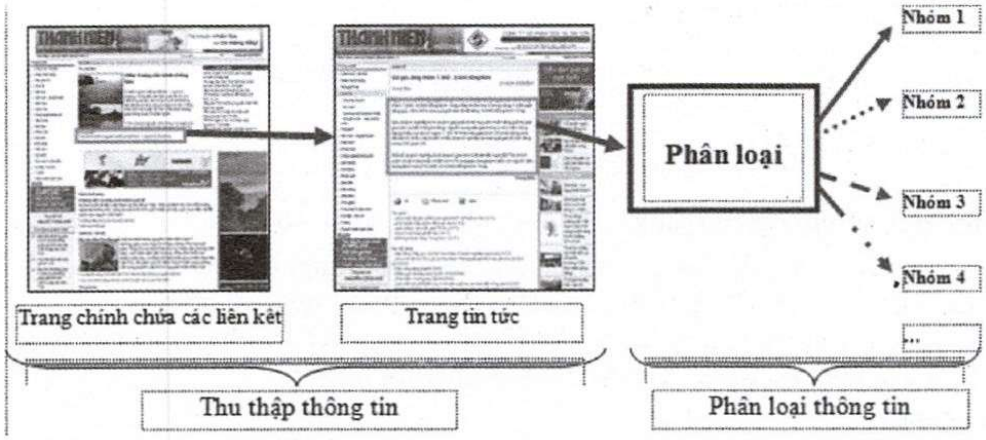
ỨNG DỤNG THUẬT TOÁN PHÂN LỚP RÚT TRÍCH THÔNG TIN VĂN BẢN FSVM TRÊN INTERNET

Vũ Thanh Nguyên[(1)], Trang Nhật Quang[(2)]
(1) Trường Đại học Công nghệ Thông tin, ĐHQG-HCM
(2) Sở Công Nghiệp Thành phố Hồ Chí Minh
(Bài nhận ngày 08 tháng 04 năm 2008, hoàn chỉnh sửa chữa ngày 04 tháng 10 năm 2008)

TÓM TẮT: *Bài báo đã sử dụng kỹ thuật rút trích thông tin tự động và phân loại văn bản bằng phương pháp SVM (Support vector machine), FSVM (Fuzzy SVM), kết hợp với phân loại đa lớp mờ. Kết quả ứng dụng của nghiên cứu dùng trong rút trích thông tin, thu thập tin tức của các website hành chính của các Sở, ban, ngành thành phố nhằm cung cấp cho người dân, doanh nghiệp các thông tin về chủ trương chính sách, thông tin của thành phố trong hoạt động hành chánh công.*

1. GIỚI THIỆU

Hiện đã có một số nghiên cứu về rút trích văn bản và phân loại văn bản, trong bài báo này nhóm nghiên cứu tìm hiểu các kỹ thuật trên và áp dụng vào một ứng dụng thực tế là thu thập và phân loại thông tin trên các trang báo điện tử phục vụ cho việc cung cấp tin tức trên các trang web hành chính thành phố. Các thông tin này có thể do các cơ quan tự cung cấp hoặc thu thập được trên các trang web của Bộ, Chính phủ và các trang báo điện tử khác. Phần thu thập thông tin sử dụng phương pháp nhận dạng mẫu [2],[9], [11] để có thể tự động rút trích thông tin từ các trang web tin tức. Phần phân loại thông tin tác giả sử dụng kỹ thuật phân loại văn bản Fuzzy Support Vector Machines (FSVMs) [12] kết hợp với phân loại đa lớp mờ [5] do kết quả phân loại rất tốt của phương pháp này theo các đề tài đã nghiên cứu 0, [5], [8], [12]. Sơ đồ thực hiện gồm hai bước chính là thu thập thông tin và phân loại thông tin cụ thể như sau:



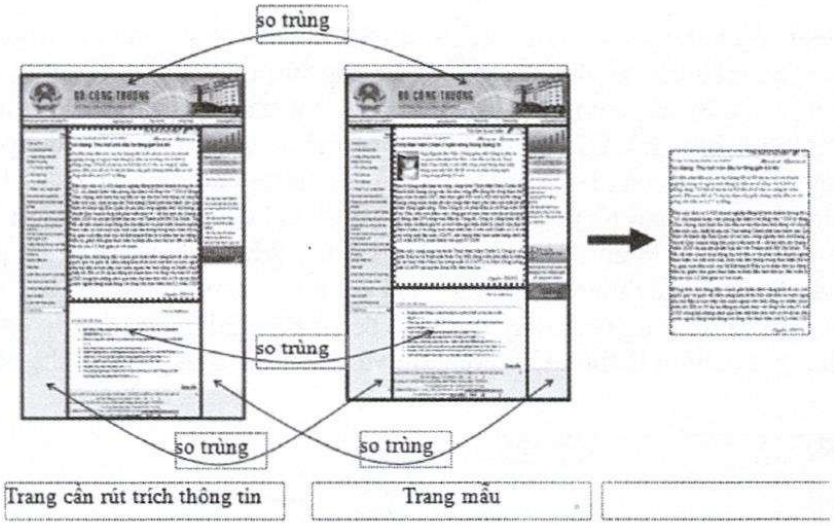
Hình 1. Sơ đồ thực hiện.

