

一個藉由改善稀疏矩陣處理方式之協同過濾式推薦

*李麗華
朝陽科技大學
資訊管理系
教授

lhli@cyut.edu.tw

李富民
朝陽科技大學
資訊管理系
副教授

fmlee@cyut.edu.tw

陳柏良
朝陽科技大學
資訊管理系
研究生

s10214616@gm.cyut.edu.tw

鄭詠元
朝陽科技大學
資訊管理系
研究生

s10214615@gm.cyut.edu.tw

摘要

現在網際網路科技的發達，網路資料量日益增多，使得人們逐漸無法處理過於龐大的資料，也因此出現了資訊過載 (Information Overloading) 的議題。當資料量變得龐大時，不只會帶來資訊過載的問題，在資料儲存上也常會出現因使用者數目龐大、資料項目過多而造成稀疏矩陣，例如：擁有成千上百產品或數以千計書籍的網路書城，其使用者交易的書目通常為個位數，因此“使用者”在數以千計的“產品項目”項下，只能針對少數幾項產品有交易資訊，如此就會形成資料儲存時的稀疏性。

過去有許多學者提到稀疏性問題，然而在協同過濾式推薦研究上，則較少有學者針對此一議題提出有時效性之解決方法。有鑑於此，本研究提出一個藉由改善稀疏矩陣處理方式之協同過濾式推薦，提升處理稀疏矩陣在運算上的時效性。本研究所提之方法較傳統常用的皮爾森相關係數，在時效上與準確度上都有較佳的結果。

關鍵詞：推薦系統、協同過濾、稀疏性

Abstract

As the development and usage of Internet keeps growing, the sheer volume of data on the Internet makes it difficult for user to handle. As a result, users must deal with the information overloading problem. Even when the amount of data keeps booming such as hundreds of products, thousands of books, however, the amount of transactions for each product item will still be similar. Therefore, the transaction data among thousands of items and thousands of uses may form the sparse matrix. For collaborative filtering

recommendation, there are many researchers proposing great methodologies to handle user-item-matrix dataset, however, not many researchers deal with the performance of sparse matrix problem. This is the reason why this research proposes a novel solution for collaborative filtering recommendation. The proposed method has compared to the traditional Pearson Correlation Coefficient method and the results show that our proposed method is better in terms of time efficient and precision.

Keywords: Recommender System, Collaborative Filtering, Sparsity Matrix

1. 導論

自從二十一世紀起人類即邁向網際網路時代[35]，由於科技的發展讓網際網路的交易與流量蒸蒸日上[36]，現今社會各式各樣的事務越來越依靠網際網路，讓網際網路成為不可或缺的生活形態。由於人們能夠在網際網路上獲得的資料越來越多，因此讓網路使用者愈來愈難去處理這麼多又複雜的資訊量，這種資訊不斷蓬勃成長、資訊過多難以處理的現象，我們就叫它為資訊過載 (Information Overloading) 問題[26]。

為協助解決資訊過載的問題，因此必須依靠許多工具來協助過濾或幫忙篩選，也因此有越來越多的網站開始運用推薦系統 (Recommendation Systems) 來協助資訊的篩選和查找。目前許多購物網站如 Amazon.com、影音網站如 CDNow、手機 APP 下載商店如 PLAY 等網站，也都有提供推薦系統讓使用者來快速找到他們想要的資訊。

資訊過載的問題在現今的網際網路上可說無處不在，例如：若我們開啟任一種瀏覽器進行搜尋時，其搜尋的結果都呈現出數以百萬計

的條目，導致使用者通常只選擇前幾頁的條目查看，且往往會失去耐心繼續查看更多內容。在這種情況下，使用者將可能找不到自己真正需要的資訊，而這樣的問題也是資訊過載常見到的問題。

為解決或協助減少資訊過載這樣的問題，目前許多網站會運用推薦系統或使用者評分機制來幫助使用者進行更多內容的選擇。以電子商務為例，推薦系統就被用來協助使用者找到其過去曾購買、曾點選或與其偏好相關的產品，運用推薦系統可以減少使用者在龐大的資料與商品中搜尋的時間，也可以協助使用者較準確的找到自己想要的資訊或商品。因此推薦系統在電子商務網站中，具有協助使用者更快找到自己想要的產品的功能，相對的也可以讓商家更快銷售出商品，因此推薦系統的建置逐漸成為電子商務網站的重要功能之一。

現今電子商務網站上都已建置提供使用者對交易或產品的評分、評價或評論，如果使用者數量不斷增、同時產品的項目也不斷增加時，這些日積月累的龐大資料，在儲存使用者對產品購買的交易紀錄或評價紀錄時，經常會形成稀疏矩陣，也就是說，使用者購買其需求的產品項目通常是少數幾項，登錄到數以千計的產品項目中僅會有這些少數項目有購買紀錄，因此對每一位個使用者而言，其產品購買後的紀錄以及“產品 vs. 評分”(item vs. ranking)資料，通常也會是一個稀疏矩陣。

學者 Ziqi Lin[20]的研究曾提到，由於稀疏矩陣的資料量分散，會導致在相似度上的計算不易找到相似的使用者，因此如何合宜的處理稀疏性的資料(資訊)來獲取較精確的推薦內容，對協同過濾式推薦(Collaborative Filtering Recommendation)而言是一個很重要的課題。學者 Hao Ji[13]的研究則提到稀疏性資料的處理是一個很大的挑戰，因此他的研究提出一個結合 item-based 與 user-based 的協同過濾式方法來解決此一問題。另有學者 Peng-yu Lu[19]在其研究中提到，稀疏性的資料對傳統的協同過濾式推薦是一個不易處理的議題，此學者提出了混合式的方法並結合協同過濾式與上下文的方法來試圖解決這個問題。

如上所述，協同過濾式推薦的方法可能會碰到稀疏性的資料，而且在資料處理和尋找或計算相似性資料，通常是耗時又不易準確推薦，因此本研究提出一個藉由改善稀疏矩陣處理

方式之協同過濾式推薦研究，本研究之目標如下：

1. 本研究將針對協同過濾式推薦提出一個能有效處理稀疏性矩陣的方法以及針對此稀疏矩陣的相似度比較之計算方法。
2. 傳統的協同過濾式經常運用皮爾森相關係數來進行相似度計算，因此本研究所提之方法將與皮爾森相關係數進行比較，將針對計算的時效性以及推薦的準確性進行比較。

