New fuzzy multi-class method to train SVM classifier

Taoufik Guernine
Département d'Informatique, Faculté de Sciences
Université de Sherbrooke
Sherbrooke, Canada
Taoufik.guernine@usherbrooke.ca

Kacem Zeroual
Département d'Informatique, Faculté de Sciences
Université de Sherbrooke
Sherbrooke, Canada
Kacem.zeroual@usherbrooke.ca

***Abstract*—In this paper we present a new classification method based on Support Vector Machine (SVM) to treat multi-class problems. In the context of multi-class problems, we have to separate large number of classes. SVM becomes an important machine learning tool to handle multi-class problems. Usually, SVM classifiers are implemented to deal with binary classification problems. In order to handle multi-class problems, we present a new method that builds dynamically a hierarchical structure from training data. Our multi-class method is based on three main concepts : Hierarchical classification, Fuzzy logic and SVM. We combine multiple binary SVMs to solve multi-class problems. The proposed method divides the original problem into sub-problems in order to reduce its complexity.**

***Keywords*-Classification; SVM; Fuzzy logic;**

I. Introduction

Solving multi-class problems with high performance is a challenging problem because there is an important increasing processing of data in databases. Until now, multi-class problems remain among the primary worry in the field of classification. Furthermore, the manual classification is not able to keep up with the growth of data. An automatic classification becomes necessary. Many machine learning methods and statistical techniques has been proposed : Decision trees [1], Nearest neighbor classifiers [2], Bayesian models [3] and Support Vector Machine [4].

Unlike the other classifiers, SVM classifiers find an optimal hyperplane maximizing the marge between two classes. Generally, SVM is used for binary classification but its extension to multi-class problems remains an open research topic [5]. There are two techniques for extending SVM to multi-class problems. The first technique consists in resolving optimization problems where the whole training data set is used [6]. This technique requires huge time to train all the data set. The second technique consists in constructing binary classifiers from the root until leaves [7]. The original problem is subdivided into simple binary sub-problems. Each sub-problem contains a small portion of data and is less complex than the original problem. In this paper, we are interested in subdividing the original problem into binary sub-problems. We propose a new classification method based on SVM to treat multi-class problems. The proposed method uses a fuzzy hierarchical structure to extract relationships between objects. It introduces the transitive closure measure to discover fuzzy similarity between objects. Training data set of SVM obtained a priori by the transitive closure Min-Max assures discriminating between positive and negative classes. Introducing membership values extracted from transitive closure matrix to SVM optimization problem allows high performance.

The remainder of this paper is organized as follows. In section II, we provide an overview of related works. In section III, we give a brief review of SVM. In section IV, we describe the fuzzy hierarchical classification method. In section V, we present our experimental results. Our future research works are presented in section VI.

II. Related Works

The most important issue in multi-class problems is the existence of confusion classes [8]. The hierarchical structure is among techniques used to solve the confusion classes. The multi-class problems based on SVM is mainly related to hierarchical multi-class pattern recognition problems. Most of recent works used hierarchical structure to address the classification task. In [9], they proposed a new classification algorithm based on a hierarchical structure. The algorithm consists of the following stages : (i) generating category information tree (ii) hierarchical feature propagation (iii) feature selection of category information and (iv) single path traversal. The proposed hierarchical classification system allows adding new categories as required, organizing the web pages into a tree structure and classifying web pages by searching through only one path of the tree structure. In [10], authors explore a hierarchical classification to classify heterogenous collections of the web content. They used hierarchical structure in order to distinguish a second level category from other categories within the same top level. They introduced SVM at each level to obtain a hierarchy. In [11], authors added fuzzy membership values to each input data and reformulate the SVM optimization problem. The membership values make more contribution in the classification process. The proposed fuzzy SVM can solve different kinds of multi-class problems. In [12], the fuzzy set theory is introduced in the classifying module. The authors proposed a One-against-all fuzzy SVM (OAA-

 ISBN:978-1-61208-115-1

FSVM) classifier to implement a multi-class classification system. The empirical results obtained by the proposed system show that OAA-FSVM method performs better than OAA-SVM method.

