關鍵字為基礎的多主題概念飄移學習

林熙禎　　林文羽
國立中央大學資訊管理學系
sjlin@mgt.ncu.edu.tw
100423033@cc.ncu.edu.tw

摘要

隨著網際網路的蓬勃發展，使用者能夠輕易取 得大量的資訊。然而，在此同時，使用者也得面對 資訊過載的問題，如何有效取得當下使用者興趣的 資訊是資訊過濾系統主要目的。然而使用者興趣會 隨著時間轉變，並且包含多種概念，這就成為了多 標籤分類下的概念飄移問題；同時文件也常屬於多 個類別，若僅依照文件的主要概念，將之分類，則 可能讓使用者錯過潛在感興趣的相關文件。本研究 提出一個以字詞網路為基礎的使用者模型，透過它 可以依照使用者對於多個概念的喜好對文件進行 過濾，而在目標概念發生變化時，也能夠適當的偵 測並更新模型。

關鍵詞：概念飄移、資訊過濾、使用者模型

1. 前言

資訊過濾系統是以一種以使用者的回饋建構的使用者模型為基礎，從大量的文字檔案組成的資料串流中，過濾掉不相關資訊的自動化資訊系統[1]，例如個人化新聞過濾器，垃圾郵件過濾等等。由於是以使用者模型為基礎，一個有效的模型便成為了資訊過濾系統是否成功的重要關鍵。

然而，在現實生活中，使用者的興趣並非一成不變，對於資訊的需求也會隨著時間而改變。當使用者的需求改變時，過去使用者所提供的回饋已不再適合用來作為過濾的參考依據。在文件分類問題中，Tsymbal於2004的研究將這種資料隨著時間改變而產生不同分布的問題稱為概念飄移[2]。

此外，一份新聞文件可能不僅僅只包含一個概念，往往可能同時容納多種概念，而兩者可能原本沒有直接關聯，但因為某些事件或因素使得兩者相互影響。

例如一篇新聞是關於奢侈稅，奢侈稅的公布使得房市產生影響，然而該篇文章內容多介紹奢侈稅新法的內容、實施辦法、時間等等，文末關於影響層面可能只有專家的一小段話，因此主要概念為法律，若因此而將之歸類為法律新聞，則關注房市新聞的人就會因此而遺漏其感興趣的文件，即使房地產的影響在該篇文章只是次要概念。

2. 文獻探討

依照概念飄移的轉換速度與程度，Žliobaitė於2010的研究，將它分為四種類型[3]：突發性的概念飄移(sudden drift)，漸進式飄移(gradual drift)，增量式飄移(incremental drift)與重複性內容(reoccurring context)。目前的研究對於漸進式與增量式飄移有一定的適應能力，然而在遭遇突發性的概念飄移時，往往因為使用者提供的回饋不足，缺乏快速的反應能力。

概念飄移的問題，目前已經有許多的研究提出解決方法，依照調整的時間點，可以分為持續學習的方法 (Evolving learners)以及以偵測為基礎的學習方法 (Learners with triggers)。

在持續學習的方法中，合議分類器(Ensemble Classifier)是目前最廣為使用的方法。透過同時訓練多個分類器，再依據分類器的準確性來給予不同的權重，最終再以投票的方式，來決定其分類，以提升準確性。如[4]，這類方法的主要缺點在於運算成本較高，並且僅僅考慮分類演算法的多樣性(diversity)，不能保證其分類準確度，如何選擇、結合分類器的預測結果將是這類方法是否成功的重要關鍵。

而以偵測為基礎的學習方法，則是透過偵測的機制，來決定是否對學習器或分類器進行調整。其中最主要的方法為調整視窗法。依照視窗內的訓練資料訓練出來的分類器，其分類的準確率是否穩定來調整大小。若準確率穩定，表示使用者興趣相當穩定，此時可以擴大現有的視窗大小；相反的，若在一個時間區間準確率突然下降，則使用者可能發生概念飄移，則依照預先定義的參數，做適當的縮小，如[5]與李浩平2011的方法[6]。

