在 WWW 上動態調整建議引擎之設計

Design a Dynamically Adaptive Recommendation Engine on WWW

蘇怡仁　焦惠津　蔡尚榮
國立成功大學電機工程系
台南市大學路一號
iansu@eembox.ee.ncku.edu.tw

摘要

由於網際網路的快速發展，經由全球資訊網來搜尋資訊及網路購物已成為趨勢，為了要提升網站的競爭力，網站管理者除了要不斷地更新及充實網頁的內容以獲得使用者的青睞外，藉由分析使用者目前瀏覽網頁的方式，結合其他使用者類似的瀏覽網頁之使用紀錄，希望能夠有效地評估出使用者所欲尋找的網頁，並即時地提供精確的網頁建議，透過這種讓使用者感覺為其量身訂做之個人化(Personalize) 網頁的呈現方式，加強網站使用的方便性，建立使用者對網站的忠誠度，進而有再次瀏覽該網站的興趣。本研究係利用 web usage mining 之技術，將網站使用紀錄檔中所儲存使用者瀏覽網頁的習慣加以分析歸類，並參考目前網站使用者瀏覽的方式，動態結合 association rules、clustering 及統計等 knowledge discovery 技術，即使網站管理者在線上新增、刪除或異動部分的網頁或其鏈結對象，具有動態調整建議能力之 recommendation engine 一樣不受影響，可以提供使用者最佳網頁連結的建議。

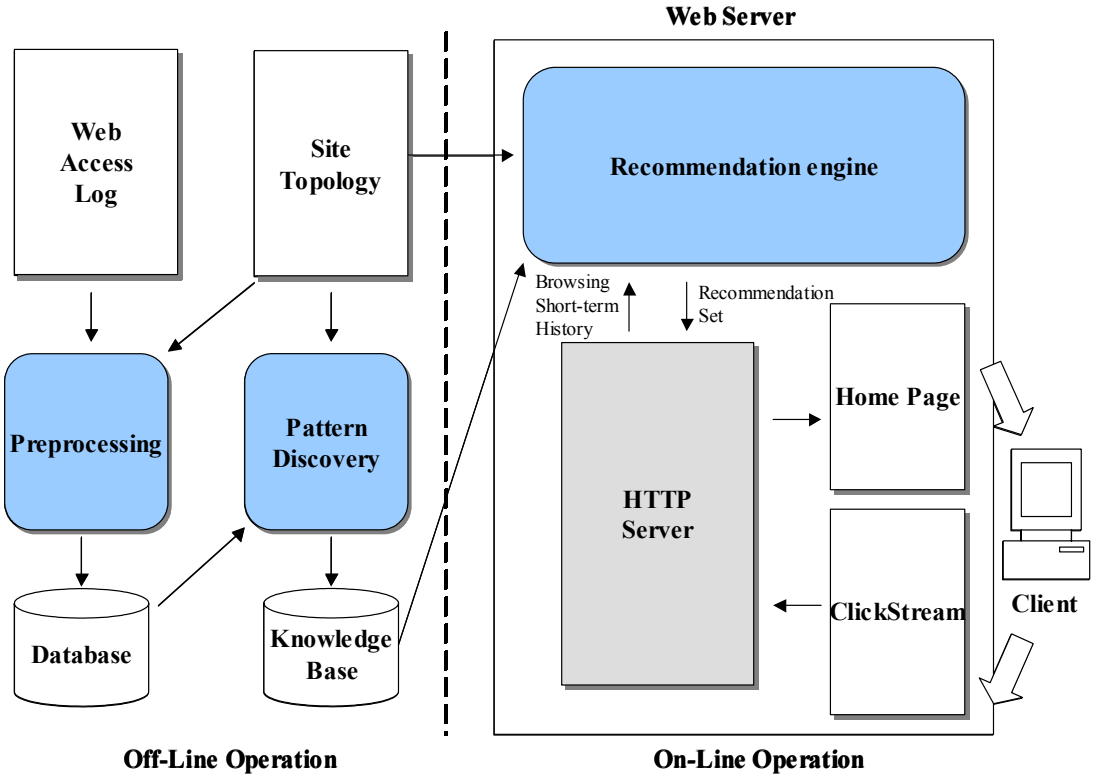
關鍵詞：Recommendation Engine、Web Usage Mining、Association Rules、Clustering

