Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XIII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Nha Trang, ngày 8-9/10/2020
DOI: 10.15625/vap.2020.00199

PHÂN LỚP DỮ LIỆU DỰA VÀO PHƯƠNG PHÁP LỰA CHỌN ĐẶC TRƯNG SỬ DỤNG PHỤ THUỘC HÀM XẤP XỈ

Phan Anh Phong, Lê Văn Thành, Nguyễn Hải Yến
Viện Kỹ thuật và Công nghệ, Trường Đại học Vinh
phongpa@gmail.com, thanh.cntt.dhv@gmail.com, nguyenhaiyen1632@gmail.com

TÓM TẮT: *Lựa chọn đặc trưng là kỹ thuật chọn ra một tập con các đặc trưng phù hợp, liên quan từ tập dữ liệu gốc bằng cách loại bỏ các đặc trưng nhiễu, dư thừa không liên quan nhằm tăng hiệu năng cũng như giảm chi phí trong quá trình khai phá dữ liệu hay học máy. Bài báo này nghiên cứu phương pháp phân lớp dữ liệu dựa vào kỹ thuật lựa chọn đặc trưng với phụ thuộc hàm xấp xỉ và độ đo lỗi g_3. Một số thử nghiệm phân lớp trên các tập dữ liệu thực tế cho thấy sự phù hợp của hướng nghiên cứu.*

Từ khóa: *Phân lớp dữ liệu, lựa chọn đặc trưng, phụ thuộc hàm xấp xỉ.*

I. GIỚI THIỆU

Lựa chọn đặc trưng là một trong những vấn đề quan trọng trong lĩnh vực khai phá dữ liệu và học máy. Mục đích chính của lựa chọn đặc trưng là tìm ra các đặc trưng hữu ích để mô hình hóa hệ thống, theo đó làm tăng hiệu năng và giảm thời gian thực hiện cho hệ thống. Về bản chất lựa chọn đặc trưng là quá trình tính toán mức độ quan trọng của từng đặc trưng hoặc từng nhóm đặc trưng và sau đó chọn tập con hữu ích nhất trong không gian đặc trưng đó để xây dựng hệ thống [2, 5, 6].

Phân lớp dữ liệu là một bài toán tiêu biểu của khai phá dữ liệu, phần lớn dữ liệu trong bài toán phân lớp khi thu thập được đều có số đặc trưng (thuộc tính) rất nhiều, có thể lên tới hàng chục, hàng trăm, thậm chí là hàng nghìn đặc trưng, chẳng hạn như các bộ dữ liệu trong y tế, trong sinh học,.... Ngoài ra, các đặc trưng này có thể có những đặc trưng dư thừa và ít hữu ích cho khai phá dữ liệu và học máy. Việc xây dựng mô hình phân lớp từ dữ liệu như vậy có thể dẫn đến hiệu năng phân lớp thấp cả về tốc độ và khả năng dự báo vì trong dữ liệu có những đặc trưng nhiễu, đặc trưng ít liên quan. Vì vậy, đã có nhiều nghiên cứu cố gắng giải quyết vấn đề này bằng cách sử dụng các kỹ thuật lựa chọn đặc trưng. Hiện nay có 3 cách tiếp cận chính để lựa chọn đặc trưng [6, 7]:

- Mô hình lọc (Filter). Đây là cách lựa chọn đặc trưng sử dụng tính toán trọng số (độ quan trọng), có thể là mối quan hệ giữa các thuộc tính và lớp, sau đó chọn các đặc trưng có trọng số cao hơn một ngưỡng cụ thể. Các thuật toán trong danh mục này bao gồm CfsSubsetEval, độ lợi thông tin và Chi-Square,....

- Mô hình đóng gói (Wrapper). Mô hình này tìm kiếm tập con các đặc trưng tốt bằng cách đánh giá chất lượng của các tập đặc trưng. Việc đánh giá chất lượng thường sử dụng độ chính xác phân lớp của các thuật toán học.

- Mô hình nhúng (Embedded). Mô hình nhúng là sự kết hợp ưu điểm của mô hình lọc và đóng gói bằng cách sử dụng đồng thời tiêu chí đánh giá độc lập và các thuật toán học để đánh giá tập con các đặc trưng, theo đó giúp cải tiến hiệu năng phân lớp.

