基于Markov模型的口令猜测算法代码实现

一般情况下，用户构造口令的顺序是从前向后依次进行的。根据这一特点，Narayanan等人在2005年首次将Markov链技术引入口令猜测中来。与PCFG算法不同，基于Markov模型的口令猜测算法对整个口令进行训练，通过从左到右的字符之间的联系来计算口令的概率。该算法也分为训练和猜测集生成两个阶段，以下介绍实现细节。（针对此前代码存在的问题，本次代码进行了相应的修改。对于start symbol问题，已经充分理解并实现；对于阈值问题，优化了阈值的分配方式。综合来看，实验效果得到了很大的提升）

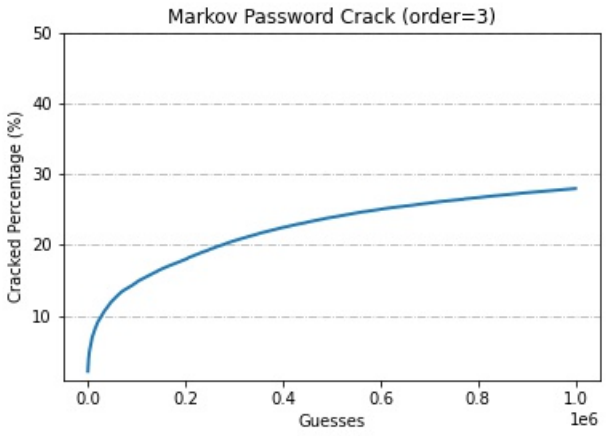
实现细节

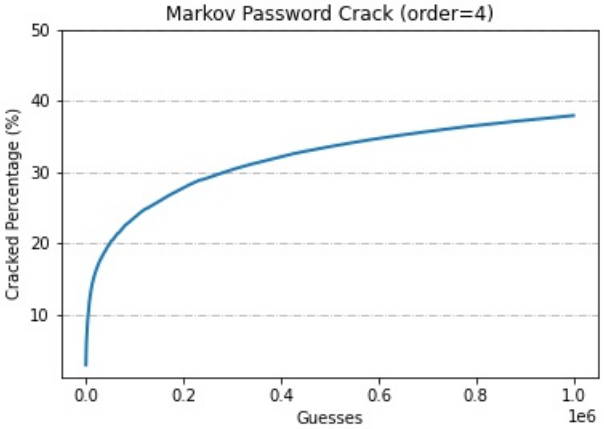
该算法的实现主要包括三个部分：口令集预处理、口令集训练、口令猜测，使用的编程语言为python3.7，具体细节如下所示：

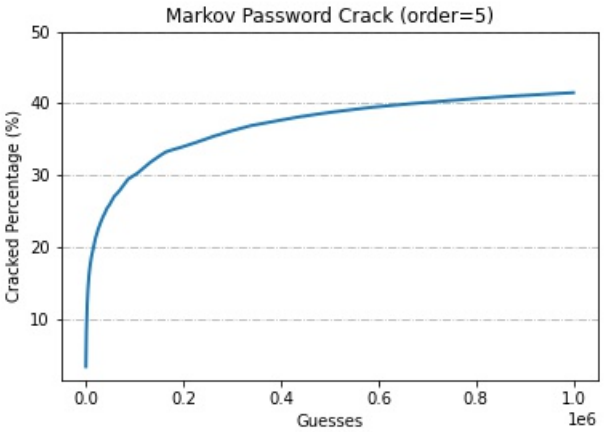
- 口令集预处理：实验中所使用的口令集为Rockyou，去掉了包含非ASCLL或者空格的口令，剩余口令总数为32460357。由于口令总数很大，这里我没有使用全部的口令进行训练和测试。我从数据集中随机选择了2000000条口令，然后将这些口令拆分为训练集和测试集（各占50%）。代码中设置了number参数，可以选择使用数据的量。除此之外，在随机选择数据时，设置了随机种子seed，方便实验结果的复现。
- 口令集训练：口令集训练的目的是统计出各个字串在训练集中出现的频数。统计频数时，对于口令开头字母出现的频率单独进行统计，其余字串的频数均存放在一个字典中。我使用了End-Symbol正规化技术，频数统计时也会对口令结束标志符号进行统计。频数统计完成之后，利用Laplace平滑技术来计算概率，然后对每个字串后面出现的字母依据概率值大小进行排序。
- 口令集猜测：这里使用优先队列的思想来对猜测口令进行存储和遍历，如果当前队列前端的字符串最后一个字符为口令结束标识符，则将其输出进行口令猜解，否则根据其字串在概率表中的统计情况，在该字符串后继续添加字符并计算概率，然后插入队列。为了减少队列对内存的损耗，再将每一个字符串插入队列之前，要对其概率值进行判断。只有当其概率值大于预设阈值时，才准许插入队列。

实验结果

按照上述实验设置进行了实验，由于实验机器性能有限，这里使用的测试集口令总数为1000000，猜测次数为1000000，order=3,4,5，下图展示了口令破解速度的变化趋势。order=3时，猜测出的口令数目为279517；order=4时，猜测出的口令数目为378980；order=5时，猜测出的口令数目为414806。




Markov Password Crack (order=4)
Cracked Percentage (%)
50
40
30
20
10
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Guesses


Markov Password Crack (order=5)
Cracked Percentage (%)
50
40
30
20
10
0.0
0.2
0.4
0.6
0.8
1.0
1e6
Guesses