

黃純敏、林重佑、黃進瑞 (2013),『結合學習向量量化與協同過濾之交換混合式過濾電影推薦架構』,《資訊管理學報》,第二十卷,第四期,頁 423-448。

## 結合學習向量量化與協同過濾之交換混合式 過濾電影推薦架構

黃純敏\*

國立雲林科技大學資訊管理系

林重佑

國立雲林科技大學資訊管理系

黃進瑞

國立雲林科技大學資訊管理系

### 摘要

內容過濾與協同過濾是經常用於提供個人化服務的技術,近年來則多偏向結合各種監督式學習的混合式過濾方式,並以三層或多層式網路架構產生推薦結果,然而其設計不易且有網路收斂效率低的問題。本研究以學習向量量化(Learning Vector Quantization; LVQ)簡約的兩層式網路架構,運用交換(Switching)混合過濾策略產生推薦內容。研究以 MovieLens 資料集驗證方法架構。實驗發現,學習向量量化可快速學習使用者多變的喜好。搭配交換混合過濾策略,可產生適切的個人化內容,滿足不同使用者的推薦需求。研究結果顯示,本架構確可改善內容過濾與協同過濾各自的缺點,整體精確率為 79%,召回率為 82%。

**關鍵詞：**學習向量量化、推薦系統、混合過濾、協同過濾、內容過濾。

---

\* 本文通訊作者。電子郵件信箱：huangcm@yuntech.edu.tw  
2013/01/28 投稿；2013/05/13 修訂；2013/09/13 接受

Huang, C.M., Lin, C.Y. and Huang, J.R. (2013), 'Combining LVQ and Collaboration Filtering on Switching Hybrid Movie Recommendation', *Journal of Information Management*, Vol. 20, No. 4, pp. 423-448.

## Combining LVQ and Collaboration Filtering on Switching Hybrid Movie Recommendation

Chuen-Min Huang \*

Department of Information Management, National Yunlin University of Science & Technology

Chung-Yu Lin

Department of Information Management, National Yunlin University of Science & Technology

Jin-Ruei Huang

Department of Information Management, National Yunlin University of Science & Technology

### Abstract

Content-based filtering and collaborative filtering are often used to provide personalized services technology. Recently, lots of supervised neural networks are combined with hybrid recommendation and adopted three layers or multiple layers to construct recommendation. Their drawbacks are slow convergence and hard to design. In this paper, we presented a novel switching hybrid recommendation framework based on two-layer Learning Vector Quantization (LVQ) to provide personalized recommendations. MovieLens data set was used to test our framework and the experiment indicated LVQ can quickly detect and learn from user preferences. Results showed that switching hybrid strategy provides promising personalized recommendation and satisfied the needs of different users. Our experiment gains 79% of precision, and the recall rate also reaches 82%.

**Keywords:** Learning Vector Quantization, Recommendation System, Hybrid Filtering, Collaborative Filtering, Content-based Filtering

---

\* Corresponding author. Email: huangcm@yuntech.edu.tw  
2013/01/28 received; 2013/05/13 revised; 2013/09/13 accepted

## 壹、緒論

網路科技不斷的進步，商業活動亦受網際網路的影響，許多公司將各式服務合併至此平台，從傳統以銷售商品為導向的商業活動，轉變為以滿足顧客需求為主的商業行為，提供各式各樣的個人化服務，進而促成電子商務蓬勃的發展。在各公司不斷地投入電子商務的環境下，促使店家必須主動想辦法了解使用者多變的興趣與喜好，量身打造個人化商品來提高獲利。一般而言，內容過濾與協同過濾是提供個人化服務最主要的方法（Chou et al. 2010; Lang 1995; Yong et al. 2010; Yu 1999），它們也是推薦系統經常使用的過濾技術。內容過濾主要找出使用者的商品紀錄，計算商品之間的相似度，以產生推薦結果。此技術只需要取得使用者個人的商品紀錄和商品資料，即可處理。此為內容過濾的優點，但其無法處理不能被量化的內容，則是一大缺陷。協同過濾則計算使用者之間對商品項目評分的相似度，以產生推薦內容。此技術依使用者的喜好風格進行推薦，能提供符合使用者需求的結果，但是冷啟動造成的資料稀疏（Data sparsity）問題卻是難以克服的限制。因此許多學者將內容過濾與協同過濾做合併，以這兩種推薦技術的優點，彌補各自的缺陷，因而衍生混合過濾推薦技術的相關研究。Burke 回顧過去的混合過濾系統發展，歸納七種不同的混合策略（Burke 2002; 2007），提供研究學者參考，並已被廣泛引用在各相關推薦系統研究（Barragáns-Martínez et al. 2010; de Campos et al. 2010; Drachsler et al. 2008; Lekakos & Caravelas 2008）。

近二十多年來，類神經網路的研究與應用受到學術界的普遍重視。其中 Kohonen 提出學習向量量化（Learning Vector Quantization; LVQ）神經網路架構（Kohonen 1986），該架構包含輸入層與輸出層，具有簡單設計、快速分類與快速收斂等特性。部分研究嘗試結合類神經與推薦系統，提供個人化的推薦服務。其中（Christakou & Stafylopatis 2005; Lekakos & Caravelas 2008; Marovic et al. 2011）以 MovieLens 電影資料集的資料，結合類神經網路與協同過濾產生個人化推薦（GroupLens, 2011）。然而這些研究皆屬於較為複雜的監督式三層網路學習架構，包含輸入層、隱藏層、輸出層。過去 LVQ 在分類（Martín-Valdivia et al. 2003; Martín-Valdivia et al. 2007; Umer & Khiyal 2007）、電子郵件過濾（Chuan et al. 2005）等應用有良好的分類成效，但至目前仍未見到合併 LVQ 與推薦系統的相關應用。

本研究以電影推薦為範疇，運用 MovieLens 資料集，提出一個應用 LVQ 與協同過濾技術的混合過濾推薦架構，利用 LVQ 快速分類、收斂的特點，讓系統自動調整、學習使用者喜好，並結合 Burke 所提出的交換混合過濾策略，導引系統因應每位使用者不同的推薦需求，適時提供最適當的內容。由於 LVQ 並無可依循的規則來決定合適的 codebook vector 個數，欲提高 LVQ 的正確率，需盡量平衡訓練

資料集的類別分布並仔細規劃訓練參數的設定（學習率、訓練回合）。在本實驗過程中，對於如何決定 codebook vector 個數，來獲得在 LVQ 網路最佳收斂幅度及成本效益間取得權衡；讓使用少數資料集的 LVQ 內容過濾能與協同過濾以交換式策略互補，給予使用者正確的推薦結果，都是主要的挑戰議題。在此研究中，我們採用先前學者(Khuwaja 2003; Mizuno et al. 2010; 葉怡成 2003)所提試誤法(trial and error)，有系統地嘗試各項分合檢測與分析，研究結果顯示，推薦結果的精確率及召回率都有優異的成果。

本文第貳節回顧與本研究相關的工作，第參節詳細描述本研究的研究架構。第肆節提出評估方法，檢視所提出的概念成效收益。於第伍節提出總結，對往後研究發展提出未來展望。

## 貳、文獻探討

### 一、學習向量量化

向量量化(VQ)技術於1980年代開始有學者提出此方法(Gray 1984; Linde et al. 1980)，過去此技術被應用於語音壓縮與影像壓縮等處理應用。VQ為LVQ的前身，其基礎概念為利用取樣的資料群集，表示全部的輸入向量(input vector)。輸入向量依據它們本身的資料屬性，量化分配到較小的群集。在每群集能找到一個質心(centroid)，代表該群的中心點，相似屬性的輸入向量具有聚合效果，向量之間會彼此群聚在一起。

