Journal of Universal Computer Science, vol. 15, no. 4 (2009), 840-858
submitted: 30/6/08, accepted: 27/2/09, appeared: 28/2/09 © J.UCS

Multilayer Ensemble Pruning via Novel Multi-sub-swarm Particle Swarm Optimization

Jun Zhang
(Intelligent Computing Lab, Institute of Intelligent Machines, Chinese Academy of Sciences
P.O.Box 1130, Hefei, Anhui 230031, China
School of Electronic Science and Technology, Anhui University, Hefei Anhui, China
wwwzhangjun@gmail.com)

Kwok-Wing Chau
(Department of Civil & Structural Engineering, The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
cekwchau@inet.polyu.edu.hk)

Abstract: Recently, classifier ensemble methods are gaining more and more attention in the machine-learning and data-mining communities. In most cases, the performance of an ensemble is better than a single classifier. Many methods for creating diverse classifiers were developed during the past decade. When these diverse classifiers are generated, it is important to select the proper base classifier to join the ensemble. Usually, this selection process is called pruning the ensemble. In general, the ensemble pruning is a selection process in which an optimal combination will be selected from many existing base classifiers. Some base classifiers containing useful information may be excluded in this pruning process. To avoid this problem, the multilayer ensemble pruning model is used in this paper. In this model, the pruning of one layer can be seen as a multimodal optimization problem. A novel multi-sub-swarm particle swarm optimization (MSSPSO) is used here to find multi-solutions for this multilayer ensemble pruning model. In this model, each base classifier will generate an oracle output. Each layer will use MSSPSO algorithm to generate a different pruning based on previous oracle output. A series of experiments using UCI dataset is conducted, the experimental results show that the multilayer ensemble pruning via MSSPSO algorithm can improve the generalization performance of the multi-classifiers ensemble system. Besides, the experimental results show a relationship between the diversity and the pruning technique.

Keywords: Particle swarm optimization; ensemble pruning; classifier ensemble; multi-layer ensemble model
Categories: L.1.3

1 Introduction

The traditional pattern recognition methods usually used some methods to find a best complicated classifier to solve the problem [Huang, 1996)] [Huang, 1997] [Huang, 1999] [Wang et al., 2005] [Niklas and Paul, 2007]. However, it is hard to use for a user who lacks expertise on these specific classifiers. In recent years, ensemble methods have gained more and more attention in the machine-learning and data-mining communities. In most cases, the performance of an ensemble is better than a single classifier. It is generally agreed that the performance of an ensemble system relies on the creation of a collection of diverse yet accurate base classifiers [Kuncheva

and Whitaker, 2003] [Shipp and Kuncheva, 2002] [Kittler et al., 1998]. A number of methods has been developed for creating diverse classifiers in ensemble systems. Among them are Random Subspaces [Ho, 1998], Bagging [Breiman, 1996] and Boosting [Freund and Schapire, 1996] [Breiman, 1998] [Kuncheva et al., 2002] [Schapire et al., 1998]. The Random Subspaces method creates various classifiers by using different subsets of features as training set. Bagging generates diverse classifiers by randomly selecting subsets of samples as training sets to train the base classifiers. Boosting also uses parts of samples to train classifiers, however, difficult samples have a greater probability of being selected and easier samples have less chance of being used for training. When the diverse classifier have been generated by different methods, it becomes more important how to select the base classifier to join the ensemble. It is believed that the optimal combinations of classifiers should have good individual performances and at the same time preserve sufficient level of diversity [Sharkey et al., 1997]. Usually, this selection process is called pruning the ensemble [Margineantu and Dietterich, 1997]. There are two reasons for pruning the ensemble: first, the performance of an ensemble consisting of some classifiers could be better than *all* [Zhou et al., 2002]; second, the ensemble methods require a large number of memories to store all the base classifiers, the ensemble pruning can hugely decrease the memory used in real world application.

Currently, no efficient criterion exists for selecting classifiers to join the ensemble. Neither individual performance nor diversity on their own can be used to select the appropriate base classifier [Rogova, 1994] [Ruta and Gabrys, 2001] [Zeuobi and Cunningham, 2001]. The majority voting error (MVE) of the validation set is usually used as a criterion. In general, the genetic algorithm (GA) or other algorithms is used for selecting the base classifier. When an ensemble has a smaller MVE, the ensemble is considered to be having better performance [Mukherjee and Fine, 1996] [Zhou and Tang, 2003], However, these methods usually only get one optimal selection from many existing base classifiers, some classifiers with useful information can be excluded from the ensemble. Dymitr Ruta [Ruta, 2003] [Ruta and Gabrys, 2005] proposes a novel multilayer selection-fusion model, which is implemented by evolutionary algorithms and majority voting. In fact, this method converts the inflexible majority voting combiner into a very flexible model. Ruta claims that the model can improve the system's generalization performance effectively. However, this model still had some minor drawbacks: first, it only employed ordinary evolutionary algorithm, so that some measures must be taken to avoid finding the same solution. Actually, many niching techniques [Zhang et al., 2006b] of the evolutionary computation can be used to solve this problem. Second, the relationship of the diversity and the multilayer selection is not fully explored. Although Ruta pointed out that the performance of the second layer improved a lot, the reason for this is not detailed. This paper will reveal the relationship of the diversity and the multilayer selection for further research. Besides, in some sense this model equals weight majority voting. This situation may be lead to the overfitting of the model [Duin, 2002]. The overfitting problem is not discussed in Ruta's model.

Recently, Particle swarm optimization (PSO) has drawn more researchers' attention. PSO has a few parameters to adjust and is easy to implement, which has found applications in many areas. In machine learning area, PSO is used for feature selection [Agrafiotis and Cedeño, 2002] and training the neural network [Zhang et al.,

2006a] [Eberhart and Shi, 1998] [Eberhart and Hu, 1999]. This paper proposes a multilayer ensemble pruning based on a novel multi-sub-swarm particle swarm optimization (MSSPSO) [Zhang et al., 2007]. The multi-sub-swarm PSO algorithm can find multi-solutions effectively. In every layer, MSSPSO is used to detect multi-ensemble pruning based on previous oracle output. In this way, the multilayer ensemble pruning model is formed. Besides, some relations of the diversity and the multilayer selection model will be revealed. In fact, if the diversity of one layer's output is small, the performance of the next layer will be difficulty to get improvement. And more importantly, this paper proposes a novel fitness for PSO to select the output of different layers to take part in ensemble. The generalization ability of the proposed model will be improved effectively by this method.

