A Meta-learning Approach that Learns the Bias of a Classifier

김영준* · 홍철의** · 김윤호***

Yeong-Joon Kim* · Chul-Eui Hong** · and Yoon-Ho Kim***

Abstract

DELVAUX is an inductive learning environment that learns Bayesian classification rules from a set of examples. In DELVAUX, a genetic algorithm approach is employed to learn the best rule-set, in which a population consists of rule-sets and rule-sets generate offspring by exchanging some of their rules. We have explored a meta-learning approach in the DELVAUX learning environment to improve the classification performance of the DELVAUX system. The meta-learning approach learns the bias of a classifier so that it can evaluate the prediction made by the classifier for a given example and thereby improve the overall performance of a classifier system. The paper discusses the meta-learning approach in details and presents some empirical results that show the improvement we can achieve with the meta-learning approach.

key words: genetic algorithms, inductive learning, Bayesian approaches, classifier systems, meta-learning approach

* 상명대학교 전자계산학과 전임강사
** 상명대학교 정보과학과 전임강사
*** 상명대학교 소프트웨어학과 전임강사

1. Introduction

It has been observed in various expert system projects that it is quite complicate and time-consuming to extract rules from experts, especially if their decision making uses approximate reasoning techniques (Eick and Metha, 1993). Over the last three years, we have developed an inductive learning environment called DELVAUX for classification tasks that learns PROSPECTOR-style, Bayesian rules from a set of examples (Eick et al. 1994, Kim and Eick 1995, Eick et al. 1996). The objective of the DELVAUX project is to explore to what extent machine learning techniques can be used for the rule generation process in expert systems. In DELVAUX, a genetic algorithm approach is employed to learn the best rule-set, in which a population consists of rule-sets and rule-sets participate in the breeding process in proportion to their fitness, generating offspring by exchanging some of their rules. Rule learning using genetic algorithms has been studied intensively in the fields of classifier systems(Wilson and Goldberg 1989, Janikow 1993) and designing fuzzy controllers(Mark and Vidal 1994, Kim and Li 1994). Experiments revealed that the DELVAUX system could produce a quite comparable learning performance to other inductive learning environments, such as neural networks and decision trees(Kim and Eick 1995, Eick et al. 1996).

Recently, several researchers have explored meta-learning approaches to combine several classifiers, each of which is computed by different algorithms, to improve the overall prediction accuracy(Zhang et al. 1992, Chan and Stolfo 1993). In their approaches, several classifiers are first learned from the same training set. Then, the next level classifier, a meta-classifier, is obtained from the predictions made by the classifiers on the examples in the training set and the correct classifications. When an instance is being classified, the classifiers first make their predictions. The predictions are then presented to the meta-classifier, which makes the final prediction. Empirical results reveal that their meta-learning approaches can improve the overall prediction accuracy of a classifier system.

Inspired by their works, we have explored a meta-learning approach in the DELVAUX learning environment. However, our approach is different from theirs in the sense that we use a meta-learning approach to obtain a meta-classifier that can decide whether the prediction made by a classifier is correct or not. In our approach, the DELVAUX system learns the classification behavior of a classifier, yielding a meta-classifier. Then, in the classification process, the meta-classifier is used to decide whether the prediction made by the classifier for a given example is correct or not. If the meta-classifier considers the prediction made by the classifier as incorrect one, the example is classified again by another classifier which is trained with examples that are classified incorrectly by the first classifier.

The paper is organized as follows. Section 2 briefly discusses the architecture of the DELVAUX learning environment, semantics of Bayesian rules, and genetic algorithms in DELVAUX. Section 3 explains the meta-learning approach. Section 4 presents some empirical results obtained with the meta-learning approach in the DELVAUX learning environ-ment. Finally, Section 5 concludes the paper.

2. The DELVAUX learning environment

The objective of this section is to give a brief description of the architecture of the DELVAUX learning environment, and to explain the semantics of Bayesian classification rules that the DELVAUX system tries to learn (a more detailed discussion of these topics has been given in (Eick et al. 1994, Kim and Eick 1995, Eick et al. 1996)).

2.1 The architecture of the DELVAUX learning environment

In the DELVAUX learning environment, the set of all available examples is partitioned into a training/test set pair. Similar to other inductive learning algorithms, we try to learn rule-sets using the training set. These rule-sets are then applied to examples belonging to the test set, evaluating the quality of the rules learned by the DELVAUX system. In DELVAUX, genetic algorithms are used to learn a Bayesian rule-set for classification tasks.

