

基於權重式歐幾里德距離之 音樂推薦系統設計

蘇軍維

國立屏東科技大學
資訊管理系
研究生

m9656027@mail.npust.edu.tw

潘雅婷

國立屏東科技大學
資訊管理系
研究生

m9756031@mail.npust.edu.tw

劉寧漢*

國立屏東科技大學
資訊管理系
助理教授

gregliu@mail.npust.edu.tw

摘要

不同的推薦系統各自擁有專屬的推薦機制，並藉由資料過濾、分群等程序，篩選出適合的項目推薦給使用者。在網路與數位科技的快速發展下，帶動了線上收聽音樂的熱潮。音樂推薦系統也搭著這波熱潮，提供使用者於網路上收聽音樂的訴求。目前常見的音樂推薦系統，大多都是以擷取出的音樂特徵值為依據，再進行比對、分群等程序。這樣的方式，在各種推薦機制中都有不錯的效果，但是卻忽略特徵值與使用者之間的影響程度。不同的使用者，對於相同特徵值的變動，所產生的影響程度並不相同，若用同樣的判斷方式，勢必在進行判斷時，產生無法避免的誤差。有鑑於此，本研究針對使用者進行簡單的詢答，藉此計算使用者權重矩陣，瞭解使用者與各個特徵值之間的影响程度，藉此提升音樂推薦系統的推薦準確率。

關鍵詞：音樂推薦系統、權重、歐幾里德距離、內容過濾。¹

1. 前言

網路的方便性與多功能性，受到越來越多使用者的青睞，使用者們漸漸的開始將收聽音樂的習慣移轉至網路上。但是太多的選擇往往也會造成反效果，在這百家爭鳴的音樂市場下，新人歌手數不勝收，以台灣市場來看，一年發行的本土唱片專輯就有上百張，其中包含主流與非主流的音樂專輯、熱門歌手、新人歌手等，這麼多的選擇，使用者於挑選音樂時，

一定會有相當程度的困擾。

因此有許多的業者，推出以播放音樂為主的線上音樂網站，提供使用者於網路上收聽音樂的訴求。這類型的音樂網站，不僅僅提供使用者收聽音樂的訴求，讓使用者自行挑選欲播放的歌曲清單，更會藉由各種判斷機制，推薦音樂給使用者，讓使用者可以輕易的獲得新的音樂資訊，解決自行搜尋音樂的困擾。

傳統的音樂推薦方式，大多都是以歌曲點播率、銷售成績等做為判斷依據，再搜尋出適合的歌曲推薦給使用者。在個人化需求與差異性行銷的趨勢下，傳統的音樂推薦方式並無法滿足每個使用者。例如以歌曲點播率與銷售成績做為推薦依據，雖可滿足大部分的使用者，但對於少數使用者而言，這樣的推薦方式他們並不能接受。可能的原因在於，這些使用者喜歡較冷門的音樂，或是這些音樂他們早就已經收聽過了。因此某些線上音樂網站的業者為了吸引更多的使用者，會針對個別使用者發展出專有的推薦機制，藉此提升使用者於線上收聽音樂時的滿意度。在學術界上，也有許多的學者在個人化音樂推薦的領域上，發現許多問題，也提出了許多的解決方法，增加個人化音樂推薦的正確率與效率。個人化音樂推薦方式，顯然已經成為現代音樂推薦的趨勢。

目前常見的音樂推薦方式，通常是利用音訊資料的特徵值做為參數，計算使用者與使用者之間或音樂與音樂之間的距離。距離較接近者，定義為擁有較高的相似度，再根據音樂推薦機制，推薦使用者適合的歌曲。此種類型的音樂推薦方式，除了音樂推薦機制的處理程序會對實驗結果產生影響外，另一項主要的因素，則是在於特徵值的擷取，有效的特徵值能成功的提升推薦的正確率與效率。反之，不適合的特徵值，不僅無法提升音樂推薦的準確度與效率，更可能嚴重降低預期的實驗數據。

* 為通訊作者

目前已有許多的學者，針對不同的因素與看法，提出了許多以不同特徵值為基礎的音樂推薦機制，有些推薦機制更一次使用了十種以上的音樂特徵值。每位學者的方法都各有各的特色，也皆成功的提升了音樂推薦的準確率，改善了原本不足的地方，但在討論各個特徵值是否能有效提升音樂推薦機制準確度的同時，卻也因此忽略了使用者對於每個特徵值的感受程度並不一定相同。

某些使用者，可能針對某些特定的特徵值感覺特別敏銳，且此特徵值能有效的表示使用者的喜好。如果利用傳統的距離計算公式，做為推薦的判斷依據，並無法特別凸顯特定特徵值，對於使用者的影響程度，更可能因為其他特徵值的影響，導致結果不如預期。如圖1中的User A，假設長方形的藍色區塊表示User A喜歡的歌曲分佈區域，dimension1與dimension2分別代表不同的音訊資料特徵值。由User A的喜好分佈情形可以看出，在不影響喜好的情形下，User A較能接受dimension1的資料變動，較不能接受dimension2的資料變動。由於這樣的特性，造成User A的喜好呈現長方形的分佈。

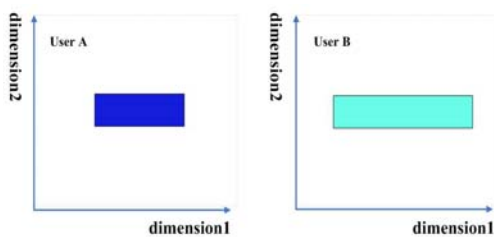


圖 1 使用者與音訊資料特徵值關係圖

傳統的音樂推薦機制，每個特徵值都給予相同的權重，因此計算出的分佈區域大多以圓形或正方形為主。類似User A的特殊分佈情形，以傳統的音樂推薦機制並無有效涵蓋User A的喜好分佈。除此之外，我們再將User A與User B做進一步的比較，假設長方形的青色區塊表示User B喜歡的歌曲分佈區域，dimension1與dimension2分別代表與User A相同的音訊資料特徵值。由圖中可以發現，在不影響喜好的情形下，User A與User B對於dimension2的資料變動接受區域大約相同，但是User B對於dimension1的接受區域就比User A大的多，這表示dimension1資料變動會產生不同程度的影響給User A與User B。

由此可見，每個特徵值都給予相同權重的條件下，在音樂推薦的過程中，會產生許多的問題。有鑑於此，本研究針對每個使用者進行

簡單的詢答，並根據詢答結果，篩選出有效的資料，計算每個使用者專屬的使用者權重矩陣，再搭配新的距離計算公式，推薦使用者適合的歌曲，藉此提升推薦的準確率與效率。

本研究共分為5個章節，第2章為文獻探討，我們針對目前推薦系統的類型作描述與相關文獻的討論；第3章為研究方法，說明本研究之研究流程說明；第4章則是最後的實驗數據以及說明討論；第5章則說明結論與未來研究方向。

