SỬ DỤNG CÂY QUYẾT ĐỊNH TRONG KHAI PHÁ DỮ LIỆU

● NGUYỄN THỊ VIỆT HÀ

TÓM TẮT:

Trong những thập niên gần đây, số lượng cũng như dung lượng của các cơ sở dữ liệu tăng lên nhanh chóng. Ước lượng thông tin trên toàn cầu tăng khoảng gấp 2 lần sau 2 năm. Lượng dữ liệu khổng lồ này thực sự trở thành nguồn tài nguyên giá trị. Tuy nhiên, lượng dữ liệu mà chúng ta lưu trữ trở nên quá nhiều, gây khó khăn cho việc lấy ra được những thông tin hữu ích. Bài viết bàn về phương pháp sử dụng cây quyết định trong khai phá dữ liệu.

Từ khóa: Cây quyết định, cơ sở dữ liệu, khai phá dữ liệu, dung lượng, cơ sở dữ liệu.

1. Đặt vấn đề

Sự bùng nổ và phát triển của công nghệ thông tin trong những năm gần đây đã mang lại hiệu quả đối với khoa học cũng như các hoạt động thực tế. Sự phát triển mạnh mẽ của công nghệ thông tin đã làm cho khả năng thu thập và lưu trữ thông tin của các hệ thống thông tin tăng nhanh chóng, lượng dữ liệu lưu trữ trở nên quá nhiều, gây lúng túng cho việc lấy ra được những thông tin hữu ích. Do vậy, cần có những kỹ thuật và công cụ mới để tự động chuyển đổi lượng dữ liệu khổng lồ kia thành các tri thức hữu ích. Từ đó, các kỹ thuật khai phá dữ liệu đã trở thành một trong những lĩnh vực thu hút nhiều nhà khoa học quan tâm nghiên cứu

Khai phá dữ liệu đã được ứng dụng rộng rãi trong nhiều lĩnh vực: sinh học, y học, viễn thông, giáo dục, trí tuệ nhân tạo, cơ sở dữ liệu, thuật toán học, tính toán song song và tốc độ cao, thu thập tri thức cho các hệ chuyên gia. Đặc biệt, khai phá dữ liệu rất gần gũi với lĩnh vực thống kê, sử dụng các phương pháp thống kê để mô hình dữ liệu và phát hiện các luật

Việc khai thác những thông tin tiềm ẩn mang tính dự đoán từ những cơ sở dữ liệu lớn là mục tiêu chính của khai phá dữ liệu. Những công cụ khai phá dữ liệu có thể dự đoán những xu hướng trong tương lai, do đó cho phép các tổ chức, doanh nghiệp ra quyết định kịp thời được định hướng bởi tri thức mà công nghệ khai phá dữ liệu đem lại. Sự phân tích dữ liệu một cách tự động và mang tính dự báo của khai phá dữ liệu ưu thế hơn hẳn so với sự phân tích thông thường dựa trên những sự kiện trong quá khứ của các hệ hỗ trợ ra quyết định truyền thống trước đây. Công cụ khai phá dữ liệu cũng có thể trả lời các câu hỏi trong lĩnh vực kinh doanh mà trước đây tốn nhiều thời gian để xử lý.

Một trong những phương pháp khai phá dữ liệu có hiệu quả, được ứng dụng nhiều và được nhiều nhà khoa học nghiên cứu nhiều năm qua là phương pháp Cây quyết định. Với khả năng ứng dụng thiết thực vào đời sống xã hội của phương pháp này, nội dung "Sử dụng cây quyết định trong khai phá dữ liệu" được xây dựng và tổng hợp dựa trên một số nghiên cứu chủ yếu trong lĩnh vực khai phá dữ liệu của các nhà nghiên cứu trong những năm gần đây ở một số hội nghị quốc tế và một số bài báo được công bố trên các tạp chí chuyên ngành, trên internet...

