OPTIMIZATION FOR ENSEMBLE SYSTEM BASED ON MULTIPLE OBJECTIVES GENETIC APPROACH

TIEN THANH NGUYEN¹, ALAN WEE CHUNG LIEW¹, XUAN CUONG PHAM², MAI PHUONG NGUYEN³

¹ School of Information Technology, Griffith University, Australia.

² School of Information Technology, Hanoi University of Science and Technology, Vietnam.

³ College of Business, Massey University, New Zealand.

Email: tienthanh.nguyen2@griffithuni.edu.au

Abstract:

This paper introduces a mechanism to learn optimal combining classifier algorithms associated with features and classifiers from an ensemble system. By using Genetic Algorithm approach that focuses on 3 objectives namely number of correct classified observations and number of selected features and classifiers, optimal solution can be achieved after several interactions of crossover and mutation. We also employ OWA operator in which a weight vector is built by Linear Decreasing (LD) function to average outputs from combining algorithms. Experiments on 2 well-known UCI Machine Learning Repository datasets demonstrate benefits of our approach compared with other state-of-art ensemble method like Decision Template and SCANN as well as all fixed combining algorithms in the ensemble system.

Keywords:

Combining Classifiers; Ensemble Method; Genetic Algorithm; multiple objectives; OWA operator

1. Introduction

1.1 Combining Classifiers based on Stacking Algorithm

In this paper, we concentrate on ensemble models with fixed base classifiers and their combined outputs to make predictions for unlabeled observations in test set. Wolpert [1] introduced Stacking Algorithm to generate meta-data as output of base classifiers by dividing the original dataset to B disjointed parts in which each part plays as test set while the others as training set. Label of observations in test set will be predicted by model formed by base classifiers and training set. Several state-of-art combining algorithms have been proposed which operate on meta-data to generate hypothesis about relationship between feature vector and its class label. A simple combining approach that just work with meta-data of test set was introduced by Kittler [6]. It consists of 6 combining rules, namely Sum, Product, Vote, Min, Max, and Average. According to our knowledge, Vote and Sum are the most common rules. Kuncheva [2] introduced the concept of

Decision Template for each class generated from meta-data and the associated class label. She proposed 11 measurements between each Decision Template and meta-data of an observation to construct a classification framework. Meanwhile, Mezta [7] combined 3 algorithms namely Stacking, Correspondent Analysis and K Nearest Neighbor in a single algorithm called SCANN. His idea was to find relationship between rows (include all observations) and columns (include classification results of all classifiers). To implement it, he built an indicator matrix using {0, 1} labels and then used Singular Value Decomposition (SVD) transformation to make new scaled space. Ting [5] proposed Multiple Response Linear Regression algorithm (MLR) to combine posterior probabilities of each observation based on sum of weights calculated from K Linear Regression functions.

Recently, Szepannek [9] developed idea from pairwise combining by finding which classifier is best for a pair including Class i and Class j ($i, j = \overline{1, M}$ $i \neq j$) and used a pairwise coupling algorithm to combine outputs of all pairs to make posterior for each class. Zhang [10] used linear programming to find weight that each classifier puts on a particular class. Sen [8] introduced an approach inspired by MLR by applying hinge loss function to the combiner instead of using conventional least square loss. By using new function with regularization, he proposed three different combination, namely weighted sum, dependent weighted sum and linear stacked generalization based on different regularizations with group sparsity.

1.2. Genetic Approach in ensemble system

There are well-known optimization strategies applied to ensemble system to discover optimal subset of features and classifiers to improve the performance of classifier fusion. Here, we only discuss approaches based on Genetic Algorithm. Kuncheva [4] introduced two GA versions in which features are selected by join and disjoint mechanism. In

the former, she encoded feature by $\{0,1,...,K\}$ where k ($1 \leq k \leq K$) means that feature is only used by k^{th} classifier and 0 means that feature is not used by any classifiers. In the latter, she added classifier encoding in the same chromosome with feature encoding but both of them work independently in crossover and mutation stage. She developed an encoding method based on Venn diagram for feature encoding and integer values for classifier encoding. Nanni [11] employed GA to improve SCANN [7] by building representations where each includes encoding of M classes. Gabrys [12] tried to put classifier, feature and rule encoding in a single chromosome as 3-dimensional cube. These approaches, however, are difficult to implement because of the complicated crossover stage. We address these issues with a new GA model for multi classifier system which achieves both effectiveness and ease of implementation.