在2011年李浩平提出的方法[6]，可以讓過濾系統在遇到突發性概念飄移時，訓練資料不足的情況下，提供一個足夠精準的資訊過濾模型。然而使用K-core對字詞網路所萃取的核心特徵雖然能夠找出文件的主要概念，若應用於多標籤文件顯然不足，新聞常是包含多概念的；另一方面，語意中心的建立方式也僅適用於使用者目標概念為一個情況下，在現今對於概念飄移的研究，大多都是在目標概念只有一個的情況下進行。然而，在實際的應用中，使用者所感興趣的目標概念常常不只一種，並且會隨著時間、空間的改

變，使用者對於每一個概念的偏好程度也會有各自的變化，這就成為了多標籤分類中的概念飄移問題，然而目前關於概念飄移的研究，卻多是以傳統的單標籤分類問題為主。如Xioufis, et al. 於2011提出的研究[7]指出目前多標籤分類問題下產生的概念飄移問題，多半透過問題轉化的方式，將問題轉化成多個單標籤分類問題，再個別解決發生的概念飄移問題。然而，此作法因為問題轉化的方式容易將因為將訓練集改變而產生類別不平衡(Class imbalance)的問題，並且忽略了各個目標概念也有各自的變化。

李浩平的方法中[6]大量依賴Google，導致實做過程中容易被鎖 IP、速度慢、Google資料量不固定導致 NGD 值變動大等缺點。因此本研究主要是利用鄭奕駿2012[8]以離線搜尋Wikipedia解決對Google過度依賴問題的研究，改善[6]之不足。透過分析NGD形成的字詞網路，達以下目標：

(1) 建立符合使用者興趣的多概念模型。

(2) 透過使用者模型找出潛在感興趣的文件。

(3) 適應多目標概念情況下發生的概念飄移。

3. 系統設計與架構

圖 1 使用情境

使用情境如圖 1所示，假設有一新聞閱讀器，它會定時從各個新聞來源接收新的新聞，並且依據使用者興趣過濾。對於過濾的結果，使用者可以給予回饋，讓新聞閱讀器可以隨著使用者的目前興趣而改變。

本研究提出的系統架構分成兩個階段，分別為使用者模型建立階段(圖 2)、文件過濾階段(圖 3)。一開始，先由使用者回饋數篇文件，建立使用者模型，接著依照使用者模型對於新進文件進行過濾，並以F-measure偵測概念是否穩定，若發生飄移，段依照使用者回饋的文件對模型進行更新。

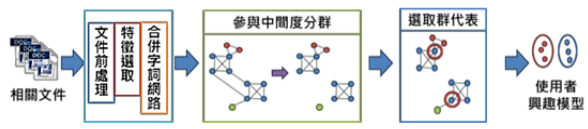

圖 2 系統架構圖：使用者模型建立階段

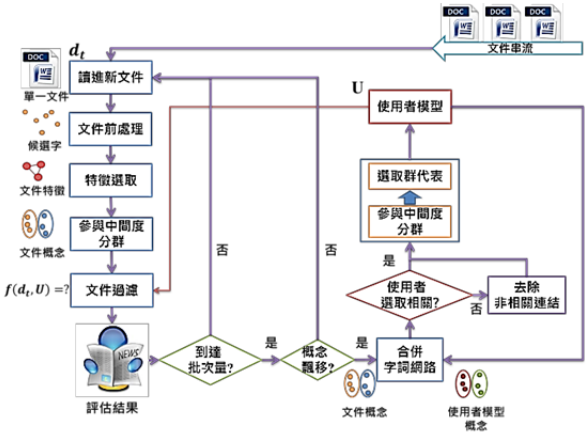

圖 3 系統架構圖：文件過濾階段

3.1 文件前處理與特徵選取

在圖 2與圖 3的文件前處理部分，本研究延用[6]的方法，經由詞性與關鍵字合併、過濾字詞長度、Wikipedia搜尋結果數三個過濾條件篩選出候選字；而在特徵選取部分，本研究將篩選後的候選詞，兩兩計算NGD並排序，去除NGD大於1與無限大後取前50%建立字詞網路。