Kohonen 運用向量量化概念，在1986年提出LVQ類神經演算法(Kohonen 1986)。LVQ定義了一組具有類別屬性的輸入向量空間，以反覆學習能力，將輸入向量分為正確的範例與錯誤的範例。若輸入向量與輸出解答隸屬同一類，則正向調整權重(reward)；若輸入向量與輸出解答不屬於同一類，則反向調整神經元之間的權重(punishment)(Martín-Valdivia et al. 2007)。Kohonen所提出的學習向量量化的網路結構，捨棄隱藏層(hidden layer)，輸入層直接連接到輸出層，架構為一個兩層式的神經網路模型。輸入層為每一個帶有類別(class)屬性的input vector，輸出層上的每一顆神經元，為input vector對應到的類別屬性。本研究採此種簡約的兩層式神經網路。圖1為LVQ網路架構圖。輸入層裡的每一顆神經元表示為輸入向量 $X_i$ ， $i=1, 2, \dots, N$ ，共有 $N$ 個輸入範例。輸出層的每一個神經元，代表輸出向量 $Y_j$ ， $j=1, 2, \dots, m$ ，共有 $m$ 個輸出類別。在每一個輸入向量與輸出向量間，由不同的權重向量相互連接，這一個個 $w_{ji}$ ，就是所謂的codebook vectors。LVQ的監督式競爭學習特性，包含兩種執行階段：訓練階段(memorization，學習訓練資料內的正確範例)與測試階段(recall，運用測試資料使神經網路回想在訓練階段建立的網路模型，進行分類評估與測試)。當一個輸入向量 $X_i$ 與輸出向量 $Y_j$

在訓練階段具有相當高的關聯，連結  $X_i$  與  $Y_j$  的 codebook vector  $w_{ji}$  將在每一次的學習步驟中，不斷提升權重，拉近距離。反之，則不斷降低權重，使  $X_i$  與  $Y_j$  相互遠離。

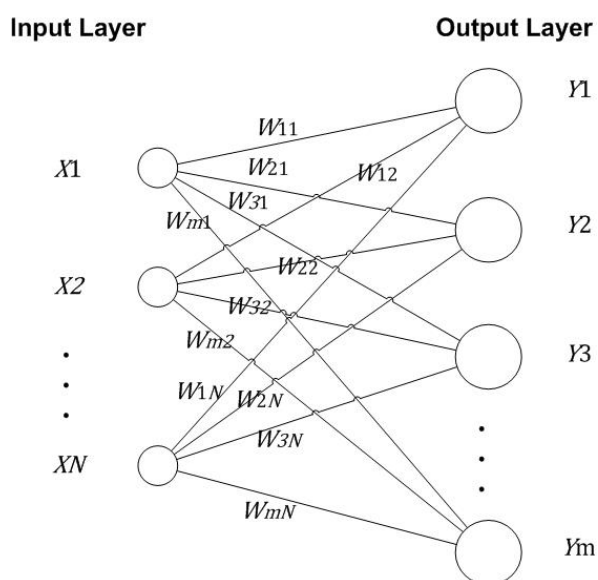


圖 1：LVQ 神經網路架構 (Martín-Valdivia et al. 2007)

一般來說，codebook vectors 的合理數量並不容易決定，欲提高 LVQ 的正確率，需盡量平衡訓練資料集的類別分布並仔細規劃訓練參數的設定（學習率、訓練回合）。過去研究以試誤法 (Khuwaja 2003; Mizuno et al. 2010; 葉怡成 2003) 來檢測 LVQ 在何種參數設定能讓精確率與效率達到最佳設定，以使學習模型達到目的。由於 LVQ 的訓練時間顯著低於其他類神經網路，常用來處理大量資料集，因此十分適合用來解決分類及視覺辨識等相關問題。在 (Martín-Valdivia et al. 2007) 研究中，LVQ 被應用在路透社文件 (Reuters-21578) 自動分類與語意分析等兩種用途。而 (Umer & Khiyal 2007) 同樣也使用 Reuters-21578 資料集，將 LVQ 應用於分類的範疇。這兩者的實驗結果都認為學習向量量化具有傑出的分類表現。其中特別的是，(Umer & Khiyal 2007) 特別專注比較於 LVQ 不同版本演算法的分類績效，在該研究中依 Kohonen 提出的五種不同版本的 LVQ 演算法：LVQ1、LVQ2.1、LVQ3、OLVQ1、OLVQ3，分析不同版本的學習向量量化演算法，本研究使用其實驗比較結果中準確性最佳的 OLVQ1 演算法。

## 二、推薦系統

推薦系統的主要概念在於把適當的物品推薦給對該物品有興趣的使用者。與傳統的資訊檢索系統不同的部分在於，使用者毋須主動下關鍵字搜尋，系統只要藉由其瀏覽紀錄，也能夠主動推薦結果（Das et al. 2007）。而推薦系統的分類，主要分成兩種推薦方式：分別為內容過濾（content-based filtering）與協同過濾（collaborative filtering）（Ahn 2008; Kamahara et al. 2005; Sung 2006）。這兩種推薦方式除了都有資料稀疏及冷啟動的問題，也有其各自的缺點。有些研究者為了克服內容過濾與協同過濾的缺點，把這兩項推薦技術混合搭配。並且證實混合過濾（hybrid filtering）技術是相當優良的推薦方法（Adomavicius & Tuzhilin 2005; Balabanović & Shoham 1997; Burke 2002; de Campos et al. 2010）。

### （一）內容過濾

內容過濾是一種利用項目本身的特徵值，群聚類似品項，以當作推薦的依據（Lekakos & Caravelas 2008）。並以使用者回饋進行推薦結果修正，當使用者因某些特徵的不同而不喜歡某項商品，則降低這些特徵的權重；反之，則增加相關特徵的權重。此技術根據使用者過去對於商品的喜好，擇其所愛，推薦其可能會喜歡的商品。以書籍推薦系統為例，系統擷取使用者過去的閱讀紀錄，分析該使用者常看的書籍當中，有哪些作者、作品類別、以及作品經常出現的字詞（Lekakos & Caravelas 2008）。著名的內容過濾推薦系統如NewsWeeder（Lang 1995）和LIBRA（Mooney & Roy 2000），但其缺點可歸納為以下兩點（Kamahara et al. 2005）：

1. 只能根據使用者過去使用的紀錄判斷，但是不能隨時間顯示使用者喜好的轉變。
2. 對於新進的品項，礙於使用者未曾接觸，沒有與新進項目類似的特徵，因此無法提供推薦。

### （二）協同過濾

協同過濾運用一組彼此擁有相同喜好或品味的使用者，藉由物以類聚原理，推薦使用者可能喜愛的物品（Adomavicius & Tuzhilin 2005; Breese et al. 1998）。Pearson correlation coefficient 是經常被用來計算使用者間彼此相似程度的演算法（Lekakos & Caravelas 2008）。

協同過濾最早、最知名的系統為 Tapestry（Goldberg et al. 1992）。它是一個協助使用者過濾電子郵件的系統。其他著名的推薦系統為新聞推薦系統 GroupLens（Resnick et al. 1994）與電影推薦系統 MovieLens（Adomavicius & Tuzhilin 2005）。協同過濾常用的演算法有兩種：Model-based 與 Memory-based 演算法（Breese et al. 1998; de Campos et al. 2010; Symeonidis et al. 2008），茲分別介紹如下：

1. Model-based：此演算法運用資料庫裡的所有使用者的評分資料來建立模型

架構。這個模型能夠幫助推薦系統產生預測結果給使用者（Breese et al. 1998）。此演算法認為資料庫內彼此相似的物品，可能大多都有類似的評分。而建立 model 的三種方法，包含了 Bayesian network、clustering、以及 rule-based methods（Sarwar et al. 2001）。由於此演算法需要依賴使用者評分紀錄來分群分類，建立的學習模型，才能給予使用者推薦，如當發生給予評分的使用者項目過少，將造成相似度計算的困難，這也就是資料稀疏的問題，此外對於新使用者，因尚未對任何資料進行評分，系統則無法進行推薦。（Lekakos & Caravelas 2008; Acilar & Arslan 2009）。