1.3. OWA operator

The Ordered Weighted Averaging operator (OWA) [16] is one of the most well-known operation applied commonly in Decision Making Systems. This operator has been using to compute average value based on weight but instead of focusing on original data, it is linked with order of data; as a result elements on specific location like head, tail or middle can receive more attention than the others.

An OWA operator of dimension P is a mapping from $\square^P \rightarrow \square$ that has an associated weight vector $\{\omega_i | i = \overline{1, P}\}$

such that $\omega_i \in [0,1]$ and $\sum_{i=1}^P \omega_i = 1$. The function

$F(a_i | i = \overline{1, P})$ determines aggregated value of the arguments

$A = \{a_i | i = \overline{1, P}\}$: $F = \sum_{i=1}^P b_i \omega_i$ where b_i is i^{th} largest value

in A .

An interesting thing to work with OWA operators is the type of weights to be used. We introduce four popular types of function to build the weight vector for each $k = \overline{1, K}$, which are Constant (eqn. 1), Gaussian (eqn. 2), Linear Increasing (eqn. 3) and Linear Decreasing (eqn. 4).

$$\omega(k) = \frac{1}{K} \quad (1)$$

$$\omega(k) = \frac{e^{-\left[\frac{(k-\mu)^2}{2\sigma^2}\right]}}{\sum_{k=1}^K e^{-\left[\frac{(k-\mu)^2}{2\sigma^2}\right]}} \quad (2)$$

$$\mu = \frac{K+1}{2} \text{ and } \sigma = \sqrt{\frac{1}{K} \sum_{k=1}^K (k-\mu)^2}$$

$$\omega(k) = \frac{2k}{K(K+1)} \quad (3)$$

$$\omega(k) = \frac{2(K+1-k)}{K(K+1)} \quad (4)$$

The novelty of our work is that in our ensemble, several fixed combining algorithms are present simultaneously. We propose a combining mechanism based on OWA operator to learning from training set to achieve:

- An optimal subset of combining algorithms
- An optimal subset of feature for each combination
- An optimal subset of classifier for each combination

Optimal solution resulted from multiple objective GA approach will be selected to predict label for all unlabeled observations.

2. Methodology

We denote N as the number of observations, K as the number of base classifiers, M as the number of classes $\{W_j\}$, D as dimension of the original data, and T as the number of combining classifiers. For each observation X_i , $P_k(W_j | X_i)$ is the probability that X_i belongs to class W_j given by k^{th} classifier. The meta-data (also called Level1 data) of all observations, a $N \times MK$ -posterior probability matrix $\{P_k(W_j | X_i)\}$ $j = \overline{1, M}$ $k = \overline{1, K}$ $i = \overline{1, N}$, is defined by:

$$\begin{bmatrix} P_1(W_1 | X_1) \dots P_1(W_M | X_1) & P_2(W_1 | X_1) \dots P_2(W_M | X_1) & \dots & P_K(W_1 | X_1) \dots P_K(W_M | X_1) \\ P_1(W_1 | X_2) \dots P_1(W_M | X_2) & P_2(W_1 | X_2) \dots P_2(W_M | X_2) & \dots & P_K(W_1 | X_2) \dots P_K(W_M | X_2) \\ \dots & \dots & \dots & \dots \\ P_1(W_1 | X_N) \dots P_1(W_M | X_N) & P_2(W_1 | X_N) \dots P_2(W_M | X_N) & \dots & P_K(W_1 | X_N) \dots P_K(W_M | X_N) \end{bmatrix} \quad (5)$$

