Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.00043

LỰA CHỌN THUỘC TÍNH THÔNG QUA GOM CỤM SỬ DỤNG MỘT BIẾN THỂ CỦA THÔNG TIN

Phạm Công Xuyên, Nguyễn Thanh Tùng

Đại học Lạc Hồng

pcxuyen@lhu.edu.vn, nttung@lhu.edu.vn

TÓM TẮT: *Lựa chọn thuộc tính là vấn đề rất quan trọng trong phân lớp và gom cụm dữ liệu và rất khó giải quyết khi số lượng các thuộc tính trong tập dữ liệu huấn luyện là rất lớn.*

Bài báo này trình bày một phương pháp lựa chọn thuộc tính thông qua gom cụm sử dụng một metric đặc biệt, đó là một biến thể của thông tin trong lý thuyết thông tin, và thuật toán k-medoids. Khi các thuộc tính đã được gom thành các cụm, các thuộc tính trong cùng một cụm sẽ tương tự nhau, một thuộc tính của một cụm có thể đại diện cho các thuộc tính khác trong cụm; tập các thuộc tính đại diện của các cụm có thể được lấy làm tập thuộc tính rút gọn thay cho tập tất cả các thuộc tính có trong tập dữ liệu ban đầu để thực hiện nhiệm vụ phân lớp các đối tượng. Thuật toán lựa chọn thuộc tính theo phương pháp đề xuất cũng được xây dựng và cài đặt. Kết quả thực nghiệm cho thấy phương pháp đề xuất có khả năng lựa chọn thuộc tính phân lớp với độ chính xác khá cao, khi số k cụm cần gom được lựa chọn một cách thích hợp. Ngoài ra, phương pháp đề xuất có những ưu điểm quan trọng, đó là giúp người sử dụng có thể hiểu được cấu trúc của tập dữ liệu cần phân tích và mức độ quan trọng tương đối giữa các thuộc tính.

Từ khóa: *Lựa chọn thuộc tính, Gom cụm thuộc tính, Phân lớp, k-medoids, Biến thể của thông tin.*

I. MỞ ĐẦU

Cùng với sự phát triển mạnh mẽ của khoa học máy tính, Internet và một số ngành khoa học mới như Tin-sinh học (Bio-informatics), kích thước của những cơ sở dữ liệu con người thu thập được ngày một lớn. Điều này làm cho các thuật toán khai phá dữ liệu cũng như các thuật toán học truyền thống trở nên chậm chạp, không thể xử lý thông tin một cách hiệu quả.

Rút gọn thuộc tính (attribute reduction) hay còn gọi là rút gọn đặc trưng (feature reduction) là quá trình nhằm thu gọn số thuộc tính mô tả các đối tượng mẫu mà không làm mất đi khả năng phân biệt. Rút gọn thuộc tính mang lại nhiều lợi ích cho việc học từ dữ liệu, chẳng hạn cải thiện chất lượng dữ liệu, giảm bớt chi phí tính toán, tránh được hiện tượng quá khớp (overfitting), làm tăng độ hiệu quả dự đoán,…. Do đó, trong những năm gần đây, nghiên cứu các phương pháp rút gọn thuộc tính đã trở thành đề tài thu hút sự quan tâm của nhiều nhà khoa học. Rút gọn thuộc tính đã và đang được ứng dụng rộng rãi vào nhiều lĩnh vực khác nhau như Tin-sinh học, phân loại văn bản, phục hồi ảnh, phát hiện xâm nhập mạng, ….

Cho đến nay, nhiều phương pháp rút gọn thuộc tính đã được đề xuất. Nhìn chung, chúng có thể được chia làm hai loại chính [10, 21]: *lựa chọn* (selection) thuộc tính và *Biến đổi* (transform).

Biến đổi thuộc tính là việc tạo ra một số lượng ít hơn các *thuộc tính mới* từ các thuộc tính ban đầu. Phân tích thành phần chính và phân tích thành phần độc lập là hai phương pháp biến đổi thuộc tính thường được sử dụng [10]. Thông thường, các phương pháp biến đổi thuộc tính cho phép rút gọn thuộc tính xuống mức tối thiểu. Tuy vậy, các phương pháp này có hai nhược điểm quan trọng, đó là đòi hỏi khối lượng tính toán lớn và cho kết quả khó lý giải.

Lựa chọn thuộc tính là quá trình chọn ra một tập con từ tập thuộc tính ban đầu căn cứ vào khả năng phân biệt các đối tượng của các thuộc tính. Trong quá trình tìm lời giải cho một bài toán phân lớp, lựa chọn thuộc tính là bước vô cùng quan trọng; mục đích của nó là tìm ra tập thuộc tính rút gọn cực đại hóa khả năng nhận dạng mẫu. Khái niệm tập thuộc tính rút gọn được sử dụng trong nhiều lĩnh vực. Chẳng hạn, trong lý thuyết tập thô của Pawlak [13, 17]. Với một tập dữ liệu cụ thể, có thể tồn tại nhiều tập thuộc tính rút gọn. Một tập thuộc tính rút gọn được gọi là tối tiểu nếu nó không thể rút gọn được hơn nữa. Tìm một tập thuộc tính rút gọn tối tiểu là bài toán NP-khó [17]. Nhiều nhà nghiên cứu đã đề xuất các phương pháp tìm tập thuộc tính rút gọn xấp xỉ, tức là tập rút gọn chấp nhận một độ dung sai nhất định.

