基於MapReduce技術之時間區段性的網頁瀏覽樣式偏好度探勘

賴鈺芬　　葉介山　　林彩雯
靜宜大學資訊管理學系
ukliop0930@gmail.com, jsyeh@pu.edu.tw, s9871011@gm.pu.edu.tw

摘要

為了有效分析使用者的瀏覽樣式，本研究結合網站使用探勘與利潤探勘的概念提出網頁瀏覽樣式之偏好度探勘演算法。另一方面，網頁可能依照時間、季節或月份的變化而有不同的熱門程度。本研究進一步提出不同時間區段的網頁瀏覽樣式之偏好度探勘的模型與演算法，加強時間變化的概念去找出更精確的網頁項目集。本研究並採用雲端 Hadoop 平台環境，以 MapReduce 技術提升網頁瀏覽樣式利潤偏好度探勘的效能。

實驗結果顯示，本研究所提出基於 MapReduce 技術的網頁瀏覽樣式利潤偏好度探勘的演算法，能產生更好的執行效能；不同時間區段的網頁瀏覽樣式之偏好度探勘能找出更精確的網頁項目集。

關鍵詞：網頁探勘、利潤探勘、雲端運算、MapReduce、資料探勘。

Abstract

In order to effectively analyze the user's web navigation patterns, this study adopts the concept of preference mining proposed by Chen and Yeh, and introduces a new definition of web page's weight for the preference mining on web navigation patterns. Moreover, the popularity of web pages may vary during different seasons or time periods. Therefore, this study also proposes a novel model of the preference mining with time periods on web navigation patterns to strengthen the concept of time and discover more accurate web itemsets. This study implements the proposed algorithm on the cloud computing platform, Hadoop, by utilizing MapReduce technology.

The experimental results show that the proposed algorithm on Hadoop platform is more efficient and scalable than on a single machine platform. The preference mining with time periods on web navigation patterns can discover more accurate frequent web itemsets.

Keywords: Data mining, Utility mining, Cloud computing, MapReduce, Web mining.

1. 緒論

網站使用度探勘(Web Usage Mining)[10]主要是藉由網頁記錄檔做分析，在有用的網頁中找出隱而未覺的瀏覽樣式。2007 年，Zhou 等學者[13]提出把利潤探勘演算法從商品的計算改變成網頁記錄檔的分析，透過兩階段利潤探勘方式，可以找尋通過門檻值的高利潤路徑瀏覽模式。2010 年，Chen 和 Yeh 學者**錯誤! 找不到參照來源。**提出網頁瀏覽樣式之利潤偏好度探勘(Utility Preference Mining)，做法上與傳統不同的是加入選擇偏好度與時間偏好度來定義網頁的利潤價值。透過探勘的結果，網頁伺服器可以預測接下來被存取的網頁，並且提供網頁設計師了解更多有意義的資訊。

此外，商品的熱賣程度通常跟季節或是月份有很大的關係，例如:暖暖包通常在冬季較為熱賣，或者是冰淇淋在夏天比較暢銷。但是以往的演算法可能無法精確的找出高利潤產品，例如：冰淇淋在夏季產生很高的利潤，可是在這一年中的利潤，卻是沒有超過門檻值的，所以冰淇淋這個商品會被修剪。根據這個想法，2011 年 Lan 等學者[6]提出以時區性的利潤探勘找出有超過門檻值的高利潤項目，該兩階段演算法，於第一階段找出不同時區超過門檻值的候選項目集，於第二階段以第一階段有過門檻值的項目集計算實際利潤，如果有超過門檻值就屬於高利潤商品組合。相同地，這樣的現象也常發生在網站上，網頁瀏覽次數會因為季節或是月份而有所變化，例如：購物網站中，在 5-9 月夏季的商品瀏覽次數的頻率是比較高的。所以 Supriya 和 Indira[9]所提出以二階段演算法使用分時區的概念計算網頁紀錄，這樣可以提供網頁設計師以分時區性的結果去針對網頁加以改善。

此外，隨著資訊化的發展，許多領域都面臨著

大規模的資料量，如今，有效率的處理大量資料變成主要的議題。分散式系統是使用許多低成本的電腦一起運算資料去進行探勘，雲端運算[11]屬於分散式系統的一種，它是基於網際網路的運算方式，可以處理大量資料探勘，解決單一機器的效能問題。

