行政院國家科學委員會專題研究計畫成果報告

決策樹轉換成法則時移除不相關的值及遺失分支問題探討

計畫編號：NSC 88-2213-E-032-001

執行期限：87 年 8 月 1 日至 88 年 7 月 31 日

主持人：蔣定安　　執行機構:私立淡江大學資訊工程學系

一、中文摘要

為了避免在分類法則中檢查一些不必要或不相關的條件，決策樹中不相關值問題因此產生。我們提出一個演算法來移除從決策樹轉換成分類法則過程中所產生的不相關條件。

從樹的建構方式來看，我們的演算法不僅適合像 ID3 這類的非遞增推導演算法，而且也適合像 ID5R 這類的遞增推導演算法，因此我們的演算法僅和決策樹的語意有關，在花少許的計算成本之下，我們的演算法可以和現行任何的演算法整合在一起，此外我們的演算法亦可解決遺失分支問題。

關鍵詞：資料發掘、決策樹、不相關值、分類法則

Abstract

To avoid checking unnecessary or irrelevant conditions of the classification rules, the irrelevant values problem of the decision tree is addressed. We propose an algorithm to remove irrelevant conditions of the classification rules in the process of converting the decision tree to the classification rules according to the semantics of the decision tree. Our algorithm is not only suitable to the tree constructed by non-incremental tree-induction algorithms, such as ID3, but also to that constructed by incremental tree-induction algorithms, such as ID5R. Since our algorithm depends only on the semantics of the decision tree, our algorithm can be integrated into any existing tree-construction algorithm with negligible increase in computational cost concerning that of constructing the decision tree. Moreover, as a side effect, the missing branches problem can also be solved by our algorithm.

Keywords: Data Mining, |Decision Tree, Irrelevant Value, Classification Rule

二、緣由與目的

「分類」是資料發掘中一項主要的技術，目前已經有很多不同的分類技術被提出來，並且廣泛的應用到許多用途上。在這些技術中最廣為人知的方法，即是利用決策樹來發掘深藏在資料庫中的有用知識。

自從 ID3[1, 2, 3, 4]被提出來建立決策樹後，就有許多不同的研究者對 ID3 演算法做進一步的改良，期望能夠更有效的建立決策樹。在這份研究報告當中，我們提供了一個全新的觀點來解決不相關值的問題，我們考慮在將決策樹轉換成法則的過程中將不相關值移除，所以不相關的條件不會出現在所得到的分類法則中。因為我們所提出的演算法的輸入是一個決策樹，所以我們的演算法可以輕易的和既有的 ID3 演算法合併使用。但由於這些演算法皆繼承 ID3 演算法的本質及特性，所以我們僅針對 ID3 演算法做一討論。

ID3 演算法檢查每個候選特徵的資訊增益（Information Gain），選出最佳的特徵作為根節點，並將資料

（Training data）分類到這個決策樹的每一個分支內，這個過程一直進行到在一個樹葉內的所有資料都屬於同一個類別為止。當 ID3 選擇一個特徵作為節點並向下分支時，它會對於資料中所出現的每一個特徵值建立一個分支，然而，並不是每一個特徵值都和分類過程有關；因此，決策樹或是其子樹若含有不相關值，它將會使分類法則（Classification Rules）過度特殊化，造成決策時間的浪費，產生了「不相關值問題」（Irrelevant Value Problem）。

三、問題陳述

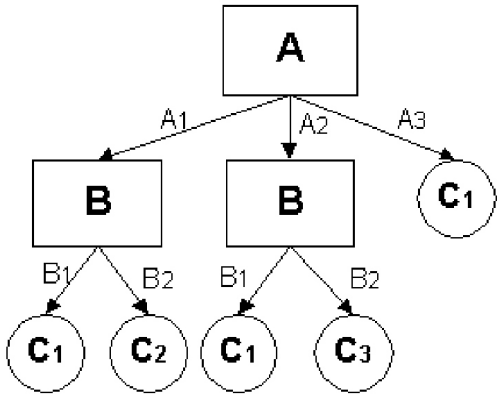
當一棵決策樹被推導出來之後，它將被用來做決策判斷，但由於決策樹本身的特性，使得決策樹之中有時會產生過度分支或過度特殊的問題，進而影響決策或分類的時間及效率，所以，用決策樹來解決分類問題，下列的不相關值問題（Irrelevant Value Problem）及遺失分支問題（Missing Branches Problem），都是不可不注意的，以下我們將針對這兩個問題做一簡單的介紹。