Nhìn chung, có bốn loại phương pháp lựa chọn thuộc tính đã được nghiên cứu, đó là *gói gọn* (wrapper), *lọc* (filter), *tối ưu mục tiêu trực tiếp* (direct objective optimization) và *gom cụm thuộc tính* (attribute clustering) [10, 5]. Thuật toán kiểu gói gọn là thuật toán chọn ra những thuộc tính có hiệu năng dự đoán cao được đánh giá *thông qua việc áp dụng một thuật toán học* cụ thể. Do lấy khả năng dự đoán làm tiêu chuẩn xem xét, các thuật toán gói gọn có thể cho kết quả lựa chọn tốt hơn các thuật toán khác. Tuy vậy, vì mỗi thuật toán kiểu gói gọn gắn liền với một thuật toán học cụ thể nên chúng thường là các thuật toán không tổng quát, lại đòi hỏi một khối lượng tính toán lớn, do đó không khả thi đối với các cơ sở dữ liệu lớn. Ngược lại với phương pháp kiểu gói gọn, phương pháp lọc chủ yếu tìm ra tập thuộc tính rút gọn từ không gian các thuộc tính ban đầu dựa trên một tiêu chuẩn đánh giá đã cho. Tiêu chuẩn đánh giá này *độc lập với thuật toán học*. Do hiệu quả về mặt tính toán, phương pháp lọc thường được sử dụng khi cơ sở dữ liệu có kích thước lớn. Gần đây, Rodriguez và cộng sự [6] đã đề xuất một phương pháp tiên tiến cho phép chuyển bài toán lựa

chọn thuộc tính về bài toán tối ưu hóa kinh điển. Cuối cùng, phương pháp gom cụm thuộc tính sử dụng một ý tưởng rất dễ hiểu, đó là thay thế một cụm các thuộc tính "tương tự nhau" bằng một thuộc tính trung tâm [1-5]. Tuy nhiên, đối với loại phương pháp này, có hai vấn đề cốt lõi cần phải được xem xét một cách kỹ càng, đó là việc lựa chọn hàm đo độ tương tự và thuật toán gom cụm sẽ sử dụng. Hàm đo độ tương tự đo đạc mức độ tương tự giữa hai thuộc tính trong không gian các thuộc tính; thuật toán gom cụm gom các thuộc tính thành các nhóm sử dụng hàm đo độ tương tự đã chọn. Phương pháp lựa chọn thuộc tính thông qua gom cụm có các ưu điểm rất quan trọng, đó là giúp người sử dụng có thể hiểu rõ hơn cấu trúc của tập dữ liệu cần phân tích, mức độ quan trọng tương đối giữa các thuộc tính, mở ra khả năng các thuộc tính trong cùng một cụm có thể thay thế cho nhau khi một thuộc tính nào đó có giá trị bị thiếu.

Gần đây đã có một số công trình nghiên cứu phương pháp lựa chọn thuộc tính thông qua gom cụm. Trong [7], R. Butterworth và công sự đã sử dụng Barthélemy-Montjardet metric và thuật toán gom cụm phân cấp AGNES để gom cụm thuộc tính. Trong [1], T. Hong và Y. Liou đã đề xuất cách tiếp cận bài toán lựa chọn thuộc tính thông qua gom cụm sử dụng độ phụ thuộc tương đối giữa các thuộc tính và thuật toán *k*-means. Phương pháp này cho phép tìm được một tập thuộc tính rút gọn xấp xỉ cho việc phân lớp, đồng thời nhóm được các thuộc tính có độ tương tự cao vào cùng một cụm. Một phương pháp lựa chọn và thay thế thuộc tính sử dụng ý tưởng tương tự cũng đã được T. Hong và cộng sự trình bày trong [3]. Nhằm nâng cao hiệu năng của thuật toán đã trình bày trong [1], trong [2] và [4] T. Hong và cộng sự đã đề xuất một phương pháp gom cụm thuộc tính mới sử dụng độ thay thế thay cho độ phụ thuộc giữa các thuộc tính, thuật giải di truyền (GA) và thuật giải di truyền nhóm (grouping genetic algorithm - GGA) thay cho thuật toán *k*-means để lựa chọn thuộc tính. Độ thay thế của một tập con thuộc tính trong dữ liệu huấn luyện là độ chính xác của nó trong phân lớp. Các thuộc tính sẽ được nhóm vào cùng một cụm nếu chúng có mức độ thay thế cao. Trong [5], X. Zhao và cộng sự đề xuất phương pháp lựa chọn thuộc tính thông qua gom cụm thuộc tính sử dụng hai kỹ thuật hiện đại: hệ số thông tin cực đại và phương pháp gom cụm lan truyền tính tương tự (affinity propagation clustering). Theo các tác giả, hệ số thông tin cực đại là một độ đo hiệu quả sự liên quan, còn lan truyền tính tương tự cho phép gom cụm các đối tượng dựa trên mối quan hệ tự nhiên giữa chúng.

Trong bài báo này, chúng tôi trình bày một phương pháp lựa chọn thuộc tính thông qua gom cụm sử dụng metric là một biến thể của thông tin (variation of information) trong lý thuyết thông tin và thuật toán gom cum *k*-medoids. Lưu ý rằng, cho đến nay, trong số các tiêu chuẩn đánh giá các tập thuộc tính lựa chọn, tiêu chuẩn dựa vào độ đo thông tin là tiêu chuẩn được xem xét nhiều nhất. Lý do là vì entropy thông tin là số đo tốt nhất cho việc lượng hóa độ không chắc chắn của các thuộc tính [20]. Thuật toán *k*-medoids có thể làm việc với ma trận khoảng cách bất kỳ và ít chịu ảnh hưởng bởi các dữ liệu ngoại lai hơn *k*-means [11]. Thuật toán lựa chọn thuộc tính theo phương pháp đề xuất cũng được xây dựng và cài đặt. Kết quả thực nghiệm cho thấy phương pháp đề xuất có khả năng lựa chọn thuộc tính phân lớp với độ chính xác khá cao, khi số cụm k sử dụng để gom các thuộc tính được lựa chọn thích hợp.

Nội dung còn lại của bài báo được bố cục như sau: Mục II trình bày một số khái niệm liên quan. Mục III trình bày thuật toán ASC lựa chọn thuộc tính thông qua gom cụm, cùng với ví dụ minh họa . Các kết quả thực nghiệm và thảo luận được mô tả trong mục IV. Cuối cùng, các kết luận và hướng nghiên cứu tiếp theo được nêu trong mục V.

II. CÁC KIẾN THỨC LIÊN QUAN

A. Metric Biến thể của thông tin (Variation of Information)

Trong phân tích dữ liệu, một bảng dữ liệu gồm n hàng ứng với n đối tượng, p cột ứng với p thuộc tính dùng để mô tả các đối tượng thường được gọi là một hệ thông tin. Hình thức hóa, người ta định nghĩa hệ thông tin là một bộ đôi $S = (U, A)$, trong đó U là một tập hữu hạn, không rỗng các đối tượng, A là một tập hữu hạn, không rỗng các thuộc tính.

