

# 應用隱性回饋與柏拉圖分佈於數位學習平台上的 學習元件推薦

鍾哲民 王宗一 邱迪凱 蔡昆樺

國立成功大學工程科學系

tsaikunhua@msn.com

## 摘要

數位學習已漸漸成一種學習的新潮流，而面對數位環境中大量的學習元件，如何讓使用者搜尋並使用合適的元件變成一個重要的議題。本文提出二階層推薦架構並結合辭書來滿足使用者的需求，首先在上層的知識概念推薦，它藉由辭書技術將使用者的查詢語句轉換為數個辭書概念，接著在第二部分實體元件推薦中，它根據使用者的興趣來推薦學習元件，其中本文採用 LOM 的欄位來描述學習元件特徵並利用柏拉圖分佈來解決興趣分析的誤差，其外為了便利使用者使用本系統，隱性回饋的機制被用來蒐集使用者的回饋進而分析使用者的興趣，透過本系統，可以提供使用者完全個人化的學習元件推薦。

**關鍵詞：**SCORM，學習元件，隱性回饋，柏拉圖分佈

## 一、前言

隨著電腦與網路技術的快速發展，傳統的學習方式與環境已有了重大的改變。在學習方式上，學生的學習方式從書本的學習慢慢轉移至電腦上的數位學習，其中學習教材從枯燥的文字變成了可互動的畫面，影像與聲音。相同地，學習環境也從教室走向網路上的學習平台。學生可以透過此學習平台與

各地的同學與老師一起上課，或者在任何時間選擇自己欲學習的知識來學習，這樣的學習環境讓學生有更高的彈性可以規畫自己的學習課程與時間。

目前越來越多的大學或組織都開發了私人的網路學習平台供學生來使用，但是這些平台都存在了一個共通的缺點-異質性，每個平台都使用自己的方式去描述與定義他們各自的教材，因此這些教材都很難被分享於不同的平台間，有鑑於此，一些研究組織共同發展了一個用來描述與交換分享教材的協定- SCORM[1,2,3]。此協定中，它使用了固定的辭彙來描述教材的屬性分類與屬性值，只要基於此協定所發展出來的平台便能使用與分享符合 SCORM 的教材。

隨著數位教材的增加，當使用者面對大量可選擇的教材時，學習平台要如何有效的提供合適的教材給使用者，這將會是一個重要的課題。試想一個平台不能提供符合學生興趣的教材，那麼學生還有意願再去使用此平台學習嗎？通常學習的意願來自學生的興趣，如果能夠掌握住學生的興趣，那麼針對學生興趣所提供教材應該是能吸引學生的目光，進而使學生有意願來使用此平台學習。

在本文先前的研究中[8]，一套基於顯性回饋為基礎的教材推薦機制已經被發展出來，在此機制中，學生學習完一個教材後，必須給予此教材一個分數來表達他對於此教材的喜好程度，雖然此方式能夠較真實

地反應學生的興趣，但是由於它是基於學生回饋的分數，如果學生很隨意地回饋分數，那麼最後的推薦結果將是十分糟糕，此外，要求學生回饋也可能造成他們的不便與厭煩，久而久之，學生不是隨便給分數就是沒意願再去使用此平台。因此本論文提出一套基於隱性回饋的推薦方法，此方法毋須學生回饋分數，它經由蒐集學生點閱的教材，並分析這些教材的屬性，進而推論出學生的興趣為何，在分析學生興趣過程中，此方法導入柏拉圖分佈去降低興趣分析的誤差，同時也引入熱門推薦的概念，推薦最符合學生興趣的學習元件。

## 二、相關研究

### (一) 共享物件參考模型(SCORM)

共享物件參考模型的目的是用來描述與定義一個教材的協定。它是由幾個主要的組織 IEEE LTSC、AICC、ARIADNE 與 IMS 所發展的協定修改而來。在此協定中，它使用學習元件元資料 [5](Learning Object Metadata, LOM) 來描述教材屬性特徵。它採取九大分類來描述一個教材，*General*、*Lifecycle*、*Meta-metadata*、*Technical*、*Educational*、*Rights*、*Relation*、*Annotation* 與 *Classification*。本文擷取這些欄位中的部分欄位來表示學習元件的特徵，作為評估使用者興趣的依據。

