PHÂN LOẠI VĂN BẢN VỚI MÁY HỌC VECTOR HỖ TRỢ VÀ CÂY QUYẾT ĐỊNH

Trần Cao Đệ và Phạm Nguyên Khang[1]

ABSTRACT

Text document classification, basically, can be considered as a classification problem. Automatic text document classification is to assign a label to a new document based on the similarity of the document with labeled documents in the training set. Many machine learning and data mining methods have been applied in text document classification such as: Naive Bayes, decision tree, k – Nearest neighbor, neural network,…

Support vector machine (SVM) is an efficient classification algorithm. It has been applied to machine learning and recognition field. However, it is still not efficient in applying to text document classification because, by the nature, this problem often deals with a large feature space. This paper focuses on applying SVM to text document classification and compares the efficiency of the method with the one of decision tree, a traditional classification algorithm. The research illustrates that SVM along with the feature selection based on the singular value decomposition (SVD) is much better than decision tree method.

Keywords: Decision tree, Support vector machine (SVM), text document classification, single value decomposition (SVD)

Title: Text document classification with support vector machine and decision tree

TÓM TẮT

Bài toán phân loại văn bản, thực chất, có thể xem là bài toán phân lớp. Phân loại văn bản tự động là việc gán các nhãn phân loại lên một văn bản mới dựa trên mức độ tương tự của văn bản đó so với các văn bản đã được gán nhãn trong tập huấn luyện. Nhiều kỹ thuật máy học và khai phá dữ liệu đã được áp dụng vào bài toán phân loại văn bản, chẳng hạn: phương pháp quyết định dựa vào Bayes ngây thơ (Naive Bayes), cây quyết định (decision tree), k–láng giềng gần nhất (KNN), mạng nơron (neural network),…

Máy học vectơ hỗ trợ (SVM) là một giải thuật phân lớp có hiệu quả cao và đã được áp dụng nhiều trong lĩnh vực khai phá dữ liệu và nhận dạng. Tuy nhiên SVM chưa được áp dụng một cách có hiệu quả vào phân loại văn bản vì đặc điểm của bài toán phân loại văn bản là không gian đặc trưng thường rất lớn. Bài viết này nghiên cứu máy học vector hỗ trợ (SVM), áp dụng nó vào bài toán phân loại văn bản và so sánh hiệu quả của nó với hiệu quả của giải thuật phân lớp cổ điển, rất phổ biến đó là cây quyết định. Nghiên cứu chỉ ra rằng SVM với cách lựa chọn đặc trưng bằng phương pháp tách giá trị đơn (SVD) cho kết quả tốt hơn so với cây quyết định.

Từ khóa: Cây quyết định, máy học vector hỗ trợ, phân loại văn bản, tách giá trị đơn

1 GIỚI THIỆU BÀI TOÁN PHÂN LOẠI VĂN BẢN

Phân loại văn bản là một bài toán xử lí văn bản cổ điển, đó là ánh xạ một văn bản vào một chủ đề đã biết trong một tập hữu hạn các chủ đề dựa trên ngữ nghĩa của văn bản. Ví dụ một bài viết trong một tờ báo có thể thuộc một (hoặc một vài) chủ

[1] Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

đề nào đó (như *thể thao, sức khỏe, công nghệ thông tin,…*). Việc tự động phân loại văn bản vào một chủ đề nào đó giúp cho việc sắp xếp, lưu trữ và truy vấn tài liệu dễ dàng hơn về sau.

Đặc điểm nổi bật của bài toán này là sự đa dạng của chủ đề văn bản và tính đa chủ đề của văn bản. Tính đa chủ đề của văn bản làm cho sự phân loại chỉ mang tính tương đối và có phần chủ quan, nếu do con người thực hiện, và dễ bị nhập nhằng khi phân loại tự động. Rõ ràng một bài viết về *Giáo dục* cũng có thể xếp vào *Kinh tế* nếu như bài viết bàn về tiền nong đầu tư cho giáo dục và tác động của đầu tư này đến kinh tế - xã hội. Về bản chất, một văn bản là một tập hợp từ ngữ có liên quan với nhau tạo nên nội dung ngữ nghĩa của văn bản. Từ ngữ của một văn bản là đa dạng do tính đa dạng của ngôn ngữ (đồng nghĩa, đa nghĩa, từ vay mượn nước ngoài,…) và số lượng từ cần xét là lớn. Ở đây cần lưu ý rằng, một văn bản có thể có số lượng từ ngữ không nhiều, nhưng số lượng từ ngữ cần xét là rất nhiều vì phải bao hàm tất cả các từ của ngôn ngữ đang xét.

