Tạp chí Tin học và Điều khiển học, T. 18, S. 4 (2002), 323–332

VỀ CÁC THUẬT TOÁN XÂY DỰNG CÂY QUYẾT ĐỊNH VÀ RÚT GỌN TẬP LUẬT

BÙI THẾ HỒNG

Abstract. Decision tree is one of methods for classification of large data sets into various classes. This paper presents algorithms for generating and pruning decision trees and proposes a method to reduce the rule set derived from a decision tree by statistical tests.

Tóm tắt. Cây quyết định là một trong những phương pháp dùng để phân chia các tập dữ liệu lớn thành các lớp. Bài báo này đề cập đến các thuật toán xây dựng và rút gọn các cây quyết định và đề xuất một phương pháp rút gọn các luật quyết định bằng các phép kiểm định thống kê.

1. CÂY QUYẾT ĐỊNH

Việc xây dựng các cây quyết định chính là quá trình phát hiện ra các luật phân chia tập dữ liệu đã cho thành các lớp đã được định nghĩa trước. Trong thực tế, tập các cây quyết định có thể có đối với bài toán này rất lớn và rất khó có thể duyệt hết được một cách tường tận. Độ phức tạp tính toán của việc tìm một cây phân lớp tối ưu là NP([2]).

Một cây quyết định là một cấu trúc hình cây, trong đó:

+ Mỗi đỉnh trong (đỉnh có thể khai triển được) biểu thị cho một phép thử đối với một thuộc tính;
+ Mỗi nhánh biểu thị cho một kết quả của phép thử;
+ Các đỉnh lá (các đỉnh không khai triển được) biểu thị các lớp hoặc các phân bố lớp;
+ Đỉnh trên cùng trong một cây được gọi là gốc.

Việc sinh cây quyết định bao gồm hai giai đoạn:

(i) Xây dựng cây:

+ Tại thời điểm khởi đầu, tất cả các ca (case) dữ liệu học đều nằm tại gốc;
+ Các ca dữ liệu được phân chia đệ qui trên cơ sở các thuộc tính được chọn.

(ii) Rút gọn cây:

+ Phát hiện và bỏ đi các nhánh chứa các điểm dị thường và nhiễu trong dữ liệu.

Hầu hết các thuật toán dựa vào qui nạp hiện có đều sử dụng phương pháp của Hunt [1] làm thuật toán cơ sở. Dưới đây là mô tả qui nạp phương pháp của Hunt dùng để xây dựng một cây quyết định từ một tập T các ca học với các lớp được ký hiệu là $\{C_1, C_2, ..., C_k\}$.

Trường hợp 1. T chứa một hoặc nhiều ca, tất cả đều thuộc về một lớp đơn C_j: Cây quyết định cho T là một lá định dạng lớp C_j.

Trường hợp 2. T không chứa ca nào: Cây quyết định cho T là một lá, nhưng lớp được gắn với lá này phải được xác định từ các thuộc tính không thuộc T.

