Advanced Science and Technology Letters
Vol.29 (SIP 2013), pp.50-54
http://dx.doi.org/10.14257/astl.2013.29.10

# A Meta-Learning Approach for Combining Multiple Classifiers

Yeongjoon Kim and Chuleui Hong

Department of Computer Science, Sangmyung University, Seoul, Korea
{yjkim, hongch}@smu.ac.kr

**Abstract.** This paper presents an approach for building a multi-classifier system in a Mean Field Genetic Algorithm (MGA)-based inductive learning environment. Multiple base classifiers are combined to build a multi-classifier system. A base classifier consists of a general classifier and a meta-classifier. The general classifier performs regular classification task. The meta-classifier evaluates classification result of its general classifier and decides whether the base classifier participates into a final decision-making process or not. MGA is a hybrid algorithm of Mean Field Annealing (MFA) and Simulated annealing-like Genetic Algorithm (SGA). It combines benefit of rapid convergence property of MFA and effective genetic operations of SGA.

**Keywords:** Multi-classifier, hybrid algorithms, inductive learning

## 1 Introduction

The main purpose of creating a complex multi-classifier system is to obtain better classification performance than the performance offered by its components – individual base classifiers [1,2]. Doan et. al.[3] and Fan et. al.[4] have explored a multi-strategy learning approach that applies multiple learner modules to a given problem, then combines the predictions of modules using a meta-learner.

We explore a meta-learning approach for building a multi-classifier system in our Mean Field Genetic Algorithm (MGA)-based inductive learning environment. In our approach, several base classifiers are obtained from given training data set by executing MGA-based inductive learning system multiple times. Then, base classifiers are combined to build a multi-classifier system. A base classifier consists of a general classifier and a meta-classifier. The general classifier performs regular classification task. The meta-classifier evaluates classification result of its general classifier and decides whether the base classifier participates into a final decision-making process or not. Mean Field Genetic Algorithm (MGA) is a hybrid algorithm based on mean field annealing (MFA) and genetic algorithm (GA)[5]. MFA has the characteristics of rapid convergence to the equilibrium state while the simulated annealing (SA) takes long time to reach the equilibrium state.

ISSN: 2287-1233 ASTL

## 2 Multi-classifier System

### 2.1 Learning Classification Rules

From give examples, the system learns PROSPECTOR-style rules [6] that have the form, "If E then H with S =s; N = n", where S and N are odds-multipliers, measuring the sufficiency and necessity of E for H. In general, PROSPECTOR rules work with odds instead of probabilities, using the following conversion from probabilities to odds: O(H) = P(H)/(1-P(H)). That is, a probability of 0.75 is converted to odds of 3 (=0.75 / 0.25).

The system learns two kinds of rules from given examples: is-high rules and is-close-to rules. An example of is-high rule is "If is-high (A) then D with S=3; N=0.1", and it produces an odds-multiplier between 3 and 0.1 based on relative highness of value for attribute A of a given example to other examples. An example of is-close-to rule is "If is-close-to (A, a) then D with S = 4; N = 0.2", and it produces an odds-multiplier between 4 and 0.2 based on closeness of value for attribute A to a certain constant a.

A rule-set that consists of learned rules makes a decision as follows: assuming that we have decision candidates $DC=\{D_1,...,D_n\}$ with prior odds $O(D_1),...,O(D_n)$, these odds are updated by firing rules of the rule-set, yielding posterior odds $O(D_1'),...,O(D_n')$, and a decision candidate with the highest posterior odds is selected.

### 2.2 Building a Multi-classifier system

The training process to obtain a base classifier consists of two phases. In the first phase, a general classifier, GC, is learned from given training data set. The training data set for GC, denoted by TGC, is a regular training data set in which each example is represented with a list of attribute values and classification of the example. In the next phase, a meta-classifier, MC, is trained for the training data set, denoted by TMC. TMC is obtained as follows:

Step 1. Classify training examples in TGC with GC. For the prediction of GC for each training example, classify it into one of two classes, 0 or 1, where 0 represents incorrect decision and 1 represents correct decision.

Step 2. Construct training data set TMC for MC as follows:
Element in TMC = (attribute values of an example in TGC, PGC, CPGC)
where PGC is prediction of GC and CPGC is classification of PGC.

For a given unknown example to be classified, if MC classifies prediction of GC as correct one, the base classifier participates into a final decision-making process with classification result of GC. Otherwise it withdraws.

 

## 3 Mean Field Genetic Algorithm

In this paper, a genetic algorithm approach is used to learn appropriate *is-high* rules and *is-close-to* rules from given examples. A population consists of a fixed number of rule-sets and rule-sets themselves are represented in chromosomal representation as ordered sequences of rules $r_1,...,r_n$. Fig. 1 depicts the chromosomal representation of rule-set and rule structure.

