Classification of Vietnamese Documents Using Support Vector Machine

Bui Thanh Hung
Engineering - Technology Faculty
Thu Dau Mot University

Abstract: In this paper, we present studies on Vietnamese document classification problem using Support Vector Machine (SVM). SVM is a learning method with ability to automatically tune the capacity of the learning machine by maximizing the margin between positive and negative examples in order to optimize the generalization performance, SVM has a large potential for the successful applications in the field of text categorization. This paper presents the results of the experiment of Vietnamese text categorization with SVM

Keywords: Support Vector Machine, Text classification

I. INTRODUCTION

Automatic classification is one of the classic problems in the field of text data processing. This is an important issue when dealing with large amounts of data. There have been many studies in the world that have shown positive results in this direction. However, the research and application of Vietnamese text is still limited. Many reasons are specific to the Vietnamese language vocabulary and sentences.

In the field of data mining, text classification methods have been based on decisive methods such as Bayes decision, decision tree, K-nearest neighbor, neural network, etc. These methods have given The results are acceptable and are used in practice. In recent years, classification methods using the SVM have been of interest and use in many fields of identification and classification. SVM is a family of kernel-based methods to minimize the risk of estimation. The SVM method is derived from statistical theories developed by Vapnik and Chervonenkis and has a great potential for theoretical as well as practical applications. The empirical tests show that the SVM method is well suited to classification problems as well as in many other applications (such as handwriting recognition, face detection in images, estimates regression, ...). Compared with other classification methods, the classification ability of SVM is equal or significantly better.

In this paper, we first describe the basis of the SVM method and the algorithm for solving the quadratic problem arising from this method.

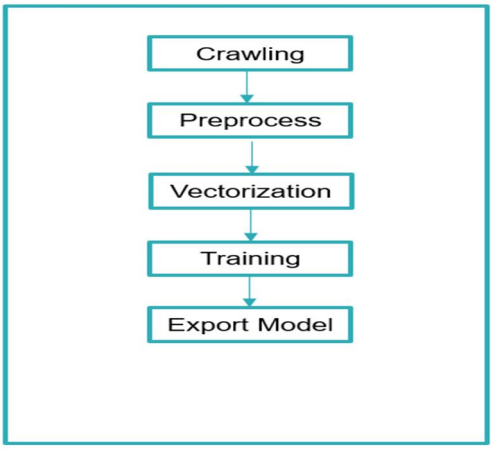

Figure 1. The proposed model

The next part deals with the problem of classifying text in vector expressions.

We emphasize the pre-processing aspect, character selection, text representation, and conformity analysis of the SVM method applied to the text classification problem. The last part is the SVM application results in the Vietnamese text classification. These experiments were designed to verify the SVM classification ability of Vietnamese text.

II. PROPOSED MODEL

With document classification problem, there are many methods based on Naïve Bayes, decision tree, K-nearest neighbor,… These methods achieved acceptable result and they were used in real life.

In this document, we will use Support Vector Machine for our solution because of it advantages with others.

Our proposed model will present in the Figure 1. Our proposed model will contain following step:

Collecting data: Dataset in machine learning is really important, we build dataset by collect Vietnamese newspapers from big News websites, newspapers have already classified into categories. We split data into 2 parts: first part for training and second one for testing.

Preprocessing data: After collecting data from websites, the aim of preprocessing data is removing some redundancy parts of its content such as html tags.

Transform words to bag of word and TI-IDF features: Transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task.

Training by SVM model with extracted features: Training and exporting model using SVM algorithms and its libraries.

Evaluate the proposed model: Testing model by testing model with testing data, then evaluate it using score and f1 score.

III. DOCUMENT CLASSIFY

3.1 Vetorization

The transformation from text to vector is important, it helps the machine understand the natural language and also understand the importance of each word in the data. We use vectorization to extract features for SVM model.

There are four steps in vectorization:

- ✓ Separation of words
- ✓ Put words into the vocabulary
- ✓ Performed in vector form
- ✓ Calculate the weight of the word by TFIDF (if the concept and calculation of TF-IDF)

- ***Term frequency – TF:*** Word weight is the frequency of occurrence of that word in the document. This weighting says that a word is important to a document if it appears multiple times in the document.
- ***TFIDF:*** Word Weight is the product of the frequency from the TF and the inverse document frequency and is determined by the formula.

$$\text{IDF} = \log(N / DF) + 1 \qquad (13)$$

where N is the size of the training material;

DF is the frequency of a document: the number of documents that a word appears.