Khái niệm phụ thuộc hàm được đưa ra bởi Codd có vai trò quan trọng trong lý thuyết cơ sở dữ liệu quan hệ. Các phụ thuộc hàm rất hữu ích trong việc phân tích và thiết kế cơ sở dữ liệu quan hệ. Tuy nhiên, trong thực tế, do có một số giá trị dữ liệu không chính xác hoặc một số ngoại lệ nào đó làm cho các phụ thuộc hàm không thỏa. Sự phụ thuộc tuyệt đối này dường như quá nghiêm ngặt khi ta hình dung tới một quan hệ có hàng nghìn bộ, trong khi đó, chỉ có khoảng vài chục bộ vi phạm phụ thuộc hàm. Điều này làm mất tính chất phụ thuộc vốn có giữa các thuộc tính (đặc trưng) của dữ liệu. Vì vậy, có nhiều nghiên cứu đã mở rộng khái niệm phụ thuộc hàm thành phụ thuộc hàm xấp xỉ, các phụ thuộc này cho phép có một số lượng lỗi nhất định của các bộ dữ liệu đối với phụ thuộc hàm. Các phụ thuộc hàm xấp xỉ không những giúp ta thấy được mối quan hệ tiềm ẩn giữa các thuộc tính mà còn giúp ta thuận tiện hơn trong việc phân tích dữ liệu và đánh giá thông tin [1, 4].

Xuất phát từ cách nhìn đó, bài báo này đề xuất một phương pháp phân lớp dữ liệu dựa vào phụ thuộc hàm xấp xỉ và độ đo lỗi g_3. Cách tiếp cận sử dụng phụ thuộc hàm và phụ thuộc hàm xấp xỉ để lựa chọn đặc trưng đã được Uncu và Turken nghiên cứu [3]. Tuy nhiên, trong nghiên cứu đó, các tác giả đã đề xuất một thuật toán dựa vào phụ thuộc hàm truyền thống và thuật toán K-láng giềng gần nhất để xác định các biến vào quan trọng cho các hệ thống với thuộc tính vào có miền giá trị liên tục. Ngoài ra để cho thuật toán lựa chọn đặc trưng có khả năng đáp ứng với nhiễu, các phụ thuộc hàm xấp xỉ được sử dụng kết hợp với giá trị độ thuộc để đối phó với tính không chắc chắn trong dữ liệu. Điểm khác biệt của bài báo này với nghiên cứu của Uncu và Turken là sử dụng phụ thuộc hàm xấp xỉ với độ đo lỗi g_3 để lựa chọn đặc trưng cho các bộ phân lớp. Hơn nữa các thử nghiệm được thực hiện trên các bộ dữ liệu thực tế của UCI có độ tin cậy cao trong đánh giá hiệu năng của các phương pháp phân lớp [10].

Bài báo được tổ chức như sau, tiếp theo phần mở đầu, phần II trình bày ngắn gọn các kiến thức liên quan về phụ thuộc hàm, phụ thuộc hàm xấp xỉ và thuật toán khai phá phụ thuộc hàm xấp xỉ mức 1. Phương pháp đề nghị được mô

tả chi tiết trong phần III. Các kết quả thử nghiệm của phương pháp được trình bày trong phần IV và cuối cùng là kết luận và hướng phát triển.

II. KIẾN THỨC CHUẨN BỊ

A. Phụ thuộc hàm

Cho một tập hữu hạn các thuộc tính U = (A1, A2, ..., An), r là một quan hệ trên U, X và Y là hai tập con của U. Khi đó X → Y (đọc là X xác định Y hoặc Y phụ thuộc hàm vào X) nếu với mọi bộ t1, t2 ∈ r mà t1[X] = t2[X] thì t1[Y] = t2[Y]. Hay nói khác đi, với mỗi quan hệ r trên U, thuộc tính Y phụ thuộc hàm vào thuộc tính X nếu với mỗi giá trị của thuộc tính X xác định một giá trị duy nhất của thuộc tính Y. Như vậy, phụ thuộc hàm cho thấy mối tương quan giữa các thuộc tính của một quan hệ. Bài toán phát hiện các phụ thuộc hàm từ dữ liệu đã nhận được rất nhiều nghiên cứu trong thiết kế cơ sở dữ liệu quan hệ và trong khai phá tri thức.

Để khai phá các phụ thuộc hàm, người ta thường sử dụng phương pháp phân lớp tương đương, tức là chia bản ghi của các quan hệ thành các nhóm dựa trên các giá trị khác nhau cho mỗi thuộc tính. Đối với mỗi thuộc tính, số lượng các nhóm đúng bằng với số các giá trị khác nhau cho các thuộc tính đó. Mỗi nhóm được gọi là một lớp tương đương.

