

# 情境式個人化穿著推薦系統

王宗一 邱迪凱 鍾博欽 張維高

國立成功大學 工程科學系

智慧型網路應用實驗室

wti535@mail.ncku.edu.tw

## 一、前言

### 摘要

電子商務在網路上的茁壯，讓消費者可以在網路上方便消費，但面對琳瑯滿目的商品，消費者不易於選擇符合他們所需求的產品。而推薦系統的發展便是為了方便消費者在龐大的可能選擇之中找尋符合他們的需求。推薦系統發展至今應用廣泛，從電子商務的服務到推薦書籍、餐廳、電玩、旅行路線等等現實生活上的需求皆有涵蓋，但是推薦穿著搭配這一環是較少人觸及的。在現實生活裡，人們想了解不同場合所適合的穿著，自己的身材該如何穿，怎樣的顏色搭配比較好，這都是使用者在搭配穿著上所遭遇到困難，人們因此會去詢問別人意見或是參考時尚雜誌，從這樣的行為下人們所找尋的資訊都是具體的衣服屬性上的對應，例如顏色、廠牌、材質、大小和種類。根據這樣的想法，本論文提出一個情境式個人化穿著推薦系統，系統透過收集使用者所描述的具體之衣服屬性、使用者的偏好和色彩學意象上的分析來建立起使用者的個人穿著知識，幫助使用者快速找尋針對特定情境下適合且喜愛的穿著。

**關鍵詞：**推薦系統、協同過濾式推薦、內容導向式推薦、色彩意象。

網際網路的發展越來越蓬勃和 Web 2.0[17]的概念下，人們習慣利用網路來達到資訊分享、學習、溝通的目的。因此從詢問朋友意見到閱讀雜誌的行為模式，都因為網路的發展慢慢的改變人們的行為，又由於數位相機的普及，拍攝照片後只需要透過個人電腦處理，很容易將所拍攝的照片與人分享。例如部落格(Blog)不管是明星、知名人士或是一般人，常常會提供自己的穿著照片，放置自己的部落格提供自己的一點穿著意見；網路上也有許多討論穿著的論壇或電子佈告欄，仿造雜誌的形式，透過上傳照片，詢問別人穿著意見，討論著自己的穿著與別人的穿著是否合適。人們接收資訊的選擇變多了，如此一來卻面臨資訊過載(Information overload)的問題，人們必須經由大家的意見整理出自己想要資訊，而推薦系統(Recommendation System)發展便可以解決資訊過載的問題。

## 二、相關文獻探討

### (一) 推薦系統(Recommendation System)

推薦系統發展的主要目的為解決資訊過載的問題和提供使用者個人化的推薦、內容和服務。近年來已經廣泛的研究與應用，像推薦書籍和 CDs 的 Amazon.com[6]、推薦電影的 MovieLens[1]

和推薦新聞資訊的 VERSIFI Technologies (AdaptiveInfo.com 的前身) [3]，此外像 Microsoft 已經整合推薦系統在他們的商業服務裡[2]。

Resnick 等人[12]提出推薦系統的解釋為：「使用者提供推薦的項目給系統當輸入，系統收集這些資訊並加以處理，再提供合適的項目給接受者。」，Burke[14]解釋為：「任何一個可以產生個人化推薦結果的系統，或是能在大量的可能選擇中、以個人化方式引導使用者找到感興趣或有用項目的系統，皆可稱為推薦系統。」另外，Herlocker[9]解釋為：「推薦系統採用使用者群體的意見，來協助社群中的個體更有效地從潛在的大量選擇中辨識出感興趣的內容。」Sweetser 和 Wyeth[13]解釋推薦系統的功能為：運用由專家編寫的資訊或是從觀察消費者行為所學習而來的消費者意向知識，來引導處理消費者選擇不知所措的產品上。」，隨其逐漸發展至今日、推薦系統已涵蓋比起以往更廣的範圍。

在推薦系統中，通常可根據推薦方式分類如下：

- **內容導向式推薦 (Content-based Recommendation)**

系統以使用者過往使用的物品內容資訊做為推薦的依據。內容導向式系統經常利用關鍵字(Keywords)來描述內容特徵，例如 Fab[6]系統便以網頁內容中 100 個最重要得詞彙作為代表推薦網頁給使用者，另外像 Syskill & Webert 系統[10]則是利用 128 個最重要的字來描述文件。像這樣在文件中定義關鍵字重要性的權重問題，在資訊擷取概念中最著名的方法即為 TF-IDF(Term Frequency/Inverse Document Frequency)[7]，以計算詞彙出現頻率為主要概念，可視為內容式導向式推薦系統的基礎。假設  $N$  為所有可以推薦給使用者的

文件數， $n_i$  為文件有關鍵字  $k_j$  出現的文件數，假設  $f_{i,j}$  為關鍵字  $k_j$  在文件  $d_j$  所出現的次數，則可以定義關鍵字  $k_j$  出現在文件  $d_j$  的詞頻， $TF_{i,j}$  如式子(2-1)：