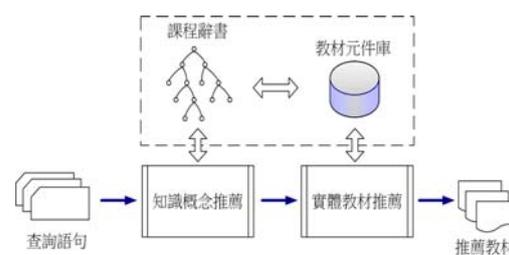
### (二) 推薦模型

目前推薦模型主要可以分為三種方式，基於內容推薦模型 (Content-based recommendation)[12]、協同式推薦模型 (Collaborative recommendation)[7,13,14,15] 與混合式推薦 (Hybrid recommendation)[6,11]。基於內容方式的推薦是分析使用者過去使用的物件來推薦，例

如，假使使用者過去經常觀看體育節目，那麼採取內容推薦模型，系統將推薦體育節目給使用者觀看而不會推薦綜藝節目或是新聞節目。採用此方式的問題在於可能會導致推薦陷於局部化推薦，此意謂系統將只推薦體育節目其他節目將都不會推薦。而協同式推薦模型是能解決內容推薦模型的局部化推薦問題，它參考其它使用者的回饋來預測並推薦物件，例如假使使用者 A 給予節目 w 高評價，而另有兩使用者 B 與 C 同時給予節目 w 與節目 b 高評價，根據協同式推薦，它也許會推薦節目 b 給使用者 A。雖然協同式的推薦能解決內容推薦的問題，但是它也存在一些問題，假如使用者使用的資料不夠多時，也會導致預測錯誤。因此目前多數的推薦系統多採取混合式推薦模型，它綜合上述兩種方法的優點來推薦物件給使用者。

### (三) 先前研究與成果

在本文先前研究[9,10]提出的推薦模型主要將推薦模型分為兩個層級，其包括知識概念推薦與實體教材推薦。



圖一、兩層式教學元件推薦

在圖一中，首先使用者能輸入一查詢語句來表示其欲學習的知識，接著在知識概念的推薦層級中，它透過特定的知識辭書去分析使用者真正欲學習的知識為何，在知識概念推薦層級，它將使用者的意圖轉換成數個知識概念，接著針對每一個特定知識概念，實體教材推薦層級根據使用者的興趣來推薦合

適的學習元件給予使用者學習。在本篇論文中，它試圖發展一套新的實體教材的推薦方法，因為先前的研究採取顯性回饋方式來分析使用者的興趣近來推薦學習元件，但是要求使用者的回饋除了造成使用者不便外，推薦的精確度也會因為使用者的回饋有所影響(例如，使用者隨意的回饋分數或是沒有回饋，通常會導致不準確的推薦)。有鑑於這些原因，本文提出一套基於隱性的回饋方法來解決上述的問題。

### 三、興趣推薦模型

本文提出的推薦模型是基於學習者興趣的推薦模型。此模型能分析學習者所使用過的學習元件，進而了解使用者對於那一些元件特徵感興趣。

#### (一) 使用者興趣分析

針對 SCORM 的標準，每一個學習元件都能透過 LOM 所定義的欄位來描述它。這些欄位可以被視為學習元件的特徵，因此如果欲了解使用者的興趣，可以藉由分析使用者所使用過的學習元件特徵值來得知。假使某一個特徵值在使用者的學習過程中經常出現，那麼它也許代表了它是使用者所感興趣的特徵值之一，因此本文定義一特徵值分數公式來表示使用者對於一個特徵值的偏好程度。

$$PS(fv) = \log \frac{P(fv|UIP)}{P(fv)} \quad (1)$$

在公式(1)中， $PS(fv)$ 表示使用者對於特徵值  $fv$  的偏好分數，其中  $P(fv)$ 表示特徵值  $fv$  出現在候選推薦中的機率，而  $P(fv|UIP)$ 表示特徵值  $fv$  出現在使用者過去使用紀錄( $UIP$ )的機率，假如  $PS(fv)$ 越高表示使用者對此特徵值