Trên thế giới đã có nhiều công trình nghiên cứu đạt những kết quả khả quan, nhất là đối với phân loại văn bản tiếng Anh. Tuy vậy, các nghiên cứu và ứng dụng đối với văn bản tiếng Việt còn nhiều hạn chế do khó khăn về tách từ và câu. Có thể liệt kê một số công trình nghiên cứu trong nước với các hướng tiếp cận khác nhau cho bài toán phân loại văn bản, bao gồm: phân loại với máy học vectơ hỗ trợ [1], cách tiếp cận sử dụng lý thuyết tập thô [2], cách tiếp cận thống kê hình vị [3], cách tiếp cận sử dụng phương pháp học không giám sát và đánh chỉ mục [4], cách tiếp cận theo luật kết hợp [5]. Theo các kết quả trình bày trong các công trình đó thì những cách tiếp cận nêu trên đều cho kết quả khá tốt. Tuy nhiên khó có thể so sánh các kết quả ở trên với nhau vì tập dữ liệu thực nghiệm của mỗi phương pháp là khác nhau. Bài viết này so sánh hiệu quả của hai cách tiếp cận phân loại văn bản: phân loại với giải thuật cây quyết định và phân loại với máy học vector hỗ trợ kết hợp với phân tích giá trị đơn (SVD).

Theo cả hai cách tiếp cận này, trước hết, văn bản được coi như là một tập hợp các từ. Để thực hiện tách từ chúng tôi đã áp dụng giải thuật MMSEG [6]. Phần tiếp theo sẽ trình bày cụ thể mô hình hóa văn bản trước khi áp dụng phân lớp theo giải thuật cây quyết định và phân lớp theo SVM.

2 MÔ HÌNH HÓA VĂN BẢN

Trên thực tế, để có thể áp dụng một giải thuật tách từ, văn bản cần qua bước tiền xử lí cơ bản: chuẩn hóa dấu, chuẩn hóa "i" và "y", chuẩn hóa font,… Tuy nhiên các bước này sẽ không được đề cập ở đây do giới hạn trang bài viết. Có thể xem văn bản là tập hợp các từ. Khái niệm "từ" ở đây theo nghĩa là một chuỗi kí tự liên tiếp nhau trong văn bản, không nhất thiết phải là một từ có nghĩa trong ngôn ngữ. Việc xác định "từ" hay tách từ sẽ được thực hiện bằng một giải thuật nào đó. Hiện nay phương pháp MMSEG [6] và các cải tiến của nó được áp dụng rộng rãi trong tách từ tiếng Việt. Một số đề xuất tách từ độc lập với ngôn ngữ như phương pháp n-gram; chẳng hạn trong tiếng Việt cứ lấy hai tiếng liên tiếp đứng cạnh nhau trong văn bản làm "2-gram". Như vậy một "2-gram" không nhất thiết phải là một từ đúng trong tiếng Việt. Trong nghiên cứu này, chúng tôi dùng giải thuật MMSEG để tách từ tiếng Việt. Giải thuật này có nguồn gốc là để tách tiếng Trung Quốc [7] với độ chính xác 99%. Nhiều nghiên cứu đã áp dụng giải thuật MMSEG vào tách từ tiếng Việt nhưng chưa thấy có báo cáo chính thức nào về kết quả tách từ. Tuy nhiên, trong nghiên cứu của chúng tôi, MMSEG có thể áp dụng vào bài toán phân loại văn bản vì: MMSEG tách từ với độ chính xác khá cao trên 95%; tỷ lệ sai sót trong tách từ khoảng 5% không ảnh hưởng lớn đến kết quả phân loại.

Sau khi tách từ, văn bản được xem như là một tập hợp các "từ". Chữ từ trong dấu ngoặc vì nó là từ sinh ra bởi giải thuật tách từ, nó không nhất thiết phải có nghĩa trong ngôn ngữ. Với giải thuật MMSEG thì các "từ" được tách đều có nghĩa (có trong từ điển), tuy nhiên nó không nhất thiết phải đúng hoàn toàn trong ngữ cảnh của văn bản (ngữ nghĩa). Hình 1 cho ví dụ về một đoạn văn bản được tách theo giải thuật MMSEG.

Ai/ cũng/ biết/ không gian/ có thể/ tác động/ đến/ con người.

Mặt trời/ gây nên/ nhiều/ vấn đề/ nơi/ một số/ người/ nhạy cảm/ trước/ những/ đổi thay/ của/ thời tiết.

Bên cạnh/ việc/ gây nên/ biến động/ thuỷ triều/ mặt trăng/ còn/ là/ nguyên nhân/ của/ hiện tượng/ mộng du/ bước đi/ trong khi/ ngủ.

Dường như/ ai/ cũng/ nghe nói/ địa cầu/ chúng ta/ có thể/ là/ nơi/ đổ bộ/ của/ các/ thiên thạch/ vào/ một/ ngày/ vô định/ nào đó.

Hình 1: Ví dụ về tách từ với giải thuật MMSEG

Rõ ràng rằng, các từ trong văn bản có mức độ quan trọng khác nhau đối với văn bản và cả trong phân loại văn bản. Một số từ như từ nối, từ chỉ số lượng ("và", "các", "những, "mỗi",…) không mang tính phân biệt trong khi phân loại. Ngoài ra, còn có rất nhiều từ khác cũng không có giá trị phân loại ví dụ như từ xuất hiện hầu khắp các văn bản hay dùng không phổ biến trong văn bản, những từ gọi là stopword này cần được loại bỏ. Có nhiều cách loại bỏ stopword, chẳng hạn dùng 1 danh sách các stopword hoặc loại bỏ theo tần suất xuất hiện của từ (chỉ số TF*IDF). Trong thực nghiệm chúng tôi dùng một danh sách stopword kết hợp với việc loại bỏ các từ có chỉ số TF*IDF thấp. Chỉ số TF*IDF thấp tức là từ xuất hiện hầu khắp các băn bản hoặc từ rất ít xuất hiện.

