# Support Vector Machines Combined With Fuzzy C-Means For Text Classification

**Vu Thanh Nguyen**

University of Information Technology

**Summary**

In this paper we implement an application of support vector machines for collecting and classifying information on the Internet to support administrative websites of local government services in providing information. We also propose a method using support vector machines combined with fuzzy c-means to improve the classification and compare with our previous work on fuzzy support vector machines.

***Keywords:*** *SVM, FSVM, FCSVM, fuzzy c-means*

## 1. Introduction

One of the important tasks of local government services of HoChiMinh city is to provide citizens and companies with policies and information which they are in charge. The information can be provided by the services themselves, or collecting from other news websites. Therefore, we design an application to collect and classify information on many news websites automatically. The implementation chart of application includes two main steps: information collection step and information classification step.

In information collection step, we search for news WebPages on news websites first and then use matching algorithm built on sample identification method ([2],[7],[9]) to automatically extract information on news WebPages.

In information classification step, we use support vector machines combined with fuzzy c-means method (FCSVM) and fuzzy multiclass classification to improve classification results.

Fuzzy support vector machines are an innovation of support vector machines. It has been developed by Chun Fu Lin and Shen De Wang ([4]). It increases classification accuracy especially in case training data have noise. In this paper we propose a method using support vector machines combined with fuzzy c-means to improve the quality of training data by removing noise data. Therefore, the classification result will be better. Then we compare the proposed method with fuzzy support vector machines (FSVM).

Manuscript received March 5, 2010

Manuscript revised March 20, 2010

## 2. Extracting information from WebPages by matching algorithm

The extracting method by matching algorithm allows for extracting information zone which contains the main information on the website exactly. This method is made by matching two WebPages, one need to extract and the other sample webpage to determine the common presentation frame for both WebPages. From this common presentation frame, we can extract the main content from the necessary webpage.

In order to extract information by matching, the two WebPages are parsed into two trees A and B respectively and then perform matching on these two trees. We use HtmlParser library to analyze into multi-branches tree with root. The tree has three types of node: tag node, text node and remark node.

Two nodes are matched:

1. If two nodes are tag nodes and have the same tag name.
2. If two nodes are text nodes or remark nodes, they are matched when all text content of two nodes are the same.

Other cases are mismatched.

### Matching algorithm as follows:

**Input** : two root nodes of tree A and B.

**Output** :

- retList: list of extracted nodes contain information of webpage (text node, remark node)

- weight: maximum matches of matching algorithm.

- **Case 1 : Two root nodes A and B are not similar:**
  - retList = null
  - weight = 0
- **Case 2 : A node has no child node**
  - retList = null
  - weight = 1
- **Case 3: Node A has child node and B has not**

- retList = child nodes contain information of node A

- weight = 1

- **Case 4: Both A and B has child node**

Call recursive matching algorithm on i-th and j-th child tree of A and of B:

- retList include:

  - Nodes contain information not participating into matching
  - Nodes contain information (Case 3) from matching

- weight = maximum matches between A and B.

## 3. Information Classification

### 3.1 Support vector machines (SVM)

Let's view a problem in classifying text by SVM ([1], [6]) details as follows:

**Problem** : Check whether a certain text d belonging to a given class c? If $d \in c$ then d is labeled 1 otherwise d shall be labeled –1.

Supposedly, we select a specific set $T=\{t_1, t_2, \ldots, t_n\}$, then each text $d_i$ shall be presented by a data vector $x_i=(w_{i1}, w_{i2}, \ldots, w_{in})$, $w_{ij} \in R$ which is the weight of the word $t_i$ in text $d_i$.

The training data of SVM is the set of texts to be pre-labeled $Tr=\{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\}$, $y_i \in \{+1, -1\}$, the pair $(x_i, y_i)$ is understood that vector $x_i$ is labeled $y_i$. The idea of SVM is to find an optimal hyperplan f(x) in a space with n-dimension to classify data in a way so that all of the $x_+$ points labeled 1 belong to the positive of hyperplan $(f(x_+)>0)$, and $x_-$ points labeled –1 belong to the negative of hyperplan $(f(x_-)<0)$. Then, determination of a text $x \notin Tr$, whether it belongs to class c, corresponding to the sign of f(x). If $f(x)>0$ then $x \in c$, if $f(x) \le 0$ then $x \notin c$.
Given a set of data:

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

**Case 1**
If *Tr* data set can be classified in linear without noise, we can find a linear hyperplan in the formula (1) to classify data set. The optimal hyperplan is equivalent to solution of the following optimal problem:

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

**Case 2**
*Tr* training data set can be classified in linear with noise which means points labeled positive belong to negative side and points labeled negative belong to positive side of the hyperplan. Problem (1) becomes:

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

$\xi_i$ is a slack variable, $\xi_i \ge 0$; C is predetermined parameter determining ties up value. The bigger C is, the higher empirical risk is.

