International Journal of Computational Intelligence Systems, Vol. 7, Supplement 2 (July 2014), 35-43

Adaptive generalized ensemble construction with feature selection and its application in recommendation

Jin Tian, Nan Feng*
Department of Information Management & Management Science, Tianjin University
Tianjin, 300072, P.R. China
E-mail: fengnan@tju.edu.cn

Received 3 November 2013

Accepted 10 March 2014

Abstract

This paper presents an adaptive generalized ensemble method with refined feature selection strategy and self-adjusted mechanism for ensemble size. The coevolutionary algorithm is introduced to optimize the ensemble and the feature weighting. There are two stages in the proposed method. In the coevolutionary stage, a component network corresponds to a subpopulation and the feature set is designed in another subpopulation. All subpopulations are coevolved simultaneously. Moreover, the study on the ensemble size is conducted in the structure refining stage. Finally, we apply the proposed approach to a recommendation task. Experimental results indicate that the proposed algorithm can achieve good classification performance, small feature subsets and compact ensemble structure.

Keywords: Ensemble learning, Feature selection, Coevolution, Recommendation

1. Introduction

With the proliferation of Internet technology, much information has been available online. The personalized recommendation, which has been essential for plenty of commercial and social websites, such as Amazon, eBay, and Yotube, can provide related information or product suggestions which are satisfied users, and can allow websites to better serve their customers and remain competitive [1,2]. Generally, the recommendation based on the user ratings can be treated as a classification or predict task, in which the classifiers or predictors are commonly developed by the data mining methods to explore the potential user interesting patterns and user relations by using the rating data.

Ensemble learning is currently an active research area in the data mining and neural networks communities for classification and clustering tasks. The ensemble methods combine multiple component classifiers to obtain better classification or predictive performance. Dietterichhas presented the ensemble learning which integrated multiple classifiers into a whole ensemble model as one of the four most important directions in the machine learning research [3]. Numerous of ensemble approaches have been investigated for personalized recommendation.

Moreover, many real world problems involve in large amounts of input features and modeling with the rating data also introduces a number of Big Data challenges. These rating data are usually both in high volume and high dimension. Many of the enormous ratings might be redundant which result in both

* Corresponding author: fengnan@tju.edu.cn

degrading the predict performance and decelerating the learning speed. Meanwhile, it is difficult to find prominent ratings that provide sufficient information to depict the user's interest patterns. The feature selection attempt to reduce the high dimensions and to find a suitable subset of features that performs best in the classification tasks. Thus feature selection strategies are fully developed in complex classification problems. In addition, the suitable ensemble size might be variable for the scales of different datasets, although it has been suggested in the literature that much of the reduction in error appears to have occurred after ten to fifteen classifiers [4]. Thus, both the ensemble size and the selected feature should be adjusted during the optimal process of the ensemble model, which aims to improve the ensemble performance and speed the training time, especially in complex high-dimensional datasets.

In this paper, we propose an adaptive generalized ensemble learning method (referred as AGEM), which aims to execute the ensemble optimization and the feature subset selection simultaneously, and adjust the ensemble size dynamically during the training process according to different datasets. The whole AGEM consist of two stages: the ensemble optimization stage and the structure refining stage. In the first stage, the optimization is executed by using a cooperative coevolutionary algorithm (Co-CEA) with a multi-population paradigm. Initially, M components generate M subpopulations and the feature set is designed in another subpopulation. All the (M+1) subpopulations are evolved simultaneously. During the cooperative process, individuals of each subpopulation are evaluated based on the performance they achieve in conjunction with representatives from the other subpopulations. In the second stage, a further study on the ensemble size is conducted to find the appropriate ensemble size for different datasets on the structure refining process. The performance of the proposed algorithm is verified on both UCI datasets and the recommendation dataset.

The remainder of the paper is organized as follows. In the next section, a brief literature review of ensemble learning is provided. Section 3 elaborates the details of the AGEM. The simulated experiments are described and their results discussed in Section 4 for evaluating the effectiveness of AGEM in comparison with other ensemble algorithms. Finally, Section 5 concludes the key points of the paper and offers suggestions for the future research.

