XỬ LÝ Ý KIẾN PHẢN HỒI CỦA NGƯỜI HỌC DỰA TRÊN PHƯƠNG PHÁP PHÂN LOẠI VĂN BẢN

Phạm Thị Kim Ngoan[a*], Nguyễn Hải Triều[a]

[a]Khoa Công nghệ Thông tin, Trường Đại học Nha Trang, Khánh Hòa, Việt Nam
[*]Tác giả liên hệ: Email: ngoanptk@ntu.edu.vn

Lịch sử bài báo
Nhận ngày 27 tháng 02 năm 2020
Chỉnh sửa ngày 29 tháng 4 năm 2020 | Chấp nhận đăng ngày 15 tháng 6 năm 2020

Tóm tắt

Đảm bảo chất lượng đào tạo đang nhận được nhiều sự quan tâm của các cơ sở đào tạo đại học. Người học đóng vai trò quan trọng trong việc đảm bảo chất lượng đào tạo. Với mục tiêu hiểu được các phản hồi của người học về các hoạt động đào tạo tại Trường Đại học Nha Trang (ĐHNT) nhằm góp phần nâng cao chất lượng đào tạo của Nhà trường, chúng tôi đề xuất xử lý các ý kiến phản hồi của người học thông qua việc tự động phân loại và gán nhãn các ý kiến phản hồi của người học. Việc phân loại và dự đoán các nhãn được thực hiện dựa trên phương pháp Support Vector Machine (SVM) và Naive Bayes Classifier (NBC). Thực nghiệm cho kết quả khả quan trên tập dữ liệu ý kiến của người học trường ĐHNT với phương pháp SVM và NBC tương ứng là 92.13% và 90.10%.

Từ khóa: Naive Bayesian Classification (NBC); Phân loại văn bản; Support Vector Machine (SVM); Ý kiến người học.

DOI: http://dx.doi.org/10.37569/DalatUniversity.10.3.667(2020)

Loại bài báo: Bài báo nghiên cứu gốc có bình duyệt

HANDLING OF STUDENT FEEDBACK BASED ON TEXT CLASSIFICATION

Pham Thi Kim Ngoan[a*], Nguyen Hai Trieu[a]

[a]The Information Technology Faculty, Nha Trang University, Khanhhoa, Vietnam
[*]Corresponding author: Email: ngoanptk@ntu.edu.vn

Article history
Received: February 27th, 2020
Received in revised form: April 29th, 2020 | Accepted: June 15th, 2020

Abstract

Ensuring quality training has been receiving a lot of attention from university training establishments. Learners play an important role in quality assurance in training and education. To understand the meaning of student feedback on training activities at Nha Trang University (NTU) and to improve the university's training, we propose to handle student feedback through automatic feedback classification and labeling. The classification and prediction of labels are based on the Support Vector Machine (SVM) and Naive Bayes Classifier (NBC) methods. Experiments with the SVM and NBC methods show positive results, 92.13% and 90.10%, respectively, for the data set of student reviews at Nha Trang University.

Keywords: Learner's feedback; Naive Bayesian Classification; Support Vector Machine; Text Classification.

DOI: http://dx.doi.org/10.37569/DalatUniversity.10.3.667(2020)

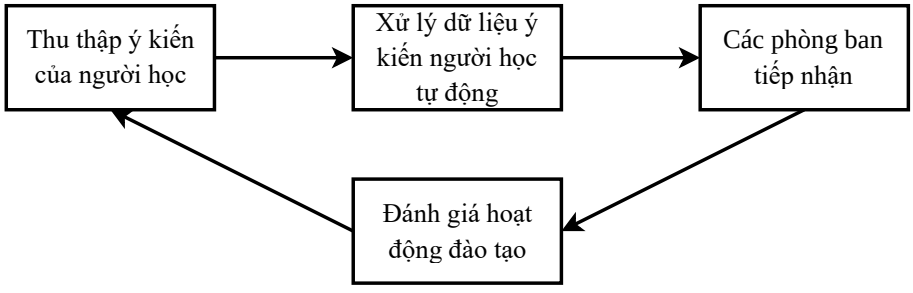
Article type: (peer-reviewed) Full-length research article

1. GIỚI THIỆU

Ở các nước phát triển, việc lấy ý kiến phản hồi của người học đã có từ lâu và là một hoạt động phổ biến. Tại Đại học Harvard (Hoa Kỳ) việc thu thập phản hồi của sinh viên diễn ra thường xuyên vào đầu học kỳ, giữa kỳ, và cuối học kỳ (Harvard University, n.d). Đại học Malta (Cộng hòa Malta) thiết kế các mẫu đánh giá về bài học, chương trình học để thu nhận các ý kiến từ người học định kỳ cuối bài, cuối chương trình (L-Università ta' Malta, 2020). Các trường đại học thông qua phản hồi từ người học nhằm thu nhận những thông tin về chất lượng giảng dạy và học tập tại Trường. Ở Việt Nam, người học đóng vai trò quan trọng trong việc đảm bảo chất lượng đào tạo. Hầu hết các trường đại học đều có các kênh để lấy ý kiến phản hồi từ người học về quá trình đào tạo, các hoạt động giảng dạy của giảng viên. Tuy nhiên, mỗi trường có cách lấy ý kiến và xử lý số liệu thu được khác nhau.

Trong nhiều năm qua, công tác lấy ý kiến phản hồi từ người học về hoạt động đào tạo là nhiệm vụ thường xuyên tại cuối mỗi học kỳ tại Trường Đại học Nha Trang (ĐHNT). Trong phiếu đánh giá của Trường, ngoài những tiêu chí định lượng còn có các câu hỏi mở. Thông qua câu hỏi mở, Trường đã nhận được rất nhiều ý kiến khác được người học phản hồi dưới dạng dữ liệu văn bản. Các ý kiến này thường liên quan đến các đề xuất của người học để nâng cao chất lượng đào tạo của Nhà trường, có nhiều ý hay nhưng chưa được xử lý, do việc xử lý thủ công gặp nhiều khó khăn và mất rất nhiều thời gian.

