HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỖ THỊ LƯƠNG

NGHIÊN CỨU MỘT SỐ THUẬT TOÁN HỌC MÁY ĐỂ PHÂN LỚP DỮ LIỆU VÀ THỬ NGHIỆM

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2019

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỖ THỊ LƯƠNG

NGHIÊN CỨU MỘT SỐ THUẬT TOÁN HỌC MÁY ĐỂ PHÂN LỚP DỮ LIỆU VÀ THỬ NGHIỆM

Chuyên ngành: Hệ thống thông tin
Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌCTS. VŨ VĂN THỎA

HÀ NỘI – 2019

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận văn có tham khảo và sử dụng các tài liệu, thông tin được đăng tải trên những tạp chí khoa học và các trang web được liệt kê trong danh mục tài liệu tham khảo. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà nội, ngày 20 tháng 11 năm 2019

Người cam đoan

Đỗ Thị Lương

LỜI CẢM ƠN

Được sự đồng ý của Học Viện Công Nghệ Bưu Chính Viễn Thông, và của thầy giáo hướng dẫn TS. Vũ Văn Thỏa, học viên đã thực hiện đề tài luận văn tốt nghiệp Thạc sĩ: "Nghiên cứu một số thuật toán học máy để phân lớp dữ liệu và thử nghiệm".

Để hoàn thành luận văn này, học viên xin chân thành cảm ơn các thầy cô giáo đã tận tình hướng dẫn, giảng dạy trong suốt quá trình học tập, nghiên cứu và rèn luyện ở Học Viện Công Nghệ Bưu Chính Viễn Thông.

Học viên xin đặc biệt gửi lời cảm ơn đến TS. Vũ Văn Thỏa, người thầy đã trực tiếp hướng dẫn trong quá trình thực hiện luận văn tốt nghiệp này. Nhờ sự động viên và chỉ bảo tận tình của thầy trong thời gian qua đã giúp học viên vượt qua những khó khăn khi nghiên cứu để luận văn được hoàn thành.

Học viên xin gửi lời cảm ơn tới gia đình, bạn bè và đồng nghiệp, những người đã luôn ở bên cổ vũ tinh thần, tạo điều kiện thuận lợi để học viên có thể học tập và hoàn thành tốt luận văn này.

Học viên đã có nhiều cố gắng để thực hiện luận văn một cách hoàn chỉnh nhất. Tuy nhiên, do còn nhiều hạn chế về kiến thức và kinh nghiệm nên không thể tránh khỏi những thiếu sót nhất định mà học viên chưa thấy được. Học viên rất mong nhận được sự góp ý của quý Thầy, Cô giáo và các bạn đồng nghiệp để luận văn được hoàn chỉnh hơn.

Học viên xin trân trọng cám ơn!

Hà Nội, ngày 20 tháng 11 năm 2019

Học viên

Đỗ Thị Lương

MỤC LỤC

LỜI CAM ĐOAN i
LỜI CẢM ƠN ii
MỤC LỤC iii
DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT v
DANH MỤC BẢNG vi
DANH MỤC HÌNH vii
MỞ ĐẦU 1
CHƯƠNG 1. TỔNG QUAN VỀ PHÂN LỚP DỮ LIỆU VÀ HỌC MÁY 3
1.1. Giới thiệu bài toán phân lớp dữ liệu và các vấn đề liên quan 3
1.1.1. *Khái niệm về phân lớp dữ liệu và bài toán phân lớp dữ liệu* *3*
1.1.2. *Quy trình giải quyết bài toán phân lớp dữ liệu* *4*
1.1.3. *Các độ đo đánh giá mô hình phân lớp dữ liệu* *6*
1.1.4. *Các phương pháp đánh giá mô hình phân lớp dữ liệu* *7*
1.1.5. *Các ứng dụng của bài toán phân lớp dữ liệu* *8*
1.1.6. *Các phương pháp phân lớp dữ liệu* *10*
1.2. Tổng quan về học máy 11
1.2.1. *Khái niệm về học máy và phân loại các kỹ thuật học máy* *11*
a. Khái niệm về học máy 11
b. Phân loại các kỹ thuật học máy 12
Học có giám sát 12
Học không giám sát 13
Học bán giám sát 14
1.2.2. *Ứng dụng học máy xây dựng mô hình phân lớp dữ liệu* *15*
1.3. Giới thiệu chung về học sâu 15
1.3.1. *Khái niệm về học sâu* *15*
1.3.2. *Hướng tiếp cận học sâu* *16*
1.4. Kết luận chương 1 18
CHƯƠNG 2. NGHIÊN CỨU MỘT SỐ THUẬT TOÁN HỌC MÁY 19
2.1. Khảo sát thuật toán cây quyết định và các vấn đề liên quan 19
2.1.1. *Giới thiệu phương pháp* *19*

2.1.2. *Xây dựng cây quyết định dựa trên Entropy* *21*
2.1.3. *Đánh giá phương pháp* *22*
2.2. Khảo sát thuật toán Bayes và các vấn đề liên quan 22
2.2.1. *Giới thiệu phương pháp* *22*
2.2.2. *Thuật toán Naïve Bayes* *23*
2.2.3. *Mạng Bayes* *24*
2.2.4. *Đánh giá phương pháp* *25*
2.3. Khảo sát thuật toán máy vectơ hỗ trợ và các vấn đề liên quan 26
2.3.1. *Giới thiệu phương pháp* *26*
2.3.2. *Thuật toán SVM tuyến tính với tập dữ liệu phân tách được* *28*
2.3.3. *Thuật toán SVM tuyến tính với tập dữ liệu không phân tách được* *32*
2.3.4. *Thuật toán SVM phi tuyến phân lớp nhị phân* *35*
2.3.5. *Thuật toán tối thiểu tuần tự SMO* *38*
2.3.6. *Thuật toán SVM phân lớp đa lớp* *38*
2.3.7. *Đánh giá phương pháp* *40*
2.4. Kết luận chương 2 41
CHƯƠNG 3. THỬ NGHIỆM VÀ ĐÁNH GIÁ 42
3.1. Khảo sát và lựa chọn bộ dữ liệu để thử nghiệm 42
3.1.1. *Giới thiệu chung* *42*
3.1.2. *Mô tả bộ dữ liệu KDD Cup 99* *43*
3.2. Xây dựng kịch bản và lựa chọn công cụ thử nghiệm 48
3.2.1. *Xây dựng kịch bản thử nghiệm* *48*
3.2.2. *Lựa chọn công cụ thử nghiệm* *49*
3.3. Triển khai thử nghiệm và đánh giá kết quả 51
3.3.1. *Mô tả thử nghiệm* *51*
3.3.2. *Kết quả thử nghiệm* *52*
3.3.3. *Đánh giá kết quả thử nghiệm* *55*
3.4. Kết luận chương 3 59
KẾT LUẬN 60
DANH MỤC TÀI LIỆU THAM KHẢO 61

DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng việt
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
CNTT	Information Technology	Công nghệ thông tin
CSDL	Database	Cơ sở dữ liệu
DoS	Denial of Service	Tấn công từ chối dịch vụ
HL	Training	Huấn luyện
KC	Test	Kiểm chứng
KDD	Knowledge Discovery and Data Mining	Phát hiện tri thức và khai phá dữ liệu
R2L	Remote to Local	Tấn công điều khiển từ xa
SVM	Support Vector Machines	Máy véc tơ hỗ trợ
SMO	Sequential Minimal Optimization	Tối thiểu tuần tự
U2R	User to Root	Tấn công chiếm quyền root
WEKA	Waikato Environment for Knowledge Acquisition	Công cụ kiểm thử học máy

DANH MỤC BẢNG

Số hiệu	Tên bảng	Trang
Bảng 3.1	Nhãn lớp và số mẫu xuất hiện trong 10% bộ dữ liệu KDD cup 99 [13]	**44**
Bảng 3.2	Các thuộc tính của bộ dữ liệu KDD cup 99 [18]	**45**
Bảng 3.3	Kết quả thử nghiệm 2 lớp của thuật toán j48	**52**
Bảng 3.4	Kết quả thử nghiệm 2 lớp của thuật toán Naïve-Bayes	**52**
Bảng 3.5	Kết quả thử nghiệm 2 lớp của thuật toán Net-Bayes	**53**
Bảng 3.6	Kết quả thử nghiệm 2 lớp của thuật toán SMO	**53**
Bảng 3.7	Tổng hợp kết quả huấn luyện 2 lớp của các thuật toán thử nghiệm	**53**
Bảng 3.8	Tổng hợp kết quả kiểm chứng 2 lớp của các thuật toán thử nghiệm	**54**
Bảng 3.9	Tổng hợp kết quả huấn luyện đa lớp của các thuật toán thử nghiệm	**54**
Bảng 3.10	Tổng hợp kết quả kiểm chứng đa lớp của các thuật toán thử nghiệm	**55**

DANH MỤC HÌNH

Số hiệu	Tên hình	Trang
Hình 1.1	Bài toán phân lớp dữ liệu	**3**
Hình 1.2	Giai đoạn xây dựng mô hình phân lớp dữ liệu	**4**
Hình 1.3	Quá trình kiểm tra đánh giá mô hình phân lớp dữ liệu	**5**
Hình 1.4	Ví dụ về quá trình giải quyết bài toán phân lớp dữ liệu	**6**
Hình 1.5	Mô hình kim tự tháp: Từ dữ liệu đến tri thức	**11**
Hình 1.6	Các quá trình học sâu	**16**
Hình 1.7	Quá trình học tăng cường	**17**
Hình 2.1	Mô hình cây quyết định	**19**
Hình 2.2	Mô hình mạng Bayes	**25**
Hình 2.3	Tầm quan trọng của biên đối với siêu phẳng phân tách	**27**
Hình 2.4	Ví dụ về biên tối ưu của siêu phẳng phân tách	**27**
Hình 2.5	Ảnh hưởng của C đến độ rộng biên	**33**
Hình 2.6	Ánh xạ từ không gian 2 chiều sang không gian 3 chiều	**36**
Hình 2.7	Phân lớp đa lớp sử dụng chiến lược OAA và OAO	**39**
Hình 3.1	Giao diện khởi động của WEKA	**49**
Hình 3.2	Biểu đồ so sánh độ chính xác của các thuật toán thử nghiệm 2 lớp	**56**

Hình 3.3	Biểu đồ so sánh độ chính xác của lớp Normal trong thử nghiệm 2 lớp	**57**
Hình 3.4	Biểu đồ so sánh độ chính xác của lớp Anomal trong thử nghiệm 2 lớp	**57**
Hình 3.5	Biểu đồ so sánh độ chính xác của mô hình trong thử nghiệm đa lớp	**58**
Hình 3.6	Mức chính xác theo lớp trong thử nghiệm đa lớp trên tập huấn luyện	**58**
Hình 3.7	Mức chính xác theo lớp trong thử nghiệm đa lớp trên tập kiểm chứng	**59**

MỞ ĐẦU

Trong thời gian gần đây, sự phát triển mạnh mẽ của công nghệ thông tin và các dịch vụ liên quan đã làm số lượng thông tin được trao đổi trên mạng Internet tăng một cách đáng kể. Số lượng thông tin được lưu trữ trong các kho dữ liệu cũng tăng với một tốc độ chóng mặt. Đồng thời, tốc độ thay đổi thông tin là cực kỳ nhanh chóng. Theo thống kê của Broder et al (2003), cứ sau 9 tháng hoặc 12 tháng lượng thông tin được lưu trữ, tìm kiếm và quản lý lại tăng gấp đôi. Hiện nay, loài người đang bước vào kỷ nguyên IoT (Internet of Things – Internet kết nối vạn vật). Thông qua internet, người dùng có nhiều cơ hội để tiếp xúc với nguồn thông tin vô cùng lớn. Tuy nhiên, cùng với nguồn thông tin vô tận đó, người dùng cũng đang phải đối mặt với sự quá tải thông tin. Đôi khi, để tìm được các thông tin cần thiết, người dùng phải chi phí một lượng thời gian khá lớn.

Với số lượng thông tin đồ sộ như vậy, một yêu cầu cấp thiết đặt ra là làm sao tổ chức, tìm kiếm và khai thác thông tin (dữ liệu) một cách hiệu quả nhất. Một trong các giải pháp được nghiên cứu để giải quyết vấn đề trên là xây dựng các mô hình tính toán dựa trên các phương pháp học máy nhằm phân loại, khai thác thông tin một cách tự động và trích xuất các tri thức hữu ích. Trong đó, bài toán phân lớp (Classification) dữ liệu có ý nghĩa hết sức quan trọng. Phân lớp dữ liệu là việc xếp các dữ liệu vào những lớp đã biết trước. Ví dụ: Phân lớp sinh viên theo kết quả học tập, phân lớp các loài thực vật, …

Bài toán phân lớp dữ liệu thường được giải quyết bằng cách sử dụng một số kỹ thuật học máy như: Thuật toán Bayes (Naive Bayes), Cây quyết định (Decision Tree), Máy vector hỗ trợ (Support Vector Machine), Mạng Nơ-ron nhân tạo (Artificial Neural Network), …

Xuất phát từ những lý do trên, học viên chọn thực hiện đề tài luận văn tốt nghiệp chương trình đào tạo thạc sĩ có tên "**Nghiên cứu một số thuật toán học máy để phân lớp dữ liệu và thử nghiệm**".

Mục tiêu của luận văn là nghiên cứu các kỹ thuật học máy để giải quyết bài toán phân lớp dữ liệu nói chung và thử nghiệm đánh giá hiệu năng của chúng trên bộ dữ liệu KDD cup 99.

Nội dung của luận văn được trình bày trong ba chương nội dung chính như sau:

Chương 1: Tổng quan về phân lớp dữ liệu và học máy.

Nội dung chính của chương 1 là khảo sát tổng quan về bài toán phân lớp dữ liệu, học máy và các vấn đề liên quan.

Chương 2: Nghiên cứu một số thuật toán học máy

Nội dung chính của chương 2 là nghiên cứu chi tiết một số kỹ thuật học máy để giải quyết bài toán phân lớp dữ liệu và một số vấn đề liên quan.

Chương 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ

Nội dung chính của chương 3 là thực hiện thử nghiệm và đánh giá các mô hình phân lớp dữ liệu dựa trên các phương pháp học máy đã nghiên cứu trong chương 2 cho bộ dữ liệu KDD cup 99.

CHƯƠNG 1. TỔNG QUAN VỀ PHÂN LỚP DỮ LIỆU VÀ HỌC MÁY

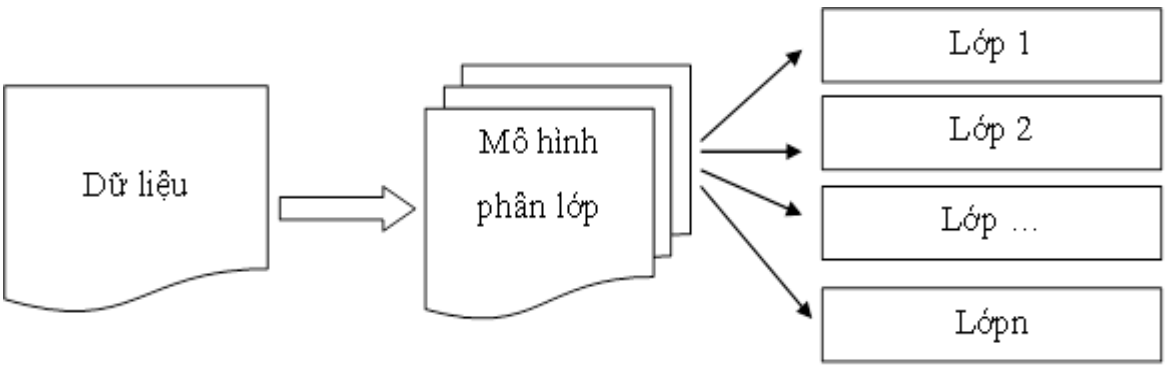
Nội dung của Chương 1 sẽ khảo sát tổng quan về bài toán phân lớp dữ liệu, học máy và các vấn đề liên quan.

1.1. Giới thiệu bài toán phân lớp dữ liệu và các vấn đề liên quan

1.1.1. Khái niệm về phân lớp dữ liệu và bài toán phân lớp dữ liệu

Phân lớp (classification) dữ liệu là một tiến trình xử lý nhằm xếp các mẫu dữ liệu hay các đối tượng vào một trong các lớp đã được định nghĩa trước. Các mẫu dữ liệu hay các đối tượng được xếp vào các lớp dựa trên giá trị của các thuộc tính (attributes) của mẫu dữ liệu hay đối tượng. Quá trình phân lớp dữ liệu kết thúc khi tất cả các dữ liệu đã được xếp vào các lớp tương ứng. Khi đó, mỗi lớp dữ liệu được đặc trưng bởi tập các thuộc tính của các đối tượng chứa trong lớp đó.

Thông thường, khi tiến hành nghiên cứu một đối tượng, hiện tượng nào đó, ta chỉ có thể dựa vào một số hữu hạn các thuộc tính đặc trưng của chúng. Nói cách khác, ta sẽ xem xét biểu diễn các đối tượng, hiện tượng trong một không gian hữu hạn chiều, mỗi chiều ứng với một đặc trưng được lựa chọn. Khi đó, phân lớp dữ liệu trở thành phân hoạch tập dữ liệu thành các tập con theo một tiêu chuẩn nhận dạng được. Như vậy, phân lớp là quá trình "nhóm" các đối tượng "giống" nhau vào "một lớp" dựa trên các đặc trưng dữ liệu của chúng. Bài toán phân lớp dữ liệu có thể được mô tả như hình 1.1 dưới đây [7].



Hình 1.1. Bài toán phân lớp dữ liệu

Ta có thể phát biểu bài toán phân lớp dữ liệu như sau:

Đầu vào của bài toán phân lớp dữ liệu:

Cho tập dữ liệu mẫu D = $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$, trong đó, $x_i = (x_{i1}, x_{i2}, .., x_{ik}) \in R^k$ là dữ liệu gồm k thuộc tính tương ứng trong tập thuộc tính A = $\{A_1, A_2, \ldots, A_k\}$ và $y_i \in C = \{c_1, c_2, \ldots, c_m\}$là nhãn các lớp dữ liêu. (1.1)

Đầu ra của bài toán phân lớp dữ liệu:

Một ánh xạ/hàm (mô hình phân lớp) F: $R^k \rightarrow C$, tương ứng mỗi phần tử $x \in R^k$ một nhãn lớp $F(x) \in C$, sao cho đối với tập mẫu D là phù hợp nhất theo nghĩa sau đây: $||F(x_i) - y_i|| \cong 0$, với mọi $(x_i, y_i) \in D$ và $|| ||$ là một độ đo nào đó. (1.2)

1.1.2. Quy trình giải quyết bài toán phân lớp dữ liệu

Bài toán phân lớp dữ liệu (1.1)-(1.2) thường được giải quyết theo 2 giai đoạn: Giai đoạn xây dựng mô hình phân lớp (còn được gọi là *giai đoạn huấn luyện*) và Giai đoạn kiểm tra đánh giá mô hình phân lớp (còn được gọi là *giai đoạn Kiểm chứng*) [7].