Sau khi loại bỏ các stopword, văn bản có thể xem như là một tập hợp các đặc trưng, đó là tập hợp các từ "quan trọng" còn lại để biểu diễn văn bản. Việc phân loại văn bản sẽ dựa trên các đặc trưng này. Tuy nhiên, có thể thấy rằng, số đặc trưng của một văn bản là lớn và không gian các đặc trưng (tất cả đặc trưng) của tất cả các văn bản đang xem xét là rất lớn, về nguyên tắc, nó bao gồm tất cả các từ trong một ngôn ngữ. Chính vì vậy, phân loại dựa trên các đặc trưng này cần phải có cách xử lí, lựa chọn đặc trưng nhằm rút ngắn số chiều của không gian đặc trưng. Trên thực tế, người ta không thể xét tất cả các từ của ngôn ngữ mà là dùng tập hợp các từ được rút ra từ một tập (đủ lớn) các văn bản đang xét (gọi là tập ngữ liệu).

Kế đến, mỗi văn bản d_i trong tập ngữ liệu đang xét sẽ được mô hình hóa như là một vector trọng số của các đặc trưng, $d_i(w_{i1},\ldots,w_{im})$. Trong bài viết này, trọng số của một từ được tính theo tần suất xuất hiện của từ trong văn bản (TF) và tần suất nghịch đảo của từ trong tập ngữ liệu (IDF).

$$w_{ij} = TF_{ij} * \log\left(\frac{N}{DF_j}\right) \qquad (1)$$

- TF_{ij} là số lần xuất hiện của từ thứ j trong văn bản thứ i.
- DF_j là tổng số văn bản có chứa từ thứ j trong tập ngữ liệu.
- N là tổng số văn bản trong tập ngữ liệu.

3 PHÂN LOẠI VĂN BẢN THEO PHƯƠNG PHÁP CÂY QUYẾT ĐỊNH

Phương pháp cây quyết định [8] có thể áp dụng vào bài toán phân loại văn bản. Dựa vào tập các văn bản huấn luyện (sau này gọi tắt là *tập huấn luyện*), xây dựng một cây quyết định. Cây quyết định có dạng là cây nhị phân, mỗi nút trong tương ứng với việc phân hoạch tập văn bản dựa trên một thuộc tính nào đó (một từ). Việc xây dựng cây quyết định phụ thuộc vào việc lựa chọn thuộc tính để phân hoạch. Theo [8], chúng tôi lựa chọn thuộc tính phân hoạch dựa trên độ lợi thông tin (information gain) lớn nhất, đó là hiệu giữa độ hỗn loạn thông tin trước và sau phân hoạch với thuộc tính đó. Độ lợi thông tin được tính toán dựa vào độ hỗn loạn thông tin (Entropy) theo công thức (2). Giả sử tập huấn luyện S chứa các văn bản thuộc k chủ đề, thì độ hỗn loạn thông tin của tập S là:

$$Entropy(S) = \sum_{i=1}^{k} (-p_i \log_2 p_i) \qquad (2)$$

Trong đó p_i là xác suất để một phần tử (1 văn bản) thuộc vào chủ đề thứ i. p_i chính là tần suất xuất hiện một văn bản thuộc chủ đề thứ i trong tập S.

Độ lợi thông tin khi dùng thuộc tính *a* phân hoạch tập S thành các tập con tùy theo giá trị của a (kí hiệu *Values(a)* trong công thức) là :

$$Gain(S,a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (3)$$

3.1 Giải thuật xây dựng cây quyết định

Đầu vào :

- Tập **M** chứa tất cả các văn bản huấn luyện đã mô hình hóa thành các vector $d_i(w_{i1},\ldots,w_{im})$
- Tập **A** chứa tất cả các từ trong tập huấn luyện M (sau khi loại stopword)
- Một tập chủ đề C.

Đầu ra : Cây quyết định dạng nhị phân cho việc phân loại theo tập chủ đề C.

Giải thuật (tham khảo [9]):

- Bắt đầu: nút gốc chứa tất cả văn bản huấn luyện.
- Nếu dữ liệu tại nút chỉ thuộc 1 chủ đề (1 lớp) thì nút là nút lá và được gán nhãn là chủ đề đó.
- Nếu một nút chứa dữ liệu không thuần nhất (thuộc các lớp khác nhau) thì lựa chọn thuộc tính phân hoạch với độ lợi thông tin lớn nhất (giả sử thuộc tính là a với giá trị y, y gọi là *giá trị phân tách*); phân chia nút này một cách đệ qui làm

hai tập M1, M2; M1 chứa các văn bản chứa a nhưng giá trị thuộc tính nhỏ hơn y, M2 chứa các văn bản chứa a và giá trị thuộc tính lớn hơn bằng y.