3.2 參與中間度分群

本研究參考[9]提出的以參與中間度為基礎的分群方法，而參與中間度的計算，則參照[10]。圖 4為此方法的虛擬碼。

```
betweennessClustering (Graph g, int NumEdgesToRemove)
1.  NumEdgesToRemove = g.getEdgesCount ()* β    //停止條件
2.  for(int I =0; i< NumEdgesToRemove ; i++){
3.      Edge to_remove = null;      // to_remove為將要被移除的邊
4.      double maxScore = 0;
5.      for(every edge e in g){      //找出參與中間度最高的邊
6.          score = getBCscore(e);
7.          if(score > maxScore ){
8.              to_remove = e;
9.              maxScore = score;
10.         }
11.     }
12.     g.removeEdge(to_remove);          //移除邊to_remove
13. }
14.  clusterSet = g. getWeakComponentClusterer(); //取得切割後之子網路
15.  return clusterSet;
```

圖 4 參與中間度分群虛擬碼

在圖 4中，β $(0<\beta\leq 1)$代表圖 g 將要被去除的總邊數的比例，並依兩者乘積當作演算法的停止條件。to_remove代表將要被移除的邊，圖g的每個邊會透過getBCscore()來取得它的參與中間度。每次找到最高的參與中間度邊後移除，直到停止條件滿足，便透過getWeakComponentClusterer()來取得切割後的子網路，每個子網路當作一群。分群後各個子字詞網路成員大於2個的則稱為一個概念。

在建立使用者興趣模型階段，如圖 2，需事先有數篇使用者回饋的相關文件。每份文件經過前處理與核心特徵選取的步驟後，各篇相關文件產生的特徵字詞，合併後兩兩透過Wikipedia搜尋結果數，計算NGD相似度，去除NGD大於1與無限大的結果，取前50%的連線節點建立字詞網路，並且令其為$NGD_{threshold}$。接著再透過參與中間度分群，以β_{multi}為去除邊的比例，萃取出足以代表使用者興趣模型的概念群。

而在文件過濾階段，如圖 3所示，對每一份新進的文件 d_t。，將透過參與中間度分群，以β_{single}為去除邊的比例，萃取出一個或以上的概念，以保留該篇文件可能擁有的次要概念。

圖 2與圖 3的選取群代表，則會在每一個概念中，依照degree的排名，挑選排行前 的節點代表該群$(0 < \gamma \leq 1)$，被挑選中的節點組合起來，便成為使用者興趣模型U。

為了因應使用者興趣的轉變，依照使用者所給予的相關與非相關回饋，會對於原本的使用者模型進行修正。

而在圖3中，若偵測到概念飄移的發生，則進入使用者模型更新的階段。在此階段系統會依照使用者給予的相關或非相關回饋來對於使用者模型進行修正。對於每一份新進的相關文件 d_t，經過以下步驟調整使用者模型U：

(1) 合併字詞網路：合併 d_t與U中概念的關鍵詞(取聯集)，兩兩配對後透過進行NGD公式的計算，將NGD值由小至大排序，去除NGD無限大的結果，並保留NGD值大於或等於$NGD_{threshold}$的連線建立字詞網路。
(2) 利用參與中間度演算法進行分群。
(3) 選取群代表：依照degree排序，於各群選擇固定比例 γ的字詞代表該群。

而對於每一份新進的非相關文件，則經過步驟(1)後，於字詞網路中，移除與非相關文件特徵相鄰的連線，進入(2)、(3)，產生新的使用者興趣模型。

3.3 文件過濾

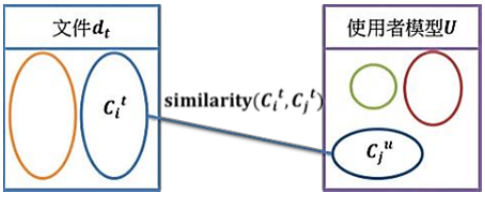

圖 5 文件過濾示意圖

此小節將介紹文件過濾階段(圖 3)，如何運用建立好的使用者模型，對於時間點t進來的文件進行過濾，判斷該文件是否與使用者的興趣相關。如圖 5所示，令similarity$\left(C_i^t, C_j^u\right)$代表文件的概念$C_i^t$與使用者模型$C_j^u$之間的相關度，可以表示成：

