|  |
| --- |
| TRƯỜNG ĐẠI HỌC HẠ LONG |
| **KHOA CÔNG NGHỆ THÔNG TIN** |

**BÀI GIẢNG**

**HỌC MÁY**

***Ngành: Khoa học Máy tính***

**Biên soạn: PGS.TS. Nguyễn Quang Hoan**

**Hà Nội 2.2022**

**GIỚI THIỆU HỌC PHẦN**

Hiệu chỉnh một vài điểm cho phù hợp với CNTT.Hạ Long như sau:

1. Cắt chương 7. Mạng Perceptron. Lý do: CNTT Hạ Long có môn mạng nơ ron chứa Mạng Percptron.
2. Chương trình của ”Học Máy” của Hoàng Xuân Huấn hay nhưng hợp với ĐH Công nghệ, có tính ”toán học”, tính ”tổng hợp”, chưa hợp lắm với CNTT.Hạ Long mang tính ứng dụng, tính ”Kỹ nghệ: Engineering”. Nên thay vì nói nhiều về công thức, mô hình toán ta thay bằng các thuật toán
3. Đổi thành 4 phần (hầu hết vẫn các chương đó) như sau

|  |
| --- |
| - Giới thiệu học phần  **PHẦN I. GIỚI THIỆU**  **Chương 1: Giới thiệu**  1.1. Khái niệm và định nghĩa học máy  1.2. Các bài toán về học  1.3. Phân loại học máy |
| **PHẦN II. HỌC GIÁM SÁT VÀ ỨNG DỤNG** |
| **Chương 2: Học lười hay học dựa trên ví dụ: Thuật toán K-NN**  2.1. Phân lớp K-láng giềng gần nhất  2. 2. Chọn đặc trưng  2.3. Đánh giá bộ phân lớp |
| **Chương 3: Cây quyết định**  3.1. Các thuật toán học cây quyết định cơ bản  3.2. Học thống kê cây quyết định-Thuật toán Quin Lan  3.3. Học dựa trên Entroppy-Thuật toán độ lộn xộn  3.4. Thuật toán ID3  3.5. Thuật toán C4.5, C5.0  3.6. Phần mềm WEKA, Các chỉ tiêu đánh giá kết quả phân lớp  3.7. Thuật toán học cây quyết định quy nạp-ILA |
| **Chương 4: Học thống kê Bayes**  4.1. Tóm tắt Lý thuyết xác suát, Lý thuyết quyết định Bayes  4.2. Phân lớp Bayes  4.3. Thuật toán Bayes và ứng dụng cho dự báo  4.4. Đánh giá bộ phân lớp  4.5. Mạng Bayes |
| **Chương 5: Hồi quy**  5.1. Khái niệm về Hồi quy  5.2. Hồi quy tuyến tính  5.3. Hồi quy Logicstic  5.4. Hồi quy phi tuyến |
| **Chương 6. Máy học véc tơ hỗ trợ**  6.1. Khái niệm  6.2. Máy véc tơ hỗ trợ |
| **PHẦN III. HỌC KHÔNG GIÁM SÁT VÀ ỨNG DỤNG**  **Chương 7: Học không giám sát**  7.1. Ước lượng hàm mật độ  7.2. Các thuật toán phân cụm cơ bản  7.3. Thuật toán phân cụm K-Mean |
| Ôn tập và kiểm tra định kỳ |
| **PHẦN IV. HỌC TĂNG CƯỜNG VÀ ỨNG DỤNG**  **Chương 8: Học tăng cường**  8.1. Tác tử và các bài toán học  8.2. Học Q (Q-learning).  8.3. Các giải thuật tiến hóa |
| **Chương 9: Kết hợp các bộ học và các hướng phát triển**  9.1. Học tập thể  9.2. Phương pháp bỏ phiếu  9.3. Kỹ thuật tạo và kết hợp bộ nhận dạng cơ sở |
| Ôn tập và kiểm tra |

PHẦN I. GIỚI THIỆU HỆ HỌC

**CHƯƠNG 1: GIỚI THIỆU HỌC MÁY**

* 1. **Khái niệm và định nghĩa học máy**

### Khái niệm học máy

* Học theo nghĩa thông thường: tiếp thu (lấy, thu nhận) kiến thức: điều chưa biết.

Học máy cũng tương tự: Máy tính tiếp thu kiến thức, tri thức.

* Khái niệm học có nghĩa rộng giống như sự thông minh (Intelligence).Học máy là một lĩnh vực của AI[.](http://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o) Theo nghĩa tự điển, học là quá trình thu nhận **kiến thức**, **Tri thức (Knowledge),** kỹ năng do người khác truyền lại hoặc đọc đi, đọc lại, nghiền ngẫm ghi nhớ (học thuộc lòng). Rộng hơn, học bao gồm cả quá trình đúc rút **tri thức** từ các quan sát, trải nghiệm thực tiễn (Experience).
* **Tri thức (nhắc lại từ TTNT)**

Trước tiên tìm hiểu về thông tin, tri thức, dữ liệu thông qua hình 1.1 sau:

Thể hiện tri thức thế nào để đưa vào máy tính:

*a)Tri thức:* Có thể định nghĩa đơn giản như sau:

*Knowledge=Sự kiện (Events or fact)+quy luật (Rules)*

*Sự kiện:* ví dụ sự kiện 1: ”Trời mưa”=*A* (*A* là ký hiệu một biến, gọi là biến ngôn ngữ)

ví dụ sự kiện 2: ”Đất ướt”=*B* (*B* là ký hiệu một biến khác, là biến ngôn ngữ).

Như vậy sự kiện, được coi như tri thức đơn gian nhất, nó như là dữ liệu, data

*Quy luật (luật), quy tắc, hay mối quan hệ giũa các sự kiện:* Từ sự kiện *A* dẫn đến kết quả *B* hay: *A->B* hoặc tương đương:

IF *A*,

THEN *B*:

đây được gọi là luật “ (nguyên) Nhân, (dẫn tới kết ) Quả. Luật còn có tên khác: Luật sinh sản, luật sinh, luật sản xuất *(Production Rule).*

**Tri thức khác dữ liệu (Data)**

*Định nghĩa về Data và các thuộc tính*. Dữ liệu là gì? Có nhiều định nghĩa khác nhau.

DATA=THÔNG TIN?

Ta làm rõ trong các hoạt động:

* *Trong lập trình*: Data=biến (Variables)+Hằng (Constant). Ví dụ trong Assembly

Ví dụ: giải phương trình bậc 2: biến: x1, x2. Hằng có : a, b, c.

* *Trong CNPM*: dữ liệu là nguyên liệu thô đưa vào sản xuất. Ví dụ điểm của SV lớp ta là nguyên luyện thô. Chế biến: điểm trung bình cả lớp, …
* Đặc tính của dữ liệu có độ đo: Byte: đơn vị nhớ của dữ liệu. Ví dụ TRANG: 5 Bytes. Mỗi byte thể hiện gì ký tự.

VD1: Hồng=Hoongf: 6 ký tự= 6 bytes.

Một byte: H=Mã gồm 8 bits, vi dụ mã ASCII: 1001 1100 (9 C): 1byte

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *1* | *0* | *0* | *1* | *1* | *1* | *0* | *0* |

*Ví dụ: Ngày quốc khánh Việt Nam: 2.9.1945: dữ liệu đã biết*

*Ngày sinh của Quang 26.2.2001 : dữ liệu chưa biết.*

b) Định nghĩa về THÔNG TIN (INFORMATION)

*Thông tin*:

* Ngày sinh Đỗ Thùy Trang: 28.3.2001: dữ liệu: chưa biết: THÔNG TIN
* Ngày 2.9.1945: Ngày Quốc Khánh: Khai sinh nước: mọi người đề biết: KHÔNG PHẢI THÔNG TIN

Dữ liệu chưa mới có thông tin, thể hiện thông báo chưa biết.

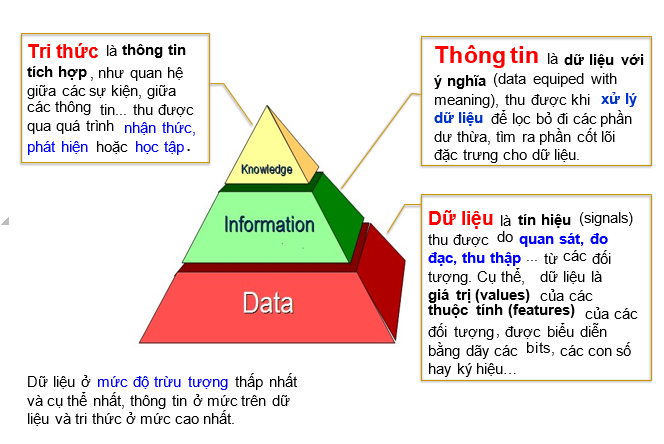
*Vậy thông tin: Độ không xác định, chưa biết (đột ngột) của dữ liệu*

Độ đo thông tin: có: là gì?

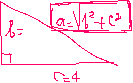
Entropy: Ký hiệu *E=-p\*log2p* (1)

*p: (Probability*: xác suất); *p=2/6* (gieo xúc xắc)=m/n (gia trị m=2 xuất hiện/tổng giá tri n=6). Trong ví dụ 2 ngày, trong đó có 1 ngày chưa biết trong tổng số 2 ngày thì xaxc suất để co thông tin: *p=1/2*.

Thay vào công thức Entropy: *E=-1/2\*log21/2*= -1/2\**log22-1*=*1/2\*log22*=1/2\*1=1/2=*50%.*



*Hình 1.1 Biểu diễn dữ liệu, thông tin, tri thức[2].*



c) Knowledge (tri thức): khác thông tin

Từ định nghĩa: Tri thức= Sự kiện (Events) + Luật (quy luật: Rule)

Tri thức= Sự kiện (Events:DATA) + Luật (quy luật: Rule: INFERENCE)

Ví dụ: Có các dữ liệu đo tam giác vuông: cạnh góc vuông b=4; c=3 (sự kiện)



Quy luật: a= 5 từ quy luật (Rule) trong toán học là *định lý*. (định lý Pitago): *a2=b2+c2*

quy luật (Rule) trong vật lý: *định luật*: *F=mv2*

quy luật (Rule) trong hóa học: *Quy tắc*: “Muốn tác dụng với axit tạo thành muối mới và axit mới.

quy luật (Rule) trong trí trí tuệ nhân tạo: *quy luật của tư duy*

*ví dụ luật bắc cầu, Luật modus Ponenns*

Quy luật khác: (chu vi)=2\*PI\*R ->PI=(chu vi)/2R=3.141592653789

Độ đo tri thức: IQ.

Một trong những vấn đề của học máy là rút ra được quy luật (bằng cách suy diễn).

* ***Học máy* (Machine Learning) mang hai nghĩa thông dụng**: 1) sử dụng máy tính để khám phá tri thức (Knowledge Discovery) từ dữ liệu, 2) sự học trong máy (tác tử: agent).
* Về phương diện công nghệ, trong đó nghiên cứu các kỹ thuật xây dựng và phát triển các chương trình máy tính có thể thích nghi và "học" từ các dữ liệu mẫu hoặc kinh nghiệm (experiment: ***E)***.
* **Định nghĩa**. Đến nay, đã có nhiều định nghĩa cho khái niệm học máy, tuy nhiên khó có một định nghĩa thỏa đáng được mọi người thừa nhận. Ví dụ: bức tranh vẽ cây cổ thụ 30 năm tuổi. Hỏi cây bao nhiêu tuổi: 30 tuổi. *Định nghĩa thế nào là cây 30 tuổi*? khó
* Định nghĩa sau đây phát biểu từ định nghĩa của Tom. Mitchell cho ta *cách nhìn toán học* của một chương trình học khi nghiên cứu, thiết kế. Nguyên bản tiếng Anh:

|  |
| --- |
| 1. ***Definition:*** A *computer program* is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. |

***Định nghĩa 1.1***. Một chương trình máy tính được gọi là học từ kinh nghiệm (thể hiện bằng dữ liệu) ***E*** đối với một lớp các nhiệm vụ ***T (Task)*** và thực hiện phép đo (Performance Measure) ***P*,** nếu việc thực hiện các nhiệm vụ tại nhiệm vụ ***T*** của nó (của chương trình máy tính) cũng như được đánh giá bằng phép đo ***P*** được cải thiện với kinh nghiệm ***E****.*

Theo định nghĩa này, người ta cần tối ưu hóa độ đo thực hiện P *(đo độ chính xác của các nhiệm vụ T: cụ thể nhiệm vụ dự báo và phân lớp)* dựa trên phân tích kinh nghiệm từ E để tìm cách thực hiện nhiệm vụ T tốt nhất.

* Cần có kinh nghiệm *E* (*Experience*): dữ liệu: dữ liệu dùng để học, gọi là dữ liệu học

*(Learning Data* hay dữ liệu huấn luyện: *Training Data*)

* Lớp các nhiệm vụ nào đó (*some class of Tasks T*): hai nhiệm vụ cơ bản, nhiệm vụ chính là:
* Phân lớp/nhận dạng (*Classification/Pattern Recognition (Indentification*)

Ví dụ lớp ta phân thành 2 lớp người: Nam, nữ (với đặc tính, đặc trưng riêng). Nếu chia nhỏ hơn thành từng người, mỗi người nhóm riêng, trở nên nhận dạng người (nhận dạng đối tượng, mỗi người có đặc tính riêng).

- Dự đoán (*Prediction*): Ví dụ dự đoán thời tiết. Dự đoán bạn *A* là Học sinh giỏi (nhờ các đặc trưng của điểm các môn học: “có điểm cao” là giá trị ngôn ngữ mang tính chất định tính; hoặc thể hiện bằng số như điểm 9, 10 là các giá trị bằng số: dữ liệu thuộc loại định lượng.

- *P* (*Performance Peasure P*): đánh giá: Vi dụ nhiệm vụ *T* (dự báo bạn *A* sinh viên Giỏi) đạt độ chính xác 95%. (95% chính là thực hiện phép đo để đánh giá). Hoặc sử dụng các tiêu chí do (đánh giá) khác. Mỗi một thuật toán học máy có thể có phép đo (đánh giá riêng)

Từ đó có thể tốm lược định nghĩa bằng công thức sau:



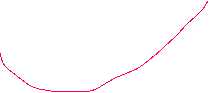
**HỌC MÁY (MACHINE LEARNING)= LEARNING PROGRAM= {E, T, P}**



Ghi chú: theo truyền thống CNTT:



**Program=Algorithms+**



Theo học máy:

**LEARNING PROGRAM= [Algorithm(for T) +Algorithm(for P)]+ Data(E)**



Sơ bộ ta biết học máy để làm gi?



***1.1.2. Tại sao cần nghiên cứu học máy?***



Sự thâm nhập mạnh mẽ của công nghệ thông tin, kinh tế, xã hội, công nghệ tri thức phát triển và tạo nên nhu cầu ứng dụng rộng rãi. Sau đây là một số phạm vi nghiên cứu, ứng dụng điển hình:

* Xây dựng các hệ nhận dạng mẫu dùng cho các thiết bị nghe nhìn cho robot và trong lĩnh vực tự động hóa, nhận dạng chữ viết tay, chuyển đổi các bài nói thành văn bản, phân tích ảnh tự động
* Tạo ra các chương trình máy tính có thể hoạt động thích nghi với môi trường thay đổi hay thực hiện các nhiệm vụ mà ban đầu chưa xác định rõ, chẳng hạn, hệ lái tự động (máy bay, ôtô, tày thủy), trò chơi hay các điều khiển robôt đa năng.



* Khám pha tri thức (Knowledge Discovery), khai phá dữ liệu (Data Mining) từ dữ liệu, đặc biệt là các cơ sở dữ liệu lớn (Big Data), để trợ giúp ra quyết định (Support Decision). Chẳng hạn, phân tích thị trường, chẩn đoán (dự báo) bệnh của bệnh nhân và xác định phương án điều trị nhờ phân tích các bệnh án lưu trữ.

***1.1.3. Một số lĩnh vực liên quan***

Trong mấy chục năm qua, các nghiên cứu khoa học và ứng dụng của học máy phát triển nhanh, kết hợp các tiến bộ của nhiều lĩnh vực khác. Sau đây là các lĩnh vực góp phần quan trọng cho nghiên cứu học máy:

* Lý thuyết xác suất và thống kê: Là tiền thân của lĩnh vực học máy, trong đó, cho phép suy luận (reasoning), suy diễn (inference) từ quan sát cụ thể để có kết luận khái quát nhờ thành tựu của giải tích ngẫu nhiên.
* Mô hình nơ ron sinh học (học ở học phần mạng nơ ron). Việc nghiên cứu cơ chế hoạt động, xử lý phi tuyến và cấu tạo hệ thần kinh sinh học nói chung cho phép tạo nên các mô hình và thuật toán phỏng sinh học, đặc biệt là các mạng nơron.
* Lý thuyết độ phức tạp tính toán (học ở Toán rời rạc). Cho phép ước lượng độ phức tạp của các nhiệm vụ học đo qua các ví dụ đào tạo, số lỗi và các thủ tục tính toán...
* Lý thuyết điều khiển thích nghi. Các thủ tục học để điều khiển quá trình nhằm tối ưu hoá mục đích định trước hay học cách đoán các trạng thái tiếp theo của quá trình điều khiển
* Tâm lý học: Cho phép mô phỏng các đáp ứng thực tế của con người, xây dựng các mô hình xử lý hiệu quả, chẳng hạn, học tăng cường.
* Các mô hình tiến hóa (chương 8). Việc nghiên cứu các mô hình tiến hóa cho phép chúng ta đưa ra các thuật toán học mô phỏng tự nhiên như: thuật toán di truyền trong CNTT (GA: Genetic Algorithm), tối ưu đàn kiến (ACO: Ant Colony Optimization), tối ưu bầy đàn (PSO: Particle Swam Optimization), hệ miễn dịch nhân tạo (AIS Artificial Immune System), thuật toán luyện thép (SA: Steel Annealing).
  1. **Các bài toán học máy**

Có nhiều bài toán chi tiết cụ thể sẽ nêu sau. Ở đây, hệ học giải quyết hai lớp nhiệm vụ***T*** chính;

* Nhiệm vụ phân lớp (*Classification)*/nhận dạng mẫu (*Pattern Recognition*)
* Nhiệm vụ dự báo (*Predication*).

#### Các bài toán học thiết lập đúng đắn

Bài toán học T được cho là thiết lập đúng khi thực sự có thể cải tiến được độ đo *P* qua kinh nghiệm E. Thông thường mô hình toán học để xây dựng thuật toán cho một bài toán học đòi hỏi phải đúng đắn theo Hadamard. Trong các bài toán thực tế, [Hadamard](http://en.wikipedia.org/wiki/Jacques_Hadamard) cho rằng một mô hình toán học ứng dụng được xem là thiết lập đúng đắn (well-posed problem) nếu nó có các tính chất:

1. Luôn tồn tại lời giải
2. Chỉ có duy nhất một lời giải
3. Khi các điều kiện ban đầu thay đổi ít thì *lời giải* cũng thay đổi ít.

Tuy nhiên, trong nhiều bài toán, điều kiện duy nhất một lời giải nhiều khi khó đáp ứng. Trong trường hợp đó người ta hay dùng phương pháp chính quy hóa (hiệu chỉnh hàm mục tiêu) để bài toán trở nên thiết lập đúng đắn.

Bài toán học phải được xác đính đúng đắn dựa trên việc xác định rõ nhiệm vụ cụ thể, độ đo việc thực hiện và nguồn dữ liệu/kinh nghiệm.

Phương pháp thông dụng nhất để đưa ra thuật toán cho các bài toán học là xây dựng một mô hình toán học phụ thuộc các tham số và dùng dữ liệu hoặc kinh nghiệm đã có để xác định giá trị thích hợp cho các tham số này.

* 1. **Phân loại học máy**

Các ứng dụng của học máy rất đa dạng, sau đây điểm qua một số lớp bài toán ứng dụng thường gặp và chia làm ba loại

* + 1. ***Học có giám sát (Surpervised Learning)***

gồm các bài toán

*Học các kết hợp hay học Bayes, hay học thống kê xác suất*

Trong nghiên cứu thị trường, người ta thường quan tâm tới các sự kiện X và Y cùng xảy ra và ước lượng xác suất có điều kiện P(Y/X) để Y xảy ra với điều kiện X xảy ra. Công việc này gọi là *học các kết hợp*. Chẳng hạn, trong ví dụ 1 mục trước, nhà cung cấp cần phân tích giỏ hàng của khách hàng qua các hóa đơn để tìm xác suất P(Y/X) để nếu khách mua sản phẩm X thì cũng mua sản phẩm Y, nhờ đó người ta có thể dự đoán được khả năng một khách hàng khi mua sản phẩm X thì sẽ mua sản phẩm Y.

*Phân loại/nhận dạng mẫu*.

Các đối tượng thuộc tập ***X*** được phân thành *k* lớp dựa trên một tập con ***D*** đã biết nhãn. Chẳng hạn, các chữ số viết tay có 10 lớp, còn bài toán đối sánh vân tay thuộc loại hai lớp: trùng với ảnh lưu trữ hay không. Bài toán phân loại thuộc về học có giám sát và là bài toán thường gặp nhất trong ứng dụng.

Nhiều khi, người ta dùng từ *phân lớp* (Classification) để thay cho *phân loại* (categorize), mặc dù thuật ngữ phân lớp có nghĩa rộng hơn, bao gồm cả *phân cụm* (Cluster). Về sau, khi không gây nên nhầm lẫn, hai từ này có thể dùng thay cho nhau. Một ứng dụng quan trọng của bài toán này là phân tích hồ sơ người vay để đánh giá rủi ro trong hoạt động tín dụng (cho vay lấy lãi), trong đó dựa trên các yếu tố đặc trưng về khả năng tài chính của người vay, ngân hàng cần đoán nhận xem khách hàng có khả năng trả nợ đúng hạn không để cho vay.

*Hồi quy hàm số (dự báo định lượng)*

Hồi quy (Regession) theo Tom Mitchell [13] (Chương 8, mục 8.2.3), Hồi quy có nghĩa là tính toán xấp xỉ (tính gần đúng) hàm đích với giá trị thực nào đó (“Regression means approximating a real-valued target function”). Trong thực tiễn, ta thường phải xác định giá trị hàm số tại những điểm chưa biết dựa trên giá trị hàm đã biết tại một số điểm. Bài toán này phát biểu như sau. Có một hàm chưa biết, nhưng biết được tậptrong gồm **N** đối tượng quan sát được:

y= (1.1)

vọng bằng không)

,

trong đó là nhiễu trắng (các biến ngẫu nhiên độc lập, cùng phân bố và có kỳ

. Ta cần tìm hàm gần đúng của

cho các đối tượng khác của ***X.*** Hàm *g* sẽ được gọi là hàm hồi quy của *f*. Nếu không quan tâm tới phân bố nhiễu thì ta gọi là bài toán xấp xỉ hàm. Khi các phân bố rộng trên tập X và đòi hỏi:

***g*(** (1.2)

)

=

thì bài toán xấp xỉ này gọi là bài toán nội suy và hàm *g* sẽ được gọi là hàm nội suy của hàm *f*.

* + 1. ***Học không giám sát (Unsurpervised Learning)***

Các bài toán trên thuộc loại học có giám sát, trong đó ta biết được nhãn của tập dữ liệu quan sát được. Trong học không giám sát, ta chỉ đơn thuần phân tích đặc điểm của tập dữ liệu đế có thông tin. Ba bài toán học không có giám sát thường gặp là: ước lượng hàm mật độ, phân cụm dữ liệu và dóng hàng (align) dựa trên cấu trúc.

Trong bài toán ước lượng hàm mật độ, có một tập mẫu dữ liệu lấy ngẫu nhiên cùng phân bố, ta cần dựa trên đó để ước lượng hàm mật độ của phân bố này.

Trong bài toán phân cụm dữ liệu, người ta chia tập dữ liệu thành các tập con (cụm) sao cho các phần từ trong cùng một cụm thì *giống nhau* hơn các phần tử khác cụm. Đặc tính *giống nhau* này thường được xác định bởi *khoảng cách*, đối tượng A giống đối tượng B hơn đối tượng C nếu khoảng cách từ A đến B nhỏ hơn khoảng cách từ A đến C. Khi tập dữ liệu cần xử lý lớn thì việc phân cụm cho phép ta giảm thời gian chạy của các ứng dụng. Tuy nhiên bài toán này là bài toán thiết lập không đúng đắn (ill-posed) và thường không duy nhất nghiệm.

Phân tích các dữ liệu có cấu trúc xâu/ trình tự (string/sequence) hoặc mạng dẫn đến các bài toán dóng hàng trong xử lý ngôn ngữ tự nhiên và tin sinh học. Việc dóng hàng các trình tự DNA, RNA, Protein và các mạng tương tác protein cho phép hiểu được các tính tương đồng và khác biệt về nhiều đặc điểm sinh học của các cá thể sinh vật và loài.

* + 1. ***Học tăng cường (Increment Learning)***

*gần như lai của (Surpervised Learning+ UnSurpervised Learning)*

Trong nhiều trường hợp, đầu ra của hệ thống là một chuỗi tác động. Khi đó mỗi tác động riêng lẻ không quan trọng mà điều quan trọng là chuỗi tác động này cần đạt được mục đích định trước. Chẳng hạn, trong các trò chơi, một nước đi không thực sự quan trọng mà quan trọng là chuỗi nước đi đưa đến kết quả thắng, ví dụ 3 nêu ở trên là trường hợp riêng của loại này.

Tương tự như phương thức học nhờ trải nghiệm cuộc sống, người ta có thể tạo ngẫu nhiên nhiều lời giải chấp nhận được và sau mỗi lần lặp điều chỉnh trọng số định hướng lựa chọn tác động để càng về sau chuỗi tác động có trọng số cao giúp ta đạt được mục đích cần có.

Bài toán học tăng cường sẽ khó hơn với các bài toán chỉ quan sát được từng phần hoặc cần hợp tác của nhiều tác tử (agent) để đạt được đích. Ba phần tiếp theo được trình bày theo phân loại.

* 1. **Các công cụ và phần mềm dùng cho Học máy**
* MS-Excel
* Phần mềm WEKA, Ngôn ngữ R, Python
* Matlab
* Các phần mềm và Thư viện trong Cloud Computing

(sẽ nói rõ trong các chương có liên quan)

* 1. **Tóm tắt, Tổng kết chương 1**
* Bài toán (Tasks: *T*): - Classification/Pattern Recognition
* Prediction
* Hai hệ thống học:

Machine Learning (ML)

Modern LM

(Deep Learning Neural Networks)

Traditional LM

*(Hình 1.5 1)*

**CÂU HỎI VÀ BÀI TẬP CHƯƠNG 1**

1. Khái niệm học
2. Định nghĩa học máy, Giải thích thế nào là E, T, P?
3. Cho Cơ sở dữ liệu dưới đây, xác định E, T trong đó T: dự báo

***PHẦN II. HỌC GIÁM SÁT VÀ ỨNG DỤNG***

**CHƯƠNG 2: K-LÁNG GIỀNG GẦN NHẤT (K-NN)**

Trong các phương pháp học, K-NN là phương pháp học có tín hiệu chỉ đạo đơn giản nhất. K-NN tìm giá trị xấp xỉ (hay nhãn, đầu ra) của hàm đích của một mẫu mới đang được xem xét dựa trên *bộ dữ liệu mẫu quan sát đã được lưu trữ.* Chương này, đầu tiên giới thiệu bài toán và thuật toán K-NN để có khái niệm trực quan sinh động, sau đó bàn luận, phát triển để có tư duy trừu tượng tổng hợp.

## 2.1. Bài toán k-láng giềng gần nhất

*2.1.1. Mục đích của thuật toán*

K-láng giềng(*K-Nearest Neighbors Algorithm: K-NN*) là thuật toán được sử dụng để phân lớp đối tượng dựa vào khoảng cách gần nhất giữa các đối tượng cần xếp lớp và tất cả các đối tượng trong dữ liệu huấn luyện và dự báo

*2.1.2. Bài toán*

***Input của bài toán (Giả thiết)***

Giả sử cho cơ sở dữ liệu thực nghiệm (**E)** như bảng 2.1, với các đầu vào gồm: thuộc tính X1: huyết áp, thuộc tính X2: nồng độ Cholesterol trong máu và đầu ra Y thể hiện  tình trạng bệnh tim của bệnh nhân. Y có hai giá trị là dương tính (+:có bệnh tim) và âm tính (-: không có bệnh). Dữ liệu có thể có sẵn từ 20 bệnh nhân khám là (Training/Learning Data): dữ liệu chỉ mang tính minh họa cho thuật toán K-NN cho nên không lấy nhiều, trong thực tế dữ liệu cần lớn hơn).

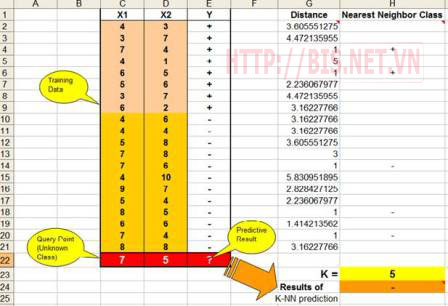
# Bảng 2.1 – Tập dữ liệu minh họa thuật toán k láng giềng

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | STT | X1 | X2 | Y | D (Euclid) | | *1* | 2 | 3 | + | 4 | | *2* | 6 | 4 | - | 5 | | *3* | 7 | 4 | + | 5.83 | | *4* | 4 | 1 | + | 6.32 | | *5* | 9 | 5 | - | 7.28 | | *6* | 6 | 6 | + | 4.12 | | *7* | 5 | 9 | + | 3.60 | | *8* | 3 | 4 | - | 3.16 | | *9* | 4 | 6 | + | 2.23 | | *10* | 2 | 9 | - | 2 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | STT | X1 | X2 | Y | D (Euclid) | | *11* | 8 | 4 | - | 6.70 | | *12* | 5 | 3 | + | 5 | | *13* | 7 | 6 | + | 5.09 | | *14* | 5 | 10 | - | 4.24 | | *15* | 10 | 7 | + | 8 | | *16* | 7 | 4 | - | 5.83 | | *17* | 8 | 5 | + | 6.32 | | *18* | 6 | 6 | - | 4.12 | | *19* | 7 | 4 | + | 5.83 | | *20* | 6 | 8 | - | 4.12 | | *21* | *5* | *7* | *?* |  | |

Cho **k=5: K** là số mẫu được chọn từ ***E*** có khoảng cách ngắn nhất với để đối sánh.

***Output (Kết luận của dự đoán)***. Bây giờ, có một bệnh nhân đến khám bệnh (chưa biết có bị bệnh tim hay không?), sau khi đo huyết áp và nồng độ Cholesterol có giá trị lần lượt là ***X01= 7 và X02= 5***. Đây là dữ liệu thử (Test): Testing Data. Cho **k=5: K** là số mẫu được chọn từ ***E*** có khoảng cách gần (ngắn) nhất với để đối sánh

Sử dụng thuật toán K-NN để dự đoán (dựa trên dữ kiệu đã phân lớp) bệnh nhân này có mắc bệnh tim hay không (tức là thuộc lớp dương tính).





*Hinh 2.1. MS-Excel cho K-NN*

***Phân tích Bảng dữ liệu thực nghiệm:***

* **Phần vào (input) của CSDL: phần thể hiện của bệnh lý, thường gọi là đặc trưng (Input Features): 2 đặc trưng, đặc tính, hay biểu hiện (tùy loại CSDL)**

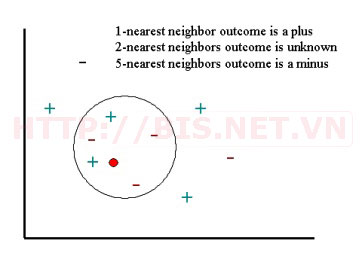
X1: huyết áp, thuộc tính X2: nồng độ Cholesterol trong máu



* **Phần ra (output) của CSDL, ký hiệu là Y, còn gọi là NHÃN (Label) và bản chất là *tín hiệu giám sát* hoặc *tín hiệu chỉ đạo* (Supervised Learning Signal). Vì có tín hiệu chỉ đạo nên thuật toán đi kèm gọi là thuật toán học có giám sát (chỉ đạo). Học có giám sát còn gọi là học có thầy (Teacher Learning Signal).**
* **Số lượng mẫu học (Num ber of Pattern Learning Examples): 20**

**Để hiểu K-NN được dùng để phân lớp thế nào và “K” là gì ta xem minh họa dưới đây.**

Trong hình dưới đây, Training Data được mô tả kết quả *y* bởi dấu (+) và dấu (-), đối tượng cần được xác định giá trị nhãn hay lớp cho nó cho điểm có tọa độ: ***xq={x01= 7 và x02= 5}*** thể hiện bằng hình tròn đỏ ở Hình 2.2. Nhiệm vụ của chúng ta là ước lượng (hay dự đoán) giá trị hay lớp của của điêm cần tìm dựa vào việc lựa chọn số láng giềng có khoảng cách gần nhất với nó. Nói cách khác chúng ta muốn biết liệu Query Point sẽ được phân vào lớp (+) hay lớp (-)





**Hình 2.2.** **Mô tả Training Data trong KNN**

**2.2. Các phương pháp tính khoảng cách**

Để xác định khoảng cách, xem xét một số phương pháp sau. Khoảng cách được tính theo từng kiểu của dữ liệu: số, nhị phân, loại.

Với dữ liệu là kiểu số:

- Khoảng cách *Minkowski*

 (2.1)

*i* = (*xi1*, *xi2*, …, *xip*) và *j* = (*xj1*, *xj2*, …, *xjp*) là 2 phần tử dữ liệu, *q* là bậc của đa thức, là số nguyên dương. *D (Distance)*

* Khoảng cách Manhattan (từ (2.1) Nếu *q* = *1*, (*d: distance)* là khoảng cách Manhattan *)*:

(2.2)

* Khoảng cách Euclidean (từ (2.1) Nếu *q* = *2*, *d* là khoảng cách Euclid.:

(2.3)

**2.3. Thuật toán K-NN giải trên MS-Excel**

Giải thuật K-NN được Fix và Hoges [1] đề xuất từ những năm 1952 là một trong những thật toán đơn gian nhưng hiệu quả, được xếp trong 10 thuật quan trọng nhất mà khai phá dữ liệu bình chọn [2]

Trình tự các bước trong thuật toán *k*-NN.:

***Input***

***Output***

B1. Xác định *k* láng giềng gần nhất, *k=5* (K số lượng mẫu có khoảng cách gần điểm cần tìm *xq (Query Point). k* nên chọn số lẻ.

B2. Tính khoảng cách giữa các đối tượng hay điểm hỏi (*Query Point)* cần tìm cần phân lớp với tất cả các đối tượng trong dữ liệu huấn luyện (thường sử dụng khoảng cách Euclid)

B3. Sắp xếp khoảng cách theo thứ tự tăng hoặc giảm dần và xác định *k* láng giềng gần nhất với đối tượng cần phân lớp

B4. Lấy tất cả các lớp của *k* láng giềng gần nhất đã xác định

B5. Dựa vào phần lớn lớp của láng giềng gần nhất để xác định lớp cho các đối tượng cần phân lớp theo quy tắc suy diên số lớn

Trong ví dụ trên, chọn tham số *k=5* có nghĩa là lấy 5 bệnh nhân có huyết áp và nồng độ Cholesterol gần giống nhất (láng giềng gần nhất) với bệnh cần chuẩn đoán và ta thấy rằng trong 5 bệnh nhân gần nhất đó có 3 người không mắc bệnh tim (giá trị Y là -) và 2 người mắc bệnh tim (giá trị Y là +). Vì vậy theo K-NN ta xếp bệnh nhân cần chuẩn đoán bệnh vào lớp – (không bị bệnh tim)

**Giải thích công thức trong Excel**

*Công thức tính khoảng cách (Euclidean) tại ô G2 là:*

**=SQRT((C2-$C$22)^2+(D2-$D$22)^2)**

*Xác định số lượng láng giềng gần nhất (ô H23):* **K= 5**

*Chọn ra  K láng giềng gần nhất của Query point (công thức tại ô H2) là:*

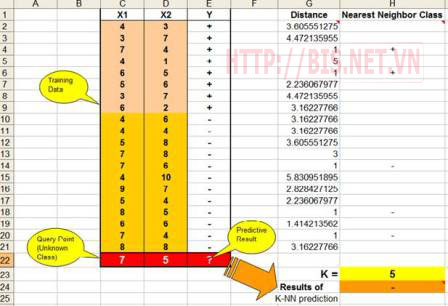
**=IF(G2<=SMALL($G$2:$G$21,$H$23),E2,"")**

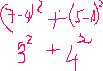
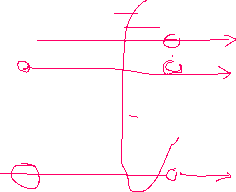
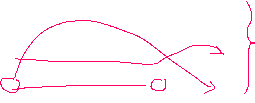
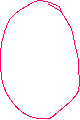
*Dự đoán lớp của Query Point (ô H24). Trong ví dụ này lớp của Query point (7,5) được xác định là* ***–*** *vì trong 5 láng giềng gần nhất với nó có 3 láng giềng thuộc lớp – (nhiều hơn số láng giềng thuộc lớp +)*

**=IF(COUNTIF(H2:H21,"+")>COUNTIF(H2:H21,"-"),"+","-")**

**Chú ý:** có thể thay đổi dữ liệu về số láng giềng K ở ô H23 cũng như thay đổi training data hay điểm tìm kiếm: Query Point): *(x1, x2 )=($c$22, $d$22)=(7, 5)* để kiểm tra kết quả của thuật toán K-NN



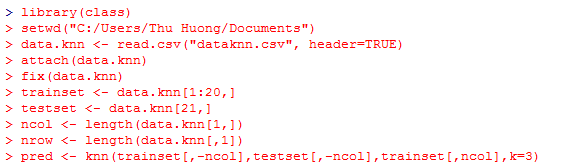




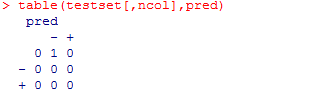
*Hinh 2.1. MS-Excel cho K-NN*

**2.4. Cài đặt với ngôn ngữ R cho ví dụ minh họa trong bảng 2.1**

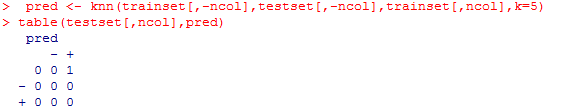




Kết quả với *k*=3: mẫu cần phân lớp có nhãn là (-)



Kết quả với *k*=5: mẫu cần phân lớp có nhãn là (+)



Kết quả trên cho thấy nhãn của mẫu cần phân lớp phụ thuộc rất nhiều vào *k.*

**2.5. Bàn luận**

***2.5.1. Đặc điểm và các tên gọi của thuật toán K-NN***

*Đặc điểm*: Thuật toán không tạo ra mô hình học máy (Learning Model) và hàm đích (Target Function) nên thuật toán xếp loại đơn giản (Simple Learning) hay học lười (Lazy Learning). Phương pháp học này dựa trên các mẫu huấn luyện/mẫu học hay các ví dụ (examples, Instances, Cases) nên thuận toán còn có tên gọi khác là (Instance-Based-Learning) hay (Case-Based-Learning). Dữ liệu thực nghiệm (E) được thường cho trong vùng lân cận giá trị hay điểm (hay ví dụ) cần tìm xq (Query Point) nên luật học này còn gọi là học địa phương (Local Learning). Trong quá trình học, thuật toán chỉ lưu để nhớ lại (Memory-Based-Learning) các nẫu học hay huấn luyện đã được cung cấp

K nên chọn là số lẻ vì…

Phương pháp đưa ra kết luận dựa trên thủ thuật (phương pháp suy diễn (Inference) gi?

( 3- >2+ -> Đối tượng nhận đầu ra (Kết quả) (-): quy luật số lớn. Hay tổng quát hơn: Phương pháp suy điễn “Bỏ phiếu bầu cử”: Giống Tư duy con người: THÔNG MINH.

K bao nhiêu thì tốt?: K=1: Không; Tối thiểu k=3…. K lớn hơn? Hy vọng chính xã hơn

Đánh giá (P): Dùng phương pháp sai số giữa tính toán và thực tế

Độ chính xác phụ thuộc vào gi? Có phụ thuộc vào độ lớn của dữ liệu học (Learning Data)

(trong ví dụ có 20, nếu có nhiều hơn). Tuy nhiên: quá nhiều: Độ phức tạp tính toán cao.

Dữ liệu của ta gồm 2 phần: Số mẫu của Dữ liệu học (Learning Data , ở ví dụ này 20) hay dữ liệu huấn luyện (Trainig Data); và số mẫu dữ liệu: thử (Testing data ở đây 2). Tỷ lệ: 2/20=10/100=10%

Bài tập về nhà: Bài toán như trên: đổi lại X01= 6 và X02= 4); k=7; khoảng cách Ơ clid.

Số đầu vào đặc trung của ví dụ là X1, x2. Tăng số đầu tức là tăng dssu hiệu thể hiện bệnh tim, thì chẩn đoán có thể chính xác hơn. Phương pháp tính dung khoảng cách tương tự.



**CHƯƠNG 3: HỌC CÂY QUYẾT ĐỊNH**



* 1. **Giới thiệu và các thuật toán học cây quyết định cơ bản**

Các thuật toán chương này tạo thnh một họ hay một lớp các thuật toán

* Thuật toán Quinlan; Thuật toán độ lộn xôn, hay dựa trên Entropy (luật học xác suất); Thuật toán ID3



* Thuật toán C4.5, C5.0, J48 (Cải biên của ID3; Thuật toán ILA (Inductive (quy nap) Learning Algorithm): chia theo giá trị đầu ra (Giá trị nhãn)
  1. **Học thống kê cây quyết định-thuật toán Quinlan**
     1. ***Bài toán***



***Input:***

Giả sử cho cơ sở dữ liệu thực nghiện **E,** còn gọi là dữ liệu huấn luyên/học**:** «RÁM NẮNG » như bảng sau:



***Bảng 2.1. Bảng dữ liệu thử nghiệm E « Rám Nắng » ban đầu***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **ĐẶC TRƯNG ĐẦU VÀO** | | | | **RA** |
| **TT**  **n** | **Tên**  ***E*** | **Màu tóc**  ***A1*** | **Chiều cao**  ***A2*** | **Cân năng**  ***A3*** | **Dùng thuốc**  ***A4*** | **Kết quả:y**  (Label) *B* |
| 1 | Hoa | Đen | Tầm thước | Nhẹ | Không | Bị rám |
| 2 | Lan | Đen | Cao | Vừa phải | Có | Không |
| 3 | Xuân | Râm | Thấp | Vừa phải | Có | Không |
| 4 | Hạ | Đen | Thấp | Vừa phải | Không | Bị Rám |
| 5 | Thu | Bạc | Tầm thước | Nặng | Không | Bị rám |
| 6 | Đông | Râm | Cao | Nặng | Không | Không |
| 7 | Mơ | Râm | Tầm thước | Nặng | Không | Không |
| 8 | Đào | Đen | Thấp | Nhẹ | Có | Không |

Giả thiết: Mô hình *A = { A1, A2, A3, A4} => B*. Đây là bài toán *A=>B*; *A*: tập đặc trưng đầu vào (*d=4*) với sô mẫu (*Pattern hay Instances=8*); Giá trị của các đặc trưng dạng chữ (ngôn ngữ), mô hình học

A1: màu tóc; 3 giá trị ngôn ngữ

A2: chiều cao: 3 giá trị ngôn ngữ

A3: cân nặng: 3 giá trị ngôn ngữ

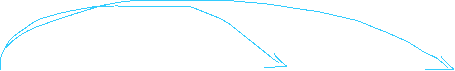
A4: dùng thuốc: 2 giá trị ngôn ngữ (nhị phân)

***Output:*** Cần xác định quy luật (Rule) để đầu ra nhận Y, N, dùng thuật toán Quinlan

* + 1. ***Phân tích dữ liệu. Chọn phương pháp: Chia để trị (Divide and Conquer)***
* Tên chương Decision Tree. *Tree=Divide*; Decision: quyết định
* , lựa chọn, nhận giá trị
* Dữ liệu đầu vào rất khó xác định đầu ra, ví dụ (xem các đặc trưng): lộn xộn, khó giải quyết vấn đề. Đề giải quyết: tìm cách chia nhỏ. Chia thế nào? Có 2 cách chia: chia theo đầu vào, chia theo đầu ra của CSDL.

