第五章

資料探勘的技術與應用

Data Mining Technologies and Applications

歐陽彥正
Yen-Jen Oyang
國立台灣大學資訊工程學系教授
Professor, Department of Computer Science and Information Engineering, National Taiwan University

葉建華
Jian-Hua Yeh
國立台灣大學資訊工程學系博士後研究員
Post-Doctoral Researcher, Department of Computer Science and Information Engineering, National Taiwan University

黃賢卿
Shien-Ching Hwang
國立台灣大學資訊工程學系博士後研究員
Post-Doctoral Researcher, Department of Computer Science and Information Engineering, National Taiwan University

壹、資料探勘技術的應用

資料探勘技術在文件分析的應用層面，大致上可分爲詞彙分群（term clustering），文件分類（document classification），以及文件分群（document clustering）等應用議題。所謂的詞彙分群，就是經由文件的分析結果，找出詞彙之間的相關程度，以便將來應用在查詢詞彙的建議，以及同義/相關詞彙的呈現上。舉例來說，當使用者查詢「非典型肺炎」這個詞彙時，搜尋引擎可以自動加入「SARS」這個詞彙的搜尋，這是因爲這二個詞彙在分群的分析上屬於同義字。另外，相關詞的部分可能呈現「和平醫院」、「衛生署」及「世界衛生組織」等與

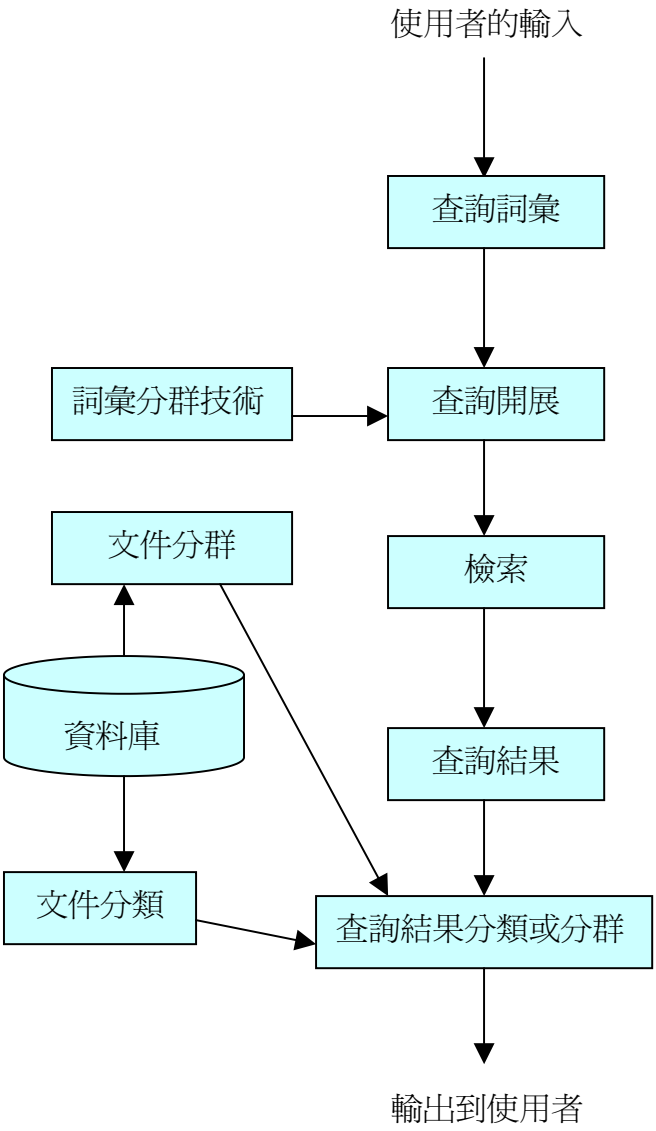
它具有高度相關的詞彙。這些都是搜尋引擎背後利用分群技術事先經由大量資料的分析以找出一些高度相關的詞彙，以提供使用者在查詢時可以得到更完整的資訊，也可以輔助使用者作更進一步查詢的建議。