越感興趣。由於一個學習元件能視為許多特徵值所聚集的集合，因此使用者對於一個學習元件的興趣程度可以表示成公式(2)：

$$IS(o) = \frac{1}{|F|} \sum PS(fv_i) = \frac{\sum w_i \cdot PS(fv_i)}{\sum w_i} \quad (2)$$

，其中  $IS(o)$ 表示使用者對於一學習元件  $o$  的興趣分數，根據公式(2)，興趣分數是由一群特徵值偏好分數的平均， $F$ 表示此學習元件的特徵值集合，此平均公式也可表示成一具有權重的公式，不同的特徵給予不同的權重，其目的是因為並非所有的特徵對於使用者都具有相同程度的重要。

#### (二) 預測誤差消除機制

經由公式(1)與(2)，可以了解使用者偏好那些特徵值並且計算出使用者對學習元件的興趣分數，但是在實際的使用上，系統所觀察到的資訊可能存在部分的誤差。有兩個主要的原因會造成這些誤差的存在，它包含了使用知識概念推薦所產生的誤差與觀察資訊不足所產生的誤差，針對這兩種誤差，分別於下段中描述，並說明如何消除它們。

**基於知識概念推薦之誤差。**在第二節中，它說明先前所設計的推薦模型，在知識概念推薦的層級中，它首先決定那些概念是使用者欲學習的知識，然後在針對每個概念下的學習元件進行推薦。明顯地，對於一特定的特徵值其出現在每一個概念下的機率是不相等的，所以再評估一個特徵值偏好分數前，它必須考慮不同概念下的機率分佈，因此偏好分數公式(1)需被修改成為下列公式(3)：

$$\begin{aligned}
PS(fv) &= \log \frac{P(fv|UIP)}{P(fv)} \\
&\approx \log \left( |C^{UIP}|^{-1} \cdot \sum_{c_i \in C} \frac{P(fv|UIP(c_i))}{P(fv|c_i)} \right)
\end{aligned}
\tag{3}$$

原本單純只計算單一特徵值在一個概念下的偏好分數被修改為考慮所有使用者已使用過概念下的偏好分數，其中  $C$  表示使用者已學習過的知識概念集合，而最後的特徵值偏好分數被定義為特徵值分別在不同概念  $c_i$  的偏好分數平均，透過此方式，它使得特徵值得偏好評估能考慮到所有的概念。

**基於觀察資訊不足之誤差。**對於大部分的推薦系統而言，假使能蒐集到使用者夠多的使用資訊，通常都能提供精確的推薦結果，但是如果在系統運作初期，或是使用者的觀察資訊還未蒐集足夠的時期，推薦的效果通常較不精確，很可能發生觀察到的資訊是較偏頗或較傾斜的情況，因而導致評估的分數與實際上真實的分數有極大的差距。從公式(1)中可以清楚發現，當特徵值出現於概念學習元件的機率固定下，影響特徵值偏好分數的因子將是分子的條件機率(意即特徵值出現於歷史紀錄次數/歷史紀錄次數)。所以觀測的資訊量(意即歷史紀錄次數多寡)將影響最後的偏好分數評估。因此本文參考[4]，提出柏拉圖分佈的概念來消除資訊稀疏所導致的評估誤差。目前在許多的研究中都採用柏拉圖分佈來描述某一事件在特定頻率下所出現的隨機機率，雖然並非所有的事件出現的特定頻率機率都能完全符合柏拉圖的分布函數，但是此一分佈已被許多實驗證明可以概略地描述事件出現頻率與其機率的關係，換句話說，可以利用此分布來表達隨機實驗中事件在特定的頻率所出現的機率。因此公式 (3)的特徵值偏好分數可以被表示成

在真實的偏好分數與與隨機機率所影響的分數總合，在下列公式(4)中，它表示真正的偏好分數與評估分數之間的關係：

$$PS^{real}(fv) = (1 - \lambda) \cdot PS^{observe}(fv) \tag{4}$$

系統觀察資訊評估出的偏好分數  $PS^{observe}(fv)$  包含真實的偏好資訊  $PS^{real}(fv)$  加上隨機機率分數，而根據柏拉圖分佈的概念，隨機機率分數可視為觀察資訊的部分比例結果，因此真實的偏好分數能表示為上述的式子。

