推薦系統中相似度評估的改良

Improvement of similarity assessment in recommendation systems

何子青
輔仁大學資訊工程學系
e-mail：jimmy82513@gmail.com

郭文彥
輔仁大學資訊工程學系
e-mail:wykuo@csie.fju.edu.tw

摘要

在我們的生活當中，推薦系統成功地運用在許多地方上，像是音樂、電影、書籍等。而也有許多論文不斷地探討找出新的演算法，為了就是要得到更高的準確度。而現在的推薦系統中，非機器學習的方式最被廣泛運用的為協同過濾式(Collaborative Filtering, CF)。而協同過濾的主要方式為透過每一個使用者之間，去尋找出他們之間的相似程度。並且再透過比較相似的使用者的評分，去預測出使用者尚未評分的項目。

而這篇論文當中我們也是使用協同過濾式的推薦系統，我們提出的方法當中主要對於找出使用者之間的相似程度上，將一般常使用的皮爾森相關係數(Pearson' s Correlation Coefficent ,PCC)[1]的演算法作一些改良。我們將在相似度的計算上面加權了使用者與使用者之間的共同項目去改善平常皮爾森相關係數的缺點。在本篇論文當中所使用的實驗資料來源為MovieLens[2]的電影資料庫，我們會把我們所提出的方法跟其他的相似度測量方式去做比較。像是MAE、RMSE等來比較出我們所提出的方法在準確率以及預測分數的誤差值比其他方法更好。並且會在不同的鄰居數目中去做比較。

關鍵字：推薦系統、協同過濾、評價預測、皮爾森相關係數

Abstract

Recommendation systems have been successfully used in many places, such as music, movies, books, and so on. There are many papers that constantly explore new algorithms, in order to get higher recommendation accuracy. In current recommendation systems, the most widely used non-machine learning method is Collaborative Filtering (CF). The main way of collaborative filtering is to find out the similarity between every user. And by comparing the ratings of similar users, to predict the score of the items that the user has not yet scored. Then, the difference between the predicted score and the actual score is measured to judge the accuracy of the prediction.

In this paper, we also use collaborative filtering. The method we propose is mainly to find the similarity between users, using the commonly usedPearson'sCorrelation Coefficent (PCC) with some improvements. We weight common projection between the user and other users on similarity calculation to mend the shortcomings of the Pearson correlation coefficient. The experimental data used in this paper is from MovieLens's film database. We will compare our proposed methods with other similarity measures. The results of the experiments will be compared by the commonly used scoring mechanisms, such as MAE, RMSE, etc., to compare the accuracy of the proposed method with the other methods. And we will compare results with different number of neighbors. We can find that when the number of neighbors is greater than 10, our method will be better than other methods.

1. 前言

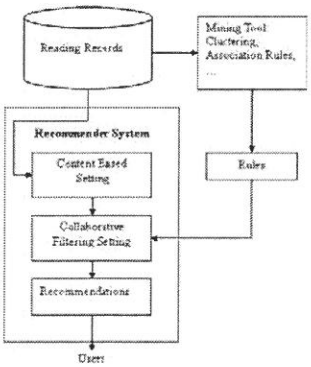
隨著現代網際網路越來越發達的狀況下，使用者面對各式各樣的資訊量大增。為了不讓使用者對於資訊過載(Information Overload)的狀況發生，產生了推薦系統(Recommendation System)。推薦系統可以從使用者的興趣、習慣，為使用者過濾出使用者不需要的資訊，並按照使用者的偏好，推薦使用者可能的感興趣的物品，讓使用者找尋相關商品或是服務時有更好的效率。推薦系統目前也應用在很多方面的，像是書籍、電影、音樂、新聞。而在這個資訊發達的世代更是有不少推

