

# 数据挖掘互评作业四：离群点分析与异常检测

赵柏翔 3120195512

代码地址: <https://github.com/LawrenceZhao9676/outlierdetection>

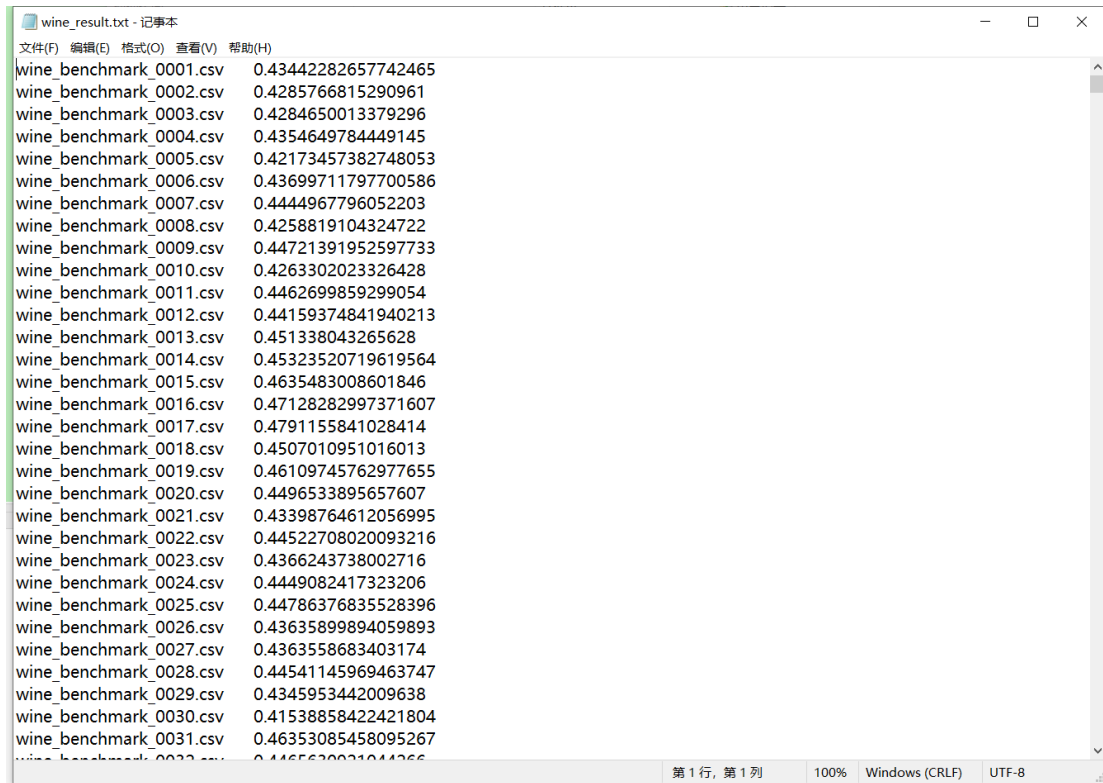
## 一、wine-benchmarks 分析

### 1. wine.py

wine.py 用来对 benchmarks 中每个数据集进行分析，分别使用 KNN、OCSVM 和 HBOS 三种方法进行异常检测，后续代码均以 HBOS 为例。

```
#导入需要的包
import pandas as pd
import numpy as np
import glob,os
from pyod.models.hbos import HBOS
#创建一个文件保存分析结果
result=open('wine_result.txt','a')
#导入数据集路径
path=r'.\wine\benchmarks'
file=os.listdir(path)
i=0
#逐个读取数据并分析
for f in file:
    df=pd.read_csv(path+'\\'+f)
    #提取出数据标签
    label=df['ground.truth']
    #删除对分析无意义的列
    df=df.drop(['ground.truth','point.id','motherset','origin'],axis=1)
    #使用 HBOS 来获取离群点
    clf = HBOS()
    score=clf.fit_predict_score(df, label)
    #将结果写入文件以便后续分析
    inputf=str(f)+'\t'+str(score)+'\n'
    result.write(inputf)
    i+=1
    if i%100==0:
        print(str(i)+'\n')
result.close()
```