Trong báo cáo này, chúng tôi đề xuất xử lý tự động các ý kiến của người học trong phiếu đánh giá tại Trường ĐHNT bằng phương pháp phân lớp và gán nhãn. Kết quả xử lý ý kiến người học sẽ hỗ trợ các phòng chức năng đánh giá các hoạt động đào tạo đã triển khai và định hướng cho các hoạt động đào tạo trong tương lai (Hình 1).

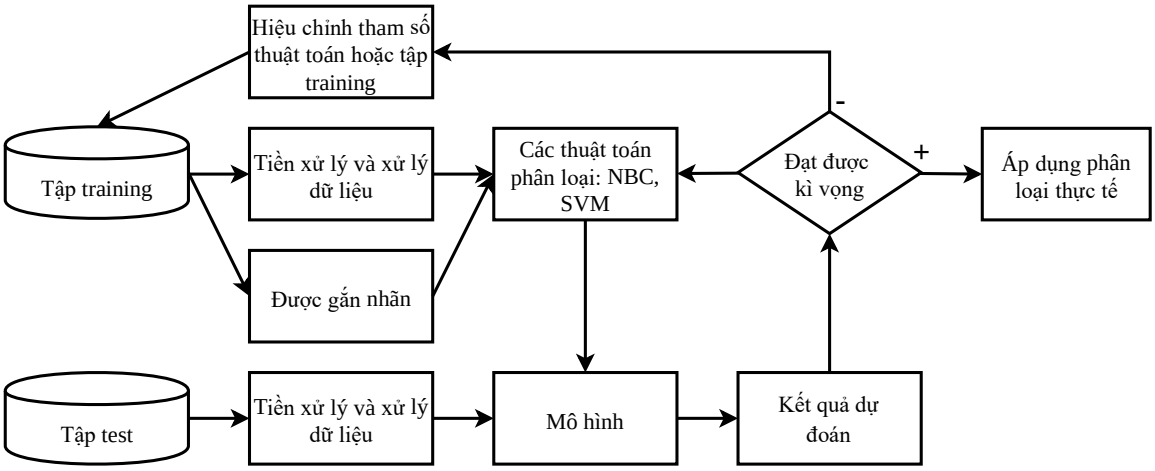


Hình 1. Chu trình xử lý ý kiến người học tại Trường ĐHNT

Các phần của báo cáo gồm: Phương pháp thực hiện, kết quả thử nghiệm và kết luận.

2. PHƯƠNG PHÁP THỰC HIỆN

Phân loại văn bản (*text*) là một bài toán thuộc lĩnh vực học máy (*Machine Learning*). Do đó, để thực hiện phân loại phải trải qua các bước như trong Hình 2.



Hình 2. Minh họa quá trình phân loại văn bản

2.1. Mô tả dữ liệu

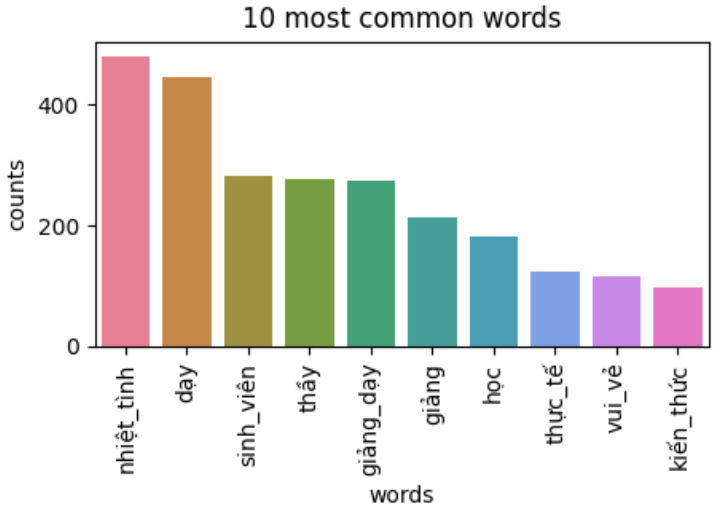
Trong phiếu đánh giá hoạt động giảng dạy của Trường ĐHNT có các câu hỏi mở để người học có thể góp ý cho Nhà trường và giảng viên nhằm nâng cao hơn nữa chất lượng giảng dạy. Hiện nay, việc lấy ý kiến của người học thông qua hệ thống góp ý trực tuyến của Trường. Sau đó, dữ liệu được xuất ra tập tin bảng tính excel để gửi cho các bên liên quan xử lý.

Tập dữ liệu chúng tôi sử dụng trong báo cáo này được lấy ngẫu nhiên một phần từ tập tin excel về ý kiến người học tại Trường Đại học Nha Trang trong học kỳ 2 năm học 2018-2019. Tập dữ liệu này mô tả các ý kiến người học đánh giá cho các hoạt động giảng dạy các học phần khác nhau của các giảng viên thuộc nhiều khoa, viện. Chúng tôi thu được tập dữ liệu gồm 2,953 ý kiến. Dựa trên ý kiến chuyên gia, chúng tôi phân tập dữ liệu thành bốn lớp ứng với các nhãn và số lượng ý kiến như Bảng 1. Tổng số văn bản cho tập training và test lần lượt là 2,064,889.