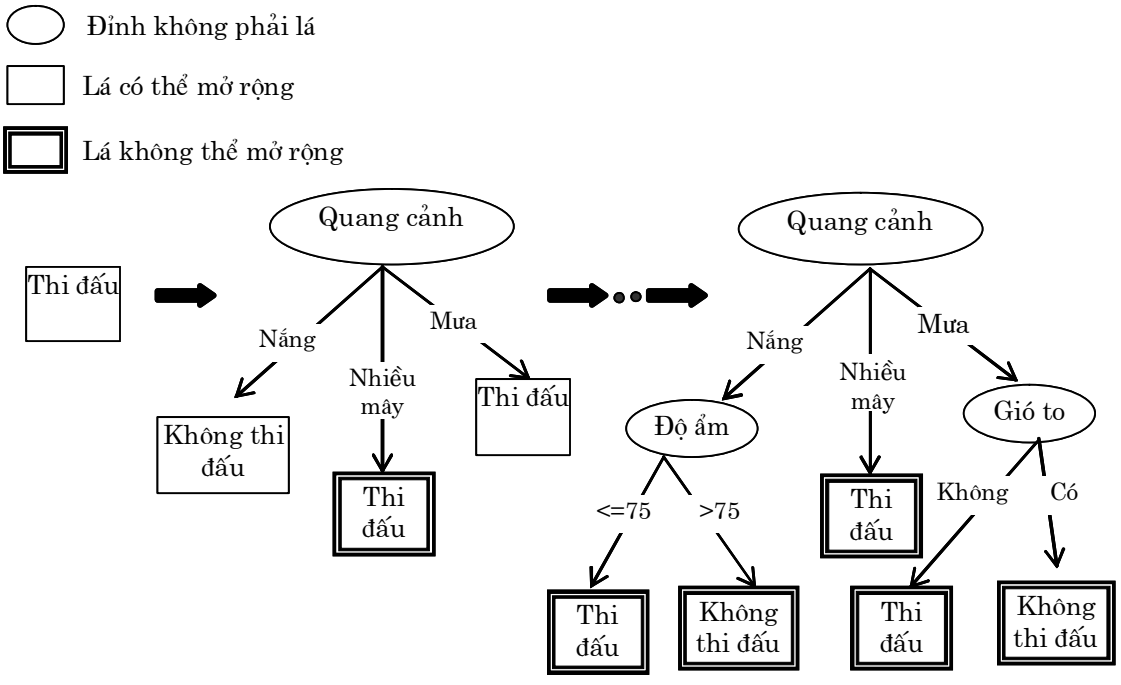
Trường hợp 3. T chứa các ca thuộc về một hỗn hợp của các lớp: Một phép thử được lựa chọn dựa vào một thuộc tính đơn có một hoặc nhiều kết quả (giá trị) loại trừ lẫn nhau $\{O_1, O_2, ..., O_n\}$. T được phân chia thành các tập con $T_1, T_2, , T_n$, trong đó T_i chứa tất cả các ca trong T có kết quả O_i của phép thử đã chọn. Cây quyết định cho T gồm một đỉnh quyết định định danh cho phép thử, và một nhánh cho mỗi kết quả có thể có. Cơ chế xây dựng cây này được áp dụng đệ quy cho từng tập con của các ca học.

Bảng 1 là một tập dữ liệu học của một ví dụ về thi đấu tennis với năm thuộc tính và hai lớp (thuộc tính Ngày được sử dụng làm định danh cho các ca). Hình 1 chỉ ra cách làm việc của thuật

toán Hunt với tập dữ liệu học này. Trong trường hợp 3 của phương pháp Hunt, một phép thử dựa trên một thuộc tính đơn được chọn để khai triển đỉnh hiện hành.

Bảng 1. Một tập dữ liệu học ([1])

Ngày	Quang cảnh	Nhiệt độ	Độ ẩm(%)	Gió to	Kết quả
N1	nắng	24	70	không	thi đấu
N2	nắng	27	90	có	không thi đấu
N3	nắng	30	85	không	không thi đấu
N4	nắng	22	95	không	không thi đấu
N5	nắng	20	70	không	thi đấu
N6	nhiều mây	22	90	có	thi đấu
N7	nhiều mây	28	75	không	thi đấu
N8	nhiều mây	18	65	có	thi đấu
N9	nhiều mây	28	75	không	thi đấu
N10	mưa	21	80	có	không thi đấu
N11	mưa	18	70	có	không thi đấu
N12	mưa	24	80	không	thi đấu
N13	mưa	20	80	không	thi đấu
N14	mưa	21	96	không	thi đấu



a)Cây phân lớp khởi đầu b)Cây phân lớp trung gian c) Cây phân lớp cuối cùng

Hình 1. Minh họa phương pháp của Hunt

2. THUẬT TOÁN ID3

Thuật toán ID3 (**Quinlan86**) là một trong những thuật toán xây dựng cây quyết định sử dụng information gain để lựa chọn thuộc tính phân lớp các đối tượng. Nó xây dựng cây theo cách từ trên xuống, bắt đầu từ một tập các đối tượng và một đặc tả của các thuộc tính. Tại mỗi đỉnh của cây, một thuộc tính có *information gain* lớn nhất sẽ được chọn để phân chia tập đối tượng. Quá trình này được thực hiện một cách đệ quy cho đến khi tập đối tượng tại một cây con đã cho trở nên thuần nhất, tức là nó chỉ chứa các đối tượng thuộc về cùng một lớp. Lớp này sẽ trở thành một lá của cây.

