推薦系統API雛形 ─ 技術簡介 (Ver. 1.1)

**作者：** 陳韋辛 Frank Chen

**題目：** 「以類神經網路演算法和回歸分析實作個性化推薦系統之API雛形」

**對鴻揚之可能貢獻：**

商品評分與推薦系統日趨普遍，本人也對具有實用性的演算法相當感興趣，因此決定研讀並實作一個即便不懂該演算法也容易使用的推薦系統API，期望能協助拓展公司未來接案的種類，如**購物網站**或是**旅遊網站**等具有推薦功能之專案。

**功能：**

預測所有顧客對未知電影或產品之評價，以便未來對該顧客進行商品推薦，此外，此套件可以動態新增、更新顧客之評價或新增新產品，而不會干擾到原先已處理過的數據導致重新計算，相當符合實際使用情形。

**程式碼：**

* Github: <https://github.com/WeiHsinChen/RecommendationAPI>
* Google Drive: <http://ppt.cc/6iumD>

**使用環境：**

* 以Python語言實作
* 使用SQLite 資料庫
* 須下載SQLAlchemy便於資料庫存取以及numpy套件以便矩陣運算

**套件優點：**

使用方便簡單，足以滿足最基本推薦系統的需求。輕量化設計，相較於其他開放源程式碼的推薦系統套件(e.g., RecommendationAPI in Drupal)更加簡單易懂，也不用像Prediction API in Google Cloud Platform或是Machine Learning Recommendation API in Microsoft Azure等需要高額使用費用。此套件也支援動態更新資料，且越新的評價在分析中會佔越高之權重，越舊的評價會佔分析中較低之權重。此外，此套件雛型能夠輕易轉型成其他推薦形式，如將商品評分紀錄改成商品點擊紀錄等，便能計算顧客對該商品的未來點擊數以預測其喜好程度。

**待改善處：**

目前的輸入參數之格式完全是便於現場示範，且以Command-Line Interface做為展示平台，但未來只須依照使用者的需求做調整即可。此外，推薦準度以及運算速度有待最佳化，並且還有許多可以擴充之功能，如數據過大時使用Hadoop等Map reduce技巧降低計算之需求或是denoising提升準確性。

**近期預計之變更：**

1. 與Twisted套件結合，或是用相同的架構不同的語言實做，並做成Web-API供前台網站做使用。
2. 若本身資料夠多，即可省略步驟一. Save raw data與步驟二. Train function，不必一次分析大量資料。一開始可以直接逐筆加入商品與顧客之評價，因為當加入新顧客時系統便會再次局部性計算並分析資料。
3. 增加Cross Validation 功能以便使用者能選取適當的神經元數量與迴圈數。
4. 目前並沒有建立delete功能的原因是就算是過時的顧客或是商品，也對分析有一定的貢獻度，因此除非資料實在過於龐大，不然沒有必要進行刪除作業，不過之後也會依照使用需求加上該功能。

**程式架構圖：**

rpsy ────── task ──── Demo.py Controller層

│ │─ RS.py 負責所有數學運算

│

│─── db ──── rs.py 負責資料庫存取

│ │─ server.py 指定資料庫位置

│ │─ utils.py 資料庫操作工具

│

│─── utils ──── data.py 資料庫操作工具

│

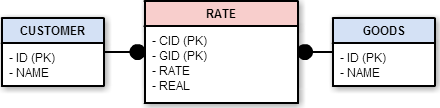
│─── metadata ──── rs.py Models

│

│─── data ──── rs.sqlite 資料庫

│─ VTable.txt 運算結果之V矩陣

│─ WTable.txt 運算結果之W矩陣

資料庫關聯圖：

* CUSTOMER ─ ID (int) 顧客ID

─ NAME (int) 顧客姓名

* GOODS ─ ID (int) 商品ID

─ NAME (int) 商品名稱

* RATE ─ CID (int) 顧客ID

─ GID (int) 商品ID

─ RATE (float) 評分

─ REAL (bit) 顧客評分(1)或預測評分(0)

