一個改善音樂推薦系統冷啟始問題之方法設計

潘雅婷
國立屏東科技大學
資訊管理系
研究生
m9756031@mail.npust.edu.tw

蔡肇彥
國立屏東科技大學
資訊管理系
研究生
m9856037@mail.npust.edu.tw

劉寧漢*
國立屏東科技大學
資訊管理系
副教授
gregliu@mail.npust.edu.tw

摘要

推薦系統普遍存在著冷啟始問題(Cold-Start problem)，即系統在不了解新進使用者的喜好情況下無法進行推薦，而新的項目沒有存在任何評分記錄也沒辦法被推薦。儘管不同的推薦系統有著各自的推薦機制來應對Cold-Start問題，一般的方法都是以隨機挑選的方式，例如系統隨機挑選幾首歌曲讓使用者聆聽，藉由使用者的回饋資料得知其音樂喜好，但是系統需要經過多次的詢答，累積一定的資訊才能夠正確的推薦音樂給使用者。若是系統可以平均的挑選不同類型的歌曲，即可快速地知道使用者的音樂喜好，因此為了減少Cold-Start的時間，本研究利用SOM方法將音樂分群再挑選出歌曲。由實驗結果得知SOM比k-means分群法、隨機法在挑選歌曲類型時還要平均，如此可以幫助系統減少Cold-Start的時間及提升系統的推薦準確率。

關鍵詞：自組織映射圖網路、音樂推薦、分群演算法。[1]

Abstract

Recommendation systems have a prevalent Cold-Start problem. The meaning of Cold-Start is that systems do not understand the new user's preferences, systems are not to recommend the music to users. In addition the new items are not to rate by anyone, they will not to be recommended. Although many recommendation systems have a solution to reduce Cold-Start, general systems utilize random to select songs. For example systems random select some songs to user so that systems will know user's preferences after they rated. But systems must cost much time to collect information so that they can recommend items to users. If systems select different type of music, they can quickly know the user's preferences. Therefore we will reduce the Cold-Start, we utilize SOM to select some songs from cluster. According to experiment, SOM select type of music more average than k-means and random. So SOM can improve Cold-Start problem and increase the precision.

Keywords: SOM, music recommendation, clustering algorithm

1. 前言

現代人的生活大多與網際網路脫離不了關係，以音樂為例，從過去的黑膠唱片、錄音帶、CD到線上音樂(online music)，這樣的演變方式讓人們越來越容易透過網際網路取得音樂資訊，不必受到時間與地點的限制，且有更多樣性的音樂選擇，不會再有只喜歡一兩首歌曲而購買整張專輯的窘境發生。這對於使用者而言是一大福利，但卻使得唱片市場景氣低靡，也造就了全球唱片業者不得不投入線上音樂這塊市場。

由於網路上充斥著大量的資訊與資料，產生資訊超載(information overload)的問題，就全球唱片市場而言，每年所發行的唱片不可勝數，更不用說地下樂團或是自己錄製的專輯等等，所以使用者要在眾多的音樂裡找出個人所喜愛的歌曲，常常需要花費大量的時間去處理與過濾，因此為了解決此類問題，相繼發展出推薦系統(Recommendation System)。

推薦系統可以從大量的資訊裡篩選出使用者有興趣的項目，並進而推薦給使用者。為了因應每位使用者的音樂喜好與需求有所不同，個人化音樂推薦也孕育而生，相較於傳統的音樂推薦系統，可以達到根據個別使用者的音樂喜好做推薦。因此如何達到個人化音樂推薦系統的有效率與準確率，是現在音樂推薦的一大重要議題。

* 為通訊作者

目前常見的音樂推薦系統主要有內容導向式過濾法(Content-Based Filtering)、合作式過濾法(Collaborative-Filtering)以及混合式推薦法(Hybrid Recommendation Method)。這些方法通常是從音訊資料裡擷取出特徵值，來計算音樂之間或者是使用者之間的距離，距離越短，即表示擁有越高的相似度，再經由不同的推薦機制，推薦合適的歌曲給使用者。

不同的推薦機制有各自的優缺點，大多是依據使用者的音樂喜好來做為推薦憑據，卻都有著相同的Cold-Start問題，即新的音樂與新進使用者。以合作式過濾法而言，當新的音樂被存入系統時，由於新的音樂沒有被使用者評分過，所以系統無法進行有效的推薦。而新進使用者在一開始進入系統時，由於系統並不知道新進使用者的音樂喜好，所以無法推薦音樂給使用者。同樣地，內容導向式過濾法也存在新進使用者的問題。故一般的應對方法是先隨機挑選幾首歌曲讓使用者試聽，使用者在聆聽系統所挑選的歌曲過後可以給予評分，得以讓系統瞭解使用者的音樂喜好，再根據評分結果推薦歌曲給使用者。

但是隨機挑選的方式會造成推薦不平均、準確率下降的現象產生。假設歌曲分佈如圖1所示，左邊的歌曲分佈較多也較密集，右邊則較少，當系統第一次挑選歌曲的時候，先挑到左邊較大集群的歌曲如圖1 (a)，系統第二次挑選歌曲時，同樣挑到左邊集群的歌曲如圖1(b)，這是因為左邊的集群歌曲分佈較密集也較大，故被挑中的次數就比較多，反觀右邊的歌曲則機率較小。若是遇到使用者剛好喜歡右邊集群的歌曲，系統也許需要花費大量的時間才會挑選到右邊集群的歌曲如圖1 (c)，因此推薦結果就不甚滿意也會影響到音樂推薦系統的準確率。