另一種資料探勘技術是針對文件分類。所謂的文件分類，就是以一個既定的分類架構爲前提，將文件集中的各個文件經分析後給予特定的類別屬性。這樣的技術需要分析文件中的各個詞彙，並藉由各項詞彙的屬性分析給予文件一個總合的判斷。目前有許多的內容提供者都會利用這項技術來提供各種不同的相關資訊服務。舉例來說，某家公司的主管可能會對特定產業的相關文件感興趣，因此便需要這類技術，來將大量文件中相關於某特定產業類別的文件予以過濾，有效地減少資訊量，同時提升必要資訊的品質與重要性。

資料探勘技術不但可以應用在既定的架構上作分類的動作，也可以應用在不特定類別的文件分群上。所謂的文件分群，就是根據使用者的行爲，來針對使用者的需求作更精確的資料過濾。這樣的分析，主要是針對使用者的行爲傾向來做更正確的資料提供。也就是說，文件分群乃是針對一個客製化的概念（customized concept）進行類似文件分類的動作，只是這項動作並沒有預先存在的分類架構爲基礎，因此在資料的處理和分析流程上都略有不同。舉例來說，一個使用者所查詢的詞彙爲「長榮航空」時，他/她可能是對這家航空公司的財務績效有興趣，但也有可能是對這家公司的航班資訊有興趣，因此，如果這位使用者最近的查詢行爲有涉及其他公司的財務資料時，智慧型的搜尋引擎就應該判斷將該航空公司的財物資料給予較高的比重。相反地，如果該使用者最近的查詢行爲比較接近航班資料時，搜尋引擎就應該將航班資料的相關資訊給予較先的排序順位。這些判斷處理，便都是文件分群重要的應用。

圖一爲使用資料探勘技術來完成檢索的整個過程，首先使用者輸入欲查詢的詞彙，系統根據輸入的詞彙在事先準備好的詞彙集中進行搜尋，搜尋完後再利用詞彙分群技術來進行查詢開展的動作，在查詢開展中如前面所述，會將詞彙分群技術找出的同義詞及相關詞一起加入搜尋。接下來系統將這些詞彙進行檢索得到檢索的結果，依檢索的結果進行資料分群或分類，這部分依使用者的需求決定，分群及分類可幫助使用者建立一個階層表，讓使用者可以更清楚了解搜尋結果的

架構。這個檢索的過程適用於大部分的檢索系統，因此也適用於檔案管理局的檢索系統。接下來我們就針對這三項資料探勘技術，作比較技術層面上的介紹。



圖一 檢索系統的流程

貳、詞彙分群技術（Term Clustering）

詞彙分群技術需要以詞彙出現頻率紀錄爲基礎，通常一篇文章中若某個特定的詞彙出現的次數很多，則通常這個詞彙與這篇文章的關連度會愈高，例如一個探討運動的文章中，應該會出現各種球類的詞彙如棒球、籃球、足球等等。在技

術層面上我們會以一個表格來記錄詞彙在文章中的關係性，這種表格我們稱它爲D-T table。通常D-T table 有兩種表示形式，一種是單純紀錄詞彙頻率的方式，另一種則牽涉到Market-basket 資料模型的紀錄方式。下圖是一個單純紀錄詞彙頻率的D-T table：其中*d*1, *d*2, …, *d*7 代表文件的編號，而*t*1, *t*2, …, *t*5 則代表各種不同的詞彙，表格中的數字即爲詞彙出現在文件中的頻率，也就是它出現的次數。這樣的D-T table 該如何建立呢？主要是由以下幾個步驟完成：

	t_1	t_2	t_3	t_4	t_5
d_1	160	354	15	22	74
d_2	42	91	32	143	87
d_3	16	71	167	72	85
d_4	34	56	46	203	92
d_5	36	82	289	51	25
d_6	7	6	225	15	54
d_7	215	392	17	54	121

圖二 單純紀錄詞彙頻率的D-T table

一、去除常用詞（stop words），例如中文的「的」、「因爲」、「所以」，以及英文的「the」、「of」、「the same as」等等，這種詞往往出現次數太多，但又不具特別意義。

二、在英文中辨識詞類變化（word stem），例如「normal 」、「normally 」、「normalization 」、「normalized 」及「normalizing」都擁有相同的詞類變化。