推薦系統API雛形 ─ 使用說明 (Ver. 1.2)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能描述 | | 初次存取評價資料 | | | | | 業務別 | | 資料存取 |
| 函式名稱 | | /Demo/save\_raw\_data | | | | | 傳輸介面 | |  |
| 使用情境 | | 初次使用API時，僅需執行一次。 | | | | | | | |
| 備 註 | | 第一次使用時，必須將全體用戶之評分紀錄文字檔(.txt)匯入資料庫，以便存取與分析，因此使用者須將評分文字檔之「相對路徑」為輸入，呼叫Demo.py中的save\_raw\_data方程式，文字檔格式參考Google問卷調查之輸出格式，如data/raw\_data.txt。 | | | | | | | |
| 輸 入 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | | 欄位名稱 | 說 明 | 欄位型態 | | 備註 | |
| 0 | Attribute | | | data | 評價資料 | char | |  | |
| 範 例 | * Command: cd C:\RecommendationAPI\rspy\task   Demo.py save\_raw\_data ../data/raw.txt   * Content-Type: "application/text" * Data:   "Timestamp Name 復仇者聯盟 鳥人  2015/05/05 Frank 5  2015/05/05 Andy 5 5  2015/05/07 Allen 5 5"   * 上述Data每行中的每筆資料相隔一個Tab，若某顧客對該筆商品沒有評價，則不用輸入其數字，如同上例Frank只評價電影”鳥人”。 | | | | | | | | |
| 輸 出 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | 欄位名稱 | | 說 明 | 欄位型態 | | 備註 | |
|  |  | |  | |  |  | |  | |
| 範 例 |  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能描述 | | 初次計算測試資料 | | | | | 業務別 | | 資料分析 |
| 函式名稱 | | /Demo/train | | | | | 傳輸介面 | |  |
| 使用情境 | | 儲存完資料後，初次計算測試資料，僅需執行一次。 | | | | | | | |
| 備 註 | | 第一次使用時，必須計算所有用戶之評分資料，以產生矩陣 V 和矩陣 W以便預測全體用戶之行為，因此使用者必須呼叫Demo.py中的train方程式，並輸入參數「神經元數量」以及「計算迴圈數」，若無輸入參數，則預設神經元數量為10，計算迴圈數為10000000。神經元數量與計算迴圈數之物理意義請詳閱「必要之參數物理意義簡介」。 | | | | | | | |
| 輸 入 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | 欄位名稱 | | 說 明 | 欄位型態 | | 備註 | |
|  |  | |  | |  |  | |  | |
| 範 例 | * Command: cd C:\RecommendationAPI\rspy\task   Demo.py train 8 50000   * 上述命令中每組字串以空格分隔，第一個數字代表「神經元數量」、第二個數字代表「計算迴圈數」 | | | | | | | | |
| 輸 出 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | | 欄位名稱 | 說 明 | 欄位型態 | | 備註 | |
| 0 | Attribute | | | Ein | 測資誤差 | char | |  | |
| String  範 例 | * Data: “0.02” | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能描述 | | 商品推薦 | | | | | 業務別 | | 商品推薦 |
| 函式名稱 | | /Demo/rec\_for\_a\_cus | | | | | 傳輸介面 | |  |
| 使用情境 | | 推薦顧客未知商品。 | | | | | | | |
| 備 註 | | 推薦顧客未評價之商品時，使用者須要呼叫Demo.py中的rec\_for\_a\_cus方程式，並輸入參數「顧客ID」以及「欲推薦數量」，若無輸入「欲推薦數量」，則欲設為3，返回值為該數量之推薦商品資料‧ | | | | | | | |
| 輸 入 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | 欄位名稱 | | 說 明 | 欄位型態 | | 備註 | |
|  |  | |  | |  |  | |  | |
| 範 例 | * Command: cd C:\RecommendationAPI\rspy\task   Demo.py rec\_for\_a\_cus 43 3   * 上述命令中每組字串以空格分隔，第一個數字代表「顧客ID」、第二個數字代表「預推薦數量」 | | | | | | | | |
| 輸 出 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | | 欄位名稱 | 說 明 | 欄位型態 | | 備註 | |
| 1 | Attribute | | | GID | 商品ID | char | |  | |
| 1 | Attribute | | | NAME | 商品名稱 | int | |  | |
| 1 | Attribute | | | RATE | 預測評分 | int | |  | |
| String  範 例 | * Content-Type: "application/json" * Data:   [{“ID”: 17, “NAME”: “求婚大作戰(日語劇)”, “RATE”: 9.56},  {“ID”: 25, “NAME”: “曼哈頓戀習曲 Began Again”, “RATE”: 9.55},  {“ID”: 12, “NAME”: “模仿遊戲”, “RATE”: 9.54}] | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能描述 | | 新增商品 | | | | | 業務別 | | 資料存取 |