一、緒論

Data Mining 的技術原本就是針對有效地尋找潛藏在大量資料中之 regular pattern 不易解決的問題所設計，而這與本研究之主題"web usage mining"所必須面對解決的問題極為類似，以著名的 Yahoo 網站為例，每天有大約一千六百六十萬個使用者瀏覽，每個小時會產生 48GB 的紀錄檔資料[1]，在面對如此龐大的瀏覽紀錄檔，想要從其中發現出使用者共同的瀏覽行為模式作為 recommendation engine 提供建議的依據，如果只單一地透過統計的結果來提供建議，事實上並無法滿足大部分使用者的需求。根據統計有大部分的使用者都是用 anonymous 來瀏覽網站，這時網站並無法有效的辨認出每一位使用者的身分而使用其個別專屬之 profile 及瀏覽紀錄檔來提供建議，所以能夠提供有效地分析大量的 clickstream 以便讓網站做出更即時的反應是個迫切解決的問題。

Web mining 主要是將 data mining 的技術應用在 WWW 上，一般的研究主要分成三大方向，包含有 web content mining、web structure mining 及 web usage mining[2]。 Web content mining 是一個自動化的處理程序，將原本機器所看不懂的網頁內容，轉換成機器可以判讀的語意，再重新整理與儲存，可利於日後自動化的處理。Web structure mining 則是經由網頁內容之 hyperlink 找出整個網站之網頁架構，一方面可使網站管理者亦於維護網站，另一方面也可以提供 search engine 利用 hyperlink 這項資訊來做 information retrieving 的工作，例如 Google[3]。Web usage mining 首先是根據使用者在網站上瀏覽後所留下的 access log，去除掉其中有些不需要的紀錄，將其重新組織整理，儘可能還原使用者完整的 navigation path，再從中尋找使用者瀏覽網頁的共同習慣，當有部分相同的瀏覽行為出現時，即可透過這些已經找出的 pattern 給予使用者瀏覽網頁的建議。

Web usage mining 的技術，依其操作時間點來區隔判別，主要可以分成兩種模式：離線模式(整批模式)及線上模式(即時模式)，如圖一所示。離線模式其主要的工作為 Preprocessing 和 Pattern Discovery 兩大類。線上模式則是透過 recommendation engine 分析

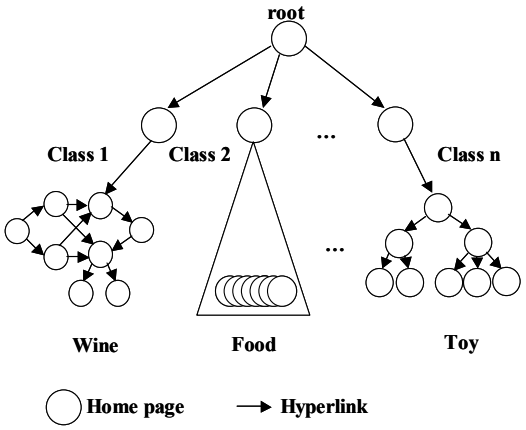


圖一、Web Usage Mining作業流程圖

瀏覽路徑動態地提供使用者即時的 hyperlink 建議。根據一般實務的經驗 Preprocessing 和 Pattern Discovery 工作量約為 80%與 20%的比率[4]。

離線模式之 Preprocessing 的工作依其順序又可細分為 data cleaning、session identify、traveling path complete 及 transaction identify[5]四個步驟。Preprocessing 所要處理的對象為 web access log. 依不同的儲存格式標準，基本上有 NCSA/CERN 的 Common Log Format (CLF)及 Extended Common Log Format (ECLF)兩種格式。依照 ECLF 儲存格式來說明，資料欄位分別為使用者端 IP 位址、使用者 ID、時間/日期、傳輸檔案名稱、連線需求方法(Get 或 Post)、連線狀態、傳輸資料量多寡、由那一網頁所參考、瀏覽器版本及使用者端的作業系統名稱。首先是 data cleaning 的步驟，其工作主要是負責從 web access log 中刪除需求檔案之副檔名不是*.htm 或*.html 的記錄，因為不僅是使用者所點選的網頁會在 web access log 留下檔案名稱的紀錄外，網頁上的每一個圖形檔也會各留下一筆記錄。其次為 session identify 的步驟，使用者從連上網站的第一個網頁開始，一直到最後離開的網頁，期間所瀏覽過的網頁串成之路徑稱之為 session。如果能夠把每一個 session 都獨立開來，就可以有效的了解使用者每次連上網站瀏覽的行為。基本上為了要避免區隔不同使用者 session 的技術會引發侵犯使用者隱私權之爭議，所以採用的辨識方法是以使用者連線的 IP 位址及瀏覽器的版本為分野點，一般的慣例如果同一個連接，兩個網頁的點選時間超過 30 分鐘，會被認定為兩個不同的 session。第三個步驟是 traveling path complete，因為使用者在瀏覽網站的過程中常常有可能會使用”返回”鍵，因為瀏覽器會自動將瀏覽過的網頁資料存入 cache 中，所以當使用者使用”返回”鍵時就會直接從 cache 中取出顯示，而不會再向網站提出網頁的需求，所以也不會在 web access log 中留下紀錄。為了要完整的捕捉到使用者的瀏覽路徑，traveling path complete 不僅會使用到 web access log，同時也必須使用 web site structure 相關的資訊。最後一個步驟是 transaction identify，在這裡 class 所指的是同一個網站上內容相似或同一個路徑 branch 下的網頁所成之集合稱之。當瀏覽的路徑從一個 class 跨到另外一個 class 的時候，這時就是目前 transaction 的結束，同時也是另一個新 transaction 的開始，藉由把一個 session 區分成較短的 transactions，並以 transaction 為單位來做 pattern discovery 的工作，可以得到比用 session 處理得到更好的效能。網站上所有的網頁依其內容區隔成幾個不同的 classes，達到縮小 problem domain 及 transaction 的長度，經由