Giải thuật dừng khi tất cả các nút lá đã được gán nhãn. Trong ứng dụng, người ta có thể không tiến hành phân hoạch nút đến khi dữ liệu đồng nhất (chỉ thuộc một lớp) mà người ta dừng phân hoạch khi số phần tử tại nút còn ít hơn một số lượng nào đó và gán nhãn nút theo luật bình chọn số đông của các phần tử chứa tại nút. Điều này nhằm cải tiến tốc độ xây dựng cây và tránh được tình trạng học vẹt.

3.2 Đánh giá một giải thuật máy học

Một số chỉ số thông dụng được dùng để đánh giá một giải thuật máy học, hay cụ thể là để đánh giá một bộ phân loại hai lớp tạm gọi là *dương* và *âm*:

- Số đúng dương (TP- True positive): số phần tử dương được phân loại dương
- Số sai âm (FN - False negative): số phần tử dương được phân loại âm
- Số đúng âm (TN- True negative): số phần tử âm được phân loại âm
- Số sai dương (FP - False positive): số phần tử âm được phân loại dương
- Độ chính xác (Precision) = TP/(TP + FP)
- Độ bao phủ (Recall) = TP/(TP + FN)
- Độ đo F1= 2*Precision*Recall/(Precision + Recall)

Các chỉ số này sẽ được dùng để đánh giá hiệu quả cây quyết định và máy học SVM về sau, trong phần thực nghiệm.

3.3 Xén tỉa cây quyết định

Cây quyết định vừa được xây dựng thường là lớn, không mang tính tổng quát mà mang tính « học vẹt » theo tập huấn luyện. Để tăng tính tổng quát của cây, làm cho cây thích ứng với các mẫu dữ liệu mới, chưa được huấn luyện, người ta cắt bớt các nhánh cây hay còn gọi là xén tỉa cây với một tập kiểm chứng độc lập với tập huấn luyện. Đây gọi là việc xén tỉa sau, giải thuật chi tiết như sau:

- Với mỗi nút trong (không phải nút lá), cắt bỏ các nhánh phân hoạch nút biến nút đó thành nút lá và gán nhãn theo luật bình chọn số đông.
- Dùng tập kiểm chứng độc lập để kiểm tra độ chính xác (precision) của cây mới sau mỗi thao tác xén.
- Nếu sau khi xén, độ chính xác của cây được tăng lên thì giữ nguyên việc xén và tiếp tục quá trình xén cho các nút trong còn lại; ngược lại thì trả lại hiện trạng ban đầu (không thực hiện việc xén tỉa).

Thuật toán dừng khi tất cả các nút đã được xem xét để xén tỉa.

Việc thực hiện xén tỉa cây như vậy có độ phức tạp thời gian lớn do phải dùng tập kiểm chứng để ước lượng lỗi sinh ra khi xén tỉa. Trong thực hành chúng tôi áp dụng giải thuật xây dựng cây với giải pháp bình chọn trên số đông, nếu số đông vượt ngưỡng đặt ra thì dừng việc phân hoạch. Như vậy, chúng tôi không thực hiện thao tác xén tỉa cây.

3.4 Thực hiện phân loại 1 văn bản mới

Các cây quyết định giờ đã được xây dựng xong và sẵn sàng để dùng cho phân loại văn bản. Văn bản mới (cần được phân loại) được coi như là một tập hợp các đặc trưng (các từ). Ta sẽ tiến hành duyệt cây quyết định để gán nhãn phân loại chủ đề cho văn bản đó. Việc duyệt cây quyết định hơi giống với duyệt và tìm kiếm trên cây nhị phân tìm kiếm:

- Nếu từ thuộc văn bản và giá trị của từ nhỏ hơn giá trị phân tách tại nút, hoặc từ không thuộc văn bản thì ta sẽ duyệt tiếp cây con trái của cây quyết định.
- Nếu từ thuộc văn bản và giá trị của từ lớn hơn giá trị phân tách tại nút thì ta sẽ duyệt cây con phải của cây quyết định.
- Quá trình này dừng khi gặp nút hiện tại là nút lá, gán nhãn cho văn bản là nhãn của nút lá đó.

4 PHÂN LOẠI VĂN BẢN VỚI MÁY HỌC VECTOR HỖ TRỢ

Gần đây phương pháp máy học vector hỗ trợ đã được áp dụng vào bài toán phân loại văn bản và đã cho thấy kết quả khả quan [1,12]. Tuy nhiên, như đã nói, bài toán phân loại văn bản có các đặc trưng là từ nên không gian đặc trưng là rất lớn, bao gồm mọi từ của ngôn ngữ hoặc trong tập ngữ liệu. Số chiều của không gian đặc trưng lớn làm gia tăng nhiễu, đó là một trở ngại chính trong việc áp dụng SVM vào phân loại văn bản. Để áp dụng có hiệu quả SVM, người ta cần tìm cách rút ngắn số chiều của không gian đặc trưng. Trong nghiên cứu [1], các tác giả đã đề xuất dùng lượng tin tương hỗ để loại bỏ bớt các đặc trưng. Trong nghiên cứu này chúng tôi dùng kỹ thuật tích giá trị đơn (SVD) để rút ngắn số chiều không gian đặc trưng.

