TRƯỜNG ĐẠI HỌC TÂY ĐÔ
KHOA KỸ THUẬT CÔNG NGHỆ

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

KHAI MỎ DỮ LIỆU VÀ KHÁM PHÁ TRI THỨC

Sinh viên thực hiện:
Họ và tên: Quách Luyl Đa
MSSV: 0751010009
Lớp: Đại học Tin học 2

Cán bộ hướng dẫn:
Ths. Dương Văn Hiếu

Cần Thơ, 2011

TRƯỜNG ĐẠI HỌC TÂY ĐÔ
KHOA KỸ THUẬT CÔNG NGHỆ

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

KHAI MỎ DỮ LIỆU VÀ KHÁM PHÁ TRI THỨC

Sinh viên thực hiện
Họ và tên: Quách Luyl Đa
MSSV: 0751010009
Lớp: Đại học Tin học 2

Cán bộ hướng dẫn
Ths. Dương Văn Hiếu

Cán bộ phản biện
Học hàm, học vị, họ và tên cán bộ phản biện

Luận văn được bảo vệ tại: Hội đồng chấm luận văn tốt nghiệp Bộ môn …….. …….. ..…..Khoa Kỹ Thuật Công Nghệ, Trường Đại học Tây Đô vào ngày .... tháng ..... năm ….

Mã số đề tài:

Có thể tìm hiểu luận văn tại:
- Thư viện: Trường Đại học Tây Đô.
- Website:

LỜI CÁM ƠN

"Đi khắp thế gian không ai tốt bằng Mẹ
Gánh nặng cuộc đời không ai khổ bằng Cha
Nước biển mênh mông không đong đầy tình Mẹ
Mây trời lồng lộng không phủ kín công Cha"

Khuyết danh Việt Nam

Đi khắp thế gian không ai tốt như mẹ, chăm lo cuộc sống cho con không ai bằng cha, gánh nặng ấy cha mẹ không nói ra, nhưng tôi có thể cảm nhận và biết được qua mái tóc bạc của mẹ, những giọt mồ hôi và làn da rám nắng của cha. Tất cả những việc làm của cha mẹ chỉ để cho gia đình được sống vui vẻ và hạnh phúc, cho anh em tôi được ăn học nên người. Gánh nặng ấy càng gia tăng và nặng nề hơn trên vai cha và trong mắt mẹ khi con bước vào ngưỡng của đại học. Với bao lo lắng từ cái ăn, cái mặc, việc học hành càng làm tăng gánh nặng cho cha mẹ. Gánh nặng ấy không thể thấy được trong tiếng cười của cha, trong ánh mắt và tiếng nói của mẹ. Tôi xin gửi lời cám ơn và kết quả học tập trong những năm tháng học xa nhà để làm món quà dâng tặng lên cha mẹ của tôi!

Tôi xin chân thành cám ơn quý thầy cô trong ban giám hiệu, các thầy cô trong khoa và các thầy cô trực tiếp giảng dạy chúng tôi, và đã cung cấp cho chúng tôi được những kiến thức, những kỹ năng cần thiết trong cuộc sống và chuyên môn. Từ đó có thể vận dụng vào trong học tập và quá trình nghiên cứu. Đặc biệt, tôi xin chân thành cám ơn thầy Dương Văn Hiếu, mặc dù có nhiều khó khăn về mặt địa lý và công việc, nhưng thầy đã tạo mọi điều kiện để hướng dẫn chúng tôi hoàn thành khóa luận này. Tôi xin chân thành cám ơn!

Khoảng thời gian theo học tại trường, với những lạ lẫm khi vừa bước vào môi trường mới, với nhiều bạn mới. Chính những người bạn cùng đồng hành với tôi trên bước đường đại học, với những lý tưởng và tính cách khác nhau. Chính những câu chuyện buồn – vui – giận – ghét và chính những sự giúp đỡ trong học tập và công tác, các bạn đã giúp tôi thêm trưởng thành hơn, trao dồi được nhiều kiến thức hơn từ các bạn. Tôi xin chân thành cám ơn!

Và lời cảm ơn sau cùng, xin chân thành cám ơn các anh chị, và cũng là những người bạn, các cô chú nhân viên trường đã giúp đỡ và quan tâm tôi trong suốt thời gian theo học tại trường. Tôi xin chân thành cám ơn!

Xin chân thành cám ơn!

MỤC LỤC

BẢNG KÝ HIỆU VÀ VIẾT TẮT

STT	TỪ VIẾT TẮT	TIẾNG ANH	NGHĨA TIẾNG VIỆT
01	ANN	Artifical neural network	Mạng thần kinh nhân tạo
02	AND, DNA	Acid DeoxyriboNucleic	Phần tử nucleotic a xít
03	GA	Genetic Algorithm	Giải thuật di truyền
04	GUI	Graphical user interface	Giao diện đồ họa người dùng
05	Item	Item	Món hàng, mục,..
06	Itemset	Itemset	Tập các mục, các hàng,…
07	KNN	K Nearest neighbour	K-lân cận gần nhất
08	KDD	Knowledge Discovery in Databases	Khám phá tri thức từ dữ liệu
09	RAM	Ram memory	Bộ nhớ ram
10	XML	Extensible Markup Language	Ngôn ngữ đánh dấu mở rộng
11	web	website, web page	Trang web

TÓM TẮT

Sự bùng nổ thông tin ngày càng lan rộng và nhanh chóng, bên cạnh dữ liệu ngày càng gia tăng về số lượng. Các nhà khoa học đã nghiên cứu về khả năng sử dụng những dữ liệu ấy để phục vụ nhu cầu kinh doanh, học tập và nghiên cứu. Việc khai thác dữ liệu dựa trên những dữ liệu đã tồn tại được gọi là khai phá dữ liệu (Data mining). Quá trình khai phá dữ liệu là bước ngoặc quan trọng cho quá trình khám phá tri thức từ dữ liệu (Knowledge Discovery in Databases).

Dựa trên dữ liệu về khai phá dữ liệu và khám phá tri thức từ dữ liệu văn bản (text mining), luận văn đi sâu vào việc tìm hiểu về quá trình khai phá dữ liệu bao gồm: tiền xử lý dữ liệu, các phương pháp khai phá dữ liệu làm nền tản, chương trình khai phá dữ liệu, lập trình xử lý 1 số thuật toán cơ bản của phương pháp khai phá dữ liệu bằng luật kết hợp và cây quyết định,..

Tuy nhiên, đề tài chưa đi khai thác được hết các khía cạnh của khai phá dữ liệu từ hình ảnh (Image mining), web (web mining),…Các phương pháp khai phá dữ liệu khác.

ABSTRACT

The explosion of information becomes more widely and quickly, besides increasing the data quantity. Scientists have been studying the possibility of using that data to serve the needs of business, learning and research activities. Mining based on historical data is called data mining. The data mining process is an very important landmark for the process of discovering knowledge from data.

In this study, we focus on understanding the data mining process including data preprocessing, common data mining techniques, data-mining programs. And, implementing the basic methods of data mining such as association rule and decision tree, …

However, the topic is not going to exploit every aspect of data-mining from image (Image mining), web (web mining), ... The data-mining methods other.

Chương I : **TỔNG QUAN**

I.1. ĐẶT VẤN ĐỀ

Ngày nay, công nghệ thông tin đã trở thành một trong những động lực quan trọng của sự phát triển. Với khả năng số hóa mọi thông tin (số, đồ thị, văn bản, hình ảnh, âm thanh, tiếng nói,…), máy tính đã trở thành một công cụ thông minh, nó được sử dụng để xử lý thông tin với nhiều dạng thông tin thuộc nhiều lĩnh vực khác nhau trong đời sống như: kinh doanh, y học,…

Bên cạnh đó, cùng với sự phát triển của công nghệ lưu trữ dữ liệu phục vụ trong công việc lưu trữ các thông tin liên quan đến nhiều mặt của cuộc sống: kinh doanh, buôn bán, … đã góp phần cải thiện cuộc sống và làm giảm bớt đi việc lưu trữ thông tin dựa trên văn bản.

Đó chính là tiền đề cho sự ra đời của nền kinh tế mới – nền kinh tế số (hay có thể gọi là nền kinh tế tri thức, nền kinh tế dựa trên tri thức). Nền kinh tế đấy đã và đang làm cho sự phát triển thông tin lưu trữ ngày càng nhiều, và khả năng linh hoạt của các phần mềm phải đảm đương nhiều công việc trong việc lựa chọn thông tin. Và trong những năm 1980, một số nhà nghiên cứu đã đưa một số kỹ thuật nhằm giải quyết các vấn đề trên, và được gọi là kỹ thuật khai phá dữ liệu (data mining).

Các kỹ thuật khai phá dữ liệu đã được các công ty kinh doanh các sản phẩm liên quan đến thông tin đã ứng dụng như:

- Duyệt web, tìm kiếm các thông tin trên Google, Google luôn đưa ra các gợi ý, có lẽ bạn sẽ nghĩ: nó đã đọc được những suy nghĩ của mình! Mà đa phần các gợi ý đấy gần như là các thông tin mà bạn cần tìm kiếm. Vì sao Google biết mình cần tìm thông tin đấy?
- Facebook, nhắc đến Facebook bạn sẽ nghĩ đến một cộng đồng với số lượng thông tin cá nhân được lưu trữ với số lượng lớn, phải nói là rất lớn. Khi bạn muốn kết bạn trên cộng đồng ấy, Facebook luôn đưa ra những gợi ý về những người bạn cho bạn kết bạn. Và những người bạn ấy gần như bạn đã quen biết ngoài cuộc sống đời thường. Bạn nghĩ tại sao nó có thể làm như vậy?
- Một ví dụ khác, đó là việc tìm và mua 1 quyển sách trên cửa hàng sách trực tuyến khổng lồ Amazon. Khi lựa chọn một quyển sách, nó luôn đưa ra cho bạn các lựa chọn về những quyển sách mà 90% là bạn cần mua. Vậy tại sao nó hiểu bạn nhiều như thế?

Và câu hỏi cuối cùng, việc xử lý thông tin của nó ra sao? Tất cả những câu hỏi đấy là một ứng dụng cụ thể của khai phá dữ liệu và khám phá tri thức. Vậy khai phá dữ liệu là gì?

I.2. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ

" Data mining là quá trình thăm dò, lựa chọn và mô hình hóa khối lượng lớn dữ liệu để tìm ra những quy luật hoặc các mối quan hệ chưa biết đầu tiên với mục đích là để có được kết quả rõ ràng và hữu ích cho các chủ sở hữu của cơ sở dữ liệu."

Qua quá trình phát triển, định nghĩa về khai phá dữ liệu ngày càng được mở rộng, và dần dần hoàn thiện:

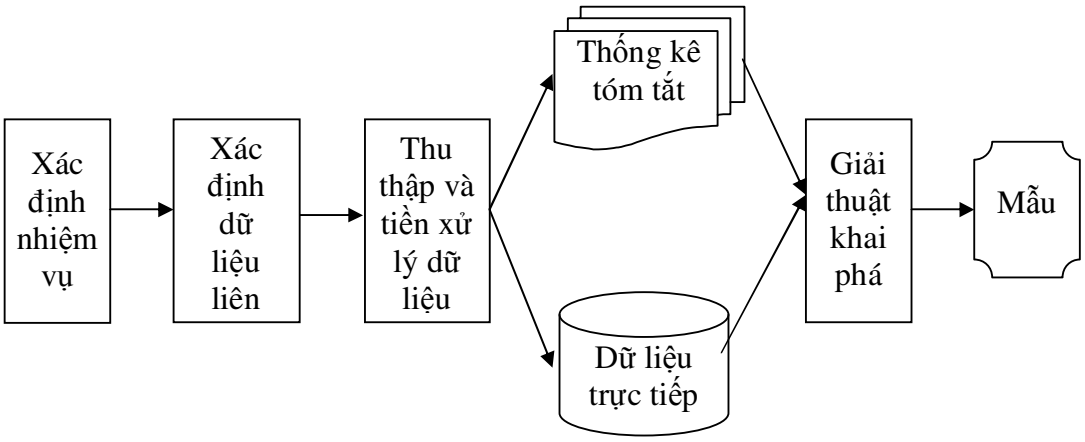
- Khai phá dữ liệu là quá trình khám phá thông tin hữu dụng trong các kho dữ liệu khổng lồ một cách tự động. Các kỹ thuật khai phá dữ liệu được triển khai dựa trên các cơ sở dữ liệu lớn nhằm tìm kiếm các mẫu hay các quy luật (pattern) mới và hữu dụng mà chưa từng được biết trước đó. Ví dụ: "Những sinh viên học giỏi các môn Toán rời rạc, Lập trình, Cấu trúc dữ liệu và Cơ sở dữ liệu thì sẽ học giỏi môn khai phá dữ liệu"

- Khai phá dữ liệu là quá trình tìm kiếm các mẫu mới, những thông tin, tri thức có ích, tiềm ẩn và mang tính dự đoán trong khối lượng dữ liệu lớn.

Các kỹ thuật khai phá dữ liệu cũng cung cấp các khả năng phán đoán (dự đoán) kết quả của các quan sát trong hiện tại và quá khứ. Khai phá dữ liệu không chỉ khám phá các thông tin hữu dụng trong các cơ sở dữ liệu (databases) hay kho dữ liệu (data respostories) mà còn liên quan đến các lĩnh vực truy xuất thông tin (information retrieval).

Ví dụ: Sử dụng hệ quản trị cơ sở dữ liệu để tìm kiếm các mẫu tin hoặc sử dụng các công cụ tìm kiếm trên Internet để tìm kiếm các trang web hoặc thông tin được lưu trữ ở các trang web cụ thể nào đó.

Data mining là 1 phần hoàn chỉnh của lĩnh vực khám phá tri thức (Knowledge Discovery). Nó là toàn bộ quá trình chuyển dữ liệu thô sang thông tin hữu dụng. Quá trình này gồm nhiều bước tiền xử lý dữ liệu đến hậu xử lý kết quả của quá trình khai phá.



Hình I-2. Quá trình khai phá dữ liệu

Các khó khăn trong việc khai thác tri thức từ dữ liệu:

a) Tính qui mô:

Với sự phát triển trong việc tạo ra dữ liệu cũng như thu thập dữ liệu, các tập hợp dữ liệu được lưu trữ ngày càng lớn (gigabytes, terabytes,petabytes) và ngày càng trở nên thông dụng. Các thuật toán khai phá dữ liệu phải có khả năng phân tích được các tập dữ liệu đó. Nhiều kỹ thuật khai phá dữ liệu triển khai các chiến lược nghiên cứu đặc biệt nhằm quản lý các vấn đề trong nghiên cứu tăng theo cấp

số nhân. Tính qui mô (scalability) yêu cầu phương pháp cài đặt của cấu trúc dữ liệu mới nhằm truy xuất được các mẩu tin một cách hiệu quả.

Ví dụ: Các thuật toán "xử lý dữ liệu ngoài bộ nhớ (RAM)" (out-of-core) rất cần thiết khi xử lý các tập dữ liệu lớn hơn dung lượng của bộ nhớ. Tính qui mô có thể được cải tiến bằng cách sử dụng các dữ liệu mẫu (samples), sử dụng các giải thuật song song và phân tán.

b) Tính đa thuộc tính:

Xử lý các tập dữ liệu có hàng trăm hay hàng nghìn thuộc tính ngày càng trở nên phổ biến. Trong lĩnh vực tin học cho sinh học, dữ liệu về gen có thể bao gồm hàng ngàn thuộc tính. Các tập dữ liệu với các thành phần dữ liệu theo thời gian hay còn được gọi là dữ liệu tuần tự (temporal/ spatial components) cũng có xu hướng có rất nhiều thuộc tính.

Ví dụ: Tập dữ liệu chứa các thông tin về địa chất ở nhiều khu vực khác nhau được thu thập lặp đi lặp lại nhiều lần, số lượng các thuộc tính có thể tăng dần theo thời gian. Các kỹ thuật phân tích dữ liệu truyền thống được thiết kế cho dữ liệu có ít thuộc tính không thể áp dụng cho trường hợp dữ liệu có nhiều thuộc tính.

c) Dữ liệu không thuần nhất và phức tạp:

Các phương pháp phân tích dữ liệu truyền thống áp dụng cho các tập hợp dữ liệu chứa các thuộc tính có cùng kiểu dữ liệu (có thể là liên tục hay rời rạc). Khi việc sử dụng khai phá dữ liệu trong kinh doanh, trong khoa học và trong y học ngày càng tăng thì cần có các kỹ thuật phân tích dữ liệu có thể áp dụng được cho các thuộc tính không thuần nhất (heterogeneous attributes). Bên cạnh đó, cũng phải áp dụng được cho các dữ liệu phức tạp.

Ví dụ: Các kiểu dữ liệu truyền thống bao gồm: tập hợp các trang web lưu văn bản và liên kết bán cấu trúc, các dữ liệu về DNA trong không gian 3 chiều, dữ liệu về thời tiết (nhiệt độ, áp suất, độ ẩm) tại nhiều vùng trên thế giới. Các kỹ thuật được phát triển cho khai phá dữ liệu cần phải quan tâm đến mối quan hệ trong dữ liệu như: mối quan hệ về nhiệt độ theo thời gian, sự liên thông giữa các đồ thị, quan hệ giữa các thành phần trong dữ liệu bán cấu trúc và XML.

d) Sở hữu và phân bố dữ liệu:

Có khi dữ liệu cần được phân tích được lưu trữ ở nhiều nơi khác nhau và được sở hữu bởi nhiều cơ quan khác nhau. Các khó khăn này đòi hỏi phải phát triển các kỹ thuật khai phá dữ liệu theo dạng phân tán. Vấn đề cần quan tâm là "làm sao hạn chế lưu lượng truyền tải dữ liệu khi thực hiện các thuật toán phân tán?", "làm sao hợp nhất dữ liệu từ các nguồn gốc khác nhau một cách hiệu quả nhất?", "làm sao đảm bảo tính an toàn và bảo mật?",…

e) Việc phân tích dữ liệu không theo cách truyền thống:

Cách tiếp cận dữ liệu theo phương pháp thống kê truyền thống dựa trên cách đặt giả thuyết và kiểm tra giả thuyết cần rất nhiều công sức để kiểm tra các giả thuyết. Các công việc phân tích dữ liệu hiện tại đòi hỏi phải đặt và kiểm tra hàng nghìn giả định một cách tuần tự. Quá trình phát triển các kỹ thuật khai phá dữ liệu đã được thúc đẩy bởi sự mong đợi một quá trình đặt và kiểm tra giả định một cách hoàn toàn tự động. Hơn nữa, dữ liệu được phân tích trong khai phá dữ

liệu là dữ liệu ngẫu nhiên nên các phương pháp phân tích truyền thống không thể áp dụng cho các tập dữ liệu phức tạp và mang tính ngẫu nhiên.

I.3. PHẠM VI CỦA ĐỀ TÀI

Đề tài đi sâu nghiên cứu về quá trình khai phá dữ liệu và khám phá tri thức từ dữ liệu. Qua việc nghiên cứu có thể tìm hiểu thêm về các kỹ thuật cơ bản trong việc tiền xử lý dữ liệu, các kỹ thuật khai phá dữ liệu cơ bản và từ đó có được những kiến thức trong việc tìm hiểu một công cụ khai phá dữ liệu, xây dựng demo khai phá dữ liệu dựa trên một số thuật toán của cây quyết định và luật kết hợp. Từ quá trình nghiên cứu và thực tiễn để có thể thấy được các vấn đề thách thức trong lĩnh vực khai phá dữ liệu.

Sử dụng cơ sở lý thuyết đã nghiên cứu để cài đặt chương trình sinh luật kết hợp và cây quyết định là 2 kỹ thuật cơ bản của quá trình khai phá dữ liệu.

I.4. PHƯƠNG PHÁP NGHIÊN CỨU

Dựa trên việc tìm hiểu các tư liệu trong lĩnh vực khai phá dữ liệu, từ đó rút ra được những kết quả của quá trình tiền xử lý dữ liệu, một số kỹ thuật khai phá dữ liệu cơ bản cùng với các thuật toán của nó. Để có được những hiểu biết về quá trình khai phá dữ liệu và khám phá tri thức.

Dựa trên quá trình tìm hiểu về khai phá dữ liệu, tiếp cận một công cụ khai phá dữ liệu, để chứng minh cho các thuật toán và giải thuật đã nghiên cứu.

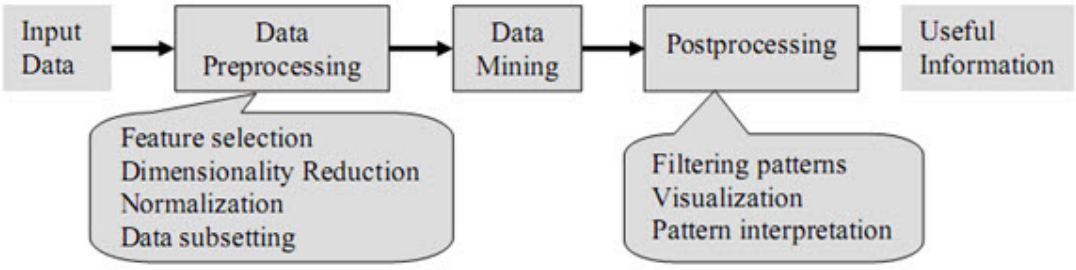
Tổng hợp các dữ liệu đã tìm hiểu, minh họa một thuật toán cơ bản trong việc khai phá dữ liệu bằng cây quyết định và luật kết hợp bằng demo cụ thể. Demo sử dụng ngôn ngữ lập trình Microsoft Visual Basic 2008 để xây dựng các thuật toán.

Chương II : **CƠ SỞ LÝ THUYẾT**

II.1. KHÁI NIỆM VỀ KHAI PHÁ DỮ LIỆU

II.1.1. Khái niệm:

Khai phá dữ liệu (Data mining) là một một bước trong quá trình khám phá tri thức trong cơ sở dữ liệu (Knowledge Discovery in Databases – KDD).



Hình II-1. Quá trình khám phá tri thức trong cơ sở dữ liệu

Tại hội nghị quốc tế lần thứ nhất về Khám phá tri thức và Khai phá dữ liệu (Knowledge Discovery and Data mining, được tổ chức ở Motreal vào năm 1995, Usama Fayaad đã đưa ra khái niệm chính thức về Data mining. Nó được sử dụng để chỉ một tập hợp các kỹ thuật phân tích được chia làm nhiều giai đoạn khác nhau, với mục tiêu kiến thức trước đây chưa biết sẽ được suy luận từ kho dữ liệu khổng lồ, mà dường như không có bất cứ một quy luật hoặc mối quan hệ rõ ràng nào. Khi thuật ngữ "Data mining" từ từ được hình thành, nó đã trở thành kiến thức dành cho việc suy luận. Điều này hết sức hữu ích vì đã bác bỏ những khía cạnh – mục đích cuối cùng của khai khoáng dữ liệu còn mơ hồ trước đó. Mục tiêu của khai khoáng dữ liệu là thu được kết quả có thể đo bằng mức độ phù hợp của dữ liệu cho các chủ sở hữu cơ sở dữ liệu–kinh doanh được thuận lợi.

Khai phá dữ liệu (Data mining) là một quá trình khám phá thông tin hữu dụng trong kho dữ liệu khổng lồ một cách tự động. Các kỹ thuật khai phá dữ liệu được triển khai trên các cơ sở dữ liệu lớn nhằm tìm kiếm các mẫu hay các qui luật (pattern) mới và hữu dụng mà chưa từng được biết trước đó. Ví dụ: Người ta thường mua đường khi mua đậu xanh, những sinh viên học giỏi các môn Toán rời rạc, lập trình, cấu trúc dữ liệu và cơ sở dữ liệu thì sẽ học giỏi môn khai phá dữ liệu.

Các kỹ thuật khai phá dữ liệu cũng cung cấp khả năng phán đoán (dự đoán) kết quả của các quan sát trong tương lai dựa vào dữ liệu hiện tại và quá khứ. Khai phá dữ liệu không chỉ là khám phá các thông tin hữu dụng trong các cơ sở dữ liệu (databases) hay kho dữ liệu (data repositories) mà còn bao gồm các công việc liên quan đến lĩnh vực truy xuất thông tin (information retrieval).

Theo sơ đồ Quá trình khám phá tri thức trong cơ sở dữ liệu (Hình II-1), ta có một số khái niệm như sau:

- Input Data: dữ liệu đầu vào, nó có thể được lưu trữ với dưới nhiều định dạng khác nhau (file text, file bảng tính, các bản quan hệ) và được lưu trữ trong kho dữ liệu tập trung hoặc phân tán nhiều nơi khác nhau.

- Data Preprocessing: Quá trình tiền xử lý dữ liệu bao gồm phân rã (puse) dữ liệu từ nhiều nguồn dữ liệu khác nhau, làm sạch (clean) dữ liệu bằng cách loại bỏ nhiễu và dữ liệu trùng nhau, lựa chọn các mẫu tin (record) và các đặc tính (feature) có liên quan đến quá trình khai thác (mine) dữ liệu. Trong thực tế, dữ liệu có thể được thu nhập và lưu trữ bằng nhiều cách khác nhau nên quá trình tiền xử lý dữ liệu là một quá trình hết sức quan trọng, khá nặng nhọc và tiêu tốn nhiều thời gian cũng như công sức.

- Postproccessing: Hậu xử lý kết quả là quá trình loại bỏ các kết quả không phù hợp hay lựa chọn các kết quả phù hợp với các công việc và nhu cầu thực tế. Các kết quả sau bước hậu xử lý sẽ được sử dụng cho các hệ thống hỗ trợ ra quyết định (Decision Support System).

II.1.2. Nhiệm vụ của khai thác dữ liệu:

Khai phá dữ liệu có 2 nhiệm vụ lớn là dự đoán và mô tả:

II.1.2.1. Nhiệm vụ dự đoán:

Mục đích của nhiệm vụ dự đoán là dự đoán giá trị của một thuộc tính cụ thể dựa trên giá trị của các thuộc tính khác. Thuộc tính được dự đoán được gọi là thuộc tính mục tiêu (target attributed) hay thuộc tính phụ thuộc (dependent variables/ attributed), thuộc tính dùng để tạo dự đoán gọi là thuộc tính mô tả hay thuộc tính độc lập (explanatory/ Indepent variables).

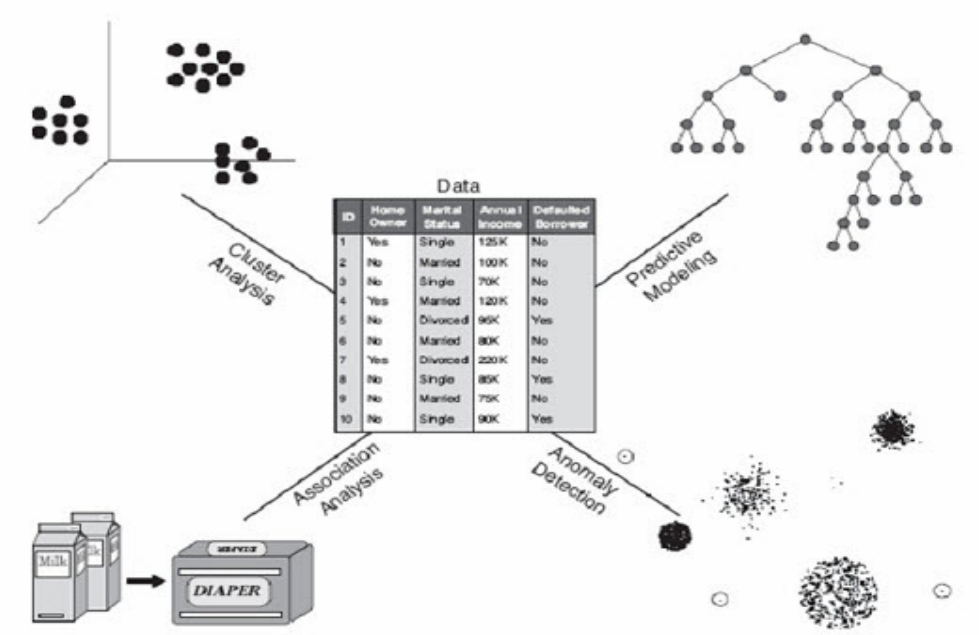
Ví dụ: Để quyết định việc cấp học bổng cho sinh viên đại học và sau đại học, người ta có thể dựa vào rất nhiều yếu tố cũng như tiêu chí khác nhau. Một trong những tiêu chí đó là khả năng thành công trong học tập của người sẽ được cấp học bổng. Làm thế nào để ước lượng được khả năng học tập của các ứng viên xin học bổng? Người/ tổ chức cấp học bổng có thể sử dụng các thông tin về sinh viên như: giới tính, độ tuổi, hoàn cảnh gia đình, tình trạng hôn nhân, nghề nghiệp.

II.1.2.2. Nhiệm vụ mô tả:

Mục đích của nhiệm vụ mô tả là lấy ra từ các mẫu (pattern) mang tính mô tả như: sự tương quan (correlation), xu hướng (trend), nhóm (cluster), đường di chuyển (trajectory) và ngoại lệ. Các mẫu này nói lên mối quan hệ giữa dữ liệu. Nhiệm vụ của phần này thường là giải thích về mặt bản chất và thường yêu cầu các kỹ thuật hậu xử lý (postproccessing) nhằm xác nhận (validate) và giải thích (explain) các kết quả.

II.1.2.3. Nhiệm vụ trọng tâm của khai phá dữ liệu:

Nhiệm vụ trọng tâm của khai báo dữ liệu là: mô hình hóa cho việc dự báo, phân tích và nhóm các đối tượng dữ liệu thành từng nhóm dựa trên những thuộc tính của chúng, phân tích và đưa ra các luật kết hợp dựa trên các dữ liệu hiện tại, phân tích và phát hiện các trường hợp ngoại lệ. Bốn nhiệm vụ ấy có thể được mô tả ngắn gọn như sau:



Hình II-2. Bốn nhiệm vụ trọng tâm của khai phá dữ liệu

a) Mô hình hóa cho việc dự báo:

Nhiệm vụ chính là xây dựng mô hình cho thuộc tính cần được dự đoán giá trị (target variable) như là một hàm của các biến độc lập (independent variable) được dùng để đoán giá trị cho target variable. Có hai kiểu mô hình dự báo (predictive modeling), đó là: phân lớp dữ liệu (classification analysis) và hồi quy (regression). Sự phân lớp dữ liệu được sử dụng cho các thuộc tính target có giá trị rời rạc. Sự hồi quy được sử dụng cho các thuộc tính target có giá trị liên tục.

Ví dụ 1: Dự đoán một người dùng Internet sẽ mua hàng trực tuyến hay không thì phải sử dụng phương pháp phân lớp vì giá trị của thuộc tính target rời rạc ("mua" và "không").

Ví dụ 2: Dự đoán giá cổ phiếu trong tương lai thì phải sử dụng phương pháp regession vì giá trị của cổ phiếu là giá trị liên tục.

Mục đích của cả phân lớp và hồi quy là tìm ra mô hình để dự đoán giá trị của một thuộc tính dựa trên các thuộc tính khác sao cho tối thiểu quá sai khác giữa các dự đoán và giá trị thực tế.

b) Phân tích kết hợp:

Phân tích kết hợp dùng để khám phá các mẫu (pattern) mà các mẫu này mô tả một cách mạnh mẽ các mối quan hệ giữa các đặc điểm của dữ liệu. Các mẫu qui luật được khám phá thông thường được biểu diễn bằng luật kết hợp. Bởi vì kích thước của không gian tìm kiếm tăng lên theo cấp số nhân nên mục đích chính của phương pháp phân tích kết hợp là kết xuất các mẫu có ý nghĩa bằng cách làm hiệu quả hay nói cách khác là phải "loại bỏ các luật có giá trị sử dụng ít".

Ví dụ: Xét các giao dịch tại một cửa hàng như bảng bên dưới:

Mã giao dịch (Transaction ID)	Mặt hàng (Items)
1	{Bread, Butter, Diapers, Milk}
2	{Coffee, Sugar, Cookies, Salmon}
3	{Bread, Butter, Coffee, Diapers, Milk, Eggs}
4	{Bread, Butter, Salmon, Chicken}
5	{Eggs, Bread, Butter}
6	{Salmon, Diapers, Milk}
7	{Bread, Tea, Sugar, Eggs}
8	{Coffee, Sugar, Chicken, Egges}
9	{Bread,Diapers, Milk, Salt}
10	{Tea, Eggs, Cookies, Diapers, Milk}

Hình II-3. Một số giao dịch tại cửa hàng

Phân tích lớp dữ liệu có thể được dùng để phân tích và tìm các mặt hàng được mua cùng với nhau để "bố trí các mặt hàng sao cho khách hàng mua hàng thuận tiện nhất".

c) Phân tích nhóm:

Phương pháp phân tích nhóm dùng để tìm các nhóm (groups) của các giá trị quan sát được (observations) có liên quan đến nhau. Các giá trị quan sát cùng một nhóm chắc chắn sẽ giống nhau nhiều hơn so với các giá trị ở các nhóm khác nhau.

Ví dụ: phân nhóm khách hàng để tìm ra các khách hàng có cùng sở thích mua sắm.

d) Phát hiện ngoại lệ:

Phát hiện các ngoại lệ là tìm các quan sát mà chúng khác rất nhiều so với các giá trị khác. Các giá trị khác biệt so với các giá trị khác được gọi là ngoại lệ (anomaly, outlier).

Ví dụ: Ứng dụng phương pháp phát hiện ngoại lệ để tìm các giao dịch "bất thường" trong lĩnh vực ngân hàng như: rửa tiền, gian lận khác trong giao dịch.

II.1.3. Ứng dụng của khai phá dữ liệu:

Từ khi ra đời, khai phá dữ liệu được ứng dụng rộng rãi, sau đây là một số ứng dụng cụ thể:

a) Thiên văn học: Xác định vị trí và hướng di chuyển của các chòm sao, các hành tinh trong hệ mặt trời dựa trên những dữ liệu về hướng di chuyển, lịch sử phát triển của nó,…

b) Phát hiện gian lận: Dựa trên những doanh thu, tài khoản phải thu, thu thập những dữ liệu hiệu quả biên của biên chế gian lận, kiểm toán tự động hoặc các kỹ thuật để phát hiện gian lận, sử dụng dữ liệu phân tích kết quả để kiểm soát biên chế phòng ngừa gian lận,…

c) Quản lý quan hệ bán hàng: Lưu trữ thông tin khách hàng, phân loại khách hàng, các thông tin mua hàng,…từ đó đưa ra các chiến lược, phương pháp kinh doanh mới nhằm mục đích:

- Khách hàng có lợi và những đặc điểm nào làm cho họ như vậy.

- Thay đổi trong hành vi mua của khách hàng – hoặc là một cơ hội hay đe dọa đối với kinh doanh.
- Những khoảng trống trong danh mục sản phẩm – cho biết qua việc bán, số lượng bán sản phẩm tăng, và lặp lại các lần mua hàng,..
- Những mặt hàng nào bố trí thuận lợi và tiện dụng cho khách hàng,..

d) Chăm sóc sức khỏe: Lưu trữ thông tin các bệnh, các hiện tượng, triệu chứng,…từ đó, dựa trên những thông tin ấy để phát hiện các bệnh và hướng điều trị cụ thể đối với các bệnh đã được phát hiện.

e) Nông nghiệp: Tìm kiếm các thông tin về rầy nâu, hướng di chuyển, lịch sử phát triển,… để đưa ra các biện pháp phòng và tránh rầy nâu phá hoại mùa màng.

f) Giáo dục: Dựa trên những thông tin về tập quán, nơi cư trú, điều kiện của xã hội, tính cách,…để đưa ra những định hướng trong việc lựa chọn nghề nghiệp cho học sinh vừa tốt nghiệp phổ thông để có hướng lựa chọn nghề nghiệp hợp lý,…

II.2. CÁC KHÁI NIỆM CƠ BẢN

II.2.1. Dữ liệu và kiểu dữ liệu:

Dữ liệu là phần tử hoặc tập hợp các phần tử mà ta gọi là tín hiệu. Nó được biểu hiện dưới các dạng như hình ảnh, âm thanh, màu sắc, mùi vị,…Từ những tín hiệu đó, chúng ta có sự hiểu biết về một sự vật, hiện tượng hay quá trình nào đó trong thế giới khách quan thông qua quá trình nhận thức.

Một tập hợp dữ liệu có thể được xem như một tập hợp các đối tượng dữ liệu. Các đối tượng dữ liệu có thể là mẫu tin (record), điểm (point), véc tơ (vector), mẫu (pattern), sự kiện (event), trường hợp (case), dữ liệu mẫu (sample), các thực thể (entity) và các kết quả quan sát (observation). Đối tượng dữ liệu được mô tả bằng các thuộc tính (attribute) mà các thuộc tính này nói lên tính chất / đặc điểm cơ bản của đối tượng dữ liệu. Trong ngữ cảnh khai phá dữ liệu, thuộc tính được gọi với những tên khác nhau như: Biến (variable), đặc trưng (characteristis), trường dữ liệu (field), tính năng (feature), kích thước (dimension).

II.2.1.1. Thuộc tính và phép đo:

a) Định nghĩa thuộc tính:

Thuộc tính là tính chất của một đối tượng mà giá trị của nó có thể khác nhau tùy vào từng đối tượng cụ thể.

Ví dụ: màu mắt, cân nặng, chiều cao là thuộc tính của con người, tùy vào từng người sẽ có giá trị khác nhau.

b) Phép tính độ đo:

Phép tính độ đo là một quy tắc (rule) hay một hàm (function) dùng để kết hợp một giá trị hoặc một ký hiệu với một thuộc tính của đối tượng, nhằm làm rõ tính chất của đối tượng.

Ví dụ: Xác định cân nặng bằng kg, chiều dài bằng mét, giới tính là nam hay nữ, số ghế trong phòng học là đủ hay thiếu,…

c) Kiểu của thuộc tính:

STT	Kiểu thuộc tính (Attributed type)	Mô tả	Ví dụ
1	Định danh (nominal)	Giá trị của thuộc tính kiểu nominal là các tên gọi hay định danh khác nhau, chỉ cung cấp vừa đủ thông tin để phân biệt giống nhau hay khác nhau ($=, \neq$).	Mã tỉnh, mã nhân viên, giới tính,..
2	Thứ tự (ordinal)	Giá trị thuộc tính kiểu ordinal cung cấp đầy đủ thông tin để phân biệt ($=, \neq$) và so sánh theo thứ tự (<, <=,>,>=).	Cao, cao hơn, cao nhất,…
3	Khoảng cách (interval)	Đối với thuộc tính kiểu interval, ngoài phân biệt cung cấp đầy đủ thông tin để phân biệt($=, \neq$), so sánh (<, <=,>,>=), sự khác nhau (+,-) giữa các giá trị là hết sức quan trọng.	Ngày tháng năm Độ C hoặc độ F
4	Tỷ lệ (ratio)	Đối với thuộc tính kiểu ratio, sự khác nhau (+,-) và tỉ lệ (*,/) giữa các giá trị là hết sức quan trọng	Số lượng, độ dài, tuổi,…

Trong đó, thuộc tính kiểu định danh và thứ tự được coi như thuộc tính dùng để phân biệt/ phân loại hay thuộc tính định danh. Thuộc tính kiểu khoảng cách và tỉ lệ được xem như là thuộc tính định lượng hay thuộc tính kiểu số.

d) Mô tả thuộc tính bằng tập hợp các giá trị:

Bằng cách dựa vào số lượng và giá trị mà thuộc tính có thể có, chúng ta có thể chia làm 3 loại thuộc tính:

- Thuộc tính nhị phân: có 2 giá trị. Thường được sử dụng với thuộc tính kiểu nhị phân, kiểu yes/no. Ví dụ: 0 và 1.

- Thuộc tính rời rạc là thuộc tính có một tập hợp hữu hạn các giá trị, có nhiều hơn 2 giá trị. Thường được sử dụng với thuộc tính kiểu số nguyên, kiểu ký tự, kiểu chuỗi ký tự. Ví dụ: mã tỉnh, số điện thoại, giới tính, số chứng minh nhân dân,…

- Thuộc tính liên tục: là thuộc tính có một tập vô hạn các giá trị liên tục hay có giá trị là các số thực, có vô hạn các giá trị. Thuộc tính liên tục thường được sử dụng là thuộc tính kiểu số thực hay số có dấu chấm động.

II.2.1.2. Kiểu của tập dữ liệu:

Có rất nhiều kiểu dữ liệu được sử dụng trong lĩnh vực khai phá dữ liệu khi có càng nhiều các tập dữ liệu được sử dụng để phân tích. Kiểu dữ liệu có thể được chia ra làm 3 nhóm lớn:

- Dữ liệu mẫu tin (record data).

- Dữ liệu trên cơ sở đồ thị (graph-based data).
- Dữ liệu có thứ tự (ordered data).

a) Tính chất tổng quát của các tập dữ liệu: Có 3 tính chất quan trọng ảnh hưởng đến việc lựa chọn và sử dụng các kỹ thuật khai phá dữ liệu là:

- Số chiều (dimensionality): Số chiều của 1 tập hợp dữ liệu là số lượng các thuộc tính mà các đối tượng trong tập dữ liệu đó sở hữu. Một trong những thách thức của lĩnh vực khai phá là dữ liệu có nhiều thuộc tính.

- Sự thưa thớt (sparsity): Đối với một số tập hợp như các thuộc tính không đối xứng. Hầu hết các thuộc tính của các đối tượng có giá trị 0 nhưng chỉ một số trường hợp không có giá trị 0. Trong thực tế, đây là một thuận lợi vì chỉ cần lưu trữ và thao tác trên các giá trị khác 0. Cách làm này sẽ làm giảm thời gian tính toán cũng như bộ nhớ lưu trữ.