本研究提出一個新的網頁瀏覽樣式利潤偏好度探勘模型，接著提出不同時間區段的網頁瀏覽樣式之偏好度探勘的演算法能更精準的找出項目集。。此外針對資料量越來越來大的環境下，本研究將網頁探勘結合雲端 Hadoop 平台環境[5]撰寫 MapReduce **錯誤! 找不到參照來源。**的程式架構，去執行分散式運算，並且使演算法達到最好的效能。

在實驗部分，分為真實資料與人工資料，真實資料採用靜宜大學招生組的網頁紀錄檔，經過多次的參數設定，可以明顯看出執行時間的差異。實驗結果顯示本研究所提出的演算法能更精確找出有用的項目集並且降低執行時間。

2. 相關文獻

2.1 雲端運算

雲端運算[11]是一種基於網際網路運算的方式，透過網路的方式共享軟硬體資源，屬於可計算大型資料量的分散式運算，許多資料探勘的運用已經漸漸使用雲端環境去做運算[1][4][14]。2009 年美國國家標準局(NIST)對於雲端運算提出定義包含：五大基礎特徵(Characteristics)、四個部屬模型(Deployment Models)以及三個服務模式(Service Models)。

Hadoop[5]是目前最常見的雲端運算平台，可以提供大量資料的分散式運算環境。Hadoop 包括提供大量儲存空間的 Hadoop Distributed File System(HDFS)[1]、分散式環境運算的 MapReduce **錯誤! 找不到參照來源。**以及類似 BigTable 分散式資料庫的 Hbase。

- MapReduce(圖)

這是一個程式開發模型，讓使用者可以簡單的撰寫程式，快速處理大量的資料。在運算的時候要拆成 Map(映射)以及 Reduce(化簡)兩個部分。首先將資料切割成不相關的區塊，切割後的區塊會被系統轉換成一組組的 key 值與 value 值分別傳給不同的 Mapper 處理，經過運算結果後輸出一組組的 key 值與 value 值。merge 階段會把 Map 的結果加以做排序把相同的 Key 值歸類為一群。接著再進入 Reduce 將結果整合，最後將整體的結果輸出。

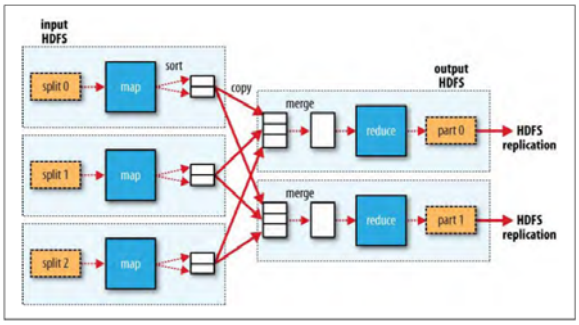

圖 1 MapReduce 架構[11]

2.2 利潤探勘

傳統的關聯法則只考慮到購買的商品是否有出現並不能完全反映出項目集的重要性，並且利潤探勘演算法也不符合向下閉合的概念，例如：假設門檻值為 40，{A,D}=42、{A}=20、{D}=80，此時{D}跟{A,D}過門檻值，但是{A}沒過門檻值，此情況無法應用在傳統的關連法則。因此在 2004 年 Yao 等學者提出利潤探勘演算法[12]，考慮到購買商品的數量以及購買商品的利潤，從中找出高利潤項目集。

二階段演算法

Liu 和 Liao 學者提出二階段演算法[7]有效的減少候選項目集的產生，讓執行的時間大幅減少。

2.3 網頁瀏覽樣式之利潤偏好度探勘

在網頁使用探勘中，以往的演算法裡網頁利的計算都是以網頁點擊的次數乘上停留的時間[8]。在 2010 年 Chen 和 Yeh 學者提出選擇偏好度及時間偏好度的概念去計算網頁利潤。選擇偏好度是網頁點擊次數在此 IP 點擊所有網頁次數的比例，時間偏好度是把網頁停留的時間除上 60，讓數值變小，使計算上速度提升，最後網頁的利潤概念是以

「選擇偏好度×時間偏好度×網頁點擊次數」。此演算法的步驟為首先對網頁紀錄檔作前處理，把不要的資料給刪除。接著去計算每筆資料中個別網頁的選擇偏好度、時間偏好度以及偏好利潤，再以二階段的方式去高估第一階段的利潤值，最後把第一階段產生出來的項目集找出實際利潤在比對實際上是否有超過門檻值。

