SỬ DỤNG LÍ THUYẾT TẬP THÔ CHO VIỆC TẠO CẤU TRÚC CÂY HAH TRONG PHÂN LỚP ĐA LỚP

VŨ THANH NGUYÊN[*], NGUYỄN ĐẠI HỮU[**], TRẦN ĐẮC TỐT[***]

TÓM TẮT

Trong bài báo này, chúng tôi sử dụng chiến lược phân lớp Half- against-Half và bộ phân lớp nhị phân Support Vector Machines (SVMs) cho bài toán phân lớp đa lớp. Trong đó, để tạo cấu trúc cây cho HAH, chúng tôi đề xuất một thuật toán dựa trên lí thuyết tập thô (Rough Set Theory – RST). Kết quả của thuật toán sẽ được so sánh với một số chiến lược phân đa lớp phổ biến dựa trên bộ phân lớp SVMs.

Từ khóa: lí thuyết tập thô, Haft-against-Haft, máy học hỗ trợ vector.

ABSTRACT

Applying Rough Set Theory in generating HAH tree structure in multi-class classificaiton

In this paper, we use Half- against-Half (HAH) strategy with binary classifier Support Vector Machines (SVMs) for multi-class classification problem, for generating HAH tree structure we propose new algorithm based on Rough Set Theory, the result will be compared with three multi-class classification general strategies of SVMs.

Keywords: Rough Set Theory, Haft-against-Haft, SVMs.

1. Giới thiệu

Hiện có nhiều nghiên cứu về phân lớp văn bản cụ thể: trong [1, 4, 5] giới thiệu một số kĩ thuật máy học cho bài toán phân lớp đa lớp như: Naive Bayes, Decision Tree, K-Láng giềng gần (KNN), mạng Neural, Support Vector Machines (SVMs), thuật toán Rocchio, Giải thuật di truyền. [9] kết hợp fuzzy c-means và fuzzy SVMs (gọi tắt là FCSVM). Trong [9], fuzzy c-means được sử dụng để lọc các dữ liệu gây nhiễu trong tập huấn luyện, sau đó SVMs được sử dụng như bộ phân lớp. [6] kết hợp Lí thuyết tập thô và SVMs cho bài toán phân lớp văn bản, trong đó RST được sử dụng để giảm độ lớp tập thuộc tính qua đó giúp SVMs cho kết quả tốt hơn. Đặc biệt, [1,4,5] nhận xét SVMs là bộ phân lớp được sử dụng phổ biến, và từ kết quả thực nghiệm [5] cho thấy SVMs là thuật toán đạt kết quả tốt nhất.

Tuy nhiên, SVMs là bộ phân lớp nhị phân, để áp dụng cho bài toán phân, một số chiến thuật đã được đề xuất như: OAR (One-against–Rest. Vapnik (1998)), OAO (One-against-One. (Kreβel (1999)), Decision Directed Acyclic Graph (DDAG. Platt et al.

[*] PGS TS, Trường Đại học Công nghệ Thông tin, ĐHQG TPHCM; *Email: nguyenvt@uit.edu.vn*

[**] ThS, Trường Đại học Kinh tế Công nghiệp Long An

[***] ThS, Trường Đại học Công nghiệp Thực phẩm TPHCM

(2000)), HAH (Haft-against-Haft).

Trong các chiến thuật này, HAH yêu cầu huấn luyện ít bộ phân lớp hơn các chiến thuật còn lại, tuy nhiên hiệu quả của HAH lại phụ thuộc vào cấu trúc cây của nó. Vì vậy, việc xây dựng một cấu trúc cây hiệu quả đặc biệt quan trọng trong chiến thuật.

Trong bài báo này, phần 2 chúng tôi giới thiệu về các khái niệm cơ bản của RST, chiến thuật HAH sử dụng các bộ phân lớp SVMs, chúng tôi đề xuất một thuật toán sử dụng RST cho việc tạo cấu trúc cây HAH. Phần 3, chúng tôi trình bày các kết quả đạt được. Phần 4, là phần kết luận và hướng nghiên cứu tiếp theo.