根據公式(3)，它產生的偏好分數可能為正 ( $P(fv|UIP) > P(fv)$ ) 或是負 ( $P(fv|UIP) < P(fv)$ )，當偏好分數為正時，它能被看作是正偏好(因為特徵值在歷史紀錄出現次數的機率大於其正常機率);反之則可以視為負偏好(特徵值在歷史紀錄出現次數機率小於其正常機率)，從正偏好的角度來說，當觀測資訊太少時，特徵值的偏好分數可能被高估或是接近真實的偏好分數，例如：

$$E := \{e_1, e_2, e_3, e_4, e_5\},$$

$$E(fv) := \{e_1, e_2, e_3\},$$

$$UIP_a := \{e_1, e_2, e_5\},$$

$$UIP_b := \{e_1, e_2, e_3, e_4, e_5\},$$

對於兩組紀錄  $a$  與  $b$  而言，其  $a$  的  $P(fv|UIP)$  為  $\log(10/9)$ ，而  $b$  的  $P(fv|UIP)$  為  $\log(15/15)$ ，明顯地，雖然  $b$  使用所有具有  $fv$  特徵值的學習元件，但是偏好分數卻相對較低，因此在這組例子中， $a$  的評估誤差也許大於  $b$  的評估誤差，根據這樣的情況，偏好評估所產生的誤差機率能表式成：

$$\lambda = (N(E(fv) \cap E(fv|UIP)))^{-\alpha}, \text{ if } PS(fv) > 0 \tag{5}$$

其中， $E(fv)$  表示擁有特徵值  $fv$  的元素集合， $E(fv|UIP)$  則表示在使用者歷史紀錄中擁有特徵值  $fv$  的紀錄集合，而誤差機率被表示成此

二集合交集數目  $N(\bullet)$  的倒數。所以當交集的數目越大則表示觀測到的誤差機率越小；反之交集數目越小，則誤差機率越大。接著考慮負偏好的情況 ( $P(fv|UIP) < P(fv)$ )，當觀測的資訊太少，特徵值的偏好分數可能是接近真實偏好分數或是被低估，考慮以下列兩個例子：

$$E := \{e_1, e_2, e_3, e_4, e_5\},$$

**Case1:**

$$E(fv) := \{e_1\},$$

$$UIP_a := \{e_1, e_2, e_3, e_2, e_4, e_5, e_3, e_4, e_5, e_3\}$$

**Case 2:**

$$E(fv) := \{e_1, e_2, e_3\}$$

$$UIP_b := \{e_1, e_4, e_5, e_4, e_5, e_4, e_5, e_4, e_5, e_5\},$$

在第一組的資訊中，偏好分數為  $\log(1/2)$ ，第二組的偏好分數則為  $\log(1/6)$ ，明顯地雖然兩組都僅使用到一個擁有特徵值的學習元件，但是第一組的偏好分數卻大於第二組，因此第一組中的偏好評估誤差大於第二組，因此可以使用特徵值出現在學習元件中的機率來表示可能出現誤差的機率，在特徵值出現在歷史紀錄的機率  $P(fv|UIP)$  相同時，當  $P(fv)$  越小則評估誤差機率越高；反之  $P(fv)$  越大則誤差機率越小，因此當偏好分數小於 0 時，其機率可定義為：

$$\lambda = (1 + P(fv))^{-b}, \text{ if } PS(fv) < 0 \quad (6)$$

其中，由於  $P(fv)$  是介於 0~1 之間的值，如果直接取倒數表式隨機機率的話，將造成機率值大於 1，因此改成取  $(1+P(fv))$  的倒數。由於觀察資訊是由許多知識概念下的觀測資訊所共同組成，因此最後特徵值的偏好公式整合如以下公式(7)與(8)：

$$PS^{real}(fv) = (1 - \lambda) \bullet \log \left( |C^{UIP}|^{-1} \bullet \sum_{c_i \in C} \frac{P(fv|UIP(c_i))}{P(fv|c_i)} \right)$$