2. 相關文獻探討

2.1 音樂特徵值

音樂特徵值是利用特定的方式，從欲表示的歌曲中擷取出來的資訊。一個音樂特徵值可以代表一首或多首不同的歌曲，而多個音樂特徵值，也可以只代表一首歌曲。在音樂推薦的機制當中，音樂特徵值扮演著非常重要的角色，傳統的音樂特徵值，大多是演唱者、曲風、年代等相關資訊，且大多需要以人工的方式來分類，分類技術發展至今，仍尚無法有效的針對音樂類型進行分類。人工的分類方式不僅耗費時間，有時也可能出現無法避免的錯誤，例如曲風的分辨，可能在各個專家認知不同或歌曲曲風不明顯的情況下，造成資料分辨的錯誤。

隨著數位科技的發展，越來越多以數位格式儲存的音訊資料出現，因此也出現了許多由數位訊號計算出來的音樂特徵值，例如音量(Volume)、音高(Pitch)、與音色(Timbre)[33]與過零率 (Zero Crossing Rate, ZCR)[23][29]等。由於這些特徵值的出現，造就了許多不同於以往的音樂推薦方式。以數位訊號計算出的音訊特徵值，不僅可以藉由電腦自動擷取，且在資訊爆炸的時代，可快速的處理大批的資料，更不會出現認知不同的情形發生，相對於傳統的音訊特徵值，在某些部分上，顯得較具有明顯的代表性。

2.2 距離公式

目前學術界有許多計算距離的公式，例如點與點之間的距離、點與線之間的距離，甚至字串與字串之間的距離，都有許多的學者在討論，也有許多的計算方式來表示。而在數學

上，距離是定義在度量空間中的一種函數，且距離的表示方式不會是負數。以下我們介紹幾種比較常見的距離公式，分別為歐幾里德距離與明考夫斯基距離。

2.2.1 歐幾里德距離

歐幾里德距離[6] (Euclidean distance)是歐幾里德提出的距離公式，又稱歐式距離，為目前最常見的距離公式。歐幾里德距離主要定義歐幾里德空間中，兩點 $x = (x_1, \dots, x_n)$ 和 $y = (y_1, \dots, y_n)$ 之間的距離，其中 x 與 y 為以一個或多個特徵值表示的相同類型物件，且其特徵值皆以數值的方式表示。距離計算公式如公式(1)所示。

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

歐幾里德距離在資訊相關領域被很頻繁的提及，例如人工智慧中的路徑尋找，便經常會利用到歐幾里德距離做起發函數(heuristic function)，而在資料分群的相關領域中，著名的演算法 k -means，也是使用歐幾里德距離，作為點與點之間的距離計算方式。除了上述領域外，歐幾里德距離在學術界其他領域上，也被廣泛的應用，例如資料分群、語音辨識或是音樂推薦等研究領域中，都常出現歐幾里德距離的應用。

2.2.2 明考夫斯基距離

明考夫斯基距離[19] (Minkowski distance) 又稱 L_p norms，可說是歐幾里德距離的延伸，假設在 d 維空間中有兩點 $x = (x_1, \dots, x_d)$ 和 $y = (y_1, \dots, y_d)$ ，則其明考夫斯基距離可表示為：

$$d(x, y) = \left(\sum_{i=1}^d |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (2)$$

明考夫斯基距離具有下列以下特性，當 $p = 1$ ，明考夫斯基距離表示 x 、 y 間各維度絕對值總和，又稱 city block distance 或 taxicab distance。當 $p = 2$ ，明考夫斯基距離即為歐幾里德距離。當 $p = \infty$ ，明考夫斯基距離為 $\max |x_i - y_i|$ ，稱為 maximum distance。

2.3 推薦系統

推薦系統(Recommend System)是一種可以在大量可能的選擇項目中，透過推薦機制，

引導使用者挑選出最感興趣或最有用資訊的個人化服務工具。不同的推薦系統可用來解決不同種類的推薦問題，目前較常見的音樂推薦系統，可根據其演算法的不同分成四大類，分為內容導向式(Content-Based, CB)、合作式過濾法(Collaborative-Filtering, CF)、混合式推薦法(Hybrid Recommendation Method)。

2.3.1 內容導向式

內容導向式的音樂推薦系統，主要根據歌曲本身所擷取出的內容特徵，來進行分析、比對，並比較資料間的關聯性，進而提供使用者建議的推薦方式。如[1][31]是從使用者過去的偏好資料當作依據的內容導向式音樂推薦系統。此種推薦方式，必須先為使用者建立特性輪廓(User Profile)，使用者特性輪廓主要是記錄一些關鍵詞及使用使用者相關資訊，例如收聽過的歌曲名稱、年份、曲風、喜歡的歌手等相關訊息。再經由過濾系統的過濾機制，過濾出與使用者特徵輪廓相吻合的音樂。

目前已經有多位學者提出以內容導向式的推薦機制做為核心的推薦系統，應用的範圍也相當的廣泛，包括新聞、文件、音樂、商品等，較著名的系統有 InfoFinder[14]、NewsWeeder[16]、SmartPad[18]。

在音樂推薦方面，Pandora[20]算是目前較著名的音樂推薦系統。早期的使用者，於網路上收聽音樂，大多都是選擇收聽廣播，但是有收聽過廣播的使用者都很清楚，廣播所播放的歌曲是由廣播DJ決定，使用者並無法選擇自己喜歡的歌曲，且廣播播放的歌曲大多都是以一般大眾可能的喜好做為推薦基礎，並無法依照個人喜好做推薦。另一方面，廣播電台為了增加收入，總是會定時的插撥一些廣告，這對單純只想聽音樂的使用者來說，會是一個很大的困擾。而Pandora便是為了解決上述問題而發展出來的音樂推薦系統。Pandora可以讓使用者自行選擇喜好的音樂類型，包括歌手、專輯、曲調、節奏等，系統便會產生專屬的音樂播放清單。

在學術上來說，Pandora是典型的內容導向式音樂推薦系統。Pandora藉由歌曲本身所擷取出的音樂特徵值做為判斷依據，例如歌手、專輯、曲調、節奏、配樂、歌詞等，再將具有相似特徵的音樂，推薦給使用者，產生專屬於該使用者的播放清單。除了Pandora之外，在內容導向式音樂推薦方面，學者Daniel

M.[4]藉由歌詞特徵分析，例如因文化背景所影響的音樂歌詞，描述出有種族衝突或反應社會現象，藉此找到相同喜好的使用者；Steffen P. & Sander van de W.[24]則是根據使用者特性輪廓及回饋評價，利用音樂內容特徵作分析基礎，來產生播放清單。

內容導向式的推薦方式，可以透過推薦項目的內容特徵，藉此能分類項目並推薦相似的項目，且不需其他使用者的資料即可作推薦，能有效解決Cold-start和資料稀疏問題，目前也已經被發展成數套系統，且可實際運作。但是此類系統也存在一些主要問題，例如無法推薦使用者從未接觸過的項目類型、特徵值不易分析、回饋資料不足等。

