***Phương pháp rừng ngẫu nhiên trong khai phá dữ liệu và ứng dụng trong việc chẩn đoán bệnh***

CHƯƠNG II

PHƯƠNG PHÁP RỪNG NGẪU NHIÊN

VÀ ỨNG DỤNG TRONG VIỆC PHÂN LỚP

**2.1. Khái niệm về rừng ngẫu nhiên**

Thuật ngữ “rừng quyết định ngẫu nhiên”được Tin Kam Ho [10] đề xuất đầu tiên vào năm 1995, nhưng thuật toán *rừng ngẫu nhiên(Random Forest-RF)*là một thuật toán mới được sử dụng trong khoảng 10 năm gần đây, và có giá trị lớn trong những thuật toán học có giám sát (Surpervised Learning).

RF là một phương pháp phân lớp thuộc tính được phát triển năm 2001 bởi Leo Breiman tại đại học California, Berkeley.

Những cải tiến đáng kể về độ chính xác của phân loại đã dẫn đến việc phát triển một nhóm cây và để chúng bỏ phiếu cho lớp phổ biến nhất. Để phát triển các quần thể này, thường các vectơ ngẫu nhiên được tạo ra chi phối sự phát triển của mỗi cây trong quần thể. Một ví dụ ban đầu là đóng bao (Breiman, 1996), trong đó để trồng mỗi cây có một lựa chọn ngẫu nhiên (không thay thế) được thực hiện từ các mẫu trong tập huấn luyện.

Một ví dụ khác là việc chọn để phân nhánh một cách ngẫu nhiên (Dietterich, 1998) trong đó tại mỗi nút, việc tách được chọn ngẫu nhiên trong số K cách tách tốt nhất. Breiman (1999) tạo ra các tập hợp bằng cách ngẫu nhiên các đầu ra trong tập huấn luyện ban đầu. Một cách tiếp cận khác là chọn một tập huấn luyện từ một tập ngẫu nhiên với các trọng số trên các mẫu của tập huấn luyện ban đầu. Ho (1998) đã viết một số bài báo về phương thức không gian con ngẫu nhiên, trong đó lựa chọn ngẫu nhiên một tập hợp con các thuộc tính sẽ sử dụng để phát triển từng cây.

Trong một bài báo quan trọng về nhận dạng ký tự bằng văn bản, Amit và Geman (1997) đã tìm thấy một số lượng lớn các tính năng hình học và tìm kiếm một lựa chọn ngẫu nhiên trong số này để phân chia tốt nhất tại mỗi nút.

Theo Breiman, yếu tố cơ bản trong tất cả các quy trình ngẫu nhiên để phân lớp là đối với cây thứ k, một vectơ ngẫu nhiên k được tạo ra, không phụ thuộc vào các vectơ ngẫu nhiên trong quá khứ 1, ...,k-1 nhưng có cùng phân phối.

Mỗi cây được trồng bằng cách sử dụng tập huấn luyện và k, dẫn đến một lớp h(x,k) trong đó x là một vectơ đầu vào. Ví dụ, trong khi đóng gói, vectơ ngẫu nhiên được tạo dưới dạng số đếm trong N hộp do N phi tiêu ném ngẫu nhiên vào các hộp đó, trong đó N là số mẫu trong tập huấn luyện. Trong lựa chọn phân chia ngẫu nhiên bao gồm một số số nguyên ngẫu nhiên độc lập giữa1 và K. Bản chất và chiều của phụ thuộc vào việc sử dụng nó trong quá trình xây dựng cây. Sau khi một số lượng lớn cây được tạo ra, chúng bỏ phiếu cho lớp phổ biến nhất. Breiman và cộng sự gọi những quy trình này là rừng ngẫu nhiên.

Định nghĩa của Leo Breiman [8]:

*“Rừng ngẫu nhiên là một lớp bao gồm một tập hợp các lớp có cấu trúc cây {h(x,k), k=1,...} trong đó {k} là các vectơ ngẫu nhiên phân phối độc lập và mỗi cây tạo ra một đơn vị bỏ phiếu cho lớp phổ biến nhất với đầu vào x”.*

Breiman cũng đồng thời là đồng tác giả của phương pháp *CART (Classification and Regression Trees)* được đánh giá là 1 trong 10 phương pháp khai phá dữ liệu kinh điển.

RF là một bộ nhận dạng bao gồm một tập bộ phận lớp cơ sở dạng cây quyết định được kết hợp theo phương thức “*bỏ phiếu (vote)*”. Trong đó, các bộ phận lớp cơ sở được xây dựng từ các tập dữ liệu con với tập con đặc trưng khác nhau được lấy ngẫu nhiên từ tập dữ liệu quan sát được.

RF là một phương pháp phân lớp và hồi quy dựa trên việc kết hợp kết quả dự đoán của một số lượng lớn các cây quyết định. Trong mô hình RF truyền thống, mỗi cây quyết định được xây dựng từ tập dữ liệu được lấy ngẫu nhiên từ tập dữ liệu ban đầu và việc phát triển các nút con từ một nút cha dựa trên thông tin trong một không gian con các thuộc tính được chọn ngẫu nhiên từ không gian n thuộc tính ban đầu. Do đó, RF xây dựng các cây quyết định từ một tập con những thuộc tính được lựa chọn ngẫu nhiên và tổng hợp kết quả dự đoán của các cây để tạo ra kết quả dự đoán cuối cùng.

Các cây quyết định được xây dựng năm 1984 trên cơ sở sử dụng thuật toán CART [8] mà không thực hiện việc cắt tỉa do đó thu được những cây với độ lệch thấp. Bên cạnh đó, mối quan hệ tương quan giữa các cây quyết định cũng được giảm thiểu nhờ việc xây dựng các không gian con thuộc tính một cách ngẫu nhiên. Như vậy, sự chính xác của RF phụ thuộc vào chất lượng dự đoán của các cây quyết định và mức độ tương quan giữa các cây quyết định.

Như trên đã nói, Rừng ngẫu nhiên là một thuật toán học có giám sát. Như tên gọi của nó, “rừng ngẫu nhiên” sử dụng các *cây (tree)* để làm nền tảng [9].

Rừng ngẫu nhiên là một tập hợp của các cây quyết định (Decision Tree), mà mỗi cây được chọn theo một thuật toán dựa vào ngẫu nhiên. Cây quyết định là tên đại diện cho một nhóm thuật toán phát triển dựa trên cây quyết định. Ở đó, mỗi *nút (node)* của cây sẽ là các thuộc tính, và các nhánh là giá trị lựa chọn của thuộc tính đó bằng cách đi theo các giá trị thuộc tính trên cây.

Cây quyết định sẽ cho ta biết giá trị dự đoán.

Nhóm thuật toán cây quyết định có một điểm mạnh đó là có thể sử dụng cho cả bài toán Phân loại (Classification) và Hồi quy (Regression).