- Độ phân giải (resolution): Trong khai phá dữ liệu, độ phân giải dữ liệu thường ở nhiều mức độ khác nhau và tính chất của dữ liệu cũng khác nhau tùy vào mức độ phân giải. Mẫu trong dữ liệu thường phụ thuộc nhiều vào mức độ phân giải. Ví dụ: Độ phân giải quá mịn thì mẫu sẽ bị mờ, độ phân giải quá thô thì mẫu sẽ mất.

b) Chi tiết về các kiểu dữ liệu trong khai phá dữ liệu:

❖ Dữ liệu dạng mẫu tin:

Hầu hết các trường hợp dữ liệu của khai phá dữ liệu là dạng mẫu tin (record data). Mỗi mẫu tin là một đối tượng dữ liệu bao gồm một tập hợp các thuộc tính. Các mẫu tin có thể được lưu trong các tập tin phẳng (flat files) hoặc lưu trong các bảng dữ liệu (table) trong cơ sở dữ liệu quan hệ. Dữ liệu dạng mẫu tin có thể là các bảng ghi trong cơ sở , giao dịch (transaction), ma trận dữ liệu (data matrix) và ma trận thuật ngữ trong văn bản (document – term matrix).

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

a) Dữ liệu mẫu tin

MSSV	Hình sự	Tích lũy số học phần	Trung bình học tập	Giáo dục quốc phòng	Giáo dục thể chất	Tốt nghiệp
0751010161	Không	Đạt	7.5	Đạt	Đạt	Đạt
0753040124	Có	Đạt	5.0	Đạt	Đạt	Không
0754010031	Không	Đạt	6.5	Đạt	Không	Không
0756020067	Không	Đạt	5.0	Đạt	Đạt	Đạt
06503141	Không	Đạt	5.5	Đạt	Đạt	Đạt
0751080024	Không	Đạt	7.0	Không	Đạt	Không
0756050046	Không	Không	6.0	Đạt	Đạt	Không

b) Ma trận dữ liệu

IDcustomers	Items
1	Bread, coke, milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Hình II-4. Các đối tượng của dữ liệu dạng mẫu tin

❖ Dữ liệu dựa trên đồ thị:

Đồ thị được coi như là 1 công cụ rất mạnh và rất thuận lợi cho việc biểu diễn dữ liệu vì nó có thể mô tả được mối quan hệ giữa các thành phần dữ liệu. Các đối tượng dữ liệu biểu diễn bằng các nút trên đồ thị còn mối quan hệ giữa các đối tượng thì được biểu diễn bằng các đường liên kết giữa các nút. Mối quan hệ giữa các đối tượng thường nói lên thông tin quan trọng về dữ liệu.

a) Các trang web được liên kết với nhau b) Cấu trúc ADN

Hình II-5. Các đối tượng dữ liệu dựa trên đồ thị

❖ Dữ liệu có thứ tự:

Trong một số trường hợp, các thuộc tính của dữ liệu mẫu tin có các mối quan hệ về mặt thời gian cũng như không gian. Dữ liệu như vậy được gọi là dữ liệu có thứ tự. Dữ liệu có thứ tự bao gồm:

- Dữ liệu được sắp xếp liên tục theo thời gian: Đây là một dạng mở rộng của dữ liệu dạng mẫu tin. Không chỉ mẫu tin và từng thuộc tính của mẫu tin cũng có sự kết hợp với thời gian (thời điểm). Ví dụ: Dữ liệu về giao dịch của khách hàng tại từng thời điểm như sau:

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

Hình II-6. Bảng dữ liệu giao dịch theo thời gian

Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Hình II-7. Bảng dữ liệu giao dịch theo thời gian (tiếp theo)

- Dữ liệu dạng chuỗi: (sequence data) là một tập hợp dữ liệu mà nó là một chuỗi các thực thể đơn lẻ giống như 1 chuỗi các con số, ký tự hay từ khóa. Rất giống với kiểu dữ liệu theo thời gian nhưng không liên quan đến thời gian (thời điểm). Ví dụ về thông tin di chuyển của loài động hay thực vật được biểu diễn như 1 chuỗi nucleotide được gọi là lag gene.

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

Hình II-8. Chuỗi dữ liệu

- Time series data: Là một dạng đặc biệt của dữ liệu dạng chuỗi, mỗi mẫu tin là 1 time series. Nói cách khác, mỗi mẫu tin là một chuỗi các giá trị đo được tại các thời điểm.

- Spatial data: Một số đối tượng có thuộc tính liên quan đến không gian hay vị trí. Ví dụ: Dữ liệu về thời tiết tại các vị trí khác nhau trên trái đất.

II.2.2. Chất lượng của dữ liệu:

Khai phá dữ liệu thường sử dụng được thu nhập cho những mục đích khác hoặc cho việc sử dụng trong tương lai hoặc không rõ ứng dụng cụ thể. Chính vì vậy mà chất lượng dữ liệu là một vấn đề cần quan tâm khi khai thác chi thức từ dữ liệu. Vì vậy, trước khi sử dụng, dữ liệu phải được xử lý để loại bỏ nhiễu, cũng như loại bỏ dữ liệu trùng nhau và dữ liệu vô ích không thể phục vụ cho công việc khai phá dữ liệu hiện tại. Vấn đề được trình bày tiếp theo là vấn đề liên quan đến chất lượng dữ liệu.

II.2.2.1. Độ đo chất lượng và vấn đề thu thập dữ liệu:

Trong cuộc sống không có gì là hoàn hảo, dữ liệu được thu nhập cũng để xử lý cũng gặp nhiều vấn đề khác nhau và không đảm bảo được chất lượng của dữ liệu phục vụ cho quá trình khai phá dữ liệu. Các vấn đề có thể là:

- Giá trị của một hoặc nhiều thuộc tính của một hoặc đối tượng có thể bị thiếu.
- Dữ liệu bị trùng lắp nhiều lần.

Nguyên nhân của những vấn đề trên có thể đến từ:

- Lỗi của con người, có thể nói đến lỗi của người thu thập dữ liệu hay tác động trực tiếp đến dữ liệu.
- Sự giới hạn của các thiết bị đo, có thể do đơn vị đo và khoảng cách quá chênh lệch.
- Lỗi trong quá trình thu thập dữ liệu.

Vì thế, vấn đề liên quan đến chất lượng của dữ liệu, chúng ta cần quan tâm đến các vấn đề sau:

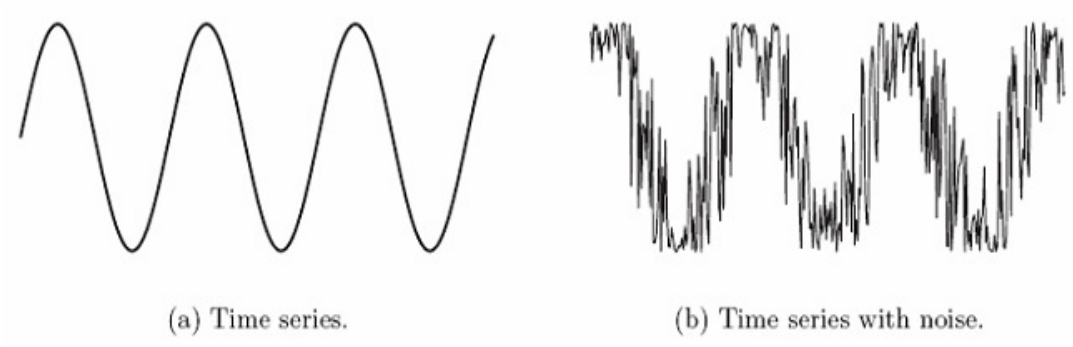
a) Lỗi đo lường và thu thập dữ liệu:

Lỗi của sự đo lường có thể đến từ các thiết bị hay chính sự tác động trực tiếp của con người. Nó được sinh ra do quá trình đo lường. Các lỗi có thể xảy ra do những nguyên nhân sau:

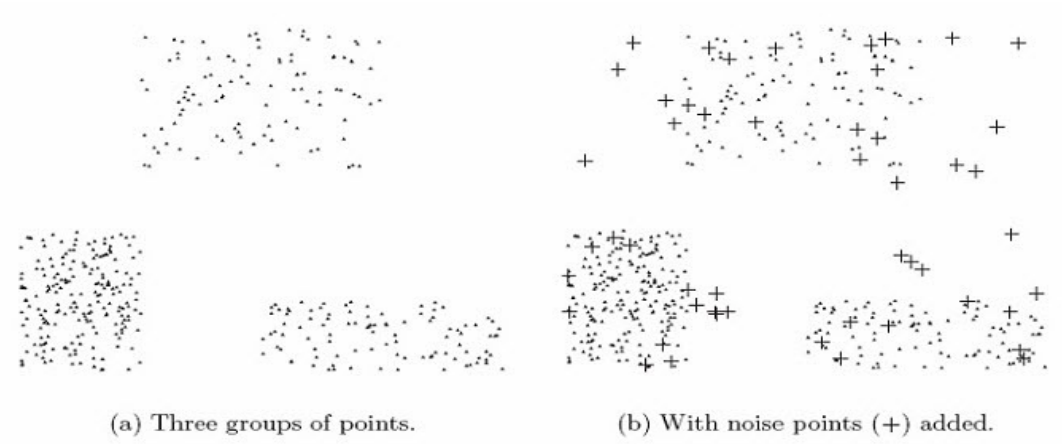
- Giá trị được lưu khác với giá trị thực. Ví dụ: Năng lượng ion hóa nguyên tử hirô là 13,6 eV, do quá trình ghi chép và lưu trữ trong thiết bị có thể là 13,9eV; 14eV; 13eV;…
- Do phương pháp đo không phù hợp hoặc thiết bị đo hay điều kiện đo không phù hợp. Ví dụ: Trong việc đo huyết áp của bệnh nhân, tư thế đo : nằm hoặc ngồi, sử dụng thiết bị đo cánh tay đo ở cổ tay hoặc ngược lại, uống cà phê trước khi đo,… cũng ảnh hưởng đến chất lượng của mỗi lần đo huyết áp.
- Lỗi thu thập dữ liệu xảy ra do quá trình thu thập dữ liệu và thường là bỏ qua một số đối tượng dữ liệu hay thuộc tính, thu thập dữ liệu không đúng. Ví dụ: Xác định bệnh sốt rét cần có các thông tin: lượng bạch cầu trong máu, thay đổi dịch não tủy, giảm canxi trong máu, hạ natri máu, bệnh nhân suy thận,…nếu thiếu 1 trong các triệu chứng và các xét nghiệm trên thì không thể có kết luận cụ thể về bệnh.

b) Nhiễu và dữ liệu bị bóp méo:

Nhiễu được hiểu là thành phần ngẫu nhiên của lỗi đo lường dữ liệu. Lỗi này có thể làm cho dữ liệu bị biến dạng, bị đưa vào các đối tượng giả mạo. Lỗi thường gặp đối với các đối tượng là âm thanh, số lượng dữ liệu quá lớn không thể gom nhóm được,…Vấn đề nhiễu và dữ liệu bị bóp méo có thể do những nguyên nhân: tiếng ồn xung quanh, phương pháp hay giải thuật xử lý nhiễu chưa hợp lý và làm mất quá nhiều thông tin,…



Hình II-9. Nhiễu trong Time series data



Hình II-10. Dữ liệu gốc và dữ liệu bị nhiễu

c) Tính chính xác, độ lệch, sự đúng đắn của dữ liệu:

Trong quá trình thống kê và thí nghiệm, chất lượng của quá trình đo lường và dữ liệu của kết quả được đo bằng tính chính xác (precision) và độ lệch (bias).

- Tính chính xác: Là gần kề của kết quả đo được lặp lại nhiều lần.
- Độ lệch: Là sự khác nhau về mặt hệ thống của những kết quả đo khi đo cùng một đại lượng.

Tính chính xác thường được đo bằng độ lệch tiêu chuẩn (standard deviation) của một tập các giá trị. Độ lệch được đo bằng sự khác nhau giữa giá trị trung bình của tập hợp các giá trị với giá trị đã biết của lượng được đo. Ví dụ: Độ đo của vật thể X sau 5 lần đo, có kết quả như sau: 1.015;0.99;1.013;1.001;0.986. Giá trị trung bình là $\overline{X}$ =1.001, độ lệch tiêu chuẩn bằng 0.013096.

- Sự đúng đắn: Sự gần đúng của các số liệu đo được với giá trị thực của lượng được đo. Sự đúng đắn (accuracy) thường được sử dụng để mô tả mức độ đo sai. Sự đúng đắn phụ thuộc vào tính chính xác và độ lệch của phép đo (kết quả đo).

d) Ngoại lệ:

Là những đối tượng dữ liệu có những đặc điểm khác xa so với hầu hết dữ liệu trong cùng 1 nhóm hay ngoại lệ là những trường hợp các đối tượng dữ liệu có giá trị của một số thuộc tính khác xa giá trị của cùng thuộc tính của các đối tượng còn lại trong nhóm. Ví dụ: Trong cùng 1 loài hoa Iris có độ dài đài hoa, độ rộng đài hoa, độ dài cuốn hoa, độ rộng cuốn hoa lần luợt có giá trị trung bình lần

lượt là: 5.1, 3.5, 1.4, 0.2; nhưng xuất hiện một bông hoa iris có các số đo tương ứng là: 7.0,3.2,4.7,1.4 và xuất hiện màu khác,…

Từ đấy có thể rút ra kết luận rằng, ngoại lệ ảnh hưởng rất lớn đến quá trình phân tích dữ liệu trong khai phá dữ liệu. Ngoại lệ cần được phát hiện và loại bỏ trong quá trình tiền xử lý dữ liệu.

e) Giá trị bị thiếu:

Giá trị bị thiếu do quá trình thu nhập dữ liệu không đảm bảo thu thập đầy đủ giá trị của tất cả các thuộc tính của các đối tượng dữ liệu. Lỗi đó có thể do 1 số nguyên nhân sau:

- Do người dùng không cung cấp dữ liệu không cung cấp đầy đủ thông tin.
- Do người thu thập dữ liệu làm mất thông tin.
- Trong 1 số trường hợp đặc biệt nào đó mà dữ liệu bị mất đi một phần.

Vì vậy, dữ liệu bị thiếu đóng một vai trò rất quan trọng trong quá trình phân tích. Việc xử lý dữ liệu có nhiều cách khác nhau, nhưng mỗi cách đều có ưu điểm, khuyết điểm và phù hợp với những tình huống cụ thể khác nhau. Do đó, cần cẩn thận trong việc lựa chọn cách xử lý nhằm tránh ảnh hưởng đến kết quả của các bước tiếp theo và độ chính xác cũng như hiệu quả của cả hệ thống. Sau đây là một vài cách xử lý cho các trường hợp cụ thể:

➢ Loại bỏ đối tượng dữ liệu hay thuộc tính: là cách xử lý đơn giản và hiệu quả trong trường hợp dữ liệu bị thiếu giá trị. Tương ứng với các trường hợp sẽ có biện pháp xử lý sau:

- Nếu đối tượng dữ liệu nào thiếu thuộc tính thì loại ra khỏi tập dữ liệu dùng để phân tích.
- Nếu có quá nhiều đối tượng cùng thiếu giá trị do một thuộc tính nào đó thì loại bỏ thuộc tính đó ra khỏi tập thuộc tính của đối tượng dữ liệu.

➢ Ước lượng giá trị bị thiếu: Trong một số trường hợp, các giá trị thiếu có thể được ước lượng từ các giá trị khác đã có trước đó.

➢ Không quan tâm đến giá trị bị thiếu trong suốt quá trình phân tích: Nhiều hướng tiếp cận của khai phá dữ liệu có thể bỏ qua các giá trị bị thiếu trong lúc phân tích.

f) Giá trị không nhất quán:

Ví dụ: 2 mẫu tin có giá trị "tỉnh thành" giống nhau nhưng giá trị "mã bưu cục" khác nhau.

Có nhiều nguyên nhân dẫn đến dữ liệu không nhất quán. Có thể do cơ sở dữ liệu thiết kế không tốt, do người cung cấp dữ liệu cung cấp sai thông tin, do người thu nhập nhập dữ liệu sai,…Dữ liệu không nhất quán có thể được phát hiện và chỉnh sửa kịp thời.

g) Dữ liệu bị trùng lắp:

Dữ liệu trong quá trình thu nhập có thể chứa nhiều đối tượng dữ liệu bị trùng lắp. Dữ liệu trùng lắp có thể do quá trình nhập liệu và thu thập dữ liệu không lặp lại nhiều lần. Dữ liệu bị trùng lắp cần được phát hiện và loại bỏ trong quá trình tiền xử lý dữ liệu.

II.2.2.2. Các vấn đề liên quan đến ứng dụng:

Chất lượng của dữ liệu có thể được quan sát và đánh giá ở nhiều gốc độ ứng dụng khác nhau. Tùy vào ứng dụng cụ thể mà chất lượng dữ liệu sẽ phù hợp với ứng dụng đó. Một số trường hợp đòi hỏi dữ liệu phải thật tốt (chất lượng cao) trong khi một số trường hợp chấp nhận dữ liệu có chứa một ít sai sót. Một số vấn đề cần quan tâm đến ứng dụng là:

a) Tính phù hợp theo thời gian:

Trong một số trường hợp, dữ liệu chỉ có giá trị sử dụng trong một khoảng thời gian nhất định kể từ khi dữ liệu được thu thập. Dữ liệu được thu thập quá lâu có thể sẽ không còn hữu dụng, không còn phản ánh đúng bản chất của sự vật.

Ví dụ: Điểm sàng đại học năm 2010 không thể áp dụng cho điểm sàng đại học năm 2010.

b) Tính liên quan:

Dữ liệu phải chứa thông tin hữu ích và cần thiết cho ứng dụng.

Ví dụ: Để xây dựng mô hình tư vấn việc chọn ngành nghề cho thí sinh thi tuyển sinh đại học. Thông tin về mức sống, sở thích, điều kiện và hoàn cảnh gia đình của thí sinh không thật sự cần thiết. Trong khi, thông tin về học lực, hạnh kiểm, sức khỏe,…lại rất quan trọng.

c) Tri thức về dữ liệu:

Một cách lý tưởng, các tập dữ liệu có được từ các tài liệu mô tả các khía cạnh khác nhau của dữ liệu. Chất lượng của tài liệu này sẽ giúp ích rất nhiều cho quá trình phân tích dữ liệu. Kiến thức về dữ liệu còn thể hiện ở việc nhận biết các đặc điểm quan trọng của dữ liệu như: tính chính xác của dữ liệu, các kiểu thuộc tính, tỉ lệ đo và nguồn gốc của dữ liệu.

II.3. Thu thập và tiền xử lý dữ liệu:

Để dữ liệu có thể ứng dụng vào quá trình khai phá dữ liệu, dữ liệu cần được thu thập và xử lý. Công việc của giai đoạn này là lựa chọn đối tượng dữ liệu và thuộc tính cho quá trình phân tích hoặc tạo ra các thuộc tính mới hoặc thay đổi thuộc tính. Mục đích cuối cùng của quá trình này là cải thiện quá trình phân tích trong khai phá dữ liệu ở khía cạnh thời gian, tiền của và chất lượng.

II.3.1. Tổng hợp dữ liệu:

Là việc gom 2 hay nhiều đối tượng dữ liệu lại với nhau, nhằm mục đích tạo thành một đối tượng.

Ví dụ: Ở một siêu thị có nhiều phòng ban, mỗi phòng ban có quyền truy cập đến hệ thống ở một lãnh vực riêng của hệ thống. Mỗi nhân viên ở siêu thị sẽ làm việc cho các văn phòng của siêu thị. Bằng việc thiết lập quyền truy cập cho các phòng của siêu thị sẽ tiết kiệm thời gian và không gian bộ nhớ hơn việc thiết lập quyền riêng cho các nhân viên.

Tổng hợp dữ liệu sẽ phải dựa trên các nguyên tắt sau:

- Đối với thuộc tính kiểu số: lấy tổng hoặc trung bình.
- Đối với các thuộc tính không phải kiểu số: có thể bỏ qua hoặc tổng hợp như là môt tập hợp các giá trị.

➢ Ưu điểm của việc kết hợp dữ liệu:

(1) Tập hợp dữ liệu sau khi kết hợp nhỏ hơn đáng kể so với tập dữ liệu ban đầu. Dung lượng bộ nhớ lưu trữ ít hơn, thời gian xử lý ngắn hơn, sử dụng các thuật toán vét cạn.

(2) Có thể coi việc kết hợp dữ liệu như là việc thay đổi giá trị và thang chia giá trị. Cung cấp góc nhìn dữ liệu ở mức cao.

(3) Dữ liệu sau khi kết hợp ổn định hơn dữ liệu đơn lẽ trước khi kết hợp.

➢ Hạn chế của việc kết hợp dữ liệu: Khả năng mất các thông tin hay chi tiết quan trọng.

II.3.2. Lấy mẫu:

Thường được sử dụng trong việc lựa chọn tập thuộc tính con dùng để phân tích và cũng là một cách làm rất hữu dụng trong khai phá dữ liệu. Mục đích chính của việc lấy mẫu là "làm giảm thời gian và tài nguyên cho quá trình phân tích dữ liệu".

➢ Nguyên tắt lấy mẫu hiệu quả:

- Lấy mẫu phải đại diện cho tập hợp dữ liệu.
- Mẫu dữ liệu phải có đầy đủ các thuộc tính như tập dữ liệu gốc.
- Phương pháp lấy mẫu phải đảm bảo tính đại diện của mẫu dữ liệu.
- Kỹ thuật lấy mẫu và số lượng mẫu phải phù hợp.

➢ Cách tiếp cận khi lấy mẫu:

(1) Lấy mẫu ngẫu nhiên (random samling): Đây là cách lấy mẫu đơn giản nhất. Đối với cách này, xác suất để chọn các phần tử trong tập hợp là như nhau. Có 2 cách biến thể của lấy mẫu ngẫu nhiên là:

- Lấy mẫu không có sự thay thế: Mỗi phần tử chỉ có thể được chọn một lần duy nhất. Khi một phần tử được chọn thì nó sẽ bị loại ra khỏi tập hợp và việc lựa chọn mẫu tiếp theo sẽ áp dụng trên các tập hợp các phần tử chưa được chọn.
- Lấy mẫu có sự lặp lại: Một phần tử có thể được chọn nhiều hơn một lần. Khi chọn một phần tử được chọn thì nó sẽ không bị loại ra khỏi tập hợp và nó sẽ có khả năng được chọn ở lần chọn tiếp theo.

(2) Để hạn chế các hiệu ứng phụ (điểm yếu) của phương pháp lấy mẫu, dữ liệu ban đầu nên được chia làm nhiều lớp. Việc chọn lấy mẫu sẽ áp dụng cho từng lớp dữ liệu nên mẫu lấy về sẽ đại diện cho cả tập hợp dữ liệu ban đầu.

Lấy mẫu theo lũy tiến (progressive sampling): Trong thực tế, rất khó xác định số lượng mẫu của từng tập dữ liệu. Lấy mẫu theo cách lũy tiến là cách lấy mẫu như sau:

- Bắt đầu với 1 lượng mẫu nhỏ.
- Tăng dần lượng mẫu cho đến khi nào đạt được kích thước phù hợp (đủ lớn).
- Dừng tăng khi nào độ chính xác của mô hình đạt đến mức ổn định.

➢ Mất thông tin trong lấy mẫu:

Vấn đề lựa chọn kích thước của tập hợp mẫu rất quan trọng vì nó ảnh hưởng đến độ chính xác của mô hình sau khi phân tích. Kích thước của mẫu càng lớn thì kết quả phân tích càng gần với kết quả phân tích của tập dữ liệu gốc nhưng ý nghĩa của việc lấy mẫu sẽ không còn nữa. Kích thước của mẫu càng nhỏ

thì sẽ dẫn đến mất thông tin và thu được kết quả phân tích khác xa so với kết quả phân tích của tập dữ liệu gốc.

a) Ảnh ban đầu

b) Ảnh mất thông tin do nhiễu muối tiêu

c) Ảnh mất thông tin do nhiễu Gause

Hình II-11. Mất thông tin khi lấy mẫu

II.3.3. Giảm bớt thuộc tính:

Giảm bớt thuộc tính chính là để chỉ các kỹ thuật làm giảm số chiều (thuộc tính) của dữ liệu bằng cách tạo ra thuộc tính mới là tập hợp của các thuộc tính cũ. Việc giảm bớt thuộc tính mang lại rất nhiều lợi ích cho quá trình phân tích dữ liệu.

II.3.3.1. Thuận lợi:

- Các thuật toán trong khai phá dữ liệu sẽ làm việc tốt hơn khi áp dụng trên tập dữ liệu có ích thuộc tính. Bởi vì, giảm bớt thuộc tính sẽ bỏ đi các thuộc tính kém quan trọng và có thể giảm được nhiễu trong dữ liệu.
- Làm cho quá trình biểu diễn (visualize) dữ liệu dễ hơn.
- Giảm thời gian và tài nguyên cho việc phân tích.

II.3.3.2. Khó khăn:

Thuật ngữ "the curse of dimensionality" dùng để chỉ hiện tượng mà nhiều kiểu phân tích dữ liệu trở nên khó khăn hơn khi số thuộc tính của dữ liệu tăng lên. Một cách đặt biệt, khi tăng số lượng thuộc tính thì dữ liệu càng trở nên thưa thớt trong không gian mà nó chiếm giữ. Tùy vào mức ảnh hưởng, nó sẽ tác động trực tiếp đến các thuật toán của khai phá dữ liệu.

- Đối với quá trình phân lớp dữ liệu (classification) là rất khó khăn, vì không đủ đối tượng dữ liệu cho việc tạo ra mô hình đáng tin cậy.
- Đối với việc gom nhóm dữ liệu (clustering), mật độ và khoảng cách giữa các đối tượng trở nên vô nghĩa.

Tóm lại, thuật toán phân lớp dữ liệu và gom nhóm dữ liệu gặp rắc rối khi dữ liệu có quá nhiều thuộc tính.

II.3.3.3. Các kỹ thuật đại số tuyến tính cho việc giảm thuộc tính:

Nhằm làm giảm bớt các thuộc tính bằng cách sử dụng kỹ thuật đại số tuyến tính để chiếu dữ liệu từ không gian nhiều chiều sang không gian có số chiều ít hơn. Các kỹ thuật thường được sử dụng là:

➢ Principal Compoment Analysis (PCA): Là kỹ thuật dùng cho các thuộc tính liên tục. Nguyên tắt của cách phân tích này là tìm thuộc tính mới có tính chất:

- Là tổ hợp tuyến tính của các thuộc tính gốc.
- Trực giao vuông góc với nhau.
- Giữ được lượng lớn nhất của sự thay đổi dữ liệu.

➢ Sigualr Value Descomposition (SVD): Là một kỹ thuật liên quan với PCA và thường được dùng để giảm số thuộc tính.

II.3.4. Lựa chọn tập thuộc tính con:

Một cách khác để giảm bớt số thuộc tính là sử dụng tập thuộc tính con. Cách làm này có thể loại bỏ được các thuộc tính dư thừa (không sử dụng) và các thuộc tính không có ý nghĩa hay không có liên quan (không sử dụng).

Ví dụ: Sử dụng thuộc tính đơn giá mua thì không cần sử dụng thuộc tính thuế giá trị gia tăng, thuộc tính mã số sinh viên không liên quan đến quá trình dự đoán khả năng học tập của sinh viên.

Từ đó, để lựa chọn tập thuộc tính con tốt nhất đòi hỏi phải có một cách tiếp cận một cách hệ thống.

II.3.4.1. Tiếp cận trong việc lựa chọn thuộc tính con:

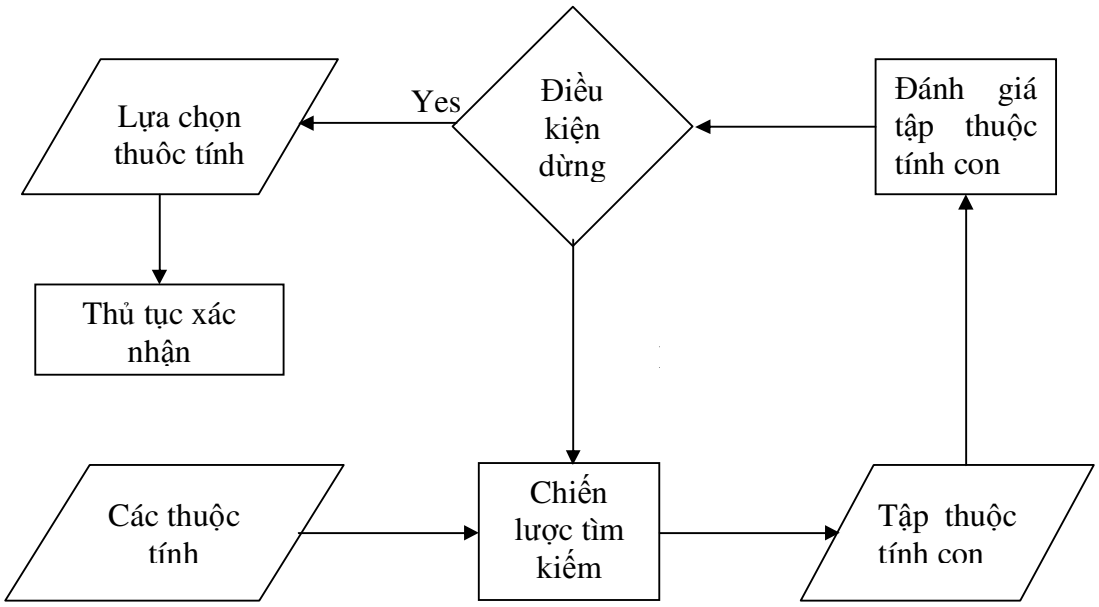
- Theo dạng nhúng (embedded aproaches): Việc lựa chọn thuộc tính xảy ra một cách tự nhiên như là một thành phần của thuật toán khai phá dữ liệu. Trong suốt quá trình xử lý, thuật toán khai phá dữ liệu sẽ quyết đin thuộc tính nào được dùng, thuộc tính nào sẽ bị bỏ qua.
- Tiếp cận theo dạng lọc (filter approaches): Thuộc tính sẽ được lựa chọn trước khi được dùng cho quá trình khai phá dữ liệu. Cách lựa chọn độc lập với các thuật toán khai phá dữ liệu.
- Tiếp cận theo dạng bao lọc (wrapper approaches): Sử dụng các thuật toán khai phá dữ liệu như một hộp đen để tìm tập thuộc tính con tốt nhất.

II.3.4.2. Qui trình lựa chọn thuộc tính con: gồm 4 phần:

- Một giá trị đo lường cho việc đánh giá một tập thuộc tính con. Việc đánh giá tập con hiện tại với các tập con khác, đòi hỏi phải có một độ đo dùng để đánh giá nhằm xác định mức độ tốt của các thuộc tính đối với một công việc cụ thể trong khai phá dữ liệu.
- Một chiến lượt tìm kiếm có khả năng điều khiển được việc sinh ra tập thuộc tính con. Về mặt ý tưởng, việc lựa chọn thuộc tính con là vét hết tất cả các tập hợp có thể có. Có thể sử dụng nhiều chiến lượt tìm kiếm khác nhau nhưng phải chú ý đến độ phức tạp của thuật toán và các ràng buộc khác.
- Điều kiện dừng là rất cần thiết vì số lượng các tập con là rất lớn và việc kiểm tra tất cả các tập con là không thực tế. Điều kiện dừng liên quan đến: số lần

lặp, so sánh kết quả đánh giá với giá trị "cầm canh" (threshold), số lượng các thuộc tính con,…

- Kiểm định và xác nhận kết quả khi các tập hợp con được chọn. Phương pháp đơn giản là áp dụng thuật toán khai phá dữ liệu trên toàn tập dữ liệu gốc và trên các tập thuộc tính con. Nếu kết quả chạy trên tập hợp con các thuộc tính mà tốt hơn hay ít nhất là gần bằng với chạy trên tất cả các thuộc tính thì sẽ dừng việc tìm thuộc tính con. Một cách khác dùng để xác định kết quả là sử dụng nhiều giải thuật lựa chọn thuộc tính khác nhau để sinh ra các tập thuộc tính con khác nhau. Sau đó so sánh kết quả của từng giải thuật lựa chọn.



Hình II-12. Kiến trúc của việc chọn tập thuộc tính

II.3.4.3. Gán trọng lượng cho thuộc tính:

Là một cách làm để loại bỏ các thuộc tính kém quan trọng và giữ lại các thuộc tính quan trọng hơn. Thuộc tính càng quan trọng thì gán trọng số càng lớn.

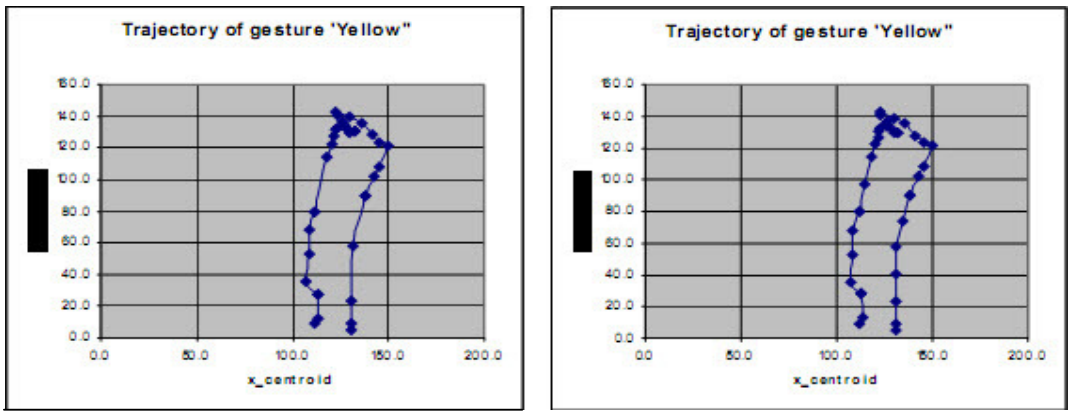
II.3.5. Tạo ra thuộc tính mới:

Thuộc tính mới thường được tạo dựa trên thuộc tính có sẵn. Một tập hợp các thuộc tính mới có thể chứa nhiều thông tin quan trọng hơn tập thuộc tính gốc. Có 3 phương pháp dùng để tạo ra thuộc tính mới là:

II.3.5.1. Trích lọc thuộc tính:

Là việc tạo ra tập thuộc tính mới dựa trên một tập thuộc tính ban đầu.

Ví dụ: Cho trước tập hợp các cử chỉ trong ngôn ngữ cử chỉ (sign language).Trích lọc các thuộc tính dùng để phân loại và nhận dạng cử chỉ. Thuộc tính này có thể là: đường di chuyển của tay, độ dài từ tâm của kí hiệu đến các điểm biên, gốc dịch chuyển của các frame hình.



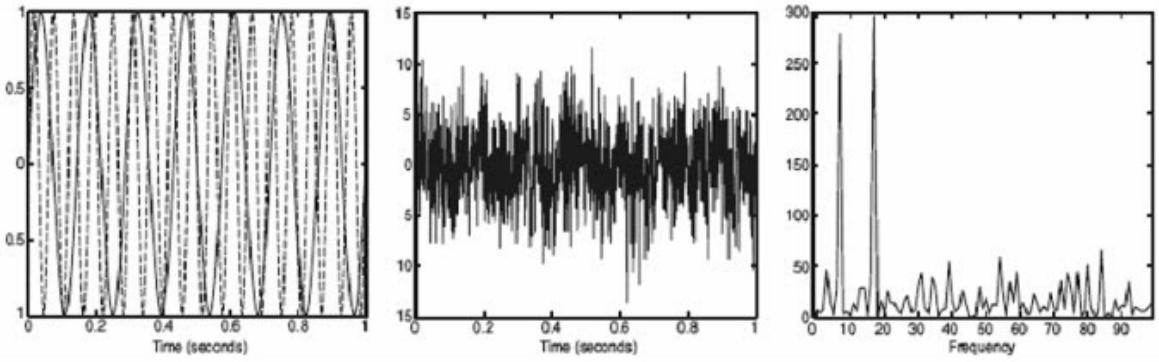
a) Đường đi "gốc" b) Đường đi đã được "làm mịn"

Hình II-13. Ví dụ về đường di chuyển của một ký hiệu trong ngôn ngữ khiếm thính của người Việt Nam

II.3.5.2. Chuyển đổi không gian:

Ở gốc độ khác nhau, việc hiểu dữ liệu có thể phát hiện ra nhiều thông tin quan trọng từ dữ liệu cũng như các thuộc tính quan trọng trong quá trình phân tích dữ liệu.

Ví dụ: Dữ liệu time series có thể chứa các chu trình. Nếu dữ liệu không bị nhiễu thì việc tìm ra các chu trình rất dễ dàng, ngược lại rất khó khăn. Đối với dữ liệu theo thời gian, thì biến đổi Fourier, Wavelet là một cách làm hiệu quả trong việc chuyển không gian dữ liệu.



a) Two time series b) Noisy time series c) Power spectrum

Hình II-14. Áp dụng biến đổi Fouries để xác định tần số quan trọng của time series data

II.3.5.3. Xây dựng thuộc tính:

Trong một số trường hợp, tập thuộc tính hiện tại của dữ liệu chứa nhiều thông tin quan trọng không thể áp dụng cho các kỹ thuật khai phá dữ liệu. Trong tình huống này, cần phải xây dựng tập thuộc tính mới dựa trên tập thuộc tính có sẵn để phù hợp với các kỹ thuật khai phá dữ liệu mà mình muốn áp dụng.

II.3.6. Rời rạc hóa và nhị phân hóa:

II.3.6.1. Nhị phân hóa:

Một số kỹ thuật đơn giản để nhị phân hóa các thuộc tính phân loại là:

- Nếu thuộc tính phân loại có tối đa m giá trị thì gán mỗi giá trị bởi một số nguyên từ 0 đến m-1.
- Nếu thuộc tính phân loại có giá trị là kiểu số có thứ tự thì sắp xếp thứ tự các giá trị.
- Chuyển các giá trị số nguyên sang số nhị phân. Số chữ số dùng để biểu diễn m giá trị là $n = \lceil \log_2^m \rceil$.

Ví dụ: Xét một biến kiểu phân loại có 5 giá trị {kém, yếu, trung bình, khá, giỏi}. Các giá trị này chuyển sang số nhị phân 3 bit như sau:

Bảng II-1

Giá trị phân loại	Giá trị số nguyên	x_1	x_2	x_3
Kém	0	0	0	0
Yếu	1	0	0	1
Trung bình	2	0	1	0
Khá	3	0	1	1
Giỏi	4	1	0	0

Tuy nhiên, việc chuyển đổi như vậy không thể hiện được mối quan hệ giữa các giá trị của thuộc tính được chuyển đổi. Ví dụ: giỏi hơn khá, khá hơn trung bình, trung bình hơn yếu, yếu hơn kém.

Một cách khác để nhị phân hóa là đổi số nguyên sang số nhị phân không đối xứng. Trong ví dụ trên, có 5 giá trị phân loại, cần 5 bits để biểu diễn nhị phân không đối xứng như sau:

Bảng II-2

Giá trị phân loại	Giá trị số nguyên	x_1	x_2	x_3	x_4	x_5
Kém	0	1	0	0	0	0
Yếu	1	0	1	0	0	0
Trung bình	2	0	0	1	0	0
Khá	3	0	0	0	1	0
Giỏi	4	0	0	0	0	1

Trong một số trường hợp, có thể áp dụng biện pháp: nếu một thuộc tính có 2 giá trị thì chỉ cần sử dụng 1 bits. Ví dụ: x_1=0 là nữ, x_2=1 là nam.

II.3.6.2. Rời rạc hóa thuộc tính liên tục:

Thường được sử dụng khi áp dụng kỹ thuật phân tích phân loại dữ liệu (classification) và kết hợp (assiociation). Một cách tổng quát, cách rời rạc hóa tốt nhất phụ thuộc vào thuật toán khai phá dữ liệu sẽ áp dụng để phân tích cũng như các thuộc tính sẽ được rời rạc hóa.

a) Rời rạc hóa các giá trị liên tục:

Đổi một thuộc tính từ liên tục sang rời rạc liên quan đến 2 vấn đề:

i. Số lượng giá trị của thuộc tính rời rạc.

ii. Cách chuyển từ giá trị liên tục sang giá trị rời rạc.

Việc đầu tiên khi thực hiện rời rạc hóa các giá trị liên tục là sắp xếp các giá trị của thuộc tính liên tục, chia các giá trị này ra làm n $\{(x_0,x_1];(x_1,x_2];\ldots(x_{n-1},x_n)\}$ đoạn bằng các sử dụng n-1 điểm chia. Công việc thực hiện kế tiếp là ánh xạ mỗi đoạn vào một giá trị rời rạc. Cách thực hiện trong rời rạc hóa có thể là : giám sát và không giám sát. Tùy vào điều kiện thực tế của kỹ thuật khai phá dữ liệu thì sẽ áp dụng.

b) Trường hợp thuộc tính phân loại có nhiều giá trị:

Cần phải kết hợp nhiều phương pháp rời rạc hóa phù hợp với kỹ thuật khai phá dữ liệu sẽ được sử dụng.