2. THU THẬP THÔNG TIN TRÊN TRANG WEB

Hiện nay rút trích thông tin trên web thường được thực hiện bằng cách sử dụng các wrapper. Một wrapper có thể được xem như là một thủ tục được thiết kế để có thể rút trích được những nội dung cần quan tâm của một nguồn thông tin nào đó. Đã có nhiều công trình nghiên cứu khác nhau trên thế giới sử dụng nhiều phương pháp tạo wrapper khác nhau để hiện

thực rút trích thông tin trên web. Các wrapper này được xây dựng bằng tay hoặc phát sinh tự động dựa trên các vùng thông tin người dùng xác định trước trên các trang web mẫu. Wrapper xây dựng theo các phương pháp này có nhược điểm là phải cập nhật lại khi có sự thay đổi cách thức trình bày trên trang web.

Phương pháp rút trích thông tin bằng cách so trùng hai trang web được xây dựng dựa trên phương pháp nhận dạng mẫu ([2]) cho phép rút trích chính xác vùng thông tin mang nội dung chính trên các trang web. Phương pháp này được thực hiện bằng cách so trùng trang web cần rút trích với một trang web mẫu để xác định khung trình bày chung của hai trang web, từ khung trình bày chung ta có thể rút trích ra được nội dung chính của trang web cần rút trích. Phương pháp này không đòi hỏi người dùng phải biết các ngôn ngữ xây dựng wrapper hay phải thay đổi wrapper khi cách trình bày thay đổi do trang web mẫu có thể lấy trực tiếp từ trang chủ và có cùng cách trình bày với trang cần rút trích. Như ví dụ minh họa hình 2: phần thông tin trong khung nét liền là thông tin về khung trình bày giống nhau giữa hai trang web, phần thông tin trong khung nét đứt là phần thông tin khác nhau mang nội dung chính của trang web, đây là nội dung ta cần lấy.



Hình 2. Rút trích thông tin bằng phương pháp so trùng.

2.1.Rút trích thông tin từ trang web bằng phương pháp so trùng

Để thực hiện rút trích thông tin bằng phương pháp so trùng, hai trang web được phân tích thành hai cây đa phân có gốc A và B rồi tiến hành so trùng trên hai cây đa phân này. Nhóm nghiên cứu sử dụng thư viện HtmlParser để phân tích trang web thành cây đa phân có gốc. Cây đa phân có ba loại nút: TagNode, TextNode và RemarkNode.

Định nghĩa.

Ma trận W: số tối đa các cặp nút so trùng giữa các cây con cấp một của A và B

Ma Trận T: trong đó T[i, j] là độ so trùng của hai rừng cây con cấp 1: A_1, A_2,..., A_i của A và B_1, B_2 ,..., B_j của B. T[i,j] được tính dựa trên T[i,j-1], T[i-1][j], và T[i-1][j-1]). Cần thực hiện các phép biến hoán vị như sau:

T1 = T[i, j-1]

T2 = T[i-1, j]

T3 = T[i-1, j-1]

T[i, j] = max (T1, T2, T3 + W[i, j])

Ma trận G : Trong đó G[i][j] lưu giữ danh sách các tham khảo đến các nút rút trích được của cây con cấp một thứ i của nút gốc A khi thực hiện giải thuật so trùng hai cây con cấp một thứ i của A và thứ j của B.

Danh sách M: Trong đó M[i][j] lưu giữ danh sách các cặp nút được so trùng khi tiến hành giải thuật so trùng giữa hai rừng cây con cấp 1: A_1, A_2,..., A_i của A và B_1, B_2 ,..., B_j của B.

Hai nút là giống nhau nếu:

Nếu hai nút cùng có kiểu TagNode, thì chỉ cần TagName của chúng giống nhau thì xem như hai nút giống nhau.

Nếu hai nút cùng có kiểu TextNode hay RemarkNode thì chỉ khi toàn bộ nội dung văn bản của nút này giống nội dung của nút kia thì hai nút mới được xem là giống nhau. Các trường hợp khác ngoài hai trường này thì đều được xem là hai nút khác nhau.

Đầu vào: Hai nút gốc cây đa phân của trang web cần rút trích (A) và trang web mẫu (B).

Đầu ra: Số nút tối đa của việc so trùng : weight; danh sách các tin rút trích được : retList.

