BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

LẠI ĐỨC HÙNG

SỬ DỤNG CÂY QUYẾT ĐỊNH ĐỂ PHÂN LOẠI DỮ LIỆU NHIỄU

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công Nghệ Thông Tin

Mã ngành: 60480201

TP. HỒ CHÍ MINH, tháng 07 năm 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

LẠI ĐỨC HÙNG

SỬ DỤNG CÂY QUYẾT ĐỊNH ĐỂ PHÂN LOẠI DỮ LIỆU NHIỄU

CÁN BỘ HƯỚNG DẪN KHOA HỌC

PGS. TS. LÊ HOÀI BẮC

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công Nghệ Thông Tin

Mã ngành: 60480201

TP. HỒ CHÍ MINH, tháng 07 năm 2015

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học:

PGS. TS LÊ HOÀI BẮC

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM (HUTECH) *ngày tháng năm 2015.*

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

TT	Họ và Tên	Chức danh Hội đồng
1		**Chủ tịch**
2		**Phản biện 1**
3		**Phản biện 2**
4		**Ủy viên**
5		**Ủy viên, Thư ký**

Xác nhận của Chủ tịch Hội đồng đánh giá Luận văn sau khi Luận văn đã sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày..... tháng….. năm 2015

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên : Lại Đức Hùng — Giới tính : Nam.

Ngày, tháng, năm sinh : 26 – 05 – 1983 — Nơi sinh : Hải Phòng.

Chuyên ngành : Công Nghệ Thông Tin — MSHV : 1341860006

I- Tên đề tài:

SỬ DỤNG CÂY QUYẾT ĐỊNH ĐỂ PHÂN LOẠI DỮ LIỆU NHIỄU

II- Nhiệm vụ và nội dung:

- Nghiên cứu về cây quyết định trong việc khai thác dữ liệu
- Nghiên cứu về dữ liệu nhiễu
- Áp dụng cây quyết định để phân loại dữ liệu nhiễu một cách hiệu quả
- Nghiên cứu, cải tiến thuật toán phân loại dữ liệu nhiễu trên cây quyết định

III- Ngày giao nhiệm vụ: 18-08-2014

IV- Ngày hoàn thành nhiệm vụ: 15-06-2015

V- Cán bộ hướng dẫn:

Phó Giáo Sư . Tiến Sĩ. Lê Hoài Bắc

CÁN BỘ HƯỚNG DẪN
(Họ tên và chữ ký)

KHOA QUẢN LÝ CHUYÊN NGÀNH
(Họ tên và chữ ký)

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này cũng như các trích dẫn hay tài liệu học thuật tham khảo đã được cảm ơn đến tác giả và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

LỜI CÁM ƠN

Trước hết, cho tôi được gửi lời cảm ơn đến sự hướng dẫn và giúp đỡ tận tình của PGS.TS. Lê Hoài Bắc.

Xin cảm ơn các bạn Trần Công Mua, Phạm Hữu Nhơn đã sát cánh và cung cấp cho tôi những kiến thức quí báu trong suốt thời gian học tập và nghiên cứu thực hiện luận văn.

Tôi cũng xin gởi lời cảm ơn đến gia đình, bạn bè và những người thân đã luôn quan tâm và giúp đỡ tôi trong suốt thời gian học tập và nghiên cứu hoàn thành luận văn này.

Luận văn không thể tránh khỏi những sai sót, rất mong nhận được ý kiến đóng góp của mọi người cho luận văn được hoàn thiện hơn.

Tôi xin chân thành cảm ơn.

TP. Hồ Chí Minh, tháng 07 năm 2015

LẠI ĐỨC HÙNG

TÓM TẮT

Phân loại dữ liệu nhiễu là một lĩnh vực rất quan trọng của khai thác dữ liệu. Thực tế thì hầu hết các cơ sở dữ liệu đều có một độ nhiễu nhất định. Do vậy rất cần các phương pháp để phân loại dữ liệu nhiễu một cách hiệu quả.

C4.5 được biết đến như là một phương pháp phổ biến, hiệu quả để xây dựng cây quyết định. Tuy nhiên nó không phù hợp lắm với những cơ sở dữ liệu nhiễu. Để phân loại dữ liệu nhiễu hiệu quả hơn, luận văn này xây dựng một thuật toán cải tiến từ thuật toán C4.5 gọi là NC4.5. NC4.5 sử dụng xác suất không chính xác (imprecise probabilities) và độ đo lường không chắc chắn (uncertainty measures) để phân loại dữ liệu nhiễu tốt hơn. NC4.5 sử dụng một tiêu chuẩn phân loại mới áp dụng cho thông tin nhiễu (Impercise Information Gain Ratio).

Kết quả thực nghiệm với dữ liệu nhiễu cho thấy thuật toán cho kết quả cây quyết định có kích thước nhỏ hơn và hiệu quả thực thi tốt hơn C4.5 và một số thuật toán khác.

ABSTRACT

Noise data classification is very important in data mining. Most database of real applications contain noisy data. We need a good method to classify noisy data.

C4.5 is a known algorithm widely used to design decision trees. But it is not good to classify noisy data. To have a better algorithm for noisy data, called NC4.5, this paper proposes to improve C4.5 algorithm by using imprecise probabilities and uncertainty measures. NC4.5 uses a new split criterion, called Imprecise Information Gain Ratio, applying uncertainty measures on convex sets of probability. NC4.5 assume that the training set is not fully reliable.

The experimental result show that NC4.5 produce smaller trees and better performance than C4.5 and some other algorithms.

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Ký hiệu, viết tắt	Ý nghĩa tiếng Việt	Ý nghĩa tiếng anh
CSDL	Cơ sở dữ liệu	Database
IDM	Mô hình không chính xác của Dirichlet	Imprecise Dirichlet Model
IG	Độ đo thông tin	Information Gain
IIGR	Tiêu chuẩn đo lường không chính xác	Imprecise Information Gain Ratio
IGR	Tỉ số độ đo thông tin	Information Gain Ratio
GPU	Bộ xử lý đồ họa	Graphics Processing Unit
Item	Mục	Item

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

CHƯƠNG 1
MỞ ĐẦU

1.1 LÝ DO CHỌN ĐỀ TÀI

Sự phát triển của công nghệ thông tin và các ứng dụng của nó đã tạo ra những cơ sở dữ liệu rất lớn. Việc khai thác các thông tin hữu ích từ các cơ sở dữ liệu này hết sức quan trọng. Khai phá dữ liệu có thể áp dụng trong nhiều lĩnh vực như: phân tích dữ liệu tài chính, công nghệ bán hàng, công nghệ viễn thông, phân tích dữ liệu sinh học, phân tích dữ liệu sinh học, …..

Tuy nhiên trong thực tế do đầu vào, do quá trình vận hành, xử lý mà các kho dữ liệu này đều có độ nhiễu nhất định. Dữ liệu nhiễu là dữ liệu bị mất, thiếu thuộc tính, hay không đồng nhất …Ứng dụng càng lớn, thời gian vận hành ứng dụng càng lâu thì dữ liệu càng dễ bị nhiễu.

Dữ liệu bị nhiễu có thể do nhiều nguyên nhân như: lỗi vận hành của phần cứng, lỗi của các thiết bị nhập liệu, các thiết bị quét dữ liệu, lỗi do lập trình, lỗi do người nhập liệu, vận hành.

Các thuật toán, phương pháp khai phá dữ liệu hiện tại như C4.5, ID3,.… đều giả định là dữ liệu hoàn toàn sạch, không bị nhiễu. Do vậy khi áp dụng các phương pháp, thuật toán này vào khai thác dữ liệu thực tế sẽ gặp khó khăn hoặc kết quả không thực sự tốt, đáng tin cậy. Do vậy rất cần các phương pháp, thuật toán có thể khai thác, phân loại dữ liệu nhiễu một cách hiệu quả.

1.2 Ý NGHĨA KHOA HỌC VÀ THỰC TIỄN

Ý Nghĩa khoa học của luận văn: nghiên cứu được ra một phương pháp phân loại dữ nhiễu một cách hiệu quả. Từ đó góp phần làm phong phú, hiệu quả hơn việc khai thác dữ liệu, nhất là những dữ liệu nhiễu.

Ý Nghĩa thực tiễn của luận văn: Phần lớn các cơ sở dữ liệu của các ứng dụng thực tế đều có một đỗ nhiễu nhất định. Do vậy nếu khai thác được các dữ liệu nhiễu này một cách hiệu quả thì sẽ có ích lợi lớn trong nhiều lĩnh vực của đời sống, khoa học. Khi phân loại dữ liệu nhiễu tốt ta có thể áp dụng để phân tích dữ liệu tài chính, công nghệ bán hàng, công nghệ viễn thông, phân tích dữ liệu sinh học, phân tích dữ liệu sinh học, ….

1.3 MỤC ĐÍCH CỦA ĐỀ TÀI

Áp dụng cây quyết định để phân loại dữ liệu nhiễu. Đưa ra thuật toán dựa trên cây quyết định để có thể khai thác các dữ liệu bị nhiễu từ đó đưa được ra các thông tin hữu ích.

1.4 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đối tượng nghiên cứu của đề tài là dữ liệu nhiễu và thuật toán cây quyết định.

Phạm vi của đề tài là thuật toán khai thác dữ liệu nhiễu dựa trên cây quyết định

1.5 PHƯƠNG PHÁP NGHIÊN CỨU

- Tiến hành thu thập và đọc các tài liệu có liên quan đến đề tài.
- Nghiên cứu tổng quan về dữ liệu nhiễu và các khái niệm có liên quan.
- Nghiên cứu về cây quyết định và các thuật toán khai thác dữ liệu dựa trên cây quyết định
- Nghiên cứu áp dụng thuật toán dựa trên cây quyết định để phân loại dữ liệu nhiễu hiệu quả.
- Xây dựng chương trình demo và đánh giá kết quả đạt được.

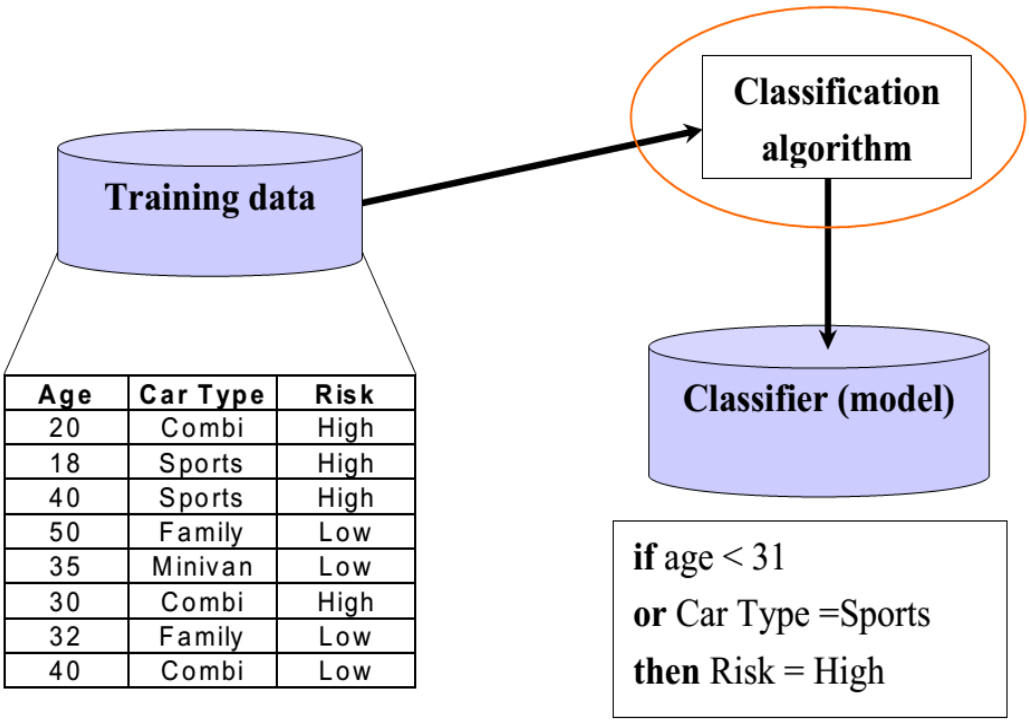
CHƯƠNG 2
TỔNG QUAN VỀ KHAI THÁC VÀ PHÂN LOẠI DỮ LIỆU

2.1 GIỚI THIỆU

Ngày nay phân lớp dữ liệu (classification) là một trong những hướng nghiên cứu chính của khai phá dữ liệu. Thực tế đặt ra nhu cầu là từ một cơ sở dữ liệu với nhiều thông tin ẩn con người có thể trích rút ra các quyết định nghiệp vụ thông minh. Phân lớp và dự đoán là hai dạng của phân tích dữ liệu nhằm trích rút ra một mô hình mô tả các lớp dữ liệu quan trọng hay dự đoán xu hướng dữ liệu tương lai. Phân lớp dự đoán giá trị của những nhãn xác định (categorical label) hay những giá trị rời rạc (discrete value), có nghĩa là phân lớp thao tác với những đối tượng dữ liệu mà có bộ giá trị là biết trước. Trong khi đó, dự đoán lại xây dựng mô hình với các hàm nhận giá trị liên tục. Ví dụ mô hình phân lớp dự báo thời tiết có thể cho biết thời tiết ngày mai là mưa, hay nắng dựa vào những thông số về độ ẩm, sức gió, nhiệt độ,… của ngày hôm nay và các ngày trước đó. Hay nhờ các luật về xu hướng mua hàng của khách hàng trong siêu thị, các nhân viên kinh doanh có thểra những quyết sách đúng đắn về lượng mặt hàng cũng như chủng loại bày bán… Một mô hình dự đoán có thể dự đoán được lượng tiền tiêu dùng của các khách hàng tiềm năng dựa trên những thông tin về thu nhập và nghề nghiệp của khách hàng. Trong những năm qua, phân lớp dữ liệu đã thu hút sựquan tâm các nhà nghiên cứu trong nhiều lĩnh vực khác nhau nhưhọc máy (machine learning), hệ chuyên gia (expert system), thống kê (statistics)... Công nghệ này cũng ứng dụng trong nhiều lĩnh vực khác nhau như: thương mại, nhà băng, maketing, nghiên cứu thị trường, bảo hiểm, y tế, giáo dục... Phần lớn các thuật toán ra đời trước đều sử dụng cơ chế dữ liệu cư trú trong bộ nhớ (memory resident), thường thao tác với lượng dữ liệu nhỏ. Một số thuật toán ra đời sau này đã sử dụng kỹ thuật cư trú trên đĩa cải thiện đáng kể khả năng mở rộng của thuật toán với những tập dữ liệu lớn lên tới hàng tỉ bản ghi [14].

