藉由客戶偏好和成本限制搜尋天際線

Finding Skyline with Customer Preferences and Cost Constraints

Yu-Chi Chung I-Fang Su Chiang Lee PinChieh Huang

CSIE of CJCU IM of Fotech CSIE of NCKU

justim@mail.cjcu.edu.tw ifang.su@gmail.com leec@mail.ncku.edu.tw pinchieh.huang@gmail.com

摘要

天際線和優控關係的分析在商業的應用上越來越受到重視。在商業應用的情境中，資料空間中的一個點，就代表了一個廠商的產品。若廠商的產品為一個 skyline point，那代表本產品是一個具有競爭力的產品。更精確地說，代表了這個產品在市場上，並沒有競爭對手。在本論文中，我們提出了「藉由客戶偏好和成本限制搜尋天際線」的問題。在本問題中，我們探討了以廠商的角度來看，資料處理的技術如何幫助廠商找出資料空間中具有足夠利潤，有競爭力，並且能對使用者有足夠足引力的產品。給一個客戶偏好的集合和成本限制，我們提出了一系列的演算法，幫助廠商尋找產品空間中符合成本限制（如此一來才有利潤），以及能吸引最多使用者的 skyline 產品。在論文中，我們詳細地定義了此一問題，提出了設計演算法的原理、pruning 的技巧，以及相關的理論。此外，我們也證明了演算法以及相關輔助定理的正確性。最後，我們做了完整的實驗評估而且展示了演算法的效能。

關鍵詞：天際線查詢、過濾演算法、客戶偏好。

Abstract

The importance of skyline and dominance relationship analysis has been well recognized in multicriteria decision-making applications. In this paper, we propose the problem of “Finding Skyline with Customer Preferences and Cost Constraints”, which utilizes the concept of dominance for business analysis from a microeconomic perspective. Our problem aims to discover the dominance relationship between products and potential customers. Given a set of customer preferences, we want to help the company to design set of competitive products so that the products can satisfy as many customer requirements as possible and the cost of producing the products is within a specified threshold. By “competitive products” we mean that the products cannot be dominated by other products in the market. We formally define the problem and discuss the difficulty of the problem. We also present the foundation and the intuition of our pruning methods. Then we proposed three efficient algorithms that utilize our pruning methods to address the problem. In addition, we also prove the correctness of our proposed algorithms. Finally, we conduct a thorough experimental evaluation that demonstrates the efficiency of our proposed algorithms.

Keywords: *Skyline query processing, User Preferences and Pruning Algorithm.*

一、簡介

本研究利用 dominant relationship analysis 的想法來找尋新產品的最佳銷售方案。在 Cuiping Li 等人於 2006 年提出的論文中[1]，利用 dominant relationship analysis 的想法來解決個體經濟學（microeconomic）中的問題。之後，相關的研究便不斷地出現，讓這個領域成為資料庫社群的一個熱門的議題。我們用底下的一個例子來說明本計畫的動機。

考量一個機製造商，其想要在市場上推出一款新產品。銷售部門進行了市場調查，調查使用者對於手機的偏好，其結果如圖 1 (a)所

示。在圖 1 (a)中，x 軸代表手機相機的像素值的倒數，y 軸代表手機的重量。圖 1 (a) 中的每一黑點都代表了一個使用者對手機的偏好。舉例而言，使用者 c_1 偏好像素較高的手機，但對於手機的重量較不在意。而使用者 c_6 則偏好較輕的手機，但對相機像素的多寡並不在意。

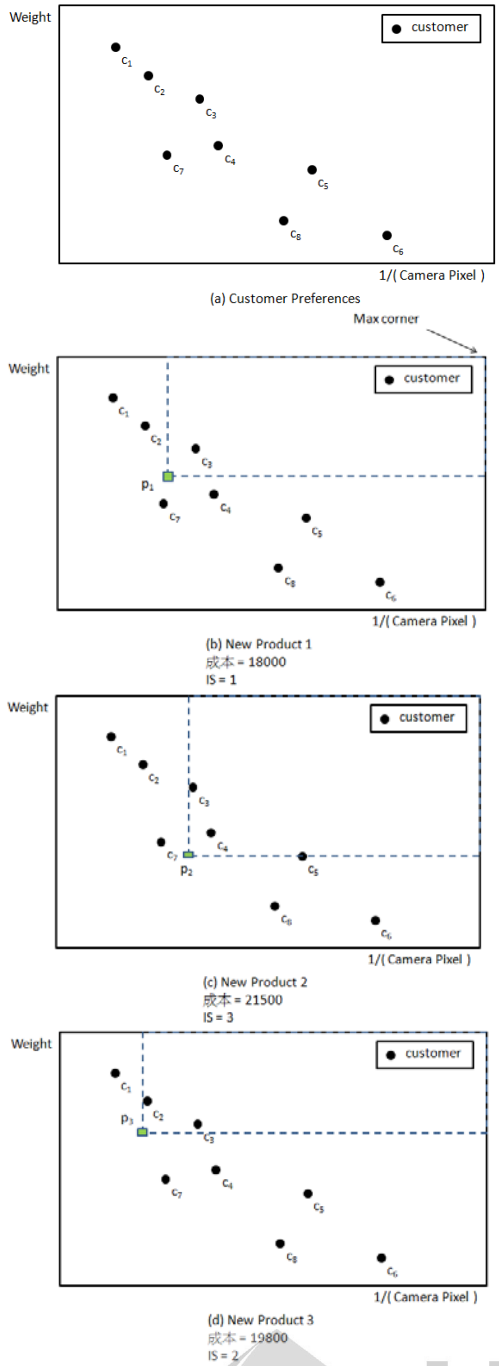


圖 1 購買手機範例

在本研究中，我們假設使用者偏好「數字較小的特性」。舉例來說，對使用者而言，手機的重量「越輕越好」，且像素的「倒數『越小越好』」 (也就是像素值越大越好的意思)。若一個產品 p，其規格都比使用者偏好 c 來得好，那我們稱產品 p *dominates* 使用者偏好 c [10, 11]。更精確地來說，就是 p 的每一個屬性，都比使用者預期來得好的意思。以數學的方式來表達，那就是：

$$p \text{ dominates } c \iff p[i] \le c[i], \forall i$$

其中 $p[i]$ 代表產品第 i 個屬性值，而 $c[i]$ 代表使用者對於第 i 屬性值的偏好。於是，若能設計一個新產品，其「規格」可以 dominate 使用者的偏好，那代表使用者便會青睞這個產品。舉例而言，假設圖 1 (b)中 c_3 代表(重量 140g，600 萬像素)的偏好，圖 1 (b) 中的方形 p_1 代表一個新的產品，其規格為（重量 135g，650 萬像素）。在這個例子中，因為(1) p_1 手機的重量 135g ≤ 使用者偏好的 140g，且(2) p_1 像素值 1/650 ≤ 使用者偏好的 1/600。

