监督学习”中，训练样本的标记信息是未知的，目标是通过对无标记训练样本的学习来揭示数据的内在性质及规律，为进一步数据分析提供基础。聚类的目的是试图将数据集中的样本划分为若干个通常不相交的子集，每个子集称为一个“簇”。

聚类既能是一个单独过程，用于寻找数据内在的分布结构，也可作为分类等其他学习任务的前驱过程。

## 1.2 性能度量

聚类性能指标大致两类，一是将聚类结果与某个“参考模型”进行比较，称为“外部指标”，二是直接考察聚类结果而不利用任何参考模型，称为“内部指标”。

聚类性能的度量的外部指标为：

Jaccard系数：

FMI指数：

Rand指数：

性能度量结果值均在[0,1]区间，值越大越好。

聚类性能的度量的内部指标为：

DB指数：

Dunn指数：

DBI的值越小越好，而DI的值越大越好。

## 1.3 距离计算

给定样本与，最常用的是闵可夫斯基距离：

其中当p=1时为曼哈顿距离，p=2时为欧式距离。

此外，对于无序属性，使用VDM距离：

其中，表示属性u上取值为a的样本数，表示在第i个样本簇中在属性u上取值为a的样本数，k为样本簇数。

明显的，闵可夫斯基距离和VDM结合可处理混合属性，当样本空间中不同属性重要性不同时，可使用加权距离，赋予权重。

## 1.4聚类算法

### 1.4.1 k均值算法（k-means）

算法核心思想：首先选定簇数，随机选择样本作为初始均值向量，依次考察每个样本与当前均值向量均值的距离，选定距离最近的簇，归于该簇。

所有样本考察结束一轮以后，分别更新，每个簇的新的均值向量，不断重复上述过程，当均值向量未更新后得到最终的簇划分。为了防止运行时间过长，设置一个最大运行轮数或者最小调整幅度阈值。

### 1.4.2学习向量量化（LVQ）

试图找到一组原型向量来刻画聚类结构，与一般聚类算法不同，LVQ假设数据样本带有类别标记，学习过程中利用样本的监督信息来辅助聚类。

算法核心思想：首先令样本带上类别标记，设置簇数q，学习目标就是找q个原型向量，并且假定q个原型向量对应的类别标记。

根据样本的类别标记和簇预设类别标记对原型向量进行随机初始化，选择样本与某个原型向量距离最相近者，当二者具有相同（不同）的类别标记时，更新原型向量为：

不断重复上述过程，迭代不同的轮数。

### 1.4.3密度聚类（DBSCAN）

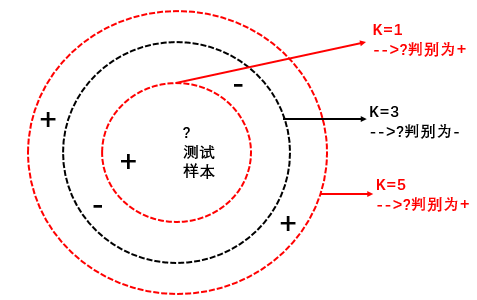
此算法假设聚类结构能通过样本分布的紧密程度确定，通常根据样本密度的角度来考察样本之间的可连接性，基于可连接样本不断扩展聚类簇以获得最终的聚类结果。

DBSCAN是著名的一种密度聚类算法，基于领域参数来刻画样本分布的紧密程度。

算法核心思想：随机选择数据集中的一个核心对象作为“种子”，由此出发确定相应的聚类簇。根据给定领域参数（e，MinPts）找到所有的核心对象，以任意核心对象出发，找出由密度可达的样本生成聚类簇，直到所有核心对象均被访问过。最终得到簇划分。

# K近邻学习

简称KNN，是一种常用的监督学习的方法。算法核心思想：一个样本在特征空间中K个最相邻样本中的大多数样本属于某一类别，则该样本属于这个类别。



**图1 k近邻分类示意图**

由图1k近邻分类器示意图，可以清晰地看出测试样本在k=1和k=5时被判别为正例，在k=3时被判别为反例。也完美体现了k近邻算法的核心思想。

# 3.weka算法实现

## 3.1数据集的准备

1、UCI完备数据集：segmentation csv

2、UCI不完备数据集：hepatitis.csv

3、UCI不平衡数据集： balance-scale.csv

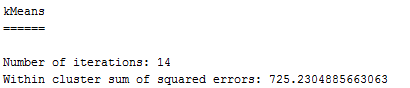
## 3.2 K-means

选择Cluster🡪SimpleKMeans

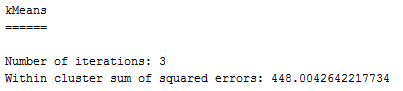
设置距离函数distanceFunction=EuclideanDistance，设置numCluster=7不断改变seed值，进行数据测试，结果如下：