三、計算詞彙間的相似度（similarity），二個詞彙的相似度愈高表示它們在語意上愈相近，例如以圖二的例子來說，我們可以使用一種稱爲cosine similarity 來計算二個詞彙*t*1 與*t*2 的相似度，公式如下：

$$sim(t_1,t_2)=(160\times354+42\times91+16\times71+34\times56+36\times82+7\times6+215\times392)$$
$$/(276.31\times549.73)=150776/151895.9=0.993$$

公式的計算結果將會在1 與1 之間，代表了*t*1 與*t*2 的相似程度。由於文件的長短不一，有時差異頗大，同時各種詞彙的出現頻率也會差異頗大，因此我們可以試著將D-T table 正規化（normalize），藉由正規化的處理，可以將文件間與詞彙間的差異有效地縮小。正規化的計算公式如下：

$$\nu_{ij} = \left(c_{ij} - \frac{(\sum_K c_{kj})(\sum_k ci_k)}{\sum\sum_{k\,h} c_{kh}} \right) \Big/ \left(\frac{(\sum_K c_{kj})(\sum_k ci_k)}{\sum\sum_{k\,h} c_{kh}} \right)^{1/2}$$

其中*c*ij 代表圖二表格中第(*i, j*)項的值，而*v*ij 則代表正規化後的數值。圖三以及圖四說明了正規化計算的加總以及計算過程：

	t 1	t 2	t 3	t 4	t 5	
d 1	160	354	15	22	74	625
d 2	42	91	32	143	87	395
d 3	16	71	167	72	85	411
d 4	34	56	46	203	92	431
d 5	36	82	289	51	25	483
d 6	7	6	225	15	54	307
d 7	215	392	17	54	121	799
	510	1052	791	560	538	3451

圖三 正規化計算：行列的加總

	t 1	t 2	t 3	t 4	t 5
d 1	7.04	11.84	-10.71	-7.89	-2.37
d 2	-2.14	-2.68	-6.15	9.86	3.24
d 3	-5.74	-4.85	7.50	0.65	2.61
d 4	-3.72	-6.58	-5.31	15.91	3.03
d 5	-4.19	-5.38	16.95	-3.09	-5.80
d 6	-5.70	-9.05	18.43	-4.93	0.89
d 7	8.91	9.51	-12.28	-6.64	-0.32

圖四 正規化計算結果

另一種D-T table 則牽涉到Market-basket 資料模型的紀錄方式。一個以Market-basket 資料模型爲基礎的D-T table，其建立過程則有以下幾項步驟：

一、這種D-T table 必須以之前正規化過的D-T table 資料爲基礎。

二、我們可以在正規化過的D-T table 資料上設定一個門檻值（threshold），將大於這個門檻值的欄位設定爲1，否則爲0。

	t_1	t_2	t_3	t_4	t_5
d_1	1	1	0	0	0
d_2	0	0	0	1	1
d_3	0	0	1	0	1
d_4	0	0	0	1	1
d_5	0	0	1	0	0
d_6	0	0	1	0	0
d_7	1	1	0	0	0

圖五 Market-basket D-T table(門檻值=1.645)

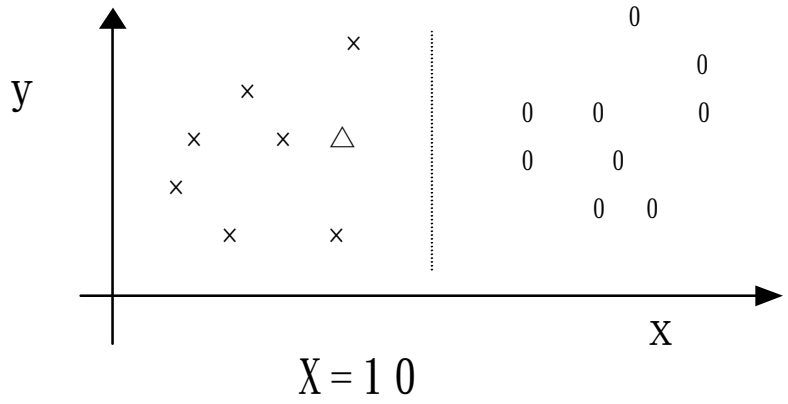
詞彙分群技術可以實際應用在各種類型的查詢開展（query expansion）上，對於查詢結果的重要性排序會有相當程度的助益。此外，詞彙分群技術也可以作爲文件分群以及分類技術的應用基礎。

