基於倒傳遞模糊類神經網路之快速天際線朋友尋找演算法

邱勝敏　　陳奕中　　楊東麟

逢甲資訊工程學系　逢甲資訊工程學系　逢甲資訊工程學系

s951010sam@gmail.com　chenyic@fcu.edu.tw　dlyang.tw@gmail.com

薛念林　　陳錫民

逢甲資訊工程學系　逢甲資訊工程學系

nlhsueh@mail.fcu.edu.tw　seeme.goo@gmail.com

摘　要

分析社群網路上的資訊並且找出相似使用者是一般推薦系統中不可或缺的一項功能，近年來使用多條件演算法天際線查詢並輔以類神經網路來搜尋相似使用者的概念也漸漸興起，但在他們所提出來的方法上仍擁有取得結果與目標使用者不相似進而消耗更多檢查時間，造成執行速度過慢的問題存在。因而本論文使用倒傳遞模糊類神經網路來輔助建立近似天際線區域來解決此一問題。模擬結果則證實了目標方法的有效性及執行效率。

關鍵字：社群網路、推薦系統、模糊類神經網路、天際線查詢

一、前言

近年來，個人化推薦系統已逐漸被熱烈討論，進而也有許多相關的研究討論出現。因為對於各個使用者而言，個人化推薦系統能提供貼近使用者喜好的建議，這類型應用其實可廣泛應用於很多相關應用上，舉凡飲食習慣，旅遊行程甚至是個人喜好等。而個人化推薦系統第一步通常是替使用者找尋相似的使用者，藉此分析相似使用者的資訊找出最符合的建議給予使用者。舉例來說，表一為推薦系統實際範例，表中有六名使用者 A, B, C, D, E 及 F 及其飲食偏好分數(滿分為 10)。若要推薦 A 一餐廳進行用餐，透過表一可明顯看出 D, E 兩人較他位使用者 C 及 F 而言，喜好上更相似於使用者 A。藉由此我們可根據 D, E 所喜好的日式料理餐廳評價來給 A 建議。

直觀上，個人化推薦系統必須先找出與使用者相似的他名使用者，利用其各個維度資訊(e.g. 年齡，性別，食物喜好，打卡景點及次數…等)，透過這些資訊來進行合併處理。在舊有方式的處理上可能會使用 Cosine similarity[10][13]及 *k*-means 演算法[9]進行處理，但是這種處理方式有一嚴重缺陷。由於每個使用者在各個維度上的資料都是獨立且無相關性，所以並不能將各個維度資料做統整處理後進行推薦。舉例來說，在 Cosine similarity 以及 *k*-means 演算法的算法上，表 1 使用者 B 由於在西式料理喜好維度上的資訊與使用者 A 差距甚遠，所以推論結果最終使用者 B 將會被視為不相似使用者。但是這顯然有誤，我們能從表 1 中可明顯看到其實使用者 B 與使用者 A 都是相同喜愛吃中式料理的人，所以以正常觀點來看使用者 B 對於中式料理餐廳的評價對於使用者 A 應該也是具有其參考性質，但這部分舊有演算法中卻無法考慮到，進而造成缺失。故應該有一新的演算法需要被提出來解決該問題。

表一 推薦系統的例子

User	中式	西式	日式
A	9	1	8
B	10	10	3
C	1	8	5
D	8	1	10
E	9	2	8
F	3	8	4

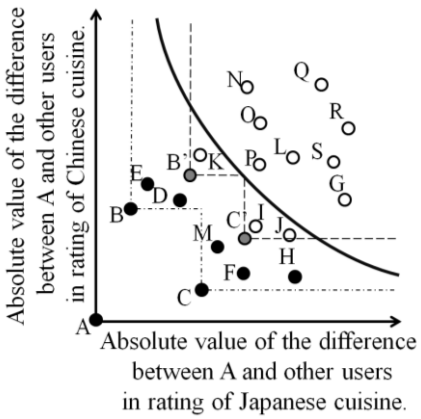


圖 1 參考資料[7]中的近似天際線區域例子

在近幾年來，多條件查詢演算法已被提出並且廣泛使用，其稱為天際線查詢[1][2][3][11][12]，天際線查詢能取得在各個維度都很優秀或部分優秀的資料點(i.e. 天際線資料點)，由於這項原因天際線查詢的出現能解決在多維度上的查詢，在過程中也會將各個維度的資訊分開考慮，因此也有愈來愈多的應用是想將天際線查詢應用在個人化推薦系統上，其中包括了 Hsu *et al*. [8]等人提出了使用天

