Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing

A Hybrid Text Classification model based on Rough Sets and Genetic Algorithms

Xiaoyue Wang, Zhen Hua, Rujiang Bai
Shandong University of Technology Library Zibo 255049, China
E-mail{wangxy, huazhen, brj}@sdut.edu.cn

Abstract

Automatic categorization of documents into pre-defined taxonomies is a crucial step in data mining and knowledge discovery. Standard machine learning techniques like support vector machines(SVM) and related large margin methods have been successfully applied for this task. Unfortunately, the high dimensionality of input feature vectors impacts on the classification speed. The kernel parameters setting for SVM in a training process impacts on the classification accuracy. Feature selection is another factor that impacts classification accuracy. The objective of this work is to reduce the dimension of feature vectors, optimizing the parameters to improve the SVM classification accuracy and speed. In order to improve classification speed we spent rough sets theory to reduce the feature vector space. We present a genetic algorithm approach for feature selection and parameters optimization to improve classification accuracy. Experimental results indicate our method is more effective than traditional SVM methods and other traditional methods.

1. Introduction

Due to the rapid growth in textual data, automatic methods for organizing the data are needed. Automatic document categorization is one of these methods. It automatically assigns the documents to a set of pre-defined classes based on its textual content. Document categorization is a crucial and well-proven instrument for organizing large volumes of textual information. There are many classification methods for textual data. A support vector machine, named SVM, was suggested by Vapnik (1995) and have recently been used in a range of problems including pattern recognition (Pontil and Verri, 1998), bioinformatics (Yu, Ostrouchov, Geist, & Samatova, 1999), and text categorization (Joachims, 1998).

When using SVM, three problems are confronted: (1)how to reduce the high dimension of feature vectors; (2)how to choose the optimal input feature subset for SVM, (3)and how to set the best kernel parameters. These three problems are crucial, because the feature subset choice influences the appropriate kernel parameters and vice versa (Frohlich and Chapelle, 2003). Therefore, obtaining the optimal feature subset and SVM parameters is important.

In the literature, only a few algorithms have been proposed for SVM feature selection (Bradley, Mangasarian, & Street, 1998; Bradley and Mangasarian, 1998; Weston et al., 2001; Guyon, Weston, Barnhill, & Bapnik, 2002; Mao, 2004). Some other genetic algorithms(GA)-based feature selection methods were proposed (Raymer, Punch, Goodman, Kuhn, & Jain, 2000; Yang and Honavar, 1998; Salcedo-Sanz, Prado-Cumplido, Perez-Cruz, & Bousono-Calzon, 2002). However, these papers focused on feature selection and did not deal with attribute reduce and parameters optimization for the SVM classifier.

In addition to the feature selection, proper parameters setting can improve the SVM classification accuracy. The parameters that should be optimized include penalty parameter C and the kernel function parameters such as the gamma (γ) for the radial basis function (RBF) kernel. To design a SVM, one must choose a kernel function, set the kernel parameters and determine a soft margin constant C (penalty parameter). The Grid algorithm is an alternative to finding the best C and gamma when using the RBF kernel function. However, this method is time consuming and does not perform well (Hsu and Lin, 2002; LaValle and Branicky, 2002).

In order to improve SVM classification speed and accuracy, we proposed a new method. First, Rough Sets Theory(RST) is used to reduce feature vectors after data preprocess. Second, using genetic algorithms to select feature and optimize the parameter for SVM.

978-0-7695-3263-9/08 $25.00 © 2008 IEEE
DOI 10.1109/SNPD.2008.142

This paper is organized as follows: a brief introduction to the SVM is given in Section 2. Section 3 describes Rough Sets Theory. Section 4 describes basic GA concepts. Section 5 describes the system overview. Include:(1)algorithm of RST-based attribute reduce;(2)GA-based feature selection and parameter optimization. Section 6 presents the experimental results from using the proposed method to classify test datasets. Section 7 draws a general conclusion and describes the future work.

2. Brief introduction of Support vector machine[1~3]

The primary idea of support vector machine (SVM) is using a high dimension space to find a hyper plane to do binary division, where the achieved error rate is minimum. An SVM can handle the problem of linear inseparability.