Việc lựa chọn một thuộc tính nào cho phép thử là rất quan trọng. Nếu chọn không thích hợp, chúng ta có thể có một cây rất phức tạp. Ví dụ, nếu ta chọn thuộc tính **Nhiệt độ** làm gốc cho cây thì cây quyết định sẽ có hình dạng như trong Hình 2. Nhưng nếu chọn thuộc tính **Quang cảnh** làm gốc thì ta lại có một cây quyết định rất đơn giản như đã chỉ ra trong Hình 1. Vậy nên chọn thuộc tính nào là tốt nhất?

Thông thường việc chọn thuộc tính đều dựa vào một độ đo được gọi là **Entropy Gains** hay còn gọi là **Information Gain** của các thuộc tính. Entropy của một thuộc tính được tính toán từ các thuộc tính phân lớp. Đối với các thuộc tính rời rạc, cần phải có các thông tin phân lớp của từng giá trị thuộc tính.

Bảng 2. Thông tin phân bố lớp của thuộc tính **Quang cảnh**

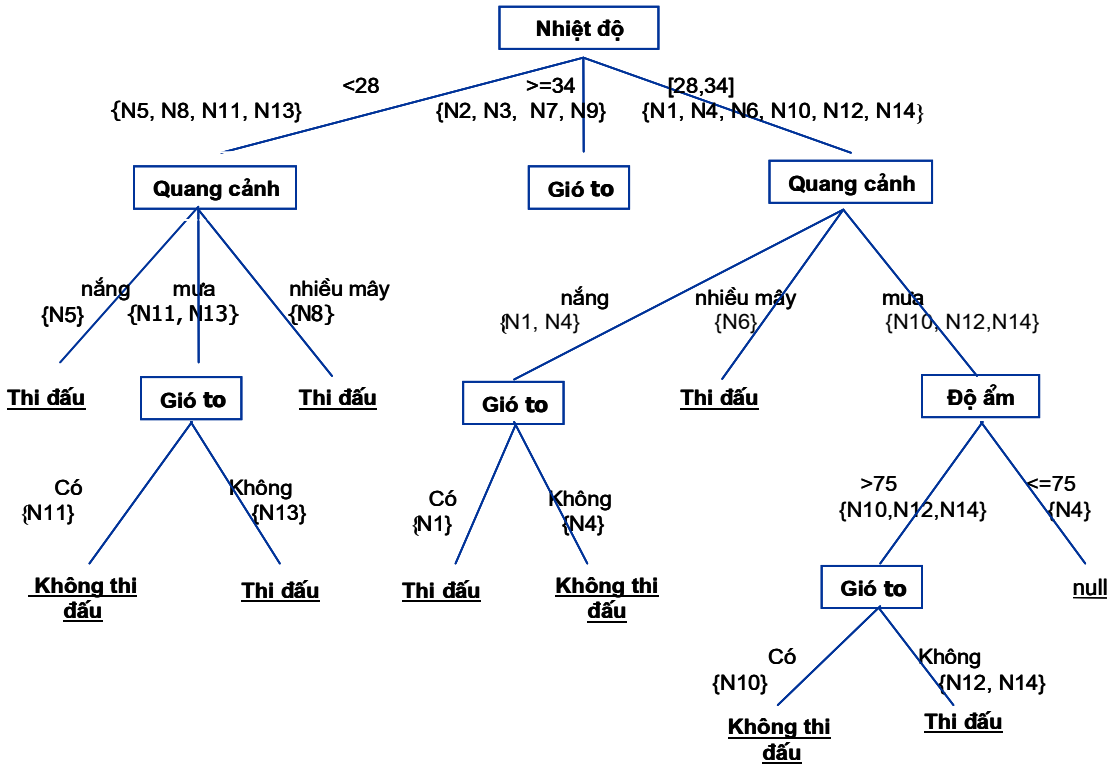
Giá trị thuộc tính	**Lớp**	
	Thi đấu	**Không thi đấu**
Nắng	2	3
Nhiều mây	4	0
Mưa	3	2

Bảng 3. Thông tin phân bố lớp của thuộc tính **Độ ẩm**

Giá trị thuộc tính	**Phép thử nhị phân**	**Lớp**	
		Thi đấu	**Không thi đấu**
65		1	0
	>	8	5
70		3	1
	>	6	4
75		5	1
	>	4	4
78		5	1
	>	4	4
80		7	2
	>	2	3
85		7	3
	>	2	2
90		8	4
	>	1	1
95		8	5
	>	1	0
96		9	5
	>	0	0