结果如下：



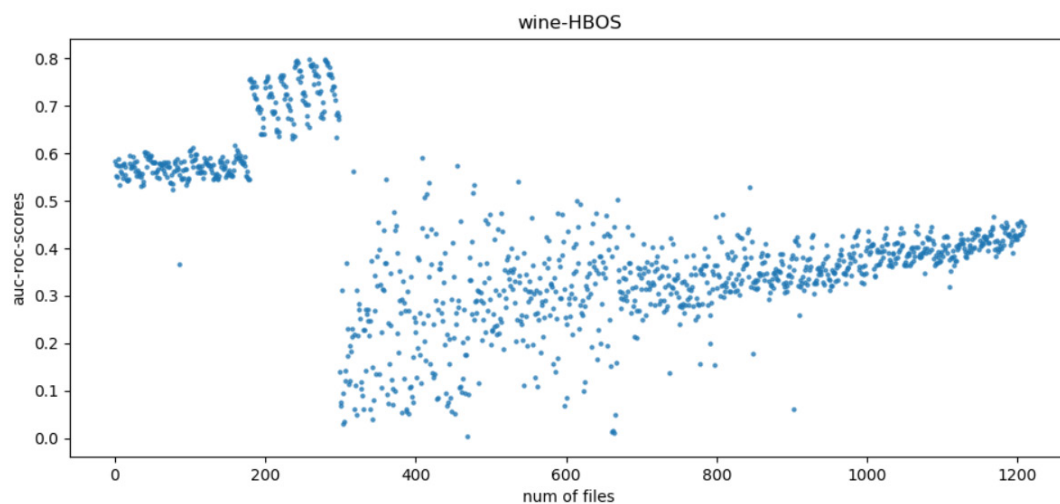
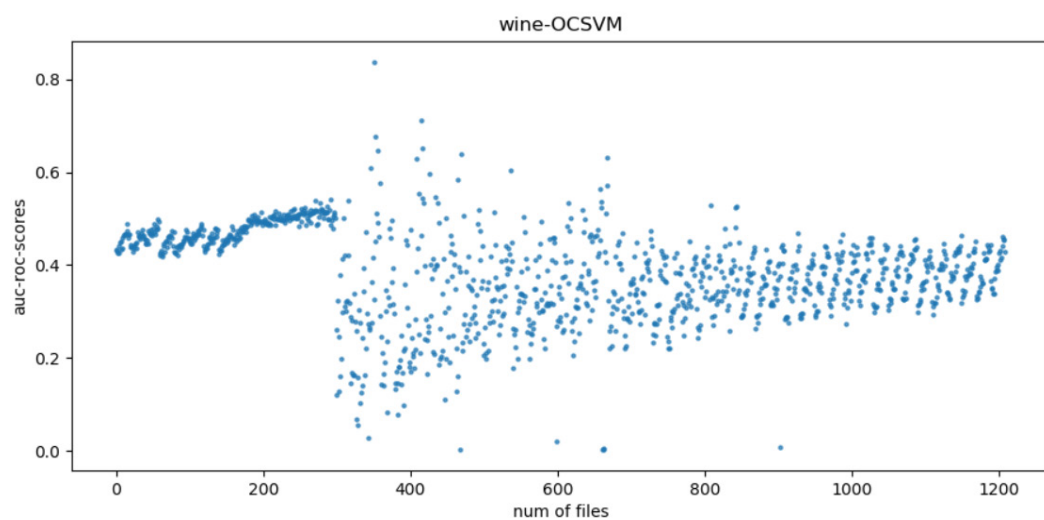
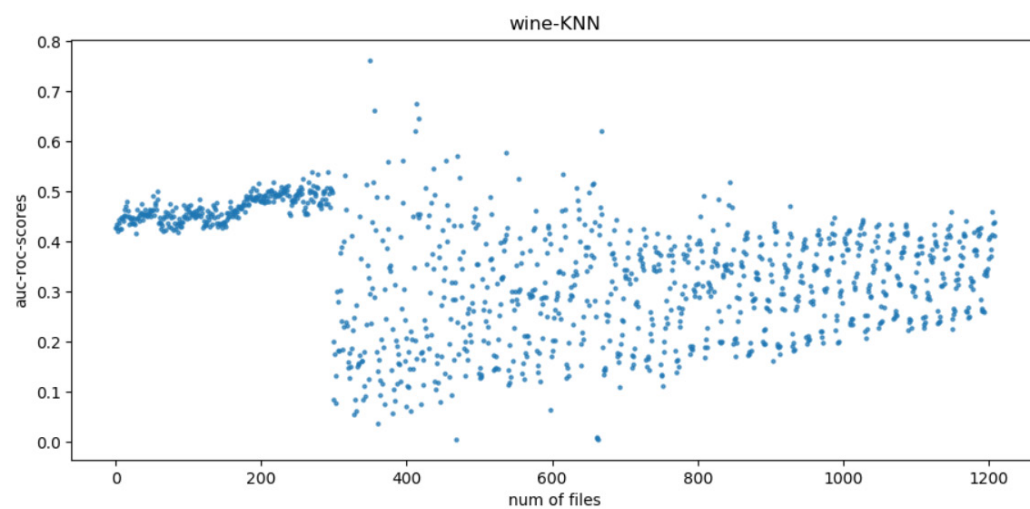
wine_benchmark_0001.csv	0.43442282657742465
wine_benchmark_0002.csv	0.4285766815290961
wine_benchmark_0003.csv	0.4284650013379296
wine_benchmark_0004.csv	0.4354649784449145
wine_benchmark_0005.csv	0.42173457382748053
wine_benchmark_0006.csv	0.43699711797700586
wine_benchmark_0007.csv	0.4444967796052203
wine_benchmark_0008.csv	0.4258819104324722
wine_benchmark_0009.csv	0.44721391952597733
wine_benchmark_0010.csv	0.4263302023326428
wine_benchmark_0011.csv	0.4462699859299054
wine_benchmark_0012.csv	0.44159374841940213
wine_benchmark_0013.csv	0.451338043265628
wine_benchmark_0014.csv	0.45323520719619564
wine_benchmark_0015.csv	0.4635483008601846
wine_benchmark_0016.csv	0.47128282997371607
wine_benchmark_0017.csv	0.4791155841028414
wine_benchmark_0018.csv	0.4507010951016013
wine_benchmark_0019.csv	0.46109745762977655
wine_benchmark_0020.csv	0.4496533895657607
wine_benchmark_0021.csv	0.43398764612056995
wine_benchmark_0022.csv	0.44522708020093216
wine_benchmark_0023.csv	0.4366243738002716
wine_benchmark_0024.csv	0.4449082417323206
wine_benchmark_0025.csv	0.44786376835528396
wine_benchmark_0026.csv	0.43635899894059893
wine_benchmark_0027.csv	0.4363558683403174
wine_benchmark_0028.csv	0.44541145969463747
wine_benchmark_0029.csv	0.4345953442009638
wine_benchmark_0030.csv	0.41538858422421804
wine_benchmark_0031.csv	0.46353085458095267
wine_benchmark_0032.csv	0.445630031044366