II.3.7. Chuyển đổi thuộc tính:

Chuyển đổi thuộc tính là việc chuyển đổi được áp dụng cho tất cả các giá trị của một thuộc tính. Có 2 kiểu chuyển đổi quan trọng là:

II.3.7.1. Sử dụng hàm đơn giản:

Trong trường hợp này, một số hàm tính toán đơn giản được sử dụng để chuyển đổi giá trị của thuộc tính. Các hàm này có thể sử dụng để chuyển đổi giá trị x của thuộc tính là: x^k, log x, e^x, 1/x, |x|, sin x, $\sqrt{x}$.

Lưu ý: Khi biến đổi dữ liệu cần lưu ý đến các khả năng có thể thay đổi bản chất của dữ liệu. Ví dụ: Hàm f(x)=1/x có thể giảm độ lớn của f(x) với x>1 nhưng lại làm tăng giá trị của f(x) đối với x<1.

II.3.7.2. Chuẩn hóa:

Mục đích là làm cho cả tập dữ liệu có một thuộc tính nào đó. Có nhiều cách để chuẩn hóa dữ liệu được áp dụng tùy vào trường hợp cụ thể.

II.4. Một số kỹ thuật khai phá dữ liệu:

II.4.1. Phân cụm dữ liệu (Cluster analysis):

II.4.1.1. Giới thiệu:

Phân tích cụm là 1 kỹ thuật thường được sử dụng trong lĩnh vực khám phá tri thức. Kỹ thuật này, thường được sử dụng trong việc gom nhóm các dữ liệu tương tự nhau hoặc các mô hình có mật độ xác định lại với nhau nhằm tạo nên 1 dữ liệu mới dựa trên nhóm dữ liệu đã cho và có thể được rút gọn hơn so với dữ liệu ban đầu. Phân tích cụm gắn liền với việc học không giám sát, khi đó dữ liệu và nhãn là không có sẵn.

Ví dụ: Khi giới thiệu 1 sản phẩm trong siêu thị, người quản lý hay nhà kinh doanh sẽ xác định 1 nhóm hoặc cụm khách hàng đã tồn tại trong lịch sử thanh toán của hệ thống, đối với việc gom nhóm khách hàng có thể là theo tuổi, thu nhập hoặc mức sống đề đưa ra được chiến lược kinh doanh và hướng tới khách hàng.

Phân tích cụm dữ liệu thường được sử dụng cho phương pháp khai thác dữ liệu mô tả. Cho một ma trận gồm n dòng dữ liệu và p cột, mục tiêu của phân tích

cụm là gom các dữ liệu và các nhóm thành nội bộ đồng nhất (nội bộ gắn kết) và không đồng nhất từ nhóm này sang nhóm khác (tách bên ngoài).

Bên cạnh đó, Phân tích cụm cũng là 1 kỹ thuật quan trọng được ứng dụng trong khai khoáng dữ liệu đa phương tiện. Mục đích là để phân tích cụm nội dung đa phương tiện với nhau để lập ra các chỉ mục hiệu quả, và được lưu trữ vào trong cơ sở dữ liệu (database) đa phương tiện.

Ví dụ: Các bức ảnh tương tự nhau có thể được Phân tích cụm với nhau để lập thành 1 chỉ mục hiệu quả; khi đó, khi thực hiện truy vấn thì kết quả trả về sử dụng 1 hình ảnh truy vấn hoặc hình ảnh mô tả, sau đó là các hình ảnh tương tự được thu hồi.

Mục tiêu chính của phương pháp phân cụm dữ liệu là nhóm các đối tượng tương tự nhau trong tập dữ liệu vào các cụm sao cho các đối tượng thuộc cùng một lớp là tương đồng còn các đối tượng thuộc các cụm khác nhau sẽ không tương đồng.

II.4.1.2. Các phương pháp phân cụm :

a) Phương pháp phân cấp: (Hierachical methods)

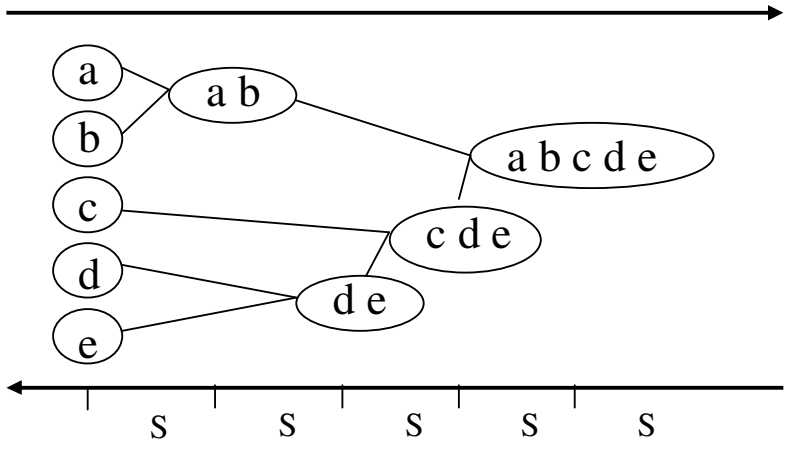
Phân cấp cụm thường được biểu diễn dưới dạng cây của các cụm. Trong đó:

- Các lá của cây biểu diễn từng đối tượng.
- Các nút trong biểu diễn các cụm.

Có 2 phương pháp tạo cây phân cấp:

➢ Phương pháp phân cấp từ trên xuống:

Bắt đầu từ cụm lớn nhất chứa tất cả các đối tượng. Chia cụm phân biệt nhất thành các cụm nhỏ hơn và tiếp diễn cho đến khi có n cụm thoả mãn điều kiện dừng.



Hình II-12. Biểu diễn của phương pháp phân cụm từ trên xuống

➢ Phương pháp phân cấp từ dưới lên:

Các bước thực hiện:

- Bước 1: Tạo n nhóm, mỗi nhóm gồm một đối tượng và lập ma trận khoảng cách cấp n.

- Bước 2:Tìm 2 nhóm u,v có khoảng cách nhỏ nhất (d_{uv})
- Bước 3: Gộp nhóm u với nhóm v. Ký hiệu nhóm mới là (uv). Lập ma trận khoảng cách mới bằng cách:

+ Loại các hàng và cột tương ứng với các nhóm u,v

+ Thêm một hàng và một cột để lưu khoảng cách của nhóm uv với các nhóm còn lại

- Bước 4: Lặp lại các bước 2 và bước 3 cho đến khi chọn được k nhóm thích hợp nhất cho bài toán hoặc chỉ có một nhóm duy nhất.

b) Phương pháp phân cụm bộ phận: (Partitional clustering methods)

➢ Mô tả các phương pháp:

Cho một cơ sở dữ liệu D chứa n đối tượng, tạo phân hoạch thành tập có k cụm sao cho:

- Mỗi cụm chứa ít nhất một đối tượng
- Mỗi đối tượng thuộc về một cụm duy nhất
- Cho trị k, tìm phân hoạch có k cụm sao cho tối ưu hoá tiêu chuẩn phân hoạch được chọn.

➢ Các phương pháp

(1) Phương pháp gom cụm k-means:

Input: Số các cụm k cần gom và cơ sở dữ liệu chứa n đối tượng.

Output: k cụm đã được gom.

Thuật giải: gồm 4 bước

- Bước1: Phân hoạch đối tượng thành k tập con (cụm) ngẫu nhiên.
- Bước 2: Tính các tâm (trung bình của các đối tượng trong cụm) cho từng cụm trong phân hoạch hiện hành.
- Bước 3: Gán mỗi đối tượng cho cụm tâm gần nhất
- Bước 4: Nếu cụm không có sự thay đổi thì dừng, ngược lại quay lại bước 2

(2) Phương pháp gom cụm k-medoid:

Input: Số các cụm k cần gom và cơ sở dữ liệu chứa n đối tượng.

Output: k cụm đã được gom.

Thuật toán:

- Bước 1: Chọn k đối tượng ngẫu nhiên làm tâm của nhóm.
- Bước 2: Gán từng đối tượng còn lại vào cụm có tâm gần nhất.
- Bước 3: Chọn ngẫu nhiên 1 đối tượng không là đối tượng tâm, và thay một trong các tâm đó bằng nó nếu nó làm thay đổi đối tượng trong cụm (gán đối tượng cho cụm có tâm gần nhất).
- Bước 4: Nếu gán tâm mới thì quay lại bước 2, ngược lại thì dừng.

(3) Dựa trên mô hình cụm : (Model-based clustering)

Các phương pháp này nhằm mục đích để phù hợp giữa dữ liệu nhất định và một số mô hình toán học tối ưu hóa. Ở đây, dữ liệu thường giả định được tạo ra từ phân phối xác suất c, thường là phân phối Gaussian hoặc Normal, xung quanh

cụm trung tâm. Điều này có thể gọi là 1 phần của thuật toán Phân tích cụm c-means.

Tối ưu hóa kỳ vọng (Exceptation Maximization – EM) là 1 thuật toán phổ biến lặp lại (interation) thuộc về danh mục của phân nhóm, thường là dựa trên mô hình. Nó khác với thuật toán c-means ở chỗ: tại mỗi điểm trên mô hình thuộc về một nhóm theo 1 trọng số (Xác suất của các thành viên). Nói cách khác, không có ranh giới nghiêm ngặc giữa các cụm. Điều đó đồng nghĩa với việc các thông số được tính toán dựa trên biện pháp là tìm trọng số. Nó cung cấp 1 mô hình thống kê của các dữ liệu và có khả năng xử lý sự không chắc chắn liên quan. Thuật toán này có thể được đặc trưng như sau:

- Khởi tạo c cụm trung tâm.
- Quá trình thực hiện gồm 2 bước và có thể chuyển đổi qua lại với nhau:
  - *Bước kỳ vọng:* (Exceptation step) Chỉ định cho dữ liệu tại điểm X_i đến cụm U_k với xác suất là:

$$P(X_i \in U_k) = p(U_k \mid X_i) = \frac{p(U_k)p(X_i \mid U_k)}{p(X_i)} \qquad \text{(CT-II-1)}$$

Với $p(X_i \mid U_k) = N(m_k, E_k(X_i))$ theo phân phối Normal theo khoảng cách m_k với kỳ vọng E_k.

  - *Bước khai thác tối đa:* Ước tính các thông số của mô hình:

$$m_k = \frac{1}{N}\sum_{i=1}^{N}\frac{X_i P(X_i \in U_k)}{\sum_j P(X_i \in U_j)} \qquad \text{(CT-II-2)}$$

Trong thực tế, bài toán sẽ hội tụ nhanh hơn, nhưng không thể đạt tối ưu. Hội tụ được đảm bảo đối với các hình thức nhất định của chức năng tối ưu hóa. Sự phức tạp tính toán là O(c*N*n*t), với n là các tính năng đầu vào.

II.4.2. Hồi quy (Regression):

II.4.2.1. Giới thiệu:

Thuật ngữ hồi quy được sử dụng đầu tiên năm 1908, bởi Pearson. Mục đích của hồi quy là:

- Vấn đề giao dịch với các ước tính của một giá trị sản xuất dựa trên giá trị đầu vào.
- Hồi quy là một kỹ thuật khai thác dữ liệu được sử dụng để phù hợp với một phương trình của tập dữ liệu.

Ngoài ra, mục đích của hồi quy là tìm hiểu thêm về mối quan hệ giữ các biến độc lập (independent) hoặc biến dự đoán (predictor) và một biến phụ thuộc (dependent) hay tiêu chuẩn (criterion). Mô hình hồi quy dựa trên việc xây dựng các đồ thị dựa trên đường thẳng để giải quyết các bài toán có mức độ khó khác nhau. Chính vì vậy, hồi quy còn được biết đến là tất cả những thuật toán liên quan đến dữ liệu số. Hình thức đơn giản nhất của hồi quy là hồi quy tuyến tính, trong đó sử dụng phương trình đại số:

$$y_i = a + b_1 x_{i1} + b_2 x_{i2} + ... + b_k x_{ik} + e_i \text{ với i=1,2,...,n (CT-II-3)}$$

Hoặc tương đương:

$$Y=XB+E \quad \text{(CT-II-4)}$$

Trong đó:

- n là tất cả các quan sát xem xét.
- Y là véc tơ cột với n dòng chứa giá trị của các biến phản ứng.
- X là ma trận với n dòng và k+1 cột, cho mỗi cột chứa giá trị của biến giải thích cho n quan sát.
- B là véc tơ với k+1 dòng có chứa tất cả các trọng số của mô hình được ước tính trên cơ sở dữ liệu: các ngăn chặn và hệ số dốc tương ứng k so với mỗi biến giải thích.
- E là véc tơ cột của n chiều dài có chứa các từ ngữ lỗi (the error terms).

Có nhiều loại hồi quy khác nhau được sử dụng trong lĩnh vực thống kê và thường được sử dụng trong lĩnh vực dự đoán, nhưng ý tưởng cơ bản của hồi quy là mô hình được tạo ra mà bản đồ giá trị từ dự đoán có giá trị xảy ra lỗi là thấp nhất trong việc đưa ra một dự đoán.

Ví dụ: một nhà nông học có thể quan tâm tới việc nghiên cứu sự phụ thuộc của sản lượng lúa vào nhiệt độ, lượng mưa, nắng, phân bón,...

II.4.2.2. Các loại hồi quy

Có 2 loại:

1) Hồi quy tuyến tính:

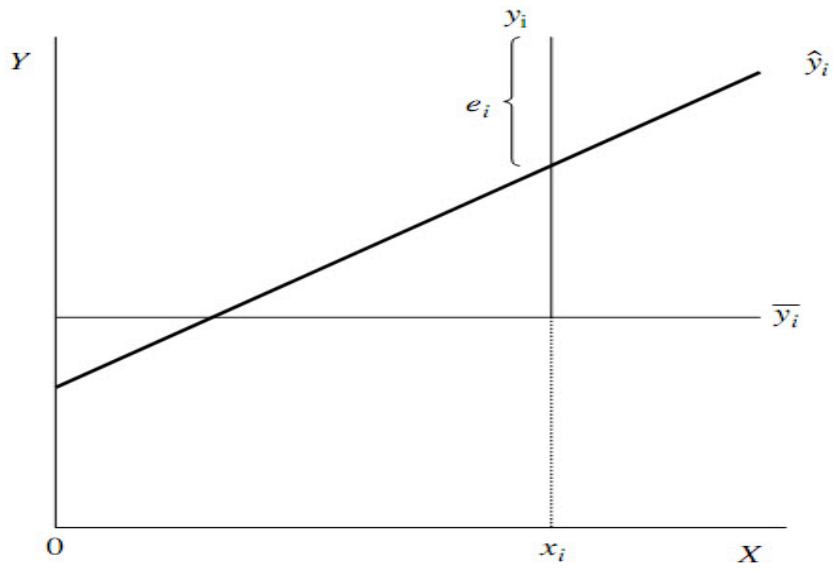
a) *Hồi quy tuyến tính hai chiều:*

Hồi quy tuyến tính 2 chiều là một phần cơ bản trong hồi quy tuyến tính. Nó đi sâu vào việc đánh giá 1 biến phụ thuộc hay phản ứng, được gây ra và giải thích bởi 1 biến khác, đó là biến độc lập hay là biến giải thích. Quá trình xây dựng và xác định biến giải thích có thể được xem như quá trình dự đoán.

Trong quá trình nghiên cứu, chúng ta sẽ sử dụng biến Y để chỉ biến phụ thuộc (phản ứng) và X cho biến độc lập (giải thích). Trong một số mô hình thống kê đơn giản có thể mô tả Y như là một hàm của X là hồi quy tuyến tính. Các mô hình hồi quy tuyến tính xác định mối quan hệ tuyến tính là mối quan hệ nhiễu giữa biến Y và X, và đối với các cặp (x_i,y_i) được quan sát và được gọi là hàm hồi quy:

$$y_i = a + bx_i + e_i \ \text{(i=1,2,…,n)} \quad \text{(CT-II-5)}$$

Trong đó:
- a là giá trị chặn (intercep) của hàm hồi quy.
- b là hệ số hồi quy (hay độ dốc của hàm hồi quy).
- e_i là lỗi ngẫu nhiên tương ứng với vị trí thứ i của hàm hồi quy.



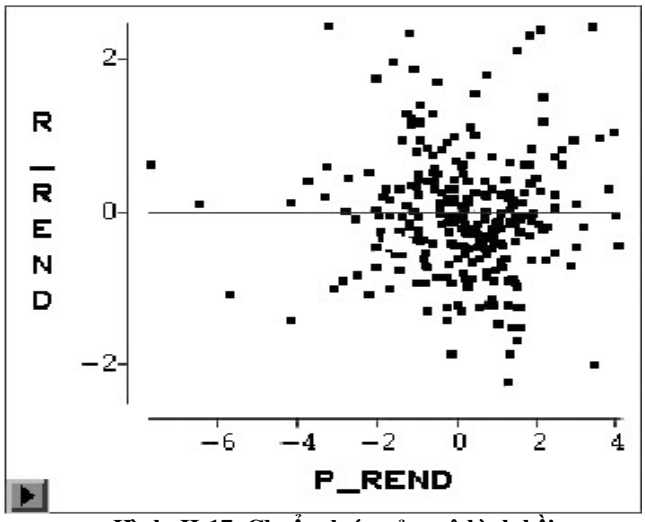
Hình II-16. Biểu diễn đường hồi quy

b) *Hồi quy tuyến tính đa chiều:*

Việc giải quyết mô hình hồi quy tuyến tính dựa trên mối quan hệ 2 chiều còn gặp nhiều khó khăn, do chỉ sử dung 1 biến độc lập (giải thích) . Chính vì thế, mô hình hồi quy tuyến tính nhiều chiều được ứng dụng để giải quyết vấn đề đó.

Giả sử tất cả các biến có trong ma trận dữ liệu, trừ các biến được gọi là biến phản ứng. Cho k là số biến giải thích. Hồi quy tuyến tính nhiều chiều được xác định bởi mối quan hệ sau:

$$y_i = a + b_1 x_{i1} + b_2 x_{i2} + ... + b_k x_{ik} + e_i \text{ với i=1,2,...,n (CT-II-6)}$$



Hình II-17. Chuẩn đoán của mô hình hồi quy

Hoặc tương đương:

$$Y=XB+E \qquad \text{(CT-II-7)}$$

Trong đó: - n là tất cả các quan sát xem xét.

- Y là véc tơ cột với n dòng chứa giá trị của các biến phản ứng.

- X là ma trận với n dòng và k+1 cột, cho mỗi cột chứa giá trị của biến giải thích cho n quan sát.

- B là véc tơ với k+1 hàng có chứa tất cả các trọng số của mô hình được ước tính trên cơ sở dữ liệu: các ngăn chặn và hệ số dốc tương ứng k so với mỗi biến giải thích.

- E là véc tơ cột của n chiều dài có chứa các từ ngữ lỗi (the error terms).

Trong trường hợp mô hình hồi quy 2 chiều được đại diện bằng 1 dòng, bây giờ (CT-II-6) tương ứng với k+1 – chiều mặt phẳng, được gọi là mặt phẳng hồi quy. Mặt phẳng này được định nghĩa là 1 phương trình:

$$y_i = a + b_1 x_{i1} + b_2 x_{i2} + ... + b_k x_{ik} \quad \text{(CT-II-8)}$$

Để xác định mặt phẳng được trang bị cần thiết để ước tính véc tơ của các tham số (a, b_1, b_2,…,b_k) trên cơ sở dữ liệu có sẵn.

2) Hồi quy lo gic:

Hồi quy tuyến tính được coi là 1 mô hình dự báo cho 1 biến đáp ứng về số lượng, còn hồi quy lo gic được xem xét một mô hình dự báo cho một biến phản ứng định tính. Một vấn đề đáp ứng chất lượng thường có thể được chia thành bài toán nhị phân. Các khóa xây dựng (bulding lock) của hầu hết các mô hình phản ứng định tính là mô hình hồi quy logic, đây là một trong những dự đoán quan trọng nhất của phương pháp khai thác.

Một mô hình được hiểu là hồi quy logic cần có các giá trị trang bị được hiểu là các xác suất mà sự kiện xảy ra trong các quần thể khác nhau.

$$\pi_i = P(Y_i = 1) \text{ với i=1,2,…,n} \quad \text{(CT-II-9)}$$

Chính xác hơn, 1 mô hình hồi quy tuyến tính cần xác định một chức năng thích hợp của các xác suất lắp đặt của sự kiện là 1 hàm tuyến tính của giá trị quan sát của các biến giải thích có sẵn. Ở đây là một ví dụ:

$$\log\left[\frac{\pi_i}{1-\pi_t}\right] = a + b_1 x_{i1} + b_2 x_{i2} + ... + b_k x_{ik} \quad \text{(CT-II-10)}$$

Vế trái xác định chức năng lo gic của xác suất được trang bị, tức là:

$$\log(\pi_i) = \log\left[\frac{\pi_i}{1-\pi_i}\right] \quad \text{(CT-II-11)}$$

Một khi π_i được tính toán, trên cở sở của dữ liệu, 1 giá trị được gán cho mỗi giá trị nhị phân $\hat{y}_i$ có thể thu được, đưa vào 1 giá trị ngưỡng của π_i với giá trị cận trên là $\hat{y}_i$=1 và cận dưới là $\hat{y}_i$=0. Không giống như hồi quy tuyến tính, các giá trị phản ứng được quan sát không thể bị phân hủy cộng tính là tổng giá trị trang bị và 1 giới hạn lỗi.

Việc lựa chọn chức năng logit để mô tả các chức năng liên kết π_i đến sự kết hợp tuyến tính của các biến giải thích, được thúc đẩy bởi một thực tế mà với

sự lựa chọn này có xu hướng về 0 và dần dần về 1. Và các giới hạn này cũng không đảm bảo rằng π_i là một xác suất hợp lệ. Một biến phản ứng nhị phân không thích hợp để sử dụng mô hình hồi quy tuyến tính để giải quyết, bởi vì 1 hàm tuyến tính là không giới hạn. Do đó, mô hình có thể dự đoán giá trị của biến phản ứng bên ngoài khoảng [0,1], điều đó là vô nghĩa. Nhưng dựa trên các kiểu liên kết để tìm ra kết quả là có thể.

II.4.2.3. Nhận xét:

a) Nhận xét chung:

Phân tích hồi quy thường được sử dụng để giải quyết các vấn đề sau:

- Ước lượng giá trị trung bình của biến phụ thuộc với giá trị đã cho của biến độc lập.
- Kiểm định giả thiết về bản chất của sự phụ thuộc.
- Dự đoán giá trị trung bình của biến phụ thuộc khi biết giá trị của các biến độc lập.
- Kết hợp các vấn đề trên.

b) Ưu điểm:

- Trong trường hợp hồi quy tuyến tính, nó xây dựng một mô hình trong đó có mối quan hệ giữa các biến độc lập và phụ thuộc được lên đến nhiệm vụ của nó và cho kết quả tối ưu. Còn đối với hồi quy logic, xây dựng một mô hình dựa trên xác suất mà sự kiện xảy ra trong quần thể.
- Cả hồi quy tuyến tính và hồi quy logic đều dựa trên dữ liệu có sẵn để xây dựng.
- Là một công cụ mạnh trong việc khai thác dữ liệu phân lớp.
- Hồi quy được giới hạn trong việc dự đoán các giá trị số.

c) Khuyết điểm:

- Hồi quy không được ứng dụng trong việc giải quyết các vấn đề khai thác dữ liệu với mục đích phân tích kết hợp.
- Trong việc xử lý với số lượng dữ liệu lớn, việc lựa chọn hồi quy cho việc khai thác dữ liệu sẽ gặp rất nhiều lỗi và nhiễu trong quá trình khai thác.

II.4.3. Cây quyết định (Decision tree):

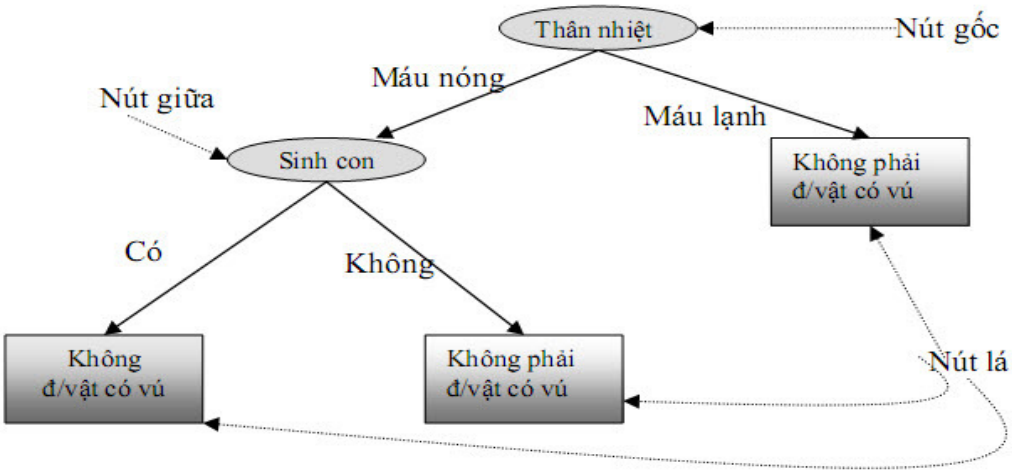
II.4.3.1. Giới thiệu:

Cây quyết định được sử dụng trong lĩnh vực khai phá dữ liệu và học máy. Cây quyết định thường được sử dụng như là một mô hình dự báo về một đối tượng mục tiêu, để có được kết luận về giá trị của mục tiêu đó. Cây quyết định còn được gọi là cây phân loại hay cây hồi quy.

Cấu trúc của một cây quyết định: trên cây quyết định có 3 loại nút

- Nút gốc: Không có cạnh vào, không có hoặc có nhiều cạnh ra.
- Nút giữa: Có chính xác một cạnh vào, có hai hay nhiều cạnh ra.
- Nút lá: có chính xác một cạnh vào, không có cạnh ra. Nút lá còn là đại diện cho phân loại, ngành đại diện hoặc liên từ của tính năng, từ đó dẫn đến những phân loại.

Trong phân tích quyết định, một cây quyết định có thể được sử dụng để đại diện rõ ràng và trực quan quyết định và ra quyết định. Trong khai phá dữ liệu, cây quyết định mô tả một dữ liệu nhưng không quyết định, các kết quả của cây phân loại dữ liệu có thể là đầu vào cho việc hỗ trợ ra quyết định.



Hình II-18. Ví dụ về cây quyết định

II.4.3.2. Giới hạn của cây quyết định:

- Vấn đề học trong cây quyết định tối ưu được biết đến là NP-complete theo các khía cạnh tối ưu và ngay cả đối với các khái niệm đơn giản. Do đó, thuật toán học của cây quyết định thực tế là dựa trên thuật toán Heuristic (*Phụ lục II*) cơ bản, như các thuật toán ham ăn (Greedy) nơi mà quyết định tối ưu được thực hiện tại địa phương của mỗi nút. Thuật toán này không thể đảm bảo cây quyết định vừa tìm được là tối ưu.
- Việc học của cây quyết định có thể tạo ra cây phức tạp, nếu dữ liệu đầu vào không khái quát các dữ liệu tốt. Điều này còn được gọi là Over-fitting, cơ chế như vậy có thể được sử dụng để cắt tỉa cây, tránh gặp phải vấn đề này.
- Có những khái niệm rất khó để học, vì thế cây quyết định không thể biểu diễn chúng một cách dễ dàng, như XOR, tương đương hoặc các vấn đề đa xử lý. Trường hợp này, cây quyết định trở thành một ngăn cản lớn.

II.4.3.3. Phương pháp xây dựng cây quyết định:

- Việc tạo cây quyết định bao gồm 2 giai đoạn : Tạo cây và tỉa cây .
  - Để tạo cây ở thời điểm bắt đầu tất cả những ví dụ huấn luyện là ở gốc sau đó phân chia ví dụ huấn luyện theo cách đệ qui dựa trên thuộc tính được chọn .
  - Việc tỉa cây là xác định và xóa những nhánh mà có phần tử hỗn loạn hoặc những phần tử nằm ngoài (những phần tử không thể phân vào một lớp nào đó) .
- Có rất nhiều biến đổi khác nhau về thuật toán xây dựng cây quyết định, mặc dù vậy chúng vẫn tuân theo những bước cơ bản sau :
  - Cây được thiết lập từ trên xuống dưới và theo cách thức chia để trị.
  - Ở thời điểm bắt đầu, các mẫu huấn luyện nằm ở gốc của cây

- Thuộc tính được phân loại (Rời rạc hóa các thuộc tính dạng phi số)
- Chọn một thuộc tính để phân chia thành các nhánh. Thuộc tính được chọn dựa trên độ đo thống kê hoặc độ đo heuristic.
- Tiếp tục lặp lại việc xây dựng cây quyết định cho các nhánh.

• Điều kiện để dừng việc phân chia:

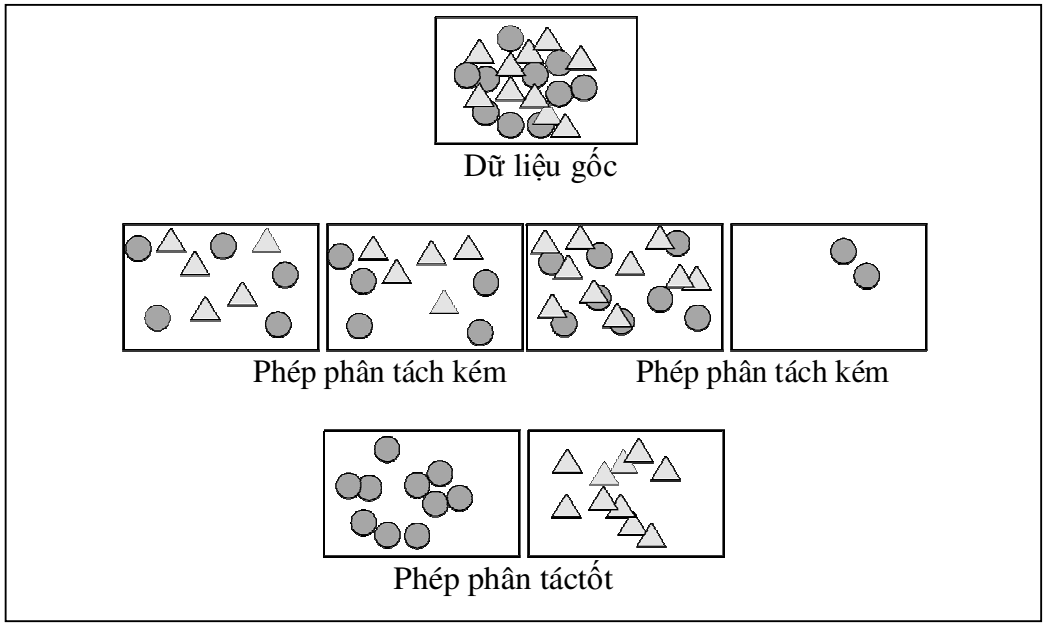
- Tất cả các mẫu rơi vào một nút thuộc về cùng một lớp (nút lá)
- Không còn thuộc tính nào có thể dùng để phân chia mẫu nữa
- Không còn lại mẫu nào tại nút.

II.4.3.4. Xây dựng cây quyết định:

1) Chọn thuộc tính phân tách:

Lúc khởi đầu, ta có trong tay một tập luyện chứa tập các bản ghi được phân loại trước – tức là giá trị của biến đích được xác định trong tất cả các trường hợp. Cây quyết định được xây dựng bằng cách phân tách các bản ghi tại mỗi nút dựa trên một thuộc tính đầu vào. Rõ ràng nhiệm vụ đầu tiên là phải chọn ra xem thuộc tính nào đưa ra được sự phân tách tốt nhất tại nút đó.

Độ đo được sử dụng để đánh giá khả năng phân tách là độ tinh khiết. Chúng ta sẽ có những phương pháp xác định để tính toán độ tinh khiết một cách chi tiết, tuy nhiên chúng đều cố gắng đạt được hiệu quả như nhau. Một sự phân tách tốt nhất là sự phân tách làm tăng độ tinh khiết của tập bản ghi với số lượng lớn nhất. Một sự phân tách tốt cũng phải tạo ra các nút có kích cỡ tương tự nhau, hay chí ít cũng không tạo ra các nút có quá ít bản ghi.



Hình II-19. Biểu diễn của các phép phân tách

Thuật toán xây dựng cây quyết định hết sức thấu đáo. Chúng bắt đầu bằng việc chọn mỗi biến đầu vào chưa được chọn và đo mức độ tăng độ tinh khiết trong các kết quả ứng với mỗi biến. Sau đó một phép tách tốt nhất sẽ được sử dụng trong phép tách khởi đầu, để tạo hai hay nhiều nút con. Nếu không phép

phân tách nào có khả năng (có thể do có quá ít bản ghi) hoặc do không có phép phân tách nào làm tăng độ tinh khiết thì thuật toán kết thúc và nút đó trở thành nút lá.

Phép phân tách trên các biến đầu vào kiểu số: đối với sự phân tách nhị phân trên một biến đầu vào, mỗi giá trị mà biến đó chứa đều có thể trở thành giá trị dự tuyển. Phép phân tách nhị phân dựa trên biến đầu vào kiểu số có dạng X < N. Để cải thiện hiệu năng, một số thuật toán không kiểm tra hết toàn bộ các giá trị của biến mà chỉ kiểm tra trên tập mẫu giá trị của biến đó.

Phép phân tách trên các biến đầu vào định tính : thuật toán đơn giản nhất trong việc phân tách trên một biến định tính là ứng với mỗi giá trị của biến đó, ta tạo một nhánh tương ứng với một lớp được phân loại. Phương pháp này được sử dụng thực sự trong một số phần mềm nhưng mang lại hiệu quả thấp. Một phương pháp phổ biến hơn đó là nhóm các lớp mà dự đoán cùng kết quả với nhau. Cụ thể, nếu hai lớp của biến đầu vào có phân phối đối với biến đích chỉ khác nhau trong một giới hạn cho phép thì hai lớp này có thể hợp nhất với nhau.

Phép phân tách với sự có mặt của các giá trị bị thiếu: một trong những điểm hay nhất của cây quyết định là nó có khả năng xử lý các giá trị bị thiếu bằng cách coi giá trị rỗng (*NULL*) là một nhánh của nó. Phương pháp này được ưa thích hơn so với việc vứt các bản ghi có giá trị thiếu hoặc cố gắng gắn giá trị nào đó cho nó bởi vì nhiều khi các giá trị rỗng cũng có ý nghĩa riêng của nó. Mặc dù phép phân tách giá trị rỗng như là một lớp riêng rẽ khá có ý nghĩa nhưng người ta thường đề xuất một giải pháp khác. Trong khai phá dữ liêu, mỗi nút chứa vài luật phân tách có thể thực hiện tại nút đó, mỗi phép phân tách đó dựa vào các biến đầu vào khác nhau. Khi giá trị rỗng xuất hiên trong biến đầu vào của phép phân tách tốt nhất, ta sử dụng phép phân tách thay thế trên biến đầu vào có phép phân tách tốt thứ hai.

2) Cách kiểm tra để chọn phép phân tách tốt nhất:

Hiện nay, có nhiều cách để đánh giá cách chia là tốt hay không tốt. Các độ đo dùng để đánh giá và lựa chọn cách chia được định nghĩa trên gốc độ sự phân phối về lớp của các mẫu tin trước và sau khi bị chia. Gọi pi=p(i|t) là tỉ lệ các mẫu tin thuộc vào lớp I của nút t. Trong cách chia đôi, giả sử có hai lớp class=0 và class=1 thì p_1=1-p_0 (Với p_0, p_1 là xác suất của class=0 và class =1) . Độ đo được phát triển cho việc lựa chọn cách chia tốt nhất dựa trên mức độ không thuần nhất (impurity) của các nút con. Độ không thuần nhất càng nhỏ thì phân phối lớp càng lệch. Độ không thuần nhất có thể được đo bằng entropy, gini, classification error. Entropy, gini, classification error tại nút t được định nghĩa như sau:

$$Entropy = -\sum_{i=0}^{c-1} p(i \mid t) \log_2 p(i \mid t) \quad \text{(CT-II-12)}$$

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i \mid t)]^2 \quad \text{(CT-II-13)}$$

$$Classification\_error(t) = 1 - \max[p(i \mid t)] \quad \text{(CT-II-14)}$$

Trong đó, c là tổng số lớp, các lớp được đánh số từ 0 đến c-1.

3) Thuật toán săn mồi:

Thuật toán săn mồi hay còn gọi là thuật toán của những người thợ săn (Hunt's algorithm). Trong thuật toán này, cây quyết định được phát triển dựa trên phương pháp đệ quy bằng cách chia tập dữ liệu học ra thành các tập con một cách liên tục. Gọi D_t là tập dữ liệu học tại nút t và $y=\{y_1, y_2,\ldots,y_c\}$ là tập giá trị của thuộc tính lớp. Thuật toán săn mồi được định nghĩa một cách đệ quy như sau:

- Bước 1: Nếu tất cả các mẩu tin trong tập D_t thuộc về một lớp y_t thì t là nút lá được gán nhãn là y_t.
- Bước 2: Nếu tập hợp D_t chứa các mẫu tin thuộc nhiều hơn một lớp thì một điều kiện kiểm tra thuộc tính được lựa chọn là chia tập D_t ra thành các tập con nhỏ hơn. Ứng với một đường ra của điều kiện kiểm tra thuộc tính là một nút con dựa trên tập hợp con của D_t.
- Bước 3: Lặp lại các bước 1, 2 cho đến khi tập D_t chỉ thuộc một lớp.

4) Thuật toán ID3:

ID3 xây dựng cây quyết định theo cách từ trên xuống. Lưu ý rằng đối với bất kỳ thuộc tính nào, chúng ta cũng có thể phân vùng tập hợp các ví dụ rèn luyện thành những tập con tách rời, mà ở đó mọi ví dụ trong một phân vùng (partition) có một giá trị chung cho thuộc tính đó. ID3 chọn một thuộc tính để kiểm tra tại nút hiện tại của cây và dùng trắc nghiệm này để phân vùng tập hợp các ví dụ; thuật toán khi đó xây dựng theo cách đệ quy một cây con cho từng phân vùng. Việc này tiếp tục cho đến khi mọi thành viên của phân vùng đều nằm trong cùng một lớp; lớp đó trở thành nút lá của cây.

Vì thứ tự của các trắc nghiệm là rất quan trọng đối với việc xây dựng một cây quyết định đơn giản, ID3 phụ thuộc rất nhiều vào tiêu chuẩn chọn lựa trắc nghiệm để làm gốc của cây.

```
1  Function induce_tree(tập_ví_dụ, tập_thuộc_tính)
2   begin
3    if mọi ví dụ trong tập_ví_dụ đều nằm trong cùng một lớp then
4       return một nút lá được gán nhãn bởi lớp đó
5   else if tập_thuộc_tính là rỗng then
6    return nút lá được gán nhãn bởi tuyển của tất cả các lớp trong tập_ví_dụ
7   else
8      begin
9       chọn một thuộc tính P, lấy nó làm gốc cho cây hiện tại;
10      xóa P ra khỏi tập_thuộc_tính;
11      với mỗi giá trị V của P
12        begin
13          tạo một nhánh của cây gán nhãn V;
14         Đặt vào phân_vùngV các ví dụ trong tập_ví_dụ có giá trị V tại thuộc
tính P;
15              Gọi induce_tree(phân_vùngV, tập_thuộc_tính), gắn kết quả vào
nhánh V
16        end
17     end
18  end
```

5) Thuật toán C4.5:

- *Dữ liệu vào:* Tập dữ liệu D, tập danh sách thuộc tính, tập nhãn lớp
- *Dữ liệu ra:* Mô hình cây quyết định
- *Thuật toán:* Tạocây(Tập dữ liệu E, tập danh sách thuộc tính F, tập nhãn lớp)

```
1  Nếu điều_kiện_dừng(E,F) = đúng
2      nútlá = CreateNode()
3      nútlá.nhãnlớp=Phânlớp(E)
4      return nútlá
5  Ngược lại
6      Nútgốc = CreateNode()
7      Nútgốc.điềukiệnkiểmtra = tìm_điểm_chia_tốt_nhất(E, F)
8          Đặt F = F \ {Nút chọn phân chia}
9      Đặt V = {v| v thoả điều kiện là phần phân chia xuất phát từ Nútgốc}
10     Lặp qua từng tập phân chia v ∈ V
11         Đặt Ev = {e | Nútgốc.điềukiệnkiểmtra(e) = v và e ∈ E}
12         Nútcon = Tạocây(Ev, F, tập nhãn lớp)
13     Dừng lặp
14 End if
15 Trả về nútgốc.
```

6) Thuật toán luật quy nạp ILA (Inductive learning algorithm)

a) Ý tưởng:

- Xác định các luật IF-THEN trực tiếp từ tập huấn luyện (phát triển luật theo hướng từ tổng quát đến cụ thể)
- Chia tập dữ liệu huấn luyện thành các bảng con theo từng giá trị của lớp.
- Thực hiện việc so sánh các giá trị của thuộc tính trong từng bảng con và tính số lần xuất hiện.

b) Các bước xây dựng:

- Bước 1: Chia bảng con có chứa m mẫu thành n bảng con (ứng với n giá trị của thuộc tính lớp).
- Bước 2: Khởi tạo số thuộc tính kết hợp j=1
- Bước 3: Xét từng bảng con, tạo danh sách các thuộc tính kết hợp (phần tử danh sách có j thuộc tính)
- Bước 4: Với mỗi phần tử trong danh sách trên, đếm số lần xuất hiện các giá trị của thuộc tính ở các dòng chưa đánh dấu của bảng con đang xét, nhưng giá trị không được xuất hiện ở các bảng con khác.