Dưới đây, nếu không nói gì khác, ta luôn giả thiết mọi thuộc tính trong A của hệ thông tin xem xét $S = (U, A)$ đều là thuộc tính phạm trù.

Cho hệ thông tin $S = (U, A)$. Nếu ta xây dựng được một metric đo đạc khoảng cách giữa các thuộc tính, khi đó ta có thể đánh giá độ gần nhau, phân cụm các thuộc tính, xác định thuộc tính trung tâm, thuộc tính quan trọng,… Các khả năng này có thể được khai thác, sử dụng vào việc nâng cao độ hiệu quả của các thuật toán khai phá dữ liệu đã có hay tạo ra những thuật toán mới. Nội dung dưới đây sẽ trình bày một phương pháp xây dựng một metric trên tập các thuộc tính.

Cho U là một tập hợp hữu hạn, khác rỗng các đối tượng. Một phân hoạch của U là một họ khác rỗng các tập con $\pi = \{B_1, B_2, \dots, B_s\}$ thỏa mãn $\sum_{i=1}^{s} B_i = U$ và $B_i \cap B_j = \emptyset$ với mọi $i \neq j$. Mỗi tập con B_i được gọi là một khối hay một lớp của π. Người ta thường ký hiệu họ tất cả các phân hoạch của U là PART(U).

Cho hệ thông tin $S = (U, A)$. Mỗi thuộc tính trong A cho phép xác định một phân hoạch của tập các đối tượng U, trong đó hai đối tượng sẽ thuộc vào cùng một khối nếu chúng có chung giá trị về thuộc tính đó.

Cho hệ thông tin $S = (U, A)$, hai thuộc tính $X, Y \in A$. Giả sử các phân hoạch sinh ra bởi X và Y lần lượt là $\pi_X = \{X_1, X_2, \dots, X_m\}$ và $\pi_Y = \{Y_1, Y_2, \dots, Y_n\}$. Khi đó, ta có thể xem X và Y là hai đại lượng ngẫu nhiên rời rạc có phân phối xác suất lần lượt là:

$$X: \begin{pmatrix} 1 & 2 & \dots & m \\ \frac{|X_1|}{|U|} & \frac{|X_2|}{|U|} & & \frac{|X_m|}{|U|} \end{pmatrix} \text{ và } Y: \begin{pmatrix} 1 & 2 & \dots & n \\ \frac{|Y_1|}{|U|} & \frac{|Y_2|}{|U|} & & \frac{|Y_n|}{|U|} \end{pmatrix},$$

trong đó $P(X=i)=|X_i|/|U|$, $i=1,\dots,m$, $P(Y=j)=|Y_j|/|U|$, $j=1,\dots,n$ và $|.|$ chỉ lực lượng của một tập hợp.

Với các phân phối xác suất trên, ta có phân phối xác suất đồng thời của X và Y là:

$$P(X=i,Y=j)=\frac{|X_i\cap Y_j|}{|U|},\quad i=1,\dots,m;\ j=1,\dots,n.$$

Các xác suất có điều kiện:

$$P(X=i\mid Y=j)=\frac{P(X=i,Y=j)}{P(Y=j)}=\frac{|X_i\cap Y_j|}{|Y_j|},\quad i=1,\dots,m;\ j=1,\dots,n.$$

Định nghĩa 4 [14]. Cho hệ thông tin $S=(U,A)$, $X\in A$ và $\pi_X=\{X_1,X_2,\dots,X_m\}$. Entropy của X được định nghĩa là:

$$E(X)=-\sum_{i=1}^{m}P(X=i)\log_2 P(X=i) \quad (1)$$

với quy ước $0\log_2 0=0$.

Định nghĩa 5. Cho hệ thông tin $S=(U,A)$, $X,Y\subseteq A$, $\pi_X=\{X_1,X_2,\dots,X_m\}$ và $\pi_Y=\{Y_1,Y_2,\dots,Y_n\}$. Entropy có điều kiện của X khi đã biết Y được định nghĩa bởi:

$$E(X|Y)=-\sum_{j=1}^{n}P(Y=j)\sum_{i=1}^{m}P(X=i\mid Y=j)\log_2 P(X=i\mid Y=j) \quad (2)$$

$i=1,2,\dots,m$ và $j=1,2,\dots,n$.

Định nghĩa 6. Cho hệ thông tin $S=(U,A)$, $X,Y\subseteq A$, $\pi_X=\{X_1,X_2,\dots,X_m\}$ và $\pi_Y=\{Y_1,Y_2,\dots,Y_n\}$. Entropy đồng thời của X và Y được định nghĩa như sau:

$$E(X,Y)=-\sum_{i=1}^{m}\sum_{j=1}^{n}P(X=i,Y=j)\log_2 P(X=i,Y=j) \quad (3)$$

$i=1,2,\dots,m$ và $j=1,2,\dots,n$.

Áp dụng các công thức (1), (2) và (3) ta có:

$$E(X|Y)=E(X,Y)-E(Y) \quad (4)$$

Mệnh đề 1. [14] Độ đo

$$d(X,Y)=E(X|Y)+E(Y|X) \quad (5)$$

là một metric trên tập tất cả các thuộc tính A trong hệ thông tin $S=(U,A)$, nghĩa là với các thuộc tính bất kỳ $X,Y,Z\in A$, ta đều có:

(i) $d(X,Y)\geq 0$ và dấu đẳng thức "=" xảy ra khi và chỉ khi $X=Y$

(ii) $d(X,Y)=d(Y,X)$

(iii) $d(X,Y)+d(Y,Z)\geq d(X,Z)$.

