概要

mooc数据集已经清洗完毕，整理为时间序列集合。
读到一篇Google编写的教程，按其给出的评测标准目前的数据集更适合结构较为简单的网络如MLP/SVM\GBDT等，正在将数据集整理成能用的格式。

数据清洗情况

每条样本都是每个注册号下被记录的一系列有序行为
样本数：67700
样本长度分布情况：
直方图请看图片
[0,2] : 1910 无效数据
[0,10] : 21379 无效数据
[10,50] : 19720
[50,100] : 7052
[100,2000] : 17370 可用数据
[2000,5000] : 322
数据分布很不均衡，但可用样本数也不少。

可用数据的选择要根据任务和模型而定：
最好情况下我想实现序列到序列的预测（预测详细行为）
其次是序列到时间段的预测（预测时间段内行为的类型、频次）AAAI那篇论文的辍学结论是基于十天内没有操作就是为辍学，我至少先要实现类似的效果。

1)若使用完整序列输入模型，tensorflow要求输入样本等长，所以必须设定阈值对样本做截断或填充无意义字段，若阈值设定过高会导致大部分样本中大量无意义字段。
虽然冗余不可避免，但是无意义字段对模型的影响我还不能给出确定的答案。
原因：
我看tensorflow的文档时读到，对于小于长度阈值的序列，可以设置mask参数，将被填充的位置告知要读取该序列的网络，不对无意义字段进行处理，这部分我还没看完。

2)若不使用完整序列，而是使用 n-gram将序列转为向量，问题就不大，毕竟总共只有20种行为，即使是长度在[2000,5000]的序列都不会使得编码后的向量过于庞大。

模型选择

简述Google给出的评测指标：
c = 样本数量/每条样本的词数量
c < 1500 使用n-gram将样本转换为向量 配以简单的模型
c > 1500 将样本描述为完整序列使用预训练模型

link https://developers.google.com/machine-learning/guides/text-classification/step-2-5

经计算，本数据集选用全部样本计算得 c = 428
有效样本计算得 c = 44 ，均小于阈值1500

我的想法是，用户在浏览mooc时被记录的行为构成的序列，可以被理解为一段文本，20种不同的行为就是20个符号，看作是一种新的语言，符号的不同顺序组合表达不同的含义。

Google的这种评价方式是基于对多个数据集的实验得出，c值可以某种程度上描述该种语言的复杂性与样本的复杂程度，可以一试，目前正在按照其路径对有效数据集使用n-gram分词并使用tf-idf算法计算每个样本的序列。

使用tf-idf，而不使用onehot或词频计数的方式的原因是onehot与词频计数不能对类似英文中'a''the'这样的词进行正确的描绘，并没有什么意义但是却有很高的词频，在本数据集中即是'play_vedio'，反而是数据集中较少出现的重要词不容易被发现，如评论类的行为。

新知识.tf-idf算法:
    tf 词频.词在单个样本种出现的频率
    idf 反向文件频率.文件总数/出现某次的文件数量
    算法是基于这样的假设:
        一个词很少出现->不重要->信息少
        一个词经常出现->重要->信息多
        一个词过于频繁的出现->不重要->信息少
    是一种简单描述词语重要程度的算法