III. SUPPORT VECTOR MACHINE

In this section we give a brief review of Support Vector Machine. We present respectively binary and multi-class classification.

A. Binary classification

Generally, SVM classifiers are designed to solve binary classification problem [13]. It consists in minimizing the empirical classification error and finding optimal hyperplane with large margin [14]. Suppose a data set $(x_i, y_i) : (i = 1, ..., n)$, where x_i corresponds to the attribute set for the i^{th} element. Let $y_i \in \{-1, +1\}$ be a labelled class. The optimal hyperplane can be found by minimizing the margin w in equation III-A :

$$(P) = \begin{cases} Min\frac{1}{2}\|w\|^2 \\ y_i(wx_i + b) \geq 1 : i \in 1, n : \forall x \in R^n \end{cases}$$

Where w and b are parameters of the model. The solution of optimization problem is given by Lagrangian :

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i[y_i(wx_i + b) - 1] \tag{1}$$

Where α_i are called the Lagrange multiplier. We can simplify the problem given by equation 1 as follows :

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{ij}^{n} \alpha_i\alpha_j y_i y_j x_i x_j \tag{2}$$

In several cases, linear solutions could not solve the optimization problem. In this situation, a non linear separator is required. The formulation of the problem is given bellow :

$$f(x) = \begin{cases} f(x) = wx_i + b \geq (1 - \xi_i) \quad if \quad y_i = 1 \\ wx_i + b \leq (1 - \xi_i) \quad if \quad y_i = -1 \\ \xi_i > 0, \forall i \end{cases}$$

The objective function will change as follows :

$$f(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i)^\kappa \tag{3}$$

Where C and ξ_i are specified by the user and represent the penalty of mis-classification. The Lagrangian is written as follows :

$$L_p = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i) - \sum_{i=1}^{n}\alpha_i[y_i(wx_i + b) - 1 + \xi_i] - \sum_{i=1}^{n}\mu_i\xi_i \tag{4}$$

We can however, simplify the problem given by equation 4 as follows :

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{ij}^{n} \alpha_i\alpha_j y_i y_j x_i x_j \tag{5}$$

The problem given by equation 5 becomes identical to the linear discrimination problem given by equation 2.

B. Multi-class classification

In order to treat multi-class problems by constructing binary problems, several methods have been proposed. The are three methods developed to deal with multi-class problems using SVM classifier at each node :

1) One-against-one method: To resolve multi-class problem, one-against-one method requires one classifier SVM_{ij} for each pair of classes (i, j). It builds $[n(n-1)/2]$ classifiers for n-class classification problem. During the test phase, the test set is evaluated by all SVM_{ij}.

Let $E = (x_i, y_i)_{i=1,n}$, be a training data set, where $x_i \in R^n$ and $y_i \in \{1, 2, ..., k\}$. For k class problem, the optimization problem to construct SVM_{ij} that separate two classes C_i and C_j is given as follows :

$$(P) = \begin{cases} \min_{w^i, b^i, \xi^i} \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \xi_t^{ij} \\ (w^{ij})^T\phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij} : y_j = 1 \\ (w^{ij})^T\phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij} : y_j \neq 1 \\ \xi_t^{ij} \geq 0 : j = 1, ..., k. \end{cases} \tag{6}$$

To determine the decision function $(f_{ij}(x) = Sgn(w_{ij}x + b_{ij}))$ which separates classes C_i and C_j, we use Max-Win strategy :

$$Sgn(x) = \begin{cases} +1 : x > 0 \\ -1 : x \leq 0 \end{cases}$$

$$x \in \begin{cases} C_i : f_{ij}(x) = 1 \\ C_j : f_{ij}(x) = -1 \end{cases}$$

The process of Max-Win strategy is given as follows :

- For each x_i :

$$f_{ij}(x) = \sum_{j \neq i, j=1}^{k} Sgn(f_{ij}(x)) \tag{7}$$

- The class of x_i is obtained by :

$$arg \max_{i:1,...,k} f_i(x) \tag{8}$$

2) One-against-all method: The one-against-all method is simple and efficient. It requires n classifiers $SVM_i : (i = 1, n)$, for n-class classification problem. During the test phase, the test set is evaluated by the SVM_i. SVM_i which shows highest decision value is chosen.