Một trong những thuật toán nổi tiếng về khai phá phụ thuộc hàm là thuật toán TANE, chi tiết hơn về thuật toán này có thể tìm hiểu trong [9]. Trong [4] đã sử dụng sử dụng SQL để kiểm tra đối với các phụ thuộc hàm X → Y. Để dễ hiểu hơn, ví dụ sau đây minh họa việc tìm phụ thuộc hàm.

Ví dụ 2.1. Cho một quan hệ r(R) chứa 5 thuộc tính R = {A, B, C, D, E} được thể hiện trong bảng sau:

Bảng 1. Quan hệ r(A, B, C, D, E)

RawID	**A**	**B**	**C**	**D**	**E**
1	1	1	1	1	1
2	1	2	2	2	1
3	2	2	1	3	2
4	2	2	1	1	2
5	2	3	2	4	2
6	3	3	1	5	3
7	3	4	2	1	3
8	3	4	3	6	3

Chúng ta sẽ xem xét và đánh giá xem phụ thuộc hàm ABCE → D có đúng hay không. Áp dụng thuật toán trong [4] ta dễ xác định được ABCE → D có đúng trên r hay không. Mã lệnh viết bằng SQL như sau:

```
SELECT STR(A,1) + STR(B,1) + STR(C,1) + STR(E,1) AS 'ABCE', COUNT(D) AS 'CountD'
FROM r
GROUP BY A, B, C, E;
```

Khi thực hiện mã lệnh SQL trên kết quả thu được như ở bảng dưới đây:

Bảng 2. Các bản ghi CountD riêng biệt được nhóm theo ABCE

ABCE	**CountD**
1111	1
1221	1
2212	**2**
2322	1
3313	1
3423	1
3433	1

Ta thấy rằng, nếu phụ thuộc hàm ABCE → D là đúng thì tất cả các giá trị CountD đều bằng 1 trong mỗi phân hoạch. Nếu CountD lớn hơn 1 chứng tỏ sẽ có nhiều hơn một bản ghi có giá trị khác nhau phân hoạch tương ứng, điều này trái với định nghĩa phụ thuộc hàm. Trong Bảng 3 cho thấy kết quả khi kiểm tra ABCE → D là CountD = 2 với ABCE = (2, 2, 1, 2) bởi vì tổ hợp ABCE = (2, 2, 1, 2) xác định 2 giá trị D khác nhau.

Bảng 3. Các bản ghi trong *r* vi phạm phụ thuộc hàm ABCE → D

A	**B**	**C**	**E**	**D**
2	2	1	2	3
2	2	1	2	1

B. Phụ thuộc hàm xấp xỉ

Gần đây nhiều nghiên cứu đã tập trung tìm ra mối liên hệ giữa các thuộc tính của đối tượng, trong đó có phụ thuộc hàm xấp xỉ. Phụ thuộc hàm xấp xỉ là một sự mở rộng của khái niệm phụ thuộc hàm truyền thống, dùng để biểu diễn mức độ phụ thuộc ở mức thuộc tính của các đối tượng. Về hình thức, phụ thuộc hàm xấp xỉ có dạng (X ⇝ Y), trong đó X và Y là các thuộc tính. Một ví dụ quen thuộc về phụ thuộc hàm xấp xỉ là Nationality ⇝ Language, ngôn ngữ của một người phụ thuộc hàm xấp xỉ vào quốc tịch của người đó. Như chúng ta biết Nationality ⇝ Language không phải luôn đúng với tất cả mọi người nhưng có thể đúng với một mức độ xấp xỉ nào đó, hay nói khác đi là với một mức độ lỗi nào đó. Để hiểu hơn về phụ thuộc hàm xấp xỉ ta xét ví dụ sau đây.

Ví dụ 2.2. Cho quan hệ r (A, B, Class) có 3 thuộc tính A, B và Class như Bảng 4:

Bảng 4. Quan hệ r (A1, A2, Class)

RawID	**A1**	**A2**	**Class**
1	1	2	A
2	1	2	A
3	2	0	A
4	2	0	A
5	2	0	B
6	3	1	B
7	1	1	C
8	3	1	C

Ta dễ thấy phụ thuộc hàm A1 → Class là không đúng, tuy nhiên, nếu ta loại các dòng 5, dòng 6 và dòng 7 (hoặc loại dòng 5, dòng 7 và 8) thì A1 → Class đúng, khi đó thuộc tính Class được gọi là phụ thuộc hàm xấp xỉ vào A1 hay nói cách khác A1 xác định hàm xấp xỉ Class với một độ lỗi nào đó.

