BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

TÌM HIỂU VỀ HỌC MÁY VÀ PHƯƠNG PHÁP HỌC KHÁI NIỆM. XÂY DỰNG MODULE MÔ PHỎNG THUẬT TOÁN FIND-S

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công Nghệ Thông Tin

Sinh viên thực hiện: Vũ Ngọc Nam

Giáo viên hướng dẫn: Ths. Vũ Mạnh Khánh

Mã số sinh viên: 111351

Chương 1:Tổng Quan Về Học máy

1.1 Giới thiệu về học máy

Học máy, nhiều tài liệu còn gọi là máy học là một lĩnh vực của trí tuệ nhân tạo cho phép máy tính có thể "học" được. Cụ thể hơn, học máy là phương pháp để tạo ra những chương trình máy tính dựa trên các tập dữ liệu.Trong chương này sẽ đưa ra câu trả lời chung nhất cho các câu hỏi:

- Tại sao phải học máy ?
- Thế nào là bài toán học máy xác định đúng ?
- Cho ví dụ minh hoạ ?
- Các vấn đề chính trong học máy ?

Từ khi phát minh ra máy tính thì người ta đã muốn làm những chương trình làm cho máy học được như người (học máy) nhưng đến nay mong muốn đó vẫn chưa thực hiện được. Tuy vậy người ta cũng đã có nhiều thành tựu trong lĩnh vực học máy như:

- Nhận dạng tiếng nói
- Xử lý ngôn ngữ tự nhiên
- Dự đoán, chuẩn trị trong y học
- Nhận dạng ADN
- Phân loại cấu trúc thiên văn mới
- Chơi cờ và một số trò chơi khác
- Lái xe tự động...
- Các ứng dụng để khám phá tri thức (Data mining) trong các CSDL lớn nhằm trợ giúp quyết định.

Học máy kế thừa thành tựu của nhiều lĩnh vực khoa học. Sau đây là một số lĩnh vực và ý tưởng chính ảnh hưởng tới học máy:

- Trí tuệ nhân tạo: Học biểu diễn tượng trưng các khái niệm; học máy như là bài toán tìm kiếm; học như là cách tiếp cận cải tiến giải bài toán; sử dụng tri thức sãn có và dữ liệu đào tạo để hướng dẫn học.
- Các phương pháp Bayes: định lý Bayes làm cơ sở để tính xác suất của các giả thuyết, Cách phân lớp Bayes, và các thuật toán ước lượng các giá trị không quan sát được.
- Lý thuyết độ phức tạp tính toán: tính độ phức tạp của các nhiệm vụ học đo qua các ví dụ đào tạo, số lỗi và các tính toán...
- Lý thuyết điều khiển: 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.
- Lý thuyết thông tin: các độ đo của nội dung thông tin và entropy;Mã tối ưu và quan hệ của chúng tới dãy đào tạo tối ưuđể mã hoá một giả thuyết.
- Triết học: những nguyên lý như Occam's razor (cho rằng giả thuyết đơn giản nhất là tôt nhất ; các phân tích luận chứng để tổng quát hoá các dữ liệu quan sát được.
- Tâm lý học và thần kinh học: các đáp ứng thực tế của con người, các mô hình neural.
- Thống kê: đặc trưng lỗi, lý thuyết lấy mẫu, khoảng tin cậy...

1.2 Các bài toán học thiết lập đúng đắn.

Trước hết ta cần phát biểu một định nghĩa chung nhất cho một chương trình học đối với một số nhiệm vụ cần học nào đó.

Định nghĩa. Một chương trình máy tính được gọi là học từ thí nghiệm E đối với lớp nhiệm vụ học T và độ đo mức thực hiện P nếu sự thực hiện các nhiệm vụ trong Tcủa nó khi đo bởi T được cải tiến qua kinh nghiệm E.

Các ví dụ:

1)Chương trình học chơi cờ với chính nó

- T: Chơi cờ
- P: Tỷ lệ thắng đối thủ
- E: chơi với chính nó hoặc với người khác hay có thầy.

Với bài toán này ta cần biểu diễn đặc trưng ván cờ như thế nào? học cái gì? thuật toán học ra sao?..

2)Nhận dạng chữ viết tay.

- T: phân lớp (nhận dạng) chữ viết tay nhờ phân tích ảnh
- P : tỷ lệ chữ nhận dạng đúng
- E : các chữ viết đã phân loại được đưa tuần tự.

3) Robot học lái

- T: lái xe trên đường cao tốc 4-làn xe nhờ sử dụng các sensor nhìn
- P: quảng cách trung bình đi được trước khi có lỗi
- E : một loạt các ảnh và lệnh lái được ghi qua quan sát của người dạy

4) Phân nhóm đối tượng. Tập đối tượng cần chia làm k nhóm sao cho mỗi đối tượng trong một nhóm tương tự nhau còn các đối tượng khác nhóm thì khác nhau.

- T: phân nhóm
- P: sự tương tự và tương quan chung
- E: Quá trình phân nhóm.

5) Chẩn trị bệnh.

Dựa trên cở dữ liệu về các bệnh án: lưu trữ triệu chứng; phác đồ điều trị và kết quả mỗi bệnh nhân, chẩn đoán bệnh và đưa phác đồ điều trị cho bệnh nhân mới.

- T: Đoán và đưa ra phác đồ điều trị
- P: Tỷ lệ chẩn đoán đúng và điều trị khỏi bệnh
- E: Các bệnh án đã biết.

1.3 Thiết kế một hệ học.

Trong mục này ta xét bài toán học chơi cờ với đối thủ để minh hoạ.

Chọn kinh nghiệm đào tạo.

Trước hết cần chọn kiểu kinh nghiệm đào tạo: trực tiếp hay gián tiếp (có liên hệ ngược) . Việc lựa chọn này có ảnh hưởng lớn tới thành công hoặc thất bại của hệ học.

Ví dụ

Trực tiếp: kinh nghiệm chơi cờ có thể cho bởi các thế cờ và nước đi đúng cho từng thế.

Gián tiếp: các thông tin bao gồm dãy nước đi và kết cục của nhiều ván chơi.

Thứ hai là chọn cách điều khiển dãy kinh nghiệm đào tạo:

Có thể là các thế cờ và nước cờ thầy cho sẵn và hệ học hoàn toàn dựa vào đó hoặc hệ chơi tự tạo ra các ván cờ và trạng thái mới hoặc có thầy hay không có thầy (tuỳ theo từng bài toán cụ thể)

Thứ ba là làm thế nào để đánh giá độ đo đích thực P qua các độ đo trên thí dụ đào tạo.

(Các nước đi qua thí nghiệm có thể không tốt vì chưa gặp đối thủ thực sự). Khi đó ta cần một số giả thiết bổ sung.

Nếu bây giờ chọn cách chơi với chính nó, ta đã mô tả cụ thể nhiệm vụ học chơi cờ:

- T: Chơi cờ
- P: Tỷ lệ thắng đối thủ
- E: chơi với chính nó

Tiếp theo ta cần chọn

1. Kiểu chính xác định tri thức để học
2. Một cách biểu diễn tri thức đích
3. Một cơ cấu học

1.4 Các vấn đề trong học máy.

Việc học máy thực chất là tìm kiếm trong không gian giả thuyết lớn một giả thuyết phù hợp nhất với dữ liệu quan sát được và các tri thức đã có. Trong ví dụ trên , không gian giả thuyết được xác định bởi các giá trị trọng số dữ liệu là các ván chơi và tri thức làcách để nước đi hợp lệ.

Các vấn đề thường gặp trong học máy là:

- Từ các ví dụ đào tạo có tồn tại thuật toán để học hàm đích hay không? thuật toán có hội tụ tới hàm đích mong muốn không? dữ liệu có đủ cho thuật toán không?thuật toán nào tốt nhất cho bài toán và cách thể hiện đang xét.
- Bao nhiêu dữ liệu đào tạo thì đủ? Trong không gian giả thuyết và các dữ liệu đang có thi ướcc lượng các mức tin cậy như thế nào?
- Các tri thức đã biết có tác dụng gì và như thế nào cho quá trình học.
- Chiến lược thí nghiệm đào tạo như thế nào là tốt nhất, cách chọn chiến lươc này như thế nào?
- Cách nào tốt nhất để đưa bài toán học về bài toán xấp xỉ hàm? Quá trình này có thể tự động hoá được không?
- Làm thế nào hệ học có thể lựa chon cách biểu diễn để cải tiến khả năng biểu diễn và học hàm đích?

Chương 2:

2.1 Các công thức xác suất thông kê cơ bản

Không gian mẫu và không gian biến cố

Không gian mẫu

Tiến hành thực hiện một phép thử (thí nghiệm). Giả sử ta không biết trước được kết quả của phép thử nhưng ta sẽ biết tập tất cả các kết quả có thể của phép thử. Ta có một *phép thử ngẫu nhiên*.

Tập tất cả các kết quả có thể của phép thử được gọi là *không gian mẫu* của phép thử, ký hiệu Ω. Với một phép thử, ta có thể xác định không gian mẫu theo nhiều cách.

Không gian biến cố

Mỗi phần tử $\omega \in \Omega$ được gọi là một biến cố sơ cấp.

Mỗi tập con của Ω được gọi là một biến cố (ngẫu nhiên), ký hiệu A, B,…Khi phép thử được thực hiện, ta nói biến cố A xảy ra nếu kết quả xuất hiện là một phần tử của A.