Whereas Level1 data of an observation X is defined by:

$$Level1(X) \square \begin{bmatrix} P_1(W_1 | X) & \dots & P_1(W_M | X) \\ \vdots & \ddots & \vdots \\ P_K(W_1 | X) & \dots & P_K(W_M | X) \end{bmatrix} \quad (6)$$

Figure 1 shows the learning mechanism on training set. Our purpose here is to find which combining algorithms

associated with classifiers plus corresponding features are optimal for a particular training set. By applying GA on the training set, solution can be obtained through several interactions of crossover and mutation among individuals in population.

Figure 1. Training process to learn optimal combining algorithms associated with selected classifiers and features

Specifically, the structure of chromosomes is given by two-part architecture (Figure 2):

- Part1: encoding for combining algorithms by using $\{0, 1\}$ elements, its length is T
- Part2: encoding for classifiers including features used by them which are employed for a particular combining algorithms by using $\{0, 1\}$ elements, its length is $(K+K*D)*T$

Traditionally, researchers have only focus on the accuracy of a classification system. In our approach, the other

two objectives that are also considered are subset of features and classifiers used by the system. By reducing the number of selected features as well as classifiers, computation and storage cost will be saved. To achieve that, GA approach is employed with 3 objectives:

- Maximize number of correct classified observations in training set, denoted by $f1$
- Minimize number of features used by system (denoted by $f2$) so as to help reduce storage cost.

- Minimize number of classifiers used by system (denoted by f_3) so as to help reduce processing time as well as improve performance.

In this paper, we define an objective function formed by f_1 , f_2 and f_3 as $F = f(f_1, f_2, f_3) = w(1)f(1) - w(2)f(2) - w(3)f(3)$, in which $w = (w(1), w(2), w(3))$ is a weight vector

specified by users. To average the outputs from the selected combining algorithms, the fourth OWA function (eqn. 4) is chosen since we focus more on combining algorithm with higher accuracy. By using Linear Decreasing function, we put larger weights on higher accuracy algorithms so as to help increase converge speed in GA approach.

Figure 2. Propose structure of chromosome in GA approach

During testing, based on the encoding of optimal individual resulted from the training process we can determine:

- Which combining classifiers are being used or not used in ensemble system?
- Which classifiers plus their corresponding features are being used or not used for a particular selected classifier?

Features and classifiers are chosen corresponding with selected combining algorithms to classify each observation in test set to form Level1 data (eqn. 6). Next, this meta-data is employed as input to each selected combining algorithm to output class label as prediction of hypothesis associated with the algorithm. All outputs will be combined by a majority voting mechanism.

3. Experimental Results

In our experiment, we simply select the 6 simple rules namely Sum, Product, Max, Median, Min and Majority Vote as the fixed combining classifiers set in the model. As mentioned above, combining rules are very simple in computation compared with other approaches like Decision Template and SCANN. Here we want to show the benefit of our model, i.e. it has only simple combining algorithms but can achieve high accuracy. Three base classifiers are used: Linear Discriminant Analysis (LDA), Naïve Bayes, and KNN (with K set to 5, denoted as KNN(5)). We also initialize several parameters in GA, as detailed in TABLE 1. A 10-fold

cross validation is used since we do not have separate training set and test set. We run the experiment 10 times so in total we have 100 test outcomes for each dataset. Evaluation is performed on Matlab2013a environment.

We choose 2 popular UCI datasets [19] namely Sonar and Vehicle in evaluation (TABLE 2). To show the advantage of our model, we compare its error rate with other single combining algorithms, including the 6 combining rules, Decision Template [2], MLR [5], and SCANN [7]. Experimental results of all algorithms are shown in TABLE 3 and 4.