$$\text{similarity}(C_i^t, C_j^u) = |\{(w_i, w_j) | w_i \in C_i^t, w_j \in C_j^u, \text{NGD}(w_i, w_j) < \text{NGD}_{\text{threshold}}\}|$$

dc_t為文件d_t的概念集合。以下公式中，$\text{sim}(d_t, U)$代表文件d 與使用者模型U的相關性程度；$f(d_t, U)$則為相關性的判定，1表示與使用者興趣相關，0則反之。

$$sim(d_t, U) = max\{\frac{\text{similarity}(C_i^t, C_j^u)}{|C_i^t| \times |C_j^u|} | C_i^t \in dc_i \text{and} C_j^u \in U\}$$

$$f(d_t, U) = \begin{cases} 1, & \exists C_i^t \in dc_i, \exists C_j^u \in U \text{ such that } \frac{\text{similarity}(C_i^t, C_j^u)}{|C_i^t| \times |C_j^u|} \geq \alpha \\ 0, & \text{otherwise} \end{cases}$$

取文件任一概念群C_i^t與使用者模型中的任一概念C_j^u之間，$\text{similarity}(C_i^t, C_j^u)$佔所有兩者可能連線數的最大比例。高過門檻值α $(0 \leq \alpha \leq 1)$才判定該文件為使用者感興趣的。

3.4 概念飄移偵測與處理

在現實生活中，使用者的興趣並非一成不變。當使用者的興趣於下個時間點突然轉變時，原本系統判定與使用者的回饋會產生衝突，造成F-measure值突然下降，為了衡量這是否為概念飄移的現象，將使用Page-Hinkley Test[11]來監測F-measure值是否有過大的起伏，藉以判定使用者的概念是否穩定，若概念穩定則不需要進行模型更新或是重建的動作。圖 6為概念飄移偵測的虛擬碼。

```
driftDetection(usermodel U, testset S)
1.  for every time t
2.      Fmeasure_t = U.getFmeasure(S);
3.      avgFmeasure_T = (1/T) Σ_{t=1}^{T} Fmeasure_t;
4.      m_T = Σ_{t=1}^{T} (Fmeasure_t − avgFmeasure_T − δ);
5.      M_T = min(m_T, t = 1,2 ..., T);
6.      PH_T = m_T − M_T;
7.      if(PH_T>λ)
8.          update(U);
```

圖 6 概念飄移偵測虛擬碼

於每個時間點，以每個批次的文件當作測試集S進行F-measure值的預估，依照PH_T衡量該時間點是否出現足夠強烈的變化。當發生概念飄移，則對使用者模型進行更新，透過前述3.2所提的方法，依使用者回饋相關、非相關文件，對使用者模型進行調整。

4. 實驗

4.1 實驗資料集介紹與評估準則

本研究使用Reuters-21578新聞資料集，來自於(Joachims, 1998)[12]，由於有些新聞資料長度過短，經過篩選，使用的單類別資料集如表 1所示。另外Reuters多類別文件則共計227篇。特徵選取的實驗資料集則來自Wikipedia，挑選10篇文章的摘要作為測試資料，以超連結的字作為特徵詞，詳細資料如表 2。

表 1 實驗資料集

類別	篇數	類別	篇數
acq	484	earn	334
cocoa	30	sugar	43
coffee	51	trade	205
crude	174		

表 2 Wikipedia 資料集

文章標題	特徵詞數	文章標題	特徵詞數
CPU	19	Bamboo	17
typhoon	21	cartoon	15
Internet	32	complexity	4
graph theory	8	algorithm	26
k-means clustering	10	Decision tree learning	8