Bảng 2 cho thấy thông tin phân lớp của thuộc tính **Quang cảnh.** Đối với một thuộc tính liên

tục, chúng ta phải xét phép thử nhị phân đối với tất cả các giá trị khác nhau của thuộc tính. Bảng 3 chỉ ra thông tin phân lớp của thuộc tính **Độ ẩm.**



Hình 2. Một cây quyết định chọn **Nhiệt độ** làm gốc

Một khi đã thu nhận được các thông tin phân lớp của tất cả các thuộc tính, chúng ta sẽ tính được Entropy. Một thuộc tính với Entropy lớn nhất sẽ được chọn làm một phép thử để khai triển cây.

2.1. Hàm Entropy

Hàm Entropy xác định tính không thuần khiết của một tập các ca dữ liệu bất kỳ. Chúng ta gọi **S** là tập các ca dương tính (ví dụ Thi đấu) và âm tính (ví dụ Không Thi đấu), $P_{\oplus}$ là tỷ lệ các ca dương tính trong S, $P_{\ominus}$ là tỷ lệ các ca âm tính trong **S**

$$\text{Entropy}(S) = - P_{\oplus} \log_2 P_{\oplus} - P_{\ominus} \log_2 P_{\ominus}$$

Ví dụ 1. Trong Bảng 1 của ví dụ thi đấu tennis, tập S có 9 ca dương và 5 ca âm (ký hiệu là [9+, 5-]).

$$\text{Entropy(S)} = \text{Entropy([9+, 5-])} = - (9/14)*\log_2(9/14) - (5/14)*\log_2(5/14) = 0.940$$

Nhận xét. Entropy bằng 0 nếu tất cả các ca trong S đều thuộc về cùng một lớp. Chẳng hạn như, nếu tất cả các ca đều dương thì $P_{\oplus} = 1$ và $P_{\ominus} = 0$, do vậy

$$\text{Entropy(S)} = -1*\log_2(1) - 0*\log_2(0) = -1* 0 - 0 * \log_2(0) = 0$$

Entropy bằng 1 nếu tập S chứa số ca dương và âm bằng nhau. Nếu số các ca này khác nhau thì entropy nằm giữa 0 và 1.

Trường hợp tổng quát, nếu S bao gồm c lớp, thì entropy của S được tính bằng công thức sau:

$$\text{Entropy(S)} = \sum_{i=1}^{c} - p_i \log_2 p_i$$

trong đó p_i là tỷ lệ của các ca thuộc lớp i trong tập S.

2.2. Độ đo (Information Gain)

Độ đo, đo mức hiệu quả của một thuộc tính trong bài toán phân lớp dữ liệu. Đó chính là sự rút gọn mà ta mong đợi khi phân chia các ca dữ liệu theo thuộc tính này. Nó được tính theo công thức sau đây:

$$\text{Gain(S, A)= Entropy(S)} - \sum_{v \in \text{Value(A)}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

trong đó Value(A) là tập tất cả các giá trị có thể có đối với thuộc tính A, và S_v là tập con của S mà A có giá trị là v.

Ví dụ 2.

Value(Gió to) = {true, false}, $S = [9+, 5-]$
S_{true}, là đỉnh con với giá trị là "true", bằng [3+,3-]
S_{false}, là đỉnh con với giá trị là "false", bằng [6+,2-]

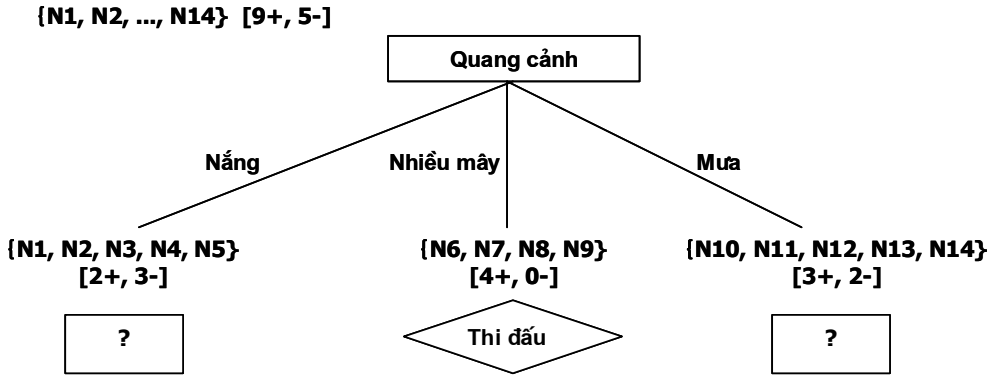
$$\text{Gain}(S, \text{ Gió to}) = \text{Entropy(S)} - \sum_{v \in \{\text{true, false}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - (6/14)\text{Entropy}(S_{\text{true}}) - (8/14)\text{Entropy}(S_{\text{false}})$$
$$= 0.940 - (6/14) * 1 - (8/14) * 0.811$$
$$= 0.048$$