**1、UCI完备数据集：segmentation**

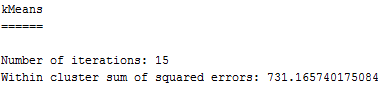
**Seed=5**时：



**Seed=7**时：

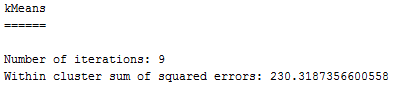


**Seed=9**时：

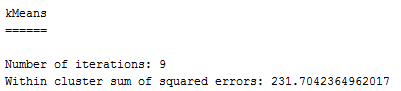


**2、UCI不完备数据集：hepatitis**

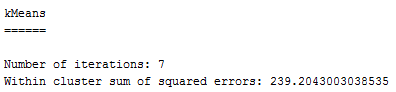
**Seed=5**时：



**Seed=7**时：

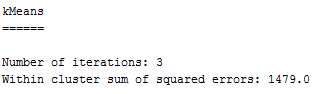


**Seed=9**时：

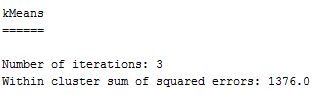


**3、UCI不平衡数据集：balance-scale.**

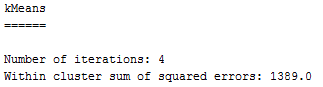
**Seed=5**时：



**Seed=7**时：



**Seed=9**时：

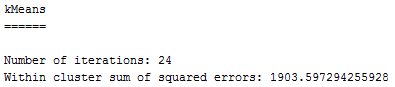


选择Cluster🡪SimpleKMeans

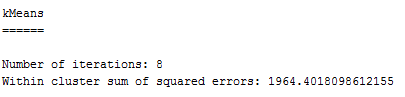
设置距离函数distanceFunction=EuclideanDistance，设置numCluster=3不断改变seed值，进行数据测试，结果如下：

**1、UCI完备数据集：segmentation**

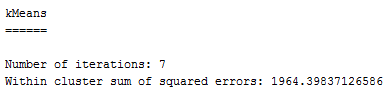
**Seed=5**时：



**Seed=7**时：

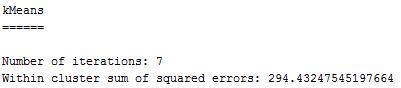


**Seed=9**时：

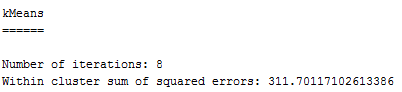


**2、UCI不完备数据集：hepatitis**

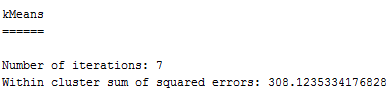
**Seed=5**时：



**Seed=7**时：

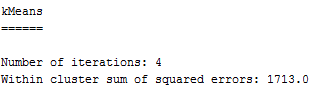


**Seed=9**时：

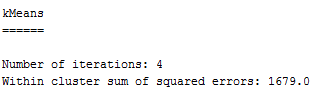


**3、UCI不完备数据集：balance-scale.**

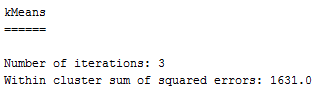
**Seed=5**时：



**Seed=7**时：



**Seed=9**时：

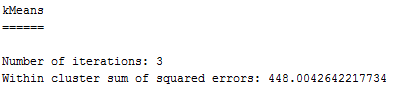


Within cluster sum of squared errors值越小，那么，就说明同一簇实例之间的距离越小，聚类的结果也就越好。根据跑出来的实验数据结果，当K增加时，较大的聚类将被分解为更小的部分。聚类越小，每个点离它的聚类均值越近（平均）。

且由数据分类结果可以知道，三种不同类型的数据集，有各自不同的适合自己的分类簇数，如完备数据集segmentation.csv在numCluster=7时，seed=7有最佳的实验结果。不断调整不同数据集参数numCluster和seed将会有很大的差别。

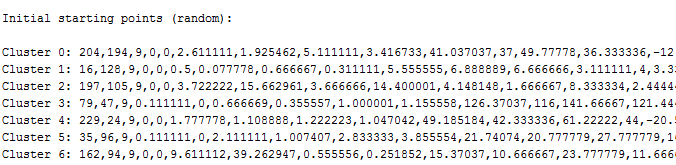
**这里选择完备数据集segmentation.csv在numCluster=7时，seed=7进行分析：**