**\***Chia theo đầu vào: (dung cho các thuật toán Quinlan, Độ lộn xộn, ID3, C4.5, C5.0; thuật toán Bayes… Với bảng dữ liệu thực nghiệm “Rám nắng” ta có thể chia như sau:

* Mức chia thứ nhất: lấy 4 đặc trưng đầu vào làm 4 gốc: Ai: i=1,2,3,4



**Cân nặng:A3**

|  |  |  |
| --- | --- | --- |
| *j=1 Nhẹ* | *j=2 Vừa* | *j=3 Nặng* |
| **Hoa** | **Lan** | **Thu** |
| **Đào** | **Xuân** | **Đông** |
|  | **Hạ** | **Mơ** |
|  |  |  |

**Chiều cao: A2**

|  |  |  |
| --- | --- | --- |
| *j=1 T.Thước* | *j=2 Cao* | *j=3 Thấp* |
| **Hoa** | **Lan** | **Xuân** |
| **Thu** | **Đông** | **Hạ** |
| **Mơ** |  | **Đào** |
|  |  |  |

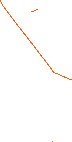
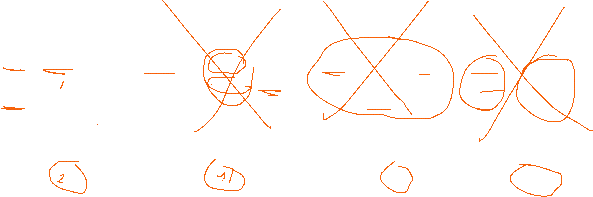
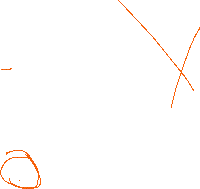
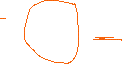
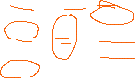
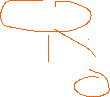
**Màu tóc: A1**

|  |  |  |
| --- | --- | --- |
| *j=1 Đen* | *j=2 Bạc* | *j=3*  *Râm* |
| **Hoa** | **Thu** | **Đông** |
| **Lan** |  | **Xuân** |
| **Hạ** |  | **Mơ** |
| **Đào** |  |  |

**Dùng thuốc: A4**

|  |  |
| --- | --- |
| *j=1*  *Có* | *j=2*  *Không* |
| **Hoa** | **Lan** |
| **Thu** | **Xuân** |
| **Hạ** | **Đào** |
| **Đông** |  |
| **Mơ** |  |

**Tỷ lệ số nhóm có giá trị đồng nhất (hoặc đồng nhất rám hoặc không) trên tổng số nhóm**



|  |  |  |  |
| --- | --- | --- | --- |
| 2/3 | 1/3 | 0/3 | 1/2 |

*Hình 3.1. Cây “RÁM NẮNG”*



* Mức 2: Chia theo giá trị (ký hiệu giá trị là *j*) và viết Tên (Mã)
* Mức 3: chia theo giá trị kết quả (lá)

*Phân tích:…*



* + 1. *Thuật toán Quinlan*



**Bước** **1:** Chia CSDL theo các đặc trưng đầu vào *A1, A2, A3, A4 (Hình 3.2)*

*Nhận thấy có thể đx viết được luật:*



**If “màu tóc” là “râm”**

**Then "kết quả’ là không rám**

Tuy nhiên, có 4 gốc, cần chọn 1 (bỏ 3 gốc còn lại), làm thế nào? *Chuyển bước 2*



**Bước 2** : Quinland đề xuất dùng véc tơ: VAi(j) = [T1(j), T2(j)]: công cụ toán để tính các sô: 2/3; 1/3; 0/3; ½.



trong đó, Ai:đặc trưng đầu vào thứ *i. i=1,2,3,4; T1(j): giá trị của véc tơ* VAi(j) với 1: “bị rám”, j Giá trị tương ứng của Ai.T2(j) số người không bị rám nắng. Trong đó: *là xác suất mô tả :*



*(3.1)*

* *VAi(j) = 1* khi và chỉ khi *Ti =1* ; còn lại khi *VAi(j) ≠1* với các giá trị khác



Tính *Ai* với *i=1,2,3,4* : 4 trường hợp như bảng sau:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A1 | J= 1 : tóc đen | VA1(j=1) = {2/4, 2/4}={0.5, 0.5} | ≠1 | 2/3 |
| J=2 : tóc bạc | *VA1(j=2) = {1/1, 0/1}={1, 0}* | =1 |
| J=3 : râm | *VA1(j=3) = {0/3, 3/3}={0, 1}* | =1 |
| A2 | J= 1 tầm thước | *V2(j=1) = {2/3, 1/3}* | ≠1 | 1/3 |
| J=2 cao | V2(j=2) = {0/2, 2/2} | =1 |
| J=3 thấp | V2(j=3) = {1/3, 2/3} | =1 |
| A3 | J= 1 nhẹ | V3(j=1) = {1/2, 1/2} | ≠1 | 0/3 |
| J=2 vừa | V3(j=2) = {1/3, 2/3} | ≠1 |
| J=3 cao | V3(j=3) = {1/3, 2/3} | ≠1 |
| A4 | J= 1 dùng thuốc | V4(j=1) = {2/5, 3/5} | ≠1 | 1/2 |
| J=2 ko dùng thuốc | V4(j=2) = {0/3, 3/3} | =1 |

**Max(.)= tỷ lệ đồng nhất. chọn A**



**BƯỚC 3**: Chọn Gốc

* Tiêu chí chọn V­i là max của tỷ lệ đồng nhất đầy ra (rám :Y, Không rám : N) để kết luận. chọn nhóm A1=2/3 : max
* Còn lại : Xóa 3 nhóm phân loại (tỉa)

**Màu tóc: A1**

|  |  |  |
| --- | --- | --- |
| *j=1 Đen* | *j=2 Bạc* | *j=3*  *Râm* |
| **Hoa** | **Thu** | **Đông** |
| **Lan** |  | **Xuân** |
| **Hạ** |  | **Mơ** |
| **Đào** |  |  |

**Tỷ lệ số nhóm có giá trị đồng nhất (hoặc đồng nhất rám hoặc không) trên tổng số nhóm**

**BƯỚC 4**: Chọn Giá trị để viết luật theo gốc đã chọn ở bước 3. Chọn giá trị nào có Viết luật cho vector *VA1(j=1)* có giá trị =1. Ta có *VA1(j=2)* ứng với luật:

Luật 1 : If ‘A1’ là (=) « Bạc »

Then ‘kết quả’ là « Rám »

VA1(j=3) ứng với luật

Luật 2 : if ‘A1’ là râm

Then ‘kết quả’ là « không rám »

*Nhận xét* : T = T1 + T2 = 1 hay tổng quát hơn n số giá trị đâu ra

trong đó n là các giá trị của đầu ra (ở ví dụ này n=2)

n= 1 : bị rám

n= 2 : không rámó 4 mẫu được đưa vào luật. Xóa 4 mẫu này khỏi CSDL. 4 mẫu còn lại chưa cho vào luật, cần phải tính tiếp. Tính thế nào ?

Đến đây chúng ta chuyển sang bướ 5 để tính tiếp.

**BƯỚC 5**: Tổ hợp chập 2 đặc trưng đầu vào, ta có 3 trường hợp sau:

Xóa nhánh A1(j=2), A1(j=3)



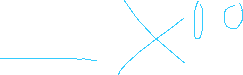
|  |  |
| --- | --- |
| J=1 đen | (a) (A1 là đen) AND (A2: chiều cao)  (b) (A1 là đen) AND (A3: cân nặng) |
| Hoa | (c) (A1 là đen) AND (A4:dùng thuốc) |
| Lan |  |
| Hạ |  |
| Đào |  |

và lặp lại các bước từ B2 đến B4.

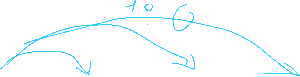
1. V(A1 là đen ^A2) (j =1 cao) =



V(A1 là đen ^A2) (j =2 thấp) =



V(A1 là đen ^A2) (j =1 tầm thước) =



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên**  **A :** | **Màu tóc**  **{A1** | **Chiều cao**  **A2** | **Cân năng**  **A3** | **Dùng thuốc**  **A4}** | **Kết quả**  (Label) B |
| 1 | Hoa | Đen | Tầm thước | Nhẹ | Không (N) | Bị rám (y) |
| 2 | Lan | Đen | Cao | Vừa phải | Có (Y) | Không (N) |
| 4 | Hạ | Đen | Thấp | Vừa phải | Không | Bị Rám |
| 8 | Đào | Đen | Thấp | Nhẹ | Có | Không |

***Bảng 2.1. Bảng dữ liệu thử nghiệm E « Rám Nắng » cò lại sau vòng 1***



1. V(A1 là đen ^A3) (j =1 nhẹ) =



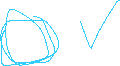
V(A1 là đen ^A3) (j =1 vừa phải) =



1. V(A1 là đen ^A4) (j =1 không) =



V(A1 là đen ^A4) (j =1 Có) =



Chọn nhóm (c) và viết 2 luật tiếp theo:

Luật 3 : If (‘A1’ là ‘đen’) ʌ (A4 là ‘không’)



Then ‘kết quả’ là ‘rám nắng’



Luật 4: If (A1 là ‘đen’) ʌ (A4 là ‘có’)

Then ‘kết quả’ là ‘không rám’



Trong trường cả tình huống đều băng 1, chọn ngẫu nhiên một trong ba trường hợp đó để viết luật. Kết thúc vòng này, ta thấy không còn dữ liệu nào chưa vào luật. kết thúc thuật toán.

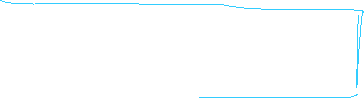
**B6.** Nếu dữ liệu vẫn chưa đưa hết vào luật, tiếp tục lấy tổ hợp chập 3, chập 4 v.v… các đặc trưng cho đến khi không còn dư liệu nào chưa tham gia vào luật. Dừng.

*Nhận xét 1.* Có thể kết hợp 4 luật thành hai luật:

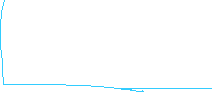
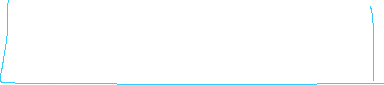
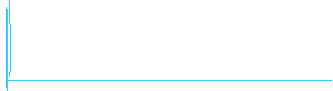
  If (A1 là “Bạc”)(A1 là ‘đen’) ʌ (A4 là ‘không’))



Then “kết quả” là “rám nắng”



If (A1 là “Râm”)( (A1 là ‘đen’) ʌ (A4 là ‘có’))



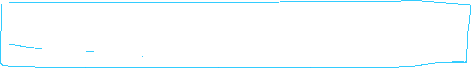
Then “kết quả” là “không rám”



(A1 là “Bạc”)(A1 là ‘đen’) ʌ (A4 là ‘không’)) và (A1 là “Râm”)( (A1 là ‘đen’) ʌ (A4 là ‘có’)) từ Logic mệnh đề thì đây là dạng chuẩn tắc tuyển.

1. *Thay luật bằng hàm*

Nếu ký hiệu giá trị *“Rám”=1; “Không Rám”=0* và hàm đích (Target Function) hay hàm đích (Goal Function: *g(x)*) thì đây chính là hàm bước nhảy đơn vị:



Chú ý rằng hàm *g(x)* là hàm xấp xỉ (gần đúng), hay có một số tài liệu còn gọi là hàm *h(x): (Hypothesics: giả thuyết),* có tài liệu khác ký hiệu là hàm *c(x) (Colclusion: kết quả, kết luận)* tùy theo ngữ cảnh, và bài toán.

Điều này tương đương với hàm tương tác đầu ra trong học của mạng nơ ron nhân tao.

Bài tập; Làm lại 2 thuật toán, viết tay, nộp lớp trưởng Ly ngày thư 5, lúc 20h00, 21h00 Ly chuyển cho thầy (15+6=21/4/2022)



* 1. **Thuật toán độ** **lộn xộn**

Thuật toán độ lộn xộn được Qinlan đề xuất

* + 1. *Bài toán*

***Input:*** Để dễ so sánh, ta sử dụng cơ sở dữ liệu «RÁM NẮNG » như bảng 2.1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên**  ***X*** | **Màu tóc**  ***A1*** | **Chiều cao**  ***A2*** | **Cân năng**  ***A3*** | **Dùng thuốc**  ***A4*** | **Kết quả:y**  (Label) *B* |
| 1 | Hoa | Đen | Tầm thước | Nhẹ | Không | Rám |
| 2 | Lan | Đen | Cao | Vừa phải | Có | Không |
| 3 | Xuân | Râm | Thấp | Vừa phải | Có | Không |
| 4 | Hạ | Đen | Thấp | Vừa phải | Không | Rám |
| 5 | Thu | Bạc | Tầm thước | Nặng | Không | Rám |
| 6 | Đông | Râm | Cao | Nặng | Không | Không |
| 7 | Mơ | Râm | Tầm thước | Nặng | Không | Không |
| 8 | Đào | Đen | Thấp | Nhẹ | Có | Không |

***Bảng 2.1. Bảng dữ liệu thử nghiệm E «Rám Nắng» ban đầu***

Giả thiết: *A = { A1, A2, A3, A4} => B*. Đây là bài toán *A=>B*; *A*: tập đặc trưng đầu vào (*d=4*) với sô mẫu (Pattern hay instances =8); Giá trị của các đặc trưng dạng chữ (ngôn ngữ)

A1: màu tóc; 3 giá trị ngôn ngữ

A2: chiều cao: 3 giá trị ngôn ngữ

A3: cân nặng: 3 giá trị ngôn ngữ

A4: dùng thuốc: 2 giá trị ngôn ngữ (nhị phân)

***Output:*** Cần xác định quy luật (Rule) để đầu ra nhận Y, N, dùng thuật toán độ lộn xộn

**BƯỚC 1**: Hoàn toàn tương tự như thuật toán Quinlan dung phương pháp “chia để trị (tức tạo cây)

**Cân nặng:A3**

|  |  |  |
| --- | --- | --- |
| *j=1 Nhẹ* | *j=2 Vừa* | *j=3 Nặng* |
| **Hoa** | **Lan** | **Thu** |
| **Đào** | **Xuân** | **Đông** |
|  | **Hạ** | **Mơ** |
|  |  |  |

**Chiều cao: A2**

|  |  |  |
| --- | --- | --- |
| *j=1 T.Thước* | *j=2 Cao* | *j=3 Thấp* |
| **Hoa** | **Lan** | **Xuân** |
| **Thu** | **Đông** | **Hạ** |
| **Mơ** |  | **Đào** |
|  |  |  |

**Màu tóc: A1**

|  |  |  |
| --- | --- | --- |
| *j=1 Đen* | *j=2 Bạc* | *j=3*  *Râm* |
| **Hoa** | **Thu** | **Đông** |
| **Lan** |  | **Xuân** |
| **Hạ** |  | **Mơ** |
| **Đào** |  |  |

**Dùng thuốc: A4**

|  |  |
| --- | --- |
| *c=1*  *Rám* | *c=2*  *Không* |
| **Hoa** | **Lan** |
| **Thu** | **Xuân** |
| **Hạ** | **Đào** |
| **Đông** |  |
| **Mơ** |  |

**Tỷ lệ số nhóm có giá trị đồng nhất (hoặc đồng nhất rám hoặc không) trên tổng số nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| 2/3 | 1/3 | 0/3 | 1/2 |

**BƯỚC 2**: Tính toán theo độ lộn xộn dùng Entropy

trong đó:

*(3.2)*

*nj*: số mẫu nhánh có đặc trưng *Ai* (trong ví dụ đặc trưng *A1:màu tóc;…)*

*nt*: số tổng số mẫu quan sát *(nt=8* trong ví dụ của chúng ta*)*

*nic*: số mẫu nhánh cây thứ *i* với giá trị đầu ra *c* (*c=1:*Rám; *c=2:* Không cho ví dụ này)

Trong ví dụ này là mức độ không xác định (Entropy) của mỗi đặc trưng *Ai.* Mỗi một đặc trưng *Ai* lại có số đối tượng ứng với các giá trị *j* khác nhau. Ví dụ, đặc trưng *A1* có ba giá trị *j; j=1* (tóc Đen) có 4 đối tượng (4 người); *j=2* (tóc Bạc) có 1 đối tượng (1 người); *j=3* (tóc Râm) có 3 đối tượng (3 người). Do số đối tượng của mỗi giá trị cho mỗi giá trị *j* khác nhau, độ tác động lên *Entropy* không đồng đều nên người ta cho mỗi giá trị *j* một hệ số, (gọi là trọng số) được tính theo tỷ lệ phần trăm. Ví dụ, tổng số đối tượng của đặc trưng *A1* có *nt* =8 (100%) với *j=1* (tóc Đen) có 4 đối tượng (4 người) thì ==0,5=50%. Để ý rằng: =1=100%.

Bây giờ ta tính *Entropy* cho bốn đặc trưng của ví dụ này, như sau:

với *Ai=A1:*

*(3.3)*

Để ý rằng, Entropi trong công thức này có hai số hạng bằng 0 ứng với giá trị *j=2* (tóc Bạc) và *j-3* (tóc Râm): độ đột ngột của thông tin là không tức là không đột ngột (nghia là đã rõ, là đồng nhất Rám và đồng nhất Không: xem trên cây Hình

1. với *Ai=A2*:
2. với *Ai=A3*:

**Cân nặng:A3**

|  |  |  |
| --- | --- | --- |
| *j=1 Nhẹ* | *j=2 Vừa* | *j=3 Nng* |
| **Hoa** | **Lan** | **Thu** |
| **Đào** | **Xuân** | **Đông** |
|  | **Hạ** | **Mơ** |
|  |  |  |

**Chiều cao: A2**

|  |  |  |
| --- | --- | --- |
| *j=1 T.Thước* | *j=2 Cao* | *j=3 Thấp* |
| **Hoa** | **Lan** | **Xuân** |
| **Thu** | **Đông** | **Hạ** |
| **Mơ** |  | **Đào** |
|  |  |  |

**Màu tóc: A1**

|  |  |  |
| --- | --- | --- |
| *j=1 Đen* | *j=2 Bạc* | *j=3*  *Râm* |
| **Hoa** | **Thu** | **Đông** |
| **Lan** |  | **Xuân** |
| **Hạ** |  | **Mơ** |
| **Đào** |  |  |

**Dùng thuốc: A4**

|  |  |
| --- | --- |
| *c=1*  *Rám* | *c=2*  *Không* |
| **Hoa** | **Lan** |
| **Thu** | **Xuân** |
| **Hạ** | **Đào** |
| **Đông** |  |
| **Mơ** |  |

**Tỷ lệ số nhóm có giá trị đồng nhất (hoặc đồng nhất rám hoặc không) trên tổng số nhóm**

1. với *Ai=A4*:

Độ lộn xộn

|  |  |  |
| --- | --- | --- |
| TT đặc trưng | Đặc trưng *Ai* | Độ lộn xộn |
| *1* | *A1: Màu tóc* | *0,50: Min* |
| 2 | A2: Chiều cao | 0,69 |
| 3 | A3: Cân nặng | 0,94 |
| 4 | A4: Dùng thuốc | 0,64 |

*Xóa 3 gốc: Chiều cao, Cân năng, Dùng thuốc***.** *Giữ lại gốc “Màu tóc”*

**BƯỚC 3:** Chọn độ lộn xộn nhỏ nhất EA1 = 0.50

**BƯỚC 4:** Viết luật, dựa vào E=0 ở EA1

Luật 1 : If ‘ A1’ là bạc

Then ‘kết quả’ là rám

Luật 2 : if ‘A1’ là râm

Then ‘kết quả’ là không rám

**BƯỚC 5**: Tổ hợp chập 2 đặc trưng đầu vào, ta có 3 trường hợp sau:

Xóa nhánh A1(j=2), A1(j=3)

|  |  |
| --- | --- |
| J=1 đen | (a) (A1 là đen) AND (A2: chiều cao)  (b) (A1 là đen) AND (A3: cân nặng) |
| Hoa | (c) (A1 là đen) AND (A4:dùng thuốc) |
| Lan |  |
| Hạ |  |
| Đào |  |

***Bảng 2.1. Bảng dữ liệu thử nghiệm E « Rám Nắng » cò lại sau vòng 1***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên**  **A :** | **Màu tóc**  **{A1** | **Chiều cao**  **A2** | **Cân năng**  **A3** | **Dùng thuốc**  **A4}** | **Kết quả**  (Label) B |
| 1 | Hoa | Đen | Tầm thước | Nhẹ | Không (N) | Bị rám (y) |
| 2 | Lan | Đen | Cao | Vừa phải | Có (Y) | Không (N) |
| 4 | Hạ | Đen | Thấp | Vừa phải | Không | Bị Rám |
| 8 | Đào | Đen | Thấp | Nhẹ | Có | Không |

và lặp lại các bước từ B2 đến B4.

Chọn E= 0

Luật 3 : If (A1 là ‘đen’) ʌ (A4 là ‘không’

Then “kết quả” là “rám nắng”

Luật 4: If (A1 là ‘đen’) ʌ (A4 là ‘có’)

Then “kết quả” là “không rám”

*Nhận xét 1.* Có thể kết hợp 4 luật thành hai luật:

  If (A1 là “Bạc”)(A1 là ‘đen’) ʌ (A4 là ‘không’))

Then “kết quả” là “rám nắng”

If (A1 là “Râm”)( (A1 là ‘đen’) ʌ (A4 là ‘có’))

Then “kết quả” là “không rám”

(A1 là “Bạc”)(A1 là ‘đen’) ʌ (A4 là ‘không’)) và (A1 là “Râm”)( (A1 là ‘đen’) ʌ (A4 là ‘có’)) từ Logic mệnh đề thì đây là dạng chuẩn tắc tuyển.

1. *Thay luật bằng hàm*

Nếu ký hiệu giá trị *“Rám”=1; “Không Rám”=0* và hàm đích (Target Function) hay hàm cung thành (gôn), (Goal Function *g(x)*) thì đây chính là hàm bước nhảy đơn vị:

Điều này tương đương với hàm tương tác đầu ra trong học của mạng nơ ron nhân tao.

***Ví dụ luyện tập:* 1. Cho Bảng 5.1. Tập dữ liệu thời tiết. Tìm quy luật hay hàm mục tiêu (hàm đích)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TT | Out-look | Temp | Humidity | Windy | Play |
| 1 | Sunny | Hot | High | False | No |
| 2 | Sunny | Hot | High | True | No |
| 3 | Overcast | Hot | High | False | Yes |
| 4 | Rainy | Mild | High | False | Yes |
| 5 | Rainy | Cool | Normal | False | Yes |
| 6 | Rainy | Cool | Normal | True | No |
| 7 | Overcast | Cool | Normal | True | Yes |
| 8 | Sunny | Mild | High | False | No |
| 9 | Sunny | Cool | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | False | Yes |
| 11 | Sunny | Mild | Normal | True | Yes |
| 12 | Overcast | Mild | High | True | Yes |
| 13 | Overcast | Hot | Normal | False | Yes |
| 14 | Rainy | Mild | High | True | No |

# Bảng 2.4 – Bảng cơ sở dữ liệu thời tiết

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ngày** | **Trời** | **Nhiệt độ** | **Độ ẩm** | **Gió** | **Chơi tennis** |
| D1 | Nắng | Cao | Cao | Yếu | Không |
| D2 | Nắng | Cao | Cao | Mạnh | Không |
| D3 | U ám | Cao | Cao | Yếu | Có |
| D4 | Mưa | Trung bình | Cao | Yếu | Có |
| D5 | Mưa | Thấp | Bình thường | Yếu | Có |
| D6 | Mưa | Thấp | Bình thường | Mạnh | Không |
| D7 | U ám | Thấp | Bình thường | Mạnh | Có |
| D8 | Nắng | Trung bình | Cao | Yếu | Không |
| D9 | Nắng | Thấp | Bình thường | Yếu | Có |
| D10 | Mưa | Trung bình | Bình thường | Yếu | Có |
| D11 | Nắng | Trung bình | Bình thường | Mạnh | Có |
| D12 | U ám | Trung bình | Cao | Mạnh | Có |
| D13 | U ám | Cao | Bình thường | Yếu | Có |
| D14 | Mưa | Trung bình | Cao | Mạnh | Không |

* + 1. ***Dùng phần mềm WEKA để tính. Cài đặt Phần mềm***
  1. **Thuật toán ID3**
     1. ***Mô hình và thuật toán ID3***

Thuật toán ID3 (Inductive Dichotomizer 3): Nghĩa đen: tách đôi quy nạp) do Qinlan đề xuất vào khoảng năm 1991, là cải tiến của thuật toán độ lộn xộn. Lấy lại bài toán “Rám Nắng”:

***Bảng 2.1. Bảng dữ liệu thử nghiệm E « Rám Nắng » ban đầu***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên :*X*** | **Màu tóc *A1*** | **Chiều cao *A2*** | **Cân năng *A3*** | **Dùng thuốc *A4*** | **Kết quả:y** |
| 1 | Hoa | Đen | Tầm thước | Nhẹ | Không | Rám |
| 2 | Lan | Đen | Cao | Vừa phải | Có | Không |
| 3 | Xuân | Râm | Thấp | Vừa phải | Có | Không |
| 4 | Hạ | Đen | Thấp | Vừa phải | Không | Rám |
| 5 | Thu | Bạc | Tầm thước | Nặng | Không | Rám |
| 6 | Đông | Râm | Cao | Nặng | Không | Không |
| 7 | Mơ | Râm | Tầm thước | Nặng | Không | Không |
| 8 | Đào | Đen | Thấp | Nhẹ | Có | Không |

Giả thiết: *A = { A1, A2, A3, A4} => B*. Đây là bài toán *A=>B*; *A*: tập đặc trưng đầu vào (*d=4*) với sô mẫu (Pattern hay instances =8); Giá trị của các đặc trưng dạng chữ (ngôn ngữ)

A1: màu tóc; 3 giá trị ngôn ngữ

A2: chiều cao: 3 giá trị ngôn ngữ

A3: cân nặng: 3 giá trị ngôn ngữ

A4: dùng thuốc: 2 giá trị ngôn ngữ (nhị phân)

***Output:*** Cần xác định các quy luật (Rule) để hàm đầu ra (hay hàm đích *g(x)* nhận Y (Rám), N Không Rám), dùng thuật toán ID3

**BƯỚC 1**: Hoàn toàn tương tự như thuật toán Quinlan dùng phương pháp “chia để trị (tức tạo cây quyết định, hay cây định danh)

**Cân nặng: A3**

|  |  |  |
| --- | --- | --- |
| *j=1 Nhẹ* | *j=2 Vừa* | *j=3 Nặng* |
| **Hoa** | **Lan** | **Thu** |
| **Đào** | **Xuân** | **Đông** |
|  | **Hạ** | **Mơ** |
|  |  |  |

**Chiều cao: A2**

|  |  |  |
| --- | --- | --- |
| *j=1 T.Thước* | *j=2 Cao* | *j=3 Thấp* |
| **Hoa** | **Lan** | **Xuân** |
| **Thu** | **Đông** | **Hạ** |
| **Mơ** |  | **Đào** |
|  |  |  |

**Màu tóc: A1**

|  |  |  |
| --- | --- | --- |
| *j=1 Đen* | *j=2 Bạc* | *j=3*  *Râm* |
| **Hoa** | **Thu** | **Đông** |
| **Lan** |  | **Xuân** |
| **Hạ** |  | **Mơ** |
| **Đào** |  |  |

**Dùng thuốc: A4**

|  |  |
| --- | --- |
| *c=1*  *Rám* | *c=2*  *Không* |
| **Hoa** | **Lan** |
| **Thu** | **Xuân** |
| **Hạ** | **Đào** |
| **Đông** |  |
| **Mơ** |  |

**Tỷ lệ số nhóm có giá trị đồng nhất (hoặc đồng nhất rám hoặc không) trên tổng số nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| 2/3 | 1/3 | 0/3 | 1/2 |

**BƯỚC 2**: Thay vì tính toán Entropy như trong thuật toán độ lộn xộn

(3.8)

người ta tính độ tăng (độ lợi) thông tin (Information Gain) ký hiệu IG, là chỉ số đánh giá độ tốt của thuộc tính (Atributte) hay đặc trưng (Features) trong việc phân chia tập dữ liệu thành những tâp con có đặc trưng đồng nhất. Độ tăng độ lợi thông tin được tính như sau:

(3.9)

Một số tài liệu ký hiệu *nj=.* Ở đây *Si; S*là lực lượng (số lượng) các đối tượng của đặc trưng thứ *i* (trong ví dụ *i=1..4*) và lực lượng (số lượng)các đối tượng của toàn bộ dữ liệu học (hay dữ liệu thử nghiệm).Có rất nhiều thuật toán khác nhau được sử dụng để học cây quyết định từ dữ liệu như ID3, C4.5, Quinlan, độ lộn xộn, Sprint, thuật giải ILA… Trong phần này sẽ giới thiệu thuật toán ID3. Trong công thức (3.10) *H(s)* là Entropy của toàn bộ dữ liệu thực nghiêm ***E:***

Trong đó: là xác suất của các nhãn phân loại. Trong trường hợp bài toán của ta, c=2

*nt*: số tổng số mẫu quan sát *(nt=8* trong ví dụ của chúng ta*),*

*ni*: số mẫu nhánh cây thứ *i* với giá trị đầu ra *c* (*c=1:*Rám; *c=2:* Không cho ví dụ này)

Giá trị của IG được sử dụng để lựa chọn thuộc tính tốt nhất tại mỗi nút. Thuộc tính được lựa chọn là thuộc tính có IG lớn nhất. Giả sử ta sử dụng bảng dữ liệu dự đoán người bị “rám năng” như bảng 2.4. Ta phân lớp “+” là lớp RÁM, lớp “-” là KHÔNG.

*S = [3+, 5-]; H(S) =*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Thuật toán độ lộn xộn*** | | ***Thuật toán ID3*** | |
| *1* | *EA1: Màu tóc* | *0,50 (Min)* | *H(S, EA1): Màu tóc* | *0.95-0,50=0.45 (Max)* |
| 2 | *EA2*: Chiều cao | 0,69 | *H(S, EA2):* Chiều cao | 0.95-0,69=0,36 |
| 3 | *EA3*: Cân nặng | 0,94 | *H(S, EA3):* Cân nặng | 0.95-0,94=0.01 |
| 4 | *EA4*: Dùng thuốc | 0,64 | *H(S, E41)*: Dùng thuốc | 0.95-0,64=0,31 |

Như vậy: Trong thuật toán ID3, tiêu chí chọn nút gốc là Max(IG)

**Bước 3.** Chọn gốc, chọn *A1****.***

**Bước 4.** Chọn giá trị để viết luật, một cách tương ứng Thuật toán độ lộn xộn

**Bước 5.** Còn lại 4 giá trị máu tóc=Đen chưa viết luật. Tổ hợp chập 2, 2 đặc trung như thuật toán độ lộn xộn, viết 2 luật (sinh viên tự làm tiếp theo)

*Nhận xét 1.* Có thể kết hợp 4 luật thành hai luật:

  If (A1 là “Bạc”)(A1 là ‘đen’) ʌ (A4 là ‘không’))

Then “kết quả” là “rám nắng”

If (A1 là “Râm”)( (A1 là ‘đen’) ʌ (A4 là ‘có’))

Then “kết quả” là “không rám”

(A1 là “Bạc”)(A1 là ‘đen’) ʌ (A4 là ‘không’)) và (A1 là “Râm”)( (A1 là ‘đen’) ʌ (A4 là ‘có’)) từ Logic mệnh đề thì đây là dạng chuẩn tắc tuyển.

1. *Thay luật bằng hàm*

Nếu ký hiệu giá trị *“Rám”=1; “Không Rám”=0* và hàm đích (Target Function) hay hàm cung thành (gôn), (Goal Function *g(x)*) thì đây chính là hàm bước nhảy đơn vị:

Điều này tương đương với hàm tương tác đầu ra trong học của mạng nơ ron nhân tao.

* + 1. ***Bàn luận và đánh giá độ chính xác của thuật toán***

Để đánh giá độ chính xác của hàm mục tiêu người ta dung ma trận nhầm lẫn (Confusion Matrix). Ma trận là bảng (hai chiều). Cách thức tạo bảng (ma trận): hàng: giá trị thực tế; cột: giá trị dự báo. Giả sử kết qua đau ra nhị phân *{T, F}*

Trang đó *T: Đúng là Yes, là “Ram nắng”…* còn *F: có rhwwe là Sai, NO, “Không”* tùy theo ngữ cảnh bài toán.

|  |  |  |  |
| --- | --- | --- | --- |
|  | GIÁ TRỊ DỰ BÁO | | |
| GIÁ TRỊ  THỰC TẾ |  | *T (True)* | *F (False)* |
| *T (True)* | *True Positive: TP* | *False Negative* |
| *F (False)* | *False Positive* | *True Negative: TN* |

Một số chỉ số thông dụng được dùng để đánh giá một giải thuật máy học. Giả sử để đánh giá một bộ phân loại hai lớp tạm gọi là dương và âm:  
 Số đúng dương (TP- True Positive): số phần tử dương thực tế được phân loại đúng dương.

Số sai âm (FN- False Negative): số phần tử dương thực tế được phân loại sai sang âm.

Số đúng âm (TN- True Negative): số phần tử âm thực tế được phân loại đúng âm.

Số sai dương (FP- False Positive): số phần tử âm thực tế được phân loại dương.

*TP Rate:* tỉ lệ những phần tử được phân loại lớp x mà đúng trên tổng số những phần tử thuộc lớp x. Cho biết tỉ lệ lớp x được phân loại đúng là bao nhiêu, tương tự với recall (Độ bao phủ hay độ hồi tưởng) có ý nghĩa tương tự như TP rate

|  |  |
| --- | --- |
| *TP Rate*=*Recall*= | (3.12) |

*FP Rate*: tỉ lệ những phần tử được phân loại lớp x, nhưng mà nó không thuộc lớp x (phân loại sai) chia cho tổng những phần tử không thuộc lớp x. Cho biết lớp x bị phân loại sai là bao nhiêu.

|  |  |
| --- | --- |
| *FP Rate*= | (3.13) |

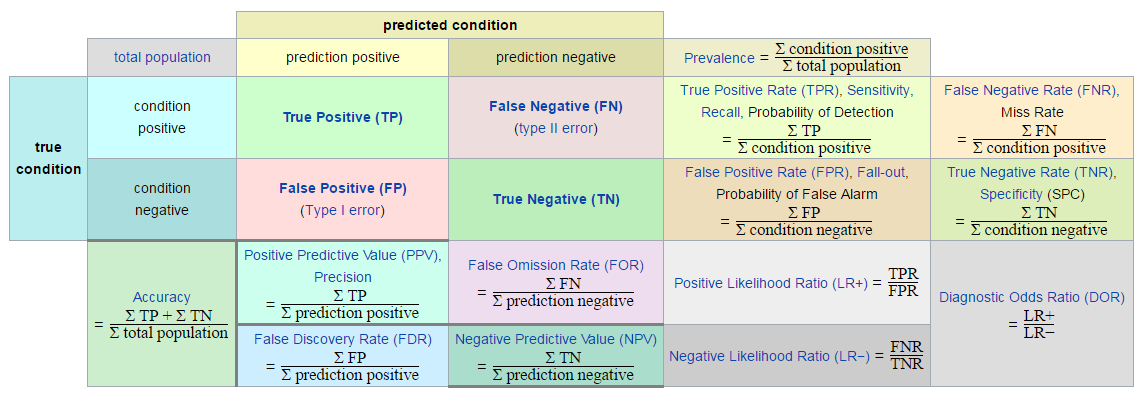
*Độ chính xác (precision):* tỉ lệ những phần tử thật sự là lớp x trên tổng số những phần tử được phân loại vào lớp x. Số kết quả chính xác chia cho số kết quả trả về.

|  |
| --- |
| *Precision* = (3.14) |

*Độ đo F1*: chỉ số cân bằng giữa độ chính xác (Precision) và độ bao phủ (Recall). Nếu độ chính xác, độ bao phủ cao và cân bằng thì độ đo F1 lớn, còn độ chính xác và độ bao phủ nhỏ và không cân bằng thì độ đo F1 nhỏ.

*Accuracy:* độ chính xác toàn bộ toàn bộ cơ sở dữ liệu

Để xem xét đầy đủ hơn ta có bảng các tiêu chí sau đây, kềm với các thuật ngữ phân loại sai số.

**

*Điều kiện dương (P): Số lượng các trường hợp thực sự tích cực trong dữ liệu*

*Điều kiện phủ định (N): Số lượng các trường hợp tiêu cực thực sự trong dữ liệu*

*True positive (TP): sự chính xác*

*Âm tính thực (TN):sự từ chối chính xác*

*Dương tính giả (FP): báo động giả, lỗi loại I*

*Sai âm tính (FN): Lỗi, loại II lỗi*

*tỷ lệ thực dương (TPR) còn được gọi là Độ nhạy (sensitivity), thu hồi (recall), tỷ lệ đạt*

|  |  |
| --- | --- |
|  | (3.1) |

*Cụ thể hoặc tỷ lệ tiêu cực thực sự (TNR)*

|  |  |
| --- | --- |
|  | (3.2) |

*Giá trị tiên đoán chính xác hoặc tích cực (PPV)*

|  |  |
| --- | --- |
|  | (3.3) |

*Giá trị tiên đoán âm (NPV)*

|  |  |
| --- | --- |
|  | (3.4) |

*Tỷ lệ bỏ lỡ hoặc tỷ lệ sai âm (FNR)*

|  |  |
| --- | --- |
|  | (3.5) |

*Rơi ra hoặc tỷ lệ dương tính giả (FPR)*

|  |  |
| --- | --- |
|  | (3.6) |

*Tỷ lệ khám phá sai (FDR)*

|  |  |
| --- | --- |
|  | (3.7) |

*Tỷ lệ bỏ sót sai (FOR)*

|  |  |
| --- | --- |
|  | (3.8) |

*Độ chính xác (ACC)*

|  |  |
| --- | --- |
|  | (3.9) |

*Điểm số F1: Là trung bình hài hòa cảu độ chính xác và độ nhạy*

|  |  |
| --- | --- |
|  | (3.10) |

*Hệ số tương quan Matthews (MCC)*

|  |  |
| --- | --- |
|  | (3.11) |

*Thông tin hoặc thông tin chính xác của người dự kiến (Bookmaker Informedness -BM)*

|  |  |
| --- | --- |
|  | (3.12) |

*Đánh dấu (MK)*

|  |  |
| --- | --- |
|  | (3.13) |

*Confusion Matrix [10] tuy nhiên, ma trận nhầm lẫn chỉ dùng cho:*

* *hai lớp phân loại*
* *các mẫu ra và mẫu dự đoán cùng số lượng*

*Đối với bài toán sử dụng mạng BAM, ma trận này chưa tượng trưng. Lý do:*

* *Đặc trưng hay số mẫu vào và ra khác nhau*
* *Không phải phân loại 2 lớp mà là nhiều lớp*

*Trên cơ sở ma trận nhầm lẫn ta phải xây dựng riêng ma trận riêng*

Một số chỉ số thông dụng được dùng để đánh giá một thuật toán máy học. Giả sử để đánh giá một bộ phân loại hai lớp tạm gọi là dương và âm:  
 Số đúng dương (TP- True positive): số phần tử dương được phân loại dương.

Số sai âm (FN- False negative): số phần tử dương được phân loại âm.

Số đúng âm (TN- True negative): số phần tử âm được phân loại âm.

Số sai dương (FP- False positive): số phần tử âm được phân loại dương.

*TP Rate:* tỉ lệ những phần tử được phân loại lớp x mà đúng trên tổng số những phần tử thuộc lớp x. Cho biết tỉ lệ lớp x được phân loại đúng là bao nhiêu, tương tự với recall.

|  |  |
| --- | --- |
| *TP Rate*= | (3.12) |

*FP Rate*: tỉ lệ những phần tử được phân loại lớp x, nhưng mà nó không thuộc lớp x (phân loại sai) chia cho tổng những phần tử không thuộc lớp x. Cho biết lớp x bị phân loại sai là bao nhiêu.

|  |  |
| --- | --- |
| *FP Rate*= | (3.13) |

Độ chính xác (precision): tỉ lệ những phần tử thật sự là lớp x trên tổng số những phần tử được phân loại vào lớp x. Số kết quả chính xác chia cho số kết quả trả về.

|  |  |
| --- | --- |
| *Precision* = | (3.14) |

Độ bao phủ (recall): có ý nghĩa tương tự như TP rate.

|  |  |
| --- | --- |
| *Recall* = | (3.15) |

Độ đo F1: chỉ số cân bằng giữa độ chính xác (precision) và độ bao phủ (recall). Nếu độ chính xác, độ bao phủ cao và cân bằng thì độ đo F1 lớn, còn độ chính xác và hồi tưởng nhỏ và không cân bằng thì độ đo F1 nhỏ.

|  |  |
| --- | --- |
| F1 = | (3.16) |

Để tính độ chính xác, chỉ cần tính tổng phần tử trên đường chéo [i,j] của ma trận confusion chia cho tổng phần tử. Tổng số phần tử không nằm trên đường chéo là lỗi dự đoán của thuật toán

Bài tập ngày 21. 1). Tìm bộ luật cho cơ sở dữ liệu “Wheather” bằng 2 thuật toán: Độ lộn xộn và thuật toán ID3.