Chú ý: công thức (5), có thể được viết lại như sau:

$$d(X,Y)=2E(X,Y)-E(X)-E(Y) \quad (6)$$

Ví dụ. Giả sử trong một tập dữ liệu có $|U|=9$, hai thuộc tính $a1$ và $a2$ sinh ra hai phân hoạch trên U lần lượt là:

$\pi_{a1}=\{\{1,2,3,4\},\{5,6,7,8,9\}\}$ và là $\pi_{a2}=\{\{1,2,8,9\},\{3,4,5,6,7\}\}$. Ta có

$$E(a1)=-\frac{4}{9}log_2\frac{4}{9}-\frac{5}{9}log_2\frac{5}{9}=0.991.$$

$$E(a2)=-\frac{6}{10}log_2\frac{6}{10}-\frac{4}{10}log_2\frac{4}{10}=0.991.$$

Phân hoạch của U sinh bởi $\{a1, a2\}$ là $\pi_{\{a1,a2\}}=\{\{1,2\},\{3,4\},\{5,6,7\},\{8,9\}\}$

$$E(a1,a2)=-\frac{2}{9}log_2\frac{2}{9}-\frac{2}{9}log_2\frac{2}{9}-\frac{3}{9}log_2\frac{3}{9}-\frac{2}{9}log_2\frac{2}{9}=1.974.$$

Áp dụng công thức (6), ta thu được khoảng cách giữa a1 và a2 là như sau:

$$d(a1,a2)=2E(a1,a2)-E(a1)-E(a2)=2\times 1.974-0.991-0.991=1.966.$$

B. Phương pháp gom cụm k-medoids

k-medoids là thuật toán gom cụm tương tự như thuật toán *k*-means. Cả hai thuật toán này đều thuộc cách tiếp cận bài toán gom cụm bằng phân hoạch và đều phải cho trước số cụm *k* cần gom. Khác với *k*-means, *k*-medoids chọn *k* medoids làm tâm của *k* cụm. Một medoid được định nghĩa là một đối tượng của một cụm có khoảng cách (độ bất tương tự) trung bình tới tất cả các đối tượng trong cụm là nhỏ nhất, tức là đối tượng nằm ở vị trí trung tâm nhất của cụm. Thuật toán *k*-medoids có thể làm việc với ma trận khoảng cách bất kỳ và *ít chịu ảnh hưởng bởi các dữ liệu ngoại lai hơn k-means*. *Các medoids* đại diện cho các cụm là *các đối tượng thực có trong tập dữ liệu*. Một công cụ hữu hiệu để chọn số *k* cụm cần gom là **silhouettes [**18].

Một thể hiện thông dụng nhất của cách tiếp cận gom cụm sử dụng *k*-medoids là thuật toán PAM (Partitioning Around Medoids). Thuật toán này là như sau [11]:

Bước 1. Chọn ngẫu nhiên *k* đối tượng từ *n* đối tượng có trong tập huấn luyện làm *k* medoids.

Bước 2. Kết hợp mỗi đối tượng với medoid gần nhất, ("gần nhất" ở đây được hiểu theo một metric nào đó).

Tính tổng chi phí E của cấu hình gom cụm hình thành:

$$E = \sum_{j=1}^{k} \sum_{u \in C_j} d(u, o_j),$$

trong đó u là một đối tượng thuộc cụm C_j , o_j là medoid của cụm C_j và $d(u, o_j)$ là khoảng cách giữa hai đối tượng u và o_j.

Bước 3. Với mỗi medoid *m*

Với mỗi đối tượng không phải medoid *o*

Thay *o* bằng *m*, rồi tính tổng chi phí E' của cấu hình gom cụm mới.

Bước 4. Chọn cấu hình gom cụm mới nếu $E' > E$, trường hợp ngược lại giữ nguyên cấu hình cũ.

Bước 5. Lặp lại các bước 3-4 cho đến khi thấy không có sự thay đổi các medoids.

III. THUẬT TOÁN ĐỀ XUẤT

A. Thuật toán lựa chọn thuộc tính ASC

Mục này trình bày thuật toán lựa chọn thuộc tính để giải một bài toán phân lớp thông qua gom cụm các thuộc tính có trong tập dữ liệu huấn luyện đã cho. Để gom cụm thuộc tính chúng tôi sử dụng thuật toán PAM (Partitioning Around Medoids) và metric "Biến thiên thông tin" trình bày trong mục II. Thuật toán đề xuất có tên gọi là ASC (Attibute Selection through Clustering). Để ASC hoạt động, trước hết chúng ta phải thực hiện một số bước tiền xử lý dữ liệu nếu cần thiết, ví dụ như rời rạc hóa dữ liệu liên tục, xử lý các đối tượng có giá trị thuộc tính bị thiếu.

Các bước chính của thuật toán ASC là như sau:

Input: hệ thông tin $S = (U, A)$ với $U = \{u_1, u_2, \dots, u_n\}$ là tập các đối tượng, $A = \{a_1, a_2, \dots, a_p\}$ là tập các thuộc tính; số *k* cụm cần gom; số *N* lần cần thực hiện thuật toán Pam gom cụm các thuộc tính.

Output: Tập con thuộc tính $B(\subset A)$ lựa chọn được.

Bước 1: Lập ma trận khoảng cách $M = (m_{ij})$ với $m_{ij} = d(a_i, a_j)$ là khoảng cách giữa hai thuộc tính a_i và a_j tính theo công thức (6) Mục II, $i, j = 1,2, \dots, p$.

Bước 2: Thực hiện thuật toán thuật toán PAM *N* lần bằng; mỗi lần thực hiện thuật toán cập nhật tần số xuất hiện của các thuộc tính trong tập các medoids đại diện cho các cụm thu được.

Bước 3: Sắp xếp tần số được làm medoid của các thuộc tính và chọn k thuộc tính có tần số cao nhất.

Bước 4: Cho ra tập con thuộc tính lựa chọn được và kết thúc.

Mục đích chính của gom cụm thuộc tính là nhằm lựa chọn các thuộc tính đại diện (các medoids) để thay thế cho tập tất cả các thuộc tính có trong tập huấn luyện. Vì tập các các medoids thu được sau mỗi lần chạy thuật toán PAM là không như nhau, để có được tập các thuộc tính đại diện tin cậy, trong ASC chúng ta cần cho thuật toán PAM chạy một số lớn lần (tùy thuộc vào số cụm k cần gom và số thuộc tính có trong tập dữ liệu) và ghi lại tần số xuất hiện của các thuộc tính trong tập đại diện thu được sau mỗi lần chạy.