薦系統應用在電子商務的環境中，像是:Netflex、Amazon。而我們想在推薦系統上不管是在準確率又或者是在反應時間上有更好的表現。

雖然現在機器學習的方式可以讓準確率提高一些，可是在學習得過程中以及為資料做的前處理所花費的時間來的很長，於是我們決定使用比較早期尋找最佳鄰居的方式來去做推薦系統。但是尋找最佳鄰居的過程中，發現在計算使用者與使用者之間的相似度中，我們看到了一些缺點。於是，我們進一步的去將我們所發現的缺點去做改善，並希望能夠間接的提升我們推薦系統的準確率。

2. 相關研究

推薦系統依據使用者的偏好或特徵，幫助使用者在眾多且雜亂的資訊中過濾出有用之資訊，或是找出像類似的群組，並且依據使用者的喜好、興趣、行為或需求，推薦出使用者潛在需求的資訊、服務 或產品如下圖一所示[3][4]。



圖一、推薦系統流程圖

我們會對於推薦系統常用的三種技術來做介紹，分別是內容導向式篩選、協同過濾式以及混合式系統。

2.1 內容導向式篩選

基於內容的過濾建議的項目與用戶過去喜歡的項目類似。然而，它不同於協作過濾，通過根據項目的內容(例如標題，年份，描述)推導項目之間的相似性，而不是其他使用者對於物品的評價。例如，如果用戶喜歡“哈利波特：神秘的魔法石”和“哈利波特：消失的密室”，然後使用標題中的詞語，推薦人可以建議“哈利波特：火盃的考驗”。或者是在基於項目內容的過濾中，描述每個項目的豐富信息被假定為以特徵向量（y)(例如標題，年份，描述）的形式可用。以“哈利波特：神秘的魔法石”為例，特徵向量可以為:(J.K.Rowling 、魔法、2001、小說)等。

這些特徵向量用於創建用戶偏好的模型。可以使用各種信息檢索（例如 TF-IDF）和機器學習技術（例如，樸素貝葉斯，支持向量機，決策樹等）來生成基於其可以生成推薦的用戶模型。

2.2 協同過濾

協同過濾簡單來說就是利用興趣相同、擁有共同經驗的群體喜好來推薦使用者可能感興趣的資訊。使用者可以透過評分並記錄下來已達到過濾的目的進而幫助別人篩選資訊。協同過濾式推薦系統最早於 1992 年由 Goldberg 等學者所提出的 Tapestry 系統[6][7]，而比較後期的協同過濾又可以分成兩大類，分別為 Memory-Based CF 以及 Model-Based CF。

(1). Memory-Based CF: 這一種類型的協同過濾會依據使用者的評分紀錄，再與其他使用者的評分紀錄去找出兩者之間的相似度。再將與使用者相似度高的其他使用者們聚集起來，並依據他們評分去進行推薦。而找尋相似使用者最常用的方法是尋找最佳鄰居方法(Find Nearest Neighbor)。而最後 Memory-Based CF 還可以再細分為兩種類型，分別為(a). User-Based (b). Item-Based[7][8]

(a). User-Based: 用相似統計的方法得到具有相似愛好或者興趣的相鄰使用者，所以稱之為以使用者為基礎（User-based）的協同過濾或基於鄰居的協同過濾(Neighbor-based Collaborative Filtering)。

(b.) Item-Based: 以使用者為基礎的協同推薦演算法隨著使用者數量的增多，計算的時間就會變長，所以在 2001 年 Sarwar 提出了基於項目的協同過濾推薦演算法[8]。以項目為基礎的協同過濾方法，透過計算項目之間的相似性來代替使用者之間的相似性。

(2).Model-Based: 以使用者為基礎(User-based)的協同過濾和以項目為基礎(Item-based)的協同過濾系統稱為以記憶為基礎(Memory based)的協同過濾技術，他們共有的缺點是資料稀疏，難以處理大資料量影響

即時結果，因此發展出以模型為基礎的協同過濾技術。以模型為基礎的協同過濾(Model-based Collaborative Filtering)是先用歷史資料得到一個模型，再用此模型進行預測。以模型為基礎的協同過濾廣泛使用的技術包括 Latent Semantic Indexing、Bayesian Networks…等，根據對一個樣本的分析得到模型。而現在除了這些以外，還會利用機器學習的方式來訓練模型，讓準確率更高。