☞ Chọn phần tử kết hợp đầu tiên có số lần xuất hiện của giá trị thuộc tính nhiều nhất và đặt tên là max-combination.

- Bước 5: Nếu max-combination=0 thì j=j+1 và quay lại bước 3.
- Bước 6: Trong bảng con đang xét, đánh dấu các dòng có xuất hiện giá trị của max-combination.

- Bước 7: Tạo luật

  IF AND (Thuộc tính = giá trị) (thuộc max-combination)

  THEN giá trị của thuộc tính lớp tương ứng với bảng con đang xét.

- Bước 8:
  - Nếu tất cả các dòng đều đánh dấu
    - Nếu còn bảng con thì chuyển qua bảng con tiếp theo và lập lại từ bước 2.
    - Ngược lại: Chấm dứt thuật toán.
  - Ngược lại (còn dòng chưa đánh dấu) thì quay lại bước 4.

II.4.3.5. Ưu điểm của cây quyết định:

- Cây quyết định dễ hiểu. Người ta có thể hiểu mô hình cây quyết định sau khi được giải thích ngắn.
- Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản hoặc không cần thiết. Các kỹ thuật khác thường đòi hỏi chuẩn hóa dữ liệu, cần tạo các biến phụ (dummy variable) và 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. Các kỹ thuật khác thường chuyên để phân tích các bộ dữ liệu chỉ gồm một loại biến. Chẳng hạn, các luật quan hệ chỉ có thể dùng cho các biến tên, trong khi mạng nơ-ron chỉ có thể dùng cho các biến có giá trị bằng số.
- Cây quyết định là một mô hình hộp trắng. Mạng nơ-ron là một ví dụ về mô hình hộp đen, do lời giải thích cho kết quả quá phức tạp để có thể hiểu được.
- Có thể thẩm định một mô hình bằng các kiểm tra thống kê. Điều này làm cho ta có thể tin tưởng vào mô hình.
- Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn. Có thể dùng máy tính cá nhân để phân tích các lượng dữ liệu lớn trong một thời gian đủ ngắn để cho phép các nhà chiến lược đưa ra quyết định dựa trên phân tích của cây quyết định.

II.4.3.6. Tính chất:

- Không cần có các giả định phân phối của các lớp trước khi xây dựng cây.
- Tìm cây tối ưu là vấn đề phức tạp vì tốn nhiều thời gian và tài nguyên cho việc tính toán và so sánh. Cách tiếp cận là dựa trên các kỹ thuật Heuristics.
- Các giải thuật xây dựng cây quyết định phải có độ phức tạp chấp nhận được.
- Đối với những cây tương đối nhỏ, rất dễ dịch sang câu lệnh IF-THEN-ELSE.
- Cây quyết định cung cấp các mô tả tập luật một cách phức tạp.
- Giải thuật xây dựng cây quyết định chịu ảnh hưởng của nhiễu.
- Sự thiếu hoặc dư thừa dữ liệu không ảnh hưởng nhiều lắm đến kết quả.
- Nếu tập training gồm nhiều mẫu tin thì cây kết quả có thể rất phức tạp và cần được "tỉa bớt nhánh" cho cây.
- Một nhánh hay cây con có thể lặp lại nhiều lần ở nhiều mức khác nhau.

II.4.4. K – lân cận gần nhất: (K Nearest neighbour-KNN)

II.4.4.1. Giới thiệu:

Thuật toán K- lân cận gần nhất là thuật toán khai khoáng dùng để phân loại dữ liệu. KNN là một thuật toán học có giám sát mà kết quả của truy vấn hiện mới được phân loại dựa trên đa số các loại KNN. Mục đích KNN được sử dụng là để phân loại các đối tượng mới dựa trên các thuộc tính và các mẫu đào tạo. Việc phân loại không sử dụng bất kỳ mô hình để điều chỉnh, mà chỉ dựa vào bộ nhớ. Với một điểm truy vấn, chúng ta tìm k đối tượng hoặc (điểm đào tạo) gần nhất để thực hiện truy vấn. Việc phân loại được sử dụng đa số trong việc phân loại K đối tượng. Bất kỳ mối quan hệ có thể được chia một cách ngẫu nhiên. KNN sử dụng trong các thuật toán phân loại lân cận là giá trị dự đoán của các ví dụ truy vấn mới.

Ví dụ: Chúng ta có dữ liệu từ bảng câu hỏi khảo sát (để lấy ý kiến người dân) và mục tiêu thử nghiệm hai thuộc tính (acid durability and strength) để phân loại xem một mô giấy là tốt hay không.

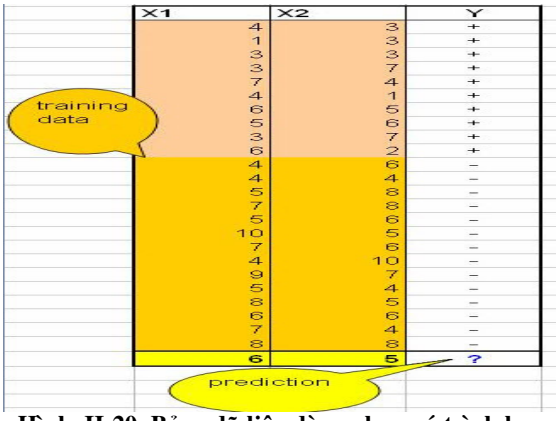
Bảng II-3

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

II.4.4.2. Tìm hiểu về K-lân cận gần nhất:

Thuật toán KNN rất đơn giản. Nó hoạt động dựa trên khoảng cách tối thiểu từ các mẫu truy vấn đến các mẫu đào tạo để xác định K- lân cận gần nhất. Sau khi chúng ta đã tập hợp được được những lân cận gần nhất, chúng ta sẽ sử dụng để dự đoán các mẫu truy vấn.

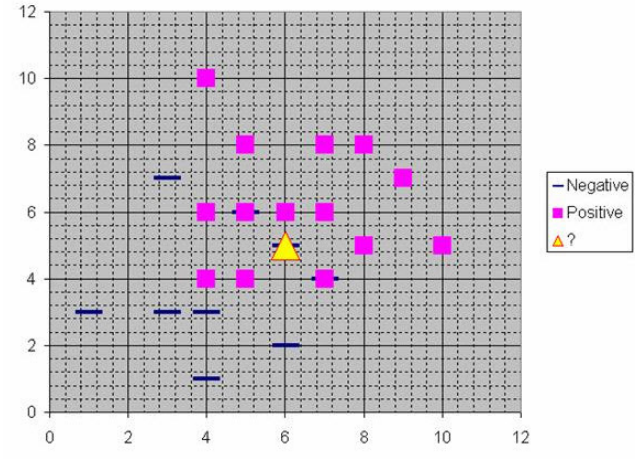
Các dữ liệu được dùng cho thuật toán KNN bao gồm: một số thuộc tính đa biến X_i sẽ được sử dụng để phân loại Y. Các dữ liệu của KNN là các dữ liệu bất kỳ, có thể là thứ tự, định danh(nominal), đến các giá trị định lượng nhưng thời điểm để chúng ta xử lý với các định lượng X_i và nhị phân (định danh) Y. Ở đây, chúng ta sẽ đối phó với loại quy mô đo lường.



Hình II-20. Bảng dữ liệu dùng cho quá trình học

Giả sử chúng ta có bảng dữ liệu sau: Với các tập dữ liệu mà chúng ta sử dụng cho quá trình học (training data) và dòng cuối cùng là vùng dữ liệu mà ta cần dự đoán (preditcion).

Đồ thị của vấn đề này được thể hiện như Hình II-21:



Hình II-21. Đồ thị biểu diễn

Giả sử chúng ta cần xác định K=8, có nghĩa là chúng ta sẽ sử dụng 8 lân cận gần nhất như một tham số của thuật toán này. Sau đó, chúng ta tính toán khoảng cách giữa các mẫu truy vấn và đào tạo. Bởi vì chúng ta sử dụng yếu tố định lượng X_i nên chúng ta có thể sử dụng khoảng cách Euclide để giải quyết bài toán này.

Giả dụ, truy vấn có tọa độ (x_1^q, x_2^q) và sự phối hợp của mẫu đào tạo là (x_1^t, x_2^t), thì bình phương khoảng cách Euclide $d_{Sq}^2 = (x_1^t - x_1^q)^2 + (x_2^t - x_2^q)^2$. Nếu X có chứa dữ liệu phân loại, hoặc định danh thì có thể áp dụng tương tự.

Bước tiếp theo để tìm K-lân cận gần nhất. Một mẫu đào tạo được gọi là lân cận gần nhất nếu khoảng cách của mẫu đào tạo này cho ác trường hợp truy vấn nhỏ hơn hoặc bằng với khoảng cách nhỏ nhất thứ K. Nói cách khác, chúng ta sẽ sắp xếp các khoảng cách của tất cả các mẫu đào tạo cho các trường hợp truy vấn và xác định khoảng cách tối thiểu thứ K.

Computation	
Distance	Nearest Neighbor sign
8	
29	
13	
13	
2	+
20	
0	+
2	+
13	
9	
5	
5	
10	
10	
2	-
16	
2	-
29	
13	
2	-
4	
1	-
2	-
13	

Hình II-22. Biểu diễn của mẫu đào tạo

Nếu khoảng cách của mẫu đào tạo là dưới mức tối thiểu thứ K, thì chúng ta tập hợp được Y thể loại của mẫu huấn luyện này.

Các bước thực hiện về cách tính toán K lân cận gần nhất, áp dụng thuật toán KNN cho các dữ liệu định lượng :

(1) Xác định tham số K-số lân cận gần nhất.

(2) Tính khoảng cách giữa tất cả các mẫu truy vấn và đào tạo

(3) Phân loại theo khoảng cách và xác định những lân cận gần nhất dựa vào khoảng cách tối thiểu thứ k.

(4) Tập hợp các giá trị của Y của những người lân cận gần nhất.

(5) Sử dụng trung bình của các lân cận gần nhất là giá trị dự đoán của các mẫu truy vấn.

II.4.4.3. Nhận xét:

- Thuật toán KNN có tác dụng để loại bỏ nhiễu trong dữ liệu huấn luyện (Đặc biệt là sử dụng hình vuông nghịch đảo của khoảng cách trọng số như là "khoảng cách).
- Thuật toán KNN sẽ hiệu quả hơn nếu dữ liệu đào tạo là lớn.

II.4.5. Giải thuật di truyền:

II.4.5.1. Giới thiệu:

Giải thuật di truyền (Genetic Algorithm – GA) là một phương pháp tìm kiếm cực trị tổng thể, kỹ thuật tối ưu tổng thể có tầm quan trọng rất lớn đối với nhiều vấn đề khác nhau trong khoa học và kỹ thuật. Trong khai phá dữ liệu, giải thuật di truyền thường được sử dụng trên nền của các kỹ thuật khác như mạng neuron hay phân lớp theo k lân cận gần nhất. Mặc dù vậy, giải thuật di truyền là một kỹ thuật rất cần thiết vì hầu hết các kỹ thuật khai phá dữ liệu tóm lại đều là vấn đề tối ưu hóa.

- Đối với mạng neuron, đó là vấn đề tìm kiếm các trọng số cho cấu trúc mạng tối ưu.
- Đối với k lân cận gần nhất, đó là vấn đề tìm các trọng số quan trọng tối ưu để áp dụng cho mỗi yếu tố dự đoán.
- Đối với cây quyết định, đó là bài toán tìm kiếm các yếu tố dự đoán tốt nhất và các giá trị để phân tách trong việc tối ưu hóa cây.

Giải thuật di truyền được đánh giá bằng hàm thích nghi để xác định các mô hình dự đoán tối ưu cho việc khai thác dữ liệu.

II.4.5.2. Cơ bản về giải thuật di truyền:

Ý tưởng của giải thuật di truyền là mô phỏng theo cơ chế của quá trình chọn lọc và di truyền trong tự nhiên. Từ tập các lời giải ban đầu, thông qua nhiều bước tiến hóa để hình thành các tập mới với những lời giải tốt hơn, cuối cùng sẽ tìm được lời giải tối ưu nhất.

GA sử dụng các thuật ngữ lấy từ di truyền học:

- Một tập hợp các lời giải được gọi là một lớp hay quần thể (population).
- Mỗi lời giải được biểu diễn bởi một nhiễm sắc thể hay các thể (chromosome).

- Nhiễm sắc thể được tạo thành từ các gien.

Một quá trình tiến hóa được thực hiện trên một quần thể tương đương với sự tìm kiếm trên không gian các lời giải có thể của bài toán. Quá trình tìm kiếm này luôn đòi hỏi sự cân bằng giữa hai mục tiêu: Khai thác lời giải tốt nhất và xem xét toàn bộ không gian tìm kiếm.

GA thực hiện tìm kiếm theo chiều hướng bằng cách duy trì tập hợp các lời giải có thể và khuyến khích sự hình thành và trao đổi thông tin giữa các hướng.

Tập lời giải phải trãi qua nhiều bước tiến hóa, tại mỗi thế hệ, một tập mới các cá thể được tạo ra, và có chứa các phần của những cá thể thích nghi nhất trong thế hệ cũ. Đồng thời, giải thuật di truyền cũng khai thác một cách có hiệu quả thông tin trước đó để suy xét trên điển hình tìm kiếm mới, với mong muốn có được sự cải thiện qua từng thế hệ. Như vậy, các đặc trưng được đánh giá tốt sẽ có cơ hội phát triển và các tính chất xấu (không thích nghi với môi trường) sẽ có xu hướng biến mất.

Giải thuật di truyền tổng quát được mô tả như sau:

```
1  PROCEDURE GeneticAlgorithm;
2   BEGIN
3       T:=0;
4       Khởi tạo lớp P(t);
5       Đánh giá lớp P(t);
6       While not(Điều_kiện_kết_thúc) do
7           Begin
8                t:=t+1;
9               Chọn lọc P(t) từ P(t-1);
10              Kết hợp các cá thể của P(t);
11              Đánh giá lớp P(t);
12           End;
13  END;
```

Trong đó:

- Tập hợp các lời giải ban đầu được khởi tạo ngẫu nhiên.

- Trong vòng lặp thứ t, GA xác định tập các nhiễm sắc thể $P(t) = \{x_1^t, x_2^t, ..., x_n^t\}$ bằng cách chọn lựa các nhiễm sắc thể thích nghi hơn là từ P(t-1). Mỗi nhiễm sắc thể x_i^t được đánh giá để xác định độ thích nghi của nó và một số thành viên của P(t) lại được tái sản xuất nhờ các toán tử Lai ghép và Đột biến.

Khi áp dụng GA để giải quyết các bài toán cụ thể, phải làm rõ các vấn đề sau:

(1) Chọn cách biểu diễn di truyền nào đối với những lời giải có thể của bài toán?

(2) Tạo tập lời giải ban đầu như thế nào?

(3) Xác định hàm đánh giá để đánh giá mức đột thích nghi của các cá thể.

(4) Xác định các toán tử di truyền để sản sinh ra con cháu.

(5) Xác định giá trị của các tham số mà GA sử dụng như kích thước các tập lời giải, xác suất áp dụng các toán tử di truyền,…

II.4.5.3. Các toán tử di truyền:

Các cá thể trong giải thuật di truyền là các chuỗi bit được tạo bởi việc cắt dán các chuỗi bit con. Mỗi chuỗi bit đại diện cho một tập các thông số trong không gian tìm kiếm, nên được gọi là lời giải tìm năng của bài toán tối ưu. Từ một chuỗi bit ta giải mã để tính lại tập các thông số, sau đó tính được giá trị của hàm mục tiêu. Từ đó, giá trị hàm mục tiêu được biến đổi thành giá trị đo phù hợp của từng chuỗi.

Quần thể ban đầu được khởi tạo ngẫu nhiên, sau đó tiến hóa từ thế hệ này sang thế hệ khác bằng các toán tử di truyền (tổng số chuỗi trong mỗi quần thể là không thay đổi). Có 3 toán tử di truyền đơn giản là:

- Tái tạo.
- Lai ghép.
- Đột biến.

a) Đánh giá độ thích nghi của cá thể và phép tái tạo

Mỗi bài toán trong thực tế có các điều kiện ràng buộc khác nhau đối với lời giải. Quá trình tìm kiếm lời giải chính là quá trình tiến hóa mà ở mỗi bước, cần phải lựa chọn các cá thể thích nghi hơn để tái sản xuất ở thế hệ sau bằng phép tái tạo.

Để đánh giá các lời giải, người ta xây dựng hàm thích nghi Fitness(). Tái tạo là quá trình sao chép các chuỗi (các cá thể) từ thế hệ trước sang thế hệ sau theo giá trị của hàm thích nghi (còn gọi là hàm mục tiêu hay hàm sức khỏe).

Toán tử này mô phỏng theo học thuyết của Darwin, chỉ có các cá thể khỏe mới có cơ hội sống sót và đóng góp con cháu vào các thế hệ sau.

Hàm thích nghi được xây dựng như sau:

- Xét lời giải P có n cá thể, với mỗi cá thể h_i thuộc P, tính độ thích nghi Fitness(h_i).
- Xác suất chọn cá thể h_i để tái sản xuất được xác định bởi công thức:

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{n} Fitness(h_j)} \quad \text{(CT-II-15)}$$

- Tại mỗi bước tiến hóa, các cá thể được chọn tái tạo là các cá thể có xác suất Pr() cao, điều này cho phép tạo ra thế hệ sau có độ thích nghi tốt hơn thế hệ trước.

Fitness() còn được dùng để xác định điểm dừng của quá trình tìm kiếm lời giải khi đã đạt được độ thích nghi chấp nhận được.

b) Lai ghép (Crossover)

Các cá thể trong quần thể sau khi đã tái tạo sẽ được chọn lai ghép với nhau. Toán tử lai ghép được coi là toán tử di truyền quan trọng nhất, nó kết hợp các đặc trưng của các cá thể bố mẹ để tạo ra hai cá thể con bằng cách tráo đổi các đoạn gien tương ứng trên hai cá thể cha mẹ.

Phép lai ghép chọn nhẫu nhiên hai chuỗi bất kì trong quần thể sau khi đã thực hiện táo tạo, đồng thời sinh ra một số ngẫu nhiên, nếu nhỏ hơn xác suất lai

ghép p_c thì thực hiện lai ghép, ngược lại chỉ thực hiện sao chép đơn giản hai chuỗi vào quần thể mới. Phép lai ghép hai chuỗi thực hiện tráo đổi hai đoạn mã cho nhau, rồi đưa hai chuỗi kết quả vào một quần thể mới.

c) Đột biến (Mutation)

Tái tạo và lai ghép chỉ tạo ra các chuỗi mới chứ không đem lại cho quần thể một thông tin mới. Phép đột biến ngăn ngừa khả năng GA chỉ tìm kiếm trên một vùng cục bộ và kết quả chỉ là cực trị địa phương.

Toán tử đột biến sẽ thay đổi ngẫu nhiên một bit thông tin của một chuỗi với xác suất đột biến p_m. Xác suất đột biến thể hiện mức độ thường xuyên được thực hiện của toán tử đột biến phải đủ nhỏ vì thực tế toán tử đột biến là toán tử tìm kiếm ngẫu nhiên.

Với phương pháp mã hóa chuỗi bit, một bit thông tin A nếu bị đột biến được biến đổi bằng công thức đơn giản: A=1-A.

Ba toán tử tái tạo, lai ghép và đột biến được tiến hành lặp đi lặp lại cho đến khi các chuỗi con chiếm toàn bộ quần thể mới. Quần thể mới bao gồm các cá thể chủa ba loại: Lai ghép nhưng không đột biến, bị đột biến sau khi lai ghép và không lai ghép cũng không đột biến mà chỉ đơn thuần là sao chép lại.

Như vậy, trong một giải thuật di truyền đơn giản, chúng ta cần xác định các thông số sau:

- Số các cá thể trong quần thể n.
- Xác suất lai ghép p_c.
- Xác suất đột biến p_m.
- Độ gối của quần thể G.

Ba thông số đầu rất dễ hiểu và đã được nhắc đến. Còn độ gối G được tác giả De Jong đưa vào năm 1975, ý nghĩa của nó là cho phép quần thể mới chứa một phần của quần thể cũ: Với G=1, tất cả các cá thể mới đều được sinh ra bởi các toán tử của giải thuật di truyền, với 0<G<1, sẽ có G*n cá thể được đưa ra trực tiếp từ quần thể cũ sang quần thể mới.

II.4.5.4. Nhận xét:

GA là một giải thuật lặp nhằm giải quyết các bài toán tìm kiếm, nó khác với các thủ tục tối ưu thông thường ở những điểm cơ bản sau:

- Giải thuật di truyền làm việc với bộ mã của tập thông số chứ không làm việc trực tiếp với giá trị của các thông số.
- Giải thuật di truyền tìm kiếm song song trên một quần thể chứ không tìm kiếm từ một điểm, mặt khác, nhờ áp dụng các toán tử di truyền, nó sẽ trao đổi thông tin giữa các điểm, như vậy sẽ giảm bớt khả năng kết thúc tại một điểm cực tiểu cục bộ mà không tìm thấy cực tiểu toàn cục.
- Giải thuật di truyền chỉ sử dụng thông tin của hàm mục tiêu để đánh giá quá trình tìm kiếm chứ không đòi hỏi các thông tin bổ trợ khác.
- Các luật chuyển đổi của giải thuật di truyền mang tính xác suất chứ không mang tính tiền định.

II.4.6. Mạng neuron nhân tạo (Neural networks):

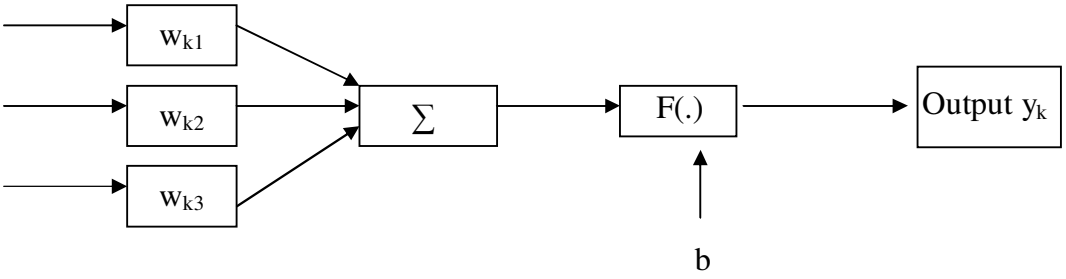
II.4.6.1. Giới thiệu:

Mạng neuron nhân tạo là một mô phỏng xử lý thông tin, được nghiên cứu ra từ hệ thống thần kinh của sinh vật, giống như bộ não để xử lý thông tin. Nó bao gồm số lượng lớn các mối gắn kết cấp cao để xử lý các yếu tố làm việc trong mối liên hệ giải quyết vấn đề rõ ràng. ANNs giống như con người, được học bởi kinh nghiệm, lưu những kinh nghiệm hiểu biết và sử dụng trong những tình huống phù hợp.

Đầu tiên ANN được giới thiệu năm 1943 bởi nhà thần kinh học Warren McCulloch và nhà logic học Walter Pits. Nhưng với những kỹ thuật trong thời gian này chưa cho phép họ nghiên cứu được nhiều. Những năm gần đây mô phỏng ANN xuất hiện và phát triển.

II.4.6.2. Cấu trúc mạng neuron:

Mỗi neuron (nút) là một đơn vị xử lý thông tin của mạng neural, là yếu tố cơ bản để cấu tạo nên mạng neuron.



Hình II-23. Cấu trúc của một mạng neuron

Trong đó:

- x_i: các tín hiệu input
- w_{kp}: trọng số của từng input
- f(.): hàm hoạt động
- y_k: kết xuất của Neural
- b: thông số ảnh hưởng đến ngưỡng ra của output

II.4.6.3. Mô hình và quá trình xử lý trong mạng neuron:

a) Hàm truyền trong mạng neuron:

Cấu trúc của mạng neuron chủ yếu được đặc trưng bởi loại của các neuron và mối liên hệ xử lý thông tin giữa chúng. Về cấu trúc của neuron, chủ yếu người ta quan tâm tới cách tổng hợp các tín hiệu đầu vào, giá trị ngưỡng tại mỗi neuron và các hàm truyền.

Hàm truyền xác định mức độ liên kết bên trong các neuron. Hàm truyền có nhiệm vụ tạo mức kích thích của neuron, từ đó sẽ làm hưng phấn hoặc ức chế các neuron khác trong mạng.

Trong lý thuyết mạng neuron, phép tổng hợp tín hiệu đầu vào của neuron I có m tín hiệu đầu vào x_j thường được ký hiệu:

$$net_i = \sum_{j=1}^{m} w_{ij} x_j ; w_{ij} = (w_{i1}, w_{i2}, ..., w_{im}) \quad \text{(CT-II-16)}$$

Tín hiệu ra tại neuron i thường ký hiệu là out_i hoặc f_i, được tính theo công thức sau với f là hàm truyền:

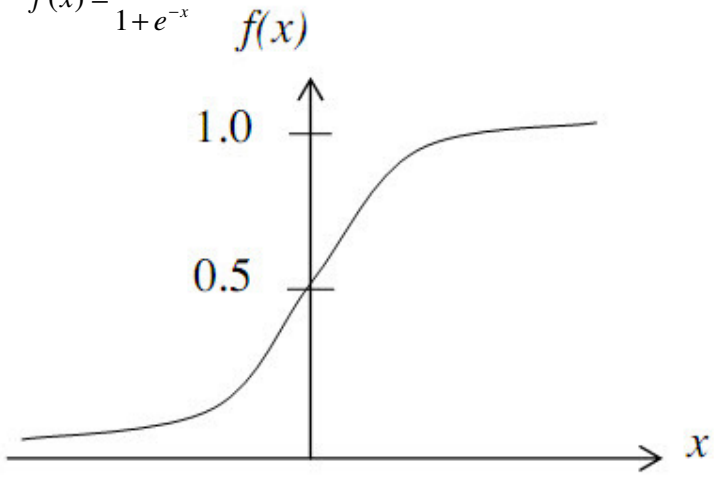
$$out_i(t) = f(net_i(t)) \qquad \text{(CT-II-17)}$$

Có nhiều hàm truyền khác nhau được sử dụng trong từng trường hợp cụ thể, các hàm truyền nói chung nên thỏa mãn các tính chất sau:

- Bị chặn: $|f(x)| \leq M, \forall x$
- Đơn điệu tăng: $f(x_1) > f(x_2), \forall x_1 > x_2$
- Khả vi liên tục: f(x) có đạo hàm f'(x) và f'(x) là hàm liên tục.

Trong thực tế, khi xét các neuron, chúng chỉ có thể có hai trạng thái là bị kích hoạt hoặc không bị kích hoạt. Nghĩa là tín hiệu ra của một neuron cần phải đảm bảo sao cho có thể nhận biết được neuron đó có bị kích hoạt hay không. Vì lý do đó, hàm truyền phải thỏa mãn điều kiện tín hiệu ra cuối cùng của neuron phải liên tục và nằm trong một giới hạn xác định (có thể là giữa 0 và 1). Sau đây là một số hàm truyền thường được sử dụng:

- Hàm ranh giới cứng (Hard-limiter): $f(x) = \begin{cases} 1, if\ (x \geq \theta) \\ 0, if\ (x < \theta) \end{cases}$
- Hàm ranh giới bất đối xứng: $f(x) = \begin{cases} 1, if\ (x \geq \theta) \\ -1, if\ (x < \theta) \end{cases}$
- Hàm Gauss: $f(x) = e^{-x^2}$
- Hàm Sigmoidal hay hàm logicstic (còn gọi là hàm chữ S): $f(x) = \dfrac{1}{1+e^{-x}}$



Hình II-3. Hàm sigmoidal

b) Cấu trúc mạng neuron:

Trong mô hình mạng neuron nhân tạo, các neuron được nối với nhau bởi các liên kết neuron, mỗi liên kết có một trọng số đặc trưng cho đặc tính kích hoạt hay ức chế giữa các neuron. Đồng thời, các neuron được nhóm lại với nhau theo cấu trúc phân lớp, bao gồm:

- Lớp vào (input layer): Các nút trong lớp vào gọi là nút vào, chúng mã hóa mẫu được đưa vào mạng xử lý. Các neuron vào không xử lý thông tin, chỉ

phân tán thông tin cho các nút khác (trên biểu đồ chúng được vẽ khác các nút ẩn và các nút ra để phân biệt giữa các nút có xử lý và không xử lý thông tin).

- Lớp ẩn (hidden layer): Các neuron ở lớp ẩn gọi là các nút ẩn vì chúng không thể quan sát trực tiếp. Chúng tạo thành các mô hình toán học phi tuyến cho mạng.

- Lớp ra (output layer): Các neuron trong lớp này gọi là các nút ra, chúng có nhiệm vụ đưa thông tin ra thích nghi với mẫu mà người sử dụng cần.

Một mạng được gọi là kết nối đầy đủ nếu tất cả các nút của một lớp được nối với tất cả các nút liền kề nó. Có nhiều loại kết nối khác nhau:

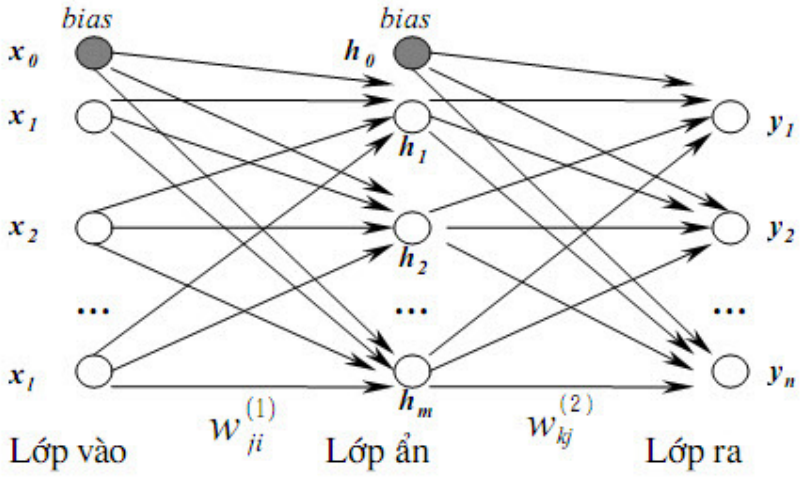
- Kết nối liên lớp là kết nối giữa các nút trong các lớp khác nhau.
- Kết nối trong lớp là kết nối giữa các nút trong cùng một lớp.
- Tự kết nối là kết nối từ một nút tới chính nó.
- Kết nối siêu lớp là kêt nối giữa các lớp khác nhau (không kề nhau).

Một kết nối bậc cao là một kết nối với nhiều nút đầu vào. Số các nút đầu vào xác định bậc kết nối và bậc kết nối của mạng là bậc kết nối bậc cao nhất.

c) Phân loại mạng neuron:

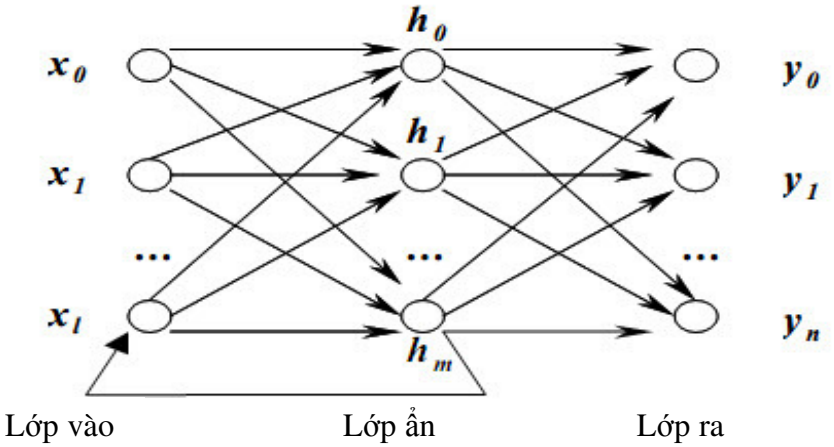
❖ Phân loại theo kiểu liên kết neuron:

- Mạng neuron truyền thẳng (feed-forward Neural netword): Trong mạng, các liên kết neuron chỉ đi theo một hướng từ lớp vào đến lớp ra, không tạo thành chu trình với các đỉnh là các neuron, các cung là các liên kết giữa chúng.



Hình II-25. Mạng neuron truyền thẳng nhiều lớp

- Mạng hồi quy: cho phép các liên kết neuron tạo thành chu trình, có thông tin được xử lý theo hai chiều. Vì các thông tin ra của các neuron được truyền lại cho các neuron đã góp phần kích hoạt chúng nên mạng hồi quy còn có khả năng lưu giữ trạng thái trong của nó dưới dạng các ngưỡng kích hoạt ngoài các trọng số liên kết neuron.

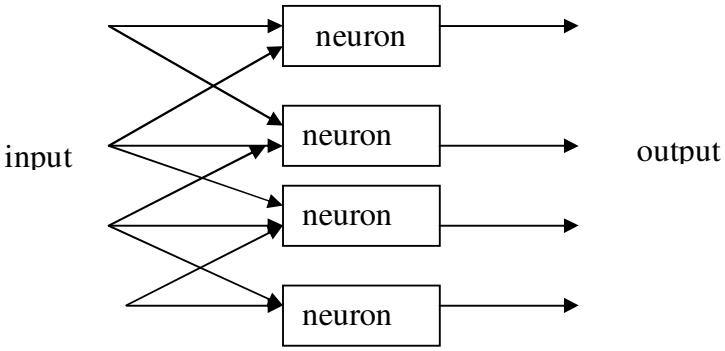


Hình II-26. Mạng neuron hồi quy

- Mạng kết nối đối xứng và không đối xứng: Mạng kết nối đối xứng là mạn thỏa mãn điều kiện: nếu có một đường nối từ nút i đến nút j thì cũng có một đường nối từ nút j đến nút I và trọng số tương ứng với hai đường nối này là bằng nhau:$w_{ij}=w_{ji}$. Và mạng không thỏa mãn điều kiện đối xứng là mạng kết nối không đối xứng.

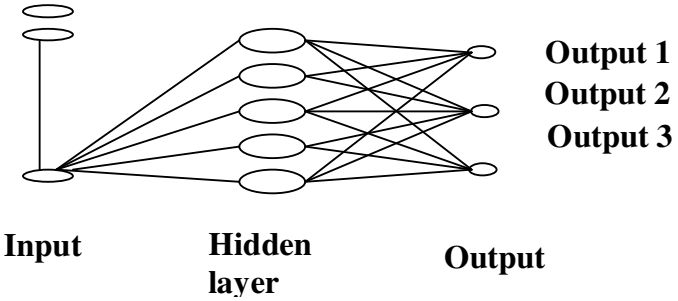
❖ Phân loại theo số lớp:

- Mạng chỉ gồm một lớp vào và một lớp ra gọi là mạng đơn lớp hay mạng một lớp.



Hình II-27. Mạng neuron đơn lớp

- Mạng có từ một lớp ẩn trở lên được gọi là mạng đa lớp hay mạng nhiều lớp.



Hình II-28. Mạng neuron đa lớp

II.4.6.4. Học và lan truyền trong mạng:

1) Học và tổng quát hóa

Mạng neuron thực hiện hai chức năng quan trọng là học và tổng quát hóa. Học là quá trình hiệu chỉnh các tham số và các trọng số liên kết trong mạng để tối thiểu hóa sai số với véc tơ đầu vào cho trước. Quá trình học dừng khi thỏa mãn một tiêu chuẩn dừng nào đó, chẳng hạn khi các trọng số của mạng tạo ra lỗi đủ nhỏ giữa đầu ra mong đợi và kết quả đầu ra tính được từ mạng.

Bài toán học có thể được mô tả như: Cho tập mẫu (X_i,Y_i) với X_i và Y_i là hai véc tơ trong không gian một chiều hoặc nhiều chiều, cần xác định bộ trọng số W_0 trên không gian tham số computer$(X_i, W_0)=Y_i$.

Quá trình học được thực hiện theo hai bước: Xác định hàm giá trị trên các tham số và tối thiểu hóa tham số trong không gian của các tham số.

Xét về mặt cấu trúc, học được chia làm hai loại là: học tham số và học cấu trúc.

- Học tham số: Là quá trình xác định một tập hợp tham số W_0 là các trọng số tốt nhất với một cấu trúc mạng cố định. Để làm được điều này, cần xây dựng một hàm giá trị dựa trên tập dữ liệu T_{min} và tập trọng số W. Hàm giá trị có thể là một hàm khả vi bất kỳ có tính chất đạt đến cực tiểu khi các đầu ra O_i đúng bằng đầu ra lý tưởng Y_i của tập mẫu. Có thể xây dựng hàm giá trị dưới dạng L_n-neuron như sau:

$$E=\frac{1}{p}\sum(y_i-O_i)^p \quad ,with \quad 1\le p\le\infty \quad \text{(CT-II-18)}$$

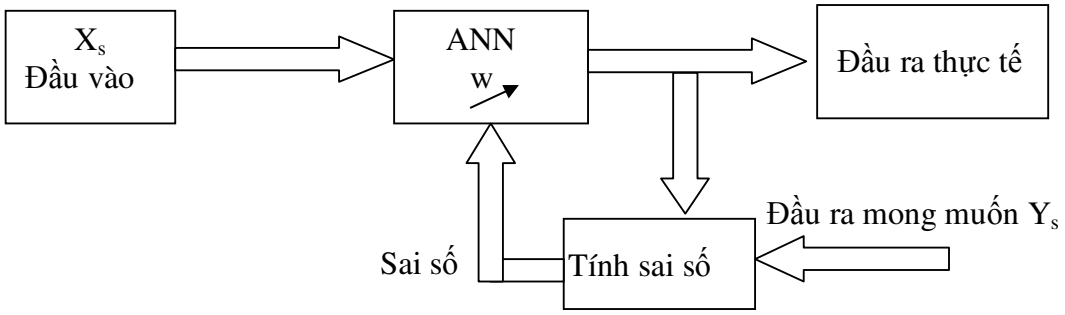
Với bộ tham số này, có thể áp dụng một giải thuật tìm kiếm nào đó trên không gian R^m của tập trọng số. Nếu thu được kết quả tốt với một cực tiểu toàn cục, ta sẽ có một tham số tốt nhất cho mạng.

- Học cấu trúc: Với học tham số, ta giả định rằng mạng có một cấu trúc cố định. Việc học cấu trúc của mạng truyền thẳng gắn với yêu cầu tìm ra số lớp của mạng L và số neuron trên mỗi lớp n_j. Tuy nhiên, với các mạng hồi quy còn phải xác định thêm các tham số ngưỡng θ của các neuron trong mạng. Một cách tổng quát là phải xác định các tham số P=(L,n_1,…,n_k, θ_1,...,θ_k).

Các kỹ thuật học của mạng neuron chỉ ra cách chỉnh sửa các trọng số liên kết mạng khi một mẫu học được đưa vào mạng. Sau đây, là các trình bày cụ thể về các kỹ thuật học:

a) Học có giám sát:

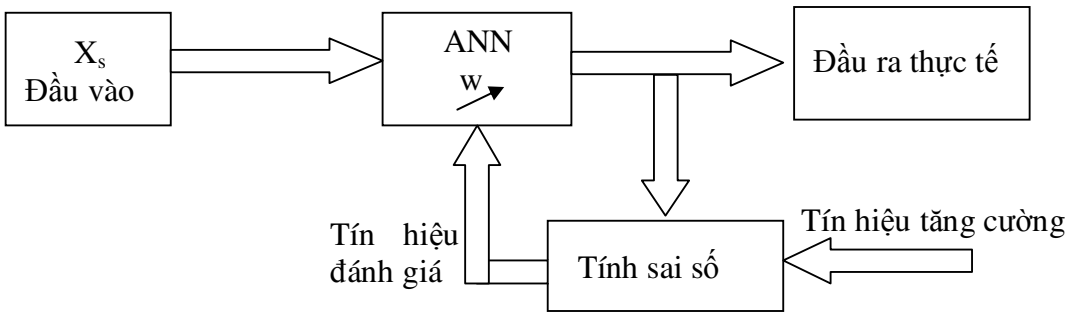
Với các phương pháp học có giám sát (supervised learning), khi đấy, mạng được cung cấp một tập mâu học $\{(X_s,Y_s)\}$ theo nghĩa X_s là tín hiệu vào, thì kết quả ra đúng của hệ phải là Y_s. Ở mỗi lần học, véc tơ tín hiệu vào X_s được đưa vào mạng, sau đó so sánh sự sai khác giữa các kết quả đúng Y_s với kết quả tính toán mạng out. Sai số này sẽ được dùng để hiệu chỉnh lại các trọng số liên kết trong mạng. Quá trình cứ tiếp tục cho đến khi thỏa mãn một tiêu chuẩn nào đó. Có hai cách sử dụng tập mẫu học: hoặc dùng các mẫu lần lượt, hết mẫu này đến mẫu khác; hoặc sử dụng đồng thời tất cả các mẫu.