Let $E = \{(x_1, y_{1j}), (x_2, y_{2j})..., (x_l, y_{lj})\}$ be a training data set, where $x_{i_{(i=1,l)}}$ represents the i^{th} observation and $y_{ij_{(j=1,k)}}$ represents the j^{th} class of the i^{th} observation. For k class problem, the formulation of the j^{th} SVM is given as follows :

$$(P) = \begin{cases} \min\limits_{w^i, b^i, \xi^i} \frac{1}{2}(w^j)^T w^j + C \sum\limits_{j=1}^{l} \xi_i^j \\ (w^j)^T \phi(x_i) + b^j \geq 1 - \xi_i^j : y_j = j \\ (w^j)^T \phi(x_i) + b^j \leq -1 + \xi_i^j : y_j \neq j \\ \xi_i^j \geq 0 : i = 1, l; j = 1, k \end{cases} \tag{9}$$

We solve the problem in (9) and obtain k decision functions :

$$(P) = \begin{cases} (w^1)^T \phi(x_i) + b^1, \\ . \\ . \\ (w^k)^T \phi(x_i) + b^k \end{cases} \tag{10}$$

The class of x_i is obtained as follows :

$$Class(x) = \arg \max_{(i=1,\ldots,l)} ((w^j)^T \phi(x_i) + b^j). \tag{11}$$

3) Directed Acyclic Graph SVM (DAGSVM): The DAGSVM method constructs also $[n(n-1)/2]$ classifiers SVM_{ij}. During the test phase, it creates a list of all candidates classes. At each test, the class that obtained negative score is eliminated from the list.

IV. SVM FUZZY HIERARCHICAL CLASSIFICATION METHOD

The new method we propose in this paper supplies an alternative to the three methods : One-against-one, One-against-all and DAGSVM. Our method is based on a fuzzy hierarchical classification technique we developed for the specification software reuse [15]. It provides also advantages to treat hierarchical multi-class problems. The method we propose consists of three steps : (A) Training data set compression by K-Mean (B) Fuzzy hierarchical classification building and (C) Introducing membership function for training SVM.

A. Training data set compression

Several works focused on reducing the number of training data set of SVM [16]. The first step in our method is compressing training data set of SVM. We apply basic K-Mean algorithm in order to regroup similar data in the same cluster and reduce time spent in training data set of SVM. The goal of this step is expressed by an objective function that depends on the proximities of the points to their centroids. To assign each object to the closest centroid, we apply equation 12 :

$$g_i = \frac{1}{m_i} \sum_{x \in C_i} x \tag{12}$$

Where g_i represents the centroid of cluster C_i, m_i represents the number of objects in the i^{th} cluster and x is an observation.

In order to measure the quality of clustering, we use the sum of the squared error (SSE), given by :

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist(g_i, x)^2 \tag{13}$$

Where k represents the number of clusters.

B. Fuzzy hierarchical classification building

1) Similarity measure: The notion of a distance between x and y has long been used in many contexts as a measure of similarity or dissimilarity between a set's elements. In this work, we define a relative generalized Hamming distance δ to compute similarity between clusters which is defined by :

$$\delta(\varsigma_i, \varsigma_j) = \frac{1}{n} \times d(\varsigma_i, \varsigma_j) = \frac{1}{n} \sum_{i=1}^{n} |\mu_{\varsigma_i}(x_i) - \mu_{\varsigma_j}(x_i)| \tag{14}$$

Where n represents the number of clusters and $d(\varsigma_i, \varsigma_j)$ is the Hamming distance between clusters ς_i and ς_j.

