

# 使用者行為分析與商品推薦應用於集點 APP

User Behavior Analysis and Commodity Recommendation for Point-Earning Apps

陳昱瑾

中央大學資工所碩士班  
julie25599@gmail.com

張嘉惠

中央大學資工系教授  
chiahui@g.ncu.edu.tw

楊佳靜

中央大學資工所碩士班  
uiuuty2003@gmail.com

陳品良

資策會  
mileschen@iii.org.tw

谷圳

資策會

cujing@iii.org.tw

廖彥鈞

中央大學資管系  
cherry84228@gmail.com

楊秉哲

資策會  
maciacClark@iii.org.tw

## 摘要

近年來，因為電子商務的發展隨之而生的推薦系統蔚為風潮。然而，推薦系統背後仰賴的是大量的數據和有效的分析，對於初創發展不久的企業來說是很大的問題。我們設計了幾種方法來預測，有傳統常見的分類器、Heuristic Scoring 方法，也有利用機器學習來建立推薦系統，整合協同式及內容式推薦系統，建立混合模型的 CCAM 演算法 (Co-clustering with augmented matrix) [1]。

本文使用的資料來自於一款以集點任務為設計導向的 App，這個 App 的任務及消費資訊，是藉由綁定 Facebook 帳戶，取得個人相關資訊及網路萃取產品資訊。不過由於這個 App 使用率低，因此大部份使用者記錄少於十筆，屬於 cold-start 使用者分析問題，是我們面臨的最大挑戰。為了獲得較充分的購買紀錄，我們嘗試鎖定使用次數較多的使用者來做分析，另外也利用網路爬蟲來增加商品的特徵資訊，實驗數據顯示 F-measure 由原本的 0.756 提升至 0.802，藉此提升推薦系統的效能並進行較為有效的推薦。

**關鍵詞：**使用者行為模版、時間序列性推薦、Co-Clustering、Matrix-Factorization。

## 1. 緒論

隨著網際網路的發展，社群網站及電子商務網站盛行，使得傳統零售市場的時空界線被打破，使用者的消費型態漸漸隨之改變成線上購物。從過去的「固定地點」到「隨處可得」的消費，對零售業儼然形成重大的變革。

為因應此變化，商家紛紛改變策略，以吸引消費者上門消費，而網路推薦的行銷手法尤為常見。然而面對資訊爆炸的時代，對於商家來說，琳瑯滿目的推薦資訊推給大部分無需求購買的消費者是徒勞無功的，而對於消費者來說，需要花更長時間找到所需的商品，更不會

有意願到店裡做消費。

為解決消費行為改變及資訊過載的問題，協助商家推薦及使用者做決策的「個人化推薦系統」因應而生。推薦系統仰賴大數據，利用使用者常瀏覽的社群或電子商務平台，蒐集並分析使用者個人資訊與瀏覽及購買行為的資訊，進而篩選並預測消費者感興趣的資訊，讓業者能有效的進行客製化推薦和行銷。而現在許多 App 也會結合集點的活動，來達到推廣各式廣告行銷的手法，例如現在時下廣為被人使用的通訊軟體 Line，也與商家合作，透過蒐集 Line 的點數，顧客可以獲得商家的折扣優惠，而另一款免費應用程式 MyCard 更是符合任務集點導向的 App。

本研究所使用的資料來自一款以集點任務作為設計導向的 App。這個 App 提供的任務共有三種型態：到店任務 (Check-in)、掃描任務 (Scan) 及消費任務 (Buy)，使用者只要完成任務即可獲得點數，藉此鼓勵 App 用戶前往實體店面消費。另外，此 App 需綁定 Facebook 登入，因此使用者的相關資訊，如年齡、性別、居住區域，按讚過的粉絲專頁均可獲得，另外使用者執行任務的各項資訊 (店家資訊、任務內容及種類、消費時段、消費金額等)，也都記錄於 App 的系統。因此能建立使用者基本資訊。

目前此集點 App 雖有 79,563 人下載 App，但執行過任務的人數只有 17,528 人，實際上有購買紀錄的只有 8,235 人 (佔總下載量的 10.3%)。而三類共 408,577 次執行任務中，然消費任務僅佔 15.7%，只有 1,715 人消費次數高於 20 次。

其中某連鎖美妝零售商的消費佔 81% 交易量，共分為 2,177 個產品及 144 項類別。從 103 年 10 月到 105 年 7 月，平均每天有 128 人到店及消費，而且消費者沒有購買的比例高達 83.4%。由此可見從到店打卡到消費的任務轉換，仍有相當大的進步空間。

現今常見的推薦系統有兩種策略：內容式過濾 (Content-based filtering or CB)[2] 及協同式

過濾 (Collaborative filtering or CF) [3]。內容式過濾是根據使用者過去瀏覽的紀錄，找出產品特徵，建立顧客的喜好模式，並且推薦類似的產品；而協同式過濾則以使用者之間偏好為主體，推薦其相似偏好使用者瀏覽的商品給另一位使用者。