An SVM uses a portion of the data to train the system and finds several support vectors that represent training data. These support vectors will be formed into a model by the SVM, representing a category. According this model, the SVM will classify a given unknown document by the following classification decision formula

$$(x_i, y_i), \cdots, (x_n, y_n), x \in R^m, y \in \{+1, -1\} \tag{1}$$

Where $(x_i, y_i), \cdots, (x_n, y_n)$ are training samples, n is the number of samples, m is the input dimension, and y belongs to the category of +1 or -1, respectively.

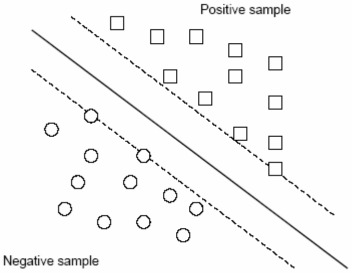


Fig. 1. The hyper plane of SVM.

In a linear problem, a hyper plane is divided into two categories. Fig. 1 shows a high dimension space divided into two categories by a hyper plane. The hyper plane formula is: (w • x)+b=0.

The classification formula is:

$$(w \bullet x_i) + b > 0 \ if \ y_i = +1 \quad (w \bullet x_i) + b < 0 \ if \ y_i = -1 \tag{2}$$

However, for many problems it is not easy to find a hyper plane to classify the data. The SVM has several kernel functions that users can apply to solve different problems, such as radial basis function, sigmoid, Polynomial etc.

Radial basis function kernel is:

$$k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{3}$$

3. Rough Sets Theory

The rough sets theory has been developed for knowledge discovery in databases and experimental data sets [4~8]. The rough sets theory deals with information represented by a table called information system. This table consists of objects (or cases) and attributes. The entries in the table are the categorical values of the features and possibly categories. It also denoted to attribute reduce.

An information system is a 4-tuple $S = \langle U, A, V, f \rangle$, where U is a finite set of objects, called the universe, A is a finite set of attributes.$V = U_{a \in A} V_a$ is a domain of attribute a, and $f := U \times A \rightarrow V$ is called an information function such that $f(x, a) \in V_a$,for $\forall a \in A, \forall x \in U$.

In the classification problems, an information system is also seen as a decision table assuming that $A = C \cup D$ and $C \cap D = \phi$, where C is a set of condition attributes and D is a set of decision attributes.

Let $S = \langle U, A, V, f \rangle$ be an information system, every $P \subseteq A$ generates a indiscernibility relation *IND(P)* on *U*, which is defined as follows:

$$\mathrm{IND(P)} = \{(\mathrm{x}, \mathrm{y}) \in \mathrm{U} \times \mathrm{U} : \mathrm{f(x,a)} = \mathrm{f(y,a)}, \forall \mathrm{a} \in \mathrm{P}\} \tag{4}$$

$U / IND(P) = \{C_1, C_2, \cdots C_k,\}$ is a partition of *U* by *P*, every C_i is an equivalence class. For $\forall x \in U$, the equivalence class of x in relation $U / IND(P)$ is defined as follows:

$$[x]_{IND(P)} = \{y \in U : f(y, a) = f(x, a), \forall a \in P\} \tag{5}$$

Let $S = \langle U, C \cup D, V, f \rangle$be a decision table, the set of attributes $P(P \subseteq C)$ is a reduction of attributes C, which satisfies the following conditions:

$$\gamma_P(D) = \gamma_C(D) \text{ and } \gamma_P(D) \neq \gamma_{P'}(D), \forall P' \subset P \tag{6}$$

A reduction of condition attributes C is a subset that can discern decision classes with the same discriminating capability as C, and none of the attributes in the reduction can be eliminated without decreasing its discriminating capability.

4. Genetic Algorithms[9~11]

GAs(Genetic Algorithms) are stochastic and evolutionary search techniques based on the principles of biological evolution, natural selection, and genetic recombination. They simulate the principle of ‘survival of the fittest’ in a population of potential solutions known as chromosomes. Each chromosome represents one possible solution to the problem or a rule in a classification. The population evolves over time through a process of competition whereby the fitness of each chromosome is evaluated using a fitness function. During each generation, a new population of chromosomes is formed in two steps. First, the chromosomes in the current population are selected to reproduce on the basis of their relative fitness. Second, the selected chromosomes are recombined using idealized genetic operators, namely crossover and mutation, to form a new set of chromosomes that are to be evaluated as the new solution of the problem. GAs are conceptually simple but computationally powerful. They are used to solve a wide variety of problems, particularly in the areas of optimization and machine learning (Davis, 1991; Grefenstette, 1994).