2. Tổng quan về khai phá dữ liệu

2.1. Tính cấp bách của việc khai phá dữ liệu

Sự phát triển của công nghệ thông tin và việc ứng dụng công nghệ thông tin trong nhiều lĩnh vực của đời sống, kinh tế xã hội những năm qua khiến lượng dữ liệu được thu thập, tích lũy ngày càng nhiều lên. Để lấy được những thông tin có tính "tri thức" trong khối dữ liệu khổng lồ này, người ta đi tìm những kỹ thuật có khả năng hợp nhất các dữ liệu từ các hệ thống dữ liệu khác nhau, chuyển đổi thành một tập hợp các cơ sở dữ liệu ổn định, có

chất lượng được sử dụng chỉ riêng cho một vài mục đích nào đó. Các kỹ thuật đó được gọi chung là kỹ thuật tạo kho dữ liệu (Data Warehousing) và môi trường các dữ liệu đó được gọi là các kho dữ liệu.

Tuy nhiên, việc khai thác dữ liệu theo truyền thống mới chỉ dừng lại ở cách khai thác dữ liệu với các kỹ thuật cao để đưa ra các dữ liệu tinh và chính xác hơn chứ chưa đưa ra được dữ liệu mang tính "tri thức". Trong khi đó, càng ngày người ta càng nhận thấy rằng nếu được phân tích thông minh thì dữ liệu sẽ là một nguồn tài nguyên quí hiếm trong cạnh tranh trên thương trường. Một giải pháp công nghệ mới được nghiên cứu, đáp ứng cả nhu cầu trong khoa học cũng như trong hoạt động thực tiễn. Đó chính là công nghệ phát hiện tri thức và khai phá dữ liệu (Knowledge Discovery and Data Mining - KDD).

2.2. Mục tiêu của khai phá dữ liệu

Thuật ngữ khai phá dữ liệu ra đời vào những năm cuối của thập kỷ 1980. Giáo sư Tom Mitchell đã đưa ra định nghĩa về khai phá dữ liệu như sau: "Khai phá dữ liệu là việc sử dụng dữ liệu lịch sử để khám phá những qui tắc và cải thiện những quyết định trong tương lai". Còn các nhà thống kê thì xem "khai phá dữ liệu như là một quá trình phân tích được thiết kế thăm dò một lượng cực lớn các dữ liệu nhằm phát hiện ra các mẫu thích hợp hoặc các mối quan hệ mang tính hệ thống giữa các biến và sau đó sẽ hợp thức hóa các kết quả tìm được bằng cách áp dụng các mẫu đã phát hiện được cho tập con mới của dữ liệu". Nói chung, khai phá dữ liệu là cốt lõi của quá trình khám phá tri thức. Nó gồm có các giải thuật khai phá dữ liệu chuyên dùng, dưới một số quy định về hiệu quả tính toán chấp nhận được.

Mục đích chung của việc phát hiện tri thức và khai phá dữ liệu là tìm ra các mẫu được quan tâm nhất hoặc các mô hình tồn tại trong cơ sở dữ liệu, nhưng chúng lại bị che dấu bởi một số lượng lớn dữ liệu.

2.3. Một số phương pháp khai phá dữ liệu phổ biến

Tầm quan trọng của việc khai phá dữ liệu trong thời đại công nghệ số đã dẫn đến sự ra đời của nhiều phương pháp khai phá dữ liệu. Hiện nay, các phương pháp khai phá dữ liệu phổ biến là: Cây quyết định và luật; Phương pháp suy diễn và quy nạp; Luật kết hợp; Phân nhóm và phân đoạn; Mạng neural; Giải thuật di truyền. Trong các phương pháp khai phá dữ liệu phổ biến hiện nay, phương pháp sử dụng Cây quyết định được xem là phương pháp mang lại hiệu quả cao nhất.