Bảng 1. Tên nhãn và số lượng văn bản của các tập dữ liệu

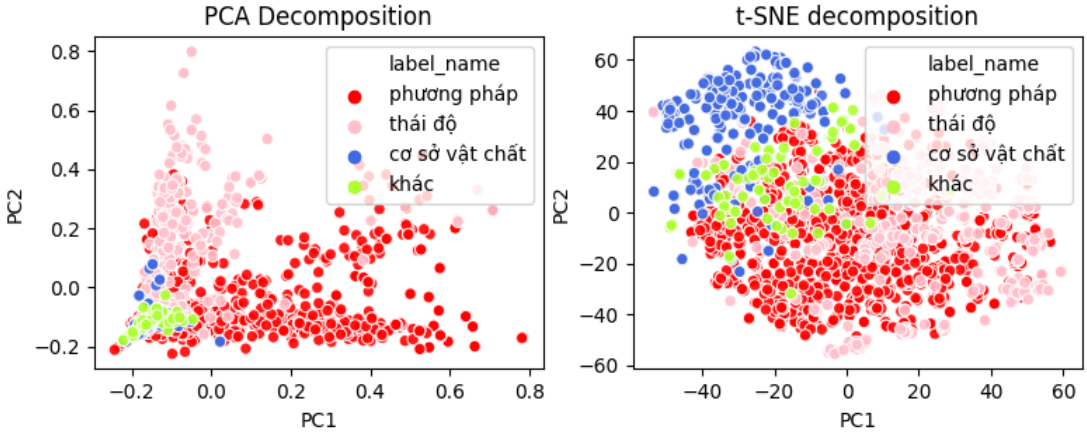
Tên nhãn	Số lượng văn bản cho tập training	Số lượng văn bản cho tập test
Phương pháp giảng dạy của giảng viên	1,099	469
Thái độ của giảng viên đối với người học	518	222
Cơ sở vật chất	355	151
Ý kiến khác	92	47

Qua phân tích dữ liệu, chúng tôi thu được thống kê 10 từ thông dụng xuất hiện nhiều nhất trong tập dữ liệu training ở Hình 3.



Hình 3. 10 từ xuất hiện nhiều nhất trong tập training

Ngoài ra, các vectors đặc trưng từ tập dữ liệu training có số chiều tương đối lớn (được đề cập ở Mục 2.2). Để có thể quan sát được sự phân bố, tương quan của các điểm dữ liệu cũng như lựa chọn mô hình phân lớp hiệu quả, chúng tôi áp dụng kỹ thuật giảm số chiều dữ liệu như PCA (*Principal Component Analysis*) và t-SNE (*t-distributed Stochastic Neighbor Embedding*) trong machine learning (Maaten & Hinton, 2008; Vũ, 2020). Bằng cách giảm số chiều của các vectors đặc trưng xuống còn hai chiều mà vẫn giữ được phần lớn thông tin quan trọng, chúng tôi vẽ được các điểm dữ liệu trong Hình 4. Dựa trên biểu đồ ở Hình 4, chúng ta có thể quan sát được rằng, sử dụng kỹ thuật t-SNE cho kết quả phân các lớp rõ ràng hơn.



Hình 4. Sự phân bố các điểm dữ liệu được vẽ bằng phương pháp PCA và t-SNE

2.2. Tiền xử lý dữ liệu

Đối với bài toán phân lớp ý kiến của người học, chúng tôi áp dụng thuật toán phổ biến hỗ trợ xử lý ngôn ngữ tự nhiên là *Bag-of-words* (*BoW*). BoW có nhiệm vụ phân tích và phân nhóm dựa theo "Bag of Words" (*corpus*) tạo ra bộ *từ điển*. Dựa vào số lần từng từ xuất hiện trong "bag", chúng tôi thu được các vector đặc trưng của văn bản. Đầu vào

của Bag-of-words là đoạn văn bản đã được tách từ (*Words segmentation*). Trong bài viết này, để thực hiện tách từ, chúng tôi sử dụng công cụ ViTokenizer của thư viện pyvi có sẵn trên *Python* do tác giả Trần (2016) xây dựng. Kết quả tách từ thu được độ chính xác từ 96%-98% (xem Bảng 2).

Bảng 2. Ví dụ tách từ tiếng Việt bằng công cụ ViTokenizer của thư viện pyvi

Câu gốc	Câu đã tách từ bằng ViTokenizer
Đầu tư thêm trang thiết bị giảng dạy	Đầu_tư thêm trang_thiết_bị giảng_dạy
Cần phải đi vào chuyên sâu vấn đề giảng dạy hơn nữa	Cần phải đi vào chuyên_sâu vấn_đề giảng_dạy hơn_nữa
Giảng dạy tận tâm	Giảng_dạy tận_tâm
Giảng viên nên chú trọng vào lý thuyết trong bài hơn	Giảng viên nên chú_trọng vào lý_thuyết trong bài hơn

Tuy nhiên theo Hồ và Đỗ (2014) và Vũ (2020), BoW có một số nhược điểm như *từ điển* chứa rất lớn số lượng từ (từ điển của tập dữ liệu "ý kiến người học Trường Đại học Nha Trang" của chúng tôi sử dụng trong bài viết này có kích thước là 1,366), dẫn đến vector đặc trưng thu được sẽ có kích thước rất lớn, có rất nhiều từ trong *từ điển* không xuất hiện trong văn bản dẫn đến trường hợp vector thưa (*sparse vector*). Để khắc phục nhược điểm này, chúng tôi áp dụng phương pháp *Term Frequency-Inverse Document Frequency* (TF-IDF) (Robertson, 2004) để đánh giá độ quan trọng của một từ dựa vào trọng số của từ đó trên toàn bộ văn bản. Tần số xuất hiện *tf* của một từ trong một văn bản dựa trên toàn bộ văn bản trong tập training được tính theo Công thức (1):

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

Trong đó, $f_{t,d}$ là số lần từ t xuất hiện trong văn bản d trên toàn bộ tổng số từ trong văn bản d. Bảng 3 thể hiện tần suất xuất hiện của một số từ trong tập training. Một vài từ có tần suất xuất hiện nhiều thường không có giá trị đặc trưng khi phân loại.