- *Các phép toán với biến cố*

Cho A, B, C, A_1, ... , A_n là các biến cố trong không gian mẫu $\Im$. Khi đó,

- A được gọi là *kéo theo* B, ký hiệu A Ì B nếu sự xảy ra của A kéo theo sự xảy ra của B.
- *Tổng* của A và B, ký hiệu A $\cup$ B là biến cố xảy ra nếu A hoặc B xảy ra.
- *Tích* của A và B, ký hiệu A $\cap$ B hay AB là biến cố xảy ra nếu A và B xảy ra.
- *Hiệu* của A và B, ký hiệu A \ B là biến cố xảy ra nếu A xảy ra và B không xảy ra.
- Biến cố $\bar{A} = \Omega \setminus A$ được gọi là *biến cố đối lập* của biến cố A.
- Nếu A $\cap$ B = Θ thì A và B được gọi là *xung khắc* với nhau.
- Dãy n biến cố $A_1, A_2, \ldots, A_n$ lập thành một *hệ đầy đủ* các biến cố nếu

i_1/ $A_1 \cup A_2 \cup \ldots \cup A_n = \Omega$

i_2/ $A_i \cap A_j = \Theta$ với mọi $i \neq j$ (xung khắc từng đôi).

Cần lưu ý một số tính chất sau:

* $A \cup B = B \cup A$; $A \cap B = B \cap A$

* $A \cup (B \cup C) = (A \cup B) \cup C = A \cup B \cup C$;

$A \cap (B \cap C) = (A \cap B) \cap C = A \cap B \cap C$

* $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$; $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

* $A \cup \bar{A} = W$; $A \cap \bar{A} = \Theta$.

* $\overline{A \cup B \cup C} = \bar{A} \cap \bar{B} \cap \bar{C}$; $\overline{A \cap B \cap C} = \bar{A} \cup \bar{B} \cup \bar{C}$

Công thức xác suất cổ điển

$$P(A) = \frac{m}{n} = \frac{\text{số khả năng thuận lợi cho A}}{\text{số khả năng có thể}}$$

Xác suất điều kiện

Trong nhiều trường hợp, một vấn đề được đặt ra là: ta có thể nói gì về xác suất của biến cố A nếu có thông tin biến cố B nào đó (liên quan tới A) đã xảy ra? Trong những trường hợp đơn giản nhất, câu trả lời khá dễ dàng. Chẳng hạn, nếu A và B xung khắc thì A không thể xảy ra, vì vậy xác suất để A xảy ra bằng 0. Trường hợp khác, nếu $B \subset A$ thì A chắc chắn xảy ra nên xác suất của nó bằng 1. Vấn đề còn lại, nếu B đã xảy ra chỉ cho ta một phần thông tin về phép thử (tức cho A) thì khi đó P(A) được xác định thế nào. Khái niệm xác suất điều kiện sẽ được sử dụng cho trường hợp này.

Định nghĩa 1.1.1. Cho không gian xác suất (W, $\mathfrak{F}$, P) và $B \in \mathfrak{F}$ với P(B) > 0. Khi đó với biến cố A bất kỳ, *xác suất điều kiện* của biến cố A với điều kiện biến cố B đã xảy ra, ký hiệu $P(A|B)$ được xác định bởi

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(AB)}{P(B)}$$

Tính chất 1.1.2.

* $0 \le P(\mathbf{A}|B) \le 1$.
* $P(\Omega|B) = 1.;\ \ P(B|B) = 1.$
* $P(\mathbf{A} \cup B|C) = P(\mathbf{A}|C) + P(B|C) - P(\mathbf{AB}|C)$.
* $P(\mathbf{A}|B) = 1 - P(\overline{\mathbf{A}}|B).$

Công thức xác suất của biến cố tích

Từ định nghĩa xác suất điều kiện ta suy ra

$$P(\mathbf{AB}) = P(\mathbf{A})P(B|\mathbf{A}) = P(B)P(\mathbf{A}|B)$$

Mở rộng cho trường hợp tổng quát ta nhận được

$$P(\mathbf{A}_1\mathbf{A}_2...\mathbf{A}_n) = P(\mathbf{A}_1)P(\mathbf{A}_2|\mathbf{A}_1)P(\mathbf{A}_3|\mathbf{A}_1\mathbf{A}_2)...P(\mathbf{A}_n|\mathbf{A}_1\mathbf{A}_2...\mathbf{A}_{n-1})$$

Sự độc lập của các biến cố

Định nghĩa 1.1.6. Hai biến cố A và B được gọi là *độc lập* với nhau nếu

$$P(AB) = P(A) \,.\, P(B)$$

Từ định nghĩa trên dễ suy ra các kết quả sau

- Hai biến cố A và B là độc lập với nhau khi và chỉ khi

$$P(\mathbf{A}|B) = P(\mathbf{A}) \text{ hoặc } P(B|\mathbf{A}) = P(B)$$

- Hai biến cố A và B là độc lập với nhau khi và chỉ khi $A, \overline{B}$ độc lập hoặc $\overline{A}, B$ là độc lập hoặc $\overline{A}, \overline{B}$ là độc lập.

Định nghĩa 1.1.7. Dãy n biến cố $B_1, B_2,..., B_n$ được gọi là

* *Độc lập từng đôi* nếu

$P(B_iB_j) = P(B_i) . P(B_j)$ với mọi $i \neq j$; $i, j = \overline{1,n}$

* *Độc lập trong toàn thể* nếu với bất kì $1 \leq i_1 < i_2 < \ldots < i_r \leq n$; $r = 2, 3, \ldots, n$ thì

$$P\left(\bigcap_{k=1}^{r} B_{i_k}\right) = \prod_{k=1}^{r} P(B_{i_k})$$

Công thức xác suất toàn phần

Giả sử $A_1, A_2, ..., A_n$ là một hệ đầy đủ các biến cố và $P(A_i) > 0$ với mọi i = 1, 2, ..., n. Khi đó với mọi biến cố A bất kỳ ta luôn có

$$P(\mathbf{A}) = P(\mathbf{A} \cap \Omega) = P\left(A \cap (\bigcup_{i=1}^{n} \mathbf{A}_i)\right) = P\left(\bigcup_{i=1}^{n} (\mathbf{A} \cap \mathbf{A}_i)\right) = \sum_{i=1}^{n} P(\mathbf{A} \cap \mathbf{A}_i)$$

Từ đó,

$$P(\mathbf{A}) = \sum_{i=1}^{n} P(\mathbf{A}_i) P(\mathbf{A}|\mathbf{A}_i)$$

Công thức trên được gọi là *công thức xác suất* toàn phần.

Công thức Bayes

Trong nhiều trường hợp ta gặp các phép thử mà trong đó có thể có điều kiện này hay điều kiện khác tham gia vào một cách ngẫu nhiên. Ta tiến hành phép thử đó và dựa theo kết quả nhận được, ta giải thích xác suất để một trong các điều kiện ngẫu nhiên tham gia vào trong phép thử là bao nhiêu. Để giải bài toán này, ta cần công thức gọi là *công thức Bayes* như sau

Định nghĩa: Giả sử $A_1, A_2, ..., A_n$ là một hệ đầy đủ các biến cố và $P(A_i) > 0$ với mọi i = 1, 2, ..., n. Khi đó nếu A là biến cố bất kỳ với P(A) > 0 ta có

$$P(\mathbf{A}_i|\mathbf{A}) = \frac{P(\mathbf{A}_i) P(\mathbf{A}|\mathbf{A}_i)}{P(\mathbf{A})} = \frac{P(\mathbf{A}_i) P(\mathbf{A}|\mathbf{A}_i)}{\sum_{j=1}^{n} P(\mathbf{A}_j) P(\mathbf{A}|\mathbf{A}_j)}$$

2.2 Chọn hàm đích.

Tiếp theo ta cần xác định đúng tri thức cần học và cách chương trình sử dụng chúng. Trong bài toán chơi cờ, tri thức học là nước đi tốt nhất trong số các nước đi hợp lệ cho mỗi thế cờ tức là tìm hàm (ánh xạ) ChooseMove: $B \to M$ trong đó B là tập các trạng thái bàn cờ (thế cờ) và M là các nước đi hợp lệ với trạng thái tương ứng . Việc tìm tri thức này thường là bài toán NP-khó nên khó xác định và ta tìm cách chọn một hàm đích thích hợp có thể cải tiến nhờ quá trình học. Hàm đích có thể cho dưới dạng khác nhau: cho bởi bảng xác định các giá trị nước đi cụ thể cho mỗi thế cờ hoặc là hàm giá trị thực V: $B \to R$ ... . Giả sử ta tìm được hàm đích dưới dạng V: $B \to R$ (R là tập số thực) sao cho nước đi tốt hơn thì hàm nhận giá trị lớn hơn. Nếu hệ học có hàm đích V này thì nước đi tốt nhất sẽ cực đại hàm V với thế cờ tương ứng.

Vấn đề đặt ra là chọn hàm đích này như thế nào? Một cách đơn giản , ta xác định V(b) với $b \in B$ như sau:

1. Nếu b là trạng thái kết thúc thắng thì V(b)=100
2. Nếu b là trạng thái kết thúc thua thì V(b)= -100
3. Nếu b là trạng thái kết thúc hoà thì V(b)= 0
4. Nếu b không là trạng thái kết thúc thì V(b)= V)b') trong đó b' là trạng thái kết thúc tốt nhất đạt được từ b nhờ chọn nước đi tốt nhất đến cuối ván.

Hàm V này không có giá trị thực hành vì trường hợp 4 ở trên không xác định được và có nhiều cách định nghĩa V nhưng đều khó xác định trong thực hành. Chúng ta có thể xấp xỉ nó bởi hàm $\overline{V}$ dạng thích hợp để làm làm hàm đích hay là chọn cách biểu diễn hàm đích.

2.3 Chọn biểu diễn cho hàm đích.

Có nhiều cách chọn hàm $\overline{V}$, có thể cho bằng bảng hoặc cho bởi các quy tắc xác định giá trị theo mỗi đặc trưng của thế cờ hay là đa thức của các giá trị đặc trưng.

Nếu dùng ký hiệu:

- x_1 là số quân đen trên bàn cờ
- x_2 là số quân trắng trên bàn cờ
- x_3 là số hậu đen trên bàn cờ
- x_4 là số hậu trắng trên bàn cờ
- x_5 là số quân đen bị đe doạ trên bàn cờ
- x_6 là số quân trắng bị đe doạ trên bàn cờ

thì $\overline{V}$ có thể xác định một cách đơn giản là một hàm tuyến tính của các đối số này.