This paper is organized as follows: In section 2, a novel multi-sub-swarm particle swarm optimization algorithm is briefly introduced. The multilayer ensemble pruning model based on MSSPSO is discussed in section 3. Section 4 gives a series of experimental results on UCI datasets. Conclusions are given in section 5.

2 Multi-sub-swarm Particle Swarm Optimization Algorithm

2.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic search algorithm. The algorithm was first developed by Dr. Eberhart and Dr. Kennedy in 1995 [Kennedy and Eberhart, 1995], inspired by social behavior of bird flocking or fish schooling. In this algorithm, many individuals, referred to as particles, are grouped into a swarm, which "flies" through multidimensional search space. Each particle in the swarm represents a candidate solution to the optimization problem. These particles fly with a certain velocity and find the global best position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of its best position (P_i) as well as the best position of its neighbors (P_g), then a new position can be computed. The original PSO were modified by Shi and Eberhart [Shi and Eberhart, 1998] with the introduction of inertia weight. The equations for the manipulation of the swarm can be written as:

$$V_i(t+1) = W * V_i(t) + C1 * rand1() * (P_i - X_i(t)) + C2 * rand2() * (P_g - X_i(t)) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t) \quad (2)$$

where $i = 1,2,\ldots N$, N is the number of the particles. W is called as inertia weight. $C1$ and $C2$ are positive constants, referred to as cognitive and social parameters, rand1 (*) and rand2 (*) are random numbers, respectively, uniformly distributed in [0..1]. The *ith* particle of the swarm can be represented by the D dimensional vector Xi. The velocity of the i_{th} particle is as V_i. The velocity is updated by the individual's own and social best experience.

2.2 Multi-sub-swarm Particle Swarm Optimization Algorithm

A multi-sub-swarm Particle Swarm Optimization Algorithm (MSSPSO) was proposed in [Zhang et al., 2007]. The proposed algorithm is an adaptive niche technique. This paper only briefly introduces the mechanism of the MSSPSO. The proposed algorithm can imitate the process of an animal colony occupying its territory in nature. Every animal colony has its own marking territory. When an invader wants to occupy the marking territory, it must compete with the owner. As a result, the winner will hold the territory while the loser will be obliged to find a new district. In MSSPSO algorithm, a multi-sub-swarm is employed to detect multi-solutions simultaneously, where each sub-swarm detects one solution. Considering that the most influential particle of a swarm is the globally optimal one in the PSO algorithm, this work only uses the globally optimal particle of each sub-swarm located in the same niche to compete with each other. The winning sub-swarm acquires a marking niche, while the loser will be re-initialized in order to explore a new area.

As a result of this competition process, a new difficulty may arise: the losing sub-swarm will possibly converge to the same niche it found before. Also, because the multi-sub-swarm was launched simultaneously, another sub-swarm will likely find the same niche too. In this work, we offer an effective solution to this problem. The PSO algorithm actually has two influencing factors for a particle to move: the global best position of the swarm and its private best position remembered at earlier time. If these two factors of a particle can be shifted, then a particle's tracking can be altered. In this paper, the algorithm does not directly change these two factors. On the contrary, the algorithm makes these particles lose their influence in their own sub-swarm. This is achieved by decreasing the fitness of a particle that invades another marking niche. By these means, the algorithm encourages the different sub-swarms to converge to different places in the search space. The modified fitness function of a particle that invades another niche must satisfy the following equation:

$$eval(x_n^i) = \begin{cases} f(X_i^n) & if \quad hill\_valley(X_i^n, P_g^k) = 1 \\ f(X_i^n) - p(X_i^n) & otherwise \end{cases} \tag{3}$$

In Equation (3), X^n_i represents the *ith* particle in the *nth* sub-swarm and P_g^k is the best particle in *kth* sub-swarm, *k* is not equal *n*. $p(X^n_i)$ represents penalty function, this paper only uses a constant penalty function. The determination of the niche radius is generally a hard work existing in most niche techniques. However, if we have a method that can determine whether or not two points of search space belong to a peak of the multimodal function, then the niche radius is not needed in this situation. So, Ursem's hill valley function [Ursem, 1999] is used here to judge whether two particles belong to one niche. The hill valley function does not need set niche radius for any multimodal function. That function is described as follows:

```
Hill_valley(ip,iq,samples)
      minfit=min(fitness(ip),fitness(iq))
      for j=1 to samples.length
          Calculate point iinterior on the line between the points ip and iq
        If (minfit> fitness(iinterior))
          return 1
          end if
      end for
return 0
```

Figure 1: Pseudo code of the hill valley function

where i_p and i_q are any two points in search space. [Fig. 2] just shows one dimensional function case. In fact, it can be easily extended to the case including arbitrary dimensions. Generally speaking, the function returns 0 if the fitness of all the interior points is greater than the minimal fitness of i_p and i_q, otherwise it returns 1. With this function, the algorithm is able to determine whether i_p and i_q belong to the same hill or not, in other words, it can be used to determine whether a point belongs to a niche.

A samples array is generally used to calculate the interior points where the hill valley function computes the fitness of these samples. The points $i_{interior}$ can be calculated as:

$$i_{\text{int}\,erior} = i_p + (i_q - i_p) \bullet samples[j] \tag{4}$$

where j is *jth* entry in the array. The upper boundary of j is the length of the samples, which is very important for the hill valley function. We refer to the length as sample rate (SR).