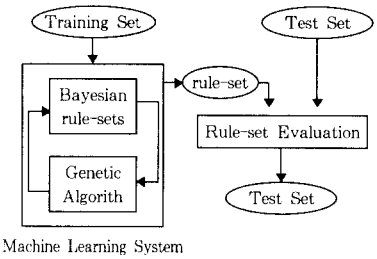

Figure 1. gives the architecture of the DELVAUX learning environment.

2.2 Bayesian Rules and Odds-Multipliers

The DELVAUX system learns PROSPECTOR-style rules (Duda et al. 1976) 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). If we have a rule, "If E then H with S=2.0; N=0.1", it expresses that the presence of E (P(E') = 1; P(E') denotes the posterior probability of E) increases the prior odds of H, O(H), by a factor 2, whereas the absence of E (P(E') = 0) decreases the prior odds of H by a

factor of 0.1. In the case that $P(E')$is neither 0.0 (yielding an odds-multiplier of N) nor 1.0 (resulting in an odds-multiplier of S), it becomes necessary to interpolate. For example, if $P(E')=0.3$, then an odds-multiplier between 0.1 and 2 has to be obtained; using a simple linear interpolation function as an example, it would be interpolated 0.7*N + 0.3*S = 0.7*0.1 + 0.3*2, yielding an odds-multiplier of 0.67 for the rule.

Two kinds of rules are employed in the DELVAUX learning environment: is-high rules and is-close-to rules. The syntax of is-high rule is

If is-high (*A*) *then D with S* = 3; *N* = 0.1

For a given example, the *is-high* rule produces an odds-multiplier between 3 and 0.1 based on the relative highness of the value for the attribute A of the 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 the closeness of the value for the attribute A to a certain constant a.

The posterior odds for a decision D, $O(D')$, is computed as follows in the DELVAUX system:

If the following rules provide evidence for the decision D

(r_1) If E_1 then D with $S=s_1$; $N=n_1$

...

(r_m) If E_m then D with $S=s_m$; $N=n_m$

then the posterior odds of D, $O(D')$, is computed as follows:

$$O(D') = O(D|E1' \wedge \ldots \wedge Em')$$
$$= O(D) * \prod^{m}_{i=1} \lambda_i$$

where $\lambda_i = O(D|E_i')/O(D)$ is the odds-multiplier of the rule r_i.

Finally, we have to discuss how decisions are chosen by Bayesian rule-sets. Assuming that we have decision candidates $DC=\{D_1,\ldots,D_n\}$ with prior odds $O(D_1),\ldots,O(D_n)$, these odds are updated by firing rules of the rule-set, yielding posterior odds $O(D_1'),\ldots,O(D_n')$; finally, the decision with the highest posterior odds is selected. If the decision chosen by the rule-set is the same as the classification of a given example, then we say that the rule-set classified the example correctly; otherwise, we say that it classified the example incorrectly.

2.3 Genetic Algorithms for Learning Bayesian Rules

In DELVAUX, a population consists of a fixed number of rule-sets and rule-sets themselves are represented in their chromosomal representation as ordered sequences of rules $r_1,\ldots,r_n$. The first generation is generated randomly. In the following evolution process, the next generation is created by applying genetic operators to the current population. In DELVAUX, 1-point crossover operators, mutation operators and inversion operators are used as genetic operators. The 1-point crossover operator creates two offspring by

exchanging some rules of the selected parents as depicted in Figure 2(a). The mutation operator selects a rule r from a rule-set and replaces it with a newly generated rule r' (see Figure 2.(b)). Inversion operators are used to change the order of rules in the chromosomal representation of a rule-set (see Figure 2.(c)).

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. During the evolution process, DELVAUX monitors the average fitness of the current generation and the maximum fitness of the current generation. If none of them has been increased by a certain amount in a certain number of generation, DELVAUX terminates its learning process.

3. A Meta-learning Approach to Learn the Bias of a Classifier

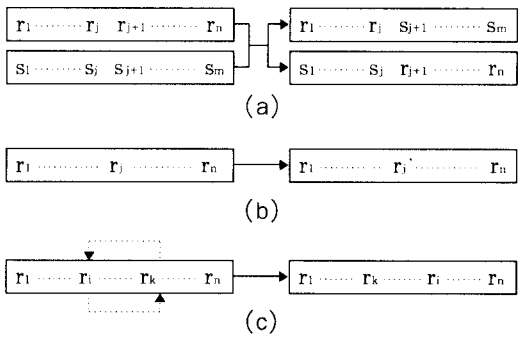

Figure 2. Genetic operators. (a)Crossover operator (b)Mutation operator (c)Inversion operato

The meta-learning approach is implemented with three classifiers CF_1, MC, and CF_2. The classifier CF_1 is a classifier obtained by running the DELVAUX system for a given training data set. The classifier CF_1 learns concept descriptions from given training data set to perform regular classification task. The classifier MC is a meta-classifier. The meta-classifier MC learns the classification behavior (i.e., the bias) of the CF_1 so that it can decide whether the prediction made by CF_1 is correct or not. The third classifier CF_2 is trained with the set of examples that are classified by MC as incorrectly classified examples by CF_1. The function of CF_2 is to classify examples that are possibly misclassified by CF_1 again, giving a second chance to them.