RF có những ưu điểm:

* Là phương pháp khá phổ biến để sàng lọc các cây được đóng gói (bagged trees).
* Có thể sử dụng cho cả bài toán phân loại và hồi quy.
* Làm việc được với dữ liệu thiếu giá trị hoặcvới dữ liệu có số chiều lớn.
* Có thể tránh được việc Overfitting với tập dữ liệu khi có nhiều cây.
* Có thể tạo mô hình cho các giá trị phân loại.
* Là một trong những phương pháp tập hợp mô hình thành công nhất.
* Kết hợp ý tưởng *"bagging" (đóng bao)* của Breiman và "*phương pháp không. gian con ngẫu nhiên*" của Tin-Kam-Ho để xây dựng một bộ sưu tập cây quyết định với các biến được kiểm soát [13].
* Mỗi cây được trồng trên mẫu “***bootstrap****”.*

*Bootstrap method* là tập hợp một số kỹ thuật phân tích dựa vào nguyên lý chọn mẫu có hoàn lại (sampling with replacement) chủ yếu dùng để ước lượng lỗi chuẩn (standard errors), độ lệch (bias) và tính toán khoảng tin cậy (confidence interval) cho các tham số, để ước tính các thông số mà thống kê thông thường không giải được.

Phương pháp Bootstrapping do nhà thống kê học Bradley Efron thuộc đại học Stanford (Mĩ) phát triển từ cuối thập niên 1979 nhưng đến khi máy tính được sử dụng phổ biến thì phương pháp này mới trở thành phương pháp phổ biến trong phân tích thống kê và được ứng dụng rộng rãi trong rất nhiều lĩnh vực khoa học. Boostrapping method được xem là phương pháp chuẩn trong phân tích thống kê và đã làm nên một cuộc cách mạng trong thống kê vì nó giải quyết được nhiều vấn đề mà trước đây tưởng như không giải được.

**Tư tưởng chính của Bootstrap method?**

Bootstrap method là phương pháp lấy mẫu có hoàn lại (sampling with replacement). Phương pháp lấy mẫu có hoàn lại có nghĩa là một cá thể (bản ghi) có thể xuất hiện nhiều lần trong một lần tạo tệp mẫu.

Giả sử ta có 5 quan sát (observation- mẫu có 1 quan sát (1 giá trị)) được gán nhãn A,B,C,D và E trên 5 quả bóng và bỏ tất cả chúng vào trong 1 cái giỏ.



Từ 5 quan sát này, ta lấy ra 1 quả bóng từ giỏ một cách ngẫu nhiên và ghi lại nhãn của chúng, sau đó bỏ lại quả bóng vừa bốc được vào giỏ và tiếp tục lấy ra một quả bóng một cách ngẫu nhiên, ghi lại nhãn của bóng và bỏ lại quả bóng vào trong giỏ và tiếp tục thực hiện việc lấy mẫu như vậy cho đến khi kết thúc. Việc lấy mẫu này gọi là lấy mẫu có hoàn lại. Kết quả của việc lấy mẫu như trên có thể như sau (giả sử kích thước mẫu (số lượng mẫu**/**số bản ghi) cần tạo là 10):

C

D

E

E

A

B

C

B

A

E

Nói cách khác, các quan sát (mẫu/dòng/bản ghi/ bản thể) có thể lặp lại trong tậpmẫu (bảng dữ liệu) và đó là đặc trưng của Bootstrap method.

Tư tưởng của Bootstrap method rất đơn giản là cách lấy mẫu có hoàn lại, vậy tại sao cần dùng Bootstrap method?

Trong thực tế, từ một mẫu ta chỉ có thể có được một số trung bình của mẫu, ta không biết được khoảng tin cậy cho số trung bình này hoặc không biết được phân bố của số trung bình ra sao. Thêm vào đó thực tế ta không biết được hàm phân bố của tổng thể nên việc ước lượng các tham số đặc trưng thông kê rất khó khăn và thiếu chính xác. Bootstrap method có thể cung cấp nhiều thông tin chi tiết hơn hơn về phân bố của số trung bình, khoảng tin cậy cũng như xác suất của số trung bình dựa trên một mẫu duy nhất.

*Bootstrap method xem một mẫu (sample) như một tổng thể (population-tập mẫu/quần thể)*.

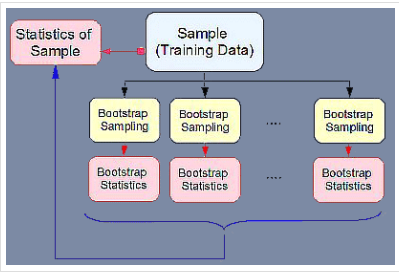
**Các bước chính của Bootstrap method:**

1.      Sinh ra các mẫu (Bootstrap sampling) ngẫu nhiên có hoàn lại kích thước n (tổng số cột) từ tổng thể (từ mẫu ban đầu).

2.      Tính các thông số thống kê đặc trưng cho của mẫu được sinh ra (mean, Confident interval, Standard Deviation, Inter Quartile,…)

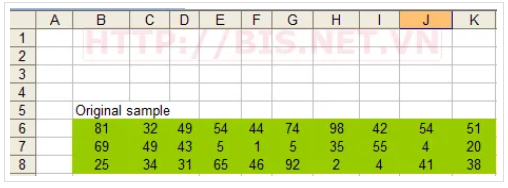
3.      Lặp lại bước 1 và bước 2 với số lần lớn (thường trên 1000)

4.      Sử dụng các ước lượng thống kê của Bootstrap sampling đã tính ở bước 2 để đánh giá độ chính xác các ước lượng thống kê của mẫu ban đầu (Original sample/ Training Data).



Ví dụ sau đây trình bày cách lấy mẫu có hoàn lại và cách tính các tham số đặc trưng thống kê dựa trên Bootstrap method. Ví dụ được minh họa trên  trên MS Excel.

Giả sử rằng ta có tậpmẫu dữ liệu ban đầu (original sample) gồm 3 quan sát như sau:



1. Định nghĩa tên “sample” tham chiếu đến vùng chứa tập dữ liệu  mẫu ban đầu ($B$6:$K$8)

2. Từ mẫu ban đầu, lấy mẫu với Bootstrap method (lấy mẫu có hoàn lại) như sau:

Giả sử ta cần lấy 201 mẫu, mỗi mẫu gồm 10 quan sát (10 giá trị của 10 thuộc tính), được lấy ngẫu nhiên (có lặp lại) từ mẫu ban dầu. Từ ô nào đó trong bảng tính (giả sử ô B10), sử dụng công thức sau (chú ý trong công thức sau “sample” là tên đã định nghĩa ở bước 1, là vùng chứa dữ liệu mẫu ban đầu):

*=INDEX(sample, ROWS(sample)\*RAND()+1,COLUMNS(sample)\*RAND()+1)*

3. Copy công thức từ ô B10 tới B10:K210 (hoặc nhiều hơn nếu muốn). Ta có được một bootstrap sample (mỗi hàng là một mẫu với 10 quan sát và ta có 201 mẫu)

4. Bước tiếp theo ta tính các thống kê của Bootstrap (bootstrap statistics). Có thể tính các thống kê nào mà chúng ta muốn. Trong ví dụ này chỉ tính các thông kê đặc trưng như số trung bình (mean), trung vị (median), khoảng cách giữa QT3- QT1 (inter quartile range)  và độ lệch chuẩn (standard deviation).