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \quad (2-1)$$

計算式(2-1)中  $\max_z f_{z,j}$  為全部關鍵字  $k_z$  有出現在文件  $d_j$  中最大的出現頻率。關鍵字出現在許多文章裡，是沒辦法區分文章的重要性，因此得建立反向的文件頻率  $IDF_i$  來結合  $TF_{i,j}$ ，關鍵字  $k_j$  的反向的文件頻率為式子(2-2)：

$$IDF_i = \log \frac{N}{n_i} \quad (2-2)$$

，如此一來便可以建立出關鍵字  $k_j$  在文件  $d_j$  的 TF-IDF 權重如式子(2-3)：

$$w_{i,j} = TF_{i,j} \times IDF_i \quad (2-3)$$

因此可以定義出文件  $d_j$  的內容為式子(2-4)：

$$Content(d_j) = (w_{1j}, \dots, w_{kj}) \quad (2-4)$$

相較於傳統資訊擷取的方法，另一種增進內容導向式推薦系統的方法為建立使用者描述記錄(User Profiles)，而使用者描述記錄的內容包含使用者的喜好、興趣和需求，完整的描述記錄能夠從使用者身上尋找出重要資訊。將使用者描述紀錄定義如式子(2-5)：

$$ContentProfile(c) = (w_{1c}, \dots, w_{kc}) \quad (2-5)$$

式子(2-5)表示為使用者  $c$  的使用者描述紀錄，為一個使用者描述關鍵字集合的權重向量，可以根據使用者評分的紀錄，來計算評過分數物品中關鍵字的權重，例如 Rocchio 演算法[8]則是利用平均的方式來計算權重，在內容導向式推薦系統，利用效用函數  $u(c, s)$  來表示值得推薦的程度，定義效用函數如式子 (2-6)：

$$u(c, s) = \text{score}(\text{ContentProfile}(c), \text{Content}(s)) \quad (2-6)$$

其中  $\text{ContentProfile}(c)$  和  $\text{Content}(s)$  各以  $TF-IDF$  向量  $\vec{w}_c$  與  $\vec{w}_s$  來表示關鍵字的權重，利用餘弦相似度來計算兩者之間的相似度，公式(2-7)如下：

$$u(c, s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2} = \frac{\sum_{i=1}^K w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^K w_{i,c}^2} \sqrt{\sum_{i=1}^K w_{i,s}^2}} \quad (2-7)$$

(2-7)式子中的  $K$  為系統中所有關鍵字的總數。

- **協同過濾式推薦 ( Collaborative Filtering Recommendation ) :**

協同過濾式推薦系統最早於 1992 年 Goldberg 等人於 Tapestry 系統[4]所提出的概念，在 1994 年 Resnick 等人於 GroupLens 系統中首次採用計算鄰近群體的演算法 [20,19]。此方法系統會藉由紀錄使用者過去對物品的評價及喜好歷程，找出與該使用者喜好相似程度高的使用者形成群集，在由群集對物品的評價推薦給使用者。以一個電影推薦系統為例，為了推薦電影給使用者，協同過濾式推薦系統根據對使用

者對電影的嘗試(對電影的評分)，找到一群具有相似興趣的使用者後，唯有在此相似使用者群集中高度評價的電影才會被推薦給未看過得使用者。

根據使用者過去所評分過的項目集合為基礎，進行預測的啟發方式。使用者  $c$  對於項目  $s$  未知的評分為  $r_{c,s}$ ，通常聚集 (Aggregate)  $N$  個最相似使用者對相同項目  $s$  的評分計算如式子(2-8)所得：

$$r_{c,s} = \text{agg} r_{c' \in \hat{C}} r_{c',s'} \quad (2-8)$$

式子 (2-8) 中  $\hat{C}$  表示與使用者  $c$  最相似且已對項目  $s$  評分的  $N$  個使用者。聚合函數 (Aggregate Function) 可以採用任意函數，將會產生不同的估計值，最常見的聚合函數(2-9)如下：

$$\begin{aligned} (a) \quad r_{c,s} &= \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s'} \\ (b) \quad r_{c,s} &= k \sum_{c' \in \hat{C}} \text{sim}(c, c') \times r_{c',s'} \\ (c) \quad r_{c,s} &= \bar{r}_c + k \sum_{c' \in \hat{C}} \text{sim}(c, c') \times (r_{c',s'} - \bar{r}_{c'}) \end{aligned} \quad (2-9)$$

式子 (2-9) 中乘數  $k$  為正規化參數 (normalizing factor)，通常會被選定為

$$k = \frac{1}{\sum_{c' \in \hat{C}} |\text{sim}(c, c')|}, \bar{r}_c \text{ 表示使用者 } c \text{ 的平均}$$

評分。公式 (2-9a) 為單純計算所有對該項目評分之相似使用者的平均評分；公式(2-9b) 以兩個使用者之間相似度高低做為評分權重衡量因素，相似度越高的使用者給予的評分權重也越大；公式(2-9c) 則對使用者評分標準範圍不一的問題做處理，將評分給予正規化，在調整過後的結果進行加權計算。本論文為採取公式(2-9c)的分法。另外