Hình II-4. Sơ đồ học có giám sát

b) Học tăng cường:

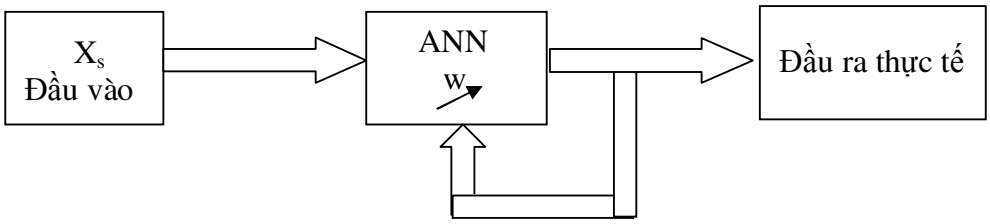
Ta thấy trong kỹ thuật học có giám sát, các véc tơ đầu ra được biết một cách chính xác, nhưng trong một số trường hợp có ít thông tin, chẳng hạn chỉ có thể nói là mạnh sinh Output quá lớn hoặc chỉ đúng khoảng 40%. Khi đó, chỉ có tín hiệu đánh giá là "True" hoặc False quay lại mạng, các thủ tục học đó gọi là thủ tục học tăng cường.



Hình II-30. Sơ đồ học tăng cường

c) Học không có giám sát:

Trong phương pháp học không giám sát (unsepervised learning), đầu ra mong muốn của mạng không được cho trước và mạng được trang bị khả năng tự tổ chức. Mạng không sử dụng mối quan hệ lớp của các mẫu học mà dùng thông tin kết hợp với nhóm các nơ-ron để thay đổi các tham số học cục bộ sao cho hợp nhất. Hệ thống học không giám sát được chia các mẫu vào các nhóm hoặc các lớp quyết định bằng cách chọn các neuron "chiến thắng" và thay đổi các trọng số tương ứng của chúng. Thông thường, việc học không giám sát dùng nhiều tham số kỹ thuật hơn học có giám sát.



Hình II-31. Sơ đồ học không giám sát

Như vậy, giải thuật học là giải thuật xuất phát từ một tập mẫu, qua quá trình huấn luyện để tìm ra bộ trọng số liên kết giữa các neuron, có thể mô tả tổng quát như sau:

- Đầu vào: Một tập mẫu gồm n phần tử.
- Đầu ra: Cấu trúc mạng và bộ trọng số các liên kết neuron.
- Giải thuật:

1. Khởi tạo trọng số của mạng, đặt i=1;
2. Đưa mẫu I vào lớp vào của mạng;
3. Sử dụng thuật toán lan truyền, nhận được giá trị các nút ra.

Nếu giá trị đầu ra của mạng đạt yêu cầu thỏa mãn tiêu chuẩn dừng thì kết thúc.

4. Sửa đổi trọng số bằng luật học của mạng;
5. Nếu i=n thì đặt lại i=1, nếu không thì tăng i len 1:i=i+1

Quay lại bước 2.

Có nhiều tiêu chuẩn dừng quá trình học, ví dụ:

- Chuẩn lỗi E nhỏ hơn một ngưỡng cho trước: E< θ.
- Các trọng số của mạng thay đổi nhiều sau khi hiệu chỉnh; $\left|w_{ij}^{new} - w_{ij}^{old}\right| < \theta$.
- Việc lặp lại bão hòa, tức là số lần vượt quá một ngưỡng N cho trước.

2) Lan truyền trong mạng

Mạng neuron lan truyền thông tin từ lớp vào đến lớp ra. Khi việc lan truyền kết thúc, thông tin tại lớp ra chính là kết quả của quá trình lan truyền.

Giải thuật lan truyền được mô tả như sau:

- Đầu vào: Một tập tín hiệu vào.
- Đầu ra: Kết quả tương ứng với tập tín hiệu vào
- Giải thuật:

1. Đưa tín hiệu vào lớp vào của mạng.
2. Tính mức tích cực của các nút trong mạng.
3. Với mạn truyền thẳng: Nếu mức tích cực của nút ra đã biết thì kết thúc.

Với mạng phản hồi: Nếu mức tích cực của nút ra bằng hoặc xấp xỉ bằng hằng số thì kết thúc. Nếu không thì quay lại bước 2.

II.4.6.5. Nhận xét:

Mạng neuron là một công cụ hữu hiệu trong các mô hình tính toán thông minh với một số đặc điểm chính sau:

- Cho phép xây dựng một mô hình tính toán có khả năng học dữ liệu cao: Chỉ cần đưa vào cho mạng một tập dữ liệu trong quá trình học là mạng có thể phát hiện những ràng buộc dữ liệu và áp dụng những ràng buộc này trong quá trình sử dụng mà không cần có thêm các tri thức về miền ứng dụng. Khả năng này cho phép xây dựng mô hình dữ liệu khá dễ dàng.

- Xử lý các quá trình phi tuyến: Mạng có khả năng xấp xỉ những ánh xạ phi tuyến tùy ý nên có thể giải được những bài toán phi tuyến phức tạp. Nó có thể thực hiện nhiều phép lọc nằm ngoài khả năng của những bộ lộc tuyến tính thông thường. Đặc trưng này rất quan trọng, ví dụ: trong xấp xỉ mạng, miễn nhiễu (chấp nhận nhiễu) và có khả năng phân lớp.

- Khả năng của các quá trình xử lý song song và phân tán: Có thể đưa vào mạng một lượng lớn các neuron liên kết với nhau theo những lược đồ với kiến trúc khác nhau. Mạng có cấu trúc song song lớn, có khả năng tăng tốc độ tính toán và hy vọng sẽ đáp ứng được yêu cầu của những hệ thống cần độ chính xác cao hơn những hệ thống truyền thống.

- Mạng neuron có khả năng học lỗi cao: Cố gắng bắt chước khả năng xử lý lỗi của bộ não, để bỏ qua các lỗi ấy và tiếp tục làm việc, điều chỉnh khi nhận tín hiệu vào có một phần thông tin bị sai lệch hoặc thiếu.

- Khả năng thích nghi và sự tổ chức: về đặc trưng này, người ta đề cập tới khả năng thích nghi và điều chỉnh bền vững dựa vào các thuật toán thích nghi và các quy tắc tự tổ chức.

- Hơn nữa, mặc dù có rất nhiều kỹ thuật và giải thuật được sử dụng trong khai phá dữ liệu, một số kỹ thuật còn được kết hợp để sử dụng có hiệu quả, song mạng neuron vẫn có những ưu điểm đáng chú ý như:

  - Tự động tìm kiếm tất cả các mối quan hệ có thể giữa các nhân tố chính.
  - Mô hình hóa tự động các bài toán phức tạp mà không cần biết trước mức độ phức tạp.
  - Có khả năng chiết xuất ra những thông tin nhanh hơn rất nhiều so với nhiều công cụ khác.

II.4.7. Luật kết hợp (Association rule):

II.4.7.1. Các định nghĩa:

1) Biểu diễn nhị phân:

Biểu diễn nhị phân là một sự biểu diễn các giá trị của giao dịch bằng một bảng nhị phân. Tổng số cột của bảng này bằng tổng số mặt hàng +1, cột đầu tiên là mã giao dịch, các cột còn lại tương ứng với mặt hàng. Mỗi dòng tương ứng với một giao dịch. Nếu giao dich I chứa các mặt hàng (x,y,z) thì giá trị của các ô (i,x), (i,y) và (i,z) là số 1, các ô còn lại có giá trị là số 0.

2) Itemset và support count:

Gọi $I=\{i_1, i_2,\ldots,i_d\}$ là tập hợp tất cả các mặt hàng (items). Và gọi $T=\{t_1, t_2,\ldots,t_N\}$ là tập hợp các giao dịch. Mỗi giao dịch tại thời gian t_i chứa một tập con của tập I. Trong phân tích kết hợp, một tập hợp chứa 0 hoặc n items được gọi là itemset. Nếu một itemset chứa k items thì được gọi là k-itemset.

Một tính chất quan trọng của itemset là support count. Support count của một itemset được định nghĩa là tổng số giao dịch chứa itemset đó. Support count của một itemset X được tính bằng công thức sau:

$$\sigma(X) = \left|\{t_i \mid X \subseteq t_i, t_i \in T\}\right| \quad \text{(CT-II-19)}$$

Trong đó, |.| ký hiệu cho số phần tử của tập hợp.

3) Luật kết hợp:

Mỗi luật kết hợp là một biểu thức suy diễn có dạng X->Y, với X và Y là hai itemset rời nhau. Sức mạnh của luật kết hợp được đo bằng support và confidence. Support xác định tỉ lệ mà một luật thỏa cho một tập dữ liệu cho trước. Confidence xác định tập các item xuất hiện thường xuyên (frequently items) trong biểu thức Y xuất hiện trong các giao dịch chứa biểu thức X. Công thức support và congidence được tính như sau:

Support: $$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \quad \text{(CT-II-20)}$$

Confidence: $$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \quad \text{(CT-II-21)}$$

Trong đó, N là tổng số giao dịch

Tính chất quan trọng của support là ở chỗ một luật kết hợp có giá trị support thấp nghĩa là có rất ít khi xảy ra tình huống theo sự suy diễn của luật đó hay nói cách khác các luật kết hợp có support thấp sẽ không hữu ích trong khi sử dụng, cần được loại bỏ.

Confidence dùng để đo độ tin cậy của luật kết hợp. Nếu một luật có độ tin cậy cao thì có nhiều giao dịch thỏa luật hay nói cách khác một luật kết hợp có confidence càng cao thì càng có giá trị sử dụng.

II.4.7.2. Phát hiện luật kết hợp:

Cho trước tập giao dịch T có N giao dịch. Yêu cầu tìm tất cả các luật thỏa điều kiện support>=minisup và confidence>=miniconf. Trong đó minisup và miniconf là giá trị support và confidence nhỏ nhất được cho trước.

Có 2 phương pháp tiếp cận việc phát hiện luật kết hợp:

- Một phương pháp tiếp cận để tìm các luật kết hợp có tên gọi "a brute-force approach" là tính giá trị support và confidence cho tất cả các luật kết hợp có thể có và xóa đi những luật có support<minisup hoặc confidence<miniconf. Phương pháp này tốn nhiều thời gian và tài nguyên vì phải liệt kê tất cả các luật có d items (d=1,2,3,…). Tổng số luật có thể kết xuất được từ một tập hợp có d items là R được tính như sau:

$$R=3^d-2^{d+1} \quad \text{(CT-II-22)}$$

- Một cách tiếp cận khác được sử dụng rất nhiều trong các thuật toán phân tích là chia vấn đề giải quyết ở trên ra thành 2 vấn đề con như sau:

(1) Tìm các itemset có tần suất xuất hiện cao (frequent itemsets):

Mục đích của phần này là tìm tất cả các itemset có support lớn hơn hoặc bằng minisup.

(2) Sinh luật:

Mục đích của phần này là sinh tất cả các luật có confidence cao từ các frequent itemset

II.4.7.3. Sinh các frequent itemset bằng nguyên lý "biết trước":

Việc tìm các frequent itemsets bằng nguyên lý biết trước được tiến hành qua từng bước. Bước thứ i (i>=2) sẽ được sử dụng kết quả của các bước trước đó (bước i-1) để loại bỏ những infrequent itemsets.

Bước khởi tạo (i=1):

- Liệt kê các tập hợp gồm 1 item được gọi là 1-itemsets.
- Tính support count cho từng 1-itemset.
- Loại bỏ các 1-itemsets không thỏa minisup.
- Kết quả cuối cùng của bước này là các 1-itemsets thỏa điều kiện minisup.

Bước i=2:

- Xuất phát từ kết quả của bước thứ i-1
- Liệt kê các tập 2-itemsets dựa trên các tập 1-itemsets là kết quả của bước 1.
- Tính support count cho tất cả các 2-itemsets.
- Loại bỏ các 2-itemsets không thỏa điều kiện minisup.
- Kết quả cuối cùng của bước này là các 2-itemsets thỏa điều kiện minisup.

Bước thứ i>=3:

- Lập lại giống như bước 2 cho đến khi nào không tìm được kết quả là các k-itemsets (với k là tổng số items) hoặc không tìm được itemset nào thỏa điều kiện minisup.

II.4.7.4. Sinh các itemsets ứng viên và cắt nhánh:

Gọi $X=\{i_1, i_2,\ldots,i_k\}$ là một candidates mới gồm k phần tử cần xét. Thuật toán này cần kiểm tra tất cả các tập con gồm k-1 phần tử của $X-\{i_j\}$ với j=1,2,…,k. Nếu tồn tại một tập hợp con của nó không thỏa điều kiện minisup thì tập X sẽ bị loại bỏ ngay lặp tức.

Có rất nhiều cách sinh ra các candidate itemset. Một thuật toán hiệu quả phải đảm bảo các yêu cầu sau:

(1) Không sinh ra các itemsets không cần thiết. Một itemsets được gọi là không cần thiết được sinh ra nếu tồn tại ít nhất 1 tập con của nó không thỏa điều kiện minisup.

(2) Phải đảm bảo sinh ra tất cả các itemsets cần thiết.

(3) Đảm bảo không lặp lại các itemsets.

Một số phương pháp được xây dựng đảm bảo các yêu cầu trên:

➢ *Phương pháp "brute-force":*

- Sinh tất cả các candidates có k phần tử.
- Loại bỏ các candidates không cần thiết.

➢ *Phương pháp "$F_{k-1}xF_1$":*

- Sinh tất cả các candidates có k phần tử từ kết quả của bước thứ k-1 và kết quả của bước 1.

- Loại bỏ các candidates không thỏa minisup.

➢ *Phương pháp "$F_{k-1}xF_{k-1}$":*

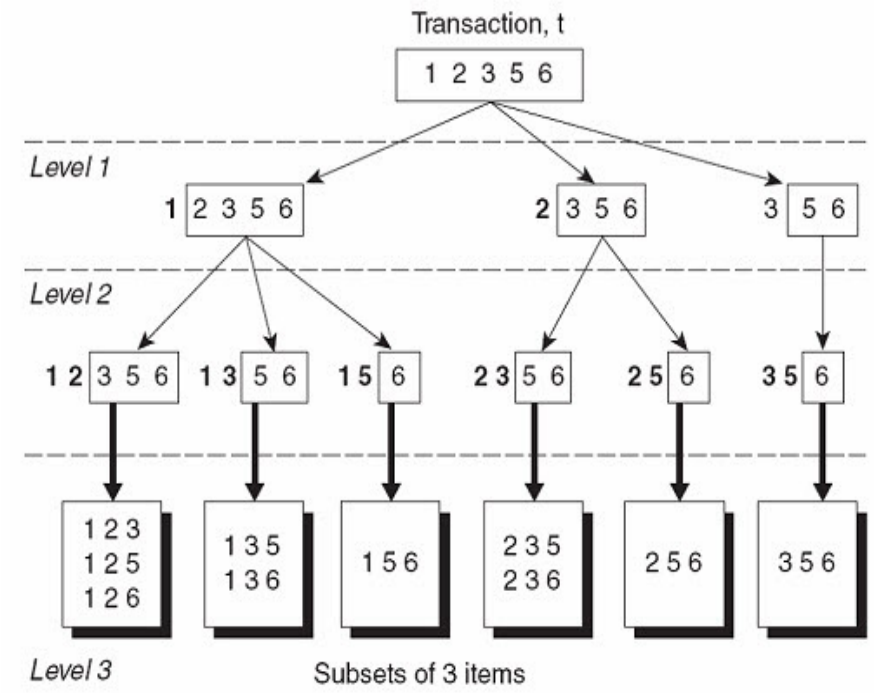
- Sử dụng kết quả của bước lặp thứ k-1.
- Trộn từng cặp itemsets có k-2 phần tử đầu giống nhau.
- Loại bỏ các candidates không thỏa minisup.

II.4.7.5. Đếm support:

Đếm support count là quá trình xác định tần suất xuất hiện của mỗi candidate itemset. Một cách làm đơn giản là so sánh từng candidate itemset với từng giao dịch trong tập hợp các giao dịnh ban đầu. Cách làm này đòi hỏi tốn nhiều thời gian cho việc so sánh.

Một phương pháp được dùng để đếm các support của các candidate itemsets hữu hiệu là sử dụng cây băm. Trong giải thuật biết trước, các candidate itemset được chia thành nhiều nhóm khác nhau và được lưu trữ trong một cây có tên là cây băm (Hash tree). Trong suốt quá trình đếm support, các itemsets được chứa trong mỗi giao dịch được lưu vào trong các ô phù hợp. Với cách làm này, chỉ cần so sánh các candidate itemsets nằm cùng một ô.

Ví dụ: Xét một giao dịch chứa 5 item được đánh số là 1,2,3,5 và 6.



Hình II-32. Minh họa cách liệt kê các 3-itemsets từ giao dịch {1,2,3,5,6}

II.4.7.6. Sinh tập luật kết hợp:

Sau khi có được tập hợp các item thỏa điều kiện minisup (frequent itemset), chúng ta sẽ đi xây dựng tập hợp này thành các luật cụ thể. Mỗi tập hợp frequent itemset có thể sinh tối đa 2^k- 2 luật kết hợp. Một luật kết hợp được sinh ra bằng cách phân tập frequent itemset Y ra thành 2 tập con rời nhau (X và Y-X) sau cho X->(Y-X) thỏa điều kiện miniconf.

a) Cắt nhánh dựa vào độ tin cậy:

Định lý: Nếu một luật kết hợp có dạng X->Y-X không thỏa điều kiện miniconf thì tất cả các luật kết hợp có dạng X'->Y-X' với X là tập con của X đều không thỏa minisup.

b) Sinh luật kết hợp trong thuật toán biết trước:

Đầu tiên, tất cả các luật có độ tin cậy cao phải là luật chỉ có một item ở vế bên phải (rule consequence). Các luật này phải được sinh ra trước sau đó được sử dụng để sinh ra các luật ứng viên (candidate rules) tiếp theo. Một luật mới được sinh ra bằng các luật ban đầu ở 2 vế. Vế trái thì lấy phần giống nhau (phép giao), còn vế phải thì lấy cả 2 (phép hợp).

II.4.7.7. Các loại bài toán:

Bài toán 1: Tìm tất cả các tập mục mà có độ hỗ trợ lớn hơn độ hỗ trợ tối thiểu do người dùng xác định. Các tập mục thoả mãn độ hỗ trợ tối thiểu được gọi là các tập mục phổ biến.

Bài toán 2: Dùng các tập mục phổ biến để sinh ra các luật mong muốn. Ý tưởng chung là nếu gọi ABCD và AB là các tập mục phổ biến, thì chúng ta có thể xác định luật nếu AB => CD giữ lại với tỷ lệ độ tin cậy:

$$conf = \frac{\sup(ABCD)}{\sup(AB)} \qquad \text{(CT-II-23)}$$

Nếu confidence≥ miniconf thì luật được giữ lại (luật này sẽ thoả mãn độ hỗ trợ tối thiểu vì ABCD là phổ biến).

II.4.7.8. Các tính chất liên quan đối với tập mục phổ biến

Tính chất 1 (Độ hỗ trợ của tập con):

Với A và B là tập các mục, nếu $A \subseteq B$ thì $\sup(A) \geq \sup(B)$.

Điều này là rõ ràng vì tất cả các giao tác của D hỗ trợ B thì cũng hỗ trợ A.

Tính chất 2:

Một tập chứa một tập không phổ biến thì cũng là tập không phổ biến.

Nếu *một mục trong B không có độ hỗ trợ tối thiểu* trên D nghĩa là $\sup(B) <$ minisup thì một tập con A của B sẽ không phải là một tập phổ biến vì $\sup(B) \leq \sup(A) <$ minisup (theo tính chất 1).

Tính chất 3: Các tập con của tập phổ biến cũng là tập phổ biến.

Nếu mục B là mục phổ biến trên D, nghĩa là $\sup(B) \geq$ minisup thì mọi tập con A của B là tập phổ biến trên D vì $\sup(A) \geq \sup(B) >$ minisup.

II.4.7.9. Các tính chất liên quan đối với luật kết hợp:

Tính chất 1:(Không hợp các luật kết hợp)

Nếu có $X \rightarrow Z$ và $Y \rightarrow Z$ trong D thì không nhất thiết $X \cup Y \rightarrow Z$ là đúng.

Xét trường hợp $X \cap Z = \varnothing$ và các tác vụ trong D hỗ trợ Z nếu và chỉ nếu chúng hỗ trợ mỗi X hoặc Y, khi đó luật $X \cup Y \rightarrow Z$ có độ hỗ trợ 0%.

Tương tự : $X \rightarrow Y \wedge X \rightarrow Z \Rightarrow X \rightarrow Y \cup Z$

Tính chất 2:(Không tách luật)

Nếu $X \cup Y \rightarrow Z$ thì $X \rightarrow Z$ và $Y \rightarrow Z$ chưa chắc xảy ra.

Tính chất 3: (Các luật kết hợp không có tính bắc cầu)

Nếu X→Y và Y→Z, chúng ta không thể suy ra X→Z.

Tính chất 4:

Nếu A→(L - A) không thoả mãn độ tin cậy cực tiểu thì luật B →(L -B) cũng không thoả mãn, với các tập mục L,A,B và $B \subseteq A \subset L$.

II.4.7.10. Các thuật toán xây dựng luật kết hợp:

a) Thuật toán Apriori:

Giới thiệu bài toán:

Bài toán được phát biểu: Tìm t có độ hỗ trợ s thỏa mãn $s \geq s_0$ và độ tin cậy c $\geq c_0$ (*s_0, c_0 là hai ngưỡng do người dùng xác định* và s_0=minisupp, c_0 =miniconf) . Ký hiệu:

- L_k tập các tập k - mục phổ biến,
- C_k tập các tập k-mục ứng cử (cả hai tập có: tập mục và độ hỗ trợ).

Bài toán đặt ra là:

1) Tìm tất cả các tập mục phổ biến với minisup nào đó.

2) Sử dụng các tập mục phổ biến để sinh ra các luật kết hợp với độ tin cậy miniconf nào đó.

Quá trình thực hiện (duyệt):

(1) Thực hiện nhiều lần duyệt lặp đi lặp lại, trong đó tập (k-1) - mục được sử dụng cho việc tìm tập k-mục. Lần thứ nhất tìm tất cả các độ hỗ trợ của các mục, xác định mục phổ biến (mục thoả mãn độ hỗ trợ cực tiểu-minisup). Giả sử tìm được L_1-mục phổ biến.

(2) Các lần duyệt còn lại: Bắt đầu kết quả tìm được bước trước nó, sử dụng các tập mục mẫu (L_1) sinh ra các tập mục phổ biến tiềm năng (ứng cử)(giả sử L_2), tìm độ hỗ trợ thực sự. Mỗi lần duyệt ta phải xác định tập mục mẫu cho lần duyệt tiếp theo.

(3) Thực hiện lặp để tìm L_3, ..., L_k cho đến khi không tìm thấy tập mục phổ biến nào nữa.

❖ Chú ý:

Ứng dụng L_{k-1} để tìm L_k bao gồm hai bước chính:

(1) Bước kết nối: tìm L_k là tập k-mục tương ứng được sinh ra bởi việc kết nối L_{k-1} với chính nó cho kết quả là C_k. Giả sử L_1, L_2 thuộc L_{k-1}. Ký hiệu L_i^j là mục thứ j trong L_i. Điều kiện là các tập mục hay các mục trong *giao dịch có thứ tự*.

(2) Bước kết nối như sau: Các thành phần L_{k-1} kết nối (nếu có chung k-2-mục đầu tiên) tức là:$(L_1[1]=L_2[1]) \cap (L_1[2]=L_2[2]) \cap ... \cap (L_1[k-2]=L_2[k-2]) \cap (L_1[k-1]=L_2[k-1])$.

(3) Bước tỉa: C_k là tập chứa L_k (có thể là tập phổ biến hoặc không) nhưng tất cả tập k-mục phổ biến được chứa trong C_k. Bước này, duyệt lần hai CSDL để tính độ hỗ trợ cho mỗi ứng cử trong C_k sẽ nhận được L_k

Thuật toán Apriori

Các kí hiệu:

L_k: Tập các k-mục phổ biến (large k-itemset) (tức tập các itemset có sup tối thiểu và có lực lượng bằng k).

Mỗi phần tử của tập này có 2 trường: itemset và support-count.

C_k: Tập các candidate k-itemset (tập các tập k-mục ứng cử viên). Mỗi phần tử trong tập này cũng có 2 trường itemset và support-count.

Nội dung thuật toán Apriori được trình bày như sau:

```
1   Input: Tập các giao dịch D, ngưỡng support tối thiểu minisup
2   Output: L- tập mục phổ biến trong D
3   Method:
4   L1={large 1-itemset (tập 1- mục phổ biến)} //tìm tất cả các tập mục phổ biến: nhận được L1
5   for (k=2; Lk-1 ≠ ∅; k++) do
6   begin
7       Ck=apriori-gen(Lk-1); //sinh ra tập ứng cử viên từ Lk-1
8       for (mỗi một giao dịch T∈ D) do
9       begin
10          CT = subset(Ck, T); //lấy tập con của T là ứng cử viên trong Ck
11          for (mỗi một ứng cử viên c∈ CT) do
12              c.count++; //tăng bộ đếm tần xuất 1 đơn vị
13          end;
14          Lk = {c ∈ Ck| c.count ≥ minisup}
15  end;
16  return ∪kLk
```

+) Trong thuật toán này, giai đoạn đầu đơn giản chỉ là việc đếm support cho các mục(item). Để xác định tập 1-mục phổ biến (L_1), người ta chỉ giữ lại các mục (item) mà sup của nó lớn hơn hoặc bằng *minisup.*

+) Trong các giai đoạn thứ k sau đó (k>1), mỗi giai đoạn gồm có 2 pha. Trước hết các large(k-1)-itemset (tập k-1- mục phổ biến) trong tập L_{k-1}được sử dụng để sinh ra các candidate itemset (tập ứng cử viên) C_k, bằng cách thực hiện hàm Apriori_gen.

+) Tiếp theo CSDL D sẽ được quét để tính support cho mỗi ứng viên trong C_k. Để việc đếm được nhanh, cần phải có một giải pháp hiệu quả để xác định các ứng viên trong C_k là có mặt trong một giao dịch T cho trước.

Vấn đề sinh tập candidate (tập ứng cử) của Apriori – Hàm Apriori_gen:

```
1   Input: tập mục phổ biến Lk-1 có kích thước k-1
2   Output: tập ứng cử viên Ck
3   Method:
4   function apriori-gen(Lk-1: tập mục phổ biến có kích thước k-1)
5   Begin
6       For (mỗi L1 ∈ Lk-1) doa
```

```
7            For (mỗi L2 ∈ Lk-1) do
8               If ((L1[1]=L2[1]) ∩ (L1[2]=L2[2]) ∩ ... ∩ (L1[k-2]=L2[k-2])
                ∩ (L1[k-1]=L2[k-1])) then
9                     c = L1 ⊕ L2; // kết nối L1 với L2 sinh ra ứng cử viên c
10              If has_infrequent_subset(c, Lk-1) then //có tập con khổ phổ
                biến của c trong Lk-1
11                    remove (c) // bước tỉa (xoá ứng cử viên c)
12              else Ck = Ck ∪ {c}; kết tập c vào Ck
13    end;
14    Return Ck;
15    End;
```

Hàm Apriori_gen với đối số là L_{k-1}(tập các large(k-1)-itemset) sẽ cho lại kết quả là một superset, tập của tất cả các large k – itemset. Sơ đồ sau là thuật toán cho hàm này.

Với nội dung trên, ta thấy hàm này có 2 bước:

- Bước nối (join step)
- Bước cắt tỉa (prune step)

Hàm Subset (trong bước tỉa) Các tập ứng cử viên C_k được lưu trữ trong một cây băm:

+) Nút lá của cây băm chứa danh sách một tập mục và đếm

+) Các nút trong chứa ở trong bảng băm

-> Hàm subset: tìm tất cả các ứng cử viên được chứa trong giao tác.

Hàm kiểm tra tập con k-1 mục của ứng cử viên k-mục không là tập phổ biến:

```
1    function has_infrequent_subset(c: ứng cử viên k-mục; Lk-1 tập phổ biến k-1
mục)
2    Begin
          //sử dụng tập mục phổ biến trước
3         For (mỗi tập con k-1 mục s của c) do
4              If s ∈ Lk-1 then return TRUE;
5    End;
```

b) Một số biến thể của thuật toán Apriori:

(1) Giải thuật AprioriTID là phần mở rộng theo hướng tiếp cận cơ bản của giải thuật Apriori. Thay vì dựa vào cơ sở dữ liệu thô giải thuật AprioriTID biểu diễn bên trong mỗi thao tác bởi các ứng viên hiện hành.

```
1     L1= {Large 1-itemset};
2     C'1 = Database D;
3     for (k=2; Lk-1 ≠ ∅ ; k++) do
4     Begin
5           Ck = apriori_gen(Lk-1);
6           C'k = ∅;
```

```
7           for tất cả t ∈ C'k-1 do
8           begin
                // xác định tập ứng viên trong Ck chứa trong giao dịch với
định
                //danh t. Tid (Transaction Code)
9               Ct = {c ∈ Ck | (c-c[k]) ∈ t.Set_of_ItemSets ^ (c-c[k-1]∈
            t.Set_of_ItemSets}
10          for những ứng viên c ∈ Ct do c.count ++;
11              if (Ct≠∅) then C'k+= < t.Tid, Ct >
12          end
13              Lk = {c ∈Ck | c.count ≥ minsup};
14      End
15      return = ∪kLk;
```

Thuật toán này cũng sử dụng hàm apriori_gen để sinh ra các tập ứng cử viên cho mỗi giai đoạn. Nhưng thuật toán này không dùng CSDL D để đếm các support với các giai đoạn k > 1 mà sử dụng tập C'_k. Mỗi phần tử của C'_k có dạng <Tid, $\{X_k\}$>, trong đó mỗi X_k là một tập phổ biến k_itemset tiềm năng trong giao dịch Tid. Khi k = 1, C'_k tương ứng với D, trong đó mỗi item i được coi là một itemset {i}. Với k>1, C'_k được sinh ra bởi C'_k+= < t.Tid, C_t >. Phần tử của C'_k tương ứng với giao dịch t là <t.Tid, {c∈ | c chứa trong t}>. Nếu một giao dịch không chứa bất kỳ tập ứngviên k_itemset nào thì C'_k sẽ không có một điểm vào nào cho giao dịch này. Do đó, số lượng điểm vào trong C'_k có thể nhỏ hơn số giao dịch trong CSDL, đặc biệt với k lớn. Hơn nữa, với các giá trị k khá lớn, mỗi điểm vào có thể nhỏ hơn giao dịch tương ứng vì một số ứng viên đã được chứa trong giao dịch. Tuy nhiên, với các giá trị k nhỏ, mỗi điểm vào có thể lớn hơn giao dịch tương ứng vì một điểm vào trong C'_k bao gồm tất cả các ứng viên k_itemset được chứa trong giao dịch.

(2) Giải thuật AprioriHybrid kết hợp cả hai hướng tiếp cận trên. Ngoài ra còn có một số các giải thuật tựa Apriori(TID), chúng được định hướng để cài trực tiếp trong SQL.

(3) Giải thuật DIC là một biến thể khác nữa của giải thuật Apriori. Giải thuật DIC làm giảm đi khoảng phân biệt nghiêm ngặt giữa việc đếm và việc phát sinh các ứng viên. Bất kỳ ứng viên nào tới được ngưỡng minsupp, thì giải thuật DIC bắt đầu phát sinh thêm các ứng viên dựa vào nó. Để thực hiện điều này giải thuật DIC dùng một prefix-tree (cây tiền tố). Ngược với hashtree, mỗi nút (nút lá hoặc nút trong) của prefix-tree được gán một ứng viên xác định trong tập phổ biến. Cách sử dụng cũng ngược với hashtree, bất cứ khi nào tới được một nút ta có thể khẳng định rằng tập *item* đã kết hợp với nút này trong giao tác đó. Hơn nữa, việc xác định độ hỗ trợ và phát sinh ứng viên khớp nhau sẽ làm giảm đi số lần duyệt cơ sở dữ liệu.

II.4.7.11. Ứng dụng:

- Được sử dụng trong việc giải quyết các vấn đề tư vấn sắp xếp danh mục các hàng hóa mà khách hàng thường xuyên lựa chọn và liên quan với nhau dựa trên các tập dữ liệu ban đầu.

- Các website sử dụng luật kết hợp như một lựa chọn theo nhu cầu của khách hàng, phục vụ nhanh chóng trong việc lựa chọn thông tin và hiển thị thông tin theo nhu cầu khách hàng.
- Một số khó khăn trong việc xây dựng luật kết hợp chủ yếu là nằm ở khâu thu thập dữ liệu, để dữ liệu chính xác thì cần một lượng dữ liệu lớn để phân tích và được cập nhật thường xuyên.

Chương III : **NỘI DUNG NGHIÊN CỨU**

III.1. NGHIÊN CỨU VỀ PHẦN MỀM KHAI PHÁ DỮ LIỆU

Trong thực tế, có rất nhiều chương trình được sử dụng nhằm phục vụ cho công việc khai phá dữ liệu như: Mathlab, Excel, minitab, SQL Server,… là các công cụ khai phá dữ liệu có bản quyền. Bên cạnh đó, cũng không thể không nhắc đến các công cụ được phát triển miễn phí như: weka, Rapid miner, Tanagra,… góp phần phục vụ nhu cầu tìm hiểu về khai phá dữ liệu. Trong quá trình nghiên cứu, chúng ta sẽ đi sâu vào nghiên cứu về phần mềm Tanagra.

III.1.1. Giới thiệu Tanagra:

Tanagra là một phần mềm khai phá dữ liệu miễn phí, phục vụ cho công việc học tập và nghiên cứu. Nó là công cụ phục vụ cho công việc khai phá dữ liệu từ phân tích dữ liệu thăm dò (exploratory), học thống kê (statistical learning), máy học (machine learning) và vùng cơ sở dữ liệu (databases area).

Dự án về Tanagra là sự kế thừa của Sipina mà việc thực hiện các thuật toán học có giám sát, đặc biệt là xây dựng các tương tác và hình ảnh của cây quyết định. Tanagra mạnh hơn, nó chứa một số phương pháp học có giám sát, như: phân nhóm, phân tích giai thừa, thống kê tham số và phi tham số, association rules, lựa chọn tính năng (feature selection) và các thuật toán xây dựng (construction algorithms).

Tác giả của Tanagra và Sipina là Ricco Rakomalala – Phó giáo sư Khoa học máy tính của trường Đại học Lyon 2, Pháp. Ông là thành viên của ERIC (Equipe de recherche en Ingénierie des Connaissances – Nhóm nghiên cứu kỷ thuật kiến trúc) do đại học Lyon tổ chức nhằm nghiên cứu về máy tính và các ứng dụng của nó.

Tanagra là một dự án mã nguồn mở, như mọi nhà nghiên cứu có thể truy cập vào mã nguồn, và thêm vào các thuật toán riêng của mình, nhưng phải đồng ý và tuân thủ giấy phép phân phối phần mềm.

Mục đích chính của dự án Tanagra là để cho các nhà nghiên cứu và sinh viên dễ sử dụng phần mềm khai phá dữ liệu, phù hợp với các chỉ tiêu hiện tại của phát triển phần mềm trong miền này (đặc biệt trong việc thiết kế GUI và cách sử dụng nó), và cho việc phân tích thiết thực và tổng hợp dữ liệu.

Mục đích thứ hai của Tanagra là để đề xuất với các nhà nghiên cứu một kiến trúc cho phép họ dễ dàng thêm các phương pháp khai thác dữ liệu của riêng họ, để so sánh kết quả của họ. Nhiều tác động của Tanagra như là nền tản thử nghiệm nhằm để cho họ đi đến những thiết yếu của công việc của họ, và để họ có thể đối phó với các phần khó khăn trong việc quản lý dữ liệu.

Mục đích thứ ba và cuối cùng, trong việc mới làm quen với hướng phát triển, bao gồm trong việc khuyếch tán một phương pháp có thể để xây dựng các loại phần mềm. Họ cần tận dụng lợi thế của truy cập miễn phí mã nguồn, để tìm các sắp xếp của phần mềm này được xây dựng, các vấn đề cần tránh, các bước chính của dự án, có các công cụ và thư viện mã để sử dụng. Bằng cách này, Tanagra có thể được coi như là một công cụ nghiệp vụ sư phạm cho việc học tập kỹ thuật lập trình.

Tanagra không bao gồm tất cả sức mạnh của các phần mềm thương mại trong lĩnh vực này như: một tập hợp nhiều nguồn dữ liệu, truy cập trực tiếp cơ sở dữ liệu và kho dữ liệu (datawarehouses), làm sạch dữ liệu, tương tác sử dụng,...

Tanagra có khả năng tương tác với Microsoft Office và Open Office.

III.1.2. Tìm hiểu về Tanagra:

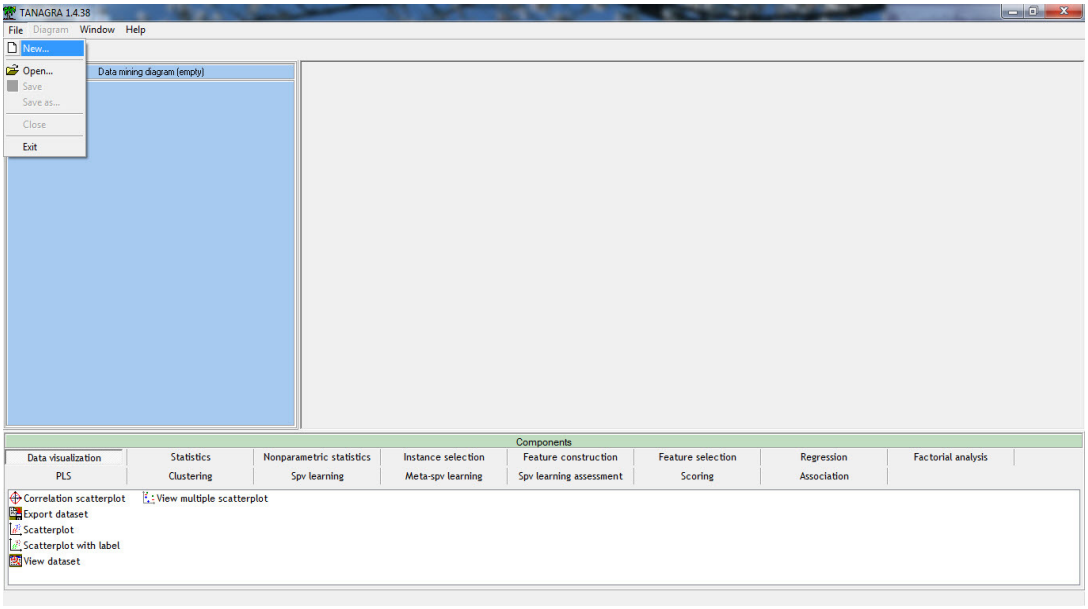
III.1.2.1. Mở một ứng dụng và tải tập tin:

Có 2 các để mở chương trình và tải tập tin:

(1) Cách 1: Mở trực tiếp từ chương trình thực thi:

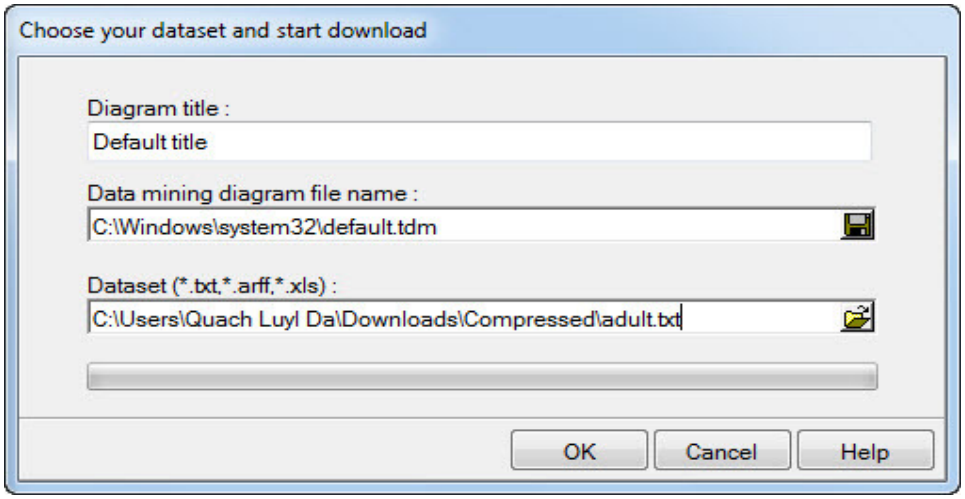
1- Khởi động chương trình Tanagra.

2- Trên menu chính, chọn File/new để mở một ứng dụng mới.