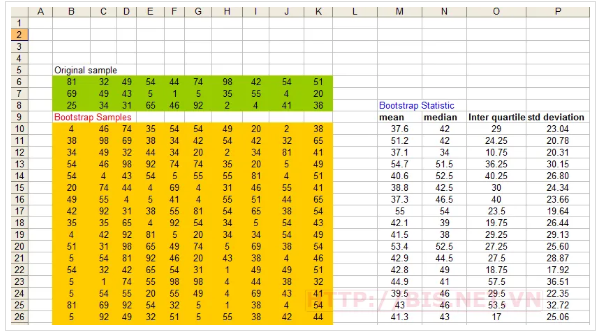
*Tính mean của 1 mẫu  =AVERAGE(B10:K10)*

*Tính số trung vị =MEDIAN(B10:K10)*

*Tính inter quartile range =QUARTILE(B10:K10,3)-QUARTILE(B10:K10,1)*

*Tính độ lệch chuẩn =STDEV(B10:K10)*

Các công thứ trên tính cho 1 mẫu vì vậy ta phải copy  các công thức trên để tính cho các mẫu còn lại. Kết quả như sau :

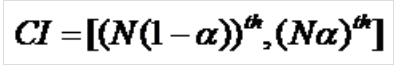


*Chú ý: Để có một bootstrap sample mới thì bấm phím F9*

**Khoảng tin cậy (Bootstrap Confidence interval)**

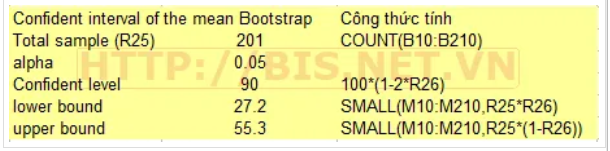
Độ tin cậy (Confident level): 100(1-2α). Trong đó α là mức ý nghĩa (thường dùng 5%)

Khoảng tin cậy được xác định như sau:



Trong đó, N: Tổng số mẫu, α: Mức ý nghĩa.

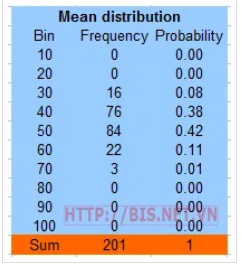
Sử dụng hàm SMALL trong MS Excel để xác định khoảng tin cậy [lower bound, upper bound] như sau:



Chú ý rằng vì phân bố mẫu có thể không tuân theo phân phối chuẩn (Normal distribution) nên ta không thể dùng công thức tính khoảng tin cậy dựa vào số trung bình và mức ý nghĩa α.

***Đồ thị phân bố tần suất của Bootstrap***

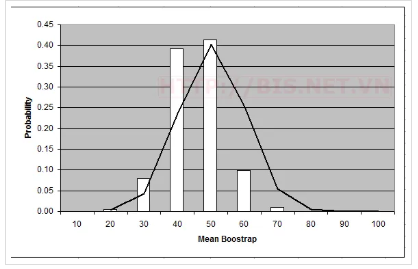
Để vẽ đồ thị phân bố tần suất của số trung bình (mean) từ Bootstrap sample đã tạo ra, ta tạo ra các khoảng (bin) và tính tần suất dùng hàm Frequency trong Excel. Vì giá trị của mẫu trong khoảng từ 0 đến 100 nên ta tạo ra 10 bin như sau:



Giả sử rằng vùng chứa giá trị 10 khoảng (Bin) là R10:R19 và Bootstrap sample là vùng  M10:M210.

Công thức tính tần suất: {=FREQUENCY(M10:M210,R10:R19)}. Đây là công thức mảng nên phải bấm phím Ctrl+Shift+Enter để thực hiện

Sau khi tính được tần suất, ta tính xác suất xuất hiện của các Bin và vẽ đồ thị phân phối của Bin và xác suất như sau:



Sử dụng Bootstrap method, ta không cần biết phân phối thực sự của tổng thể -tập dữ liệu huấn luyện (thực tế rất khó biết), chỉ với một tệp mẫu dữ liệu ban đầu, thông qua phương pháp lấy mẫu có hoàn lại, ta có thể sinh ra nhiều tệp mẫu mới theo yêu cầu nghiên cứu, từ đó ta có thể ước lượng được các tham số đặc trưng của nghiên cứu thống kê như (khoảng tin cậy, phương sai, độ lêch chuẩn,…).

Ý tưởng chìa khóa để làm nên thành công của Bootstrap method là “*đối xử với mẫu như là tổng thể*” cùng với phương pháp lấy mẫu có hoàn lại.

Cònphương pháp*Bagging*được xem như là một phương pháp tổng hợp kết quả có được từ các bootstrap.Tư tưởng chính của phương pháp này như sau:

* Cho một tập huấn luyện D={(xi, yi): xi∈ Rn}, (i=1,…,m) và giả sử chúng ta muốn có một một dự đoán nào đó đối với biến x∉ D.
* Một mẫu gồm T tập dữ liệu, mỗi tập dữ liệu gồm n phần tử được chọn lựa (xi) ngẫu nhiên từ D với sự thay thế (bootstrap). Do đó T=(D1, D2, ….,DT) trông “giống” như là một tập các tập huấn luyện được nhân bản;
* Tập huấn một máy hoặc một mô hình đối với mỗi tập Dt (t=1, 2, …,T) và lần lượt thu thập các kết quả dự báo có được trên mỗi tập này;
* Kết quả tổng hợp cuối cùng được tính toán bằng cách trung bình hóa (với bài toán regression) hoặc thông qua số *phiếu bầu (vote)* nhiều nhất (với bài toán classification).

**2.2. Phương pháp phân lớp rừng ngẫu nhiên**

Ý tưởng chính của phương pháp rừng ngẫu nhiên thể hiện ở chỗ:

* Tại mỗi lần trồng cây trên mẫu “*bootstrap”*, một mẫu ngẫu nhiên có k *thuộc tính(features)* được chọn từ n thuộc tính của tập dữ liệu huấn luyện ban đầu. Breiman gợi ý chọn k = ký hiệu .
* Đối với mỗi cây được trồng trên mẫu “*bootstrap”*, tỷ lệ lỗi cho các quan sát còn lại trong mẫu bootstrap đều được giám sát và được gọi là tỷ lệ lỗi “Out-of-bag- OOB”.
* Rừng ngẫu nhiên *đóng bao*các cây theo quan hệ “tương quan”. Mỗi cây được giả thiết là có cùng một kỳ vọng.

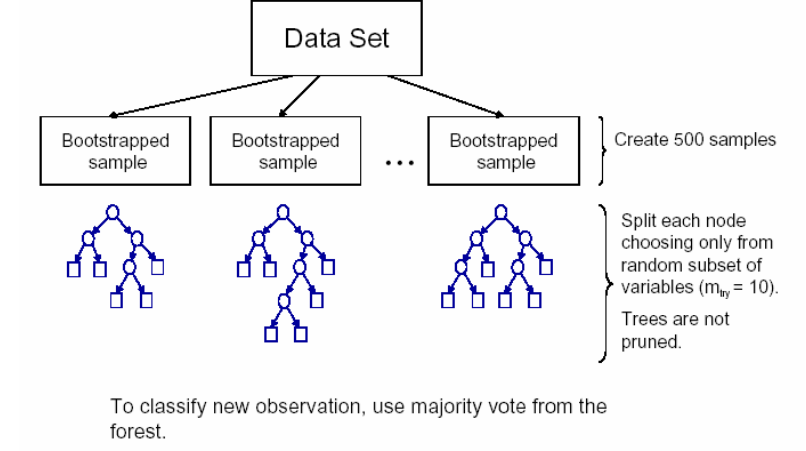
*Random Forest được xây dựng dựa trên 3 thành phần chính là:*

(1) Thuật toán xây dựng cây quyết định *CART (Classification & Regression Tree)*;

(2) Học máy;

(3) Tổng hợp theobootstrap /bagging.