也有喜好導向過濾(Preference-based)的方法可以解決不同使用者評分尺度不一的問題，著重於預測相對的喜好高低，並非純粹的評分值。

在協同過濾式推薦系統裡，為了要計算兩個使用者相似度 $sim(c, c')$ 的方法，通常會以兩個使用者皆評過的項目做為依據進行衡量，最普遍採用的兩種方法為關聯法(Correlation)與餘弦法(Cosine-based):

- 關聯法

$S_{xy}$  代表使用者  $x$  與使用者  $y$  共同評分過得項目集合，以 Pearson 相關係數來計算使用者  $x$  與使用者  $y$  的相似度如式子(2-10)所示：

$$sim(x, y) = \frac{\sum_{c \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{c \in S_{xy}} (r_{xs} - \bar{r}_x)^2 \sum_{c \in S_{xy}} (r_{ys} - \bar{r}_y)^2}} \quad (2-10)$$

- 餘弦法

將使用者  $x$  與使用者  $y$  表示為在  $m$  維空間中的兩個向量 $\vec{x}$ 和 $\vec{y}$ ， $m = |S_{xy}|$ ，則可以計算兩個向量的餘弦值表示兩者的相似度如式子(2-11)所示：

$$sim(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \times \|\vec{y}\|_2} = \frac{\sum_{c \in S_{xy}} r_{xs} r_{ys}}{\sqrt{\sum_{c \in S_{xy}} r_{xs}^2} \sqrt{\sum_{c \in S_{xy}} r_{ys}^2}} \quad (2-12)$$

- 混合式推薦 (Hybrid recommendations)：

混合式推薦方式是將內容導向式推薦及協同過濾式推薦方法合併應用，主要目的是避免兩種方式單獨實行上會遭遇的問題與限制，以取得較佳的推薦結果，合併的方法可分類如下：

1. 合併個別的推薦結果

分別運用內容導向式推薦與協同過濾式推薦方法，產生個別的推薦結果，再將推薦結果合併。從個別的推薦系統獲得推薦結果後，以線性權重結合計算分數或是投票制決定最後結果[11]。

2. 將內容導向式推薦的特性加入至協同過濾式推薦

基於協同過濾式推薦方式，並建立每位使用者內容導向式推薦的個人描繪，以這些使用者的個人描繪為輔助來計算兩個使用者的相似度。像[11]文獻中的動機就是利用此方式來解決協同過濾式推薦系統有稀疏性的缺點。

3. 將協同過濾式推薦的特性加入至內容導向式推薦

利用一群使用者之間創造出共同觀點的描繪行為，作為推薦基礎，可以改進純內容導向式推薦的效能。

4. 建立單一的聯合推薦模型

找尋兩推薦方法的特性建立起單一規則的推薦方式。例如採用機率方式合併兩種推薦，採用貝氏混合效應回歸模型(Bayesian mixed-effect regression models)等等...

## (二)色彩學

在自然界裡人們對於各種物體的認知中，色彩扮演一個極為重要的角色，從自然界的啟示裡，色彩給了我們許多強烈的暗示。因此，人類開始將色彩應用在自己的環境上、身體上和穿著上等等；人對於色彩的了解往往是由傳統的見解、風俗習慣、生活經驗、個人人格個性和直覺做為色彩認知和色彩配色上的運用[15]，色彩學的研究就單色與配色意象方面已經有概略性的評價，在相關研究應用方面已經有模糊理論、感性工學、灰色理論和類神經遺傳演算法等等...。形容詞影像比例

[16][18]，當人們用影像來表達顏色時，常用「可愛的」或「柔和的」等形容詞做描述，把顏色和一般用來表現該顏色的形容詞加以連結成果。這樣的形容詞影像比例 [16][18]，以各種視覺材料為實證進而製作出配色符號，也說明陳列出多種材質的衣服並對衣服進行評價時，以一般人來說雖然這會讓人猶豫不決，但是如果排列出各種顏色並詢問「這個顏色怎樣？」的話，不論誰都可以說出意見，由此可知在穿著上色彩學的應用絕對是直覺且實用的。

### 三、系統架構

本章節介紹此論文所採用的研究方法，並提出一個情境式個人化穿著推薦系統 (Contextual Personalized Wearing Recommendation System)。情境式個人化穿著推薦系統主要的目的是將時尚穿著推薦導入推薦系統裡，使用者藉由與情境式個人化穿著推薦系統的互動，例如提供穿著照片、對照片編輯單品資訊和對其他使用者所提供的照片回饋。情境式個人化穿著推薦系統會根據使用者所提供穿著資訊與對其他使用者的穿著回饋推薦穿著給使用者，並根據使用者個人衣櫃內的單品配對相似的穿著給使用者。本研究的系統架構圖如圖 3-1 所示：