2. 文獻探討

本章節介紹相關的文獻回顧，分為三個部份來進行說明。第一部份將針對推薦系統的相關文獻進行回顧與介紹。推薦系統中較常採用的三種方法是：(1)內容導向式推薦，這是運用使用者過去已交易或購買的紀錄來進行分析並推薦；(2)協同過濾式推薦，這是將所有使用者的歷史資料及個人偏好資訊加以分析，藉由分群的各類計算方法，找出與使用者相似的群組，再依據相似群組的內容來進行推薦；(3)混合式推薦，它結合了上述兩種或者以上的推薦系統。第二部份則將介紹傳統計算相似度的皮爾森係數計算方法，第三部份則介紹稀疏矩陣的特性與簡介。

2.1 推薦系統

推薦系統的技術始於 1990 年[29]，這個方法讓使用者可以藉由推薦系統更容易找到自己需要又或偏好的產品與服務[25]。目前推薦系統的使用在網際網路及電子商務上已經相當廣泛，目前我們生活中常用到的網站如 Amazon、CDNOW、eBay 等都有提供推薦系統[21]來協助客戶找到自己偏好的內容。

推薦系統過去已有許多相關研究[17][27][28][15]，過去 Lin 學者[17]就將社交網路的信任關係應用在網路上的推薦中，藉由運用社群好友的信任度與不信任度找出他們的關係並進行推薦。Schlick 學者[27]將推薦運用在旅遊之中，其運用社群中所有人的經驗分享來進行自動產品推薦。Ye Tao 學者[28]則將推薦應用在行動廣告上，藉此賺取利潤；Kuppili 學者[15]則以評論的字詞分析其中的感情並應用在推薦之中。由上面的介紹可以看出推薦的應用層面相當廣泛。

有關推薦系統的達成與實現有很多方法，根據推薦的方法大致可分為：(1)內容導向式

(Content-based, CB)、(2)協同過濾式 (Collaborative Filtering, CF)以及(3)混合式 (Hybrid) [1]的推薦系統(Recommender Systems)。

內容導向式推薦[5][6][12][23]是依照使用者過去的歷史紀錄做為學習的參考，選擇使用者較為相似的資訊偏好來進行推薦，例如：今天使用者 A 他特別喜歡針對手機、電腦等 3C 產品評分，則推薦會運用他經常購買及評分的手機品牌或者是電腦等周邊所出的新系列推薦給他。因為內容導向式的推薦方法不需要其他使用者的資料，它是依據目標使用者自己本身的偏好或歷史資料，因此能推薦目標使用者所偏好的产品類型或此類型的新產品，但相對的此方法則無法找出目標使用者的其它潛在偏好，因此推薦的內容是往往與過去雷同而有所限制。

協同過濾式推薦[9]則是先找出與使用者興趣較相同的使用者群組，藉由相似使用者群組的交易或購買紀錄來進行協同過濾式推薦；即依據目標使用者相類似的使用者群的經驗，推薦給目標使用者。過去的協同過濾式研究方法中，主要都以改善協同過濾式的準確率和計算效率為主[31][18][34]。另外也有學者提出混合其它方法的協同過濾推薦，例如前面曾提到的運用社群網路的關係來進行推薦[7]。

由於上述所提的二種推薦方法都有它的優缺點，特別是每個方法都有它不足之處，因此也有學者提出混合式推薦方法，藉此來互相彌補上述二種推薦方法的缺點。例如：Kavinkuma 學者[14]提出混合使用者以及 Feedback 裡的社群評論來應用於推薦。混合式推薦系統因為使用不同的方法來互相彌補上述二種推薦方法不足之處，所顧及到的面向較多，整體而言也較能達成推薦的廣度。

有關混合式推薦的分類亦有四種[8]:

- 1.使用內容導向式推薦和使用協同過濾式推薦，然後結合內容導向式推薦與協同過濾式推薦的結果。
- 2.將內容導向式推薦的部分特性結合至協同過濾式推薦之中。
- 3.將協同過濾式推薦的部分特性結合至內容導向式推薦之中。
- 4.混合多種特性所建構出的方法。

現在由於購物網站的內容項目繁多且種類繁雜，使用者尋找商品時，也逐漸感受到過多

的商品或資訊可選，以 Amazon.com 的商品為例，其銷售物高達百萬項目可選，因此電子商務網站亦設計有推薦系統讓使用者透過推薦的功能來篩選喜好的产品或服務項目，加快使用者找到自己想要的產品。總結來說，推薦系統的種類很多也可透過不同的方法來完成並應用在不同的企業上[30]。

2.2 協同過濾式推薦

協同過濾式推薦是目前較常見的推薦方式[16]。人們通常都會與自己擁有相同興趣的朋友或家人分享自己的興趣，這樣的行為往往會讓所有對這件有興趣的群組成員衍生出更多的討論和深度意見交換，同時也會觸發使用者想去購買類似或相同的產品，目前網路上有許多的專業討論社群，都是透過這樣的概念來吸引有共同偏好的人，一起在社群網站上分享或推薦自己的看法，而運用相同偏好群組的意見來進行推薦就是協同過濾式推薦的精神。

協同過濾式推薦是根據目標使用者過去的購買資訊或其個人的偏好資訊，據此來找出與此興趣偏好相同的使用者群組，即所謂的偏好相似群組，接著再利用偏好相似群組所感興趣的內容或產品，推薦給目標使用者，藉由相似群組的喜好特性當作推薦的依據[10][11]。

舉例來說，若有人想上網買手機，若他只想找某幾個品牌的手機，但不知道什麼型號或哪些功能機型較好，這時如果有協同過濾式推薦功能，則當他查詢手機品牌時，就會出現特定的手機型號或功能推薦給他/她。這些推薦往往是此網站系統依據使用者曾購買的紀錄做為分析依據，在系統資料庫裡尋找與這位使用者購買紀錄相似的使用者群組，藉由相似使用者群組過去偏好的手機購買資訊或評價資訊，將此群組偏好的機型/型號推薦給這位目標使用者(Target User)，所以使用者可以很快參考到與他偏好相同者都曾買過什麼產品。

傳統常見的協同過濾式(CF)推薦方法可簡要的分成三個步驟[24][1]:

- 1.輸入資料編碼整理 (Representation of Input Data):將使用者的購買或評分資料進行預處理，以便於建置使用者個人偏好資料庫，使用者的評分資料亦通常採用”使用者-項目矩陣”(User-item Matrix, UM)來進行。
- 2.尋找偏好相似之鄰居群 (Neighbour Formation):運用相似度量測(Similarity