在有限的數據提升推薦系統的效能和準確度，是我們面臨的最大挑戰，因此本文使用了許多方法做比較，其中 CCAM 共分類演算法，為一種整合協同式及內容式推薦系統，結合傳統推薦系統的優點，對 8,235 使用者及 144 個中分類進行共分類，得到十個使用族群及三大產品類別。實驗結果最高得到 0.402 的 F-measure，比傳統 User-based 與 Item-based CF 方法的效能好。

我們嘗試使用分類器來預測，特別取出購買次數高於 20 次以上的 1,715 位使用者，和 144 項中分類的商品組合成 User-Item pair，利用 FB 的資訊找出使用者特徵，並利用網路爬蟲來增加商品特徵，實驗結果 F-measure 提高到 0.785。

另外我們也設計了一條啟發式評分公式，公式中結合使用者所屬族群偏好、個人購買習慣與商品類別關聯幾項要素。實驗結果 F-measure 最高可達到 0.802，效能高於其他方法。

我們認為若購買紀錄夠充分必定可以使效能更提升，所以我們鎖定使用次數較多的使用者來分析。未來我們想加入時間這個要素來增加預測的效能，預計會使用 sequential pattern mining 分析產品的順序來改善推薦的效能。

## 2. 相關研究

現下推薦系統主要有兩種技術：內容式過濾(Content-based filtering or CB) [2]及協同式過濾(Collaborative filtering or CF) [3]。內容式過濾是根據顧客喜好的商品資料，找出產品特徵，建立顧客的喜好模式，並且推薦和顧客喜好的類似產品；而協同式過濾，主要以使用者點選產品之歷史資料(Click data)進行分析，透過訓練方法，找出偏好相似的使用者，推薦其喜好的商品給當前使用者。

另外，協同式過濾又可分為 Memory-based CF [4]及 Model-based CF [5]兩項。Model-based CF 會先分析歷史資料，針對資料進行模型訓練，再根據建立的模型對使用者和產品的評價進行預測；而 Memory-based CF 直接運用過去的歷史紀錄，預測使用者對商品的評價，又可分為 Item-based 或 User-based 兩種類型。User-based 的概念是尋找具有相似興趣的使用者，以此推薦該使用者喜好的商品；Item-based 則是尋找與此項目有共同購買樣板的商品，藉以找出可能購買一個能夠引起使用者興趣的項目或物品，

會與之前評分高的項目相似，所以便是利用項目之間的相似程度來當作評斷的標準。

混合式推薦系統，主要是結合以上兩者之推薦系統，常見的結合方式包含用 Memory-based CF 結合 Content-based filtering，或者 Model-based CF 結合 Content-based filtering 兩種方式。常見的作法包含先利用內容式推薦找出適合使用者的候選產品，然後再使用協同式推薦系統針對這些候選產品進行過濾，以改善預測準確率。

## 3. 資料介紹與問題定義

本研究的資料是來自一款以集點任務作為設計導向的 App，主要目的是希望鼓勵 App 用戶前往實體店面消費。這個 App 提供的任務共有三種型態：包括到店任務 (Check-in)、掃描任務 (Scan)及消費任務 (Buy)，使用者只要完成任務即可獲得點數，藉此能得到商品資訊。另外，使用者需透過 Facebook 註冊登入，因此能夠得到使用者相關資訊。綜合上述兩種資料，得到 User feature 和 Item feature，藉此更了解消費者的行為模式。

### 3.1 使用者特徵

App 藉由和 Facebook 綁定的方式進行註冊登入，藉此取得使用者的相關資料，包含使用者 FB 的 ID、性別、年齡、居住地、按讚過的粉絲專頁等等。並且，由執行任務來了解使用者的消費習慣，如光顧的店家、執行哪些任務、執行的時間、點數配發等等。我們藉由上述的資訊，作為消費者的特徵。每個特徵皆以二元方式表示，共計 50 個特徵，如表 1。

表 1. 使用者的特徵

年紀	20 歲以下、20~40 歲、40 歲以上
性別	男性、女性
居住區域	北、中、南、東、外島、來台旅遊者
FB 的分類	女學生,小資女,小資男,父親,母親,存款族,男學生,其他,保健,流行,科技愛好者,美妝,美食,旅遊,追星族,運動員,親子 (1 個小孩),親子 (2 個以上小孩)
FB 類別(食)	是否有按飲食粉絲頁的讚
Fb 類別(衣)	是否有按衣著粉絲頁的讚
Fb 類別(行)	是否有按交通粉絲頁的讚
Fb 類別(育)	是否有按教育粉絲頁的讚
Fb 類別(樂)	是否有按娛樂粉絲頁的讚
到店任務	是否執行過到店任務
掃描任務	是否執行過掃描任務
消費任務	是否執行過消費任務
消費金額	消費金額少於平均(600)、大於平均(600)
購買商品	吃喝類、穿用類、育樂類
消費時段	早、中、晚、非營業時間
消費時間	平日、假日
消費時間	節日、分節日

為了解使用者各項特徵是否有無顯著差異

性，我們採用二因子變數分析 (two-way ANOVA)，來檢定 feature 間有無達到顯著水準。

變異數分析(ANOVA 分析)是較常用來作資料分析的統計方法。檢驗方式是使用 F 檢驗 (F-test)，並透過 F 分布圖估計出 p-value，決定各組資料間有無顯著差異 (significant differences)。