Khi nghiên cứu phát hiện các phụ thuộc hàm xấp xỉ thì vấn đề xác định độ đo lỗi cho các phụ thuộc hàm loại này đóng vai trò quan trọng. Đã có nhiều tác giả đưa ra các độ đo lỗi dựa vào nhiều cách khác nhau, chi tiết hơn có thể tham khảo trong [1, 8]. Trong khuôn khổ bài báo này, chúng tôi giới thiệu độ đo lỗi g3 như sau.

Với một phụ thuộc hàm xấp xỉ khi có một bản ghi làm cho phụ thuộc hàm đó không đúng, người ta gọi bản ghi này là một trường hợp ngoại lệ. Các độ đo của phụ thuộc hàm xấp xỉ dựa trên số lượng các trường hợp ngoại lệ.

Một cặp (u, v) trong r là vi phạm phụ thuộc hàm X → Y nếu u[X] = v[X] nhưng u[Y] ≠ v[Y]. Cặp bản ghi (u, v) vi phạm phụ thuộc hàm còn được gọi là cặp ngoại lệ. Như vậy, một phụ thuộc hàm đúng trên r nếu không có cặp (u, v) nào trong r là ngoại lệ. Một bản ghi u thuộc r được gọi là vi phạm phụ thuộc hàm (bản ghi u là ngoại lệ) nếu nó là một thành phần trong cặp các bản ghi ngoại lệ.

Ký hiệu g_3 là số các bản ghi ngoại lệ ít nhất trong *r* mà khi loại bỏ chúng ra khỏi *r* thì phụ thuộc hàm trên đúng với quan hệ *r*. Khi đó, g_3 có thể được hình thức hóa như sau:

$$G_3(X \rightarrow Y, r) = |r| - \max\{|s| : s \subseteq r, s \models X \rightarrow Y\} \quad (2.1)$$

Trong công thức 2.1, |r| và |s| tương ứng là số các bản ghi trong r và s.

Độ đo lỗi g_3: Cho quan hệ r(U) và (X→Y) là một phụ thuộc hàm trên U. Gọi s ⊆ r là quan hệ sao cho có số bộ ít nhất cần phải loại bỏ khỏi r để r - s thỏa mãn phụ thuộc hàm (X→Y). Khi đó tỷ số của |s| và |r| được gọi là độ đo lỗi của phụ thuộc hàm (X→ Y) trên r, ký hiệu g_3(X→ Y, r). Như vậy:

$$g_3(X \rightarrow Y, r) = G_3(X \rightarrow Y)/|r| \quad (2.2)$$

Dễ dàng chứng minh được g_3 nằm trong đoạn [0, 1]. Giá trị g_3 càng nhỏ thì phụ thuộc hàm xấp xỉ đó càng có ý nghĩa trong mối quan hệ phụ thuộc của X và Y.

Có nhiều thuật toán để tìm phụ thuộc hàm xấp xỉ (PTHXX), việc khai phá tất cả tập phụ thuộc hàm xấp xỉ là phức tạp, chi tiết hơn có thể tham khảo trong [4, 9]. Để đơn giản và phù hợp với mục đích nghiên cứu của bài báo này, ở đây chúng tôi đề xuất cách tìm phụ thuộc hàm xấp xỉ mức 1 xác định thuộc tính phân lớp. Cụ thể hơn, đó là các phụ thuộc hàm xấp xỉ mà vế phải chỉ có thuộc tính phân lớp, còn vế trái là một thuộc tính mô tả trong bộ dữ liệu, theo lỗi g_3. Thuật toán 1 sau đây mô tả các bước để tìm tập phụ thuộc hàm xấp xỉ mức 1 xác định thuộc tính phân lớp với lỗi g_3.

Thuật toán 1. Xác định tập phụ thuộc hàm xấp xỉ mức *1* xác định thuộc tính phân lớp theo lỗi g_3

Đầu vào: Bộ dữ liệu *D* có *M*+1 thuộc tính, *r*(Ai, Class), Ai với i = 1, …, *M* là thuộc tính mô tả dữ liệu, Class là thuộc tính phân lớp, *N* số mẫu dữ liệu.

Đầu ra: Tập phụ thuộc hàm xấp xỉ mức *1*: PTHXX = {Ai ⇝ Class, g3}

Phương pháp:

Với mỗi thuộc tính Ai trong *r* thực hiện các công việc sau:

Bước 1.

1.1. Khởi tạo: *temp1* = (Ai_1, $Class_1$, $value_1$=1)

1.2. **For** *n* = 2 **to** *N* **do**:

if cặp (Ai_n, $Class_n$) đã có trong *temp1*: tăng $value_1$ lên 1

else: thêm (Ai_n, $Class_n$, 1) vào *temp1*

Bước 2.