Hình III-1. Mở ứng dụng mới

3- Nhập tiêu đề cho biểu đồ

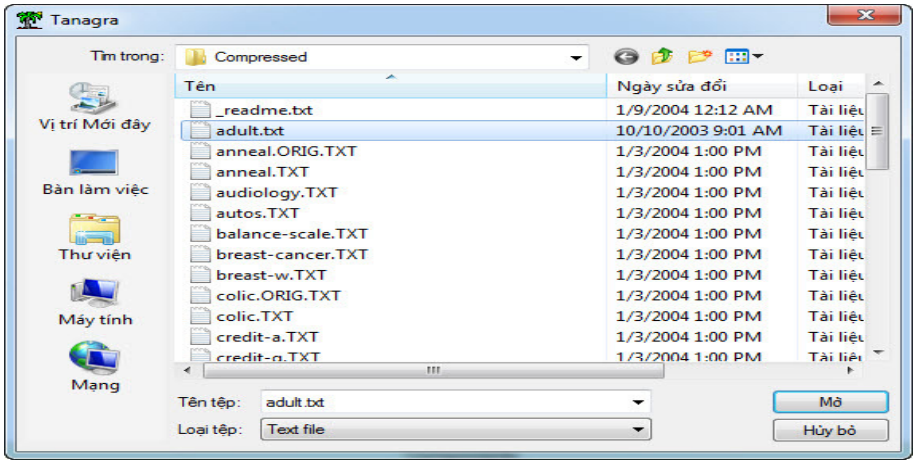


Hình III-2. Cửa sổ nhập tiêu đề

Trong đó:

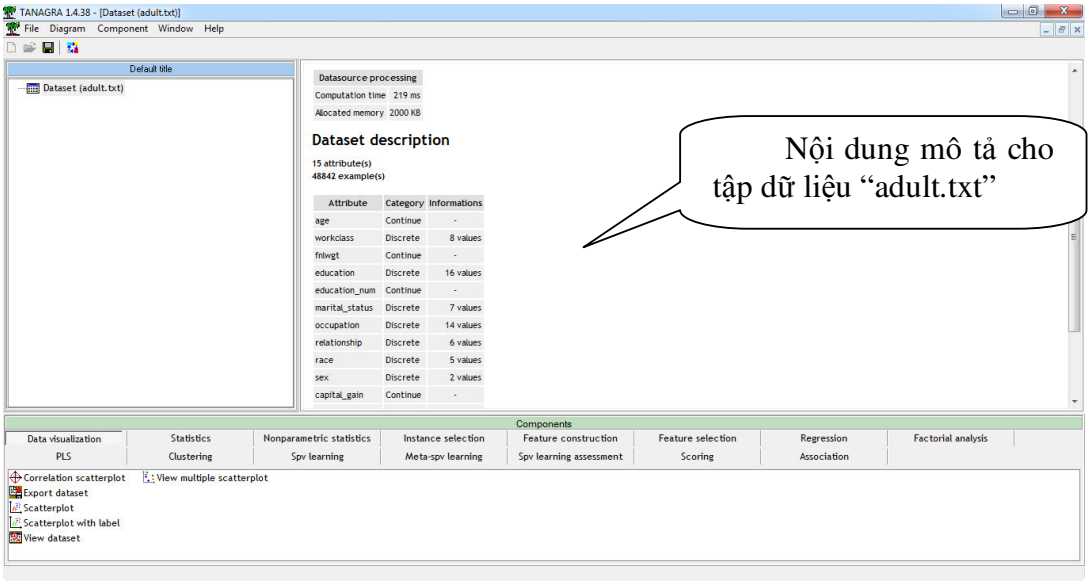
- Diagram title: Tên của sơ đồ.
- Data mining diagram file name: Tên tập tin sơ đồ khai khoáng dữ liệu, đây là tên tập tin có liên quan đến tập dữ liệu cần khai khoáng. Nó sẽ giúp tiết kiệm thời gian thực hiện. Ở đây ta chọn tập sơ đồ là mặc định của chương trình ("Default.tdm"). Chọn tên tập tin và click vào nút  .
- Dataset: Đường dẫn tập dữ liệu cần khai phá. Tại đây, Tanagra hỗ trợ cho việc phân tích dữ liệu của các định dạng như:
  - Dữ liệu kiểu Text(*.txt).
  - Dữ liệu của weka(*.arff).
  - Dữ liệu của Excel (*.xls).

Click vào nút  để mở tập đường dẫn:



Hình III-3. Cửa sở mở tập tin

Và sau đó chọn Open. Một sơ đồ mới được tạo ra dựa vào tập tin "adult.txt". Bạn có thể xem mô tả nội dung trong khung bên phải.



Hình III-4. Cửa sổ hiển thị tập tin

Một số thông tin mô tả về nội dung:

a) Datasource processing: Xử lý dữ liệu nguồn. Nơi cho biết các thông tin về nguồn dữ liệu được sử lý như: Thời gian tính toán (Computation time), bộ nhớ được cấp phát (allocated memory).

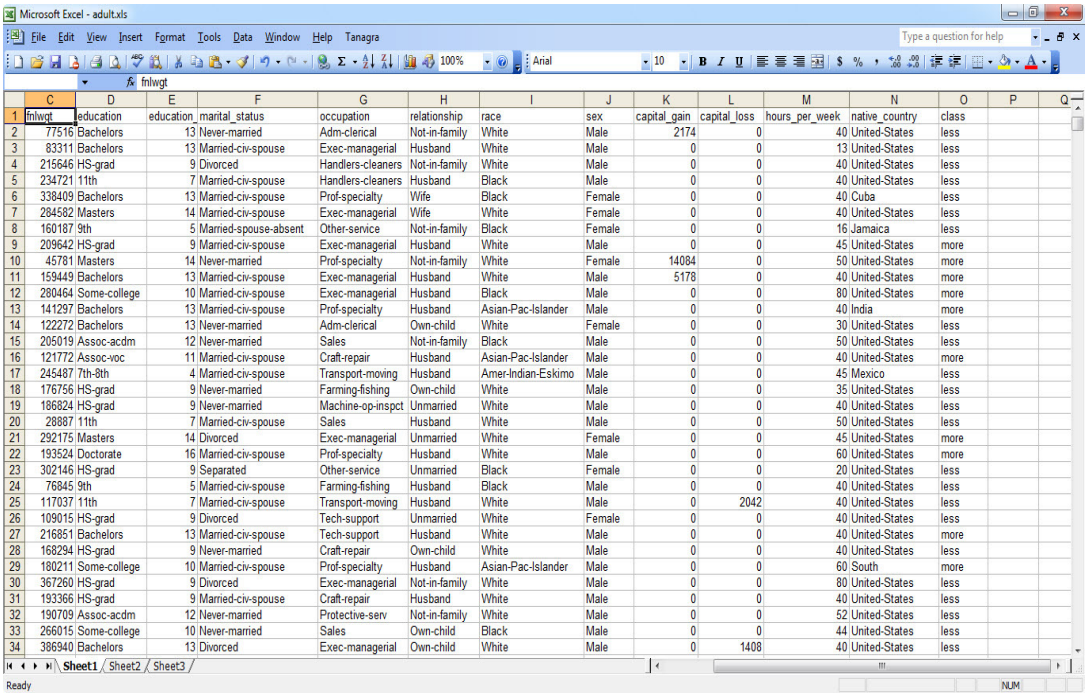
b) Dataset description: Tập dữ liệu mô tả. Mô tả về tập dữ liệu nguồn, bao gồm các thông tin như:

- Attribute: Thuộc tính của dữ liệu. Ví dụ: Age, workclass,…
- Category: Phạm trù hay kiểu của dữ liệu thuộc tính. Ví dụ: Continue, discrete,…
- Infomations:Thông tin về thuộc tính của dữ liệu hay đếm số mẫu tin có cùng thuộc tính. Ví dụ: thuộc tính workclass có kiểu là continue và có 8 giá trị tương ứng (Values).

c) Default tilte: Tên tập dữ liệu, khung này bao gồm các quá trình phân tích sẽ được thực hiện tại đây.

(2) Cách 2: Mở gián tiếp thông qua Excel:

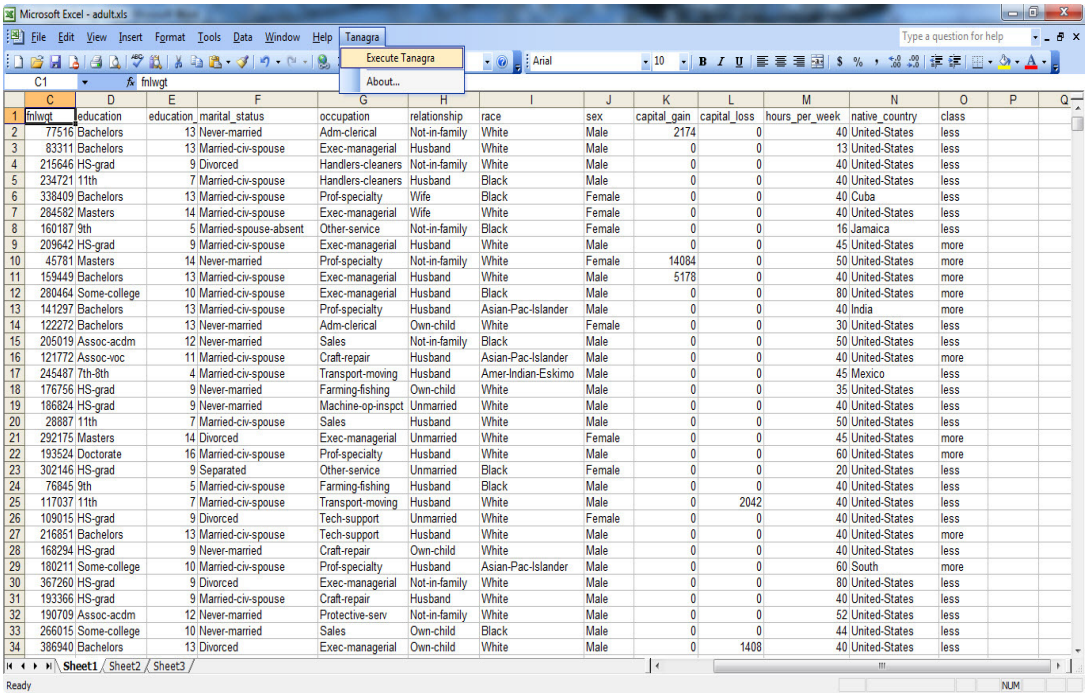
1- Mở tập dữ liệu Excel cần khai phá.



	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	fnlwgt	education	education_	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	class		
2	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	less		
3	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	less		
4	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	less		
5	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	less		
6	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	less		
7	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	less		
8	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	less		
9	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	United-States	more		
10	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States	more		
11	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40	United-States	more		
12	280464	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Black	Male	0	0	80	United-States	more		
13	141297	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	40	India	more		
14	122272	Bachelors	13	Never-married	Adm-clerical	Own-child	White	Female	0	0	30	United-States	less		
15	205019	Assoc-acdm	12	Never-married	Sales	Not-in-family	Black	Male	0	0	50	United-States	less		
16	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	0	0	40	United-States	more		
17	245487	7th-8th	4	Married-civ-spouse	Transport-moving	Husband	Amer-Indian-Eskimo	Male	0	0	45	Mexico	less		
18	176756	HS-grad	9	Never-married	Farming-fishing	Own-child	White	Male	0	0	35	United-States	less		
19	186824	HS-grad	9	Never-married	Machine-op-inspct	Unmarried	White	Male	0	0	40	United-States	less		
20	28887	11th	7	Married-civ-spouse	Sales	Husband	White	Male	0	0	50	United-States	less		
21	292175	Masters	14	Divorced	Exec-managerial	Unmarried	White	Female	0	0	45	United-States	more		
22	193524	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	60	United-States	more		
23	302146	HS-grad	9	Separated	Other-service	Unmarried	Black	Female	0	0	20	United-States	less		
24	76845	9th	5	Married-civ-spouse	Farming-fishing	Husband	Black	Male	0	0	40	United-States	less		
25	117037	11th	7	Married-civ-spouse	Transport-moving	Husband	White	Male	0	2042	40	United-States	less		
26	109015	HS-grad	9	Divorced	Tech-support	Unmarried	White	Female	0	0	40	United-States	less		
27	216851	Bachelors	13	Married-civ-spouse	Tech-support	Husband	White	Male	0	0	40	United-States	more		
28	168294	HS-grad	9	Never-married	Craft-repair	Own-child	White	Male	0	0	40	United-States	less		
29	180211	Some-college	10	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	60	South	more		
30	367260	HS-grad	9	Divorced	Exec-managerial	Not-in-family	White	Male	0	0	80	United-States	less		
31	193366	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	40	United-States	less		
32	190709	Assoc-acdm	12	Never-married	Protective-serv	Not-in-family	White	Male	0	0	52	United-States	less		
33	266015	Some-college	10	Never-married	Sales	Own-child	Black	Male	0	0	44	United-States	less		
34	386940	Bachelors	13	Divorced	Exec-managerial	Own-child	White	Male	0	1408	40	United-States	less		

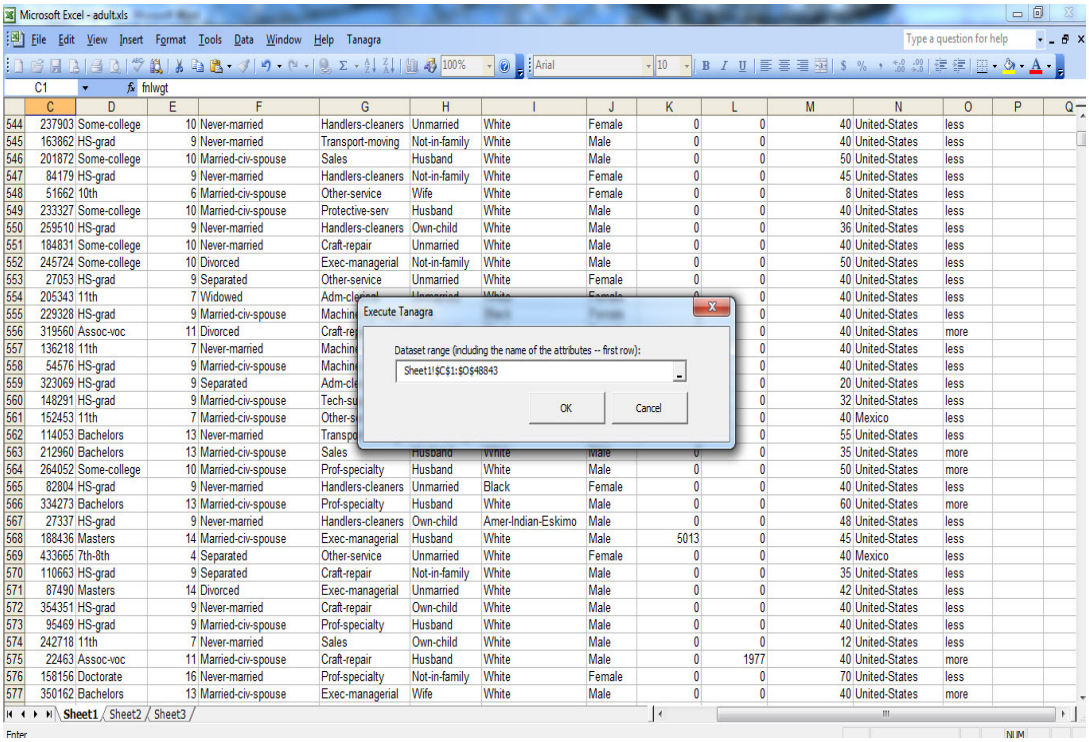
Hình III-5. Cửa sổ làm việc của Excel

2- Trên thanh menu chính, click chọn Tanagra\Excute tanagra:



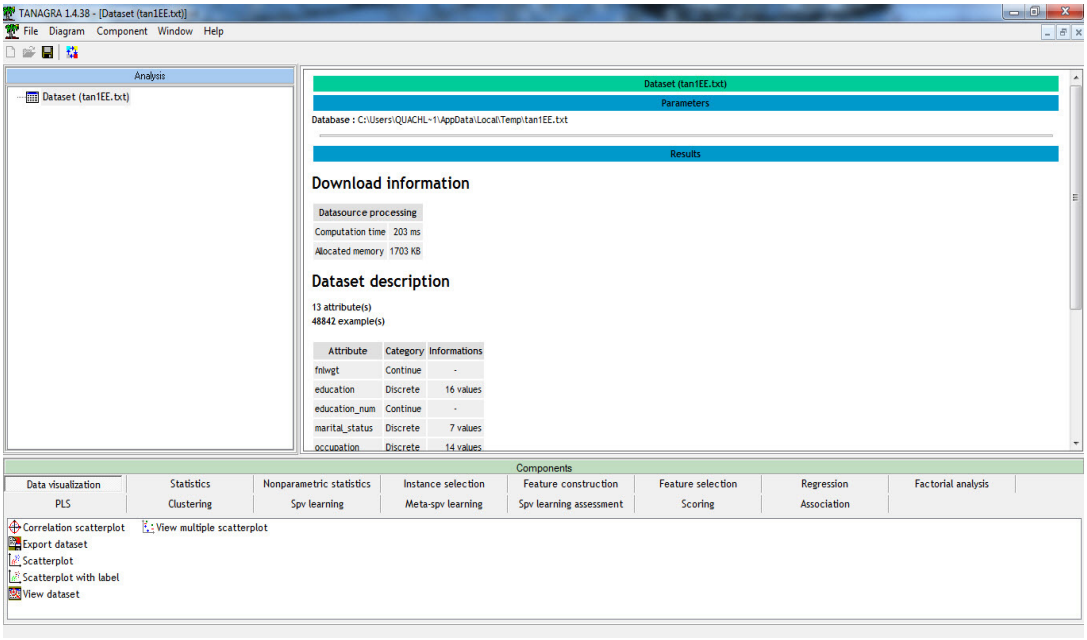
Hình III-6. Cách chuyển sang Tanagra trên Excel

3- Chọn các cột dữ liệu phục vụ cho quá trình học:



Hình III-7. Chọn dữ liệu trên Excel phục vụ vho quá trình học

4- Sau khi chọn xong, click Ok, thì chương trình Tanagra được mở:



Hình III-8. Cửa sổ chuyển đổi Excel sang Tanagra

Các thông tin được mô tả như cách 1.

III.1.2.2. Xác định giá trị Input và Target:

- Đầu vào (Input): Là những thuộc tính đầu vào phục vụ cho việc xác định mục tiêu, thuộc tính đầu vào có thể là liên tục, rời rạc,…
- Mục tiêu (Target): Được xác định dựa trên những thuộc tính đầu vào.

Ví dụ: Tập dữ liệu về động vật:

Input Target

Tên	Thân nhiệt	Da được phủ bởi	Sinh con	Sống dưới nước	Sống trên không	Có chân	Ngủ đông	Loài (class label)
Người	Máu nóng	Lông	Có	Không	Không	Có	Không	Đ/vật có vú
Con trăn	Máu lạnh	Vảy	Không	Không	Không	Không	Không	Bò sát
Cá hồi	Máu lạnh	Vảy	Không	Có	Không	Không	Không	Cá
Cá voi	Máu nóng	Lông	Có	Có	Không	Không	Không	Đ/vật có vú
Ếch	Máy lạnh	Không	Không	Một nửa	Không	Có	Có	Đ/vật lưỡng cư
Thằn lằn	Máu lạnh	Vảy	Không	Không	Không	Có	Không	Bò sát
Dơi	Máu nóng	Lông	Có	Không	Có	Có	Có	Đ/vật có vú
C/ bồ câu	Máu nóng	Lông vũ	Không	Không	Có	Có	Không	Chim
Mèo	Máu nóng	Lông mao	Có	Không	Không	Có	Không	Đ/vật có vú
Cá mập	Máu lạnh	Vẩy	Có	Có	Không	Không	Không	Cá
Rùa	Máu lạnh	Vẩy	Khong	Một nửa	Không	Có	Không	Bò sát
C/ cánh cụt	Máu nóng	Lông vũ	Không	Một nửa	Không	Có	Không	Chim
Nhím	Máu nóng	L/dạng ống	Có	Không	Không	Có	Có	Đ/vật có vú
Lương	Máu lạnh	Vảy	Không	Có	Không	Không	Không	Cá
Kỳ nhông	Máu lạnh	Không	Không	Một nửa	Không	Có	Có	Đ/vật lưỡng cư

Hình III-9. Tập dữ liệu về động vật có xương sống

III.1.2.3. Các thao tác trên Components:

Trong thanh Components chứa đựng những thành phần phục vụ cho quá trình khai phá dữ liệu như:

(1) Data visualization:

Dữ liệu được nhìn ở khia cạnh trực quan, với các kiểu biểu diễn dữ liệu của khai phá dữ liệu. Với các kiểu trình bày dữ liệu khác nhau như:

- Correlation scatterplot: Biểu đồ dạng tương quan phân tán.
- Export dataset: Xuất dữ liệu sang định dạng phục vụ cho việc báo cáo
- Scatterplot: Biểu đồ dạng phân tán.
- Scatterplot with label: Biểu đồ dạng phân tán với các nhãn đính kèm.
- View dataset: Khung nhìn dữ liệu
- View multiple scatterplot: Khung nhìn biểu đồ đa phân tán được gom nhóm.

(2) Statistics:

Công cụ phục vụ cho việc thống kê tập dữ liệu cần phân tích, với các công cụ như:

- ANOVA Randomized: phân tích phương sai cho thiết kế khối ngẫu nhiên. Phương pháp này so sánh một số mẫu có liên quan
- Bartlett's test: Kiểm tra Bartlett's cho tính đồng nhất của phương sai - k mẫu phân phối thường. So sánh phương sai của một phụ thuộc của cá nhân thuộc các nhóm khác nhau.
- Box's M test: Kiểm tra hộp M là quá trình kiểm tra tĩnh, các kiểm tra giả định phương sai có điều kiện thay đổi. Ví dụ: Các giả thuyết là các ma trận hiệp phương sai là như nhau trong các nhóm được xác định bởi giá trị của biến độc lập riêng biệt.
- Brown-Forsythe's test: Kiểm tra Brown-Forsythe's cho tính thuần nhất của phương sai – K mẫu độc lập. So sánh phương sai của một biến phụ thuộc riêng lẻ đến các nhóm khác.
- Fisher's test: Kiểm tra Fisher cho tính thuần nhất của phương sai-2 phân phối mẫu thường. So sánh phương sai của một biến phụ thuộc riêng lẻ đến 2 nhóm khác.
- Group characterization: so sánh số liệu thống kê để mô tả đặc điểm nhóm được xác định bởi thuộc tính rời rạc. Mục đích của phần này là để hiển thị nếu có sự khác biệt giữa các nhóm theo chỉ số thống kê khác nhau như: trung bình, tỷ lệ, vv
- Group exploration: nhóm thăm dò trực quan, nó là một sự tổng quát của các nhóm đặc tính. Mục đích là để tính toán thống kê mô tả về một nhóm con của tập dữ liệu, các phân nhóm được xác định bởi thuộc tính giá trị điều kiện bằng cách sử dụng các thuộc tính rời rạc.
- Hottelling's T2: so sánh đa biến của hai phương tiện với giả định phương sai có điều kiện không đổi. Các giả thuyết là các vector của các phương tiện của các thuộc tính phụ thuộcnhiều là một trong cùng một thành hai nhóm được xác định bởi giá trị của biến độc lập riêng biệt.

- Hottelling's T2 Heteroscedastic: so sánh đa biến của hai phương tiện với giả định phương sai thực sự phụ thuộc vào một biến ngẫu nhiên khác. Các giả thuyết là các vector của các phương tiện của các thuộc tính phụ thuộc nhiều là một trong cùng một thành hai nhóm được xác định bởi giá trị của biến độc lập riêng biệt.
- Linear correlation: tính toán và kiểm tra tầm quan trọng của các mối tương quan tuyến tính từ cặp của các thuộc tính liên tục: mục tiêu đầu vào, hoặc lựa chọn đầu vào chéo.
- More Univariate: thống kê mô tả chi tiết về các thuộc tính đầu vào liên tục.
- One-way ANOVA: một cách phân tích phương sai: tính toán trung bình của các thuộc tính liên tục mục tiêu theo nhóm được xác định bởi thuộc tính đầu vào .
- One-way MANOVA: một cách phân tích đa biến của phương sai: tính toán các nhóm khác biệt trên một số biến liên tục phụ thuộc đồng thời.
- Partial Correlation: tính toán và kiểm tra tầm quan trọng của sự tương quan một phần từ cặp mục tiêu-đầu vào liên tục kiểm soát các thuộc tính giá trị của biến minh họa.
- Semi-partial Correlation: tính toán và kiểm tra tầm quan trọng của sự tương quan một nửa phần từ cặp mục tiêu-đầu vào liên tục kiểm soát các thuộc tính giá trị của biến minh họa.
- T-Test: cho 2 mẫu test cho giả định bằng phương tiện – Giả định bằng với phương sai. So sánh các trung bình của một biến phụ thuộc của cá nhân trực thuộc nhóm khác nhau.
- T-Test Unequal Variance: cho 2 mẫu test cho giả định bằng phương tiện – Giả định không bằng với phương sai. So sánh các trung bình của một biến phụ thuộc của cá nhân trực thuộc nhóm khác nhau.

(3) Clustering:

- CT: Cây phân cụm. Cây phân cụm đơn nguyên tắc. Nó rất giống với cây hồi quy nhưng có thể xử lý nhiều hơn một lớp thuộc tính liên tục.
- CTP: Cây phân cụm với việc xử lý cắt tỉa để xác định kích thước bên phải của cây. Thành phần này tương tự như cây phân cụm. Một phương pháp cắt tỉa được cài đặt (một phần trong những ví dụ được lựa chọn) được thực hiện để lấy được kích thước phù hợp của cây.
- EM-Clustering: Phân cụm với kỳ vọng tuyệt đối của thuật toán phân cụm. Hỗn hợp của phương pháp Gaussian và các đầu vào là các biến liên tục.
- EM-Selection: Lựa chọn số "tốt nhất" trong cụm với EM-Clustering. Có thể được gắn vào loại chỉ có thành phần.
- HAC: Cụm phân cấp. Thành phần này sử dụng chiến lược tiêu chuẩn cụm chiến lược phân cấp. Lá của cây phân cấp tương ứng với dữ liệu được đưa vào xây dựng (học).
- K-Means: Phân cụm với thuật toán K-Means (Forgy or mcqueen). Thuộc tính được đầu vào là thuộc tính liên tục.

- Kohonen-SOM: Phân cụm với bản đồ tổ chức Kohonen. Thuộc tính đầu vào là thuộc tính liên tục.
- LVQ: Phân cụm giám sát với lượng tử của vec tơ học Kohonen (LVQ1). Thuộc tính mục tiêu riêng biệt và đầu vào là thuộc tính liên tục.
- Neighborhood Graph: Giám sát cụm với đồ thị vùng lân cận. Nó là một phương pháp thực nghiệm đang được phát triển.
- VARCLUS: Phân cụm biến bằng cách sử dụng varclus - Tiếp cận từ trên xuống. Dựa trên các biến tiềm ẩn. Liên tục đầu vào thuộc tính
- VARHCA: Phân cụm biến bằng cách sử dụng HCA – Phân tích cụm phân cấp - Phương pháp tiếp cận theo phương pháp phân tích cụm trên biến tiềm ẩn,thuộc tính đầu vào là liên tục.
- Varkmeans: Phân cụm biến bằng cách sử dụng phương pháp K-Means tiếp cận về biến tiềm ẩn. Thuộc tính đầu vào là liên tục.

(4) Spv learning:

- Binary logistic regresion: Hồi quy logic nhị phân, mục tiêu phải được thuộc tính nhị phân riêng biệt, đầu vào là giá trị liên tục.
- C4.5: Một thuật toán của cây quyết định.
- C-PLS: PLS cho phân loại (Mục tiêu nhị phân, đầu vào liên tục hoặc nhị phân).
- CS-CRT: Thuật toán cây phân loại phân biệt giá trị. Phiên bản của xử lý CART sai phân loại trong ma trận giá trị.
- CS-MC4: chi phí thuật toán cây quyết định nhạy cảm. Phiên bản này sử dụng dự toán xác suất làm mịn (một sự tổng quát của dự Laplace). Nó sẽgiảm thiểu sự mất mát bằng cách sử dụng ma trận phân loại sai chi phí cho các dự báo tốt nhất trong lá.
- ID3: Một thuật toán của cây quyết định. Lấy đạo hàm thuật toán ID3 của Quinlan (1979), một số thông số được thêm vào, nhập vào bất kỳ các thuộc tính.
- K-NN: Thuật toán K-lân cận gần nhất.
- Linear discriminant analysis: Giám sát phân tích biệt thức tuyến tính, tất cả đầu vào là liên tục, kiểm tra cộng tuyến.
- Multilayer perceptron: MLP mạng lưới thần kinh, thuộc tính đầu vào liên tục.
- Multinominal Logistic Regression: Hồi quy lo-gic đa định danh.
- Naïve bayes: Thuật toán Naïve bayes.
- Naïve bayes continuons: Giám sát phân loại Naïve Bayes cho dự đoán liên tục. Theo giả định Gaussian. Theo giả thiết phương sai có điều kiện không đổi (homoscedasticity), chúng ta có thể có được một mô hình tuyến tính (đúng) hoặc một mô hình bậc hai (sai).
- Prototype-NN: Nguyên mẫu – lân cận gần nhất, thuộc tính đầu vào liên tục, hạt nhân có thể được định nghĩa bởi phân cụm.

- Radial basis function: Mạng thần kinh RBF, thuộc tính đầu vào liên tục, phương pháp học off-line của các hạt nhân, với giải thuật phân cụm được áp dụng cho các mẫu.

(5) Feature construction:

Menu chứa các tính năng phục vụ cho việc xây dựng các dataset phù hợp với các phương pháp khai thác dữ liệu sẽ được ứng dụng, được sử dụng như tính năng tiền xử lý dữ liệu, với các công cụ như:

- Fomula: Tính toán thuộc tính mới từ biểu thức đại số. Trong menu tham số cho phép bạn xác định sự biểu hiện của một toán học mới. Thực hiện và xem các menu của biểu thức mới. Các lỗi có thể được cảnh báo và báo cáo.
- Residual Scores: tính toán các điểm hồi quy còn lại của mỗi thuộc tính mục tiêu vào các thuộc tính đầu vào. Các thuộc tính mới có thể được sử dụng trong các phân tích tiếp theo như tính toán của các mối tương quan một phần.
- Rnd Proj: tính toán một thuộc tính mới từ các sản phẩm của hai thuộc tính chọn ngẫu nhiên. Thành phần này là thử nghiệm nên không sử dụng
- Standardize: tiêu chuẩn hóa thuộc tính liên tục. Bình thường hoặc tiêu chuẩn hóa các thuộc tính liên tục sử dụng bình thường biểu thức toán học, ví dụ như: new_value= (giá trị trung bình) / độ lệch chuẩn.
- Trend: Tạo ra một xu hướng (1,2,3, .., n) trong một cột mới (thuộc tính liên tục).

(6) Spv learning assessment:

- Bootstrap: Áp dụng phương pháp học có giám sát để đánh giá với giá trị của Error rate là 0,632 và 0,632+ giá trị ngưỡng (bootstrap). Thành phần này tính toán ma trận tổng thể rắc rối (overall confusion matrix) và đưa ra một ước lượng tỷ lệ lỗi.
- Cross-validation: Áp dụng phương pháp học có giám sát để với việc đánh giá nhiều giá trị xác thực chéo. Thành phần này được thực hiện một sự lặp lại của một xác nhận tiêu chuẩn chéo và tính trung bình tổng thể của dự báo tỷ lệ lỗi.
- Hosmer Lemeshow Test: Phương pháp kiểm tra Hosmer Lemeshow cho hồi quy lo-gic nhị phân (binary logistic regression).
- Leave-One-Out: Áp dụng phương pháp học có giám sát với một giá trị bị loại bỏ. Thành phần này thực hiện đánh giá tỷ lệ lỗi (error rate) ngoài sự cho phép cho các thuật toán học có giám sát.
- Logistic Regression Residuals: Tính toán các số dư và các giải pháp ảnh hưởng cho hồi quy tuyến tính nhị phân.
- Test: Đánh giá thuật toán học có giám sát (s) trên một tập kiểm tra xác định trước. Dữ liệu phải được phân chia vào cài đặt đào tạo (training) và thử nghiệm (testing) bằng cách sử dụng một thành phần lựa chọn mẫu. Tỷ lệ kiểm tra lỗi sẽ được tính vào các mẫu không được chọn.
- Train-test: Áp dụng phương pháp học có giám sát để đánh giá việc phân cấp dữ liệu vào cài đặt học và cài đặt kiểm tra. Thành phần này thực hiện sự lặp lại của quá trình đào tạo-kiểm tra để dánh giá tỉ lệ lỗi dự báo.

(7) Feature selection:

Đây là menu lựa chọn các tính năng trước khi thực hiện giai đoạn học, nhằm mục đích cải thiện hiệu suất phân loại dữ liệu. Các tính năng thường được lựa chọn là:Backward-logit, CFS filtering, Define status, FCBF filtering (đây là dạng thường dùng phổ biến cho việc phân loại dữ liệu), Feature ranking, Fisher filtering, Relieff, Remove constant, Runs filtering, Stepdisc.

(8) Regression:

- Backward Elimination Reg: Dự đoán giá trị của một mục tiêu (nội sinh) từ những thuộc tính đầu vào (ngoại sinh), tất cả các biến là liên tục. Nó thực hiện một hồi quy đa tuyến tính và tự động tìm tập hợp con "tốt nhất" của các thuộc tính ngoại sinh sử dụng các chiến lược loại bỏ lùi. Các bước của quá trình lựa chọn sẽ được mô tả. Kết quả hồi quy tốt nhất được cung cấp.
- C-RT Regression tree: Cây hồi quy dự đoán. Dự đoán giá trị của một thuộc tính mục tiêu liên tục với một cây hồi quy, đầu vào (s) có thể là liên tục hoặc rời rạc. Các thuật toán được sử dụng là các Breiman và al. Kết quả chi tiết về trình tự cắt tỉa có thể được mô tả. Cây tốt nhất trên việc cài đặt cắt tỉa và cây chọn lựa được tô sáng.
- Dfbetas: Đo lường dfbetas ảnh hưởng của mỗi quan sát trên từng cá nhân của các hệ số hồi quy. Điểm đặc biệt của dfbetas là số lượng các sai số chuẩn mà các hệ số thay đổi khi quan sát này được loại bỏ để hồi quy. Số lượng điểm phát hiện được tổng kết trong một bảng từ kết quả chi tiết có sẵn. Chúng ta có thể sao chép các giá trị trong một bảng tính.
- Espilon SVR: Dự đoán giá trị của một thuộc tính mục tiêu từ những thuộc tính đầu vào, tất cả đều liên tục, nó thực hiện một hỗ trợ của véc tơ hồi quy. Thành phần này sử dụng thư viện LIBSVM.
- Forward Entry Regression: Dự đoán giá trị của một thuộc tính mục tiêu (nội sinh) từ những thuộc tính đầu vào (ngoại sinh), tất cả đều liên tục. Nó thực hiện hồi quy tuyến tính đa biến và tự động tìm các tập hợp con "tốt nhất" của các thuộc tính ngoại sinh sử dụng lựa chọn chiến lược lựa chọn trước. Các bước của quá trình lựa chọn được mô tả. Kết quả hồi quy tốt nhất được cung cấp.
- Multiple linear regression: Dự đoán giá trị của một thuộc tính mục tiêu từ những thuộc tính đầu vào, tất cả đều liên tục, nó thực hiện một hồi quy tuyến tính nhiều theo nguyên tắc OLS (Ordinary least square).
- NuSVR: Dự đoán giá trị của một thuộc tính mục tiêu từ những thuộc tính đầu vào, tất cả đều liên tục, nó thực hiện một hỗ trợ của véc tơ hồi quy. Thành phần này sử dụng thư viện LIBSVM.
- Outliner Detection: Phát hiện sự chênh lệch và / hoặc các điểm có ảnh hưởng cho hồi quy tuyến tính đa biến. Số điểm phát hiện được tổng kết trong một bảng. Kết quả chi tiết có sẵn. Chúng ta có thể sao chép các giá trị trong một bảng tính
- Regression Assessment: So sánh giá trị quan sát [thuộc tính mục tiêu] với giá trị dự đoán [thuộc tính đầu vào (s)] từ phân tích hồi quy. Thành phần này cho phép để làm nổi bật các mô hình tốt nhất từ các mô hình khác.

- Regression tree: Dự đoán giá trị của một mục tiêu liên tục do với một cây hồi quy, đầu vào (s) có thể là liên tục hoặc rời rạc. Các thuật toán được sử dụng là phiên bản đơn biến của cây phân cụm. Phương pháp học tập bao gồm một quá trình cắt tỉa. Kết quả chi tiết về trình tự cắt tỉa có thể được mô tả. Cây tốt nhất là cây được cài đặt cắt tỉa và cây được chọn sẽ được nhấn mạnh.

(9) Association:

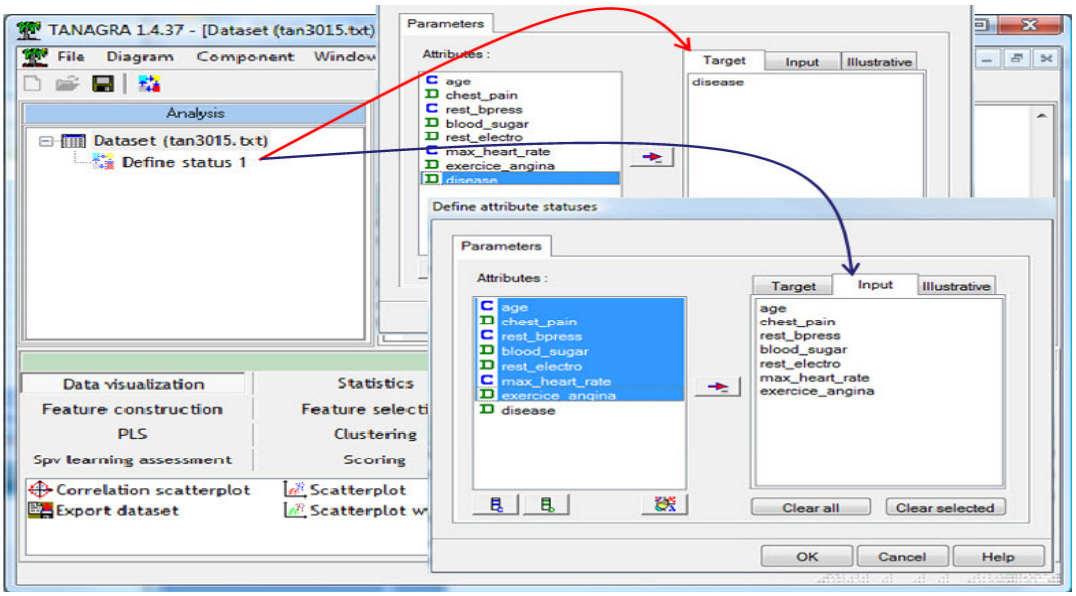
Menu chứa các thuật toán và mô hình để thiết lập theo phương pháp khai phá dữ liệu bằng luật kết hợp.

- A priori: Thành phần này tính toán luật kết hợp bằng cách sử dụng một thuật toán cơ bản được đề xuất bởi các mô tả của Agrawal "tiên nghiệm".
- A propri PT: Thuật toán A priori của Christian Borgelt được thực hiện bằng cách sử dụng cây tiền tố. Thành phần này chuẩn bị một tập tin tạm thời và gọi đến chương trình của Borgelt. Đầu ra được xử lý và xuất ra trong Tanagra. Chương trình Borgelt là một trong những chương trình tính toán nhanh chóng để đưa ra một tập luật kết hợp, nhưng hậu quả có thể chứa một chỉ mục.
- A propri MR: Thành phần này tính toán luật kết hợp theo nghiên cứu MR (2004). Đây là một phiên bản thử nghiệm trong việc xây dựng luật kết hợp.
- Assoc Outlier: Thành phần này sử dụng các nguyên tắc liên kết khai phá luật kết hợp để phát hiện các ngoại lệ (Đây là phiên bản thử nghiệm).
- Spv Assoc Rule: Giám sát luật kết hợp với bộ sinh (ví dụ: cho nhóm đặc tính đa biến). Thành phần này tính toán tất cả các quy tắc hàng đầu cho thuộc tính mục tiêu riêng biệt bằng cách sử dụng thuật toán A Priori.
- Spv Assoc Tree: Giám sát luật kết hợp với bộ sinh (ví dụ cho nhóm đặc tính đa biến. Thành phần này tính toán tất cả các quy tắc hàng đầu cho thuộc tính mục tiêu riêng biệt bằng cách sử dụng thuật toán A Priori. Thành phần này là hiệu quả hơn so với thành phần " Spv Assoc Rule ".

III.1.2.4. Phân tích các nhóm sử dụng thống kê mô tả

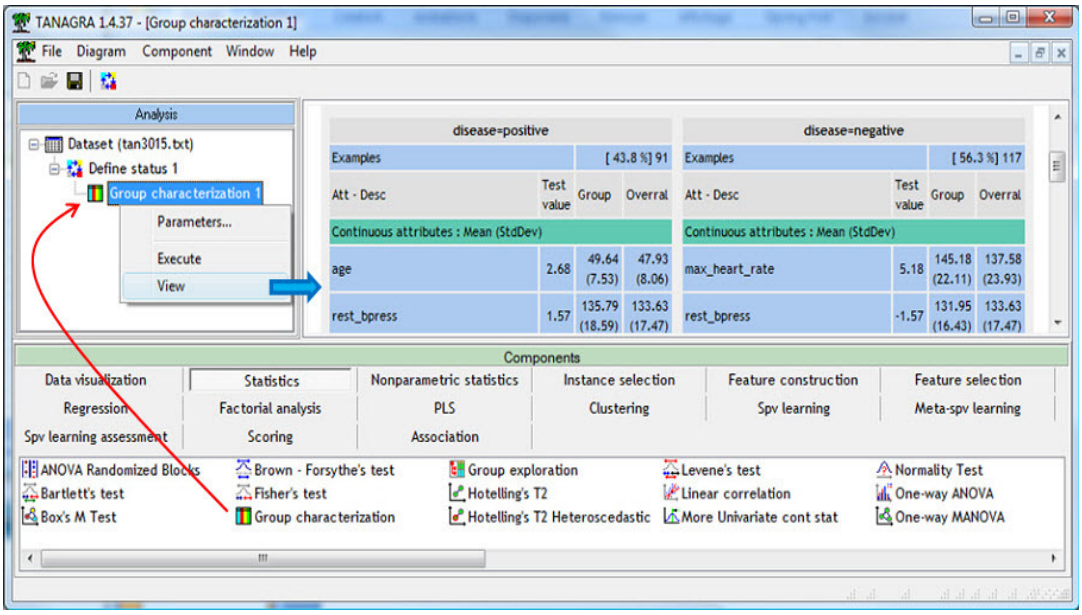
Trong khuôn khổ của quá trình học có giám sát, tính toán các thống kê mô tả của các mô tả theo các thành viên nhóm thường là các thông tin. Đây là những mô tả đơn biến của các quá trình.