Quá trình phân lớp dữ liệu gồm hai bước:

Bước thứ nhất (learning). Quá trình học nhằm xây dựng một mô hình mô tả một tập các lớp dữ liệu hay các khái niệm định trước. Đầu vào của quá trình này là một tập dữ liệu có cấu trúc được mô tả bằng các thuộc tính và được tạo ra từ tập các bộ giá trị của các thuộc tính đó. Mỗi bộ giá trị được gọi chung là một phần tử dữ liệu (data tuple), có thể là các mẫu (sample), ví dụ (example), đối tượng (object), bản ghi (record) hay trường hợp (case). Ta sử dụng các thuật ngữ này với nghĩa tương đương. Trong tập dữ liệu này, mỗi phần tử dữ liệu được giả sử thuộc về một lớp định trước, lớp ở đây là giá trị của một thuộc tính được chọn làm thuộc tính gán nhãn lớp hay thuộc tính phân lớp (class label attribute). Đầu ra của bước này thường là các quy tắc phân lớp dưới dạng luật dạng if-then, cây quyết định, công thức logic, hay mạng nơron. Quá trình này được mô tả như trong hình sau



Age	Car Type	Risk
20	Combi	High
18	Sports	High
40	Sports	High
50	Family	Low
35	Minivan	Low
30	Combi	High
32	Family	Low
40	Combi	Low

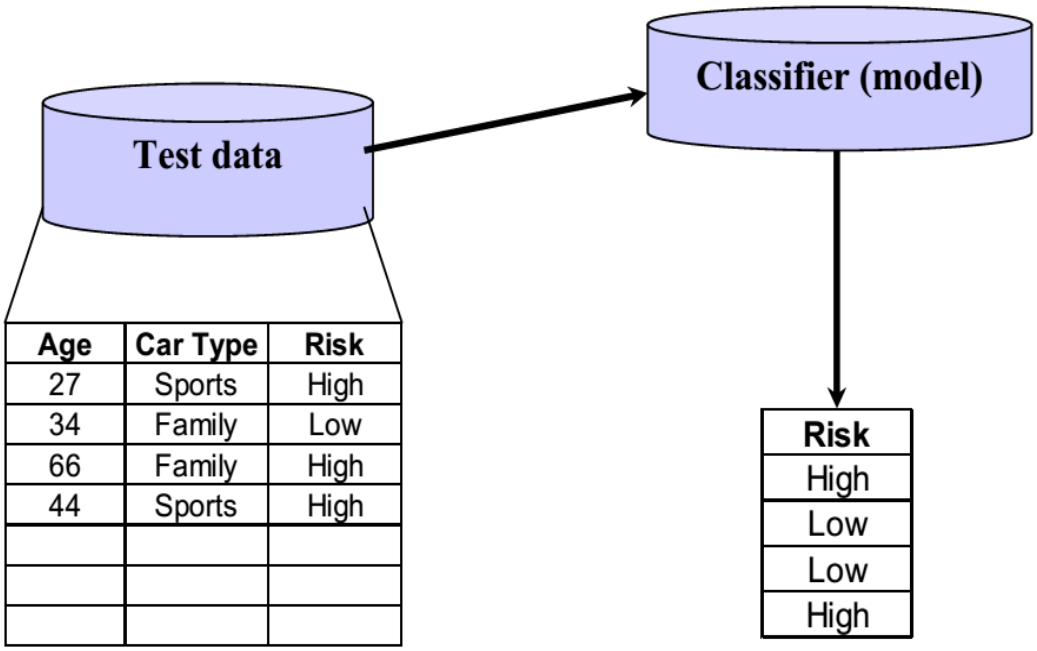
Hình 2.1 Bước xây dựng mô hình phân lớp [1]

Bước thứ hai (classification)

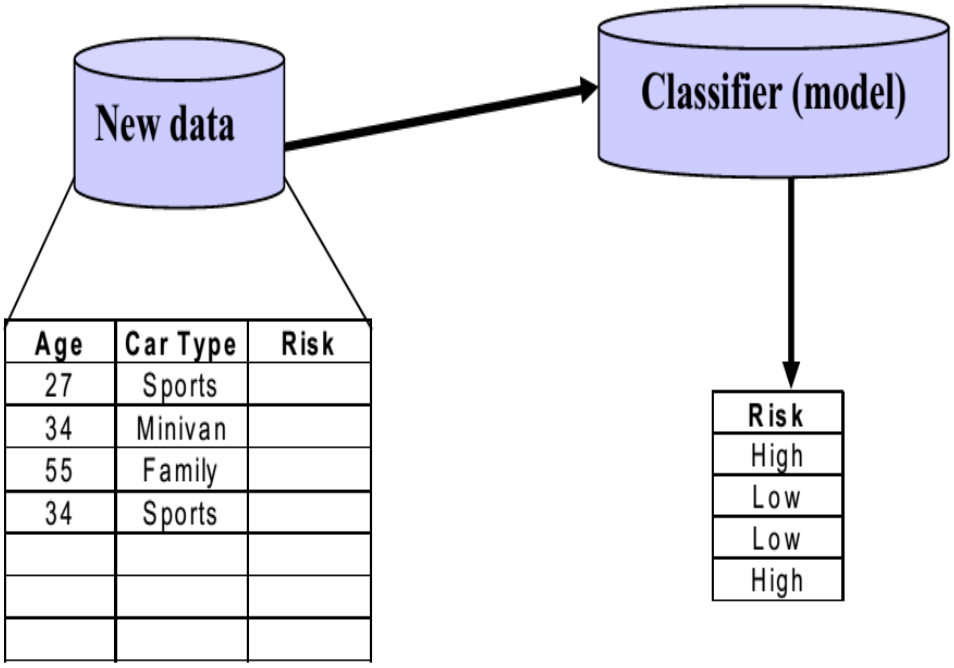
Bước thứ hai dùng mô hình đã xây dựng ở bước trước để phân lớp dữ liệu mới. Trước tiên độ chính xác mang tính chất dự đoán của mô hình phân lớp vừa tạo

ra được ước lượng. Holdout là một kỹ thuật đơn giản để ước lượng độ chính xác đó. Kỹ thuật này sử dụng một tập dữ liệu kiểm tra với các mẫu đã được gán nhãn lớp. Các mẫu này được chọn ngẫu nhiên và độc lập với các mẫu trong tập dữ liệu huấn luyện. Độ chính xác của mô hình trên tập dữ liệu kiểm tra đã đưa là tỉ lệ phần trăm các các mẫu trong tập dữ liệu kiểm tra được mô hình phân lớp đúng (so với thực tế). Nếu độ chính xác của mô hình được ước lượng dựa trên tập dữ liệu huấn luyện thì kết quả thu được là rất khả quan vì mô hình luôn có xu hướng "quá khớp" dữ liệu. Quá khớp dữ liệu là hiện tượng kết quả phân lớp trùng khít với dữ liệu thực tế vì quá trình xây dựng mô hình phân lớp từ tập dữ liệu huấn luyện có thể đã kết hợp những đặc điểm riêng biệt của tập dữ liệu đó. Do vậy cần sử dụng một tập dữ liệu kiểm tra độc lập với tập dữ liệu huấn luyện.

Nếu độ chính xác của mô hình là chấp nhận được, thì mô hình được sử dụng để phân lớp những dữ liệu tương lai, hoặc những dữ liệu mà giá trị của thuộc tính phân lớp là chưa biết.



Hình 2.2 Ước lượng độ chính xác của mô hình [1]



Hình 2.3 Phân lớp dữ liệu mới [1]

Trong mô hình phân lớp, thuật toán phân lớp giữ vai trò trung tâm, quyết định tới sự thành công của mô hình phân lớp. Do vậy chìa khóa của vấn đề phân lớp dữ liệu là tìm ra được một thuật toán phân lớp nhanh, hiệu quả, có độ chính xác cao và có khả năng mở rộng được. Trong đó khả năng mở rộng được của thuật toán được đặc biệt trú trọng và phát triển [14].

Các kỹ thuật phân lớp thường được sử dụng:

- Phân lớp cây quyết định (Decision tree classification)
- Bộ phân lớp Bayesian (Bayesian classifier)
- Mô hình phân lớp K-hàng xóm gần nhất (K-nearest neighbor classifier)
- Mạng nơron
- Phân tích thống kê
- Các thuật toán di truyền
- Phương pháp tập thô (Rough set Approach)

2.1.1 Các vấn đề liên quan đến phân lớp dữ liệu

❖ **Chuẩn bị dữ liệu cho việc phân lớp**

Việc tiền xử lý dữ liệu cho quá trình phân lớp là một việc làm không thể thiếu và có vai trò quan trọng quyết định tới sự áp dụng được hay không của mô hình phân lớp. Quá trình tiền xử lý dữ liệu sẽ giúp cải thiện độ chính xác, tính hiệu quả và khả năng mở rộng được của mô hình phân lớp.

Quá trình tiền xử lý dữ liệu gồm có các công việc sau:

Làm sạch dữ liệu

Làm sạch dữ liệu liên quan đến việc xử lý với nhiễu và giá trị thiếu (missing value) trong tập dữ liệu ban đầu. Nhiễu là các lỗi ngẫu nhiên hay các giá trị không hợp lệ của các biến trong tập dữ liệu. Để xử lý với loại lỗi này có thể dùng kỹ thuật làm trơn. Thiếu giá trị (missing value) là những ô không có giá trị của các thuộc tính. Giá trị thiếu có thể do lỗi chủ quan trong quá trình nhập liệu, hoặc trong trường hợp cụ thể giá trị của thuộc tính đó không có, hay không quan trọng. Kỹ thuật xử lý ở đây có thể bằng cách thay giá trị thiếu bằng giá trị phổ biến nhất của thuộc tính đó hoặc bằng giá trị có thể xảy ra nhất dựa trên thống kê. Mặc dù phần lớn thuật toán phân lớp đều có cơ chế xử lý với những giá trị thiếu và lỗi trong tập dữ liệu, nhưng bước tiền xử lý này có thể làm giảm sự hỗn độn trong quá trình học (xây dựng mô hình phân lớp).

Phân tích sự cần thiết của dữ liệu

Có rất nhiều thuộc tính trong tập dữ liệu có thể hoàn toàn không cần thiết hay liên quan đến một bài toán phân lớp cụ thể. Ví dụ dữ liệu về ngày trong tuần hoàn toàn không cần thiết đối với ứng dụng phân tích độ rủi ro của các khoản tiền cho vay của ngân hàng, nên thuộc tính này là dư thừa. Phân tích sự cần thiết của dữ liệu nhằm mục đích loại bỏ những thuộc tính không cần thiết, dư thừa khỏi quá trình học vì những thuộc tính đó sẽ làm chậm, phức tạp và gây ra sự hiểu sai trong quá trình học dẫn tới một mô hình phân lớp không dùng được.

Chuyển đổi dữ liệu

Việc khái quát hóa dữ liệu lên mức khái niệm cao hơn đôi khi là cần thiết trong quá trình tiền xử lý. Việc này đặc biệt hữu ích với những thuộc tính liên tục (continuous attribute hay numeric attribute). Ví dụ các giá trị số của thuộc tính thu nhập của khách hàng có thể được khái quát hóa thành các dãy giá trị rời rạc: thấp, trung bình, cao. Tương tự với những thuộc tính rời rạc (categorical attribute) như địa chỉ phố có thể được khái quát hóa lên thành thành phố. Việc khái quát hóa làm cô đọng dữ liệu học nguyên thủy, vì vậy các thao tác vào/ ra liên quan đến quá trình học sẽ giảm.

❖ **So sánh các mô hình phân lớp**

Trong từng ứng dụng cụ thể cần lựa chọn mô hình phân lớp phù hợp. Việc lựa chọn đó căn cứ vào sự so sánh các mô hình phân lớp với nhau, dựa trên các tiêu chuẩn sau:

• **Độ chính xác dự đoán** (predictive accuracy)

Độ chính xác là khả năng của mô hình để dự đoán chính xác nhãn lớp của dữ liệu mới hay dữ liệu chưa biết.

• **Tốc độ** (speed)

Tốc độ là những chi phí tính toán liên quan đến quá trình tạo ra và sử dụng mô hình.

• **Chắc chắn** (robustness)

Chắc chắn là khả năng mô hình tạo ta những dự đoán đúng từ những dữ liệu nhiễu hay dữ liệu với những giá trị thiếu.

• **Khả năng mở rộng** (scalability)

Khả năng mở rộng là khả năng thực thi hiệu quả trên lượng lớn dữ liệu của mô hình đã học.

• **Tính hiểu được** (interpretability)

Tính hiểu được là mức độ hiểu và hiểu rõ những kết quả sinh ra bởi mô hình đã học.

• **Tính đơn giản** (simplicity)

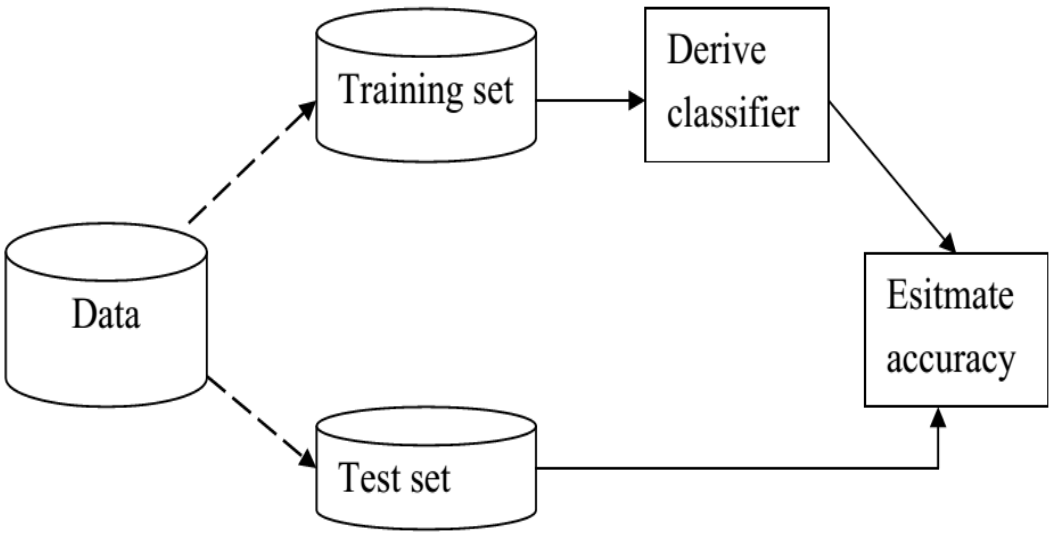
Tính đơn giản liên quan đến kích thước của cây quyết định hay độ cô đọng của các luật.

Trong các tiêu chuẩn trên, khả năng mở rộng của mô hình phân lớp được nhấn mạnh và trú trọng phát triển, đặc biệt với cây quyết định

2.1.2 Các phương pháp đánh giá độ chính xác của mô hình phân lớp

Ước lượng độ chính xác của bộ phân lớp là quan trọng ở chỗ nó cho phép dự đoán được độ chính xác của các kết quả phân lớp những dữ liệu tương lai. Độ chính xác còn giúp so sánh các mô hình phân lớp khác nhau. Ta đề cập đến hai phương pháp đánh giá phổ biến là holdoutvà k-fold cross-validation. Cả hai kỹ thuật này đều dựa trên các phân hoạch ngẫu nhiên tập dữ liệu ban đầu.

Trong phương pháp holdout, dữ liệu dưa ra được phân chia ngẫu nhiên thành 2 phần là: tập dữ liệu huấn luyện và tập dữ liệu kiểm tra. Thông thường 2/3 dữ liệu cấp cho tập dữ liệu huấn luyện, phần còn lại cho tập dữ liệu kiểm tra [18].



Hình 2.4 - Ước lượng độ chính xác của mô hình phân lớp với phương pháp holdout [1]

Trong phương pháp k-fold cross validation tập dữ liệu ban đầu được chia ngẫu nhiên thành ktập con (fold) có kích thước xấp xỉ nhau S1, S2, …, Sk. Quá trình học và test được thực hiện k lần. Tại lần lặp thứ i, Si là tập dữ liệu kiểm tra, các tập còn lại hợp thành tập dữ liệu huấn luyện. Có nghĩa là, đâu tiên việc dạy được thực hiện trên các tập S2, S3…, Sk , sau đó test trên tập S1; tiếp tục quá trình dạy được thực hiện trên tập S1, S3, S4,…, Sk , sau đó test trên tập S2; và cứ thế tiếp tục. Độ chính xác là toàn bộ số phân lớp đúng từ k lần lặp chia cho tổng số mẫu của tập dữ liệu ban đầu.

2.2 CÂY QUYẾT ĐỊNH

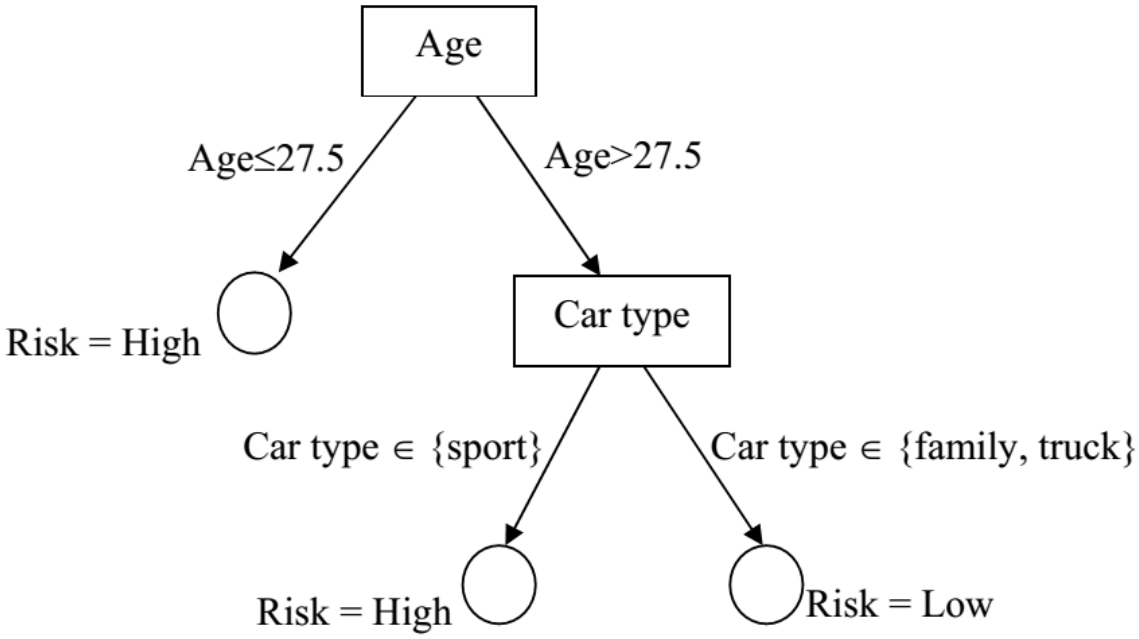
2.2.1 Cây quyết định

Trong các mô hình phân lớp đã được đề xuất, cây quyết định được coi là công cụ mạnh, phổ biến và đặc biệt thích hợp với các ứng dụng khai phá dữ liệu. Thuật toán phân lớp là nhân tố trung tâm trong một mô hình phân lớp. [25]

Việc xây dựng các cây quyết định chính là quá trình phát hiện ra các luật phân chia tập dữ liệu đã cho thành các lớp đã được định nghĩa trước. Trong thực tế, tập các cây quyết định có thể có đối với bài toán này rất lớn và rất khó có thể duyệt hết được một cách tường tận.