在評估準則方面，本研究之評估準則使用Precision、Recall以及F-measure。

而在分群結果的評估方面，則採用Newman於2004年提出的Modularity[13]，其計算公式如下：

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j)$$

其中A_{ij}為節點i、j的相鄰矩陣元素值；k_i為節點i的degree；$\delta(C_i, C_j$為節點i、節點j是否為同一群的判斷，同群為1，不同群為0；m為全網路連結數。若網路中有社群存在，則該社群內的邊的個數應該較隨機網路的邊的個數大；因此分群過後每一個社群的邊的個數皆大於與其等大的隨機網路邊的個數，則可稱其為一個好的網路分割，反之則為不理想的分割。

4.2 實驗結果與討論

(1) 實驗一：特徵選取的差異

本研究修改[6]的字詞網路建立，以去除NGD值大於1或無限大的連線後，取小至大排序前50%建立字詞網路。為驗證是否有差異，以Wikipedia為資料集(表 2)，並以文章中的超連結作為標準特徵字詞，來比較差異，此實驗將延伸保留的門檻值25%、50%、75%、100%進一步比較。實驗結果如圖 7，在保留前50%時，所納入的特徵字詞Recall值較高，而高於50%，則因為納入的字詞過多，雖然能提升Recall，但也略為降低了Precision值。考量運算效率與保留較多的特徵字詞供後續分群，本研究以50%作為保留連線的門檻。

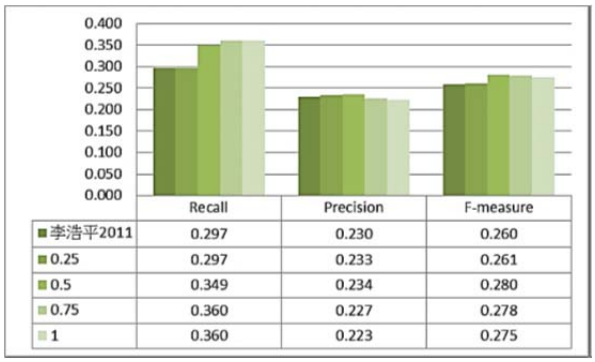

圖 7 字詞網路建立之差異比較

(2) 實驗二：本研究方法的門檻值

首先是β值，分為單篇文件字詞網路的β_{single}與多篇的β_{multi}。在β_{single}，單類別文件部分將以Reuters資料集的每個類別隨機挑選30篇文章，以門檻值0.1~0.9進行分群並以平均Modularity評估；而多類別文件則以同樣方法隨機挑選60篇進行評估，實驗結果如圖 8，以β_{single}=0 .4表現較佳。

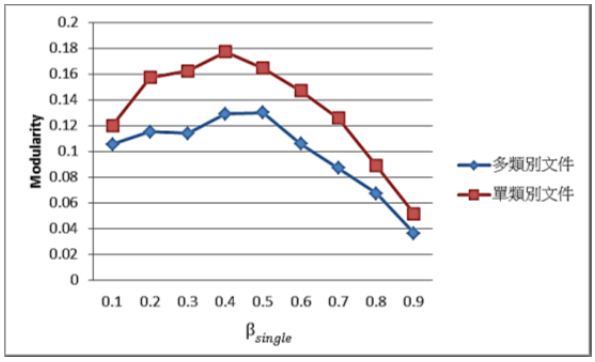

圖8 β_{single}門檻值實驗結果

在β_{multi}則以M個類別，每一類別N篇文件的合併特徵字詞網路進行分群(圖以M-N表示)，例如2-1代表2個類別，每個類別1篇，實驗結果如圖 9與圖 10，可以得知隨著文件數增加，分群效果越不顯著，類別較多，則分群效果較明顯。考量分群效果與執行效率，本研究β_{multi}訂為0.35。