2.1. Khởi tạo temp2 = (Ai_1, $value_1$), *K* là số hàng trong *temp1*

2.2. **For** *k* = 1 **to** *K* **do**:

if Ai_k chưa có trong *temp2*: thêm (Ai_k, $value_k$) vào *temp2*

else if $value_k$ > value của Ai trong *temp2*: value của Ai trong *temp2* = $value_k$

Bước 3: SoBanGhiHopLe = tổng value trong *temp2*

Bước 4. Tính g_3 cho (Ai ⇝ Class) theo công thức g_3 = (1- SoBanGhiHopLe/*N*)

Bước 5. PTHXX[i] ← (Ai, Class, g3)

Độ phức tạp thuật toán 1: *O(M*N)* với *M* là số thuộc tính mô tả của bộ dữ liệu *D*, *N* là số mẫu dữ liệu trong *D*.

III. PHƯƠNG PHÁP ĐỀ NGHỊ

Phần này trình bày chi tiết phương pháp phân lớp dữ liệu dựa vào kỹ thuật lựa chọn thuộc tính sử dụng phụ thuộc hàm xấp xỉ với độ đo lỗi g_3. Ý tưởng của phương pháp là dựa vào thuật toán tìm phụ thuộc hàm xấp xỉ và lỗi g_3 tương ứng để xác định mối liên hệ giữa thuộc tính mô tả với thuộc tính phân lớp trong tập dữ liệu. Tùy theo mức độ lỗi g_3 mà ta chọn các thuộc tính liên quan để mô hình hóa bộ phân lớp. Độ dự báo chính xác Accuray của các bộ phân lớp ứng với một tập con các thuộc tính được lưu lại để người dùng có cái nhìn tổng quan. Sau đó, tập con thuộc tính có hiệu năng Accuracy cao nhất sẽ được chọn.

Các phụ thuộc hàm xấp xỉ với lỗi g_3 nhỏ có thể xem là những mối liên hệ có giá trị tin cậy cao, vì các phụ thuộc hàm xấp xỉ này là các mẫu có sự xuất hiện thường xuyên từ tập dữ liệu huấn luyện và chúng được cho là chứa các đặc trưng quan trọng đối với thuộc tính phân lớp. Phương pháp đề xuất được hình thức hóa bởi thuật toán 2 sau đây.

Thuật toán 2. Phân lớp dữ liệu dựa vào phương pháp lựa chọn đặc trưng sử dụng phụ thuộc hàm xấp xỉ

Đầu vào:

- Bộ dữ liệu *Dataset* (*Ai, Class*), với *i* = 1, 2, …, *M*. Bộ dữ liệu có *M* +1 đặc trưng, trong đó *Class* là thuộc tính phân lớp, *M* đặc trưng còn lại là thuộc tính mô tả đối tượng.
- *K* = 10, dùng để chia bộ dữ liệu theo K-folds

Đầu ra:

Tập các đặc trưng được chọn A_k tương ứng với độ dự báo phân lớp *Accuracy* cao nhất

Phương pháp:

Bước 1. Đọc bộ dữ liệu *FS_Dataset* = *Dataset* (*Ai, Class*), với *i* = 1, 2, …, *M*

Bước 2. Đánh giá hiệu năng *Accuracy* của các bộ phân lớp với tất cả đặc trưng theo công thức 3.1:

2.1. Áp dụng thuật toán Cây quyết định hoặc Naïve Bayes để phân lớp dữ liệu *FS_Dataset* với *k*-folds

2.2. Lưu các đặc trưng *Ai* từ *FS_Dataset* và hiệu năng *Accuracy* vào *FS_Accuracy*

Bước 3. Tìm các phụ thuộc hàm xấp xỉ mức 1 xác định thuộc tính phân lớp *Class* và lỗi g_3 tương ứng:

3.1. Với mỗi đặc trưng *Aj* trong *FS_dataset*, với *j* = 1, 2, …, *M* tìm phụ thuộc hàm xấp xỉ dạng *Aj* ⇝ *Class* và lỗi g_3 theo Thuật toán1

3.2. Lưu phụ thuộc hàm xấp xỉ đã tính ở *Bước 3.1* vào tập *PTHXX*

Bước 4. Trong khi số thuộc tính của *FS_Dataset* > 1 thì:

4.1. Tìm độ lỗi g_3 lớn nhất trong tập *PTHXX* có dạng *Aj* ⇝ *Class*, $Mxg_3 = Max(g_3)$

4.2. Loại các phụ thuộc hàm xấp xỉ từ *PTHXX* có $g_3 = Mxg_3$

4.3. Loại đặc trưng *Aj* trong *FS_Dataset*

4.4. Áp dụng thuật toán Cây quyết định hoặc Naïve Bayes để phân lớp dữ liệu *FS_Dataset* với *k*-folds

