TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

Báo cáo môn học

KHAI PHÁ DỮ LIỆU

Giảng viên hướng dẫn: **TS. Lê Chí Ngọc**

Lớp: **Cao học toán tin**

HÀ NỘI, 10/2020

Mục lục

Danh sách hình vẽ

Phân công công việc

- Chương 1: Vũ Minh Linh, Phùng Trọng Hiếu
- Chương 2: Nguyễn Ngọc Đàm, Ngô Thị Trà
- Chương 3: Trần Thị Thu Hương, Nguyễn Minh Châu
- Chương 4: Nguyễn Tuấn Anh, Nguyễn Minh Quân, Nguyễn Đức Sơn
- Chương 5: Lê Thị Duyên, Nguyễn Chí Thảo

Chương 1

Đại cương về phân tích dữ liệu

1.1 Phân tích dữ liệu là gì?

Thế giới sở hữu lượng dữ liệu vô tận sẵn có để làm việc. Các công ty lớn như Google và Microsoft sử dụng dữ liệu để đưa ra quyết định, nhưng đây không phải là những nơi duy nhất thực hiện việc này. Phân tích dữ liệu (Data Analysis) còn được sử dụng bởi các doanh nghiệp nhỏ, các công ty bán lẻ, trong y học và thậm chí cả thế giới thể thao. Nó là một ngôn ngữ phổ quát và quan trọng hơn bao giờ hết.

Phân thích dữ liệu

Phân tích dữ liệu là môn khoa học phân tích dữ liệu thô để đưa ra được kết luận về thông tin đó. Nhiều kĩ thuật và qui trình phân tích dữ liệu đã được tự động hóa thành các qui trình cơ học và thuật toán để xử lí dữ liệu thô về hoạt động tiêu dùng của con người.

Phân tích dữ liệu có thể tìm ra các xu hướng và số liệu trong các khối thông tin mà có thể bị bỏ sót nếu không sử dụng kĩ thuật này. Thông tin thu được có thể được sử dụng để tối ưu hóa các qui trình làm tăng hiệu quả tổng thể của một doanh nghiệp hoặc một hệ thống.

Bất kì loại thông tin nào cũng có thể được áp dụng các kĩ thuật phân tích dữ liệu để rút ra những hiểu biết giúp cải thiện vấn đề. Ví dụ, các công ty sản xuất thường ghi lại thời gian chạy, thời gian chết và thời gian chờ đợi công việc của các máy để phân tích dữ liệu để lên kế hoạch tốt hơn cho khối lượng công việc để máy có thể hoạt động gần với công suất tối ưu.

Phân tích dữ liệu có nhiều tác dụng ngoài việc chỉ ra các nút thắt và vấn đề trong sản xuất. Ví dụ, các công ty game sử dụng phân tích dữ liệu để đặt lịch thưởng cho người chơi để giữ người chơi luôn gắn vào game.

Các dạng phân tích dữ liệu

Phân tích dữ liệu được chia thành 4 loại cơ bản:

- Phân tích mô tả: Miêu tả những gì đã xảy ra trong một khoảng thời gian nhất định. Số lượt xem trang web đã tăng lên chưa? Doanh số tháng này có lớn hơn tháng trước không?

- Phân tích chẩn đoán: Tập trung nhiều hơn vào lí do tại sao một hiện tượng nào đó xảy ra. Điều này yêu cầu dữ liệu đầu vào đa dạng hơn và cần một vài giả thuyết. Thời tiết có tác động đến doanh số bán bia không? Chiến dịch marketing mới nhất có ảnh hưởng đến doanh số không?

- Phân tích dự đoán: Cho biết những gì có thể sẽ xảy ra trong thời gian tới. Trong mùa hè nóng lần trước doanh số của ta là bao nhiêu? Có bao nhiêu mô hình thời tiết dự đoán mùa hè năm nay sẽ nóng?

- Phân tích đề xuất: Đề xuất những hành động nên thực hiện. Ví dụ, nếu xác suất rằng mùa hè năm nay là nóng được đo theo các mô hình thời tiết mà công ty sử dụng là trên 58%, công ty nên tăng thêm ca tối tại nhà máy bia và thuê thêm một bể bổ sung để tăng sản lượng.

Các bước phân tích dữ liệu

Phân tích dữ liệu là một chủ đề lớn và có thể bao gồm một số bước sau:

- Xác định mục tiêu: Bắt đầu bằng cách phác thảo một số mục tiêu được xác định rõ ràng. Để có được kết quả tốt nhất từ dữ liệu, các mục tiêu phải rõ ràng.

- Đặt câu hỏi: Tìm ra các câu hỏi muốn trả lời bằng dữ liệu. Ví dụ, những chiếc xe thể thao màu đỏ có gặp tai nạn thường xuyên hơn những phương tiện khác không? Chỉ ra công cụ phân tích dữ liệu nào sẽ nhận được kết quả tốt nhất cho câu hỏi.

- Thu thập dữ liệu: Thu thập dữ liệu hữu ích để trả lời các câu hỏi. Trong ví dụ này, dữ liệu có thể được thu thập từ nhiều nguồn khác nhau như DMV hoặc báo cáo tai nạn của cảnh sát, yêu cầu bảo hiểm và chi tiết nhập viện.

- Data Scrubbing ("làm sạch" dữ liệu): Dữ liệu thô có thể được thu thập ở một số định dạng khác nhau, với nhiều thứ không có giá trị và lộn xộn. Dữ liệu cần được "làm sạch" và chuyển đổi để các công cụ phân tích dữ liệu có thể nhập nó. Bước này rất quan trọng.

- Phân tích dữ liệu: Nhập dữ liệu "sạch" mới này vào các công cụ phân tích dữ liệu. Các công cụ này cho phép khám phá dữ liệu, tìm mẫu và trả lời câu hỏi what-if (điều gì xảy ra, nếu ...).

- Rút ra kết luận và đưa ra dự đoán: Hãy rút ra kết luận từ dữ liệu. Những kết luận này có thể được tóm tắt trong một báo cáo, biểu đồ trực quan hoặc cả hai để có được kết quả đúng.

Những đối tượng sử dụng phân tích dữ liệu

Một số lĩnh vực sử dụng phân tích dữ liệu bao gồm ngành du lịch và khách sạn, các ngành này có lượng khách quay vòng nhanh chóng. Hai ngành này có thể thu thập dữ liệu khách hàng và tìm ra liệu có vấn đề phát sinh không, nếu có thì chúng phát sinh ở đâu và cách khắc phục chúng.

Chăm sóc sức khỏe sử dụng kết hợp khối lượng lớn dữ liệu có cấu trúc và dữ liệu không cấu trúc rồi sử dụng phân tích dữ liệu để đưa ra quyết định nhanh chóng.

Tương tự, ngành bán lẻ sử dụng lượng dữ liệu lớn để đáp ứng nhu cầu luôn thay đổi của người mua hàng. Các thông tin mà nhà bán lẻ thu thập và phân tích có thể giúp họ xác định xu hướng, đề xuất sản phẩm và tăng lợi nhuận.

1.2 Các quá trình khám phá tri thức

1.2.1 Chuẩn bị dữ liệu

Dữ liệu dạng cơ sở dữ liệu

Một hệ thống cơ sở dữ liệu, còn được gọi là **hệ thống quản lý cơ sở dữ liệu (DBMS)**, bao gồm một tập hợp dữ liệu liên quan, được gọi là **cơ sở dữ liệu** và một tập hợp các chương trình phần mềm để quản lý và truy cập dữ liệu. Các chương trình phần mềm cung cấp cơ chế xác định cấu trúc cơ sở dữ liệu và lưu trữ dữ liệu; để xác định và quản lý truy cập dữ liệu đồng thời, chia sẻ hoặc phân tán; và để đảm bảo tính nhất quán và bảo mật của thông tin được lưu trữ bất chấp sự cố hệ thống hoặc nỗ lực truy cập trái phép.

Một **cơ sở dữ liệu quan hệ** là một tập hợp các bảng, mỗi trong số đó được gán một tên duy nhất. Mỗi bảng bao gồm một tập hợp các thuộc tính (cột hoặc trường) và thường lưu trữ một tập hợp lớn các tuples (hồ sơ hoặc hàng). Mỗi tuple trong một bảng quan hệ đại diện cho một đối tượng được xác định bởi một khóa duy nhất và được mô tả bởi một tập hợp các giá trị thuộc tính. Mô hình dữ liệu ngữ nghĩa, chẳng hạn như **mô hình dữ liệu quan hệ thực thể (ER)**, thường được xây dựng cho cơ sở dữ liệu quan hệ. Mô hình dữ liệu ER đại diện cho cơ sở dữ liệu dưới dạng tập hợp các thực thể và mối quan hệ của chúng.

Ví dụ Cơ sở dữ liệu quan hệ **AllElectronics**:

- *customer (cust_ID, name, address, age, occupation, annual_income, credit_information, category, ...)*
- *item (item_ID, brand, category, type, price, place_made, supplier, cost, ...)*
- *employee (empl_ID, name, category, group, salary, commission, ...)*
- *branch (branch_ID, name, address, ...)*
- *purchases (trans_ID, cust_ID, empl_ID, date, time, method_paid, amount)*
- *items_sold (trans_ID, item_ID, qty)*
- *works_at (empl_ID, branch_ID)*

Dữ liệu quan hệ có thể được truy cập bởi các truy vấn cơ sở dữ liệu được viết bằng ngôn ngữ truy vấn dữ liệu (ví dụ: SQL) hoặc với sự hỗ trợ của giao diện người dùng đồ họa. Một truy vấn nhất định được chuyển thành một tập hợp các phép toán quan hệ, chẳng hạn như phép kết nối, phép chọn và phép chiếu, và sau đó được tối ưu hóa để xử lý hiệu quả. Truy vấn cho phép truy xuất các tập con được chỉ định của dữ liệu. Giả sử rằng công việc của bạn là phân tích dữ liệu *AllElectronics*. Thông qua việc sử dụng các truy vấn quan hệ, bạn có thể hỏi những thứ như, *"Hiển thị cho tôi một danh sách tất cả các mặt hàng đã được bán trong quý cuối cùng."* Ngôn ngữ truy vấn dữ liệu cũng sử dụng các hàm tổng hợp như sum (tổng), avg (trung bình), count (số lượng), max (tối đa) và min (tối thiểu).

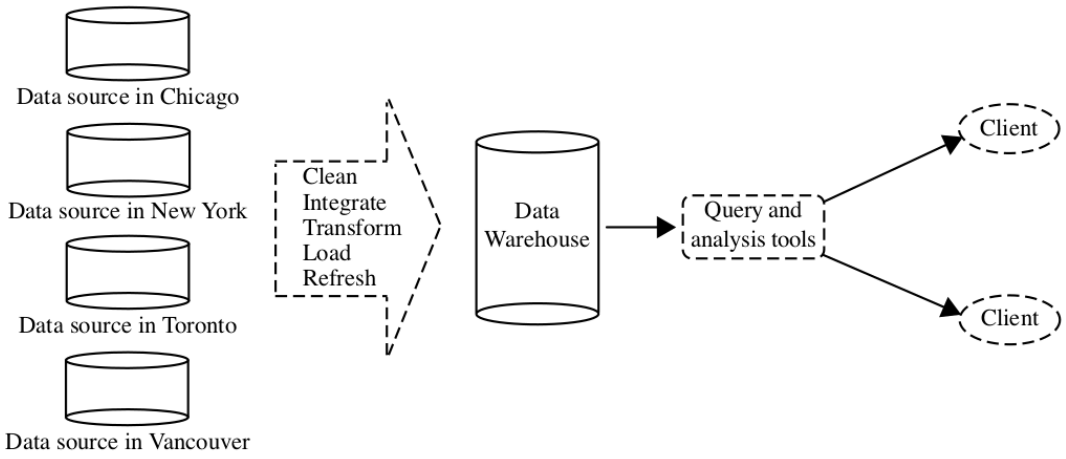
Khi khai phá cơ sở dữ liệu quan hệ, ta có thể đi xa hơn bằng cách tìm kiếm xu hướng hoặc mô hình dữ liệu. Ví dụ: các hệ thống khai phá dữ liệu có thể phân tích dữ liệu khách hàng để dự đoán rủi ro tín dụng của khách hàng mới dựa trên thu nhập, tuổi tác và thông tin tín dụng trước đó của họ. Hệ thống khai phá dữ liệu cũng có thể phát hiện sai lệch, có nghĩa là, doanh số dự kiến của mặt hàng khác xa so với năm trước. Những sai lệch như vậy sau đó có thể được điều tra thêm. Ví dụ: khai phá dữ liệu có thể phát hiện ra rằng đã có sự thay đổi trong bao bì của một mặt hàng hoặc tăng giá đáng kể.

Cơ sở dữ liệu quan hệ là một trong những kho thông tin phổ biến nhất và phong phú nhất, và do đó chúng là một dạng dữ liệu chính trong nghiên cứu khai phá dữ liệu.

Kho dữ liệu

Kho dữ liệu là một kho lưu trữ thông tin được thu thập từ nhiều nguồn, được lưu trữ theo sơ đồ thống nhất và thường nằm ở một trang web duy nhất. Kho dữ liệu được xây dựng thông qua quy trình làm sạch dữ liệu, tích hợp dữ liệu, chuyển đổi dữ liệu, tải dữ liệu và làm mới dữ liệu định kỳ.

Để tạo thuận lợi cho việc ra quyết định, dữ liệu trong kho dữ liệu được tổ chức xung quanh các chủ đề chính (ví dụ: khách hàng, mặt hàng, nhà cung cấp và hoạt động). Dữ liệu được lưu trữ để cung cấp thông tin từ góc độ lịch sử, chẳng hạn như trong 6 đến 12 tháng qua và thường được tóm tắt. Ví dụ: thay vì lưu trữ chi tiết của từng giao dịch bán hàng, kho dữ liệu có thể lưu trữ bản tóm tắt các giao dịch trên mỗi loại mặt hàng cho mỗi cửa hàng hoặc, được tóm tắt ở cấp độ cao hơn, cho từng khu vực bán hàng.



Hình 1.1: Kho dữ liệu.

Kho dữ liệu thường được mô hình hóa bởi cấu trúc dữ liệu đa chiều, được gọi là khối dữ liệu, trong đó mỗi thứ nguyên tương ứng với một thuộc tính hoặc một tập hợp các thuộc tính trong lược đồ và mỗi ô lưu trữ giá trị của một số biện pháp tổng hợp như số lượng hoặc tổng (số tiền bán hàng). Một khối dữ liệu cung cấp một cái nhìn đa hướng của dữ liệu và cho phép tính toán trước và truy cập nhanh dữ liệu tóm tắt.

Dữ liệu giao dịch

Nói chung, mỗi bản ghi trong **dữ liệu giao dịch** nắm bắt một giao dịch, chẳng hạn như mua hàng của khách hàng, đặt vé máy bay hoặc nhấp chuột của người dùng trên trang web. Một giao dịch thường bao gồm một số nhận dạng giao dịch duy nhất (*trans_ID*) và một danh sách các mục tạo nên giao dịch, chẳng hạn như các mặt hàng được mua trong giao dịch. Dữ liệu giao dịch có thể có bảng bổ sung, chứa các thông tin khác liên quan đến các giao dịch, chẳng hạn như mô tả mục, thông tin về nhân viên bán hàng hoặc chi nhánh, v.v.

Khi phân tích của *AllElectronics*, ta thể đặt ra câu hỏi, "Những mặt hàng nào được bán cùng nhau?" Loại phân tích dữ liệu giỏ hàng này sẽ cho phép kết hợp các nhóm mặt hàng lại với nhau như một chiến lược để thúc đẩy doanh số bán hàng. Ví dụ, với kiến thức rằng máy in thường được mua cùng với máy tính, ta có thể cung cấp một số máy in nhất định với mức giảm giá mạnh (hoặc thậm chí miễn phí) cho khách hàng mua máy tính được chọn, với hy vọng bán được nhiều máy tính hơn (thường đắt hơn máy in). Một hệ thống cơ sở dữ liệu truyền thống không thể thực hiện phân tích dữ liệu giỏ hàng. May mắn thay, khai phá dữ liệu trên dữ liệu giao dịch có thể làm như vậy bằng cách khai thác các tập mặt hàng thường xuyên, các tập mặt hàng thường được bán cùng nhau.

1.2.2 Tiền xử lý dữ liệu

Trong qui trình khai phá dữ liệu, công việc xử lý dữ liệu trước khi đưa vào các mô hình là rất cần thiết, bước này làm cho dữ liệu có được ban đầu qua thu thập dữ liệu (gọi là dữ liệu gốc original data) có thể áp dụng được (thích hợp) với các mô hình khai phá dữ liệu (data mining model) cụ thể. Quá trình tiền xử lý dữ liệu có thể coi là quá trình xử lý dữ liệu gốc nhằm cải thiện chất lượng dữ liệu do đó cải thiện chất lượng khai phá dữ liệu. Dữ liệu gốc gồm các dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc, được đưa vào từ các nguồn dữ liệu trong hệ thống xử lý tập tin hoặc các hệ thống cơ sở dữ liệu.

Chất lượng của dữ liệu được đánh giá trên các tiêu chí sau:

- *Accuracy* (tính chính xác): Các giá trị được ghi nhận đúng với giá trị thật.
- *Currency/timelines* (tính hiện hành): Các giá trị được ghi nhận không bị lỗi thời.
- *Completeness* (tính toàn vẹn): Tất cả các giá trị dành cho một biến/thuộc tính đều được ghi nhận.
- *Consistency* (tính nhất quán): Tất cả giá trị dữ liệu đều được biểu diễn như nhau trong tất cả các trường hợp.

Các công việc cụ thể của tiền xử lý dữ liệu bao gồm những công việc như:

- *Filtering Attributes*: Chọn các thuộc tính phù hợp với mô hình.
- *Filtering samples*: Lọc các mẫu (instances, patterns) dữ liệu cho mô hình.
- *Clean data*: Làm sạch dữ liệu như xóa bỏ các dữ liệu bất thường (outlier), dữ liệu nhiễu (noise), hiệu chỉnh những phần dữ liệu không nhất quán.
- *Transformation*: Chuyển đổi dữ liệu cho phù hợp với các mô hình như chuyển đổi dữ liệu từ numeric qua nomial hay ordinal, làm trơn dữ liệu (smoothing), kết hợp dữ liệu (aggregation), tổng quá hóa dữ liệu (generalization), chuẩn hóa dữ liệu (normalization) ...
- *Discretization* (rời rạc hóa dữ liệu): Nếu bạn có dữ liệu liên tục nhưng một vài mô hình chỉ áp dụng cho các dữ liệu rời rạc (như luật kết hợp chẳn hạn) thì bạn phải thực hiện việc rời rạc hóa dữ liệu.

1.2.3 Khai phá dữ liệu

Khai phá dữ liệu là một chủ đề liên quan tới nhiều ngành nên có thể được định nghĩa theo nhiều cách khác nhau. Để đề cập đến việc khai thác vàng từ đá hoặc cát, ta nói khai thác vàng thay vì khai thác đá hoặc cát. Tương tự như vậy, khai phá dữ liệu nên có được đặt tên đầy đủ hơn là "khai phá kiến thức từ dữ liệu". Tuy nhiên, ngắn hạn, khai phá kiến thức có thể không phản ánh sự nhấn mạnh vào khai phá từ một lượng lớn dữ liệu. Tuy nhiên, khai thác mỏ là một thuật ngữ sinh động đặc trưng cho quá trình tìm thấy một bộ nhỏ cốm quý từ rất nhiều nguyên liệu thô. Do đó, một

sự nhầm lẫn như vậy mang cả "dữ liệu" và "khai thác mở" đã trở thành một lựa chọn phổ biến. Ngoài ra, nhiều thuật ngữ khác có ý nghĩa tương tự như khai phá dữ liệu - ví dụ, khai phá kiến thức từ dữ liệu, khai phá kiến thức, phân tích dữ liệu/mô hình.

Nhiều người coi khai phá dữ liệu như một từ đồng nghĩa với một thuật ngữ được sử dụng phổ biến khác, khám phá kiến thức từ dữ liệu, trong khi những người khác xem khai phá dữ liệu chỉ là một bước thiết yếu trong quá trình khám phá kiến thức. Quá trình khám phá kiến thức được thể hiện như là một chuỗi lặp đi lặp lại của các bước sau:

1. Làm sạch dữ liệu (để loại bỏ nhiễu và dữ liệu không nhất quán).
2. Tích hợp dữ liệu (nơi có thể kết hợp nhiều nguồn dữ liệu).
3. Lựa chọn dữ liệu (nơi dữ liệu liên quan đến nhiệm vụ phân tích được truy xuất từ cơ sở dữ liệu).
4. Chuyển đổi dữ liệu (nơi dữ liệu được chuyển đổi và hợp nhất thành các biểu mẫu thích hợp để khai thác bằng cách thực hiện các hoạt động tóm tắt hoặc tổng hợp).
5. Khai phá dữ liệu (một quá trình thiết yếu trong đó các phương pháp thông minh được áp dụng để trích xuất các mẫu dữ liệu)
6. Đánh giá mẫu (để xác định các mô hình thực sự tốt đại diện cho kiến thức dựa trên các biện pháp tốt).
7. Trình bày kiến thức (nơi trực quan hóa và kỹ thuật đại diện kiến thức được sử dụng để trình bày kiến thức khai thác cho người dùng).

Các bước từ 1 đến 4 là các hình thức tiền xử lý dữ liệu khác nhau, nơi dữ liệu được chuẩn bị để khai phá. Bước khai phá dữ liệu có thể tương tác với người dùng hoặc cơ sở kiến thức. Các mẫu tốt được trình bày cho người dùng và có thể được lưu trữ dưới dạng kiến thức mới trong cơ sở kiến thức.

Từ những ý trên cho thấy khai phá dữ liệu là một bước trong quá trình khám phá kiến thức, mặc dù là một bước cần thiết vì nó phát hiện ra các mẫu ẩn để đánh giá. Tuy nhiên, trong ngành công nghiệp, trong phương tiện truyền thông, và trong các tài liệu nghiên cứu, khai phá dữ liệu hạn thường được sử dụng để chỉ toàn bộ quá trình khám phá kiến thức (có lẽ vì thuật ngữ này ngắn hơn so với khám phá kiến thức từ dữ liệu). Do đó, ta áp dụng một cái nhìn rộng về chức năng khai phá dữ liệu: **Khai phá dữ liệu** là quá trình khám phá các mẫu và kiến thức tốt từ một lượng lớn dữ liệu. Các nguồn dữ liệu có thể bao gồm cơ sở dữ liệu, kho dữ liệu, Web, kho thông tin khác hoặc dữ liệu được truyền vào hệ thống một cách tự động.

1.2.4 Đánh giá kết quả

Đánh giá mẫu dữ liệu, hay còn gọi là đánh giá thông tin thu được từ quá trình khai thác, xác định mức độ chính xác, khả năng đem lại giá trị thực sự, để trả lời cho câu hỏi "mẫu dữ liệu/thông tin thu được có cần quan tâm để phát triển chiến lược,

triển khai vào thực tế không?" Ví dụ cụ thể của bước này là xây dựng các giả thuyết và tiến hành kiểm định, dựa trên mức độ tin cậy và kết quả kiểm định để xem xét. Nếu kết quả phân tích không hợp lý, có sai sót, cần quay kiểm tra lại các bước ở trên, còn nếu kết quả phân tích đã hợp lý, độ tin cậy cao thì tiếp tục bước sau cùng dưới đây.

Không phải tất cả các mô hình được tạo ra bởi các quy trình khai thác dữ liệu đều tốt. Điều gì làm cho một mô hình tốt có thể khác nhau từ người dùng đến người dùng. Do đó, kỹ thuật là cần thiết để đánh giá sự tốt của các mô hình phát hiện dựa trên các biện pháp chủ quan. Những ước tính giá trị của các mẫu đối với một lớp người dùng nhất định, dựa trên niềm tin hoặc kỳ vọng của người dùng. Hơn nữa, bằng cách sử dụng các biện pháp tốt hoặc hạn chế do người dùng chỉ định để hướng dẫn quá trình khám phá, ta có thể tạo ra các mẫu tốt hơn và giảm không gian tìm kiếm.

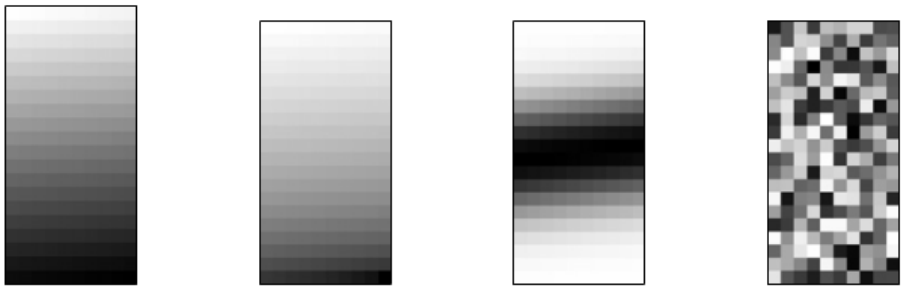
1.2.5 Hiển thị kết quả

Trực quan hóa dữ liệu nhằm mục đích truyền đạt dữ liệu rõ ràng và hiệu quả thông qua đại diện đồ họa. Trực quan hóa dữ liệu đã được sử dụng rộng rãi trong nhiều ứng dụng ví dụ, tại nơi làm việc để báo cáo, quản lý hoạt động kinh doanh và theo dõi tiến độ của nhiệm vụ. Phổ biến hơn, ta có thể tận dụng các kỹ thuật trực quan để khám phá các mối quan hệ dữ liệu mà nếu không dễ quan sát bằng cách nhìn vào dữ liệu thô. Ngày nay, mọi người cũng sử dụng trực quan hóa dữ liệu để tạo ra đồ họa thú vị.

Kỹ thuật hình ảnh hóa theo hướng pixel

Một cách đơn giản để trực quan hóa giá trị của thứ nguyên là sử dụng pixel trong đó màu sắc của pixel phản ánh giá trị của thứ nguyên. Đối với tập dữ liệu có kích thước m, các kỹ thuật định hướng pixel tạo ra các cửa sổ m trên màn hình, một cho mỗi thứ nguyên. Giá trị kích thước m của bản ghi được ánh xạ đến m pixel tại các vị trí tương ứng trong cửa sổ. Màu sắc của các pixel phản ánh các giá trị tương ứng.

Bên trong một cửa sổ, các giá trị dữ liệu được sắp xếp theo một số thứ tự toàn cầu được chia sẻ bởi tất cả các cửa sổ. Thứ tự toàn cầu có thể thu được bằng cách sắp xếp tất cả các bản ghi dữ liệu theo cách có ý nghĩa cho nhiệm vụ trong tầm tay.

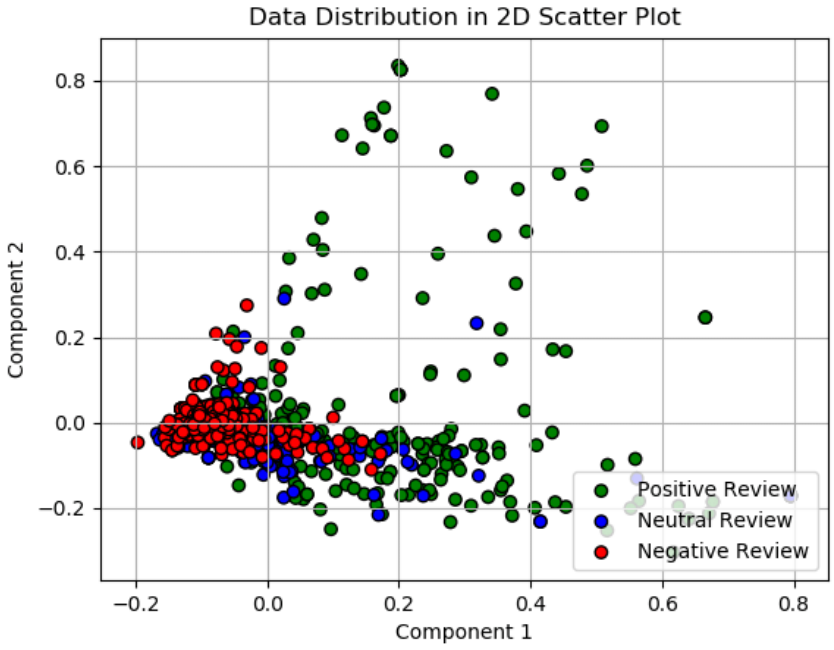


Hình 1.2: Trực quan hóa theo hướng pixel với dữ liệu *AllElectronics* bảng thông tin khách hàng bao gồm: *income*, *credit_ limit*, *transaction_ volume*, *age*.

Ta sắp xếp tất cả khách hàng theo thứ tự tăng dần thu nhập và sử dụng thứ tự này để sắp xếp dữ liệu khách hàng trong bốn cửa sổ trực quan hóa, như thể hiện trong Hình 1.2. Các màu điểm ảnh được chọn sao cho giá trị càng nhỏ, bóng càng nhẹ. Sử dụng trực quan hóa pixel, ta có thể dễ dàng quan sát những điều sau đây: tăng hạn mức tín dụng khi thu nhập tăng lên; khách hàng có thu nhập ở tầm trung có nhiều khả năng mua nhiều hơn từ *AllElectronics*; không có mối tương quan rõ ràng giữa thu nhập và tuổi tác.

Kỹ thuật trực quan hóa chiếu hình học

Một nhược điểm của các kỹ thuật hình ảnh hóa theo hướng pixel là chúng không thể giúp ta nhiều trong việc hiểu được sự phân bố dữ liệu trong một không gian đa hướng. Ví dụ, nó không thể hiển thị cho dù có một khu vực dày đặc trong một không gian con nhiều chiều. Kỹ thuật chiếu hình học giúp người dùng tìm thấy những dự đoán thú vị về các tập dữ liệu đa chiều. Thách thức trung tâm các kỹ thuật chiếu hình học cố gắng giải quyết là làm thế nào để hình dung một không gian nhiều chiều trên một màn hình 2-D.

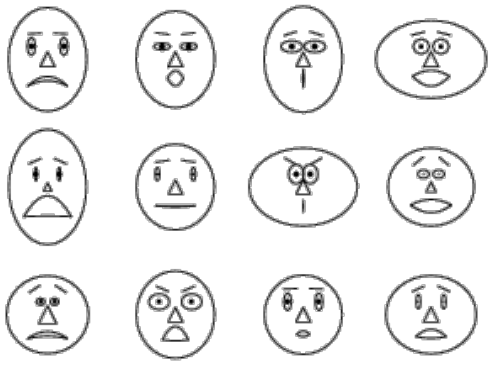


Hình 1.3: Ví dụ về trực quan hóa dữ liệu 2-D sử đồ thị phân tán

Một đồ thị phân tán hiển thị các điểm dữ liệu 2-D bằng cách sử dụng tọa độ Cartesian. Một thứ nguyên thứ ba có thể được thêm vào bằng cách sử dụng các màu sắc hoặc hình dạng khác nhau để đại diện cho các điểm dữ liệu khác nhau.

Kỹ thuật trực quan hóa dựa trên biểu tượng

Các kỹ thuật trực quan hóa dựa trên biểu tượng sử dụng các biểu tượng nhỏ để đại diện cho các giá trị dữ liệu đa diện. ta xem xét hai biểu tượng phổ biến dựa trên kỹ thuật: Khuôn mặt Chernoff và hình stick.

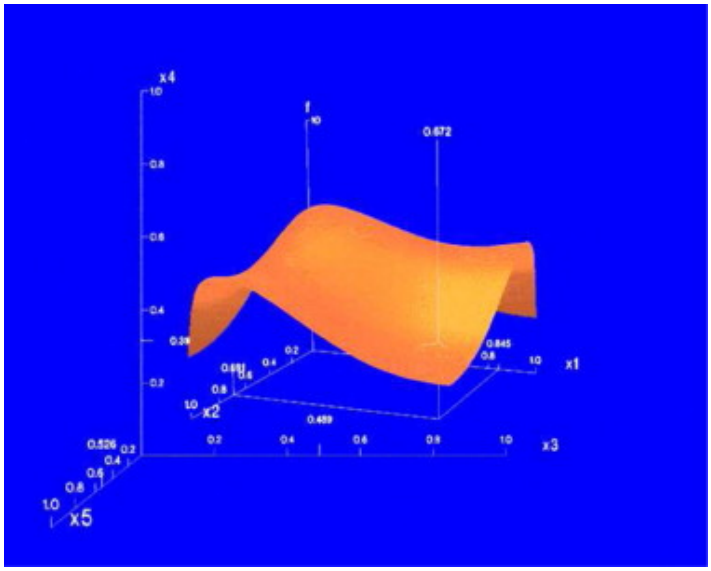


Hình 1.4: Khôn mặt Chernoff. Mỗi khuôn mặt đại diện cho một điểm dữ liệu n chiều.

Hình 1.5: Dữ liệu điều tra dân số được thể hiện bằng cách sử dụng hình stick.

Kỹ thuật trực quan hóa phân cấp

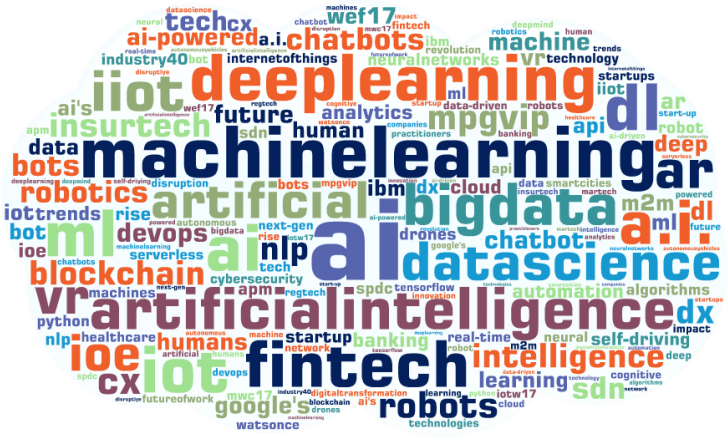
Các kỹ thuật trực quan hóa trên tập trung vào việc trực quan hóa nhiều chiều cùng một lúc. Tuy nhiên, đối với một tập dữ liệu lớn có số chiều lớn, sẽ rất khó để hình dung tất cả các kích thước cùng một lúc. Kỹ thuật trực quan hóa phân cấp phân vùng tất cả các kích thước thành các tập hợp con (tức là không gian con). Các không gian con được hình dung một cách phân cấp.



Hình 1.6: "Worlds-within-Worlds," còn được gọi là n-Vision, là một phương pháp trực quan hóa phân cấp đại diện.

Trực quan hóa dữ liệu phức tạp và quan hệ

Trong những ngày đầu, kỹ thuật trực quan hóa chủ yếu dành cho dữ liệu số. Gần đây, ngày càng có nhiều dữ liệu không phải là số, chẳng hạn như văn bản và mạng xã hội, đã có sẵn. Trực quan hóa và phân tích dữ liệu như vậy thu hút rất nhiều sự quan tâm. Có rất nhiều kỹ thuật trực quan hóa mới dành riêng cho các loại dữ liệu này. Ví dụ: nhiều người trên Web gắn thẻ các đối tượng khác nhau như hình ảnh, mục blog và đánh giá sản phẩm. *Tag cloud* là trực quan hóa thống kê các thẻ do người dùng tạo. Thông thường, trong textitTag cloud, thẻ được liệt kê theo thứ tự bảng chữ cái hoặc theo thứ tự ưa thích của người dùng. Tầm quan trọng của thẻ được biểu thị bằng cỡ phông chữ hoặc màu sắc. Hình 1.7 cho thấy một textitTag cloud để hình dung các thẻ phổ biến được sử dụng trí tuệ nhân tạo.



Hình 1.7: textitTag cloud các vấn đề

1.2.6 Diễn dịch kết quả

Kiến thức hoặc thông tin, mà ta có được thông qua quá trình khai thác dữ liệu, cần phải được trình bày theo cách mà các bên liên quan có thể sử dụng nó khi họ muốn. Dựa trên các yêu cầu kinh doanh, giai đoạn triển khai có thể đơn giản như tạo báo cáo hoặc phức tạp như quy trình khai thác dữ liệu có thể lặp lại trong toàn tổ chức. Trong giai đoạn triển khai, các kế hoạch triển khai, bảo trì và giám sát phải được tạo ra để thực hiện và cũng hỗ trợ trong tương lai. Từ quan điểm dự án, báo cáo cuối cùng của dự án cần tóm tắt kinh nghiệm dự án và xem xét dự án để xem những gì cần cải thiện tạo ra bài học kinh nghiệm. Một số dạng tri thức thu được của quá trình khai phá dữ liệu:

- Luận kết hợp.
- Cây quyết định.
- Quy tắc phân loại.
- Quy tắc quan hệ.
- Mô hình dự đoán:
  - Mô hình hàng xóm gần nhất.
  - Mô hình phân loại Bayes.
  - Mạng nơ-ron.
  - Mô hình hồi quy.
- Các mô hình phân cụm.

1.3 Các dạng dữ liệu

1.3.1 Dữ liệu nhị phân

Một thuộc tính nhị phân là một thuộc tính với chỉ có hai loại trạng thái: 0 hoặc 1, nơi 0 thường có nghĩa là không, và 1 có nghĩa là có. Thuộc tính nhị phân được gọi là Boolean nếu hai trạng thái tương ứng với đúng và sai.

Ví dụ Với thuộc tính người hút thuốc mô tả một đối tượng bệnh nhân, 1 chỉ ra rằng bệnh nhân hút thuốc, trong khi 0 chỉ ra rằng bệnh nhân không. Tương tự như vậy, giả sử bệnh nhân trải qua một bài kiểm tra y tế có hai kết quả có thể xảy ra. Các thuộc tính kiểm tra y tế là nhị phân, trong đó một giá trị của 1 có nghĩa là kết quả của các thử nghiệm cho bệnh nhân là dương tính, trong khi 0 có nghĩa là âm tính.

Một thuộc tính nhị phân là đối xứng nếu cả hai trạng thái của nó đều số lượng giá trị giống nhau; có nghĩa là, không có ưu tiên kết quả nên được mã hóa là 0 hoặc 1. Một ví dụ như vậy có thể là giới tính thuộc tính có các trạng thái nam và nữ.

Một thuộc tính nhị phân là không đối xứng nếu kết quả của các trạng thái không quan trọng như nhau, chẳng hạn như kết quả dương tính và âm tính của một xét

nghiệm HIV. Theo quy ước, ta mã hóa kết quả quan trọng nhất, thường là kết quả hiếm nhất, bằng 1 (ví dụ: dương tính với HIV) và kết quả còn lại bằng 0 (ví dụ: âm tính với HIV).

1.3.2 Dữ liệu phân lớp

Các giá trị của một thuộc tính phân lớp là biểu tượng hoặc tên của sự vật. Mỗi giá trị đại diện cho một số loại thể loại, mã hoặc trạng thái, và do đó các thuộc tính phân lớp cũng được gọi là phân loại. Các giá trị không có bất kỳ thứ tự có ý nghĩa. Trong khoa học máy tính, các giá trị còn được gọi là liệt kê.

Ví dụ Giả sử rằng màu tóc và tình trạng hôn nhân là hai thuộc tính mô tả các đối tượng người. Các giá trị có thể có cho màu tóc là đen, nâu, vàng, đỏ, xám và trắng. Các thuộc tính tình trạng hôn nhân có thể đưa vào các giá trị độc thân, kết hôn, ly dị, và góa. Cả hai màu tóc và tình trạng hôn nhân là thuộc tính phân lớp. Một ví dụ khác của một thuộc tính phân lớp là nghề nghiệp, với các giá trị giáo viên, nha sĩ, lập trình viên, nông dân.

1.3.3 Dữ liệu thứ tự

Một thuộc tính thứ tự là một thuộc tính với các giá trị có thể có một thứ tự có ý nghĩa hoặc xếp hạng trong số đó, nhưng độ lớn giữa các giá trị kế tiếp không được biết đến. Thuôc tính phân lớp được đại diện bởi mode và trung vị, giá trị trung bình không cần được xử dụng.

Ví dụ Giả sử rằng kích thước của đồ uống có sẵn tại một nhà hàng thức ăn nhanh, thuộc tính phân lớp này có ba giá trị có thể: nhỏ, trung bình và lớn. Các giá trị có một chuỗi có ý nghĩa (tương ứng với việc tăng kích thước đồ uống); tuy nhiên, ta không thể nói từ các giá trị lớn hơn bao nhiêu, nói rằng, một phương tiện là hơn một lớn. Các ví dụ khác về các thuộc tính thứ tự bao gồm lớp (ví dụ: A+, A, A-, B+, v.v.) và xếp hạng.

Lưu ý rằng các thuộc tính phân lớp, nhị phân và thứ tự là định tính. Đó là, nó mô tả một tính năng của một đối tượng mà không đưa ra một kích thước thực tế hoặc số lượng. Các giá trị của các thuộc tính định tính như vậy thường là các từ đại diện cho các danh mục. Nếu số nguyên được sử dụng, chúng đại diện cho mã máy tính cho các danh mục, trái ngược với số lượng có thể đo lường (ví dụ: 0 cho kích thước đồ uống nhỏ, 1 cho trung bình và 2 cho lớn).

1.3.4 Dữ liệu giá trị khoảng

Các thuộc tính khoảng được đo trên thang đo của các đơn vị có kích thước bằng nhau. Các giá trị của các thuộc tính khoảng có thứ tự và có thể dương, 0 hoặc âm. Do đó, ngoài việc cung cấp một bảng xếp hạng các giá trị, các thuộc tính như vậy cho phép ta so sánh và định lượng sự khác biệt giữa các giá trị. Bởi vì các thuộc tính

khoảnglà số, ta có thể tính toán giá trị trung bình của chúng, ngoài tính trung vị và mode có thể được xử dụng.

Ví dụ Thuộc tính nhiệt độ là thuộc tính khoảng. Giả sử rằng ta có giá trị nhiệt độ ngoài trời cho một số ngày khác nhau, mỗi ngày là một đối tượng. Bằng cách đặt hàng các giá trị, ta có được một thứ hạng của các đối tượng đối với nhiệt độ. Ngoài ra, ta có thể định lượng sự khác biệt giữa các giá trị. Ví dụ, nhiệt độ 20 độ C cao hơn năm độ so với nhiệt độ 15 độ C.

Nhiệt độ trong Celsius và Fahrenheit không có điểm 0 thực sự, nghĩa là cả 0 độ C và 0 độ F đều không cho biết "không có nhiệt độ". (Ví dụ, trên thang độ C, đơn vị đo là 1/100 giữa nhiệt độ tan và nhiệt độ nóng chảy của nước dưới áp suất khí quyển). Mặc dù ta có thể tính toán sự khác biệt giữa các giá trị nhiệt độ, ta không thể nói về một giá trị nhiệt độ như là một bội số của một giá trị khác. Nếu không có số không thực sự, ta không thể nói, ví dụ, rằng 10 độ C ấm gấp đôi 5 độ C. Đó là, ta không thể nói về các giá trị về tỷ lệ.

1.3.5 Dữ liệu thuộc giá trị tỉ lệ

Thuộc tính tỷ lệ thu nhỏ là thuộc tính số có điểm 0 thực sự. Nghĩa là, nếu một phép đo được chia tỷ lệ, ta có thể nói về một giá trị là bội (hoặc tỷ lệ) của một giá trị khác. Ngoài ra, các giá trị được sắp xếp theo thứ tự và ta cũng có thể tính toán sự khác biệt giữa các giá trị, cũng như trung bình, trung vị và mode.

Ví dụ Thuộc tính tỷ lệ thu nhỏ. Không giống như nhiệt độ trong Celsius và Fahrenheit, thang nhiệt Kelvin (K) có điểm được coi là một điểm 0 thực sự (0 độ K = -273,15 độ C): Đó là điểm mà tại đó các hạt bao gồm vật chất không có động năng.

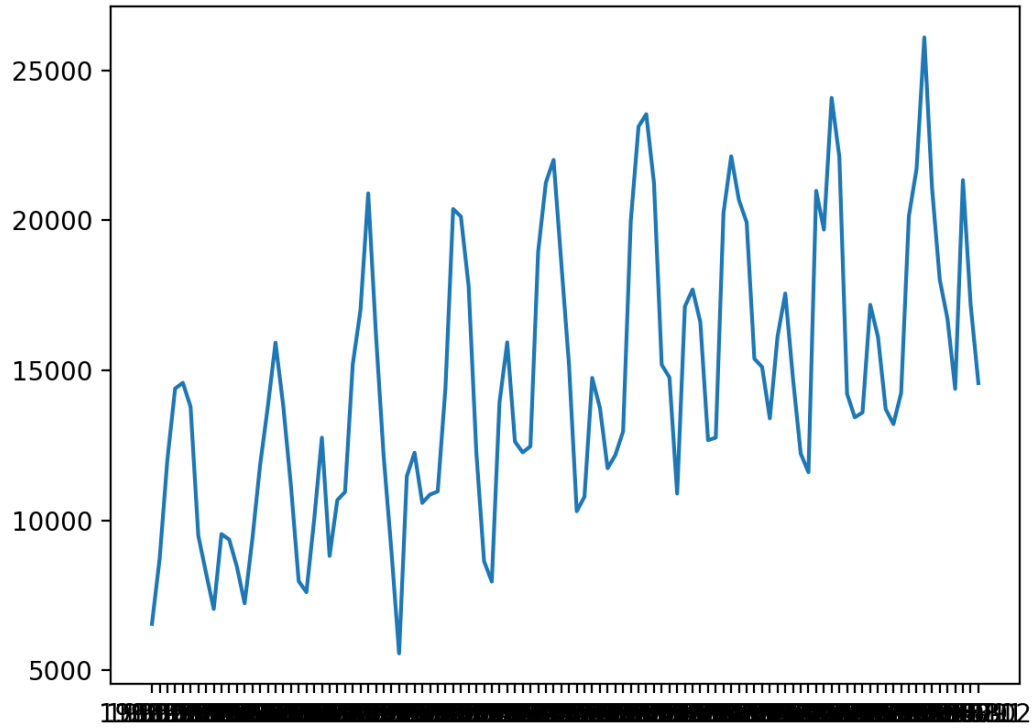
1.3.6 Dữ liệu chuỗi và chuỗi thời gian

Chuỗi thời gian trong thống kê, xử lý tín hiệu, kinh tế lượng và toán tài chính là một chuỗi các điểm dữ liệu, được đo theo từng khoảng khắc thời gian liền nhau theo một tần suất thời gian thống nhất. Một số khái niệm liên quan tới chuỗi thời gian

-

Ví dụ về chuỗi thời gian là giá đóng cửa của chỉ số Dow Jones hoặc lưu lượng hàng năm của sông Nin tại Aswan.

Phân tích chuỗi thời gian bao gồm các phương pháp để phân tích dữ liệu chuỗi thời gian, để từ đó trích xuất ra được các thuộc tính thống kê có ý nghĩa và các đặc điểm của dữ liệu. Dự đoán chuỗi thời gian là việc sử dụng mô hình để dự đoán các sự kiện thời gian dựa vào các sự kiện đã biết trong quá khứ để từ đó dự đoán các điểm dữ liệu trước khi nó xảy ra (hoặc được đo). Chuỗi thời gian thường được vẽ theo các đồ thị.



Hình 1.8: Đồ thị chuỗi thời gian

1.3.7 Dữ liệu rời rạc và liên tục

Dữ liệu rời rạc

Các thuật toán phân loại được phát triển từ lĩnh vực học máy thường nói về các thuộc tính là rời rạc hoặc liên tục. Mỗi loại có thể được xử lý một cách khác nhau. Một thuộc tính rời rạc có một tập hợp các giá trị hữu hạn hoặc vô hạn, có thể hoặc không thể được biểu thị dưới dạng số nguyên. Các thuộc tính màu tóc, người hút thuốc, kiểm tra y tế, và kích thước đồ uống mỗi có một số hữu hạn của các giá trị, và do đó là rời rạc. Lưu ý rằng các thuộc tính rời rạc có thể có các giá trị số, chẳng hạn như 0 và 1 cho các thuộc tính nhị phân hoặc, các giá trị 0 đến 110 cho tuổi thuộc tính. Một thuộc tính là vô hạn nếu tập hợp các giá trị có thể là vô hạn nhưng các giá trị có thể được đặt trong một tương ứng một-một với các số tự nhiên. Ví dụ: ID khách hàng thuộc tính là vô hạn. Số lượng khách hàng có thể phát triển đến vô cùng, nhưng trong thực tế, tập hợp các giá trị thực tế có thể đếm được (nơi các giá trị có thể được đặt trong thư từ một-một với tập hợp các số nguyên). Mã zip là một ví dụ khác.

Dữ liệu liên tục

Nếu một thuộc tính không rời rạc, nó là liên tục. Các thuật ngữ thuộc tính số và thuộc tính liên tục thường được sử dụng thay thế cho nhau trong văn học. (Điều này có thể gây nhầm lẫn bởi vì, theo nghĩa cổ điển, các giá trị liên tục là số thực, trong khi giá trị số có thể là số nguyên hoặc số thực.) Trong thực tế, các giá trị thực được thể hiện bằng cách sử dụng một số lượng hữu hạn các chữ số. Thuộc tính liên tục thường được thể hiện dưới dạng biến điểm nổi.

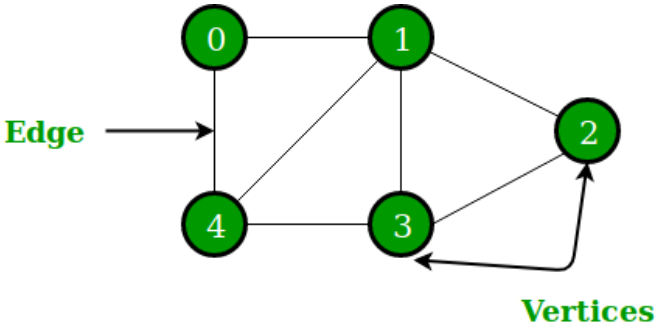
1.3.8 Dữ liệu văn bản

Dữ liệu dạng văn bản là những văn bản trong sử dụng trong các lĩnh vực của đời sống, ví dụ cụ thể, các công ty thường lưu trữ các báo cáo, các hợp đồng, dữ liệu quan trọng khác,... dưới dạng văn bản hoặc các bản ghi để tiện trao đổi thông tin giữa các cá nhân trong công ty và bên ngoài công ty (ví dụ qua e-mail). Ngày nay các dữ liệu trên được mã hóa, và số hóa phục vụ cho việc bảo mật và phân tích trong tương lai bằng các công cụ Data mining.

Khai thác văn bản bao gồm các bước cơ bản như: tiền xử lý, học mô hình, phán đoán, tổng hợp phân tích và trình bày kết quả. Tiền xử lý có thể gồm việc phân tách đoạn văn bản thành các đoạn nhỏ hơn, làm giàu văn bản bằng các tri thức bên ngoài, hoặc loại bỏ những thông tin nhiễu trong văn bản. Quá trình học là quá trình tìm ra các mẫu trong một tập các văn bản đã được tiền xử lý hoặc chưa qua tiền xử lý, kết quả quá trình học là một mô hình biểu diễn các mẫu được tìm thấy. Quá trình phán đoán là quá trình áp dùng mô hình vừa học được trên các văn bản mới, văn bản mới sẽ được gán nhãn thêm thông tin. Cuối cùng là quá trình tổng hợp và trình bày kết quả. Khai phá văn chia thành các vấn đề nhỏ hơn bao gồm phân loại tài liệu (text categorization, text classification), nhóm tài liệu (text clustering), trích xuất thực thể (concept/entity extraction), khai phá quan điểm (sentiment analysis), tóm tắt tài liệu (document summarization), và trích xuất quan hệ giữa các thực thể (entity relation modeling).