* 本研究接受科技部編號：MOST 104-2119-M-035 -002 研究計畫經費補助

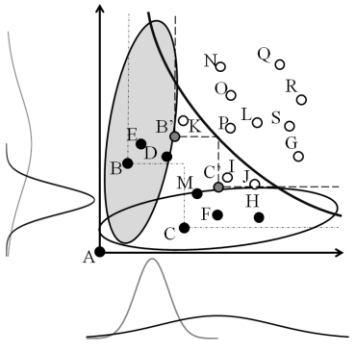


圖 2 使用模糊類神經網路來找天際線區域的概念

際線查詢的概念來幫助使用者規劃個人化的旅遊行程，Chiu *et al.* [6]等人則提出了使用 Depth-*k* 天際線查詢的處理方式來幫助尋找相似使用者，並解決了在傳統推薦系統上維度資訊不可分割的難題。以及 Chiu *et al.*[7]等人利用天際線區域的概念近一步延伸輔以類神經網路模擬出近似天際線區域 來解決在使用 Depth-*k* 天際線查詢避免搜尋不相似的使用者及執行速度過慢的問題。這些相關研究都可有效的解決在傳統推薦系統上大大小小的問題。舉例來說，圖 1 為 Chiu *et al.* [7]近似天際線區域的範例，圖中 X 軸與 Y 軸分別代表其他使用者與使用者 A 在日式料理喜好以及中式料理喜好分上的差距。在此範例中我們可以明顯看出在 B, C 比起其他使用者是更相似於使用者 A，而在沒有被支配的使用者 B, C 分別往 X 軸及 Y 軸延伸一個 α 值，藉此形成 B'以及 C'兩虛擬點。並利用類神經網路模擬出近似天際線區域的界線避免當資料點過多，會需要與行程天際線區域的所有虛擬點進行比較而造成花費大量時間成本。從上述範例我們可以得知使用 Chiu *et al.*[7]等人所提出的近似天際線區域的確能有效改善傳統個人化推薦系統的缺陷，並且也解決了 Depth-*k* 天際線查詢的缺失，但是其實在這項技術的背後仍有兩大致命缺陷。第一，由於是使用類神經網路概念去建立界定線，但是實際上由於類神經網路本身限制，利用類神經網路所建立的天際線區域可能會造成多取出非相似使用者的點，例如，在圖 1 中，使用者 I, J 及 K 其實並沒有非常相似於使用者 A，但是由於在利用近似天際線區域後，使用者 I, J 及 K 就會被選入相似使用者，這是錯誤的。第二，若想要界定這些在中間區域的使用者是否相似於使用者，也會耗費更多的檢查時間，所以會造成相當大的成本消耗。於是要有一些更新的演算法所被提出來解決上述兩大難題。

本論文提出以模糊類神經網路的概念來解決[7]的問題，我們預計使用模糊類神經來解決這些問題，如圖 2 所示，首先假定目標使用者 A，我們會利用高斯函數來幫助我們將目標區域建立出來，在圖 2 上兩橢圓內範圍即為近似天際線區域的範圍，透過調整模糊類神經網路中的模糊層，藉以模擬出更擬真的範圍，並且可以有效避免舊有僅使用類神經網路進行模擬天際線區域的第一個問題，不會抓取過多的資料點，我們所需要的檢查時間也會相對應的減少，所以也可以解決第二個問題。

本篇論文接下來篇章安排如下，第二章是天際線查詢的相關論文介紹。第三章會介紹如何從使用者的朋友中取出 Depth-*k* 天際線的朋友。第四章將會介紹如何用模糊類神經網路建立近似天際線區域。第五章為實驗模擬。最後第六章為結論。

二、相關論文

(一) Branch and Bound Skyline algorithm (BBS) [11]

此法是目前最常被使用的天際線搜尋方法之一，主要透過對資料建立索引來加速搜尋的速度。首先所有資料點會整合到 R-tree 內。R-tree 會將鄰近的資料點以 Minimum Bounding Rectangle (MBR)包圍起來。接著，僅有那些與天際線交集的 MBR 會被取出來處理，因為僅有這些 MBR 有可能包含天際線資料點。藉由此方法，大部分的資料點在進行查詢時都不會被檢查到，進而加快查詢的速度。

(二) Sort and Limit Skyline algorithm (SaLSa) [1]