2.3 混合式

當所謂的混合式推薦系統(hybrid approach)[9]就是使用兩種或以上推薦系統即為混合式推薦系統。混合式推薦系統可針對問題的特性，結合不同推薦系統的優點以彌補各推薦系統在特定問題處理上的盲點與不足之處。

2.4 預測方式評估

這一節當中，我們會介紹我們如何評估目推薦系統的好壞。評估方面我們主要會介紹三種方法，分別為 MAE、RMSE、ROC 曲線。

(1).MAE (Mean Absolute Error): 在推薦系統當中，最常被拿來做評估的方式就是 MAE。MAE 將所有預測的評分跟原本的評分做相減並且求出誤差後做總和。MAE 值越小代表預測越準確，反之相反。

(2)RMSE(Root Mean Square Error): RMSE是預測分數與原始分數誤差值的平方根。在實際測量中 RMSE 當對某一值進行甚多次的測量時，取 RMSE 可以反應測量數據偏離真實值的程度。RMSE 越小，則代表預測越準確，反之則相反。

(3).ROC 曲線：最早 ROC 是應用於訊號處理領域，目的是在於評估一個過濾器（information filtering system）區分訊號（signal）與雜訊（noise）的能力。分析 ROC 曲線之前，首先要定義所謂好壞標準（relevant/irrelevant）的界線（threshold），如同計算 Precision-Recall 時一樣，relevant/ irrelevant 的定義方式是二元的，與 Reversal Rate 使用高、低）兩個參考數值的作法不同。有了 relevant/irrelevant 的定義之後，我們就可以做出有四個元素的 confusion matrix （消費者判定 relevant/irrelevant ，和推薦系統預測 relevant/irrelevant，剛好是四種組合）。如下圖

	actual positive	actual negative
predicted positive	TP	FP
predicted negative	FN	TN

(a) Confusion Matrix

Recall = $\frac{TP}{TP+FN}$

Precision = $\frac{TP}{TP+FP}$

True Positive Rate = $\frac{TP}{TP+FN}$

False Positive Rate = $\frac{FP}{FP+TN}$

(b) Definitions of metrics

圖二、ROC 曲線

Accuracy: (TP+TN) / (P+N)

Recall = TP/P

2.5 協同過濾相似度計算

根據上面所述，我們可以知道如何判別優秀的推薦系統當中，如何去找出使用者跟使用者之間的相似度是很重要的。接下來我們會介紹一些常見相似度的方法[5]。

2.5.1 餘弦相似度

餘弦相似度主要是將使用者的評分視為一條 n 維向量，並將使用者之間的向量夾角呈現出使用者之間的相似度。餘弦相似度所算出的相似度的值會落在 0 到 1 之間，如果兩位使用者越相似的話值會越靠近 1，如果使用者越不相似則值會越接近 0。餘弦相似度的公式如下：

$$CosSim(i,\ j)=\frac{\sum_{s\in S_{i,j}}(r_{i,s})(r_{j,s})}{\sqrt{\sum_{s\in S_{i,j}}(r_{i,s}^{\ 2})(r_{j,s}^{\ 2})}}$$

$S_{i,j}$: 使用者 i 跟使用者 j 評價物品的交集。

$r_{i,s}, r_{j,s}$: 使用者 i 跟使用者 j 對於物品 s 的評價分數。

2.5.2 皮爾森相關係數

在推薦系統當中，最常被拿來使用的相似度方法為皮爾森相關係數。透過皮爾森相關係數所產生的相似度會落在 1 到-1 之間。其中越接近 1 就代表相關性越大，越靠近-1 就代表相關性低，0 表示沒有什麼關聯。皮爾森相關係數的公式如下：

$$PccSim(i,j)=\frac{\sum_{s\in S_{i,j}}(r_{i,s}-\bar{r_i})(r_{j,s}-\bar{r_j})}{\sqrt{\sum_{s\in S_{i,j}}(r_{i,s}-\bar{r_i})^2(r_{j,s}-\bar{r_j})^2}}$$