2. Literature review

In ensemble learning, there are two main techniques, Boosting and Bagging, to train the components of the ensemble and build the ensemble model. The Boosting approach trains components incrementally with those samples that previous components misclassify, while the Bagging generates the components concurrently by selecting randomly subset of the original training set. Some research work has suggested that in many domains a well-designed ensemble model may generally outperform the single component in the ensemble, especially when the components are quite different [4]. Wei et al. have proposed an ensemble approach by combining the predictions made by Positive Naïve Bayes with the classifier of positive example-based learning for the unlabeled examples [5]. García-Pedrajas et al. have used the supervised projections of random subspaces to construct ensemble, which combines the philosophy of boosting to generate supervised projection based on the misclassified samples, and then trains using all available samples in the space given by the supervised projections [6]. Bock et al. have adopted a statistical technique for nonparametric, the generalized additive model (GAM), as the ensemble component and proposed three GAM ensemble classifiers for binary classification based on Bagging, random subspace and a combination of both [7].

Additionally, many in-depth researches on the ensemble have revealed its effectives in the recommendation applications, and various ensemble approaches were investigated. Wang et al. have conducted a comparative assessment of the performance of three popular ensemble methods for sentiment classification [8]. Zheng et al. have proposed a personalized news recommendation framework using ensemble hierarchical clustering to provide attractive recommendation results, in which users are separated into several groups based on their reading histories and the ensemble model is constituted by the hierarchies of multiple user groups [9]. Tsai and Hung have applied two clustering techniques into three ensemble methods (the cluster-based similarity partitioning algorithm, the hypergraph partitioning algorithm, and majority voting)

to analyze the effective of cluster ensembles to collaborative filtering recommendation [10].

A prominent current in ensemble study is the combination ensemble learning with the evolutionary computation. Liu et al. have used mutual information to analysis similarity between components, and a diverse population of components have been evolved by adjusting the fitness sharing with mutual information [11]. García-Pedrajas et al. have developed an approach to ensemble design by means of coevolutionary algorithm to encourage collaboration among component networks [12]. Soares et al. have proposed genetic algorithm and simulated annealing based approaches for the automatic development of neural network ensembles, in which the selection of the best subset of the models is optimized by considering several key factors of ensemble, such as diversity, training ensemble members and combination strategy [13].

This paper proposes an adaptive generalized ensemble model, in which both the ensemble structure and the feature subsets are optimized dynamically during the training process. A specially designed Co-CEA is executed to find the suitable feature subsets and the optimal ensemble size.

3. AGEM Configuration

The proposed idea is that the active ensemble model with prominent features is obtained by a specially designed Co-CEA with the multi-population paradigm. Radial Basis Function Neural Network (RBFNN) is utilized as the component network. The weights between the hidden nodes and the output layer are computed by the pseudo-inverse method [15]. For simpleness, the proposed algorithm adopts the majority vote method as the ensemble output combination method. The best individuals provided from the subpopulations compose the concise ensemble model with selected feature subsets.

3.1. *Encoding*

In AGEM, each component network generates identically one subpopulation and there is another subpopulation designed for the feature set. A matrix-form of the mixed encoding is adopted for subpopulations of individuals. In this mixed encoding representation, the hidden nodes and the radius widths of the RBFNN are encoded as real-valued encoding matrices while the control vector is an additional binary string that attached to the matrix, which indicates the hidden nodes are valid or not. Thus in the t^{th} subpopulation the chromosome of the l^{th} individual, $\boldsymbol{P}_t^l$, is represented as a matrix of size $Nc_t \times (m+2)$ as below:

$$\boldsymbol{P}_t^l = \left[\boldsymbol{c}_t^l \quad \boldsymbol{\sigma}_t^l \quad \boldsymbol{b}_t^l \right],\ t = 1,2,\ldots,M\ ,\ l = 1,2,\ldots,L \qquad (1)$$

where $\boldsymbol{c}_t^l = [\boldsymbol{c}_t^{li}]_{Nc_t \times m}$ and $\boldsymbol{\sigma}_t^l = [\sigma_t^{li}]_{Nc_t \times 1}$ are the centers and radius widths of the hidden nodes; $\boldsymbol{b}_t^l$ is the control vector with the elements as 1 or 0 which means the corresponding hidden node is active or not in the design of the network structure, $i = 1,2,\ldots,Nc_t$. M is the initial size of ensemble and L is the size of the population. Nc_t is the hidden node number of the individuals in t^{th} subpopulation, and m is the size of features.

