分散式決策樹隱私權防護

李建億*　李育強**　林政穎*
國立台南大學數位學習科技學系*　南台科技大學資工系**
leeci@mail.nutn.edu.tw
lyc002@mail.stut.edu.tw
m09505004@stumail.nutn.edu.tw

摘要

由於資訊化時代的來臨，資料呈現巨量的成長。在如此巨大的資料庫中，可能隱含著對企業有用的訊息，因此必須藉助資料探勘來挖掘出有利的資訊以協助公司進行決策。另外由於跨國企業、區域性等環境因素，使得現今資料的分佈多以分散式架構來部署，在面對資料分散、協同合作等環境下，這也讓分散式相關議題成為近年來相當熱門的議題之一。企業為了創造共同利益，可能會採取策略聯盟等合作方式，而不同公司合作時，並非所有資料都能夠完全公開，因此合作的同時也必須保護企業獨一無二的資料不被其他公司所取得，所以分散式決策樹探勘的過程中必須達到保護隱私的目的。因此，我們提出了一個多單位合作之決策樹隱私權防護的方法，此方法採以 C4.5 為基礎並利用垂直屬性切割在水平資料庫環境下進行運作，此方法主要是保護不同單位間的規則不被其他單位所得知，同時又能達到準確無誤差的共同探勘結果。

關鍵字：隱私權防護、分散式決策樹探勘、C4.5

Abstract

As the recent development of the computer science, the data quantity of enterprise database increases rapidly. To extract the usefulness information from huge databases, many efficient data mining technologies have been applied. In recent years, the data mining tools are more and more powerful, and the risk of privacy leak has become an urgent problem. Privacy preserving data mining is a relatively new research area in data mining and knowledge discovery. In a common situation, databases are distributed among several organizations who would like to cooperate mining to extract global knowledge, but each party needs prevent it's privacy not directly sharing the data. Therefore, this study presents an algorithm for privacy preserving distributed decision tree based on C4.5. While this has been done for horizontally partitioned data, this study presents an algorithm for vertically partitioned attributes. Each site computes a portion of Data, and then they exchange the result to each other. The goal of this paper is to obtain correct data mining results and preserve the privacy of each site.

Keywords：Privacy Preserving、Distributed Decision Tree、C4.5

1. 前言

隨著資訊科技日新月異，資訊量因此急速的成長，在這些資料中除了本身的內容外，更可能潛藏著重要的意涵。因此，如何在如此大量的資料中，發掘潛藏有用資訊，進而將資料轉化為商機或是有助益的訊息，以提供決策人員參考，即是一個十分重要的議題，資料探勘正是為處理分析大量資料所發展出的學門，因而成為近年來相當熱門的研究議題之一[4]。另外，資料探勘的應用也延伸到許多其他不同的領域，包含有醫學界、銀行業、百貨業等等，其中更是有許多的成功案例。甚至在麻省理工學院 2001 年 1 月與 2 月份所發表的 Technology Review 中也預測，資料探勘是未來改變世界的十大新興科技之一[7]。

資料探勘發展至今存在有許多的探勘工具以及方法，但這些方法或技術隨著科技的迅速進步也顯得有些不足，甚至衍生出新的問題。尤其在處理巨大的資料內容時更顯不足。因此，必須要更為有效率之探勘演算法才能夠挖掘出資料中所代表的意義。另外，在資料的儲存上，越來越多的資料都改以數位化的方式儲存，這也讓資料的欄位變得更加的複雜，對資料探勘而言，這往往是個高維度的處理，而這樣也會對探勘的效率上有一定的影響。對於時代的變遷，探勘的環境也不侷限於單一資料庫，可能因為企業的部署而存在有更為複雜的系統環境，例如分散式或異質的資料庫等。因此，現今的技術仍然多致力於探勘技術以及效能的加強，期望能夠提升探勘的效益，但探勘技術越強大，對於資料庫中所隱含的資訊也越容易被挖掘。有些規則攸關企業機密或是個人隱私等訊息，則可能在探勘過程中，洩漏了隱私或機密，造成企業或個人的重大影響。