4.5. Lưu các thuộc tính còn lại trong *FS_Dataset* và độ dự báo chính xác *Accuray* vào *FS_Accuracy*

Bước 5. Đưa ra tập thuộc tính A_k và hiệu năng *Accuracy* lớn nhất tương ứng trong *FS_Accuracy*

IV. THỬ NGHIỆM

A. Phương pháp thử nghiệm

Để đánh giá tính hiệu quả của phương pháp đề xuất, bài báo sử dụng 2 thuật toán phân lớp, theo đó hiệu năng của các thuật toán phân lớp cho thấy ý nghĩa của phương pháp lựa chọn đặc trưng.

Hiệu năng của các bộ phân lớp sử dụng lựa chọn đặc trưng theo phụ thuộc hàm xấp xỉ được so sánh với các các bộ phân lớp tương ứng sử dụng thuật toán lựa chọn chọn đặc CfsSubsetEval và độ lợi thông tin (Information Gain). Độ đo hiệu năng cho các phương pháp là số lượng các đặc trưng được chọn càng thấp càng tốt và độ chính xác của thuật toán phân lớp càng cao càng tốt. Các thử nghiệm này đã được thực hiện trên máy tính với cấu hình core i5-8250U, 1.80 GHz, 8 GB RAM và ngôn ngữ lập trình python.

Phương pháp đề nghị được thử nghiệm với một số bộ dữ liệu tiêu biểu trong phân lớp dữ liệu từ kho UCI [10]. Chi tiết về các bộ dữ liệu này được mô tả trong Bảng 5.

Bảng 5. Chi tiết về các bộ dữ liệu thử nghiệm

Bộ dữ liệu	Số bản ghi	Số đặc trưng
Ecoli	336	9
Diabetes	768	9
Heart	303	14
Zoo	101	18

Có nhiều phương pháp để đánh giá hiệu năng của các bộ phân lớp, trong bài báo này, việc đánh giá hiệu quả phân lớp của thuật toán Naïve Bayes và Cây quyết định đối với các bộ dữ liệu được cho theo kỹ thuật k-folds. Tập mẫu ban đầu được phân chia ngẫu nhiên tới k tập con. Với k tập mẫu con này, một mẫu đơn được dùng như dữ liệu đánh giá cho việc kiểm tra mô hình và k-1 tập mẫu còn lại được sử dụng như dữ liệu huấn luyện. Quá trình đánh giá chéo được lặp lại k lần. Lấy trung bình cộng của k kết quả dự báo thu được theo công thức 3.1 ta có đánh giá hiệu năng cho bộ phân lớp. Các kết quả thử nghiệm trong mục này sử dụng tham số k =10.

$$accuracy = \frac{số\ bản\ ghi\ phân\ lớp\ đúng}{Tổng\ số\ bản\ ghi\ được\ kiểm\ tra} \quad (3.1)$$

B. Kết quả thử nghiệm

Khi thực hiện phương pháp đề nghị, các ngưỡng lỗi g_3 cho các bộ dữ liệu được xác định như ở Bảng 6 sau đây.

Bảng 6. Ngưỡng lỗi g_3 khi lựa chọn thuộc tính cho các bộ dữ liệu

Bộ dữ liệu	Ngưỡng lỗi g_3	Số đặc trưng được chọn
Ecoli	0,43	4
Diabetes	0,30	4
Heart	0,24	4
Zoo	0,50	9

Bảng 7 và Bảng 8 thể hiện tính chính xác của kết quả dự báo của các thuật toán phân lớp trên các bộ dữ liệu theo các phương pháp lựa chọn đặc trưng khác nhau. Ngoài ra, trong các bảng này các số để trong dấu ngoặc tròn (.) ở mỗi dòng là số thuộc tính của mỗi bộ dữ liệu sau khi áp dụng các phương pháp lựa chọn đặc trưng cho kết quả dự báo tốt nhất tương ứng.