1.3.9 Dữ liệu đồ thị

Một đồ thị (Graph) là một dạng biểu diễn hình ảnh của một tập các đối tượng, trong đó các cặp đối tượng được kết nối bởi các link. Các đối tượng được nối liền nhau được biểu diễn bởi các điểm được gọi là các đỉnh (vertices), và các link mà kết nối các đỉnh với nhau được gọi là các cạnh (edges). Nói chung, một đồ thị là một cặp các tập hợp (V, E), trong đó V là tập các đỉnh và E là tập các cạnh mà kết nối các cặp điểm.



Hình 1.9: Ví dụ về đô thị

Một số khái niệm được sử dụng trong đồ thị:

- Đỉnh (Vertex): Mỗi nút của hình được biểu diễn như là một đỉnh. Trong ví dụ trên, các hình tròn biểu diễn các đỉnh. Do đó, các điểm từ 0 tới 4 là các đỉnh.
- Cạnh (Edge): Cạnh biểu diễn một đường nối hai đỉnh. Trong ví dụ trên, các đường nối 0 và 1, 0 và 4, ... là các cạnh.
- Kề nhau: Hai đỉnh là kề nhau nếu chúng được kết nối với nhau thông qua một cạnh. Trong hình dưới, 0 là kề với 1; 1 là kề với 2, ...
- Đường: Đường biểu diễn một dãy các cạnh giữa hai đỉnh. Trong hình dưới, 0432 biểu diễn một đường từ đỉnh 0 tới đỉnh 2.

1.4 Các dạng phân tích dữ liệu

1.4.1 Phân tích mô tả

Phân tích mô tả (Descriptive Analytics) là một phương pháp thống kê được dùng để tìm kiếm và thu gọn những dữ liệu trong lịch sử với mục đích trích xuất được những thôn tin hữu ích, tìm ra những quy luật nằm trong dữ liệu. Tổng hợp dữ liệu và khai phá dữ liệu là hai kỹ thuật thường đường sử dụng trong phân tích mô tả để khám phá những thông tin ẩn giấu trong dữ liệu quá khứ. Đầu tiên, dữ liệu cần phải được thu thập và sắp xếp bởi các kỹ thuật tổng hợp dữ liệu với mục đích chuyển dữ liệu về những dạng thức dễ quản lý hơn, phù hợp hơn cho việc phân tích sau này. Sang đến bước thứ hai của quá trình phân tích, các công cụ khai phá dữ liệu sẽ được sử dụng cho việc tìm ra những quy của dữ liệu. Ví dụ với lượng dữ liệu về học viên được thu thập trong một lớp học, quá trình khai phá dữ liệu có thể tìm ra được những quy luật, cách thức mà người học tương tác với nội dung của bài học và với môi trường xung quanh.

Phân tích mô tả có thể được coi là bước đầu tiên trong một quy trình kinh doanh thông minh (Business Intelligence). Nó tạo ra một cơ sở cho các phân tích tiếp theo như phân tích chẩn đoán (Diagnostic Analytics) hay phân tích dự báo (Predictive Analytics).

1.4.2 Phân tích dự báo

Phân tích dự báo (Predictive Analytics) là quá trình sử dụng dữ liệu thống kê để đưa ra những dự báo dựa trên dữ liệu. Quá trình này sử dụng dữ liệu cùng với các kỹ thuật phân tích, thống kê và máy học để tạo ra một mô hình dự báo phục vụ việc tiên đoán những sự kiện có thể xảy ra trong tương lai.

Thuật ngữ "phân tích dự báo" mô tả ứng dụng của một kỹ thuật thống kê hay máy học để tạo ra một dự báo có tính định lượng về tương lai. Thông thường, các kỹ thuật học có giám sát (Supervised Learning) sẽ được sử dụng để dự đoán một giá trị nào đó trong tương lai hoặc để ước lượng một xác suất nào đó.

Phân tích dự báo được bắt đầu với một mục tiêu kinh doanh nào đó, ví dụ, giảm thiểu rác thải, tiết kiệm thời gian, hay cắt giảm chi phí. Quá trình này "nén" dữ liệu lại thành các mô hình mà sau này có thể sử dụng để dự đoán tương lai nhằm hỗ trợ cho việc đạt được những mục tiêu kinh doanh đề ra.

Luồng công việc của phân tích dự báo

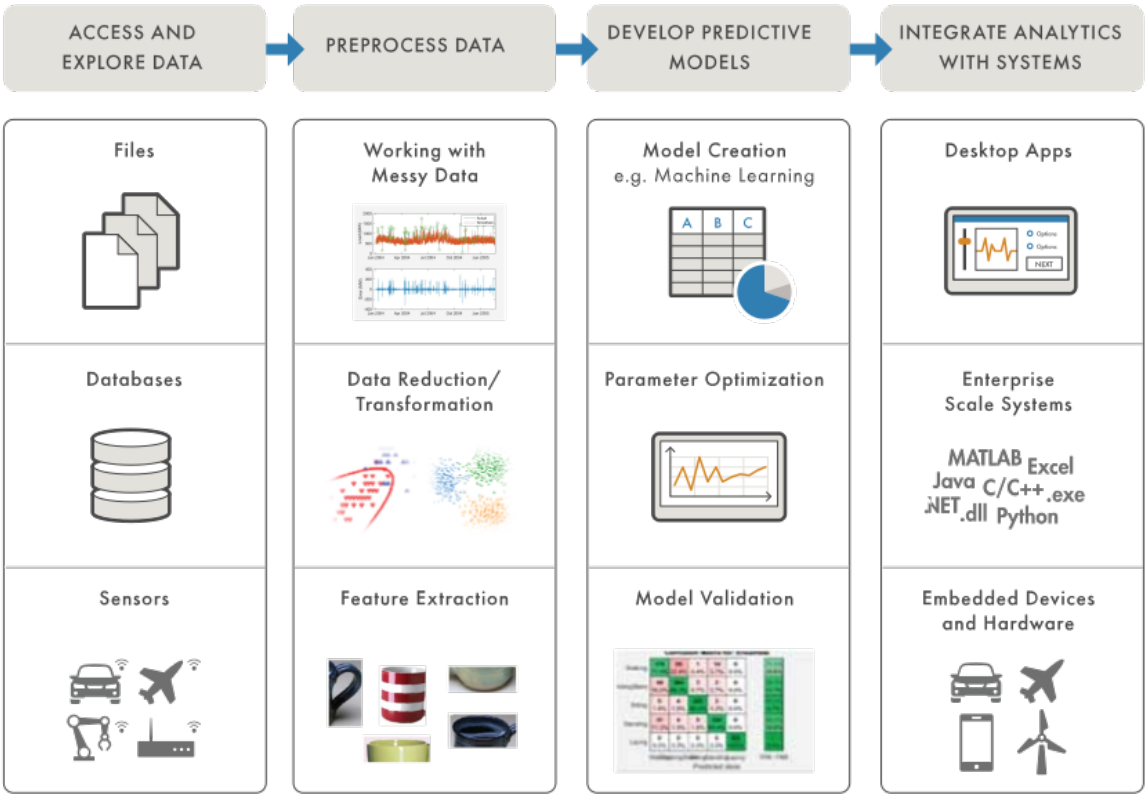
1. Thu thập dữ liệu từ các nguồn khác nhau như web, cơ sở dữ liệu, bảng tính, v.v.
2. Làm sạch dữ liệu bằng cách loại bỏ các phần tử ngoại lai và kết hợp các nguồn dữ liệu lại với nhau.
3. Xây dựng một mô hình dự báo bằng các dữ liệu tổng hợp được nhờ vào việc sử dụng các giải pháp thống kê hay máy học.
4. Tích hợp mô hình vào một hệ thống dự báo làm việc trong một môi trường thực tế.

1.4.3 Phân tích tối ưu

Phân tích tối ưu là một loại phân tích dữ liệu sử dụng công nghệ cho việc hỗ trợ quá trình ra quyết định, giúp đưa ra quyết định một cách tốt hơn thông qua phân tích các dữ liệu thô. Cụ thể, phân tích tối ưu quan tâm đến những thông tin về những tình huống có thể xảy ra trong tương lai, các tài nguyên sẵn có, hiệu năng hệ thống vận hành trong quá khứ và hiệu năng hiện tại, từ đó đề xuất ra một quá trình cho việc hành động hay chiến lược.

Phân tích tối ưu phụ thuộc vào các công nghệ về trí tuệ nhân tạo để có thể vận hành, ví dụ như máy học. Nhờ có máy học, ta có thể xử lý một lượng dữ liệu vô cùng lớn, thứ có thể nói là bất khả thi với những công nghệ trước đây. Mỗi khi có dữ liệu mới, hệ thống có thể tự động điều chỉnh để tận dụng số dữ liệu đó với một tốc độ đáng kinh ngạc.

Phân tích tối ưu có thể giúp ngăn ngừa gian lận, hạn chế rủi ro, tăng hiệu quả, đáp ứng các mục tiêu kinh doanh và tạo ra nhiều khách hàng trung thành hơn. Tuy nhiên, phân tích mô tả không phải là hoàn hảo. Nó chỉ tỏ ra hiệu quả nếu các tổ chức biết cần phải trả lời những câu hỏi nào và cần phải hành động như thế nào với những câu trả lời. Nếu các giải định đầu vào không hợp lệ, kết quả đầu ra sẽ không chính xác.



Hình 1.10: Luồng công việc của một phân tích dự báo.

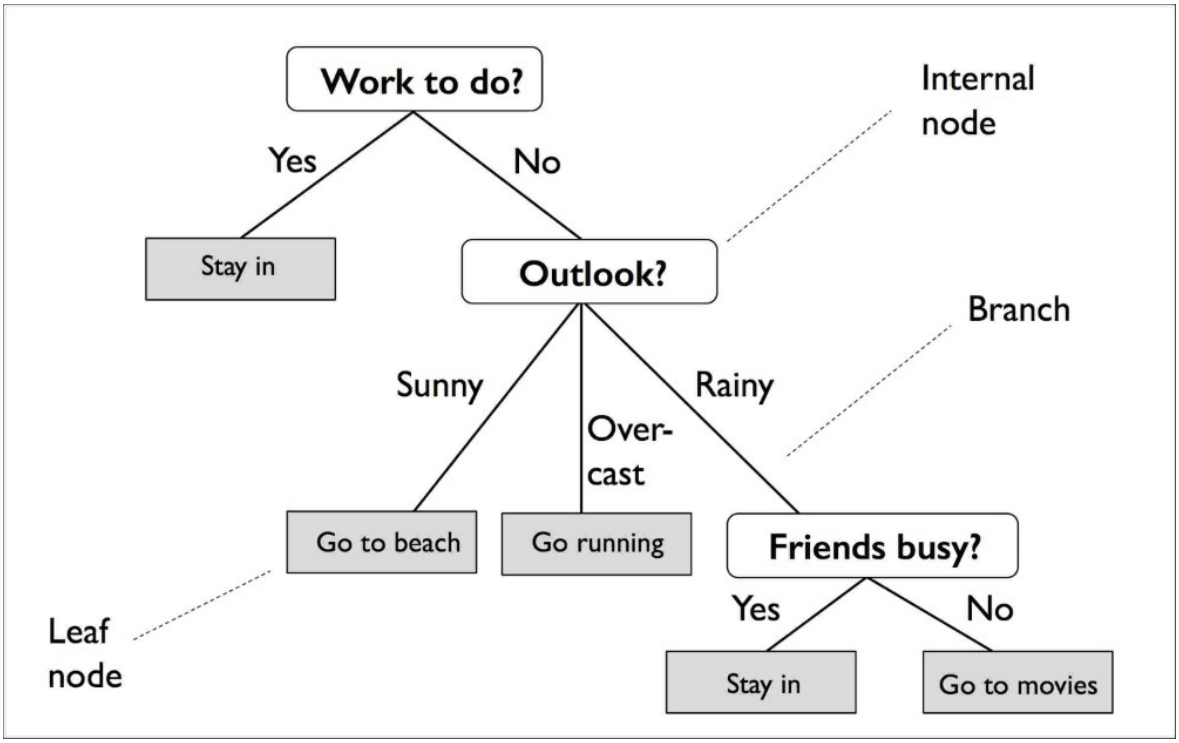
1.5 Các tác vụ phân tích dữ liệu

1.5.1 Phân lớp

Phân lớp là quá trình tìm một mô hình (hoặc hàm số) cho việc mô tả và phân tách các lớp dữ liệu của một bộ dữ liệu nào đó. Mô hình được tạo ra dựa trên những thống kê của một tập gồm các dữ liệu huấn luyện (những điểm dữ liệu đã được gán nhãn để xác định lớp mà nó thuộc về). Nhiệm vụ của mô hình sẽ là dự đoán nhãn của các đối tượng chưa được gán trước những thông tin đó.

Có nhiều cách để biểu diễn một mô hình phân lớp, nó có thể tồn tại dưới dạng các câu lệnh rẽ nhánh (các luật IF-ELSE), cây quyết định, công thức toán học hay thậm chí là mạng nơ-ron nhân tạo. Một mô hình cây quyết định có dạng một cấu trúc cây hình lưu đồ , trong đó từng nút biểu thị một câu hỏi dựa vào một giá trị thuộc tính nào đó đặc trưng cho dữ liệu, mỗi nhánh biểu thị một kết quả sau khi trả lời câu hỏi được đưa ra và các nút lá của cây đóng vai trò biểu diễn các lớp hoặc các phân phối của chúng. Mô hình cây quyết định có thể dễ dàng được chuyển đổi thành các luật phân lớp. Một mạng nơ-ron khi được sử dụng cho mục đích phân lớp thường được cấu tạo bởi các đơn vị nhỏ xử lý nhỏ hơn có dạng như nơ-ron của người. Các nơ-ron này được kết nối với nhau nhờ các liên kết có trọng số. Ngoài hai giải pháp cụ thể nêu trên, ta có thể sử dụng nhiều phương pháp khác để xây dựng nên các mô hình phân lớp, ví dụ như Naive Bayesian, Support Vector Machine và k-Nearest-Neighbor.

Như minh họa trong Hình 1.11, ta có một ví dụ đơn giản cho việc sử dụng mô hình cây quyết định cho việc phân lớp một đối tượng vào một trong hai lớp: ở nhà hoặc đi xem phim, dựa vào trạng thái hiện có của đối tượng đó như: có việc cần phải làm hay không, ngoài trời như thế nào, bạn bè có bận hay không.



Hình 1.11: Một ví dụ về việc sử dụng cây quyết định cho tác vụ phân lớp.

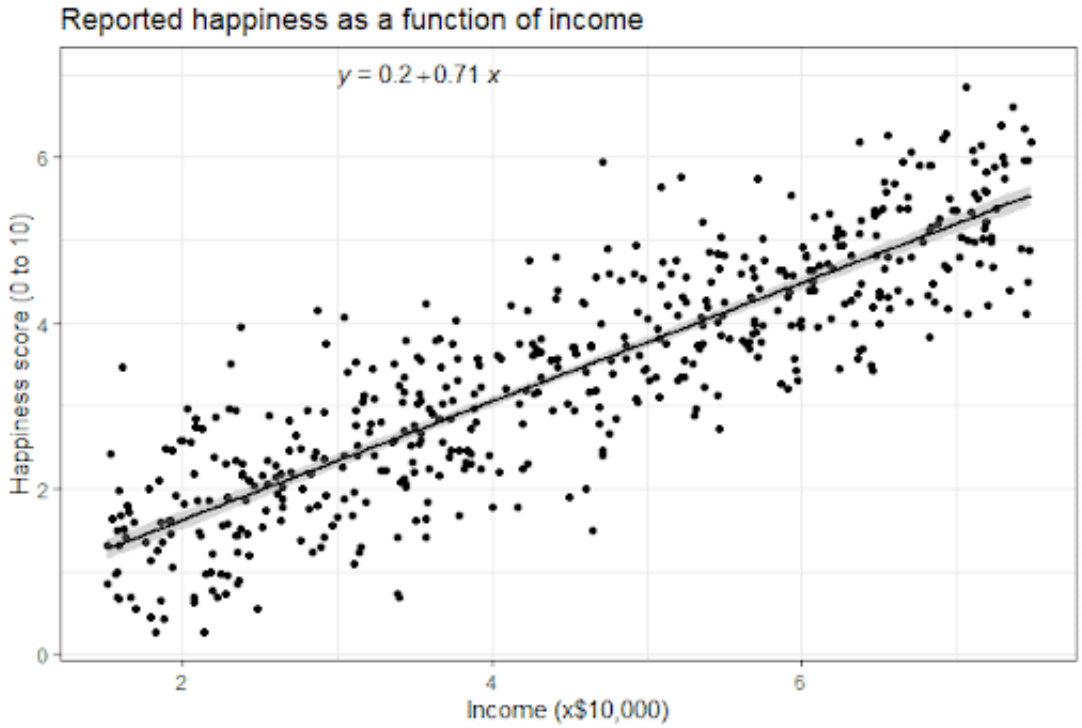
1.5.2 Phân tích hồi quy

Khác với các mô hình phân lớp tiến hành việc dự đoán trên các giá trị có tính phân loại (rời rạc và không có thứ tự), các mô hình hồi quy làm việc với những giá trị liên tục. Một cách cụ thể, các mô hình hồi quy được sử dụng để dự đoán các giá trị số còn thiếu thay vì các lớp của đối tượng như trong phân lớp. Phân tích hồi quy là một phương pháp học thống kê thường được sử dụng cho các dự đoán là những giá trị số. Ngoài ra, phân tích hồi quy còn bao gồm việc xác định xu hướng của các phân phối dựa vào dữ liệu hiện có.

Như minh họa trong Hình 1.12, ta có một ví dụ đơn giản cho việc sử dụng mô hình hồi quy tuyến tính cho việc dự đoán điểm hạnh phúc (Happyness score) cho một đối tượng dựa vào mức thu nhập của cá nhân đó.

1.5.3 Phân tích sự kết hợp

Phân tích sự kết hợp là một chức năng của khai phá dữ liệu có nhiệm vụ khám phá ra xác suất của việc xuất hiện đồng thời của các mục nằm trong một tập hợp. Những



Hình 1.12: Một ví dụ về việc sử dụng thuật toán hồi quy tuyến tính.

mối quan hệ giữa các mục thường hay xuất hiện cùng nhau được biểu diễn dưới dạng các luật kết hợp.

Luật kết hợp thường được sử dụng trong việc phân tích các giao dịch mua bán hàng hóa. Ví dụ, một khách hàng mua bia sẽ thường mua kèm với lạc rang. Với những con số cụ thể, giả sử như thông qua việc phân tích sự kết hợp ta phát hiện ra rằng 85% các phiên giao dịch có bao gồm bia cũng có xuất hiện lạc rang. Mối quan hệ đó có thể được biểu diễn bởi một luật như sau: Bia -> lạc rang với độ tự tin 85%.

Ứng dụng của việc phân tích sự kết hợp còn được gọi là phân tích giỏ thị trường (Market-basket Analysis). Nó có tác dụng vô cùng lớn trong việc điều phối bán hàng, tung ra các chương trình khuyến mãi và tìm các xu thế trong việc kinh doanh. Ngoài ra, việc phân tích giỏ thị trường còn có thể giúp cho việc sắp xếp các mặt hàng trong siêu thị.

Phân tích sự kết hợp còn có những ứng dụng vô cùng quan trọng trong rất nhiều lĩnh vực khác. Ví du, trong thương mại điện tử, luật kết hợp có thể được sử dụng để cá nhân hóa trang web của từng người dùng. Một mô hình luật kết hợp có thể tìm ra được một luật như việc một người dùng ghé thăm trang A và trang B có 60% khả năng cũng sẽ ghé thăm trang C trong cùng một phiên làm việc. Dựa trên luật đó, ta có thể tự động tạo ra một đường dẫn đến trang C cho người dùng đó. Luật kết hợp có thể được biểu diễn như sau: A và B -> C với độ tự tin 60%

Như minh họa trong Hình 1.13, ta có một ví dụ đơn giản của ứng dụng phân tích sự kết hợp cho việc tìm các mặt hàng thường hay được mua cùng với nhau; hay nói cách khác, các mặt hàng thường xuất hiện cùng nhau trong các hóa đơn mua hàng.

Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B\&C \Rightarrow D$	1/5	1/3	5/9

Hình 1.13: Một ví dụ về việc phân tích sự kết hợp cho các đơn hàng của siêu thị.

1.5.4 Phân tích phân cụm

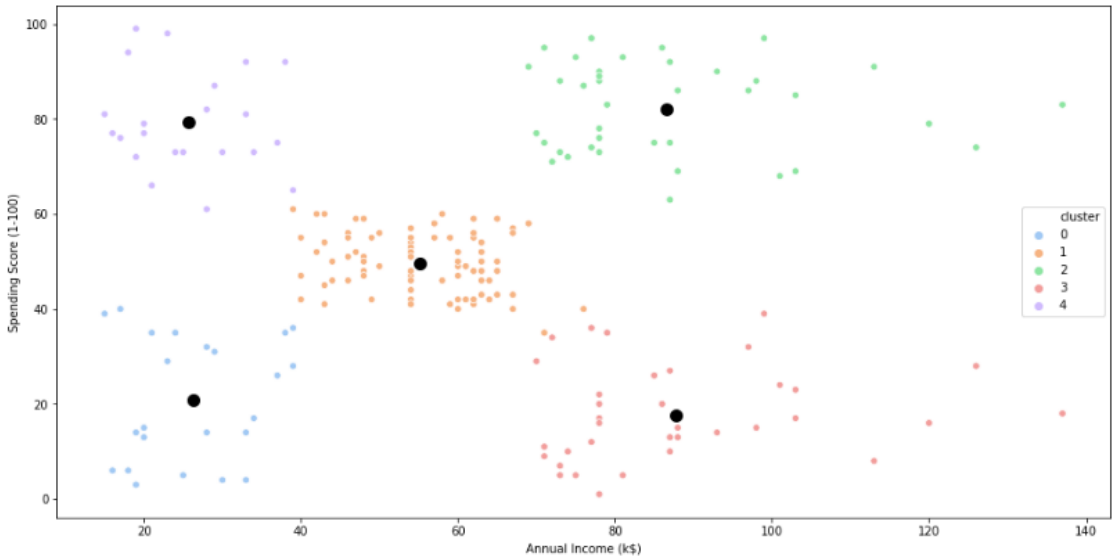
Khác với phân tích phân lớp và phân tích hồi quy, phân tích phân cụm được liệt kê vào danh sách các thuật toán học không giám sát (Unsupervised Learning); điều này có nghĩa là các mô hình phân cụm không quan tâm đến nhãn của các điểm dữ liệu trong suốt quá trình huấn luyện. Trong nhiều trường hợp, nhãn lớp của dữ liệu thậm chí không tồn tại lúc bắt đầu của quá trình phân tích. Phân tích phân cụm có thể được sử dụng để tạo ra nhãn lớp cho một tập các điểm dữ liệu. Các đối tượng được phân cụm dựa trên nguyên lý tối đa hóa khoảng cách của các phần tử khác cụm và tối thiểu hóa khoảng cách giữa các phần tử thuộc cùng một cụm. Hay nói cách khác, các cụm được hình thành sao cho các phần tử thuộc cùng một cụm phải giống nhau nhiều hơn khi so với những phần tử khác hay thậm chí là khác xa so với các phần tử thuộc cụm khác. Mỗi cụm sau đó có thể được xem như một lớp đối tượng, từ đó sinh ra được luật để dùng sau này.

Như minh họa trong Hình 1.14, ta có một ví dụ đơn giản cho việc sử dụng mô hình K-Mean cho việc phân cụm khách hàng.

1.5.5 Phân tích chuỗi và chuỗi thời gian

Một chuỗi thời gian là một chuỗi các điểm dữ liệu được đánh chỉ mục theo thứ tự thời gian. Thông thường, một chuỗi thời gian là một chuỗi tuần tự các điểm dữ liệu được lấy cách đều nhau theo thời gian. Phân tích chuỗi thời gian là một công cụ vô cùng hữu ích cho phép ta thấy được sự biến đổi của các loại dữ liệu khác nhau như dữ liệu bất động sản, tài chính, ... theo thời gian. Một ví dụ về dữ liệu dạng chuỗi thời gian được minh họa như trong Hình 1.15.

Phân tích chuỗi thời gian bao gồm các phương pháp phục vụ việc phân tích dữ liệu dạng chuỗi thời gian với mục đích trích xuất các đặc trưng về thống kê cũng như các đặc trưng hữu ích khác của dữ liệu. Dự đoán chuỗi thời gian là việc sử dụng một mô hình để tiến hành đưa ra những dự đoán cho các giá trị trong tương lai dựa vào các



Hình 1.14: Một ví dụ về việc sử dụng K-Mean để tiến hành phân tích phân cụm.

dữ liệu quá khứ quan sát được.

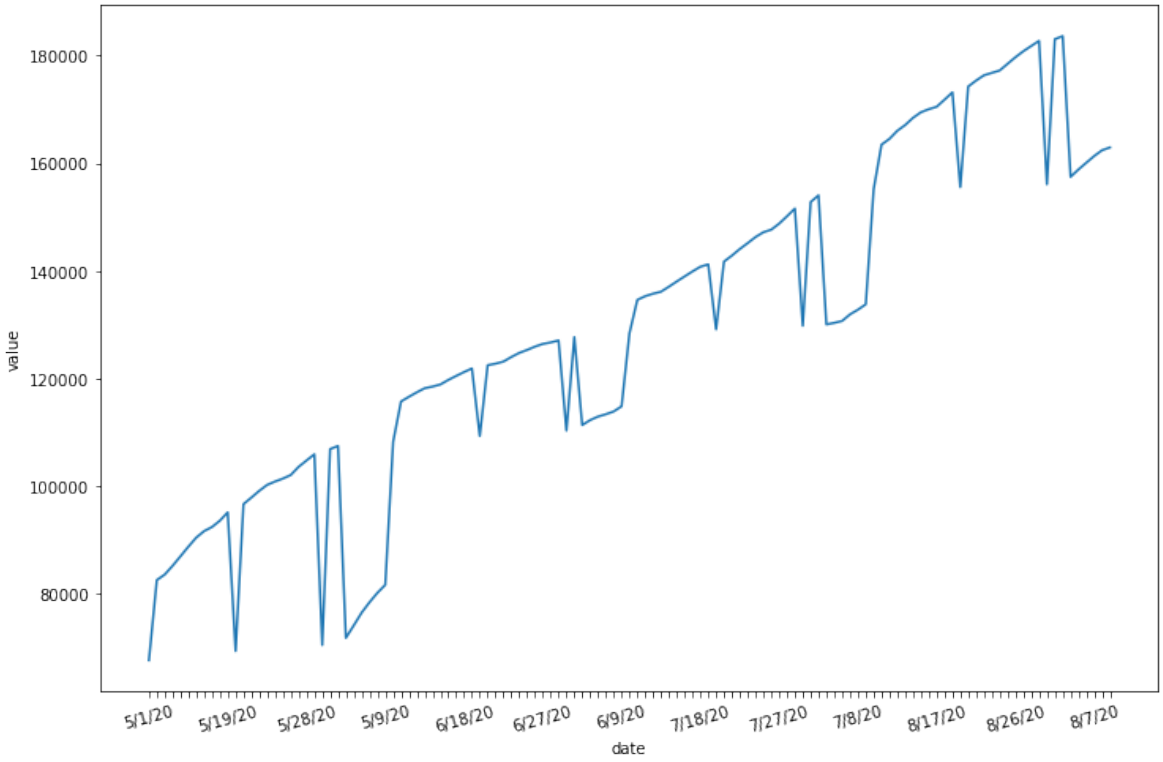
Một cách tự nhiên, dữ liệu chuỗi thời gian có đặc tính sắp xếp theo thứ tự thời gian trong đó. Phân tích chuỗi thời gian khác với phân tích dữ liệu về không gian với các quan sát thường liên quan đến vị trí địa lý. Một mô hình ngẫu nhiên cho một chuỗi thời gian sẽ thường phản ánh một sự thật rằng các quan sát gần nhau theo gian sẽ có liên quan chặt chẽ hơn những quan sát cách xa nhau. Thêm vào đó, các mô hình chuỗi thời gian thường tận dụng đặc tính thứ tự một chiều của thời gian, do đó những dữ liệu tại một thời điểm nào đó sẽ có liên quan đến dữ liệu ghi nhận được trong quá khứ thay vì là của tương lai.

Phân tích chuỗi thời gian có thể được áp dụng cho nhiều loại dữ liệu khác nhau, từ dữ liệu rời rạc cho đến dữ liệu liên tục.

1.6 Một số khái niệm máy học

1.6.1 Máy học là gì?

Máy học (Machine Learning) nghiên cứu về cách thức máy tính điện tử có thể học (hoặc nâng cao khả năng vận hành) dựa trên dữ liệu. Một mảng nghiên cứu chính của máy học liên quan đến việc lập trình máy tính cho phép nó có thể tự động học để có thể nhận dạng các quy luật phức tạp và đưa ra những quyết định thông minh dựa vào dữ liệu. Ví dụ, một bài toán kinh điển trong máy học đó là lập trình một chiếc máy tính sao cho nó có thể tự động nhận dạng các mã địa chính viết tay trên các bức thư sau khi được huấn luyện với một tập các ví dụ có sẵn.



Hình 1.15: Số lượng người chết do COVID19 tại Mỹ từ đầu tháng 5 đến cuối tháng 8 năm 2020.

1.6.2 Học không giám sát

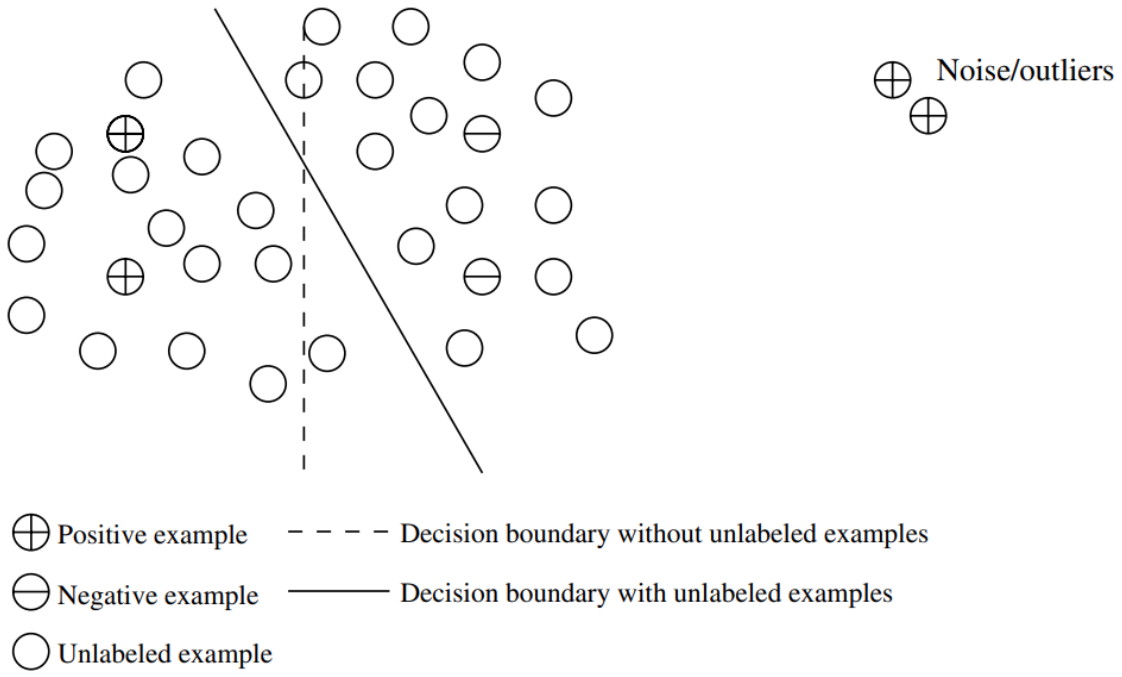
Học không giám sát (Unsupervised Learning) là một lớp các thuật toán máy học cho phép máy tính học từ những dữ liệu không hoặc chưa được gán nhãn. Phân cụm có thể được coi là một thuật toán như vậy. Trong thực tế, ta có thể sử dụng phân tích phân cụm để khám phá ra các lớp nằm trong dữ liệu. Ví dụ, một giải pháp học không giám sát có thể nhận vào một tập các ảnh của các chữ số viết tay từ 0 đến 9. Giả sử ta tìm được tổng cộng 10 cụm bằng một thuật toán phân cụm nào đó. Các cụm này có thể tương ứng với mười chữ số từ 0 đến 9; tuy nhiên, mô hình được huấn luyện với bộ dữ liệu đó không thể cho chúng ta biết ý nghĩa của các cụm tìm được là gì, do dữ liệu dùng cho việc huấn luyện không hề được gán nhãn.

1.6.3 Học có giám sát

Ngược lại với học không giám sát, học có giám sát (Supervised Learning) yêu cầu dữ liệu đầu vào cần phải được gán nhãn. Nói cách khác, cụm từ "có giám sát" ở đây xuất phát từ việc dữ liệu dùng cho việc huấn luyện đã được gán nhãn từ trước. Ví dụ trong bài toán nhận dạng mã địa chính, một tập hợp các ảnh biểu diễn những chữ số viết tay của mã địa chính và các bản dịch tương ứng cho chúng mà máy tính có thể hiểu sẽ được sử dụng như những ví dụ cho quá trình huấn luyện mô hình, giám sát quá trình học của mô hình được chọn.

1.6.4 Học bán giám sát

Học bán giám sát (Semi-supervised Learning) là một lớp các bài toán máy học sử dụng cả dữ liệu được và không được gán nhãn khi huấn luyện mô hình. Với một cách tiếp cận cụ thể, các điểm dữ liệu được gán nhãn được sử dụng để huấn luyện cho mô hình phân lớp và dữ liệu không gán nhãn được sử dụng để tinh chỉnh các đường biên quyết định dùng để phân tách các lớp. Một ví dụ về bài toán phân lớp nhị phân, ta có thể coi các điểm dữ liệu xuất hiện trong bài toán chỉ có thể thuộc về một trong hai lớp: âm tính và dương tính. Trong Hình 1.16, nếu không quan tâm đến những điểm dữ liệu không được gán nhã, đường nét đứt nằm trên hình vẽ có lẽ là biên quyết định tốt nhất mà mô hình có thể học được để phân cách hai lớp dữ liệu với nhau. Khi đưa thêm các điểm dữ liệu không được gán nhãn vào việc xây dựng biên quyết định, ta sẽ thu được đường nét liền thay cho đường nét đứt ban đầu. Thêm vào đó, ta có thể nhận thấy hai điểm dữ liệu nằm ở góc phía trên bên phải ngoài cùng có thể là hai điểm ngoại lai.



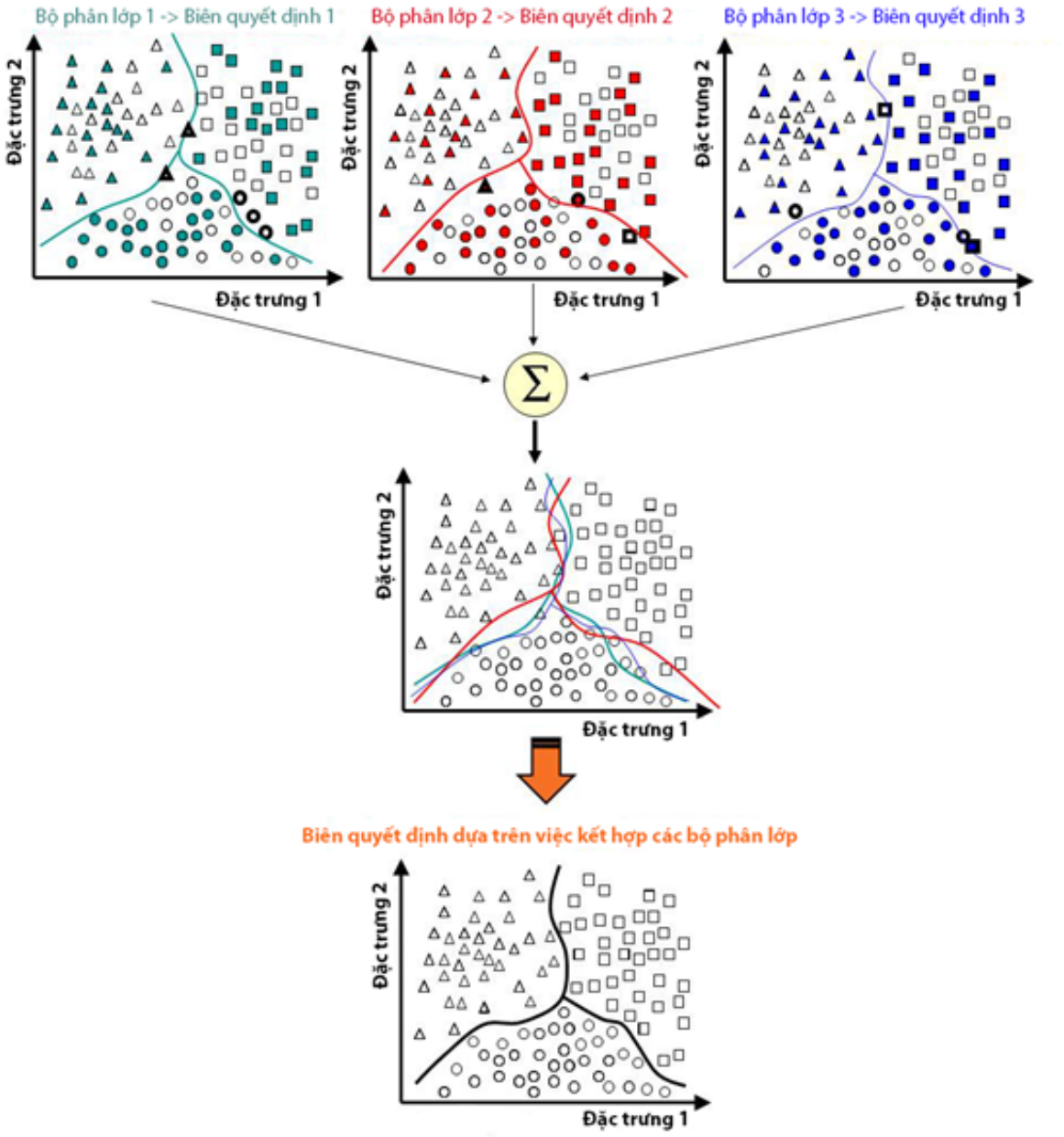
Hình 1.16: Học bán giám sát.

1.6.5 Học kết hợp

Học kết hợp là việc kết hợp nhiều mô hình lại với nhau nhằm mục đích thu được một giải pháp cho độ chính xác cao hơn bất kỳ mô hình độc lập nào được sử dụng. Giả sử ta muốn làm một bài toán phân lớp, có rất nhiều thuật toán để ta có thể sử dụng như perceptron nhiều lớp (Multi-Layer Perceptron – MLP), máy hỗ trợ vectơ (Support Vector Machines – SVM), cây quyết định (Decision Tree), mạng nơ-ron nhân tạo (Artificial Neural Networks – ANN), ... Mỗi thuật toán đều có những ưu nhược

điểm khác nhau, vậy giữa hàng tá các sự lựa chọn như vậy, ta nên chọn cái nào? Tiếp theo, sau khi đã chọn được một thuật toán ưng ý cho bài toán của mình, ta lại vướng phải một vấn đề là với thuật toán phân lớp mà ta đã chọn ta nên sử dụng bộ siêu tham số (hyperparameter) nào cho nó? Chẳng hạn khi sử dụng mạng nơ-ron nhân tạo, số lớp của mô hình, tỉ lệ học (learning rate), số vòng lặp mà trong mỗi vòng thuật toán sẽ đi qua hết toàn bộ tập dữ liệu luyện (epoch), ... đều là những siêu tham số mà ta phải quyết định. Với mỗi một bộ siêu tham số như vậy, ta lại có thể thu được những kết quả khác nhau trên cùng một dữ liệu đầu vào. Để khắc phục vấn đề này, ta có thể thử nhiều bộ siêu tham số khác nhau cho mô hình một cách có hệ thống và sau đó chọn ra bộ siêu tham số cho ta kết quả tốt nhất. Tuy nhiên, phương pháp này không thực sự hoàn hảo, kết quả thu được trên tập dữ liệu luyện (training set) và tập dữ liệu xác nhận (validation set) tốt không có nghĩa là nó cũng sẽ tốt trên bộ dữ liệu kiểm tra cuối cùng (test set). Trả lời cho hai câu hỏi trên cũng chính là việc đưa ra lựa chọn trong số tất cả (có thể là vô cùng) các bộ phân lớp có thể sử dụng cho bài toán của chúng ta. Ta có thể chọn một cách ngẫu nhiên, nhưng khả năng ta chọn được một bộ phân lớp tốt thường rất thấp. Vậy thay vì chỉ chọn ra một mô hình, tại sao ta không chọn nhiều mô hình và kết hợp sử dụng các kết quả của chúng, ví dụ như đơn giản chỉ là lấy trung bình của các kết quả đó, để đưa ra quyết định cuối cùng? Hay nói cách khác, thay vì chỉ dựa toàn bộ vào một mô hình duy nhất để đưa ra quyết định, ta sẽ dựa vào nhiều mô hình để việc đưa ra quyết định trở nên "khách quan" hơn. Cần phải đặc biệt lưu ý rằng không có gì đảm bảo việc kết hợp nhiều mô hình với nhau sẽ luôn cho ta kết quả tốt hơn khi chỉ sử dụng một mô hình tốt nhất trong những cái mà ta có. Tuy nhiên, việc này chắc chắn sẽ giúp ta giảm khả năng chọn phải một mô hình tồi.

Để quá trình này đạt hiệu quả, các mô hình được dùng để kết hợp với nhau phải có một mức độ đa dạng giữa chúng. Trong ngữ cảnh của bài toán phân lớp, sự đa dạng này (thường đạt được bằng cách sử dụng các bộ siêu tham số khác nhau cho từng bộ phân lớp) cho phép các bộ phân lớp sinh ra các biên quyết định khác nhau. Khi kết hợp các biên quyết định này lại với nhau một cách chiến lược ta có thể sẽ thu được biên quyết định mới tốt hơn sản phẩm của từng bộ phân lớp ban đầu. Một ví dụ đơn giản minh họa cho điều này được thể hiện trong Hình 1.17.



Hình 1.17: Kết hợp nhiều bộ phân lớp với nhau để thu được biên quyết định mới tốt hơn.

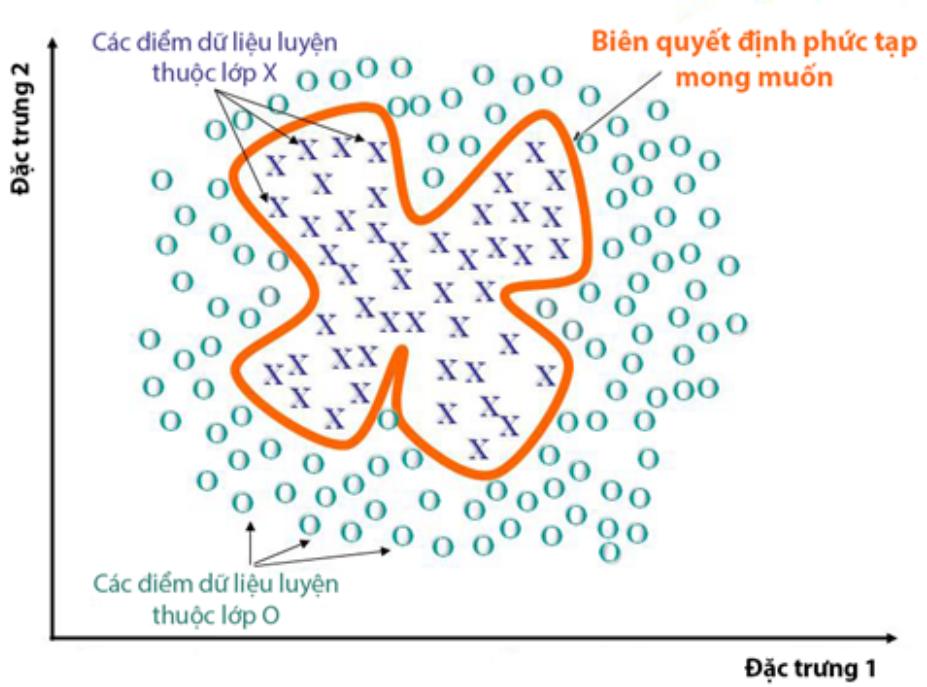
Quá nhiều hoặc quá ít dữ liệu

Các hệ thống học kết hợp thường tỏ ra vô cùng hữu ích khi làm việc với những bộ dữ liệu cực lớn hoặc khi ta có quá ít dữ liệu cho việc luyện mô hình. Khi bộ dữ liệu luyện quá lớn, việc luyện một mô hình có thể tận dụng được nhiều nhất lượng thông tin mà bộ dự liệu đó mang lại trở nên khó khăn. Để giải quyết vấn đề này, ta có thể chia bộ dữ liệu ban đầu một cách chiến lược thành nhiều bộ dữ liệu nhỏ hơn, rồi sau đó dùng chúng để luyện các bộ phân lớp độc lập mà sau này có thể được kết hợp với nhau

bằng một luật kết hợp phù hợp. Ngược lại, khi có quá ít dữ liệu, phương pháp lấy mẫu bootstrap có thể trở nên vô cùng hữu ích khi sử dụng để luyện nhiều mô hình khác nhau. Phương pháp lấy mẫu bootstrap là phương pháp lấy mẫu có hoàn lại (sampling with replacement), nghĩa là một cá thể có thể xuất hiện nhiều lần trong một lần lấy mẫu.

Chia để trị

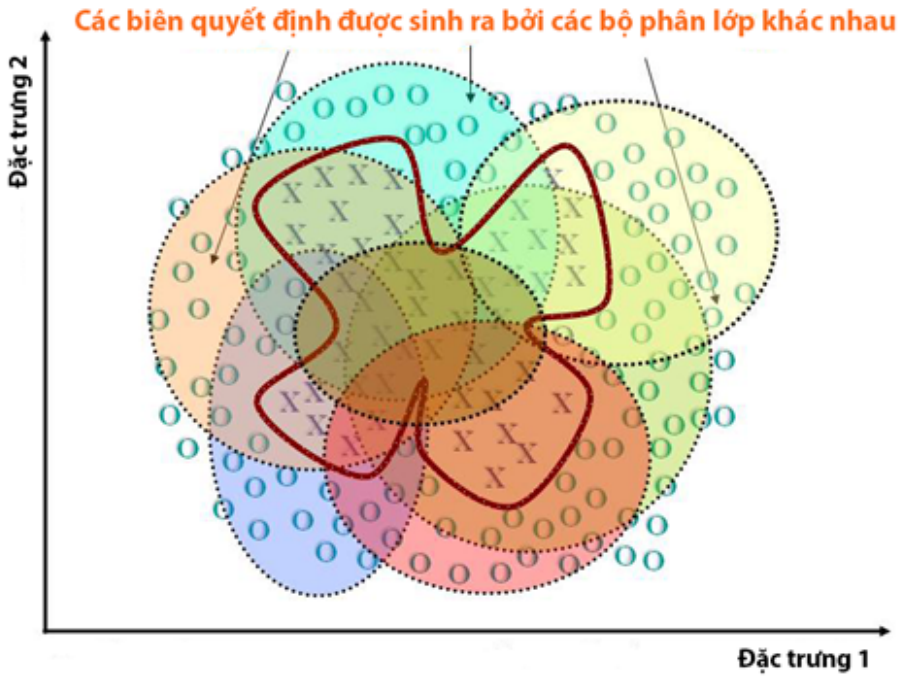
Thực tế cho thấy, trong một bài toán phân lớp cụ thể, biên quyết định được dùng để phân lớp dữ liệu có thể quá phức tạp hoặc nằm ngoài khoảng không gian mà mô hình được chọn có thể biểu diễn. Lấy ví dụ như bài toán phân lớp nhị phân trong không gian hai chiều được minh họa trong Hình 1.18, rõ ràng chỉ một bộ phân lớp tuyến tính không thể học được biên quyết định phức tạp của bài toán này. Tuy nhiên, khi kết hợp nhiều bộ phân lớp tuyến tính lại với nhau một cách phù hợp, ta có thể sử dụng mô hình kết hợp này để học biên quyết định của một bài toán phân lớp phi tuyến phức tạp bất kỳ nào đó.



Hình 1.18: Một bài toán phân lớp nhị phân trong không gian hai chiều có biên quyết định phức tạp.

Vẫn xét bài toán trên nhưng trong trường hợp này ta sử dụng một bộ phân lớp có khả năng sinh ra được biên quyết định có dạng một vòng tròn khép kín; ta thấy rằng

biên quyết định này vẫn không thể phân lớp được một cách có hiệu quả cho bài toán đang xét. Một lần nữa, thay vì chỉ sử dụng một bộ phân lớp duy nhất, ta sẽ sử dụng nhiều bộ phân lớp và kết hợp các kết quả của từng bộ phân lớp này lại theo phương pháp bỏ phiếu quá bán (Majority Voting) kết hợp các kết quả đó lại với nhau để cho ra kết quả cuối cùng. Nếu đầu ra của các bộ phân lớp là độc lập với nhau và có ít nhất một nửa trong số chúng phân lớp đúng cho một điểm dữ liệu bất kỳ nào đó, mô hình học kết hợp sử dụng các bộ phân lớp này có thể dễ dàng học được biên quyết định phức tạp của bài toán ban đầu (Hình 1.19).



Hình 1.19: Một sự kết hợp các bộ phân lớp có biên quyết định dạng vòng tròn khép kín.

Tư tưởng chung ở đây là thay vì chỉ sử dụng một bộ phân lớp để "xấp xỉ" biên quyết định của một bài toán phân lớp cụ thể nào đó, ta có thể chia không gian của bộ dữ liệu luyện thành những phần nhỏ hơn và dễ học hơn để đưa vào từng bộ phân lớp luyện cho những tập con này. Biên quyết định thực sự của bài toán phức tạp ban đầu sẽ được xấp xỉ bởi việc kết hợp các kết quả của những bộ phân lớp đó một cách thích hợp.

Sự hợp nhất dữ liệu

Trong nhiều bài toán thực tế có áp dụng máy học chẳng hạn như các ứng dụng giúp đưa ra quyết định một cách tự động từ những dữ liệu đầu vào, ta có thể cung cấp cho nó rất nhiều dữ liệu từ các nguồn khác nhau, những dữ liệu có thể chứa những thông tin cần thiết cho việc đưa ra quyết định. Việc sử dụng nhiều nguồn dữ liệu để luyện mô hình, nếu được thực hiện một cách hợp lý, có thể giúp ta đạt được những kết quả tốt hơn khi chỉ dựa vào một nguồn dữ liệu duy nhất. Ví dụ như để chẩn đoán cho một bệnh nhân bị bệnh thần kinh, các bác sĩ có thể sử dụng nhiều nguồn dữ liệu khác nhau về bệnh nhân đó như kết quả đo điện não đồ (dữ liệu dạng chuỗi thời gian một chiều), các dữ liệu thu được từ chụp cộng hưởng từ (Magnetic Resonance Imaging – MRI), hay chụp cắt lớp phát xạ positron (Positron Emission Tomography – PET) (dữ liệu không gian hai chiều), cùng với các thông tin khác của bệnh nhân như tuổi tác, giới tính, ... (các giá trị vô hướng và/hoặc các giá trị phân loại). Ta có thể đưa toàn bộ đống dữ liệu này vào để luyện cho một mô hình duy nhất; tuy nhiên, việc này thường rất khó để có thể thực hiện được. Thay vào đó, ta sẽ sử dụng nhiều bộ phân lớp khác nhau, từng bộ phân lớp sẽ được luyện trên từng bộ dữ liệu một cách độc lập. Quyết định cuối cùng sẽ là kết quả của việc kết hợp các quyết định được thực hiện bởi từng bộ phân lớp bằng một luật kết hợp nào đó.