TABLE 1: INITIALIZATION VALUE OF PARAMETERS IN GA

Parameters	Value
Population Size	20
Mutation Rate (Part 1)	0.02
Mutation Rate (Part 2)	0.03
Max of interaction	50
Weight vector for multiple objectives	$w(1) = 1, w(2) = 0.1$ and $w(3) = 0.1$ (Sonar) $w(1) = 0.5, w(2) = 0.1$ and $w(3) = 0.1$ (Vehicle)

TABLE 2: INFORMATION OF 2 DATASETS IN EXPERIMENT

Name of Dataset	Sonar	Vehicle
Number of Observations	208	864
Number of Attributes	60	18
Feature Type	Real	Integer
Number of Classes	2	4

TABLE 3. ERROR RATES OF 6 COMBINING RULES ON 2 DATASETS

File name	Sum		Product		Max		Min		Median		Majority Vote	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Sonar	0.2259	9.55E-03	0.2285	9.81E-03	0.226	7.01E-03	0.2298	9.32E-03	0.2104	1.00E-03	0.2079	8.16E-03
Vehicle	0.2671	1.38E-03	0.2645	1.37E-03	0.2937	1.54E-03	0.2737	1.57E-03	0.2858	1.57E-03	0.3194	2.01E-03

TABLE 4. ERROR RATES OF OTHER COMBINING ALGORITHMS AND PROPOSED MODEL

File name	Decision Template		SCANN		MLR		Proposed model	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Sonar	0.2129	8.80E-03	0.2128	8.01E-03	0.1974	7.20E-03	0.1987	6.84E-03
Vehicle	0.2161	1.50E-03	0.2224	1.54E-03	0.2139	1.40E-03	0.2292	1.39E-03

TABLE 5. AVERAGE NUMBER OF SELECTED FEATURES AND CLASSIFIERS IN 100 TESTS

	# of selected features	# of selected classifiers
Sonar	37.18	1.62
Vehicle	16.66	1.57

Figure 3. Trend of combining algorithms selected among all experiments (Left) Vehicle (Right) Sonar

Remarkably, the proposed model is better than all combining algorithms in fixed set for both datasets. Comparing with other state-of-art algorithms like Decision Template, SCANN and MLR, our model is highly competitive. This is an excellent outcome since our model is formed by 6 simple combining rules (TABLE 3 and 4).

Another interesting point to note is the average number of features and classifiers chosen. Only 37.18 and 16.66 features are used by our model in the final solution among the entire 60 and 18 features on Sonar and Vehicle, respectively. Besides, our model only uses on average 1.62 and 1.57 classifiers from the 3 input classifiers on the 2 data files (TABLE 5).

In addition, the percentage of different combining algorithms used in 200 tests is illustrated in Figure 3. It is worth noticing that up to 198 cases, only single combining algorithm is chosen as optimal solution. Clearly, by cutting t smallest values from the sequence $a_1 \leq a_2 \leq \dots \leq a_T$, we will have a new sequence with higher average. Hence, if a chromosome has T algorithms ordered by correct number of

classified observations $a_1 \leq a_2 \leq \dots \leq a_T$, it may be replaced by chromosomes in the next generation with fewer algorithms by cutting the smallest values in the sequence. As a result, in the end, the final chromosome with optimal solution only has several algorithms with same accuracy. Hence, we use a special OWA operator to focus more attention on outstanding algorithms. This boosts the system's performance by increasing the speed of convergence in GA.

4. Conclusion and Future Development

In this paper, we have introduced a novel approach to learn optimal subset of combining classifier algorithms corresponding with subset of classifiers and features to construct a powerful and flexible ensemble system. We used multiple objectives GA to discover optimal solution of system architecture by proposing novel structure for chromosome and combined outputs from selected combination by OWA operator. Experiments conducted on two well-known datasets demonstrates the advantage of our

approach that optimal combining algorithms with optimal features and classifiers can be found.

