探勘群體成對比較資料之最大共識與衝突

鄭麗珍　　賴旻廷
東吳大學資訊管理所　　東吳大學資訊管理所
lijencheng@csim.scu.edu.tw　allex010328@hotmail.com

摘要

群體共識排序的問題已漸漸被應用在商業領域上，如決策支援系統、團體推薦系統等。群體排序的概念是利用個人的項目喜好資訊去整理排序出全體中大家的項目喜好排序，藉此了解到在群體中大家對於項目群的喜好順序，並應用於產品開發或行銷上。以往的群體共識排序僅利用使用者填答資訊進行排序，這樣會錯失其中所隱含的資料，因此本研究利用使用者序列延伸圖將使用者的成對比較資料進行擴張，讓隱含在資料中的資訊可以顯現出來，並利用本研究所提出的演算法，進而達到更準確的共識序列排序。本研究因應不同的資料共識率以及支持度，讓決策者可以更方便清晰的了解到客戶群的共識序列，而做出更好的因應策略。

關鍵詞：資料探勘、群體決策、最大共識序列、群體排序、成對比較

1. 緒論

許多考量面上，群體共識序列在商業上不管是公司內部的研發部門、面對市場的行銷團隊或者是直接接觸終端使用者的販售商店都需要獲得並了解客戶群中的共識序列(Consensus sequence)。因為客戶群中的共識序列(Consensus sequence)代表著客戶群裡最多人喜歡的商品排列，決策者可以利用這份資訊了解到整個群體內大家對於產品的共同意見。因此如何找出團體中最大的喜好序列？利用個人對於商品的喜好資料以及群體中所有成員的喜好序列去歸納、計算此群體中的最大共識序列(Consensus sequence)，讓公司的行銷團隊、銷售據點如何應用在行銷、商品販售最大的獲利資訊成為重要的問題。

由於科技的演進以及網路日益發達，因此在網路討論商品的趨勢越來越熱烈，許多使用者可以方便的寫出使用者自己對於物品所喜歡的商品比較。而同儕之間的影響往往也具有非常大的影響力，也因為網路硬體設備的快速發展，所以同儕已經不是只是局限於身邊的朋友，因此不管是使用者身邊亦或是網路上的群組使用者對於產品的共同意識也慢慢的受到大家的重視，因此群體中共識的問題不僅牽涉到個人的喜好優劣，也存在著這整個群體間的共同喜好序列。但是產品項目每天都會改變，產品的更新也比以往更加快速，而資訊的流動傳播日益增快，所以商品的資訊瞬息萬變，因此利用共識序列可以讓消費者清楚知道大家對於市場上產品的看法，簡單的了解現階段其他消費者對於這些產品的喜歡順序。

近年來，如何解決群體中的排序問題已漸漸成為一個重要的問題，此問題也漸漸的被廣泛應用於許多應用，如決策支援系統（(Cook, Golany et al. 2005)，(Fernandez and Olmedo 2005)）、團體推薦系統(Chen and Cheng 2008)、機器學習(Fagin, Kumar et al. 2003)以及網頁搜尋策略（(Beg and Ahmad 2003)；(Cohen 1999)）等。以往的研究，用來向決策者表達個人喜好的輸入格式有三種格式，第一種是分別為每一樣物品填入自己所喜愛的評等（rating），例如對於物品 A、物品 B 以及物品 C，消費者分別填入 5 分、3 分以及 6 分，如此一來決策者可以了解到消費者最喜歡物品 C，相對的物品 B 比較不喜歡。以往的商品喜好評分等級都是採用評等（rating），但是這種方式消費者無法依照自己的喜好程度給予評分，因為若是消費者給予兩個物品相同的評分等級，數字上的意義會代表兩者是相同的，但這並不意

味著兩者之間沒有喜好程度的差異，所以這種表達方式容易出現失真的現象。

利用成對比較(pairwise)的填答方式讓消費者可以清楚的表達出自己所喜好的商品種類是比較可行的方式。在所有的物品中，將物品兩兩比較，例如消費者寫出物品 A 優於物品 C、物品 C 優於物品 B，如此一來決策者可以清楚的知道消費者在物品 A 與物品 C 中會挑選物品 A。

成對比較(pairwise)允許消費者個人可以擁有多筆成對資料(pair)的比較資料，也可以擁有相同的成對比較(pairwise)的比較資料，或是在資料群中可以存在互相衝突成對比較(conflict pairwise comparison)資料。過去的研究都要求消費者一次做完所有商品評分標準是有困難的，因為商品的種類繁多加上消費者無法清楚的在一個時間點下分辨出所有的商品優劣，所以是有困難性的。因此本研究允許消費者可以在一段時間內有數次的機會反覆填寫，依照自己實際的親身感覺去寫出自己的喜愛比較是比較人性化的。

本研究利用使用者序列延伸圖將使用者原本的成對比較資料進行連結擴張，進一步利用路徑圖將隱含在原本成對比較資料中的資訊找尋出來，增加其資料完整性。並且利用每個消費者所提供的商品喜好資訊去加以歸納推算出一段時間內的群體內最大的共識序列，利用消費者個人的資料權重以及全體的資料權重去衡量並找出全體內最大的共識序列，用以代表這群體中大家的喜好序列，其中互相衝突的成對比較也是重點之一，因此衝突成對比較(conflict pairwise comparison)資料也需要深入探討，並列入最大共識的評估方法中。