2. Phương pháp

2.1. Lí thuyết tập thô

Hệ thống thông tin (Information System)

Trong lí thuyết tập thô, một hệ thống thông tin là một bộ có dạng *IS= (U, A)*, trong đó U là tập vũ trụ (U khác rỗng, là tập các đối tượng), *A* được gọi là tập thuộc tính (A khác rỗng và xác định).Với mỗi thuộc tính $a \in A$ ta có tương ứng một tập V_a, sao cho $a: U \rightarrow V_a$. V_a được gọi là tập giá trị của a hay miền giá trị của thuộc tính a. $a(x) \in V_a$ được gọi là giá trị thuộc tính a của đối tượng x thuộc U.

Quan hệ bất khả phân biệt (Indiscernibility relation)

Với bất kì $B \subseteq A$, chúng ta có quan hệ:

$IND(B) = \{(x,y) \in UxU | \forall a \in B, a(x) = a(y)\}$

IND(B) gọi là quan hệ B – bất khả phân biệt (*B-indiscernibility relation)*. Nếu $x, y \in IND(B)$, *x* và *y* được gọi là bất khả phân biệt trên tập B. Các lớp tương đương của quan hệ bất khả phân biệt trên B được kí hiệu là $[x]_B$.

Xấp xỉ dưới và xấp xỉ trên (Lower and upper approximations)

Cho một tập đối tượng $X \subseteq U$ và tập thuộc tính B ($B \subseteq A$). X có thể được xấp xỉ bởi các xấp xỉ dưới và xấp xỉ trên.

- *Xấp xỉ dưới (Lower approximation)* (hay miền khẳng định, được kí hiệu bởi $\underline{B}X$) là tập các đối tượng của U mà khi sử dụng các thuộc tính trong B, ta có thể xác định chúng chắc chắn thuộc X:

$$\underline{B}X = \{x \mid [x]_B \subseteq X\}$$

- Xấp xỉ trên (*Upper approximation* - kí hiệu bởi $\bar{B}$X) là tập đối tượng trong U mà sử dụng các thuộc tính trong B ta xác định chúng có thể thuộc X:

$$\bar{B}X = \{x \mid [x]_B \cap X \neq \emptyset\}$$

Định nghĩa tập thô

Tập thô là một bộ $<\underline{B}X, \bar{B}X>$ trong đó $\underline{B}X$ là xấp xỉ dưới và $\bar{B}X$ là xấp xỉ trên.

Độ chính xác thô của việc biểu diễn bởi X được cho bởi (Pawlak 1991):

$0 \leq \alpha_B(X) = \underline{B}X/\bar{B}X \leq 1$

Nếu $\alpha_B(X) = 1$ thì X là tập cổ điển, ngược lại nếu $\alpha_B(X) < 1$ thì X là tập thô.

Sự phụ thuộc thuộc tính

Cho 2 tập phân biệt P, Q của tập thuộc tính. Các lớp tương đương của P cho bởi $[x]_P$, và các lớp tương đương của Q cho bởi $[x]_Q$. Với $[x]_Q = \{Q_1, Q_2, Q_3, \ldots, Q_N\}$. Độ phụ thuộc của tập thuộc tính Q trên tập thuộc tính P, kí hiệu $\gamma_P(Q)$ được cho bởi:

$$\gamma_P(Q) = \frac{\sum_{i=1}^{N} |\underline{P}Q_i|}{|U|} \leq 1 \quad (1)$$

2.2. Support Vector Marchines (SVMs)

Cho tập huấn luyện D gồm n điểm có dạng sau :

$D = \{(x_i, y_i) | x_i \in R^p, y_i \in \{-1, 1\}\}$; i = 1, 2, 3,..., n

trong đó:

x_i là vector p-chiều

y_i được gán 1 hoặc -1 (lớp của điểm thứ i^{th} trong tập huấn luyện)

Ý tưởng của SVMs là tìm một siêu phẳng tối ưu *f(x)* trong không gian p-chiều, mà siêu phẳng này phân chia các điểm có $y_i=1$ (mẫu dương) và $y_i=-1$ (mẫu âm) với lề cực đại. Mỗi siêu phẳng trong không gian p-chiều là tập các điểm x có dạng:

$w^T.x - b = 0$

trong đó:

w^T là vector trọng số

b vô hướng

Để tìm siêu phẳng tối ưu, ta chọn w và b sao cho lề là cực đại. Nghĩa là ta chọn w và b sao cho 2 siêu phẳng song song có khoảng cách cực đại trong khi vẫn có thể phân chia dữ liệu. Hai siêu phẳng song song này được cho bởi:

$w^T.x - b = 1$ và $w^T.x - b = -1$

Nếu các điểm dữ liệu có thể phân chia tuyến tính, siêu phẳng tối ưu là lời giải của bài toán tối ưu sau:

$$\begin{cases} \underset{w}{Min}\ \Phi(w) = \frac{1}{2}\|w\|^2 \\ y_i(w^T.x_i + b) \geq 1, \qquad i = 1, \ldots, l \end{cases}$$

Nếu các điểm dữ liệu trong tập huấn luyện được phân chia tuyến tính và có các điểm nhiễu (các mẫu âm nhưng thuộc phần dương, hoặc các mẫu dương nhưng thuộc phần âm), bài toán trở thành:

$$\begin{cases} Min\Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_i(w^T.x_i + b) \geq 1 - \xi_i \quad i = 1,\dots,l \\ \xi_i \geq 0 \quad i = 1,\dots,l \end{cases}$$

Nếu các điểm dữ liệu trong tập huấn luyện không được phân chia tuyến tính, chúng sẽ được ánh xạ lên một không gian q-chiều (p>q) để chúng có thể được phân chia. Để làm việc này, ta cần định nghĩa một hàm ánh xạ, gọi là hàm nhân (kernel function). Một vài hàm nhân phổ biến:

Linear function: $K(x_i, x_j) = x_i^t.x_j$

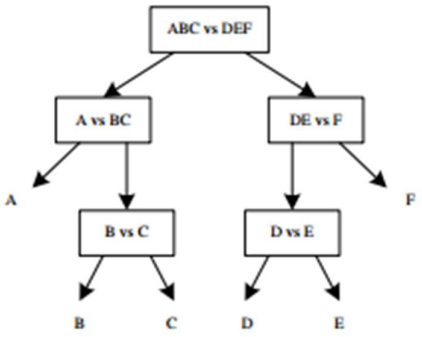
Polynomial function: $K(x_i, x_j) = (x_i.x_j + 1)^d$

Radial basis function-RBF:

$$K(x_i, x_j) = exp(-\gamma(x_i - x_j))^2, \gamma \in R^+$$

2.3. Chiến thuật HAH sử dụng bộ phân lớp nhị phân SVMs

SVMs là bộ phân lớp nhị phân, để sử dụng nó cho bài toán phân lớp đa lớp, người ta sử dụng một số chiến thuật sau: OAO, OAR, DDAG, HAH. Trong các chiến thuật này thì HAH được xây dựng dựa trên việc chia đệ quy N-lớp thành 2 tập lớp. Cấu trúc cây HAH tương tự như cây quyết định, mỗi nút lá là một bộ phân lớp nhị phân SVMs giúp phân một mẫu vào một trong hai lớp xác định. Trong giai đoạn huấn luyện, HAH cây xây dựng (N-1) bộ phân lớp SVMs cho bài toán N-lớp. Và trong giai đoạn phân lớp, để phân lớp một mẫu, HAH cần duyệt qua $log_2(N)$ bộ phân lớp . Hình 1 là ví dụ về một cấu trúc cây HAH cho bài toán 6-Lớp.



Hình 1. *Cấu trúc cây HAH cho bài toán 6 lớp*

Ta sẽ phân tích một số chiến thuật phân lớp đa lớp phổ biến:

OAO (One-against-One): trong chiến thuật này, ở giai đoạn huấn luyện, ta cần xây dựng $\frac{N(N-1)}{2}$ bộ phân lớp SVMs. Trong giai đoạn phân lớp, một mẫu được phân lớp bằng cách duyệt qua $\frac{N(N-1)}{2}$ bộ phân lớp, nếu một mẫu được phân vào lớp i^{th} thì điểm của lớp i^{th} tăng lên một 1. Lớp của mẫu được xác định là lớp có điểm cao nhất.