因此我們說 p_1 的規格 dominate c_3 使用者的偏好，因此 c_3 這個使用者，會青睞規格為（135g，650 像素）的手機 p_1。

我們定義新產品 p 及 max corner (即資料空間的右上角點)所形成的區域，為 p 的 *dominating region* （在圖 1 (b)以虛線框表示）。所有落於 dominating region 的使用者，都會被 p 所 dominate，因此就是 p 的目標使用者群。以圖 1 (b)來說，p_1 的 dominating region 以虛線方框表示。c_3 落於此 dominating region 中，因此 c_3 是 p_1 的目標客戶群。

圖 1 (c)及圖 1 (d)展示了另外兩種新手機的規格 p_2 及 p_3，以及它們對應的 dominating regions。對於 p_2 而言，c_3、c_4 及 c_5 落於其 dominating region 中，所以 c_3、c_4 及 c_5 是 p_2 的目標客戶。同理，c_2 及 c_3 為 p_3 的目標客戶。

在本研究中，我們定義落在產品 dominating region 的使用者總數，為此產品的「影響分數」 (*influence score*；*IS*)。產品的 *IS* 越高，代表此產品的影響力越高，也代表此產品在市場上受到越多使用者的青睞。以圖 1 (a)而言，p_1 的 dominating region 中只有 c_3，因此 p_1 的 IS 為 1。使用同樣的方法，可以得到圖 1 (b)中，p_2 的 IS 為 1+1+1=3，而圖 1 (c)中，p_3 的 IS = 1+1=2。

在這樣的定義下，我們只要能設計出一個落於資料空間「左下角」的產品（即重量無限輕，且像素值無限大）的手機，那麼必可 dominate 所有的使用者，也就成為最受歡迎的手機。然而，設計這樣的產品必然所費不貸，對公司的財務造成大量的負擔。因此，手機製造商要思考如何在「成本」可負擔的情況下，吸引最多的使用者。因此我們定義一個產品 p 為是一個「具競爭力」的產品，則在市場上，沒有其他的產品可以 dominate p。這裏的 dominate 定義如下：給定兩個產品 p_i 及 p_j，我

們說 p_i dominates p_j，那麼就代表底下的事情成立：p_i 的生產成本較 p_j 來得低，且 p_i 的影響力較 p_j 來得高。

現在我們想問底下的問題：給定一個商品的成本下限(以 $cost_L$ 表示)及成本上限(以 $cost_U$ 表示)和一群客戶的偏好，是否可以找到「所有」的產品 S，使得若產品$p \in S$，則 p 的成本必小於或等於 $cost_U$、大於或等於 $cost_L$，並且 p 為具競爭力的產品。

$cost_U$ 代表了一個產品的生產成本的上限。若產品的生產成本超過 $cost_U$，代表廠商無法由此產品獲取利潤。我們用 $cost_U$ 來確保廠商所生產的產品都是可以獲利的。廠商雖然都會盡力追求利潤最大化，但也不能一直壓低成本來達到這個目的。這是因為壓低成本有很高的機會是透過降低商品的規格來達成。而降低產品規格，將連帶影響到產品在市場上的影響力。因此我們使用 $cost_L$ 來限制產品的生產成本下限。另一個使用 $cost_U$ 及 $cost_L$ 來限制生產成本的原因是避免找出過多的生產方案。由於資料空間的每一點都是一個可行的新產品。因此我們使用 $cost_U$ 及 $cost_L$ 來限制可行解的空間，使得所產生的生產方案不要過多。

解決這個問題直覺的方法是找出所有的產品，過濾成本介於 $cost_U$ 及 $cost_L$ 的產品，之後再執行 dominance test 以找出具競爭力的產品。然而，由於資料空間中有無限多個點，且每個點都是一個可能的產品。因此直覺得做法需要極為大量的運算來掃描空間中所有的資料點，在實務上並不可行。

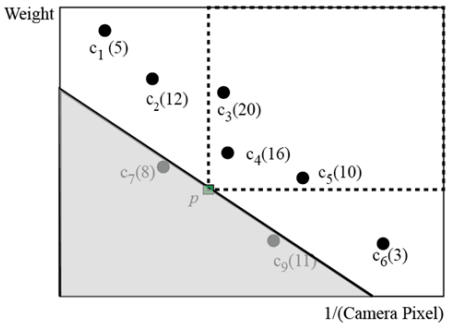
就我們所知，本研究是第一個提出並解決此問題的研究。最接近我們的研究是由 Cuiping Li 等人於 2006 年提出的研究[1]，其中有一個題目是關於固定成本下生產一個新產品。本研究認為只侷限在一個成本並不恰當，應該是給定一個合理的成本區間，而研究[1]中的方法必須要給定很多個甚至無限多個成本才有可能達到本論文題目的需求。例如題目的成本上限是 11000、成本下限是 10000，那用論文[1]的方法就要執行 1000 次(11000-10000)方能解決問論文的問題。明顯的，研究[1]的解法在解決本問題時，會帶來很大的計算效能的浪費。因此並不是一個可行的解決方案。

本研究剩餘部分章節規劃如下：第二章將介紹完整的相關研究。我們在第三章正式定義我們的問題，並在第四章提出處理問題的演算法，接著在第五章呈現實驗後的結果。最後則為結論與未來工作。