Hình dưới đây thể hiện các bước phân lớp Random forest:

*Hình* …. Minh họa các bước trong Random Forests

**2.3. Đặc trưng của cách tiếp cận rừng ngẫu nhiên**

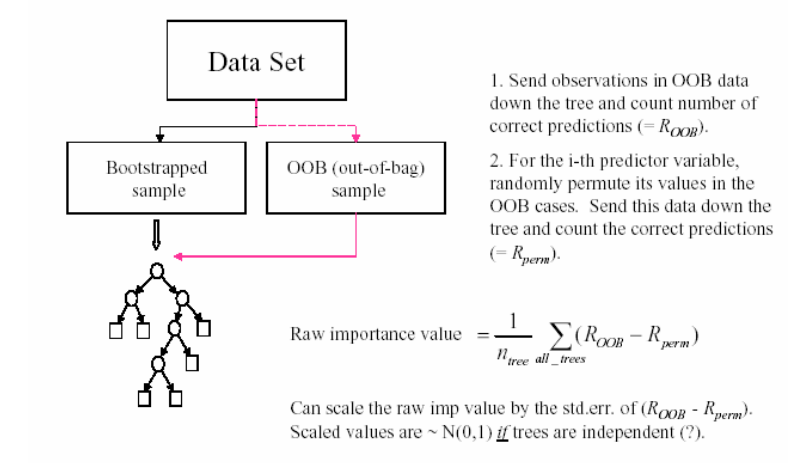
Giải thuật rừng ngẫu nhiên tạo ra một tập hợp các cây quyết định không cắt tỉa, mỗi cây được xây dựng trên tập mẫu bootstrap, Bagging, tại mỗi nút phân hoạch tốt nhất được thực hiện từ việc chọn ngẫu nhiên một tập con các thuộc tính. Tỷ lệ lỗi tổng quát của rừng phụ thuộc vào độ chính xác của từng cây thành viên trong rừng và sự phụ thuộc lẫn nhau giữa các cây thành viên. Giải thuật rừng ngẫu nhiên xây dựng cây không cắt tỉa nhằm giữ cho thành phần lỗi bias thấp và dùng tính ngẫu nhiên để điều khiển tính tương quan thấp giữa các cây trong rừng. Tiếp cận rừng ngẫu nhiên có độ chính xác cao khi so sánh với các thuật toán có giám sát hiện nay. Rừng ngẫu nhiên học nhanh, chịu đựng nhiễu tốt. Giải thuật rừng ngẫu nhiên là mô hình có độ chính xác cao đáp ứng được yêu cầu thực tiễn cho vấn đề phân loại, hồi quy.

Trong phương pháp RF, có hai yếu tố quan trọng là:

1-Dữ liệu Out-Of-Bag –OOB

2-Thuộc tính quan trọng

Khi tập mẫu được rút ra từ một tập huấn luyện của một cây với sự thay thế (bagging), thì theo ước tính có khoảng 1/3 các phần từ không có nằm trong mẫu này. Điều này có nghĩa là chỉ có khoảng 2/3 các phần tử trong tập huấn luyện tham gia vào trong các tính toán của chúng ta, và 1/3 các phần tử này được gọi là dữ liệu Out-Of-Bag (OOB). Dữ liệu OOB được sử dụng để ước lượng lỗi tạo ra từ việc kết hợp các kết quả từ các cây tổng hợp trong random forest cũng như dùng để ước tính độ quan trọng của thuộc tính (variable important).



*Hình* ...: Cách đánh giá Out-Of-Bag và tính quan trọng của biến

Việc thực hiện các tính toán để xác định thuộc tính quan trọng trong RF cũng gần như tương tự việc sử dụng OOB để tính toán lỗi trong RF. Cách thực hiện như sau: Giả sử chúng ta cần xác định “thuộc tính quan trọng” của thuộc tính thứ m. Đầu tiên tính ROOB, sau đó hoán vị ngẫu nhiên các giá trị của thuộc tính m trong dữ liệu OOB, lần lượt “gửi” các giá trị này xuống cây và “đếm” số các dự đoán đúng ta gọi việc tính toán này đối với thuộc tính là “Rperm”.

Độ quan trọng thuộc tính được tính như sau: Trong trường hợp giá trị của thuộc tính quan trọng trên mỗi cây là độc lập thì chúng ta có thể tính được lỗi chuẩn (standard error) của ROOB – Rperm.

Cách làm việc của rừng ngẫu nhiên như sau:

Chúng ta có thể nghĩ đến một ví dụ đơn giản trong cuộc sống, giả sử chúng ta muốn tìm hiểu một địa danh cho chuyến du lịch sắp tới, sẽ cần đi hỏi một người bạn để tham khảo ý kiến.

Nhưng, ý kiến của người bạn này có thể không khách quan cho lắm. Chúng ta liền đi hỏi thêm một vài người nữa, và tổng hợp lại để cho ra quyết định đi hay không. Nếu coi mỗi ý kiến của những người góp ý là một cây quyết định, thì chúng ta đã có hình dung phần nào được Random Forest rồi.

Random Forest hoạt động bằng cách đánh giá nhiều cây quyết định ngẫu nhiên, và lấy ra kết quả được đánh giá tốt nhất trong số kết quả trả về.

Người ta tạo ra được một Random Forest theo quy trình sau:

* Bước 1: Chọn ngẫu nhiên **“**k**”** thuộc tính (features) từ tập **“**n**”** thuộc tính của tập dữ liệu mẫu ban đầu.Chú ý chọn k <<n.
* Bước 2: Từ tập có “k” features, tính toán ra node **“**d**”** là tốt nhất dùng để phân nhánh.
* Bước 3: Chia các node con theo node tốt nhất vừa tìm được
* Bước 4: Lặp lại bước 1-3cho đến khi đạt đến k node. Kết quả sẽ được 1 cây.
* Bước 5: Lặp lại Tbước 1-4để tạo ra **“**T**”**cây (rừng có T cây)

Các hoạt động của rừng ngẫu nhiên để dự đoán như sau:

Sau khi Random Forest đã được huấn luyện,người sử dụng các bước sau để dự đoán:

* Bước 1: Chọn lọc các thuộc tính cần kiểm thử (test features)và sử dụng phương pháp Cây quyết định đã tạo ra để dự đoán kết quả, lưu nó vào một danh sách (tạo 1 cây).
* Bước 2: Tính toán số lượng *ứng viên (vote)* trên toàn rừng cho từng kết quả.
* Bước 3: Lấy kết quả có số lượng bỏ phiếu lớn nhất làm kết quả cuối cho mô hình.

Quá trình học của Random Forest bao gồm việc sử dụng ngẫu nhiên giá trị đầu vào, hoặc kết hợp các giá trị đó tại mỗi node trong quá trình dựng từng câyquyết định. Kết quả của Random Forest, qua thực nghiệm cho thấy là tốt hơn khi so sánh với một số thuật toán khác vì Random Forest có một số ưu thế như:

* Độ chính xác của nó tương tự nhau, trong một số trường hợp còn tốt hơn.
* Thuật toán giải quyết tốt các bài toán có nhiều dữ liệu nhiễu.
* Thuật toán chạy nhanh hơn so với bagging hoặc boosting.
* Có những sự ước lượng nội tại như độ chính xác của mô hình phỏngđoán hoặc độ mạnh và liên quan giữa các thuộc tính.
* Dễ dàng thực hiện song song.
* Tuy nhiên để đạt được các tính chất ưu việt trên, thời gian thực thi củathuật toán khá lâu và phải sử dụng nhiều tài nguyên của hệ thống.