3. 研究方法

為了有效分析使用者的瀏覽樣式重要性，本研究針對網頁的重要性給予不同的權重值，並且結合網站使用探勘與利潤探勘的概念提出網頁瀏覽樣式之偏好度探勘演算法以及延伸這個演算法架構針對不同時間區段去做網頁瀏覽樣式之偏好度探勘。

3.1 網頁瀏覽樣式之偏好度探勘

符號與定義

定義 1. $I = \{URL_1, URL_2, \cdots, URL_m\}$為所有網頁集合，共有 m 個網頁。

定義 2. $D = \{T_1, T_2, \cdots T_n\}$為經過資料預處理後的瀏覽紀錄的集合，其中 T_i為單一 IP 的瀏覽紀錄，共有 n 筆資料。

定義 3. $o(URL_p, T_q)$為T_q裡連結URL_p的次數。

定義 4. $s(URL_p)$為URL_p網址的權重利潤值。

定義 5. $sp(URL_p, T_q)$為URL_p在T_q裡所拜訪網頁的次數比例，稱為**選擇偏好度**。

$$sp(URL_p, T_q) = \frac{o(URL_p, T_q)}{\left(\sum_{URL_p \in T_q} o(URL_r, T_q)\right)} \quad (7)$$

定義 6. $tp(URL_p, T_q)$為T_q裡所拜訪URL_p網頁的停留時間，稱為**時間偏好度。**

$$tp(URL_p, T_q) = t(URL_p, T_q)/60 \quad (8)$$

定義 7. $pu(URL_p, T_q)$為URL_p在T_q裡的**偏好利潤** (preference utility)

$$pu(URL_p, T_q) = s(URL_p) \times sp(URL_p, T_q) \times tp(URL_p, T_q) \quad (9)$$

定義 8. 給定網頁項目集 $X = \{URL_1, URL_2 \cdots, URL_k\}$，$X \subseteq T_q$， $pu(X, T_q)$為T_q交易裡包含網頁項目集 X 的利潤加總。

$$pu(X, T_q) = \sum_{URL_p \in X} pu(URL_p, T_q) \quad (10)$$

定義 9. 給定網頁項目集X， $pu(X)$ 為資料庫 D 裡包含網頁項目集 X 的利潤加總

$$pu(X) = \sum_{T_q \in D} \sum_{URL_p \in X} pu(URL_p, T_q) \quad (11)$$

定義 10. $tpu(T_q)$為T_q所以點選的網頁利潤總和。

$$tpu(T_q) = \sum_{URL_p \in T_q} pu(URL_p, T_q) \quad (12)$$

定義 11. 給定門檻值λ，$minutil$定義如下

$$minutil = \lambda \times \sum_{T_q \in D} tpu(T_q) \quad (13)$$

定義 12. 給定網頁項目集 X， $tpu(X)$為資料庫 D 裡的包含網頁項目集 X 的T_q利潤加總

$$tpu(X) = \sum_{X \subseteq T_q \in D} tpu(T_q) \quad (14)$$

定義 13. 給定網頁項目集 X， X在資料庫D中的偏好利潤為

$$pu(X) = \sum_{X \subseteq T_q \in D} pu(X, T_q) \quad (15)$$

定義 14. 如果$tpu(X)$大於或等於 $minutil$，則項目集 X 稱為**高交易偏好利潤**(High Transactional Preference Utility, HTPU) 網頁項目集。

定義 15. 如果$pu(X)$大於或等於 $minutil$，則項目集 X 稱為**高偏好利潤**(High Preference Utility, HPU) 網頁項目集。

網頁瀏覽樣式之偏好度探勘模型的定義如下：

給定網頁集合$I = \{URL_1, URL_2, \cdots, URL_m\}$、瀏覽紀錄的集合 $D = \{T_1, T_2, \cdots, T_n\}$及門檻值$\lambda$，網頁瀏覽樣式之偏好度探勘模型可挖掘網頁集合$I$中所有高偏好利潤的網頁項目集。

結合雲端運算

近幾年來雲端運算的興起，使得龐大資料量的運算可以得到好的運算效能，本研究結合 Hadoop 平台，以 MapReduce 為基本架構去執行平行化計算，針對可以平行處理的部分寫成 Map 程式，分析出來的結果透過 Reduce 程式彙整，最後輸出其