The *TFIDF* weight factor adds the *DF* value to the *TF*. When a word appears in fewer documents (corresponding to a small *DF* value), the greater the likelihood of distinguishing documents based on that word.

3.2 Support vector machine (SVM)

The basic characteristic that determines the classification ability of a classifier is the generalized efficiency, or the ability to classify new data based on the knowledge accumulated during the training. The coaching algorithm is considered to be good if, after training, the generalization efficiency of the receiver is high. Generalized efficiency depends on two parameters: training error and ability of machine learning. In which the training error is the classification error rate on the training data set. Machine learning capacity is determined by the size of VapnikChervonenkis (VC size). VC size is an important concept for a family of separators (or classifiers). This quantity is determined by the maximum number of points that the function can completely divide in the object space. A good classifier is the lowest-performing classifier (that is, the simplest) and ensures a small training error. The SVM method is based on this idea.

Consider the simplest classifier problem - classify the class with the sample dataset:

$$\{(x_i, y_i) \mid i = 1, 2, ..., N, x_i \in R^m\}$$

Where samples are object vectors categorized as positive and negative samples:

- Positive samples are x_i samples in the field of interest and labeled $y_i = 1$;

- Negative samples are samples of x_i not belonging to the domain of interest and labeled $y_i = -1$;

In this case, the SVM classifier is the superset of the sample that separates positive samples from negative samples with the maximum difference, in which the difference - also known as the margin - is defined by the distance between samples positively and

negatively near the hyperplane (Figure 2). This hyperplane is called the super-smooth margin.

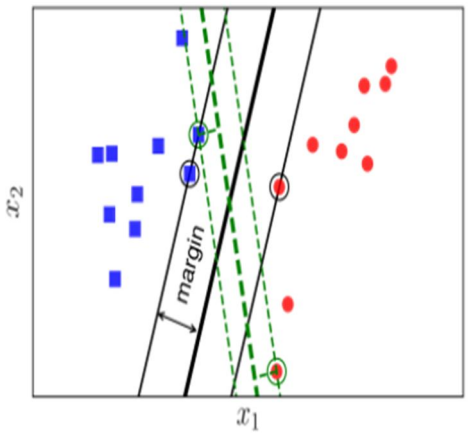

Figure 2: Hyperplane separates positive samples from negative samples.

If the training dataset is *linearly separable*, we have the following constraints:

$$\mathbf{w}^T\mathbf{x}_i + b \geq +1 \text{ if } y_i = +1 \quad (2)$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1 \text{ if } y_i = -1 \quad (3)$$

Two hyperplanes with the equation $\mathbf{w}^T\mathbf{x} + b = \pm 1$ are called superseded planes (the dashed lines in Figure 1).

3.3 Document Classify

We used LinearSVC algorithms for our problems because of its advantages with large text corpus. The input of LinearSVC is a set of extracted features(X) and a set of corresponding labels(y).

In pseudocode, the training algorithm for an OvR learner constructed from a binary classification learner *L* is as follows:

Inputs:

- *L*, a learner (training algorithm for binary classifiers)
- Samples *X (features extracted in the previous step)*
- labels *y* where $y_i \in \{1, \ldots K\}$ is the label for the sample X_i

Output:

- A list of classifiers f_k for k ∈ {1, …, K}

Procedure:

For each k in {1, …, K}

- Construct a new label vector z where z_i= 1 if y_i = k and z_i = 0 otherwise
- Apply *L* to *X*, z to obtain f_k

Making decisions means applying all classifiers to an unseen sample x and predicting the label k for which the corresponding classifier reports the highest confidence score:

$$\hat{y} = \underset{k \in \{1 \ldots K\}}{\operatorname{argmax}} f_k(x)$$

After training process finished, a model can predict any unseen documents.

4 EXPERIMENT RESULT

We have implemented a SVM application experiment in Vietnamese text classification. A sample of 84132 documents is available from http://vnexpress.net, http://tuoitre.vn/, http://thanhnien.vn/, http://nld.com.vn/ (table 1). This brochure is divided into two parts: 50% is used as a training manual, 50% is used as a test document as shown in Table 1.