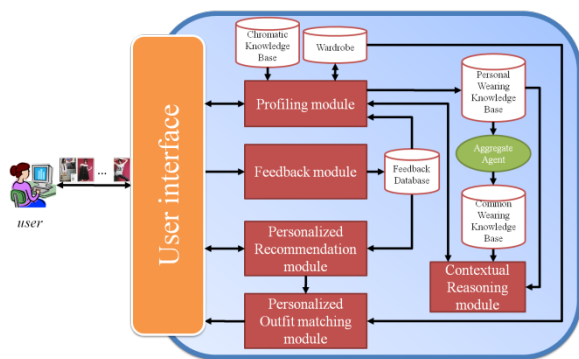


圖 3-1 系統架構圖

#### (一) 色彩學知識庫(Chromatic Knowledge Base)

此資料庫是採用 [16]，根據裡面所提供的配色，我們採用 2653 種配色，26 種配色語意建立出本系統的色彩學知識資料庫。

#### (二) 個人衣櫃(Wardrobe)

個人衣櫃主要是要紀錄使用者所提供的穿著與穿著資訊，在情境式個人化穿著推薦系統中使用者是用一張照片來表示自己的穿著，透過系統編輯單品來提供穿著資訊，這些資訊我們都會紀錄在個人衣櫃資料庫裡。使用者所提供的穿著與單品實做資料表，可示意如圖 3-2：

Outfits		Items	
PK	OutfitID	PK,FK1	OutfitID
	Presenter ImgName Style Occasion Update		ItemID Type Brand Color Size Material ...

圖 3-2 個人衣櫃資料庫實作

此資料表上詳細所紀錄的單品屬性有種類 (type)、品牌 (brand)、顏色 (color)、大小 (size) 和材質 (material)。當編輯完後，系統會根據所編輯的穿著資訊推論出建議的情境給使用者，使用者可以不接受此建議而改變情境，在穿著都會將此資料紀錄起來。

#### (三) 描繪模組(Profiling Module)

描繪模組主要的目的是要描繪出使用者的個人穿著知識，描繪模組的示意如圖 3-3 所示：



圖 3-3 描繪模組

如圖 3-4 所示描繪模組的主要資料來源為個人衣櫃內的穿著和對別人穿著的回饋。首先從個人衣櫃內的穿著來說明，如上小節所述，當使用者編輯完穿著資訊時，系統會建議情境給使用者，使用者可以贊同或是不贊同，以下我們先定義一個個人屬性值(Personal Attribute Value, PAV)如式子(3-1)所示：

$$PAV_{a,c}^u = [Points|Count] \quad (3-1)$$

式子(3-1)中  $PAV_{a,c}^u$  表示使用者  $u$  在情境  $c$  下對屬性  $a$  接受程度，其中  $Points$  總得分， $Count$  為總共計算的次數。每次編輯穿著後，當系統給情境建議時，若使用者接受供照片的使用者所描述的情境則我們做遞增，不贊同則做遞減，對改變的情境

做遞增。例如  $PAV_{襯衫,正式風}^{Amy} = [10|15]$  表示說使用者  $Amy$  的屬性襯衫在正式風的 15 次統計中共得 10 分。

本系統採取 RGB 顏色系統，在此系統裡，如 #FF0000 和 #FE0000 都是紅色，在肉眼上並分辨不出，所以我們必須計算顏色上的相似度處理，顏色相識度計算如式子(3-2)所示：

$$RGBSIM(color_1, color_2) = 1 - \frac{\sqrt{(r_1-r_2)^2+(g_1-g_2)^2+(b_1-b_2)^2}}{MaxED} \quad (3-2)$$

式子(3-2)表示為  $color_1, color_2$  的相似度， $MaxED$  為 RGB 系統的最大歐式距離 (Euclidean Distance) 即為白色 (#FFFFFF) 和黑色 (#000000) 的歐式距離約為 441.67， $RGBSIM$  的值介於在 [0,1] 區間，1 為最相似。利用此相似度與色彩學

知識庫比對，找尋一組最像似的相關形容詞。

#### (四) 情境推論模組(Contextual Reasoning Module)

此模組最主要的功能，是根據使用者所編輯的穿著資訊，來推測建議該穿著所適合的情境給使用者。我們先定義一個共同屬性值(Common Attribute Value, CAV)式子(3-3)所示：

$$CAV_{a,c} \in \{H, M, L\} \quad (3-3)$$

如式子(3-3)中  $CAV_{a,c}$  表示為在屬性  $a$  在情境  $c$  下的共同屬性值為一個高、中或低程度。高代表為屬性  $a$  在情境  $c$  下是大眾所接受度為強烈的。情境推論模組如圖 3-4 所示：

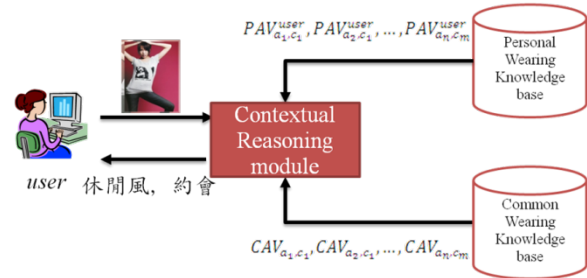


圖 3-4 情境推論模組

如圖 3-4 所示，情境推論模組會去查詢個人穿著知識庫與共同穿著知識庫內的所有屬性值來計算出合適的情境給使用者。收集整理一般人的穿著常識，手動建立基本的共同穿著知識作為系統的初始設定。另外必須將個人穿著知識與共同穿著知識做相對應動作，其相對應的式子(3-4)如下：

$$mappingValue(PAV_{a,c}^u) = \begin{cases} H, & \frac{Points}{Count} > 0.67 \\ M, & 0.33 \leq \frac{Points}{Count} \leq 0.67 \\ L, & \frac{Points}{Count} < 0.33 \end{cases}$$