2.3.2 合作式過濾法

首先提出合作式過濾法概念的人為Goldberg D.[7]，此方法也稱作社會過濾(Social Filtering)方法。在1992年，Goldberg等學者在Tapestry系統中提出合作式過濾法的基本概念，Tapestry的主要目的在過濾電子郵件，以應付龐大的電子郵件。使用者可透過TQL(Tapestry Query Language)的查詢語法建立查詢，並經由不同的查詢所過濾出的結果，找出自己有興趣的信件。不同於現今的推薦系統，使用者需被動的輸入查詢指令，再藉由Tapestry根據所輸入的指令，過濾出使用者喜好的信件，而非由系統根據判斷機制，主動的針對使用者的喜好進行推薦。Goldberg D.學者提出的概念被後代的學者廣為接受，在多年的發展下，過濾技術已經成為了推薦系統的核心技術。

合作式過濾法的推薦機制，主要是藉由同一群體成員之間的喜好做為判斷依據，推薦喜好較高的項目給同一群體中相似度較高的使用者。合作式過濾法會根據使用者歷史資料與使用者喜好，利用特定的分群演算法，將相似的使用者聚集於同一群集。再於同一群集內尋找相似度較高的使用者，並由尋找出的使用者的歷史資料中，透過推薦機制，挑選出適合的項目進行推薦。

合作式過濾法的推薦方式有點像是我們台語所說的『吃好倒相報』。一般來說，朋友之間如果有發生什麼愉快的事情，便會與周遭的擁有相同興趣朋友分享，例如某天看到好笑的電視節目、聽到了好聽的歌曲等。而經由朋友推薦的項目，理論上來說也較容易被人接

受。合作式過濾法的推薦方式，便是利用這樣的特性，推薦項目給同一群集內擁有相同喜好的使用者。較著名的系統有GroupLens[13]、Referral Web[12]、PHOAKS[27]、Siteseer[21]、Ringo[25]。

iLike[11]屬於線上音樂推薦系統，提供免費音樂可供試聽和下載。iLike把「發現新音樂」視為最重要的功能，使用者可在網站的歌手庫中選擇喜愛的歌手，通過這個喜好的選擇，系統能幫助使用者找出關聯喜好的使用者，以實現同好的音樂社群(Social network service, SNS)。當使用者擁有專屬的音樂社群後，iLike便能幫助使用者於線上收聽音樂時，尋找到最新的音樂，並與好友彼此分享收聽的音樂內容。

Last.fm[17]與iLike相同，都是屬於線上音樂推薦系統。Last.fm與iLike推薦方式的差異在於分群方式，Last.fm不需透過使用者主動建立自己的喜好，例如輸入喜好歌手名稱、歌曲等相關資訊。而是透過使用者所點播過的音樂，去計算使用者之間的相似程度，便可找出同好相似群作出推薦。這樣的方式可以避免某些使用者不願意提供資料，造成音樂推薦的限制。

合作式過濾法可透過其他使用者的經驗，過濾較複雜難以表述的概念(如：個人喜好的歌曲內容特徵)，且能過濾不易進行自動內容分析的資料，如音樂類型或多媒體資料等。在推薦歌曲類型方面，合作式過濾法具有推薦新項目的能力，系統除了會推薦使用者較常收聽的音樂類型外，也會藉由與其他使用者比較的結果，推薦使用者從未聽過的歌曲類型，讓使用者於收聽音樂時，有意想不到的效果產生，這也是合作式過濾和內容導向式兩種推薦系方式最大的差別所在。

合作式過濾法為基礎的推薦系統是目前推薦系統中最普遍與成功的技術之一，許多的推薦系統，都朝著合作式過濾的方向改進，而合作式過濾法也解決了某些內容導向式推薦機制所遭遇到的問題，例如無法推薦使用者從未接觸過的歌曲類型、特徵值不易分析與回饋資料不足等。與內容導向式相比較之下，合作式過濾法確實在後續的應用上具有較大的優勢。

雖然如此，合作式過濾的推薦方式，仍然存在幾個較嚴重的問題，例如無法推薦音樂給新加入的使用者(冷起始問題)、資料稀疏，降低推薦準確度、擴充性問題、特殊使用者、使

用者興趣轉移等。

2.3.3 混合式推薦法

內容導向式與合作式過濾法為基礎的推薦系統，成功的在特定的領域與使用情況下，得到顯著的成果。但是仍然分別存在著某些難以解決的問題，例如新的類型無法在內容導向式為基礎的推薦系統中被推薦，合作式過濾法為基礎的推薦系統難以解決冷起始的問題。再比較過兩個推薦機制後，更可發現這兩種推薦機制，擁有相輔相成的效果，舉例來說，內容導向式為基礎的推薦系統，無法推薦使用者未曾接觸過的類型，但是合作式過濾法為基礎的推薦系統卻可以。合作式過濾法為基礎的推薦系統，擁有冷起始的問題，但是在內容導向式為基礎的推薦系統中，卻不會出現這樣的問題。由此可見，兩種推薦機制都各自擁有優缺點，而且可以幫助另一種推薦機制解決某些特定的問題。

有鑑於此，便有學者提出將內容導向式和合作式過濾法作結合，產生混合式推薦法，設法解決上述推薦機制的缺點，以產生最佳推薦結果。混合式推薦法的概念，在目前的學術領域中，廣泛的被眾家學者應用。常見的混合式推薦法大多都是以內容導向式與合作式過濾法為基礎的推薦系統，例如Fab[2]、Personal Tango[3]、ProfBuilder[32]、RAAP[5]與Smart Radio[10]皆是著名的混合式推薦系統。

普遍來說，混合式的推薦機制與其他的推薦機制，於概念上並無其他的不同，仍然必須先瞭解使用者的喜好，再來進行相關的音樂推薦。因此瞭解各個推薦機制的優缺點，並將具有互補功效的推薦機制有效結合，才是混合式推薦機制的精髓所在。

2.4 k-NN algorithm

k-NN[15]全名是k-th nearest neighbor，中文意思是「第k位最接近的鄰居」，而k所表示的是搜尋出的最接近鄰居數目。舉例來說，3NN表示需搜尋出3個最接近的鄰居節點，6NN表示需搜尋出6個最接近的鄰居節點。

k-NN演算法，會以一個樣本點當作種子，也就是中心點。然後與其他點做比較，搜尋出與種子距離最接近的k個點。如圖2所示，深藍色的點即為種子，我們比較種子與其他節

點的距離，並找出距離最接近的3個點，即是3NN演算法。

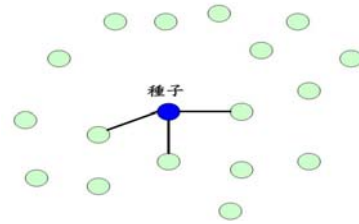


圖 2 3NN 演算法示意圖

k-NN演算法的應用範圍非常廣泛，尤其在推薦系統的應用領域中，常常可以看見k-NN演算法的應用。以音樂推薦為例，大多數的推薦系統，在搜尋相似歌曲時，皆是使用k-NN演算法。