(1) Giai đoạn huấn luyện

Giai đoạn này nhằm xây dựng một mô hình phân lớp dựa trên mô tả tập các lớp dữ liệu hoặc các khái niệm được xác định trước. Trong giai đoạn huấn luyện, thuật toán phân lớp được sử dụng để xây dựng bộ phân lớp bằng cách phân tích hay "học" từ một tập các dữ liệu huấn luyện (training set) và các nhãn lớp tương ứng của chúng.

Quá trình thực hiện giai đoạn học được mô tả trong hình 1.2.

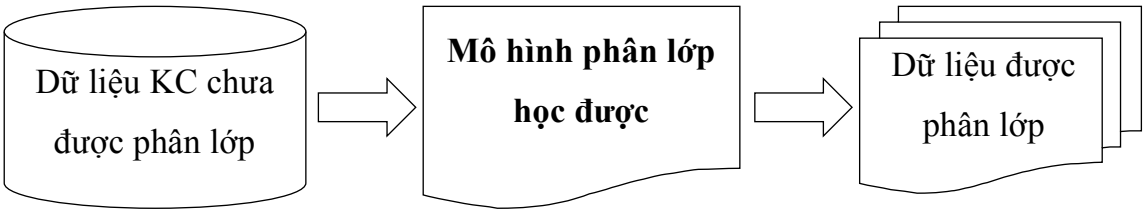
Hình 1.2. Giai đoạn xây dựng mô hình phân lớp dữ liệu

Kết quả của giai đoạn học là đưa ra một mô hình (bộ) phân lớp dữ liệu. Bộ phân lớp dữ liệu có thể là các công thức toán học, hoặc bộ các quy tắc hoặc các luật quyết định để gán nhãn lớp cho mỗi dữ liệu trong tập các dữ liệu huấn luyện.

(2) Giai đoạn kiểm chứng

Trong giai đoạn này, mô hình phân lớp có được ở giai đoạn trước sẽ được sử dụng để thực hiện phân lớp thử nghiệm và đánh giá mô hình. Tập dữ liệu được sử dụng trong giai đoạn này được gọi là tập các dữ liệu Test hay tập kiểm chứng (KC). Do đó, trong giai đoạn này cần sử dụng một tập dữ liệu kiểm chứng độc lập với tập dữ liệu huấn luyện (HL) ở giai đoạn trước.

Quá trình thực hiện giai đoạn phân lớp thử nghiệm được mô tả trong hình 1.3.

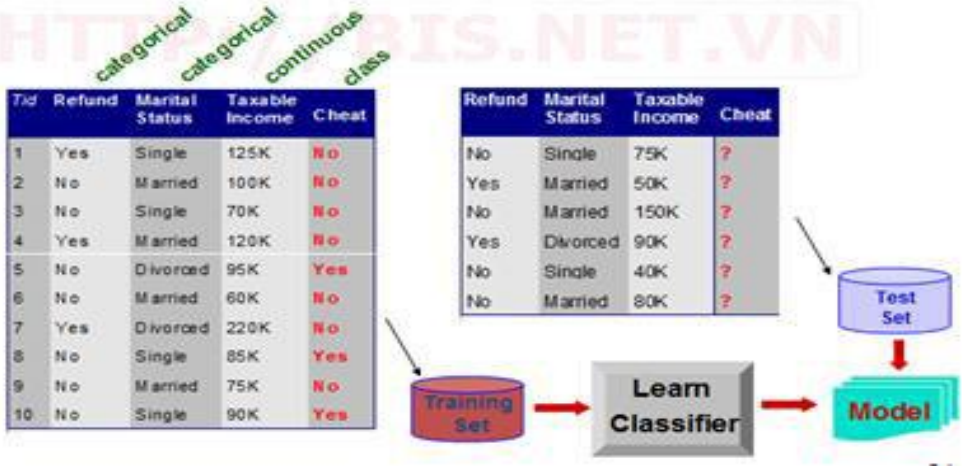


Hình 1.3. Quá trình kiểm tra đánh giá mô hình phân lớp dữ liệu

Các thông tin (kết quả) trong quá trình phân lớp thử nghiệm lại có thể sử dụng trong quá trình học tiếp theo.

Sau khi thực hiện hai giai đoạn trên, mô hình phân lớp phù hợp nhất theo một nghĩa nào đó (thông qua các độ đo đánh giá mô hình) sẽ được lựa chọn để thực hiện phân lớp dữ liệu trong các bài toán ứng dụng khác nhau trong thực tế.

Hình 1.4 dưới đây mô tả một ví dụ về quá trình thực hiện giải quyết bài toán phân lớp dữ liệu (1.1) – (1.2) [19].



Hình 1.4. Ví dụ về quá trình giải quyết bài toán phân lớp dữ liệu

1.1.3. Các độ đo đánh giá mô hình phân lớp dữ liệu

Sự phù hợp, tính hiệu quả của bất kỳ mô hình phân lớp dữ liệu nào cũng thường được xác định thông qua các độ đo được mô tả dưới đây [7].

Xét một lớp $c_i \in C = \{c_1, c_2, \ldots, c_m\}$ trong bài toán phân lớp dữ liệu (1.1) – (1.2). Các mẫu dữ liệu thuộc lớp c_i gọi là các phần tử dương (Positive). Các mẫu dữ liệu không thuộc lớp c_i gọi là các phần tử âm (Negative). Khi sử dụng các bộ phân lớp để thực hiện phân lớp dữ liệu thử nghiệm có thể xảy ra các trường hợp sau đây:

- Trường hợp đúng dương (True Positive): Phần tử dương được phân loại đúng là dương.

- Trường hợp sai dương (Fasle Positive): Phần tử âm được phân loại sai thành âm.

- Trường hợp đúng âm (True Nagetive): Phần tử âm được phân loại đúng là âm.

- Trường hợp sai âm (Fasle Nagetive): Phần tử dương được phân loại sai thành âm.

Ký hiệu *TP* (hoặc TP_i) là số lượng mẫu dữ liệu thuộc lớp c_i được phân loại đúng (chính xác) vào lớp c_i; *FP* (hoặc FP_i) là số lượng mẫu dữ liệu không thuộc lớp c_i bị phân loại sai vào lớp c_i; *TN* (hoặc TN_i) là số lượng mẫu dữ liệu không thuộc lớp c_i được phân loại chính xác và *FN* (hoặc FN_i) là số lượng mẫu dữ liệu thuộc lớp c_i bị phân loại sai vào các lớp khác với lớp c_i;

Dựa vào các đại lượng trên, có các độ đo để đánh giá hiệu quả của mô hình phân lớp dữ liệu như sau:

(1) Độ đo Precision (Mức chính xác)

- Định nghĩa: Precision = TP / (TP + FP).

- Ý nghĩa: Giá trị Precision càng cao thể hiện khả năng càng cao để một kết quả phân lớp dữ liệu được đưa ra bởi bộ phân lớp là chính xác.

(2) Độ đo Recall (Độ bao phủ, độ nhạy hoặc độ triệu hồi)

- Định nghĩa: Recall = TP / (TP + FN).

- Ý nghĩa: Giá trị Recall càng cao thể hiện khả năng kết quả đúng trong số các kết quả đưa ra của bộ phân lớp càng cao.

(3) Độ đo Accuracy (Độ chính xác)

- Định nghĩa: Accuracy = (TP + TN) / (TP + TN + FP + FN) * 100%.

- Ý nghĩa: Accuracy phản ánh độ chính xác chung của bộ phân lớp dữ liệu..

(4) Độ đo F-Measure

- Định nghĩa: F-Measure = 2.(Precision.Recall) / (Precision + Recall).

- Ý nghĩa: F-Measure là độ đo nhằm đánh giá độ chính xác thông qua quá trình kiểm chứng dựa trên sự xem xét đến hai độ đo là Precision và Recall. Giá trị F-Measure càng cao phản ánh độ chính xác càng cao của bộ phân lớp dữ liệu. Có thể coi độ đo F-Measure là trung bình điều hòa của hai độ đo Precision và Recall.

(5) Độ đo Specitivity (Độ đặc hiệu)

- Định nghĩa: Specitivity = TN/(TN+FP).

- Ý nghĩa: Độ đo Specitivity đánh giá khả năng một dữ liệu là phần tử âm được bộ phân lớp cho ra kết quả chính xác.

1.1.4. Các phương pháp đánh giá mô hình phân lớp dữ liệu

Đánh giá độ phù hợp (chính xác) và hiệu quả của mô hình phân lớp sẽ cho phép dự đoán được độ chính xác của các kết quả phân lớp dữ liệu tương lai. Đồng thời, độ phù hợp còn là cơ sở để so sánh các mô hình phân lớp khác nhau để lựa chọn mô hình phân lớp tốt nhất cho từng ứng dụng cụ thể cho các bài toán thực tế. Do đó, phương pháp đánh giá cũng có vai trò khá quan trọng.

Trong mục này, luận văn khảo sát hai phương pháp phổ biến thường được sử dụng trong đánh giá mô hình phân lớp là hold-out và k-fold cross-validation. Cả hai kỹ thuật này đều dựa trên các phân hoạch ngẫu nhiên tập dữ liệu ban đầu một cách phù hợp nhất [12].

Phương pháp Hold-out

Đối với phương pháp hold-out (Kiểm tra phân đôi), tập dữ liệu mẫu được phân chia ngẫu nhiên thành 2 phần là: tập dữ liệu huấn luyện và tập dữ liệu kiểm chứng. Thông thường, 2/3 dữ liệu được sử dụng cho tập dữ liệu huấn luyện, phần còn lại cấp cho tập dữ liệu kiểm chứng.

Phương pháp k-fold cross validation

Trong phương pháp k-fold cross validation (Kiểm tra chéo k-fold), quá trình được thực hiện như sau:

Bước 1: Chia ngẫu nhiên tập dữ liệu ban đầu S thành k tập dữ liệu (fold) có kích thước gần bằng nhau S_1, S_2,…, S_k.

Bước 2: Lặp lại thủ tục sau k lần với i= 1, 2, .., k.

- Dùng tập S_i ($1 \leq i \leq k$) làm tập kiểm tra. Gộp k-1 tập còn lại thành tập huấn luyện.

- Tiến hành Huấn luyện mô hình phân lớp trên tập huấn luyện.

- Đánh giá độ chính xác của mô hình trên tập kiểm tra,

Bước 3:

- Đánh giác độ chính xác của mô hình tính bằng trung bình cộng độ chính xác trên k lần kiểm tra ở bước trên.

- Chọn mô hình có độ chính xác trung bình lớn nhất.

Trong thực tế, thông thường chọn k= 10.

1.1.5. Các ứng dụng của bài toán phân lớp dữ liệu

Bài toán phân lớp dữ liệu có rất nhiều ứng dụng trong các lĩnh vực khoa học, công nghệ và đời sống xã hội. Dưới đây, luận văn liệt kê một số ứng dụng chủ yếu của phân lớp dữ liệu.

Ứng dụng trong khai phá dữ liệu

Trong quá trình khai phá dữ liệu, phân lớp dữ liệu trước hết có thể làm giảm độ phức tạp của không gian dữ liệu cần khai phá do mỗi lớp dữ liệu được xem xét thông qua một đại diện của lớp đó. Mặt khác, phân lớp dữ liệu giúp cho quá trình lưu trữ, quản lý và tìm kiếm dữ liệu được thuận tiện hơn.

Ứng dụng trong lĩnh vực tài chính, ngân hàng

Phân lớp dữ liệu có thể ứng dụng dự báo các rủi ro trong đầu tư tài chính và thị trường chứng khoán. Nó có thể ứng dụng để phân lớp các khách hàng, khoản vay để ngân hàng có chính sách phù hợp khi quản lý và xử lý nợ xấu, … .

Ứng dụng trong thương mại

Phân lớp dữ liệu được ứng dụng trong phân tích dữ liệu khách hàng, hoạch định chính sách marketing hiệu quả cũng như phát hiện các gian lận thương mại.

Ứng dụng trong sinh học

Phân lớp dữ liệu được sử dụng để tìm kiếm, so sánh các hệ gen và thông tin di chuyền, tìm mối liên hệ giữa các hệ gen hỗ trợ chẩn đoán một số bệnh di chuyền.

Ứng dụng trong y tế

Gần đây việc ứng dụng phân lớp dữ liệu y học ngày càng hoàn thiện trong việc tìm ra mối liên hệ giữa các triệu chứng lâm sàng, cận lâm sàng, giữa các bệnh với nhau để hỗ trợ chẩn đoán, điều trị và tiên lượng bệnh.

Trong chẩn đoán, phân lớp dữ liệu dùng để nhận dạng và phân loại mẫu trong các thuộc tính đa biến của bệnh nhân.

Trong điều trị, phân loại dữ liệu dùng để chọn lựa phương pháp điều trị phù hợp hiệu quả nhất và trong tiên lượng là dự đoán kết quả điều trị, phẫu thuật dựa trên những kết quả điều trị trước đó và tình trạng hiện tại của người bệnh.

Ngoài ra có thể hỗ trợ cảnh báo dịch bệnh.

Ứng dụng trong an ninh mạng

Phân lớp dữ liệu được ứng dụng trong việc phân loại các truy cập mạng, cảnh báo các tấn công mạng để người dùng và các nhà cung cấp dịch vụ đề phòng và có các biện pháp phù hợp bảo đảm an ninh mạng.

Ứng dụng trong các vấn đề xã hội

Phân lớp dữ liệu được ứng dụng trong quá trình xử lý các dư luận xã hội tích cực và tiêu cực để cơ quan quản lý đưa ra các chính sách phù hợp. Đồng thời có thể hỗ trợ phát hiện tội phạm, quản lý các đối tượng khủng bố nhằm tăng cường an ninh quốc gia, đảm bảo trật tự xã hội.

1.1.6. Các phương pháp phân lớp dữ liệu

Do ý nghĩa quan trọng trong các ứng dụng của bài toán phân lớp dữ liệu (1.1) – (1.2), rất nhiều các phương pháp khác nhau đã được đề xuất để xây dựng các mô hình phân lớp dữ liệu. Các phương pháp đó bắt nguồn từ những lĩnh vực nghiên cứu khác nhau và thường sử dụng các cách tiếp cận xây dựng mô hình rất đa dạng. Chúng có nhiều hình thức khác nhau và có thể được phân loại dựa vào các tiêu chí cơ bản sau:

- Cách thức tiền xử lý dữ liệu mẫu (đặc biệt đối với các trường hợp dữ liệu bị thiếu và nhiễu).

- Cách thức xử lý các kiểu thuộc tính khác nhau của dữ liệu mẫu (có thứ tự, rời rạc hoặc liên tục).

- Cách thức thể hiện của mô hình phân lớp dữ liệu (dưới dạng công thức toán học, bộ quy tắc hay luật quyết định phân lớp).

- Cách thức rút gọn, giảm số thuộc tính của dữ liệu cần thiết để cho ra quyết định phân lớp.

- Hiệu quả của bộ phân lớp xây dựng được đối với bài toán cụ thể được xem xét.

Tất cả các phương pháp tiếp cập xây dựng mô hình phân lớp dữ liệu khác nhau đều có khả năng phân lớp cho một mẫu dữ liệu mới chưa biết dựa vào những mẫu tương tự đã được học. Các phương pháp phân lớp dữ liệu tiêu biểu có thể kể đến bao gồm: Phương pháp dựa trên các phân tích, tổng hợp, thống kê, Phương pháp dựa trên tiếp cận tập thô và phương pháp sử dụng các kỹ thuật học máy.

Trong các phương pháp kể trên, phương pháp sử dụng các kỹ thuật học máy thường được sử dụng trong quá trình xây dựng các mô hình phân lớp và thu được nhiều kết quả tích cực. Đây cũng chính là chủ đề nghiên cứu của luận văn.

Trong mục tiếp theo, luận văn trình bày tổng quan về học máy sử dụng trong việc giải quyết bài toán phân lớp (1)-(2).

1.2. Tổng quan về học máy

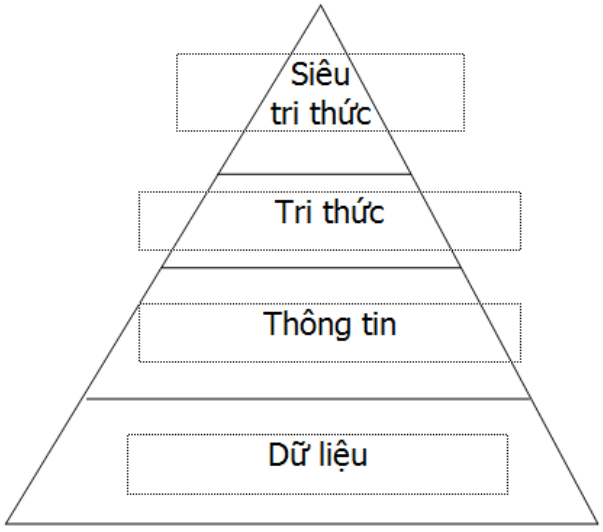
1.2.1. Khái niệm về học máy và phân loại các kỹ thuật học máy

a. Khái niệm về học máy

Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể .

Rất khó để định nghĩa một cách chính xác về học máy. “Học - learn” có ý nghĩa khác nhau trong từng lĩnh vực: tâm lý học, giáo dục, trí tuệ nhân tạo, …

Một định nghĩa rộng nhất: “học máy là một cụm từ dùng để chỉ khả năng một chương trình máy tính để tăng tính thực thi dựa trên những kinh nghiệm đã trải qua” hoặc “học máy là để chỉ khả năng một chương trình có thể phát sinh ra một cấu trúc dữ liệu mới khác với các cấu trúc dữ liệu cũ”. Lợi điểm của các phương pháp học máy là nó phát sinh ra các luật tường minh, có thể được sửa đổi, hoặc được huấn luyện trong một giới hạn nhất định. Các phương pháp học máy hoạt động trên các dữ liệu có đặc tả thông tin. Các thông tin được trình bày theo một cấu trúc gồm 4 mức được gọi là kim tự tháp tri thức như trong hình 1.5 dưới đây [12].



Hình 1.5. Mô hình kim tự tháp: Từ dữ liệu đến tri thức

Học máy là sự tự động của quy trình học và việc học thì tương đương với việc xây dựng những luật dựa trên việc quan sát trạng thái trên cơ sở dữ liệu và những sự chuyển hoá của chúng. Đây là lĩnh vực rộng lớn không chỉ bao gồm việc học từ mẫu, mà còn học tăng cường, học với "thầy", ... Các thuật toán học dựa trên bộ dữ liệu mẫu và những thông tin liên quan để làm đầu vào và trả về một mô hình diễn tả những kết quả học làm đầu ra.

Học máy kiểm tra những ví dụ trước đó và kiểm tra luôn cả những kết quả của chúng khi xuất và học làm cách nào để tái tạo lại những kết quả này và tạo nên những sự tổng quát hóa cho những trường hợp mới.

Nói chung, học máy sử dụng một tập hữu hạn dữ liệu được gọi là tập huấn luyện. Tập này chứa những mẫu dữ liệu mà nó được viết bằng mã theo một cách nào đó để máy có thể đọc và hiểu được. Tuy nhiên, tập huấn luyện bao giờ cũng hữu hạn do đó không phải toàn bộ dữ liệu sẽ được học một cách chính xác.

Một tiến trình học máy gồm 2 giai đoạn:

Giai đoạn học: hệ thống phân tích dữ liệu và nhận ra sự mối quan hệ (có thể là phi tuyến hoặc tuyến tính) giữa các đối tượng dữ liệu. Kết quả của việc học có thể là: nhóm các đối tượng vào trong các lớp, tạo ra các luật, tiên đoán lớp cho các đối tượng mới.

Giai đoạn thử nghiệm (testing): Mối quan hệ (các luật, lớp...) được tạo ra phải được kiểm nghiệm lại bằng một số hàm tính toán thực thi trên một phần của tập dữ liệu huấn luyện hoặc trên một tập dữ liệu lớn.

b. Phân loại các kỹ thuật học máy

Các thuật toán học máy được chia làm 3 loại: học có giám sát, học không giám sát và học bán giám sát.

Học có giám sát

Đây là cách học từ những mẫu dữ liệu mà ở đó các kỹ thuật học máy giúp hệ thống xây dựng cách xác định những lớp dữ liệu. Hệ thống phải tìm một sự mô tả cho từng lớp (đặc tính của mẫu dữ liệu). Người ta có thể sử dụng các luật phân loại

hình thành trong quá trình học và phân lớp để có thể sử dụng dự báo các lớp dữ liệu sau này.

Xét bài toán phân lớp dữ liệu (1)-(2).

Thuật toán học máy giám sát tìm kiếm không gian của những giả thuyết có thể có, ký hiệu là H. Đối với công việc phân lớp có thể xem giả thuyết như một tiêu chí phân lớp. Thuật toán học máy tìm ra những giả thuyết bằng cách khám phá ra những đặc trưng chung của những ví dụ mẫu thể hiện cho mỗi lớp. Kết quả nhận được thường ở dạng luật (Nếu ... thì). Khi áp dụng cho những mẫu dữ liệu mới, cần dựa trên những giả thuyết đã có để dự báo những phân lớp tương ứng của chúng.

Đối với một hay nhiều giả thuyết, cần tìm ước lượng tốt nhất dưới dạng một hàm $f : X \rightarrow C$. Nếu như không gian giả thuyết lớn, thì cần một tập dữ liệu huấn luyện đủ lớn nhằm tìm kiếm một hàm xấp xỉ tốt nhất f.

Học không giám sát

Đây là việc học từ quan sát và khám phá. Hệ thống khai thác dữ liệu được ứng dụng với những đối tượng nhưng không có lớp được định nghĩa trước, mà để nó phải tự hệ thống quan sát những mẫu và nhận ra mẫu. Hệ thống này dẫn đến một tập lớp, mỗi lớp có một tập mẫu được khám phá trong tập dữ liệu. Học không giám sát còn gọi là học từ quan sát và khám phá.

Trong trường hợp chỉ có ít, hay gần như không có tri thức về dữ liệu đầu vào, khi đó một hệ thống học không giám sát sẽ khám phá ra những phân lớp của dữ liệu, bằng cách tìm ra những thuộc tính, đặc trưng chung của những mẫu hình thành nên tập dữ liệu. Một thuật toán học máy có giám sát luôn có thể biến đổi thành một thuật toán học máy không giám sát khi sử dụng nhãn lớp được lựa chọn song song với quá trình học.

Đối với một bài toán mà những mẫu dữ liệu được mô tả bởi n đặc trưng, người ta có thể chạy thuật toán học có giám sát n-lần, mỗi lần với một đặc trưng khác nhau đóng vai trò thuộc tính lớp, mà chúng ta đang tiên đoán. Kết quả sẽ là n tiêu chí phân lớp (n bộ phân lớp), với hy vọng là ít nhất một trong n bộ phân lớp đó là đúng.

Có rất nhiều thuật toán học không giám sát được ra đời và phát triển nhằm giải quyết bài toán phân cụm phục vụ khai thác hiệu quả nguồn dữ liệu chưa gán nhãn nhiều và rất đa dạng. Việc lựa chọn sử dụng thuật toán nào tuỳ thuộc vào dữ liệu và mục đích của từng bài toán. Trong đó các thuật toán thường được sử dụng như: K-means, Luật kết hợp, ...

Học bán giám sát

Học bán giám sát là các thuật toán học tích hợp từ học giám sát và học không giám sát. Học bán giám sát sử dụng cả dữ liệu đã gán nhãn và chưa gán nhãn để huấn luyện - điển hình là một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn.

Nội dung chính của học bán giám sát là hệ thống sử dụng một tập huấn luyện (training set) gồm 2 phần: các ví dụ huấn luyện có nhãn, thường với số lượng (rất) ít, và các ví dụ học không có nhãn, thường với số lượng (rất) nhiều. Các dữ liệu gán nhãn thường hiếm, đắt và rất mất thời gian xử lý, đòi hỏi sự nỗ lực của con người. Trong khi đó, dữ liệu chưa gán nhãn thì có rất nhiều, nhưng để sử dụng vào mục đích cụ thể thì rất khó. Vì vậy ý tưởng kết hợp giữa dữ liệu chưa gán nhãn và dữ liệu đã gán nhãn để xây dựng một tập phân lớp tốt hơn là nội dung chính của học bán giám sát. Bởi vậy học bán giám sát là một ý tưởng tốt để giảm bớt công việc của con người và cải thiện độ chính xác lên mức cao hơn.

Một thuật toán học bán giám sát được sử dụng sẽ học các mẫu có nhãn, sau đó tiến hành gán nhãn cho một số (có lựa chọn) các mẫu không có nhãn - một cách hợp lý, có đánh giá chất lượng công việc hay độ chính xác. Tiếp theo, chọn các ví dụ vừa được gán nhãn có độ tin cậy cao (vượt trên một ngưỡng chọn trước) đưa vào kết hợp với tập dữ liệu có nhãn, tạo thành một tập dữ liệu huấn luyện mới.

Áp dụng một phương pháp kiểm thử (có thể kết hợp với một tập dữ liệu đã biết trước nhãn) để đánh giá hiệu năng/độ chính xác của mô hình.

Học bán giám sát đứng giữa học không giám sát (không có bất kì dữ liệu đã được nhãn nào) và có giám sát (toàn bộ dữ liệu đều được gán nhãn). Việc học bán

giám sát tận dụng những ưu điểm của việc học giám sát và học không giám sát và loại bỏ những khuyết điểm thường gặp trên hai kiểu học này.

1.2.2. Ứng dụng học máy xây dựng mô hình phân lớp dữ liệu

Học máy có ứng dụng rộng khắp trong các ngành khoa học và công nghệ, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ.

Qua các nội dung trình bày ở trên, có thể nhận thấy sự tương đồng giữa quá trình học máy và quá trình phân lớp dữ liệu. Do đó, hầu hết các kỹ thuật học máy đều có thể sử dụng để xây dựng các mô hình phân lớp dữ liệu.

Các phương pháp phân lớp dữ liệu tiêu biểu dựa trên kỹ thuật học máy có thể kể đến bao gồm:

- Phương pháp Cây quyết định.
- Phương pháp Bayes (Suy luận Bayes, mạng bayes).
- Phương pháp Máy vectơ hỗ trợ (SVM).
- Phương pháp Mạng no-ron nhân tạo (Artificial Neural Network - ANN).

Trong luận văn sẽ nghiên cứu và thử nghiệm ba phương pháp phân lớp dữ liệu là Phương pháp Cây quyết định, Phương pháp Bayes và Phương pháp Máy vectơ hỗ trợ.

1.3. Giới thiệu chung về học sâu

1.3.1. Khái niệm về học sâu

Ban đầu thuật ngữ học sâu (*Deep Learning*) xuất hiện trong quá trình xây dựng các mạng nơ-ron sâu (deep neural networks) nhằm xử lý tốt hơn các bài toán phức tap. Trong mạng nơ-ron sâu sẽ bao gồm nhiều lớp. Ví dụ, mô hình mạng nơ-ron sâu Google LeNet để nhận dạng hình ảnh có 22 lớp. Khi đó, đầu ra của một lớp nào đó sẽ được sử dung như là đầu vào của lớp kế tiếp.

Do đó, quá trình học máy sẽ *sâu* hơn và hiệu quả hy vọng sẽ đạt được cao hơn.

Như vậy, học sâu là một chi của ngành học máy dựa trên một tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bằng cách khác bao gồm nhiều biến đổi phi tuyến.

Các quá trình học sâu có thể mô tả như trong hình 1.6 [19].

Hình 1.6. Các quá trình học sâu

Một trong những hứa hẹn của học sâu là thay thế các phương pháp thủ công bằng các thuật toán hiệu quả đối với học không giám sát hoặc bán giám sát và tính năng phân cấp. Học sâu vượt trội hơn so với học máy truyền thống trong xử lý các vấn đề phức tạp như nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên, phân loại hình ảnh, ….

1.3.2. Hướng tiếp cận học sâu

Hướng tiếp cận học sâu đầu tiên thường được kể đến là các mạng nơ-ron sâu. Dưới đây, luận văn liệt kê một số dạng mạng nơ-ron sâu tham khảo trên mạng Internet.

Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) được xây dựng để xử lý hình ảnh. CNN thực hiện so sánh hình ảnh theo từng mảnh (còn gọi là các feature). Khi xem xét một hình ảnh mới, CNN không biết chính xác các feature nào sẽ khớp nên sẽ thử tất cả các mảnh có thể. Khi tính toán sự khớp của một feature trên toàn bộ ảnh, đã tạo thành một filter (bộ lọc). Các bộ lọc được xây dựng nhờ sử dụng công thức tích chập.

Mạng nơ-ron lặp

Mạng nơ-ron lặp (Recurrent neural network - RNN) là một mạng nơ-ron nhiều lớp, có thể lưu trữ thông tin trong các nút bối cảnh, cho phép nó tìm hiểu các chuỗi dữ liệu và xuất ra một số hoặc một chuỗi khác. Nói một cách đơn giản, đó là một mạng nơ-ron nhân tạo có kết nối giữa các nơ-ron bao gồm các vòng. RNN rất

phù hợp để xử lý dữ liệu đầu vào các chuỗi. Do đó, RNN thường được lựa chọn để xử lý văn bản hoặc tiếng nói.

Mạng nơ-ron chuyển đổi

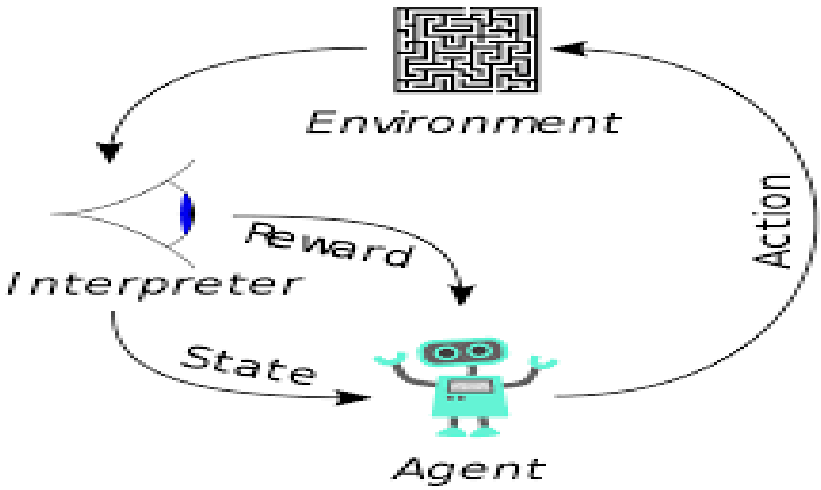
Mạng nơ-ron chuyển đổi (Convolutional neural networks - CNN) là một mạng nơ-ron nhiều lớp với kiến trúc độc đáo được thiết kế để trích xuất các tính năng ngày càng phức tạp của dữ liệu ở mỗi lớp để xác định đầu ra phù hợp.

CNN chủ yếu được sử dụng khi có xử lý các bộ dữ liệu phi cấu trúc và cần trích xuất thông tin từ nó.

Hướng tiếp cận học sâu tiếp theo là học sâu củng cố (hay học tăng cường).

Học tăng cường

Quá trình học tăng cường có thể mô tả như trong hình 1.7 [19].



Hình 1.7. Quá trình học tăng cường

Thông qua kỹ thuật học tăng cường, phần mềm hoặc máy có thể tự học cách hoạt động trong các môi trường. Trong quá trình học tăng cường, máy không được cung cấp các hướng dẫn về kết quả. Thay vào đó, máy tuân theo cơ chế thử nghiệm và lỗi để xây dựng và lựa chọn các kết quả phù hợp.

Một hướng tiếp cận học sâu khác là kết hợp nhiều thuật toán học máy với nhau để có được độ chính xác cao hơn so với chỉ sử dụng một thuật toán duy nhất. Các phương pháp Ensemble và AdaBoost (Freund & Schapire, 1995) là các ví dụ điển hình cho hướng tiếp cận này.

Phương pháp Ensemble kết hợp các mô hình khác nhau với mục tiêu đạt được tỷ lệ lỗi phân loại thấp hơn so với sử dụng một mô hình duy nhất. Khái niệm "mô hình" trong các phương pháp kết hợp được hiểu theo nghĩa rộng, bao gồm không chỉ việc thực hiện các thuật toán học khác nhau, hoặc tạo ra nhiều tập huấn luyện cho cùng một thuật toán học, mà còn là sinh ra các bộ phân loại chung kết hợp với nhau để nâng cao độ chính xác phân loại

AdaBoost là một bộ phân loại mạnh phi tuyến phức. AdaBoost hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để tạo nên một bộ phân loại mạnh và sử dụng trọng số để đánh dấu các mẫu khó nhận dạng.

1.4. Kết luận chương 1

Trong chương 1 của luận văn đã giới thiệu bài toán phân lớp dữ liệu và khảo sát quy trình phân lớp dữ liệu cũng như các độ đo đánh giá các mô hình phân lớp dữ liệu và các ứng dụng khác nhau của phân lớp dữ liệu.

Trong chương này luận văn cũng trình bày tổng quan về các học máy và giới thiệu về học sâu.

Trong chương tiếp theo luận văn sẽ nghiên cứu ba thuật toán học máy để xây dựng mô hình phân lớp là cây quyết định, Bayes và máy vectơ hỗ trợ.

CHƯƠNG 2. NGHIÊN CỨU MỘT SỐ THUẬT TOÁN HỌC MÁY

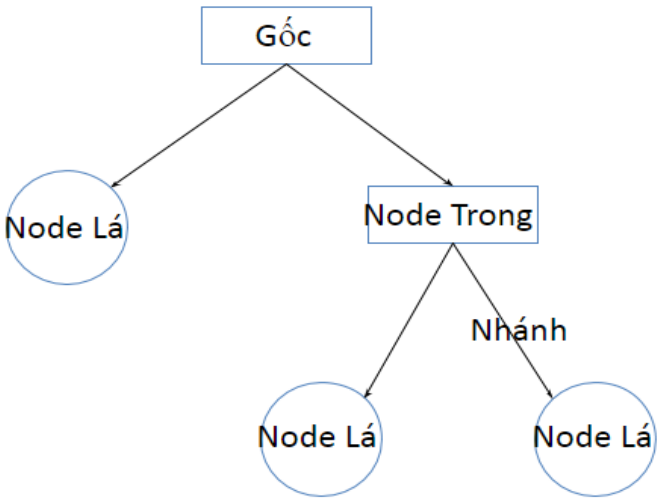
Trong chương 2 luận văn sẽ nghiên cứu chi tiết một số kỹ thuật học máy để giải quyết bài toán phân lớp dữ liệu và một số vấn đề liên quan. Các thuật toán sẽ khảo sát gồm: Cây quyết định, Bayes và Máy vectơ hỗ trợ.

2.1. Khảo sát thuật toán cây quyết định và các vấn đề liên quan

2.1.1. Giới thiệu phương pháp

Cây quyết định là một cấu trúc ra quyết định có dạng cây. Cây quyết định nhận đầu vào là một bộ giá trị các thuộc tính mô tả một đối tượng hay một tình huống và trả về một giá trị rời rạc. Mỗi bộ thuộc tính đầu vào được gọi là một mẫu hay một ví dụ, đầu ra gọi là lớp hay nhãn phân lớp. Khi đó, với tập thuộc tính đầu vào được cho dưới dạng véc tơ x, nhãn phân lớp đầu ra được ký hiệu là y thì cây quyết định có thể xem như một hàm f(x) = y.

Cây quyết định được biểu diễn dưới dạng một cấu trúc cây như trong Hình 2.1 dưới đây.



Hình 2.1. Mô hình cây quyết định

Trong cây quyết định, mỗi nút trung gian, tức là nút khác với nút lá và nút gốc, tương ứng với phép kiểm tra một thuộc tính. Mỗi nhánh phía dưới của nút đó tương ứng với một giá trị của thuộc tính hay một kết quả của phép thử. Khác với nút

trung gian, nút lá không chứa thuộc tính mà chứa nhãn phân lớp. Để xác định nhãn phân lớp cho một dữ liệu mẫu nào đó, ta cho dữ liệu mẫu chuyển động từ gốc cây về phía nút lá. Tại mỗi nút, thuộc tính tương ứng với nút được kiểm tra, tùy theo giá trị của thuộc tính đó mà dữ liệu mẫu được chuyển xuống nhánh tương ứng bên dưới. Quá trình này lặp lại cho đến khi dữ liệu mẫu tới được nút lá và được nhận nhãn phân lớp là nhãn của nút lá tương ứng.

Quá trình xây dựng cây quyết định thường được thực hiện như sau:

(1) Bắt đầu từ nút đơn biểu diễn tất cả các mẫu.

(2) Nếu các mẫu thuộc về cùng một lớp, nút đang xét trở thành nút lá và được gán nhãn bằng lớp đó.

(3) Ngược lại, dùng độ đo thuộc tính để chọn thuộc tính sẽ phân tách tốt nhất các mẫu vào các lớp.

(4) Một nhánh được tạo cho từng giá trị của thuộc tính được chọn và các mẫu được phân hoạch theo.

(5) Lặp lại tiến trình trên để tạo cây quyết định.

(6) Tiến trình kết thúc chỉ khi bất kỳ điều kiện nào sau đây là đúng.

- Tất cả các mẫu cho một nút cho trước đều thuộc về cùng một lớp.
- Không còn thuộc tính nào mà mẫu có thể dựa vào để phân hoạch xa hơn.
- Không còn mẫu nào cho nhánh.

Tuy nhiên, nếu không chọn được thuộc tính phân loại hợp lý tại mỗi nút, có thể sẽ tạo ra cây quyết định rất phức tạp. Trong thực tế, thường sử dụng hai cách sau để tạo được cây quyết định phù hợp:

- Dừng phát triển cây sớm hơn bình thường, trước khi đạt tới điểm phân lớp hoàn hảo tập dữ liệu huấn luyện.

- Sử dụng các kỹ thuật “cắt”, “tỉa” cây phù hợp.

Trong các mục tiếp theo, luận văn sẽ khảo sát một số kỹ thuật xây dựng cây quyết định.

2.1.2. Xây dựng cây quyết định dựa trên Entropy

Khái niệm entropy của một tập S được định nghĩa trong lý thuyết thông tin là số lượng mong đợi các bít cần thiết để mã hóa thông tin về lớp của một thành viên rút ra một cách ngẫu nhiên từ tập S. Trong trường hợp tối ưu, mã có độ dài ngắn nhất. Theo lý thuyết thông tin, mã có độ dài tối ưu là mã gán $-\log_2 p$ bits cho thông điệp có xác suất là p.

Trong trường hợp S là tập mẫu thì mỗi thành viên của S là một mẫu. Mỗi mẫu thuộc một lớp hay có một giá trị phân loại. Giả sử các mẫu trong tập S thuộc về một trong c lớp, trong đó lớp thứ i ($1 \leq i \leq c$) có tỷ lệ là p_i.

Độ đo Entropy của tập mẫu S được định nghĩa bởi công thức sau [15]:

$$\text{Entropy}(S) = \sum_{i=1}^{C} - p_i \log_2 p_i \qquad (2.1)$$

Về bản chất, độ đo Entropy phản ánh tính không đồng nhất của tập mẫu S so với các lớp có trong i. Entropy là một số đo để đo độ pha trộn của một tập mẫu.

Trên cơ sở đó, ta sẽ định nghĩa một phép đo hiệu suất phân loại các mẫu của một thuộc tính. Phép đo này gọi là lượng thông tin thu thêm và ký hiệu là Gain.

Một cách chính xác hơn, Gain(S,A) của thuộc tính A, trên tập S, được định nghĩa như sau:

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

Trong đó Values(A) là tập hợp có thể có các giá trị của thuộc tính A, và S_v là tập con của S chứa các mẫu có thuộc tính A mang giá trị v. Giá trị Gain(S, A) được sử dụng làm độ đo lựa chọn thuộc tính phân chia dữ liệu tại mỗi nút trong quá trình xây dựng cây quyết định. Thuộc tính được chọn là thuộc tính cho lượng thông tin thu thêm lớn nhất.

Các thuật toán xây dựng cây quyết định dựa trên Entropy có thể tóm tắt như sau [15]*:*

- Với mỗi thuộc tính A chưa sử dụng tính Gain (S, A) theo công thức trên;

- Chọn thuộc tính P sao cho Gain(S, P) có giá trị lớn nhất trong các thuộc tính A kể trên;

- Gán node tương ứng với thuộc tính P.

Các thuật toán xây dựng cây quyết định ID3 (Iterative Dichotomiser 3), C4.5, J48 là các thuật toán dựa trên Entropy. Trong đó, ID3 được Quinlan xây dựng vào năm 1979. Sau đó, J. Ross Quinlan đã phát triển ID3 thành C4.5. Thuật toán J48 là một dạng của C4.5 thường được cài đặt sử dụng trong thực tế.

2.1.3. Đánh giá phương pháp

Mô hình phân lớp dữ liệu sử dụng cây quyết định có các ưu điểm sau đây.

- Cây quyết định tự giải thích và khi được gắn kết lại, chúng có thể dễ dàng tự sinh ra. Nói cách khác, nếu cây quyết định mà có số lượng nút lá vừa phải thì người không chuyên cũng dễ dàng hiểu được nó. Hơn nữa, cây quyết định cũng có thể chuyển sang tập luật. Vì vậy, cây quyết định được xem như là dễ hiểu, dễ sử dụng khi phân lớp dữ liệu.

- Cây quyết định có thể xử lý được nhiều kiểu các thuộc tính đầu vào. Cây quyết định được xem như là một phương pháp phi tham số.

Bên cạnh đó, cây quyết định cũng có những nhược điểm sau đây:

- Khi cây quyết định sử dụng phương pháp "chia để trị", chúng có thể thực hiện tốt nếu tồn tại một số thuộc tính liên quan chặt chẽ với nhau, nhưng sẽ khó khăn nếu một số tương tác phức tạp xuất hiện.

- Các đặc tính liên quan của cây quyết định dẫn đến những khó khăn khác như là độ nhạy với tập huấn luyện, các thuộc tính không phù hợp, hay có nhiễu.

2.2. Khảo sát thuật toán Bayes và các vấn đề liên quan

2.2.1. Giới thiệu phương pháp

Ý tưởng cơ bản của cách tiếp cận phân lớp dữ liệu Bayes là sử dụng công thức Bayes về xác suất có điều kiện để lựa chọn kết quả phân lớp là sự kiện có xác suất lớn nhất.

Công thức Bayes:

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)} \qquad (2.2)$$

Trong đó:

- H (Hypothesis) là giả thuyết và E (Evidence) là chứng cứ hỗ trợ cho giả thuyết H.

- P(E|H): xác suất E xảy ra khi H xảy ra (xác suất có điều kiện, khả năng của E khi H đúng) thường gọi là xác suất tiên nghiệm.

- P(H|E): xác suất hậu nghiệm của H nếu biết E.

Một số thuật toán phân lớp dữ liệu được đề xuất dựa trên công thức (2.2). Trong mục tiếp theo, luận văn sẽ khảo sát thuật toán Naive Bayes và mạng Bayes.

2.2.2. Thuật toán Naïve Bayes

Thuật toán phân lớp Naive Bayes (Naive Bayes Classification - NBC) thường được gọi ngắn gọn là thuật toán là Naive Bayes [19]. Thuật toán Naive Bayes dựa trên định lý Bayes (2.2) để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê.

Xét bài toán phân lớp dữ liệu (1.1)-(1.2). Mô hình phân lớp dữ liệu Bayes được xây dựng dựa trên công thức (2.2) với mỗi lớp dữ liệu $c_i \in C = \{c_1, c_2, \ldots, c_m\}$ như sau:

- Lựa chọn sự kiện H = "Dữ liệu mẫu thuộc lớp c_i"; E = "Thỏa mãn điều kiện đối với một số thuộc tính thuộc A".

- Tính các xác suất P(E), P(H) và P(E|H) trong tập các mẫu dữ liệu huấn luyện.

- Tính xác suất P(H|E) theo công thức (2.2).

- Lựa chọn E sao cho xác suất P(H|E) đạt giá trị lớn nhất.

Để thực hiện phân lớp đối với dữ liệu mới $z = (z_1, z_2, \ldots, z_k)$ ta sẽ tiến hành như sau:

- Tính xác suất P(H|(z_1, z_2, …, z_k)) theo công thức (2.2) theo nghĩa các thuộc tính của Z xét trên E tương ứng;

- Xuất kết quả xếp dữ liệu Z vào lớp c_i ứng với lớp có xác suất tính được ở bước trên là lớn nhất.

2.2.3. Mạng Bayes

Mạng Bayes (Bayesian network) là một mô hình xác suất dạng đồ thị [19]. Mạng Bayes là một đồ thị có hướng không chứa chu trình bao gồm:

- Các nút biểu diễn các biến ngẫu nhiên (gọi tắt là biến);
- Các cung biểu diễn các quan hệ phụ thuộc thống kê giữa các biến và phân phối xác suất địa phương cho mỗi giá trị nếu cho trước giá trị của các cha của nó.

Nếu có một cung hướng từ nút A tới nút B thì B gọi là con của A và A được gọi là cha của B. Khi đó, biến B phụ thuộc trực tiếp vào biến A. Với mỗi biến B, ký hiệu tập hợp các cha của B là Parent(B).

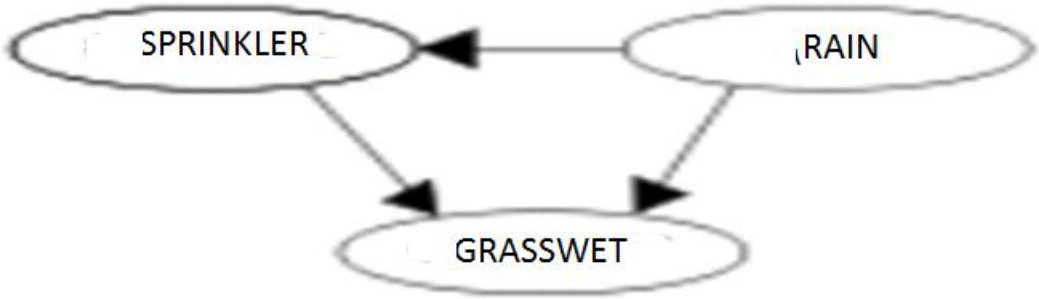
Với mỗi biến X_i ($1 \leq i \leq n$) định nghĩa xác suất có điều kiện phụ thuộc của các biến là tích của các xác suất địa phương:

$$P(X_1, \ldots, X_n) = \prod P(X_i|Parent(X_i), 1 \leq i \leq n) \quad (2.3)$$

Nếu biến X_i không có cha thì xác suất địa phương của X_i là không có điều kiện. Ngược lại, xác suất địa phương của X là có điều kiện. Nếu biến được biểu diễn bởi một nút được quan sát, thì ta nói rằng nút đó là một chứng cứ (evidence node).

Xét một mạng Bayes đơn giản mô tả sự việc đất bị ướt (GRASSWET - G) được mô tả trong hình 2.2 dưới đây.

Có hai lý do để xảy ra hiện tượng này là do được tưới nước (SPRINKLER - G) hoặc do trời mưa (RAIN - R). Trong trường hợp này, mỗi biến có hai trạng thái có thể là: T (đúng) hoặc F (sai). Khi đó, xác suất phụ thuộc có điều kiện được tính theo công thức (2.3) như sau: P(G,S,R) = P(G | S,R).P(S | R).P(R).



Hình 2.2. Mô hình mạng Bayes

Giả sử có các số liệu quan sát được như sau:

P(R=T) = 0,2; P(R= F) = 0,8
P(R= F, S= F)= 0,6; P(R= F, S= T)= 0,4; P(R= T, S= F)= 0,99; P(R= T, S= T)= 0,01
P(R= F, S= F, G= F)= 1,0; P(R= F, S= F, G= T)= 0,0; P(R= F, S= T, G= F)= 0,1; P(R= F, S= T, G= T)= 0,9; P(R= T, S= F, G= F)= 0,2; P(R= T, S= F, G= T)= 0,8; P(R= T, S= T, G= F)= 0,1; P(R= T, S= T, G= T)= 0,9

Từ đó, có thể tính P(RAIN=T | GRASSWET=T) = 35.77%. Như vậy, nếu đất bị ướt thì có khả năng trời mưa là 35,77%.

Xét bài toán phân lớp dữ liệu (1.1)-(1.2). Mô hình phân lớp dữ liệu sử dụng mạng Bayes được xây dựng dựa trên công thức (2.3) với mỗi lớp dữ liệu $c_i \in C = \{c_1, c_2, \ldots, c_m\}$ sao cho xác suất địa phương tại nút biểu diễn c_i có giá trị lớn nhất.

2.2.4. Đánh giá phương pháp

So với các phương pháp khác, phương pháp phân lớp dữ liệu Bayes lập luận theo kinh nghiệm được tích lũy và áp dụng vào mô hình phân lớp đối tượng khá linh hoạt và phù hợp với đặc trưng của bài toán cụ thể. Các cơ chế ước lượng trong phương pháp này cũng gần gũi với cách suy luận thông thường. Phương pháp phân lớp dữ liệu Bayes được ứng dụng rất rộng rãi bởi tính dễ hiểu và dễ triển khai.

Tuy nhiên, phương pháp phân lớp dữ liệu Bayes cho hiệu quả không cao trong trường hợp tập dữ liệu mẫu có độ phức tạp lớn và các thuộc tính của dữ liệu mẫu có quan hệ phụ thuộc hoặc không đầy đủ. Trong những trường hợp này, có thể sử dụng mạng Bayes.

2.3. Khảo sát thuật toán máy vectơ hỗ trợ và các vấn đề liên quan

2.3.1. Giới thiệu phương pháp

Máy vector hỗ trợ (Support Vector Machines - SVM) được Cortes và Vapnik giới thiệu vào năm 1995 trên cơ sở mở rộng từ chuyên đề lý thuyết học thống kê (Vapnik 1982), dựa trên nguyên tắc tối thiểu rủi ro cấu trúc (structural risk minimization). Ý tưởng chính của SVM để giải quyết bài toán phân lớp (1.1)-(1.2) là ánh xạ tập dữ liệu mẫu thành các vector điểm trong không gian vector R^d và tìm các siêu phẳng có hướng để chia tách chúng thành các lớp khác nhau.

Định lý 2.1 sau đây đảm bảo cơ sở toán học cho SVM [5].

Định lý 2.1: Cho tập hợp gồm m điểm trong không gian R^d. Ta chọn một điểm nào đó trong chúng làm điểm gốc và tạo thành m-1 vector điểm. Khi đó m điểm đã cho có thể được phân tách bởi một siêu phẳng có hướng khi và chỉ khi tập hợp các vector điểm là độc lập tuyến tính.

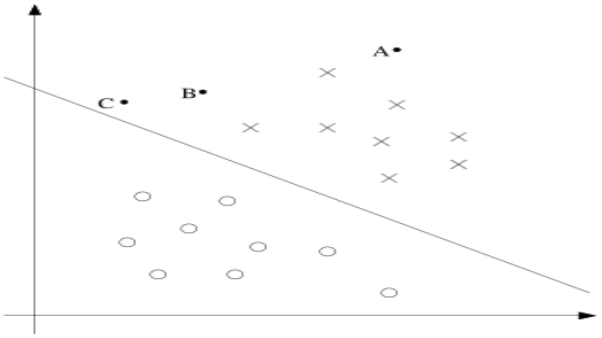
Khoảng cách của điểm dữ liệu gần nhất của mỗi lớp đến siêu phẳng phân tách gọi là biên (hay lề). Trong số các siêu phẳng thỏa mãn định lý 2.1, siêu phẳng tối ưu có biên lớn nhất sẽ được lựa để phân tách các điểm. Các kỹ thuật SVM nhằm nghiên cứu xây dựng các siêu phẳng tối ưu này một cách hiệu quả nhất.

Xét bài toán phân lớp dữ liệu (1.1)-(1.2). Số lượng m các lớp trong tập C có thể nhận giá trị bất kỳ lớn hơn 1. Tuy nhiên, có thể quy bài toán phân lớp tổng quát về bài toán phân lớp dữ liệu với m =2. Bài toán này được gọi là phân lớp nhị phân.

Trong bài toán phân lớp nhị phân, các dữ liệu mẫu x_i được biểu diễn dưới dạng véc tơ trong không gian véc tơ R^d. Các mẫu dương là các mẫu x_i thuộc lĩnh vực quan tâm được gán nhãn $y_i = +1$; các mẫu âm là các mẫu x_i không thuộc lĩnh vực quan tâm được gán nhãn $y_i = -1$.

Cần xác định một siêu phẳng ranh giới có biên lớn nhất để phân tách tập hợp các mẫu thành hai lớp dữ liệu có nhãn tương ứng là +1 và -1.

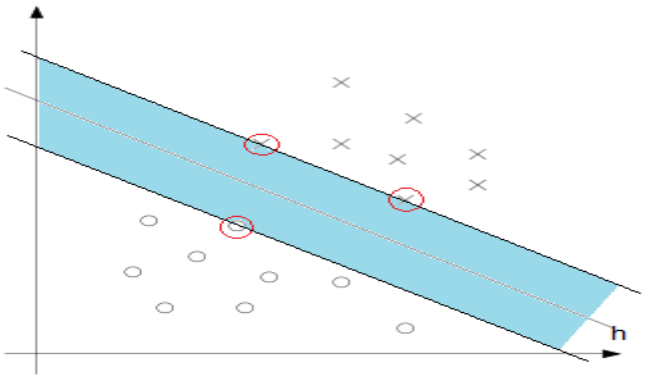
Độ chính xác của bộ phân lớp SVM phụ thuộc vào độ lớn của biên. Tầm quan trọng của biên được minh họa trong hình 2.3.



Hình 2.3. Hình Tầm quan trọng của biên đối với siêu phẳng phân tách

Trong hình 2.3, có thể nhận thấy rằng, các điểm có khoảng cách đến siêu phẳng lớn như điểm A thì khi được yêu cầu phân lớp cho A dễ dàng gán nó vào lớp +1. Trong khi đó với điểm C ngay sát siêu phẳng sẽ được dự đoán thuộc lớp +1 nhưng có thể thành lớp -1 nếu có một sự thay đổi nhỏ của siêu phẳng. Điểm B nằm giữa hai trường hợp này. Tổng quát, tất cả các điểm cách siêu phẳng một khoảng đủ lớn đều được phân lớp một cách chính xác. Ngược lại, thì kết quả phân lớp có thể không chính xác. Vì vậy, khoảng cách biên càng lớn thì siêu phẳng quyết định càng tốt và độ chính xác phân loại càng cao.

Trong hình 2.4 [5] mô tả trường hợp siêu phẳng phân tách có biên tối ưu.



Hình 2.4. Ví dụ về biên tối ưu của siêu phẳng phân tách

Quan sát hình 2.4, có thể nhận thấy các điểm được khoanh tròn chính là các điểm gần siêu phẳng phân tách h nhất và được gọi là các vector hỗ trợ (Support vectors). Hai siêu phẳng song song với h và đi qua các vector hỗ trợ còn được gọi là lề (margin). Phần được tô màu là khoảng cách từ h đến các điểm gần nhất của hai lớp được gọi là biên.

Mỗi siêu phẳng có thể được biểu diễn dưới dạng:

$$w.x + b = 0. \quad (2.4)$$

Trong đó:

- w là vector pháp tuyến của siêu phẳng.
- b là một số thực với $\frac{b}{||w||}$ là khoảng cách giữa gốc tọa độ và siêu phẳng theo hướng vector pháp tuyến w.
- w.x biểu thị cho tích vô hướng của w và x.

Hàm $f(x_i) = \vec{w}.\vec{x_i} + b$ tương đương với

$$f(x_i) = w_1.x_{i1} + w_2x_{i2} + \ldots + w_nx_{in} + b \quad (2.5)$$

Đặt

$$h(\overrightarrow{x_i}) = sign\, f(x_i) = sign\,(\vec{w}.\vec{x_i} + b) = \begin{cases} +1, & \vec{w}.\vec{x_i} + b \geq 0 \\ -1, & \vec{w}.\vec{x_i} + b < 0 \end{cases} \quad (2.6)$$

Dữ liệu sẽ được phân loại dựa vào $h(\overrightarrow{x_i})$ thành hai lớp -1 và +1 theo (2.6).

Mục tiêu của SVM là tìm w và b sao cho siêu phẳng phân tách tập dữ liệu huấn luyện dạng w.x + b = 0 có lề lớn nhất. Trong các mục tiếp theo, luận văn sẽ khảo sát các kỹ thuật SVM để đạt được mục tiêu đặt ra.

2.3.2. Thuật toán SVM tuyến tính với tập dữ liệu phân tách được

Giả sử tập huấn luyện $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ có thể phân tách tuyến tính được. Hai lề của siêu phẳng w.x + b = 0 sẽ là:

Lề cộng: w.x + b = +1. Lề trừ: w.x + b = -1

Công thức tính khoảng cách từ một điểm x_i tới một siêu phẳng w.x + b = 0 là:

$$\frac{|w.x_i + b|}{||w||} \quad (2.7)$$

Với ||w|| là độ dài của w:

$$\|w\| = \sqrt{< w.w >} = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2} \quad (2.8)$$

Để tính độ rộng của biên, tính tổng khoảng cách từ 2 lề cộng và lề trừ đến siêu phẳng w.x + b = 0. Chọn 2 điểm x_1 và x_2 trên siêu phẳng w.x + b = 0 nghĩa là:

$$w.x_1 + b = 0 \text{ và } w.x_2 + b = 0 \quad (2.9)$$

Theo (2.7) khoảng cách từ x_1 đến lề cộng là:

$$d_+ = \frac{|w.x_i + b - 1|}{||w||} \quad (2.10)$$

Từ đó:

$$d_+ = \frac{|w.x_i + b - 1|}{||w||} = \frac{|-1|}{||w||} = \frac{1}{||w||} \quad (2.11)$$

Tương tự với d_-, suy ra độ rộng biên (m):

$$m = d_+ + d_- = \frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{||w||} \quad (2.12)$$

Như vậy việc tìm siêu phẳng tối ưu tương đương với việc tìm cực đại hóa

$$\frac{2}{||w||}$$

với điều kiện:

$$\begin{cases} w.x_i + b \geq 1, & y_i = 1 \\ w.x_i + b \leq -1, & y_i = -1 \end{cases} \quad (2.13)$$

$\forall i = 1,\ldots,n$

Tương đương:

$$\min_{w,b} ||w|| \text{ với ràng buộc } y_i(w.x_i + b) \geq 1 \ \forall \ i = 1,\ldots,n \quad (2.14)$$

Bài toán này rất khó giải nhưng nếu chuyển mục tiêu từ ||w|| sang $\frac{1}{2}||w||^2$ thì bài toán chuyển sang bài toán quy hoạch lồi (hàm mục tiêu lồi, ràng buộc tuyến tính) có nghiệm tối ưu tương đương với bài toán cũ. Vậy cần giải bài toán:

$$\min_{w,b} \frac{1}{2}||w||^2$$

với ràng buộc $y_i(< w.x_i > + b) \geq 1 \ \forall \ i = 1,\ldots,n$ (2.15)

Để giải bài toán trên cần xét bài toán cực tiểu hóa f(x) với điều kiện g(x) ≤ 0.

Điều kiện cần để x_0 là một lời giải:

$$\begin{cases} \dfrac{\partial}{\partial x}\big(f(x) + \alpha g(x)\big)\Big|_{x = x_0} = 0 \\ g(x) \le 0 \end{cases} \tag{2.16}$$

Với α là hệ số nhân Lagrange.

Trong trường hợp có nhiều ràng buộc đẳng thức $g_1(x) = 0$ (i = 1,…,n) thì cần phải có hệ số nhân Lagrange cho mỗi ràng buộc:

$$\begin{cases} \dfrac{\partial}{\partial x}\left(f(x) + \sum_{i=1}^{n} \alpha_i g_i(x)\right)\Big|_{x = x_0} = 0 \\ g_i(x) \le 0 \end{cases} \tag{2.17}$$

Với α_i là hệ số nhân Lagrange.

Hàm Lagrange đối với (2.16) chính là:

$$L = f(x) + \sum_{i=1}^{n} \alpha_i g_i(x) \tag{2.18}$$

Như vậy suy ra hàm Lagrange đối với (2.14) là:

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(< w.x_i > + b - 1)] \tag{2.19}$$

Với $\alpha_i \ge 0$ là hệ số nhân Lagrange.

Để tìm $\underline{L}(\alpha) = \inf_{w,b}(L(w,b,\alpha))$ ta tính đạo hàm L(w,b,α) theo w, b và đặt bằng 0.

$$\begin{cases} \dfrac{\partial L(w, b, \alpha)}{\partial \alpha} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \\ \dfrac{\partial L(w, b, \alpha)}{\partial \alpha} = \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} w = \sum_{i=1}^{n} \alpha_i y_i x_i \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \quad (2.20)$$

Thế vào L(w,b,α) ta có:

$$\underline{L}(\alpha) = \frac{1}{2}\left\|\sum_{i=1}^{n} \alpha_i y_i x_i\right\|^2 - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j < x_j x_i > + \sum_{i=1}^{n} \alpha_i$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i x_j >$$

Vậy bài toán đối ngẫu Lagrange là:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i x_j > \quad (2.21)$$

Với điều kiện

$$\begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i \geq 0, i = 1, \dots, n \end{cases}$$

Áp dụng điều kiện Karush-Kuhn-Tucker (KKT) một trong hai điều kiện KKT ở nghiệm tối ưu của bài toán tối ưu trên là:

$$\alpha_i [y_i (< w, x > -b) - 1] = 0$$

Có hai trường hợp xảy ra với α_i đó là:

- $\alpha_i = 0$, vì $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ nên mẫu học x_i sẽ không tham gia vào việc tính toán w và b.
- $\alpha_i > 0$ suy ra

  $y_i (< w, x > -b) = 1$

Nghĩa là x_i sẽ nằm trên một trong hai lề $w.x + b = +1$ hoặc $w.x + b = -1$. Khi đó xi được gọi là các vector hỗ trợ.

Ta tính được:

$$b = y_i - w.x_i \quad (2.22)$$

Trong thực tế, gọi SV là tập các vector hỗ trợ, N_{SV} = số vector hỗ trợ, khi đó b được tính bằng công thức:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (y_i - w.x_i) \quad (2.23)$$

Đối với một mẫu cần phân lớp chỉ cần tính giá trị:

$$\text{sign}(< w.x > + b) = \text{sign}(\sum_{x_i \in SV} \alpha_i.y_i < x_i.x > + b) \quad (2.24)$$

Nếu (2.24) trả về 1 thì mẫu được phân vào lớp có nhãn dương và ngược lại.

2.3.3. Thuật toán SVM tuyến tính với tập dữ liệu không phân tách được

Trường hợp SVM tuyến tính với tập dữ liệu phân tách được là một trường hợp lí tưởng. Với cách tìm lề lớn nhất như trên chỉ giải được khi dữ liệu phân tách được và cách tìm lề này gọi là lề cứng (hard margin). Trong thực tế, dữ liệu huấn luyện có thể bị nhiễu hoặc gán nhãn sai. Một số điểm thuộc lớp +1 nhưng lại nằm trong vùng của lớp -1. Trong trường hợp này cần phải mềm hóa các ràng buộc hay còn gọi là sử dụng C-SVM với lề mềm (soft margin). C-SVM có thể gán nhãn sai cho một số mẫu huấn luyện. Nếu không tìm được siêu phẳng nào phân tách được hai lớp dữ liệu thì C-SVM sẽ chọn một siêu phẳng phân tách các dữ liệu huấn luyện tốt nhất có thể đồng thời cực đại hóa khoảng cách giữa siêu phẳng với các dữ liệu được gán nhãn đúng.

Để giải quyết các trường hợp nêu trên cần nới lỏng các điều kiện bằng cách sử dụng $\xi_i \geq 0$ như sau:

$<w.x_i> + b \geq 1 - \xi_i$ nếu $y_i = +1$

$<w.x_i> + b \leq -1 + \xi_i$ nếu $y_i = -1$

Đối với một mẫu bị lỗi thì $\xi_i > 1$ và $\sum \xi_i$ sẽ là giới hạn trên của lỗi trong tập dữ liệu huấn luyện.

Như vậy, cần phải tích hợp lỗi trong hàm tối ưu mục tiêu bằng cách gán giá trị chi phí cho các lỗi vào hàm mục tiêu mới. Bài toàn tối ưu gốc chuyển thành như sau:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C(\sum_{i=1}^{n} \xi_i)^k$$

Với các ràng buộc

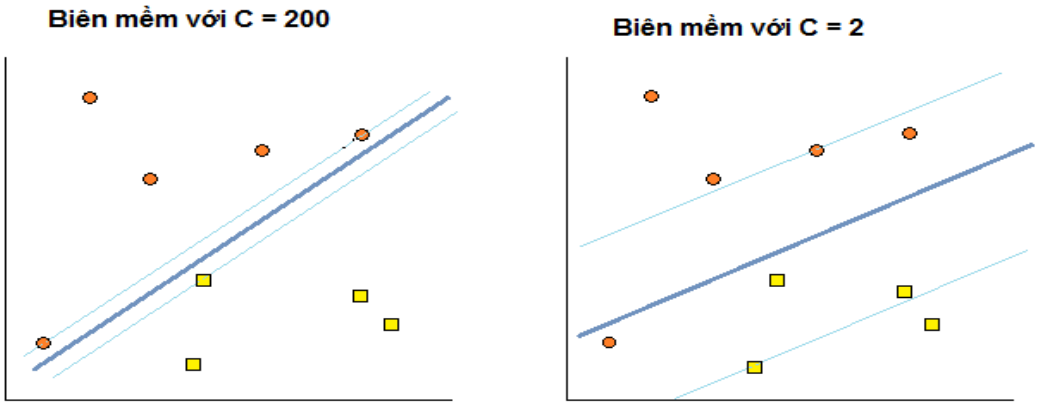
$$\begin{cases} y_i(< w.x_i > +b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

i=1,…,n (2.25)

Trong đó C > 0 là tham số xác định mức độ chi phí lỗi (penalty degree). C càng lớn thì mức độ chi phí đối với các lỗi càng cao. Nó ảnh hưởng đến độ cực đại biên và làm giảm số lượng các biến phụ ξ_i. Giá trị k=1 được sử dụng phổ biến để có biểu thức đối ngẫu đơn giản hơn.

Như vậy, khác với biên cứng, ngoài tìm cực tiểu hóa của $\|w\|^2$ còn phải thêm vào khoảng cách của các điểm lỗi đến vị trí đúng của nó.

Ảnh hưởng của C đến độ rộng biên và số lượng các biến phụ ξ_i sẽ được thấy rõ hơn trong hình 2.5 dưới đây [5].



Hình 2.5. Ảnh hưởng của C đến độ rộng biên

Trong hình 2.5 ta thấy với biên mềm C=200 độ rộng của biên là rất bé, chỉ có các điểm ngay sát siêu phẳng mới chịu ảnh hưởng lớn. Điều nay làm gia tăng xác suất phân lớp lỗi. Còn biên mềm với C=2 thì độ rộng biên lớn hơn, bỏ qua một số điểm ở gần lề khiến tăng số lượng các biến phụ ξ_i, hướng của siêu phẳng cũng thay đổi vì thế xác suất lỗi cũng giảm.

Bây giờ ta tìm bài toán đối ngẫu Lagrange:

$$L_P = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right) - \sum_{i=1}^{n}\alpha_i[y_i(\langle w.x_i\rangle + b) - 1 + \xi_i] - \sum_{i=1}^{n}\mu_i\xi_i \quad (2.26)$$

Lấy đạo hàm theo w, b, ξ ta có:

$$\begin{cases} \dfrac{\partial L_P}{\partial w} = w - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \\ \dfrac{\partial L_P}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0 \\ \dfrac{\partial L_P}{\partial \xi_i} = C\left(\sum_{i=1}^{n}\xi_i\right) - \alpha_i - \mu_i = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} w = \sum_{i=1}^{n}\alpha_i y_i x_i \\ \sum_{i=1}^{n}\alpha_i y_i = 0 \\ \alpha_i + \mu_i = C\left(\sum_{i=1}^{n}\xi_i\right) = \delta \end{cases} \quad (2.27)$$

Thay vào ta được:

$$\underline{L} = inf_{w,b,\xi} L_P = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i x_j\rangle \quad (2.28)$$

Vậy bài toán đối ngẫu Lagrange là:

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i x_j\rangle \quad (2.29)$$

Với các ràng buộc

$$\begin{cases} \sum_{i=1}^{n}\alpha_i y_i = 0 \\ 0 \le \alpha_i \le C,\ \forall i = 1, \dots, n \end{cases}$$

Để tính b chọn bất kì một i nào đó để $0 \le \alpha_i \le C$ suy ra:

$$b = \frac{1}{y_i} - \sum_{i=1}^{n} \alpha_i y_i \langle x_i . x \rangle \quad (2.30)$$

Siêu phẳng phân tách dữ liệu:

$$f(x) = \langle w . x \rangle + b = \sum_{i=1}^{n} \alpha_i y_i \langle x_i . x \rangle + b = 0 \quad (2.31)$$

Để phân lớp một ví dụ mới chỉ cần tính sign($\langle w . x \rangle + b$) như với lề cứng.

2.3.4. Thuật toán SVM phi tuyến phân lớp nhị phân

Trong thực tế các tập dữ liệu huấn luyện có ranh giới quyết định là không tuyến tính vì vậy rất khó giải quyết. Tuy nhiên có thể chuyển tập dữ liệu huấn luyện này về dạng tuyến tính quen thuộc bằng cách ánh xạ dữ liệu này sang một không gian lớn hơn gọi là không gian đặc trưng (feature space). Với không gian đặc trưng phù hợp thì dữ liệu huấn luyện sau khi ánh xạ sẽ trở tuyến tính và phân tách dữ liệu sẽ ít lỗi hơn so với không gian ban đầu. Phương pháp SVM phi tuyến có thể phân thành hai bước như sau:

Bước 1: Chuyển đổi không gian dữ liệu ban đầu sang một không gian đặc trưng khác (thường có số chiều lớn hơn), khi đó dữ liệu huấn luyện có thể phân tách tuyến tính được.

Bước 2: Áp dụng các công thức như với SVM tuyến tính.

Giả sử dữ liệu x_i ban đầu thuộc không gian R^n ta sử dụng một hàm ánh xạ ϕ để chuyển tập dữ liệu x_i sang không gian R^m.

$$\phi: R^n \to R^m$$

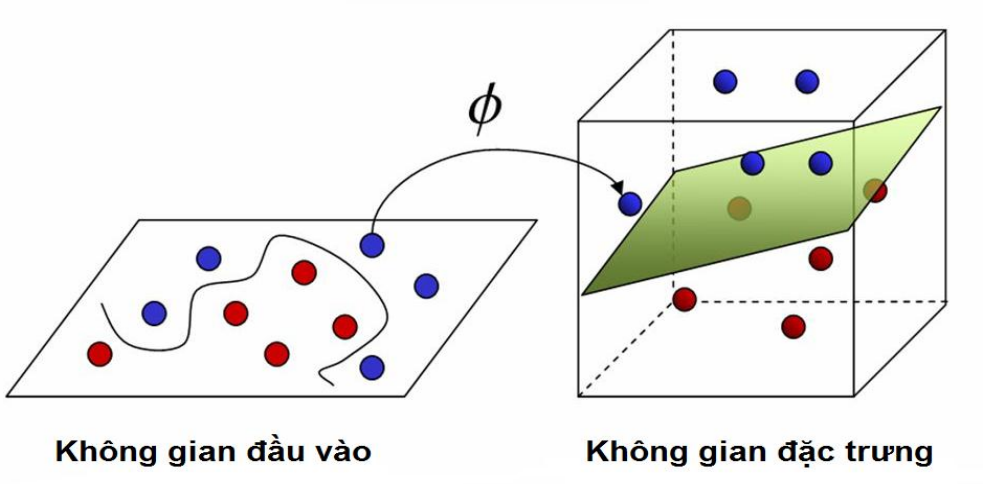
$$x \mapsto \phi(x)$$

Tập huấn luyện ban đầu

T = {(x_1, y_1), (x_2, y_2), …, (x_n, y_n)}

Được ánh xạ thành tập

T’ = {($\phi(x_1)$, y_1), ($\phi(x_2)$, y_2), …, ($\phi(x_n)$, y_n)}



Hình 2.6. Ánh xạ từ không gian 2 chiều sang không gian 3 chiều

Như ví dụ trong hình 2.6 [5], trong không gian 2 chiều không thể tìm được một siêu phẳng phân tách dữ liệu thành hai phần riêng biệt. Tuy nhiên, sau khi ánh xạ dữ liệu huấn luyện từ không gian R^2 sang không gian R^3 thì dễ dàng xác định được ngay siêu phẳng phân tách dữ liệu. Khi đó x_i trong không gian R^2 sẽ tương ứng với ϕ_i trong không gian R^3. Bài toán tối ưu trở thành:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

Với ràng buộc:

$$\begin{cases} y_i(< w.\phi(x_i) > +b) \geq 1 - \xi_i & \forall i = 1, \dots, n \\ \xi_i \geq 0 & \forall i = 1, \dots, n \end{cases}$$

Bài toán đối ngẫu Lagrange tương ứng là:

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \langle\phi(x_i)\phi(x_j)\rangle \qquad (2.32)$$

Với các ràng buộc

$$\begin{cases} \sum_{i=1}^{n}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \ \forall i = 1, \dots, n \end{cases}$$

Siêu phẳng phân tách dữ liệu:

$$f(\phi(x)) = \langle w.\phi(x)\rangle + b = \sum_{i=1}^{n} \alpha_i y_i \langle \phi(x_i)\phi(x_j)\rangle + b = 0.$$

Việc chuyển đổi không gian trực tiếp sẽ gặp vấn đề về chi phí thời gian nếu số chiều của không gian đầu vào là quá lớn, hoặc một số trường hợp với số chiều không gian đầu vào nhỏ nhưng vẫn tạo ra không gian đặc trưng có số chiều lớn. May mắn là ϕ(x) chỉ xuất hiện dưới dạng tích vô hướng ϕ(x).ϕ(z) mà không xuất hiện riêng rẽ. Chính vì vậy sử dụng hàm nhân có thể giải quyết được vấn đề này.

Hàm nhân có một số tính chất như sau:

- Một hàm nhân K được xác định khi tồn tại ϕ sao cho K(x,y) = ϕ(x).ϕ(y).
- Giả sử có m điểm mẫu, ta lập một ma trận $K_{i,j}$ = $K(x_i x_j)$ với i,j=1,…m.

Người ta chứng minh được rằng: Nếu K là hàm nhân thì ma trận $K_{i,j}$ sẽ là ma trận nửa xác định dương (có các giá trị riêng của ma trận ≥ 0).

- Nếu $K_1(x,y)$ và $K_2(x,y)$ là hàm nhân thì $K_3(x,y)$ cũng là hàm nhân

$$\begin{cases} K_3(x,y) = K_1(x,y) + K_2(x,y) \\ K_3(x,y) = \alpha K_1(x,y)\ \alpha \in R^+ \\ K_3(x,y) = K_1(x,y).K_2(x,y) \end{cases}$$

Việc lựa chọn hàm nhân phụ thuộc vào từng ứng dụng cụ thể. Đối với phân loại văn bản nên sử dụng các hàm nhân đa thức với số bậc thấp vì số chiều đặc trưng của chúng đã đủ lớn. Một số hàm nhân thường được sử dụng đó là:

- Hàm nhân đa thức: $K(x,y) = (x.y + c)^d$ với d là bậc đa thức. Chiều của không gian đặc trưng tương ứng với hàm nhân này là q = C_{n+d-1}^{d}. Hàm nhân K(x,y) này có thể chuyển tất cả các mặt cong bậc d trong không gian R^n thành một siêu phẳng trong không gian đặc trưng.
- Hàm bán kính cơ bản (Radial Basis Function) $K(x,y) = e^{\frac{-\|x-z\|^2}{2\sigma}}$ với σ>0.

Chiều của không gian đặc trưng với hàm nhân này là ∞ nên nó có thể chuyển mặt cong bất kì trong không gian R^n thành một siêu phẳng trong không gian đặc trưng.

- Hàm nhân tuyến tính (Linear Kernel) $K(x,y) = (x^T.y)^d$.

2.3.5. Thuật toán tối thiểu tuần tự SMO

Cả hai bài toán gốc và bài toán đối ngẫu của thuật toán SVM đều là bài toán tối ưu bậc 2 (Quadratic Programming) và đều có thể giải bằng phương pháp điểm trong (interior-point methods). Tuy nhiên, khi số lượng mẫu học N lớn thì ma trận K cũng lớn lên theo bậc 2 của N. Vì vậy phương pháp điểm trong cũng có thời gian chạy rất lâu cỡ N^3. Vì vậy, chúng ta phải lợi dụng cấu trúc của bài toán tối ưu trong thuật toán SVM để tăng tốc độ tối ưu hóa.

Thuật toán tối thiểu tuần tự (Sequential Minimal Optimization – SMO) là thuật toán tối ưu dành riêng cho phương pháp SVM do J. Platt đưa ra vào năm 1998. Ý tưởng chính của thuật toán này là:

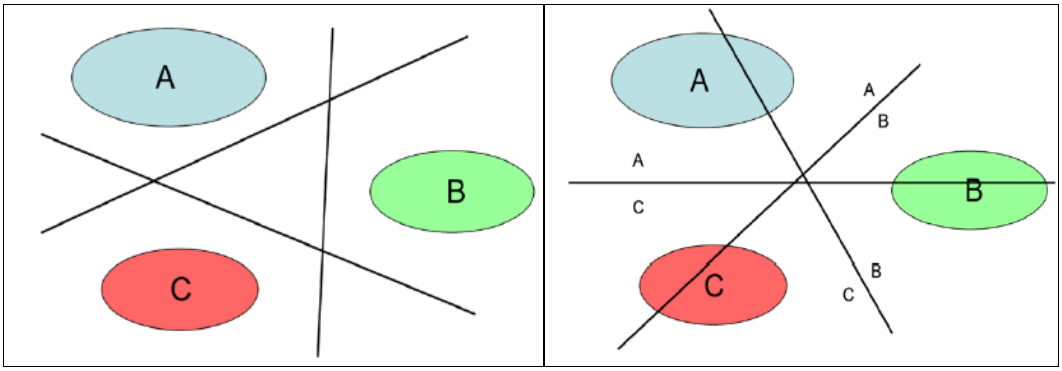
- Thay vì khống chế tất cả các ràng buộc, ta cố định phần lớn các biến λ_i và chỉ tối ưu hóa một cặp (λ_i, λ_j) nào đó.
- Giá trị tối ưu của cặp (λ_i, λ_j) có thể viết dưới dạng công thức (của dữ liệu và các biến λ_i khác) chứ không cần chạy một thuật toán tối ưu nào cả.
- Lần lượt chọn các cặp (λ_i, λ_j) theo một tiêu chí (heuristics) nào đó để thuật toán nhanh chóng hội tụ về nghiệm tối ưu.

Thuật toán tối thiểu tuần tự SMO được sử dụng trong hầu hết tất cả bài toán cài đặt thuật toán SVM.

2.3.6. Thuật toán SVM phân lớp đa lớp

Các kỹ trình bày trong các mục trên áp dụng cho phân lớp nhị phân. Trong mục này, luận văn sẽ khảo sát phương pháp SVM phân lớp đa lớp.Ý tưởng giải quyết bài toán phân lớp đa lớp là chuyển về thực hiện nhiều bài toán con phân lớp nhị phân. Khi đó các thuật toán nghiên cứu trong các mục trên sẽ được sử dụng trong cho mỗi bài toán con.

Xét bài toán phân lớp dữ liệu (1.2)-(1.3) với số lớp m > 2. Để giải quyết bài toán này sẽ tiến hành giải một số bài toán phân lớp nhị phân. Các chiến lược phân lớp đa lớp phổ biến này là One-against-All (OAA) và One-against-One (OAO) [3], [5].



(a): Chiến lược OAA (b): Chiến lược OAO

Hình 2.7. Phân lớp đa lớp sử dụng chiến lược OAA và OAO

Trong hình 2.7, chiến lược OAA và OAO phải xây dựng các siêu phẳng để phân tách từng lớp ra khỏi tất cả các lớp khác theo chiến lược khác nhau.

Chiến lược One-against-All (OAA – Chiến lược 1/m)

Chiến lược này sử dụng (m-1) bộ phân lớp nhị phân đối với m lớp. Bài toán phân lớp m lớp được chuyển thành m-1 bài toán phân lớp nhị phân. Trong đó, bộ phân lớp nhị phân thứ i được xây dựng trên qui ước mẫu thuộc lớp thứ i là mẫu dương (+1) và tất cả các mẫu thuộc các lớp còn lại là mẫu âm (-1). Hàm quyết định thứ i dùng để phân lớp thứ i và những lớp còn lại có dạng:

$$D_i(x) = w_i x + b_i.$$

Siêu phẳng $D_i(x) = 0$ tạo thành siêu phẳng phân chia tối ưu, các véc tơ hỗ trợ thuộc lớp i thoả $D_i(x) = 1$ và các véc tơ hỗ trợ thuộc các lớp còn lại thoả $D_i(x) = -1$. Nếu véc tơ dữ liệu x thoả mãn điều kiện $D_i(x) > 0$ đối với i duy nhất, x sẽ được phân vào lớp thứ i.

Tuy nhiên nếu điều kiện $D_i(x) > 0$ thoả mãn đối với nhiều i, hoặc không thoả đối với i nào thì trong trường hợp này không thể phân loại được véc tơ x. Để khắc phục nhược điểm này, chiến lược One-against-One (OAO) được đề xuất sử dụng.

Chiến lược One-against-One (OAO – Chiến lược 1/1)

Trong chiến lược OAO ta sử dụng m(m-1)/2 bộ phân lớp nhị phân được xây dựng để phân tách hai lớp (i, j), i = 1, 2, .., m-1, j = i+1, …, m. Trong đó, mẫu thuộc lớp i là mẫu dương (+1) và mẫu thuộc lớp j là mẫu âm (-1). Sau đó, sử dụng phương pháp lựa chọn theo đa số để kết hợp các bộ phân loại này để xác định được kết quả phân loại cuối cùng.

Hàm quyết định phân lớp của lớp i đối với lớp j trong chiến lược OAO là:

$$D_{ij}(x) = w_{ij}\, x + b_{ij}$$

$$D_{ij}(x) = -D_{ij}(x)$$

Đối với một vector x cần tính:

$$D_i(x) = \sum_{j \neq i, j=1}^{n} \text{sign}(D_{ij}(x))$$

$$\text{Với: } \text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Khi đó, x được phân vào lớp i sao cho: $D_i(x) = \underset{j=1,\ldots,n}{\text{argmax}} D_j(x)$.

Tuy nhiên nếu điều kiện $\underset{j=1,\ldots,n}{\text{argmax}} D_j(x)$ được thỏa mãn đối với nhiều i thì trong trường hợp này cũng không thể xác định được x thuộc lớp nào. Để giải quyết vấn đề này có thể sử dụng phân lớp đa lớp mờ. Trong phạm vi của luận văn chưa xét đến vấn đề này

2.3.7. Đánh giá phương pháp

Ưu điểm nổi bật của phương pháp SVM là thực hiện tối ưu toàn cục cho mô hình phân lớp. Do đó, mô hình SVM có chất lượng cao, chịu đựng được nhiễu. Mặt khác, SVM là một phương pháp tốt (phù hợp) đối với những bài toán phân lớp có

không gian biểu diễn thuộc tính lớn. Các đối tượng cần phân lớp được biểu diễn bởi một tập rất lớn các thuộc tính.

Tuy nhiên, phương pháp SVM cũng có một số nhược điểm:

- SVM chỉ làm việc với không gian đầu vào là các số thực. Đối với các thuộc tính định danh (nominal), cần chuyển các giá trị định danh thành các giá trị số.
- Độ phức tạp tính toán tương đối lớn.
- So với các phương pháp cây quyết định hoặc phương pháp Bayes, các kết quả dựa trên SVM khó hiểu hơn và khó giải thích.

Trong thực tế, phương pháp SVM được sử dụng trong nhiều bài toán phân lớp khác nhau như phân loại các văn bản, tài liệu Web, nhận dạng hình ảnh hay phân loại các chức năng các protein trong ứng dụng sinh học.

2.4. Kết luận chương 2

Chương 2 đã khảo sát tương đối chi tiết các kỹ thuật học máy: phương pháp cây quyết định, phương pháp Bayes và phương pháp SVM. Đây là các kỹ thuật học máy thường được ứng dụng giải quyết bài toán phân lớp dữ liệu.

Trong chương 3 tiếp theo, luận văn sẽ áp dụng thử nghiệm các phương pháp trên cho bài toán phân loại tấn công mạng trên bộ dữ liệu KDD cup 99.

CHƯƠNG 3. THỬ NGHIỆM VÀ ĐÁNH GIÁ

Trong chương 3 luận văn nghiên cứu thử nghiệm và đánh giá hiệu năng các kỹ thuật phân lớp dữ liệu dựa trên các phương pháp học máy đã nghiên cứu trong chương 2 trên bộ dữ liệu KDD cup 99.

3.1. Khảo sát và lựa chọn bộ dữ liệu để thử nghiệm

3.1.1. Giới thiệu chung

An ninh mạng là vấn đề an ninh phi truyền thống, còn khá mới mẻ nhưng ngày càng được thế giới và Việt Nam quan tâm cả cấp vĩ mô và vi mô.

Tại Việt Nam hiện có trên 55% dân số đang sử dụng điện thoại di động, trên 52% dân số sử dụng Internet [22]. Việt Nam đứng thứ 4 trên thế giới về thời gian sử dụng Internet và đứng thứ 22 trên thế giới tính theo dân số về số người sử dụng mạng xã hội. Hằng năm, Việt Nam phải chịu hàng ngàn cuộc tấn công mạng và Việt Nam đứng thứ 20 trên thế giới về xếp hạng các quốc gia bị tấn công mạng nhiều nhất, chịu thiệt hại lên tới 10.400 tỉ đồng riêng năm 2016 so với mức 8.700 tỉ đồng năm 2015 [17].

Trong năm 2017, Việt Nam đã hứng chịu rất nhiều các vụ tấn công mạng và để lại rất nhiều hậu quả nặng nề. Chỉ riêng quý 1 năm 2017, Việt Nam đã có gần 7700 sự cố tấn công mạng tại Việt Nam. Đến giữa tháng 9 số lượng các sự cố tấn công mạng đã lên đến gần 10000 [20] (số liệu của Trung tâm ứng cứu khẩn cấp máy tính Việt Nam – VNCERT). Trong đó có 1762 sự cố website lừa đảo, 4595 sự cố phát tán mã độc và 3607 sự cố tấn công thay đổi giao diện.

Theo báo cáo an ninh website của CyStack, chỉ trong quý 3 năm 2018 đã có 1.183 website của Việt Nam bị tin tặc tấn công và kiểm soát. Trong đó, các website giới thiệu sản phẩm và dịch vụ của doanh nghiệp là đối tượng bị tin tặc tấn công nhiều nhất (chiếm 71,51%). Vị trí thứ hai là các website thương mại điện tử (chiếm 13,86%).

Tháng 11/2018, Diễn đàn RaidForums đã đăng tải thông tin được cho là dữ liệu của hơn 5 triệu khách hàng của chuỗi bán lẻ thiết bị Thế giới di động. Những

thông tin bị rỏ rì bao gồm địa chỉ email, lịch sử giao dịch và thậm chí là cả số thẻ ngân hàng. Ngay sau đó, dữ liệu được cho là các hợp đồng trong chương trình F.Friends của FPT Shop cũng bị rò rỉ. Một số công ty Việt Nam như: Công ty cổ phần Con cưng, Ngân hàng hợp tác xã Việt Nam, ... cũng trở thành đích nhắm cho tin tặc.

Theo thống kê từ Trung tâm Giám sát an toàn không gian mạng quốc gia trực thuộc Cục An toàn thông tin (Bộ Thông tin và Truyền thông), có khoảng 4,7 triệu địa chỉ IP của Việt Nam thường xuyên nằm trong các mạng mã độc lớn (số liệu tháng 11/2018).

Trong quý I/2019, VNCERT ghi nhận có 4.770 sự cố tấn công mạng vào các trang web của Việt Nam. Cũng trong thời gian này hệ thống giám sát của VNCERT ghi nhận tổng cộng có hơn 78,3 triệu sự kiện mất an toàn thông tin tại Việt Nam.

Các thông tin và số liệu trên cho thấy một thực trạng đáng báo động về tấn công mạng tại Việt Nam hiện nay.

Như vậy, vấn đề phòng chống tấn công mạng đang là chủ đề nghiên cứu trở nên cấp thiết hơn trong bối cảnh bùng nổ cách mạng công nghệ truyền thông, Internet vạn vật và mạng xã hội gia tăng kết nối toàn cầu. Một trong những hướng nghiên cứu là xây dựng các hệ thống phòng chống tấn công mạng dựa trên các kỹ thuật học máy [16].

Từ những lý do trên, luận văn lựa chọn bộ dữ liệu về tấn công mạng KDD Cup 99 để thử nghiệm và đánh giá các mô hình phân lớp dữ liệu dựa trên các phương pháp học máy đã nghiên cứu trong chương 2.

3.1.2. Mô tả bộ dữ liệu KDD Cup 99

Dưới sự bảo trợ của Cơ quan Quản lý Nghiên cứu Dự Án Phòng Thủ Tiên tiến thuộc Bộ Quốc phòng Mỹ (DARPA) và phòng thí nghiệm nghiên cứu không quân (AFRL), năm 1998 phòng thí nghiệm MIT Lincoln đã thu thập và phân phối bộ dữ liệu được coi là bộ dữ liệu tiêu chuẩn cho việc đánh giá các nghiên cứu trong hệ thống phát hiện xâm nhập mạng máy tính. Dữ liệu được sử dụng trong cuộc thi KDD cup 99 là một phiên bản của bộ dữ liệu DARPA 98 [18].

Tập dữ liệu đầy đủ của bộ KDD cup 99 chứa 4.898.430 dòng dữ liệu, đây là một khối lượng dữ liệu lớn. Trong nghiên cứu và thử nghiệm, tập dữ liệu 10% của bộ KDD cup 99 thường được lựa chọn. Tập 10% của bộ KDD 99 tuy là tập con nhưng nó mang đầy đủ dữ liệu cho các loại hình tấn công khác nhau, đầy đủ thông tin quan trọng để thử nghiệm. Bảng 3.1 sau đây cho thấy số mẫu của các kiểu tấn công xuất hiện trong 10% bộ dữ liệu KDD cup 99 và nhãn lớp của chúng.

Bảng 3.1: Nhãn lớp và số mẫu xuất hiện trong 10% bộ dữ liệu KDD cup 99 [13]

Kiểu tấn công	Số mẫu ban đầu	Nhãn lớp
Back	2,203	DOS
land	21	DOS
Neptune	107,201	DOS
pod	264	DOS
smurf	280,790	DOS
teardrop	979	DOS
satan	1,589	PROBE
ipsweep	1,247	PROBE
nmap	231	PROBE
portsweep	1,040	PROBE
normal	97,277	NORMAL
Guess_passwd	53	R2L
ftp_write	8	R2L
imap	12	R2L
phf	4	R2L
multihop	7	R2L
warzemaster	20	R2L
warzclient	1,020	R2L
spy	2	R2L
Buffer_overflow	30	U2R
Loadmodule	9	U2R
perl	3	U2R
rootkit	10	U2R

Từ bảng trên, các kiểu tấn công khác nhau trong bộ dữ liệu được nhóm thành 5 loại (gán nhãn lớp) của bộ dữ liệu KDD cup'99 bao gồm:

1. Normal: dữ liệu thể hiện loại kết nối TCP/IP bình thường;

2. DoS (Denial of Service): dữ liệu thể hiện loại tấn công từ chối dịch vụ;

3. Probe: dữ liệu thể hiện loại tấn công thăm dò;

4. R2L (Remote to Local): dữ liệu thể hiện loại tấn công từ xa khi hacker cố gắng xâm nhập vào mạng hoặc các máy tính trong mạng;

5. U2R **(**User to Root): dữ liệu thể hiện loại tấn công chiếm quyền Root (quyền cao nhất) bằng việc leo thang đặc quyền từ quyền người dùng bình thường lên quyền Root.

Trong bộ dữ liệu KDD cup 99, với mỗi kết nối TCP/IP có 41 thuộc tính số và phi số được trích xuất. Đồng thời, mỗi kết nối được gán nhãn (thuộc tính 42) giúp phân biệt kết nối bình thường (Normal) và các tấn công. Các thuộc tính của bộ dữ liệu KDD cup 99 được mô tả chi tiết trong Bảng 3.2 dưới đây.

Bảng 3.2: Các thuộc tính của bộ dữ liệu KDD cup 99 [18]

TT	Tên thuộc tính	Mô tả	Tính chất	Ví dụ
1	Duration	Chiều dài (số giây) của kết nối.	Liên tục	0
2	Protocol_type	Loại giao thức, ví dụtcp, udp, vv..	Rời rạc	tcp
3	Service	Dịch vụ mạng trên các điểm đến ví dụ http,telnet, vv..	Rời rạc	http
4	Src_bytes	Số byte dữ liệu từ nguồn đến đích	Liên tục	SF
5	DTt_bytes	Số byte dữ liệu từ đích đến nguồn	Liên tục	181
6	Flag	Trạng thái bình thường hoặc lỗi của kết nối	Rời rạc	5450
7	Land	1 nếu kết nối là from/to cùng máy chủ/cổng; 0 nếu ngược lại	Rời rạc	0
8	Wrong_fragment	Số lượng đoạn "sai"	Liên tục	0
9	Urgent	Số gói tin khẩn cấp	Liên tục	0

TT	Tên thuộc tính	Mô tả	Tính chất	Ví dụ
10	Hot	Chỉ số “hot”	Liên tục	0
11	Num_failed_logins	Số lần đăng nhập không thành công	Liên tục	0
12	Logged_in	1 nếu đăng nhập thành công; 0 nếu ngược lại	Rời rạc	1
13	Num_compromised	Số lượng điều kiện thỏa hiệp	Liên tục	0
14	Root_shell	Bằng 1 nếu thu được root shell; 0 nếu ngược lại	Rời rạc	0
15	Su_attempted	Bằng 1nếu cố gắng thực hiện lệnh "su root"; 0 nếu ngược lại	Rời rạc	0
16	Num_root	Số lần truy cập quyền “root”	Liên tục	0
17	Num_file_creations	Số hoạt động tạo tập tin	Liên tục	0
18	Num_shells	Số lượng shell prompts	Liên tục	0
19	Num_access_files	Kiểm soát số lần truy cập file	Liên tục	0
20	Num_outbound_cmDT	Số lượng lệnh outbound trong 1 phiên ftp	Liên tục	0
21	Is_host_login	Bằng 1nếu đăng nhập thuộc về danh sách “máy chủ” đã biết, 0 nếu ngược lại	Rời rạc	0
22	Is_guest_login	Bằng 1 nếu đăng nhập là một tài khoản khách, 0 nếu ngược lại	Rời rạc	0
23	Count	Số lượng kết nối đến các máy chủ tương tự giống như các kết nối hiện hành trong 2 giây đã qua.	Liên tục	8
24	Serror_rate	Số % kết nối có lỗi “SYN”	Liên tục	8
25	Rerror_rate	Số % kết nối có lỗi“REJ”	Liên tục	0.00
26	Same_srv_rate	Số % các kết nối đến những dịch vụ tương tự	Liên tục	0.00
27	Diff_srv_rate	% kết nối với các dịch vụ khác nhau.	Liên tục	0.00
28	Srv_count	số kết nối đến cùng dịch vụ với kết nối hiện hành trong hai giây qua	Liên tục	0.00

TT	Tên thuộc tính	Mô tả	Tính chất	Ví dụ
29	Srv_serror_rate	% kết nối có lỗi "SYN" từ các dịch vụ	Liên tục	1.00
30	Srv_rerror_rate	% kết nối có lỗi "REJ" từ các dịch vụ.	Liên tục	0.00
31	Srv_diff_host_rate	Tỉ lệ % kết nối đến máy chủ khác nhau từ dịch vụ	Liên tục	0.00
32	DTt_host_count	Đếm các kết nối có cùng một đích đến.	Liên tục	9
33	DTt_host_srv_count	Đếm các kết nối có cùng 1host đích và sử dụng các dịch vụ tương tự.	Liên tục	9
34	DTt_host_same_srv_rate	% các kết nối có cùng 1host đích và sử dụng cácdịch vụ tương tự	Liên tục	1.00
35	DTt_host_diff_srv_rate	% các dịch vụ khác nhau trên các host hiện hành	Liên tục	0.00
36	DTt_host_same_src_port_rate	% các kết nối đến các host hiện thời có cùng cổng src	Liên tục	0.11
37	DTt_host_srv_diff_host_rate	% các kết nối đến các dịch vụ tương tự đến từ các host khác nhau	Liên tục	0.00
38	DTt_host_serror_rate	% các kết nối đến các host hiện thời có một lỗi SO	Liên tục	0.00
39	DTt_host_srv_serror_rate	% các kết nối đến các host hiện hành và dịch vụ quy định rằng có một lỗi SO	Liên tục	0.00
40	DTt_host_rerror_rate	% các kết nối đến các host hiện thời có một lỗi RST	Liên tục	0.00
41	DTt_host_srv_rerror_rate	% các kết nối đến các máy chủ hiện hành và dịch vụ quy định rằng có một lỗi RST	Liên tục	0.00
42	Nhãn	Kết nối bình thường/tấn công	Tượng trưng	Normal

Ví dụ về một vài dòng dữ liệu trong bộ KDD cup 99:

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.

0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,smurf.

Một số chuyên gia phát hiện xâm nhập mạng cho rằng, hầu hết các loại tấn công mới là các biến thể của các loại tấn công đã biết và dấu hiệu của các loại tấn công đã biết có thể đủ để nắm bắt được các biến thể mới lạ.

3.2. Xây dựng kịch bản và lựa chọn công cụ thử nghiệm

3.2.1. Xây dựng kịch bản thử nghiệm

Bài toán đặt ra là phân loại kiểu tấn công trong bộ dữ liệu KDD cup 99 nhằm hỗ trợ cho các hệ thống phát hiện xâm nhập mạng. Đây là bài toán được nhiều tác giả quan tâm nghiên cứu trong thời gian gần đây. Có thể tham khảo các kết quả nghiên cứu chi tiết trong các tài liệu [1], [2], [6], [8], [9], [11] và [16].

Trong mục này, luận văn sẽ thực hiện thử nghiệm với bài toán sau:

Đầu vào của bài toán:

(1) Bộ dữ liệu KDD cup 99;

(2) Các thuật toán thử nghiệm:

- Thuật toán Cây quyết định (Decision Tree);
- Thuật toán Bayes;
- Thuật toán máy vecto hỗ trợ (SMV).

Đầu ra của bài toán:

Các độ đo đánh giá hiệu năng các mô hình phân loại kiểu tấn công sử dụng các thuật toán thử nghiệm trên bộ dữ liệu KDD cup 99.

Luận văn sẽ tiến hành thử nghiệm theo hai kịch bản trình bày dưới đây.

Kịch bản thứ nhất:

Trong kịch bản này, luận văn sẽ thực hiện phân lớp dữ liệu trong KDD cup 99 thành 2 lớp: kết nối bình thường (Normal) và kết nối tấn công (anomaly - không bình thường). Lý do là trong các hệ thống phát hiện xâm nhập mạng, trước hết cần quan tâm đến các kết nối tấn công để hệ thống xem xét cảnh báo và đề xuất các giải pháp xử lý phù hợp.

Kịch bản thứ hai:

Trong kịch bản thứ hai, luận văn sẽ thực hiện phân lớp dữ liệu trong KDD cup 99 thành các lớp như trình bày trong mục 3.1.2: Normal (dữ liệu thể hiện loại

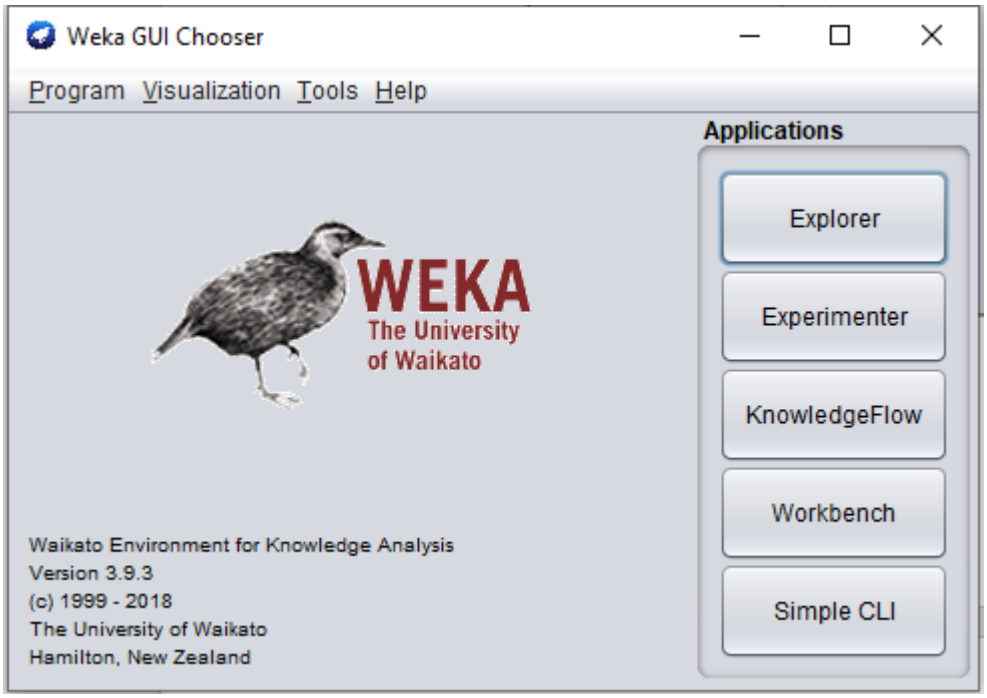
kết nối TCP/IP bình thường); DoS (dữ liệu thuộc loại tấn công từ chối dịch vụ); Probe (dữ liệu thuộc loại tấn công thăm dò); R2L (dữ liệu thuộc loại tấn công từ xa) và U2R (dữ liệu thuộc loại tấn công chiếm quyền Root).

Kết quả phân loại chi tiết các kiểu tấn công sẽ hỗ trợ hệ thống phát hiện xâm nhập mạng đề xuất các giải pháp xử lý phù hợp nhất có thể.

3.2.2. Lựa chọn công cụ thử nghiệm

Weka là một phần mềm miễn phí về học máy được viết bằng Java, phát triển bởi University of Wekato. Weka có thể coi như là bộ sưu tập các thuật toán về học máy dùng trong phân tích và khai phá dữ liệu. Các thuật toán đã được xây dựng sẵn và người dùng chỉ việc lựa chọn để sử dụng. Do đó Weka rất thích hợp cho việc thử nghiệm các mô hình mà không mất thời gian để xây dựng chúng. Weka có giao diện sử dụng đồ hoạ trực quan và cả chế độ command line. Ngoài các thuật toán về học máy như dự đoán, phân loại, phân cụm, Weka còn có các công cụ để trực quan hoá dữ liệu rất hữu ích trong quá trình nghiên cứu, phân tích dữ liệu lớn.

Từ những lý do trên, luận văn lựa chọn công cụ thực nghiệm là phần mềm Weka version 3.9. [21].



Hình 3.1. Giao diện khởi động của WEKA

Các tính năng chính của Weka:

- Weka bao gồm một tập các công cụ tiền xử lý dữ liệu, các thuật toán học máy để khai phá dữ liệu và các phương pháp thử nghiệm đánh giá.

- Weka có giao diện đồ hoạ (gồm cả tính năng hiển thị hoá dữ liệu)

- Weka bao gồm các môi trường cho phép so sánh các thuật toán học máy trên bộ dữ liệu do người dùng lựa chọn.

Các môi trường chính trong Weka:

(1) Simple CLI: giao diện đơn giản kiểu dòng lệnh (như MS-DOS).

(2) Explorer: môi trường cho phép sử dụng tất cả các khả năng của Weka để khám phá dữ liệu.

(3) Experimenter: môi trường cho phép tiến hành các thí nghiệm và thực hiện các kiểm tra thống kê (statistical tests) giữa các mô hình máy học. Môi trường này bao gồm:

- **Preprocess:** Để chọn và thay đổi (xử lý) dữ liệu làm việc.
- **Classify:** Để huấn luyện và kiểm tra các mô hình học máy (phân loại, hoặc hồi quy/dự đoán).
- **Cluster:** Để học các nhóm từ dữ liệu (phân cụm).
- **Associate:** Để khám phá các luật kết hợp từ dữ liệu.
- **Select attributes:** Để xác định và lựa chọn các thuộc tính liên quan (quan trọng) nhất của dữ liệu.
- **Visualize:** Để xem (hiển thị) biểu đồ tương tác 2 chiều đối với dữ liệu.

(4) KnowledgerFlow: môi trường cho phép bạn tương tác đồ hoạ kiểu kéo/ thả để thiết kế các bước (các thành phần) của một thí nghiệm.

Để tiến hành thử nghiệm, cần lựa chọn "Explorer": giao diện cho phép sử dụng tất cả các chức năng cơ sở của Weka bằng cách lựa chọn menu.

Để đánh giá hiệu năng các bộ phân loại cần lựa chọn các tuỳ chọn cho việc kiểm tra trong (test options) bao gồm:

- *Use training set*: Bộ phân loại học được sẽ được đánh giá trên tập học.

- *Supplied test set*: Sử dụng một tập dữ liệu khác (với tập huấn luyện) để cho việc đánh giá.

- *Cross-validation*: Tập dữ liệu sẽ được chia đều thành k tập (folds) có kích thước xấp xỉ nhau, và bộ phân loại học được sẽ được đánh giá bởi phương pháp cross-validation.

- *Percentage split*. Chỉ định tỷ lệ phân chia tập dữ liệu.

3.3. Triển khai thử nghiệm và đánh giá kết quả

3.3.1. Mô tả thử nghiệm

Máy tính sử dụng cho quá trình chạy Weka để đánh giá hiệu năng các thuật toán là laptop có cấu hình:

- Bộ xử lý Intel R-core i5 2418M,

- RAM: 4GB.

Bộ công cụ weka phiên bản 3.9. 3.

Dữ liệu huấn luyện các mô hình là tập KDDTrain+_20Percent.arff chứa 25192 bản ghi, số thuộc tính là 42 (bao gồm cả nhãn).

Dữ liệu kiểm chứng là tập KDDTest-21.arff. chứa 11850 bản ghi.

Các thuật toán lựa chọn thử nghiệm:

- Phương pháp Cây quyết định sử dụng j48.

- Phương pháp Bayes sử dụng Naïve-Bayes và Bayes-Net.

- Phương pháp SVM sử dụng SMO.

Thực hiện thử nghiệm theo hai kịch bản nêu ở mục 3.2.1.

Các bước thực hiện:

Bước 1. Huấn luyện các mô hình phân lớp và đánh giá *Cross-validation* với k=10.

Bước 2. Kiểm chứng mô hình với tuỳ chọn đánh giá là *Supplied test set.*

3.3.2. Kết quả thử nghiệm

Trong mục này luận văn trình bày một số kết quả chính trích ra từ các bản log khi chạy trên Weka. Do giới hạn về số trang của luận văn nên không thể nêu chi tiết các thao tác trên Weka.

(1) Kết quả giai đoạn huấn luyện của các mô hình theo kịch bản 1

Kết quả thử nghiệm đối với j48 trình bày trong bảng 3.3.

Bảng 3.3: Kết quả thử nghiệm 2 lớp của thuật toán j48

```
=== Detailed Accuracy By Class ===
TP Rate FP Rate  Precision  Recall   F-Measure     Class
0.996   0.004    0.996      0.996     0.996        normal
0.996   0.004    0.995      0.996     0.995        anomaly
0.996   0.004    0.996      0.996     0.996        (Avg.)

=== Confusion Matrix ===
  a        b   <-- classified as
 13389    60 |     a = normal
    51 11692 |     b = anomaly
```

Kết quả thử nghiệm đối với Naïve-Bayes trình bày trong bảng 3.4.

Bảng 3.4: Kết quả thử nghiệm 2 lớp của thuật toán Naïve-Bayes

```
=== Detailed Accuracy By Class ===
TP Rate FP Rate  Precision  Recall   F-Measure      Class
0,912   0,123    0,895      0,912     0,903         normal
0,877   0,088    0,897      0,877     0,887         anomaly
0,896   0,106    0,896      0,896     0,896        (Avg.)


=== Confusion Matrix ===
     a     b   <-- classified as
 12272  1177 |     a = normal
  1445 10298 |     b = anomaly
```

Kết quả thử nghiệm đối với Bayes-Net trình bày trong bảng 3.5.

Bảng 3.5: Kết quả thử nghiệm 2 lớp của thuật toán Net-Bayes

```
=== Detailed Accuracy By Class ===
TP Rate FP Rate  Precision  Recall   F-Measure     Class
0,991    0,064    0,947      0,991    0,969        normal
0,936    0,009    0,989      0,936    0,962        anomaly
0,966    0,038    0,967      0,966    0,966         (Avg.)
=== Confusion Matrix ===
  a       b   <-- classified as
 13330   119 |     a = normal
   747 10996 |     b = anomaly
```

Kết quả thử nghiệm đối với SMO trình bày trong bảng 3.6.

Bảng 3.6: Kết quả thử nghiệm 2 lớp của thuật toán SMO

```
=== Detailed Accuracy By Class ===
TP Rate FP Rate  Precision  Recall   F-Measure      Class
0.986   0.041    0.965      0.986    0.975          normal
0.959   0.014    0.984      0.959    0.971          anomaly
0.973   0.029    0.974      0.973    0.973          (Avg.)
=== Confusion Matrix ===
     a     b   <-- classified as
 13261   188 |     a = normal
   485 11258 |     b = anomaly
```

Kết quả các độ đo của các thuật toán thử nghiệm trong bước huấn luyện theo kịch bản 1 được tổng hợp trong bảng 3.7.

Bảng 3.7: Tổng hợp kết quả huấn luyện 2 lớp của các thuật toán thử nghiệm

Thuật toán	accuracy (%)	Normal			Anomaly		
		Pre	Rec	F1	Pre	Rec	F1
J48	99.55	99.6	99.6	99.6	99.5	99.6	99.5
NaiveBayes	89.59	89.5	91.2	90.3	89.7	87.7	88.7
BayesNet	96.56	94.7	99.1	96.9	98.9	93.6	96.2
SMO	97.32	96.5	98.6	97.5	98.4	95.9	97.1

(2) Kết quả giai đoạn kiểm chứng của các mô hình theo kịch bản 1

Kết quả kiểm chứng các mô hình được tổng hợp trong bảng 3.8.

Bảng 3.8: Tổng hợp kết quả kiểm chứng 2 lớp của các thuật toán thử nghiệm

Thuật toán	accuracy (%)	Normal			Anomaly		
		Pre	Rec	F1	Pre	Rec	F1
J48	63.97	32	87.3	46.8	95.4	58.8	72.8
NaiveBayes	55.77	24.3	67.8	35.8	88.2	53.1	66.3
BayesNet	51.68	25.7	87.8	39.8	94.2	43.7	59.7
SMO	52.7	22.7	66.9	33.9	87.1	49.5	63.2

(3) Kết quả giai đoạn huấn luyện thử nghiệm theo kịch bản 2

Tương tự như đối với kịch bản 1, kết quả thử nghiệm đối với các thuật toán j48, NaiveBaye, BayesNet và SMO theo kịch bản 2 được tổng hợp trong bảng 3.9.

Bảng 3.9: Tổng hợp kết quả huấn luyện đa lớp của các thuật toán thử nghiệm

Các lớp	Các độ đo	Thuật toán			
		J48	NaiveBayes	BayesNet	SMO
Normal	Prec	99.40	95.80	97.80	97.60
	Rec	99.70	77.90	95.10	98.80
	F1	99.50	85.90	96.40	98.20
DoS	Prec	99.80	96.50	99.80	99.30
	Rec	99.90	95.00	93.60	98.00
	F1	99.80	95.80	96.60	98.70
U2R	Prec	42.90	0.60	3.40	66.70
	Rec	27.30	72.70	63.60	18.20
	F1	33.30	1.10	6.50	28.60
R2L	Prec	99.10	22.20	47.80	77.50
	Rec	82.80	52.20	94.30	64.10
	F1	86.70	31.10	63.40	70.20
Probe	Prec	99.10	61.40	79.20	96.80

Các lớp	Các độ đo	Thuật toán			
		J48	NaiveBayes	BayesNet	SMO
	Rec	98.30	88.00	98.10	96.40
	F1	98.70	72.40	87.70	96.60
accuracy (%)		99.44	84.86	94.79	97.98

(4) Kết quả giai đoạn kiểm chứng thử nghiệm theo kịch bản 2

Tương tự như đối với kịch bản 1, kết quả thử nghiệm đối với các thuật toán j48, NaiveBaye, BayesNet và SMO theo kịch bản 2 được tổng hợp trong bảng 3.10.

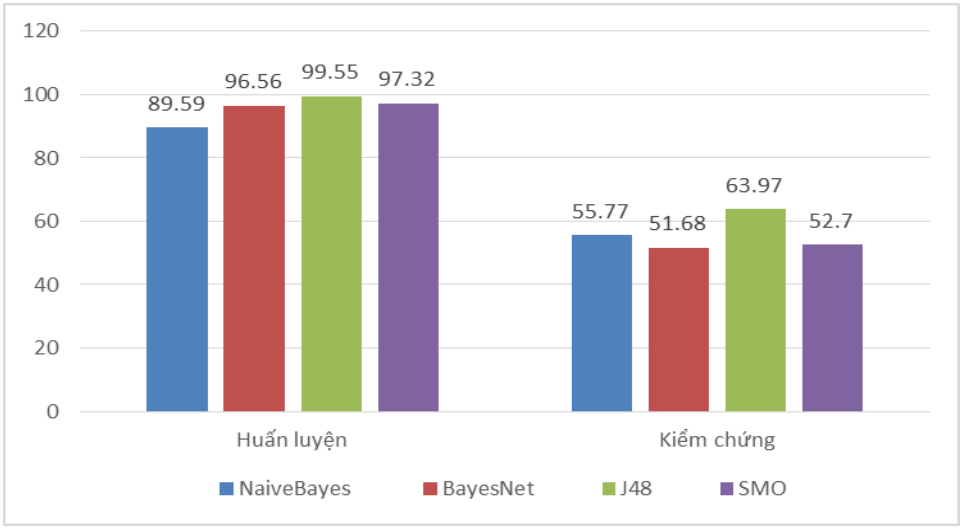
Bảng 3.10: Tổng hợp kết quả kiểm chứng đa lớp của các thuật toán thử nghiệm

Các lớp	Các độ đo	Thuật toán			
		J48	NaiveBayes	BayesNet	SMO
Normal	Prec	58.00	40.30	53.20	41.50
	Rec	87.00	54.70	84.10	69.10
	F1	69.60	46.40	65.20	51.80
DoS	Prec	97.30	79.40	98.60	95.70
	Rec	96.80	69.10	61.30	86.50
	F1	97.00	73.90	75.60	90.90
U2R	Prec	76.50	2.40	9.70	83.30
	Rec	35.10	32.40	62.20	13.50
	F1	48.10	4.50	16.80	23.30
R2L	Prec	16.70	22.20	81.70	22.20
	Rec	0.10	1.00	19.90	0.20
	F1	0.20	1.80	32.00	0.30
Probe	Prec	83.80	71.50	68.70	51.60
	Rec	99.70	92.00	99.80	56.20
	F1	81.10	80.50	81.40	53.80
accuracy (%)		77.04	56.16	66.70	61.12

3.3.3. Đánh giá kết quả thử nghiệm

Dựa vào kết quả thử nghiệm đã trình bày ở trên, trong mục này luận văn sẽ thực hiện phân tích và đánh giá kết quả.

Kết quả độ chính xác của các thuật toán thử nghiệm theo kịch bản 1 trên tập huấn luyện và tập kiểm chứng được biểu diễn dưới dạng biểu đồ như trong hình 3.2.



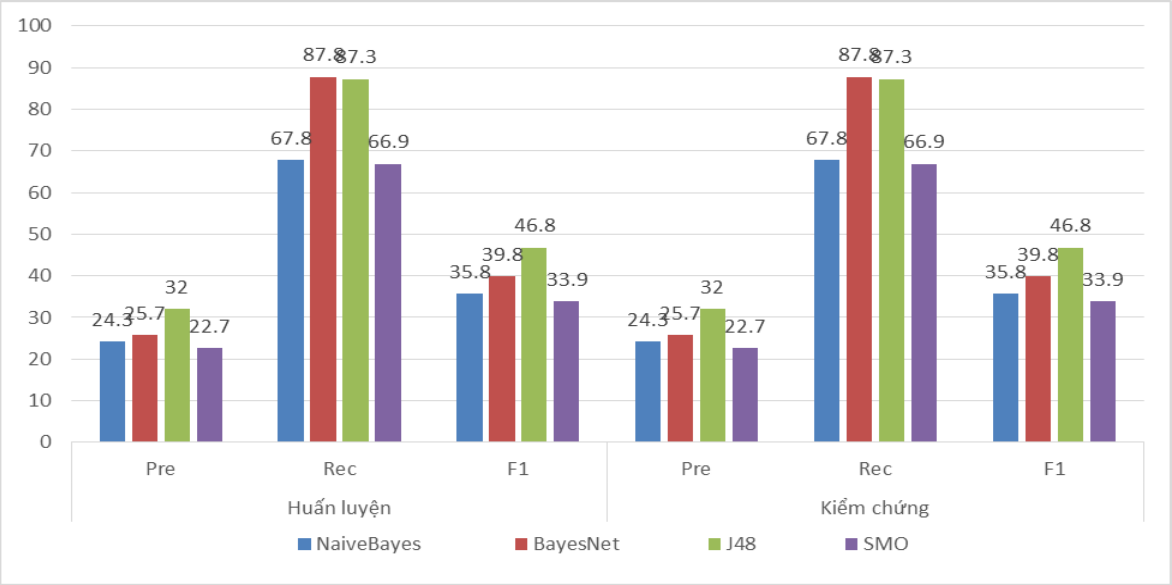
Hình 3.2 Biểu đồ so sánh độ chính xác của các thuật toán thử nghiệm 2 lớp

Quan sát biểu đồ trên hình 3.2 nhận thấy rằng, các thuật toán thử nghiệm đều cho kết quả có tỉ lệ phân loại chính xác cao trên tập huấn luyện (từ 90% trở lên).

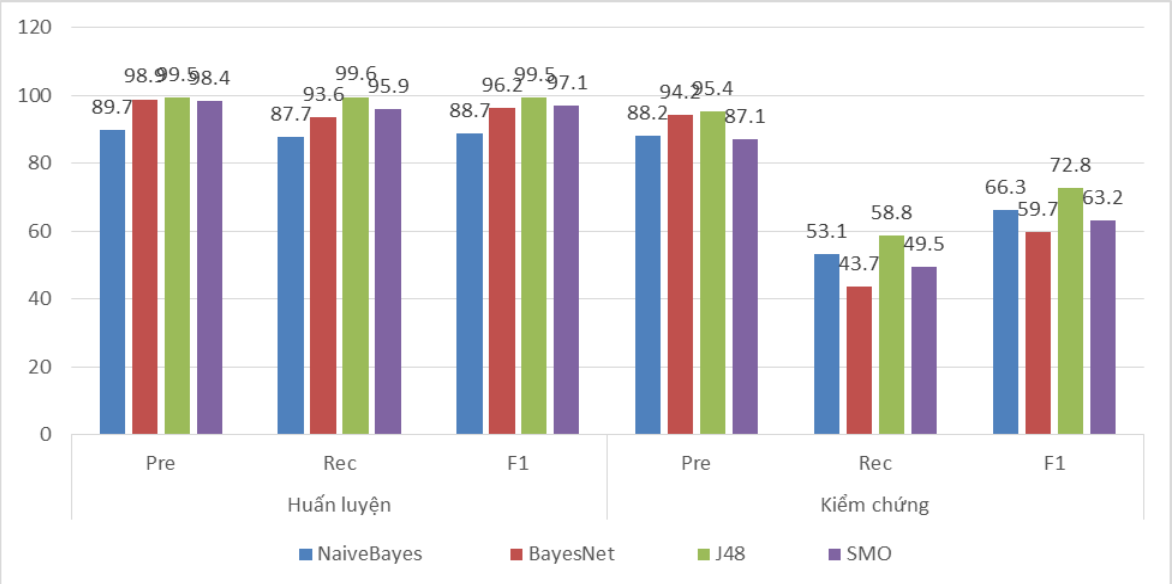
Trong đó, mô hình cây quyết định (j48) có tỉ lệ phân loại chính xác cao nhất (99.55%) và mô hình Naïve Bayes tỉ lệ phân loại chính xác thấp nhất (89.59%).

Tuy nhiên, khi thực hiện kiểm thử tỷ lệ phân loại chính xác bị sụt giảm rõ rệt chỉ còn trên 51%. Trong đó, mô hình cây quyết định (j48) có tỉ lệ phân loại chính xác cao nhất (63.97%) và mô hình Bayes Net tỉ lệ phân loại chính xác thấp nhất (51.68%). Lý do của sự sụt giảm là tập huấn luyện còn nhỏ, cần phải có kích thước lớn hơn để đảm bảo kết quả khi kiểm chứng.

Có thể so sánh kết quả phân loại thử nghiệm theo lớp bình thường (Normal) và lớp tấn công (Anomaly**)** theo các biểu đồ trên hình 3.3 và 3.4.

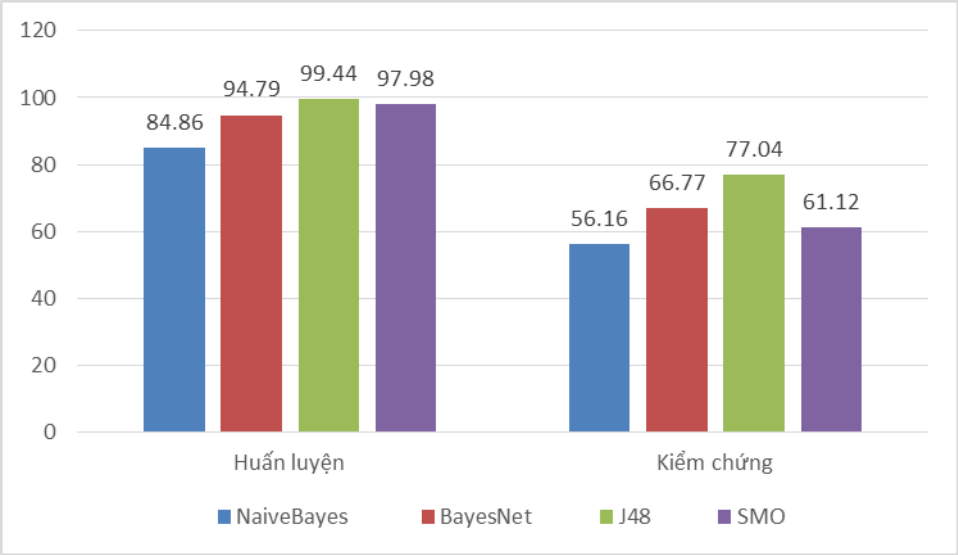


Hình 3.3 Biểu đồ so sánh độ chính xác của lớp Normal trong thử nghiệm 2 lớp



Hình 3.4 Biểu đồ so sánh độ chính xác của lớp Anomal trong thử nghiệm 2 lớp

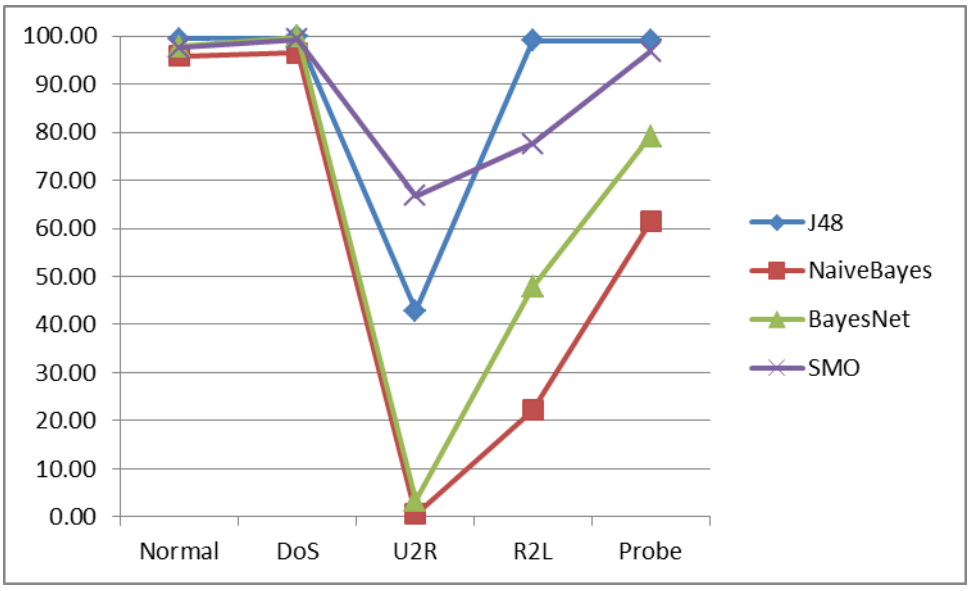
Kết quả độ chính xác của các thuật toán thử nghiệm theo kịch bản 2 trên tập huấn luyện và tập kiểm chứng được biểu diễn dưới dạng biểu đồ như trong hình 3.5.



Hình 3.5 Biểu đồ so sánh độ chính xác của mô hình trong thử nghiệm đa lớp

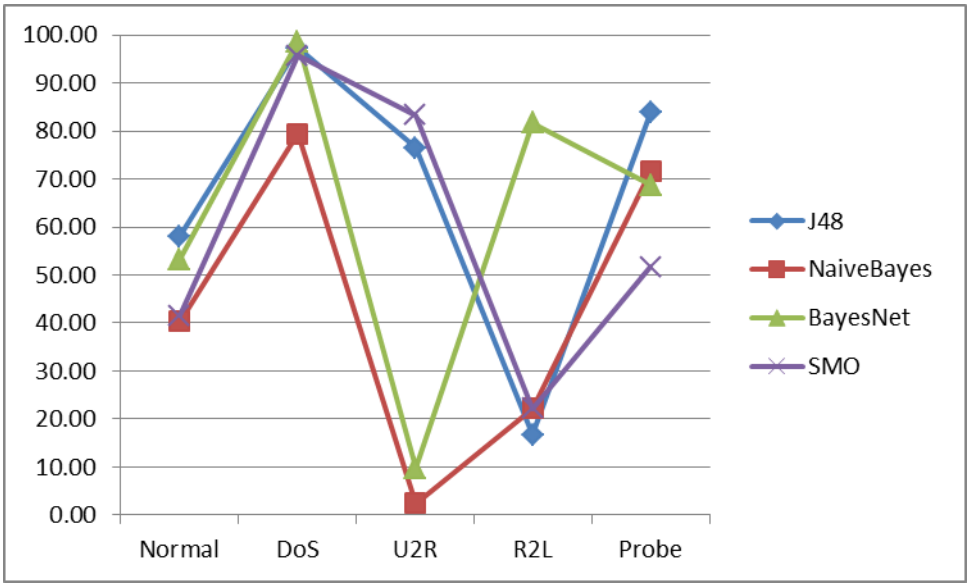
Quan sát trên hình 3.2 và 3.5 nhận thấy kết quả các mô hình khi thực hiện phân lớp đa lớp khi kiểm chứng cho kết quả độ chính xác cao hơn khi chỉ thực hiện phân lớp 2 lớp. Điều này có thể được lý giải là các mô hình khi thực hiện phân lớp đa lớp sẽ phù hợp hơn.

Hình 3.6 trình bày biểu đồ thống kê mức chính xác (Precision) theo từng lớp của các mô hình thử nghiệm đa lớp trên tập huấn luyện.



Hình 3.6 Mức chính xác theo lớp trong thử nghiệm đa lớp trên tập huấn luyện

Hình 3.7 trình bày biểu đồ thống kê mức chính xác (Precision) theo từng lớp của các mô hình thử nghiệm phân lớp đa lớp trên tập kiểm chứng.



Hình 3.7 Mức chính xác theo lớp trong thử nghiệm đa lớp trên tập kiểm chứng

Tóm lại, trong cả hai kịch bản thử nghiệm, mô hình cây quyết định và mô hình SVM có độ chính khá tốt. Điều này cũng phù hợp với thực tế là hai mô hình này thường được sử dụng để xây dựng các bộ phân lớp.

3.4. Kết luận chương 3

Trong chương 3 luận văn đã tiến hành thử nghiệm các thuật toán học máy nghiên cứu trong chương 2 cho bài toán phân loại tấn công mạng với bộ dữ liệu KDD cup 99.

Kết quả thử nghiệm bước đầu cho thấy các thuật toán học máy có thể triển khai trong thực tế và phù hợp với các yêu cầu đề ra cho bài toán phân lớp dữ liệu.

KẾT LUẬN

Các kết quả đạt được của luận văn:

Với mục tiêu nghiên cứu các thuật toán học máy cho bài toán phân lớp dữ liệu và thử nghiệm, luận văn đã đạt được một số kết quả sau đây:

- Nghiên cứu tổng quan về bài toán phân lớp dữ liệu và các vấn đề liên quan.
- Khảo sát tổng quan về học máy nhằm bài toán phân lớp dữ liệu.
- Giới thiệu chung về học sâu.
- Khảo sát chi tiết các phương pháp học máy: Cây quyết định, Bayes và SVM.
- Khảo sát bộ dữ liệu tấn công mạng KDD cup 99.
- Thực hiện thử nghiệm các thuật toán học máy j48, Naïve Bayes, Bayes Net và SMO để phân loại các kiểu tấn công mạng đối với bộ dữ liệu NSL-KDD.

Tuy nhiên, do hạn chế về mặt thời gian, luận văn chưa tiến hành thử nghiệm với các bộ dữ liệu lớn, Do đó, hiệu quả thử nghiệm chưa cao.

Hướng phát triển tiếp theo:

- Thực hiện xây dựng và triển khai hệ thống phân lớp dữ liệu sử dụng thuật toán học máy cho các bài toán thực tế.
- Nghiên cứu các kỹ thuật học sâu cho bài toán phân lớp dữ liệu.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Hoàng Ngọc Thanh, Trần Văn Lăng, Hoàng Tùng (2016) – "Một tiếp cận máy học để phân lớp các kiểu tấn công trong hệ thống phát hiện xâm nhập mạng", Kỷ yếu Hội nghị khoa học Quốc gia FAIR'9, T. 502-507.

Tiếng Anh

[2] I. Ahrmad, A.B. Abdullah, A.S. Alghamdi (2009) – "Application of Artificial Neural Netword in Detection ò Probing Attacks", IEEE, ISEA 2009.

[3] E. L. Allwein, R. E. Schapire, and Y. Singer (2001) – "*Reducing multiclass to binary: A unifying approach for margin classifiers*" - The Journal of Machine Learning Research, V.1, pp. 113–141.

[4] Tapan Bagchi, Rahul Samant, Milan Joshi (2013) – "*SVM Classifiers Built Using Imperfect Training Data*" - International Conference on Mathematical Techniques In Engineering Applications, ICMTEA 2013-BM-003.

[5] Christopher J.C. Burges (2000) – "*A Tutorial on Support Vector Machines for Pattern Recognition*" – Kluwer Academic Publishers, Boston.

[6] Debasish Das, Utpal Sharma, D.K. Bhattacharyya (2010) - "An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching", International Journal of Computer Applications, Volume 1, No. 25, pp. 28 – 33.

[7] Han J., Kamber M. (2011) – "*Data mining: Concepts and Techniques*" - 3nd Edition, Morgan Kaufman Publishers.

[8] M.M. Javidi, M.H. Nattaj (2013)-"A New and Quick Method to Detect DoS Attacks by Neural Networks", Journal of Mathematics and Computer Sciense, Vol.6, pp. 85-96.

[9] A. Joshi, V. Geetha (2014) - "SQLi detection using machine learning,", Control, Instrumentation, Communication and Computational Technologies, pp. 1111–1115.

[10] R. Komiya, I. Paik, M. Hisada (2011) - "Classification of malicious web code by machine learning", Awareness Science and Technology (iCAST), pp. 406–411.

[11] Lee I., Jeong S., Yeo S. and Moon J. (2012) – "A novel method for SQL injection attack detection based on removing SQL query attribute values", Mathematical and Computer Modelling, 55(1), pp.58-68.

[12] T. M. Mitchell [1997] – "Machine Learning", McGraw-Hill.

[13] Olusola A.A., Oladele A.S. and Abosede D.O. (2010) –"Analysis of KDD'99 Intrusion Detection Dataset for Selection of Relevance Features" – WCECS, Vol 1.

[14] Siddiqui M.K. and Naahid S. (2013) – "Analysis of KDD CUP 99 Dataset using Clustering based Data Mining" - International Journal of Database Theory and Application, V. 6, No 5, pp. 23-34.

[15] O'Sullivan, Dympna, et al. (2008) - "Using Secondary Knowledge to Support Decision Tree Classification of Retrospective Clinical Data" - Mining Complex Data (2008), pp. 238-251.

[16] A.S. Unal, M. Hacibeyoglu (2018)-" Detection of DDoS Attacks in Network Traffic Using Deep Learning", ICATCES 18, pp. 722-726.

Trang Web

[17] http://bkav.com.vn/

[18] http://kdd.ics.uci.edu/databases/kddcup99

[19] https://vi.wikipedia.org.

[20] http://vncert.gov.vn

[21] http://www.cs.waikato.ac.nz/ml/weka/

[22] https://www.vnnic.vn/