(7)

$$\lambda = \begin{cases} \left( N(\bigcup_{c_i \in C^{UIP}} E(fv|c_i) \cap \bigcup_{c_i \in C^{UIP}} E(fv|UIP(c_i))) \right)^{-x} \\ \quad , \text{ if } PS(fv) > 0 \\ \left( 1 + \frac{\sum_{c_i \in C^{UIP}} E(fv|UIP(c_i))}{\sum_{c_i \in C^{UIP}} E(UIP(c_i))} \right)^{-y} \\ \quad , \text{ otherwise} \end{cases} \quad (8)$$

### (三) 熱門程度評估

雖然根據興趣分數的推薦，可以讓較符合使用者興趣的學習元件被優先推薦給使用者，但是通常在使用者使用此系統一段時間後，當使用者興趣樣式(UIP)逐漸形成時，局部性推薦的問題將會慢慢浮現。原因是當興趣樣式形成後，系統在推薦學習元件時，符合興趣樣式的學習元件總是被優先推薦，對於那些使用者未曾使用或是最近加入的學習元件，明顯地，它們總是被排序於推薦末端，但是這些學習元件對於使用者而言或許是感興趣的，因此本文加入熱門程度的評估概念，來計算學習元件在最近的熱門程度，如此，目前較熱門的學習元件也會有機會被優先推薦給使用者使用，此熱門程度評估的公式描述如下公式(9)：

$$PD(o) = \sum_{i \in W} pd_i(o) \sum pd_i(o),$$

$$\text{where } pd_i(o) = \begin{cases} \frac{t_i}{T}, & \text{if } t_i < T \\ 1, & \text{otherwise} \end{cases} \quad (9)$$

在公式(9)，W 代表在最近一段時間內使用過學習元件 o 的使用者集合，集合的大小能反映出此學習元件最近熱門的程度，同時，公式也考慮使用者使用時間的因素，因為採取隱性的回饋機制僅蒐集學習元件被點閱的次數，但僅點閱不代表使用者真正使用了此學習元件，所以公式透過評估使用者使用的時間來決定熱門程度，由於評估一個學習元

件要學習多久時間也是個難以決定的因素，因此時間門檻值的設定必須考量不同的學習內容與狀況，公式中  $t_i$  表示在  $W$  中一位使用者使用此學習元件的時間，而  $T$  表示將所有  $W$  中的使用者時間由小到大排序後的中位數，只要使用者時間大於  $T$ ，則此使用者被視為真正使用了此學習元件，假如小於  $T$  則認定為使用者只有  $\frac{t_i}{T}$  的機率真正使用了此學習元件，因此熱門程度最後能表示成最近一段時間內使用者使用此學習元件的機率總和。

#### (四) 學習元件推薦

每個候選的學習元件經過系統的評估後，會產生兩個評估分數，使用者對於此學習元件的興趣分數  $IS$  與此學習元件最近的熱門程度分數  $PD$ 。而學習元件的推薦分數可表示成：

$$RS(o) = IS(o) \cdot PD(o) \quad (10)$$

其中  $RS$  表示學習元件的推薦分數，當每個學習元件的推薦分數計算出來後，系統將根據學習元件的推薦分數來排序，分數越高的學習元件將會被排序在較高的順位被推薦。

### 四、系統設計

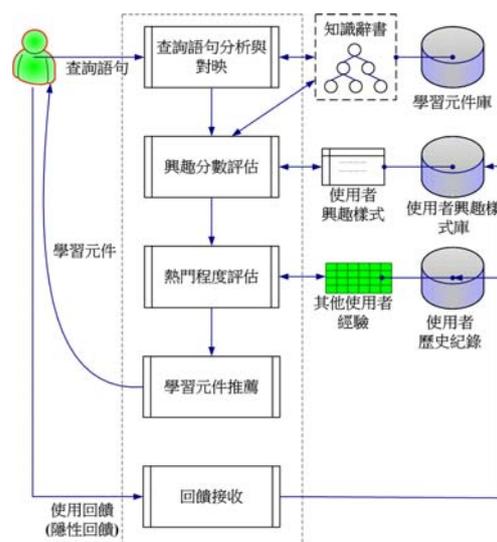
在本節，首先介紹本系統的推薦的基本架構，並說明每一個模組所負責的功能為何，並針對主要推薦評估的模組，採用虛擬碼描述如何實作。