$S_{i,j}$: 使用者 i 跟使用者 j 評價物品交集。

$r_{i,s}, r_{j,s}$: 使用者 i 跟使用者 j 對於物品 s 的評價分數。

$\bar{r}_i, \bar{r}_j$: 使用者 i 跟使用者 j 的平均評價分數。

2.5.3 傑卡德相似度

傑卡德相似度的計算只考量兩個使用者共同評分的項目數量而已，

所以相似度的範圍為 0 到 1，可以理解成使用者們沒有交集或者是交集一樣。因此他們計算相似度的公式如下：

$$\mathrm{JaccSim}(x,y) = \frac{|\mathrm{User}_i \cap \mathrm{User}_j|}{|\mathrm{User}_i \cup \mathrm{User}_j|}$$

傑卡德的問題很明確，因為只考慮使用者之間的共同評分項目數量來做計算，所以對於使用者可能比較感興趣或是比較不喜歡的東西就無法表示出來。

2.5.4 歐幾里得距離相似度

歐幾里德距離相似度為所有相似度當中最直覺最容易理解的方法。它以使用者們評價的物品以及評分分數形成一條 n 維向量，並計算他們之間的直線距離。歐幾里德相似度的範圍為 0 到 1 ，使用者之間的相似度越大，距離越小。公式如下：

$$\mathrm{Distance}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$\mathrm{EuclidSin}(x,y) = \frac{1}{(1+\mathrm{Distance}(\mathrm{User}\ x, \mathrm{User}\ y))}$$

3. 方法設計

在第一章跟第二章當中，我們可以看到在推薦系統中不管是協同過濾或者是內容導向式的篩選，尋找相似的使用者是很重要的，這樣才能夠給予準確的推薦。尋找相似的使用者方法有很多種，像是第二章所提到的餘弦相似度、歐幾里得距離相似度、皮爾森相關係數，也有詳細介紹過他們的優缺點。本論文所使用的方式為皮爾森相關係數，使用這個計算相似度的方式是因為皮爾森係數在對於推薦引擎以及資料探勘中是最常使用的，而準確率也相當高。可是我們在使用上卻發現這個相似度計算中在特定的情況下會有缺點，像是在不同使用者當中，因為樣本數的不同卻會導致出一樣的相似度。所以我們決定加以改進[10][11]。

3.1 系統流程

協同過濾推薦系統可以簡單分成四個步驟，分別為：

1. 讀取使用者評分資料 2. 計算出使用者之間的相似度 3. 尋找相鄰的鄰居 4. 推薦分數。而接下來我們會做詳細的介紹。

測試資料中包含使用者對於電影的評分資料，而這一份資料中會有使用者對於每一部電影所打的一個評分分數，而我們將每一位使用者對於每一部電影的評分分數做整合，合併成一個使用者X電影的分數矩陣，如表五。

表五、使用者評分矩陣

user id / movie id	1	2	3	4	…	943
1	2	4		4	…	
2		3			…	
…	…	…	…	…	…	…
1643	1		3	2	…	1

而我們將每一位使用者所評分的分數視為一條使用者向量，以上面表格為例：使用者 u 的分數向量則為 USER u:[2, 4, 0, 4, …, 0]。其中向未評分的部分我們一率用 0 做表示。之後再將其它的使用者也視為向量，所以我們最後會得到每一位使用者自己的向量表示，而接下來我們將會對於這些向量去做相似度的計算，計算方式我們先以皮爾森相關係數為主，一方面也是因為這是我們之後要優化的演算法。