Thuật giải so trùng cụ thể như sau:

TH 1 : Hai nút gốc A và B không giống nhau:

Danh sách các tin rút trích được trả về của giải thuật:

retList = null

Số nút tối đa của của việc so trùng giữa A và B:

weight = 0

TH 2 : Nút A không có nút con

Danh sách các tin rút trích được trả về của giải thuật:

retList = null

Số nút tối đa của của việc so trùng giữa A và B:

weight = 1

TH 3: Nút A có nút con, nút B không có nút con

Danh sách các tin rút trích được trả về của giải thuật:

retList = các nút con chứa tin tức của nút A

Số nút tối đa của của việc so trùng giữa A và B:

weight = 1

TH 4: Nút A và B đều có nút con

Khởi tạo:

Gọi n, m lần lượt là số cây con cấp 1 của A và B

Với mọi i = 1... n, j = 1 ..m

T[i][j] = 0

M[i][j] = null

G[i][j] = null

Tiến hành so trùng:

Với mọi i = 1... n, j = 1 ..m

T1 = T[i][j - 1]

T2 = T[i - 1][j]

Gọi đệ quy giải thuật so trùng trên cây con thứ i của A và j của B:

G[i][j] = danh sách các nút rút trích được trả về từ giải thuật so trùng

tmpWeight = số nút tối đa của việc so trùng từ giải thuật so trùng

T3 = T[i - 1][j - 1] + tmpWeight

T[i][j] = max(T1, T2, T3);

Nếu (T[i][j] == T1) thì M[i][j] = M[i][j-1];

Nếu (T[i][j] == T2) thì M[i][j] = M[i - 1][j];

Nếu (T[i][j] == T3) thì M[i][j] = M[i - 1][j - 1] ∪ (i, j);

Thực hiện rút trích thông tin:

Danh sách các tin rút trích retList được trả về của giải thuật gồm:

Các nút chứa tin tức không tham gia vào phép so trùng.

Danh sách các tin rút trích được từ phép so trùng được chứa trong G:

Gọi k là số cây con cấp 1 của A tham gia vào phép so trùng.

Với mọi i = 1…k

Gọi posA, posB lần lược là vị trí cây con cấp 1 của A và B tham gia vào phép so trùng.

posA = M[m][n].get(i).posA

posB = M[m][n].get(i).posB

retList = retList ∪ G[posA][posB]

Số nút tối đa của của việc so trùng giữa A và B:

weight= T[m][n] + 1

2.2.Tìm kiếm các trang web tin tức

Có thể nhận thấy các trang chủ của các trang tin tức thường được cập nhật liên kết đến những trang tin mới nhất. Vì vậy để tìm kiếm các tin tức mới nhất ta phải bắt đầu từ các trang chủ. Các liên kết thu được từ trang chủ này có thể dẫn đến một trang web tin tức, hoặc không phải. Dễ dàng xác định đâu là trang web tin tức bằng cách lần lược so sánh trang web đó với một trang web tin tức mẫu. Nếu trang web đó có cùng cách trình bày với trang web mẫu này thì được xem như là trang web tin tức. Để kiểm tra một trang web có cùng cách trình bày với một trang web mẫu hay không, sử dụng khái niệm tỉ lệ khung K của một trang web ([2])

Tỉ lệ khung K = weight/Tổng số nút của trang web cần rút trích.

ở đó, weight là số nút so trùng giữa hai trang web khi tiến hành giải thuật ở trên

Trang web nếu có cùng cách trình bày với trang web mẫu sẽ có tỉ lệ khung K∈[Kmin, 1).

Kmin = Số nút tạo nên khung trang web mẫu /Tổng số nút của trang web mẫu

Sau khi xác định đâu là trang web tin tức, ta sẽ tiến hành rút trích các thông tin này ra, dành cho việc phân loại thông tin, bằng cách sử dụng giải thuật đã nêu ở trên.

3. PHÂN LOẠI THÔNG TIN

3.1.Phương pháp SVM (Support vector machines).

Chúng ta hãy xem xét một bài toán phân loại văn bản bằng phương pháp SVMs (0, [10], [12]) cụ thể như sau:

Bài toán: Kiểm tra xem một tài liệu bất kỳ d thuộc hay không thuộc một phân loại c cho trước? Nếu d∈c thì d được gán nhãn là 1, ngược lại thì d được gán nhãn là –1.

Giả sử, chúng ta lựa chọn được tập các đặc trưng là $T=\{t_1, t_2, \ldots, t_n\}$, thì mỗi văn bản d_i sẽ được biểu diễn bằng một vector dữ liệu $x_i=(w_{i1}, w_{i2}, \ldots, w_{in})$, $w_{ij}\in R$ là trọng số của từ t_j trong

văn bản d_i. Như vậy, tọa độ của mỗi vector dữ liệu x_i tương ứng với tọa độ của một điểm trong không gian R^n.