在統計上，p-value 越小，拒絕虛無假說 ( $H_0$ ) 的證據越充分，即所作的 F 檢驗顯著差異性越高。當  $p < 0.05$ ，則定義此 F 檢驗具有顯著差異；當  $p < 0.01$ ，則定義 F 檢驗具有非常顯著的差異。

我們希望透過 ANOVA 分析來了解消費者行為的差異，進而找出那些 feature 對於消費者行為有較為顯著的影響。因此，我們挑選 10 個具有代表性的使用者特徵作交叉分析。

首先，假設各組資料之間的母體皆為常態分配、母體變異數相同、母體隨機抽取的樣本皆獨立。然後，採用二因子變數分析，以此能評估兩個特徵有無明顯的交互作用。

結果如表 2，我們將各組資料具有顯著差異的，即此檢驗的  $p < 0.05$ ，用空心菱形框標示。若各組資料具有非常顯著的差異，即此檢驗的  $p < 0.01$ ，則用實心菱形框標示。

表 2. ANOVA 分析結果

◇ 具有顯著差異 ( $p < 0.05$ ) ◆ 具有非常顯著差異 ( $p < 0.01$ )

	性別	身軀	居住地	FB 類別	任務	消費高低	商品類別	早中晚	平/假日	(非)節日
性別										
年紀										
居住地		◇								
FB 類別			◆							
任務		◇	◆							
消費高低		◇	◆							
商品類別			◆		◇	◇				
早中晚			◆	◆	◆	◆	◇			
平/假日			◇							
(非)節日			◇							

由表二，透過實驗結果得出兩個結論：居住地及早中晚的消費時段和其他 feature 具有高度相關性，另外年紀、任務類別、消費高低和商品類別與其他 feature 有較次等程度的相關性。

### 3.2 產品特徵

在商品資訊上，主要的消費任務來自某美妝零售商的消費交易 (超過 8 成)。此美妝零售商為個人美妝生活用品百貨，在全台灣有 137 家的分店，分店遍及城市到鄉鎮，尤其在鄉下地區，大多沒有百貨公司進駐，這種中型生活百貨足以吸引全城的民眾消費光顧。此商家販售的商品有美妝、飾品、食品、五金用品等等，多達 2,177 項產品，分為 28 大類、144 中分類及 377 小分類。如表 3 所示。

表 3. 大分類、中分類與小分類列表 (部分)

大分類	中分類	小分類
3C 產品	DIY 用品	1028 Visual Therapy
保養品	乳酸飲料	GATSBY(男仕)
包包收納	乾類	LUCIDO-L
季節搭配類	事務用品	L 夾/U 型夾/附桿文件套
家庭生活	休閒衣褲	Maybelline 媚比琳

另外，我們也利用網路爬蟲，取得產品相關資訊。首先，將產品中分類和小分類的 Google 搜尋的前十筆 title，透過 Python 的 package jieba 進行斷詞，並結合兩種關鍵字提取 (keyword extraction) 的方式—TFIDF[6] 和 TextRank[7]。

TFIDF 是常見於文本檢索的統計方式，用來評估字詞在文本中的重要性。而 TextRank 是在 PageRank 公式的改良下的統計方法，對文本中字詞作更進一步的權重算法。以下是 PageRank 和 TextRank 的公式。

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} * S(V_j) \quad (1)$$

其中  $S(V_i)$  是第  $i$  個頁面的 PageRank， $d$  為阻尼係數，為 0 到 1 的數值，通常設為 0.85。 $In(V_i)$  是指向  $i$  個網頁的網頁集合， $Out(V_j)$  是網頁集合指向  $j$  個網頁。

$$WS(V_i) =$$

$$(1 - d) + d * \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{k \in Out(V_j)} w_{jk}} * WS(V_j) \quad (2)$$

其中  $WS(V_i)$  為在句子  $i$  的權重， $W_{jk}$  為句子  $j$  和  $k$  的相似度。

另外，我們對 TFIDF 和 TextRank 作 POS 詞性篩選，從 jieba 總共 55 類的詞性中，挑出四種詞性—名詞、動詞、地名、動名詞，作 POS 篩選，過濾掉不需要的資訊。

利用此三種方法，包含無詞性篩選 TFIDF 和有詞性篩選的 TFIDF、TextRank，各自提取前五個關鍵字作聯集，作為 Item feature。結果如下表 4 和表 5。表 4 中的第一項—DIY 用品可看出關鍵字中的 DIY、手工、用品、德昌 (商家)、設計、材料等有相關。大體上，這些關鍵字能補充中分類和小分類的資訊，即作為產品特徵。

表 4. 中分類之關鍵字提取結果 (部分)

