分类实践——KNN

李蕴哲

liyunzhe@whu.edu.cn

2019 年 10 月 31 日

方法：KNN

kNN 算法的核心思想是如果一个样本在特征空间中的 k 个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性。该方法在确定分类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。kNN 方法在类别决策时，只与极少量的相邻样本有关。

KNN 代码

实现代码：

testing

```
1 def distance(k, X_train, Y_train, x):
2 assert 1 <= k <= X_train.shape[0], "K must be valid"
3 assert X_train.shape[0] == Y_train.shape[0], "the size of
      X_train must equal to the size of y_train"
4 assert X_train.shape[1] == x.shape[0], "the feature number
       of x must be equal to X_train"
5 distance = [np.sum(abs(x_train - x)) for x_train in
      X_train]
6 nearest = np.argsort(distance)
7 topk_y = [Y_train[i] for i in nearest[:k]]
8 votes = Counter(topk_y)
9 return votes.most_common(1)[0][0]
```

观察到数据是离散值，因此采用了曼哈顿距离，但是并没有任何的作用。

数据预处理

数据集没有换，从上回的讨论中已经可以看出降维到第一维之后效果奇佳，但是这次分类，我降维到了三维，以获得更多表现的数据，效果也有所提升。
k 的数量选的是 3，因为这样的 k 最简单也最方便进行选举。

效果

最后的效果 NMI 为 0.83：

```
def distance(k, X_train, Y_train, x):
    assert 1 <= k <= X_train.shape[0], "K must be valid"
    assert X_train.shape[0] == Y_train.shape[0], "the size of X_train must equal to the size of y_train"
    assert X_train.shape[1] == x.shape[0], "the feature number of x must be equal to X_train"
    distance = [np.sum(abs(x_train - x)) for x_train in X_train]
    nearest = np.argsort(distance)
    topk_y = [Y_train[i] for i in nearest[:k]]
    votes = Counter(topk_y)
    return votes.most_common(1)[0][0]


if __name__ == "__main__":
    for xr in x_check:
    #x = np.array([8,4,5,1,2,1,7,3,1])
        label = distance(3, x_study, labels, xr)
        #print(label)

        total.append(label)
    #print(total)

    result_NMI=metrics.normalized_mutual_info_score(label_check, total)
    print("result_NMI:",result_NMI)
```

```
result_NMI: 0.8391705180325764
```

感谢聆听