Chúng ta chèn các thành phần DIFINE STATUS vào biểu đồ. Chúng ta cài đặt DISEASE như là mục tiêu (Target), các biến khác là đầu vào (Input).



Hình III-10. Quá trình cài đặt biến đầu vào và kết quả

Sau đó, chúng ta thêm thành phần GROUP CHARACTERIZATION (trong tab Statistics) vào biểu đồ.



Hình III-11. Thực hiện chức năng Group Characterization

Chúng ta sẽ có kết quả như bên dưới:

Results

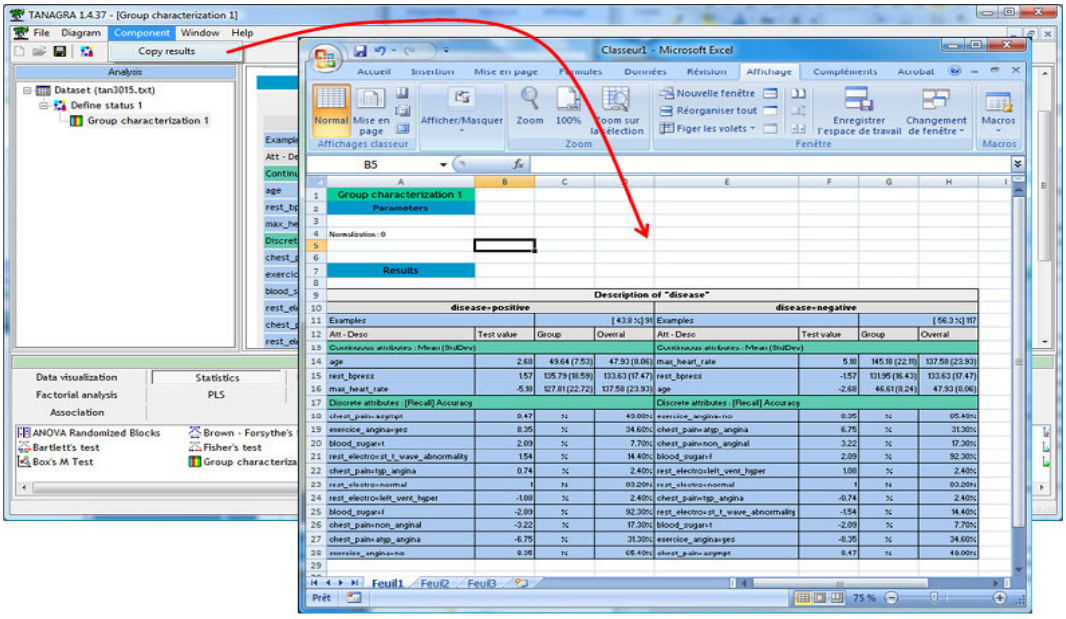
Description of "disease"

disease=positive				disease=negative			
Examples			[43.8 %] 91	Examples			[56.3 %] 117
Att - Desc	Test value	Group	Overral	Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)				Continuous attributes : Mean (StdDev)			
age	2.68	49.64 (7.53)	47.93 (8.06)	max_heart_rate	5.18	145.18 (22.11)	137.58 (23.93)
rest_bpress	1.57	135.79 (18.59)	133.63 (17.47)	rest_bpress	-1.57	131.95 (16.43)	133.63 (17.47)
max_heart_rate	-5.18	127.81 (22.72)	137.58 (23.93)	age	-2.68	46.61 (8.24)	47.93 (8.06)
Discrete attributes : [Recall] Accuracy				Discrete attributes : [Recall] Accuracy			
chest_pain=asympt	8.47	[73.5 %] 82.4 %	49.0 %	exercice_angina=no	8.35	[77.2 %] 89.7 %	65.4 %
exercice_angina=yes	8.35	[83.3 %] 65.9 %	34.6 %	chest_pain=atyp_angina	6.75	[90.8 %] 50.4 %	31.3 %
blood_sugar=t	2.09	[68.8 %] 12.1 %	7.7 %	chest_pain=non_anginal	3.22	[80.6 %] 24.8 %	17.3 %
rest_electro=st_t_wave_abnormality	1.54	[56.7 %] 18.7 %	14.4 %	blood_sugar=f	2.09	[58.3 %] 95.7 %	92.3 %
chest_pain=typ_angina	0.74	[60.0 %] 3.3 %	2.4 %	rest_electro=left_vent_hyper	1.08	[80.0 %] 3.4 %	2.4 %
rest_electro=normal	-1.00	[42.2 %] 80.2 %	83.2 %	rest_electro=normal	1.00	[57.8 %] 85.5 %	83.2 %
rest_electro=left_vent_hyper	-1.08	[20.0 %] 1.1 %	2.4 %	chest_pain=typ_angina	-0.74	[40.0 %] 1.7 %	2.4 %
blood_sugar=f	-2.09	[41.7 %] 87.9 %	92.3 %	rest_electro=st_t_wave_abnormality	-1.54	[43.3 %] 11.1 %	14.4 %
chest_pain=non_anginal	-3.22	[19.4 %] 7.7 %	17.3 %	blood_sugar=t	-2.09	[31.3 %] 4.3 %	7.7 %
chest_pain=atyp_angina	-6.75	[9.2 %] 6.6 %	31.3 %	exercice_angina=yes	-8.35	[16.7 %] 10.3 %	34.6 %
exercice_angina=no	-8.35	[22.8 %] 34.1 %	65.4 %	chest_pain=asympt	-8.47	[26.5 %] 23.1 %	49.0 %

Hình III-12. Hiển thị dữ liệu trong menu động view

III.1.2.5. Sao chép kết quả vào Excel:

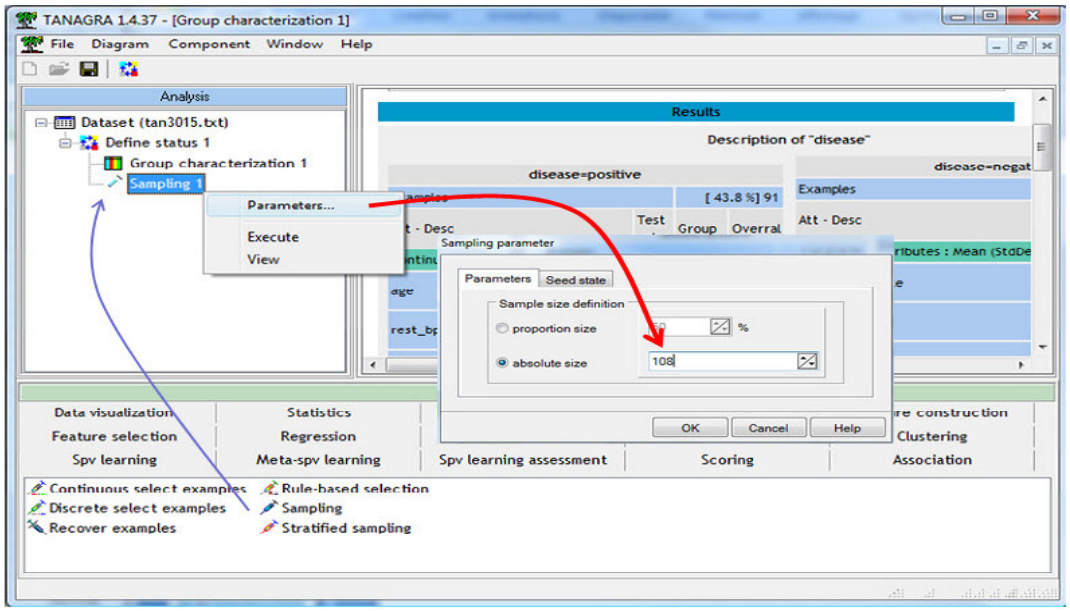
Vào Tanagra, chúng ta click vào menu COMPONENT / COPY RESULTS. Chúng ta có thể dán vào một bảng tính mới (Ctrl+V). Các giá trị được xác nhập vào các ô của Excel. Cấu trúc của bảng sẽ được lưu giữ lại. Tất nhiên, từ Excel, chúng ta có thể kết hợp bất kỳ loại thông tin nào vào các tài liệu khác.



Hình III-13. Quá trình sao chép kết quả vào Excel

❖ Tạo các mẫu học và kiểm tra:

Chúng ta chèn các thành thần Sampling (trong tab INSTANCE SELECTION) vào biểu đồ. Chúng ta click chuột vào menu ngữ cảnh PARAMETERS. Chúng ta chọn số mẫu đào tạo tại mục absolute size (ở đây, ta chọn số mẫu là 108).



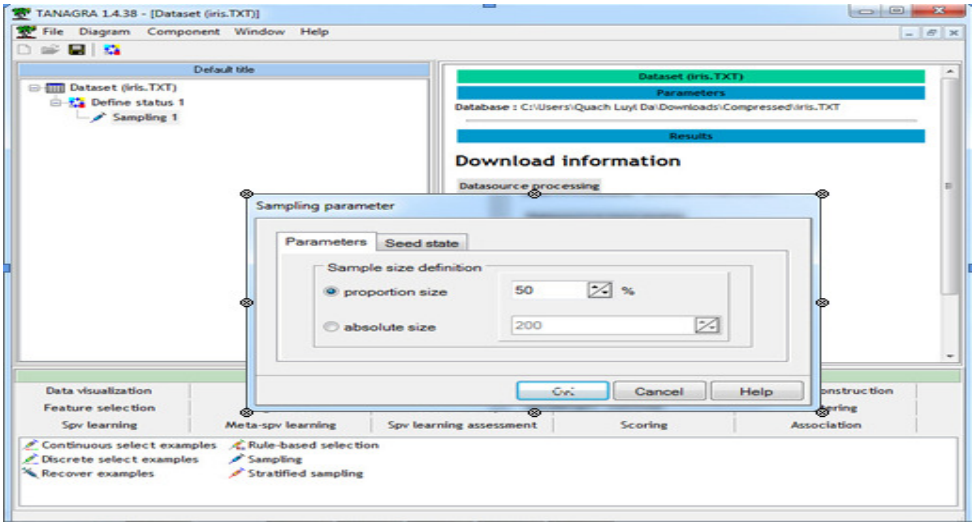
Hình III-14. Quá trình tạo mẫu học

Chúng ta xác nhận các thiết lập bằng cách nhấn vào nút OK. Sau đó, chúng ta click vào menu ngữ cảnh View để thực hiện lấy mẫu.

III.1.3. Ứng dụng Tanagra:

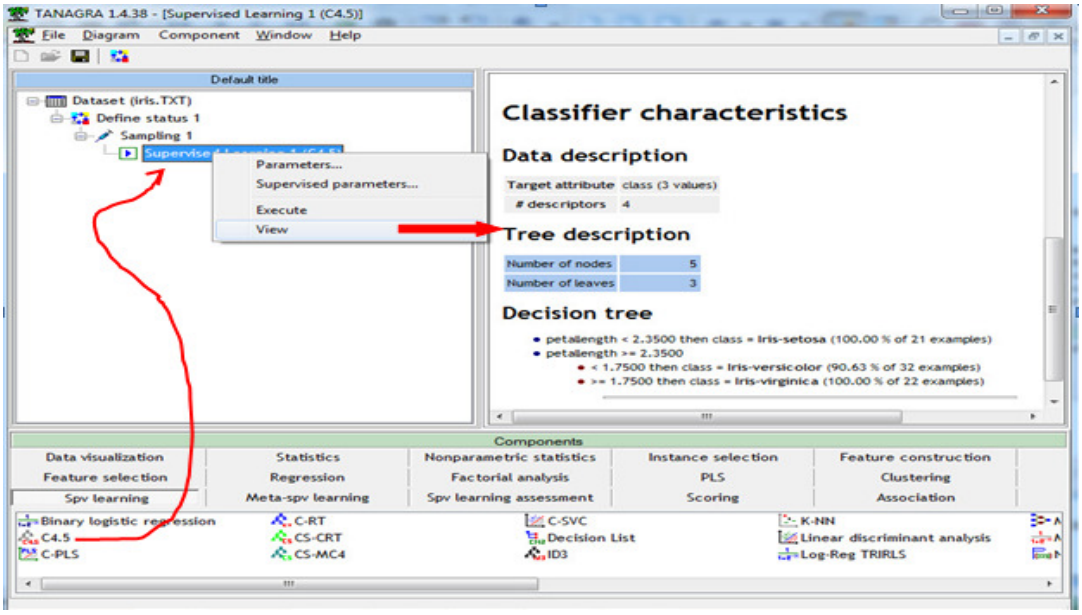
Bước đào tạo (Training test): Chúng ta sẽ sử dụng dữ liệu huấn luyện là phân loại hoa Iris ("Iris.txt"),với các quá trình thực hiện như sau:

- Tạo mẫu học, ở đây ta sẽ lấy số lượng mẫu đào tạo là mặc định proportion size.



15. Chọn số lượng mẫu đào tạo.

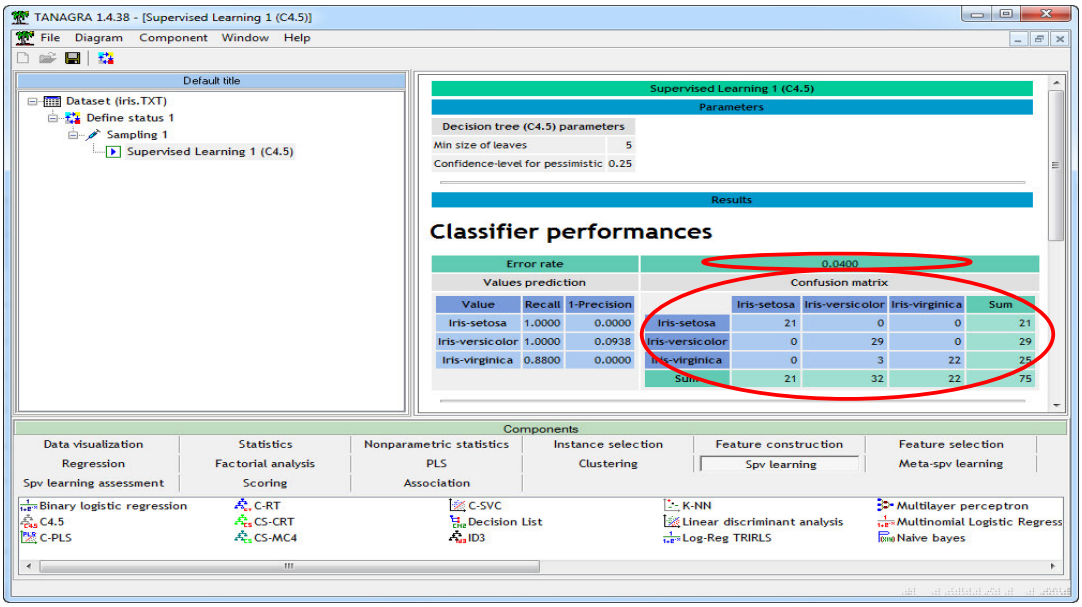
- Dựa trên tập dữ liệu Sampling 1 vừa tạo. Chúng ta chèn C4.5 (trong tab Spv learning) vào biểu đồ. Sau đó, chúng ta click vào menu ngữ cảnh View để xem kết quả.



III-16. Kết quả của quá trình học.

Bước kiểm tra:

- Tỉ lệ lỗi (Error rate) là 0.0400, tức là 0.4%. Tỉ lệ này khả quan đối với mô hình cây quyết định.



III-17. Xác định tỷ lệ lỗi.

III.2. CHƯƠNG TRÌNH ỨNG DỤNG:

III.2.1. Khai phá dữ liệu bằng luật kết hợp:

III.2.1.1. Tìm hiểu về lý thuyết:

Bài toán cơ sở:

Cho tập dữ liệu giao dịch D như sau:

TID	Items
100	M1,M2,M5
200	M2,M4
300	M2,M3
400	M1,M2,M4
500	M1,M3
600	M2,M3
700	M1,M3
800	M1,M2,M3,M5
900	M1,M2,M3

Mục tiêu:

- Liệt kê các tập phổ biến.
- Tìm tất cả các luật thỏa mãn điều kiện minisup=22% và miniconf=50%.

Giải quyết bài toán bằng cách thủ công:

Minisup=22%=2/9

TID	Items
100	M1,M2,M5
200	M2,M4
300	M2,M3
400	M1,M2,M4
500	M1,M3
600	M2,M3
700	M1,M3
800	M1,M2,M3,M5
900	M1,M2,M3

Hình III-18. Chuyển CSDL D sang nhị phân

TID	M1	M2	M3	M4	M5
100	1	1	0	0	1
200	0	1	0	1	0
300	0	1	1	0	0
400	1	1	0	1	0
500	1	0	1	0	0
600	0	1	1	0	0
700	1	0	1	0	0
800	1	1	1	0	1
900	1	1	1	0	0

Hình III-19. Cơ sở dữ liệu D

Các tập phổ biến được sinh ra và thỏa điều kiện minisup = 22% là:

Itemset	Supp
{M1}	6
{M2}	7
{M3}	6
{M4}	2
{M5}	2

Hình III-20. 1-itemset

Itemset	Supp
{M1,M2}	4
{M1,M3}	4
{M1,M5}	2
{M2,M3}	4
{M2,M4}	2
{M2,M5}	2

Hình III-21. 2-Itemsets

Itemset	Supp
{M1,M2,M3}	2
{M1,M2,M5}	2

Hình III-22. 3-Itemsets

Các tập luật được sinh ra và thỏa điều kiện minisup=22% và miniconf=70% là:

- M1^M5->M2
- M2^M5->M1
- M5->M1^M2
- M5->M1
- M5->M2
- M4-M2

III.2.1.2. Chương trình ứng dụng và kiểm tra:

Dữ liệu vào:

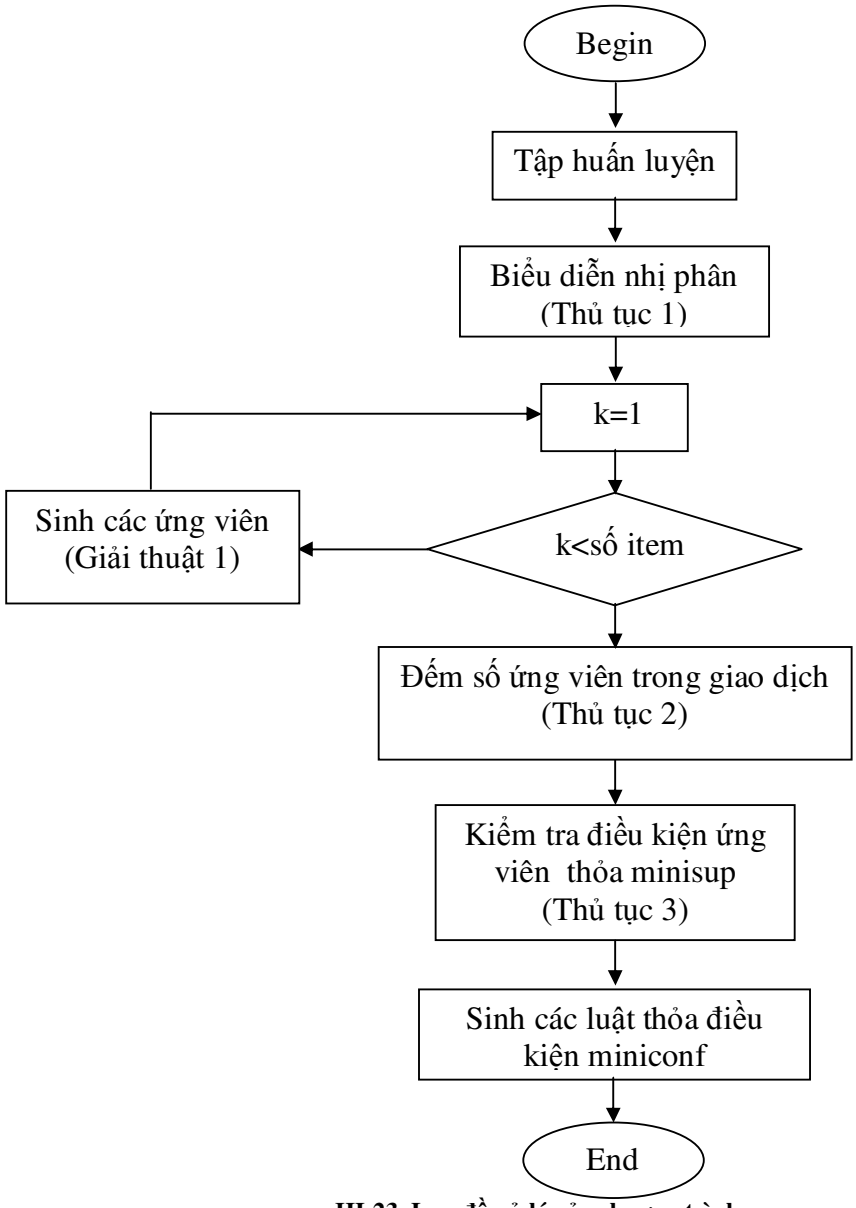
- Tập dữ liệu text phục vụ cho quá trình học, với định dạng gồm 2 cột, mỗi cột cách nhau bằng tab, mỗi dòng cách nhau bằng phím enter.
- Chỉ số minisup và miniconf phục vụ cho việc kiểm tra

Dữ liệu ra: Các luật được sinh ra dựa trên tập dữ liệu text.

Giải thuật: *Sử dụng thuật toán Apriori*

- Tiền xử lý dữ liệu bằng cách biểu diễn các giá trị của dữ liệu bằng bảng nhị phân (**Thủ tục 1**).
- Sinh các ứng viên bằng cách sử dụng cây băm (**Giải thuật 1**).
- Đếm các ứng viên xuất hiện trong giao dịch (**Thủ tục 2**).
- Kiểm tra các ứng viên thỏa điều kiện minisup (**Thủ tục 3**).
- Sinh các luật dựa trên các ứng cử viên thỏa điều kiện miniconf (**Thủ tục 4**).
- Kiểm tra tồn tại của ứng viên trong tập ứng viên (**Giải thuật 2**).
- Sinh ra vế phải của luật dựa trên vế trái của luật (**Giải thuật 3**).

Lưu đồ xử lý của chương trình:



III-23. Lưu đồ xử lý của chương trình.

Thủ tục 1:

Chức năng:

- Chuyển dữ liệu sang bảng dữ liệu nhị phân, loại bỏ dữ liệu nhiễu và phục vụ cho quá trình xử lý.

Dữ liệu vào:

- Tập dữ liệu huấn luyện sau khi đã qua xử lý.

Dữ liệu ra:

- Dữ liệu sau khi chuyển sang nhị phân.

Giải thuật:

- Khai báo itemset(chiều dài thể hiện các giao dịch (*length*), chiều ngang là các items(*length_buff*)).
- Lặp i=0 đến length_buff
  Gán itemset(0,i)= itemset
- Lặp i=0 đến lenghth
  Lấy các item trong các giao dịch
  Lặp j=0 đến số item trong giao dịch
  Lặp k=0 đến length_buff
  If item =itemset(0,k)
  Itemset(i+1,k)=1

Giải thuật 1:

Chức năng:

- Sinh các k itemset ứng viên từ các item trong tập dữ liệu.

Dữ liệu vào:

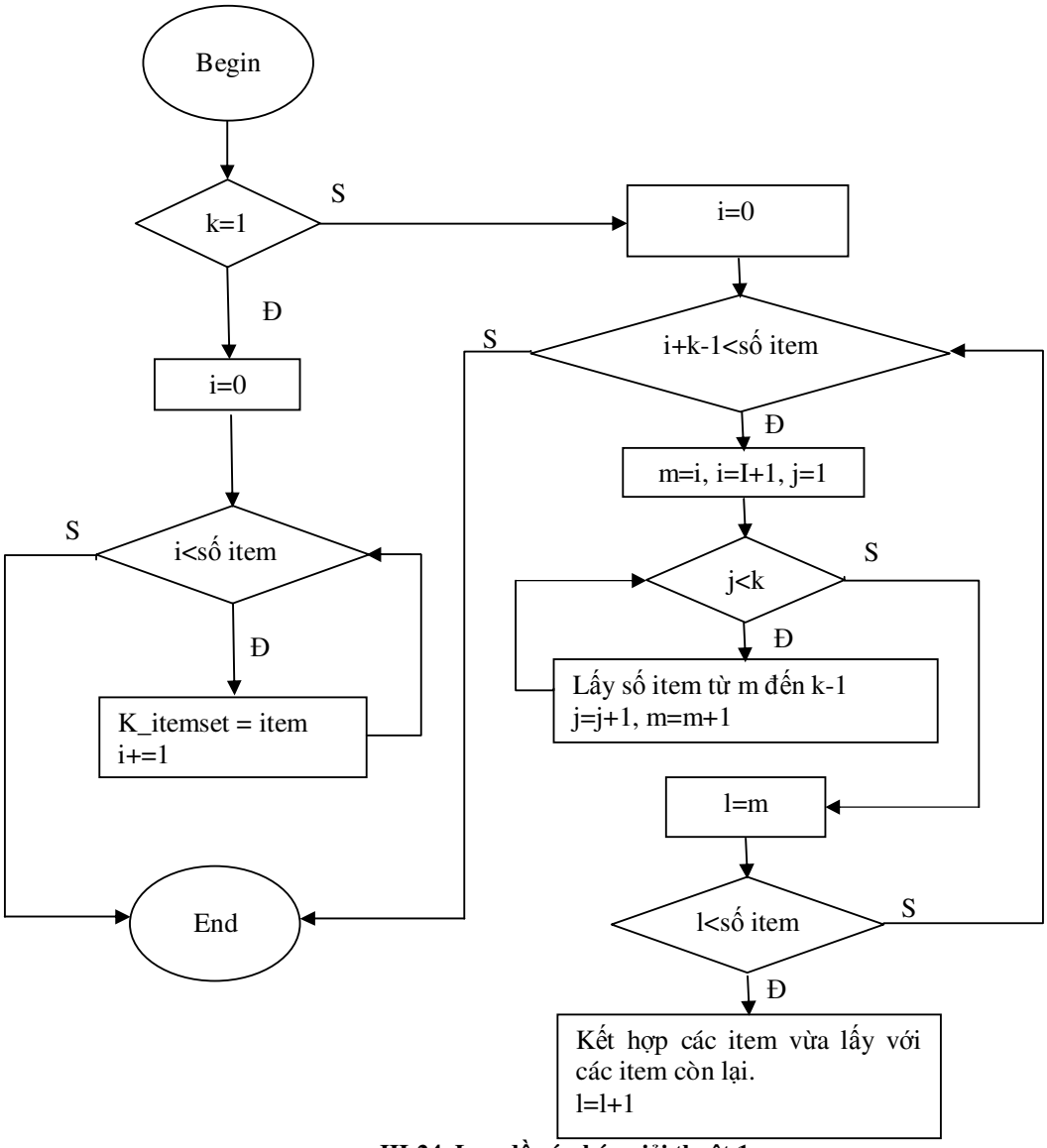
- Chỉ số k.
- Các item trong giao dịch.

Dữ liệu ra:

- Các tập k itemset.

Giải thuật:

- If k=1 then
  Lặp i=0 đến số item
  k_itemset = item
- Else
  i=0
  Lặp khi (i+k-1)< số item
  m=i
  tăng i
  Lặp j=1 đến k
  Lấy số item đến k-1
  Lặp l=m đến số item
  Kết hợp các item vừa lấy với các item còn lại để sinh ra tập ứng viên.



III-24. Lưu đồ cú pháp giải thuật 1.

Thủ tục 2:

Chức năng:

- Đếm các ứng viên xuất hiện trong các giao dịch.

Dữ liệu vào:

- Tập ứng viên k-itemset.
- Bảng giao dịch biểu diễn nhị phân (itemset(,)).

Dữ liệu ra:

- Tập ứng viên thỏa điều kiện minisup.

Giải thuật:

- Khai báo *f_k* có kiểu dữ liệu gồm itemset và count.
- Lặp i=0 đến chiều dài của tập k-itemset.

  F_k(i).itemset = k-itemt(i)
- Lặp i=0 đến chiều dài của mảng f_k

Khai báo mảng *tmp* để lưu trữ các itemset của f_k
Khai bao *tmpc* để lưu từng giá trị của mảng tmp
Khai báo mảng *vi_tri* để lưu giữ vị trí của từng item trong bảng nhị phân
Lặp j =0 đến chiều dài của mảng tmp
tmpc=tmp(j)
Lặp k=0 đến chiều rộng của itemset(,)
If tmpc=itemset(0,k) then vi_tri(j)=k
If chiều dài của mảng vi_tri là 1
Lặp k=1 đến chiều dài của mảng itemset
If (k, vi_tri(0))=1 then f_k(i).count+1
Else
Lặp j=0 đến chiều dài của itemset(,)
If itemset(j,vi_tri(0))=1 then
Khai báo biến true_false= True để dừng lặp
Lặp k=1 đến chiều dài mảng vị trí và biến true_false đúng
If itemset(j,vi_tri(k))=1
Else true_false=False và dừng vòng lặp
If true_false=True *then* f_k(i).count+1

Thủ tục 3:

Chức năng:

- Kiểm tra và loại bỏ các ứng viên không thỏa điều kiện minisup

Dữ liệu vào:

- Tập các ứng viên f_k.
- Điều kiện minisup

Dữ liệu ra:

- Các ứng viên thỏa điều kiện minisupp.

Giải thuật:

- Khai báo biến *f_k_count* để đếm số ứng viên thỏa điều kiện *minisup*.
- Khai báo *so_tap_dataset* là số lượng các giao dịch.
- Khai báo biến *minisup* để lưu trữ giá trị kiểm tra
- Lặp i=0 đến chiều dài mảng f_k
  If f_k.count/so_tap_dataset>minisup và các *f_k* phải chứa 2 item
  Then f_k_count+1
- Khai báo mảng luat có chiều dài là f_k_count gồm 2 biến item và count.
- Khai báo j=0

- Lặp i=0 đến chiều dài của f_k

  If f_k.count/so_tap_dataset>minisup và các f_k phải chứa 2 item

  Then luat(j).item=f_k(i).item, luat(j).count=f_k(i).count, tăng i, tăng j.

Thủ tục 4:

Chức năng:

- Sinh các luật ứng viên dựa trên các tập ứng viên thảo điều kiện.

Dữ liệu vào:

- Các tập ứng viên thỏa điều kiện minisupp (luat())
- Điều kiện miniconf.

Dữ liệu ra:

- Các luật thỏa điều kiện miniconf.

Giải thuật:

- Khai báo mảng *luat_sinh()* là một mảng kiểu chuỗi.
- Khai báo biến *itemset_l* là biến kiểu chuỗi để lưu tập ứng viên được xét.
- Khai báo biến *miniconf* để lưu điều kiện loại bỏ.
- Lặp i=0 đến chiều dài của mảng luật

  itemset_l=luat(i).item

  Lặp j=0 đến số itemset có trong itemset_l

  Sinh các ứng viên dựa vào giải thuật 1

  Lưu các luật được sinh ra vào mảng luat_sinh.

  Lặp k=0 đến chiều dài mảng luật sinh

  If f_k(i).count/ ktra(luat_sinh(k),f_k))>= *miniconf*

  luat(k)-> hieu_tap(itemset_l, luat(k)

Giải thuật 2: Ktra()

Chức năng:

- Kiểm tra tồn tại của ứng viên trong tập ứng viên.

Dữ liệu vào:

- Ứng viên và tập ứng viên.

Dữ liệu ra:

- Giá trị của ứng viên.

Giải thuật:

- Lặp i= 0 đến chiều dài mảng ứng viên.

  If ứng viên tồn tại *then* tra về giá trị tồn tại của ứng viên.

  Else tăng i.

Giải thuật 3:hieu_tap()

Chức năng:

- Sinh ra vế phải của luật dựa trên vế trái của luật.

Dữ liệu vào:

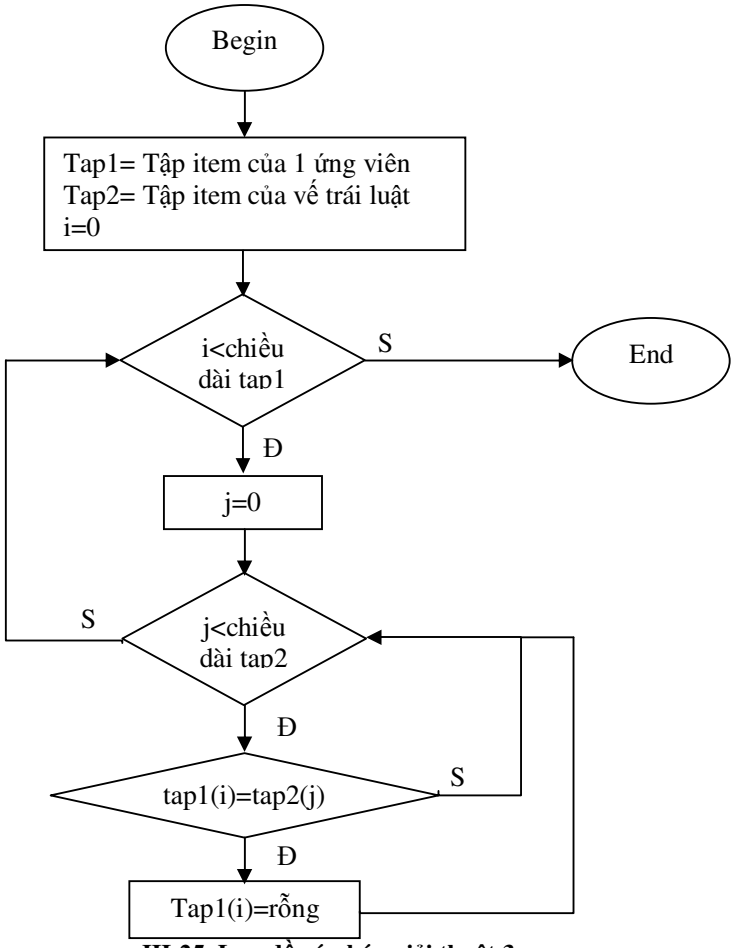
- Chuỗi ứng viên và vế trái của luật.

Dữ liệu ra:

- Vế phải của luật.

Giải thuật:

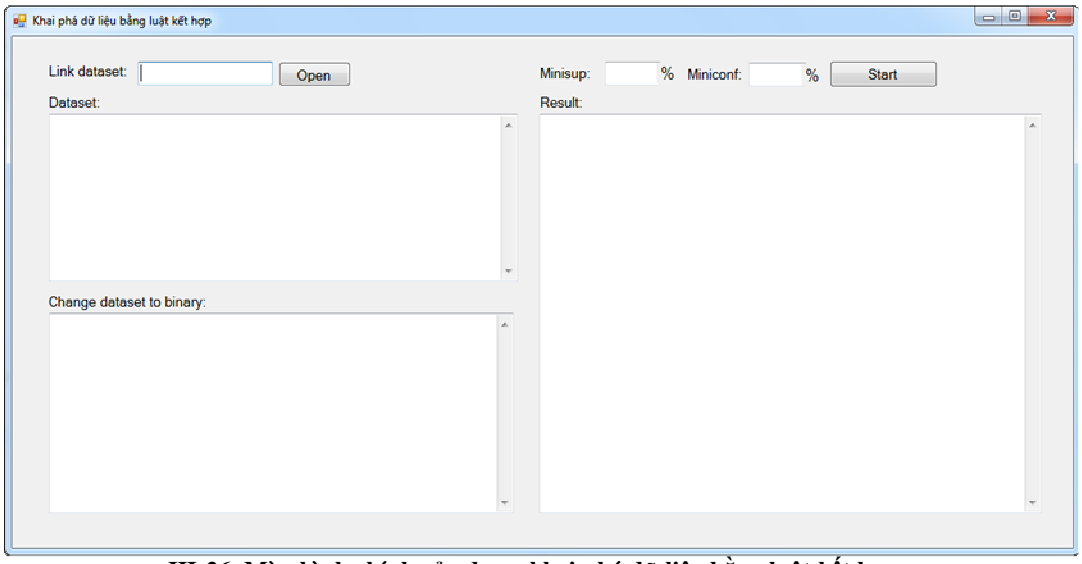
- Khai bao *tap1* lưu các item của ứng viên.
- Khai báo *tap2* lưu các item của vế trái luật.
- Lặp i=0 đến chiều dài mảng *tap1*
  Lặp j=0 đến chiều dài mảng *tap 2*
  If tap1(i)=tap2(j) *then* tap1(i)=rong.
- Lấy các giá trị tap1() khác rỗng sẽ được vế phải của luật.



III-25. Lưu đồ cú pháp giải thuật 3.

III.2.1.3. Demo minh họa:

Màn hình chính của chương trình như sau:



III-26. Màn hình chính của demo khai phá dữ liệu bằng luật kết hợp.

Chương trình khai phá dữ liệu bằng luật kết hợp có các thành phần sau đây:

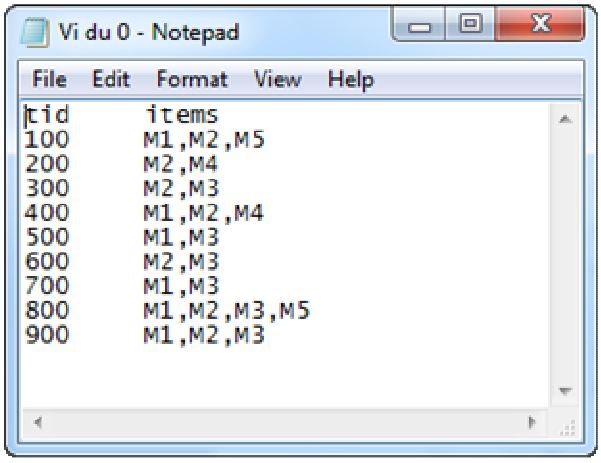
- Textbox Link dataset: Chứa đường dẫn của tập dữ liệu.
- Textbox Dataset: Hiển thị nội dung tập dữ liệu.
- Textbox Change dataset to binary: Hiển thị biểu diễn của tập giao dịch ở dạng nhị phân.
- Textbox Minisup: Nhập vào điều kiện minisup để loại các item không thỏa điều kiện.
- Textbox Miniconf: Nhập vào điều kiện để loại bỏ các luật không thỏa điều kiện.
- Textbox Result: Hiển thị kết quả sau khi khai phá gồm các tập ứng viên và các luật thỏa điều kiện minisup và miniconf.
- Button Open: Mở tập dữ liệu và hiển thị đường dẫn và nội dung tập dữ liệu vào textbox Link dataset và Dataset.
- Button Start: Bắt đầu quá trình khai phá và hiển thị vào textbox Result.
- Button Save to file: Lưu kết quả của quá trình khai phá dữ liệu của tập giao dịch ở dạng file TXT.

Mở tập dữ liệu huấn luyện:

Tập dữ liệu huấn luyện là tập có cấu trúc được lưu trữ trong tập TXT, chứa các giao dịch (Itemset). Cấu trúc gồm:

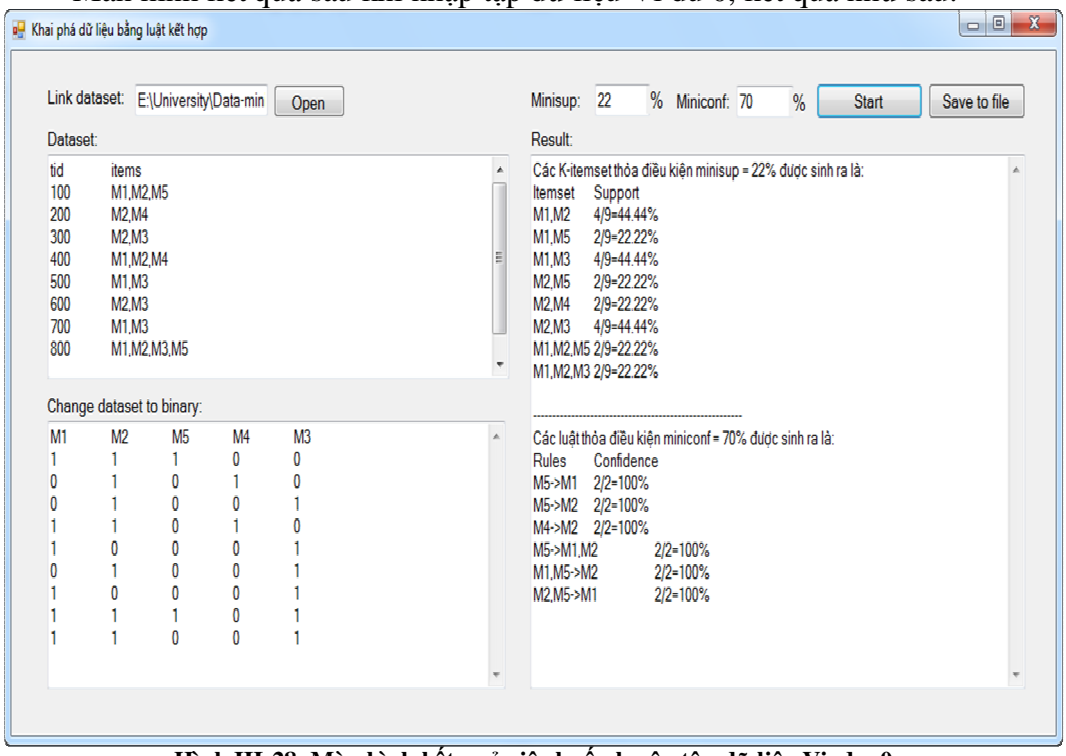
- 2 trường dữ liệu thể hiện mã giao dich và giao dịch, cách nhau bằng phím Tab.
- Các dòng cách nhau bằng phím Enter.
- Mỗi dòng có các item, cách nhau bằng dấu ",".