有效地減少 candidate itemset 的數量進而加快分析處理，以禮品網站之網頁為例，可以分類為食品、酒類、展覽品及玩具等 class 如圖二所示。



圖二、網頁依其內容分類圖

Data mining 所找出的 pattern 主要可以分成兩大類:敘述性的(descriptive)與預測性的(predictive)。敘述性的 pattern 描述了在資料庫裡資料的特性;預測性的 pattern 則根據現有的資料去分析推斷，做為日後預測、分類或決策的依據。Pattern discovery 最主要的任務就是從 preprocessing 處理過後的 access log 資料庫中找出具有特殊意義的 pattern 作為預估使用者下一個瀏覽網頁的依據，其中較常被使用的呈現型態有 association rules、clustering 及 classification 等。當 pattern 被找出後就要執行 pattern evaluation，其目的是要保留真正頻繁出現或具有特殊意義的 pattern，所以說只有滿足於某些訊息強度標準且分別保有特別 knowledge 的 pattern 才會在 pattern evaluation 過程後留下來。

線上模式之 recommendation engine 的工作除了一方面儲存每一個正在網站上瀏覽的行為外，另一方面更要隨時準備依據使用者目前短程的瀏覽紀錄和在 pattern discovery 所發現的 pattern 做 pattern matching 的工作，找出幾個最適合的網頁給予使用者最佳瀏覽網頁的建議，但是為了顧及使用者使用上的感覺，這一部分的工作必須同時兼顧準確性與即時性。

接下來本論文的內容架構如下：第二節相關研究，介紹目前在 pattern discovery 階段之 pattern 的分類及找出的方法；第三節介紹 clustering 中之 Feature Matrices model；第四節說明如何設計具有動態調整能力的 recommendation engine；第五節根據所設計的的系統架構所做的結論及未來研究方向。

二、相關研究

在本研究中使用到 association rules 及 clustering 於 web usage mining，以下就針對這兩種 pattern discovery 的方法來討論。

2-1 Association Rules

Association rules 基本上是要找尋資料屬性間的關係，以一個較有名氣的 Market Basket Analysis 中的 association rules 而言，在某一個事件所有項目的集合 $\Omega=\{A_1, A_2,\cdots, A_k\}$ 中找出所有的 A_i 和 A_j，使得

$$P(A_i \cap A_j) \geq 0.05 \qquad \text{式(1)}$$

$$P(A_i \mid A_j) \geq 0.1 \qquad \text{式(2)}$$

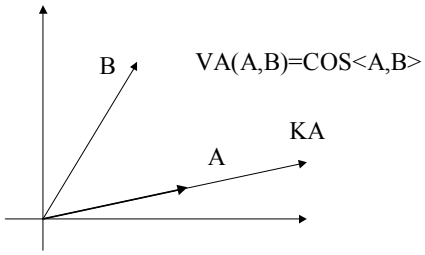
這裡的 A_i 與 A_j 皆是屬於顧客可能會購買的項目，(1)、(2)式皆滿足 0.05 的 support 與 0.1 的 confidence 兩個 threshold 的值才會被列入考慮[6]。其中 support 是指在所有的 transaction 中出現的機率，confidence 則為出現 A_j 後再出現 A_i 的機率。最有名的演算法為 Apriori[6]，透過有效的 threshold 值的設定有效的刪減 candidate itemset 的數目，大幅減少 association rules 尋找的問題空間。另外在[7]中引進了 hash function 的使用，在減少 candidate itemset 數量上也得到了很好的效果。