Dữ liệu huấn luyện của SVMs là tập các văn bản đã được gán nhãn trước Tr=$\{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$, trong đó, x_i là vector dữ liệu biểu diễn văn bản d_i ($x_i \in R^n$), $y_i \in \{+1, -1\}$, cặp (x_i, y_i) được hiểu là vector x_i được gán nhãn là y_i. Ý tưởng của SVMs là tìm một mặt hình học (siêu phẳng) f(x) "tốt nhất" trong không gian n-chiều để phân chia dữ liệu sao cho tất cả các điểm x_+ được gán nhãn 1 thuộc về phía dương của siêu phẳng ($f(x_+)>0$), các điểm x_- được gán nhãn –1 thuộc về phía âm của siêu phẳng ($f(x_-)<0$). Với bài toán phân loại SVMs, một siêu phẳng phân chia dữ liệu được gọi là "tốt nhất", nếu khoảng cách từ điểm dữ liệu gần nhất đến siêu phẳng là lớn nhất. Khi đó, việc xác định một tài liệu $x \notin Tr$ có thuộc phân loại c hay không, tương ứng với việc xét dấu của f(x), nếu $f(x)>0$ thì $x \in c$, nếu $f(x) \leq 0$ thì $x \notin c$.

Cho tập dữ liệu

$$Tr = \{(x_1, y_1), \ldots, (x_l, y_l)\}, \qquad x_i \in R^n, y_i \in \{-1, 1\}$$

Trường hợp 1

Tập dữ liệu Tr có thể phân chia tuyến tính được mà không có nhiễu thì chúng ta có thể tìm được một siêu phẳng tuyến tính có dạng (1) để phân chia tập dữ liệu này. Siêu phẳng tốt nhất tương đương với việc giải bài toán tối ưu sau:

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

Trường hợp 2

Tập dữ liệu huấn luyện *Tr* có thể phân chia được tuyến tính nhưng có nhiễu nghĩa là điểm có nhãn dương nhưng lại thuộc về phía âm của siêu phẳng, điểm có nhãn âm thuộc về phía dương của siêu phẳng. Bài toán (1) trở thành:

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

ξ_i gọi là các biến nới lỏng (slack variable) $\xi_i \geq 0$.

C là tham số xác định trước, định nghĩa giá trị ràng buộc, C càng lớn thì mức độ vi phạm đối với những lỗi thực nghiệm càng cao.

Trường hợp 3

Tuy nhiên không phải tập dữ liệu nào cũng có thể phân chia tuyến tính được. Trong trường hợp này, chúng ta sẽ ánh xạ các vector dữ liệu *x* từ không gian n-chiều vào một không gian m-chiều (m>n) , sao cho trong không gian m-chiều này tập dữ liệu có thể phân chia tuyến tính được. Giả sử ϕ là một ánh xạ phi tuyến tính từ không gian R^n vào không gian R^m.

$$\phi: \; R^n \rightarrow R^m$$

Khi đó, vector x_i trong không gian R^n sẽ tương ứng với vector $\phi(x_i)$ trong không gian R^m.

Thay $\phi(x_i)$ vào (2) ta có (3):

$$\begin{cases} \text{Min } \Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_i(w^T.\phi(x_i)+b) \geq 1-\xi_i, \quad i=1,...,l \\ \xi_i \geq 0 \quad i=1,...,l \end{cases} \tag{3}$$

Việc tính toán trực tiếp $\phi(x_i)$ là phức tạp và khó khăn. Nếu biết hàm nhân (Kernel function) $K(x_i, x_j)$, để tính tích vô hướng $\phi(x_i)\phi(x_j)$ trong không gian m-chiều, thì chúng ta không cần làm việc trực tiếp với ánh xạ $\phi(x_i)$.

$$K(x_i, x_j) = \phi(x_i)\phi(x_j) \tag{4}$$

Một số hàm nhân hay dùng trong phân loại văn bản là :

Hàm tuyến tính (linear): $K(x_i, x_j) = x_i^T x_j$ (5)

Hàm đa thức (polynomial function) : $K(x_i, x_j)=(x_i x_j+1)^d$ (6)

Hàm RBF (radial basis function) : $K(x_i, x_j)=exp(-\gamma(x_i-x_j)^2), \ \gamma \in R^+$ (7)

3.2.Phương pháp FSVM (Fuzzy Support Vector Machines)

Trong SVMs thông thường thì các điểm dữ liệu đều có giá trị như nhau, mỗi một điểm sẽ thuộc hoàn toàn vào một trong hai lớp. Tuy nhiên trong nhiều trường hợp có một vài điểm sẽ không thuộc chính xác vào một lớp nào đó, những điểm này được gọi là những điểm nhiễu, hơn nữa mỗi điểm dữ liệu có thể sẽ không có ý nghĩa như nhau đối với siêu phẳng. Để giải quyết vấn đề này Lin CF. và Wang SD (2002) đã giới thiệu phương pháp FSVMs bằng cách sử dụng một hàm thành viên để xác định giá trị đóng góp của mỗi điểm dữ liệu đầu vào của SVMs vào việc hình thành siêu phẳng.

Bài toán được mô tả như sau:

$$\begin{cases} \text{Min } \Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}s_i\xi_i \\ y_i(w^T.\phi(x_i)+b) \geq 1-\xi_i, \quad i=1,...,l \\ \xi_i \geq 0 \quad i=1,...,l \end{cases} \tag{8}$$

s_i là một hàm thành viên thỏa $\sigma \leq s_i \leq 1$, σ là một hằng số đủ nhỏ > 0 thể hiện mức độ ảnh hưởng của điểm x_i đối với một lớp. Giá trị s_i có thể làm giảm giá trị của biến ξ_i, vì vậy điểm x_i tương ứng với ξ_i có thể được giảm mức độ ảnh hưởng hơn.

