Machine Learning Example 3：隱馬爾可夫模型

作者：廖柄煸

摘要

隱馬爾可夫模型（Hidden Markov Model)是一種統計模型，用來描述一個含有隱含未知參數的馬爾可夫過程。其從可觀察的參數中確定該過程的隱含參數，然後利用這些參數來作進一步的分析。本研究透過序列資料集建構，經利用學習演算法學習出對應的隱藏式馬柯夫模型，最後產出最有可能產生該序列O的狀態路徑，該技術應用非常廣泛，如語音、簽名辨識、中文斷詞/分詞或光學字符識別、機器翻譯、基因等等。

關鍵詞: 隱藏式馬柯夫模型

一、緒論

1.1 動機（探勘所選用的資料集之動機）

隨著科技日新月異，資料急遽的成長，從未知的資料中獲得潛在的規則是十分珍貴，隱馬可夫模型（Hidden Markov Model）是機器學習（Machine Learning）領域中常常用到的理論模型，從語音辨識（Speech Recognition）、手勢辨識（gesture recognition），到生物資訊學（Bioinformatics）裡的種種應用，都可以見到這個工具的身影。

生物資訊學（bioinformatics）是另一個大量使用到 HMM 的領域，從 DNA 序列的比對到演化歷程的推論，只要是跟基因序列有關的，幾乎都看得到 HMM 的應用。以 DNA 定序為例，一段採集到的 DNA 序列，包含了「外顯子」（exon）和「內隱子」（intron）兩種段落，兩者在細胞複製上有不同的功能，但都是由眾多的基因（gene，有 A, T, C, G 四種）排列成的序列，因此在一串看得到的基因序列中，要如何標記出哪一段是「外顯子」，哪一段又是「內隱子」，這些看不到的段落，也是 HMM 可以發揮作用之處。簡單的說，「外顯子」和「內隱子」各自包含 A，T， C，G 基因的比例不同，於是我們可以利用 HMM 相關的演算法，找出哪一個基因是「外顯子」和「內隱子」的起點或終點。另外，股票的價格變化也是一個「序列」，許多人 HMM 運用在預測股價的狀態上。。

1.2 目的

隱藏式馬可夫模型在 60 年代後期被提出，早期因為訓練需要花費大量的計算，因此比較少被應用，直到提出較佳的訓練方式時，這個方法才漸漸的受到重視。隱藏式馬可夫模型特點為在辨識能使用到時序上的資訊，因此在有時間先後關係的資料辨識上有著相當的優勢，目前在語音辨識上有著傑出的表現。近年來因為多媒體的發展，也被大量的應用在影像、影片的資料辨識上。

隱藏式馬可夫模型是由馬可夫鏈所延伸出來的模型，在一個馬可夫鏈中，能經由統計在 t 個時間觀測序列(Observation Sequence)的觀測值，進而估測當時間點 t+1 時，該序列的狀態為何。在馬可夫鏈的應用中，被觀測對像的狀

態是已知的、能直接量測到的。可是當觀測序列的狀態是不能直接量測得到時。如研究中之頭部狀態，我們並不能直接量測臉是處於何種角度，我們只能量測到特徵臉在低維主軸向的分布，這類型的應用馬可夫鏈便有所限制。而 Hidden Markov
Model 有三個問題需要處理：

1. 已知模型參數 λ λ, 求出 $P(O^i \mid M)$
2. 已知模型參數 λ 與 O^i, 找出 Q* = (q1q2…qT)
3. 已知資料集 X = $\{O^k\}$, 找出模型參數 M

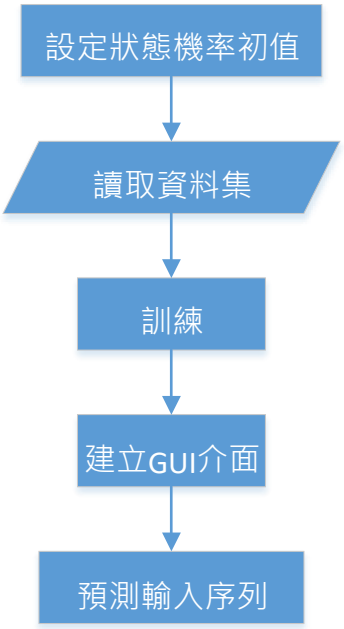
本實驗將對第三點問題：「已知資料集 X = { O^k }, 找出模型參數 λ」進行處理。利用學習演算法找出對應的隱藏式馬柯夫模型 M，輸出模型參數：M = (A, B, π)，A 為 State transition probabilities、B 為 Observation probabilities、π 為 Initial state probabilities。

起始參數估計，在 HMM 的架構中，一開始就需要決定的參數有四個：HMM 的狀態數目 (N)、λ = (A, B, π)。其中，π 和 A 可以使用均勻分布 (uniform probability distribution)；但是如果 B 也使用均勻分布，會讓 HMM 將所有狀態視為相同的狀態，而無法得到有效的參數。因此，要想辦法決定 HMM 的狀態數目和 B 這兩個參數的起始值。

2、 方法

使用 Python 套件 pomegranate(http://pomegranate.readthedocs.io/)，來訓練 HMM 的模型。而本實驗程式的流程如下圖，pomegranate 需要先給定狀態轉移機率矩陣與初始狀態機率的起始值方可進行訓練。

程式使用 Tkinter 來設計 GUI 畫面。Tkinter(Tk Interface)模組是 Python 的標準 Tk GUI 工具包的接口。Tk 和 Tkinter 可以在大多數的 Unix 平台下使用，同樣可以應用在 Windows 和 Macintosh 系統裡，並良好的運行在絕大多數平台中。



圖一：程式流程

3、 實驗

3.1 資料集　本實驗採用 hmmData.txt 資料集，進行 HMM 的模擬設計。

3.2 實驗設計（實驗如何進行、參數如何設定等）

假設該 HMM 有三個狀態{1, 2, 3}、每個狀態有四種可能輸出{a, b, c, d}，其餘的參數皆使用預設值。程式設計了簡單 GUI，可以輸入一個序列 Q，然後利用上述學得的模型 M，計算並輸出 M 產生該序列的機率有多少？以及最有可能產生該序列 Q 的狀態路徑為何？例如：輸入一個序列 Q = 'abcddcbaddccbbaa'。

3.3 實驗結果

訓練前 π 為[0.1 0.1 0.8]

A 為
[0.3 0.4 0.3]
[0.3 0.3 0.4]
[0.1 0.1 0.8]
訓練後 π 為[0.24196415 0.36506534 0.3929705
]

A 為
[4.09259686e-01 1.18390972e-01 4.72349342e-01]
[4.12689376e-02 9.58731007e-01 5.52060204e-08]
[2.41964151e-01 3.65065345e-01 3.92970504e-01]

輸入序列”abc”得到對數機率為：-5.0308，路徑如下圖

圖二：序列”abc”測試

輸入序列”cdcdcdaacd”得到對數機率為：-15.9855，路徑如下圖

圖三：序列”cdcdcdaacd”測試

4、 結論由以上實驗結果得知序列越長，出現的機率越小。看到一個觀察序列 abcddcbaddccbbaa ，但是看不到狀態序列 s_1 s_2 ..... sT 的情況下，找出所有可能路徑的機率總和。我們只能看到觀察序列的”果”，但是我們看不到狀態序列的”因”。此外使用 pomegranate 套件無法顯示 Initial State 是哪一個。