4.1 Phân tích giá trị đơn (SVD)

Phân tích giá trị đơn là phân tích toán học nền tảng trong kỹ thuật chỉ mục ngữ nghĩa tiềm ẩn (LSI-Latent Semantic Indexing) đã được dùng rộng rãi trong tìm kiếm và thu hồi thông tin dạng văn bản. Ý tưởng chính của giải thuật [10,11] như sau:

Cho ma trận **A** (kích thước mxn), ma trận **A** luôn luôn phân tích được thành tích của ba ma trận theo dạng: $\mathbf{A} = \mathbf{U}\sum\mathbf{V}^{\mathbf{T}}$, trong đó:

- **U** là ma trận trực giao mxm có các cột là các vectơ đơn bên trái của **A**.
- $\sum$ là ma trận mxn có đường chéo chứa các giá trị đơn, không âm có thứ tự giảm dần:
- $\delta_1 \geq \delta_2 \geq \ldots \geq \delta_{\min(m,n)} \geq 0$
- **V** là ma trận trực giao nxn có các cột là các vectơ đơn bên phải của **A**.

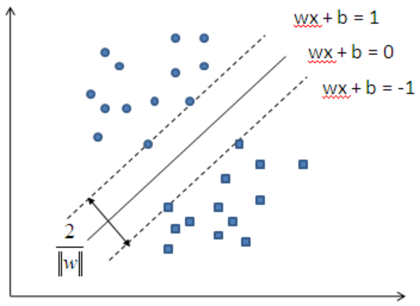
Hạng của ma trận **A** là số các số khác 0 trên đường chéo chính của ma trận $\sum$**.** Thông thường **A** là một ma trận thưa có kích thước lớn. Để giảm số chiều của ma trận người ta thường tìm cách xấp xỉ ma trận **A** (có hạng r) bằng một ma trận $\mathbf{A_k}$ có hạng là k nhỏ hơn r rất nhiều. Ma trận xấp xỉ của **A** theo kỹ thuật này chính là: $\mathbf{A_k} = \mathbf{U_k}\sum_k\mathbf{V_k}^{\mathbf{T}}$**,** trong đó

- $\mathbf{U_k}$ là ma trận trực giao mxk có các cột là k cột đầu của ma trận U.
- $\sum_k$ là ma trận đường chéo kxk chứa k phần tử đầu tiên $\delta_1, \delta_2, \ldots, \delta_k$ trên đường chéo chính.
- $\mathbf{V_k}$ là ma trận trực giao nxk có các cột là k cột đầu của ma trận **V**.

Việc xấp xỉ này có thể xem như chuyển không gian đang xét (r chiều) về không gian k chiều, với k nhỏ hơn rất nhiều so với r. Về mặt thực hành việc cắt ma trận A về số chiều k còn loại bỏ nhiễu và tăng cường các mối liên kết ngữ nghĩa tiềm ẩn giữa các từ trong tập văn bản. Chúng tôi sẽ áp dụng kỹ thuật xấp xỉ này để rút ngắn số chiều của không gian đặc trưng. Khởi đầu, mỗi văn bản được mô hình hóa thành một vectơ cột trong không gian xác định bởi A_{mxn}. Sau khi cắt A_{mxn} về A_k, các tất cả các vectơ đang xét đều được chiếu lên không gian A_k để có số chiều k theo công thức:

$$\mathrm{Proj}(x) = x^T \mathbf{U_k} \mathbf{\sum_k}^{-1} \tag{4}$$

4.2 Máy học véctơ hỗ trợ



Hình 2: Ví dụ siêu phẳng với lề cực đại trong R^2

Máy học véctơ hỗ trợ (SVM) là một giải thuật máy học dựa trên lý thuyết học thống kê do Vapnik và Chervonenkis xây dựng [13]. Bài toán cơ bản của SVM là bài toán phân loại hai lớp: Cho trước n điểm trong không gian d chiều (mỗi điểm thuộc vào một lớp kí hiệu là +1 hoặc –1, mục đích của giải thuật SVM là tìm một siêu phẳng (hyperplane) phân hoạch tối ưu cho phép chia các điểm này thành hai phần sao cho các điểm cùng một lớp nằm về một phía với siêu phẳng này. Hình 2 cho một minh họa phân lớp với SVM trong mặt phẳng.

Xét tập dữ liệu mẫu có thể tách rời tuyến tính $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$ với $x_i \in R^d$ và $y_i \in \{\pm 1\}$. Siêu phẳng tối ưu phân tập dữ liệu này thành hai lớp là siêu phẳng có thể tách rời dữ liệu thành hai lớp riêng biệt với lề (margin) lớn nhất. Tức là, cần tìm siêu phẳng H: y = w.x + b = 0 và hai siêu phẳng H1, H2 hỗ trợ song song với H và có cùng khoảng cách đến H. Với điều kiện không có phần tử nào của tập mẫu nằm giữa H1 và H2, khi đó:

w.x + b >= +1 với y = +1

w.x + b >= -1 với y = -1

Kết hợp hai điều kiện trên ta có y(w.x + b) >= 1.