Bằng cách sử dụng các hệ số Lagrangian ta có thể chuyển về bài toán lập trình Quadratic:

$$\begin{cases} \max_{\alpha} L_D(\alpha) = \sum_{i=1}^{l}a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ \sum_{i=1}^{l}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq s_i C \quad i=1,2,...,l \end{cases} \tag{9}$$

Chọn hàm thành viên

Việc chọn hàm thành viên s_i thích hợp rất quan trọng trong FSVMs. Theo Chun hàm thành viên s_i dùng để giảm mức độ ảnh hưởng của những điểm dữ liệu nhiễu được mô tả trong [12]

là một hàm xác định khoảng cách giữa điểm dữ liệu x_i với trung tâm của nhóm tương ứng với x_i.

Gọi C^+ là tập chứa các điểm x_i với $y_i = 1$

$C^+ = \{x_i | x_i \in S$ và $y_i = 1\}$

Tương tự gọi $C^- = \{x_i | x_i \in S$ và $y_i = -1\}$

X_+ và X_- là trung tâm của lớp C^+, C^-.

Bán kính của lớp C^+

$$r_+ = max||X_+ - x_i|| \qquad với\ x_i \in C^+ \tag{10}$$

và bán kính của lớp C^- là:

$$r_- = max||X_- - x_i|| \qquad với\ x_i \in C^- \tag{11}$$

Hàm thành viên s_i được định nghĩa như sau:

$$s_i = \begin{cases} 1 - \| X_+ - x_i \| / (r_+ + \delta) & \text{nếu } x_i \in C^+ \\ 1 - \| X_- - x_i \| / (r_- + \delta) & \text{nếu } x_i \in C^- \end{cases} \tag{12}$$

δ là một hằng số để tránh trường hợp $s_i = 0$

Tuy nhiên FSVMs với hàm thành viên (12) vẫn chưa đạt kết quả tốt do việc tính toán khoảng cách giữa các điểm dữ liệu với trung tâm của nhóm được tiến hành ở không gian đầu vào, không gian n chiều. Trong khi đó trong trường hợp tập dữ liệu không thể phân chia tuyến tính, để hình thành siêu phẳng ta phải đưa dữ liệu về một không gian khác với số chiều m cao hơn gọi là không gian đặc trưng (feature space). Vì vậy để có thể đạt kết quả tốt hơn, Xiufeng Jiang, Zhang Yi và Jian Cheng Lv (2006) đã xây dựng một hàm thành viên khác dựa trên ý tưởng của hàm thành viên (12) nhưng được tính toán trong không gian đặc trưng m chiều.

Giả sử ϕ là một ánh xạ phi tuyến tính từ không gian R^n vào không gian R^m.

$$\phi: R^n \to R^m$$

Khi đó, vector x_i trong không gian R^n sẽ tương ứng với vector $\phi(x_i)$ trong không gian R^m.

Định nghĩa ϕ_+ là trung tâm của lớp C^+ trong không gian đặc trưng:

$$\phi_+ = \frac{1}{n_+} \sum_{x_i \in C^+} \phi(x_i) \tag{13}$$

n_+ là số phần tử của lớp C^+

và ϕ_- là trung tâm của lớp C^- trong không gian đặc trưng:

$$\phi_- = \frac{1}{n_-} \sum_{x_i \in C^-} \phi(x_i) \tag{14}$$

n_- là số phần tử của lớp C^-

Định nghĩa bán kính của C^+:

$$r_+ = \max \| \phi_+ - \phi(x_i) \| \qquad với\ x_i \in C^+ \tag{15}$$

và bán kính của C^-:

$$r_- = \max \| \phi_- - \phi(x_i) \| \qquad với\ x_i \in C^- \tag{16}$$

Khi đó,

$$\begin{aligned} r_+^2 &= \max \| \phi(x') - \phi_+ \|^2 \\ &= \max \{ \phi^2(x') - 2\phi(x').\phi_+ + \phi_+^2 \} \\ &= \max \{ \phi^2(x') - \frac{2}{n_+} \sum_{x_i \in C^+} \phi(x_i).\phi(x') + \frac{1}{n_+^2} \sum_{x_i \in C^+} \sum_{x_j \in C^+} \phi(x_i).\phi(x_j) \} \\ &= \max \{ K(x',x') - \frac{2}{n_+} \sum_{x_i \in C^+} K(x_i,x') + \frac{1}{n_+^2} \sum_{x_i \in C^+} \sum_{x_j \in C^+} K(x_i,x_j) \} \end{aligned} \tag{17}$$

Với x' ∈ C^+ và n_+ là số mẫu huấn luyện trong lớp C^+. Tương tự :

$$r_-^2 = \max\{K(x',x') - \frac{2}{n_-}\sum_{x_i \in C^-} K(x_i,x') + \frac{1}{n_-^2}\sum_{x_i \in C^-}\sum_{x_j \in C^-} K(x_i,x_j)\} \quad (18)$$

Với x' ∈ C^- và n_- là số mẫu huấn luyện trong lớp C^-.