3. 系統架構與方法

各類型的音樂推薦機制，雖然分別解決了音樂推薦系統在進行音樂推薦時，於不同情況上所遭遇到的問題，但在各項的實驗結果與數據中也顯示出，特定使用者的實驗結果總是不如預期。例如合作式過濾法為基礎或內容導向式為基礎的音樂推薦，都會於音樂推薦機制中，出現計算相似度的距離公式，並藉由計算的結果找出適合的推薦歌曲，但是總是有特定使用者對於推薦歌曲不甚滿意，即使經過人工智慧演算法的訓練，試圖找出使用者的真正喜好，也無法有效的提升使用者的滿意度。

會影響實驗結果的因素，除了某些使用者對於喜好音樂類型的改變速度過快，或是根本不喜歡聽音樂等無法解決的因素外，最主要的原因還是在於，每個音樂特徵值對於使用者的影響程度不同。每個使用者的喜好不同，相對的喜歡的特徵值也不同。雖然有些使用者，可能喜歡相同的音樂特徵值，但是針對特徵值變動與使用者喜好之間的影响也不太可能相同。舉例來說，使用者A、B對於兩個音樂特徵值，過零率與音高標準差能直接影響其喜好。使用者A喜好的過零率介於0.5~0.6之間，音高標準差介於0.4~0.85之間，使用者B喜好的過零率介於0.4~0.85之間，音高標準差介於0.5~0.6之間，兩位使用者喜好的特徵值雖然相同，但是特徵值接受範圍卻相反。在這樣的情形下，雖然特徵值過零率與音高標準差對兩位使用者皆有直接的影响，但在兩個特徵值相比較之下，對於使用者A而言，過零率有較大的影响程度，而使用者B卻剛好相反。

這樣的情形，不管是在使用者分群、相似歌曲搜尋等音樂推薦過程中，都會造成一定程度的影響。

為了解決音樂特徵值對於各個使用者影響程度不同所造成的問題，必須瞭解音樂特徵值對於各個使用者的影響程度，並依照影響程度給予不同的權重，再進行之後的音樂推薦機制。本研究透過簡單的詢答，利用詢答的結果，搭配本研究提出的使用者權重計算公式，計算出使用者權重矩陣，再進行音樂推薦，藉此提升音樂推薦系統的效率，主要程序如下圖 3。

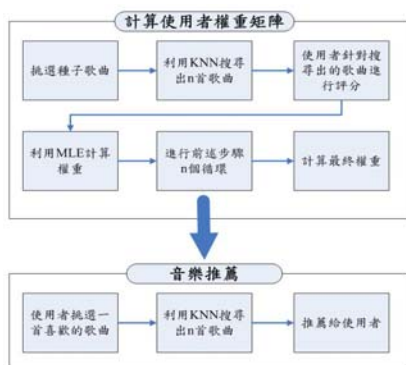


圖 3 系統流程圖

主要流程分成兩個階段，第一階段為計算使用者權重矩陣，主要目的在藉由詢答結果，找出各個特徵值對於使用者的影響程度。首先，系統隨機挑選一首歌曲做為種子歌曲。系統會以種子歌曲為中心，利用 k -NN 演算法，搜尋出 k 首歐幾里德距離最小的歌曲，並由使用者進行相似度的評分，相似度越高，歌曲的評分就越高。待使用者評分完 k 首歌曲後，挑選出評分較高的歌曲，做為計算權重的依據。上述步驟持續進行數個循環，最後再由每個循環所計算出的各維度權重，計算出最終的使用者權重矩陣。

得到使用者專屬的使用者權重矩陣後，便可藉由得到的權重，進行音樂推薦。一開始由使用者選擇一首自己想聽的歌曲與想聽的歌曲數目，接著系統會由權重式 k -NN 演算法，搜尋出適合的歌曲，並推薦給使用者。

3.1 音樂特徵值

早期的音樂特徵值，大多是演唱者、曲風、年代等資訊。這些以文字表示的音樂特徵值，在擷取的過程中，通常都必須透過人工的方式取得，因此顯的相當沒效率。不僅如此，早期以文字表示的音樂特徵值，在比較的過程

中也会有很多限制。為了解決這樣的問題，最好的解決方式是由將文字表示的資料，轉換成以數字表示的資料。因此有多位學者開始藉由數位訊號分析音訊資料，並藉由特定的計算方式，求出音樂特徵值。

以數字表示的音樂特徵值，在相似度的計算上，便可藉由簡單的距離公式，輕易的計算出各個特徵值之間的差異，並且在比較的過程中，也可以輕易的比較出差異程度的大小。相較於以文字表示的音樂特徵值，也比較不會有比較錯誤的情形出現。本研究將從音樂組成的四個主要要素 Hammer E. R. and Malcolm S. C[8]，包括音質(Tone Color)、音高(Pitch)、音長(Duration)、力度(Dynamic)中取得其中 15 種不同維度的特徵值，透過 Echo Nest Analyze API[28]，將歌曲分解成片段(Segment)，再從每個片段計算其中不同維度的特徵值，當作本研究的參數，本研究所擷取的特徵值如下 [26]：

- 從音質要素中擷取出的特徵值

1. 節奏(Tempo)：從樂曲進行的拍子速度來看，可用來表示節奏的快慢，經由 discrete wavelet transform 後的資料，計算音高資料中相鄰音符的平均時間差可得之，用以描述音樂的節奏特性。
2. 節奏信度(Tempo Confidence)：估算節奏的信度(介於 0~1)。

- 音高(Pitch)：為單音符的升降，構成音的高低，屬於音的水平變化。偵測多重音高，將音高利用 pitch histogram 統計，以描述音樂中音高的特性。我們利用 MIDINOTE 擷取特徵工具取出其中相關特徵值。

3. 平均音高(Mean Pitch)：計算總音軌音高音符的平均數。
4. 音高亂度(Pitch Entropy)：計算總音高的亂度特徵值。
5. 音高標準差(Pitch Standard)：計算音高的標準差。

- 音長(Duration)：可依振動時間的長短而定，「音長」：表振動的時間長；「音短」：表振動的時間短。同樣以 MIDINOTE 工具，取得其中能描述音樂的音長特性。