本研究在設定的門檻值方面，在 Chen and Cheng 的研究中是利用認同支持度大於最小認同支持度以及衝突支持度小於最大衝突支持度這兩個門檻來做篩選資料的條件。但這會造成資料失真的問題，在之前的研究中當認同的比例超過門檻，且只要衝突比例也高於門檻值就會被刪除。但當使用者意見完全充分表達，例如像投票率很高的情形下，若是有 65%都認同支持某個議題，而反對的意見雖然高過預設門檻值 25%，此時因為認同與衝突有一段不小的差距，因此本研究提出新的篩選方法。認為只要認同比例與反對的比例高過於最小認同的門檻值才是共識資料，因此會比以往的研究多思考了正反向差距的問題。所謂的共識，是要考量到贊同的人數與反對人數之間差距夠大，這樣的才是真正的共識。

本研究提出的方法之貢獻如下：

(1) 允許可以在一段時間之內，分次填寫喜好成對比較資料，可以是部分資訊也可以是完整資訊。
(2) 利用使用者序列延伸圖，將使用者原有資料進行擴張，並找出隱含其中的有用資訊，增加判斷正確性。
(3) 考慮群體內使用者的衝突資料進行探勘，找出模糊意見即認同與反對均等的特殊樣式。這樣的資料很有趣，表達出喜好與反對的勢力是一樣的，需要繼續努力協商取得更大的共識。
(4) 利用贊同的人數與反對人數之間差距夠大，這樣的才是真正的共識，讓當使用者意見充分表達時，亦即完美的完整資訊時，共識資料不會因超過最大衝突支持度而被忽略。

本研究利用使用者序列延伸圖搭配探勘的演算法找出群體中的共識序列，並利用不同的資料共識率、支持度可以應用於不同的情境環境下。決策者可以依照不同的環境，利用不同的資料共識率、支持度進行共識序列探勘，可以較為清楚的了解到消費者的喜好序列，進而提供更有用的資訊給公司做為策略之用，使公司的策略更能有效的了解群體意見。

本文總共分為六章，其架構如下所述，第二章文獻探討，描述回顧以往文獻所做研究；第三章問題定義，定義本研究所使用的方法問題定義；第四章演算法與流程架構，包含本研究所提出的演算法及流程步驟；的五章實驗結果，分析並討論本研究所成線隻實驗結果並加以討論；第六章為結論，針對本研究結果統整結論。

2. 文獻探討

2.1. 群體排序問題

在群體決策的領域中，經常利用排序問題來解決決議或決策上的方式。所以群體排序問題也稱做為排序聚合問題，其觀念及架構級為將群體內各個成員的個別意見加以統計總合成為群體中的共同意見，用來代表群體最後的共識結果。在過去的研究中，傳統的群體排序問題可以利用三個面向來分類：顧客提供完整的喜好物品資料、希望輸出的格式以及用來表達顧客喜好的資料輸出呈現方式。

關於群體排序問題有三種輸入格式，權重模式、成對比較模式以及排序名單模式。根據輸出結果，以前的方法可分為兩大類：完全排序或是部份排序。每個方法都有本身的優勢，並已成功的應用在許多方面。以往大部分的方法都是密切關注如何減少全部輸入資料之間的分歧排名，最後獲得總排名名單，代表所取得的共識(Chen and Cheng 2009)。事實上，用戶的共識可能有不和諧的部分容易被忽略，這可能是一個缺點，因為他們可能成為另一種不同類別的共識。

2.2. 成對比較

在群體排序的方法中，有學者致力於成對比較這個方法，利用填答者對於物品的一對一比較，使用這種高低的喜好程度去表現出物品之間的優劣。只有比較分數容易出錯，因為不同的評論者對於一件事物或是不同事物的評論角度會有所不同，所以無法簡單的利用分數去分辨出所有事物的優劣。

在 Jacob Baskin(2008)的論文研究中有提到權重比較在群體排序中的缺點以及缺失，例如群體中要如何決定一個物件？或是哪一個物件比較好，通常需要成員共同決定出哪個物件比較適合。但是這個方法要如何選擇並計算？通常所使用的方法就是將所有成員的優劣評分總和平均，利用平均出來的數字代表群體的共同意見。但是利用分數來評斷一個事物會有需多潛在的問題，例如在滿分 10 分的情況下，所有人的平均分數總和平均為 7 分，但是平均出來的結果 7 分代表什麼意思？高或低沒人敢斷定，因為每個評論者評論一件事物的態度不一樣，所以某些人覺得 6 分太高，從未給過如此的分數；也可能有評論家認為 6 分是屬於不及格的分數等級。所以只利用分數會有一些潛在因素，導致物品的得分無法代表評論者真正的意思(Baskin 2008)。

因此利用成對比較可以解決上述的疑慮,若是某評論者喜歡給比較的事物高分,所以在事物 1 以及事物 2 中，事物 1 的分數比事物 2 高級代表他比較喜歡事物 1；換而言之，在喜歡給低分的評論者的狀況下意味著只需比較兩兩事物之間的分數，就可知道事物之間的彼此優劣情形。

而在填答的輸入資料方面，可以分為完整資料比較跟不完整資料比較，顧名思義完整的資料比較是在所有的物品之間皆有兩兩對應的比較關係；而另一方面，不完整的資料比較就由填答者自行選取所喜歡或所知道的部分寫出高低優劣，以下我們將分為這兩的部分簡述之。