(3-4)

將高、中和低給予對應的值分別為 3 分、2 分和 1 分，利用如圖 3-5 所示的方式來推論情境給使用者。

配色形容詞：時尚的

種類: T-shirt	休閒	...	運動	
品牌: NET	T-shirt	H	...	H
大小: 合身	NET	H	...	L
顏色: #FFFFFF	合身	H	...	L
材質: 棉質	牛仔長褲	H	...	L
種類: 牛仔長褲	單寧	H	...	L
品牌: NET	棉質	H	...	H
大小: 合身	時尚的	M	...	M
顏色: #000000		2.85		1.71
材質: 丹寧				

圖 3-5 情境推論

### (五) 回饋模組(Feedback Module)

此模組主要是在收集使用者的回饋，本系統的回饋資料主要有 2 種，一為使用者純粹對照片上穿著的美觀程度，另外為情境的認同。收集回饋的主要目的為建立使用者的個人穿著資訊和提供給個人化推薦模組計算產生推薦給使用者。

### (六) 個人化推薦模組 (Personalized Recommendation Module)

此模組的輸入為一群穿著的集合，主要的目的在推測使用者所感興趣的穿著和此穿著是否適合使用者的身材。此模組為一個混合式推薦系統，主要有三個子模組，分別為身材相似模組(Body Similarity, BS)、內容導向式(Content-based, CB)和協同式過濾式(Collaborative Filter, CF)，如圖 3-6 所示：

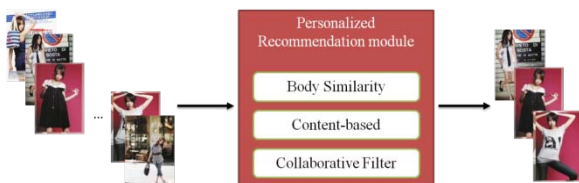


圖 3-6 個人化推薦模組

本系統的身材相似的計算方式，利用身體質量指數(Body Mass Index, BMI)，此指數是世界衛生組織(WHO)推薦作為一個利用身高與體重的比例來評估肥胖指標的重要依據。其 BMI 公式(3-5)如下：

$$BMI = \frac{W}{H^2} \quad (3-5)$$

利用 BMI 求出身材相似度公式(3-6)如下：

$$BodySim(u_i, u_j) = \begin{cases} \frac{BMI_i}{BMI_j}, & \text{if } \frac{BMI_i}{BMI_j} \leq 1 \\ 1 - \left| 1 - \frac{BMI_i}{BMI_j} \right|, & \text{if } \frac{BMI_i}{BMI_j} > 1 \end{cases} \quad (3-6)$$

第二個子模組為內容導向式。首先情境式個人化穿著推薦系統會動態的建立使用者個人喜好檔案(User Preference Profile, UPP)，從使用者的個人衣櫃中找尋屬於該情境的穿著及評過別人的穿著中分數大於 3 的穿著，利用圖 3-7 方式將所有屬性建立出個人喜好檔案另外我們為了要提高推薦的精確度，將每一個穿著分成六個主要部份為頭部、上半身、下半身、外套、腳部及配件，將每個屬性依照這六個部份分開計算再取一個平均值，因為外套與褲子來比對，似乎沒有意義。



圖 3-7 UPP 建立

當情境式個人化穿著推薦系統建立出個人喜好檔案後，計算個人樣式與穿著的相似度計算，其計算演算法如圖 3-8 所示：

```

OutfitSimUPP()
{
  MatchedAttribute=0;
  ForEach(item i in outfit o)
  {
    ForEach(attribute a in item i)
    {
      if(a exists in UPP)
      {
        Find the PAV or CAV, av;
        MatchedAttribute += av;
      }
    }
  }
  Return MatchedAttribute / the number of attributes in UPP;
}

```

圖 3-8 個人喜好檔案與穿著相似度計算演算法

最後一個子模組為協同過濾式模組，此模組主要是要找尋興趣與自己相似的使用者，稱為相似興趣的鄰居(Nearest Neighbor)，參考相似興趣的鄰居的意見，找尋出使用者感興趣的穿著，情境式個人化穿著推薦系統找尋相似興趣鄰居的演算法如圖 3-9 所示：

```

KNearestNeighbor(user a, outfit o)
{
  Neighbors={};
  Find the users who ever rated outfit ;

  ForEach(user u in )
  {
    Add u and sim(a,u) to Neighbors;
  }

  Resort sim(a,u) in Neighbors;
  Return Top K Neighbors;
}

```

圖 3-9 尋相似興趣鄰居的演算法

圖 3-9 中相似度計算式(3-7)如下：

$$sim(u_i, u_j) = \frac{\sum_{o \in O_{u_i \cap u_j}} (r_{u_i, o} - \bar{r}_{u_i}) \times (r_{u_j, o} - \bar{r}_{u_j})}{\sqrt{\sum_{o \in O_{u_i \cap u_j}} (r_{u_i, o} - \bar{r}_{u_i})^2 \sum_{o \in O_{u_i \cap u_j}} (r_{u_j, o} - \bar{r}_{u_j})^2}} \quad (3-7)$$