In the meta-learning approach, the training process consists of three phases. In the first training phase, the classifier CF_1 is learned from the given training data set. The training data set for the classifiers CF_1 (denoted by TCF_1) is a regular training data set, in which each example is represented with a list of attribute values and the classification of the example. In the next training phase, the meta-classifier MC is trained for the training data (denoted by TMC), in which each element is represented with the attribute values of an example in TCF_1, the prediction made by CF_1 for the example, and the classification of the prediction of CF_1 which is either 0 or 1, where 0 represents incorrect decision and 1 represents correct decision. Finally, in the third training phase, the classifier CF_2 is

trained for the examples that MC classifies as incorrectly classified examples by CF_1. In the data set for the CF_2 (denoted by TCF_2), each element is represented with the attribute values of an example in TCF_1, the prediction of CF_1, and the classification of the example. Figure 3 depicts the training phase of the meta-learning approach.

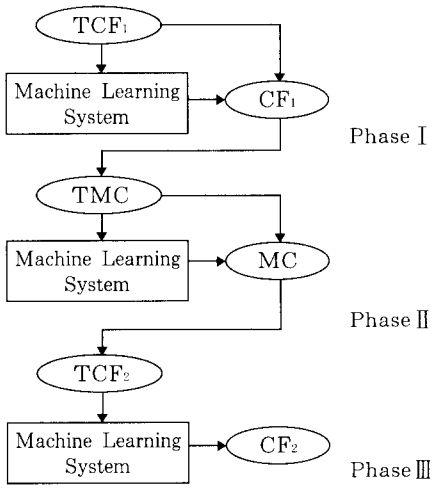

Figure 3 The training phase of the meta-learning approach.

In the meta-learning approach, the system classifies a given unknown example t_e as follows:

Step 1. The classifier CF_1 classifies the given example t_e.

Step 2. The classifier MC decides the correctness of the prediction of CF_1.

Step 3. If MC classifies the prediction of CF_1 as correct, the system returns the prediction of CF_1 as the final prediction. Otherwise, the example te along with the prediction of CF_1 is turned into CF_2.

Step 4. CF_2 classifies the given example and the system selects the decision of CF_2 as its final decision.

Figure 4 depicts the classification process of the meta-learning approach. In Figure 4, D_{CF1} denotes the prediction of CF_1.

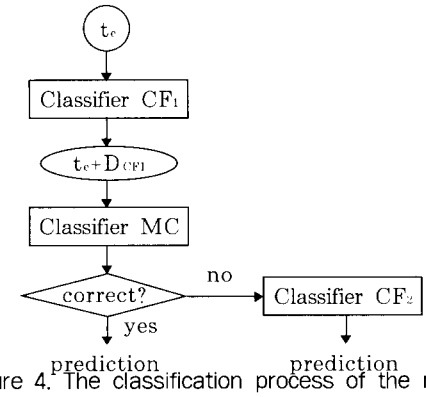

Figure 4. The classification process of the meta-learning approach

4. Empirical Results

Some experiments were performed to evaluate the performance of the meta-learning approach in the DELVAUX learning environment. We used two data collections: the glass data collection (GL) and the soybean diseases data collection(SBD). The characteristics of each data collection are as follows:

- Glass data collection (GL):the data set contains 214 instances obtained from 6 different kinds (building-windows-float processed, building-windows-non-float processed, containers,...)of glass, in which each instance is represented with 9 numeric-valued attributes (Na, Fe, K, ...).

- Soybean diseases data collection (SBD)
  :each of fifteen diseases that affect soybean plants is described by thirty five attributes. The data set contains 290 instances.

For each data set, training/testing data set are generated by equally dividing the data set into two subsets. Then, for each training/testing data set, the classifier CF_1 is learned first using the DELVAUX system. In the following steps, the meta-classifier MC and the classifier CF_2 are learned and the performance of the basic DELVAUX system (i.e., CF_1) and that of the meta-learning approach are evaluated. This whole process is repeated five times. The following parameters were used in these experiments:

- population size : 15 rule-sets (constant)
- upper limit of the rule-set size : 50 rules
- lower limit of the rule-set size : 3 rules
- inversion rate : 5% (constant)
- crossover rate : 98% (constant)
- mutation rate : 40% (constant)
- maximum number of generations : 2000

The results of experiments are depicted in the Table 1 and Table 2. In the tables, "Basic DELVAUX" denotes the performance of the basic DELVAUX system depicted in the section 2 and "Meta DELVAUX" denotes the performance of the meta-learning approach. For the glass data collection, the meta-learning approach improves the performance of DELVAUX by more than 11.2% for training and by 1.0% for testing. For the soybean diseases data collection, the performance of DELVAUX is increased quite significantly for both training and testing, improving the performance of the system by more than 25% and 20%, respectively.