接著於第二節中，針對資料探勘的技術與決策樹方法中相關的隱私權防護的發展進行說明探討。接著第三節開始介紹本論文的方法與技術應用於分散式決策樹隱私權防護中。第四節針對本論文所提的方法進行安全性等相關分析，證明確實能夠保護各個單位間的隱私權。最後是本論文的總結以及未來的研究方向。

2. 文獻探討

資料探勘可稱之為資料庫或資料集中知識的發掘（Knowledge Discovery in Databases，簡稱KDD）[4]，也就是可以從一個大型資料庫中萃取出有用的知識。而資料探勘的技術也包含許多類型，包含有關聯法則(Association rule)、群集(Clustering)、分類(Classification)等方法。關聯法則(Association Rule)主要是找出最常發生或出現的事件的規則，最典型的案例就是購物籃分析，藉由顧客的購物明細，來分析客戶的購物特性，以得知顧客的購物習性，進而推薦符合顧客興趣的商品，提升顧客的購買機會。群集(Clustering)則是根據分群的原則將資料分群，讓群組裡的資料相似度達最大，群組間的相似度達最小，而群集分析的目的主要是將群集與群集間的差異找出來。分類(Classification)則是從已知的類別物件集合中，依據資料的屬性建立分類模式，來描述物件屬性與類別的關係。接著透過未經分類的資料或新的資料依此模式進行預測。

在分類的方法中，最常使用的是決策樹，並利用決策樹來進行分類。決策樹是一種常用於分類與預測的演算法，決策樹是一種語意樹(Semantic Tree)，其結構上有根(Root)、節點(Node)以及樹葉(Leaf)等結構，其中節點代表某項屬性的測試，節點的分支則代表某項屬性的測試結果，最後樹葉節點代表歸類樣本的類別。決策樹可對資料庫訓練資料觀察其過去的行為或歷史資料，以將過去的資料轉化成為一套可用來分類及預測的模型，藉以推估其未來資料所可能的結果，同時也透過樹狀結構表示，使其結果更具視覺化的效用，讓使用者能夠更容易了解，另外也能夠以 If-then 的方式來表達樹狀結構所代表的意涵。目前決策樹也有許多方法，包含有 CHAID、ID3、CART、C4.5、C5.0 等。

根據[1]隱私權的定義中，隱私權可分為獨立個體以及整體組織兩大類型隱私權。獨立個體隱私表示個別資料的內容；整體組織隱私則表示整體資料中所代表的意涵。對於獨立個體隱私或是整體組織隱私都各有其重要性，進行探勘時都必須將這些因素考慮其中。對於隱私權資料探勘的研究上早在 1996 年 Chris Clifton[2]就提出了對於資料探勘中安全性以及隱私權的問題。他們文中也提到可能可以保護隱私資料探勘的方式並也說明其中的困難之處，其中主要有下列幾種，包含存取限制(Limiting Access)、模糊資料(Fuzz the data)、排除不需要的群組(Eliminate unnecessary groupings)、增加資料(Augment the data)、審核(Audit)。

對於隱私權防護的議題上，Verykios 等人[8]於 2004 年針對隱私防護資料探勘的研究進行整理及分類，分別從資料分佈（Data Distribution）、資料修改（Data Modification）、資料探勘演算法（Data Mining Algorithm）、資料或規則隱藏（Data or Rule hiding）以及隱私防護（Privacy Preservation）五個層面進行分析。在隱私權防護的議題中，其隱私防護資料探勘的方法上往往不存在有唯一的解決方案，反而必須依據實際的情況、所使用的探勘演算法或是資料上的限制等不同因素進行考量，進行個別的設計與處理，也因此造成使用同樣類型的演算法但卻有數種的解決方法。對於隱私權防護的議題上，本論文主要針對水平分散式環境下之決策樹隱私權防護進行研究。