Our work can be extended in several directions. For example, in scenarios where more than one combining algorithm is chosen, voting strategy plays an important role to gather hypotheses to form single one. Here we simply used majority vote strategy on the predictions of selected combining algorithms. To improve effectiveness and efficiency of final prediction, other methods such as Majority Teach Minority by Zhou [18] can be considered.

Another interesting point to be addressed is averaging outputs from combining classifiers to form an objective function in the fitness function. In our experiment, we used a particular OWA operator to compute average value for the correctly classified observations. An open question here is exploring new strategies to combine outputs from the entire fixed input combining algorithms.

References

- [1] D.H. Wolpert, Stacked Generalization, Neural Networks, Vol. 5, No. 2, pp. 241-259, Pergamon Press, 1992.
- [2] L. I. Kuncheva, James C. Bezdek and Robert P. W. Duin, Decision Templates for Multi Classifier Fusion: An Experimental Comparison, Pattern Recognition, 34, (2), pp.299-314, 2001.
- [3] L. I. Kuncheva, A theoretical Study on Six Classifier Fusion Strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.2, February 2002.
- [4] L. I. Kuncheva and Lakhmi C. Jain, Designing Classifier Fusion Systems by Genetic Algorithms, IEEE Transactions on Evolution Computation. Vol.4, No.4, September 2000.
- [5] Kai Ming Ting, Ian H. Witten, Issues in Stacked Generation, Journal of Artificial In Intelligence Research 10, pp. 271-289, 1999.
- [6] Josef Kittler, Mohamad Hatef, Robert P. W. Duin and Jiri Matas, On Combining Classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No.3, March 1998.
- [7] Christopher Merz, Using Correspondence Analysis to Combine Classifiers, Machine Learning 36, pp. 33-58, 1999.
- [8] Mehmet Umut Sen, Hakan Erdogan, Linear classifier combination and selection using group sparse regularization and hinge loss, Pattern Recognition Letters 34, pp. 265-274, 2013.
- [9] G. Szepannek, B. Bischl and C. Weihs, On the combination of locally optimal pairwise classifiers, Engineering Applications of Artificial Intelligent, Vol. 22, pp. 79-85, 2009.
- [10] Zhang L., Zhou W. D., Sparse ensembles using weighted combination methods based on linear programming, Pattern Recognition, Vol. 44, pp. 97-106, 2011.
- [11] Loris Nanni and Alessandra Lumini, A genetic encoding approach for learning methods for combining classifiers, Expert Systems with Applications 36, pp.7510-7514, 2009.
- [12] Bogdan Gabrys and Dymitr Ruta, Genetic Algorithms in classifier fusion, Applied Soft Computing 6, pp.337-347, 2006.
- [13] Micheal L.Raymer, William F.Punch, Erik D.Goodman, Leslie A.Kuhn and Anil K.Jain, Dimensionality Reduction using Genetic Algorithms, IEEE Transactions on Evolutionary Computation, Vol. 4, No. 2, July 2000.
- [14] Agapito Ledezma, Ricardo Aler, Araceli Sanchís and Daniel Borrajo, GA-stacking: Evolutionary stacked generalization, Journal Intelligent Data Analysis, Vol. 14, No. 1, pp. 89-119, Jan 2010.
- [15] Naif Alajlan, Yakoub Bazi, HaikelS.ALHichri, Farid Melgani, and Ronald R. Yager, Using OWA Fusion Operators for the Classification of Hyperspectral Images, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 6, No. 2, April 2013.
- [16] Dimitar Filev and Ronald R. Yager, On the issue of obtaining OWA operator weights, Fuzzy Sets and Systems, Vol. 94, pp.157-169, 1998.
- [17] Pasi Luukka and Onesfole Kurama, Similarity classifier with ordered weighted averaging operators, Expert Systems with Applications, Vol. 40, pp. 995-1002, 2013.
- [18] Z. H. Zhou and M. Li, Tri-training: Exploiting unlabeled data using three classifiers, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 11, pp. 1529- 1541, 2005.
- [19] UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml/datasets.html>.