$\overline{V} = w_0 + w_1x_1 + ... + w_6x_6$

Trong đó w_0 ...w_6 là các hệ số cần xác định.

Công việc học bây giờ là:

- nhiệm vụ T: Chơi cờ
- Độ đo thực hiện P: Tỷ lệ thắng đối thủ
- Kinh nghiệm đào tạo E: chơi với chính nó
- Hàm đích V: $B \rightarrow R$
- Biểu diễn hàm đích: $\overline{V} = w_0 + w_1x_1 + ... + w_6x_6$

Nếu $\overline{V}$ được xác định thì nước đi tốt nhất là nước đi hợp lệ làm cực đại $\overline{V}$ cho mỗi thế tương ứng và chương trình học là tìm cách xác định các hệ số w_0 ...w_6 cho $\overline{V}$ qua kinh nghiệm E.

2.4 Chọn thuật toán xấp xỉ hàm .

Để xác định $\overline{V}$ ta cần tập mẫu đào tạo < b, $V_{train}(b)$> chẳng hạn khi x_2= 0 thì quân đen thắng nên $V_{train}(b)$=100 và ta có mẫu đào tạo :

< (x_1=3, x_2=0,x_2=0,x_3=0,x_4=0,x_5=0,x_6=0), 100>

Thường thì ban đầu các trọng số w được khởi tạo một cách thích hợp và sau đó được hiệu chỉnh dần qua các mẫu đào tạo.

Ước lượng giá trị đào tạo

Nếu b là thế kết thúc thì $V_{train}(b)$ được xác định còn khi b là thế cờ trung gian thì $V_{train}(b)$ xác định theo quy tắc:

$$V_{train}(b) \leftarrow \overline{V}(\text{successor}(b))$$

Điều ta cần là qua quá trình học giá trị xấp xỉ của $V_{train}(b)$ hội tụ tới giá trị đúng.

Hiệu chỉnh trọng số.

Một cách tiếp cận thường hùng để hiệu chỉnh trọng số w (giả thuyết) là phương pháp bình phương tối thiểu LMS nhờ cực tiểu sai số

$$E = \sum_{<b,V_{train}(b)\in Q} (V_{train}(b) - \overline{V}(b))^2$$

Trong đó Q là tập ví dụ đào tạo .

Quy tắc cập nhật trọng số LMS như sau.

Với mỗi ví dụ đào tạo < b, $V_{train}(b)$> ,

- Dùng trọng số hiện tại tính $\overline{V}$ (b)
- Với mỗi w_i cập nhật:

$$w_i \leftarrow w_i + \eta(V_{train} - \overline{V}(b))x_i)$$

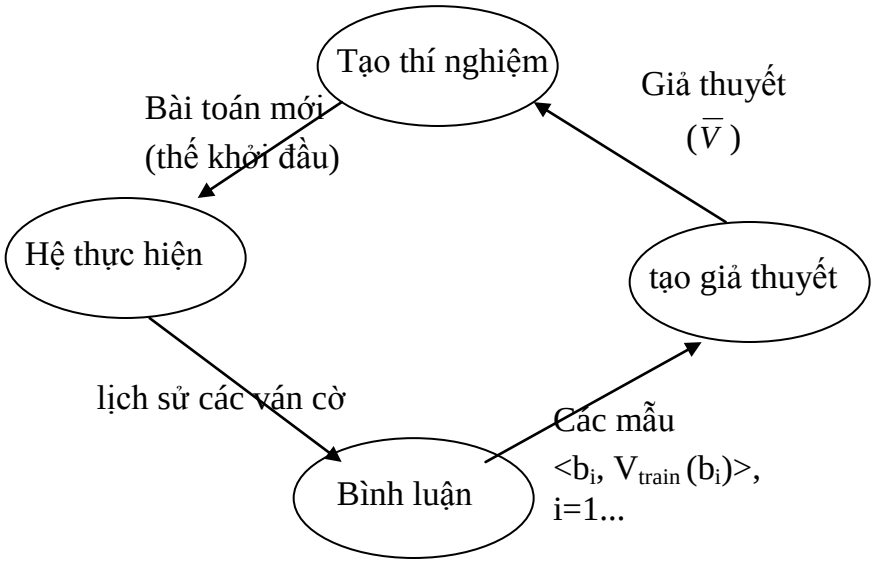
trong đó η là hằng số thuộc khoảng (0,1) gọi là tốc độ học.

2.5 Thiết kế cuối cùng.

Thiết kế cuối cùng cho hệ học chơi cờ gồm 4 môđun chương trình chính trong hình 1.1

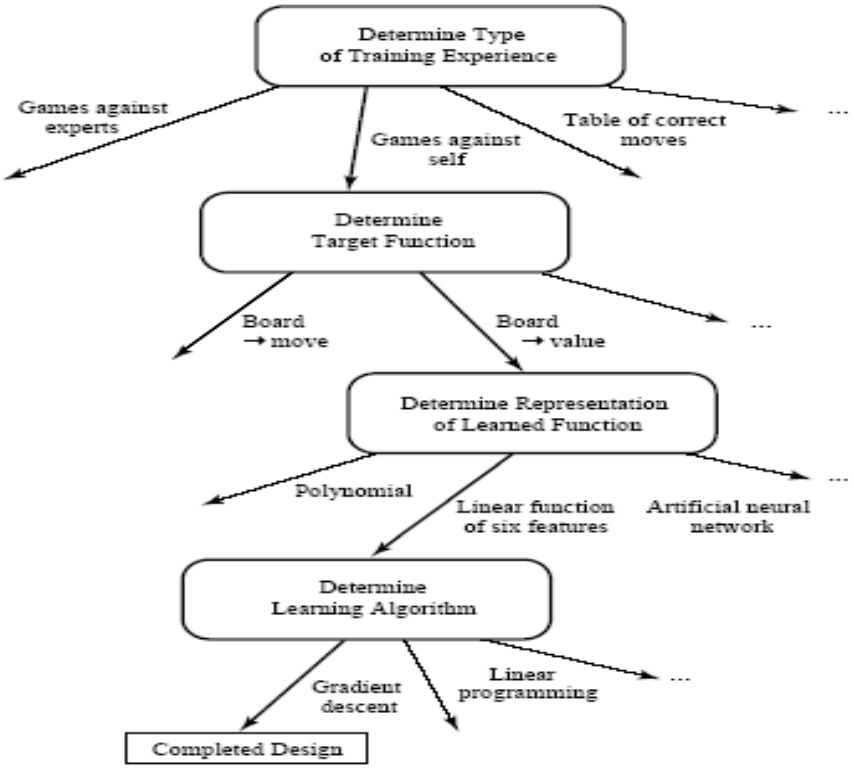
- Hệ thực hiện. Lấy input là thế cờ được tào ra và dùng hàm đích để xác định nước đi tiếp qua đánh giá $\overline{V}$.

- Môđun bình luận. Nhờ theo dõi toàn bộ các ván ván chơi để ước lượng V_{train} cho các ví dụ .
- Bộ tạo giả thuyết. Dùng các ví dụ đào tạo, đưa ra ước lượng tham số của hàm đích $\overline{V}$.
- Bộ tạo thí nghiệm. Tạo ra thế cò mới bắt đầu cuộc chới theo yêu cầu nào đó.



Hình 1.1. Thiết kế cuối cùng cho chương trình học chơi cờ

Quá trình thiết kế chơi cờ mô tả trong hình 1.2.



Hình 1.2 tóm tắt quá trình thiết kế học chơi cờ

Chương 3 Học khái niệm và sắp thứ tự từ tổng quát đến chi tiết

Bài toán quy nạp một hàm chung (tổng quát) từ các ví dụ cụ thể là trung tâm của việc học. Chương này xét bài toán xác định một phạm trù chung từ các ví dụ đúng và sai. Bài toán này có thể thiết lập như là bài toán tìm kiếm trong không gian giả thiết tiềm năng để có giả thuyết phù hợp nhất với các mẫu được xét nhờ cách sắp thứ tự các giả thuyết từ tổng quát đến chi tiết.

Các điểm chính:

- Học từ các ví dụ (mẫu)
- Sắp thứ tự các giả thuyết từ tổng quát đến chi tiết

- Không gian tường thuật và thuật toán loại trừ ứng cử
- Lấy ví dụ
- Sự cần có khuynh hướng quy nạp

3.1. Giới thiệu.

Nhiều trường hợp ta cần học một khát niệm tổng quát từ các ví dụ cụ thể như : cây, chim, mèo, ôtô, xe máy...từ các trường hợp quan sát cụ thể và ta goi là học khái niệm. Trong đó ta biết một hàm logic trên tâp đối tượng hoặc biến cố và cần xác định nó trên tập rộng hơn. Trong chương này ta xét bài toán suy diễn tự động định nghĩa tổng quát một khái niệm nào đó từ các ví dụ cụ thể đúng hoặc sai.

Học khái niệm: Suy luận một hàm logic từ tập ví dụ các đầu vào và đầu ra của nó.

3.2.Ví dụ về học khái niệm.

Giả sử ta học khái niệm : *ngày bạn A thích chơi môn thể thao dưới nước* từ tập dữ liêu các ngày ban A chơi hoặc không được mô tả từ các thuộc tính : Bầu trời, nhiệt độ, độ ẩm, gió, nước, dự báo (cho trong bảng 2.1). Ta cần dự báo anh ta có chơi hay không trong một ngày nào đó với giá trị thuộc tính tương ứng đã biết của ngày này.

Bảng 2.1. Các ví dụ về những ngày chơi hoặc không của A.