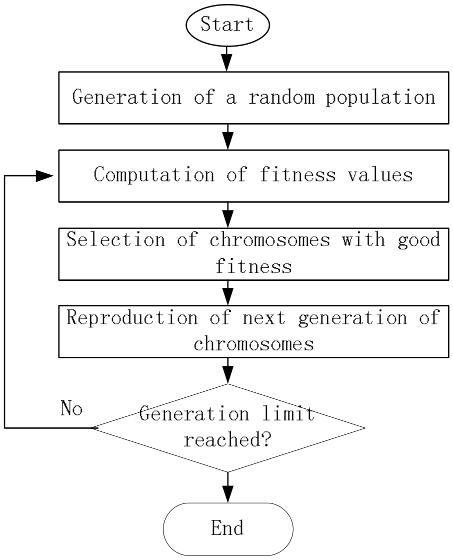

Start
Generation of a random population
Computation of fitness values
Selection of chromosomes with good fitness
Reproduction of next generation of chromosomes
No
Generation limit reached?
End

Fig.2. A typical GA program flow

Fig. 2 shows the flow of a typical GA program. It begins with a population of chromosomes either generated randomly or gleaned from some known domain knowledge. Subsequently, it proceeds to evaluate the fitness of all the chromosomes, select good chromosomes for reproduction, and produce the next generation of chromosomes. More specifically, each chromosome is evaluated according to a given performance criterion or fitness function, and is assigned a fitness score. Using the fitness value attained by each chromosome, good chromosomes are selected to undergo reproduction. Reproduction involves the creation of offspring using two operators, namely crossover and mutation (Fig. 3). By randomly selecting a common crossover site on two parent chromosomes, two new chromosomes are produced. During the process of reproduction, mutation may take place. For example, the binary value of Bit 2 in Fig. 3 has been changed from 0 to 1. The above process of fitness evaluation, chromosome selection, and reproduction of the next generation of chromosomes continues for a predetermined number of generations or until an acceptable performance level is reached.

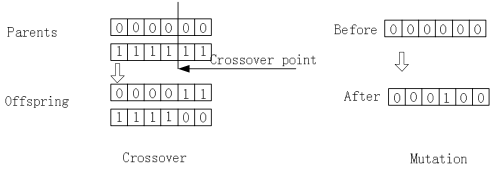


Fig. 3. Genetic crossover and mutation operation

5. System Overview

To improve classification accuracy and speed we proposed a hybrid solution called RGSC(Rough sets and Genetic algorithms for SVM classifier).The system architectures shown in Fig.4. The detailed explanation is as follows:

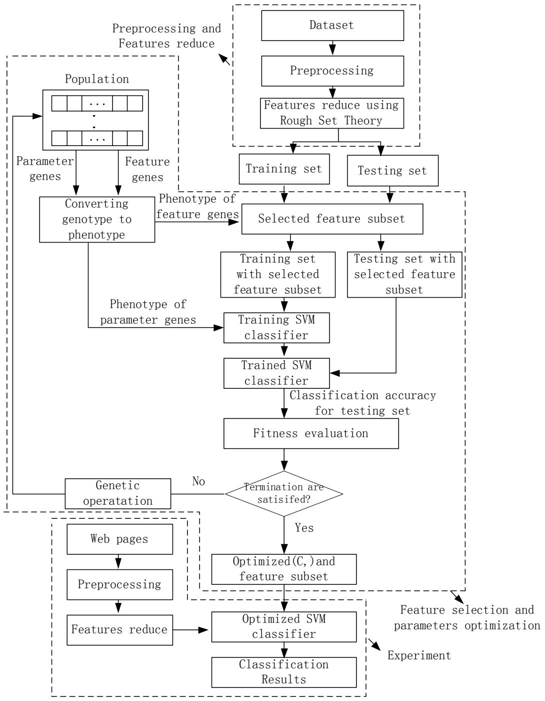

Preprocessing and Features reduce
Dataset
Preprocessing
Features reduce using Rough Set Theory
Population
Parameter genes
Feature genes
Training set
Testing set
Converting genotype to phenotype
Phenotype of feature genes
Selected feature subset
Training set with selected feature subset
Testing set with selected feature subset
Phenotype of parameter genes
Training SVM classifier
Trained SVM classifier
Classification accuracy for testing set
Fitness evaluation
Genetic operatation
No
Termination are satisifed?
Yes
Optimized(C,)and feature subset
Feature selection and parameters optimization
Web pages
Preprocessing
Features reduce
Optimized SVM classifier
Classification Results
Experiment

Fig.4. System architectures

(1) Preprocessing: preprocessing includes remove HTML tags, segment word and construct Vector Space Model.