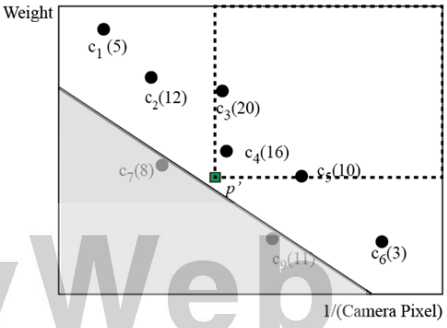
二、相關研究

在 Kleinberg 等人所發表的論文[8]中揭示了以個體經濟學的觀點來探討 data mining 的問題後，在資料庫領域中，有許多相關的研究發表。其中 Li 等人於 2006 年所發表的論文[1]，是第一篇以透過 dominance relationship analysis 來探討廠商如何透過使用者的偏好，來設計新產品的論文。其後有數篇具影響力的論文[2-6]在國際重要的期刊或研討會被發表。足見這個研究議題正方興未艾。

Li 等人在 2006 年所提出的論文中[1]利用 dominance relationship analysis 來探討廠商如何在一定成本的考量下，決定自己產品的每個屬性(即設計新產品)。我們利用圖 3 來說明。在圖 3 中，使用者偏好以圓點表示。若在資料空間中的任意一點，便決定了一個新產品的所有屬性。如圖 2 (a)中的 p 點，其重量為 2.1，且像素屬性為 300 萬。圖 2 (a)中的斜線則是產品的「成本線」，在成本線以下灰色的域為「無利潤區」。若產品落在灰色的區域，表示這個產品的設計無法讓廠商獲得利潤。在[1]中探討如何在「成本線」上決定一個點 p，這個 p 代表一個新產品，且其可以 dominate 最多的使用者偏好。舉例而言，圖 2 (a)中的 p 點就是一個在成本線上，可以 dominate 最多使用者偏好的產品。



(a) [1] 論文中舉的例子



(b) 若將產品放置在p' 點，會更省成本且影響力不變。

圖 2 在成本線上決定產品的問題。

[1]所探討的問題和本研究的問題有點類似。不同之處說明如下：

1. [1]的方法不一定能找到最佳的成本價值。[1]中的方法是在成本線上找到一點，其可以 dominate 最多使用者偏好（即影響力最大），圖 2 (a)是一個例子。但以商業的角度而言，要達到圖 2 (a)的影響力，可能有更省錢的方式，如圖 3 (b)所示。若新產品落在 p' 點，那可以更省錢，但達到相同的影響力。

2. 我們的方法是在成本限制下，找出「所有」具影響力銷售方案。就大部分的商業公司而言，只找出「一個」銷售方案是不切實際的，因為還有許多的隱藏的因素沒有考量（例如品牌忠誠度…等）。因此業務部門若能提出多的方案，較能被決策部門來接受。[1]中所提的方法是可以用來解決本研究所提出的問題，但會很沒有效率。直覺的作法是在資料空間中畫出無限條「成本線」。對每一條成本線都執行[1]所提出的演算法。明顯的，這樣的解法會造成大量的計算成本，現實上是不可行的。

3. 本研究中採用了和[1]不一樣的成本計算方式，這也會造成演算法設計上的修改。

Zhang 等人延續了[1]的想法，引入了賽局理論（game theory）工具，探討如何使用 dominance relationship analysis 來幫助廠商設計新產品[2]。在論文[2]中，除了考量客戶偏好外，也引入了競爭對手的產品。廠商在設計新產品時，要考慮競爭對手產品的屬性，讓整個市場達到納許均衡（Nash Equilibrium）[9]。論文[2]與本研究的差別在於如同前述，廠商只會生產一個新的產品，但本研究則是討論多個銷售方案的設計。

在 Peng 等人提出的論文中[3]，探討了兩個問題：(1)如何設定「現有」產品的價格，可使得現有產品成為市場中的 skyline，這個問題稱為 finding simple competitive price problem，(2) 如何設定「現有」產品的價格，可使得現有產品可以 k-dominate 現有市場中的產品，這個問題稱為 finding k-dominating competitive price problem。舉例而言，市場上有一群旅館，它們都有它們的價格，以及距離海灘的距離。現在給定一間旅館，其距離海灘的距離已經決定了，那麼如何設置這間旅館的價格，使得這間旅館可以成為現有市場上的 skyline？和本專題最大的差別在於[3]並沒有考量客戶的偏好，而本專案有。另外，[3]只調動「price」這個屬性，來讓產品成為 skyline。而本專案中，由於是要建立新的產品，所以要考量產品「所有」的屬性，讓產品成為 skyline。

Lu 等人[4]探討了如何以最少的「成本」，「升級」一個現有的產品，使這些產品為 skyline。令 P 是一群現有的產品，而 T 是一群 uncompetitive products。[4]中討論如何在 T 中選出 k 個 products，其有最少的 cost，且可以升級這 k 個 products，讓這些升級後的 products 不被 P 中的產品給 dominate。此處的「升級」，的是決定產品的每個屬性，因此以較寬鬆的角度來看，可以說是在設計一個新產品。作者替每個屬性指定了一個升級函式，透過指定升級函式，便可以計算每個屬性值所帶來的成本，因此可以計算出一個產品的總升級成本。[4]與本計畫最大的差別是並不考量到使用者偏好，而我們是由使用者偏好的觀點來探討設計新產品的問題。

Peng 等人[5]探討了如何決定產品的 price，讓產品可以成為現有市場的 skyline，並且獲得最大的利潤。我們用圖 4 來解釋[5]所探討的問題。圖 4 (a)中展示了現有市場中的產品，而圖 4 (b)中則是廠商的產品。現在廠商要在圖 4 (b)中的產品中挑 k 個（在本例中，k=2），使得這 k 個產品都為 skyline（即不能被圖 4 (a)中的產品給 dominate，且這 k 個產品彼此互不 dominate），且獲利要最高。其中，一個產品的獲利設定為 price-cost。在這個例子中，當 k=2 時，q1 的 price 設為 250，而 q2 的 price 設為 300，是最好的選擇。[5]和我們的差別在於我們有考量客戶偏好，而[5]沒有。另外，我們要考量到所有維度的屬性值，而[5]只需要考量到價格這個屬性值。