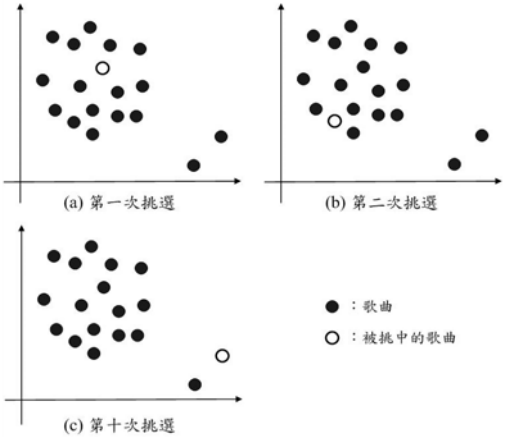

圖 1 歌曲分佈圖

由此可見，系統快速地知道使用者的音樂喜好，對於整個系統的推薦準確率能夠有效的提升，故本研究利用SOM(Self-Organizing Map)將音樂資料庫裡的音樂做群聚並從中挑選歌曲，再與k-means分群法、隨機法比較在抽樣的歌曲類型是否能滿足樣式之普遍性，藉此改善Cold-Start問題及提升系統的推薦準確率。

本文共分為五個章節，第二章為文獻探討，說明目前推薦系統的類型與相關文獻的討論；第三章為系統架構與方法，說明本研究所提出的方法；第四章分析實驗數據之結果；第五章結論，總結本研究所提出的論點與未來研究方向。

2. 相關文獻探討

2.1 推薦系統

推薦系統可以從大量的資訊如電影、音樂、書籍等等，篩選出使用者有可能感興趣的項目，進而推薦給使用者，以達到個人化的效果。Resnick and Varian[21]認為資訊過濾系統(Information Filtering System)也就是推薦系統，除了過濾資訊外會推薦使用者感興趣的項目，而且推薦的人與被推薦的人未必是認識的。Burke[7]說明任何一個系統可以產生個人化推薦的結果，或者是能夠在大量的選擇中，以個人化的方式幫助使用者找到有用或有興趣的資訊，皆可稱為推薦系統。

推薦系統的流程主要分為三個部分，首先系統需要先蒐集使用者的喜好資訊，再依據使用者的資訊進行篩選與推薦，最後使用者給予評價並回饋給系統。目前最為常見的音樂推薦系統主要有內容導向式過濾法、合作式過濾法以及混合式推薦法，以下將分別說明這三種推薦方法。

2.1.1 內容導向式過濾法

內容導向式過濾法是從資訊檢索(Information Retrieval)[22]和資訊過濾(Information Filtering)[5]的領域衍生而來，是擷取出項目的內容屬性當作特徵值，再比對特徵值之間的相似程度與關聯性，並推薦比對後相似程度較高的項目給使用者。如[3][27]首先會先建立使用者輪廓(User Profile)，在User Profile中儲存一些關鍵詞及相關資訊，以了解使用者的音樂喜好，例如使用者的基本資料、

喜歡什麼類型的音樂、演唱者等等，而每一首音樂的內容屬性可以當作不同的特徵值，例如歌曲名稱、演唱者、歌曲類型等，系統會依照使用者輪廓去比對並篩選出相類似的音樂做推薦。

內容導向式過濾法已經被多位學者拿來應用於不同的領域上，包括新聞、文件、商品、音樂等，較著名的系統有NewsWeeder[18]、InfoFinder[17]、SmartPad[19]、WebWatcher[2]。

知名音樂網站 Pandora 即以此方法幫助使用者找到喜愛的音樂，其所發展出來的音樂基因體計畫(Music Genome Project)，是分析每一首音樂的內容如旋律、和聲、節奏、樂器、歌詞等等屬性，擷取出將近四百個屬性，使用者只要在搜尋處輸入喜愛的歌曲名稱或演唱者，系統即會根據使用者的輸入找出最相似的歌曲，不管是眾所周知或是鮮為人知的歌曲，都能推薦給使用者，使用者在聆聽過後即可給予評價，這些評價資料會回饋給系統以作為下一次推薦的依據。

內容導向式過濾法的優點是根據使用者的歷史資料，去計算項目之間的相似程度，且不需要透過其他使用者的資料，即可推薦相似程度較高的項目。但缺點是無法推薦使用者尚未接觸過的項目、無法以文字表達的項目則無法推薦和使用者回饋資料不足等問題。

儘管此系統可以依照使用者的喜好作推薦，但是當遇到新進使用者時，即產生 Cold-Start 問題，因為系統在不了解使用者的音樂喜好情況下，會先隨機挑選幾個項目給使用者評分，再藉由使用者回饋資料得知其喜好，因此需要花費大量的時間去處理與過濾，才能找到使用者的真正喜好。

2.1.2 合作式過濾法

由具有共同興趣或相似行為的使用者所形成的個別社群，並利用社群裡其他成員對項目的評價做加權計算，來預測出使用者對此項目的喜好程度，因此又稱為社會過濾(Social Filtering)。

合作式過濾法是在 1992 年由Goldberg[11]等人首先提出的Tapestry系統，該系統主要是在過濾電子郵件，使用者利用TQL(Tapestry Query Language)查詢語法建立查詢，透過不同的查詢過濾出使用者有興趣的電子郵件。但缺點是使用者必須自行輸入查詢指令，系統才能依照使用者所輸入的指令，過濾出使用者所喜好的郵件，而並非由系統主動地針對使用者的喜好進行推薦。