Since $\mu_{\varsigma_i}(x_i)$ and $\mu_{\varsigma_j}(x_i) \in [0,1]$, $\forall\ i = 1, n \Rightarrow$

$$0 \leq \delta(\varsigma_i, \varsigma_j) \leq 1. \tag{15}$$

2) Fuzzy subsets: Let K be a universe of discourse, $A \subset K$, and $K = \{x_i\}$. An element x of K belonging to A is defined as : $x \in A$. Let $\mu_A(x)$ be a characteristic function whose value indicates whether x belongs to A according to :

$$\mu_A(x) = \begin{cases} 1 & if \quad x \in A \\ 0 & if \quad x \notin A \end{cases} \tag{16}$$

The characteristic function $\mu_A(x)$ takes its values in the interval [0,1]. It is defined as a mapping :

$$\mu_A(x) : A \rightarrow \{0, 1\} \tag{17}$$

The fuzzy logic is based on partial membership function. An object is belonging to one or more than a class in the same case. Let A be a sub set, defined by its membership function μ_A. The membership function $\mu_A(x)$ of an object x used in fuzzy set theory is defined as follows : An object x does not belong to class C if the membership function $\mu_C(x) = 0$, belongs a little to class C if $\mu_C(x)$ border to 0, belongs enough to class C if $\mu_C(x)$ does not border to 0 nor to 1, belongs strongly to class C if $\mu_C(x)$ border to 1 and belongs completely to class C if $\mu_C(x) = 1$.

 ISBN:978-1-61208-115-1

3) Fuzzy operators: Let A and B be fuzzy subsets of universe K. The fuzzy operators on the fuzzy subset A and B of K are given as follows :

- **Intersection operator (AND)**
  The membership function used by [17] to define the set ($A \cap B$), is given by the minimum of membership functions μ_A and μ_B as follows :

$$\forall x \in X : \mu_{A \cap B}(x) = min\{\mu_A(x), \mu_B(x)\}. \tag{18}$$

- **Union operator (OR)**
  The membership function defines the set ($A \cup B$) is given by the maximum of membership functions μ_A and μ_B as follows :

$$\forall x \in X : \mu_{A \cup B}(x) = max\{\mu_A(x), \mu_B(x)\}. \tag{19}$$

4) Transitive closure of a fuzzy relation: To extract ambiguous relationships between objects, we used the theory of fuzzy sets [17]. It is defined by their memberships function. In our work, we used Min-Max transitivity relation to find fuzzy relationships between objects :

$\forall\ x,\ y,\ z \in K\times\ K \times K$:

$$\mu_R(x,z) \leq Min_y[Max(\mu_R(x,y), \mu_R(y,z))] \tag{20}$$

We compute the transitive closure Min-Max given by equation 20 until we obtain transitive closure Γ equals to Γ=$R^{\kappa-1}$=R^{κ} at κ levels. This equality assures the existence of a hierarchy. This relation gives the transitive distance Min-Max which locates the level of each objects and find the short link between these objects. Let $C_i = \{x_{i1}, x_{i2}..., x_{in}\}$ and $C_j = \{x_{1j}, x_{2j}, ..., x_{nj}\}$ be two clusters obtained by the similarity matrix. The fuzzy shortest link between two clusters is given as follows : $\Gamma_{ij} = \vee[(x_{i1} \wedge x_{1j}), (x_{i2} \wedge x_{2j}), ..., (x_{in} \wedge x_{nj})]$.

C. Introducing membership function for training SVM

In this step, we train fuzzy SVM at each node of the hierarchy to subdivide the original problem into binary sub-problems.

Let M be a set of classes $C = \{c_1, c_2, ..., c_k\}$, where k is the number of clusters obtained by K-Mean in the first step ($k \leq n$).