在分散式決策樹隱私權防護上，過去的文獻許多都以探討垂直資料庫為主，在水平資料上相對不足。底下我們將介紹過去提出的水平分散式環境下之文獻。首先在 F. Emekci 等人[3]所提的文獻中，他們使用了以 ID3 為基礎的隱私權防護之分散式探勘演算法，其中主要使用了多項式資料交換的技術來達到隱私權防護的效果，雖然他們使用了 Shamir's secret sharing 技術，能夠提供較高的資料隱密性，但在多單位時會造成高次方的運算，增加運算的複雜度，也因此增加了通訊成本。此外，他們的方法所能達到的準確率也隨著參與節點數而有所變化。

另外在 Xiao 等人所提的方法[10]中，使用了以 ID3 為基礎的隱私權防護方法。主要採用 SMC(Secure Multiparty Computation)安全多單位運算的方式，他們主要提出了 xLogx 與 FindMax 這兩個協定，這兩個協定主要採以安全多單位運算的方式將資料進行隱藏。Xiao 等人也於 2006 年提出另一篇決策樹隱私權防護的方法[9]，此方法是以 C4.5 為基礎的分散式隱私權防護方法，採方法也是採以安全多單位運算的方式來達到隱私權防護的目的，但其方法會耗費大量的通訊成本以及時間。

目前的探勘演算法大多執著於效能上的研究，對於隱私權防護並未做過多的考量。因此，如何在有效的探勘環境下同時兼顧隱私權的防護則是現在與未來的重要議題之一。另外在探勘的準確率上，兼顧隱私權的同時往往會犧牲掉準確性，但在某些情況下，準確率是無法被犧牲的，例如醫學上的研究，些許誤差則可能造成嚴重的影響，因此必須具有相當精準的正確率。

為了解決分散式隱私權防護探勘的問題，我們提出了一個在分散式的環境下的隱私權防護探勘方法，其中主要考量讓不同的單位間進行決策訊息的交換，以創造出合作時的最大利益，同時設計其隱私權解決方案，保護企業本身的資訊，不被其他單位所得知。另外，我們也兼顧準確率的考量，在隱私權保護的情況下也能夠達到精準的正確性。

3. 分散式決策樹隱私權防護

3.1 水平分散式資料散佈

為了解決分散式決策樹隱私權防護的問題，底下我們提出了在水平分散式環境下的C4.5隱私權防護方法，針對水平資料庫上的隱私權的問題進行考量並對其做安全性防護。主要環境示意圖如下圖1，在分散式架構下可能存在有許多單位要進行合作探勘，如底下存在有P個單位，每個單位的資料庫皆存在有相同或相似的屬性。在此環境下，如何兼顧準確性與隱私權防護，則是我們最為主要的目標。

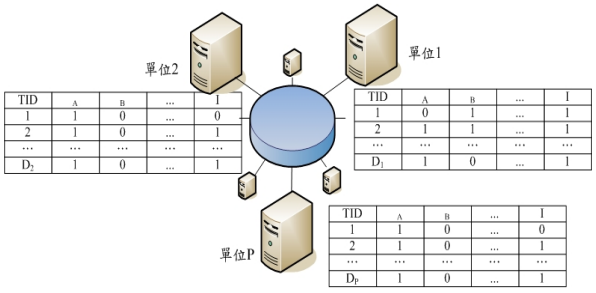

圖 1 水平分散式資料散佈

3.2 資料交換與運作架構

在水平的資料散佈環境下，進行資料交換時，採以垂直式的屬性為單位，建立資料交換的模式。如下圖2所示，我們將屬性進行分配，每個單位運算特定的屬性，如單位1運算A屬性、單位2運算B屬性等等。其他單位則將非本身單位所運算的資料傳送至該屬性運算的單位，如單位1傳送屬性B的資料給單位2、單位2傳送屬性A的資料給單位1等等，同時運算該特定屬性的單位也接收其他單位所傳來的資料，接著每個單位將運算完的結果進行交換，完成交換後會找出最合適的屬性作為切點，並傳送給其他單位，接著再進行下一階段的運算。透過此運作架構，我們不需要交換全部資料，即可建構出最後的決策樹。