Package	*Distance-to-beach (km)*	*Price*
p_1	*7.0*	*200*
p_2	*4.0*	*350*
p_3	*1.0*	*500*
p_4	*3.0*	*600*

（a）現有市場中的產品（旅館）

Package	*Distance-to-beach (km)*	*Price*	*Cost*
q_1	*5.0*	*?*	*100*
q_2	*4.5*	*?*	*200*
q_3	*0.5*	*?*	*400*

(b) 在這個表格中挑出k個產品，並決定它們的價格，使得這k個產品為skyline，且獲利最高。其中每個產品的獲利為 price − cost。

圖 3 設定新產品的價格，使新產品得以成為 skyline，且獲利最高（取自[5]）。

Lin 等人[6]探討了考量現有的競爭對手，以及客戶偏好，如何在公司中所生產的 n 個雛型產品中，選出 k 個產品，讓這 k 個產品有最大的競爭力。這裏的競爭力，指的是本計畫中的影響力（即 influence score；IS）。[6]與本研究所探討的問題不同處在於：(1) [6]的目的是由「現有」產品中挑選 k 個最具競爭力的產品，而我們是要決定產品的屬性，使得新產品具有競爭力。兩者在問題的本值上有所不同。(2)[6]

中所選出來產品並不定是 skyline。也就是說，即使公司的產品 p 被市場上現有的產品 dominate，但只要 p 可以 dominate 使用者偏好，那麼[6]假設這些使用者依然會購買 p。相反的，本研究要求所設計出來的新產品一定要是市場上的 skyline 產品。(3) 本研究中是以「總成本」的限制為出發點，希望能找出具競爭力的銷售套件，但[6]中並沒有考量「總成本」的限制。

由前述的分析可知，雖然目前已經有少數的論文由廠商的角度出發，利用 dominance relationship analysis 來設計新的產品，或是改善現有的產品，達到提升競爭力（或影響力）的目的。但這些論文與本研究所考量的方向均有所不同。另外，現有的論文均發表在國際知名的期刊或是會議上，足見這個議題的重要性，還有發展性。

三、問題定義

本小節中，我們正式定義我們的問題。表格 1 列出了本研究中所有的符號，以及它們所代表的意義，我們會在本小節中說明每個符號的定義。

表格 1 研究中所用的符號以及它們所代表的意義。

Notation	說明
D	d-dimensional space
$P=\{p_1, p_2,\ldots,p_m\}$	P 是一群產品的集合
p_i	一個產品 p_i
$p_i[k]$	產品 p_i 的第 k 個屬性值
$p_i.IS$	產品 p_i 的影響力
$p_i.c$	產品 p_i 的製造成本（cost）
$p_i \prec p_j$	產品 p_i dominates 產品 p_j
$C = \{c_1, c_2,\ldots,c_n\}$	一群使用者偏好的集合
c_i	一個使用者偏好
$c_i[k]$	使用者 c_i 對於產品的第 k 個屬性值的要求
$cost_U$	廠商所設定每個產品的生產成本上限（即產品的生產成品不得超過這個上限）
$cost_L$	廠商所設定每個產品的生產成本下限（即產品的生產成品必須達到這個下限）

定義 $c_i = (c_i[1], c_i[2], \ldots, c_i[d])$ 為一群使用者的偏好 (preference)，其中 $c_i[k]$ 為使用者對第 k 屬性的要求。公司在生產產品之前，會先做市調，市調中會提出數種產品的規格(即一群 $(c_i[1], c_i[2], \ldots, c_i[d])$)，並且讓使用者勾選其所偏好的產品有那些。以圖 1 (a)而言，c_1, c_2,…, c_8 代表有 8 位使用者對 8 個不同的規格感興趣。

定義 p_i 為公司所製造的一個產品(如手機、notebook… 等)。p_i 包含了 d 個屬性 $\{A_1, A_2,\ldots,A_d\}$。令 $p_i[k]$ 為 p_i 的第 k 個屬性值，且每個屬性都是一個 real number。在不失一般性下，我們假設屬性值越小，越受使用者喜愛。因此若產品 p_i 的第 k 個屬性小於等於使用者偏好 c_i 的第 k 個屬性（即 $p_i[k] \le c_i[k]$），那我們說 c_i 的使用者群"滿意" p_i 的第 k 個屬性值。我們說產品 p_i **滿足**(或說 **dominate**)一個使用者需求 c_i，那表示 $p_i[k] \le c_i[k], for\ 1 \le k \le d$。以圖 1 (b)而言，$p_1$ 的每個屬性值都滿足 $p_1[k] \le c_3[k]$，所以我們說產品 p_1 滿足（dominate）使用者偏好 c_3。

$p_i.c$ 是 p_i 製造成本。我們採用[4]的定義。給定一個屬性 A_i 的屬性值 v_i，我們定義一個 *attribute cost function* f_{A_i}，其可以將屬性值 v_i 對應到一個實數（即價錢）。

Definition 1: $f_{A_i}: v_i \to \mathcal{R}$

在本研究中，我們假設 attribute cost function 是 *monotonic function*。也就是說，若 $v_i \le v_j$ (白話來說，就是屬性值 v_i「**優於**」v_j)，那麼 $f_{A_i}(v_i) \ge f_{A_i}(v_j)$。我們認為這是很合理的假設，因為若屬性值 v_i 優於 v_j，那麼對於 v_i 付出的成本，就要高於對於 v_j 付出的成本。

$c(p_i)$是一個 *product cost function*：給定一個產品 p_i，$c(p_i)$回傳 p_i 的製造成本。我們設定 $c(p_i)$「加總」了 p_i 每個屬性值的成本，其定義如下：

Definition 1:

$$p_i.c = c(p_i) = f_{A_1}(p_i[1]) + f_{A_2}(p_i[2]) + \cdots + f_{A_d}(p_i[d]) = \sum_{k=1}^{d} f_{A_k}(p_i[k])$$

$p_i.IS$ 代表產品 p_i 的影響力，也就是 p_i 會有多少個顧客願意來購買此產品。

$cost_U$ 代表了一個產品的生產成本的上限，若產品的生產成本超過 $cost_U$，代表廠商無法由此產品獲取利潤。$cost_L$ 代表了一個產品的生產成本的下限，壓低成本有很高的機會是透過降低商品的規格來達成，而降低產品規格，將連帶影響到產品在市場上的影響力。