Ước lượng độ tin cậy

Tính có cấu trúc của một hệ thống học kết hợp cho phép ta có thể gán cho từng quyết định được đưa ra bởi hệ thống đó một độ tin cậy. Giả sử ta có một bộ phân lớp được kết hợp từ nhiều bộ phân lớp cơ sở khác nhau. Khi tiến hành đưa ra quyết định cho một giá trị đầu vào cụ thể, nếu phần lớn các bộ phân lớp cơ sở đều đồng ý với một quyết định nào đó, điều đó có thể hiểu là mô hình học kết hợp của chúng ta rất tự tin vào quyết định này. Tuy nhiên, khi một nửa các bộ phân lớp cơ sở đưa ra một quyết định nào đó trong khi nửa còn lại đưa ra một quyết định khác, điều đó có nghĩa là mô hình học kết hợp của chúng ta không tự tin vào quyết định cuối cùng mà nó đưa ra. Điều đáng lưu ý ở đây đó là quyết định được đưa ra bởi một mô hình học kết hợp cho một dữ liệu đầu vào nào đó có độ tự tin cao không có nghĩa là quyết định đó là đúng và ngược lại. Tuy nhiên, theo như những quan sát từ thực tế, mô hình học kết hợp khi đưa ra một quyết định đúng thì thường có độ tự tin cao và đưa ra một quyết định sai khi có độ tự tin thấp. Theo cách tiếp cận đó, các quyết định được đưa ra bởi mô hình học kết hợp có thể được sử dụng để ước lượng các xác suất hậu nghiệm của các quyết định phân lớp.

Chương 2

Tiền xử lý dữ liệu

2.1 Ý nghĩa của tiền xử lý

Dữ liệu được xem là chất lượng nếu chúng đáp ứng các yêu cầu và mục đích sử dụng. Có nhiều các yếu tố để so sánh chất lượng của dữ liệu, ví dụ như độ chính xác, tính đầy đủ, nhất quán, kịp thời, đáng tin cậy hay khả năng diễn giải.

Hãy tưởng tượng rằng bạn là người quản lý tại một cửa hàng điện tử và được giao nhiệm vụ phân tích dữ liệu của công ty liên quan đến doanh số bán hàng của chi nhánh bạn. Bạn ngay lập tức thực hiện nhiệm vụ nà bằng cách kiểm tra cẩn thận cơ sở dữ liệu và kho dữ liệu của công ty, xác định và chọn các thuộc tính hoặc thứ nguyên (ví dụ: mặt hàng, giá và đơn vị đã bán) để đưa vào phân tích của bạn. Lúc này, có thể bạn nhận thấy rằng một số thuộc tính không được ghi lại các bộ giá trị. Chẳng hạn, bạn muốn có thông tin về việc liệu mỗi mặt hàng đã mua có được quảng cáo là đang giảm giá hay không, nhưng bạn phát hiện ra rằng thông tin này đã không được ghi lại. Hơn nữa, người dùng hệ thống cơ sở dữ liệu của bạn đã báo cáo lỗi, hoặc giá trị lưu lại là bất thường hoặc có sự không nhất quán trong dữ liệu được ghi lại đối với một số các giao dịch. Nói cách khác, dữ liệu bạn muốn phân tích bằng các kỹ thuật khai thác dữ liệu là không đầy đủ (thiếu các giá trị thuộc tính hoặc các thuộc tính nhất định được quan tâm hoặc chỉ chứa dữ liệu tổng hợp); không chính xác hoặc nhiễu (chứa lỗi hoặc giá trị sai lệch so với kỳ vọng); và không nhất quán (chứa sự khác biệt trong các mã bộ phận được sử dụng để phân loại các mặt hàng).

Ví dụ này minh họa ba trong số các yếu tố xác định chất lượng của dữ liệu: độ chính xác, tính đầy đủ và tính nhất quán. Dữ liệu không chính xác, không đầy đủ và không nhất quán là các tình trạng khá phổ biến của cơ sở dữ liệu và kho dữ liệu lớn trong thế giới thực. Có nhiều lý do có thể xảy ra đối với dữ liệu không chính xác (tức là có các giá trị thuộc tính không chính xác). Các công cụ thu thập dữ liệu được sử dụng có thể bị lỗi hoặc có thể có lỗi do con người hoặc máy tính khi nhập dữ liệu. Đôi khi, người dùng có thể cố tình gửi các giá trị dữ liệu không chính xác cho các trường bắt buộc khi họ không muốn gửi thông tin cá nhân (ví dụ: bằng cách chọn giá trị mặc định "ngày 01 tháng 01" được hiển thị cho ngày sinh). Lỗi trong quá trình truyền dữ liệu cũng có thể xảy ra, có thể có những hạn chế về công nghệ chẳng hạn như kích thước bộ đệm hạn chế để điều phối việc truyền và tiêu thụ dữ liệu đồng bộ. Dữ liệu không chính xác cũng có thể do sự mâu thuẫn trong quy ước đặt tên hoặc dữ liệu mã hoặc định dạng

không nhất quán cho các trường đầu vào (ví dụ: ngày). Các bộ giá trị trùng lặp cũng cần được làm sạch dữ liệu.

Dữ liệu không đầy đủ có thể xảy ra vì một số lý do: thuộc tính quan tâm có thể không luôn sẵn sàng, chẳng hạn như thông tin khách hàng cho dữ liệu giao dịch bán hàng; dữ liệu khác có thể không được đưa vào đơn giản vì chúng không được coi là quan trọng vào thời điểm đó của mục nhập'; dữ liệu liên quan có thể không được ghi lại do hiểu nhầm hoặc do trục trặc thiết bị; dữ liệu không nhất quán với dữ liệu đã ghi khác có thể đã bị xóa. Hơn nữa, việc ghi lại lịch sử dữ liệu hoặc các sửa đổi có thể đã bị bỏ qua. Thiếu dữ liệu, đặc biệt đối với các bộ dữ liệu bị thiếu đối với một số thuộc tính, có thể cần được tìm ra.

Chất lượng dữ liệu phụ thuộc vào mục đích sử dụng dữ liệu. Hai người dùng khác nhau có thể có những đánh giá rất khác nhau về chất lượng của một cơ sở dữ liệu nhất định. Ví dụ, một nhà phân tích tiếp thị có thể cần truy cập vào cơ sở dữ liệu được đề cập trước đó để biết danh sách địa chỉ khách hàng. Một số địa chỉ đã lỗi thời hoặc không chính xác, nhưng nhìn chung, 80% địa chỉ là chính xác. Nhà phân tích tiếp thị coi đây là một cơ sở dữ liệu khách hàng lớn cho các mục đích tiếp thị mục tiêu và hài lòng với độ chính xác của cơ sở dữ liệu, trong khi nếu là giám đốc bán hàng, bạn sẽ thấy dữ liệu không chính xác.

Tính kịp thời cũng ảnh hưởng đến chất lượng dữ liệu. Giả sử rằng bạn đang giám sát việc phân phối tiền thưởng doanh số hàng tháng cho các đại diện bán hàng hàng đầu tại cửa hàng điện máy của bạn. Tuy nhiên, đại diện bán hàng không nộp hồ sơ bán hàng của họ đúng hạn vào cuối tháng. Ngoài ra còn có một số hiệu chỉnh và điều chỉnh sau cuối tháng. Trong một khoảng thời gian sau mỗi tháng, dữ liệu được lưu trữ trong cơ sở dữ liệu không đầy đủ. Tuy nhiên, khi tất cả dữ liệu được nhận, nó là chính xác. Thực tế việc dữ liệu cuối tháng không được cập nhật kịp thời có tác động tiêu cực đến chất lượng dữ liệu.

Hai yếu tố khác ảnh hưởng đến chất lượng dữ liệu là độ tin cậy và khả năng diễn giải. Độ tin cậy phản ánh mức độ tin cậy của dữ liệu bởi người dùng, trong khi khả năng diễn giải phản ánh dữ liệu được hiểu dễ dàng như thế nào. Giả sử rằng một cơ sở dữ liệu, tại một thời điểm, có một số lỗi, tất cả đều đã được sửa chữa. Tuy nhiên, các lỗi trong quá khứ đã gây ra nhiều vấn đề cho người dùng bộ phận bán hàng, và do đó họ không còn tin tưởng vào dữ liệu. Các dữ liệu cũng sử dụng nhiều mã kế toán mà phòng kinh doanh không biết cách thông dịch. Mặc dù cơ sở dữ liệu hiện đã chính xác, đầy đủ, nhất quán và kịp thời, người dùng bộ phận bán hàng có thể coi nó là chất lượng thấp do độ tin cậy kém và khả năng diễn giải không tốt.

Vì vậy, trong quy trình khai phá dữ liệu, công việc tiền xử lý dữ liệu trước khi đưa vào các mô hình là rất cần thiết, bước này làm cho dữ liệu có được ban đầu qua thu thập dữ liệu (gọi là dữ liệu gốc) có thể áp dụng được một cách phù hợp với các mô hình khai phá dữ liệu cụ thể.

2.2 Làm sạch dữ liệu

2.2.1 Dữ liệu mất mát

Khi phân tích dữ liệu thường xảy ra trường hợp nhiều bộ dữ liệu không có giá trị được ghi lại cho một số thuộc tính do nguyên nhân chủ quan như tác nhân con người hay nguyên nhân khách quan như sai sót trong quá trình nhập dữ liệu. Khi đó, chúng ta có thể sử dụng các phương pháp sau để giải quyết vấn đề mất mát dữ liệu:

1. Xóa đi đối tượng bị thiếu giá trị: Điều này thường được thực hiện khi thiếu nhãn của lớp (giả sử nhiệm vụ khai thác liên quan đến phân loại). Phương pháp này không hiệu quả lắm, trừ khi đối tượng chỉ chứa một số thuộc tính bị thiếu giá trị. Nó đặc biệt kém khi tỷ lệ giá trị bị thiếu trên mỗi thuộc tính thay đổi đáng kể. Bằng cách bỏ đi các giá trị này, chúng ta đồng thời không sử dụng các giá trị còn lại của thuộc tính đó. Dữ liệu như vậy có thể có hữu ích cho nhiệm vụ trước mắt.

2. Lấp đầy giá trị còn thiếu theo cách thủ công: Nói chung, cách tiếp cận này tốn thời gian và có thể không khả thi với một tập dữ liệu lớn với nhiều giá trị bị thiếu.

3. Sử dụng một hằng số chung để điền vào giá trị bị thiếu: Thay thế tất cả các giá trị thuộc tính bị thiếu bởi cùng một hằng số, chẳng hạn như nhãn "Không xác định" hoặc -∞. Nếu các giá trị bị thiếu được thay thế bằng "Không xác định", thì chương trình khai thác có thể nhầm tưởng rằng chúng tạo thành một khái niệm mới, vì chúng đều có một giá trị chung - đó là "Không xác định." Do đó, mặc dù phương pháp này đơn giản, nhưng nó không phải là hoàn hảo.

4. Sử dụng độ đo giá trị trung tâm của thuộc tính cho giá trị bị thiếu (ví dụ: giá trị trung bình hoặc trung vị): giá trị này là giá trị "ở giữa" của phân phối dữ liệu. Đối với phân phối dữ liệu bình thường (đối xứng), giá trị trung bình có thể được sử dụng, trong khi phân phối dữ liệu lệch nên sử dụng dải phân cách. Ví dụ: giả sử rằng phân phối dữ liệu liên quan đến thu nhập của khách hàng của một cửa hàng là đối xứng và thu nhập trung bình là $ 56,000, ta sẽ sử dụng giá trị này để thay thế giá trị còn thiếu cho thu nhập của khách hàng nào bị thiếu.

5. Sử dụng giá trị trung bình của thuộc tính hoặc trung vị cho tất cả các mẫu thuộc cùng một lớp với đối tượng đã cho: Ví dụ: nếu phân loại khách hàng theo rủi ro tín dụng, chúng ta có thể thay thế giá trị còn thiếu bằng giá trị thu nhập trung bình cho khách hàng trong cùng một loại rủi ro tín dụng như của nhóm đã cho. Nếu phân phối dữ liệu cho một lớp nhất định bị lệch, giá trị trung vị là lựa chọn tốt hơn.

6. Sử dụng giá trị đúng nhất có thể để điền vào giá trị còn thiếu: Giá trị này có thể được xác định với hồi quy, các công cụ dựa trên suy luận sử dụng hình thức Bayes, hoặc quy nạp cây quyết định. Ví dụ: bằng cách sử dụng các thuộc tính khách hàng khác trong tập dữ liệu của mình, bạn có thể xây dựng một cây quyết định để dự đoán các giá trị còn thiếu cho thu nhập.

2.2.2 Lỗi và phần tử ngoại lai

a, Lỗi dữ liệu

Lỗi (hay còn gọi là nhiễu) là một sai số ngẫu nhiên hoặc lỗi ngẫu nhiên đối với giá trị của một thuộc tính. Các giá trị của thuộc tính có thể bị lỗi vì lỗi của các thiết bị thu thập dữ liệu, lỗi khi nhập dữ liệu hay lỗi trong quá trình truyền dữ liệu. Chúng ta có một vài phương pháp làm mịn dữ liệu như sau:

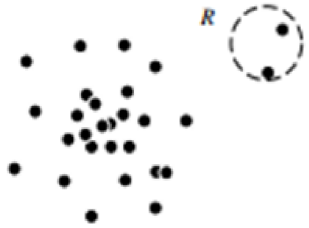
- Phân khoảng (Binning): Sắp xếp dữ liệu và phân chia thành các khoảng (bins) có tần số xuất hiện giá trị như nhau. Sau đó, mỗi khoảng dữ liệu có thể được biểu diễn bằng trung bình, trung vị hoặc các giới hạn của các giá trị trong khoảng đó.
- Hồi quy (Regression): gán dữ luệu với một hàm hồi quy
- Phân cụm (Clustering): mục đích để phát hiện và loại bỏ các ngoại lai sau khi đã xác định các cụm

b, Phần tử ngoại lai

1. Định nghĩa

   Giả sử rằng một quy trình thống kê nhất định được sử dụng để tạo ra một tập hợp các đối tượng dữ liệu. Một ngoại lệ là một đối tượng dữ liệu làm lệch đáng kể so với phần còn lại của các đối tượng, như thể nó được tạo ra bởi một cơ chế khác. Để dễ trình bày trong chương này, chúng tôi có thể đề cập đến các đối tượng dữ liệu không phải là ngoại lệ, dữ liệu bình thường hay dữ liệu dự kiến. Tương tự như vậy, chúng ta có thể gọi các ngoại lệ là dữ liệu bất thường.

   Ví dụ trong hình 2.1, hầu hết các đối tượng tuân theo phân phối Gaussian. Tuy nhiên, các đối tượng trong khu vực R có ý nghĩa khác nhau. Không chắc là chúng tuân theo phân phối giống như các đối tượng khác trong tập dữ liệu. Do đó, các đối tượng trong R là các ngoại lệ trong tập dữ liệu.



Hình 2.1: Minh họa phần tử ngoại lai

Dữ liệu ngoại lai khác với dữ liệu nhiễu, nhiễu là lỗi hoặc phương sai trong biến do. Ví dụ: một khách hàng có thể tạo ra một số giao dịch nhiễu hoặc sai lệch ví dụ như một bữa trưa lớn hơn một ngày hoặc uống thêm một ly caffe so với bình thường, các giao dịch như vậy chúng ta không nên coi là giao dịch bất thường, điều này sẽ ảnh hưởng rất nhiều vì nếu cảnh báo như vậy thì khách hàng sẽ cảm

thấy phiền toái và công ty có thể mất đi khách hàng về những báo động sai lệch. Vì vậy việc loại bỏ nhiễu trước khi phát hiện điểm ngoại lai là vô cùng cần thiết. Các phần tử ngoại lai thường được tạo ra do các cơ chế không giống như phần còn lại của dữ liệu. Vì thế trong việc phát hiện phần tử ngoại lai, điều quan trọng là chứng minh được tại sao chúng được tạo ra bởi các cơ chế khác. Để thực hiện điều này chúng ta phải đưa ra các giả định khác nhau trên phần còn lại của dữ liệu và cho thấy các phần tử ngoại lai được phát hiện vi phạm các giả định đó một cách đáng kể.

Phát hiện phần tử ngoại lai cũng liên quan đến phát hiện mới trong các bộ dữ liệu đang phát triển. Ví dụ: bằng cách giám sát một trang web truyền thông xã hội nơi có nội dung mới, phát hiện tính mới có thể xác định kịp thời các chủ đề và xu hướng mới. Chủ đề ban đầu có thể xuất hiện như ngoại lệ. Trong phạm vi này, phát hiện ngoại lai và phát hiện tính mới có một số điểm tương đồng trong phương pháp mô hình hóa và phát hiện. Tuy nhiên, một điểm khác biệt quan trọng giữa hai loại đó là trong phát hiện tính mới, một khi các chủ đề mới được xác định, chúng thường được đưa vào mô hình hành vi thông thường để các trường hợp theo dõi không còn được coi là ngoại lệ nữa.

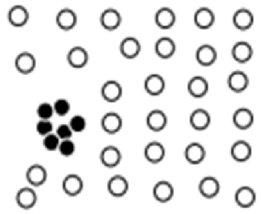
2. Phân loại phần tử ngoại lai Nói chung, các phần tử ngoại lai có thể được phân loại thành ba loại, đó là các ngoại lai toàn cục (Global Outliers), các ngoại lai theo ngữ cảnh/ có điều kiện (Contextual/ Conditional Outliers) và các ngoại lai tập thể (Collective Outliers):

   (a) Ngoại lai toàn cục (Global Outliers): Trong một tập dữ liệu nhất định, một đối tượng dữ liệu là một ngoại lai toàn cục nếu nó lệch đáng kể so với phần còn lại của tập dữ liệu. Các ngoại lai toàn cục đôi khi được gọi là điểm dị thường và là loại phần tử ngoại lai đơn giản nhất. Hầu hết các phương pháp phát hiện ngoại lai đều nhằm mục đích phát hiện các ngoại lai toàn cục.

   (b) Ngoại lai theo ngữ cảnh/ có điều kiện (Contextual/ Conditional Outliers): Trong một tập dữ liệu nhất định, một đối tượng dữ liệu là một ngoại lai theo ngữ cảnh nếu nó làm sai lệch đáng kể đối với bối cảnh cụ thể của đối tượng. Các ngoại lai theo ngữ cảnh còn được gọi là các ngoại lai có điều kiện vì chúng có điều kiện trên bối cảnh đã chọn. Do đó, trong phát hiện ngoại lai loại này, ngữ cảnh phải được xác định cụ thể như là một phần của vấn đề.

   Không giống như phát hiện ngoại lai toàn cục, trong phát hiện ngoại lai theo ngữ cảnh, việc một đối tượng dữ liệu có phải là ngoại lai hay không phụ thuộc vào không chỉ các thuộc tính hành vi mà còn cả các thuộc tính theo ngữ cảnh. Sự kết hợp của các giá trị thuộc tính hành vi có thể được coi là ngoại lai trong một bối cảnh (ví dụ: 28C là ngoại lệ cho mùa đông), nhưng không phải là ngoại lệ trong bối cảnh khác (ví dụ: 28C không phải là ngoại lệ cho mùa hè). Phát hiện ngoại lai toàn cục có thể được coi là một trường hợp đặc biệt của phát hiện ngoại cảnh theo ngữ cảnh, trong đó tập hợp các thuộc tính theo ngữ cảnh trống. Nói cách khác, phát hiện ngoại lai toàn cầu sử dụng toàn bộ tập dữ liệu làm bối cảnh.

(c) Ngoại lai tập thể (Collective Outliers): Trong hình 2.2, các đối tượng màu đen nói chung tạo thành một tập các phần tử ngoại lai vì mật độ của các đối tượng đó cao hơn nhiều so với phần còn lại trong tập dữ liệu. Tuy nhiên, mỗi đối tượng màu đen riêng lẻ không phải là một ngoại lai đối với toàn bộ tập dữ liệu.



Hình 2.2: Minh họa các phần tử ngoại lai tập thể

Phát hiện ngoại lai tập thể có nhiều ứng dụng quan trọng. Ví dụ, trong phát hiện xâm nhập, gói từ chối dịch vụ từ máy tính này sang máy tính khác được coi là bình thường và hoàn toàn không phải là một ngoại lệ. Tuy nhiên, nếu một số máy tính tiếp tục gửi các gói từ chối dịch vụ cho nhau, thì toàn bộ chúng nên được coi là một ngoại lệ phổ biến. Các máy tính liên quan có thể bị nghi ngờ là bị xâm phạm bởi một cuộc tấn công. Một ví dụ khác, giao dịch chứng khoán giữa hai bên được coi là bình thường. Tuy nhiên, một tập hợp lớn các giao dịch của cùng một cổ phiếu giữa một bên nhỏ trong một thời gian ngắn là các ngoại lệ tập thể vì chúng có thể là bằng chứng của một số người thao túng thị trường.

3. Phương pháp phát hiện phần tử ngoại lai

   Có nhiều phương pháp phát hiện ngoại lệ trong tài liệu và trong thực tế:

   - Các phương pháp giám sát, không giám sát và bán giám sát
   - Các phương pháp thống kê, các phương pháp dựa trên lân cận và các phương pháp phân cụm.

2.3 Biến đổi dữ liệu

2.3.1 Trích chọn đặc trưng

Trích chọn đặc trưng là phương pháp lựa chọn tập hợp con m đặc trưng từ M đặc trưng gốc ban đầu. Có 2 tiêu chí để trích chọn đặc trưng đó là trích chọn đặc trưng rời rạc và trích chọn vector đặc trưng.

Trích chọn đặc trưng rời rạc

Trong trường hợp này, các đặc trưng được xử lý riêng lẻ. Trước tiên ta cần lựa chọn độ đo phù hợp với bài toán thực tế, ví dụ như đối với bài toán phân lớp Có thể áp dụng

tiêu chí đo lường khả năng phân tách lớp như ROC, FDR, phân kỳ, ... Giá trị của độ đo $C(k)$ được tính cho từng đặc trưng $k, k = 1, 2..M$. Các đặc trưng sau đó được sắp xếp theo thứ tự giá trị tốt giảm dần của $C(k)$. Sau đó, ta có thể sử dụng m đặc trưng đầu tiên mang giá trị $C(k)$ tốt nhất để tạo thành vector đặc trưng. Phương pháp này có thể đơn giản về mặt chi phí tính toán, tuy nhiên bỏ qua mối tương quan giữa các đặc trưng và gặp phải trường hợp m đặc trưng được lựa chọn có mối tương quan lớn hơn so với 1 hoặc nhiều các đặc trưng trong $M - m$ các đặc trưng không được chọn khác. Do vậy, ta có thể thực hiện việc lựa chọn dựa vào đồng thời giá trị $C(k)$ và hệ số tương quan p_{ij}:

$$p_{ij} = \frac{\sum_{n=1}^{N} x_{ni} x_{nj}}{\sqrt{\sum_{n=1}^{N} x_{ni} \sum_{n=1}^{N} x_{nj}}}. \tag{1.3.1-1}$$

trong đó x_{nk} là giá trị của đặc trưng thứ k ở bản ghi n trong tổng N bản ghi.

Thủ tục trích chọn đặc trưng có thể được thực hiện như sau:

- Lựa chọn tiêu chí độ đo C và tính giá trị độ đo $C(k)$ cho tất cả các đặc trưng $k, k = 1, 2, ..M$, sắp xếp theo tính giảm dần và chọn ra đặc trưng có giá trị độ đo C tốt nhất, giả thiết đó là x_{i_1}.

- Để lựa chọn đặc trưng thứ hai, ta tính hệ số tương quan p_{ij} giữa đặc trưng i_1 và $M - 1$ đặc trưng còn lại, tức là $p_{i_1 j}, j \neq i_1$.

- Lựa chọn đặc trưng x_{i_2} sao cho:

$$i_2 = \arg \max_j \{\alpha_1 C(j) - \alpha_2 |p_{i_1 j}|\}. \tag{1.3.1-2}$$

  với α_1 và α_2 là hệ số đánh giá mức độ quan trọng theo 2 tiêu chí là khả năng thực hiện nhiệm vụ và mức độ tương quan với đặc trưng đã được lựa chọn.

- Tương tự, ta lựa chọn $x_{i_k}, k = 3, ..m$ sao cho:

$$i_k = \arg \max_j \{\alpha_1 C(j) - \frac{\alpha_2}{k-1} \sum_{r=1}^{k-1} |p_{i_r j}|\}. \tag{1.3.1-3}$$

  với $j \neq i_r, r = 1, 2...k - 1$

Trích chọn vector đặc trưng

Việc xử lý các đặc trưng riêng lẻ, tức là các đặc trưng vô hướng, có ưu điểm là tính đơn giản trong tính toán nhưng có thể không hiệu quả đối với các vấn đề phức tạp và đối với các đối tượng có tính tương quan cao. Bây giờ chúng ta sẽ tập trung vào các kỹ thuật đo lường khả năng thực hiện nhiệm vụ, ví dụ như nhiệm vụ phân loại của các vector đặc trưng. Ta dễ dàng thấy được chi phí tính toán là yếu tố hạn chế chính của cách tiếp cận như vậy. Lấy ví dụ ta có $M = 20, m = 5$, số lượng các cách để trích chọn vector đặc trưng là 15504 cách. Và đôi khi, quyết định lựa chọn không dựa trên độ đo

C mà dựa trên chính hiệu suất thực hiện nhiệm vụ phân loại, nghĩa là đối với mỗi tổ hợp vector đặc trưng, hiệu suất và tỉ lệ lỗi phải được tính toán và tăng tính phức tạp cho quá trình trích chọn đặc trưng. Để giảm độ phức tạp, một số kĩ thuật tìm kiếm hiệu quả đã được đề xuất.

Tìm kiếm ngược tuần tự

Ta sẽ lấy ví dụ về phương pháp này. Cho $M = 4$, $m = 2$, và các đặc trưng ban đầu được kí hiệu là $[x_1, x_2, x_3, x_4]$. Quy trình lựa chọn bao gồm các bước sau:

- Tính toán giá trị độ đo $C(k)$ cho 4 đặc trưng.
- Loại bỏ một đặc trưng ra khỏi tổ hợp ban đầu, ta có các trường hợp $[x_1, x_2, x_3]$, $[x_1, x_2, x_4]$, $[x_1,$ Lựa chọn tổ hợp tốt nhất, lấy ví dụ là $[x_1, x_2, x_3]$.
- Từ tổ hợp 3 đặc trưng trên, thực hiện tương tự và loại bỏ một đặc trưng để được tổ hợp 2 đặc trưng.

Như vậy, bắt đầu từ M đặc trưng, ở mỗi vòng lặp thực hiện loại bỏ 1 đặc trưng cho đến khi thu được m đặc trưng tốt nhất. Rõ ràng đây là một thủ tục tìm kiếm dưới mức tối ưu, vì không đảm bảo rằng vector 2 chiều tối ưu phải bắt nguồn từ vector 3 chiều tối ưu. Số lượng tìm kiếm tổ hợp qua phương pháp này là $1+1/2((M+1)M-m(m+1))$, nhỏ hơn nhiều lần so với việc tìm kiếm toàn bộ.

Tìm kiếm tuần tự

Ngược lại so với lựa chọn ngược tuần tự, ta có quy trình thực hiện đối với ví dụ ở trên:

- Tính toán giá trị độ đo $C(k)$ cho 4 đặc trưng, lựa chọn đặc trưng có độ đo tốt nhất, chẳng hạn là x_1.
- Xây dựng các tổ hợp 2 đặc trưng có thể có mà bao gồm đặc trưng x_1 là $[x_1, x_2], [x_1, x_3], [x_1, x_4]$.
- Tính giá trị độ đo $C(k)$ cho các tổ hợp và lựa chọn tổ hợp tốt nhất, ví dụ là $[x_1, x_2]$.

Số lượng tìm kiếm tổ hợp qua phương pháp này là $Mm - m(m-1)/2$. Như vậy, theo quan điểm tính toán, kĩ thuật tìm kiếm ngược tuần tự hiệu quả hơn kĩ thuật tìm kiếm tuần tự.

Tìm kiếm nổi

Hai phương pháp trên đều gặp phải hạn chế là "hiệu ứng làm tổ". Tức là, khi một đặc trưng bị loại bỏ trong phương pháp tìm kiếm ngược tuần tự, sẽ không có khả năng nó được xem xét lại lần nữa. Điều ngược lại là đúng đối với thủ tục tìm kiếm tuần tự: một khi đặc trưng được chọn, không có cách nào để nó bị loại bỏ sau này. Phương pháp tìm kiếm nổi sẽ khắc phục hạn chế này khi xem xét các đặc trưng đã bị loại bỏ trước đó và đồng thời có thể loại bỏ các đặc trưng đã chọn trước đó.

Gọi $X_k = x_1, x_2, ..x_k$ là tập hợp k đặc trưng tốt nhất và Y_{m-k} là tập hợp $m-k$ đặc trưng còn lại. Đồng thời ta cũng lưu lại tất cả các tập hợp con tốt nhất có kích thước

thấp hơn như $X_2, X_3, ...X_{k-1}$. Cơ sở lý luận chính của phương pháp này được tóm tắt như sau: Ở bước tiếp theo, k + 1 tập hợp con tốt nhất X_{k+1} được hình thành bằng cách "mượn" một phần tử từ Y_{m-k}. Sau đó, quay lại tập hợp con thứ nguyên thấp hơn đã chọn trước đó 1 đơn vị để kiểm tra xem việc bao gồm phần tử mới này có cải thiện tiêu chí độ đo C hay không. Nếu có, phần tử mới sẽ thay thế một trong các đặc trưng đã chọn trước đó. Và nếu không, ta thu được X_{k+1} và tiếp tục vòng lặp. Các bước của thuật toán được mô tả như sau:

- Bước 1 (Bổ sung): $x_{k+1} = argmax_{y \in Y_{m-k}} C(X_k, y)$, tức là lựa chọn phần tử từ Y_{m-k} sao cho khi phần tử này kết hợp với X_k ta thu được tổ hợp $X_{k+1} = X_k, x_{k+1}$ tốt nhất.
- Bước 2 (Kiểm định):
  - $x_r = argmax_{y \in X_{k+1}} C(X_{k+1} - y)$, tức là chọn ra đặc trưng có ít tác dụng nhất trong việc cải thiện tiêu chí độ đo khi loại bỏ nó ra khỏi X_{k+1}.
  - Nếu $r = k + 1$, đặt $k = k + 1$ và quay lại bước 1.
  - Nếu $r \neq k + 1$ và $C(X_{k+1} - x_r) < C(X_k)$ quay lại bước 1, tức là nếu xóa bỏ x_r ra khỏi tổ hợp sẽ không cải thiện so với tổ hợp k tốt nhất trước đó, không có tìm kiếm ngược tuần tự nào được thực hiện.
  - Nếu $k = 2$ đặt $X_k = X_{k+1} - x_r$ và $C(X_k) = C(X_{k+1} - x_r)$ và quay lại bước 1.
- Bước 3 (Loại bỏ):
  - Xóa x_r, tức là $X'_k = X_{k+1} - x_r$.
  - $x_s = argmax_{y \in X'_k} C(X'_k - y)$, tức là tìm kiếm đặc trưng ít quan trọng nhất trong tổ hợp mới.
  - Nếu $C(X'_k - x_s) < C(X_{k-1})$ thì $X_k = X'_k$ và quay lại bước 1. Thủ tục tìm kiếm ngược tuần tự sẽ không được thực hiện.
  - Đặt $X'_{k-1} = X'_k - x_s$ và $k = k - 1$.
  - Nếu $k = 2$ đặt $X_k = X'_k$ và $C(X_k) = C(X'_k)$ và quay lại bước 1.
  - Quay lại bước 3.

Thuật toán được khởi tạo bằng cách thực thi thuật toán tìm kiếm tuần tự để tạo thành X_2 và kết thúc khi m đặc trưng đã được lựa chọn. Mặc dù thuật toán không đảm bảo tìm thấy tất cả các tập hợp con đặc trưng tốt nhất, nhưng nó dẫn đến hiệu suất được cải thiện đáng kể so với phương pháp tuần tự và ngược tuần tự, với chi phí là tăng độ phức tạp.

2.3.2 Chuẩn hóa dữ liệu

Việc xử lý các thuộc tính nhận giá trị ở các thang khác nhau có thể dẫn đến hiện tượng lệch, thiên vị (bias) về các thuộc tính ở dải giá trị cao, gây ra mất mát thông

tin của các thuộc tính ở dải giá trị thấp. Nói một cách đơn giản, khi có nhiều thuộc tính nhưng các thuộc tính có giá trị ở các thang, dải khác nhau có thể dẫn đến mô hình mô hình hóa dữ liệu kém khi thực hiện các hoạt động khai thác dữ liệu. Do vậy, cần phải được chuẩn hóa để mang tất cả các thuộc tính về cùng một thang giá trị. Trong hầu hết các trường hợp, việc chuẩn hóa là bắt buộc trước khi thực sự tiến hành xử lý các thuộc tính. Chuẩn hóa được sử dụng để chia tỷ lệ dữ liệu của một hoặc một số thuộc tính sao cho cùng nằm trong một phạm vi mà thông thường hay sử dụng như $[-1.0, 1.0], [0.0, 1.0]$. Có 3 phương pháp chuẩn hóa dữ liệu hay được sử dụng đó là chuẩn hóa tỷ lệ thập phân, chuẩn hóa min-max và chuẩn hóa z-score.

Chuẩn hóa tỷ lệ thập phân

Phương pháp chuẩn hóa thực hiện bằng cách di chuyển dấu thập phân các giá trị của dữ liệu. Để chuẩn hóa dữ liệu bằng kỹ thuật này, ta chia mỗi giá trị của dữ liệu cho giá trị tuyệt đối lớn nhất (1.3.2-1), hoặc cho số 10^j nhỏ nhất không bé hơn giá trị tuyệt đối lớn nhất đó (1.3.2-2). Dễ thấy rằng biên của thang sau phép biến đổi này là $[0.0, 1.0]$.

$$v_{max} = max(|v_i|)$$

$$v'_i = \frac{v_i}{v_{max}}. \tag{1.3.2-1}$$

$$v'_i = \frac{v_i}{10^j}. \tag{1.3.2-1}$$

trong đó j là số nguyên nhỏ nhất sao cho $v_{max} < 10^j$.

Ví dụ: cho dữ liệu đầu vào là: -10, 201, 301, -401, 501, 601, 701. Để chuẩn hóa dữ liệu trên:

- Bước 1: Giá trị tuyệt đối lớn nhất trong dữ liệu đã cho $v_{max} = 701$
- Bước 2: Chia dữ liệu đã cho cho 1000 (tức là $j = 3$)
- Kết quả: Dữ liệu chuẩn hóa lần lượt là: -0.010, 0.201, 0.301, -0.401, 0.501, 0.601, 0.701.

Chuẩn hóa min-max

Trong kỹ thuật chuẩn hóa dữ liệu này, phép biến đổi tuyến tính được thực hiện trên dữ liệu gốc. Mỗi giá trị v_i trong dữ liệu sử dụng giá trị lớn nhất v_{max} và nhỏ nhất v_{min} của dữ liệu ban đầu để biến đổi về giá trị v'_i thông qua công thức như sau:

$$v'_i = \frac{v_i - v_{min}}{v_{max} - v_{min}}(new\_max - new\_min) + new\_min. \tag{1.3.2-3}$$

trong đó new_min, new_max lần lượt là giá trị tối thiểu và giá trị tối đa của dải giá trị mới, hay còn gọi là biên của phạm vi. Thông thường ta chọn $(new\_min, new\_max) = (0.0, 1.0)$.

Ví dụ:

- Cho dữ liệu đầu vào là: -10, 201, 301, -401, 501, 601, 701.
- Chọn $(new\_min, new\_max) = (0.0, 1.0)$.
- Kết quả lần lượt là: 0.355, 0,546, 0,637, 0.000, 0.819, 0.902, 1.000.

Chuẩn hóa z-score

Trong kỹ thuật này, các giá trị v_i được chuẩn hóa về giá trị v_i' dựa trên giá trị trung bình μ_A và độ lệch chuẩn σ_A của dữ liệu A. Công thức được sử dụng là:

$$v_i' = \frac{v_i - \mu_A}{\sigma_A}. \tag{1.3.2-4}$$

Ví dụ:

- Cho dữ liệu đầu vào là: -10, 201, 301, -401, 501, 601, 701.
- Giá trị trung bình: $\mu_A = 270.571$.
- Độ lệch chuẩn: $\sigma_A = 355.174$.
- Kết quả lần lượt là: -0.790, -0.196, 0.086, -1.891, 0.649, 0.930, 1.211.

2.3.3 Rời rạc hóa

Rời rạc hóa là phương pháp biến đổi dữ liệu thô sang dữ liệu trong miền giá trị rời rạc hữu hạn. Trong đó, dữ liệu thô ban đầu có thể nhận giá trị trong miền liên tục (kích thước, khối lượng, tiền tệ, ...) hoặc miền rời rạc (tuổi tác, số lượng, ...). Rời rạc hóa thông thường được sử dụng trong nhiệm vụ giảm số chiều dữ liệu và làm mịn dữ liệu khi thực hiện việc nhóm các miền giá trị có cùng một thuộc tính nào đó cùng nhận một giá trị, ví dụ như việc nhóm số tuổi vào các thuộc tính trẻ $(1-18)$, trưởng thành $(18-35)$, trung niên $(35-60)$, già (>60). Sau đây ta sẽ trình bày 3 phương pháp hay sử dụng để rời rạc hóa dữ liệu: Phương pháp nhóm theo bin, phương pháp phân tích biểu đồ và phương pháp rời rạc hóa dựa trên phân cụm, cây quyết định và phân tích tương quan.

Phương pháp nhóm theo bin

Hình 1.3-1: Hệ thống phân cấp giá tiền dựa trên nhóm theo bin []

Phương pháp nhóm theo bin là một kỹ thuật chia nhỏ từ trên xuống dựa trên một số lượng các bin xác định, thông qua đó tạo ra một hệ thống phân cấp (ví dụ hình 1.3-1). Dữ liệu thông qua hệ thống phân cấp sẽ được rời rạc hóa và đồng thời cũng được giảm số chiều. Miền giá trị của các thuộc tính được rời rạc hóa bằng cách áp dụng cách xếp bin có độ rộng bằng nhau hoặc tần số bằng nhau, sau đó thay thế từng giá trị bin bằng giá trị trung bình, trung vị hoặc biên của bin. Các kỹ thuật này có thể được áp dụng đệ quy vào các phân vùng kết quả để tạo ra các cấu trúc phân cấp. Ta có ví dụ sau minh họa việc nhóm theo bin:

- Dữ liệu đầu vào (giá tiền): 4, 8, 15, 21, 21, 24, 25, 28, 34.
- Phân vào các bin có kích thước bằng nhau:
  - Bin 1: 4, 8, 15.
  - Bin 2: 21, 21, 24.
  - Bin 3: 25, 28, 34.
- Thay thế bằng giá trị trung bình:
  - Bin 1: 9, 9, 9.
  - Bin 2: 22, 22, 22.
  - Bin 3: 29, 29, 29.
- Thay thế bằng giá trị biên:
  - Bin 1: 4, 4, 15.
  - Bin 2: 21, 21, 24.
  - Bin 3: 25, 25, 34.

Ưu điểm của phương pháp nhóm theo bin đó là dễ dàng thực hiện, không sử dụng thông tin lớp và do đó là một kỹ thuật rời rạc hóa không giám sát. Tuy nhiên nhược điểm của nó đó là phải định nghĩa trước số lượng các bin và kích thước tương ứng. Việc định nghĩa như vậy sẽ gây ra tác động quan trọng tới kết quả và thông thường đòi hỏi phải có kiến thức nhất định về thuộc tính của dữ liệu, mục đích của bài toán cũng như quá trình thử và sai lâu dài nhằm đạt được kết quả tốt nhất.

Phương pháp phân tích biểu đồ

Hình 1.3-2: Ví dụ biểu đồ giá tiền []

Hình 1.3-3: Ví dụ phân thành các vùng có kích thước bằng nhau []

Giống như phương pháp nhóm theo bin, phân tích biểu đồ là một kỹ thuật rời rạc hóa không giám sát bởi vì không sử dụng thông tin lớp. Thông qua biểu đồ ta tiến hành việc phân đoạn các giá trị của một thuộc tính thành các phạm vi riêng biệt được gọi là phân vùng, nhóm hoặc bin.

Các quy tắc phân đoạn khác nhau có thể được sử dụng để xác định biểu đồ. Ví dụ: trong biểu đồ có chiều rộng bằng nhau, các giá trị được phân chia thành các phân vùng hoặc phạm vi có kích thước bằng nhau (ví dụ hình 1.3-2, 1.3-3 thực hiện việc phân vùng biểu đồ giá tiền vào các phân vùng có kích thước bằng nhau).

Với biểu đồ tần số bằng nhau, các giá trị được phân vùng sao cho lý tưởng nhất là mỗi phân vùng chứa cùng một số bộ dữ liệu. Thuật toán phân tích biểu đồ có thể được áp dụng đệ quy cho mỗi phân vùng để tự động tạo ra một hệ thống phân cấp đa

cấp, với thủ tục kết thúc khi đạt đến một số lượng các cấp được xác định trước. Kích thước tối thiểu cũng có thể được áp dụng cho mỗi cấp nhằm điều khiển quy trình đệ quy, bằng việc chỉ định chiều rộng tối thiểu của một phân vùng hoặc số lượng giá trị tối thiểu cho mỗi phân vùng ở mỗi cấp. Ngoài ra biểu đồ cũng có thể được phân vùng dựa trên phân tích phân cụm của phân phối dữ liệu, như được mô tả sau đây.

Phương pháp rời rạc hóa dựa trên phân cụm, cây quyết định và phân tích tương quan

Phân tích phân cụm, phân tích cây quyết định và phân tích tương quan có thể được sử dụng để rời rạc hóa dữ liệu.

Phân tích phân cụm là một phương pháp phân tích dữ liệu phổ biến. Một thuật toán phân cụm có thể được áp dụng để rời rạc một thuộc tính nào đó bằng cách phân chia các giá trị thành các cụm hoặc nhóm. Phân cụm xem xét sự phân bố các giá trị của thuộc tính, cũng như mức độ gần nhau của các điểm dữ liệu, và do đó có thể tạo ra kết quả rời rạc hóa chất lượng cao. Phân cụm có thể được sử dụng để tạo ra hệ thống phân cấp cho thuộc tính bằng cách tuân theo chiến lược tách top-down hoặc chiến lược hợp nhất bottom-up, trong đó mỗi cụm tạo thành một nút của hệ thống. Cuối cùng, mỗi cụm hoặc phân vùng ban đầu có thể được phân tách thêm thành nhiều nhóm con, tạo thành cấp thấp hơn. Sau đó, các cụm được hình thành bằng cách nhóm nhiều lần các cụm lân cận để tạo thành hệ thống phân cấp cao hơn.

Kỹ thuật tạo cây quyết định để phân loại cũng có thể được áp dụng cho nhiệm vụ rời rạc hóa. Kỹ thuật này thường sử dụng phương pháp tách top-down. Không giống như các phương pháp khác được đề cập cho đến nay, tiếp cận dựa trên cây quyết định là phương pháp rời rạc hóa có giám sát, tức là chúng sử dụng thông tin nhãn lớp của dữ liệu đầu vào. Ví dụ trên một tập dữ liệu về các triệu chứng của bệnh nhân trong đó mỗi bệnh nhân có một nhãn lớp chẩn đoán các bệnh liên quan. Thông tin phân phối của các lớp được sử dụng trong tính toán và xác định điểm phân tách của hệ thống phân cấp. Ý tưởng trực quan nhất đó là chọn các điểm phân tách sao cho một phân vùng nhất định chứa càng nhiều bộ giá trị của cùng một lớp càng tốt. Entropy là thước đo phổ biến nhất được sử dụng cho mục đích này. Để tách biệt một thuộc tính số, phương pháp chọn giá trị của thuộc tính có entropy nhỏ nhất làm điểm phân tách và phân vùng đệ quy các khoảng kết quả để tạo ra sự hệ thống phân cấp, và thông qua đó tiến hành rời rạc hóa. Bởi vì rời rạc hóa dựa trên cây quyết định sử dụng thông tin lớp, nhiều khả năng các ranh giới khoảng (điểm phân tách) được xác định sẽ được đặt ở vị trí có thể giúp cải thiện độ chính xác của bài toán phân loại.

Rời rạc hóa sử dụng phân tích tương quan: ChiMerge là một phương pháp tiêu biểu dựa trên χ^2. Các phương pháp rời rạc hóa nghiên cứu cho đến thời điểm này đều sử dụng chiến lược chia tách top-down. Điều này trái ngược với ChiMerge, sử dụng cách tiếp cận bottom-up bằng cách tìm các khoảng lân cận tốt nhất và sau đó hợp nhất chúng để tạo thành các khoảng lớn hơn theo một cách đệ quy. Tương tự với tiếp cận dựa trên cây quyết định, ChiMerge là phương pháp có giám sát khi sử dụng thông tin lớp. Phương pháp này dựa trên tư tưởng đó là các tần số lớp phải tương đối nhất quán trong một khoảng. Điều đó có nghĩa là nếu hai khoảng liền kề mà có phân phối các lớp rất giống nhau thì các khoảng đó có thể được hợp nhất, nếu không, chúng sẽ

vẫn tách biệt.

Ký hiệu (A_i, B_j) là trường hợp thuộc tính A nhận giá trị a_i và thuộc tính B nhận giá trị b_j, ta có công thức χ^2 đặc tả mối tương quan giữa 2 thuộc tính này:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{\text{ij}} - e_{\text{ij}})^2}{e_{\text{ij}}}. \tag{1.3.3-1}$$

trong đó o_{ij} là tần suất quan sát được của sự kiện (A_i, B_j) và e_{ij} là tần suất dự kiến của sự kiện đó, có thể tính theo công thức:

$$e_{\text{ij}} = \frac{count(A = a_i) * count(B = b_i)}{n}. \tag{1.3.3-2}$$

với n là số lượng các bản ghi, $count(A = a_i)$ là số lượng các bản ghi mà thuộc tính A nhận giá trị a_i, tương tự với $count(B = b_j)$. χ^2-tests trên kiểm định giả thiết rằng thuộc tính A và B là độc lập, tức là không có mối tương quan nào giữa chúng.

Thông qua χ^2-tests, ChiMerge tiến hành như sau: Ban đầu, mỗi giá trị riêng biệt của một thuộc tính được coi là một khoảng. χ^2-tests được thực hiện cho mỗi cặp khoảng liền kề. Các khoảng liền kề đảm bảo một lượng giá trị χ^2 (được gọi là ngưỡng) sẽ được hợp nhất với nhau, vì giá trị χ^2 thấp có nghĩa là có sự tương đồng về phân phối lớp. Quá trình hợp nhất này tiến hành đệ quy cho đến khi đáp ứng tiêu chí dừng xác định trước.

2.3.4 Chiếu dữ liệu

Trong chiếu dữ liệu, dữ liệu được chuyển đổi hoặc hợp nhất thành các dạng thích hợp để khai thác. Các chiến lược chiếu dữ liệu bao gồm:

- Làm mịn: có tác dụng loại bỏ nhiễu khỏi dữ liệu. Các kỹ thuật bao gồm nhóm thành các bin, hồi quy và phân cụm.
- Xây dựng và trích chọn đặc trưng: các thuộc tính được xây dựng hoặc lựa chọn từ tập hợp các thuộc tính đã cho để giúp ích quá trình khai thác.
- Tổng hợp: các phép toán tổng hợp hoặc tóm tắt được áp dụng cho dữ liệu nhằm xây dựng những trường giá trị mới. Ví dụ: dữ liệu bán hàng hàng ngày có thể được tổng hợp để tính tổng số tiền hàng tháng và hàng năm. Bước này thường được sử dụng trong việc xây dựng một khối dữ liệu để phân tích ở nhiều mức trừu tượng.
- Chuẩn hóa: các thuộc tính của tập dữ liệu được chia tỷ lệ để nằm trong cùng một phạm vi nhỏ hơn, chẳng hạn như -1,0 đến 1,0 hoặc 0,0 đến 1,0. Chuẩn hóa đã được đề cập cụ thể trong phần [].
- Rời rạc hóa: các giá trị thô của thuộc tính số (ví dụ: tuổi) được thay thế bằng nhãn khoảng thời gian (ví dụ: 0–10, 11–20,...) hoặc dưới dạng nhãn bậc cao (ví dụ: thanh niên, người lớn, người cao tuổi). Các nhãn có thể được tổ chức đệ quy thành các khái niệm cấp cao hơn, dẫn đến hình thành hệ thống phân cấp cho các

thuộc tính số và có thể xác định nhiều hơn một hệ thống phân cấp cho cùng một thuộc tính để đáp ứng nhu cầu của nhiều nhiệm vụ khác nhau. Phương pháp rời rạc hóa đã được trình bày cụ thể trong phần [].

- Xây dựng hệ thống phân cấp cho dữ liệu trừu tượng: các thuộc tính, ví dụ như 'đường phố' có thể được khái quát hóa thành các khái niệm cấp cao hơn, như 'thành phố' hoặc 'quốc gia'. Nhiều cấu trúc phân cấp cho các thuộc tính trừu tượng được ngầm định trong lược đồ cơ sở dữ liệu và có thể được xác định tự động ở mức định nghĩa lược đồ.

Có nhiều sự chồng chéo giữa các nhiệm vụ tiền xử lý dữ liệu. Các phương pháp làm mịn dữ liệu, trích chọn đặc trưng, chuẩn hóa, rời rạc hóa đã được trình bày ở các phần trước, ta sẽ tập trung vào các phương pháp xây dựng đặc trưng, tổng hợp đặc trưng và xây dựng hệ thống phân cấp cho dữ liệu trừu tượng.

Xây dựng đặc trưng

Phần này ta sẽ đề cập đến phương pháp phân tích thành phần chính (PCA). Mục đích của phương pháp này là xây dựng những m đặc trưng (thuộc tính) không tương quan với nhau từ tập các đặc trưng gốc ban đầu. Phương pháp này đồng thời nằm trong nhiệm vụ trích chọn đặc trưng và giảm thiểu dữ liệu.

Gọi x là vector của các mẫu đầu vào. Trong trường hợp là một mảng hình ảnh, x có thể được hình thành bằng cách duỗi thẳng của các phần tử mảng. Việc tạo ra các đặc trưng mới không tương quan lẫn nhau để tránh dư thừa thông tin, tức là xây dựng phép biến đổi từ x sang y thông qua ma trận biến đổi X sao cho:

$$y = A^T x. \tag{1.3.4-1}$$

với $E[y(i)y(j)] = 0, i \neq j$. Từ định nghĩa của ma trận tương quan, ta có:

$$R_y = E[yy^T] = E[A^T x x^T A] = A^T R_x A. \tag{1.3.4-2}$$

Tuy nhiên, ta biết được R_x là ma trận đối xứng nên các vector riêng của nó là trực giao lẫn nhau. Do đó, nếu ma trận A được chọn sao các cột của nó là đôi một trực giao $a_i, i = 0, 1..N-1$ của R_x thì R_y là ma trận đường chéo

$$R_y = A^T R_x A = \Lambda. \tag{1.3.4-3}$$

trong đó Λ là ma trận đường chéo có các phần tử trên đường chéo của nó là các giá trị riêng tương ứng $\Lambda_i, i = 0, 1, ..., N-1$ của R_x. Hơn nữa, giả sử R_x là xác định dương thì các giá trị riêng là dương. Phép biến đổi này còn được gọi là biến đổi Karhunen-Loève (KL) và nó đạt được mục tiêu ban đầu của chúng ta kì vọng là tạo ra các đặc trưng không tương quan lẫn nhau. Phép biến đổi KL có ý nghĩa cơ bản trong nhận dạng mẫu và trong một số ứng dụng xử lý tín hiệu và hình ảnh. Từ công thức 1.3.4-1, ta có:

$$x = \sum_{i=0}^{N-1} y(i) a_i \qquad y(i) = a_i^T x. \tag{1.3.4-4}$$

Ta sẽ định nghĩa vector trong không gian m chiều:

$$\hat{x} = \sum_{i=0}^{m-1} y(i)a_i \ . \tag{1.3.4-5}$$

trong đó chỉ có m vector cơ bản là được sử dụng. Như vậy, đây là phép chiếu của x lên không gian con được sinh bởi các hệ trực chuẩn các vector riêng. Ta sẽ thực hiện xây dựng $\hat{x}$ sao cho gần đúng x.