Bình phương khoảng cách giữa $x_i \in C^+$ và trung tâm của lớp trong không gian đặc trưng có thể được tính như sau:

$$d_{i+}^2 = K(x_i,x_i) - \frac{2}{n_+}\sum_{x_j \in C^+} K(x_i,x_j) + \frac{1}{n_+^2}\sum_{x_j \in C^+}\sum_{x_k \in C^+} K(x_j,x_k)\} \quad (19)$$

Tương tự như vậy bình phương khoảng cách giữa $x_i \in C^-$ và trung tâm của lớp trong không gian đặc trưng có thể được tính như sau:

$$d_{i-}^2 = K(x_i,x_i) - \frac{2}{n_-}\sum_{x_j \in C^-} K(x_i,x_j) + \frac{1}{n_-^2}\sum_{x_j \in C^-}\sum_{x_k \in C^-} K(x_j,x_k)\} \quad (20)$$

Với mỗi i (i=1,…, *l*), hàm thành viên s_i được mô tả như sau:

$$s_i = \begin{cases} 1 - \sqrt{\| d_{i+}^2 \| / (r_+^2 + \delta)} & \text{nếu } x_i \in C^+ \\ 1 - \sqrt{\| d_{i-}^2 \| / (r_-^2 + \delta)} & \text{nếu } x_i \in C^- \end{cases} \quad (21)$$

Ta thấy s_i là một hàm của trung tâm và bán kính của mỗi lớp trong không gian đặc trưng.

Theo kết quả thử nghiệm của Xiufeng Jiang, Zhang Yi và Jian Cheng Lv hàm thành viên theo công thức (21) bằng cách sử dụng hàm nhân để tính toán trong không gian m chiều có thể làm giảm ảnh hưởng của các điểm nhiễu hiệu quả hơn hàm thành viên của Lin CF và Wang SD và cho kết quả phân loại tốt hơn [12].

4. PHÂN LOẠI ĐA LỚP

Ý tưởng của bài toán phân loại đa lớp là chuyển về bài toán phân loại hai lớp bằng cách xây dựng nhiều bộ phân loại hai lớp để giải quyết. Các chiến lược phân loại đa lớp phổ biến này là One-against-One (OAO) và One-against-Rest (OAR) ([5] - [7]).

4.1.Chiến lược One-against-Rest (OAR)

Trong chiến lược này ta sử dụng (n-1) bộ phân loại đối với n lớp. Bài toán phân loại n lớp được chuyển thành n bài toán phân loại hai lớp. Trong đó bộ phân loại hai lớp thứ i được xây dựng trên lớp thứ i và tất cả các lớp còn lại. Hàm quyết định thứ i dùng để phân lớp thứ i và những lớp còn lại có dạng:

$$D_i(x) = w_i^t x + b_i$$

Siêu phẳng $D_i(x) = 0$ hình thành siêu phẳng phân chia tối ưu, các support vector thuộc lớp i thỏa $D_i(x) = 1$ và các support vector thuộc lớp còn lại thỏa $D_i(x) = -1$. Nếu vector dữ liệu x thỏa mãn điều kiện $D_i(x) > 0$ đối với duy nhất một i, x sẽ được phân vào lớp thứ i.

Tuy nhiên nếu điều kiện $D_i(x) > 0$ thỏa mãn đối với nhiều i, hoặc không thỏa đối với i nào thì trong trường hợp này ta không thể phân loại được vector x. Để giải quyết vấn đề này chiến lược One-against-One (OAO) được đề xuất sử dụng.

4.2.Chiến lược One-against-One (OAO)

Trong chiến lược này ta sử dụng n(n-1)/2 bộ phân loại hai lớp được xây dựng bằng cách bắt cặp từng hai lớp một và sử dụng phương pháp lựa chọn theo đa số để kết hợp các bộ phân loại này để xác định được kết quả phân loại cuối cùng. Số lượng các bộ phân loại là n(n-1)/2.

So với chiến lược OAR thì chiến lược này ngoài ưu điểm giảm bớt vùng không thể phân loại mà còn làm tăng độ chính xác của việc phân loại ([3],[4]). Trong chiến lược OAR ta phải xây dựng một siêu phẳng để tách một lớp ra khỏi các lớp còn lại, việc này đòi hỏi sự phức tạp và có thể không chính xác. Tuy nhiên trong chiến lược OAO ta chỉ cần phân tách một lớp ra khỏi một lớp khác mà thôi.

Chiến lược OAR chỉ cần n-1 bộ phân loại cho n lớp. Trong khi đó chiến lược OAO lại cần đến n(n-1)/2 bộ phân loại. Nhưng số mẫu huấn luyện cho từng bộ phân loại trong OAO lại ít hơn và việc phân loại cũng đơn giản hơn. Vì vậy chiến lược OAO có độ chính xác cao hơn nhưng chi phí để xây dựng lại tương đương với chiến lược OAR ([3],[4]).