Như vậy, RF là một phương pháp phân lớp tốt do:

(1) Trong RF các sai số (variance) được giảm thiểu do kết quả của RF được tổng hợp thông qua *nhiều người học (learners);*

(2) Việc chọn ngẫu nhiên tại mỗi bước trong RF sẽ làm giảm mối tương quan (correlation) giữa các người học trong việc tổng hợp các kết quả.

Ngoài ra, chúng ta cũng thấy rằng tỷ lệ lỗi của rừng cây phân lớp phụ thuộc vào lỗi riêng của từng cây trong rừng cũng như mỗi tương quan giữa các cây.

**2.4. Giải thuật rừng ngẫu nhiên**

Từ những năm 1990, cộng đồng máy học đã nghiên cứu cách để kết hợp nhiều mô hình phân loại thành tập hợp các mô hình phân loại để cho tính chính xác cao hơn so với chỉ một mô hình phân loại. Mục đích của các mô hình tập hợp là làm giảm variance và/hoặc bias của các giải thuật học.

Bias là khái niệm về lỗi của mô hình học (không liên quan đến dữ liệu học) và variance là lỗi do tính biến thiên của mô hình so với tính ngẫu nhiên của các mẫu dữ liệu học. (Buntine, 1992) đã giới thiệu các kỹ thuật Bayes để giảm variance của các phương pháp học. Phương pháp xếp chồng (Wolpert, 1992) hướng tới việc cực tiểu hóa bias của các giải thuật học. Trong khi (Freund and Schapire, 1997) đưa ra Boosting,… thì giảm variance của giải thuật học nhưng không làm tăng bias quá nhiều.

Phương pháp rừng ngẫu nhiên [8-Breiman] là một trong những phương pháp tập hợp mô hình thành công nhất. Giải thuật rừng ngẫu nhiên xây dựng cây không cắt tỉa nhằm giữ cho bias thấp và dùng tính ngẫu nhiên để điều khiển tính tương quan thấp giữa các cây trong rừng.

***2.4.1.Ý tưởng cuả giải thuật Random forest***

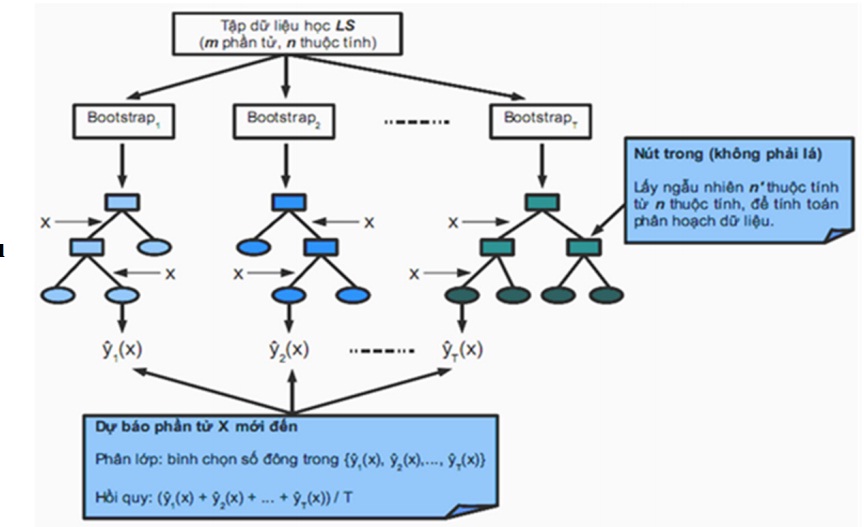
Ý tưởng cuả giải thuật Random Forest (RF) cho phân lớp được diễn giải như sau:

* Lấy ra T bộ mẫu bootstrap từ tập huấn luyện (rừng có T cây).
* Đối với mỗi mẫu bootstrap, xây dựng một cây phân lớp chưa được tỉa (unpruned tree) theo hướng dẫn sau: Tại mỗi nút thay vì chọn một phân chia tốt nhất trong tất cả các biến dự đoán, ta chọn ngẫu nhiên một mẫu m của các biến dự đoán sau đó chọn một phân chia tốt nhất trong các biến này.
* Đưa ra các dự đoán bằng cách tổng hợp các dự đoán của T cây.

*Ý tưởng của giải thuật rừng ngẫu nhiên có thể được diễn giải theo cách khác:*

* Từ tập dữ liệu huấn luyện (Learning Samples-LS) có m phần tử (bộ/dòng) và n biến (thuộc tính/cột), xây dựng T cây quyết định một cách độc lập nhau.
* Mô hình cây quyết định thứ t (t=1,...,T) được xây dựng trên tập mẫu Bootstrap thứ t (lấy mẫu m phần tử có hoàn lại từ tập huấn luyện).
* Tại nút trong(không phải nút gốc hay nút lá), chọn ngẫu nhiên n' biến (n'<<n) và tính toán phân hoạch tốt nhất dựa trên n' biến này.
* Cây được xây dựng đến độ sâu tối đa không cắt tỉa.

Kết thúc quá trình xây dựng T mô hình cơ sở, dùng chiến lược bình chọn số đông (bỏ phiếu) để phân lớp một phần tử(bộ/dòng).



Hình ..: Giải thuật rừng ngẫu nhiên

***2.4.2.Thuật toán xây dựng rừng ngẫu nhiên***

Người ta chia quá trình xây dựng rừng ngẫu nhiên thành ba pha:

* Pha 1: Tạo ngẫu nhiêntập dữ liệu từ tập mẫu ban đầu
* Pha 2: Xây dựng các cây cơ sở
* Pha 3: Kết hợp các cây cơ sở theo phương thức bỏ phiếu (lấy số đông).

Trong pha tạo tập ngẫu nhiên dữ liệu: Từ tập mẫu học ban đầu LS,

LS , Rngồm m mẫu, n thuộc tính (bảng gồm m hàng và n thuộc tính độc lập) chọn ngẫu nhiên n’đặc trưng (thuộc tính): n’(<n) và chọn ngẫu nhiên mẫu (hàng): m’ (<m) từ LS rồi chiếu nó lên các đặc trưng được chọn này tạo thành một tập mẫu học mới LSt(t=1,…,T) để tạo cây quyết định Tt (t=1,…,T), T là số cây quyết định tạo thành rừng) [1],

Chú ý, số đặc trưng n’theoBreiman gợi ý nên chọn đối với bài toán phân lớp là n’=, trong đólà ký hiệu phần nguyên của x.

Từ các pha trên, người ta xây dựng thuật toán rừng ngẫu nhiên như sau:

Thuật toán xây dựng rừng ngẫu nhiên

gồm 2 bước:

Bước 1: Với (t=1,2,…,T) thực hiện:

* Lấy ngẫu nhiên n’đặc trưng (thuộc tính) {}của LS (tập dữ liệu huấn luyện ban đầu);
* Lấy ngẫu nhiên tập Rt gồm m’ngẫu nhiên hàng dữ liệu trong LS;
* LSt = Hình chiếu của Rtlên các đặc trưng {};
* Xây dựng cây quyết định Tttừ tập LSt; Ttcó bộ nhận dạng Ct

Bước 2: Kết hợp bỏ phiếu trọng số đều

Đầu ra của hệ cho đối tượng x sẽ là:

+ Đối với bài toán hồi quy: C () = (;

+ Đối với bài toán phân lớp: Kết hợp các cây, sử dụng chiến lược bình chọn theo số đông.