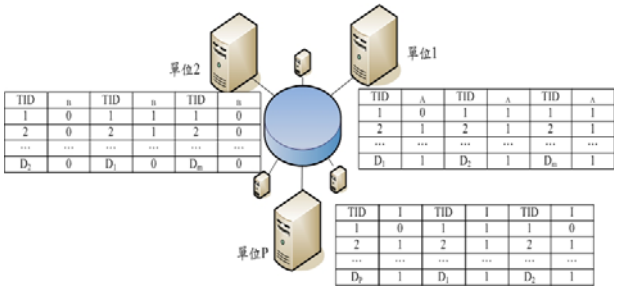

圖2 垂直資料運算架構

3.3垂直屬性資料交換

本論文主要是針對分散式決策樹之隱私權進行研究及方法設計，並以C4.5[6]為基礎，在水平資料散佈情況下運行，同時，各個分散單位進行資料交換時必須相互傳遞各屬性的資料，即每個單位屬性的垂直切割。將不同屬性的資料交由不同的單位進行共同運算，並將運算完的結果進行交換，以建構出最後的整體決策樹。

演算法: 水平資料分佈之垂直屬性資料交換。
輸入: 各參與單位之資料集 輸出: 所有單位合作探勘之決策樹 開始 1.與其他單位溝通，並依據資料比例，決定本身所需進行運算的屬性。 2.對非自己所運算的其餘屬性進行資料統計透過廣播傳送給其他單位。 3.將收到其他單位所傳來的同樣屬性資料做統計，並運算此屬性之 Gain Ratio。 4.將步驟 3 所得的 Gain Ratio 廣播傳給其他單位。 5.接收其他單位步驟 4 傳來的 Gain Ratio 值，並決定何屬性作為分割之屬性。 6.(1)根據步驟 5 的屬性作為分類依據，對於其下每一分支剩餘的非自己運算之屬性統計值傳給其他單位。 (2)其他單位在步驟 5 所得的分類屬性下，將收到的屬性資料進行統計，計算此屬性之 Gain Ratio。 (3)將步驟(2)之 Gain Ratio 傳送給其他單位。 (4)決定何屬性為該分支底下之分類節點。 7.重複 6 之程序，直到資料無法繼續進行運算分割。 輸出

在我們所提出的方法中，為了證明確實能夠達到精準的正確性，底下我們將證明本研究所提之分散式運算方法與資料集中時所建構之決策樹相同。

證明：假設存在有 P 個單位、K 個屬性、每個屬性各有 V 個分割。對任一單位 i，存在有屬性 j，屬性 j 有 s 個分類，其單位間屬性的資料表示為 D_i^{js}，(i∈P，j∈K，s∈V)。資料加總情況如下：

資料集中時，各單位資料 $D_{1=}\sum_{j=1}^{k}\sum_{s=1}^{v}D_1^{js}$ 、$D_{2=}\sum_{j=1}^{k}\sum_{s=1}^{v}D_2^{js}$ ... $D_{p=}\sum_{j=1}^{k}\sum_{s=1}^{v}D_p^{js}$ ，整體資料 $D=D_1+D_2、...、+D_p=\sum_{j=1}^{k}\sum_{s=1}^{v}D_1^{js}+\sum_{j=1}^{k}\sum_{s=1}^{v}D_2^{js}+、...、+\sum_{j=1}^{k}\sum_{s=1}^{v}D_p^{js}=\sum_{i=1}^{p}\sum_{j=1}^{k}\sum_{s=1}^{v}D_i^{js}$ (1)

另外，對任一屬性 j 而言，其屬性資料運算加總如下：資料加總 $D^j=D_1^{js}+D_2^{js}$ 、...、$D_p^{js}=\sum_{s=1}^{v}D_1^{js}+\sum_{s=1}^{v}D_2^{js}+.....+\sum_{s=1}^{v}D_p^{js}=\sum_{i=1}^{p}\sum_{s=1}^{v}D_i^{js}$ 。(2)

對於 $\forall j$ 之總和為 $\sum_{j=1}^{k}\sum_{i=1}^{p}\sum_{s=1}^{v} D_i^{js}$ (3)

