PCFG口令破解算法代码实现

PCFG是一种完全自动的、建立在严密的概率上下文无关法基础之上的漫步口令猜测算法，该算法的核心思想是将每一条口令看作是由字母段L、数字段D、特殊字符段S根据一定的模式互相组合而成的。我们可以通过对大量的口令数据进行分析，统计出这些口令可能的组合模式，从而利用该模式生成更多的口令，进行口令猜测。

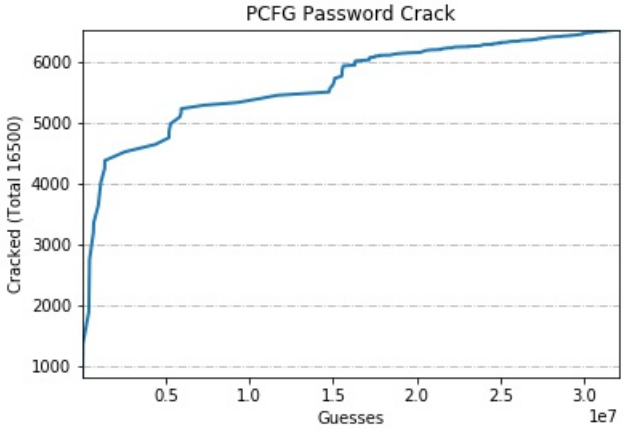
实现细节

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MySpace，去掉了包含非ASCLL或者空格的口令，剩余口令总数为41251。然后将这些口令随机拆分为训练集和测试集，分别存放在trainword.txt和testword.txt文件中。代码中设置了参数eps可以对训练集和测试集的分配比例进行调节，本文将eps设置为0.4，即测试集占口令总数的40%（16500）。除此之外，在随机拆分训练集和测试集时，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口令模式频率和字符组件频率。对于字符组件频率的统计，论文中提到的方法只统计了数字段和D和特殊S。在实现过程中，我也统计了字母段L的频率，同时字母段L将作为字典参与口令猜测过程。（根据测试集生成的口令模式一共有1134种）
- **口令猜测：**口令猜测的过程类似于树的遍历，这里采用类似于深度优先遍历的方法。分别对所有的口令模式频率表和字符组件频率表由大到小进行排序，从频率最高的口令模式开始进行口令猜测，根据字符组件频率表依次生成该模式下所有可能的口令，然后在进行下一个口令模式的猜测。对于每个生成的口令，需要计算其可能出现的概率，如果该概率值大于预设的阈值（比如0.000000001），则可以将其输出并与测试集中的口令进行比对。反之，则将该口令丢弃。实验中所有的猜测口令都存放在guess文件夹下对应的口令模式文件中。

实验结果

按照上述实验设置进行了实验。在代码刚开始运行阶段，正确猜解口令的速度很快。两分钟的时间内大概能够正确猜解6000多个口令，之后正确猜解的速度逐渐降低，下图展示了口令破解速度的变化趋势。



由于生成口令时我们设置了阈值，只有超过该阈值的口令才会进行匹配，所以真正参与口令猜测的口令数量要远远小于实际生成的口令数量，上图横坐标展示的就是参与口令猜测的口令数量。这里展示的结果是利用口令模式表中前400种左右的模式生成的口令，正确猜解的口令数量为6529（总共16500）。