Ví dụ	Bầu trời	nhiệt độ	độ ẩm	gió	nước	dự báo	thích chơi
1	nắng	ấm	trung bình	mạnh	ấm	không đổi	có
2	nắng	ấm	cao	mạnh	ấm	không đổi	có
3	mưa	lạnh	cao	mạnh	ấm	đổi	không
4	nắng	ấm	cao	mạnh	lạnh	đổi	có

Ta bắt đầu từ xét các giả thuyết chỉ gồm các liên kết các ràng buộc trên các thuộc tính. Các thuộc tính có thể là:

- Ký hiệu bởi ? nếu bất cứ giá trị nào cũng được chấp nhận
- Ký hiệu bởi ϕ nếu không giá trị nào cũng được chấp nhận
- Một giá trị cụ thể của thuộc tính.

Chẳng hạn giả thuyết A chỉ chơi khi trời lạnh và độ ẩm cao được biểu diễn bởi:

<?, lạnh, cao,?,?,?>,

Giả thuyết tổng quát nhất : mọi ngày đều chơi là :<?, ?, ?,?,?,?>,

Còn giả thuyết chi tiết nhất: không ngày nào chơi là: $<\phi,\phi,\phi,\phi,\phi,\phi>$.

Tổng quát, ta biết tập mẫu và giá trị hàm đích của nó, tập các giả thuyết có thể được xét ta tìm hàm đích cho mẫu chưa biết. Bài toán học khái niệm ngày chơi thể thao cho trong bảng 2.2 dưới đây.

Bảng 2.2. Nhiệm vụ học khái niệm ngày ưa thể thao.

+Đã cho:

-Tập mẫu X: Các ngày với giá trị thuộc tính có thể nhận:

- Bầu trời (nắng, nhiều mây, mưa)
- Nhiệt độ (ấm , lạnh)
- Độ ẩm (trung bình, cao)
- Gió (mạnh, yếu)
- Nước (ấm, lạnh)
- Dự báo (không đổi, đổi)

-Các giả thuyết H:Mỗi giả thuyết là một liên kết các giá trị thuộc tính, chúng có thể là ?,ϕ, hoặc giá trị cụ thể.

- Khái niệm đích c: ưa thể thao: $X \rightarrow \{0,1\}$

- Dữ liệu huấn luyện D: các ví dụ có hoặc không trong bảng 2.1.

+ Xác định một giả thuyết h sao cho h(x)= c(x) với mọi x thuộc X.

Các khái niệm đưa ra:

- Tập D đã biết giá trị đích gọi là tập mẫu. Khái niệm hay hàm cần học gọi là khái niệm đích và ký hiệu là c tức là $c: X \rightarrow \{0,1\}$. Trong ví dụ này c(x)=1 nếu chơi và c(x)= 0 nếu không chơi. Khi học, hệ học được giới thiệu một tập mẫu với giá trị đích đã biết.
- Mẫu với c(x)= 1 gọi là mẫu (ví dụ) dương; mẫu c(x)= 0 gọi là mẫu (ví dụ) âm.
- Ký hiệu H là tập các giả thuyết có thể được xét để nhận dạng hàm đích (tập H được xác định bởi người thiết kế). Ta cần tìm $h \in H$ sao cho h(x)=c(x)với mọi $x \in X$.

Giả thuyết học quy nạp:

Học quy nạp phải dựa trên giả thiết rằng: bất cứ giả thuyết nào xấp xỉ tốt hàm đích trên tập mẫu đủ lớn thì cũng xấp xỉ tốt hàm đích trên các mẫu chưa biết.

Các giả thuyết thoả mãn giả thiết trên gọi là giả thuyết học quy nạp.

3.3. Học khái niệm như là bài toán tìm kiếm.

Học khái niệm có thể xem là bài toán tìm trên H giả thuyết phù hợp tốt nhất với tập mẫu. Với bài toán trên có thể có 3.2.2.2.2.2=96 mẫu thí dụ khác biệt và rất nhiều các giả thuyết khác nhau trong H.Ta cần tìm thuật toán tìm kiếm có hiệu quả.

Thứ tự tổng quát đến chi tiết của các giả thuyết.

Nhiều thuật toán học khái niệm bàng cách tìm kiếm trong không gian giả thuyết bằng cách sử dụng cấu trúc thứ tự: *tổng quát đến chi tiết của không gian giả thuyết*. Nhờ cấu

trúc này mà ta có thể tìm kiếm vét cạn không gian giả thuyết (thậm chí vô hạn) mà không cần đánh số giả thuyết.

Để minh hoạ thứ tự này ta xét hai giả thuyết:

h_1= <nắng,?,?, mạnh, ? , ? >

h_2= <nắng,?,?, ?, ? , ? >

và xét tập mẫu được h_1; h_2 phân lớp dương. Bởi vì h_2 ít ràng buộc hơn nên nó có nhiều mẫu dương hơn h_1. Ta nói h_2 tổng quát hơn h_1.

Trước khi định nghĩa chính xác quan hệ thứ tự tổng quát ta cần khái niệm mẫu x thoả mãn giả thuyết h.

Định nghĩa 1: $\forall$ phần tử $x \in X$; $h \in H$ ta nói x ***thoả mãn*** h nếu h(x)=1.

Bây giờ ta đĩnh nghĩa thứ tự tổng quát (chi tiết) của các giả thuyết.

Định nghĩa 2: Giả sử h_j và h_k là hai hàm giá trị bun xác định trên X. Ta nói h_j tổng quát hơn h_k (và viết là $h_j \geq_g h_k$) nếu và chỉ nếu:

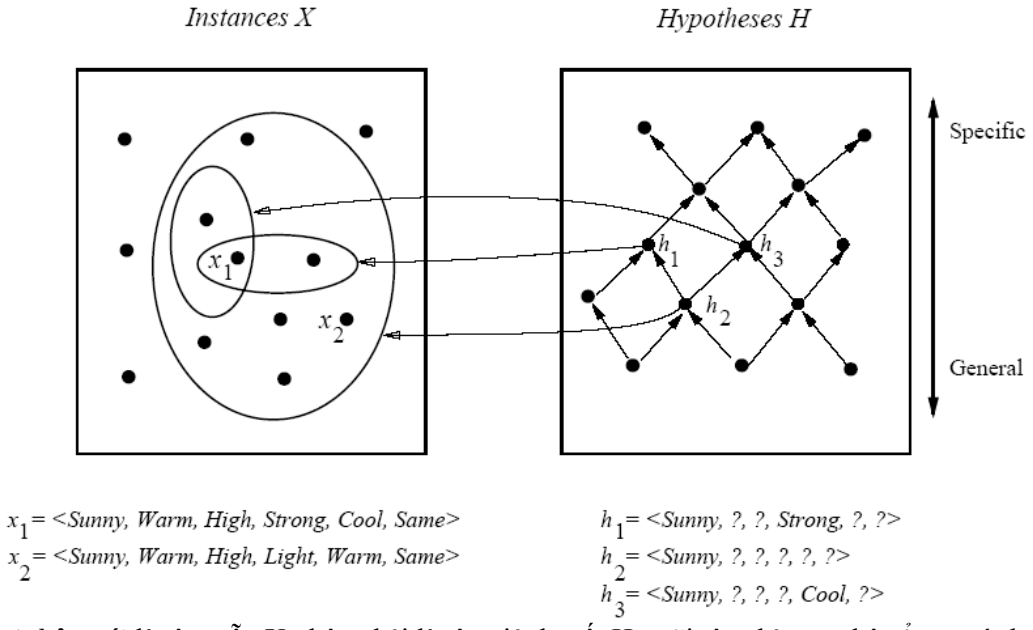
$(\forall x \in X)\ [h_k(x)=1 \rightarrow h_j(x)=1]$.

Ta cũng nói nói h_j tổng quát chặt hơn h_k (và viết là $h_j >_g h_k$) nếu và chỉ nếu:

$(h_j \geq_g h_k) \wedge (h_k \neq h_j)$.

Trong các trường hợp trên ta cũng nói h_k chi tiết (chặt) hơn h_j.

Để minh hoạ định nghĩa này ta xét ba giả thuyết h_1; h_2; h_3 trong hình 2.1 (ví dụ ưa thể thao của ta). Trong ví dụ này h_2 tổng quát hơn h_1 và h_3 còn h_1 và h_3 không so sánh được với nhau.



Hình 2.1. bên trái là tập mẫu X ; bên phải là tập giả thuyết H; mũi tên chỉ quan hệ *tổng quát hơn*

Chú ý rằng quan hệ $\geq_g$ và $>_g$ là độc lập với khái niệm đích ; chúng chỉ phụ thuộc vào các mẫu thoả mãn hai giả thuyết mà không phụ thuộc vào sự phân lớp các mẫu phù hợp với khái niệm đích.

Quan hệ $\geq_g$ rất quan trọng vì nó cho một cấu trúc có ích trên không gian giả thuyết H đối với mọi bài toán học khái niệm.

3.4. Fìnd-S: tìm một giả thuyết chi tiết nhất.

Để tìm một giả thuyết chi tiết nhất phù hợp với tập ví dụ đào tạo, ta bắầot giả thuyết chi tiết nhất và nhờ cấu trúc thứ tự đã biết, tổng quát hoá giả thuyết này mỗi khi có một mẫu dương không thoả mãn.

Thuật toán Find-S được trình bày trong bảng 2.3.

1. Khởi tạo h là giả thuyết chi tiết nhất: h $\leftarrow <\phi,\phi,\phi,\phi,\phi,\phi>$
2. Với mỗi mẫu đào tạo dương x thực hiện.

