二〇一九年一月二日星期三

上午10时28分

元旦节后开工。

早上提交了工程师转助研申请。

开始元旦假期中自己的实验假设，即基于静态的26JS与33维度的ATF的分析，主要分析对象为CERT5.2用户。

基于静态分析假设，更容易调整参数得到最好的结果。

先基于原有的26JS维度与新的33维度的ATF进行研究吧！

下午3时36分

分别按照传统的JS与ATF进行实验，首先，进行KMeans分析，可以得到：

26JS: [-1]用户可以分成2个群簇，分别包含

0 1564

1 197 （显然应该是我们特别关注的部分）

而使用33维度的ATF实验，有 针对ATF\_lst的

最佳自动K值为[2:10] 2 : 0.36697501148725215

0 197 个

1 1564 个

下面继续使用后续的ATF聚类结果进行实验，并且默认采用1564个作为训练集，剩余197个[-1]+[+1]作为测试集输出结果；

计算召回率与FPR

197/1564=0.12...

下面开始记录针对ATF特征的预测结果：

自动PCA（）

[**0.05**, 0.9, 0.47005241751310445, 0.5362318840579711, **0.12556278139069535**, 0.6651270207852193, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=15**, shrinking=True, tol=0.01,

verbose=False), **0.5172413793103449, 0.6, 0.4**]

[**0.01**, 1.0, 0.5362318840579711, 0.5362318840579711, **0.128064032016008**, 0.6766743648960739, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.01, **random\_state=15**, shrinking=True, tol=0.01,

verbose=False), **0.5172413793103449, 0.6, 0.4**]

[0.01, 1.0, 0.5362318840579711, 0.5362318840579711, **0.128064032016008**, 0.6766743648960739, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.01, **random\_state=0**, shrinking=True, tol=0.01,

verbose=False), **0.5172413793103449, 0.6, 0.4**]

[0.05, 1.0, 0.5362318840579711, 0.5362318840579711, 0.12556278139069535, 0.6651270207852193, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=5**, shrinking=True, tol=0.01,

verbose=False), **0.5172413793103449, 0.6, 0.4**]

[0.05, 1.0, 0.5362318840579711, 0.5362318840579711, 0.12556278139069535, 0.6651270207852193, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=10**, shrinking=True, tol=0.01,

verbose=False), **0.5172413793103449, 0.6, 0.4**]

[0.05, 1.0, 0.5362318840579711, 0.5362318840579711, 0.12556278139069535, 0.6651270207852193, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=100**, shrinking=True, tol=0.01,

verbose=False), **0.5172413793103449, 0.6, 0.4**]

下面开始调整PCA的维度

PCA=3

[0.05, 1.0, 0.6811594202898551, 0.6811594202898551, 0.13356678339169584, 0.7251732101616628, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=100**, shrinking=True, tol=0.01,

verbose=False), **0.6551724137931034, 0.6, 1.0**]

[0.05, 1.0, 0.6811594202898551, 0.6811594202898551, 0.13356678339169584, 0.7251732101616628, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=0**, shrinking=True, tol=0.01,

verbose=False), **0.6551724137931034, 0.6, 1.0**]

PCA=1

*[0.05, 1.0, 0.8695652173913043, 0.8695652173913043, 0.17588204318062137, 0.19709854927463732, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',*

*max\_iter=-1, nu=0.05, random\_state=10, shrinking=True, tol=0.01,*

*verbose=False), 0.7931034482758621, 0.9, 1.0]*

[0.05, 1.0, 0.8695652173913043, 0.8695652173913043, 0.16708354177088544, 0.9099307159353349, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=10**, shrinking=True, tol=0.01,

verbose=False), **0.7931034482758621, 0.9, 1.0**]

[0.15, 0.9, 0.6888400722100181, **0.782608695652174**, 0.15507753876938468, 0.8406466512702079, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.15, random\_state=10, shrinking=True, tol=0.01,

verbose=False), **0.7586206896551724, 0.7666666666666667, 0.9**]

PCA=2

[0.05, 1.0, 0.7971014492753623, 0.7971014492753623, 0.14357178589294647, 0.789838337182448, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto', kernel='rbf',

max\_iter=-1, nu=0.05, **random\_state=10**, shrinking=True, tol=0.01,

verbose=False), **0.7931034482758621, 0.7666666666666667, 0.9**]

[0.05, 1.0, 0.8260869565217391, 0.8260869565217391, 0.1535767883941971, 0.8406466512702079, OneClassSVM(cache\_size=200, coef0=0.0, degree=3, gamma='auto',

kernel='linear', max\_iter=-1, nu=0.05, random\_state=10,

shrinking=True, tol=0.01, verbose=False), 0.7931034482758621, 0.8, 1.0]

今天下午最大的结果就是，通过PCA=1的KMeans+OCSVM 可以得到了recall基本满意，FPR=0.167的结果