6. 平均音長(Mean Duration)：計算平均的時間長度。

7. 音長亂度(Duration Entropy): 計算總音長的亂度特徵值。
8. 音長標準差(Duration Standard): 計算音長的標準差。
9. 規則性音長變異數(Segment Duration Variance): 有規則性或重複的音長變化。
- 力度(Dynamic): 音的強弱, 依振動大小而定, 振幅越大表音越強。將使用MIDINOTE工具, 計算出Volume值, 用以代表每首歌曲的強弱特性。
10. 音量最大平均(Loudness Max Mean): 片段中最大響度變化的平均值。
11. 音量最大變異數(Loudness Max Variance): 片段中最大響度變化。
12. 音量啟始平均(Loudness Begin Mean): 片段中啟始響度變化的平均值。
13. 音量力度平均(Loudness Dynamics Mean): 片段中啟始到結束的範圍變化平均值。
14. 音量力度變異數(Loudness Dynamics Variance): 片段範圍中力度的變化。
15. 音量(Loudness): 計算總音軌中的響度。

3.2 使用者權重矩陣計算

在系統進行使用者權重矩陣計算之前, 會先針對使用者進行簡單的詢答, 藉此得到使用者權重矩陣計算的相關資料。在詢答的過程中, 使用者會針對系統挑選出的歌曲進行相似度的評分, 評分的範圍為1~10分, 相似度越高, 使用者所給的評分便越高。當使用者進行完測試的評分後, 我們便可以針對評分的結果, 計算使用者權重。在計算的過程中, 只會篩選出評分高於等於8的音樂資料, 來做進一步的使用者權重計算。

在本研究所使用的音樂特徵值中, 各個特徵值有各自不同的資料範圍。這樣的現象, 在進行權重計算的過程中, 不同的音樂特徵值在計算距離時, 可能出現不平等的結果。例如經過篩選的資料結果顯示, 音訊特徵值平均音長與音長標準差, 平均音長的值介於七十到一百五十之間, 音長標準差的值介於五至二十之間。以直覺來說, 權重計算的結果將會顯示, 音長標準差的權重將會大於平均音長, 因為以資料的分佈範圍來看, 音長標準差的資料分佈範圍較小, 因此便可推測使用者對於音長標準差的變動較敏感。

雖然這樣的推測結果不一定是正確的, 除了可能受到權重的計算方式所影響之外, 實際的資料分佈情形也有可能影響最後的計算結果。但是各個特徵值的資料分佈範圍, 確實會對最後的權重產生影響。因此在計算距離前, 我們將資料做正規化的動作, 避免出現上述的情形, 影響實驗結果。正規化的公式如公式(3), $data$ 為欲正規化的資料, D_{max} 表示該維度中, 所有資料的最大值, D_{min} 表示該維度中, 所有資料的最小值。在本研究中, 將資料正規化介於零到一百之間, 讓所有資料維度在計算的過程中, 處於一個平等的條件, 避免影響實驗結果。

$$T_{data} = \frac{(D_{max} - data) * 100}{D_{max} - D_{min}} \quad (3)$$

本研究計算使用者權重所選用的方法為最大似估計法(Maximum Likelihood Estimation, MLE)。我們假設系統隨機挑選出的種子歌曲為 q_i , 且 q_i 為 d 維密度空間上的一點。 M 為利用 k -NN搜尋出與 q_i 距離最近的 n 首歌曲集合, M 可進一步表示為 $M = \{S_i \in R^+, i = 1, 2, 3, \dots, n\}$, $S_1 \sim S_n$ 為 k -NN搜尋出的歌曲。

為了簡化運算, 我們將 $S_i, i = 1, 2, 3, \dots, n$ 之間假設為相互獨立的事件, 則發生 M 事件集合的機率則可以公式(4)表示, α 為使用者針對歌曲 S_i 的評分。

$$P(M | U_{weight}, q_i) = \prod_{S_i \in M} P(S_i | U_{weight}, q_i)^\alpha \quad (4)$$

由於 S_i 為多維密度空間的一點, 各個維度代表的是歌曲的特徵, 例如節奏、平均音高、平均音長等。因此 U_{weight} 代表的是各個維度的權重, 可將 U_{weight} 進一步表示如公式(5), w_d 所代表的是第 d 個維度的權重。

$$U_{weight} = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_d \end{bmatrix} \quad (5)$$

為了找出 $P(M | U_{weight}, q_i)$ 的最大值, 我們必須求出適當的 U_{weight} 來表示 M , 使得 $P(M | U_{weight}, q)$ 擁有最大值, 可表示如公式(6), 而 S_i 的機率密度函數, 則可表示如公式(7)。

$$\langle U_{weight}, q_i \rangle = \arg \max_{\langle U_{weight}, q \rangle} P(M | U_{weight}, q_i) \quad (6)$$

$$P(S_i | U_{weight}, q_i) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\prod_{k=1}^d r_k}} \exp\left[-\frac{1}{2} D_r(S_i, q_i)\right] \quad (7)$$

$$D_r(S_i, q_i) = \sum_{k=1}^d r_k (S_k - q_k)^2$$

本研究假設使用者對於歌曲的喜好符合高斯分佈，則我們可以利用最大概似估計法，來估計最適合 M 事件集合的 U_{weight} ，藉此得到各個維度的權重。首先我們需針對 M 取對數，藉此簡化運算，如公式(8)。接著再針對 $w_i, i=1,2,3,\dots,d$ 進行偏微分，並令其等於零，如公式(9)。

$$L(M|U_{weight}, q_i) = \sum_{s_i \in M} \alpha_k \ln P(S_i | U_{weight}, q_i) \quad (8)$$

$$= \sum_{s_i \in M} \alpha_k \left[-\frac{d}{2} \ln(2\pi) + \frac{1}{2} \sum_{k=1}^d \ln w_k - \frac{1}{2} D_i(S_i, q_i) \right]$$

$$\frac{\partial L(M|U_{weight}, q_i)}{\partial w} = 0 \quad (9)$$

$$\frac{\partial L(M|U_{weight}, q_i)}{\partial w} = \sum_{s_i \in M} \left[\alpha_k \frac{1 - w_i (S_{ki} - q_{ii})^2}{2w_i} \right] = 0$$

最後，我們可以得到簡化後的結果如公式(10)， w_i^* 為維度 l 的權重。

$$w_i^* = \frac{\sum_{s_k \in M} \alpha_k}{\sum_{s_k \in M} \alpha_k (S_{ki} - q_{ii})^2} \quad (10)$$