Áp dụng cực tiểu tổn thất trung bình (MSE), ta có:

$$L = E\left[||x - \hat{x}||^2\right] = E\left[\left\|\sum_{i=m}^{N-1} y(i)a_i\right\|^2\right]. \tag{1.3.4-6}$$

Ta thực hiện chọn $\hat{x}$ sao cho cực tiểu giá trị L trên. Từ 1.3.4-6 và sử dụng tính chất trực chuẩn của các vector riêng, ta có:

$$L = E\left[\left\|\sum_{i=m}^{N-1} y(i)a_i\right\|^2\right] = E\left[\sum_i \sum_j (y(i)a_i^T)(y(j)a_j)\right]. \tag{1.3.4-7}$$

$$= \sum_{i=m}^{N-1} E[\mathrm{y}^2(i)] = \sum_{i=m}^{N-1} a_i^T E[\mathrm{x}\mathrm{x}^T]a_i. \tag{1.3.4-8}$$

Kết hợp với biểu thức 1.3.4-6 và định nghĩa vector riêng, ta được kết quả cuối cùng:

$$L = E\left[||x - x||^2\right] = \sum_{i=m}^{N-1} a_i^T \lambda_i a_i = \sum_{i=m}^{N-1} \lambda_i. \tag{1.3.4-9}$$

Do đó, nếu chúng ta chọn trong (1.3.4-5) các giá trị riêng tương ứng với m giá trị riêng lớn nhất của ma trận tương quan, thì sai số trong (1.3.4-9) được giảm thiểu, là tổng của $N - m$ giá trị riêng nhỏ nhất. Hơn nữa, nó có thể được chỉ ra rằng đây cũng là MSE tối thiểu, so với bất kỳ phép gần đúng nào khác của x theo vector m-chiều.

Tổng hợp dữ liệu

Khác với việc giảm thiểu dữ liệu bằng phương pháp phân tích thành phần chính ở trên, trong phần này ta sẽ đề cập tới việc xây dựng thêm các đặc trưng cơ bản và có ý nghĩa từ tập các đặc trưng đã cho.

Lấy ví dụ cơ bản, hãy tưởng tượng rằng ta đã thu thập dữ liệu để phân tích. Những dữ liệu này bao gồm doanh số bán hàng của AllElectronics mỗi quý, trong các năm 2008 đến 2010. Tuy nhiên, chúng ta quan tâm đến doanh số hàng năm (tổng mỗi năm), hơn là tổng mỗi quý. Do đó, dữ liệu có thể được tổng hợp để có được kết quả tóm tắt tổng doanh số bán hàng mỗi năm thay vì mỗi quý. Sự tổng hợp này được minh họa trong hình 1.3.4-1. Tập dữ liệu kết quả có khối lượng nhỏ hơn, không làm mất thông tin cần thiết cho nhiệm vụ phân tích.

Hình 1.3.4-1: Dữ liệu AllElectronics từ năm 2008-2010 []

Như vậy, việc tổng hợp dữ liệu đòi hỏi cần phải có kiến thức liên quan tới các thuộc tính mà ta cần phân tích. Việc xây dựng được đặc trưng có ý nghĩa hay không sẽ ảnh hưởng lớn tới kết quả của nhiệm vụ ban đầu.

Xây dựng hệ thống phân cấp cho dữ liệu trừu tượng

Chúng ta xem xét biến đổi dữ liệu cho dữ liệu trừu tượng. Đặc biệt, chúng ta nghiên cứu sự tạo hệ thống phân cấp cho các thuộc tính trừu tượng. Các thuộc tính trừu tượng có một số lượng hữu hạn (nhưng có thể lớn) các giá trị riêng biệt, không có thứ tự giữa các giá trị, ví dụ như vị trí địa lý, loại công việc, loại vật phẩm... Định nghĩa thủ công về phân cấp có thể là một công việc tốn thời gian và đòi hỏi phải có kiến thức chuyên môn. Tuy nhiên, nhiều cấu trúc phân cấp là ẩn trong lược đồ cơ sở dữ liệu và có thể được xác định tự động ở cấp độ định nghĩa lược đồ. Khái niệm phân cấp có thể được sử dụng để biến đổi dữ liệu thành nhiều cấp độ chi tiết. Ví dụ: các mẫu khai thác dữ liệu liên quan đến bán hàng có thể liên quan đến các khu vực hoặc quốc gia cụ thể. Ta có 4 phương pháp tạo phân cấp cho dữ liệu trừu tượng như sau:

- Đặc tả thứ tự từng phần của các thuộc tính một cách trực tiếp bởi kiến thức chuyên môn: Phân cấp cho các thuộc tính hoặc thứ nguyên thường liên quan đến một nhóm thuộc tính. Người dùng hoặc chuyên gia có thể dễ dàng xác định hệ thống phân cấp bằng cách chỉ định thứ tự một phần hoặc toàn bộ các thuộc tính. Ví dụ: giả sử rằng cơ sở dữ liệu quan hệ chứa nhóm thuộc tính sau: đường phố, thành phố, tỉnh hoặc bang và quốc gia. Hệ thống phân cấp có thể được xác định bằng cách chỉ định thứ tự giữa các thuộc tính này, chẳng hạn như đường phố $<$ thành phố $<$ tỉnh hoặc bang $<$ quốc gia.

- Đặc tả một phần của hệ thống phân cấp bằng cách nhóm dữ liệu: Về cơ bản, đây là định nghĩa thủ công một phần của hệ thống phân cấp. Trong một cơ sở dữ liệu lớn, trong một số trường hợp sẽ không khả thi nếu xác định toàn bộ hệ thống phân cấp bằng cách liệt kê và sắp thứ tự các giá trị một cách rõ ràng. Ngược lại, chúng ta có thể dễ dàng chỉ định các nhóm cho một phần nhỏ dữ liệu trung gian. Ví dụ: sau khi chỉ định tỉnh và quốc gia đó tạo thành một hệ thống phân cấp ở cấp giản đồ, người dùng có thể xác định một số cấp trung gian theo cách thủ công, chẳng hạn như {"Thanh Trì", "Đống Đa", "Hoàng Mai"} $\subset$ "Hà Nội" và {“Hà Nội“, “thành phố Hồ Chí Minh“, “Hải Phòng“, “quần đảo Hoàng Sa“, “Trường Sa“} $\subset$ “Việt Nam".

- Đặc tả một tập hợp các thuộc tính, nhưng không xác định thứ tự từng phần của chúng: Ta có thể chỉ định một tập hợp các thuộc tính tạo thành một hệ thống phân cấp, nhưng bỏ qua việc xác định thứ tự từng phần của chúng. Sau đó, hệ thống có thể tự động tạo thứ tự thuộc tính để tạo ra một hệ thống phân cấp có ý nghĩa. Trong trường hợp không có kiến thức chuyên môn về ngữ nghĩa dữ liệu, ta không thể tìm thấy thứ tự phân cấp cho một tập hợp các thuộc tính trừu tượng tùy ý. Tuy nhiên ta có nhận xét rằng vì các khái niệm cấp cao hơn thường bao

gồm một số khái niệm cấp dưới trực thuộc, một thuộc tính xác định cấp khái niệm cao (ví dụ: quốc gia) thường sẽ chứa một số lượng giá trị khác biệt nhỏ hơn một thuộc tính xác định cấp khái niệm thấp hơn (ví dụ: đường phố, quận, huyện...). Dựa trên quan sát này, một hệ thống phân cấp có thể được tạo tự động dựa trên số lượng các giá trị riêng biệt cho mỗi thuộc tính trong tập thuộc tính đã cho. Thuộc tính có các giá trị khác biệt nhất được đặt ở cấp phân cấp thấp nhất. Số lượng giá trị khác biệt của một thuộc tính càng thấp, thì thuộc tính đó càng cao trong hệ thống phân cấp khái niệm được tạo. Quy tắc heuristic này hoạt động tốt trong nhiều trường hợp. Và sau thi thực hiện quá trình heuristic, một số hoán đổi hoặc điều chỉnh cấp cục bộ có thể được những người có chuyên môn về lĩnh vực của dữ liệu áp dụng khi cần thiết sau khi kiểm tra hệ thống phân cấp đã tạo.

- Đặc tả chỉ một tập hợp con của các thuộc tính: Được sử dụng khi chỉ có một ý tưởng mơ hồ về những gì nên được bao gồm trong hệ thống phân cấp. Do đó, có thể chỉ bao gồm một tập hợp con của các thuộc tính có liên quan trong đặc tả phân cấp. Ví dụ: thay vì bao gồm tất cả các thuộc tính có liên quan theo thứ bậc cho thuộc tính dạng "vị trí", ta có thể chỉ chỉ định thuộc tính dạng "đường phố" và "thành phố". Để xử lý các cấu trúc phân cấp được chỉ định một phần như vậy, điều quan trọng là phải nhúng ngữ nghĩa dữ liệu vào lược đồ cơ sở dữ liệu để các thuộc tính có kết nối ngữ nghĩa chặt chẽ có thể được kết nối lại với nhau. Bằng cách này, đặc tả của một thuộc tính có thể kích hoạt toàn bộ nhóm thuộc tính liên kết chặt chẽ về mặt ngữ nghĩa được “kéo vào” để tạo thành một hệ thống phân cấp hoàn chỉnh.

2.3.5 Lấy mẫu dữ liệu

Lấy mẫu có thể được sử dụng như một phương pháp giảm dữ liệu vì nó cho phép một tập dữ liệu lớn được biểu diễn bằng một mẫu dữ liệu ngẫu nhiên nhỏ hơn nhiều lần (hoặc có thể gọi là tập hợp con). Giả sử rằng một tập dữ liệu lớn D chứa N bản ghi. Ta sẽ tiến hành xem xét các cách phổ biến nhất mà có thể lấy mẫu D để giảm dữ liệu (hình 1.3-4).

Hình 1.3-4: Phương pháp lấy mẫu dữ liệu []

- Lấy mẫu ngẫu nhiên đơn giản không có thay thế (SRSWOR) có kích thước s: Mẫu này được tạo ra bằng cách lựa chọn s trong số N bản ghi từ D ($s < N$), trong đó xác suất lựa chọn bất kì bản ghi nào trong D là $1/N$, nghĩa là, tất cả các bản ghi đều là có khả năng được lấy mẫu như nhau.

- Lấy mẫu ngẫu nhiên đơn giản có thay thế (SRSWR) có kích thước s: Phương pháp này tương tự như SRSWOR, ngoại trừ mỗi lần một bản ghi được rút ra từ D, nó được ghi lại và sau đó có thể tiếp tục được lựa chọn. Nghĩa là, sau khi một bản ghi được rút ra, nó được đặt trở lại D để nó có thể được lựa chọn lại ở các lần chọn bản ghi tới.

- Lấy mẫu dựa trên phân cụm: Nếu các bản ghi trong D được nhóm thành M cụm tách rời nhau, ta quy về việc lấy mẫu ngẫu nhiên đơn giản s cụm trong số M cụm, với $s < M$. Ví dụ cơ bản: các bản ghi trong cơ sở dữ liệu thường được truy xuất trong một trang tại một thời điểm, vì vậy rằng mỗi trang có thể được coi là một cụm. Bên cạnh đó, ta có thể thực hiện giảm dữ liệu bằng cách áp dụng SRSWOR cho các trang, dẫn đến một mẫu cụm của các bản ghi. Các tiêu chí phân cụm khác nhau cũng có thể được sử dụng nhằm mang lại sự đa dạng và tùy biến cho kết quả.

- Lấy mẫu phân tầng: Nếu D được chia thành các phần rời rạc lẫn nhau được gọi là các tầng, thì lấy mẫu phân tầng của D được thực hiện bằng cách thu được mẫu ngẫu nhiên đơn giản tại mỗi tầng. Điều này giúp đảm bảo mẫu có tính đại diện, đặc biệt khi dữ liệu bị lệch. Ví dụ, một mẫu phân tầng có thể được lấy từ dữ liệu khách hàng, trong đó phân tầng cho từng nhóm đối tượng khách hàng dựa trên tuổi tác, giới tính. Bằng cách này, nhóm tuổi có số lượng khách hàng nhỏ nhất vẫn sẽ đảm bảo được đại diện trong mẫu dữ liệu.

Một ưu điểm của việc lấy mẫu để giảm dữ liệu là chi phí lấy mẫu tỷ lệ thuận với kích thước s của mẫu và tỉ lệ nghịch với kích thước N của tập dữ liệu D. Do đó, độ phức tạp của việc lấy mẫu có khả năng là tuyến tính với kích thước của dữ liệu. Các kỹ thuật rút gọn dữ liệu khác có thể yêu cầu ít nhất một lần duyệt hoàn toàn qua bộ dữ liệu D. Đối với kích thước mẫu cố định, độ phức tạp lấy mẫu chỉ tăng tuyến tính khi số thứ nguyên (số trường) n của dữ liệu tăng lên, trong khi các kỹ thuật khác (chẳng hạn như sử dụng biểu đồ) tăng theo cấp số nhân ở n.

Khi áp dụng để giảm dữ liệu, lấy mẫu thường được sử dụng nhất để ước tính câu trả lời cho một truy vấn tổng hợp. Có thể sử dụng định lý giới hạn trung tâm để xác định cỡ mẫu vừa đủ để ước lượng một hàm đã cho với một sai số chấp nhận được, và kích thước mẫu s này có thể cực kỳ nhỏ so với N. Lấy mẫu là một lựa chọn tự nhiên để cải tiến dần tập dữ liệu đã được giảm thiểu khi mà một tập như vậy có thể được tinh chỉnh thêm bằng cách tăng kích thước mẫu.

Chương 3

Phân tích mô tả

Chương 4

Phân tích dự báo

Phần I: Phân tích hồi quy và đánh giá mô hình

4.1 Phân tích hồi quy

Trong thống kê, **phân tích hồi quy** (Regression Analysis) là một họ các quy trình thống kê để ước lượng mối quan hệ giữa một biến phụ thuộc vào một hay một số biến độc lập khác nhau. Một trong những phương pháp hồi quy phổ biến nhất là hồi quy tuyến tính, tức tìm ra một đường thẳng (hay phức tạp hơn là một tổ hợp tuyến tính) mà khớp với dữ liệu theo một tiêu chí nhất định nào đó. Ví dụ, phương pháp bình phương cực tiểu sẽ tìm một đường thẳng sao cho tổng khoảng cách từ các điểm dữ liệu đến đường thẳng (hoặc siêu phẳng) là nhỏ nhất.

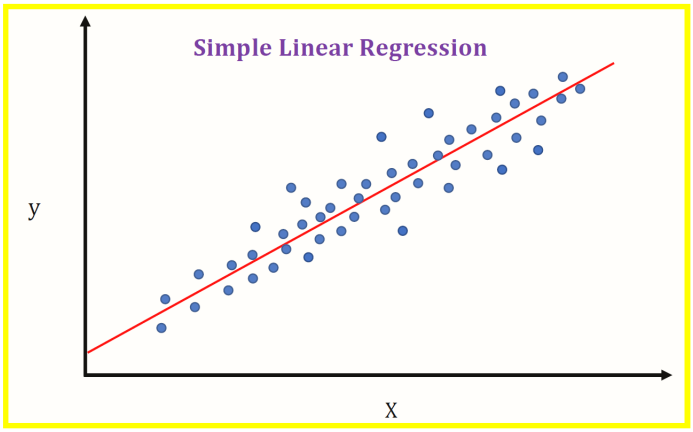
Phân tích hồi quy chủ yếu được sử dụng cho hai mục đích chính. Thứ nhất, người ta thường sử dụng phương pháp này cho những bài toán dự báo, ví dụ như giá chứng khoán, hay giá xăng dầu,... Thứ hai, phân tích hồi quy có thể được dùng để biểu diễn mối quan hệ phụ thuộc giữa những biến số khác nhau.

Trong thực tế, nhà nghiên cứu trước hết cần phải lựa chọn mô hình mà họ muốn ước lượng, sau đó lựa chọn phương pháp (ví dụ như bình phương cực tiểu) để ước lượng các tham số có trong mô hình. Mô hình hồi quy bao gồm những thành phần sau:

- **Tham số chưa biết** (unknown parameters), thường được ký hiệu là một vector β.
- **Biến độc lập** (independent variables), là biến ngẫu nhiên mà có những quan sát được ghi trong bộ dữ liệu. Biến độc lập thường được ký hiệu là vector X là biến ngẫu nhiên mà ta giả sử rằng phụ thuộc vào các biến X. Biến phụ thuộc thường được ký hiệu là Y.
- **Sai số**, biểu thị sai số của phép hồi quy và thường được ký hiệu là e.

Mô hình hồi quy giả định rằng Y là một hàm số của X và β, với e biểu thị sai số của mô hình:

$$Y = f(X, \beta) + e \tag{4.1}$$



Hình 4.1: Minh họa mô hình hồi quy tuyến tính đơn.

Người ta sẽ phải ước lượng các tham số của hàm $f(X_i, \beta)$ mà phù hợp với bộ dữ liệu nhất. Để có thể thực hiện phân tích hồi quy, dạng của hàm số f nói trên cần phải cố định. Việc lựa chọn dạng hàm của f đôi khi phụ thuộc vào cảm quan của người sử dụng mô hình, hoặc dựa vào những kinh nghiệm trước đó về mối quan hệ của Y_i theo X_i. Trong những trường hợp khác, người ta thường sử dụng những mô hình phổ thông như hồi quy tuyến tính hay hồi quy logistic để xây dựng mô hình.

4.1.1 Hồi quy tuyến tính đơn

Tổng quan

Hồi quy tuyến tính đơn (Simple linear regression) là một mô hình hồi quy tuyến tính, trong đó biến phụ thuộc Y chỉ phụ thuộc tuyến tính vào một biến độc lập duy nhất, tức là X là vector một chiều. Như vậy, tập dữ liệu sẽ bao gồm những điểm trong mặt phẳng, và công việc của chúng ta đó chính là tìm ra một đường thẳng trong mặt phẳng mà khớp với những điểm dữ liệu đó.

Thông thường, người ta thường giả sử rằng phương pháp bình phương cực tiểu (ordianary least squares) sẽ được sử dụng để ước lượng các tham số của mô hình. Một số phương pháp ước lượng tham số khác có thể kể đến như cực tiểu độ lệch chuẩn tuyệt đối (least absolute deviation) hay ước lượng Theil-Sen.

Mô hình

Xét phương trình:

$$y = \alpha + \beta x \tag{4.2}$$

Phương trình này mô tả một đường thẳng trong mặt phẳng với độ dốc là β và tung độ gốc (y-intercept) là α. Trên thực tế, các điểm dữ liệu không hoàn toàn thỏa mãn phương trình trên mà sẽ lệch khỏi đường thẳng một đoạn là một đại lượng ngẫu nhiên mà ta gọi là sai số ϵ. Giả sử chúng ta có một bộ dữ liệu gồm n điểm, gọi là

$\{(x_i, y_i), i = \overline{1, n}\}$. Ta có thể biểu diễn mối quan hệ của y_i theo x_i và ϵ_i như sau:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

Mối quan hệ giữa những tham số α, β và các điểm dữ liệu được gọi là mô hình hồi quy tuyến tính. Mục đích của chúng ta là tìm ra những ước lượng $\hat{\alpha}$ và $\hat{\beta}$ cho các tham số α và β mà khớp với dữ liệu nhất. Như đã đề cập ở trên, tiêu chí đánh giá cho độ khớp này mặc định sẽ là bình phương cực tiểu, tức sẽ cực tiểu hóa tổng bình phương các giá trị ϵ_i. Như vậy, bài toán hồi quy của chúng ta có thể đưa về bài toán tối ưu sau:

$$\underset{\alpha,\beta}{argmin} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

.

Sau khi khai triển và thu được dạng toán phương của α và β, ta có thể tính được giá trị tối ưu $\hat{\alpha}$ và $\hat{\beta}$ của hai tham số trên như sau:

$$\begin{aligned}
\hat{\alpha} &= \hat{y} - \hat{\beta} \\
\hat{\beta} &= \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \\
&= \frac{s_{x,y}}{s_x^2} \\
&= r_{xy}\frac{s_y}{s_x}.
\end{aligned}$$

Với:

- $\overline{x}$, $\overline{y}$ là giá trị trung bình của x_i và y_i.
- r_{xy} là hệ số tương quan mẫu (sample correlation coefficient) giữa x và y.
- s_x và s_y là độ lệch chuẩn mẫu chưa hiệu chỉnh (uncorrected sample standard deviation) của x và y.
- s^2 và s_{xy} theo thứ tự là phương sai mẫu và hiệp phương sai mẫu.

Thay các giá trị $\hat{\alpha}$ và $\hat{\beta}$ vào phương trình 4.2, ta thu được:

$$\frac{f - \overline{y}}{s_y} = r_{xy}\frac{x - \overline{x}}{s_x}.$$

Điều này chứng tỏ r_{xy} là độ dốc của đường thẳng hồi quy. Tương tự với ký hiệu $\overline{x}$, ta có thể ký hiệu dấu gạch ngang trên đầu để biểu thị giá trị trung bình của một đại lượng. Ví dụ như:

$$\overline{xy} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i$$

. Như vậy, giá trị r_{xy} sẽ được cho bởi:

$$r_{xy} = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\sqrt{(\overline{x^2} - \overline{x}^2)(\overline{y^2} - \overline{y}^2)}}$$

.

Các tính chất số học của mô hình hồi quy tuyến tính đơn

Ta có thể đến một số tính chất của mô hình hồi quy tuyến tính đơn:

- Đường hồi quy sẽ đi qua trọng tâm (center ò mas point), $(\overline{x}, \overline{y})$, nếu mô hình có bao gồm phần chặn (intercept term).
- Tổng của phần thặng dư $\hat{\epsilon_i}$ bằng 0 nếu mô hình bao gồm phần chặn:

$$\sum_{i=1}^{n} \hat{\epsilon_i} = 0$$

- Phần thặng dư và giá trị x có hệ số tương quan bằng 0, tức là chúng không tương quan cho dù mô hình có bao gồm phần chặn hay không, tức là:

$$\sum_{i=1}^{n} x_i \hat{\epsilon}_i = 0$$

Các tính chất thống kê của mô hình hồi quy tuyến tính đơn

Khi miêu tả các tính chất thống kê, ta cần phần sử dụng những thuật ngữ mô hình thống kê (statistical model).

1. **Tính không chệch** (unbiasedness): Các ước lượng của α và β, tức $\hat{\alpha}$ và $\hat{\beta}$ là các ước lượng không chệch.
2. **Khoảng tin cậy** (confidence interval): các công thức tính *alpha* và β được mô tả ở phần trước có thể coi là một ước lượng điểm của α, β, tức là hệ số của đường thẳng hồi quy của một bộ dữ liệu. Tuy nhiên, những công thức trên không cho biết độ chính xác của những ước lượng trên. Do đó, thuật ngữ khoảng tin cậy được giới thiệu ở đây để đưa ra một bộ giá trị hợp lý cho bộ ước lượng trong trường hợp các thí nghiệm được lặp đi lặp lại nhiều lần.

   Phương pháp để xây dựng khoảng tin cậy cho các hệ số của mô hình hồi quy tuyến tính dựa trên giả sử về phân phối chuẩn, tức là:

   (a) Các sai số hồi quy tuần theo phân phối chuẩn, hay cụ thể hơn là những ồn trắng (white noise) - có kỳ vọng bằng 0.

   (b) Số các điểm dữ liệu, hay số quan sát n là đủ lớn.

Ví dụ minh họa

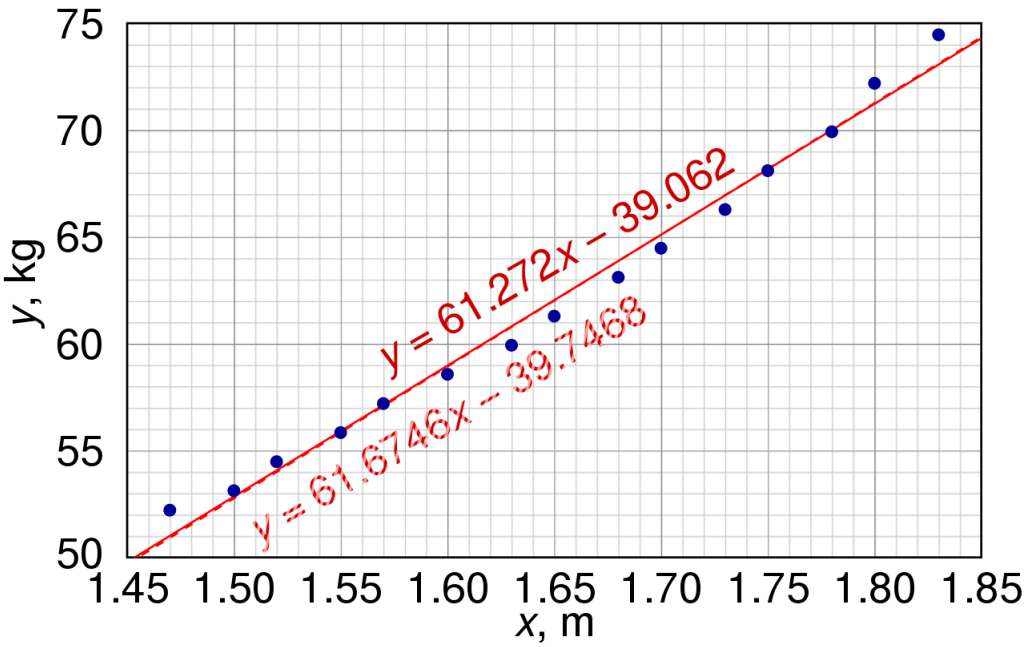
Xét một bộ dữ liệu về chiều cao và cân nặng của phụ nữ Mỹ trong khoảng từ 30 đến 39 tuổi. Sau khi thu gọn và trung bình hóa dữ liệu, người ta có bảng số liệu sau :

Chiều cao (m)	1.47	1.50	1.52	1.55	1.57	1.60	1.63	1.65	1.68	1.70	1.73	1.75	1.78	1.80
Cân nặng (kg)	52.1	53.1	54.8	55.8	57.2	58.5	59.9	61.9	63.1	64.4	66.8	68.1	69.9	72.1

Ta sẽ thực hiện hồi quy cân nặng dựa trên chiều cao. Ký hiệu x là cân nặng, y là chiều cao, ta sẽ tính được những giá trị sau:

$$
\begin{aligned}
S_x &= \sum x_i = 24.76 \\
S_y &= \sum y_i = 931.17 \\
S_{xx} &= \sum x_i^2 = 41.0532 \\
S_{yy} &= \sum y_i^2 = 58498.5439 \\
S_{xy} &= \sum x_i y_i = 1548.2453
\end{aligned}
$$

Sau khi tính toán những đại lượng trên, ta có thể dùng chúng để tính toán những tham số có trong mô hình hồi quy tuyến tính:



Hình 4.2: Đồ thị hồi quy

$$
\begin{aligned}
\hat{\beta} &= \frac{nS_{xy} - S_x S_y}{nS_{xx} - S_x^2} = 61.272 \\
\hat{\alpha} &= \frac{1}{n} S_y - \hat{\beta} \frac{1}{n} S_x = -39.062 \\
s_\epsilon^2 &= \frac{1}{n(n-2)} [nS_{yy} - S_y^2 - \hat{\beta}^2 (nS_{xx} - S_x^2)] = 0.5762 \\
s_{\hat{\beta}}^2 &= \frac{ns_\epsilon^2}{nS_{xx} - S_x^2} = 3.1539 \\
s_{\hat{\alpha}}^2 &= s_{\hat{\beta}}^2 \frac{1}{n} S_{xx} = 8.63185
\end{aligned}
$$

Khoảng tin cậy ứng với độ tin cậy 95% cho α và β là:

$$\alpha \in [\hat{\alpha} \pm t^*_{13} s_\alpha] = [-45.4, -32.7]; \beta \in [\hat{\beta} \pm t^*_{13} s_\beta] = [57.4, 65.1]$$

4.1.2 Hồi quy tuyến tính bội

Tổng quan

Hồi quy tuyến tính bội (multiple linear regression), hay gọi tắt là hồi quy tuyến tính, là một mô hình thống kê rất phổ biến. Khác với hồi quy tuyến tính đơn, hồi quy tuyến tính biểu diễn giá trị của một đại lượng thông qua hai hay nhiều đại lượng khác.

Trong mô hình hồi quy tuyến tính, mối quan hệ giữa các đại lượng được biểu diễn qua hàm dự đoán tuyến tính (linear predictor function), với các tham số của hàm được ước lượng qua bộ dữ liệu. Những mô hình như vậy được gọi là mô hình tuyến tính. Hồi quy tuyến tính là mô hình hồi quy được quan tâm nhiều nhất và có nhiều ứng dụng thực tiễn, bởi những mô hình tuyến tính thường dễ dàng ước lượng tham số hơn là những mô hình phi tuyến. Một số ứng dụng của mô hình hồi quy tuyến tính có thể kể đến như:

- Nếu mục đích của người sử dụng là dự báo, hoặc giảm thiểu sai số, mô hình hồi quy tuyến tính có thể được sử dụng để khớp vào những điểm dữ liệu đã biết. Sau khi khớp mô hình, từ những thông tin mới thu thập được, ta có thể sử dụng mô hình để tính toán những giá trị ứng với những thông tin mới đó.

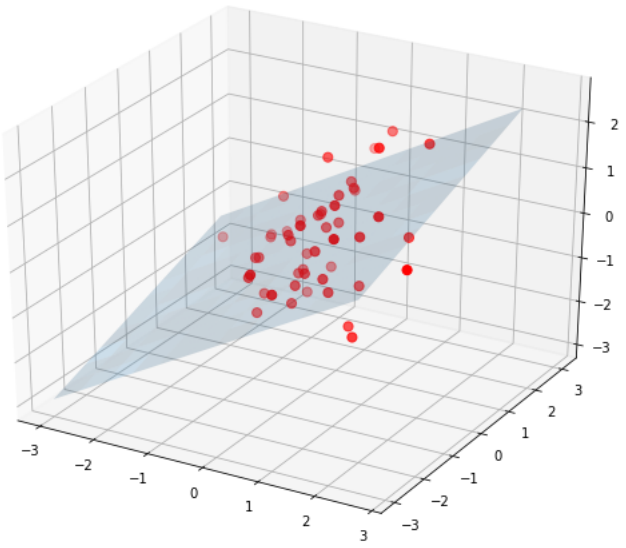
- Hồi quy tuyến tính cũng có thể được sử dụng để đánh giá mối tương quan giữa đại lượng cần hồi quy và những đại lượng sử dụng để hồi quy. Nếu hệ số của đại lượng sử dụng để hồi quy có trong mô hình là lớn, tức là đại lượng hồi quy phụ thuộc mạnh vào đại lượng sử dụng đề hồi quy này, và ngược lại.

Cũng giống như hồi quy tuyến tính đơn, phương pháp phổ biến để ước lượng các tham số trong mô hình hồi quy tuyến tính đó chính là bình phương cực tiểu. Do đó, người ta thường nhầm lẫn giữa cụm từ "bình phương cực tiểu" và "hồi quy tuyến tính". Trên thực tế, mô hình hồi quy tuyến tính có thể sử dụng phương pháp ước lượng khác, và bình phương cực tiểu cũng có thể áp dụng cho những mô hình phi tuyến. Do đó, hai cụm từ trên không nhất thiết là đồng nhất, mặc dù chúng có liên quan mật thiết đến nhau.

Mô hình

Xét một bộ dữ liệu $y_i, x_{i1}, \ldots, x_{ip}{}_{i=1}^{n}$, một mô hình tuyến tính giả sử rằng mối quan hệ giữa y và vector x là tuyến tính. Mối quan hệ này được mô hình hóa với sai số ϵ - một biến ngẫu nhiên biểu diễn nhiễu của mô hình tuyến tính. Mô hình tuyến tính trên sẽ có dạng:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \ldots, n,$$



Hình 4.3: Minh họa hồi quy tuyến tính bội.

Với $A^\top$ chỉ ma trận chuyển vị của ma trận A và $\boldsymbol{x_i}^\top\beta$ là tích vô hướng giữa hai vector β và x_i, $i = \overline{1,n}$. Ta cũng có thể viết lại phương trình trên ở dạng ma trận như sau:

$$y = X\beta + \epsilon$$

, với

$$\begin{aligned} \mathbf{y} &= \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix} \\ \boldsymbol{\beta} &= \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}. \end{aligned} \tag{4.3}$$

Một số lưu ý:

- $\boldsymbol{y}$ là vector các giá trị được quan sát $y_i (i = \overline{1,n})$ của biến được gọi là biến nội sinh (endogenous variable), biến phản hồi (response variable) hay biến phụ thuộc. Biến này đôi khi được gọi là biến được dự đoán (predicted variable) và không nên nhầm lẫn nó với giá trị dữ đoán (predicted values) ký hiệu là $\hat{y}$.
- $\boldsymbol{X}$ có thể được coi là ma trận các vector hàng x_i hoặc ma trận các vector cột X_j,

thường được gọi là biến ngoại sinh (exogenous value), biến giải thích (explanatory value), biến đầu vào (input variables), hay biến độc lập.

- $\boldsymbol{\beta}$ là một vector $(p+1)$-chiều, với β_0 được gọi là hệ số chặn. Hệ số của ma trận được gọi là hệ số hồi quy của mô hình hồi quy tuyến tính.
- $\boldsymbol{\epsilon}$ là vector gồm các giá trị ϵ_i. Nó được gọi là phần sai số hay phần nhiễu của mô hình hồi quy.

Khớp mô hình với một bộ dữ liệu tức là phải ước lượng giá trị cho hệ số hồi quy β sao cho phần sai số $\epsilon = y - X\beta$ nhỏ nhất. Thông thường, ta dùng tổng bình phương các giá trị ϵ_i làm tiêu chí khớp mô hình (phương pháp bình phương cực tiểu).

Phương pháp ước lượng

Rất nhiều phương pháp đã được nghiên cứu nhằm ước lượng các tham số cho mô hình hồi quy tuyến tính bội. Một số phương pháp có thể kể đến như bình phương cực tiểu, cực đại hàm hợp lý,... Ở phân này, ta sẽ tập trung đi sâu vào phương pháp ước lượng bình phương cực tiểu.

Giả sử rằng biến độc lập là $x_i = [x_1^i, \ldots, x_n^i]$ và tham số của mô hình là $\beta = [\beta_0, \ldots, \beta_n]$, như vậy giá trị dự đoán của mô hình sẽ là:

$$y_i \approx \beta_0 + \sum_{j=1}^{n} \beta_j x_j^i.$$

Nếu ta mở rộng $x_i = [1, x_1^i, \ldots, x_n^i]$, thì giá trị y_i sẽ bằng tích vô hướng của vector tham số *beta* và vector x_i:

$$y_i \approx \sum_{j=0}^{n} \beta_j x_j^i = \beta \cdot x.$$

Sử dụng bình phương cực tiểu làm phương pháp ước lượng tham số, như vậy tham số tối ưu $\hat{\beta}$ sẽ bằng:

$$\overrightarrow{\hat{\beta}} = \arg\min_{\vec{\beta}} L(D, \vec{\beta}) = \arg\min_{\vec{\beta}} \sum_{i=1}^{n} \left(\vec{\beta} \cdot \overrightarrow{x_i} - y_i \right)^2,$$

Với D là tập dữ liệu. Nếu viết các biến phụ thuộc và biến độc lập ở dạng ma trận, thì hàm mục tiêu có thể viết lại thành:

$$\begin{aligned} L(D, \vec{\beta}) &= \|X\vec{\beta} - Y\|^2 \\ &= (X\vec{\beta} - Y)^T (X\vec{\beta} - Y) \\ &= Y^T Y - Y^T X\vec{\beta} - \vec{\beta}^T X^T Y + \vec{\beta}^T X^T X\vec{\beta}. \end{aligned}$$

Do hàm mục tiêu là lồi, giá trị cực tiểu của nó sẽ đạt tại điểm có gradient bằng 0. Gradient của hàm số là:

$$\begin{aligned} \frac{\partial L(D, \vec{\beta})}{\partial \vec{\beta}} &= \frac{\partial \left(Y^T Y - Y^T X\vec{\beta} - \vec{\beta}^T X^T Y + \vec{\beta}^T X^T X\vec{\beta} \right)}{\partial \vec{\beta}} \\ &= -2Y^T X + 2\vec{\beta}^T X^T X. \end{aligned}$$

Giải phương trình gradient bằng 0, ta thu được tham số tối ưu của mô hình:

$$\begin{aligned} -2Y^TX + 2\vec{\beta}^T X^T X = 0 \\ \Rightarrow Y^TX = \vec{\beta}^T X^T X \\ \Rightarrow X^TY = X^TX\vec{\beta} \\ \Rightarrow \overrightarrow{\hat{\beta}} = \left(X^TX\right)^{-1} X^TY. \end{aligned}$$

Một số ứng dụng

Hồi quy tuyến tính bội được sử dụng rộng dãi trong sinh học, phân tích hành vi hay xã hội học, ... Nó được xem như là một trong số phương pháp quan trọng nhất trong những ngành khoa học nói trên.

Trong tài chính, mô hình định giá tài sản vốn (captial asset pricing model) sử dụng hồi quy tuyến tính để phân tích và đánh giá rủi do của khoản đầu tư.

Trong kinh tế học, hồi quy tuyến tính là phương pháp thực nghiệm được sử dụng nhiều nhất. Nó được dùng để dự báo nhu cầu tiêu thụ của thị trường, dự báo các khoản đầu tư bất động sản và hàng hóa, tổng giá trị xuất nhập khẩu của một quốc giá, hay dự đoán về thị trường lao động.

4.1.3 Hồi quy logistic

Tổng quan

Trong thống kê, mô hình logistic (logistic model) là một công cụ để mô hình hóa xác suất xảy ra của một sự kiện hay một lớp, ví dụ đỗ/trượt, thua/thắng, ... Nó có thể mở rộng với nhiều nhãn, ví dụ quyết định xem con vật xuất hiện trong bức ảnh là chó, mèo hay sư tử, ... Mỗi vật trong một bức ảnh sẽ được gán một vector ứng với xác suất vật đó thuộc về một lớp nào đó.

Hồi quy logistic là một mô hình thống kê mà dạng cơ bản của nó được sử dụng để mô hình hóa một biến phụ thuộc nhị phân (nhận một trong hai giá trị là 0 hoặc 1). Trong phân tích hồi quy, hồi quy logistic sẽ ước lượng tham số của một mô hình logistic. Về mặt toán học, một mô hình hồi quy logistic nhị phân gồm một biển phụ thuộc nhận giá trị nhị phân, biến phụ thuộc có thể nhận giá trị rời rạc hoặc liên tục tùy ý.

So sánh với hồi quy tuyến tính

Để cho trực quan, xét một tập dữ liệu gồm thời gian dành cho việc ôn thi và số điểm đạt được ở đợt thi đó. Khi đó, hồi quy logistic và hồi quy tuyến tính có thể được sử dụng để dự đoán những giá trị khác nhau:

- **Hồi quy tuyến tính**: sử dụng để dự đoán điểm của thí sinh (trong khoảng từ 0 đến 100 điểm). Giá trị trả về của hồi quy tuyến tính là một miền liên tục.
- **Hồi quy logistic**: sử dụng dự đoán liệu thí sinh đó trượt hay là qua môn. Giá trị trả về của hồi quy logistic thuộc một tập rời rạc.

Hàm logistic

Hàm logistic, còn gọi là hàm sigmoid, là một hàm nhận giá trị trong miền $\mathbb{R}$, và trả về giá trị trong khoảng $(0, 1)$:

$$\begin{aligned} \sigma : \mathbb{R} &\longrightarrow (0,1) \\ t &\longmapsto \sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^t} \end{aligned}$$

Đồ thị của hàm logistic được minh họa ở Hình 4.4.

Hình 4.4: Đồ thị của hàm logistic

Giả sử t là một hàm tuyến tính của một biến độc lập x. Ta có thể biểu diễn t như sau:

$$t = \beta_0 + \beta_1 x.$$

Như vậy, hàm logistic $p : \mathbb{R} \longrightarrow (0, 1)$ có thể viết dưới dạng:

$$p(x) = \sigma(t) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}.$$

Trong mô hình hồi quy logistic, $p(x)$ được biểu diễn bởi xác suất của một biến phụ thuộc Y thuộc về một (trong trường hợp 1 chiều) hay nhiều (trong trường hợp hai hay nhiều chiều) nhãn nào đó.

Trường hợp nhiều biến

Trong trường hợp nhiều biến, biểu thức $\beta_0 + \beta_1 x$ có thể biểu diễn lại thành $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m = \beta_0 + \sum_{i=1}^{m} \beta_i x_i$. Như vậy, các phương trình sẽ là:

$$\log \frac{p}{1 - p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m,$$

và

$$p = \frac{1}{1 + b^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m)}}$$

và thông thường người ta thường gán $b = e$.

Mô hình

Hồi quy logistic là một mô hình quan trọng trong học máy (machine learning). Mục đích của nó là để mô hình hóa xác suất của biến ngẫu nhiên Y bằng 0 hoặc 1 với những điểm dữ liệu được cho trước. Xét một mô hình tuyến tính tổng quát với tham số của mô hình là θ,

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}} = \Pr(Y = 1 \mid X; \theta).$$

Như vậy

$$\Pr(Y = 0 \mid X; \theta) = 1 - h_\theta(X).$$

Do $Y \in \{0, 1\}$ nên $\Pr(y \mid X; \theta) = h_\theta(X)^y \left(1 - h_\theta(X)\right)^{(1-y)}$. Chúng ta sẽ tình giá trị hàm hợp lý với giả sử rằng các quan sát trong tập dữ liệu là độc lập và cùng tuân theo phân phối Bernoulli:

$$\begin{aligned} L(\theta \mid y; x) &= \Pr(Y \mid X; \theta) \\ &= \prod_i \Pr\left(y_i \mid x_i; \theta\right) \\ &= \prod_i h_\theta\left(x_i\right)^{y_i} \left(1 - h_\theta\left(x_i\right)\right)^{(1-y_i)}. \end{aligned}$$

Thông thường, người ta sẽ sử dụng phương pháp hướng giảm gradient (gradient descent) để cực đại hóa log của hàm hợp lý:

$$N^{-1} \log L(\theta \mid y; x) = N^{-1} \sum_{i=1}^{N} \log \Pr\left(y_i \mid x_i; \theta\right).$$

Giả sử rằng các cặp (x, y) đều được lấy từ một phân phối nền nào đó, ta sẽ có:

$$\lim_{N \to +\infty} N^{-1} \sum_{i=1}^{N} \log \Pr\left(y_i \mid x_i; \theta\right) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \Pr(X = x, Y = y) \log \Pr(Y = y \mid X = x; \theta)$$

$$= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \Pr(X = x, Y = y) \left(- \log \frac{\Pr(Y = y \mid X = x)}{\Pr(Y = y \mid X = x; \theta)} + \log \Pr(Y = y \mid X = x) \right)$$

$$= - D_{\mathrm{KL}}\left(Y \| Y_\theta\right) - H(Y \mid X).$$

Với $H(X \mid Y)$ là entropy có điều kiện và D_{KL} là phân kỳ Kullback-Leibler (Kullback-Leibler divergence). Điều này có nghĩa là, để cực đại hóa hàm hợp lý của mô hình, chúng ta cần phải cực tiểu hóa phân kỳ KL của mô hình với phân phối entropy cực đại, tức là tìm kiếm mô hình mà sử dụng ít giả thuyết về tham số của mô hình đó nhất.

Phương pháp IRLS

Người ta cũng có thể sử dụng phương pháp IRLS (iteratively reweighted least squares) để tính toán các tham số của mô hình hồi quy logistic nhị phân. Nếu mô hình được biểu diễn dưới dạng ma trận, với tham số $\mathbf{w}^T = [\beta_0, \beta_1, \beta_2, \ldots]$, các biến độc lập $\mathbf{x}(i) = [1, x_1(i), x_2(i), \ldots]^T$, giá trị của w có thể được tính bởi phép lặp sau:

$$\mathbf{w}_{k+1} = \left(\mathbf{X}^T \mathbf{S}_k \mathbf{X}\right)^{-1} \mathbf{X}^T \left(\mathbf{S}_k \mathbf{X} \mathbf{w}_k + \mathbf{y} - \boldsymbol{\mu}_k\right),$$

với $\mathbf{S} = \mathrm{diag}(\mu(i)(1 - \mu(i)))$ là ma trận hệ số, $\boldsymbol{\mu} = [\mu(1), \mu(2), \ldots]$ là vector kỳ vọng,

$$\mathbf{X} = \begin{bmatrix} 1 & x_1(1) & x_2(1) & \ldots \\ 1 & x_1(2) & x_2(2) & \ldots \\ \vdots & \vdots & \vdots & \end{bmatrix},$$

là mà trận hồi quy và $\mathbf{y}(i) = [y(1), y(2), \ldots]^T$ là vector gồm các giá trị phản hồi (response variable) trong tập dữ liệu.

Ứng dụng

Hồi quy logistic được sử dụng trong rất nhiều lĩnh vực, bao gồm học máy, dịch tễ học hay các ngành khoa học xã hội. Ví dụ, Điểm số đánh giá mức độ chấn thương và thương tích (Trauma and Injury Serverity score), được sử dụng để đánh giá cơ hội sống của bệnh nhân, do Boyd và các cộng sự phát triển từ mô hình hồi quy logistic. Hồi quy logistic có thể được dùng để chẩn đoán một số căn bệnh như tiểu đường, tim mạch,... dựa trên những chỉ số sinh học như chiều cao, cân nặng, độ tuổi giới tính, tình trạng máu, ... Conditional random fields, một mở rộng của hồi quy logistic cho dữ liệu dạng chuỗi, thường được sử dụng trong những bài toán xử lý ngôn ngữ tự nhiên.

Ví dụ minh họa

Một nhóm 20 sinh viên dành từ 0 đến 6 tiếng để chuẩn bị cho một kỳ thi. Dữ liệu về thời gian ôn thi và kết quả thi được cho trong bảng dưới đây:

Số giờ	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.5	2.75	3.00
Kết quả	0	0	0	0	0	0	1	0	1	0	1	0

Số giờ	3.25	3.50	4.00	4.25	4.5	4.75	5.00	5.00
Kết quả	1	0	1	1	1	1	1	1

Từ đây, ta sẽ tìm được hệ số của β_0 và β_1 lần lượt bằng -4.0777 và 1.5046. Như vậy, xác suất để sinh viên vượt qua kỳ thi sẽ bằng:

$$p(x) = \frac{1}{1 + e^{-(1.5046.x - 4.0777)}},$$

với x là số giờ mà sinh viên dành để ôn bài chuẩn bị cho kỳ thi. Từ đây, ta có thể dự đoán được xác suất mà thí sinh vượt qua kỳ thi dựa trên thời gian ôn bài của họ. Ví dụ một sinh viên dành 2 giờ để ôn thi sẽ có xác suất vượt qua kỳ thi bằng:

$$p(2) = \frac{1}{1 + e^{-(1.5046 \cdot 2 - 4.0777)}} = 0.26.$$

Hay một thí sinh bỏ ra 4 tiếng để ôn thi sẽ có xác suất vượt qua kỳ thi bằng:

$$p(4) = \frac{1}{1 + e^{-(1.5046 \cdot 4 - 4.0777)}} = 0.87$$

4.2 Đánh giá mô hình

Khi ta xây dựng một mô hình phân lớp, chắc hẳn sẽ có những câu hỏi xuất hiện trong đầu chúng ta. Ví dụ, giả sử ta đang dùng dữ liệu của những lần mua hàng tại một cửa hàng nào đó để phân tích hành vi mua sắm của người tiêu dùng. Chúng ta muốn biết độ chính xác của mô hình này khi dự đoán hành vi mua hàng trong tương lai và những hành vi trong tương lai này hoàn toán không có trong tập luyện. Chúng ta có thể xây dựng một vài mô hình và so sánh sự chính xác của những mô hình này với nhau. Nhưng độ chính xác là gì? làm sao chúng ta có thể ước lượng được độ chính xác? Mô hình nào là phù hợp nhất trong số các mô hình mà chúng ta đã xây dựng? Những câu hỏi trên sẽ được trả lời trong phần này.

4.2.1 Sai số

Trong thống kê và tối ưu, sai số là khái niệm dùng để chỉ độ chênh lệch của giá trị quan sát được của một đại lượng so với giá trị tính toán được theo mô hình của đại lượng đó.

Người ta có thể dùng nhiều phương pháp để tính sai số. Mỗi chuẩn khác nhau sẽ tương ứng với một sai số khác nhau. Có thể kể đến một số phương pháp tính sai số phổ biến như chuẩn bình phương, giá trị tuyệt đối, ...

Để cho đơn giản, xét một bộ dữ liệu $D = \{(x_i, y_i)\}_{i=1}^n$ và sử dụng mô hình hồi quy tuyến tính đơn để khớp với bộ dữ liệu này. Sau khi khớp mô hình, ta sẽ viết được mô hình trên dưới dạng:

$$y = \hat{\beta}_0 + \hat{\beta}_1 x,$$

với $\hat{\beta}_0$ và $\hat{\beta}_1$ là những ước lượng của tham số mô hình nói trên. Như vậy, sai số của mô hình nếu dùng chuẩn bình phương sẽ bằng:

$$\epsilon = \sqrt{\sum_{i=1}^{n} (y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i)^2}.$$

Nếu dùng chuẩn L_1, sai số của mô hình sẽ bằng:

$$\epsilon = \sum_{i=1}^{n} |y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i|$$

Sai số của mô hình có thể được dùng để quyết định xem nên lựa chọn mô hình nào cho bài toán thực tế. Nếu ngay cả khi với các tham số tối ưu của mô hình, sai số vẫn rất lớn thì có thể mô hình mà chúng ta đang áp dụng không phù hợp với bài toán đó. Chúng ta nên cân nhắc thay đổi dạng của mô hình (ví dụ từ hồi quy tuyến tính đơn sang hồi quy tuyến tính bội), hoặc đổi mô hình (ví dụ từ hồi quy tuyến tính sang hồi quy logistic).

4.2.2 Độ chính xác

Độ chính xác là một đại lượng dùng để quyết định xem mức độ chính xác của mô hình, thường là mô hình phân lớp, trong việc dự đoán lớp của một đối tượng cho trước.

Để đơn giản, ta xét một bài toán phân lớp nhị phân trên tập dữ liệu $D = \{(x_i, y_i)\}_{i=1}^n$, với $y_i \in \{0, 1\}$, với f là mô hình dùng để phân lớp. Như vậy, ứng với mỗi đối tượng x_i thì mô hình sẽ dự báo đối tượng này thuộc lớp $f(x_i)$. Ta định nghĩa tập positive **P** là tập các đối tượng thuộc nhãn một và tập negative **N** là tập các đối tượng thuộc nhãn 0. Gọi true **T** là tập các đối tượng được nhận diện và false **N** là tập các đối tượng không được. Khi đó ta sẽ định nghĩa một số thuật ngữ như sau:

- Accuracy: độ chính xác khi nhận diện, được tính bởi $\frac{TP+TN}{P+N}$
- Error rate: sai số khi nhận diện, được tính bởi $\frac{FP+FN}{P+N}$

- Recall: Tỉ lệ nhận diện đúng những đối tượng thuộc tập positive, được tính bởi $\frac{TP}{P}$
- Specificity: Tỉ lệ nhận diện đúng các đối tượng thuộc tập negative, được tính bởi $\frac{TN}{N}$
- Precision: Được tính bởi $\frac{TP}{TP+FP}$

Từ đây, ta có thể thấy được rằng, khi đánh giá một mô hình, ta không nên chỉ quan tâm đến Accuracy của mô hình, mà còn phải xem xét đến những yếu tố khác như Precision và Recall của mô hình. Một mô hình có accuracy cao, nhưng nó lại bỏ qua hầu hết những đối tượng đáng lẽ ra cần nhận diện (recall thấp), thì mô hình đó vẫn là mô hình không tốt. Ta cần phải cân đối giữa những yếu tố này để lựa chọn ra mô hình phù hợp nhất.