3.1 不相關值問題（Irrelevant Value Problem）

ID3 類型的演算法是選擇最佳的特徵作為根節點，並且強迫每一筆 instance 依照這個特徵去分類，並且，它會將所有的特徵值全部分支出來，但是，由於 ID3 的特性，會產生不相關值的問題，而此問題會增加決策時的成本。若將資料發掘實際應用在醫院對病人的診斷上，由於此不相關值的產生，不但會增加病人不必要的檢查，同時更可能增加病人的危險度，所以對不相關值的處理，是一個非常重要的問題。過分的把子樹特殊化（over-specialization）會造成不必要的過度分支（over-branch），有時候，當我們將決策樹轉換成法則時，有些分支可以刪除，而不會影響正確性。

我們以圖一、表一及表二來說明：

圖一中的決策樹可以直接轉換成



圖一　含有不相關值的決策樹

2	$A_1 \wedge B_1 \Rightarrow C_1$
3	$A_1 \wedge B_2 \Rightarrow C_2$
4	$A_2 \wedge B_1 \Rightarrow C_1$
5	$A_2 \wedge B_2 \Rightarrow C_3$

一一. 圖一的決策樹所轉換的法則

如表一所示的五條法則（rule）。但如果我們再仔細的分析圖一所示的決策樹，我們可以發現，由於 $A_3 \wedge B_1 \Rightarrow C_1$ 隱含在 $A_3 \Rightarrow C_1$ 中，所以決策樹中實際上存在著 $A_1 \wedge B_1 \Rightarrow C_1$、$A_2 \wedge B_1 \Rightarrow C_1$ 及 $A_3 \wedge B_1 \Rightarrow C_1$，加上特徵 A 只有 A_1、A_2、A_3 三個特徵值，所以可以推論出 A_1 在 $A_1 \wedge B_1 \Rightarrow C_1$ 中及 A_2 在 $A_2 \wedge B_1 \Rightarrow C_1$ 中都是一個不相關值，所以法則 $A_1 \wedge B_1 \Rightarrow C_1$ 及

$A_2 \wedge B_1 \Rightarrow C_1$ 可以被簡化成 $B_1 \Rightarrow C_1$。於是我們可以得到如表二所示經過化簡之後的決策法則。

1	$A_3 \Rightarrow C_1$
2	$B_1 \Rightarrow C_1$
3	$A_1 \wedge B_2 \Rightarrow C_2$
4	$A_2 \wedge B_2 \Rightarrow C_3$