B. Ví dụ

Xét tập dữ liệu cho trong Bảng 1.

Bảng 1. Tập dữ liệu "Animal world"

#	Animal	Hair	Teeth	Eye	Feather	Feet	Eat	Milk	Fly	Swim
1	Tiger	Y	pointed	forward	N	claw	meat	Y	N	Y
2	Cheetah	Y	pointed	forward	N	claw	meat	Y	N	Y
3	Giraffe	Y	blunt	side	N	hoof	grass	Y	N	N
4	Zebra	Y	blunt	side	N	hoof	grass	Y	N	N
5	Ostrich	N	N	side	Y	claw	grain	N	N	N
6	Penguin	N	N	side	Y	web	fish	N	N	Y
7	Albatross	N	N	side	Y	craw	grain	N	Y	Y
8	Eagle	N	N	forward	Y	craw	meat	N	Y	N
9	Viper	N	pointed	forward	N	N	meat	N	N	N

Đây là một hệ thông tin có tập thuộc tính điều kiện A = {Hair, Teeth, Eye, Feather, Feet, Eat, Milk, Fly}, gồm 8 thuộc tính của 9 loài động vật, bên cạnh đó có thêm một thuộc tính gọi là thuộc tính quyết định Swim, cho biết một động vật với các thuộc tính đã cho có biết bơi hay không.

Với tập dữ liệu đã cho thuật toán đề xuất ASC hoạt động như sau.

Bước 1: Lập ma trận khoảng cách giữa các cặp thuộc tính. Áp dụng công thức (6) Mục II, ta thu được ma trận khoảng cách cho trong Bảng 2.

Bảng 2. Khoảng cách giữa các cặp thuộc tính

	Hair	Teeth	Eye	Feather	Feet	Eat	Milk	Fly
Hair	0	1.151	1.966	0.802	1.744	1.733	0	1.305
Teeth	1.151	0	1.259	0.539	1.205	1.026	1.151	1.654
Eye	1.966	1.259	0	1.798	1.744	0.845	1.966	1.749
Feather	0.802	0.539	1.798	0	1.744	1.565	0.802	1.115
Feet	1.744	1.205	1.744	1.744	0	0.899	1.744	1.971
Eat	1.733	1.026	0.845	1.565	0.899	0	1.733	2.236
Milk	0	1.151	1.966	0.802	1.744	1.733	0	1.305
Fly	1.305	1.654	1.749	1.115	1.971	2.236	1.305	0

Bước 2: Thực hiện gom cụm các tính bằng thuật toán PAM với ma trận khoảng cách tính được trong bước 1.

Vòng lặp 1.

Bước 2.1.1: Chọn ngẫu nhiên k = 2 thuộc tính làm tâm ban đầu cho hai cụm. Giả sử hai thuộc tính được chọn làm medoid (tâm) ban đầu cho hai cụm C_1 và C_2 tương ứng là Eye và Feet.

Bước 2.1.2: Liên kết các thuộc tính không phải medoid với medoid gần nhất để hình thành các cụm.

Căn cứ vào các khoảng cách cho trong Bảng 3, cụm C_1 sẽ bao gồm {Eat, Fly, Eye}, còn cụm C_2 sẽ là {Hair, Teeth, Feather, Milk, Feet}. Tổng chi phí E của cấu hình gom cụm này là:

Bảng 3. Khoảng cách giữa các thuộc tính không phải tâm cụm và các tâm cụm

Cụm C_1		Cụm C_2	
Cặp thuộc tính	Khoảng cách	Cặp thuộc tính	Khoảng cách
d(Hair, Eye)	1.966	*d*(Hair, Feet)	1.744
d(Teeth, Eye)	1.259	*d*(Teeth, Feet)	1.205
d(Feather, Eye)	1.798	*d*(Feather, Feet)	1.744
d(Eat, Eye)	0.845	*d*(Eat, Feet)	0.899
d(Milk, Eye)	1.966	*d*(Milk, Feet)	1.744
d(Fly, Eye)	1.749	*d*(Fly, Feet)	1.971

$$E_1 = (d(\text{Eat, Eye}) + d(\text{Fly, Eye}))$$
$$+\big(d(\text{Hair, Feet}) + d(\text{Teeth, Feet}) + d(\text{Feather, Feet}) + d(\text{Milk, Feet})\big)$$
$$= (0.845 + 1.749) + (1.744 + 1.205 + 1.744 + 1.744) = 9.031\,.$$

***Bước 2.1.3**:* Chọn một thuộc tính không phải medoid, chẳng hạn Feather. Lấy Feather thay Feed làm medoid của cụm C_2 ($cụm$ chứa Feather).

Kết hợp mỗi đối tượng với medoid gần nhất để hình thành các cụm.

Bảng 4. Khoảng cách giữa các thuộc tính không phải tâm cụm và các tâm cụm

Cụm C_1		Cụm C_2	
Cặp thuộc tính	Khoảng cách	Cặp thuộc tính	Khoảng cách
d(Hair, Eye)	1.966	*d*(Hair, Feather)	0.802
d(Teeth, Eye)	1.259	*d*(Teeth, Feather)	0.539
d(Feet, Eye)	1.744	*d*(Feet, Feather)	1.744
d(Eat, Eye)	0.845	*d*(Eat, Feather)	1.565
d(Milk, Eye)	1.966	*d*(Milk, Feather)	0.802
d(Fly, Eye)	1.749	*d*(Fly, Feather)	1.115

Căn cứ vào các khoảng cách cho trong Bảng 4, ta được C_1 sẽ bao gồm {Eat, Eye}, còn cụm C_2 sẽ là {Hair, Teeth, Feet, Milk, Fly, Feather}.

Tính tổng chi phí E_2 của cấu hình gom cụm mới hình thành:

$$E_2 = d(\text{Eat, Eye}) + (d(\text{Hair, Feather}) + d(\text{Teeth, Feather}) + (\text{Feet, Feather }) + d(\text{Milk, Feather}) + d(\text{Fly, Feather}))$$
$$= 0.845 + (0.802 + 0.539 + 1.744 + 0.802 + 1.115) = \mathbf{5.807}.$$

***Bước 2.1.4**:* Vì $E_2 = 5.807 < 9.031 = E_1$, ta chấp nhận cấu hình gom cụm mới C_1 = {Eat, Eye}, C_2 = {Hair, Teeth, Feet, Milk, Fly, Feather}, với hai medoids là Eye và Feather, tổng chi phí là $E_2 = \mathbf{5.807}$.