這個天際線查詢演算法是透過資料特徵值得到之門檻值檢查此資料點是不是可成為天際線資料點。特徵值可能是各資料點維度中的總和，乘積以及最小值等。流程上當各個資料點進行檢查的時候，會先計算資料點的特徵值，並且與門檻值相互比較，若小於門檻值之下即可成為天際線資料點。另外除了這種方式以外還可透過特徵值進行排序，透過這樣方式只要檢查到某項資料點不符合天際線資料點(即特徵值大於門檻值)，而在這項資料以後的資料都不需要進行檢查了，因為其後各項資料點必然不是天際線資料點，並因此加快查詢速度。

三、Sorted First Skyline Algorithm

Sorted First Skyline(SFS) Algorithm 是熱門的天際線演算法中的其中一種。其優點在於可以在資料量過小並且不利於建立 R-tree 的情況下，利用較短的運算時間得到天際線結果。此情境與本論文的情境相符，所以會直接使用此演算法。SFS 的核心概念如下，其中曼哈頓距離[4]代表資料點在不同維度之總和。

Lemma 1: 曼哈頓距離較大的資料點 q 會將曼哈頓距離較小的資料點 p 所支配。

Proof:利用反證法來證明此法，若 q 支配 p，且 q 的曼哈頓距離也大於 p。我們根據支配定義推得，若 q 支配 p 則表示 q 在各維度都數值都比 p 來得更佳。所以在所有維度數值總和(曼哈頓距離)上也必然比 p 更小，但這明顯與假設相互違背，即可得證。

依照此定理我們能將社群網路中使用者依照曼哈頓距離進行排序，從曼哈頓距離最小的使用者

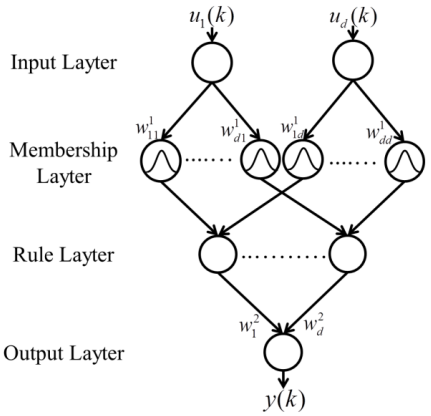


圖 3 本論文所使用之模糊類神經網路

開始往曼哈頓距離較大的使用者進行支配比較。在每次比較中可獲得下列三種情況[6]。

- ***p* 與 *q* 無法比較:**此狀況表示 *q* 可能成為天際線朋友，所以會被保留後續進行檢驗。
- ***p* 支配 *q*，但 *q* 被少於 *k* 個使用者支配:**此狀況下 *q* 仍可能成為 Depth-*k* 天際線朋友所以其支配次數會被加 1，並保留在演算法中等待進一步的處理。
- ***p* 支配 *q*，但 *q* 被 *k* 個的使用者支配:**此狀況下 *q* 已不可能成為天際線朋友所以會被直接從資料集中刪去。

四、使用模糊類神經網路來逼近天際線區域

(一) 目標模糊類神經網路的訓練資料

本論文訓練資料來自於目標使用者的好友資訊，其中訓練資料的輸入為好友資訊的座標，舉例來說，如圖 2 中使用者 B 以及使用者 K 等會輸入各自座標值。而訓練資料輸出則為該使用者是否落於天際線區域的範圍內，例如圖 2 中的使用者 B 將會輸出 1，使用者 K 則會輸出-1，其中輸出值 1 表示落於天際線區域內，-1 則反之。

(二) 目標模糊類神經網路的架構

本論文預計使用模糊類神經網路架構如圖 3 所示，其中共分別為 Input Layer, Membership Layer, Rule Layer 以及 Output Layer 等四層，其中 Input Layer 節點數目為資料點維度數，舉例來說，假設今日我們要建立如圖 2 所示之天際線區域，則目標模糊類神經網路 Input Layer 節點數應為 2，因為在圖 2 中，僅共有 2 個維度的資料。而 Output Layer 僅有一個節點數，負責輸出該資料點 Q 是否落於目標天際線區域的範圍內(值介於-1~1 之間)，若數值輸出愈接近 1 則表示該資料點有較高機率落於天際線區域內，反之若趨近於-1 則表示該點幾乎不可能位於天際線區域內。以下圖 3 為了簡化討論，首先定義模糊類神經網路第 *j* 層第 *i* 個節點的輸入以及輸出分別為 u^j_i 及 o^j_i。在圖 3 中的模糊類神經網路預計網路函數如下所示。