2. Memory-based：此演算法同樣運用資料庫裡所有的評分資料進行推薦，將使用者可能感興趣的預測結果回傳給使用者。這種演算法可以依其需要的推薦數量以及鄰居數的參數，計算相似度來預測估計值，並不需要建立出學習模型，是目前常用的演算法，但依然有資料稀疏與新使用者問題存在。本研究所使用的 MovieLens 資料集，由於評分資料已建立完成，且有足夠的使用者與評分項目，因此不會有上述問題產生。

### （三）混合過濾

混合過濾是指結合了兩個或兩個以上的推薦元件，最常見的混合方式為結合內容過濾與協同過濾（Barragáns-Martínez et al. 2010）。Burke（2002; 2007）調查以往的混合過濾推薦系統，定義混合過濾共有七個不同策略：

1. Weighted：將兩種推薦元件的輸出，使用單一線性加權處理，為七種策略中最簡單的一種。但僅適用於所採用的兩種推薦元件具有同等分量或穩定的正確率。
2. Mixed：顧及資料稀疏問題，此策略結合不同推薦元件，將推薦結果並排陳列以進行推薦。至於如何將推薦結果合理的排序，則是一大挑戰。此外，此種方式僅適用於線上即時推薦，並不適用於回溯資料實驗。
3. Switching：將推薦元件依據推薦情況排序，如果排序最高的推薦元件無法輸出滿意的推薦結果，則由第二順位遞補，以此類推。不同的推薦元件，可設定各自的門檻值。此策略適用於單一推薦元件可能無法滿足所有使用者的情況。
4. Feature Combination：把資料集 A 的 features，置入演算法以處理資料集 B 的資料。不同於其他混合策略以處理不同推薦元件為訴求，此策略僅運作於一個推薦元件的不同資料集。比方由內容過濾推薦系統建立出使用者的喜好，如使用者 1 與使用者 2 都喜愛電影 X 時，將結合電影的特徵值（電影 X 是由演員 1 與演員 2 所主演），推導出使用者 1 與使用者 2 可能都喜愛演員 1 與演員 2 所演的電影。

5. Feature Augmentation：此策略與 Feature Combination 在某些程度上相似，不同的是此推薦策略並未使用推薦元件原有領域的特徵值，卻透過推薦邏輯增加新的特徵值。例如：利用關聯法則（association rule）處理協同過濾的資料，並將所產生的新內容特徵值給內容過濾元件使用。此策略適用於處理回溯資料。
6. Cascade：鑒於有些推薦機制輸出過多同評等的品項，此策略嚴格要求將評等區分階層。若一個混合過濾系統有兩個推薦元件，會先訂出主要與次要推薦元件，假若主要推薦元件所產生的推薦結果雷同的話，將則啟動次要推薦元件重新評量，進而產生推薦結果。此策略並不適用於可量化的推薦機制。
7. Meta-level：從一個推薦元件學習到的模型，當作其他推薦元件的輸入。此策略與 Feature Augmentation 類似，但不同處為 Meta-level 將某一個推薦元件學習到的模型，用來取代舊有的資料集，而另一推薦元件就會應用此學習模型進行計算與推薦。然而此策略的可行性常被質疑，因為很難找到一種推薦元件可完全充當另一機制的輸入。

前述的七項混合策略，其中 Weighted 僅適用於所採用的兩種推薦元件具有穩定的正確率；Mixed 不適用於回溯資料實驗，僅適用於線上即時推薦，而 Feature Combination 僅著眼於一種推薦元件。本研究以內容過濾為主，協同過濾為輔，資料集為回溯性、使用者所評量的結果並無穩性的正確性，因此上述三種都不適用。此外 Cascade、Feature Augmentation 與 Meta-level 三種策略會因推薦元件結合順序不同，產生不同推薦結果，本研究以內容過濾為優先，資料集可數值化，易於區分出優先順序。因此排除這三種策略的選擇。最後決定以 Switching 策略混合過濾技術產生推薦結果。

## 參、研究方法

### 一、方法概觀

本研究以美國明尼蘇達大學（University of Minnesota）的 MovieLens 資料庫為實驗對象。研究架構共分三種推薦方式，如圖 2 所示，應用 LVQ 演算法進行內容過濾、協同過濾，以及結合兩種技術的 Switching 混合過濾推薦策略。內容過濾依據 Movie Dataset 與 User Rating Dataset 分析其喜愛的電影有哪些特徵與評分紀錄，以產生推薦結果。而協同過濾應用 User Rating Dataset 計算 K 位與使用者最相似的鄰居，過濾出使用者最值得觀看的 Top 5 清單。以往採用 MovieLens 為實驗資料集的混合過濾電影推薦研究，多以電影的導演（director）、演員（actor）、類型（kind）進行推薦（Christakou & Stafylopatis 2005; Lekakos & Caravelas 2008; Marovic et al.



2011)，因此本研究沿用上述三項電影特徵用於 LVQ 分類。表 1 描述該資料集特色。由於本研究僅提供個人化的推薦內容，不對使用者分群，人口統計資訊並沒有應用於本研究。在電影類別資料中，總共有 19 類資料，如表 2

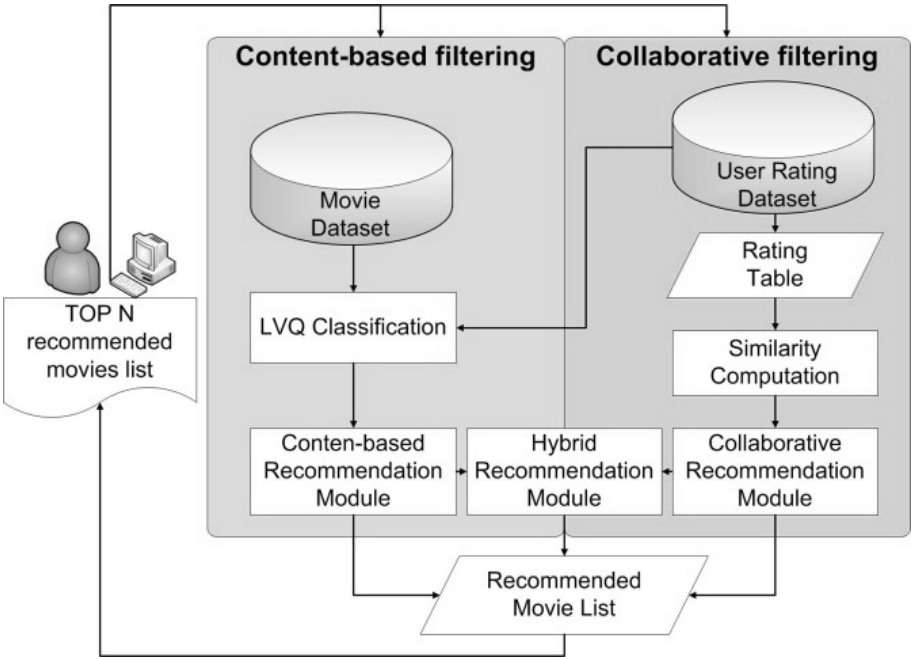


圖 2：本研究架構圖

表 1：MovieLens 100k 資料集描述

MovieLens 100k 資料集描述	
1.	資料集共有 943 位使用者，對 1682 部電影的評分，總計 100,000 筆評分紀錄。
2.	每位使用者評分的範圍從 1 分到 5 分，1 分代表非常不喜歡，5 分代表非常喜歡。
3.	每位使用者至少評分過 20 部電影。
4.	每部電影的類型（genre）由 Internet Movies Data Base（IMDB）定義。 URL: <a href="http://www.imdb.com/">http://www.imdb.com/</a>
5.	資料集包含簡易的使用者人口統計資訊，包括：性別（sex）、職業（profession）、年齡（age）等屬性。
6.	資料從 1997 年 9 月 19 日開始蒐集，直到 1998 年 4 月 22 日為止。