2) Nghiên cứu và nêu ý nghĩa của các độ đo “Pcesition”, “Recall”, “Accuracy”, F1

3. Nghiên cứu và thử chậy phần mềm WEKA.

***Nhận xét:***

1. Thuật toán ID3 đơn giản nhưng có độ quá khớp (Over-Fiting) cao do việc chọn một vài cành có tỷ lệ đối tượng không đồng đều để xác định luật.
2. *Độ quá khớp (Over-Fiting*) là độ đo chính xác cao đối với dữ liệu huấn luyện, nhưng kém chính xác với những dữ liệu mới, mặc dù chỉ khác vài giá trị đặc trưng so với những dữ liệu đặc trưng đã được huấn luyện.
3. Khi nghiên cứu dữ liệu huấn luyện, tồn tại dữ liệu có thuộc tính *Ai* có nhiều giá trị *j* hơn các thuộc tính khác (Ví dụ thuộc tính *Màu tóc* có 3 giá trị; trong khi thuộc tính *Đung thuốc* chỉ có 2 giá trị) thì độ lợi thông tin *IG* tăng trên các thuộc tính nhiều giá trị phân chia (phân hoạch). Để giảm bớt độ lệch này, Quinlan đề xuất tỷ số độ lợi **.**

Tỷ số độ lợi t ính đến số lượng và độ lớn của các nhánh khi chọn một thuộc tính để chia (phân hoạch), được tính bằng độ lợi thông tin chia cho thông tin của phân phối dữ liệu trên các nhánh. Đấy chính là tử tương cải biên của Thuật toán C4,5 được Quinlan cải tiến năm 1993 từ ID3 là một trong nhiều thuật toán khác khắc phục nhược điểm này. Ta xem xét cụ thể hơn thuật toán C4.5 sau đây.

* 1. **Thuật toán C4.5**
     1. ***Thuật toán C4.5, cải biên của ID3***

Khi dữ liệu có thuộc tính có nhiều giá trị hơn các thuộc tính khác, độ lợi thông tin tăng trên các thuộc tính có nhiều giá trị phân hoạch. Để giảm bớt sự lệch này, Quinlan cũng đề nghị sử dụng tỉ số độ lợi. Tỉ số độ lợi tính đến số lượng và độ lớn của các nhánh khi chọn một thuộc tính phân hoạch, được tính bằng độ lợi thông tin chia cho thông tin của phân phối dữ liệu trên các nhánh. Giả sử khi sử dụng thuộc tính *A* phân hoạch dữ liệu *S* thành *v* phần, thông tin của phân phối dữ liệu được tính như công thức (2.4):

( 2.4)

Và tỉ số độ lợi được tính như công thức (2.5):

(2.5)

Trong mô hình phân lớp C4.5, có thể dùng một trong hai loại chỉ số *Information Gain* hay *Gain Ratio* để xác định thuộc tính tốt nhất, trong đó *Gain Ratio* là lựa chọn mặc định. Trong phần mềm WEKA, thuật toán Tree.J48 chính là thuật toán C4.5 được viết trên ngôn ngữ Java.

***3.5.2. Xử lý những giá trị thiếu trong C4.5***

Dữ liệu thiếu là giá trị của thuộc tính không xuất hiện trong một vài trường hợp có thể do lỗi trong quá trình nhập bản ghi vào cơ sở dữ liệu hoặc giá trị của thuộc tính đó được đánh giá là không cần thiết trong những trường hợp đó. Trong quá trình xây dựng cây từ tập dữ liệu đào tạo *S, B* là **test** dựa trên thuộc tính *Aa* với các giá trị đầu ra là *b1,b2,…,bt*. Tập *S0* là tập con các case trong *S* mà có giá trị thuộc tính *Aa* không biết và *Si* biểu diễn các case với đầu ra là *bi* trong test *B*. Khi đó độ đo độ lợi thông tin của test *B* giảm vì chúng ta không phân được lớp nào từ các case trong *S0* và được tính theo công thức 2.6:

(2.6)

trong đó:

*S* là tập dữ liệu huấn luyện

*B* là tập dữ liệu thử (Test)

Tập con *S0* là tập con các trường hợp (Case) trong *S* mà có giá trị thuộc tính *Aa* không biết

*Si* biễu diễn các trường hợp với đầu ra là *bi* trong *B.* Từ đó, *P(S, B)* cũng thay đổi như sau:

 ( 2.7)

Hai thay đổi này làm giảm giá trị của phép thử (Test) liên quan đến thuộc tính có tỉ lệ giá trị thiếu cao. Nếu Phép thử (Test) *B* được chọn, C4.5 không tạo một nhánh riêng trên cây quyết định cho *S0*. Thay vào đó, thuật toán có cơ chế phân chia các case trong *S0* về các tập con *Si* là tập con mà có giá trị thuộc tính Test xác định theo trọng số :

 (2.8).

***3.5.3. Ví dụ minh họa giải thuật C4.5***

***Bài toán****.* Lấy ví dụ đối với bài toán phân chia ngày phù hợp hay không phù hợp cho việc chơi tennis theo điều kiện thời tiết như :

**Bảng 2.1. Bảng dữ liệu học chơi Tennis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Day*** | ***Outlook*** | ***Temperature*** | ***Humidity*** | ***Wind*** | ***PlayTennis*** |
| Dl | Sunny | Hot | High | False | No |
| D2 | Sunny | Hot | High | True | No |
| D3 | Overcast | Hot | High | False | Yes |
| D4 | Rain | Mild | High | False | Yes |
| D5 | Rain | Cool | Normal | False | Yes |
| D6 | Rain | Cool | Normal | True | No |
| D7 | Overcast | Cool | Normal | True | Yes |
| D8 | Sunny | Mild | High | False | No |
| D9 | Sunny | Cool | Normal | False | Yes |
| Dl0 | Rain | Mild | Normal | False | Yes |
| Dl1 | Sunny | Mild | Normal | True | Yes |
| Dl2 | Overcast | Mild | High | True | Yes |
| Dl3 | Overcast | Hot | Normal | False | Yes |
| Dl4 | Rain | Mild | High | True | No |

Từ tập dữ liệu học chơi Tennis trong bảng trên, áp dụng thuật toán C4.5 để xây dựng mô hình cây quyết định có chơi Tennis hay không với tập các đối tượng thử nghiệm {***D1,..,D14***} được đưa vào và 4 thuộc tính *[Outlook, Temp, Humidity, Windy]*  với lớp (hay nhãn) “+” là lớp *Yes*, lớp “-” là lớp *No*

*Bài giải*

***B1. Đo độ hỗn loạn (Entropy) trước khi phân hoạch:***

*B2. Với thuộc tính Outlook*

*B2,1.* Values (*Outlook*) = {*Sunny, Overcast, Rainy*}







*B2.2.* Entropy của *S* đối với thuộc tính “*Outlook*” sau khi phân hoạch là:



B2.3. Độ lợi thông tin *Gain(S, Outlook* tương ứng:

*B2.4. Gain(S, Outlook)= Entropy(S) – Entropy(S, Outlook) = 0.94 – 0.69 = 0.25*

*B2.5.* Tỷ suất lợi ích Gain Ratio

✓*B3. Với thuộc tính Temp*

*B3.1* Values (*Temp*) = {*Hot,Mild, Cool*}







*B3.2.*

*B3.3*.*Gain(S, Temp)= Entropy(S) – Entropy(S, Temp) = 0.94 – 0.91 = 0.03*

*B3.4.*

*B3.5.*

✓*B4. Với thuộc tính Humidity*

*B4.1.* Values (*Humidity*) = {*High, Normal*}





*B4.2*,

*B4,3*. *Gain(S, Humidity)= Entropy(S) – Entropy(S, Humidity) = 0.94 – 0.79 = 0.15*

*B4.4.*

*B4.5.*

*B5. Với thuộc tính Windy*

*B5.1.* Values(Windy)={True, False}

*B5,2,* 

*B5.3. Gain(S,* *)= Entropy(S) –.Entropy(S,* *)=0.94 – 0.89=0.05*

*B5.4.*

*B5.5.*

**B6. Tiêu chí chọn gốc đầu tiên: *Max*(***GainRatio)*

Ta thấy từ bước 2 đến bước 5, tổng hợp lại ta có bảng sau về tỷ suất độ lợi (và 3 phương pháp)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Thuộc tính | *ĐLXộn-Entropy* | *ID3-(Information Gain)* |  |
| 1 | ***Outlook*** | *0.69 (Min)* | *0.25 (Max)* | (Max) |
| 2 | ***Temperature*** | *0.91* |  |  |
| 3 | ***Humidity*** | 0.79 | 0.15 |  |
| 4 | *Wind* | 0.89 |  |  |

Từ bảng trên, thuộc tính *Outlook* có *GainRatio = 0,16* là lớn nhất nên nó được dùng làm nút gốc để chia (phân hoạch) cây tiếp theo.



**B7.** Với gốc *Outlook*  đã được xác định trong **B6**  có ba nhánh {Rainy, Sunny, Overcast}trong đó nhánh *Outlook=Overcast* có Entropy = 0 (có nghĩa là nhánh có nhãn đồng nhất là “**yes”** nên quá trình học cây cho nhánh này dừng lại với nút lá có nhãn là “*Yes*”. Ta viết luật cho nhánh này. Như vậy, từ 1 nút lá này ta sẽ viết được 1 luật quyết định tương ứng:

***Luật 1: IF (Outlook = Overcast) THEN Play = Yes***

**B8.** *Đối với nhánh Outlook=Sunny*, quá trình học cây được tiếp tục với tập dữ liệu con tại nhánh đó, Ta có:

*B8.0.* 

1. *Với thuộc tính Temp*

*B8.1.* Values (*Temp*) = {*Hot,Mild, Cool*}











1. *Với thuộc tính Humidity*

Values *(Humidity*) = {*High, Normal*}

1. *Với thuộc tính Windy*

Values (*Windy*) = {*True, False*}





Entropy của đối với thuộc tính “*Windy*” sau khi phân hoạch là:



Vì thuộc tính *Humidity* có *GainRatio=1* nên ta chọn thuộc tính này làmnút phân hoạch cây tiếp theo. Đối với nhánh *Humidity=High* có Entropy = 0 nên quá trình học cây cho nhánh này dừng lại với nút lá có nhãn là “*No*”. Đối với nhánh *Humidity=Normal* có Entropy = 0 nên quá trình học cây cho nhánh này dừng lại với nút lá có nhãn là “*Yes*”. Như vậy, từ 2 nút lá này ta sẽ viết được 2 luật quyết định tương ứng:

***Luật 2: IF (Outlook = Sunny) AND (Humidity=High) THEN Play=No***

***Luật 3: IF (Outlook = Sunny) AND (Humidity=Normal) THEN Play=Yes***

Như vậy các lá đã được gán nhãn nên quá trình phân hoạch cây sẽ dừng lại tại nhánh này.

-Tiếp tục phân hoạch cây tiếp theo tại nhánh *Outlook=Rainy* ta có:

✓*Thuộc tính Temp*

Values (Temp) = {Mild, Cool}









✓*Thuộc tính Humidity*

Values (*Humidity*) = {*High,Normal*}











**

✓*Thuộc tính Windy*

*Values*(*Windy*)={*True, False*}









Như vậy ta chọn *Windy* có tỷ số độ lợi lớn nhất với *GainRatio=1* làm nút phân hoạch cây tiếp theo. Đối với nhánh *Windy=True* có Entropy=0 nên quá trình học cây cho nhánh này dừng lại với nút lá có nhãn là “*No*”. Đối với nhánh *Windy=False* có Entropy=0 nên quá trình học cây cho nhánh này dừng lại với nút lá có nhãn là “*Yes*”. Như vậy, từ 2 nút lá này ta sẽ viết được 2 luật tương ứng:

**Luật 4: IF *(Outlook = Rainy)* AND (*Windy=True*) THEN *Play=No***

**Luật 5: IF *(Outlook = Rainy)* AND (*Windy=False*) THEN *Play=Yes***

Các lá đã được gán nhãn nên quá trình phân hoạch cây sẽ dừng lại tại nhánh này. Như vậy đã kết thúc quá trình xây dựng cây, cuối cùng ta có được cây quyết định hoàn chỉnh như hình 2.1:

High

Normal

False

True

Rainy

Overcast

Sunny

Outlook

Humidity

Windy

Hình 3.5 : Cây quyết định cho tập **dữ liệu học chơi Tennis**



Mô hình cây quyết định được xây dựng từ việc học trên tập dữ liệu Weather để đưa ra quy luật có chơi tennis hay không. Với mô hình cây quyết định hình 2.1 ta nhận thấy cây có 5 nút lá tương ứng với 5 luật quyết định:

Luật 1: IF *(Outlook = Overcast)* THEN *Play = Yes*

Luật 2: IF *(Outlook = Sunny)* AND (*Humidity=High*) THEN *Play=No*

Luật 3: IF *(Outlook = Sunny)* AND (*Humidity=Normal*) THEN *Play=Yes*

Luật 4: IF *(Outlook = Rainy)* AND (*Windy=True*) THEN *Play=No*

Luật 5: IF *(Outlook = Rainy)* AND (*Windy=False*) THEN *Play=Yes*

-Với các dữ liệu được đưa vào từ tập dữ liệu học chơi Tennis, ta có kết quả của dữ liệu quan sát được đưa ra. Những đối tượng *chơi* Tennis là: {D3, D4, D5, D7, D9, D10, D11, D12, D13}

Những đối tượng *chơi* Tennis là: {D1, D2, D6, D8, D14}

***Remarks:***

1. Phương pháp Cây Quyết định thực hiện học máy theo 2 công đoạn (hay 2 pha)

* Pha 1: Học (learning): tìm hàm xấp xỉ cho mục tiêu(Target Function) *g(X)->Nhãn*

(thương mô tả theo bộ luật): Phân loại dựa trên Tập dữ liệu Huấn luyện/học hay các ví dụ.

* Pha2: Pha thử (Tets) DỰ ĐOÁN, PHỎNG ĐOÁN.

1. Dữ liệu thử nghiệm ứng với 2 công đoạn đó là ***E=(Data)****training* ***+****(****Data)****Test*

Có 2 casch phân chia dữ liêu cho hai công đoạn:

* Cách 1: Percentage Split (Phân chia %) ***(Data)E [100%]=(Data)****training [70%]****+****(****Data)****Test[30%]*
* Cách 2: *Cross-Validation* (Xác thực chéo): Chia (Split) ***(Data)E*** thành *k-Folds (k-tập); k=10;*
* Thực hiện Học máy *k-lần.* Mỗi lần *(k=1, 2, ...)* lấy 9 phần học, 1 phần để thử

Và tính độ chính xác (độ đo ***P***) như: Accuracy, Racall, Precission, F1,...

* Sau *k-lần như vây,* Độ chính xác tổng thể của các độ đo như: Accuracy, Racall, Precission, F1,... được lấy trung bình cộng của *k-lần*.

1. Phương pháp cây của Quinlan còn cải tiến với tên gọi C5.0 tương đương với thuật tóan CART

ở đó sử dụng chỉ số GINI, với tiêu chuẩn Min(GINI). Đọc thêm: Lê Thanh Nghị (ĐH Cần Thơ) dùng ngôn ngữ ***R.***

1. Các phương pháp nêu trên sử dụng rộng rãi, và là họ thuật toán đứng đầu trong Top 10 thuật toán được sử dụng nhiều nhất trong Học máy và Khai phá dữ liệu; 1) Quinlan (dùng phương pháp Thống kê); 2) Độ lộn xộn. 3) ID3. 4)C4.5 (dùng Entropy có gốc Xác suất); 5)C.5.0.
2. Phương pháp cây quyết (TREE) đồng nghĩa với phương chia để trị (DEVIDE & CONQUER)

Có hai cách chia dữ liệu huấn luyện: cách chia theo đầu vào (ví dụ như phương pháp cây quyết định chia thành các đặc trưng (gọi là gốc). Cách chia thứ 2: cách chia theo giá trị đầu ra (gọi là nhãn. Trong bài toán “Weather” có 2 giá trị đầu ra gồm *{****yes, no}*** hay còn gọi là *NHÃN* (Label). Cây quyết định, thì Quyết định ở đây đúng nghĩa là: *“có (yes) đi chơi Tennis không?”.* Tuy nhiên, trong các bài toán và ngữ cảnh khác thì ví dụ trong dữ liệu “Có rám hay không”, do đó Quyết định ở đây được hiểu khác nghĩa ví dụ bị ảnh hưởng, tác động...

Thuật toán mà cây được chia theo các giá trị đầu ra (hay nhãn) được giới thiệu sau đây và dược gọi *là thuật tóan ILA.*

* 1. **Thuật toán học cây quyết định quy nạp ILA**

***ILA? (****Inductive (quy nạp) Learning Algorithm:* Thuật toán này cũng sử dụng phương pháp Devide & Conquer. Khác với các thuật toán cây quyết định đã học: Chia (Devide) theo nhãn (giá trị đầu ra).

* + 1. ***Bài toán***

*Input: Cho cơ sở dữ liệu “Weather”*

*Output: Tìm (học) hàm (thể hiện qua các luật (Rules) xấp xỉ hàm đích (Target Function)*

* + 1. ***Thuật toán ILA***

**Bước 1**: Chia (phân hoạch) theo đặc trưng đầu ra thành hai bảng như dưới đây

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ***Đổi TT mới*** | ***Outlook*** | ***Temp*** | ***Humidity*** | ***Windy*** | ***Play*** | | D1 | Overcast | Hot | High | False | Yes | | D2 | Rainy | Mild | High | False | Yes | | 3 | Rainy | Cool | Normal | False | Yes | | 4 | Overcast | Cool | Normal | True | Yes | | 5 | Sunny | Cool | Normal | False | Yes | | 6 | Rainy | Mild | Normal | False | Yes | | 7 | Sunny | Mild | Normal | True | Yes | | 8 | Overcast | Mild | High | True | Yes | | D9 | Overcast | Hot | Normal | False | Yes | | 1 | Sunny | Hot | High | False | No | | 2 | Sunny | Hot | High | True | No | | 3 | Rainy | Cool | Normal | True | No | | 4 | Sunny | Mild | High | False | No | | 5 | Rainy | Mild | High | True | No | | |  |  |  | | --- | --- | --- | | Ai | j | Điểm | | A1  Outlook | j = 1 (Overcast) | 4 | | j = 2 (Rainy) | 3 | | j = 3 (Sunny) | 2 | | A2  Temperature | j = 1 (Hot) | 2 | | j = 2 (Mild) | 4 | | j = 3 (Cool) | 3 | | A3  Humidity | j = 1 (High) | 3 | | j = 2 (Normal) | 6(max) | | A4  Windy | j = 1 (False) | 6(Max) | | j = 2 (True) | 3 |   **BẢNG 1 VỚI NHÃN –‘YES”**   |  |  |  | | --- | --- | --- | | Ai | j | Điểm | |  | j = 1 (Rainy) | 2 | | j = 3 (Sunny) | 3 | | A2 | j = 1 (Hot) | 2 | | j = 2 (Mild) | 2 | | j = 3 (Cool) | 1 | | A3 | j = 1 (High) | 4 | | j = 2 (Normal) | 2 | | A4 | j = 1 (False) | 4 | | j = 2 (True) | 3 |   **BẢNG 2 VỚI NHÃN –‘NO”** |

**Bước 2:** Chọn bảng 1. Đếm số đối tượng và cho điểm. Điểm số lớn nhất sẽ được chọn. Trường hợp có những giá trị của các đặc trưng khác nhau cùng điểm, về nguyên tắc chọn tùy ý.

**Bước 3.** Viết luật. Tiêu chí: chọn giá trị của thuộc tính nào có điểm cao nhất (ứng với trường hợp chiếm số ngày nhiều nhất). Trường hợp có các ngày lớn nhất bằng nhau, chọn tùy ý, Dựa vào bảng 1, ta chọn đượcviết luật:

**Luật 1: IF “Windy” là “False”**



**THEN “Play” là “Yes”**

**Bước 4**: Bảng dữ liệu 1 còn 3 ngày với thuộc tính “Windy”=”True”. Để học tiếp, tổ hợp chấp 2 thuộc tính.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ***Đổi số mới*** | ***Outlook***  ***A1*** | ***Temp***  ***A2*** | ***Humidity***  ***A3*** | ***Windy***  ***A4*** | ***Play*** | | 4 | Overcast | Cool | Normal | True | Yes | | 7 | Sunny | Mild | Normal | True | Yes | | 8 | Overcast | Mild | High | True | Yes | | |  |  |  | | --- | --- | --- | | Ai | J | Điểm | | (A4 là True) ^ A1 | j=1 (Overcast) | 2 | | j = 2 (Sunny) | 1 | | (A4 là True) ^ A2 | j = 1 (Cool) | 1 | | j = 2 (Mild) | 2 | | (A4 là True) ^ A3 | j = 1 (Normal) | 2 | | j = 2 (High) | 1 | |

Viết luật 2: **IF**  (A4 là True) ^ (A1 là Overcast)

**THEN “Play” là “Yes”**

**Bước 5**: Lặp lại bước 4. Ta tổ hợp chập 3, thì có hai trường hợp, và lập bảng tương tự. Tuy nhiên, trong trường hợp này chỉ còn 1 ngày. Sau hai luật được lập ở bảng 1, số bản ghi còn lại chưa được xếp luật còn lại như sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Đổi số mới*** | ***Outlook***  ***A1*** | ***Temp***  ***A2*** | ***Humidity***  ***A3*** | ***Windy***  ***A4*** | ***Play*** | |
| 7 | Sunny | Mild | Normal | True | | Yes | |

nên có 2 luật là tương đương:

Viết luật 3: **IF**  (A4 là True) ^ (A1 là Overcast) ^( ***Humidity(A3)* là** Normal)

**THEN “Play” là “Yes”**

**Hoặc: IF** (A4 là True) ^ (A1 là Overcast) ^( ***Temp(A2)* là** Mild)

**THEN “Play” là “Yes”.**

Sau ba luật được lập ở bảng 1, số bản ghi đã được xếp luật hết. Ta chuyển sang xét lặp lại cho bảng hai một cách tương tự.

**Bước 6. Lăp lại các bước trên với bảng 2**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ***Đổi TT mới*** | ***Outlook***  ***A1*** | ***Temp***  ***A2*** | ***Humidity***  ***A3*** | ***Windy***  ***A4*** | ***Play***  ***=N0*** | | 1 | Sunny | Hot | High | False | No | | 2 | Sunny | Hot | High | True | No | | 3 | Rainy | Cool | Normal | True | No | | 4 | Sunny | Mild | High | False | No | | 5 | Rainy | Mild | High | True | No | | |  |  |  | | --- | --- | --- | | Ai | j | Điểm | | A1 | j = 1 (Rainy) | 2 | | j = 3 (Sunny) | 3 | | A2 | j = 1 (Hot) | 2 | | j = 2 (Mild) | 2 | | j = 3 (Cool) | 1 | | A3 | j = 1 (High) | 4 | | j = 2 (Normal) | 1 | | A4 | j = 1 (False) | 2 | | j = 2 (True) | 3 |   **BẢNG 2 VỚI NHÃN –‘NO”** |

**Luật 4: : IF** A3 là High

**THEN “Play” là “No”**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Đổi TT mới*** | ***Outlook***  ***A1*** | ***Temp***  ***A2*** | ***Humidity***  ***A3*** | ***Windy***  ***A4*** | ***Play***  ***=N0*** |
| 3 | Rainy | Cool | Normal | True | No |

**Tổ hợp chập 2 và có 3 luật tương đương:**

**Luật 5: IF** (A3 là Normal) ^( ***A1* là** Rainy)

**THEN “Play” là “No”**

**Hoặc IF** (A3 là Normal) ^( ***A2* là** Cool)

**THEN “Play” là “No”**

**Hoặc IF** (A3 là Normal) ^( ***A4* là** True)

**THEN “Play” là “No”**

Có thể kết hợp các luật 1, 2, 3 thành một luật với (Play là “yes”). Có thể kết hợp các luật 4, 5, thành một luật thứ hai với (Play là “No”) để trở thành một hàm (Sinh viên tự làm)

***Remark:***

1. Thuật toán ILA dựa theo thống kê nên có thể gọi là luật học thống kê (Statistic Learning Rule). Mặt khác, dữ liệu cũng được chia thành 2 bảng như một cây, cho nên cũng có thể xếp vào loại học cây quyết định. Trường hợp đầu ra có nhiều hơn 2, ta chia nhiều nhiều hơn 2 bảng. Việc học cũng tiến hành tương tự.
2. Do thuật toán thuộc loại cây, cũng có thể dùng các tiêu chí đánh giá độ chính xác bằng ma trận nhầm lẫn, hoặc các độ đo chính xác khác (sẽ nghiên cứu sau)
   1. **Các thuật toán học cây quyết định có tính chất tổ hợp**
3. Do cây quyết định có độ chính xác không cao, độ quá khớp (Over-Fitting) lớn nên ngườ ta chủ trương kết nhiều cây để học. Khi dữ liệu lớn (Big Data) thì dữ liệu được coi như rừng cây (Tree Forest) và có nhiều cách học tổ hợp. Có thể kể đến:

* Phương pháp Bagging
* Phương pháp Boosteping
* Phương pháp Cây trong rừng ngẫu nhiên (Random Tree)
* Phương pháp rừng ngẫu nhiên (Random Forest)

(tham khảo của Đỗ Thanh Nghị, Đại học Cần Thơ: Nguyên lý Học máy, Khai mỏ dữ liệu)

* Nhiều phương pháp khác (Xem References)

1. Rất nhiều phần mềm, và phương pháp mới cải tiến cũng được phát triển (Xem thêm các tài liệu khác) trong thời đại của AI, cụ thể là học máy.

**BÀI TẬP LỚN VÀ THÍ NGHIÊM 1 (Machine Learning) Ngày 2 tháng 3 năm 2022**

1. *Phần Bài tập lớn:* Cho tệp dữ liệu thời tiết. Sử dụng tuật toán C4.5 rút ra các hàm hàm xâp xỉ đích dưới dạng các luật.
2. *Phần Thí nghiệm ảo.*
   1. Cho dữ liệu Rám nắng
   2. Cho dữ liệu “Weather”.

Hãy a) Dùng phần mềm Weka, Chạy in kết quả

b) In sơ đồ cây quyết định và giải thích các chi tiêu đánh giá.

1. *Bài tập và thí nghiệm chương**4*
   1. Giả sử có hai phương pháp xét nghiệm A và B độc lập với nhau và cho phép phát hiện cùng một loại virus. Phương pháp A cho phép phát hiện 95% trường hợp nhiễm virus thật nhưng lại cho kết quả dương tính đối với 10% số người không có virus. Phương pháp B chỉ phát hiện được 90% trường hợp nhiễm virus thật nhưng chỉ cho kết quả dương tính sai với 5% số người không nhiễm virus. Biết rằng xác suất nhiễm virus trong cộng đồng dân cư là 1%. Giả sử một người xét nghiệm bằng một trong hai phương pháp trên và có kết quả dương tính với virus. Trong trường hợp sử dụng phương pháp nào thì kết quả đáng tin cậy hơn (xác suất người đó nhiễm virus thật cao hơn)? Trình bầy phương pháp tính xác suất cụ thể cho câu trả lời.
   2. Nam báo cáo cô giáo đã làm bài tập nhưng quên vở ở nhà. Từ kinh nghiệm giảng dậy của mình, cô giáo biết rằng chỉ 1% số sinh viên đã làm bài tập quên vở và báo cáo với cô giáo như vậy. Trong khi đó, một nửa số sinh viên chưa làm bài tập sẽ báo cáo quên vở. Thống kê cũng cho thấy số sinh viên làm bài tập chiếm 90% sinh viên cả lớp. Hãy tính xác suất Nam nói thật.

* 1. Hãy chứng minh công thức *P(X, Y|Z)=P(X |Z) P(Y|Z)* tương đương với mỗi công thức sau:

*P(X|Y,Z)=P(X|Z) và P(Y|X, Z)=P(Y |Z).*

1. Hãy sử dụng phần mềm WEKA, chọn chức năng phân lớp, với tập dữ liệu « Weather », thử nghiệm, báo cáo, đanh giá độ chính xác, in kết quả. Ghi chú: tiêu chí đánh giá phân lớp dùng thuật toán Bayes, trong WEKA ; gọi là thuật toán Naive Bayes. Trong khi đó mạng Bayes gọi là Bayes Net. Các tiêu chí đánh giá độ chính xác phân lớp của Thuật toán phan Bayes, dùng ma trận nhầm lẫn (*Cofusion Matrix)*, giông các thuật toán cay quyết định với các chỉ tiêu : *Accuracy, Recall, Precision, F-Measure…*
2. *Thử tính hai hàm*

hồi quy tuyến tính và hồi quy logicstic theo hàm đã cho theo bảng dữ liệu “Bẹnh nhân”. Thử hai-ba dữ liệu để rút ra kết luận, hàm nào khớp với hàm đích tìm được? Vẽ sơ đồ theo dính toán và bảng biến thiên.

1. Trong WEKA CÓ TỆP DỮ LIỆU VỀ CÁC LOẠI HOA IRIS. ThỰc hiện mở twpj chạy chức năng phân cụm, báo cáo kết quả (In danh sach tóm tắt+Đanh giá dung ma trận nhầm lẫn)

*Yêu cầu 1) Thời hạn nộp (13 ngày): Tức thứ 3 tuần sau nữa (Ngày 15 tháng 3), 20h00 nộp bản mềm cho Hồng, 21h00 cùng ngày gửi thầy.*

1. *Đánh máy. Ai giống nhau chỉ lấy ½ điểm. Kết quả lấy điểm của lần Kiểm tra 1.*

**CHƯƠNG 4. THUẬT TOÁN BAYES**

* 1. **Tóm tắt Lý thuyết xác suất. Lý thuyết quyết định Bayes**

Bayes: Tên nhà khoa học về xác suất. Ông nổi tiếng với định lý vè xác suất có điều kiện Phần này nhắc lại một số kiến thức về xác suất cần thiết cho những nội dung tiếp theo. Không chỉ nhắc lại, chúng ta nghiên cứu sâu thêm những tính chất liên quan đến các sự kiện có tính logic [1. 0] hoặc [T (đúng), F (sai)] có tính ứng dụng nhiều trong học máy. Sau đó cung cấp ví dụ và bài toán dự báo sử dụng thuật toán Bayes.

***4.1.1. Các tiên đề xác suất***

Giả sử *A, B* là các biến ngẫu nhiên (hoặc các sự kiện, hoặc các biến cố). Khi sử dụng trong suy diễn xác suất, ta có ba tiên đề xác suất như sau:

1. (0=0%) ≤ *P* (*A*) ≤ (1=100%). (4.1)

Trong đó, *P*(*A*) là xác suất của sự kiện *A; P*(*A*) (số thực). Ví dụ, trong bảng cơ sở dữ liệu thời tiết, nếu gọi *A là sự kiện có (yes) số ngày chơi Tennis là m=9* trong tổng số các ngày thử nghiệm *n=14* ngày thử nghiệm thì *P*(*A*)=9/14). Đây là tiên đề áp dụng cho các sự kiện liên tục, rời rạc. Hai tiên đề dưới đây liên quan tới các sự kiện rời rạc hay sự kiện nhận giá trị logic (1 hoặc 0).

1. *P*(*A=true*)=1; *P*(*A=false*) = *0*. (4.2)

Sự kiện *A* có thể nhận một trong hai giá trị true (đúng) và false (sai). Ví dụ, cho *A* là mệnh đề hay sự kiện “*Ngày có (True) đi chơi Tennis”* có thể nhận giá trị đúng (hoặc 1) hoặc mệnh đề hay sự kiện “*Ngày không (False) đi chơi Tennis”* có thể nhận giá trị sai (hoặc 0).

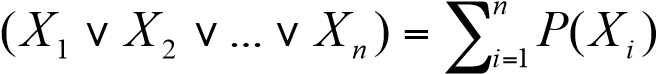
1. *P*(*A*v*B*)=*P*(*A*)+*P*(*B*)–*P*(*A*∧*B*) (4.3)

Có thể minh họa tiên đề thứ 3 bằng biểu đồ Venn, với giả sử tập các sự kiện *A và B* giao nhau

*P*(*A*∧*B*)

|  |  |
| --- | --- |
| *P* (*A* v *B*) = *P* (*A*) + *P* (*B*) – *P* (*A* ∧ *B*)  *P(B)*  *P(A)* | *P*(*A*v*B*)=*P*(*A*)+*P*(*B*)  *P(B)*  *P(A)* |

Một cách tổng quát cho tiên đề 3, khi có nhiều sự kiện ngẫu nhiên *Xi; i=1..n* là sự kiện ngẫu nhiên không giao nhau:

*P* (4.3’)

*Ghi chú:* Trường hợp sự kiện *A* và *B* không giao nhau: *P*(*A*v*B*)=*P*(*A*)+*P*(*B*)

***4.1.2. Các tính chất xác suất***

Ngoài các tiên đề trên, xác suất có một số tính chất quan trọng sau

1. *P*(¬*A*)=1–*P*(*A*) (4.4)
2. *P*(*A*)=*P*(*A***^***B*)+*P*(*A***^**¬*B*) (4.5)

B nhận giá trị nhị phân.

1. Σa *P*(*A*=*a*) = 1 (4.6)

trong đó, tổng lấy theo các giá trị *a* thuộc miền giá trị của *A,*

***4.1.3. Xác suất với biến ngẫu nhiên đa trị***

Ở trên ta đã xét trường hợp biễn ngẫu nhiên có thể nhận một trong hai giá trị true hoặc false. Trong suy diễn nói chung, biễn ngẫu nhiên có thể nhận một trong nhiều hơn hai giá trị. Trong phần này ta sẽ xem xét các tính chất xác suất cho trường hợp biến ngẫu nhiên đa trị.

Giả sử *A* là biến ngẫu nhiên có thể nhận một trong *n* giá trị {*v*1, *v*2, …, *vn*}. Do *A* chỉ có thể nhận một trong các giá trị này nên hai sự kiện (*A*=*vi*) và (*A*= *vj*) không thể đồng thời đúng nếu *i*≠*j*, do vậy:

*P*(*A*=*vi*∧*A*=*vj*)=*0* (4.7)

Tiếp theo, do ít nhất một trong các sự kiện (*A*= *vi*), *i*=*1*,…,*n* phải đúng nên:

*P*(*A*= *v1*∨*A*=*v2* ∨...∨*A*= *vn*)=1

Sử dụng các tiên đề xác suất và hai tính chất trên, có thể chứng minh:

*P*(*A*=*v*1∨*A*=*v*2∨...∨*A*=*vk* )=∑*P*(*A*=*vi*), với *k* ≤ *n*, *i*=*1..k*.

do đó có thể suy ra:

*P*(*A*=*vi*) =1

Tiếp theo, giả sử B là biến ngẫu nhiên nhị phân. Sử dụng tiên đề xác suất và các tính chất trên, có thể chứng minh:

*P*(*B*∧[*A*=*v*1 ∨*A*=*v*2 ∨... ∨*A*=*vk* ]) =∑*P*(*B*∧*A*=*vi* )

Từ đây suy ra:

*P*(*B*) =∑*P*(*B*∧*A*=*vi* )

Các tính chất nói trên sẽ được sử dụng khi trình bầy các vấn đề liên quan tới suy diễn cho trường hợp biến ngẫu nhiên đa trị trong các phần sau.

***4.1.4. Xác suất của các sự kiện xảy ra đồng thời***

Xác suất của các sự kiện xẩy ra đồng thời là xác suất quan sát được khi đồng thời xẩy ra các sự kiện đó. Xác suất đồng thời quan sát thấy các sự kiện *V*1 = *v*1, và *V*2 = *v*2, …, *Vn* = *vn* được ký hiệu như sau:

*P* (*V*1=*v*1 ∧*V*2=*v*2∧…∧*Vn*=*vn*)

hoặc theo luật suy diễn nhập hội:

*P* (*V*1=*v*1, *V*2 = *v*2, …, *Vn* = *vn*)

Một bài toán suy diễn xác suất gồm *n* biến ngẫu nhiên, mỗi biến có thể nhận một số giá trị rời rạc với những xác suất nhất định. Phân bố xác suất đồng thời xác định xác suất xẩy ra từng tổ hợp giá trị của tất cả *n* biến ngẫu nhiên.

Phân bố xác suất đồng thời đóng vai trò quan trọng trong suy diễn xác suất. Với một bài toán suy diễn xác suất, nếu chúng ta biết phân bố xác suất đồng thời tức là xác suất tất cả các tổ hợp giá trị của các biến liên quan thì ta có thể tính được xác suất mọi mệnh đề liên quan tới bài toán đang xét.

***Ví dụ:*** Cho 3 biến nhị phân (True, False): “Chim”, “Non”, “Bay được”. Ta có các xác suất đồng thời cho trong bảng 4.1. Mỗi d ng trong bảng này tương ứng với một tổ hợp giá trị của ba biến ngẫu nhiên đang xét. Số d ng trong bảng bằng tổng số tổ hợp có thể có, tức là 23 trong trường hợp này và 2*m* trong trường hợp ta có *m* biến nhị phân. Cột ngoài cùng bên phải của bảng chứa xác suất xẩy ra tổ hợp giá trị tương ứng. Lưu ý rằng, theo các tiên đề xác suất, tổng của tất cả các xác suất đồng thời trong bảng phải bằng 1.

Từ bảng xác suất này, ta có thể tính giá trị mọi xác suất liên quan tới 3 biến của bài toán. Sau đây là ví dụ tính một số xác suất:

Xác suất một vật nào đó là “chim”:

P(Chim=T)= 0+ 0,2+0,04+0.01= 0,25.

Xác suất “chim không biết bay”:

P(Chim=T, Bay= F)= 0,04 + 0,01=0,05.

***Trường hợp tổng quát***

Trong trường hợp tổng quát, xác suất của bất cứ tổ hợp giá trị *E* nào của một hoặc nhiều trong số *n* biến có thể tính từ bảng xác suất đồng thời của *n* biến đó như sau:

*P*(*E*) = ∑*P*(*các*\_*dòng*); *các*\_*dòng*\_*trùng*\_*với*\_*E*

Có thể dễ dàng kiểm tra các ví dụ trên là những trường hợp riêng của công thức này. Chẳng hạn, ở ví dụ “chim không biết bay”, *E* là (Chim=T, Bay=F) và *P*(*E*) bằng tổng xác suất ở các dạng có Chim=T và Bay=F.

**Bảng 4.1. Ví dụ bảng xác suất đồng thời cho ba biến nhị phân**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chim** | **Bay được** | **Non** | **Xác suất** |
| T | T | T | 0 |
| T | T | F | 0,2 |
| T | F | T | 0,04 |
| T | F | F | 0,01 |
| F | T | T | 0,01 |
| F | T | F | 0,01 |
| F | F | T | 0,23 |
| F | F | F | 0,5 |

* 1. **Xác suất có điều kiện**

Xác suất có điều kiện hay xác xuất Bayes. Ký hiệu *P*(*A*|*B*).

Trong công thức *P(A|B);* biến (hay sự kiện) *A* được gọi là ***biến hỏi;***

biến (hay sự kiện) *B* được gọi là ***biến điều kiện, hay bằng chứng.***

***4.2.1. Định nghĩa***

Xác suất có điều kiện, ký hiệu là *P*(*A*|*B*) là xác suất xẩy ra cả sự kiện *A và* cả sự kiện *B* đồng thời xảy ra, hay chung nhau (tức là: *P*(*A*∧*B*) hay tương đương với *P*(*A*, *B*) khi sự kiện *B* (hay biến cố *B* đã xảy ra). Dễ dàng mô tả hay biểu diễn *P*(*A*|*B*) như sau:

(4.8)

Từ đó, dẫn tới công thức toán học, mô tả xác suất điều kiện theo định nghĩa như sau:

*P(A|B) = P*(*A*∧*B*)/*P*(*B*) (4.9)

Quá trình suy diễn theo Bayes là quá trình tính xác suất điều kiện của kết luận khi biết bằng chứng (***bằng chứng*** *có thể ký hiệu là* ***E****: (****E****vidence)*. Cụ thể, khi biết các bằng chứng *E*1, …, *En*, suy diễn xác suất được thực hiện bằng cách tính xác suất điều kiện *P*(*Q*|*E*1, …, *En*), tức là niềm tin kết luận ***Q*** (***Q****uery)* đúng khi có các bằng chứng *E*1, …, *En*.