Bảng 7. Hiệu năng của thuật toán phân lớp Cây quyết định với các phương pháp lựa chọn đặc trưng

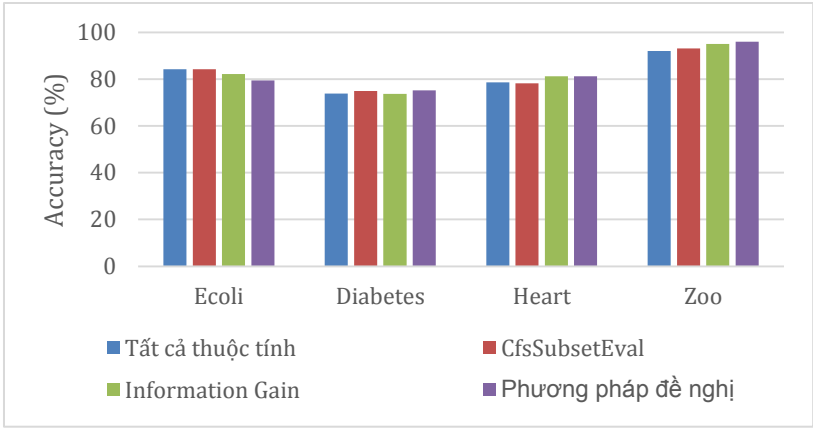
Bộ dữ liệu	Tất cả thuộc tính	CfsSubsetEval	Information Gain	Phương pháp đề nghị
Ecoli	**84,2262 %**	**84,2262% (6)**	82,1429 % (4)	79,4643 % (4)
Diabetes	73,83%	74,86% (4)	73,70% (5)	**75,13% (4)**
Heart	78,5479 %	78,2178% (7)	**81,1881% (5)**	**81,1881 % (5)**
Zoo	92,08%	93,07% (9)	95,05% (13)	**96,04% (9)**

Với thuật toán Cây quyết định và phương pháp lựa chọn đặc trưng theo phụ thuộc hàm xấp xỉ đều cho kết quả dự báo tốt hơn phương pháp CfsSubsetEval và độ lợi thông tin trên các bộ dữ liệu Diabetes, Heart và Zoo. Với thuật toán phân lớp Naïve Bayes phương pháp lựa chọn đặc trưng theo phụ thuộc hàm xấp xỉ cũng cho kết quả dự báo tốt nhất trên bộ dữ liệu Zoo, khá tốt với bộ dữ liệu Diabetes. Trong khi đó, với 2 bộ dữ liệu Heart và Ecoli đều cho kết quả chưa được như mong muốn. Tuy nhiên, với bộ phân lớp Naïve Bayes với phương pháp lựa chọn đặc trưng sử dụng phụ thuộc hàm xấp xỉ khả quan hơn phương pháp lựa chọn đặc trưng theo độ lợi thông tin. Chi tiết hơn về kết quả dự báo và số đặc trưng được chọn có thể xem ở Bảng 7 và Bảng 8. Một thể hiện trực quan về hiệu năng của các phương pháp phân lớp trên các bộ dữ liệu được trình bày trong Hình 1 và Hình 2.

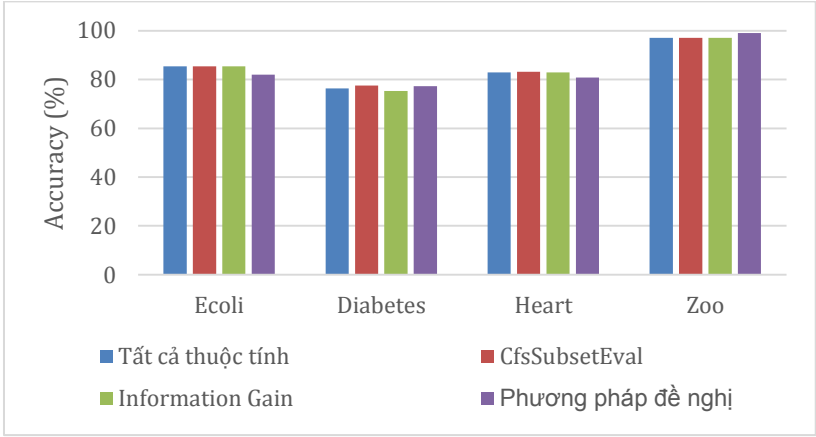
Bảng 8. Hiệu năng của thuật toán phân lớp Naïve Bayes với các phương pháp lựa chọn đặc trưng

Bộ dữ liệu	Tất cả thuộc tính	CfsSubsetEval	Information Gain	Phương pháp đề nghị
Ecoli	**85,4167 %**	**85,4167 % (6)**	**85,4167 % (4)**	81,9429 % (4)
Diabetes	76,3021%	**77,48% (4)**	75,2604 % (5)	**77,2078 % (5)**
Heart	82,8383 %	**83,1683 % (7)**	82,8383% (6)	80,7634 % (5)
Zoo	97,0297 %	97,0297 % (9)	97,0297 % (9)	**99,00 % (9)**