2-2 Clustering

Clustering 演算法最主要的做法是將大量的資料依其相似性區分成幾個不同的 cluster，對於每一筆資料屬於那一個 cluster 事先通常未知，藉由相似度量測計算出其 neighbor，進而判斷是屬於那一個 cluster。在 anonymous web usage mining 中利用屬於同一 cluster 中的組成成員皆具有相類似瀏覽行為的特性來提供建議，因為使用者都是以 anonymous 來瀏覽網頁，所以我們無法有效的辨認出使用者的身分，基本上的做法是先收集使用者瀏覽行為，再做相似度的計算與在依其計算結果來判定是隸屬於那一個 cluster，然後使用該 cluster 共同的行為模式來提供建議。目前為止大部分的研究集中在 Collaborative Filtering 演算法，其中最成功的方法是 Markov model(Probabilistic based)[8]和 Vector model (Distance based)[9]兩種。

2-3 Similarity Measurement

通常用相似度計算來判斷資料隸屬於那一個 cluster，相似度的計算方式則有 angular distance 之 vector angle 及 Euclidean distance 兩種[10]。Vector angle 是利用兩個向量之間的夾角大小作為近似值判斷的依據，當角度越小時則表示兩個向量相似度越高，反之則相似度越低，角度最小的 cluster 就是將目前向量分給該 cluster，如圖三及式(三)所示。

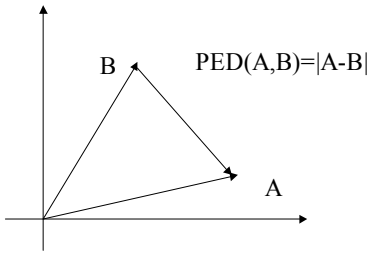


圖三、Vector Angle 相似度量測法

式(三)

$$VA(\vec{A},\vec{B}) = \cos\theta = \frac{\vec{A}\bullet\vec{B}}{\left|\vec{A}\right|\left|\vec{B}\right|} = \frac{\sum_{i=1}^{N} a_i b_i}{(\sum_{i=1}^{N} a_i^2)^{\frac{1}{2}} (\sum_{i=1}^{N} b_i^2)^{\frac{1}{2}}}$$

Euclidean distance 基本做法恰跟 vector angle 相反，藉由將兩個向量間相異的程度量化，Euclidean distance 越大就表示兩個向量的相異度越高，如圖四及式(四) Pure Euclidean Distance 的計算所示。



圖四、Euclidean distance 相似度量測法

式(四)

$$PED(\vec{A},\vec{B}) = \left|\vec{A}-\vec{B}\right| = (\sum_{i=1}^{N}(a_i - b_i)^2)^{\frac{1}{2}}$$

三、Feature Matrices Model

Feature Matrices(FM) model[10]其設計的主要目的是為了有效的解決 anonymous 之 web usage mining 的問題，因為大部分的使用者都是以 anonymous 的身分來瀏覽網站，造成網站無法正確地辨識每一位使用者的身分，這時網站便無法使用使用者個別之 profile 及獨有的使用紀錄來提供使用者適時的瀏覽建議，只能以 Collaborative Filtering 的方式，透過將使用者瀏覽的行為 clustering 的方式來提供使用者建議。基本上 FM model 是屬於 vector model 的延伸，為什麼不使用前一節所敘述之 Markov model 和 vector model 而要重新引進 FM model 呢？因為 Markov model 雖然可以捕捉到網路瀏覽順序的資訊，但是 time complexity 太高，而無法提供因應使用者短程的瀏覽行為，動態地提供網頁瀏覽的建議。vector model 恰恰相反，它具有即時分析的能力卻無法表示出使用者瀏覽的順序。在[10]中使用者目前短程瀏覽的行為透過 PPED 的 dynamic clustering 之演算法來判定是屬於那一個 cluster，這種 partial match 的方法具有非常好的 performance，可以滿足網站提供使用者動態且及時瀏覽建議的需求。

3-1 FM Model

簡言之，FM model 就是把要 problem space 中被列入考慮之特性予以有效的量化。應用在 web usage mining 這個方面，有三個特性是目前被考慮的，網頁被點選的次數(hit count，H)、網頁瀏覽的順序(browsing sequence，S)及每個網頁被使用者瀏覽的時間長短(visit time，T)。其中前兩者是屬於 spatial 特性，而最後一個是 temporal 的特性。當使用者瀏覽網站完整的路徑所成之 session 依網頁內容屬於不同的 class 而分割成數個長短不一的 transaction，每一個 transaction 就是要用 FM model 依前面所敘述之特性 H、S 及 T 來 mapping。