Phần II: Phân lớp

Phân lớp hay phân loại (Classification) là một hình thức phân tích dữ liệu trích xuất các mô hình mô tả các lớp dữ liệu quan trọng. Các mô hình như vậy, được gọi là bộ phân lớp, dự đoán các nhãn lớp phân lớp (rời rạc, không có thứ tự). Chẳng hạn chúng ta có thể xây dựng một mô hình phân loại để phân loại các đơn xin vay ngân hàng là an toàn hay rủi ro. Những phân tích như vậy có thể giúp chúng ta hiểu rõ hơn về dữ liệu. Nhiều phương pháp phân lớp đã được các nhà nghiên cứu đề xuất trong học máy, nhận dạng mẫu và thống kê. Hầu hết các thuật toán thường giả sử kích thước dữ liệu nhỏ. Trong những năm gần đây đây, các nghiên cứu khai phá dữ liệu đã phát triển các kỹ thuật phân lớp và dự đoán xử lý trên tập dữ liệu lớn. Phân lớp có nhiều ứng dụng, bao gồm phát hiện gian lận, tiếp thị mục tiêu, dự đoán hiệu suất, sản xuất và chẩn đoán y tế.

4.3 Khái niệm về phân lớp

4.3.1 Phân lớp là gì?

Một nhân viên cho vay của ngân hàng cần phân tích dữ liệu để tìm hiểu về những người vay tiền với mục đích đánh giá người nào là "an toàn" và người nào là "rủi ro" cho ngân hàng. Một người quản lý tiếp thị tại AllElect Electronics cần phân tích dữ liệu để giúp dự đoán xem khách hàng có mua máy tính mới hay không. Một nhà nghiên cứu y tế muốn phân tích dữ liệu ung thư để dự đoán một trong ba phương pháp điều trị cụ thể mà bệnh nhân nên nhận. Trong mỗi ví dụ này, nhiệm vụ phân tích dữ liệu là phân lớp, trong đó mô hình được xây dựng để dự đoán các nhãn lớp, chẳng hạn như "An toàn" hay "Rủi ro" đối với dữ liệu ứng dụng cho vay; "Có" hoặc "Không" cho các dữ liệu tiếp thị; hoặc "điều trị A", "điều trị B", hoặc "điều trị C" đối với các dữ liệu y tế. Các loại này có thể được biểu diễn bằng các giá trị rời rạc, trong đó thứ tự giữa các giá trị không có ý nghĩa. Một người quản lý tiếp thị muốn dự đoán một khách hàng cụ thể sẽ tiêu bao nhiêu tiền trong một lần mua hàng tại AllElect Electronics. Nhiệm vụ phân tích dữ liệu này là một ví dụ về dự đoán hồi quy trong đó mô hình được xây dựng dự đoán là một hàm có giá trị liên tục hoặc giá trị được sắp xếp, trái ngược với nhãn lớp. Mô hình này là một dự đoán. Phân tích hồi quy là một phương pháp thống kê thường được sử dụng để dự đoán số; do đó hai thuật ngữ có xu hướng được sử dụng đồng nghĩa, mặc dù các phương pháp khác để dự đoán số tồn tại. Phân lớp và dự đoán hồi quy là hai loại vấn đề dự đoán chính.

4.3.2 Cách tiếp cận chung để phân lớp

"Phân lớp dữ liệu hoạt động như thế nào?" Phân lớp dữ liệu là một quá trình gồm hai bước, bao gồm một bước học tập (nơi xây dựng mô hình phân lớp) và bước phân lớp (trong đó mô hình được sử dụng để dự đoán nhãn lớp cho dữ liệu đã cho). Quá trình này được hiển thị cho dữ liệu ứng dụng cho vay của Hình 4.5. (Dữ liệu được đơn giản hóa cho mục đích minh họa). Trong thực tế, chúng ta có thể mong đợi nhiều thuộc

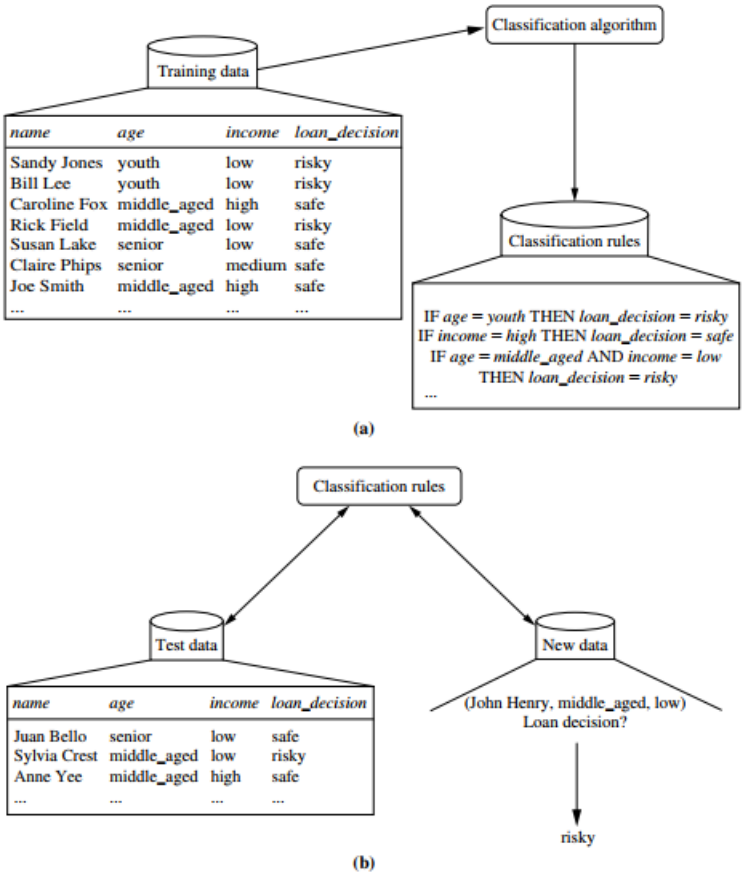
tính hơn sẽ được xem xét. Trong bước đầu tiên, một trình phân lớp được xây dựng mô tả một tập hợp các lớp hoặc khái niệm dữ liệu được xác định trước. Đây là bước học tập (hoặc giai đoạn đào tạo), trong đó thuật toán phân lớp xây dựng trình phân lớp bằng cách phân tích hoặc học hỏi từ một tập huấn luyện được tạo thành từ các bộ dữ liệu và nhãn lớp liên quan của chúng. Một bộ dữ liệu X, được biểu thị bằng một vectơ thuộc tính n chiều $X = (x_1, x_2, ..., x_n)$, mô tả n các phép đo được thực hiện trên bộ dữ liệu từ n thuộc tính cơ sở dữ liệu, lần lượt là $A_1, A_2, ..., A_n$. Mỗi bộ dữ liệu X được coi là thuộc về một lớp được xác định trước hay được xác định bởi một thuộc tính khác gọi là thuộc tính nhãn lớp. Thuộc tính nhãn lớp là giá trị rời rạc và không có thứ tự. Nó được phân loại (hoặc định danh) trong đó mỗi giá trị phục vụ như một danh mục hoặc lớp. Các bộ dữ liệu riêng lẻ tạo thành tập huấn luyện được tham chiếu như là bộ dữ liệu đào tạo và lấy mẫu ngẫu nhiên từ cơ sở dữ liệu được phân tích. Trong ngữ cảnh phân lớp, các bộ dữ liệu có thể được gọi là mẫu, ví dụ, trường hợp, điểm dữ liệu hoặc đối tượng.

Bởi vì nhãn lớp của mỗi bộ dữ liệu huấn luyện được cung cấp, bước này còn được gọi là học có giám sát (nghĩa là việc học của trình phân lớp được giám sát bởi vì nó được chỉ định cho mỗi điểm thuộc bộ dữ liệu huấn luyện thuộc về lớp nào). Nó tương phản với việc học tập không giám sát (hoặc phân cụm), trong đó không biết nhãn lớp của mỗi bộ huấn luyện, và số lượng hoặc tập hợp các lớp sẽ học có thể không được biết trước. Ví dụ: nếu chúng ta không có sẵn dữ liệu quyết định cho vay đối với tập huấn luyện, chúng ta có thể sử dụng phân cụm để cố gắng xác định các nhóm.

Bước đầu tiên của quy trình phân lớp này cũng có thể được xem là việc học ánh xạ hoặc hàm $y = f(X)$, có thể dự đoán nhãn lớp liên quan y của một bộ dữ liệu X. Trong quan điểm này, chúng ta muốn tìm hiểu ánh xạ hoặc hàm phân tách các lớp dữ liệu. Thông thường, ánh xạ này được thể hiện dưới dạng quy tắc phân lớp cây quyết định hoặc công thức toán học. Trong ví dụ của chúng ta, ánh xạ được biểu diễn dưới dạng các quy tắc phân lớp xác định các ứng dụng cho vay là an toàn hoặc rủi ro 4.5(a). Các quy tắc có thể được sử dụng để phân lớp các bộ dữ liệu trong tương lai, cũng như cung cấp cái nhìn sâu sắc hơn về nội dung dữ liệu. Họ cũng cung cấp một đại diện dữ liệu mẫu.

"Điều gì nói về độ chính xác khi phân lớp?" Ở bước thứ hai 4.5(b), mô hình được sử dụng để phân lớp. Đầu tiên, độ chính xác dự đoán của phân lớp được ước tính. Nếu chúng ta sử dụng tập huấn luyện để đo độ chính xác của bộ phân lớp, độ chính xác có thể rất cao, bởi vì bộ phân lớp có xu hướng phù hợp với dữ liệu (nghĩa là trong quá trình học, nó có thể kết hợp một số dị thường cụ thể của dữ liệu đào tạo không có trong dữ liệu chung thiết lập tổng thể). Do đó, một bộ kiểm tra được sử dụng, bao gồm các bộ kiểm tra và nhãn lớp liên quan của chúng. Chúng độc lập với các bộ dữ liệu huấn luyện, có nghĩa là chúng không được sử dụng để xây dựng bộ phân lớp. Độ chính xác của bộ phân lớp trên một bộ kiểm tra nhất định là tỷ lệ phần trăm của bộ kiểm tra được phân lớp chính xác bởi bộ phân lớp. Nhãn lớp liên quan của mỗi bộ kiểm tra được so sánh với dự đoán lớp của bộ phân lớp đã học cho bộ dữ liệu đó. Có một số phương pháp để ước tính độ chính xác của phân lớp. Nếu độ chính xác của trình phân lớp được coi là chấp nhận được, thì trình phân lớp có thể được sử dụng để phân lớp các bộ dữ liệu trong tương lai mà nhãn lớp không được biết. (Dữ liệu này cũng được đề cập trong tài liệu học máy như là dữ liệu chưa biết trước đó hoặc dữ liệu

chưa từng thấy trước đó.) Ví dụ, các quy tắc phân lớp đã học trong Hình 4.5(a) phân tích dữ liệu từ các ứng dụng cho vay trước đây có thể được sử dụng để phê duyệt hoặc từ chối người xin vay mới hoặc tương lai.



Hình 4.5: Quá trình phân lớp dữ liệu: (a) Học tập: Dữ liệu đào tạo được phân tích bằng thuật toán phân lớp. Ở đây, thuộc tính nhãn lớp là quyết định cho vay và mô hình đã học được thể hiện dưới dạng các quy tắc phân lớp. (b) Phân lớp: Dữ liệu thử nghiệm được sử dụng để ước tính độ chính xác của các quy tắc phân lớp. Nếu độ chính xác được coi là chấp nhận được, các quy tắc có thể được áp dụng để phân lớp các bộ dữ liệu mới.

4.4 Phương pháp học lười

Phương pháp *"học lười"* hay học tập từ các *"lân cận"* là khi được đưa ra một bộ dữ liệu đào tạo, một người lười học chỉ cần lưu trữ nó (hoặc chỉ thực hiện một chút xử lý nhỏ) và đợi cho đến khi nó được cung cấp một bộ dữ liệu kiểm tra. Chỉ khi nhìn thấy bộ kiểm tra thì nó mới thực hiện tổng quát hóa để phân loại bộ dữ liệu dựa trên sự tương tự của nó với các bộ huấn luyện được lưu trữ.

4.4.1 Thuật toán k láng giềng gần nhất

Phương pháp k láng giềng gần nhất (k-Nearest-Neighbor) (KNN) được mô tả lần đầu tiên vào đầu những năm 1950. Phương pháp tốn nhiều công sức và tài nguyên khi được luyện trên tập dữ liệu lớn và không được sử dụng phổ biến. Cho đến những năm 1960 khi sức mạnh tính toán tăng lên đáng kể, thuật toán đã được sử dụng rộng rãi trong lĩnh vực nhận dạng mẫu.

Thuật toán phân loại láng giềng gần nhất dựa trên việc học bằng phép loại suy, tức là bằng cách so sánh một bộ dữ liệu thử nghiệm đã cho với các bộ dữ liệu huấn luyện có sự *"tương tự"* với nó. Các bộ dữ liệu huấn luyện là được mô tả bởi n thuộc tính. Mỗi bộ đại diện là một điểm trong không gian n chiều. Trong cách này, tất cả các bộ dữ liệu huấn luyện được lưu trữ trong một không gian n chiều. Khi được đưa ra một bộ dữ liệu không xác định, bộ phân lớp KNN tìm kiếm không gian mẫu cho k dữ liệu luyện gần nhất với dữ liệu mới. k điểm trong dữ liệu huấn luyện là lân cận của điểm dữ liệu mới.

Khái niệm gần nhất được định nghĩa theo số đo khoảng cách, ví dụ như khoảng cách Euclidean. Khoảng cách Euclide giữa hai điểm hay hai bộ $X_1 = (x_{11}, x_{12}, ..., x_{1n})$ và $X_2 = (x_{21}, x_{22}, ..., x_{2n})$ là

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}.$$

Thông thường, chúng ta chuẩn hóa các giá trị của từng thuộc tính trước khi đưa vào công thức Euclide. Điều này có ý nghĩa giảm chênh lệch các thuộc tính có giá trị lớn (ví dụ: thu nhập) với các thuộc tính có giá trị nhỏ hơn (ví dụ: thuộc tính nhị phân). Ví dụ: chuẩn hóa tối đa có thể được sử dụng để chuyển đổi giá trị v của thuộc tính số A thành v trong phạm vi $[0, 1]$ bằng tính toán

$$v' = \frac{v - min_A}{max_A - min_A},$$

trong đó min_A và max_A là các giá trị tối thiểu và lớn nhất của thuộc tính A.

Đối với KNN, bộ dữ liệu không xác định được chỉ định hay gán nhãn bằng lớp phổ biến nhất trong số k lân cận gần nhất của nó. Khi $k = 1$, bộ dữ liệu không xác định được chỉ định lớp của bộ huấn luyện gần nhất với nó trong không gian mẫu. Các phân loại lân cận gần nhất cũng có thể được sử dụng để dự đoán số, nghĩa là để trả về một dự đoán có giá trị thực cho một dữ liệu chưa biết. Trong trường hợp này, bộ phân lớp

trả về giá trị trung bình của nhãn có giá trị thực liên quan đến k hàng xóm gần nhất của dữ liệu mới.

"Vậy làm thế nào có thể xác định một giá trị tốt nhất cho k?" Điều này có thể được xác định bằng thực nghiệm. Bắt đầu với $k = 1$, ta sử dụng bộ kiểm tra để ước tính tỷ lệ lỗi của phân lớp. Quá trình này có thể được lặp lại mỗi lần bằng cách tăng k để cho phép thêm một người hàng xóm. Giá trị k cho tỷ lệ lỗi tối thiểu có thể được chọn. Nói chung, số bộ dữ liệu luyện càng lớn, giá trị của k sẽ càng lớn (vì vậy phân lớp và quyết định dự đoán số có thể dựa trên một phần lớn hơn của các bộ lưu trữ). Khi số lượng bộ dữ liệu huấn luyện tiến đến vô cùng và $k = 1$, tỷ lệ lỗi có thể không tệ hơn hai lần tỷ lệ lỗi Bayes (sau này là lý thuyết tối thiểu). Nếu k cũng tiến đến vô cùng, tỷ lệ lỗi tiệm cận tỷ lệ lỗi Bayes.

4.4.2 Phân lớp lập luận theo trường hợp

Các bộ phân lớp lập luận dựa trên trường hợp (CBR) sử dụng cơ sở dữ liệu về các giải pháp vấn đề để giải quyết vấn đề mới. Không giống KNN, nơi lưu trữ các bộ dữ liệu luyện như các điểm trong không gian Euclide, CBR lưu trữ các bộ dữ liệu hoặc các trường hợp trên mạng để giải quyết vấn đề phức tạp mô tả tượng trưng. Các ứng dụng kinh doanh của CBR bao gồm giải quyết vấn đề trợ giúp dịch vụ khách hàng, trong đó các trường hợp mô tả các vấn đề chẩn đoán liên quan đến sản phẩm. CBR cũng đã được áp dụng cho các lĩnh vực như kỹ thuật và luật pháp, trong đó các tình huống là một trong hai thiết kế kỹ thuật hoặc phán quyết pháp lý, tương ứng. Giáo dục y tế là một lĩnh vực khác cho CBR, nơi lịch sử và phương pháp điều trị bệnh nhân được sử dụng để giúp chẩn đoán và điều trị bệnh nhân mới.

Khi được đưa ra một trường hợp mới để phân loại, một lập luận dựa trên trường hợp trước tiên sẽ kiểm tra xem có tồn tại trường hợp luyện giống hệt hay không. Nếu một trường hợp được tìm thấy, thì giải pháp đi kèm cho trường hợp đó được trả lại. Nếu không tìm thấy trường hợp giống hệt nhau, thì lập luận dựa trên trường hợp sẽ tìm kiếm trường hợp luyện có các thành phần tương tự như trường hợp mới. Về mặt khái niệm, các trường hợp luyện này có thể được coi là hàng xóm của trường hợp mới. Nếu trường hợp được biểu diễn dưới dạng đồ thị, điều này liên quan đến việc tìm kiếm các đồ thị con tương tự như các đồ thị con trong trường hợp mới. Các lập luận dựa trên trường hợp cố gắng kết hợp các giải pháp của các trường hợp luyện lân cận để đề xuất một giải pháp cho trường hợp mới. Nếu sự không tương thích phát sinh với các giải pháp riêng lẻ, thì quay lại để tìm kiếm các giải pháp khác có thể là cần thiết các lập luận dựa trên trường hợp có thể sử dụng kiến thức nền tảng và chiến lược giải quyết vấn đề để đề xuất một giải pháp kết hợp khả thi.

4.5 Thuật toán phân lớp Naive Bayes

"Phương pháp phân lớp Bayes là gì?" Phương pháp phân lớp Bayes là phân lớp thống kê. Phương pháp này có thể dự đoán các xác suất thành viên của lớp, chẳng hạn như xác suất một điểm dữ liệu đã cho thuộc về một lớp cụ thể. Phân lớp Bayes dựa trên định lý Bayes. Những nghiên cứu so sánh các thuật toán phân lớp đã tìm thấy một bộ phân lớp Bayes đơn giản được gọi là phân lớp Naive Bayes có thể so sánh về hiệu suất với phân lớp cây quyết định và các phân lớp mạng neural được chọn. Phân lớp Bayes cũng đã thể hiện độ chính xác và tốc độ cao khi áp dụng cho các cơ sở dữ liệu lớn. Trong thuật toán Navie Bayes yêu cầu giả thiết về sự độc lập của các chiều dữ liệu.

4.5.1 Định lý Bayes

Định lý Bayes được đặt theo tên của Thomas Bayes, một giáo sĩ người Anh, người đã sớm nghiên cứu trong lý thuyết xác suất và quyết định trong thế kỷ 18. Theo thuật ngữ Bayes, X được coi là "bằng chứng". Thông thường, nó được mô tả bằng các phép đo được thực hiện trên một tập hợp các thuộc tính n. Đặt H là một số giả thuyết, chẳng hạn như bộ dữ liệu X thuộc về một lớp được chỉ định C. Đối với các vấn đề xác định lớp , chúng ta muốn xác định $P(H|X)$, xác suất mà giả thuyết H được đưa ra "bằng chứng" dữ liệu X. Nói cách khác, chúng ta đang tìm kiếm xác suất mà bộ X thuộc về lớp C, với điều kiện là chúng ta biết mô tả thuộc tính của X.

$P(H|X)$ là xác suất sau, hoặc xác suất hậu nghiệm của H có điều kiện trên X. Ví dụ: giả sử bộ dữ liệu của chúng ta được giới hạn cho khách hàng được mô tả theo thuộc tính age (tuổi) và income (thu nhập); X là khách hàng 35 tuổi với thu nhập 40.000 USD. Giả sử H là giả thuyết rằng khách hàng của chúng ta sẽ mua máy tính. Sau đó, $P(H|X)$ phản ánh xác suất khách hàng X sẽ mua một máy tính cho chúng ta biết tuổi và thu nhập của khách hàng.

Ngược lại, $P(H)$ là xác suất trước, hoặc xác suất tiên nghiệm của H. Ví dụ, đây là xác suất mà bất kỳ khách hàng cụ thể nào sẽ mua máy tính, bất kể tuổi tác, thu nhập hoặc bất kỳ thông tin nào khác, cho vấn đề đó. Xác suất hậu nghiệm, $P(H|X)$, dựa trên nhiều thông tin hơn (ví dụ: thông tin khách hàng) so với xác suất trước đó, $P(H)$, độc lập với X.

Tương tự, $P(X|H)$ là xác suất hậu nghiệm của X dựa trên H. Đó là xác suất mà một khách hàng 35 tuổi và kiếm được 40.000 đô la, cho rằng chúng ta biết khách hàng sẽ mua máy tính.

$P(X)$ là xác suất trước của X. Sử dụng ví dụ của chúng ta, đó là xác suất mà một người trong nhóm khách hàng của chúng ta 35 tuổi và kiếm được 40.000 đô la.

"Những xác suất này được ước tính như thế nào?" $P(H)$, $P(X|H)$ và $P(X)$ có thể được ước tính từ dữ liệu đã cho, như chúng ta sẽ thấy tiếp theo. Định lý Bayes rất hữu ích ở chỗ nó cung cấp cách tính xác suất sau $P(H|X)$ từ $P(H)$, $P(X|H)$ và $P(X)$. Định lý Bayes là:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Bây giờ chúng ta đã có cách đó, trong phần tiếp theo, chúng ta sẽ xem xét định lý

Bayes được sử dụng như thế nào trong phân lớp Naive Bayes .

4.5.2 Phân lớp Naive Bayes

Trình phân lớp Naive Bayes hoặc phân lớp Bayes đơn giản hoạt động như sau:

1. Đặt D là tập huấn luyện các bộ dữ liệu và nhãn lớp liên quan của chúng. Như thường lệ, mỗi bộ dữ liệu được biểu thị bằng vectơ thuộc tính n chiều, $X = (x_1, x_2, \ldots, x_n)$ mô tả n phép đo được thực hiện trên bộ dữ liệu từ n thuộc tính, lần lượt là $A_1, A_2, \ldots, A_n$.

2. Giả sử có m lớp $C_1, C_2, \ldots, C_m$. Đưa ra một bộ X, trình phân lớp sẽ dự đoán rằng X thuộc về lớp có xác suất hậu nghiệm cao nhất dựa trên X. Nghĩa là, trình phân lớp Bayes dự đoán rằng X thuộc về lớp C_i khi và chỉ khi:

$$P(C_i|X) > P(C_j|X), 1 \leq j \leq m, j \neq i.$$

   Do đó, chúng ta tối đa hóa $P(C_i|X)$. Lớp C_i để $P(C_i|X)$ đạt giá trị cực đại được gọi là giả thuyết tối đa. Theo định lý của Bayes

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}.$$

3. Vì $P(X)$ là hằng số cho tất cả các lớp, chỉ $P(X|C_i)P(C_i)$ cần được cực đại. Nếu các xác suất của lớp không được biết đến, thì người ta thường cho rằng các lớp có xác suất xảy ra như nhau, nghĩa là $P(C_1) = P(C_2) = ... = P(C_m)$ và do đó chúng ta sẽ cực đại hóa $P(X|C_i)$. Mặt khác, chúng ta tối đa hóa $P(X|C_i)P(C_i)$. Lưu ý rằng xác suất của lớp có thể được ước tính bởi $P(C_i) = |C_{i,D}|/|D|$ trong đó $|C_{i,D}|$ là số dữ liệu đào tạo của lớp C_i trong D.

4. Với các tập dữ liệu có nhiều thuộc tính, sẽ rất tốn kém khi tính toán $P(X|C_i)$. Để giảm tính toán trong việc đánh giá $P(X|C_i)$, giả định chưa từng có về sự độc lập có điều kiện được đưa ra. Điều này giả định rằng các giá trị của các thuộc tính độc lập có điều kiện với nhau, được đưa ra nhãn lớp của dữ liệu (nghĩa là không có mối quan hệ phụ thuộc giữa các thuộc tính). Như vậy

$$P(X|C_i) = \prod_{k+1}^{n} P(x_k|C_i) = P(x_1|C_i)P(x_2|C_i)...P(x_n|C_i).$$

   Chúng ta có thể dễ dàng ước tính xác suất $P(x_1|C_i), P(x_2|C_i), \ldots, P(x_n|C_i)$ từ các bộ dữ liệu đào tạo. Lưu ý rằng ở đây x_k đề cập đến giá trị của thuộc tính A_k cho dữ liệu X. Đối với mỗi thuộc tính, chúng ta xem xét thuộc tính đó là rời rạc hay có giá trị liên tục. Chẳng hạn, để tính $P(X|C_i)$ chúng ta xem xét các yếu tố sau:

   (a) Nếu A_k là rời rạc, thì $P(x_k|C_i)$ là số bộ dữ liệu của lớp C_i trong D có giá trị x_k cho A_k, chia bởi $|C_{i,D}|$ số bộ dữ liệu của lớp C_i trong D.

(b) Nếu A_k có giá trị liên tục, thì chúng ta cần thực hiện thêm một chút công việc, nhưng việc tính toán khá đơn giản. Một thuộc tính có giá trị liên tục thường được giả định là có phân phối Gauss với giá trị trung bình μ và độ lệch chuẩn σ, được xác định bởi

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(x-\mu)^2}{2\sigma^2}$$

Như vậy,

$$P(x_k, C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

Chúng ta cần tính toán μ_{C_i} và σ_{C_i}, là giá trị trung bình và độ lệch chuẩn tương ứng của các giá trị của thuộc tính A_k để đào tạo các bộ dữ liệu của lớp C_i. Sau đó thay hai số lượng này vào biểu thức, cùng với x_k, để ước tính $P(x_k|C_i)$.

5. Để dự đoán nhãn lớp của X, $P(X|C_i)P(C_i)$ được ước tính cho mỗi lớp C_i. Trình phân lớp dự đoán rằng nhãn lớp của X là lớp C_i khi và chỉ khi

$$P(C_i|X) > P(C_j|X), 1 \leq j \leq m, j \neq i.$$

Nói cách khác, nhãn lớp dự đoán là lớp C_i mà $P(X|C_i)P(C_i)$ đạt cực đại.

"Các phân lớp Bayes có hiệu quả như thế nào?" Các nghiên cứu thực nghiệm khác nhau về phân lớp này so với các phân lớp cây quyết định và mạng nơ-ron đã tìm thấy nó có thể so sánh được trong một số lĩnh vực. Về lý thuyết, các phân lớp Bayes có tỷ lệ lỗi tối thiểu so với tất cả các phân lớp khác. Tuy nhiên, trong thực tế, điều này không phải lúc nào cũng đúng, do không chính xác trong các giả định được đưa ra để sử dụng, chẳng hạn như tính độc lập có điều kiện của lớp và thiếu dữ liệu xác suất có sẵn. Các trình phân lớp Bayes cũng hữu ích ở chỗ chúng cung cấp một sự biện minh về mặt lý thuyết cho các phân lớp khác không sử dụng rõ ràng định lý Bayes. Ví dụ, theo các giả định nhất định, có thể chỉ ra rằng nhiều thuật toán mạng nơ-ron và thuật toán khớp đường cong đưa ra giả thuyết tối đa, cũng như phân loại Naive Bayes.

Ví dụ *Dự đoán nhãn lớp bằng cách sử dụng phân lớp Bayes*. Chúng ta muốn dự đoán nhãn lớp của một dữ liệu bằng cách sử dụng phân lớp Naive Bayes. Dữ liệu huấn luyện đã được hiển thị trước đó trong Bảng 4.6. Các bộ dữ liệu được mô tả theo thuộc tính age, income, student và credit rating. Thuộc tính nhãn lớp là buys computer, có hai giá trị riêng biệt (cụ thể là yes, no). Đặt C_1 tương ứng với lớp buys computer = yes và C_2 tương ứng với buys computer = no. Bộ dữ liệu chúng ta muốn phân lớp là X = (age = youth, income = medium, student = yes, credit rating = fair)

Chúng ta cần tối đa hóa $P(X|C_i)P(C_i)$ với $i = 1, 2$. $P(C_i)$-xác suất trước của mỗi lớp, có thể được tính dựa trên các bộ dữ liệu đào tạo:

$$P(\text{buys computer} = \text{yes}) = 9/14 = 0.643$$

$$P(\text{buys computer} = \text{no}) = 5/14 = 0.357$$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Hình 4.6: Dữ liệu huấn luyện từ cơ sở dữ liệu khách hàng của AllElectronicsh.

Để tính $P(X|C_i)$ với $i = 1, 2$, chúng ta tính các xác suất có điều kiện sau:

$$P(\text{age} = \text{youth} \mid \text{buys computer} = \text{yes}) = 2/9 = 0.222$$

$$P(\text{age} = \text{youth} \mid \text{buys computer} = \text{no}) = 3/5 = 0.600$$

$$P(\text{income} = \text{medium} \mid \text{buys computer} = \text{yes}) = 4/9 = 0.444$$

$$P(\text{income} = \text{medium} \mid \text{buys computer} = \text{no}) = 2/5 = 0.400$$

$$P(\text{studient} = \text{yes} \mid \text{buys computer} = \text{yes}) = 6/9 = 0.467$$

$$P(\text{studient} = \text{yes} \mid \text{buys computer} = \text{no}) = 1/5 = 0.200$$

$$P(\text{credit rating} = \text{fair} \mid \text{buys computer} = \text{yes}) = 6/9 = 0.467$$

$$P(\text{credit rating} = \text{fair} \mid \text{buys computer} = \text{no}) = 2/5 = 0.400$$

Sử dụng các xác suất này, chúng ta có được

$$P(X|\text{buys computer} = \text{yes}) = P(\text{age} = \text{youth} \mid \text{buys computer} = \text{yes})\times$$

$P(\text{income} = \text{medium} \mid \text{buys computer} = \text{yes})\times P(\text{student} = \text{yes} \mid \text{buys computer} = \text{yes})\times$
$P(\text{credit rating} = \text{fair} \mid \text{buys computer} = \text{yes}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$

Tương tự

$$P(\text{X}|\text{buys computer} = \text{no}) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$$

Để tìm lớp C_i, tối đa hóa $P(X|C_i)P(C_i)$, chúng ta tính toán

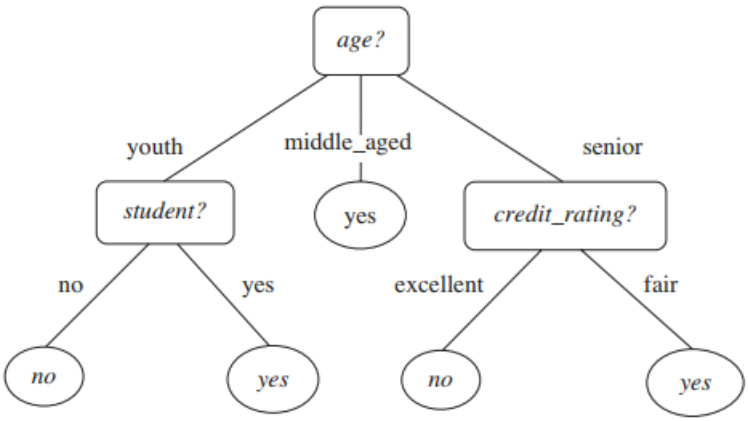
$$P(X|\text{buys computer} = \text{yes})P(\text{buys computer} = \text{yes}) = 0.0440.643 = 0.028$$

$$P(X|\text{buys computer} = \text{no})P(\text{buys computer} = \text{no}) = 0.0190.357 = 0.007$$

Do đó, trình phân loại Naive Bayes dự đoán buys computer=yes cho dữ liệu X.

4.6 Cây quyết định quy nạp

Cây quyết định quy nạp là việc học các cây quyết định từ các bộ huấn luyện được dán nhãn lớp. Cây quyết định là một cấu trúc cây giống như sơ đồ, trong đó mỗi nút bên trong (không là nút lá) biểu thị một kiểm tra trên một thuộc tính, mỗi nhánh biểu thị một kết quả của kiểm tra và mỗi nút lá (hoặc nút đầu cuối) giữ một nhãn lớp. Nút trên cùng trong cây là nút gốc. Một cây quyết định điển hình được hiển thị trong Hình 4.7. Nó đại diện cho khái niệm mua máy tính, nghĩa là, nó dự đoán liệu một khách hàng tại AllElect Electronics có khả năng mua máy tính hay không. Các nút trong được biểu thị bằng hình chữ nhật và các nút lá được biểu thị bằng hình bầu dục. Một số thuật toán cây quyết định chỉ tạo ra các cây nhị phân (trong đó mỗi nút nội bộ phân nhánh ra hai nút khác), trong khi các thuật toán khác có thể tạo ra các cây không nhị phân.



Hình 4.7: Cây quyết định cho khái niệm mua máy tính, cho biết liệu khách hàng của AllElect Electronic có khả năng mua máy tính hay không. Mỗi nút nội bộ (không là lá) đại diện cho một thử nghiệm trên một thuộc tính. Mỗi nút lá đại diện cho một lớp (buy computer = yes hoặc buy computer = no).

"Các loại cây quyết định được sử dụng để phân lớp như thế nào?" Cho một bộ dữ liệu X, mà nhãn lớp liên quan không xác định, các giá trị thuộc tính của bộ dữ liệu được kiểm tra đối với cây quyết định. Một đường dẫn được truy tìm từ gốc đến nút lá, dự đoán lớp cho bộ dữ liệu đó. Cây quyết định có thể dễ dàng được chuyển đổi đến quy tắc phân lớp.

"Tại sao các trình phân lớp cây quyết định lại phổ biến đến vậy?" Việc xây dựng các trình phân lớp cây quyết định không yêu cầu bất kỳ kiến thức miền hoặc cài đặt tham số nào, do đó nó thích hợp để tìm kiếm ra tri thức. Cây quyết định có thể xử lý dữ liệu đa chiều. Đại diện về kiến thức thu được ở dạng cây là trực quan và thường dễ bị đồng hóa bởi con người. Các bước học tập và phân lớp của cây quyết định quy nạp rất đơn giản và nhanh chóng. Nói chung, phân lớp cây quyết định có độ chính xác

tốt. Tuy nhiên, việc sử dụng thành công có thể phụ thuộc vào dữ liệu. Các thuật toán cây quyết định quy nạp đã được sử dụng để phân lớp trong nhiều lĩnh vực ứng dụng như y học, chế tạo và sản xuất, phân tích tài chính, thiên văn học và sinh học phân tử. Cây quyết định là cơ sở của một số hệ thống quy tắc thương mại.

Trong Phần 4.4.1, chúng ta mô tả một thuật toán cơ bản để học cây quyết định. Trong quá trình xây dựng cây, các phương pháp chọn thuộc tính được sử dụng để chọn thuộc tính phân vùng tốt nhất cho các bộ dữ liệu thành các lớp riêng biệt. Các phương pháp phổ biến của lựa chọn thuộc tính được đưa ra trong Mục 4.4.2. Khi cây quyết định được xây dựng, nhiều nhánh có thể phản ánh sai số hoặc ngoại lai trong dữ liệu huấn luyện. Cắt tỉa cây là cố gắng xác định và loại bỏ các nhánh như vậy, với mục tiêu cải thiện độ chính xác phân lớp trên dữ liệu không nhìn thấy. Cắt tỉa cây được mô tả trong Phần 4.4.3.

4.6.1 Thuật toán cây quyết định

Vào cuối những năm 1970 và đầu những năm 1980, Ross Quinlan, một nhà nghiên cứu về học máy, đã phát triển một thuật toán cây quyết định được gọi là ID3 (Iterative Dichotomiser). Công việc này được mở rộng trên công trình trước đó về hệ thống học khái niệm, được mô tả bởi E. B. Hunt, J. Marin và P. T. Stone. Quinlan sau đó đã trình bày C4.5 (một phiên bản kế nhiệm của ID3), trở thành một điểm chuẩn mà các thuật toán học tập có giám sát mới hơn thường được so sánh. Năm 1984, một nhóm các nhà thống kê (L. Breiman, J. Friedman, R. Olshen, và C. Stone) đã xuất bản cuốn sách *Classification and Regression Trees (CART)*, mô tả quá trình tạo ra cây quyết định nhị phân.

ID3, C4.5 và CART áp dụng cách tiếp cận tham lam, trong đó cây quyết định được xây dựng theo cách phân chia và sử dụng thủ tục đệ quy từ trên xuống. Phần lớn các thuật toán để quy nạp cây quyết định cũng tuân theo cách tiếp cận từ trên xuống, bắt đầu với một tập hợp các bộ giá trị đào tạo và các nhãn lớp liên quan của chúng. Tập huấn luyện được phân chia một cách đệ quy thành các tập con nhỏ hơn khi cây đang được xây dựng. Các chiến lược của thuật toán như sau:

- Thuật toán được gọi với ba tham số: *D, attribute list (danh sách thuộc tính) và attribute selection method (phương thức lựa chọn các thuộc tính).* Chúng ta đề cập đến D như là một phân vùng dữ liệu. Ban đầu, nó là bộ hoàn chỉnh của các bộ dữ liệu huấn luyện và nhãn lớp liên quan của chúng Phương thức lựa chọn thuộc tính chỉ định một thủ tục heuristic để chọn thuộc tính "tốt nhất" phân biệt các bộ giá trị đã cho theo lớp. Thủ tục này cần một cách thức để lựa chọn thuộc tính, chẳng hạn như dựa trên chỉ số Information gain hoặc Gini index.

- Cây bắt đầu như một nút đơn N, đại diện cho các bộ dữ liệu huấn luyện trong D (bước 1).

  Thuật toán: Tạo cây quyết định. Tạo một cây quyết định từ các bộ dữ liệu đào tạo của phân vùng dữ liệu D.

  Đầu vào

– Phân vùng dữ liệu D, là một tập hợp các bộ dữ liệu huấn luyện và nhãn lớp liên quan của chúng;

– Danh sách thuộc tính, tập hợp các thuộc tính ứng cử viên;

– Phương thức lựa chọn thuộc tính, một thủ tục để xác định tiêu chí chia tách phân vùng tốt nhất các bộ dữ liệu thành các lớp riêng lẻ. Tiêu chí này bao gồm một *splitting attribute* (thuộc tính phân tách) và có thể là *split-point* điểm chia hoặc *splitting subset* (tập hợp con chia).

Đầu ra: Một cây quyết định.

Phương thức

Method:

(1) create a node N;
(2) **if** tuples in D are all of the same class, C, **then**
(3) return N as a leaf node labeled with the class C;
(4) **if** *attribute_list* is empty **then**
(5) return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply **Attribute_selection_method**(D, *attribute_list*) to **find** the "best" *splitting_criterion*;
(7) label node N with *splitting_criterion*;
(8) **if** *splitting_attribute* is discrete-valued **and**
multiway splits allowed **then** // not restricted to binary trees
(9) *attribute_list* ← *attribute_list* − *splitting_attribute*; // remove *splitting_attribute*
(10) **for each** outcome j of *splitting_criterion*
// partition the tuples and grow subtrees for each partition
(11) let D_j be the set of data tuples in D satisfying outcome j; // a partition
(12) **if** D_j is empty **then**
(13) attach a leaf labeled with the majority class in D to node N;
(14) **else** attach the node returned by **Generate_decision_tree**(D_j, *attribute_list*) to node N;
endfor
(15) return N;

Hình 4.8: Thuật toán cơ bản để tạo cây quyết định từ bộ dữ liệu luyện.

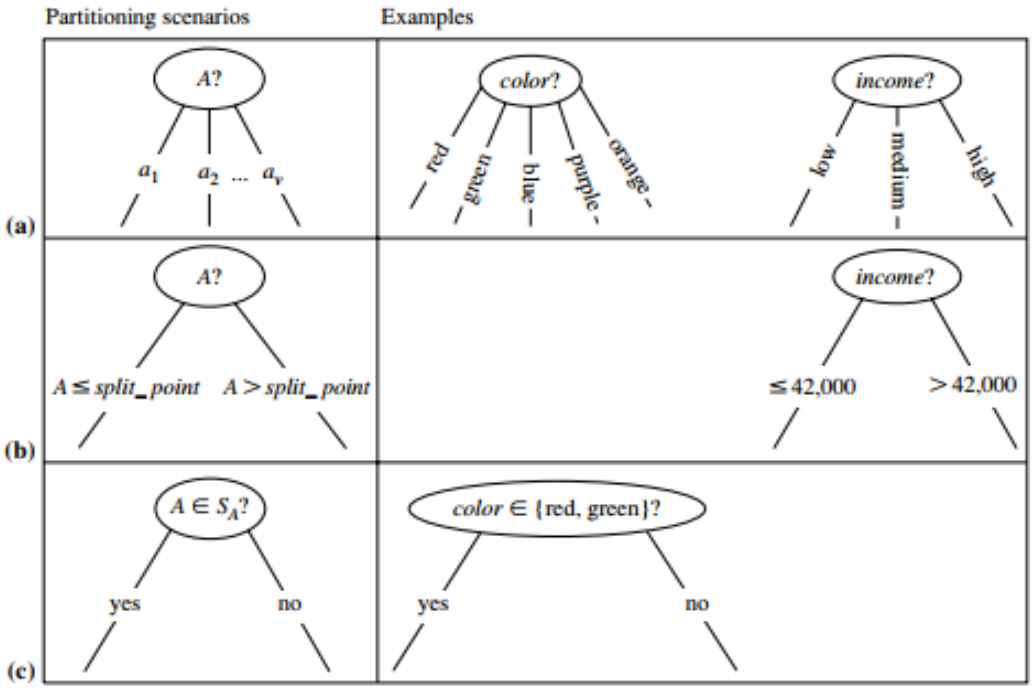
- Nếu các bộ dữ liệu trong D là cùng một lớp, thì nút N trở thành một chiếc lá và được gắn nhãn với lớp đó (bước 2 và 3). Lưu ý rằng bước 4 và 5 là các điều kiện kết thúc. Tất cả các điều kiện kết thúc được giải thích ở cuối thuật toán.

- Mặt khác, thuật toán gọi phương thức chọn thuộc tính để xác định tiêu chí chia tách. Tiêu chí phân tách cho chúng ta biết thuộc tính nào cần kiểm tra tại nút N bằng cách xác định cách tốt nhất để phân tách hoặc phân vùng các bộ dữ liệu trong D thành các lớp riêng lẻ (bước 6). Tiêu chí phân tách cũng cho chúng ta biết những nhánh nào sẽ phát triển từ nút N liên quan đến kết quả của bài kiểm tra đã chọn. Cụ thể hơn, tiêu chí chia tách chỉ ra thuộc tính chia tách và cũng có thể chỉ ra một điểm chia. Tiêu chí chia tách được xác định sao cho lý tưởng nhất là các phân vùng kết quả ở mỗi nhánh càng "tinh khiết" càng tốt . Một phân vùng là "tinh khiết" nếu tất cả các bộ dữ liệu trong nó thuộc về cùng

một lớp. Nói cách khác, nếu chúng ta chia các bộ dữ liệu trong D theo kết quả loại trừ lẫn nhau của tiêu chí chia tách, chúng ta hy vọng các phân vùng kết quả sẽ "tinh khiết" nhất có thể.

- Nút N được gắn nhãn với tiêu chí chia tách, đóng vai trò kiểm tra tại nút (bước 7). Một nhánh được phát triển từ nút N cho mỗi kết quả của tiêu chí chia tách. Các bộ dữ liệu trong D được phân vùng tương ứng (bước 10 đến 11). Có ba trường hợp có thể xảy ra, như được minh họa trong Hình 4.9. Đặt A là thuộc tính chia. A có v giá trị riêng biệt, $a_1, a_2, ..., a_v$, dựa trên dữ liệu đào tạo.

    1. A có giá trị rời rạc: Trong trường hợp này, kết quả của thử nghiệm tại nút N tương ứng trực tiếp với các giá trị đã biết của A. Một nhánh được tạo cho mỗi giá trị đã biết a_j của A và được gắn nhãn với giá trị đó (Hình 4.9a). Phân vùng D_j là tập hợp con của các bộ dữ liệu được gắn nhãn lớp trong D có giá trị a_j là A. Bởi vì tất cả các bộ dữ liệu trong một phân vùng nhất định có cùng giá trị cho A, không cần xem xét phân vùng trong các bộ dữ liệu trong tương lai. Do đó, nó bị xóa khỏi danh sách thuộc tính (bước 8 và 9).
    2. A có giá trị liên tục: Trong trường hợp này, thử nghiệm tại nút N có hai kết quả có thể xảy ra, tương ứng với các điều kiện $A \leq$ điểm chia và $A >$ điểm chia tương ứng, trong đó điểm chia là điểm phân tách được trả về bằng phương pháp chọn Thuộc tính như một phần của tiêu chí chia. (Trong thực tế, điểm chia a, thường được coi là trung điểm của hai giá trị liền kề đã biết của A và do đó có thể không thực sự là giá trị có sẵn của A từ dữ liệu đào tạo.) Hai nhánh được trồng từ N và được dán nhãn theo đến các kết quả trước đó (Hình 4.9b). Các bộ dữ liệu được phân vùng sao cho D_1 giữ tập hợp con của các bộ dữ liệu được gắn nhãn lớp trong D mà $A \leq$ điểm chia, trong khi D_2 giữ phần còn lại.



Hình 4.9: Hình cho thấy ba khả năng phân vùng các bộ dữ liệu dựa trên tiêu chí chia tách, mỗi bộ có các ví dụ. Đặt A là thuộc tính tách. (a) Nếu A có giá trị rời rạc, thì một nhánh được tăng cho mỗi giá trị đã biết của A. (b) Nếu A có giá trị liên tục, thì hai nhánh được tăng trưởng, tương ứng với điểm chia $\leq$ A và điểm chia $> A$. (c) Nếu A có giá trị rời rạc và phải tạo ra cây nhị phân, thì phép thử có dạng $A \in S_A$, trong đó S_A là tập con tách cho A.

3. A có giá trị rời rạc và cây nhị phân phải được tạo ra (như được quy định bởi phương pháp lựa chọn thuộc tính hoặc thuật toán đang được sử dụng): Thử nghiệm tại nút N có dạng hình thức "$A \in S_A$?" Trong đó S_A là tập hợp con cho A , được trả về bằng phương pháp chọn thuộc tính như một phần của tiêu chí chia tách. Nó là tập hợp con của các giá trị đã biết của A. Nếu một bộ dữ liệu đã cho có giá trị aj của A và nếu $a_j \in S_A$, thì thử nghiệm tại nút N được thỏa mãn. Hai nhánh được trồng từ N (Hình 4.9c). Theo quy ước, nhánh bên trái của N được dán nhãn có để D_1 tương ứng với tập hợp con của các bộ dữ liệu được gắn nhãn lớp trong D thỏa mãn thử nghiệm. Nhánh bên phải của N được dán nhãn không sao cho D_2 tương ứng với tập hợp các bộ dữ liệu được gắn nhãn lớp từ D không thỏa mãn thử nghiệm.

- Thuật toán sử dụng cùng một quy trình đệ quy để tạo thành cây quyết định cho các bộ dữ liệu tại mỗi phân vùng kết quả D_j của D (bước 14).

- Phân vùng đệ quy chỉ dừng khi bất kỳ một trong các điều kiện kết thúc sau là đúng:

1. Tất cả các bộ dữ liệu trong phân vùng D (đại diện tại nút N) thuộc về cùng một lớp (bước 2 và 3).
2. Không có thuộc tính nào còn lại trên đó các bộ dữ liệu có thể được phân vùng thêm (bước 4). Trong trường hợp này, bỏ phiếu đa số được sử dụng (bước 5). Điều này liên quan đến việc chuyển hướng nút N thành một chiếc lá và gắn nhãn nó với lớp phổ biến nhất trong D. Ngoài ra, phân phối lớp của các bộ nút có thể được lưu trữ.
3. Không có bộ dữ liệu nào cho một nhánh nhất định, nghĩa là phân vùng D_j trống (bước 12). Trong trường hợp này, một chiếc lá được tạo ra với lớp đa số trong D (bước 13).

- Cây quyết định được trả về (bước 15).

Độ phức tạp tính toán của thuật toán đưa ra tập huấn luyện D là $O(n \times |D| \times log(|D|))$, trong đó n là số thuộc tính mô tả các bộ dữ liệu trong D và $|D|$ là số bộ dữ liệu huấn luyện trong D. Điều này có nghĩa là chi phí tính toán để trồng cây phát triển tối đa $n \times |D| \times log(|D|)$ với $|D|$ bộ dữ liệu. Các phiên bản nâng cao của cây quyết định quy nạp cũng đã được đề xuất. Khi được cung cấp dữ liệu đào tạo mới, các cấu trúc này sẽ tái cấu trúc cây quyết định có được từ việc học trên dữ liệu đào tạo trước đó, thay vì học lại một cây mới từ đầu. Sự khác nhau trong thuật toán cây quyết định bao gồm cách các thuộc tính được chọn trong việc tạo cây và các cơ chế được sử dụng để cắt tỉa. Thuật toán cơ bản được mô tả trước đó yêu cầu một lần vượt qua các bộ dữ liệu huấn luyện trong D cho mỗi cấp độ của cây. Điều này có thể dẫn đến thời gian đào tạo dài và thiếu bộ nhớ khả dụng khi xử lý các cơ sở dữ liệu lớn.

4.6.2 Các phương pháp lựa chọn thuộc tính

Một phương pháp lựa chọn thuộc tính là một heuristic để chọn tiêu chí phân tách tách biệt tốt nhất một phân vùng dữ liệu nhất định D của các bộ dữ liệu đào tạo được gắn nhãn lớp thành các lớp riêng lẻ. Nếu chúng ta chia D thành các phân vùng nhỏ hơn theo kết quả của tiêu chí chia tách, lý tưởng là mỗi phân vùng sẽ thuần túy (nghĩa là tất cả các bộ dữ liệu rơi vào một phân vùng đã cho sẽ thuộc cùng một lớp). Về mặt khái niệm, tiêu chí chia tách tốt nhất là một tiêu chí có kết quả chặt chẽ nhất trong một trường hợp như vậy. Các phương pháp lựa chọn thuộc tính còn được gọi là quy tắc chia tách vì chúng xác định cách các bộ dữ liệu tại một nút nhất định được phân chia.

