HW3 选择二

18340013 陈琮昊

理论部分：

> 理论部分：习题 12.1 假设一个 DTMC 对一个离散并且有 N 个可能取值（状态）的随机变量 X 的演化进行建模. 证明我们需要N^2-1 个数来完全描述这个 DTMC.

证明：

对于一个离散且有N种取值的随机变量，用马尔科夫链描述该随机过程时，首先有一个转移概率矩阵，其规模为$N\times N$，由于转移矩阵具有行和为1的性质，因此每一行只需$N-1$个值即可；而该矩阵一共有N行，因此需要$N(N-1)$个数；又因为还要定义初始状态下分别取这N个值的概率，这N个概率值同样满足和为1，因此需要$N-1$个数，因此总共需要
$N(N-1)+(N-1)=(N+1)(N-1)=N^2-1$个数来描述该马尔科夫链。

编程部分：

> 编程部分：
>
> 利用 HMM 进行中文分词，数据集可自行准备（中文文字不得低于 1000 字）

`HMM`的参数为一个五元组，在中文分词里，这五个参数的具体含义如下：

状态值集合为(B,M,E,S)，观察值集合为所有汉字，这两个参数与三个矩阵通过`Viterbi`算法连接起来：观察值集合为`Viterbi`算法的输入，状态值集合为`Viterbi`算法的输出，在输入和输出之间，三个矩阵则是`Viterbi`算法执行需要借助的参数。

相关代码已打包至文件夹内，可以运行、测试。

运行环境：`Windows+Python3.6`

其中`hmm.py`为主程序，实现了用`HMM`进行中文分词；`r_hmm_data.pkl`则是将词频等信息保存下来以便后续使用；`trainCorpus.txt_utf8`为训练集；`testset.txt`为测试集；`log.txt`为测试集下的分词结果。（编码格式均为`utf-8`）

代码部分简介如下：

```
class HMM(object):
    def __init__(self, load=False):...

    # 将相关信息写入临时文件，方便后续使用，使用时读该文件即可
    def savepara(self):...

    def loadpara(self):...

    # 初始化参数
    def initpara(self, trained=False):...

    def labelmark(self, text):...

    # 从语料中获取词频
    def traincorpus(self, file_path, trained=False):...

    # 词频转化为概率
    def calculate(self):...

    def viterbi(self, text):...

    def cut(self, text, best_path):...

    def use_cut(self, text):...
```

`__init__`~`initpara`都是一些初始化的内容；

`labelmark`则是给句子中每个词打上标签，如下：

```
def labelmark(self, text):
    length = len(text)
    if length == 1:
        return ['S']
    else:
        return ['B'] + ['M'] * (length - 2) + ['E']
```

`traincorpus`则是训练过程：取出来字、词，并打上标签，并统计词频；

`calculate`则是计算状态转移概率、发射概率和初始分布概率；

`viterbi`则是`viterbi`算法实现，课上相关内容不再介绍；

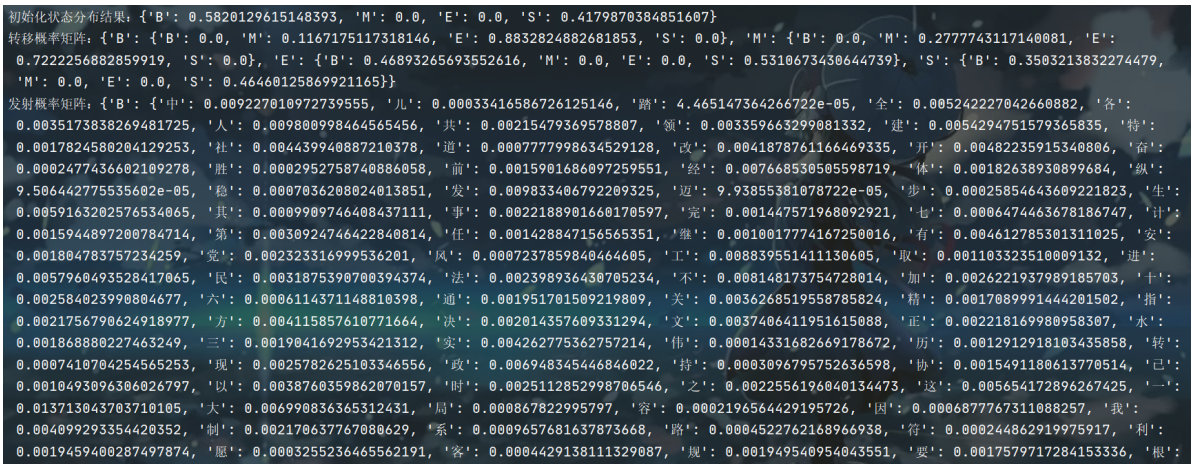
`cut`和`use_cut`则是进行分词（使用`viterbi`算法）。

`main`函数很好理解，就是先用训练集训练，然后用训练好的结果进行测试，将得到的分词结果写入`log.txt`：

```python
if __name__=='__main__':
    f = open('testset.txt', 'r', encoding='utf-8')
    lines = f.readlines()              # 读取测试集数据
    testcase = list()
    for line in lines:
        line = line.strip('\n')
        testcase.append(line)
    print(len(testcase))
    log = open('log.txt', mode='a', encoding='utf-8')
    for i in range(len(testcase)):
        training = HMM(load=True)
        print("第{}行分词结果已写入文件".format(i))
        print(training.use_cut(testcase[i]), file=log)
    # training = HMM(load=True)
    # initmatrix, transprobmatrix, observematrix = training.calculate()
    # print("初始化状态分布结果：{}".format(initmatrix))
    # print("转移概率矩阵：{}".format(transprobmatrix))
    # print("发射概率矩阵：{}".format(observematrix))
```

测试集内有2000行语句，每测试完一行语句会有提示性输出，整个测试过程大约用时1 min。中文分词结果详见文件夹内的`log.txt`。

在代码文件内main函数被注释掉的最后几行代码为打印三个概率矩阵，可以去掉注释并运行代码得到三个概率矩阵的结果，这里只截取一部分：



代码文件内有注释可以帮助阅读整个代码。