Vòng lặp 2. Lặp lại bước 3-4 trong PAM**.**

Bảng 5. Khoảng cách giữa các thuộc tính không phải tâm cụm và các tâm cụm

Cụm C_1		Cụm C_2	
Cặp thuộc tính	Khoảng cách	Cặp thuộc tính	Khoảng cách
d(Feather, Eat)	1.565	*d*(Hair, Feather)	0.802
d(Teeth, Eat)	1.026	*d*(Teeth, Feather)	0.539
d(Feet, Eat)	0.899	*d*(Feet, Feather)	1.744
d(Eye, Eat)	0.845	*d*(Eye, Feather)	1.798
d(Milk, Eat)	1.733	*d*(Milk, Feather)	0.802
d(Fly, Eat)	2.236	*d*(Fly, Feather)	1.115

Bước 2.2.3: Chọn một thuộc tính không phải medoid, chẳng hạn Eat. Lấy Eat thay Eye làm medoid của cụm C_1 ($cụm$ chứa Eat).

Kết hợp mỗi đối tượng với medoid gần nhất để hình thành các cụm.

Căn cứ vào các khoảng cách cho trong Bảng 5, ta được C_1 sẽ bao gồm {Eye, Feet, Eat}, còn cụm C_2 sẽ là {Hair, Teeth, Milk, Fly, Feather} .

Tính tổng chi phí E_3 của cấu hình gom cụm mới hình thành:

$$E_3 = (d(\text{Eat, Eye}) + d(\text{Feet, Eat})) + d(\text{Hair, Feather}) + d(\text{Teeth, Feather}) + d(\text{Milk, feather}) + d(\text{Fly, feather})$$
$$= (0.845 + 0.899) + (0.802 + 0.539 + 0.802 + 1.115) = 5.002.$$

***Bước 2.2.4**:* Vì $E_3 = 5.002 < 5.807 = E_2$, ta chấp nhận cấu hình gom cụm mới $C_1 = \{\text{Eye, Feet, Eat}\}$, $C_2 = \{\text{Hair, Teeth, Milk, Fly, Feather}\}$, với hai medoids là Eat và Feather, tổng chi phí là $E_2 = 5.002$.

Vòng lặp 3. Lặp lại bước 3-4 trong PAM**.**

Tiếp tục lặp lại các bước 3-4 với các thuộc tính không phải medoid khác, có thể thấy cấu hình gom cụm với C_1 = {Eye, Feet, Eat} và C_2 = {Hair, Teeth, Milk, Fly, Feather} là tốt nhất, hai medoids của hai cụm là Eat và Feather, tổng chi phí là $E_3 = 5.002$. Thuật toán kết thúc.

Chúng ta đã cho PAM chạy 20 lần với các medoids ban đầu khác nhau và thu được Eat và Feather là hai thuộc tính có tần số làm medoid cao nhất. Như vậy, **Eat** và **Feather** sẽ được lựa chọn để thay thế tập tất cả các thuộc tính điều kiện để tìm các quy tắc phân lớp động vật có biết bơi hay không.

IV. TÍNH TOÁN THỰC NGHIỆM

Để đánh giá mức hiệu quả của thuật toán ASC, chúng tôi đã tiến hành tính toán thực nghiệm với hai tập dữ liệu lấy từ kho dữ liệu UCI [8], đó là các tập dữ liệu zoo và votes.

Tập dữ liệu **votes** ghi lại ý kiến của 435 nghị sĩ Mỹ về 16 vấn đề, trong đó đối với mỗi vấn đề chỉ được phép trả lời ”yes” hay ”no” và các nghị sĩ thuộc một trong hai đảng: Dân chủ hay Cộng hòa. Như vậy **votes** là tập dữ liệu phân lớp, với 435 đối tượng, 16 thuộc tính điều kiện và một thuộc tính quyết định (cho biết nghị sĩ thuộc đảng nào). Tất cả các thuộc tính đều thuộc loại phạm trù (categorical), một số thuộc tính có giá trị thiếu. Để xử lý giá trị thiếu, chúng tôi đã sử dụng phương pháp Concept Closest Fit [16].

Zoo cũng là tập dữ liệu phân lớp, gồm 101 đối tượng (động vật), mỗi đối tượng được mô tả bởi 16 thuộc tính điều kiện và một thuộc tính quyết định (thể loại động vật, có thể nhận một trong 7 giá trị nguyên từ 1 đến 7). Tất cả các thuộc tính đều thuộc loại phạm trù (categorical), không có giá trị thiếu.

Thuật toán đề xuất ASC được cài đặt trong môi trường hệ thống R và chạy trên máy tính cá nhân với hệ điều hành Windows 10, bộ xử lý Pentium dual core 2.70 GHz CPU, 2.00 GB RAM. Hàm gom cụm pam() trong gói chương trình **cluster** của hệ thống R [15,21] được sử dụng để gom cụm các thuộc tính.

Đối với mỗi tập dữ liệu trên đây, trước tiên chúng tôi tạm thời loại bỏ thuộc tính quyết định, rồi tiến hành lựa chọn thuộc tính bằng thuật toán ASC với các thuộc tính điều kiện còn lại. Thuộc tính quyết định được sử dụng để thực hiện việc kiểm thử, đánh giá khả năng phân lớp của các tập thuộc tính lựa chọn được. Thuật toán phân lớp sử dụng để kiểm thử, đánh giá khả năng này là C5.0 có trong gói chương trình C50 của hệ thống R.

Đối với mỗi tập dữ liệu chúng tôi đã cho thuật toán chạy 120 lần và ghi lại tần số xuất hiện của các thuộc tính trong tập các medoid thu được sau mỗi lần chạy. Bảng 6 và 7 ghi lại tần số làm medoid của các thuộc tính trong hai tập zoo và votes với số k cụm cần gom khác nhau.

