# 广州理工学院实验报告

学院：广州理工学院 专业：计算机科学与技术 班级：21计科21班

姓名： 张亚彬 学号：20210411432106组 实验时间：2024年5月28日

指导教师签字： 成绩：

|  |
| --- |
| 实验项目名称：  实验5：隐马尔可夫模型应用实验   * 实验目的和要求：   （1）掌握隐马尔可夫模型的应用方法。  （2）掌握序列标注问题的基本处理方法。 |
| * 主要仪器设备：  1. 个人计算机：每位学生需要使用一台装有anocanda的个人计算机，以进行实验； 2. 操作系统：计算机上应安装支持anocanda的操作系统，如Windows、macOS等；   网络连接（如需要下载python相关库）等。 |
| * 实验内容和原理：   **实验内容**  （1）本次实验的具体内容是模拟全拼输入法。根据提供的训练语料（以小说《天龙八部》生成）包括许多以句子为单位的中文序列和对应的全拼拼音序列组成的训练样本，训练样本示例如下：  （2）对训练样本进行统计，得到HMM模型要求的初始概率、转移概率和观测概率，实例化hmmlearn.hmm模型，对测试样本进行预测。  （3）在提供的hmm\_model.ipynb文件中填写代码，完成指定任务。  **实验原理**  状态序列：汉字序列；观测序列：拼音序列。  初始概率：描述每个状态作为序列起始的概率。转移概率：在隐状态序列中从一个状态转移到另一个状态的概率。观测概率：在给定状态下观测值的概率。根据统计训练样本中的初始概率、转移概率和观测概率构建HMM模型，对于测试样本中的拼音序列，使用训练好的HMM模型进行预测，得到最可能的汉字序列。 |
| * 操作方法和实验步骤（绘图）：   **# 打开文件"traindata.txt"，使用'utf-8'编码**  **file = open("traindata.txt", encoding='utf-8')**  **# 初始化两个字典，一个用于存储中文字符及其出现次数，一个用于存储拼音及其出现次数**  **char\_dict = {} # 存储中文字符及出现次数的字典**  **pinyin\_dict = {} # 存储拼音及出现次数的字典**  **# 读取文件的每一行**  **lines = file.readlines()**  **# 使用一个布尔变量tag来切换处理中文字符和拼音的字典**  **tag =True**  **# 遍历文件的每一行**  **for line in lines:**  **# 如果tag为True，处理当前行的中文字符，将temp\_dict设置为char\_dict**  **if tag:**  **temp\_dict = char\_dict**  **tag = False**  **# 否则，处理当前行的拼音，将temp\_dict设置为pinyin\_dict**  **else:**  **temp\_dict = pinyin\_dict**  **tag = True**  **# 将当前行分割成单词（以空格分隔）**  **for cent in line.split():**  **# 如果单词已经在temp\_dict中，将其计数加1**  **if cent in temp\_dict:**  **temp\_dict[cent] += 1**  **# 否则，将单词添加到字典中，计数初始化为1**  **else:**  **temp\_dict[cent] = 1**  **# 定义初始概率矩阵pi、转移概率矩阵A和观测概率矩阵B**  **# 与教材上的处理方式不同，采用全1的平滑处理方式，该处理是符合hmmlearn.hmm的要求。**  **# 但是这种处理方式简单，仅适合学生入门，效果有时不好。**  **pi = np.ones(len(char\_id))**  **A = np.ones((len(char\_id), len(char\_id)))**  **B = np.ones((len(char\_id), len(pinyin\_id)))**  **# 统计3个概率矩阵**  **n\_lines = len(lines)**  **for i in range(0, n\_lines, 2):  # 每两行数据是一组（中文字符和拼音）**  **chars = lines[i].split()  # 获取中文字符序列，并使用空格作为分隔符进行分割**  **pinyins = lines[i+1].split() # 获取拼音序列，并使用空格作为分隔符进行分割**  **# 更新初始概率矩阵pi**  **for char in chars:**  **if char in char\_id:**  **pi[char\_id.index(char)] += 1**  **# 更新观测概率矩阵B**  **for j in range(len(chars)):**  **if chars[j] in char\_id and pinyins[j] in pinyin\_id:**  **B[char\_id.index(chars[j]), pinyin\_id.index(pinyins[j])] += 1**  **# 更新转移概率矩阵A**  **for j in range(len(chars) - 1):**  **if chars[j] in char\_id and chars[j+1] in char\_id:**  **A[char\_id.index(chars[j]), char\_id.index(chars[j+1])] += 1**  **# 归一化概率矩阵**  **pi /= np.sum(pi)**  **A /= np.sum(A, axis=1, keepdims=True)  # 按列归一化**  **B /= np.sum(B, axis=1, keepdims=True)  # 按列归一化** |
| * 实验结果与分析   实验结果表明，使用HMM模型对《天龙八部》训练数据进行学习后，能有效预测测试样本的汉字序列。给定一个拼音序列，模型成功地输出了对应的汉字，如"段誉问你是乔峰还是萧峰"。这验证了模型在捕捉汉字与拼音关系上的能力。然而，实际应用中，模型可能无法处理未见过的拼音组合或罕见汉字，导致预测准确率受限。未来工作可优化模型参数，提高对未知数据的泛化能力。 |

批阅评语：