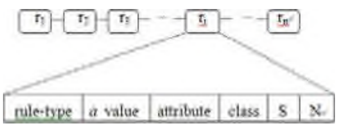


**Fig. 1.** Chromosomal representation of rule-set and rule structure

1-point crossover operator and mutation operator are used as genetic operators. The 1-point crossover operator creates two offspring by exchanging some rules of selected parents. The mutation operator selects a rule $r$ from a rule-set and replaces it with a newly generated rule $r'$. Parents are selected based on their fitness, using the popular roulette wheel method. Fitness of a rule-set is evaluated by the percentage of examples the rule-set classifies correctly.

We modified GA such that the new evolved state is accepted with a Metropolis criterion like simulated annealing in order to keep the convergence property of MFA. The modified GA is called SGA. $\Delta C$ is the cost change of new state from old state. It is made by subtracting the cost of new state from that of old one. $T$ is the current temperature.

$$\Pr[\Delta C \text{ is accepted}] = \min\left(1, \exp\left(\frac{\Delta C}{T}\right)\right)$$

MFA is derived from Simulate Annealing(SA) based on mean field approximation method in physics. The spin matrix is made up of $N \times R$ where $N$ is the number of classifications and $R$ is the number of rules. The randomly selected class-$i$ is responsible for updating its spin value, $s_{ip}$. The objective function, $C(s)$, of MFA is same as that of GA. When the rule-sets of all classes are classified correctly, the objective value will be 0.

$$C(s) = \sum_{i=0}^{N-1} \sum_{j \neq i} \sum_{p=0}^{R-1} s_{ip} s_{jp} p_{ij}$$

$N$ : The number of classes
$R$ : The number of rules
$s_{ip}$ : The probability of class $i$ mapping to rule $p$
$s_{jp}$ : The probability of class $j$ mapping to rule $p$
$p_{ij}$ : The percentage of falsely classified for class $i$ and $j$

A new hybrid algorithm called MGA combines the merits of mean field annealing (MFA) and simulated annealing-like genetic algorithm (SGA). MFA can reach the thermal equilibrium faster than simulated annealing and GA has powerful and various genetic operations such as selection, crossover and mutation. First, MFA is applied on a spin matrix to reach the thermal equilibrium fast. After the thermal equilibrium is reached, the population for GA is made according to the distribution of rules of classes in the spin matrix. Next, GA operations are applied on the population while keeping the thermal equilibrium by transiting the new state with Metropolis criteria. MFA and SGA are applied by turns until the system freeze.

## 4 Experiments

We performed some experiments to evaluate performance of the multi-classifier system built in our GA-based learning environment. We used two data collections: glass data collection (GLD) and soybean diseases data collection (SBD). GLD contains 214 instances obtained from 6 different kinds of glass, in which each instance is represented with 9 numeric-valued attributes. SBD contains 290 instances obtained from soybean plants affected with one of fifteen diseases. Each instance is described by 35 attributes.

For each data collection, training/testing data sets were generated by equally dividing data set into two subsets. Then, base classifiers were learned by executing the learning system multiple times with training data set. In the following step, a genetic algorithm is applied to find the subset of base classifiers that provides the best performance. Finally, a multi-classifier system was constructed with base classifiers, and performance of it was evaluated with testing data set. This whole process was repeated five times.

For GLD, our approach improves classification performance by more than 11.5% for training data set and 4.9% for testing data set. For SBD, classification performance is increased quite significantly for both training and testing data set, improving performance of the system by more than 15.7% and 13.5%, respectively.

## 5 Conclusions

We have explored an approach for building a multi-classifier system in a MGA-based inductive learning environment. In our approach, several base classifiers are combined to build a multi-classifier system. A base classifier consists of a general classifier and a meta-classifier and the role of a meta-classifier is to evaluate classification result of its general classifier and decide whether the base classifier participates into a final decision-making process or not. Experiments reveal that our approach improves performance of a classification system quite significantly.

## References

1. Lorrentz, P., Howells, W. G. J., McDonald-Maier, K. D.: A Novel Weightless Artificial Neural Based Multi-Classifier for Complex Classifications. Neural processing letters, vol. 31, no. 1, pp. 25--44. (2010)
2. Ranawana, R., Palade, V.: Multi-classifier systems—review and a roadmap for developers, University of Oxford Computing Laboratory (24 Apr) (2006)

 

3. Doan, A., Domingos, P., Halevy, A.: Learning to Match the Schemas of Data Sources : A Multistrategy Approach, Machine Learning, vol. 50, no. 3, pp. 279--301 (2003)
4. Fan, D. W., Chan, P. K., Stolfo, S. J.: A Comparative Evaluation of Combiner and Stacked Generalization. In: IMLM-96, pp. 40--46 (1996)
5. W. Kim, W., Hong, C.: A Distributed Hybrid Algorithm for Composite Stock Cutting, Applied Mathematics and Information Sciences, vol. 6, no. 2s, (2012)
6. Duda, R., Hart, P., Nilsson, J.: Subjective Bayesian methods for rule-based inference systems, In: Proceedings of National Computer Conference, pp.1075--1082. (1976)

 