Trong mô hình phân lớp, thuật toán phân lớp giữ vai trò trung tâm, quyết định tới sự thành công của mô hình phân lớp. Do vậy chìa khóa của vấn đề phân lớp dữ liệu là tìm ra được một thuật toán phân lớp nhanh, hiệu quả, có độ chính xác cao và có khả năng mở rộng được

Cây quyết định là biểu đồ phát triển có cấu trúc dạng cây, như mô tả tron hình vẽ sau:



Hình 2.5 Ví dụ về cây quyết định [1]

Trong cây quyết định:

• Gốc: là nút trên cùng của cây

• Nút trong: biểu diễn một kiểm tra trên một thuộc tính đơn (hình chữ nhật)

• Nhánh: biểu diễn các kết quả của kiểm tra trên node trong (mũi tên)

• Nút lá: biểu diễn lớp hay sự phân phối lớp (hình tròn)

Để phân lớp mẫu dữ liệu chưa biết, giá trị các thuộc tính của mẫu được đưa vào kiểm tra trên cây quyết định. Mỗi mẫu tương ứng có một đường đi từ gốc đến lá và lá biểu diễn dự đoán giá trị phân lớp mẫu đó.

2.2.2 Các vấn đề trong khai phá dữ liệu sử dụng cây quyết định

Các vấn đề đặc thù trong khi học hay phân lớp dữ liệu bằng cây quyết định gồm: xác định độ sâu để phát triển cây quyết định, xử lý với những thuộc tính liên tục, chọn phép đo lựa chọn thuộc tính thích hợp, sử dụng tập dữ liệu huấn luyện với những giá trị thuộc tính bị thiếu, sử dụng các thuộc tính với những chi phí khác nhau, và cải thiện hiệu năng tính toán. Sau đây khóa luận sẽ đề cập đến những vấn

đề chính đã được giải quyết trong các thuật toán phân lớp dựa trên cây quyết định. [26]

Tránh "quá khớp" dữ liệu

Thế nào là "quá khớp" dữ liệu? Có thể hiểu đây là hiện tượng cây quyết định chứa một số đặc trưng riêng của tập dữ liệu huấn luyện, nếu lấy chính tập traning data để test lại mô hình phân lớp thì độ chính xác sẽ rất cao, trong khi đối với những dữ liệu tương lai khác nếu sử dụng cây đó lại không đạt được độ chính xác như vậy.

Quá khớp dữ liệu là một khó khăn đáng kể đối với học bằng cây quyết định và những phương pháp học khác. Đặc biệt khi số lượng ví dụ trong tập dữ liệu huấn luyện quá ít, hay có nhiễu trong dữ liệu.

Có hai phương pháp tránh "quá khớp" dữ liệu trong cây quyết định:

- Dừng phát triển cây sớm hơn bình thường, trước khi đạt tới điểm phân lớp hoàn hảo tập dữ liệu huấn luyện. Với phương pháp này, một thách thức đặt ra là phải ước lượng chính xác thời điểm dừng phát triển cây.
- Cho phép cây có thể"quá khớp" dữ liệu, sau đó sẽ cắt, tỉa cây.

Mặc dù phương pháp thứ nhất có vẻ trực tiếp hơn, nhưng với phương pháp thứ hai thì cây quyết định được sinh ra được thực nghiệm chứng minh là thành công hơn trong thực tế. Hơn nữa việc cắt tỉa cây quyết định còn giúp tổng quát hóa, và cải thiện độ chính xác của mô hình phân lớp. Dù thực hiện phương pháp nào thì vấn đề mấu chốt ở đây là tiêu chuẩn nào được sử dụng để xác định kích thước hợp lý của cây cuối cùng.

Thao tác với thuộc tính liên tục

Việc thao tác với thuộc tính liên tục trên cây quyết định hoàn toàn không đơn giản như với thuộc tính rời rạc.

Thuộc tính rời rạc có tập giá trị (domain) xác định từ trước và là tập hợp các giá trị rời rạc. Ví dụ loại ô tô là một thuộc tính rời rạc với tập giá trị là: {xe tải, xe khách, xe con, taxi}.Việc phân chia dữ liệu dựa vào phép kiểm tra giá trị của thuộc tính rời rạc được chọn tại một ví dụ cụ thể có thuộc tập giá trị của thuộc tính đó hay

không: value(A) ∈X vớ Xi ⊂domain(A). Đây là phép kiểm tra logic đơn giản, không tốn nhiều tài nguyên tính toán. Trong khi đó, với thuộc tính liên tục (thuộc tính dạng số) thì tập giá trị là không xác định trước. Chính vì vậy, trong quá trình phát triển cây, cần sửdụng kiểm tra dạng nhị phân: value(A) ≤ θ. Với θ là hằng số ngưỡng (threshold) được lần lượt xác định dựa trên từng giá trị riêng biệt hay từng cặp giá trị liền nhau (theo thứ tự đã sắp xếp) của thuộc tính liên tục đang xem xét trong tập dữ liệu huấn luyện. Điều đó có nghĩa là nếu thuộc tính liên tục A trong tập dữ liệu huấn luyện có giá trị phân biệt thì cần thực hiện d-1 lần kiểm tra value(A) ≤ θi với i = 1..d-1 để tìm ra ngưỡng θ tốt nhất tương ứng với thuộc tính đó. Việc xác định giá trị của θ và tiêu chuẩn tìm θ tốt nhất tùy vào chiến lược của từng thuật toán [13]. Trong thuật toán C4.5, θi được chọn là giá trị trung bình của hai giá trị liền kề nhau trong dãy giá trị đã sắp xếp.

Ngoài ra còn một số vấn đề liên quan đến sinh tập luật hay xử lý với giá trị thiếu, giá trị nhiễu.

2.2.3 Đánh giá cây quyết định trong lĩnh vực khai phá dữ liệu

❖ ***Điểm mạnh, lợi ích của cây quyết định***

Khả năng sinh ra các quy tắc hiểu được. Cây quyết định có khả năng sinh ra các quy tắc có thể chuyển đổi được sang dạng tiếng Anh, hoặc các câu lệnh SQL. Đây là ưu điểm nổi bật của kỹ thuật này.Thậm chí với những tập dữ liệu lớn khiến cho hình dáng cây quyết định lớn và phức tạp, việc đi theo bất cứ đường nào trên cây là dễ dàng theo nghĩa phổ biến và rõ ràng.

Do vậy sự giải thích cho bất cứ một sự phân lớp hay dự đoán nào đều tương đối minh bạch.

Khả năng thực thi trong những lĩnh vực hướng quy tắc. Điều này có nghe có vẻ hiển nhiên, nhưng quy tắc quy nạp nói chung và cây quyết định nói riêng là lựa chọn hoàn hảo cho những lĩnh vực thực sựlà các quy tắc. Rất nhiều lĩnh vực từ di truyền tới các quá trình công nghiệp thực sự chứa các quy tắc ẩn, không rõ ràng (underlying rules) do khá phức tạp và tối nghĩa bởi những dữ liệu nhiễu. Cây

quyết định là một sự lựa chọn tự nhiên khi chúng ta nghi ngờ sự tồn tại của các quy tắc ẩn, không rõ ràng.

Dễ dàng tính toán trong khi phân lớp. Mặc dù như chúng ta đã biết, cây quyết định có thể chứa nhiều định dạng, nhưng trong thực tế, các thuật toán sử dụng để tạo ra cây quyết định thường tạo ra những cây với số phân nhánh thấp và các test đơn giản tại từng node. Những test điển hình là: so sánh số, xem xét phần tửcủa một tập hợp, và các phép nối đơn giản. Khi thực thi trên máy tính, những test này chuyển thành các toán hàm logic và số nguyên là những toán hạng thực thi nhanh và không đắt. Đây là một ưu điểm quan trọng bởi trong môi trường thương mại, các mô hình dự đoán thường được sử dụng để phân lớp hàng triệu thậm trí hàng tỉ bản ghi.

Khả năng xử lý với cả thuộc tính liên tục và thuộc tính rời rạc. Cây quyết định xử lý "tốt" như nhau với thuộc tính liên tục và thuộc tính rời rạc. Tuy rằng với thuộc tính liên tục cần nhiều tài nguyên tính toán hơn. Những thuộc tính rời rạc đã từng gây ra những vấn đề với mạng neural và các kỹ thuật thống kê lại thực sự dễ dàng thao tác với các tiêu chuẩn phân chia (splitting criteria) trên cây quyết định: mỗi nhánh tương ứng với từng phân tách tập dữ liệu theo giá trị của thuộc tính được chọn để phát triển tại node đó. Các thuộc tính liên tục cũng dễ dàng phân chia bằng việc chọn ra một số gọi là ngưỡng trong tập các giá trị đã sắp xếp của thuộc tính đó. Sau khi chọn được ngưỡng tốt nhất, tập dữ liệu phân chia theo test nhị phân của ngưỡng đó.

Thể hiện rõ ràng những thuộc tính tốt nhất. Các thuật toán xây dựng cây quyết định đưa ra thuộc tính mà phân chia tốt nhất tập dữ liệu huấn luyện bắt đầu từ node gốc của cây. Từ đó có thể thấy những thuộc tính nào là quan trọng nhất cho việc dự đoán hay phân lớp.

- ***Điểm yếu của cây quyết định***

Dù có những sức mạnh nổi bật trên, cây quyết định vẫn không tránh khỏi có những điểm yếu. Đó là cây quyết định không thích hợp lắm với những bài toán với

mục tiêu là dự đoán giá trị của thuộc tính liên tục như thu nhập, huyết áp hay lãi xuất ngân hàng,… Cây quyết định cũng khó giải quyết với những dữ liệu thời gian liên tục nếu không bỏ ra nhiều công sức cho việc đặt ra sự biểu diễn dữ liệu theo các mẫu liên tục.

Dễ xảy ra lỗi khi có quá nhiều lớp. Một số cây quyết định chỉ thao tác với những lớp giá trị nhị phân dạng yes/no hay accept/reject. Số khác lại có thể chỉ định các bản ghi vào một số lớp bất kỳ, nhưng dễ xảy ra lỗi khi số ví dụ huấn luyện ứng với một lớp là nhỏ. Điều này xảy ra càng nhanh hơn với cây mà có nhiều tầng hay có nhiều nhánh trên một node.

Chi phí tính toán cao để huấn luyện. Điều này nghe có vẻ mâu thuẫn với khẳng định ưu điểm của cây quyết định ở trên. Nhưng quá trình phát triển cây quyết định đắt vềmặt tính toán. Vì cây quyết định có rất nhiều node trong trước khi đi đến lá cuối cùng. Tại từng node, cần tính một độ đo (hay tiêu chuẩn phân chia) trên từng thuộc tính, với thuộc tính liên tục phải thêm thao tác xắp xếp lại tập dữ liệu theo thứtựgiá trịcủa thuộc tính đó. Sau đó mới có thể chọn được một thuộc tính phát triển và tương ứng là một phân chia tốt nhất. Một vài thuật toán sử dụng tổ hợp các thuộc tính kết hợp với nhau có trọng số để phát triển cây quyết định. Quá trình cắt cụt cây cũng "đắt" vì nhiều cây con ứng cử phải được tạo ra và so sánh.

2.2.4 Xây dựng cây quyết định

Quá trình xây dựng cây quyết định gồm hai giai đoạn:

Giai đoạn thứ nhất phát triển cây quyết định: Giai đoạn này phát triển bắt đầu từgốc, đến từng nhánh và phát triển quy nạp theo cách thức chia đểtrịcho tới khi đạt được cây quyết định với tất cảcác lá được gán nhãn lớp.

Giai đoạn thứ hai cắt, tỉa bớt các cành nhánh trên cây quyết định. Giai đoạn này nhằm mục đích đơn giản hóa và khái quát hóa từ đó làm tăng độ chính xác của cây quyết định bằng cách loại bỏ sự phụ thuộc vào mức độ nhiễu của dữ liệu huấn luyện mang tính chất thống kê, hay những sự biến đổi mà có thể là đặc tính riêng biệt của dữ liệu huấn luyện.Giai đoạn này chỉ truy cập dữ liệu trên cây quyết định

đã được phát triển trong giai đoạn trước và quá trình thực nghiệm cho thấy giai đoạn này không tốn nhiều tài nguyên tính toán, như với phần lớn các thuật toán, giai đoạn này chiếm khoảng dưới 1% tổng thời gian xây dựng mô hình phân lớp [7]

Do vậy, ở đây chúng ta chỉtập trung vào nghiên cứu giai đoạn phát triển cây quyết định. Dưới đây là khung công việc của giai đoạn này:

1) Chọn thuộc tính "tốt" nhất bằng một độ đo đã định trước

2) Phát triển cây bằng việc thêm các nhánh tương ứng với từng giá trị của thuộc tính đã chọn

3) Sắp xếp, phân chia tập dữ liệu huấn luyện tới node con

4) Nếu các ví dụ được phân lớp rõ ràng thì dừng.

Ngược lại: lặp lại bước 1 tới bước 4 cho từng node con

2.3 CÁC THUẬT TOÁN XÂY DỰNG CÂY QUYẾT ĐỊNH

2.3.1 Tư tưởng chung

Phần lớn các thuật toán phân lớp dữ liệu dựa trên cây quyết định có mã giả như sau:

```
Make Tree (Training Data T)
{
        Partition(T)
}
Partition(Data S)
{
        if (all points in S are in the same class) then
                return
        for each attribute A do
                evaluate splits on attribute A;
        use best split found to partition S into S1, S2,..., Sk
        Partition(S1)
        Partition(S2)
        ...
        Partition(Sk)
}
```

Hình 2.6 Mã giả của thuật toán phân lớp dữ liệu dựa trên cây quyết định

Các thuật toán phân lớp như C4.5, CDP, SLIQ và SPRINT đều sử dụng phương pháp của Hunt làm tư tưởng chủ đạo. Phương pháp này được Hunt và các đồng sự nghĩ ra vào những năm cuối thập kỷ 50 đầu thập kỷ 60.

Mô tả quy nạp phương pháp Hunt:

Giả sử xây dựng cây quyết định từ T là tập training data và các lớp được biểu diễn dưới dạng tập C = {C1, C2, …,Ck}

Trường hợp 1: T chứa các trường hợp thuộc về một lớp đơn Cj, cây quyết định ứng với T là một lá tương ứng với lớp Cj

Trường hợp 2: T chứa các trường hợp thuộc về nhiều lớp khác nhau trong tập C. Một kiểm tra được chọn trên một thuộc tính có nhiều giá trị{O1,O2, ….,On}. Trong nhiều ứng dụng n thường được chọn là 2, khi đó tạo ra cây quyết định nhị phân. Tập T được chia thành các tập con T1, T2, …, Tn, với Ti chứa tất cả các trường hợp trong T mà có kết quả là Oi trong kiểm tra đã chọn. Cây quyết định ứng với T bao gồm một node biểu diễn kiểm tra được chọn, và mỗi nhánh tương ứng với

mỗi kết quả có thể của kiểm tra đó. Cách thức xây dựng cây tương tự được áp dụng đệ quy cho từng tập con của tập training data.

