BABY使用指南

> written by SPICY_NOODLES 2024.10.2
>
> 用于世界上最帅的its小组获得一等奖使用!
>
> 机密级文件：命名为何如此抽象？不希望倒下后我们的py可以被敌人直接使用

文件简要介绍

CSV文件介绍

请注意：文件名中的**50/500/2000**均为train_data原始数据集中的末尾***flight_id***序号，并非数据集的准确大小。如*train_data_15.csv*实际只有**10**条数据，*train_data_50.csv*实际只有**36**条数据，其生成的*washed_data_50.csv*实际只有**36**条数据

csv文件类型	用途	例如
train_data	包含不同大小的训练集	train_data_50.csv
washed_data	清洗后的训练集	washed_data_50_1.csv
dtw	dtw距离矩阵	dtw_50.csv
ans	只包含***fight_id***和***label***	ans_50_eps10000_ms2.csv

> **ans为什么命名这么长?**
>
> 要包含数据大小：50
>
> 体现DBSCAN聚类的方式
>
> eps（可归为一类的航班之间最大距离）：10000
>
> ms（min_samples 最少几个航班聚为一类）：2

py文件简要介绍

py文件	用途	耗时
Baby_creater1	直接由**train_data**生成**ans**	巨tmjb长
baby_data	由**train_data**生成**washed_data**保存	很长
baby_data_to_dtw	由**washed_data**生成**dtw**保存	有bug
baby_to_dtw	直接由**train_data**生成**dtw**保存	非常长
baby_dtw_to_DBSCAN	**dtw**生成聚类结果**ans**	短

> **使用说明**
>
> Baby_creater1为一步出结果的py，但耗时太太太太太太长不宜聚类调试，可根据其他小baby来修改这个大的，该py应作为最终测验提交版本

> 建议直接用baby_to_dtw.py生成一个需要大小的dtw（会花很多时间，但是一劳永逸，以后再也不需要数据清洗和dtw了哈哈哈哈哈哈哈哈哈
>
> 有了这个dtw.csv以后，直接用*baby_dtw_to_DBSCAN.py*跑几秒就出结果存在ans里了。然后就可以极其高效地开始DBSCAN的调试和优化了

使用说明

baby_to_dtw.py

1.修改读取的数据集（在第63行）

```
# 需要修改begin

    df = pd.read_csv('train_data_50.csv')

# 需要修改end
```

2.修改存储的dtw距离矩阵名称（在第110行）

```
#存储距离矩阵
np.savetxt("dtw_50.csv", dtw_matrix, delimiter=',')
```

baby_dtw_to_DBSCAN.py

1.修改读取的距离矩阵（在第8行）

```
# 读取已有的距离矩阵
dtw_matrix = np.loadtxt(open("dtw_50.csv", "rb"), delimiter=",", skiprows=0)
```

2.修改DBSCAN参数：eps和min_samples（在第21行）

```
# 使用DBSCAN进行聚类
'''这里eps是聚类距离半径最大值，相当于圆的半径，min_samples就是一个类内最小有几个样本'''
dbscan = DBSCAN(metric='precomputed', eps=10000, min_samples=2)
labels = dbscan.fit_predict(dtw_matrix)
```

3.修改存储的ans聚类结果名称 包含相应的eps和min_sample (在第28行)

```
# 存储ans标注数据大小和eps等聚类方式
df.to_csv('ans_50_eps10000_ms2.csv')
```

项目目前进度

事项	情况	优化
数据清洗	已完成	暂无需
DTW	已完成	暂无需
DBSCAN	已完成	**最需优化**
15聚类结果	已完成	
50聚类结果	已完成	
2000聚类结果		
全数据聚类结果		
绘图		