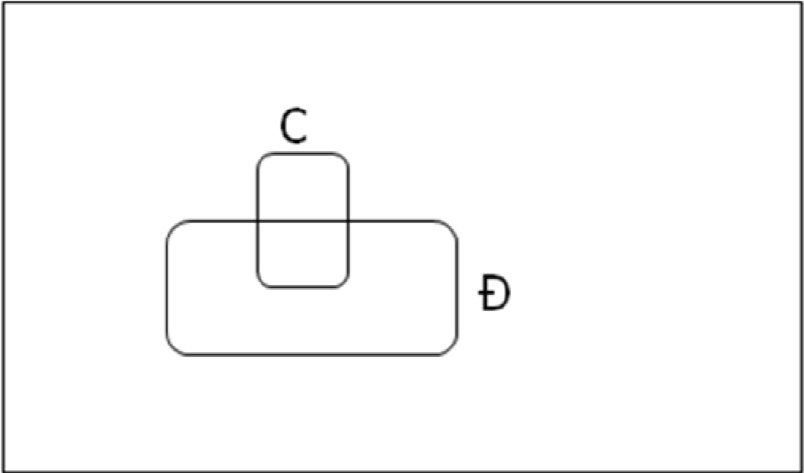
***Ví dụ***

Để hiểu rõ hơn về ý nghĩa của xác suất điều kiện, xét ví dụ với sơ đồ Venn được thể hiện trên hình vẽ dưới đây. Trong hình vẽ, hình chữ nhật bên ngoài thể hiện thế giới *U (Univese: tập vũ trụ)* của bài toán, trong trường hợp này là toàn bộ người trên trái đất chẳng hạn. Hình chữ nhật *C* là tập con, thể hiện tập hợp những người bị bệnh cúm và hình chữ nhật *Đ* là tập hợp những người bị đau đầu. *P(C)* là xác suất một người bị cúm và tính bằng tỷ lệ hình chữ nhật *C* so với hình chữ nhật to, ở đây là số người được khảo sát. Tương tự, *P(Đ)* là xác suất một người bị đau đầu. Phần giao của *C* và *Đ* tức là những người vừa bị cúm vừa đau đầu, hay *C∧Đ*. Theo hình vẽ, ta có:

*P(C)=1/40; P(Đ)=1/10; P(C∧Đ)=1/80*

Theo hình vẽ, cả cúm và đau đầu đều có xác suất không lớn. Tuy nhiên, nếu ta biết một người bị cúm, tức là nằm trong hình chữ nhật *C* thì sẽ có 1/2 xác suất người đó nằm trong hình chữ nhật Đ, tức là 50% xác suất người đó bị đau đầu. Nói cách khác, ta có:

*P(Đ|C) =P(C∧Đ)/P(C)=1/80\*1/40=1/2*



*P(Đ|C)=P(C∧Đ)/P(C*)=

*P(C∧ Đ) = 1/80*

Đau đầu: *P(Đ)=1/10*

Cúm: *P(C)=1/40*

***4.2.2. Các tính chất của xác suất điều kiện***

1. **Qui tắc nhân**

*P*(*A*, *B*)=*P(A∧B)=P*(*A*|*B*)\*P(*B*) (4.10)

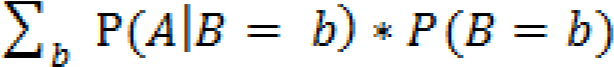
1. **Qui tắc chuỗi (là mở rộng của quy tắc nhân)**

*P*(*A*, *B*, *C*, *D*)=*P(A∧B∧C∧D)=P* (*A*|*B*, *C*, *D*)\**P*(*B*|*C*, *D*)\**P*(*C*|*D*)\**P*(*D*) (4.11)

1. **Quy tắc chuỗi có điều kiện**

*P*(*A*, *B*|*C*)=*P*(*A*|*B*, *C*)\**P*(*B*|*C*) (4.12)

1. **Quy tắc cộng**

*P* (*A*) =  (4.13)

***4.2.3. Tính xác suất điều kiện từ bảng xác suất đồng thời***

Trong trường hợp có đầy đủ xác suất đồng thời như đã xem xét ở trên, ta có thể thực hiện suy diễn thông qua tính xác suất điều kiện. Sau đây là một số ví dụ minh họa sử dụng bảng xác suất đồng thời trong phần trên với ký hiệu “Chim” là C, “Bay” là B và “Non” là N

***Ví dụ.*** Giả sử các bằng chứng cho thấy một sinh vật biết bay, cần tính xác suất sinh vật đó là chim. Theo định nghĩa ta có:

*P* (Chim|Bay)=*P*(*C*|*B*) =*P*(*C*, *B*)/*P* (*B*)=

*P(C, B, N) + P(C, B, ~N)*

*P(C, B, N) + P(~C, B, N) + P(C, ~B, N) + P(~C, ~B, N)*

= (0 + 0,2) / (0 + 0,04 + 0,01 +0,23)

***Ví dụ***. Ví dụ này minh họa cho việc kết hợp nhiều bằng chứng. Giả sử có bằng chứng sinh vật biết bay và không c n non, cần tính xác suất đây không phải là Chim. Ta có:

*P*(~ *C*,*B*,~ *N*) = 0,01

*P* (¬Chim | Bay, ¬Non) =  *0,048*

*P*(*C*,*B*,~ *N*) +*P*(~ *C*,*B*,~ *N*) 0,01+ 0,2

Trong trường hợp tổng quát, khi cho bảng xác suất đồng thời của n biến V1, …, Vn, ta có thể tính xác suất của một số biến này khi biết giá trị một số biến khác như sau:

**Công thức tổng quát**

*P (V1 = v1, … Vk = vk| Vk+i = vk+i, … Vn = vn)=*

tổng các d ng có *V*1 = *v*1, …, *VK* = *vk, Vk+i = vk+i, … Vn = vn*

= (4.14)

tổng các d ng có *Vk+i = vk+i, … Vn = vn*

Một cách hình thức hơn, gọi các biến cần tính xác suất là Q, các biến đã biết là E, các biến c n lại (ngoài E và Q) là Y, ta có

∑*P*(*Q*, *E*,*Y*)

*P*(*Q* | *E*) = *Y* (4.15)

∑*P*(*Q*, *E*,*Y*)

*Q*,*Y*

Mặc dù công thức trên tương đối đơn giản nhưng trên thực tế, khi số lượng biến tăng lên, số lượng các d ng chứa tổ hợp giá trị các biến ở tử số và mẫu số của công thức trên sẽ tăng theo hàm mũ, dẫn tới kích thước bảng xác suất đồng thời quá lớn nên không thể sử dụng cho suy diễn xác suất được. Trên thực tế, việc suy diễn với bảng xác suất đồng thời chỉ có thể thực hiện được với những bài toán có dưới 10 biến ngẫu nhiên nhị phân.

***4.2.4. Tính độc lập xác suất***

Tính độc lập xác suất là một trong những tính chất quan trọng, cho phép giảm số lượng xác suất cần biết khi xây dựng bảng xác suất đồng thời.

Tính độc lập xác suất được định nghĩa như sau: sự kiện A độc lập về xác suất với sự kiện B nếu:

*P*(*A* |*B*)=*P*(*A*) (4.16)

Tức là biết giá trị của B không cho ta thêm thông tin gì về A.

***Ví dụ***. Giả sử A là sự kiện một sinh viên thi môn trí tuệ nhân tạo được 10 điểm, và B là sự kiện năm nay là năm nhuận. Việc biết năm nay là năm nhuận hay không rõ ràng không làm tăng thêm xác suất sinh viên được 10 điểm trí tuệ nhân tạo và do vậy ta có *P* (*A* | *B*) = *P* (*A*), tức là A độc lập xác suất với B.

Tương đương với công thức trên, khi A độc lập với B, ta có:

*P*(*A*, *B*) = *P*(*A*) *P*(*B*) (4.10)

và

*P*(*B* | *A*) = *P*(*B*) (4.11)

(người đọc có thể tự chứng minh hai công thức này bằng cách sử dụng định nghĩa xác suất điều kiện).

Công thức cuối cho thấy sự độc lập xác suất có tính đối xứng: nếu *A* độc lập xác suất với *B* thì ngược lại ta có *B* độc lập xác suất với *A*. Từ hai công thức trên cũng có thể nhận thấy từ hai xác suất *P*(*A*) và *P*(*B*), ta có thể tính được toàn bộ xác suất đồng thời của hai biến ngẫu nhiên A và B như sau:

*P* (*A*, *B*) = *P*(*A*) *P*(*B*) (4.12)

*P* (*A*, ¬*B*) = *P*(*A*) *P*(¬*B*) = *P*(*A*) (1− *P*(*B*)) (4.13)

*P (¬A, B) = P(¬A) P(B) = (1−P(A)) P(B)* (4.14)

*P (¬A, ¬B) = P(¬A) P(¬B) = (1−P(A)) (1− P(B))* (4.15)

Như vậy, thay vì cần biết 4 xác suất đồng thời, nay ta chỉ cần biết 2 xác suất. Nhận xét này cho thấy tính độc lập xác suất cho phép biểu diễn rút gọn bảng xác suất đồng thời, làm cơ sở cho việc suy diễn trong những bài toán suy diễn xác suất với nhiều biến ngẫu nhiên.

***4.2.5.******Độc lập xác suất có điều kiện***

Ở trên ta đã xét trường hợp độc lập xác suất không điều kiện. Mở rộng cho trường hợp xác suất điều kiện, ta nói rằng A độc lập có điều kiện với B khi biết C nếu:

*P* (*A* | *B*, *C*) = *P* (*A* | *C*) hoặc *P* (*B* | *C*) *P*(*A*,*C*) = *P* (*B* | *C*)

khi đó ta có:

*P*(*A*, *B* | *C*) = *P* (*A* | *C*) *P*(*B* | *C*) (4.16)

Ý nghĩa: nếu đã biết giá trị của C thì việc biết giá trị của B không cho ta thêm thông tin về A.

Ví dụ: giả sử ký hiệu A là sự kiện sinh viên đi học muộn, B là sự kiện sinh viên mặc áo mưa, C là trời mưa. Thông thường, nếu quan sát thấy sinh viên mặc áo mưa (B = true) thì xác suất có sinh viên đi học muộn sẽ tăng lên (do đường bị ngập, phương tiện di chuyển chậm hơn), tức là P(A|B) ≠ P(A). Tuy nhiên, nếu ta đã biết trời mưa (C= true) và biết xác suất đi học muộn tăng lên, thì việc nhìn thấy sinh viên mặc áo mưa không làm tăng tiếp đổi xác suất sinh viên đi học muộn nữa.

Tức là: *P*(*A*| ,*B*, *C*)=*P*(*A*|*C*).

***4.3.5. Định lý hay quy tắc Bayes***

***Định lý hay quy tắc Bayes***

***Chứng minh:***

Theo định nghĩa: *P(A|B) = P*(*A*∧*B*)/*P*(*B*) hay (1)

Tương tự, ta có:  (2)

Từ tính gIao hoán của phép “ tức là: hay:

Vì 2 vế của công thức Bayes đều tính xác suất điều kiện cả. Cần gì phải tráo đổi giữa các biến hỏi và biến điều kiện?

Quy tắc Bayes đóng vai trò quan trọng trong suy diễn xác suất. Thông thường, bài toán suy diễn đòi hỏi tính *P*(*A*|*B*). Tuy nhiên, trong đại đa số các trường hợp, việc tính *P*(*B*|*A*) có thể sử dụng rất ít các phép tính và do đó dễ dàng hơn so với tính *P*(*A*|*B*). Khi đó, quy tắc Bayes cho phép ta quy việc tính *P*(*A*|*B*) về tính *P* (*B*|*A*).

***Ví dụ*** ***minh họa ý nghĩa của định lý Bayes***:

Giả sử cần tính xác suất một người bị cúm khi biết người đó đau đầu, tức là tính *P(cúm|đau đầu)*. Để tính xác suất này, cần xác định có bao nhiêu người bị đau đầu trong một cộng đồng dân cư, sau đó đếm xem có bao nhiêu người trong số người đau đầu bị cúm. Rõ ràng việc thống kê những người đau đầu tương đối khó khăn do không phải ai đau đầu cũng do cúm và cũng thông báo cho cơ sở y tế. Ngược lại, để tính *P(đau đầu|cúm),* ta cần đếm số người bị cúm trong số những người đau đầu tức là tính *P(cúm|đau đầu)*. Việc này được thực hiện tương đối dễ dàng hơn với số ca (trường hợp) chắc chắn sẽ ít hơn rất nhiều.

Sau đây, ta xem xét ví dụ sử dụng quy tắc Bayes cho suy diễn.

**4.3. Thuật toán Bayes và ứng dụng cho dự báo**

***4.3.1.******Bài toán***

***Input :*** chobảng dữ liệu học chơi Tennis

**Bảng 4.2. Bảng dữ liệu học chơi Tennis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***A=*** | ***A1*** | ***A2*** | ***A3*** | ***A4*** | ***B*** |
| ***Day*** | ***Outlook (Sky)*** | ***Temperature*** | ***Humidity*** | ***Wind*** | ***PlayTennis*** |
| *Dl* | Sunny | Hot | High | False | No |
| *D2* | Sunny | Hot | High | True | No |
| *D3* | Overcast | Hot | High | False | Yes |
| *D4* | Rainy | Mild | High | False | Yes |
| *D5* | Rainy | Cool | Normal | False | Yes |
| *D6* | Rainy | Cool | Normal | True | No |
| *D7* | Overcast | Cool | Normal | True | Yes |
| *D8* | Sunny | Mild | High | False | No |
| *D9* | Sunny | Cool | Normal | False | Yes |
| *Dl0* | Rain | Mild | Normal | False | Yes |
| *Dl1* | Sunny | Mild | Normal | True | Yes |
| *Dl2* | Overcast | Mild | High | True | Yes |
| *Dl3* | Overcast | Hot | Normal | False | Yes |
| *Dl4* | Rain | Mild | High | True | No |

***Output:*** Dữ liệu của mẫu tin 1 cần dự báo Play= No? Play=Yes dùng thuật toán Bayes

|  |  |
| --- | --- |
|  | **Outlook Temp Humidity Windy Play: B** |
| *D15* | Sunny Hot High True ? |

***4.3.2.******Thuật toán Bayes***

Thuật toán Bayes về cơ bản thuộc loại cây quyết định. Nhưng kỹ thuật tính khác với họ cây quyết định của Quinlan :

* Trong cây quyết định : sử dụng toán xác suất và Entropy
* Trong cây Bayes : sử dụng toán xác suất có điều kiện hay xác suất Bayes

Nói cách khác : Thuật toán Bayes vẫn dùng phương pháp « Devide & Conquer » : Chia để trị.

**Bước 1**: *Chia (phân hoạch) theo đặc trưng đầu vào* hay còn gọi là lập mô hình

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GỐC: Outlook: A1 | | | Temp:A2 | | | Humidity:A3 | | | Windy: A4 | | | Play | |
| Yes | | No | Yes | | No | Yes | | No | Yes | | No | Yes | No |
| Sunny  Overcast  Rainy | 2  4  3 | 3  0  2 | Hot  Wild  Cool | 2  4  3 | 2  2  1 | High  Normal | 3  6 | 4  1 | False  True | 6  3 | 2  3 | 9 | 5 |



**Bước 2:** *Sử dụng công thức Bayes*

***B2.1.*** *Xác định biến hỏi (B(yes, no)=Play(yes, no)); Biến điều kiện*

*(A={A1, A2, A3, A4}={A1 A2A3A4}).* Từ đó, tính 2 công thức với:

***B2.2.*** (4)

(5)

Trong công thức (4), (5): =

=

Tính Để đơn giản, ta giả sử các đặc trưng đầu vào A1…A4 độc lập (không phụ thuộc) của xác suất có điều kiện. Khi đó ta có:

Từ đó, chúng ta có:

(6)

(7)

***B2.3.*** Lấy các giá trị yêu cầu và thay vào hai công thức cuối cho mỗi thuộc tính để dự báo:

|  |  |
| --- | --- |
|  | **Outlook: A1 Temp :A2 Humidity:A3 Windy:A4 Play:B** |
| *D15* | Sunny Hot High True ? |

= (8)

= (9)

***B3. So sánh (8) với (9)***

Hai công thức tính xác suất có điều kiện có mẫu số *P(A)* chung, để só sánh chúng, ta chỉ cần so sánh hai tử số của chúng. Dễ thấy (9)>(8), suy diễn, suy ra,

*Ghi chú: Kết luận mà ta có được bằng phép SUY DIỄN theo quy luật: SỐ LỚN (MAX), hay lối suy diễn này là “luật bầu cử, bỏ phiếu”*

**\**

* 1. **Đánh giá bộ phân lớp**

1. Thuật toán trên gọi là thuật toán Bayes ngây thơ hay đơn giản (Simple). Lý do: ta đã giả thiết các đặc trưng *A1: Outlook,…,* *A4: Windy* độc lập với nhau nên mới có công thức (6) và (7).
2. Phương pháp suy diễn ở đây là áp dụng quy luật số lớn hay quy tắc bỏ phiếu theo nghĩa xác suất của biến hỏi có giá trị nào lớn hơn, thì chọn. Trong ví dụ bài toán “Weather” nhãn (đầu ra ) được dự đoán: Không đi chơi
3. Trường hợp chúng ta dự đoán

|  |  |
| --- | --- |
|  | **Outlook Temp Humidity Windy Play** |
| *D15* | Sunny Hot High True ? |

Hoàn toàn trùng hợp với mẫu ngày “D2” trong bảng dữ liệu 14 ngày được đưa vào huấn luyện. Điều đó khẳng định phương pháp dự báo chính xác. Sinh viên, tự chọn một mẫu khác trường hợp để dự đoán, ví dụ

|  |  |
| --- | --- |
|  | **Outlook Temp Humidity Windy Play** |
| *D15* | Overcast Hot High True ? |

1. Trường hợp các đặc trưng *A1: Outlook,…, A4*: Windy không độc lập với nhau thì các công thức (6) và (7) không dùng được. Muốn tính các trường hợp đó, một trong các phương pháp là tính xác suất có điều kiện và sử dụng mạng Bayes (phần tiếp theo), hoặc sử dụng công thức đầy đủ của xác suất có điều kiện (nếu có dữ liệu)
   1. **Mạng Bayes (xem tài liệu [6])**

**Bài tập và thí nghiệm chương 4**

1. Giả sử có hai phương pháp xét nghiệm A và B độc lập với nhau và cho phép phát hiện cùng một loại virus. Phương pháp A cho phép phát hiện 95% trường hợp nhiễm virus thật nhưng lại cho kết quả dương tính đối với 10% số người không có virus. Phương pháp B chỉ phát hiện được 90% trường hợp nhiễm virus thật nhưng chỉ cho kết quả dương tính sai với 5% số người không nhiễm virus. Biết rằng xác suất nhiễm virus trong cộng đồng dân cư là 1%. Giả sử một người xét nghiệm bằng một trong hai phương pháp trên và có kết quả dương tính với virus. Trong trường hợp sử dụng phương pháp nào thì kết quả đáng tin cậy hơn (xác suất người đó nhiễm virus thật cao hơn)? Trình bầy phương pháp tính xác suất cụ thể cho câu trả lời.
2. Nam báo cáo cô giáo đã làm bài tập nhưng quên vở ở nhà. Từ kinh nghiệm giảng dậy của mình, cô giáo biết rằng chỉ 1% số sinh viên đã làm bài tập quên vở và báo cáo với cô giáo như vậy. Trong khi đó, một nửa số sinh viên chưa làm bài tập sẽ báo cáo quên vở. Thống kê cũng cho thấy số sinh viên làm bài tập chiếm 90% sinh viên cả lớp. Hãy tính xác suất Nam nói thật.

1. Hãy chứng minh công thức *P(X, Y|Z)=P(X|Z)P(Y|Z)* tương đương với mỗi công thức sau

*P(X |Y, Z)=P(X |Z) và P(Y|X, Z)=P(Y|Z).*

1. Hãy sử dụng phần mềm WEKA, chọn chức năng phân lớp, với tập dữ liệu « Weather », thử nghiệm, báo cáo, đanh giá độ chính xác, in kết quả. Ghi chú: tiêu chí đánh giá phân lớp dùng thuật toán Bayes, trong WEKA ; gọi là thuật toán Naive Bayes. Trong khi đó mạng Bayes gọi là Bayes Net. Các tiêu chí đánh giá độ chính xác phân lớp của Thuật toán phan Bayes, dùng ma trận nhầm lẫn (*Cofusion Matrix)*, giống các thuật toán cây quyết định với các chỉ tiêu : *Accuracy, Recall, Precision, F-Measure…*

**CHƯƠNG 5**

**HỒI QUY**

* 1. **Khái niệm về hồi quy**
     1. ***Bản chất của hồi quy (Regression) là gi?***

1. *Khái niệm và thuật ngữ*

Đây là thuật ngữ Hán-Việt, có thể dịch: quay về để quy kết. Trong ngữ cảnh học máy, sau khi ta xác định được hàm quan hệ giữa tập đầu vào ***X*** và tập đầu ra ***Y***, ta lấy hàm kết quả đó ***Y=f(X)*** quy kết (tức dự báo) cho những giá trị chưa biết ***X*** mới. Hàm quan hệ mà ta nói có thể là hàm dạng đường thẳng (tuyến tính), dạng đường cong (phi tuyến), dạng liên tục có giới hạn trong miền chuẩn {0,1} gọi là hồi quy logic. Về bản chất hồi quy cũng có một trình tự như bài toán phân lớp/dự báo: trong đó: ***X*** là tập dữ liệu đầu để huấn luyện, ***Y:*** nhãn. Sự khác nhau, ta sẽ làm rõ sau.

1. *Bản chất của hồi quy*

Theo Tom Mitchell [13], hồi quy có nghĩa là (tìm) hàm xấp xỉ (tính gần đúng) với một hàm đích (*Target Function*) *y=f(x)* mà hàm đó có thể nhận các giá trị thực (Real-Valued): Nguyên văn: “Regression means approximating a real-valued target function”. Bắt đầu chú ý sự khác biệt: Các bài toán đã học trong phân loại như cây quyết định “Logic-valued target function”, trong khi đó, hồi quy “Real-Valued Target Function”. Từ đó ta biết hồi quy để tính hàm ra là hàm liên tục, hay giá trị số, hay phương pháp định lượng, còn phân lớp/dự báo thì hàm đầu ra là rời rạc (là nhị phân, là tam phân, tứ phân…) hay phương pháp dự báo định tính. Đây là sự khác nhau cơ bản.

* + 1. ***Ý nghĩa và mục đích của bài toán hồi quy***

***Như vậy*,** Học máy có thể chia thành hai nhóm bài toán dựa trên dạng dữ liệu đầu vào

* Nếu dữ liệu đầu, vào dạng ngôn ngữ (hay dạng nhóm: Category hay Nominal (dữ liệu danh nghĩa, hay có tài liệu gọi là dữ liệu định danh) thì ta gọi là phân lớp/phân loại/nhận dạng… Nhiều trường hợp, có dữ liệu số, muốn áp dụng phương pháp phân lớp cho kiểu dữ liệu này, người ta phân nhóm, tức tạo Category rồi áp dụng các phương pháp đã nêu. Dự đoán hay dự báo kiểu phân lớp còn gọi là dự báo theo ***định tính***.
* Nếu dữ liệu đầu vào dạng số (Numeric) thì ta dùng phương pháp hồi quy… Dự báo dùng phương pháp hồi quy dựa trên giá trị bằng số nên gọi là dự báo theo ***định lượng***.

Phương pháp hồi quy trong khoa học thuộc loại kinh điển, hình thành và phát triển sớm hơn phương pháp phân lớp, mang tính toán học, áp dụng trong quy hoạch (tuyến tính, phi tuyến) ứng dụng đặc biệt nhiều trong các bài toán kinh tế, bài toán dự báo khí tượng, thủy văn trên các số liệu đo được cụ thể. Tuy nhiên, các bài toán xã hội mang tính cảm nhận của con người, khó đo chính xác mà chỉ thể hiện bằng giá trị ngôn ngữ là xu hướng phát triển mang tính hiện đại (Hiện đại theo nghĩa: thời đại hiện nay).

Ngoài ra, hồi quy là bài toán cho giá trị nhãn (hay có đầu ra mong muốn), nên thuộc loại học có tín hiệu chỉ đạo (*Super-vised Learning).*

Có nhiều loại hồi quy khác nhau, dựa theo dạng của hàm *f(x).* Nếu *f(x)* có dạng đường thẳng, thì *f(x)* gọi là hàm hồi quy tuyến tính. Nếu *f(x)* có dạng đường cong (không thẳng) *f(x)* gọi là hàm hồi quy phi tuyến. Nếu *f(x)* với ***X là nhieu biến,*** *f(x)* gọi là hàm hồi quy bội. Ta xét cụ thể.

* 1. **. Hồi quy tuyến tính**
     1. ***Hồi quy đơn (một biến): single variable***
        1. *Khái niệm, phân tích, hàm hồi quy một biến*

Đầu tiên, xét hồi quy một biến (hay hồi quy đơn) tuyến tính.Giả sử có tập hợp đầu vào ***X***; tập thâm niên (các năm) làm việc và tập đầu ra ***Y***; mức lương thụ hưởng (theo nghìn đô la) của một tập dữ liệu huấn luyện hay đơn giản là cơ sở dữ liệu sau [15].

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Bảng 5.1. Bảng dữ liệu về lương [15]*** | | | | | | *Hình 1. Quan hệ lương và kinh nghiệm* |
| TT (i) | ***X:* Năm** | ***Y:* Lương** | TT | ***X:* Năm** | ***Y:* Lương** | *y=w0+wx*  *x*  *y* |
| *1* | 3 | 30 | *6* | 6 | 43 |
| *2* | 8 | 57 | *7* | 11 | 59 |
| *3* | 9 | 64 | *8* | 21 | 90 |
| *4* | 13 | 72 | *9* | *1* | 20 |
| *5* | *6* | 36 | *10* | 16 | 83 |

Chúng ta sẽ tự hỏi: liệu mức lương có tỷ lệ thuận với năm công tác không? Tỷ lệ thuận đó về mặt toán học được biểu diễn bằng một hàm đường thẳng, hàm quan hệ đó gọi là hàm tuyến tính. Giả sử nó là hàm tuyến tính với *y là đầu ra (hay đáp ứng); x là đầu vào (hay biến độc lập)*:

(5.1)

trong đó: là các tham số. Ở lĩnh vực hồi quy, ngươi ta đặt ra tên riêng gọi là các hệ số hồi quy hay là các trọng số (weighs).

Trong trường hợp tổng quát, để xác định trong (5.1) với *y, x* là liên tục, về mặt toán học, ta tìm trước, bằng cách vi phân 2 vế của (5.1):

(5.2)

Trong trường hợp bảng dữ liệu ở bảng trên trên với *y, x* là đa giá trị ta có , về mặt toán học, ta tính

= (5.3)

*;*  là các giá trị trung bình cộng của các đầu ra, các đầu vào một cách tương ứng từ bảng dữ liệu thực nghiệm.

Trong thực hành, chúng ta thường nhân tử và mẫu số của (5.3) với ta có:

= ; (5.4)

trong đó : số lượng (lực lượng) của các mẫu quan sát hay số lượng các trường hợp. Ở bài toán của ta:

* + - 1. *Ví dụ hồi quy một biến*

Dựa vào Bảng 5.1, và công thức (5.4) tính được *;*  ta có:

Dựa theo công thức (5.1) ta có:

(5.5)

Rõ ràng, đây là hàm tuyến tính cần tìm.

* + - 1. *Mô hình hồi quy một biến*

Rõ ràng, ví dụ trên cho thấy hàm tương quan bậc nhất có thể có sai số. Để mô tả bài toán đầy đủ, theo nghĩa có đánh giá, (kể cả phân tích sai số), ta nêu mô hình học máy theo hồi quy đơn. Dựa vào phân tích (5.1), ta có mô hình tổng quát cho hồi quy đơn tuyến tính sau:

(5.6)

trong đó: là các tham số của mô hình, được ước lượng bằng phương pháp bình phương tối thiểu; là sai số. Các sai số được giả thiết:

* Giả thiết trung bình sai số bằng *0,* độc lập và chuẩn hóa*.* Nói theo toán học, là biến ngẫu nhiên, có phân bố chuẩn, với giá trị trung bình cộng bằng không hoặc phương trình tính kỳ vọng theo thống kê, xác suất bằng *0:* *.*
* Giả thiết phương sai là hằng số*.* và không phụ thuộc vào biến số *x*

Từ (5.5), ta có thể nội suy (suy diễn trong phạm vi bảng dữ liệu huấn luyện) và ngoại suy (suy diễn các giá trị của *y (lương)* khi cho bất kỳ giá trị ngoài nào của *x (số năm công tác).*

***Bài tập:***

***1.*** Sinh viên tự cho ví dụ minh họa và thử để dự đoán. Vẽ lại sơ đồ có thứ nguyên (giá trị của trục, giá trị của từng mẫu ứng với các điểm theo bảng dữ liệu huấn luyên đã cho.

***2****.* Dựa trên cách tính hồi quy đơn, sinh viên xây dựng thuật toán tìm hàm hồi quy hay hàm tương quan (có thể dung theo bước)

* + 1. ***Hồi quy bội hay hồi quy tuyến tính nhiều biến***

* + - 1. *Khái niệm, phân tích, hàm hồi quy bội*

Các khái niệm hồi quy bội tương tự hồi quy đơn. Điểm khác nhau cơ bản là thay vì mô tả một biến ***x=[****xi]= [x1, …, xn]* trong đó, *n* là số các trường hợp, hay ví dụ (ở ví dụ trên *n=10*) thì ở đây, hồi quy bội có *m* biến tức là ***X={x1,…xj ,…xm}.*** Mỗi ***xj=****[xij]=[ x11,…,xij,…,xnm].* Khi đó:

(5.7)

trong đó: là tập các tham số. Ở lĩnh vực hồi quy, nên ngươi ta đặt ra tên riêng gọi là các hệ số hồi quy hay là các trọng số (weighs).

Trong trường hợp tổng quát, để xác định trong (5.7) với ***y****,* ***X***là liên tục, về mặt toán học, ta tìm tập các đạo hàm riêng theo (5.7):

*l=1..m* (5.8)

* + - 1. *Mô hình hồi quy bội trong các phần mềm SPSS*

Rõ ràng, ví dụ trên cho thấy có thể có sai số và mô hình cũng có dạng tương tự. Tuy nhiên, mô tả toán học trở nên phức tạp hơn. Để khắc phục việc mô tả và tính toán, hiện nay có các gói phần mềm thống kê, xác suất được cung cấp đầy đủ như: SAS, SPSS, S-Plus…

* 1. **. Hồi quy phi tuyến**

Chúng ta sẽ tự hỏi: liệu có tồn tại mô hình không phải là mô hinh tuyến tính được mô tả như (5.1) không? Có, đó là mô hình quan hệ vào ra dạng hàm logarit, hàm mũ, hàm lượng giác, các hàm dạng đa thức v.v… đều có dạng đường cong, gọi là các hàm phi tuyến.

Ví dụ điển hình là hàm mô tả quan hệ vào ra theo đa thức:

(5.9)

Để dự báo, đơn giản nhất, người ta chuyển dạng bậc cao về dạng tuyến tính bội bằng cách đổi biến

Khi đó (5.9) trở thành (5.10)

Và việc giải được tiến hành như hồi quy tuyến tính bội.

* 1. **. Hồi quy logicstic**

Hồi quy bội đã nêu là phương pháp phổ dụng để xấp xỉ quan hệ vào ra nhằm dự báo cho nhiều trường hợp đầu vào khác nhau. Tuy nhiên trong đại đa số các trường hợp, biến đầu ra thường dùng cho phân loại phổ biến hơn biến liên tục, khi đó, các nhà phân tích có xu hướng chuyển tới một phương pháp khác tương tự như hồi quy bội, gọi là hồi quy logistic. Hồi quy logistic là phương pháp mô tả quan hệ giữa biến phân loại và một tập các biến dự đoán.

### 5.4.1. Khái niệm và ví dụ hàm hồi quy logistic

Ví dụ dưới đây xét quan hệ một biến đầu vào ***X***; tuổi của bệnh nhân có hay không *[1, 0]* một bệnh *y* cho trong tập dữ liệu huấn luyện hay đơn giản là cơ sở dữ liệu sau [15].

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Bảng 5.1. Bảng dữ liệu về BỆNH NHÂN*** | | | | | | *Hình 1. Quan hệ TUỔI và BỆNH* |
| ID | ***X: Tuổi*** | ***Y: Bệnh*** | ID | ***X:* Năm** | ***Y: Bệnh*** | *1*  *x*  *y*    *0* |
| *1* | 25 | 0 | *11* | 50 | 0 |
| *2* | 29 | 0 | *12* | 59 | 1 |
| *3* | 30 | 0 | *13* | 60 | 0 |
| *4* | 31 | 0 | *14* | 62 | 0 |
| *5* | 32 | 0 | *15* | 68 | 1 |
| *6* | *41* | 0 | *16* | 72 | 0 |
| *7* | 41 | 0 | *16* | 79 | 1 |
| *8* | *42* | 0 | *18* | 80 | 0 |
| *9* | *44* | 1 | *19* | 81 | 1 |
| *10* | *49* | 1 | *20* | 84 | 1 |

### Phân tích hàm hồi quy logistic

1. *Phân tích:*

Dữ liệu là dùng cho phân lớp theo nghĩa đầu ra **Y** (nhãn) có giá trị nhị phân {0,1};

*Đầu vào:* ***X*** mang giá trị số (Numeric). Dữ liệu ở đây mang tính chất lai giữa hồi quy và phân loại. Ta gọi là hồi quy Logicstic. Do lai, ta xét cả hai trường hợp:

* Giả thiết thứ nhất: Trên sơ đồ, đường hồi quy tuyến tính mô tả bằng đường thẳng (màu đỏ) dạng như (5.11)

(5.11)

Từ bảng dữ liệu “Bệnh nhân”: ?;

* Giả thiết thứ 2: Đường hồi quy Logistic mô tả bằng hàm Sigmoid (màu xanh) dạng (5.12):

(5.12)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 25 | 29 | 30 | 31 | … |  | 80 | 81 | 84 |
|  |  |  |  |  |  |  |  |  |  |

Sau đó, ta sẽ đánh giá: với đầu vào *x* là tập dữ liệu thử, hàm (thể hiện qua mô tả bằng đường tuyến tính và phí tuyến) nào khớp, chính xác hơn?

**5.5. Hiệu suất của mô hình**

Một khi bạn xây dựng mô hình, câu hỏi tiếp theo đến trong đầu là để biết liệu mô hình của bạn có đủ để dự đoán trong tương lai hoặc là mối quan hệ mà bạn đã xây dựng giữa các biến phụ thuộc và độc lập là đủ hay không. Vì mục đích này có nhiều chỉ số mà chúng ta cần tham khảo

R – Square (R^2) (5.13)

Công thức tính R^2 sẽ bằng:

alt (5.14)

* **Tổng các diện tích (TSS)**: TSS là một phép đo tổng biến thiên trong tỷ lệ đáp ứng / biến phụ thuộc Y và có thể được coi là số lượng biến thiên vốn có trong đáp ứng trước khi hồi quy được thực hiện.
* **Sum of Squares (RSS)**: RSS đo lường lượng biến đổi còn lại không giải thích được sau khi thực hiện hồi quy.
* (**TSS - RSS)** đo lường mức độ thay đổi trong đáp ứng được giải thích (hoặc loại bỏ) bằng cách thực hiện hồi quy

trong đó, *N* là số quan sát được sử dụng để phù hợp với mô hình, σx là độ lệch chuẩn của *x*, và *σy* là độ lệch chuẩn của *y.*

* R2 giao động từ 0 đến 1.
* R2 của 0 nghĩa là biến phụ thuộc không thể dự đoán được từ biến độc lập
* R2 của 1 có nghĩa là biến phụ thuộc có thể được dự đoán mà không có sai số từ biến độc lập
* Một R2 giữa 0 và 1 chỉ ra mức độ mà biến phụ thuộc có thể dự đoán được. Một R2 của 0.20 có nghĩa là 20 phần trăm của phương sai trong Y có thể dự đoán được từ X; Một R2 của 0.40 có nghĩa là 40 phần trăm là có thể dự đoán v.v...

***Root Mean Square Error (RMSE).***

RMSE cho biết mức độ phân tán các giá trị dự đoán từ các giá trị thực tế. Công thức tính RMSE là

alt (5.15)

*N*: Tổng số quan sát

Mặc dù RMSE là một đánh giá tốt cho các sai số nhưng vấn đề với nó là nó rất dễ bị ảnh hưởng bởi phạm vi của biến phụ thuộc của bạn. Nếu biến phụ thuộc của bạn có dải biến thiên hẹp, RMSE của bạn sẽ thấp và nếu biến phụ thuộc có phạm vi rộng RMSE sẽ cao. Do đó, RMSE là một số liệu tốt để so sánh giữa các lần lặp lại khác nhau của mô hình

***Mean Absolute Percentage Error (MAPE)***

Để khắc phục những hạn chế của RMSE, các nhà phân tích thích sử dụng MAPE so với RMSE. MAPE cho sai số trong tỷ lệ phần trăm và do đó so sánh được giữa các mô hình. Công thức tính MAPE có thể được viết như sau:

alt (5.16)

*N*: tổng số quan sát

**5.6. Các mô hình hồi quy khác**

Dễ nhận thấy, mỗi khi các nhà phân tích học máy hoặc thống kê đưa ra một dạng hàm (mô tả bằng một đường thẳng hay đường cong) với một phương pháp nào đó thì người ta đặt cho 1 tên. Lý do: trong tự nhiên, nhiều hiện tượng, quan hệ có biểu hiện rất phức tạp. Khoa học (toán học) liệu có theo kịp không, chưa? Hoặc có thể có những hàm đã tìm nhưng độ chính xác kém thì người ta tìm các đường (hàm) dự báo chuẩn hơn. Ví dụ: Dự báo bệnh Covid phát triển thế nào vào tháng 4, 5 tới?

Một số phương pháp khác có thể kể đến như:

* Toán học: hồi quy Logarit, hồi quy theo đường hình chuông (Guas) Thống kê: Đường hồi quy theo phân bổ chuẩn, theo phân bổ Poit xông, Logistic

CHƯƠNG 6

**MÁY HỌC VÉC TƠ HỖ TRỢ**

**6.1. Khái niệm**

## 6.1.1. Giới thiệu

Máy vec tơ hỗ trợ (*Support Vector Machine: SVM*) được Vapnik nghiên cứu từ những năm 1965. Đến năm 1995, thuật toán được Vapnik và Chervonenkis chính thức phát triển mạnh mẽ, đứng hàng thứ 10 trong khai phá dữ liệu; được coi là công cụ mạnh cho những bài toán phân lớp phi tuyến tính; được xem là một trong các phương pháp phân lớp có giám sát không tham số, tinh vi nhất cho đến nay. Phương pháp này thực hiện phân lớp dựa trên nguyên lý cực tiểu hóa rủi ro (sai số) có cấu trúc *SRM* (*Structural Risk Minimization)*.

SVM có dữ liệu đầu vào là một hàm nhân (Kernel Function) tạo ra mô hình mới, dẫn tới giải bài toán quy hoạch toàn phương (*Quadratic Programming)* luôn có kết quả tối ưu toàn cục *(Global Optimzation)*. Bằng việc sử dụng hàm nhân, SVM cung cấp các mô hình hiệu quả, khá chính xác cho các bài toán phân lớp, hồi quy tuyến tính và phi tuyến. Các hàm công cụ đa dạng của SVM cho phép tạo không gian chuyển đổi để xây dựng mặt phẳng phân lớp.

Ứng dụng của Phương pháp SVM chủ yếu:

­ Nhận dạng: tiếng nói, ảnh, chữ viết tay (hơn mạng nơron)

­ Phân loại văn bản, khai phá dữ liệu văn bản

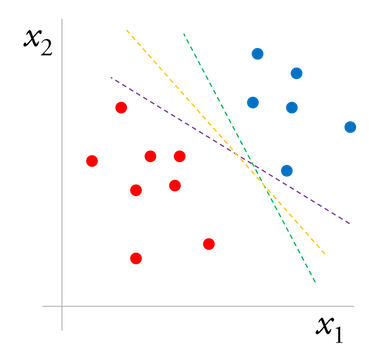
­ Phân tích dữ liệu theo thời gian

­ Phân tích dữ liệu gien, nhận dạng bệnh, công nghệ bào chế thuốc

­ Phân tích dữ liệu marketing

**6.1.2. Ý tưởng**

Cho trước tập dữ luyện huấn luyện được biểu diễn trong không gian vector; trong đó, mỗi sự kiện là một điểm, phương pháp tìm ra một đường phân cách trong mặt phẳng hoặc tìm một siêu mặt phẳng *(Hyperplane)* trong không gian nhiều chiều*,* ký hiệu siêu phẳng là *h* quyết định tốt nhất có thể chia các điểm trên không gian này thành 2 phần (hay hai lớp) riêng biệt tương ứng hai lớp: lớp “+” và lớp “-“ hay lớp “1” và lớp “-1”.



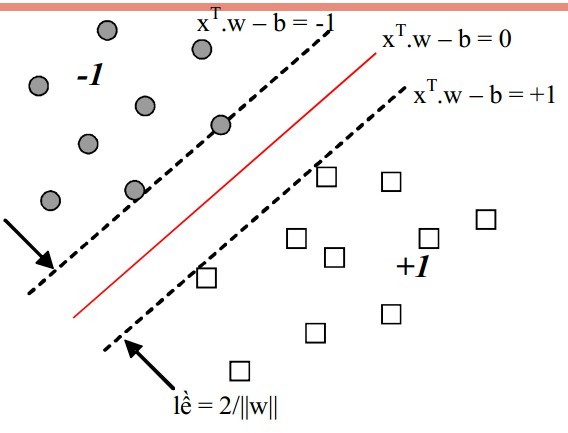


**Hình 6.1. Đường (siêu phẳng) phân lớp tuyến tính**



Hình dung, ta có bộ dữ liệu gồm các điểm trên và dưới, đặt trên cùng một mặt phẳng. Ta có thể tìm được đường thẳng để phân chia các tập điểm trên và dưới như Hình 6.1. Cho *m* phần tử: *x1,…, xi,…, xm* trong không gian *n* chiều, có nhãn (lớp, hay giá trị đầu ra) tương ứng là: *y1,…, yi,…, ym*; trong đó nhãn *yi={1,-1}.* Rõ ràng, có rất nhiều siêu phẳng tuyến tính (Hình 6.1 vẽ ba đường tuyến tính trong mặt phảng để minh họa) phân chia dữ liệu thành hai phần hoàn hảo (không có lỗi phân chia). Câu hỏi được đặt ra: mô hình tuyến tính nào là mô hình phân lớp tối ưu?

Theo tư tưởng của học máy, siêu phẳng tối ưu là siêu phẳng tách được hai lớp xa nhau nhất có thể. Siêu phẳng trong không gian *n* chiều được biểu diễn bằng véc tơ pháp tuyến ***w=****[w1,…, w1,…, wn]* và độ lệch ***b*** của siêu phẳng so với gốc của tọa độ. Máy học SVM tìm siêu phẳng ***(w, b)*** để phân chia dữ liệu theo cách sau đây (Hình 6.2):



*Hình 6.2. Siêu phẳng* ***(w, b),*** *lề của phân lơp tuyến tính*

***6.1.3. Khái niệm các véc tơ hỗ trợ (Support Vectors). Phương trình siêu phẳng tuyến tính***

Để dễ hình dung các véc tơ hỗ trợ *(Support Vectors)* đường biên của lớp trên sẽ đi qua một (hoặc một vài) điểm trên. Đường biên của lớp dưới sẽ đi qua một (hoặc một vài) điểm dưới. Các điểm trên, dưới nằm trên 2 đường biên được gọi là các *support vectors*, vì chúng có nhiệm vụ hỗ trợ *(support)* để tìm ra siêu phẳng. Đó là nguồn gốc của tên gọi thuật toán **VSM**: **Support Vector Machine**

Có thể mô tả hai siêu phẳng song song bằng hai công thức sau:

(6.1)

(6.2)

Kết hợp (6.1) và (6.2) bằng cách nhân hai vế của (6.1) với ; nhân hai vế của (6.2) với ta có:

(6.3)

Hoặc

***6.1.4. Định nghĩa lề***

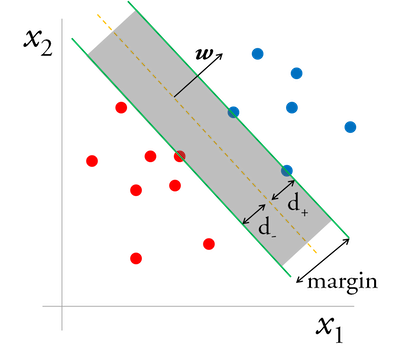
Hiệu quả xác định siêu mặt phẳng này được quyết định bởi khoảng cách của điểm gần mặt phẳng nhất của mỗi lớp. Khoảng cách càng lớn thì mặt phẳng quyết định càng tốt đồng nghĩa với việc phân loại càng chính xác và ngược lại. Mục đích cuối cùng của phương pháp là tìm được khoảng cách hay biên lớn nhất. Trong SVM sử dụng thuật ngữ lề (***Margin)*** để chỉ khoảng cách giữa hai siêu phẳng (trong trường hợp không gian 2 chiều là đường thẳng) đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp (Hình 6.2).

*Đinh nghìa lề.* Lề *(Margin)* là khoảng cách giữa hai siêu phẳng hỗ trợ song song

(6.4)

trong đó, là độ lớn của véc tơ trọng số ***W*** theo chuẩn 2 *(2-norm*)

Trong Hình 6.3 hai siêu phẳng để phân chia tập các điểm thành hai lớp: lớp trên (dương), lớp dưới (âm). Vậy, siêu phẳng như thế nào được coi là tối ưu? siêu phẳng tối ưu là siêu phẳng phân cách 2 lớp dữ liệu cách xa nhau nhất. Ngoài ra, người ta còn dùng khái niệm khoảng cách (Distance) dương *d+ (*và âm *d-)* để chỉ khoảng cách từ đường thẳng (hoặc siêu phẳng) dương (và âm) đến đường thẳng (siêu phẳng) trung bình (lấy trung bình giữa siêu phẳng dương và âm) thể hiện trên Hình 6.3.