Cây quyết định là phương pháp mô tả tri thức dạng đơn giản nhằm phân các đối tượng dữ liệu thành một số lớp nhất định. Các nút của cây được gán nhãn là tên các thuộc tính, các cạnh được gán các giá trị có thể của các thuộc tính, các lá miêu tả các lớp khác nhau. Các đối tượng được phân lớp theo các đường đi trên cây, qua các cạnh tương ứng với giá trị của các thuộc tính của đối tượng tới lá.

Cây quyết định là phương pháp dùng trong các bài toán phân loại dữ liệu theo một tiêu chuẩn nào đó dựa trên mức độ khác nhau của thuộc tính. Cây quyết định và luật có ưu điểm là hình thức miêu tả đơn giản, mô hình suy diễn khá dễ hiểu đối với người sử dụng. Tuy nhiên, giới hạn của nó là miêu tả cây và luật chỉ có thể biểu diễn được một số dạng chức năng và vì vậy giới hạn cả về độ chính xác của mô hình.

3. Kỹ thuật khai phá dữ liệu sử dụng cây quyết định

3.1. Giới thiệu kỹ thuật khai phá dữ liệu sử dụng cây quyết định

Kỹ thuật cây quyết định là một công cụ mạnh và hiệu quả trong việc phân lớp và dự báo. Các đối tượng dữ liệu được phân thành các lớp. Các giá trị của đối tượng dữ liệu chưa biết sẽ được dự đoán, dự báo. Tri thức được rút ra trong kỹ thuật này thường được mô tả dưới dạng đơn giản, trực quan, dễ hiểu đối với người sử dụng.

Cây quyết định là một mô tả tri thức dạng đơn giản nhằm phân các đối tượng dữ liệu thành một số lớp nhất định. Các nút của cây được gán nhãn là tên của các thuộc tính, các cạnh được gán các giá trị có thể của các thuộc tính, các lá mô tả các lớp khác nhau. Các đối tượng được phân lớp theo các đường đi trên cây, qua các cạnh tương ứng với giá trị của thuộc tính của đối tượng tới lá.

Quá trình xây dựng cây quyết định 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.

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 bổ lớp. Đỉnh trên cùng trong một cây được gọi là gốc.

Việc sinh cây quyết định được chia làm 2 giai đoạn:

- Xây dựng cây: tại thời điểm khởi đầu, tất cả các case dữ liệu học đều nằm tại gốc. Các case dữ liệu được phân chia đệ quy trên cơ sở các thuộc tính được chọn.

- Rút gọn cây: Phát hiện và bỏ đi các nhánh chứa các điểm dị thường và nhiễu trong dữ liệu.

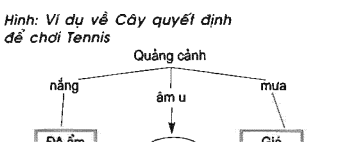
Ví dụ: Cây quyết định chơi tennis có các thuộc tính. Quang cảnh, độ ẩm không khí, sức gió.

Thuộc tính Quang cảnh = {Nắng, Âm u, Mưa}

Thuộc tính độ ẩm = {Cao, TB}

Thuộc tính Sức gió = {Mạnh, nhẹ}

Hình: Ví dụ về Cây quyết định để chơi Tennis



3.2. Thuật toán sử dụng cho việc xây dựng cây quyết định

Trong khai phá dữ liệu bằng cây quyết định thì xây dựng cây là vấn đề mấu chốt và quan trọng nhất. Các thuật toán xây dựng cây quyết định đã được các nhà khoa học phát triển, công bố và cải tiến qua thời gian. Dưới đây là một số thuật toán xây dựng cây quyết định:

3.2.1. Thuật toán CLS

Thuật toán CLS được thiết kế theo chiến lược chia để trị từ trên xuống và gồm những bước sau:

Bước 1. Tạo một nút T, nút này gồm tất cả các mẫu của tập huấn luyện.