式子(3-6)中  $r_{u_i, o}$  為使用者  $i$  對穿著  $o$  的評分， $\bar{r}_{u_i}$  為使用者  $i$  的平均分數。之後我們要計算使用者對某個穿著的興趣分數 (Interest Points,  $IP$ )，興趣分數如公式(3-8)所示：

$$IP(u_i, o) = \bar{r}_{u_i} + k \sum_{u \in N} sim(u_i, u) \times (r_{u, o} - \bar{r}_u) \quad (3-8)$$

$$k = \frac{1}{\sum_{u \in N} |sim(u_i, u)|} \quad (3-9)$$

式子(3-8)中  $IP(u_i, o)$  表示為使用者  $i$  對穿著  $o$  的興趣分數，其中  $\bar{r}_{u_i}$  為使用者  $i$  評分的平均分數， $r_{u, o}$  為鄰居  $u$  對穿著  $o$  所打得評分， $k$  計算方式如式子(3-9)所表示為一個正規化參數(normalizing factor)，主要是將鄰居的相似度做正規化的動作。之後我們用以下的演算法對所有的穿著算出使用者的興趣分數，演算法如圖 3-10 所示：

```

CollaborativeRecommendation()
{
  Find the outfits that possible be recommended, po;
  Neighbors = KNearestNeighbors;
  ForEach(outfit o in po)
  {
    Add IP(a,o) to o;
  }
}

```

圖 3-10 協同式過濾計算使用者興趣分數



經由這三個子模組，每一個可能被推薦的穿著都會有 BS、CB 和 CF 這三個分數，經由線性加權權重來計算總分(Total Point,  $TP$ )，式子(3-10)是我們計算總分的方式：

$$TP = \alpha BS + \beta CB + \gamma CF \quad (3-10)$$

在式子(3-10)中  $\alpha + \beta + \gamma = 1$  且  $0 \leq \alpha, \beta, \gamma \leq 1$ 。計算完所有穿著的總分後我們排序分數，推薦分數最高的三個穿著給使用者。

### (七) 個人化穿著配對模組(Personalized Outfit Matching Module)

本模組主要的目的是根據系統所推薦的穿著，找尋使用者衣櫃內的單品做最合適的配對，本模組的示意圖如圖 3-11 所示：

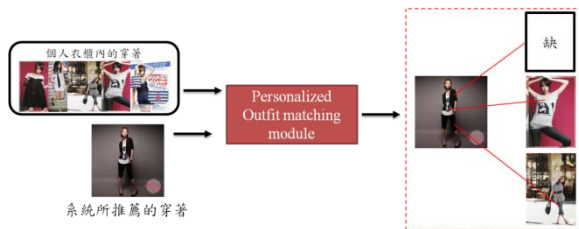


圖 3-11 個人化穿著配對模組

如圖 3-11 所示，根據使用者個人衣櫃內的單品內容與所推薦的穿著單品內容比對，排列組合應該是不止一套，但是我們必須找出一套最相似的配對給使用者參考。首先我們根據系統所推薦的穿著上每一個單品建立所屬的串列，在每一個串列找尋個人衣櫃內相同種類單品加入，然後將所有的串列做排列得出候選的配對穿著。之後將候選的配對穿著中每一個單品與推薦的穿著中的單品，用圖 3-12 中的演算法算出一個相似分數：

```

ItemSimilarity()
{
    point=0;
    items of recommended outfits,  $r_i$ ;
    items of Personal Wardrobe,  $w_i$ ;
    Foreach( item  $i$  in  $r_i$ )
    {
        Foreach(item  $j$  in  $w_i$ )
        {
            if( $i.brand = j.brand$ ) point++;
            if( $i.size = j.size$ ) point++;
            point +=  $RGBSIM(i.color, j.color)$ ;
            if( $i.material = j.material$ ) point++;
        }
    }
}

```

圖 3-12 單品相似度演算法

每套候選配對穿著，會加權出一個相似總分，之後將加權後的配對穿著，依分數排列，一一列舉給使用者參考。

### (八) 聚集代理人(Aggregate Agent)

聚集代理人主要的目的是收集眾人的個人穿著知識庫來更新共同穿著知識庫，如圖 3-13 所示：



圖 3-13 聚集代理人

如圖 3-13 所示，聚集代理人會收集所有人的 PAV 轉換成 CAV 然後儲存在共同穿著知識庫裡。利用這樣的方法來建立共同的穿著知識可以解決冷啟動(Cold Start)的問題、建立出大眾的穿著知識和找出時尚的變化，保持系統的延展性。

## 四、實驗結果

推薦系統準確率的衡量方式，最常用的統計方法為平均絕對誤差 (Mean Absolute Error, MAE)，是比較測試資料的

預測結果與實際結果之間的誤差大小，*MAE* 越小即代表誤差越小。*MAE* 的計算方式如式子(4-1)所表示：

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (4-1)$$

如式子(4-1)所示  $n$  為總資料量， $f_i$  表示為預測值， $y_i$  表示為實際值， $e_i$  代表為兩者之間的誤差，本研究則採用 *MAE* 來衡量推薦方式的準確率。