再來，我們將每個對於剩下其他人的相似度存在一個矩陣，矩陣的大小為 USER X USER。USER 為所有使用者的數量，有了這個矩陣之後，接下來我們會給訂一個最佳鄰居值 K，而這個 K 值我們會去尋找說對於每一個使用者跟他最相似的 K 位使用者，並且透過這 K 位試用者的評分資料去預測電影的評分分數。

3.2 研究方法

而在實作的過程中我們發現因為皮爾森相關係數只關注兩個用戶對它們評價的項目的評分，而沒有考慮到他們評價的其他項目。更重要的是，相似性可能並不總是反映真實的相關性，因為當兩個用戶對不同數量的項目進行評分時，它們可以給出相同的相似性分數。舉個例子來說，當今天有三位使用者 u,y,m。而使用者 u 以及使用者 y 一起共同評分了 a 個項目，相似度為 Sim(u,y)=s；而另外一組使用者 u 與使用者 m 一起共同評分了 b 個項目，相似度 Sim(u,m)=s，其中 a>b。在這個例子當中，雖然我們發現在皮爾森相似度中兩者的相似度值是一樣的，可是我們可以發現相對於(u,m)這兩位使用者，(u,y)的相似度更加讓人信任，因為他們共同評分的項目來的更多，更有說服力。所以，我們覺得由兩個用戶評分的共同項目的數量應該納入相似度以反映更好的相似性。

於是我們接著又想到說是否可以將每個使用者的因素給摻入進去。所以我們設了兩個一維矩陣叫一個叫做 userCount[]，長度為所有的使用者數目，在這個矩陣當中會記錄每一個使用者總共評分了多少項目；而另外一個叫做 threshold[]去計算當下使用者跟其他使用者平均共同評分數目。有了這兩個數值後接下來我們就開始去加權原本的相似度算法。

而我們如何去算出新的相似度呢?我們會分成兩種情形。因為本篇論文主要在改進信任度較低的相似度，那怎麼樣才算是相似度較低呢?這時候我們就可以使用到剛剛的 threshold 的矩陣。如果兩個使用者的共同評分數目都大於使用他們分別的平均評分數目的話，我們將會視為這兩個使用者是可信任的來源。

If(k>=threshold[u] && k>=threshold[v])

K:兩個使用者 u,v 的共同評分數目。

Threshold[u]:使用者 u 對於其他使用者的平均共同評分數目。

Threshold[v]:使用者 v 對於其他使用者的平均共同評分數目。

如果上述條件達成的話，我們一樣按照原來的皮爾森相關係數去做相似度的計算，但如果沒有達成的話，我們就會將皮爾森相關係數的相似度作加權並且產生了新的相似度算法 New Similarity(NSim)。這樣一方面可以保留住皮爾森相關係數的優點，也可以對於它的缺點去做改進。而加權後的公式如下:

$$NSim(u,v)=\frac{P_u}{usercount[u]}sim(u,v)+\frac{P_v}{usercount[v]}sim(u,v)$$

K:兩位使用者 u,v 的共同評分數目。

Pu =|k-userCount[u]|

Pv =|k-userCount[v]|

有了上面的公式之後，我們可以對我們的資料去做實驗。我們的資料來源以及實驗的進行還有結果都會在第四章去做介紹，並加上實驗數據去證實我們所提出來的方法比起之前傳統的相似度算法來得比較好。

4. 實驗結果

在這一章當中，我們會將我們提出的方法與其他的相似度預測方法去做比較，來證明我們所提出的方法得出來的預測分數更加精準。在 4.2 節我們會介紹我們的測試資料來源以及我們使用工具。接下來在 4.3 節我們會分成兩個部分來做介紹，第一個部分我們會將所提出的方法跟餘弦相似度、皮爾森相關係數、傑卡爾相似度等來做比較，比較的方式則是透過 MAE。接著第二個部分我們則是會將所有的使用者做一個偽 ROC 的電影推薦，並再觀察這些我們推薦的電影是否原本就符合我們使用者的喜好，而算出一個推薦的成功的機率。