Kostov[16]等人則說明合作式過濾法的必要步驟，首先系統會依照使用者的喜好，透過特定的分群演算法，將與自己具有相同喜好的其他使用者聚集在同一集群中，在此集群裡使用者對於其他每一位使用者都存在著一個相似度(similarity)，再把使用者之間的相似度套用到預測模組，即可算出使用者對某項目的喜好程度。較著名的系統有GroupLens[15]、Ringo[24]、Referral Web[13]、Siteseer[23]、Amazon。

Last.fm 為線上音樂推薦系統，其系統會記錄使用者所聽過的每一首歌曲，並搜尋資料庫裡其他相似的成員，以推薦使用者可能會感興趣的音樂，這些推薦清單會顯示在該網站所提供給使用者的個人網頁上，還包括了使用者的聆聽習慣等記錄與不同的分類標籤，使用者在聆聽音樂後一樣可以留下評分，以做為下一次推薦的依據，如果使用者對音樂或是樂手感到興趣，可以利用超連結看到相關資訊，也可以主動推薦給其他使用者。使用者的個人網頁除了顯示自己的播放清單外，還可以看到與自己興趣相似的其他成員與播放清單，這些設計都是為了讓使用者可以更清楚地了解自己的音樂喜好。

合作式過濾法與內容導向式過濾法的差別，主要在於合作式過濾法是經由使用者的喜好去找出相似度高的其他使用者來形成社群，並推薦這個社群裡其他成員所喜歡的項目給使用者，如Tzeng Y. S.[25]根據使用者已聽過的歌曲評價找出有共同興趣的社群成員，並推薦其他成員所評價過歌曲給使用者。合作式過濾法的優點是能夠過濾多媒體資料內容，且透過其他成員推薦可能會有未接觸過的項目產生，藉此挖掘出使用者的真正需求。儘管諸多系統使用合作式過濾的方法，但仍然存在一些缺點如Cold-start、資料稀疏、擴充性等問題。

目前已經有學者針對Cold-start問題而發展出不同的解決方法，如Ahn, H. J.[1]利用PIP (Proximity-Impact-Popularity)測量方法，幫助新進使用者在沒有評分記錄的情況下，有效的推薦電影給使用者。Poirier, D.[20]等人利用意見探勘(opinion mining)將使用者對於電影的評論進行分類，再搭配合作式過濾法推薦電影給使用者。Nouali, O.[26]以本體論(ontology)

為基礎,利用語意網(semantic web)的語意資料來描述使用者與電影的特性,並結合相似度測量方法找出相似電影作推薦。

2.1.3 混合式推薦法

混合式推薦法是由兩種以上的推薦方法所合併而成,因為每一種推薦方法都有不同的優缺點,若進行合併可以改善其缺點。常見的混合式推薦法是內容導向式過濾法及合作式過濾法的組合,由於合作式過濾法可以處理任何型態的項目,如多媒體資料,有別於內容導向式過濾法只能處理以文字為主的項目。且合作式過濾法是根據使用者對項目的評價去找到相似使用者,並推薦其它項目給有需求的成員,而系統所推薦的項目是跟使用者之前評價的項目截然不同,因此使用者可能會有意外的驚喜。同樣地,內容導向式過濾法解決了合作式過濾法的大部分問題,如 Sparsity Problem、Scalability Problem。

Beth L.[6]、Chen H. C.[8]使用以User Profile為基礎的方法,將使用者的過去喜好記錄與音樂內容特徵做分類,找出相似使用者後,推薦其喜好音樂類型給其他使用者,並計算使用者間的User Profile相似度,以產生相同社群的使用者推薦清單。

混合式推薦法較著名的系統有Fab[4]、Personal Tango[9]、ProfBuilder[28]、Smart Radio[12]、RAAP[10]。

由此可見,要做到個人化的推薦是目前音樂推薦系統的一大議題。而內容導向式過濾法是計算音樂之間特徵值的相似程度,並推薦相似程度較高的音樂,即使有新的音樂存入系統,也只需要擷取出音樂的特徵值,即可進行比對與篩選的動作,以達到個人化推薦的目的,因此本研究所使用的方法是屬於內容導向式過濾法。

2.2 音樂特徵值

在過去音樂資料是以人工的方式依演唱者、歌曲名稱、歌曲類別等項目分類,並建立音樂資料庫供使用者查詢與利用,但是當資料量日漸增多的情況下,要查詢音樂反而不是一件容易的事,況且我們常常只記得歌曲旋律,而不記得演唱者或是歌曲名稱,因此便有自動化分類及內涵式查詢(content-based retrieval)的系統產生。現今大部分的研究是從音樂資料內擷取出特徵值,因為這些特徵值足以代表音樂本身的特性,如節奏(rhythm)、旋律(melody)、和弦(chord)等,用來建立索引以便快速地找到音樂資料,故擷取音樂特徵值是一個相當重要的步驟。

音樂有不同的表示法,主要有音訊(audio signals)和符號兩種。音訊是指發聲體在震動時對空氣產生壓縮與伸張的效果,而形成聲波,可以利用過零率(Zero Crossing Rate)、梅爾倒頻譜係數(MFCC)、短時傅立葉轉換(Short Time Fourier Transforms)等方法從音訊中擷取出音高(pitch)、音量(volume)、音色(timbre)等特徵值。符號表示法如 MIDI (Musical Instrument Digital Interface,樂器數位介面),是一種電子通訊協定,讓具有 MIDI 裝置的數位器材互相傳遞音符與控制資料,這些資料不是聲音,而是用來對數位器材進行聲音表現的指揮控制,例如要彈哪個音、持續多久等,所以在 MIDI 檔案中可以得知每一個音符的音高、起始時間(onset time)跟音長(duration)。

