【算法总结（第2期）】数据挖掘十大算法—— Kmeans

原创 稀饭的写作小屋 稀饭居然不在家 2019-07-03

收录于话题
#数据分析算法总结

24个

十大算法 —— K均值聚类

1、基本介绍

（1）概述：K-均值聚类是一种动态聚类的方法。其主要适用于分类问题。该算法给出一组对象（记录），聚类或分类的目标是把这些对象分割成组或集群，使得这些对象相比于组间，在组内更趋于相似。K-均值聚类是一种无监督学习的方法，因为不需要事先标记的数据。K-均值算法在实践中容易实施和运行，速度相对较快，算法内容也非常容易修改。

（2）优点

[1] 算法简单、迅速；

[2] 对于处理大数据集，该算法是相对可伸缩和高效的，因为它的复杂度大约是O（nkt）。其中n是所有对象的数目，k是分类的数目，t是迭代的次数，该算法经常以局部最优结束；

[3] 当类是密集、球状或者团状，且类与类之间区别明显时，该算法聚类效果很好。

（3）缺点

[1] 该算法只有在类的平均值被定义的情况下才能使用，不适用于某些分类属性的数据；

[2] 对初值比较敏感，对于不同的初始值可能会导致不同的聚类结果；

[3] 不适合于发现非凸面形状的类，或者大小差别很大的类；

[4] 对于“噪声”和孤立点数据敏感，少量的该类数据能够对平均值产生极大影响。

（4）该算法使用时的一些注意事项

[1] 算法中的K值需要认真选取；

[2] 要慎重选取初始的聚类中心，如果选择不当可能很容易陷入局部最优；

[3] 样本要随机选取，可以提高算法的收敛速度。

2、算法流程

（1）问题说明

【已知】：样本集 $X=(x_1,x_2,\ldots\ldots,x_n)$ 中每一个特征向量 x_i $i=1,2,\ldots\ldots,n$ 的情况。

【待求】：将样本集 $X=(x_1,x_2,\ldots\ldots,x_n)$ 进行分类。

（2）算法步骤（文字描述版）

[1] 第一步：判断样本集可以分为几类，设定好类个数 k；

[2] 第二步：在样本集 X 中，随机选择 k 个数据点作为初始聚类的中心；

[3] 第三步：计算样本集中每一个数据点到这 k 个聚类中心的距离，一共 nk 个距离；

[4] 第四步：将每个数据点归到离它最近的聚类中心的类别中，重复 n 次，直到每一个数据点都进行了归类（对于已经设定为类中心的点，其到它自己的距离最小，为 0）；

[5] 第五步：待所有样本点归类完成后，重新计算每一类的中心，并计算误差衡量指标；

[6] 第六步：比较误差衡量指标是否在给定阈值内，如果小于等于阈值，输出分类结果；如果大于阈值，以新得到的聚类中心，重复“第三步 → 第五步”，直到收敛。

（3）算法步骤（数学描述版）

[1] 第一步： 输入样本集 $X=(x_1,x_2,\ldots\ldots,x_n)$，设定聚类个数 k，$t=0$；

[2] 第二步： 随机选取 $X^{(t)}=(x_1^t,x_2^t,\ldots\ldots,x_k^t)$ 为初始的聚类中心，设定迭代次数上限 N，设定误差收敛条件 $\varepsilon>0$；

【注】：这里 x_i^t 不一定等于 x_i，x_i^t 是从 X 中随机抽取出来的。

[3] 第三步： 从 $t=0$ 到 $t=N$，循环下面几步，直到收敛：

① 计算 $X=(x_1,x_2,\ldots\ldots,x_n)$ 中每一个点到 $X^{(t)}=(x_1^t,x_2^t,\ldots\ldots,x_k^t)$ 中每一个中心的距离；

② 计算新的中心 $X^{(t+1)}=(x_1^{t+1},x_2^{t+1},\ldots\ldots,x_k^{t+1})$ 和误差衡量指标。常用 $x_i^{(t+1)}=\dfrac{\sum x_{n_i}}{n_i}$（$n_i$ 表示归在第 i 类的所有数据点的个数，$\sum x_{n_i}$ 表示第 i 类的所有数据点的特征（数值）之和），误差衡量指标常用 $\sum\sum\|x_{n_i}-x_i^t\|^2$，即每一个类的误差平方和（该类中所有点到类中心的距离的平方）的加总；

③ 判断误差衡量指标是否≤ ε。如果满足条件，输出聚类情况。如果不满足条件，$t=t+1$，以新的聚类中心 $X^{(t+1)}=(x_1^{t+1},x_2^{t+1},\ldots\ldots,x_k^{t+1})$ 代回第一步.