設任一單位 i，其資料為 $D_i = \sum_{j=1}^{k}\sum_{s=1}^{v} D_i^{js}$，對於 $\forall i$ 之總和為 $\sum_{i=1}^{p}\sum_{j=1}^{k}\sum_{s=1}^{v} D_i^{js}$ (4)

由公式(1)及(3)(4)得知資料集中與屬性分散式加總以及分散式單位資料加總之運算結果相同。

在分散式環境下，對任一單位 i，運算屬性 j，s 個分類，進行資料傳遞時，會傳送 D_i^{js} 的資料給其他單位，接著也會收到其他單位所傳來的資料，單位 i 總共收到的資料為 $D_1^{js} + D_2^{js} + 、\cdots、+ D_{i-1}^{js}$，因此對任一單位 i，包含其本身資料總和為 $(D_1^{js} + D_2^{js} 、 \ldots 、 D_{i-1}^{js} + D_{i+1}^{js} 、 \cdots 、 D_p^{js}) + D_i^{js} = \sum_{i=1}^{p}\sum_{s=1}^{v} D_i^{js}$ (5)

由上述可得知(2)與(5)結果相同，表示本研究所提之方法，其運算結果會與資料集中時所運算之結果相同。另外在 Entropy 公式中，當屬性資料加總相同時，則其運算結果也相同。因此我們也可得知其最後運算結果相同，其所建構出的樹狀也相同。

3.4 樹狀的修剪

根據 Error Rate 的意思[7]主要說明若某葉節點被歸類在某一類別時，則此葉節點的錯誤率如下所示，「該葉節點中不屬於此類別的資料筆數 E_v」除以「該葉節點中的資料筆數 E」，若是計算出來的錯誤率大於或等於使用者所定義的門檻 e 時，則進行修剪。透過此方法能夠計算出每個節點的錯誤率，進而判斷哪些節點造成決策樹錯誤率的升高，再將這些節點進行修剪。由上述定義中可知，樹的修剪是以個別屬性為單位進行修剪，透過此屬性產生的子樹來運算其錯誤率。在本研究所提的方法中，也是以屬性作為運算的單位，因此在樹狀的切割上不需要再做額外的資訊交換，而是由原本運算該屬性的單位負責即可。即使決策樹修剪過後，仍還是可以得到相同的樹狀。

4.安全性與通訊分析

4.1 推測樹狀的情況

1.推測全部可能的樹狀

過去的文獻[5]中已明確指出要得到最佳的二元樹是一個 NP-Complete 問題，而決策樹的樹狀會依據屬性的分類進行樹狀的分支，而分支的數量可能存在有 N 個分支，因此，要找出一個最為合適的最佳解，其複雜度更勝二元樹許多，因此要直接建構所有可能的樹狀仍然是一個 NP-Complete 問題。如底下表 1 案例就存在有超過 $4x10^6$ 種可能。

表 1 Golf 範例資料集

Outlook	Temp	Humidity	Windy	Class
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

由共同的樹狀加上本身的樹狀來推測其他單位的樹狀的情況中，在只得到本身及整體樹狀的情形下，推測出其他單位樹狀的情形最小的機會就是共同探勘所產生整體樹狀的結果，最大機會則是依據所得到的資訊來決定其最大可能，包含有整體樹狀中的子樹、樹狀所出現的屬性等因素。一般情況，若沒其他額外的資訊，則此情形與知道本身及整體的樹狀，然後推測其他單位的樹狀相似，不管三者樹狀皆不同或是相似，其機率皆是相同的。而在三個以上的單位，若沒其他額外的資料，更無法進行正確的推測，此外樹狀的變化由於分支的數量、屬性的個數等因素，存在有太多的變化，因此無法由一致性的表示方式來說明推測樹狀的可能。底下我們針對推測樹狀的可能情況進行說明：