2.3. 最大共識序列

大部分以往的方法，無論所輸入的資料格式為何，都希望可以升成一個較為完整的排序列表。如上面所述，若沒有協商一致的排序清單，會迫使所生成的排序清單會幣多於利。為了彌補這個缺點，我們提出了一個新的方法，利用協商一致的清單列表去生成一個最大化的同意看法。主要的概念來自於「共識決策理論」(consensus decision-making theory)，強調需要參與者的同意所生成的過程，它不僅探討了使用者中同意的多數，但也解決或減輕了反對意見的少數派。為了支持這一決策過程中，我們必須找出最大協商一致的清單，從用戶的排名數據，並確定衝突的辦法需要進一步協商。因此，該演算法在

我們近期的工作目的是要產生這兩種類型的結果。雖然我們可以利用協商一致的決策支持取得最大共識序列，而新方法中主要的弱點是需要要求使用者輸入我們所需要的要求，以至於可以生成排序序列。輸入的格式需要許多不同的相異物件是非常困難的，因此我們尋求放寬規定，使用者可以更靈活以及動態的方式提供個人的喜好資料。

3. 問題定義

本研究允許每個使用者可以輸入多筆成對比較(pairwise comparison)的資料，由於是多次重複的輸入喜好資料，所以在同一位使用者輸入的資料中有些成對比較資料會多次出現，群體之間也可能有衝突的成對比較資料存在。本研究定義 U 代表所有使用者的集合，$U = \{ u_1, u_2, u_3, \ldots, u_n \}$；項目集合 I 是由所有用來比較的項目所組成，$I = \{ i_1, i_2, i_3, \ldots, i_n \}$。每一個使用者都可以利用成對比較的方式表達自己對於兩個物件的喜好比較，$p = \{ i_r \oplus i_s \}$ $r \neq s$ 其中項目 $i_r, i_s \in I$ 且 比較符號(comparator) $\oplus \in \{>, \approx\}$. 其中 $p = \{ i_r > i_s \}$代表使用者喜歡項目 i_r 勝過 i_s，$p = \{ i_r \approx i_s \}$代表使用者對於項目 i_r, i_r 的喜歡程度沒有差別。

每個使用者可以在不同時間輸入多筆成對比較的資料，使用者 i 的使用者序列為一成對比較資料集其表達為 $S_i = \{ p_{i,1}, p_{i,2}, \ldots p_{i,j} \ldots \}$。舉例而言， $S_1 = \{ p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4} \} = \{ \{ i_1 > i_3 \}, \{ i_3 > i_5 \}, \{ i_5 \approx i_2 \}, \{ i_4 > i_2 \} \}$，代表意思為使用者 u_1 所填寫的成對比較資料有四筆，分別為 $p_{1,1} = \{ i_1 > i_3 \}$, $p_{1,2} = \{ i_3 > i_5 \}$, $p_{1,3} = \{ i_5 \approx i_2 \}$, $p_{1,4} = \{ i_4 > i_2 \}$。

使用者 u_i 的使用者序列 $S_i = \{ p_{i,1}, p_{i,2}, \ldots p_{i,j} \ldots \}$為一個沒有衝突存在的個人成對比較集合。因此我們將利用遞移性將個人所填答的成對比較進行延伸，我們將利用以下兩個步驟將原本的成對比較連結成一張使用者個人的資料圖型，利用下列步驟將使用者個人序列 S_i 建立成為有向圖：

(1). 將使用者序列 S_i 中的項目 i_r 放入有向圖中

(2). 若 $i_r > i_s$ 這個關係是存在於使用者序列 S_i 中，我們就曾加一條有向實線 (i_r, i_s)到圖形中，意味著連接 i_r 以及 i_s 並且是 $i_r > i_s$。

(3). 若 $i_r \approx i_s$ 這個關係是存在於使用者序列 S_i 中，我們就曾加一條有向虛線 (i_r, i_s)到圖形中，意味著連接 i_r 以及 i_s 並且是 $i_r > i_s$。

在步驟一建立好使用者個人的序列圖型後，我們將會把圖型拆解成一個個的成對比較，而成對比較項目之間之間的關係會利用圖形的路徑做為判斷。例如，我們在 i_r 與 i_s 之間若是皆存在著虛線，我們就可以判斷 i_r 與 i_s 是屬於 $i_r \approx i_s$；另一方面，若是在 i_x 與 i_y 之間也有路徑存在，且之間有某些實線連接著，我們即可以判斷兩個項目之間的關係為“ > ”。若兩者之間沒有線存在，我們即無法判斷其之間的關係。

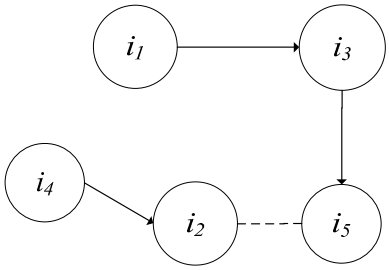

圖 1 使用者序列延伸圖

我們利用圖 1 為例子說明使用者序列延伸的做法。我們將使用者序列 $S_1 = \{ p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4} \} = \{ \{ i_1 > i_3 \}, \{ i_3 > i_5 \}, \{ i_5 \approx i_2 \}, \{ i_4 > i_2 \} \}$利用步驟一將所有的項目以及彼此之間的關係加進圖型中，因此我們可以得到圖 1。其中 $i_1 \sim i_5$ 為使用者所提到的項目符號，項目之間的實線代表兩個項目之間是屬於大於的關係，如 $i_1 > i_3$。然後，兩個項目之間若是虛線，其代表的意義為等於，如 $i_3 = i_5$。