中分類	關鍵字
DIY 用品	DIY, 手工, 用品, 德昌, 設計, 材料, 購物
乳酸飲料	飲料, 乳酸, 乳酸菌, 市售, 太甜, 網路, 結果, 開水, 營養師
乾類	肉乾類, 果乾類, 產品, 乾類, 介紹, 圖片, 農畜, 上購, 台灣, 結果, 魚骨, 搜尋
事務用品	事務, 用品, 拍賣, 文具, Yahoo, 公事, 公司, 機會
休閒衣褲	居家, 睡衣, 購物網, momo, 內塑衣, 休閒, 購物, 內塑, 運動, 拍賣

表 5. 小分類之關鍵字提取結果(部分)

小分類	關鍵字
1028 Visual Therapy	1028, Therapy, Visual, 保養, 上妝, 網路, 購物金, 圖片, 台灣, 結果, 經驗, 累積
DODO	Dodo, Facebook, DoDo, 豆豆, 烏甜點, 會員, 帳號, 甜點, 搜尋, 豆友, 登入
ELSIA	ELSIA, 艾兒, 粉餅, 希亞, 透白, 高絲, 采保, 美妝
GATSBY(男仕)	GATSBY, 男士, 傑士, 產品, gatsby, 圖片, 男仕, 保養, 總覽
LUCIDO-L	LUCIDO, 樂絲朵, Facebook, Lucido, UrCosme, 食記, 電棒, 化妝品, 棠小, 快乾, 結果, 絲絨, 空氣感

### 3.3 使用者購買商品之因素特徵

我們嘗試找出幾個可能影響購買行為的因素，建立使用者與商品的配對關係，首先想到的是使用者族群的購買偏好，不同族群有著不同的購買習慣，像是小資女族群購買美妝產品的比例較購買其他商品的比例來的高，我們定義族群購買偏好的計算公式如下。

$$f(u, i) = \frac{\sum_{v \in G(u)} c(v, i)}{\sum_{j \in I} \sum_{v \in G(u)} c(v, j)} \quad (4)$$

其中  $u$  代表 user、 $i$  代表商品類別。 $\sum_{v \in G(u)} c(v, i)$  表示一年內使用者  $u$  所屬族群  $G(u)$  購買商品類別  $i$  的次數，使用者的族群是由 Facebook 的資訊來分類，共分為 18 類族群，有男學生、女學生、存款族、親子等族群。分母為  $u$  所屬族群總購買所有商品類別的次數。

接著我們認為個人購買偏好也是影響購買行為的關鍵要素，我們觀察使用者購買某項商品的週期，並在使用者可能購買時推薦給他。也就是說超過使用者  $u$  購買商品類別  $i$  的時間點，但使用者尚未購買，則將分數記為 1.5 倍。個人購買偏好的計算公式如下。

$$p_t(u, i) = \begin{cases} c(u, i), & \text{if } t - l_{u,i} < P_{u,i} \\ c(u, i) * 1.5, & \text{otherwise} \end{cases} \quad (5)$$

其中  $t$  表示當下的時間點、 $l_{u,i}$  表示上一次使用者  $u$  購買商品類別  $i$  的時間、 $P_{u,i}$  表示預測使用者  $u$  會購買商品類別  $i$  的時間，我們以上次購買時間點加上平均購買週期作為預測購買的時間。

同時購買的商品也是我們所感興趣的要素，我們利用常被一起購買的商品，來推測某項商品被購買的分數，計算公式如下。

$$h(u, i) = \text{Max}_{j \in I(u)} (c(u, j) * \text{supp}(i, j)) \quad (6)$$

$I(u)$  為使用者  $u$  在一年內購買該商品類別集合。 $\text{supp}(i, j)$  則為在總交易量下，商品類別  $i$  和商品類別  $j$  同時出現在交易中的比例。並將該比例乘以使用者  $u$  購買該項產品類別  $j$  的次數  $c(u, j)$ ，最後找出最大值，作為同時購買該項產品的分數。

### 3.4 問題定義

我們的目標是針對不同消費者，推薦專屬的商品類別，進而促進消費。首先我們考慮到要選擇哪些消費者來作商品推薦。

目前 App 雖有 7 萬多人下載 App，但執行過任務的人數只有 17,528 人，實際上有購買紀錄的只有 8,235 人，其中消費次數少於 10 次的人數又超過一半，如圖 1。表示我們僅能掌握非常少數使用者的部份消費習慣，這也是我們所面臨的第一個挑戰。

這個 App 大多數的使用者，多執行到店或掃描任務來賺取點數，只有少部分的使用者有消費，在這些少數消費的使用者中，大多又只購買過一次，所以我們將會比較所有使用者與購買次數較多的人進行預測，觀察其效能。

圖 1 為使用者與其累計購買商品數的統計表。舉例而言，使用者  $U$  在第一次消費時，購買 2 個商品 A 與 3 個商品 B，又在第二次消費時，購買 5 個商品 A，則使用者  $U$  總購買 10 個商品，屬於第 3 條 bar (6~10 個)。

