BÀI 6. MỘT SỐ HỆ THỐNG VÀ CÔNG NGHỆ TRÍ TUỆ NHÂN TẠO

1. Một số hệ thống trí tuệ nhân tạo

1.1. Thị giác máy tính

Thị giác máy tính là một lĩnh vực trong Artificial Intelligence Trí tuệ nhân tạo) và Computer Science (Khoa học máy tính) nhằm giúp máy tính có được khả năng nhìn và hiểu giống như con người.

Thị giác máy tính (computer vision) được định nghĩa là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh và, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng. Thị giác máy tính cũng được mô tả là sự tổng thể của một dải rộng các quá trình tự động và tích hợp và các thể hiện cho các nhận thức thị giác

Quà trình mô phỏng thị giác con người được chia thành 3 giai đoạn nối tiếp (tương tự cách con người nhìn): mô phỏng mắt (thu nhận - khó), mô phỏng vỏ não thị giác (xử lý - rất khó) và mô phỏng phần còn lại của bộ não (phân tích - khó nhất).

Thu nhận

Mô phỏng mắt là lĩnh vực đạt được nhiều thành công nhất. Chúng ta đã tạo ra các cảm biến, vi xử lý hình ảnh giống khả năng nhìn của mắt người và thậm chí còn tốt hơn.

Camera có thể chụp hàng ngàn ảnh mỗi giây và nhận diện từ xa với độ chính xác cao. Tuy nhiên cảm biến camera tốt nhất cũng không thể nhận diện được một quả bóng chứ đừng nói là bắt được chúng. Nói cách khác, phần cứng bị giới hạn khi không có phần mềm - đến giờ vẫn là khó khăn lớn nhất. Tuy vậy, camera ngày nay cũng khá linh hoạt và làm nền tảng tốt để nghiên cứu.

Mô tả

Bộ não được xây dựng từ con số 0 với các hình ảnh dần dần lấp đầy tâm trí, nó làm nhiệm vụ liên quan tới thị giác nhiều hơn bất kì công việc nào khác và việc này đều xuống tới cấp độ tế bào. Hàng tỉ tế bào phối hợp để lấy ra các hình mẫu, bắt được tín hiệu.

Một nhóm nơ-ron sẽ báo cho nhóm khác khi có sự khác biệt dọc theo một đường thẳng (theo một góc nào đó, như chuyển động nhanh hơn hay theo một hướng khác). Nghiên cứu đầu tiên về thị giác máy tính cho rằng mạng lưới nơ-ron phức tạp tới nỗi không thể hiểu nổi khi tiếp cận theo hướng lý giải từ trên xuống dưới. Với một số đối tượng thì cách này cũng hiệu quả nhưng khi mô tả từng đối tượng, từ nhiều góc nhìn, nhiều biến thể về màu sắc, chuyển động và nhiều thứ khác thì hãy hình dung sẽ khó thế nào. Ngay cả mức

nhận thức của một em bé cũng sẽ cần lượng dữ liệu lớn vô cùng. Cách tiếp cận từ dưới lên bắt chước cách não bộ hoạt động có vẻ hứa hẹn hơn. Những năm qua chứng kiến sự bùng nổ của các nghiên cứu và sử dụng hệ thống này trong việc bắt chước não người. Quá trình nhận diện hình mẫu vẫn đang tăng tốc và chúng ta vẫn liên tục đạt được tiến bộ.

Thấu hiểu

Ta có thể xây dựng một hệ thống nhận diện được một quả táo, từ bất cứ góc nào, trong bất kì tình huống nào, dù đứng im hay chuyển động nhưng chúng không thể nhận diện được một quả cam, không thể nói cho ta quả táo là gì, có ăn được không, lớn nhỏ ra sao hay dùng để làm gì. Như vậy phần cứng và phần mềm tốt cũng không làm được gì nếu không có hệ điều hành.

Đó chính là phần còn lại của bộ não: bộ nhớ ngắn/dài hạn, dữ liệu từ các giác quan, sự chú ý, nhận thức, bài học khi tương tác với thế giới… được viết lên mạng lưới nơ-ron kết nối phức tạp hơn bất cứ thứ gì chúng ta từng thấy, theo cách mà chúng ta không thể hiểu. Đó là nơi mà khoa học máy tính và trí tuệ nhân tạo gặp mặt.

Dù mới trong thời kì sơ khai, thị giác máy tính vẫn vô cùng hữu ích. Nó có mặt trong camera nhận diện khuôn mặt (Face ID) và nụ cười. Nó giúp xe tự lái nhận diện biển báo, người đi đường. Nó nằm trong các robot trong nhà máy, nhận diện sản phẩm, truyền cho con người.

1.2. Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Xử lý ngôn ngữ tự nhiên ra đời từ những năm 40 của thế kỷ 20, trải qua các giai đoạn phát triển với nhiều phương pháp và mô hình xử lý khác nhau. Có thể kể tới các phương pháp sử dụng ô-tô-mát và mô hình xác suất (những năm 50), các phương pháp dựa trên ký hiệu, các phương pháp ngẫu nhiên (những năm 70), các phương pháp sử dụng học máy truyền thống (những năm đầu thế kỷ 21), và đặc biệt là sự bùng nổ của học sâu trong thập kỷ vừa qua.

Xử lý ngôn ngữ tự nhiên có thể được chia ra thành hai nhánh lớn, không hoàn toàn độc lập, bao gồm xử lý tiếng nói (speech processing) và xử lý văn bản (text processing). Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý

ngôn ngữ của con người ở dạng tiếng nói (dữ liệu âm thanh). Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Nếu như nhận dạng tiếng nói là chuyển ngôn ngữ từ dạng tiếng nói sang dạng văn bản thì ngược lại, tổng hợp tiếng nói chuyển ngôn ngữ từ dạng văn bản thành tiếng nói. Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Các ứng dụng quan trọng của xử lý văn bản bao gồm tìm kiếm và truy xuất thông tin, dịch máy, tóm tắt văn bản tự động, hay kiểm lỗi chính tả tự động. Xử lý văn bản đôi khi được chia tiếp thành hai nhánh nhỏ hơn bao gồm hiểu văn bản và sinh văn bản. Nếu như hiểu liên quan tới các bài toán phân tích văn bản thì sinh liên quan tới nhiệm vụ tạo ra văn bản mới như trong các ứng dụng về dịch máy hoặc tóm tắt văn bản tự động.

Xử lý văn bản bao gồm 4 bước chính sau:

1. **Phân tích hình vị:** là sự nhận biết, phân tích, và miêu tả cấu trúc của hình vị trong một ngôn ngữ cho trước và các đơn vị ngôn ngữ khác, như từ gốc, biên từ, phụ tố, từ loại, v.v. Trong xử lý tiếng Việt, hai bài toán điển hình trong phần này là tách từ (word segmentation) và gán nhãn từ loại (part-of-speech tagging).

2. **Phân tích cú pháp:** là quy trình phân tích một chuỗi các biểu tượng, ở dạng ngôn ngữ tự nhiên hoặc ngôn ngữ máy tính, tuân theo văn phạm hình thức. Văn phạm hình thức thường dùng trong phân tích cú pháp của ngôn ngữ tự nhiên bao gồm Văn phạm phi ngữ cảnh (Context-free grammar – CFG), Văn phạm danh mục kết nối (Combinatory categorial grammar – CCG), và Văn phạm phụ thuộc (Dependency grammar – DG). Đầu vào của quá trình phân tích là một câu gồm một chuỗi từ và nhãn từ loại của chúng, và đầu ra là một cây phân tích thể hiện cấu trúc cú pháp của câu đó.

3. **Phân tích ngữ nghĩa:** là quá trình liên hệ cấu trúc ngữ nghĩa, từ cấp độ cụm từ, mệnh đề, câu và đoạn đến cấp độ toàn bài viết, với ý nghĩa độc lập của chúng. Nói cách khác, việc này nhằm tìm ra ngữ nghĩa của đầu vào ngôn từ. Phân tích ngữ nghĩa bao gồm hai mức độ: Ngữ nghĩa từ vựng biểu hiện các ý nghĩa của những từ thành phần, và phân biệt nghĩa của từ; Ngữ nghĩa thành phần liên quan đến cách thức các từ liên kết để hình thành những nghĩa rộng hơn.

4. **Phân tích diễn ngôn:** là phân tích văn bản có xét tới mối quan hệ giữa ngôn ngữ và ngữ cảnh sử dụng (context-of-use). Phân tích diễn ngôn, do đó, được thực hiện ở mức độ đoạn văn hoặc toàn bộ văn bản thay vì chỉ phân tích riêng ở mức câu.

Một Số Ứng Dụng Của NLP

NLP ngày càng được ứng dụng nhiều. Một số ứng dụng có thể kể đến như:

1. **Nhận dạng tiếng nói** (Automatic Speech Recognition – ASR, hoặc Speech To Text – STT) chuyển đổi ngôn ngữ từ dạng tiếng nói sang dạng văn bản, thường được ứng dụng trong các chương trình điều khiển qua giọng nói.

2. **Tổng hợp tiếng nói** (Speech synthesis hoặc Text to Speech – TTS) chuyển đổi ngôn ngữ từ dạng văn bản sang tiếng nói, thường được dùng trong đọc văn bản tự động.

3. **Truy xuất thông tin** (Information Retrieval – IR) có nhiệm vụ tìm các tài liệu dưới dạng không có cấu trúc (thường là văn bản) đáp ứng nhu cầu về thông tin từ những nguồn tổng hợp lớn. Những hệ thống truy xuất thông tin phổ biến nhất bao gồm các công cụ tìm kiếm như Google, Yahoo, hoặc Bing search. Những công cụ này cho phép tiếp nhận một câu truy vấn dưới dạng ngôn ngữ tự nhiên làm đầu vào và cho ra một danh sách các tài liệu được sắp xếp theo mức độ phù hợp.

4. **Trích chọn thông tin** (Information Extraction – IE) nhận diện một số loại thực thể được xác định trước, mối quan hệ giữa các thực thể và các sự kiện trong văn bản ngôn ngữ tự nhiên. Khác với truy xuất thông tin trả về một danh sách các văn bản hợp lệ thì trích chọn thông tin trả về chính xác thông tin mà người dùng cần. Những thông tin này có thể là về con người, địa điểm, tổ chức, ngày tháng, hoặc thậm chí tên công ty, mẫu sản phẩm hay giá cả.

5. **Trả lời câu hỏi** (Question Answering – QA) có khả năng tự động trả lời câu hỏi của con người ở dạng ngôn ngữ tự nhiên bằng cách truy xuất thông tin từ một tập hợp tài liệu. Một hệ thống QA đặc trưng thường bao gồm ba mô đun: Mô đun xử lý truy vấn (Query Processing Module) – tiến hành phân loại câu hỏi và mở rộng truy vấn; Mô đun xử lý tài liệu (Document Processing Module) – tiến hành truy xuất thông tin để tìm ra tài liệu thích hợp; và Mô hình xử lý câu trả lời (Answer Processing Module) – trích chọn câu trả lời từ tài liệu đã được truy xuất.

6. **Tóm tắt văn bản tự động** (Automatic Text Summarization) là bài toán thu gọn văn bản đầu vào để cho ra một bản tóm tắt ngắn gọn với những nội dung quan trọng nhất của văn bản gốc. Có hai phương pháp chính trong tóm tắt, là phương pháp trích xuất (extractive) và phương pháp tóm lược ý (abstractive). Những bản tóm tắt trích xuất được hình thành bằng cách ghép một số câu được lấy y nguyên từ văn bản cần thu gọn. Những bản tóm lược ý thường truyền đạt những thông tin chính của đầu vào và có thể sử dụng lại những cụm từ hay mệnh đề trong đó, nhưng nhìn chung được thể hiện ở ngôn ngữ của người tóm tắt.

7. **Chatbot** là việc chương trình máy tính có khả năng trò chuyện (chat), hỏi đáp với con người qua hình thức hội thoại dưới dạng văn bản (text). Chatbot thường được sử

dụng trong ứng dụng hỗ trợ khách hàng, giúp người dùng tìm kiếm thông tin sản phẩm, hoặc giải đáp thắc mắc.

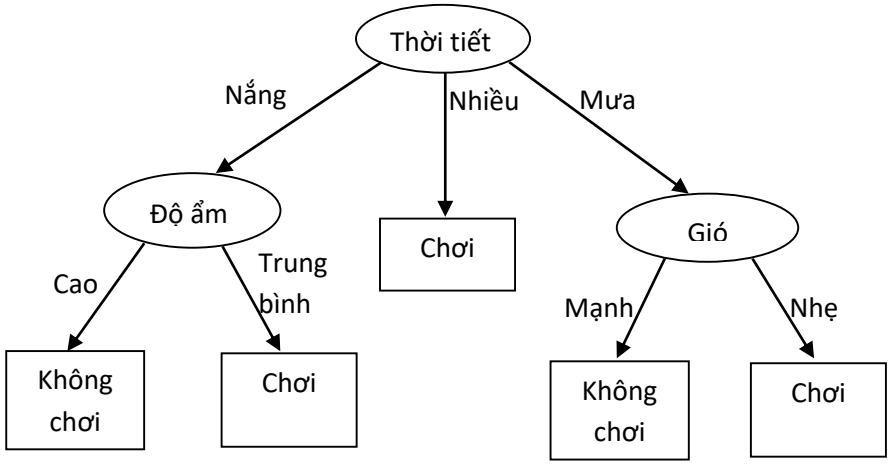
8. **Dịch máy** (Machine Translation – MT) là việc sử dụng máy tính để tự động hóa một phần hoặc toàn bộ quá trình dịch từ ngôn ngữ này sang ngôn ngữ khác. Các phương pháp dịch máy phổ biến bao gồm dịch máy dựa trên ví dụ (example-based machine translation – EBMT), dịch máy dựa trên luật (rule-based machine translation – RBMT), dịch máy thống kê (statistical machine translation – SMT), và dịch máy sử dụng mạng nơ-ron (neural machine translation).

9. **Kiểm lỗi chính tả tự động** là việc sử dụng máy tính để tự động phát hiện các lỗi chính tả trong văn bản (lỗi từ vựng, lỗi ngữ pháp, lỗi ngữ nghĩa) và đưa ra gợi ý cách chỉnh sửa lỗi.

2. Các công nghệ trí tuệ nhân tạo

2.1. Cây quyết định

Cây quyết định được dùng để đưa ra tập luật if – then nhằm mục đích dự báo, giúp con người nhận biết về tập dữ liệu. Cây quyết định cho phép phân loại đối tượng tuỳ thuộc vào các điều kiện tại các nút trong cây, bắt đầu từ gốc cây tới các nút sát lá-Nút xác định phân loại đối tượng. Mỗi nút trong của cây xác định điều kiện đối với thuộc tính mô tả của đối tượng. Mỗi nhánh tương ứng với điều kiện: Nút (thuộc tính) bằng giá trị nào đó. Đối tượng được phân loại nhờ tích hợp các điều kiện bắt đầu từ nút gốc của cây và các thuộc tính mô tả với giá trị của thuộc tính đối tượng.



Hình 4.1:Một ví dụ về cây quyết định

Hình 4.1 là cây quyết định phân loại xem thời tiết như thế nào thì phù hợp với việc chơi tennis.

a) Tạo cây quyết định

Xét bảng dữ liệu T = (A, D) trong đó A = {A_1, A_2,..., A_n } là tập thuộc tính dẫn xuất, D = {r_1, r_2, ..., r_n} là thuộc tính mục tiêu. Vấn đề đặt ra là trong tập thuộc tính A ta phải chọn thuộc tính nào để phân hoạch? Một trong các phương pháp đó là dựa vào độ lợi thông tin. Hay còn gọi là thuật giải ID3.

Lựa chọn chủ yếu trong giải thuật ID3 là chọn thuộc tính nào để đưa vào mỗi nút trong cây. Ta sẽ chọn thuộc tính phân rã tập mẫu tốt nhất. Thước đo độ tốt của việc chọn lựa thuộc tính là gì? Ta cần xác định một độ đo thống kê, gọi là thông tin thu được, đánh giá từng thuộc tính được chọn tốt như thế nào còn phụ thuộc vào việc phân loại mục tiêu của tập mẫu. ID3 sử dụng thông tin thu được đánh giá để chọn ra thuộc tính cho mỗi bước giữa những thuộc tính ứng viên, trong quá trình phát triển cây.

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Để đánh giá chính xác thông tin thu được, dùng Entropy(S): Độ bất định (độ pha trộn/độ hỗn tạp) của S liên quan đến sự phân loại đang xét

Trong đó p_i là xác suất xuất hiện trạng thái i của hệ thống. Theo lý thuyết thông tin: mã có độ dài tối ưu là mã gán $-\log_2 p$ bits cho thông điệp có xác suất là p. S là một tập huấn luyện.

Nếu gọi $p_\oplus$ là xác suất xuất hiện các ví dụ dương trong tập S, $p_\ominus$ là xác suất xuất hiện các ví dụ âm trong tập S. Entropy đo độ bất định của tập S sẽ là:

$$Entropy(S) = -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

Quy định 0.log0 = 0

Chẳng hạn với tập S gồm 14 mẫu có chung một vài giá trị logic gồm 9 mẫu dương và 5 mẫu âm. Khi đó đại lượng Entropy của tập S liên quan đến sự phân loại logic này là:

Entropy([9+, 5-]) = - (9/14)$\log_2$(9/14) - (5/14)$\log_2$(5/14) = 0,940

Chú ý :

Đại lượng Entropy = 0 nếu tất cả thành viên của tập S cùng thuộc một lớp (vì nếu tất cả là dương (P+ = 1), do đó P- = 0, Entropy(S) = $-1\log_2 1 - 0\log_2 0 = 0$).

Đại lượng Entropy(S) = 1 khi tập S chứa tỉ lệ tập mẫu âm và mẫu dương là như nhau. Nếu tập S chứa tập mẫu âm và tập mẫu dương có tỉ lệ P+ khác P- thì Entropy(S) $\in$ (0,1).

Dựa trên sự xác định entropy, ta tính Gain(S, A) = Lượng giảm entropy mong đợi qua việc chia các ví dụ theo thuộc tính A

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

Ví dụ 4.1: Xem xét nhiệm vụ học được đưa ra bởi tập mẫu dưới đây , thuộc tính mục tiêu ở đây là: chơi tennis có giá trị là *có* hoặc *không*, giá trị thuộc tính này dự đoán dựa vào các thuộc tính mô tả.

Ngày	**Thời tiết**	**Nhiệt độ**	**Độ ẩm**	**Gió**	**Chơi tennis**
D1	Nắng	Nóng	Cao	Nhẹ	Không
D2	Nắng	Nóng	Cao	Mạnh	Không
D3	Nhiều mây	Nóng	Cao	Nhẹ	Có
D4	Mưa	Dễ chịu	Cao	Nhẹ	Có
D5	Mưa	Lạnh	Trung bình	Nhẹ	Có
D6	Mưa	Lạnh	Trung bình	Mạnh	Không
D7	Nhiều mây	Lạnh	Trung bình	Mạnh	Có
D8	Nắng	Dễ chịu	Cao	Nhẹ	Không
D9	Nắng	Lạnh	Trung bình	Nhẹ	Có
D10	Mưa	Dễ chịu	Trung bình	Nhẹ	Có
D11	Nắng	Dễ chịu	Trung bình	Mạnh	Có
D12	Nhiều mây	Dễ chịu	Cao	Mạnh	Có
D13	Nhiều mây	Nóng	Trung bình	Nhẹ	Có
D14	Mưa	Dễ chịu	Cao	Mạnh	Không

Giải quyết bước đầu tiên của giải thuật, tạo nút đỉnh của cây quyết định. Nên đưa thuộc tính nào vào cây đầu tiên? ID3 xác định thông tin thu được cho mỗi thuộc tính ứng cử (thời tiết, nhiệt độ, độ ẩm và gió) sau đó chọn một trong số đó mà có thông tin thu được cao nhất.

Giá trị thông tin thu được cho mỗi thuộc tính là:

Gain(S, thời tiết) = 0,246

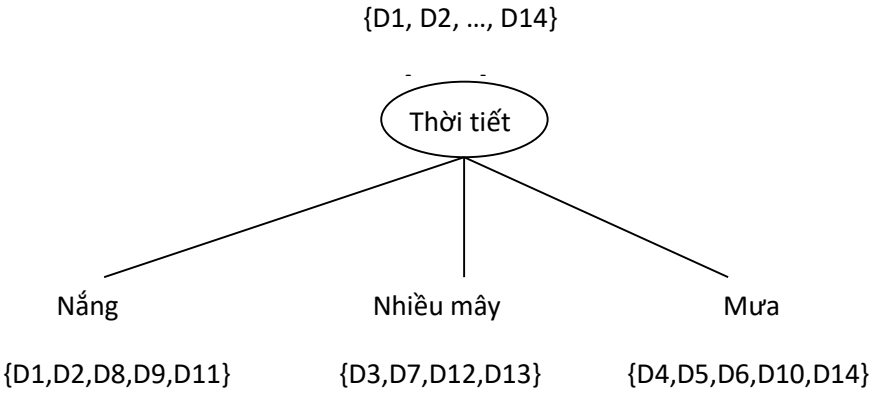
Gain(S, độ ẩm) = 0,151

Gain(S, gió) = 0,048

Gain(S, nhiệt độ) = 0,029

Trong đó tập S là tập mẫu ở bảng trên

Theo đánh giá thông tin thu được, thuộc tính thời tiết cung cấp dự đoán tốt nhất về thuộc tính mục tiêu "*chơi tennis"* trên tập mẫu. Do đó, thuộc tính "*thời tiết"* được chọn là thuộc tính quyết định cho nút gốc, nhánh được tạo ra dưới nút gốc tương ứng với mỗi giá trị của thuộc tính thời tiết (như nắng, mưa, nhiều mây) cùng với tập mẫu sẽ them vào mỗi nút con mới.