Measure)尋找喜好相近的使用者，然後將相似使用者建立鄰居群(Neighborhood)，這是協同過濾式推薦最關鍵的部分。另外，為能夠提升推薦品質，避免相似度低的使用者影響推薦結果的準確性，Herlocker 等學者[9]提出保持良好的預測準確度及預測效能，可以透過限制鄰居群範圍來做到。該研究[9]表示，鄰居群若擁有高的相似度則會提高預測結果的價值性[9]。

3. 鄰居群範圍的決定通常有兩種方法：(a)最近相鄰法 (k-Nearest Neighbour, k-NN) [9][22][25] 及 (b) 門檻值相似度 (Similarity Threshold)[22]。

4. 產生推薦 (Generation of recommendation)：依據鄰居群資料找出 Top-N 排序項目做為推薦給目標使用者的項目。

較常見的推薦產生方法約有兩種：(1)最頻繁項目 (Most-frequent Item) 的推薦，由相似群組 (Similar User Group) 中找出目標使用者最頻繁被購買且目標使用者尚未購買過的項目，放入推薦清單 (Recommendation List, RL) 中；(2)關聯法則 (Association Rule-based) 推薦，依據目標使用者所買過的項目，運用項目關聯法則，把具有最高信心值 (Confidence Value) 且目標使用者尚未購買的項目，放入推薦清單中。

儘管協同過濾式推薦的相關研究顯示其預測成果不錯，但協同過濾式推薦的成效仍會受到使用者群組的影響，即如果使用者相似群組內的成員其相似度不夠高，則預測目標使用者的偏好時，將可能有所偏差。所以，如何找出與目標使用者相似的群組，是協同過濾式推薦方法中十分重要的關鍵。此外，協同過濾式推薦在處理使用者購買或偏好資訊時，亦可能遭遇下列問題[1]：

1. 稀疏性：即使用者的交易資料、評分量不足時，容易產生矩陣的稀疏性。
2. 延展性：當資料不斷成長變得龐大時，則可能影響到推薦的準確性。
3. 冷起始：系統剛開始啟用時，使用者交易量太少而有資料不足的情形，亦可能導致無法進行推薦。
4. 語意理解性：協同過濾式推薦的方法，並沒有特別針對語意內容加以理解。

2.3 皮爾森相關係數

在傳統的協同過濾式推薦方法中，皮爾森相關係數 (Pearson's Correlation Coefficient, PCC) 了 [5][25] 是最常被拿來做為使用者相似度的計算方法。皮爾森相關係數是將兩個使用者的評分相關性做比較，PCC 的計算公式如下面公式(1)。

$$pcc_sim(u_a, u_i) = \frac{\sum_{j=1}^m (r_{a,j} - \bar{r}_a)(r_{i,j} - \bar{r}_i)}{\sqrt{\sum_{j=1}^m (r_{a,j} - \bar{r}_a)^2} \sqrt{\sum_{j=1}^m (r_{i,j} - \bar{r}_i)^2}} \quad (1)$$

在公式(1)的皮爾森相關係數中， $\bar{r}_a$ 代表使用者 u_a 對所有產品評分 (rating) 的平均數， $\bar{r}_i$ 代表目標使用者 u_i 對其所有買過的產品之評分平均數， j 是兩位使用者 u_a 和 u_i 他們對產品 j 有共同的評分項目， m 則是兩位使用者擁有共同評分產品項目的數量，而 $pcc_sim(u_a, u_i)$ 則是指使用者 u_a 和 u_i 的相關程度。

運用皮爾森相關係數方法找出的值其範圍介於 1 到 -1 之間，若值是在 0 到 1 之間的話代表此兩個使用者有正向的關係，即表示兩個使用者是相似的，但若值在 0 到 -1 之間的話，則表示找到的相關性是負向關係，表示兩個使用者是不相似的。

皮爾森相關係數方法亦有其計算的問題：

1. 資料稀疏性 (Sparsity)：

使用者評分資料太過分散，則可能導致皮爾森找不到 u_a 和 u_i 的共同的評分項，例如：以下面表 2.1 做簡單的示意，如此則無法進行計算，所以無法找出相似使用者。

表 2.1 資料稀疏性示意圖

產品 使用者	I_1	I_2	I_3
C_1	5		
C_2		3	4

2. 使用者至少須對 2 個產品以上有評分紀錄 (參見表 2.2)，否則無法進行平均數與差異性的計算。

表 2.2 須有 2 個以上評分項示意圖

產品 使用者	I_1	I_2	I_3
C_1	5	4	
C_2	3	2	

3. 若使用者評分紀錄相同，則公式無法計算：因評分分數若都相同，分母所有項目相減為

0，則會造成計算錯誤。(參見表 2.3 示意)

表 2.3 評分相同示意

產品 使用者	I_1	I_2	I_3
C_1	5	4	
C_2	3	2	

4. 不同評分但平均值相同時，無法判別差異：如表格 2.4 之示意，如果 2 個使用者 C_1 和 C_2 對產品的評分並不同，但因平均值相同，則以皮爾森相似係數來計算會找到相同的答案。如此會誤會 C_1 和 C_2 具有相似性。

表 2.4 評分不同但卻算出相似

產品 使用者	I_1	I_2	I_3
C_1	5	4	3
C_2	3	4	5

2.4 稀疏性與稀疏矩陣

由網際網路的發展及電子商務的普及，客戶和產品項目的數量通常會不斷增加，因而使資料量日益龐大。在這種情況下所儲存的使用者資料、產品資料及使用對產品的評分資料矩陣，其內容可能會變得很稀疏，即我們前面所提到的稀疏性問題。

以商務網站如亞馬遜(Amazon.com)為例，數以萬計的產品項目往往其分類就令人眼花撩亂，加上全球數以百萬的使用者會員，這是網路普及後的常見現象。學者曹一鳴[4]則指出，現在的使用者與產品規模日益擴增，成千上萬的使用者及多如牛毛的產品，導致使用者間購買產品項目越來越不易重疊。如此一來，難免會形成資料矩陣的稀疏性(如圖表 2.1)，這也往往會影響協同過濾式的推薦成效[2][3]。

	I_1	I_2	I_3	I_4	I_5	I_6	I_7		I_j
C_1							3		
C_2		4							
C_3					5			...	
C_4									
C_5			2						
C_i	5					1			

圖 2.1 稀疏矩陣示意圖

3. 研究架構

本研究的目的旨在提出一個可以協助處理稀疏矩陣且可以提升協同過濾式推薦的計算效能與推薦準確率的方法。本研究以電子商務網站為對象，針對使用者對產品的評分資料來加以研究分析，研究內容以加快使用者之相似性比對計算，藉此找出相似偏好的群組為主要之研究方法。本研究所提之方法說明如後，研究步驟請參見圖 3.1。