First, we compute the average transitive closure of all classes from the transitive closure matrix by the equation :

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\Gamma_{ij} \tag{21}$$

Where n represents the number of values of Γ_{ij} in transitive closure matrix and Γ_{ij} represents fuzzy similarity value between C_i and C_j that are obtained by transitive closure.

Second, we compute the average of transitive closures of each class according to the following equation :

$$\upsilon_i = \frac{1}{k}\sum_{j=1}^{n}\Gamma_{ij}, j:1,n \tag{22}$$

The fuzzy membership υ_i, which is the average similarity between C_i and the rest $(k-1)$ of classes, is extracted from the transitive closure matrix.

Suppose that $E = \{(C_1, y_1, \upsilon_1), ..., (C_k, y_k, \upsilon_k)\}$ a set of training data with associated membership, where $C_i \in R^k$, $y_i = \{-1, +1\}$ and $0 \leq \upsilon_i \leq 1$.

In our work and in order to handle multi-class with high precision, we introduced fuzzy membership function in the training SVM step. Each row i of the transitive closure matrix defines the membership between class i and the others classes. To construct positive and negative classes, we compute for each class C_i the membership value υ_i. At each node of the hierarchy, the problem can be defined as follows :

$$SVM = \begin{cases} \{C_i\} \cup SVM_{ij}^{+} : \upsilon_i > \bar{X} \\ \{C_i\} \cup SVM_{ij}^{-} : \upsilon_i \leq \bar{X} \end{cases} \tag{23}$$

The optimization problem given by our fuzzy SVM in (23) is given as follows :

$$\begin{cases} \frac{1}{2}w^T \cdot w + C\sum_{i=1}^{k}\upsilon_i\xi_i \\ y_i(w \cdot x_i + b) > 1 - \upsilon_i\xi_i \\ \upsilon_i\xi_i \geq 0 : i = 1, ..., k \end{cases} \tag{24}$$

Where C, ξ_i represent the penalties of mis-classification and $\upsilon_i\xi_i$ represents error of classification with different weights.

Using the Lagrangian multiplier, the problem is given as follows :

$$\begin{cases} Max : w(\alpha) = \sum_{i=1}^{k}\alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \\ Subject : \sum_{i=1}^{k} y_i x_i = 0, 0 \leq \alpha_i \leq \upsilon_i C : i = 1, k \end{cases} \tag{25}$$

We repeat the process at each node of the hierarchy until reaching leaves containing only one class. Consequently, we obtain a descendant hierarchical classification represented by a succession of classes. Each class contains similar objects. The advantage of our method is that training data set of SVM obtained a priori by the transitive closure assures discriminating between positive and negative classes.

 ISBN:978-1-61208-115-1

V. Experimental results

A. Data

In this paper, we compareded the performance of our method with those of the methods : One-against-one, One-against-all and DAGSVM. We used three different problems available in [18]. The first problem is Iris database which contains 150 records grouped equally in three classes. The second problem is Glass database which contains 214 records distributed in six classes. The third problem is Letter database which contains 16000 records distributed in twenty six classes. We give detail of the three problems in Table I.

Table I
Problem detail

Problem	Data	Class	Attributes
Iris	150	3	4
Glass	214	6	9
Letter	16000	26	16

B. Experimental

- **Compression step**

To show how the compression step is usefull, we conducted two experiments. The first experiment consists in applying K-Mean to training data set step with the original data set replaced by clusters centroid. In the second experiment, we apply our method,without calling K-Mean. Table II shows results given by the two experiments.

Table II
Compression step influence on SVM-CHF performance

Data	With K-Mean			Without K-Mean		
	# SVM	Training time	Accuracy	# SVM	Training time	Accuracy
Iris	**2**	**0.021**	98.00	3	0.05	**98.23**
Glass	**4**	**0.05**	77.63	6	11	**78.10**
Lettre	**21**	**110**	98.35	24	255	**98.45**

In the first step, our method performs better in number of SVMs and training time criteria. Using K-Mean algorithm reduced automatically the number of SVMs and cost training time. In the second step we used the original data set wich allows slightly better accuracy compared with accuracy result obtained in the first step. Since the two first criteria in the classification domain are very important, we introduced the K-Mean algorithm in the process of our method.