Tương tự như vậy, ta có thể tính được **độ đo** cho các thuộc tính còn lại của ví dụ trong Bảng 1. Đối với thuộc tính Độ ẩm, ta lấy độ ẩm 75% để chia các ca thành hai phần, một phần ứng với các ca có độ ẩm $\leq 75\%$ được gọi là có độ ẩm Bình thường ([5+, 1-]), phần còn lại được gọi là có độ ẩm Cao ([4+, 4-]). Còn đối với thuộc tính Nhiệt độ, ta sẽ chia thành 3 mức, các ngày có nhiệt độ nhỏ hơn 21° được gọi là Lạnh (4 ngày), các ngày có nhiệt độ lớn hơn hoặc bằng 21° đến nhỏ hơn 27° được gọi là Ấm (6 ngày), và còn lại là những ngày có nhiệt độ lớn hơn hoặc bằng 27° được gọi là Nóng (4 ngày).

Gain(S, Quang cảnh) = 0.246
Gain(S, Gió to) = 0.048
Gain(S, Nhiệt độ) = 0.029
Gain(S, Độ ẩm) = 0.045

Từ đây ta thấy rằng **độ đo** của S đối với thuộc tính Quang cảnh là lớn nhất trong số 4 thuộc tính. Như vậy, có thể quyết định chọn Quang cảnh làm thuộc tính đầu tiên để khai triển cây. Hình 3 là khai triển của cây quyết định theo thuộc tính Quang cảnh.



Hình 3. Khai triển cây theo thuộc tính đã chọn

Tương tự như vậy, ta có thể tiến hành triển khai các nút ở mức tiếp theo.

$S_{\text{nắng}}$ = {N1, N2, N3, N4, N5}

Entropy($S_{\text{nắng}}$) = - (2/50*$\log_2$(2/5) - (3/5)*$\log_2$(3/5) = 0.970

Gain($S_{\text{nắng}}$, Độ ẩm)= 0.970 - (3/5) *0.0 - (2/5)*0.0 = **0.970**

Gain($S_{\text{nắng}}$, Nhiệt độ) = 0.970 - (2/5)*0.0 - (2/5)*1.0 - (1/5)*0.0 = **0.570**

Gain($S_{\text{nắng}}$, Gió to) = 0.970 - (2/5) * 1.0 - (3/5)*0.918 = **0.019**

Từ các giá trị của Entropy Gain, ta thấy Độ ẩm là thuộc tính tốt nhất cho đỉnh nằm dưới nhánh Nắng của thuộc tính Quang cảnh.

Tiếp tục quá trình trên cho tất cả các đỉnh và sẽ dừng khi không còn đỉnh nào có thể khai triển được nữa. Cây kết quả sẽ có dạng như Phần c) của Hình 1.

2.3. Thuật toán C4.5

Thuật toán này do Quinlan đưa ra năm 1993. Thuật toán C4.5 sinh ra một cây quyết định phân lớp đối với một tập dữ liệu đã cho bằng cách phân chia đệ qui dữ liệu. Cây quyết định được triển khai theo chiến lược *chiều sâu trước* (Depth-first). Thuật toán này xét tất cả các phép thử có thể phân chia tập dữ liệu đã cho và chọn ra một phép thử cho GainRatio tốt nhất. GainRatio cũng là một độ đo sự hiệu quả của một thuộc tính trong thuật toán triển khai cây quyết định. Nó được tính trên cơ sở của **độ đo** như sau:

$$\text{GainRatio}(S, A) \equiv \frac{\text{Gain}(S, A)}{\text{SplitInformation}(S, A)}$$

$$\text{SplitInformation}(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

trong đó S_i là tập con của S với A có giá trị là v_i

Đối với mỗi thuộc tính rời rạc, chúng ta phải xét một phép thử với tất cả các giá trị khác nhau của nó. Còn đối với mỗi thuộc tính liên tục, ta phải xét các phép thử nhị phân cho mọi giá trị phân biệt của thuộc tính này. Để thu thập Entropy Gain của tất cả các phép thử nhị phân này một cách hữu hiệu thì tập dữ liệu thuộc về đỉnh đang xét phải được phân loại theo các giá trị của thuộc tính liên tục và entropy gains của phép cắt nhị phân dựa trên mỗi giá trị phân biệt của thuộc tính này được tính toán bằng một lần duyệt các dữ liệu đã được phân loại. Quá trình này được thực hiện đối với mọi thuộc tính liên tục.

3. THUẬT TOÁN RÚT GỌN CÁC LUẬT QUYẾT ĐỊNH

3.1. Sinh các luật phân lớp từ cây quyết định

Bảng 4. Tập luật phân lớp chưa rút gọn

Luật	**Nếu**	**Thì**
1	Trời nắng Độ ẩm bình thường	Thi đấu
2	Trời nắng Độ ẩm cao	Không thi đấu
3	Trời nhiều mây	Thi đấu
4	Trời mưa Có gió to	Không thi đấu
5	Trời mưa Không có gió to	Thi đấu

Một khi đã xây dựng được cây quyết định, ta có thể chuyển cây quyết định này thành một tập các luật phân lớp tương đương. Ví dụ, từ cây quyết định của tập dữ liệu học của Bảng 1, ta rút ra các luật phân lớp như chỉ ra trong Bảng 4.

3.2. Rút gọn các luật phân lớp

Sau khi thu được một tập các luật phân lớp từ cây quyết định, cần phải tiến hành rút gọn các luật dư thừa nếu có. Dưới đây, chúng tôi đề xuất một phương pháp đơn giản sử dụng các phép thử thống kê để loại bỏ các luật không cần thiết. Phương pháp này bao gồm các bước sau đây:

1) Loại bỏ các tiền đề không cần thiết để đơn giản hoá các luật.
+ Xây dựng các bảng ngẫu nhiên (contingency table) cho mỗi luật có chứa nhiều hơn một tiền đề.
+ Kiểm chứng sự độc lập của kết quả đối với một tiền đề bằng một trong các phép thử sau:
- Sử dụng phép thử Khi bình phương nếu các tần xuất mong đợi lớn hơn 10.
- Sử dụng phép thử Yates nếu các tần xuất mong đợi nằm trong khoảng [5,10].
- Sử dụng phép thử Fisher nếu các tần xuất này nhỏ hơn 5.

2) Loại bỏ các luật không cần thiết để rút gọn tập luật
+ Một khi đã đơn giản hoá các luật bằng cách loại bỏ các tiền đề dư thừa thì có thể rút gọn toàn bộ tập luật bằng cách bỏ đi các luật không cần thiết.
+ Thử thay thế các luật có chung kết quả chung nhất bằng một luật mặc nhiên được tự động áp dụng khi không có luật nào khác thích hợp.

Các bảng ngẫu nhiên

Sau đây là một bảng ngẫu nhiên dùng để biểu diễn một luật dưới dạng bảng:

	C_1	C_2	Các tổng biên
R_1	x_1	x_2	$R_{1T} = x_1 + x_2$
R_2	x_3	x_4	$R_{2T} = x_3 + x_4$
Các tổng biên	$C_{1T} = x_1 + x_3$	$C_{2T} = x_2 + x_4$	$T = x_1 + x_2 + x_3 + x_4$

Trong đó:

+ R_1 và R_2 biểu diễn các trạng thái Boole của một tiền đề đối với các kết luận C_1 và C_2 (C_2 là phủ định của C_1).
+ x_1 cho đến x_4 biểu diễn tần xuất của từng cặp tiền đề - kết luận.
+ $R_{1T}, R_{2T}, C_{1T}, C_{2T}$ là các tổng biên của các dòng các cột tương ứng.