3.1 研究架構

本研究所之方法共分四個階段(如圖 3.1)。第一個階段是先到商務網站取得使用者對產品的評分資料，並針對資料內容進行資料整理，同時將資料內容有誤或資料缺漏的部份做清理動作；第二階段則是將已整理好的資料進行資料轉換，亦即將使用者的評分資料轉換成所謂的二元偏好值。所謂的二元偏好轉換是指若評分值大於平均值以上，則視為使用者對該產品具有偏好，因此將此偏好值轉換為 1，其它低於平均值的則轉換為 0，另將原本僅有使用者評分資料的集合，轉換成使用者對產品的 user-item 矩陣；第三階段則是將目標使用者的資料和轉換過的 user-item 矩陣中的每一位使用者進行資料比對，找出和目標使用者較相似者，將這些較相似的使用者視為相似群組。

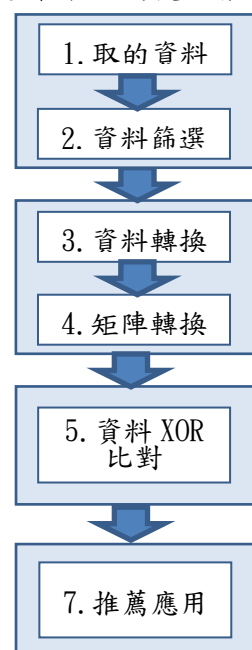


圖 3.1 本研究的架構圖

為了加速比對的速度，本研究運用 XOR 方法進行快速的相似度比對，詳細做法將在下面

進行說明。運用 XOR 比對後，所有比對後的資料再經反向值轉換，即相似者為 1、不相似者為 0 的轉換輸出；第四階段則是依據第三階所找出的相似群組與目標使用者的相關偏好，做為推薦給目標使用者的依據，並給予目標使用者相關產品的推薦。以下針各階段進行說明。

階段一：

本研究運用的資料是以大部份網站都有提供的產品購買後評分資料做為研究內容，例如：產品滿意度評分、星等評分、推薦值等。首先本研究由網站取得使用者對產品的相關評分資料，此資料集本研究命名為 Customer Rating File(CRF)，CRF 資料集包含使用者的編號集合(C)、產品編號集合(I)、使用者對產品的評分值集合(R)，即 $CRF=\{C,I,R\}$ ，而個別的資料集內容分別為 $C=\{c_1,c_2,c_3,...,c_i\}$ ， $I=\{i_1,i_2,i_3,...,i_j\}$ ， $R=\{r_1,r_2,r_3,...,r_k\}$ ， $\forall i,j,k \in N$ 。藉由網路上所蒐集到的這些資料來進行本研究的實驗與分析。

接下來，針對上面的 C 、 I 、 R 資料集進行資料篩選。本研究定義產品 i_j 的評分值 r_k 是由使用者 c_i 對產品 i_j 所產生的回饋值。以下圖為例，假設有 5 個使用者針對多項產品 i_j 回饋其評分值，以五等第的評分值為例如圖 3.2。

c_i	i_j	r_k
c_1	i_3	5
c_1	i_8	2
c_2	i_1	4
c_3	i_3	5
c_3	i_4	1
c_4	i_3	3
c_4	i_9	4
c_4	i_1	5
c_5	i_{10}	2

圖 3.2 使用者評分紀錄範例

由於商務網站上的產品項目可能數以萬計，如：Amazon.com 或博克萊網站 (<http://www.books.com.tw>)，很顯然的每位使用者不可能都購買過所有的產品，甚至可以說，每位使用者曾購買過的產品極其有限，因此使用者能回饋的產品項目評分也很有限，甚至稀疏。誠如 Junrui Yang[32]所提，協同過濾式的方法會有稀疏性(Sparsity)的問題，則如果使用者評分資料的矩陣越稀疏，使用者之間的相似

性就會越小。因此為能合宜的處理資料矩陣稀疏性問題，本研究把所有使用者對產品項目沒有評分過的項目先預設為”NA”(Not Available)。然後針對每一項產品，設 IRC (Item Rating Count)為產品曾被評分的次數總計，若 IRC 低於門檻值(Threshold value)則先將該產品項目刪除，這是因為評分項目過少的產品，仍屬於冷起始(cold start)[33]狀態，也就是說該產品項目尚未被足夠的使用者購買，因此不建議用來做為推薦分析的項目。而這類產品將來可採用其它的模式來進行促銷，例如：”買 A 送 B”、”紅標商品搭配綠標商品”、”促銷折扣商品”等方法來加以推薦。由於本研究希望針對較常被購買的項目來做推薦研究，因此評分項目低於 IRC 的項目(item)，將先由資料集(I)中清除。

同樣的，所蒐集到的資料中，如有新進的使用者僅買過 1 次或少數幾次產品，如其產品購買項目低於 IP (Item Purchased)值，則本研究亦會先將該類使用者自 CRF 資料集中刪除，因為這類使用者其購買項目太少，仍不足以用來做為推薦的依據。

第一階段我們將 CRF 資料集先做好上述的資料整理與篩選，藉此精簡欲推薦分析的資料內容，同時也可以提升下個階段的資料處理速度及計算效能。

階段二：

在第二階段中，針對上述已整理完的 CRF 資料進行資料轉換。若以五等第的產品評分回饋意見來看，若使用者對產品的評分為 4~5 分即高於平均值 3 分，則代表使用者對於產品是屬於滿意且有正向評價，因此我們將此範圍的分數轉換為 1，代表使用者對此產品有偏好；相反的，若使用者對產品的評分為 1 分~2 分之間，則代表使用者對產品的評價較不滿意，則我們將此分數轉換為 0，即代表對此產品沒有偏好；而 3 分代表使用者對產品屬於沒意見或中庸看法，但這也表示沒有特別的喜好，因此對於偏好推薦來看，這類沒有特別的喜好评價，應該不具備有推薦給它人的強烈因素，因此亦不適合做為推薦的依據，所以本研究將 3 分的評價值亦轉換為 0 分。下面圖 3.3 即轉換圖 3.2 之後的結果。

c_i	i_j	r_k
c_1	i_3	1
c_1	i_8	0
c_2	i_1	1
c_3	i_3	1
c_3	i_4	0
c_4	i_3	0
c_4	i_9	1
c_4	i_1	1
c_5	i_{10}	0