Khoảng cách của siêu phẳng H1 và H2 đến H là $\|w\|$. Ta cần tìm siêu phẳng H với lề lớn nhất, tức là giải bài toán tối ưu tìm $\min_{w,b}\|w\|$ với ràng buộc y(w.x + b) >= 1. Người ta có thể chuyển bài toán sang bài toán tương đương nhưng dễ giải hơn là $\min_{w,b}\frac{1}{2}\|w\|^2$ với ràng buộc y(w.x + b) >= 1. Lời giải cho bài toán tối ưu này là cực tiểu hóa hàm Lagrange:

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(<w,x_i> + b) - 1\right) \tag{5}$$

Trong đó α là các hệ số Lagrange, α≥0. Sau đó người ta chuyển thành bài toán đối ngẫu là cực đại hóa hàm W(α):

$$\max_{\alpha} W(\alpha) = \max_{\alpha}\left(\min_{w,b} L(w,b,\alpha)\right) \quad (6)$$

Từ đó giải để tìm được các giá trị tối ưu cho w,b và α. Về sau, việc phân loại một mẫu mới chỉ là việc kiểm tra hàm dấu sign(wx +b).

Lời giải tìm siêu phẳng tối ưu trên có thể mở rộng trong trường hợp dữ liệu không thể tách rời tuyến tính [11] bằng cách ánh xạ dữ liệu vào một không gian có số chiều lớn hơn bằng cách sử dụng một hàm nhân K (kernel). Một số hàm nhân thường dùng được cho trong bảng 1.

Bảng 1: Một số hàm nhân thường dùng

Kiểu hàm nhân	Công thức
Linear kernel	K(x, y) = x . y
Polynomial kernel	$K(x, y) = (x .y + 1)^d$
Radial basis function (Gaussian) kernel	$K(x,y)=e^{\frac{-\lvert x-y\rvert^2}{2\sigma^2}}$
Hyperbolic tangent kernel	K(x, y) = tanh (a.x.y - b)

Ở đây chúng tôi không có ý định đi sâu vào chi tiết giải bài toán tìm siêu phẳng này, độc giả quan tâm có thể tìm lời giải trong công trình của Vapnik [13]. Chúng tôi sử dụng phần mềm Weka [14] để thực hiện các tính toán phân lớp và kiểm tra phương pháp đề xuất.

5 KẾT QUẢ THỰC NGHIỆM

Trong thực nghiệm, có 7842 văn bản thuộc 10 chủ đề khác nhau đã được tập hợp dùng để xây dựng máy học và kiểm chứng hiệu quả. Các văn bản được sưu tập từ các trang báo điện tử phổ biến bằng tiếng việt như *vnexpress.net, vietnamnet.vn, dantri.com.vn*. Sau khi tách từ và loại bỏ stopword, số từ còn lại là 14275 từ. Sau khi mô hình hóa, mỗi văn bản là một vector trọng số các từ, trong đó các trọng số là chỉ số TF*IDF như đã trình bày. Như vậy tập ngữ liệu được mô hình hóa như là một ma trận chứa TF*IDF của các từ và có kích thước 14275 x 7842 phần tử. Bảng 2 cho số liệu thống kê số văn bản thuộc mỗi chủ đề. Trong mỗi chủ đề, 500 văn bản được chọn một cách ngẫu nhiên để huấn luyện, tức là xây dựng cây quyết định hoặc huấn luyện máy học SVM. Số văn bản còn lại để kiểm chứng độc lập. Để tiện gọi tên hai tập này được đặt tên là *tập huấn luyện* và *tập kiểm chứng độc lập*.

Việc đánh giá dựa vào các chỉ số độ chính xác (Precision), độ bao phủ (Recall) và F1. Kết quả kiểm chứng các cây quyết định với *tập kiểm chứng độc lập* được cho trong bảng 3. Các chỉ số kiểm chứng nói trên được cho trong bảng 5 và so sánh với kết quả kiểm chứng với máy học SVM.

Bảng 2: 10 chủ đề và số lượng mẫu dùng trong thực nghiệm

Tên lớp	Số mẫu huấn luyện	Số mẫu kiểm chứng	Tổng số mẫu (văn bản)
CNTT	500	286	786
ĐTVT	500	282	782
Giáo dục	500	299	799
Ẩm thực	500	291	791
Bất động sản	500	265	765
Khoa học	500	282	782
Kinh tế	500	291	791
Y học	500	287	787
Thể thao	500	288	788
Giải trí	500	271	771
Tổng cộng	*5000*	*2842*	*7842*

Bảng 3: Kết quả kiểm chứng bộ phân lớp bằng cây quyết định

Tên lớp	Mã lớp	1	2	3	4	5	6	7	8	9	10
CNTT	1	**250**	6	8	3	4	3	2	3	3	4
ĐTVT	2	12	**227**	6	5	5	4	7	5	6	5
Giáo dục	7	9	10	**231**	7	9	6	5	12	5	5
Ẩm thực	4	2	10	2	**253**	6	3	2	3	7	3
Bất động sản	5	2	5	5	5	**225**	7	4	2	5	5
Khoa học	6	4	3	8	8	7	**226**	5	7	8	6
Kinh tế	7	5	7	4	5	5	7	**243**	7	5	3
Y học	8	4	5	5	6	5	3	4	**245**	6	4
Thể thao	9	1	0	2	1	3	3	3	1	**273**	1
Giải trí	10	7	4	6	9	7	6	7	6	6	**213**