結果。針對每個 MapReduce 步驟中的 Key 值與 Value 值之定義與說明如下：

Step 1. 針對 weblog 資料進行前處理，篩選出符合的資料。

步驟 1 以 Map 為例，Key 值為日期,IP 而 Value 值為網址,時間，Reduce 就是把相同 Key 的日期,IP 的資料彙整成同一筆資料。

Step 2. 計算每個 IP 的停留時間，並且算出時間偏好度(停留時間/60)，以及此 IP 所點擊網頁的總次數。

步驟 2 是以 IP 和網頁當作 Key 值的結合去計算網頁在此 IP 的時間偏好度以及所點擊的次數。

Step 3. 計算每個 IP 點擊網頁的總次數。

步驟 3 是為了選擇偏好度計算的時候需要此 IP 的所有點擊網頁次數，所以以 IP 當作 Key 去計算每個 IP 的點擊網頁加總。

Step 4. 計算每個 IP 的網頁選擇偏好度(點擊網頁次數/此 IP 總共點擊網頁的次數)，以及計算偏好利潤值(選擇偏好度×時間偏好度×網頁權重值)，最後計算 tpu(此 IP 所有的偏好利潤加總)。

步驟 4 計算完美個網頁的選擇偏好度之後，再去計算網頁的偏好利潤值，因為第一階段計算網頁偏好利潤值(比實際高估利潤值)的需要所以在 Reduce 的時候去計算每個 IP 的總偏好利潤值。

Step 5. 計算門檻值，擁有此網頁 IP 的 TPU 加總(第一階段)，以及此網頁的實際利潤(第二階段)。

步驟 5 為了計算門檻值的設定，加總資料庫裡所有 IP 的網頁利潤值，並且去計算第一階段演算法所要求的偏好利潤值以及當第一階段通過之後要在第二階段裡比較的實際偏好利潤值。

Step 6. 做二階段的篩選，找出通過門檻值的項目集。並且去生成下一階段的候選項目集，直到找不出下一階段候選項目集為止。

Step 7. 先去篩選符合項目集的紀錄，並且回到步驟 6 去做運算。

步驟 7 當有生成候選項目集的時候，進入 Map 先找尋此 IP 是否有此項目集，有的話就把它們的值抓取出來，再由 Reduce 彙整出結果存在本機端再到第六步驟去做運算。

演算法運算停止的方式如下：當判別第一階段找出來的項目集為 1 個以下的時候這樣無法生成候選項目集所以停止運算、當第一階段找出來的項目集為 2 個以上，但是無法生成候選項目集時也停止運算，最後是當資料庫裡找不到候選項目集組合的時候停止運算。

3.2 不同時間區段的網頁瀏覽樣式之偏好度探勘

此節將說明本研究如何延伸網頁瀏覽樣式之偏好度探勘演算法，以針對不同時間區段的資料進行分析，並介紹分時區性網頁瀏覽樣式之偏好度探勘的架構以及範例說明，最後也提出運用在雲端環境上的演算法。

符號與定義

定義 1. $I = \{URL_1, URL_2, \cdots, URL_m\}$為所有網頁集合，共有 m 個網頁。

定義 2. $D = P_1 \cup P_2 \cup \ldots \cup P_k$為經過資料預處理後的瀏覽紀錄的集合，其中 P_i為互不交集的時間區段， $P_i = \{T_{i,1}, T_{i,2}, \cdots, T_{i,n_i}\}$，$T_{i,j}$為單一 IP 的瀏覽紀錄。

定義 3. $a(URL_p, P_q)$=1 代表URL_p網址會出現於P_q時間區段的瀏覽紀錄。

$a(URL_p, P_q)$=0 代表URL_p網址完全不會出現於P_q時間區段的瀏覽紀錄。

定義 4. 給定網頁項目集 $X = \{URL_1, URL_2 \cdots, URL_k\}$，$a(X, P_q) = \prod_{URL_p \in X} a(URL_p, P_q)$。$a(X, P_q) = 1$代表網頁項目集$X$中的所有$URL_p$網址皆出現於$P_q$時間區間。

定義 5. $o(URL_p, T_{i,j})$為$T_{i.j}$裡連結URL_p的次數。依此定義，如果$a(URL_p, P_q)$=0 則 $o(URL_p, T_{q,j})$必定為 0，其中$T_{q,j} \in P_q$。

定義 6. $s(URL_p)$為URL_p網址的權重利潤值。

定義 7. $sp(URL_p, T_{i,j})$為URL_p在$T_{i,j}$裡所拜訪網頁的次數比例，稱為**選擇偏好度**。

$$sp(URL_p, T_{i,j}) = \frac{o(URL_p, T_{i,j})}{\left(\sum_{URL_p \in T_{i,j}} o(URL_r, T_{i,j})\right)} \quad (16)$$