然後我們可以利用步驟二將這個圖型進行拆解，將兩個項目之間是有路徑連接的圖型進行關係辨別，以達到我們講使用者序列延伸的效果。如原本四個成對比較

在經由延伸後，可以得到 $ES_1 = \{ \{ i_1 > i_3\}, \{ i_3 > i_5 \}, \{ i_5 \approx i_2\}, \{ i_4 > i_2\}, \{i_1 > i_5\}, \{i_1 > i_2\},\{ i_3 > i_2 \},\{ i_4 > i_5\}\}$成為七個成對比較的延伸使用者序列。

我們將利用一個關係表格將比對任兩成對比較 p_1 與 p_2 彼此之間的關係。若成員項目都一樣，則這兩個成對比較可以用一個關係函數來表示 $Rel(p_1, p_2)$。這兩個成對比較之關係有兩種：認同關係(**comply**)與衝突(**conflict**)。認同關係是指兩成對比較的成員項目的比較關係是一樣；若兩成對比較的成員項目比較關係是不一樣稱之為衝突。這關係函數用表 1 作說明。

表 1 兩個成對比較資料之間的關係

$Rel(p_1, p_2)$	$p_1 = \{i_r > i_s\}$	$p_1 = \{i_r \approx i_s\}$
$p_2 = \{i_r > i_s\}$	認同關係	衝突
$p_2 = \{i_r \approx i_s\}$	衝突	認同關係
$p_2 = \{ i_s > i_r \}$	衝突	衝突

Definition：認同成對比較(comply pairwise comparison)

有一成對比較 p 與某個使用者序列 S_i 具有認同關係，是指成對比較 p 之成員項目在使用者序列 S_i 中也有出現，且與所對應之成對比較 p'的關係函數為認同關係，亦即 $Rel(p, p')$為認同關係。

例如：

有一成對比較 $p = \{i_1 > i_3\}$，另有一個使用者序列 $S_1 = \{\{ i_1 > i_3\}, \{ i_3 > i_5 \}, \{ i_5 > i_2\}, \{ i_4 > i_2\}\}$，因為在成對比較 p 出現的項目$\{i_1, i_3\}$都在使用者序列 S_1 出現，且有一成對比較 $p' = \{i_1 > i_3\}$與之對應，且關係函數 $Rel(p, p')$為認同關係，故稱成對比較 p 與使用者序列 S_1 具有認同關係。

Definition：衝突成對比較(conflict pairwise comparison)

有一成對比較 p 與某個使用者序列 S_i 具有衝突關係，是指成對比較 p 出現的項目在使用者序列 S_i 中出現，且與所對應之成對比較 p'的關係函數為衝突關係，亦即 $Rel(p, p')$為衝突關係(**conflict**)。

例如：

有一成對比較 $p = \{i_1 > i_3\}$，另有一個使用者序列 $S_2 = \{\{i_3 > i_1\}, \{ i_3 > i_5 \}\}$，因為所有在成對比較 p 出現的項目$\{i_1, i_3\}$都在使用者序列 S_1 出現，且當以相同順序的項目$\{i_1 > i_3\}$出現在使用者序列 S_2 時，且有一成對比較 $p' = \{i_3 > i_1\}$與之對應，且關係函數 $Rel(p, p')$為衝突關係，故稱成對比較 p 與使用者序列 S_2 衝突。

Definition：序列 (sequence)

序列(sequence) α 代表一群不同的項目所組成的順序排列的清單。項目(item)個數代表序列的長度；若是長度為 k 的序列稱為 k 序列. 一個 k 序列可以表示為 $\alpha = \{i_{a_1} \oplus i_{a_2} \oplus ... \oplus i_{a_k}\}$ 其中 $i_{a_k} \in I, 1 \le a_k \le n$ 且比較符號(comparator) $\oplus \in \{>, \approx\}$。注意遇到連續幾個項目的比較符號為"≈", 則項目要按照字母順序排列。舉例, $\alpha = \{i_1 > i_3 \ge i_2 \approx i_5\}$即為 4-序列。

Definition：認同序列 (comply sequence)

有一序列α與某個使用者延伸序列 ES_i 具有認同關係，必須滿足下列條件：

(1). 所有在序列α 出現的項目都必須在使用者序列 S_i 中出現，且序列α 出現的項目個數要小於使用者延伸序列 ES_i 出現的項目個數 $|ES_i| > |\alpha|$

(2). 在序列 α 中，依序選取兩兩項目的成對比較的資料，要與使用者延伸序列 ES_i 中對應到的成對比較是認同關係

例如：有一序列 $\alpha = \{i_1 > i_3 > i_5\}$，另有一個使用者延伸序列 $ES_1 = \{\{i_1 > i_3\}, \{ i_3 > i_5 \}, \{ i_5 > i_2\}, \{ i_4 > i_2\}\}$，因為所有在序列 α 出現的項目$\{i_1, i_3, i_5\}$都在使用者延伸序列 ES_i 出現，且序列 α 依序選取兩兩項目的成對比較為$\{i_1 > i_3\}, \{ i_3 > i_5 \}$與使用者延伸序列 ES_i 所對應之成對比較的符號都符合，故稱序列 α 為使用者延伸序列 ES_i 之認同序列。

Definition：使用者個人認同序列支持度 (user's complying support)

$|ES_i|$為使用者延伸序列的成對比較的個數，而序列 α 的使用者個人認同支持度 $usup_i(\alpha)$定義如下：

$usup_i(\alpha) = \frac{|\{p_{i,j} | p_{i,j} \epsilon ES_i \cap \alpha \text{ is complied with } p_{i,j}\}|}{|ES_i|}$ (1)