Với mỗi thuộc tính a_i trong h Nếu ràng buộc a_i thoả mãn thì giữ nguyên Ngược lại thay a_i trong h bởi ràng buộc tổng quát hơn thoả mãn x.
Bảng 2.3: Thuật toán Find-S

Chẳng hạn nếu đưa ví dụ 1 trong bảng 2.1:

x_1= < nắng, ấm, trung bình, mạnh, ấm, không đổi>

thì h← < nắng, ấm, trung bình, mạnh, ấm, không đổi>

Nếu đưa thêm ví dụ 2

x_2 = < nắng, ấm, cao, mạnh, ấm, không đổi>

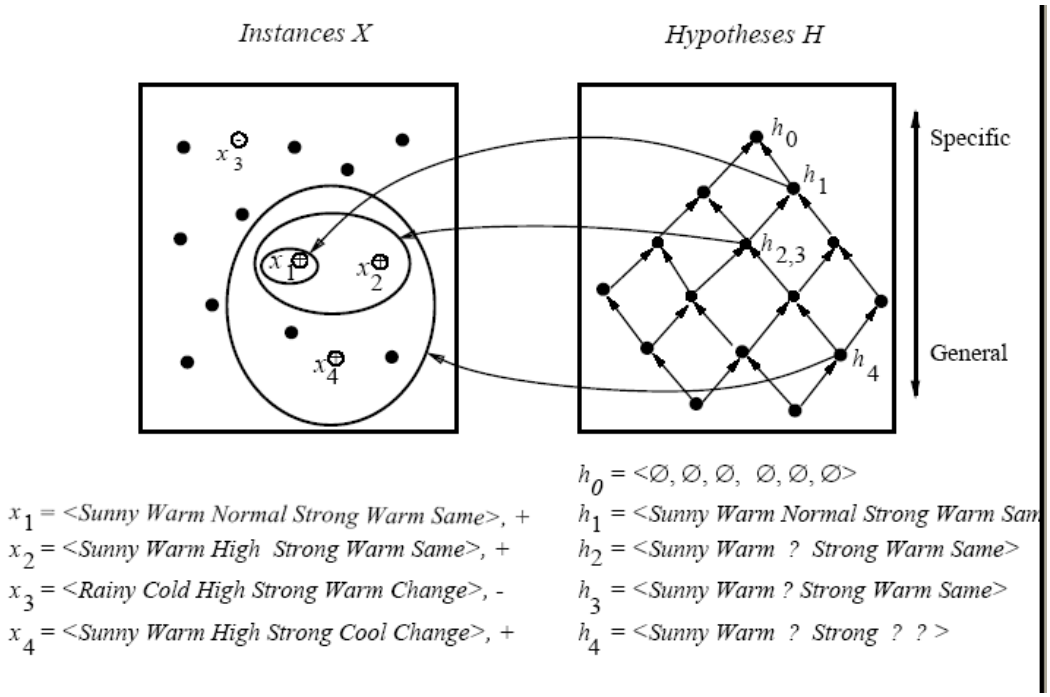
thì h← < nắng, ấm, ?, mạnh, ấm, không đổi>

Thuật toán này bỏ qua các thí dụ sai (thứ 3) và tới ví dụ thứ tư

x_4 = < nắng, ấm, cao, mạnh, lạnh, đổi>

thì h← < nắng, ấm, ?, mạnh, ?, ?> .

Ví dụ được minh hoạ trong hình 2.2.



Hình 2.2. thuật toán Find-S cho ví dụ ở bảng 2.1

- Thuật toán này đòi hỏi không *gian giả thuyết cho bởi liên kết các ràng buộc* của thuộc tính. Khi đó ta tìm được giả thuyết chi tiết nhất phù hợp với các mẫu dương và nó cũng sẽ đúng với mẫu âm khi giả thuyết đúng và tập mẫu cho đúng. Tuy vậy còn một số câu hỏi chưa trả lời đối với thuật toán này là:
- Thuật toán có hội tụ tới giả thuyết đúng không? (nó có học được khái niệm không?) Mặc dù thuật toán này cho ta giả thuyết phù hợp với tập mẫu đào tạo nhưng ta không biết có bao nhiêu giả thuyết phù hợp như vậy và liệu ta có thể tìm ra đúng 1 giả thuyết đúng hay không? Nếu không thì ta cũng cần ước lượng được tính không chắc chắn của nó.
- Tại sao ta ưa giả thuyết chi tiết nhất? các giả thuyết khác thì sao? (tổng quát nhất hoặc trung gian
- Nếu có ví dụ sai thì sao?

- Trong ví dụ này có một không gian giả thuyết H. Trong thực tế có thể có nhiều không gian giả thuyết và có thể có nhiều giả thuyết chi tiết nhất.

3.5. Không gian tường thuật và thuật toán loại trừ ứng cử.

Trong mục này đưa ra thuật toán cho ta các giả thuyết phù hợp với tập mẫu đào tạo.

3.5.1 Biểu diễn các giả thuyết.

Thuật toán trong mục này sẽ tìm các giả thuyết mô tả được thù hợp với các mẫu quan sát.

Trước hết ta cần định nghĩa.

Định nghĩa 3. Một giả thuyết h gọi là phù hợp với tập dữ liệu đào tạo D nếu và chỉ nếu h(x)=c(x) với mỗi <x,c(x)> trong D.

$$\text{consistent(h,D)} \equiv (\forall < x, c(x) > \in D) h(x) = c(x)$$

Chú ý khái niệm phù hợp ở đây khác khái niệm thoả mãn trong định nghĩa 1.

Thuật toán loại trừ ứng cử sẽ tìm tất cả các giả thuyết phù hợp với tập ví dụ quan sát được. Tập này gọi là không gian tường thuật đối với không gian giả thuyết H và tập mẫu đào tạo D.

Định nghĩa 4. Không gian tường thuật (được ký hiệu là $VS_{H,D}$ đối với không gian giả thuyết H và tập dữ liệu đào tạo D là tập con của H phù hợp với tập mẫu D.

$$VS_{H,D} = \{h \in H \mid \text{consistent}(h, D)\}$$

3.5.2 Thuật toán liệt kê loại trừ ứng cử:(List-then-eliminate algorithm)

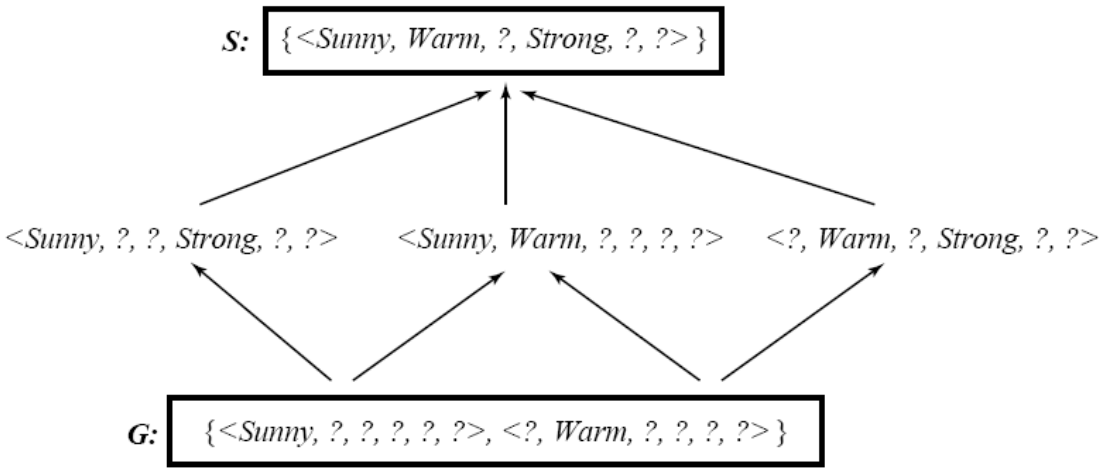
Một cách đơn giản để biểu diễn không gian tường thuật là liệt kê mọi phần tử của nó. Ý tưởng này dẫn tới thuật toán liệt kê-loại trừ ứng cử được mô tả trong bảng 2.4. Thuật toán này khởi tạo toàn bộ không gian giả thuyết rồi sau đó loại trừ dần các giả thuyết không phù hợp với các ví dụ quan sát được

Bảng 2.4. Thuật toán liệt kê loại trừ ứng cử
1. $V_{H,D} \leftarrow$ danh sách các giả thuyết trong H 2. Với mỗi <x,c(x)> trong D loại trừ các giả thuyết $h \in V_{H,D}$ mà $h(x) \neq c(x)$ 3. Đầu ra $V_{H,D}$

Về nguyên tắc thì thuật toán này có thể áp dụng khi không gian giả thuyết H hữu hạn nhưng trong thựng hành thì thường không thể liệt kê và tìm kiếm vét cạn H.

3.5.3 Một cách biểu diễn compact đối với khôn gian tường thuật.

Người ta có thể biểu diễn không gian tường thuật nhờ thứ tự tổng quát đến chi tiết bằng cách chỉ ra cận dưới chi tiết nhất và cận trên tổng quát nhất của $V_{H,D}$ mà không phải liệt kê chúng. Để minh hoạ cho cách biểu diễn này ta trở lại với bài toán học khái niệm thích chơi thể thao với các ví dụ trong bảng 2.1. Thuật toán Find-S cho ta giả thuyết chi tiết nhất là: < nắng, ấm, ?, mạnh, ?, ?> thực ra có sáu giả thuyết trong H phù hợp với tập mẫu trong bảng 2.1, các giả thuyết này được mô tả trong hình 2.3.



Hình 2.3 Không gian tường thuật với tập biên tổng quát và chi tiết cho bài toán học khái niệm ở bảng 2.1

Ý tưởng này dẫn đến thuật toán loại trừ ứng cử. Để trình bày thuật toán ta cần đến định nghĩa các tập biên tổng quát G và chi tiết S.

Định nghĩa 5. **Biên tổng quát** G đối với không gian giả thuyết H và tập dữ liệu đào tạo D là tập các giả thuyết tổng quát nhất của H phù hợp với D.

$$G \equiv \{g \in H \mid consistent(g, D) \wedge (\neg g' \in H)[g' >_g g) \wedge consistent(g', D)\}$$

Định nghĩa 6. **Biên chi tiết S** đối với không gian giả thuyết H và tập dữ liệu đào tạo D là tập các giả thuyết chi tiết nhất của H phù hợp với D.

$$S \equiv \{s \in H \mid consistent(s, D) \wedge (\neg s' \in H)[s >_g s') \wedge consistent(s', D)\}$$

Tập G và S này hoàn toàn có thể đặc tả $V_{H,D}$ nhờ định lý sau.