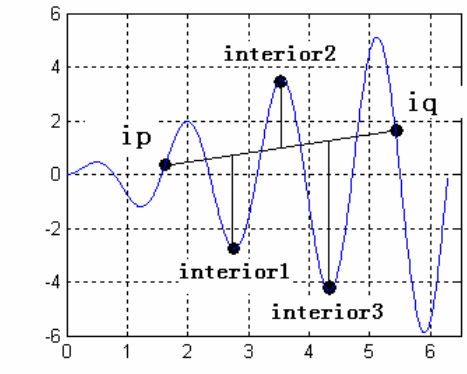

Figure 2: The diagram for hill valley function

The multi-sub-swarm niche PSO algorithm can be described as follows. The multi-sub-swarm was launched simultaneously. The niche in which the best particle of each sub-swarm is located is marked as that sub-swarm's territory. The marking territory can be shifted, with the best particle moving to another niche. When two different

sub-swarms occupy the same niche, the best particles of each sub-swarm compete with each other. The loser will be re-initialized. The particles of the other sub-swarm that invade a marking niche will be punished. The fitness of such particles will become smaller through equation 3. Also, the remembered particle position of each sub-swarm must be updated. If the remembered position is located in the other marked niche (since the marked niche can shift with the best particle moving), its fitness must be decreased. The basic algorithm can be described as follows:

```
Algorithm Multi-sub-swarm niche PSO algorithm
  Create and initialize N sub-swarm of PSO algorithm
  Repeat
    For each sub-swarm,
      If the best particle of different sub-swarms are located in the
same niche
          Compare their fitness: the smaller is marked as loser, the
larger as winner
      Else
          Mark the sub-swarm as winner
      End if
    Next
    Reinitialize the loser sub-swarm; mark the winner's niche
    For every particle and remembered particle position of each
sub-swarm
      If the particle invades another marked niche
        Use equation 3 to decrease the fitness of the particle
      End if
    Next
  Train each sub-swarm as original PSO algorithm
Until all sub-swarms converge or stopping condition is met
```

Figure 3: the pseudo-code of Multi-sub-swarm niche PSO algorithm

2.3 Binary Particle Swarm Optimization

However, the particle swarm optimization is a real-valued algorithm in its original version. When the PSO is used in the proposed multilayer ensemble pruning model, binary version PSO should be adopted. In this work, a simple binary version PSO proposed by [Kennedy and Eberhart, 1997] is adopted. The velocity of a particle is used as a probability to determine whether a bit will be in one state or zero. The whole mathematical description is given as follows:

$$S(v) = \frac{1}{1+\exp^{-v}} \tag{5}$$

$$x_{id} = \begin{cases} 1 & if \quad rand(*) \le S(v) \\ 0 & otherwise \end{cases} \tag{6}$$

where v is the velocity of corresponding particle. rand(*) is a random number uniformly distribution between [0,1].

3 Multilayer Ensemble Pruning Model

Usually, the ensemble pruning gets only one optimal selection as an ensemble result. Some classifiers with useful information may be excluded in this pruning process. On the contrary, the multilayer ensemble pruning model can take full advantage of the useful information owned by every base classifier. The whole model is composed of many layers and each layer consists of many ensembles. In this situation, each classifier will have an opportunity to participate in one ensemble. In this model, the pruning of one layer can be seen as a multimodal optimization problem. In fact, the recognition rate of each ensemble composed of different classifiers will be different. More than one ensemble may achieve the better performance. [Fig. 4] is drawn from a real dataset experiment and shows the schematic diagram. See [Fig. 4]: X-axis represents the ensemble size, the rightmost point on the axis indicates all classifiers joining the ensemble and the leftmost point denotes no classifiers joining. Y-axis represents the recognition rate of each ensemble. This figure shows that the pruning of ensemble has formed a complicated multimodal optimization problem. In the first layer of the proposed model, each node is the oracle output of the base classifier on training or validation dataset. In the other layer, one node is a selection of the oracle output from previous layer. This node can also be seen as an ensemble. Each ensemble is a different selection and each oracle output will have an opportunity to be selected for the next layer ensemble. So, every layer will form multi-ensembles using majority voting rule and each ensemble is a new oracle output. A schematic diagram of four layers ensemble pruning model is shown in [Fig. 5].

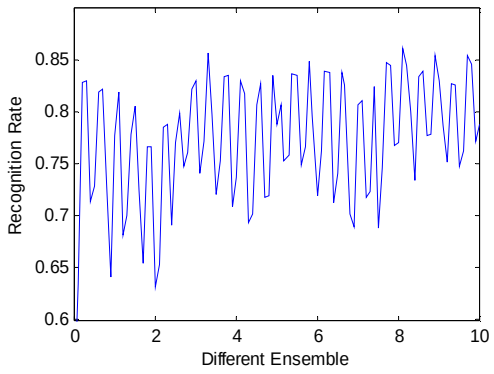

Figure 4: Ensemble-Recognition Rate schematic diagram

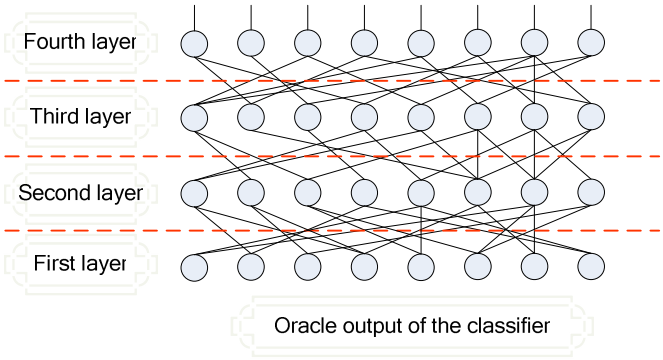

Figure 5: Multilayer ensemble pruning schematic diagram

Here, the oracle output is the foundation of the multilayer ensemble pruning model. The oracle output can be defined as below: Given L classifiers, $D=\{D_1,\ldots D_L\}$, let $y_i=[y_{i1},\ldots,y_{iL}]^T$ denote the output of the L classifiers for input sample x_i from the training or validation dataset, where y_{ij} denotes the output of the *jth* classifier for the *ith* input sample. The meaning of the output takes the form $y_{ij}=0$ for correct and $y_{ij}=1$ for error. In the multilayer ensemble pruning model, each ensemble will get a pruning vector that is used to select a corresponding oracle output. Let $H_i=[h_{i1},\ldots\ldots,h_{iL}]^T$ is the *ith* layer pruning vector, where each h_{ij} indicates that the corresponding *jth* classifier (or oracle output) whether included or excluded (where $h_{ij}=1$ represent inclusion and $h_{ij}=0$ represent exclusion) in ensemble. The target of each pruning ensemble is to get an appropriate binary vector for the selection of a corresponding classifier.