- **Kernel function**

In order to choose the best kernel function of each problem, we tested different kernel functions : Polynomial (d=2,3,...,8) and RBF ($\gamma = 0.1, 0.2, ..., 1$). We choose only results where SVM performs well. The results are given in Table III.

For **Iris** (k=3) and **Glass** (k=6) problems, polynomial function gives best results. For **Letter** (k=26) problem, RBF function performs best. In our case, polynomial function

Table III
Accuracy obtained by polynomial and RBF kernel functions

Data	$SVM_{Poly:d}$				$SVM_{RB:\gamma}$			
	2	4	6	8	0.1	0.2	0.4	1.0
Iris	**0.98**	0.95	0.94	0.94	0.97	0.96	0.90	0.96
Glass	0.66	**0.77**	0.76	0.69	0.66	0.69	0.72	0.65
Letter	0.54	0.67	0.88	0.87	0.78	0.93	**0.98**	0.92

performs better when the number of classes is small. High accuracy is obtained when $C = 2^{10}$, $C = 2^{11}$ and $C = 2^{11}$ for Iris, Glass and Letter problems respectively. The proposed method proved high performance for the three problems (Iris : 98.00%, Glass : 77.63% and Letter : 98.35%).

- **Accuracy comparison**

We use accuracy criterion to evaluate our results with results obtained by methods : One-against-one, One-against-all and DAGSVM. To obtain high accuracy, we tested our method with different values of C : (2^2,..., 2^{12}). Accuracy is obtained from confusion matrix. Our accuracy comparison results are compared with : One-against-one, One-against-all and DAGSVM (see Table IV). The proposed method proved high performance for the three problems.

Table IV
Accuracy comparison

Problem	One-gainst-one	One-gainst-rest	DAGSVM	Our Proposed Method
Iris	97.33	96.67	97.36	**98.00**
Glass	71.49	71.96	72.22	**77.63**
Letter	97.98	97.88	96.73	**98.35**

- **The fuzzy membership function influences on the classifier performance**

In this section, we tested the influence of the fuzzy membership function on the classifier performance. We varied υ_i in the range from 0.1 to 0.8. Figure 1 shows that the high performance is obtained when υ_i is equal to 0.31, 0.22 and 0.32 for problems Iris, Glass and Letter respectively. These fuzzy values are extracted from the transitive closure matrix. The values υ_i are introduced to train SVM. Choosing υ_i from transitive closure matrix allows our method to perform better.

VI. Conclusion

In this paper, we proposed a new fuzzy SVM hierarchical method to handle multi-class problems. The fuzzy hierarchical structure consists in subdividing the original problem into simple binary problems. Our method takes its advantage from using fuzzy hierarchical classification and fuzzy Support Vector Machine. Furthermore, it has the advantage of using only values from the similarity matrix

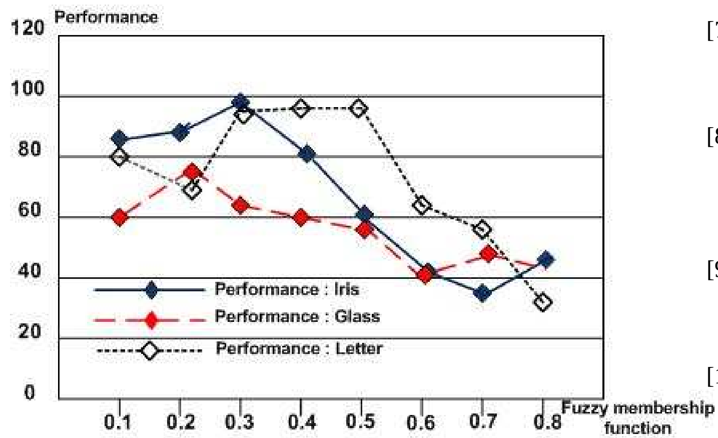

Figure 1. Fuzzy membership influence.

for the SVM training rather than using values randomly. Similarity matrix assures a priori separate classes in the hierarchy.