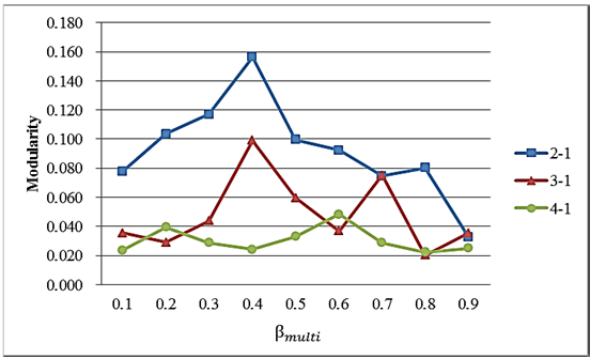

圖9 β_{multi}門檻值實驗：單類別合併

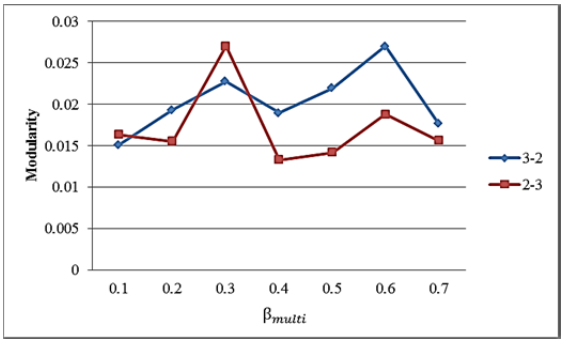

圖 10 β_1門檻值實驗：多類別合併

門檻值α實驗，將於選定資料集的7個類別中，各類別每次隨機挑選5篇作為該類別的訓練文件建立模型，接著在7類別各隨機挑選10篇作為測試文件，訓練文件與測試文件之間不重複。70篇的測試文件以sim(d_t,U)公式排序。分別以γ=0.1、0.25、0.5、0.75、1，反覆進行5次。每次的排序結果中，以相關文件的出現位置計算出該位置的precision與recall值，而實驗結果如圖 11，以γ為0.75、1較佳由於γ=0.75、1結果相當接近，考量運算速度，取γ為0.75；而由F-measure曲線中，可以得知最佳值落在recall=0.7，因此取出各個實驗中，recall=0.7對應之sim(d_t,U)再取平均，得出α=0.525。

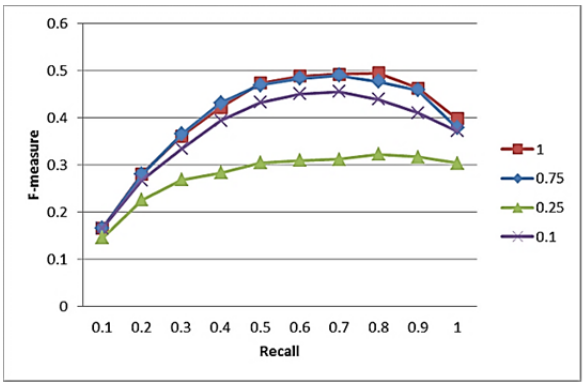

圖 11 α門檻值實驗

(3) 實驗三：找出潛在相關文件的能力評估

為評估參與中間度對單篇文件形成的字詞網路分群，是否可以有效找出使用者可能感興趣的潛在相關文件，以trade類別為例，取5篇作為訓練文件，並於多類別文件中隨機取包含trade類的10篇與不包含trade類的40篇，以sim(d_t,U)排序，考慮γ為0.75、1兩種情況，各反覆進行五次取平均，得到結果如圖 12，可以發現在recall=0.7達到最高值0.668。而γ=0.75效果較佳，可能的原因是因為多類別文件所包含的概念較多，保留75%代表使用者模型的概念可以去除較為不重要的關鍵詞，減少誤判的機會。

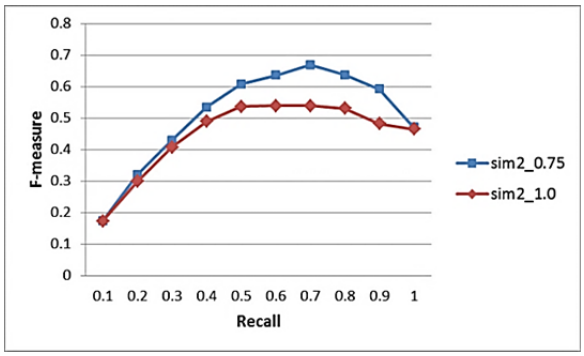

圖 12 找出潛在相關文件實驗結果

(4)實驗四：使用者模型學習能力評估

為測試本研究提出的使用者模型是否能夠適應多主題的學習，對於使用者回饋的相關性與非相關文件能夠對模型有適當的調整，將以一個情境，模擬原本使用者的興趣為主題trade，接著到第一個轉折點，使用者興趣轉變為trade與crude，接著再到第二個轉折點，回到主題trade。