The odd selection must usually be considered in an ensemble problem. The odd selection of an ensemble can avoid the tie when using majority voting (MV) strategy. So we usually want to select the odd number classifiers rather than evens. Many researchers using genetic algorithms must take some special measures such as pairwise mutation, crossover to avoid the even number classifiers selected. Nevertheless, in this multilayer ensemble pruning model, even number classifiers in certain layers may be useful in later selection. When preventing the even number selection in an evolutionary process, it will be very possible to prevent the true optimum in the final generation. In fact, there is no need for these rigid changes in an evolutionary algorithm. If we low the fitness of the even selection, the odd population will slowly get the prominent position in an evolutionary process. The best individual in the last generation will be the odd selection. In this paper, the altered fitness rule is very simple: when the tie happened for classifying a sample, this sample is taken as error recognition.

3.1 Selection Criterion

The selection criterion is very important for an ensemble pruning. However, no theory can guide us to select an appropriate classifier to join the ensemble. In practical application, the most common selection criterion is majority voting error (MVE). The MVE is a measure for the majority voting error rate. In this paper, all output is based

on the oracle output of previous layer. Assuming that the L classifiers voting for an input sample x_i, the majority voting output can be define as follows:

$$y_i^{MV} = \begin{cases} 1, & if \sum_{j=1}^{L} y_{ij} \geq \frac{L}{2} \\ 0, & if \sum_{j=1}^{L} y_{ij} < \frac{L}{2} \end{cases} \tag{7}$$

The *MVE* can be formulated as:

$$MVE = \frac{1}{M} \sum_{i=1}^{M} y_i^{MV} \tag{8}$$

In Equation (8), M is the total number of the input samples. Usually, the MVE is calculated by the validation set. However, if adequate and diverse oracle output is provided, it is very possible to form an overfiting ensemble. This paper proposes a novel selection criterion, average majority voting error. The $\overline{MVE}$ is calculated not only by the validation set, but also training set, then the $\overline{MVE}$ is defined as:

$$\overline{MVE} = \frac{1}{2}(MVE^T + MVE^V) \tag{9}$$

Where MVE^T represents the *MVE* of training set and MVE^V represents the MVE of the validation set.

3.2 The Diversity Measures

Diversity measures have been used to find out what happens within the ensemble. Statisticians have developed several measures of agreement or disagreement between classifiers. This work only wants to reveal the relation between the diversity and the multi-layer pruning model. The most widely used measure is the Kappa statistic [Agresti, 1990)] [Cohen, 1960]. Kappa-error diagram [Margineantu and Dietterich, 1997] is used here to visualize this relation. It is defined as follows.

Given two classifiers D_a and D_b and a dataset containing M examples, the cell C_{ij} contains the number of examples x for which $D_a(x)=i$ and $D_b(x)=j$. If D_a and D_b are identical on the dataset, then all non-zero counts will appear along the diagonal. If Da and Db are very different, then there should be a large number of counts off the diagonal. In this work, the classifier only takes the oracle output. Here we can define

$$\theta_1 = \frac{\sum_{i=0}^{1} C_{ii}}{M} \tag{10}$$

Equation (10) can be used as a measure of agreement. We can define

$$\theta_2 = \sum_{i=0}^{1} \left(\sum_{j=0}^{1} \frac{C_{ij}}{M} \cdot \sum_{j=0}^{1} \frac{C_{ji}}{M} \right) \tag{11}$$

to be the probability that the two classifiers agree by chance. Then the Kappa statistic is defined as follows:

$$K = \frac{\theta_1 - \theta_2}{1 - \theta_2} \tag{12}$$

K=0 when the agreement of the two classifiers equals that expected by chance, and *K=1* when the two classifiers agree on every example.

4 Experimental Results and Discussion

4.1 Experimental Setup

Our main objective is to explore the effect of the multilayer ensemble pruning model and the second objective is to reveal the relation between the diversity and the proposed model. In order to achieve this goal, a series of experiments is arranged in this work. These experiments were conducted using a publicly available dataset provided by UCI machine learning repository [Merz and Murphy, 1998]. See [Tab. 1] show the datasets employed in this work.

Dataset	Features	Classes	Sample Size
Diabetes	8	2	768
Wisconsin Breast Cancer	9	2	699
Iris	4	3	150
Glass	9	6	214
Liver Disorder	6	2	345

Table 1: The UCI datasets employed

To show the influence of diversity on pruning, the base classifiers are generated by a different ensemble algorithm. In all experiments, the Bagging and Adaboost algorithm is used to generate twenty-one base classifiers for ensemble pruning. The artificial neural network (ANN) is employed as a base classifier, standard backpropagation algorithm is used for training, and the transfer function for each hidden and output unit is sigmoid function. The number of hidden units is set 10, this number is arbitrary setting. The target of these experiments is not a search of the optimal architecture of the ANN. The Kappa-error diagram is used here for visualizing the diversity. For each pair of classifiers produced by Bagging or Adaboost algorithm, Kappa is computed on the training or validation set. X-axis of the Kappa-Error diagram represents a Kappa and Y-axis represents the average error rate of corresponding pair of classifiers. All experiments are based on 10-fold cross-validation methods: the whole dataset is divided into ten parts, any eight parts are used as training set and one is used as validation set, the remaining part is used as testing set. Ten experiments are conducted for each dataset.

The parameters of the multi-sub-swarm PSO algorithm are set as follows: the number of sub-swarm is set equal to the number of classifiers. In this work, the number of sub-swarms is set as twenty-one. The sample rate (SN) of hill valley is set two and the sample array is set [0.01, 0.09]. Other parameters are set as the default values of the ordinary PSO. Here, the linear decrease weight PSO (LDW-PSO) [28] is

used. The start w is set 0.9 and the end w is set 0.4, 30 particles is used in each sub-swarm. The iterative number is set as 50.

4.2 Experimental Results and Discussion

The oracle output is created on twenty-one base classifiers generated by Adaboost and Bagging algorithm from the training or validation set respectively. Then the multilayer ensemble pruning model is based on these oracle output. In every layer, MSSPSO algorithm is employed to choose the previous output to join the next ensemble. The number of the layer is set 4.