Hàm quyết định phân lớp của lớp i đối với lớp j trong chiến lược OAO là:

$$D_{ij}(x) = w_{ij}^t x + b_{ij}$$

$$D_{ij}(x) = -D_{ji}(x)$$

Đối với một vector x ta tính :

$$D_i(x) = \sum_{j \neq i, j=1}^{n} sign(D_{ij}(x))$$

với $$sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Và x được phân vào lớp i sao cho: $\arg\max_{i=1,\ldots,n} D_i(x)$

Tuy nhiên nếu điều kiện $\arg\max_{i=1,\ldots,n} D_i(x)$ được thỏa mãn đối với nhiều i thì trong trường hợp này cũng không thể xác định được x thuộc lớp nào.

Để giải quyết vấn đề này Shigeo Abe và Takuya Inoue đã giới thiệu Phân loại đa lớp mờ ([5]).

4.3.Phân loại đa lớp mờ

Phương pháp phân loại đa lớp mờ được xây dựng trên phương pháp phân loại đa lớp OAO kết hợp với việc sử dụng một hàm thành viên để xác định kết quả phân loại khi vector x không thể phân loại được theo chiến lược OAO.

Đối với siêu phẳng tối ưu $D_{ij}(x) = 0 (i \neq j)$ chúng ta định nghĩa các hàm thành viên như sau:

$$m_{ij}(x) = \begin{cases} 1 & \text{với } D_{ij}(x) \geq 1, \\ D_{ij}(x) & \text{còn lại} \end{cases}$$

Từ các $m_{ij}(x)(j \neq i, j = 1,\ldots,n)$, chúng ta định nghĩa hàm thành viên thứ i của vector x như sau:

$$m_i(x) = \min_{j=1,\ldots,n} m_{ij}(x)$$

Công thức trên tương đương với

$$m_i(x) = \min_{j \neq i, j=1,\ldots,n} D_{ij}(x)$$

Bây giờ x được phân loại vào lớp i theo công thức

$$\arg\max_{i=1,\ldots,n} m_i(x)$$

5.KẾT QUẢ THỬ NGHIỆM

Nhóm nghiên cứu tiến hành xây dựng chương trình thu thập và phân loại đối với những thông tin thuộc lĩnh vực ngành Công nghiệp với năm nhóm ngành con: Dệt May (gồm dệt may và da giầy); Cơ Khí (các ngành cơ khí, ô tô); Điện và Dầu Khí (dầu khí, nhiên liệu) và nhóm tất cả các ngành con còn lại. Tác giả thử nghiệm chương trình trên các trang web chứa các tin tức liên quan đến ngành công nghiệp. Kết quả thử nghiệm của các bộ phân loại FSVMs với tập dữ liệu huấn luyện gồm 750 văn bản, tập kiểm tra gồm 250 văn bản thuộc 5 nhóm, sử dụng hàm nhân đa thức d=2, C=20 như sau:

Bảng 1. Kết quả thử nghiệm các bộ phân loại FSVMs.

STT	Bộ phân loại	Precision	Recall	F-score
1.	Điện – Cơ Khí	0.820	0.976	0.891
2.	Điện – Dệt May	0.980	1.000	0.990
3.	Điện – Dầu Khí	0.840	0.933	0.884
4.	Điện – Các Nhóm Khác	0.960	0.906	0.932
5.	Cơ Khí – Dệt May	0.940	0.979	0.959
6.	Cơ Khí – Dầu Khí	0.960	1.000	0.980
7.	Cơ Khí–Các Nhóm Khác	0.940	0.959	0.949
8.	Dệt May – Dầu Khí	0.920	0.979	0.948
9.	Dệt May–Các Nhóm Khác	0.940	0.979	0.959
10.	Dầu Khí – Các Nhóm Khác	0.960	0.980	0.970

Kết quả thử nghiệm của chương trình thu thập và phân loại thông tin trên một số trang báo tin tức có liên quan đến ngành công nghiệp như sau:

Bảng 2.Kết quả thử nghiệm chương trình.

STT	Trang web	Số tin lấy được	Số tin thuộc ngành Công nghiệp	Số tin thuộc ngành công nghiệp được phân loại đúng	% Số tin thuộc ngành công nghiệp được phân loại đúng
1.	Bộ Công thương	18	17	14	82.4
2.	Báo Thương mại – mục Công nghiệp	48	11	9	81.8
3.	Báo Đầu tư	31	19	16	84.2
4.	Thời báo kinh tế Việt Nam	49	36	32	88.9
5.	Phòng Thương mại và Công thương Việt Nam	16	5	4	80.0
6.	Thông tấn xã Việt Nam	39	26	21	80.8
7.	Báo Tuổi trẻ - mục Kinh Tế	135	26	20	76.9
8.	Trung tâm tư vấn công nghiệp TP	66	61	49	80.3
9.	Báo Thanh niên – mục Kinh Tế	37	17	14	82.4
10.	Sài gòn giải phóng – mục Công nghiệp	16	16	14	87.5