| 函式名稱 | | /Demo/add\_goods | | | | | 傳輸介面 | |  |
| 使用情境 | | 新商品上架時，將此新商品加入預測系統。 | | | | | | | |
| 備 註 | | 新增商品時，須呼叫Demo.py中的add\_goods方程式，並將商品名稱之文字檔(.txt)作為輸入參數即可。 | | | | | | | |
| 輸 入 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | 欄位名稱 | | 說 明 | 欄位型態 | | 備註 | |
|  |  | |  | |  |  | |  | |
| 範 例 | * Command: cd C:\RecommendationAPI\rspy\task   Demo.py add\_goods ../data/new\_goods.txt   * Content-Type: "application/json" * Data: [“鴻揚超人首部曲”, “鴻揚超人二部曲”] | | | | | | | | |
| 輸 出 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | | 欄位名稱 | 說 明 | 欄位型態 | | 備註 | |
| 1 | Attribute | | | ID\_List | 新商品之商品ID | Char | |  | |
| String  範 例 | * Data: [“29”, “30”] | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能描述 | | 新增顧客 | | | | | 業務別 | | 資料存取 |
| 函式名稱 | | /Demo/add\_a\_cus | | | | | 傳輸介面 | |  |
| 使用情境 | | 新顧客(會員)加入時，將此新顧客加入預測系統。 | | | | | | | |
| 備 註 | | 新增顧客時，須呼叫Demo.py中的add\_a\_cus方程式，並將該用戶之評分文字檔(.txt)作為輸入參數即可。 | | | | | | | |
| 輸 入 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | | 欄位名稱 | 說 明 | 欄位型態 | | 備註 | |
| 1 | Attribute | | | Name | 顧客姓名 | char | |  | |
| 2 | Attribute | | | GID | 商品ID | int | |  | |
| 2 | Attribute | | | RATE | 評分 | int | |  | |
| 範 例 | * Command: cd C:\RecommendationAPI\rspy\task   Demo.py add\_a\_cus ../data/new\_cus.txt   * Content-Type: "application/json" * Data:   {"NAME":"新顧客之姓名",  "DATA": [{"GID": 1, "RATE":5},  {"GID": 3, "RATE":6},  {"GID": 14, "RATE":1},  {"GID": 16, "RATE":3},  {"GID": 21, "RATE":9}]} | | | | | | | | |
| 輸 出 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | 欄位名稱 | | 說 明 | 欄位型態 | | 備註 | |
| 1 | Attribute | | ID | | 新顧客之ID | Char | |  | |
| String  範 例 | * Data: “44” | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能描述 | | 更新顧客資料 | | | | | 業務別 | | 資料存取 |
| 函式名稱 | | /Demo/update\_a\_cus | | | | | 傳輸介面 | |  |
| 使用情境 | | 顧客更改評價或會員名稱時。 | | | | | | | |
| 備 註 | | 更新顧客評分時，須呼叫Demo.py中的update\_a\_cus方程式，並將該用戶之評分文字檔(.txt)作為輸入參數即可。 | | | | | | | |
| 輸 入 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | | 欄位名稱 | 說 明 | 欄位型態 | | 備註 | |
| 1 | Attribute | | | ID | 顧客ID | int | |  | |
| 1 | Attribute | | | Name | 顧客姓名 | char | |  | |
| 2 | Attribute | | | GID | 商品ID | int | |  | |
| 2 | Attribute | | | RATE | 評分 | int | |  | |
| 範 例 | * Command: cd C:\RecommendationAPI\rspy\task   Demo.py update\_a\_cus ../data/update\_cus.txt   * Content-Type: "application/json" * Data:   {“ID”: 43,  "NAME":"新顧客",  "DATA": [{"GID": 25, "RATE":1},  {"GID": 26, "RATE":1},  {"GID": 27, "RATE":1}]} | | | | | | | | |
| 輸 出 資 料 說 明 | | | | | | | | | |
| 層 級 | 屬性 | | 欄位名稱 | | 說 明 | 欄位型態 | | 備註 | |
| 1 | Attribute | | ID | | 更新顧客之ID | Char | |  | |
| String  範 例 | * Data: “43” | | | | | | | | |