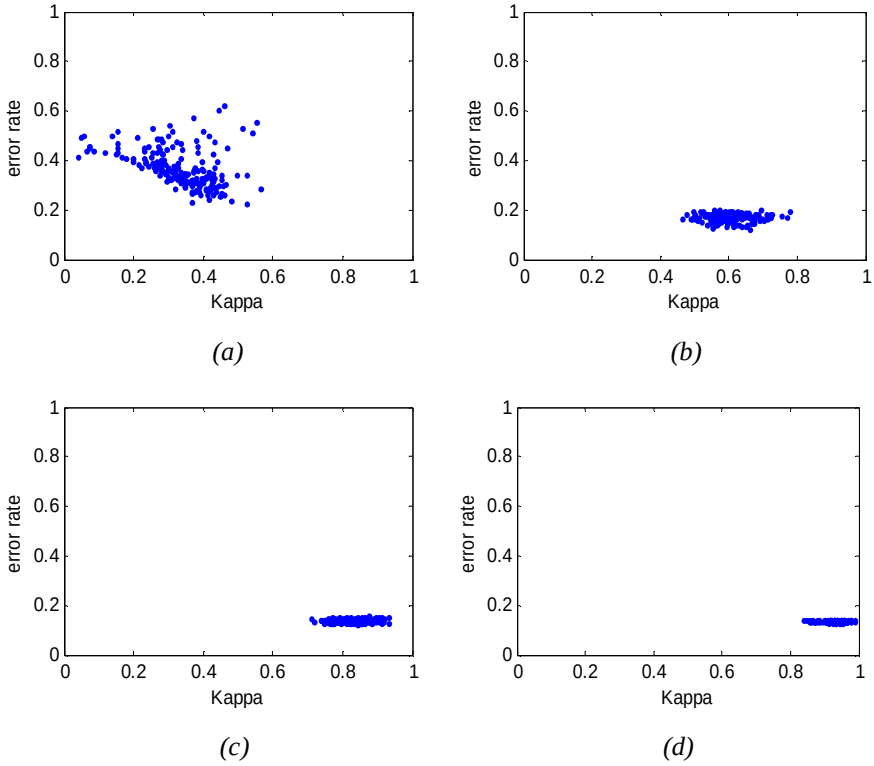

Figure 6: Kappa-Error diagrams on train samples of diabetes dataset in different layer based on Adaboost, (a) is first layer, (b) is second layer, (c) is third layer and (d) is fourth layer

[Fig. 6] show Kappa-error diagram on train set of diabetes dataset based on Adaboost. Kappa equals zero when the agreement of the two outputs equals that expected by chance, and Kappa equals one when two outputs agree on every samples. (a) shows the diversity of the first layer of the proposed model. We can see that the diversity and the error rate of the base classifiers is very large in this layer. (b)-(d) show the diversity of 2-4 layer in proposed multilayer ensemble model respectively. These figures show that the diversity and the error-rate of each layer will decrease

with the number of the layer increasing. The proposed model improved the performance of the ensemble.

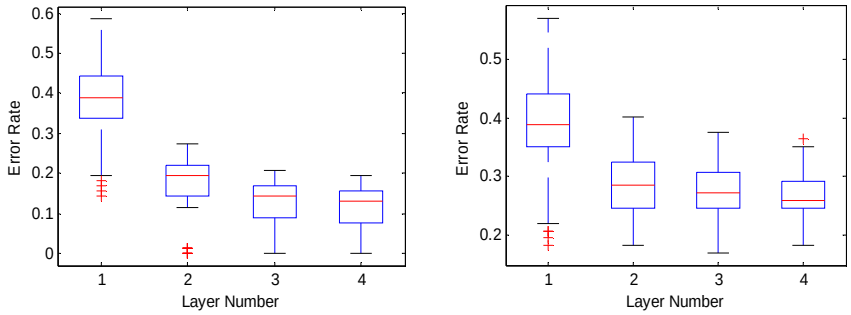

Figure 7: The layer number-Error rate diagrams on diabetes dataset based on Adaboost, 4 layers model and 21 nodes at each layer, left shows the error rate of the validate set, right shows the error rate of the test set

[Fig. 7] shows the corresponding ten experimental results on diabetes dataset based on Adaboost. The left one shows the results of validate set and the right one shows test set. The two figures clearly show that the proposed multilayer model can improve the ensemble performance, especially the first layer. This situation agrees with the Kappa-Error diagram. Since the diversity of an ensemble is decreasing with the pruning being carried out, it is difficult to improve the performance of an ensemble. For a single best output, the error rate of the test set will possibly increase with the error of validation sets decreasing. The overfitting may happen in this situation. However, we can see that the average error rate of test set is still decreased from [Fig. 7], although the improved performance is very small in 3 and 4 layer.

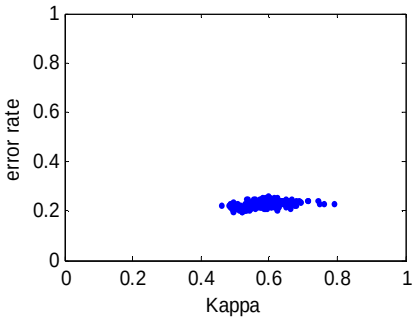

Figure 8: Kappa-Error diagram on train samples of diabetes dataset based on Bagging in first layer

[Fig. 8] shows the Kappa-Error diagram on train set of diabetes dataset based on Bagging algorithm, this is the first layer. [Fig. 8] indicates that the diversity of Bagging is smaller than Adaboost. Also, [Fig. 8] shows that the average error-rate of each pair of base classifiers of the Bagging algorithm is smaller than the Adaboost

algorithm. [Fig. 9] shows corresponding multilayer pruning results of diabetes dataset based on Bagging. These figures indicate that the performance of the multilayer ensemble model can be still improved. But the improved performance of the Bagging is smaller than Adaboost because of the diversity decreasing.

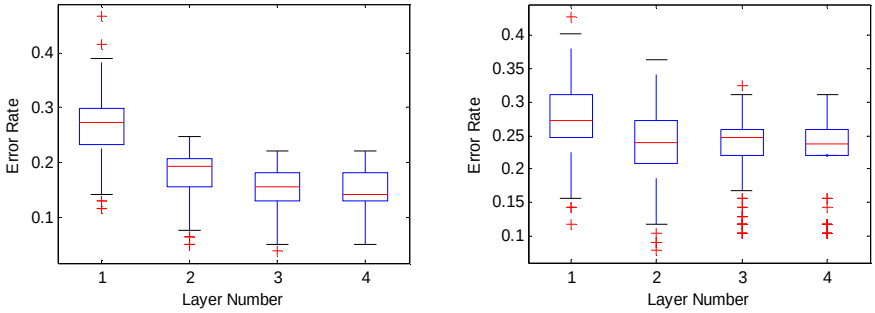

Figure 9: The layer number-Error rate diagrams on diabetes dataset based on Bagging, 4 layer model and 21 nodes at each layer, the left image shows the error rate of the validate set, the right one shows the error rate of the test set.