Trường hợp 3: T không chứa trường hợp nào nào. Cây quyết định ứng với T là một lá, nhưng lớp gắn với lá đó phải được xác định từ những thông tin khác ngoài T. Ví dụ C4.5 chọn giá trị phân lớp là lớp phổ biến nhất tại cha của node này [26].

2.3.2 Thuật toán ID3

Thuật toán ID3 được phát biểu bởi Quinlan (trường đại học Syney, Australia) và được công bố vào cuối thập niên 70 của thế kỷ 20. Sau đó, thuật toán ID3 được giới thiệu và trình bày trong mục phần giới thiệu về cây quyết định và máy học năm 1986. với khả năng lựa chọn thuộc tính tốt nhất để tiếp tục triển khai cây tại mỗi bước. ID3 là một trong những thuật toán xây dựng cây quyết định sử dụng information gain để lựa chọn thuộc tính phân lớp các đối tượng. [25] ID3 xây dựng cây quyết định từ trên- xuống (top -down). Nó xây dựng cây theo cách từ trên xuống, bắt đầu từ một tập các đối tượng và một đặc tả của các thuộc tính. Tại mỗi đỉnh của cây, một thuộc tính có information gain lớn nhất sẽ được chọn để phân chia tập đối tượng. Quá trình này được thực hiện một cách đệ qui cho đến khi một tập đối tượng tại một cây con đã cho trở nên thuần nhất, tức là nó chỉ chứa các đối tượng thuộc về cùng một lớp. Lớp này sẽ trở thành một lá của cây. Việc lựa chọn một thuộc tính nào cho phép thử là rất quan trọng. Nếu chọn không thích hợp, chúng ta có thể có một cây rất phức tạp [1].

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

Entropy: Dùng để đo tính thuần nhất của một tập dữ liệu. Entropy của một tập S được tính theo công thức

$$\text{Entropy}(S) = - P^{+} \log_2 (P^{+}) \, P^{-} \log_2 (P^{-})$$

Trong trường hợp các mẫu dữ liệu có hai thuộc tính phân lớp "yes" (+), "no" (-). Ký hiệu P^+ là để chỉ tỷ lệ các mẫu có giá trị của thuộc tính quyết định là "yes", và P^- là tỷ lệ các mẫu có giá trị của thuộc tính quyết định là "no" trong tập S.

Trường hợp tổng quát, đối với tập con S có n phân lớp thì ta có công thức sau:

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

Trong đó P_i là tỷ lệ các mẫu thuộc lớp i trên tập hợp S các mẫu kiểm tra.

Các trường hợp đặc biệt

- Nếu tất cả các mẫu thành viên trong tập S đều thuộc cùng một lớp thì Entropy(S) = 0

- Nếu trong tập S có số mẫu phân bổ đều nhau vào các lớp thì Entropy(S) =1

- Các trường hợp còn lại 0< Entropy(S)<1

Information Gain (viết tắt là Gain): Gain là đại lượng dùng để đo tính hiệu quả của một thuộc tính được lựa chọn cho việc phân lớp. Đại lượng này được tính thông qua hai giá trị Information và Entropy [26].

Cho tập dữ liệu S gồm có n thuộc tính Ai (i=1,2…n) giá trị Information của thuộc tính A i ký hiệu là Information (Ai) được xác định bởi công thức

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

Giá trị Gain của thuộc tính A trong tập S ký hiệu là Gain(S,A) và được tính theo công thức sau:

$$Gain(S, A) = \text{Entropy(S)} - \sum_{\text{V} \notin \text{value(A)}} \frac{|S_V|}{|S|} \text{Entropy}(\text{S}_\text{V})$$

Trong đó :

S là tập hợp ban đầu với thuộc tính A. Các giá trị của v tương ứng là các giá trị của thuộc tính A.

S_v bằng tập hợp con của tập S mà có thuộc tính A mang giá trị v

$|S_v|$ là số phần tử của tập S_v

|S| là số phần tử của tập S.

Trong quá trình xây dựng cây quyết định theo thuật toán ID3 tại mỗi bước triển khai cây, thuộc tính được chọn để triển khai là thuộc tính có giá trị Gain lớn nhất.

Thuật toán xây dựng cây quyết định trong thuật toán ID3

Function induce_tree*(tập_ví_dụ, tập_thuộc_tính)*

begin

if mọi ví dụ trong tập_ví_dụ đều nằm trong cùng một lớp then

return một nút lá được gán nhãn bởi lớp đó

else if tập_thuộc_tính là rỗng then

return nút lá được gán nhãn bởi tuyển của tất cả các lớp trong tập_ví_dụ

else begin

chọn một thuộc tính P, lấy nó làm gốc cho cây hiện tại;

xóa P ra khỏi tập_thuộc_tính;

với mỗi giá trị V của P

begin

tạo một nhánh của cây gán nhãn V;

Đặt vào phân_vùng V các ví dụ trong tập_ví_dụ có giá trị V tại thuộc tính P;

Gọi induce_tree(phân_vùng V, tập_thuộc_tính), gắn kết quả vào nhánh V

End

End

End

Với việc tính toán giá trị Gain để lựa chọn thuộc tính tối ưu cho việc triển khai cây, thuật toán ID3 được xem là một cải tiến của thuật toán CLS. Tuy nhiên thuật toán ID3 không có khả năng xử lý đối với những dữ liệu có chứa thuộc tính số - thuộc tính liên tục (numeric attribute) và khó khăn trong việc xử lý các dữ liệu thiếu (missing data) và dữ liệu nhiễu.

2.3.3 Thuật toán C4.5

Thuật toán C4.5 là một thuật toán được cải tiến từ thuật toán ID3 với việc cho phép xử lý trên tập dữ liệu có các thuộc tính số (numeric atributes) và và làm việc được với tập dữ liệu bị thiếu và bị nhiễu. Nó thực hiện phân lớp tập mẫu dữ liệu theo chiến lược ưu tiên theo chiều sâu (Depth -First). Thuật toán xét tất cả các phép thử có thể để phân chia tập dữ liệu đã cho và chọn ra một phép thử có giá trị GainRatio tốt nhất. GainRatio là một đại lượng để đánh giá độ hiệu quả của thuộc tính dùng để thực phép tách trong thuật toán để phát triển cây quyết định. GainRatio được tính dựa trên kết quả tính toán đại lượng Information Gain theo công thức sau [26]:

$$GainRation(X,T) = \frac{Gain(X,T)}{SplitInfo(X,T)}$$

Với

$$Split\text{info}(X,T) = -\sum_{\text{V} \notin \text{Value(X)}} \frac{|\text{T}_i|}{|T|} log_2 \frac{|\text{T}_i|}{|T|}$$

Trong đó:

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

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

Đối với các thuộc tính liên tục, chúng ta tiến hành phép thử nhị phân cho mọi giá trị của thuộc tính đó. Để thu thập được giá trị Entropy gain của tất cả các phép thử nhị phân một cách hữu hiệu ta tiến hành xắp xếp các dữ liệu theo giá trị của thuộc tính liên tục đó bằng thuật toán Quicksort

Thuật toán xây dựng cây quyết định C4.5

Function *xay_dung_cay*(T)

{

Tính toán tần xuất các giá trị trong các lớp của T;

If *(nếu các mẫu thuộc cùng một lớp hoặc có rất ít mẫu khác lớp)* Then

Trả về 1 nút lá

Else *Tạo một nút quyết định N;*

For *Với mỗi thuộc tính A* Do *Tính giá trị Gain(A);*

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

If (*N.test là thuộc tính liên tục)* Then *Tìm ngưỡng cho phép tách của N.test*;

For *(Với mỗi tập con T` được tách ra từ tập T)* Do

(T` được tách ra theo quy tắc:

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

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

)

If (*Kiểm tra, nếu T' rỗng)* Then

Gán nút con này của nút N là nút lá;

Else

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

}

Tính toán các lỗi của nút N;

Trả về nút N;

}

Một số công thức được sử dụng

$$\text{Info}_x(\text{T}) = -\sum_{i=1}^{n} \frac{|\text{T}_i|}{|T|} * Info(\text{T}_i)$$

$$\text{Gain (X)} = \text{Info(T)} - \text{Info}_x\text{(T)} \qquad (2.3)$$

Công thức (2.3) được sử dụng làm tiêu chuẩn để lựa chọn thuộc tính khi phân lớp. Thuộc tính được chọn là thuộc tính có giá trị Gain tính theo (2.3) đạt giá trị lớn nhất.

CHƯƠNG 3
SỬ DỤNG CÂY QUYẾT ĐỊNH ĐỂ PHÂN LOẠI DỮ LIỆU NHIỄU

3.1 GIỚI THIỆU

Thuật toán C4.5 đã được sử dụng rộng rãi để thiết kế cây quyết định. Tuy nhiên nó chưa thực sự hiệu quả trong việc phân loại dữ liệu nhiễu. Luận văn đưa ra một thuật toán gọi là NC4.5, một bản điều chỉnh của C4.5. Phương pháp này sử dụng một lý thuyết dựa trên xác suất không chính xác (imprecise probability) và đo lường không chắc chắn (uncertainty measures). Nó sử dụng một tiêu chuẩn phân nhanh mới gọi là Tỉ Số Thu Thập Thông Tin Không Chắc Chắn (Imprecise Information Gain Ratio), áp dụng đo lường không chắc chắn dựa trên các tập lồi của phân phố xác suất (các tập credal) [3]. Theo khía cạnh này, NC4.5 xây dựng những cây để gải quyết các vấn đề phân loại với giả thiết rằng các tập huấn luyện không hoàn toàn đáng tin cậy. Luận văn đã có nhiều thực nghiệm để so sánh thủ tục mới này với các thủ tục khác và đưa ra kết luận sau: trong lĩnh vực phân loại dữ liệu nhiễu, NC4.5 đạt được những cây nhỏ hơn, thực thi tốt hơn C4.5 cổ điển.

Cho một tập dữ liệu nhiễu được tạo bởi 15 thực thể, trong đó 9 thực thể thuộc lớp A và 6 thực thể lớp B. Tập dữ liệu này có 2 thuộc tính nhị phân là X_1 và X_2. Theo giá trị của các thuộc tính, các thực thể được tổ chức như sau:

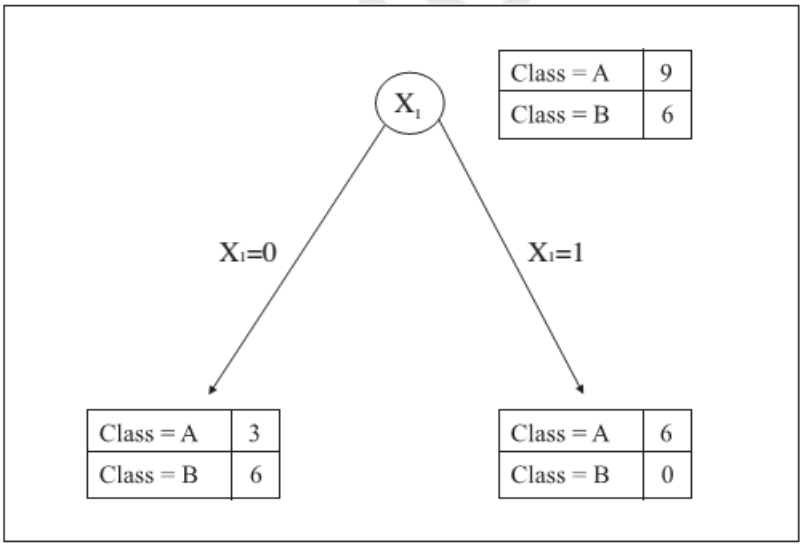
$X_1 = 0 \rightarrow$ (3 thực thể của lớp A; 6 thực thể của lớp B)

$X_1 = 1 \rightarrow$ (6 thực thể của lớp A; 0 thực thể của lớp B)

$X_2 = 0 \rightarrow$ (1 thực thể của lớp A; 5 thực thể của lớp B)

$X_2 = 1 \rightarrow$ (8 thực thể của lớp A; 1 thực thể của lớp B)

Nếu tập dữ liệu này được tìm thấy ở một nốt của một cây. Thuật toán C4.5 sẽ chọn thuộc tính X_1 để phân nhánh cho nốt này (hình 3.1).



Hình 3.1 Sự phân nhánh của một nốt dữ liệu nhiễu được thực hiện bởi C4.5

Chúng ta có thể giả định rằng tập huấn luyện bị nhiễu vì nó chứa một điểm ngoại lai khi X_2=1 và lớp thuộc lớp B (Ví dụ: trời mưa mà lại thi đấu). Theo cách này, quá trình làm sạch dữ liệu sẽ tạo thành 10 thực thể của lớp A và 5 thực thể của lớp B, được tổ chức như sau:

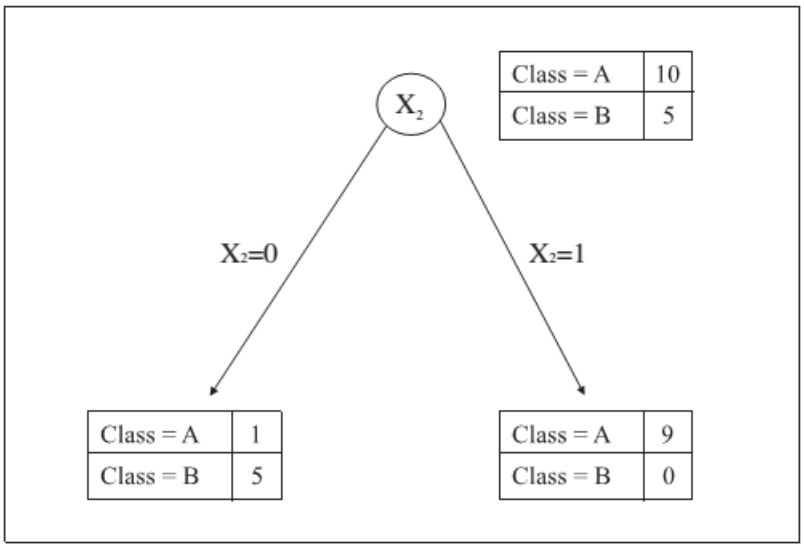
$X_1 = 0$ → (4 thực thể của lớp A; 5 thực thể của lớp B)

$X_1 = 1$ → (6 thực thể của lớp A; 0 thực thể của lớp B)

$X_2 = 0$ → (1 thực thể của lớp A; 5 thực thể của lớp B)

$X_2 = 1$ → (9 thực thể của lớp A; 0 thực thể của lớp B)

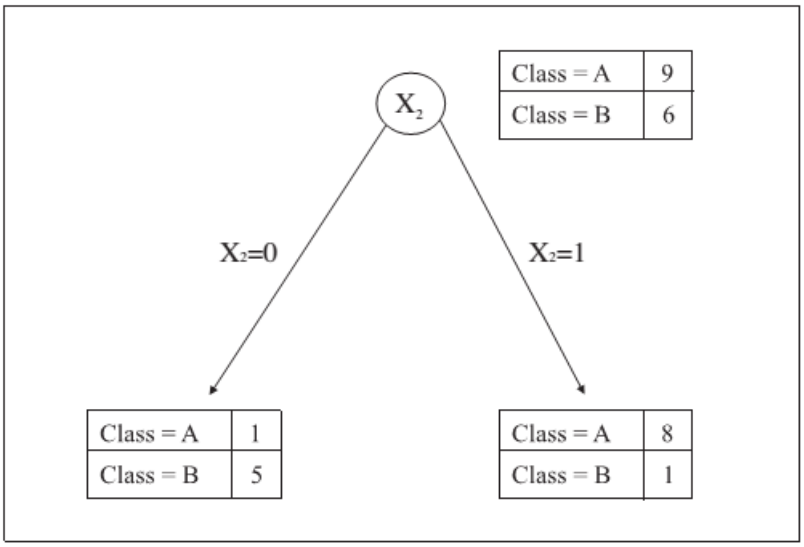
Nếu tập dữ liệu này được tìm thấy ở một nốt của cây, khi đó thuật toán C4.5 sẽ chọn thuộc tính X_2 để phân nhánh (hình 3.2).