圖 3.3 分數轉換示意圖

當評分資料轉換成偏好二元資料後，接下來則進行矩陣的轉換。由於本研究提出以 XOR 進行快速的偏好邏輯式比對，因此會先將原始偏好資料轉換成 user-item 矩陣，並將沒有任何評價資料的地方填入 NA(Not Available)值。轉換後的 user-item matrix 請參見下圖。

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
c_1	NA	NA	1	0	1	NA	NA	0	1
c_2	1	0	1	NA	NA	0	1	0	NA
c_3	1	1	1	0	NA	0	0	NA	NA
c_4	NA	0	0	1	0	1	1	1	NA
c_5	0	NA	NA	1	1	NA	0	NA	0

圖 3.4 user-item 矩陣示意圖

由於電子商務網站的商品及使用者數量很多，經資料整理後相關的使用者編號或產品編號亦可能為非連續號(即不斷跳號)，如此會造成矩陣呈現過多的無效欄位，因此對資料集的使用者編號(c_i)及產品編號(i_j)進行重新編號並刪除矩陣中所有無效欄位。

階段三：

由 CRF 矩陣轉換成 user-item 矩陣完成後，即運用 user-item 矩陣來進行偏好邏輯性比對，本研究運用 XOR 比對計算，假設目標使用者是 c_1 則比對的使用者則為 c_2 、 c_3 、……直到全部的使用者都比對完為止(參見圖 3.5)。每一次的比對我們將比對結果儲存在偏好相似度矩陣(Preference Similarity Matrix)中。

階段四：

獲得偏好相似度矩陣(Preference Similarity Matrix)後，此矩陣可供我們篩選出

與目標使用者相似的群組。在偏好相似矩陣中有三種數值 1(True)、0(False)、NA，經上面步驟運用 XOR 比對完後，每一個資料格中的數值代若為 1 則代表二個使用者對該產品偏好不同，若為 0(False)則代表偏好相同、若為 NA 則代表目標使用者和比對的使用者他們二者沒有共同購買此產品的評價資料。

如果欲明確表達偏好相似性，我們可將 True 和 False 兩個值互換，即將上述的 False 值(偏好相似)轉換為 1，反之 True 值(偏好不同)轉換為 0。若為了減少轉換的時間，加速計算步驟，則可省略此步驟。

為明確表達計算內容，在此採用偏好相似轉換為 1，偏好不同轉換為 0 的方式來加以說明。若 c_i 對 c_i 所有產品評價偏好值有 1(對產品 i 有共同偏好)的部份進行加總，則可求得 c_i 和 c_i 二位使用者的共同偏好產品數量。在進行推薦時，可以將計算加總後屬於 TOP-N 或大於門檻值 α 的這些 user 視為目標使用者的相似偏好群組，接下來即可將相似偏好群組的產品購買資訊提供給目標使用者，進行推薦。



	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
c_1	NA	NA	1	0	1	NA	NA	0	1
c_2	1	0	1	NA	NA	0	1	0	NA
c_3	1	1	1	0	NA	0	0	NA	NA
c_4	NA	0	0	1	0	1	1	1	NA
c_5	0	NA	NA	1	1	NA	0	NA	0

圖 3.5 XOR 比對示意圖

4. 實驗與結果

本研究提出一個能減輕矩陣稀疏性缺點的偏好相似協同過濾式推薦方法，本研究所提的方法，是以使用者過去的評分紀錄與其他使用者的評分紀錄做為偏好的相似性比對，藉此尋找到相似的群組，並透過相似群組成員的產品偏好來進行推薦。

為了驗證本研究所提的推薦方法具有時間效能與準確率，故本研究將與傳統協同過濾式最常用的皮爾森相關係數方法進行比較，比對的項目，主要是以計算時間及準確率為主，以下介紹本研究的實驗。

基於本研究須針對矩陣進行大量的運算

及資料處理，因此本研究採用 R 語言來進行資料的編碼、資料的清理及資料的比對與運算。以下說明本研究的做法與操作過程。

4.1 實驗工具

本研究採用 R 語言的軟體工具 Rstudio 做為程式設計平台、透過 Microsoft office Excel、MATLAB 來做前置資料處理及相關的矩陣變換。各類軟體工具的功能說明如表 4.1。

表 4.1 本研究之實驗工具及其功能

軟體工具	功能使用
MATLAB	使用者評分資料表製作
Rstudio	程式撰寫、測試及執行
Microsoft office Excel	顯示實驗資料表及實驗資料前置處理

4.2 實驗資料

本研究的實驗資料取自著名的產品評價網 Epinions.com，資料係為 2013 年 11 月和 12 月使用者對於曾購買過的產品項目給予評分的資料，資料集包括使用者(user)、產品(item)、評分(rating)三個欄位及項目(參見圖 4.1 的欄位示範)，而評分資料則是採 5 等第評分值，越接近 5 分代表使用者對於該產品愈感到滿意，相反的愈接近 1 分則代表使用者對該產品越不滿意。由圖 4.1 可看出，資料矩陣包含了使用者編號(c_i)、產品編號(i_j)以及使用者 c_i 對產品 i_j 的評分。本研究共蒐集到 664,824 筆。

Customer	Item	Rating
c_1	i_1	5
c_2	i_2	2
c_3	i_3	3
c_4	i_4	3
c_5	i_5	5
c_6	i_6	2
c_7	i_7	4

圖 4.1 本研究之資料蒐集範例

4.3 篩選資料

針對取得的 664,824 筆資料內容進行資料預處理，即去除被評價次數太少的產品或評分項太少的使用者，因這些資料量太少，

將不足以形成偏好群組。本研究留下產品項目被評分的次數大於 10 次以上，以及曾對 4 項產品以上留下評分內容的使用者，總計篩選出 10,972 筆。這裡面包含 1,723 位使用者以及 278 個產品項目。其稀疏度計算為： $[1-(\text{評分內容}/\text{矩陣容量})]$ ，故本實驗所取得的真實資料其稀疏度大約為 97.8% (即 $97.8\% = 1 - 10,972 / (1,723 * 278)$)。

4.4 資料轉換

接下來針對篩選後的資料集進行資料偏好的轉換及序號重編。由於原本的資料集資料數量有超過 6 萬多筆，篩選後僅剩 1 萬多筆，因此資料編號可能跳號且差距可能很大，因此對資料的使用者代號(c_i)及產品編號(i_j)進行重編。重編後即接著進行評分(rating)偏好的轉換，即將評分值大於 3 的轉換為 1，小於等於 3 的轉換為 0，參見圖 4.2 之範例。