Bảng 3. Bảng tần suất xuất hiện của một số từ trong toàn bộ văn bản

Từ	cân_đối	sinh_viên	dễ	dạy	hiểu	nhiệt_tình
Tần suất	1	296	442	457	473	516

Để giảm giá trị đặc trưng của các từ thường xuyên ở Bảng 3, chúng tôi sẽ tính *idf* theo Công thức (2):

$$idf(t,D) = \log\left(1 + \frac{|D|}{1+|d \in D : t \in d|}\right) + 1 \quad (2)$$

Trong đó, $|D|$ là tổng số văn bản trong tập training. Mẫu số là số văn bản trong tập traning có chứa từ t. Trong Công thức (2) được cộng thêm 1 vì nếu một từ không xuất hiện ở bất cứ văn bản nào trong tập training thì mẫu số sẽ bằng 0. Bảng 4 cho thấy rằng các từ thường xuất hiện ở Bảng 3 đã được đánh lại trọng số quan trọng trong toàn văn bản. Các từ có trọng số càng cao thì càng có giá trị trong phân loại.

Bảng 4. Kết quả tính giá trị trọng số của *idf*

Từ	nhiệt_tình	hiểu	dạy	dễ	sinh_viên	cân_đối
idf values	2.388719	2.497321	2.550667	2.573762	3.091622	7.939738

Sau khi tìm được *tf*, *idf*, công thức *tf − idf* được tính theo Công thức (3):

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \quad (3)$$

2.3. Các thuật toán phân loại

Sau khi tiền xử lý bộ dữ liệu thô "ý kiến người học tại Trường ĐHNT" ở trên, chúng tôi sẽ áp dụng các thuật toán Machine Learning trên bộ dữ liệu vừa thu được. Trong Hồ và Đỗ (2014) và Vũ (2020) đã nêu có rất nhiều thuật toán phân loại văn bản như Naive Bayes Classifier, Decision Tree (Random Forest), Vector Support Machine (SVM), Boosting and Bagging algorithms, Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM, Bi-LSTM), và SLDA. Việc lựa chọn mô hình nào tốt sẽ phụ thuộc vào bộ dữ liệu văn bản đầu vào. Trong khuôn khổ bài viết này, chúng tôi sẽ sử dụng phương pháp NBC (*Naive Bayes Classification*) và SVM vào việc phân loại ý kiến người học của Trường ĐHNT cũng như đánh giá độ hiệu quả của từng phương pháp.

2.3.1. Naive Bayes Classifier (NBC)

Naive Bayes Classification (NBC) là một thuật toán phân loại thuộc nhóm Supervised Learning (học có giám sát) dựa trên tính toán xác suất áp dụng Định lý Bayes. Trong Han, Kamber, và Pei (2011), Karthika và Sairam (2015), và Zhang (2004) đề cập kỹ thuật Naive Bayesian ban đầu dựa trên định nghĩa về xác suất có điều kiện (*conditional probability*) và "Maximum likelihood". Định lý Bayes dùng để tính xác suất ngẫu nhiên của sự kiện *y* khi biết các "*feature vector*" x = $x_1,\ldots,x_n$ ta dùng Công thức (4):

$$P(y|x_1,\ldots,x_n) = \frac{P(y)P(x_1,\ldots,x_n|y)}{P(x_1,\ldots,x_n)} \quad (4)$$

Giả sử rằng các thành phần của "*feature vector*" *x* là độc lập với nhau ta có Công thức (5):

$$P(x|y) = P(x_1 \cap x_2 \cap \ldots \cap x_n|y) = \prod_{i=1}^{n} P(x_i|y) \quad (5)$$

Từ giả thiết của định lý Bayes ở Công thức (4) và (5) được viết lại thành Công thức (6):

$$P(y|x_1,\ldots,x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,\ldots,x_n)} \quad (6)$$

Ở các phương trình trên, ta có mẫu số $P(x_1,\ldots,x_n)$ là các hằng số đầu vào đã cho và không phụ thuộc vào $P(y \mid x_1,\ldots,x_n)$. Do đó, chúng ta có thể áp dụng quy tắc phân loại như sau (Công thức (7)):

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y) \quad (7)$$

Trong đó, $\propto$ là phép tỉ lệ thuận. Công thức (7) được viết lại như sau (Công thức (8)):

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y) \quad (8)$$

Chúng ta có thể sử dụng ước lượng Maximum A Posteriori (MAP) hoặc Maximum Likelihood để tính các phân phối $P(y)$ và $P(x_i \mid y)$ dựa trên tần số tương đối của lớp y trong training data. Ước lượng Maximum Likelihood đưa ra giả sử rằng *feature vector x* tuân theo một phân phối bất kì và được mô tả bằng tham số θ. Trong Vũ (2020), ý tưởng chính của Maximum Likelihood là việc đi tìm bộ tham số θ để xác suất $\theta = \max_{\theta} P(x_1, \dots, x_n|0)$ đạt giá trị lớn nhất. Trong đó, $P(x_1|\theta)$ là một xác suất có điều kiện và $P(x_1,\dots,x_n|\theta)$ là xác suất để toàn bộ các sự kiện $x_1,\dots,x_n$ xảy ra đồng thời (*likelihood*). Với giả thiết từ định lý Bayes rằng các thành phần của *feature vector x* là độc lập, ta có thể quy về bài toán tối ưu (Công thức (9)):

$$\theta = \max_{\theta} \prod_{i=1}^{n} P(x_i|\theta) \quad (9)$$

Bài toán tối ưu (Công thức (9)) được viết lại dưới dạng tương đương bằng cách lấy *log* của vế phải ta được Công thức (10):

$$\theta = \max_{\theta} \sum_{i=1}^{n} \log\left(P(x_i|\theta)\right) \quad (10)$$

Phương trình trên ta có thể áp dụng *log* vào vế phải vì *log* là một hàm đồng biến trên tập các số dương và một biểu thức sẽ là lớn nhất nếu *log* của nó là lớn nhất. Do đó, bài toán Maximum Likelihood được đưa về bài toán Maximum Log–likelihood. Áp dụng quy tắc ở Công thức (10) vào Công thức (8), ta thu được Công thức (11):

$$\hat{y} = arg \max_{y} = \log\big(P(y)\big) + \sum_{i=1}^{n} \log\left(P(x_i|y)\right) \quad (11)$$