**2.5.Quá trình phát triển ứng dụng phương pháp rừng ngẫu nhiên**

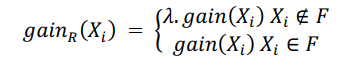
Phương pháp rừng ngẫu nhiên có khả năng ứng dụng rộng rãi trong kinh tế -xã hội như trong các lĩnh vực ngân hàng (Banking), y tế (Medicine), thương mại điện tử (E-Commerce) và thị trường chứng khoán (Stock Market).

Trong thực tế RF đã trở thành một công cụ tin cậy cho phân tích dữ liệu, đặc biệt là dữ liệu tin sinh học (Bureau *et al.*, 2005; Goldstein *etal.*, 2010; Goldstein *et al.*, 2011; Winham *et al.*,2012). Tuy nhiên, tiếp cận RF ban đầu của Breiman chỉ hiệu quả cho phân tích dữ liệu có số chiều thấp (Bureau *et al.*, 2005; Lunetta *etal.*, 2004). Mô hình RF truyền thống không thểáp dụng trên dữ liệu có số chiều lớn, có thể lên đến hàng ngàn hay trăm ngàn gen. Nguyên nhân là trong quá trình xây dựng cây quyết định, tại mỗi nút, RF sử dụng một tập con những thuộc tính được lựa chọn ngẫu nhiên từ tập thuộc tính ban đầu. Vì vậy khi xử lý với các dữ liệu nhiều chiều như dữ liệu gen, RF có thể lựa chọn ngẫu nhiên những gen không có ảnh hưởng đến biến đích và từ đó tạo ra cây quyết định có chất lượng dự đoán thấp.

Gần đây, một số phương pháp rừng ngẫu nhiên cải tiến đã được đề xuất để thực hiện lựa chọn các thuộc tính giúp cải thiện quá trình lựa chọn thuộc tính và tăng hiệu quả dự đoán với các bộ dữ liệu nhiều chiều và nhiều nhiễu như phương pháp rừng ngẫu nhiên điều hòa (Regularized Random Forest- RRF) (Deng and Runger, 2012), rừng ngẫu nhiên điều hòa có điều hướng (Guided Regularized Random Forests-GRRF) (Deng and Runger, 2013) và phương pháp rừng ngẫu nhiên có điều hướng (Guided Random Forest) (Deng, 2013).

Như đã phân tích ở trên, RF nguyên bản của Breiman không phù hợp cho phân tích dữ liệu biểu hiện gen có số chiều lớn, vì việc lấy mẫu trong không gian con thuộc tính có thể dẫn tới việc chọn phải những mẫu không tốt và kết quả là nhiều cây quyết định có chất lượng thấp, dẫn đến giảm khả năng dự đoán của RF. Để khắc phục nhược điểm này, năm 2012, Deng và Runger đề xuất mô hình rừng ngẫu nhiên điều hòa (RRF). Các tác giả đã thay đổi cách tính độ đo cho mỗi thuộc tính để giảm số thuộc tính mới được chọn cho việc thực hiện phân tách nút tại bước xây dựng cây. Nếu thuộc tính mới Xi và Xj có độ quan trọng là như nhau mà thuộc tính Xj đã từng được chọn để phân tách nút thì RRF ưu tiên chọn thuộc tính Xj. Thuộc tính mới Xi chỉ được chọn khi chỉ số *gain* của Xi lớn hơn chỉ số *gain* của tất cả các thuộc tính đã được chọn trong các nút trước.

Gọi F là tập các thuộc tính đã được sử dụng ở các lần chia trước trong mô hình rừng. Độ đo mới của các thuộc tính được tính như sau:



* ở đây *λ*€ [0, 1] là hệ số phạt; λ càng nhỏ thì phạt càng lớn đối với những thuộc tính không thuộc tập *F*. RRF sử dụng *gainR(·)* để tách nút.
* Một số phương pháp rừng ngẫu nhiên cải tiến có thể kiểm lại như sau:

*1. Rừng ngẫu nhiên điều hòa có điều hướng (GRRF)*

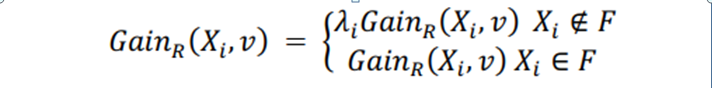
Trong phương pháp rừng ngẫu nhiên điều hòa, Deng *et al.* (2012) đã thay đổi cách tính độ đo quan trọng của mỗi thuộc tính do đó RRF làm giảm độ lệch (bias) so với RF nguyên bản. Tuy nhiên các chỉ số đo độ quan trọng thuộc tính này được đánh giá dựa trên một phần của dữ liệu huấn luyện tại mỗi nút của cây so với tất cả các thuộc tính đã được chọn để xây dựng cây trong rừng.

Mặt khác đối với các tập dữ liệu có số mẫu nhỏ, số chiều lớn thì có rất nhiều các thuộc tính có cùng độ đo. Với N mẫu thì số lượng tối đa các thuộc tính có các chỉ số *Gini* khác nhau trong bài toán phân lớp nhị phân là (N(N 2)/4)-1 (Deng and Runger, 2013). Ví dụ ta có

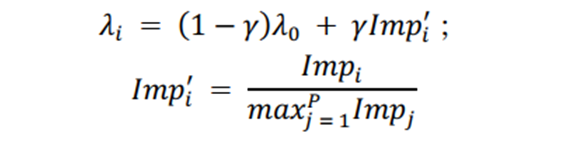
30 mẫu có số chiều là 3.000, như vậy có lớn nhất là 239 thuộc tính có độ đo khác nhau và 3.000-239 = 2.761 thuộc tính cùng độ đo. Chính vì vậy RRF phải chọn ngẫu nhiên một trong các thuộc tính đó để tách nút. Các thuộc tính này có thể là những thuộc tính không tốt (không hoặc ít có liên quan đến biến đích) dẫn đến khả năng dự đoán của rừng RRF không cao.

Xuất phát từ lý do trên, Deng *et al.* (2013) đã đề xuất phương pháp rừng ngẫu nhiều điều hòa có điều hướng (Guided Regularized Random Forests, GRRF) để khắc phục nhược điểm của RRF. Ở phương pháp GRRF các tác giả tính độ quan trọng thuộc tính dựa trên độ quan trọng thuộc tính được tạo ra bởi RF gốc trên toàn bộ tập dữ liệu ban đầu. Do vậy chỉ số *Gini* của các thuộc tính có độ quan trọng khác nhau sẽ có giá trị khác nhau. Khi đó với các bài toán có số mẫu nhỏ, số chiều lớn như dữ liệu gen, GRRF sẽ chọn được các thuộc tính chia nút tốt hơn và kết quả phân lớp cũng tốt hơn (Deng and Runger, 2013).