(1)若是整體樹狀出現的屬性等於本身樹狀所出現的屬性或是包含本身樹狀的所有屬性：此情況被推測出的最大機會主要根據整體樹狀及其子樹的變化決定。若整體決策樹已經包含三個單位各自決策樹所出現的屬性，其中某屬性皆未出現，則推測其他單位決策樹的可能性，就會由整體決策樹的所有可能子樹來決定，而非全部屬性。在兩個單位時，由本身樹狀以及整體樹狀推測出對方單位的可能性較高，因為其整體的決策樹可能都包含雙方的子樹。若三個單位以上時，若要單獨推測某單位的樹狀，可能必須事先考慮其他單位的可能情形，因此存在有更多的變化因素，若要推測出其他單位決策樹的機會仍須視整體決策樹的所有以及其可能的子樹來決定。若要直接從某單位與整體決策樹來推測，則具有相當的困難度。

(2)若是整體決策樹出現的屬性少於本身樹狀所出現的屬性或部分屬性相異於本身樹狀：在此情況

下，由於整體決策樹出現的屬性少於或部分不同於本身樹狀的屬性，因此要由整體決策樹的資訊來推測推測其他單位的可能性，則顯的更為複雜。除了整體決策樹外，更須考量其他單位所可能出現的屬性，因此被推測出的最大機會就可能為整體樹狀。

4.2 藉由得到的資料推測樹狀

在分散式環境下，進行聯合探勘進行決策樹建構時，必須與其他單位進行資料交換或傳送相關訊息，才能夠建構出完整的決策樹。在建構決策樹的過程中，有某些必須交換的內容，包含有 Gain Ratio 值、屬性的資料筆數，最後建構完成時，每個協同探勘單位都可得到整體樹狀訊息。

進行決策樹建構時，首先就必須先決定出哪個屬性擁有最大的 Gain Ratio 值，因此必須將本身資料筆數與其他單位交換，當運算完成後每個單位再將運算該屬性的 Gain Ratio 值傳送給其他單位。在此情況下，參與運算的每個單位皆能夠得到該層屬性的 Gain Ratio 值，當知道其他單位傳來的 Gain Ratio 值後，其他單位可能利用本身的資料來推測其他單位的情況，在此情形下主要有兩種類型的場景，分別是兩個單位時的情況以及三個以上單位時的情況。由於一開始有進行整體資料筆數的交換，因此其他單位可能藉由排列組合來推測可能的值，底下我們將分析在上述情況下安全性的資訊。

(1)兩單位合作時：在兩個單位的情況下，底下我們將說明其中一方推測另一方的資料變化總數。若要透過 Gain Ratio 與屬性的資料筆數來猜測對方的資料特徵，會根據傳的資料筆數而有不同的排列組合。假設存在有一屬性 A，其分類共有 a 種結果，以及存在有一目標屬性 Y，其目標類別為 Y_1、Y_2.....Y_k，共 k 種結果，其中一方得到整體的 Gain Ratio 以及整體目標類別分類資料總筆數 D_{y1}、D_{y2}.....D_{yk}。當某一方進行推測時，主要是經由估算出對方屬性的可能變化量，因此透過排列組合方式將收到的資料筆數再加上本身的資料來進行排列組合，分配給屬性的分類結果。經由上述可得知可能透過將整體資料排列組合來推測其可能的數值，其情形如下：

$$H^{a}_{D_{yk}} = C^{a+D_{yk}-1}_{D_{yk}}$$

(2)三單位以上合作時：在三單位以上合作的情況下，底下我們將說明其中一方推測另一方的變化量所可能的情況，若要單純推測某一單位的數值將比兩單位的情況下更為複雜，因為收到的資料中就包含了其他單位的總和因此要推測某一單位時，也必須同時考量其他單位的情況。假設存在有 P 個單位進行聯合探勘，且存在一屬性 A，其分類共有 a 種結果，以及存在有一目標屬性 Y，其目標類別為 Y_1、Y_2.....Y_k，共 k 種結果，其中一方得到整體的 Gain Ratio 以及目標類別資料的總筆數 D_{y1}、D_{y2}.....D_{yk}，經由上述可能透過將總資料排列組合來推測其可能的數值，其情形如下，存在有幾個單位 P 則有幾個 H。

$$\sum_{r=0}^{Yr} H^{a}_{r} * H^{a}_{r1} * \ldots * H^{a}_{rx} \text{，} (r+r_1+\ldots+r_x = D_{yk})$$