Hình 4.2. Cây quyết định sau lần phân hoạch đầu tiên

Mọi mẫu mà có *thời tiết* = 'nhiều mây' thì là mẫu dương với thuộc tính chơi tennis. Do vậy nút này trở thành nút lá với sự phân loại thuộc tính *chơi tennis* = 'Có'. Trái lại với những nút con tương ứng với *thời tiết* = 'nắng' và "*thời tiết*" = 'mưa' có giá trị Entropy ≠ 0 và cây quyết định sẽ phát triển xa hơn dưới những nút này.

Quá trình chọn thuộc tính mới để phân loại tập mẫu lặp lại cho mỗi nút con. Lúc này chỉ sử dụng những mẫu có liên quan tới nút này. Những thuộc tính mô tả có sự kết hợp chặt chẽ hơn trong cây đã được ngăn chặn. Bởi vậy mà bất kì thuộc tính đưa ra nào có thể xuất hiện theo bất kì nhánh nào của cây. Quá trình xử lí còn tiếp cho mỗi nút lá mới cho đến khi hai điều kiện sau thoả mãn: Tập thuộc tính rỗng (mọi thuộc tính đều đã nằm dọc theo những nhánh của cây) hoặc tất cả những mẫu có liên quan với nút lá này có cùng giá trị thuộc tính mục tiêu (giá trị entropy của chúng = 0).

2.2. **Học dựa trên xác suất**

Kỹ thuật này có thể hiểu đơn giản như sau: với một mẫu dữ liệu cần phân lớp, ta tính xác suất có điều kiện để mẫu dữ liệu đó rơi vào từng lớp trong tập các lớp đã biết trước. Mẫu dữ liệu sẽ được phân vào lớp nào có xác suất cao nhất.

a) Một số khái niệm ban đầu

Hiện tượng tất yếu: là những hiện tượng nếu được thực hiện ở điều kiện giống nhau thì cho kết quả giống nhau. Chẳng hạn khi đun nước đến 100^0C thì nước sôi. Hiện tượng tất yếu là đối tượng nghiên cứu của Vật lý, Hóa học.

Hiện tượng ngẫu nhiên: là những hiện tượng dù đã được quan sát ở điều kiện giống nhau, nhưng kết quả có thể khác nhau. Ví dụ: tung đồng xu, và quan sát xem đồng xu là "*sấp*" hay "*ngửa*". Hiện tượng ngẫu nhiên là đối tượng nghiên cứu của xác suất học.

Trong một hiện tượng ngẫu nhiên ta không thể biết được chắc chắn kết quả xảy ra như thế nào, nhưng có thể hình dung ra được các khả năng mà kết quả có thể xảy ra. Tập hợp tất cả các kết quả có thể xảy ra được gọi là không gian mẫu, ký hiệu là Ω. Ví dụ: tung một đồng xu, Ω = {*sấp, ngửa*}; tung một con xúc sắc và tính điểm, Ω = {1, 2, 3, 4, 5, 6} ...

Biến cố: là một tập con của không gian mẫu, ký hiệu là: *A, B, C*... Ví dụ: **t**ung một con xúc sắc, gọi *A* là biến cố được số điểm chẵn và *B* là biến cố được số điểm lẻ thì *A* = {2, 4, 6}, *B* = {1, 3, 5}. Vì các biến cố là các tập hợp, nên ta thường sử dụng các phép tính trên tập hợp cho biến cố:

- Phép hội : $A \cup B$ (*A* hay *B* xảy ra).
- Phép giao: $A \cap B = AB$ (*A* và *B* xảy ra).
- Phép bù: $\overline{A} = \Omega \setminus A$ (*A* không xảy ra).

Khi quan sát các hiện tượng, ta thấy có những hiện tượng thường xuyên xảy ra, có những hiện tượng ít xảy ra. Xác suất là một con số đo lường mức độ thường xuyên xảy ra của một biến cố. Xác suất xảy ra một biến cố *A* (hay xác suất của *A*), ký hiệu là *P(A)* là tỷ lệ giữa số lần biến cố *A* xảy ra và số lượng tất cả các biến cố:

$$\mathrm{P(A)} = \frac{|A|}{|\Omega|}. \qquad (4.1)$$

Tính chất cơ bản của xác suất:

$0 \le P(A) \le 1$;

$P(true) = 1$;

$P(false) = 0$;

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

Xác suất có điều kiện:

P(A|B) là phần của không gian mà trong đó *A* là đúng, với điều kiện (đã biết) là *B* đúng. Nói cách khác, P(*A*|*B*) là xác suất xảy ra biến cố *A* với điều kiện là có xảy ra biến cố *B*, thường được gọi là "*xác suất của A nếu có B*". Ví dụ:

A: *Tôi sẽ đi đá bóng vào ngày mai,*
B: Trời sẽ không mưa vào ngày mai,

P(A|B): Xác suất của việc tôi sẽ đi đá bóng vào ngày mai nếu (đã biết rằng) trời sẽ không mưa vào ngày mai.

Gọi **P(A, B)** là xác suất xảy ra đồng thời hai sự kiện *A* và *B* và *P(B)* là xác suất xảy ra sự kiện *B*. Dễ dàng thấy rằng:

$$P(A \mid B) = \frac{P(A,B)}{P(B)}. \quad (4.2)$$

Công thức xác suất toàn phần:

Nếu $B_1 + B_2 + ... + B_n = \Omega$ và $B_iB_j = \varnothing \ \forall i \neq j$, khi đó với biến cố *A* liên quan được tính theo công thức:

$$P(A) = \sum_{i=1}^{n} P(A \mid B_i)P(B_i). \quad (4.3)$$

b) Định lý Bayes

Cho *h* là một giả thiết và *x* là tập các giá trị quan sát được. Khi đó, xác suất để giả thiết *h* là đúng khi biết *x* được tính như sau:

$$P(h \mid X) = \frac{P(X \mid h)P(h)}{P(X)} \quad (4.4)$$

trong đó:

***P(h)*:** xác suất tiên nghiệm của giả thiết *h*. Đây là xác suất để giả thiết *h* là đúng mà không liên quan gì tới *X*. Nó được gọi là "*tiên nghiệm*" với hàm ý rằng nó không quan tâm tới bất kỳ thông tin nào của *X*.

P(X): xác suất tiên nghiệm của việc quan sát được *X*. Đây là xác suất xảy ra *X* mà không quan tâm tới *h*.

***P(X|h)*:** xác suất xảy ra *X*, nếu biết giả thiết *h* là đúng. Nói cách khác, đây là xác suất xảy ra *X* khi biết giả thiết *h* đã xảy ra.

c) Phân lớp bằng kỹ thuật Naïve Bayes

Trước tiên, ta xét bài toán phân lớp. Cho một tập dữ liệu huấn luyện $X \in R^{n\times(m+1)}$ gồm *n* mẫu dữ liệu, mỗi mẫu có *m* thuộc tính và một thuộc tính lớp. Mỗi mẫu huấn luyện $x \in X$ được biểu diễn là một vectơ *m*+1 chiều $x(x_1, x_2, ..., x_m, y)$ trong đó gồm *m* thành phần dữ liệu và *y* là nhãn lớp. Cho một tập xác định các nhãn lớp $C = \{c_1, c_2,..., c_q\}$ gồm *q* lớp. Dễ thấy $y \in C$. Cho một mẫu dữ liệu mới $z \in R^m$ và *z* được biểu diễn bằng: $z(z_1, z_2,...,z_m)$. Hãy xác định lớp của *z*.

Để xác định lớp của *z*, một cách đơn giản là ta tính xác suất xảy ra khả năng *z* được phân vào từng lớp c_i, i=1..q, tức khả năng xảy ra c_i. Mẫu *z* sẽ được phân vào lớp nào có xác suất xảy ra cao nhất.

Tuy nhiên, mẫu *z* là xác định với các thành phần quan sát được là z_1, z_2,...z_m. Do đó, xác suất để *z* thuộc vào lớp c_i phải là xác suất có điều kiện $P(c_i \mid z_1, z_2,...,z_m)$ và được ký hiệu là $P(c_i|z)$. Theo định lý Bayes, xác suất này được tính như sau:

$$P(c_i \mid z) = \frac{P(z \mid c_i)\,P(c_i)}{P(z)}. \quad (4.5)$$

Trong phương pháp *Naïve Bayes*, từ *Naïve* có hàm ý giả sử rằng các thuộc tính là độc lập có điều kiện đối với các thuộc tính khác. Do đó

$$P(z \mid c_i) = \prod_{j=1}^{m} P(z_j \mid c_i), \quad (4.6)$$

và (4.5) trở thành:

$$P(c_i \mid z) = \frac{\prod_{j=1}^{m} P(z_j \mid c_i)\,P(c_i)}{P(z)}. \quad (4.7)$$

Mẫu dữ liệu *z* sẽ được phân vào lớp c_k nếu $P(c_k|z)$ là lớn nhất tức:

$$c_k = \arg\max_{c_i \in C} p(c_i \mid z) = \arg\max_{c_i \in C} \frac{\prod_{j=1}^{m} P(z_j \mid c_i).P(c_i)}{P(z)}. \quad (4.8)$$

Vì $P(z)$ là hằng số đối với các c_i khác nhau, do vậy (3.8) tương đương với:

$$c_k = \arg\max_{c_i \in C} \prod_{j=1}^{m} P(z_j \mid c_i).P(c_i). \quad (4.9)$$

Một cách đơn giản hơn, để xác định lớp cho mẫu dữ liệu *z*, ta lần lượt tính các giá trị của biểu thức $\prod_{j=1}^{m} P(z_j \mid c_i).P(c_i)$ với từng lớp $c_i \in \{c1, c_2,...,c_q\}$. Lớp c_i nào cho giá trị của biểu thức lớn nhất sẽ là lớp của *z*. Quá trình phân lớp sử dụng phương pháp *Naïve Bayes* bao gồm hai bước:

Bước 1: Đối với mỗi lớp $c_i \in C$, tính giá trị của:

- Xác suất tiên nghiệm $P(c_i)$. Xác suất này được tính xấp xỉ bằng tổng số mẫu thuộc lớp c_i trên tổng số mẫu của bộ dữ liệu huấn luyện.

- Đối với mỗi giá trị thuộc tính z_j, tính $P(z_j|c_i)$ là xác suất xảy ra của giá trị đó trong lớp c_i. Giá trị này cũng được tính xấp xỉ bằng tỷ lệ các mẫu có giá trị trên thuộc tính thứ j là z_j trong số các mẫu thuộc lớp c_i.

Bước 2: Cần xác định lớp cho một mẫu dữ liệu mới z, ta thực hiện:

- Đối với mỗi lớp $c_i \in C$, tính giá trị của biểu thức:

$$\prod_{j=1}^{m} P(z_j \mid c_i) p(c_i), \tag{4.10}$$

- Xác định lớp của z là c_k:

$$c_k = \underset{c_i \in C}{\operatorname{argmax}} \prod_{j=1}^{m} P(z_j \mid c_i) P(c_i). \tag{4.11}$$

Ví dụ 4.2. Cho bảng dữ liệu huấn luyện gồm 14 mẫu về quyết định (có hay không) mua máy tính như trong bảng, dựa vào các quan sát về tuổi (*Age*), thu nhập (*Income*), có là sinh viên hay không (*Student*) và tình hình tín dụng (*Credit*).

ID	**Age**	**Income**	**Student**	**Credit**	**Buy**
1	Young	High	No	Fair	no
2	Young	High	No	Excellent	no
3	Medium	High	No	Fair	yes
4	Old	Medium	No	Fair	yes
5	Old	Low	Yes	Fair	yes
6	Old	Low	Yes	Excellent	no
7	Medium	Low	Yes	Excellent	yes
8	Young	Medium	No	Fair	yes
9	Young	Low	Yes	Fair	yes
10	Old	Medium	Yes	Fair	yes
11	Young	Medium	Yes	Excellent	yes
12	Medium	Medium	No	Excellent	yes
13	Medium	High	Yes	Fair	yes

14	Old	Medium	No	Excellent	no

Cho một mẫu dữ liệu cần phân lớp *x*(*Youth, Medium, Yes, Fair*), tức xác định xem một sinh viên trẻ với thu nhập trung bình và mức đánh giá tín dụng bình thường sẽ có quyết định mua một chiếc máy tính hay không.

Dễ dàng thấy số mẫu dữ liệu n=14; số thuộc tính dữ liệu m=4 (do không xem xét thuộc tính *ID*); thuộc tính lớp là *Buy* với tập các lớp C={*yes, no*} gồm 2 lớp. Quá trình xác định lớp cho mẫu dữ liệu *x* trải qua hai bước:

Bước 1: với mỗi lớp $c_i \in C$:

- Xét c_1=*yes*: dễ dàng tình được *P(yes)* = 10/14. Ta tiếp tục tính $P(x_j|c_1)$:

P(Age=Young | Buy=yes) = 3/10;

P(Income=Medium | Buy=yes) = 5/10;

P(Student = Yes | Buy = yes) = 6/10;

P(Credit = Fair | Buy = yes) = 7/10.

- Xét c_2=*no*: dễ dàng tính được *P(no)* = 4/10. Ta tiếp tục tính $P(x_j|c_2)$:

P(Age=Young | Buy=no) = 2/4;

P(Income=Medium | Buy=no) = 1/4;

P(Student = Yes | Buy = no) = 1/4;

P(Credit = Fair | Buy = no) = 1/10.

Bước 2: Sử dụng các kết quả vừa tính, ta được:

- $\prod_{j=1}^{m} P(x_j \mid c_1)$ = *P(Age = Youth | Buy = Yes)* ×

P(Income = Medium | Buy = Yes) ×

P(Student = Yes | Buy = Yes) ×

P(Credit = Fair | Buy = Yes)

$= \frac{3}{10}\frac{5}{10}\frac{6}{10}\frac{7}{10} = 0{,}063.$

$\prod_{j=1}^{m} P(x_j \mid c_1).\mathrm{P}(c_1) = 0.063*10/14 =$ **0.045**.

- $\prod_{j=1}^{m} P(x_j \mid c_2)$ = *P(Age = Youth | Buy = no)* ×

P(Income = Medium | Buy = no) ×

P(Student = Yes | Buy = no) ×

P(Credit = Fair | Buy = no)

$$= \frac{2}{4}\frac{1}{4}\frac{1}{4}\frac{1}{4} = 0{,}0078.$$

$\prod_{j=1}^{m} P(x_j \mid c_2)\, P(c_2)$ = 0.0078*4/14 = **0.0022**.

Vậy mẫu dữ liệu *x* được phân vào lớp c_1 hay lớp của *x* là "*yes*".

Phương pháp Naïve Bayes trong trường hợp dữ liệu liên tục

Các thuộc tính trong Bảng 3.1 đều có giá trị rời rạc. Trong trường hợp thuộc tính có giá trị liên tục, ta có thể áp dụng các phương pháp rời rạc hóa. Nếu không rời rạc hóa dữ liệu, thay vì tính xác suất, ta sử dụng hàm mật độ xác suất. Thông thường, ta hay giả thiết là dữ liệu trong mỗi lớp c_i của các thuộc tính liên tục tuân theo phân bố *Gauss* và phương pháp lúc này được gọi là *Gauss Naïve Bayes*.

Xét thuộc tính *A* với các giá trị liên tục. Khi đó, ta phân đoạn các giá trị của *A* theo từng lớp. Với mỗi lớp c_i, ta tính μ_i là giá trị trung bình và σ_i^2 là phương sai của các giá trị của *A* trong lớp c_i (với N_i là số mẫu thuộc lớp c_i và y_i là lớp của mẫu x_i):

$$\mu_i = \frac{1}{N_i} \sum_{x_i : y_i = c_i} x_i,$$

$$\sigma_i^2 = \frac{1}{N_i - 1} \sum_{x_i : y_i = c_i} (x_i - \mu_i)^2. \qquad (4.12)$$

Giá trị $P(x|c_i)$ khi đó gọi là phân bố xác suất của *x* vào lớp c_i và được tính bằng:

$$p(x \mid c_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}. \qquad (4.13)$$

Ví dụ 4.3. Xét bảng dữ liệu sau, được xây dựng bằng cách thay thế cột *Income* (thu nhập) bằng các giá trị thực và dữ liệu được sắp lại trên cột *Buy* như sau:

ID	Age	Income	Student	Credit	Buy
1	Young	3.1	No	Fair	no

2	Young	2.8	No	Excellent	no
3	Old	3	Yes	Excellent	no
4	Old	3.7	No	Excellent	no
5	Medium	5.9	No	Fair	yes
6	Old	6	No	Fair	yes
7	Old	6.1	Yes	Fair	yes
8	Medium	3.1	Yes	Excellent	yes
9	Young	7	No	Fair	yes
10	Young	2.5	Yes	Fair	yes
11	Old	3.1	Yes	Fair	yes
12	Young	3.9	Yes	Excellent	yes
13	Medium	7.5	No	Excellent	yes
14	Medium	4.6	Yes	Fair	yes

Giả sử mẫu dữ liệu *x(Youth, 5.2, Yes, Fair)* cần phân lớp. Các giá trị *P(Income=5.2 | Buy=yes)* và *P(Income = 5.2 | Buy = no)* cần phải tính lại.

Xét lớp c_1 = *yes*, ta dễ dàng tính được μ_1 = 4.97 là giá trị trung bình trên cột *Income* của các mẫu thuộc lớp *yes* và σ_1^2 = 3.12 là phương sai tương ứng. Tương tự với lớp c_2=*no*, ta tính được μ_2 = 3.15 và σ_2^2 = 0.15. Ta có bảng các giá trị trung bình và phương sai trên từng lớp của cột *Income* như sau:

Giá trị / Lớp	μ	σ^2
yes	4.97	3.12
no	3.15	0.15

Vậy *P(Income=5.2 | Buy=yes)* = $\frac{1}{\sqrt{2*\pi*3.12}} e^{-\frac{(5.2-4.97)^2}{2*3.12}} \approx 0.191$ và

P(Income = 5.2 | Buy = no) = $\frac{1}{\sqrt{2*\pi*0.15}} e^{-\frac{(5.2-3.15)^2}{2*0.15}} \approx 1.7966.10^{-12}$.

Và do đó, $P(x|c_1) = \frac{3}{10} 0.191 \frac{6}{10} \frac{7}{10} \approx 0.0241$ và

$P(x|c_2) = \frac{2}{4} 1.7966 * 10^{-12} \frac{1}{4} \frac{1}{4} \approx 5.6 * 10^{-14}$.

Mẫu dữ liệu *x* được phân vào lớp *yes*.

Trường hợp xuất hiện xác suất bằng không

Xét một mẫu dữ liệu cần phân lớp $x(x_1, x_2,...,x_m)$. Xét giá trị x_j trên thuộc tính *j*. Nếu không có mẫu dữ liệu nào trong lớp c_i có giá trị trên thuộc tính *j* là x_j thì hiển nhiên $P(x_j|c_i) = 0$. Điều này kéo theo $P(c_i).\prod_{j=1}^{m} P(x_j \mid c_i) = 0$.

Giải pháp đưa ra là sử dụng ước lượng *Laplace* để ước lượng $P(x_j|c_i)$ thay cho giá trị 0 đã tính được ở trên.

Giả sử rằng ta có bộ dữ liệu với các thuộc tính giống như Bảng 3.1 và trong lớp c_1="*yes*" có 1000 mẫu dữ liệu. Xét thuộc tính *Income* của 1000 mẫu trên với 0 mẫu có giá trị *Income="Low";* 990 mẫu dữ liệu có *Income="Medium"* và 10 mẫu có *Income="High"*. Khi đó, các xác suất *P(Income="Low" | Buy=yes), P(Income="Medium" | Buy=yes)* và *P(Income="High" | Buy=yes)* lần lượt được xấp xỉ bằng 0, 990/1000 và 10/1000. Do đó, với một mẫu *x* cần phân lớp có *Income="Low",* ví dụ *x(Youth, Low, Yes, Fair)* ta tính được

$$\prod_{j=1}^{m} P(x_j \mid c_1) = P(Income="Low" \mid Buy=yes) \times P(Income="Medium" \mid Buy=yes) \times$$

$$P(Income="High" \mid Buy=yes) \approx \frac{0}{1000} \frac{990}{1000} \frac{10}{1000} = 0.$$

Để tránh trường hợp này, ta giả sử rằng số mẫu dữ liệu trong lớp "*yes*" là lớn và do đó, nếu ta bổ sung 01 mẫu dữ liệu cho mỗi tập có *Income="Low"*, *Income="Medium"* và *Income=" High"* thì việc này không ảnh hưởng nhiều tới các xác suất đã tính. Nhưng khi đó, các xác suất *P(Income="Low" | Buy=yes), P(Income="Medium" | Buy=yes)* và *P(Income="High" | Buy=yes)* sẽ thay đổi và lần lượt bằng 1/1003, 991/1003 và 11/1003 và

$$\prod_{j=1}^{m} P(x_j \mid c_1) \approx \frac{1}{1003} \frac{991}{1003} \frac{11}{1003} \approx 0.000011.$$