例如：使用者在網站瀏覽的 session(A)為

$$x_a \to x_1 \to x_2 \to x_3 \to x_2 \to x_b \to x_c$$

其中網頁 X_1、X_2 及 X_3 屬於同一個 class，所以 transaction(A)為

$$x_1 \to x_2 \to x_3 \to x_2$$

所以針對 transaction(A)之 universal feature matrices 可以表示為

$$U^{fm} = \{M^{H}_{3^2}, M^{S}_{3^2}, M^{T}_{3}\}$$

依特性不同可以分別表示 FM model 成

$$M_{3} = [(x_1), (x_2), (x_3)]$$

$$M_{3^2} = \begin{bmatrix} (x_1,x_1), & (x_1,x_2), & (x_1,x_3), \\ (x_2,x_1), & (x_2,x_2), & (x_2,x_3), \\ (x_3,x_1), & (x_3,x_2), & (x_3,x_3), \end{bmatrix}$$

套用在本例中，FM model 如下所示

$$M^{H}_{3^2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad M^{S}_{3^2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 3 & 0 \end{bmatrix}$$

$$M^{T}_{3} = [10 \quad 20 \quad 10]$$

(假設每頁瀏覽時間長度為 10 秒)

至於屬於 cluster 之不同的 FM 要如何形成 cluster model 呢？式(五)表示目前 cluster model，式(六)說明當有一新的 FM 要如何加入此 cluster model 中，經過這取平均值的方式可以很快的完成 dynamic clustering 的動作。

式(五) $$M^{F} = \frac{1}{N}\sum_{i=1}^{N} M^{F}_{i}$$

式(六) $$M^{F} \leftarrow \frac{1}{N+1}(N \times M^{F} + M^{F}_{j})$$

3-2 PPED (Projected Pure Euclidean Distance)

PPED 是改進 PED 式(四)相異度計算之時間複雜度的方法，將使用者目前線上短程瀏覽的路徑轉成 FM，極有可能出現的結果都是 sparse matrix，如果依照 PED 的計算方式，一但有新的對象要決定是屬於那一個 cluster 就要和所有 cluster 中同性質的 FM 來計算，因為每一個 FM 都是儲存該 class 內所有網頁相互間的資訊，所以要 matrix 對 matrix 做所有 element 的比較，其結果將會大幅增加計算的時間，因為只考慮短程瀏覽紀錄，所以不需要對 class 內所有的網頁來做考量，因為這將是違反 real-time 的需求，所以在 PPED 裡以使用者短程瀏覽路徑之網頁順序為主，來跟所有的 class 做之相異度的計算，如此利用有效的 partial order matching 的方式，以便在更短的時間內提供使用者瀏覽的建議。

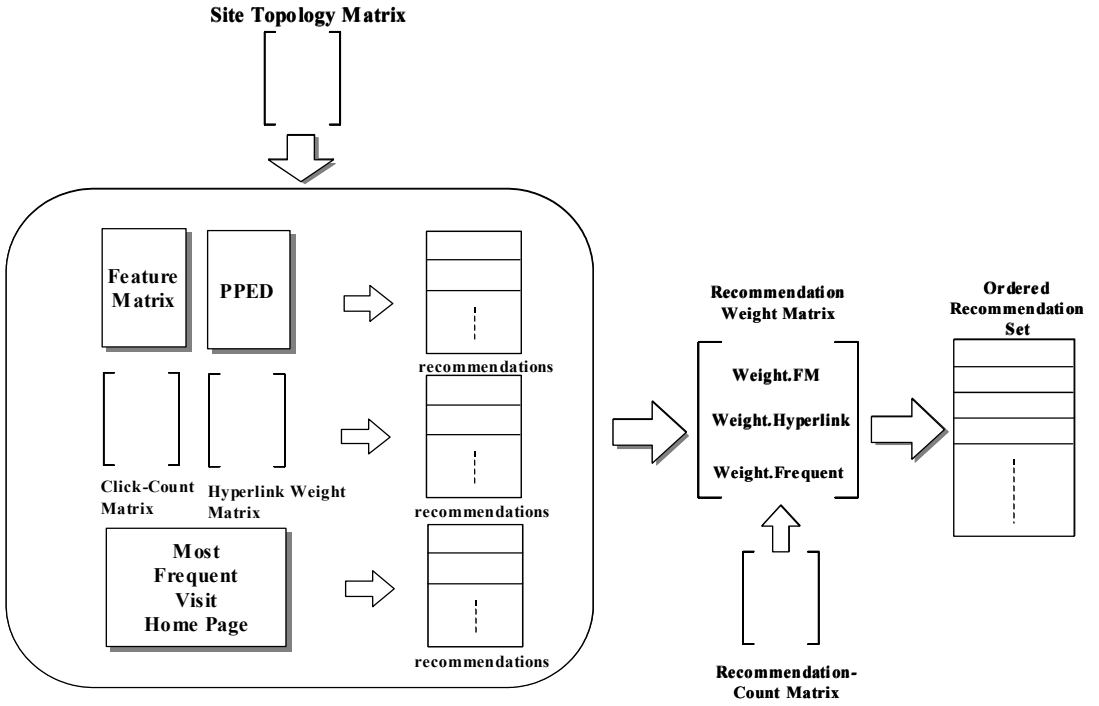
式(七)

$$PPED(\vec{A}, \vec{B}) = \left(\sum_{i=1, a_i \neq 0}^{N} (a_i - b_i)^2\right)^{\frac{1}{2}}$$