Tương tự OAO, ***DDAG (Decision Directed Acyclic Graph)*** xây dựng cùng số lượng bộ phân lớp trong giai đoạn kiểm thử, nhưng để phân lớp một mẫu DDAG cần duyệt qua (N-1) bộ phân lớp SVMs.

OAR (One-against-Rest): Ở giai đoạn huấn luyện, ta xây dựng N bộ phân lớp SVMs, mỗi bộ phân lớp sẽ phân một mẫu thuộc về 1 lớp hoặc N-1 lớp còn lại. Trong giai đoạn phân lớp, lớp của mẫu được gán cho bộ SVMs có lề lớn nhất so với các bộ phân lớp còn lại. Nhược điểm của OAR là khi có nhiều hơn 1 lớp có lề lớn nhất thì mẫu không được phân lớp.

Vì vậy, ta thấy HAH cần ít bộ phân lớp hơn cần phải xây dựng trong giai đoạn huấn luyện hơn so với các phương pháp khác. Và trong giai đoạn phân lớp HAH chỉ cần duyệt qua $\log_2(N)$ bộ phân lớp (OAO cần duyệt $\frac{N(N-1)}{2}$, DDAG cần duyệt N-1, và OAR cần duyệt N). Tuy nhiên, hiệu suất HAH lại phụ thuộc vào cấu trúc cây của nó. Trong phần tiếp theo chung tôi sẽ đề xuất một thuật toán tạo cấu trúc cây HAH dựa trên lí thuyết tập thô.

2.4. Sử dụng RST tạo cấu trúc cây HAH

Trong phần này, chúng tôi đề xuất một thuật toán cho việc tạo cấu trúc cây HAH sử dụng RST. Đầu tiên, tập huấn luyện sẽ được tiền xử lí và rút trích đặt trưng. Sau đó, tập huấn luyện sẽ được chuyển thành một Hệ thống Thông tin có dạng *I= (U, A)*, trong đó U là tập các tài liệu trong tập huấn luyện, A là tập thuộc tính (các từ trong tập huấn luyện).

Gọi d là thuộc tính quyết định ($d \in A$ và d định nghĩa lớp của một đối tượng trong U). Từ công thức (1), với mỗi thuộc tính $a \in A\ (a \neq d)$ ta tính độ phụ thuộc của d vào a bởi công thức:

$$\gamma_{\{a\}}(\{d\}) = \frac{\sum_{i=1}^{N} |\underline{P}\{d\}_i|}{|U|} \quad (2)$$

Dựa trên độ phụ thuộc này, ta sắp xếp các thuộc tính trong {A-{d}} giảm dần. Tiếp theo, với mỗi lớp trong tập huấn luyện, ta tạo ra một vector $G= (a_1, a_2, \dots a_c)$, trong đó:

$a_j=0\ (j=1, 2, \dots c)$ nếu a_j không xuất hiện trong lớp, ngược lại $a_j =1\ (a_j \in \{A-\{d\}\})$.

$c =|A|-1$ là số lượng các thuộc tính không phải là thuộc tính quyết định trong tập huấn luyện.

Sau khi có một tập các vector của từng lớp, ta tính độ tương đương của lớp thứ i^{th} với các lớp còn lại. Để tính, chúng tôi đề xuất công thức:

$$sim(v_1, v_2) = \frac{\sum_{k=0}^{c}(C-k)*a_{k1}*a_{k2}}{c} \quad (3)$$

Trong đó a_{k1}, a_{k2} là giá trị thuộc tính thứ k^{th} của vector v_1, v_2

Tổng độ tương tự giữa lớp thứ i^{th} vector và các lớp còn lại được lưu trong phần tử thứ i^{th} của mảng *sim[n]* (trong đó n là số lớp).

Tiếp theo, ta tính trung bình của các phần tử trong *sim[n]*. Dựa trên giá trị trung bình này, ta chia n lớp thành 2, một nhóm (gọi là nhóm trái) gồm các lớp có sim lớn hơn giá trị trung bình, và một nhóm (gọi là nhóm phải) gồm các lớp mà giá trị sim nhỏ hơn giá trị trung bình. Lặp lại đến khi cả nhóm trái và nhóm phải chỉ còn 1 phần tử.