### (一) 實驗流程

本研究實驗流程共分三個階段，第一階段為募集資料來源，本階段為了累積大量的穿著來源與使用者的回饋資料，本研究找尋願意參與實驗的自願者，提供穿著資訊，在 2009 年 6 月這段時間，本系統共有 25 名註冊使用過本系統，25 名使用者分別 11 名男性，14 名為女性，這些使用者在使用階段提供的 86 套穿著，196 個單品所組合而成，收集穿著資訊結束後，要求使用者任意的回饋資訊，在此共收集了 75 筆回饋。

在第二階段系統正式的開始啟動推薦功能。收集了使用者實際給的回饋值與系統所預測值，用來推測系統推薦結果準確度，在此階段本實驗共收集了 163 筆回饋，比較推薦結果的準確率之外，本研究的實驗分析輔以使用者回饋問卷，問卷設計的內容為根據 TAM (Technology Acceptance Model) [5] 所設計。

### (二) 實驗分析

根據推薦系統準確率的評估方式，本研究的推薦方式主要是根據個人化推薦模組的三個子模組分別為身材相似度模組、內容導向式模組和協同過濾式模組，想了解怎樣的權重比例比較適合本系統，本研究從第二階段使用者開始使用本系統推薦功能，共收集了 163 次推薦紀錄，並由這

紀錄來計算出不同參數的預測值，並從這些參數的比重了解系統制定參數的比重，測試數據如圖 4-1 所示：

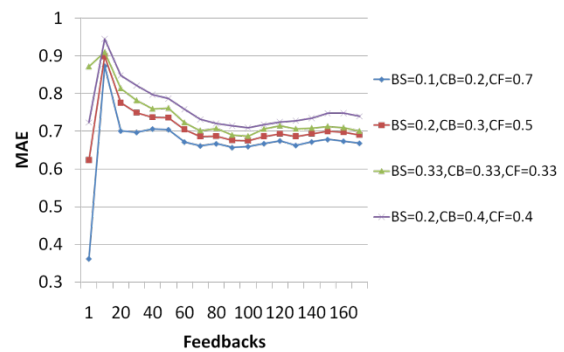


圖 4-1 參數比重比較圖

從圖 4-1 所示，BS 為身材相似模組，CB 為內容導向式模組，CF 為協同過濾是模組，當權重比例為 BS=0.1, CB=0.2, CF=0.7 時 *MAE* 測試為最小，由此可知當 CF 權重比例比較大時，其精確度會比較好，這就好比現實生活中，詢問別人意見和參考時尚雜誌一樣。但以權重比例為 BS=0.33, CB=0.33, CF=0.33 與權重比例為 BS=0.2, CB=0.4, CF=0.4 來比較時，雖然後者的 CF 權重比較高，但其精確度較低，主要的原因為內容導向式的權重不宜調整的太大，在本研究的觀察，穿著搭配這方面，使用者是容易改變其個人偏好的。另外在身材相似度模組也不適合比內容導向式模組權重大，主要因為一般使用者在找尋穿著時，會先考慮到自己的偏好找尋合適的穿著，在根據自己的身材來挑選合不合適，但追究另一個原因，使用本系統的使用者皆屬於在標準身材內，在找尋穿著時並不會考慮其身材是否合適。

本研究想了解有無個人化推薦的準確度比較，*MAE* 測試如圖 4-2 所示：

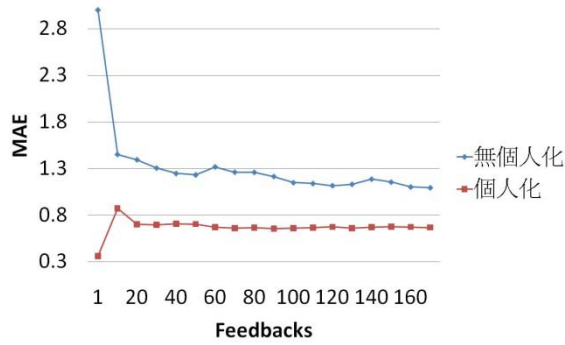


圖 4-2 有無個人化推薦比較

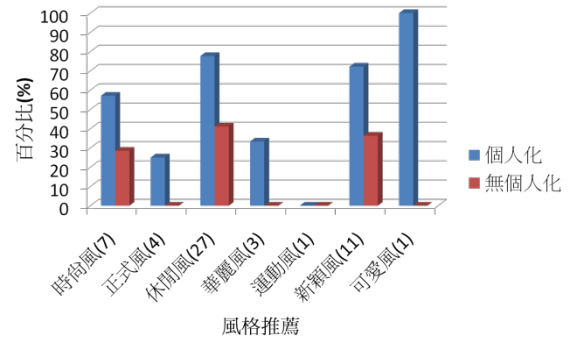


圖 4-4 風格推薦百分比

由圖 4-2 所示，本研究無個人化推薦則採用平均分數推薦方式，個人化推薦則採用本系統個人化推薦模組設計的方式做推薦，由圖 4-2 所知無個人化推薦 MAE 收斂在約 1.2 左右，而有個人化推薦則 MAE 收斂約在 0.7 左右，其精確度明顯的優於無個人化推薦方式。