四、Recommendation Engine 之設計

為什麼會有這個想法要設計一個具有即時地動態調整建議能力的建議引擎呢？首先我們強烈地認為線上使用者瀏覽網頁的方式是一個動態的行為，例如早上上網的族群跟下午或晚上的族群比較是截然不同，線上使用者可能隨時都在改變瀏覽網頁的興趣及目的。所以建議引擎之建議網頁的產生方式，除了要顧及高度的準確性外，更必須要能夠跟隨使用者瀏覽趨勢的改變而彈性地提供更動建議方向。若只是很單純地使用從離線模式下分析所發現到的 navigation pattern 來產生建議網頁，就現實考量的確是無法提供使用者準確的建議。而且網站有可能基於某些目的下會隨時時時新增、異動或移除部分網頁的內容或其鏈結對象，例如新聞網站有新的新聞網頁網頁要增加及舊的新聞網頁要移除，特別是即時新聞的網頁的不分更是如此；另外 E-Commerce 的網站也會面臨有新推出產品的網頁要增加和舊產品出清且不再進貨時，該項產品網頁要被刪除的動作，一但上述的任一情況發生，因為原先之 navigation pattern 有可能就不是完全適用，建議引擎應有馬上察覺及因應的機制。

首先為了要能夠準確地提供使用者動態調整的瀏覽建議，其資訊的來源就不能單純的只有離線作業時所發現的 pattern 及目前使用者的短程瀏覽紀錄，還必須時時掌握目前線上使用者瀏覽的趨勢及網站內容有無異動情形，繼而隨時更新網站架構資訊，雖然 FM model 有 dynamic clustering 的方法，但是我們認為還是不足以完全表示出線上資訊的重要性。所以提供瀏覽建議的依據必須同時考慮離



圖五、Recommendation Engine產生建議之架構圖

線資訊及線上資訊。

本研究主要就是要解決 recommendation engine 所面臨的兩個問題(1)在網站管理者更動網站架構及內容後，即使在離線模式所找出的 pattern 不完全適用時，仍然可以準確地提供使用者下一個瀏覽網頁的建議。(2)可以隨時掌握使用者瀏覽網頁內容喜好的改變，機動地調整建議的內容，並依其預估之重要性予以排列。為了滿足上述之需求除了使用[10]所設計的 FM model 利用 PPED 的方式去透過 partial matching 來尋找所屬的 cluster 外，在圖五中我們設計 site topology matrix 來儲存整個網站的架構，利用四個不同時間長短的 click-count matrix(短、中、長及歷史)來分別統計 hyperlink 被點選的次數，比較其相互間的關係來動態調整 hyperlink weight matrix 之內容，進而捕捉使用者的短期使用之趨勢的資訊。另外藉由 most frequent visit web page 來記錄網站上那一個網頁被瀏覽次數做多，作為建議產生方法之一。再將這些經由各種建議方法所產生的建議網頁乘上對應在 recommendation weight matrix 中的數值，然後所有的建議網頁依各自之訊息強度排序，透過 HTTP sever 併入網頁中來提供給使用者作瀏覽的參考。

Site Topology Matrix其設計的目的主要是為了以一個有效存取的方式來表示整個網站上網頁相互間連結的關係。其資訊的取得只需透過一個簡單的crawler程式，就可以很快的完成web structure mining的工作，有很多的研究是以string的方式來儲存，這種方式最主要的問題除了在維護資料的結構完整性不易外，也較困難在其中很快的找到所要的資訊，主要的癥結是儲存的資料是要給機器處理，必須以機器的角度來設計。特別是當處理到traveling path complete的程序時，屬於同一個使用者之上下兩筆網頁需求，可以很直接的判斷出是否有直接的hyperlink參考，如果沒有就可以直接透過該session的瀏覽路徑判斷是否使用"返回"鍵或是直接使用了網站所提供的recommendation，這可以省去parsing string才能得到的資訊，大幅縮短處理的時間。值得注意的是當整個網站的網頁數目太多時，將網頁依其分屬不同class，以class內的網頁為對象來做site topology matrix，避免matrix的空間太過於龐大。