每個學習階段，會有5個評估點。第一個學習階段每個評估點使用者會回饋一篇主題trade以更新使用者模型；而第二個學習階段的每個評估點，使用者則會回饋一篇主題trade，與一篇主題crude的共計兩篇相關文件回饋以更新使用者模型；到了第三階段，每個評估點則會回饋一篇主題trade作為相關文件，一篇主題crude作為非相關文件，來更新使用者模型。

每個評估點，將會以從7個類別各自挑選10篇做為測試文件(與訓練文件不重複)，以先前所訂定的門檻值α、β、γ進行過濾，再以precision、recall、F-measure來評估該點的過濾效能。

依照調整時間點，調整方式分為每次調整(圖示以incremental表示)、偵測後調整(圖示以detect表示)、不調整(圖示以base表示)。偵測後調整的方式將以PH_T偵測概念飄移，以δ = -0.05用以偵測是否有過多的減少，λ = 0.15作為是否更新的判斷門檻值，在time6至time12因PH_T大於0.15，模型將進行更新。而實驗結果如圖 13至圖 15。

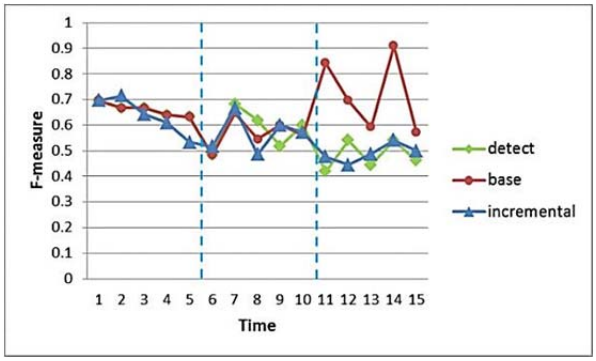

圖 13 學習能力F-measure 評估

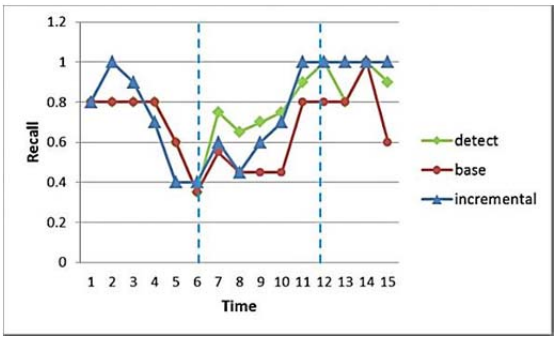

圖 14 學習能力Recall 評估

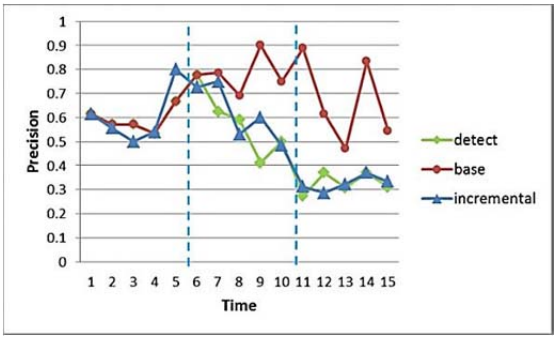

圖 15 學習能力Precision 評估

由圖 13至圖 15可發現，隨著每次更新模型，F-measure隨之下降，會有這樣的現象主要是因為γ=0.75，每次更新將會於各群依照degree排序挑選前75% 來代表該群，更新後會使得使用者模型字數遞減，並且每次的參與中間度分群會去除35%的連線，若文章的篇幅過短，幾次分群下來會讓原本存在於使用者模型的關鍵詞被去除，讓使用者模型概念的字數遞減，而使得誤判率提高，precision下降，但同時也提高了recall值，由於目標概念穩定，若依照此法更新反而會讓使用者模型的過濾效能不穩定。