Nếu như RRF gán hệ số phạt như nhau cho tất cả các thuộc tính mới thì GRRF sử dụng những thuộc tính có độ quan trọng lớn từ RF truyền thống để “*hướng dẫn*” quá trình lựa chọn thuộc tính mới phân tách nút*.* Thuộc tính có độ quan trọng cao thì được gán giá trị *λ* cao, ngược lại thuộc tính có độ quan trọng thuộc tính thấp được gán giá trị *λ* thấp. Công thức tính độ quan trọng cho các thuộc tính mới tại nút *v* trong GRRF như sau:



Với λi∈ (0, 1] là hệ số phạt của Xi và λi được tính như sau:



Trong đó λ0∈ (0, 1] là hệ số điều khiển mức độ điều hướng (trong mô hình RRF). Còn hệ số γ ∈ [0, 1] điều khiển độ quan trọng của một thuộc tính (đã được chuẩn hóa). Khi γ = 0 thì GRRF chính là RRF. Một thuộc tính có độ quan trọng lớn sẽ bị phạt ít hơn. Để thay đổi kích thước tập con thuộc tính được chọn ta có thể thay đổi các giá trị của λ0 và γ và để giảm tham số cho mô hình GRRF các tác giả chọn λ0 = 1. Khi đó, ta có:

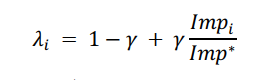


***2. Rừng ngẫu nhiên có điều hướng (Guided Random Forest, GRF)***

Tương tự như phương pháp lựa chọn thuộc tính GRRF, Deng *et al.* (2013) đã đề xuất phương pháp rừng ngẫu nhiên có điều hướng bằng cách sử dụng các độ đo độ quan trọng thuộc tính từ RF nguyên bản. Tuy nhiên, các cây trong GRRF được xây dựng một cách tuần tự, liên quan chặt chẽ và không cho phép tính toán song song, trong khi các cây trong GRF được xây dựng một cách độc lập và có thể được thực hiện song song. Phương pháp này cũng cho phép sử dụng các chỉ số đo độ quan trọng khác độ đo độ thuộc tính từ phương pháp rừng ngẫu nhiên gốc (các chỉ số có thể được cung cấp bởi chính người dùng thông qua chỉ số λi). Tư tưởng chính của GRF là tăng trọng số gain(Xi) dựa vào độ đo độ quan trọng thuộc tính từ RF nguyên bản



Trong đó, *gain(Xi)* là độ đo *Gini* của thuộc tính *Xi* để thực hiện tách nút và λi được tính như sau:



Với *Impi,Imp\** là độ đo thuộc tính và giá trị lớn nhất của độ đo thuộc tính từ phương pháp RF nguyên bản. *Imp/Imp\**∈ [0, 1] là hệ số chuẩn hóa độ quan trọng thuộc tính, ∈ [0, 1] là hệ số quan trọng. Ở phương pháp GRF, các thuộc tính có độ quan trọng nhỏ hơn sẽ bị phạt nhiều hơn và độ phạt tăng khi tăng (GRF trở thành RF khi = 0).

Theo các phân tích trên, chúng ta thấy sự khác biệt căn bản giữa GRF và GRRF là các thuộc tính được sử dụng để xây dựng các cây trước trong đó của rừng GRRF có thể tiếp tục được sử dụng (ảnh hưởng) để xây dựng cây hiện tại, nhưng ngược lại cách xây dựng cây của GRF những thuộc tính đã được sử dụng xây dựng cây trước sẽ không được sử dụng lại (không ảnh hưởng) để xây dựng cây hiện tại.

Các thuộc tính được sử dụng trong mô hình GRRF là có liên quan đến *biến đích/biến mục tiêu/biến phân loại* và không lựa chọn lặp lại (những gen có chức năng tương tự) trong khi các đặc trưng được sử dụng trong một mô hình GRF là có liên quan đến biến đích và có thể lựa chọn lặp lại (các gen có thể được chọn lại hoặc chứa các gen có chức năng tương tự).

***Kết luận chương II***

Chương 3

ỨNG DỤNG PHƯƠNG PHÁP RỪNG NGẪU NHIÊN

TRONG VIỆC CHẨN ĐOÁN BỆNH

**3.1. Mô tả bài toán và yêu cầu chẩn đoán Bệnh cúm**

Cúm là một bệnh truyền nhiễm cấp tính lây theo đường hô hấp, do các vi rút cúm A,B,C gây nên. Bệnh khởi phát đột ngột bằng sốt cao, nhức đầu, đau mỏi toàn thân và những dấu hiệu hô hấp, dễ dẫn đến viêm phổi, tỷ lệ tử vong cao.

Trong thế kỷ XX nhiều đại dịch cúm đã xảy ra với số mắc và tỷ lệ tử vong cao. Tuy nhiên lâm sàng bệnh đã được mô tả nhiều thế kỷ trước (A Hirsd, 1881-1886). Năm 1933 W.Smith, C.Andrews, P.Laidpow xác định được vi rút cúm A. Năm 1940 T.Francis và T.Magill phát hiện vi rút cúm B, năm 1949 R.Taylor phát hiện vi rút cúm C.

Bằng các kỹ thuật sinh học phân tử, các nhà khoa học đã xác định thủ phạm gây ra vụ đại dịch cúm đầu tiên năm 1918-1919 (cúm TâyBan Nha) là vi rút cúm A chủng H1N1 gây tử vong 20 triệu người, đại dịch cúm châu á năm 1957- 1958 là do cúm A chủng H2N2 làm khoảng 1 triệu người tử vong. Cúm Hồng Kông năm 1968- 1969 do cúm A H3N2, cúm Nga năm 1977 do chủng H1N1,…

Virút cúm A có khả năng thay đổi cấu trúc kháng nguyên. Quá trình lai ghép, tái tổ hợp giữa virut cúm A ở người Virut cúm A ở động vật sẽ tạo thành chủng virut cúm mới. Vì vậy virut cúm A là thủ phạm gây ra các đại dịch, virut cúm B thường gây các vụ dịch. Khu vực, virut cúm C thường gây các dịch tản phát. Cứ khoảng 10- 14 năm lại có một đại dịch cúm xảy ra. Dịch cúm A(H5 N1) lây từ gia cầm sang người xảy ra ở Hồng Kông đang có nguy cơ lan rộng thành đại dịch.

Khảo sát tại trạm y tế xã, phường, từ cơ sở dữ liệu ban đầu thống kê được từ 300 bệnh nhân, theo một số biểu hiện lâm sàng (thuộc tính điều kiện) về bệnh cúm. Tập huấn luyện sưu tập được gồm có 300 dòng (ứng với 300 bệnh nhân). Mỗi hàng (tương ứng với 1 bệnh nhân) gồm có 14 thuộc tính điều kiện và thuộc tính thứ 15 là thuộc tính quyết định -Cúm.

Chúng ta cần xây dựng cây quyết định để chẩn đoán bệnh nhân có mắc bệnh cúm hay không (thuộc tính quyết định là Cúm ={Có, Không}).

**3.2. Xây dựng mô hình và nội dung giải quyết bài toán**

Mục đích của bài toán chẩn đoán bệnh lâm sàng là dựa vào các triệu chứng bệnh nhân mắc phải mà đưa ra được kết luận bệnh nhân có mắc bệnh hay không và bệnh nhân có những triệu chứng nào thì kết luận được bệnh nhân mắc bệnh nên đối với bài toán chẩn đoán bệnh lâm sàng thì ta chủ yếu sử dụng mối quan hệ giữa cây quyết định và phụ thuộc hàm để xây dựng các luật kết hợp hay các luật chẩn đoán trong quá trình điều trị bệnh.

Trong thực tế ta thường dựa vào các triệu chứng để chẩn đoán bệnh và mỗi một bệnh nhân lại có các triệu chứng có thể là rất khác nhau vì vậy việc xây dựng cây quyết định theo một phương pháp lựa chọn thích hợp tỏ ra thật sự cần thiết và mang nhiều ý nghĩa trong thực tế đối với công tác chẩn đoán bệnh.