In the feature set subpopulation, the chromosome is encoded as a binary matrix, every row of which denotes a feature weighting vector to a certain component in the ensemble. The l^{th} individual in the $(M+1)^{\text{th}}$ subpopulation is designed as below:

$$\boldsymbol{P}_{M+1}^l = [\boldsymbol{v}_t^l]_{M \times 1} = [v_{tk}^l]_{M \times m},\ l = 1,2,\ldots,L \qquad (2)$$

$\boldsymbol{v}_t^l$ is the feature weighting vector of the t^{th} component network that represented as a vector of m binary bits, where v_{tk}^l indicates that the k^{th} feature is selected or not with 1 or 0 respectively. $k = 1,2,\ldots,m$.

3.2. *Multiobjective Evaluation of Individuals*

In order to favor the cooperation of the components, individuals are evaluated throughout the evolutionary process in a multiobjective paradigm. For individuals in the first M subpopulations and the last subpopulation, different objectives are defined separately, with fully consideration of both classification performance and the cooperation with the rest of the ensemble.

For implementing the coevolution paradigm, a representative in each subpopulation should be contributed to form the complete ensemble $\boldsymbol{\Theta}$. Thus, the fitness of individuals in one subpopulation are calculated in conjunction with the representatives of other subpopulations. The best individual in each subpopulation is chosen as the representative to compose the elite pool $\boldsymbol{\Theta}^* = \{\boldsymbol{P}_1^*, \boldsymbol{P}_2^*, \ldots, \boldsymbol{P}_M^*, \boldsymbol{P}_{M+1}^*\}$, M is the initial ensemble size. Generally, the classification

accuracy and the component diversity are the two main points that should be simultaneously considered in the ensemble training process. Besides, since the proposed model employs a feature selection mechanism, the number of the selected feature is also important for the individuals' fitness evaluation and selection. Three objectives are consequently adopted in the proposed method.

3.2.1 *Classification accuracy*

This objective, which is calculated by the ensemble combination output, aims to measure the performance of individuals in classification tasks when associating with other representatives of the ensemble.

Note that the various feature weighting vectors for the ensemble components are evolved separately in the $(M+1)^{\text{th}}$ subpopulation, and fulfill the feature selection by cooperating with the individuals in other M subpopulations. In a component network, a certain sample's label is determined by the largest output unit among all units of the output layer, while the final ensemble decision is made by the majority voting in the ensemble components.

Thus the t^{th} individual in the t^{th} subpopulation, $\boldsymbol{P}_t^l$, gets its first fitness by calculating the combination output of the estimated ensemble structure $\boldsymbol{\Theta}_t^l = \{\boldsymbol{P}_1^*, \ldots, \boldsymbol{P}_{t-1}^*, \boldsymbol{P}_t^l, \boldsymbol{P}_{t+1}^*, \ldots, \boldsymbol{P}_M^*, \boldsymbol{P}_{M+1}^*\}$. For the individuals in the first M subpopulations, the output of $\boldsymbol{P}_t^l$ can be expressed as $\boldsymbol{y}(\boldsymbol{P}_t^l, \boldsymbol{v}_t^*) = [y_t^{l1}, \ldots, y_t^{ln}] = \boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{v}_t^*)$ and the labels of the samples are decided by this component as $o(\boldsymbol{P}_t^l, \boldsymbol{v}_t^*) = \arg\max\{y_t^{l1}, \ldots, y_t^{ln}\}$, where n is the size of class. Accordingly, the whole ensemble output is $O(\boldsymbol{\Theta}_t^l) = MVot\{o(\boldsymbol{P}_1^*, \boldsymbol{v}_1^*), \ldots, o(\boldsymbol{P}_t^l, \boldsymbol{v}_t^*), \ldots, o(\boldsymbol{P}_M^*, \boldsymbol{v}_M^*)\}$, where *MVot* donates the majority voting and $\boldsymbol{P}_{M+1}^* = [\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_M^*]^T$. In the $(M+1)^{\text{th}}$ subpopulation, $\boldsymbol{P}_t^l = [\boldsymbol{v}_1^l, \ldots, \boldsymbol{v}_M^l]^T$, and the ensemble output is $O(\boldsymbol{\Theta}_{M+1}^l) = MVot\{o(\boldsymbol{P}_1^*, \boldsymbol{v}_1^l), \ldots, o(\boldsymbol{P}_M^*, \boldsymbol{v}_M^l)\}$.