Bước 2. Nếu tất cả các mẫu trong T có thuộc tính quyết định mang giá trị "yes" (hay thuộc cùng một lớp), thì gán nhãn cho nút T là "yes" và dừng lại. T lúc này là nút lá.

Bước 3. Nếu tất cả các mẫu trong T có thuộc tính quyết định mang giá trị "no" (hay thuộc cùng một lớp), thì gán nhãn cho nút T là "no" và dừng lại. T lúc này là nút lá.

Bước 4. Trường hợp ngược lại các mẫu của tập huấn luyện thuộc cả hai lớp "yes" và "no" thì:

1. Chọn một thuộc tính X có các giá trị $v_1, v_2, ..v_n$.

2. Chia tập mẫu trong T thành các tập con $T_1, T_2, .., Tn$. Dựa theo các giá trị của X.

3. Tạo n nút con T_i (i = 1, 2,...,n) với nút cha là nút T.

4. Tạo các nhánh nối từ nút T đến các nút T_i (i = 1, 2,...,n)

Bước 5. Thực hiện lặp cho các nút con T_i (i =1,2..n) và quay trở lại bước 2.

3.2.2. Thuật toán ID3

Thuật toán ID3 biểu diễn các khái niệm ở dạng cây quyết định. Biểu diễn này cho phép chúng ta xác định phân loại của đối tượng bằng cách kiểm tra giá trị của nó trên một số thuộc tính nào đó. Nhiệm vụ của thuật toán ID3 là học cây quyết định từ một tập dữ liệu rèn luyện.

Thuật toán ID3 xây dựng cây quyết định sử dụng *Information gain* để lựa chọn thuộc tính phân lớp các đối tượng. Nó xây dựng cây quyết định theo cách từ trên xuống *(top - down)*, bắt đầu từ một tập các đối tượng và 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 trở nên thuần nhất, tức là nó chỉ chứa các đối tượng thuộc về cùng một lớp. Tập này sẽ trở thành một lá của cây. Việc lựa chọn một thuộc tính nào đó cho phép thử là rất quan trọng. Nếu chọn không thích hợp, chúng ta có thể có một cây rất phức tạp. Để làm được việc này thuật toán ID3 có sử dụng tới 2 hàm Entropy và Entropy Gains (hay còn gọi là Information Gain viết tắt là Gain).

Thuật toán ID3 cho kết quả tối ưu hơn thuật toán CLS khi áp dụng trên cùng tập dữ liệu. Nhưng thuật toán này chưa giải quyết được vấn đề thuộc tính số hay liên tục; vấn đề dữ liệu bị thiếu, bị nhiễu... Giải quyết những vấn đề này, Quinlan đã phát triển cải tiến thuật toán ID3 và công bố thuật toán C4.5 vào năm 1993.

2.2.3. Thuật toán C4.5

Thuật toán C4.5 là một cải tiến từ thuật toán ID3 với việc phân lớp dữ liệu trên các thuộc tính số, thuộc tính liên tục và làm việc được với các tập dữ liệu bị thiếu *(Missing Data)* và bị nhiễu *(Noisy Data)*. Thuật toán C4.5 thực hiện phân lớp tập mẫu dữ liệu theo chiến lược ưu tiên theo 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 có giá trị GainRatio tốt nhất, GainRatio cũng là một đại lượng để đánh giá sự hiệu quả một thuộc tính trong thuật toán để triển khai cây quyết định. Nó được tính trên cơ sở đại lượng Information Gain theo công thức sau:

$$GainRation(X,T) = \frac{Gain(X,T)}{SplitInformation(X,T)} \quad (3.1)$$

$$SplitInformation(X,T) = -\sum_{i=1..n} \frac{|T_i|}{|T|} Log_2 \frac{|T_i|}{|T|} \quad (3.2)$$

Trong đó:

Ti là tập con của tập T ứng với thuộc tính X= giá trị là v_i

Value (X) là tập các giá trị của thuộc tính X

Thuật toán xây dựng cây quyết định C4.5

Mô tả thuật toán dưới dạng giả mã như sau:

Function xay_dung_cay(T)

1. <Tính toán tần xuất các giá trị trong các lớp của T>:

2. If <Kiểm tra các mẫu, nếu thuộc cùng một lớp hoặc có rất ít mẫu khác lớp> Then <Trả về 1 nút lá>

Else <Tạo một nút quyết định N>;

3. For <Với mỗi thuộc tính A> Do <Tính giá trị Gain(A)>;

4. <Tại nút N, thực hiện việc kiểm tra để chọn ra thuộc tính có giá trị Gain tốt nhất (lớn nhất). Gọi N.test là thuộc tính có Gain lớn nhất>.

5. If <Nếu N.test là thuộc tính liên tục> Then <Tìm ngưỡng cho phép tách của N.test>;

6. For <Với mỗi tập con T' được tách ra từ tập T> Do

T' được tách ra theo quy tắc:

- Nếu N.test là thuộc tính liên tục tách theo ngưỡng ở bước 5

- Nếu N.test là thuộc tính phân loại rời rạc tách theo các giá trị của thuộc tính này.

7. {If <Kiểm tra nếu T' rỗng>} Then

<Gán nút con này của nút N là nút lá>;

Else

8. <Gán nút con này là nút được trả về bằng cách gọi đệ qui lại đối với hàm xay_dung_cay(T'). với tập T'>;

9. <Tính toán các lỗi của nút N>;

<Trả về nút N>;

Giả sử tập mẫu dữ liệu T được mô tả bằng m thuộc tính ứng viên. Số lượng mẫu trong tập mẫu dữ liệu T được ký hiệu là |T|. Các thuộc tính dùng để phân chia tập mẫu được ký hiệu là $C_1,C_2...C_k$. Quá trình cây được tiến hành từ trên xuống dưới. Đầu tiên ta xác định nút gốc, sau đó xác định các nhánh xuất phát từ gốc này. Tập T được chia thành các tập con theo các giá trị của thuộc tính được xét tại nút gốc.

Tìm thuộc tính phân lớp tại nút gốc: Do T có m thuộc tính nên có m khả năng để lựa chọn thuộc tính. Một số thuật toán thì trong quá trình xây dựng cây mỗi thuộc tính chỉ được xét một lần, nhưng với thuật toán này một thuộc tính có thể được xét nhiều lần.

Xét thuộc tính X có n giá trị lần lượt là $L_1,L_2,..L_n$. Khi đó, ta có thể chia tập T ra thành n tập con X_i(i=1..n) theo các giá trị của X khi X= $L_1,L_2,...L_n$.

Tần xuất freq (C_j,T) là số lượng mẫu của tập T nào đó được xếp vào lớp con C_j. Xác suất để một mẫu thuộc lớp C_j khi lấy bất kỳ một mẫu từ T là:

$$P = \frac{freq(C_j,T)}{|S|} \quad (3.3)$$

Theo Lý thuyết thông tin, số lượng các thông tin chứa trong mẫu phụ thuộc vào khả năng xác suất của nó.

$$\log_2\left(\frac{1}{p}\right) \quad (3.4)$$

Vì ta sử dụng loga nhị phân nên công thức (3.4) sẽ cho ta số lượng biểu diễn bằng bit.