舉例而言，若是使用者 i 的使用延伸者序列 S_i 有 9 個成對比較資料，所以 $|ES_i| = 9$。其中有 2 個成對比較資料與序列 α 是認同關係，因此這個序列 α 的使用者個人認同支持度為 2/9。

Definition：認同序列支持度 (Comply support)

假設資料庫有 m 個使用者，則認同序列 α 的整體支持度定義如下：

$$cmp\_sup(\alpha) = \sum_{i=1,m} usup_i(\alpha)/m \quad (2)$$

當序列 α 滿足 $cmp\_sup(\alpha) \geq cmp\_minsup$ 稱為一認同序列，其中 cmp_minsup 為一使用者自訂門檻。

舉例而言，$p_1=\{i_1 > i_2\}$對於使用者 1、使用者 2 以及使用者 3 的個人權重分別為 0.17、0.8、0.33。因此，我們即可以利用我們的定義算出針對 $p_1=\{i_1 > i_2\}$的群體權重為三人總和平均 0.433。

Definition：使用者個人衝突序列支持度 (user's conflict support)

$|ES_i|$為使用者延伸序列的成對比較的個數，而序列β 的使用者個人衝突支持度 $cf_usup_i(\beta)$定義如下：

$$cf\_usup_i(\beta) = \frac{|\{p_{i,j} | p_{i,j} \epsilon ES_i \cap \alpha \text{ is conflict with } p_{i,j}\}|}{|ES_i|} \quad (3)$$

舉例而言，若是使用者 i 的使用者延伸序列 S_i 有 9 個成對比較資料，所以 $|ES_i| = 9$，其中有 3 個成對比較資料與序列β 是衝突關係，因此序列β 的使用者個人衝突支持度為 3/9。

舉例而言，若是使用者 i 的使用者序列擁有 9 個成對比較資料，其中有 2 個成對比較資料為 $p_1=\{i_1 > i_2\}$和 $p_2=\{i_1 > i_2\}$。所以 $p=\{i_2 > i_1\}$和使用者個人資料有這 2 個序列衝突關係，因此這一個使用者個人衝突支持度為 2/9=2/9。

Definition：衝突序列支持度 (conflict support)

假設資料庫有 m 個使用者，則衝突序列β 的整體支持度定義如下：

$$cmp\_sup(\beta) = \sum_{i=1,m} cf\_usupi(\beta)/m \quad (4)$$

Definition ： 共識序列 (Consensus sequence)

當一個認同序列 cs 滿足下列條件，稱為一個共識序列

$(cmp\_sup(cs) - cf\_sup(cs)) \geq minsup$

$minsup$ 為使用者自訂門檻。

我們利用 $cmp\_sup(cs)$ - $cf_sup(cs)$的差距代表共識強度。此門檻可以解決在群體意見充分表達(大部分使用者皆有表達其意見)的情形下，序列不會因為其負面機率跨過門檻而被移除掉。例如在高投票率的情形下，某序列的 $cmp_sup(cs)$為 65%、$cf_sup(cs)$為 20%，在以往的文獻做法中，此序列會因衝突性過於門檻而被移除，但其認同與衝突的差距還是存在著不小的差距，因此本研究利用 $cmp\_sup(cs)$ - $cf_sup(cs)$差距來代表此序列的重要性。

Definition ： 模糊資料對 *(Ambiguous pair)*

除了共識序列之外，我們定義了另外資料型態，叫做模糊資料對。若有一成對比較 p 他的認同序列支持度與最大衝突序列支持度相差不遠，定義如下：

$| cmp\_sup(cs) - cf\_sup(cs) | < \varepsilon$ 且 $cmp\_sup(\alpha) * cf\_sup(\alpha) \neq 0$

ε 是一個自訂門檻。

4. 演算法與流程架構

在這個階段中，我們利用我們所提出的演算法找出群體中的最大共識序列，並考慮其個人的資料權重以及全體中的認同意見權重和衝突意見權重，進而找出全體共識序列。

定理一：

所有共識序列的子序列集合都必須為共識序列。

定理一說明在所找出的共識序列中，其所有的子集合皆須為共識序列，符合訂理才為共識序列。舉例說明，若{A>B>C}為共識序列，其子集合{A>B}、{B>C}以及{A>C}也皆須為共識序列。

接下來，我們將利用一段演算法來表

達對於最大共識序列探勘的想法做呈現，演算法的說明以及步驟如下：

演算法：

步驟一　利用 *genES*(I)延伸使用者的成對比較資料

步驟二　使用 *genL2*(C_2)產生 2-大型共識資料項目集的 L_2

步驟三　呼叫 *ambiguous*(L_2)，找尋模糊資料對

步驟四　重覆產生 k-大型共識資料項目集的 L_k，

for(k = 3; $L_{k-1} \neq \emptyset$;k++)do

C_k= *genLk*(L_{k-1})

L_k = { $cs \in C_k$ | *(cmp_sup(cs) - cf_sup(cs)* ≥ *minsup*

}

演算法步驟詳細說明如下：

步驟一：

利用 *genES*(I)將資料從輸入的文件利用 2 個步驟形成使用者喜好圖型。此步驟中，首先利用實線以及虛線將使用者的成對比較資料進行畫圖，並一一的將所有的項目連結而成一個有向路徑圖。然後，將圖型中項目兩兩抓出，並將項目間有路徑的關係找出，以達到找出使用者成對比較中隱含的成對比較資料，並利用使用者延伸序列(ES)進行共識序列探勘。