Định lý 2.1. Định lý biểu diễn không gian tường thuật. Giả sử X là tập mẫu tuỳ ývà H là tập giả thuyết giá trị Bun xác định trên X. Cho C:X$\rightarrow${0,1} là một khái niệm đích xác định trên X và D là tập mẫu đào tạo tuỳ ý của nó :{x,c(x)}Vowis mọi X, H,c và D sao cho G và S xác định thì

$$V_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s)\}.$$

3.5.4 Thuật toán loại trừ ứng cử.

Trong thuật toán này ta khởi tạo G và S bởi giả thuyết G_0 và S_0 là các giả thuyết tổng quát và chi tiết nhất:

$$G \leftarrow G_0 = <?,?,?,?,?,?>$$

$$S \leftarrow S_0 = <\phi,\phi,\phi,\phi,\phi,\phi>.$$

Khi mỗi mẫu đào tạo được xem xét, G và S được chi tiết hoá và tổng quát hoá để loại trừ khỏi $V_{H,D}$ các giả thuyết không phù hợp với dữ liệu tương ứng. Sau khi tập mẫu D đã được xét thì $V_{H,D}$ chỉ chứa các giả thuyết phù hợp với D trong H. Thuật toán được tóm tắt trong bảng 2.5.

Bảng 2.5. Thuật toán loại trừ ứng cử.

1. Khởi tạo G là tập giả thuyết tổng quát nhất trong H

   Khởi tạo S là tập giả thuyết chi tiết nhất trong H.

2.(Lặp)Với mỗi ví dụ đào tạo d={x,c(x)}, thực hiện:

   +Nếu d là ví dụ dương (c(x)=1)

      -lấy khỏi G các giả thuyết không phù hợp với d

      -Với mỗi s trong S không phù hợp với d:

- lấy s khỏi S
- thêm vào S các s tổng quát hoá chi tiết nhất h của smà h phù hợp với D và có một phần tử trong G tổng quát hơn h.
- Lấy khỏi S các giả thuyết tổng quát hơn các giả thuyết trong S

+Nếu d là ví dụ âm (c(x)=0)

      -lấy khỏi S các giả thuyết không phù hợp với d

-Với mỗi g trong G không phù hợp với d:

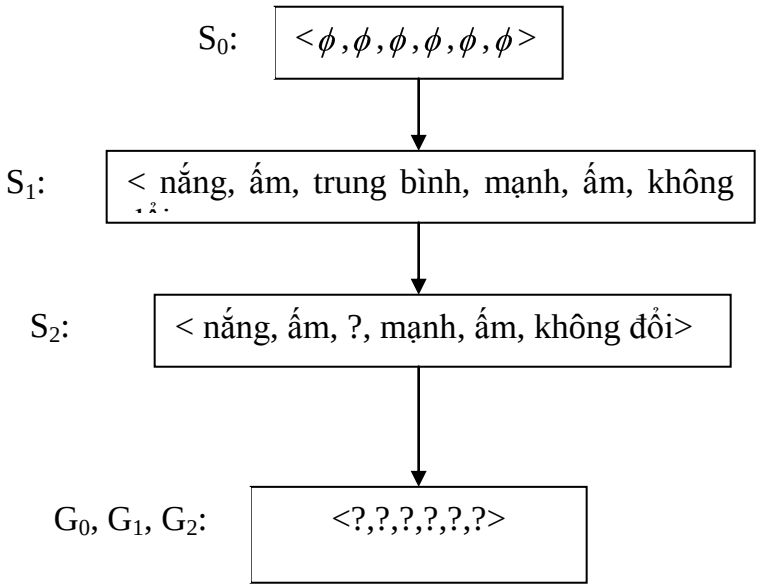
- lấy g khỏi G
- thêm vào G các chi tiết hoá nhỏ nhất h của g mà h phù hợp với D và có một phần tử trong S chi tiết hơn h.
- Lấy khỏi G các giả thuyết ít tổng uát hơn các giả thuyết trong G

Một ví dụ minh hoạ.

Hình 2.4 biểu diễn thuật toán khi áp dụng 2 ví dụ đầu trong bảng 2.1. Tập biên khởi tạo bởi G_0 và S_0 .

1:< nắng, ấm, trung bình, mạnh, ấm, không đổi> c=1

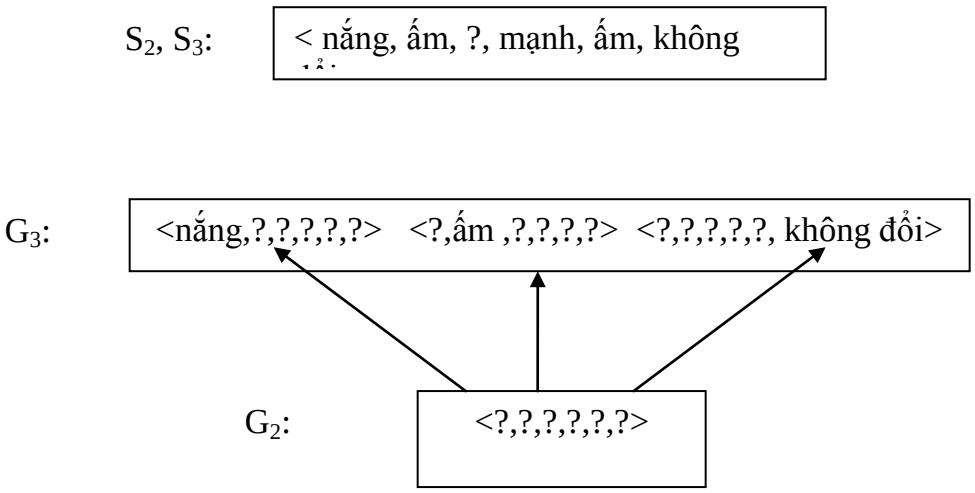
2:< nắng, ấm, cao, mạnh, ấm, không đổi> c=1



Hình 2.4. xử lý các ví dụ 1 và 2

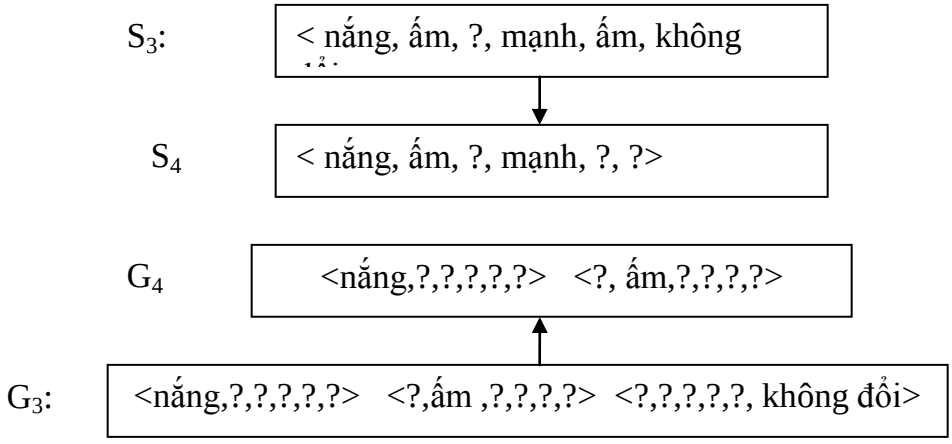
Ví dụ dương tăng tổng quát của biên chi tiết còn ví dụ âm tăng chi tiết của biên tổng quát.

Bây giờ ta xét ví dụ 3: :< mưa, lạnh, cao, mạnh, ấm, đổi> ;c=0 minh hoạ trong hình 2.5.



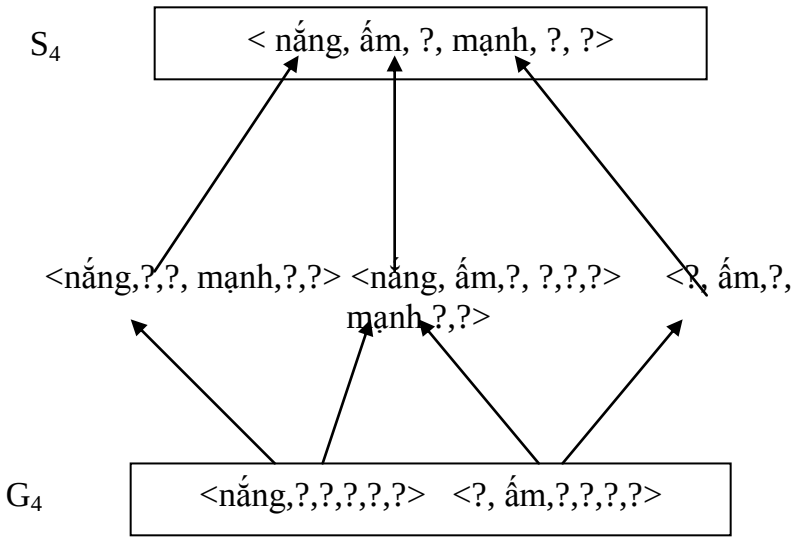
Hình 2.5 xử lý ví dụ 3

Ví dụ 4 4: <nắng, ấm, cao, mạnh,ấm, đổi> c=1 được xử lý trong hình 2.6.



Hình 2.6 xử lý ví dụ 4

Kết quả $V_{H,D}$ được mô tả trong hình 2.7



Hình 2.7: $V_{H, D}$

3.6. Các chú ý về thuật toán loại trừ ứng cử.

1) không gian tường thuật có hội tụ tới giả thuyết đúng không?

Hội tụ nếu mẫu đào tạo không có lỗi và trong H có giả thuyết đúng

2) Nên lấy ví dụ nào tiếp theo?

Trường hợp đực xét là ví dụ có thầy. Nếu ta phải tạo thí dụ thì nên chon thí dụ thoả mãn đúng một nửa không gian tường thuật hiện có như vâyh ta giảm được một nửavà ta tìm được giả thuyết đúng với $\log_2|VS|$ thí nghiệm

3) Nhận biết mẫu mới như thế nào. Giả sử $V_{H,D}$ là không duy nhất khi có ví dụ mới mà các giả thuyết đều thống nhất thì ta xác định được giá trị chung. Khi không thống nhất thì tuỳ theo tỷ lệ và phân bố xác suất để quyết định.