**Hình 6.3. Mô tả lề và khoảng cách với dữ liệu tách rời**

Vì sao vế phải của hai phương trình trên chọn là 1 và −1. Đây là một phép dịch chuyển đường thẳng cơ bản về hai phía cho một đơn vị.

Ví dụ: giả sử bằng phương pháp giải tích thông thường, ta tìm được ba phương trình đường thẳng tương ứng là:

*2x1​+3x2​+5 (:4+1)=0  
2x1​+3x2​+9=0  
2x1​+3x2​+1=0*Chia tất cả cho 4 để thu được phương trình như định nghĩa:  
\frac{1}{2}x\_1 + \frac{3}{4}x\_2 + \frac{5}{4} = *0; 21​x1​+43​x2​+45​=0  
\frac{1}{2}x\_1 + \frac{3}{4}x\_2 + \frac{5}{4} = 1; 21​x1​+43​x2​+45​=1  
\frac{1}{2}x\_1 + \frac{3}{4}x\_2 + \frac{5}{4} = -1; 21​x1​+43​x2​+45​=−1*

**6.2. Tìm siêu phẳng tối ưu** **bằng cách cực đại hóa lề**

**Có 3 trường hợp:**

***6.2.1. Trường hợp 1: Tìm siêu phẳng tối ưu******cho dữ liệu tách rời***

Thuật toán SVM cố gắng tối ưu bằng cách tìm cực đại (maximize) giá trị lề này, từ đó tìm ra siêu phẳng **tối ưu** để phân lớp dữ liệu. Một điểm đặc biệt trong phương pháp SVM là siêu phẳng *h* chỉ phụ thuộc vào các vector hỗ trợ. Đây là sự khác biệt so với các phương pháp học khác. (Các phương pháp khác có kết quả phân loại phụ thuộc vào toàn bộ dữ liệu; khi dữ liệu thay đổi thì kết quả cũng thay đổi theo).

Để có siêu mặt phẳng *h* ta đi giải bài toán tìm cực đại (maximize) giá trị lề là đối ngẫu cũng là tương đương với bài toán:

(6.5)

(6.5) phải thỏa mãn điều kiện (6.3), tức là các ràng buộc

Đây là bài toán quy hoạch toàn phương *(Quadratic Programming)* có thể được giải bằng toán tử Lagrange để có thể thu được siêu phẳng phân lớp *(****w, b)*** của SVM.

|  |
| --- |
| *Ghi chú:* Bài toán quy hoạch *(Programming)* thuộc lớp các bài toán tối ưu gồm:   * Quy hoạch tuyến tính *(Linear Programming),* được giảibằng phương pháp đơn hình, trong đó hàm mục tiêu là tích vô hướng *<c, x>* được xác định bởi:   *<c, x>=*   * Quy hoạch toàn phương *(Quadratic Programming),* Hàm mục tiêu như (6.5); ràng buộc như (6.3).   Ngoài hai loại quy hoạch phổ biến trên, người ta còn sử dụng nhiều trong khoa học, CNTT các quy hoạch:   * Quy hoạch rời rạc *(Discrete Programming).* * Quy hoạch Belman *(Belman Programming)…* |

Sử dụng các nhân tử (hệ số) Lagrange như sau:

(6.5a)

Để tìm cần đạo hàm riêng của theo ***w*** và ***b*** rồi cho bằng 0. Như sau:

(6.5b)

(6.5c)

Khi đó công thức (6.5a) có thể viết lại là:

(6.5d)

Từ đó, bài toán quy hoạch toàn phương (6.5) có thể viết lại:

|  |
| --- |
| (6.5e)  Với điều kiện ràng buộc: |

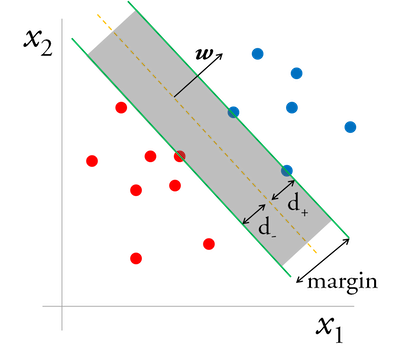
Khi cần dự đoán (phân lớp) cho một phần tử mới đến *x* nào đó dựa trên siêu phẳng đã được giải *(****w, b),*** ta có thể tính (hay dự báo*: Predict*) theo công thức sau:

*Predict(****x****)= sign(****w.x-b****)* (6.6)

Có thể giải thích công thức (6.6) như sau. Khi phần tử mới đến ***x****,* tính giá trị của *(****wx-b****).*

Nếu *(****wx-b)>0*** thì gán nhãn cho ***x*** là (+1); ngược lại (-1). Đây là tính cho dữ liệu tách rời như hình 6.3.

***6.2.2. Trường hớp 2: Tìm siêu phẳng tối ưu*** *(****w, b) cho dữ liệu không tách rời***





**Hình 6.4. Mô tả lề và khoảng cách với dữ liệu không tách rời**

Trong trường hợp dữ liệu không tách rời (Hình 6.4), máy học SVM vẫn có thể xử lý được bằng cách xem xét bất kỳ điểm ***x****i* nào nằm sai phía (được xem là lỗi) so với siêu phẳng hỗ trợ của nó. Khoảng cách lỗi được biểu diễn bởi ***di>0*** (điểm ***x****i* nào nằm đúng phía (được xem không lỗi) so với siêu phẳng hỗ trợ của nó ***di=0***). Việc tìm kiếm siêu phẳng tối ưu trong trường hợp này giải quyết cùng lúc hai mục tiêu là cực đại hóa lề (cực tiểu hàm  ở công thức (6.5) và cực tiểu hóa lỗi. Đây cũng là bài toán quy hoạch toàn phương *(Quadratic Programming)* có thêm một hàm tối ưu và tham số ***di*** sau:

(6.7)

với các điều kiện ràng buộc:

Lời giải của (6.7) có thể tìm được tương đương trong bài toán đối ngẫu của chúng tương tự như trường hợp dữ liệu tách rời ta có thể phát biểu lại bài toán (6.7)

|  |
| --- |
| (6.7e)  Với điều kiện ràng buộc: |

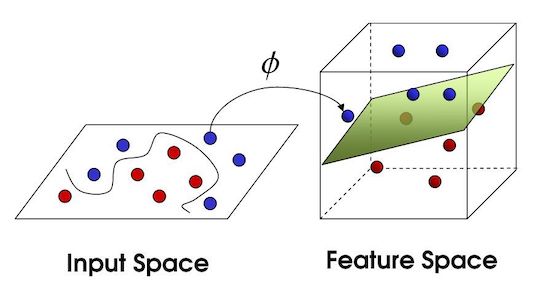
Khi cần dự đoán (phân lớp) cho một phần tử mới đến *x* nào đó dựa trên siêu phẳng đã được giải *(****w, b),*** ta có thể tính (hay dự báo*: Predict*) theo công thức sau:

*Predict(****x****)=)* (6.8)

Mặc dù Giải thuật SVM cơ bản giải quyết được bài toán phân lớp tuyến tính, nhưng đối với phân lớp phi tuyến cũng cần phải nghiên cứu.

***6.2.2. Trường hớp 3: Tìm siêu phẳng tối ưu******cho phân lớp phi tuyến. Hàm nhân***

|  |  |
| --- | --- |
| Hình 6.5. Phân lớp phi tuyến | Hình 6.6. Phân lớp phi tuyến trong không gian trung gian |



***Hình 6.7. Ánh xạ tập dữ liệu từ không gian 2 chiều sang không gian 3 chiều***

Xét vấn đề phân lớp trong không gian 2 chiều *(****d1, d2****)* như hình 6.5. Để giải quyết vấn đề, người ta chuyển không gian đầu vào *(Input Space (****d1, d2****)* qua không gian trung gian hay không gian đặc trưng (*Feature Space),* hình 6.6 và hình 6.7 có nhiều chiều hơn cụ thể là 5 chiều (*(****d1, d2, d1d2****, (****d1)2*** *,(****d2****)2)* để phân hoạch. Vấn đề phân lớp đầu vào trở thành vấn đề phân lớp không gian đặc trưng. Ở đây chỉ cần hai chiều (*(****d1)2****,* ***d2****)* là đủ để có thể phân lớp tuyến tính (xem hình 6.6).



Tuy nhiên, việc từ không gian đầu vào thành không gian đặc trưng rất lớn và khó xác định, dẫn tới việc điều chỉnh tham số của SVM khó, mô hình không khả thi. Rất may, có một lớp hàm nhân (Không gian Hilbert Schmidt) có thể giúp SVM làm việc trực tiếp trong không gian đầu vào nhưng ngầm định xử lý ở không gian đặc trưng. Không có bất kỳ thay đổi nào trong giải thuật; việc duy nhất cần làm là thay tích vô hướng của hai véc tơ <***u, v>***  trong các công thức (6.5e); (6.7e); (6.8) bằng hàm nhân *(Kernel Function)* ***K(u, v).*** Các hàm nhân cơ bản được dùng phổ biến như sau:

* Tuyến tính: ***K(u, v)= u.v***
* Đa thức bậc *d*: ***K(u, v)= (u.v+c)d, d=2,3,4,…*** là bậc của đa thức
* Hàm cơ sở xuyên tâm *(Radial Basic Function: RBF):*

***K(u, v)= exp(-)=***

Trong các bài toán của ta, bằng cách thay thế tích vô hướng 2 véc tơ ***xi,xj*** đầu vào trong công thức (6.7e) bằng hàm nhân ***K(xi, xj*  *)*** *công thức (6.9) là tổng quát cho cả trường hợp phi tuyến và tuyến tính* để tìm mặt (phẳng hoặc cong) phân cách.

|  |
| --- |
| (6.7e)  Với điều kiện ràng buộc: |

|  |
| --- |
| (6.9)  Với điều kiện ràng buộc: |

Tương tự, trong công thức (6.8) thay thế tích vô hướng 2 véc tơ ***xi,x*** bằng hàm nhân ***K(xi, xj*  *)*** thành *công thức (6.10) là dự đoán cho cả trường hợp phi tuyến và tuyến tính*.

*Predict(****x****)=)* (6.8)

*Predict(****x****)=)* (6.10)

Có thể sử dụng thư viện *LibSVM* để thực hiên các ví dụ minh họa

|  |
| --- |
| *Nhắc lại kiến thức nhân véc tơ (Ma trận)*  Cho 2 véc tơ: *ký hiệu bằng chữ in đậm* ***x, y với x=***   * Tích 2 véc tơ * Tích 2 véc tơ * Tích 2 véc tơ   ***: Scalar: vô hướng*** |

**6.3. Giải thuật học máy SVM**

Cơ chế hoạt động của máy học SVM có thể mô tả bằng thuật toán như hình 6.8.

|  |
| --- |
| ***Đầu vào (Input)***   * Tập dữ liệu huấn luyện gồm *m* phần tử: *x1,…, xi,…, xm* trong không gian *n* chiều, có nhãn (lớp, hay giá trị đầu ra) tương ứng là: *y1,…, yi,…, ym*; trong đó nhãn *yi={1,-1};* * Hàm ***K*** với các tham số;   ***Huấn luyện (Traning)***   * Giải bài toán quy hoạch toàn phương   (6.9)  Với điều kiện ràng buộc:   * Kết quả thu được nhân tử Lagrange * Tập véc tơ *SV* ( * Tính độ lệch ***b*** từ các véc tơ hỗ trợ   ***Phân lớp (Classifying)***  Phân lớp cho phần tử mới ***x*** theo công thức (6.10):  *Predict(****x****)=)* |

**Hình 6.8. Thuật toán máy học SVM**

**6.4. SVM với việc phân nhiều lớp**

Ví dụ cần chia dữ liệu thành *k=*3 lớp? hoặc tổng quát *k>2?* Người ta có một số cách như sau:

* *Cách 1-tất cả (1 vs all) [Vapnik 1995].* Mỗi mô hình phân thành 1 lớp từ tất cả các lớp khác; và phải xây dụng *k mô hình* cho *k* lớp.
* *Cách 1-1 (1 vs 1) [Kreel 1999].* Mỗi mô hình tách đôi, và xây dựng: *k\*9k-1)/2* mo hình như vầy.
* *Phương pháp khác:* phân tách 2 nhóm: *[Vural and Dy 2004].* Mỗi nhóm có thể bao nhiều lớp, xác định cách phân xhia sao cho có lới nhất.

**6.5. Một số ví dụ minh họa**

Giả sử phương trình đường thẳng phân chia trong không gian hai chiều (*x1, x2)*:

*w1​x1​+w2​x2​ - b=0*

Đây chính là đường thẳng nằm giữa hai lề trên và dưới trong Hình 6.3.

Đường thẳng trên (dương), và dưới (âm) là hai véc tơ hỗ trợ của hai lớp dữ liệu lần lượt là:

*w1​x1​+w2​x2​-b=1*

*w1​x1​+w2​x2​-b=−1*

Vì sao vế phải của hai phương trình trên chọn là 1 và −1. Đây là một phép dịch chuyển đường thẳng cơ bản về hai phía cho một đơn vị.

Ví dụ: giả sử bằng phương pháp giải tích thông thường, ta tìm được ba phương trình đường thẳng tương ứng là:

*2x1​+3x2​+5=0  
2x1​+3x2​+9=0  
2x1​+3x2​+1=0*Chia tất cả cho 4 để thu được phương trình như định nghĩa:  
\frac{1}{2}x\_1 + \frac{3}{4}x\_2 + \frac{5}{4} = *0; 21​x1​+43​x2​+45​=0  
\frac{1}{2}x\_1 + \frac{3}{4}x\_2 + \frac{5}{4} = 1; 21​x1​+43​x2​+45​=1  
\frac{1}{2}x\_1 + \frac{3}{4}x\_2 + \frac{5}{4} = -1; 21​x1​+43​x2​+45​=−1*

***Với không gian 2 chiều***, Margin giữa 2 đường thẳng được tính bằng công thức:

*\text{margin}=\frac{2}{\sqrt{w\_1^2+w\_2^2}}margin=w12​+w22​​2*​

***Với không gian nhiều chiều,*** Margincần tìm là phương trình siêu phẳng:

*\mathbf{w}^T\mathbf{x}+b=0 wTx+b=0.*

Margin sẽ được tính bằng công thức:

*\text{margin}=\frac{2}{||\mathbf{w}||}margin=∣∣w∣∣2*

*​*

***6.4. Bài toán tìm lề cực đại***

*6.4.1. Lề mềm (Soft Margin)*

Để tránh quá khớp (**overfitting)** và có margin cao, ta chấp nhận việc một vài dữ liệu có thể bỏ qua (không được chia chính xác). Ví dụ, 1 bóng trên bị lọt sang vùng của bóng dưới. Dữ liệu này được gọi là *nhiễu*. Margin trong trường hợp này gọi là **Soft Margin**.

*6.4.2. Lề cứng (Hard Margin)*

**Hard Margin** ám chỉ việc tìm được Margin mà không nhiễu. Nói cách khác, tất cả các dữ liệu đều thoả mãn đúng sự phân chia. Với các bái toán thực tế, việc tìm được **Hard Margin** nhiều khi là bất khả thi. Vì thế, việc chấp nhận sai lệch ở một mức độ chấp nhận được là vô cùng cần thiết.

*6.4.3. Bài toán tìm lề cứng cực đại (không có nhiễu)*

Bài toán tìm lề cứng cực đại là bài toán thuộc lĩnh vực quy hoạch toàn phương (**Quadratic Programming),**

(6.5)

với điều kiện biên (6.3):.

Giải bài toán này ta thu được bộ tham số *(****W, b)***, thay vào (6.1) và (6.2) hoặc (6.3) ta thu được hai siêu phẳng phân lớp. Khi gặp một véc tơ ***x*** mới, ta sẽ dựa trên bộ tham số siêu phẳng *(****W, b)*** để dự đoán (Predict) ***x*** thuộc lớp dương (+1) hay lớp âm (-1) bằng cách sử dụng hàm dấu (Sign Function) để cho kết quả tương ứng như sau:

(6.6)

*6.4.4. Bài toán tìm lề mềm cực đại (có nhiễu)*

Đây là trường hợp dữ liệu không tách rời nhau một cách phân biệt như Hình 6.3 mà có một vài điểm từ vùng dương (+1) vượt biên (hay vượt lề) sang phía âm (-1) hoặc ngược lại. Trong trường hợp này, VSM vẫn có thể xử lý được bằng cách xem trường hợp đó là lỗi. Khoảng cách lỗi được xác định bằng khoảng cách từ siêu phẳng hỗ trợ dương (hoặc âm) vượt qua đường phân cách sang phía âm (hoặc dương) và ký hiệu là *zi* được biểu diễn bởi . khi điểm nằm đúng phía của siêu phẳng hỗ trợ. Ngược lại, .

Việc tìm siêu phẳng tối ưu trong trường hợp này thực hiện hai mục tiêu: i) cực đại hóa lề giống trường hợp không nhiễu theo phương trình (6.5); ii) cực tiểu hóa lỗi khoảng cách *zi*. Thuật toán VSM trong trường hợp này dẫn đến giải bài toán quy hoạch toàn phương tổng hai mục tiêu:

(6.7)

với điều kiện biên (6.3) nhưng thêm lỗi khoảng cách *zi*:

Hằng số *c>0* được sử dụng để chỉnh lề và lỗi.

*6.4.5. Lời giải bài toán tìm lề cứng cực đại (không có nhiễu)*

Lời giải bài toán quy hoạch toàn phương (6.5) hay (6.7) của VSM có thể tìm được bằng lời giải tương đương trong bài toán đối ngẫu của chúng. Bài toán quy hoạch toàn phương không nhiễu (6.5) có thể viết lại dưới dạng khác có sử dụng nhân tử Lagrange (Lagrange Dual Problem) như sau:

Của VSM Trong cài đặt SVM, người ta giới thiệu tham số *CC* với quy ước sau:

*C=\inftyC=∞*: Không cho phép sai lệch, đồng nghĩa với **Hard Margin**.

*CC lớn:* Cho phép sai lệch nhỏ, thu được Margin nhỏ.

*CC nhỏ*: Cho phép sai lệch lớn, thu được Margin lớn.

Tuỳ bài toán cụ thể, cần điểu chỉnh tham số CC này để thu được kết quả tốt nhất.

*6.4.6.*  Ví dụ Hiện nay, có nhiều thư viện để giải bài toán này như [CVOPT](http://cvxopt.org/examples/tutorial/qp.html). Trên thực tế ta chỉ sử dụng các thư viện có sẵn thay vì tự cài đặt thuật toán.

Xét ví dụ đơn giản: Chạy thử bằng thư viện *Scikit-Learn* với 2 lớp dữ liệu sau:

*Positive events  [(x1​, x2​)=[(1, 3),(3, 3),(4, 0)]*

*Negative events  [(x1​, x2​)=[(0, 0),(1, 2),(2, 0)]*

*Scikit-learn* cung cấp sẵn thư viện để giải *SVM* là [*SVC*](http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html) (nếu chưa có thư viện này có thể cài đặt đơn giản bằng *pip*(thay bằng *pip3* nếu muốn cài cho *Python 3*).

*pip install scikit-learn*

Ta chỉ cần viết mã vài dòng lệnh đơn giản là có thể chạy thử được thư viện này. Ở đây ta định nghĩa 2 lớp dữ liệu: *X1* có nhãn *Positive (1)*, *X2* có nhãn *Negative (-1).* *X* là mảng chứa cả 2 lớp dữ liệu *X1, X2; y* là mảng nhãn (*label*) của *X*.

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 | import numpy as np  from sklearn.svm import SVC  X1=[[1,3],[3,3],[4,0],[3,0], [2,2]]  y1=[1, 1, 1, 1, 1]  X2=[[0,0], [1,1], [1,2], [2,0]]  y2=[-1, -1, -1, -1]  X=np.array(X1+X2)  y=y1+y2  clf=SVC(kernel='linear', C=1E10)  clf.fit(X, y)  print clf.support vectors\_ |

Ở đoạn mã trên, chọn *kernel* là tuyến tính (*linear*). Chú ý, có thể chọn nhiều *kernel* khác phức tạp hơn, nhưng mục đích chỉ để thử, chọn *linear* đảm bảo chương trình chạy nhanh. Đặt*C* giả trị*1E10* hiểu là một giá trị cực lớn, mục đích để tìm Hard Margin.

Kết quả mảng các các véc tơ hỗ trợ (*support vectors)* được in ra như sau:

*[[1. 2.]*

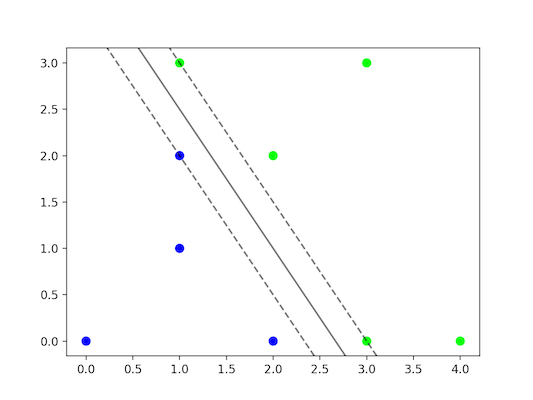
*[1. 3.]*

*[3. 0.]]*

Thêm hàm sau, sử dụng thư viện **matplotlib** để mô tả ở dạng đồ thị (cho trực quan):

|  |  |
| --- | --- |
| 1  2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 | import matplotlib.pyplot as plt  def plot\_svc\_decision\_function(clf, ax=None, plot\_support=True):  """Plot the decision function for a 2D SVC"""  if ax is None:  ax=plt.gca()  xlim=ax.get\_xlim()  ylim=ax.get\_ylim()  # create grid to evaluate model  x=np.linspace(xlim[0], xlim[1], 30)  y=np.linspace(ylim[0], ylim[1], 30)  Y, X=np.meshgrid(y, x)  xy=np.vstack([X.ravel(), Y.ravel()]).T  P=clf.decision\_function(xy).reshape(X.shape)  # plot decision boundary and margins  ax.contour(X, Y, P, colors='k', levels=[-1, 0, 1], alpha=0.5,  linestyles=['--', '-', '--'])  # plot support vectors  if plot\_support:  ax.scatter(clf.support\_vectors\_[:,0],clf.support\_vectors\_[:, 1], s=300, linewidth=1, facecolors='none');  ax.set\_xlim(xlim)  ax.set\_ylim(ylim)  plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='brg');  plot\_svc\_decision\_function(clf)  plt.show() |

Đồ thị sẽ được in ra như hình dưới:



**Điều chỉnh tham số *C***

Thay *CC* bằng 1 số nhỏ hơn, ví dụ *C=5*, ta được Margin lớn hơn mà đường phân chia vẫn đúng với tât cả dữ liệu (hình dưới).

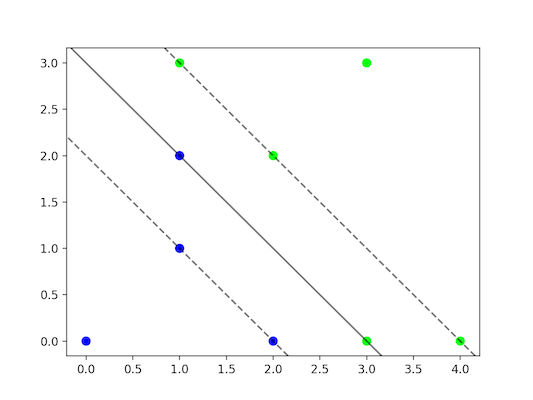


Trong bài toán thực tế, việc điều chỉnh số*CC* là quan trọng. Nó giúp cho hàm phần chia thêm phần linh hoạt.

**Tối ưu tham số C**

*Scikit learn* cung cấp công cụ tính toán tham số *C* tối ưu là *[GridSearchCV](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)*.  
Thêm vài dòng lệnh dưới đây, có thể tìm được *C* tối ưu với tập dữ liệu ví dụ:

|  |  |
| --- | --- |
| 1  2 3 4 5 6 7 8 | rom sklearn.model\_selection import GridSearchCV  parameter\_candidates =  [{'C': [0.001, 0.01, 0.1, 1, 5, 10, 100, 1000], 'kernel': ['linear']},]  clf = GridSearchCV(estimator=SVC(),  param\_grid=parameter\_candidates, n\_jobs=-1)  clf.fit(X, y)  print('Best score:', clf.best\_score\_)  print('Best C:',clf.best\_estimator\_.C) |



*Chú ý*: *parameter\_candidates* chứa các tham số cần tối ưu để thực hiện thử. Ở đây, cần thử trên không gian 2 chiều, *fix kernel* là tuyến tính (linear) và tham số *C* trong tập 7 giá trị: [0.001, 0.01, 0.1, 1, 5, 10, 100, 1000]. Sau khi chạy đoạn mã trên, ta được kết quả:

Best score: 0.7778

Best C:1

Thay *C=1* vào đoạn mã ban đầu, ta có đồ thị như hình dưới. Đường phần chia 2 lớp dữ liệu có vẻ là *hợp lý* nhất trong các trường hợp thử nghiệm.

## 6.5. Thử nghiệm thuật toán VSM với bộ dữ liệu MNIST

**MNIST** (*Modified National Institute of Standards and Technology Database*) là bộ cơ sở dữ liệu về chữ số viết tay, được cải biên từ bộ cơ sở dữ liệu gốc của NIST giúp dễ sử dụng hơn. MNIST Database vô cùng đồ sộ gồm 60k dữ liệu huấn luyện và 10k dữ liệu thử, được sử dụng phổ biến trong các thuật toán nhận dạng hình ảnh.

Mỗi dữ liệu trong bộ dữ liệu này là một ảnh đen trắng, kích thước 28x28 (784 pixels), mỗi pixel có giá trị trong khoảng [0…255]. 0 tương ứng với màu trắng, 255 tương ứng với màu đen. Có thể tải bộ dữ liệu này. Tải về và giải nén ta được 4 tệp như sau:

*train-images-idx3-ubyte: training set images*

*train-labels-idx1-ubyte: training set labels*

*t10k-images-idx3-ubyte: test set images*

*t10k-labels-idx1-ubyte: test set labels*

Chú ý 4 tệp này chứa dữ liệu các điểm ảnh ở dạng dữ liệu nhị phân

***6.5.1. Thao tác với bộ cơ sở dữ liệu***

Để dễ hình dung bộ dữ liệu vừa tải, ta sẽ viết 1 đoạn code python để đọc dữ liệu từ file trên và đổi ảnh sang định dạng *png*.

*6.5.1.1. Đọc dữ liệu nhị phân*

Viết 2 hàm *get\_images()* và *get\_labels()* để đổi (convert) dữ liệu từ tệp nhị phân sang dạng mảng như sau:

***preview\_mnist.py***

|  |  |
| --- | --- |
| 1  2 3 4 5 6 7 8 9 10 12 13 14 15 16  17 | def get\_images(img\_file, number):  f=open(img\_file, "rb") # Open file in binary mode  f.read(16) # Skip 16 bytes header  images=[]  for i in range(number):  image = []  for j in range(28\*28):  image.append(ord(f.read(1)))  images.append(image)  return images  def get\_labels(label\_file, number):  l=open(label\_file, "rb") # Open file in binary mode  l.read(8) # Skip 8 bytes header  labels=[]  for i in range(number): labels.append(ord(l.read(1)))  return labels |

*number* trong 2 hàm *get\_images(img\_file, number)*và *get\_labels(label\_file,* *number)* là số dữ liệu mà ta muốn lấy từ tệp nhị phân.

*6.5.1.2. Chuyển sang định dạng png*

Từ 2 mảng ảnh và nhãn thu được ở trên, viết hàm chuyển đổi dữ liệu sang tệp ảnh theo định dạng *png* như sau:

***preview\_mnist.py***

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 | import os  import numpy as np  from skimage import io  def convert\_png(images, labels, directory):  if not os.path.exists(directory):os.mkdir(directory)  for i in range(len(images)):  out=os.path.join(directory,"%06d-num%d.png"%(i,labels[i]))  io.imsave(out, np.array(images[i]).reshape(28,28)) |

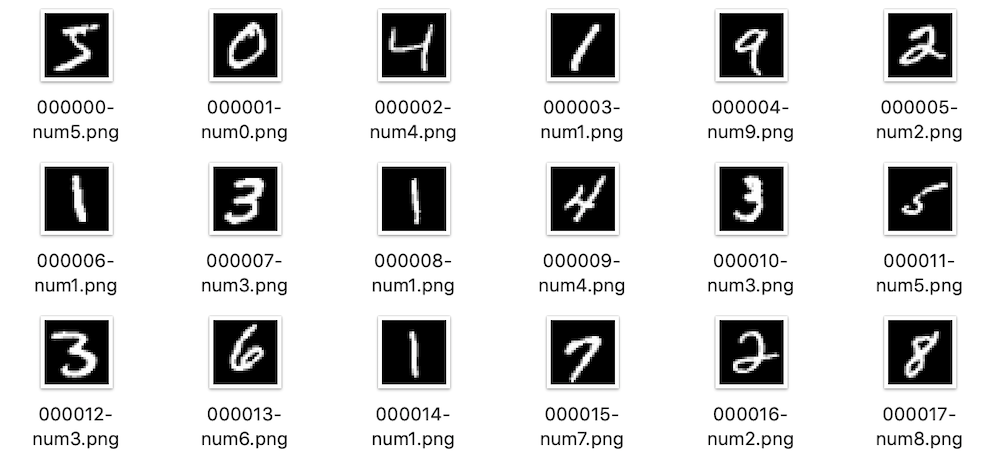
Sử dụng thư viện [*scikit-image*](http://scikit-image.org/docs/dev/install.html) để tạo ảnh theo định dạng *png*. Nếu chưa có thư viện này, cần cài đặt bằng *pip*(dùng *pip3*nếu muốn cài cho *Python 3*): *pip3 install scikit-image*

Chạy các hàm vừa viết với tệp dữ liệu huấn luyện , in ra 100 ảnh như sau:

***preview\_mnist.py***

|  |  |
| --- | --- |
| 1 2 3 4 | number=100  train\_images=get\_images("mnist/train-images-idx3-ubyte", number)  train\_labels=get\_labels("mnist/train-labels-idx1-ubyte", number)  convert\_png(train\_images, train\_labels, "preview") |

*Chú ý:* 4 tệp nhị phân tải về ta có thể lưu trong cùng thư mục *mnist*. Ảnh với định dạng *png* được in ra sẽ nằm trong thư mục *preview*, như hình dưới:



*6.5.1.3. Chuyển sang định dạng csv*

Ta cũng có thể đổi tệp nhị phân sang định dạng *csv* dùng các lệnh sau:

***preview\_mnist.py***

|  |  |
| --- | --- |
| 1 2 3 4 | def output\_csv(images, labels, out\_file):o = open(out\_file, "w")  for i in range(len(images)):  o.write(",".join(str(x) for x in [labels[i]] + images[i]) + "\n")  o.close() |

*6.5.1.4. Tải dữ liệu bằng thư viện có sẵn*

Có thể sử dụng thư viện có sẵn, ví dụ thư viện [*python-mnist*](https://github.com/sorki/python-mnist). Sau khi cài đặt, chỉ cần code thêm vài dòng đơn giản như sau là có thể tải được dữ liệu cần dùng:

|  |  |
| --- | --- |
| 1 2 3 4 | from mnist import MNIST  mndata = MNIST('./mnist')  train\_images, train\_labels=mndata.load\_training()  test\_images, test\_labels=mndata.load\_testing() |

***6.5.2. Huấn luyện và thử với SVM***

*6.5.2.1. Sử dụng thư viện SVC*

Ở đây ta sử dụng thư viện SVM có sẵn của *scikit-learn* là [*SVC*](https://1upnote.me/post/2018/11/ds-ml-svm-mnist/(https:/scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html)).

*6.5.2.2. Chuẩn hóa dữ liệu*

Trước khi đưa dữ liệu vào train, ta cần thực hiện chuẩn hoá dữ liệu ([Feature Scaling](https://en.wikipedia.org/wiki/Feature_scaling)).  
Dữ liệu có thể đến từ nhiều nguồn, với các đơn vị đo khác nhau, các thành phần có giá trị chênh lệch lớn. Để tính toán cần đưa tất cả dữ liệu về một chuẩn chung. Các thành phần có giá trị nằm cùng trong 1 khoảng như [0, 1][0,1] hoặc [-1, 1][−1,1].

Để đưa giá trị về khoảng [0,1][0,1], sử dụng phương pháp cân chỉnh lại (rescaling) bằng công thức:

*x’=\frac{x-\min(x)}{\max(x)-\min(x)}x’=max(x)−min(x)x−min(x)​*

Áp dụng với dữ liệu đã cho, mỗi điểm ảnh có giá trị trong khoảng 0..255, do vậy ta có *min(x)=0, max(x)=255*. Từ đó, ta có công thức chuẩn hoá dữ liệu:

*x’=\frac{x}{255}x’=255x​.*

Thực hiện chuẩn hoá dữ liệu mảng *train\_images* dễ dàng bằng thư viện *numpy*:

|  |  |
| --- | --- |
| 1 2 | import numpy  train\_images = numpy.array(train\_images)/255 |

*6.5.2.3. Huấn luyện*

Sử dụng dữ liệu học để huấn luyện bằng đoạn mã sau:

**svm\_mnist.py**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 | import numpy as np  from sklearn import svm, metrics  print("TRAIN")  TRAINING\_SIZE=10000  train\_images=get\_images("mnist/train-images-idx3-ubyte", TRAINING\_SIZE)  train\_images=np.array(train\_images)/255  train\_labels=get\_labels("mnist/train-labels-idx1-ubyte", TRAINING\_SIZE)  clf=svm.SVC()  clf.fit(train\_images, train\_labels) |

Đoạn mã trên, vì mục đích chạy thử, ta chỉ tải *TRAINING\_SIZE* là *10KB* dữ liệu, giúp rút ngắn thời gian huấn luyện so với việc tải toàn bộ *60KB*. Tuy nhiên, có thể khiến cho mô hình sau khi huấn luyên có độ chính xác không cao (điều này ta sẽ thử sau).

*6.5.2.4. Thử*

Tải từ tệp dữ liệu thử và tiến hành thử với mô hình vừa huấn luyên:

**svm\_mnist.py**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 | TEST\_SIZE=500  test\_images=get\_images("mnist/t10k-images-idx3-ubyte",  TEST\_SIZE)  test\_images=np.array(test\_images)/255  test\_labels=get\_labels("mnist/t10k-labels-idx1-ubyte",  TEST\_SIZE)  print("PREDICT")  predict=clf.predict(test\_images)  print("RESULT")  ac\_scoremetrics.accuracy\_score(test\_labels, predict)  cl\_report=metrics.classification\_report(test\_labels, predict)  print("Score =", ac\_score)  print(cl\_report) |

Ở đoạn mã trên, đặt *TEST\_SIZE*=*500* dữ liệu với mục đích thử, thu được:

Score=0.916. **precision recall f1-score support**

0 0.89 0.95 0.92 42

1 0.97 1.00 0.99 67

2 0.96 0.89 0.92 55

3 0.93 0.84 0.88 45

4 0.90 0.96 0.93 55

5 0.84 0.92 0.88 50

6 0.97 0.86 0.91 43

7 0.87 0.92 0.89 49

8 0.90 0.88 0.89 40

9 0.94 0.89 0.91 54

micro avg 0.92 0.92 0.92 500

macro avg 0.92 0.91 0.91 500

weighted avg 0.92 0.92 0.92 500

Mô hình thu được có độ chính xác 0.916. Ta có thể tìm cách đẩy độ chính xác cao hơn bằng cách điều chỉnh các tham số của SVC cho phù hợp.

*6.5.2.5. Điều chỉnh các tham số*

Sử dụng ***GridSearchCV*** để tìm tham số tối ưu tham số cho *svc*. Tìm *C* tốt nhất trong tập [0.001, 0.01, 0.1, 1, 5, 10, 100, 1000] bằng đoạn mã sau:

from sklearn.model\_selection import GridSearchCV

parameter\_candidates = [{'C': [0.001, 0.01, 0.1, 1, 5, 10, 100, 1000]},]

clf=GridSearchCV(estimator=svm.SVC(),param\_grid=parameter\_candidates,n\_jobs=-1)

clf.fit(train\_images, train\_labels)

print('Best score:', clf.best\_score\_)

print('Best C:',clf.best\_estimator\_.C)

Kết quả:

Best score: 0.936

Best C: 100

Thay *C=100* vào hàm khởi tạo SVC: *clf =svm.SVC(C=100)*. Tiến hành huấn luyện và thử lại với dữ liệu ban đầu, thu được kết quả tốt hơn trước với: *Score=0.95*

precision recall f1-score support

0 0.95 1.00 0.98 42

1 0.99 1.00 0.99 67

2 0.91 0.93 0.92 55

3 0.91 0.93 0.92 45

4 0.92 0.98 0.95 55

5 0.98 0.94 0.96 50

6 0.98 0.93 0.95 43

7 0.92 0.96 0.94 49

8 0.97 0.90 0.94 40

9 0.98 0.91 0.94 54

micro avg 0.95 0.95 0.95 500

macro avg 0.95 0.95 0.95 500

weighted avg 0.95 0.95 0.95 500

Ngoài tham số *C*, ta còn có thể tối ưu *kernel* (với mặc định (default) đang sử dụng la *kernel* *“rbf”),* *gamma* (default đang là *auto*) để thu được kết quả tốt hơn. Ngoài ra còn có thể tăng khối lượng tập dữ liệu huấn luyện (hiện ta đang set TRAINING\_SIZE là 10k)

*6.5.2.6. Lưu cất mô hình vào tệp*

Sau khi huấn luyện và thu được mô hình với độ chính xác tùy chọn, có thể lưu mô hình ra tệp. Chú ý, cần cài thư viện *joblib* bằng *pip* nếu trong máy chua có.

from joblib import dump, load

dump(clf, 'mnist-svm.joblib')

Hoặc load model có sẵn từ file:

clf = load('mnist-svm.joblib')

**6.6**. **Kết luận**

***6.6.1. Ưu điểm***

Là một kĩ thuật phân lớp khá phổ biến, SVM thể hiện được nhiều ưu điểm trong số đó có việc tính toán hiệu quả trên các tập dữ liệu lớn *(Big Data)*. Có thể kể thêm một số ưu điểm của phương pháp này như:

Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian có số chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi số chiều có thể cực lớn.

Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.

Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng hành nhân (Kernel Function) cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.

***6.6.2. Nhược điểm***

Bài toán số chiều lớn: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (*n*) thì SVM cho kết quả khá tồi.

Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào. Tuy nhiên hiệu quả của việc phân lớp có thể được xác định dựa vào khái niệm lề (margin) từ điểm dữ liệu mới đến siêu phẳng phân lớp mà chúng ta đã bàn luận ở trên.

PHẦN III. HỌC KHÔNG GIÁM SÁT VÀ ỨNG DỤNG

**CHƯƠNG 7**

## GIẢI THUẬT GOM CỤM K-MEAN

### Bài toán gom nhóm dữ liệu

Phân cụm (Clustering) là kỹ thuật thuộc lớp các phương pháp học không giám sát. Mục đích của phân cụm dữ liệu là tìm ra bản chất bên trong các nhóm của dữ liệu. Trong thực tế, dữ liệu thường không có nhiều thông tin sẵn có (được gán nhãn) mà chúng ta cần phải xây dựng mô hình gom cụm dữ liệu.

Giả sử cho tập dữ liệu *X={x1, x2,…, xm}* gồm *m* phần tử. Mục đích của bài toán gom nhóm dữ liệu là gom các phần tử lại với nhau để tạo thành các nhóm sao cho các phần tử tương tự nhau sẽ nằm chung một nhóm, các phần tử khác nhau sẽ nằm ở các nhóm khác nhau.

*Ví dụ điển hình:* Tại 1 huyện nào đó cần bố trí 3 cụm, đặt tên số cụm gọi là số *K*, mỗi cụm 1 trường cấp 3. Cần chọn ví trí thế nào để đặt địa điểm của trường sao cho học sinh đến điểm học ngắn nhất.

Kỹ thuật phân cụm dữ liệu có thể ứng dụng trong nhiều lĩnh vực thực tế như:

* Marketing: Xác định các nhóm khách hàng (khách hàng tiềm năng, khách hàng giá trị, phân loại và dự đoán hành vi khách hàng…) sử dụng sản phẩm hay dịch vụ của công ty để giúp công ty có chiến lược kinh doanh hiệu quả hơn
* Biology: Phận nhóm động vật và thực vật dựa vào các thuộc tính của chúng
* Libraries:  Theo dõi độc giả, sách, dự đoán nhu cầu của độc giả…
* Insurance, Finance: Phân nhóm các đối tượng sử dụng bảo hiểm và các dịch vụ tài chính, dự đoán xu hướng (Trend) của khách hàng, phát hiện gian lận tài chính (Identifying Frauds)
* WWW:  Phân loại tài liệu (Document Classification); phân loại người dùng web (Clustering Weblog).

### Giải thuật gom cụm k-Means

1. **Mục đích của thuật toán**

Cho tập dữ liệu *X* và số nhóm *k* cho trước, giải thuật *k-Means* tìm cách phân *m* phần tử dữ liệu vào *k* nhóm (mỗi phần tử chỉ thuộc một nhóm) sao cho tổng khoảng cách của các phần tử đến tâm của nhóm mà nó thuộc về là nhỏ nhất.

1. **Trình bày thuật toán**

Các bước chính của giải thuật được thực hiện như sau:

*B1. Khởi động ngẫu nhiên k tâm của k nhóm. Một tâm điểm là một điểm, có tọa độ*

*B2. Mỗi phần tử được gán cho tâm gần nhất với nó dựa vào khoảng cách.*

*B3. Cập nhật lại các tâm của k nhóm, mỗi tâm là giá trị trung bình của các phần tử trong nhóm của nó*

*B4. Lặp lại bước B2, B3 cho đến khi không còn sự thay đổi nhóm nào của các phần tử.*

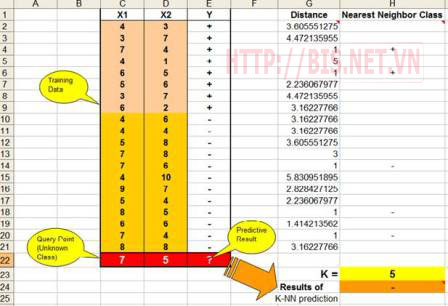
**c. Bài toán ví dụ và lời giải**

Giả sử ta có bảng dữ liệu gồm 2 thuộc tính như bảng 7.1

# Bảng 7.1 – Bảng dữ liệu minh họa thuật toán kMeans

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | STT | A1 | A2 | | 1 | 5.40 | 3.40 | | 2 | 7.90 | 3.80 | | 3 | 5.80 | 2.70 | | 4 | 6.20 | 2.90 | | 5 | 4.40 | 2.90 | | 6 | *5.80* | *4.00* | | 7 | 6.10 | 2.80 | | 8 | 4.30 | 3.00 | | 9 | 6.70 | 3.00 | | 10 | *6.20* | *3.40* | |  |

So sánh K-NN và K-Means?



*Thuật toán:*

***Input:***

Áp dụng giải thuật k-Means lên tập dữ liệu này với *k*=2

Tâm khởi động ngẫu nhiên là: tâm số 1 có tọa độ C1 (6.20, 3.40); tâm số 1 có tọa độ: C2 (5.80, 2.70)

***Output:*** *tìm tọa độ tâm chính xác để tổng khoảng cách của tất cả đối tượng đến tâm nhỏ nhất.*

**Lần lặp 1**: Gán nhóm cho các phần tử bằng cách tính khoảng cách Ơ-clid từ phần tử đến các tâm và chọn nhóm có tâm gần nhất, với *i=1...10.* Nguyên tắc gán cụm: So sánh:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | A1 | A2 |  |  | Nhóm gần nhất |
| 1 | 5.40 | 3.40 | 0.64 | 0.65 | 1 |
| 2 | 7.90 | 3.80 | 3.05 | 5.62 | 1 |
| 3 | 5.80 | 2.70 | 0.65 | 0 | 2 |
| 4 | 6.20 | 2.90 | 0.25 | 0.2 | 2 |
| 5 | 4.40 | 2.90 | 3.49 | 2 | 2 |
| 6 | 5.80 | 4.00 | 0.52 | 1.69 | 1 |
| 7 | 6.10 | 2.80 | 0.37 | 0.1 | 2 |
| 8 | 4.30 | 3.00 | 3.77 | 2.34 | 2 |
| 9 | 6.70 | 3.00 | 0.41 | 0.9 | 1 |
| 10 | 6.20 | 3.40 | 0 | 0.65 | 1 |
| *C1* | *6.40* | *3.52* |  |  |  |
| *C2* | *5.36* | *2.86* |  |  |  |

Tính lại tâm cho các nhóm:

**Lần lặp 2:**

C1 (6.4, 3.52); C2 (5.36, 2.86)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | A1 | A2 |  |  | Nhóm gần nhất |
| 1 | 5.40 | 3.40 | 1.0144 | 0.2932 | 2 |
| 2 | 7.90 | 3.80 | 2.3284 | 7.3916 | 1 |
| 3 | 5.80 | 2.70 | 1.0324 | 0.2192 | 2 |
| 4 | 6.20 | 2.90 | 0.4244 | 0.7072 | 1 |
| 5 | 4.40 | 2.90 | 4.3844 | 0.9232 | 2 |
| 6 | 5.80 | 4.00 | 0.5904 | 1.4932 | 1 |
| 7 | 6.10 | 2.80 | 0.6084 | 0.5512 | 2 |
| 8 | 4.30 | 3.00 | 4.6804 | 1.1432 | 2 |
| 9 | 6.70 | 3.00 | 0.3604 | 1.8152 | 1 |
| 10 | 6.20 | 3.40 | 0.0544 | 0.9972 | 1 |

Tính lại tâm cho các nhóm:

**Lần lặp 3**: C1 (6.56, 3.42); C2 (5.2, 2.9)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | A1 | A2 |  |  | Nhóm gần nhất |
| 1 | 5.40 | 3.40 | 1.346 | 0.2336 | 2 |
| 2 | 7.90 | 3.80 | 1.94 | 7.9956 | 1 |
| 3 | 5.80 | 2.70 | 1.096 | 0.4276 | 2 |
| 4 | 6.20 | 2.90 | 0.4 | 1.0036 | 1 |
| 5 | 4.40 | 2.90 | 4.936 | 0.6436 | 2 |
| 6 | 5.80 | 4.00 | 0.914 | 1.4416 | 1 |
| 7 | 6.10 | 2.80 | 0.596 | 0.8356 | 1 |
| 8 | 4.30 | 3.00 | 5.284 | 0.8116 | 2 |
| 9 | 6.70 | 3.00 | 0.196 | 2.2516 | 1 |
| 10 | 6.20 | 3.40 | 0.13 | 1.1936 | 1 |

Tính lại tâm cho các nhóm

**Lần lặp thứ 4**: C1 (6.48, 3.32); C2 (4.975, 3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i | A1 | A2 |  |  | Nhóm gần nhất |
| 1 | 5.40 | 3.40 | 1.1728 | 0.340625 | 2 |
| 2 | 7.90 | 3.80 | 2.2468 | 9.195625 | 1 |
| 3 | 5.80 | 2.70 | 0.8468 | 0.770625 | 2 |
| 4 | 6.20 | 2.90 | 0.2548 | 1.510625 | 1 |
| 5 | 4.40 | 2.90 | 4.5028 | 0.340625 | 2 |
| 6 | 5.80 | 4.00 | 0.9248 | 1.680625 | 1 |
| 7 | 6.10 | 2.80 | 0.4148 | 1.305625 | 1 |
| 8 | 4.30 | 3.00 | 4.8548 | 0.455625 | 2 |
| 9 | 6.70 | 3.00 | 0.1508 | 2.975625 | 1 |
| 10 | 6.20 | 3.40 | 0.0848 | 1.660625 | 1 |

Tính lại tâm cho các nhóm:

Tâm không thay đổi nên giải thuật dừng. Kết thúc.

### Cài đặt thuật toán với ngôn ngữ R

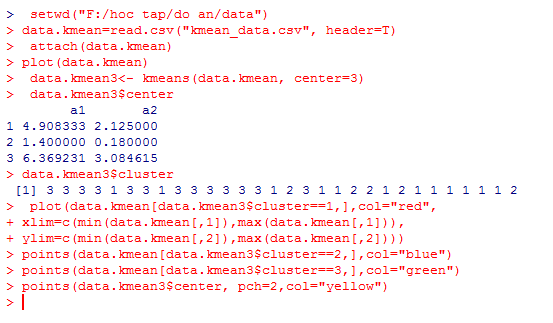
a. Cài đặt thuật toán với dữ liệu cho trước

Giả sử có 1 file dữ liệu excel cho trước có tên là kmean\_data.csv như trong bảng 7.2:

# Bảng 7.2 – File dữ liệu minh họa thuật toán kMeans

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | a1 | a2 | | 5.4 | 3.4 | | 7.9 | 3.8 | | 5.8 | 2.7 | | 6.2 | 2.9 | | 4.4 | 2.9 | | 5.8 | 4 | | 6.1 | 2.8 | | 4.3 | 3 | | 6.7 | 3 | | 6.2 | 3.4 | | |  |  | | --- | --- | | a1 | a2 | | 7 | 3.2 | | 7.2 | 3.2 | | 6.3 | 2.9 | | 5.8 | 2.8 | | 4.8 | 3.1 | | 1.7 | 0.2 | | 6.4 | 2 | | 5.1 | 1.9 | | 4.3 | 1.3 | | 1.4 | 0.2 | | |  |  | | --- | --- | | a1 | a2 | | 1.2 | 0.2 | | 4 | 1.3 | | 1.1 | 0.1 | | 5.2 | 2.3 | | 5.4 | 2.3 | | 4.7 | 1.4 | | 6 | 1.8 | | 5.6 | 1.8 | | 5.1 | 2.4 | | 1.6 | 0.2 | |

Ta tiến hành giải thuật gom cụm kmeans trong R như sau:



# Hình 7.1 – Đồ thị phân cụm k-means với tập dữ liệu cho trước

Kết quả ta thu được hình vẽ như hình 7.1:

*b. Cài đặt thuật toán với bộ dữ liệu chuẩn Iris*

* Cài đặt giải thuật kMeans sử dụng hàm *cclust()*nằm trong gói thư viện *cclust*

Hàm *cclust()* có các tham số như sau:

*cclust (x, center, iter.max=100, verbose=FALSE, dist=”euclidean”, method=”kmeans”,…)*

Giải thích các tham số:

+ *x* là tập dữ liệu (ma trận) không chứa nhãn phân loại

+ *centers* là số nhóm

+ *iter.max* là số bước lặp tối đa

+ *verbose* cho phép hiển thị thông tin trong quá trình xây dựng mô hình

+ *dist* là độ đo khoảng cách euclidean hoặc manhattan

+ *method* là phương pháp gom nhóm có thể là kmeans, hardcl hay neuralgas

…: các tham số còn lại có thể dùng đến

Kết quả trả về của hàm là đối tượng có chứa các trường như:

+ *center*: số lượng nhóm

+ *initcenters*: tâm khởi tạo ngẫu nhiên

+ *cluster*: vector chỉ số nhóm của dữ liệu

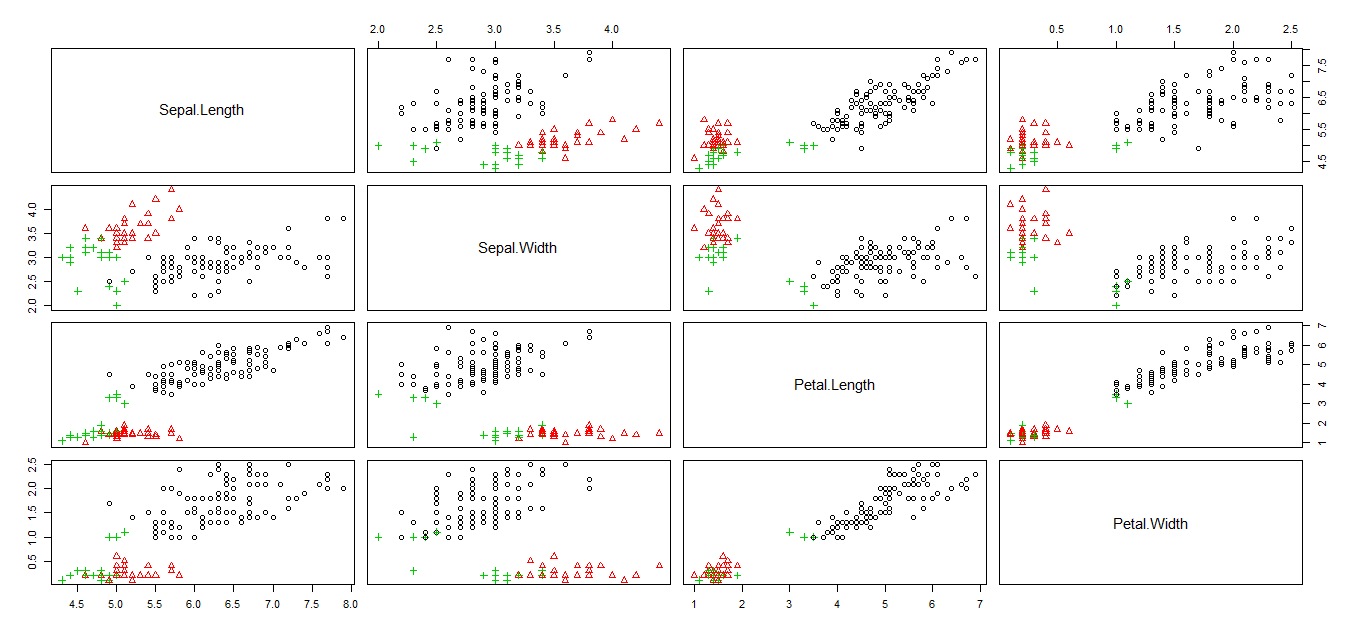
+ *size*: kích thước của các nhóm

+ *iter*: số lần lặp để có kết quả

Chương trình được viết như sau:



Kết quả phân cụm được hiển thị trên ma trận scatterplot 2 chiều như hình 7.2



# Hình 7.2. Ma trận scatterplot 2 chiều cho kết quả gom cụm của k*-*Mean sử dụng hàm cclust

### Kết luận

Giải thuật gom cụm kMeans có một số ưu điểm và nhược điểm như sau:

*Ưu điểm:* Đơn giản*.* Cho kết quả dễ hiểu

*Nhược điểm:*

* Số nhóm k luôn phải được xác định trước
* Kết quả phụ thuộc vào tham số k
* Khả năng chịu đựng nhiễu không tốt (do ảnh hưởng của các phần tử cá biệt)
* Việc khởi tạo phần tử trung tâm của nhóm ban đầu ảnh hưởng tới sự phân chia đối tượng vào nhóm trong trường hợp dữ liệu không lớn
* Không xác định được rõ ràng vùng của nhóm, cùng 1 đối tượng, nó có thể được đưa vào nhóm này hoặc nhóm khác khi dung lượng dữ liệu thay đổi.
* Điều kiện khởi tạo có ảnh hưởng lớn đến kết quả. Điều kiện khởi tạo khác nhau có thể cho ra kết quả phân vùng nhóm khác nhau.
* Không xác định được mức độ ảnh hưởng của thuộc tính đến quá trình tạo nhóm.

Mặc dù có những khuyết điểm trên nhưng trong thực tế giải thuật gom cụm k-Means được sử dụng rất phổ biến do tính đơn giản và dễ hiểu của nó.

Hướng khắc phục:

* Với dữ liệu nhỏ, thuật toán có thể có những hạn chế. Vì vậy, để khắc phục được những hạn chế trên, nên sử dụng thuật toán kMeans trong trường hợp dữ liệu lớn
* Với những hạn chế về việc phân nhóm, ta có thể dùng phương pháp xác định trung tuyến, trung vị thay vì xác định mean

Hướng phát triển:

* Phát triển giải thuật trong trường hợp các kiểu dữ liệu phức tạp
* Làm tăng tốc độ xử lí
* Xử lí vấn đề các tham số đầu vào của giải thuật
* Diễn dịch kết quả sinh ra
* Phương pháp kiểm chứng chất lượng mô hình

**BÀI TẬP CHƯƠNG 7**

Trong WEKA CÓ TỆP DỮ LIỆU VỀ CÁC LOẠI HOA IRIS. Thực hiện mở tệp chạy chức năng phân cụm, báo cáo kết quả (In danh sach tóm tắt+Đánh giá dùng ma trận nhầm lẫn.

PHẦN IV. HỌC TĂNG CƯỜNG VÀ ỨNG DỤNG

**CHƯƠNG 8**

**HỌC TĂNG CƯỜNG**

**8.1. Các giải thuật tiến hóa**

# Trong trí tuệ nhân tạo nói chung và trong học máy nói riêng, một trong các khuynh hướng mang tính chủ đạo là thông minh hóa máy móc bằng cách phỏng sinh học. Điều này thể hiện bởi việc phỏng não người, ta có lĩnh vực mạng nơ ron; học máy ta có các thuật toán dựa theo hoạt động và tư duy của người và tính toán tối ưu dựa vào thuyết tiến hóa. Việc nghiên cứu các mô hình tiến hóa cho phép chúng ta đưa ra các thuật toán học mô phỏng tự nhiên như: thuật toán di truyền trong CNTT (*GA: Genetic Algorithm*), tối ưu đàn kiến (*ACO: Ant Colony Optimization*), tối ưu bầy đàn (*PSO: Particle Swam Optimization*), hệ miễn dịch nhân tạo (*AIS Artificial Immune System)*, thuật toán luyện thép (*SA: Steel Annealing*) v.v… Các thuật toán phỏng sinh học này không chỉ áp dụng để tạo các ra các chương trình phần mềm để đưa vào mô phỏng như trong phần mềm *MATLAB,* các phần mềm mã nguồn mở, các phần mềm xử lý ảnh, xử lý tiếng nói, xử lý ngôn ngữ tự nhiên khác mà còn nhúng vào trong các thiết bị điều khiển, nhất là trong các thiết bị điều khiển thông minh.

Trong phạm vi chương trình này, giới thiệu giải thuật điển hình, phổ biến, đó là giải thuật di truyền *GA*, dựa trên học thuyết di truyền của Darwin. Thuật toán tối ưu đàn kiến (*ACO: Ant Colony Optimization*) dựa trên đặc trưng đánh dấu đường đi của kiến bằng cách tạo mùi đặc biệt và được sử dụng trong học máy như một tham số tăng cường trên bài toán tìm đường đi tối ưu (xem thêm tài liệu [3]. Bài toán tối ưu đàn kiến cũng dựa trên thuật toán tối ưu (tìm đường đi ngắn nhất của người bán hàng du lịch (*Travelling Sale Man: TSM)*.

## 8.1.1. Giải thuật di truyền

## 8.1.1.1 Các khái niệm, thuật ngữ của giải thuật di truyền dùng trong kỹ thuật tính toán

*a) Khái niệm về giải thuật di truyền.*

Giải thuật di truyền (*Genetic Algorithm: GA)* là kỹ thuật chung giúp giải quyết vấn đề-bàitoán bằng cách mô phỏng sự tiến hóa của con người hay của sinh vật nói chung (dựa trên thuyết tiến hóa muôn loài của Darwin) trong điều kiện qui định sẵn của môi trường. Mục tiêu của GA không đưa ra lời giải chính xác mà đưa ra *lời giải tương đối tối ưu* *(hợp lý)*.Mục tiêu của GA được khái quát như sau:

* Trừu tượng hoá và mô phỏng quá trình thích nghi trong hệ thống tự nhiên.
* Thiết kế phần mềm, chương trình mô phỏng, nhằm duy trì các cơ chế quan trọng của hệ thống tự nhiên.

1. *Các thuật ngữ của giải thuật di truyền dùng trong tính toán (CNTT)*

Trong học máy, sử dụng một số thuật ngữ của ngành di truyền học như: nhiễm sắc thể *(*NST), (tiếng Anh: *Chromosomes)*, quầnthể (*Population*), Gen... NST được tạo thành từ các Gen (được biểu diễn bằng một chuỗi các Gen). Mỗi Gen mang một số đặc trưng và có vị trí nhất định, và là phần cơ sở nhất trong NST. Mỗi NST sẽ biểu diễn một lời giải của bài toán. Bảng dưới đây cho biết những ý nghĩa về thuật ngữ và tham số cơ bản của sinh học được chuyển đổi hay mã hóa cho phù hợp với tính toán trong CNTT.

**Các thuật ngữ về dữ liệu**

* *Gen.* Trong sinh học, Gen là phần tử cơ bản để tạo nên NST. Ví dụ, Người da đen, da trắng hay da vàng… khác nhau từ các phần tử cơ bản đó là từ gen. Gen trong tính toán *(Computing)-* CNTT được xác định là các phần tử của một số *(Numer)*. Lý do có lẽ rất đơn giản: tính toán hiển nhiên dựa trên số. Số trong *Computing* được tạo nên từ các hệ đếm với mỗi con số (ứng với gen) của từng hệ đếm. Như vậy, mỗi con số (hay chữ số) trong một số gọi là gen. Con số tiếng Anh gọi là: *digit; một số ví dụ (2001)* tiếng Anh là: *number*
* Bảng dưới đây, mô tả bốn thuật ngữ cơ bản nhất tương ứng từ sinh học, chuyển nghĩa sang tính toán (CNTT).

|  |  |  |
| --- | --- | --- |
| **STT** | **Sinh học** | **Công nghệ Thông tin** |
| 1 | Gen | Con số trong hệ đếm (*digit) của* Hệ đếm: Nhị phân, Bát phân, Hecxa, Thập phân… |
| 2 | Nhiễm sắc thể (*Chromosome*)  hay cá thể (*Individual*) | Tập hợp *n* các con số, gọi là độ lớn của NST. Ví dụ, Gen nhị phân với *n=5* có NST tương ứng là: (01100). |
| 3 | Quần thể (*Population)*: nhóm cá thể thực hiện việc tiến hóa. | Tập hợp nhiều NST. Ví dụ, có *m=3* NST: [011001, 00000, 11111]. *m* gọi là độ lớn của quần thể. |
| 4 | Thế hệ (*Generation*): một vòng đời trong tiến hóa. | Một vòng lặp của thuật toán GA để tạo thực hiện một vòng đời tiến hóa. |

## 8.1.1.2. Các toán tử trong giải thuật di truyền

### Toán tử sinh sản

Toán tử sinh sản gồm hai quá trình: sinh sản (phép tái sinh), chọn lọc (phép chọn).

* **Phép tái sinh:** là quá trình cácNSTđược tạo nên trên cơ sở độ thích nghi cho trước.

Độ thíchnghi được mô tả bằng một hàm, gán giá trị thực, tương ứng với mỗi NST trong quần thể. Quá trình này được mô tả như sau:

* **Phép chọn lọc (tự nhiên):** là quá trình loại bỏ cácNSTkém (không hợp lý) trong quần thể. Quá trình nàyđược mô tả như sau:
* Sắp xếp quần thể theo thứ tự mức độ thích nghi giảm dần.
* Loại bỏ các NST kém ở cuối dãy. Giữ lại cáccá thể tốt nhất.

### Toán tử lai hay toán tử ghép chéo

*Ghép chéo (Cross-Solver) hay còn gọi là phép lai ghép hay đơn giản chỉ là lai* là quá trình tạoNST của một thế hệ mới trên cơ sở cácNST cha-mẹ (thế hệ trước) bằng cách ghép một đoạn NST của cha với 1 đoạn NST của mẹ với nhau. Quy ước: độ lớn của NST thế hệ con bằng độ lớn NST mẹ và bằng độ lớn NST của cha (để đơn giản). Toán tử ghép chéo được gán với một xác suất *pc*. Quá trình được mô tả như sau:

* Chọn ngẫu nhiên cặp NST (cha-mẹ) trong quần thể. Giả sử, NST cha-mẹ cũng như của quần thể có cùng độ lớn *m* như nhau.
* Tạo một số ngẫu nhiên trong khoảng từ 1 tới *m-1* (gọi là điểm ghép chéo). Điểm ghép chéo chia NST của cha-mẹ thành hai chuỗi con có độ dài *m1, m2;* với *m1+ m2=m.*  Đưa NST mới vào quần thể quần thể mới.

### Toán tử đột biến

*Đột biến (Multation)* là hiện tượngNSTcon mang một số đặc tính không có của NST của cả cha và của mẹ.

* + Chọn ngẫu nhiên một NST trong quần thể;
  + Tạo một số ngẫu nhiên *k* trong khoảng từ 1 tới *m,1 ≤ k ≤ m* ;
  + Thay đổi bit thứ *k*. Đưa NST vào quần thể để tham gia quá trình tiến hoá ở thế hệ tiếp theo.

## 8.1.1.3. Giải thuật di truyền

* Giải thuật di truyền dạng bước.

***Input.*** Giả sử cho tệp CSDL

***Output:***

***Bước 1****:* Bước khởi tạo. Chọn loại Gen. Xác định kích thước *n* của một NST*. Khởi tạo* một quần thể ban đầu gồm *m* nhiễm sắc thể *(m:*độ lớn hay kích thước của một quần thể)*.*

***Bước 2****: Xác định giá trị mục tiêu cho từng NST tương ứng.*

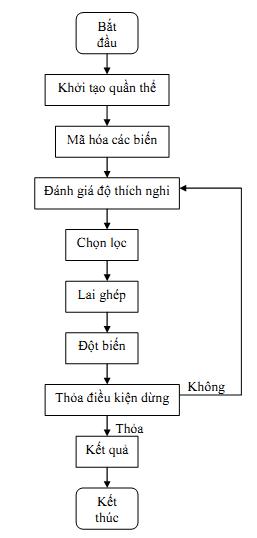
***Bước 3****: Tạo các NST mới dựa trên các toán tử di truyền.*

***Bước 4****: Xác định hàm mục tiêu cho các NST mới và đưa vào quần thể.*

***Bước 5****: Loại bớt các NST có độ thích nghi thấp.*

***Bước 6****: Kiểm tra thỏa mãn điều kiện dừng. Nếu điều kiện đúng, lấy ra NST tốt nhất, giải thuật dừng lại; ngược lại, quay về bước 3.*

* Giải thuật di truyền dạng lưu đồ



*Hình 5.1 : Giải di truyền dạng lưu đồ*

* Giải thuật di truyền dạng mã giả

*Bắt đầu*

*t=0; Khởi tạo P(t)*

*Tính độ thích nghi cho các cá thể thuộc P(t);*

*Khi (điều kiện dừng chưa thỏa) lặp*

*t = t + 1;*

*Chọn lọc P(t);*

*Lai ghép P(t);*

*Đột biến P(t);*

*Hết lặp*

*Kết thúc*

* Điều kiện dừng. Thường chọn một trong hai cách để dừng thuật toán:

*Cách 1*: Cho một số *t* là số thế hệ, tương ứng với số lần lặp. Ưu điểm: dễ tưởng tượng. Nhược điểm: Kết quả có thể chưa đạt mong muốn hoặc quá thừa các bước

*Cách 2*: Kiểm tra đầu ra mong muốn. Khi kết quả đầu ra thực tế gần bằng nhau thì dừng.

***8.2****.* ***Các bài toán và ví dụ ứng dụng***

*Bài toán.* Tìm giá trị lớn nhất (cực đại) của hàm

*y=f(x)=*

với*x* nguyên trong khoảng [0; 31] với các gen dạng nhị phân, kích thước của quần thể *m=4;* Điểm trích mẫu lai của bố mẹ tùy chọn. Điều kiện dừng: giá trị ra (*yi)* chênh nhau không đáng kể.

Theo các tài liệu thống kê, trung bình: xác suất lai ghép: *pc=0,7*, và xác suất đột biến *pm=0,001.*

*Phân tích.*1) Gọi một số nguyên *x* là một NST (hay cá thể). Gen các con số của hệ nhị phân tức *G.* Cần xác định độ lớn của NST, *n*. Ta có .

Hàm hợp lý ở đây là hàm mục tiêu, trong đó *x* là NST hay cá thể (trong bài tóan cụ thể này, nó là một số) cần tìm để cho hàm: *y=f(x)=* là giá trị cực đại. Để ý rằng đây không phải là nhãn để học theo nó. Nó có tác dụng để tự đánh giá một các thể (được hiểu là chọn lọc) trong một quần thể chọn trước.

1. Điểm trích mẫu lai của bố mẹ tùy chọn; Giá trị khởi tạo, tùy chọn được cho ở bảng 8.1
2. Để dễ hình dung, ta vễ đồ thị *y=(2x-x2 /16)* theo bảng biến thiến với các giá trị đặc biệt *xi =[0, 16, 32]* và các giá trị ứng với các NST *xi =[2, 9, 19, 24]* ở bảng sau:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *0* | *2* | *9* | *16* | *19* | *24* | *32* |  |
| *y=(2x-x2 /16)*  =*2x(1-x/32)* | 0 | 3.75 | 12.94 | 16  (Max) | 15.44 | 12 | 0 |  |

Đồ thị cho ở cột cuối cùng của bảng 8.1. Giá trị cực đại theo phương pháp giải tích là *x=16,* bằng cách đạo hàm hàm và cho bằng *0*: *y’=(2x-x2/16)’=2-x/8=0; x=16, y=16* là để tham khảo, đối chiếu với phương pháp GA mà thôi, nó không phải là tín hiệu chỉ đạo.

**Bảng 8.1. Bảng quần thể ngẫu nhiên ban đầu của nhiễm sắc thể**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *i*  *(1)* | (*xi)2*  (*2)* | (*xi)10*  *(3)* | *f(xi)*  *=(2x-x2 /16)*  *(4)* | *g(xi)=*  *(5)* | *e(xi)*  *=m\* g(xi)*  *(6)* | *nx*  *(7)* |  |
| *1* | 00010 | 2 | 3.75 | 0.09 | 0.4 | 0 |
| *2* | 01001 | 9 | 12.94 | 0.29 | 1,1 | 1 |
| *3* | 10011 | 19 | 15.44 | 0.35 | 1.5 | 2 |
| *4* | 11000 | 24 | 12.00 | 0.27 | 1.0 | 1 |
|  |  |  | 44.13 | 1 | 4 |  |

### 

***Bước 1.*** Từ phân tích trên ta có, với Gen nhị phân, độ lớn của NST *n=5;* độ lớn của quần thể chọn *m=4.* Bốn NST chọn giá trị ngẫu nhiên, cho ở cột thứ 2, giá trị thập phân tương ức cho ở cột thứ 3.

***Bước 2.*** Mục đích bước 2 là thực hiện phép chọn lọc. Để chọn lọc ta tiến hành đánh giá chất lượng của từng các thể (NST) trong quần thể bằng các công doạn sau:

*Bước 2.1.* Tính *f(xi)=(2x-x216)* cho mỗi cá theerr, kết quả ở cột 4.

*Bước 2.2.* Tính độ (hay tỷ số) phù hợp (*Fitness Ratio, %)* tương đối của từng cá thể trong một quần thể, ký hiệu là *gi(x)=eval(vi)*. Độ phù hợp ở đây được hiểu là tỷ trọng của mỗi NST thể hiện bằng hàm trong tổng số giá trị thích nghi (hay tổng các giá trị trọng số) của toàn bộ quần thể đã cho:

*, %* (5.2)

với *f (xi)* là hàm mục tiêu hay hàm hợp lý và tổng giá trị thích nghi của quần thể

Kết quả, cho ở cột 5.

*Bước 2.3.* Xây dụng hàm chọn lọc từ công đoạn tính bằng cách nhân với số NST (*m)* được hàm ; kết quả ở cột 6 trong bảng. Đây chỉ là thủ thuật để loại bỏ. Nhờ hàm ta lấy hàm nguyên của ; tức tính: *nx =int(;* trong đó *int(* cho giá trị nguyên bằng cách làm tròn, với cận trên, kết quả ở cột 7 trong bảng. Thủ tục trên mang tính toán học hóa cho việc chọn lọc. Bằng cách làm tròn, ta loại cá thể *nx =int(* nhỏ nhất.

Người ta còn dùng một kỹ thuật lựa chọn khác là: Vòng tròn lựa chọn (Roulette Wheel Selection): Lựa chọ dự theo bánh xe Roulette) để đánh giá. Thực vậy, nếu coi tổng ở cột thứ 5 trong bảng bằng *1=100%* như một vòng tròn, thì mỗi cá thể chiếm một lượng phần tram tương ứng trong vong tròn. Lượng phần tram này tương ứng với xác xuất của cá thể đó được chọn. Người ta quay bánh xe Roulette, sau nhiều lần, chắc chắn các thể nào có xắc suất thấp nhất sẽ xẩy ra ít nhất, và bị loại. Từ đó, phương pháp lựa chọn này còn gọi là phương pháp lựa chọn theo xác suất. (xem thêm ví dụ tiếp sau)

|  |  |
| --- | --- |
|  |  |
| 0.09=9% *=p1* | 9%  35%  29%  27% |
| 0.29=29%= *p2* |
| 0.35=35%*=p3* |
| 0.27=27%*=p4* |

***Bước 3.* Bước lai.** Sau khi loại một hoặc một số c thể kém chất lượng, ta phải có hai công đoạn chuẩn bị: *Bước 3.1 và Bước 3.2*

*Bước 3.1.* Bổ sung cá thể cho quần thể (Bảng 8.2) để dẩm bảo độ lớn *m=4.* Cách chọn khá đơn giản: chọn thêm 1 cá thể bất kỳ giống các cá thể còn lại, Ví dụ chọn *(19)10 =(10011)2.* Kết quả ta có cột 2 Bảng 8.2.

*Bước 3.2.* Tiến hành xây “tổ ấm” cho quần thể mới bằng cách chọn một quần thể “khác giới”, ví dụ như cột 3, Bảng 8.2.

*Bước 3.3.* Chuẩn bị lai. Chọn điểm lai

*Bước 3.* Lai. Lấy một phần của cha+ phần còn lại của mẹ, Kết quả như cột 7 và 8, Bảng 8.2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *i*  *(1)* | (*xi)2*  *P1*  (*2)* | (*xi)10*  *P1*  (3*)* | *(gm)2*  (4*)* | *(gm)10*  (5*)* | Chọn  điểm lai  (6*)* | *(P2)2*  (7*)* | *(P2)10*  (8*)* | Đột biến  *(P2)2*  (9) | Đột biến  *(P2)10*  *(10)* |
| *1* | *100* . 11 | 19 | 010 . *01* | 9 | 3 | *100 01* | *17* | *100 01* | *17* |
| *2* | *010* . 01 | 9 | 100 . *11* | 19 | 3 | *010 11* | *11* | *011 11* | *15* |
| *3* | *1* . 0011 | 19 | 1 . 1000 | 24 | 1 | *1 1000* | *24* | *11000* | *24* |
| *4* | 1 . 1000 | 24 | 1 . 0011 | 19 | 1 | *1 0011* | *19* | *10011* | *19* |

***Bước 4.* Bước đột biến.** Đột biến là sự biến đổi Gen (là một chữ số trong một số) của thế hệ con, sao cho gen đó không giống gen của bố-mẹ ở thế hệ trước. Ví dụ, trong nhiễm sắc thể thứ 2, Bảng 8.2, gen thứ 3, kể từ trái sang, gen của bố-mẹ đều là gen “*0*”, kết quả lai cho thế hệ con cũng nhận được gen *“0”.* Khi đó, giá trị mới *(15)10*  gần tới điểm cực đại hơn giá trị cũ là *(11)10.* Trường hợp tổng quát, đột biến có thể tốt hơn hoặc xấu đi. Trong thực hành, nếu đột biến tốt hơn thì tiến hành. Lý do, đột biến trong thuyết tiến hóa và thực tế, có khoảng (0,1-0.5)%.

***Bước 5.* Đánh giá độ hợp lý (hay thích nghi).** Thủ tục lạp lại như bảng 8.1. Kế quả mới được cho ở Bảng 8.3.

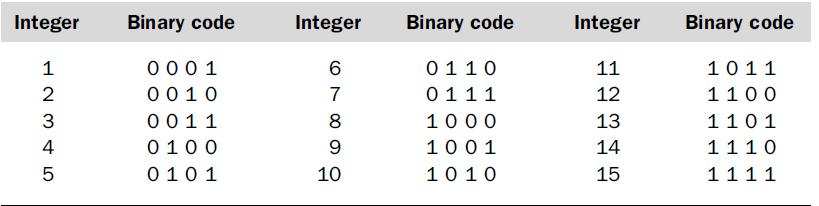
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i*  *(1)* | (*xi)2*  (*2)* | (*xi)10*  *(3)* | *f(xi)*  *=(2x-x2 /16)*  *(4)* | *g(xi)=*  *(5)* | *e(xi)*  *=m\* g(xi)*  *(6)* | *nx*  *(7)* |
| *1* | *100 01* | *17* | 15.9375 | 0.268704 | 1.074816 | 1 |
| *2* | *011 11* | *15* | 15.9375 | 0.268704 | 1.074816 | 1 |
| *3* | *11000* | *24* | 12 | 0.202318 | 0.809273 | 1 |
| *4* | *10011* | *19* | 15.4375 | 0.260274 | 1.041096 | 1 |
|  |  |  | 59.3125 | 1 |  |  |

***Bước 6.* Kiểm tra điều kiện dừng.** Xét *nx* ta thầy đã đồng nhất. có thể dừng thuật toán.

## Ví dụ 2 về giải thuật di truyền

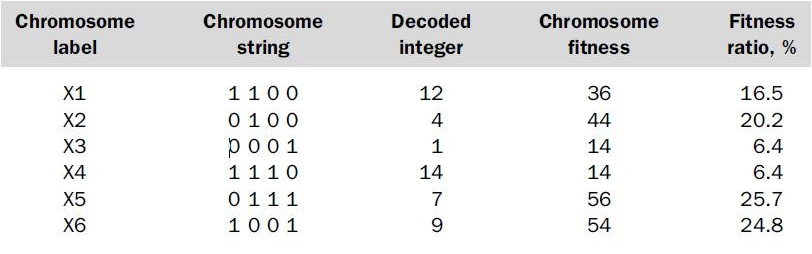
### Ví dụ giải thuật di truyền với hàm một biến

**Bài toán**: Tìm giá trị lớn nhất của hàm*(15x-x2)*với*x*trong khoảng [0;15]. Chúng ta cóthể giả định *x* chỉ nhận giá trị nguyên, do đó NST có thể được xây dựng với các gen:



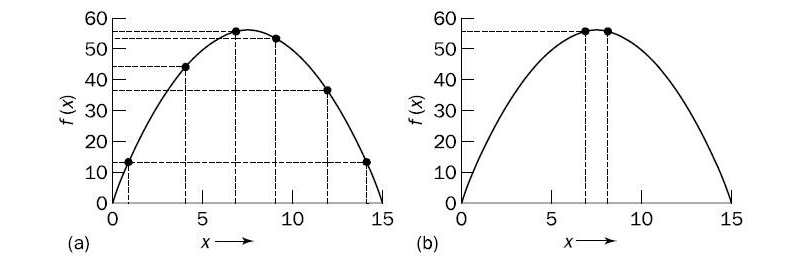
Giả sử kích thước của quần thể NST*N= 6*. Theo các tài liệu thống kê dược, trung bình: xác suất lai ghép *pc* *= 0,7*, và các đột biến *pm* *= 0,001*. Hàm *f(x) = 15x-x2* của GA tạo quần thể NST ban đầu bằng cách điền các chuỗi 4-bit với những giá trị ngẫu nhiên 1 và 0. Quần thể ban đầu như trong trên.

**Bảng 5.2. Bảng quần thể ngẫu nhiên ban đầu của nhiễm sắc thể**



Một vấn đề khó khăn trong tính toán là một quần thể có hàng ngàn nhiễm sắc thể.

Bước tiếp theo là tính toán sự phù hợp của mỗi NST riêng lẻ. Các kết quả cũng được thể hiện trong Bảng 5.2. Sự tương thích trung bình của quần thể ban đầu là 36. Để cải thiện nó, quần thể ban đầu được thay đổi bằng cách sử dụng lựa chọn, chéo và đột biến, toán tử di truyền. Trong chọn lọc tự nhiên, chỉ có các loài thích hợp nhất có thể sống sót, giống, và do đó truyền gen cho thế hệ tiếp theo. GA sử dụng một cách tiếp cận tương tự, nhưng không giống như bản chất, quy mô quần thể NST không thay đổi so một thế hệ kế tiếp.



**Hình 5.2. Hàm huấn luyện và phân bố nhiễm sắc thể**

*(a)Sự phân bố của nhiễm sắc thể ban đầu;(b) Sự phân bố của nhiễm sắc thể sau huấn luyện*

Làm thế nào chúng ta có thể duy trì kích thước của các hằng số, và đồng thời cải thiện sự tương thích trung bình của nó?



**Hình 5.3. Vòng tròn lựa chọn (Roulette Wheel Selection)**

Cột cuối cùng trong Bảng 5.2 cho thấy tỷ lệ tương thích NST của cá nhân với tổng thể của quần thể. Tỷ lệ này xác định NST được chọn để giao phối. Như vậy, các NST *X5* và *X6* có cơ hội được chọn bằng nhau, trong khi NST *X3* và *X4* có xác suất được chọn rất thấp. Kết quả là, sự tương thích trung bình của NSTcải thiện từ một thế hệ tiếp theo.

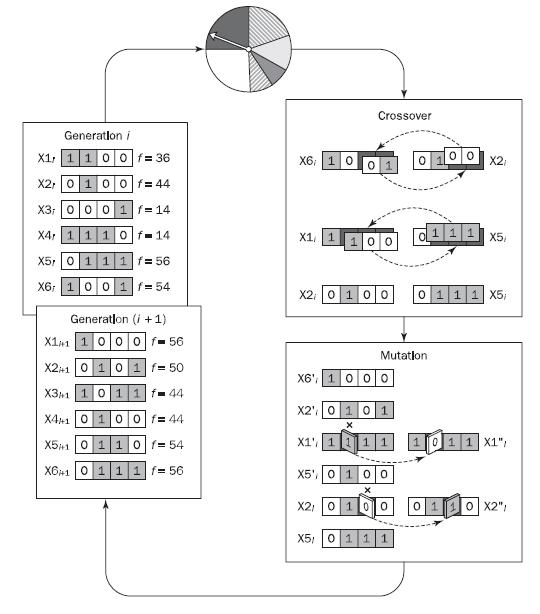
Một trong những lựa chọn kỹ thuật thường được sử dụng NST là lựa chọn bánh xe roulette (Goldberg, 1989; Davis, 1991). *Hình 5.4.1* minh họa ví dụ của chúng ta. Như bạn có thể thấy, mỗi NSTđược đưa ra một lát của một bánh xe tròn. Các khu vực của các slice trong các bánh xe bằng với tỷ lệ NST tương thích (xem Bảng 5.4). Ví dụ, các NST và *X5, X6* (NSTphù hợp nhất) chiếm diện tích lớn nhất, trong khi cácNST *X3* và *X4* (phù hợpnhất) có phân đoạn nhỏ hơn nhiều trong bánh xe. Để chọn một NST cho lai, một số ngẫu nhiên được tạo ra trong khoảng [0; 100], và NST có đoạn kéo dài số ngẫu nhiên được chọn. Nó cũng giống như quay một bánh xe tròn nơi mỗi NST có một phân khúc trên các bánh xe tỷ lệ với sự tương thích của mình. Các bánh xe tròn được chia, và khi mũi tên đi kèm với phần còn lại trên một trong các phân đoạn, tương ứng NST được chọn.

Trong ví dụ, chúng ta chọn một quần thể ban đầu của sáu nhiễm sắc thể. Vì vậy, để lập quần thể cùng trong thế hệ tiếp theo, các đường tròn sẽ được tách sáu lần. Hai lần đầu tiên có thể chọn NST *X6* và *X2* đến trở thành cha mẹ, cặp thứ hai của lần tiếp theo có thể chọn NST *X1* và *X5*, và hai lượt cuối cùng có thể chọnNST *X2* và *X5*.

Khi một cặp NST cha mẹ được chọn, các toán tử chéo được áp dụng.

*Làm thế nào để lai (hay ghép chéo)?*

Đầu tiên, các nhà ghép chéo chọn ngẫu nhiên một điểm giao nhau nơi hai NST cha mẹ khác nhau, và sau đó trao đổi các phần NST sau điểm đó. Kết quả là, hai đứa con mới được tạo ra. Ví dụ, các NST *X6* và *X2* có thể vượt qua sau khi các gen thứ hai trong mỗi để sản xuất hai con, như thể hiện trong *hình 5.3*.



**Hình 5.4. Kết quả thế hệ lai các cặp NST được lựa chọn**

Nếu một cặp NST không vượt qua, sau đó NST nhân bản, con được tạo ra như là bản sao chính xác của mỗi cặp bố mẹ. Ví dụ, các NST mẹ *X2* và *X5* có thể không vượt qua. Thay vào đó, họ tạo ra thế hệ lai là bản sao chính xác của cặp NST *(hình 5.3)*.

Một giá trị 0,7 cho xác suất lai chéo thường cho kết quả tốt. Sau khi lựa chọn, sự tương thích trung bình của quần thể NST đã được cải thiện.

*Đột biến đại diện cho những gì?*

Đột biến là sự kiện hiếm trong tự nhiên, đại diện cho một sự thay đổi trong gen. Nó có thể cải thiện đáng kể thường cho kết quả tốt. Vì vậy, đột biến được sử dụng ở nhiều nơi. Hà Lan giới thiệu đột biến như một nền điều hành (Hà Lan, 1975). Vai trò của nó là tìm kiếm theo thuật toán tối ưu. Chuỗi các lựa chọn và hoạt động chéo có thể dừng tại bất kỳ vị trí nào trong các giải pháp. Dưới điều kiện như vậy, tất cả NST giống nhau, do đó các tập huấn luyện trung bình của quần thể không được cải thiện. Tuy nhiên, các giải pháp có thể tối ưu, hay đúng hơn là tối ưu cục bộ, vì các thuật toán tìm kiếm không thể tiến hành thêm nữa. Đột biến tương đương với một phép tìm kiếm ngẫu nhiên.