表 2：MovieLens 19 類電影對照表

No.	Name	No.	Name	No.	Name
1	Unknown	8	Documentary	15	Romance
2	Action	9	Drama	16	Sci-Fi
3	Adventure	10	Fantasy	17	Thriller
4	Animation	11	Film-Noir	18	War
5	Children's	12	Horror	19	Western
6	Comedy	13	Musical		
7	Crime	14	Mystery		

本研究依 Burke 提出的交換混合策略，以內容過濾為優先並設置兩種原則：(1) 若使用者不能從內容過濾的 LVQ 神經網路得到推薦結果，則使用協同過濾進行推薦。(2) 若協同過濾不能提供推薦結果，則隨機讓使用者觀看最新的電影資訊，增加使用者評分電影的機會。

## 二、內容式過濾

### (一) LVQ 分類機制

本實驗為每位使用者設置三個 LVQ 神經網路，分別為導演、演員、類型。以不同數量的神經元個數及訓練步驟數進行訓練，以找出最理想的參數設定。如果不同的實驗步驟，產生相同的情況（亦即產生相同的正確率），則以最小神經元個數為優勝者。

LVQ 內容過濾核心演算法說明如下（Kohonen et al. 1996）：

1. 初始化 codebook vectors  $W_i$ 、學習率  $\alpha$ 。  
（學習率  $\alpha$  初始值推薦設定為 0.3 或 0.5）
2. 隨機選擇輸入向量  $X_i$ 。
3. 找出所有輸入向量之中，與 codebook vector 最接近的向量在公式(1)， $w_c$  是一個與輸入向量  $X_i$  有最小歐幾里得距離的 codebook vector，公式(2)為一個找出最小歐幾里得距離的 codebook vector 的範例。

$$\|x_i - w_c\| = \min_k \|x_i - w_k\| \quad (1)$$

$$c = \arg \min_k \|x_i - w_k\| \quad (2)$$

4. 運用輸入向量所帶的屬性，與 codebook vector 所帶類別比較，更新權重。  
如公式(3)， $W_c(t)$  代表網路執行第  $t$  次學習步驟。若  $W_c$  與  $X_i$  為不同類，每次進行學習時，將學習率  $\alpha$  乘上輸入向量  $X_i$  與 codebook vector 的  $W_c$  遠離之後的距離，最後更新該次 codebook vector  $W_c(t)$  權重設定。

$$W_c(t+1) = W_c(t) - \alpha(t) [X_i(t) - W_c(t)] \quad (3)$$

5. 降低學習率  $\alpha$ ，重複執行步驟 0 直到所有的訓練階段結束，或是網路已經穩定。學習率  $\alpha(t)$  經過每一次訓練步驟會不斷收斂縮小， $(0 < \alpha(t) < 1)$ 。

由於 LVQ 為監督式學習，使用者看過的每一部電影，即為一筆輸入向量，各向量都必須帶有正確的訓練範例，需先把使用者的電影紀錄調整為電影—電影屬性矩陣，再將矩陣輸入到 LVQ 網路進行學習訓練，建立使用者喜好模型。

表 3 說明電影—電影屬性矩陣。當某位使用者使用內容過濾進行推薦，系統即擷取該使用者過去對電影的評分紀錄。本例中縱軸代表使用者共看過 272 部電影，橫軸表示這些電影總共出現過 19 位演員，如演員 2 在第二部電影演出，以「1」表示，而演員 2 沒有演出第一部電影，以「0」表示。而橫軸最後一欄「Rating」欄位，代表使用者給予該電影評分。本研究將評分定義為低評分（1-3 分），是不值得推薦的電影、高評分（4-5 分）是值得推薦的影片。LVQ 藉由不斷調整 codebook vector 的權重，使得神經網路能夠學習高評分與低評分電影屬性。

表 3：電影屬性與評分矩陣

	Actor1	Actor2	...	Actor19	Rating
Film1	0	0		0	5
Film2	0	1		0	3
...	...	...		...	...
Film272	0	0	...	0	3

## （二）內容式推薦

當 LVQ 完成分類階段以後，同時也完成 LVQ 的網路學習訓練階段。推薦模組對測試資料進行分析，將三個神經網路（導演、演員、類型）分類的結果，從五等第資料轉變為二元資料，判斷哪些電影的原始評分為 4-5 分，並把測試資料中預測結果為高評分的電影推薦給使用者。反之，則不把隸屬於低評分的電影推薦給使用者。

### 三、協同過濾

#### (一) 相似度計算

本研究採用 Pearson Correlation 計算使用者之間的相似程度。對某一位使用協同過濾的使用者而言，系統首先從使用者評分資料庫中，找出其他使用者的評分紀錄，將其曾經給予過同部電影評分計算相似程度，計算如公式(4)：

$$r = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2} \sqrt{\sum_i (Y_i - \bar{Y})^2}} \quad (4)$$

$X_i$  代表使用者 X 對電影 i 的評分， $\bar{X}$  代表該使用者過去所有電影評分的平均值。而  $Y_i$  代表資料庫中的其他使用者 Y 對電影 i 的評分， $\bar{Y}$  代表其他使用者 Y 過去所有電影評分的平均值。其中  $r = [-1...1]$ 。若使用者 X 和其他使用者 Y 的相似程度越高（r 值越趨近於 1），代表他們對電影的喜好相當類似；若相似程度越小（r 值趨近於 -1），代表這兩位使用者的對電影的喜好有很大的差異。若有一個五等級電影評分矩陣 M 存在四位使用者 Ken、Lee、Meg、Nan 與六部電影，ID 分別為 1、2、3、4、5、6（如表 4）。表內的數字代表各使用者對電影的評分，而「？」代表協同過濾欲推估使用者 Ken 及使用者 Nan 對電影 6 的評分。若計算 Ken 與 Lee 的關聯程度，則可應用上述公式，進行相似度運算：

表 4：電影評分矩陣 M

Movie ID	Ken	Lee	Meg	Nan
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	?	2	5	?

$$r_{KL} = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2} \sqrt{\sum_i (Y_i - \bar{Y})^2}} = \frac{-2-2-2-2}{\sqrt{10}\sqrt{10}} = -0.8$$

從計算結果得知，Ken 與 Lee 的相似程度為 -0.8，代表這兩位使用者對電影的喜好並不同。其他使用者之間的相似度計算，也以此類推。

## (二) 協同式推薦

當系統計算完待推薦使用者 X 與其他使用者的相似程度後，協同過濾推薦機制即運用預測演算法，計算使用者 X 相似的鄰居使用者 J 位的加權平均數，來預測使用者 X 對某一個待推薦物品 p 的喜好程度。計算如公式(5)：

$$X_p = \bar{X} + \frac{\sum_1^J (J_p - \bar{J}) r_{XJ}}{\sum_1^J |r_{XJ}|} \quad (5)$$

$X_p$  代表即將推薦給使用者 X 的電影 p 之預測值， $\bar{X}$  代表使用者 X 過去所有電影評分平均值。 $r_{XJ}$  表示使用者 X 與鄰居 J 的相似程度。 $J_p$  代表鄰居 J 對於電影 p 的評分，而  $\bar{J}$  為鄰居 J 過去所有電影評分平均值。若協同過濾系統要預測 Ken 對 Movie ID = 6 的喜好程度，則會應用 Lee 與 Meg 的評分紀錄（因為他們都有對第六部電影給予評分），進行預測值的估算。

$$\begin{aligned} X_{movieID=6} &= \bar{X} + \frac{\sum_1^J (J_6 - \bar{J}) r_{XJ}}{\sum_1^J |r_{XJ}|} \\ &= 3 + \frac{3r_{XM} - r_{XL}}{|r_{XM}| + |r_{XL}|} = 3 + \frac{2 - (-0.8)}{|1| + |-0.8|} = 4.56 \end{aligned}$$