Thuật toán:

Input: Tập huấn luyện D, tập lớp C= {C_1, …, C_n}

Output: H là cấu trúc cây HAH

B 1. Chuyển D thành *I= (U, A)*

B 2. $\forall a \in A\ (a \neq d)$ tính:

$$\gamma_{\{a\}}(\{d\}) = \frac{\sum_{i=1}^{N} |P\{d\}_i|}{|U|}$$

B3. Dựa trên kết quả ở B2, tạo một hệ thống thông tin mới *I'= (U, A')*, trong đó *A'* được sắp xếp giảm dần theo độ phụ thuộc của tập thuộc tính A dựa trên độ phụ thuộc $\gamma_{\{a\}}(\{d\})$.

B4. Với mỗi lớp trong tập huấn luyện D, tạo G= (a_1, a_2,… a_c), trong đó

$$a_j = \begin{cases} 0 \text{ nếu } a_j \text{ không xuất hiện trong lớp} \\ 1 \text{ nếu } a_j \text{ xuất hiện trong lớp} \end{cases}$$

Với j = 1,…,c

B5. Khởi tạo *sim[n]*, n là số lớp trong C

B6. Tính *sim[i]*; với i=0,…, n-1 (n là số lớp); theo công thức:

$\sum\_(k = 0)\text{^}n \blacksquare sim(v\_i, v\_k)$

Trong đó *sim(v_i, v_k)* được tính bởi (3) nếu $i \neq k$, ngược lại *sim(v_i, v_k)=0*

B7. $H = \emptyset, ClassSet = C$, I = 0. (Mỗi phần tử trong ClassSet là 1 tập lớp).

Step 8. $While\ (i! = size\ of\ ClassSet)$

Begin

avg = trung bình độ tương tương của các phần tử trong ClassSet(i);

/ Trong ClassSet(i) là phần tử thứ i của danh sách ClassSet */*

$ClassLeft = \emptyset$;

Add các phần từ trong ClassSet(i) có sim >= avg vào danh sách ClassLetf;

ClassRight = ClassSet – ClassLeft;

/ ClassRigh gồm các phần tử có sim<avg */*

IF(size of ClassLeft >0) Thêm ClassLeft vào ClassSet;

IF(size of ClassRight >0) Thêm ClassRigh vào ClassSet;

H.add(ClassLeft +"vs"+ClassRight);

i++;

End

B9. Return H.

Ở đây, sử dụng bộ dữ liệu Reuters-R8 để diễn giải thuật toán. Bảng 1 cho biết độ tương tự của một lớp với các lớp còn lại.

Bảng 1. *Độ tương tự của một lớp với các lớp còn lại*

	Acq (0)	Crude (1)	Earn (2)	Grain (3)	Interest (4)	money-fx (5)	Ship (6)	Trade (7)
Acq (0)	0	591	730	350	452	504	435	575
Crude (1)	591	0	585	315	404	463	404	512
Earn (2)	730	585	0	339	444	495	422	556
Grain (3)	350	315	339	0	272	302	263	325
Interest (4)	452	404	444	272	0	398	308	407
money-fx (5)	504	463	495	302	398	0	351	476
Ship (6)	435	404	422	263	308	351	0	391
Trade (7)	575	512	556	325	407	476	391	0
Sim[i]	3640	3276	3576	2168	2689	2992	2577	3244

Ta có: $H = \emptyset, ClassSet = C = \{\{0,1,2,3,4,5,6,7\}\}$, i=0

$avg = \frac{\sum_{l=0}^{7} sim[l]}{7} = 3020.8$

Sau đó ta thêm {0, 1, 2, 7} vào *ClassLeft* (các lớp có sim >= avg), thêm {3, 4, 5, 6} tới *ClassRight*

ClassSet={{0, 1, 2, 3, 4, 5, 6, 7} , {0, 1, 2, 7}, {3, 4, 5, 6}}

H={{0, 1, 2, 7 vs 3, 4, 5, 6}}

Tiếp theo, khi i=1

$avg = \frac{sim[0]+sim[1]+sim[2]+sim[7]}{4} = 3434.675$

Thêm {0, 2} vào *ClassLeft*, và {1, 7} vào *ClassRight*.