Ví dụ: Tập dữ liệu giao dich Vi_du_0 như sau:



Hình III-27. Tập dữ liệu Vi_du_0

Màn hình kết quả sau khi nhập tập dữ liệu Vi du 0, kết quả như sau:



Hình III-28. Màn hình kết quả việc huấn luyện tập dữ liệu Vi_du_0

III.2.2. Khai phá dữ liệu bằng cây quyết định:

III.2.2.1. Tìm hiểu về lý thuyết:

Bài toán cơ sở:

Cho tập dữ liệu như sau:

No	Size	Color	Shape	Decision
1	Vừa	Xanh dương	Hộp	Yes
2	Nhỏ	đỏ	Nón	No
3	Nhỏ	đỏ	Cầu	Yes
4	Lớn	đỏ	Nón	No
5	Lớn	Xanh lá cây	Trụ	Yes
6	Lớn	đỏ	Trụ	No
7	Lớn	Xanh lá cây	Cầu	Yes

Hình III-29. Tập dữ liệu ví dụ

Mục tiêu:

Xây dựng các luật IF-THEN dựa trên thuật toán ILA (Đã tìm hiểu ở mục II.4.3.4)

Giải quyết bài toán bằng lý thuyết:

- Bước 1: Chia bảng chứa m mẫu thành n bảng con

Bảng con 1				
No	Size	Color	Shape	Decision
1 (1)	Vừa	Xanh dương	Hộp	Yes
2 (3)	Nhỏ	đỏ	Cầu	Yes
3 (5)	Lớn	Xanh lá cây	Trụ	Yes
4 (7)	Lớn	Xanh lá cây	Cầu	Yes
Bảng con 2				
No	Size	Color	Shape	Decision
1 (2)	Nhỏ	đỏ	Nón	No
2 (4)	Lớn	đỏ	Nón	No
3 (6)	Lớn	đỏ	Trụ	No

Hình III-30. Tập dữ liệu sau khi chia làm 2 bảng con

☞ *Xét bảng con 1:*

- Bước 2: Cho j = 1.
- Bước 3: Danh sách các thuộc tính kết hợp là {|Size|, |Color|, |Shape|}.
- Bước 4: Chọn max-combination = "Xanh lá cây".
- Bước 5: Vì max-combination ≠ rỗng, nên không làm gì.
- Bước 6: Đánh dấu dòng 3,4.
- Bước 7: R1: IF Color= "Xanh lá cây" THEN Decision= "Yes"
- Bước 8: Vì Bảng con 1 còn có thuộc tính chưa đánh dấu, nên quay lại bước 4.

Bảng con 1				
No	Size	Color	Shape	Decision
1 (1)	Vừa	Xanh dương	Hộp	Yes
2 (3)	Nhỏ	đỏ	Cầu	Yes
3 (5)	Lớn	Xanh lá cây	Trụ	Yes
4 (7)	Lớn	Xanh lá cây	Cầu	Yes

Hình III-31. Bảng con 1 sau khi đánh dấu dòng 3,4.

- Bước 4: Chọn max-combination = "Vừa"
- Bước 5: Vì max-combination ≠ rỗng, nên không làm gì.
- Bước 6: Đánh dấu dòng 1.
- Bước 7: R2: IF Size = "Vừa" THEN Decision = "Yes".
- Bước 8: Vì Bảng con 1 còn có thuộc tính chưa đánh dấu, nên quay lại bước 4.

Bảng con 1				
No	Size	Color	Shape	Decision
1 (1)	Vừa	Xanh dương	Hộp	Yes
2 (3)	Nhỏ	đỏ	Cầu	Yes
3 (5)	Lớn	Xanh lá cây	Trụ	Yes
4 (7)	Lớn	Xanh lá cây	Cầu	Yes

Hình III-32. Bảng con 1 sau khi đánh dấu dòng 1,3,4.

- Bước 4: Chọn max-combination = "Cầu".
- Bước 5: Vì max-combination ≠ rỗng, nên không làm gì.
- Bước 6: Đánh dấu dòng 2.
- Bước 7: R3: IF Shape = "Cầu" THEN Decision = "Yes".
- Bước 8: Vì Bảng con 1 đã được đánh dấu tất cả nên chuyển sang Bảng con 2.

☞ *Xét bảng con 2:*

Bảng con 2				
No	Size	Color	Shape	Decision
1 (2)	Nhỏ	đỏ	Nón	No
2 (4)	Lớn	đỏ	Nón	No
3 (6)	Lớn	đỏ	Trụ	No

Hình III-33. Xét bảng con 2.

- Bước 2: j=1
- Bước 3: Danh sách các thuộc tính kết hợp là {|size|, |color|, |shape|}.
- Bước 4: Chọn max-combination = "Nón"
- Bước 5: Vì max-combination ≠ rỗng, nên không làm gì.
- Bước 6: Đánh dấu dòng 1,2.
- Bước 7: R4: IF Shape = "Nón" THEN Decision = "No".

- Bước 8: Vì còn dòng chưa đánh dấu nên quay lại bước 4.

Bảng con 2				
No	Size	Color	Shape	Decision
1 (2)	Nhỏ	đỏ	Nón	No
2 (4)	Lớn	đỏ	Nón	No
3 (6)	**Lớn**	**đỏ**	**Trụ**	**No**

Hình III-34. Bảng con 2 sau khi đánh dấu dòng 1,2.

- Bước 4: max-combination = {}
- Bước 5: Vì max-combination = rỗng, nên j=2 và quay lại bước 3.
- Bước 3: Danh sách các thuộc tính kết hợp là {|Size, Color|,|Size, Shape|, |Color, Shape|}
- Bước 4: Chọn max-combination = "Lớn" và "đỏ".
- Bước 5: Vì max-combination ≠ rỗng, nên không làm gì.
- Bước 6: Đánh dấu dòng 3.
- Bước 7: R5: IF Size = "Lớn" AND Color = "đỏ" THEN decision = "No".
- Bước 8: Vì các bảng đều đã được xét nên kết thúc thuật toán.

Tập luật được sinh ra là:

R1: IF Color= "Xanh lá cây" THEN Decision= "Yes"

R2: IF Size = "Vừa" THEN Decision = "Yes"

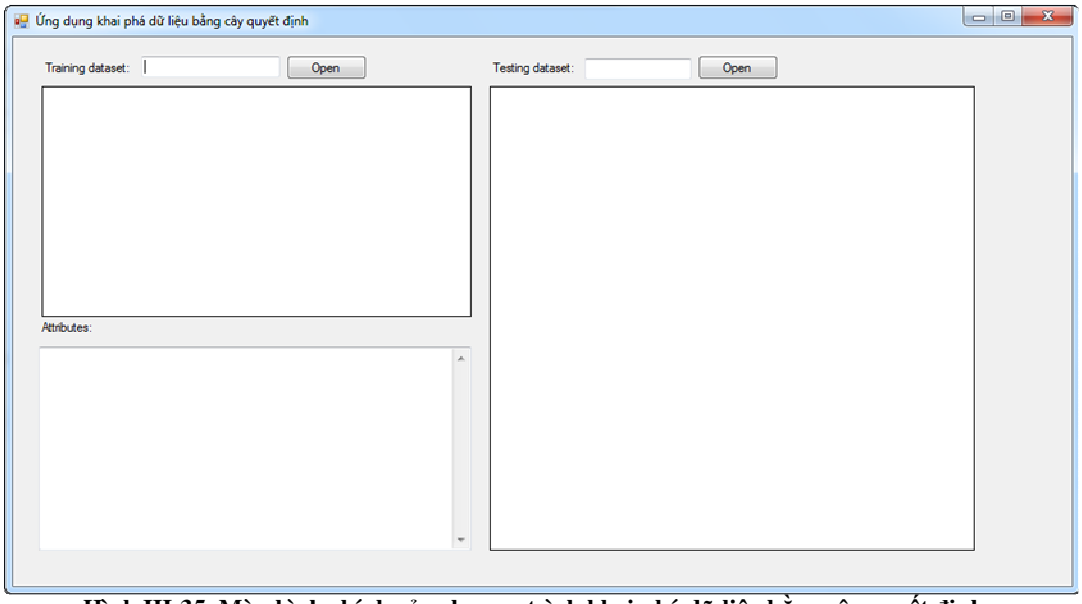
R3: IF Shape = "Cầu" THEN Decision = "Yes"

R4: IF Shape = "Nón" THEN Decision = "No"

R5: IF Size = "Lớn" AND Color = "đỏ" THEN Decision = "No"

III.2.2.2. Demo minh họa:

➢ Màn hình chính của chương trình:



Hình III-35. Màn hình chính của chương trình khai phá dữ liệu bằng cây quyết định.

➢ *Chương trình khai phá dữ liệu bằng cây quyết định có các thành phần sau đây:*

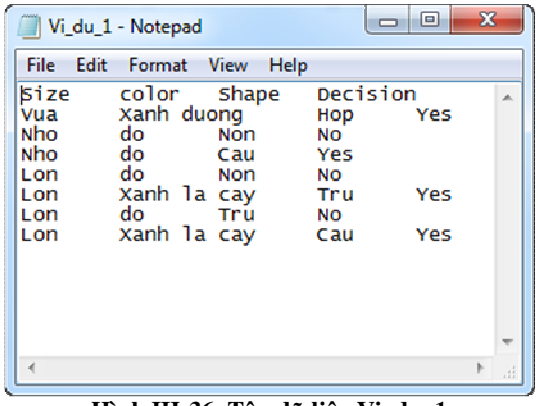
- Textbox Training dataset: Đường dẫn tập dữ liệu huấn luyện và hiển thị nội dung của tập dữ liệu huấn luyện ở datagridview bên dưới.
- Textbox Attribute: Hiển thị kết quả của quá trình khai phá dữ liệu.
- Textbox Testing dataset: Đường dẫn của tập dữ liệu kiểm tra độ chính xác của tập luật và hiển thị nội dung của tập kiểm tra ở datagridview bên dưới.
- Button Open: Mở tập dữ liệu huấn luyện và tập dữ liệu kiểm tra.

➢ *Mở tập dữ liệu:*

Tập dữ liệu huấn luyện là tập có cấu trúc được lưu trữ trong tập TXT, chứa các giao dịch (Itemset). Cấu trúc gồm:

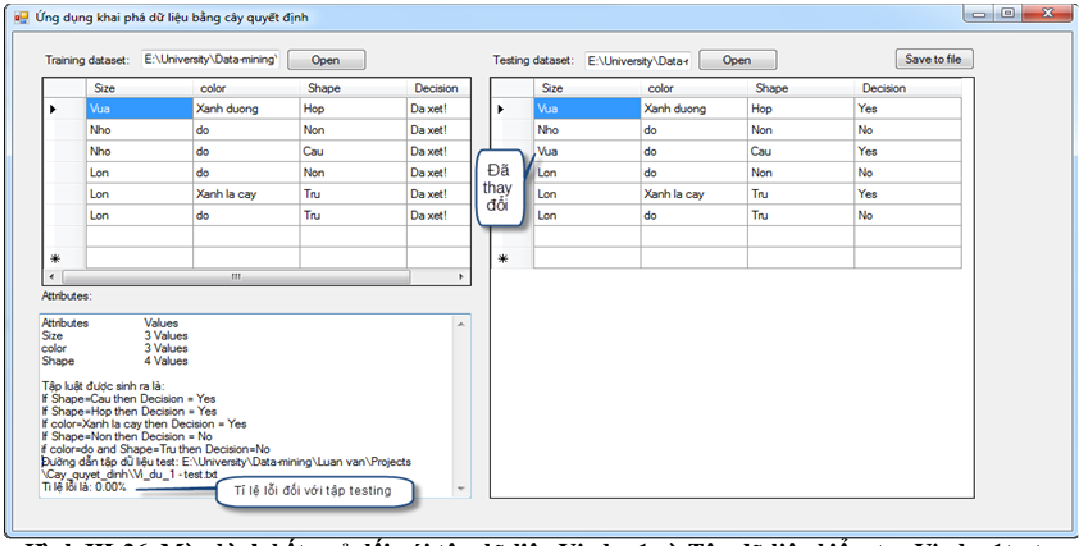
- 2 trường dữ liệu thể hiện mã giao dich và giao dịch, cách nhau bằng phím Tab.
- Các dòng cách nhau bằng phím Enter.

Ví dụ: Tập dữ liệu Vi_du_1 như sau:



Hình III-36. Tập dữ liệu Vi_du_1

➢ *Màn hình kết quả đối với tập dữ liệu Vi_du_1 và tập dữ liệu kiểm tra Vi_du_1test như sau:*



Hình III-36. Màn hình kết quả đối với tập dữ liệu Vi_du_1 và Tập dữ liệu kiểm tra Vi_du_1test.

KẾT LUẬN VÀ KIẾN NGHỊ

I. KẾT LUẬN:

Luận văn tập trung nghiên cứu các quá trình khai phá dữ liệu từ dữ liệu thô ban đầu đến dữ liệu đã qua xử lý và phục vụ cho quá trình khám phá tri thức. Qua việc nghiên cứu các phương pháp và các giải thuật khai phá dữ liệu, luận văn cho thấy được sự hữu ích của dữ liệu phục vụ cho quá trình kinh doanh, nghiên cứu và học tập.

Một số kết quả đạt được:

- Tổng kết những vấn đề nghiên cứu về khai phá dữ liệu và khám phá tri thức từ dữ liệu.
- Tìm hiểu về các kỹ thuật khai phá dữ liệu, làm nền tản cho quá trình khám phá tri thức từ dữ liệu.
- Tìm hiểu về chương trình khai phá dữ liệu phục vụ cho quá trình nghiên cứu và học tập.
- Đã làm sáng rõ sự cần thiết của việc khai phá dữ liệu và ứng dụng tri thức trong lĩnh vực kinh doanh, nghiên cứu và học tập.
- Áp dụng những vấn đề nghiên cứu về kỹ thuật khai phá dữ liệu bằng luật kết hợp và cây quyết định vào khai phá dữ liệu cơ bản.

Một số hướng phát triển:

- Tìm hiểu thêm về các phương pháp khai phá dữ liệu khác.
- Mở rộng nghiên cứu khai phá dữ liệu từ hình ảnh và web.
- Tìm hiểu thêm về ngôn ngữ lập trình để có thể cải tiến, rút ngắn các giải thuật và thủ tục, cũng như xây dựng thêm các thuật toán khai phá dữ liệu để có thể phục vụ cho công việc nghiên cứu.

II. KIẾN NGHỊ:

Trong quá trình nghiên cứu tôi đã học hỏi được rất nhiều về một kiến thức mới và hiểu được tầm quan trọng của dữ liệu sau khi sử dụng. Ngoài mục đích lưu trữ còn có thể phục vụ tốt cho công việc định hướng kinh doanh, nghiên cứu và học tập. Đó là "nguồn tài nguyên" hữu ích phục vụ cho đời sống con người. Qua quá trình nghiên cứu, tôi đã hiểu và biết được sự hữu ích của dữ liệu qua các một số khai phá dữ liệu.

Bên cạnh đó, vì thời gian nghiên cứu còn hạn hẹp nên chỉ nghiên cứu được các kỹ thuật khai phá dữ liệu cơ bản trên dữ liệu là các tập tin văn bản, chưa đi sâu vào nghiên cứu khai phá dữ liệu trên dữ liệu là hình ảnh, âm thanh, web,….Đồng thời, do môn học còn mới nên chưa nắm bắt được hết nội dung của môn học. Chính vì vậy, rất mong nhà trường và ban chủ nhiệm khoa xem xét việc đưa môn học này vào giảng dạy, để các khóa sinh viên sau sẽ tiếp thu được một môn học mới và hữu ích cho cuộc sống. Riêng tôi, sẽ cố gắng phát triển hoàn chỉnh về cơ sở lý thuyết về khai phá dữ liệu, tìm hiểu thêm về ngôn ngữ lập trình để có thể đưa ra được các giải pháp và giải thuật mới phục vụ cho nhu cầu học tập và công việc sau khi ra trường.

PHỤ LỤC

Phụ lục I: Đo khoảng cách giữa 2 đối tượng dữ liệu:

1) Khoảng cách Euclid (Euclid distance) giữa 2 điểm x, y trong không gian n chiều:

$$d(x,y) = \sqrt{\sum_{i=1}^{k}(x_k - y_k)^2}$$

Trong đó: - n là số chiều.

- x_k, y_k là giá trị thuộc tính thứ k của đối tượng x và y.

2) Khoảng cách Haming (Haming distance) giữa hai đối tượng là số bit khác nhau giữa hai đối tượng chỉ có giá trị nhị phân.

3) Khoảng cách lớn nhất (Supremun distance) là khoảng cách lớn nhất giữa 2 thuộc tính bất kì của 2 đối tượng.

4) Khoảng cách Minkowski (Minkowski distance) giữa 2 đối tượng được định nghĩa như sau:

$$d(x,y) = \left(\sum_{k=1}^{n}|x_k - y_k|^r\right)^{1/r}$$ trong đó r là tham số

- Khi r = 1: Khoảng cách Minkowski trở thành khoảng cách Hamming.
- Khi r = 2: Khoảng cách Minkowski trở thành khoảng cách Euclid.
- Khi r = ∞: Khoảng cách Minkowski trở thành khoảng cách lớn nhất.

Phụ lục II: Thuật giải Heuristic

Thuật giải Heuristic là một sự mở rộng khái niệm thuật toán. Nó thể hiện cách giải bài toán với các đặc tính sau:

- Thường tìm được lời giải tốt (nhưng không chắc là lời giải tốt nhất).
- Giải bài toán theo thuật giải Heuristic thường thể hiện khá dễ dàng và nhanh chóng đưa ra kết luận hơn so với giải thuật tối ưu, vì vậy chi phí thấp hơn.
- Thuật toán Heuristic thường thể hiện khá tự nhiên, gần gũi với cách suy nghĩ và hành động của con người.

Có nhiều phương pháp để xây dựng một giải thuật Heuristic, trong đó người ta thường dựa trên các nguyên lý cơ bản sau:

- Nguyên lý vét cạn thông minh: Trong một bài toán tìm kiếm nào đó, khi không gian tìm kiếm lớn, người ta thường tìm cách giới hạn lại không gian tìm kiếm hoặc thực hiện một kết quả dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu.
- Nguyên lý tham ăn (Greedy): Lấy tiêu chuẩn tối ưu (trên phạm vi toàn cục) của bài toán để làm tiêu chuẩn chọn lựa hành động cho phạm vi cục bộ của từng bước (hay từng giai đoạn) trong quá trình tìm kiếm lời giải.
- Nguyên lý thứ tự: Thực hiện hành động dựa trên một cấu trúc thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được một lời giải tốt.

- Hàm Heuristic: Trong việc xây dựng các giải thuật Heuristic, người ta thường dùng các hàm Heuristic. Đó là các hàm đánh giá thô, giá trị của hàm phụ thuộc vào trạng thái hiện tại của bài toán tại mỗi bước giải. Nhờ giá trị này người ta có thể chọn được cách hành động tương đối hợp lý cho từng bước của giải thuật.

Phụ lục III: Hướng dẫn sử dụng chương trình khai phá luật kết hợp.

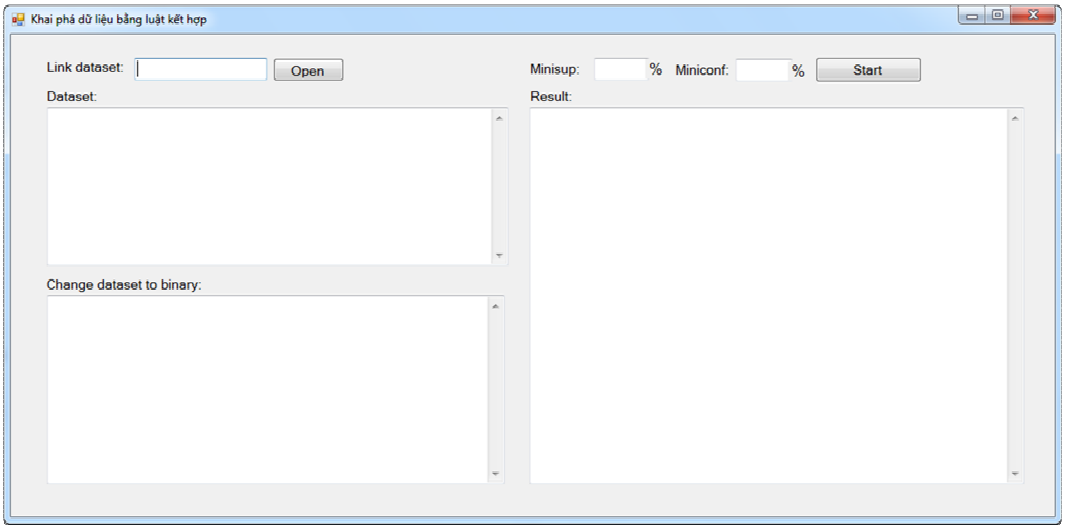
Để chạy chương trình khai phá dữ liệu bằng luật kết hợp, bạn cần chạy kích hoạt file "Luat_ket_hop.exe".

Bạn cần chuẩn bị một tập dữ liệu thỏa yêu cầu sau:

- Một tập dữ liệu gồm 2 cột, mỗi cột cách nhau bằng phím Tab.
- Tập dữ liệu gồm nhiều dòng (record), mỗi dòng cách nhau bằng phím Enter.
- Mỗi dòng có các item, cách nhau bằng dấu ",".

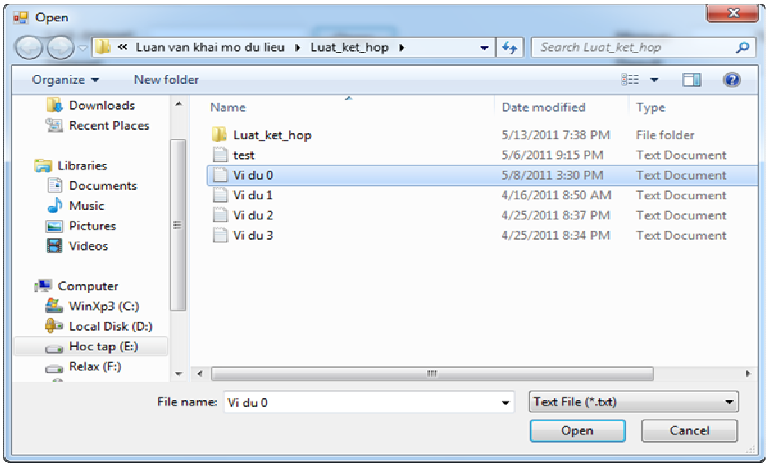
Chạy một ứng dụng:

- Màn hình chính sau khi kích hoạt file chương trình:



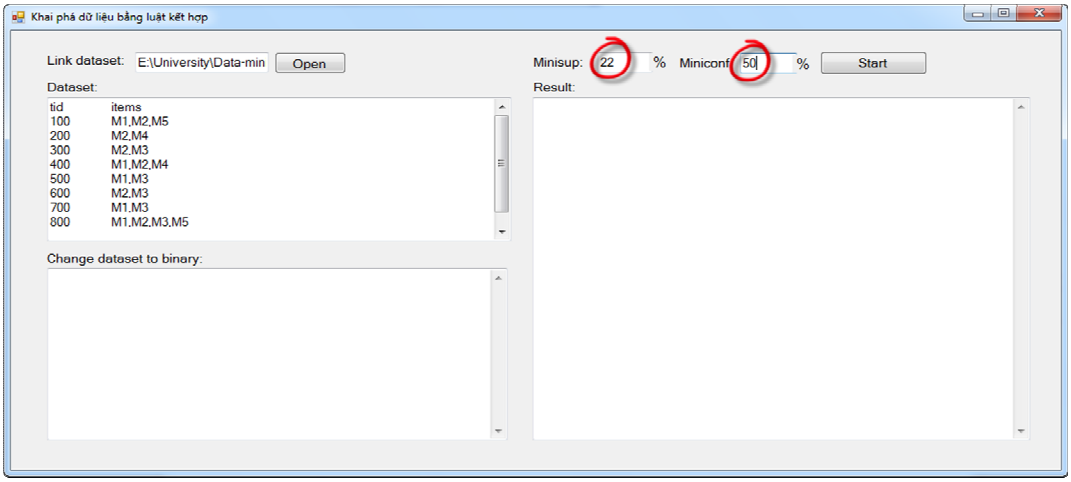
Hình 1: Màn hình chính của chương trình khai phá dữ liệu Luật kết hợp

- Click chọn nút nhấn Open để mở tập dữ liệu, chọn tập dữ liệu "Vi du 0":



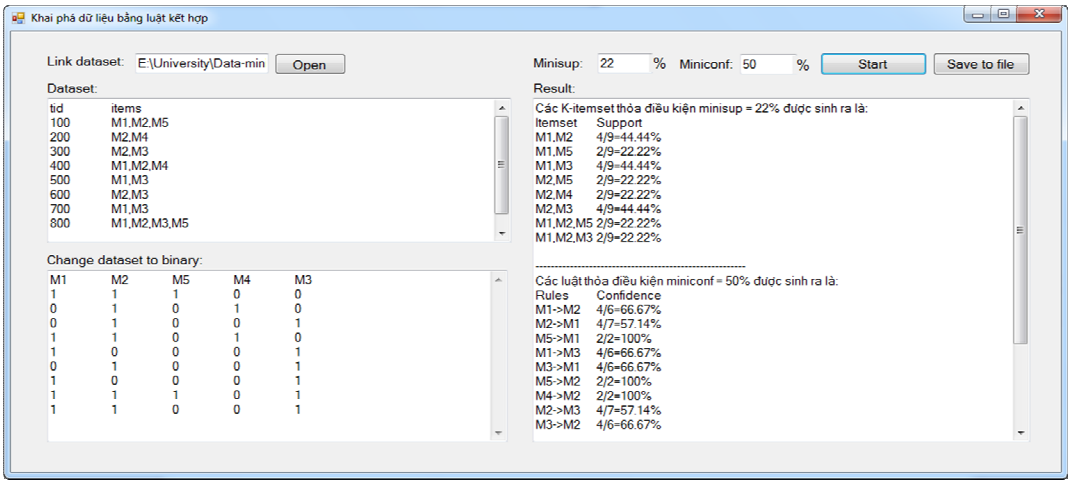
Hình 2. Của sổ mở 1 tập dữ liệu.

- Sau khi mở tập dữ liệu, màn hình sau xuất hiện:



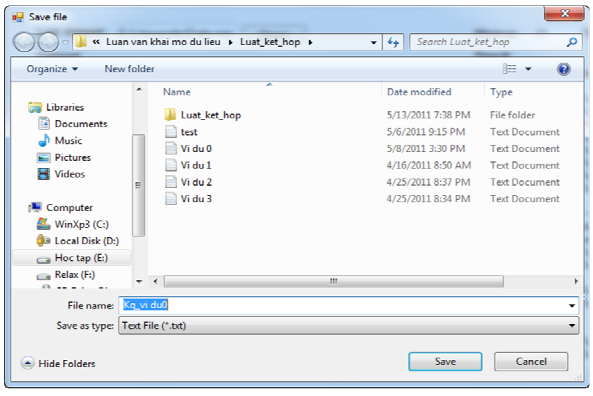
Hình 3. Màn hình điền các chỉ số minisup và miniconf để sinh luật.

- Sau khi đã nhập các chỉ số minisup và miniconf, click vào nút Start để bắt đầu khai phá và kết quả của việc khai phá tập dữ liệu Vi du 0 như sau:



Hình 4. Kết quả của chương trình.

- Sau khi kết quả xuất hiện tại textbox Result, nút nhấn Save to file hiện lên để bạn có thể lưu tập luật vừa khai phá thành một file text dữ liệu để sử dụng sau này.



Hình 5. Của sổ lưu kết quả.

Phụ lục IV: Hướng dẫn sử dụng chương trình khai phá bằng cây quyết định.

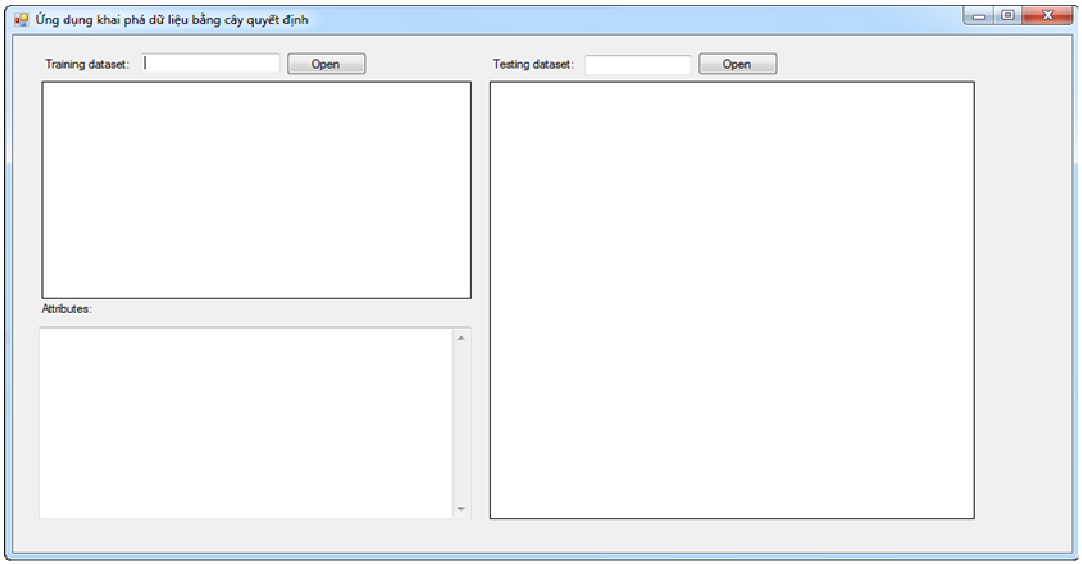
Để chạy chương trình khai phá dữ liệu bằng cây quyết định, bạn cần chạy kích hoạt file "Cay_quyet_dinh.exe".

Bạn cần chuẩn bị một tập dữ liệu thỏa yêu cầu sau:

- Một tập dữ liệu gồm nhiều cột, mỗi cột cách nhau bằng phím Tab.
- Tập dữ liệu gồm nhiều dòng (record), mỗi dòng cách nhau bằng phím Enter.

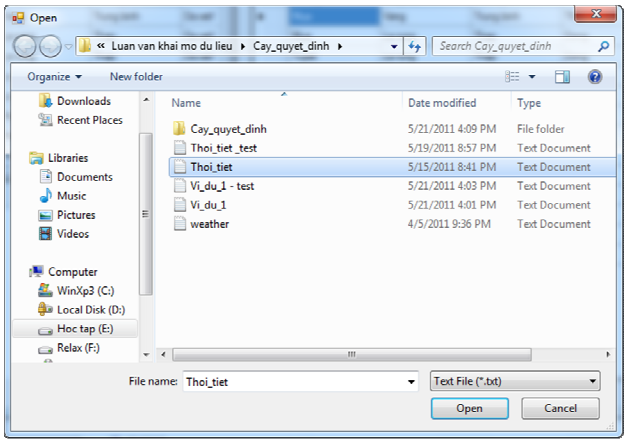
Cách chạy một ứng dụng:

- Màn hình chính của chương trình:



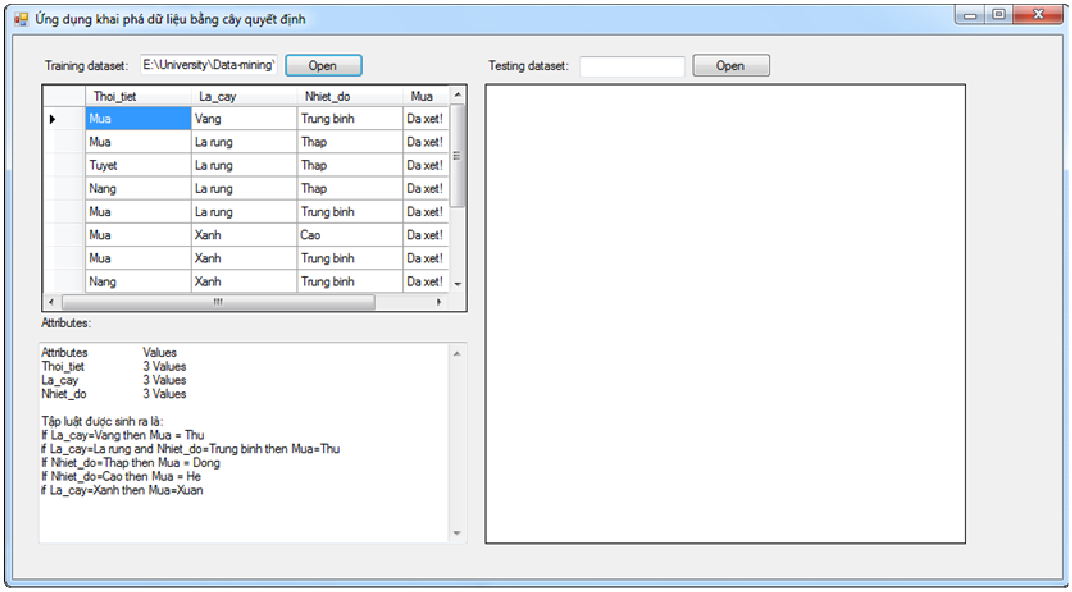
Hình 6. Màn hình chính của chương trình khai phá dữ liệu băng cây quyết định.

- Từ màn hình chính, chọn tập dữ liệu huấn luyện bằng cách nhấn nút Open của textbox Training dataset, ở đây ta chọn tập dữ liệu "Thoi_tiet.txt":



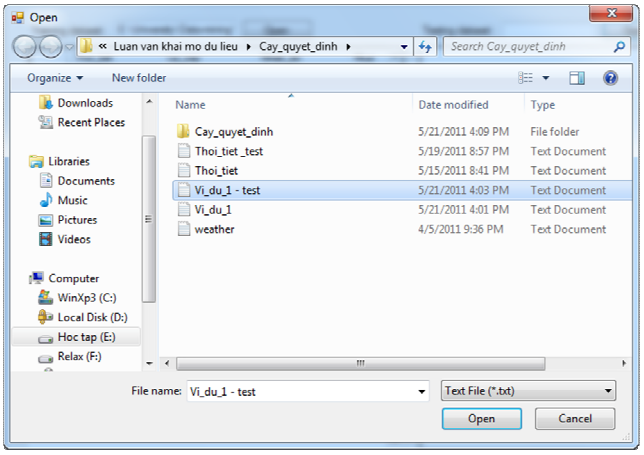
Hình 7. Cửa sổ chọn tập dữ liệu để huấn luyện.

- Kết quả sau khi chọn tập dữ liệu và sinh các luật dựa trên tập dữ liệu đó, tập luật này được sử dụng suốt trong quá trình kiểm tra:



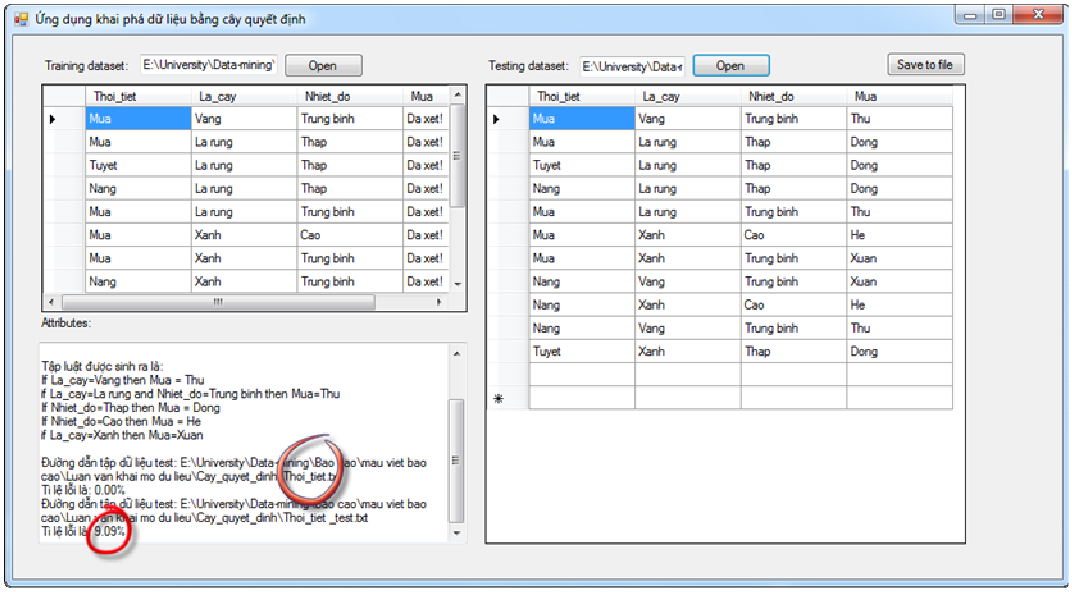
Hình 8.Cửa sổ sau khi chọn tập dữ liệu huấn luyện và các luật được sinh ra.

- Sau khi chọn tập dữ liệu huấn luyện, để kiểm tra luật ta chọn tập dữ liệu kiểm tra bằng cách nhấn nút Open của textbox Testing dataset, ở đây ta chọn tập dữ liệu "Thoi_tiet_test.txt":



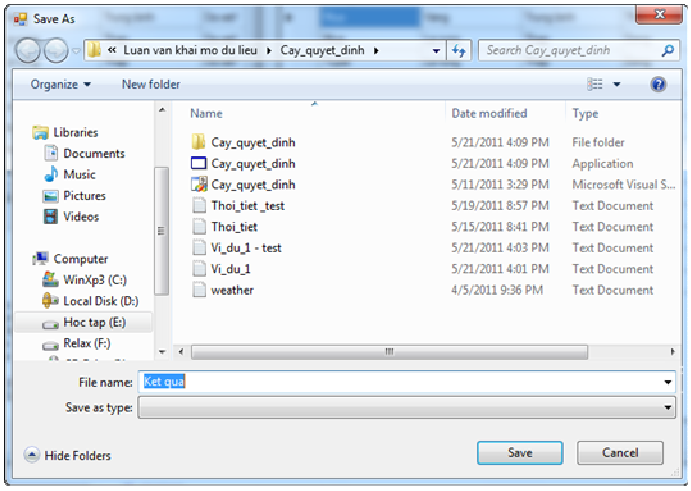
Hình 9. Cửa sổ mở tập dữ liệu kiểm tra luật.

- Sau khi chọn tập dữ liệu kiểm tra tính đúng đắng dựa trên luật được sinh ra, kết quả được hiển thị như sau:



Hình 10. Màn hình kết quả đối với việc kiểm tra tập luật.

- Sau khi chọn tập dữ liệu kiểm tra, nút nhấn Save to file xuất hiện được dùng để lưu kết quả thành tập tin dữ liệu TXT để thuận tiện cho việc sử dụng:



Hình 11. Cửa sổ lưu tập luật và kết quả kiểm tra.

Lưu ý: Chương trình khai phá dữ liệu bằng luật kết hợp muốn sinh ra tập luật dựa trên tập dữ liệu khác cần tắt chương trình và làm lại từ đầu.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1]. Dương Văn Hiếu. *Khai khoáng dữ liệu*. Khoa Công nghệ thông tin – Đại học Cần Thơ.

[2]. Nguyễn Hoàng Tú Oanh. *Bài giảng Khai phá dữ liệu và ứng dụng*. Đại học Khoa học Tự nhiên – Đại học Quốc gia Thành phố Hồ Chí Minh.

[3]. Nguyễn Nhật Quang. *Bài giảng Khai phá dữ liệu*. Viện Công nghệ thông tin và truyền thông - Đại học Bách khoa Hà Nội. Năm học 2010-2011.

Tài liệu tiếng Anh

[1]. Alex Berson & Stephen Smith, and Kurt Thearling. *Building Data Mining Applications for CRM*

[2]. John F.Elder IV & Dean w.Abbott. *A Comparison of Leading Data mining tools*. New York. 1998.

[3]. Michael J.A.Berry & Gordon S.Linoff. *Data mining techniques for marketing, sales, and customer relationship*. Indiannapolis, Indiana. 2004.

[4]. Paolo Giudici. Applied *data mining statistical methods for business and industry*. University of Pavia, Italia. 2003.

[6]. Pang-Ning Tan, Michael Steinbach, Vipin Kumar. *Introduction to Data mining*. *Pearson International*. 2006. Chapter 4,6.

[7]. *Tanagra – Data Mining Tutorials*. http://data-mining-tutorials.blogspot.com.

[8]. Xianjun Ni. *Reasearch of Data mining Based on Neural Networks*. World Academy of Science, Engineering and technology 39. 2008. www.waset.org/journals/waset/v39/v39-72.

[9]. *Decision Tree & Data mining*. www.decisiontrees.net .

[10]. John wiley&Sons. *Data mining multimedia, soft computing, and bioinformatics*. New Jersey- Canada. 2003

[11]. *Tanagra help*.

[12]. *Weka help*.

[13]. *Câu lạc bộ visual basic*. http://www.caulacbovb.com/