2.3 分群演算法

本研究利用分群方法將音樂資料庫裡的音樂做群聚,為的是加快後面處理程序的速度,以達到快速找到使用者的真正音樂喜好目標。

資料分群是把相似程度高的資料分在同一群集(cluster)裡,使得組內相似程度高,組外相似程度低,這麼做是為了先過濾資料,以加快後面資料處理的速度。如果是遇到資料來源有限的情況,可以利用資料分群的結果做一些假設,以探索資料之間的相互關係。資料分群是依照資料之間的相似程度去做分群,而資料之間的相似程度在不同領域有不同的距離測量方法,歐幾里得距離(Euclidean Distance)則是最常被用來當作距離測量的一種方法,如下公式(1)所示。

$$d(a,\ b) = \sqrt{\sum_{i=1}^{k} (a_i - b_i)^2} \qquad (1)$$

分群演算法可區分為分割法(Partitioning method)、階層法(Hierarchical method)、密度法(Density-based method)、網格法(Grid-based method)與模式法(Model-based method),以下將分別做介紹。

2.3.1 分割法

分割法是將N個物件分割成K群，K值需要事先決定且要小於等於N，以確保每一群裡至少包含一個物件，而每一個物件皆屬於某一群。一開始先設定K值後，在初始分割結果中以反覆計算各資料點與群中心點的距離，藉以改善分群的結果，最後得到K個群集。此方法的優點是可以反覆修正結果，缺點是需要花費大量時間在計算所有可能的分群組合上，且分群後的形狀較趨向於圓形群集，常見的演算法如k-means、k-medoids、CLARANS。

2.3.2 階層法

階層法的處理程序是以樹狀結構的方式呈現，分群過程可分為凝聚法(Agglomerative approach)與分裂法(Divisive approach)兩種。凝聚法屬於buttom-up特性，在N個資料節點中，每一個節點皆是獨立的一群，接著計算每一個群集之間的距離，將距離最短的兩個群集合併，即相似度高的資料進行合併，由下而上的方式直到全部資料凝聚成一個大群集，Single-Link、Complete-Link、Average-Link、Centroid、Ward's method皆屬於凝聚法。

分裂法與凝聚法相反，以top-down的方式由上而下分裂而成，一開始將所有的資料節點視為同一個群集，接著由下每一回合分裂出較小群集，直到所有的資料節點各自形成一個群集。

2.3.3 密度法

密度法與分割法的差異在於，分割法是以距離做為分群的依據，造成分群後的群集形狀趨向於圓形，如果要找到其他形狀的群集則可使用密度法。其主要作法是計算某範圍內的資料點是否大於門檻值(threshold)，若是則合併為同一個群集，讓群集內的密度高於群集外的密度，藉此過濾掉極端值(outlier)或雜訊(noise)，具代表性的演算法有DBSCAN、OPTICS。

2.3.4 網格法

網格法是將資料空間量化(quantize)成一定數量的方格(cell)，並以方格為單位進行資料處理運算，運算速度取決於cell的數目，而非資料點的多寡，不同於分割法與階層法需要對每一個資料點進行運算，可以減少資料處理的時間，網格分群演算法有STING、CLIQUE。

2.3.5 模式法

模式法是假設(hypothesized)每一個群集都有一個數學模式，接著去找出與此群集最能匹配的數學模式。採用此演算法的主要有統計方法(statistical approach)或類神經網路方法(neural network approach)，以模式法為基礎的演算法有COBWEB、SOM、EM。

本研究利用SOM方法而不使用其他分群方法，是因為SOM可以映射出輸入樣本的特性，讓系統可以較平均地挑選音樂給使用者。再藉由使用者的回饋資料即可知道使用者的音樂喜好，如此可以達到快速地找到使用者真正的音樂喜好。

而其他分群方法在評估之後不採用是由於，分割法需要事先設定分群數目，如果分群數目取決不當，分群後的品質會不穩定，例如同一集群中可能存在兩種以上類型的音樂。階層法需要比對每個資料點的相似程度，因此計算量大，需要大量的時間才能分群完畢。密度法是計算某範圍內的資料點是否大於門檻值，來進行分群與擴張，藉此過濾掉極端值或雜訊，因不符合本研究的需求，所以不採用。網格法雖然分群速度快，但跟密度法一樣可以過濾雜訊，且有些方法架構複雜，也不使用。

因此本研究在考量過後決定採用SOM方法，並且與一般推薦系統常使用的k-means分群法、隨機法做比較。

2.4 自組織映射圖網路

在類神經網路模型中，自組織映射圖網路(Self-Organizing Map, SOM)為無監督式學習網路(unsupervised learning network)的一種，屬於競爭式架構，是在1980年由Kohonen所提出。無監督式學習網路是將訓練樣本(training examples)經過數次學習後，找出潛在的群聚(cluster)規則。其基本原理相當於大腦中具有相似功能的細胞會聚在一起，產生物以類聚的特性，自組織映射圖網路即模仿此特性，當學習完畢後，鄰近區域內的神經元會具有相似的