給定任意兩個產品 p_i 及 p_j，我們可以定義其 dominate 的關係。我們說 p_i dominates p_j（以 $p_i \prec p_j$ 表示），那麼代表 p_i 所代表產品的影響力，勝過 p_j 所代表產品的影響力，且生產 p_i 的成本，低於生產 p_j 的成本。以數學式來表示話，就是：

Definition 2:

$$p_i \prec p_j \Leftrightarrow p_i.IS \ge p_j.IS \wedge p_i.c \le p_j.c \wedge (p_i.IS \ne p_j.IS \wedge p_i.c \ne p_j.c)$$

明顯地，若一個產品被其他的產品給 dominate，那表示這個產品不具「競爭力」。

一個產品p若無其他的產品可以dominate它，那我們稱 ***p* 為一個 skyline 產品**。其意義為 p 在市場上具有一定的競爭優勢(可能在生產成本上有優勢，不然就是影響力上有優勢，或者兩者皆有之)。

瞭解前述的定義後，我們可以正式地定義本研究所要解決的問題：

問題定義：(在生產成本限制下，尋找所有具競爭力的"產品")。給定一個使用者偏好的集合 $C=\{c_1, c_2,\ldots,c_n\}$，還有一個產品的生產成本限制 $cost_U$、$cost_L$。找到所有產品的集合 $S_k=\{p_1, p_2,\ldots,p_k\}$，使得$\forall p_i, p_j$：$p_i \in S_k$ 且 $p_j \notin S_k$, 則 $p_i \succ p_j$、$cost_L \leq p_i.c \leq cost_U$。

若廠商想要生產一個產品時(例如 Apple 想產生一款 iPhone)，可以使用本研究的方法來找出最具競爭力產品的規格。也就是說在本研究找出來的產品集合 S 中，挑一個產品出來生產。本研究所提出的方法可能包含很多 skyline 產品，可以再依其他無法量化的因素(如品牌忠誠度，手機的外型...等)來決定最後的生產方案。

接下來第四章就要討論如何解決這個問題，以及本研究提出的 Basic Algorithm for finding Skyline with Customer preference and Cost constraint (BASCC) Algorithm。

四、在生產成本限制下尋找最具競爭力的產品

在這個章節，我們提出了解決在生產成本限制下尋找最具競爭力的產品的方法。根據之前對於問題的敘述，直覺的做法需要掃描空間中所有的資料點，這需要非常大量的運算，因此我們改善了直覺的作法，所以我們在 section A 中提出一個基本的想法。在 section B 中，我們觀察到一些不必要的計算，並套用在 section A 的想法中，我們提出了 basic algorithm for finding Skyline with Customer preference and Cost constraint with pruning (SCCP)。

A. Product Combination Algorithm (PCA)

直覺的想法是在每個產品有生產成本限制下，找出資料空間中具競爭力的產品(也就是 skyline 的產品)。乍看之下，資料空間中有無限多個點，而每一個點都是一個生產方案。因此只要能評估這無限個點，就可以找出所有的 skyline 產品了。明顯地，這個方法在實務上是不可行的。

另一個想法是由客戶偏好出發。我們發現可以透過「組合客戶偏好」的方式，產生新的產品。我們用圖 4 來說明。首先我們先談如何針對「一位」使用者 c，以最少的成本，生產一個符合 c 要求的產品。接著再將這個想法推廣到多位使用者的情境中。

對於「任何 1 個」客戶偏好 c_i，我們所生產產品的規格，只要落在 c_i 的 dominating region 中，即可滿足 c_i。c_i 的 dominating region 定義為 ci 與 data space 原點所形成的矩形。圖 4 中的虛線框展示了 c_3 的 dominating region。在此矩形中(包含矩形的 boundary)的產品，都可以 dominate c_3，也就是滿足 c_3 的要求。那麼所有滿足 c_i 要求的產品中，何者成本最小呢？落於 c_i dominating region 的 upper-right corner 的產品，也就是產品規格和使用者要求「一模一樣」的產品，就是成本最小，而且可以滿足 c_i 的產品。例如圖 4 中，p_3 及 p_3' 都位於 c_3 的 dominating region 中，所以它們都可以滿足 c_3 的需求。因為 p_3 的規格較 p_3' 差，依照 Definition 1，p_3 有較低的成本。

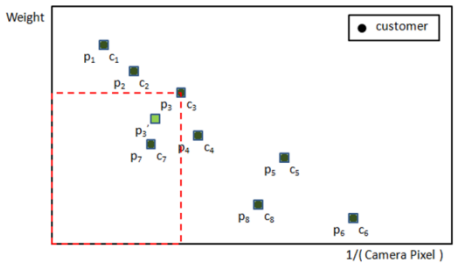


圖 4 對於每個客戶偏好，都可以找到一個可以 dominate 它的產品(以方框表示)。這個產品剛好就與客戶偏好重疊。

圖 4 中共有 8 個客戶偏好(c_1~c_8)，，對於每個使用者偏好，我們可以因此我們可以依照上文的想法，產生 8 個對應的最佳產品 p_1~p_8。

對於「任意 2 個」客戶偏好，我們也可以利用兩個客戶偏好組合而成的 dominating region 來產生一個新產品。如圖 5 所示，紅色框框就是 c_4 及 c_8 組合而成的 dominating region，這個 dominating region 裡面的產品都可以 dominate c_4 及 c_8，由前述可以知道我們可以在 dominating region 的 upper-right corner 產生一個新的產品 p_9，p_9 一定可以滿足 c_4 及 c_8 這兩個客戶而且成本最小。dominating region 的 upper-right corner 產生的方式就是由兩個客戶偏好中選出其最好的屬性值[1]。例如 c_4 提供其第 1 維度的屬性值(以 c_4 [1]表示)，而 c_8 提供第 2 維度的屬性值(以 c_8[2])表示，於是新產品的屬性值 p_9=< c_4 [1], c_8 [2]>。由於資料空間