#### (一) 推薦基本架構

圖二描述本系統的基本推薦架構，當使用者將欲學習的課程透過查詢語句的方式

輸入於系統後，在查詢語句分析與對映的模組，它首先將此查詢語句分解並剖析出有意義的關鍵字，接著透過對映知識辭書中概念，決定那一些知識概念是使用者想要學習的，此過程也稱作知識概念的推薦，針對此推薦的詳細過程可參考先前的研究[10]，語句所對映的知識辭書，在本系統中是被定位成具有索引與知識描述的結構，辭書中的每一個概念能視為一個子知識的描述，而每個概念都會指向學習元件庫中每個描述此概念的學習元件。

當系統確定使用者欲學習的知識概念後，接著會分別針對每一個知識概念下的學習元件來進行使用者興趣評估與熱門程度評估，最後整合此二分數來達成學習元件的推薦。而在最下面的回饋接收模組是用來記錄使用者的使用紀錄，當使用者點選一個學習元件時，回饋接收模組將紀錄此學習元件的特徵於使用者興趣樣式庫，而此使用紀錄與使用時間也會記錄在使用者歷史紀錄中。在接下來的三小節中，分別描述如何實作興趣分數評估模組、熱門程度評估模組與回饋接收模組。



圖二、學習元件推薦系統基本架構

## (二) 興趣分數分析模組設計

在本系統中，興趣分數分析模組目的是用來分析使用者對於某一概念下的學習元件其分別的喜好程度，簡單的說，當學習元件興趣分數越高，其意謂著使用者應該越喜好此元件，因此透過推薦高興趣分數的學習元件，可以預期使用者將會樂意於學習。在下圖三，它描述了評估興趣分數所使用的虛擬碼。此興趣分數評估演算法主要包含兩個主要部分，第一個函數 `getISofMaterials` 用來取得使用者的興趣樣式與所要評估的學習元件資訊，其中 `UIP` 集合儲存使用者興趣樣式，其元素紀錄每個特徵值 `fv` 的偏好分數 `PS`，而 `LO` 存放指定概念下的學習元件資訊，包括每一個的名稱 `loname` 與特徵值集合 `FV`，最後對於指定的概念，函數會回傳集合 `M`，其包含學習元件名稱，興趣分數與熱門程度(熱門程度的值將在熱門評估演算法中描述)。在第二個函數 `estimateIS` 中，它用來評估一學習元件的興趣分數，它首先比較每一個特徵的特徵值是否出現在使用者興趣樣式中，如果有的話，則累計此特徵值分數，最後平均的特徵值分數將當作此學習元件的興趣分數。根據公式(2)的描述，興趣分數的平均採取權重方式計算，事實上在學習元件的 `LOM` 中，並非所有特徵都是使用者所關心的，因此本文僅挑選使用者較關心的特徵作為評估的興趣分數的依據。

---

**INPUT:** User's ID, *uid* and Concept name, *cname*.

**OUTPUT:** A set of learning objects, *M*.

procedure `getISofMaterials(uid, cname)`

Begin

    new a *UIP*; // A set of (fv,PS)s

    new a *LO*; // A set of (loname,FV)s

    new a *M*; // A set of (loname,IS,PD)s

*UIP*:=get the user *uid*'s interest pattern from the  
    User Interest Patterns Database;

---

*LO*:=get the data of all learning objects in the  
    concept, *cname* from the Materials

    Repository;

    int index:=0;

    float score;

    while (*LO*!=NULL)

    Begin

        new a *m*; // (loname,FV)

*m*:= *LO*[index];

        score:=estimateIS(*UIP*, *m*);

*M*[index].loname=*LO*[index].loname;

*M*[index].IS:=score;

        index:=index+1;

    End

    return *M*;

End

//-----

**INPUT:** A user's interest pattern, *UIP* and a  
learning object, *m*.

**OUTPUT:** The interest score of *m*.

procedure `estimateIS(UIP, m)`

Begin

    float iscore:=0;

    int index:=0;

    for (compare each feature value *fv* in *m.FV* with  
        *UIP*)