Trên thực tế, giả thiết Naive Bayes Classifier đưa ra hầu như không thể xảy ra. Nhưng điều này lại giúp bài toán trở nên đơn giản, hoạt động hiệu quả và cực kì nhanh chóng trong nhiều trường hợp thực tế như bài toán phân loại văn bản, lọc tin nhắn rác hay lọc spam email. Việc tính toán phân phối $P(x_i|y)$ phụ thuộc vào loại dữ liệu. Trong trường hợp này là bài toán phân loại văn bản, chúng tôi sẽ sử dụng phân phối "*Multinomial Naive Bayes"*. Trong mô hình phân phối này, giá trị của thành phần x_i trong mỗi feature vector chính là số lần từ thứ i xuất hiện trong văn bản đó. Phân phối Multinomial Naive Bayes được tham số hóa bởi vector $\theta_y = (\theta_{y1},\dots,\theta_{yn})$ cho mỗi class y, trong đó n là số lượng các đặc trưng hay nói cách khác, n là kích thước của từ điển trong Bag-of-words (n = 1,366 trên bộ dữ liệu training của chúng tôi). θ_{yi} là xác suất $P(x_i|y)$ của đặc trưng thứ i rơi vào các mẫu thuộc class y.

Như đã đề cập ở trên, θ_y được ước lượng bằng cách sử dụng *smoothed version of maximum likelihood* (tương ứng với việc đếm tần suất xuất hiện của từ thứ i trong văn bản) như sau (Công thức (12)):

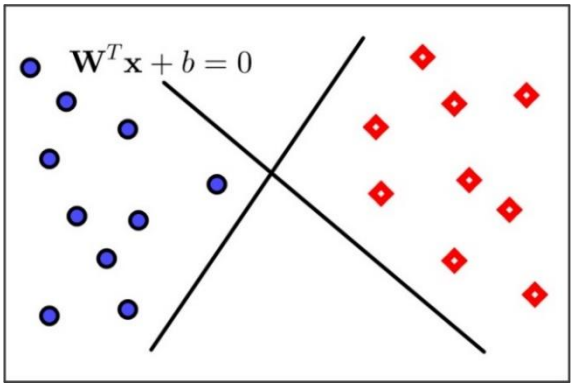
$$\hat{\theta}_{yi} = \frac{N_{yi}+\alpha}{N_y+\alpha n} \quad (12)$$

Trong đó: $N_{yi} = \sum_{x \in T} x_i$ là tổng số lần xuất hiện của đặc trưng thứ *i* rơi vào các văn bản của class *y* trong tập traning *T*; $N_y = \sum_{i=1}^{n} N_{yi}$ là tổng số lần của tất cả các đặc trưng $x_1,...,x_n$ rơi vào class *y*.

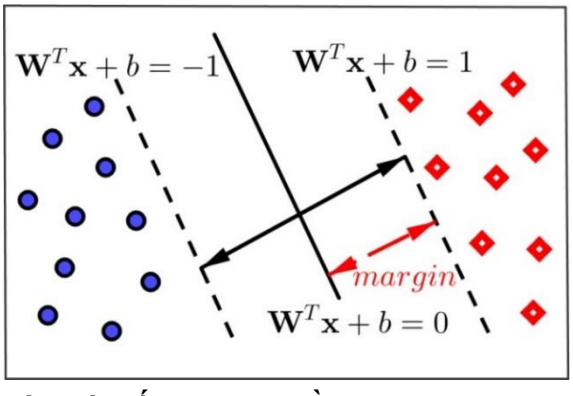
Công thức 12 có thể tránh được hạn chế khi một đặc trưng mới thứ *i* không xuất hiện lần nào trong class *y* của tập traning *T* với mọi $\alpha > 0$. Thông thường, khi chọn $\alpha = 1$ thì được gọi là *Laplace smoothing*, $\alpha < 1$ là *Lidstone smoothing*.

2.3.2. Support Vector Machine (SVM)

Bên cạnh việc sử dụng phương pháp phân loại văn bản đơn giản như NBC, trong bài viết này chúng tôi cũng sử dụng phương pháp Support Vector Machine để phân loại ý kiến người học ở Trường Đại Học Nha Trang. Các nghiên cứu của Joachims (1998); Srivastava và Bhambhu (2010); Trần và Phạm (2012) dựa trên phương pháp SVM cho bài toán phân loại văn bản đều có kết quả rất tốt. SVM cũng là một phương pháp học có giám sát (*supervised learning*) trong các mô hình nhận dạng mẫu dựa trên việc cực đại hóa dải biên phân lớp (*max margin classification*) và lựa chọn các kernel phù hợp (Hình 5 và 6). Phương pháp này có thể hoạt động với các dữ liệu được phân tách tuyến tính và phi tuyến.



Hình 5. Minh họa mặt phân cách giữa hai class



Hình 6. Minh họa bài toán tối ưu SVM bằng cách tìm đường phân chia để thu được max margin

Kỹ thuật của phương pháp SVM được mô tả tổng quát trong không gian *d* chiều như sau: Cho trước $x_1,\ldots,x_N$ điểm và mỗi điểm thuộc vào một class bất kì, cần tìm một siêu phẳng (*hyperplane*) phân hoạch tối ưu sao cho dấu của hàm ước lượng $H = xsign(w^T x + b); w \in R^d, b \in R$ sẽ thể hiện được điểm dữ liệu $x_i \in R^d$ nằm ở cụm dữ liệu nào. Để dễ dàng hiểu được ý tưởng của phương pháp SVM, chúng ta xem xét bài toán phân loại hai lớp trong không gian hai chiều như hình minh họa. Rõ ràng trong Hình 6 chúng ta có thể tìm được nhiều đường phân tách, nhưng nếu chọn được một đường phân tách tối ưu như Hình 6 thì kết quả sẽ tốt hơn. Nhiệm vụ của phương pháp SVM là đi tìm đường thẳng (siêu phẳng) như Hình 6. Xem xét tập training có dữ liệu có thể tách rời tuyến tính $(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)$. Với mỗi điểm x_i tương ứng với nhãn $y_i \in \pm 1$ (dấu về phía âm hoặc dương), ta thu được đường phân tách giữa hai class là H : $w^T x + b = w_1x_1 + w_2x_2 + b = 0$ và hai đường thẳng biên gốc H_1, H_{-1} song song với H và có cùng khoảng cách đến H. Với cặp dữ liệu (x_n,y_n) bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là $\frac{y_n(w^T x_n+b)}{||w||_2}$. Trong Hình 6, *margin* được tính là khoảng cách gần nhất từ một điểm bất kì trong class nào tới mặt phân cách: $margin = \min_n \frac{y_n(w^T x_n+b)}{||w||_2}$.