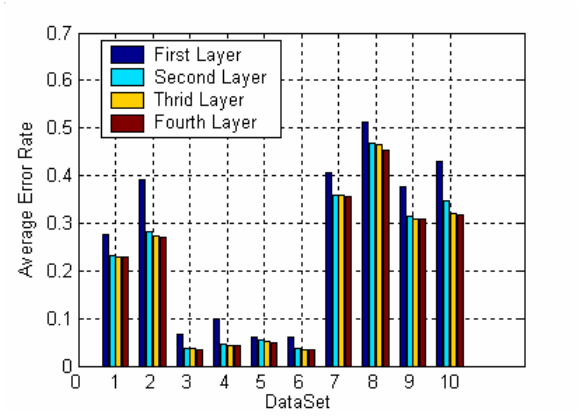

Figure 10: The average error-rate on different dataset

The experimental results of the other dataset are shown in [Fig. 11-Fig. 16], these figures show similar experimental results: in all experiments, the second-layer of the model can improve the performance hugely because the diversity of the first layer is larger. Besides, the performance of the ensemble depends on the recognition rate of the base classifier. Although the improved performance of the ensemble based on Bagging is smaller than the Adaboost, the performance of the ensemble based on Adaboost is still poorer than the ensemble based on Bagging in all datasets except in iris dataset. The reason is due to that the performance of base classifier *of* Adaboost is similar with that of the Bagging on the iris dataset**.** [Tab.2] shows the average error rate on testing output for all datasets. We can see that the average error rate will decrease with the layer increase from [Tab. 2]. [Fig.10] shows the corresponding experimental results. (See [Fig. 10], labels 1, 2 are the results of the diabetes dataset

based on Bagging and Adaboost respectively, labels 3,4 are Wisconsin Breast Cancer dataset, labels 5, 6 are Iris dataset, labels 7, 8 are Glass dataset and labels 9, 10 are liver disorder dataset.) The experimental results show that the multilayer ensemble pruning model can improve the performance of the ensemble. Besides, if more diverse base classifiers can be provided, the performance of the multilayer ensemble pruning model will be significantly improved.

Dataset	**Ensemble algorithm**	**First layer**	**Second layer**	**Third layer**	**Fourth layer**
Diabetes	Bagging	0.27482	0.23257	0.22985	0.22922
	Adaboost	0.39169	0.28175	0.27232	0.27165
Wisconsin Breast Cancer	Bagging	0.06632	0.036813	0.035918	0.034695
	Adaboost	0.10038	0.044865	0.044191	0.044186
Iris	Bagging	0.061905	0.053333	0.051429	0.048571
	Adaboost	0.060635	0.035556	0.033651	0.032698
Glass	Bagging	0.40595	0.35988	0.35747	0.35671
	Adaboost	0.51315	0.46786	0.46396	0.45471
Liver Disorder	Bagging	0.3759	0.31469	0.30946	0.3093
	Adaboost	0.42866	0.34787	0.32107	0.31839

Table 2: The average errors rate of test set on different dataset

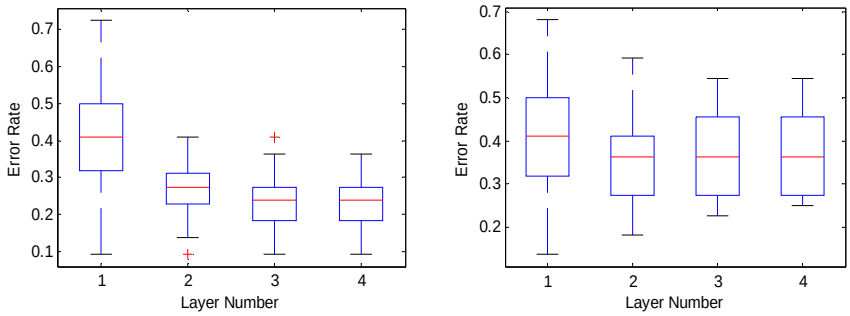

Figure 11: The experimental results of glass dataset based on Bagging, 4 layer model and 21 nodes at each layer, the left shows the error rate of the validate set, the right shows the error rate of the test set

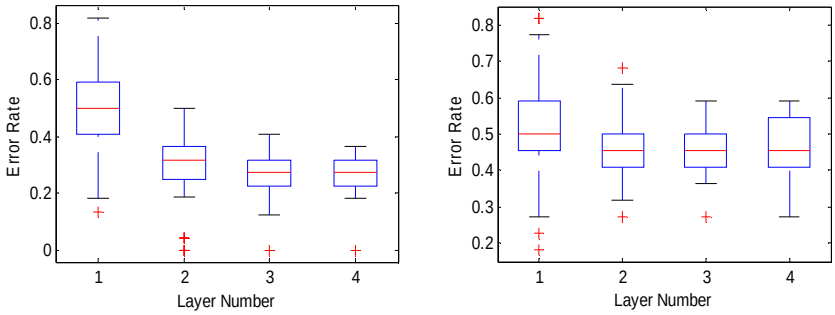

Figure 12: The experimental results of glass dataset based on Adaboost, the left shows the error rate of the validate set, the right shows the error rate of the test set

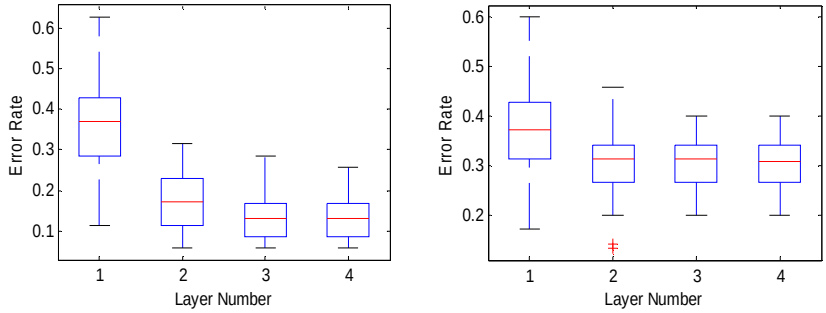

Figure 13: The experimental results of liver disorder dataset based on Bagging, the left shows the error rate of the validate set, the right shows the error rate of the test set