Input Layer:本層節點並不做任何動作，僅將輸入層傳遞前往第二層。此層第 *i* 節點數學表示公式為:

$$O_i^1 = u_i^1 = r. \qquad (1)$$

Membership Layer:本層節點會先結合前一層的輸入資訊，接著節點們會使用高斯函數來將輸入資訊轉為「模糊的敘述」。其中第 *i*, *j* 個節點可表示為

$$O_{ij}^2 = esp\left\{\frac{\left(O_i^1 - m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}\right\}, \qquad (2)$$

其中，m_{ij} 和 σ_{ij} 是高斯函數的平均值及標準差。

Rule Layer:此層節點利用了「AND」的方式進行整合上一層模糊資訊，整合方法如下:

$$O^3 = \prod \mathrm{u}_i^3. \qquad (3)$$

Output Layer:此為模糊類神經網路最後一層，本層作為輸出需要負責進行解模糊化的動作。前一層的模糊資訊傳遞到此時，會透過一線性組合解模糊化，並輸出模糊類神經網路最終結果。公式如下:

$$y_m = fl = \sum_{j=1}^{m} w_{mj} \prod_{i=1}^{n} esp\left[\frac{[x_i - m_{ij}]^2}{(\sigma_{ij})^2}\right]. \quad (4)$$

(三) 目標模糊類神經的訓練演算法

本論文預計使用倒傳遞演算法[5]來訓練模糊類神經網路，假設訓練演算法主要目標為

$$\mathrm{E}(w,k) = 1/2(y^d(k) - y(k))^2 = 1/2e(k)^2, \quad (5)$$

其中 $e(k) = y^d(k) - y(k)$，為理想輸出與網路輸出的誤差值。以下使用 *v* 來作為模糊類神經網路內的總稱，內部包含了 *w*, *m*, *σ* 等參數調整公式為

$$v(k+1) = v(k) - \zeta\left(\frac{\partial E}{\partial v}\right), \qquad (6)$$

其中 $\partial E/\partial v$ 代表了參數 *v* 對於誤差 *E* 的影響，根據此公式分別帶入 *w*, *m*, *σ* 等參數可推得 *m(k)* , *σ*(*k*) 及 *w*(*k*) 參數調整公式為

$$m(k+1) = m(k) + \zeta ew \prod o^2 \frac{1}{\sigma}, \quad (7)$$

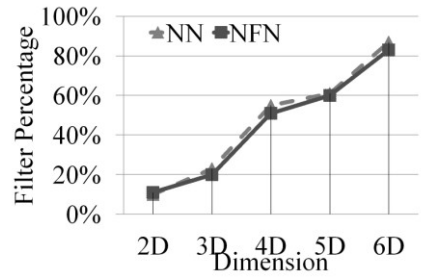
$$\sigma(k+1) = \sigma(k) + \zeta ew \prod o^2 \left[\frac{o^1 - m}{\sigma^2}\right], (8)$$

$$w(k+1) = w(k) + \zeta eo^3. \qquad (9)$$

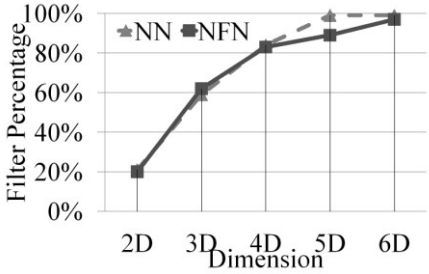
最終藉由上述公式可分別調整第一層與第二層間權重 *w* 及第二層中高斯函數平均值 *m* 及標準差 *σ*，進而使模糊類神經網路完成倒傳遞演算法。

五、Simulation

這個章節我們使用了對於天際線查詢來說，一般情況的 Independent 資料集以及在最差情況下的 Anti-correlated 資料集為例，其中各個資料及都是以人工的方式產生各一百萬筆的資料。此外，由於



(a)



(b)

圖 4 過濾比例與維度之關係(a) Independent 資料集，(b) Anti-correlated 資料集。

在天際線相關演算法中維度是一種大條件，所以本論文主要驗證在於不同維度數目時，我們所提出的演算法可以幫助我們過濾出更精確的使用者。比較對象則是 Chiu *et al.*[7]等人所提出的類神經網路所建立之近似天際線區域演算法。所有實驗皆以 Matlab 撰寫。