Hình 3.2 Sự phân nhánh của một nốt dữ liệu sạch được thực hiện bởi C4.5

Chúng ta có thể thấy thuật toán C4.5 sinh ra một cây con không đúng khi nó xử lý dữ liệu nhiễu, bởi vì nó cho rằng tập huấn luyện là tin cậy. Sau đó, quá trình tỉa cây cho rằng tập huấn luyện không đáng tin cậy để giải quyết vấn đề này. Tuy nhiên, quá trình tỉa cây này chỉ có thể xóa cây con không chính xác đã được sinh ra. Nó không thể điều chỉnh cây con không chính xác này như thể hiện trong hình 3.2. Lý tưởng là việc ta phân nhánh như hình 3.2, sau đó thực hiện quá trình tỉa cây. Quá trình này có thể thực hiện bằng sử dụng cây quyết định dựa trên xác suất không chính xác (imprecise).

Cây quyết định Credal coi rằng tập huấn luyện là không đáng tin cậy khi quá chính lựa chọn thuộc tính được diễn ra. Do vậy vấn đề trên được giải quyết. Nếu tập dữ liệu nhiễu xuất hiện trong một cây creedal, thì sau đó thuộc tính X_2 được lựa chọn để phân nhánh (hình 3.3). Do vậy nếu chúng ta thiết kế một thuật toán cây quyết định creedal cải tiến từ C4.5, thì chúng ta có được thuật toán xem xét tập huấn luyện không tin cậy khi quá trình tỉa cây và lựa chọn thuộc tính diễn ra.



Hình 3.3 Sự phân nhánh của một nốt dữ liệu nhiễu được thực hiện bởi cây quyết định credal

3.2 CÂY QUYẾT ĐỊNH CREDAL

Tiêu chuẩn phân nhánh được đưa vào để xây dựng cây quyết định credal [2] dựa trên xác suất không chính xác và áp dụng đo lường không chắc chắn trên các tập credal. Cơ sở của phương pháp này được mô tả như sau. [4]

Cho một biến Z mà giá trị của nó thuộc về {z1,…,zk}. Ta giả định rằng một phân phối xác suất p(zj), j=1,…,k định nghĩa cho mỗi giá trị zj từ một tập dữ liệu.

Lý thuyết Imprecise Dirichlet Model (IDM) của Walley được sử dụng để ước lượng khoảng cách xác suất từ tậ dữ liệu cho mỗi giá trị của biến Z. IDM ước lượng xác suất của mỗi giá trị biến zj trong khoảng:[4]

$$p(z_j) \in \left[\frac{n_{z_j}}{N+s}, \frac{n_{z_j}+s}{N+s}\right], j = 1, \dots, k;$$

Với n_{zj} là độ phổ biến của tập giá trị ($Z=z_j$) trong tập dữ liệu, N là kích thước mẫu và s là một siêu tham số mà nó không phụ thuộc vào không gian mẫu. Giá trị của tham số s này quyết định tốc độ tại những giá trị của xác suất lớn hơn hay nhỏ hơn trung bình khi kích thước mẫu tăng lên. Các giá trị lớn hơn của s có thể đưa ra

nhiều suy luận hơn. Walley không đưa ra một đề xuất cụ thể nhưng ông gợi ý hai ứng viên: s=1 hoặc s=2.

Một thuộc tính quan trọng của mô hình này là các khoảng cách sẽ rộng hơn nếu kích thước mẫu sẽ nhỏ hơn. Vì vậy phương pháp này đưa ra các khoảng cách chính xác hơn khi mà N tăng lên.

Vấn đề này đưa ra một loại cụ thể của tập lồi của các phân phối xác suất trên biến Z, K(Z). Tập này được định nghĩa như sau:

$$K(Z) = \left\{p|p(z_j) \in \left[\frac{n_{z_j}}{N+s}, \frac{n_{z_j}+s}{N+s}\right], j = 1, \dots, k\right\}. \quad (3)$$

Trên loại này của các tập credal, đo lường không chắc chắn có thể được áp dụng. Thủ tục đẻ xây dựng các CDT sử dụng hàn entropy trên định nghĩa tập credal, một đo lường không chắc chắn được thiết lập trên các tập credal. Hàm này được gọi là H^* [7]:

$$H*(K(Z)) = max\{H(p)|p \in K(Z)\} \quad (4)$$

Với hàm H là hàm entropy của Shannon

H^* là một đo lường rời rạc của thông tin kết hợp của hai thành phần [9]:

a) Một đo lường xung đột hoặc ẩu mà thể hiện sự sắp xếp của các mẫu của mỗi lớp trong tập huấn luyện. Đo lường này liên quan đến entropy của xác suất trong tập lồi.
b) Một đo lường không xác định thể hiện sự không chắc chắn từ kích thước của tập huấn luyện. Đo lường này liên quan đến kích thước của tập lồi.

Thủ tục để tính H^* có chi phí thấp với s ϵ (0,2) (theo Abellán & Moral (2006)). Thủ tục để IDM đạt giá trị thấp nhất với s=1. Vì lý do này, ta sử dụng giá trị s=1 trong phần thực nghiệm.

Đầu tiên thủ tục này được dùng trong việc quyết định tập:

$$A = \left\{ z_j \middle| n_{z_j} = min_i\{n_{z_i}\} \right\} \quad (5)$$

Khi đó sự phân phối với entropy lớn nhất là

$$p^*(z_i) = \begin{cases} \frac{n_{z_i}}{N+s} \; if \; z_i \notin A \\ \frac{n_{z_i}+s/l}{N+s} \; if \; z_i \in A \end{cases} ; i = 1, \dots, k;$$

Với l là số thành phần của A.

3.3 THUẬT TOÁN NC4.5

Phương pháp để xây dựng các cây NC4.5 tương tự như thuật toán C4.5. Sự khác biệt chính là NC4.5 ước lượng xác suất giá trị của các tính năng và biến phân lớp bằng cách sử dụng xác suất không chính xác. Như trong thủ tục CDT, một đo lường không chắc chắn trên các tập credal được sử dụng để định nghĩa một tiêu chuẩn phân nhánh mới. Theo cách này, NC4.5 xem tập huấn luyện là không thực sự đáng tin cậy vì nó có thể bị ảnh hưởng bởi nhiễu thuộc tính hoặc nhiễu lớp [14].

NC4.5 được tạo ra bằng cách thay thế tiêu chuẩn phân nhánh Info-Gain Ratio từ C4.5 bằng Impercise Info-Gain Ratio (IIGR).

Trong bài toán phân lớp, cho C là biến lớp, $\{X_1, X_2, \dots, X_m\}$ là tập các thuộc tính và X là một thuộc tính; khi đó [3]:

$$IIGR^{\mathfrak{D}}(C, X) = \frac{IIG^{\mathfrak{D}}(C, X)}{H(X)}$$

Trong đó Imprecise Info-Gain (IIG) được tính như sau:

$$IIG^{\mathfrak{D}}(C, X) = \mathrm{H} * \left(K^{\mathfrak{D}}(\mathrm{C})\right) - \sum_i P^{\mathfrak{D}}(X = x_i) H^*(K^{\mathfrak{D}}(C|X = x_i))$$

Với $K^D(C)$ và $K^D(C|X = x_i)$ là các tập credal thu được thông qua IDM cho các biến C và $(C|X = x_i)$ tương ứng cho một phân mảnh D của tập dữ liệu; $P^D(X=x^i)(i=1,\dots,n)$ là một phân phối xác suất thuộc về tập credal $K^D(X)$.

Chúng ta chọn phân phối xác suất P^D từ $K^D(X)$ mà tối đa hóa biểu thức sau:

$$\sum_i P\,(X = x_i) H(C|X = x_i)).$$

Không phức tạp để tính phân phối xác suất này. Cho x_{j0} là một giá trị của X như vậy H(C|X=x_i) là tối đa. Khi đó phân phố xác suất P^D sẽ là

$$P^{\mathfrak{D}}(x_i) = \begin{cases} \frac{n_{x_i}}{N+s} \; if \; i \neq j_0 \\ \frac{n_{x_i}+s}{N+s} \; if \; i = j_0 \end{cases} . (9)$$

Tiêu chuẩn IIGR khác biệt so với tiêu chuẩn phân loại thường dùng. Nó dựa trên lý thuyết của việc tối đa hóa sự không chắc chắn, thường được sử dụng trong lý thuyết thông tin cổ điển, nơi mà dựa trên việc tối đa hóa entropy [6]. Lý thuyết này thể hiện rằng sự phân phối xác suất với việc tối đa entropy.

Mỗi nút trong một cây quyết định tạo ra một phân nhánh của tập dữ liệu (Nút gốc D được xem như toàn bộ tập dữ liệu). Hơn nữa, mỗi nút No có một kết hợp với danh sách £ nhãn thuộc tính (không ở trong đường dẫn từ nút gốc tới nút No).

Thủ tục NC4.5 được mô tả như sau:

Procedure BuilNC4.5Tree(No, £)

1. If £ = Θ then exit
2. Cho D là một phần kết hợp với nốt No
3. If |D| < số lượng thể hiện tối thiểu, then exit
4. Tính P^D (X = x_i) (i=1,…,n) trên tập lồi K^D (X)
5. Tính giá trị

   α=MAX $x_{j \epsilon M}$ {$IIGR^D(C, X_j)$ }

   với M= {$X_j \epsilon$ £ / IIG^D (C,X_j) >avg $x_{j \epsilon £}${$IIG^D(X)$ }}

6. If α <=0 then Exit
7. Else
8. Cho X_i là một biến có α lớn nhất
9. Loại bỏ X_i khỏi £
10. Gán X_i cho nốt No
11. For each giá trị x_i của X_i

12. Thêm một nốt No_i
13. Gắn No_i là nốt lá của No
14. Gọi thủ tục BuilNC4.5Tree(No_i, £)

❖ **Ý tưởng chính của thủ tục:**

Tiêu chuẩn phân nhánh: Imprecise Info-Gain Ratio (IIGR) được sử dụng.

Gán nhãn các nốt lá: giá trị có thể xảy ra nhất của biến lớp trong phân vùng liên kết với một nốt lá được thêm vào như là một nhãn. Do vậy, nhãn của lớp cho nốt lá No kết hớp với phân vùng D là:

$$Class(No, \mathfrak{D}) = \max_{c_i \in C} |\{I_j \in \mathfrak{D}/class(I_j) = c_i, j = 1, \dots |\mathfrak{D}|\}|$$

Với class(I_j) là lớp của thể hiện Ij ∈ D và |D| là số lượng thể hiện trong D

Điều kiện dừng: Sự phân nhánh của cây quyết định sẽ dừng khi sự đo lường không chắc chắn không giảm (α<=0, bước 6) hoặc khi không còn thuộc tính nào được thêm vào cây (£=Θ, bước 1) hoặc số lượng tối thiểu các nốt ở mỗi lá không đủ (bước 3).

Sự phân nhánh cũng dừng khi không có thuộc tính phân nhánh nào sử dụng "tiêu chuẩn phân nhánh" kể trên như trong C4.5 truyền thống.

Xử lý các thuộc tính số: tương tự như C4.5 nhưng sử dụng IIG thay cho đo lường IG

Đối phó với việc thiếu giá trị: tương tự như C4.5 nhưng sử dụng IIG thay cho đo lường IG

Xử lý sau cắt tỉa: tương tự như C4.5

CHƯƠNG 4
THỰC NGHIỆM – ĐÁNH GIÁ KẾT QUẢ

Để đo tính hiệu quả của thuật toán NC4.5 so sánh với các thuật toán khác, thí nghiệm được thực hiện trên các bộ dữ liệu chuẩn lấy về từ địa chỉ http://archive.ics.uci.edu/ml/datasets.html.

Máy tính thực hiện: Lenovo Thinkpad Twist, bộ xử lý Core i5-3317U 1.7 GHz và 4 GB bộ nhớ chính, chạy trên hệ điều hành Windows 8 Pro 64 bit, ngôn ngữ lập trình Java.

Hình 4.1 Giao diện chương trình

Choose an algorithm: chọn lựa thuật toán

Choose input file: Chọn file CSDL được chuyển về dạng *.txt

Set output file: Chọn file lưu kết quả của thuật toán

Open output file when the algorithm terminates: Mở file chứa kết quả khi thuật toán thực hiện xong.

Run algorithm: Thi hành thuật toán được chọn

Khi thực hiện xong, thuật toán sẽ hiển thị các kết quả được phân tích trong phần đánh giá thực nghiệm.

4.1 BỘ DỮ LIỆU

Bảng 4.1 Liệt kê đặc tính của các bộ dữ liệu thực nghiệm

Data set	N	Feat	Num	Nom	k	Range
Anneal	898	38	6	32	6	2–10
Arrhythmia	452	279	206	73	16	2
Audiology	226	69	0	69	24	2–6
Autos	205	25	15	10	7	2–22
Balance-scale	625	4	4	0	3	–
Breast-cancer	286	9	0	9	2	2–13
Wisconsin-breast-cancer	699	9	9	0	2	–
Car	1728	6	0	6	4	3–4
CMC	1473	9	2	7	3	2–4
Horse-colic	368	22	7	15	2	2–6
Credit-rating	690	15	6	9	2	2–14
German-credit	1000	20	7	13	2	2–11
Dermatology	366	34	1	33	6	2–4
Pima-diabetes	768	8	8	0	2	–
Ecoli	366	7	7	0	7	–
Glass	214	9	9	0	7	–
Haberman	306	3	2	1	2	12
Cleveland-14-heart-disease	303	13	6	7	5	2–14
Hungarian-14-heart-disease	294	13	6	7	5	2–14
Heart-statlog	270	13	13	0	2	–
Hepatitis	155	19	4	15	2	2
Hypothyroid	3772	30	7	23	4	2–4
Ionosphere	351	35	35	0	2	–
Iris	150	4	4	0	3	–
kr-vs-kp	3196	36	0	36	2	2–3
Letter	20000	16	16	0	26	–
Liver-disorders	345	6	6	0	2	–
Lymphography	146	18	3	15	4	2–8
mfeat-pixel	2000	240	0	240	10	4–6
Nursery	12960	8	0	8	4	2–4
Optdigits	5620	64	64	0	10	–
Page-blocks	5473	10	10	0	5	–
Pendigits	10992	16	16	0	10	–
Primary-tumor	339	17	0	17	21	2–3
Segment	2310	19	16	0	7	–
Sick	3772	29	7	22	2	2

Solar-flare2	1066	12	0	6	3	2–8
Sonar	208	60	60	0	2	–
Soybean	683	35	0	35	19	2–7
Spambase	4601	57	57	0	2	–
Spectrometer	531	101	100	1	48	4
Splice	3190	60	0	60	3	4–6
Sponge	76	44	0	44	3	2–9
Tae	151	5	3	2	3	2
Vehicle	946	18	18	0	4	–
Vote	435	16	0	16	2	2
Vowel	990	11	10	1	11	2
Waveform	5000	40	40	0	3	–
Wine	178	13	13	0	3	–
Zoo	101	16	1	16	7	2

Mô tả các ký hiệu trong bảng liệt kê tập dữ liệu: cột "N" là số lượng thực thể trong tập dữ liệu, cột "Feat" là số lương đặc tính hoặc biến thuộc tính, cột "Num" là số lượng các biến dạng số, cột "Nom" là số lượng các biến dạng tên, cột "k" là số lượng các trường hợp hoặc trạng thái của biến các lớp và cột "Range" là số lượng các trạng thái của các biến tên trong mỗi tập dữ liệu.