Application	Basic DELVAUX		Meta DELVAUX	
	TRAIN	TEST	TRAIN	TEST
GL1	0.7248	0.6286	0.7798	0.6381
GL2	0.7156	0.6190	0.8349	0.6381
GL3	0.7064	0.5905	0.8165	0.6190
GL4	0.6972	0.6286	0.8349	0.6095
GL5	0.6972	0.6286	0.8349	0.6381
AVG	0.7082	0.6191	0.8202	0.6290

Table 1. Basic DELVAUX vs. DELVAUX with Meta-learning (GL)

Application	Basic DELVAUX		Meta DELVAUX	
	TRAIN	TEST	TRAIN	TEST
SBD1	0.6897	0.5517	0.8897	0.7586
SBD2	0.6483	0.5310	0.8828	0.7103
SBD3	0.6345	0.5172	0.8690	0.7172
SBD4	0.6138	0.5172	0.8966	0.7310
SBD5	0.5862	0.5241	0.8690	0.7172
AVG	0.6345	0.5282	0.8814	0.7269

Table 2. Basic DELVAUX vs. DELVAUX with Meta-learning (SBD)

5. Conclusions

A meta-learning approach has been explored. In the meta-learning approach, a meta-classifier learns the bias of a classifier so that it can evaluate the prediction made by the classifier for a given example. In the meta-learning approach, the prediction of the classifier for an unknown example is evaluated by the meta-classifier. If the meta-classifier classifies the prediction as correct one, the prediction of the classifier is accepted as the final decision of the classifier system. Otherwise, the example is classified again by the second classifier that is trained with examples that are misclassified by the first classifier.

Experiments revealed that the meta-learning approach improved the performance of the DELVAUX system quite significantly. Even though, we have discussed our meta-learning approach in the DELVAUX learning environment, we can easily apply our meta-learning approach to other inductive learning environments such as neural networks or decision trees learning environments.

In the meta-learning approach, the performance of the system depends on the performance of the meta-classifier. As a future work, it might be interesting to develop a system that can bias the first classifier in a certain direction so that the meta-classifier can achieve best performance. Moreover, further study should be performed to find a better way to construct the training data set for the CF_2.

References

Chan, P. and S. Stolfo, "Experiments on multistrategy learning by meta-learning," in *Proceedings of the Second International Working Notes on Multistrategy Learning*, 1993, pp. 314-323

Duda, R., P. Hart and J. Nilsson, "Subjective Bayesian methods for rule-based inference systems," in *Proc. National Computer Conference*, (1976), 1075-1082.

Eick, C. F. and N. Metha, "Decision making involving imperfect knowledge," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-23, (1993), 840-850.

Eick, C. F., Y-J. Kim, and N. Secomandi, "Enhancing diversity for a genetic algorithm learning environment for classification tasks," in *Int. Conf. on Tools with Artificial Intelligence*, New Orleans, Nov. (1994).

Eick, C. F., Y-J. Kim, N. Secomandi, and E. Toto, "DELVAUX-An Environment that Learns Bayesian Rule-sets with Genetic Algorithms," in *The Third World Congress on Expert Systems*, Seoul, Korea, Feb. (1996).

Janikow, C, "A knowledge-intensive genetic algorithm for supervised learning," *Machine Learning*, vol. 13, (1993), 189-228.

Kim, Y-J. and C. F. Eick, "Multi-rule-set decision-making schemes for a genetic algorithm learning environment for classification tasks," in *The 4th Annual Conference on Evolutionary Programming*, (1995), 773-788.

Kim, C. and Y. Li, "Design of sophisticated

fuzzy logic controllers using genetic algorithms," in *Third IEEE International Conference on Fuzzy Systems*, (1994).

Mark, G. Cooper and Jacques Vidal, "Genetic design of fuzzy controllers: The cart and jointed pole problem," in *Third IEEE International Conference on Fuzzy Systems*, (1994).

Wilson, S. and D. Goldberg, "A critical review of classifier systems," *in Proc. 3rd Int. Conf. Genetic Algorithms*, Morgan Kaufmann Publishers, (1989), 244-255.

Zhang, X., J. Mesirov and D. Waltz, "A hybrid system for protein secondary structure prediction," *Journal of Molecular Biology*, 1992, vol. 225, pp. 1049-1063