功能，也就是具有相似的連結加權值[14]。

競爭式學習法為雙層類神經網路，當輸入樣本與輸出神經元距離最短者即為優勝(winner)神經元，此優勝神經元可以獲得最高的權重調整，也就是所謂的贏者全拿學習法則(winner-take-all learning rule)。SOM同樣屬於競爭式學習法，但不同的是SOM有鄰近區域(neighborhood)的概念，當找出優勝神經元後，鄰近區域內的非優勝神經元亦可得到學習，且在網路學習過程中鄰近區域會逐漸縮小，直到網路收斂或達到學習循環次數為止。

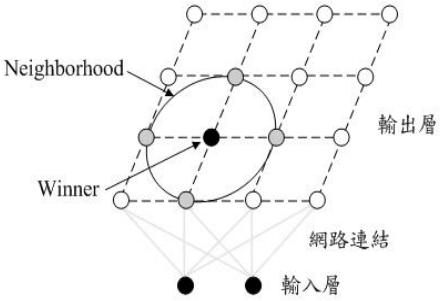

圖 2 自組織映射圖網路架構

SOM的架構包含輸入層、輸出層與網路連結，輸入層即訓練樣本的輸入向量，輸出層即訓練樣本的群聚，通常以二維的網路拓樸結構呈現，如上圖2所示，每一個輸入樣本反覆對輸出神經元做訓練，讓輸出神經元彼此競爭，以取得調整連結加權值的機會，最後網路收斂或達到學習循環次數即停止學習，由結果可看出輸出神經元映射出輸入樣本的特性。

3. 系統架構與方法

3.1 問題定義與方法

不同類型的音樂推薦系統，大致上都幫助使用者找到喜好的音樂，但在實驗結果中也透露出，一些特定的使用者對系統所推薦的音樂不甚滿意。這是因為推薦系統是依據使用者的歷史資料來當作推薦的憑據，所以不同種類的系統都有著 Cold-Start 問題，以內容導向式過濾法而言，當新進使用者進入系統時，由於系統一開始並不知道新進使用者的音樂喜好，所以會先隨機挑選幾首歌曲讓使用者試聽，使用者在聆聽系統所挑選的歌曲過後可以給予評分，得以讓系統知道使用者的音樂喜好，再根據評分結果推薦歌曲給使用者，但是系統需要經過多次的詢答才能找到使用者的真正喜好，造成 Cold-Start 的時間過長。

而且隨機挑選的方式會有推薦不平均的現象產生，例如流行音樂最為大眾所接受，所以系統在隨機挑選音樂的時後，常常會挑選到流行音樂，而其他類型的音樂被挑中的機率則較小。若是使用者剛好喜歡比較冷門的音樂，系統也許需要花費大量的時間才會挑選到冷門的歌曲，因此推薦結果就不甚滿意也會影響到音樂推薦系統的準確率。

由於音樂需要讓使用者花時間聆聽的特性，系統在初始時如果選取太多音樂給新進使用者評分，新進使用者大多無法完成評分的動作，因此一般的推薦系統都只隨機挑選幾首歌曲給使用者評分。在此本研究為了解決隨機挑選所造成推薦不平均的結果，本研究利用 SOM 將音樂資料庫裡的音樂作群聚，再從中挑選歌曲，並判斷系統是否能夠平均的挑選到不同的歌曲類型，其系統流程如下圖 3 所示。

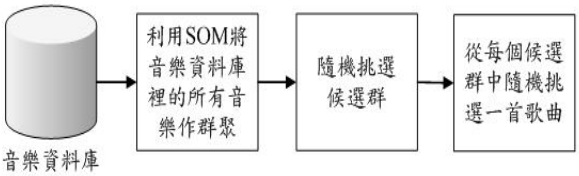

圖 3 系統流程圖

首先建立一個音樂資料庫，將擷取出的音樂特徵值當作 SOM 的輸入樣本，經過 SOM 訓練後可以得知音樂資料庫裡的音樂分佈狀況，再隨機挑選候選群，並從每一個候選群中挑選一首歌曲。假設音樂資料庫裡存放十首歌曲，輸出神經元為 4X4，經過 SOM 訓練後相似的歌曲會聚集在一起，如下圖 4，系統隨機挑選兩群候選群，再從這兩個候選群中挑出一首歌曲，紅色字體(01、05)即為被選中的歌曲。

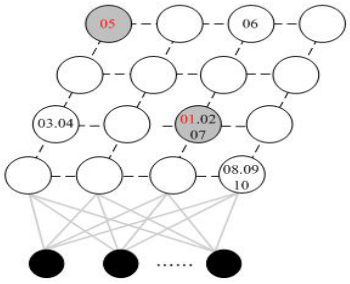

圖 4 SOM 分群示意圖

3.2 音樂特徵值

初期的音樂特徵值大多是以文字表示，例如演唱者、音樂類型、發行年代等資訊，這些音樂特徵值通常是以人工的方式取得，既沒效

率也有很多的限制。為了解決這樣的問題，有許多學者開始分析音訊資料，並藉由特定的計算方式，擷取出音樂特徵值。

音樂特徵值擷取出來後是以數字表示，因此可以利用距離公式計算各個特徵值的相似度，並找出各個特徵值之間的差異。本研究利用擷取特徵工具，把每首歌曲的特徵值給擷取出來，共有 12 種不同維度的特徵值，當作 SOM 的輸入樣本，其音樂特徵值如下：