到了第二個階段(time 6 ~ time 10)，延續著第一階段的現象，precision低於原始模型，但recall值因為有持續加入新的類別文件，因此recall值較原始模型高。

到了最後一個階段(time 11 ~ time 15)，每個評估點除了有一篇trade類別的相關文件，還有crude類別的非相關文件來進行模型更新，然而非相關文件的模型更新需仰賴文件中特徵出現於使用者模型中才能有效更新，因此效果並不顯著，而是緩慢、漸進地提升precision，而recall值則因為第二階段模型的完備，以及持續加入trade類別文件更新而維持高的recall值。

5. 結論與未來研究方向

本研究提出一個以NGD建立的字詞網路為基礎的使用者模型，透過它可以依照使用者對於多個概念的喜好對文件進行過濾，而在喜好發生變化時，也能夠適當的偵測並更新模型。並且透過參與中間度為基礎的分群方法，可以找出文件所包含的主要、次要概念，透過適當的門檻值，能過濾出使用者可能感興趣的潛在相關文件。另一方面，在概念飄移發生時，也能夠透過對於F-measure的預測值，決定是否對於整個模型進行更新的動作。綜合以上論述，本研究的主要貢獻包括：

(1) 建立適用於多主題的使用者模型，容納多概念於一個模型，並依此對文件進行過濾。
(2) 透過使用者模型找出使用者可能感興趣的潛在相關文件。
(3) 偵測多主題的情況下，發生的概念飄移。

然而，目前系統仍具有改進空間。對於未來的研究方向，可以朝以下幾點進行改進：

(1) 改變概念飄移偵測方式：當目標概念越多，則以F-measure之變化也將越不明顯。
(2) 改進使用者模型的建立：參與中間度分群隨字詞網路擴大，其運算效能、分群效果越差。
(3) 建立不同的過濾門檻：可依照各資料集的情況訂定不同門檻達到最佳化。
(4) 有效的模型更新方式：因過濾方法容易受模型字詞多寡變化而影響，當模型改變而過濾門檻應當有適當的改變。

參考文獻

[1] U. Hanani, B. Shapira, and P. Shoval, "Information filtering: Overview of issues, research and systems," User Modeling and User-Adapted Interaction, vol. 11, pp. 203-259, 2001.

[2] A. Tsymbal, "The problem of concept drift: definitions and related work," Computer Science Department, Trinity College Dublin, 2004.

[3] I. Žliobaitė, "Learning under concept drift: an overview," arXiv preprint arXiv:1010.4784, 2010.

[4] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," Information Fusion, vol. 9, pp. 56-68, 2008.

[5] R. Klinkenberg and T. Joachims, "Detecting concept drift with support vector machines," in Proceedings of the Seventeenth International Conference on Machine Learning (ICML), 2000.

[6] 李浩平, "運用NGD建立適用於使用者回饋資訊不足之文件過濾系統," 碩士論文, 國立中央大學, 民國100年.

[7] E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas, "Dealing with concept drift and class imbalance in multi-label stream classification," in Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two, pp. 1583-1588, 2011.

[8] 鄭奕駿, "離線搜尋Wikipedia以縮減NGD運算時間之研究," 碩士論文, 國立中央大學, 民國101年.

[9] M. Girvan and M. E. Newman, "Community structure in social and biological networks," Proceedings of the National Academy of Sciences, vol. 99, pp. 7821-7826, 2002.

[10] U. Brandes, "A faster algorithm for betweenness centrality," Journal of Mathematical Sociology, vol. 25, pp. 163-177, 2001.

[11] E. Page, "Continuous inspection schemes," Biometrika, vol. 41, pp. 100-115, 1954.

[12] T. Joachims, Text categorization with support vector machines: Learning with many relevant features: Springer, 1998.

[13] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical review E, vol. 69, p. 026113, 2004