定義 8. $tp(URL_p, T_{i,j})$為$T_{i,j}$裡所拜訪URL_p網頁的停留時間，稱為**時間偏好度。**

$$tp(URL_p, T_{i,j}) = t(URL_p, T_{i,j})/60 \quad (17)$$

定義 9. $pu(URL_p, T_{i,j})$為URL_p在$T_{i,j}$裡的**偏好利潤** (preference utility)

$$pu(URL_p, T_{i,j}) = s(URL_p) \times sp(URL_p, T_{i,j}) \times tp(URL_p, T_{i,j}) \quad (18)$$

定義 10.給定網頁項目集 $X = \{URL_1, URL_2 \cdots, URL_k\}$，$X \subseteq T_{i,j}$，$pu(X, T_{i,j})$為$T_{i,j}$裡包含網頁項目集 X 的利潤加總。

$$pu(X, T_{i,j}) = \sum_{URL_p \in X} pu(URL_p, T_{i,j}) \quad (19)$$

定義 11.給定網頁項目集X，$pu(X, P_q)$ 為時間區段裡P_q包含網頁項目集 X 的利潤加總

$$pu(X, P_q) = \sum_{T_{p,j} \in P_q} \sum_{URL_p \in X} pu(URL_p, T_{p,j}) \quad (20)$$

定義 12. $tpu(T_{i,j})$為$T_{i,j}$所有點選網頁的利潤加總。

$$tpu(T_{i,j}) = \sum_{URL_p \in T_{i,j}} pu(URL_p, T_{i,j}) \quad (21)$$

定義 13. 給定門檻值λ，時間區段裡 P_q內的最小門檻值$minutil_{P_q}$定義如下

$$minutil_{P_q} = \lambda \times \sum_{T_{i,j} \in P_q} tpu(T_{i,j}) \quad (22)$$

定義 14. 給定網頁項目集X，$tpu(X)$為資料庫 D 裡的包含網頁項目集 X 的$T_{i,j}$利潤加總

$$tpu(X) = \sum_{X \subseteq T_{i,j} \in D} tpu(T_{i,j}) \quad (23)$$

定義 15. 給定網頁項目集 X，X在資料庫D中的偏好利潤為

$$pu(X) = \sum_{X \subseteq T_{i,j} \in D} pu(X, T_{i,j}) \quad (24)$$

定義 16. 如果 $tpu(X)$大於或等於$\sum_{\forall P_q, a(X,P_q)=1} minutil_{P_q}$，即$X$所出現的所有時間區段內的最小門檻值$minutil_{P_q}$之加總，則項目集 X 稱為時間區段性的**高交易偏好利潤**(High Transactional Preference Utility, HTPU) 網頁項目集。

定義 17. 如果 $pu(X)$大於或等於$\sum_{\forall P_q, a(X,P_q)=1} minutil_{P_q}$，則項目集 X 稱為**時間區段性的高偏好利潤** (High Preference Utility, HPU) 網頁項目集。

