机器学习 作业五

Hugo Zhang

1

DBSCAN算法如下：

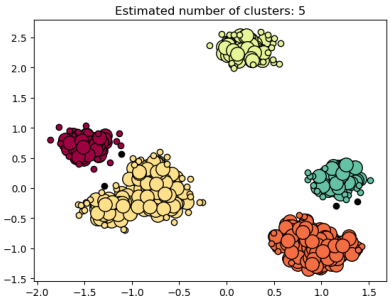
Algorithm DBSCAN Algorithm

Require:
训练数据集$D = \{x_i\}_{i=1}^N, x_i \in \mathcal{X} \subseteq R^d$;
邻域半径$\epsilon > 0$;
密度阈值$MinPts > 0$;
Ensure:
每个点属于的簇
将所有样本标记为unvisited
簇集合为$\Omega = \emptyset$
while 存在节点标记是unvisited **do**
  随机选择一个unvisited的点x_i 标记为visited
  找到其$\epsilon-$ 邻域为$N_\epsilon(x_i) = \{x_j \in D | dist(x_i, x_j) \leq \epsilon\}$
  if 邻域内包含不少于$MinPts$ 个样本点 **then**
    创建一个簇$C = x_i$
    令N 为邻域里unvisited点的集合
    while N不为空 **do**
      取出一个点x_j标记为visited，令$N = N - \{x_j\}$，令$C = C \bigcup x_j$
      找到x_j 的$\epsilon-$ 邻域N'
      if 该邻域内包含不少于$MinPts$ 个样本点 **then**
        将该邻域内unvisited的点放入N
      end if
    end while
    获得的新的簇C 放入Ω
  else
    将x_i 标记为噪声
  end if
  如果其邻域内包含不少于$MinPts$ 个样本点，则x_i 为核心对象
end while
return 簇的集合Ω;

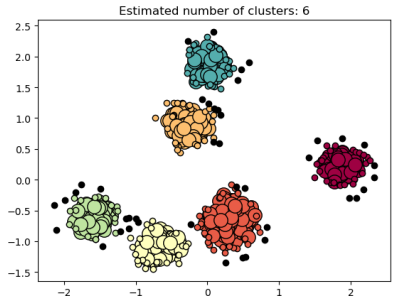
为了验证dbscan算法，我使用生成的数据集进行尝试，首先随机生成2到10个blob然后用本算法进行聚类，进行5次得到的实验结果如下。

blob	cluster	homogeneity	completeness	v-measure	adjusted-rand	mutual-info	silhouette
9	5	0.665	0.982	0.793	0.521	0.661	0.671
7	6	0.848	0.886	0.866	0.787	0.846	0.631
4	3	0.749	0.984	0.850	0.710	0.748	0.771
10	8	0.853	0.982	0.913	0.726	0.850	0.742
2	2	0.984	0.893	0.936	0.968	0.893	0.786

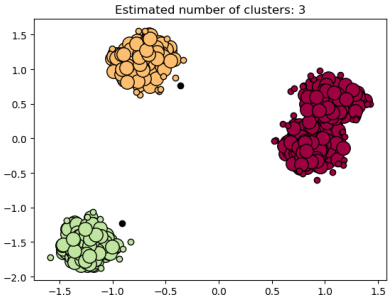
可见dbscan可以取得一定的结果，并且可以发现任意形状的簇，如果两个blob很近很可能会被认为是同一个cluster。如果我们对信息没有任何先验的信息，可以用dbscan发现一些数据规律。可视化结果如下所示。



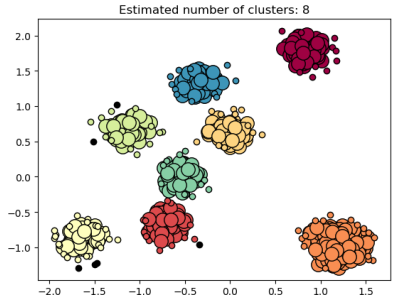
Trial 1



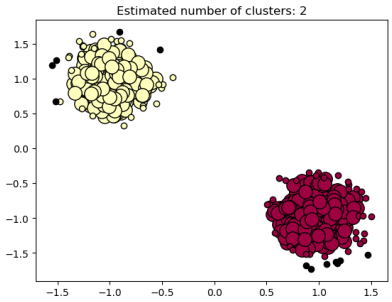
Trial 2



Trial 3



Trial 4



Trial 5

2

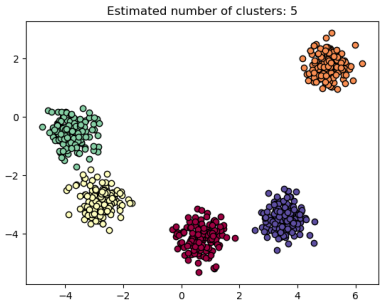
一般而言只有基于原型的聚类方法需要提前确定好簇数k。如果根据特定的任务我们已经知道k的值，那么可以直接确定k；但如果并不能提前知道k的值，要确定k一般有很多方法，例如elbow method、使用AIC或BIC或DIC标准、silhouette method、cross validation等。本文尝试使用silhouette和BIC方法来确定k簇数。

此处仍使用生成数据集，算法方面使用kmeans，每次实验生成2到10个blob，进行三次实验的数据如下所示，表格中的数分别表示silhouette(BIC)：

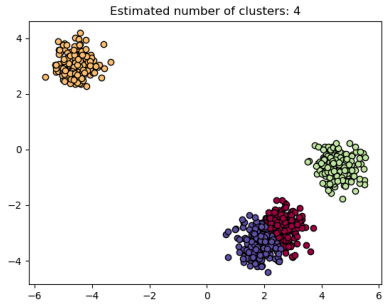
k=blob-2	k=blob-1	k=blob	k=blob+1	k=blob+2
0.720(-2863.752)	0.734(-2597.013)	0.764(-1973.084)	0.677(-2025.600)	0.568(-2085.840)
0.823(-2682.519)	0.794(-1747.516)	0.624(-1740.246)	0.581(-1803.127)	0.597(-1837.502)
0.657(-2779.788)	0.684(-2564.994)	0.664(-2518.266)	0.642(-2547.061)	0.580(-2577.481)

根据以上结果，可以发现BIC对cluster数的选择较为精准，三次实验中都是k=blob时BIC最高；silhouette方法在有些blob较为接近时无法发现问题。当然，有可能是BIC更适合随机生成blob时的场景。对于具体聚类应用场景还应根据先验知识选择不同的方法。

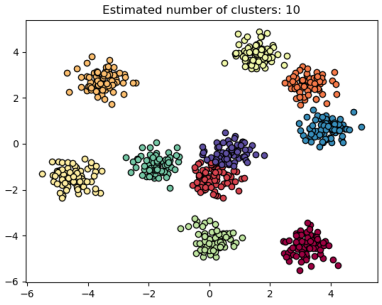
作为参考，可视化结果如下所示。



Trial 1



Trial 2



Trial 3