Xét công thức:

$$Info(T) = -\sum_{i=1}^{k} \frac{freq(C_i,T)}{|T|} * \log_2\left(\frac{freq(C_i,T)}{|T|}\right) \quad (3.5)$$

Công thức này đánh giá số lượng thông tin trung bình cần thiết để phân lớp các mẫu trong tập hợp T. Áp dụng công thức (3.5) sau khi phân chia theo X cho ta công thức sau:

$$Info_x(T) = -\sum_{i=1}^{n} \frac{T_i}{T} * Info(T_i) \quad (3.6)$$

Khi đó, công thức (3.7) sẽ đưa ra tiêu chuẩn để lựa chọn thuộc tính khi phân lớp:

$$Gain(X,T) = Info(T) - Info_x(T) \quad (3.7)$$

Tiêu chuẩn (3.7) được tính toán với các thuộc tính và thuộc tính được lựa chọn tại một nút là thuộc tính có Gain lớn nhất. Thuộc tính được chọn sẽ được dùng để phân lớp tập mẫu dữ liệu tại nút đó. Các nhánh sau khi được phân chia thì tiếp tục được tính toán để xác định thuộc tính phân loại tiếp theo để xây dựng cây. Quá trình được tiếp tục cho

đến khi các mẫu trong tập dữ liệu được phân lớp hoàn toàn. Nếu một tập nào đó có các mẫu thuộc cùng một lớp thì ta đánh dấu nó là một nút lá.

Đối với thuộc tính số (thuộc tính liên tục) ta cần chọn một ngưỡng (Threshold) nào đó để so sánh giá trị trong thuộc tính. Giả sử, thuộc tính số có tập hợp giá trị hữu hạn phân biệt được biểu diễn bằng các giá trị $V_1,V_2,...V_n$ (Với giả thiết rằng: $V_1<V_2<...<V_n$). Trước tiên, các giá trị của thuộc tính này sẽ được sắp xếp theo các giá trị cụ thể. Sau đó chọn bất kỳ một giá trị giữa Vi và Vi+1 để chia các mẫu thành hai tập hợp. Một giá trị nằm bên trái và một giá trị nằm bên phải. Có thể chọn giá trị trung bình của Vi và Vi+1 như sau:

$$TH_i = \frac{V_i + V_{i+1}}{2} \qquad (3.8)$$

Chia T thành 2 tập T1 và T2 như sau:

$T_1 = \{V_j \mid V_j <= TH_i\}$ và $T_2 = \{V_j \mid V_j > TH_i\}$

Ứng với mỗi giá trị V như thế, ta tính được giá trị gain tương ứng theo phép tách như trên. Phép tách nào đưa ra được giá trị Gain ứng với thuộc tính X và tập mẫu T là lớn nhất thì phép tách đó được lựa chọn để phân tách tập mẫu T theo các giá trị của thuộc tính X.

Ví dụ minh họa cho thuật toán C4.5: Xét tập dữ liệu T là bảng thống kê mối quan hệ mức độ nguy hiểm khi lái xe và độ tuổi của lái xe được cho trong bảng sau:

Bảng thống kê mức độ nguy hiểm khi lái xe và độ tuổi của lái xe

ID	Tuổi	Loại xe	Mức độ nguy hiểm
1	23	Gia đình	Cao
2	18	Thể thao	Cao
3	43	Thể thao	Cao
4	64	Gia đình	Thấp
5	32	Xe tải	Thấp
6	20	Gia đình	Cao
7	43	Gia đình	Thấp
8	32	Thể thao	Cao
9	43	Xe tải	Thấp

Trong đó:

- Thuộc tính định danh là **ID**
- Thuộc tính ứng viên: **Tuổi, Loại xe**
- Thuộc tính phân lớp: **Mức độ nguy hiểm**

2.2.4. Rút gọn cây quyết định

Một cây quyết định được xây dựng dựa trên một tập dữ liệu học có thể có nhiều nhánh hoặc nhiều lá là do dữ liệu bị nhiễu hoặc bị thiếu. Số lượng các mẫu huấn luyện quá ít không đủ đại diện cho một qui luật, nhưng trong trường hợp đó thuật toán xây dựng cây vẫn tạo ra các nút dựa trên số lượng mẫu quá ít đó. Trong trường hợp này, nếu thuật toán vẫn cứ phát triển cây thì ta sẽ dẫn đến một tình huống mà gọi là tình trạng "Over fitting" trong cây quyết định. Để giải quyết tình trạng Over fitting này người ta sử dụng phương pháp cắt tỉa cây quyết định.