(2) Feature reduction by rough sets. Our objective is to find a reduction with minimal number of attributes, describes in Alg. 1.

(3) Converting genotype to phenotype. This step will convert each parameter and feature chromosome from its genotype into a phenotype.

(4) Feature subset. After the genetic operation and converting each feature subset chromosome from the genotype into the phenotype, a feature subset can be determined.

(5) Fitness evaluation. For each chromosome representing C, and selected features, training dataset is used to train the SVM classifier, while the testing dataset is used to calculate classification accuracy. When the classification accuracy is obtained, each chromosome is evaluated by fitness function—formula (8).

(6) Termination criteria. When the termination criteria are satisfied, the process ends; otherwise, we proceed with the next generation.

(7) Genetic operation. In this step, the system searches for better solutions by genetic operations, including selection, crossover, mutation, and replacement.

(8) Input the preprocessed data sets into the obtained optimized SVM classifier.

5.1 Algorithm of RST-based feature reduce

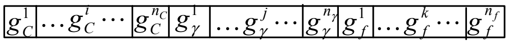
Based on section 3 described. We proposed rough set feature reduction algorithm of finding a reduction of a decision table, which is outlined below.

```
Algorithm：Rough Sets Attribute Reduction algorithm
Input: decision table $T=\langle U,C\cup D,V,f\rangle$, $U=\{x_1,x_2,\cdots x_m\}$, $C=\{C_1,C_2,\cdots C_n\}$
Output: a reduction of $T$, denoted as Redu.
1. construct the binary discernibility matrix $M$ of $T$;
2. delete the rows in the M which are all 0's, Redu=$\phi$
/* delete pairs of inconsistent objects*/
3. while $(M\neq\phi)$
4.     {(1) select an attribute ci in the M with the highest discernibility degree (if there are several $c_j$ (j=1,2,…,m) with the same highest discernibility degree, choose randomly an attribute from them);
5.     (2) Redu $\leftarrow$ Redu $\cup\{c_i\}$;
6.     (3) remove the rows which have ''1'' in the $c_i$ column from M;
7.     (4) remove the $c_i$ column from M; }endwhile
/* the following steps remove redundant attributes from Redu */
8. suppose that Redu =$\{r_1,r_2,\cdots r_k\}$ contains k attributes which are sorted by the order of entering Redu, $r_k$ is the first attributes chosen into Redu, $r_1$ is the last one chosen into Redu.
9. get the binary discernibility matrix MR of decision table TR=$\langle U,\mathrm{Re}du\cup\{d\},V,f\rangle$;
10. delete the rows in the MR which are all 0's;
11. for i = 2 to k{
12.         remove the $r_i$ column from MR;
13.         if(no row in the MR is all 0's){
14.             Redu $\leftarrow$ Redu – { $r_i$ };
15.         else
16.         Put the $r_i$ column back to MR;
17.       Endif;}
18. Endfor;}
```

Alg. 1. Rough Sets Attribute Reduction algorithm

5.2 Chromosome design

To implement our proposed approach, this research used the RBF kernel function for the SVM classifier because the RBF kernel function can analysis higher-dimensional data and requires that only two parameters, C and be defined (Hsu, Chang, & Lin, 2003; Lin and Lin, 2003). When the RBF kernel is selected, the parameters (C and) and features used as input attributes must be optimized using our proposed GA-based system. Therefore, the chromosome comprises three parts, C, , and the features mask. However, these chromosomes have different parameters when other types of kernel functions are selected. The binary coding system was used to represent the chromosome.



