例题：隐式马尔可夫模型识别XSS攻击

隐马尔可夫模型与隐式马尔可夫链

隐马尔可夫模型HMM(全称Hidden Markov Model)是一种经典的机器学习模型，特别是在序列数据建模方面。它基于马尔可夫过程，但增加了一个隐藏层，即隐式马尔可夫链，使得模型能够捕捉隐式状态之间的转移概率。

HMM与三个经典问题

在之前的学习中我们了解到了隐式马尔可夫过程的三个经典问题，即概率计算问题，参数估计问题，解码问题，接下来我们通过这三个经典问题来解释HMM的学习和预测过程：

首先我们需要通过训练样本数据来确定隐马尔可夫的三要素（初始状态概率向量，一步转移概率矩阵和发射矩阵），那么根据之前的学习如果我们拥有足够多的正常请求的观察序列，理论上我们可以通过计算参数估计问题的方法还原出最符合该数据集的隐式马尔可夫模型三要素，这个过程也就是HMM的学习过程。

在学习出最符合当前数据的隐马尔可夫模型后，如果我们得到了一个新的观察序列，我们可以通过学习到的HMM三要素，通过解决概率计算问题来得到这样一个观察序列出现的概率，这个过程就是HMM的预测过程。

XSS攻击的HMM建模

我们想要利用隐式马尔可夫模型识别XSS攻击，本质上就是想要用前面所说的方法利用一个经过正常样本训练的HMM对于给定的http请求进行预测概率，如果概率过低则说明在默认请求正常的情况下，这样一个特定的http请求出现的概率过低，反过来讲将这样一个请求判定为正常的风险很高，那么这样一个数据就应该被过滤掉。

因此我们的主要任务就是将http请求转换为HMM可学习的数据模式，首先我们对于获得的GET请求进行URL解码获得其中信息。这里我们的URL解码可以直接使用python中的一些库函数如urlparse，unquote进行直接处理。

但由于信息中各种字符取值过多直接进行训练会导致模型复杂度过高，出现训练速度慢等问题，所以我们将其进行泛化调整以简化模型，这里泛化规则如下：

> [a-z, A-z]泛化为 A
>
> [0-9]泛化为 N
>
> [\ - _] 泛化为 C
>
> 其他字符泛化为 T

> 如样本admin123将会被泛化为AAAAANNN

经泛化后状态序列的状态数得到限制，可用于HMM训练。

由于python中的hmmlearn包中内置HMM模型的学习和预测过程，所以我们可以直接调用hmmlearn完成实验。

这里给出该实验可能用到的各种函数

```
model = hmm.GaussianHMM(n_components=N, covariance_type="full", n_iter=100)
```

高斯分布下的HMM模型建立
n_components: 这是一个整数，表示模型中的隐藏状态数，即模型将有多少个状态。在隐马尔可夫模型中，这些状态是未观测的，但可以通过观测到的数据序列来推断。
covariance_type: 这是一个字符串，用于指定高斯分布的协方差矩阵的类型。
n_iter: 这是一个整数，表示在训练模型时的最大迭代次数。增加迭代次数可能会提高模型的性能，但也会增加计算时间。

```
model.fit(X,X_lens) # HMM类的内置函数，用于训练模型，其中X为待训练向量，X_lens为样本长度
model.score(vers)   # HMM类的内置函数，用于对测试向量vers进行评分，可通过设置阈值进行异常检测
```

我们利用fit()函数一次次使用正常样本优化模型参数，使用score()函数对于每一个给定的请求进行概率评分，如果概率低于一定值则将其过滤。

经训练，过滤成功率达到80%

实验结果图片