Căn cứ vào tần số làm medoid của các thuộc tính cho trong Bảng 6, ta thấy đối với tập dữ liệu zoo, nếu ta gom các thuộc tính thành $k = 7$, thì tập thuộc tính được lựa chọn sẽ là A = {fins (12), eggs (3), feathers (2), backbone (9), milk (4), venomous (11), predator (7)}; còn nếu $k = 11$, thì tập thuộc tính được lựa chọn sẽ là B = {fins (12), eggs (3), backbone (9), feathers (2), milk (4), catsize (16), predator (7), tail (14), domestic (15), toothed (8), legs (13)}.

Căn cứ vào tần số làm medoid của các thuộc tính cho trong Bảng 7, ta thấy đối với tập dữ liệu votes, nếu ta gom các thuộc tính thành $k = 8$, thì tập thuộc tính được lựa chọn sẽ là C = {el-salvador-aid (5), physician-fee-freeze (4), export-administration-act-south-africa (16), aid-to-nicaraguan-contras (8), handicapped-infants (1), religious-groups-in-schools (6), immigration (10), superfund-right-to-sue (13)}; còn nếu $k = 10$, thì tập thuộc tính được lựa chọn sẽ là D = {el-salvador-aid (5), physician-fee-freeze (4), handicapped-infants (1), superfund-right-to-sue (13), export-administration-act-south-africa (16), water-project-cost-sharing (2), religious-groups-in-schools (6), immigration (10), crime (14), duty-free-exports (15)}.

Bảng 6. Số làm medoid của các thuộc tính trong tập zoo

ID	Tên thuộc tính	Số lần làm medoid khi k = 7	Số lần làm medoid khi k = 11
1	hair	34	75
2	feathers	**73**	**88**
3	eggs	**75**	**91**
4	milk	**59**	**83**
5	airborne	39	81
6	aquatic	30	74
7	predator	**53**	**82**
8	toothed	39	**80**
9	backbone	**63**	**89**
10	breathes	29	75
11	venomous	**56**	76
12	fins	**99**	**99**
13	legs	51	**80**
14	tail	52	**82**
15	domestic	37	**82**
16	catsize	51	**83**

Bảng 7. Số làm medoid của các thuộc tính trong tập votes

ID	Tên thuộc tính	Số lần làm medoid **khi k = 8**	Số lần làm medoid **khi k = 10**
1	handicapped-infants	**60**	**76**
2	water-project-cost-sharing	59	**75**
3	adoption-of-the-budget-resolution	54	72
4	physician-fee-freeze	**67**	**78**
5	el-salvador-aid	**84**	**90**
6	religious-groups-in-schools	**60**	**75**
7	anti-satellite-test-ban	52	71
8	aid-to-nicaraguan-contras	**62**	74
9	mx-missile	44	64
10	immigration	**60**	**75**
11	synfuels-corporation-cutback	59	74
12	education-spending	57	74
13	superfund-right-to-sue	**60**	**76**
14	crime	59	**75**
15	duty-free-exports	59	**75**
16	export-administration-act-south-africa	**64**	**76**

Bảng 6 và 7 cho thấy, khi số cụm quy định tăng dần thì tần số được làm medoid (đại diện cụm) của các thuộc tính cũng sẽ trở nên đồng đều hơn. Có hiện tượng này là vì khi số cụm càng tăng thì độ tương tự giữa các thuộc tính trong cùng một cụm cũng sẽ tăng; khả năng được làm medoid của các thuộc tính trở nên đồng đều. Một khi vai trò của các thuộc tính trong cùng một cụm gần như nhau, để lựa chọn một thuộc tính khác làm đại diện ta có thể xem xét đến các yếu tố kinh tế (chẳng hạn chi phí đo đạc) để lựa chọn một thuộc tính khác thay cho medoid của cụm.

Để đánh giá độ hiệu quả của phương pháp lựa chọn thuộc tính bằng thuật toán ASC, chúng tôi đã sử dụng thuật toán C5.0 của gói chương trình C50 trong hệ thống R để xây dựng các thuật phân lớp trên các tập dữ liệu thu được bằng phép chiếu các tập dữ liệu ban đầu lên tập các thuộc tính lựa chọn được từ hai tập dữ liệu zoo và votes. Độ chính xác trung bình thu được của các thuật phân lớp này với kiểm thứ chéo 10 lần được cho trong Bảng 8

Bảng 8. Độ chính xác trung bình của các thuật phân lớp

Tập con thuộc tính được chọn	Độ chính xác phân lớp trung bình
Tập A của tập zoo dữ liệu zoo	90.05 %
Tập B của tập zoo dữ liệu zoo	92.10 %
Tập C của tập zoo dữ liệu votes	87.7 %
Tập D của tập zoo dữ liệu votes	95.4 %

Bảng 8 cho thấy phương pháp đề xuất có khả năng lựa chọn thuộc tính phân lớp với độ chính xác khá cao, và độ chính cũng phụ thuộc vào số cụm ta sử dụng để gom và lựa chọn các thuộc tính.

V. KẾT LUẬN

Trong bài báo này chúng tôi đã trình bày một phương pháp lựa chọn thuộc tính thông qua gom cụm sử dụng một metric đặc biệt, đó là một biến thể của thông tin trong lý thuyết thông tin, và thuật toán *k*-medoids. Một khi các thuộc tính đã được gom thành các cụm theo một độ tương tự, các thuộc tính trong cùng một cụm sẽ tương tự nhau, một thuộc tính của một cụm có thể đại diện cho các thuộc tính khác trong cụm; tập các thuộc tính đại diện của các cụm có thể được lấy làm tập thuộc tính rút gọn thay cho tập tất cả các thuộc tính có trong tập dữ liệu huấn luyện để thực hiện nhiệm vụ phân lớp các đối tượng. Một thuật toán lựa chọn thuộc tính theo phương pháp đề xuất cũng được xây dựng và cài đặt. Kết quả thực nghiệm cho thấy phương pháp đề xuất có khả năng lựa chọn thuộc tính phân lớp với độ chính xác khá cao, khi số cụm *k* sử dụng để gom các thuộc tính được lựa chọn thích hợp. Ngoài ra, phương pháp đề xuất có những ưu điểm rất quan trọng. Đó là:

- Giúp người sử dụng hiểu rõ hơn về cấu trúc của tập dữ liệu cần phân tích, mức độ quan trọng tương đối giữa các thuộc tính và những thuộc tính nào có ảnh hưởng lớn nhất thông qua quá trình gom cụm chúng.