以此類推，我們可藉由對各個維度偏微分的結果，求出各個維度的 w ， w 代表該維度的權重， w 越大，代表使用者對於該維度的變動越敏感。我們需對使用者進行多次的詢答，因為一次的詢答並不足以代表所有使用者的喜好。不同類型的音樂，針對使用者而言，音樂特徵值的影響程度也會有所不同。因此，為了找出足以代表使用者，對於各種音樂類型特徵值影響程度的權重，我們針對使用者進行多次的詢答，並將結果帶入公式(11)，即可求出各個維度的最終權重(w_{Ui})，其中 Tn 代表的是詢答次數。

$$w_{Ui} = \frac{\sum_{t=1}^{Tn} W_t}{Tn}, Tn=1,2,3,\dots,n \quad (11)$$

3.3 權重式k-NN音樂推薦

得到使用者權重矩陣後，我們便可根據使用者權重矩陣進行音樂推薦，音樂推薦的流程如下圖 4。

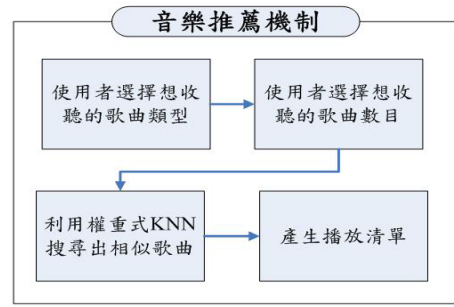


圖 4 音樂推薦機制

使用者可選擇想收聽的歌曲類型以及想收聽的歌曲數目，在歌曲類型選擇上面，我們讓使用者直接選擇歌曲，而不是音樂類型。舉例來說，我們的選擇清單可能包含多位歌手的多首歌曲，像是張惠妹的「人質」、伍佰的「Last Dance」、吳宗憲的「留心」等，而不是要求使用者選擇音樂類型，像是搖滾、鄉村等。

這樣的方式，主要是考慮到每個使用者對於音樂類型的認知不同。雖然使用者於收聽音樂時，能根據聽到的內容判斷是否為自己喜好的音樂，但是使用者卻可能不太瞭解該首歌曲的音樂類型，因為一般的使用者並不一定具備基本的音樂知識，有些人可能連基本的節奏與節拍的分辨能力都有問題。這樣的現象將會造成，專業的音樂人員與一般使用者的認知不同的情形發生。因此我們選用一個最直覺的方式，由使用者直接選擇歌曲，不僅可以避免上述的情況發生，使用者也可以直覺的選擇希望收聽的歌曲。

傳統的 k -NN 演算法，大多都是以歐幾里德距離做為距離計算公式。歐幾里德距離公式也是目前最常見的距離計算公式，不僅可以用來表示點與點之間在多維密度空間的距離，且計算的方式也較為簡單，較不會耗用系統資源。但是原本的歐幾里德距離公式並未考慮特徵值與使用者之間的影響程度，因此在本研究中，將歐幾里德距離公式加以改良，將計算出的距離乘上使用者權重，藉此顯示出特徵值與使用者之間的影響程度，改良後的公式如公式(12)。

$$d(x, y) = \sqrt{\sum_{i=1}^n w_i (x_i - y_i)^2} \quad (12)$$

我們以使用者選擇的歌曲做為種子歌曲，再藉由權重式 k -NN 演算法，依照使用者想收聽的歌曲數目，搜尋符合使用者需求的相似歌曲，最後利用隨機排列的方式，產生播放清單。

4. 實驗分析

4.1 系統環境建立與評估

本研究以Web環境為基礎實做音樂播放平台，使用者可透過瀏覽器使用雛形系統所提供的個人化音樂推薦服務。系統的開發環境為Microsoft Visual Studio 2005，開發程式語言為C#。後端資料庫系統採用Microsoft Office Access 2003，Client端為IE5.0以上瀏覽器，音樂播放器為Windows Media Player。硬體環境為Pentium 4 1.5G CPU、2G RAM，作業系統為Window XP SP2。

歌曲來源是由網路上擁有音樂背景者所歸類的MIDI歌曲，並將所蒐集到的歌曲，經由工具程式處理轉為可比對的格式後，存放於音樂資料庫中，MIDI檔歌曲共收集200首，其中類別項目如表1所示。

表1 音樂資料庫分類表

編號	風格	數量
1~35	搖滾音樂	35
36~65	爵士音樂	30
66~95	藍調音樂	30
96~130	抒情音樂	35
131~165	鄉村音樂	35
166~200	舞曲	35

4.2 系統運作流程

本研究的系統運作流程分成詢答與音樂推薦兩個階段，分別為詢答階段與音樂推薦階段。詢答階段主要針對新使用者，新使用者於登入後，需先經過系統25回合的詢答，才能享有音樂推薦的服務。詢答的目的是為了得到計算使用者權重的相關資料，增加音樂推薦的準確度。使用者完成25回合的詢答後，系統會根據使用者接受詢答所得到的評分資料，計算使用者與每個特徵值之間的權重。

使用者完成25回合詢答後，便可選擇喜歡的歌曲與想要收聽的歌曲數目，系統會根據使用者輸入的歌曲與數目，搭配使用者權重，搜尋出適合的推薦歌曲。使用者可根據推薦的歌曲進行評分，提供系統做為日後統計與分析等其他用途。

4.3 實驗結果與分析

本研究的實驗對象共63人，經由統計的結果得知男女的比例約為7：3，年齡分佈於23歲到35歲之間，實驗對象皆為網路高度使用族群。目前系統仍然持續接受使用者註冊，希望藉此得到更多的樣本資料以供進一步了解與分析。

為了驗證本研究所提出的基於「權重式歐幾里德距離」為概念的音樂推薦系統架構，將透過實證方式評估實驗結果。評估系統的準則我們將採用推薦準確率(Precision)指標。推薦準確率(Precision)指的是推薦系統經由各種的推薦機制，建構出推薦模組後，此模組對於測試資料能準確推薦之比例。換句話說，也就是系統所推薦的項目當中，有多少的比例是使用者當前有興趣的。準確率的公式定義如下：

$$\text{Precision Ratio} = \frac{\text{Number of accept items}}{\text{Number of recommend items}} * 100\% \quad (13)$$

推薦準確率公式中，Number of Recommend Items指的是系統推薦給使用者的推薦數量，Number of Accept Items指的是使用者接收推薦的數量。本研究中，推薦的項目為音樂，因此Number of Recommend Items所代表的是系統推薦給使用者的歌曲數目，Number of Accept Items指的是使用者接受推薦的歌曲數目。推薦準確率指標已被廣為用在各類推薦系統的評估標準上，例如Herlocker E. R.[9]、Sarwar B. M.[22]、Vucetic S.[30]、Yu K.[34]，以下我們將利用上述評估指標分別對各實驗作詳細說明，並在最後列出我們的實驗結果與分析討論。

- 實驗一、比較詢答次數對推薦準確率的影響

在本實驗中，將針對每位使用者進行25回合的詢答。系統於每回合詢答階段，隨機挑選一首歌曲做為種子歌曲，並以種子歌曲做為依據，利用k-NN演算法搜尋出與種子歌曲距離最近的15首歌曲，再以隨機排列的方式產生歌曲數目為15的詢答清單。使用者需針對詢答清單中的歌曲進行相似度評分，評分的依據為比較種子歌曲與詢答清單中歌曲的相似度，相似度越高便給予越高的評分，評分範圍為1~10分。