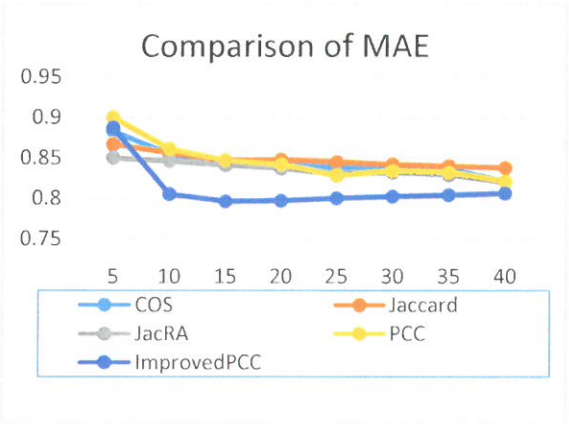
4.1 資料來源以及使用工具

在這一次實驗當中，我們所使用的資料來源是 MovieLens。MovieLens 是一個開放式的電影評價資料庫。我們這一次所使用的數據集為 ml-100k。其中裡面包含了 943 名使用者對於 1682 部電影的評價，而選定的用戶對至少 20 部電影進行了評分，每個用戶都由一個 ID 代表。

4.2MAE 以及 RMSE

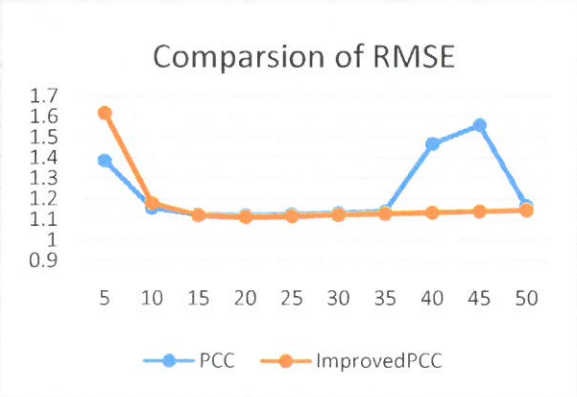
在這一節，我們將會將我們所提出的方法跟餘弦相似度、皮爾森相關係數、傑卡爾相似度來做比較，比較的方式為 MAE 跟 RMSE。而在實驗當中，我們會比較在不同的推薦鄰居數目當中，每一種相似度的 MAE 跟 RMSE 值會是多

少。而在推薦鄰居數目當中，我們給訂了範圍從 5 開始到 40，並且每五個人為一個級距。而鄰居數目之所以只到 40 的原因在於人數到了 30-40 人時，MAE 值會逐漸平穩並且差異不大如圖三。



圖三、實驗比較數據圖

從圖三可以發現，在一開始 K 從 0-5 的時候，跟其他的方法比較起來還不是那麼的好，可是當 K>5 之後我們的 MAE 值有著明顯的下降，而在 K=15 的時候可以發現我們方法的 MAE 值來到了最小，代表當鄰居人數在 15 人的時候，我們的效能最佳。



圖四、實驗二數據比較圖

接著我們特別再將我們提出的方法以及皮爾森相關係數的演算法算出 RMSE 的值。可以看到在鄰居數目比較少的情況我們提出的方法的值比較高以外， 之後的情形我們都贏過皮爾森相關係數。也達到了一開始我們所提出想要優化皮爾森相關係數的目的。但有一些小缺點是我們所提出的方法中，可以發現在鄰居數目比較小的時候會很不穩定準確率不高，但鄰居數目只要超過 10 個左右後就會比較穩定。