g_C^1	$\cdots g_C^i\cdots$	$g_C^{n_C}$	g_γ^1	$\cdots g_\gamma^j\cdots$	$g_\gamma^{n_\gamma}$	g_f^1	$\cdots g_f^k\cdots$	$g_f^{n_f}$

Fig.5. The chromosome comprises three parts, C, γ, and the features mask

Fig. 5 shows the binary chromosome representation of our design. In Fig. 5, $g_C^1\sim g_C^{n_C}$ represents the value of parameter C, $g_{\gamma}^{1} \sim g_{\gamma}^{n_{\gamma}}$ represents the parameter value , and $g_{f}^{1} \sim g_{f}^{n_{f}}$ represents the feature mask. nc is the number of bits representing parameter C, nr is the number of bits representing parameter , and nf is the number of bits representing the features. Note that we can choose nc and n according to the calculation precision required, and that n equals the number of features varying from the different datasets.

In Fig. 5, the bit strings representing the genotype of parameter C and should be transformed into phenotype by Eq. (7). Note that the precision of representing parameter depends on the length of the bit string (nc and n); and the minimum and maximum value of the parameter is determined by the user. For chromosome representing the feature mask, the bit with value '1' represents the feature is selected, and '0' indicates feature is not selected.

$$p = \min_{p} + \frac{\max_{p} + \min_{p}}{2^{l} - 1} \times d \qquad (7)$$

P phenotype of bit string

minp minimum value of the parameter

maxp maximum value of the parameter

d decimal value of bit string

l length of bit string

5.3 Fitness function

Classification accuracy, the number of selected features, and the feature cost are the three criteria used to design a fitness function. Thus, for the individual (chromosome) with high classification accuracy, a small number of features, and low total feature cost produce a high fitness value. We solve the multiple criteria problem by creating a single objective fitness function that combines the three goals into one. As defined by formula (23), the fitness has two predefined weights: (i) WA for the classification accuracy; (ii) WF for the summation of the selected feature (with nonzero Fi) multiplying its cost. The weight accuracy can be adjusted to 100% if accuracy is the most important. Generally, WA can be set from 75 to 100% according to user's requirements. If we do not have the feature cost information, the cost Ci can be set to the same value, e.g. '1' or another number. The chromosome with high fitness value has high probability to be preserved to the next generation, so user should appropriately define these settings according to his requirements.

$$fitness = W_{A} \times SVM\_accuracy + W_{F} \times \left(\sum_{i=1}^{n_f} C_{i} \times F_{i}\right)^{-1} \qquad (8)$$

WA SVM classification accuracy weight

SVM_accuracy SVM classification accuracy

WF weight for the number of features

Ci cost of feature *i*

Fi '1' represents that feature *i* is selected; '0' represents that feature *i* is not selected

6. Experiments

In this section, we designed an experiment to test the performance of the proposed RGSC. We also investigated k-NN and Decision tree to compare their classification performances. The experiments are described below.

6.1 Experiment environment

Our implementation was carried out on the YALE(Yet Another Learning Environment) 3.3 development environment(Available at:http://rapid-i.com/).Feature reduction by Rough Sets Theory carried out on ROSETTA(you can download it from http://rosetta.sourceforge.net/) .The empirical evaluation was performed on Intel Pentium IV CPU running at 3.0 GHz and 1GB RAM.

6.2 Data set

To provide an overview on the base line accuracy of the classifiers and to compare them with various studies, the Reuters 21578 corpus was taken in our experiments (this collection is publicly available at: http://www.research.att.com/~lewis/reuters21578.html) . These stories average about 200 words in length. Various splits of the Reuters 21578 can be used, whereas we followed the ModApte split in which 75% of the stories (9603 stories) are used to build classifiers and the remaining 25% (3299 stories) to test the accuracy of the resulting models in reproducing the manual category assignments. From this split, all categories (including the documents not assigned to any category), which have no training or test document were deleted. The resulting data set has 90 different categories and is the same as that used by Joachims (1998).

6.3 The performance measure

Given a binary-classification problem of topic versus not-topic, recall is the ratio of the correct topic cases to the total topic cases. Precision is the ratio of correct topic cases to the total predicted topic cases. The standard evaluation criterion for the Reuters benchmark is the breakeven point, at which precision equals recall, and the F1 measure, which is defined as (2×precision× recall)/(precision + recall).