[1] 請注意，在本論文中，我們假設使用者偏好較小的屬性值。

中有 8 個客戶偏好，因此我們可以產生種C_2^8新產品。

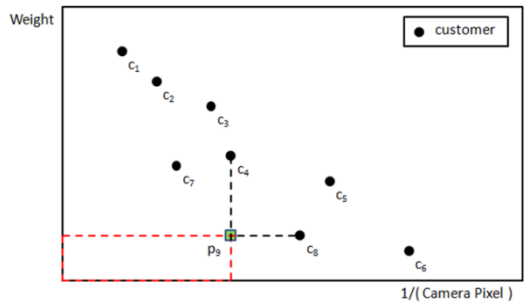


圖 5 任意兩個客戶偏好，可以組合出一個產品。

透過前述的分析，我們可以正式定義我們的產品組合演算法。我們令這個產品組合演算法名 PCA (product combination algorithm)。令$S = \{c_1^s, c_2^s, \dots, c_m^s\}$為一群使用者偏好，且$S \subseteq C$，其中$c_i^s$是 S 中的第 i 個使用者偏好。p 是一個廠商所要生產的產品，$p[i]$是 p 的第 i 維度的屬性值。我們可以透過底下的方式得到$p[i]$：

$$p[i] = min(c_1^s[i], c_2^s[i], \dots, c_m^s[i]), for\ i = 1, 2, \dots, d)$$

以圖 5 為例，$S = \{c_4, c_8\}$，所產生的新產品為 p_9，p_9 的A_1及A_2屬性值可以以底下的方式得到：

$$p_9[1] = min(c_4[1], c_8[1]) = c_4[1]$$

$$p_9[2] = min(c_4[2], c_8[2]) = c_8[2]$$

底下的定理證明了，我們的產品組合演算法所產生的新產品 p，其可以(1)以最少的成本，且(2) dominates S 中所有的使用者偏好。

Theorem 1：令$S = \{c_1^s, c_2^s, \dots, c_m^s\}$為一群使用者偏好，$p$ 是利用演算法 *PCA* 所產生的一個新產品，則 p 可以 dominates S 中所有的使用者偏好。

Proof：因為$p[i] = min(c_1^s[i], c_2^s[i], \dots, c_m^s[i]), for\ i = 1,2 \dots, d$。所以對於 S 中任意的使用者偏好c_i^s，都存在有$p[i] \le c_j^s[i]$ 的情況。因此 p dominates $c_j^s\ for\ j = 1, 2, \dots, m$。

Theorem 2：令$S = \{c_1^s, c_2^s, \dots, c_m^s\}$為一群使用者偏好，$p$ 是利用演算法 PCA 所產生的一個新產品，則 p 是所有可以 dominate S 中所有使用者偏好的產品中，成本最少的。

Proof：因為$p[i] = min(c_1^s[i], c_2^s[i], \dots, c_m^s[i]), for\ i = 1,2 \dots, d$。令 q 是另一個廠商所要生產的產品，$q[i]$是 q 的第 i 維度的屬性值。假設 $c(p) > c(q)$，根據 Definition 1 和 Definition 2 得知，$\exists\, i\ s.t.\ q[i] > p[i]$，因此必定存在任一 c_j^s for $j = 1, 2, \dots, m$ 無法被 q dominate。由此得證，p 是所有可以 dominate S 中所有使用者偏好的產品中，成本最少的。

依照這樣的想法，對於「任意 3 個」客戶偏好，「任意 4 個」客戶偏好…，以及「任意 8 個」客戶偏好，我們都可以組合出一個新的產品。因此我們總共可以產生$C_1^8 + C_2^8 + \dots + C_8^8$種產品，在所有產品的規格都決定後，就可以計算每個產品的 influence score，以及成本。之後再進行 dominance test，即可找出 skyline 的產品。因此若資料空間中有 N 種客戶偏好，我們可以得到整個問題的複雜度為：

$$C_1^N + C_2^N + \dots + C_N^N$$

這樣一來，似乎這個問題會面臨「組合爆炸」的困境。然而，經過仔細的分析，我們發現，其實問題並沒有這麼複雜，section B 就是要討論一些發現。

B. Basic algorithm for finding Skyline with Customer preference and Cost constraint with pruning (SCCP)

由以上的演算法處理後我們得到另外四個觀察現象，讓我們可以設計 pruning 機制，減少所要評估的生產產品數，以增加演算法的效率。以下我們分別介紹這四個觀察，並說明如何利用這個觀察來設計我們的演算法。

Observation 1：令 d 是資料空間的維度，我們發現其實問題的複雜度只有 $C_1^N + C_2^N + \dots + C_d^N$。因為 d 遠小於 N，因此這個問題的解空間，並沒有想像中的大。至於為何問題的複雜度為$C_1^N + C_2^N + \dots + C_d^N$？那是因為在 d 維空間中，「最多」只要組合出 d 個屬性值，就能產生一個新的生產方案，因此就不需要考量剩餘 $C_{d+1}^N + C_{d+2}^N + \dots + C_N^N$ 的組合了。所以我們提出了一個定理。

Theorem 3：令N為使用者偏好的數量，D為 dimension，組合方案只需要考慮$C_1^N + C_2^N + \dots + C_d^N$的組合。

由於頁數限制的問題我們將相關的證明記錄在技術報告中。

我們用圖 5 來說明。c_4、c_5、c_8的組合客戶偏好產生出來的新產品是 p_9，而 c_4、c_8的組合客戶偏好產生出來的新產品也是 p_9。會產生這個情況的原因是因為 $p_9[1]$的值是來自於 $c_4[1]$、$p_9[2]$的值是來自於 $c_8[2]$，c_5的屬性值就沒有用到，所以要產生 p_9 這個產品只需要 c_4 和 c_8 就夠了。

接著我們討論一個基於第一個觀察所設

計出來的 *Basic Algorithm for finding Skyline with Customer preference and Cost constraint* (BASCC) algorithm。BASCC 可以解決在生產成本限制下尋找最具競爭力的產品的方法，但卻會消耗大量的計算資源。接著我們提出一些更好的 pruning 機制，其可以減少 BASCC 的計算量。最後，基於這些 pruning 機制，我們改良了 BASCC 演算法，提出 *basic algorithm for finding Skyline with Customer preference and Cost constraint with pruning* (SCCP)。