使用者完成25回合的詢答後，系統根據使用者的評分，分別計算出進行5、10、15、20、

25回合的詢答所得到的使用者權重，如下表2。接著我們要求使用者選擇一首喜好的歌曲，之後系統根據使用者挑選的歌曲產生推薦清單。

表2 使用者權重矩陣與總計詢答回合數對照表

使用者權重矩陣	詢答回合	總計詢答回合數
5's U_{weight}	1~5	5回合
10's U_{weight}	1~10	10回合
15's U_{weight}	1~15	15回合
20's U_{weight}	1~20	20回合
25's U_{weight}	1~25	25回合

推薦清單的產生方式，是根據表2中的五個使用者權重矩陣，利用權重式 k -NN演算法，個別產生一組歌曲數目為10的推薦清單，再將所有的歌曲以隨機的方式排列，並剔除掉重複的歌曲。換句話說，每首歌曲在推薦清單中只會出現一次，而每次彙整後的推薦清單歌曲數目是不固定的。我們針對每位使用者進行十次的不同歌曲的推薦，所得到的統計結果如下圖5。



圖5 比較詢答次數與推薦準確率

由實驗結果可以看出，系統的推薦準確率隨著詢答回合數的增加而增加。且在25回合的詢答下所得到的使用者權重矩陣，已可讓系統擁有83%的推薦準確率。由此可見，詢答回合數的增加，可以有效的增加推薦準確率。主要原因在於，多回合的詢答，讓使用者可針對不同類型的歌曲進行詢答，因此所得到的使用者權重矩陣可有效的應用於不同的歌曲類型上。

- 實驗二、比較詢答清單歌曲數目對推薦準確率的影響

在這次的實驗，我們繼續延用上一個實驗

中，系統於詢答階段所得到的回饋資料。從實驗一的結果可以發現，使用者於進行25回合的詢答後，具有最佳的推薦準確率，因此我們以25回合的詢答做為基礎，計算實驗二的使用者權重矩陣。

在實驗一中，我們提供使用者15首歌曲的詢答清單，且這15首歌曲是利用 k -NN演算法搜尋出與種子歌曲距離最接近的15首歌曲。因此，我們延用實驗一所得到的資料，分別擷取3NN、6NN、9NN、12NN與15NN的評分資料，計算當詢答清單歌曲數目為3、6、9、12、15時的使用者權重矩陣，如下表3。

表3 使用者權重矩陣與詢答清單歌曲數對照表

使用者權重矩陣	詢答清單歌曲數	總計詢答回合數
3NN's U_{weight}	3	25
6NN's U_{weight}	6	25
9NN's U_{weight}	9	25
12NN's U_{weight}	12	25
15NN's U_{weight}	15	25

當系統得到於不同詢答清單歌曲數目的情況下，所計算的使用者權重矩陣後，我們依據實驗一的推薦清單產生方式，產生一組推薦清單，並要求使用者針對推薦的歌曲進行喜好的評分。在每位使用者進行十次的不同歌曲的推薦並進行評分後，所得到的統計結果如下圖6。

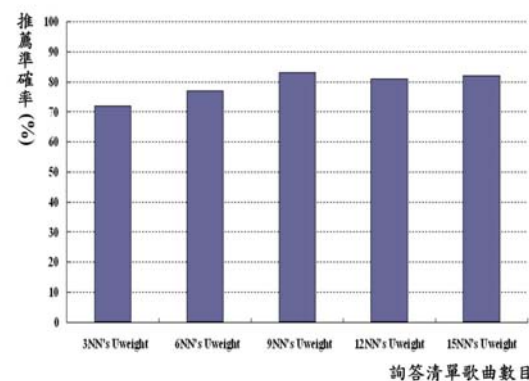


圖6 比較詢答清單歌曲數目與推薦準確率

實驗結果顯示，除了使用3NN's U_{weight} 的進行推薦清單運算，所得到的實驗結果較差外，其他的實驗結果皆有80%左右的推薦準確率。進一步比較下，我們可以發現，使用9NN's U_{weight} 、12NN's U_{weight} 與15NN's U_{weight} 進行推薦清單運算時，推薦準確率皆可達到80%以上，

其中又以9NN's U_{weight} 進行推薦清單運算時，所得到的推薦準確率最高，但是與其他兩種情況比較下，差距其實並不大。

由此可見，在詢答階段中，詢答清單歌曲數量的增加，並無法保證可以提升推薦準確率。主要原因在於，詢答清單歌曲數目增加，並不表示可提供計算使用者權重矩陣的資料增加。換句話說，系統進行使用者權重運算時，僅會篩選出使用者進行相似度評分時較高的歌曲，不符合標準的歌曲便不會提供給系統進行使用者權重計算。因此詢答清單的歌曲數目增加，雖然可以增加使用者詢答時的歌曲，但是增加的歌曲與種子歌曲相似度差異過大，使得使用者給予過低的評分，導致無法將資料納入使用者權重矩陣的運算過程。

● 實驗三、 k -NN與權重式 k -NN推薦準確率之比較

由實驗一與實驗二的結果可以得知，在詢答階段中詢答回合數與詢答清單歌曲數目，皆會影響後續的系統推薦準確率。接著我們進一步比較權重式 k -NN與傳統的 k -NN為基礎的音樂推薦系統之間推薦準確率的差別。

首先，我們要求使用者選擇一首喜好的歌曲做為種子歌曲，再分別利用權重式 k -NN與傳統的 k -NN演算法搜尋出15首與種子歌曲最相似的歌曲，並將兩種演算法搜尋出的歌曲混合，並剔除掉重複的歌曲，最後以隨機排列的方式產生推薦清單。

在這次的測試實驗中，我們使用實驗二中的9NN's U_{weight} 做為使用者權重矩陣。我們重複進行15次的實驗，並要求使用者針對推薦的歌曲內容進行喜好的評分，評分的內容分為喜歡與不喜歡兩種選項。最後再分別比較當搜尋歌曲數為3、6、9、12與15首歌曲時，權重式 k -NN演算法與傳統的 k -NN演算法，在推薦準確率上的差別，我們所得到的統計數據如下圖7。

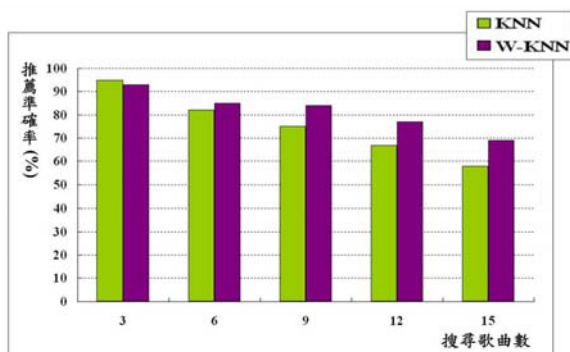


圖 7 k -NN 與權重式 k -NN 推薦準確率之比較

將評分資料細分並由統計結果可以發現，權重式 k -NN的推薦準確率明顯大於傳統的 k -NN演算法。傳統的 k -NN演算法雖然在前3、6首歌曲的評分仍有高於80%的推薦準確率，但是搜尋出的歌曲增加時，推薦準確率便開始明顯的下滑，在前12與15首歌曲的評分上，已經只剩五、六十百分比的推薦準確率。反觀權重式 k -NN演算法，雖然在前15首歌曲的推薦準確率只有接近70%左右，但是其他情況下，推薦準確率仍有80%左右。且隨著推薦歌曲數目的增加，權重式 k -NN演算法推薦準確率下降的幅度，也小於傳統的 k -NN演算法。