Customer	Item	Rating
c_1	i_1	1
c_2	i_2	0
c_3	i_3	0
c_4	i_4	0
c_5	i_5	1
c_6	i_6	0
c_7	i_7	1

圖 4.2 評分資料轉換為偏好值

4.5 矩陣轉換

做完 rating 資料的 0 與 1 偏好轉換後，接下來將進行 XOR 比對判斷資料的相似度，也就是找出目標使用者和比對的使用者他們對每個產品的偏好評分是否有相似性，為了加速 XOR 的比對，我們須先將資料矩陣透過 R 語言轉換成 user-item 矩陣，如下圖所示，其中沒有資料的部份一律填入 NA。

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
c_1	0	0	1	NA	1	0	NA	NA
c_2	1	0	1	NA	NA	0	NA	NA
c_3	NA	0	1	NA	NA	NA	0	1
c_4	NA	NA	NA	1	0	0	0	1

圖 4.3 User-Item Matix

4.6 資料 XOR 比對

上述建置好的 user-item 矩陣，進行 XOR 相似度偏好比對。這在兩兩使用者比對過程中，若其中一人有 NA 值則表示無法比較，故仍會呈現 NA。若比對時兩個值均為 1 或均為 0，則代表此二個使用者對該產品有相同偏好，XOR 運算後得到 False，若二個值不同則代表偏好不同，XOR 運算後得到 True。這裡的計算方式我們稱之為偏好邏輯性比對，XOR 計算後產生之矩陣範例參見下圖。

	i_1	i_2	i_3	i_4	i_5
C_1 vs. C_2	NA	NA	NA	NA	NA
C_1 vs. C_3	NA	True	NA	NA	True
C_1 vs. C_4	False	NA	NA	NA	NA
C_1 vs. C_5	NA	NA	NA	False	NA
C_1 vs. C_6	False	NA	NA	False	NA
C_1 vs. C_7	True	True	NA	NA	NA
C_1 vs. C_8	NA	NA	NA	True	False

圖 4.4 資料 XOR 比對後範例

當然在此階段，若為了容易理解矩陣內容的相似性與否，亦可將 False/True 值對調。對調後，則 True 代表兩個使用者對該產品有共同偏好，而 False 則代表兩個使用者對該產品沒有共同偏好。

4.7 推薦應用

為了能找出相似使用者，接著我們將圖 3.9 所示的每一個比對橫列找出”相似”性的加總，加總數愈高的表示這使用者與目標使用者的偏好愈接近。由於在前面圖 3.9 所示中，我們已經有了 FALSE 和 TRUE 的值，我們將此邏輯值轉換為數字再進行加總計算即可。相似的加總值高(即相似的數值高)則代表兩個使用者的相似度高。相似度高的使用者將被選取做為協同過濾式推薦的參考群組。

4.8 實驗設計與比較

為將本研究的方法與傳統的皮爾森相似係數計算進行比較，因此本研究進行相似度計算時效上及推薦準確率上的實驗設計。

首先將資料分為訓練資料和測試資料。本研究針對 278 個產品中，先隨機抽出 5 個、10 個、15 個產品等三組不同的實驗資料集合

(命名為”取出 5”、”取出 10”、”取出 15”)來做為推薦預測的測試資料集。被抽出後所剩下的資料集則做為訓練資料集，此資料集將先用來做為尋找相似群的計算時效比較，以及建立相似群組後，用來針對上述三組測試資料做產品推薦預測，也就是針對”取出 5”、”取出 10”、”取出 15”三個不同的測試資料集，找出推薦的準確度。

本研究隨機抽取的三組資料，分別運用本研究的方法及皮爾森相關係數來進行計算並找出相似群組，而實驗的結果則呈現此二種方法的時效性及推薦準確度之實驗結果。

4.9 實驗結果

經本研究之程式實際訓練及測試後，本研究提出的方法與皮爾森相關係數進行比較後，不論在哪一組資料的測試，本研究的方法在時效性上都較皮爾森相關係數快(參見圖 4.5)。

如下圖 4.5 所示，以時間效能來看，本研究在三組不同的測試資料中，在計算的時效性上，均優於皮爾森相關係數方法，而其中在”取出 10”的這一組資料上，計算效能最佳。

在推薦準確度方面，本研究的方法也比皮爾森相關係數為佳，請參見圖 4.6。圖 4.6 中，以隨機取出 10 個產品來做推薦測試的結果來看，本研究推薦的準確率為 0.694，而皮爾森相關係數的準確率為 0.25，因此本研究的方法所找到的相似群組，其所推薦的產品之準確率，較傳統皮爾森相關係數方法為佳。

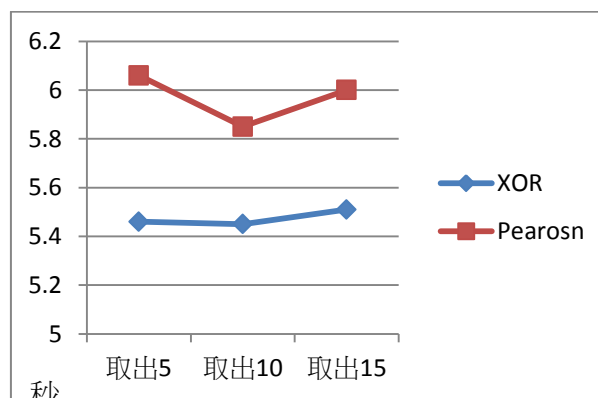


圖 4.5 XOR & 皮爾森相關係數執行時間比較

由上述二個實驗結果可以看出，以現今資料量龐大的網路商品評分資訊來看，皮爾森相

關係數在相似使用者的計算與建立相似使用者群組上，並無法有效的找到合宜的相似群組，因此其預測準確度很可能趨近於 0(參見圖 4.6 的”取出 5”和”取出 10”的準確率)。探究其原因，主要是未評分的 NA 資料太多，讓皮爾森相關係數無法準確的算出每一個使用者評分的趨勢，因而容易誤判。反觀本研究的方法則可以處理資料稀疏時的預測問題。