參、文件分類技術（ Document Classification）

在介紹過詞彙分群的概念與技術之後，我們接著討論文件分類的技術。對文件分類技術而言，之前所提到的詞彙或是詞彙群集（term cluster）都可以視爲是文件中的特徵（feature）或是屬性（attribute）。特徵是一種比詞彙更抽象的觀念，例如運動是一個詞彙，不過他也可以視爲一個特徵，若視爲詞彙時，所代表的就是單單「運動」這個字眼。若是把它視爲特徵，則他可以代表各式各樣的運動如「棒球」、「籃球」或「足球」等等。

在文件分類的技術中，常用的資訊，便是之前文中所提到的Market-basket D-T table。而在文件分類的前處理過程中，有一個重要的步驟，稱之爲特徵選擇（feature selection）。所謂的特徵選擇，就是要根據我們所要進行的分類程序，將其中相關的重要特徵選擇出來，同時排除不相關的特徵。例如假設目前要分類的主題爲一個財經方面的主題，則特徵選擇出來的詞彙會是一些與財經有關的詞彙如「股票」、「投資理財」、「金融保險」等等。換句話說，與分類無關的詞彙或是詞彙集，將會在這個程序中被過濾掉。這樣的過濾/選擇程序有許多好處，因爲不相關的詞彙將會擾亂分類過程中的計算判斷。舉例來說，圖六中的資料分布，若是選定以Y 軸爲特徵，則會導致不正確的分類結果。

顯而易見地，「o」與「x」可以由x=10 這條直線來予以切割分類，如果我們以X 軸爲分類特徵的話，我們將可以正確預測「△」所屬類別。以下步驟說明了特徵選擇的處理過程：



圖六文件分類範例

一、去除常用詞（stop words），例如中文的「的」、「因爲」、「所以」，以及英文的「the」、「of」、「thesame as」等等，這種詞往往出現次數太多，但又不具特別意義。

二、在英文中辨識詞類變化（word stem），例如「normal 」、「normally 」、「normalization 」、「normalized 」及「normalizing」都擁有相同的詞類變化。

三、使用特定的特徵選擇程序來進一步去除與分類不相關的詞彙，通常可以使用chi-square的相依性測試。以下爲實際計算的範例（使用chi-square 測試）：

(一)我們使用以下的Market-basket D-T table 爲計算範例：

		t_1	t_2	t_3
	D_1	1	0	0
Class1	D_2	1	0	1
	D_3	1	0	1
	D_4	1	0	0
==========				
	D_5	0	1	0
Class2	D_6	0	1	0
	D_7	0	1	0
	D_8	1	1	0

(二)對t1 而言，我們將所屬類別整理如下：

	t_1	$\sim t_1$	
c_1	4	0	4
$\sim c_1$	1	3	4
	5	3	8

而其chi-square 測試計算如下：

$$\chi^2=(4-(1/2)\times(5/8)\times 8)^2/(1/2)\times(5/8)\times 8)+(0-(1/2)\times(5/8)\times 8)^2/(1/2)\times(5/8)\times 8)+(1-(1/2)\times(5/8)\times 8)^2/(1/2)\times(5/8)\times 8)+(3-(1/2)\times(5/8)\times 8)^2/(1/2)\times(5/8)\times 8)=4.8>\chi^2_{0.05}=3.841$$

根據chi-square 的計算結果，顯示*t*1 是可以選定的特徵值。

(三)對t2 而言，我們將所屬類別整理如下：

	t_2	$\sim t_2$	
c_1	0	4	4
$\sim c_1$	4	0	4
	4	4	8

而其chi-square 測試計算如下：

$$\chi^2=8$$

根據chi-square 的計算結果，顯示*t*2 是可以

選定的特徵值。

(四)對t3 而言，我們將所屬類別整理如下：

	t_3	$\sim t_3$	
c_1	2	2	4
$\sim c_1$	1	3	4
	3	5	8

而其chi-square 測試計算如下：

$$\chi^2=0.53<\chi^2_{0.05}$$

根據chi-square 的計算結果，顯示*t*3 不可以是選定的特徵值。

除了特徵選擇程序之外，另外還有一個特徵重要性（feature weighting）計算

程序，可以用來幫助更正確的分類處理。所謂的特徵重要性計算，就是要將選定的特徵再根據其重要性予以區分，以利分類程序的使用。特徵重要性主要是依據之前的特徵選擇計算數值爲基礎進行判斷。舉例來說，前例t1 的重要性可以用4.8或是4.8½ (=2.19)來計算，而t2 的重要性則可以用8 或是8½(=2.83)來計算。