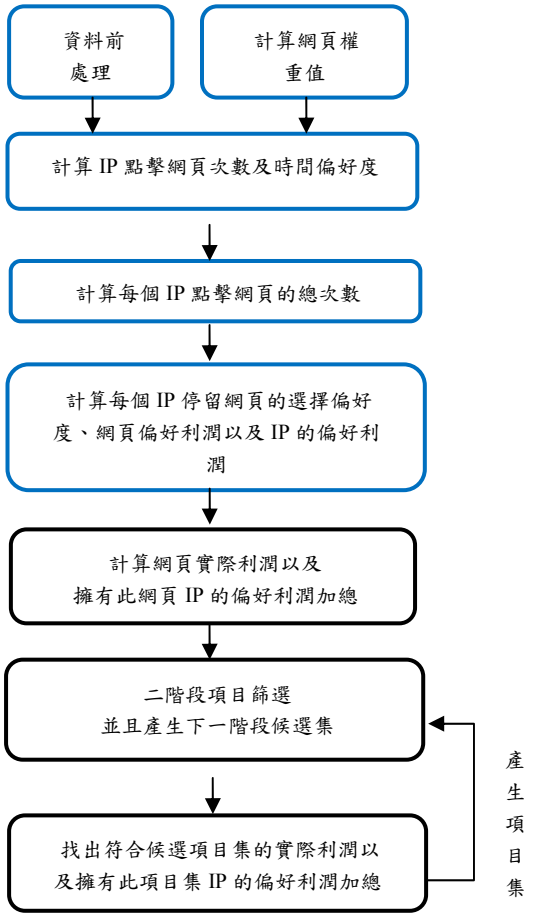

圖 2 分段式的雲端程式架構

時間區段性的網頁瀏覽樣式之偏好度探勘模型的定義如下：

給定網頁集合$I = \{URL_1, URL_2, \cdots, URL_m\}$、瀏覽紀錄的集合 $D = P_1 \cup P_2 \cup \ldots \cup P_k$，其中 P_i為互不交集的時間區段，$P_i = \{T_{i,1}, T_{i,2}, \cdots, T_{i,n_i}\}$，$T_{i,j}$為單

一 IP 的瀏覽紀錄，以及門檻值λ，時間區段性的網頁瀏覽樣式之偏好度探勘模型可挖掘網頁集合*I*中所有時間區段性的高偏好利潤的網頁項目集。

結合雲端運算

針對不同時段的演算法，本研究也結合 Hadoop 平台，以 MapReduce 架構去執行平行化程式。圖為屬於分段式的雲端程式架構，藍色框框是以分時段去做計算。

4. 實驗分析

4.1 實驗環境

本研究所使用的實驗環境為 Dell 伺服器，雙 AMD Opteron(tm) Processor 4180 2.6 GHz、六核心 CPU、16G RAM。Hadoop 平台是以 Dell 伺服器切割成 8 台虛擬機器，每一台虛擬機器的規格有 1 個 CPU 、1 G memory。實驗分別運行在 1 台、2 台、4 台以及 8 台虛擬機器下，以觀測各別的執行效能。本研究採用之 Hadoop 版本為 0.20.2，程式碼以 Java 語言撰寫。

4.2 實驗數據

本實驗採用兩種不同的數據，一種為靜宜大學招生組網頁紀錄檔，另一種為人工生成方式產生資料，以下將針對這兩個數據做介紹：

靜宜大學招生組網頁紀錄檔

網頁紀錄檔包含了很多資訊(例如:時間、連結網頁、通訊協定...等)，針對網頁紀錄檔探勘的時候要對資料做前處理。本研究分別針對 10 天、20 天以及 31 天的網頁記錄數據進行實驗，相關數據資訊如表。此數據將使用在網頁瀏覽樣式之偏好度探勘。

圖顯示不同資料量在不同虛擬環境的執行時間。當資料量為 705MB 的時候，可以看出來其實 8 台虛擬機器的時間很接近 4 台虛擬機器，這個結

表 1 招生組網頁數據統計

日期	數據大小	數據筆數	IP 數	網頁數(包含 pdf,doc, ….)
5/1~5/10	705MB	7999180	9351	1828
5/1~5/20	1.39GB	16379425	15706	2290
5/1~5/31	2.29GB	26631003	21208	2487

人工生成方式產生資料

因為傳統的 IBM Quest Synthetic Data Generator 沒有辦法生成周期性資料，所以本研究以 Java 程式語言撰寫人工生成的資料，輸入的參數包含資料筆數、每筆交易資料最多可連結的網頁(網頁可重複)、網頁個數(以.php 為結尾)、網頁停留的時間以及網頁所屬的時段，皆以亂數方式形成。

人工生成產生兩個文字檔案，分別是資料的內容以及網頁所屬的時段。內容文件檔命名的方式為"TxxxLxxxIxxxVxxxPxxxPtxxx.txt"，T 為資料筆數、L 為每筆資料的最大長度、I 為項目個數、V 為最大網頁停留時間、P 時段、Pt 為每個時段的資料筆數。

4.3 實驗結果

本研究進行三種不同的實驗，第一種實驗為網頁瀏覽樣式之偏好度探勘，其中網頁利潤預設為 1。第二種實驗跟第一種實驗不同的地方是在網頁利潤以 PageRank 分析出來的權重。第三種實驗為不同時間區段的網頁瀏覽樣式之偏好度探勘，其中網頁利潤預設為 1。

實驗一

首先本研究假設每個網頁的利潤值為 1 以門檻值為 0.1、0.075、0.05 以及 0.025 去觀察在不同實驗環境上所執行的效能。**錯誤! 找不到參照來源。**顯示在不同門檻值第一階段以及第二階段超過門檻值的項目集個數。