- 音高(Pitch)：代表聲音頻率的高低，波長越短、頻率震動越快，音高就越高。在樂譜上音符的高低位置，即代表聲音的音高，任何樂器 Middle C 的 Do，如以 MIDI 音符編號表示則為 60。

1. Average Pitch：整首歌曲中每個音符的音高平均數。
2. Pitch Standard Deviations：整首歌曲中每個音符的音高標準差。
3. Pitch Entropy：每個音符的音高在整首歌曲中出現的次數平均。
4. Pitch Density：每個音符的音高在整首歌曲中所佔的比例。
5. Pitch Interval Entropy：Pitch Interval 代表兩個連續音符的音高差，計算每個音高差在整首歌曲中出現的次數平均。
6. Refrain Average Pitch：副歌部份每個音符的音高平均數。
7. Refrain Pitch Standard Deviations：副歌部份每個音符的音高標準差。
8. Refrain Pitch Entropy：每個音符在副歌部份出現的次數平均。
9. Refrain Pitch Density：每個音符在副歌部份所佔的比例。

- 音長(Duration)：聲音震動的長度，在樂譜上是指每個音符持續的長度。

10. Average Duration：整首歌曲中每個音符的音長平均數。
11. Duration Entropy：每個音符的音長在整首歌曲中出現的次數平均。

- 速度(Tempo)：代表節拍的快慢程度，即一分鐘拍子的速度(Beats Per Minute, BPM)。例如指定一個四分音符等於 120bpm，則四分音符的長度即一分鐘除以 120，也就等於 0.5 秒。bpm 越大，拍子的速度就越快，通用的節拍標記如下表 1 所示。

表 1 節拍標記表

Largo	40-60bpm	最緩版
Larghetto	60-66bpm	甚緩板
Adagio	66-76bpm	慢板
Andante	76-108bpm	行板
Moderato	108-120bpm	中板
Allegro	120-168bpm	快板
Presto	168-200bpm	急板
Prestissimo	200bpm 以上	最急板

12. Tempo：計算歌曲的速度。

音樂特徵值擷取完畢後，為了讓特徵值不出現極端值，而影響到實驗結果，將進行資料正規化的動作，讓資料介於零到一百之間，如公式(2)，dmax為該維度中所有資料的最大值，dmin為該維度中所有資料的最小值，α為欲正規化的資料。

$$A_\alpha = \frac{(d_{\max} - \alpha) * 100}{d_{\max} - d_{\min}} \qquad (2)$$

4. 實驗分析

4.1 系統環境建立與評估

實驗環境為桌上型電腦，CPU為Intel 2.4GHz，配載1G RAM ，作業系統為Window XP SP2。系統的開發環境為Microsoft Visual Studio 2005，開發程式語言為C#。資料庫系統採用Microsoft Office Access 2003。

歌曲來源是由網路上所蒐集到的208首MIDI歌曲，經由MIDINOTE擷取出音樂特徵值，並存放於音樂資料庫裡。由於目前沒有系統可以有效的對音樂類型進行分類，例如一首搖滾歌曲經過特徵值計算距離後，其距離跟電子音樂比較接近，因此被歸類在電子音樂類型裡。為了避免此種情形影響實驗結果，我們以人工的方式將所蒐集到的歌曲分成不同的類型，藉此判斷SOM與隨機法、k-means分群法在挑選歌曲類型時是否平均。

4.2 系統運作流程

由於音樂需要花時間去聆聽的特性，推薦系統在一開始提供太多取樣音樂給新進使用者評分，新進使用者大多無法完成啟始之歌曲評價作為。因此本研究將從資料庫中挑選出十個候選群，再從這十個候選群中各取出一首歌曲，並於實驗中驗證比較利用隨機法、k-means分群法、SOM方法在抽樣的歌曲類別是否能

滿足樣式之普遍性。

對於k-means分群法及SOM方法我們將設定相同之群組數，例如SOM採5X5節點數時，k-means之群組數k = 25。另外歌曲抽樣的程序為：先隨機選擇候選群，再由候選群中隨機選擇群內歌曲作為抽樣歌曲。

本研究的實驗主要是為了比較隨機法、k-means分群法與SOM方法的抽樣結果，因此SOM在不同輸出神經元的實驗中，參數統一設定為：學習率初始值為0.9，學習率最小值為0，學習率遞減係數為0.975，鄰近半徑初始值為3，鄰近半徑最小值為0.1，鄰近半徑縮小因子為0.95，實驗中訓練次數在100次時已經趨近收斂，因此訓練次數設定為100。

4.3 實驗結果與分析

實驗中透過SOM訓練後，可得知音樂資料庫裡的歌曲分佈狀況，再隨機挑選十個候選群，並從十個候選群中隨機挑選一首群內歌曲作為抽樣歌曲，為了與k-means分群法、隨機法比較所挑選的十首歌曲類型是否滿足樣式之普遍性，我們執行1000次的歌曲挑選並計算出每個音樂類型被選中的次數。

當SOM輸出神經元為5X5時，與k-means分群法、隨機法的歌曲類型在經過1000次挑選後，被選中的次數比較如下表2。

表 2 SOM 輸出神經元 5X5、k-means 分群法、隨機法之比較

方法 / 類型	SOM	k-means	隨機
搖滾	168	201	317
藍調	167	186	259
鄉村	164	127	26
舞曲	165	131	40
電子	165	143	32
抒情	171	212	326