Các tổng biên và T (tổng tất cả các tần xuất của bảng) được sử dụng để tính các giá trị mong đợi tại các ô dùng trong phép thử độc lập.

Phép thử sự độc lập

Cho một bảng ngẫu nhiên gồm r dòng và c cột:
1) Tính các tổng biên.
2) Tính tổng tần xuất T của bảng.
3) Tính các tần xuất mong đợi cho mỗi ô theo công thức:

$$e_i = \frac{R_{iT} * C_{iT}}{T}$$

Trong đó, R_{iT} và C_{iT} là các tổng dòng và tổng cột tương ứng của ô i trong bảng ngẫu nhiên.
4) Chọn phép thử cần sử dụng để tính χ^2 dựa vào tần xuất mong đợi cao nhất m:

Nếu	thì sử dụng
$m > 10$	Phép thử Khi bình phương
$5 \leq m \leq 10$	Phép thử Yates
$m < 5$	Phép thử Fisher

5) Tính χ^2 theo phép thử đã chọn.
6) Tính bậc tự do $df = (r-1)*(c-1)$.
7) Sử dụng một bảng Khi bình phương với χ^2 và df đã tính được để xác định xem liệu các kết luận có độc lập với tiền đề tại mức ý nghĩa đã chọn không.

+ Giả sử $\alpha = 0.05$.
+ Nếu $\chi^2 > \chi^2_\alpha$ thì loại bỏ giả thiết không về tính độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Tức là giữ lại các tiền đề này vì các kết luận phụ thuộc vào chúng.
+ Nếu $\chi^2 \leq \chi^2_\alpha$ thì chấp nhận giả thuyết không về tính độc lập. Chúng ta sẽ loại bỏ các tiền đề này vì các kết luận độc lập với chúng.

Sau đây là các công thức để tính χ^2 cho các phép thử:

Phép thử Khi bình phương:

$$\chi^2 = \sum_i \frac{(0_i - e_i)^2}{e_i}$$

Phép thử Yates:

$$\chi^2 = \sum_i \frac{(|0_i - e_i| - 0.5)^2}{e_i}$$

Chúng ta sẽ thử rút gọn tập luật của ví dụ Thi đấu tennis theo phương pháp trên đây. Tập luật chưa rút gọn được liệt kê trong Bảng 4.

Giả sử mức ý nghĩa được lấy là $\alpha = 0.05$. Các dữ liệu học được nhân lên bốn lần để có thể sử dụng phép thử Khi bình phương.

3.2.1. Loại bỏ các tiền đề không cần thiết

a) Xét hai tiền đề trong luật 1: Trời nắng và Độ ẩm bình thường

+ Trời nắng

Thực tế

	Thi đấu	**Không thi đấu**	**Tổng biên**
Trời nắng	8	12	20
Không nắng	28	8	36
Tổng biên	36	20	56

Mong đợi

	Thi đấu	**Không thi đấu**
Trời nắng	12.9	7.1
Không nắng	23.1	12.9

Do tần xuất mong đợi lớn nhất $m = 23.1$ nên ta có thể chọn phép thử độc lập Khi bình phương.
Từ đây, theo công thức ta có $\chi^2 = 7.99$.
Bậc tự do của bảng là $df = (r-1)*(c-1) = (2-1)*(2-1) = 1$
Từ bảng Khi bình phương, ta có $\chi^2_\alpha = 3.84$.
Vì $\chi^2 = 7.99 > \chi^2_\alpha = 3.84$, nên chúng ta loại bỏ giả thiết không về tính độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Do vậy, theo dữ liệu học thì việc Thi đấu tennis phụ thuộc vào trời nắng hay không. Như vậy, chúng ta không thể loại bỏ tiền đề này.