Để huấn luyện máy học SVM, tập ngữ liệu đang xét (đã được mô hình hóa như ma trận $A_{14275x7842}$) sẽ được phân tích giá trị đơn và rút ngắn số chiều về k=200. Tất cả các vector tương ứng với 7842 văn bản đều được chiếu lên không gian A_{200} bằng công thức (4). Máy học SVM được huấn luyện bằng *tập huấn luyện* đã được dùng để xây dựng cây quyết định. *Tập kiểm chứng độc lập* một lần nữa được dùng để kiểm chứng hiệu quả máy học SVM. Kết quả kiểm chứng được cho trong bảng 4 và các chỉ số đánh giá được cho trong bảng 5 để so sánh với phân lớp theo cây quyết định. Máy học SVM trong thực nghiệm này là máy học với hàm nhân (kernel) RBF, với tham số C bằng 12 và Gama bằng 2^{-8}. Thực nghiệm cũng đã được làm với một số tham số khác của C và Gama, bộ tham số nói trên được chọn bằng phương pháp thử và sai. Do tham số Gama nhỏ nên có thể dùng máy học SVM với hàm nhân tuyến tính (linear kernel). Kết quả thực nghiệm trên cùng bộ dữ liệu với hàm nhân tuyến tính (C=10 và eps=0.01) cho kết quả tốt hơn trên hàm nhân RBF một ít, nhưng không có khác biệt nhiều. Vì vậy có thể dùng hàm nhân RBF hay hàm nhân tuyến tính với các tham số như vừa nêu.

Bảng 4: Kết quả kiểm chứng bộ phân lớp bằng máy học SVM

Tên lớp	Mã lớp	1	2	3	4	5	6	7	8	9	10
CNTT	1	**265**	5	3	1	2	1	3	3	2	1
ĐTVT	2	11	**246**	3	2	5	4	3	3	2	3
Giáo dục	3	3	5	**276**	2	3	4	1	3	1	1
Ẩm thực	4	1	1	3	**273**	1	3	4	2	2	1
Bất động sản	5	1	3	2	1	**249**	2	3	2	1	1
Khoa học	6	4	3	7	4	1	**251**	3	4	2	3
Kinh tế	7	4	7	2	5	5	3	**254**	3	4	4
Y học	8	3	3	5	3	1	3	4	**258**	4	3
Thể thao	9	2	3	2	2	2	3	1	2	**269**	2
Giải trí	10	2	3	3	0	2	5	3	3	6	**244**

Từ số liệu kiểm chứng chi tiết trong bảng 3 và 4 có thể tính toán các chỉ số đánh giá: Precision, Recall và F1 như trong bảng 5.

Bảng 5: So sánh hiệu quả phân loại văn bản với cây quyết định và với máy học SVM

Tên lớp	Cây quyết định			Máy học SVM		
	Precision	Recall	F1	Precision	Recall	F1
CNTT	84.5%	87.4%	85.9%	89.5%	92.7%	91.1%
ĐTVT	81.9%	80.5%	81.2%	88.2%	87.2%	87.7%
Giáo dục	83.4%	77.3%	80.2%	90.2%	92.3%	91.2%
Ẩm thực	83.8%	86.9%	85.3%	93.2%	93.8%	93.5%
Bất động sản	81.5%	84.9%	83.2%	91.9%	94.0%	92.9%
Khoa học	84.3%	80.1%	82.2%	90.0%	89.0%	89.5%
Kinh tế	86.2%	83.5%	84.8%	91.0%	87.3%	89.1%
Y học	84.9%	89.9%	87.3%	91.2%	89.9%	90.5%
Thể thao	84.3%	94.8%	89.2%	91.8%	93.4%	92.6%
Giải trí	85.5%	78.6%	81.9%	92.8%	90.0%	91.4%
		Trung bình	**84.1%**		**Trung bình**	**91.0%**

Như vậy với máy học SVM kết hợp với phân tích giá trị đơn để rút ngắn số chiều của không gian đặc trưng sẽ cho kết quả phân loại văn bản tốt hơn là phương pháp cây quyết định. Chúng tôi cũng đã thử nghiệm dùng SVM với không gian đặc trưng ban đầu, chưa rút gọn số chiều. Kết quả cho thấy nếu dùng SVM với không gian đặc trưng nguyên thủy thì kết quả thấp (chỉ số F1 trung bình thu được trên thực nghiệm là 85.2%), chỉ gần tương đương với hiệu quả của cây quyết định như đã trình bày trong bảng 5. Việc phân tích giá trị đơn và rút ngắn số chiều không gian đặc trưng đã góp phần tăng độ chính xác của máy học SVM do đã loại bỏ bớt nhiễu và tăng cường mối liên hệ ngữ nghĩa giữa các từ trong không gian đặc trưng.