3.7. Khuynh hướng quy nạp.

- Thuật toán ta xét có giả thuyết là H chứa giả thuyết đúng. Xét ví dụ :

Các ví dụ về những ngày chơi hoặc không của A.

Ví dụ	Bầu trời	nhiệt độ	độ ẩm	gió	nước	dự báo	thích chơi
1	nắng	ấm	trung bình	mạnh	mát	đổi	có
2	mây	ấm	trung bình	mạnh	mát	đổi	có
3	mưa	ấm	trung bình	mạnh	mát	đổi	không

Giả thuyết chi tiết phù hợp với hai ví dụ đầu là S_2= < ?, ấm, trung bình, mạnh, mát, đổi> nhưng không đúng với ví dụ 3. Sở dĩ như vậy vì khuynh hướng học chỉ là các liên kết ràng buộc.

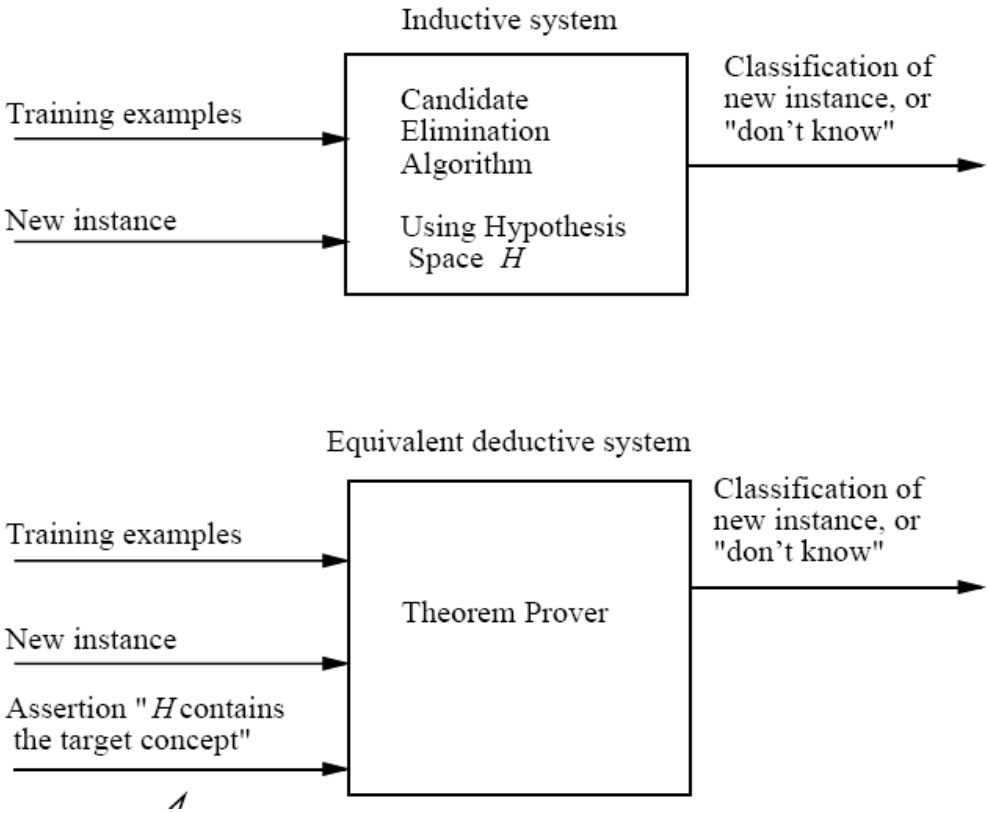
- Học không có khuynh hướng là xét mọi giả thuyết có thể .

Định nghĩa 7: Xét L là thuật toán học trên tập mẫu X. Giả sử C là khái niệm trên X, D_c = {x,c(x)}laf tập dữ liệu đào tạo. Giả sử L(x_i, D_c) là phân lớp gán cho x_i bởi L sau khi đào tạo trên D_c. Khuynh hường đào tạo của L là tập các khẳng định B sao cho với mọi khái niệm đích c và ví dụ đào tạo tương ứng D_c ta có:

$$(\forall x_i \in X)\left[B \wedge D_c \wedge x_i\right) \vdash L(x_i, D_c)$$

- Khuynh hướng quy nạp của thuật toán loại trừ ứng cử là khái niệm đích c thuộc vào không gian giả thuyết H.

Sơ đồ xử lý thuật toán trong hình dưới đây.



Chương 4 :Thực nghiệm

4.1 Giới thiệu về module mô phỏng thuật toán Find - S:

Chương trình mô phỏng em xây dựng trên Netbean 7.0 dùng ngôn ngữ lập trình Java.

4.1.1 Mô tả chung:

* Đầu vào:

- Xây dựng tập dữ liệu đầu vào (tập mẫu)

- Tạo mới tập dữ liệu đầu vào

- Mở tập dữ liệu đầu vào có sẵn

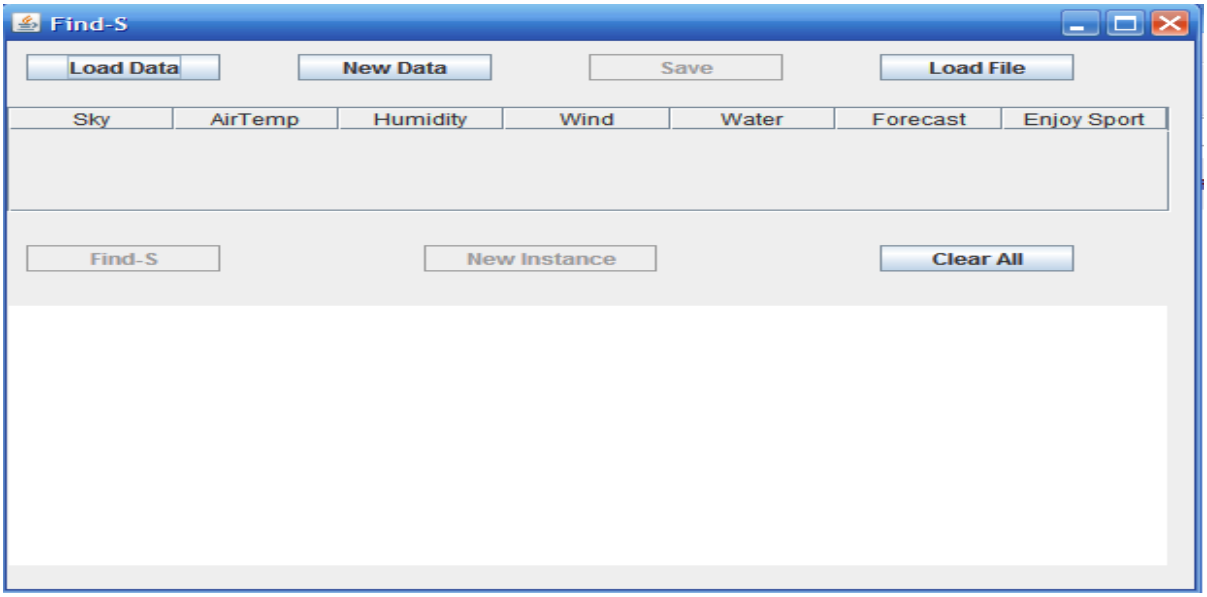
* Chạy chương trình để tìm được giả thuyết chi tiết nhất hay tập quy tắc dựa trên tập dữ liệu đầu vào.

* Đầu ra:

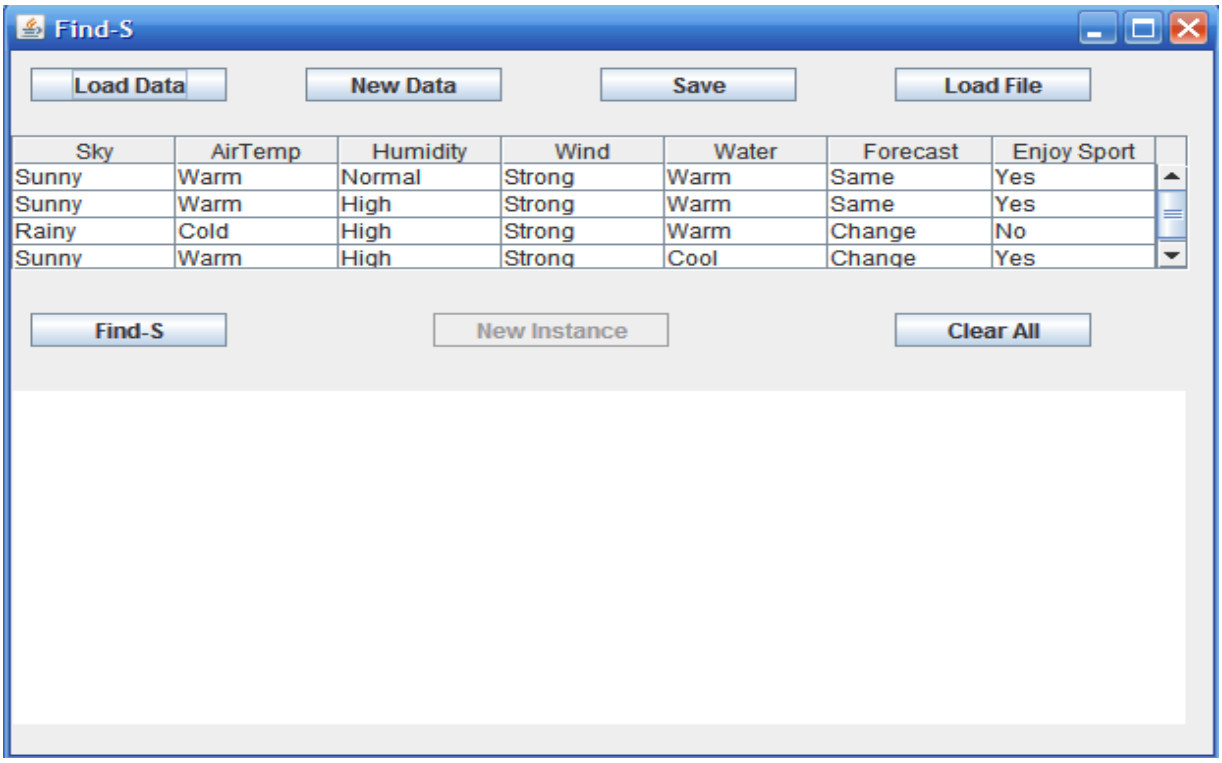
- Tập quy tắc đúng cho dữ liệu đầu vào

* Nhập vào một dữ liệu đầu vào tính ra kết quả (Yes ,No) dựa vào tập quy tắc.