步驟二：

第二個步驟是利用 *genL2*(C_2)。首先，C_2 是由步驟一的程是利用使用者的資料而來，利用其使用者的成對比較資料包括比較的元素，以及元素間的運算元。而在 C_2 的資料庫中若是成對比較資料符合 *(cmp_sup(cs) - cf_sup(cs))* ≥ *minsup*，我們就將此一成對比較資料放進 L_2 中。

步驟三：

進行完前兩個步驟後，我們將得到群體中所有使用者的成對比較資料中，較具重要性的比較資料存放於 L_2 中。此時我們利用 ambiguous(L_2)去找尋群體中是否有存在著無法比較出成對比較資料中兩個元素的重要性，也就是我們所稱的模糊資料對*(Ambiguous pair)*。*利用* | *cmp_sup*(*p*) - *cf_sup*(*p*) | < ε 且 *cmp_sup*(*p*)* *cf_sup*(*p*) ≠ *0* 來找出符合此條件的成對比較，稱之為模糊資料對*(Ambiguous pair)*。

步驟四：

利用定理一，我們知道 C_k 是由 L_{k-1} 產生合併而來。因此我們利用 genLk(L_{k-1})來產生 C_k。等到我們獲取 C_k 時，再利用 (*cmp_sup*(*cs*) - *cf_sup*(*cs*)) ≥ *minsup* 來檢驗 C_k 中的候選序列，然後將符合條件的候選序列放進大型序列 L_k 中。這個第四步驟會一直進行反覆計算，直到沒有輸入值進入 genLk(L_{k-1})中；或是沒有候選序列符合條件，造成大型序列為空集合，此時程式出現終止條件，停止運算。

步驟五：

在前面步驟中所找出來的最大共識序列，須將其子集合出現在共識序列中的部分刪除，只留下最長的共識序列部份。

我們利用圖 2 來表示整體的共識序列探勘的演算法處理流程。一開始我們將使用者對於物品間進行評比，所提供的資料為成對比較的格式，例如 User1~User5 所提供的成對比較樣本。然後，我們將依據每個使用者所給的成對比較個數進行個人評比權重配置，以 User1 為例，總共提供 10 個成對比較的資料，因此他所提供的每一份資料配重為 1/10，如此一來可以正規化每個使用者所給予的意見，依序將所有使用者的資料進行配重以利後續步驟。

我們將群體中的個人權重加總，成為群體權重，在 C_2 的部份即為我們加總個人權重並加以平均而成的群體權重。例如 A>B 而言，User1~User5 分別為 0.2、0、0.125、0.125 還有 0.25，因此 A>B 的群體權重為 0.7/5 = 0.14。並且依序將所有的成對比較利用此方法進行計算並記錄而成 C_2 資料。

然後利用門檻值進行篩選，並且考慮其符合支持度以及衝突支持度如例子中是

以門檻值 1/20 進行篩選，從 C_2 中找出符合門檻值的資料放入 L_2 中，成為大型共識序列。並且在此階段，我們利用認同支持度和衝突支持度之間差異太過於小的成對較資料取出，此為本研究定義中的資料模糊對，例子中 B>F、F>B 即為資料模糊對。

接下來將 L_2 中的資料進行合併，將兩個成對比較資料利用相同的項目元素部份進行合併。圖 2 的例子中 A>B 以及 B>D 合併成為 A>B>D，即是利用中間相同的元素部份"B"進行合併，其他項目也是利用這種方法依序將所有成對比較進行合併。合併後，項目的權重配置我們採取合併前的兩個成對比較資料的機率值最小值為合併後機率值，如 A>B 權重為 7/50、B>D 為 2/21，因此 A>B>D 為 2/21，並檢查其 A>D 的衝突性。然後依序將所有合併後的序列進行權重計算，並放入 C_3 中。

在 C_3 中我們一樣利用最小支持度當作門檻值進行篩選而成為 L_3，因此我們找出 L_3={A>B>D，C>A>E，C>A>B，F>A>B}等共識序列。接下來，我們將 L_3 進行合併，一樣利用中間相同的項目進行配對合併。例如：C>A>B 與 A>B>D 兩個序列中，有著 A>B 這部份相同，因此我們可以將 C>A>B 與 A>B>D 合併成為 C>A>B>D。而權重部份，我們依樣採用最小值並進行衝突性檢查，例如 C>A>B 為 1/20，而 A>B>D 為 2/21，因此我們除了取最小值 1/20 外還會檢查資料中是否有與 C>D 為衝突的資料存在，而決定其權重。將所有的序列計算完全重，並放入 C_4 資料中。

在 C_4 中，我們也利用最小支持度進行資料篩選，而最後得到 L_4 為 C>A>B>D 與 F>A>B>D 兩個共識序列。然後，因為資料無法再度進行合併，所以程式即會執行到這邊，並傳回所找到的共識序列。此例子的最大共識序列為 C>A>B>D 與 F>A>B>D 兩個。