此外，在使用特徵選擇的計算結果之外，還有另一種計算特徵重要性的方式，稱之爲「inverse document frequencies」，這種方法我們將稍後介紹。

接著特徵選擇與特徵重要性處理之後，接下來就要正式進行文件分類的程序。我們在此介紹使用cosine 相似度（cosine similarity）的分類計算方式。這種分類計算方式主要是運用先前所計算出具有特徵重要性的D-T table，將其表達成向量形式（vector representation），然後進行計算。在下面的範例中，我們是使用了cosine相似度，配合KNN 演算法來進行分類程序：

一、以下爲D-T table 範例：

	t_1	t_2
D_1	2.19	0
D_2	2.19	0
D_3	2.19	0
D_4	2.19	0
D_5	0	2.83
D_6	0	2.83
D_7	0	2.83
D_8	2.19	2.83

二、接著我們給定Dt = <1 × 2.19, 1 × 2.83>，則

cosine 相似度（我們以sim 代表）爲：

sim<*D*1, *Dt*> = *sim*<*D*2, *Dt* > = *sim*<*D*3, *Dt* >

= *sim*<*D*4, *Dt* > = 0.61

sim<*D*5, *Dt*> = *sim*<*D*6, *Dt* > = *sim*<*D*7, *Dt* >

= 0.79

sim<*D*8, *Dt*> = 1

根據計算結果，我們便會判定*D*t 爲第二類

（*D*5, *D*6, *D*7）。

肆、文件分群技術（Document Clustering）

文件分群與文件分類最主要的不同，除了文件分類擁有既定的分類架構而文件分群沒有之外，文件分類擁有訓練資料集合（trainingset），也是文件分群技術中所無法使用的。這可以想像成文件分類是經由事先訓練學習後，再利用所學習的經驗加以對新的文章做分類。例如當你要學習有關運動方面的分類，可以事先找來一些文章，這些文章事先經由人工標註那些是運動類文章，那些是非運動類文章，在經由這些文章的學習後就可以了解運動類文章的特性。另外一方面，在文件分群，並沒有訓練資料集合的概念，如此我們無法在文件分群技術中使用諸如特徵重要性等等計算。儘管如此， 我們仍然可以使用所謂的「inverse document frequency」（idf）來計算文件中詞彙的重要性。以下便是計算一個詞彙*tk* 的idf 計算公式：

$$\text{idf}(t_k) = \log \frac{\text{文章數}}{\text{包含 } t_k \text{ 的文章數}}$$

我們用以下的D-T table 爲例，計算各個詞彙的idf 值：

	t_1	t_2	t_3	t_4	t_5
D_1	1	0	0	1	0
D_2	0	0	1	0	1
D_3	0	1	0	1	1
D_4	1	1	0	1	0
D_5	0	1	1	0	1
D_6	0	0	1	1	0

- *idf* (*t*1)=0.48
- *idf* (*t*2)= *idf* (*t*3)= *idf* (*t*5)=0.30
- *idf* (*t*4)=0.18

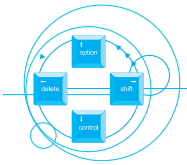
圖七 D-T table 以及所計算的idf 值

因此，我們便可以將各個文件表示爲向量形式：

*D*1 = <0.48, 0, 0, 0.18, 0>

*D*2 = <0, 0, 0.30, 0, 0.30>

*D*3 = <0, 0.30, 0, 0.18, 0.30>



*D*4 = <0.48, 0.30, 0, 0.18, 0>

*D*5 = <0, 0.30, 0.30, 0, 0.30>

*D*6 = <0, 0, 0.30, 0.18, 0>

一旦形成了向量形式，我們就可以運用cosine 相似度來進行文件相似度的計算。根據兩兩文件計算cosine 相似度的結果，我們可以畫成下表：

	D_1	**D_2**	**D_3**	**D_4**	**D_5**	**D_6**
D_1	-	0	0.137	0.862	0	0.180
D_2		-	0.46	0	0.816	0.606
D_3			-	0.447	0.751	0.201
D_4				-	0.291	0.156
D_5					-	0.494
D_6						-