Qua thử nghiệm cho thấy phương pháp phân lớp dựa trên lựa chọn thuộc tính sử dụng phụ thuộc hàm xấp xỉ khá phù hợp với các bộ dữ liệu có nhiều thuộc tính rời rạc. Đặc điểm của bộ dữ liệu Ecoli có nhiều thuộc tính liên tục nên phương pháp đề nghị cho hiệu năng phân lớp kém hơn hai phương pháp lựa chọn đặc trưng kia. Điểm yếu này là dễ hiểu vì khái niệm phụ thuộc hàm xấp xỉ đang được định nghĩa dựa trên sự so sánh bằng nhau tuyệt đối giữa các giá trị.



Hình 1. Kết quả về tính chính xác của thuật toán phân lớp cây quyết định với các kỹ thuật lựa chọn đặc trưng



Hình 2. Kết quả về tính chính xác của thuật toán phân lớp Naïve Bayes với các kỹ thuật lựa chọn đặc trưng

V. KẾT LUẬN

Lựa chọn đặc trưng dựa vào phụ thuộc hàm xấp xỉ là một cách tiếp cận rất tự nhiên để giữ lại những đặc trưng liên quan, theo đó loại bỏ các đặc trưng dư thừa và ít liên quan trong dữ liệu. Kết quả của việc lựa chọn đặc trưng theo phương pháp này là một tập con các đặc trưng từ tập đặc trưng ban đầu nhưng vẫn đảm bảo các tính chất của dữ liệu gốc. Qua kết quả thử nghiệm với 4 bộ dữ liệu tiêu biểu từ UCI đã chứng tỏ được khả năng ứng dụng của phương phương pháp đề nghị với các bộ dữ liệu có nhiều thuộc tính rời rạc. Với các bộ dữ liệu có nhiều thuộc tính liên tục phương pháp đề xuất có hiệu năng phân lớp chưa tốt, từ đây đặt ra vấn đề cần giải quyết tiếp theo, đó là nghiên cứu phụ thuộc hàm xấp xỉ mở rộng với các độ đo lỗi khác và thử nghiệm với các bộ dữ liệu có số chiều cao.

TÀI LIỆU THAM KHẢO

[1] J, Atoum, "Approximate Functional Dependencies Mining Using Association Rules Specificity Interestingness Measure", British Journal of Mathematics & Computer Science, 15 (5), 1-10, 2016.

[2] J, Li, K, Cheng, S, Wang, F, Morstatter, R, P, Trevino, J, Tang, & H, Liu, "Feature Selection: A Data Perspective", ACM Computing Surveys, Vol. 50, No. 6, pp. 1-73, 2016,

[3] O, Uncu, I,B, Turksen, "Two step feature selection: approximate functional dependency approach using membership values", In Proceeding(s) of FUZZ-IEEE Conference, pp. 1643 - 1648, 2004.

[4] V, Matos, B, Grasser, "SQL-based Discovery of Exact and Approximate Functional Dependencies", SIGCSE Bulletin, Vol. 36, No. 4, pp. 58-63, 2004.

[5] I, Feddaoui, F, Felhi, J, Akaichi, "EXTRACT: new extraction algorithm of association rules from frequent itemsets", In Proceeding(s) of the IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp. 752-756, 2016.

[6] O, Villacampa, "Feature Selection and Classification Methods for Decision Making: A Comparative Analysis", PhD, Thesis, Nova Southeastern University, 2015.

[7] I, Guyon, and A, Elisseeff, "An Introduction to Feature Extraction, Feature Extraction", Foundations and Applications, 207(10), 740, 2006.

[8] C, Giannella, E, Robertson, "On an Information Theoretic Approximation Measure for Functional Dependencies", Computer Science Department, Indiana University, Bloomington, 2000.

[9] J, Atoum, "Mining approximate FDs from databases based on minimal cover and equivalent classes", European Journal of Scientific Research, 2009.

[10] https://archive,ics,uci,edu/ml/index,php.

DATA CLASSIFICATION BASED ON FEATURE SELECTION WITH THE APPROXIMATE FUNCTIONAL DEPENDENCE

Phan Anh Phong, Le Van Thanh, Nguyen Hai Yen

ABSTRACT: *Feature selection is a technique of selecting a subset of relevant and related features from the original dataset by eliminating noise and redundant features in order to increase performance as well as reduce internal costs in the data mining or machine learning process, This paper proposed a new data classification method based on approximate functional dependence and error measure g_3, The classification experiments on the actual datasets indicate the appropriate of the research direction.*