Bài toán tối ưu trong SVM trở thành bài toán xác định *w* và *b* sao cho *margin* đạt giá trị lớn nhất (Công thức (13)).

$$\begin{aligned}(w,b) &= arg\max_{w,b}\left\{\min_n \frac{y_n(w^T x_n + b)}{\left|\left|w\right|\right|_2}\right\} \\ &= arg\max_{w,b}\left\{\frac{1}{\left|\left|w\right|\right|_2}\min_n y_n\,(w^T x_n + b)\right\}\end{aligned} \tag{13}$$

Giả sử rằng không có phần tử nào của tập mẫu nằm giữa H_1 và H_{-1}, tức là $w.x+b >= +1$ với $y = +1$ và $w.x + b >= -1$ với $y = -1$, ta thu được Công thức (14).

$$y_n(w^T x_n + b) = 1, \forall n = 1,2,\ldots,N \tag{14}$$

Bài toán tối ưu (Công thức 13) đồng nghĩa với việc $||w||$ đạt nhỏ nhất với ràng buộc ở Công thức (14).

$$\begin{aligned}(w,b) &= arg\max_{w,b}\frac{1}{\left|\left|w\right|\right|_2} \\ &subject\ to:\quad y_n(w^T x_n + b) \geq 1, \forall n = 1,2,\ldots,N\end{aligned} \tag{15}$$

Trong đó, Phương trình (15) đã chuyển sang dạng lấy bình phương và chia đôi để dễ dàng tính toán hơn và tối ưu lồi (cả hàm mục tiêu và hàm ràng buộc đều là lồi). Chúng ta có thể giải bài toán lồi này thông qua bài toán đối ngẫu của nó bằng cách cực tiểu hóa hàm Lagrange (Công thức (16)):

$$\mathcal{L}(w,b,\lambda) = \frac{1}{2}||w||_2^2 + \sum_{n=1}^{N}\lambda_n\left(1 - y_n(w^T x_n + b)\right) \tag{16}$$

Với $\lambda = [\lambda_1,\lambda_2,...,\lambda_N]^T$ là các hệ số Lagrange, $\lambda_n \geq 0$, $\forall n$. Tiếp theo, bài toán được chuyển thành bài toán đối ngẫu bằng cách cực đại hóa hàm λ (Công thức (17)):

$$\lambda = \arg\max_{\lambda} \left[\min_{w,b} \mathcal{L}(w, b, \lambda)\right] \quad (17)$$

subject to: $\lambda \geq 0$,

$$\sum_{n=1}^{N} \lambda_n y_n = 0$$

Giải λ có thể được thực hiện bằng phương pháp quy hoạch động bậc 2 (*Quadratic Programing*). Từ đó ta có thể tìm được các tham số:

$$w = \sum_{i=1}^{N} \lambda_i y_i x_1, \qquad b = y_i - \sum_{j=1}^{N} \lambda_j y_j x_j^T x_i$$

Trong đó, *Support Vector*: (x_i,y_i) là một tập điểm dữ liệu bất kì nào đó nằm trên đường biên gốc. Cuối cùng, khi phân loại một mẫu mới sẽ tiến hành kiểm tra hàm dấu *sign*(wx + *b*).

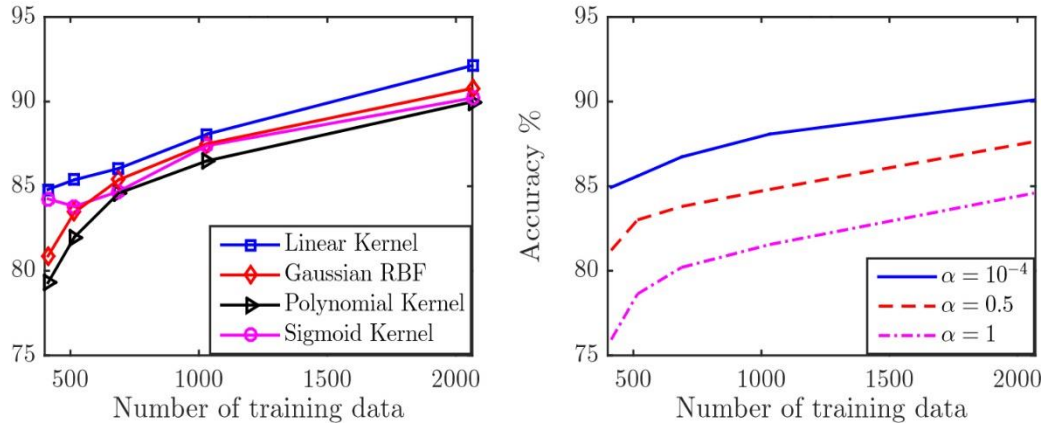
Trong thực tế, dữ liệu được phân tách từ tập traning là phi tuyến, có sự chồng lấn nhau (nhiễu). Dẫn đến các siêu phẳng bây giờ có thể là một mặt cong để phù hợp phân tách dữ liệu. Siêu phẳng này có thể tìm thông qua ánh xạ dữ liệu vào một không gian có số chiều lớn hơn bằng cách sử dụng một hàm nhân *K* (kernel) thoả mãn điều kiện Mercer. Trong Nandi (2014) nêu một số kernel phổ biến thường được sử dụng theo Bảng 5.