**Case 3**
*Tr* training data set cannot be classified in linear. In this case, data vector x is mapped from a n-dimension space into a m-dimension space (m>n), so that data can be linear classified in m-dimension space. Supposed that $\phi$ is a non-linear mapping from $R^n$ space into $R^m$ space.

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

Hence, vector $x_i$ in $R^n$ space will be correlative to vector $\phi(x_i)$ in $R^m$ space

Replace (2) with $\phi(x_i)$, result in (3):

$$\begin{cases} \operatorname{Min} \Phi(w,\xi) = \dfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ y_i(w^T.\phi(x_i) + b) \ge 1 - \xi_i, \qquad i = 1,\ldots,l \\ \xi_i \ge 0 \qquad i = 1,\ldots,l \end{cases} \qquad (3)$$

Straightly calculating $(x_i)$ is difficult and complicated. So a kernel function $K(x_i,x_j)$ is used to calculate scalar product $\phi(x_i)\phi(x_j)$ in m-dimension space.

$$K(x_i, x_j) = \phi(x_i)\phi(x_j)$$

Some kernel functions are often used in text classification, include:

***Linear Function:*** $K(x_i, x_j) = x_i^T x_j$

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

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

### 3.2 Fuzzy Support Vector Machines (FSVM)

The training data usually has noise data points. These points are not belonging to a class correctly or completely. They will affect the process of training. There are several methods used to solve this problem. One of these methods is Fuzzy Support Vector Machines ([4]), this method is effective in reducing the influence of noise data points to the results of training.

In normal SVM, each point entirely belongs to either two class. However, in some cases some points belong to a class incompletely. These points are called noise points. Moreover, each point of data may not have the same meaning to hyperplan. Solving this problem, Lin CF. and Wang SD have introduced FSVM method by utilizing a membership function to determine contribution value of each point in SVM training data.
The problem is described as follows:

$$\begin{cases} \mathrm{Min}\,\Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} s_i\xi_i \\ y_i(w^T.\phi(x_i)+b) \geq 1-\xi_i, \quad i=1,...,l \\ \xi_i \geq 0 \quad i=1,...,l \end{cases} \quad (4)$$

$s_i$ is a member function satisfying $\sigma \leq s_i \leq 1$, $\sigma$ is a constant > 0, representing effective value of $x_i$ point to a class. The value $s_i$ can decrease the value of $\xi_i$ variable, so $x_i$ point corresponding to $\xi_i$ may reduce effective value.

### 3.3 Support vector machines combined with Fuzzy c-means (FCSVM)

Reducing the influence of noise data in training data set in some cases are not good, especially when there are too many noise points, these points still affect the process of formatting hyperplan. So there are several different approaches to remove the influence of this noise data points. One of this approach is the combination of SVM and k-NN ([3]). Similar to that approach, our proposed method is combining support vector machines with fuzzy c-means clustering algorithm to remove noise data points from the training data set.

Using fuzzy c-means algorithm on a set of training data, we will have two clusters. Each cluster is labeled +1 or -1 base on the center point of each cluster. Check on each data point of clusters, remove from the training data set if those data points' labels are not the same with cluster's label.

$(x_i, y_i)$: vector $x_i$ is labeled $y_i$
n: number of training data vector

X is a set of training data:
  $X = \{(x_i,y_i)\}$, i = 1,…, n
A, B are output sets of fuzzy c-means algorithm. AUB=X
  Input: $X = \{(x_i,y_i)\}$
  Output:
    $A = \{(x_{ai}, y_{ai})\}$; $(x+_a, y+_a)$: vector center of A
    $B = \{(x_{bi}, y_{bi})\}$; $(x+_b, y+_b)$: vector center of B
    For each $(x_{ai}, y_{ai})$ in A
      if $y_{ai} \neq y+_a$
        $X = X\backslash\{(x_{ai}, y_{ai})\}$
    For each $(x_{bi}, y_{bi})$ in B
      if $y_{bi} \neq y+_b$
$X = X\backslash\{(x_{bi}, y_{bi})\}$

## 4. Multiclass classification

In multiclass classification, we apply fuzzy multiclass classification method (FOAO) ([5]) According to FOAO, n(n-1)/2 classifiers are built by catching in pairs and then combining the results of these classifiers to determine the final classification result. We use FCSVM to build these classifiers.

FOAO is based on OAO strategy and combined with a membership function to determine classification result when vector x is unclassified by the OAO strategy.