    Begin

*fv*:= *m.FV*[index];

        if (*fv* exists in *UIP*)

            iscore:=iscore+(*PS*(*fv*) in *UIP*);

        iscore:=iscore/*FV*;

        return iscore;

    End

End

---

圖三、興趣分數評估演算法

### (三) 熱門程度評估模組設計

熱門程度的評估模組，能反映目前此學習元件的熱門程度，並強化了原本只有興趣分數評估的機制，讓使用者不但可以使用到自己感興趣的學習元件同時也可以使用到最近較熱門的學習元件。在圖四中，它描述了熱門程度評估演算法的內容，在第一個函數 `getPDofMaterials`，它接收從興趣分數評估模組所傳出的學習物件集合  $M$ ，接著它取出每一個學習元件，依序計算每一個的熱門程度，計算熱門程度的主要函數描述於第二個函數 `estimatePD`，首先它取出物件的名稱，並搜尋目前那一些使用者已經使用過此元件，並且將使用者使用此元件的時間放置於集合  $U$ ，接著根據使用時間的長短，排序  $U$  中的每一個使用時間由小到大，並存於  $SortedU$ ，接著計算位於中位數的使用時間當作評估標準，位於中位數後的使用時間設為 1，而小於或等於的設為(本身使用時間/中位數的時間)，最後累加這些值成為熱門程度的分數。

---

**INPUT:** Concept Name,  $cname$  and a set of learning objects  $M$ .

**OUTPUT:** A set of learning objects,  $M$ .

procedure `getPDofMaterials(cname, M)`

Begin

    int  $index:=0$ ;

    for (each learning object in  $M$ )

        Begin

$M[index].PD:=estimatePD($

$M[index].loname, cname);$

$index:=index+1$ ;

        End

    return  $M$ ;

End

//-----

---

**INPUT:** Learning object name,  $loname$  and

---

a concept name,  $cname$ .

**OUTPUT:** Popular degree of the learning

object,  $loname$ .

procedure `estimatePD(loname, cname)`

Begin

    int  $index=0$ ;

    new a  $U$ ; // a set of study times of the users

    for (check each user  $u$ )

        Begin

            if (the object  $loname$  exists in  $u$ 's  $UIP(cname)$ )

                Begin

$U[index]:=$ the study time of learning

                        object  $loname$  in  $u$ 's  $UIP(cname)$ ;

$index:=index+1$ ;

                End

    End

    new a  $SortedU$ ;

$SortedU:=$ Sorting the elements in  $U$  from low  
        to high by the study time;

    int  $middle=|SortedU|/2$ ;

    for (each element in  $SortedU$ )

        Begin

            if ( $index<middle$ )

$SortedU[index]:=$

$SortedU[index]/SortedU[middle]$ ;

            else

$SortedU[index]:=1$ ;

        End

    float  $pdegree:=$ the sum of all elements in  $SortedU$ ;

    return  $pdegree$ ;

end;

---

圖四、熱門程度評估演算法

### (四) 回饋接收模組設計

隨著使用者使用的過程，系統會開始調整對使用者所評估的興趣分數，本文採取的方式為批次的調整，也就是以使用者每一次登入至登出的期間所使用的學習元件作為

一個批次，進行評估的調整，因此每一批次結束後，系統將調整所有有發生變動特徵值的特徵值分數，圖五描述了如何重新評估特徵值偏好分數，首先當使用者完成一連串的學習元件使用後，使用資訊  $R$  會回傳給回饋接收模組， $R$  為使用資訊的集合其包含三個元素(loname,stime,cname)，第一個為學習元件名稱，第二個為使用此元件的時間，第三個表示此元件所屬的概念，所有的學習元件資訊會先記錄於使用者歷史紀錄與使用者興趣樣式庫，接著針對有變動的特徵值(變動意謂此特徵值使用紀錄有增加)重新計算其特徵值分數。

**INPUT:** User's ID  $uid$ , a set of learning objects  $R$ .

procedure updatePS( $uid,R$ )

Begin

int index:=0;

for (calculate each learning object in  $R$ )

Begin

new a  $M$ ; //(loname,cname, $F$ ),  $F$  is the set of feature values

$M[index].loname:=R[index].loname$ ;

$M[index].cname:=R[index].cname$ ;