## 2. wine-result.py

wine-result.py 用来对结果进行可视化

```
#导入需要的包
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#读取 HBOS 分析的结果
df=pd.read_csv('wine_result.txt',delimiter='\t')
df.columns=['num','result']
plt.figure()
#提取文件中结果，并绘制曲线
plt.plot(df['result'])
plt.xlabel("num of files")
plt.ylabel("auc-roc-scores")
plt.title("wine")
plt.show()
```

**KNN、OCSVM 和 HBOS 三种方法进行异常检测，可视化结果如下：**



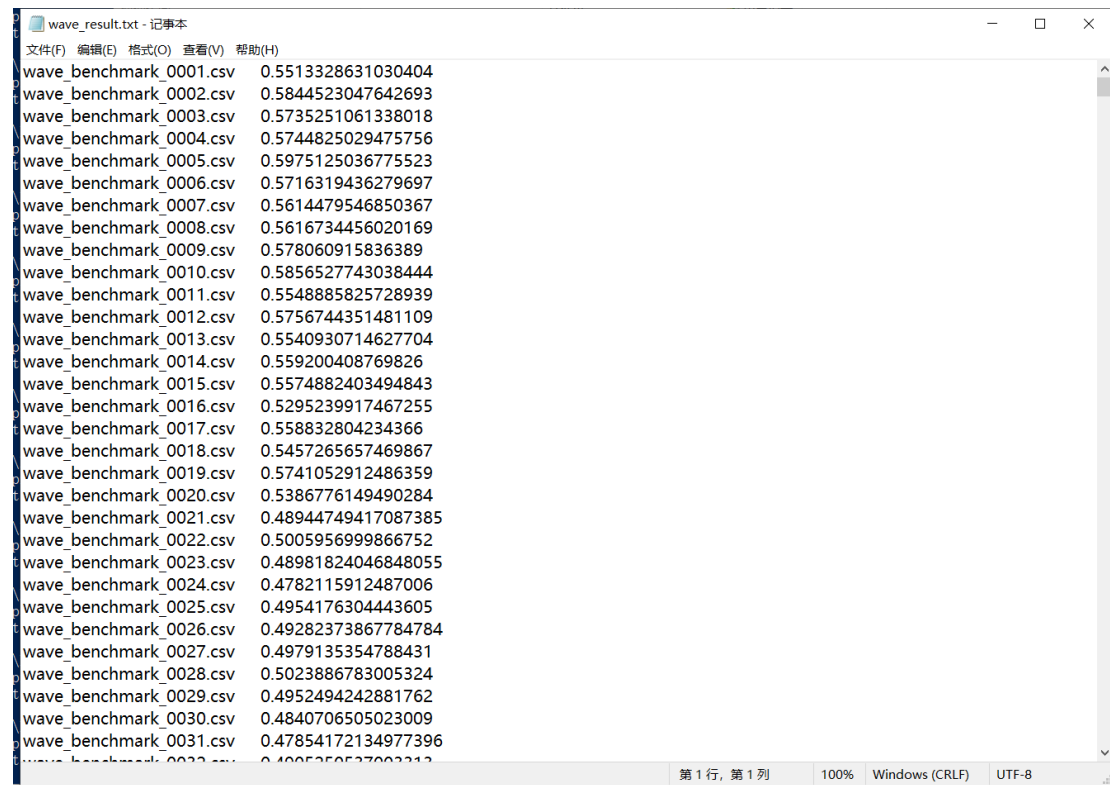
## 二、wave-benchmarks 分析

### 1. wave.py

wave.py 用来对 benchmarks 中每个数据集进行分析，分别使用 KNN、OCSVM 和 HBOS 三种方法进行异常检测，后续代码均以 HBOS 为例。

```
#导入需要的包
import pandas as pd
import numpy as np
import glob,os
from pyod.models.hbos import HBOS
#创建一个文件保存分析结果
result=open('wave_result.txt','a')
#导入数据集路径
path=r'.\wave\benchmarks'
file=os.listdir(path)
i=0
#逐个读取数据并分析
for f in file:
    df=pd.read_csv(path+'\\'+f)
    #提取出数据标签
    label=df['ground.truth']
    #删除对分析无意义的列
    df=df.drop(['ground.truth','point.id','motherset','origin'],axis=1)
    #使用 HBOS 来获取离群点
    clf = HBOS()
    score=clf.fit_predict_score(df, label)
    #将结果写入文件以便后续分析
    inputf=str(f)+'\t'+str(score)+'\n'
    result.write(inputf)
    i+=1
    if i%100==0:
        print(str(i)+'\n')
result.close()
```