圖 4 是我們的模擬結果。從這兩張圖我們可以得知，我們所提出的使用模糊類神經網路的確較 Chiu *et al.*[7]等人所提出的類神經網路所建立之近似天際線區域效果更加優越。

此外，在過濾資料點所用時間部分，由於我們的方法與[7]的方法所提出之類神經網路的時間複雜度都是 O(*n*)，其中 *n* 是資料點個數，所以在過濾同一個資料集時，時間並不會差異過大。這也就是說，我們的方法能夠使用相同的時間來過濾掉較多的點，因此我們的方法將較[7]優秀。

六、Conclusion

本論文改善了就有天際線區域的技術來幫助在社群網路中尋找與使用者相似之使用者。這種新方法改善了[7]所提出利用類神經網路建立近似天際線區域的演算法缺陷，其中包含了(1)使用類神經網路概念去建立界定線，造成多取出非相似使用者的點，以及(2)若想要界定這些在中間區域的使用者是否相似於使用者，會耗費更多的檢查時間，造成相當大的成本消耗。最終，模擬的部分證實我們的方法的有效性及效率。

本論文目前方式雖然已經證實較舊有方式更佳，但仍有部分缺陷存在，所以未來預計使用基因演算法來更新演算法，因為基因演算法是擁有獲得最佳解的方式，所以可以有效幫助我們未來在設計新的演算法。

參考文獻

[1] I. Bartolini, P. Ciaccia, and M. Patella, “SaLSa: Computing the skyline without scanning the whole sky,” *proceeding on ACM International Conference on Information and Knowledge Management*, pp. 405-414, 2006.

[2] S. Borzsonyi, D. Kossmann, and K. Stocker, “The skyline operator,” *proceeding on ICDE*, pp. 235-254, 2001.

[3] Y. C. Chen and C. Lee, “Depth-k Skyline Query for Unquantifiable Attributes in Distributed Systems,” *proceeding on International Conference on Artificial Intelligence and Soft Computing (ASC)*, 2011.

[4] Y. C. Chen and C. Lee, “The *σ*-Neighborhood Skyline Queries,” *Information Sciences*, vol. 322, no. 11, pp. 92-114, 2015.

[5] Y. C. Chen and C. Lee, "A Neural Skyline Filter for Accelerating the Skyline Search Algorithms," *Expert Systems*, vol. 32, no. 1, pp. 108-131, 2015.

[6] S. M. Chiu, Y. C. Chen, H. Y., Su, and Y. L. Hsu, "Finding Similar Users in Social Networks by Using the Depth-k Skyline Query," *Proceeding on IEEE conf. on Consumer Electronics*, 2015.

[7] C. C. Hou, C. K. Chang, Y. C. Chen, H. Y. Su, and Y. L. Hsu, "Finding Similar Users in Social Networks by Using the Neural-Based Skyline Region," *proceeding on Int. Conf. on Artificial Intelligence for Engineering*, 2015.

[8] W. T. Hsu, Y. T. Wen, L. Y. Wei, W. C. Peng, "Skyline Travel Routes: Exploring Skyline for Trip Planning," *proceeding on IEEE Int. Conf. on Mobile Data Management (MDM)*, pp. 31-36, 2014.

[9] M. J. Li, M. K. Ng, Y. M. Cheung, and J. Z. Huang, “Agglomerative fuzzy K-Means clustering algorithm with selection of number of clusters,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp.1519-1534, 2008.

[10] E. H. C. Lu, C. Y. Chen, and V. S. Tseng, "Personalized Trip Recommendation with Multiple Constraints by Mining User Check-in Behaviors," *proceedings on International Conference on Advances in Geographic Information Systems*, 2012.

[11] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “An optimal and progressive algorithm for skyline queries,” *proceeding on ACM SIGMOD International Conference on Management of data*, 2003.

[12] Z. Peng and C. Wang, " Member promotion in social networks via skyline," *World Wide Web*, vol. 17, no. 4, pp. 457-492, 2014.

[13] O. Shamir and N. Tishby, "Stability and model selection in *k*-means clustering," *Machine Learning*, vol. 80, no. 2-3, pp.213-243, 2010.

[14] Jeen-Shing Wang and Yen-Ping Chen, "A Hammerstein Recurrent Neuro-Fuzzy Network with an Online Minimal Realization Learning Algorithm," *IEEE Transactions on Fuzzy Systems*, vol.16, no. 6, pp. 1597-1612, 2008.