This objective is calculated as follows:

$$Accu_v(\boldsymbol{P}_t^l) = \frac{N_{rv}(\boldsymbol{\Theta}_t^l)}{N_v} \tag{3}$$

where $N_{rv}(\boldsymbol{\Theta}_t^l)$ is the size of samples that classified properly on the validation set by the ensemble structure $\boldsymbol{\Theta}_t^l$ and N_v is the size of the validation set.

In experiments, avoiding that the similar accuracies result in smaller selection pressure in the evolution process, the objective is modified as:

$$f_1(\boldsymbol{P}_t^l) = \alpha(1-\alpha)^{I(\boldsymbol{P}_t^l)} \tag{4}$$

where $I(\boldsymbol{P}_t^l)$ is the inversely sort order of $\boldsymbol{P}_t^l$ based on accuracies of all $\boldsymbol{P}_t^l (t = 1, \ldots, M+1)$. $\alpha \in (0,1)$ is a fixed real number whose default value is 0.4 .

3.2.2 *Diversity measure*

The second objective aims to assess the component diversity and the Yule's Q statistic is used to measure the similarity of two component networks [16].

$$Q_{ij} = \frac{N^{11}N^{00} - N^{01}N^{10} + 1}{N^{11}N^{00} + N^{01}N^{10} + 1} \tag{5}$$

where N^{ab} is the number of samples that classified correctly ($a = 1$) or incorrectly ($a = 0$) by the component i, and correctly ($b = 1$) or incorrectly ($b = 0$) by the component j. Q varies between -1 and 1.

In the first M subpopulations, the diversity of individual $\boldsymbol{P}_t^l$ is computed by measuring the difference between $\boldsymbol{P}_t^l$ and the representatives in other subpopulations. Q_{ts}^l is the Q values calculated by $\boldsymbol{P}_t^l$ and the representative $\boldsymbol{P}_s^*$, $s = 1, \ldots, M$ and $s \neq t$. The average of these Qs is a explicit index that denotes the diversity of $\boldsymbol{P}_t^l$:

$$\bar{Q}_t^l = \frac{\sum_{s=1,\ldots,M;s\neq t} Q_{ts}^l}{M-1} \ (t = 1, \ldots, M) \tag{6}$$

For the individuals in the $(M+1)^{\text{th}}$ subpopulation, the Q value denotes the diversity between two representatives in the elite pool by associating with the feature subsets that provided by the individual $\boldsymbol{P}_{M+1}^l$. And the diversity measure is defined as:

$$\bar{Q}_{M+1}^l = \frac{2 \times \sum_{t=1}^{M} \sum_{s=t+1}^{M} \tilde{Q}_{ts}^l}{M(M-1)} \tag{7}$$

where $\tilde{Q}_{ts}^l$ is the Q value of $\boldsymbol{P}_{M+1}^l$ that assess the diversity between the representative network $\boldsymbol{P}_t^*$ and $\boldsymbol{P}_s^*$ associated with the corresponding feature weighting vectors in $\boldsymbol{P}_{M+1}^l$.

In order to normalize this measure to vary from 0 to 1, the objective is modified as:

$$f_2(\boldsymbol{P}_t^l) = \frac{1-\bar{Q}_t^l}{2} \tag{8}$$

3.2.3 *Feature subset size*

This objective tends to find a compact feature subset and is only assigned to the individuals in the $(M+1)^{\text{th}}$ subpopulation. The average feature subset size selected by the individual $\boldsymbol{P}_{M+1}^l = [\boldsymbol{v}_t^l]_{M\times 1}$ is expressed as:

$$avef_{M+1}^l = \frac{1}{M}\left(\sum_{t=1}^{M} Nf_t^l\right) \tag{9}$$

where Nf_t^l is the feature number selected by the feature weighting vector $\boldsymbol{v}_t^l$, i.e. the number of 1 in $\boldsymbol{v}_t^l$. And the third objective is defined as:

$$f_3(\boldsymbol{P}_{M+1}^l) = \frac{1}{2}\times \exp\left(-\frac{avef_{M+1}^l}{m}\right) \tag{10}$$

m is the total number of features. This objective is employed to keep a balance between the classification performance of the estimated ensemble model and the complexity of the selected feature size.

3.3. *Implementation*

3.3.1 *Coevolution stage*

Bootstrap resampling is applied to generate M training subsets from the original training data. The initial component networks in the ensemble are then trained with these data subsets. L individuals are generated for one initial component network to create one subpopulation with the control vectors initialized as binary bits randomly. The individuals in the $(M+1)^{\text{th}}$ subpopulation are initialized as binary matrixes.

The selection operation adopted in AGEM is based on the Pareto ranking with the multiobjective optimality [17]. And the tournament selection is used to select mating individuals. In addition, the elitist selection [18] is utilized to keep the best individuals survive directly to the next generation, which will save the optimal ones once they are found in the evolutionary process.

The proposed algorithm adopts the two-point crossover to exchange some gene between two individuals to produce offspring. The individuals selected for the crossover operation are divided into groups and each group has only two individuals. For every group two crossover points are determined randomly. The genes of the network structure part in the two parents between the two points are exchanged to produce two offspring. Note that all bits of the control vectors and the feature weighting matrixes should not be zeros for both offspring.

In the first M subpopulations, a special designed mutation operator is implemented in order to make the real-valued encoding part and the binary-valued part mutate separately. A ratio, p_{ad}, has been employed to determine which type of mutation occurs: parametric mutation in the real-valued part or structure mutation in the binary bits. For a hidden node $\boldsymbol{c}_t^{li}$ in $\boldsymbol{P}_t^l$, a random number r_{ad} is generated. If $p_{ad} > r_{ad}$, the mutation only alters the control bit which implies a modification of the structure of the component network. If $p_{ad} \le r_{ad}$ and the corresponding control bit is 1, the parametric mutation is carried out to make changes on the real-valued genes without modifying the component's topology.

$$\boldsymbol{c}_t^{li\prime} = \boldsymbol{c}_t^{li} + N(0,1)\times(\boldsymbol{c}_t^{*,i} - \boldsymbol{c}_t^{li}) \tag{11}$$

$$\sigma_t^{li\prime} = \sigma_t^{li} + N(0,1)\times(\sigma_t^{*,i} - \sigma_t^{li}) \tag{12}$$

where $\boldsymbol{c}_t^{li\prime}$ and $\sigma_t^{li\prime}$ are the mutated values, $\boldsymbol{c}_t^{li}$ and σ_t^{li} are the current values, $\boldsymbol{c}_t^{*,i}$ and $\sigma_t^{*,i}$ are the corresponding hidden node centers and radiuses of the hidden nodes in the elite pool. The random number $N(0,1)$ obeys the standard normal distribution.

In the $(M+1)^{\text{th}}$ subpopulation, a mutation ratio, p_m, has been introduced to decide whether the mutation occurs on a certain individual. For every individual, a random number r_l is generated. If $r_l < p_m$, the mutation occurs in the l^{th} individual by inverting q% of the total bits in the binary matrix.

3.3.2 *Heuristic structure refining stage*

Some early researches indicate that the addition of new components to the ensemble does not always improve the classification performance when the ensemble size increases [14]. Moreover, the ensemble size M is usually pre-designed and fixed during the coevolutionary process which might be not suitable for various datasets with different sizes of samples. In the

structure refining stage, we investigate the appropriate size of an ensemble for different datasets.

After the coevolutionary process, we have obtained M representatives from the parallel subpopulations and one representative from the feature weighting subpopulation to compose the ensemble. The performances of the (M+1) component representatives are then evaluated by two indexes:

(i) Independence performance

This index aims to assess the components when they complete the classification task independently with no information from others. For each elite component $\boldsymbol{P}_t^*$ ($t = 1,\ldots,M$), there is a corresponding feature weighting vector, $\boldsymbol{v}_t^*$, which is the t^{th} row vector of $\boldsymbol{P}_{M+1}^*$ .This objective is represented as the ratio of samples that are correctly classified in the validation set by $\boldsymbol{P}_t^*$ assisted with the feature weight vector $\boldsymbol{v}_t^*$:

$$Obj_1(\boldsymbol{P}_t^*) = \frac{N_{rv}(\boldsymbol{P}_t^*, \boldsymbol{v}_t^*)}{N_v} \tag{13}$$

(ii) Diversity of the component with other representatives

We adopt the Pairwise Failure Crediting (PFC) method [19] to get the diversity value between the elite representatives.

$$Obj_2(\boldsymbol{P}_t^*) = PFC_t = \frac{\sum_{j=1,\ldots,M;\, j\neq t} h_{tj}}{M-1} \tag{14}$$

where h_{tj} measures the diversity between $\boldsymbol{P}_t^*$ and $\boldsymbol{P}_j^*$ using the Hamming distance.

The elite components are evaluated by the two indexes and ranked by the Pareto ranking. The component that has the least contribution is removed. The coevolution stage and the structure refining stage are operated in turn until the validation accuracy is decreased. The retained representatives in the elite pool are output as the final ensemble model.

4. Experimental Studies

Experiments were conducted on both UCI datasets and the recommendation dataset to study the behavior of the proposed method. The experiment parameters used in the AGEM algorithm were set as follows. The population size L was 50, the maximum generations G was 200, and the initial ensemble size M=25. p_c was 0.8, p_m was 0.2, and p_{ad} was 0.6.

4.1. *Experiment* 1

Experiments were carried out on 8 datasets from the UCI Repository, which have different number of samples (from 208 to 5000) and features (from 9 to 60).

Firstly, the proposed algorithm was compared with the initial ensemble which has the fixed ensemble size and uses the whole feature sets. Table 1 shows the performance of the two algorithms, including the testing accuracies, the number of the feature set, Num_f, and the reduced ensemble size M*.

Table 1. Performance of AGEM and Initial Ensemble

Dataset	Initial Ensemble		AGEM		
	Test	Num_f	Test	Num_f	M*
Crx	0.7102	15	0.8530	9.613	13.47
German	0.6581	24	0.7473	13.22	18.83
Glass	0.5560	9	0.6970	5.077	12.70
Iono	0.8596	34	0.9324	19.11	8.400
Sonar	0.7123	60	0.7591	28.59	11.37
Vehicle	0.7236	18	0.7362	9.707	23.13
Votes	0.9433	16	0.9529	6.928	10.23
Wave	0.8045	21	0.8609	13.90	18.39
Ave.	0.7460	24.63	0.8173	13.27	14.56

The experimental results show that the average testing accuracies achieved by the AGEM increase 9.59% compared to the initial ensemble. Meanwhile, the final value of M is above 15 only on three datasets, and on the other datasets the proposed algorithm yields concise ensemble models whose size are only about 10. The experimental results reveal the dependence of M on the dataset size, and there exists a subset of primary components in the ensemble that can perform as well as all components. In addition, the average number of the selected features is about half of the original. Specially, in the two datasets Sonar and Votes, more than half of the features are removed in the proposed models.

Secondly, experiments were carried out with the target of testing the proposed method against some conventional ensemble methods, such as AdaBoost (AB) [20], BAgging (BA) [21], EnsembleSelection (ES) [22], and RandomSubspace (RS) [23]. Table 2 reports the testing accuracies of the AGEM and these compared

algorithms. The highest accuracy in each dataset is outlined in bold.

Table 2. Performance of AGEM and other ensemble algorithms

Datasets	AGEM	AB	BA	ES	RS
Crx	**0.8530**	0.8527	0.5313	0.7879	0.7925
German	**0.7473**	0.7293	0.7444	0.7233	0.7276
Glass	0.6970	**0.7002**	0.6995	0.6692	0.6940
Iono	0.9324	**0.9332**	0.9143	0.9032	0.9162
Sonar	0.7591	**0.8246**	0.7624	0.7502	0.7202
Vehicle	**0.7362**	0.6785	0.6623	0.7098	0.7229
Votes	0.9529	0.9473	0.9476	**0.9540**	0.9479
Wave	**0.8609**	0.6636	0.8175	0.8054	0.8190
Ave.	0.8173	0.7912	0.7599	0.7879	0.7925

As shown in Table 2, the AGEM is able to achieve useful results on most datasets. The performance of the proposed model is clearly better than the results of the other compared algorithms on Crx, German, Vehicle, Wave, and in average as well. Totally, the AGEM performs competitively on most datasets compared with other ensemble algorithms.

4.2. *Experiment* 2

In order to assess the proposed algorithm for the ratings user-item matrix, we used Movielens datasets which is available on the website of GroupLens Research Group. We adopted the rating dataset that consists of approximately 100 thousand ratings for 1682 movies by 943 users. The rating scores were on a numeric five-point scale with (1, 2, 3, 4, 5). From the classification perspective, the movies that one user has given rating scores constitute the training samples and the other users' rating to these movies are the features of the samples. The target of the classification to recommendation is to predict the exact rating or identify the user's potential preference about those movies that he doesn't give the rating.

In the experiment, we selected 100 most active users as the target user. Other users' ratings for those movies that the target users had rated are the training and the testing subsets. The two subsets were not overlapped and the ratio of them is about 10:1. The average size of the training datasets was about 1300 and each sample has 942 features (ratings). According to the rating scores, we converted the target ratings into two classes: 'the user like this movie' (the rating scores are 3,4,5) and 'the user dislike this movie' (the rating scores are 1,2). Fig.1 gives the average testing performance of 100 users obtained by AGEM and the compared algorithms.

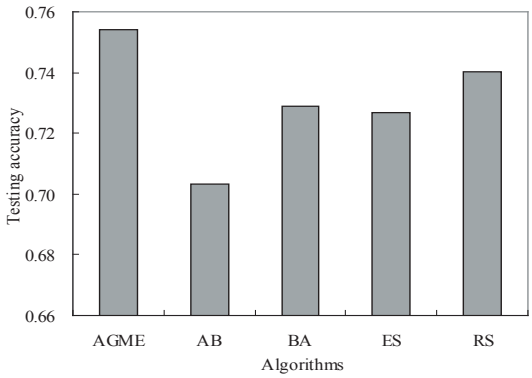

Fig.1. The average testing accuracy of the five algorithms

The experimental results indicate that the AGEM achieves the best performance among the five ensemble algorithms. Moreover, about half of the features are removed by the AGEM, which means the proposed algorithm can filter the irrelevant ratings and obtain higher classification accuracy with a more compact rating set.

5. Conclusions

This paper has presented an effective generalized ensemble learning model which selects a refined feature subset for each component and self-adjusted mechanism for ensemble size. The proposed model is based on multi-population coevolution and applied to the recommendation problem. The Co-CEA is introduced to optimize both the ensemble structure and the feature weighting. A component network of the ensemble in the proposed model corresponds to a separate subpopulation and the feature set is designed in another subpopulation. All subpopulations are coevolved in parallel. There are two stages in the modeling process: during the cooperative stage, individuals in one subpopulation are evaluated in the context of the ensemble with the representatives of other subpopulations, while the structure refining stage offers a further study on the ensemble size and the suitable ensemble size is then modified by removing the less-contribution component networks.

The performance has been thoroughly verified over a set of real-world datasets with different features. Experimental results illustrated that the classification performance of the proposed algorithm is superior to some conventional ensemble algorithms on both UCI datasets and recommendation dataset. And the proposed algorithm can also select small feature subsets and modify compact ensemble structure. To sum up, the AGEM offers a competitive and powerful classification approach for both traditional classification problems and recommendation tasks.

There are two issues to be addressed in the future research. One is to adopt the real version of the feature weighting vectors. The other is the introduction of subspace learning to refine the input feature space of each component, and increase the diversity within the ensemble components.