结果如下：



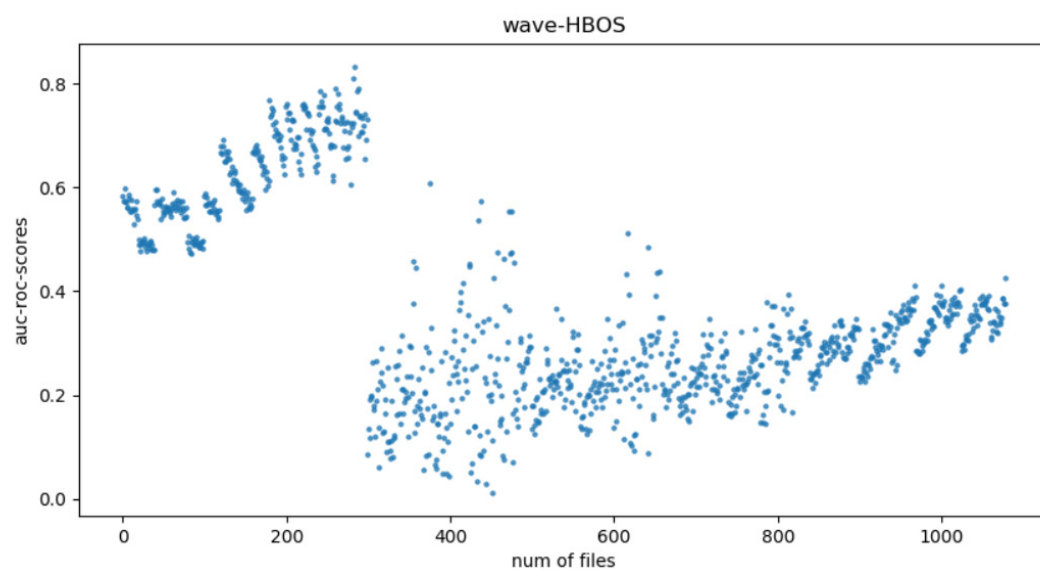
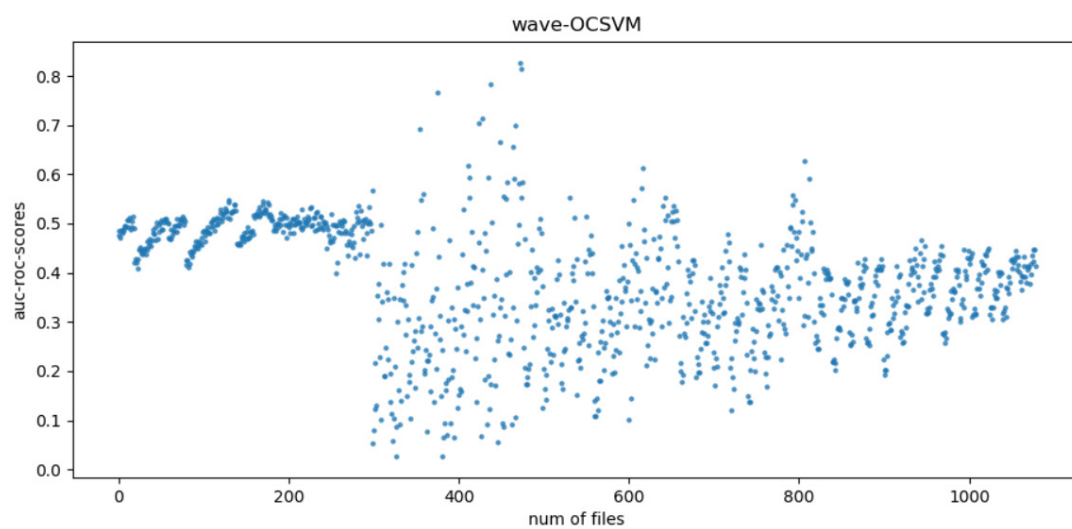
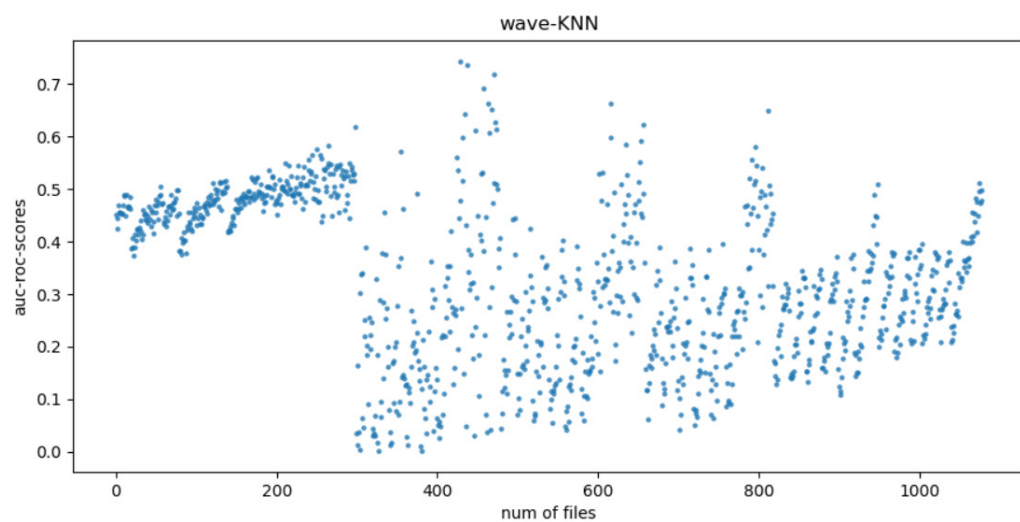
wave_benchmark_0001.csv	0.5513328631030404
wave_benchmark_0002.csv	0.5844523047642693
wave_benchmark_0003.csv	0.5735251061338018
wave_benchmark_0004.csv	0.5744825029475756
wave_benchmark_0005.csv	0.5975125036775523
wave_benchmark_0006.csv	0.5716319436279697
wave_benchmark_0007.csv	0.5614479546850367
wave_benchmark_0008.csv	0.5616734456020169
wave_benchmark_0009.csv	0.578060915836389
wave_benchmark_0010.csv	0.5856527743038444
wave_benchmark_0011.csv	0.5548885825728939
wave_benchmark_0012.csv	0.5756744351481109
wave_benchmark_0013.csv	0.5540930714627704
wave_benchmark_0014.csv	0.559200408769826
wave_benchmark_0015.csv	0.5574882403494843
wave_benchmark_0016.csv	0.5295239917467255
wave_benchmark_0017.csv	0.558832804234366
wave_benchmark_0018.csv	0.5457265657469867
wave_benchmark_0019.csv	0.5741052912486359
wave_benchmark_0020.csv	0.5386776149490284
wave_benchmark_0021.csv	0.48944749417087385
wave_benchmark_0022.csv	0.5005956999866752