圖八 文件之間的cosine 相似度

有了兩兩文件之間的相似度之後，接著我們就可以使用一些既有的分群演算法來進行文件分群程序。在此我們以一種分群演算法complete-link 爲例， 建構出圖九的結構表（dendrogram）。

根據圖九的結構表，如果我們使用相似度爲0.3 的門檻值，我們將可以得到以下三類：

{*D*1, *D*4}, { *D*2, *D*5, *D*6}, { *D*3}.

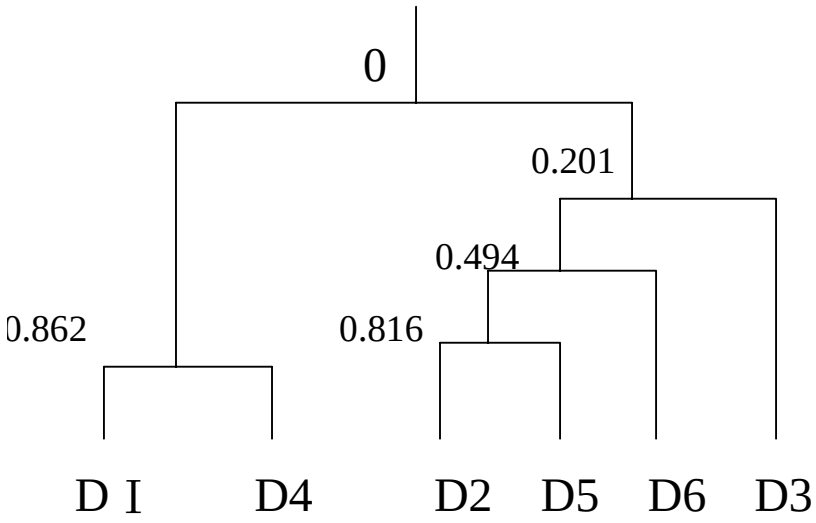
而若我們以0.5 爲門檻值的話，則我們會得到以下四類：

{ *D*1, *D*4}, { *D*2, *D*5}, { *D*6}, { *D*3}.

反之，若我們將門檻值設定爲0.2 的話，則我們僅會得到兩類文件：

{ *D*1, *D*4}, { *D*2, *D*5, *D*6, *D*3}.

就文件分群而言，主要的部分是計算相似度與使用分群演算法。

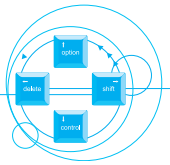


圖九 complete-link 演算法建構出的結構表

伍、結語

觀察目前資訊數位化的程度，我們可以知道數位化文件的累積速度，是相當驚人的。也正因爲如此，更需要相關的資料探勘技術來予以輔助，才可以有效縮減使用者所必須面對的資訊數量，同時也可以加速資訊處理的速度。本文已針對文件集的各項探勘技術，做一概念性的介紹。在應用層面上，不論是使用者行爲分析或是文件集的統計分析，都是資料探勘技術所欲強調與發展的議題。資料探勘的研究仍不斷進行中，相信將來可以發展出更有效的技術，來協助人們搜尋、組織及整理資料。

【原刊載於檔案管理局出版之檔案季刊（九十二年六月）第二卷第二期】



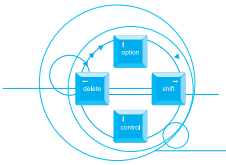
參考資料：

[1] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2000.

[2] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.

[3] A. K. Jain, M. N. Murty, P. J. Flynn, "Data Clustering: A Review," ACM Computing Surveys,vol. 31, no. 3, pp. 264-323, 1999.

[4] Y. J. Oyang, C. Y. Chen, and T. W. Yang, "A Study on the Hierarchical Data Clustering Algorithm Based on Gravity Theory," Proceedings of 5th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 350-361, 2001.


option
delete
shift
command