+ Độ ẩm bình thường

Thực tế

	Thi đấu	**Không thi đấu**	**Tổng biên**
Độ ẩm bình thường	20	4	24
Độ ẩm không bình thường	16	16	32
Tổng biên	36	20	56

Mong đợi

	Thi đấu	**Không thi đấu**
Độ ẩm bình thường	15.4	8.6
Độ ẩm không bình thường	20.6	11.4

Do tần xuất mong đợi lớn nhất $m = 20.6$, nên ta có thể chọn phép thử độc lập Khi bình phương.
Từ đây, theo công thức ta có $\chi^2 = 6.64$.
Bậc tự do của bảng là $df = (r-1) * (c-1) = (2-1) * (2-1) = 1$
Từ bảng Khi bình phương, ta có $\chi^2_\alpha = 3.84$.
Vì $\chi^2 = 7.99 > \chi^2_\alpha = 3.84$, nên chúng ta loại bỏ giả thiết không về tính độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Do vậy, theo dữ liệu học thì việc Thi đấu tennis phụ thuộc vào Độ ẩm có bình thường hay không. Như vậy, chúng ta không thể loại bỏ tiền đề này.

b) Xét tiếp hai tiền đề trong luật 4: Trời mưa và Có gió to

+ Trời mưa

Thực tế

	Thi đấu	**Không Thi đấu**	**Tổng biên**
Trời mưa	12	8	20
Trời không mưa	24	12	36
Tổng biên	36	20	56

Mong đợi

	Thi đấu	**Không thi đấu**
Trời mưa	12.9	7.1
Trời không mưa	32.1	12.9

Từ đây, theo công thức ta có $\chi^2 = 0.25$.
Vì $\chi^2 = 0.25 < \chi^2_\alpha = 3.84$, nên chúng ta chấp nhận giả thiết không về tính độc. Do vậy, theo dữ liệu học thì việc Thi đấu tennis không phụ thuộc vào Trời mưa. Như vậy, chúng ta có thể loại bỏ tiền đề này trong các luật 4 và 5.

+ Gió to

Thực tế

	Thi đấu	**Không thi đấu**	**Tổng biên**
Gió to	8	12	20
Không gió to	28	8	36
Tổng biên	36	20	56

Mong đợi

	Thi đấu	Không thi đấu
Gió to	12.9	7.1
Không gió to	23.1	12.9

Từ đây, theo công thức ta có $\chi^2 = 7.99$.
Vì $\chi^2 = 7.99 > \chi^2_\alpha = 3.84$, nên chúng ta loại bỏ giả thiết không về tính độc lập và chấp nhận giả thiết thay thế về tính phụ thuộc. Do vậy, theo dữ liệu học thì việc Thi đấu tennis phụ thuộc vào Gió to. Như vậy, chúng ta không thể loại bỏ tiền đề này.

3.2.2. Loại bỏ các luật không cần thiết

Qua việc thử các tiền đề trong tập luật, chúng ta thấy rằng không thể loại bỏ hoàn toàn một luật nào mà chỉ loại bỏ được tiền đề Trời mưa trong các luật 4 và 5. Bảng luật thu gọn được cho trong Bảng 5.

Bảng 5. Tập luật phân lớp đã rút gọn

Luật	Nếu	Thì
1	Trời nắng Độ ẩm bình thường	Thi đấu
2	Trời nắng Độ ẩm cao	Không thi đấu
3	Trời nhiều mây	Thi đấu
4	Có gió to	Không thi đấu
5	Không có gió to	Thi đấu

4. KẾT LUẬN

Bài báo đã đề cập đến phương pháp cây quyết định để phân lớp dữ liệu trong các cơ sở dữ liệu lớn và đề xuất một phương thức rút gọn tập các luật bằng các phép thử thống kê nhằm loại bỏ các tiền đề cũng như các luật không cần thiết. Việc rút gọn cây quyết định và rút gọn tập các luật được sinh từ cây quyết định sẽ làm giảm đáng kể thời gian phân lớp các ca dữ liệu trong các cơ sở dữ liệu lớn và cần phải có những thuật toán thật hữu hiệu. Hiện nay, các vấn đề này đang được rất nhiều người quan tâm nghiên cứu.

TÀI LIỆU THAM KHẢO

[1] Ho Tu Bao, "Knowledge Discovery and Data Mining", UNESCO, Training Material on Informatics, (1999).
[2] Karuna Pande Joshi, *Analysis of Data Mining Algorithms* 1997.
[3] Two Crows "Introduction to Data Mining and Knowledge Discovery" (1999).
[4] R. Agrawal, A. Arning, T. Bollinger, M. Mehta, J. Shafer, R. Srikant, The Quest Data Mining System, *Proc. of the 2nd Int'l Conference on Knowledge Discovery in Databases and Data Mining*, Portland, Oregon, August, (1996).

Nhận bài ngày 24 - 5 - 2002

Viện Công nghệ Thông tin