果是因為資料量比較小，所以效能上的比較沒有太大的差異。

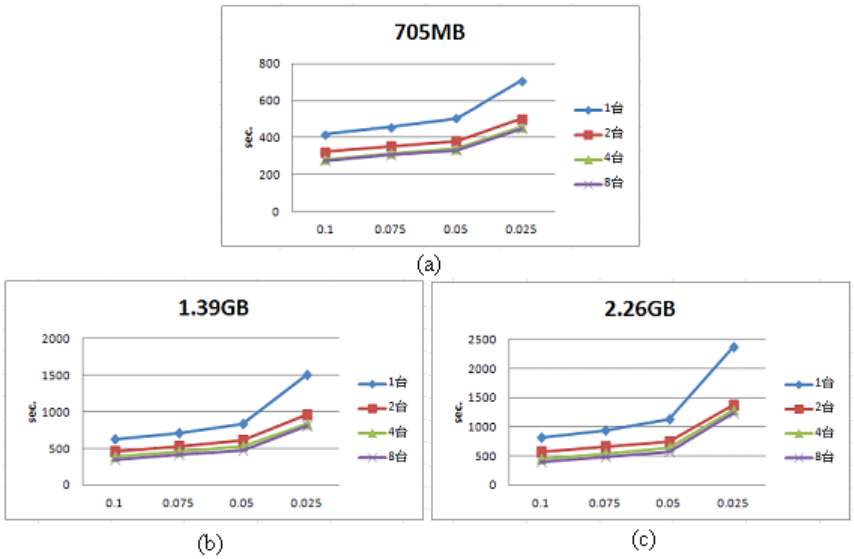

圖 3 不同資料量在不同虛擬機器的執行時間

為了凸顯 Hadoop 的特性，本研究擴大資料量，將資料量分成 1 個月、2 個月以及 3 個月， 圖顯示當資料量越大的時候在不同虛擬機器之間執行的時間差異越大，凸顯 Hadoop 適合在大量的資料下運行。

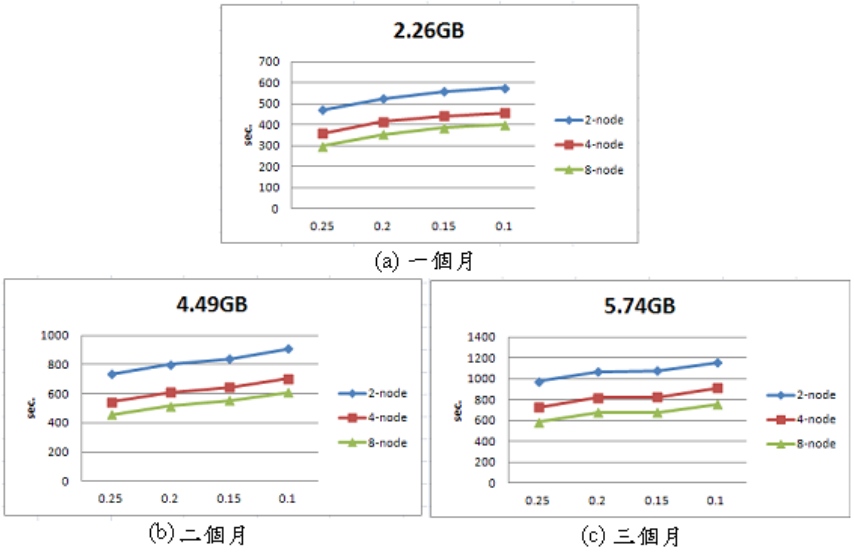

圖 4 一、二、三個月的資料量分別在不同虛擬機器的執行時間

實驗二

針對不同時間區段的網頁瀏覽樣式之偏好度探勘，本研究將資料量降低變成資料量 50 萬筆、每筆交易最大長度為 10、項目集 100、網頁利潤值 1-10、停留時間 1-3000 秒以及分成 4 個時間區段，門檻值設為 0.009。簡稱為「T500000L10I100V3000P4」，表顯示資料數據統計。然而現實生活中商品有不同時段的分類，在這些時段內也會有一些商品組合是比較多人購買的。

表 2 生成資料數據統計

時間區段	網頁個數	資料筆數	平均長度
一	44	110931	4.69
二	41	111008	4.67
三	50	111085	4.73
四	44	110907	4.70