In this experiment, for the pre-processing, we use a Pyvi toolkit to tokenize words in document. We remove stopwords in the text using a list of available stop words.

We evaluate our proposed model base on Confusion Matrix, Accuracy and F1-score.

Document Type	Train (33759)	Test (50373)
Chính trị	3159	2036
Khoa học	1820	2096
Kinh doanh	2552	5276
Pháp luật	3868	3788
Sức khỏe	3384	5417
Thế giới	2898	6716
Thể thao	5298	6667
Văn hóa	3080	6250
Vi tính	2481	4560

Table 1: Sample documents used in the Vietnamese text classification experiment.

Table 2 shows our confusion matrix.

Label	Total file	True prediction
Chính Trị	7567	6810
Đời Sống	2036	1303
Khoa Học	2096	1655
Kinh Doanh	5276	4748
Pháp Luật	3788	3484
Sức Khỏe	5417	5200
Thế Giới	6716	6245
Thể Thao	6667	6600
Văn Hóa	6250	5998
Vi Tính	4560	4423

Table 2: Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Table 3 shows our result in Accuracy.

Label	Accuracy
Total	**92%**
Chính Trị	89%
Đời Sống	63%
Khoa Học	79%
Kinh Doanh	89%
Pháp Luật	92%
Sức Khỏe	95%
Thế Giới	93%
Thể Thao	98%
Văn Hóa	95%
Vi Tính	96%

Table 3: Accuracy

Model	F1-Score
SVM	89.4%

Table 4: F1- Score

The formula for the F1 score is:

$$F1 = \frac{2 * (precision * recall)}{Precision + recall}$$

Table 4 shows our F1-score. **Compare with Naïve Bayes model**

To evaluate our proposed model, we install Naïve Bayes model and compare the result of our model with Naïve Bayes model. The result shows that our proposed model get better than Naïve Bayes model both on F1 and accuracy score as shown in Table 5.

Model	F1-Score	Score
SVM	89.4%	92%
Naïve Bayes	89%	90%

Table 5: Compare with Naïve Bayes model

We build a web application to visualize our proposed model for Vietnamese document classify. Figure 3 shows our web application.

Figure 3: Web application

5 CONCLUSION

In this paper, we investigated the efficiency of the SVM classification method. This is a classifier that automatically adjusts the parameters to optimize classification efficiency even in high-dimensional feature spaces. SVM classifiers are suitable for text classification. In testing with the Vietnamese text classification problem, classification accuracy was 92% acceptable under practical conditions. At present, we are continuing to research the improvement of pre-processing of text, building standard training samples as well as adjusting SVM algorithms to further improve classification accuracy.

6 REFERENCES

[1] B. BOSER, I. GUYON, V. VAPNIK, "*A training algorithm for optimal margin classifiers*", Proceedings of the Fifth Annual Workshop on Computational Learning Theory (ACM), pp 144-152, 1992.

[2] C. BURGES, "*A tutorial on Support Vector Machines for pattern recognition*", Proceedings of International Conference on Data Mining and Knowledge Discovery, Vol 2, No 2, pp 121-167, 1998.

[3] S. DUMAIS, J. PLATT, D. HECKERMAN, M. SAHAMI, "*Inductive learning algorithms and representations for text categorization*", Proceedings of Conference on Information and Knowledge Management (CIKM), pp 148-155, 1998.

[4] T. JOACHIMS, "*Text categorization with Support Vector Machines: Learning with many relevant features*", Technical Report 23, LS VIII, University of Dortmund, 1997

[5] Sch¨olkopf, B., A. Smola, R. C.Williamson, and P. L. Bartlett (2000). "New support vector algorithms". Neural Computation 12(5), 1207–1245

[6] Jana Novovicová, "Text document classification", ERCIM News No. 62, July 2005

[7] Rupali Bhaisare,T. RajuRao 2013 "Review On Text Mining With Pattern Discovery".

[8] Vandana Korde, C Namrata Mahender, "TEXT CLASSIFICATION AND CLASSIFIERS: A SURVEY", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March 2012

[9] M. IKONOMAKIS, S. KOTSIANTIS, V. TAMPAKAS, "Text Classification Using Machine Learning Techniques", WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, August 2005, pp. 966-974