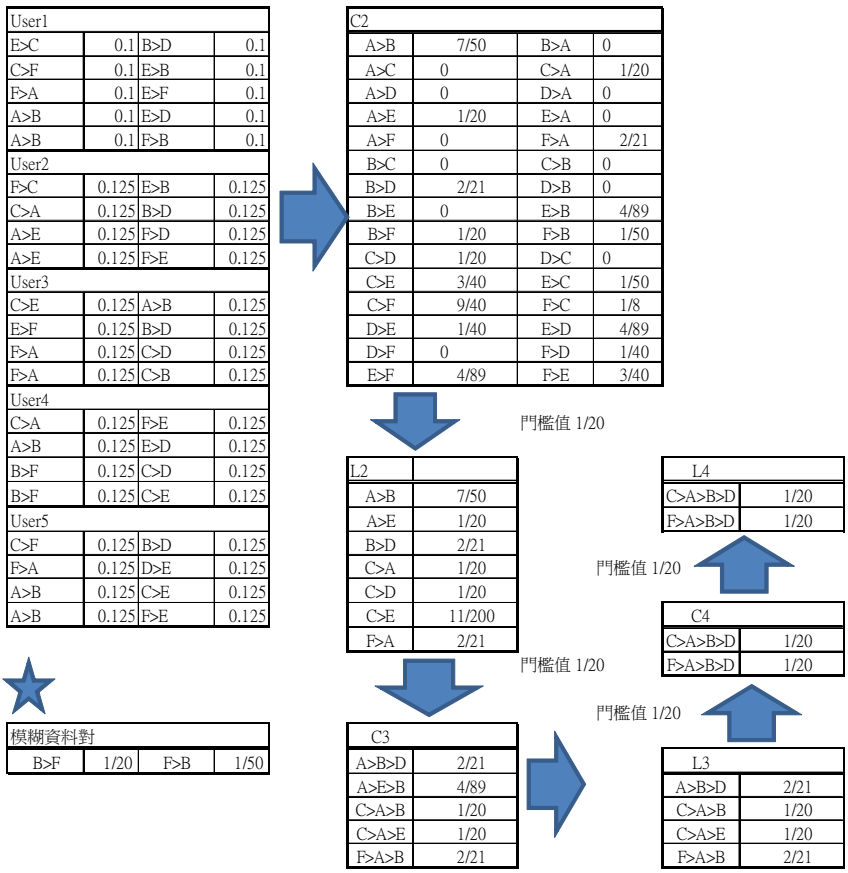

圖 2 共識序列探勘演算法操作流程

5. 實驗結果

為了評估本研究所提出演算的效益，我們利用測試資料進行實驗評估。在這個章節中，我們將呈現我們所做的各項實驗，以及在不同實驗參數數據下的結果，並且對於實驗結果加以討論。

在實驗資料中，我們提供 20 個比較項目數(item)、500 位使用者人數(user)來進行共識序列探勘實驗，並且在實驗中加入參數調整，例如我們利用不同的資料共識率(cr)、支持度(Sup)用來測試不同情境的資料狀況。共識率為實驗測試資料群中所隱含的共識序列所占的機率。

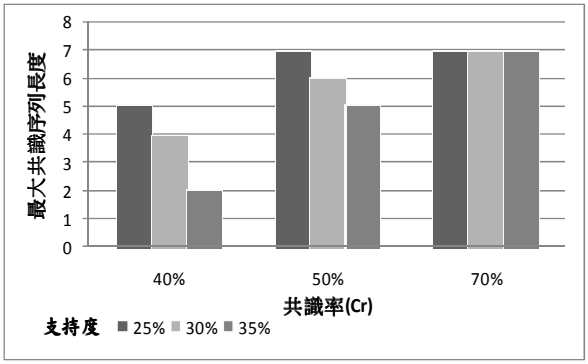

圖 3 不同共識率下、支持度的最大共識序列長度

在圖 3 中，我們所呈現的資料為共識率為 40%，50%，70%下，利用不同支持度而產生的最大共識序列長度。橫座標為資料的共識率。縱座標所呈現的為共識序列的最大長度。此外，我們並利用不同的支持度進行實驗測試，圖表中的支持度分別為 25%、30%以及 35%的狀況下所出現的最大共識序列長度。

在圖 3 中，共識率 40%、支持度為 35%的情形下，我們可以看到其最大共識序列長度只為 2，很明顯低於支持度 25%以及 30%的狀況，分別為 4 和 5。我們判斷原因為，資料群中共識率為 40%，其餘無共識的資料為 60%，因此會互相干擾，所以在支持度要求較高的情形下，會造成共識序列無法變長的狀況。而在資料共識率 50%以及 70%的測試實驗中，我們發現支持度不論是低(25%)或是高(35%)，所呈現出來的共識序列長度皆為相近，大約長度皆為 6~7 左右。所以我們認為在我們的演算法中，資料共識率高於一個門檻值，我們即可以找出相當長度的共識序列，並判別出此群體中的共識序列為何。

除了最大共識序列數量以外，我們也利用不同共識率及不同支持度的情形去實驗測試各長度的共識序列所產生的數量。

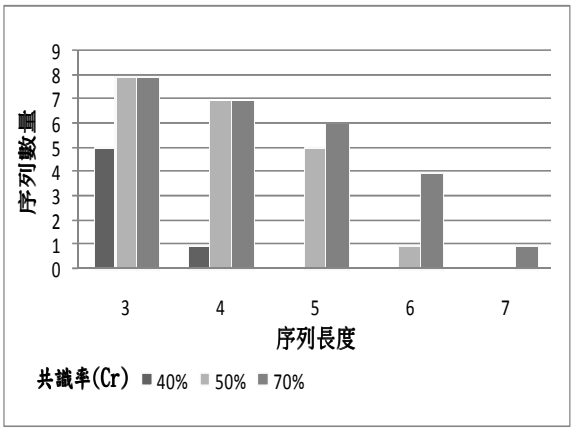

圖 4 支持度為 30%、不同共識率下序列長度數量

在圖 4 中，橫座標為共識序列的長度，分別為 3~7；而縱座標為共識序列的數量。我們希望可以在支持度為 30%的情形下，利用不同的資料共識率所呈現的實驗結果去呈現所找出來的共識序列數量，並對其所探勘出來的共識序列長度以及各長度的數量進行討論。

