CHƯƠNG 1: TỔNG QUAN

1. Mục tiêu nghiên cứu

Trong báo cáo này, mục tiêu chính là đưa ra được một hệ thống khuyến nghị amine phù hợp nhất với nhu cầu của người dùng. Hệ thống có thể đưa vào ứng dụng được, nhằm mục tiêu gia tăng xác suất chọn được những bộ anime mà khách hàng yêu thích, làm tăng khả năng hứng thú cũng như trải nghiệm dịch vụ một cách tốt nhất khi sử dụng hệ thống.

Để làm được điều đó, trước hết chúng ta cần tìm hiểu các phương pháp khai phá dữ liệu và các kỹ thuật liên quan đến khuyến nghị sản phẩm. Sau đó tiến hành xây dựng được một hệ thống mô hình phục vụ cho việc dự đoán xu thế tìm kiếm anime của khách hàng, đâu là những bộ anime được người dùng yêu thích và đánh giá cao.

1. Phương pháp nghiên cứu

Phương pháp thu thập dữ liệu: các dữ liệu về kho phim, nội dung, đánh giá của khán giả đối với các bộ phim anime. Ngoài ra, kết hợp thu thập thông tin phân tích hành vi người tiêu dùng làm cơ sở đưa ra các khuyến nghị phù hợp thông qua phương pháp khảo sát.

Phương pháp phân tích thiết kế: thiết kế mô hình khai phá luật kết hợp, mạng Nerual, cây quyết định, mạng Bayes.

Sử dụng ngôn ngữ lập trình Python để xây dựng giải thuật.

Phương pháp thực nghiệm: dựa trên nguồn dữ liệu thu thập và tổng hợp được, tiến hành chuẩn hóa dữ liệu và ứng dụng xây dựng một hệ thống khuyến nghị các bộ phim anime khán giả nên xem.

Nguồn dữ liệu dự kiến cho đề tài là những file data sẵn có dành cho nghiên cứu khoa học được trích xuất từ Datasets-for-Recommender-Systems. Công cụ dùng để xử lý và phân tích dữ liệu là những phần mềm có thể xây dựng giải thuật và kiểm chứng kết quả.

1. Đối tượng nghiên cứu và phạm vi nghiên cứu

Hướng đến các bạn giới trẻ có sở thích xem những bộ phim hoạt hình thể loại anime trên địa bàn Thành phố Hồ Chí Minh

1. Kết cấu đề tài

Chương 1: Tổng quan.

1.1. Mục tiêu nghiên cứu.

1.2. Phương pháp nghiên cứu.

1.3. Đối tượng và phạm vi nghiên cứu.

1.4. Kết cấu đề tài.

Chương 2: Cơ sở lý thuyết.

2.1. Tổng quan về khai phá dữ liệu.

2.2. Các thuật toán máy học.

2.3. Hệ thống khuyến nghị.

Chương 3: Xây dựng hệ thống khuyến nghị anime.

3.1. Ứng dụng, triển khai hệ thống khuyến nghị anime.

3.2. Bài toán.

3.3. Xây dựng hệ thống.

Chương 4: Kết luận.

4.1. Kết quả đạt được.

4.2. Hướng phát triển.

Tài liệu tham khảo.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

1. Tổng quan về khai phá dữ liệu (data mining)
   1. Tại sao cần phải khai phá dữ liệu

Trong thời đại kỷ thuật số, công nghệ thông tin không ngừng phát triển và có những tiến bộ vượt bậc. Đa số các lĩnh vực đều cần đến sự hỗ trợ của công nghệ thông tin, công nghệ giúp nắm bắt thông tin một cách nhanh chóng, thay vì xử lý lưu trữ theo cách truyền thống bằng giấy, với sự trợ giúp của công nghệ thông tin có thể tiết kiệm thời gian, chi phí in ấn bằng những hóa đơn điện tử hay lưu trữ đám mây, chúng ta có thể lưu trữ trong một khoảng thời gian lâu, dễ dàng sửa đổi cập nhật thông tin cũng như tìm kiếm thông tin một cách nhanh chóng, tiện lợi.

Tuy nhiên, chúng ta cũng phải đối mặt với việc phải xử lý một số lượng rất lớn thông tin. Các thông tin thay đổi nhanh chóng từ các tính năng vượt trội mà công nghệ thông tin mang lại: tìm kiếm, cập nhật, sửa đổi, lưu trữ lâu dài,...khiến số lượng thông tin số hóa tăng dần theo cấp số nhân. Do vậy, việc khai phá tri thức từ dữ liệu để đáp ứng nhu cầu nắm bắt thông tin có vai trò vô cùng quan trọng. Các phương pháp khai thác và phân tích dữ liệu truyền thống không đáp ứng được sự bùng nổ của thông tin. Việc khai phá tri thức từ dữ liệu là một trong những vấn đề đã và đang nhận được nhiều sự quan tâm của các nhà nghiên cứu. Lĩnh vực phát hiện tri thức từ các cơ sở dữ liệu đã được hình thành, trong đó khai phá dữ liệu (data mining) được xem là trung tâm của lĩnh vực nghiên cứu và ứng dụng này.

* 1. Định nghĩa khai phá dữ liệu

Khai phá dữ liệu (data mining) là quá trình tìm kiếm và phân tích một lượng lớn dữ liệu thô để thu thập các mẫu và trích xuất thông tin hữu ích. Thuật ngữ Dataming ám chỉ việc tìm kiếm một tập hợp nhỏ có giá trị từ một số lượng lớn các dữ liệu thô. Ngoài bước phân tích thô, nó còn liên quan tới cơ sở dữ liệu và các khía cạnh quản lý dữ liệu, xử lý dữ liệu trước, suy xét mô hình và suy luận thống kê, các cân nhắc phức tạp, xuất kết quả về các cấu trúc được phát hiện, hiện hình hóa và cập nhật trực tuyến.

Khai phá dữ liệu là một bước trong bảy bước của quá trình KDD (Knowleadge Discovery in Database) và KDD được xem như 7 quá trình khác nhau theo thứ tự sau:

1. Làm sạch dữ liệu (data cleaning & preprocessing)s: Loại bỏ nhiễu và các dữ liệu không cần thiết.

2. Tích hợp dữ liệu: (data integration): quá trình hợp nhất dữ liệu thành những kho dữ liệu (data warehouses & data marts) sau khi đã làm sạch và tiền xử lý (data cleaning & preprocessing).

3. Trích chọn dữ liệu (data selection): trích chọn dữ liệu từ những kho dữ liệu và sau đó chuyển đổi về dạng thích hợp cho quá trình khai thác tri thức. Quá trình này bao gồm cả việc xử lý với dữ liệu nhiễu (noisy data), dữ liệu không đầy đủ (incomplete data), .v.v.

4. Chuyển đổi dữ liệu: Các dữ liệu được chuyển đổi sang các dạng phù hợp cho quá trình xử lý

5. Khai phá dữ liệu(data mining): Là một trong các bước quan trọng nhất, trong đó sử dụng những phương pháp thông minh để chắt lọc ra những mẫu dữ liệu.

6. Ước lượng mẫu (knowledge evaluation): Quá trình đánh giá các kết quả tìm được thông qua các độ đo nào đó.

7. Biểu diễn tri thức (knowledge presentation): Quá trình này sử dụng các kỹ thuật để biểu diễn và thể hiện trực quan cho người dùng.

* 1. Các chức năng chính của khai phá dữ liệu

Data Mining được chia nhỏ thành một số hướng chính như sau:

* Mô tả khái niệm (concept description): thiên về mô tả, tổng hợp và tóm tắt khái niệm. Ví dụ: tóm tắt văn bản.
* Luật kết hợp (association rules): là dạng luật biểu diễn tri thứ ở dạng khá đơn giản. Ví dụ: “60 % nam giới vào siêu thị nếu mua bia thì có tới 80% trong số họ sẽ mua thêm thịt bò khô”. Luật kết hợp được ứng dụng nhiều trong lĩnh vực kính doanh, y học, tin-sinh, tài chính & thị trường chứng khoán, .v.v.
* Phân lớp và dự đoán (classification & prediction): xếp một đối tượng vào một trong những lớp đã biết trước. Ví dụ: phân lớp vùng địa lý theo dữ liệu thời tiết. Hướng tiếp cận này thường sử dụng một số kỹ thuật của machine learning như cây quyết định (decision tree), mạng nơ ron nhân tạo (neural network), .v.v. Người ta còn gọi phân lớp là học có giám sát (học có thầy).
* Phân cụm (clustering): xếp các đối tượng theo từng cụm (số lượng cũng như tên của cụm chưa được biết trước. Người ta còn gọi phân cụm là học không giám sát (học không thầy).
* Khai phá chuỗi (sequential/temporal patterns): tương tự như khai phá luật kết hợp nhưng có thêm tính thứ tự và tính thời gian. Hướng tiếp cận này được ứng dụng nhiều trong lĩnh vực tài chính và thị trường chứng khoán vì nó có tính dự báo cáo.
  1. Ứng dụng của khai phá dữ liệu

Các lĩnh vực có ứng dụng khai thác dữ liệu bao gồm:

* Phân tích dữ liệu và hỗ trợ ra quyết định (data analysis & decision support)
* Bảo hiểm (insurance)
* Thương mại điện tử (e-commerce)
* Tài chính và thị trường chứng khoán (finance & stock market)
* Điều trị y học (medical treatment)
* Text mining & Web mining
* Tin-sinh (bio-informatics)
* Marketing
* .v.v.

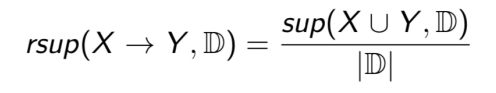
1. Các thuật toán học máy
   1. Cây quyết định
   2. Luật kết hợp

Khai phá luật kết hợp - Accessociation rule mining là một kỹ thuật quan trọng của khai phá dữ liệu lần đầu tiên được Rakesh Agrawal, Tomas Imielinski, Arun Swami đề xuất năm 1993. Một luật kết hợp là một phương pháp dựa trên quy tắc để tìm mối quan hệ, tần suất xuất hiện giữa các biến trong một bộ dữ liệu nhất định. Các phương pháp này cho phép các doanh nghiệp hiểu rõ hơn về mối quan hệ giữa các sản phẩm khác nhau. Hiểu thói quen tiêu dùng của khách hàng cho phép các doanh nghiệp phát triển các chiến lược và động cơ khuyến nghị bán chéo tốt hơn.

Khai phá luật kết hợp tìm ra mối liên hệ, sự tương quan, giữa các sự kiện, những sự kiện xuất hiện thường xuyên, cùng nhau một cách đồng thời. Nhiệm vụ chính của khai phá luật kết hợp là phát hiện ra các tập con cùng xuất hiện trong một khối lượng giao dịch lớn của một cơ sở dữ liệu cho trước. Từ đó tìm ra các quy luật, các chiến lược kinh doanh hiệu quả và thông minh từ kho dữ liệu.

Luật kết hợp là mối quan hệ giữa các tập thuộc tính trong cơ sở dữ liệu. Luật kết hợp là phương tiện hữu ích để khám phá các mối liên kết trong dữ liệu. Một luật kết hợp là một mệnh đề kéo theo có dạng X -> Y, trong đó X, Y ⊆ I, thỏa mãn điều kiện X giao Y = rỗng. Các tập hợp X và Y được gọi là các tập hợp thuộc tính (itemset). Tập X gọi là nguyên nhân, tập Y gọi là hệ quả. Có 2 độ đo quan trọng đối với luật kết hợp: Độ hỗ trợ (support) và độ tin cậy (confidence), được định nghĩa như phần dưới đây.

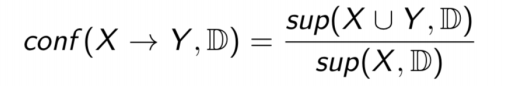
* Độ hỗ trợ
  + Độ hỗ trợ của một luật kết hợp X -> Y là tỷ lệ giữa số lượng các bản ghi chứa tập hợp X -> Y, so với tổng số các bản ghi trong D - Ký hiệu supp (X -> Y).
  + Độ hỗ trợ tương đối của luật X->Y trong cở sở dữ liệu D kí hiệu là rsup(X->Y, D) là số phần trăm các giao dịch trong D chứa cả X và Y.



Nếu độ hỗ trợ của một kết kết hợp X -> Y là 30% thì có nghĩa là 30% tổng số bản ghi chứa X hợp Y. Như vậy độ hỗ trợ mang ý nghĩa thống kê của luật.

* Độ tin cậy

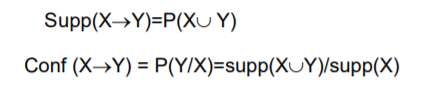
Độ tin cậy (confidence) của luật X → Y trong D, ký hiệu conf (X → Y , D), là tỉ lệ giữa số giao dịch chứa cả X và Y trên số giao dịch chỉ chứa X.



Ký hiệu độ tin cậy của một luật r là conf(r). Ta có 0 <= conf(r) <= 1

Độ hỗ trợ và độ tin cậy có xác xuất như sau:

* + Độ hỗ trợ là xác xuất trong giao dịch chứa cả X và Y.
  + Độ tin cậy là xác xuất có điều kiện mà một giao dịch trong D chứa Y trong khi đã chứa X (bản chất vẫn là mức độ in cậy của luật).



**Kết luận**

* Luật X -> Y được gọi là phổ biến nếu sup(X->Y, D) >= minsup (minsup do người dùng định nghĩa)
* Luật X-> Y được gọi là mạnh nếu độ tin cậy của nó lớn hơn hoặc bằng một ngưỡng minconf do người dùng định nghĩa: conf (X->Y) >= minconf

VD: Cơ sở giao dịch D = {T1, T2, T3, T4, T5, T6} trong đó:

* T1 = {A, B, D, E}
* T2 = {B, C, E}
* T3 = {A, B, D, E}
* T4 = {A, B, C, E}
* T5 = {A, B, C, D, E}
* T6 = {B, C, D}

Xét luật {B, C} -> {E} hay BC->E

* sup(BC->E, D) = sup(BCE, D) = 3 (số lần xuất hiện của bộ ba BCE trong các giao dịch thuộc D)
* conf(BC->E, D) = sup(BCE, D)/sup(BC, D) = 3/4 (=75%)

Xét luật {A, D} -> {B, E} hay AB->DE

* Độ hỗ trợ sup(AD->BE, D) = sup(ABDE, D) = 3
* Độ tin cậy conf(AD->BE, D) = sup(ABDE, D)/sup(AD, D) = 3/3 (=100%)

Tức ta có thể đưa ra kết luận là nếu trong giao dịch có chứa AD thì chắc chắn sẽ chứa BE

**Tính chất**

* Tính chất 1: Giả sử A,B ⊆ I là hai tập hợp với A⊆B thì sup(A) >= sup(B). Như vậy, những bản ghi nào chứa tập hợp B thì cũng chứa tập hợp A
* Tính chất 2: Giả sử A, B là hai tập hợp, A,B ⊆ I, nếu B là tập phổ biến và A⊆B thì A cũng là tập phổ biến. Vì nếu B là tập phổ biến thì sup(B) >= minsup, mọi tập hợp A là con của tập hợp B đều là tập phổ biến trong cơ sở dữ liệu D vì sup(A) >= sup(B) (Tính chất 1)
* Tính chất 3: Giả sử A, B là hai tập hợp, A ⊆ B và A là tập hợp không thường xuyên thì B cũng là tập hợp không thường xuyên (Tính chất 1) (Tức nếu A là tập hợp không phổ biến thì mọi tập cha của nó cũng không biến)

**Giới thiệu về luật kết hợp trong khai phá dữ liệu:**

Trong lĩnh vực Data Mining, mục đích của luật kết hợp (Association Rule - AR) là tìm ra các mối quan hệ giữa các đối tượng trong khối lượng lớn dữ liệu. Nội dung cơ bản của luật kết hợp được tóm tắt như dưới đây ([[1]](#footnote-1))

Cho cơ sở dữliệu giao dịchTgồmtập các giao dịch t1,t2,…,tn.

T = {t1, t2,…, tn}. Mỗi giao dịch ti bao gồm tập các đối tượng

I (gọi là itemset). I = {i1, i2,…, im}. Một itemset gồmk items, gọi là k-itemset.

Mục đích của luật kết hợp là tìm ra sự kết hợp (tương quan) giữa các items.

Những luật kết hợp này có dạng: X →Y

Hai tiêu chí rất quan trọng trong việc đánh giá luật kết hợp đó là độ hỗ trợ (support) và độ tin cậy (confidence). Công thức tính độ hỗ trợ và độ tin cậy của luật kết hợp X→Y [2]: ( → ) = ( ∪ ) = ( ∪ ) ( → ) = (|) = ( ∪ ) () Trong đó: (): Số giao dịch chứa X; N: Tổng số giao dịch; Các luật kết hợp có độ hỗ trợ và độ tin cậy lớn hơn hoặc bằng độ hỗ trợ tối thiểu (min\_sup) và độ tin cậy tối thiểu (min\_conf) gọi là các luật mạnh.min\_sup và min\_conf gọi là các giá trị ngưỡng (threshold), được xác định trước khi sinh các luật kết hợp ([[2]](#footnote-2)) , ([[3]](#footnote-3))

**Các phương pháp khai phá:**

Khai phá luật kết hợp là việc phát hiện ra các luật kết hợp thỏa mãn các ngưỡng độ hỗ trợ (minsup) và ngưỡng độ tin cậy (minconf) cho trước. Bài toán khai phá luật kết hợp gồm hai pha:

* Pha 1: Khai phá tất cả các tập phổ biến (FI) trong CSDL D với ngưỡng độ hộ trợ tối thiểu minsup (thường có độ tính toán cao và chiếm phần lớn thời gian trong khai phá luật kết hợp)
* Pha 2: Sinh ra tất cả các luật mạnh từ các tập phổ biến khai phá được từ pha trước với ngưỡng tiến cậy tối thiểu mà minconf.

**Thuật toán 1 – Thuật toán cơ bản:**

Đầu vào: I, D, minsup, minconf

Đầu ra: Các luật kết hợp thỏa mãn ngưỡng độ hỗ trợ minsup , ngưỡng độ tin cậy minconf. Thuật toán:

1. Tìm tất cả các tập hợp các tính chất có độ hỗ trợ >= minsup.
2. Từ các tập hợp mới tìm ra, tạo ra các luật kết hợp có độ tin cậy >= minconf.

**Thuật toán 2- Tìm luật kết hợp khi đã biết các tập hợp thường xuyên:**

Đầu vào: I, D, minsup, minconf, tập phổ biến S

Đầu ra: Các luật kết hợp thỏa mãn ngưỡng độ hỗ trợ minsup, ngưỡng độ tin cậy minconf.

Thuật toán:

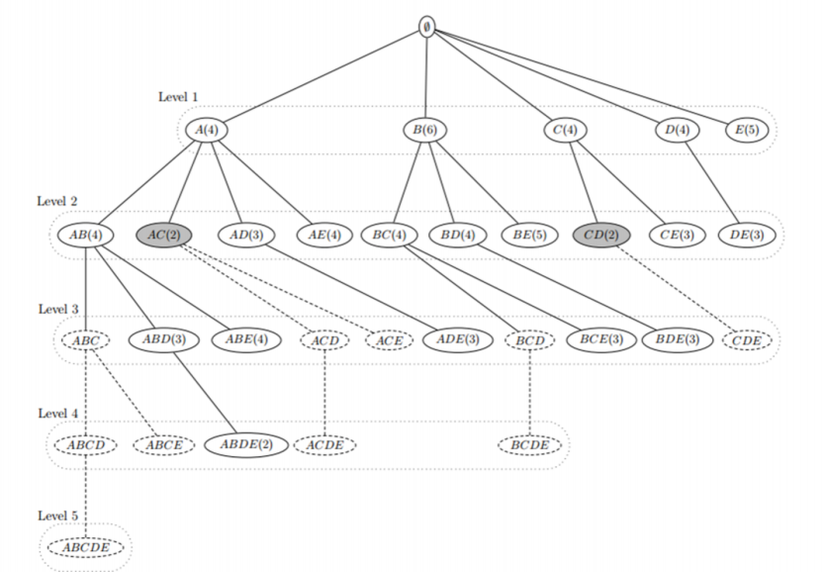
1. Lấy ra một tập phổ biến s⊆S, và một tập con X ⊆ s.
2. Xét luật kết hợp có dạng X → (sX), đánh giá độ tin cậy của nó xem có nhỏ hơn minconf hay không. Thực chất, tập hợp S mà ta xét đóng vai trò của tập hợp giao S = X hợp Y, và do X giao (S – X) = rỗng, nên coi như Y = S – X.

Các thuật toán xoay quanh khai phá luật kết hợp chủ yếu nêu ra các giải pháp để đẩy nhanh việc thực hiện Pha 1 là tìm tất cả các tập phổ biến.

**Thuật toán Apiori**

Thuật toán dựa trên hai tính chất của tập phổ biến là:

* Nếu tập A phổ biến thì mọi tập con B của A đều phổ biến
* Nếu tập con A không phổ biến thì mọi tập cha của A là B đều không phổ biến



Các nút màu xam là các tập không phổ biến tức có sup < minsup

VD: Tập AC với sup(AC, D) = 2 (<minsup) thì các tập cha của AC là ACD, ACE, ACDE chắc chắn không phải là tập phổ biến -> bị loại ra khỏi cây và không cần tính độ hỗ trợ của chúng -> cây được cắt tỉa.

VD: Cơ sở giao dịch D, minsup=3, D = {T1, T2, T3, T4, T5, T6} trong đó:

* T1 = {A, B, D, E}
* T2 = {B, C, E}
* T3 = {A, B, D, E}
* T4 = {A, B, C, E}
* T5 = {A, B, C, D, E}
* T6 = {B, C, D}

Tập ứng viên C1

|  |  |
| --- | --- |
| **1-itemset** | **Độ hỗ trợ** |
| A | 4 |
| B | 6 |
| C | 4 |
| D | 4 |
| E | 5 |

Tập phổ biến F1

|  |  |
| --- | --- |
| **1-itemset** | **Độ hỗ trợ** |
| A | 4 |
| B | 6 |
| C | 4 |
| D | 4 |
| E | 5 |

Tập ứng viên C2 sinh từ tập ứng viên F1

|  |  |
| --- | --- |
| **2-itemset** | **Độ hỗ trợ** |
| AB | 4 |
| AC | 2 |
| AD | 3 |
| AE | 4 |
| BC | 4 |
| BD | 4 |
| BE | 5 |
| CD | 2 |
| CE | 3 |
| DE | 3 |

Tập phổ biến F2

|  |  |
| --- | --- |
| **2-itemset** | **Độ hỗ trợ** |
| AB | 4 |
| AD | 3 |
| AE | 4 |
| BC | 4 |
| BD | 4 |
| BE | 5 |
| CE | 3 |
| DE | 3 |

Tập ứng viên C3 sinh từ tập ứng viên F2: Không xét những tập chứa AC, CD

|  |  |
| --- | --- |
| **3-itemset** | **Độ hỗ trợ** |
| ABD | 3 |
| ABE | 4 |
| ADE | 3 |
| BCE | 3 |
| BDE | 3 |

Tập phổ biến F3

|  |  |
| --- | --- |
| **3-itemset** | **Độ hỗ trợ** |
| ABD | 3 |
| ABE | 4 |
| ADE | 3 |
| BCE | 3 |
| BDE | 3 |

Tập ứng viên C4 được sinh từ tập phổ biến F3

|  |  |
| --- | --- |
| **4-itemset** | **Độ hỗ trợ** |
| ABDE | 3 |

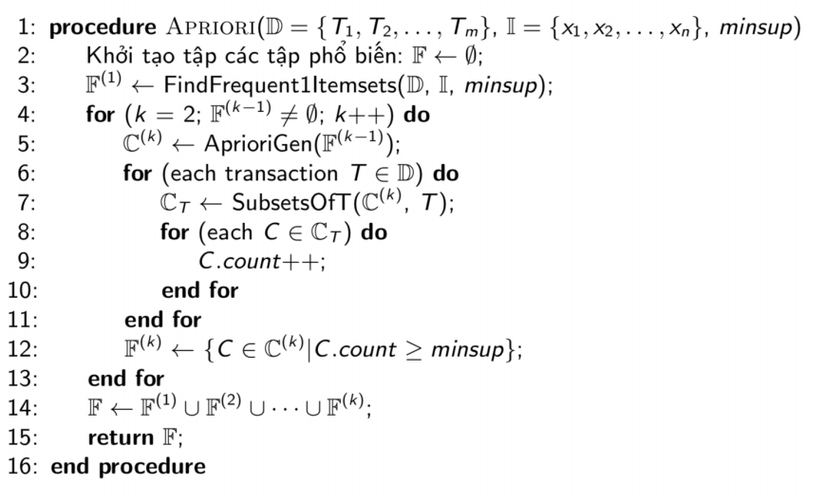
Tập phổ biến F4

|  |  |
| --- | --- |
| **4-itemset** | **Độ hỗ trợ** |
| ABDE | 3 |

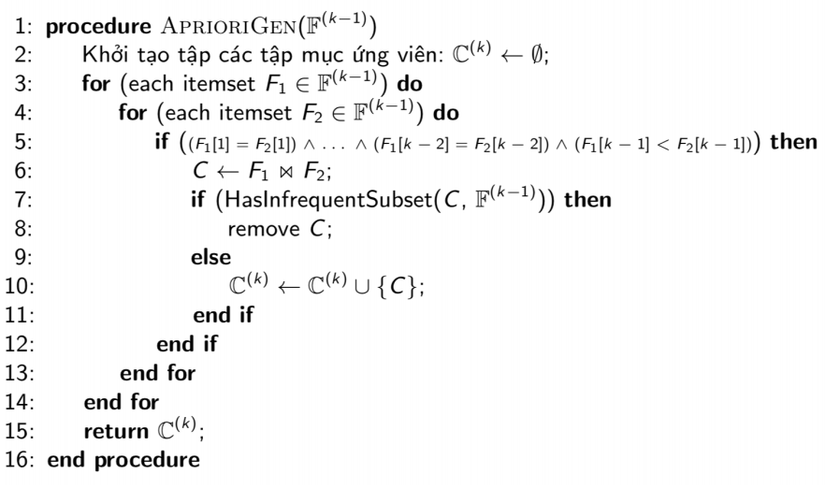
Thuật toán Apriori

Đầu vào: cơ sở giao dịch D và tập mục I, minsup (dòng 1)

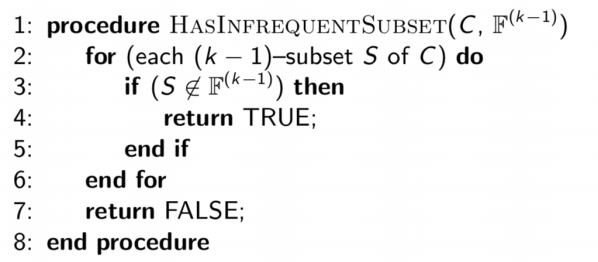
* Bước 1: Xét tập các tập phổ biến F = [] (dòng 2)
* Bước 2: Tìm tất cả tập phổ biến F1 (dòng 3)
* Bước 3: Xét tất cả các tập ứng viên Ck được sinh ra từ tập phổ biến Fk-1 (dòng 4)
* Bước 4: Sinh tập ứng viên Ck từ tập phổ biến Fk-1 (dòng 5)
* Bước 5: Tính sup của các tập ứng viên trong C (dòng 6->10)
* Bước 6: Nếu sup của ứng viên >= minsup thì ứng viên là một tập phổ biến được đưa vào tập phổ biến Fk (dòng 12)
* Thực hiện bước 4, 5, 6 cho đến khi không tìm được tập phổ biến
* Bước 7: Tập phổ biến bằng hợp của các tập phổ biến F1 -> Fk (dòng 15)



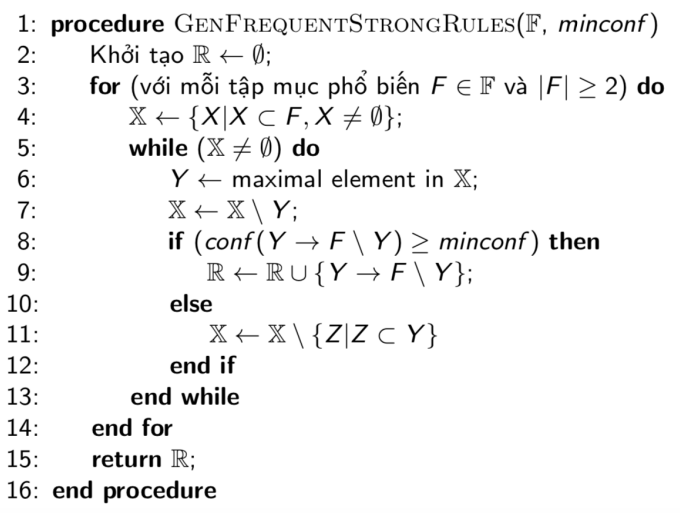
Hàm sinh các tập ứng viên từ tập phổ biến



Kiểm tra tập có chưa tập không phổ biến hay không



Sinh luật kết hợp phổ biến và mạnh từ tập phổ biến Đầu vào: Tập tất cả các tập phổ biến Đầu ra: Tập tất cả các luật phổ biến mạnh



* Bước 1: Khởi tạo tập luật mạnh R = [] (dòng 2)
* Bước 2: Xét từng tập phổ biến F (dòng 3)
* Bước 3: Tập X là tập các tập con của tập phổ biến F (X khác rỗng) (dòng 4)
* Bước 4: Trong khi X khác rỗng thực hiên vòng lặp (dòng 5)
* Bước 5: Y = phần tử lớn nhất trong X (dòng 6)
* Bước 6: Xét X = X \ Y (dòng 7)
* Bước 7: Nếu conf(Y ->F \ Y) >= minconf thì luật Y->F\Y là luật mạnh. Ngược lại xét X <- X {Z là tập con của Y} (dòng 8->12)
* Bước 8: Trả tập luật mạnh R (dòng 15)

**Ưu và nhược điểm của phương pháp Apriori**

**Ưu điểm**

* Dùng tính chất của tập phổ biến có thể cắt tỉa được nhiều nhánh, giảm bớt việc sinh các tập ứng viên từ những tập không hợp lệ (tập không phổ biến)

**Nhược điểm**

* Vẫn cần sinh một số lượng lớn các tập ứng viên. Cần quét cơ sở dữ liệu nhiều lần trong quá trình thực hiện thuật toán.
  1. Mạng Bayes

**Định nghĩa**

Một mạng Bayes là một đồ thị có hướng phi chu trình mà trong đó:

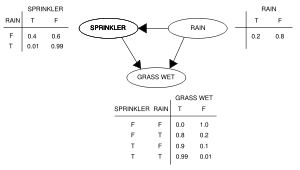
* các nút biểu diễn các biến
* các cung biểu diễn các quan hệ phụ thuộc thống kê giữa các biến và phân phối xác suất địa phương cho mỗi giá trị nếu cho trước giá trị của các cha của nó

Nếu có một cung từ nút A tới nút B, thì biến B phụ thuộc trực tiếp vào biến A, và A được gọi là cha của B. Nếu với mỗi biến Xi, i ∈{1,..,N} tập hợp các biến cha được ký hiệu bởi parents(Xi), thì phân phối có điều kiện phụ thuộc của các biến là tích của các phân phối địa phương

Description: https://voer.edu.vn/file/51959

Nếu Xi không có cha, ta nói rằng phân phối xác suất địa phương của nó là không có điều kiện, nếu không, nó là có điều kiện. Nếu biến được biểu diễn bởi một nút được quan sát, thì ta nói rằng nút đó là một nút hiển nhiên (evidence node).

Các câu hỏi về sự phụ thuộc không tương đẳng giữa các biến có thể được trả lời bằng cách nghiên cứu đồ thị. Có thể chứng minh rằng trong đồ thị, tính độc lập có điều kiện được biểu diễn bởi tính chất đồ thị d-separation: cho trước một số nút hiển nhiên cụ thể, các nút X và Y là d-separated trong đồ thị khi và chỉ khi các biến X và Y là độc lập, biết trước các biến hiển nhiên tương ứng. Tập hợp gồm tất cả các nút khác mà X có thể phụ thuộc trực tiếp được cho bởi Markov blanket của X.

Một ưu điểm của mạng Bayes là, về mặt trực quan, con người có thể hiểu các quan hệ phụ thuộc trực tiếp và các phân phối địa phương dễ dàng hơn là phân phối có điều kiện phụ thuộc hoàn chỉnh.

**Ví dụ**

*Một mạng Bayes đơn giản.*

Nếu có hai lý do cho việc cỏ bị ướt (GRASSWET): hoặc do được tưới nước (SPRINKLER), hoặc do trời mưa (RAIN), thì tình huống này có thể được mô hình hóa bởi một mạng Bayes. Ở đây, các biến có hai trạng thái có thể: T (đúng) và F (sai).

Hàm xác suất phụ thuộc có điều kiện là

Pr(GRASSWET,SPRINKLER,RAIN) = Pr(GRASSWET | SPRINKLER,RAIN).Pr(SPRINKLER | RAIN).Pr(RAIN)

Mô hình có thể trả lời các câu hỏi như "Nếu cỏ ướt thì khả năng trời mưa là bao nhiêu?" bằng cách sử dụng các công thức xác suất có điều kiện và lấy tổng tất cả các biến trở ngại (nuisance variable):

Description: https://voer.edu.vn/file/51960

Description: https://voer.edu.vn/file/51957

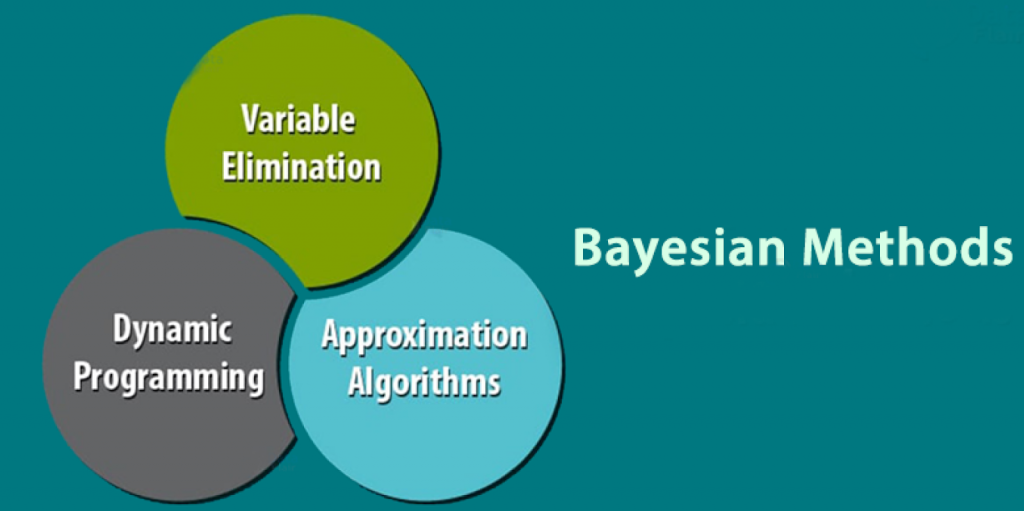
Thay thế các giá trị số, ta được Pr(RAIN=T | GRASSWET=T) = 891/2491 ≈ 35.77%.

Cách khác: (P(G=T,S=F,R=T) + P(G=T,S=T,R=T)) / (P(G=T,S=F,R=F) + P(G=T,S=T,R=F) + P(G=T,S=F,R=T) + P(G=T,S=T,R=T)) = (15.84%+0.198%) / (0.0%+28.8%+15.84%+0.198%) = 16.038% / 44.838% ≈ 35.77%.

**Các phương pháp Bayes**

Có ba phương pháp khác nhau trong mạng Bayes:

* Loại bỏ biến
* Lập trình năng động
* Các thuật toán gần đúng



**Loại bỏ biến**

Description: https://websitehcm.com/wp-content/uploads/2022/01/image-24.png

Để thực hiện việc định biên hiệu quả, bạn có thể sử dụng Phân phối xác suất chung. Trong phương pháp này, bạn có thể tổng hợp các thuật ngữ không liên quan.

Ví dụ:

Chúng tôi sẽ xóa các biến từ trong ra ngoài. Bắt đầu với biến Xm và kết thúc với biến X1. Ở đây chúng tôi thực hiện các phép tính tổng từ phải sang trái, lưu trữ các kết quả trung gian (hệ số) để tránh tính toán lại. Để loại bỏ biến Xi, chúng tôi bắt đầu bằng cách thu thập tất cả các yếu tố đề cập đến Xi và loại bỏ chúng khỏi tập hợp các yếu tố của chúng tôi. Trong quá trình tính toán, độ phức tạp của việc loại bỏ biến phụ thuộc vào kích thước của yếu tố lớn nhất. Điều này, đến lượt nó, phụ thuộc vào thứ tự loại bỏ các biến. Trong mạng cây đa, độ phức tạp không gian và thời gian là tuyến tính trong kích thước của mạng.

**Lập trình động**

Giả sử bạn muốn tính toán một số lợi nhuận. Nếu bạn sử dụng thuật toán loại bỏ biến, thì sẽ có một số phép tính dư thừa. Để tránh chúng, bạn có thể sử dụng phương pháp Lập trình động (DP) . Nó hoạt động theo kiểu mô hình đồ họa.

Khi đồ thị Mạng Bayes có dạng xoay vòng (nghĩa là dạng cây), thì bạn có thể sử dụng thuật toán truyền thông điệp cục bộ. Đây là một thuật toán chuyển tiếp-lùi đơn giản cho chuỗi HMM . Khi Mạng Bayes có các chu kỳ vô hướng, sẽ có nguy cơ tính hai lần bởi các thuật toán truyền thông điệp cục bộ.Để tránh điều này, bạn có thể chuyển đổi Mạng Bayes không định hướng thành một cây bằng cách nhóm các nút lại với nhau. Do đó, cây kết quả được gọi là cây nối.

Nhược điểm lớn của thuật toán Lập trình động là mất nhiều thời gian hơn để chạy ở kích thước cụm lớn. Cảm ứng cho độ rộng của kích thước đồ thị đề cập đến sự gia tăng thời gian xử lý. Giảm thiểu chiều rộng của đồ thị là một bài toán khó NP . NP là một loại các bài toán tính toán. Bạn có thể giải quyết vấn đề này với sự trợ giúp của máy Turing không xác định.

*Tìm hiểu về các ứng dụng mạng Bayes trong thế giới thực hàng đầu*

**Các thuật toán xấp xỉ**

Các mô hình có cấu trúc lặp lại như mô hình chuỗi thời gian đa biến được sử dụng để phân tích hình ảnh và có độ rộng cảm ứng cao. Vì vậy, chúng sẽ mất rất nhiều thời gian nếu bạn cố gắng suy luận chúng bằng thuật toán loại bỏ biến hoặc lập trình động.

Đối với những mô hình này, bạn cũng có thể sử dụng kỹ thuật xấp xỉ. Hãy để chúng tôi xem một số kỹ thuật xấp xỉ chính

**Phương pháp biến đổi**

Đây là một trong những kỹ thuật trường trung bình đơn giản nhất được sử dụng để tính gần đúng của các thuật toán. Nó sử dụng quy tắc xấp xỉ tổng của các biến ngẫu nhiên có giá trị lớn bằng cách sử dụng phương tiện của chúng.

**Mạng Bayes nhân quả**

Một mạng Bayes nhân quả là một mạng Bayes mà trong đó các cung có hướng của đồ thị được hiểu là các quan hệ nhân quả trong một miền xác định có thực nào đó. Các cung có hướng không nhất thiết phải được hiểu là các quan hệ nhân quả; tuy nhiên, trong thực tiễn, tri thức về các quan hệ nhân quả rất hay được dùng để hướng dẫn vẽ các đồ thị mạng Bayes, kết quả là có được các mạng Bayes nhân quả.

**Học cấu trúc**

Trong trường hợp đơn giản nhất, một mạng Bayes được xây dựng bởi một chuyên gia và rồi được dùng để thực hiện việc suy luận. Trong các ứng dụng khác, công việc xây dựng mạng quá phức tạp đối với con người. Trong trường hợp này, cấu trúc và các tham số mạng của các phân bố địa phương phải được học từ dữ liệu.

Học cấu trúc của một mạng Bayes (nghĩa là học đồ thị) là một phần rất quan trọng của ngành học máy. Giả thiết rằng dữ liệu được sinh từ một mạng Bayes và rằng tất cả các biến là thấy được trong mọi lần lặp, việc tối ưu hóa dựa trên phương pháp tìm kiếm có thể được dùng để tìm cấu trúc mạng. Việc này đòi hỏi một hàm tính điểm (scoring function) và một chiến lược tìm kiếm. Một hàm tính điểm thông dụng là xác suất hậu nghiệm (posterior probability) của cấu trúc khi cho trước dữ liệu huấn luyện. Quá trình tìm kiếm duyệt toàn bộ để trả về một cấu trúc có số điểm tối ưu đòi hỏi thời gian cấp siêu lũy thừa (superexponential) theo số lượng biến. Một chiến lược tìm kiếm địa phương thực hiện các thay đổi tăng dần hướng tới việc nâng cao điểm số của cấu trúc. Một thuật toán tìm kiếm toàn cục như Phương pháp Monte Carlo xích Markov (Markov chain Monte Carlo) có thể tránh việc bị bẫy trong một cực tiểu địa phương.

**Học tham số**

Để cụ thể hóa mạng Bayes và biểu diễn đầy đủ các phân bố xác suất phụ thuộc có điều kiện, đối với mỗi biến X, cần phải chỉ ra phân bố xác suất X theo các cha của X. Phân bố của X theo các cha của nó có thể có hình thức bất kỳ. Người ta thường dùng các phân bố rời rạc hay phân bố Gauss, do các phân bố này làm đơn giản việc tính toán. Đôi khi, khi chỉ biết được các ràng buộc của các phân số; người ta có thể dùng nguyên lý entropy cực đại để xác định một phân bố đơn, phân bố với entropy cực đại thỏa mãn các ràng buộc đó. (Tương tự, trong ngữ cảnh cụ thẻ của một mạng Bayes động, người ta thường lấy phân bố có điều kiện cho sự phát triển theo thời gian của trạng thái ẩn để cực đại hóa hệ số entropy (entropy rate) của quá trình ngẫu nhiên được nói đến.)

Thông thường, các phân bố có điều kiện này bao gồm các tham số chưa biết và phải được ước lượng từ dữ liệu, đôi khi bằng cách tiếp cận khả năng cực đại (maximum likelihood). Việc cực đại hóa trực tiếp khả năng (hay xác suất hậu nghiệm) thường phức tạp khi có các biến không được quan sát. Một cách tiếp cận truyền thống đối với vấn đề này là thuật toán cực đại hóa kỳ vọng (expectation-maximization algorithm), thuật toán này luân phiên giữa việc tính toán các giá trị mong đợi của các biến không được quan sát theo dữ liệu quan sát được, với việc cực đại hóa khả năng (hay hậu nghiệm) hoàn chỉnh với giả thuyết rằng các giá trị mong đợi đã tính được là đúng đắn. Dưới các điều kiện chính quy và vừa phải, quá trình này hội tụ về các giá trị khả năng cực đại (hay hậu nghiệm cực đại) cho các tham số. Một cách tiếp cận Bayes đầy đủ hơn đối với việc học tham số là coi các tham số như là các biến không quan sát được khác và tính một phân bố hậu nghiệm đầy đủ trên toàn bộ các nút theo dữ liệu quan sát được, sau đó tách các tham số ra. Cách tiếp cận này có thể có chi phí cao và dẫn đến các mô hình có số chiều lớn, do đó trong thực tế, các cách tiếp cận truyền thống thường được sử dụng hơn.

**Suy luận**

Do mạng Bayes là một mô hình hoàn chỉnh cho các biến và các quan hệ giữa chúng, có thể dùng mạng Bayes để trả lời các truy vấn xác suất về các biến này. Ví dụ, mạng Bayes có thể được dùng để tìm tri thức mới nhất về trạng thái của một tập con gồm các biến khi các biến khác (các biến hiển nhiên) được quan sát. Quá trình tính phân bố hậu nghiệm này của các biến khi cho trước các biến hiển nhiên được gọi là suy luận xác suất. Quá trình hậu nghiệm cho ra một thống kê đủ phổ quát (universal sufficient statistic) cho các ứng dụng phát hiện, khi người ta muốn chọn các giá trị cho một tập con các biến nhằm mục đích cực tiểu hóa một hàm phí tổn nào đó, chẳng hạn xác suất của lỗi quyết định. Do đó, có thể coi mạng Bayes là một cơ chế cho việc xây dựng tự động các mở rộng của định lý Bayes cho các bài toán phức tạp hơn.

**Ứng dụng**Mạng Bayes được dùng cho việc mô hình hóa tri thức trong các mạng điều hòa gene (gene regulatory network), trong các hệ thống y học, kỹ thuật, phân tích văn bản, xử lý ảnh, dung hợp dữ liệu, và các hệ hỗ trợ quyết định (decision support system).

**- Ứng dụng mạng Bayes trong lĩnh vực hệ thống khuyến nghị :** có thể giúp cải thiện khả năng đề xuất sản phẩm hoặc nội dung phù hợp với từng người dùng dựa trên các thông tin khách hàng đã có sẵn. Dưới đây là một số ứng dụng cụ thể:

1. Hệ thống khuyến nghị sản phẩm: Mạng Bayes có thể được sử dụng để xác định xác suất một khách hàng sẽ quan tâm đến các sản phẩm cụ thể dựa trên dữ liệu lịch sử mua hàng của họ, thông tin cá nhân và các yếu tố khác. Các bước tiếp theo có thể là tạo ra danh sách khuyến nghị các sản phẩm tương tự hoặc phù hợp với sở thích và hành vi mua hàng của khách hàng.

2. Hệ thống khuyến nghị đa năng: Mạng Bayes cũng có thể được sử dụng để xác định xác suất một người dùng sẽ quan tâm hoặc tương tác với các nội dung khác nhau, chẳng hạn như bài viết, video, hoặc các sự kiện sắp tới. Dựa trên thông tin về hành vi trước đây của người dùng, hệ thống có thể đề xuất những nội dung phù hợp với sở thích và quan tâm của người dùng.

3. Hệ thống khuyến nghị chủ động: Mạng Bayes cũng có thể được sử dụng để đo độ phù hợp giữa nhu cầu và mong muốn của người dùng. Hệ thống có thể lưu trữ thông tin về sở thích của người dùng và dựa trên đó, đề xuất những nội dung, sản phẩm, hoặc dịch vụ phù hợp cũng như tạo ra các thông báo tương tác để thu hút và duy trì sự quan tâm của người dùng.

4. Hệ thống khuyến nghị kiểm soát rủi ro: Mạng Bayes có thể được sử dụng để đo lường và dự đoán mức độ rủi ro của một giao dịch hoặc quyết định cụ thể. Dựa trên các dữ liệu lịch sử về rủi ro và yếu tố có liên quan, hệ thống có thể đưa ra các khuyến nghị để giảm thiểu rủi ro và tăng cường an toàn cho người dùng.

Mạng Bayes có thể được áp dụng đa dạng trong lĩnh vực hệ thống khuyến nghị để cải thiện trải nghiệm người dùng và tăng cường hiệu quả kinh doanh.

**Ứng dụng mạng Bayes trong lĩnh vực hệ thống khuyến nghị có thể mang lại nhiều lợi ích, bao gồm:**

1. Đề xuất kết quả tốt hơn: Mạng Bayes có khả năng tính toán xác suất đề xuất cho từng mục tiêu dựa trên thông tin hiện có. Điều này giúp hệ thống khuyến nghị ra được kết quả tổng thể mà người dùng có thể quan tâm đến. Bằng cách sử dụng mô hình xác suất này, hệ thống có khả năng cung cấp các đề xuất phù hợp và chính xác hơn.

2. Cải thiện trải nghiệm người dùng: Với việc sử dụng mạng Bayes, hệ thống có khả năng hiểu rõ hơn về sở thích và hành vi của từng người dùng. Điều này giúp cá nhân hóa đề xuất và tối ưu hóa trải nghiệm người dùng. Khi người dùng tin tưởng và hài lòng với kết quả khuyến nghị, họ có xu hướng tiếp tục sử dụng hệ thống và tăng khả năng tương tác và mua hàng.

3. Tăng khả năng dự đoán: Mạng Bayes dựa trên kết quả của quá khứ để hiểu và dự đoán hành vi tương lai của người dùng. Thông qua việc thu thập thông tin liên quan đến mục tiêu khuyến nghị và phân tích xác suất, mạng Bayes có thể xác định các yếu tố quan trọng có ảnh hưởng đến kết quả mong muốn. Điều này cung cấp cho hệ thống khuyến nghị khả năng dự đoán tốt hơn và nâng cao khả năng dự báo kết quả.

4. Tối ưu hóa chiến lược khuyến nghị: Mạng Bayes cung cấp một khung phương pháp linh hoạt để phân tích dữ liệu và tối ưu hóa chiến lược khuyến nghị. Dựa trên phân tích xác suất và các yếu tố liên quan, hệ thống có thể tìm kiếm, so sánh và áp dụng các chiến lược tốt nhất để đưa ra gợi ý phù hợp cho người dùng.

Tổng quan, ứng dụng mạng Bayes trong hệ thống khuyến nghị có thể cải thiện đáng kể khả năng đề xuất kết quả phù hợp và tăng cường trải nghiệm người dùng, dẫn đến sự tăng cường hiệu quả kinh doanh và sự hài lòng của người dùng.

* 1. Mạng Nerual

**Tổng quan về mạng neural**

**Mạng neural nhân tạo**

Mạng neural nhân tạo (Artifical Neural Networks) mô phỏng lại mạng noron sinh học là một cấu trúc khối gồm các đơn vị tính toán đơn giản được liên kết chặt chẽ với nhautrong đó các liên kết giữa các noron quyết định chức năng của mạng.

**Các đặc trưng cơ bản của mạng nơron**

- Gồm một tập các đơn vị xử lý (các noron nhân tạo)

- Trạng thái kích hoạt hay đầu ra của đơn vị xử lý

- Liên kết giữa các đơn vị. Xét tổng quát, mỗi liên kết được định nghĩa bởi một trọng số Wjk cho ta biết hiệu ứng mà tín hiệu của đơn vị j có trên đơn vị k

- Một luật lan truyền quyết định cách tính tín hiệu ra của từng đơn vị từ đầu vào của nó

- Một hàm kích hoạt, hay hàm chuyển (activation function, transfer function), xác định mức độ kích hoạt khác dựa trên mức độ kích hoạt hiện tại

- Một đơn vị điều chỉnh (độ lệch) (bias, offset) của mỗi đơn vị

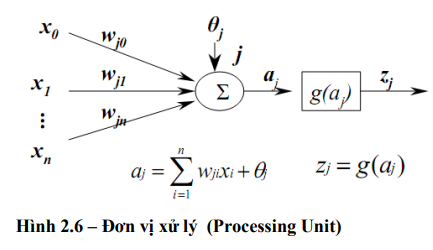
- Phương pháp thu thập thông tin (luật học - learning rule)

- Môi trường hệ thống có thể hoạt động.

**Các thành phần cơ bản của mạng nơron nhân tạo**

**Đơn vị xử lý**

Còn được gọi là một neural hay một nút (node), thực hiện một công việc rất đơn giản: nó nhận tín hiệu vào từ các đơn vị phía trước hay một nguồn bên ngoài và sử dụng chúng để tính tín hiệu ra sẽ được lan truyền sang các đơn vị khác.



**Hình 2.6 – Đơn vị xử lý (Processing Unit)**

Trong đó:

xi : các đầu vào

wji : các trọng số tương ứng với các đầu vào

θj : độ lệch (bias)

aj : đầu vào mạng (net-input)

zj : đầu ra của nơron

g(x): hàm chuyển (hàm kích hoạt).

Trong một mạng nơron có ba kiểu đơn vị:

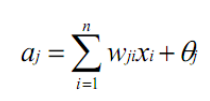
- Các đơn vị đầu vào (Input units), nhận tín hiệu từ bên ngoài.

- Các đơn vị đầu ra (Output units), gửi dữ liệu ra bên ngoài.

- Các đơn vị ẩn (Hidden units), tín hiệu vào (input) và ra (output) của nó nằm trong mạng.

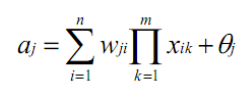
Mỗi đơn vị j có thể có một hoặc nhiều đầu vào: x0, x1, x2, … xn, nhưng chỉ có một đầu ra zj. Một đầu vào tới một đơn vị có thể là dữ liệu từ bên ngoài mạng, hoặc đầu ra của một đơn vị khác, hoặc là đầu ra của chính nó.

**Hàm kết hợp**

Mỗi một đơn vị trong một mạng kết hợp các giá trị đưa vào nó thông qua các liên kết với các đơn vị khác, sinh ra một giá trị gọi là net input. Hàm thực hiện nhiệm vụ này gọi là hàm kết hợp (combination function), được định nghĩa bởi một luật lan truyền cụ thể. Trong phần lớn các mạng nơron, chúng ta giả sử rằng mỗi một đơn vị cung cấp một bộ cộng như là đầu vào cho đơn vị mà nó có liên kết. Tổng đầu vào đơn vị j đơn giản chỉ là tổng trọng số của các đầu ra riêng lẻ từ các đơn vị kết nối cộng thêm ngưỡng hay độ lệch (bias) θj :

Trường hợp wji > 0, nơron được coi là đang ở trong trạng thái kích thích. Tương tự, nếu như wji < 0, nơron ở trạng thái kiềm chế. Chúng ta gọi các đơn vị với luật lan truyền như trên là các sigma units.

Trong một vài trường hợp người ta cũng có thể sử dụng các luật lan truyền phức tạp hơn. Một trong số đó là luật sigma-pi, có dạng như sau:



Rất nhiều hàm kết hợp sử dụng một "độ lệch" hay "ngưỡng" để tính net input tới đơn vị. Đối với một đơn vị đầu ra tuyến tính, thông thường, θj được chọn là hằng số và trong bài toán xấp xỉ đa thức θj = 1.

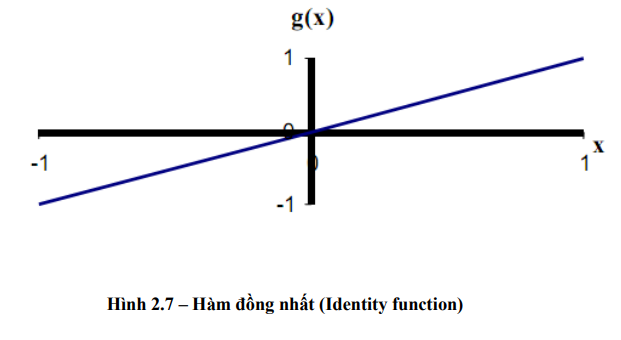
**Hàm kích hoạt**

Phần lớn các đơn vị trong mạng nơron chuyển net input bằng cách sử dụng một hàm vô hướng (scalar-to-scalar function) gọi là hàm kích hoạt, kết quả của hàm này là một giá trị gọi là mức độ kích hoạt của đơn vị (unit's activation). Loại trừ khả năng đơn vị đó thuộc lớp ra, giá trị kích hoạt được đưa vào một hay nhiều đơn vị khác. Các hàm kích hoạt thường bị ép vào một khoảng giá trị xác định, do đó thường được gọi là các hàm bẹp (squashing). Các hàm kích hoạt hay được sử dụng là:

* **Hàm đồng nhất (Linear function, Identity function )**

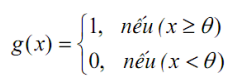


Nếu coi các đầu vào là một đơn vị thì chúng sẽ sử dụng hàm này. Đôi khi một hằng số được nhân với net-input để tạo ra một hàm đồng nhất.

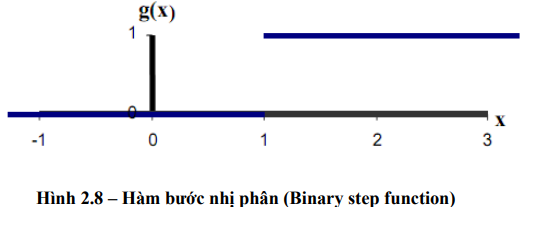


* **Hàm bước nhị phân (Binary step function, Hard limit function)**

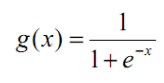
Hàm này cũng được biết đến với tên "Hàm ngưỡng" (Threshold function hay Heaviside function). Đầu ra của hàm này được giới hạn vào một trong hai giá trị:



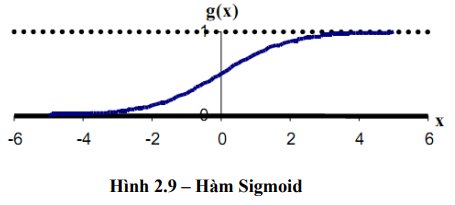
Dạng hàm này được sử dụng trong các mạng chỉ có một lớp. Trong hình vẽ sau, θ được chọn bằng 1.



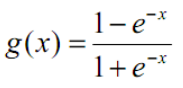
* **Hàm sigmoid (Sigmoid function (logsig))**



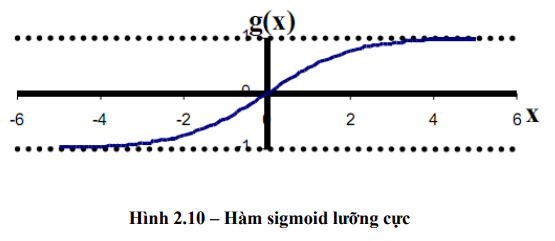
Hàm này đặc biệt thuận lợi khi sử dụng cho các mạng được huấn luyện (trained) bởi thuật toán Lan truyền ngược (back-propagation), bởi vì nó dễ lấy đạo hàm, do đó có thể giảm đáng kể tính toán trong quá trình huấn luyện. Hàm này được ứng dụng cho các chương trình ứng dụng mà các đầu ra mong muốn rơi vào khoảng [0,1].



* **Hàm sigmoid lưỡng cực (Bipolar sigmoid function (tansig))**



Hàm này có các thuộc tính tương tự hàm sigmoid. Nó làm việc tốt đối với các ứng dụng có đầu ra yêu cầu trong khoảng [-1,1].



Các hàm chuyển của các đơn vị ẩn (hidden units) là cần thiết để biểu diễn sự phi tuyến vào trong mạng. Lý do là hợp thành của các hàm đồng nhất là một hàm đồng nhất. Mặc dù vậy nhưng nó mang tính chất phi tuyến (nghĩa là, khả năng biểu diễn các hàm phi tuyến) làm cho các mạng nhiều tầng có khả năng rất tốt trong biểu diễn các ánh xạ phi tuyến. Tuy nhiên, đối với luật học lan truyền ngược, hàm phải khả vi (differentiable) và sẽ có ích nếu như hàm được gắn trong một khoảng nào đó. Do vậy, hàm sigmoid là lựa chọn thông dụng nhất.

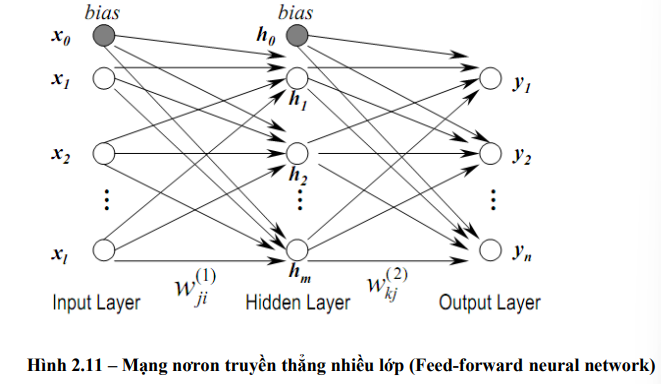
Đối với các đơn vị đầu ra (output units), các hàm chuyển cần được chọn sao cho phù hợp với sự phân phối của các giá trị đích mong muốn. Chúng ta đã thấy rằng đối với các giá trị ra trong khoảng [0,1], hàm sigmoid là có ích; đối với các giá trị đích mong muốn là liên tục trong khoảng đó thì hàm này cũng vẫn có ích, nó có thể cho ta các giá trị ra hay giá trị đích được căn trong một khoảng của hàm kích hoạt đầu ra. Nhưng nếu các giá trị đích không được biết trước khoảng xác định thì hàm hay được sử dụng nhất là hàm đồng nhất (identity function). Nếu giá trị mong muốn là dương nhưng không biết cận trên thì nên sử dụng một hàm kích hoạt dạng mũ (exponential output activation function).

**Các hình trạng của mạng**

Hình trạng của mạng được định nghĩa bởi: số lớp (layers), số đơn vị trên mỗi lớp, và sự liên kết giữa các lớp như thế nào. Các mạng về tổng thể được chia thành hai loại dựa trên cách thức liên kết các đơn vị:

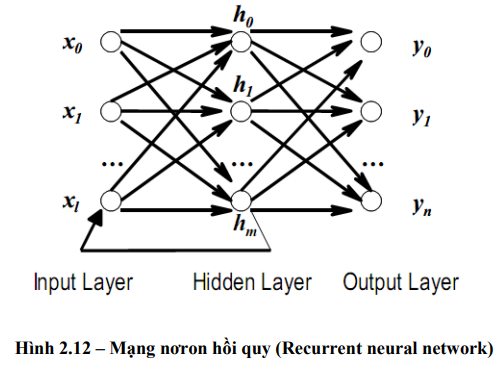
**Mạng truyền thẳng (Feed-forward neural network):**

Dòng dữ liệu từ đơn vị đầu vào đến đơn vị đầu ra chỉ được truyền thẳng. Việc xử lý dữ liệu có thể mở rộng ra nhiều lớp, nhưng không có các liên kết phản hồi. Nghĩa là, các liên kết mở rộng từ các đơn vị đầu ra tới các đơn vị đầu vào trong cùng một lớp hay các lớp trước đó là không cho phép.



**Mạng hồi quy (Recurrent neural network):**

Có chứa các liên kết ngược. Khác với mạng truyền thẳng, các thuộc tính động của mạng mới quan trọng. Trong một số trường hợp, các giá trị kích hoạt của các đơn vị trải qua quá trình nới lỏng (tăng giảm số đơn vị và thay đổi các liên kết) cho đến khi mạng đạt đến một trạng thái ổn định và các giá trị kích hoạt không thay đổi nữa. Trong các ứng dụng khác mà cách chạy động tạo thành đầu ra của mạng thì những sự thay đổi các giá trị kích hoạt là đáng quan tâm.

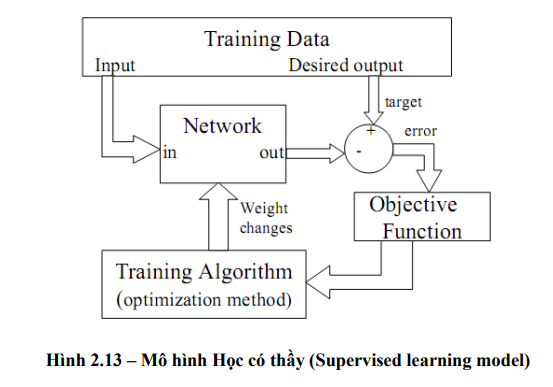


**Huấn luyện mạng**

Chức năng của một mạng nơron được quyết định bởi các nhân tố như: hình trạng mạng (số lớp, số đơn vị trên mỗi tầng, và cách mà các lớp được liên kết với nhau) và các trọng số của các liên kết bên trong mạng. Hình trạng của mạng thường là cố định, và các trọng số được quyết định bởi một thuật toán huấn luyện (training algorithm). Tiến trình điều chỉnh các trọng số để mạng “nhận biết” được quan hệ giữa đầu vào và đích mong muốn được gọi là học (learning) hay huấn luyện (training). Rất nhiều thuật toán học đã được phát minh để tìm ra tập trọng số tối ưu làm giải pháp cho các bài toán. Các thuật toán đó có thể chia làm hai nhóm chính: Học có thầy (Supervised learning) và Học không có thầy (Unsupervised Learning).

**Học có thầy (Supervised learning):**

Mạng được huấn luyện bằng cách cung cấp cho nó các cặp mẫu đầu vào và các đầu ra mong muốn (target values). Các cặp được cung cấp bởi "thầy giáo", hay bởi hệ thống trên đó mạng hoạt động. Sự khác biệt giữa các đầu ra thực tế so với các đầu ra mong muốn được thuật toán sử dụng để thích ứng các trọng số trong mạng. Điều này thường được đưa ra như một bài toán xấp xỉ hàm số - cho dữ liệu huấn luyện bao gồm các cặp mẫu đầu vào x, và một đích tương ứng t, mục đích là tìm ra hàm f(x) thoả mãn tất cả các mẫu học đầu vào.

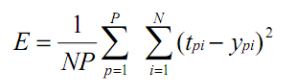


**Học không có thầy (Unsupervised Learning):**

Với cách học không có thầy, không có phản hồi từ môi trường để chỉ ra rằng đầu ra của mạng là đúng. Mạng sẽ phải khám phá các đặc trưng, các điều chỉnh, các mối tương quan, hay các lớp trong dữ liệu vào một cách tự động. Trong thực tế, đối với phần lớn các biến thể của học không có thầy, các đích trùng với đầu vào. Nói một cách khác, học không có thầy luôn thực hiện một công việc tương tự như một mạng tự liên hợp, cô đọng thông tin từ dữ liệu vào.

**Hàm mục tiêu**

Để huấn luyện một mạng và xét xem nó thực hiện tốt đến đâu, ta cần xây dựng một hàm mục tiêu (hay hàm giá) để cung cấp cách thức đánh giá khả năng hệ thống một cách không nhập nhằng. Việc chọn hàm mục tiêu là rất quan trọng bởi vì hàm này thể hiện các mục tiêu thiết kế và quyết định thuật toán huấn luyện nào có thể được áp dụng. Để phát triển một hàm mục tiêu đo được chính xác cái chúng ta muốn không phải là việc dễ dàng. Một vài hàm cơ bản được sử dụng rất rộng rãi. Một trong số chúng là hàm tổng bình phương lỗi (sum of squares error function)



Trong đó

p: số thứ tự mẫu trong tập huấn luyện

i : số thứ tự của đơn vị đầu ra

tpi và ypi : tương ứng là đầu ra mong muốn và đầu ra thực tế của mạng cho đơn vị đầu ra thứ i trên mẫu thứ p.

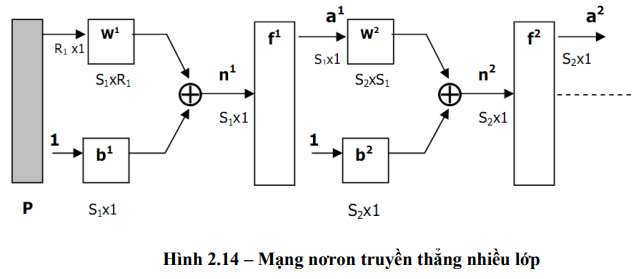
Trong các ứng dụng thực tế, nếu cần thiết có thể làm phức tạp hàm số với một vài yếu tố khác để có thể kiểm soát được sự phức tạp của mô hình.

**Mạng truyền thẳng và thuật toán lan truyền ngược**

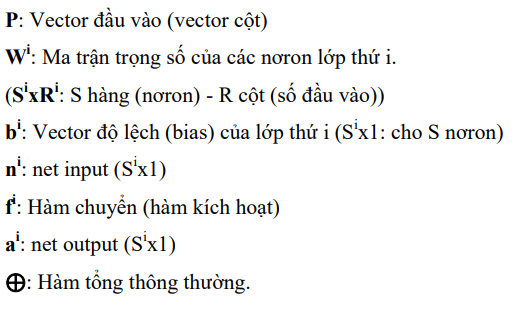
Để đơn giản và tránh hiểu nhầm, mạng truyền thẳng trình bày trong phần này là mạng truyền thẳng có nhiều lớp (MLP - MultiLayer Perceptron). Đây là một trong những mạng truyền thẳng điển hình, thường được sử dụng trong các hệ thống nhận dạng.

**Mạng truyền thẳng MLP**

Một mạng truyền thẳng nhiều lớp bao gồm một lớp vào, một lớp ra và một hoặc nhiều lớp ẩn. Các nơron đầu vào thực chất không phải các nơron theo đúng nghĩa, bởi lẽ chúng không thực hiện bất kỳ một tính toán nào trên dữ liệu vào, đơn giản nó chỉ tiếp nhận các dữ liệu vào và chuyển cho các lớp kế tiếp. Các nơron ở lớp ẩn và lớp ra mới thực sự thực hiện các tính toán, kết quả được định dạng bởi hàm đầu ra (hàm chuyển). Cụm từ “truyền thẳng” (feed forward) (không phải là trái nghĩa của lan truyền ngược) liên quan đến một thực tế là tất cả các nơron chỉ có thể được kết nối với nhau theo một hướng: tới một hay nhiều các nơron khác trong lớp kế tiếp (loại trừ các nơron ở lớp ra).



Trong đó



Mỗi liên kết gắn với một trọng số, trọng số này được thêm vào trong quá trình tín hiệu đi qua liên kết đó. Các trọng số có thể dương, thể hiện trạng thái kích thích, hay âm, thể hiện trạng thái kiềm chế. Mỗi nơron tính toán mức kích hoạt của chúng bằng cách cộng tổng các đầu vào và đưa ra hàm chuyển. Một khi đầu ra của tất cả các nơron trong một lớp mạng cụ thể đã thực hiện xong tính toán thì lớp kế tiếp có thể bắt đầu thực hiện tính toán của mình bởi vì đầu ra của lớp hiện tại tạo ra đầu vào của lớp kế tiếp. Khi tất cả các nơron đã thực hiện tính toán thì kết quả được trả lại bởi các nơron đầu ra. Tuy nhiên, có thể là chưa đúng yêu cầu, khi đó một thuật toán huấn luyện cần được áp dụng để điều chỉnh các tham số của mạng.

Xét trường hợp mạng có hai lớp như hình 2.14, công thức tính toán cho đầu ra như sau:



Mạng có nhiều lớp có khả năng tốt hơn là các mạng chỉ có một lớp, chẳng hạn như mạng hai lớp với lớp thứ nhất sử dụng hàm sigmoid và lớp thứ hai dùng hàm đồng nhất có thể áp dụng để xấp xỉ các hàm toán học khá tốt, trong khi các mạng chỉ có một lớp thì không có khả năng này.

**Thiết kế cấu trúc mạng**

Mặc dù, về mặt lý thuyết, có tồn tại một mạng có thể mô phỏng một bài toán với độ chính xác bất kỳ. Tuy nhiên, để có thể tìm ra mạng này không phải là điều đơn giản. Để định nghĩa chính xác một kiến trúc mạng như: cần sử dụng bao nhiêu lớp ẩn, mỗi lớp ẩn cần có bao nhiêu đơn vị xử lý cho một bài toán cụ thể là một công việc hết sức khó khăn.

**Số lớp ẩn:**

Vì các mạng có hai lớp ẩn có thể thể hiện các hàm với dáng điệu bất kỳ, nên, về lý thuyết, không có lý do nào sử dụng các mạng có nhiều hơn hai lớp ẩn. Người ta đã xác định rằng đối với phần lớn các bài toán cụ thể, chỉ cần sử dụng một lớp ẩn cho mạng là đủ. Các bài toán sử dụng hai lớp ẩn hiếm khi xảy ra trong thực tế. Thậm chí đối với các bài toán cần sử dụng nhiều hơn một lớp ẩn thì trong phần lớn các trường hợp trong thực tế, sử dụng chỉ một lớp ẩn cho ta hiệu năng tốt hơn là sử dụng nhiều hơn một lớp. Việc huấn luyện mạng thường rất chậm khi mà số lớp ẩn sử dụng càng nhiều.

**Số đơn vị trong lớp ẩn:**

Một vấn đề quan trọng trong việc thiết kế một mạng là cần có bao nhiêu đơn vị trong mỗi lớp. Sử dụng quá ít đơn vị có thể dẫn đến việc không thể nhận dạng được các tín hiệu đầy đủ trong một tập dữ liệu phức tạp, hay thiếu ăn khớp (underfitting). Sử dụng quá nhiều đơn vị sẽ tăng thời gian luyện mạng, có lẽ là quá nhiều để luyện khi mà không thể luyện mạng trong một khoảng thời gian hợp lý. Số lượng lớn các đơn vị có thể dẫn đến tình trạng thừa ăn khớp (overfitting), trong trường hợp này mạng có quá nhiều thông tin, hoặc lượng thông tin trong tập dữ liệu mẫu (training set) không đủ các dữ liệu đặc trưng để huấn luyện mạng.

Số lượng tốt nhất của các đơn vị ẩn phụ thuộc vào rất nhiều yếu tố - số đầu vào, đầu ra của mạng, số trường hợp trong tập mẫu, độ nhiễu của dữ liệu đích, độ phức tạp của hàm lỗi, kiến trúc mạng và thuật toán luyện mạng.

Trong phần lớn các trường hợp, không có một cách để có thể dễ dàng xác định được số tối ưu các đơn vị trong lớp ẩn mà không phải luyện mạng sử dụng số các đơn vị trong lớp ẩn khác nhau và dự báo lỗi tổng quát hóa của từng lựa chọn. Cách tốt nhất là sử dụng phương pháp thử-sai (trial-and-error). Trong thực tế, có thể sử dụng phương pháp Lựa chọn tiến (forward selection) hay Lựa chọn lùi (backward selection) để xác định số đơn vị trong lớp ẩn.

Lựa chọn tiến bắt đầu với việc chọn một luật hợp lý cho việc đánh giá hiệu năng của mạng. Sau đó, ta chọn một số nhỏ các đơn vị ẩn, luyện và thử mạng; ghi lại hiệu năng của mạng. Sau đó, tăng một chút số đơn vị ẩn; luyện và thử lại cho đến khi lỗi là chấp nhận được, hoặc không có tiến triển đáng kể so với trước.

Lựa chọn lùi, ngược với lựa chọn tiến, bắt đầu với một số lớn các đơn vị trong lớp ẩn, sau đó giảm dần đi. Quá trình này rất tốn thời gian nhưng sẽ giúp ta tìm được số lượng đơn vị phù hợp cho lớp ẩn.

**Thuật toán lan truyền ngược (Back-Propagation)**

Cần có một sự phân biệt giữa kiến trúc của một mạng và thuật toán học của nó, các mô tả trong các mục trên mục đích là nhằm làm rõ các yếu tố về kiến trúc của mạng và cách mà mạng tính toán các đầu ra từ tập các đầu vào. Sau đây là mô tả của thuật toán học sử dụng để điều chỉnh hiệu năng của mạng sao cho mạng có khả năng sinh ra được các kết quả mong muốn.

Về cơ bản có hai dạng thuật toán để luyện mạng: học có thầy và học không có thầy. Các mạng nơron truyền thẳng nhiều lớp được luyện bằng phương pháp học có thầy. Phương pháp này căn bản dựa trên việc yêu cầu mạng thực hiện chức năng của nó và sau đó trả lại kết quả, kết hợp kết quả này với các đầu ra mong muốn để điều chỉnh các tham số của mạng, nghĩa là mạng sẽ học thông qua những sai sót của nó.

Thuật toán lan truyền ngược là dạng tổng quát của thuật toán trung bình bình phương tối thiểu (Least Means Square-LMS). Thuật toán này thuộc dạng thuật toán xấp xỉ để tìm các điểm mà tại đó hiệu năng của mạng là tối ưu. Chỉ số tối ưu (performance index) thường được xác định bởi một hàm số của ma trận trọng số và các đầu vào nào đó mà trong quá trình tìm hiểu bài toán đặt ra.

Bỏ qua sự phức tạp về mặt toán học, thuận toán có thể phát biểu đơn giản như sau:

* ***Bước 1: Lan truyền xuôi các tính toán trong mạng truyền thẳng*** - Khi đó, đầu ra của một lớp trở thành đầu vào của lớp kế tiếp. Phương trình thể hiện hoạt động này như sau (trong đó M là số lớp trong mạng) :



- Các nơron trong lớp thứ nhất nhận các tín hiệu từ bên ngoài (với p chính là điểm bắt đầu của phương trình hình 2.14)



- Đầu ra của lớp cuối cùng được xem là đầu ra của mạng:



* ***Bước 2: Lan truyền lỗi (hay độ nhạy cảm) ngược lại qua mạng***

- Thuật toán lan truyền ngược sử dụng chỉ số hiệu năng là trung bình bình phương lỗi của đầu ra so với giá trị đích. Đầu vào của thuật toán chính là tập các cặp mô tả hoạt động đúng của mạng:



Trong đó pi là một đầu vào và ti là đầu ra mong muốn tương ứng, với i=1..Q.

- Mỗi đầu vào đưa vào mạng, đầu ra của mạng đối với nó được đem so sánh với đầu ra mong muốn.Thuật toán sẽ điều chỉnh các tham số của mạng để tối thiểu hóa trung bình bình phương lỗi.

* ***Bước 3: Cập nhật lại các trọng số và độ lệch tương ứng***

Có nhiều thuật toán học máy được sử dụng trong mạng Neural, dưới đây là một số thuật toán phổ biến:

1. Feedforward Neural Networks: Đây là loại mạng Neural cơ bản nhất và là cơ sở cho các mạng Neural phức tạp hơn. Đầu vào đi từ các lớp đầu tiên, được truyền qua các lớp ẩn và cuối cùng đến lớp đầu ra. Mạng này không có kết nối ngược và không lưu trữ bất kỳ trạng thái nào giữa các lần tính toán.

2. Convolutional Neural Networks (CNNs): CNNs là loại mạng Neural thường được sử dụng cho hình ảnh và dữ liệu không gian. Chúng sử dụng các lớp tích chập để trích xuất các đặc trưng từ dữ liệu đầu vào và có khả năng học và nhận dạng các mẫu trong hình ảnh.

3. Recurrent Neural Networks (RNNs): RNNs được sử dụng để xử lý dữ liệu tuần tự hoặc dữ liệu có thời gian liên quan. Với khả năng lưu trạng thái trước đó, RNNs có thể xem xét thông tin từ quá khứ trong việc dự đoán thông tin tại thời điểm hiện tại. Một biến thể phổ biến của RNNs là Long Short-Term Memory (LSTM), được sử dụng để giải quyết vấn đề biến mất gradient.

4. Generative Adversarial Networks (GANs): GANs là một kiểu mạng Neural đặc biệt, bao gồm hai mạng: một mạng sinh (generator) và một mạng phân biệt (discriminator). Mạng sinh cố gắng tạo ra dữ liệu mới giống với dữ liệu huấn luyện, trong khi mạng phân biệt cố gắng phân biệt dữ liệu giả tạo từ dữ liệu thật. Qua quá trình cạnh tranh, GANs có thể tạo ra dữ liệu mới có chất lượng tốt.

5. Autoencoders: Autoencoders là mạng Neural được sử dụng trong việc nén và phục hồi dữ liệu. Mục tiêu của autoencoders là học một representational space nhỏ hơn cho dữ liệu đầu vào và tái tạo lại dữ liệu gần giống với dữ liệu ban đầu.

Đây chỉ là một số thuật toán học máy mạng Neural phổ biến. Còn nhiều thuật toán khác như MLP, LSTM, Gated Recurrent Unit (GRU), và nhiều biến thể và cải tiến khác được phát triển để phù hợp với các vấn đề khác nhau.

Một hệ thống khuyến nghị là một ứng dụng trong đó mục tiêu là đề xuất các sản phẩm, dịch vụ hoặc nội dung phù hợp nhất cho người dùng dựa trên hành vi hoặc thông tin cá nhân của họ. Mạng Neural đã được sử dụng thành công trong hệ thống khuyến nghị để cải thiện chất lượng và độ chính xác của các đề xuất.

Mạng Neural là một phương pháp trong lĩnh vực học máy và trí tuệ nhân tạo mô phỏng cấu trúc và hoạt động của hệ thống thần kinh sinh học trong não người. Nó được sử dụng để xử lý thông tin, nhận dạng mẫu và học từ dữ liệu.

Áp dụng mạng Neural vào một hệ thống khuyến nghị thường bao gồm các bước sau:

1. Tiền xử lý dữ liệu: Dữ liệu từ người dùng và sản phẩm cần được xử lý và biểu diễn dưới dạng đầu vào phù hợp cho mạng Neural. Ví dụ, thông tin về người dùng có thể được mã hóa thành vector đặc trưng, trong khi thông tin về sản phẩm có thể được mã hóa dưới dạng vector tương tự.
2. Xây dựng mô hình mạng Neural: Mạng Neural có thể được xây dựng với các kiến trúc phù hợp như mạng Neural lặp lại (RNN), mạng Neural tái tạo (Autoencoders) hoặc mạng Neural đầy đủ đóng (Fully Connected Neural Networks). Mạng Neural sẽ nhận đầu vào từ người dùng và sản phẩm để dự đoán sở thích và hành vi.
3. Huấn luyện mô hình: Mô hình mạng Neural được huấn luyện trên dữ liệu huấn luyện, trong đó dữ liệu được gán nhãn với đánh giá hoặc phản hồi từ người dùng. Quá trình này giúp mô hình học cách tương tác và dự đoán đúng các sở thích và đề xuất phù hợp.
4. Đánh giá và điều chỉnh: Mô hình được đánh giá dựa trên các tiêu chí như độ chính xác, độ phù hợp hoặc độ tương đồng. Nếu cần, mô hình có thể được điều chỉnh lại để nâng cao hiệu suất và đáp ứng tốt hơn nhu cầu của người dùng.
5. Triển khai và cải tiến: Khi mô hình có hiệu suất tốt, nó có thể được triển khai vào hệ thống khuyến nghị để đề xuất các sản phẩm, dịch vụ hoặc nội dung cho người dùng. Hệ thống cũng có thể được cải tiến và thích nghi với sự phản hồi và dữ liệu mới.

Tóm lại, mạng Neural có thể được áp dụng vào hệ thống khuyến nghị để tạo ra các đề xuất phù hợp và tối ưu cho người dùng dựa trên các đặc trưng và hành vi của họ. Việc sử dụng mạng Neural đem lại khả năng học và cải thiện chất lượng khuyến nghị, đồng thời cung cấp khả năng linh hoạt và mở rộng cho các ứng dụng khác nhau.

1. Hệ thống khuyến nghị
   1. Khái niệm

Hệ thống khuyến nghị (Recommender System) sử dụng công nghệ AI để thực hiện phân tích và hiểu khối dữ liệu cá nhân, từ đó, đưa ra các dự đoán, gợi ý đề xuất phù hợp với sở thích của người dùng tại thời điểm bất kỳ trên các ứng dụng và nền tảng trực tuyến giúp tiết kiệm thời gian tìm kiếm, truy cập nội dung dễ dàng, đồng thời, giúp doanh nghiệp nâng cao trải nghiệm khách hàng.

Nhắc đến AI, nhiều người sẽ nghĩ ngay tới các ứng dụng nhận dạng ảnh, chatbot, hay xe tự lái. Trên thực tế, ứng dụng AI đã được sử dụng phổ biến từ lâu và hiện có mặt trên hầu hết các nền tảng trực tuyến, không phải chỉ trong các ứng dụng nói trên mà trong các hệ thống khuyến nghị (recommender systems hay RS).

Hệ thống khuyến nghị là các công cụ và kỹ thuật phần mềm cung cấp các đề xuất về các hạng mục/nội dung đề xuất cho người dùng. Các đề xuất liên quan đến các quá trình ra quyết định khác nhau tại những thời điểm bất kỳ cho người dùng, ví dụ như mua hàng hóa nào, nghe nhạc gì hoặc đọc tin tức gì. Hệ thống khuyến nghị có thể sử dụng các kỹ thuật AI như học máy để hiểu được sở thích của người dùng, nhờ vậy đưa ra những dự đoán và khuyến nghị những sản phẩm/dịch vụ/nội dung mà người dùng có thể quan tâm (hàng hoá, phim, sách, video, tin tức, bài hát, khách sạn, khoá học, v.v.).

* 1. Vai trò

Hệ thống khuyến nghị có nhiều vai trò. Chúng giúp người dùng tìm kiếm sản phẩm, dịch vụ hoặc nội dung phù hợp với sở thích và nhu cầu của họ một cách nhanh chóng và dễ dàng. Điều này giúp tiết kiệm thời gian và công sức cho người dùng, đồng thời cũng giúp các nhà cung cấp sản phẩm và dịch vụ tiếp cận được đối tượng khách hàng mục tiêu một cách hiệu quả hơn. Hệ thống khuyến nghị cũng có thể giúp người dùng khám phá ra những sản phẩm, dịch vụ hoặc nội dung mới mà họ có thể chưa biết đến trước đó. Đây là một công cụ hữu ích để tăng trải nghiệm người dùng và tăng doanh thu cho các nhà cung cấp.

Hệ thống khuyến nghị được ứng dụng trong nhiều lĩnh vực khác nhau, bao gồm thương mại điện tử, giải trí, du lịch, giáo dục và nhiều lĩnh vực khác. Chúng giúp người dùng tìm kiếm sản phẩm, dịch vụ hoặc nội dung phù hợp với sở thích và nhu cầu của họ một cách nhanh chóng và dễ dàng. Điều này giúp tiết kiệm thời gian và công sức cho người dùng, đồng thời cũng giúp các nhà cung cấp sản phẩm và dịch vụ tiếp cận được đối tượng khách hàng mục tiêu một cách hiệu quả hơn. Ví dụ, khi bạn mua sắm trực tuyến, hệ thống khuyến nghị có thể đề xuất cho bạn các sản phẩm liên quan đến những gì bạn đã xem hoặc mua trước đó. Khi bạn xem phim hoặc nghe nhạc trực tuyến, hệ thống khuyến nghị có thể đề xuất cho bạn các bộ phim hoặc bài hát liên quan đến sở thích của bạn. Hệ thống khuyến nghị cũng có thể giúp người dùng khám phá ra những sản phẩm, dịch vụ hoặc nội dung mới mà họ có thể chưa biết đến trước đó. Ngoài ra, hệ thống khuyến nghị cũng được sử dụng trong các ứng dụng hẹn hò để đề xuất cho người dùng những người có cùng sở thích và lý tưởng.

Hệ thống khuyến nghị đóng vai trò quan trọng trong thương mại điện tử. Chúng giúp người dùng tìm kiếm sản phẩm và dịch vụ phù hợp với sở thích và nhu cầu của họ một cách nhanh chóng và dễ dàng. Điều này giúp tiết kiệm thời gian và công sức cho người dùng, đồng thời cũng giúp các nhà cung cấp sản phẩm và dịch vụ tiếp cận được đối tượng khách hàng mục tiêu một cách hiệu quả hơn. Giúp người dùng khám phá ra những sản phẩm và dịch vụ mới mà họ chưa có thể chưa biết đến trước đó.

Hệ thống khuyến nghị anime có vai trò quan trọng trong việc giúp người dùng tìm kiếm và khám phá các bộ anime phù hợp với sở thích và nhu cầu của họ. Hệ thống này sử dụng các kỹ thuật lọc cộng tác để đề xuất các bộ anime dựa trên sở thích của người dùng và các người dùng khác có cùng sở thích. Điều này giúp người dùng tiết kiệm thời gian và công sức trong việc tìm kiếm, đồng thời cũng giúp họ khám phá ra những bộ anime mới mà họ có thể chưa biết đến trước đó.

* 1. Các phương pháp khuyến nghị:

Những nghiên cứu liên quan

Phương pháp khuyến nghị dựa trên sự phổ biến (Popularity-based recommendation) là một phương pháp đơn giản nhưng hiệu quả trong việc đưa ra các đề xuất khi hệ thống không biết người dùng là ai, như thế nò hoặc một người dùng mới. Phương pháp này đưa ra các đề xuất dựa trên mức độ phổ biến của các sản phẩm/dịch vụ/nội dung trong cộng đồng người dùng.

Ví dụ, trong một hệ thống khuyến nghị phim, phương pháp này sẽ đưa ra các đề xuất cho người dùng dựa trên số lượng lượt xem hoặc số lượng lượt đánh giá cao của các bộ phim. Những bộ phim có số lượng lượt xem hoặc số lượng lượt đánh giá cao sẽ được đề xuất cho người dùng.

Phương pháp này có ưu điểm là đơn giản và dễ triển khai, tuy nhiên nó không thể cung cấp các đề xuất cá nhân hóa cho từng người dùng một cách chính xác.

Phương pháp khuyến nghị dựa trên nội dung (Content-based recommendation) là một phương pháp đưa ra các đề xuất cho người dùng dựa trên các thuộc tính của các sản phẩm/ dịch vụ/ nội dung mà người dùng đã thể hiện sự quan tâm trong quá khứ.

Ví dụ, trong một hệ thống khuyến nghị phim, phương pháp này sẽ đưa ra các đề xuất cho người dùng dựa trên các thuộc tính của các bộ phim mà người dùng đã xem và dánh giá cao trong quá khứ, chẳng hạn như thể loại, diễn viên hoặc đạo diễn.

Phương pháp này có ưu điểm là có thể cung cấp các đề xuất cá nhân hóa cho từng người dùng một cách chính xác. Tuy nhiên, nó có thể không đưa ra được các đề xuất mới lạ cho người dùng vì chỉ dựa trên sở thích của người dùng trong quá khứ.

Phương pháp khuyến nghị lọc cộng tác (Collaborative Filtering) Lọc cộng tác có hai loại chính: lọc cộng tác dựa trên người dùng (User-based Collaborative Filtering) và lọc cộng tác dựa trên mục (Item-based Collaborative Filtering).

Lọc cộng tác dựa trên người dùng tìm kiếm những người dùng có cùng kiểu xếp hạng với người dùng đang hoạt động (người dùng có dự đoán). Sử dụng xếp hạng từ những người dùng có cùng chí hướng được tìm thấy ở bước 1 để tính toán dự đoán cho người dùng đang hoạt động.

Lọc cộng tác dựa trên mục là một loại hệ thống khuyến cáo được dựa trên sự giống nhau giữa các mặt hàng tính bằng giá người dùng đã cung cấp cho các mặt hàng. Nó giúp giải quyết các vấn đề mà bộ lọc cộng tác dựa trên người dùng mắc phải, chẳng hạn như khi hệ thống có nhiều múc với ít mục được xếp hạng hơn.

Phương pháp kết hợp/ tổng hợp (Hybrid/Ensemble) được tích hợp từ hai phương pháp lọc cộng tác (Collaborative Filtering) và phương pháp khuyến nghị dựa trên nội dung (Content-based recommendation) hiệu suất của mỗi thuật toán khuyến nghị đơn lẻ này, tuy nhiên, đều có giới hạn và mỗi thuật toán có điểm mạnh và điểm yếu riêng. Trong cuộc khảo sát của Borràs và cộng sự, 2014 [2], có thể nhận thấy xu hướng ngày càng tăng trong việc khai thác các kỹ thuật lọc khuyến nghị kể từ năm 2012, chủ yếu trong các hệ thống lai. Từ năm 2008 đến năm 2011, chỉ có 25% hệ thống sử dụng phương pháp này, trong khi từ năm 2012, tỷ lệ này đã tăng lên 75%.

1. Nguyễn Đức Thuần, Nhập môn khai phá dữ liệu và quản trị tri thức, NXB Thông tin và truyền thông, 2013. [↑](#footnote-ref-1)
2. Jiawei Han and Micheline Kamber, Datamining: Concepts and Techniques, Simon Fraser University, 2011. [↑](#footnote-ref-2)
3. Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami; Mining Association Rules Between Sets of Items in Large Databases, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216, Washington, D.C., May 1993 [↑](#footnote-ref-3)