Phương pháp lựa chọn thuộc tính cung cấp một thứ hạng cho từng thuộc tính mô tả các bộ dữ liệu huấn luyện đã cho. Thuộc tính có điểm số tốt nhất được chọn làm thuộc tính chia cho các bộ dữ liệu đã cho. Nếu thuộc tính chia tách có giá trị liên tục hoặc nếu chúng ta bị giới hạn ở các cây nhị phân, thì tương ứng, một điểm phân tách hoặc một tập hợp con cũng phải được xác định là một phần của tiêu chí chia tách. Nút cây được tạo cho phân vùng D được gắn nhãn với tiêu chí chia tách, các nhánh được phát triển cho từng kết quả của tiêu chí và các bộ dữ liệu được phân vùng tương ứng. Phần này mô tả ba biện pháp lựa chọn thuộc tính phổ biến là: thông tin đạt được (Information gain), tỷ lệ khuếch đại (gain ratio) và chỉ số Gini.

Ký hiệu được sử dụng ở đây là như sau. Xét D, phân vùng dữ liệu, là tập hợp các bộ dữ liệu được gắn nhãn lớp. Giả sử thuộc tính nhãn lớp có m giá trị riêng biệt xác định m các lớp riêng biệt, C_i(với $i = 1, ..., m$). Đặt $C_{i,D}$ là tập hợp các bộ dữ liệu của lớp C_i trong D. Giả sử $|D|$ và $|C_{i,D}|$ lần lượt là số bộ dữ liệu trong D và $C_{i,D}$.

Information gain

ID3 sử dụng mức thông tin đạt được (Information gain) làm thước đo lựa chọn thuộc tính của nó. Biện pháp này dựa trên công trình tiên phong của Claude Shannon về lý thuyết thông tin, nghiên cứu về giá trị hoặc "nội dung thông tin" của các thông điệp. Đặt nút N đại diện hoặc giữ các bộ phân vùng D. Thuộc tính có mức thông tin đạt được (Information gain) cao nhất được chọn làm thuộc tính chia cho nút N. Thuộc tính này giảm thiểu thông tin cần thiết để phân lớp các bộ dữ liệu trong các phân vùng kết quả và phản ánh mức độ ngẫu nhiên hoặc ít tạp chất trong các phân vùng này. Cách tiếp cận như vậy sẽ giảm thiểu số lượng thử nghiệm dự kiến cần thiết để phân lớp một bộ dữ liệu nhất định và đảm bảo rằng một cây đơn giản (nhưng không nhất thiết là đơn giản nhất) được tìm thấy. Thông tin dự kiến cần thiết để phân lớp một bộ dữ liệu trong D được cung cấp bởi:

$$Info(D) = \sum_{i=1}^{m} p_i \log(p_i)$$

trong đó p_i là xác suất khác không mà một hàng tùy ý trong D thuộc về lớp C_i và được ước tính bởi $|C_{i,D}|/|D|$. Hàm log cơ số 2 được sử dụng, vì thông tin được mã hóa theo bit. $Info(D)$ chỉ là lượng thông tin trung bình cần thiết để xác định nhãn lớp của một hàng trong D. Lưu ý rằng, tại thời điểm này, thông tin chúng ta có chỉ dựa trên tỷ lệ của các bộ dữ liệu của mỗi lớp. $Info(D)$ còn được gọi là entropy của D.

Bây giờ, giả sử chúng ta đã phân vùng các bộ dữ liệu trong D trên một số thuộc tính A có v giá trị riêng biệt: $a_1, a_2, ..., a_v$, như được quan sát từ dữ liệu huấn luyện. Nếu A có giá trị rời rạc, các giá trị này tương ứng trực tiếp với kết quả v của thử nghiệm trên A. Thuộc tính A có thể được sử dụng để phân chia D thành v phân vùng hoặc tập hợp con $D_1, D_2, ..., D_v$, trong đó D_j chứa các bộ dữ liệu trong D có kết quả a_j của A. Các phân vùng này sẽ tương ứng với các nhánh được phát triển từ nút N. Lý tưởng nhất, chúng ta muốn phân vùng này tạo ra một phân lớp chính xác của các bộ dữ liệu. Đó là, chúng ta muốn cho mỗi phân vùng là tinh khiết. Tuy nhiên, nhiều khả năng các phân vùng sẽ không tinh khiết (ví dụ: trong đó một phân vùng có thể chứa một bộ các bộ dữ liệu từ các lớp khác nhau thay vì từ một lớp duy nhất).

Chúng ta vẫn cần thêm bao nhiêu thông tin (sau khi phân vùng) để đi đến một phân lớp chính xác? Số lượng này được đo bằng:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j).$$

Thuật ngữ $(|D_j|)/(|D|)$ đóng vai trò là trọng số của phân vùng thứ j. $Info_A(D)$ là thông tin dự kiến cần thiết để phân loại một bộ từ D dựa trên phân vùng theo A. Thông tin dự kiến càng nhỏ, độ tinh khiết của các phân vùng càng lớn.

Thông tin đạt được (Information gain) được định nghĩa là sự khác biệt giữa yêu cầu thông tin ban đầu (nghĩa là chỉ dựa trên tỷ lệ của các lớp) và yêu cầu mới (nghĩa

là có được sau khi phân vùng trên A). Đó là:

$$Gain(A) = Info(D) - Info_A(D).$$

Nói cách khác, $Gain(A)$ cho chúng ta biết sẽ kiếm được bao nhiêu khi phân nhánh trên A. Đó là mức giảm dự kiến trong yêu cầu thông tin gây ra bởi việc biết giá trị của A. Thuộc tính A có thông tin thu được cao nhất là $Gain(A)$, được chọn làm thuộc tính chia tách tại nút N. Điều này tương đương với việc chúng ta muốn phân vùng trên thuộc tính A sẽ thực hiện phân lớp tốt nhất, sao cho lượng thông tin cần để hoàn thành phân lớp các bộ dữ liệu là tối thiểu (tức là $Info_A(D)$ nhỏ nhất).

Gain Ratio

Các biện pháp thông tin đạt được (Imfomation gain) là thiên về các thử nghiệm với nhiều kết quả. Đó là, nó thích chọn các thuộc tính có số lượng giá trị lớn. Ví dụ: xem xét một thuộc tính hoạt động như một định danh duy nhất, chẳng hạn như product ID. Một phân chia trên product ID sẽ dẫn đến một số lượng lớn các phân vùng (có nhiều giá trị), mỗi phân vùng chỉ chứa một bộ. Vì mỗi phân vùng là tinh khiết, thông tin cần thiết để phân lớp tập dữ liệu D dựa trên phân vùng này sẽ là $Info_{productID}(D) = 0$. Do đó, thông tin thu được bằng cách phân vùng trên thuộc tính này là tối đa. Rõ ràng, một phân vùng như vậy là vô ích đối với phân lớp.

C4.5, một kế thừa của ID3, sử dụng một phần mở rộng từ thông tin đạt được (Information gain) được gọi là tỷ lệ khuếch đại (Gain Ratio), cố gắng khắc phục sự thiên vị này. Nó áp dụng một loại bình thường hóa để thông tin đạt được bằng cách sử dụng thông tin phân tách của điểm giá trị được định nghĩa tương tự với Info(D) như sau :

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log(\frac{|D_j|}{|D|}).$$

Giá trị này biểu thị thông tin tiềm năng được tạo bằng cách chia tập dữ liệu huấn luyện D thành các phân vùng v tương ứng với kết quả v của thử nghiệm trên thuộc tính A. Lưu ý rằng, đối với mỗi kết quả, nó xem xét số lượng bộ dữ liệu có kết quả đó đối với tổng số bộ dữ liệu trong D. Nó khác với mức thông tin đạt được (Information gain), đo lường thông tin liên quan đến phân lớp được thu thập dựa trên cùng một phân vùng. Tỷ lệ khuếch đại (Gain ratio) được xác định là:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}.$$

Thuộc tính có tỷ lệ đạt được (gain ratio) tối đa được chọn làm thuộc tính tách. Tuy nhiên, lưu ý rằng khi thông tin phân tách tiến đến 0, tỷ lệ trở nên không ổn định. Một ràng buộc được thêm vào để tránh điều này, theo đó mức thông tin đạt được (Information gain) của bài kiểm tra được chọn phải lớn nhất ít nhất bằng mức tăng trung bình so với tất cả các bài kiểm tra.

Chỉ số Gini

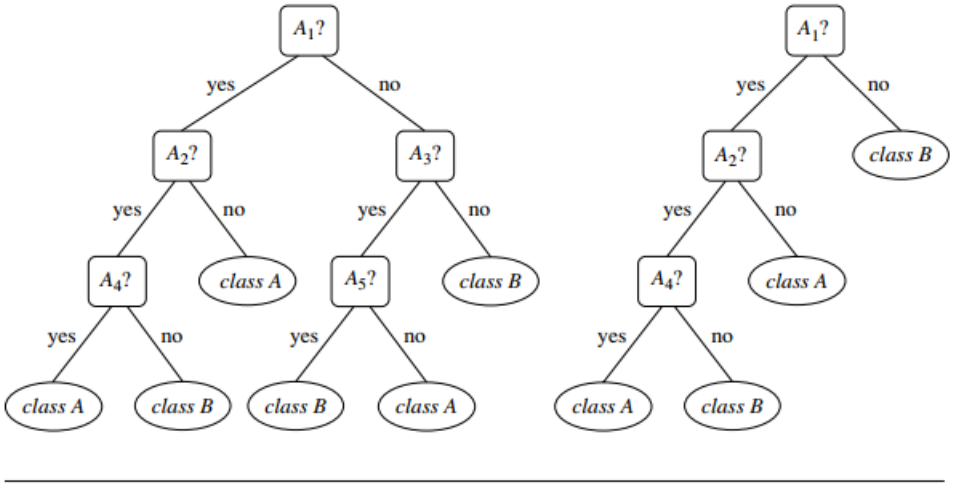
Chỉ số Gini được sử dụng trong CART. Sử dụng ký hiệu được mô tả trước đây, chỉ số Gini đo lường tạp chất của D, phân vùng dữ liệu hoặc bộ bộ dữ liệu đào tạo, như sau:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2.$$

Trong đó p_i là xác suất mà một hàng trong D thuộc về lớp C_i và được ước tính bởi $|C_{i,D}|/|D|$. Tổng được tính trên m lớp.

4.6.3 Cắt tỉa cây

Khi cây quyết định được xây dựng, cây xuất hiện nhiều nhánh sẽ phản ánh sự bất thường trong quá trình huấn luyện dữ liệu do các dữ liệu nhiễu hoặc có các giá trị ngoại lệ. Phương thức cắt tỉa cây giúp giải quyết vấn đề khi cây quyết định tạo ra quá khớp với dữ liệu huấn luyện. Các phương pháp như vậy thường sử dụng các biện pháp thống kê để loại bỏ các nhánh có độ tin cậy thấp nhất. Một cây chưa được cắt tỉa và một phiên bản được cắt tỉa của nó được hiển thị trong Hình 4.10. Cây tỉa có xu hướng nhỏ hơn và ít phức tạp hơn và do đó dễ hiểu hơn. Chúng thường nhanh hơn và tốt hơn trong việc phân lớp chính xác dữ liệu thử nghiệm độc lập (nghĩa là các bộ dữ liệu chưa từng thấy trước đây) so với các cây chưa được xử lý.



Hình 4.10: Cây quyết định khi chưa cắt tỉa và cây quyết định đã cắt tỉa.

"Làm thế nào để thực hiện việc cắt tỉa cây?" Có một cách tiếp cận phổ biến để cắt tỉa cây:*prepruning* và *postpruning*.

Theo cách tiếp cận prepruning, một cây được cắt tỉa bằng cách tạm dừng việc xây dựng cây sớm (ví dụ: bằng cách quyết định không phân tách hoặc phân vùng tập hợp các bộ dữ liệu huấn luyện tại một nút nhất định). Khi dừng lại, nút trở thành một chiếc lá. Chiếc lá có thể giữ lớp thường xuyên nhất trong số các bộ con hoặc phân phối xác suất của các bộ dữ liệu đó. Khi xây dựng một cây, các biện pháp như ý nghĩa thống kê, Information gain, chỉ số Gini,..., có thể được sử dụng để đánh giá mức độ tốt của sự phân chia. Nếu phân vùng các bộ dữ liệu tại một nút sẽ dẫn đến sự phân tách nằm dưới ngưỡng được chỉ định trước, thì việc phân vùng tiếp theo của tập hợp con đã cho bị dừng lại. Tuy nhiên, có những khó khăn trong việc lựa chọn một ngưỡng thích hợp. Ngưỡng cao có thể dẫn đến cây quá đơn giản, trong khi ngưỡng thấp có thể dẫn đến rất ít đơn giản hóa.

Cách tiếp cận thứ hai và phổ biến hơn là postpruning, loại bỏ cây con ra khỏi cây khi đã xây dựng xong. Một cây con tại một nút cho trước được cắt tỉa bằng cách loại bỏ các nhánh của nó và thay thế nó bằng một chiếc lá. Chiếc lá được dán nhãn với lớp thường xuyên nhất trong số các cây con được thay thế. Ví dụ, chú ý đến cây con tại nút "A_3?" trong cây chưa được chỉnh sửa của Hình 4.10. Giả sử rằng lớp phổ biến nhất trong cây con này là lớp B. Trong phiên bản cắt tỉa của cây, cây con trong câu hỏi được cắt tỉa bằng cách thay thế nó bằng lớp lá B.

Ngoài ra, prepruning và postpruning có thể được xen kẽ cho một cách tiếp cận kết hợp . Postpruning đòi hỏi tính toán nhiều hơn prepruning, nhưng nói chung dẫn đến một cây đáng tin cậy hơn. Không có phương pháp cắt tỉa đơn lẻ nào được tìm thấy là vượt trội so với tất cả các phương pháp khác. Mặc dù một số phương pháp cắt tỉa phụ thuộc vào sự sẵn có của dữ liệu bổ sung để cắt tỉa, nhưng điều này thường không phải là vấn đề đáng lo ngại khi xử lý các cơ sở dữ liệu lớn.

4.7 Máy vector hỗ trợ

Trong phần này sẽ nghiên cứu các support vector machines (SVM), một phương pháp để phân lớp cả dữ liệu tuyến tính và phi tuyến. SVM là một thuật toán hoạt động như sau: Nó sử dụng ánh xạ phi tuyến để biến đổi dữ liệu huấn luyện ban đầu vào một chiều cao hơn. Trong kích thước mới này, nó tìm kiếm siêu phẳng phân tách tuyến tối ưu. Với ánh xạ phi tuyến thích hợp đến kích thước đủ cao, dữ liệu từ hai lớp luôn có thể được phân tách bằng một siêu phẳng. SVM tìm thấy siêu phẳng này sử dụng các vectơ hỗ trợ và các lề (được xác định bởi các vectơ hỗ trợ). Bài báo đầu tiên về SVM được trình bày vào năm 1992 bởi Vladimir Vapnik và các đồng nghiệp Bernhard Boser và Isabelle Guyon, mặc dù nền tảng cho SVM đã khoảng từ những năm 1960 (bao gồm tác phẩm ban đầu của Vapnik và Alexei Chervonenkis về lý thuyết thống kê học). Mặc dù thời gian luyện dù nhanh nhất của SVM có thể cực kỳ chậm, chúng có độ chính xác khá cao, nhờ khả năng mô hình hóa các ranh giới quyết định phi tuyến phức tạp. Các vectơ hỗ trợ được tìm thấy cũng cung cấp một mô tả nhỏ gọn về những gì đã luyện mô hình. Các SVM có thể được sử dụng để dự đoán số cũng như trong bài toán phân lớp. Nó có thể được áp dụng cho một số lĩnh vực, bao gồm: nhận dạng chữ số viết tay, đối tượng nhận dạng và nhận dạng người nói, cũng như dự đoán chuỗi thời gian.

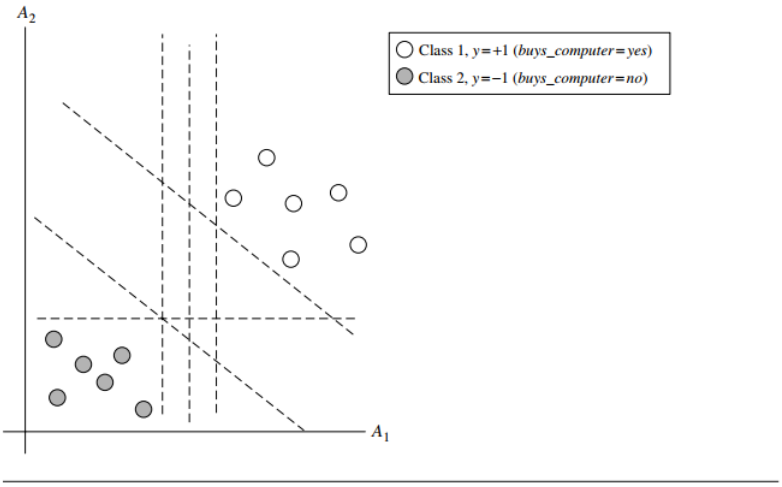
4.7.1 Trường hợp dữ liệu được phân tách tuyến tính

Đặt tập dữ liệu D là $(X_1, y_1), (X_2, y_2), ..., (X_{|D|}, y_{|D|})$, trong đó X_i là tập hợp các bộ dữ liệu luyện liên kết với nhãn y_i. Mỗi y_i có thể lấy một trong hai giá trị, tương ứng $+1$ hoặc -1 (tức là, $y_i \in \{+1, -1\}$), tương ứng với các lớp mua máy tính = có và mua máy tính = không. Để hỗ trợ trực quan hóa, hãy xét một ví dụ dựa trên hai thuộc tính đầu vào là A_1 và A_2, như trong 4.11. Từ đồ thị, ta thấy rằng dữ liệu 2 chiều có thể phân tách tuyến tính (hoặc viết tắt là tuyến tính), bởi vì một đường thẳng có thể được vẽ để tách tất cả các bộ dữ liệu của lớp $+1$ từ tất cả các bộ dữ liệu của lớp -1.

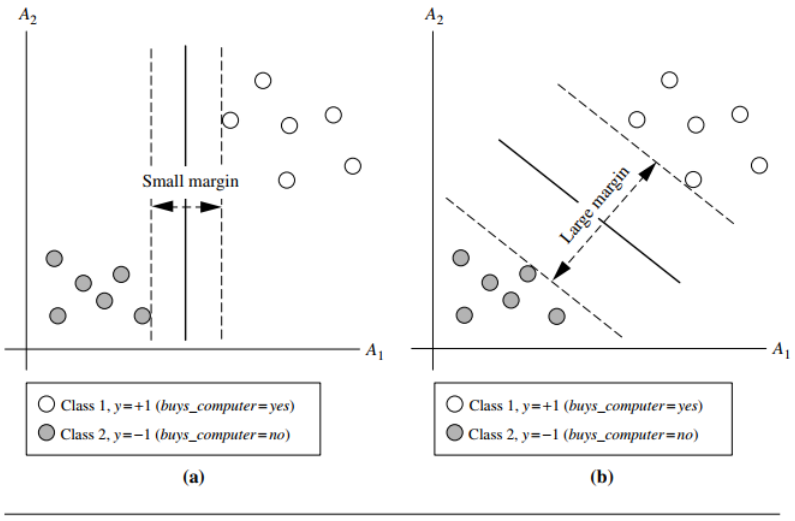
Có vô số đường phân tách có thể được tìm ra. Ta sẽ muốn tìm đường tốt nhất, điều mà ta hi vọng sẽ có ít lỗi trong phân lớp nhất trên bộ dữ liệu mới. Vậy làm thế nào để có thể tìm thấy dòng tốt nhất này? Lưu ý rằng nếu dữ liệu ở đây là 3-D (tức là, với ba thuộc tính), ta muốn tìm mặt phẳng phân tách tốt nhất. Tổng quát hóa đến n kích thước, ta muốn tìm siêu phẳng tốt nhất. Vì vậy, nói một cách tổng quát, làm thế nào ta có thể tìm thấy siêu phẳng tốt nhất? SVM tiếp cận vấn đề này bằng cách tìm kiếm siêu phẳng cận biên tối đa. Xét hình **??**, cho thấy hai siêu phẳng khả thi và lề (margins) của chúng. Ta thấy cả hai siêu phẳng có thể phân lớp chính xác tất cả các bộ dữ liệu đã cho. Tuy nhiên, theo trực giác, ta hy vọng siêu phẳng có lề lớn hơn sẽ mang lại độ chính xác cao hơn tại phân lớp các bộ dữ liệu trong tương lai so với siêu phẳng với lề nhỏ hơn. Đây là lí do tại sao (trong giai đoạn luyện mô hình), SVM sẽ tìm siêu phẳng với lề lớn nhất, nghĩa là siêu phẳng biên tối đa (*maximum marginal hyperplane* - MMH). Liên kết lề cho sự phân tách lớn nhất giữa các lớp.

Một siêu phẳng phân tách có thể được viết dưới dạng:

$$W.X + b = 0$$



Hình 4.11: Dữ liệu 2-D là dữ liệu phân tách tuyến tính.



Hình 4.12: Hai siêu phẳng khả thi và lề (margins) của chúng.

Trong đó W là một vectơ trọng số, cụ thể là, $W = \{w_1, w_2, ..., w_n\}$; n là số lượng thuộc tính; và b là một vô hướng, thường được gọi là một hệ số tự do. Để hỗ trợ trực quan hóa, hãy xem xét hai đầu vào các thuộc tính, A_1 và A_2, như trong 4.12 (b). Các bộ dữ liệu luyện là 2-D (ví dụ: $X = (x_1, x_2)$), trong đó x_1 và x_2 lần lượt là các giá trị của các thuộc tính A_1 và A_2 cho X. Nếu ta nghĩ về b là trọng lượng bổ sung, w_0, chúng ta có thể viết lại phương trình như sau:

$$w_0 + w_1x_1 + w_2x_2 = 0.$$

Do đó, bất kỳ điểm nào nằm trên siêu phẳng phân tách đều thỏa mãn

$$w_0 + w_1x_1 + w_2x_2 > 0$$

Tương tự, bất kỳ điểm nào nằm dưới siêu phẳng phân tách đều thỏa mãn

$$w_0 + w_1x_1 + w_2x_2 < 0$$

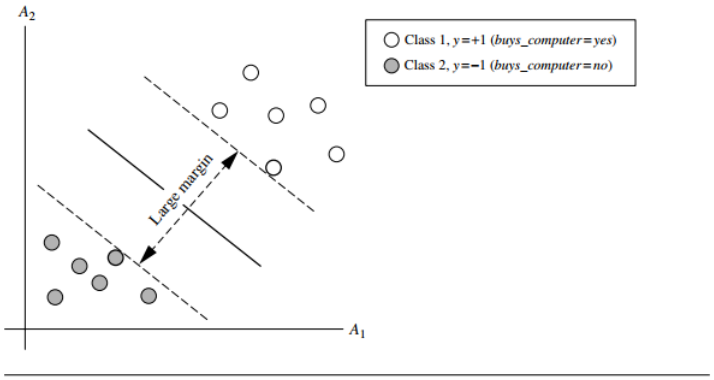
Các trọng số có thể được điều chỉnh sao cho các siêu phẳng xác định các cạnh bên của các lề có thể được viết như sau:

$$H_1 : w_0 + w_1x_1 + w_2x_2 \geq 1 \text{với } y_i = +1$$
$$H_2 : w_0 + w_1x_1 + w_2x_2 \leq 1 \text{với } y_i = -1.$$

Nghĩa là, bất kỳ điểm dữ liệu nào rơi vào hoặc trên H_1 đều thuộc về lớp $+1$ và bất kỳ điểm dữ liệu nào rơi vào hoặc dưới H_2 thuộc lớp -1. Kết hợp hai bất đẳng thức của các phương trình, ta nhận được

$$y_i(w_0 + w_1x_1 + w_2x_2) \geq 1, \forall i$$

Mỗi bộ dữ liệu luyện nằm trong siêu phẳng H_1 và H_2 được gọi là các vector hỗ trợ (support vectors). Trong hình 4.13 các vector hỗ trợ được biểu diễn bởi các điểm tròn có viền dày.



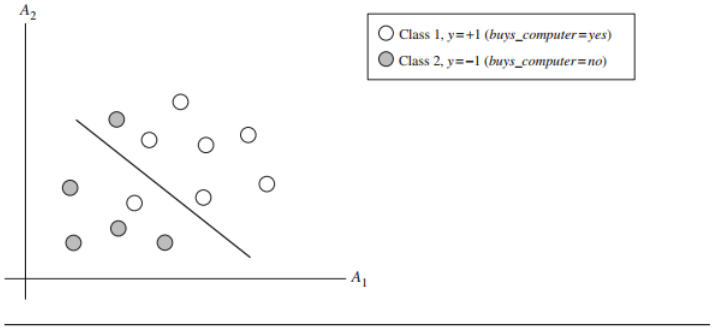
Hình 4.13: Các vector hỗ trợ. SVM tìm thấy siêu phẳng phân tách tối đa, tức là siêu phẳng có khoảng cách tối đa giữa các bộ dữ liệu gần nhất.

Chúng ta có thể tìm được công thức cho kích thước của lề là lớn nhất. Khoảng cách từ siêu phẳng phân tách đến bất kỳ điểm nào trên H_1 là $\frac{1}{||W||}$, trong đó $||W||$ là chuẩn Euclide của W. Theo định nghĩa này, khoảng cách này bằng với khoảng cách từ điểm bất kỳ trên H_2 đến siêu phẳng phân tách. Như vậy độ rộng lớn nhất của lề là $\frac{2}{||W||}$.

"Như vậy, có cách nào để tìm được siêu phẳng biên tối đa và các vector hỗ trợ?" Chúng ta có thể sử dụng các kiến thức về bài toán tối ưu lồi dạng toàn phương, sử dụng công thức Lagrangian và sử dụng điều kiện Karush-Kuhn-Tucker (KKT) để giải quyết bài toán trên.

4.7.2 Trường hợp dữ liệu không thể phân tách tuyến tính

Trong phần trước, ta đã tìm hiểu về SVM tuyến tính để phân lớp dữ liệu có thể phân tách tuyến tính, nhưng điều gì xảy ra nếu dữ liệu không thể phân tách tuyến tính, như trong Hình **??**? Trong trường hợp như vậy, không đường thẳng nào có thể được tách các lớp. Các SVM tuyến tính mà ta đã nghiên cứu sẽ không thể tìm thấy một giải pháp khả thi ở đây. Tin tốt là cách tiếp cận được mô tả cho các SVM tuyến tính có thể được mở rộng đến tạo các SVM phi tuyến để phân lớp dữ liệu tuyến tính không thể tách rời (còn gọi là dữ liệu tách rời phi tuyến hoặc viết tắt là dữ liệu phi tuyến). Những SVM như vậy có khả năng tìm kiếm ranh giới quyết định phi tuyến (tức là, siêu phẳng phi tuyến) trong không gian đầu vào. Vì vậy, bạn có thể hỏi, về cách thức mà ta có thể mở rộng cách tiếp cận tuyến tính? SVM bằng cách mở rộng cách tiếp cận cho các SVM tuyến tính như sau. Có hai bước chính, trong bước đầu tiên, ta chuyển đổi dữ liệu đầu vào ban đầu sang không gian nhiều chiều hơn sử dụng ánh xạ phi tuyến. Một số ánh xạ phi tuyến phổ biến có thể được sử dụng trong này bước, như ta sẽ mô tả thêm tiếp theo. Khi dữ liệu đã được chuyển đổi thành không gian mới cao hơn, bước thứ hai tìm kiếm một siêu phẳng tách tuyến tính trong mới không gian. Ta lại kết thúc với một vấn đề tối ưu hóa bậc hai có thể được giải quyết bằng cách sử dụng công thức SVM tuyến tính. Siêu phẳng cận biên tối đa được tìm thấy trong không gian mới tương ứng với một siêu mặt phân cách phi tuyến trong không gian ban đầu.

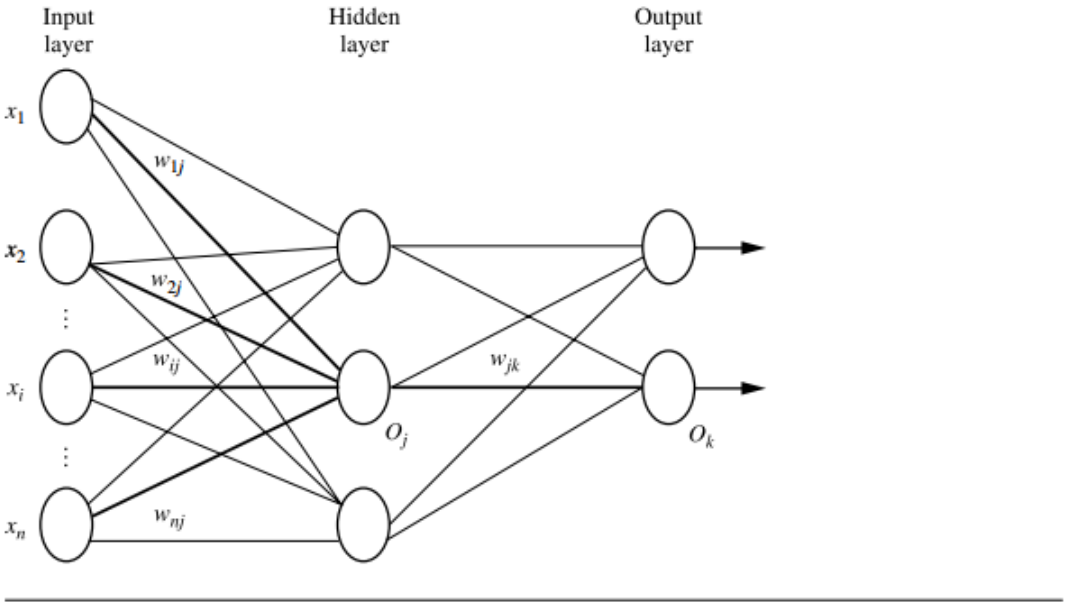


Hình 4.14: Một ví dụ 2-D đơn giản trường hợp dữ liệu không thể phân tách tuyến tính.

4.8 Mạng Neural

4.8.1 Mạng neural đa lớp

Thuật toán lan truyền ngược thực hiện việc học trên một mạng neural đa lớp. Nó học lặp đi lặp lại một tập hợp các trọng số để dự đoán nhãn lớp của các bộ giá trị. Một mạng neural đa lớp bao gồm một lớp đầu vào, một hoặc nhiều lớp ẩn và một lớp đầu ra. Ví dụ về mạng neural chuyển tiếp nhiều lớp được hiển thị trong hình **??**.



Hình 4.15: Cấu trúc của một mạng neural đa lớp.

Mỗi lớp được tạo thành từ các đơn vị (biểu diễn bởi các node hình tròn). Các đầu vào cho mạng tương ứng với các thuộc tính được đo cho từng bộ đào tạo. Các đầu vào được cấp đồng thời vào các thiết bị tạo nên lớp đầu vào. Các đầu vào này đi qua lớp đầu vào và sau đó được tăng trọng lượng và được cung cấp đồng thời cho lớp thứ hai của các đơn vị "giống neural", được gọi là lớp ẩn. Đầu ra của các đơn vị lớp ẩn có thể được đưa vào một lớp ẩn khác. Số lượng các lớp ẩn là tùy ý, mặc dù trong thực tế, thường chỉ một lớp ẩn được sử dụng. Các đầu ra có trọng số của lớp ẩn cuối cùng là đầu vào cho các đơn vị tạo lên lớp đầu ra, phát ra dự đoán của mạng cho các bộ giá trị nhất định.

Các đơn vị trong lớp đầu vào được gọi là đơn vị đầu vào. Các đơn vị trong các lớp ẩn và lớp đầu ra đôi khi được gọi là tế bào thần kinh, do biểu tượng sinh học của chúng cơ sở, hoặc dưới dạng đơn vị đầu ra. Mạng neural đa lớp được thể hiện trong Hình 4.15 có hai các lớp đơn vị đầu ra. Do đó, chúng ta nói rằng đó là một mạng neural hai lớp. (Các lớp đầu vào không được tính vì nó chỉ dùng để chuyển các giá trị đầu vào cho lớp tiếp theo lớp.) Tương tự, một mạng chứa hai lớp ẩn được gọi là mạng neural ba lớp . Nó là một mạng chuyển tiếp vì không có trọng số nào quay trở lại cho một

đơn vị đầu vào hoặc cho một đơn vị đầu ra của lớp trước đó. Nó được kết nối đầy đủ trong đó mỗi đơn vị cung cấp đầu vào cho mỗi đơn vị trong lớp chuyển tiếp tiếp theo.

Mỗi đơn vị đầu ra lấy làm đầu vào và tính tổng trọng số của các đầu ra từ các đơn vị trong lớp trước (xem Hình 4.15). Nó áp dụng một hàm phi tuyến (kích hoạt) cho đầu vào có trọng số. Các mạng neural truyền tới nhiều lớp có thể lập mô hình dự đoán lớp dưới dạng kết hợp phi tuyến của các đầu vào. Từ quan điểm thống kê, họ thực hiện hồi quy phi tuyến. Mạng chuyển tiếp nguồn cấp dữ liệu nhiều lớp, đã đủ ẩn đơn vị và đủ các mẫu đào tạo, có thể gần đúng với bất kỳ chức năng nào.

4.8.2 Định nghĩa cấu trúc mạng

"Làm cách nào để thiết kế cấu trúc liên kết của mạng nơ-ron?" Trước khi luyện dữ liệu, người dùng phải quyết định về cấu trúc mạng bằng cách chỉ định số lượng đơn vị trong đầu vào lớp, số lớp ẩn (nếu nhiều hơn một), số đơn vị trong mỗi lớp lớp ẩn và số lượng đơn vị trong lớp đầu ra.

Chuẩn hóa các giá trị đầu vào cho từng thuộc tính được đo trong dữ liệu luyện sẽ giúp tăng tốc giai đoạn học tập. Thông thường, các giá trị đầu vào được chuẩn hóa để từ 0.0 đến 1.0. Các thuộc tính có giá trị rời rạc có thể được mã hóa để có một đơn vị đầu vào cho mỗi giá trị miền. Ví dụ: nếu một thuộc tính A có ba giá trị, cụ thể là $\{a_0, a_1, a_2\}$ thì chúng ta có thể gán ba đơn vị đầu vào để đại diện cho A, gọi là I_0, I_1, I_2. Mỗi đơn vị được khởi tạo là 0. Nếu $A = a_0$ thì $I_0 = 1$ và phần còn lại gán giá trị 0.

Mạng neural có thể được sử dụng cho cả hai phân loại (để dự đoán nhãn lớp của một bộ đã cho) và dự đoán số (để dự đoán đầu ra có giá trị liên tục). Để phân loại, một đơn vị đầu ra có thể được sử dụng để đại diện cho hai lớp (trong đó giá trị 1 đại diện cho một lớp và giá trị 0 đại diện cho lớp kia). Nếu có nhiều hơn hai các lớp, một đơn vị đầu ra đại diện mỗi lớp được sử dụng.

Không có quy tắc rõ ràng nào về số lượng đơn vị lớp ẩn "tốt nhất". Thiết kế mạng là một quá trình thử và sai và có thể ảnh hưởng đến độ chính xác của mạng được đào tạo. Giá trị ban đầu của các trọng số cũng có thể ảnh hưởng đến độ chính xác kết quả. Một lần mạng đã được đào tạo và độ chính xác của nó không được coi là chấp nhận được, nó thường là lặp lại quá trình đào tạo với một cấu trúc liên kết mạng khác hoặc một tập hợp ban đầu khác trọng lượng. Các kỹ thuật để ước tính độ chính xác có thể được sử dụng để giúp quyết định khi nào một mạng được chấp nhận đã được tìm thấy. Một số các kỹ thuật tự động đã được đề xuất để tìm kiếm cấu trúc mạng "tốt". Chúng thường sử dụng phương pháp leo đồi bắt đầu với cấu trúc ban đầu là sửa đổi có chọn lọc.

4.8.3 Lan truyền ngược

"Lan truyền ngược hoạt động như thế nào?" Lan truyền ngược học bằng cách xử lý lặp đi lặp lại một tập dữ liệu gồm các bộ đào tạo, so sánh dự đoán của mạng cho từng bộ với giá trị mục tiêu thực tế đã biết. Giá trị đích có thể là nhãn lớp đã biết của quá trình đào tạo (cho các bài toán phân loại) hoặc một giá trị liên tục (cho dự đoán số). Đối với mỗi bộ đào tạo, trọng số được sửa đổi để giảm thiểu sai số trung bình giữa dự đoán của mạng và giá trị mục tiêu thực tế. Những sửa đổi này là được thực hiện theo hướng "ngược" (tức là từ lớp đầu ra) qua mỗi ẩn lớp xuống lớp ẩn đầu tiên (do đó có tên là backpropagation). Mặc dù chưa chắc chắn nhưng nói chung các trọng số cuối cùng sẽ hội tụ và quá trình học dừng lại. Thuật toán được tóm tắt trong 4.16. Các bước liên quan được thể hiện trong điều khoản đầu vào, đầu ra và lỗi, và có thể có vẻ khó hiểu nếu đây là lần đầu tiên bạn xem mạng neural học. Tuy nhiên, khi bạn đã quen với quá trình này, bạn sẽ thấy rằng mỗi bước vốn dĩ rất đơn giản. Các bước được mô tả tiếp theo.

Thuật toán lan truyền ngược Mạng neural để phân loại hoặc dự đoán số sử dụng thuật toán lan truyền ngược.

Input:

- Tập dữ liệu bao gồm các bộ dữ liệu huấn luyện và các giá trị mục tiêu liên quan của chúng;
- l, learning rate;
- Mạng neural đa lớp.

Output: Mạng neural được đào tạo.

Khởi tạo trọng số: Các trọng số trong mạng được khởi tạo thành các số ngẫu nhiên nhỏ (ví dụ: nằm trong khoảng từ $-1,0$ đến $1,0$ hoặc $-0,5$ đến $0,5$). Mỗi đơn vị có một bias liên quan đến nó. Các bias được khởi tạo tương tự cho các số ngẫu nhiên nhỏ.

Mỗi bộ đào tạo, X, được xử lý theo các bước sau.

Truyền các đầu vào về phía trước: Đầu tiên, bộ dữ liệu luyện được đưa vào đầu vào của mạng lớp. Các đầu vào đi qua các đơn vị đầu vào, không thay đổi. Đối với một đơn vị đầu vào, j, đầu ra của nó, O_j, là đầu vào để tính toán I_j. Tiếp theo, đầu vào và đầu ra của mỗi đơn vị trong lớp ẩn và lớp đầu ra được tính toán. Đầu vào cho một đơn vị trong ẩn hoặc các lớp đầu ra được tính toán như một sự kết hợp tuyến tính của các đầu vào của nó. Để giúp minh họa điều này điểm, một lớp ẩn hoặc đơn vị lớp đầu ra được thể hiện trong Hình **??**. Mỗi đơn vị như vậy có một số đầu vào cho nó, trên thực tế, là đầu ra của các đơn vị được kết nối với nó trong lớp trước đó. Mỗi kết nối có một trọng số. Để tính toán đầu vào ròng cho thiết bị, mỗi đầu vào được kết nối với thiết bị được nhân

(1) Initialize all weights and biases in *network*;
(2) **while** terminating condition is not satisfied {
(3) **for** each training tuple $\boldsymbol{X}$ in D {
(4) // Propagate the inputs forward:
(5) **for** each input layer unit j {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) **for** each hidden or output layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10) // Backpropagate the errors:
(11) **for** each unit j in the output layer
(12) $Err_j = O_j(1 - O_j)(T_j - O_j)$; // compute the error
(13) **for** each unit j in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k
(15) **for** each weight w_{ij} in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) **for** each bias θ_j in *network* {
(19) $\Delta \theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
(21) } }

Hình 4.16: Thuật toán lan truyền ngược.

với trọng số tương ứng của nó và đây là tổng. Cho một đơn vị, j trong lớp ẩn hoặc lớp đầu ra tính toán I_j, cho đơn vị j là

$$I_j = \sum_i w_{ij} O_i + \theta_j,$$

trong đó w_{ij} là trọng số của kết nối từ đơn vị i ở lớp trước đến đơn vị j; O_i là đầu ra của đơn vị i từ lớp trước; và θ_j là bias của đơn vị. Các bias như một ngưỡng trong đó nó phục vụ để thay đổi hoạt động của đơn vị.

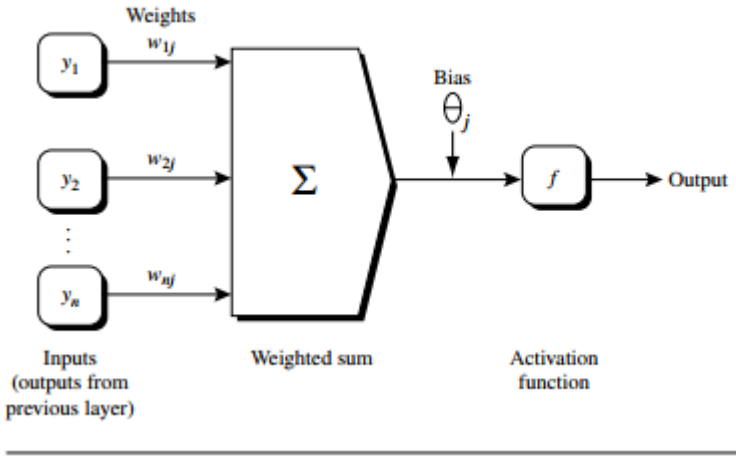
Mỗi đơn vị trong lớp ẩn và lớp đầu ra nhận đầu vào của nó và sau đó áp dụng một hàm kích hoạt cho nó, như được minh họa trong Hình **??** Hàm tượng trưng cho sự kích hoạt của neural được đại diện bởi đơn vị. Hàm logistic, hoặc sigmoid, được sử dụng. Được đầu vào thuần I_j đến đơn vị j, sau đó O_j, đầu ra của đơn vị j, được tính là

$$O_j = \frac{1}{1 + e^{-I_j}}$$

Hàm này còn được gọi là hàm thu nhỏ vì nó ánh xạ một đầu vào lớn miền vào phạm vi nhỏ hơn từ 0 đến 1. Hàm logistic là phi tuyến và có thể phân biệt, cho phép thuật toán lan truyền ngược mô hình hóa các vấn đề phân loại mà tuyến tính không thể tách rời.

Chúng ta tính toán các giá trị đầu ra, O_j, cho mỗi lớp ẩn, lên đến và bao gồm lớp đầu ra, đưa ra dự đoán của mạng. Trong thực tế, bạn nên cache (tức là lưu)

các giá trị đầu ra trung gian ở mỗi đơn vị khi chúng được yêu cầu lại sau khi lan truyền ngược lỗi. Thủ thuật này có thể làm giảm đáng kể lượng yêu cầu tính toán.



Hình 4.17

Sai số của lan truyền ngược: Lỗi được được lan truyền ngược tính toán bằng cách cập nhật các trọng số và bias để phản ánh lỗi dự đoán của mạng. Đối với một đơn vị j trong đầu ra lớp, lỗi Err_j được tính toán bởi

$$Err_j = O_j(1 - O_j)(T_j - O_j)$$

trong đó O_j là đầu ra thực tế của đơn vị j và T_j là giá trị mục tiêu đã biết của dữ liệu luyện.

Sai số của một lớp ẩn đơn vị j là

$$Err_j = O_j(1 - O_j) \sum_k Err_k w_j k,$$

trong đó w_{jk} là trọng số kết nối từ đơn vị i đến đơn vị k trong lớp ẩn tiếp theo và Err_k là sai số của đơn vị k.

Các trọng số và bias được cập nhật để phản ánh các sai số được truyền. Trọng số được cập nhật bằng các phương trình sau, trong đó w_{ij} là sự thay đổi trọng số w_{ij}

$$\Delta w_{ij} = (l) Err_j O_i.$$

$$w_{ij} = w_{ij} + \Delta w_{ij}.$$

trong đó l là tốc độ học (learning rate), là một hằng số thường có giá trị từ 0.0 đến 1.0. Lan truyền ngược tối ưu các trọng số bằng cách sử dụng thuật toán gradient descent.

Các bias được cập nhập theo phương trình sau, trong đó $\Delta\theta_j$ là sự thay đổi của bias θ_j:

$$\Delta\theta_j = (l)Err_j.$$
$$\theta_j = \theta_j + \Delta\theta_j.$$

Điều kiện dừng:

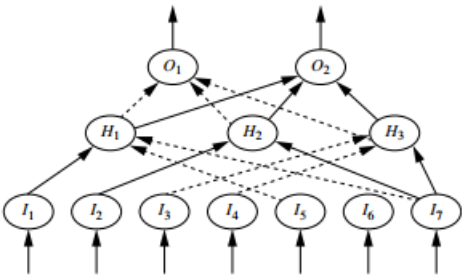
- Tất cả Δw_{ij} trong vòng lặp trước đều nhỏ đến mức thấp hơn một số quy định.
- Tỷ lệ phần trăm bộ giá bị phân loại sai trong vòng lặp trước là dưới một số ngưỡng cho trước.
- Lặp hết các vòng lặp.

"Hiệu suất của lan truyền ngược thế nào?" Hiệu quả tính toán phụ thuộc vào dành thời gian đào tạo mạng lưới. Cho trước $|D|$ bộ dữ liệu và trọng số w, mỗi vòng lặp yêu cầu thời gian $O(|D| \times w)$. Tuy nhiên, trong trường hợp xấu nhất, có thể là số mũ theo n, số lượng đầu vào. Trên thực tế, thời gian cần thiết cho các mạng để hội tụ là rất khác nhau.

4.8.4 Bên trong Hộp đen: Sự lan truyền ngược và khả năng diễn giải

"Mạng neural giống như một hộp đen. Làm cách nào để có thể 'hiểu' sự lan truyền ngược mạng đã học?" Một nhược điểm lớn của mạng neural nằm ở kiến thức của chúng sự đại diện. Kiến thức thu nhận được dưới dạng một mạng lưới các đơn vị được kết nối bởi các liên kết có trọng số rất khó đối với con người để giải thích. Yếu tố này đã thúc đẩy nghiên cứu trong trích xuất kiến thức được nhúng trong mạng nơ-ron được đào tạo và biểu diễn kiến thức một cách tượng trưng. Các phương pháp bao gồm trích xuất các quy tắc từ mạng và độ nhạy phân tích.

Các thuật toán khác nhau để trích xuất quy tắc đã được đề xuất. Các phương pháp thường áp đặt các hạn chế liên quan đến các thủ tục được sử dụng trong đào tạo mạng nơ-ron đã cho, cấu trúc liên kết mạng và sự tùy ý của các giá trị đầu vào. Các mạng được kết nối đầy đủ rất khó để hiểu rõ ràng. Do đó, thường là bước đầu tiên trong trích xuất các quy tắc từ mạng nơ-ron là việc cắt tỉa mạng. Điều này bao gồm việc đơn giản hóa cấu trúc mạng bằng cách loại bỏ các liên kết có trọng số ít ảnh hưởng nhất đến mạng lưới. Ví dụ: một liên kết có trọng số có thể bị xóa nếu việc xóa như vậy không dẫn đến giảm độ chính xác phân loại của mạng.



Identify sets of common activation values for each hidden node, H_i:
for H_1: (−1,0,1)
for H_2: (0,1)
for H_3: (−1,0.24,1)

Derive rules relating common activation values with output nodes, O_j:
IF (H_2=0 AND H_3=−1) OR
(H_1=−1 AND H_2=1 AND H_3=−1) OR
(H_1=−1 AND H_2=0 AND H_3=0.24)
THEN O_1=1, O_2=0
ELSE O_1=0, O_2=1

Derive rules relating input nodes, I_j, to output nodes, O_j:
IF (I_2=0 AND I_7=0) THEN H_2=0
IF (I_4=1 AND I_6=1) THEN H_3=−1
IF (I_5=0) THEN H_3=−1

Obtain rules relating inputs and output classes:
IF (I_2=0 AND I_7=0 AND I_4=1 AND I_6=1) THEN class=1
IF (I_2=0 AND I_7=0 AND I_5=0) THEN class=1

Hình 4.18

Sau khi mạng được đào tạo đã được lược bớt, một số phương pháp tiếp cận sau đó sẽ thực hiện liên kết, đơn vị hoặc cụm giá trị kích hoạt. Trong một phương pháp, ví dụ,

phân cụm được sử dụng để tìm tập hợp các giá trị kích hoạt chung cho mỗi đơn vị ẩn trong một mạng nơ-ron hai lớp đã được huấn luyện nhất định (Hình 4.18). Sự kết hợp của các giá trị kích hoạt này cho mỗi đơn vị ẩn được phân tích. Các quy tắc bắt nguồn từ sự kết hợp liên quan của các giá trị kích hoạtvới các giá trị đơn vị đầu ra tương ứng. Tương tự, các bộ giá trị đầu vào và kích hoạt các giá trị được nghiên cứu để rút ra các quy tắc mô tả mối quan hệ giữa lớp đầu vào và "đơn vị lớp" ẩn? Cuối cùng, hai bộ quy tắc có thể được kết hợp để tạo thành Quy tắc IF-THEN. Các thuật toán khác có thể rút ra các quy tắc của các dạng khác, bao gồm M-of-N quy tắc (trong đó M trong số N điều kiện nhất định trong tiền đề quy tắc phải đúng với kết quả là quy tắc được áp dụng), cây quyết định với thử nghiệm M-of-N, lý thuyết mờ và automat hữu hạn.

Phân tích độ nhạy được sử dụng để đánh giá tác động của một biến đầu vào nhất định đối với đầu ra mạng. Đầu vào cho biến là khác nhau trong khi các biến đầu vào còn lại được cố định ở một số giá trị. Trong khi đó, các thay đổi trong đầu ra mạng được giám sát. Các kiến thức thu được từ biểu mẫu phân tích này có thể được biểu diễn trong các quy tắc như "IF X giảm 5% THÌ Y tăng 8%".

Phần III: Phân tích chuỗi thời gian

Mô hình chuỗi thời gian là một lĩnh vực nghiên cứu năng động đã thu hút sự quan tâm của cộng đồng các nhà nghiên cứu trong vài thập kỷ qua. Mục đích chính của mô hình chuỗi thời gian là thu thập cẩn thận và nghiên cứu chặt chẽ các quan sát trong quá khứ của chuỗi thời gian để phát triển một mô hình thích hợp mô tả cấu trúc vốn có của chuỗi. Mô hình này sau đó được sử dụng để tạo ra các giá trị trong tương lai cho chuỗi, tức là để đưa ra dự báo. Do đó, dự báo chuỗi thời gian có thể được gọi là hành động dự đoán tương lai bằng cách hiểu quá khứ. Do tầm quan trọng không thể thiếu của dự báo chuỗi thời gian trong nhiều lĩnh vực thực tế như kinh doanh, kinh tế, tài chính, khoa học và kỹ thuật, cần chú ý phù hợp để đưa một mô hình phù hợp vào chuỗi thời gian cơ bản . Rõ ràng là một dự báo chuỗi thời gian thành công phụ thuộc vào một mô hình phù hợp. Các nhà nghiên cứu đã nỗ lực rất nhiều trong nhiều năm để phát triển các mô hình hiệu quả nhằm cải thiện độ chính xác của dự báo. Nhiều mô hình phổ biến đã được báo cáo trong tài liệu để cải thiện độ chính xác và hiệu quả của mô hình và dự báo chuỗi thời gian. Mục đích của báo cáo này là trình bày mô tả ngắn gọn về một số mô hình dự báo chuỗi thời gian phổ biến được sử dụng trong thực tế, với các đặc điểm nổi bật của chúng. Trong báo cáo này, em sẽ trình bày về các yếu tố của chuỗi thời gian, quá trình dừng và một số mô hình dự báo chuối thời gian như Hot-Winner, ARIMA, SARIMA, SARIMAX và mô hình GRACH.