5. 結論與未來研究方向

傳統的音樂推薦系統，很少考慮到使用者與特徵值之間的影響程度。在本研究中，利用簡單的詢答所得到的回饋資料，計算出使用者權重矩陣，藉此提升音樂推薦系統的推薦準確率。實驗結果證明，這樣的方式確實有不錯的成效。不僅可以清楚的藉由使用者權重矩陣得知各個特徵值對於使用者的影響程度，更可瞭解各個特徵值的重要性。

本研究所提出的使用者權重矩陣，其實是一種概念性的方法，可以應用於多種類型的推薦系統中。加入使用者權重矩陣的概念，不僅不會改變原有推薦系統的程序，更可增加各個特徵值的準確度，讓系統於推薦時，擁有更可靠的依據。

在後續的發展中，可以比較使用者權重矩陣，在不同的推薦機制，是否會影響推薦準確率。例如使用者權重矩陣更提供合作式過濾推薦系統做為特徵比對的依據，藉此提升合作式過濾推薦系統的推薦準確率。另一方面，本研究於計算使用者權重矩陣前，並無針對相關資料進行分群，目的在於不希望因為分群的差別，影響實驗的結果。但是這也導致在本研究中，並無法比較使用者權重矩陣於不同群體之間的差別，希望在後續的研究中，能繼續針對此一問題做進一步的討論。

Acknowledgement

This work was partially supported by the NSC in Taiwan under the contract numbers NSC 98 - 2200 - E - 020 - 004 and NSC98-2221-E-020-025-MY2.

參考文獻

- [1] 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.
- [2] Balabanovic, M. and Shoham, Y., "Fab: Content-Based, Collaborative Recommendation," *Communications of the ACM*, Vol.40, pp. 66-72, 1997.
- [3] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and Sartin, M., "Combining Content based and Collaborative Filters in an Online Newspaper," *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation. University of California*, Berkeley, 1999.
- [4] Daniel, M. and Sally, J. C., "Sociology And Music Recommendation Systems," *Austrian Computer Society(OCG)*, 2007.
- [5] Delgado, J., Ishii, N. and Ura, T., "Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents," *In Proc. Second Int. Workshop*, CIA'98, 1998.
- [6] Euclidean distance, http://en.wikipedia.org/wiki/Euclidean_distance, Retrieved date: 2009/10/21, 2009.
- [7] Goldberg, D., D., Nichols, B. M., Oki, and Terry, D., "Using Collaborative Filtering to Weave an Information Tapestry," *Communications of the ACM*, 1992.
- [8] Hammer, E. R. and COLE, M. S., "*Guided listening; a textbook for music appreciation.*", 1992.
- [9] Hammer, E. R. and Malcolm, S. C., "Guided listening; a textbook for music appreciation," 1992.
- [10] Hayes, C. and Cunningham, P., "Smart Radio-Building Music Radio On the Fly", *In Proceedings of Expert Systems*, 2000.
- [11] iLike.com, <http://www.ilike.com>, Retrieved date: 2009/11/5, 2009.
- [12] 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.
- [13] Konstan, J. A., Miller, B. N. and Maltz, D., "GroupLens: Applying collaborative filtering to Usenet news," *Communications of the ACM*, Vol.40, No.3, pp.77-87, 1997.
- [14] 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.
- [15] k-nearest neighbor algorithm, http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm, Retrieved date: 2009/11/5, 2009.
- [16] Lang, K., "Newsweeder : Learning to Filter Netnews," *Proceedings of the 12th International Conference on Machine Learning*, pp.331-339, 1995.
- [17] Last.fm, <http://www.last.fm>, Retrieved date: 2009/10/6, 2009.
- [18] 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.
- [19] Minkowski distance, http://en.wikipedia.org/wiki/Minkowski_distance, Retrieved date: 2009/10/21, 2009.
- [20] Pandora.com, <http://www.pandora.com>, Retrieved date: 2009/10/29, 2009.
- [21] Rucker, J. and Polanco, J. M., "Siteseer: Personalized Navigation for the Web," *Communications of ACM*, Vol.40, No.3, pp.73-75, 1997.
- [22] Sarwar B. M., Karypis G., Konstan, J. A. and Riedl, J., "Application of Dimensionality Reduction in Recommender System-A Case Study," *ACM Web KDD Web Mining for E-Commerce Workshop*, 2000.
- [23] Scheirer, E. and Slaney, M., "Construction and evaluation of a robust multifeature speech/music discriminator, " *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1331-1334, 1997.
- [24] Steffen, P. and Sander V. W., "User evaluation of a new interactive playlist generation concept," *International Conference on Music Information Retrieval, ISMIR*, pp.638-643, 2005.

- [25] Shardanand, U. and Maes, P., "Social information filtering: algorithms for automating "word of mouth"," *Proceedings of the Conference on Human Factors in Computing Systems* , Denver, CO, ACM, pp.210-217, 1995.
- [26] Slaney, M., Weinberger, K. and White, W., "Learning a Metric for Music Similarity", *ISMIR*, 2008.
- [27] Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J., "PHOAKS: A System for Sharing Recommendations," *Communications of ACM*, Vol.40, No.3, pp.59-62, 1997.
- [28] The Echo Nest Analyze, <http://developer.echonest.com/docs/analyze/xml>, Retrieved date: 2009/10/21, 2009.
- [29] Tzanetakis, G. and Cook. P., "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, Vol.10, No. 5, pp. 293-302, 2002.
- [30] Vucetic, S. and Obradovic, Z. "A regression-base approach for scaling-up personalized recommender systems in e-commerce," *ACM Web KDD Web Mining for E-Commerce Workshop*, 2000.
- [31] Wang, H. W., Wang, J., Yang, and Yu, P. S., "Clustering by Pattern Similarity in Large Data Sets", *Proceeding of ACM Special Interest Group on Management of Data*, 2002.
- [32] Wasfi, A. M. A., "Collecting User Access Patterns for Building user Profiles and Collaborative Filtering," *In Int. Conf. On Intelligent User Interfaces*, 1999.
- [33] Wrigley, S. N., Brown, G. J., Wan, V. and Renals, S., "Speech and crosstalk detection in multichannel audio," *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 1, pp.84-91, 2005.
- [34] Yu K., Xu, X., Ester, M. and Kriegel, H.P., "Selecting relevant instances for efficient accurate collaborative filtering," *Proceedings of the 10th International conference on Information and knowledge management*, pp.239-246, 2001.