ClassSet = {{0, 1, 2, 3, 4, 5, 6, 7}, {0, 1, 2, 7}, {3, 4, 5, 6},{0, 2},{1, 7}}

H={{0, 1, 2, 7 vs 3, 4, 5, 6},{0, 2 vs 1, 7}}

Khi i=2

$avg = \frac{sim[3]+sim[4]+sim[5]+sim[6]}{4} = 2606.925$

Thêm {4, 5} vào *ClassLeft*, và {3, 6} vào *ClassRight*.

ClassSe t= {{0, 1, 2, 3, 4, 5, 6, 7}, {0, 1, 2, 7}, {3, 4, 5, 6}, {0, 2},{1, 7},{4, 5},{3,

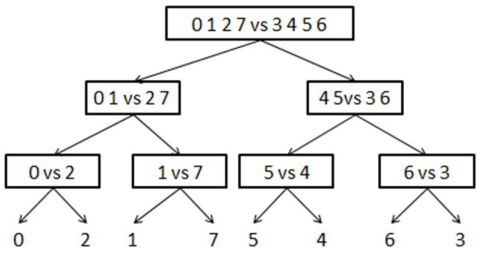
6}}

H={{0, 1, 2, 7 vs 3, 4, 5, 6},{0, 2 vs 1, 7},{4, 5 vs 3, 6}}

Tiếp tục, khi kết thúc thuật toán, ta thu được H như sau:

H={{0, 1, 2, 7 vs 3, 4, 5, 6},{0, 2 vs 1, 7},{4, 5 vs 3, 6}, {0 vs 2}, {1 vs 7}, {5 vs 4}, {6 vs 3}}

Hình 2 chỉ ra cấu trúc HAH dựa trên thuật toán đề xuất.



Hình 2. *Cấu trúc HAH dựa trên thuật toán đề xuất*

3. Kết quả thực nghiệm

Chúng tôi áp dụng phương pháp đề xuất trên 2 bộ dữ liệu: 20 Newsgroups (với 20 danh mục, 11.293 tài liệu trong tập huấn luyện, 7528 trong tập kiểm thử) và Reuters-21.578 R8 (với 8 danh mục, 5485 tài liệu trong tập huấn luyện, 2189 trong tập kiểm thử). Testing System: Intel® Pentium® CPU G630 2.27Ghz x 2, Memory 2GB, OS: Windows 7 Professonal.

Kết quả của phương pháp đề xuất sẽ được so sánh với một số chiến thuật phân đa lớp phổ biến. Bảng 2 biểu diễn kết quả phân lớp trên bộ R8.

Bảng 2. *Kết quả thực nghiệm trên bộ R8*

No	Cat	F-Score			
		OAR	**OAO**	**DDAG**	**HAH**
1	acq	0.961	0.926	0.93	0.928
2	crude	0.769	0.807	0.796	0.801
3	earn	0.981	0.986	0.986	0.986
4	grain	0.3	0.5	0.571	0.533
5	interest	0.638	0.732	0.75	0.741
6	money-fx	0.426	0.6	0.658	0.628
7	ship	0.359	0.609	0.532	0.568
8	trade	0.805	0.832	0.765	0.797
Average		**0.655**	**0.749**	**0.749**	**0.748**