- Cung cấp khả năng biểu diễn các quy tắc phân lớp một cách mềm dẻo hơn, các thuộc tính trong cùng một cụm có thể thay thế cho nhau trong mỗi quy tắc.

Dựa trên các kết quả nghiên cứu được, trong thời gian tới, chúng tôi sẽ nghiên cứu cải tiến thuật toán ASC hơn nữa ở hai khía cạnh: (1) làm sao để lựa chọn được số k cụm thuộc tính cần gom, (2) áp dụng thuật toán gom cụm nào sẽ cho kết quả tốt hơn. Đồng thời chúng tôi cũng sẽ tiến hành thêm các tính toán thực nghiệm nhằm so sánh hiệu năng của phương pháp đề xuất với các phương pháp lựa chọn thuộc tính thông qua gom cụm đã có [1-5], cũng như một số thuật toán thông dụng như Relief và CFS [10, 20] .

TÀI LIỆU THAM KHẢO

[1] T. Hong, and Y. Liou, "Attribute Clustering in High Dimensional Feature Spaces," *The International Conference on Conference on Machine Learning and Cybernetics*, 19-22, 2007.

[2] T. Hong, P. Wang, Y. Lee, An effective attribute clustering approach for feature selection and replacement, *Cybernetics and Systems*, 40 (8), 657–669, 2009.

[3] T. Hong, Y. Liou, S. Wang, Bay Vo, Feature selection and replacement by clustering attributes. *Vietnam Journal of Computer Science* (2014) 1:47–55.

[4] T. Hong, C. Chen, F. Ling, Using group genetic algorithm to improve performance of attribute clustering. *Applied Soft Computing*, Volume 29 Issue C, 371-378, 2015.

[5] X. Zhao, W. Deng, and Y. Shi, Feature Selection with Attributes Clustering by Maximal Information Coefficient. *Procedia Computer Science*, 17 (2013), 70 –79.

[6] I. Rodriguez-Lujan, R. Huerta, C. Elkan, C. Cruz, Quadratic programming feature selection, *Journal of Machine Learning Research*, 11, 1491–1516, 2010.

[7] Richard Butterworth, Gregory Piatetsky-Shapiro, Dan A. Simovici, On Feature Selection through Clustering. *Proceeding ICDM '05, Proceedings of the Fifth IEEE International Conference on Data Mining*, 581-584. http://www.cs.umb.edu/~dsim/papersps/simovici-Feature1.pdf, 2005.

[8] A. Frank, A. Asuncion, UCI machine learning repository, http://archive.ics.uci.edu/ml, university of California, Irvine, School of Information and Computer Sciences (2010).

[9] Cao F. Y., Liang J. Y. , Li D. Y., Bai L., A new initialization method for categorical data clustering, *Expert Syst. Appl*. 36, 10223–10228, 2009.

[10] Guyon E. and Elisseeff A., An introduction to variable and feature selection. J. *of Machine Learning Research*, pages 1157–1182, 2003.

[11] Han, J., Kamber, M., *Data mining: concepts and techniques*. Morgan Kaufmann, San Francisco (2006).

[12] Kaufman L. and Rousseeuw P. J., *Finding Groups in Data – An Introduction to Cluster Analysis*. Wiley Interscience, New York, 1990.

[13] Komorowski, J., Polkowski, L., Skowron, A.: *Rough sets: a tutorial*. http://www.let.uu.nl/esslli/Courses/skowron/skowron.ps

[14] Lopez de Mantaras R., A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning*, 6, N^{o} 1, 81-92, 1991.

[15] J. Maindonald and J. Brown., *Data Analysis and Graphics Using R*. Cambridge University Press, Cambridge, 2003.

[16] Jerzy W. Grzymala-Busse, Witold J. Grzymala-Busse, Handling Missing Attribute Values. Chapter in *Data Mining and Knowledge Discovery Handbook*, Second Edition, Springer US, pp 37-57, 2006.

[17] Pawlak, Z., Skowron, A., Rudiments of rough sets. *Int. J. Comput. Inf. Sci*.177(1), 3–27 (2007).

[18] Peter J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis luster analysis. *Journal of Computational and Applied Mathematics*, 20 (1987), 53-65.

[19] Suchita S. Mesakar, M. S. Chaudhari, Review Paper on Data Clustering of Categorical Data. *International Journal of Engineering Research & Technology*, Vol. 1 Issue 10, December, 2012.

[20] D. Zongker and A. Jain. Algorithms for feature selection: An evaluation. In *Proceedings of the International Conference on Pattern Recognition*, 18–22, 1996.

[21] Yanchang Zhao, *R and Data Mining: Examples and Case Studies*. Published by Elsevier in December 2012. ftp://cran.r-project.org/pub/R/doc/contrib/Zhao_R_and_data_mining.pdf

ATTRIBUTE SELECTION THROUGH CLUSTERING USING A VARIATION OF INFORMATION

Pham Cong Xuyen, Nguyen Thanh Tung

Lac Hong University

pcxuyen@lhu.edu.vn, nttung@lhu.edu.vn

ABSTRACT: *Attributes selection is a very important issue in classifying and clustering data, and is difficult to solve when the number of attributes in the training data set is very large.*

This paper presents a method for selecting attributes through clustering using a special metric, which is a variation of information in information theory, and k-medoids. Since the attributes are grouped into several clusters according to their similarity degrees, an attribute selected from a cluster can thus represent the attributes within the same cluster. The set of representative attributes of clusters can be used for classification such that the whole attributes space can be greatly reduced. The algorithm for selecting attributes according to the proposed method is also implemented to verify its effects. Experimental results suggest that the proposed method has the potential to select attributes for classifying data with high accuracy when the number of clusters needed is appropriately selected. In addition, the proposed approach has the important advantages of allowing users to understand the structure of the analyzed data and the relative importance of attributes for the selection process.