根據第一個觀察，我們設計了一個基本的演算法。在枚舉$C_1^N + C_2^N + \cdots + C_d^N$種組合時，每種組合所產生的產品，他的成本若是介於 $cost_U$ 和 $cost_L$ 之間，就放入 candidate set 作 skyline 運算。skyline 的運算是根據產品 p 的兩個屬性，一個是成本 $p.c$，求得方式為根據產品 p 的各個屬性值而來，之前已說明過。另一個屬性是產品的 IS，也就是 $p.IS$，$p.IS$ 可以利用計算 p 的 dominator 得知。既然我們已經知道組合產品 p 的 $p.c$ 和 $p.IS$，所以當我們把組合產品 p 放入 candidate set 時就可以進行 skyline 運算。我們稱此演算法為 BASCC，共分為四個步驟：

Step 1：枚舉$C_1^N + C_2^N + \cdots + C_d^N$種組合。

Step 2：用 PCA 計算出每種組合所產生的產品 p，若 $p.c$ 是介於 $cost_U$ 和 $cost_L$ 之間，則計算 $p.IS$ 後放入 candidate set。

Step 3：在 candidate set 作 skyline 運算。

Step 4：輸出 candidate set 為結果，結果包含每個競爭力產品的成本、影響力、各維度屬性值。

Observation 2： 利用 attribute cost function 為 monotonic function 的特性，可以提前過濾會超出成本限制的銷售方案。

我們定義一個或多個客戶偏好為基準點，意思是接下來要做組合客戶偏好時，都是固定由是基準點的客戶偏好和其他非基準點的客戶偏好作組合。一個基準點產生的產品 p 與另一個客戶偏好 c 作組合產生產品稱為p'，根據 Definition 2 和 monotonic function 的特性可知，用 PCA 產生的產品 p 和p'，$p.c \le p'.c$。

Theorem 4：令$S = \{c_1^s, c_2^s, \ldots, c_m^s\}$為一群客戶偏好且為基準點，$p$為基準點產生的產品，$c_{m+1}$為另一個客戶偏好，假設 p'為p和c_{m+1}組合而成的產品，則$p.c \le p'.c$。

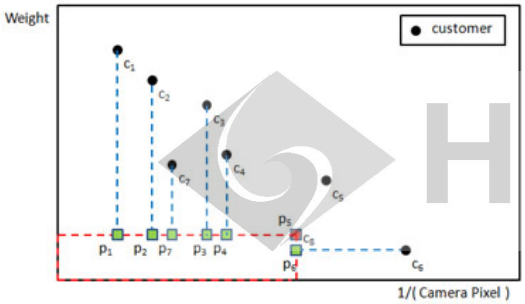


圖 6. 以c_8為基準點和其他客戶偏好組何產生的產品

以圖 6 為例，p_5是由$S = \{c_8\}$所組成，也就是說：

$$p_5[1] = c_8[1]$$
$$p_5[2] = c_8[2]$$

我們可以令 S 為基準點，使用 S 加入其它的客戶偏好，產生新的產品。例如令$S' = S + c_4 = \{c_4, c_8\}$。則將$S'$輸入 PCA 中，可得另一新產品$p_4$：

$$p_4[1] = min(c_4[1], c_8[1]) = c_4[1]$$
$$p_4[2] = min(c_4[2], c_8[2]) = c_8[2]$$

根據 Definition 2 得知，p_4的製造成本必定不小於p_5的製造成本，即$p_4.c \ge p_5.c$。

所以當我們在組合客戶偏好時，每多組何一個客戶偏好，成本只會越來越大或是一樣大，不會變小。也就是說，所組合出產品的成本，會隨著所加入的客戶偏好數以非遞減的方式成長。當組合產品的成本超過 $cost_U$ 的時候，就不需要在組合其他客戶偏好了，因為組合出來的產品成本一定會超過 $cost_U$。根據以上的敘述再結合 BASCC，我們提出了演算法 SCCP。

SCCP 一開始會先利用遞迴函式建構所有的產品。每建立一個新的產品時都會在基準點中加入一個新的 customer preference，以產生一個新的基準點S'。之後利用S'產生新的產品p'。若p'其 cost 介於$cost_U$及$cost_L$間，那麼就將p'加入 candidate set 中並計算 skyline。接著若p'其 cost 小於 $cost_L$則此建構函式再以S'為一個新的基準點，遞迴呼叫自己，再產生一個新的產品。當此建構函式發現新的基準點S'的 cardinality 超過 data space 的 dimensionality d 時，便會終止遞迴。

五、實驗與討論

本章節主要針對我們的演算法做實驗並對這些實驗做相關的分析討論。我們提出的實驗數據是根據三種資料分布：uniform (UN)，correlated (CO)，and anti-correlated (AC)。Table III 表示實驗中的各個參數變化。我們每個實驗都比較三種演算法的時間效率，第一種是 BASCC。第二種是 SCCP。以下是我們實驗的參數，之後會一一做解釋。

表 2 參數設定

Parameter	Ranges
Data size (C)	100,500,**1000**,2000,4000,6000,8000,10000
Dimension (D)	2,3,4,5,6,7,8,9,10,11,12
Cost range	**0.5%**,1%,5%,10%
Cost position of the range	0%,10%,20%,30%,40%,50%,60%,70%,80%,90%,100%

第一個實驗是要實驗成本區間位置對執行時間和具競爭力產品數的影響。BASCC 因為沒有 pruning 的原因，所以在三種資料形態無論區間位置為何都有基本的執行時間，像圖 7,8,9 所示，在三種資料形態在所有區間位置直行時間至少 27 秒左右的下界。在 uniform data (圖 7) SCCP 的執行時間都是在 40%左右的位置開始明顯成長，因為組合數開始急遽增