Unlike other classification methods, our method requires a number less than or equal to $(k-1)$ SVM classifiers from the root until leaves (see Table II). The number of SVM required reduce automatically the cost of training SVM time.

In this work, we find that introducing the membership values extracted from transitive closure matrix to SVM optimization problem gives a high accuracy.

Our future works consists in adapting our method to video sequencing problem in order to extract fuzzy relations between objects. Moreover, we will create a new dynamic kernel function to handle automatically classification process.

REFERENCES

[1] S. Weiss and C. Apte and F. Damerau and D. Johnson and F. Oles and H. Goetz, *Maximizing text mining performance*. IEEE Intelligent Systems : 2–8, 1999.

[2] X. Xie, *A validity measure for fuzzy clustering*. IEEE Transactions Pattern analysis and Machine Intelligence : 841-847, 1991.

[3] D. Koller and M. Sahami, *Hierarchically classifying documents using very few words*. Proceedings of the 14^{th} International Conference on Machine Learning, Nashville : 171–178, 1997.

[4] T. Joachims, *Text Categorization with Support Vector Machines : Learning with many relevant features*. Proceedings of the 10^{th} European conference on machine learning : 1998.

[5] Y. Guermeur and A. Elisseeff and H. Paugam, *A new multiclass SVM based on a uniform convergence result*. IJCNN : 183-188, 2000.

[6] C. Hsu and C. Lin, *A comparison of methods for multi-class support vector machines*. IEEE Trans Neural Networks : 415–425, 2002.

[7] G. Zhang, *Support Vector Machine with Huffman tree architecture for multi-class classification*. Lecture Notes in computer Science : 24–33, 2005.

[8] F. Schwenker, *Hierarchical Support Vector Machines for Multi-Class Pattern Recognition*. The 4^{th} International Conference on knowledge-Based Intelligent Engineering Systems, Allied Technologies, Brighton, UK : 2000.

[9] X. PENG and B. CHOI, *Automatic Web Page Classification in a Dynamic and Hierarchical Way*. IEEE International Conference on Data Mining : 386–393, 2002.

[10] S. Dumais and H. Chen, *Hierarchical Classification of Web Content*. Proceedings of the 23^{rd} annual international ACM SIGIR conference on Research and development in information retrieval, Athens, Greece : 256–263, 2000.

[11] L. Chun-Fu and W. Sheng-De, *Fuzzy Support Vector Machines*. IEEE Transactions On Neural Networks : 464–471, 2000.

[12] T. Wang and H. Chiang. *Fuzzy support vector machine for multi-class text categorization*. Information Processing and Management : 914–929, 2007.

[13] V. Vapnik, *Statistical Learning Theory*. John Wiley, Sons : 1998.

[14] S. Tuffery, *Data Mining et statistique decisionnelle : L'intelligence des donnes*. Technip. Paris : 2007.

[15] J. Nkoghe and K. Zeroual and W. Shengrui, *Specification Reuse : A Fuzzy Approach*. International Joint Conference on Artificial Intelligence : 1995.

[16] B. Xiaojuan and S. Qilong and C. Hao and T. Xuyan, *Compression method based on training data set*. Journal of Systems Engineering and Electronics : 198–201, 2008.

[17] L. Zadeh, *Fuzzy Sets*. Journal of Information and Control : 338-353, 1965.

[18] C. L. blake and C. J. Merz, *UCI repository of machine learning databases*. Available at ftp//ftp.ics.uci.edu/pub/machine-learning-databases, 1998.

 ISBN:978-1-61208-115-1