在圖型中，首先我們發現在資料共識率 40%的情形下，共識序列的數量呈現相對的弱勢，原因也如在圖 3 中所提到的，測試資料容易受到其他沒有共識的資料所影響，因此所呈現出來的共識序列數量以及其長度皆不理想。

而第二部份的實驗為資料共識率為 50%以及 70%情形下，我們發現共識序列的長度以及數量也頗為接近。這部份這相當符合我們一般日常生活所呈現的狀態，因為一般一群人中投票，票數只要超過半數，大致上的決定即為出現在這一部份人的決定中。因此本演算法可以在一定的資料共識率門檻值後找出決策者所要共識序列資料。我們不只可以找出最長的長度，也可以找出在不同長度下所有的共識序列，以供決策者做為決策上的判定之用。

經由上面所提到的兩個實驗數據，我們可以了解到資料共識率對於共識序列的

探勘相當重要。因此我們接下來的實驗將支持度拉高為 35%，並針對資料共識率 50%以及 70%去做實驗結果分析及評論。

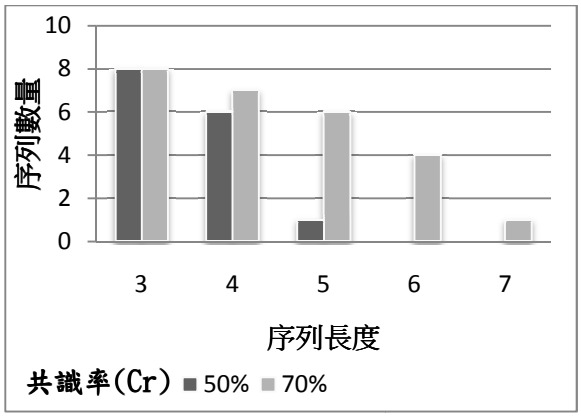

圖 5 支持度為 35%、不同共識率下序列長度數量

在圖 5 中，橫座標為序列的長度，縱座標為序列的數量。在這個實驗中，我們希望針對資料共識率為 50%以及 70%去做分析，利用增加支持度去區分出資料共識率較高的兩個測試資料群。我們可以在圖 5 中發現，資料共識率為 50%的實驗資料在支持度 35%的狀態下，共識序列的長度以及數量相對不太理想；相對的在資料共識率 70%的測試資料中，共識序列的長度以及其產生的共識序列數量皆較為突出。在共識序列長度為 5 的結果中，兩個測試資料就有明顯的差距，50%共識率的資料甚至在長度為 6、7 的長度中皆為 0。

因此，我們認為其資料共識率較為高的情形，可以利用在較具有明顯意識的環境中，可以利用本演算法去找出此群體中大家所認為的共同意見，利用較高的支持度找出較長且較為準確的共識序列。

6. 結論

市場上的產品更新日新月異，在眾多的產品中，決策者如何了解消費者心中的產品排序越來越重要，因此本研究利用使用者序列延伸圖將使用者所填答的成對比較資料進行延伸，並找出其中所隱含的資訊，可增加原本的成對比較資料，增加其準確度。並透過本研究所提出的共識序列演算法方法進行共識序列的探勘，找出大家對於產品的共識排序；此研究可以應用於不同資料共識度的群體中，並且利用不同的支持度進行共識序列探勘，在不同環境下找出較為準確且有用的共識序列，讓決策者可以利用其做為決策制定的依據。

7. 致謝

本研究受國科會計畫補助(計畫編號：NSC 98-2410-H-031 -001)，特此致謝。

文獻探討

[1] Baskin, J. (2008). Comparing Apples and Oranges: Using Consensus Rankings for Decision Support.

[2] Beg, M. M. S., & Ahmad, N. (2003). Soft computing techniques for rank aggregation on the WorldWideWeb. *WorldWideWeb-Internet andWeb Information Systems, 6*(1), 5-22.

[3] Chen, Y.-L., & Cheng, L.-C. (2008). A novel collaborative filtering approach for recommending ranked items. [doi: DOI: 10.1016/j.eswa.2007.04.004]. *Expert Systems with Applications, 34*(4), 2396-2405.

[4] Chen, Y.-L., & Cheng, L.-C. (2009). Mining maximum consensus sequences from group ranking data. [doi: DOI: 10.1016/j.ejor.2008.09.004]. *European Journal of Operational Research, 198*(1), 241-251.

[5] Cohen, W. (1999). Learning to order things. *Journal of Artificial Intelligence Research, 10*, 243.

[6] Cook, W. D., Golany, B., Kress, M., Penn, M., & Raviv, T. (2005). Optimal allocation of proposals to reviewers to facilitate effective ranking. *51, 4*, 65661.

[7] Fagin, R., Kumar, R., & Sivakumar, D. (2003). *Efficient similarity search and classification via rank aggregation*. Paper presented at the Proceedings of the 2003 ACM SIGMOD international conference on Management of data.

[8] Fernandez, E., & Olmedo, R. (2005). An agent model based on ideas of

concordance and discordance for group ranking problems. [doi: DOI: 10.1016/j.dss.2004.01.004]. *Decision Support Systems, 39*(3), 429-443.