實驗結果顯示，當SOM輸出神經元越接近歌曲類型數目時，系統所挑選出來的歌曲類型也就越平均，與隨機法相比，可以增加小群被挑選的次數。這是因為SOM是以輸入樣本與輸出神經元做反覆計算，並調整網路連結值，最後可從輸出神經元看出輸入樣本的特性，例如比較大群的集群可能會被分配到三個神經元，而小群的集群也會有一個神經元，若是依此法挑選歌曲給使用者，就可以兼顧到小的族群，且對於一些特定使用者而言，可以減少系統蒐集資料的時間，即Cold-Start的時間。而k-means的群組數與SOM的神經元數目相同，經過資料分群過後品質較不穩定，可能造成較大集群裡的歌曲被分割到不同的集群中，所以挑選出來的歌曲類型沒有較SOM平均，但結果還是比隨機法好。最後的隨機法在挑選歌曲時，通常從歌曲分佈大又密集的集群中挑選，因此比較冷門的類別就不容易挑選到，呈現出極端的差距。

當SOM輸出神經元為10X10時，與k-means分群法、隨機法的比較如下表3。

表 3 SOM 輸出神經元 10X10、k-means 分群法、隨機法之比較

方法 / 類別	SOM	k-means	隨機
搖滾	192	226	312
藍調	186	196	285
鄉村	132	98	22
舞曲	147	121	31
電子	142	119	29
抒情	201	240	321

由實驗結果可以得知，SOM的輸出神經元與k-means的分群數目都是100，所以隨機挑選出的歌曲類型重複性也跟著增加，這是由於SOM輸出神經元與k-means的群組數目越接近資料量時，等於一首歌曲被分為一群，這樣的挑選結果就會越趨近於隨機法，呈現出明顯的差距。但是這三種方法比較之下，仍可看出SOM挑選出來的歌曲類型比隨機法、k-means分群法還要平均。

由於本研究的音樂資料庫只存放208首歌曲，並從中隨機挑選出十群候選群，再從這十群中各取出一首歌曲樣本與隨機法、k-means分群法做比較，所以SOM的輸出神經元至少要4X4以上，為了比較在SOM輸出神經元越接近歌曲類型數目、資料量時的差別，故本研究只比較SOM輸出神經元為5X5與10X10的挑選狀況。

從實驗結果可以得知，當SOM輸出神經元數目越接近歌曲類型數目時，系統所挑選的歌曲類型也就越平均，而SOM輸出神經元數目趨近於資料量時，實驗結果也就越接近隨機法。雖然SOM在輸出神經元數目越接近資料量時，挑選結果也就越接近隨機法，所以只要神經元數目不接近資料量，例如輸出神經元為14X14，挑選結果都會比k-means分群法、隨機法來的平均。

由於本研究是屬於內容導向式過濾法，除了以音樂特徵值去計算相似程度外，還以人工

的方式將歌曲分類成不同的音樂類型，為的是在實驗的部分可以比較SOM與隨機法、k-means分群法，在挑選歌曲類型時是否平均。

而一般的推薦系統是以隨機法挑選歌曲，對於一些特定的使用者而言，是需要經過多次的詢答，才能累積夠多的使用者資訊，再正確的推薦給使用者所喜歡的音樂。如此可知系統在挑選歌曲時，能夠平均的在不同類型中挑選，即可快速地知道使用者的音樂喜好，也可減少系統Cold-Start的時間。

5. 結論與未來研究方向

由於推薦系統普遍存在Cold-Start問題，即新進使用者初始使用系統時，系統並不瞭解新進使用者的音樂喜好，所以會隨機挑選歌曲讓使用者評分，並記錄使用者的評分結果，經過反覆詢答後，即可知道使用者的音樂喜好。但是對於一些較喜歡冷門音樂的特定使用者而言，系統必須花費大量的時間去累積使用者資訊，才會找到使用著的喜好。

因此在本研究中利用SOM物以類聚的特性，讓音樂資料庫裡相似的音樂聚集在一起，再隨機選擇候選群後，從每群中挑選出歌曲。這與一般隨機挑選的方式不同點在於，隨機挑選會先從歌曲分佈較大且密集的集群中挑選歌曲，也就是大部分使用者所能接受的熱門音樂，但對於特定使用者所喜歡的冷門音樂，系統也許需要花費大量的時間才會挑選到，造成推薦結果不準確。且從實驗結果得知，本研究所提出的方法可以平均的挑選到不同的音樂類型，如果將挑選出來的歌曲給使用者評分，再藉由使用者回饋資料即可快速地找到使用者的音樂喜好，以改善Cold-Start問題，而且還可以提升系統的推薦準確率。

本研究所提出的方法，算是推薦系統的前處理程序，只要有新進使用者進入系統時，即可利用此方法找到使用者的音樂喜好，達到有效的推薦。另外，當新的音樂產生時，就算沒有任何評分記錄，也可經過SOM訓練後適當的推薦給使用者。因此該方法可以應用在不同機制的推薦系統中。

未來的研究方向可以利用本方法搭配不同的推薦機制，並調整SOM的參數設定，比較SOM在不同輸出神經元狀態下，是否可以有效的減少Cold-Start的時間及提升系統的推薦準確率。

誌謝

本研究承國科會專題研究計畫(編號：NSC98-2200-E-020-004 及 NSC98-2221-E-020-025-MY2)補助經費，謹此誌謝。

參考文獻

[1] Ahn, H. J., “A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem,” ***Information Sciences***, Vol. 178, pp. 37-51, 2008.