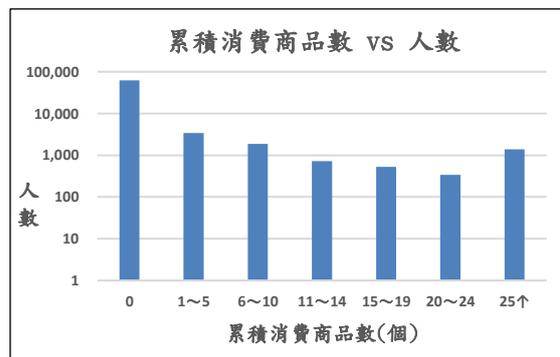


圖 1. 累積消費商品數 vs 人數

接著面臨的還有商品資訊不足的問題。關於商品的資訊僅有商品名稱，以及其大、中、小的分類。所以我們將 144 中分類和 377 小分類透過網路爬蟲並關鍵字提取，以提供更多產

品相關資訊和特徵。

#### 4. 方法設計

我們採用幾個方法來做預測，有 CCAM、分類器、以及啟發式評分公式，利用這些方法找出分數高的來推薦。

##### 4.1 CCAM 推薦方法

我們參考 Co-clustering with augmented matrices (CCAM)[1]的作法，如圖 2，他是利用過去的購買記錄、使用者特性資料(User profile)及產品特徵(Item feature)等來源進行分群，以解決共分群時僅能針對購買記錄進行分群的問題；待共分群完之後，產生共分群的結果作為分類資料之特徵，再利用監督式演算法(Supervised Learning)對輸入資料進行訓練，最小化其預測誤差，最後產生預測模型(Prediction Model)。

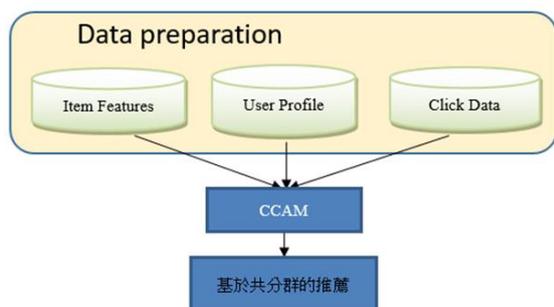


圖 2. CCAM 推薦系統架構圖

我們將過去的購買記錄、使用者特性資料(User profile)及產品特徵(Item feature)轉換為三個矩陣。

第一個矩陣的兩個維度分別是使用者與商品類別。我們選出有消費紀錄的 8,235 位使用者，商品則是以 144 類中分類做代表。矩陣中以 1 表示某使用者有購買某項商品，0 表示無購買紀錄。在選擇商品分類時，我們考慮到大分類包含的商品太廣泛，無法適當地找到適合推薦的商品，而小分類又細分到商品品牌，站在廠商的角度，他們希望可以自行選擇要推薦的品牌，例如挑選出利潤較高的品牌來推薦，又或是考量目前商品的庫存情況，使廠商能自行調配推銷的商品。

第二個矩陣是使用者及其特性(User profile)，與上個矩陣相同的 8,235 位使用者，將使用者的基本資料、FB 粉絲頁的喜好與任務執行的次數和時間等 50 個特徵，作為使用者特徵。矩陣中以 1 表示某使用者符合某特徵，0 表示不符合某特徵。

最後一個矩陣是產品與產品資訊，我們選

擇 144 類代表商品，並以小分類及 web 篩選的產品特徵做為商品的特徵，共有 1,171 個特徵。CCAM 輸入的 3 個矩陣統計資料如表 6。

表 6. CCAM 輸入資料統計

matrix	matrix size	非 0 比例
User-item	8,235*144	5.1%
User-user feature	8,235*50	26.2%
item-item feature	144*1,171	1.0%

透過 CCAM 演算法，我們可以同時將最近的使用者與商品類別分為同一群。如圖 3 所示為經過 CCAM 演算法後的結果，我們將有購買紀錄的 8,235 位使用者分為 10 群，144 類商品分為 3 群。

圖中每個小黑點表示某使用者有購買某商品類別，而有幾個區域較黑，代表著那群使用者大多數都有購買那群的商品類別。有些區域較稀疏，代表著那群使用者較沒有購買那群商品。

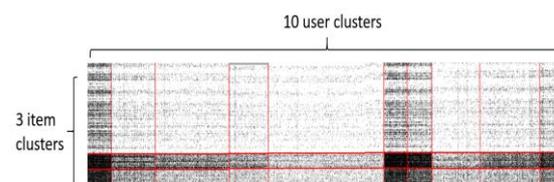


圖 3. 所有購買資訊的 CCAM 結果

利用 CCAM 結果，我們發現某些商品類別被分在同一群，表示這群商品為這群使用者所需要的，如下表 7 為我們將商品類別歸類後的分群結果。

表 7. 商品類別的分群結果

大分類	第一類	第二類	第三類	大分類	第一類	第二類	第三類
生活居家	▽	▽	▽	衛生用品		■	
零食	▽	▽	▽	購物袋		■	
包包收納	▽		▽	3C 產品			●
皮件領帶	▽		▽	文具			●
季節搭配類	▽		▽	食品			●
服飾	▽		▽	旅行			●
飲料	▽		▽	旅遊			●
盥洗用品	▽		▽	菸酒			●
沐浴		▽	▽	節慶用品			●
保養品		▽	▽	運動			●
整髮用品		▽	▽	飾品配件			●
醫療保健		▽	▽	親子			●
襪類		▽	▽	寵物			●
美妝		■		贈品			●