4.9 Phân tích chuỗi thời gian

4.9.1 Các yếu tố của chuỗi thời gian

Khái niệm chuỗi thời gian

Chuỗi thời gian là một quá trình ngẫu nhiên $\{X_t,\ t \in \mathrm{T}\}$ phụ thuộc theo biến thời gian $t \in \mathrm{T}$ được biểu thị qua dãy các quan sát. Chuỗi thời gian có thế là liên tục hoặc rời rạc phụ thuộc vào tập T là đếm được hay không đếm được. Chuỗi thời gian là rời rạc nếu các quan sát được thực hiện trên tập thời gian T đếm được, ngược là chuỗi thời gian liên tục. Các ứng dụng thực tế thường sử dụng các chuỗi thời gian rời rạc với khoảng thời gian cách đều (phút, giờ, ngày, tuần, tháng, quý, năm,...). Ví dụ, các chỉ số nhiệt độ, dòng chảy của sông, nồng độ của một quá trình hóa học, v.v. có thể được ghi lại thành một chuỗi thời gian liên tục. Mặt khác, dân số của một thành phố cụ thể, sản xuất của một công ty, tỷ giá hối đoái giữa hai loại tiền tệ khác nhau có thể đại diện cho chuỗi thời gian rời rạc. Dữ liệu được quan sát trong một chuỗi thời gian rời rạc được giả định là được đo như một biến liên tục bằng cách sử dụng thang số thực. Hơn nữa, một chuỗi thời gian liên tục có thể dễ dàng chuyển đổi thành chuỗi thời gian rời rạc bằng cách hợp nhất dữ liệu với nhau trong một khoảng thời gian xác định.

Một chuỗi thời gian ghi lại giá trị của một biến đơn thì được gọi gọi là chuỗi thời gian đơn chiều. Nhưng nếu chuỗi thời gian ghi lại giá trị của nhiều hơn một biến thì nó là chuỗi thời gian đa chiều.

Các thành phần của chuỗi thời gian

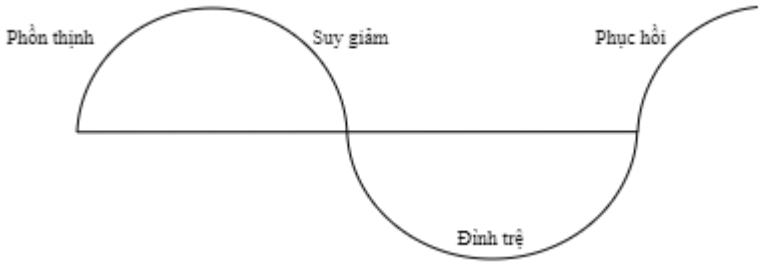
Một chuỗi thời gian tổng quát thường bao gồm bốn thành phần chính, thứ mà có thể chia tách từ dữ liệu đã được quan sát. Các thành phần này có thể được liệt kê như sau: thành phần xu thế, thành phần chu kì, thành phần mùa và thành phần ngẫu nhiên. Trong mục này, chúng ta sẽ mô tả vắn tắt về bốn thành phần này.

Xu hướng chung của một chuỗi thời gian tăng, giảm hoặc trì trệ trong một thời gian dài được gọi là xu thế của chuỗi thời gian. Như vậy, có thể nói xu thế là sự vận động dài hạn trong một chuỗi thời gian. Ví dụ, chuỗi liên quan đến sự gia tăng dân số, số nhà trong một thành phố, v.v. cho thấy xu hướng tăng, trong khi xu hướng giảm có thể được quan sát trong chuỗi liên quan đến tỷ lệ tử vong, dịch bệnh, v.v.

Sự thay đổi theo mùa trong một chuỗi thời gian là những biến động trong một năm của chuỗi theo mùa. Các yếu tố quan trọng gây ra sự biến đổi theo mùa là: điều kiện thời tiết khí hậu, phong tục, tập quán truyền thống, v.v. Ví dụ doanh số bán kem tăng vào mùa hè, doanh số bán hàng vải len tăng vào mùa đông. Sự thay đổi theo mùa là một yếu tố quan trọng đối với các nhà kinh doanh, chủ cửa hàng và nhà sản xuất để lập kế hoạch tương lai đúng đắn.

Sự thay đổi theo chu kỳ trong chuỗi thời gian mô tả những thay đổi trung hạn trong chuỗi do hoàn cảnh gây ra, những thay đổi này lặp lại theo chu kỳ. Thời gian của một chu kỳ kéo dài trong một khoảng thời gian dài hơn, thường là hai năm trở lên. Hầu hết các chuỗi thời gian kinh tế và tài chính cho thấy một số loại biến đổi theo chu kỳ. Ví dụ, một chu kỳ kinh doanh bao gồm bốn giai đoạn:

- Phát triển phồn tịnh
- Suy giảm doanh thu
- Đình trệ
- Phục hồi



Hình 4.19: Bốn pha của chu kì kinh doanh

Ví dụ về chuỗi thời gian

Các thay đổi bất thường hoặc ngẫu nhiên trong một chuỗi thời gian là do ảnh hưởng không thể đoán trước được, không thường xuyên và cũng không lặp lại theo một mô hình cụ thể. Các thành phần này xuất hiện là do các sự cố như chiến tranh, đình công, động đất, lũ lụt, cách mạng, v.v. Không có kỹ thuật thống kê xác định để đo các thành phần ngẫu nhiên trong một chuỗi thời gian.

Xem xét ảnh hưởng của bốn thành phần này, hai loại mô hình khác nhau thường được sử dụng cho một chuỗi thời gian. Mô hình cộng và mô hình nhân.

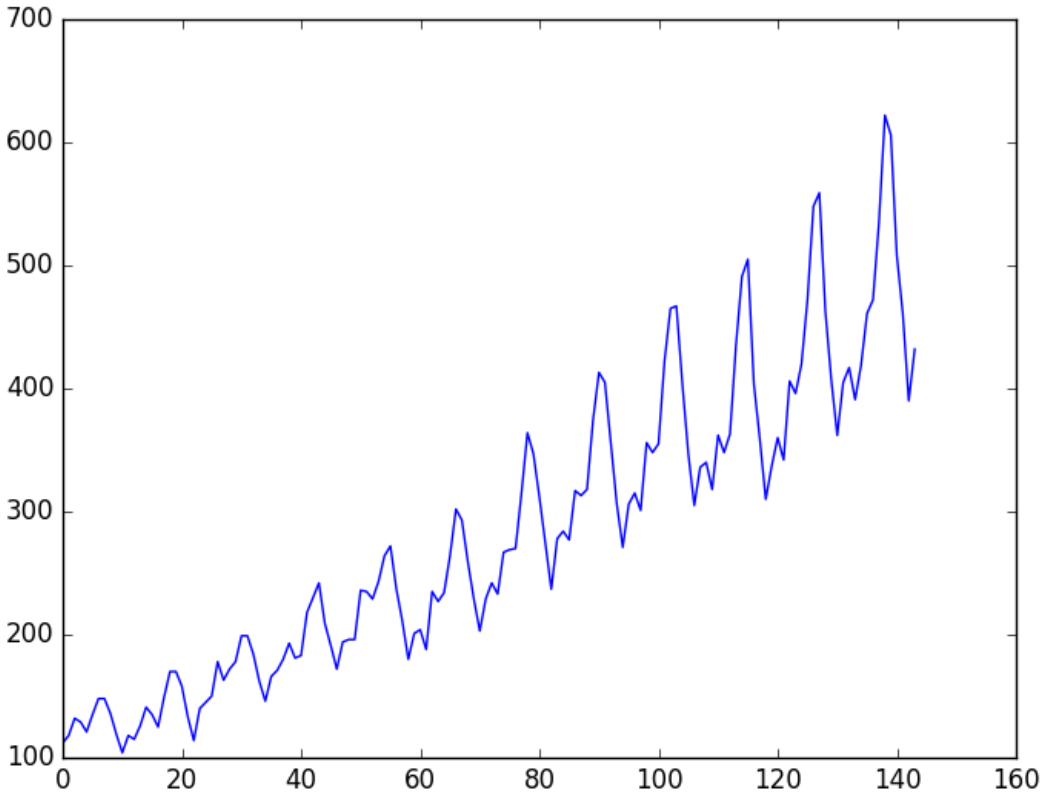
Mô hình nhân: $Y(t) = T(t) \times S(t) \times C(t) \times I(t)$

Mô hình cộng: $Y(t) = T(t) + S(t) + C(t) + I(t)$

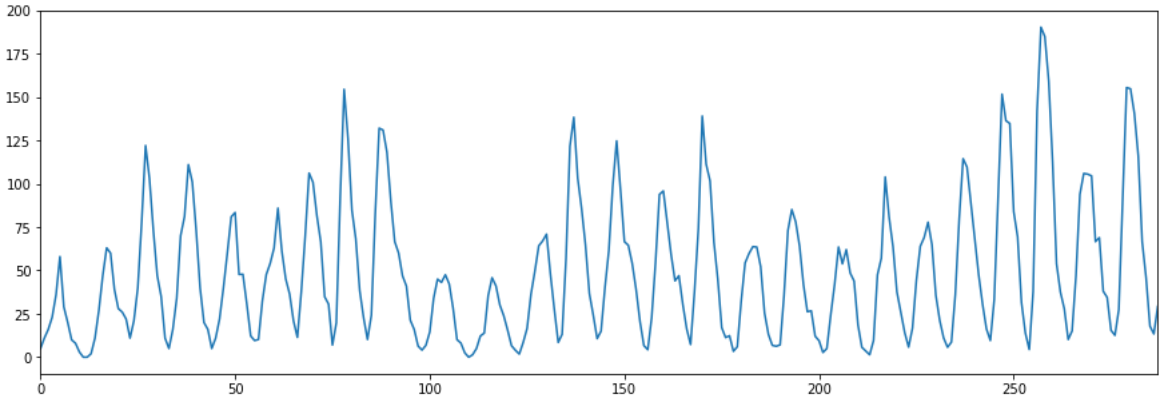
Trong 2 công thức trên, $Y(t)$ là dữ liệu được quan sát và $T(t)$, $S(t)$, $C(t)$, $I(t)$ tương ứng là thành phần xu thế, thành phần mùa, thành phần chu kì, thành phần ngẫu nhiên tại thời điểm t.

Mô hình nhân dựa trên giả sử rằng bốn thành phần của chuỗi thời gian không độc lập và chúng có sự ảnh hưởng lẫn nhau. Trái lại, mô hình cộng giả sử rằng bốn thành phần này độc lập với nhau.

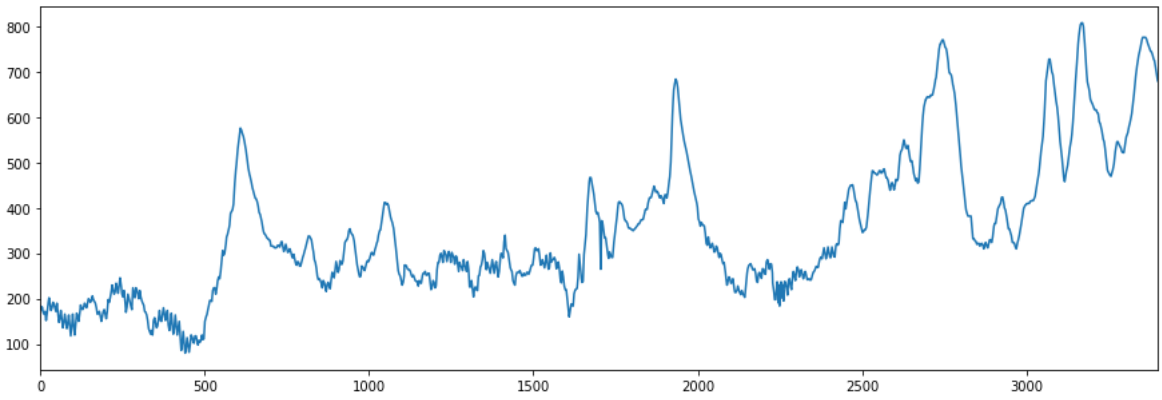
Các quan sát chuỗi thời gian thường gặp trong nhiều lĩnh vực như kinh doanh, kinh tế, công nghiệp, kỹ thuật và khoa học,... Tùy theo tính chất của phân tích và nhu cầu thực tế, có thể có nhiều loại chuỗi thời gian khác nhau. Để hình dung khuôn mẫu cơ bản của dữ liệu, thường một chuỗi thời gian được biểu diễn bằng biểu đồ, trong đó các quan sát được vẽ theo thời gian tương ứng. Dưới đây là 3 chuỗi thời gian được biểu diễn trên biểu đồ:



Hình 4.20: Chuỗi dữ liệu hành khách quốc tế



Hình 4.21: Dữ liệu số lượng các điểm đen mặt trời



Hình 4.22: Dữ liệu mực nước trên sông Hồng vào mùa mưai

Chuỗi thời gian thứ nhất là chuỗi thời gian có tính mùa ghi lại số lượng các hành khách quốc tế hàng tháng từ tháng 1 năm 1949 đến tháng 12 năm 1960. Chuỗi thời gian thứ hai là số lượng các điểm đen của mặt trời được tập hợp từ năm 1700 đến năm 1987. Bộ dữ liệu này còn được gọi với tên Wolf's Sunspot. Các nghiên cứu về điểm đen mặt trời có ý nghĩa thực tiễn trong các lĩnh vực địa vật lý, khoa học môi trường, khí tượng học. Chuỗi dữ liệu được coi là phi tuyến và không Gaussian và thường được sử dụng để đánh giá sự hiệu quả của các mô hình phi tuyến. Bộ dữ liệu thứ 3 trong hình số 4 gồm 3400 điểm dữ liệu mực nước trên sông Hồng được ghi lại cách nhau 2 giờ trong 3 mùa mưa của năm 2015, 2016, 2017. Mùa mưa diễn ra từ tháng 6 đến tháng 9 hàng năm.

Giới thiệu về phân tích chuỗi thời gian

Trong thực tế, một mô hình phù hợp được khớp vào một chuỗi thời gian đã có và các tham số tương ứng của mô hình được ước lượng bằng cách sử dụng các giá trị dữ liệu đã được thu thập từ quá khứ. Thủ tục khớp chuỗi thời gian với một mô hình thích hợp được gọi là phân tích chuỗi thời gian. Nó bao gồm các phương pháp cố gắng hiểu tính chất của chuỗi và thường hữu ích cho việc dự báo và mô phỏng trong tương lai. Trong dự báo chuỗi thời gian, các quan sát trong quá khứ được thu thập và phân tích để phát triển một mô hình toán học phù hợp nắm bắt quá trình tạo dữ liệu cơ bản cho chuỗi. Các sự kiện trong tương lai được dữ báo sử dụng mô hình đã được khớp với dữ liệu trong quá khứ. Cách tiếp cận này đặc biệt hữu ích khi không có nhiều kiến thức về mẫu thống kê của các quan sát trong quá khứ hoặc khi thiếu một mô hình giải thích thỏa đáng. Dự báo chuỗi thời gian có các ứng dụng quan trọng trong đã dạng lĩnh vực. Thông thường các quyết định chiến lược có giá trị và các biện pháp phòng ngừa được thực hiện dựa trên kết quả dự báo. Vì vậy để có một kết quả dự báo tốt thì tìm ra một mô hình phù hợp với chuỗi dữ liệu là hết sức quan trọng. Trong nhiều thập kỷ qua, các nhà nghiên cứu đã có nhiều nỗ lực để phát triển và cải tiến các mô hình dự báo chuỗi thời gian phù hợp.

4.9.2 Quá trình dừng

Chuỗi thời gian và quá trình ngẫu nhiên

Một chuỗi thời gian có bản chất không xác định, tức là chúng ta không thể dự đoán chắc chắn điều gì sẽ xảy ra trong tương lai. Nói chung, chuỗi thời gian $x(t)$, $t = 0, 1, 2$ được giả định tuân theo một số mô hình xác suất nào đó, mô tả phân phối đồng nhất của biến ngẫu nhiên $x(t)$. Biểu thức toán học mô tả cấu trúc xác suất của một chuỗi thời gian được gọi là một quá trình ngẫu nhiên. Do đó, chuỗi các quan sát trong thực tế được tạo bởi một quá trình ngẫu nhiên. Một giả định thông thường là các biến chuỗi thời gian $x(t)$ là độc lập và được phân phối giống nhau (i.i.d) tuân theo luật phân phối chuẩn. Nhưng trong thực tế, các chuỗi thời gian không độc lập và có cũng phân phối xác suất. Chúng thường tuân theo một số khuôn mẫu trong thời gian dài. Ví dụ: nếu nhiệt độ ngày hôm nay của một thành phố cụ thể là cực kỳ cao, thì có thể phỏng đoán một cách hợp lý rằng nhiệt độ ngày mai cũng sẽ có khả năng cao. Đây là lý do tại sao dự báo chuỗi thời gian sử dụng một kỹ thuật thích hợp, mang lại kết quả gần với giá trị thực tế.

Quá trình dừng trong chuỗi thời gian

Khái niệm về tính dừng của một quá trình ngẫu nhiên có thể được hình dung như một dạng cân bằng thống kê. Các thuộc tính thống kê như giá trị trung bình và phương sai của một quá trình dừng không phụ thuộc vào thời gian. Đó là điều kiện cần để xây dựng mô hình chuỗi thời gian hữu ích cho việc dự báo trong tương lai. Có hai loại quá trình dừng được định nghĩa dưới đây:

Một quá trình $\{x(t),\ t = 0, 1, 2, \dots\}$ là dừng mạnh hay dừng chặt nếu hàm phân phối xác xuất đồng thời của $\{x_{t-s},\ x_{t-s+1}, \dots, x_t, \dots,\ x_{t+s-1}, x_{t+s}\}$ là độc lập tại t với mọi s. Vì vậy, đối với một quá trình dừng chặt, sự phân phối đồng thời của bất kỳ tập hợp các biến ngẫu nhiên sinh ra từ quá trình là không phụ thuộc thời gian.

Tuy nhiên, đối với các ứng dụng thực tế, không phải lúc nào cũng cần đến giả định về điểm dừng mạnh và do đó, một dạng yếu hơn được được xét đến. Một quá trình ngẫu nhiên là dừng yếu nếu hàm tự hiệp phương sai của 2 biến ngẫu nhiên tại hai thời điểm không phụ thuộc vào thời điểm xét mà chỉ phụ thuộc khoảng giữa 2 thời điểm đang xét đến. Ví dụ một quá trình ngẫu nhiên $\{x(t),\ t = 0, 1, 2, \dots\}$ là một quá trình dừng loại hai nếu kỳ vọng và phương sai không đổi theo thời gian và giá trị hàm tự hiệp phương sai $Cov(x_t, x_{t-s})$ chỉ phụ thuộc vào s.

Một quá trình dừng yếu tuân theo luật phân phối chuẩn thì nó là một quá trình dừng chặt. Một số kiểm định thống kê như kiểm định do Dickey và Fuller đưa ra thường được sử dụng để phát hiện tính dừng trong dữ liệu chuỗi thời gian.

Khái niệm tính dừng là một ý tưởng toán học được xây dựng để đơn giản hóa sự phát triển lý thuyết và thực tiễn của các quá trình ngẫu nhiên. Để thiết kế một mô hình thích hợp, phù hợp cho dự báo trong tương lai, chuỗi thời gian được mong chờ đạt được tính dừng. Tất nhiên, điều này không phải lúc nào cũng đạt được. Như đã nêu bởi Hipel và McLeod, dữ liệu quan sát trong lịch sử càng lớn, thì khả năng chuỗi thời gian thể hiện các đặc điểm không đứng yên càng lớn. Tuy nhiên, trong khoảng thời gian tương đối ngắn, người ta có thể lập mô hình chuỗi thích hợp bằng cách sử dụng quá trình ngẫu nhiên dừng. Thông thường, chuỗi thời gian bao gồm thành phần xu thế hoặc thành phần mùa hoặc cả hai. Những thành phần này ảnh hưởng đến tính dừng trong chuỗi thời gian. Trong các trường hợp cụ thể, chúng ta có thể sử dụng phép sai phân, biến đổi logarit để loại bỏ thành phần xu thế và khiến cho chuỗi thời gian đạt được tính dừng. Trong những phần tiếp theo, chúng ta có thể sử dụng đến phép toán sai phân để chuỗi thời gian đạt được tính dừng.

4.9.3 Mô hình Holt-Winner

Bất kỳ chuỗi thời gian trong hoạt động kinh doanh đều thể hiện hành vi theo mùa, chẳng hạn như nhu cầu về quần áo hoặc đồ chơi. Do đó, các vấn đề dự báo theo mùa có tầm quan trọng đáng kể. Trong phần này, chúng ta sẽ tập chung vào phân tích dữ liệu chuỗi theo mùa bằng cách sử dụng các phương pháp liên tiến lũy thừa Holt-Winters. Hai mô hình được bàn luận là mô hình mùa nhân tính và mô hình mùa cộng tính.

Mô hình nhân tính và mô hình cộng tính

Thuật ngữ tiếng anh, mùa nhân tính và cộng tính được gọi là "multiplicative seasonality" và "additive seasonality". Thông thường, dữ liệu chuỗi thời gian hiển thị hành vi theo mùa. Tính mùa được thể hiện trong dữ liệu chuỗi thời gian thể hiện là hành vi tự lặp lại sau mỗi khoảng thời gian L. Thuật ngữ mùa được sử dụng để biểu thị khoảng

thời gian trước khi hành vi bắt đầu được lặp lại. L là chiều dài mùa của các chu kỳ. Ví dụ: doanh số bán đồ chơi hàng năm có thể sẽ đạt đỉnh vào các tháng 11 và 12 và có thể trong mùa hè, doanh thu nhỏ hơn nhiều. Mô hình này có thể lặp lại hàng năm, tuy nhiên, mức tăng tương đối của doanh số bán hàng trong tháng 12 có thể thay đổi từ từ theo từng năm. Ví dụ, trong tháng 12 hàng năm doanh thu bán một loại đồ chơi có thể tăng 1 triệu đô-la. Vì vậy, chúng ta có thể cộng thêm vào kết quả dự báo doanh thu tháng 12 là 1 triệu đô-la để bù lại cho sự biến động theo mùa. Trong trường hợp như vừa nêu, ta gọi là tính mùa cộng tính. Một trường hợp khác mà chúng ta có thể xem xét, trong tháng 12 doanh thu bán đồ chơi không phải tăng thêm 1 triệu đô-la mà là tăng 40% trung bình các tháng khác, điều này đồng nghĩa với việc doanh thu được nhân với 1.4. Do đó, khi doanh số bán đồ chơi nói chung là không tốt, thì mức tăng doanh thu tuyệt đối (đô la) trong tháng 12 cũng sẽ không tốt (nhưng tỷ lệ phần trăm sẽ không đổi). Nếu doanh số bán đồ chơi cao, thì mức tăng doanh số bán hàng (đô la) tuyệt đối sẽ tăng lên một cách tương ứng. Một lần nữa, trong trường hợp này, doanh số bán hàng tăng theo một yếu tố nhất định và thành phần theo mùa do đó có tính chất nhân.

Liên tiến lũy thừa đơn

Liên tiến lũy thừa đơn được biết đến là một loại mô hình liên tiến lũy thừa đơn giản. Mô hình liên tuyến lũy thừa đơn giản thường được sử dụng cho dự báo ngắn hạn, thường là dự báo cho 1 tháng tiếp theo trong tương lai. Mô hình giả định rằng dữ liệu dao động xung quanh một giá trị trung bình ổn định.

Công thức cụ thể cho mô hình liên tiến lũ thừa đơn giản như sau:

$$S_t = \alpha * X_t + (1 - \alpha) * S_{t-1}$$

$$f_{t+1} = S_t$$

Khi chúng ta áp dụng thủ tục đệ quy đến mỗi quan sát kế tiến trong chuỗi thời gian, mỗi giá trị được dự báo S_t được tính toán như là trung bình có trọng số của giá trị quan sát hiện tại và giá trị đã được dự báo thời gian phía trước. Giá trị được dữ báo tại bước thời gian phía trước lại được tính toán từ giá trị quan sát phía trước và giá trị được dự báo trước nữa.

Vì vậy, mỗi giá trị được dự báo là trung bình có trọng số của các quan sát phía trước nơi mà các trọng số giảm theo cấp số nhân tùy thuộc vào giá trị tham số α. Nếu giá trị tham số bằng 1 thì các quan sát phía trước bị phớt lờ, còn nếu giá trị tham số bằng 0 thì quan sát hiện tại bị phớt lờ, và giá trị dự báo bằng chính giá trị được dữ báo tại bước thời gian phía trước.

Giai đoạn khởi tạo cho S_t là một giai đoạn quan trọng trong việc tính toán tất cả các giá trị tiếp theo. Khởi tạo S_0 bằng giá trị X_0 là một phương pháp khởi tạo. Một cách khác có thể sử dụng là gán nó bằng trung bình bốn hoặc năm quan sát đầu tiên. Giá trị tham số α thì việc lựa chọn giá trị khởi tạo S_0 càng quan trọng.

Mô hình liên tiến lũy thừa có xử lý khuynh

Mô hình liên tiến lũy thừa có xử lý khuynh hay còn gọi là mô hình Holt hoặc mô hình liên tiến lũy thừa cấp 2. Phương pháp này được sử dụng khi dữ liệu có thành phần xu thế. Liên tiến lũy thừa với khuynh hoạt động giống như liên tiến lũy thừa ngoại trừ hai thành phần phải được cập nhật mỗi kỳ là mức và khuynh. Mức là một giá trị ước tính của dữ liệu vào cuối mỗi kỳ. Khuynh là một giá trị ước tính về tăng

trưởng trung bình vào cuối mỗi chu kì. Công thức cụ thể cho mô hình liên tiến có xử lý khuynh như sau:

$$S_t = \alpha * \ y_t + (1-\alpha) * (S_{t-1} - b_{t-1}), \ 0 \leq \alpha \leq 1$$

$$b_t = \gamma * (S_t - S_{t-1}) + (1-\gamma) * b_{t-1}, \ 0 \leq \gamma \leq 1$$

$$f_{t+1} = \ S_t + b_t$$

Với $\gamma \approx 0$: khuynh trong quá khứ có vai trò tương đương nhau. Với $\gamma \approx 1$ khuynh hiện tại có vai trò quan trọng. Có một vài phương pháp được sử dụng để khởi tạo giá trị S_t và b_t. S_0 được gán bằng y_0. Đối với giá trị khởi tạo b_0, chúng ta có thể sử dụng một vài cách sau đây:

$$b_0 = \ y_1 - y_0$$

$$b_0 = [(y_1 - y_0) + (y_2 - y_1) + (y_3 - y_2)] / 3$$

$$b_0 = (y_n - y_1)/(n-1)$$

Mô hình liên tiến lũy thừa có khuynh và mùa

Mô hình này được sử dụng khi dữ liệu bao gồm cả thành phần xu thế và thành phần mùa. Mô hình này còn được gọi với cái tên khác là mô hình Holt-Winter được đặt theo tên của 2 nhà sáng tạo ra nó. Trong mô hình này, chúng ta có thể phân biệt thành 2 mô hình cụ thế của nó dựa trên kiểu mùa đã được trình bày trong phần trên:

- Mô hình mùa nhân tính
- Mô hình mùa cộng tính

Mô hình mùa nhân tính

Trước khi đi vào mô tả các phương trình sử dụng để dự báo chúng ta sẽ nêu ra một vài ký hiệu được sử dụng trong mô hình mùa nhân tính.

Gọi mức loại bỏ mùa của quá trình tại cuối chu kỳ T là R_T.

Tại thời điểm cuối chu kỳ t, ta có $\overline{R}_t$ là ước lượng cho mức loại bỏ mùa. $\overline{G}_t$ là ước lượng cho thành phần xu thế. $\overline{S}_t$ là ước lượng cho thành phần mùa.

$$\overline{R}_t = \alpha * \frac{y_t}{\overline{S}_{t-L}} + (1-\alpha) * \left(\overline{R}_{t-1} + \overline{G}_{t-1}\right)$$

$$\overline{G}_t = \beta * \left(\overline{S}_t - \overline{S}_{t-1}\right) + (1-\beta) * \overline{G}_{t-1}$$

$$\overline{S}_t = \gamma * \frac{y_t}{\overline{S}_t} + (1-\gamma) * \overline{S}_{t-L}$$

Giá trị dự báo cho chu kỳ tiếp theo được cho bởi công thức:

$$y_t = (\overline{R}_{t-1} + \overline{G}_{t-1}) * \overline{S}_{t-L}$$

Giá trị dự báo cho nhiều bước thời gian phía trước

$$y_{t+T} = (\overline{R}_{t-1} + T * \overline{G}_{t-1}) * \overline{S}_{t+T-L}$$

Mô hình mùa cộng tính

Chúng ta sẽ đi trực tiếp vào các phương trình được sử dụng trong mô hình mùa cộng tính:

$$\overline{R}_t = \alpha * (y_t - \overline{S}_{t-L}) + (1-\alpha) * \left(\overline{R}_{t-1} + \overline{G}_{t-1}\right)$$

$$\overline{G}_t = \beta * \left(\overline{S}_t - \overline{S}_{t-1}\right) + (1-\beta) * \overline{G}_{t-1}$$

$$\overline{S}_t = \gamma * (y_t - \overline{S}_t) + (1-\gamma) * \overline{S}_{t-L}$$

Giá trị dự báo cho chu kỳ tiếp theo được cho bởi công thức dưới đây:

$$y_t = \overline{R}_{t-1} + \overline{G}_{t-1} + \overline{S}_{t-L}$$

4.9.4 Mô hình ARIMA

Mô hình tự hồi quy kết hợp trung bình trượt là một kỹ thuật mô hình hóa tuyến tính phổ biến áp dụng cho chuỗi thời gian đơn chiều. Trong kỹ thuật này, dữ liệu chuỗi thời gian trước hết phải kiểm tra tính dừng. Nếu chuỗi dữ liêu chưa có tính dừng, chúng ta áp dụng phép toán sai phân cho chuỗi dữ liệu. Phép toán sai phân được áp dụng nhiều lần đến khi chuỗi dữ liệu đạt được tính dừng. Nếu phép toán sai phân được áp dụng d lần, ta nói rằng bậc sai phân của mô hình ARIMA là d. Trong thực nghiệm, chúng ta có thể áp dụng một số cách dưới đây để kiểm tra tính dừng:

- Vẽ đồ thị biểu diễn bộ dữ liệu.
- Tiến hành chia tách bộ dữ liệu rồi tính trung bình và phương sai trên tập đã được chia tách và so sánh chúng.
- Sử dụng kiểm định Dickey-Fuller để kiểm tra tính dừng.

Kết quả sau khi lấy sai phân, ta có bộ dữ liệu sẽ được mô hình hóa qua mô hình tự hồi quy trung bình trượt (ARMA). Giá trị của chuỗi tai thời điểm t là y_t được xem như một hàm của p giá trị tai thời điểm trước đó (y_{t-1}, y_{t-2}, ..., y_{t-p}) và lỗi tại các thời điểm $t,\ t-1,\ t-2,\ \ldots,\ t-p$ ta đặt là: $\eta_t,\ \eta_{t-1}, \ldots,\ \eta_{t-p}$. Ta có công thức:

$$y_t = \ a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_p y_{t-p} + \eta_t + b_1 \eta_{t-1} + \cdots + \ b_q \eta_{t-q}$$

Trong công thức trên, a_1 đến a_p là các hệ số tự hồi quy(AR) và b_1 đến b_q là các hệ số trung bình trượt (MA). Vì vậy, mô hình mà chúng ta quan tâm được kí hiệu là ARIMA(p, d, q). Mô hình tự hồi quy trung bình trượt (ARMA) giả sử rằng chuỗi lỗi $\eta_t$là nhiễu trắng và có phân phối chuẩn. Mô hình tự hồi quy tích hợp trung bình trượt (ARIMA) được ứng dụng trong dự báo chuỗi thời gian gồm 3 bước: nhận diện mô hình (p, q); ước lượng các hệ số của mô hình; dự báo.

Trong bước thứ nhất, sau khi chuỗi thời gian đã đạt được tính dừng, $p,\ q$ được xác định dựa trên phân tích hàm tự tương quan (ACF - AutoCorrelation Function) và hàm tự tương quan riêng (PACF - Partial AutoCorrelation Function). Ta tiến hành vẽ hàm tự tương quan và tự tương quan riêng để xác định p cho quá trình tự hồi quy (AR), q cho quá trình trung bình trượt (MA). Quá trình phân tích được mô tả vắn tắt trong bảng:

	ACF	PACF
AR(p)	Giảm từ từ về 0 them hàm mũ hoặc sóng hình sin	Giảm ngay về 0 sau độ trễ p
MA(q)	Giảm ngay về 0 sau độ trên q	Giảm từ về 0 theo hàm mũ hoặc sóng hình sin
ARMA(p, q)	Giảm theo hàm mũ	Giảm theo hàm mũ

Trong bước thứ hai, các hệ số của mô hình được ước lượng sử dụng phương pháp ước lượng hợp lý cực đại Gaussian (Gaussian Maximum Likelihood Estimation- GMLE). Mô hình được kiểm định dựa trên chỉ số AIC (Akakine Information Criterion - là 1 tiêu chí nhằm tìm ra mô hình có sự mất mát thông tin nhỏ nhất trong việc lựa chọn mô hình) hoặc MSE (Mean Squared Error - là 1 thước đo lỗi của mô hình),... Mô hình tự hồi quy tích hợp trung bình trượt phù hợp nhất là mô hình có giá trị AIC nhỏ nhất. Trong thực tế, ta có thể sử dụng phương pháp tìm kiếm lưới để đưa ra một mô hình thích hợp nhất. Trong phương pháp tìm kiếm lưới, chúng ta tiến hành vòng lặp qua các bộ thông số p, d, q được lựa chọn có AIC nhỏ nhất. Trong bước cuối cùng, giá trị tương lai được dự báo dựa vào các giá trị trong quá khứ và các hệ số đã được ước lượng.

Đối với từng bài toán cụ thể, vấn đề chọn một mô hình phù hợp với bài toán đặt ra là một vấn đề quan trọng. Do đó, chúng ta cần nắm bắt được những ưu điểm và hạn chế của mô hình được đưa ra.

Những ưu điểm khi sử dụng ARIMA:

- ARIMA khá mềm dẻo có thể áp dụng cho nhiều bài toán dự báo chuỗi thời gian.
- ARIMA thích hợp cho các bài toán dự báo ngắn hạn.

Những mặt hạn chế của khi sử dụng ARIMA:

- Trong ARIMA, một cấu trúc tương quan tuyến tính được giả định giữa các giá trị trong chuỗi thời gian. Do đó, ARIMA chỉ phát hiện được các khuôn mẫu tuyến tính có trong chuỗi dữ liệu còn khuôn mẫu không tuyến tính thì không được phát hiện.
- Quá trình nhận diện mô hình ARIMA thường tốn chi phí tính toán và độ tin cậy của mô hình phụ thuộc vào kỹ năng và kinh nghiệm của người dự báo ví dụ như trong hàng thứ ba trong bảng trên hàm ACF, PACF không giảm ngay về 0 sau độ trễ p, q mà chỉ tắt dần theo hàm mũ. Vậy bằng cách nào ta có thể nhận diện p, q? Một giải pháp được lựa chọn là tìm kiếm lưới và so sánh các mô hình với nhau rồi chọn ra mô hình tốt nhất dựa trên chỉ số AIC nhỏ nhất.
- Các giá trị trong tương lai được dự báo phụ thuộc vào quá khứ nên đối với bài toán dự báo dài hạn, việc lựa chọn ARIMA là không phù hợp.

4.9.5 Mô hình SARIMA và SARIMAX

Mô hình SARIMA

Trong phần này, chúng ta sẽ xem xét đến một vài biến thể của mô hình ARIMA để áp dụng cho những chuỗi thời gian không đạt tính dừng và tồn tại thành phần mùa trong dữ liệu. Thông thường, sự phụ thuộc vào quá khứ có xu hướng xảy ra mạnh mẽ nhất ở bội số của một số độ trễ cơ bản theo mùa. ARIMA theo mùa (SARIMA) được sử dụng khi chuỗi thời gian thể hiện sự biến đổi theo mùa. Các hiện tượng tự nhiên như nhiệt độ và lượng mưa có thành phần mạnh mẽ tương ứng với các mùa. Do đó, sự biến đổi tự nhiên của nhiều quá trình vật lý, sinh học và kinh tế có xu hướng khớp với sự biến động theo mùa. Vậy nên, với các chuỗi dữ liệu có tính chất trên rất thích hợp để khớp với mô hình có các đa thức tự hồi quy và trung bình động xác định với độ trễ theo mùa. Mô hình trung bình động tự hồi quy theo mùa thuần túy $(P,\ Q)\ s$ được giới thiệu bởi Shumay và Stoffer năm 2010:

$$\varphi_P\left(B^S\right) x_t = \theta_Q(B^S) w_t$$

Với các toán tử được định nghĩa dưới đây:

$$\varphi_P\left(B^S\right) = 1 - \varphi_{1S}B^S - \varphi_{2S}B^{2S} - \cdots - \varphi_{pS}B^{pS}$$

$$\theta_Q\left(B^S\right) = 1 + \theta_{1S}B^S + \theta_{1S}B^{2S} + \cdots + +\theta_{QS}B^{QS}$$

Các toán tử trên lần lượt là toán tử tự hồi quy mùa bậc P và toán tử trung bình trượt bậc Q với chu kì mùa S. Một cách tổng quát, chúng ta có thể tổ hợp các toán tử mùa và toán tử không có tính chất mùa trong một mô hình trung bình trượt tự hồi quy mùa, ký hiệu $ARMA(p,q) \times (P,Q)s$ và chúng ta có thể viết lại chúng như dưới đây:

$$\varphi_P\left(B^S\right)\phi(B)x_t = \theta_Q(B^S)\varepsilon(B)w_t$$

Ký hiệu tự hồi quy theo mùa (P) và ký hiệu trung bình động theo mùa (Q) sẽ tạo thành mô hình trung bình động tích hợp tự hồi quy theo mùa nhân tính, được ký hiệu $ARIMA\,(p,d,q) \times (P,D,Q)s$ theo Box và Jenkins năm 1976.

$$\varphi_P\left(B^S\right)\phi(B)\nabla_S^D x_t = \alpha + \theta_Q(B^S)\varepsilon(B)w_t$$

Với w_t là một quá trình nhiễu trắng. Các thành phần tự hồi quy và trung bình trượt không theo tính chất mùa được biểu diễn với các đa thức $\phi(B)$, $\varepsilon(B)$ bậc $p,\ q$. Các thành phần sai phân mùa và sai phân không mùa có thể được viết như sau: $\nabla_S^D = (1-B^S)^D$, $\nabla^d = (1-B)^d$

Mô hình SARIMAX

Mô hình SARIMAX là mô hình SARIMA có thêm các biến ngoại sinh hay nói cách khác mô hình SARIMAX áp dụng cho các chuỗi thời gian đa chiều $SARIMAX\ (p,d,q)(P,D,Q)s\ (X$ với X là vec-tơ biến ngoại sinh. Các biến ngoại sinh này có thể được mô hình hóa bởi mô hình hồi quy đa biến như sau:

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \cdots + \beta_k X_{k,t} + \omega_t \qquad (*)$$

Với $X_{1,t}, X_{2,t}, \ldots, X_{k,t}$ là các quan sát của k biến ngoại sinh ứng với biến phụ thuộc Y_t. $\beta_0, \beta_1, \ldots, \beta_k$ là các hệ số hồi quy của các biến ngoại sinh. ω_t là thành phần dư thừa ngẫu nhiên độc lập với chuỗi thời gian đầu vào. Chuỗi thời gian dư thừa này có thể được mô hình hóa bởi mô hình ARIMA như sau

$$\omega_t = \frac{\theta_Q(B^S)\varepsilon(B)w_t}{\varphi_P\left(B^S\right)\phi(B)(1-B)^d(1-B^S)^D} \quad (**)$$

Mô hình SARIMAX có được bằng cách thay thế (**) vào (*)

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \cdots + \beta_k X_{k,t} + \frac{\theta_Q(B^S)\varepsilon(B)w_t}{\varphi_P\left(B^S\right)\phi(B)(1-B)^d(1-B^S)^D}$$

Mô hình SARIMX cũng bao gồm 5 bước chính:

- Nhận diện mô hình: Trong bước này, chúng ta lựa chọn bậc sai phân d, bậc sai phân mùa D, chiều dài mùa S, bậc tự hồi quy không mang tính chất mùa p, bậc tự hồi quy mùa P, bậc trung bình trượt không mạng tính chất mùa q, bậc trung bình trượt mùa Q. Hàm tự tương quan (ACF) và hàm tự tương quan riêng (PACF) được sử dụng để nhận diện mô hình.
- Ước lượng các tham số: từ các thông số đã lựa chọn ở bước 1 ta tiến hành ước lượng các hệ số của mô hình.
- Chuẩn đoán mô hình: mô hình được chuẩn đoán sử dụng kiểm định Ljung-Box để kiểm tra tính phù hợp của mô hình. Nếu thành phần dư thừa không tuân theo phân phối chuẩn thì sử dụng tiếp bước 4, ngược lại thì sử dụng bước 5.
- Gộp các biến ngoại sinh: các biến ngoại sinh có liên quan được đưa vào mô hình và được mô hình hóa bởi mô hình hồi quy tuyến tính.
- Dự báo và đánh giá mô hình: Mô hình đã được chẩn đoán được đánh bằng cách sử dụng mẫu ngoài. Mô hình đã được đánh giá được sử dụng để dự báo các giá trị trong tương lai.

4.9.6 Mô hình GARCH

Mô hình ARCH

Năm 1982, Engle đã đề xuất mô hình ARCH (Auto-regressive Conditional Heteoskedastic).

$$\sigma_t^2 = \alpha_0 + \alpha_1\varepsilon_{t-1}^2 + \alpha_1\varepsilon_{t-2}^2 + \cdots + \alpha_q\varepsilon_{t-q}^2 = \alpha_0 + \sum_{i=1}^{q}\alpha_i\varepsilon_{t-i}^2$$

Baillie và Bollerslev (1989) đã giải thích sự biến đổi của các thuật ngữ sai số đã được thay đổi từ hằng số thành một chuỗi ngẫu nhiên. Teräsvirta (2006) đã chỉ ra, ε_t có kỳ vọng có điều kiện và phương sai có điều kiện dựa trên tập thông tin I_{t-1}.

$$E\left(\varepsilon_t \mid I_{t-1}\right) = 0$$

$$\sigma_t^2 = E(\varepsilon_t^2 | I_{t-1})$$

Ở đây, ta có:

$$\varepsilon_t = z_t \sigma_t$$

$$z_t \sim N(0, 1)$$

Vì vậy, $\{\varepsilon_t\}$ tuân theo luật phân phối chuẩn với kỳ vọng 0 và phương sai bằng σ_t^2,

$$\varepsilon_t \sim N(0, \sigma_t^2)$$

Giả sử $\alpha_0 > 0$ và $\alpha_i \geq 0,\ i = 1, \ldots, q,\ \alpha_1 + \cdots + \alpha_q < 1$ để đảm bảo $\{\sigma_t^2\}$ đạt được tính dừng yếu.

Mô hình GARCH

Bollerslev (1986) và Taylor (1986) đề xuất mô hình ARCH tổng quát (GARCH – generalized ARCH) để thay thế mô hình ARCH.

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

Alexander và Lazar (2006) giả sử $\alpha_0 > 0$ và $\alpha_i \geq 0,\ i = 1,\ \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 < 1$ để đảm bảo $\{\sigma_t^2\}$ đạt được tính dừng yếu. Enocksson và Skoog (2012) đã chỉ ra một số hạn chế trên mô hình GARCH. Điều quan trọng nhất là mô hình GARCH không thể nắm bắt được hiệu suất không đối xứng. Sau đó, để cải thiện vấn đề này, Nelson (1991) đề xuất mô hình EGARCH và Glosten, Jagannathan và Runkel (1993) đề xuất mô hình GJR-GARCH.

Chương 5

Phân tích phần tử ngoại lai

Hãy tưởng tượng bạn là một người kiểm kê giao dịch ở một công ty thẻ tín dụng. Để bảo vệ các khách hàng khỏi các vụ lừa đảo tín dụng, bạn để ý kĩ lưỡng tới các hoạt động sử dụng thẻ khác biệt so với những trường hợp bình thường. Ví dụ, việc mua hàng với số tiền lớn hơn nhiều so với bình thường của chủ thẻ, hay nó được thực hiện ở xa thành phố người đấy sống, thì giao dịch đó có thể coi là đáng nghi. Bạn muốn phát hiện những giao dịch như vậy ngay khi nó xảy ra và liên lạc với chủ thẻ để xác nhận. Đây là trường hợp thường thấy trong các công ty thẻ tín dụng. *Vậy kĩ thuật khai phá dữ liệu nào có thể giúp chung ta phát hiện những giao dịch đáng ngờ*
Hầu hết các giao dịch thẻ tín dụng đều bình thường. Tuy nhiên, nếu một chiếc thẻ bị mất trộm, những mẫu (pattern) giao dịch thường thay đổi một cách đáng kể - địa điểm của những lần mua sắm và những thứ được mua thường rất khác so với chủ thẻ và những khách hàng khác. Một ý tưởng thiết yếu của việc phát hiện thẻ tín dụng bị lấy cắp là xác định những giao dịch khác biệt so với bình thường.
Phát hiện phần tử ngoại lai (Outliers detection - anomaly detection) là quá trình tìm các đối tượng dư liệu có hành vi rất khác biệt so với bình thường. Những đối tượng đó được gọi là **outliers (phần tử ngoại lai)** hoặc **anomalies (phần tử dị thường)**. Không chỉ có ứng dụng trong phát hiện lừa đảo, phát hiện phần tử ngoại lai rất quan trọng trong y tế, an toàn và an ninh công cộng, phát hiện hư hại trong công nghiệp, xử lí ảnh, các thiết bị cảm biến và phát hiện xâm nhập.

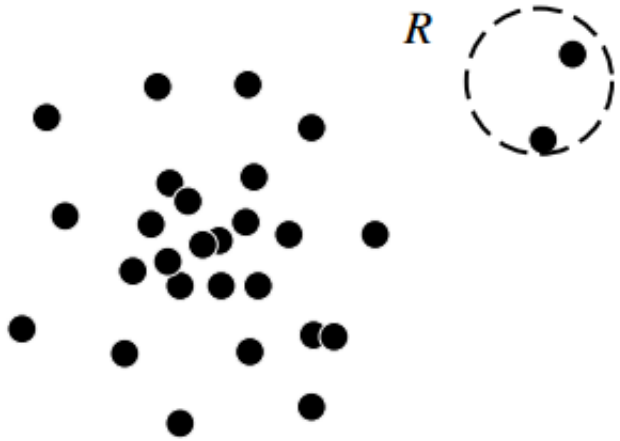
Phát hiện ngoại lai và phân tích phân cụm là hai nhiệm vụ gắn liện với nhau. Mục (1) sẽ định nghĩa các loại ngoại lai khá nhau. Mục (2) giới thiệu tổng quan các phương pháp phát hiện phần tử ngoại lai. Trong phần còn lại của báo cáo này, các phương pháp phát hiện ngoại lai sẽ được trình bày chi tiết. Các phương pháp đấy được phân loại thành thống kê (Mục 3), dựa trên khoảng cách (Mục 4), dựa trên phân cụm (Mục 5), và dựa trên phân lớp (Mục 6). Mục 7 trình bày về những phần tử ngoại lai theo bối cảnh và tập hợp các phần tử ngoại lai, và ở mục 8 là phát hiện bất thường trong không gian nhiều chiều.

5.1 Phần tử ngoại lai và phân tích phần tử ngoại lai

Phần này sẽ trình bày định nghĩa về các phần tử ngoại lai, phân loại các phần tử ngoại lai và sau đó bàn về những thách thức của phân tích phần tử ngoại lai một cách tổng quát.

5.1.1 Phần tử ngoại lai là gì

Giả sử một quá trình thống kê đã sinh ra được một tập các đối tượng dữ liệu. Một *phần tử ngoại lai* là một đối tượng khác biệt rõ rệt so với các đối tượng khác như thể nó được sinh ra từ một cơ chế khác. Để dễ dàng trình bay, những đối tượng dữ liệu không phải ngoại lai sẽ được gọi là "bình thường" hoặc dữ liệu mong muốn. Tương tự chúng ta có thể gọi các ngoại lệ là dữ liệu bất thường. Ví dụ 1: Điểm ngoại lai. Trong Hình 1, hầu hết các đối tượng tuân theo phân phối Gaussian. Tuy nhiên, các đối tượng trong khu vực R có ý nghĩa khác nhau. Không chắc là chúng tuân theo phân phối giống như các đối tượng khác trong tập dữ liệu. Do đó, các đối tượng trong R là các ngoại lệ trong tập dữ liệu.



Hình 5.1: Đối tượng trong R là phần tử ngoại lai

Dữ liệu ngoại lai khác với dữ liệu nhiễu, nhiễu là lỗi hoặc phương sai trong biến đo lường được. Ví dụ: một khách hàng có thể tạo ra một số giao dịch nhiễu hoặc sai lệch ví dụ như một bữa trưa lớn hơn mọi ngày hoặc uống thêm một ly caffe so với bình thường, các giao dịch như vậy chúng ta không nên coi là giao dịch bất thường. Điều này sẽ ảnh hưởng rất nhiều đến trải nghiệm khách hàng vì nếu cảnh báo sai thì khách hàng sẽ cảm thấy phiền toái và công ty có thể mất đi khách hàng đó . Vì vậy việc loại bỏ nhiễu trước khi phát hiện điểm ngoại lai là vô cùng cần thiết.

Điểm ngoại lai thường được tạo ra bởi các cơ chế không giống như phần còn lại của dữ liệu. Vì thế trong việc phát hiện ngoại lệ điều quan trọng là phải chứng minh được tại sao các ngoại lệ được phát hiện được tạo ra bởi các cơ chế khác. Để thực hiện chúng ta phải đưa ra các giả định khác nhau trên phần còn lại của dữ liệu và cho thấy các ngoại lệ được phát hiện vi phạm các giả định đó một cách đáng kể.

5.1.2 Phân loại phần tử ngoại lai

Nói chung, phần tử ngoại lai có thể được phân loại thành ba loại, đó là các ngoại lệ toàn cầu (global outliers), các ngoại lệ bối cảnh (hoặc có điều kiện) và các ngoại lệ tập thể.

Ngoại lệ toàn cục - Global Outliers

Trong một tập dữ liệu nhất định, một đối tượng dữ liệu là một ngoại lệ toàn cục nếu nó lệch đáng kể so với phần còn lại của tập dữ liệu. Các ngoại lệ toàn cầu đôi khi được gọi là điểm dị thường và là loại ngoại lệ đơn giản nhất. Hầu hết các phương pháp phát hiện ngoại lệ đều nhằm mục đích phát triển các ngoại lệ toàn cục.

Ví dụ 2: Các ngoại lệ toàn cầu. Hãy xem xét các điểm trong *Hình 1* một lần nữa. Các điểm trong R chệch nhiều so với phần còn lại của tập dữ liệu, do đó chúng là các ngoại lệ toàn cầu.

Để phát hiện các ngoại lệ toàn cầu, một vấn đề quan trọng là tìm ra một phép đo độ lệch thích hợp đối với ứng dụng được đề cập. Các phép đo khác nhau được cung cấp và dựa trên các phương pháp phát hiện ngoại lệ này được phân chia thành các loại khác nhau.

Phát hiện ngoại lệ toàn cầu là quan trọng trong nhiều ứng dụng. Ví dụ, xem xét phát hiện xâm nhập trong các mạng máy tính. Nếu hành vi của máy tính rất khác so với thông thường (ví dụ: một số lượng lớn các gói được phát hành trong một thời gian ngắn), thì hành vi này có thể được coi là ngoại lệ toàn cục và máy tính tương ứng là nạn nhân bị nghi ngờ là hack. Một ví dụ khác, trong các hệ thống kiểm toán giao dịch truyền thống, các giao dịch không tuân theo các quy định được coi là ngoại lệ toàn cầu và cần được tổ chức để kiểm tra thêm.