Bảng 5. Các kernels thông dụng

Hàm	Công thức
Linear Kernel	$K(x,y) = < x,y >$
Polynomial Kernel	$K(x,y) = < x,y >^d$, *d*: bậc của đa thức
Gaussian RBF	$K(x,y) = exp\ -\frac{1}{2\sigma^2} \parallel x - y \parallel^2) = exp(-\gamma \parallel x - y \parallel^2)$
Sigmoid Kernel	$K(x,y) = tanh\ (\gamma x^T y + \beta)$, $\gamma,\beta \geq 0$

3. KẾT QUẢ

Áp dụng các bước đã trình bày ở phần phương pháp thực hiện cho bài toán phân loại trên tập ý của kiến người học Trường ĐHNT. Cụ thể, sử dụng thuật toán SVM trên tập training có kích thước lần lượt là *413, 516, 688, 1032, 2064* với các kernels ở Bảng 5, ta thu được kết quả dự đoán ở Hình 7a. Hình 7b thể hiện độ dự đoán chính xác của thuật toán NBC trên cùng tập traning với mô hình SVM.

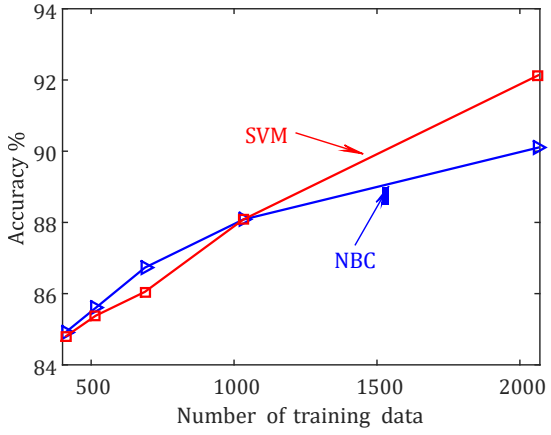


(a) (b)

Hình 7. Kết quả dự đoán (%) của thuật toán SVM và NBC

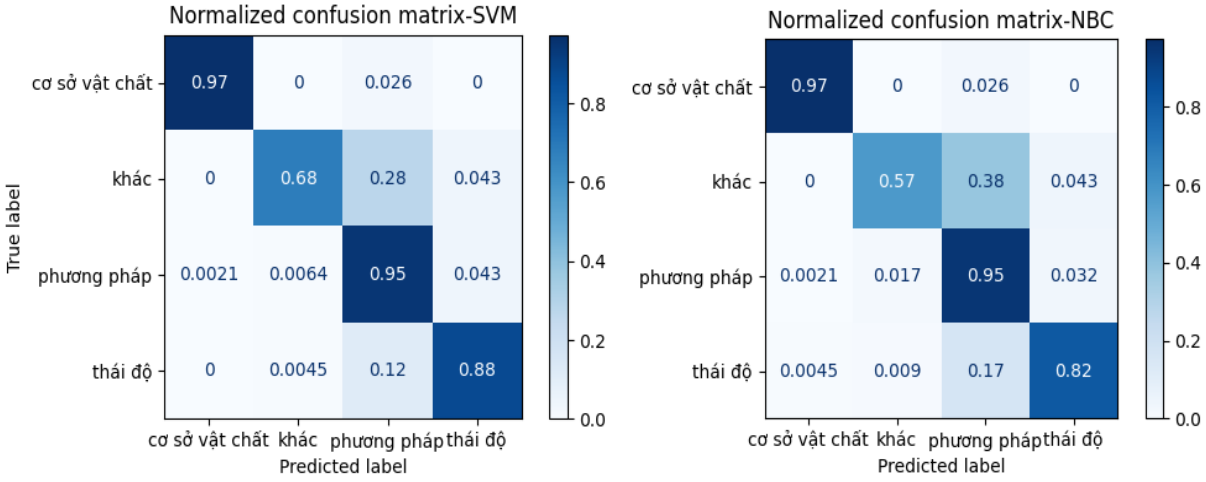
Ghi chú: a) SVM sử dụng Linear Kernel, Polynomial Kernel, Gaussian RBF, và Sigmoid Kernel và b) NBC với $\alpha = 10^{-4}$,0.5,1.

Theo kết quả Hình 7, khi kích thước tập training tăng lên dẫn đến số chiều của các vectors đặc trưng lớn thì phương pháp SVM làm việc hiệu quả hơn phương pháp Naive Bayes, điều này phù hợp với chứng minh của Joachims (1999). Chúng ta cũng có thể quan sát được trong Hình 7a, Linear kernel của phương pháp SVM cho kết quả tốt nhất trên các tập training. Hơn nữa, khi sử dụng các tham số mặc định như α=1 cho mô hình Naive Bayes thì kết quả baseline khá thấp, đặc biệt đối với tập training có kích thước nhỏ (dưới 76%). Do đó, chúng tôi đã tối ưu các tham số để thu được kết quả tốt nhất trên tập dữ liệu ý kiến người học của Trường Đại học Nha Trang. Các tham số được sử dụng trong tính toán trên là $\alpha = 10^{-4}$, 0.5,1 cho thuật toán NBC; $d = 2$, $\gamma = 1/1{,}366$, và $\beta = 0$ cho thuật toán SVM. Đối với tập training có kích thước là 2,064 ý kiến và tập test là 889 ý kiến, chúng tôi thu được độ dự đoán chính xác cao nhất của mô hình NBC là 90.10% và SVM là 92.13%. Rõ ràng, với tập dữ liệu có nhiễu và có số chiều vectors đặc trưng lớn như trong bộ dữ liệu chúng tôi đã sử dụng thì phương pháp SVM cho kết quả dự đoán tốt hơn phương pháp NBC, điều này được thể hiện trong trong kết quả quả so sánh bên dưới. Hình 8 biểu diễn kết quả so sánh của hai phương pháp SVM và NBC.