從以上結果得知，Ken 極有可能喜歡電影 ID 為 6 的這部影片。上述例子為推薦 Top 1 最值得觀看的電影清單。推薦系統另一重要目的，即縮小使用者檢索眾多資料的成本，並且提供最符合使用者需求的產品。因此，本研究限制協同過濾所推薦的資訊，一次最多不超過 5 筆，協同過濾最後產出 Top 5 的推薦結果給使用者。

## 四、交換式策略混合過濾

本研究應用交換式策略混合過濾的推薦方式簡述如下：

1. 內容過濾運用 LVQ 分類某位使用者看過的電影具有哪些特徵，並加上使用者對電影的評分，推薦使用者可能喜歡的內容。若內容過濾不足以提供有效的推薦結果，依混合過濾的 Switching 機制，利用協同過濾來給予使用者推薦結果。
2. 協同過濾計算某使用者與其他 K 個喜好最相近的鄰居相似度，產生推薦結果。並依相似度高低，排序推薦內容。若協同過濾亦不足以提供有效的推

薦結果，則隨機讓使用者觀看最新的電影資訊，增加使用者評分電影的機會。

## 肆、實驗結果與討論

本研究實驗流程共區分為三個部分，分別使用內容過濾、協同過濾，以及混合過濾等實驗。在內容過濾部分，使用試誤法，學習率在0.3與0.5的情況下，觀察使用LVQ正確率的變化。在協同過濾的部分，本研究運用精確率的計算，驗證推薦系統的準確度。

### 一、評估指標

#### (一) 平均絕對誤差

平均絕對誤差 Mean Absolute Error (MAE) 是一個用來評估數值型預測資料與使用者實際評分間誤差大小的指標 (Sarwar et al. 2001)。在公式(6)， $N$  代表某一個操作推薦系統的使用者所有電影評分資料總數。令  $q_i, i=1, \dots, N$  為某一演算法估計得到的預測值，則  $q_j, j=1, \dots, N$  為該使用者實際給定的評分。當 MAE 的值越來越小，表示該演算法的正確率越來越高。

$$MAE = \frac{\sum_{i=1}^N |p_i - q_j|}{N} \quad (6)$$

#### (二) 精確率、召回率

資訊檢索領域中經常使用精確率、召回率兩種指標。精確率為所有取回資料中，相關資料的比率為多少個。而召回率為評估系統對於所有相關資料中，正確取回的個數有多少個。

表 5 指出精確率為推薦系統推估的項目中，使用者所需項目比例，如公式(7)。召回率為所有使用者所需項目中，推薦系統推估項目的比例，如公式(8)。

表 5：Precision、Recall 對照表

	推薦項目	沒推薦項目
使用者所需項目	A	B
使用者不需要項目	C	D

$$\text{Precision} = \frac{A}{A+C} \quad (7)$$

$$\text{Recall} = \frac{A}{A+B}$$

(8)

二、LVQ 內容過濾績效驗證

LVQ 評估的目的有二：(1)分析不同神經網路在不同評分筆數最佳正確率的變化、(2)評估 LVQ 推薦的精準度。實驗以 MAE 評估 LVQ 在內容過濾的正確率表現績效，藉此找出學習向量量化網路的最佳化參數設定。LVQ 為監督式學習神經網路，網路設定以試誤法進行，以求找到最佳分類解答。試誤法是一種用來解決問題以獲取知識常見的方法，通常適用於沒有明顯規則可以套用於待解問題時使用 (Putnam 1965)。主要透過不斷嘗試，並在犯了許多錯誤之後而獲致成功的過程。在嘗試的過程中，可能有許多動作或步驟，看來十分愚蠢或不必要，隨著動作不斷反覆，錯誤不斷遞減，最後獲致完全的成功 (Callander 2011)。甚至專業的圖書館員也常以此方法協助使用者進行資訊檢索 (Swanson 1977)。當然，我們通過嘗試錯誤所學習到東西，並不像通過嘗試成功學到的那樣多，那樣好。同樣，在學習中，我們不能在錯誤的方法中尋求到正確的答案，只能在正確的方法中找到正確的答案。嘗試錯誤法也不代表一定要不經意地亂試解法，使用者可以有條理地操控各個變因來整理出最有機會成功解決問題的解法。亦即，通過試誤法的學習，要能盡量減少或避免錯誤，獲得學習成功。因此，本研究採用 (Kohonen 2001) 的建議，設定學習率與網路訓練次數初始值。該研究建議學習率分別以 0.3 或 0.5 進行測試。網路訓練次數，則設定為 30 至 50 倍於 codebook vector 個數。而 codebook vector 數量，則需要數倍於輸出層神經元個數 (葉怡成 2003)。因本研究設計三種 LVQ 神經網路，輸出層共 5 類 (5 顆神經元)，並從實驗結果觀察得知，過多的 codebook vector 個數，徒增計算成本，最後設計 codebook vector 個數為 2 到 40 倍於輸出層神經元個數，藉此觀察 LVQ 網路收斂的幅度。LVQ 於本實驗參數設定對照如表 6 所示。

表 6：LVQ 網路參數初始值設定對照表

學習率	0.3	0.5
Codebook vector 個數	整數倍於輸出層神經元個數。本實驗三種神經網路的輸出層個數各有 5 顆神經元，codebook vector 個數從 10 顆到 200 顆神經元。	
網路訓練次數	Codebook vector 個數之 30 倍至 50 倍。	

為評估 LVQ 的推薦結果與 LVQ 正確分類的下限，本研究觀察 MovieLens 資

料集提供的使用者評分紀錄，由於資料庫中的 943 位使用者，最少的評分筆數為 20 筆，最多為 737 筆，本研究以均分方式將評分筆數以每 100 筆作分割，分為 8 個區塊，從每一區塊進行抽樣，觀察評分筆數的變化，對 LVQ 正確率的影響。圖 3 呈現三種不同神經網路 MAE 的變化結果。本研究以交叉驗證 (n-fold cross validation,  $n=10$ ) 方式進行實驗，發現在評分資料筆數介於 201 到 300 筆、701 到 737 筆間，擁有較低的誤差值 (較佳的正確率)，其平均誤差值分別為：0.1471 以及 0.1371。

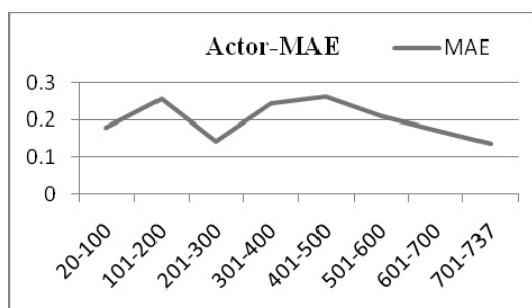


圖 3：Actor 整體評分筆數誤差值分布

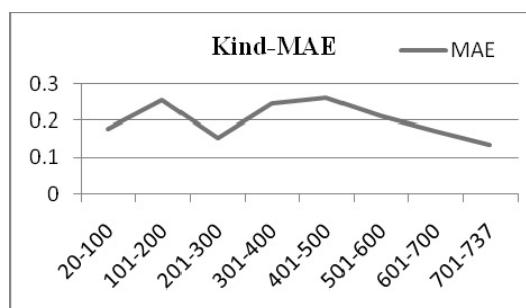


圖 4：Kind 整體評分筆數誤差值分布

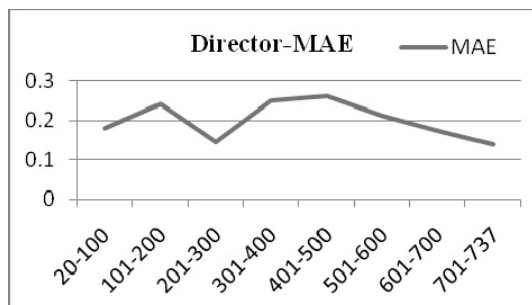


圖 5：Director 整體評分筆數誤差值分布

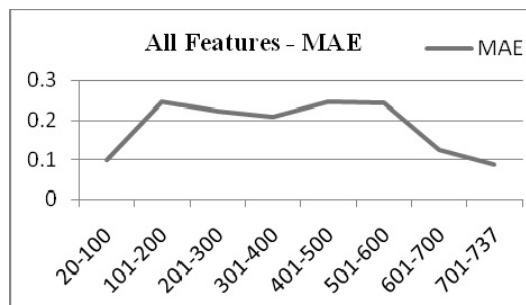


圖 6：合併三項屬性整體評分筆數誤差值分布

從抽樣結果得知，MAE 的值容易受到個別使用者的評分資料影響。每位使用者依據看過電影數目的多寡，各自有不同的評分筆數，若各個使用者清楚地表示曾經看過電影的喜好程度有哪些，且各評分的資料筆數彼此相近，LVQ 對於各評分有足夠的範例進行學習時，分類正確率能更為提升。

