Data Classification Using Genetic Parallel Programming

Sin Man Cheang, Kin Hong Lee, and Kwong Sak Leung

Department of Computer Science and Engineering
The Chinese University of Hong Kong, Hong Kong
{smcheang,khlee,ksleung}@cse.cuhk.edu.hk

Abstract. A novel Linear Genetic Programming (LGP) paradigm called Genetic Parallel Programming (GPP) has been proposed to evolve parallel programs based on a Multi-ALU Processor. It is found that GPP can evolve parallel programs for Data Classification problems. In this paper, five binary-class UCI Machine Learning Repository databases are used to test the effectiveness of the proposed GPP-classifier. The main advantages of employing GPP for data classification are: 1) speeding up evolutionary process by parallel hardware fitness evaluation; and 2) discovering parallel algorithms automatically. Experimental results show that the GPP-classifier evolves simple classification programs with good generalization performance. The accuracies of these evolved classifiers are comparable to other existing classification algorithms.

Data Classification is a supervised learning process that learns a classifier from a training set. The learned classifier can be used to classify unseen data records. Lim *et al.* have performed a sophisticated study on 16 UCI Machine Learning Repository databases by 33 different data classification algorithms [1]. Their experimental results are used for comparison with the proposed GPP-classifier. A novel LGP paradigm – Genetic Parallel Programming (GPP) [2,3] is employed to learn data classifiers. In GPP, individual programs are represented in a sequence of parallel instructions. Each parallel instruction consists of multiple subinstructions in order to perform multiple operations in each processor clock cycle simultaneously. A parallel program is executed on a specially designed Multi-ALU Processor (MAP). The main purpose of this paper is to demonstrate that GPP can evolve data classifiers to solve real-world data classification problems. Experimental results show that GPP can evolve binary-class data classifiers with comparable generalization accuracy to the other 33 existing data classification methods presented in [1].

We adopt the 10-fold cross-validation method to estimate the classification error rate (CE) of the GPP-classifier. 10 training sets are used to learn 10 classifiers that are tested with their corresponding test sets to obtain 10 test set CE. The 10 test set CE are averaged to estimate the generalized CE. We measure the classification accuracy and the generalization performance. A good generalized classifier gives similar levels of performance on the training and test sets. Furthermore, the GPP-classifier has adopted three techniques to avoid overtraining: 1) limiting the size of genetic programs; 2) penalizing over-trained individual programs; and 3) monitoring generalization performance over the evolution. All experiments have been run on a software GPP-classifier system. It produces a parallel assembly program together with a corre-

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1918–1919, 2003.

spondent serialized C code segment. Table 1 below shows the best, average, and standard deviation (*stddev*) of training set CE and test set CE of 10 independent runs (10-fold cross-validation on each run).

Table 1. Training set CE and test set CE of the GPP-classifier

	training set CE (%)			test set CE (%)			
	best	*average*	*stddev*	*best*	*average*	*stddev*	*%ΔCE*
bcw	2.7	2.9	0.09	3.5	3.9	0.29	25.6%
bld	27.3	28.0	0.69	29.3	31.7	1.74	11.7%
pid	22.5	22.7	0.11	23.7	24.5	0.42	7.3%
hea	14.4	14.8	0.24	16.0	18.9	1.78	21.7%
vot	3.9	4.1	0.10	4.1	4.6	0.23	10.8%
average							15.4%

In Table 1 above, the last column shows the percentage differences (*%ΔCE*) of the average training set CE and test set CE. The average *%ΔCE* of the five databases is 15.4%. It is shown that GPP can learn parallel programs to solve real-world data classification problems. Experimental results show that GPP is able to learn human understandable classifiers with comparable generalization performance to other classification algorithms. Even without tailor-making the GPP configurations for individual problem, good quality classifiers are evolved. The generalization performance of the GPP-classifier is higher than the average of the 33 benchmark algorithms in [1]. It shows that the GPP-classifier has the power to learn very simple but accurate classifiers with a suitable overtraining control strategy. Besides, the GPP-classifier automatically determines the structure of the solution program without prior knowledge of the databases. Even though the results show that classification program evolved by GPP has comparable generalization performance to other classification algorithms, further improvements can be carried out. In spite of adopting overtraining control strategies, the GPP-classifier still suffers from some overtraining, i.e. the training set CE are higher than test set CE in Table 1 above. In order to obtain a good generalization performance, we shall work out an appropriate terminating condition to detect overtraining.

References

1. Lim, T.S., Loh, W.Y., Shih, Y.S.: A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. Machine Learning Journal, Vol.40, Kluwer Academic (2000) 203–229
2. Leung, K.S., Lee, K.H., Cheang, S.M.: Evolving Parallel Machine Programs for a Multi-ALU Processor. Proc. of IEEE Congress on Evolutionary Computation (2002) 1703–1708
3. Leung, K.S., Lee, K.H., Cheang, S.M.: Genetic Parallel Programming – Evolving Linear Machine Codes on a Multiple-ALU Processor. Proc. of International Conference on Intelligence in Engineering and Technology, Univ. Malaysia Sabah (2002) 207–213