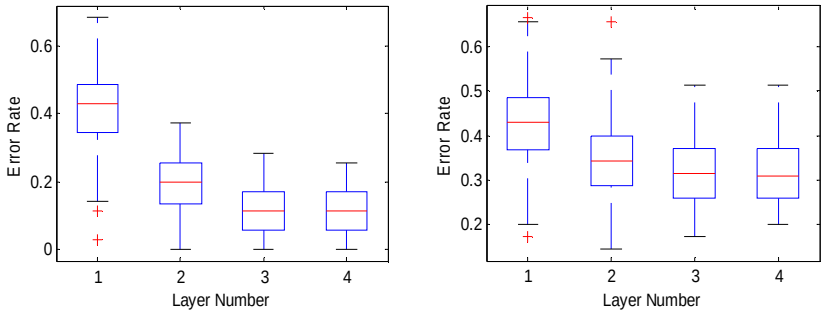

Figure 14: The experimental results of liver disorder dataset based on Adaboost, the left shows the error rate of the validate set, the right shows the error rate of the test set

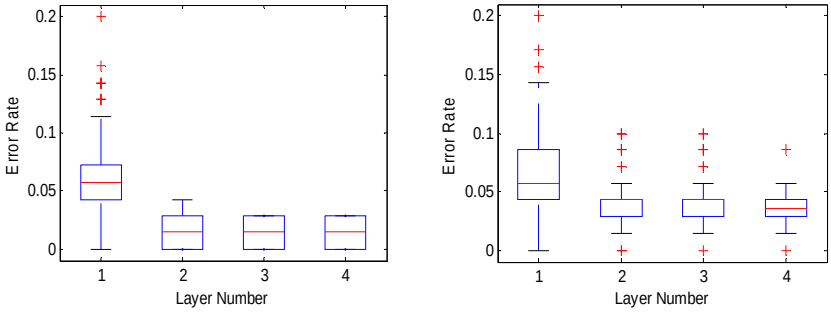

Figure 15: The experimental results of Wisconsin breast cancer dataset based on Bagging, the left shows the error rate of the validate set, the right shows the error rate of the test set

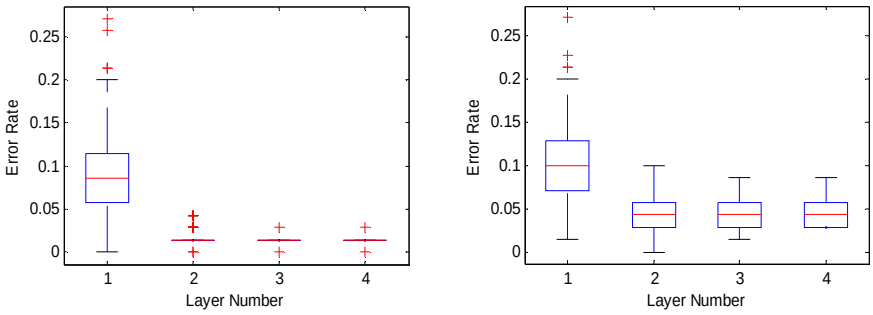

Figure 16: The experimental results of Wisconsin breast cancer dataset based on Adaboost, the left shows the error rate of the validate set, the right shows the error rate of the test set

5 Conclusion

This paper proposed a novel multilayer ensemble pruning model via multi-sub-swarm particle swarm optimization. Several UCI datasets were used to test the performance of the multilayer ensemble pruning model. The experimental results showed that the generalization performance of the multilayer ensemble pruning model is better than the original ensemble. The performance of each layer improved with the increas of the layer number. However, the experimental results showed that the performance of the pruning technique depends on the diverse base classifiers. The proposed model cannot avoid this problem. If more diverse base classifiers are provided, the proposed model will play a more important role for the whole ensemble. In future, we will focus on how to generate more diverse base classifiers and further improve the diversity of the each layer in the proposed model. In this way, the performance of the proposed model will be hugely improved.

Acknowledgements

The authors gratefully acknowledge the support of the National Science Foundation of China (Nos. 60772130) and the "211 Project" of Anhui University by Chinese Government.

References

[Agrafiotis and Cedeño, 2002] Agrafiotis, D. K. and Cedeño, W.:"Feature Selection for Structure-activity Correlation using Binary Particle Swarms"; Journal of Medicinal Chemistry, 45, (2002) 1098-1107.

[Agresti, 1990] Agresti, A.: "Categorical Data Analysis"; John Wiley and Son. Inc. (1990).

[Breiman, 1996] Breiman, L.: "Bagging Predictors"; Machine Learning, 24, 2, (1996) 123-140.

[Breiman 1998] Breiman, L.: "Arcing classifiers"; Annals of Statistics, 26, (1998) 801-849.

[Cohen, 1960] Cohen, J.:"A Coefficient of Agreement for Nominal Scales"; Educational and Psychological Meas., 20, (1960), 37-46.

[Duin, 2002] Duin, R.P.W.:"The Combining Classifier: to Train or not to Train?"; in Proceedings of 16th International Conference on Pattern Recognition, (2002).

[Huang, 1996] Huang, D.S.: "Systematic Theory of Neural Networks for Pattern Recognition"; Publishing House of Electronic Industry of China, Beijing, (1996).

[Huang, 1997] Huang, D. S.: "The United Adaptive Learning Algorithm for the Link Weights and the Shape Prameters in RBFN for Pattern Recognition"; International Journal of Pattern Recognition and Artificial Intelligence, 11, 6, (1997), 873-888.

[Huang, 1999] Huang, D.S.: "Radial Basis Probabilistic Neural Networks: Model and Application "; International Journal of Pattern Recognition and Artificial Intelligence, 13, 7, (1999), 1083-1101.

[Eberhart and Shi, 1998] Eberhart, R. C. and Shi, Y. H.:"Evolving Artificial Neural Networks"; in International Conference on Neural Networks and Brain, Beijing, P.R.C., (1998).