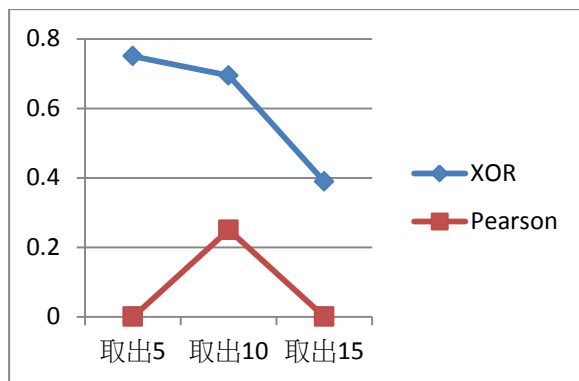


圖 4.6 XOR & 皮爾森相關係數準確率比較

為了更明確的比較本研究與皮爾森相關係數的差異性，本研究依據圖 4.5 和圖 4.5 的實驗結查觀察出，和其它取出 5 筆及 15 筆的情況本研究均顯著的優於皮爾森相關係數的方法，不過也觀察到在隨機取出 10 筆產品的效能上，本研究 and 皮爾森相關係數的實驗結果是較為接近的，因此本研究再針對隨機取出 10 筆產品的資料集繼續延伸其實驗內容。即本研究再針對隨機抽出 10 筆做更進一止的資料測試。在此本研究再做三次的隨機抽出 10 筆，分別命名為 L10-1-1、L10-1-2、L10-1-3 資料集。經實驗結果呈現如圖 4.7 和圖 4.8。

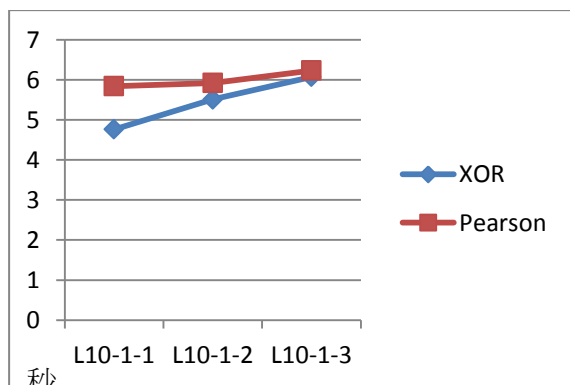


圖 4.7 L10-1 時間比較

由以圖 4.3 和圖 4.4 可看出本研究的方法比皮爾森相關係數不論在計算時間效能上或

產品推薦的準確度上仍然要比皮爾森相關係數為佳。在時間比對上 L10-1-1 比皮爾森快了 1.08 秒、L10-1-2 比皮爾森快了 0.41 秒、L10-1-3 比皮爾森快了 0.16 秒，在準確率方面 L10-1-1 比皮爾森準確度高 33%、L10-1-2 比皮爾森高準確度高 45%、L10-1-3 比皮爾森高出 50%。

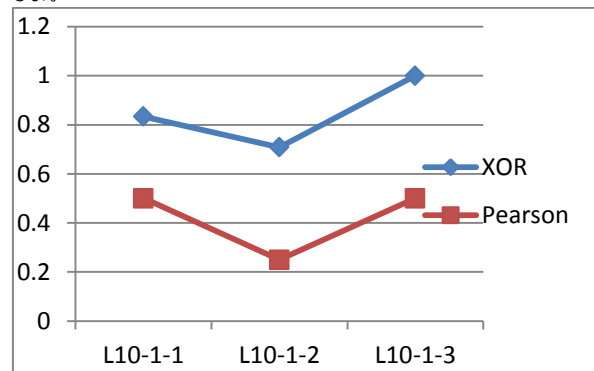


圖 4.8 L10-1 準確率比較

5. 結論與未來研究

隨著現在科技的發達與網際網路的普及，使得人們必須面對更多也更龐大的資料，這使得使用者必須面對資訊過載的問題。目前有許多電子商務網站，提供推薦系統來協助使用者可以找到更準確的資訊與商品。

過去有關推薦系統的相關研究，協同過濾式推薦是最常被運用的一種推薦方法。協同過濾式推薦主要是針對目標使用者過去的歷史資料或其偏好資訊來找出相似使用者群組，接著再依據相似使用者群族的資訊提供給目標使用者進行推薦。

由於目前電子商務網站所提供的產品資訊非常多，例如：Amazon.com 的商品不但分類項目多，商品的種類也很多。因此在蒐集這類使用者資訊時，會實際面臨到每個產品被購買的資料量並非很多，也就是會面臨到資料稀疏的情形。因此在處理這類資料時，如果用傳統的方法來進行協同過濾式推薦，將會耗費許多時間，特別是在進行相似性比對時，會因為使用者數量或者產品數量過多使得運算時間過長。有鑑於此，本研究提出一個藉由改善稀疏矩陣處理方式之協同過濾式推薦，提升協同過濾式建立相似群組的計算效能，增加推薦的準確性。

由本研究所提的方法經實驗結果呈現，本研究與皮爾森相似度分群的比較上，本研究所提之方法在時效性和準確性上都比傳統的皮爾森相似度為佳。

在未來研究上，由於本研究目前僅針對皮爾森相關係數進行比較，因此在未來將再與其它不同的協同過濾式方法進行比較；同時本研究目前所提的方法在計算時間上雖然已比皮爾森相關係數的計算更快，但仍可再優化程式撰寫及矩陣處理技術，特別在針對稀疏矩陣所造成的計算時間加長方面及資料儲存的技術上持續給予優化之研究。