Ngoại lệ bối cảnh - Contextual Outliers

Nhiệt độ hôm nay là 28C. Có phải là ngoại lệ không? Điều đó còn phụ thuộc vào mùa, vào thời gian và địa điểm đo nhiệt độ! Nếu đó là vào mùa đông ở Toronto, đó là một ngoại lệ. Nếu đó là một ngày hè ở Toronto, thì đó là chuyện bình thường. Không giống như phát hiện ngoại lệ toàn cục, trong trường hợp này, giá trị nhiệt độ của ngày hôm nay có phải là ngoại lệ hay không phụ thuộc vào bối cảnh, ngày, địa điểm và có thể một số yếu tố khác.

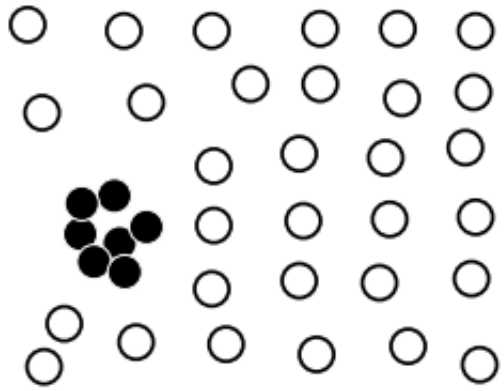
Trong một tập dữ liệu nhất định, một đối tượng dữ liệu là một ngoại lệ ngữ cảnh nếu nó sai lệch đáng kể đối với bối cảnh cụ thể của đối tượng. Các ngoại lệ theo ngữ cảnh còn được gọi là các ngoại lệ có điều kiện vì chúng có điều kiện là các bối cảnh đã chọn. Do đó, trong phát hiện ngoại lệ theo ngữ cảnh, bối cảnh phải được xác định cụ thể như là một phần của bài toán. Nói chung, trong phát hiện ngoại lệ theo ngữ cảnh, các thuộc tính của các đối tượng dữ liệu được đề cập được chia thành hai nhóm:

- Thuộc tính bối cảnh: Bối cảnh để xem xét các đối tượng dữ liệu. Trong ví dụ về

nhiệt độ, các thuộc tính bối cảnh có thể là ngày và vị trí.

- Thuộc tính hành vi: Đây là các đặc điểm của đối tượng, và được sử dụng để đánh giá xem đối tượng có phải là ngoại lệ hay không. Trong ví dụ về nhiệt độ, các thuộc tính hành vi có thể là nhiệt độ, độ ẩm và áp suất.

Không giống như phát hiện ngoại lệ toàn cục, trong phát hiện ngoại lệ theo ngữ cảnh, việc một đối tượng dữ liệu có phải là ngoại lệ hay không phụ thuộc vào không chỉ các thuộc tính hành vi mà còn cả các thuộc tính theo ngữ cảnh. Sự kết hợp của các giá trị thuộc tính hành vi có thể được coi là ngoại lệ trong một bối cảnh (ví dụ: 28C là ngoại lệ cho mùa đông Toronto), nhưng không phải là ngoại lệ trong bối cảnh khác (ví dụ: 28C không phải là ngoại lệ cho mùa hè Toronto).
Phát hiện ngoại lệ toàn cầu có thể được coi là một trường hợp đặc biệt của phát hiện ngoại cảnh theo ngữ cảnh trong đó tập hợp các thuộc tính theo ngữ cảnh trống. Nói cách khác, phát hiện ngoại lệ toàn cầu sử dụng toàn bộ tập dữ liệu làm bối cảnh.
Ví dụ 3 Các ngoại lệ theo ngữ cảnh. Trong phát hiện gian lận thẻ tín dụng, ngoài các ngoại lệ toàn cầu, một nhà phân tích có thể xem xét các ngoại lệ trong các bối cảnh khác nhau. Hãy xem xét những khách hàng sử dụng hơn 90% hạn mức tín dụng của họ. Nếu một khách hàng như vậy được xem là thuộc về một nhóm khách hàng có giới hạn tín dụng thấp, thì hành vi đó có thể không được coi là ngoại lệ. Tuy nhiên, hành vi tương tự của khách hàng từ một nhóm thu nhập cao có thể được coi là ngoại lệ nếu số dư của họ thường vượt quá giới hạn tín dụng. Những ngoại lệ như vậy có thể dẫn đến các cơ hội kinh doanh mà việc tăng giới hạn tín dụng cho những khách hàng đó có thể mang lại doanh thu mới.
hất lượng phát hiện ngoại lệ theo ngữ cảnh trong một ứng dụng phụ thuộc vào ý nghĩa của các thuộc tính theo ngữ cảnh, ngoài việc đo lường độ lệch của một đối tượng với đa số trong không gian của các thuộc tính hành vi. Thường xuyên hơn không, các thuộc tính theo ngữ cảnh nên được xác định bởi các chuyên gia tên miền, có thể được coi là một phần của kiến thức nền đầu vào. Trong nhiều ứng dụng, việc thu thập thông tin cần thiết để xác định các thuộc tính theo ngữ cảnh cũng như không thu thập dữ liệu thuộc tính theo ngữ cảnh chất lượng cao là dễ dàng.
Làm thế nào chúng ta có thể hình thành các bối cảnh có ý nghĩa trong phát hiện ngoại lệ theo ngữ cảnh? Một phương pháp đơn giản chỉ sử dụng các nhóm của các thuộc tính theo ngữ cảnh làm bối cảnh. Tuy nhiên, điều này có thể không hiệu quả vì nhiều nhóm có thể có dữ liệu và / hoặc nhiễu không cần thiết. Một phương thức tổng quát hơn sử dụng sự gần gũi của các đối tượng dữ liệu trong không gian của các thuộc tính theo ngữ cảnh.
Ngoại lệ tập thể - Collective Outliers Trong Hình 2, các đối tượng màu đen nói chung tạo thành một tập thể ngoại lệ vì mật độ của các đối tượng đó cao hơn nhiều so với phần còn lại trong tập dữ liệu. Tuy nhiên, mỗi đối tượng màu đen riêng lẻ không phải là một ngoại lệ đối với toàn bộ tập dữ liệu.



Hình 5.2: Đối tượng màu đen là ngoại lệ tập thể

Phát hiện ngoại lệ tập thể có nhiều ứng dụng quan trọng. Ví dụ, trong phát hiện xâm nhập, gói từ chối dịch vụ từ máy tính này sang máy tính khác được coi là bình thường và hoàn toàn không phải là một ngoại lệ. Tuy nhiên, nếu một số máy tính tiếp tục gửi các gói từ chối dịch vụ cho nhau, thì toàn bộ chúng nên được coi là một ngoại lệ phổ biến. Các máy tính liên quan có thể bị nghi ngờ là bị xâm phạm bởi một cuộc tấn công. Một ví dụ khác, giao dịch chứng khoán giữa hai bên được coi là bình thường. Tuy nhiên, một tập hợp lớn các giao dịch của cùng một cổ phiếu giữa một bên nhỏ trong một thời gian ngắn là các ngoại lệ tập thể vì chúng có thể là bằng chứng của một số người thao túng thị trường.
Không giống như phát hiện ngoại lệ toàn cầu hoặc theo ngữ cảnh, trong phát hiện ngoại lệ tập thể, chúng ta phải xem xét không chỉ hành vi của từng đối tượng, mà cả đối tượng của các nhóm đối tượng. Do đó, để phát hiện các ngoại lệ tập thể, chúng ta cần có kiến thức nền tảng về mối quan hệ giữa các đối tượng dữ liệu như khoảng cách hoặc các phép đo tương tự giữa các đối tượng.
Tóm lại, một tập dữ liệu có thể có nhiều loại ngoại lệ. Hơn nữa, một đối tượng có thể thuộc về nhiều loại ngoại lệ. Trong kinh doanh, các ngoại lệ khác nhau có thể được sử dụng trong các ứng dụng khác nhau hoặc cho các mục đích khác nhau. Phát hiện ngoại lệ toàn cầu là đơn giản nhất. Phát hiện ngoại cảnh bối cảnh đòi hỏi thông tin cơ bản để xác định các thuộc tính và bối cảnh theo ngữ cảnh. Phát hiện ngoại lệ tập thể đòi hỏi thông tin cơ bản để mô hình hóa mối quan hệ giữa các đối tượng với các nhóm ngoại lệ.

5.1.3 Thách thức trong việc phát hiện điểm ngoại lai

Phát hiện ngoại lệ rất hữu ích trong nhiều ứng dụng nhưng phải đối mặt với nhiều thách thức như sau:

- Mô hình hóa các đối tượng bình thường và các ngoại lệ một cách hiệu quả. Chất lượng phát hiện ngoại lệ phụ thuộc rất nhiều vào mô hình của các đối tượng và ngoại lệ thông thường (không sớm hơn). Thông thường, việc xây dựng một mô

hình toàn diện cho tính quy tắc dữ liệu là rất khó khăn, nếu không nói là không thể. Điều này một phần vì khó có thể liệt kê tất cả các hành vi bình thường có thể có trong một ứng dụng.

- Ranh giới giữa tính chuẩn và dữ liệu bất thường (ngoại lệ) thường không rõ ràng. Thay vào đó, có thể có một phạm vi rộng của khu vực màu xám. Do đó, trong khi một số phương thức phát hiện bên ngoài chỉ định cho từng đối tượng trong dữ liệu đầu vào, đặt nhãn của một cách bình thường hoặc thông thường, thì các phương thức khác gán cho mỗi đối tượng một số điểm đo mức độ ngoại lệ của đối tượng.

- Ứng dụng phát hiện ngoại lệ. Về mặt kỹ thuật, việc chọn thước đo độ tương tự / khoảng cách và mô hình mối quan hệ để mô tả các đối tượng dữ liệu là rất quan trọng trong phát hiện ngoại lệ. Thật không may, những lựa chọn như vậy thường phụ thuộc vào ứng dụng. Các ứng dụng khác nhau có thể có các yêu cầu rất khác nhau. Ví dụ, trong phân tích dữ liệu của phòng khám, một sai lệch nhỏ có thể đủ quan trọng để biện minh cho một ngoại lệ. Ngược lại, trong phân tích tiếp thị, các đối tượng thường chịu các ứng dụng lớn hơn và do đó cần có độ lệch lớn hơn đáng kể để biện minh cho một ngoại lệ. Phát hiện ngoại lệ Phụ thuộc cao vào loại ứng dụng khiến cho không thể phát triển một phương pháp phát hiện ngoại lệ có thể áp dụng phổ biến. Thay vào đó, các phương pháp phát hiện ngoại lệ riêng được dành riêng cho các ứng dụng cụ thể phải được phát triển.

- Xử lý nhiễu trong phát hiện ngoại lệ. Như đã đề cập trước đó, các ngoại lệ khác với nhiễu. Người ta cũng biết rằng chất lượng của các tập dữ liệu thực có xu hướng kém. Nhiễu thường không thể tránh khỏi tồn tại trong dữ liệu được thu thập trong nhiều ứng dụng. Nhiễu có thể xuất hiện dưới dạng độ lệch trong giá trị thuộc tính hoặc thậm chí là giá trị thiếu. Chất lượng dữ liệu thấp và sự hiện diện của nhiễu mang lại thách thức lớn cho việc phát hiện ngoại lệ. Họ có thể bóp méo dữ liệu, làm mờ sự phân biệt giữa các đối tượng bình thường và các ngoại lệ. Ngoài ra, nhiễu và dữ liệu bị thiếu có thể che giấu các ngoại lệ của ốp lưng và làm giảm hiệu quả của việc phát hiện bên ngoài.

- Trong một số tình huống ứng dụng, người dùng có thể không chỉ muốn phát hiện các ngoại lệ, mà còn hiểu tại sao các đối tượng được phát hiện là ngoại lệ. Để đáp ứng yêu cầu dễ hiểu, một phương pháp phát hiện ngoại lệ phải cung cấp một số lý do phát hiện. Ví dụ, một phương pháp thống kê có thể được sử dụng để xác định mức độ mà một đối tượng có thể là ngoại lệ dựa trên khả năng đối tượng được tạo bởi cùng một cơ chế tạo ra phần lớn dữ liệu. Khả năng càng nhỏ, đối tượng càng ít được tạo ra bởi cùng một cơ chế và đối tượng càng có khả năng là một ngoại lệ.

Phần còn lại của chương này thảo luận về cách tiếp cận để phát hiện ngoại lệ. Có nhiều phương pháp phát hiện ngoại lệ trong tài liệu và trong thực tế. Ở đây, chúng tôi trình bày hai cách trực giao để phân loại các phương pháp phát hiện ngoại lệ. Đầu tiên, chúng tôi phân loại các phương pháp phát hiện ngoại lệ tùy theo mẫu dữ liệu để phân tích có được cung cấp với các chuyên gia tên miền cung cấp các nhãn có thể được

sử dụng để xây dựng mô hình phát hiện ngoại lệ hay không. Thứ hai, chúng tôi chia các phương thức thành các nhóm theo các giả định của chúng về các đối tượng bình thường so với các ngoại lệ.

5.2 Tiếp cận dựa trên thống kê

Các phương pháp thống kê để phát hiện ngoại lệ đưa ra các giả định về tính quy tắc dữ liệu. Họ cho rằng các đối tượng bình thường trong một tập dữ liệu được tạo ra bởi một quá trình ngẫu nhiên (một mô hình tổng quát). Do đó, các đối tượng bình thường xảy ra ở các vùng có xác suất cao đối với mô hình ngẫu nhiên và các đối tượng ở các vùng có xác suất thấp là các ngoại lệ.

Ý tưởng chung đằng sau các phương pháp thống kê để phát hiện ngoại lệ là tìm hiểu một mô hình tổng quát, kết hợp tập dữ liệu đã cho và sau đó xác định các đối tượng đó trong các vùng có xác suất thấp của mô hình là các ngoại lệ. Tuy nhiên, có nhiều cách khác nhau để tìm hiểu các mô hình tổng quát. Nói chung, các phương pháp thống kê để phát hiện ngoại lệ có thể được chia thành hai loại chính: phương pháp tham số và phương pháp không tham số, theo cách các mô hình được xác định và học hỏi.

Phương pháp tham số giả định rằng các đối tượng dữ liệu bình thường được tạo bởi phân phối số liệu với tham số y. Hàm mật độ xác suất của phân phối tham số $f(x, y)$ đưa ra xác suất mà đối tượng x được tạo bởi phân phối. Giá trị này càng nhỏ, x càng có khả năng là một ngoại lệ. Một phương pháp không tham số không giả định một mô hình thống kê tiên nghiệm. Thay vào đó, một phương pháp không tham số cố gắng xác định mô hình từ dữ liệu đầu vào. Lưu ý rằng hầu hết các phương pháp không tham số không cho rằng mô hình hoàn toàn không có tham số. Ví dụ về các phương pháp không tham số bao gồm biểu đồ mật độ.

5.2.1 Các phương pháp tham số

Trong mục này, chúng tôi giới thiệu một số phương pháp tham số đơn giản nhưng thực tế để phát hiện ngoại lệ. Chúng tôi sẽ thảo luận về các phương pháp cho dữ liệu đơn biến dựa trên phân phối thông thường. Sau đó chúng tôi thảo luận cách xử lý dữ liệu đa biến bằng nhiều phân phối tham số.

- Phát hiện các ngoại lệ đơn biến dựa trên phân phối chuẩn.
  Dữ liệu chỉ liên quan đến một thuộc tính hoặc biến được gọi là dữ liệu đơn biến. Để đơn giản, chúng tôi thường chọn giả định rằng dữ liệu được tạo từ phân phối bình thường. Sau đó chúng ta có thể tìm hiểu các tham số của phân phối bình thường từ dữ liệu đầu vào và xác định các điểm có xác suất thấp là ngoại lệ.
  Hãy bắt đầu với dữ liệu đơn biến. Chúng tôi sẽ cố gắng phát hiện các ngoại lệ bằng cách giả sử dữ liệu tuân theo phân phối bình thường.
  Ví dụ 8: Phát hiện ngoại lệ đơn lẻ bằng maximun likelihood. Giả sử giá trị nhiệt độ trung bình của một thành phố trong tháng 7 trong 10 năm qua, theo thứ tự tăng dần giá trị, 24.0 C, 28.9C, 28.9C, 29.0, 29.1C, 29.1C, 29.2C , 29.2C, 29.3C

và 29.4C. Hãy giả sử rằng nhiệt độ trung bình tuân theo phân phối bình thường, được xác định bởi hai tham số: giá trị trung bình, và độ lệch chuẩn.
Chúng ta có thể sử dụng phương pháp maximun likelihood để ước tính các tham số μ và. Đó là, chúng tôi tối đa hóa chức năng hàm log likelihood:

$$\ln\left(\mu, \sigma^2\right) = \sum_{i}^{n} \tag{5.1}$$

Trong đó n là tổng số mẫu, là 10 trong ví dụ này.
lấy đạo hàm tương ứng với μ và σ^2 và giải kết quả của các điều kiện đầu tiên dẫn đến các MLE sau đây:

$$\hat{\mu} = \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2.$$

trong ví dụ này,ta có:

image/image04.png

Theo đó ta có $\sigma = \sqrt{2.29} = 1.51$
Giá trị sai lệch nhất,$24,0C$, cách $4.61C$ so với giá trị trung bình ước tính. Ta biết rằng vùng $\mu \pm 3\sigma$ chứa 99.7% dữ liệu theo giả định bình thường.
Vì $\frac{4.61}{1.51} = 3.04 > 3$, xác suất giá trị $24.0C$ được sinh ra bởi phân phối chuẩn là nhỏ hơn 0.15% và do đó có thể xác định rằng nó là ngoại lai.
Ví dụ trên xây dựng một phương pháp phát hiện ngoại lai đơn giản nhưng thực tế. Nó đơn giản là gắn nhãn bất kỳ đối tượng nào là ngoại lai nếu nó cách trung bình của phân phối ước tính hơn 3, trong đó là độ lệch chuẩn.
Phát hiện các ngoại lai đa biến Dữ liệu liên quan đến hai hoặc nhiều thuộc tính hoặc biến là dữ liệu đa biến. Nhiều phương pháp phát hiện ngoại lệ đơn biến có thể được mở rộng để xử lý dữ liệu đa biến. Ý tưởng trung tâm là biến đổi nhiệm vụ phát hiện ngoại lệ đa biến thành một vấn đề phát hiện ngoại lệ đơn biến. Ở đây, chúng tôi sử dụng hai ví dụ để minh họa ý tưởng này. **Sử dụng hỗn hợp các tham số phân phối** Nếu chúng ta giả định rằng dữ liệu được tạo bởi một phân phối bình thường, thì điều này hoạt động tốt trong nhiều tình huống. Tuy nhiên, giả định này có thể quá đơn giản khi phân phối dữ liệu thực tế phức tạp. Trong các trường hợp như vậy, chúng tôi thay vào đó giả định rằng dữ liệu được tạo bởi hỗn hợp các phân phối tham số.

5.2.2 Các phương pháp không tham số

Trong các phương pháp không tham số để phát hiện ngoại lệ, mô hình của dữ liệu bình thường, được học từ dữ liệu đầu vào, thay vì giả sử một dữ liệu tiên nghiệm. Các phương pháp không tham số thường đưa ra ít giả định hơn về dữ liệu và do đó có thể được áp dụng trong nhiều tình huống hơn. Ví dụ 12.13 Phát hiện ngoại lệ bằng biểu đồ. AllElect Electronics ghi lại số tiền mua cho mỗi giao dịch của khách hàng. Hình 5 sử dụng biểu đồ để biểu đồ số tiền này dưới dạng tỷ lệ phần trăm, cho tất cả các giao dịch. Ví dụ: 60% số tiền giao dịch nằm trong khoảng từ $ 0,00 đến $ 1000.

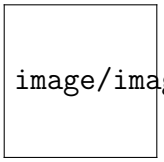
Hình 5.3: biểu đồ số tiền mua sắm trong giao dịch

Như được minh họa trong ví dụ trước, biểu đồ là một mô hình thống kê phi số liệu được sử dụng thường xuyên có thể được sử dụng để phát hiện các ngoại lệ. Thủ tục bao gồm hai bước sau đây.
Bước 1: Xây dựng biểu đồ. Trong bước này, chúng tôi xây dựng một biểu đồ sử dụng dữ liệu đầu vào (dữ liệu đào tạo). Biểu đồ có thể là đơn biến như trong hoặc đa biến nếu dữ liệu đầu vào là đa chiều.
Lưu ý rằng mặc dù các phương pháp không tham số không giả định bất kỳ mô hình thống kê tiên nghiệm nào, nhưng chúng thường yêu cầu các tham số cụ thể của người dùng để tìm hiểu các mô hình từ dữ liệu. Ví dụ: để xây dựng biểu đồ tốt, người dùng phải chỉ định loại biểu đồ (ví dụ: chiều rộng bằng hoặc độ sâu bằng nhau) và các tham số khác (ví dụ: số lượng thùng trong biểu đồ hoặc kích thước của mỗi thùng). Không giống như các phương thức tham số, các tham số này không chỉ định các loại phân phối dữ liệu (ví dụ: Gaussian).
Bước 2: Phát hiện ngoại lệ. Để xác định xem một đối tượng, o, có phải là ngoại lệ hay không, chúng ta có thể kiểm tra đối tượng đó với biểu đồ. Theo cách tiếp cận đơn giản nhất, nếu đối tượng rơi vào một trong các thùng biểu đồ, thì đối tượng được coi là bình thường. Mặt khác, nó được coi là một ngoại lệ.
Đối với một cách tiếp cận tinh vi hơn, chúng ta có thể sử dụng biểu đồ để gán điểm số cao hơn cho đối tượng. Trong ví dụ 12.13, chúng ta có thể để một đối tượng Điểm số ngoại lệ là nghịch đảo của thể tích của thùng trong đó đối tượng rơi. Ví dụ: điểm ngoại lệ cho số tiền giao dịch là $ 7500 là $1/0.2 = 500$ và với số tiền giao dịch là $ 385 là $1/0.6 = 1{,}67$. Điểm số cho thấy số tiền giao dịch là $ 7500 có nhiều khả năng là một ngoại lệ so với $ 385. Một nhược điểm của việc sử dụng biểu đồ như một mô hình không tham số để phát hiện ngoại lệ là khó có thể chọn kích thước thùng thích hợp. Một mặt, nếu kích thước thùng được đặt quá nhỏ, nhiều vật thể bình thường có thể rơi vào các thùng rỗng hoặc hiếm, và do đó bị nhầm lẫn là ngoại lệ. Điều này dẫn đến tỷ lệ dương tính giả cao và độ chính xác thấp. Mặt khác, nếu kích thước thùng được đặt quá cao, các vật thể ngoại lai có thể ltrate vào một số thùng thường xuyên và do đó, được cải trang thành bình thường. Điều này dẫn đến tỷ lệ âm tính giả cao và thu

hồi thấp. Để khắc phục vấn đề này, chúng ta có thể áp dụng ước tính mật độ hạt nhân để ước tính phân bố mật độ xác suất của dữ liệu. Chúng tôi coi một đối tượng quan sát là một chỉ số có mật độ xác suất cao ở khu vực xung quanh. Mật độ xác suất tại một điểm phụ thuộc vào khoảng cách từ điểm này đến các đối tượng quan sát. Chúng tôi sử dụng hàm kernel để mô hình hóa sự phát triển của một điểm mẫu trong vùng lân cận của nó. Một hạt nhân K () là một hàm tích hợp có giá trị thực không âm có trong hai điều kiện sau:

- $\int -\infty^{\infty} K(u)du = 1.$
- $K(-u) = K(u)$ với mọi u.

Một hạt nhân được sử dụng thường xuyên là một hàm Gaussian chuẩn với giá trị trung bình 0 và phương sai 1:



Trong đó $K()$ là kernel và h là băng thông đóng vai trò là tham số làm mịn.
Một khi hàm phân phối xác suất của một tập dữ liệu được xấp xỉ thông qua xấp xỉ mật độ hạt nhân, ta có thể sử dụng hàm mật độ xấp xỉ $\bar{f}$ để xác định ngoại lai. VỚi một đối tượng o, $\bar{f}(o)$ đưa ra xác suất ước tính rằng đối tượng được tạo ra bởi quá trình ngẫu nhiên. Nếu $\bar{f}(o)$ cao, thì đối tượng có khả năng bình thường. Nếu không, o có khả năng là một ngoại lai. Bước này thường tương tự như bước tương ứng trong các phương pháp tham số. Tóm lại, các phương pháp thống kê để phát hiện ngoại lệ tìm hiểu các mô hình từ dữ liệu để phân biệt các đối tượng dữ liệu thông thường từ các ngoại lệ. Một lợi thế của việc sử dụng các phương pháp thống kê là việc phát hiện ngoại lệ có thể có ý nghĩa thống kê. Tất nhiên, điều này chỉ đúng nếu giả định thống kê được thực hiện về dữ liệu cơ bản đáp ứng các ràng buộc trong thực tế. Việc phân phối dữ liệu của dữ liệu chiều cao thường phức tạp và khó hiểu. Do đó, các phương pháp thống kê để phát hiện ngoại lệ trên dữ liệu chiều cao vẫn là một thách thức lớn. Phát hiện ngoại lệ cho dữ liệu chiều cao được đề cập thêm trong Phần 8. Chi phí tính toán của các phương pháp thống kê phụ thuộc vào các mô hình. Khi các mô hình tham số đơn giản được sử dụng (ví dụ: Gaussian), việc kết nối các tham số thường mất thời gian. Khi các mô hình phức tạp hơn được sử dụng (ví dụ: các mô hình hỗn hợp, trong đó thuật toán EM được sử dụng trong học tập), việc xấp xỉ các giá trị tham số tốt nhất thường mất vài lần lặp. Tuy nhiên, mỗi lần lặp thường là tuyến tính đối với kích thước tập hợp dữ liệu. Đối với ước tính mật độ hạt nhân, chi phí học tập mô hình có thể lên tới bậc hai. Một khi mô hình được học, chi phí phát hiện ngoại lệ thường rất nhỏ cho mỗi đối tượng.

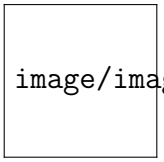
5.3 Các phương phấp phân cụm

Khái niệm về các ngoại lệ có liên quan cao đến các cụm. Phương pháp tiếp cận dựa trên cụm phát hiện các ngoại lệ bằng cách kiểm tra mối quan hệ giữa các đối tượng và

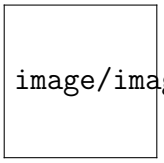
cụm. Theo trực giác, một ngoại lệ là một đối tượng thuộc về một cụm nhỏ và từ xa, hoặc không thuộc về bất kỳ cụm nào.
Điều này dẫn đến ba cách tiếp cận chung để phát hiện ngoại lệ dựa trên cụm.

- Đối tượng có thuộc cụm nào không? Nếu không, thì nó được xác định là ngoại lệ.
- Có một khoảng cách lớn giữa đối tượng và cụm mà nó gần nhất không? Nếu có, nó là một ngoại lệ.
- Là một phần đối tượng của một cụm nhỏ hoặc thưa thớt? Nếu có, thì tất cả các đối tượng trong cụm đó là ngoại lệ

Ví dụ 15: Phát hiện các ngoại lệ là các đối tượng không thuộc bất kỳ cụm nào. Động vật Gregarious (ví dụ, dê và hươu) sống và di chuyển trong ocks. Sử dụng phát hiện ngoại lệ, chúng ta có thể xác định các ngoại lệ là động vật không phải là một phần của ock. Những động vật như vậy có thể bị mất hoặc bị thương.
Trong hình 10, mỗi điểm đại diện cho một động vật sống trong một nhóm. Sử dụng phương pháp phân cụm dựa trên mật độ, chẳng hạn như DBSCAN, chúng tôi lưu ý rằng các điểm đen thuộc về các cụm. Điểm trắng, a, không thuộc về bất kỳ cụm nào, và do đó được tuyên bố là ngoại lệ.
Cách tiếp cận thứ hai để phát hiện ngoại lệ dựa trên phân cụm xem xét khoảng cách giữa một đối tượng và cụm mà nó gần nhất. Nếu khoảng cách lớn, thì đối tượng có khả năng là một ngoại lệ đối với cụm. Do đó, phương pháp này phát hiện các ngoại lệ riêng lẻ đối với các cụm.
Ví dụ 16 Phát hiện ngoại lệ dựa trên cụm sao sử dụng khoảng cách đến cụm gần nhất. Sử dụng phương pháp phân cụm k-mean, chúng ta có thể phân vùng các điểm dữ liệu được hiển thị trong Hình 11 thành ba cụm, như được hiển thị bằng các ký hiệu khác nhau. Tâm của mỗi cụm được đánh dấu bằng a.
Đối với mỗi đối tượng, o, chúng ta có thể gán một số điểm ngoại lệ cho đối tượng theo sự khác biệt giữa đối tượng và tâm gần nhất với đối tượng. Giả sử trung tâm gần nhất với o là co; thì khoảng cách giữa o và co là dist(o,co)/Ico và khoảng cách trung bình giữa co và các đối tượng được gán cho o là lco. Tỷ lệ dist (o, co) đo lường mức độ dist (o, co) nổi bật so với mức trung bình. Tỷ lệ này càng lớn, o càng xa trung tâm và càng có nhiều khả năng o là một ngoại lệ. Trong Hình 11, các điểm a, b và c tương đối xa các trung tâm tương ứng của chúng và do đó bị nghi ngờ là ngoại lệ. Cách tiếp cận này cũng có thể được sử dụng để phát hiện xâm nhập, như được mô tả trong Ví dụ 17.



Hình 5.4: điểm a là ngoại lai vì cách quá xa các cụm còn lại



Hình 5.5: điểm a,b,c các xa các tâm cụm

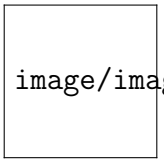
Ví dụ 12.17 Phát hiện xâm nhập bằng cách phát hiện ngoại lệ dựa trên cụm. Một phương thức bootstrap đã được phát triển để phát hiện sự xâm nhập vào dữ liệu kết nối TCP bằng cách xem xét sự giống nhau giữa các điểm dữ liệu và các cụm trong một tập dữ liệu huấn luyện. Phương pháp bao gồm ba bước.

1. Một tập dữ liệu huấn luyện được sử dụng để tìm ra các mẫu dữ liệu thông thường. Đặc biệt, dữ liệu kết nối TCP được phân đoạn theo ngày tháng. Các mục thường xuyên được tìm thấy trong mỗi phân khúc. Các mục thường xuyên nằm trong phần lớn các phân đoạn được coi là các mẫu dữ liệu thông thường và được gọi là các kết nối cơ sở.

2. Các kết nối trong dữ liệu huấn luyện có chứa các kết nối cơ sở được coi là không tấn công. Các kết nối như vậy được nhóm lại thành các nhóm.

3. Các điểm dữ liệu trong tập dữ liệu gốc được so sánh với các cụm được khai thác ở bước 2. Bất kỳ điểm nào được coi là ngoại lệ đối với các cụm được tuyên bố là một cuộc tấn công có thể xảy ra.

Ví dụ 12,18 Phát hiện các ngoại lệ trong các cụm nhỏ. Các điểm dữ liệu trong Hình 12 tạo thành ba cụm: cụm lớn, C1 và C2 và một cụm nhỏ, C3. Đối tượng o không thuộc về cụm nào. Sử dụng CBLOF, FindCBLOF có thể xác định o cũng như các điểm trong cụm C3 là ngoại lệ. Đối với o, cụm lớn gần nhất là C1. CBLOF đơn giản là sự tương đồng giữa o và C1, nhỏ. Đối với các điểm trong C3, cụm lớn gần nhất là C2. Mặc dù có ba điểm trong cụm C3, sự tương đồng giữa các điểm đó và cụm C2 là thấp và | C3 | = 3 là nhỏ; do đó, điểm số CBLOF của các điểm trong C3 là nhỏ. Phương pháp tiếp cận dựa trên cụm có thể phải chịu chi phí tính toán cao nếu chúng phải tìm cụm trước khi phát hiện các ngoại lệ. Một số kỹ thuật đã được phát triển để cải thiện hiệu quả. Ví dụ, phân cụm chiều rộng là một kỹ thuật thời gian tuyến tính được sử dụng trong một số phương pháp phát hiện ngoại lệ. Ý tưởng rất đơn giản nhưng hiệu quả. Một điểm được gán cho một cụm nếu tâm của cụm nằm trong ngưỡng khoảng cách được xác định trước từ điểm đó. Nếu một điểm không thể được chỉ định cho bất kỳ cụm hiện có, một cụm mới được tạo. Ngưỡng khoảng cách có thể được học từ dữ liệu đào tạo trong các điều kiện nhất định. Các phương pháp phát hiện ngoại lệ dựa trên cụm có những ưu điểm sau. Đầu tiên, họ có thể phát hiện các ngoại lệ mà không yêu cầu bất kỳ dữ liệu được dán nhãn nào, nghĩa là theo cách không được giám sát. Họ làm việc cho nhiều loại dữ liệu. Các cụm có thể được coi là tóm tắt của dữ liệu. Sau khi thu được các cụm, các phương thức dựa trên cụm chỉ cần so sánh bất kỳ đối tượng nào với các cụm để xác định xem đối tượng có phải là ngoại lệ hay không. Quá trình này thường nhanh vì số lượng cụm thường nhỏ so với tổng số đối tượng.



Hình 5.6: : Điểm ngoại lai trong cụm nhỏ

Một điểm yếu của phát hiện ngoại lệ dựa trên cụm là hiệu quả của nó, nó phụ thuộc nhiều vào phương pháp phân cụm được sử dụng. Các phương pháp như vậy có thể không được tối ưu hóa để phát hiện ngoại lệ. Các phương pháp phân cụm thường tốn kém cho các tập dữ liệu lớn, có thể đóng vai trò là nút cổ chai.

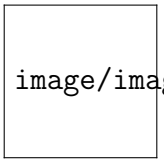
5.4 Các phương pháp dựa trên phân lớp

Phát hiện ngoại lệ có thể được coi là một vấn đề phân loại nếu có sẵn một bộ dữ liệu huấn luyện với nhãn lớp. Ý tưởng chung về các phương pháp phát hiện ngoại lệ dựa trên phân loại là đào tạo một mô hình phân loại có thể phân biệt dữ liệu bình thường với các ngoại lệ.
Hãy xem xét một tập huấn luyện có chứa các mẫu được dán nhãn là Bình thường và một số khác được dán nhãn là ngoại lệ. Sau đó, một lớp học có thể được xây dựng dựa trên tập huấn luyện. Bất kỳ phương pháp phân loại nào cũng có thể được sử dụng .Tuy nhiên, cách tiếp cận vũ phu này không hoạt động tốt để phát hiện ngoại lệ vì tập huấn luyện thường bị sai lệch nhiều. Đó là, số lượng mẫu bình thường có khả năng vượt xa số lượng mẫu ngoại lệ. Sự mất cân bằng này, trong đó số lượng mẫu ngoại lệ có thể không đủ, có thể ngăn chúng ta xây dựng một phân loại chính xác. Xem xét phát hiện xâm nhập trong một hệ thống, ví dụ. Bởi vì hầu hết các truy cập hệ thống là bình thường, rất dễ dàng để có được một đại diện tốt của các sự kiện bình thường. Tuy nhiên, không thể liệt kê tất cả các cuộc xâm nhập tiềm tàng, vì các nỗ lực mới và bất ngờ xảy ra theo thời gian. Do đó, chúng tôi chỉ còn lại một đại diện không thường xuyên của các mẫu ngoại lệ (hoặc xâm nhập).
Để vượt qua thách thức này, các phương pháp phát hiện ngoại lệ dựa trên phân loại thường sử dụng mô hình một lớp. Đó là, một lớp học được xây dựng để chỉ mô tả lớp bình thường. Bất kỳ mẫu nào không thuộc về lớp bình thường đều được coi là ngoại lệ.
Ví dụ 19 Phát hiện ngoại lệ bằng mô hình một lớp. Hãy xem xét tập huấn luyện được chỉ ra trong Hình 13, trong đó các điểm trắng là các mẫu được dán nhãn là bình thường và điểm đen là các mẫu được dán nhãn là ngoại lệ. Để xây dựng một mô hình để phát hiện ngoại lệ, chúng ta có thể tìm hiểu ranh giới quyết định của lớp thông thường bằng cách sử dụng phân loại các phương thức như SVM , như được minh họa. Đưa ra một đối tượng mới, nếu đối tượng nằm trong giới hạn quyết định của lớp bình thường, thì nó được coi là trường hợp bình thường. Nếu đối tượng nằm ngoài ranh giới quyết định, nó được tuyên bố là ngoại lệ.
Một lợi thế của việc chỉ sử dụng mô hình của lớp bình thường để phát hiện các ngoại lệ là mô hình có thể phát hiện các ngoại lệ mới có thể không xuất hiện gần với bất kỳ

đối tượng ngoại lệ nào trong tập huấn luyện. Điều này xảy ra miễn là các ngoại lệ mới như vậy nằm ngoài ranh giới quyết định của lớp bình thường.

Ý tưởng sử dụng ranh giới quyết định của lớp bình thường có thể được mở rộng để xử lý các tình huống trong đó các đối tượng bình thường có thể thuộc về nhiều lớp, chẳng hạn như trong cụm mờ. Ví dụ, AllElect Electronics chấp nhận các mặt hàng được trả lại. Người mua có thể trả lại các mặt hàng vì một số lý do (tương ứng với các loại lớp), chẳng hạn như lỗi thiết kế sản phẩm của sản phẩm và sản phẩm bị hư hỏng trong quá trình vận chuyển. Mỗi lớp như vậy được coi là bình thường. Để phát hiện các trường hợp ngoại lệ, AllElect Electronics có thể tìm hiểu một mô hình cho mỗi lớp bình thường. Để xác định xem một trường hợp có phải là ngoại lệ hay không, chúng ta có thể chạy từng mô hình trên vỏ. Nếu trường hợp không có bất kỳ mô hình nào, thì nó được tuyên bố là ngoại lệ.

Các phương pháp dựa trên phân loại và phương pháp phân cụm có thể được kết hợp để phát hiện các ngoại lệ theo cách học bán giám sát.



Hình 5.7: Học một model của lớp bình thường

Hình 5.8: Xác định điểm ngoại lai bằng học bán giám sát

Ví dụ 20 Phát hiện ngoại lệ bằng cách học bán giám sát. Hãy xem xét Hình 14, trong đó các đối tượng được gắn nhãn là bình thường và hay ngoại lệ, hay không có nhãn nào cả. Sử dụng cách tiếp cận dựa trên cụm, chúng tôi tìm thấy một cụm lớn, C và một cụm nhỏ, C1. Vì một số đối tượng trong C mang nhãn Bình thường, nên chúng tôi có thể coi tất cả các đối tượng trong cụm này (bao gồm cả các đối tượng không có nhãn) là các đối tượng bình thường. Chúng tôi sử dụng mô hình một lớp của cụm này để xác định các đối tượng bình thường trong phát hiện ngoại lệ. Tương tự, vì một số đối tượng trong cụm C1 mang nhãn hiệu ngoại lệ, nên chúng tôi khai báo tất cả các đối tượng trong C1 là ngoại lệ. Bất kỳ đối tượng nào không thuộc mô hình cho C (ví dụ: a) cũng được coi là ngoại lệ.

Các phương pháp dựa trên phân loại có thể kết hợp kiến thức miền của con người vào quá trình phát hiện bằng cách học hỏi từ các mẫu được dán nhãn. Khi mô hình phân loại được xây dựng, quá trình phát hiện ngoại lệ sẽ nhanh chóng. Nó chỉ cần so sánh các đối tượng được kiểm tra với mô hình đã học từ dữ liệu đào tạo. Chất lượng của các phương pháp dựa trên phân loại phụ thuộc rất nhiều vào tính khả dụng và chất lượng của tập huấn luyện. Trong nhiều ứng dụng, rất khó để có được dữ liệu đào tạo

đại diện và chất lượng cao, điều này hạn chế khả năng áp dụng các phương pháp dựa trên phân loại.

5.5 Phương pháp bán giám sát

Trong nhiều ứng dụng, mặc dù có được một số ví dụ được dán nhãn là khả thi, số lượng các ví dụ được dán nhãn như vậy thường rất ít. Chúng tôi có thể gặp các trường hợp chỉ có một tập hợp nhỏ các đối tượng bình thường và / hoặc ngoại lệ được gắn nhãn, nhưng hầu hết dữ liệu không được gắn nhãn. Các phương pháp phát hiện ngoại lệ được giám sát bán đã được phát triển để giải quyết các tình huống như vậy.
Trong nhiều ứng dụng, mặc dù có được một số ví dụ được dán nhãn là khả thi, số lượng các ví dụ được dán nhãn như vậy thường rất ít. Chúng tôi có thể gặp các trường hợp chỉ có một tập hợp nhỏ các đối tượng bình thường và / hoặc ngoại lệ được gắn nhãn, nhưng hầu hết dữ liệu không được gắn nhãn. Các phương pháp phát hiện ngoại lệ được giám sát bán đã được phát triển để giải quyết các tình huống như vậy.
Nếu chỉ có một số ngoại lệ được dán nhãn là có sẵn, phát hiện ngoại lệ được giám sát bán là khó khăn. Một số lượng nhỏ các ngoại lệ được dán nhãn không có khả năng đại diện cho tất cả các ngoại lệ có thể. Do đó, việc xây dựng một mô hình cho các ngoại lệ chỉ dựa trên một vài ngoại lệ được dán nhãn là không có hiệu quả. Để cải thiện chất lượng phát hiện ngoại lệ, chúng tôi có thể nhận trợ giúp từ các mô hình cho các đối tượng bình thường học được từ các phương pháp không giám sát.
Để biết thêm thông tin về các phương pháp bán giám sát, độc giả quan tâm được tham khảo các ghi chú thư mục ở cuối chương này

5.6 Phương pháp giám sát(Supervised Methods)

Phương pháp giám sát mô hình dữ liệu bình thường và bất thường. Các chuyên gia kiểm tra và dán nhãn một mẫu của dữ liệu cơ bản. Phát hiện ngoại lệ sau đó có thể được mô hình hóa thành một vấn đề phân loại. Nhiệm vụ là học một lớp có thể nhận ra các ngoại lệ. Mẫu được sử dụng cho đào tạo và thử nghiệm. Trong một số ứng dụng, các chuyên gia có thể chỉ dán nhãn cho các đối tượng bình thường và bất kỳ đối tượng nào khác không phù hợp với mô hình của các đối tượng bình thường được báo cáo là ngoại lệ. Các phương thức khác mô hình hóa các ngoại lệ và xử lý các đối tượng không khớp với mô hình của các ngoại lệ như bình thường.
Mặc dù nhiều phương pháp phân loại có thể được áp dụng, những thách thức đối với việc phát hiện ngoại lệ có giám sát bao gồm:

- Hai lớp (tức là, các đối tượng bình thường so với các ngoại lệ) không cân bằng. Đó là, dân số của các ngoại lệ thường nhỏ hơn nhiều so với các đối tượng bình thường. Do đó, các phương pháp xử lý các lớp không cân bằng có thể được sử dụng, chẳng hạn như lấy mẫu quá mức (tức là, sao chép) các ngoại lệ để tăng phân phối của chúng trong tập huấn luyện được sử dụng để xây dựng lớp. Do số lượng nhỏ các ngoại lệ trong dữ liệu, dữ liệu mẫu được kiểm tra bởi các chuyên gia tên miền và được sử dụng trong đào tạo thậm chí có thể không đại diện cho

phân phối ngoại lệ. Việc thiếu các mẫu ngoại lệ có thể hạn chế khả năng của các lớp được xây dựng như vậy. Để giải quyết những vấn đề này, một số phương pháp tạo ra các ngoại lệ của nghệ thuật.

- Trong nhiều ứng dụng phát hiện ngoại lệ, việc bắt càng nhiều ngoại lệ càng tốt (nghĩa là độ nhạy hoặc thu hồi phát hiện ngoại lệ) quan trọng hơn nhiều so với việc không gắn nhãn sai đối tượng bình thường như ngoại lệ. Do đó, khi một phương pháp phân loại được sử dụng để phát hiện ngoại lệ có giám sát, nó phải được giải thích một cách thích hợp để xem xét lợi ích của ứng dụng khi thu hồi.

Tóm lại, các phương pháp được phát hiện ngoại lệ được giám sát phải cẩn thận trong cách chúng huấn luyện và cách chúng diễn giải tỷ lệ phân loại do thực tế là các ngoại lệ rất hiếm so với các mẫu dữ liệu khác.

5.7 Phương pháp không giám sát

Trong một số trường hợp ứng dụng, các đối tượng được gắn nhãn là bình thường, hoặc không có sẵn. Vì vậy, một phương pháp học tập không giám sát phải được sử dụng.
Các phương pháp phát hiện ngoại lệ không được giám sát đưa ra một giả định ngầm định: Các đối tượng bình thường có phần bị nhóm lại. Khác Nói cách khác, một phương pháp phát hiện ngoại lệ không được giám sát hy vọng rằng các đối tượng bình thường theo mô hình thường xuyên hơn nhiều so với các ngoại lệ. Các đối tượng bình thường không phải rơi vào một nhóm có độ tương tự cao. Thay vào đó, họ có thể tạo thành nhiều nhóm, trong đó mỗi nhóm có các tính năng riêng biệt. Tuy nhiên, một ngoại lệ dự kiến sẽ xảy ra ở rất xa trong không gian đặc trưng từ bất kỳ nhóm đối tượng bình thường nào. Giả định này có thể không đúng mọi lúc. Ví dụ, trong Hình 2, các đối tượng bình thường không chia sẻ bất kỳ mẫu mạnh nào. Thay vào đó, chúng được phân phối đồng đều. Các ngoại lệ tập thể, tuy nhiên, chia sẻ sự tương đồng cao trong một khu vực nhỏ. Phương pháp không giám sát có thể phát hiện các ngoại lệ như vậy một cách hiệu quả. Trong một số ứng dụng, các đối tượng bình thường được phân phối đa dạng và nhiều đối tượng như vậy không tuân theo các mẫu mạnh. Ví dụ, trong một số vấn đề phát hiện xâm nhập và phát hiện vi-rút máy tính, các hoạt động bình thường rất đa dạng và nhiều trường hợp không thuộc các cụm chất lượng cao. Trong các trường hợp như vậy, các phương thức không được giám sát có thể có tỷ lệ dương tính giả cao, chúng có thể đánh dấu sai nhiều đối tượng bình thường là ngoại lệ (xâm nhập hoặc vi rút trong các ứng dụng này) và để nhiều ngoại lệ thực sự không bị phát hiện. Do sự giống nhau cao giữa sự xâm nhập và vi-rút (tức là, chúng phải tấn công các tài nguyên chính trong các hệ thống đích), mô hình hóa các ngoại lệ bằng các phương pháp được giám sát có thể hiệu quả hơn nhiều.
Nhiều phương pháp phân cụm có thể được điều chỉnh để hoạt động như các phương pháp phát hiện ngoại lệ không được giám sát. Ý tưởng trung tâm là các cụm đầu tiên, và sau đó các đối tượng dữ liệu không thuộc về bất kỳ cụm nào được phát hiện là ngoại lệ. Tuy nhiên, các phương pháp như vậy bị hai vấn đề. Đầu tiên, một đối tượng dữ liệu không thuộc bất kỳ cụm nào có thể bị nhiễu thay vì ngoại lệ. Thứ hai, thường

tốn kém cho các cụm thứ nhất và sau đó là các ngoại lệ. Người ta thường cho rằng có số lượng ngoại lệ ít hơn nhiều so với các đối tượng bình thường. Phải xử lý một lượng lớn các mục nhập dữ liệu không nhắm mục tiêu (tức là, các đối tượng bình thường) trước khi người ta có thể chạm vào thịt thật (tức là, các ngoại lệ) có thể không hấp dẫn. Các phương pháp phát hiện ngoại lệ không giám sát mới nhất phát triển các ý tưởng thông minh khác nhau để giải quyết trực tiếp các ngoại lệ mà không cần cụm rõ ràng và hoàn toàn.

5.8 Tổng kết

- Giả sử rằng một quy trình thống kê nhất định được sử dụng để tạo ra một tập hợp các đối tượng dữ liệu. Một ngoại lai là một đối tượng dữ liệu làm lệch đáng kể so với phần còn lại của các đối tượng, như thể nó được tạo ra bởi một cơ chế khác.

- **Các loại ngoại lai** bao gồm các ngoại lệ toàn cầu, các ngoại lệ theo ngữ cảnh và các ngoại lệ tập thể. Một đối tượng có thể nhiều hơn một loại ngoại lệ.

- **Ngoại lai toàn cục** là hình thức đơn giản nhất của ngoại lệ và dễ phát hiện nhất. Một ngoại lệ theo ngữ cảnh làm sai lệch đáng kể đối với bối cảnh cụ thể của vật thể (ví dụ: giá trị nhiệt độ Toronto là 28C là ngoại lệ nếu xảy ra trong bối cảnh mùa đông). Một tập hợp con của các đối tượng dữ liệu tạo thành một tập thể ngoại lệ nếu toàn bộ các đối tượng sai lệch đáng kể so với toàn bộ tập dữ liệu, mặc dù các đối tượng dữ liệu riêng lẻ có thể không nằm ngoài. Phát hiện ngoại lệ tập thể đòi hỏi thông tin cơ bản để mô hình hóa các mối quan hệ giữa các đối tượng với các nhóm ngoại lệ.

- **Những thách thức trong phát hiện ngoại lai** Những thách thức trong phát hiện ngoại lai

- Các phương pháp phát hiện ngoại lệ có thể được phân loại theo liệu mẫu dữ liệu để phân tích có được cung cấp với các nhãn do chuyên gia cung cấp có thể được sử dụng để xây dựng mô hình phát hiện ngoại lệ hay không. Trong trường hợp này, các phương pháp phát hiện được giám sát, bán giám sát hoặc không giám sát. Ngoài ra, các phương pháp phát hiện ngoại lệ có thể được tổ chức theo các giả định của chúng đối với các đối tượng bình thường so với bên ngoài. Phân loại này bao gồm các phương pháp thống kê, phương pháp dựa trên vùng lân cận và phương pháp dựa trên cụm.

- **Các phương pháp phát hiện ngoại lệ thống kê (hoặc phương pháp dựa trên mô hình)** giả định rằng các đối tượng dữ liệu bình thường theo mô hình thống kê, trong đó dữ liệu không theo mô hình được coi là ngoại lệ. Các phương pháp như vậy có thể là tham số (họ cho rằng dữ liệu được tạo bởi phân phối tham số) hoặc không tham số (họ học một mô hình cho dữ liệu, thay vì giả sử là một tiên nghiệm). Các phương pháp tham số cho dữ liệu đa biến có thể sử dụng khoảng cách Mahalanobis, thống kê ξ 2 hoặc hỗn hợp các mô hình tham số thứ

cấp. Biểu đồ và ước tính mật độ hạt nhân là ví dụ về các phương pháp không tham số.

- **Các phương pháp phát hiện ngoại lai dựa trên phân cụm** cho rằng các đối tượng dữ liệu bình thường thuộc các cụm lớn và dày đặc, trong khi các ngoại lệ thuộc về các cụm nhỏ hoặc thưa thớt hoặc không thuộc về bất kỳ cụm nào.
- **Các phương pháp phát hiện ngoại lai dựa trên phân loại** thường sử dụng mô hình một lớp. Đó là, một lớp học được xây dựng để chỉ mô tả lớp bình thường. Bất kỳ mẫu nào không thuộc về lớp bình thường đều được coi là ngoại lệ.

Tài liệu tham khảo

[1] Jiawei Han, Micheline Kamber, Jian Pei,
Data Mining - Concepts and Techniques (Third Edition)