由上表得知美妝、衛生用品與購物袋(正方

形符號)在同一群,也就是他們的商品關聯度會較高,直觀上推薦同一群的商品類別給使用者,較容易推薦成功。3C 產品、文具等商品類別(圓形符號)則是另一個群。

共分群完之後,再將共分群的結果作為分類資料之特徵,再利用監督式演算法(Supervised Learning)對輸入資料進行訓練,輸入資料(如下表 8),最後產生預測模型(Prediction Model)。

表 8. Supervised Learning 的輸入資料

Pair <i>u, a<sub>i</sub></i>	User profile			Item feature			User C.C. feature			Item C.C. feature			Target
	$Q_1$	...	$Q_n$	$P_1$	...	$P_n$	$UA_{u_1}$	...	$UA_{u_n}$	$AU_{a_1}$	...	$AU_{a_n}$	
	0	...	1	1	...	0	12	...	20	-1	...	-1	0
	0	...	1	0	...	1	-1	...	-1	20	...	30	1
	1	...	0	1	...	1	25	...	35	0	...	10	0

## 4.2 分類器介紹

我們利用 weka 中的 libSVM [8]分類器對資料進行訓練及測試,主要希望透過分類的方式對使用者進行預測,以知道使用者們是否將會購買該項產品。

針對有購買的商品的 8,235 位使用者,我們特別取出使用頻率達 20 次以上的 1,715 位使用者,和 144 項中分類的商品組合成 User-Item pair,特徵的部分則結合 50 個使用者特徵(如表 1)、375 個產品特徵以及 796 個從 web 篩選後的關鍵字作為特徵,這些特徵全部轉換為二元特徵的形式,最後預測使用者是否會購買該項產品(標記為 1 或 0),如下列式(3),共計 1,221 個特徵。

$$k(u, i) \begin{cases} 1, \text{使用者 } u \text{ 有購買產品 } i \\ 0, \text{使用者 } u \text{ 沒有購買產品 } i \end{cases} \quad (3)$$

最後在預測使用者是否會購買該項產品時,將給予每一組 User-Item pair 一個信心數值(confidence),並給定一個門檻值  $\theta=0.5$ ,當信心數值大於  $\theta$  則預測該使用者會購買該項產品,反之亦然。

## 4.3 啟發式評分

這邊我們結合族群購買偏好、個人購買偏好以及同時購買的商品,這三個較具影響力的要素,設計了一條啟發式評分(Heuristic Scoring)公式,計算某使用者對某項產品可能購買的分數,分數越高表示使用者越有可能購買某項商品,最後選出分數較高的來推薦。計算公式如下。

$$\text{Score}(u, i) = W_1 * f(u, i) + W_2 * p(u, i) + W_3 * h(u, i) \quad (7)$$

$f(u, i)$  表示族群購買偏好、 $p(u, i)$  表示個人購買偏好、 $h(u, i)$  表示商品類別關聯。分別給權重  $W_1$ 、 $W_2$ 、 $W_3$ ,分別為 0.5、0.4 及 0.1,族群購買偏好為第一重要的因素,其次為個人購買偏好,最後是同時購買商品。

## 5. 實驗分析與結果

實驗部份我們將在相同推薦數下,比較上一段介紹的幾種方法。再加上 Baseline 的 User-based 和 Baseline 的 Item-based 方法做比較。

Baseline 的 User-based 是對每位使用者而言,皆推薦最熱門的商品給他,以 Baseline 的 User-based top-5 為例,我們將推薦最熱銷的 5 項商品給所有使用者。Baseline 的 Item-based 則是以 item 為出發點,將各項商品推薦給 top-k 的使用者,以 Baseline 的 Item-based top-50 為例,我們將找出最常購買的 50 位“常客”,將所有商品類別皆推薦給這 50 位使用者。

我們將計算 F-measure 來比較各種方法的效能。計算 Precision (hit rate)、Recall、F-measure 公式如下。

$$\text{Precision}(\text{hit rate}) = \frac{c(\text{rnb})}{c(\text{r})} \quad (8)$$

$$\text{Recall} = \frac{c(\text{rnb})}{c(\text{b})} \quad (9)$$

$$F = \frac{2RP}{(R+P)} \quad (10)$$

其中  $r$  表示我們推薦給使用者的商品,  $b$  表示使用者購買的商品。

首先我們對所有有消費紀錄的使用者來做預測,發現啟發式評分的 F-measure 最高,達到 0.756。接著我們藉由找出較頻繁的使用者與網路爬蟲來增加商品資訊,以改善預測的效能,啟發式評分的 F-measure 依然高於其他方法, F-measure 達到 0.802。最後我們考慮時間這個要素,拆分訓練與測試資料,不過所有方法在這個部分的效能都下降。