4.3 惡意單位

一般進行合作探勘時，往往預設都將其他單位視為可信任的單位，但這中間也可能存在有惡意單位，以詐取其他單位的隱私資料。底下我們將探討本論文所提出的方法，若是存在有惡意單位時的情況，其個別單位資料是否會遭到惡意節點所得知。

1.若是傳遞假的統計值

假設某單位傳給單位 1 屬性 I 的值是偽造的，如 D_2 與 D_2^j，透過下列公式可得知，透過其偽造數據，則產生錯誤的結果，即使惡意單位取得此錯誤的值，則此惡意單位也無法推測出正確的數值。

Entropy(All)=

$$-\sum_{j=1}^{k}\left(\frac{D_1^j + D_2^j + \ldots + D_p^j}{D_1 + D_2 + \ldots + D_p}\right)\log_2\left(\frac{D_1^j + D_2^j + \ldots + D_p^j}{D_1 + D_2 + \ldots + D_p}\right)$$

Entropy(I)=

$$-\sum_{s=1}^{v}\left(\frac{D_1^{js} + D_2^{js} + \ldots + D_p^{js}}{D_1 + D_2 + \ldots + D_p}\right) I\left(D_1^{js} + D_2^{js} + \ldots + D_p^{js}\right)$$

2.若是傳送偽造的 Gain Ratio 值

C4.5 決策樹主要根據 Gain Ratio 做為分類依據，若傳遞偽造的 Gain Ratio 值，惡意單位也無法獲得正確的分類結果，更可能使得樹狀結構出現錯誤的分類，因此也無法從中獲得其他單位的資料。

4.4 通訊量分析

本論文方法對於通訊的成本上，主要依據下面幾個變量，分別是參與單位的數量、屬性的個數、分支的數量以及樹狀的深度。假設一共有 P 個單位參與聯合探勘，存在有 I 屬性，樹狀的深度為 R，其中 I≧R，每個屬性傳送統計後數據的通訊量為 W，比較 Gain Ratio 大小的通訊量為 T。另外我們傳送的是統計後的數據，而非整體資料，因此在通訊量上明顯的比傳統傳遞整體資料內容的方法，降低了大量通訊的成本。資料的傳遞上我們是採以廣播的方式傳送給其他單位，總共傳送的次數共 P(P-1) 次。在屬性的部份，於根節點時所有的屬性都會參與運算，因此在根節點時運算屬性的個數共為 I 個，第二層則運算共 I-1 個、第三層則運算共 I-2 個，第 R 層則運算共 R-1 個，如圖 3 所示。

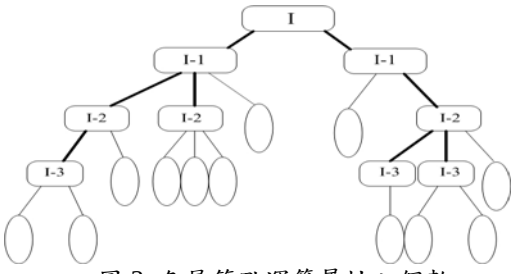

圖 3 各層節點運算屬性之個數

1. 若是參與單位的數量大於或等於屬性的數量

若是存在有 P=I，則每個單位運算一個屬性，若是 P>I，則有些單位不需要運算屬性，但這些屬性也必須傳送統計後的數據其他單位進行運算，因此仍須進行資料的交換，底下我們就以最大通訊量的假設來說明通訊時的成本，也就是計算 P=I 時的通訊成本。例如存在有 15 個節點、10 個屬性，其中有 5 個節點不需要運算，但我們以最大假設的通訊量：每個單位都要運算來看待，因此其運算量近似於 P=I 的情況。以下則是各層的通訊量：

根節點：[P*(P-1)*$\frac{I}{P}$*W]+ [P*(P-1)*$\frac{I}{P}$*T]，其中 W>T，因此最大通訊量可表示為 2[P*(P-1)*$\frac{I}{P}$*W]=2[(P-1)*I*W]