*Làm thế nào để công việc điều hành đột biến?*

Lựa chọn ngẫu nhiên một nhiễm sắc thể. Ví dụ, NST *X10* có thể đột biến ở gen thứ hai của nó, và các NST *X2* trong gen thứ ba (hình 5.4). Đột biến có thể xảy ra trong bất kỳ gen nào trong NST với một xác suất nhất định. Xác suất đột biến khá nhỏ trong tự nhiên, thường nằm trong khoảng giữa 0,001 và 0,01. Trong ví dụ này, bài toán chỉ có một biến rất dễ dàng để đại diện. Chúng ta sẽ thử tìm 'đỉnh' chức năng của hai biến.

**8.3. Bàn luận**

**1)** Thuật toán GA là giải thuật tiến giải bài toán tối ưu, thế hệ sau, có khả năng tốt hơn thế hệ trước bằng cách khéo (nghệ thuật) lựa chọn trong tự nhiên. Việc cải thiện thông qua 3 phép: chọ lựa, phép lai, phép đột biến. Giải thuật này thực hiện việc học theo tang cường. Hàm hợp lý gần như là hàm mục tiêu để so sánh cho việc lựa chọn, nó không hoàn toàn là một nhẫn. Vì vậy học theo cách này, từng bước các cá thể được tang về hướng đích

2) Đích trong nhiều trường hợp do chúng ta tự đặt

3) Giải thuật chứa đựng nhiều yếu tố ngẫu nhiên gồm

- Chọn giá trị của của bước khởi tạo: ngẫu nhiên;

- Chọn cặp để tạo cặp: ngẫu nhiên

- Điểm lai: ngẫu nhiên

- Đột biến hay không do ta chọn: ngẫu nhiên (ngoài ra cho gen nào đột biến cũng ngẫu nhiên

Từ đó thuật toán chứa đựng nhiều yếu tố ngẫu nhiên, nên một bài toán có nhiều lựa chọn khác nhau, dẫn tới kết quả có thể cũng khác nhau.

1. Trong ví dụ vừa xét, ta chỉ tiến hóa một biến *x.* Nhiều bài toán được ứng dụng cho tiến hóa nhiều biến khác nhau
2. Đây là bài toán tối ưu. Tối ưu này dùng cách học tăng cường. Cũng bài toán tối ưu, masuy học SVM cũng tối ưu dựa trên nhãn đầu ra {1, -1} thuộc bài toán học có tín hiệu chỉ đạo với kỹ thuật quy hoạch tối ưu toàn phương.
3. Ví dụ vừa nêu chỉ thể hiện phương pháp. Nhiều bài toán không mô tả toán học thì, GA là lợi thế để giải các bài toán đó.

**CÂU HỎI VÀ BÀI TẬP**

1. Thế nào là nhiễm sắc thể? Cách biểu diễn nhiễm sắc thể
2. Các toán tử sử dụng trong giải thuật di truyền
3. Trình bày thuật toán di truyền
4. Cho hàm hợp lý *(-x2 +15x)* với *x* trong khoảng *[0;15],* giả định *x* lấy giá trị nguyên.
   1. Xác định kích thước của nhiễm sắc thể với gen được mã hóa nhị phân *[0, 1];*
   2. Chỉ dùng toán tử lai ghép, tìm giá trị cực đại

CHƯƠNG 9

**KẾT HỢP CÁC BỘ HỌC VÀ CÁC HƯỚNG PHÁT TRIỂN**

Đã có nhiều thuật toán học khác nhau được giới thiệu trong các chương trước.

Lê ddifnh Việt: Các thuật toán đa học, Lê Công Tuấn Minh:

*Nhóm học Surpervising Learning Rules*

*1) K-NN*

*2) Thuật toán Quinlan*

*3) Thuật toán Độ lộn xộn*

*4) Nguyễn Đức Anh (không có mic)Nguyễn Đức Tắng ID3*

*5) C4.5*

*6) C5.0*

*7) ILA*

*9) Bayes*

*10) SVM*

*11) Thuật toán Hồi quy (Tuyến tính, Hồi quy Bội, Hồi quy Logistic, Hồi quy phi tuyến)*

*Nhóm không có tín hiệu chỉ đạo: Nguyễ Vawn Thành*

*11) K-Means*

*Học tăng cường*

*12) GA*

Tuy nhiên, khi áp dụng thì không có thuật toán nào tạo nên được một bộ học thực sự chính xác hơn hẳn các phương pháp khác. Một cách tiếp cận có hiệu quả để tăng độ chính xác là kết hợp các bộ học nhận dạng với nhau để được một bộ tốt hơn hay còn gọi là học tập thể, hoặc học kết hợp.

Chương này giới thiệu các kỹ thuật thông dụng trong kết hợp các bộ học: bỏ phiếu, tạo các bộ học cơ sở bằng cách nhặt dữ liệu theo gói (*Bagging*) và nhặt định hướng (*Boosting*), rừng ngẫu nhiên, kiến trúc bậc thang.

**9.1. Học kết hợp**

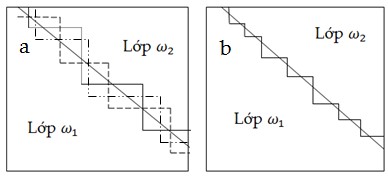
Với mỗi bài toán phân lớp hoặc hồi quy cụ thể, người ta thường có nhiều thuật toán học để khi xây dựng bộ học. Cùng một thuật toán, có thể chọn các tham số khác nhau hoặc sử dụng tập dữ liệu huấn luyện khác nhau, nên kết quả sẽ cho các bộ nhận dạng khác nhau.

*Ví dụ*. Cùng dùng mạng MLP nhưng ta chọn số lớp (tầng) ẩn/nơron lớp (tầng) ẩn hay số đặc trưng khác nhau thì kết quả huấn luyện cho các bộ nhận dạng có chất lượng khác nhau. Cùng là phương pháp học *k-NN* nhưng với mỗi *k* cho ta một bộ nhận dạng và rất khó khẳng định được tham số *k* nào là tối ưu.

Những thuật toán cho cùng lớp bài toán thường tuân theo luật “*không có bữa trưa miễn phí (no Free Lunch Theory)*”, tức là không có thuật toán tốt hơn hẳn các thuật toán khác; mà mỗi thuật toán có ưu/nhược điểm riêng. Khi thực hiện nhận dạng, mỗi bộ huấn luyện theo thuật toán tương ứng có những lớp mẫu nhận dạng tốt và yếu khác nhau. Việc kết hợp một cách hợp lý các bộ nhận dạng có thể cho ta bộ nhận dạng mới có nhiều ưu điểm hơn và ta gọi ***là học tập thể hay học kết hợp*** (*Ensemble Learning*).

Một bộ phân lớp được kết hợp từ 3 bộ phân lớp cơ sở khác nhau theo cách bỏ phiếu được minh họa trong hình 10.1. Hãy tưởng tượng hai lớp 1 và 2 có biên là đường thẳng nhưng hai bộ phân lớp cơ sở tạo nên các biên tương ứng là đường liền và các đường đứt như trong hình 9.1a, còn hình 9.1b chỉ ra rằng bộ phân lớp kết hợp theo hình thức bỏ phiếu cho ta biên quyết định gần với biên thực hơn.





**Hình 10.1. a) Biên quyết định của mỗi bộ phân lớp. b) Biên kết hợp ba bộ**

Như vậy, mỗi cách học cho ta một bộ nhận dạng cơ sở; nhờ kết hợp các bộ nhận dạng thành phần có được mà ta có một bộ nhận dạng tốt hơn. Các bộ nhận dạng cơ sở này thường được xây dựng theo các tiếp cận sau đây:

1. *Dùng các thuật toán huấn luyện khác nhau*. Các thuật toán này sử dụng các giả thiết khác nhau về dữ liệu, các bộ học có thể phụ thuộc tham số hoặc không. Khi kết hợp các bộ học, ta được giải phóng khỏi các giả thiết áp đặt này.
2. *Mỗi bộ học dùng cách chọn đặc trưng khác nhau*. Chẳng hạn chúng ta dùng thuật toán SVM để phân biệt chữ viết tay nhưng cách chọn đặc trưng có thể là nội dung ảnh hay qua phép biến đổi nào đó.
3. *Có thể sử dụng cùng một thuật toán nhưng có tham số khác nhau*. Chẳng hạn đều sử dụng thuật toán *k-láng giềng gần nhất* nhưng với *k* khác nhau.
4. *Cùng một thuật toán nhưng sử dụng các tập dữ liệu huấn luyện khác nhau*.

Thông thường thì các bộ nhận dạng được xây dựng theo hai cách tiếp cận đầu có thời gian chạy khác nhau và bộ nhận dạng chính xác hơn thường đòi hỏi thời gian xử lý nhiều hơn.

Khi có các bộ nhận dạng cơ sở, bộ nhận dạng tập thể được kết hợp theo các kiểu tôpô đa dạng để cho ta những bộ mới tốt hơn các bộ thành phần. Trong đó phương thưc kết hợp đơn giản và dễ dùng nhất là phương pháp bỏ phiếu.

### 9.2. Phương pháp bỏ phiếu

Một cách đơn giản để kết hợp các bộ học cơ sở là dùng phương pháp bỏ phiếu nhờ kiến trúc song song, đầu ra được quyết định nhờ kết quả tổng hợp có trọng số của các bộ nhận dạng thành phần. Đối với đối tượng ***x*** cần gán nhãn, nếu mỗi bộ học cơ sở *Ci*cho quyết định **d***i* với trọng số *ý kiến* *wi* tương ứng thì đầu ra của bộ kết hợp đối với mẫu nàyđược tính theo công thức:

(9.1a)

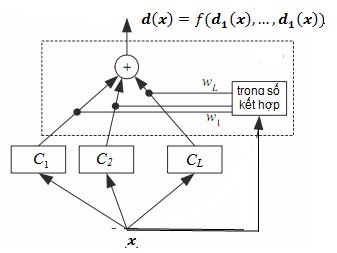
cho bài toán hồi quy, và theo đa số có trọng số của tập:

(9.1b)

cho bài toán phân lớp. Các trọng số có thể chọn bằng nhau. Tổng quát hơn, ta có thể quyết định bằng một hàm tổng hợp phi tuyến *f* nào đó:

(9.2)

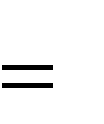
Sơ đồ quyết định tổng quát của quyết định theo hình thức bỏ phiếu được mô tả trong hình 10.2.



**Hình 9.2. Sơ đồ kết hợp các bộ nhận dạng nhờ bỏ phiếu**

Việc huấn luyện các bộ thành phần của bộ học tập thể này có thể sử dụng một trong các phương thức sau:

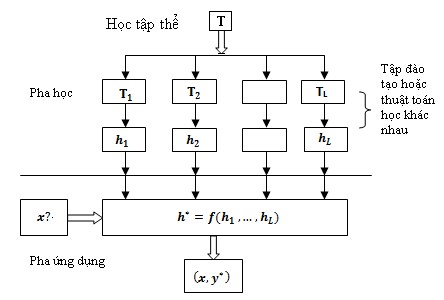
* *L* thuật toán huấn luyện khác nhau.
* Một thuật toán nhưng *L* tập dữ liệu đào tạo hay tham số khác nhau.
* Một thuật toán nhưng dùng tập dữ liệu với tập đặc trưng khác nhau
* Kết hợp các phương thức trên.

 Việc học tập thể *T* bao gồm các quá trình huấn luyện ***T****i* cho bộ học để cho giả thuyết *hi* tương ứng và chúng được kết hợp thành giả thuyết *h*\*. Khi ứng dụng nhận dạng mẫu ***x***, giả thuyết *h*\* sẽ cho ta nhãn **y**\**h*\* (*x*) như minh họa trong hình 9.3.

**Hình**

**10.3.**

Sơ đồ học tập thể của các bộ học



### 9.3. Kỹ thuật tạo và kết hợp bộ nhận dạng cơ sở

Mục này giới thiêu ba kỹ thuật thông dụng để tạo và kết hợp các bộ học cơ sở nhờ dùng một thuật toán học với cách tạo và dùng dữ liệu khác nhau: nhặt theo gói (bagging), nhặt định hướng bao gồm boosting và adaboost, rừng ngẫu nhiên (Random forest).

#### *9.3.1. Nhặt theo gói*

*Nhặt theo gói* (*Bagging*) là phương pháp học tập thể đơn giản và thông dụng nhất, mặt khác, nó giúp giảm phương sai khi dùng cho bài toán hồi quy. Phương pháp này dùng cùng một thuật toán để xây dựng các bộ nhận dạng cơ sở bằng cách lấy ngẫu nhiên từ tập dữ liệu đào tạo một tập dữ liệu cho một bộ học, sau đó kết nối song song các bộ nhận dạng có được và quyết định với trọng số đều. Cụ thể như sau.

 Giả sử ta có tập dữ liệu đào tạo *D*  gồm N mẫu đã gán nhãn . Từ một thuật toán cơ sở (chẳng hạn ID3, K-NN hay mạng nơ ron) ta huấn luyện *L* bộ học trên *L* tập con dữ liệu được lấy ngẫu nhiên của *D*. Để có *L* tập dữ liệu này, ta chọn trước một số *M* < *N* và với mỗi *i* 1,2,...,*L*, lấy ngẫu nhiên phân bố đều từ *D* tập *Di* gồm *M* đối tượng làm dữ liệu đào tạo. Sau khi huấn luyện bằng thuật toán đã chọn trên các tập dữ liệu *Di iL* 1 này ta có các bộ nhận dạng *Ci iL* 1.. Các bộ nhận dạng này được kết hợp theo hình thức bỏ phiếu với trọng số bằng nhau, kiến trúc của bộ học này được mô tả trong hình 10.4.



*N*

*k*

*k*

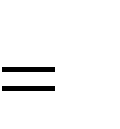
*k*

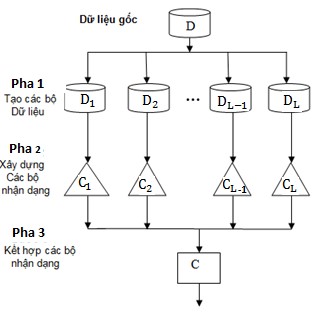
*y*

*x*

1

,





**Hình 9.4. Kiến trúc của hệ học theo phương pháp Bagging**

Thuật toán xây dựng bộ nhận dạng được mô tả trong bảng 9.1.

**Bảng 9.1. Thuật toán Bagging**

|  |
| --- |
| Bước 1. Lặp với i=1,, 2, <, K:  1.1. *Di*  M mẫu đào tạo lấy ngẫu nhiên từ D;  1.2. *hi*  Học từ *Di* theo thuật toán được chọn;  Bước 2. Kết hợp các *hi* theo hình thức bỏ phiếu với trọng số đều. |

Kỹ thuật này có thể áp dụng cho cả bài toán phân lớp và hồi quy .Dietterich và Bakiri (1991) thử nghiệm phân lớp cho các tập dữ liệu thực và dữ liệu chuẩn trên UCI với 100 bộ phân lớp cơ sở đã giảm 10% lỗi cho dữ liệu thực và 6% đến 42% lỗi cho dữ liệu trên UCI.

#### *9.3.2. Nhặt định hướng*

Mục đích của phương pháp nhặt định hướng là để cải tiến độ chính xác của thuật toán phân lớp đã cho. Trong phương pháp này người ta xây dựng các *bộ học yếu*, trong đó tập dữ liệu huấn luyện dùng cho bộ học sau được lấy định hướng dựa trên lỗi của các bộ học trước.

Dưới đây giới thiệu hai kỹ thuật thông dụng của cách tiếp cận này là *boosting* và *adaboost*

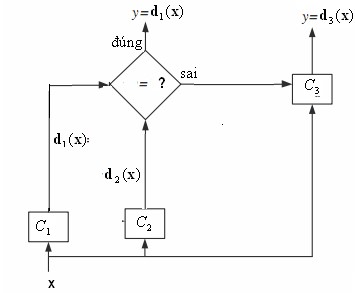
#### Boosting

Đối với bài toán phân chia thành hai lớp, phương pháp này tạo ra 3 bộ phân lớp yếu theo qúa trình sau đây.

Trước tiên người ta lấy ngẫu nhiên tập *D*1 gồm *n*1 (*N*) mẫu trong tập dữ liệu huấn luyện *D*. Sau đó huấn luyện bộ phân lớp *C*1 từ tập dữ liệu *D*1. *C*1 là bộ học *yếu*, tức là tỷ lệ lỗi cao (nếu độ chính xác cao thì mức độ cải tiến sẽ thấp, tuy nhiên nó có thể có tỷ lệ lỗi thấp trong tập dữ liệu huấn luyện).

Tập dữ liệu huấn luyện *D*2 được xây dựng dựa trên thông tin định hướng của bộ phân lớp *C*1, trong đó một nửa được *C*1 phân lớp đúng còn một nửa phân lớp sai. Cụ thể như sau: lấy ngẫu nhiên, *r* (0,1) nếu *r* 1/2 thì lấy từng mẫu một trong *D* chưa thuộc *D*2 cho *C*1 đoán nhận cho đến lúc sai thì bổ sung vào *D*2, ngược lại nếu *r* 1/2 thì lấy từng mẫu một còn trong D chưa thuộc *D*2 cho *C*1 đoán nhận cho đến lúc đúng thì bổ sung vào *D*2. Thủ tục lặp theo cách này cho đến khi *D*2 có *n*2 dữ liệu gồm một nửa *C*1 nhận dạng đúng và một nửa nhận dạng sai. Huấn luyện bằng tập dữ liệu *D*2 ta được bộ phân lớp *C*2.

Tiếp theo ta xây dựng tập dữ liệu huấn luyện *D*3 gồm các dữ liệu mà *C*1 và *C*2 không thống nhất. Ta lần lượt lấy mẫu ngẫu nhiên từ *D*, nếu *C*1 và *C* 2 phân loại không thống nhất thì bổ sung nó vào *D*3, ngược lại thì mẫu này bị bỏ qua. Theo cách này ta xây dựng được *D*3 và dùng để huấn luyện bộ phân lớp *C*3. Bây giờ ba bộ này được kết hợp như sau. Nếu mẫu ***x*** mà kết quả đoán nhận của *C*1 và *C*2 thống nhất thì nhận nhãn chung này, ngược lại, nếu *C*1 và *C*2 không thống nhất thì dùng nhãn đoán nhận của *C*3. Ký hiệu là kết quả đoán nhận của bộ học *Ci* đỗi với mẫu ***x*** thì nhãn ***y*** của nó được đoán nhận theo sơ đồ ở hình 10.5.



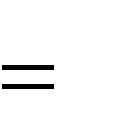
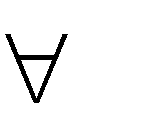
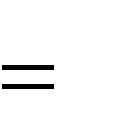
**Hình 9.5. Sơ đồ quyết định sử dụng Boosting**

Freund và Schapire (1996) đề xuất phương pháp nhặt định hướng thích nghi (Adaboost) cho phép thiết kế tiếp tục các bộ học cho đến khi chất lượng được cải tiến thực sự.

#### Adaboost

Trong Adaboost, mỗi lần xây dựng tập huấn luyện *Di* thì các mẫu **x** *j* trong *D* được lấy với xác suất *pij* . Ban đầu các xác suất này như nhau: *p*1*j* 1 1,..,*N* để

*j*

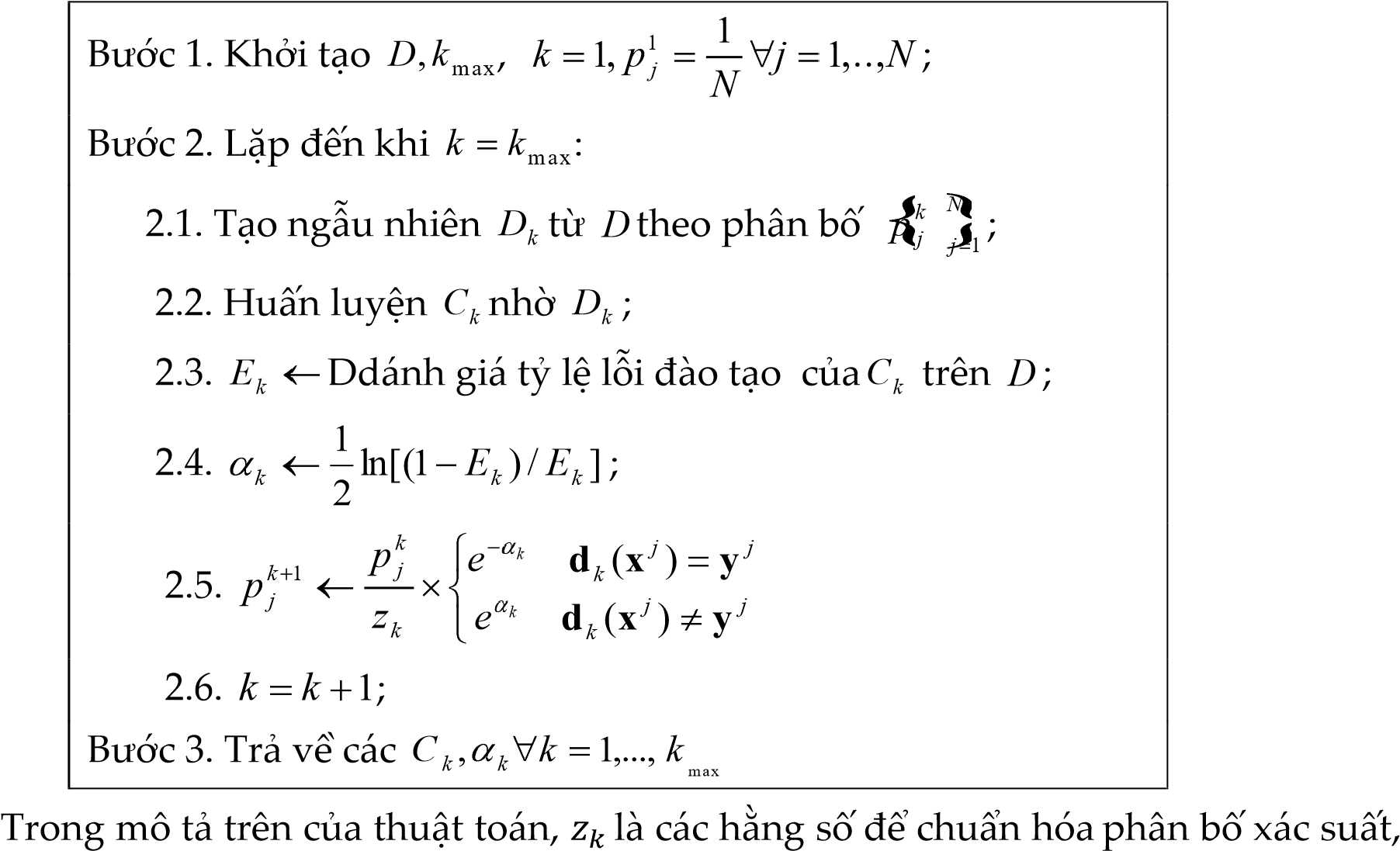


xây dựng *D*1 và dùng nó để huấn luyện bộ phân lớp *C*1 . Sau mỗi bước lặp các xác suất này tăng dần với các mẫu phân lớp sai và giảm dần với mẫu phân lớp đúng. Giả sử ở bước lặp thứ *k*, ta đã chọn tập đào tạo *Dk* với phân bố xác suất *p kj Nj* 1 trong

*D* và huấn luyện được bộ phân lớp *Ck*, ta tăng xác suất chọn *p kj* 1 *Nj* 1 đối với các mẫu trong *D* mà *Ck* phân lớp sai và giảm xác suất chọn đối với các mẫu phân lớp đúng.

Đến nay có nhiều biến thể của Adaboost được sử dụng, một thể hiện của nó được mô tả trong bảng 9.2.

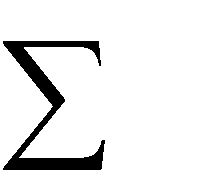
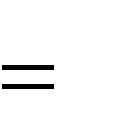
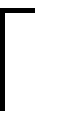
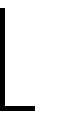
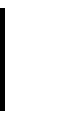
**Bảng 9.2. Một biến thể của Adaboost**



**y** *j*là nhãn đúng của **x** *j* , **d***i*(**x**) là kết quả đoán nhận của bộ học *Ci* đối với mẫu ***x*.** Kết

qủa phân lớp cuối cùng của mẫu ***x*** sẽ là:

*g*(**x**) (10.3)



max

1

)

(

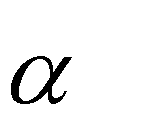
*k*

*k*

*k*

**x**

**d**



trong đó tổng sẽ lấy theo từng nhãn lớp, còn ngoặc vuông được hiểu là lấy nhãn lớp có trọng số tổng hợp lớn nhất.

##### ***9.3.3. Rừng ngẫu nhiên***

Rừng ngẫu nhiên là phương pháp học tập thể rất thích hợp cho xử lý dữ liệu có số chiều lớn, vì vậy đang thu hút nhiều người quan tâm nghiên cứu, áp dụng. Định nghĩa sau đây giải thích tên gọi của phương pháp này.

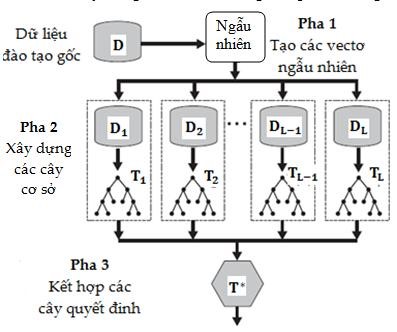
***Định nghĩa.*** Rừng ngẫu nhiên là một bộ nhận dạng bao gồm một tập bộ phân lớp cơ sở dạng cây quyết định được kết hợp theo phương thức bỏ phiếu. Các bộ cơ sở được xây dựng từ các tập con dữ liệu với đặc trưng khác nhau được lấy ngẫu nhiên từ tập dữ liệu đào tạo.

Định nghĩa cho thấy thủ tục xây dựng rừng ngẫu nhiên gồm ba pha:

* *Pha 1:* tạo dữ liệu (tạo vectơ ngẫu nhiên),
* *Pha2:* xây dựng các cây cơ sở,
* *Pha 3:* kết hợp các cây cơ sở theo phương thức bỏ phiếu.

Điểm mới nhất ở đây là pha tạo dữ liệu. Giả sử là tập dữ liệu đào tạo với số chiều *n* lớn, pha tạo vectơ ngẫu nhiên thực hiện như sau.

Chọn trước các số tự nhiên *M* (<*N*) và *m* (<*n*). Để có mỗi tập dữ liệu cho xây đựng cây quyết định , ta chọn ngẫu nhiên *m* đặc trưng trong số *n* đặc trưng của D và lấy ngẫu nhiên *M* đối tượng từ D rồi chiếu nó lên các đặc trưng được chọn này. Việc chọn đặc trưng và lấy dữ liệu từ D gọi là tạo vectơ ngẫu nhiên. Quá trình xây dựng rừng ngẫu nhiên được mô tả trong hình 9.6.

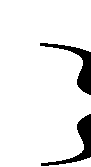
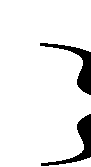


**Hình 9.6. Sơ đồ xây dựng rừng ngẫu nhiên**

*Chọn số đặc trưng m*

Rừng ngẫu nhiên thường áp dụng cho các bài toán phân lớp hoăc hồi quy có số chiều *n* lớn, khi đó số đặc trưng *m* cho mỗi tập dữ liệu cây được chọn nhỏ hơn *n* nhiều. Breiman gợi ý chọn *m* như sau:

Đối với bài toán phân lớp, , trong đó ký hiệu phần nguyên của *a*.

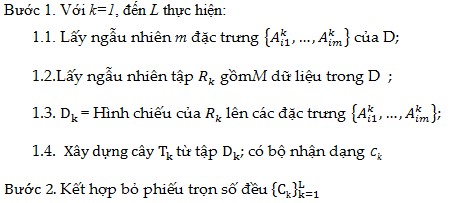


Đối với bài toán hồi quy,

Hiện nay việc chọn số đặc trưng, số đối tượng dữ liệu *M* cho mỗi tập và cách chọn ngẫu nhiên để lấy các đặc trưng thế nào là tốt cho từng lớp bài toán vẫn đang là chủ đề mở và được nhiều người nghiên cứu. Các cây quyết định được kết hợp song song theo phương pháp bỏ phiếu với đầu ra có trọng số đều cho bởi các công thức (9.1a-b). Thuật toán xây dựng rừng ngẫu nhiên được mô tả trong bảng

9.3.

**Bảng 9.3. Thuật toán xây dựng rừng ngẫu nhiên cho phân lớp hoặc hồi quy**



### 9.4. Kiến trúc bậc thang

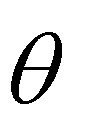
Trong mục 9.3, ta đã xét các bộ phân lớp cơ sở tạo nên nhờ cùng một thuật toán huấn luyện nhưng dùng các tập dữ liệu khác nhau để huấn luyện. Trong thực tế, có thể dùng các thuật toán và cách trích chọn đặc trưng khác nhau để xây dựng các bộ nhận dạng cơ sở. Khi đó các bộ phân lớp này thường có hiệu quả và thời gian chạy khác nhau. Thông thường thì bộ phân lớp có độ chính xác cao sẽ tốn nhiều thời gian chạy hơn. Trong trường hợp đó, kiến trúc bậc thang là phương pháp hiệu quả để kết hợp các bộ phân lớp này.

Ta xét *L* bộ phân lớp, được sắp theo thứ tự tăng dần về độ chính xác: *C*1 , *C* 2 ,..., *C L*. Mỗi bộ phân lớp *Ci* có hàm đánh giá độ tin cậy *wi* và ngường tin cậy sao cho với mẫu ***x*** mà thì nhãn hầu chắc chắn đúng (độ chính xác cao). Khi đó các bộ phân lớp này có thể kết hợp theo kiến trúc bậc thang như trong hình 9.7. Các mẫu được đưa tuần tự cho từng bộ phân lớp bắt đầu từ *C*1 , nếu bộ phân lớp *Ck* đang xét dự đoán nhãn mẫu ***x*** với độ tin cậy lớn hơn ngưỡng thì lấy nhãn **d***k*(**x**) cho nó, ngược lại thì đưa cho bộ *Ck*1 đoán nhận, cuối cùng thì dành cho bộ *CL*. Bộ phân lớp kết hợp theo kiến trúc này thường có độ chính xác cao hơn các bộ thành phần và thời gian chạy trung bình để xử lý một mẫu cũng ít hơn.

*i*



*k*





**Hình 10.7. Kiến trúc bậc thang của L bộ phân lớp**

### 9.5. Kiến trúc lai

Lai MPL với GA: giống kiến trúc bậc thang. GA cho kết quả tối ưu toàn cục, trong khi MPL cho kết quả tối ưu lỗi cục bộ. Ta tiến hành GA trước để tối ưu toàn cục, sau đó cho thuật toán Lan truyền ngược thực hiện tối cục bộ (min hơn) khi đã có tối ưu toàn cục.

#### KẾT LUẬN

Trong học có giám sát để xây dựng bộ phân lớp hoặc hồi quy, nếu dùng một thuật toán và dùng một tập dữ liệu đào tạo thì ta chỉ được một bộ nhận dạng yếu, tức là sai số lớn. Học tập thể là một phương pháp để tăng độ chính xác của bộ nhận dạng. Trong cách tiếp cận này, người ta xây dựng các bộ nhận dạng cơ sở theo các phương thức:

1) dùng các thuật toán huấn luyện khác nhau.

2) dùng một thuật toán nhưng sử dụng các tập dữ liệu đào tạo hay tham số khác nhau,

3) dùng một thuật toán nhưng dùng tập dữ liệu với tập đặc trưng khác nhau.

4) kết hợp các phương thức trên.

Bỏ phiếu là phương pháp kết hợp các bộ nhận dạng đơn giản và thông dụng nhất. Trong sơ đồ này, các bộ nhận dạng cơ sở được kết nối song song và đầu ra tổng hợp thường dùng nhất là dạng tuyến tính, có thể lấy trọng số đều. Phương pháp bỏ phiếu không chỉ tăng độ chính xác mà trong các bài toán hồi quy, nó còn cho kết quả có phương sai thấp hơn các bộ thành phần.

Bagging là phương pháp đơn giản nhất để xây dựng bộ học tập thể , trong đó các bộ nhận dạng thành phận được xây dựng nhờ dùng cùng một thuật toán nhưng dùng các tập dữ liệu đào tạo khác nhau được lấy ngẫu nhiên có hoàn lại các tập dữ liệu gốc.

Các kỹ thuật nhặt theo hướng bao gồm boosting và adaboost khác với bagging ở chỗ phân bố xác suất lấy dữ liệu thay đổi theo thứ tự lấy tập dữ liệu đào tạo của bộ nhận dạng cơ sở. Phương thức kết hợp các bộ cơ sở trong kỹ thuật boosting không theo kiến trúc song song, nhờ đó tiết kiệm thời gian chạy hơn.

Rừng ngẫu nhiên là phương pháp thích hợp cho các bài toán phân lớp hoặc hồi quy với dữ liệu có số chiều cao. Trong đó mỗi bộ nhận dạng cơ sở là một cây quyết định, được xây dựng từ tập dữ liệu có số chiều nhỏ nhờ chiếu một tập dữ liệu lấy ngẫu nhiên từ tập dữ liệu đào tạo gốc lên tập đặc trưng đươc chọn ngẫu nhiên.

Khi các bộ phân lớp được xây dựng nhờ dùng các thuật toán khác nhau, có độ chính xác và thời gian chạy khác nhau thì có thể dùng kiến trúc bậc thang để tăng độ chính xác và giảm thời gian chạy trung bình so với bộ nhận dạng thành phần có độ chính xác cao nhất nhưng thời gian chạy cũng lâu nhất.

#### BÀI TẬP CHƯƠNG 9

1. Giả sử mỗi bộ phân lớp cơ sở đều có xác suất đúng p >1/2, hãy ước lượng xác suất đúng khi bỏ phiếu theo đa số của *L* bộ phân lớp.
2. Hãy đề xuất một lược đồ boosting cho bài toán nhiều lớp.
3. Tìm một lược đồ điều chỉnh xác suất chọn tập đào tạo theo adaboost khác với lược đồ ở mục 10.3.
4. Hãy đề xuất một cách chọn đặc trưng để tạo các tập dữ liệu cho cây quyết định của phương pháp rừng ngẫu nhiên. Giải thích vì sao cách chọn này tốt hơn chọn phân bố đều.
5. Tại sao trong kiến trúc bậc thang ta xếp các bộ phân lớp tăng dần về độ chính xác?

Ôn tập và hướng dẫn viết tiêu luận thu hoạch

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

[1] Hoàng Kiếm (2005), *Giáo trình trí tuệ nhân tạo*, NXB ĐHQG TP Hồ Chí Minh.

[2] Hoàng Kiếm, Đỗ Phúc, Đỗ Văn Nhơn (2007), *Giáo trình các hệ cơ sở tri thức*, NXB ĐHQG TP Hồ Chí Minh.

[3] Hoàng Xuân Huấn (2015), *Giáo trình học máy*, NXB ĐHQG Hà Nội.

[4] Đỗ Thanh Nghị (2012), *Nguyên lý học máy,* NXB ĐH Cần Thơ.

[5] Đỗ Thanh Nghị (2012), *Khai mỏ dữ liệu,* NXB ĐH Cần Thơ.

[6] Từ Minh Phương (2016), *Nhập môn trí tuệ nhân tạo,* NXB Thông tin và Truyền thông

[7] Đinh Mạnh Tường (2002), *Trí tuệ nhân tạo,* NXB Khoa học và Kỹ thuật.

[8] Nguyễn Quang Hoan (2017), *Giáo trình Các hệ cơ sở tri thức,* Học viện Công nghệ Bưu chính Viễn thông, Bộ Thông tin Truyền thông.

[9] Lê Văn Phùng, Quách Xuân Trường (2012). *Khai phá dữ liệu,* NXB Thông tin và Truyền thông,

[10] Nguyễn Như Phong (2008). *Tính toán mềm và ứng dụng.* NXB Khoa học và Kỹ thuật,

**Tiếng Anh**

[11] Fix E , Hodges J. (1952), *Discriminatory Analysis: Small Simple Performance in Technical Report.* 21-49-004, USAF School of Aviation Medicine, Randolph Field. USA.

[12] Wu X. and Kumar V (2009). *Top 10 Algorithms in Data Mining,* Chapmanand Hall/CRC.

[13] E. Alpaydin (2010), *Introduction to Machine Learning*, 2010, Second Edition, MIT Press

[14] T. Mitchell (1997), *Machine Learning*, 1997, McGraw-Hill.

[15] Dawn. E Holmes et al (2006), *Innovation in Machine Learning,* Springer.

[16] Han J. and M. Kamber (2006). *Data Mining -Concept and Technique.* Second Edition, Morgan Kaufmann Publisher, 2006.

[17]

**APPENDIX 1 (PHỤ LỤC 1)**

**8.2.3 A Note on Terminology**

Much of the literature on nearest-neighbor methods and weighted local regression uses a terminology that has arisen from the field of statistical pattern recognition. In reading that literature, it is useful to know the following terms:

* Regression means approximating a real-valued target function.
* Residual is the error á(x) - f (x) in approximating the target function.
* Kernel function is the function of distance that is used to determine the weight of each training example. In other words, the kernel function is the function K such that wi = K(d(xi, x,)).

**2.2.1** Notation Throughout this book, we employ the following terminology when discussing concept learning problems. The set of items over which the concept is defined is called the set of instances, which we denote by X.

In the current example, X is the set of all possible days, each represented by the attributes Sky, AirTemp, Humidity, Wind, Water, and Forecast.

The concept or function to be learned is called the target concept, which we denote by c.

In general, c can be any Boolean-valued function defined over the instances X; that is, c : X + áO, 1). In the current example, the target concept corresponds to the value of the attribute EnjoySport (i.e., c(x) = 1 if EnjoySport = Yes, and c(x) = 0 if EnjoySport = No).

- 0 Given: 0 Instances X: Possible days, each described by the attributes

0 Sky (with possible values Sunny, Cloudy, and Rainy),

0 AirTemp (with values Warm and Cold),

0 Humidity (with values Normal and High),

0 Wind (with values Strong and Weak),

0 Water (with values Warm and Cool), and

0 Forecast (with values Same and Change).

0 Hypotheses H: Each hypothesis is described by a conjunction of constraints on the attributes Sky, AirTemp, Humidity, Wind, Water, and Forecast.

The constraints may be "?" (any value is acceptable), "0 (no value is acceptable), or a specific value. 0 Target concept c: EnjoySport: X + (0,l) 0 Training examples D: Positive and negative examples of the target function (see Table 2.1). 0 Determine: 0 A hypothesis h in H such that h(x) = c(x) for all x in X. **TABLE 2.2** The EnjoySport concept learning task. When learning the target concept, the learner is represented.

**2.3 CONCEPT LEARNING AS SEARCH (page 23. Tom Mitchell)**

Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation. The goal of this search is to find the hypothesis that best fits the training examples. It is important to note that by selecting a hypothesis representation, the designer of the learning algorithm implicitly defines the space of all hypotheses that the program can ever represent and therefore can ever learn. Consider, for example, the instances X and hypotheses H in the EnjoySport learning task.

Example Sky AirTemp Humidity Wind Water Forecast EnjoySport 1 Sunny Warm Normal Strong Warm Same Yes 2 Sunny Warm High Strong Warm Same Yes 3 Rainy Cold High Strong Warm Change No 4 Sunny Warm High Strong Cool Change Yes TABLE 2.1 Positive and negative training examples for the target concept EnjoySport

**2.2.2 The Inductive Learning Hypothesis**

Notice that although the learning task is to determine a hypothesis h identical to the target concept c over the entire set of instances X, the only information available about c is its value over the training examples. Therefore, inductive learning algorithms can at best guarantee that the output hypothesis fits the target concept over the training data. Lacking any further information, our assumption is that the best hypothesis regarding unseen instances is the hypothesis that best fits the observed training data. This is the fundamental assumption of inductive learning, and we will have much more to say about it throughout this book. We state it here informally and will revisit and analyze this assumption more formally and more quantitatively in Chapters 5, 6, and 7. The inductive learning hypothesis. Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

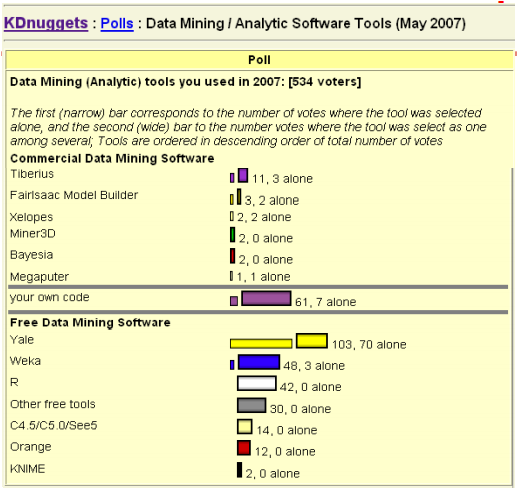
Given that the attribute Sky has three possible values, and that AirTemp, Humidity, Wind, Water, and Forecast each have two possible values, the instance space X contains exactly 3 .2 2 .2 2 .2 = 96 distinct instances. A similar calculation shows that there are 5.4-4 -4 -4.4 = 5 120 syntactically distinct hypotheses within H. Notice, however, that every hypothesis containing one or more "IZI" symbols represents the empty set of instances; that is, it classifies every instance as negative. Therefore, the number of semantically distinct hypotheses is only 1 + (4.3.3.3.3.3) = 973. Our EnjoySport example is a very simple learning task, with a relatively small, finite hypothesis space. Most practical learning tasks involve much larger, sometimes infinite, hypothesis spaces. If we view learning as a search problem, then it is natural that our study of learning algorithms will exa~the different strategies for searching the hypothesis space. We will be particula ly interested in algorithms capable of efficiently searching very large or infinite hypothesis spaces, to find the hypotheses that best fit the training data. 2.3.1 General-to-Specific Ordering of Hypot

**APPENDIX II** **(PHỤ LỤC II)**

# GIỚI THIỆU VỀ WEKA

## 2.1 Giới thiệu tổng quan về phần mềm Weka

Weka là phần mềm được viết bằng ngôn ngữ JAVA do Witten và Frank xây dựng phục vụ lĩnh vực học máy và khai phá dữ liệu. Weka là phần mềm miễn phí được rất nhiều người dùng được xếp hạng như hình 2.1:

****

*Hình 2.1 Biểu diễn số lượng người dùng các phần mềm trong DM[8]*

1. Weka gồm các phương pháp học máy cơ bản phục vụ cho các mục tiêu sau:

* Tiền xử lý dữ liệu, các phương pháp học máy và đánh giá mô hình.
* Sử dụng đồ họa để biểu diễn dữ liệu.
* Là môi trường dùng để so sánh các giải thuật học máy, khai phá dữ liệu.