6.4 Simulate

Figure.6, Figure.7 and Figure.8 show the performance of our proposed method against to the decision tree (Weiss et al., 1999) and the k-NN classifiers (Aas & Eikvil, 1999) for the ten most frequent categories.

The precision of the k-NN, Decision tree, and RGSC is shown in Figure.6, the recall of the k-NN, Decision tree, and RGSC is shown in Figure.7, the F1-value of the k-NN, Decision tree, and RGSC is shown in Figure.8 , and the speed of the k-NN, Decision tree, and RGSC is shown in Figure.9

The average precision for k-NN, Decision tree and RGSC are 75.6, 84.9 and 90.7% respectively. With the exception of category Grain, Crude, Wheat, Corn, the precision of each category for RGSC is higher than other two methods. This indicates that the RGSC methods perform generally high precision.

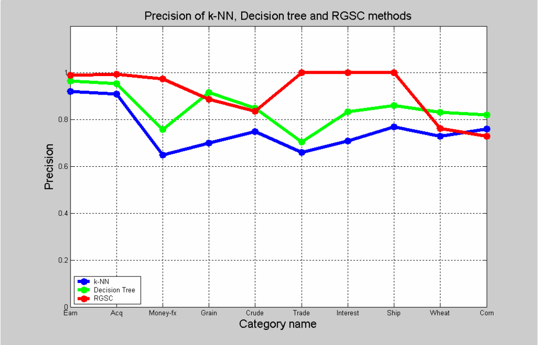


Fig.6. The precision of the k-NN, Decision tree, and RGSC

The average recall results for the RGSC, k-NN and the Decision tree are 95.1, 82.3, and 87.8%, respectively. The recall of k-NN and Decision tree are nearly the same and both are lower than the RGSC. The RGSC can classify documents into the correct category mapping to precision, with a high recall ratio (Fig. 6).

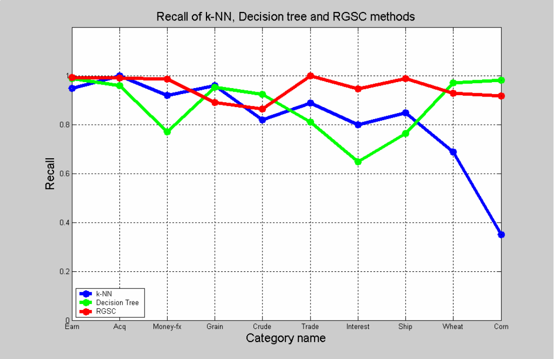


Fig.7. The recall of the k-NN, Decision tree, and RGSC

The simulated result shows that the average F1-values of the RGSC, k-NN and the Decision tree are 92.5,78.8 and 87.9%, respectively. This indicates that the RGSC yields a better classification result than the other two methods. But RGSC classification performs lower than Decision tree for the “Grain”, “Crude”, “Sheep” and “Wheat” categories. We now try to explain why the performance of those four categories is poorer than other categories. We find that the “Grain”, “Crude”, “Sheep” and “Wheat” categories contain a smaller number of documents in Reuters 21578. This indicates that the RGSC is able to effectively process categories with large documents. But poorer with smaller documents. Fig. 7 shows that among the three methods, the RGSC has the average highest classification result.

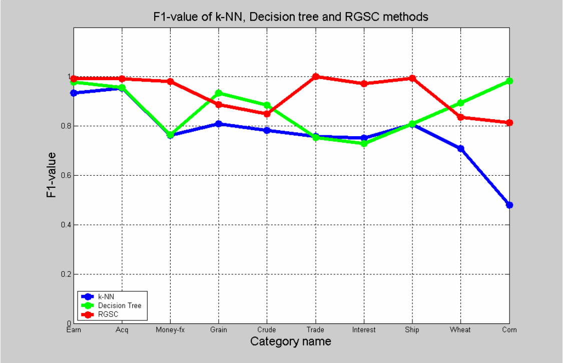


Fig.8. The F1-value of the k-NN, Decision tree, and RGSC

The speed for RGSC, k-NN and Decision tree are 13.3, 20.5 and 35.7 seconds respectively. This indicates that the RGSC is more effective than the other two methods. It easily explain that the feature is reduced by Rough Set Theory before input the SVM classifier.