第二層節點：[(P*(P-1)*$\frac{I-1}{P}$*W)+(P*(P-1)*$\frac{I-1}{P}$*T)]*根節點分支的數量，也可表示為 2[(P*(P-1)*$\frac{I-1}{P}$*W)= 2[(P-1)*(I-1)*W] *根節點分支數。

第 R 層節點：[(P*(P-1)*$\frac{I-(R-1)}{P}$*W)+(P*(P-1)*$\frac{I-(R-1)}{P}$*T)]*第 R-1 層分支數量，也可表示為 2[(P*(P-1)*$\frac{I-(R-1)}{P}$*W)= 2[(P-1)*(I-(R-1))*W] *第 R-1 層分支數量

由上述，可推論出由根節點到第 R 層節點總共最大的通訊量為 O(2RW(P-1)(I-R+1)*最大分之數量)，其中 W 與 T 皆為相當小的通訊量。

2. 若是屬性的數量小於參與單位的數量

若是存在有 P<I，則有些單位需要運算多個屬性，若 P<I 則 I/P 存在有倍數關係，雖然存在有倍數關係，但其根本仍然是處理共 I 個屬性的通訊量，通訊的多寡仍取決於運算單位 P。如 35 個屬性、10 個單位則取最大運算值：每個單位最大處理 4 倍通訊量，但其屬性個數總共為 35 個，最大的通訊量仍然是運算 35 個屬性時的通訊量。因此其通訊成本也與 P=I 時的情況相似。

5. 結論

為了保護分散式決策樹探勘中，參與節點的隱私，我們提出了水平資料分佈之垂直屬性資料交換機制，來提供參與節點的隱私保護。本方法中除了具有隱私權防護的能力外，更能夠在建構決策樹的過程中達到正確無誤差的結果。本論文主要是針對水平資料庫進行考量，未來期望能朝向同時存在有水平與垂直混合式的資料庫進行。另外，目前本論文所提的方法主要是對於個別單位規則隱私的保護，未來也期望朝向兼具原始資料內容的保護。

6.參考文獻

[1] C. Clifton, M. Kantarcioglu, and J. Vaidya, "Defining Privacy for Data Mining," in *National Science Foundation Workshop on Next Generation Data Mining* Baltimore, MD, 2002, pp. 126-133.

[2] C. Clifton and D. Marks, "Security and Privacy Implications of Data Mining," in *Proceedings of the ACM SIGMOD Workshop on Data Mining and Knowledge Discovery*, Montreal, Canada, 1996, pp. 15-19.

[3] F. Emekci, O. D. Sahin, D. Agrawal, and A. E. Abbadi, "Privacy Preserving Decision Tree Learning over Multiple Parties," *Data & Knowledge Engineering* vol. 63, pp. 348-361, November 2007.

[4] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, second ed.: Morgan Kaufmann 2006.

[5] L. Hyafil and R. L. Rivest, "Constructing Optimal Binary Trees is NP-complete," *Information Processing Letters*, vol. 5, pp. 15-17, 1976.

[6] J. R. Quinlan, *C4.5: programs for machine learning*: Morgan Kaufmann Publishers Inc. , 1993.

[7] M. T. Review, "10 Emerging Technologies That Will Change the World, http://www.globalfuture.com/mit-trends2001.htm, " 2001.

[8] E. B. Vassilios S. Verykios , Igor Nai Fovino , Loredana Parasiliti Provenzaand Yucel Saygin and Yannis Theodoridis, "State-of-the-Art in Privacy Preserving Data Mining," *ACM SIGMOD Record*, vol. 33, pp. 50-57, March 2004.

[9] M.-J. Xiao, K. Han, L.-S. Huang, and J.-Y. Li, "Privacy Preserving C4.5 Algorithm over Horizontally Partitioned Data," in *Proceedings of the Fifth International Conference on Grid and Cooperative Computing* 2006, pp. 78-85.

[10] M.-J. Xiao, L.-S. Huang, Y.-L. Luo, and H. Shen, "Privacy Preserving ID3 Algorithm over Horizontally Partitioned Data," in *Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies(PDCAT05)*, 2005, pp. 239-243.