[Eberhart and Hu, 1999] Eberhart, R. C. and Hu, X.:"Human Tremor Analysis using Particle Swarm Optimization"; in Proceeding Congress on Evolutionary Computation, Washington, DC: Piscataway, (1999).

[Freund and Schapire, 1996] Freund, Y. and Schapire, R.,"Experiments with a New Boosting Algorithm"; in Proceedings of the International Conference in Machine Learning, San Francisco, CA: Morgan Kaufmann (1996).

[Ho, 1998] Ho, T.K.:"The Random Space Method for Constructing Decision Forests"; IEEE Transactions on Pattern Analysis and Machine Intelligence, 20,8, (1998), 832-844.

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. C.: "Particle Swarm Optimization"; in Proceedings Of IEEE International Conference on Neural Networks (ICNN). Perth, Australia,(1995).

[Kennedy and Eberhart, 1997] Kennedy, J. and Eberhart, R. C.:"A Discrete Binary Version of the Particle Swarm Algorithm"; in Proceeding 1997 Conference on Systems, Man, and Cybernetics, Piscataway, (1997).

[Kittler et al., 1998] Kittler, J., Hatef, M., Duin, R., and Matas, J.:"On Combining Classifiers"; IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 3, (1998), 226-239.

[Kuncheva et al., 2002] Kuncheva, L.I., Skurichina, M., Duin, R.P.W.:"An experimental Study on Diversity for Bagging and Boosting with Linear Classifiers"; Information Fusion, 3,2, (2002), 245-258.

[Kuncheva and Whitaker, 2003] Kuncheva, L.I. and Whitaker, C.J. :"Measures of Diversity in Classifier Ensembles and their Relationship with the Ensemble Accuracy"; Machine Learning, 51, 2,(2003), 181-207.

[Margineantu and Dietterich, 1997] Margineantu, D. D. and Dietterich, T.G.,"Pruning Adaptive Boosting", in 14th International Conference on Machine Learning, (1997).

[Merz and Murphy, 1998] Merz, C. and Murphy, P.:"`UCI repository of machine learning databases";, www.ics.uci.edu/mlearn/MLRepository.html, (1998).

[Mukherjee and Fine, 1996] Mukherjee, S. and Fine,T.L.: "Ensemble Pruning Algorithms for Accelerated Training"; in IEEE International Conference on Neural Networks,(1996).

[Niklas and Paul, 2007] Niklas, L. and Paul, D.: "Evaluating Learning Algorithms and Classifiers", International Journal of Intelligent Information and Database Systems, 1,1, (2007), 37-52.

[Rogova, 1994] Rogova, G.: "Combining the Results of Several Neural Network Classifiers"; Neural Networks 7,5, (1994), 777-781.

[Ruta and Gabrys, 2001] Ruta, D. and Gabrys, B." Analysis of the Correlation between Majority Voting Error and the Diversity Measures in Multiple Classifier Systems"; in Proceedings of the 4th International Symposium, on Soft Computing, Paisley, UK, (2001).

[Ruta, 2003] Ruta, D.:"Multilayer Selection-Fusion Model for Pattern Classification"; in Proceedings of the IASTED Artificial Intelligence and Application Conference 2003. Benalmadena, Spain, (2003).

[Ruta and Gabrys, 2005] Ruta, D. and Gabrys,B.:"Classifier selection for majority voting"; Information Fusion, 6,1, (2005), 63-81.

[Schapire et al., 1998] Schapire, R.E., Freund, Y., Bartlett, P. and Lee, W.S.: "Boosting the Margin: a New Explanation for the Effectiveness of Voting Methods"; Annals of Statistics, 26,5,(1998), 1651-1686.

[Sharkey et al., 1997] Sharkey, A.J.C. and Sharkey, N.E.:"Combining Diverse Neural Net";. The Knowledge Engineering Review 12,3, (1997)231-247.

[Shi and Eberhart, 1998] Shi, Y. H. and Eberhart, R. C.:"A Modified Particle Swarm Optimizer"; in Proceeding of IEEE World Conference on Computational Intelligence, Anchorage, Alaska, May, (1998).

[Shipp and Kuncheva, 2002] Shipp, C. A. and L. I. Kuncheva,L. I.:" Relationships between Combination Methods and Measures of Diversity in Combining Classifiers"; Information Fusion 3,2, (2002), 135-148.

[Ursem, 1999] Ursem, R. K.:"Multinational Evolutionary Algorithms"; in Proceedings of Congress of Evolutionary Computation, (1999).

[Wang et al., 2005] Wang, J.Q., Wu, X.D. and Zhang, C.Q.: "Support Vector Machines based on K-means Clustering for Real-time Business Intelligence Systems"; International Journal of Business Intelligence and Data Mining, 1,1,(2005), 54 – 64.

[Zeuobi and Cunningham, 2001] Zeuobi, G. and Cunningham, P."Using Diversity in Preparing Ensembles of Classifiers based on Different Feature Subsets to Minimise Generalisation Error"; in Proceedings of the 12th European Conference on Machine Learning, (2001).

[Zhang et al., 2006a] Zhang, J. R., Zhang, J., Lok, T. M. andLyu, M. R.:"A Hybrid Particle Swarm Optimization - Back-propagation Algorithm for Feedforward Neural Network Training"; Applied Mathematics and Computation, 185,2, (2006), 757-1186.

[Zhang et al., 2006b] Zhang, J., Huang, D.S., Lok, T. M. and Lyu, M.R.:"A Novel Adaptive Sequential Niche Technique for Multimodal Function Optimization"; Neurocomputing, 69,16-18,(2006) 2396-2401.

[Zhang et al., 2007] Zhang, J., Huang, D.S., Liu, K. H.:"Multi-Sub-Swarm Particle Swarm Optimization Algorithm for Multimodal Function Optimization"; IEEE Congress on Evolutionary Computation 2007 (CEC2007), Singapore, September 25-28 (2007).

[Zhou et al., 2002] Zhou, Z. H., Wu, J.X. and Tang, W.: "Ensembling Neural Networks: Many Could be Better than All"; Artificial Intelligence,137,1-2, (2002), 239-263.

[Zhou and Tang, 2003] Zhou, Z.H. and Tang, W.:"Selective Ensemble of Decision Trees"; in 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC. Chongqing, China, (2003).