### 5.1 保留部分 User-Item pairs 做測試

首先我們對於有消費紀錄的 8,235 位使用者與 144 類商品類別做實驗,比較在相同推薦數下,不同方法之效能,如下圖 4。

由圖 4,我們得知 Heuristic Scoring 的效能明顯優於其他所有的方法,在推薦數少的時候, Heuristic Scoring 的 F-measure 達到 0.756,大約是其他方法的 2 倍。基於 CCAM 演算法的最高為 0.402。另外我們發現基於 CCAM 演算法的 User-based 與 User-based CF 效能幾乎相同, F-

measure 最多只差 0.001 的而已。我們認為多數方法預測較差的原因是因為資料中，大多數使用者都只有 1 次的消費行為，對於這些人很難有好的預測。所以接下來我們選出購買較頻繁的使用者來做實驗，以提高實驗效能。

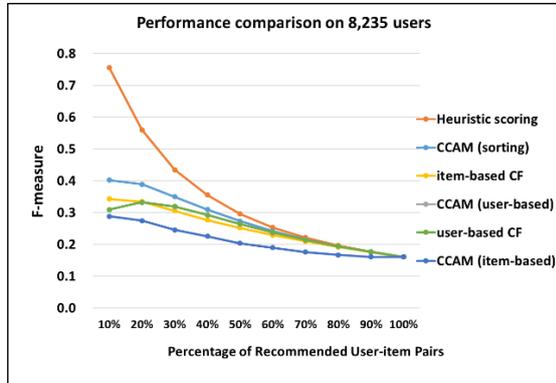


圖 4. Comparison of Performance on 8,235 users

接著我們選出消費次數多於 20 次的 1,715 位使用者來分析，有了較充分的購買紀錄後，從圖 5 中，我們發現大多數的方法效能皆提升不少，Heuristic Scoring 方法的效能還是最佳的，F-measure 最高達到 0.802。而其他方法也由原本 0.3 左右的 F-measure 提升至 0.5~0.6。基於 CCAM 演算法的 User-based 與 User-based CF 效能還是非常相近，他們的 F-measure 最高也將近 0.6。圖中我們還得知，在 User-based 的效能會比 Item-based 的效能好。

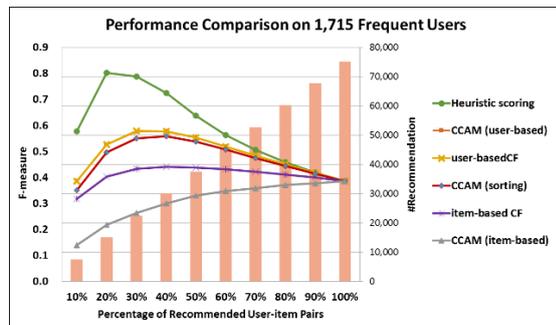


圖 5. Performance Comparison on 1,715 Frequent Users

上圖中每條 Bar 代表對於每位使用者的推薦數，對使用者而言，一次接收到太多的訊息是很負擔的，這樣的推薦效果會比較差，若我們將這些推薦資訊拆分為幾次來推薦，不僅不會造成使用者的負擔，而且還可以增加消費者的來店率，進而提升消費的可能性。我們嘗試調整 CCAM 的使用者群數，改為和 Heuristic Scoring 中族群購買偏好一樣的 18 群，實驗結果與原本分為 3 群的效能大致相同。

在看到這一系列的效能表現後，我們也希望以其他方式看能否增進效能。我們準備兩組數據，一組為只有使用者特徵和產品特徵，另一組 expanded feature 則是擴充如上述從 google 關鍵字過濾的 feature，最後結果呈現如下圖 6，在 libSVM 預設門檻值為 0.5 的前提下，可以看到 expanded feature 的效能優於 feature 效能，可以想見 expanded feature 其實還是能在增進效能這一塊達到實質上的幫助。

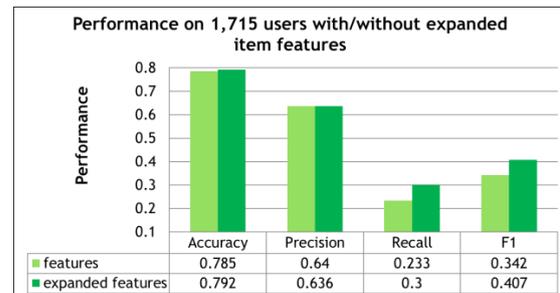


圖 6. Performance on 1,715 users with/without expanded item feature

最後，以 1,715 位使用者為基底，我們準備不同的特徵集，並且算出在相同門檻值下，四個特徵集會有什麼不同的表現。FS1 代表的是 50 個使用者特徵加上 1,171 個產品特徵，FS2 為 FS1 再加上 4 個來自 Heuristic Scoring 的分數，FS3 為僅來自 4 個 Heuristic Scoring 的分數，FS4 則是將 FS1 跑完 libSVM 得到的信心值直接取出，做為第 1 個特徵值，再加上 4 個 Heuristic Scoring 的分數，共 5 個特徵值。

在圖 7 中橫軸所代表的是，門檻值從 1 至 0.1，大於門檻值的我們預測該使用者會購買，並且對 FS1 到 FS4 都算出 10 個門檻值下的 F-measure，可以看到 FS1 的整體表現最為優異，最高的 F-measure 為 0.49，出現在門檻值為 0.8 時。而另外三個特徵值都含有四個 Heuristic Scoring 的分數，呈現出的結果差異性並不大。我們從實驗結果得知，分類器在評斷 User-Item pair 時，給出的信心分數都很高，並且多集中在 0.9 以上，所以門檻值設在 1~0.9 的區段，F-measure 有明顯往下掉的情形，而在這之後的 F-measure 呈現緩慢下降。