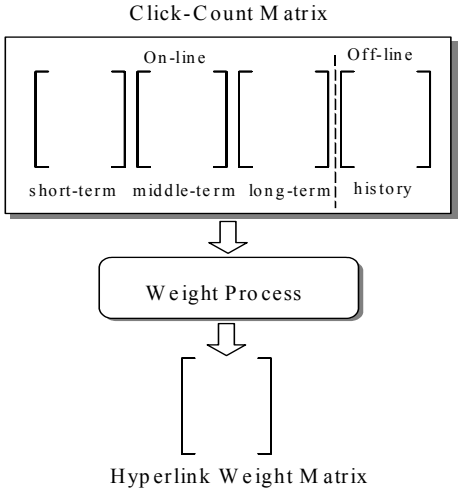
Click-Count Matrix 紀錄所對應之 site topology matrix 內所有 hyperlink 被使用的次數多寡，在 site topology matrix 中不為 0 的位置，表示該位置對應有一個 hyperlink 存在，每次只要有使用者點選 hyperlink，就在 click-count matrix 對該位置做加 1 的動作，要注意的是如果使用者點選 hyperlink，但對應到 site topology matrix 位置為 0，此現象表示使用者只是使用了 recommendation engine 的建議，所以不需做加 1 的動作，但要紀錄該項建議是由

那種建議方法所產生。依 click-count matrix 累計統計的時間可以分為 on-line 和 off-line 兩種。前者統計線上使用者點選的次數，用以判斷目前網頁短期的使用趨勢；後者則在做 web usage mining 的離線作業時才進行統計，是使用者長期使用的象徵。

基本上 click-count matrix 是本研究的重心所在，on-line click-count matrix 每隔一段固定單位時間就要被分別儲存一次，其目的是將紀錄結果轉換成更高層次的意義。利用這種紀錄 hyperlink 使用次數的方式，可以隨時掌握使用者喜好的行為是否和先前有所改變或是對哪些網頁瀏覽的次數增加，透過固定時間依照 click-count matrix 的內容做一次各個 hyperlink 之 weight 調整，所以一但使用者對於某些網頁瀏覽興趣逐漸提昇或減少，recommendation engine 可以馬上察覺並逐漸調整其建議內容。其做法是以短中長三個不同時間長短的 click-count matrix 統計結果為依據，依其漸增或漸減的參考次數來動態調整相對應之 **Hyperlink Weight Matrix** 中的內容，如圖六所示，其判斷的依據及調整方式之規則由表一來說明。

另外我們也統計了整個網站內所有網頁的瀏覽次數，因為雖然有 click-count matrix 紀錄 hyperlink 被選擇的次數，但是考慮的對象之範圍可能僅侷限在一個 class 之內，換言之只是 local optimize，若以尋找 global optimize 的角度考量，則建議形成的另一個方法就是在網站內 **Most Frequent Visit Home Page**。要注意的一點是網頁的候選者必須要扣除掉 root 網頁，因為大部分的使用者都是從這一網頁開始瀏覽的。

最後是 N*1 的 **Recommendation Weight Matrix**，其 N 值大小對應到產生建議方法之多寡，換言之以本研究所列舉的方法為例，有 FM model、click-count matrix 及 most frequent web page 三種，這時 recommendation weight matrix 就是 3*1 的陣列。儘管只有三種產生建議的方法，但是有可能每一種方法產生不只一個建議對象，因為這些網頁都同時滿足該項建議方法被建議之 threshold 值，但要成為真正的建議對象之前，必須乘上 recommendation weight matrix 中該項建議產生方法所對應的 weight，所有建議再依其計算結果大小排序，於是要推薦給使用者的 recommendation set 這時才算完成。要注意的是如果有依各建議對象是被幾種建議方法重複產生時，則該建議在各項方法乘完 weight 之後，還必須多做一次加總，表示這個建議同時被幾種方法所推薦，有著極強的訊息強度。



圖六、Click-Count Matrix vs. Hyperlink Weight Matrix

表一、weight 動態調整表

Id 值	短->中	中->長	長->歷史	調整方式
0	0	0	0	0
1	0	0	1	X
2	0	1	0	X
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1^{+}
6	1	1	0	1^{+}
7	1	1	1	1^{++}

0->減少　X->不變　1->增加　+->訊息強度

基本上 recommendation weight matrix 和 hyperlink weight matrix 之 weight 產生及調整的方法是相同的。當使用者皆使用了那一項建議，則該方法所對應之 count 就做加一的動作，透過每各單位時間所產生的 **Recommendation-Count Matrix**，再將結果產生短中長的 matrix，然後運用表一 weight 判斷調整法則，來產生隨著使用者行為動態調整的 recommendation weight matrix。