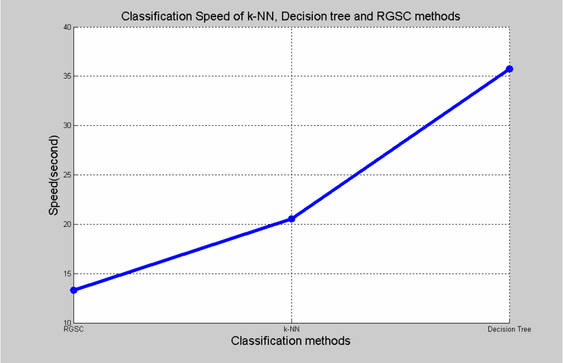


Fig.9. The speed of the k-NN, Decision tree, and RGSC

In general the performance of our approach is best in average.

7. Conclusion

In this paper, we have proposed a document classification method using an SVM based on Rough Sets Theory and Genetic Algorithms. The feature vectors are reduced by Rough Set Theory. The feature vectors are selected and parameters optimization by Genetic Algorithms. The experimental results show that the RGSC we proposed yields the best result of these three methods. The experiment also demonstrated that the RGSC yields better accuracy even with a large data set. When the larger category has more training data, the RGSC is able categorize documents more accuracy. In future research, we will emphasis on the kernel function selection and parameters optimization for the Genetic Algorithms to improve the performance of RGSC.

References

1. Burges, C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, (1998).2(2), 121–167.

2. Chang, C. C., & Lin, C.J. LIBSVM: A library for support vector machines. Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm. (2001)

3. Cristianini, N., & Shawe-Taylor, J. An introduction to support vector machines. Cambridge: Cambridge University Press. (2000).100-103

4. S.K. Pal, A. Skowron (Eds.), Rough Fuzzy Hybridization: A New Trend in Decision-Making, Springer, Singapore, (1983) 36-70

5. Z. Pawlak, Rough sets, Int. J. Comput. Sci. 11 (1982) 341-356.

6. Z. Pawlak, in: Rough Sets, Theoretical Aspects of Reasoning About Data, Kluwer, Dordrecht,(1991)10-50.

7. Z. Pawlak, A. Skowron, Rough membership functions, in: R.R. Yaeger, M. Fedrizzi, J. Kacprzyk (Eds.), Advances in the Dempster Shafer Theory of Evidence, Wiley Inc, New York, Chichester, Brisbane, Toronto, Singapore, (1994) 251-271.

8. Z. Pawlak, S.K.M. Wong, W. Ziarko, Rough sets: probabilistic versus deterministic approach, Int. J. Man-Mach. Stud. 29 (1988) 81-85.

9. Davis, L. Handbook of genetic algorithms. New York, NY: Nostrand Reinhold. (1991)55-61

10. Goldberg, D. E. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley. (1989).23-32

11. Grefenstette, J. J. Genetic algorithms for machine learning. Boston, MA: Kluwer. (1994).100-106

12. Vapnik, V. N. The nature of statistical learning theory. New York: Springer-Verlag. (1995).61-70

13. Frohlich, H., & Chapelle, O. Feature selection for support vector machines by means of genetic algorithms. Proceedings of the 15th IEEE international conference on tools with artificial intelligence, Sacramento, CA, USA (2003). 142–148.

14. Yu, G. X., Ostrouchov, G., Geist, A., & Samatova, N. F. An SVMbased algorithm for identification of photosynthesis-specific genome features. Second IEEE computer society bioinformatics conference, CA, USA (2003).235–243.

15. Bradley, P. S., Mangasarian, O. L., & Street, W. N. Feature selection via mathematical programming. INFORMS Journal on Computing, 10, (1998).209–217.

16. Hsu, C. W., Chang, C. C., & Lin, C. J. A practical guide to support vector classification. Available at: http://www.csie.ntu.edu.tw/~cjlin/ papers/guide/guide.pdf. (2003).

17. LaValle, S. M.,& Branicky, M. S. On the relationship between classical grid search and probabilistic roadmaps. International Journal of Robotics Research, 23(7–8), (2002). 673–692.

18. Joachims, T. Text categorization with support vector machines. In Proceedings of European conference on machine learning (ECML) Chemintz, DE. (1998) 137–142.

19. Pontil, M., & Verri, A. Support vector machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6), (1998). 637–646.