Số tin lấy được ở một số trang web là khá lớn vì trên trang chủ có thể có các liên kết đến các trang chứa các chủ đề của trang web, trang này có thể có cùng cách trình bày với trang web mẫu và số lượng là tương đối lớn (Ví dụ trên trang báo Tuổi trẻ là lớn hơn 100 trang). Tuy nhiên các trang chủ đề này là tĩnh và chỉ thu thập được trong lần đầu tiên. Sau mỗi lần thu thập cần lưu lại các liên kết này và chỉ khi có những liên kết mới được cập nhật thì ta mới tiến hành thu thập. Các liên kết này thường là những tin tức mới cập nhật.

Kết quả rút trích sẽ không tốt đối với một số trang web trình bày tin tức theo dạng script như trang chủ của VNExpress, Vietnam Net. Tuy nhiên các trang chủ đề của các trang web này lại trình bày theo cách thông thường. Số các trang web trình bày theo cách này không phổ biến và có thể khắc phục bằng cách chọn nơi bắt đầu thu thập tin tức từ các trang chủ đề thay vì trang chủ.

6.KẾT LUẬN

Trong bài báo này nhóm nghiên cứu đã xây dựng chương trình thu thập và phân loại thông tin trên các trang báo điện tử phục vụ cho việc cung cấp tin tức cho các trang web hành chính của thành phố. Phương pháp rút trích thông tin bằng cách so trùng hai cây đa phân dựa trên phương pháp nhận dạng mẫu làm việc xây dựng các wrapper trở nên đơn giản, cho phép rút trích chính xác vùng thông tin mang nội dung chính trên trang web tin tức giúp việc phân loại được tốt hơn. Phần phân loại thông tin chúng tôi sử dụng phương pháp FSVMs nhằm làm giảm ảnh hưởng của những điểm dữ liệu nhiễu. Ngoài ra phương pháp phân loại đa lớp mờ là một cải tiến, giúp phân loại các điểm dữ liệu không phân loại được theo chiến lược phân loại đa lớp OAO.

USING FSVM CLASSFICATION ALGORITHM FOR EXTRACTING TEXTS ON INTERNET

Vu Thanh Nguyen[(1)], Trang Nhat Quang[(2)]
(1)University of Information Technology
(2) Industry Department HoChiMinh City

ABSTRACT: *This paper is used automatically to extract information and classify texts by SVM method, FSVM method with fuzzy multiclass classification. This research result is used to collect information from admin webpage of local goverment departments, offices for providing citizens, companies about news of HCM city goverment policy, needed information for operating public admin.*

TÀI LIỆU THAM KHẢO

[1]. Nguyễn Thị Kim Ngân, *Phân loại văn bản tiếng Việt bằng phương pháp support vector machines,* Đại học Bách Khoa Hà Nội, Hà Nội, (2004).

[2]. Lê Phú, *Rút trích tự động các khối văn bản mang tin tức chính trên các trang báo*, Đại học Bách Khoa TP.HCM, TPHCM, (2005).

[3]. Johannes F¨urnkranz, *Pairwise Classification as an Ensemble Technique*, Proceedings of the 13th European Conference on Machine Learning, Springer Verlag, pp. 97-110, (2002).

[4]. Johannes F¨urnkranz, *Round Robin Rule Learning*, in Proceedings of the 18th Interneational Conference on Machine Learning (ICML-01), pp. 146-153, (2001).

[5]. Shigeo Abe and Takuya Inoue, *Fuzzy Support Vector Machines for Multiclass Problems, ESANN'2002 proceedings*, pp. 113-118, (2002).

[6]. Shigeo Abe and Takuya Inoue, Fuzzy *Support Vector Machines for Pattern Classification*, In Proceeding of International Joint Conference on Neural Networks (IJCNN '01), volume 2, pp. 1449-1454, (2001).

[7]. Tsui-Feng Hu, *Fuzzy Correlation and Support Vector Learning Approach to Multi-Categorization of Documents*, Master Thesis, Institute of Information Management I-Shou University, Taiwan, (2004).

[8]. T.Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, in Proceedings of ECML-98, 10th European Conference on Machine Learning, number 1398, pp. 137–142, (1998).

[9]. Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo, *RoadRunner: Towards Automatic Data Extraction from Large Web Sites,* The VLDB Journal, pp. 109-118, (2001).

[10].Vladimir N.Vapnik, *The Nature of Statiscal Learning Theory – Second Edition*, Springer, New York, (2000).

[11].Wuu Yang, *Identifying Syntactic Differences Between Two Programs,* Software-Practice & Experience, Volume 21(7), pp. 739-755, (1991).

[12].Xiufeng Jiang, Zhang Yi and Jian Cheng Lv, *Fuzzy SVM with a new fuzzy membership function, Neural Computing and Applications*, Volume 15(3), pp. 268-276, (2006).