4.3ROC 曲線

在這一小節當中，我們嘗試來使用類似 ROC 曲線的做法。因為一般來說 ROC 曲線是用在於 Model-Based 的推薦系統上。並且可以將推薦結果分成四種類型，分別為：TP(推薦系統推薦，使用者本身推薦)、TN(推薦系統推薦，使用者不推薦)、FP(推薦系統不推薦，使用者推薦)、FN(推薦系統不推薦，使用者不推薦)。而為了想讓我們的實驗更生活化一點，想知道說我們的推薦系統是否可行，於是我們做了一個類似 ROC 曲線的實驗。而我們的執行的方式則是會利用我們所預測的分數當作系統是否推薦，而比較的標準則是以使用者的喜好為主。而使用者的喜好我們就會用每位使用者的平均評分分數(以四捨五入做處理)來做代替。所以最後我們最後的結果一樣可以分成四個部分，可分為：

TP(推薦系統預測分數>=使用者平均評分 && 使用者原始評分>=使用者平均評分分數)

TN(推薦系統預測分數>=使用者平均評分 && 使用者原始評分<使用者平均評分分數)

FP(推薦系統預測分數<=使用者平均評分 && 使用者原始評分>=使用者平均評分分數)

FN(推薦系統預測分數<=使用者平均評分 && 使用者原始評分<使用者平均評分分數)

在第二章的 2.4 小節 ROC 曲線的部分介紹的公式 Accuracy 以及 Recall，計算之後，發現我們所提出來的方法算出來的數值可以算出 Accuracy=0.7055 以及 Recall=0.99961。代表說我們推薦系統的準確率可以達到 70%，是個不錯的數字。其中 Recall 達到九成九可以說明我們推薦系統不管是在推薦或是不推薦的準確率上是很高的。

5. 結論與未來展望

在本篇論文當中，我們對於皮爾森相關係數相似度算法做優化，而成效在第四章的實驗結果中可以發現準確率比起其他的相似度算法有明顯的提升。可是對於協同過濾法本身的某些缺點還是無法改進，像是冷起始問題以及資料稀疏性的問題。如果能夠解決資料稀疏性的問題的話，相信計算出使用者以及使用者之間的相似度上會有更好的準確率，並且在預測上會更精準，誤差值來得更小。

另一方面，可以去嘗試結合機器學習的方式，像是透過深度學習的方式去找尋使用者與使用者之間的相連性，又或者是架出一個混合式的架構，像是同時算出使用者以及使用者之間的相似度再加上電影與電影之間的相似度來去做預測，相信對於預測的準確率會再往上提昇。

參考文獻

[1] Wu, X., Huang, Y. and Wang, S. (2017), "A New Similarity Computation Method in Collaborative Filtering Based Recommendation System," *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1-5, 2017.

[2] MovieLens datasets, https://grouplens.org/datasets/movielens/

[3] Hsu, Mei-Hua., "A personalized English learning recommender system for ESL students. " *Expert Systems with Applications Volume. 34, Issue: 1*, pp. 683-688, 2008.

[4] 李建宏，「可改善推薦系統評價預測之使用者分群方法研究」，國立中正大學電機工程系碩士班碩士論文，pp.1-73，2013.

[5] J.A. Hyung, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences 17837–51*, 2008.

[6] Goldberg, D., Nichols, D., Oki, B. M., andTerry D., "Using Collaborative Filteringto Weave an Information Tapestry." *Communication of the ACM*, pp.61-70, 1992.

[7] R.Burke, " Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction archive,'' *Volumn 12 Issue 4*, pp. 331-370, November 2002.

[8] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *in Proceedings of the International Conference on the World Wide Web*, pp. 285–295, 2001.

[9] S. Spiegel, "A Hybrid Approach to Recommender Systems based on Matrix Factorization," *Technical University Berlin, Department for Agent Technologies and Telecommunications*, 2009.

[10] X. Wu, B. Cheng, J. Chen, "Collaborative Filtering Service Recommendation Based on a Novel Similarity Computation Method," *IEEE Transactions on Services Computing*, 2015.

[11] SAMIYAH AL-ANAZI, PANDIAN VASANT, M. ABDULLAH-AL-WADUD, "An Improved Similarity Metric for Recommender Systems," *International Journal of Computers*, pp. 154-157, 2016.