[4] 第四步： 输出最终的分类情况 $G=(G_1,G_2,\ldots\ldots,G_k)$。

3、详细例子

（1）例子一（靠嘴模拟）

【目标】：已知小明班上 30 名同学的成绩特征向量 $X=(x_1,x_2,\ldots\ldots,x_{30})$（语文成绩、数学成绩、英语成绩），将小明班上的同学进行分类。

【操作】

[1] 第一步：考虑学霸、普通人、学渣三种分类，设定 $k=3$；

[2] 第二步：从 $X=(x_1,x_2,\ldots\ldots,x_{30})$ 中选出 3 个数据点作为初始聚类中心，假设选出来的 3 个数据点是 $\mu^1=x_3$、$\mu^2=x_{11}$ 和 $\mu^3=x_{23}$；

[3] 第三步：计算 x_i 到 μ^1 的距离 d_i^1，同理得到 d_i^2 和 d_i^3，这里 $i=1,2,\ldots\ldots,30$，一共得到 $3\times30=90$ 个距离；

[4] 第四步：对每个 x_i，计算 $\min\{d_i^1,d_i^2,d_i^3\}$，则 x_i 的类别为 $\min\{d_i^1,d_i^2,d_i^3\}$ 的类别（即计算得到 $\min\{d_i^1,d_i^2,d_i^3\}=d_i^2$，则 x_i 归于类别 G_2）；

[5] 第五步：待所有 x_i 都归类之后，计算误差衡量指标。如果误差衡量指标小于等于给定阈值，则输出分类结果。如果误差衡量指标大于给定阈值，则计算新的聚类中心，并返回第三步，直到收敛。

（2）例子二（R语言实操）

[1] 代码

```
library(class)    # 加载class包

k1<-kmeans(as.matrix(iris[,1:4]),center=3)    # 使用class包对iris数据集进行聚类，

k1  # 查看聚类的效果，between_SS / total_SS这一项>80%，表示聚类效果不错

table(iris[which(k1$cluster==1),5])    # 查看聚类分在第一类的实际情况

table(iris[which(k1$cluster==2),5])    # 查看聚类分在第二类的实际情况

table(iris[which(k1$cluster==3),5])    # 查看聚类分在第三类的实际情况

```

[2] 结果

```
> library(class)    # 加载class包
>
> kl<-kmeans(as.matrix(iris[,1:4]),center=3)    # 使用class包对iris数据集进行$
>
> kl  # 查看聚类的效果，between_SS / total_SS这一项>80%，表示聚类效果不错
K-means clustering with 3 clusters of sizes 62, 38, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.901613    2.748387     4.393548    1.433871
2     6.850000    3.073684     5.742105    2.071053
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [37] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [73] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2
[109] 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2
[145] 2 2 1 2 2 1

Within cluster sum of squares by cluster:
[1] 39.82097 23.87947 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
>
> table(iris[which(kl$cluster==1),5])    # 查看聚类分在第一类的实际情况

    setosa versicolor  virginica
         0         48         14
>
> table(iris[which(kl$cluster==2),5])    # 查看聚类分在第二类的实际情况

    setosa versicolor  virginica
         0          2         36
>
> table(iris[which(kl$cluster==3),5])    # 查看聚类分在第三类的实际情况

    setosa versicolor  virginica
        50          0          0
```

从结果来看，还是有一定的分类错误，比如分在第一类和分在第二类的就有不同的实际类别情况。这可能是由于初始的类别中心选择不当，或者算法流程本身的问题所导致的，存在一定的误差~