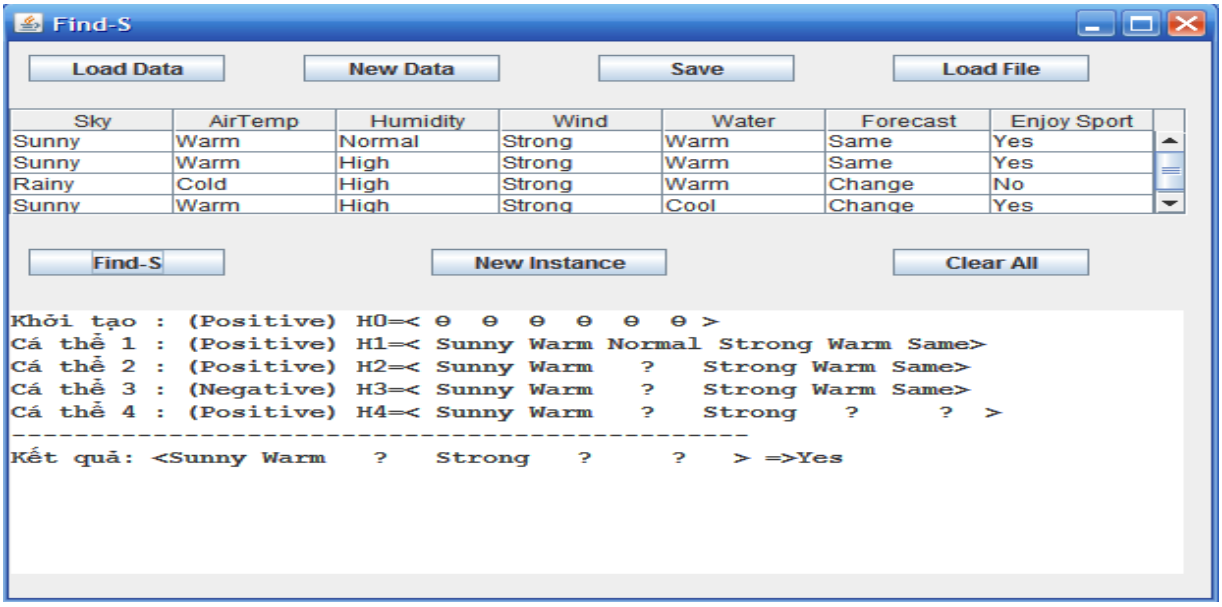
4.1.2 Chương trình :



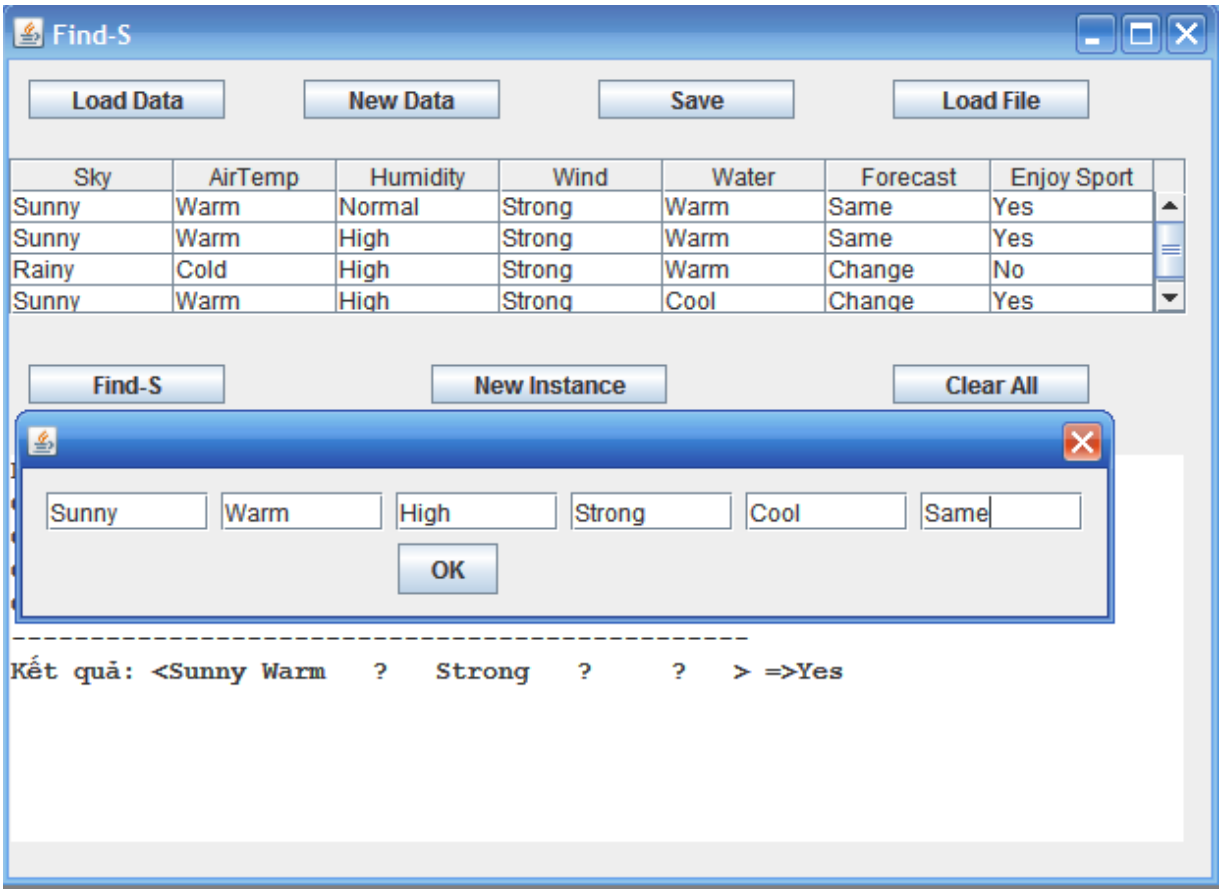
- Load Data:



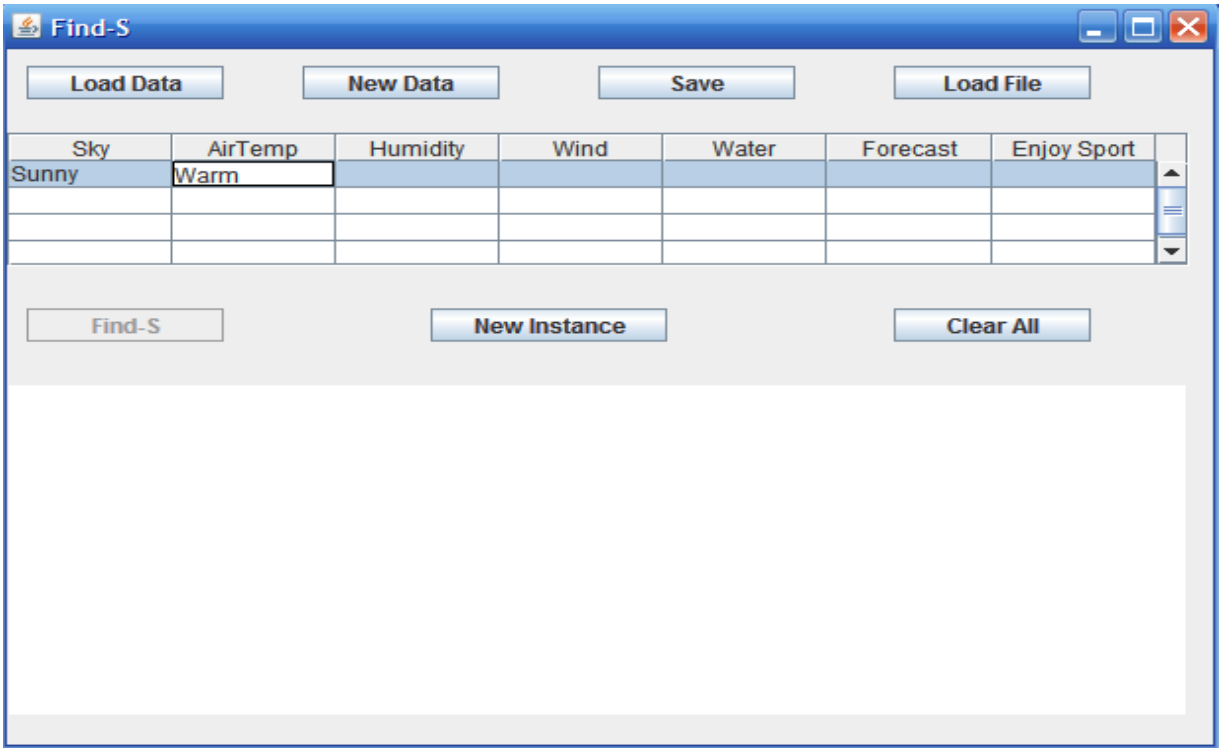
Chạy thuật toán:



Nhập mới phần tử và kiểm tra:



Nhập mới tập dữ liệu đầu vào:



Load File Data: