****

**人工智能上机实验报告**

题目： **隐式马尔科夫模型实验**

**Hidden Markov Models**

**学院名称 智能与计算学部**

**专 业 软件工程**

**学生姓名 郎文翀**

**学 号 3019244247**

**年 级 2019级**

**班 级 5班**

**时 间 2021.11.15**

## Experimental Requirements

# 阅读题干，简述实验要求

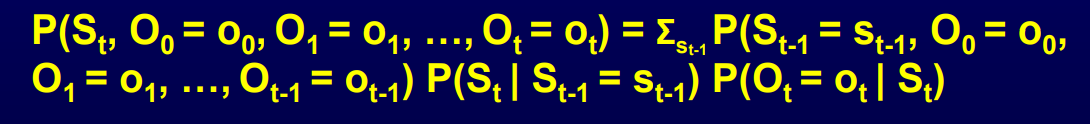
# Read the description, and conclude the requirements of the experiment.

# Introduce the tasks of your experiment

#阅读题干以后略有疑惑，但是将题意抽象出来以后发现实际上就是要实现用矩

#阵计算下面的隐式马尔科夫模型的公式：

After reading the stem of the question, I was slightly confused, but after abstracting the meaning of the question, I found that it is actually to implement the following formula for calculating the implicit Markov model using matrices.



## Configuration

# 介绍你的环境和配置

# The configuration of your experiment

# The environment required for your experiment

#环境只需要保证配置好Python环境即可，同时我使用了vscode编辑器进行代码编辑，安

#装了一系列代码美化插件和python插件。

The environment just needs to ensure that the Python environment is configured, while I used the vscode editor for code editing and installed a series of code beautification plugins and python plugins.

## Implementation

# 描述实验内容

# 详细介绍实验过程的基本步骤，有必要的情况下可以通过图片、伪代码等形式介绍

# 不建议直接粘贴所有代码，必要的情况下可以粘贴部分代码段，请保证格式正确，且不影响阅读体验。

# Describe the implementation of the experiment

# Elaborate each step of your experiment, you can use figures and pseudocode to fully explain your work.

# Please DO NOT copy the entire code to this report, you can copy code snippets (some part of your code) if necessary. Please make sure the pattern of the code is clear and up to coding standard. Please make sure your code does not influence the reading.

#首先我们要知道预测公式是

#P(S\_t+1|O1:t)=Σ\_t[P(S\_t=s\_t|O1:t)\*P(S\_t+1|S\_t=s\_t)\*P(O\_t=o\_t|S\_t)]根据题干现#在已经给出在O1:t基础上了S\_t取不同的状态值k的概率向量了即#P(S\_t=k|O1:t)(k取任意状态值)

#todo提示告诉我们现在要做的就是当前状态矩阵\*状态转移矩阵刚好就是预测

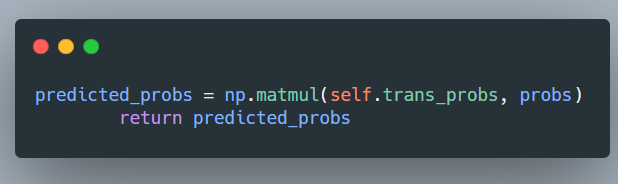
#公式=右边的前两项

#那么我们只需要用P(S\_t+1|S\_t)的矩阵去和给出的probs向量相乘即得到了在#O1:t基础上S\_t+1取不同状态值k的概率了如下

First we need to know that the prediction formula is P(S\_t+1|O1:t)=Σ\_t[P(S\_t=s\_t|O1:t)\*P(S\_t+1|S\_t=s\_t)\*P(O\_t=o\_t|S\_t)] According to the question stem has now given the probability vector of S\_t taking different state values k based on O1:t i.e. P(S\_t=k|O1:t)(k takes any state value)

todo hint tells us that all we have to do now is the current state matrix \* state transfer matrix is just the first two terms of the prediction formula = right

Then we just need to use the matrix of P(S\_t+1|S\_t) to multiply with the given probs vector to get the probability of S\_t+1 taking different state values k on the basis of O1:t as follows



#根据提示，这里我们要更新probs向量

#实际上这里就是再求预测公式的前两项乘积后的中间结果和最后一项的乘积

#首先我们知道O[b]是我们在状态t时从不同位置获取到的观测值向量，他存储

#了此时当前状态不同位置的观测值

#然后我们要根据提示完成更新，做法就是用当前的probs向量去乘S\_t即当前状#态基础上得到的不同的观察值O^b\_t=o[b]的概率向量

#同时t时刻得到不同的观测值b概率可以根据题干self.obs probs[b,o[b],s]获得，#因此代码如下

#注意，初始时要先把所有的概率设置为1,

According to the hint, here we have to update the probs vector

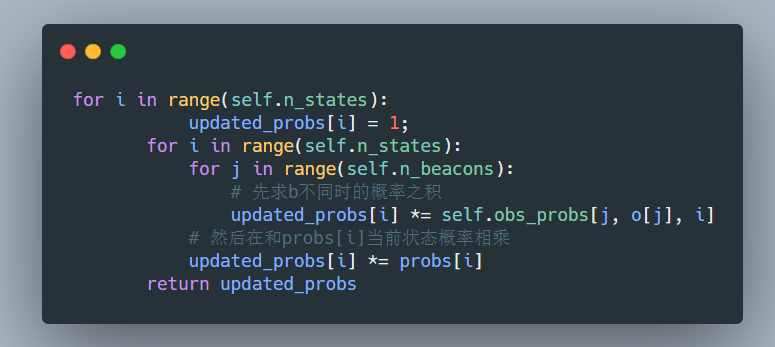
In fact, this is the product of the first two terms of the prediction formula and the last term of the intermediate result

First we know that O[b] is a vector of observations obtained from different locations at state t, which stores the observations at different locations in the current state

Then we have to complete the update according to the prompt, the practice is to use the current probs vector to multiply S\_t that is, the current state based on the probability vector of different observations obtained O^b\_t = o[b

Also the probability of getting different observations b at moment t can be obtained according to the question self.obs probs[b,o[b],s], so the code is as follows

Note that initially, all probabilities should be set to 1 first,



#现在我们已经完成了预测公式

#P(S\_t+1|O1:t)=Σ\_t[P(S\_t=s\_t|O1:t)\*P(S\_t+1|S\_t=s\_t)\*P(O\_t=o\_t|S\_t)]核心的乘积#运算步骤

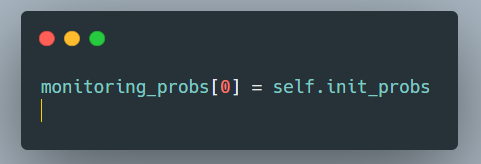
#那么接下来我们需要将两个核心的运算步骤组合起来完成最终的预测公式

#首先我们初始化0状态

Now we have completed the core product operation of the prediction formula P(S\_t+1|O1:t)=Σ\_t[P(S\_t=s\_t|O1:t)\*P(S\_t+1|S\_t=s\_t)\*P(O\_t=o\_t|S\_t)]

Then next we need to combine the two core operation steps to complete the final prediction formula

First we initialize the 0 state



#然后我们使用之前完成的predict函数方法进行第一步前两项的运算

#然后再使用这个结果去调用update完成与P(O\_t|S\_t)的运算，一定要注意传递

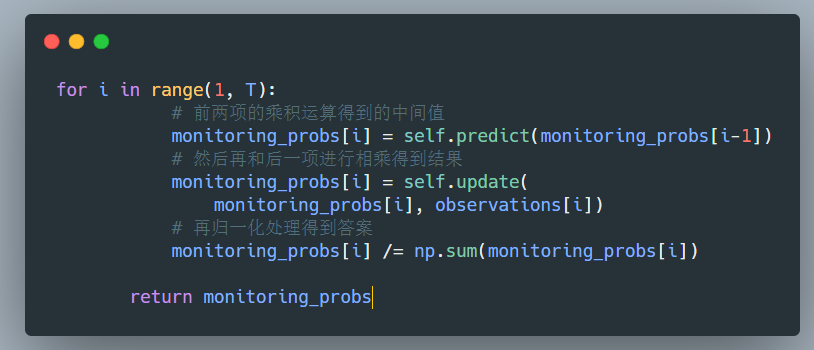
#的O[b]观测向量要保证切片正确

#然后我们再进行归一化处理即得到了预测的概率结果

Then we use the previously completed predict function method to perform the first two operations of the first step

Then use this result to call update to complete the operation with P(O\_t|S\_t), be sure to pay attention to the O[b] observation vector passed to ensure that the slice is correct

Then we normalize again to remember the predicted probability result



## 4. Result Analysis

# 介绍实验结果；

# 对实验结果进行分析，

# 例如：该结果是否符合你的预期，有什么优势（或者不足）

# 允许使用图片、表格等格式详细介绍你的实验结果

# 不建议直接粘贴所有代码，必要的情况下可以粘贴部分代码段，请保证格式正确，且不影响阅读体验。

# Explain the result of your experiment

# Analyze your result

# Here are some points of view you can used for analysis

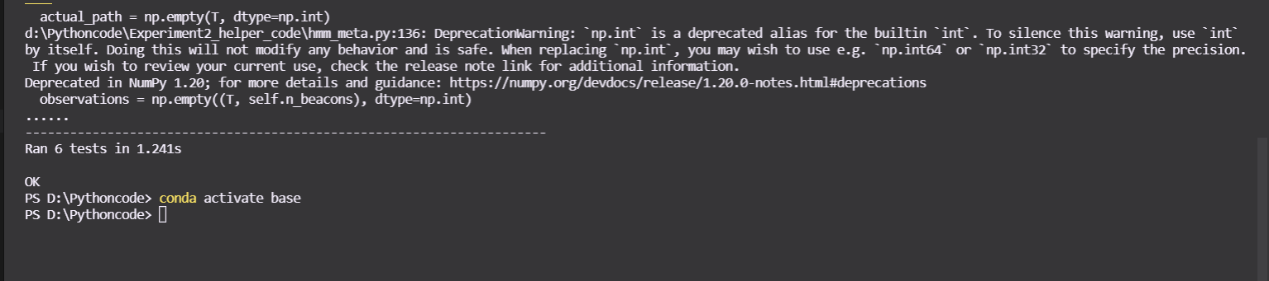
# Does the result meet your expectations ? What are the advantages or advantages of your implementation.

# You can use figures and tables to conclude the results of your experiment.

# Please DO NOT copy the entire code to this report, you can copy code snippets (some part of your code) if necessary. Please make sure the pattern of the code is clear and up to coding standard. Please make sure your code does not influence the reading.

#接下来我们运行test.py文件查看能否成功运行，最终终端输出如下：

Next we run the test.py file to see if we can successfully solve the problem, and the final terminal output is as follows.



输出为ok说明成功完成任务。具体的代码可以查看附件或者是我的git仓库：

## 5. Conclusion

# 简要介绍实验结论

# 介绍你的心得体会，以及对实验过程的反思

# Brief Introduction on your experiment results.

# How do you feel about your result and How do you feel about the experiment?

本次实验总的来说还是比较顺利，因为实在人工智能考试后完成，因此对HMM模型较为了解，这里一定要注意由于使用的是矩阵形式的相乘，和课件上对单个概率的求解还是由略微区别的，因此需要现在草稿纸上进行矩阵形式的公式规划。同时在本次实验时发现测试文件text是由问题的，他把测试数组定义放到了入口函数的下方 导致出现了未定义的错误，如果您也和我有类似的情况，那么请将测试数组移动至最前方即class声明的上方即可正常运行进行测试。

This experiment in general is relatively smooth, because it is done after the AI exam, so the HMM model is better understood, it must be noted that because the use of the matrix form of multiplication, and the class on the individual probability of the solution is slightly different, so you need to do the matrix form of the formula planning on the draft paper. At the same time in this experiment found that the test file text is by the problem, he put the test array definition to the bottom of the entry function resulting in an undefined error, if you also have a similar situation with me, then please move the test array to the forefront that is above the class declaration can be run normally for testing.

Translated with www.DeepL.com/Translator (free version)