Hình 8. So sánh kết quả dự đoán của mô hình NBC và SVM

Áp dụng các tham số tối ưu nhất ở trên cho cả hai phương pháp vào tập training có kích thước lớn nhất là 2,064, chúng ta có thể quan sát được chi tiết các điểm dữ liệu đã được phân vào lớp nào bằng cách sử dụng Confusion Matrix (Vũ, 2020).



(a) **(b)**

Hình 9. Confusion matrix cho hai phương pháp

Ghi chú: a) SVM và b) NBC.

Dựa vào Hình 9, ta thấy rằng các điểm dữ liệu ở lớp "ý kiến khác" bị phân loại nhầm nhiều nhất và chúng được phân loại nhầm vào lớp "phương pháp giảng dạy", "thái độ của giảng viên đối với người học". Dữ liệu ở lớp "phương pháp giảng dạy" và "cơ sở vật chất" được phân loại đúng nhiều nhất. Kết quả phân loại cho lớp "ý kiến khác" và lớp "thái độ của giảng viên đối với người học" của phương pháp SVM cao hơn phương pháp NBC. Từ những nhận xét trên, để cải thiện kết quả dự đoán, chúng tôi cần phải hiệu chỉnh lại tập training cho lớp "ý kiến khác" để thu được kết quả dự đoán tốt hơn.

Ngoài ra, tập dữ liệu chúng tôi đang sử dụng bị mất cân bằng dữ liệu giữa các lớp, có sự chênh lệch rất lớn. Do đó, phương pháp precision và recall được sử dụng để đánh giá hiệu quả phân loại của mô hình (Vũ, 2020) (Bảng 6 và 7).

Bảng 6. Ước lượng dựa trên Precision và Recall cho phương pháp NBC

Class	Precision	Recall	F1-score	Support
Cơ sở vật chất	0.99	0.97	0.98	151
Khác	0.73	0.57	0.64	47
Phương pháp	0.88	0.95	0.91	469
Thái độ	0.91	0.82	0.86	222
Macro avg	0.88	0.83	0.85	889
Weighted avg	0.90	0.90	0.90	889

Bảng 7. Ước lượng dựa trên Precision và Recall cho phương pháp SVM

Class	Precision	Recall	F1-score	Support
Cơ sở vật chất	0.99	0.97	0.98	151
Khác	0.89	0.68	0.77	47
Phương pháp	0.91	0.95	0.93	469
Thái độ	0.90	0.88	0.89	222
Macro avg	0.92	0.87	0.89	889
Weighted avg	0.92	0.92	0.92	889

Dựa trên kết quả của F1-score chúng ta có thể đi đến kết luận rằng mô hình SVM hoạt động khá tốt đối với tập dữ liệu "ý kiến người học tại Trường ĐHNT".

4. KẾT LUẬN

Trong bài báo này, với mong muốn hiểu được các phản hồi của người học về các hoạt động đào tạo của Nhà trường, chúng tôi đã đề xuất xử lý tự động ý kiến người học dựa trên các phương pháp phân loại văn bản. Kết quả thử nghiệm khá khả quan trên tập dữ liệu ý kiến người học tại Trường ĐHNT-một tập dữ liệu có nhiễu với phương pháp SVM là 92.13% và NBC là 90.10%. Bước tiếp theo, chúng tôi sẽ thực hiện tối ưu mô hình, thử nghiệm với các phương pháp phân loại khác để cải thiện độ chính xác của mô hình phân lớp, xử lý và phân tích thêm để hiểu rõ hơn các ý kiến của người học, từ đó hỗ trợ các đơn vị chức năng đưa ra các đề xuất phù hợp để nâng cao chất lượng đào tạo của Nhà trường.

TÀI LIỆU THAM KHẢO

Han, J., Kamber, M., & Pei, J. (2011). *Data mining concepts and techniqu*es (3rd ed.). Massachusetts, USA: Morgan Kaufmann Publishing.

Harvard University. (n.d). *Getting feedback.* Retrieved from https://bokcenter.harvard.edu/getting-feedback.

Hồ, T. T., & Đỗ, P. (2014). Mô hình tích hợp khám phá, phân lớp và gán nhãn chủ đề tiếp cận theo mô hình chủ đề. *Tạp chí phát triển KH&CN, 17*(K4-2014), 73-85.

Joachims, T. (1998). *Text categorization with Support Vector Machines: Learning with many relevant features*. Paper presented at The 10th European Conference on Machine Learning (ECML-98), Chemnitz, Germany.

Joachims, T. (1999). *Transductive inference for text classification using Support Vector Machines*. Paper presented at The 16th International Conference on Machine Learning (ICML'99), San Francisco, USA.

Karthika, S., & Sairam, N. (2015). Naïve Bayesian classifer for educational qualifcation. *Indian Journal of Science and Technology, 8*(16), 1-5.

L-Universitàta' Malta (UM) (2020). *Student feedback.* Retrieved from https://www.um.edu.mt/ services/administrativesupport/apqru/studentfeedback.

Maaten, L.V., & Hinton, G. E. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579-2605.

Nandi, M. (2014). *Kernel theory recitation*. Pennsylvania, USA: Carnegie Mellon University-Machine Learning Department Publishing.

Robertson, S. E. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, *60*(5), 503-520.

Srivastava, D., & Bhambhu, L. (2010). Data classification using Support Vector Machine. *Journal of Theoretical and Applied Information Technology, 12*(1), 1-7.

Trần, C. Đ., & Phạm, N. K. (2012). Phân loại văn bản với máy học vector hỗ trợ và cây quyết định. *Tạp chí Khoa học Trường Đại học Cần Thơ*, (21a), 52-63.

Trần, V. T. (2016). *Python Vietnamese toolkit*. Retrieved from https://pypi.org/project/pyvi/.

Vũ, H. T. (2020). *Machine Learning cơ bản.* Retrieved from https://github.com/tiep vupsu/ebookMLCB.

Zhang, H. (2004). *The optimality of Naive Bayes*. Paper presented at The 17th International Florida Artificial Intelligence Research Society Conference, Florida, USA.