一則.－－移除不相關值之後的法則

很顯然的，移除不相關值之後，不但法則變少了，而且原來的法則 2、4 所要判斷的特徵也減少了，但正確性卻沒有減少。

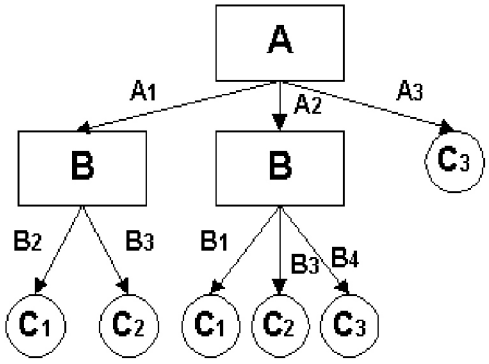
3.2 遺失分支問題（Missing Branches Problem）

雖然本篇文章並不討論遺失分支的問題，但由於我們的演算法亦能部分的解決此一問題，所以本章亦對此一問題做一簡單介紹。

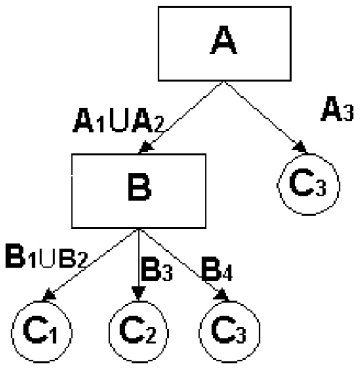
在和 ID3 類似的資料發掘演算法所推導出來的決策樹中，常會發現節點的分支並沒有完整，某些節點只有其中幾個特徵值的分支，而沒有另外幾個特徵值的分支；這是因為資料發掘分類問題的處理程序，在特殊化（Specialization）的過程中，如果所有用來建造決策樹的 training data 中，都沒有包含有這個特徵值的 instance ，則建造出來的決策樹就不會有包含該特徵值的分支。這個就稱做「遺失分支問題」。

遺失分支看起來似乎並沒有什麼大問題，但是，它卻造成決策樹漏掉了一些重要資訊，如果我們能夠把遺失的分支補起來，則很多原本無法用這棵決策樹來判斷的 instances，都在「遺失分支」問題解決之後能夠正確的判斷。因此，它會影響決策樹判斷資料的正確率。

我們以圖二來解釋遺失分支的問題，假設原本 training set 建立出來的決策樹如圖二，我們可以觀察到兩個節點 B 都沒有完整的把所有的特徵值都分支出來，特徵 B 總共有四個特徵值，但是這兩個節點 B 卻各只有兩個及三個分支，所以，它們總共有三條遺失的分支：$A_1 \cap B_1$、$A_1 \cap B_4$、$A_2 \cap B_2$，我們如果把這幾支遺失的分支補上去，可以發現其實分支 A_1 和分支 A_2 可以合併，形成如圖三的決策樹，這個新的決策樹看起來分支變少了，樹葉也變少了，事實上它所隱含的資訊卻增加了，原本的決策樹只有包含六條法則，但是新的決策樹卻包含了九條法則。



圖二　含有遺失分支的決策樹



圖三　移除不相關節點的決策樹

四、解決方法

雖然建立決策樹方式可分為non-incremental 方式和 incremental 方式，由於此兩種方法最後都產生與 ID3 相同的決策樹，所以我們將針對 ID3 tree 做討論。

我們令 A 是一組特徵 {A_1，A_2，…，A_n}，C 是一組類別{C_1，C_2，…，C_s}，而特徵 A_j 可能有的特徵值表示為 domain（A_j），於是，決策樹的分支（Branch）Br 可以被表示成（Br[A_1]，Br[A_2]，…，Br[A_n]，C_i），或（Br[A_1，A_2，…，A_n]，C_i），其中 Br[Aj]是特徵 Aj 在分支 Br 的特徵值；更進一步的，這個分支可以輕易的被轉換成下面的法則（Rule）[26]：

$Br[A_1] \wedge Br[A_2] \wedge \cdots \wedge Br[A_n] \Rightarrow C_i$

在 ID3 類型的演算法當中，當某一個特徵被選出來作為分支中的節點時，它所有的特徵值就會被強迫分支出來；然而，並非每一個特徵值都是有用的，在決策樹某分支 Br 中，會存在某些特徵值 Br[A_j]是沒有意義的，也就是說，在分支 Br 中，即使不去檢查特徵值 Br[A_j]，也可以得到相同的類別，這些特徵值就被視為該分支中的不相關值（Irrelevant Values）。一個特徵值 Br[A_j]若被視為某分支的不相關值，就表示在這個分支中它可以被刪除，或者可以被同一個 domain 中的任何一個特徵值所取代，而不影響法則的正確性。因此，根據不相關值的意思，當特徵值 Br[A_j]是一個不相關值，包含特徵 A_1…A_n 的分支 Br 可以表示如下：

{(Br[A_1，…，A_{j-1}]，a_{jk}，Br[A_{j+1}，…，A_n]，C_i) ｜$a_{jk} \in dom(A_j)$}

其中 $dom(A_j)$ 是 A_j 的 domain。因為這些分支並不是明顯的表示在決策樹當中，所以稱為虛擬分支(Virtual Branches)。我們依據虛擬分支的觀念，定出下面的定理將表示識別分支中不相關值的方法。