4.2 ĐÁNH GIÁ THỰC NGHIỆM

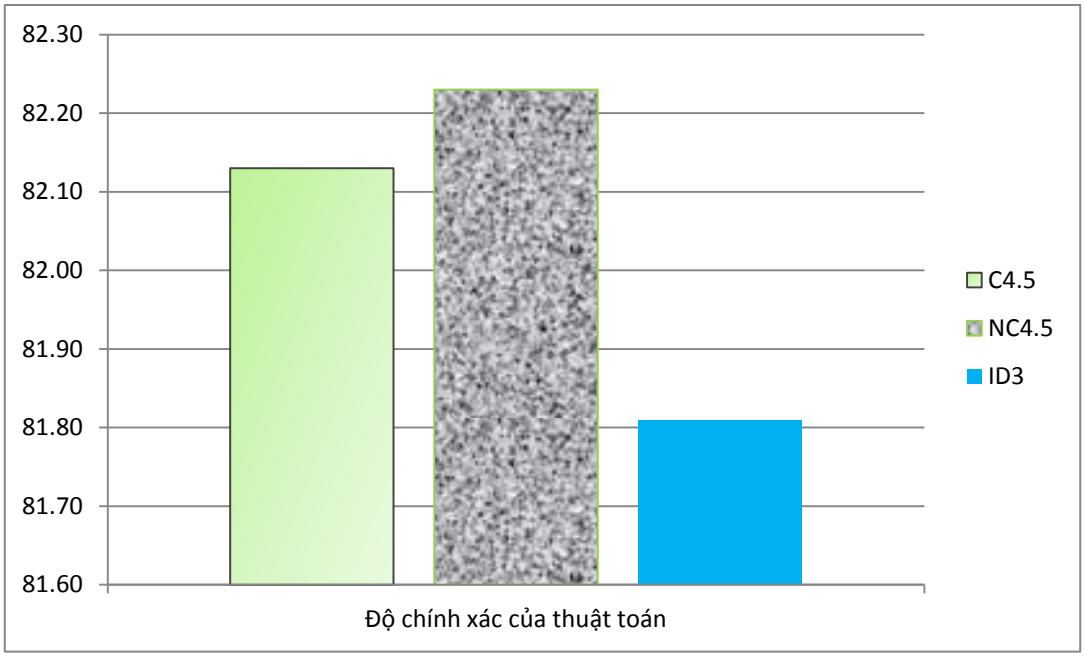
Các bảng dưới đây mô tả kết quả thực nghiệm khi so sánh NC4.5 với các thuật toán khác.

Bảng 4.2 Kết quả về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%

Dataset	**C4.5**	**NC4.5**	**ID3**
Anneal	98.6	98.19	99
Arrhythmia	64.1	67.55	64.9
Audiology	76.5	78.58	76
Autos	82.4	74.52	77.5
Balance-scale	79.4	78	79.5
Breast-cancer	68.2	71.44	68.3
Wisconsin-breast-cancer	94.4	95.08	94.6
Car	93.7	91.42	94
CMC	49.2	52.01	49.6
Horse-colic	82.1	84.64	82.9

Credit-rating	82.2	85.46	81.3
German-credit	68.1	70.17	69.7
Dermatology	94	93.82	92.2
Pima-diabetes	73.9	73.19	73.8
Ecoli	82.5	81.9	83.1
Glass	67.8	63.66	67.9
Haberman	70.5	73.89	70.6
Cleveland-14-heart-disease	76.4	76.6	78.6
Hungarian-14-heart-disease	78.6	82.54	76.1
Heart-statlog	76.8	80.04	77.9
Hepatitis	78.6	79.84	78.8
Hypothyroid	99.5	99.53	99.6
Ionosphere	89.83	88.35	88.15
Iris	94.8	94.73	94.8
kr-vs-kp	99.44	99.4	99.42
Letter	88.02	87.57	88
Liver-disorders	65.37	64.18	65.75
Lymphography	75.42	78.51	73.42
mfeat-pixel	78.42	79.58	75.66
Nursery	98.69	96.3	98.64
Optdigits	90.48	90.77	91.1
Page-blocks	96.78	96.72	96.92
Pendigits	96.54	96.39	96.39
Primary-tumor	42.6	42.19	38.59
Segment	96.8	96.04	96.77
Sick	98.77	98.77	98.82
Solar-flare2	99.49	99.53	99.38
Sonar	73.42	71.47	73.53
Soybean	90.69	92.5	86.76
Spambase	92.42	92.61	92.83
Spectrometer	47.31	45.52	43.49
Splice	92.16	93.81	91.37
Sponge	91.68	94.11	92.7
Tae	58.6	53.2	58.21
Vehicle	72.18	72.84	72.67
Vote	95.76	96.04	95.56
Vowel	81.63	77.87	84.09
Waveform	75.12	76.05	75.7
Wine	93.2	92.13	93.83
Zoo	93.41	92.42	92.01
Trung bình	82.13	82.23	81.81

Từ bảng số liệu này ta thấy nhìn chung độ chính xác của NC4.5 so với C4.5 và ID3 khi áp dụng với dữ liệu không nhiễu không hơn được bao nhiêu. Trong trường hợp này NC4.5 không phát huy được ưu điểm của mình.



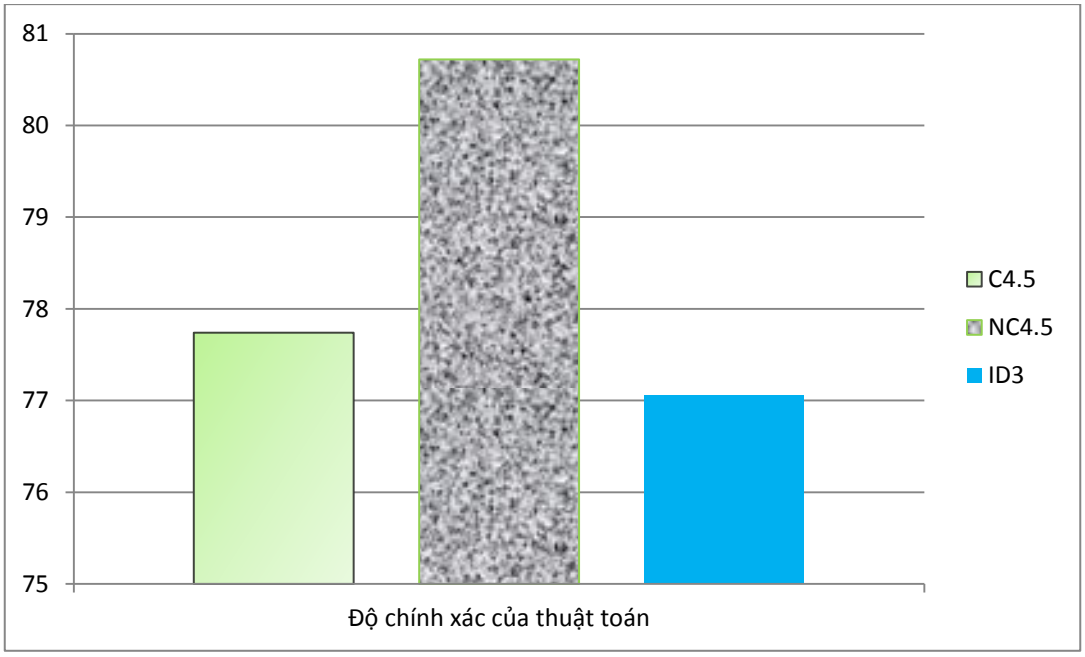
Hình 4.2 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%

Bảng 4.3 Kết quả về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 10%

Dataset	**C4.5**	**NC4.5**	**ID3**
Anneal	96.05	97.84	96.31
Arrhythmia	59.81	64.77	57.65
Audiology	75.06	76.98	71.43
Autos	74.73	71.57	68.88
Balance-scale	76.56	78.46	76.35
Breast-cancer	66.31	68.68	64.85
Wisconsin-breast-cancer	92	94.29	92.2
Car	88.8	90.9	88.47
CMC	47.25	50.07	47.3
Horse-colic	79.74	83.52	78.93
Credit-rating	76.8	85.04	75.8
German-credit	65.23	69.01	65.94
Dermatology	88	92.66	85.73
Pima-diabetes	72.04	73.58	72.2
Ecoli	77.77	81.52	77.89
Glass	64.07	65.29	63.4
Haberman	68.86	73.4	69.02
Cleveland-14-heart-disease	72.93	76.31	73.83
Hungarian-14-heart-disease	75.73	80.73	73.73
Heart-statlog	72.33	77.67	72.48
Hepatitis	73.84	78.17	75.27
Hypothyroid	95.55	99.38	95.74
Ionosphere	86.3	87.3	85.31
Iris	89.73	93.6	89.2
kr-vs-kp	93.51	98.04	93.05
Letter	85.01	86.27	84.45
Liver-disorders	62.15	61.11	62.27
Lymphography	71.31	74.1	69.59
mfeat-pixel	71.79	76.88	67.51
Nursery	91.59	96.23	91.39
Optdigits	82.49	87.49	83.01
Page-blocks	93.95	96.67	94.06
Pendigits	89.66	95.19	89.62
Primary-tumor	38.7	39.53	37.85

Segment	90.09	95.02	90.31
Sick	95.68	98.07	95.53
Solar-flare2	98.22	99.5	98.32
Sonar	67.56	70.53	69.34
Soybean	86.85	91.7	79.55
Spambase	89.87	91.56	89.38
Spectrometer	42.63	43.01	38.78
Splice	81.23	90.87	80.3
Sponge	83	89.07	84.8
Tae	52.23	49.01	52.55
Vehicle	66.17	69.63	65.72
Vote	92.2	94.39	92.52
Vowel	78.19	75.15	78.42
Waveform	69.1	74.96	69.07
Wine	86.17	89.22	86.28
Zoo	91.99	92.1	91.49
Trung bình	77.74	80.72	77.06

Từ bảng số liệu này ta thấy rằng độ chính xác của NC4.5 đã được cải tiến rõ rệt, hơn hẳn hai thuật toán còn lại khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 10%



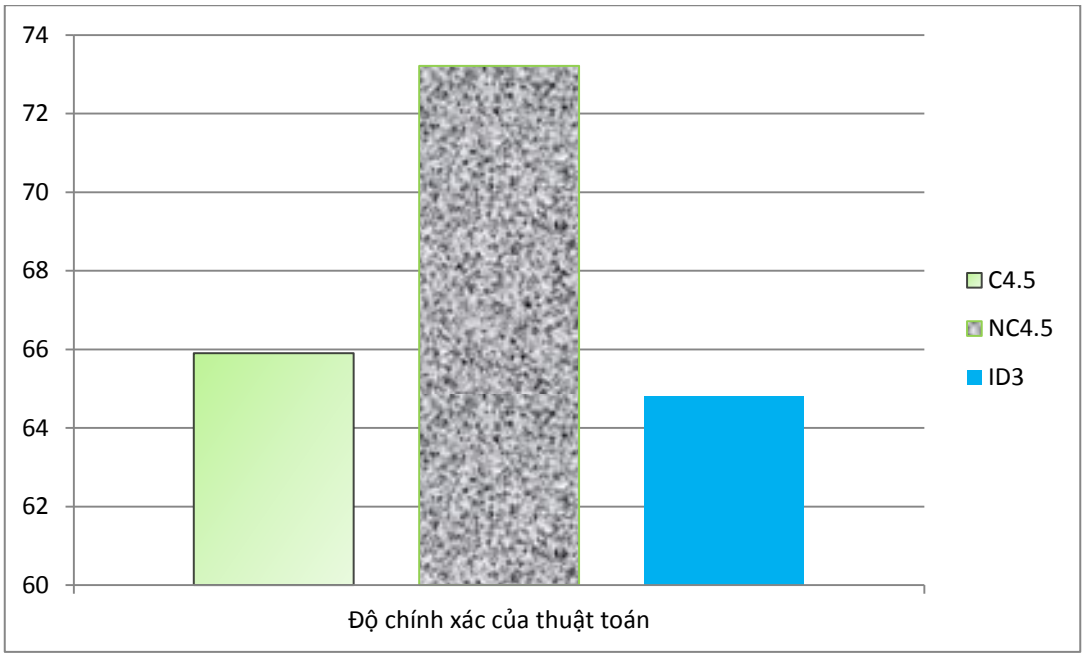
Hình 4.3 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 10%

Bảng 4.4 Kết quả về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 30%

Tập dữ liệu	**C4.5**	**NC4.5**	**ID3**
Anneal	80.5	90.84	79.8
Arrhythmia	46.6	59.71	44.8
Audiology	63.9	69.45	53.4
Autos	56.7	58.4	52.2
Balance-scale	64.8	73.61	64.4
Breast-cancer	58.8	62.71	58.6
Wisconsin-breast-cancer	84.6	91.45	84.6
Car	73.9	82.8	73.4
CMC	43	45.23	43.2
Horse-colic	68.4	73.91	66.1
Credit-rating	64.4	73.03	63.6
German-credit	58	60.57	59.4
Dermatology	69.2	79.93	67.5
Pima-diabetes	68.9	69.43	68.5
Ecoli	64.1	77.71	64.1
Glass	52.5	59.82	52.3
Haberman	63.9	66.44	63.9
Cleveland-14-heart-disease	60	69.09	60
Hungarian-14-heart-disease	70.8	79.94	66.2
Heart-statlog	63.1	71.19	62.6
Hepatitis	62.2	70.41	62.2
Hypothyroid	79.5	97.45	80.5
Ionosphere	77.9	79.84	77.1
Iris	78.6	88.27	78
kr-vs-kp	72.2	81.16	72.4
Letter	71.7	77.44	71.4
Liver-disorders	56.7	55.37	57
Lymphography	56.5	63.43	54.9
mfeat-pixel	57.8	64.57	53.6
Nursery	72.2	88.95	71.7
Optdigits	63	73.59	63.8
Page-blocks	83.8	96.04	83.1
Pendigits	69.9	86.66	70.1
Primary-tumor	34.7	34.9	33.7
Segment	70.7	89.86	71.5
Sick	87.7	94.96	87.8

Solar-flare2	91.4	96.9	91.9
Sonar	60.7	63.34	61.1
Soybean	73	86.32	58.8
Spambase	85.4	87.44	84.5
Spectrometer	31.9	35.4	29.1
Splice	62.5	68	61.9
Sponge	63.5	71.29	62.9
Tae	45.7	43.31	44.9
Vehicle	53.1	62.99	53.2
Vote	79.1	84.45	78.7
Vowel	64.7	65	64.1
Waveform	57.2	69.84	56.5
Wine	70	82.4	70.1
Zoo	84.1	85.49	85.8
Trung Bình	65.9	73.21	64.8

Từ bảng số liệu này ta thấy rằng độ chính xác của NC4.5 đã được cải tiến rõ rệt, hơn hẳn hai thuật toán còn lại khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 30%

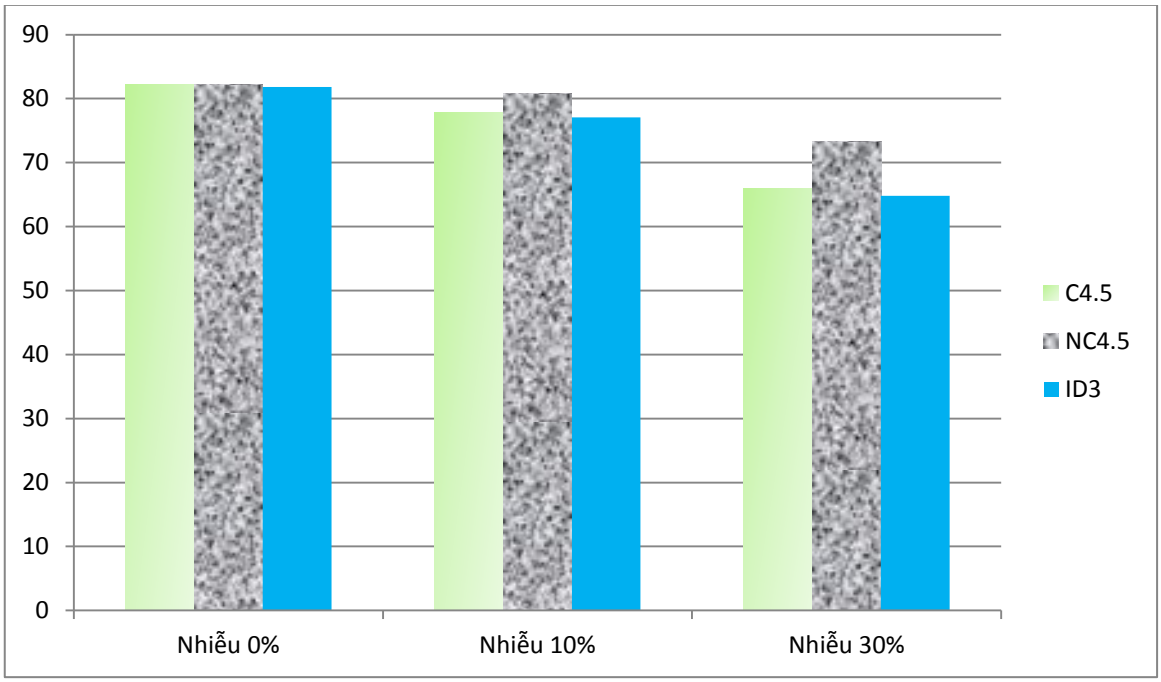


Hình 4.4 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 30%

Bảng 4.5 Kết quả về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Cây	**Nhiễu 0%**	**Nhiễu 10%**	**Nhiễu 30%**
C4.5	82.13	77.74	65.86
NC4.5	82.23	80.72	73.21
ID3	81.81	77.06	64.82

Từ bảng số liệu này ta thấy rằng khi độ nhiễu của dữ liệu càng cang thì độ chính xác của NC4.5 được cải tiến càng rõ rệt, hơn hẳn hai thuật toán còn lại. Còn khi độ nhiễu bằng 0% thì sự chênh lệch dường như không đáng kể.