**1、Within cluster sum of squared errors: 448.0042642217734**



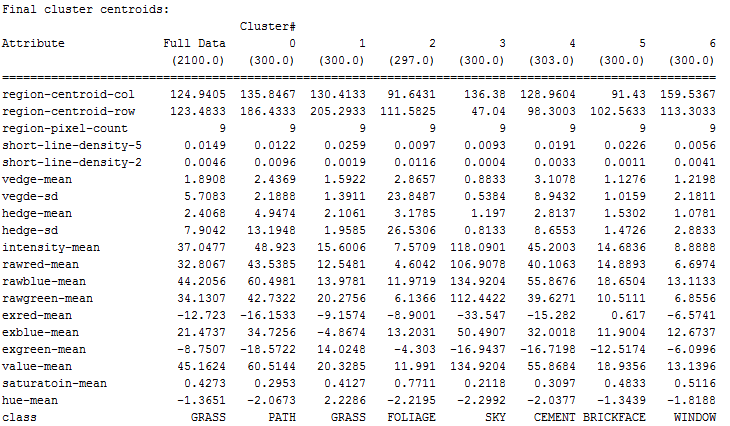
前面也提到，这是评价聚类好坏的标准，数值越小说明同一簇实例之间的距离越小。将seed值进行更改，就会得到不同的结果，我们应该尝试不同的seed值，并且选取这个数值最小的结果，多尝试不同的seed值，或许比448.0042642217734更小的结果可以得到。

**2、初始化簇中心：**



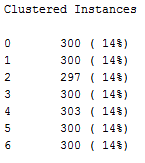
这里看到有七类，k-means是随机选择初始化簇中心。

**3、簇中心：**



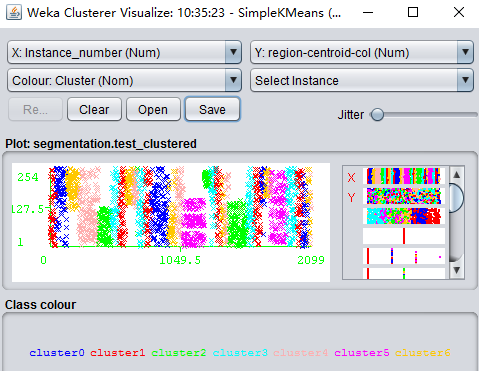
Cluster centroids列出了各个簇中心的位置。对于数值型的属性，簇中心就是它的均值（Mean）；分类的就是它的众数，也就是说这个属性上取值为众数值的实例最多。

**4、Clustered Instances：**

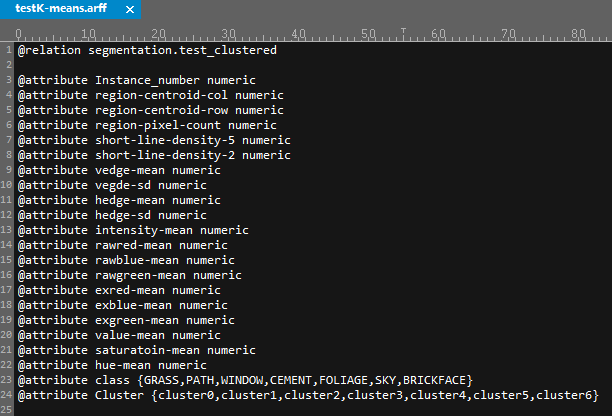


这里展示的是各个簇中实例的数目及百分比，这里没有未聚类进簇的样本（k-means对离群点很敏感）。、

**5、 weka cluster visualize：**



为了分析可视化结果，可以在weka cluster visualize里面看见聚类结果，颜色不同代表每个类别的聚类簇，点击这里的save按钮，可以将这里跑出来的聚类结果保存为.arff文件，这里将聚类结果保存为：testK-means.arff文件，打开文件：



可以看见，在这个新生成的聚类结果ARFF文件中，比最初的数据文件多了两个属性分别是：instance\_number属性表示某实例的编号，Cluster属性表示聚类算法给出的该实例所在的簇。

## 3.3 DBSCAN

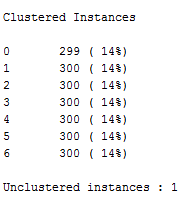
与K-means不同的是，DBSCAN算法的簇数由算法自动确定。

选择Cluster🡪DBCSAN

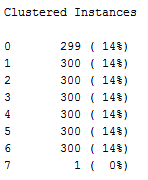
设置距离函数distanceFunction=EuclideanDistance，不断改变epsilon(默认为0.6)值和minPoints（默认为6）（这两个参数为邻域参数，是影响DBSCAN很大的因素，由此决定算法的核心对象）进行数据测试，结果如下：