參考文獻

- [1]李麗華、李富民、蔡美慧、廖品妍 (2009), “以偏好為導向的協同過濾推薦”, 2009 年中華商管科技學會年會暨學術研討會, 台灣台南縣, Oct. 23, 2009, pp.52-53。
- [2]曾小波, 魏祖寬, 金在弘, (2010) “協同過濾系統的矩陣稀疏性問題研究,” 計算機應用(Journal of Computer Applications)第 30 卷第 4 期, pp. 1079-1082。
- [3]顧立志, (2014) “協同過濾數據稀疏性問題研究,” 《計算機光盤軟件與應用》2014 年第 8 期, pp. 101-102。
- [4]曹一鸣, (2012), “協同過濾推薦瓶頸問題綜述,” 軟件 2012 第 33 卷第 12 期, pp. 315-321。
- [5]H. J. Ahn, (2008), “A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem,” Information Sciences, Vol. 178, No. 1, pp. 37-51.
- [6]M. Y. H. Al-Shamri, K. K. Bharadwaj, (2008), “Fuzzy-genetic approach to recommender systems based on a novel hybrid user model,” Expert Systems with Applications, Vol. 35, No. 3, pp. 1386-1399.
- [7]Wei Chen, Simon Fong, (2010), “Social Network Collaborative Filtering Framework and Online Trust Factors: a Case Study on Facebook,” The Fifth International Conference on Digital Information Management (ICDIM), pp. 266-273, Alexandria, VA, USA.
- [8]Y. L. Chen, L. C. Cheng, (2008), “A novel collaborative filtering approach for recommending ranked items,” Expert Systems with Applications, Vol. 34, No. 4, pp. 2396-2405.
- [9]J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, (1999), “An algorithmic framework for performing collaborative filtering,” The 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230-237, New York, NY, USA.
- [10]J. Herlocker, J. A. Konstan, and J. Riedl (2002), “An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms,” Information Retrieval, Vol. 5, No. 4, pp. 287-310.
- [11]J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl (2004), “Evaluating collaborative filtering recommender systems,” ACM Transactions on Information Systems, Vol. 22, No. 1, pp. 5-53.
- [12]B. Jeong, J. Lee, and H. Cho, (2009), “An iterative semi-explicit rating method for building collaborative recommender systems,” Expert Systems with Applications, Vol. 36, No. 3, Part 2, pp. 6181-6186.
- [13]Hao Ji, Jinfeng Li, Changrui Ren, Miao He, (2013), “Hybrid Collaborative Filtering Model for improved Recommendation,” Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 142-145, Dongguan, China.
- [14]V. Kavinkumar, Rahul Reddy Rachamalla, Balasubramanian Rohit, M. Sridhar, Sridharan.K, Dr. D.Venkataraman, (2015), “A Hybrid Approach for Recommendation System with Added Feedback Component,” 2015 IEEE, International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 745-752, Kerala Kochi, India.
- [15]Venkatanareshbabu Kuppili, Gayatri Pradip Kudchadker, Ankush Arora, Deepak Kumar, (2015), “Variance based Product Recommendation using Clustering and Sentiment Analysis,” 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI), pp. 1-5, Kanpur, India.
- [16]K. Kwon, J. Cho, and Y. Park, (2009), “Multidimensional credibility model for neighbor selection in collaborative recommendation,” Expert Systems with Applications, Vol. 36, No. 3, Part 2, pp. 7114-7122.

- [17] Chun-Cheng Lin, Tzong-Shyan Lin, Wan-Yu Liu, (2012), "A Trust and Distrust Mechanism for a Social Network-based Recommendation System," The 15th International Symposium on Wireless Personal Multimedia Communications, pp. 172-176, Taipei, Taiwan .
- [18] Jian-ping Liu, Yong Wang, Feng-Hua Yan, (2010), "An Improved Collaborative Filtering Recommendation Algorithm," The 1st International Conference on Networking and Distributed Computing, pp. 194-198, Madrid, Spain.
- [19] Peng-yu Lu, Xiao-xiao WU, De-ning Teng, (2015), "Hybrid Recommendation Algorithm for E-commerce Website," 8th International Symposium on Computational Intelligence and Design, pp. 197-200, Hangzhou, China.
- [20] Ziqi Lin, Wancheng Ni, Haidong Zhang, Meijing Zhao, Yiping Yang, (2015), "A K-medoids Algorithm Based Method to Alleviate the Data Sparsity in Collaborative Filtering," Proceedings of the 34th Chinese Control Conference, pp. 4974-4979, Hangzhou, China.
- [21] Zhang Mu, Chen Yi, Zhang Xiaohong, Lai Junyong, (2009), "Study on the Recommendation Technology for Tourism Information Service," 2009 IEEE, Second International Symposium on Computational Intelligence and Design, pp.410-415.
- [22] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, (2004), "An evaluation of neighbourhood formation on the performance of collaborative filtering," Artificial Intelligence Review, Vol. 21, No. 3, pp. 215-228.
- [23] M. Papagelis, D. Plexousakis, (2005), "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents," Engineering Applications of Artificial Intelligence, Vol. 18, No. 7, pp. 781-789.
- [24] C.H. Piao, J. Zhao, and L. J. Zheng, (2009), "Research on entropy-based collaborative filtering algorithm and personalized recommendation in e-commerce," Service Oriented Computing and Applications, Vol. 3, No. 2, pp. 147-157.
- [25] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, (2000), "Analysis of recommendation algorithms for e-commerce," The 2nd ACM Conference on Electronic Commerce, pp. 158-167, Minneapolis, Minnesota.
- [26] J. B. Schafer, J. Konstan, and J. Riedi (1999), "Recommender systems in e-commerce," Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 158-166, Denver, Colorado, USA.
- [27] Sabine Schlick, Isabella Eigner and Alex Fechner, (2015), "Using Collective Intelligence to Generate Trend-based Travel Recommendations," The 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), pp. 177-185, Lisbon, Portugal.
- [28] Ye Tao, (2015) "Design Of Large Scale Mobile Advertising Recommendation System," The 4th International Conference on Computer Science and Network Technology (ICCSNT), pp. 763-767, Harbin, China.
- [29] Quan Wen, Jianmin He, (2006), "Personalized Recommendation Services Based on Service-Oriented Architecture," IEEE, Asia-Pacific Conference on Services Computing (APSCC'06), pp. 356-361, Guangzhou, Guangzhou, China.
- [30] Li-Tung Weng, Yue Xu, Yuefeng Li, (2005), "A framework for e-commerce oriented recommendation systems," Proceedings of the 2005 International Conference on Active Media Technology, 2005. (AMT 2005), pp. 309-314, Kagawa, Japan.
- [31] Yueping Wu, Jianguo Zheng, (2010), "A Collaborative Filtering Recommendation Algorithm Based on Improved Similarity Measure Method," IEEE International Conference on Progress in Informatics and Computing, pp. 246-249, Shanghai, China.
- [32] Junrui Yang, Cai Yang, Xiaowei Hu, (2016), "A Study of Hybrid Recommendation Algorithm Based On User," 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), pp. 261-264, Hangzhou China.
- [33] Peng Yi, Chen Yang, Xiaoming Zhou, Chen Li, (2016), "A Movie Cold-Start Recommendation Method Optimized Similarity Measure," 2016 16th International Symposium on Communications and Information Technologies (ISCIT), pp.

231-234, China Qingdao.

- [34] Jinbo Zhang, Zhiqing Lin, Bo Xiao, Chuang Zhang, (2009), "An optimized item-based collaborative filtering recommendation algorithm," IEEE International Conference on Network Infrastructure and Digital Content, pp. 414-418.
- [35] <http://dns-learning.twnic.net.tw/internet/intro1.html>
- [36] <http://www.inside.com.tw/2010/05/14/internet-and-apple>