定義一：令 Br=(Br[A_1]，…，Br[A_k]，C_i)是決策樹的一個分支，則 Br 在 A_1…A_n 上的虛擬分支是：

{ (Br[A_1，A_2，…，A_k]，a_{k+1}，…，a_n，C_i) ｜$a_j \in Dom(A_j)$ and j=k+1…n}

為了要簡單的識別不相關值，我們做了以下的定義：

定義二：令 Br=(Br[A_1，…，A_k]，C_{i1})是決策樹的一個分支，若(Br[A_1，…，A_k]，C_{i2}) 是 Br'的虛擬分支的一部分，當 $C_{i1} \neq C_{i2}$ 時，則稱為 Br 和 Br'衝突（conflict）。

根據虛擬分支的定義，以及分支互相衝突或是相容的定義，我們就可以從分支的相關法則當中，移除不相關值。下面的定理說明了如何移除不相關值。

定理 1：令 Br 和 Br'是 ID3 決策樹中兩個經過非樹葉節點的分支，Br=(Br[A_1，…，A_{j-1}]，Br[P]，Br[A_j，，…，A_{k1}]，C_{i1})，且 Br'=(Br'[A_1，…，A_{j-1}]，

C_{i2}），令 $A=\{A_j, \cdots, A_{k1}\}$，$A_1$ 是兩個分支中相同的特徵，而 A_2 是在 Br 中存在而在 Br'中不存在的非樹葉節點，即是 $A_1 \subseteq A$，$A_2 \subseteq A$，$A_1 \cup A_2 = A$。則 Br 和 Br'必互相衝突，若且唯若 $Br[A_1]=Br'[A_1]$ 及 $C_{i1} \neq C_{i2}$。

證明：令 $A=\{A_j, \cdots, A_{k1}\}$，$A_1$ 是兩個分支中相同的特徵，而 A_2 在 Br 中存在而在 Br'中不存在的非樹葉節點，其中 $A_1 \subseteq A$，$A_2 \subseteq A$，$Br[A_1] \neq Br'[A_1]$ 代表 $\exists A_k'$，$Br[A_k'] \neq Br'[A_k']$，其中 $A_k' \in A_1$；因此，根據定理 1，這兩個分支由於特徵值不相同，所以一定不可能互相衝突。並且，若這兩個分支的類別相同，它們也一定不可能互相衝突，所以，我們假設當 $Br[A_1]=Br'[A_1]$ 和 $C_{i1} \neq C_{i2}$，則 Br 和 Br' 必互相衝突。

假設 Br 不和 Br'相衝突，因為 Br 和 Br'都經過決策樹中的非樹葉節點 P，且 $Br[A_1]=Br'[A_1]$，則 $Br[A_1, \cdots, A_{j-1}, A_1]$ 必等於 $Br'[A_1, \cdots, A_{k2}, A_1]$。又因為 $A_2=0$，$(Br[A_1, \cdots, A_{j-1}, A_1], C_{i2})$ 一定是 Br'虛擬分支的一部份。然而，因為 $C_{i1} \neq C_{i2}$，Br 必定和 Br'相衝突，這和前題矛盾。

其他相關定理及實驗結果請參閱所附之文章。

五、參考文獻

[1] J. R. Quinlan, "Learning efficient classification procedures and their application to chess end games," in *Machine Learning: An Artificial intelligence approach*, Michalski, Carbonell and Mitchell. eds. Morgan Kaufmann, 1983 pp. 463-482.

[2] J. R. Quinlan, "Induction of decision tree," *Machine Learning* 1, 1986, pp. 81-106.

[3] J. R. Quinlan, "Simplifying decision trees," *Int. J. Man-Machine Studies*, vol. 27, 1987, pp. 221-234.

[4] J. R. Quinlan, "C4.5 : Program for Machine Learning", Morgan Kaufmann, 1993.