本研究另外發現在一開始評分筆數的增加，會讓 MAE 值遽升；但隨著評分筆數的成長，MAE 值也隨之降低。除了觀察上述三個神經網路各自的 MAE 變化之外，本研究試圖讓 LVQ 神經網路同時考慮到三種電影屬性，觀察 MAE 的變化，



結果顯示同時考慮較多的電影屬性，在評分筆數最少及最多兩端（20-100、701-737 筆）下，誤差值較小（分別為 0.1 與 0.0871），有較佳的分類正確率如圖 6 所示，與過往的研究（Li & Kim 2003）比較，本實驗的 MAE 值有明顯的降低。可知使用者對電影的評分筆數越多，LVQ 分類正確率更趨於穩定。

雖然正確率的大小代表 LVQ 分類績效之良窳，但正確率是否影響電影推薦之結果？需要進一步驗證。因此，本研究接續進行分類結果的推薦功能評估，評估指標著重在電影推薦的精確率與召回率。此實驗將個別使用者的評分資料切割成訓練資料與測試資料，若系統對測試資料預測的評分落在 4 到 5 分時，而使用者實際上給的評分亦為 4 或 5 分，則判定 LVQ 成功推薦電影給使用者；反之，則不推薦該電影給使用者。以圖解釋某一位使用者測試資料的評估結果，該混合矩陣在 X 軸上的 a、b、c、d、e 代表 LVQ 演算法的預測分類結果，Y 軸上的  $a=R1$ 、 $b=R2$ ... 到  $e=R5$ ，代表每一筆測試資料的正確解答。圖形共分成四個區域：區域 A 代表系統成功推薦的電影共有 39 筆，區域 B 代表系統未能正確推薦給使用者需要的 9 筆電影資料，區域 C 代表系統推薦使用者不需要的 22 筆電影，而區域 D 代表系統未推薦且使用者也不需要的電影筆數，共 10 筆。最後以精確率與召回率評估內容過濾推薦績效。本例求得精確率為 0.64，召回率為 0.81。

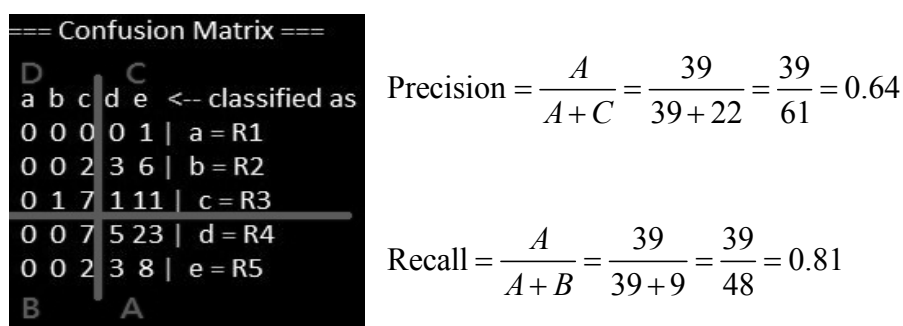


圖 7：LVQ 內容過濾混合矩陣

實驗抽樣 30 名使用者資料，如前所述，在評分筆數最少及最多兩端（20-100、701-737 筆）下，誤差值較小（分別為 0.1 與 0.0871），有較佳的分類正確率。以下實驗即抽取此二區段的使用者，來設定網路學習率與訓練回合數，評估每位使用者在成功推薦的情況下（預測評分為 4 到 5 分），三個不同網路（導演、演員、類型）個別產生與合併產生之精確率與召回率，結果如表 7 所示。個別神經網路精確率約為六成，推測原因可能是因單一的電影屬性，難以提供準確的精確度。例如當多部電影由同一位導演所監製，但該導演所執導的片子，可能是不同的電影類型。如 Steven Spielberg 導過的電影主題，包含科幻與戰爭類型，因此系統無

法以唯一特徵來描述導演的特性，但相對的，能夠找到較多相關的電影，造成精確度較低，而召回率較高的結果。但將三個神經網路結合後，精確率雖並未提高，召回率則趨近百分之百，由此發現合併所有神經網路，雖提高了資料維度，結果依然很穩定。

表 7： Actor、Director、Kind 網路之 Precision 與 Recall

Networks	Precision	Recall
Actor	0.65	0.89
Director	0.62	0.98
Kind	0.62	0.95
All Features	0.61	0.99

### 三、協同過濾績效驗證

本研究設定每位使用者會得到 Top 5 的推薦清單。為確認這 5 筆推薦出來的電影，是否為使用者所需要？因此，本研究評估協同過濾計算出來的推薦清單中，系統成功預測的比例為何，以及在所有該被推薦給使用者的項目中，找回多少筆數。

實驗觀察以不同的鄰居數，推薦系統所產生的推估值，滿足使用者的預期需要的程度。本實驗參考的電影資料集，運用評分值的大小做為門檻值，認為系統產生的推估值至少要大於 3 分，才視為成功的推薦項目。

以一實例解釋協同過濾評估方式，本研究擷取系統中 User ID 為 655 的使用者，觀察鄰居數在等於 5 或 10 的情況下，精確率與召回率的變化。由表 8 得知，當  $K=10$ ，Top 5 推薦清單中的每一個項目的推估值，均大於 4，精確率與召回率均為 1。由表 9 得知，當  $K=5$ ，Top 5 推薦清單的精確率為 1，召回率為 0.8。然而發現雖然精確率高，但計算出來的評估值偏低，甚至有兩個值（3.0272453 及 3.0119343）幾乎趨近於低評分，原因可能為選擇較低的鄰居數會使得比對各使用者的相似度時，沒有足夠的樣本數據來提升我們的預測評估值。從此例得知，選擇較大的  $K$  值，會有較好的精確度與召回率。但其實過多的鄰居數，也將增加計算成本，需要更多時間進行相似度計算。因此，鄰居數的選定需自行權衡。

其後本實驗將鄰居數  $K$  設為 10，觀察 30 位使用者精確率與召回率的變化程度。實驗結果發現，協同過濾推薦的精確率達 0.8，召回率約為 0.73。

表 8：UserID = 655 推薦清單與估計值（ $K = 10$ ）

Recommended Movie ID	Estimate Value
309	5
173	4.512753
1025	4.482218
878	4.4762607
358	4.336608

表 9：UserID = 655 推薦清單與估計值（ $K = 5$ ）

Recommended MovieID	Estimate Value
748	3.5043457
343	3.4940329
689	3.0272453
355	3.0119343
322	2.5229743

表 8 中， $K = 10$  的預測精確率與召回率

$$\text{Precision} = \frac{A}{A+C} = \frac{\text{推薦門檻值} \geq 3 \text{ 的推薦項目}}{\text{推薦門檻值} \geq 3 \text{ 的推薦項目} + \text{推薦門檻值} < 3 \text{ 的推薦項目}} = \frac{5}{5+0} = 1$$

$$\text{Precision} = \frac{A}{A+B} = \frac{\text{推薦門檻值} \geq 3 \text{ 的推薦項目}}{\text{推薦門檻值} \geq 3 \text{ 的推薦項目} + \text{推薦門檻值} \geq 3 \text{ 的未推薦項目}} = \frac{5}{5+0} = 1$$