wave_benchmark_0023.csv	0.48981824046848055
wave_benchmark_0024.csv	0.4782115912487006
wave_benchmark_0025.csv	0.4954176304443605
wave_benchmark_0026.csv	0.49282373867784784
wave_benchmark_0027.csv	0.4979135354788431
wave_benchmark_0028.csv	0.5023886783005324
wave_benchmark_0029.csv	0.4952494242881762
wave_benchmark_0030.csv	0.4840706505023009
wave_benchmark_0031.csv	0.47854172134977396
wave_benchmark_0032.csv	0.4905350537003313

## 2. wave-result.py

wave-result.py 用来对结果进行可视化

```
#导入需要的包
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#读取 HBOS 分析的结果
df=pd.read_csv('wave_result.txt',delimiter='\t')
df.columns=['num','result']
plt.figure()
#提取文件中结果，并绘制曲线
plt.plot(df['result'])
plt.xlabel("num of files")
plt.ylabel("auc-roc-scores")
plt.title("wave")
plt.show()
```

KNN、OCSVM 和 HBOS 三种方法进行异常检测，可视化结果如下：



### 三、分析

KNN 是采用不同特征值之间的距离方法进行分类的一个分类算法；HBOS 是一种基于频数直方图的无监督异常点检测算法，该方法为每一个样本进行异常评分，评分越高越可能是异常点；OCSVM 即单类支持向量机，该模型将数据样本通过核函数映射到高维特征空间，使其具有更良好的聚集性，在特征空间中求解一个最优超平面实现目标数据与坐标原点的最大分离。这三种方法各具特性，从结果可以看出随着噪声特征加入的增多，离群点分析的结果所获得的分数波动越来越大，通过在 wine 和 wave 两个数据集下比较而言，HBOS 在一些数据集上的效果远优于其他两种方法，同时结果的波动也较小。