最後我們發現 Heuristic Scoring 方法在先前與其他方法比較時表現較佳，但將 Heuristic Scoring 的分數加入分類器後表現的結果不如預期。所以我們認為調配出好的權重，對於 Heuristic Scoring 的表現有很大的影響力。

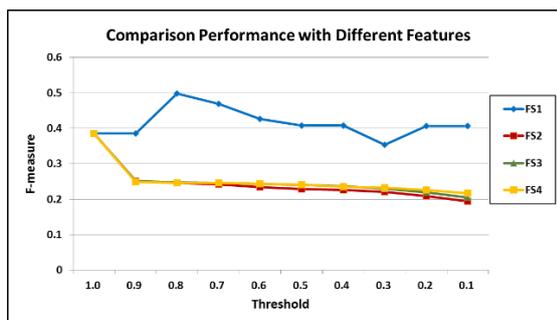


圖 7. Comparison Performance with Different Feature Sets

## 5.2 以時間切分做預測

另外我們認為 Heuristic scoring 效能較好，是因為有考慮近一年的消費習慣，時間這個因素對消費預測有著一定的影響力。所以我們以時間拆分資料，希望能以過去的資料來建 model，來預測下一期的購買情況。選擇以過去 1 年多的資料作為訓練資料，來預測最近 2 個月的購買結果，預測的效能如圖 8。

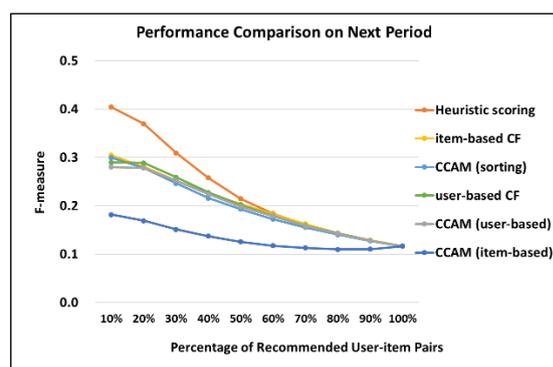


圖 8. 預測下期的效能比較

我們發現所有方法的 F-measure 都比之前的降低了，Heuristic Scoring 方法的效能仍然還是最佳的，CCAM 的 User-based 的效能與 User-based CF 的效能及 Item-based CF 的效能相近。

在實務的推薦上，我們不可能從資料中拆出某些使用者與商品做預測。我們必須仰賴過去的購買資訊對未來做預測。雖然目前實作上的效能不如以常客之完整購買紀錄與增加產品特徵效能做的好，這是我們接下來的挑戰，未來我們將加入時間的資訊考慮以循序樣式探勘 (Sequential Pattern Mining)，來改進對下一期的預測。

## 6. 結論與未來工作

實驗結果發現，不管是使用者消費紀錄或是商品的資訊，得到的資訊越充分，對我們的預測會越有幫助，所以我們藉由網路爬蟲，以獲得更多的商品資訊，並對經常消費的使用者，

利用這些充分得消費紀錄，以達到更好的預測效果。另外由實驗結果，我們得知利用 User-based 的推薦，效能會好於 Item-based 的推薦。

未來我們將會針對消費者購買的地點，進一步分析，找出區域性的消費方式與習慣來改善我們的推薦系統。另外我們將由商品的週期性與順序性來加強對商品的分析，商品的週期性像是衛生棉等週期性產品，若在下個購買時間前推薦給他，可以提升購買率。並且利用 sequential pattern mining 分析產品的順序性來改善推薦的效能。

## 致謝

感謝經濟部補助財團法人資訊工業策進會「105 年度服務系統體系驅動新興事業研發計畫(1/4)」以協助本研究的進行。

## 參考文獻

- [1] M.-L. Wu, C.-H. Chang, and R.-Z. Liu, "Co-clustering with augmented matrix", *Appl. Intell.*, vol. 39(1), pp. 153–164, 2013.
- [2] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendation," In: *Proc. 18th National Conf. Artificial Intell.*, pp. 187–192, 2002.
- [3] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender system," *The Adaptive Web*, LNCS 4321, pp. 291–324, 2007.
- [4] J. Delgado and N. Ishii, "Memory-based weighted-majority prediction for recommender systems," In: *ACM SIGIR'00 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.
- [5] D. Billsus and M. Pazzani, "Learning collaborative information filters," In: *Int. Conf. Machine Learning*, Morgan Kaufmann Publishers, 1998.
- [6] A. Rajaraman and J. D. Ullman, "Data mining," *Mining of Massive Datasets*, pp. 1–17, 2011.
- [7] R. Mihalcea and P. Tarau. "TextRank: Bringing order into texts," *Assoc. for Computational Linguistics*, 2004.
- [8] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2(27), pp. 1–27, 2011.