**1、UCI完备数据集：segmentation**

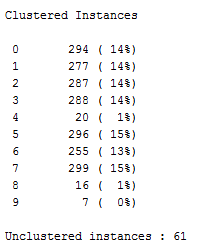
**epsilon=0.9，minPoints=6**时：



**epsilon=0.9，minPoints=1**时：

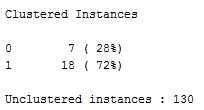


**epsilon=0.4，minPoints=6**时：

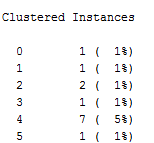


**2、UCI不完备数据集：hepatitis.**

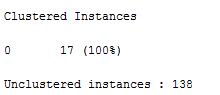
**epsilon=0.9，minPoints=6**时：



**epsilon=0.9，minPoints=1**时：

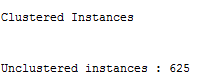


**epsilon=0.4，minPoints=6**时：

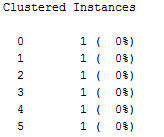


**3、UCI不平衡数据集：balance-scale**

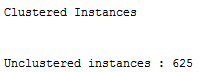
**epsilon=0.9，minPoints=6**时：



**epsilon=0.9，minPoints=1**时：



**epsilon=0.4，minPoints=6**时：



epsilon和minPoints为邻域参数，是影响DBSCAN最大的因素，由此决定算法的核心对象。并且，由实验结果也可以看见，调整参数时，数据集的分类结果会产生不同的结果，以完备数据集segmentation为例，epsilon=0.9，minPoints=6时，分类结果为7类，有一个样本未分类，相当于样本在这个参数下的离群值。此外，对于不平衡数据集**balance-scale**需要在设置邻域参数较大的情况下有比较好的聚类效果，展示的结果（参数设置为了更好的进行对比）来看均不能进入聚类簇中，这也是需要注意的地方。

## 3.4 KNN

选择Classify🡪IBK，选择五折交叉验证，不断改变KNN的值（默认为1）,进行数据测试，结果如下：

当KNN= 1时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.964 | 0.964 | 0.964 | 0.979 | 96.43% | 0.02s |
| hepatitis | 0.793 | 0.800 | 0.796 | 0.677 | 80% | 0s |
| Balance-scale | 0.745 | 0.808 | 0.775 | 0.883 | 80.8% | 0s |

当KNN= 3时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.952 | 0.951 | 0.951 | 0.987 | 95.10% | 0s |
| hepatitis | 0.805 | 0.819 | 0.809 | 0.786 | 81.94% | 0s |
| Balance-scale | 0.745 | 0.808 | 0.775 | 0.883 | 80.8% | 0s |

当KNN= 5时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.948 | 0.947 | 0.947 | 0.991 | 94.67% | 0s |
| hepatitis | 0.828 | 0.839 | 0.831 | 0.806 | 83.87% | 0.02s |
| Balance-scale | 0.745 | 0.808 | 0.775 | 0.883 | 80.8% | 0s |

当KNN= 7时：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | Precision | Recall值 | F值 | AUC值 | Correctly | Time |
| segment | 0.944 | 0.943 | 0.943 | 0.993 | 94.29% | 0s |
| hepatitis | 0.828 | 0.839 | 0.831 | 0.798 | 83.87% | 0s |
| Balance-scale | 0.745 | 0.808 | 0.775 | 0.883 | 80.8% | 0s |

KNN算法中，调节k值可以得到不同的结果，准确率也会有很大的不同。K值减小，学习的近似误差会减小，但模型会变得复杂；K值增加，学习的近似误差会增加，只是模型会简单。通常会选择k值很小的情况，因为准确率更高。

从实验结果来看，segment在k值为1时效果最好，而不平衡数据集balance-scale则基本没有什么变化，正好印证了采样KNN算法会很好的避免样本不平衡问题。

# 4.本周学习心得体会

本周学习了《机器学习 周志华版》中的聚类、降维与度量学习的理论知识，并分别就聚类算法中的K-means、DBACAN以及降维与度量学习中的KNN进行数据测试。聚类里面算法比较多，有原型聚类、密度聚类、层次聚类，其中原型聚类算法对原型进行初始化，然后对原型进行迭代更新求解；密度聚类基于邻域参数刻画样本分布的紧密程度；层次聚类在不同层次对数据集进行划分，形成树形的聚类结构。总的来说聚类算法相对比较简单。