The decision function of i-th class and j-th class in the OAO strategy is following:

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

According to the optimal hyperplan $D_{ij}(x) = 0 (i \neq j)$, the membership functions are defined:

$$m_{ij}(x) = \begin{cases} 1 & with\ D_{ij}(x) \geq 1, \\ D_{ij}(x) & other \end{cases}$$

From $m_{ij}(x)(j \neq i, j = 1,...,n)$, the i-th x vector membership function is defined as follows:

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

The above formula is equivalent to:

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

Now x is classified to i-th class by using the formula:

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

## 5. Experiments

We implemented a program collecting and classifying information in the industry field that includes 5 sub-industries: textile, mechanics, electricity, petroleum and other industries. We have test this program on webpages that contain the industry information.

To show the effectiveness of the proposed method, we compare the performance of FSVM with FCSVM.

Table 1 lists the experiment results of the FSVM and FCSVM classifiers with 6000 texts training set and 2000 texts verifying set; RBF kernels ($\gamma = 0.8$).

Table 1: The experiment results of FSVM and FCSVM classifiers.

| No | Classifiers | F-score | |
|---|---|---|---|
| | | FSVM | FCSVM |
| 1 | Electricity – Mechanics | 0.917 | 0.920 |
| 2 | Electricity – Textile | 0.890 | 0.911 |
| 3 | Electricity – Petroleum | 0.910 | 0.900 |
| 4 | Electricity – Others | 0.920 | 0.888 |
| 5 | Mechanics – Textile | 0.870 | 0.915 |
| 6 | Mechanics – Petroleum | 0.917 | 0.917 |
| 7 | Mechanics – Others | 0.911 | 0.900 |
| 8 | Textile – Petroleum | 0.913 | 0.926 |
| 9 | Textile – Others | 0.934 | 0.947 |
| 10 | Petroleum – Others | 0.891 | 0.899 |
| Average | | **0.907** | **0.914** |

Table 2: The program experiment results.

| No | News Website | The percentage of right classification industry news | |
|---|---|---|---|
| | | FSVM | FCSVM |
| 1 | Ministry of Industry and Trade | 84.00 | 88.00 |
| 2 | Trade newspaper | 84.91 | 88.68 |
| 3 | Investment magazine | 84.21 | 89.47 |
| 4 | VietNam economical journal | 87.76 | 85.71 |
| 5 | VietNam trade and industry Chamber | 88.46 | 92.31 |
| 6 | VietNam press agency | 87.18 | 84.62 |
| 7 | TuoiTre newspaper | 85.71 | 85.71 |
| 8 | Industry Consultant Center HCMC | 85.00 | 86.67 |
| 9 | ThanhNien newspaper | 86.49 | 83.78 |
| 10 | Saigon GiaiPhong | 89.29 | 92.86 |
| **Average** | | **86.30** | **87.78** |

Table 2 lists the experiment results of program in comparing two algorithms FSVM, FCSVM combined with fuzzy multiclass classification for multiclass classification. FCSVM are not always superior to FSVM. But overall results when using FCSVM are higher than FSVM.

## 6. Conclusions

In this paper, we combine support vector machines with fuzzy c-means in an application automatically extracting and classifying information on the Internet. We use matching algorithm to extract information exactly on WebPages in order to help classification better. The information is classified by support vector machines combined with fuzzy c-means. We combine FCSVM with fuzzy multiclass classification. The proposed method gives a good result when compared with FSVM especially in case there's much noise in training data.

## 7. References

[1] Nguyen Thi Kim Ngan (2004), Vietnam Language Text Classification by support vector machines method, Politecnical University Hanoi City.

[2] Le Phu (2005), Automatically Extracting text blocks that contains main information on e-newspapers, Politecnical University HCM City.

[3] Blanzieri, E., and Melgani, F. (2007), Instance-Based Spam Filtering Using SVM Nearest Neighbor Classification, Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference, May 7-9, 2007, Key West, Florida, USA. 2007.

[4] Lin CF, Wang SD (2002), Fuzzy support vector machines, IEEE Trans Neural Netw 13(2):464–471.

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

[6] T.Joachims (1998), 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.

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

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

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

**Vu Thanh Nguyen**
The author born in 1969 in Da Nang, VietNam. He graduated University of Odessa (USSR), in 1992, specialized in Information Technology. He postgraduated on doctoral thesis in 1996 at the Academy of Science of Russia, specialized in IT. Now he is the Dean of Software Engineering of University of Information Technology, VietNam National University HoChiMinh City.
Research: Knowledge Engineering, Information Systems and software Engineering.