Bảng 3. *Kết quả thực nghiệm trên bộ dữ liệu 20newgroup*

No	Categories	F-Score			
		OVA	**OVO**	**DDAG**	**HAH**
1	alt.atheism	0.542	0.614	0.545	0.568
2	comp.graphics	0.254	0.607	0.452	0.416
3	comp.os.ms-windows.misc	0.354	0.481	0.392	0.474
4	comp.sys.ibm.pc.hardware	0.309	0.556	0.452	0.49
5	comp.sys.mac.hardware	0.429	0.486	0.429	0.558
6	comp.windows.x	0.327	0.546	0.51	0.507
7	misc.forsale	0.544	0.741	0.751	0.61
8	rec.autos	0.547	0.572	0.491	0.675
9	rec.motorcycles	0.693	0.739	0.724	0.783
10	rec.sport.baseball	0.675	0.69	0.622	0.596
11	rec.sport.hockey	0.684	0.689	0.689	0.79
12	sci.crypt	0.659	0.707	0.677	0.734
13	sci.electronics	0.336	0.445	0.455	0.531
14	sci.med	0.444	0.49	0.523	0.598
15	sci.space	0.55	0.619	0.645	0.713
16	soc.religion.christian	0.626	0.744	0.746	0.692
17	talk.politics.guns	0.613	0.706	0.709	0.641
18	talk.politics.mideast	0.604	0.593	0.649	0.725
19	talk.politics.misc	0.355	0.445	0.503	0.555
20	talk.religion.misc	0.315	0.482	0.597	0.476
Average		**0.493**	**0.598**	**0.578**	**0.607**

Bảng 4. *Thời gian huấn luyện và kiểm thử trene các bộ dữ liệu theo chiến thuật phân lớp*

	Reuters-21578 R8		**20 Newsgroup**	
	Training	***Testing***	***Training***	***Testing***
OAR	35	3	1208	30
OAO	21	9	372	302
DDAG	21	9	372	107
HAH	14	2	382	25

4. Kết luận

HAH là một chiến thuật hiệu quả trong phân lớp đa lớp, vì nó yêu cầu xây dựng ít bộ phân lớp hơn trong giai đoạn huấn luyện cũng như duyệt qua ít bộ phân lớp hơn khi phân lớp. Tuy nhiên, hiệu suất của nó lại phụ thuộc cấu trúc cây, trong bài báo này chúng tôi đề xuất phương pháp tạo cây dựa trên RST. Kết quả thực nghiệm cho thấy, phương pháp đề xuất mang lại độ chính xác cao hơn các phương pháp phân lớp khác như: OAO, OVR, và DDAG.

Ghi chú:

Nghiên cứu này được tài trợ bởi Đại học Quốc gia TP Hồ Chí Minh (VNU-HCM) trong đề tài mã số C2014-26-04.

TÀI LIỆU THAM KHẢO

1. Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, Khairullah khan (2010), "A Review of Machine Learning Algorithms for Text-Documents Classification", *Journal of advances in information techology*, Vol. 1 (1), 4-20.
2. D. K. Srivastava, K. S. Patnaik, L. Bhambhu (2010), "Data Classification: A Rough-SVM Approach", *Contemporary Engineering Sciences*, Vol. 3 (2), 77 – 86.
3. Hansheng Lei, Venu Govindaraju (2005), "Half-Against-Half Multi-class Support Vector Machines", *6th International Workshop*.
4. Mita K. Dalal, Mukesh A. Zaveri (2011), "Automatic Text Classification: A Technical Review", *International Journal of Computer Applications*, Vol. 28 (2)–No.2, 37-40.
5. Neha Mehra, Surendra Gupta, (2013), "Survey on Multiclass Classification Methods", *(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 4 (4), 572 – 576.
6. Nasim VasfiSisi, Mohammad Reza Feizi Derakhshi, (2013), "Text Classification with Machine Learning Algorithms", *Journal of Basic and Applied Scientific Research*, 30-35.
7. Tutut Herawan and Wan Maseri Wan Mohd, (2013), "RMF: Rough Set Membership Function-based for Clustering Web Transactions", *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 8 (6), 105-118.
8. Xiaoyong LIU, Hui FU (2012), "A Hybrid Algorithm for Text Classification Problem", *Guangdong Polytechnic Normal University*, 8-11.
9. Vu Thanh Nguyen (2010), "Support Vector Machines Combined With Fuzzy C - Means For Text Classification", *IJCSNS International Journal of Computer Science and Network Security*, Vol.10(3).

(Ngày Tòa soạn nhận được bài: 29-01-2015; ngày phản biện đánh giá: 02-02-2015; ngày chấp nhận đăng: 18-5-2015)