$M[index].F:=$ take out the feature values of the learning object

$M[index].loname$  from Material

Repository;

if ( $M[index].loname$  doesn't exists in

$UIP^{uid}(M[index].cname)$ )

Begin

for (each element  $fv$  in  $M[index].F$ )

Begin

if ( $fv$  exists in  $UIP^{uid}(M[index].cname)$ )

$N(fv):=N(fv)+1$ ;

/\*  $N(fv)$  represents the number of

learning

objects which have the feature value

$fv$  in  $UIP^{uid}(M[index].cname)*/$

else

$N(fv):=1$ ;

Calculate the  $P(fv|UIP^{uid}(cname))$ ;

Calculate the  $\lambda_{cname}(fv)$  by the formula (8);

Calculate the  $PS^{real}(fv)$  by the formula (7);

End

write each  $PS^{real}(fv)$  to the  $UIP^{uid}$ ;

End

End

圖五、特徵值分數評估演算法

## 五、結論與未來工作

在本文中，它描述了如何使用隱性回饋的方法來完成個人化的學習元件推薦，並且導入 LOM 的概念來描述學習元件特徵，並利用柏拉圖分佈函數來降低分析興趣分數的誤差，同時也採用熱門程度的概念來加強推薦的效果，在未來的工作，本文先將採用特定領域下的數位學習，來評估此推薦系統再推薦學習元件的準確性，針對結果來調整隨機機率的參數，以真正的滿足適性化學習元件推薦。

## 六、誌謝

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] Advanced Distributed Learning (ADL) initiative. *SCORM Specifications – The SCORM Content Aggregation Model Version 1.2.*

- [2] Advanced Distributed Learning (ADL) initiative. *SCORM Specifications – The SCORM Run-Time Environment Version 1.2*.
- [3] Advanced Distributed Learning (ADL) initiative. *SCORM Specifications – SCORM Version 1.3 Application Profile Working Draft Version 1.0*.
- [4] A. Popescul, L. H. Ungar, D. M. Pennock and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments", Proc. 17<sup>th</sup> Conf. Uncertainty in Artificial Intelligence, 2001.
- [5] Final LOM Draft Standard, <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html>.
- [6] I. Soboroff and C. Nicholas, "Combining Content and Collaboration in Text Filtering", Proc. Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering, Aug. 1999.
- [7] J. A. Konstan, B. N. Miller, D. J. Maltz, L. Herlocker, L. R. Gordon and J. Riedl, "GroupLens: Applying Collaborative Filtering to Usenet News. Comm", ACM, vol. 40, no. 3, 77-87, 1997.
- [8] K. H. Tsai, M. C. Lee and T. I. Wang, "A Learning Objects Recommendation Model based on the Preference and Ontological Approaches", The 6th IEEE International Conference on Advanced Learning Technologies, Kerkrade, The Netherlands.
- [9] M. C. Lee, D. Y. Ye and T. I. Wang, "Java Learning Object Ontology", The 5th IEEE International Conference on Advanced Learning Technologies, 538-542, Kaohsiung, Taiwan, July 5-8, 2005.
- [10] M. C. Lee, K. H. Tsai, D. Y. Ye and T. I. Wang, "A Service-Based Framework for Personalized Learning Objects Retrieval & Recommendation", The 5th International Conference on Web-based Learning, July 19-21, 2006, Penang, Malaysia.
- [11] P. Melville, R. J. Mooney and R. Nagarajan, "Content Boosted Collaborative Filtering for Improved Recommendations", Proc. 18th Nat'l Conf. Artificial Intelligence, 2002.
- [12] R. J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization", Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation, 1999.
- [13] R. Jin, L. Si and C. Zhai, "Preference-Based Graphic Models for Collaborative Filtering", Proc. 19th Conf. Uncertainty in Artificial Intelligence, Aug. 2003.
- [14] R. Jin, L. Si, C. Zhai and J. Callan, "Collaborative Filtering with Decoupled Models for Preferences and Ratings", Proc. 12th Int'l Conf. Information and Knowledge Management, Nov. 2003.
- [15] T. Hofmann, "Latent Semantic Models for Collaborative Filtering", ACM Trans. Information Systems, vol. 22, no. 1, 89-115, 2004.