表 9 中， $K = 5$  的預測精確率與召回率

$$\text{Precision} = \frac{A}{A+C} = \frac{\text{推薦門檻值} \geq 3 \text{ 的推薦項目}}{\text{推薦門檻值} \geq 3 \text{ 的推薦項目} + \text{推薦門檻值} < 3 \text{ 的推薦項目}} = \frac{4}{4+1} = 0.8$$

$$\text{Precision} = \frac{A}{A+B} = \frac{\text{推薦門檻值} \geq 3 \text{ 的推薦項目}}{\text{推薦門檻值} \geq 3 \text{ 的推薦項目} + \text{推薦門檻值} \geq 3 \text{ 的未推薦項目}} = \frac{4}{4+0} = 1$$

#### 四、混合過濾績效驗證

評估混合過濾的重點為判斷內容過濾無法正確提供推薦時，協同過濾能夠成功提出推薦電影的程度；以及協同過濾無法正確提供推薦時，內容過濾成功提出推薦電影的程度。評估指標亦為推薦系統的精確率與召回率。實驗從 30 位使用者觀察混合過濾的推薦績效，來縱觀本研究所有實驗，如圖 8 所示。本研究提出的 LVQ 內容過濾推薦精確率達到六成，召回率趨近於百分之百，協同過濾精確率為 0.8，召回率為 0.73，藉由 Switching 策略能夠協助使用者避免無法得到推薦結果的困境，使最後的精確率與召回率皆取得平衡，實驗結果整體精確率較內容過濾佳，提升至 0.79；召回率較協同過濾佳，提升至 0.82。

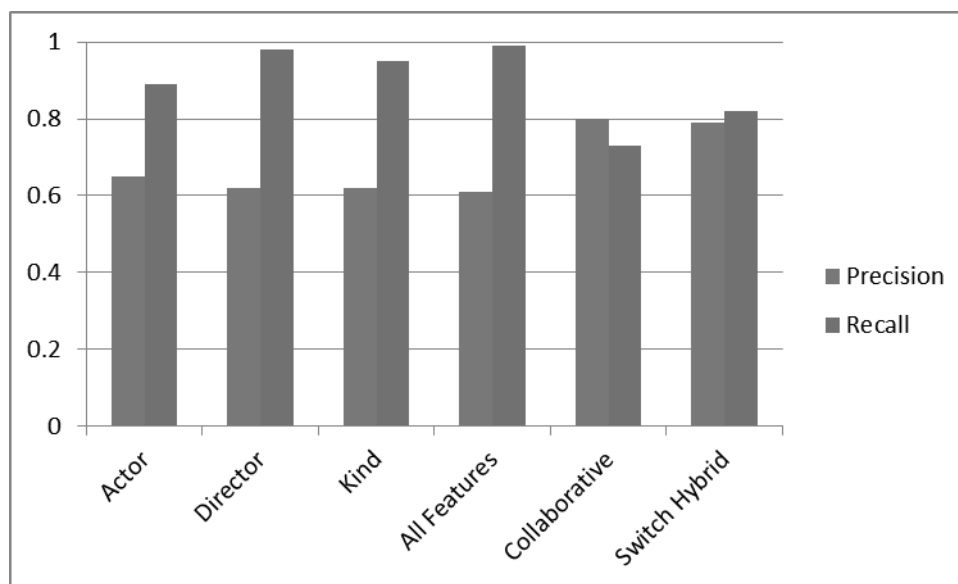


圖 8：本研究推薦方法之 Precision、Recall

#### 伍、結論

過去研究嘗試結合類神經與推薦系統，提供個人化的推薦服務。然而這些研究皆屬於較為複雜的監督式三層網路學習架構，此外，過去 LVQ 在分類、電子郵件過濾等應用雖有良好的分類成效，但至目前仍未見到合併 LVQ 與推薦系統的相關應用。再者，過去研究指出內容過濾與協同過濾兩種推薦方式，都有資料稀疏及冷啟動的問題，本研究依據 Burke 調查以往的混合過濾推薦系統，所定義的七個不同策略，發現其中交換式策略適合處理回溯性資料集，可將推薦元件依據推

薦情況排序，如果排序最高的推薦元件無法輸出滿意的推薦結果，則由第二順位遞補，以此類推。不同的推薦元件，可設定各自的門檻值。此策略適用於單一推薦元件可能無法滿足所有使用者的情況，也可能用來解決資料稀疏及冷啟動的問題，由於本研究以 MovieLens 資料庫為實驗對象，性質屬於回溯性資料集、使用者所評量的結果並無穩性的正確性，且以內容過濾為優先，協同過濾為輔，因此採用此策略進行實驗。

綜合實驗結果發現(1)若同時將多個電影屬性結合成一個神經網路，精確率並未提高，召回率則趨近百分之百，由此發現合併所有神經網路，雖提高了資料維度，結果依然很穩定。未來進行 LVQ 分類，可考慮提高 LVQ 演算法的參數設定，以期獲取更佳的學習效果。(2)學習向量量化的 codebook vector 個數多寡，影響其分類正確率，但設定過多的 codebook vector 個數，將增加計算成本。(3)促使內容過濾產生的推薦成功因素，取決於各個使用者對電影高低評分筆數的多寡。如使用者給予評分越少，則愈不容易提供適切地推薦結果，例如給予高評分（4-5 分）的電影很少，則不易將認同度高的電影推薦給使用者；因此，LVQ 正確分類，有較小的 MAE，對推薦結果影響亦有限。(4)影響協同過濾成功推薦的因素，主要為使用者鄰居個數 K 的設置。選擇較大的 K 值，會有較好的精確度與召回率。但過多的鄰居數，也將增加計算成本，需要更多時間進行相似度計算。因此，鄰居數目也需要不斷測試以尋得最佳推薦結果。

本研究以 LVQ 結合使用者對電影的評分與電影屬性（演員、導演、電影類型），學習使用者喜好，產生推薦內容，由於僅以小樣本做為測試資料，雖然產生較低的精確率，但再藉由協同過濾運用使用者對電影的評分找出相同喜好的使用者群集，推薦結果依然亮眼。實驗發現本研究提出的交換策略混合過濾架構，確實改善內容過濾和協同過濾演算法各自不足之處。往後研究者欲結合神經網路與協同過濾等研究時，對於策略的選擇，可參考本論文分析的意見，進行其他相關實驗。可參考本研究之分析。此外，在內容式過濾類神經網路的學習，若能加入更多種電影屬性，例如電影的文字描述（如劇情、電影標題...等屬性），提供神經網路學習，可望能夠進一步提升神經網路分類的正確率。

## 參考文獻

- 葉怡成（2003），*類神經網路模式應用與實作*，儒林圖書有限公司，臺北市。
- Acilar, A.M. and Arslan, A. (2009), 'A collaborative filtering method based on artificial immune network', *Expert Systems with Applications*, Vol. 36, No. 4, pp. 8324-8332. doi: 10.1016/j.eswa.2008.10.029
- Adomavicius, G. and Tuzhilin, A. (2005), 'Toward the next generation of recommender