Bài toán chẩn đoán bệnh lâm sàng được thiết kế xử lý trong luận văn theo chiều xây dựng các cây quyết định bằng mô hình “rừng ngẫu nhiên” được thiết kế như sau: Từ CSDL với 13 thuộc tính điều kiện cho một bệnh nhân. Mỗi bệnh nhân lại có những giá trị khác nhau trong cùng một thuộc tính (Có, Không). Từ đó, chúng ta sẽ xây dựng được luật IF ... THEN ... dùng để chẩn đoán bệnh.

Nội dung giải quyết bài toán sẽ được thực hiện đúng như hướng dẫn trong mục 2.4.2 đã trình bày ở trên theo ba pha:

* Pha 1: Tạo ngẫu nhiên tập dữ liệu từ tập mẫu ban đầu
* Pha 2: Xây dựng các cây cơ sở
* Pha 3: Kết hợp các cây cơ sở theo phương thức bỏ phiếu (lấy số đông).

**3.3. Tập mẫu học đầu vào và các chức năng của chương trình thử nghiệm**

***3.3.1. Tập mẫu học đầu vào***

Tập mẫu học đầu vào gồm 15 cột, trong đó có 14 thuộc tính điều kiện và 1 thuộc tính quyết định (phân lớp) là “cúm“.

Thuộc tính Độ tuổi có 4 giá trị: Từ 1-16T, Từ 17-30T, Từ 31-50T, Trên 50T;

Thuộc tính Giới tính có 2 giá trị: Nam, Nữ;

Thuộc tính Đau đầu có 2 giá trị: không, có;

Thuộc tính Đau cơ có 2 giá trị: không, có;

Thuộc tính Thân nhiệt có 3 giá trị: bình thường, cao, rất cao;

Thuộc tính Ớn lạnh có 2 giá trị: không, có;

Thuộc tính Chóng mặt có 2 giá trị: không, có;

Thuộc tính Mệt mỏi có có 2 giá trị: không, có;

Thuộc tính Ho có 2 giá trị: không, có;

Thuộc tính Đau họng có 2 giá trị: không, có;

Thuộc tính Chảy nước mũi có 2 giá trị: không, có;

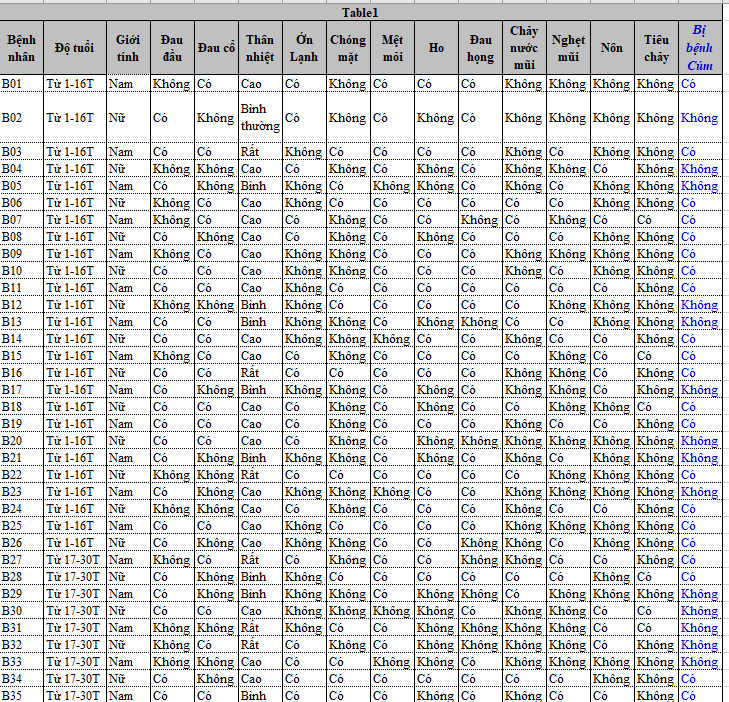
Thuộc tính Nghẹt mũi có 2 giá trị: không, có;

Thuộc tính Nôn có 2 giá trị: không, có;

Thuộc tính Tiêu chảy có 2 giá trị: không, có;

Thuộc tính quyết định (Cúm) có 2 giá trị: không, có.

*Bảng 1- Tập huấn luyện đầu vào*

**

***3.3.2. Các chức năng của chương trình thử nghiệm***

Chương trình thử nghiệm cần có các chức năng sau:

1-Nhập dữ liệu đầu vào (từ file Excel)

2-Tạo các vectơ ngẫu nhiên (véc tơ đặc trưng)

3-Xây dựng các Bootstrap theo các véc tơ đặc trưng

4-Xây dựng rừng ngẫu nhiên tương ứng với các Bootstrap vừa mới xây dựng

5-Tạo cây quyết định dự đoán bằng phương pháp bỏ phiếu

6-Thử nghiệm chẩn đoán bệnh với bệnh nhân mới

**3.4. Quy trình ứng dụng giải thuật “rừng ngẫu nhiên“ cho bài toán chẩn đoán bệnh cúm**

Quy trình ứng dụng giải thuật “rừng ngẫu nhiên“ cho bài toán chẩn đoán bệnh cúm được thực hiện theo các bước sau:

*+Bước 1- Chuyển dữ liệu về dạng số và lưu thành tệp trong Excel*

Bài toán chẩn đoán bệnh cúm trước hết cần được chuyển về dữ liệu dạng số: các thuộc tính có giá trị: 0- không, 1- có;

Riêng thuộc tính Thân nhiệt có 3 giá trị: bình thường, cao, rất cao. Để đơn giản, chúng ta chuyển về dạng có 2 giá trị: 0- bình thường, 1- cao hoặc rất cao

Kết quả, chúng ta có Bảng Input như dưới đây.

*Bảng Input-Tập dữ liệu huấn luyện*

| **table1** | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Benhnhan** | **Daudau** | **Dauco** | **Thannhiet** | **Onlanh** | **Chongmat** | **Metmoi** | **Ho** | **Dauhong** | **Chaynuocmui** | **Nghetmui** | **Non** | **Tieuchay** | **Cum** |
| B01 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| B02 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| B03 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| B04 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| B05 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| B06 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| B07 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| B08 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| B09 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| B10 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| B11 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| B12 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| B13 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| B14 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| B15 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| B16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| B17 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| B18 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| B19 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| B20 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B21 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| B22 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| B23 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| B24 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| B25 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| B26 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| B27 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| B28 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| B29 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| B30 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| B31 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| B32 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| B33 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| B34 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| B35 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| B36 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| B37 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| B38 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| B39 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| B40 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| B41 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| B42 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| B43 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| B44 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| B45 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| B46 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| B47 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| B48 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| B49 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| B50 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

*+Bước 2- Thực hiện các pha trong giải thuật rừng ngẫu nhiên*

*Một số giao diện chính:*

-Giao diện chủ của chương trình:

-Giao diện nhập dữ liệu huấn luyện:

-Giao diện tạo véc tơ đặc trưng ngẫu nhiên:

-Giao diện tập dữ liệu Bootstrap:

-Giao diện cây quyết định con trong rừng:

-Giao diện cây quyết định tổng thể được chọn theo phương pháp bỏ phiếu:

-Giao diện chẩn đoán bệnh nhân mới:

**3.6. Đánh giá kết quả thử nghiệm**

-Về kết quả thu được:

-Về so sánh với một số phương pháp khác:

-Tồn tại:

-Hướng phát triển:

*Kết luận chương III*