1. Weka website:

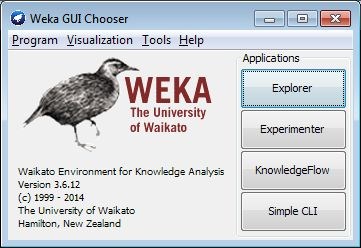
http://www.cs.waikato.ac.nz/ml/weka/​

1. Tài liệu hướng dẫn sử dụng weka:

<http://www.cs.waikato.ac.nz/ml/weka/documentation.html>.

1. Cơ sở dữ liệu:

http://www.cs.waikato.ac.nz/ml/weka/datasets.html​

****  ***Hình 2.2. Giao diện chính của Weka***

Các môi trường chính của phần mềm Weka:

* **Experimenter:** Môi trường cho phép tiến hành các thí nghiệm và thực hiện các kiểm tra thống kê (Statistical Tests) giữa các mô hình học máy.
* **KnowledgeFlow:** Môi trường cho phép tương tác đồ họa kiểu kéo/thả để thiết kế các bước (các thành phần) của một thí nghiệm.
* **Simple CLI:** Giao diện đơn giản kiểu dòng lệnh (như MS-DOS).
* **Explorer:** Môi trường sử dụng các khả năng của Weka để khai phá dữ liệu**.**

**2.2. Định dạng dữ liệu (ARFF)**

ARFF (Attribute Relation File Format) là định dạng dữ liệu chuyên biệt của Weka, tổ chức dữ liệu theo cấu trúc được quy định trước.

Cấu trúc tập tin \*.ARFF gồm 2 phần:

* Head file: chứa khai báo quan hệ, danh sách các thuộc tính (tên, kiểu dữ liệu).
* Data: gồm nhiều dòng, mỗi dòng thể hiện giá trị các thuộc tính cho một mẫu học Learning Pattern).
  + 1. ***Head file***

­ Comments: bắt đầu bằng dấu %

­ Relations: ***@relation <relation­name>***

Trong đó, *<relation­name>:* là một xâu cho biết tên của tập dữ liệu. Nếu một xâu chứa dấu cách thì xâu phải nằm trong dấu ‘?’ hoặc “?”. Xâu không thể bắt đầu bằng các kí tự ‘!’, ‘{’, ‘}’, ‘,’, ‘%’

­ Khai báo kiểu dữ liệu (Data Declaration):

***@attribute <tên thuộc tính/đặc trưng> <kiểu dữ liệu>***

* *<Tên thuộc tính/đặc trưng>:* là xâu bắt đầu bằng chữ cái. Nếu tồn tại dấu cách

trong xâu thì xâu phải nằm trong dấu ‘?’ hoặc “?”.

* *Kiểu dữ liệu:*

+ integer: số nguyên

+ real: số thực

+ numeric: là kiểu dữ liệu số bao gồm số nguyên và số thực.

+ string: là kiểu dữ liệu dạng chuỗi

+ nominal (categorical data): là kiểu dữ liệu danh sách, dữ liệu dạng ngôn ngữ (chữ)

*Ví dụ:*

Đặc trưng hay thuộc tính

@attribute *outlook*: *{sunny, overcast, rainy}*:

Thuộc tính kiểu định danh, đặc trưng đầu vào;

@attribute *play: {yes, no}:*

Thuộc tính phân lớp (mặc định là thuộc tính cuối cùng, đầu ra hay nhãn)

+ date: kiểu dữ liệu thời gian (ngày tháng năm, giờ phút giây)

*Cấu trúc:*

**@attribute <name> date [<date-format>]**

*Ví dụ:* @ATTRIBUTE timestamp DATE "yyyy­MM­dd HH:mm:ss"

+ Relational

*Cấu trúc:*

*@attribute <name> relationa <further attribute definitions>*

*@end <name>*

* + 1. ***Dữ liệu***
* Các giá trị nằm trong phần data phải tuyệt đối theo đúng thông tin sẽ khai báo trong header.
* Dữ liệu thiếu được biểu diễn bằng dấu?
* Dữ liệu quan hệ phải được kèm theo trong dấu ngoặc kép "..."

Ví dụ: MUSK-188,"42,...,30",1. Ví dụ minh họa:

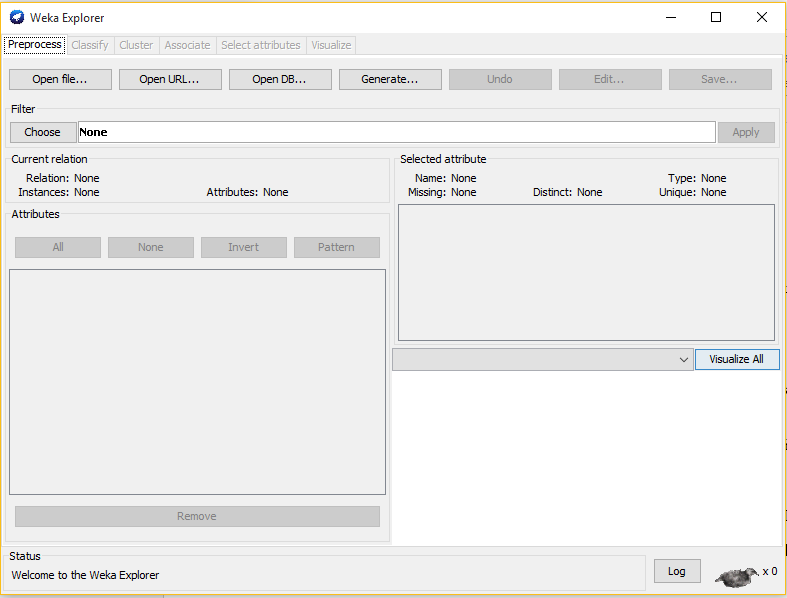
@DATA

*5.1,3.5,1.4,0.2, Iris-setosa*

*4.9,3.0,1.4,0.2, Iris-setosa*

*4.7,3.2,1.3,0.2, Iris-setosa*

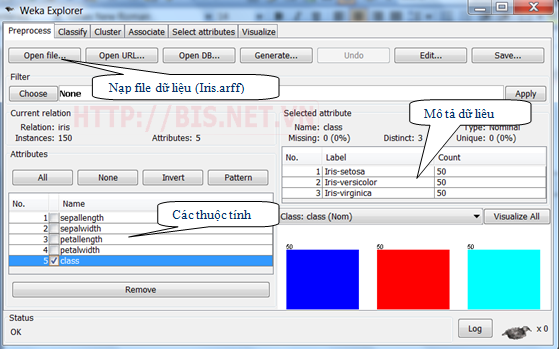
* 1. **Explorer.** Giao diện chính Explorer như hình 2.3



*Hình 2.3 Giao diện chính của Weka Explorer*

1. Các chức năng chính - Môi trường Explorer:
2. Preprocess (tiền xử lý dữ liệu): Chọn và thay đổi dữ liệu.
3. Classify: huấn luyện, kiểm tra các mô hình học máy (phân loại, hồi quy/dự đoán)
4. Phân cụm (Cluster). Học các nhóm từ dữ liệu.
5. Luật kết hợp (Associate): Khám phá các luật kết hợp từ dữ liệu.
6. Lựa chọn thuộc tính (Select Attributes): Lựa chọn các thuộc tính liên quan (quan trọng) nhất của dữ liệu.
7. Trực quan hóa (Visualize): Để hiển thị biểu đồ tương tác hai chiều đối với dữ liệu.
   * 1. ***Tiền xử lý dữ liệu***

Dữ liệu có thể được nhập vào từ một tập tin có khuôn dạng ARFF, CSV. Dữ liệu cũng có thể được đọc vào từ một địa chỉ URL hoặc từ một cơ sở dữ liệu thông qua JDBC.



*Hình 2.4 Giao diện Preprocess trong Weka*

Trong khai phá dữ liệu, xử lý dữ liệu trước khi đưa vào mô hình là cần thiết. Bước này tạo dữ liệu ban đầu (gọi là dữ liệu gốc: original data). Các việc của tiền xử lý dữ liệu gồm:

* Filtering Attributes: Chọn các thuộc tính phù hợp với mô hình.
* Filtering Samples: Lọc các mẫu (Instances, Patterns) dữ liệu cho mô hình.
* Clean data: Làm sạch dữ liệu như xóa bỏ các dữ liệu bất thường (Outlier).
* Transformation: Chuyển đổi dữ liệu cho phù hợp với các mô hình như chuyển đổi dữ liệu từ dữ liệu dạng số (Numeric) qua dữ liệu dạng định danh (Nomial) hay dạng dữ liệu thứ bậc (Ordinal).
* Discretization (rời rạc hóa dữ liệu): Nếu có dữ liệu liên tục nhưng mô hình chỉ áp dụng cho các dữ liệu rời rạc (như luật kết hợp chẳn hạn) phải thực hiện việc rời rạc hóa dữ liệu. Trong Weka, rời rạc hóa bằng cách chọn:

*filters.unsupervised.attribute.Discretize và chọn bins tương ứng*.

### Các bộ phân lớp

1. ***Classifier***

Các bộ phân lớp của WEKA ứng với các mô hình dự đoán các đại lượng kiểu định danh (phân lớp) hoặc các đại lượng kiểu số (hồi quy/dự đoán)

Các kĩ thuật phân lớp được hỗ trợ bởi WEKA:

* Naive Bayes Classifier and Bayesian Networks
* Decision Trees
* Support Vector Machines
* Neural Networks
* Lựa chọn các tùy chọn bộ phân lớp
* Lựa chọn các tùy chọn cho việc kiểm tra (Test Options).
* Use Training Set: bộ phân loại học được sẽ được đánh giá trên tập học
* Supplied Test Set: sử dụng một tập dữ liệu khác (với tập học) để đánh giá.
* Cross-Validation: tập dữ liệu sẽ được chia đều thành *k* tập (*k-folds*) có kích thước xấp xỉ nhau, và bộ phân loại học được sẽ đánh giá bởi phương pháp xác nhận chéo (*Cross-Validation).* Cross-Validation là kỹ thuật chủ yếu được sử dụng trong xây dựng mô hình dự đoán *(Predictive Model)*, trong đó, dữ liệu gốc sẽ được chia thành *n* phần bằng nhau (*n-fold*), và quá trình huấn luyện/thử mô hình *(Train/Test Model)* thực hiện lặp lại *n* lần. Tại mỗi lần *Train/Test Model*, 1 phần dữ liệu dùng để thử (*Test*) và *(n-1)* phần còn lại dùng để huấn luyện *(Train*), kết quả sẽ được lây trung bình cộng sau *n* lần. Người ta đã chứng minh 10-fold Cross –Validation là tối ưu.
* Percentage Split. Chỉ định tỷ lệ phân chia dữ liệu đối với việc đánh giá.
  + Classifier Outputhiển thị các thông tin quan trọng:
  + Run information. Các tùy chọn đối với mô hình học, tên của tập dữ liệu, số lượng các ví dụ, các thuộc tính, và thí nghiệm.
  + Classifier model (full training set). Biểu diễn (dạng text) của bộ phân lớp học được.
  + Predictions on test data. Thông tin chi tiết về các dự đoán của bộ phân lớp đối với tập kiểm tra.
  + Summary. Các thống kê về mức độ chính xác của bộ phân lớp, đối với thí nghiệm đã chọn.
  + Detailed Accuracy by Class. Thông tin chi tiết về mức độ chính xác của bộ phân lớp đối với mỗi lớp.
  + Confusion Matrix. Các thành phần của ma trận này thể hiện số lượng các ví dụ kiểm tra (*Test Instances*) được phân lớp đúng và phân lớp sai.

1. ***Phân cụm (Clustering)***

* Các bộ phân cụm (Cluster Builders) của WEKA tương ứng với các mô hình tìm các nhóm của các ví dụ tương tự đối với một tập dữ liệu.
* Các kĩ thuật phân cum được hỗ trợ bởi Weka:
  + Expectation Maximization (EM)
  + K-Means

1. Các bộ phân cụm có thể hiển thị kết quả và so sánh với cụm (lớp ) thực tế.
2. Lựa chọn một bộ phân cụm (*Cluster Builder*).
3. Lựa chọn chế độ phân cụm (*Cluster Mode*).
   1. Use Training Set. Các cụm học được sẽ được kiểm tra đối với tập học.
   2. Supplied Test Set. Sử dụng tập dữ liệu khác để kiểm tra cụm học được.
   3. Percentage Split. Chỉ định tỷ lệ phân chia tập dữ liệu ban đầu cho việc xây dựng tập kiểm tra.
   4. Classes to Clusters Evaluation. So sánh độ chính xác của các cụm học được đối với các lớp được chỉ định.
4. Store Clusters for Visualization: Lưu các bộ phân lớp trong bộ nhớ, để hiển thị sau đó.
5. Ignore attributes: Chọn các thuộc tính không tham gia quá trình học các cụm.
6. ***Associate Rules (Luật kếp hợp)***
7. Lựa chọn một mô hình (giải thuật) phát hiện luật kết hợp.
8. Associator output hiển thị các thông tin quan trọng.
   1. Run Information. Các tùy chọn đối với mô hình phát hiện luật kết hợp, tên của tập dữ liệu, số lượng các ví dụ, các thuộc tính.
   2. Associator Model (Full Training Set). Biểu diễn (dạng text) của tập các luật kết hợp phát hiện được.
      1. Độ hỗ trợ tối thiểu (Minimum Support)
      2. Độ tin cậy tối thiểu (Minimum Confidence)
      3. Kích thước của các tập mục thường xuyên (large/frequent itemsets)
      4. Liệt kê các luật kết hợp tìm được.
   3. **Giới thiệu bộ dữ liệu *bank.arff* và *Iris.arff* sử dụng trong Weka**

Bộ dữ liệu sử dụng để phân cụm là bộ dữ liệu Bank.arff và Iris.arff của phần mềm Weka.

*Bảng 2.1 Số thuộc tính và đối tượng của các bộ dữ liệu*

|  |  |  |
| --- | --- | --- |
| ***Bộ dữ liệu*** | ***Số thuộc tính*** | ***Số đối tượng*** |
| B bank.arff | 11 | 600 |
| Iris.arff | 5 | 150 |

Dữ liệu về khách ngân hàng (bank.arff) gồm 11 thuộc tính và 600 khách hàng (samples, instances, patterns). Cấu trúc file bank.arff trong bảng 2.2

*Bảng 2.2 Mô tả các thuộc tính cho dữ liệu khách hàng ngân hàng*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Attribute** | **Date Type** | **Explanation** |
| 1 | age | Nominal | Tuổi khách hàng |
| 2 | sex | Nominal | Giới tính |
| 3 | region | Nominal | Khu vực cư trú |
| 4 | income | Nominal | Thu nhập |
| 5 | married | Nominal | Tình trạng gia đình |
| 6 | children | Nominal | Số con |
| 7 | car | Nominal | Có xe hơi? |
| 8 | save\_act | Nominal | Có tài khoản tiết kiệm không? |
| 9 | current\_act | Nominal | Hiện tại có tài khoản không? |
| 10 | mortgage | Nominal | Có thế chấp không? |
| 11 | pep | Nominal | Khách hàng có kế hoạch trả nợ không? |

Dữ liệu (iris.arff) gồm 5 thuộc tính và 150 Samples. Cấu trúc file trong bảng 2.3 như:

*Bảng 2.3 Mô tả các thuộc tính cho dữ liệu về hoa Iris*

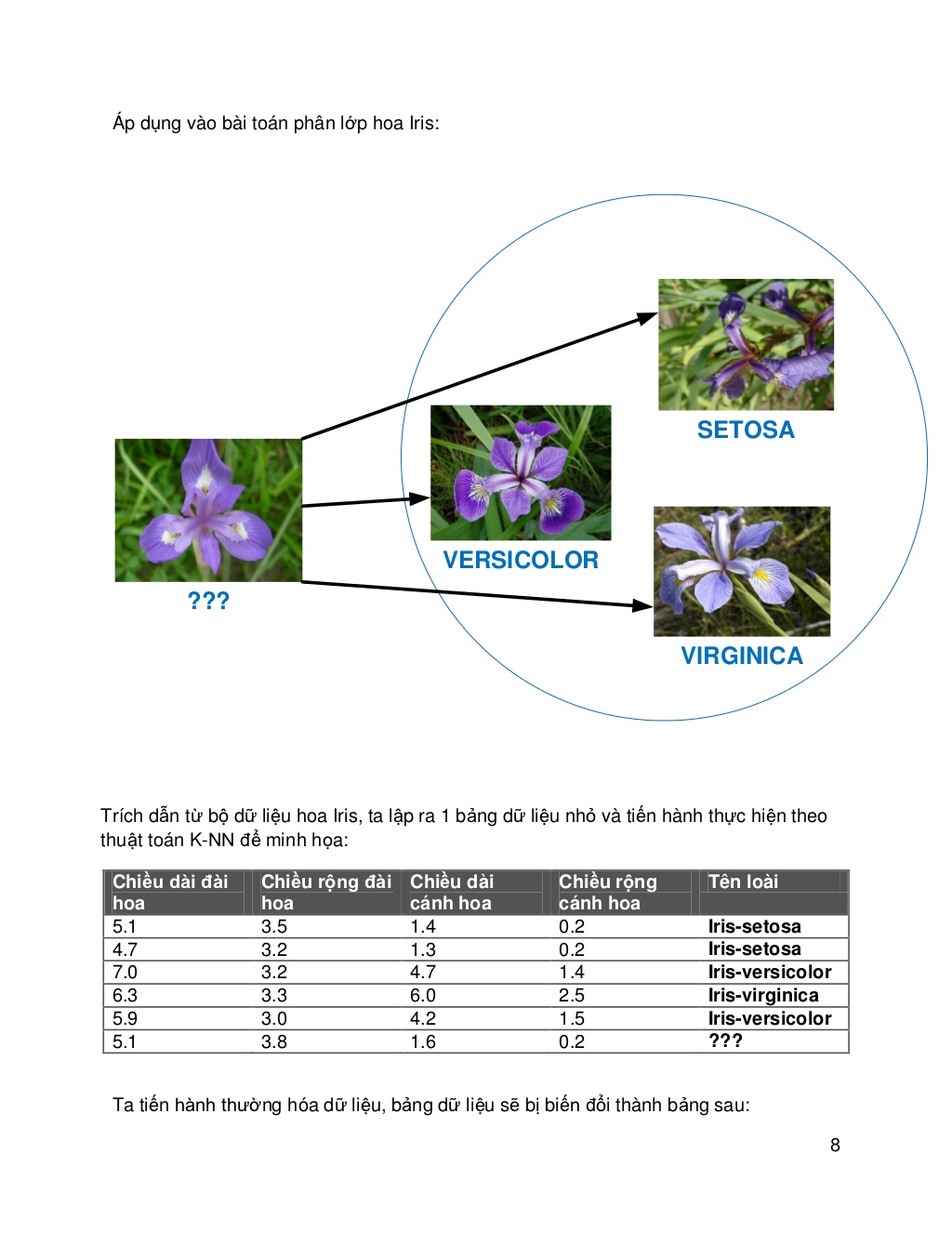
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Attribute** | **Data Type** | **Explanation** |
| 1 | Sepallength | Numeric | Độ dài đài hoa |
| 2 | Petallength | Numeric | Độ dài cánh hoa |
| 3 | Sepalwidth | Numeric | Độ rộng đài hoa |
| 4 | Petalwidth | Numeric | Độ rộng cánh hoa |
| 5 | Class | Nominal | Thuộc tính phân loại hoa (Setosa,Vesicolor và Virginica) |

Trong các chương trình chạy các giải thuật trong chương 3 thì thực hiện rời rạc hóa dữ liệu Iris.arff theo 2 trường hợp: bins=5 và bins=10.

**2.4.1. GIỚI THIỆU VỀ HOA IRIS** **VÀ BỘ DỮ LIỆU IRIS**

*Iris tên tiếng Việt là hoa Diên Vĩ. Tiếng Hy Lạp, "Iris" là Cầu Vồng. Trong thần thoại Hy Lạp, Iris là tên một nữ thần Hy Lạp, là tín sứ của thần Zeus. Nàng thường xuất hiện dưới hình một chiếc cầu vồng. Nàng là người đưa tin trên đỉnh Olympus, mang thông điệp của các vị thần linh từ "con mắt Thiên Đường" xuống cho nhân loại trên trái đất qua vòng cung cầu vồng rực rỡ. Từ Iris còn có nghĩa là "con mắt Thiên Đường" (Theeye of Heaven). Iris có nghĩa khác là tròng đen trong con mắt chúng ta. Điều này ngụ ý rằng, mỗi chúng ta mang trong mình một mảnh của Thiên Đường. Những người đàn ông Hy Lạp thường trồng hoa Diên Vĩ tím trên mộ người phụ nữ mà họ yêu thương để tỏ lòng tôn kính nữ thần Iris, người có sứ mệnh dẫn dắt những linh hồn phụ nữ này đến chốn Thiên Đàng (the Elysian Fields).*

*Diên Vĩ là loài cây lưu niên có thân vươn cao, lá hình lưỡi kiếm và những đóa hoa to nhiều màu sắc với ba cánh và ba đài hoa rũ xuống. Có hơn 300 loài hoa Diên Vĩ khác nhau với các màu xanh da trời nhạt, tím, vàng, trắng, hồng, cam, nâu, đỏ...đa dạng như sắc cầu vồng. Hoa Diên Vĩ được xem như sứ giả mang đến những điềm lành. niềm hy vọng. Ba cánh hoa Diên Vĩ đại diện cho lòng Trung Thành, sự Khôn Ngoan và lòng Dũng Cảm. Hoa Diên Vĩ vàng là biểu tượng của ngọn lửa và niềm đam mê. Hoa Diên Vĩ đã từng được thấy ở sa mạc, đầm lầy hay cả miền Bắc cực Siberia lạnh giá; nhiều nhất vẫn là ở các vùng khí hậu ôn hòa. Hoa Diên Vĩ đã là biểu tượng của hoàng gia và sự che chở của thần thánh suốt hàng thế kỷ trên khắp thế giới. Loài hoa đầy sức thu hút này được rất nhiều người ngưỡng mộ. Các vị vua chúa nước Pháp đã dùng nó làm biểu tượng hoàng gia, gọi nó là Fleur-de-lis. "Fleur-de-lis" có nguồn gốc từ tên "Fleur-de-Louis", sau thời vua Louis VII, năm1147. Theo thời gian, tên đó chuyển thành "Fleur-de-Luce", có nghĩa là hoa của ánh sáng (Flower of Light); đến nay, nó được gọi là "Fleur-de-Lys", hay “Flower of the Lily” (Lily : Hoa Huệ Tây, Loa Kèn, Bách Hợp). Fleur-de-Lis đã là biểu tượng của nước Pháp từ thế kỷ 13. Hoàng gia Pháp trang trí hoa Diên Vĩ trên áo choàng, các đồ vật trong cung điện và trên những bức tường như biểu hiện của sự toàn bích, ánh sáng và cuộc sống. Có nhiều truyền thuyết khác nhau giải thích tại sao hoa Diên Vĩ được chế độ quân chủ Pháp chọn làm biểu tượng. Tương truyền rằng, Clovis, vua nước Pháp triều đại Mêrôvê khi đối mặt với đội quân thiện chiến của Alamanni (Đức) đến xâm chiếm vương quốc mình, ông đã nói với hoàng hậu Clotida rằng ông sẽ theo đạo và chịu rửa tội nếu như Chúa phù hộ ông đánh thắng trận này (trước đó hoàng hậu đã nhiều lần khuyên chồng mình vào đạo nhưng ông vẫn không nghe). Cuối cùng, ông thắng thật và nhận Fleur-de-Lis làm biểu tượng. Tiếp đó, vào thế kỷ 12, vua Louis trở thành hoàng đế nước Pháp đầu tiên khắc họa hoa Diên Vĩ trên chiếc khiên của mình. Nữ anh hùng nước Pháp, Joan of Arc (Jeanne dArc) đã mang theo lá cờ trắng có biểu tượng Chúa hộ mệnh của hoàng gia (hoa Diên Vĩ) khi bà đánh bại quân Anh tại Orléans (1429). Ở Nhật, hoa Diên Vĩ tượng trưng cho chí khí anh hùng và dòng dõi quý phái. Hoa Diên Vĩ là một phần quan trọng trong lễ hội mùa xuân dành cho các bé trai.*



**2.4.2.**  **Tập dữ liệu thời tiết**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TT*** | ***Outlook Temp Humidity Windy Play*** | | | | |
| 1 | Sunny | Hot | High | False | No |
| 2 | Sunny | Hot | High | True | No |
| 3 | Overcast | Hot | High | False | Yes |
| 4 | Rainy | Mild | High | False | Yes |
| 5 | Rainy | Cool | Normal | False | Yes |
| 6 | Rainy | Cool | Normal | True | No |
| 7 | Overcast | Cool | Normal | True | Yes |
| 8 | Sunny | Mild | High | False | No |
| 9 | Sunny | Cool | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | False | Yes |
| 11 | Sunny | Mild | Normal | True | Yes |
| 12 | Overcast | Mild | High | True | Yes |
| 13 | Overcast | Hot | Normal | False | Yes |
| 14 | Rainy | Mild | High | True | No |

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dữ liệu chơi golf dạng số (Numeric)** | | | | |
| **Các biến độc lập** | | | |  |
| **Quang cảnh** | **Nhiệt độ** | **Độ ẩm** | **Gió** | **Chơi** |
| Nắng | 85 | 85 | không | không |
| Nắng | 80 | 90 | có | không |
| Âm u | 83 | 78 | không | có |
| Mưa | 70 | 96 | không | có |
| Mưa | 68 | 80 | không | có |
| Mưa | 65 | 70 | có | không |
| Âm u | 64 | 65 | có | có |
| Nắng | 72 | 95 | không | không |
| Nắng | 69 | 70 | không | có |
| Mưa | 75 | 80 | không | có |
| Nắng | 75 | 70 | có | có |
| Âm u | 72 | 90 | có | có |
| Âm u | 81 | 75 | không | có |
| mưa | 71 | 80 | có | không |

# Bảng 2.4 – Bảng cơ sở dữ liệu thời tiết (tiếng Việt) và bài toán dạng định danh

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ngày*** | ***Trời*** | ***Nhiệt độ*** | ***Độ ẩm*** | ***Gió*** | ***Chơi tennis*** |
| D1 | Nắng | Cao | Cao | Yếu | Không |
| D2 | Nắng | Cao | Cao | Mạnh | Không |
| D3 | U ám | Cao | Cao | Yếu | Có |
| D4 | Mưa | Trung bình | Cao | Yếu | Có |
| D5 | Mưa | Thấp | Bình thường | Yếu | Có |
| D6 | Mưa | Thấp | Bình thường | Mạnh | Không |
| D7 | U ám | Thấp | Bình thường | Mạnh | Có |
| D8 | Nắng | Trung bình | Cao | Yếu | Không |
| D9 | Nắng | Thấp | Bình thường | Yếu | Có |
| D10 | Mưa | Trung bình | Bình thường | Yếu | Có |
| D11 | Nắng | Trung bình | Bình thường | Mạnh | Có |
| D12 | U ám | Trung bình | Cao | Mạnh | Có |
| D13 | U ám | Cao | Bình thường | Yếu | Có |
| D14 | Mưa | Trung bình | Cao | Mạnh | Không |

Giả sử ta xác định nhãn phân loại *(Prediction)* cho ví dụ sau

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ngày*** | ***Trời*** | ***Nhiệt độ*** | ***Độ ẩm*** | ***Gió*** | ***Chơi tennis*** |
| D1 | Mưa | Trung bình | Bình thường | Mạnh | ? |

**APPENDIX III** (PHỤ LỤC III)

# MỘT SỐ BÀI TẬP MẪU

**THUẬT TOÁN QUYN LAN**

Bước 1: Chia phân hoạch theo đặc trưng đầu vào

2

6

7

11

12

14

True

1

3

4

5

8

9

10

13

False

5

6

7

9

10

11

13

Normal

1

2

3

4

8

12

14

High

1

2

8

9

11

Cool

1

2

3

13

Hot

4

8

10

11

12

14

Mild

A1­­­: Windy

A1­­­: Humidity

A2­­­: Temp

4

5

6

10

14

Rainy

1

2

8

9

11

Sunny

3

7

12

13

Overcast

A1­­­: Outlook

Ai­ =

Bước 2: Tiêu chí gốc theo Quyn Lan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ai** | **j** | **Vectơ VAi (j)** | **VAi (j) = 1** | **Tỷ lệ = 1** |
| A1 | j = 1  (Sunny)  j = 2  (Overcast)  j = 3  (Rainy) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; }  VAi (3) = {Tr1(3); Tr2(3)} = { ; } | # 1  = 1  # 1 | 1/3 |
| A2 | j = 1  (Hot)  j = 2  (Mild)  j = 3  (Cool) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; }  VAi (3) = {Tr1(3); Tr2(3)} = { ; } | # 1  # 1  # 1 | 0/3 |
| A3 | j = 1  (High)  j = 2  (Normal) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; } | # 1  # 1 | 0/2 |
| A4 | j = 1  (True)  j = 2  (False) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; } | # 1  # 1 | 0/2 |

Bước 3: Dựa vào bảng ta thấy A1 có tỷ lệ đồng nhất cao nhất

Bước 4: Vậy dựa vào tiêu chí A1 ta có luật sau:

*Luật 1: IF “Outlook” là “Overcast” THEN “Play” là “Yes”*

Bước 5: Tổ hợp chập hai thuộc tính

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ai** | **j** | **Vectơ VAi (j)** | **VAi (j) = 1** | **Tỷ lệ = 1** |
| A1là Sunny ^ A2 | j = 1  (Hot)  j = 2  (Mild)  j = 3  (Cool) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; }  VAi (3) = {Tr1(3); Tr2(3)} = { ; } | = 1  # 1  = 1 | 2/3 |
| A1là Sunny ^ A3 | j = 1  (High)  j = 2  (Normal | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; } | = 1  = 1 | 2/2 |
| A1là Sunny ^ A4 | j = 1  (True)  j = 2  (False) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; } | # 1  # 1 | 0/2 |

Dựa vào bảng ta thấy A1 là Snuny ^ A3  có tỷ lệ đồng nhất cao nhất

Vậy dựa vào tiêu chí A1 là Snuny ^ A3  ta có luật sau:

*Luật 2: IF “Outlook” là “Sunny” and “Humidity” là “High” THEN “Play” là “No”*

*Luật 3: IF “Outlook” là “Sunny” and “Humidity” là “Normal” THEN “Play” là “Yes”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ai** | **j** | **Vectơ VAi (j)** | **VAi (j) = 1** | **Tỷ lệ = 1** |
| A1là Rainy ^ A2 | j = 1  (Hot)  j = 2  (Mild)  j = 3  (Cool) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; }  VAi (3) = {Tr1(3); Tr2(3)} = { ; } | = 1  # 1  # 1 | 1/3 |
| A1là Rainy ^ A3 | j = 1  (High)  j = 2  (Normal) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; } | # 1  # 1 | 0/2 |
| A1là Rainy ^ A4 | j = 1  (True)  j = 2  (False) | VAi (1) = {Tr1(1); Tr2(1)} = { ; }  VAi (2) = {Tr1(2); Tr2(2)} = { ; } | = 1  = 1 | 2/2 |

Dựa vào bảng ta thấy A1 là Rainy ^ A4  có tỷ lệ đồng nhất cao nhất

Vậy dựa vào tiêu chí A1 là Rainy ^ A4  ta có luật sau:

*Luật 4: IF “Outlook” là “Rainy” and “Windy” là “True” THEN “Play” là “No”*

*Luật 5: IF “Outlook” là “Rainy” and “Windy” là “False” THEN “Play” là “Yes”*

**THUẬT TOÁN ĐỘ LỘN XỘN**

Bước 1: Chia phân hoạch theo đặc trưng đầu vào

2

6

7

11

12

14

True

1

3

4

5

8

9

10

13

False

5

6

7

9

10

11

13

Normal

1

2

3

4

8

12

14

High

1

2

8

9

11

Cool

1

2

3

13

Hot

4

8

10

11

12

14

Mild

A1­­­: Windy

A1­­­: Humidity

A2­­­: Temp

4

5

6

10

14

Rainy

1

2

8

9

11

Sunny

3

7

12

13

Overcast

A1­­­: Outlook

Ai­ =

Bước 2: Tính độ lộn xộn

E A1 (j) = [- - ] + [- - ] + [- - ] = 0,69

E A2 (j) = [- - ] + [- - ] + [- - ] = 0,91

E A3 (j) = [- - ] + [- - ] = 0,73

E A4 (j) = [- - ] + [- - ] = 0,89

Bước 3: Chọn tiêu chí gốc có Entropy min

Bước 4: Dựa vào số hạng Entropy trong tiêu chí A1 ta có luật sau:

*Luật 1: IF “Outlook” là “Overcast” THEN “Play” là “Yes”*

Bước 5: Tổ hợp chập 2 thuộc tính

E ( A1 là Sunny) ^ A2 = [- - ] + [- - ] + [- - ]

= 0,4

E ( A1 là Sunny) ^ A3 = [- - ] + [- - ] = 0

E ( A1 là Sunny) ^ A4 = [- - ] + [- - ] = 0,95

Chọn tiêu chí gốc có Entropy min

Dựa vào số hạng Entropy trong tiêu chí A1 ta có luật sau:

*Luật 2: IF “Outlook” là “Sunny” and “Humidity” là “High” THEN “Play” là “No”*

*Luật 3: IF “Outlook” là “Sunny” and “Humidity” là “Normal” THEN “Play” là “Yes”*

E ( A1 là Rainy) ^ A2 = [- - ] + [- - ] + [- - ]

= 0,95

E ( A1 là Rainy) ^ A3 = [- - ] + [- - ] = 0,95

E ( A1 là Rainy) ^ A4 = [- - ] + [- - ] = 0

Chọn tiêu chí gốc có Entropy min

Dựa vào số hạng Entropy trong tiêu chí A1 ta có luật sau:

*Luật 4: IF “Outlook” là “Rainy” and “Windy” là “True” THEN “Play” là “No”*

*Luật 5: IF “Outlook” là “Rainy” and “Windy” là “False” THEN “Play” là “Yes”*

**THUẬT TOÁN ID3**

Bước 1: Chia phân hoạch theo đặc trưng đầu vào

2

6

7

11

12

14

True

1

3

4

5

8

9

10

13

False

5

6

7

9

10

11

13

Normal

1

2

3

4

8

12

14

High

1

2

8

9

11

Cool

1

2

3

13

Hot

4

8

10

11

12

14

Mild

A1­­­: Windy

A1­­­: Humidity

A2­­­: Temp

4

5

6

10

14

Rainy

1

2

8

9

11

Sunny

3

7

12

13

Overcast

A1­­­: Outlook

Ai­ =

Bước 2: Tính

E(A) = E(Y,N) = E(9,5) = - P+ - P- -

E(A) = E(Y,N) = E(9,5) = - - = 0,94

Mặt khác ta có:

|  |  |  |
| --- | --- | --- |
| E(A1) = 0,69 | ⇨ | = 0,25 |
| E(A2) = 0,91 |  | = 0,03 |
| E(A3) = 0,73 |  | = 0,21 |
| E(A4) = 0,89 |  | = 0,05 |

Bước 3:

E(A1 là Sunny) = E(Y,N) = E(2,3) = - P+ - P- -

E(A1 là Sunny) = E(Y,N) = E(2,3) = - - = 0,97

Mặt khác ta có:

|  |  |  |
| --- | --- | --- |
| E(A1 là Sunny) ^ A2  = 0,4 | ⇨ | = 0,57 |
| E(A1 là Sunny) ^ A3  = 0 |  | = 0,97 |
| E(A1 là Sunny) ^ A4  = 0,95 |  | = 0,02 |

E(A1 là Rainy) = E(Y,N) = E(3,2) = - P+ - P- -

E(A1 là Rainy) = E(Y,N) = E(3,2) = - - = 0,97

Mặt khác ta có:

|  |  |  |
| --- | --- | --- |
| E(A1 là Rainy) ^ A2  = 0,95 | ⇨ | = 0,02 |
| E(A1 là Rainy) ^ A3  = 0,95 |  | = 0,02 |
| E(A1 là Rainy) ^ A4  = 0 |  | = 0,97 |

**THUẬT TOÁN TLA**

Bước 1: Chia phân hoạch theo đặc trưng đầu ra

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Đổi số mới | Outlook | Temp | Humidity | Windy | Play |
| 1 | Overcast | Hot | High | False | Yes |
| 2 | Rainy | Mild | High | False | Yes |
| 3 | Rainy | Cool | Normal | False | Yes |
| 4 | Overcast | Cool | Normal | True | Yes |
| 5 | Sunny | Cool | Normal | False | Yes |
| 6 | Rainy | Mild | Normal | False | Yes |
| 7 | Sunny | Mild | Normal | True | Yes |
| 8 | Overcast | Mild | High | True | Yes |
| 9 | Overcast | Hot | Normal | False | Yes |
| 1 | Sunny | Hot | High | False | No |
| 2 | Sunny | Hot | High | True | No |
| 3 | Rainy | Cool | Normal | True | No |
| 4 | Sunny | Mild | High | False | No |
| 5 | Rainy | Mild | High | True | No |

Bước 2:

|  |  |  |
| --- | --- | --- |
| Ai | j | Điểm |
| A1 | j = 1 (Overcast) | 4 |
| j = 2 (Rainy) | 0 |
| j = 3 (Sunny) | 0 |
| A2 | j = 1 (Hot) | 0 |
| j = 2 (Mild) | 0 |
| j = 3 (Cool) | 0 |
| A3 | j = 1 (High) | 0 |
| j = 2 (Normal) | 0 |
| A4 | j = 1 (False) | 0 |
| j = 2 (True) | 0 |

Dựa vào bảng ta chọn được Outlook – Overcast viết luật:

*Luật 1: IF “Outlook” là “Overcast” THEN “Play” là “Yes”*

Bước 3: Tổ hợp chấp 2 tiêu chuẩn

|  |  |  |
| --- | --- | --- |
| Ai | J | Điểm |
| A1là Sunny ^ A2 | j = 1 (Hot) | 0 |
| j = 2 (Mild) | 1 |
| j = 3 (Cool) | 1 |
| A1là Sunny ^ A3 | j = 1 (High) | 0 |
| j = 2 (Normal) | 2 |
| A1là Sunny ^ A4 | j = 1 (False) | 1 |
| j = 2 (True) | 1 |

Dựa vào bảng ta chọn được Outlook – Sunny ^ A3 viết luật:

*Luật 2: IF “Outlook” là “Sunny” and “Humidity” là “High” THEN “Play” là “No”*

*Luật 3: IF “Outlook” là “Sunny” and “Humidity” là “Normal” THEN “Play” là “Yes”*

|  |  |  |
| --- | --- | --- |
| Ai | J | Điểm |
| A1là Rainy ^ A2 | j = 1 (Hot) | 0 |
| j = 2 (Mild) | 2 |
| j = 3 (Cool) | 1 |
| A1là Rainy ^ A3 | j = 1 (High) | 1 |
| j = 2 (Normal) | 2 |
| A1là Rainy ^ A4 | j = 1 (False) | 3 |
| j = 2 (True) | 0 |

Dựa vào bảng ta chọn được Outlook – Sunny ^ A3 viết luật:

*Luật 4: IF “Outlook” là “Rainy” and “Windy” là “True” THEN “Play” là “No”*

*Luật 5: IF “Outlook” là “Rainy” and “Windy” là “False” THEN “Play” là “Yes”*

**THUẬT TOÁN BAYES**

Để minh họa thuật giải Bayes, chúng ta sử dụng bảng:

Tập hợp dữ liệu học về dự báo thời tiết

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ngày** | **Trời** | **Nhiệt độ** | **Độ ẩm** | **Gió** | **Chơi Tennis** |
| D1 | Nắng | Nóng | Cao | Yếu | Không |
| D2 | Nắng | Nóng | Cao | Mạnh | Không |
| D3 | Nhiều mây | Nóng | Cao | Yếu | Có |
| D4 | Mưa | Trung bình | Cao | Yếu | Có |
| D5 | Mưa | Ấm áp | Bình thường | Yếu | Có |
| D6 | Mưa | Lạnh | Bình thường | Mạnh | Không |
| D7 | Nhiều mây | Lạnh | Bình thường | Mạnh | Có |
| D8 | Nắng | Ấm áp | Cao | Yếu | Không |
| D9 | Nắng | Lạnh | Bình thường | Yếu | Có |
| D10 | Mưa | Ấm áp | Bình thường | Yếu | Có |
| D11 | Nắng | Ấm áp | Bình thường | Mạnh | Có |
| D12 | Nhiều mây | Ấm áp | Cao | Mạnh | Có |
| D13 | Nhiều mây | Nóng | Bình thường | Yếu | Có |
| D14 | Mưa | Ấm áp | Cao | Mạnh | Không |

Dữ liệu của mẫu tin cần dự báo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trời** | **Nhiệt độ** | **Độ ẩm** | **Gió** | **Chơi Tennis** |
| Nắng | Lạnh | Cao | Mạnh | ? |

Dữ liệu của mẫu tin cần dự báo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trời** | **Nhiệt độ** | **Độ ẩm** | **Gió** | **Chơi Tennis** |
| Nắng | Nóng | Cao | Yếu | ? |

Bước 1: Phân hoạch theo đặc trưng đầu vào

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Trời** | | | **Nhiệt độ** | | | **Độ ẩm** | | | **Gió** | | | **Chơi Tennis** | |
| Có | | Không | Có | | Không | Có | | Không | Có | | Không | Có | Không |
| Nắng  Nhiều mây  Mưa | 2  4  3 | 3  0  2 | Nóng  Ấm áp  Lạnh | 2  4  3 | 2  2  1 | Cao  Bình thường | 3  6 | 4  1 | Yếu  Mạnh | 6  3 | 2  3 | 9 | 5 |

Bước 1: Chia phân hoạch theo đặc trưng đầu vào

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Outlook | | | Temp | | | Humidity | | | Windy | | | Play | |
| Yes | | No | Yes | | No | Yes | | No | Yes | | No | Yes | No |
| Sunny  Overcast  Rainy | 2  4  3 | 3  0  2 | Hot  Wild  Cool | 2  4  3 | 2  2  1 | High  Normal | 3  6 | 4  1 | False  True | 6  3 | 2  3 | 9 | 5 |

Bước 2:

C:\Users\Administrator\Desktop\2.png P(Outlook = Sunny/Yes) . P(Temp = Hot/Yes) . P(Humidity = High/Yes) . P(Windy = False/Yes)  . P(Yes)

P(X)

=

C:\Users\Administrator\Desktop\1.png P(Outlook = Sunny/No) . P(Temp = Hot/No) . P(Humidity = High/No) . P(Windy = False/No)  . P(No)

P(X)

=