加的關係；在 80%開始減少，因為組合數開始下降的關係。執行時間 SCCP 不見得都比 BASCC 還要好，因為在超過 80%的時候，每個組合產品的計算在 cost 的計算會多花很多時間，多花的時間已超過 pruning 省下的時間。同理在 correlated data 和 anti-correlated data 都發生相同的情況。在 correlated data (圖 8)中，接近 50~60%的位置資料點最多，因此可能的組合數也越多，在 60%的位置執行時間達到最高，無論是 BASCC 還是 SCCP。在圖 9 中可以發現，具競爭力的產品數和執行時間沒有直接關係，而是跟資料分布有關係。在 uniform data 中，在 80%附近的位置較有可能產生較多的產品，因為可能的組合最多。在 correlated data 中，因為在 50%附近的資料點最多，所以產生的競爭力產品數才會較多，反之，離 50%越遠的位置，競爭力產品數越少。在 anti-correlated data 中，跟 uniform data 類似，是在 80%附近較有可能產生較多的產品，不一樣的地方是，40%以前由於很少資料點，所以很少有產品出現。由於頁數限制之因素，我們將其餘詳細的實驗呈現在技術報告中。

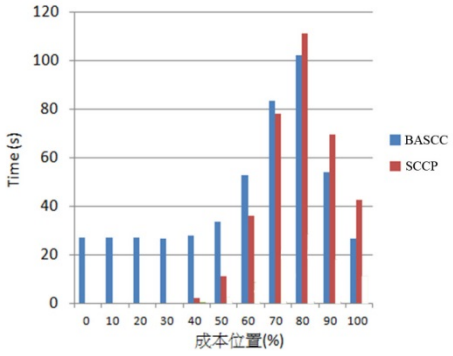


圖 7. Uniform data, N=1000,D=3

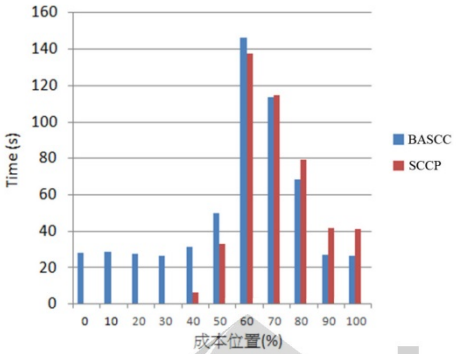
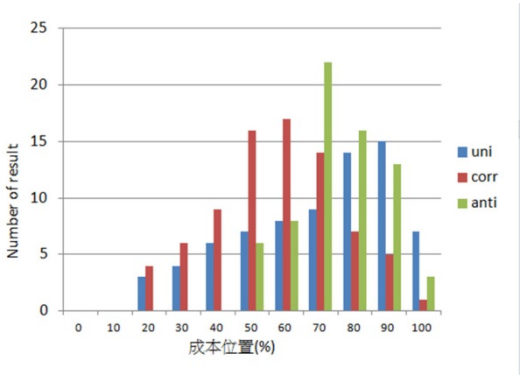


圖 8. Correlated data, N=1000,D=3

六、結論

在實驗中，我們可以發現到當 $cost_U$ - $cost_L$ 過大時，有可能產生非常多的具競爭力產品。除了提醒使用者避免選擇的 $cost_L$、$cost_U$ 差距過大外，我們可以請使用者訂定一個 k 值，這個 k 值代表最後會回傳最多 k 個具競爭力產品，若本身回傳的產品數量就小於 k 個，則結果不變。至於要怎麼取 k 個具競爭力產品則是我們要探討的問題。我們可以假設產品成本界於 $cost_U$ 和 $cost_L$ 之間都是使用者可以接受的範圍，那可以想像使用者會想讓成本的影響力(IS)越大越好。所以，在所有具競爭力的產品當中，應該選出產品影響力(IS)最大的 k 個產品為最佳選擇。因此，我們可以在演算法中作一些修正，例如說加了一個 pruning 法則，我們訂定一個 threshold 值，代表 IS 第 k 大的產品 IS 值，當我們的組合產品未達此 threshold 值時，就



不用放入 candidate set 作 skyline 運算。

圖 9. N=1000,D=3

致謝

本文為作者群之科技部專題研究計劃(102-2221-E-309-011- 與 100-2221-E-006-249-MY3 ）之部分研究成果，作者群特此感謝科技部的研究經費補助。

參考文獻

[1] C. Li, B. C. Ooi, A. K. H. Tung, S. Wang. DADA: A data cube for dominant relationship analysis. In SIGMOD 2006.

[2] Z. Zhang, L. Lakshmanan, and A. K.H. Tung, "On domination game analysis for microeconomic data mining," In TKDD 2009.

[3] Y.Peng, C.-W. Wong, “Finding Competitive price,” In the proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Orlando, Florida on November 5-8, 2013

[4] H Lu, C. S. Jensen, “Upgrading uncompetitive products economically,” In ICDE 2012.

[5] Q. Wan, R. C.-W. Wong, and Y. Peng, “Finding top-k profitable products,” In Proceedings of ICDE, pp. 1055-1066, 2011.

[6] Chen-Yi Lin, Jia-Ling Koh, and A. L. P. Chen, “Determining k-most demanding products with the maximum expected number of total customers,” IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 8, pp. 1732-1747, Aug. 2013.

[7] Q. Wan, R. C.-W. Wong, I. F. Ilyas, M. T. Ozsu, and Y. Peng, “Creating competitive products,” In VLDB, 2009.

[8] Kleinbega, J., Papadimitriou, C. and Raghavan, P., “A microeconomic view of data mining,” In Data Mining Knowledge Discovery, Vol. 2(4), pp. 311-322, 1998.

[9] Osborne, M. J. and Rubinstein, A., “A course in game theory,” The MIT press, 1994.

[10] S. Borzsonyi, D. Kossmann, and K. Stocker. The skyline operator. In ICDE, 2001.

[11] K.-L, Tan, P. Eng, and B. Ooi. Efficient progressive skyline computation. In VLDB, 2001.

[12] 史考特・安東尼，馬克・強生，約瑟夫・辛費爾德以及伊莉莎白・亞特曼，“創新者的成長指南”，天下雜誌出版，2010。

[13] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang, “Selecting stars: the k most representative skyline operator,” in Proceedings of the 23rd International Conference on Data Engineering, pp. 86-95, 2007.