6 KẾT LUẬN

Trong bài viết này chúng tôi đã trình bày phương pháp phân loại văn bản dựa trên máy học SVM. Đóng góp của chúng tôi là đã đề xuất dùng kỹ thuật phân tích giá trị đơn (SVD) để rút ngắn số chiều của không gian đặc trưng. Chúng tôi đã kiểm chứng đề xuất này trên 2842 tập tin độc lập tập huấn luyện thuộc 10 chủ đề với máy học SVM cài đặt trong phần mềm Weka. Kết quả cho thấy rằng việc dùng SVD để phân tích và rút gọn số chiều của không gian đặc trưng đã nâng cao hiệu quả phân lớp SVM. Thực nghiệm đã so sánh kết quả phân lớp với SVM với kết quả phân lớp với cây quyết định, qua đó cho thấy rằng SVM thực sự tốt hơn cây quyết định khi số chiều không gian đặc trưng được rút gọn một cách hợp lí. Việc rút gọn đặc trưng còn giúp cho không gian lưu trữ giảm xuống và thời gian thực hiện phân lớp nhanh hơn vì số chiều của không gian đặc trưng nhỏ hơn nhiều so với số chiều của không gian đặc trưng ban đầu.

Các kiểm chứng thực nghiệm dựa trên tập hợp các mẫu độc lập với các mẫu dùng để xây dựng máy học cho thấy rằng hiệu quả của máy học SVM trong bài toán phân loại văn bản là ổn định, không phải là học vẹt. Việc phân tích giá trị đơn để rút gọn số chiều của không gian đặc trưng là hoàn toàn thích hợp cho bài toán phân loại văn bản, một bài toán mà không gian đặc trưng lớn, có nhiều nhiễu.

Kết quả nghiên cứu này có thể áp dụng vào các bài toán phân lớp và nhận dạng khác như nhận dạng chữ viết tay, nhận dạng hình ảnh (mặt người, vân tay). Các bài toán này về bản chất không khác so với bài toán phân loại văn bản vì qui trình xử lí, phương pháp xử lí là tương tự nhau: rút trích đặc trưng, lựa chọn đặc trưng, máy học và phân lớp. Chúng tôi sẽ tiếp tục nghiên cứu việc lựa chọn đặc trưng bằng phân tích giá trị đơn SVD và hi vọng sẽ cải tiến hiệu quả nhận dạng ảnh nói chung, nhận dạng chữ viết tay nói riêng.

TÀI LIỆU THAM KHẢO

1. Nguyễn Linh Giang, Nguyễn Mạnh Hiển, Phân loại văn bản tiếng Việt với bộ phân loại vectơ hỗ trợ SVM. Tạp chí CNTT&TT, Tháng 6 năm 2006.
2. Nguyễn Ngọc Bình, "Dùng lý thuyết tập thô và các kỹ thuật khác để phân loại, phân cụm văn bản tiếng Việt", Kỷ yếu hội thảo ICT.rda'04. Hà nội 2004.
3. Nguyễn Linh Giang, Nguyễn Duy Hải, "Mô hình thống kê hình vị tiếng Việt và ứng dụng", Chuyên san "Các công trình nghiên cứu, triển khai Công nghệ Thông tin và Viễn thông, Tạp chí Bưu chính Viễn thông, số 1, tháng 7-1999, trang 61-67. 1999
4. Huỳnh Quyết Thắng, Đinh Thị Thu Phương, "Tiếp cận phương pháp học không giám sát trong học có giám sát với bài toán phân lớp văn bản tiếng Việt và đề xuất cải tiến công thức tính độ liên quan giữa hai văn bản trong mô hình vectơ", Kỷ yếu Hội thảo ICT.rda'04, trang 251-261, Hà Nội 2005.
5. Đỗ Phúc, Nghiên cứu ứng dụng tập phổ biến và luật kết hợp vào bài toán phân loại văn bản tiếng Việt có xem xét ngữ nghĩa, Tạp chí phát triển KH&CN, tập 9, số 2, pp. 23-32, năm 2006
6. Chih-Hao Tsai, MMSEG: A Word Identification System for Mandarin Chinese Text Based on Two Variants of the Maximum Matching Algorithm. http://technology.chtsai.org/MMSEG/, 2000.
7. Keh-Jiann Chen, Shing-Huan Liu, Word Identification for Mandarin Chinese sentences, proceedings of Coling 92, Nantes, pp. 23-28, 1992.
8. Quinlan J., C4.5: Programs for Machine Learning, Morgan Kaufman Publishers, 1993.
9. Đỗ Thanh Nghị, Khai mỏ dữ liệu – minh họa bằng ngôn ngữ R (chương 4), NXB Đại học Cần Thơ, 2010.

10. M.W. Berry, Z. Drmac, E.R. Jessup; Matrices, Vectơ Spaces and Information Retrieval; Society for Industrial and Applied Mathematics, Vol. 41, No. 2, 1999. pp. 335-362.
11. T. Letsche M. Berry; Large-scale Information Retrieval with Latent Semantic Analysis. SIGIR 2001, pp. 19-25
12. Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In European Conference on Machine Learning (ECML), 1998.
13. V.Vapnik. The Nature of Statistical Learning Theory. Springer, NewYork, 1995.
14. Weka, http://www.cs.waikato.ac.nz/ml/weka/