[2] Armstrong, R., Freitag, D., Joachims, T. and Mitchell, T., “WebWatcher: A Learning Apprentice for the World Wide Web,” ***In Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous***, Distributed Environments, Stanford, CA, 1995.

[3] Balabanovic, M. and Shoham, Y., “Learning Information Retrieval Agents: Experiments with Automated Web Browsing,” ***AAAI Spring Symposium on Information Gathering from Heterogeneous***, Distributed Environments, Stanford, 1995.

[4] Balabanovic, M. and Shoham, Y., “Fab: content-based, collaborative recommendation,” ***Communications of the ACM***, Vol. 40, pp. 66-72, 1997.

[5] Belkin, N. J. and Croft, W. B., “Information Filtering and Information Retrieval: two sides of the same coin?” ***Communications of the ACM***, Vol. 35, pp. 29-38, 1992.

[6] Beth L., “Music recommendation from song sets,” ***International Conference on Music Information Retrieval***, pp.425-428, 2004.

[7] Burke, R., “Hybrid Recommender Systems: Survey and Experiments,” ***User Modeling and User-Adapted Interaction***, Vol. 12, pp. 331-370, 2002.

[8] Chen, H. C. and Chen, A. L. P., “A music recommendation system based on music data grouping and user interests,” ***Proceedings of ACM International Conference on Information and Knowledge Management***, pp. 231-238, 2001.

[9] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M.,

"Combining Content-Based and Collaborative Filters in an Online Newspaper," ***In Proceeding of the ACM SIGIR Workshop on Recommender Systems***, 1999.

[10] Delgado, J., Ishii, N. and Ura, T., "Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents," ***In Proceedings of the Second International Workshop***, CIA'98, 1998.

[11] Goldberg, D., Nichols, D., Oki, B. M., Terry, D., "Using Collaborative Filtering to Weave an Information Tapestry," ***Communications of the ACM***, Vol. 35, pp. 61-70, 1992.

[12] Hayes, C. and Cunningham, P., "Smart Radio-Building Music Radio On the Fly", ***In Proceedings of Expert Systems***, 2000.

[13] Kautz, H., Selman, B. and Shah, M., "Referral Web: Combining Social Networks and Collaborative Filtering," ***Communications of the ACM***, Vol.40, No.3, pp. 63-65, 1997.

[14] Kohonen, T., "The self-organizing map," ***Proceedings of the IEEE***, Vol. 78, pp. 1464-1480, 1990.

[15] Konstan, J. A., Miller, B. N. and Maltz, D., "GroupLens: Applying collaborative filtering to Usenet news," ***Communications of the ACM***, Vo1.40, No.3, pp.77-87, 1997.

[16] Kostov, V., Naito, E., and Ozawa, J., "Cellular Phone Ringing Tone Recommendation System Based on Collaborative Filtering Method," ***In Proceeding of IEEE International Symposium on Computational Intelligence in Robotics and Automation***, Vol. 1, pp. 378-383, 2003.

[17] Krulwich, B. and Burkey, C., "The InfoFinder agent: Learning user interests through heuristic phrase extraction," ***IEEE Intelligent Systems Journal (Expert)***, Vol.12, No.5, pp.22-27, 1997.

[18] Lang, K., "Newsweeder : Learning to Filter Netnews," ***In Proceedings of the 12th International Conference on Machine Learning***, pp.331-339, 1995.

[19] Lawrence, R.D., Almasi, G. S., Kotlyar, V., Viveros, M.S. and Duri., S. S., "Personalization of Supermarket Product Recommendations," ***Data Mining and Knowledge Discovery***, Vol.5, No.1-2, pp.11-32, 2001.

[20] Poirier, D., Fessant, F., and Tellier, I., "Reducing the Cold-Start Problem in Content Recommendation through Opinion Classification," ***IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology***, Vol. 1, pp. 204-207, 2010.

[21] Resnick, P. and Varian, H. R., "Recommender systems," ***Communications of the ACM***, Vol. 40, pp. 56-58, 1997.

[22] Ricardo, B. Y. and Berthier, R. N., "Modern Information Retrieval." ***Addison-Wesley***, 1999.

[23] Rucker, J. and Polanco, J. M., "Siteseer: Personalized Navigation for the Web," ***Communications of ACM***, Vol.40, No.3, pp.73-75, 1997.

[24] Shardanand, U. and Maes, P., "Social Information Filtering: Algorithms for Automating 「Word of Mouth」," ***In Proceeding of ACM CHI'95 Conference on Human Factors in Computing Systems***, pp. 210-217, 1995.

[25] Tzeng, Y. S., Chen, H.C., and Chen, A. L. P., "A New Approach for Rating-Based Collaborative Music Recommendation Using Personal Preferences and Opinions From Trustable Users," ***In Proceeding of The IASTED Conference on Internet and Multimedia Systems and Applications (IMSA)***, 2004.

[26] Nouali, O., and Belloui, A., "Using semantic web to reduce the cold-start problems in recommendation systems," ***IEEE Second International Conference on the Applications of Digital Information and Web Technologies***, pp. 525-530, 2009.

[27] Wang, H. W., Wang, J., Yang, and Yu, P. S., "Clustering by Pattern Similarity in Large Data Sets", ***In Proceeding of ACM Special Interest Group on Management of Data***, 2002.

[28] Wasfi, A. M. A., "Collecting User Access Patterns for Building user Profiles and Collaborative Filtering," ***In Proceedings of the ACM Conference on Intelligent User Interfaces***, 1999.