如圖 4-3 和圖 4-4 所示，由使用者針對特殊風格與場合推薦下，贊同與否的紀錄，根據此紀錄，本研究分析有無個人化推薦的結果，在無個人化部份，是使用平均情境推薦，即推薦一個平均值大的情境給使用者，並無透過系統收集個人穿著知識來推薦給使用者。

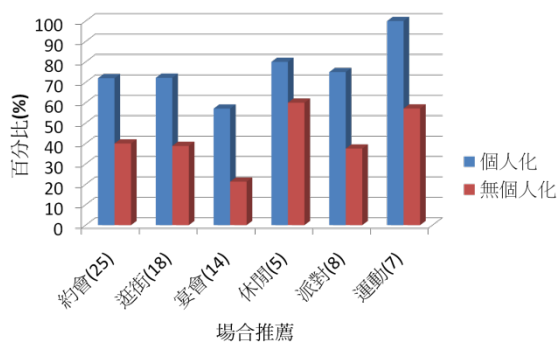


圖 4-3 場合推薦百分比

由圖 4-3 與圖 4-4 所示，有個人化的贊成百分比都大於無個人，另外可以分析，像圖 4-11 中的休閒風在 27 次推薦中被贊同大約 70%，由此可知當被推薦的次數越多，所推薦的情境越能符合使用者所認同的情境，相對的在華麗風的 4 次推薦中只被贊同了一次，明顯的系統是透過收集穿著知識來針對特定情境下的穿著給使用者。

## 五、結論

這是一個資訊爆炸的時代，任何與生活上息息相關的資訊，皆容易在網路上取得，因此在如何找到合適的資訊，皆是未來所有資訊系統的重點，本研究所提出的情境式個人化穿著推薦系統主要目的為針對特定的情境下幫助使用者在龐大的穿著資訊中找尋合適自己的穿著，同時也提供了一個良好的資訊平台讓使用者提供穿著與參與穿著搭配上的討論，在此情況下建立使用者的穿著知識和建立情境之間的關連性，在實驗結果顯示本研究的構思是合理且可行，在參與使用者的問卷顯示，本研究的確實讓使用者認知有用且易用，並且達到使用者意圖使用本系統。

## 六、致謝

本研究承蒙國科會計畫

NSC95-2221-E-006-158-MY3 經費部分補助，特此感謝。

## 七、參考文獻

- [1] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl, "MovieLens Unplugged: Experiences with an Occasionally Contented Recommender System." *Proc. Int'l Conf. Intelligent User Interfaces*, 2003.
- [2] C.C. Peddy and D. Armentrout, "Building Solutions with Microsoft Commerce Server 2002." *Microsoft Press*, 2003.
- [3] D. Billsus, C.A. Brunk, C. Evans, B. Gladish, and M. Pazzani, "Adaptive Interfaces for Ubiquitous Web Access." *Comm, ACM*, vol.45, no. 5, pp.34-38,2002.
- [4] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," *Comm. ACM*, vol. 35, no. 12, pp. 67-70,1991.
- [5] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology." *MIS Quarterly*, 13(3), 319-340, 1989.
- [6] G. Linden, B. Smith, and J. York, "Amazon.com Recommendation: Item-to-Item Collaborative filtering." *IEEE Internet Computing*, Jan./Feb. 2003.
- [7] G. Salton, "Automatic Text Processing." *Addison-Wesley*, 1989.
- [8] J.J. Rocchio, "Relevance Feedback in Information Retrieval," *SMART Retrieval System- Experiments in Automatic Document Processing*, G. Salton, ed., chapter 14, Prentice Hall, 1971.
- [9] J.L. Herlocker, J.A. Konstan and L.G. Terveen, "Evaluating collaborative filtering recommender system," *ACM Transactions on Information Systems*, vol.22(1), pp. 5-53,2004.
- [10] M. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning*, vol. 27, pp. 313-331, 1997.
- [11] M. Pazzani, "A Framework for Collaborative, Content-Based, and Demographic Filtering," *Artificial Intelligence Rev.*, pp. 393-408, Dec. 1999.
- [12] P. Resnick and H.R. Varian, "Recommender Systems," *Communications of the ACM*, vol.40(3), pp. 56-58, 1997.
- [13] P. Sweetser and P. Wyeth, "GameFlow: a model for evaluating player enjoyment in games," *ACM Computers in Entertainment(CIE)*, vol.3(3),2005.
- [14] R. Burke, "Hybrid Recommender systems: survey and experiment," *User Model. User Adapt. Inter.*, vol.12, pp. 331-370,2002.
- [15] 歐秀明，賴來洋，"實用色彩學"，雄獅圖書公司，八十八年。
- [16] Youngjin.com 著，"給設計師的專業配色典 - The Color for Designer."，博碩文化，2009。
- [17] [http://zh.wikipedia.org/wiki/Web\\_2.0](http://zh.wikipedia.org/wiki/Web_2.0)
- [18] <http://www.iricolor.com/main.asp>