Acknowledgements

The work was supported by the National Science Fund for Distinguished Young Scholars of China (Grant No. 70925005) and the General Program of the National Science Foundation of China (Grant Nos. 71001076, 71101103, and 71271149). The authors are very grateful to all anonymous reviewers whose invaluable comments and suggestions substantially helped improve the quality of the paper.

References

1. P. Resnick, H.R. Varian, Recommender systems, *Communications of the ACM* 40 (3) (1997) 56–58.
2. J.B. Schafer, J. Konstan, J. Riedl, Recommender systems in e-commerce, in *Proc. of the ACM Conference on Electronic Commerce*, Denver, Colorado, 1999, pp. 158–166.
3. T.G. Dietterich, Machine learning research: four current directions, *AI Magazine* 18(4) (1997) 97-136.
4. A. Krogh and J. Vedelsby, Neural network ensembles, cross validation, and active learning, *Advances in Neural Information Processing Systems* 7(1995) 231-238.
5. C.P. Wei, H.C. Chen, and T.H. Cheng, Effective spam filtering: A single-class learning and ensemble approach, *Decision Support Systems* 45(3) (2008) 491-503.
6. N. García-Pedrajas, J. Maudes-Raedo, C. García-Osorio, *et al*., Supervised subspace projections for constructing ensembles of classifiers. *Information Sciences* 193 (2012) 1–21
7. K.W. De Bock, K. Coussement, D.V. den Poel, Ensemble classification based on generalized additive models. *Computational Statistics and Data Analysis* 54 (2010) 1535-1546.
8. G. Wang, J.S. Sun, J. Ma, et al., Sentiment classification: The contribution of ensemble learning, *Decision Support Systems* (2013), http://dx.doi.org/10.1016/j.dss.2013.08.002
9. L. Zheng, L. Li, W.X. Hong. PENETRATE: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications* 40 (2013) 2127–2136.
10. C.F. Tsai and C. Hung. Cluster ensembles in collaborative filtering recommendation. *Applied Soft Computing* 12 (2012) 1417–1425.
11. Y. Liu, X. Yao, Q. Zhao, *et al*., Evolving a cooperative population of neural networks by minimizing mutual information, In: *Proc. of the Congress on Evolutionary Computation*, Seoul, Korea, (2001) 384-389.
12. N. García-Pedrajas, C. Hervás-Martínez, and D. Ortiz-Boyer, Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification, *IEEE Transactions on Evolutionary Computation* 9 (2005) 271-302.
13. S. Soares, C.H. Antunes, R. Araújo, Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development, *Neurocomputing* 121 (2013) 498–511.
14. D. Opitz, and R. Maclin, Popular ensemble methods: an empirical study, *Journal of Artificial Intelligence Research* 11 (1999) 169-198.
15. D. Casasent and X.W. Chen, Radial Basis Function Neural Networks for Nonlinear Fisher Discrimination and Neyman-Pearson Classification, *Neural Networks* 16 (2003) 529-535.
16. L.I. Kuncheva and C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine Learning* 51(2) (2003) 181–207.
17. S.G. Ficici and J.B. Pollack, Pareto optimality in coevolutionary learning, In: *European Conference on Artificial Life*, (2001) 316-325.
18. M.Q. Li, J.S. Kou, *et al*., The Basic Theories and Applications in GA. *Science Press, Beijing* (2002)
19. A. Chandra, X. Yao, Ensemble Learning Using Multi-Objective Evolutionary Algorithms, *Journal of Mathematical Modelling and Algorithms* 5(4) (2006) 417-425
20. Y. Freund, R. Schapire, Experiments with a New Boosting Algorithm. in *Thirteenth International Conference on Machine Learning*, 1996, pp. 148-156.
21. L. Breiman, Bagging Predictors, *Machine Learning* 24(2) (1996) 123-140
22. R. Caruana, A. Niculescu, G. Crew, *et al.*, Ensemble selection from libraries of models. in: Proceedings of the 21st International Conference on Machine Learning, 2004, pp. 18-25.

23. T.K. Ho, The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8) (1998) 832-844.