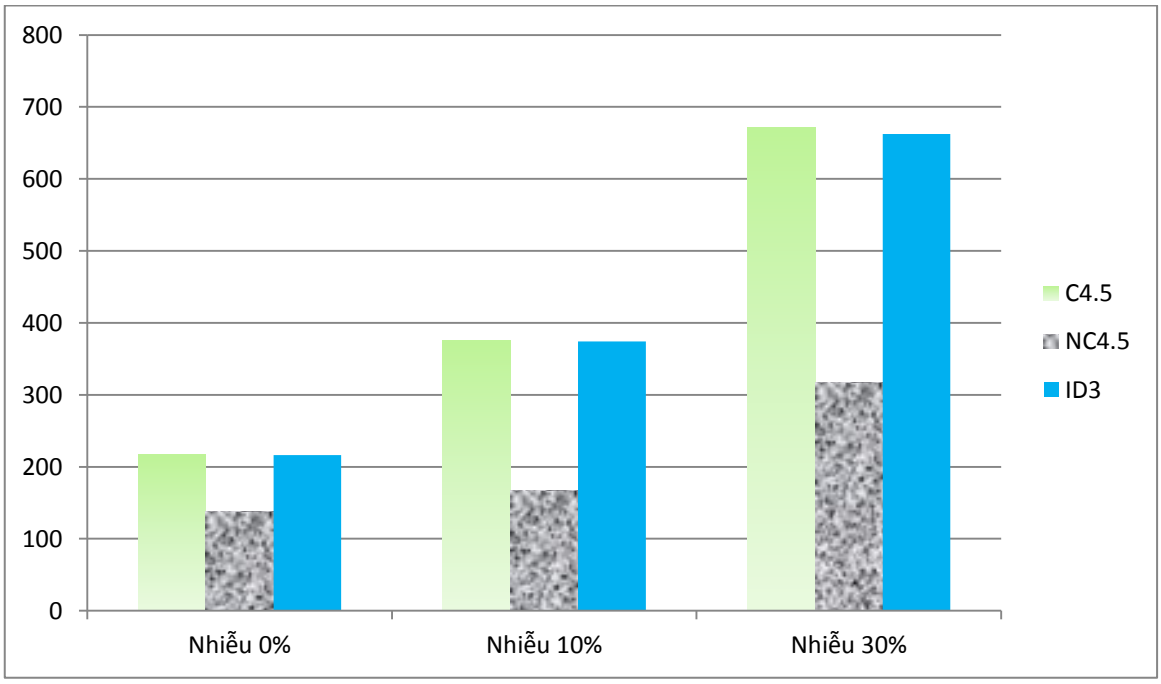


Hình 4.5 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Bảng 4.6 Kết quả về kích thước trung bình của cây cho C4.5, NC4.5, ID3 (không tỉa) khi áp dụng trên tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Cây	**Nhiễu 0%**	**Nhiễu 10%**	**Nhiễu 30%**
C4.5	216.98	376.37	672.13
NC4.5	138.78	167.09	317.92
ID3	216.15	373.97	662.42

Từ bảng số liệu này ta thấy rõ sự khác biệt về kích thước trung bình của cây NC4.5 so với C4.5, ID3 khi áp dụng trên tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%, 10% và 30%. NC4.5 cho cây có kích thước nhỏ hơn nhiều so với hai thuật toán còn lại, đặc biệt là khi đỗ nhiễu càng tăng thì sự cải thiện về kích thước trung bình của cây càng lớn. Trong khi đó kích thước cây của hai thuật toán C4.5 và ID3 gần như tương tự nhau.



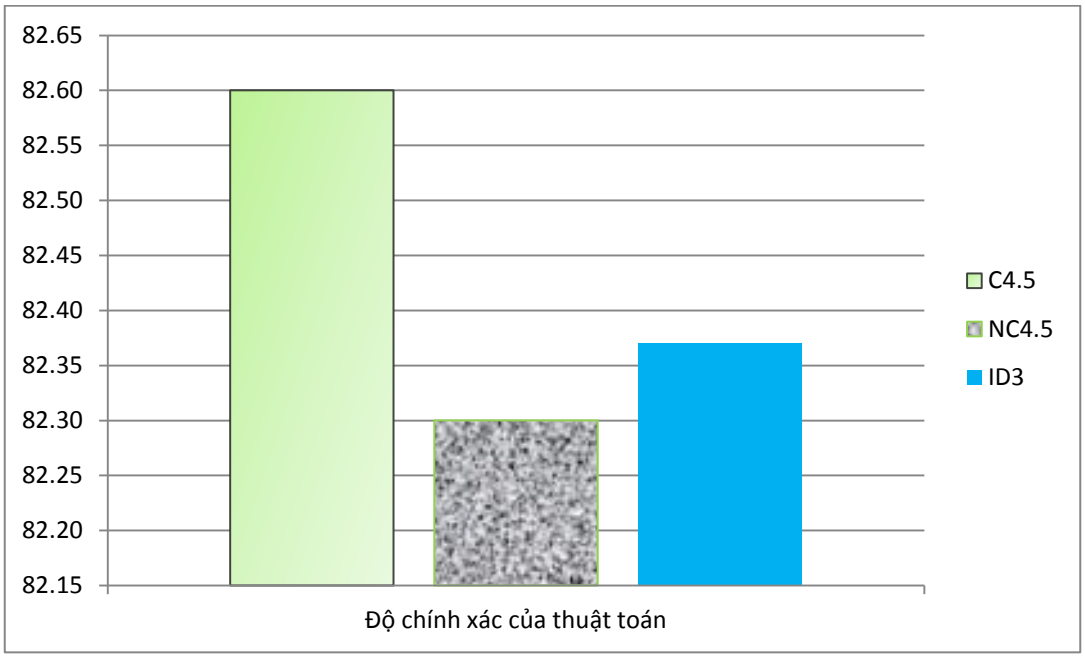
Hình 4.6 Biểu đồ so sánh về kích thước trung bình của cây tạo bởi C4.5, NC4.5, ID3 (không tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Bảng 4.7 Độ chính xác của C4.5, NC4.5 và ID3 (có tỉa) khi được áp dụng trên các tập dữ liệu với độ nhiễu ngẫu nhiên bằng 0%

Dataset	**C4.5**	**NC4.5**	**ID3**
Anneal	98.6	98.36	98.99
Arrhythmia	65.7	67.68	65.15
Audiology	77.3	78.94	76.91
Autos	81.8	74.57	78.24
Balance-scale	77.8	77.33	77.69
Breast-cancer	74.3	74.84	71.75
Wisconsin-breast-cancer	95	95.12	95.35
Car	92.2	91.16	93.02
CMC	51.4	52.8	52.06
Horse-colic	85.2	85.18	84.34
Credit-rating	85.6	85.43	84.03
German-credit	71.3	71.34	71.98
Dermatology	94.1	94.26	93.49
Pima-diabetes	74.5	74.15	74.39
Ecoli	82.8	81.6	83.61
Glass	67.6	63.61	67.67
Haberman	72.2	71.18	72.03
Cleveland-14-heart-disease	76.9	76.53	79.3
Hungarian-14-heart-disease	80.2	82.33	76.77
Heart-statlog	78.2	80.33	78.81
Hepatitis	79.2	79.79	80.33
Hypothyroid	99.5	99.52	99.56
Ionosphere	89.7	88.18	88.04
Iris	94.7	94.73	94.73
kr-vs-kp	99.4	99.45	99.42
Letter	88	87.58	87.97
Liver-disorders	65.8	64.53	66.16
Lymphography	75.8	78.31	75.01
mfeat-pixel	78.7	79.76	77.12
Nursery	97.2	96.3	97.1
Optdigits	90.5	90.83	91.1
Page-blocks	97	96.69	97.09

Pendigits	96.5	96.42	96.39
Primary-tumor	41.4	42.33	39.92
Segment	96.8	96.04	96.74
Sick	98.7	98.79	98.85
Solar-flare2	99.5	99.53	99.53
Sonar	73.6	71.37	73.53
Soybean	91.8	92.4	89.94
Spambase	92.7	92.56	93.11
Spectrometer	47.5	45.54	43.37
Splice	94.2	94.04	93.57
Sponge	92.5	92.5	92.5
Tae	57.4	53.26	57.62
Vehicle	72.3	72.78	72.71
Vote	96.6	96.59	96.11
Vowel	80.2	77.88	83.63
Waveform	75.3	76.07	75.83
Wine	93.2	92.13	93.83
Zoo	92.6	92.42	92.01
Trung bình	82.6	82.3	82.37

Từ bảng số liệu này ta thấy độ chính xác của NC4.5 so với C4.5 và ID3 (có tỉa) khi áp dụng với dữ liệu không nhiễu không hơn, thậm chí còn thấp hơn. Trong trường hợp này NC4.5 không phát huy được ưu điểm của mình.



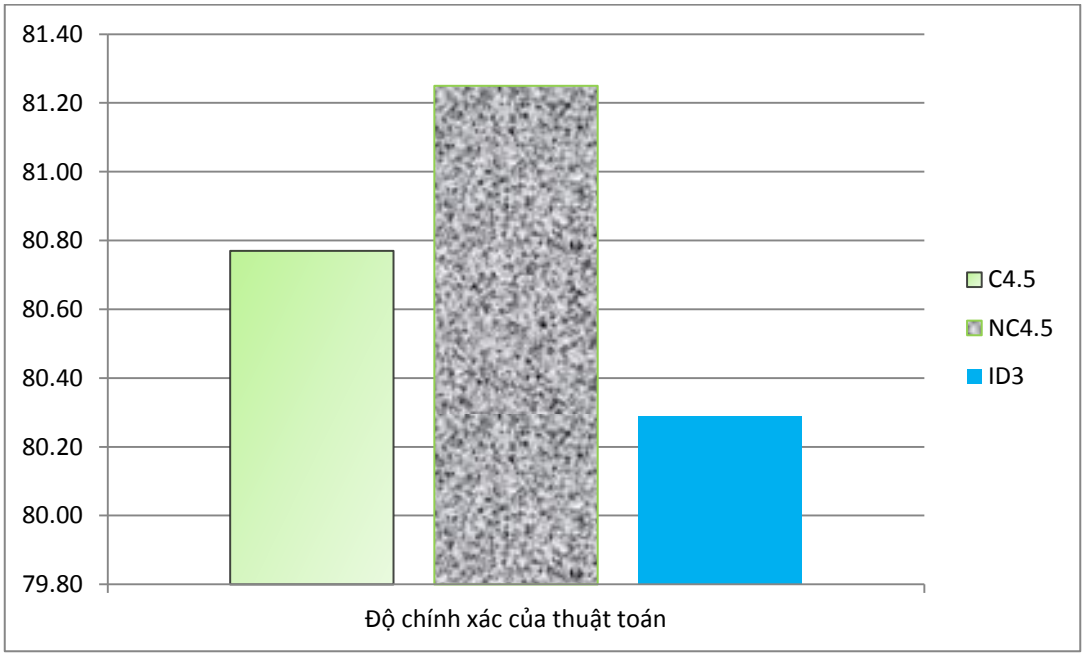
Hình 4.7 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (có tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%

Bảng 4.8 Độ chính xác của C4.5, NC4.5 và ID3 (có tỉa) khi được áp dụng trên các tập dữ liệu với độ nhiễu ngẫu nhiên bằng 10%

Dataset	**C4.5**	**NC4.5**	**ID3**
Anneal	98.37	98.23	98.42
Arrhythmia	62.54	65.76	58.44
Audiology	77.53	77.39	72.7
Autos	74.72	71.65	69.61
Balance-scale	78.11	78.26	77.82
Breast-cancer	71.13	72.07	70.75
Wisconsin-breast-cancer	93.72	94.28	94.06
Car	90.92	90.53	90.74
CMC	49.95	51.36	50.36
Horse-colic	84.61	85.1	84.5
Credit-rating	84.78	85.23	84.22
German-credit	71.18	71.38	71.72
Dermatology	93.31	93.12	91.06
Pima-diabetes	72.37	73.83	72.56
Ecoli	81.87	81.49	82.04
Glass	65.37	65.57	64.55
Haberman	72.32	72.39	72.29
Cleveland-14-heart-disease	75.78	76.94	77.56
Hungarian-14-heart-disease	79.78	80.94	77.03
Heart-statlog	75.63	78.41	76.04
Hepatitis	77.88	80.19	78.62
Hypothyroid	99.4	99.44	99.43
Ionosphere	86.9	87.04	85.79
Iris	92.73	93.53	92.47
kr-vs-kp	98.97	98.95	98.8
Letter	86.74	86.67	86.38
Liver-disorders	62.38	61.69	62.73
Lymphography	75.11	74.78	76.53
mfeat-pixel	76.77	77.97	74.36
Nursery	96.29	96.08	96
Optdigits	88.47	88.94	88.86
Page-blocks	96.7	96.78	96.79
Pendigits	95.37	95.49	95.2
Primary-tumor	39.59	40.39	40.09
Segment	95.06	95.17	95.03
Sick	98.22	98.24	98.22

Solar-flare2	99.53	99.53	99.53
Sonar	67.56	70.39	69.34
Soybean	90.54	91.74	85.85
Spambase	90.96	91.52	90.57
Spectrometer	43.2	43.07	39.64
Splice	93.05	93.08	92.48
Sponge	91.8	91.66	92.5
Tae	50.77	49.01	51.61
Vehicle	68.51	69.99	68.26
Vote	95.74	95.45	95.28
Vowel	77.13	75.26	78.37
Waveform	69.51	75.13	69.5
Wine	87.35	89.39	87.36
Zoo	92.39	92.1	92.19
Trung bình	80.77	81.25	80.29

Từ bảng số liệu này ta thấy độ chính xác của NC4.5 so với C4.5 và ID3 (có tỉa) khi áp dụng với dữ liệu nhiễu 10% tốt hơn. Trong trường hợp này NC4.5 đã phát huy được ưu điểm của mình.



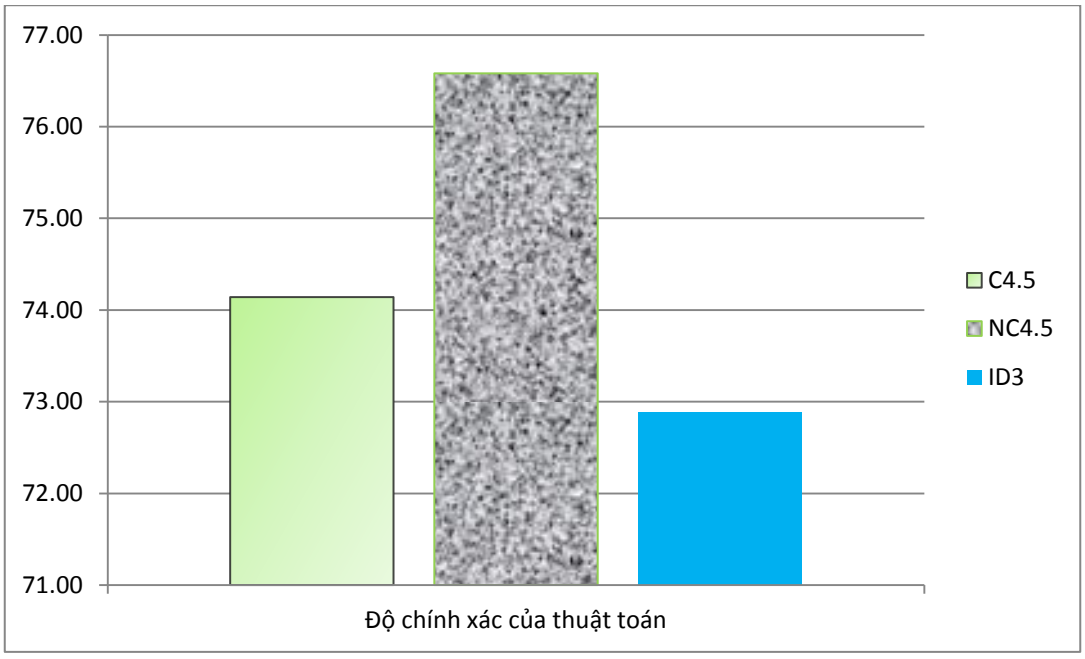
Hình 4.8 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (có tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 10%

Bảng 4.9 Độ chính xác của C4.5, NC4.5 và ID3 (có tỉa) khi được áp dụng trên các tập dữ liệu với độ nhiễu ngẫu nhiên bằng 30%.