Cắt tỉa cây là việc trộn một cây con trong vào một nút lá của nó. Đó là:

+ Sử dụng một tập dữ liệu khác được rút ra từ trong tập dữ liệu học ban đầu.

+ Tại một nút của cây, nếu sự chính xác khi không chia tách cao hơn sự chính xác khi được chia tách, khi đó hãy thay thế cây con này bằng một nút lá tương ứng, nhãn của nút lá này được gán là nhãn của lớp đa số (phổ biến) trong tập các mẫu tại nút đó.

Kết quả của việc cắt tỉa cây là nhằm:

+ Thu được cây kết quả tối ưu hơn, độ chính xác cao hơn, đó tin tưởng cao hơn.

+ Thu nhận được những tập dữ liệu đã qua kiểm nghiệm, các tiêu chuẩn khác đạt được chính xác hơn.

Để cắt tỉa cây quyết định thường sử dụng 2 chiến lược: Chiến lược tiền cắt tỉa (Prepruning) và chiến lược hậu cắt tỉa (Postpruning).

4. Kết luận

So với các phương pháp khai phá dữ liệu khác, cây quyết định có một số ưu điểm: Cây quyết định tương đối dễ hiểu. Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản, đôi khi không cần thiết phải xử lý dữ liệu trước khi tiến hành khai phá. Trong khi đó, các kỹ thuật khác thường đòi hỏi phải có các thao tác xử lý dữ liệu phức tạp hơn, như: chuẩn hóa dữ liệu, tạo ra các biến phụ hay loại bỏ các giá trị rỗng. Cây quyết định có thể xử lý cả dữ liệu có giá trị bằng số, và dữ liệu có giá trị là tên thể loại dạng phân loại rời rạc. Các kỹ thuật khác thường chuyên để phân tích các bộ dữ liệu chỉ gồm một loại biến.

Tuy nhiên, các kỹ thuật khai phá dữ liệu sử dụng cây quyết là rất đa dạng và phong phú, phụ thuộc vào sự sáng tạo của người triển khai ứng dụng và vào kết quả của các lĩnh vực nghiên cứu khác ■

TÀI LIỆU THAM KHẢO:

1. Vũ Đức Thi, Cơ sở dữ liệu - Kiến thức và thực hành, Nhà xuất bản Thống kê (1997).

2. Vũ Đức Thi, Thuật toán trong tin học, Nhà xuất bản Khoa học kỹ thuật (1999).

3. Nguyễn Thanh Thủy, Khai phá dữ liệu - Kĩ thuật và ứng dụng. Hà Nội Tháng 8 - 2001.

4 Han J. and Kamber (2000), Data mining Concepts and Techniques, Morgan Kaufmann.

5. Murthy, S.K (1998), "Automatic construction of decision trees from data: A multi - disciplication survey". Data mining and Knowledge Discovery 2(4), pp 345 - 389.

Ngày nhận bài: 10/8/2019
Ngày phản biện đánh giá và sửa chữa: 20/8/2019
Ngày chấp nhận đăng bài: 30/8/019

Thông tin tác giả:
ThS. NGUYỄN THỊ VIỆT HÀ
Trưởng khoa Khoa học cơ bản
Trường Cao đẳng Công nghệ và Kinh tế Công nghiệp

USING THE DECISION TREE IN DATA MINING

● Master. **NGUYEN THI VIET HA**
Dean, Faculty of Basic Siences
College of Technology, Economics and Industries

ABSTRACT:

In recent decades, the number and size of databases have increased rapidly. It is expected that the global data doubles its size every two years. Data is becoming a really valuable resource. However, it is difficult to extract useful information due to the vast amount of data.

Keywords: Decision tree, databases, data mining, capacity.