- systems: a survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734-749.
- Ahn, H.J. (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. doi: 10.1016/j.ins.2007.07.024
- Balabanović, M. and Shoham, Y. (1997), 'Fab: content-based, collaborative recommendation', *Communications of the ACM*, Vol. 40, No. 3, pp. 66-72. doi: 10.1145/245108.245124
- Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A. and Peleteiro, A. (2010), 'A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition', *Information Sciences*, Vol. 180, No. 22, pp. 4290-4311. doi: DOI: 10.1016/j.ins.2010.07.024
- Breese, J.S., Heckerman, D. and Kadie, C. (1998), 'Empirical analysis of predictive algorithms for collaborative filtering', *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI-98)*, Madison, Wisconsin, July 24-26, pp. 43-52.
- Burke, R. (2002), 'Hybrid recommender systems: survey and experiments', *User Modeling and User-Adapted Interaction*, Vol. 12, No.4, pp. 331-370. doi: 10.1023/A:1021240730564
- Burke, R. (2007), 'Hybrid web recommender systems', in Brusilovsky, P., Kobsa, A. and Nejdl, W. (Eds.), *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer-Verlag Berlin, Heidelberg, pp. 377-408.
- Callander, S. (2011), 'Searching and learning by trial and error', *The American Economic Review*, Vol. 101, No. 6, pp. 2277-2308.
- Chou, P.H., Li, P.H., Chen, K.K. and Wu, M.J. (2010), 'Integrating web mining and neural network for personalized e-commerce automatic service', *Expert Systems with Applications*, Vol. 37, No. 4, pp. 2898-2910.
- Christakou, C. and Stafylopatis, A. (2005), 'A hybrid movie recommender system based on neural networks', *Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications (ISDA '05)*, Wroclaw, Poland, Sept 8-10, pp. 500-505.
- Chuan, Z., Xianliang, L., Mengshu, H. and Xu, Z. (2005), 'A LVQ-based neural network anti-spam email approach', *SIGOPS Operating Systems Review*, Vol. 39, No. 1, pp. 34-39. doi: 10.1145/1044552.1044555

- Das, A.S., Datar, M., Garg, A. and Rajaram, S. (2007), 'Google news personalization: scalable online collaborative filtering', *Proceedings of the Sixteenth international conference on World Wide Web (WWW2007)*, Banff, Alberta, Canada, May 8-12, pp. 271-280
- de Campos, L.M., Fernández-Luna, J.M., Huete, J.F. and Rueda-Morales, M.A. (2010), 'Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian networks', *International Journal of Approximate Reasoning*, Vol. 51, No. 7, pp. 785-799. doi: DOI: 10.1016/j.ijar.2010.04.001
- Drachsler, H., Hummel, H.G.K. and Koper, R. (2008), 'Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model', *International Journal of Learning Technology*, Vol. 3, No. 4, pp. 404-423. doi: 10.1504/ijlt.2008.019376
- Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. (1992), 'Using collaborative filtering to weave an information tapestry', *Communications of the ACM*, Vol. 35, No. 12, pp. 61-70. doi: 10.1145/138859.138867
- Gray, R. (1984), 'Vector quantization', *IEEE ASSP Magazine*, Vol. 1, No. 2, pp. 4-29.
- Kamahara, J., Asakawa, T., Shimojo, S. and Miyahara, H. (2005), 'A community-based recommendation system to reveal unexpected interests', *Proceedings of the Eleventh International Multimedia Modelling Conference (MMM 2005)*, Melbourne, Australia, Jan 12-14, pp. 433-438.
- Khuwaja, G.A. (2003), 'LVQ base models for recognition of human faces', *International Journal of Computer Applications in Technology*, Vol. 16, No. 4, pp. 181-193.
- Kohonen, T. (1986), *Learning Vector Quantization for Pattern Recognition*, Helsinki University of Technology, FINLAND.
- Kohonen, T. (2001), 'Learning vector quantization', *Self-Organizing Maps*, Vol. 30, pp. 245-261.
- Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J. and Torkkola, K. (1996), *LVQ PAK: the Learning Vector Quantization Program Package*, Helsinki University of Technology, FINLAND.
- Lang, K. (1995), 'NewsWeeder: learning to filter netnews', *Proceedings of the 12th International Conference on Machine Learning (ICML95)*, Lake Tahoe, CA, July 9-12, pp. 331-339.
- Lekakos, G. and Caravelas, P. (2008), 'A hybrid approach for movie recommendation', *Multimedia Tools and Applications*, Vol. 36, No. 1-2, pp. 55-70. doi:

- 10.1007/s11042-006-0082-7
- Li, Q. and Kim, B.M. (2003), 'An approach for combining content-based and collaborative filters', *Proceedings of the sixth international workshop on Information retrieval with Asian languages*, Sapporo, Japan, July 7, pp. 17-24.
- Linde, Y., Buzo, A. and Gray, R. (1980), 'An algorithm for vector quantizer design', *IEEE Transactions on Communications*, Vol. 28, No. 1, pp. 84-95.
- Marovic, M., Mihokovic, M., Miksa, M., Pribil, S. and Tus, A. (2011), 'Automatic movie ratings prediction using machine learning', *Proceedings of the 34th International Convention on Information and Communication Technology, Electronics and Microelectronics ((MIPRO 2011))*, Opatija, Croatia, , May 23-27, pp. 1640-1645.
- Martín-Valdivia, M.T., García-Vega, M. and Ureña-López, L.A. (2003), 'LVQ for text categorization using a multilingual linguistic resource', *Neurocomputing*, Vol. 55, No. 3-4, pp. 665-679. doi: 10.1016/s0925-2312(02)00633-1
- Martín-Valdivia, M.T., Ureña-López, L.A. and García-Vega, M. (2007), 'The learning vector quantization algorithm applied to automatic text classification tasks', *Neural Networks*, Vol. 20, No. 6, pp. 748-756. doi: 10.1016/j.neunet.2006.12.005
- Mizuno, Y., Mabuchi, H., Chakraborty, G. and Matsuhara, M. (2010), 'Clustering of EEG data using maximum entropy method and LVQ', *Proceedings of the 10th WSEAS international conference on Systems theory and scientific computation (ISTASC'10)*, Taipei, Taiwan, August 20-22, pp. 647-163.
- mludwig. (2011), 'MovieLens Data Sets | GroupLens Research', available at <http://www.grouplens.org/node/73> (accessed 23 February 2012)
- Mooney, R.J. and Roy, L. (2000), 'Content-based book recommending using learning for text categorization', *Proceedings of the fifth ACM conference on Digital libraries (ACM DL 2000)*, San Antonio, Texas, June 2-7, pp. 195-204.
- Putnam, H. (1965), 'Trial and error predicates and the solution to a problem of Mostowski', *The Journal of Symbolic Logic*, Vol. 30, No. 1, pp. 49-57.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994), 'GroupLens: an open architecture for collaborative filtering of netnews', *Proceedings of the 1994 ACM conference on Computer Supported Cooperative Work (CSCW '94)*, Chapel Hill, North Carolina, October 22-26, pp 175-186.
- Sarwar, B., Karypis, G., Konstan, J. and Reidl, J. (2001), 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the Tenth international conference on World Wide Web (WWW10)*, Hong Kong, China, May 1-5, pp.



- 285-295.
- Sung, H.H. (2006), 'Digital content recommender on the Internet', *IEEE Intelligent Systems*, Vol. 21, No. 2, pp. 70-77.
- Swanson, D.R. (1977), 'Information retrieval as a trial-and-error process', *The Library Quarterly: Information, Community, Policy*, Vol. 47, No. 2, pp. 128-148.
- Symeonidis, P., Nanopoulos, A. and Manolopoulos, Y. (2008), 'Providing justifications in recommender systems', *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 38, No. 6, pp. 1262-1272.
- Umer, M. and Khiyal, M. (2007), 'Classification of textual documents using learning vector quantization', *Information Technology Journal*, Vol. 6, No. 1, pp. 154-159.
- Yong, D., Zhonghai, W., Cong, T., Huayou, S., Hu, X. and Zhong, C. (2010), 'A hybrid movie recommender based on ontology and neural networks', *Proceeding of the IEEE/ACM International Conference on Green Computing and Communications, and International Conference on Cyber, Physical and Social Computing (GreenCom-CPSCOM 2010)*, Hangzhou, China, December, 18-20, pp. 846-851.
- Yu, P.S. (1999), 'Data mining and personalization technologies', *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications (DASFAA' 99)*, Hsinchu, Taiwan, April 19-21, pp. 6-13.