Dataset	**C4.5**	**NC4.5**	**ID3**
Anneal	96.03	95.85	95.24
Arrhythmia	49.15	62.06	45.09
Audiology	70.88	70.68	60.25
Autos	57.92	60.35	53.81
Balance-scale	74.16	75.02	73.52
Breast-cancer	68.65	67.61	67.49
Wisconsin-breast-cancer	89.24	92.27	89.43
Car	86	85.97	85.89
CMC	46.39	47.7	45.59
Horse-colic	79.63	80.48	75
Credit-rating	74.58	81.41	71.77
German-credit	63.09	63.7	66.05
Dermatology	87.64	88.95	86.56
Pima-diabetes	69.39	69.67	68.93
Ecoli	75.27	79.78	73.63
Glass	55.23	60.49	54.69
Haberman	68.83	72.85	68.87
Cleveland-14-heart-disease	68	71.57	67.97
Hungarian-14-heart-disease	78.16	80.81	74.68
Heart-statlog	65.52	72.33	64.7
Hepatitis	68.15	73.36	68.63
Hypothyroid	98.59	98.96	98.41
Ionosphere	78.18	80.04	77.3
Iris	84	89	84.07
kr-vs-kp	91.13	90.97	90.53
Letter	82.13	82.54	81.62
Liver-disorders	56.83	55.45	57.06
Lymphography	66.33	68.11	68.59
mfeat-pixel	71.98	73.19	68.43
Nursery	93.99	94.3	93.46
Optdigits	76.91	80.77	70.24
Page-blocks	94.91	96.25	94.81
Pendigits	89.21	92.25	88.02
Primary-tumor	37.67	37.76	38.44
Segment	85.35	91.92	84.33
Sick	95.2	97.14	95.29

Solar-flare2	99.53	99.49	99.53
Sonar	60.84	63.34	61.1
Soybean	88.45	89.34	72.78
Spambase	86.07	87.69	85.32
Spectrometer	33.02	35.61	29.72
Splice	81.21	80.06	81.85
Sponge	88.84	86.71	92.5
Tae	45.86	43.64	45.26
Vehicle	56.06	63.5	55.56
Vote	90.99	91.55	91.38
Vowel	66.01	65.61	64.16
Waveform	57.32	70.08	56.59
Wine	71.02	82.91	70.98
Zoo	87.65	87.74	89.05
Trung bình	74.14	76.58	72.88

Từ bảng số liệu này ta thấy độ chính xác của NC4.5 so với C4.5 và ID3 (có tỉa) khi áp dụng với dữ liệu nhiễu 30% đã tốt hơn nhiều. Trong trường hợp này NC4.5 đã phát huy rất rõ được ưu điểm của mình.

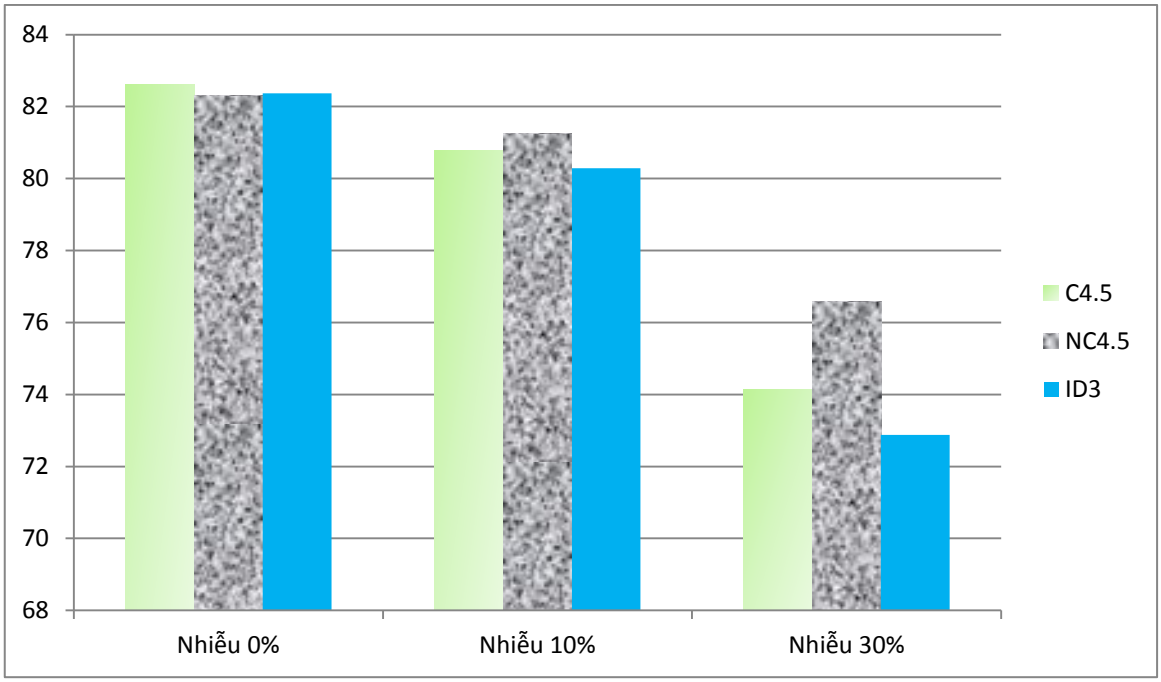


Hình 4.9 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (có tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 30%

Bảng 4.10 Độ chính xác trung bình của C4.5, NC4.5 and ID3 (có tỉa) khi được áp dụng trên các tập dữ liệu với độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Cây	**Nhiễu 0%**	**Nhiễu 10%**	**Nhiễu 30%**
C4.5	82.62	80.77	74.14
NC4.5	82.3	81.25	76.58
ID3	82.37	80.29	72.88

Từ bảng số liệu này ta thấy rằng khi độ nhiễu của dữ liệu càng cang thì độ chính xác của NC4.5 được cải tiến càng rõ rệt, hơn hẳn hai thuật toán còn lại. Còn khi độ nhiễu bằng 0% thì sự chênh lệch dường như không đáng kể.

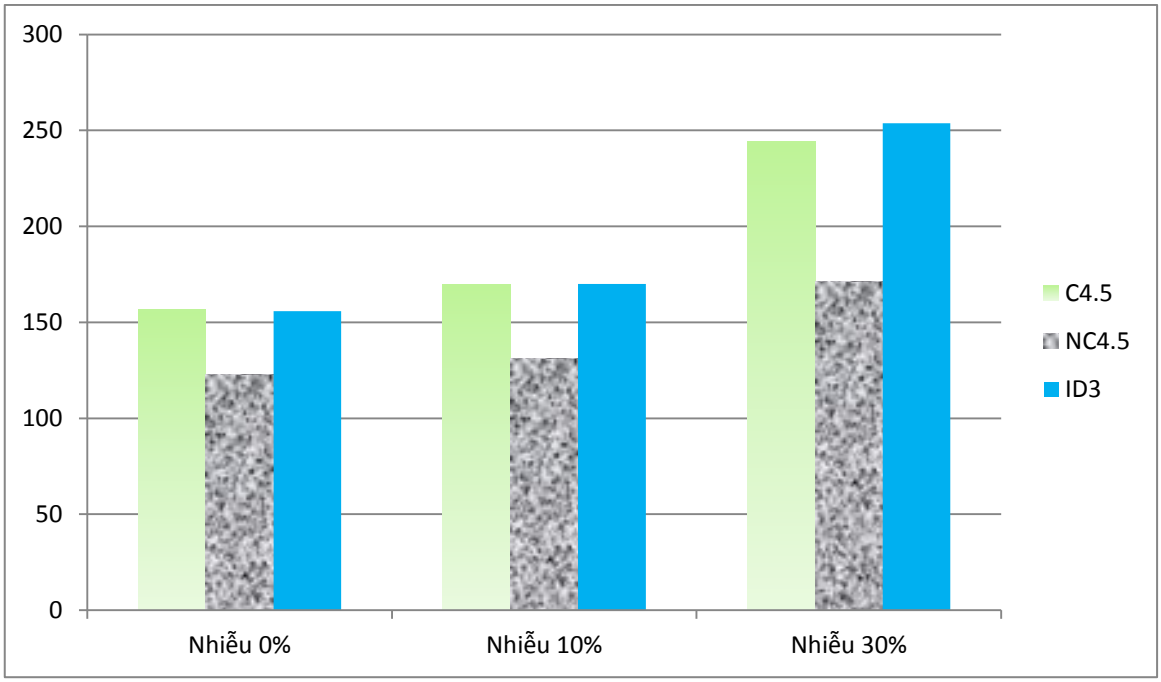


Hình 4.10 Biểu đồ so sánh độ về độ chính xác của C4.5, NC4.5, ID3 (có tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Bảng 4.11 Kết quả trung bình về kích thước cây của C4.5, NC4.5 và ID3 (có tỉa) khi được áp dụng trên các tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Cây	Nhiễu 0%	Nhiễu 10%	Nhiễu 30%
C4.5	156.54	170.02	244.05
NC4.5	122.67	131.06	171.39
ID3	155.83	170.03	253.73

Từ bảng số liệu này ta thấy rõ sự khác biệt về kích thước trung bình của cây NC4.5 so với C4.5, ID3 (có tỉa) khi áp dụng trên tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%, 10% và 30%. NC4.5 cho cây có kích thước nhỏ hơn nhiều so với hai thuật toán còn lại, đặc biệt là khi đỗ nhiễu càng tăng thì sự cải thiện về kích thước trung bình của cây càng lớn. Trong khi đó kích thước cây của hai thuật toán C4.5 và ID3 gần như tương tự nhau.



Hình 4.11 Biểu đồ so sánh về kích thước trung bình của cây tạo bởi C4.5, NC4.5, ID3 (có tỉa) khi áp dụng với tập dữ liệu có độ nhiễu ngẫu nhiên bằng 0%; 10% và 30%.

Từ các bảng kết quả thực nghiệm và các biểu đồ so sánh trên thấy rằng NC4.5 có độ chính xác cao hơn, kích thước cây nhỏ hơn và hiệu quả các hơn các thuật toán C4.5, ID3 khi áp dụng trên các tập dữ liệu có nhiễu.

CHƯƠNG 5
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT LUẬN

Phân loại dữ liệu nhiễu là một lĩnh vực quan trọng của khái thác dữ liệu. Luận văn đã đưa được ra một phương pháp xây dựng cây quyết định gọi là NC4.5. Phương pháp này có nhiều cải tiến so với thuật toán C4.5 bằng cách sử dụng xác suất mơ hồ và đo lường không chắc chắn. Do vậy nó hoạt động hiệu quả hơn các thuật toán trước đây trong việc khai thác dữ liệu nhiễu.

Kết quả thực nghiệm cho thấy thuật toán NC4.5 có cải tiến so với các thuật toán trước đây về hiệu quả, độ chính xác và kích thước cây quyết định trong việc phân loại dữ liệu nhiễu. Nó là một phương pháp phù hợp để phân loại những dữ liệu nhiễu.

5.2 HƯỚNG PHÁT TRIỂN

Hướng phát triển của luận văn là áp dụng phương pháp này vào các ứng dụng thực tế, đưa ra các phân loại dữ liệu, các thông tin quyết định hữu ích cho các lĩnh vực, ngành nghề cụ thể, đặc biệt là những lĩnh vực mà cơ sở dữ liệu của nó bị nhiễu, khó áp dụng các phương pháp phân loại thông thường.

Cơ sở dữ liệu trong thực tế thường rất lớn, cho nên trong tương lai cần nghiên cứu để cải tiến phương pháp về kích thước cây quyết định và thời gian thực thuật toán tốt hơn nữa.

TÀI LIỆU THAM KHẢO:

[1] Lê Hoài Bắc (2013), *Bài giảng môn Data Mining*, Đại học KHTN (Đại học Quốc gia Tp.HCM).

[2] Abellán, J., & Moral, S. (2003). Building classification trees using the total uncertainty criterion. International Journal of Intelligent Systems, 18(12), 1215–1225.

[3] Abellán, J., & Moral, S. (2005). Upper entropy of credal sets. Applications to creedal classification.International Journal of Approximate Reasoning, 39(2–3), 235–255.

[4] Abellán, J. (2006). Uncertainty measures on probability intervals from Imprecise Dirichlet model.International Journal of General Systems, 35(5), 509–528.

[5] Abellán, J., & Moral, S. (2006). An algorithm that computes the upper entropy for order-2 capacities.International Journal of Uncertainty, Fuzziness and Knowledge-879 Based Systems, 14(2), 141–154.

[6] Abellán, J., Klir, G. J., & Moral, S. (2006). Disaggregated total uncertainty measure for credal sets.International Journal of General Systems, 35(1), 29–44.

[7] Abellán, J., & Masegosa, A. (2008). Requirements for total uncertainty measures in Dempster–Shafer theory of evidence.International Journal of General Systems, 37(6), 733–747.

[8] Abellán, J., & Masegosa, A. (2009). A filter-wrapper method to select variables for the Naive Bayes classifier based on credal decision trees.International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 17(6), 833–854.

[9] Abellán, J., & Masegosa, A. R. (2009). An experimental study about simple decision trees for Bagging ensemble on data sets with classification noise. In

C. Sossai & G. Chemello (Eds.),ECSQARU. LNCS(Vol. 5590, pp. 446–456). Springer.

[10] Abellán, J., & Masegosa, A. (2012). Bagging schemes on the presence of noise in classification.Expert Systems with Applications, 39(8), 6827–6837.

[11] Abellán, J., Baker, R. M., Coolen, F. P. A., Crossman, R., & Masegosa, A. (2014). Classification with decision trees from a nonparametric predictive inference perspective.Computational Statistics and Data Analysis, 71, 789–802.

[12] Abellán, J., & Mantas, C. J. (2014). Improving experimental studies about ensembles of classifers for bankruptcy prediction and credit scoring. Expert Systems with Applications, 41, 3825–3830.

[13] Demsar, J. (2006). Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research, 7, 1–30.

[14] Fayyad, U. M., & Irani, K. B. (1993). Multi-valued interval discretization of continuous-valued attributes for classification learning. InProceedings of the 13th international joint conference on artificial intelligence(pp. 1022–1027). San Mateo: Morgan Kaufman.

[15] Frenay, B., & Verleysen, M. (in press). Classification in the presence of label noise: Asurvey. IEEE Transactions on Neural Networks and Learning Systems. <http://dx.doi.org/10.1109/TNNLS.2013.2292894>.

[16] Alcalá-Fdez, J., Sánchez, L., García, S., Del Jesus, M. J., Ventura, S., Garrell, J. M., et al.

[17] (2009). KEEL: A software tool to assess evolutionary algorithms to data mining problems.Soft Computing, 13(3), 307–318.

[18] Klir, G. J. (2006).Uncertainty and information. Foundations of generalized information

[19] theory. Hoboken, NJ: John Wiley. Mantas, C. J., & Abellán, J. (2014).

Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data. Expert Systems with Applications, 41, 2514–2525.

[20] Quinlan, J. R. (1986). Induction of decision trees.Machine Learning, 1, 81–106.

[21] Quinlan, J. R. (1999). Programs for machine learning. Morgan Kaufmann series in machine learning.

[22] Rokach, L., & Maimon, O. (2010). Classification trees. Data mining and knowledge discovery handbook (pp. 149–174).

[23] Walley, P. (1996). Inferences from multinomial data, learning about a bag of marbles.Journal of the Royal Statistical Society, Series B, 58, 3–57.

[24] Wang, Y. (2010). Imprecise probabilities based on generalised intervals for system reliability assessment. International Journal of Reliability and Safety, 4(30), 319–342.

[25] Witten, I. H., & Frank, E. (2005).Data mining, practical machine learning tools and techniques(2nd edition.). San Francisco: Morgan Kaufman.

[26] Weichselberger, K. (2000). The theory of interval-probability as a unifying concept for uncertainty. International Journal of Approximate Reaso