表 3 2-itemset 通過項目集個數

	第一階段	第二階段
不同時段演算法	2830	0
不分期演算法	3	0

表 兩種演算法在 2-itemset 通過項目集個數。第一階段在不同時段演算法中產生 2830 個項目集，原因是不分時段的門檻值界定是以時間區段出現的總和，所以降低的門檻值;在不分期演算法所產生 3 個項目集原因是門檻值以全部時段的總和運算，造成門檻值提高的情況下，有些屬於時間區段的高利潤項目集沒有通過門檻值。第二階段兩者演算法接沒有項目集通過門檻值成為高利潤項目集的原因是資料生成時所停留的時間和網頁出現次數都是平均分散，網頁組合機率較低很難找出通過門檻值的組合。

不同時段的演算法確實可以找出更多有意義的項目集。可以運用在網路商城上面，依照不同的時間區段所產生出來的項目集設計網頁，例如：在不同時間區段的時候，可以把首頁放入一些高利潤項目，使消費者瀏覽網站時更快速的瀏覽當季熱門商品。未來要克服的問題是資料量過大如何運用 Hadoop 去產生出更好的效能。

5. 結論

本研究提出網頁瀏覽樣式之偏好度探勘。利用 Chen 和 Yeh 學者所提出的選擇偏好度及時間偏好度在加上 Pagerank 的網頁運算權重值。這三個變數的相乘做為利潤值的計算。可以更準確的看出網頁的重要性。網頁利潤探勘演算法結合雲端技術解決當分析的資料過多所造成執行時間過長的問題。本研究之實驗結果顯示：實驗一的執行效能確實有提升許多；而實驗二與實驗一比較之下，在適當定義網頁權重值的情況更能看出高利潤項目集的重要性。

因應不同網頁有其被瀏覽的熱門時段或季節，本研究提出不同時間區段的網頁瀏覽樣式之偏好度探勘的演算法，針對不同時間區段的資料進行分析，實驗結果顯示：不同時間區段演算法可以比網頁利潤探勘演算法更加準確並找出更多有意義的網頁項目集，該實驗的結果也可以提供給網站設計師或是網站管理者做網頁維護或修改的參考。

參考文獻

[1] D. Borthakur, “The Hadoop distributed file system: architecture and design,” http://lucene.apache.org/hadoop/hdfs.html, 2012.

[2] Y. C. Chen and J. S. Yeh, “Preference utility mining of web navigation patterns,” IET International Conference on Frontier Computing. Theory, Technologies & Applications (CP568) Taichung, Taiwan, pp.49-54, 2010.

[3] J. Ekanayake , S. Pallickara, and G. Fox “MapReduce for data intensive scientific analyses,” Proceedings of the 4th IEEE International Conference on eScience, pp.277-284, 2008.

[4] L. Huang, X. W. Wang, Y. D. Zhai, and B. Yang,, “Extraction of user profile based on the hadoop framework,” Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), pp.5301-5306, 2009.

[5] Hadoop, http://hadoop.apache.org, 2012.

[6] G. C. Lan, T. P. Hong, and V. S. Tseng, “Discovery of high utility itemsets from on-shelf time periods of products,” Expert Systems with Application, vol. 38, no. 5, pp. 5851-5857, 2011.

[7] Y. Liu, W. K. Liao, and A. Choudhary, “A fast high utility itemsets mining algorithm,” Proceedings of the 1st International Conference on Utility-Based Data Mining, pp. 90-99, 2005.

[8] E. Poovammal and P. Cigith, “Mining web path traversals based on generation of FP tree with utility” CCSEIT 2011, CCIS 204, pp. 94-100, 2011.

[9] Jyotsna Supriya .P and D.N.V.S.L.S. Indira, “A modern two–phased system for on-shelf utility mining on web transactions” International Journal of Engineering Science and Technology(IJEST) vol. 3 no.7, pp. 5796-5801, 2011.

[10] S. S. Tseng, “Data mining,” ISBN 10-9-574-42236-4, Flag Publishing Co., Taipei, 2005.

[11] T. White, “Hadoop: The Definitive Guide.”, ISBN 0-596-52197-9, O’Reilly Press, 2009.

[12] H. Yao, H. J. Hamilton, and C. J. Butz, “A foundational approach to mining itemset utilities from databases,” Proceedings of the 3rd SIAM International Conference on Data Mining, pp. 482-486, 2004.

[13] L. Zhou, Y. Liu, J. Wang, and Y. Shi, “Utility-based web path traversal pattern mining,” Proceedings of the 7th IEEE International conference on Data Mining Workshops, pp. 373–378, 2007.

[14] W. Zhao, H. Ma, and Q. He, “Parallel K-means clustering based on MapReduce,” Proceedings of the 1st International Conference on Cloud Computing (CloudCom), pp. 674-679, 2009.