五、結論

在本篇論文中，我們針對在 WWW 上具有動態調整建議能力之 recommendation engine 的設計除了引進 FM model 的技術外，

也提出了(1)click-count matrix 及 hyperlink weight matrix 來依據使用者的短期瀏覽趨勢來提供建議，(2)site topology matrix 以數值取代字串來儲存網站的架構，及時反應網站架構的變化，(3)recommendation weight matrix 及 recommendation count matrix 透過使用者使用建議的次數來調整建議的排序，(4)同時利用三種方法提出建議來增加建議涵蓋範圍，以改進 recommendation engine 建議的準確率。希望透過這些新方法的加入，可以讓網站之 recommendation engine 的反應能力加以改進，特別是當使用者的 browsing short-term history 只有一、二頁，尚無法區隔隸屬於那一個 cluster 時，也能透過簡單的統計技術來提供建議。不只依據在離線作業時所找出之 pattern 來做建議，並且增加幾種不同產生方法來給使用者更多建議的選擇，增加建議所涵蓋的範圍，以解決單一方法產生建議個數太少的問題。

在未來的研究除了朝向更即時且精確的建議引擎改進外，同時也考慮其他三個方向，第一項是增加使用者資料的來源，透過 web access log 要捕捉使用者在網站上的所有行為，然後判斷屬於那一個 cluster，依照該 cluster 共同行為的 pattern 來產生 recommendation，事實上是不足的，因為基本上 web access log 是為了網站除錯的目的所設計，因此有許多資訊是不包含在 web access log 中的，例如在網站上進行 e-commerce 的行為，線上採購者有許多行為發生的時機在 e-commerce 上是被重視的，這些事件有在購物車中買了哪些貨品、買賣金額之多寡、哪些貨品被放入購物車中後來又移出，及最重要的是在什麼情況下完成交易或是放棄這次交易，所我們強烈的為除了 web access log 外，application server log 也必須一併列入考慮與分析。

第二項是加強對隱私權的處理，為了更確實的掌握目前連線之使用者的確實身分，有很多現行的做法引發關心隱私權處理的問題，例如在使用者端放入 cookie 以便辨認使用者身分、透過 packet sniffer 直接捕捉所有進出網站的資料或是放入代理人程式(agent)主動跟網站回報資訊，即使現在 W3C 倡導的隱私權限平台的專案（Platform for Privacy Preferences Project，P3P），一份用來強化隱私權策略能力的建議書，也還沒有在網路社會重形成一普遍之共識。

最後 XML 文件在 WWW 上已形成不可輕忽之趨勢，所以除了要對 HTML 的網頁做 web usage mining 外，在未來如何增加對 XML 文件的處理，也是我們目前為止想要積極研究的方向。

六、參考文獻

[1] http://docs.yahoo.com/docs/pr/release634.html

[2] R. Cooley, B. Mobasher and J. Srivastava, “Web Mining: Information and Pattern Discovery on the World Wide Web”, In Proceedings of ICTAI’97, 1997.

[3] S. Chakrabarti, B.E. Dom, D. Gibson, J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagoplan, and A. Tomkins, “Mining the Link Structure of the World Wide Web,” IEEE Computer, PP.60-67, August 1999.

[4] M. S. Chen, J. Han and P. S. Yu, ``Data Mining: An Overview from Database Perspective," IEEE Trans. on Knowledge and Data Engineering, Vol. 8, No. 6, pp. 866-883, December 1996.

[5] R. Cooley, B. Mobasher and J. Srivastava, “Data preparation for mining World Wide Web browsing patterns”, Journal of Knowledge and Information Systems 1, 1, 1999.

[6] R. Agrawal, R. Srikant: "Fast Algorithms for Mining Association Rules", in Proc. of the 20th VLDB, Santiago, Chile, September 1994.

[7] J. S. Park, M. S. Chen and P. S. Yu, “An Effective Hash Based Algorithm for Mining Association Rules”, IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 2, pp.209-221, 1998.

[8] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, “Visualization of Navigation Patterns on Web Site Using Model Based Clustering”, In Tech. Report MSR-TR-))-18, Microsoft Research, Microsoft Corporation, Redmond, WA, March 2000.

[9] A. Joshi and R. Krishnapuram, “Robust Fuzzy Clustering Methods to Support Web Mining”, In Proc. of SIGMOD ’98 Workshop on Data Mining and Knowledge Discovery, Seattle, June 1998.

[10] C. Shahabi, F. Banaei-Kashani, J. Faruque, and A. Faisal, ”Feature Matrices: A Model for Efficient and Anonymous Web Usage Mining”, EC-Web 2001, Germany, September 2001